

Marcos Paulo Moro

**Eficiência Energética e Qualidade de Serviço
em Centros de Dados**

**Campo Grande, MS
6 de novembro de 2019**

Marcos Paulo Moro

Eficiência Energética e Qualidade de Serviço em Centros de Dados

Universidade Federal de Mato Grosso do Sul
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação
Doutorado em Ciência da Computação

Orientador: Ronaldo Alves Ferreira
Coorientador: Renato Porfírio Ishii

Campo Grande, MS
6 de novembro de 2019

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus, por ser essencial em minha vida, aos meus filhos Marcos Caio Lopes Moro, Gabriella Lopes Moro e Barbara Lopes Moro, minha mãe Maria Aparecida Delmonico, minha irmã Andréia Moro, minha tia Neuza Delmonico e minha amada namorada Solange Tieko Sakaguti.

AGRADECIMENTOS

Aos meus filhos pelo apoio dado durante o desenvolvimento deste trabalho, tendo sempre a compreensão da minha ausência durante todo o período de estudos.

À minha namorada, Solange Tiekko Sakaguti, pela compreensão e motivação dada nos meses de escrita da tese.

Ao meu pai de coração José Carlos (in memoriam), com saudades e toda a minha gratidão.

Em especial, ao Professor Ronaldo Alves Ferreira, por seu papel fundamental no desenvolvimento dos estudos e pela orientação, seu grande desprendimento em ajudar-nos e amizade sincera.

Ao meu coorientador, Professor Renato Porfirio Ishii, pelo tempo e apoio dado durante o doutorado.

À professora Edna Ayako Hoshino, pela orientação no desenvolvimento da modelagem do problema de mapeamento de VMs.

Aos professores Cristiano Costa Argemon Vieira, Fábio Moreira Costa, Fabrício Sérgio de Paula, Lisandro Zambenedetti Granville, Marcelo Henriques de Carvalho e Marcos Mansano Furlan por aceitarem participar no comitê de avaliação desta tese.

À Fundação de Apoio ao Desenvolvimento do Ensino, Ciência e Tecnologia do Estado de Mato Grosso do Sul (Fundect) pelo apoio financeiro recebido para o desenvolvimento desta tese.

"Seja lá o que você fizer, seja bom nisso."

Abraham Lincoln

Eficiência Energética e Qualidade de Serviço em Centros de Dados

Autor: Marcos Paulo Moro

Orientador: Dr. Ronaldo Alves Ferreira

Coorientador: Renato Porfírio Ishii

Os centros de dados estão entre os maiores consumidores de energia elétrica da área de tecnologia da informação. O consumo de um único centro de dados chega a ser superior ao de uma cidade com 100 mil habitantes. Centros de dados não são famosos apenas por consumirem grandes quantidades de energia, mas também por serem ineficientes. Estudos recentes mostram que, em média, mais de 50% da energia consumida por um centro de dados são desperdiçados por causa de gerenciamento ineficiente de seus recursos.

O consumo energético de um centro de dados é dominado por três dimensões: servidores, sistema de refrigeração e infraestrutura de rede. As propostas para gerenciamento energético existentes na literatura, normalmente, buscam otimizar uma ou duas dessas dimensões sem levar em consideração as demais. Além disso, elas não consideram como essas otimizações para economia de energia impactam os parâmetros de desempenho do centro de dados.

Este trabalho primeiro propõe uma arquitetura de gerenciamento e simulação de centros de dados para estudos de eficiência energética. Com base nessa arquitetura, este trabalho desenvolve o simulador SimDC3D que simula realisticamente os subsistemas de maior consumo de energia elétrica de um centro de dados e como algoritmos de escalonamento e realocação de máquinas virtuais, de controle do sistema de refrigeração e de otimização de topologia de rede interferem em parâmetros de desempenho.

Este trabalho também propõe dois novos algoritmos de escalonamento e consolidação de máquinas virtuais, cada um com quatro variações, um algoritmo para controle do sistema de refrigeração e um algoritmo de otimização da topologia da rede do centro de dados. Os

algoritmos de escalonamento são altamente parametrizáveis para que um operador de centro de dados possa otimizar uma das dimensões de forma mais agressiva do que outras, ou seja, para que ele possa atribuir pesos diferentes para os subsistemas. Além disso, este trabalho faz uma avaliação extensiva, com centenas de simulações, dos algoritmos propostos usando diferentes cargas de trabalho e topologias de rede e os compara com outras propostas existentes na literatura. Os resultados mostram a importância de se parametrizar adequadamente os algoritmos de otimização para que eles não apenas economizem energia, mas também evitem violações de acordos de nível de serviço estabelecidos entre operadores de centro de dados e seus clientes.

Palavras-chave: Centro de Dados, Simulação, Economia de Energia.

Energy Efficiency and Quality of Service in Data Centers

Author: Marcos Paulo Moro

Advisor: Dr. Ronaldo Alves Ferreira

Co-advisor: Dr. Renato Porfírio Ishii

Data centers are among the largest consumers of electric energy in the information technology area. The energy consumption of a single data center can be larger than the consumption of a city with 100 thousands inhabitants. Data centers are not only famous for consuming large quantities of energy but also for being inefficient. Recent studies show that, on average, more than 50% of the energy consumed by a data center are wasted because of inefficient management of its resources.

The energy consumption of a data center is dominated by three dimensions: servers, cooling system, and network infrastructure. The current proposals for energy management, generally, try to optimize one or two of these dimensions without considering the others. Moreover, they do not consider how these energy reduction optimizations impact performance parameters of the data center.

This work first proposes an architecture for management and simulation of data centers for energy efficiency studies. Based on this architecture, this work develops the simulator SimDC3D that simulates realistically the subsystems that consume the most energy in a data center and how algorithms for scheduling and relocation of virtual machines, controlling the cooling system, and optimizing the network topology interfere with the performance parameters of the data center.

This work also proposes two new algorithms for scheduling and relocation of virtual machines, each one with four variations, one algorithm for controlling the cooling system, and one algorithm for optimizing the network topology of the data center. The scheduling and relocation algorithms are highly parameterized so that a data-center operator can optimize one

of the dimensions more aggressively than the others, *i.e.*, he can assign different weights to the subsystems. Moreover, this work does an extensive evaluation, with hundreds of simulations, of the proposed algorithms under different workloads and topologies and compares them with the existing proposals. The results show the importance of setting the appropriate parameters for the optimization algorithms so that they can not only save energy but also avoid violations of service-level agreements settled between data center operators and their customers.

Keywords: Data Center, Simulation, Power Saving.

LISTA DE FIGURAS

Figura 2.1 – Exemplo de mapa térmico de um centro de dados gerado pelo simulador CoolSim (COOLSIM, 2019).	12
Figura 2.2 – Topologia de rede de 3 níveis.	15
Figura 2.3 – Exemplo de rede Clos entre comutadores de agregação e núcleo fornecendo um <i>backbone</i> ricamente conectado e adequado para VLB. Redesenhado de (GREENBERG et al., 2009).	16
Figura 2.4 – Exemplo de um topologia Portland com $k = 4$. Redesenhado de (BOLLA et al., 2009).	17
Figura 2.5 – (a) A $BCube_k$ é construída a partir de n $BCube_{k-1}$ e n^k comutadores com n -port. (b) Um $BCube_1$ com $n = 4$. Nesta rede $BCube_1$, cada servidor tem duas portas, extraído de (GUO et al., 2009).	17
Figura 2.6 – Topologia utilizada pela arquitetura DCell, extraído de (COUTO; CAMPISTA; COSTA, 2012).	18
Figura 4.1 – Proposta de uma arquitetura de gestão energética centralizada.	35
Figura 4.2 – Arquitetura do SimDC3D.	39
Figura 4.3 – Arquitetura FNSS redesenhada de (SAINO; COCORA; PAVLOU, 2013).	43
Figura 4.4 – Arquivo de configuração do simulador FNSS.	44
Figura 4.5 – Configurações para a matriz de tráfego a ser gerada pelo FNSS.	45
Figura 4.6 – Diagrama do fluxo de controle das migrações de VMs no SimDC3D.	47
Figura 4.7 – Diagrama de transições dos servidores no SimDC3D.	49
Figura 4.8 – Modelo dos fluxos de dados do SimDC3D.	53
Figura 4.9 – Estudo de caso de otimização na infraestrutura de rede.	57
Figura 4.10–Estatísticas extraídas de simulações do SimDC3D.	59
Figura 4.11–Estatísticas extraídas de simulações do SimDC3D.	60
Figura 4.12–Estatísticas extraídas de simulações do SimDC3D.	61

Figura 4.13–Estatísticas de utilização do centro de dados.	62
Figura 5.1 – Relação entre expoente e <i>ranking</i> da variável temperatura.	70
Figura 6.1 – Topologias Simuladas nos Estudos de Caso.	92
Figura 6.2 – Cargas de trabalho aplicadas no centro de dados.	93

LISTA DE TABELAS

Tabela 1.1 – Características de simuladores para centro de dados.	5
Tabela 3.1 – Características de simuladores para centro de dados versus SimDC3D. . . .	31
Tabela 6.1 – Características dos traços utilizados nos estudos de caso do SimDC3D. . . .	92
Tabela 6.2 – Desempenho do centro de dados em função da redução do número de servidores no banco.	99
Tabela 6.3 – Desempenho do centro de dados x controle de temperatura dinâmico.	100
Tabela 6.4 – Desempenho do centro de dados x estratégias de alocação de VMs para os traços LANL-CM5 e METACENTRUM.	103
Tabela 6.5 – Desempenho do centro de dados x estratégias de alocação de VMs para os traços SHARCNET e UNILU-GAIA.	104
Tabela 6.6 – Desempenho do centro de dados x estratégias de alocação de VMs para o traço RICC.	105
Tabela 6.7 – Consumo energético da infraestrutura de rede sem otimização.	106
Tabela 6.8 – Desempenho da infraestrutura de rede com otimização x estratégias de alocação de VMs.	106
Tabela 6.9 – Desempenho da infraestrutura de rede com otimização e ALR x estratégias de alocação de VMs.	107
Tabela 6.10–Desempenho do preditor de CPU.	108
Tabela 6.11–Raiz do erro quadrático médio da variável temperatura.	111
Tabela 6.12–Erro médio dos valores preditos em relação ao real da variável temperatura.	112
Tabela 6.13–Desvio padrão dos valores preditos em relação ao real da variável temperatura.	112
Tabela 6.14–Coeficiente de variação dos valores preditos em relação ao real da variável temperatura.	113
Tabela 6.15–Raiz do erro quadrático médio da variável temperatura para CLENGTH de 5 a 20 para os algoritmos RBF e Polynom.	114

Tabela 6.16–Erro médio da variável temperatura para CLENGTH de 5 a 20 para os algoritmos RBF e Polynom.	115
Tabela 6.17–Desvio padrão da variável temperatura para CLENGTH de 5 a 20 para os algoritmos RBF e Polynom.	116
Tabela 6.18–Coeficiente de variação da variável temperatura para CLENGTH de 5 a 20 para os algoritmos RBF e Polynom.	117
Tabela 6.19–Desempenho de um centro de dados tradicional x centro de dados verde. . .	118

LISTA DE SIGLAS E ACRÔNIMOS

ALR *Adaptive Link Rate*

API Interface de Programação de Aplicações (*do inglês Application Programming Interface*)

ASHRAE *American Society of Heating, Refrigerating and Air Conditioning Engineers*

CFD Dinâmica de Fluidos Computacional (*do inglês Computational Fluid Dynamics*)

CPU Unidade Central de Processamento (*do inglês Central Processing Unit*)

DCiE *Datacenter Infrastructure Efficiency*

DPR Desvio Padrão Relativo

DVFS *Dynamic Voltage and Frequency Scaling*

FNSS *Fast Network Simulation Setup*

HRM Matriz de Recirculação de Calor (*do inglês Heat Recirculation Matrix*)

IaaS Infraestrutura como Serviço (*do inglês Infrastructure as a Service*)

kW Quilowatt

MIPS Milhões de Instruções Por Segundo

MRAM Memória de Acesso Aleatório Magnetoresistive (*do inglês Magneto-resistive Random Access Memory*)

MSE Erro Quadrático Médio (*do inglês Mean Square Error*)

PABFD Power Aware Best Fit Decreasing

PaaS Plataforma como serviço (*do inglês Platform as a Service*)

PCS Ajuste e Limitação de Potência (*do inglês Power Capping and Shifting*)

PUE *Power Usage Effectiveness*

QoS Qualidade de Serviço (*do inglês Quality of Service*)

RMSE Raiz do Erro Quadrático Médio (*do inglês Root Mean Square Error*)

SaaS Software como Serviço (*do inglês Software as a Service*)

SF Servidores Físicos

SLA Acordo de Nível de Serviço (*do inglês Service-Level Agreement*)

ST *Single Thresold*

SWF Formato de Carga de Trabalho Padrão (*do inglês Standard Workload Format*)

TFP *Total Facility Power*

TI Tecnologia da Informação

tPUE *Total Facility Power*

UPS *Uninterruptible Power Supply*

VM Máquinas Virtuais (*do inglês Virtual Machine*)

XML Linguagem de Marcação Extensível (*do inglês Extensible Markup Language*)

LISTA DE ALGORITMOS

1	ALGORITMO DE CÁLCULO DE VARIABILIDADE DE UNIDADE CENTRAL DE PROCESSAMENTO (<i>do inglês Central Processing Unit</i>) (CPU)	42
2	ALGORITMO DE CÁLCULO DOS PERÍODOS ON/OFF DE TRANSMISSÃO DOS ENLACES	54
3	ALGORITMO 3DMOBFD - COMPLEXIDADE $O(n^2)$	69
4	ALGORITMO 3DMOBFD_PREDICTION - COMPLEXIDADE $O(n^4)$	70
5	ALGORITMO 3DMOBFD_POOL - COMPLEXIDADE $O(n^2)$	71
6	ALGORITMO 3DMOBFD_POOL_AND_PREDICTION - COMPLEXIDADE $O(n^4)$	72
7	ALGORITMO 2D - COMPLEXIDADE $O(n^2)$	73
8	ALGORITMO 2D_PREDICTION - COMPLEXIDADE $O(n^4)$	74
9	ALGORITMO 2D_POOL - COMPLEXIDADE $O(n^2)$	75
10	ALGORITMO 2D_POOL_AND_PREDICTION - COMPLEXIDADE $O(n^4)$	76
11	ALGORITMO BEST_PERFORMANCE - COMPLEXIDADE $O(n^2)$	77
12	ALGORITMO LOW_TEMP_FIRST - COMPLEXIDADE $O(n^2)$	77
13	ALGORITMO RANDOM - COMPLEXIDADE $O(n)$	78
14	ALGORITMO MIN_HR - COMPLEXIDADE $O(n^2)$	79
15	ALGORITMO CENTER_RACK_FIRST - COMPLEXIDADE $O(n^2)$	80
16	ALGORITMO 3DMOBFD - COMPLEXIDADE $O(n^2)$	82
17	ALGORITMO 2D - COMPLEXIDADE $O(n^2)$	83
18	ALGORITMO HIGH_TEMPERATURE - COMPLEXIDADE $O(n^2)$	84
19	ALGORITMO FFD - COMPLEXIDADE $O(n^2)$	84
20	ALGORITMO PABFD - COMPLEXIDADE $O(n^2)$	85
21	ALGORITMO PADRÃO DO SIMWARE - COMPLEXIDADE $O(n^2)$	86
22	ALGORITMO AGRESSIVO DO SIMDC3D - COMPLEXIDADE $O(n^2)$	87
23	ALGORITMO OTIMIZAÇÃO TOPOLOGIA DE REDE - COMPLEXIDADE $O(n^2)$	89

1	Introdução	1
1.1	Consumo Energético em Centros de Dados	1
1.2	Redução do Consumo Energético em Centros de Dados	2
1.3	Simulações de Centros de Dados	3
1.4	Qualidade de Serviço	4
1.5	Contribuições	6
1.6	Organização da Tese	7
2	Conceitos Básicos	8
2.1	Centro de Dados Verde	8
2.2	Estratégias de Economia de Energia para Servidores	10
2.2.1	Computação com Energia Proporcional	10
2.2.2	Provisionamento Dinâmico	11
2.2.3	Virtualização	11
2.2.4	Ajuste e Limitação de Potência	11
2.3	Estratégias de Economia de Energia para o Sistema de Refrigeração	12
2.4	Estratégias de Economia de Energia para a Infraestrutura de Rede	13
2.4.1	Topologias de Rede para Centro de Dados	15
2.4.1.1	VL2	15
2.4.1.2	Portland e FatTree	16
2.4.1.3	BCube	16
2.4.1.4	DCell	17
2.4.2	Comparação de Custos entre Topologias de Rede para Centro de Dados	19
2.5	Conceitos de Séries Temporais	19
2.6	Estatísticas para Análise de Desempenho de Preditores em Simulações para Centro de Dados	21

2.6.1	Raiz do Erro Quadrático Médio (RMSE)	21
2.6.2	Coefficiente de Variação	22
2.7	Considerações Finais	22
3	Trabalhos Relacionados	23
3.1	Centros de Dados Verdes	23
3.2	Otimizações em Servidores	24
3.3	Otimizações no Sistema de Refrigeração	26
3.4	Otimizações na Infraestrutura de Rede	28
3.5	Softwares de Simulação para Ambientes de Centro de Dados	29
3.6	Considerações Finais	32
4	O Simulador SimDC3D	33
4.1	Arquitetura Genérica de Gestão de Energia para Centros de Dados Virtualizados	34
4.2	Concepção do SimDC3D	36
4.3	Arquitetura do SimDC3D	38
4.3.1	Arquivos de Entrada	38
4.3.1.1	Parâmetros de Simulação	39
4.3.1.2	Cargas de Trabalho	41
4.3.1.3	Topologias de Rede e Matriz de Tráfego	42
4.3.2	Fluxo de Execução e Modelagem de Simulação com o SimDC3D . . .	44
4.3.2.1	Centro de Dados	44
4.3.2.2	Banco de Servidores	48
4.3.2.3	Sistema de Refrigeração	49
4.3.2.4	Infraestrutura de Rede	51
4.4	Estatísticas de Desempenho Calculadas pelo SimDC3D	57
4.4.1	Acordos de Nível de Serviço	59
4.4.2	Métricas de Desempenho do Centro de Dados	62
4.4.3	<i>Rack Cooling Index (RCI)</i>	63
4.5	Considerações Finais	64
5	Otimizações em Centros de Dados	66
5.1	Algoritmos de Escalonamento de Máquinas Virtuais	66
5.1.1	Escalonador 3DMOBFD	67
5.1.2	Escalonador 3DMOBFD com Preditor	69
5.1.3	Escalonador 3DMOBFD com Banco de Servidores	70
5.1.4	Escalonador 3DMOBFD com Preditor e Banco de Servidores	71
5.1.5	Escalonador 2D	71
5.1.6	Escalonador 2D_PREDICTION	73
5.1.7	Escalonador 2D_POOL	73
5.1.8	Escalonador 2D_POOL_AND_PREDICTION	74
5.1.9	Escalonadores do SimWare	75

5.1.9.1	Escalonador BEST_PERFORMANCE	75
5.1.9.2	Escalonador LOW_TEMP_FIRST	76
5.1.9.3	Escalonador RANDOM	76
5.1.9.4	Escalonador MIN_HR	78
5.1.9.5	Escalonador CENTER_RACK_FIRST	78
5.2	Algoritmos de Realocação de Máquinas Virtuais	79
5.2.1	Algoritmo 3DMOBFD	81
5.2.2	Algoritmo 2D	82
5.2.3	Algoritmo HIGH_TEMPERATURE	82
5.2.4	Algoritmo FFD (<i>The First Fit Decreasing</i>)	83
5.2.5	Algoritmo PABFD (<i>Power Aware Best Fit Decreasing</i>)	84
5.3	Algoritmos de Controle de Temperatura do Sistema de Refrigeração	85
5.3.1	Algoritmo Padrão do SimWare	86
5.3.2	Algoritmo Agressivo do SimDC3D	86
5.4	Algoritmo de Otimização da Topologia da Rede	87
5.5	Considerações Finais	88
6	Resultados Experimentais no Simulador SimDC3D	90
6.1	Parâmetros do Centro de Dados	90
6.2	Parâmetros de Simulação	93
6.3	Indicadores de Desempenho do Centro de Dados	95
6.4	Estudos de Caso	97
6.4.1	Cenário 1: Avaliação do Impacto do Banco de Servidores no Consumo Energético e no Desempenho do Centro de Dados	98
6.4.2	Cenário 2: Avaliação do Controle de Temperatura nos Indicadores de Desempenho do Centro de Dados e na Temperatura dos Servidores	100
6.4.3	Cenário 3: Avaliação de Diferentes Cargas de Processamento nos Indicadores de Desempenho	101
6.4.4	Cenário 4: Avaliação de Otimizações de Topologias de Rede	103
6.4.5	Cenário 5: Avaliação de Preditor de Uso de CPU na Prevenção de Sobrecargas	108
6.4.6	Cenário 6: Avaliação das Estratégias de Predição do SimDC3D	110
6.4.6.1	Erro Médio e Desvio Padrão	111
6.4.6.2	Desempenho dos Preditores <i>RBF</i> e <i>Polynom</i>	111
6.4.7	Cenário 7: Centro de Dados Tradicional versus Centro de Dados Verde	115
6.5	Considerações Finais	118
7	Conclusão e Trabalhos Futuros	120
	Referências	123
A	Modelagem do Problema de Mapeamento de Máquina Virtual	130
A.1	Problema da Mochila	131

A.2	Avaliação do Mapeamento	133
A.3	Função Objetivo	134
B	Parâmetros do Simulador SimDC3D	138
C	Implementação do Algoritmo de Predição RBF	153
D	Implementação do Algoritmo de Predição POLYNOM	161
E	Resultados de Simulação Fornecidos pelo SimDC3D	166

CAPÍTULO 1

INTRODUÇÃO

Centros de dados são construções especializadas para hospedar sistemas computacionais e sistemas auxiliares de comunicação, refrigeração, armazenamento e distribuição de energia. Tradicionalmente, as empresas construíam e mantinham seus próprios centros de dados. Entretanto, a complexidade de gerenciamento, os custos operacionais e a ineficiência de se manter uma infraestrutura dedicada em cada empresa fizeram com que essa tendência mudasse. Recentemente, as empresas cada vez mais estão migrando serviços mantidos em centros de dados tradicionais para serviços na nuvem, como Infraestrutura como Serviço (*do inglês Infrastructure as a Service*) (IaaS), Plataforma como serviço (*do inglês Platform as a Service*) (PaaS) e Software como Serviço (*do inglês Software as a Service*) (SaaS) (GARTNER, 2018). Segundo o relatório da Gartner (GARTNER, 2018), 80% das empresas migrarão seus centros de dados tradicionais para nuvens computacionais até 2025 contra os 10% atuais. Com esse crescimento previsto, provedores de nuvem precisam buscar maneiras de reduzir seus custos operacionais e ao mesmo tempo melhorar a qualidade dos serviços oferecidos. O consumo de energia de um centro de dados é um dos maiores fatores das despesas operacionais.

1.1 Consumo Energético em Centros de Dados

Os centros de dados estão entre os maiores consumidores de energia elétrica da área de computação. O consumo de um único centro de dados — como o da Google, por exemplo — chega a ser superior ao de uma cidade com 100 mil habitantes (REPORTER, 2014). Segundo Singh *et al.* (SINGH *et al.*, 2017), os centros de dados nos Estados Unidos vão consumir aproximadamente 73 bilhões de watts por hora até 2020. A tendência é que esse consumo de energia aumente ainda mais, pois a demanda por serviços computacionais tem aumentado significativamente, principalmente pela popularização da computação em nuvem.

Além de contribuir significativamente para os custos operacionais, o consumo energético de um centro de dados contribui para os custos ambientais, dependendo principalmente da matriz

energética de cada país. Por exemplo, nos Estados Unidos onde há uma grande concentração de centro de dados, 79% da produção de energia são provenientes da queima de combustíveis fósseis, que são extremamente poluentes. Já as despesas operacionais dependem diretamente da eficiência energética do centro de dados. A conta de energia de um centro de dados típico representa aproximadamente 45% do total das despesas operacionais e, em certos casos, pode chegar a até 75% (SAMS, 2010). Se o consumo de energia continuar aumentando, ele pode facilmente ultrapassar o custo de *hardware* por uma grande margem (XU; WANG, 2014).

O consumo energético de um centro de dados é dominado por três dimensões: servidores, sistema de refrigeração e infraestrutura de rede. Essas três dimensões em conjunto com o sistema de armazenamento são responsáveis por 90% do consumo total de eletricidade de um centro de dados (PATEL; MAKWANA, 2016). Portanto, desenvolver algoritmos para otimizar o consumo de energia nessas três dimensões é fundamental para se reduzir os custos operacionais e ambientais dos centros de dados.

1.2 Redução do Consumo Energético em Centros de Dados

Os centros de dados não são famosos apenas por consumirem grandes quantidades de energia, mas também por serem ineficientes. A principal razão para ineficiência energética de um centro de dados é a baixa utilização média de seus recursos. Um centro de dados precisa ser provisionado com infraestruturas de computação e comunicação capazes de lidar com a natureza dinâmica da demanda de recursos. Para garantir serviços ininterruptos e com qualidade aceitável, operadores de centro de dados normalmente constroem infraestruturas tolerantes a falhas e super dimensionadas. Porém, durante períodos de baixa demanda, elas ficam subutilizadas ou completamente ociosas, fazendo com que os custos do super dimensionamento sejam significativos e impactem os balanços financeiros dos provedores.

Os servidores têm se tornado mais eficientes no uso da energia elétrica. Técnicas como *Dynamic Voltage and Frequency Scaling* (DVFS) permitem que o sistema operacional diminua a frequência e a voltagem do processador de um servidor quando ele está com utilização baixa para economizar energia. Entretanto, ainda há oportunidades para economia de energia em como eles são utilizados e interagem com as outras duas dimensões de maior consumo. Os trabalhos existentes na literatura propõem algoritmos para se economizar energia considerando uma ou duas dimensões em conjunto, como por exemplo:

1. *Hardware*: Beloglazov *et al.* (BELOGLAZOV; BUYYA, 2012) e Ricciardi *et al.* (RICCIARDI *et al.*, 2011) propõem algoritmos de migração de máquinas virtuais em que consideram apenas as informações dos servidores para a eliminação rápida de pontos de sobrecarga ou ociosidade nos servidores, permitindo que servidores sejam desligados ou colocados em hibernação para economizar energia.

2. Sistema de Refrigeração: Moore *et al.* (MOORE *et al.*, 2005) e Kant *et al.* (KANT; MURUGAN; DU, 2011) usam as temperaturas de entrada e saída de cada servidor para decidir em qual servidor uma determinada carga de trabalho deve ser alocada para se reduzir o consumo energético do sistema de refrigeração.
3. Infraestrutura de Rede: Heller *et al.* (HELLER *et al.*, 2010) e Wang *et al.* (WANG *et al.*, 2012) propõem a consolidação de fluxos de dados em um pequeno conjunto de enlaces e comutadores, que são suficientes para servir a demanda da largura de banda na maior parte do tempo, para poder desligar os equipamentos que não são necessários.
4. Servidores e Rede: Mend *et al.* (MENG; PAPPAS; ZHANG, 2010) e Shirayanagi *et al.* (SHIRAYANAGI; YAMADA; KONO, 2012) propõem esquemas de migração de máquinas virtuais em que consideram os recursos dos servidores e o tráfego entre as máquinas virtuais para realizar a migração.

Esses são apenas alguns exemplos de estratégias que podem ser utilizadas e combinadas para se otimizar a utilização de recursos e reduzir o consumo energético de um centro de dados. Vários outros trabalhos recentes também propõem algoritmos na mesma linha (BELOGLAZOV; BUYYA, 2012; RONG *et al.*, 2016; AKHTER; OTHMAN, 2016). Porém, algoritmos que otimizam apenas uma ou duas das dimensões de maior consumo podem esconder aumentos significativos no consumo das demais dimensões. Além disso, otimizações de energia podem aumentar os tempos de resposta e violar acordos de nível de serviço.

1.3 Simulações de Centros de Dados

Devido à dificuldade de se avaliar propostas de economia de energia em centros de dados reais, os trabalhos existentes na literatura, na grande maioria, reportam resultados de simulações. Entretanto, os simuladores utilizados são limitados e não cobrem adequadamente todas as dimensões de maior consumo, além de ignorar fontes importantes de consumo de energia e não simular subsistemas ou interações físicas dentro do centro de dados.

A Tabela 1.1 mostra as características dos principais simuladores disponíveis na literatura: SimWare (YEO; LEE, 2012), CloudSim (CALHEIROS *et al.*, 2011), NetworkCloudSim (GARG; BUYYA, 2011), DCSim (TIGHE *et al.*, 2012) e GreenCloud (KLIAZOVICH; BOUVRY; KHAN, 2012). Os dados da tabela permitem concluir que:

1. Nenhum simulador atual simula todas as dimensões de maior consumo (servidores, sistema de refrigeração e infraestrutura de rede) em um único ambiente de simulação com um nível de realismo aceitável.
2. O simulador SimWare simula o sistema de refrigeração e os servidores. Entretanto, ele não simula a infraestrutura de rede, migrações de máquinas virtuais, parâmetros de

Qualidade de Serviço (*do inglês Quality of Service*) (QoS) e o uso de preditores para escalonamento e realocação de cargas de trabalho.

3. O simulador CloudSim não simula o sistema de refrigeração e não considera o consumo energético das ventoinhas dos servidores, pois ele também não simula o fluxo de ar nos servidores. Além disso, ele não simula a infraestrutura de rede e parâmetros de QoS. Ele ainda apresenta limitações no tipo de carga de trabalho que pode simular realisticamente.
4. O Network CloudSim é uma extensão do CloudSim para simular a infraestrutura de rede. Porém, ele não implementa técnicas fundamentais para economia de energia na rede, como *Adaptive Link Rate* (ALR) e otimizações de topologia.
5. O DCSim simula apenas os servidores do centro de dados e é capaz de descrever apenas o consumo energético da CPU. Ele utiliza cargas de trabalho sintéticas e não apresenta um modelo detalhado de energia.
6. O simulador GreenCloud não suporta máquinas virtuais. Ele apenas simula uma carga de processamento em cada servidor e, conseqüentemente, é bastante limitado para estudos de eficiência energética, além de não simular parâmetros de QoS.

O que se pode concluir da discussão acima é que os simuladores atuais deixam muito a desejar com relação às suas capacidades de avaliação de propostas de economia de energia e como elas interferem nos consumos de energia e desempenho dos diferentes subsistemas de um centro de dados. Portanto, para se avançar em estudos que combinem eficiência energética e qualidade de serviço há necessidade de novas ferramentas de simulação. Um dos objetivos deste trabalho é cobrir essa lacuna e desenvolver um novo simulador com todas as características listadas na Tabela 1.1 e que será descrito no Capítulo 4.

1.4 Qualidade de Serviço

A virtualização é uma tecnologia que ajuda a reduzir o consumo energético de um centro de dados ao permitir que várias Máquinas Virtuais (*do inglês Virtual Machine*) (VM)s sejam consolidadas em um único servidor. Com isso, é possível reduzir a quantidade de servidores ligados e, conseqüentemente, economizar energia. Provedores de serviços em nuvem usam a virtualização para concentrar VMs de diferentes clientes em um único servidor e evitar desperdícios relacionados a baixa utilização do *hardware*. Porém, a concentração de uma grande quantidade de máquinas virtuais em um servidor físico pode degradar significativamente os serviços oferecidos e deixar os clientes insatisfeitos.

A relação comercial entre cliente e provedor de serviços em nuvem é formalizada por um documento de Acordo de Nível de Serviço (*do inglês Service-Level Agreement*) (SLA). Esse documento descreve as garantias e a maneira como os indicadores de desempenho serão

Tabela 1.1 – Características de simuladores para centro de dados.

	SimWare	CloudSim	Network CloudSim	DCSim	GreenCloud
Código Aberto	✓	✓	✓	✓	✓
Tempo Simulação	Segundos	Segundos	Segundos	Minuto	Minuto
Modelo Detalhado de Energia	✓	✓	✓	✗	✓
Modelo de Carga de Trabalho	Traços Reais	Cargas Sintéticas	Cargas Sintéticas	Cargas Sintéticas	Cargas Sintéticas
Recursos Simulados	CPU, Ventoinhas, Sistema de Refrigeração	CPU, Memória, Disco	CPU, Memória, Disco, Comutadores	CPU	CPU, Comutadores
Ventilação Interna	✓	✓ Limitado	✓ Limitado	✗	✗
Refrigeração	(HRM)	✗	✗	✗	✗
Rede	✗	✗	✓	✗	✓
Modelo de Comunicação de Rede	✗	✓ Limitado	Nível Pacote	✗	Nível Pacote
Modelo de Energia da CPU	Modelo Linear	Modelo Linear	Modelo Linear	Modelo Linear	Baseado Frequência
Suporta VMs	✓	✓	✓	✓	✗
Escalonamento de VMs	✓	✓	✓	✓	✗
Migração de VMs	✗	✓	✓	✓	✗
Políticas de Seleção de VMs	✗	✓	✓	✗	✗
Políticas de Controle Temperatura	✓ Limitado	✗	✗	✗	✗
Simula Diferentes Topologias	✗	✗	✓	✗	✓
Políticas de Otimização Topologia	✗	✗	✗	✗	✗
Modo de Operação dos Servidores	Ligado	Ligado, Hibernando	Ligado, Hibernando	Ligado, Hibernando	Ligado, Desligado
QoS	✗	✓	✓	✓	✗
Preditores	✗	✓	✓	✗	✗

mantidos. O documento também descreve as penalidades para o provedor caso haja alguma violação de SLA. Portanto, para garantir a utilização eficiente dos recursos de seus centros de dados, os provedores de serviço precisam balancear adequadamente o desempenho oferecido e o consumo energético, pois uma consolidação agressiva de VMs pode levar à perda de desempenho e, conseqüentemente, à quebra nos acordos de nível de serviço. Nesse caso, a economia de energia obtida com a consolidação não compensaria as multas previstas nos SLAs, além de gerar insatisfações nos clientes e, possivelmente, rompimentos de contratos comerciais. A construção de algoritmos eficientes de alocação de máquinas virtuais em servidores para se economizar energia e ao mesmo tempo não violar acordos de nível de serviço é um problema de pesquisa desafiador que será abordado neste trabalho no Capítulo 5.

1.5 Contribuições

Este trabalho propõe inicialmente uma arquitetura de gerenciamento e simulação para estudos de eficiência energética de centros de dados virtualizados que inclui os principais subsistemas de forma integrada, de modo que seja possível avaliar algoritmos de escalonamento de máquinas virtuais sob diferentes cargas de trabalho e seus impactos no consumo energético e no desempenho dos demais subsistemas. Além disso, a arquitetura incorpora módulos de monitoramento e otimização do sistema de refrigeração e da infraestrutura de rede.

Com base na arquitetura proposta, este trabalho desenvolve o simulador SimDC3D que simula de forma realista os subsistemas de maior consumo de energia elétrica, incluindo o consumo de energia das ventoinhas dos servidores, tempo de propagação do ar frio e recirculação de calor dentro do centro de dados, tempo de migração de máquinas virtuais e consumo de energia de interfaces de rede. Além disso, o SimDC3D permite analisar os impactos de diferentes algoritmos de otimização em métricas de desempenho e acordos de nível de serviço, consistindo, portanto, em uma ferramenta essencial para a construção e avaliação de algoritmos que minimizem o consumo energético sem violações de SLA.

O SimDC3D simula um controlador central cuja função principal é monitorar periodicamente os servidores, o sistema de refrigeração e a infraestrutura de rede a procura de oportunidades de otimização. Além disso, ele identifica problemas causados por enlaces congestionados ou servidores sobrecarregados, subutilizados, superaquecidos ou quando um preditor indicar que haverá sobrecarga de utilização de CPU e realoca máquinas virtuais se for necessário. Se algum dos casos ocorrer, o controlador seleciona uma ou mais máquinas virtuais para migrar para outro servidor.

Ao término da simulação, o SimDC3D fornece o consumo energético detalhado de cada dispositivo, o total de migrações de máquinas virtuais, as temperaturas médias do centro de dados e dos chassis e estatísticas das filas de escalonamento, do tráfego na rede, dos preditores e das violações de nível de serviço. Os níveis de serviço que são avaliados no SimDC3D são

sobrecarga da CPU (período de tempo que um servidor opera com 100% de utilização de CPU) e o tempo que uma máquina virtual fica indisponível durante o processo de migração (*downtime*).

O SimDC3D é um simulador de código aberto com mais de 15 mil linhas de código em C++. Ele possui uma arquitetura extensível que permite a incorporação de novos algoritmos de otimização por meio de especializações de classes abstratas predefinidas para os principais subsistemas.

Este trabalho também propõe dois novos algoritmos de escalonamento e consolidação de máquinas virtuais, cada um com quatro variações, um algoritmo para controle do sistema de refrigeração e um algoritmo para otimização da topologia da rede do centro de dados. Além disso, este trabalho faz uma avaliação extensiva com centenas de simulações dos algoritmos propostos usando diferentes cargas de trabalho e topologias de rede e os compara com outras propostas existentes na literatura. Os experimentos realizados mostram resultados de consumo de energia nas três dimensões de maior consumo e como os algoritmos de economia de energia afetam parâmetros de desempenho, como acordos de nível de serviço. Este é o primeiro trabalho que avalia extensivamente algoritmos de economia de energia considerando as três dimensões de maior consumo e seus efeitos em parâmetros de desempenho.

1.6 Organização da Tese

O restante desta tese está organizado da seguinte maneira. O Capítulo 2 apresenta conceitos básicos de centros de dados e de consumo energético que são essenciais para o entendimento deste trabalho. No Capítulo 3, são apresentados os principais trabalhos existentes na literatura que estão relacionados com esta tese. Além disso, esse capítulo descreve algoritmos de otimização para os vários subsistemas de um centro de dados e os principais simuladores existentes para estudos de economia de energia em centros de dados. O Capítulo 4 detalha o simulador SimDC3D e a arquitetura genérica para economia de energia em um centro de dados proposta nesta tese. No Capítulo 5, são descritos os algoritmos de otimização propostos nesta tese. A avaliação experimental dos algoritmos de otimização propostos e comparações com outros trabalhos existentes estão descritas no Capítulo 6. O Capítulo 7 apresenta as conclusões desta tese e discute possíveis trabalhos futuros.

CAPÍTULO 2

CONCEITOS BÁSICOS

Segundo o relatório divulgado pela Gartner (GARTNER, 2018), oitenta por cento das empresas migrarão seus centros de dados tradicionais para nuvens computacionais até 2025 contra os 10% atuais. Em um futuro próximo, os centros de dados estarão dedicados a serviços muito específicos e que não podem ser executados em outro lugar ou quando executados localmente sejam mais eficientes economicamente em comparação com a nuvem. Segundo o mesmo relatório, com o aumento da oferta de provedores de computação em nuvem, serviços de borda (*edge computing*) e SaaS, não há justificativas para se manter as atuais infraestruturas de centros de dados.

Com o aumento da demanda de IaaS, PaaS e SaaS, os operadores de computação em nuvem necessitam cada vez mais de uma infraestrutura para absorver as necessidades das empresas que estão passando pelo processo de migração dos seus centros de dados. Com isso, eles estão sendo instigados a construir centros de dados que usem recursos—*e.g.*, energia, água e materiais—de forma mais eficiente e que tenham menos impacto sobre as pessoas e o meio ambiente. Essa preocupação crescente com o meio ambiente e com a sustentabilidade é a origem da expressão *verde*.

O restante deste capítulo está organizado da seguinte forma. A Seção 2.1 apresenta os principais conceitos de centro de dados verde. As Seções 2.2, 2.3 e 2.4 apresentam estratégias para economia de energia nas dimensões servidor, sistema de refrigeração e infraestrutura de rede, respectivamente. Na Seção 2.5 são discutidos os principais conceitos de séries temporais. A Seção 2.6 apresenta as estatísticas que são utilizadas para avaliar o desempenho dos algoritmos de predição e, por último, a Seção 2.7 apresenta as conclusões finais deste capítulo.

2.1 Centro de Dados Verde

O termo verde pode ser usado para designar muitas coisas, como uma instalação que foi construída usando menos recursos naturais, um espaço de hospedagem que produz menos poluentes ou um ambiente que consome menos energia. Estendendo esse conceito, pode-se dizer que um centro de dados verde é um ambiente de computação que usa recursos de maneira mais eficiente e tem menos impacto sobre as pessoas e o meio ambiente. Porém, a principal questão em um de centro de dados verde é o consumo eficiente de energia elétrica.

Centros de dados consomem uma quantidade incrível de energia. A Comissão Europeia estima que eles consumirão 104 bilhões de kWh até 2020 (ALGER, 2009). Esse crescimento dramático no consumo pode ser atribuído em parte à proliferação geral de tecnologias e ao uso da Internet em toda a sociedade. Mais e-mails, mensagens instantâneas, pesquisas na web, transações financeiras on-line e downloads de vídeo ocorrem hoje do que ocorriam há alguns anos. Conseqüentemente, mais servidores, equipamentos de rede e dispositivos de armazenamento são necessários para lidar com essas demandas, resultando em maior consumo de energia.

Os servidores de hoje são fisicamente menores do que os de alguns anos atrás, permitindo que uma maior quantidade seja colocada em cada rack de um centro de dados. Mesmo que um novo servidor consuma menos energia do que seus predecessores, o fato de se instalar mais equipamentos no mesmo espaço físico implica em maior consumo energético.

Um aumento na quantidade de servidores também aumenta os pontos de calor e sobrecarrega o sistema de refrigeração, tornando mais difícil manter os centros de dados antigos alinhados com as demandas atuais. Considerando que os custos de energia estão extremamente altos e que o sistema de refrigeração representa 40% do consumo energético total de um centro de dados (OXLEY et al., 2018), manter um sistema antigo pode ser muito desvantajoso economicamente para uma empresa.

Atualmente, os equipamentos de TI são cada vez mais eficientes em termos de energia e ecologicamente corretos. Porém, somente utilizar equipamentos mais eficientes energeticamente não é suficiente. Para atingir uma maior eficiência energética são necessárias medidas mais agressivas. Diversas tecnologias estão disponíveis para reduzir o consumo energético em cada uma das dimensões de maior consumo (servidores, sistema de refrigeração e infraestrutura de rede) do centro de dados. Em servidores, algumas tecnologias visam os componentes de *hardware* (processadores, fontes, memória, entre outros.) e outras propõem escalonadores dinâmicos para consolidação de máquinas virtuais. No sistema de refrigeração, há métodos de escalonamento das cargas de trabalho de acordo com as temperaturas de entrada de cada servidor e controle dinâmico de temperatura no sistema de refrigeração. Por último, na infraestrutura de rede, a reorganização dinâmica da topologia de acordo com o tráfego é uma das principais ideias.

Nas próximas seções, são apresentados os principais conceitos utilizados em estratégias para economia de energia para cada uma das dimensões de maior consumo energético de centros de dados.

2.2 Estratégias de Economia de Energia para Servidores

Políticas ineficientes de gerenciamento exploram mal os recursos de um centro de dados. Normalmente, os centros de dados são provisionados em excesso para garantir alta disponibilidade de serviços e QoS para as aplicações. Em média, 30% dos servidores em nuvem utilizam de 10 a 15% de sua capacidade na maior parte do tempo (UDDIN et al., 2013). A baixa utilização dos recursos computacionais de um servidor resulta em altos custos operacionais e energéticos.

O principal desafio para economia de energia em centro de dados é conciliar economia de energia com desempenho. A fim de economizar energia, os centros de dados teriam menor disponibilidade de servidores ou recursos de rede, podendo em certos momentos comprometer a capacidade de atendimento aos usuários.

Para compreender esses desafios, Chang Ge *et al.* (GE; SUN; WANG, 2013) propõem uma divisão dos modelos de economia de energia em quatro categorias: computação com energia proporcional (*energy proportional computing*), provisionamento dinâmico (*dynamic provisioning*), virtualização (*virtualization*), ajuste e limitação de potência (*Power capping and shifting*). As próximas quatro seções discutem essas categorias.

2.2.1 Computação com Energia Proporcional

A expressão “computação com energia proporcional” foi introduzida por Barroso *et al.* (BARROSO; HOLZLE, 2007) ao observarem que servidores raramente estão ociosos ou em carga máxima. A utilização da CPU nos servidores fica na maior parte do tempo em um intervalo que varia entre 10 e 50% (BARROSO; HOLZLE, 2007). Por outro lado, um servidor ocioso consome entre 49% e 78% de energia em relação ao consumo quando ele está operando na sua potência máxima.

Conforme Ge *et al.* (GE; SUN; WANG, 2013), o consumo de energia em um servidor deveria ser proporcional ao seu volume de trabalho e nenhuma energia deveria ser consumida por um servidor ocioso. Embora esse objetivo seja teórico, esse princípio tem motivado pesquisas sobre o uso proporcional de energia em servidores e seus componentes. Uma conquista importante é a tecnologia DVFS proposta por Wu *et al.* (WU et al., 2005), que ajusta a voltagem e a frequência usadas pela CPU de acordo com a carga de processamento. Processadores operando com voltagens e frequências mais baixas consomem menos energia. Além disso, memórias como a Memória de Acesso Aleatório Magnetoresistive (*do inglês Magneto-resistive Random Access Memory*) (MRAM) usam uma nova célula de memória e

reduzem pela metade o consumo de energia durante o processo de escrita de dados. Interfaces de rede com a tecnologia 802.3az (Green Ethernet) também auxiliam na redução do consumo de energia na infraestrutura de rede.

O lucro em potencial que a proporcionalidade de energia pode introduzir nos centros de dados e seus benefícios ainda não são possíveis, pois os fabricantes de *hardware* ainda não disponibilizam todas as técnicas existentes em seus dispositivos. Atualmente, a técnica DVFS em processadores é a abordagem mais utilizada na proporcionalidade de energia para os componentes de *hardware* dos servidores. Na prática, essa técnica está longe de ser suficiente para a implantação de centros de dados energeticamente eficientes.

2.2.2 Provisionamento Dinâmico

O objetivo do provisionamento dinâmico é adaptar a quantidade de servidores ligados com base na carga atual ou futura do centro de dados. Fora dos horários de pico, alguns servidores devem receber as cargas e os demais devem ser desligados ou colocados em modo de hibernação. Conforme Ge *et al.* (GE; SUN; WANG, 2013), redistribuir as cargas de trabalho no menor número de servidores deve considerar três desafios importantes: não sobrecarregar os servidores ativos, não degradar a qualidade do serviço oferecido e não reduzir a tolerância a falhas significativamente. Portanto, estratégias de provisionamento dinâmico devem balancear economia de energia com qualidade de serviço e tolerância a falhas.

2.2.3 Virtualização

A virtualização permite que uma única máquina física execute múltiplas virtuais, possibilitando a utilização mais eficiente de recursos. No entanto, a hospedagem conjunta de várias VMs pode degradar o desempenho das aplicações devido à contenção de recursos (HU *et al.*, 2013; NATHAN; KULKARNI; BELLUR, 2013). Para melhorar o desempenho das aplicações, máquinas virtuais podem ser migradas para servidores com mais recursos disponíveis. Porém, o próprio processo de migração pode afetar o desempenho das aplicações durante o tempo que elas estão sendo migradas (Xu *et al.*, 2014; LUČANIN *et al.*, 2014). Portanto, uma estratégia de consolidação deve considerar o tempo de migração no processo de consolidação para não afetar significativamente o desempenho das aplicações.

Alguns tipos de migração de máquinas virtuais, como migrações baseadas em pré-cópia da memória ou de *trace/checkpoint* e pós-cópia procuram minimizar o impactado causado pela migração, obtendo um baixo *downtime*¹ para minimizar violações de acordos de nível de serviço (SLA) entre os clientes e provedores.

¹ É o tempo durante o qual a execução da VM que está sendo migrada é interrompida. No mínimo, esse intervalo de tempo inclui a transferência dos estados do processador, dos dispositivos de rede e do disco.

2.2.4 Ajuste e Limitação de Potência

A técnica de Ajuste e Limitação de Potência (*do inglês Power Capping and Shifting*) (PCS) é empregada em centros de dados com restrições orçamentárias de energia (GE; SUN; WANG, 2013). O PCS foi desenvolvido para permitir que o desempenho de um servidor seja limitado em relação ao consumo de energia global do centro de dados. Neste caso, o desempenho dos processadores em um servidor fica limitado à quantidade de energia que lhe é atribuída. O PCS define o limite superior da quantidade de energia que é alocada a cada *cluster* de servidores no centro de dados, garantindo que eles não violem as restrições orçamentárias de energia.

O PCS ajusta dinamicamente a quantidade de energia que é atribuída a cada servidor do *cluster*, reduzindo, conseqüentemente, sua capacidade de processamento. O desempenho geral dos servidores pode ser otimizado com um conjunto ideal de alocação de potência. Wang *et al.* (WANG; CHEN, 2008) mostram que o PCS pode atingir economias significativas de energia.

2.3 Estratégias de Economia de Energia para o Sistema de Refrigeração

Os centros de dados recentes estão sendo construídos com sistemas de refrigeração mais eficientes em relação aos centros de dados mais antigos. Entretanto, o custo de refrigeração representa pelo menos 40% do custo total de energia de um centro de dados (OXLEY *et al.*, 2018). Sendo assim, um dos principais desafios para se diminuir os custos operacionais de um centro de dados é minimizar a necessidade de refrigeração e, em conseqüência, melhorar a eficiência energética do centro de dados como um todo.

Em centros de dados, os servidores emitem calor ao executarem processamento. Porém, eles não podem funcionar com temperaturas acima da máxima permitida pelo fabricante do dispositivo. O sistema de refrigeração deve, portanto, fornecer ar frio nas entradas de ar dos servidores a uma temperatura que mantenha os servidores sempre abaixo dessa temperatura máxima. O grande problema é que quanto menor a temperatura do ar fornecido, maior o consumo de energia do sistema de refrigeração. Um desafio neste caso é desenvolver algoritmos que aloquem cargas nos servidores de modo a não elevar consideravelmente suas temperaturas para evitar aumento do consumo de energia do sistema de refrigeração.

O ar frio injetado na frente de um servidor é liberado como ar quente em sua saída. Dependendo de como o centro de dados é projetado, o ar quente pode recircular pelo centro de dados e voltar para as entradas de ar frio, fazendo com que a temperatura interna dos servidores aumente. Dessa forma, a recirculação de calor pode causar o aparecimento de pontos quentes no centro de dados, provocando o aumento no fornecimento de ar frio, como pode ser visto na Figura 2.1.

A diminuição da temperatura de saída de um sistema de refrigeração o obriga a operar com um coeficiente de desempenho menor, ou seja, a relação do calor removido sobre a energia

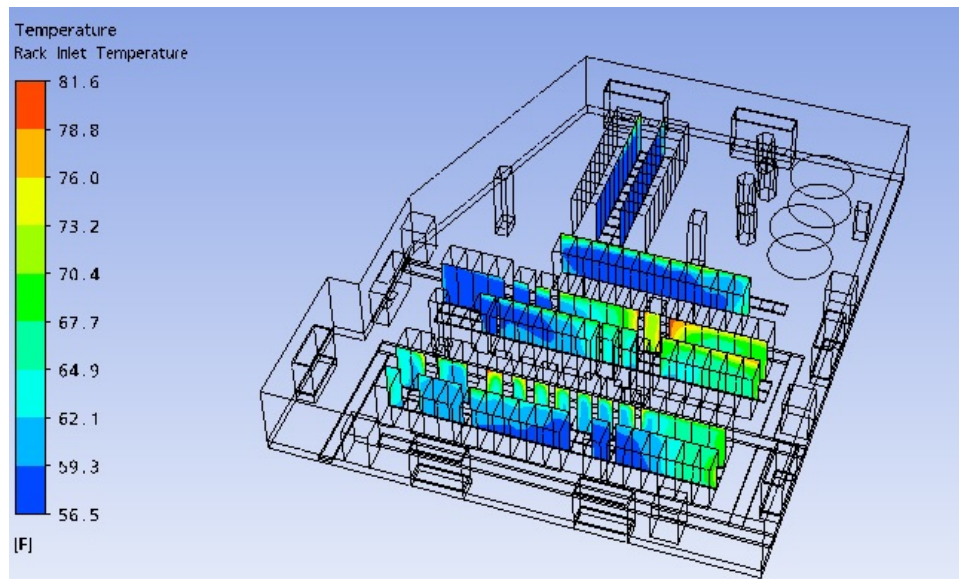


Figura 2.1 – Exemplo de mapa térmico de um centro de dados gerado pelo simulador CoolSim (COOLSIM, 2019).

necessária para isso, o que aumenta consideravelmente o custo de resfriamento. Sendo assim, a recirculação de calor desempenha um papel importante na eficiência energética do centro de dados.

A modelagem da recirculação de calor dentro de um centro de dados é normalmente feita com modelos de Dinâmica de Fluídos Computacional (*do inglês Computational Fluid Dynamics*) (CFD) ou mapas térmicos. Esses modelos levam em consideração a utilização de CPU dos servidores e a organização (*layout*) do centro de dados para prever a distribuição da temperatura nos diferentes pontos do centro de dados. No entanto, modelos de CFD e mapas térmicos são computacionalmente caros para produzir e, portanto, não são adequados para decisões de alocação de tarefas em tempo real. Para superar essa limitação computacional, simulações normalmente empregam uma Matriz de Recirculação de Calor (*do inglês Heat Recirculation Matrix*) (HRM) que captura com boa precisão a recirculação de calor dentro de um centro de dados e pode ser utilizada para decisões em tempo real.

Uma proposta para solucionar o problema de gestão de ar condicionado é a alocação de cargas de trabalho levando em consideração o estado térmico do centro de dados. A ideia é prever os efeitos térmicos de um posicionamento de uma ou mais tarefas e alocar as cargas de trabalho de modo a se obter melhores resultados térmicos (MOORE et al., 2005). A alocação adequada das cargas de trabalho melhora o coeficiente de desempenho do sistema de refrigeração e reduz o consumo energético e, conseqüentemente, os custos operacionais.

2.4 Estratégias de Economia de Energia para a Infraestrutura de Rede

Normalmente, a infraestrutura de rede de um centro de dados é projetada para suportar cargas de pico e oferecer tolerância a falhas. Entretanto, ela fica subutilizada em condições normais de operação, gerando, portanto, oportunidades para economia de energia. Nos últimos anos, esforços para reduzir os gastos desnecessários na infraestrutura de rede resultaram no conceito de “redes verdes”.

Bianzino *et al.* (BIANZINO et al., 2010) classificam as abordagens para economia de energia em redes em quatro grupos:

1. **Consolidação de recursos:** redimensiona os recursos para reduzir o consumo global da rede quando os dispositivos estão sendo subutilizados. O tráfego em uma rede, normalmente, tem um comportamento diário e semanal bem conhecido, gerando oportunidades de adaptação do nível de provisionamento de acordo com as condições de uso sem afetar o desempenho das aplicações.
2. **Virtualização:** aloca diferentes serviços, ou redes virtuais, em um mesmo dispositivo. Essa alocação resulta em um menor consumo de energia desde que o único dispositivo sob alta carga de processamento consuma menos energia que vários dispositivos com carga leve. Esse geralmente é o caso, pois o consumo fixo de um dispositivo de rede é muito maior que o consumo que varia com o tráfego de rede, como detalhado mais adiante.
3. **Conexão seletiva:** permite o desligamento de recursos ociosos na rede por algum tempo e de forma transparente para os demais dispositivos do centro de dados. As tarefas de um dispositivo desligado são assumidas por outros nós, como *proxies*, que assumem sua identidade momentaneamente de modo que nenhuma alteração fundamental seja necessária nos protocolos de rede.
4. **Computação proporcional:** altera as velocidades das interfaces de rede, e consequentemente dos enlaces, de acordo com seus níveis de utilização. Uma interface de rede operando em velocidade menor consome menos energia do que quando operando em velocidades maiores.

Mahadevan *et al.* (MAHADEVAN; BANERJEE; SHARMA, 2010) propõem duas técnicas para diminuir o consumo energético em um infraestrutura de rede. O objetivo da primeira técnica é desativar todas as portas dos comutadores que não estão sendo utilizadas. A segunda propõe uma adaptação da taxa de transmissão com base na taxa de utilização de cada porta. Baseado em medições, é possível determinar portas com baixa utilização que não apresentam uma grande variabilidade ao longo do tempo. Essa técnica só é viável em portas operando em 1 Gbps ou 10 Gbps, que podem ser configuradas para operar em 10 ou 100 Mbps.

Um comutador de rede é geralmente composto por chassi, matriz de comutação (*switching fabric*), *line-cards* e portas. O chassi de um comutador e seu sistema de refrigeração (ventiladores) geralmente consomem uma quantidade fixa de energia. A matriz de comutação é responsável em estabelecer a conexão entre as portas, enquanto os *buffers* dos *line-cards* mantêm todos os pacotes de entrada e de saída. As portas consomem uma quantidade de energia que varia de acordo com a velocidade de transmissão. De acordo com medições em (MAHADEVAN et al., 2009b), o consumo de energia em um comutador de acesso com 48 portas varia de 76 a 150 Watts. Aproximadamente 40 Watts podem ser adicionados, se o comutador passar a operar em sua capacidade máxima. Estas medidas indicam que cada porta de um comutador consome de um a dois Watts. Porém, o chassi do comutador, a matriz de comutação e os *line-cards* consomem a maior parte da energia. Portanto, economias de energia mais significativas são alcançadas principalmente quando os comutadores são desligados por meio de métodos de conexão seletiva que otimizam a topologia de acordo com a demanda de tráfego.

Além dessas técnicas, a topologia da rede influi decisivamente nas estratégias que podem ser utilizadas para se reduzir o consumo energético. A próxima seção apresenta as principais topologias de rede para centro de dados propostas nos últimos anos.

2.4.1 Topologias de Rede para Centro de Dados

Al-Fares et al. (AL-FARES; LOUKISSAS; VAHDAT, 2008), Greenberg et al. (GREENBERG et al., 2009), Abu-Libdeh et al. (ABU-LIBDEH et al., 2010) e Guo et al. (GUO et al., 2008; GUO et al., 2009) propõem novas topologias de rede para centro de dados que visam reduzir custos, melhorar a largura de banda de bisseção, aumentar a tolerância a falhas e reduzir o consumo energético. Dentre essas propostas, (AL-FARES; LOUKISSAS; VAHDAT, 2008) e (GREENBERG et al., 2009) usam apenas comutadores para encaminhar o tráfego entre servidores enquanto as demais exigem que os servidores participem do encaminhamento de pacotes.

Muitos centros de dados utilizam uma topologia de rede com três camadas. Nessa topologia, a camada inferior é conhecida como a camada de acesso, em que cada servidor se conecta a comutadores de acesso. Cada comutador de acesso se conecta a comutadores de agregação e cada comutador na camada de agregação se conecta a comutadores de núcleo. A Figura 2.2 mostra uma topologia três níveis enraizada em dois comutadores de núcleo.

Limitações na escalabilidade da topologia, endereçamento e inserção de servidores são problemas que levaram pesquisadores a proporem novas topologias de rede para centro de dados. Algumas topologias são mais eficientes energeticamente do que outras e suportam bisseção completa. As próximas seções apresentam as quatro topologias mais significativas recentemente propostas.

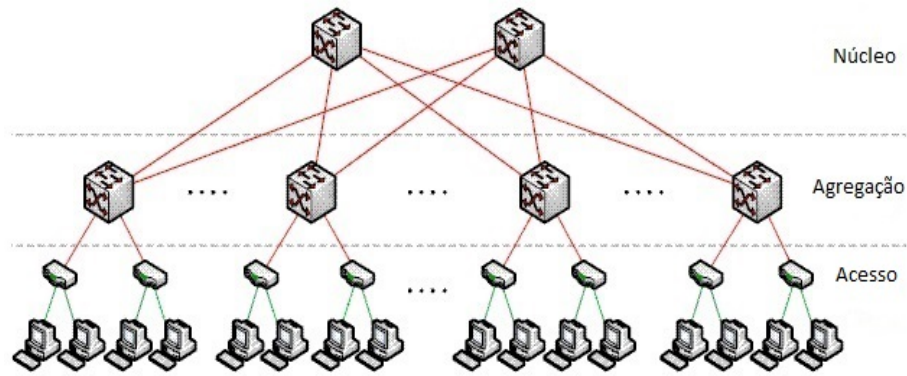


Figura 2.2 – Topologia de rede de 3 níveis.

2.4.1.1 VL2

Greenberg *et al.* (GREENBERG *et al.*, 2009) propõem a topologia VL2, que forma uma topologia Clos², para obter altos níveis de banda passante e tolerância a falhas nas camadas de núcleo e agregação. Nesta topologia, os comutadores de agregação estão conectados com os comutadores do núcleo formando um grafo bipartido completo. O tráfego originado nos comutadores de acesso é transmitido para as camadas de agregação e núcleo com o uso do algoritmo de balanceamento de carga de Valiant (VALIANT, 2008). No algoritmo de Valiant, os pacotes são encaminhados primeiro para um comutador de núcleo selecionado aleatoriamente e de lá encaminhados para o destino final.

Para tráfego imprevisível, a topologia VL2 balanceia a carga entre todos os enlaces disponíveis ao selecionar aleatoriamente um comutador do núcleo como um destino intermediário. A Figura 2.3 mostra uma topologia VL2.

2.4.1.2 Portland e FatTree

Mysore *et al.* (MYSORE *et al.*, 2009) propõem a topologia Portland que é uma adaptação da topologia FatTree proposta por Leiserson *et al.* (LEISERSON, 1985) e adaptada para centros de dados por (AL-FARES; LOUKISSAS; VAHDAT, 2008). Ela é construída em torno do conceito de *Pods*, que é uma coleção de comutadores de acesso e de agregação que forma um grafo bipartido completo. Cada *Pod* está conectado com todos os comutadores de núcleo, distribuindo uniformemente os *up-links* entre todos os comutadores de agregação do *Pod* e formando um segundo grafo bipartido completo entre comutadores de núcleo e os *Pods*. Portland assume que todos os comutadores são idênticos e possuem o mesmo número de portas.

O número de portas disponíveis por comutador é o único parâmetro que determina o número total de *Pods* e, em consequência, o número total de comutadores necessários e o número de servidores que podem se conectar à rede. Se k é o número de portas em cada comutador, então, existem k *Pods*, com $\frac{k}{2}$ comutadores de acesso e $\frac{k}{2}$ comutadores de agregação em cada *Pod*. Cada *Pod* está ligado com $\frac{k^2}{4}$ comutadores de núcleo e com $\frac{k^2}{4}$ servidores. No total, existem

² Topologia desenvolvida por Charles Clos do MIT (CLOS, 1953).

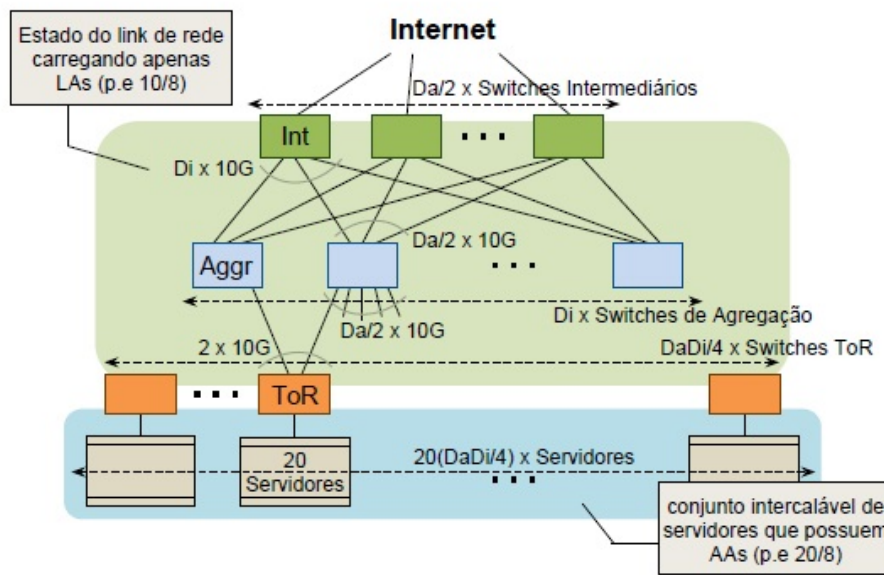


Figura 2.3 – Exemplo de rede Clos entre comutadores de agregação e núcleo fornecendo um *backbone* ricamente conectado e adequado para VLB. Redesenhado de (GREENBERG et al., 2009).

$\frac{5k^2}{4}$ comutadores que interconectam $\frac{k^3}{4}$ servidores. A Figura 2.4 mostra uma topologia Portland para $k = 4$.

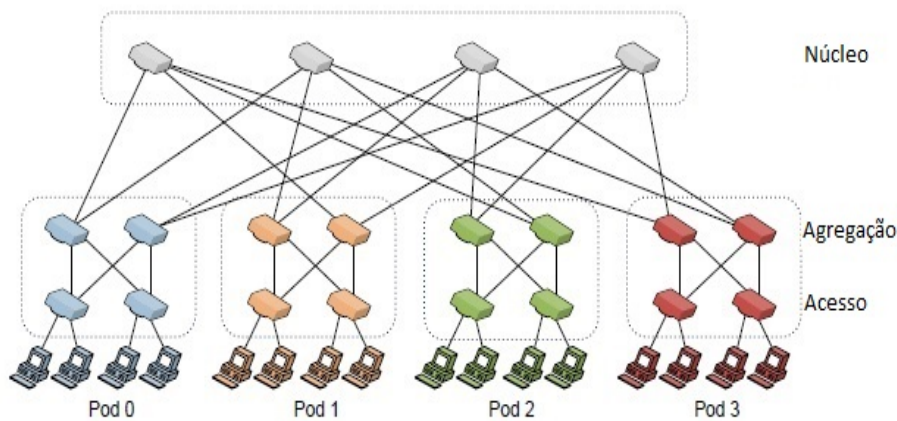


Figura 2.4 – Exemplo de um topologia Portland com $k = 4$. Redesenhado de (BOLLA et al., 2009).

2.4.1.3 BCube

BCube (GUO et al., 2009) é uma topologia de rede multinível para centro de dados tendo como característica marcante a participação dos servidores no encaminhamento de pacotes. Ela é uma estrutura definida recursivamente em que o nível 0, $BCube_0$, consiste de n servidores que se conectam com um conjunto de comutadores de n -portas. Um $BCube_k$ consiste de n $BCube_{k-1}$ conectados a n^k comutadores de n -portas. Servidores são rotulados com base em suas

localizações na estrutura BCube. Por exemplo, um rótulo 4.2.3 indica que o terceiro servidor do $BCube_0$ está dentro do segundo $BCube_1$, que por sua vez está dentro do quarto $BCube_2$. Os servidores usam esses rótulos para determinar suas conexões. Mais especificamente, servidores cujos rótulos diferem apenas no nível i conectam-se junto com um comutador no nível $BCube_i$. A arquitetura BCube forma um hipercubo.

Note que o número de servidores conectados, o número de comutadores e o número total de portas de cada comutador em uma topologia BCube são calculados em função de n e k , que determinam o número de níveis da topologia. A Figura 2.5 mostra uma topologia BCube para $k = 1$ e $n = 4$.

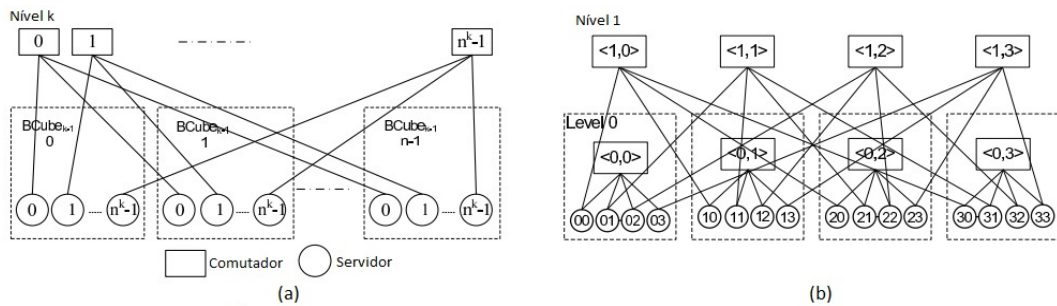


Figura 2.5 – (a) A $BCube_k$ é construída a partir de n $BCube_{k-1}$ e n^k comutadores com n -port. (b) Um $BCube_1$ com $n = 4$. Nesta rede $BCube_1$, cada servidor tem duas portas, extraído de (GUO et al., 2009).

2.4.1.4 DCell

DCell (GUO et al., 2008) é uma estrutura definida recursivamente, na qual uma DCell de alto nível é construída a partir de muitas DCells de baixo nível. Outra característica da topologia DCell é a utilização de servidores e minicomutadores (ao invés de comutadores de grande porte) no roteamento de pacotes. O componente principal da topologia é a rede $DCell_0$, constituída por um comutador ligado a n servidores. A Figura 2.6 mostra a construção de uma $DCell_1$ utilizando $n + 1$ redes $DCell_0$, em que $n = 4$. Neste caso, uma $DCell_0 x$ é conectada a outra $DCell_0 y$ por meio de enlace que conecta um servidor em x com qualquer outro servidor em y .

Em uma topologia DCell a comunicação entre os servidores pertencentes à mesma célula é realizada via comutador da célula. No exemplo da Figura 2.6, essa ligação é vista na comunicação entre os servidores 2 e 3 da $DCell_00$. A comunicação entre servidores em células diferentes pode ser feita de duas maneiras. De forma direta, como mostra a comunicação entre o servidor 1 na $DCell_02$ e o servidor 2 na $DCell_03$, ou por uma combinação de servidores e comutadores, como visto na comunicação entre o Servidor 1 na $DCell_01$ e o Servidor 1 na $DCell_04$.

Na arquitetura DCell, os comutadores conectam apenas os servidores de sua DCell. A ligação direta entre diferentes redes DCell é sempre realizada por intermédio dos servidores. Para se construir uma $DCell_k$ são necessárias $n + 1$ redes $DCell_{k-1}$. Cada servidor em uma

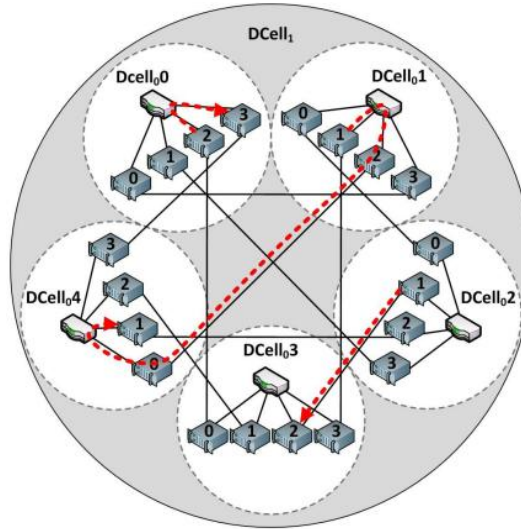


Figura 2.6 – Topologia utilizada pela arquitetura DCell, extraído de (COUTO; CAMPISTA; COSTA, 2012).

rede $DCell_k$ dispõe de $k + 1$ enlaces, o primeiro enlace (enlace de nível 0) em cada servidor é conectado ao comutador da $DCell_0$ do qual ele faz parte, já o segundo enlace conecta o servidor a um nó de uma mesma $DCell_1$, mas em uma $DCell_0$ vizinha. O enlace de nível i de um servidor o conecta a uma $DCell_{i-1}$ vizinha dentro de uma mesma $DCell_i$. O procedimento de construção de uma rede DCell é executado a partir de um algoritmo proposto por Guo *et al.* (GUO et al., 2008).

As Equações 2.1 e 2.2 são utilizadas para calcular a capacidade da rede e o número de servidores, respectivamente.

$$g_k = t_{k-1} + 1 \quad (2.1)$$

$$t_k = g_k * t_{k-1} \quad (2.2)$$

Em que:

g_k é o número de redes $DCell_{k-1}$ no $DCell_k$;

t_k é o número de servidores no $DCell_k$.

A $DCell_0$ é um caso especial em $g_0 = 1$ e $t_0 = n$.

2.4.2 Comparação de Custos entre Topologias de Rede para Centro de Dados

Shang *et al.* (SHANG; LI; XU, 2012) comparam diferentes arquiteturas em termos de custo e consumo energético e mostram que projetos híbridos atingem o menor custo. Além disso, eles mostram que os resultados obtidos são dependentes da carga de trabalho. Para uma carga

de trabalho elástica, é possível reutilizar os núcleos das CPUs para a computação tradicional. Entretanto, uma carga de trabalho inelástica favorece projetos baseados em comutadores. Eles constataram ainda que redes com cargas de trabalho em níveis abaixo de 50% geram as maiores economias nas diferentes topologias. Em contrapartida, cargas de trabalho em níveis altos, acima de 80%, aproximam o consumo energético nas diferentes topologias.

Benson *et al.* (BENSON *et al.*, 2010) analisam as características de tráfego por meio de traços de centros de dados em produção e mostram que a utilização média dos enlaces na camada de agregação é de apenas 8% durante 95% do tempo, enquanto que a utilização dos enlaces na camada de acesso e na camada do núcleo estão aproximadamente entre 20% e 40%, respectivamente. Uma topologia composta por servidores para ser competitiva teria que garantir que os núcleos da CPU fossem dinamicamente compartilhados por ambas as tarefas, quais sejam computação e rede. Uma topologia baseada em servidores representaria uma mudança tão radical em um projeto de rede tradicional que ela poderia encontrar resistência principalmente pelos operadores de centro de dados. As topologias baseadas em servidores devem apresentar vantagens significativas sobre outras topologias além de custos.

Esses trabalhos mostram que a escolha adequada da topologia da rede do centro de dados pode produzir maiores economias de energia. Em particular, topologias com elevado grau de redundância oferecem maiores oportunidades de otimização.

2.5 Conceitos de Séries Temporais

O SimDC3D disponibiliza os algoritmos de predição *RBF* e *Polynom*, descritos nos Apêndices C e D, que podem ser utilizados para prever o comportamento futuro de uma determinada variável a ser utilizada no escalonamento e consolidação de máquinas virtuais. Esta seção apresenta conceitos básicos de séries temporais que são essenciais para se entender o uso e a avaliação desses algoritmos de predição em escalonadores de máquinas virtuais.

A análise de séries temporais permite extrair as tendências dos dados ao longo do tempo. Tais tendências podem ter um comportamento ascendente, descendente ou sazonal, no qual os dados podem ter comportamentos similares em diferentes intervalos de tempo.

Séries temporais são constituídas por observações sequenciais coletadas de uma ou mais variáveis ordenadas no tempo. Para séries univariadas, as observações podem ser representadas no formato (x_1, x_2, \dots, x_n) em que os índices são instantes de tempo $t \in [1, n]$ e n é o número de observações. Em séries multivariadas, k representa o número de variáveis observadas em cada instante de tempo t . Essas séries são representadas no formato $(x_{1t}, x_{2t}, \dots, x_{kt})$, em que $t \in [1, n]$.

A modelagem de séries temporais permite compreender seu comportamento e suas propriedades. Além disso, é possível prever seu comportamento no tempo, o que auxilia no processo de tomada de decisão ao antecipar o comportamento da série. No contexto deste trabalho, busca-se antecipar comportamentos de carga de trabalho, temperatura de entrada dos

servidores e outras variáveis necessárias para o escalonamento de máquinas virtuais em um centro de dados.

Segundo Ishii (ISHII, 2010), o estudo de séries temporais envolve a investigação de seus processos geradores. Esse estudo é necessário para reconhecer o processo gerador de uma série e auxiliar na definição de qual técnica será utilizada na modelagem. Estocasticidade, linearidade e estacionariedade são aspectos que precisam ser identificados e classificados em uma série.

A estocasticidade determina se uma série temporal é determinística ou estocástica. Uma série determinística é caracterizada por um conjunto de dados que se repete ao longo do tempo em escala similar ou em diferentes proporções. Séries determinísticas podem ser modeladas por equações diferenciais determinísticas, predizendo o comportamento futuro por meio de observações passadas. Segundo Ishii (ISHII, 2010), as condições iniciais e perturbações mínimas em alguns desses sistemas podem dificultar previsões de longo prazo. Já em séries temporais estocásticas, as observações futuras são dependentes do estado anterior e de componentes aleatórios. Técnicas estatísticas apresentam melhores resultados para a modelagem de sistemas estocásticos.

Para se determinar a linearidade, é necessário realizar uma análise supondo que cada ocorrência da série seja obtida aleatoriamente com base em uma estrutura probabilística. Uma vez identificadas as estruturas lineares da série temporal, é possível afirmar se ela é linear ou não-linear. Atualmente a técnica que utiliza rede neural multicamadas apresenta um desempenho similar e, em alguns casos, superior em comparação com outras técnicas para essa identificação.

Por último, é necessário avaliar a estacionariedade da série temporal. Segundo Morettin *et al.* (MORETTIN; TOLOI, 2006), uma série temporal é estacionária quando existe um estado particular de equilíbrio estatístico e ela se desenvolve no tempo aleatoriamente em torno de uma média e variância constantes, refletindo alguma forma de equilíbrio. Mudanças no tempo de uma observação original não afetam as propriedades de séries estacionárias (BOX; JENKINS; REINSEL, 1994). Séries temporais que não satisfazem as condições de estacionariedade são denominadas não estacionárias.

Tanto o RBF quanto o Polynom, que estão implementados no SimDC3D, são preditores utilizados com séries temporais estocásticas e não lineares, que são representativas de traços reais de cargas de trabalho de centros de dados.

2.6 Estatísticas para Análise de Desempenho de Preditores em Simulações para Centro de Dados

As seções a seguir apresentam as medidas descritivas que são calculadas para avaliar os preditores implementados no SimDC3D.

2.6.1 Raiz do Erro Quadrático Médio (RMSE)

O Erro Quadrático Médio (*do inglês Mean Square Error*) (MSE), definido na Equação 2.3, é uma medida frequentemente usada na verificação mais precisa de modelos numéricos. O MSE é mais sensível aos grandes erros, por elevar as diferenças individuais ao quadrado gerando resultados sempre positivos. O MSE é útil na comparação entre preditores, principalmente se um deles for viciado. Para analisar os resultados, observa-se a tendência do MSE; quanto mais próximo de zero mais exato é o preditor; chegando a um MSE igual a zero, indica um preditor perfeito.

$$MSE = \sum_{t=1}^n (Q_{Pred}(t) - Q_{Obs}(t))^2 \quad (2.3)$$

Em que:

$Q_{Pred}(t)$: valor predito da variável simulada;

$Q_{Obs}(t)$: valor observado da variável simulada;

A raiz quadrada do MSE, ou Raiz do Erro Quadrático Médio (*do inglês Root Mean Square Error*) (RMSE), é comumente usada para expressar a precisão dos resultados numéricos, com a vantagem de que ela apresenta valores de erro nas mesmas dimensões das variáveis analisadas. A RMSE é calculada pela Equação 2.4.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (Q_{Obs}(t) - Q_{Pred}(t))^2} \quad (2.4)$$

Em que:

$Q_{Pred}(t)$: valor predito da variável simulada;

$Q_{Obs}(t)$: valor observado da variável simulada;

n : número de elementos da amostra;

2.6.2 Coeficiente de Variação

O Coeficiente de Variação (C_v) ou Desvio Padrão Relativo (DPR) é uma medida padronizada de dispersão de uma distribuição de probabilidade ou de uma distribuição de frequência. O C_v é definido como a razão do desvio padrão (σ) pela média (μ), como mostra a Equação 2.5. Os resultados indicam a extensão da variabilidade em relação à média.

$$C_v = 100 * \frac{\sigma}{\mu} \quad (2.5)$$

Para compreender o desvio padrão no contexto da média dos dados é necessário analisar o C_v . Além disso, o C_v é útil na comparação entre conjuntos de dados com diferentes unidades

ou médias muito diferentes. Também, verifica-se que o C_v é adequado para comparar a precisão de diferentes experimentos.

O C_v deve ser calculado apenas para valores não negativos. O intervalo de valores do coeficiente de variação C_v no SimDC3D para temperatura varia entre 10 a 35°C, sendo assim, é possível que seja analisado o C_v de cada servidor.

2.7 Considerações Finais

A redução do consumo energético em servidores, sistema de refrigeração e infraestrutura de rede é um grande desafio para os operadores de centros de dados. Nesse capítulo, foram introduzidos conceitos fundamentais para compreender os desafios necessários para implementar centros de dados verdes. A questão fundamental em todos os conceitos apresentados é a criação de soluções inovadoras, capazes de entregar maior capacidade computacional com menor consumo de energia sem degradação do desempenho que pode levar a uma quebra dos acordos de nível de serviço.

CAPÍTULO 3

TRABALHOS RELACIONADOS

Os principais trabalhos existentes na literatura para economia de energia em centros de dados consideram, normalmente, uma ou duas das principais dimensões de consumo energético. Conseqüentemente, os resultados obtidos nesses trabalhos não avaliam o impacto da otimização em todas as dimensões do centro de dados, tendo como principal consequência, um consumo energético que pode ser até mesmo maior se todas as dimensões fossem consideradas. Por exemplo, se um escalonador de máquinas virtuais utilizar apenas critérios de *hardware* dos servidores para mapear máquinas virtuais em físicas, o resultado final desse mapeamento pode alterar profundamente a distribuição de calor e o tráfego na rede do centro de dados. A nova alocação pode gerar mais calor em certas regiões do centro de dados e necessitar temperaturas mais baixas ou um fluxo maior de ar frio, o que leva ao aumento do consumo energético do sistema de refrigeração. Da mesma forma, a nova alocação pode fazer com que o tráfego aumente em determinados enlaces, resultando em aumento de consumo de energia em seus comutadores. Portanto, pode não ser possível se obter economia de energia nessas dimensões ou, no pior caso, pode haver aumentos superiores às economias obtidas nos servidores.

O restante deste capítulo está organizado da seguinte forma. A Seção 3.1 aborda os principais trabalhos referentes a centro de dados verde. As Seções 3.2, 3.3 e 3.4 apresentam os principais trabalhos para economia de energia nas dimensões servidor, sistema de refrigeração e infraestrutura de rede, respectivamente. Na Seção 3.5, são relacionados e discutidos os principais simuladores para centro de dados. Por último, a Seção 3.6 apresenta as conclusões finais deste capítulo.

3.1 Centros de Dados Verdes

Rong *et al.* (RONG et al., 2016) fazem uma ampla revisão das tecnologias de economia de energia para diferentes estágios de implantação de um centro de dados. Os autores argumentam que um ambiente adequado pode reduzir em torno de 30% o consumo do ar condicionado e

servidores de baixa potência podem reduzir o consumo de 25% a 30%. Além disso, algoritmos de escalonamento e controle podem reduzir entre 10 e 15% do consumo total de energia por meio de otimizações do centro de dados. Os resultados que iremos mostrar nesta tese corroboram partes das observações feitas em (RONG et al., 2016).

Akhter *et al.* (AKHTER; OTHMAN, 2016) discutem vários métodos para reduzir o consumo de energia em centros de dados. Além disso, eles apresentam os principais desafios e questões futuras na gestão energética, como: (i) considerar o uso da energia da memória, em virtude das tecnologias de virtualização exigirem mais memória; (ii) implementar de forma combinada diversos tipos de otimização, como otimização de rede, temperatura e servidores; e (iii) implantar múltiplos estágios de suspensão para fornecer controle avançado de desempenho e de consumo de energia em vários dispositivos do centro de dados. O SimDC3D é uma ferramenta que auxilia pesquisadores a enfrentar esses desafios, pois ele é capaz de simular de forma integrada as três dimensões de maior consumo de um centro de dados. Além disso, esta tese propõe um algoritmo parametrizável que permite atribuir pesos diferentes a essas dimensões de maior consumo.

3.2 Otimizações em Servidores

Para economia de energia nos servidores, há vários trabalhos na literatura (BELOGLAZOV; BUYYA, 2012; GE; SUN; WANG, 2013; RICCIARDI et al., 2011). Beloglazov *et al.* (BELOGLAZOV; BUYYA, 2012) apresentam a algoritmo Power Aware Best Fit Decreasing (PABFD) para realocação de máquinas virtuais. Esse trabalho é considerado seminal na área e o algoritmo PABFD será detalhado no Capítulo 5, pois ele será usado como base de comparações para os algoritmos propostos nesta tese.

Beloglazov *et al.* (BELOGLAZOV; BUYYA, 2010) propuseram uma heurística para realocação dinâmica de máquinas virtuais em um ambiente de centro de dados que oferece infraestrutura-como-um-serviço (IaaS) de acordo com as exigências de recursos e com garantias de qualidade de serviço. Eles consideram uma arquitetura em que os servidores físicos e virtuais são heterogêneos. Os objetivos principais da realocação são minimizar o número de servidores físicos utilizados, para que os servidores ociosos sejam desligados, e diminuir as violações de SLA abaixo de 1%. Cada servidor físico é caracterizado pelo desempenho medido em Milhões de Instruções Por Segundo (MIPS), pela quantidade de memória RAM, pela largura de banda e pela capacidade de armazenamento em disco. Os usuários enviam pedidos de provisionamento de VMs informando as características desejadas de cada VM e o escalonador aloca as VMs em servidores físicos enquanto houver recursos disponíveis.

A arquitetura de gerenciamento proposta contém um administrador global e administradores locais. Um administrador local reside em cada servidor e tem a função de monitorar a utilização dos recursos de cada VM. Ele avalia continuamente a utilização da CPU, redimensiona as VMs

de acordo com suas necessidades de recursos e decide quando e quais VMs devem ser migradas do servidor.

O administrador global reside em um servidor mestre e coleta informações dos administradores locais para manter a visão global de utilização dos recursos das VMs. Com essas informações, o administrador global dispara comandos para realocar as VMs quando necessário. Os administradores locais são responsáveis por atender às solicitações do administrador global, tais como redimensionar ou migrar as VMs e alterar os estados do servidor (ligado, desligado, hibernando). A maioria dos servidores inativos é mantida desligada, mas o administrador global mantém alguns servidores em hibernação para permitir uma resposta rápida em momentos de pico, devido ao menor tempo de transição de hibernando para ligado.

Um outro fator a ser considerado em realocações de VMs é o número de migrações. Segundo estudos de Voorsluys *et al.* (VOORSLUYS *et al.*, 2009), durante a migração de uma VM que está em operação, o desempenho das aplicações nela rodando sofre um impacto negativo. A degradação do desempenho e a inatividade dependem principalmente da quantidade de páginas de memória que são alteradas durante o processo de migração. Cada classe de aplicação apresenta uma degradação diferente. Por exemplo, em uma aplicação web, a degradação de desempenho médio incluindo o tempo de inatividade pode ser estimada em cerca de 10% da utilização da CPU (VOORSLUYS *et al.*, 2009). Esse tempo de inatividade que ocorre em cada migração pode causar violações de SLA, o que motiva a construção de algoritmos que minimizem o número de migrações.

Beloglazov *et al.* (BELOGLAZOV; BUYYA, 2010) propõem uma equação para calcular o custo das migrações (Equação 3.1) que leva em consideração a quantidade total de memória usada em função da largura de banda disponível. Eles também apresentam uma métrica, descrita na Equação 3.2 que mede a quebra dos acordos de nível de serviço de um centro de dados.

$$U_{d_j} = 0.1 \int_{t_0}^{t_0+T_{m_j}} u_j(t) dt, \quad (3.1)$$

$$T_{m_j} = \frac{M_j}{B_j}$$

Em que:

U_{d_j} : é a degradação total do desempenho da VM j ,

t_0 : é o tempo em que a migração é iniciada,

T_{m_j} : é o tempo em que a migração é finalizada,

$u_j(t)$: é a utilização de CPU pela VM j ,

M_j : é a quantidade de memória usada pela VM j ,

B_j : é a largura de banda da rede disponível.

$$SLA = \frac{\sum_{j=1}^M \int_t U_{r_j}(t) - U_{a_j}(t) dt}{\sum_{j=1}^M \int_t U_{r_j}(t) dt} \quad (3.2)$$

Em que:

$U_{r_j}(t)$: total de MIPS solicitado por todas as VMs;

$U_{a_j}(t)$: total de MIPS efetivamente alocado para todas as VMs’;

M : é o número de VMs.

A degradação do desempenho e o SLA são métricas utilizadas para medir o comportamento das estratégias de migração de máquinas virtuais. Estratégias que reduzem o número de migrações são desejáveis, pois reduzem a degradação do serviços existentes no centro de dados. Além disso, alocar corretamente uma máquina virtual em um servidor que não gere sobrecarga futura pode reduzir quebras dos acordos de nível de serviço do centro de dados.

Beloglazov *et al.* (BELOGLAZOV; BUYYA, 2010) também propõem algumas estratégias de migração. A primeira é de limite único *Single Threshold* (ST), que estabelece um limite superior de utilização da CPU para os servidores. As demais heurísticas visam manter a utilização da CPU dos servidores entre limites superiores e inferiores de utilização. Se a utilização da CPU de um servidor cair abaixo do limite inferior, todas as VMs têm que ser migradas desse servidor para outro, permitindo que ele seja desligado. Se a utilização exceder o limite superior, algumas VMs devem ser migradas para reduzir a utilização da CPU para evitar possíveis violações de acordos de nível de serviço.

Lago *et al.* (LAGO; MADEIRA; BITTENCOURT, 2012) propõem duas heurísticas de realocação de máquinas virtuais em centros de dados não federados sem o conhecimento prévio das cargas de trabalho e considerando que os servidores podem ser homogêneos ou heterogêneos. A primeira heurística tem a finalidade de alocar servidores para as máquinas virtuais considerando somente a economia de energia. A segunda heurística, por sua vez, adiciona qualidade de serviço ao escalonamento. Diferentemente de Beloglazov *et al.* (BELOGLAZOV; BUYYA, 2010), Lago *et al.* (LAGO; MADEIRA; BITTENCOURT, 2012) utilizam as tecnologias DVFS e controle de ventoinhas (*Fan Control*) em seus cálculos para estimar o consumo energético de um processador. Apesar do desconhecimento prévio das cargas de trabalho, os algoritmos propostos se adaptam dinamicamente às necessidades de processamento e melhoram o consumo de energia *on-the-fly*.

Para avaliar suas heurísticas, Lago *et al.* (LAGO; MADEIRA; BITTENCOURT, 2012) utilizaram o simulador *CloudSim* (BUYYA; RANJAN; CALHEIROS, 2009), mas implementaram outros escalonadores para comparações, como *Round Robin*, *Best Resource Selection*, *Minimum Power Diff* e *Lago Allocator*. Os resultados mostram economias entre 31,83% e 54,65% em centros de dados pequenos e homogêneos. Porém, os melhores ganhos são obtidos em centros de dados muito grandes e heterogêneos em que eles reportam reduções de até 68,78% no consumo de energia.

3.3 Otimizações no Sistema de Refrigeração

O sistema de refrigeração é responsável em média por 40% do consumo energético de um centro de dados (OXLEY et al., 2018). Um aumento na utilização dos servidores interfere diretamente no desempenho do sistema de refrigeração, pois os servidores aquecem e precisam de mais ar frio para manter suas temperaturas dentro de intervalos aceitáveis. Segundo Moore *et al.* (MOORE et al., 2005), uma política de gerenciamento térmico, que considera o sistema de refrigeração, a disposição física dos centros de dados e os componentes de TI pode gerar benefícios como:

1. **Diminuição dos custos de refrigeração:** o gerenciamento térmico inteligente reduz o consumo energético. Por exemplo, ao se elevar a temperatura de um centro de dados com 1.000 racks em 5°C, pode-se reduzir o consumo energético do sistema de refrigeração de 20% a 40% (MOORE et al., 2005; PATEL et al., 2003).
2. **Aumento da confiabilidade do hardware:** um servidor típico necessita que a temperatura do ar nas suas entradas frontais esteja no intervalo de 20°C a 30°C. A cada 10°C de aumento acima dos 21°C, a confiabilidade da eletrônica diminui em 50% a longo prazo (SULLIVAN, 2000). Cole *et al.* (COLE, 2000) mostram que um aumento de 15°C duplica as taxas de falha nos discos rígidos.
3. **Diminuição dos tempos de resposta transientes:** as condições do centro de dados podem mudar rapidamente devido a picos transitórios na utilização dos servidores ou falhas nas unidades de refrigeração. Essas alterações no comportamento podem causar um aumento de calor no centro de dados devido aos tempos de resposta lentos do sistema de refrigeração, ocasionando danos como os explicados no item anterior.

O fluxo de ar dentro de um centro de dados é complexo, não intuitivo e pode ser alterado facilmente. Alterações na geração de calor dos servidores ou de temperatura nas unidades de refrigeração podem levar alguns minutos para se propagar pela sala, complicando assim a caracterização de fluxo de ar dentro dos centros de dados. Moore *et al.* (MOORE et al., 2005) propõem a alocação ciente de calor em centros de dados em que as unidades de refrigeração podem reagir à elevação da temperatura causada pelo aumento da utilização dos servidores, como também controlar a propagação do calor.

O objetivo da alocação de carga de trabalho ciente de calor é evitar que as temperaturas de entrada e saída de qualquer servidor cruzem um limiar predefinido de segurança. Moore *et al.* (MOORE et al., 2005) propõem três algoritmos para a reduzir os custos de refrigeração com a alocação ciente de calor:

1. **Uniform Workload:** calcula a energia total consumida pelo centro de dados e distribui as cargas de trabalho uniformemente em cada um dos servidores. Por exemplo, se cada servidor consumir 150 Watts quando ocioso e 285 Watts quando está com utilização

máxima, uma distribuição uniforme de 40% coloca $((285-180)*0,40)+150 = 204$ Watts em cada servidor. Dessa forma, as cargas de trabalho são alocadas em cada servidor de forma a não ultrapassar o consumo médio estipulado.

2. **CoolestInlets**: ordena os servidores por suas temperaturas de entrada e aloca as cargas de trabalho em servidores que estão na parte mais fria do centro de dados.
3. **OnePassAnalog**: atribui orçamentos de energia para criar perfil de exaustão uniforme e evitar a formação de quaisquer desequilíbrios de calor ou de "pontos quentes". Na fase de calibração, ele aloca uma carga de trabalho uniforme em cada servidor e observa a temperatura de entrada em cada um deles. Em seguida, o algoritmo seleciona uma tupla de referência (energia, temperatura de saída) representada por (P_{ref}, T_{ref}^{out}) . Este ponto de referência pode ser um servidor, ou a média de consumo de energia dos servidores, e as temperaturas de saída dentro de um corredor ou em todo o centro de dados (T_i^{out}). Com estes valores, o algoritmo calcula o orçamento de energia para cada servidor usando a equação:

$$P_i = \frac{T_{ref}^{out}}{T_i^{out}} * P_{ref} \quad (3.3)$$

O orçamento de energia de um servidor P_i é inversamente proporcional à sua temperatura de saída, T_i^{out} . A carga de processamento aumenta para os servidores em áreas frias e diminui para os servidores em áreas quentes. O *OnePassAnalog* responde ao acúmulo de calor e altera o orçamento de energia nos locais onde há aumento de calor.

Dentre os três algoritmos descritos, o algoritmo *OnePassAnalog* é o que apresenta uma menor variação da temperatura em um centro de dados com 60% de utilização (em torno de 4°C), o que indica a ocorrência de menos pontos quentes e desequilíbrios de calor (MOORE et al., 2005).

3.4 Otimizações na Infraestrutura de Rede

Abts *et al.* (ABTS et al., 2010) propõem monitorar a utilização dos enlaces, estimar dinamicamente suas necessidades futuras e reduzir as taxas de transmissão dos enlaces quando possível para se obter economia de energia. Mahadevan *et al.* (MAHADEVAN et al., 2009a; MAHADEVAN; SHAH; BASH, 2010; MAHADEVAN; BANERJEE; SHARMA, 2010), Gunaratne *et al.* (GUNARATNE et al., 2008) e Nedevschi *et al.* (NEDEVSCHI et al., 2008) propõem métodos semelhantes na redução do consumo de energia em comutadores. Como essas propostas atuam somente na variação do tráfego da rede, elas resultam em pouca economia de energia e não são suficientes para se atingir a proporcionalidade de consumo desejada, pois o consumo fixo de energia de um comutador é muitas vezes superior ao seu consumo variável que depende do volume de tráfego.

Heller *et al.* (HELLER et al., 2010) apresentam o *ElasticTree*, um otimizador que monitora continuamente as condições de tráfego da rede e calcula dinamicamente um conjunto de comutadores e enlaces que devem permanecer ligados para cumprir metas de desempenho e tolerância a falhas. Os fluxos de dados são consolidados nesse conjunto e os elementos desnecessários são desligados. Heller *et al.* avaliaram o *ElasticTree* experimentalmente e mostraram que ele reduz de 25% a 40% o consumo médio de energia da infraestrutura de rede. Os experimentos da Seção 6.4.4 corroboram esses resultados e mostram economias de até 31,34%. Além disso, mostram que algumas camadas da topologia da rede podem ter economias de mais de 70%.

Carpo (WANG et al., 2012) trabalha de forma semelhante ao *ElasticTree*, mas consolida os fluxos de dados com base em suas correlações. Utilizando traços reais, *Carpo* mostra que as demandas por largura de banda de diferentes fluxos não sofrem picos exatamente ao mesmo tempo. Com isso, ele aloca dois fluxos no mesmo enlace somente se eles tiverem correlações negativas, ou seja, tiverem picos de utilização em momentos distintos. Outra observação importante em (WANG et al., 2012) é que 90% das demandas de largura de banda são geralmente menores ou iguais à metade das demandas de pico, permitindo que fluxos sejam consolidados sem extrapolar as capacidades dos enlaces. Segundo Wang *et al.* (WANG et al., 2012), são necessárias três etapas para a consolidação do tráfego: (i) análise de correlação entre os fluxos; (ii) consolidação dos fluxos nos enlaces selecionados; e (iii) mudanças nas taxas de transmissão dos enlaces. Os resultados experimentais demonstraram que *Carpo* pode economizar até 46% de energia em uma rede de centro de dados, com um aumento não significativo no atraso. *Carpo* obteve economias de energia de 19% em relação ao *ElasticTree* e de 95% em relação ao trabalho de Abts *et al.* (ABTS et al., 2010) (*GoogleP*).

3.5 Softwares de Simulação para Ambientes de Centro de Dados

A avaliação de políticas de economia de energia em um centro de dados é extremamente desafiadora. Idealmente, novos algoritmos deveriam ser avaliados em um centro de dados real e com cargas de trabalho representativas de aplicações reais. Entretanto, as universidades e institutos de pesquisa não possuem *testbeds* para esse tipo de avaliação e apenas algumas poucas empresas no mundo possuem centros de dados de escala adequada. A alternativa mais viável, portanto, para esse tipo de avaliação é a simulação. Gupta *et al.* (GUPTA et al., 2011) enumera diversas características que um ambiente de simulação de centro de dados deve ter:

1. **Processamento automatizado:** executar a simulação sem a intervenção do usuário;
2. **Capacidade de análise on-line:** tomar decisões em tempo real com base nas mudanças do ambiente físico no centro de dados;

3. **Análise interativa do projeto:** permitir testes durante o projeto e análises em diferentes configurações do centro de dados antes de sua implantação;
4. **Capacidade de análise térmica:** disponibilizar análise térmica em tempo real, caracterizando os efeitos térmicos que ocorrem no centro de dados após o escalonamento e a consolidação de máquinas virtuais;
5. **Gerenciamento de energia e carga de trabalho:** possibilitar o escalonamento de cargas de trabalho, além de controlar os modos de operação dos servidores (ligado, desligado, hibernando, entre outros.);
6. **Consideração da interdependência ciberfísica:** proporcionar retroalimentação de informações sobre padrões de temperatura e fluxos de ar no centro de dados para os algoritmos de gestão e operação dos servidores e unidades de refrigeração para alcançar uma operação eficiente de energia.

Além dessas características listadas em (GUPTA et al., 2011), este trabalho argumenta que um ambiente de simulação deve incorporar as seguintes características:

1. **Gerenciamento da infraestrutura de rede:** simular a infraestrutura da rede e relacioná-la com as demais dimensões do centro de dados;
2. **Estratégias de escalonamento e consolidação:** disponibilizar mecanismos para implementar algoritmos de escalonamento e consolidação de máquinas virtuais que utilizem informações geradas por todas as dimensões do centro de dados em suas decisões de alocação;
3. **Capacidade de predição:** disponibilizar ferramentas de predição para apoiar decisões de escalonamento e consolidação de VMs.

Um simulador permite a análise de um centro de dados, concentrando-se nas questões de qualidade e de componentes específico em diferentes cenários. Esta ferramenta permite avaliar hipóteses, além de oferecer benefícios significativos, como testar serviços em um ambiente controlável, experimentar diferentes cenários de carga de trabalho e avaliar o desempenho dos recursos em uma infraestrutura simulada para desenvolver e testar técnicas de economia de energia.

Para comparar as características do SimDC3D com outros simuladores semelhantes, realizamos um extensa busca na literatura para identificá-los. Os simuladores mais representativos encontrados na literatura científica são: CloudSim (CALHEIROS et al., 2011), SPECI (Simulation Program for Elastic Cloud Infrastructures) (SRIRAM, 2009), Green-Cloud (KLIAZOVICH; BOUVRY; KHAN, 2012), OCT (Open Cloud Testbed) (GROSSMAN et al., 2009), Open Cirrus (AVETISYAN et al., 2010), GroudSim (OSTERMANN et al., 2011), NetworkCloudSim (GARG; BUYYA, 2011), EMUSIM (GUSTEDT; JEANNOT;

QUINSON, 2009), DCSim (TIGHE et al., 2012), iCanCloud (NÚÑEZ et al., 2010), CDOSim (FITTKAU; FREY; HASSELBRING, 2012), TeachCloud (JARARWEH et al., 2013), SimWare (YEO; LEE, 2012). Dentre eles, cinco simuladores (CloudSim (CALHEIROS et al., 2011), NetworkCloudSim (GARG; BUYYA, 2011), DCSim (TIGHE et al., 2012), GreenCloud (KLIAZOVICH; BOUVRY; KHAN, 2012) e o SimWare (YEO; LEE, 2012)) foram selecionados para comparações com o SimDC3D e um resumo com as principais características dos simuladores pode ser visto na Tabela 3.1.

O simulador SimWare (YEO; LEE, 2012) disponibiliza um ambiente de simulação para centro de dados virtualizado. Desenvolvido em C++, ele permite estender as políticas de escalonamento de máquinas virtuais. O SimWare implementa o sistema de refrigeração, servidores e máquinas virtuais. Entretanto, ele não disponibiliza recursos para a migração de máquinas virtuais e nem implementa a infraestrutura de rede. Além dessas limitações, a utilização de CPU de cada máquina virtual é extraída de um traço real e mantida constante durante toda a simulação, ou seja, o SimWare não implementa variabilidade de utilização da CPU. A ausência dessa funcionalidade o impossibilita de fazer análises realistas de quebras de acordo de nível de serviço, pois, como a utilização é constante, o escalonador nunca aloca uma máquina virtual em um servidor que não tenha capacidade de CPU disponível para recebê-la.

O CloudSim (CALHEIROS et al., 2011) visa oferecer os recursos necessários para a simulação de ambientes computacionais em nuvem. Desenvolvido inteiramente em Java, o simulador é extensível, adaptável e permite a criação de simulações em grande escala com alto grau de customização. A arquitetura do CloudSim é composta de três níveis: (i) motor de simulação; (ii) classes funcionais; e (iii) código de usuário. O CloudSim permite a implementação de políticas de alocação de máquinas virtuais entre os servidores de um mesmo centro de dados, escalonadores de máquinas virtuais dentro de um servidor e escalonadores de *cloudlets* em máquinas virtuais. Entretanto, o CloudSim não simula o sistema de refrigeração do centro de dados.

O Network Cloud (CALHEIROS et al., 2011) é uma versão estendida do CloudSim e foi projetado para implementar a camada de rede no CloudSim. Ele usa um arquivo BRITE como entrada que descreve a topologia da rede. Os arquivos de topologia contêm o número de nós e as várias entidades envolvidas na simulação. Cada entidade deve ser mapeada para um único nó BRITE para que a rede CloudSim possa funcionar de forma eficiente. O Network CloudSim geralmente é usado para simular o comportamento do tráfego de rede no CloudSim.

O DCSim (TIGHE et al., 2012) simula a camada de IaaS e se concentra em centros de dados virtualizados. No DCSim, um centro de dados consiste em um conjunto de servidores interconectados, regidos por um conjunto de políticas de gerenciamento. Cada servidor possui um conjunto de gerenciadores de recursos que lidam com a alocação de recursos locais, um agendador de CPU que decide como as VMs são executadas e um modelo de potência que mostra quanta energia está sendo consumida por cada servidor. Cada servidor executa um conjunto de VMs e cada VM executa um único aplicativo. O DCSim suporta operações de

Tabela 3.1 – Características de simuladores para centro de dados versus SimDC3D.

	SimDC3D	SimWare	CloudSim	Network CloudSim	DCSim	GreenCloud
Código Aberto	✓	✓	✓	✓	✓	✓
Tempo Simulação	Segundos	Segundos	Segundos	Segundos	Minuto	Minuto
Modelo Detalhado de Energia	✓	✓	✓	✓	✗	✓
Modelo de Carga de Trabalho	Traços Reais / Cargas Sintéticas	Traços Reais	Cargas Sintéticas	Cargas Sintéticas	Cargas Sintéticas	Cargas Sintéticas
Recursos	CPU, Memória, Ventoinhas, Comutadores, Sistema de Refrigeração	CPU, Ventoinhas, Sistema de Refrigeração	CPU, Memória, Disco	CPU, Memória, Disco, Comutadores	CPU	CPU, Comutadores
Ventilação Interna	✓	✓	Limitado	Limitado	✗	✗
Refrigeração	(HRM)	(HRM)	✗	✗	✗	✗
Rede	✓	✗	✗	✓	✗	✓
Modelo Comunicação da Rede	✓	✗	Limitado	Nível Pacote	✗	Nível Pacote
Modelo de Energia da CPU	Modelo Linear	Modelo Linear	Modelo Linear	Modelo Linear	Modelo Linear	Baseado Frequência
Suporta VMs	✓	✓	✓	✓	✓	✗
Escalonamento de VMs	✓	✓	✓	✓	✓	✗
Migração de VMs	✓	✗	✓	✓	✓	✗
Políticas Seleção de VMs	✓	✗	✓	✓	✗	✗
Políticas de Controle Temperatura	✓	Limitado	✗	✗	✗	✗
Simula Diferentes Topologias	✓	✗	✗	✗	✗	✓
Políticas de Otimização Topologia	✓	✗	✗	✗	✗	✗
Modo de Operação dos Servidores	Inicializando, Ligado, Desligado, Hibernando, Desligando	Ligado	Ligado, Hibernando	Ligado, Hibernando	Ligado, Hibernando	Ligado, Desligado
QoS	✓	✗	✓	✓	✓	✗
Preditores	✓	✗	✓	✓	✗	✗

gerenciamento das VMs, como migração e replicação on-line. As necessidades de recursos de cada VM no DCSim são simuladas por uma aplicação que varia o nível de recursos requeridos pela VM para simular uma carga de trabalho real.

O GreenCloud (KLIAZOVICH; BOUVRY; KHAN, 2012) é um simulador em nível de pacote para centro de dados virtualizado com foco na comunicação em uma nuvem. Ele oferece uma modelagem detalhada da energia consumida pelos equipamentos de TI do centro de dados, como servidores, comutadores e enlaces de comunicação. O GreenCloud pode ser usado para desenvolver novas soluções de monitoramento, alocação de recursos, escalonamento de carga de trabalho, bem como otimização de protocolos de comunicação e infraestrutura de rede. O GreenCloud é uma extensão do simulador de rede NS2 (ISSARIYAKUL; HOSSAIN, 2010). As principais características do GreenCloud são: foco em redes na nuvem e conscientização sobre energia; simulação de recursos de CPU, memória, armazenamento e rede; modelos de energia independentes para cada tipo de recurso; suporte de virtualização e migração de VMs; alocação de recursos com reconhecimento de rede; e implementação completa dos protocolos TCP/IP.

3.6 Considerações Finais

Este capítulo apresenta os principais trabalhos relacionados em cada dimensão de centros de dados que são alvos de economia de energia. O que se pode perceber é que a maioria dos trabalhos analisam as dimensões de modo isolado ou no máximo em pares. Além disso, as ferramentas de simulação existentes não simulam adequadamente todas as dimensões de forma integrada. A Tabela 1.1 no Capítulo 1 mostra como as propostas apresentadas neste capítulo lidam com os diversos parâmetros necessários para se ter um ambiente de simulação adequado para avaliação de algoritmos para economia de energia em centros de dados.

CAPÍTULO 4

O SIMULADOR SIMDC3D

Simulações de centro de dados são complexas e requerem uma grande quantidade de informações e tempo de processamento para atingir um grau de realismo satisfatório. Atualmente, nenhum simulador descrito na literatura científica incorpora adequadamente todas as três dimensões (3D) de maior consumo energético de um centro de dados (servidores, sistema de refrigeração e infraestrutura de rede). O SimDC3D foi desenvolvido para ser um simulador que reproduz realisticamente essas três dimensões.

O SimDC3D disponibiliza vários algoritmos de escalonamento de máquinas virtuais, esquemas de controle de temperatura e topologias de rede. Além disso, ele simula o fluxo de ar frio entre o sistema de refrigeração e os dispositivos do centro de dados, o consumo de energia das CPUs e das ventoinhas dos servidores, o consumo de energia dos dispositivos de rede e a recirculação de calor dentro do centro de dados. O SimDC3D foi baseado inicialmente no SimWare (YEO; LEE, 2012), mas evoluiu significativamente para incluir novas funcionalidades como migração de máquinas virtuais, infraestrutura de rede, algoritmos de otimização de topologia e modelos de tráfego de rede, uso de um banco de servidores de reserva para atender altas demandas em curto espaço de tempo, algoritmos de predição e novos algoritmos de escalonamento. O simulador contém mais de 15 mil linhas de código em C++ e está disponível em (MORO, 2018).

O restante deste capítulo está organizado da seguinte forma. A Seção 4.1 propõe uma arquitetura genérica para gerenciamento de energia em um centro de dados e que é usada como base para o desenvolvimento do SimDC3D. Na Seção 4.2, são apresentadas decisões de projeto que levaram à concepção do SimDC3D. A Seção 4.3 apresenta a arquitetura de implementação do SimDC3D e sua integração com outros módulos de simulação que compõem. Ela descreve também os principais parâmetros e classes do simulador. A Seção 4.4 discute os principais resultados de simulação reportados pelo SimDC3D. Por último, a Seção 4.5 apresenta as conclusões finais deste capítulo.

4.1 Arquitetura Genérica de Gestão de Energia para Centros de Dados Virtualizados

Os centros de dados são normalmente provisionados para atender às demandas de pico. Entretanto, a carga de trabalho não é constante e apresenta na maior parte do tempo períodos de baixa demanda. Segundo Kliazovich *et al.* (KLIAZOVICH; BOUVRY; KHAN, 2010) apenas 30% em média dos recursos são necessários nos períodos de baixa utilização. Isso significa que os recursos restantes (70%) podem ser desligados, suspensos ou colocados em modos de baixo consumo para economizar energia. Para obter maior eficiência no gerenciamento de recursos, é desejável que um controlador central tenha uma visão global do centro de dados para alocar as cargas de trabalho em um conjunto mínimo de recursos computacionais e de comunicação e reprogramar o sistema de refrigeração de acordo com a demanda.

Adotar um controlador central, que tome decisões levando em consideração as três dimensões de maior consumo ao mesmo tempo, é um avanço em relação a diversos estudos (BELOGLAZOV; BUYYA, 2010; BELOGLAZOV; BUYYA, May; HELLER *et al.*, 2010; KANT; MURUGAN; DU, 2011; MENG; PAPPAS; ZHANG, 2010; MOORE *et al.*, 2005; RICCIARDI *et al.*, 2011; SHIRAYANAGI; YAMADA; KONO, 2012; WANG *et al.*, 2012) que propuseram otimizações para melhorar a eficiência energética em uma dimensão específica sem analisar seus impactos nas outras dimensões do centro de dados.

Para exemplificar essa afirmação, observe novamente a Figura 2.1 que mostra as temperaturas de entrada nas partes frontais dos *racks*. Pode-se observar que as temperaturas estão variando entre 13.61°C (56.5°F) e 27.55°C (81.6°F). Portanto, é possível concluir que o sistema de refrigeração não entrega ar frio de modo igual em todo o centro de dados. Servidores que estão localizados em regiões mais quentes do centro de dados sofrem com a escassez de ar gelado. Um otimizador que alocar máquinas virtuais para servidores que estão com as temperaturas de entrada elevadas vai aumentar sua carga de trabalho e gerar vários efeitos indesejáveis. Primeiro, a temperatura do servidor aumentará por causa do incremento da carga de trabalho no processador. Segundo, as ventoinhas dos servidores aumentarão suas rotações para manter as temperaturas baixas e consumirão mais energia. Por último, o sistema de controle térmico diminuirá a temperatura do ar condicionado para evitar sobreaquecimento dos servidores e consumirá mais energia.

A Figura 4.1 apresenta uma proposta de arquitetura de gestão energética para centro de dados virtualizados com um controlador central. O controlador é responsável por escalonar e consolidar máquinas virtuais, otimizar a topologia da rede e controlar a temperatura do ar condicionado. Os módulos monitores alimentam os diversos algoritmos de otimização com informações dos servidores referentes a utilização de CPU e memória, volume de tráfego de rede e temperatura. O objetivo do controlador é maximizar a economia de energia em todo o centro de dados e não mais em dimensões específicas. Além disso, ele deve minimizar as violações de SLA e garantir a qualidade-de-serviço (QoS) das aplicações.

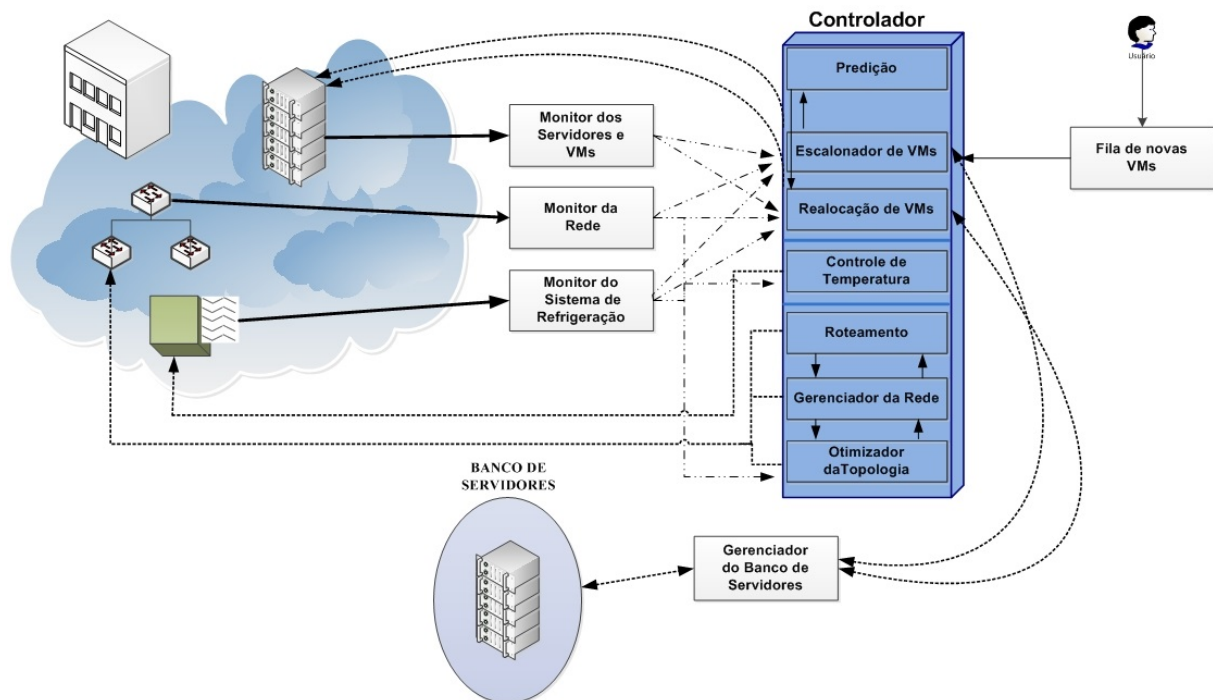


Figura 4.1 – Proposta de uma arquitetura de gestão energética centralizada.

Uma arquitetura com todos os módulos integrados e com o controlador central tomando decisões com informações globais visa aumentar a confiabilidade das decisões e aumentar a eficiência energética do centro de dados. Os módulos da arquitetura têm as seguintes funções:

- **Filas de novas VMs:** armazenam e ordena as VMs que serão alocadas no centro de dados.
- **Monitores:** coletam informações de utilização e estatísticas dos vários subsistemas do centro de dados—como filas de escalonamento, servidores, comutadores e sistema de refrigeração—e alimenta os algoritmos de otimização e de realocação com as informações coletadas.
- **Predição:** analisa séries temporais produzidas pelos módulos monitores e calcula novos parâmetros para o escalonador de VMs. A função deste módulo é oferecer a possibilidade de se fazer predição com base em múltiplas variáveis que são monitoradas.
- **Escalonador de VMs:** aloca as novas VMs no centro de dados objetivando reduzir o consumo energético.
- **Realocação de VMs:** recalcula a localização das VMs no centro de dados objetivando: a) minimizar o número de máquinas físicas; b) aproximar máquinas que tenham uma alta correlação de tráfego; c) distribuir o calor gerado pelo processamento dos servidores nas regiões mais frias do centro de dados; e d) eliminar o congestionamento de enlaces .

- **Controle de temperatura:** analisa as condições atuais de temperatura do centro de dados e dos servidores e determina novos parâmetros para o sistema de refrigeração como, por exemplo, a temperatura do ar frio a ser fornecido.
- **Roteamento:** calcula as rotas na rede do centro de dados.
- **Módulo gerenciador da rede:** encaminha as solicitações para desativar comutadores, de alteração de rota e de velocidade dos enlaces para os comutadores.
- **Otimizador da topologia:** verifica a possibilidade de desligar enlaces e comutadores com base em medições do tráfego de rede para produzir uma nova topologia que consuma menos energia.
- **Gerenciamento do banco de servidores:** gerencia um conjunto de servidores que ficam desocupados para atender às demandas dos escalonadores de VMs quando não há nenhum servidor disponível.

O restante deste capítulo detalha o simulador SimDC3D, que é uma implementação desta arquitetura proposta.

4.2 Concepção do SimDC3D

O SimDC3D implementa a arquitetura de gestão de energia discutida na Seção 4.1 e foi desenvolvido a partir de uma versão pública do simulador SimWare (YEO; LEE, 2012). A decisão de se usar o SimWare como base foi devida à sua capacidade de simular alguns subsistemas importantes, como ventilação dos servidores e recirculação de calor dentro do centro de dados. Para simular as temperaturas de entrada de cada servidor, o SimWare estima o fluxo de calor gerado por todos os servidores ativos do centro de dados utilizando uma matriz de recirculação de calor (HRM). O consumo energético do sistema de refrigeração é calculado pelas equações propostas por Moore *et al.* (MOORE *et al.*, 2005). Já o tempo de propagação do ar frio no centro de dados é calculado por um algoritmo de termodinâmica que calcula o tempo de percurso desde a saída de ar no piso elevado até a entrada de ar dos servidores. Para calcular o consumo de energia do sistema de ventilação de um servidor, o SimWare considera a temperatura de entrada do servidor e a carga de processamento da CPU. A partir dessas informações ele calcula a rotação dos ventiladores e o consumo energético.

Apesar dessas funcionalidades serem fundamentais para a simulação de um centro de dados e de oferecer alguns algoritmos de escalonamento, o SimWare não suporta várias funcionalidades necessárias para se realizar simulações realistas de um centro de dados, como:

1. **Migração de VMs:** o SimWare não simula migração de máquinas virtuais. Quando novas VMs chegam ao centro de dados, elas são escalonadas em um determinado servidor e permanecem no mesmo servidor até o fim de sua execução.

2. **Consolidação de VMs:** mecanismos e políticas de consolidação de VMs também não estão disponíveis. O SimWare não desativa temporariamente e nem realoca as VMs para minimizar o número de servidores ativos, ou seja, ele não faz provisionamento dinâmico.
3. **Infraestrutura de rede:** o SimWare não simula a infraestrutura de rede.
4. **Algoritmos de predição:** o SimWare não oferece suporte para a utilização de preditores no escalonamento de VMs.
5. **CPU:** o SimWare simula o consumo energético da CPU, porém as cargas de trabalho de cada VM são constantes, conforme extraídas do traço. Essa característica impossibilita estudos de violação de SLA, pois os escalonadores alocam as cargas de trabalho nas CPUs com limite constante e inferior a 100% de utilização.
6. **Memória:** os servidores e máquinas virtuais não possuem especificações de utilização de memória.
7. **Acordo de nível de serviço (SLA):** o SimWare não simula SLA, pois as VMs produzem cargas constantes de utilização de CPU e não capturam variabilidades de uso de CPU durante a execução.
8. **Estados dos servidores:** o SimWare implementa dois modos (ligado/desligado) para descrever o estado de um servidor. Independentemente de estarem executando VMs, todos os servidores permanecem ligados durante toda a simulação e consumindo energia mesmo quando ociosos.
9. **Banco de servidores:** o SimWare não implementa banco de servidores.
10. **Políticas de ajustes de temperatura:** há duas políticas de controle do termostato do ar condicionado no SimWare. A primeira é constante e a segunda é dinâmica, mas com uma única estratégia de ajuste.

Outra limitação importante do SimWare e que torna simulações extremamente demoradas é sua implementação em uma única *thread*. Isso faz com que ele seja incapaz de explorar arquiteturas com múltiplos processadores ou núcleos (*cores*) de processamento para acelerar as simulações. O SimDC3D, por outro lado, foi implementado com múltiplas *threads* para acelerar a simulação e explorar os processadores modernos com múltiplos núcleos de processamento. No SimDC3D, as seguintes subrotinas são executadas em paralelo: simulação dos servidores e suas máquinas virtuais, execução do cálculo da temperatura de cada servidor, simulação do fluxo de dados e cálculo da correlação de tráfego. A paralelização dessas subrotinas resultou em ganhos significativos nos tempos de execução. Simulações que levavam mais de um no SimWare passaram a executar em menos de uma hora. Além disso, o SimDC3D estende significativamente as funcionalidades do SimWare e remove todas as limitações listadas acima.

O principal objetivo do SimDC3D é disponibilizar um ambiente de simulação para que se possa combinar diversas estratégias de economia de energia em cada uma das dimensões e obter o desempenho energético do centro de dados como um todo. Com essa arquitetura, é possível avaliar estratégias de otimização e seus respectivos resultados em um ambiente integrado, demonstrando efetivamente as principais vantagens e desvantagens de uma proposta de economia para centro de dados. Por exemplo, nos simuladores atuais, testar uma política de migração resulta em obter o consumo energético da dimensão servidor. Entretanto, uma política de migração que não avalie características térmicas do centro de dados pode gerar um impacto negativo na dimensão refrigeração e também um aumento no consumo energético do sistema de ventilação dos servidores.

Moore *et al.* (MOORE et al., 2005) relatam que é possível diminuir o consumo de energia no sistema de refrigeração aumentando a temperatura do centro de dados. Essa estratégia somente é viável se alocarmos as cargas de trabalho em um conjunto de servidores de forma que seja possível aumentar a temperatura do centro de dados sem ultrapassar os limites térmicos de cada servidor ativo. Portanto, uma ferramenta de simulação que seja capaz de avaliar todos os subsistemas relevantes para uma estratégia de economia de energia é fundamental para estudos confiáveis.

4.3 Arquitetura do SimDC3D

A Figura 4.2 mostra a arquitetura de implementação do SimDC3D. As funcionalidades da arquitetura foram implementadas em 17 classes (*CRAC*, *DataCenter*, *DataFlow*, *JobQueue*, *PredictionAlgorithms*, *TopologyOptimizationAlgorithms*, *PoolServers*, *SchedulingAlgorithm*, *Rack*, *Server*, *ServerOptimizationAlgorithms*, *SingleJob*, *TopologySimDC3D*, *SWFLine*, *Users*, *TrafficMatrixSimDC3D*, *VirtualMachine*) sendo que algumas podem ser estendidas para modelar e simular novas estratégias de economia de energia para um centro de dados virtualizado.

Além das principais classes do simulador, a Figura 4.2 mostra os arquivos de entrada que são necessários para executar uma simulação (*e.g.*, traço, topologia, matriz de tráfego) e as estatísticas produzidas pelo simulador. As cinco classes não mostradas dão suporte às classes principais ilustradas na figura. A figura também mostra a sequência de execução de uma simulação, indicada pelas linhas sólidas, e a interação entre as classes, representada pelas linhas tracejadas.

O SimDC3D permite que pesquisadores estendam as classes *CRAC*, *SchedulingAlgorithm*, *ServerOptimizationAlgorithms*, *TopologySimDC3D* e *TopologyOptimizationAlgorithms* para incorporar novas estratégias de controle de temperatura, algoritmos de escalonamento e realocação de máquinas virtuais, algoritmos de roteamento e otimização de topologia de rede, disponibilizando, portanto, um ambiente de simulação flexível para estudos de consumo energético em centros de dados.

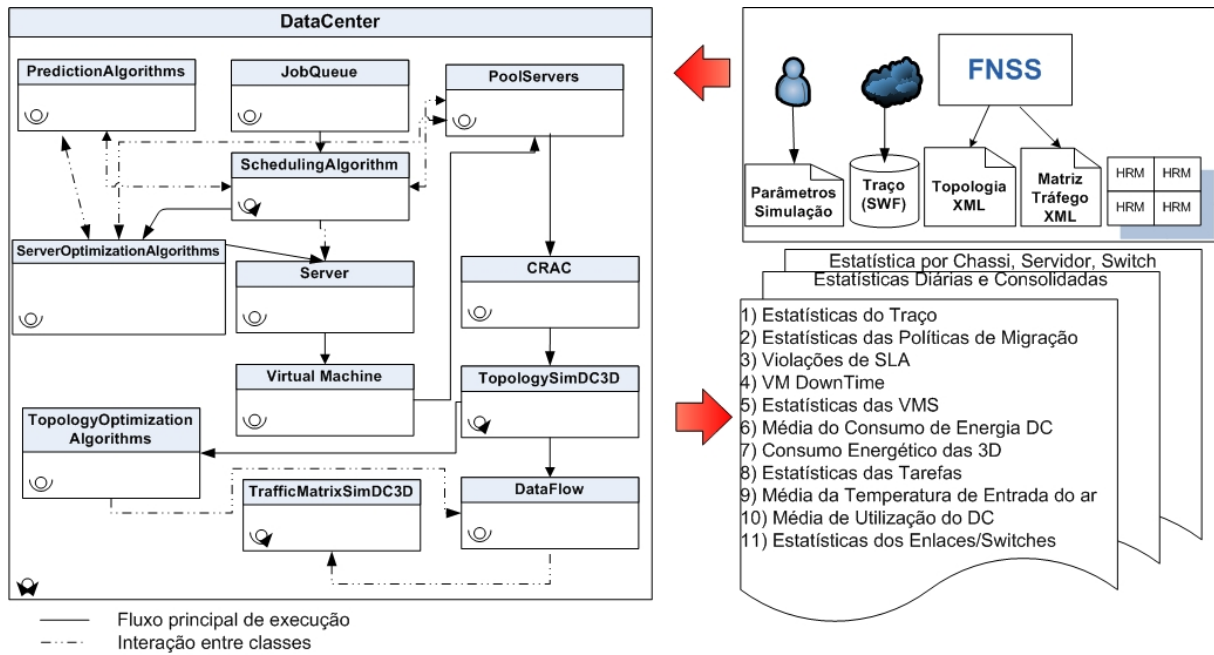


Figura 4.2 – Arquitetura do SimDC3D.

4.3.1 Arquivos de Entrada

Para executar uma simulação, são necessários os seguintes arquivos: parâmetros de simulação, cargas de trabalho, topologia de rede, matriz de tráfego e uma matriz de recirculação de calor. Os parâmetros de simulação são definidos em arquivo no formato texto usando um par *nome=valor* para cada parâmetro. O arquivo com as cargas de trabalho deve estar no Formato de Carga de Trabalho Padrão (*do inglês Standard Workload Format*) (SWF) e pode ser obtido em vários repositórios na Internet como, por exemplo, em (HEBREW, 2018). Por último, os arquivos de topologia e tráfego de rede são gerados pelo simulador *Fast Network Simulation Setup* (FNSS) que será discutido mais detalhadamente na Seção 4.3.1.3. As próximas seções detalham os parâmetros do simulador e os arquivos auxiliares.

4.3.1.1 Parâmetros de Simulação

O arquivo de configuração contém todos os parâmetros necessários para configurar uma simulação. Ao todo são 137 parâmetros divididos em 14 categorias que cobrem as diferentes configurações do centro de dados. Os principais parâmetros são resumidos abaixo. Detalhes adicionais e demais parâmetros estão descritos no Apêndice B.

1. **Parâmetros gerais:** definem os parâmetros gerais da simulação, como tempo de simulação, sementes para os geradores de números aleatórios, número de *threads*, arquivos de log, entre outros.
2. **Configurações dos servidores:** contém as especificações dos servidores, como quantidade de chassis, servidores por chassis, processadores e quantidade de memória

de cada servidor e consumo de energia das ventoinhas dos servidores. Também são definidos nesta categoria os tempos para mudança de estado (*power on*, *power off*, *hibernating on*, *hibernating off*), o consumo energético do servidor em hibernação e o modelo de consumo de energia dos servidores (Modelo Linear, PowerModelSpecPower, entre outros.).

3. **Configurações das VMs:** especificam as configurações das máquinas virtuais, como quantidade de memória RAM e tamanho da imagem da VM em disco. Além disso, nesta categoria é definido o parâmetro usado para calcular o tempo de *downtime* durante a migração de uma VM.
4. **Parâmetros do ar condicionado (CRAC):** especificam a política de controle de temperatura (dinâmica ou estática), tempo de viagem de ar frio (instantâneo ou não) e as temperaturas máximas e mínimas permitidas nos servidores.
5. **Definições e parâmetros do algoritmo de escalonamento:** definem qual algoritmo de escalonamento será utilizado e o tipo de preditor (se necessário). O SimDC3D possui dois preditores (Polynom e RBF) e 13 algoritmos de escalonamento que serão detalhados na Seção 5.1.
6. **Parâmetros gerais dos algoritmos de predição:** especificam os parâmetros gerais dos algoritmos de predição, como tamanho da série histórica e o intervalo de medição da variável que será predita. Por exemplo, se for definido um valor de 60 para o intervalo de medição, o controlador armazena o valor da variável na série histórica a cada 60 segundos.
7. **Parâmetros do algoritmo de predição RBF:** definem os parâmetros do algoritmo RBF, como dimensão, atraso, centro, passo, entre outros.
8. **Parâmetros do algoritmo de predição Polynom:** definem os parâmetros do algoritmo Polynom, como a ordem polinomial e o *cast*.
9. **Definições e parâmetros das cargas de trabalho das VMs:** especificam como as cargas de trabalho são simuladas. O SimDC3D permite dois modos: estático ou dinâmico. No modo estático, o percentual de utilização de CPU é extraído do traço e a VM que processa a carga de trabalho tem uma utilização de CPU constante. No modo dinâmico, é usada uma distribuição uniforme para simular a variabilidade de CPU gerada pela carga de trabalho da VM.
10. **Definições e parâmetros dos algoritmos de migração das VMs:** determinam quais algoritmos são utilizados em cada uma das políticas de migração. Também estão disponíveis os parâmetros utilizados pelos algoritmos quando necessário. As políticas de migração estão descritas na Seção 4.3.2.1 e os algoritmos na Seção 5.2.

11. **Parâmetros do algoritmo 3DMOBFD:** contém os parâmetros do algoritmo 3DMOBFD, como as variáveis que compõem o peso, o peso de cada variável e os expoentes de cada componente do peso. O algoritmo 3DMOBFD está detalhado na Seção 5.1.1.
12. **Parâmetros da infraestrutura de rede:** especificam o número de racks e quantidades de portas e consumos energéticos do chassi, dos *line cards* e das portas de cada tipo de comutador, além do nome do arquivo FNSS com a topologia da rede.
13. **Parâmetros da Matriz de Tráfego:** especificam o nome do arquivo com a matriz de tráfego gerada sinteticamente, número de nós de rede, intervalo e número de períodos da matriz de tráfego, sementes dos geradores de números aleatórios para se determinar períodos de rajada e silêncio, percentuais de comunicação interna e externa do centro de dados, algoritmo de otimização de topologia de rede e se há ou não ajuste de velocidade de enlaces.
14. **Parâmetro do banco de servidores:** especifica o tamanho do banco de servidores.

A lista completa de parâmetros está definida no Apêndice B.

4.3.1.2 Cargas de Trabalho

O SimDC3D utiliza um arquivo de traço no formato SWF extraído de *clusters* reais (HEBREW, 2018) para simular as cargas de trabalho que são executadas nas VMs. Cada carga de trabalho possui as seguintes informações: identificador da carga de trabalho (id), tempo de chegada, tempo total de processamento, número de CPUs utilizadas, quantidade de memória e utilização média das CPUs.

O método *EveryASecond()* da classe *Users* verifica a cada segundo de tempo simulado quais cargas de trabalho chegaram no centro de dados durante o último segundo simulado e devem ser processadas. Para isso, ele compara o tempo de início da carga de trabalho com o *clock* do simulador. Após selecionadas, as cargas são inseridas em uma fila gerenciada pela classe *JobQueue*. O SimDC3D cria uma VM, representada pela classe *VirtualMachine*, para cada processador que a carga de trabalho exige. As VMs são então inseridas em uma fila (*qWaitingVMs*) para serem escalonadas pelo algoritmo de escalonamento definido pelo parâmetro de simulação *SCHEDULING_ALGORITHM*.

A utilização de CPU que cada carga de trabalho gera em uma VM pode ser estática ou dinâmica e pode ser configurada pelo parâmetro *DYNAMIC_WORKLOAD_VMS*. No modo estático, a média de utilização de CPU é extraída do traço e atribuída a uma VM. Nesse modo, a CPU não possui variabilidade de utilização, permanecendo com uma utilização constante. O modo estático é útil para estudos que consideram apenas consumo energético e não se preocupam com aspectos de desempenho, pois uma utilização constante permite que um escalonador aloque cargas até o limite da CPU, mas sem ultrapassá-lo a ponto de gerar violações de SLA.

No modo dinâmico, cada VM que está em execução no centro de dados invoca o Algoritmo 1 a cada segundo para calcular a quantidade de CPU que a carga de trabalho irá exigir naquele instante. Nesse modo, as VMs requisitam para o servidor hospedeiro uma utilização de CPU que varia ao longo do tempo. Um servidor hospedando várias VMs experimenta, portanto, uma variabilidade de CPU conforme a demanda de todas as VMs. Este modo é útil para avaliar as violações de SLA, pois variações na carga de trabalho podem levar a CPU a períodos com utilização de 100%.

Algoritmo 1: ALGORITMO DE CÁLCULO DE VARIABILIDADE DE CPU

Entrada: *cpuLoadRatio*
Saída: Utilização CPU

```

1 início
2    $cpugen = adjustCPUutilization + (cpuUsageURD2(*generator) * cpuLoadRatio * 2);$ 
3   se  $cpugen > 1$  então
4     |  $adjustCPUutilization = cpugen - 1;$ 
5     |  $cpugen = 1;$ 
6   senão
7     |  $adjustCPUutilization = 0;$ 
8   fim
9    $dynamicCPULoadRatio = cpugen;$ 
10 fim
```

O Algoritmo 1 utiliza como parâmetro de entrada a média de utilização de CPU extraída da carga de trabalho e um gerador de números aleatórios com distribuição uniforme. A média é dada pela variável *cpuLoadRatio* e o gerador pela função *cpuUsageURD2(*generator)* (Linha 2). Caso o valor gerado de utilização de CPU ultrapasse 100%, o excedente é armazenado em *adjustCPUutilization* e acrescido ao valor de utilização no próximo segundo simulado. O algoritmo retorna uma utilização de CPU a cada segundo de modo que ao final da simulação a média de utilização se aproxime da média extraída do traço.

4.3.1.3 Topologias de Rede e Matriz de Tráfego

O SimDC3D utiliza o simulador FNSS (SAINO; COCORÁ; PAVLOU, 2013) para gerar os arquivos com a topologia da rede e a matriz de tráfego. O FNSS está dividido em duas partes. A primeira contém o *core* FNSS que gera um cenário de simulação parametrizado e exportável em arquivos Linguagem de Marcação Extensível (*do inglês Extensible Markup Language*) (XML) para serem consumidos por outros simuladores. A segunda é formada por uma Interface de Programação de Aplicações (*do inglês Application Programming Interface*) (API) para a linguagem de programação C++ com classes que são compiladas e usadas pelo SimDC3D. A API disponibiliza métodos para importação dos arquivos gerados pelo *core* FNSS e para manipulação da topologia e da matriz de tráfego contidas nos arquivos importados.

A API do FNSS possui 15 classes e diversos métodos. Alguns desses métodos e classes foram modificados para fazer a integração do FNSS com o SimDC3D. As alterações consistem em novos atributos e métodos para simular o consumo energético, funcionalidades para ligar e desligar os comutadores, novos atributos para armazenar as estatísticas de tráfego dos enlaces e atributos e métodos auxiliares que são necessários para a integração entre os simuladores.

O core FNSS é implementado em *python* e está dividido em 11 módulos, agrupados em três pacotes: *topologies*, *netconfig* e *traffic*, como mostra a Figura 4.3.

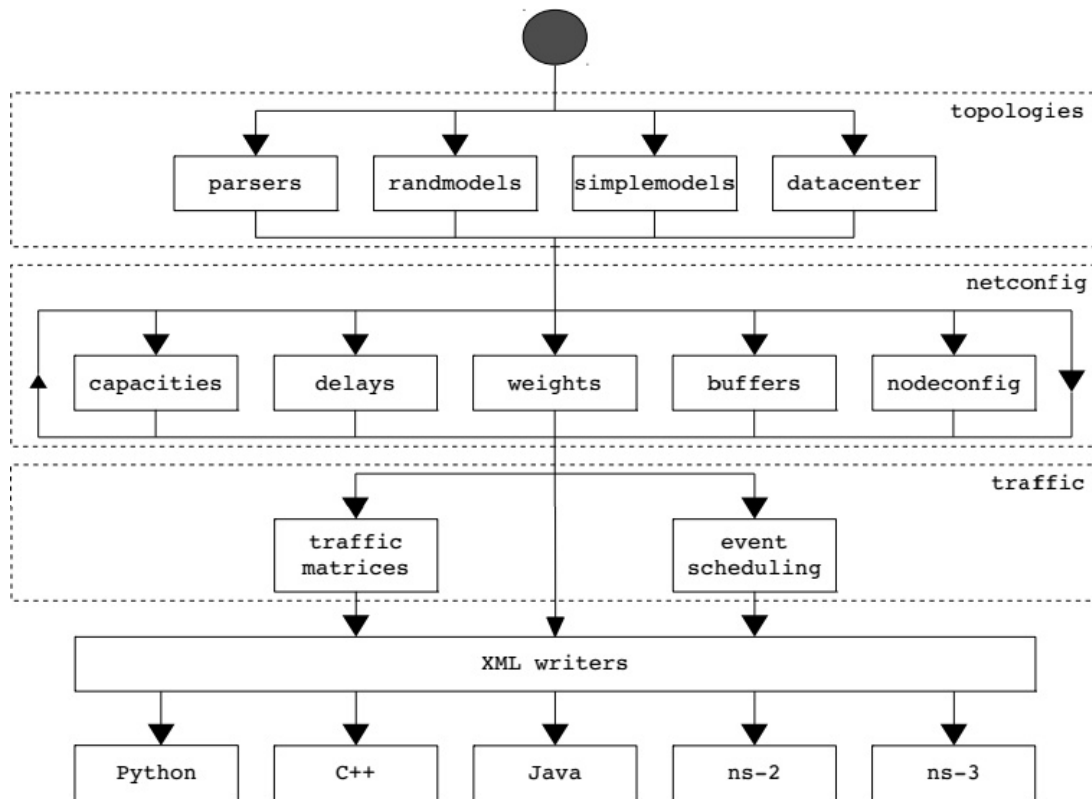


Figura 4.3 – Arquitetura FNSS redesenhada de (SAINO; COCORÁ; PAVLOU, 2013).

O pacote *topologies* contém as funções para gerar três classes de topologias: (i) topologias sintéticas (*Erdős-Rényi*, *Waxman*, *Barabási-Albert*, *Extended Barabási-Albert*, *Generalized Linear Preference*); (ii) topologias para centro de dados (*two e three tier* (BENSON et al., 2010), *fat tree* (AL-FARES; LOUKISSAS; VAHDAT, 2008) e *BCube* (GUO et al., 2009)); e (iii) topologias simples (*k-ary tree*, *dumbbell*, *line*, *star*, *ring* e *full mesh*). A configuração da topologia no arquivo de parâmetros pode ser observada na Linha 11 da Figura 4.4.

O pacote *netconfig* contém as funções para atribuir os parâmetros de configuração para a topologia escolhida, tais como: capacidade (Linhas 27 e 28 da Figura 4.4), custo dos enlaces (Linha 31) e atraso (Linha 34). Além disso, ele permite a parametrização dos tamanhos dos *buffers* dos comutadores, pilha de protocolos e aplicações dos nós.

Por último, o pacote *traffic* contém as funcionalidades para a geração das matrizes de tráfego sintéticas (Linhas 38 a 50 da Figura 4.5), agendamento dos eventos (*e.g.*, falhas de nó ou enlace) e funcionalidades para exportar os arquivos XMLs (Linhas 54 e 58).

O FNSS possibilita a geração de matrizes de tráfego estático ou dinâmico. As matrizes com tráfego estático contém volumes de tráfego em um único ponto no tempo e as matrizes de tráfego dinâmico contém uma sequência de matrizes com volume de tráfego gerado em momentos diferentes dentro de um período de 24 horas. As matrizes de tráfego dinâmico são

```

1  """
2  Datacenter topology / Simulation SimDC3D
3  =====
4  """
5  import fnss
6  import networkx as nx
7  import random
8
9  # create a topology with 4 core switches, 13 edge switches and 40 hosts
10 # per switch (i.e. 520 hosts in total)
11 topology = fnss.two_tier_topology(n_core=4, n_edge=13, n_hosts=40)
12
13 # assign capacities
14 # let's set links connecting servers to edge switches to 1 Gbps
15 # and links connecting core and edge switches to 10 Gbps.
16
17 # get list of core_edge links and edge_leaf links
18 link_types = nx.get_edge_attributes(topology, 'type')
19
20 core_edge_links = [link for link in link_types
21                    if link_types[link] == 'core_edge']
22 edge_leaf_links = [link for link in link_types
23                   if link_types[link] == 'edge_leaf']
24
25 # assign capacities
26
27 fnss.set_capacities_constant(topology, 1000, 'Mbps', edge_leaf_links)
28 fnss.set_capacities_constant(topology, 10000, 'Mbps', core_edge_links)
29
30 # assign weight 1 to all links
31 fnss.set_weights_constant(topology, 1)
32
33 # assign delay of 10 nanoseconds to each link
34 fnss.set_delays_constant(topology, 1, 'ms')

```

Figura 4.4 – Arquivo de configuração do simulador FNSS.

ciclo estacionárias, ou seja, elas simulam um comportamento sazonal dentro de um período de 24 horas.

O modelo matemático que descreve o comportamento das matrizes de tráfego gerado pelo FNSS está descrito em Nucci *et al.* (NUCCI; SRIDHARAN; TAFT, 2005). Uma simulação utilizando matrizes de tráfego dinâmico permite ter uma variabilidade do tráfego ao longo do período simulado. Com isso, é possível simular períodos de pico durante diferentes momentos do dia para cada uma das VMs do centro de dados.

Alguns parâmetros são utilizados para configurar a matriz de tráfego que será utilizada em uma simulação. O parâmetro *NAME_FILE_MATRIX_TRAFFIC* define qual matriz de tráfego deve ser usada, o parâmetro *NUMBER_OF_SAMPLES_TRAFFIC_MATRIX* especifica a quantidade de matrizes de tráfego por dia que o arquivo deve conter e, por último, o parâmetro *NUMBER_OF_PERIODS* determina o total de dias gerados no arquivo.

Os métodos das classes *DataFlow*, *TopologySimDC3D*, *TopologyOptimizationAlgorithms* e *TrafficMatrixSimDC3D* do *SimDC3D* (Figura 4.2) são responsáveis por invocar os métodos das classes do FNSS, implementando, portanto, uma camada de integração entre os dois simuladores.

```

37 # generate cyclostationary traffic matrix (period 7 days, 24 samples per day)
38 tmc = fnss.sin_cyclostationary_traffic_matrix(
39     topology,
40     mean=0.8, # average flow in TM is 0,5 Gbps
41     stddev=0.05, # this is the std among average flows of different OD pairs
42     gamma=0.8, # gamma and log_psi are parameters for fitting the std of
43     log_psi=-0.33, # volume fluctuations over time. Look at Nucci et al. paper
44     delta=0.2, # traffic variation from period max and avg as fraction of average
45     n=48, # number of samples per each period
46     periods=1, # number of periods
47     max_u=0.9, # max link utilization desired
48     origin_nodes=None, # Specify origin and destination nodes. If None,
49     destination_nodes=None # all nodes of the topology are both
50 ) # origin and destination nodes of traffic
51 # origin and destination nodes of traffic
52
53 # save traffic matrices on files
54 fnss.write_traffic_matrix(tmc, 'c:\Matriz\cyclostationary48.xml')
55
56
57 # save topology to a file
58 fnss.write_topology(topology, 'c:\Matriz\datacenter_topology.xml')

```

Figura 4.5 – Configurações para a matriz de tráfego a ser gerada pelo FNSS.

4.3.2 Fluxo de Execução e Modelagem de Simulação com o SimDC3D

O SimDC3D é um simulador altamente parametrizável e flexível que permite configurações de vários cenários de simulação. Esta seção descreve os principais parâmetros do simulador e como eles são usados para modelar diferentes simulações. O Apêndice B detalha todos os parâmetros do arquivo de configuração principal do SimDC3D.

4.3.2.1 Centro de Dados

O núcleo central do simulador é executado na classe *DataCenter* pelo método *EveryASecond()*. Ele, inicialmente, verifica se há cargas de trabalho para serem processadas, isto é, compara o *clock* do simulador com os tempos de chegada das cargas de trabalho no centro de dados. Cada carga de trabalho é alocada em uma ou mais máquinas virtuais de acordo com o número de processadores extraído do traço. As máquinas virtuais são, então, inseridas na fila de escalonamento. Em seguida, o controlador central é invocado e executa algoritmo de escalonamento definido no parâmetro *SCHEDULING_ALGORITHM*. O SimDC3D fornece diversas informações para os escalonadores, tais como: estatísticas dos recursos dos servidores, informações de temperatura e estatísticas de tráfego. Atualmente, o SimDC3D disponibiliza várias estratégias de escalonamento que serão discutidas na Seção 5.1. Ademais, é possível especializar a classe *SchedulingAlgorithm* para criar novas estratégias de escalonamento de acordo com as políticas de economia de energia que se deseja implementar.

Após alocar as cargas de trabalho e escalonar as VMs nos servidores, o controlador coleta informações dos servidores a procura de oportunidades de otimização e na resolução de problemas. Uma realocação de máquina virtual ocorre nos casos abaixo de acordo com limiares definidos no arquivo de configuração:

1. no momento que o controlador detecta que os limites superior (sobrecarga) ou inferior (subutilizado) de uso de CPU são ultrapassados. Esses valores são definidos pelos parâmetros *THRESHOLD_TOP_USE_OF_CPU* e *THRESHOLD_BOTTOM_USE_OF_CPU*.
2. no momento que a temperatura de emergência for ultrapassada em um servidor. A temperatura de emergência é definida pelo parâmetro *EMERGENCY_TEMPERATURE*.
3. no momento que os enlaces entre os servidores e os comutadores de acesso estiverem congestionados (e.g., > 80% da capacidade do enlace). O valor do percentual é definido no parâmetro *TRAFFIC_LIMIT_FOR_ADAPTATION*.
4. quando houver uma alta correlação de tráfego entre duas VMs e elas estiverem em servidores distintos e em *racks* diferentes. O cálculo de correlação é habilitado pelo parâmetro *CALCULATE_TRAFFIC_CORRELATION*.
5. quando o algoritmo de escalonamento estiver utilizando um preditor e este indicar que ocorrerá sobrecarga de utilização da CPU em qualquer um dos servidores ativos do centro de dados. A utilização de predição de CPU é definida pelo parâmetro *USING_PREDICTION_TO_OVERLOAD* e o parâmetro *PREDICTION_ALGORITHM_OVERLOAD* define o algoritmo a ser utilizado.

Se algum desses casos ocorrer, o SimDC3D seleciona as VMs a serem migradas de acordo com uma ordenação baseada em um *ranking* que combina diferentes variáveis, como utilização de CPU e memória. O parâmetro *VMS_ALGORITHM_SELECTION* especifica o método de ordenação utilizado. Atualmente, o SimDC3D disponibiliza os seguintes métodos de ordenação:

1. *higher_utilization_average_CPU/less_memory*: maior utilização média de CPU e menor uso de memória;
2. *higher_utilization_CPU/less_memory*: maior utilização de CPU no instante t e menor uso de memória;
3. *higher_utilization_average_CPU/less_memory/higher_IO_network*: maior utilização média de CPU, menor uso de memória e maior tráfego de rede;
4. *higher_utilization_average_CPU*: maior utilização média de CPU;
5. *less_memory/higher_IO_network*: menor uso de memória e maior tráfego de rede;
6. *random_selection*: escolha aleatória.

Após definidas as VMs a serem migradas, é necessário escolher os servidores que as receberão. O SimDC3D permite que seja utilizada uma única estratégia para todos os casos de realocação de máquina virtual ou uma estratégia específica para cada caso. Por exemplo,

as máquinas virtuais que estão sendo realocadas por sobrecarga de CPU podem utilizar um algoritmo de seleção de servidores diferente das que estão sendo realocadas por sobrecarga de temperatura ou por enlaces congestionados. Os parâmetros a seguir determinam qual algoritmo é utilizado para selecionar um servidor destino em cada um dos casos de realocação de máquina virtual citados anteriormente.

- *OPTIMIZATION_ALGORITHM_OVERLOAD_UTILIZATION*: especifica o algoritmo de seleção de servidores para as máquinas virtuais que estão migrando por sobrecarga de CPU ou por indicação do preditor de CPU.
- *OPTIMIZATION_ALGORITHM_LOW_UTILIZATION*: determina o algoritmo de seleção para as VMs que estão migrando de servidores com baixo processamento (subutilizado).
- *OPTIMIZATION_ALGORITHM_LINK_OVERLOAD*: indica o algoritmo de seleção para as VMs que estão migrando de servidores com enlaces congestionados.
- *OPTIMIZATION_ALGORITHM_OVERHEATING_TEMPERATURE*: define o algoritmo de seleção para as VMs que estão migrando de servidores com sobrecarga térmica.
- *OPTIMIZATION_ALGORITHM_CORRELATION*: estipula o algoritmo de seleção para as VMs que estão migrando devido a uma alta correlação do tráfego entre duas ou mais VMs.

A especialização da classe abstrata *ServerOptimizationAlgorithms* permite estender as estratégias existentes ou desenvolver um novo algoritmo de seleção. O fluxo de monitoramento, seleção e migração das VMs no SimDC3D está ilustrado na Figura 4.6.

O SimDC3D disponibiliza preditores para estimar o comportamento futuro de algumas variáveis do servidor, como utilização de CPU, memória, temperatura, tráfego ou qualquer outra variável que possa ser utilizada pelos algoritmos de escalonamento e migração de máquinas virtuais. Novos algoritmos de predição podem ser incorporados e utilizados pelo controlador do centro de dados por meio de especialização da classe *PredictionAlgorithms*. Os algoritmos de predição podem auxiliar, por exemplo, na prevenção de sobrecarga de CPU. Neste caso, o controlador pode reduzir antecipadamente a quantidade de máquinas virtuais em um servidor com previsão de sobrecarga. A utilização desses algoritmos é importante na prevenção de quebras dos acordos de nível de serviço (SLAs) do centro de dados, como será mostrado no Capítulo 6.

Após concluir o escalonamento e a realocação de máquinas virtuais, o método *EveryASecond()* da classe *DataCenter* executa o método *EveryASecond()* da classe *Server* para cada servidor ativo do centro de dados. Este método simula diversos subsistemas dos servidores. Inicialmente, ele invoca o método para recalcular a nova temperatura do servidor, uma vez que todas as máquinas virtuais já foram escalonadas ou realocadas no centro de dados. Após o cálculo, diversas estatísticas do servidor são atualizadas e o preditor de CPU é invocado para calcular a utilização de CPU futura.

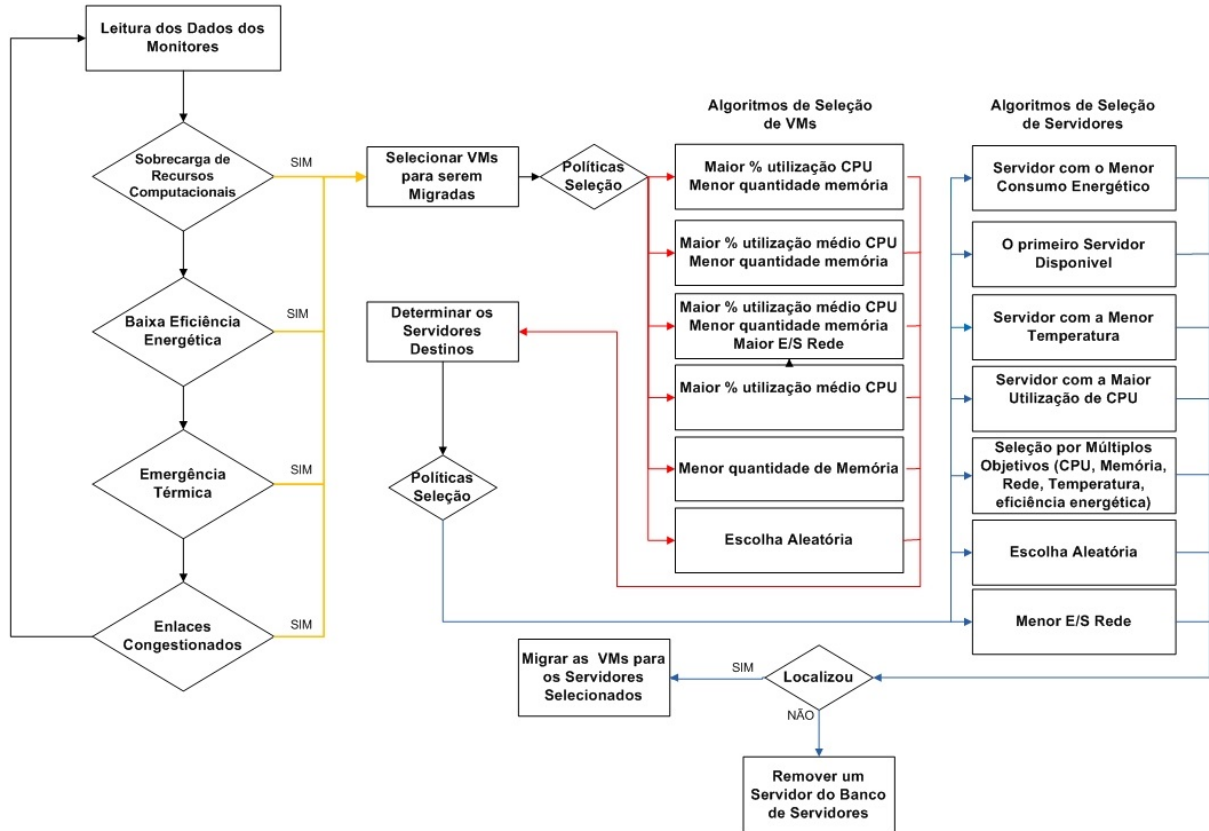


Figura 4.6 – Diagrama do fluxo de controle das migrações de VMs no SimDC3D.

Em seguida, o método *EveryASecond()* de cada máquina virtual (classe *VirtualMachine*) alocado no servidor é executado. Esse método simula a execução da carga de trabalho na VM. Nessa etapa, podem ocorrer dois casos. No primeiro, o servidor tem capacidade de processamento suficiente para atender todas as requisições das máquinas virtuais por CPU. Neste caso, não será calculado nenhum atraso no processamento solicitado pelas VMs. No segundo, o servidor não tem capacidade de processamento suficiente. Nesta situação, será inserido um atraso no processamento da carga de trabalho de cada máquina virtual alocada no servidor. Uma máquina virtual levará mais tempo para executar uma determinada carga de trabalho quando alocada por muito tempo em servidores sobrecarregados.

Quando uma VM termina a execução de sua carga de trabalho, ela é removida do servidor e a utilização da CPU e o consumo energético do servidor são recalculados.

O último subsistema a ser invocado pelo método *EveryASecond()* da classe *DataCenter* é o do banco de servidores. Neste caso, o método *EveryASecond()* da classe *PoolServer* é invocado para gerenciar as funcionalidades do banco de servidores e realizar as mudanças de estado dos servidores quando necessárias.

4.3.2.2 Banco de Servidores

O objetivo do banco de servidores é manter um conjunto de servidores ligados para atender rapidamente às demandas de escalonamento ou migração de VMs no momento em que não há

nenhum servidor disponível para atendê-las. O SimDC3D utiliza o parâmetro *POOL_SIZE* para configurar o banco de servidores. Se ele for igual zero, a simulação não utiliza o banco. Caso contrário, o parâmetro determina a quantidade de servidores que pertencem ao banco.

No SimDC3D, o banco de servidores é representado pela classe *PoolServers*. Essa classe disponibiliza um conjunto de métodos que gerenciam a inserção, remoção e substituição de servidores no banco. Além disso, o módulo de gerenciamento do banco de servidores é responsável por coordenar as mudanças de estados (desligado, inicializando, ligado, hibernando ou saindo de hibernação) dos servidores. A Figura 4.7 descreve a transição entre os estados de um servidor.

Os parâmetros *HIBERNATING_POWER_ON*, *HIBERNATING_POWER_OFF*, *POWER_OFF* e *POWER_ON* determinam quanto tempo duram as mudanças de estado correspondentes. Servidores no estado inicializando, desligado, entrando ou saindo de hibernação não podem receber VMs. Um servidor está apto a receber uma VM quando está no estado ligado e não pertence ao banco de servidores. O único estado em que não há consumo de energia é no estado desligado.

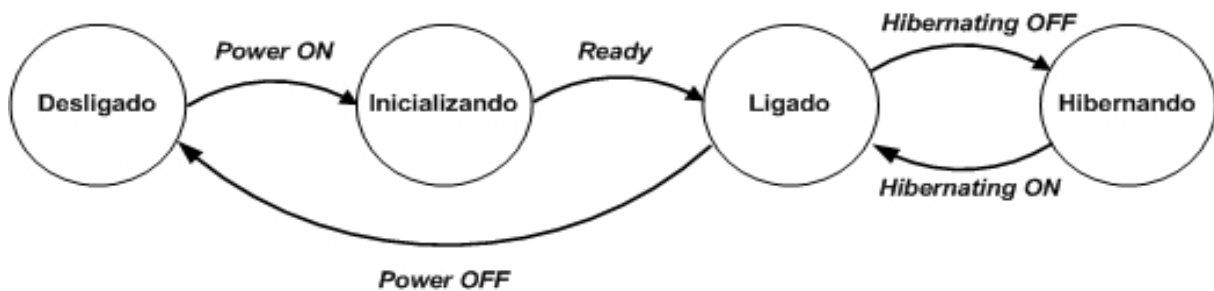


Figura 4.7 – Diagrama de transições dos servidores no SimDC3D.

O SimDC3D disponibiliza dois modos de operação para otimizar os servidores que foram selecionados para serem desligados. No primeiro modo, todos os servidores são desligados e não há consumo de energia. Já no segundo, os servidores selecionados são colocados em modo de hibernação e há um pequeno consumo de energia por servidor. O parâmetro *HIBERNATING_SERVER* define se os servidores são desligados ou ficam hibernando quando são tirados de operação e o parâmetro *POWER_CONSUMPTION_HIBERNATING* define o consumo de energia de cada servidor no modo de hibernação.

O método *EveryASecond()* da classe *PoolServer* verifica, a cada segundo, a quantidade de servidores disponíveis no banco. Caso algum servidor seja utilizado, outro é, imediatamente, ligado ou acordado para substituir o servidor removido do banco. Esse método também é responsável em gerenciar as transições de estado dos servidores, controlando o tempo para executar cada transição (inicializando, ligado, desligado, hibernando ou saindo de hibernação).

Ao finalizar a execução desse método, o método *EveryASecond()* da classe *DataCenter* invoca o método *PoliceTemperature()* da classe *CRAC* para ajustar a temperatura do sistema de refrigeração do centro de dados.

4.3.2.3 Sistema de Refrigeração

O ciclo de refrigeração em um centro de dados funciona da seguinte maneira. O sistema de refrigeração opera por meio da extração de calor do centro de dados, bombeando ar frio na sala, geralmente por intermédio de uma câmara de pressão sob um piso pressurizado. A pressão força o ar frio para cima por intermédio de um sistema de ventilação que entra na sala em frente aos *racks*. O sistema de refrigeração do servidor puxa o ar frio para dentro do servidor, refrigerando-o e empurrando o ar quente para fora por meio de aberturas existentes na parte traseira do servidor. O ar quente sobe com a ajuda de ventiladores ou por meio de dutos que o direcionam para a câmara de pressão presente no teto, sugando o ar de volta para o sistema de refrigeração. Os aparelhos de ar condicionado forçam o ar quente a passar por um sistema de refrigeração, que o esfria novamente. Após esse processo, o ar frio novamente é empurrado para a câmara de pressão no piso, reiniciando o ciclo de refrigeração.

A eficiência de um sistema de refrigeração depende da troca de calor realizada pelo sistema de refrigeração, cujo trabalho é mover o calor a partir de um local (dentro da sala) para outro (fora da sala). As bombas de calor são responsáveis por mover o ar frio/quente e são classificadas por um coeficiente de desempenho (*COP*) dado pela Equação (4.1):

$$COP = \frac{HeatMoved}{WorkDone} = \frac{Q}{W} \quad (4.1)$$

$$W = \frac{Q}{COP} \quad (4.2)$$

Em que:

Q é a proporção do calor retirado;

W é a quantidade de trabalho necessária para remover o calor (W).

O trabalho necessário para remover o calor é inversamente proporcional ao *COP*, como mostra a Equação (4.2). De acordo com Moore *et al.* (MOORE *et al.*, 2005), um *COP* elevado indica um processo mais eficiente, que requer menos trabalho para remover uma quantidade constante de calor. Por exemplo, um ciclo de refrigeração com um *COP* de dois irá consumir 50 kW para remover 100 kW de calor, ao passo que um ciclo com um *COP* de cinco consumirá 20 kW para remover 100 kW de calor.

Durante um ciclo de resfriamento, o *COP* não é constante. Para obter economia nos custos de refrigeração, é essencial aumentar a temperatura de alimentação na câmara de pressão, movendo o sistema de refrigeração para uma faixa de operação mais eficiente. Entretanto, tal faixa não pode extrapolar o limiar térmico do servidor.

O *COP* utilizado no SimDC3D é dado pela Equação 4.3. Essa equação foi inicialmente descrita em (MOORE *et al.*, 2005) e foi definida a partir de um ar condicionado de água gelada do centro de dados da HP.

$$COP = 0.0068T_{Supply\ ar}^2 + 0.0008T_{Supply\ ar} + 0.458 \quad (4.3)$$

Para ajustar a temperatura do centro de dados, o SimDC3D disponibiliza dois modos de controle do sistema de refrigeração. No primeiro, uma temperatura é definida para o ar condicionado pelo parâmetro *CRAC_SUPPLY_TEMP_IS_CONSTANT_AT* e não é alterada durante toda a simulação. No segundo, o controle de temperatura do ar condicionado é dinâmico e políticas de controle de temperatura podem ser implementadas especializando o método *PoliceTemperature()* da classe *CRAC* como será discutido na Seção 5.3. Por exemplo, o algoritmo *Agressivo* funciona da seguinte maneira. O ar condicionado é inicializado a uma temperatura de 10°C e aumenta em $0,1^{\circ}\text{C}$ a cada segundo até que a maior temperatura de um servidor seja de 28°C . A partir daí, o algoritmo aumenta a temperatura em $0,01^{\circ}\text{C}$ até que haja algum servidor com temperatura igual a $29,5^{\circ}\text{C}$. Desse ponto em diante, o algoritmo não aumenta mais a temperatura. Caso algum servidor atinja a temperatura de emergência (*EMERGENCY_TEMPERATURE*), o algoritmo começa a diminuir a temperatura do ar condicionado em $0,2^{\circ}\text{C}$ até que todos os servidores possuam temperaturas abaixo da temperatura de emergência. Esse algoritmo reduz em mais de 20% o consumo energético do sistema de refrigeração em comparação com o sistema de refrigeração fornecendo ar a uma temperatura constante de 19°C , como será discutido no Capítulo 6.

Para se calcular a temperatura de um servidor, é necessário analisar o fluxo de calor dentro do centro de dados e o consumo energético dos diversos dispositivos existentes. O fluxo de calor é incorporado no simulador por meio de uma matriz de recirculação de calor (HRM). A HRM é construída usando algoritmos de dinâmica de fluidos computacionais que modelam o comportamento dos fluxos de ar dentro do centro de dados (GILBERT, 2012).

Segundo Yeo *et al.* (YEO; LEE, 2012), uma HRM converte o calor gerado por um servidor específico em um aumento na temperatura de todos os outros servidores devido a recirculação de calor. Por exemplo, se o centro de dados possuir 50 servidores, o tamanho de uma HRM é de 50×50 . A primeira linha da matriz é usada para calcular o quanto a temperatura de entrada de um servidor é afetada pelo calor gerado por todos os 50 servidores. A multiplicação da primeira linha da matriz pelo consumo energético de todos os servidores produzirá a temperatura de entrada do primeiro servidor (YEO; LEE, 2012). Ou seja, cada posição da matriz (i, j) indica a contribuição do servidor j para o aumento da temperatura do servidor i . Além disso, a temperatura de cada servidor varia de acordo com a sua localização no centro de dados devido à recirculação de calor. A HRM leva em consideração a distribuição dos servidores no centro de dados e o consumo de energia (em Watts) em diferença de temperatura (em $^{\circ}\text{C}$) entre um servidor e os outros servidores.

4.3.2.4 Infraestrutura de Rede

A simulação da infraestrutura de rede no SimDC3D é configurada com até quatro parâmetros. Primeiro, o parâmetro *SIMULATES_NETWORK* especifica se a infraestrutura de rede deve ser simulada. Caso ele seja *FALSE*, o simulador desconsidera todos os demais parâmetros e não apresenta qualquer resultado dessa dimensão. Segundo, o parâmetro

SIMULATE_TRAFFIC_MATRIX determina se uma matriz de tráfego será utilizada para simular a comunicação entre as VMs. Se esse parâmetro for *FALSE* e o parâmetro *SIMULATES_NETWORK* for *TRUE*, o SimDC3D simula uma topologia de rede, porém não haverá tráfego entre as VMs e alguns recursos serão desabilitados, como adaptação das taxas de enlaces e otimização da topologia. Nesse caso, o SimDC3D calcula somente o consumo energético dos comutadores. Por último, o parâmetro *SIMULATE_TOPOLOGY_OPTIMIZATION* define se o SimDC3D deve executar ou não algum algoritmo de otimização da topologia de rede. Caso ele seja *TRUE*, o parâmetro *OPTIMIZATION_ALGORITHM_TOPOLOGY* especifica qual algoritmo deve ser utilizado.

Os parâmetros *TYPE_TOPOLOGY* e *NAME_FILE_TOPOLOGY* definem a topologia a ser simulada. O primeiro parâmetro define o tipo da topologia. O SimDC3D disponibiliza atualmente quatro topologias de redes: *two e three tier*, *fat tree* e *BCube*. Essas topologias são parametrizadas no simulador *core* FNSS e importadas no SimDC3D como descrito na Seção 4.3.1.3. O segundo parâmetro especifica o nome do arquivo que contém a topologia gerada pelo *core* FNSS. O construtor da classe *TopologySimDC3D* invoca os métodos do FNSS para importar e parametrizar a topologia de rede. Usualmente, centro de dados utiliza a topologia de três camadas, como já discutido no Capítulo 2 e mostrado na Figura 2.2. Essa topologia permite que técnicas como a de conexão seletiva sejam aplicadas para reduzir o consumo energético e torná-lo proporcional ao volume de tráfego.

Para simular a infraestrutura de rede, o método *EveryASecond()* da classe *DataCenter* invoca sequencialmente três métodos da classe *TopologySimDC3D* e um da classe *TopologyOptimizationAlgorithms*. Este último método é especializado para implementar diferentes algoritmos de otimização.

O primeiro método invocado (*EveryASecond_P1()*) verifica a cada segundo se algum enlace está congestionado (tráfego > 80% da capacidade do enlace). Caso haja algum, o controlador verifica se existem enlaces redundantes desligados e os ativa para aliviar ou eliminar o congestionamento se possível. Em topologias em que não há enlaces redundantes, o controlador tenta migrar máquinas virtuais para diminuir o tráfego e eliminar focos de sobrecarga como descrito na Seção 4.3.2.1. Outra tarefa desse método é desligar comutadores que estejam com todas as portas desativadas.

O segundo método invocado (*EveryASecond_P2()*) é responsável pelo roteamento, controle do tráfego entre as VMs e atualização das estatísticas de tráfego nos enlaces. Para simular o tráfego no SimDC3D, é necessário fornecer uma matriz de tráfego e definir um método para calcular as rajadas de transmissão. O SimDC3D utiliza matrizes de tráfego geradas sinteticamente pelo simulador *core* FNSS, discutido na Seção 4.3.1.3.

Para relacionar uma VM com um nó na matriz de tráfego, o SimDC3D atribui um *ID* escolhido aleatoriamente para cada VM que chega ao centro de dados e vincula esse *ID* a um nó da matriz de tráfego. Com isso, as transmissões de dados da VM para outras VMs dentro do centro de dados são ditadas pelos volumes de tráfego definidos na matriz. Além disso, a VM pode transmitir parte do seu tráfego para fora do centro de dados.

Os parâmetros de simulação *PROPORTION_OF_EXTERNAL_COMMUNICATION* e *PROPORTION_OF_INTERNAL_COMMUNICATION_BETWEEN_VMS* definem os percentuais de tráfego externo (enviado para fora do centro de dados) e interno (enviado para outras VMs), respectivamente. Quando uma VM é selecionada para comunicar-se dentro do centro de dados, uma outra VM é selecionada aleatoriamente como destino, dentro de todas as VMs ativas. Uma vez estabelecida uma relação entre duas VMs, o próximo passo é criar um fluxo de dados entre elas. Esse fluxo é implementado pela classe *DataFlow*.

Se a VM for selecionada para se comunicar com nós fora do centro de dados, um dos comutadores de núcleo é selecionado aleatoriamente para ser o destino da comunicação entre a VM e o exterior do centro de dados. Neste caso, um fluxo de dados é estabelecido entre a VM e o comutador de núcleo selecionado.

O tráfego de rede entre os nós ocorre em rajadas, com períodos de transmissão e silêncio. Para calcular os períodos de transmissão (*ON*) e de silêncio (*OFF*), o SimDC3D modela os períodos de (*ON*) e (*OFF*) de acordo com os resultados descritos em (BENSON et al., 2010) que mostram que esses períodos podem ser descritos por uma distribuição log-normal. Os parâmetros da distribuição (média e variância) são extraídos da matriz de tráfego gerada pelo FNSS e os períodos são calculados usando o Algoritmo 2.

A variável *mean* na Linha 2 é passada como parâmetro e contém a média extraída da matriz de tráfego entre os dois nós. Nas Linhas 7 e 8, o gerador de números aleatórios com uma distribuição log-normal é inicializado (*lognormal_distribution*) para os períodos de *ON* e *OFF*. Os métodos *Period_ON()* e *Period_OFF()*, vistos nas Linhas 12 e 15, retornam os números aleatórios gerados para cada fluxo de dados simulado. A variável *nextPeriod_OFF* é inicializada no construtor da classe *DataFlow* de cada VM como *false* e a variável *linkSpeed* contém a menor velocidade dos enlaces entre os dois nós.

Esse algoritmo permite distribuir o volume de tráfego entre dois nós extraídos da matriz de tráfego ao longo dos períodos de *ON* de cada VM. Uma VM pode ter *n* fluxos de dados com outras VMs ou com o mundo exterior, como mostra a Figura 4.8. Cada fluxo tem o seu comportamento modelado conforme descrito no Algoritmo 2. O total de tráfego que um servidor produz é a somatória do tráfego que cada VM gera. Quando uma VM é migrada de um servidor para outro, os seus fluxos de dados também são migrados.

O terceiro método invocado da classe *TopologySimDC3D* (*EveryASecond_P3()*) executa as funcionalidades de adaptação dinâmica das velocidades dos enlaces de acordo com o tráfego e calcula o consumo energético dos comutadores.

O parâmetro *DYNAMIC_SPEED_ADAPTATION* determina se o controlador muda as velocidades das portas dos comutadores de acordo com o volume de tráfego do enlace. A Equação 4.4 descreve o cálculo do valor de referência usado para a comparação com o tráfego atual. Esse valor é utilizado como referencia para determinar se a velocidade deve ser alterada ou não. A variável *TRAFFIC_LIMIT_FOR_ADAPTATION* é um parâmetro definido no arquivo de configuração. As mudanças de velocidade podem ser para cima ou para baixo da velocidade

Algoritmo 2: ALGORITMO DE CÁLCULO DOS PERÍODOS ON/OFF DE TRANSMISSÃO DOS ENLACES

Entrada: *mean*
Saída: Tráfego entre as VMs

```

1 início
2   on_mean = mean;
3   off_mean = 1. - on_mean;
4   on_mean2 = on_mean * on_mean;
5   off_mean2 = off_mean * off_mean;
6   k2 = k * k;
7   on_distribution = new lognormal_distribution < double >
   (log(on_mean2/(k2 + 1))/2., sqrt(log(k2 + 1)));
8   off_distribution = new lognormal_distribution < double >
   (log(off_mean2/(k2 + 1))/2., sqrt(log(k2 + 1)));
9   enquanto (true) faça
10    se ((period_OFF + period_ON) < 1) então
11     se (nextPeriod_OFF) então
12      | period_OFF+ = Period_OFF();
13      | nextPeriod_OFF = false;
14     senão
15      | period_ON+ = Period_ON();
16      | nextPeriod_OFF = true;
17     fim
18    senão
19     se (nextPeriod_OFF) então
20      | period_OFF = 0;
21      se ((1 - period_ON) >= 0) então
22       | flowMBPS = period_ON * linkSpeed;
23       | period_ON = 0;
24      senão
25       | flowMBPS = 1 * linkSpeed;
26       | period_ON = period_ON - 1;
27      fim
28      senão
29       se ((1 - period_OFF) >= 0) então
30        | period_OFF = 1 - period_OFF;
31        senão
32         | period_OFF = period_OFF - 1;
33        fim
34        flowMBPS = period_ON * linkSpeed;
35        period_ON = 0;
36      fim
37    fim
38  fim
39 fim
  
```

atual. O parâmetro *RUNTIME_LINK_ADAPTATION* determina a frequência com que o controlador verifica se os enlaces podem mudar as velocidades de transmissão seguindo a sequência das taxas de transmissão do padrão *Ethernet* (10/100/1.000/10.000 Mbit/s). O tráfego é comparado com o valor de referência, que é uma porcentagem da velocidade imediatamente anterior em relação à velocidade atual seguindo a sequência de velocidades do protocolo *Ethernet*. Caso o tráfego atual seja menor que o valor de referência, a velocidade é alterada para baixo.

$$RV = (TRAFFIC_LIMIT_FOR_ADAPTATION/100) * PreviousSpeed \quad (4.4)$$

Em que:

TRAFFIC_LIMIT_FOR_ADAPTATION: valor de referência definido no arquivo de configuração;

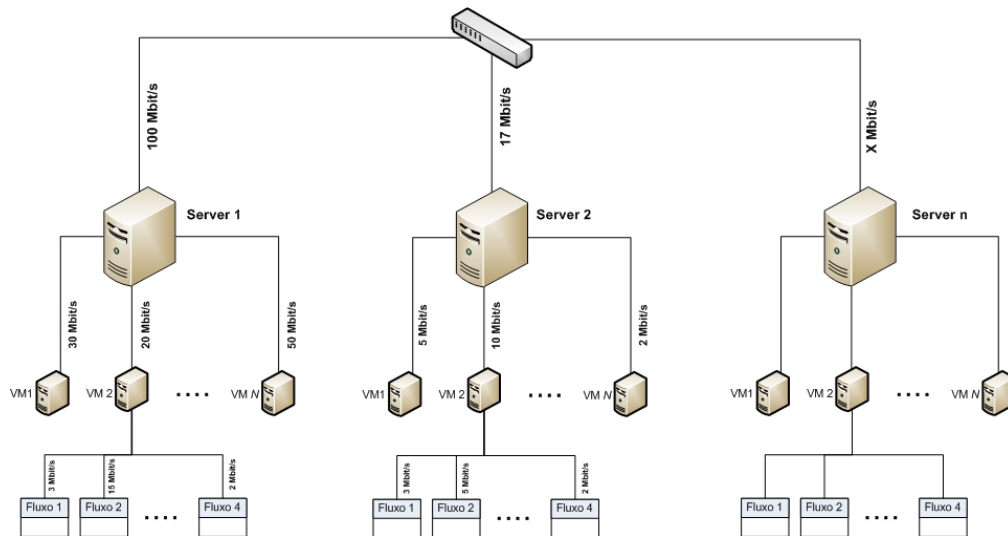


Figura 4.8 – Modelo dos fluxos de dados do SimDC3D.

PreviousSpeed: velocidade anterior à velocidade atual do enlace seguindo as velocidades do protocolo *Ethernet*.

Para exemplificar a redução de velocidade do enlace, suponha que o valor utilizado no parâmetro *TRAFFIC_LIMIT_FOR_ADAPTATION* seja 80, a velocidade atual do enlace seja de 1.000 Mbit/s e o tráfego total no instante T seja de 70 Mbit/s. O valor de referência é calculado como mostra a Equação 4.5:

$$RV = (80/100) * 100$$

$$RV = 80 \quad (4.5)$$

Em que:

80 é o valor extraído do parâmetro *TRAFFIC_LIMIT_FOR_ADAPTATION*;

100 é a velocidade anterior em relação a velocidade atual (1000 Mbit/s) definido no protocolo *Ethernet*.

Como o tráfego do enlace no instante T é menor que o valor de referência, o controlador altera a velocidade da porta do comutador para 100 Mbit/s. O simulador executa esse procedimento conforme o tempo estabelecido no parâmetro *RUNTIME_LINK_ADAPTATION*.

Para dissipar rapidamente congestionamentos devido a mudanças das velocidades das portas, o controlador verifica a cada segundo se o tráfego atual é superior à velocidade de referência. Caso isso seja verdadeiro, o controlador aumenta a velocidade da porta do comutador para uma velocidade superior em relação à atual, obedecendo a sequência de velocidade do padrão *Ethernet* até o limite máximo determinado na configuração da porta.

Para se calcular a energia consumida pela infraestrutura de rede, é necessário informar no arquivo de parâmetros o consumo energético do chassi, *linecard* e das portas nas velocidades

10, 100, 1.000 e 10.000 Mbps para cada tipo de comutador (acesso, agregação e núcleo). O consumo energético de um comutador é calculado pela Equação (4.6).

$$S_{\text{comutador}} = \begin{cases} C_{\text{chassi}} + L_{\text{LineCard}} + \sum_{l=1}^P C_l^{\text{porta}}, & \text{se o comutador está ligado} \\ 0, & \text{se o comutador está desligado} \end{cases} \quad (4.6)$$

A energia total do comutador é a soma da energia consumida pelo *chassi* e pelo *linecard*, que são constantes, mais a energia consumida por todas as portas ativas do comutador, que são dependentes das velocidades. As portas dos comutadores conectadas aos servidores que estão desligados ou em hibernação não consomem energia.

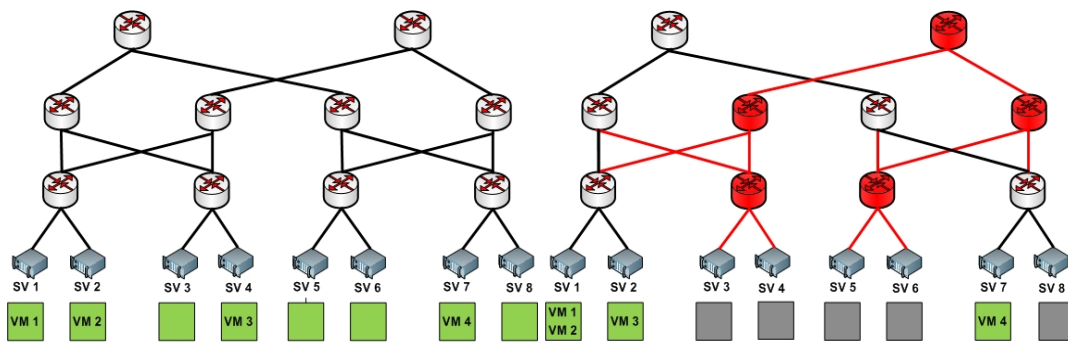
A última etapa da simulação da infraestrutura de rede realiza o cálculo da correlação de tráfego entre as VMs e executa o algoritmo de otimização da topologia. A correlação só é calculada se o parâmetro *CALCULATE_TRAFFIC_CORRELATION* for *TRUE*. Uma vez habilitado, é possível otimizar os servidores com as informações da correlação como descrito na Seção 4.3.2.1.

A otimização da topologia só é possível em centro de dados que provê alta disponibilidade na infraestrutura de rede. Operadores de centro de dados implementam redundância nos enlaces e em comutadores. Como consequência, a infraestrutura de rede é superdimensionada e opera abaixo de sua capacidade máxima a maior parte do tempo. Benson *et al.* (BENSON *et al.*, 2010) mostram, por meio de análises de traços de centros de dados em produção, que a utilização média dos enlaces na camada de agregação é de apenas 8% durante 95% do tempo, enquanto que a utilização dos enlaces na camada de acesso e na camada de núcleo são de aproximadamente 20% e 40%, respectivamente.

O SimDC3D permite testar políticas que se beneficiam dos períodos de baixa utilização para obter economias de energia na infraestrutura de rede. O objetivo dessas políticas é agrupar máquinas virtuais de tal modo que seja possível desligar a maior quantidade de comutadores e enlaces obtendo economias significativas nessa dimensão. Entretanto, esse processo de otimização é dependente de uma matriz de tráfego, que é um requisito necessário para simular o tráfego entre as máquinas virtuais.

A Figura 4.9(a) mostra um exemplo de otimização para os servidores físicos **SV-1**, **SV-2**, **SV-4** e **SV-7**, e as máquinas virtuais, *V1*, *V2*, *V3* e *V4*, respectivamente. Todos os servidores físicos estão conectados a comutadores de acesso. A matriz de tráfego indica que o volume de tráfego trocado por *V1* com *V2* e *V3* é alto e que a rede pode ser otimizada se elas ficarem no mesmo servidor ou em servidores conectados ao mesmo comutador de acesso. No processo de otimização, os requisitos de *hardware* das máquinas virtuais *V1* e *V2* permitem que elas sejam consolidadas no servidor **SV-1**, realizando apenas uma transferência de máquina virtual.

A máquina virtual *V3* é alocada no servidor físico **SV-2**, por ter disponibilidade de *hardware*, e principalmente por conectar *V1* e *V3* no mesmo comutador de acesso. Os requisitos de *hardware* não permitem a alocação da máquina virtual *V4* no servidor físico **SV-2**.



(a) Exemplo de um estudo de caso não otimizado. (b) Exemplo de um estudo de caso otimizado.

Figura 4.9 – Estudo de caso de otimização na infraestrutura de rede.

Ao final, todos os comutadores e enlaces que não fazem parte da comunicação são desligados, como mostra a Figura 4.9(b).

O SimDC3D possibilita simulações que usam técnicas de conexão seletiva. Para tal, é necessário implementar dois algoritmos. O primeiro se encarrega de otimizar as VMs e servidores com ênfase na otimização da rede. Essa política é implementada especializando a classe abstrata *ServerOptimizationAlgorithms*. O segundo se encarrega de otimizar a topologia de rede especializando a classe abstrata *TopologyOptimizationAlgorithms*.

A otimização de topologia consiste basicamente em alocar o tráfego da rede na menor quantidade possível de enlaces e comutadores e desativar temporariamente as redundâncias implementadas na topologia. Em topologias quando há mais de um caminho entre as camadas, o SimDC3D disponibiliza o parâmetro *EXECUTE_LOAD_BALANCING*, que habilita ou desabilita o balanceamento de carga entre as camadas. Quando o balanceamento de carga está ativo, os fluxos de comunicação são divididos igualmente entre todos os caminhos existente entre as camadas.

Os algoritmos de otimização interagem com a topologia por meio de métodos da classe *TopologySimDC3D*, que disponibilizam as seguintes funcionalidades: desligar e ligar comutadores, desativar e ativar portas, acessar as configurações de cada nó na topologia e obter a capacidade do enlace e a largura de banda utilizada.

Por último, o método *EveryASecond()* da classe *DataCenter* coleta informações de todo o centro de dados para gerar as estatísticas da simulação da infraestrutura de rede.

4.4 Estatísticas de Desempenho Calculadas pelo SimDC3D

As informações do centro de dados são coletadas a cada segundo e são utilizadas para calcular as estatísticas de migração, consumo energético, violações de SLA, temperatura e cargas de trabalho. Esta seção apresenta uma visão geral dos principais resultados e o Apêndice E lista todos os resultados gerados pelo SimDC3D.

Os resultados de uma simulação estão divididos nas seguintes seções:

1. **Traço:** apresenta as informações de tempo médio, variância e desvio padrão dos intervalos de tempo de chegada das cargas de trabalho e os seus respectivos tempos de execução. Também são calculados o número máximo de VMs que foram executadas simultaneamente para atender as demandas das carga de trabalho. E, por fim, são apresentadas as estatísticas de tempo de permanência das VMs na fila de escalonamento (média, variância e desvio padrão), como mostra a Figura 4.10(a).
2. **Preditores:** exibe estatísticas e as margens de erro dos algoritmos de predição. Nessa seção, são calculados o erro médio, a raiz do erro quadrático médio (RMSE), variância, desvio padrão e coeficiente de variação dos valores preditos em relação ao real de cada servidor. A Seção 6.4.6 discute resultados de simulação dos preditores.
3. **Políticas de migração:** mostra as informações diárias e consolidadas do número de migrações de cada política descrita na Seção 4.3.2.1. Um resumo dessas estatísticas pode ser visto na Figura 4.10(a). Ao final dessa seção são exibidos o total de migrações que ocorreram por servidor.
4. **Violações de SLA e Downtime:** reporta resultados de SLA e *Downtime* de cada servidor no centro de dados, como ilustrado na Figura 4.11(a) e 4.11(b).
5. **Máquinas Virtuais:** apresenta o total de VMs criadas, escalonadas e que finalizaram suas tarefas no centro de dados.
6. **Consumo energético e indicadores de desempenho:** exibe as médias e o total da energia consumida para cada uma das dimensões simuladas, como mostram as Figuras 4.10(b), 4.12(b). Também são calculados os indicadores *Power Usage Effectiveness* (PUE), para medir a eficiência energética do centro de dados, juntamente com *Total Facility Power* (tPUE). Esta seção também reporta a utilização média total do centro de dados a cada 24 horas, como pode ser visto na Figura 4.13
7. **Temperatura:** mostra informações da temperatura média dos chassis, como pode ser visto na Figura 4.12(a). Também são exibidos os tempos em segundos em que cada chassi e servidores ficaram acima da temperatura máxima permitida. Esta temperatura é definida no parâmetro *EMERGENCY_TEMPERATURE*.
8. **Tráfego:** reporta o total, a média, o maior e menor pico de utilização do tráfego em cada enlace do centro de dados.

O restante desta seção descreve os cálculos de algumas estatísticas que são reportadas no arquivo de resultados.

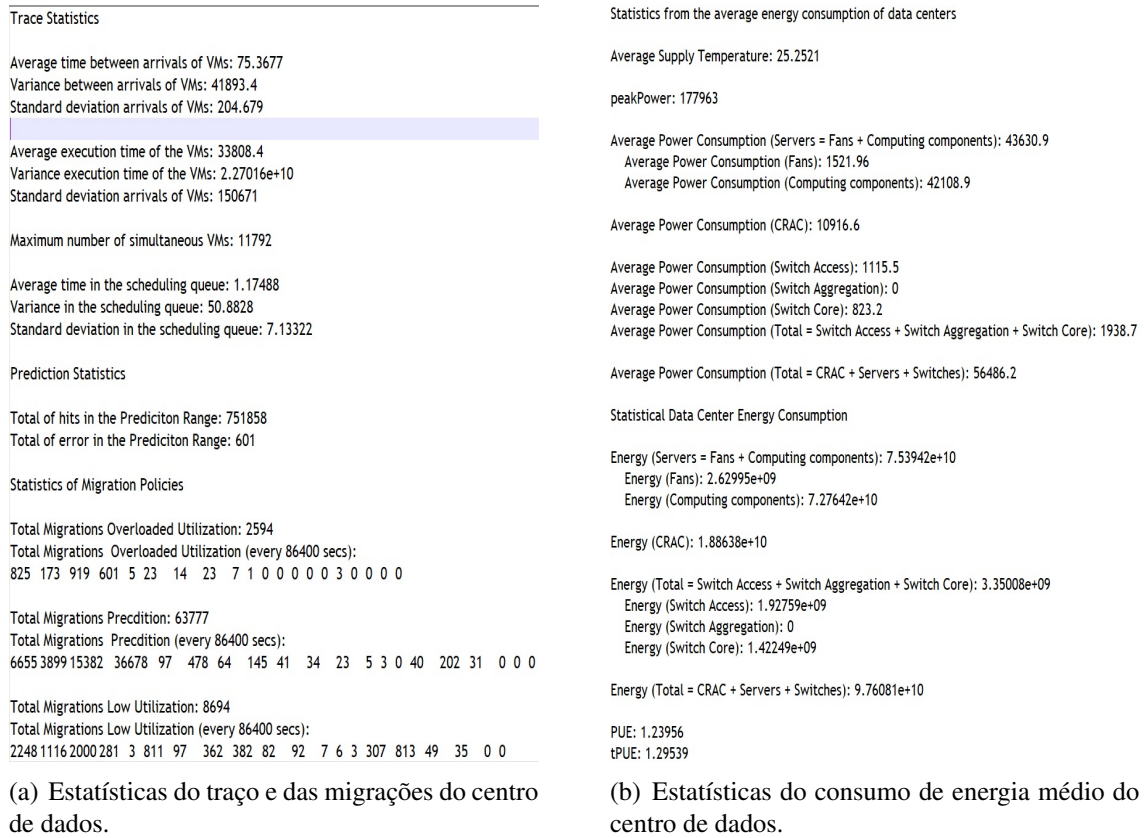


Figura 4.10 – Estatísticas extraídas de simulações do SimDC3D.

4.4.1 Acordos de Nível de Serviço

Em centro de dados virtualizados, atender aos requisitos de QoS é extremamente importante, uma vez que quebras em acordos de nível de serviço geram prejuízos financeiros para os operadores de centro de dados. Segundo Beloglazov *et al.* (BELOGLAZOV; BUYYA, 2012), os requisitos de QoS são normalmente descritos na forma de SLAs. Os SLAs podem descrever os rendimentos mínimos ou os tempos máximos de resposta no centro de dados. O SimDC3D utiliza duas métricas para medir violações de SLA. A primeira, proposta por Beloglazov *et al.* (BELOGLAZOV; BUYYA, 2012), calcula a porcentagem de tempo na qual servidores ativos experimentaram a utilização de CPU em 100%, como mostra a Equação 4.7. A segunda, proposta por Luvcanin *et al.* (LUČANIN *et al.*, 2014), mede a degradação do desempenho de uma VM durante o processo de migração, como mostra a Equação 4.8.

$$SLAT = \frac{1}{N} \sum_{i=1}^N \frac{T_{s_i}}{T_{a_i}}, \quad (4.7)$$

Em que:

N : é a quantidade total de servidores;

T_{s_i} : é o tempo total durante o qual o servidor i experimentou a utilização de 100% levando a um violação de SLA;

SLA Violation

```

Chassi 14 0.000407787 0.000625467 0.00111069 0.00280856 0.00196283 0.00121054 0.00138625 0.000881447 0.000775685 0.000190873
Chassi 15 0.000145296 0.00159711 0.000660468 0.00153303 0.000125226 3.61858e-05 0.00021724 0.00048477 6.26383e-05 0.00019151
Chassi 16 0 3.60101e-06 0.00028343 0.000496079 0.00146253 0.00248629 0.000570946 0.00084779 3.89311e-06 0.000551574
Chassi 17 0.000819179 0.00131288 0 0.000358499 0.00206544 0.00080587 0 0.000773802 0.00110787 0.000270837
Chassi 18 6.36019e-06 0 0.000982595 0.00123593 0.000379102 0.000682656 0.000649825 9.55357e-05 0.00237426 0.000459192
Chassi 19 0.000322061 0.0011864 0.000774638 0.00060935 0.000519915 0.000424024 0.00211692 0.0012256 0.000387388 0.000458725
Chassi 20 0.000253715 0.00131987 0 0.0020159 0.0020881 0 0.000560446 0.0011276 0.000776438 0.00070854
Chassi 21 0.000406009 0.000238958 0.00067434 3.91927e-05 0.00119038 0.000502302 0.000165973 8.57769e-06 0 0.000361031
Chassi 22 0.000577419 0.000267585 0.00122475 0.000221611 0.00193464 0.000489343 0.000663173 0.00286655 0.00146121 0.00276834
Chassi 23 0.000357996 0.000470388 0.000127401 0.000374549 0.000642714 0.000289959 0.000228089 0.000862023 0.00102597 0.000455587
Chassi 24 0.000778274 0.00065851 0.000193002 0.000986575 0.000308585 0.000372765 0.00112691 0.000218706 0.00177214 0.000255376
Chassi 25 3.34304e-06 0 0.00199286 0.000123092 0.000892174 0.000974461 0.000367536 0.00098145 0.00048783 0.000319867
Chassi 26 8.11254e-05 0.0019479 3.39132e-05 9.91658e-05 0.000352347 0.000635037 0 1.83523e-05 0 0.000446443
Chassi 27 0 7.29974e-05 0.000165988 0.00120542 0.00123313 0.000397334 0.000734707 0.000927861 1.16786e-05 1.21806e-05
Chassi 28 0.000757568 0.00057194 0.000780375 8.60382e-06 0 0.000531806 0.00202528 0.000167785 2.86053e-05
Chassi 29 0.0013931 0.000170758 9.86547e-05 0.00160736 0.000601551 0.000673862 0.000472736 0.000637585 0.00105478 0.00172776
Chassi 30 0.000100431 0.000116326 0.000457914 0.000316298 0.000454394 0.000211282 0.000389956 0.00017611 0.000470948 0
Chassi 31 6.57872e-06 0.00132065 0.000169172 0.000360958 0.000202954 0 0 9.18415e-07 0 0.000347047
Chassi 32 0.000300405 0.000229177 0.00015727 0 0 0.000153285 0.000210056 0.000284397 0 0.000562562
Chassi 33 0 4.34715e-05 0.000209275 0.000971515 2.58815e-05 0.00017117 0.000516539 0.000945297 0.00130583 0.000661887
Chassi 34 0.000431717 0.000818404 0.000618921 0.000394193 0.000247246 0.000561759 0.000596855 0.000732331 0.000788122 0.000957926
Chassi 35 0.000291485 0.000144194 0.000802103 0.000169371 0.000838311 1.35103e-05 0.000747388 0.00140224 0 3.88547e-05
Chassi 36 1.18997e-05 0.000268723 0.000411589 0.00022655 0 0.000802358 0.000371538 0 2.09153e-06
Chassi 37 0 0 0.00027645 0.000620938 2.82229e-05 0.00044992 0.000504548 0 0.000549498 0
Chassi 38 0.000193605 0.000238604 0.000397576 1.79036e-05 0.000202111 0.000163084 0.00222534 0.000401654 0.000483205 0.00043407
Chassi 39 0.00074818 0.00173972 0.000587949 5.11274e-05 0.00134216 0.000148285 0.000334849 0.000410521 0.000490167 0.00252567
Chassi 40 0.00279799 0 6.18747e-06 0.00367295 0.00226343 1.61613e-05 2.69424e-05 0.0010077 0.000927445 0.000622735
Chassi 41 0.00390297 0.00142715 0 0.0011595 0.000630655 0.00141196 1.77882e-05 0.000283304 0.00110342 0.00442065
Chassi 42 0.000784273 0.0010455 9.64956e-05 0.000908829 0.000434662 0.000246796 0.000251624 0.00230695 0.00098292 0.00141353
Chassi 43 0.00344648 0 4.36433e-05 0.000121293 0.000414303 0.000393361 0.00131184 0.000903995 0 0.000407735
Chassi 44 0.00138557 0.000803423 0.00132631 0.000678429 0.00135783 0.00365684 0.00156085 0.000684215 0.00327113 0.00220574
Chassi 45 0 0.00187418 0.00140743 0.00220065 0 0 0.00121077 0.00147948 0.00170089 0.000437836
Chassi 46 0.00795666 0.00110492 0.00500261 0.0100268 0.000662254 0.00267594 0.000528718 0.000458265 0.000494708 4.41703e-05
Chassi 47 9.52707e-05 0.000166381 0.000227941 0.00116396 0.000524684 0 9.40947e-07 0.000586562 0.00732087 0.000607397
Chassi 48 0 0.00155768 0.0011945 0.000200714 0.000290667 0.000407861 0.00034837 6.25021e-05 0.000305394 0.000788224
Chassi 49 0.00186462 0.000237668 0.000630264 0.00148562 0.000300778 0.000817303 0.00164933 0.00134573 0.000827977 0.00127899
    
```

SLAT 0.154354

(a) Estatísticas das violações de SLA por servidor.

VM DownTime

```

Chassi 16 99.9455 99.9828 99.9833 99.9527 99.9252 99.9692 99.9831 99.9939 99.9494 99.953
Chassi 17 99.9676 99.9587 99.9823 99.9785 99.9682 99.985 99.9536 99.9402 99.9567 99.991
Chassi 18 99.9749 99.9635 99.9604 99.9795 99.9669 99.9843 99.9874 99.9543 99.9856 99.9439
Chassi 19 99.9955 99.907 99.9861 99.967 99.9661 99.9917 99.9768 99.992 99.9805 99.987
Chassi 20 99.9842 99.9684 99.956 99.9713 99.9324 99.9841 99.9906 99.8498 99.9877 99.9721
Chassi 21 99.9798 99.9942 99.968 99.9662 99.9714 99.9372 99.9693 99.926 99.9882 99.9909
Chassi 22 99.9949 99.9968 99.983 99.9777 99.9744 99.9846 99.9733 99.9829 99.9434 99.9837
Chassi 23 99.9649 99.9906 99.9784 99.9858 99.9783 99.974 99.9212 99.9806 99.9329 99.9768
Chassi 24 99.9444 99.9768 99.9788 99.9706 99.9918 99.9795 99.9667 99.9881 99.9506 99.9864
Chassi 25 99.9832 99.964 99.9913 99.9855 99.9508 99.8946 99.9935 99.9737 99.9957 99.8902
Chassi 26 99.9576 99.9197 99.987 99.9883 99.9769 99.9484 99.9896 99.9784 99.9352 99.9575
Chassi 27 99.9404 99.9725 99.9815 99.9542 99.9901 99.9854 99.9816 99.954 99.9172 99.9534
Chassi 28 99.9819 99.9869 99.9688 99.9781 99.9335 99.9994 99.9614 99.9789 99.9713 99.9879
Chassi 29 99.9133 99.9936 99.9719 99.9838 99.9622 99.9759 99.9877 99.9906 99.9295 99.9789
Chassi 30 99.985 99.9482 99.9826 99.9866 99.9786 99.9772 99.9829 99.9573 99.9993 99.9563
Chassi 31 99.9623 99.9435 99.8343 99.8512 99.9876 99.9868 99.8965 99.9838 99.9518 99.9739
Chassi 32 99.9904 99.9606 99.9881 99.9542 99.9289 99.9678 99.9945 99.9856 99.9221 99.9528
Chassi 33 99.9927 99.9879 99.9609 99.9442 99.9917 99.9935 99.9888 99.9731 99.9811 99.9794
Chassi 34 99.9753 99.9963 99.9783 99.9904 99.9849 99.9501 99.9778 99.9635 99.9823 99.9469
Chassi 35 99.9961 99.9871 99.9657 99.9789 99.9857 99.9331 99.9918 99.9711 99.9561 99.9799
Chassi 36 99.9353 99.9477 99.9333 99.9909 99.9407 99.9517 99.9517 99.9898 99.9242 99.9246
Chassi 37 99.9752 99.9832 99.8996 99.9839 99.9901 99.9916 99.9337 99.9972 99.9857 99.9401
Chassi 38 99.9828 99.9832 99.9788 99.9831 99.9925 99.9442 99.9686 99.9837 99.9458 99.9432
Chassi 39 99.9268 99.992 99.9879 99.9489 99.9656 99.9692 99.9398 99.9893 99.9645 99.9612
Chassi 40 99.9797 99.9898 99.9543 99.9792 99.9788 99.991 99.9854 99.9795 99.9967 99.965
Chassi 41 99.9873 99.9934 99.9847 99.9832 99.9877 99.9887 99.8839 99.9801 99.9936 99.9097
Chassi 42 99.9849 99.9682 99.9808 99.9082 99.9805 99.9753 99.9842 99.9993 99.9928 99.9434
Chassi 43 99.9896 99.9218 99.9902 99.9462 99.9968 99.951 99.9567 99.9672 99.9992 99.9916
Chassi 44 99.9593 99.9654 99.9805 99.932 99.9865 99.9811 99.9594 99.972 99.9949 99.9766
Chassi 45 99.9955 99.9348 99.9926 99.98 99.9906 99.9524 0.00 99.9748 99.9979 99.9843
Chassi 46 99.9897 99.9661 99.9607 99.9763 99.9365 99.9763 99.9864 99.9805 99.9899 99.9559
Chassi 47 99.9941 99.9661 99.989 99.976 99.9764 99.9731 99.9965 99.9976 99.973 99.9811
Chassi 48 99.9521 99.9949 99.9823 99.9763 99.9782 99.9899 99.9961 99.9832 99.9078 99.9803
Chassi 49 99.9911 99.9953 99.9926 99.9856 99.9683 99.9727 99.9311 99.9736 99.988 99.9833
    
```

Average SLA DownTime per Active Host 99.7735

(b) Estatísticas de Downtime por servidor.

Figura 4.11 – Estatísticas extraídas de simulações do SimDC3D.

```

Per Chassis Avg Inlet Temperature:
25.6252 25.9927 26.4611 28.286 29.5028 25.4631 25.4494 25.6825 26.7346 29.1817
25.4683 25.8579 26.7368 27.6303 28.8322 25.4695 25.5579 26.0066 26.7917 28.6682
25.6421 26.0086 26.6534 28.2728 28.9095 25.6271 25.9115 26.3277 27.6661 28.487
25.4538 25.5591 26.024 26.3536 28.0735 25.4528 25.8009 26.6046 27.2855 28.3765
25.4493 25.4371 25.6302 26.5454 28.4594 25.5978 25.9008 26.2302 27.5351 28.4188

Per Chassis Seconds (avg) Over Emergency Temp:
0 0 0 0 106.7 0 0 0 0 9
0 0 0 0 32.8 0 0 0 0 27.2
0 0 0 0 73.4 0 0 0 0 8
0 0 0 0 0 0 0 0 0 8
0 0 0 0 7 0 0 0 0 18.8

Per Server Seconds Over Emergency Temp:
Chassi 0 0 0 0 0 0 0 0 0 0 0
Chassi 1 0 0 0 0 0 0 0 0 0 0
Chassi 2 0 0 0 0 0 0 0 0 0 0
Chassi 3 0 0 0 0 0 0 0 0 0 0
Chassi 4 138 100 83 155 49 76 126 118 113 109
Chassi 5 0 0 0 0 0 0 0 0 0 0
Chassi 6 0 0 0 0 0 0 0 0 0 0
Chassi 7 0 0 0 0 0 0 0 0 0 0
Chassi 8 0 0 0 0 0 0 0 0 0 0
Chassi 9 11 12 11 11 12 11 11 0 11 0
Chassi 10 0 0 0 0 0 0 0 0 0 0
Chassi 11 0 0 0 0 0 0 0 0 0 0
Chassi 12 0 0 0 0 0 0 0 0 0 0
Chassi 13 0 0 0 0 0 0 0 0 0 0
Chassi 14 33 25 29 39 33 32 25 33 39 40
Chassi 15 0 0 0 0 0 0 0 0 0 0
Chassi 16 0 0 0 0 0 0 0 0 0 0
Chassi 17 0 0 0 0 0 0 0 0 0 0
Chassi 18 0 0 0 0 0 0 0 0 0 0
Chassi 19 22 3 31 33 25 33 40 33 26 26
Chassi 20 0 0 0 0 0 0 0 0 0 0
Chassi 21 0 0 0 0 0 0 0 0 0 0
Chassi 22 0 0 0 0 0 0 0 0 0 0
Chassi 23 0 0 0 0 0 0 0 0 0 0
    
```

(a) Estatísticas de temperatura do centro de dados.

```

Total Power Draw (Computing) (every 86400 secs):
27386.1 49048.1 72143.3 110011 114273 81527 68988.3 65296.8 46660.5 42456.3 36143.3 35135.3 34990.8 34243.5 33221.4 6404.93 4099.87 3702.59 3454.25 3430.83

Total Power Draw (CRAC) (every 86400 secs):
5580.73 10413.3 18690.3 33313.6 36258.5 21880.8 17216.3 16048.2 10286.3 9035.54 7570.49 7333.83 7292.83 7104.15 6879.49 1098.1 651.678 588.456 546.285 542.581

Total Power Draw (Switch Access) (every 86400 secs):
1062.7 1124.69 1188.49 1284.63 1302.6 1241.07 1206.08 1194.39 1133.4 1120.31 1097.31 1094.05 1093.82 1092.01 1088.7 1004.45 997.115 995.254 994.5 994.5

Total Power Draw (Switch Aggregation) (every 86400 secs):
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Total Power Draw (Switch Core) (every 86400 secs):
823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2 823.2

Total Power Draw (Total = Switch Access + Switch Aggregation + Switch Core) (every 86400 secs):
1885.92 1947.89 2011.69 2107.83 2125.8 2064.27 2029.28 2017.59 1956.6 1943.51 1920.51 1917.25 1917.02 1915.21 1911.9 1827.65 1820.32 1818.45 1817.7 1817.7

Total Power Draw (Computing + CRAC + Switches) (every 86400 secs):
34852.8 61409.2 92845.3 145433 152658 105472 88233.9 83362.6 58903.3 53435.3 45634.3 44386.4 44200.6 43262.8 42012.8 9330.68 6571.86 6109.5 5818.24 5791.11

Total Energy Draw (Computing) (every 86400 secs):
2.36616e+09 4.23775e+09 6.23318e+09 9.50498e+09 9.87321e+09 7.04394e+09 5.96059e+09 5.64164e+09 4.03146e+09 3.66822e+09 3.12278e+09 3.03569e+09

Total Energy Draw (CRAC) (every 86400 secs):
4.82175e+08 8.99706e+08 1.61484e+09 2.87829e+09 3.13273e+09 1.8905e+09 1.48749e+09 1.38656e+09 8.88733e+08 7.80671e+08 6.5409e+08 6.33643e+08

Total Energy Draw (Switch Access) (every 86400 secs):
9.18189e+07 9.71732e+07 1.02685e+08 1.10992e+08 1.12545e+08 1.07229e+08 1.04205e+08 1.03196e+08 9.79253e+07 9.67947e+07 9.48072e+07 9.45258e+07

Total Energy Draw (Switch Aggregation) (every 86400 secs):
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Total Energy Draw (Switch Core) (every 86400 secs):
7.11245e+07 7.11245e+07 7.11245e+07 7.11245e+07 7.11245e+07 7.11245e+07 7.11245e+07 7.11245e+07 7.11245e+07 7.11245e+07 7.11245e+07 7.11245e+07

Total Energy Draw (Total = Switch Access + Switch Aggregation + Switch Core) (every 86400 secs):
1.62943e+08 1.68298e+08 1.7381e+08 1.82116e+08 1.83669e+08 1.78353e+08 1.7533e+08 1.7432e+08 1.6905e+08 1.67919e+08 1.65932e+08 1.6565e+08
    
```

(b) Estatísticas do consumo de energia do centro de dados a cada 24 horas.

Figura 4.12 – Estatísticas extraídas de simulações do SimDC3D.

```

Average Utilization (Data Center Level): 0.215035
Utilization of the data center (every 86400 secs):
0.139114 0.26605 0.409001 0.668176 0.685714 0.409819 0.332238 0.312894 0.21891 0.195984 0.17027 0.165238 0.163977 0.158845 0.153685 0.0139653 0.00252609

Total computing power (every 86400 secs):
130.948 226.293 324.44 472.351 500 405.342 351.503 333.529 239.685 219.552 184.163 179.151 178.8 176.01 170.924 41.3041 30.0234 27.1599 26.26

Per Chassis Avg Power:
79.2863 117.35 136.47 104.76 124.766 148.607 152.945 92.9433 112.14 83.0187
135.176 86.9034 77.7302 73.8713 80.7575 133.536 84.5052 56.9678 62.7768 84.6141
52.2241 67.3086 67.8403 81.3281 72.1533 66.25 70.9465 59.3644 62.113 72.0457
122.31 77.3803 70.0093 67.1757 70.8515 105.539 76.6631 59.8144 59.9766 71.3822
82.3316 123.389 61.5871 51.4782 70.5217 55.1291 51.3014 80.7093 76.9697 75.1497

Per Chassis Avg Computing Power (max utilization):
0.343923 0.52226 0.656629 0.495587 0.55467 0.662172 0.70656 0.48128 0.601455 0.383653
0.632971 0.417663 0.374285 0.367212 0.432456 0.621796 0.399884 0.279112 0.336718 0.424965
0.248342 0.363149 0.339419 0.389689 0.35388 0.335855 0.34545 0.284315 0.312385 0.355645
0.571097 0.386412 0.372236 0.355317 0.348544 0.502298 0.380085 0.298839 0.312363 0.383149
0.389947 0.581016 0.300964 0.255544 0.35869 0.280022 0.238902 0.400753 0.389091 0.380584

```

Figura 4.13 – Estatísticas de utilização do centro de dados.

T_{a_i} : é o tempo total de servidores i no estado ativo;

$$T_{down} = \frac{V_{mem} D^n}{R^{n+1}} + T_{resume}, \quad n = \left\lceil \log_{\frac{D}{R}} \frac{V_{thd}}{V_{mem}} \right\rceil \quad (4.8)$$

Em que:

V_{mem} : quantidade total de memória da VM;

D : taxa de memória suja;

R : taxa de transferência da rede;

T_{resume} : tempo necessário para reiniciar uma VM

V_{thd} : limite de terminação da pré-cópia

4.4.2 Métricas de Desempenho do Centro de Dados

Para medir a eficiência energética do centro de dados, o consórcio *The Green Grid* (LLC, 2014) definiu algumas métricas, entre elas o PUE , $DCiE$, TFP e $AnnualCost$.

Os valores de PUE e *Datacenter Infrastructure Efficiency* ($DCiE$) são definidos pelas Equações (4.9) e (4.10). Essas métricas se baseiam na relação entre a energia consumida pela instalação inteira e a energia consumida pelos equipamentos da infraestrutura de Tecnologia da Informação (TI).

$$PUE = \frac{TotalFacilityPower}{ITEquipmentPower} \quad (4.9)$$

$$DCiE = \frac{1}{PUE} = \frac{ITEquipmentPower}{TotalFacilityPower} * 100\% \quad (4.10)$$

A variável “*Total Facility Power*” especifica toda a energia dedicada ao centro de dados e a variável “*IT Equipment Power*” especifica a energia consumida somente pelos equipamentos de TI dentro do centro de dados.

Um $PUE = 1$ indica um centro de dados ideal, no qual toda a energia fornecida pela rede elétrica é dedicada exclusivamente para os equipamentos de TI, não havendo gasto de energia com refrigeração e nem com distribuição de energia. Um $PUE < 1$ indica que há uma geração local de energia com base em fontes térmicas residuais e, por último, um $PUE > 1$ indica que a demanda do centro de dados é maior do que a energia necessária para alimentar os equipamentos de TI. Por exemplo, um $PUE = 2$ indica que a demanda de energia é duas vezes maior do que a energia necessária para alimentar os equipamentos de TI. Neste caso, se os equipamentos de TI (servidores, comutadores) consomem 1.000 watts, outros 1.000 watts são consumidos pelos sistemas de refrigeração e distribuição de energia. Nesse exemplo, o valor de $DCiE$ é de 50% e indica que os equipamentos de TI consomem 50% da energia total no centro de dados. O PUE e o $DCiE$ são valores dinâmicos, que podem mudar com as alterações de carga de trabalho no centro de dados.

O *Total Facility Power* (TFP) inclui toda a energia consumida por todos os equipamentos do centro de dados, como servidores, *storage*, comutadores, estações de trabalho, e mais a energia do sistema de refrigeração, *Uninterruptible Power Supply* (UPS) e iluminação.

A energia total da instalação, no cálculo do PUE , é a soma total da energia do sistema de refrigeração, R , e a carga térmica total, T . Esses valores são expressos em kW como mostra a Equação (4.11):

$$TFP = R + T \quad (4.11)$$

Os custos de operação anual são calculados usando a energia total de instalação (TFP) e o custo da energia elétrica ($Cost$). O custo da eletricidade é dado em unidades de moeda/kW-Hr. Como a energia total da instalação é em Quilowatt (kW), um fator de conversão (CF) é utilizado para converter o custo por hora para o custo por ano como mostra a Equação (4.12):

$$AnnualCost = TFP * Cost * CF \quad (4.12)$$

4.4.3 Rack Cooling Index (RCI)

O índice de refrigeração do *rack* (RCI)¹ é calculado usando a média em graus da temperatura de entrada do *rack*, variando sobre a temperatura recomendada pela *American Society of Heating, Refrigerating and Air Conditioning Engineers* (ASHRAE), entre 18 °C a 27 °C. Os dois índices de refrigeração do *rack* são mostrados nas Equações (4.13) e (4.14), ambos expressos em porcentagens. O RCI_{HI} fornece uma medida da frequência da temperatura de entrada no *rack* dos equipamentos, acima da temperatura máxima recomenda pela ASHRAE. E o RCI_{LO}

¹ É uma marca registrada ANCIS, Inc (www.ancis.us).

fornece uma medida da frequência da temperatura de entrada no *rack* dos equipamentos, abaixo da temperatura mínima recomenda pela ASHRAE.

$$RCI_{HI} = \left\{ 1 - \left\{ \frac{\sum_{i=1}^n (T_i - T_{R_{HI}})}{N(T_{A_{HI}} - T_{R_{HI}})} \right\} \right\} * 100\% \quad (4.13)$$

Em que:

$T_{R_{HI}}$ é a temperatura máxima recomendada pela ASHRAE (27°C);

$T_{A_{HI}}$ é a temperatura máxima permitida pela ASHRAE (32°C);

T_i é a temperatura máxima da entrada no i^{th} *rack*;

n é o número de *racks* com $T_i > T_{R_{HI}}$;

N é o número total de *racks* na amostra.

O índice RCI_{LO} é definido por:

$$RCI_{LO} = \left\{ 1 - \left\{ \frac{\sum_{i=1}^n (T_{R_{LO}} - T_i)}{N(T_{R_{LO}} - T_{A_{LO}})} \right\} \right\} * 100\% \quad (4.14)$$

Em que:

$T_{R_{LO}}$ é a temperatura mínima recomendada pela ASHRAE (18°C);

$T_{A_{LO}}$ é a temperatura mínima permitida pela ASHRAE (15°C);

T_i é a temperatura mínima na entrada no *rack* i^{th} ;

n é o número de *racks* com $T_i < T_{R_{LO}}$;

N é o número total de *racks* na amostra.

O *rack* nunca deve ficar fora da faixa recomendada e o valor ideal deve ser de 100% para ambos os índices. Valores entre 90% e 100% são aceitáveis e são considerados bons, enquanto valores abaixo de 90% são considerados ruins.

4.5 Considerações Finais

Este capítulo descreveu a arquitetura do SimDC3D, suas principais funcionalidades e sua integração com o simulador FNSS. As simulações no SimDC3D são configuradas por meio de um conjunto de parâmetros descritos em um arquivo de configuração, conforme apresentado detalhadamente no Apêndice B. O uso de um grande conjunto de parâmetros permite customizar as simulações sem a necessidade de se recompilar o simulador. Este capítulo apresentou também as principais características simuladas em cada dimensão do centro de dados visando auxiliar a compreensão do funcionamento do SimDC3D e as principais estatísticas reportadas ao final de uma simulação.

É possível concluir que o SimDC3D foi desenvolvido para dois perfis de usuários. O primeiro não precisa modificar o simulador e pode simplesmente parametrizá-lo para executar suas simulações e extrair os resultados. Para esse perfil, o SimDC3D proporciona um grau de aprendizado rápido e é de fácil utilização. O segundo tipo de usuário pode implementar e testar novas estratégias no simulador. Por mais que a estrutura interna do SimDC3D permita

implementá-las rapidamente, o grau de dificuldade para compreender todo o funcionamento do SimDC3D, que possui 33 classes com mais de 600 métodos, é mais elevado do que simplesmente sua utilização.

O SimDC3D é um ferramenta de simulação que pode auxiliar pesquisadores e operadores de centro de dados a testarem diversas estratégias de escalonamento e consolidação de máquinas virtuais, controle do sistema de refrigeração e otimização da topologia de redes. Por meio do SimDC3D é possível extrair resultados que permitem uma análise do impacto das estratégias adotadas em todas as dimensões do centro de dados, diferente de outros simuladores que disponibilizam análises em uma ou, no máximo, em duas dimensões, não relacionando as decisões tomadas em cada dimensão com as demais.

CAPÍTULO 5

OTIMIZAÇÕES EM CENTROS DE DADOS

Os algoritmos de otimização propostos neste trabalho têm por objetivo reduzir o consumo energético do centro de dados e afetar o mínimo possível as quebras de acordos de nível de serviço. É desejável que cada otimização executada em uma dimensão reduza o seu consumo energético e não impacte negativamente nas demais. Os algoritmos de otimização estão divididos de acordo com as dimensões do centro de dados. Mais especificamente, na dimensão servidores, são apresentados algoritmos de escalonamento e de realocação de máquinas virtuais; para o sistema de refrigeração, são apresentados algoritmos de controle de temperatura; e para a infraestrutura de rede, são apresentados algoritmos de otimização de topologia. Este capítulo também apresenta algoritmos propostos em outros trabalhos e que estão implementados no SimDC3D para fins de comparação.

O restante deste capítulo está organizado da seguinte forma. Na Seção 5.1, são apresentados os algoritmos de escalonamento de máquinas virtuais. A Seção 5.2 explica os objetivos e detalha os algoritmos de realocação de máquinas virtuais. Na Seção 5.3, são descritos algoritmos de controle de temperatura do ar condicionado. A Seção 5.4 descreve o algoritmo de otimização da topologia de rede desenvolvido neste trabalho e a Seção 5.5 apresenta as conclusões finais desse capítulo.

5.1 Algoritmos de Escalonamento de Máquinas Virtuais

O SimDC3D disponibiliza 13 algoritmos de escalonamento de máquinas virtuais. Além disso, como discutido no Capítulo 4, ele pode ser estendido para incorporar novas propostas. Escalonadores são componentes chaves em estratégias de economia de energia em centro de dados, pois a alocação de uma máquina virtual em um servidor aumenta a utilização de CPU, o consumo de energia e a temperatura do servidor, além de interferir no seu tráfego de rede. Portanto, um bom algoritmo de escalonamento deve levar em consideração diversos parâmetros do centro de dados antes de decidir em qual servidor alocar uma nova máquina virtual.

O parâmetro *SCHEDULING_ALGORITHM* especifica qual escalonador deve ser utilizado durante a simulação. O SimDC3D disponibiliza atualmente sete algoritmos de escalonamento, sendo que dois (3DMOBFD, 2D) tem quatro variações: 3DMOBFD, 3DMOBFD_PREDICTION, 3DMOBFD_POOL, 3DMOBFD_POOL_AND_PREDICTION, 2D, 2D_PREDICTION, 2D_POOL, 2D_POOL_AND_PREDICTION. Os algoritmos BEST_PERFORMANCE, RANDOM, LOW_TEMP_FIRST, MIN_HR, CENTER_RACK_FIRST foram migrados do SimWare. O restante desta seção detalha esses algoritmos.

5.1.1 Escalonador 3DMOBFD

A alocação de máquinas virtuais em servidores pode ser modelada como um problema de empacotamento com recipientes de diferentes tamanhos e custos em que os recipientes representam os servidores e os pacotes as máquinas virtuais (BELOGLAZOV; BUYYA, 2012). Os tamanhos dos recipientes são as capacidades de CPU, memória e larguras de banda das conexões de rede dos servidores. O Apêndice A apresenta uma formulação matemática desse problema usando programação linear inteira e com uma função objetivo que minimiza o consumo energético. Porém, como é bem conhecido, o problema do empacotamento é NP-Difícil e, normalmente, é resolvido por heurísticas (YUE; ZHANG, 1995). Uma das heurísticas mais simples e eficazes para o problema do empacotamento é a BFD (*Best Fit Decreasing*), que aloca um pacote em um recipiente que possui o menor espaço capaz de acomodar o pacote, ou seja, o que deixa o menor espaço vazio após alocar o pacote (YUE; ZHANG, 1995). Alguns algoritmos usam variantes do BFD para alocação de máquinas virtuais, como, por exemplo, o PABFD (*Power-Aware Best Fit Decreasing*) que aloca uma máquina virtual em um servidor que causará o menor acréscimo no consumo de energia (BELOGLAZOV, 2013).

Seguindo essa mesma filosofia, o algoritmo 3DMOBFD (*3 Dimensional Multiple Objectives Best Fit Decreasing*), mostrado no Algoritmo 3, resolve o problema de alocação de máquinas virtuais, mas usa como critério de alocação uma classificação (*ranking*) dos servidores que leva em consideração a temperatura, a utilização de CPU, o consumo de energia, a quantidade de memória utilizada e o tráfego do servidor. O *ranking* do servidor i , RK_i , é calculado pelas Equações 5.1 a 5.6.

$$RK_i = S_{i_{Temp}} + S_{i_{CPU}} + S_{i_{Power}} + S_{i_{Memory}} + S_{i_{Traffic}} \quad (5.1)$$

onde

$$S_{i_{Temp}} = \alpha * (W_{Temp} * (1 - (\frac{CT_i - LT}{HT - LT})^{e_t})) \quad (5.2)$$

$$S_{i_{CPU}} = \beta * (W_{CPU} * (1 - U_i^{e_{CPU}})) \quad (5.3)$$

$$S_{i_{Power}} = \gamma * ((W_{Power} * (1 - (\frac{CE_i - LE}{HE - LE})^{e_E}))) \quad (5.4)$$

$$S_{i_{Memory}} = \delta * (W_{Memory} * ((\frac{TM_i - CM_i}{TM_i})^{e_M})) \quad (5.5)$$

$$S_{i_{Traffic}} = \epsilon * (W_{Traffic} * (1 - (\frac{Tr_i}{BW_i})^{e_{Tr}})) \quad (5.6)$$

Em que:

CT_i é a temperatura atual do servidor i ,

HT é a maior temperatura permitida,

LT é a menor temperatura permitida,

U_i é a utilização de CPU do servidor i ,

CE_i é o consumo atual de energia do servidor i ,

LE é o menor consumo de energia permitido para um servidor,

HE é o maior consumo de energia permitido para um servidor,

TM_i é a memória total do servidor i ,

CM_i é a memória em uso do servidor i ,

Tr_i é o tráfego atual do servidor i ,

BW_i é a largura de banda da conexão de rede do servidor i .

Os coeficientes $\alpha, \beta, \gamma, \delta$, e ϵ são variáveis binárias e determinam se um parâmetro deve ser usado (igual a um) ou não (igual a zero) no cálculo do *ranking* RK_i . Os coeficientes $W_{Temp}, W_{CPU}, W_{Power}, W_{Memory}$ e $W_{Traffic}$ são parâmetros que determinam o peso de cada variável. Se os parâmetros forem iguais a um, todas as variáveis terão o mesmo peso. Entretanto, se o objetivo for dar pesos diferentes para cada variável, deve-se usar valores menores que um de modo que a somatória de todos os pesos seja igual a um. Os expoentes das equações auxiliares e_t, e_{CPU}, e_E, e_M , e e_{Tr} serão discutidos abaixo.

O Algoritmo 3 considera um centro de dados com n chassis e m servidores por chassi. Ele calcula inicialmente o *ranking* (RK_i) de todos os servidores e em seguida ordena as máquinas virtuais em ordem decrescente de utilização de CPU. A alocação de uma máquina virtual é feita no servidor com o maior *ranking* e que possui recursos suficientes para recebê-la. Um servidor possui capacidade para receber uma máquina virtual se sua utilização de CPU for menor que $THRESHOLD_TOP_USE_OF_CPU$, sua temperatura for menor que $EMERGENCY_TEMPERATURE$, a soma de memória das VMs for menor que sua memória e a soma do tráfego das VMs for menor que 80% da capacidade do seu enlace. Em seguida, o algoritmo recalcula os valores das variáveis utilizadas já considerando a máquina virtual alocada para um servidor.

O *ranking* proposto possui vários atributos desejáveis. Primeiro, o uso de várias variáveis no *ranking* possibilita a construção de combinações para a implementação de diferentes estratégias

Algoritmo 3: ALGORITMO 3DMOBFD - COMPLEXIDADE $O(n^2)$.

```

Entrada: ListaServidores, ListaVMs
Saída: Alocação VM
1 início
2   para n em NUMERO_DE_CHASSI faça
3     para m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
4       |   CalculaRanking(ListaServidores[i][j]);
5       fim
6     fim
7   Ordena.Descendente.Utilizacao.Media.CPU(ListaVMs);
8   para vm em ListaVMs faça
9     Ordena.Descendente.Ranking(ListaServidores);
10    para servidor em ListaServidores faça
11      se servidor tem recursos suficientes para vm então
12        |   Aloca vm no servidor.Selecionado;
13        |   RecalculaRanking(servidor.Selecionado);
14        |   Parar;
15      fim
16    fim
17  fim
18 fim

```

por meio das variáveis binárias $\alpha, \beta, \gamma, \delta, \epsilon$. Segundo, a utilização de pesos permite que uma variável tenha maior ou menor influência no cálculo do *ranking*. Terceiro, a parametrização das potências nas equações auxiliares permite adotar estratégias mais conservadoras (expoente igual a um) ou mais agressivas (expoentes maiores que um) para uma determinada variável, como pode ser visto na Figura 5.1. Por exemplo, um expoente maior que um na equação S_{iTemp} , permite que temperaturas mais elevadas tenham valores próximos de um, conseqüentemente, um servidor que está com uma temperatura elevada, mas ainda inferior ao limiar máximo não será descartado. Por outro lado, um expoente igual a um, a variável S_{iTemp} tende para zero, diminuindo o valor do *ranking* do servidor e, portanto, o escalonador escolheria um servidor mais frio. A Seção 6.4 apresenta resultados obtidos com duas combinações no cálculo do *ranking* do algoritmo 3DMOBFD.

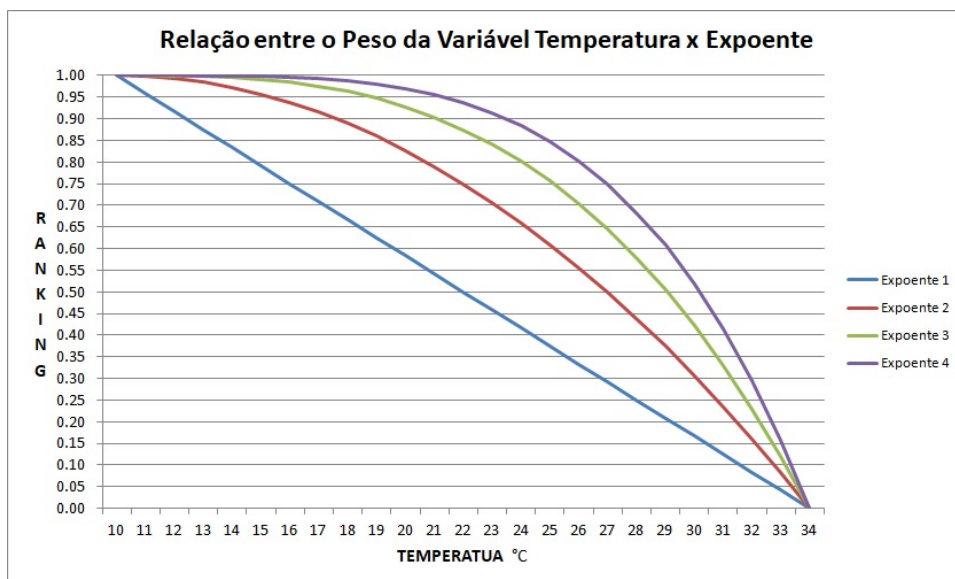


Figura 5.1 – Relação entre expoente e *ranking* da variável temperatura.

5.1.2 Escalonador 3DMOBFD com Preditor

O escalonador 3DMOBFD_PREDICTION não modifica significativamente a lógica do Algoritmo 3. A principal alteração é o uso de predições para o cálculo do *ranking* de um servidor, como mostra o Algoritmo 4. A diferença do 3DMOBFD com predição para o 3DMOBFD está na execução do método *CalculaPredicao()* na Linha 6 do Algoritmo 4 que deve ser inserida após a Linha 3 do Algoritmo 3. Esse método executa o cálculo para prever o comportamento futuro de uma variável, nesse caso a temperatura.

Algoritmo 4: ALGORITMO 3DMOBFD_PREDICTION - COMPLEXIDADE $O(n^4)$.

```

1      .
2      .
3      .
4  para n em NUMERO_DE_CHASSI faça
5      | para m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
6      | | CalculaPredicao(ListaServ[i][j])
7      | | CalculaRanking(ListaServ[i][j]);
8      | fim
9  fim
10     .
11     .
12     .

```

Após executar o método, a variável CT_i é substituída pela variável CT_{Futura_i} , como mostra a Equação 5.7. Essa variável contém o resultado de previsão da temperatura futura do servidor i calculado pelo algoritmo de predição.

$$S_{i_{Temp}} = \alpha * (W_{Temp} * (1 - (\frac{CT_{Futura_i} - LT}{HT - LT})^{et})) \quad (5.7)$$

O cálculo do *ranking*, neste caso, considera a temperatura futura do servidor e não mais a temperatura atual. Porém, diversos fatores ainda interferem no valor final do *ranking*, como o peso de cada uma das variáveis, o avanço no tempo e a precisão do algoritmo de predição. A Seção 6.4.6 avalia a precisão dos algoritmos de predição comparando os valores preditos com os reais.

5.1.3 Escalonador 3DMOBFD com Banco de Servidores

O algoritmo 3DMOBFD_POOL modifica o Algoritmo 3 para utilizar um banco de servidores que ficam disponíveis para alocações quando não há recursos suficientes nos servidores em operação, como mostra o Algoritmo 5. Na versão sem o banco (Algoritmo 3), quando uma máquina virtual não pode ser escalonada em um servidor em operação, ela permanece em uma fila de espera até que haja recursos disponíveis em algum servidor que está em operação ou até que um novo servidor seja ligado, aumentando, portanto, seu tempo de espera para entrada em operação. No Algoritmo 5, quando uma máquina virtual é alocada em um servidor do banco, um novo *ranking* é calculado e o servidor sai do banco e entra em operação. O trecho descrito

no Algoritmo 5 substitui as Linhas de 8 a 17 do Algoritmo 3.

Algoritmo 5: ALGORITMO 3DMOBFD_POOL - COMPLEXIDADE $O(n^2)$..

```

1      .
2      .
3      .
4  para vm em ListaVMs faça
5      Ordena.Descendente.Ranking(Lista.Servidores);
6      escalonado = falso;
7      para servidor em ListaServidores faça
8          se servidor tem recursos suficientes para vm então
9              Aloca vm no servidorSelecionado;
10             RecalculaRanking(servidorSelecionado);
11             escalonado = verdadeiro;
12             Parar;
13         fim
14     fim
15     se não escalonado então
16         remove servidor do banco de servidores;
17         Aloca vm no servidor;
18         RecalculaRanking(servidorSelecionado);
19     fim
20 fim
21     .
22     .
23     .

```

5.1.4 Escalonador 3DMOBFD com Preditor e Banco de Servidores

O escalonador 3DMOBFD_POOL_AND_PREDICTION, descrito no Algoritmo 6, é uma fusão dos escalonadores 3DMOBFD_PREDICTION e 3DMOBFD_POOL. Nesta variação do escalonador, o cálculo do *ranking* é modificado para considerar a temperatura futura. Além disso, o banco de servidores é utilizado quando não é possível alocar uma máquina virtual em um dos servidores em operação. As modificações ocorrem na Linha 4 e nas Linhas 20 a 24 do Algoritmo 6, ou seja, as mudanças devem ser inseridas após a Linha 3 e 16 do Algoritmo 3, respectivamente.

5.1.5 Escalonador 2D

O escalonador 2D (Algoritmo 7) prioriza as variáveis de temperatura e recirculação de calor no centro de dados. A ideia principal do algoritmo é semelhante à do 3DMOBFD, ou seja, os servidores também são selecionados usando um *ranking*. Entretanto, o cálculo do *ranking* é dado pela Equação 5.8 e utiliza variáveis de duas dimensões do centro de dados. As variáveis de peso utilizadas na equação podem ser configuradas por meio dos parâmetros: *SCHEDULER_2D_WEIGHT_TEMPERATURE*, *SCHEDULER_2D_WEIGHT_HEAT_RECIRCULATION* e *SCHEDULER_2D_WEIGHT_LOAD_CPU*.

Algoritmo 6: ALGORITMO 3DMOBFD_POOL_AND_PREDICTION -
 COMPLEXIDADE $O(n^4)$.

Entrada: *ListaServidores, ListaVMs*
Saída: Alocação VM

```

1 início
2   para n em NUMERO_DE_CHASSI faça
3     para m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
4       CalculaPredicao(ListaServ[i][j])
5       CalculaRanking(ListaServ[i][j]);
6     fim
7   fim
8   Ordena.Descendente.Utilizacao.Media.CPU(ListaVMs);
9   para vm em ListaVMs faça
10    Ordena.Descendente.Ranking(ListaServidores);
11    escalonado = falso;
12    para servidor em ListaServidores faça
13      se servidor tem recursos suficientes para vm então
14        Aloca vm no servidorSelecionado;
15        RecalculaRanking(servidorSelecionado);
16        escalonado = verdadeiro;
17        Parar;
18      fim
19    fim
20    se não escalonado então
21      remove servidor do banco de servidores;
22      Aloca vm no servidor;
23      RecalculaRanking(servidorSelecionado);
24    fim
25  fim
26 fim
  
```

$$\begin{aligned}
 RK = & (PesoTemp * (Temperatura/34)) + (PesoRecirc * (RecCalor_{Servidor_i}/0,001)) + \\
 & (PesoCPU * (UtilCPU));
 \end{aligned}
 \tag{5.8}$$

O algoritmo inicia calculando o valor do *ranking* para cada servidor com capacidade para receber uma máquina virtual. Em seguida, os servidores são ordenados ascendentemente pelo *ranking*. Note que este *ranking* é diferente daquele usado pelo 3DMOBFD, pois aqui um valor menor tem prioridade sobre um valor maior. Isso faz sentido, pois o que se busca é um servidor com temperatura e utilização de CPU mais baixas e que cause menos recirculação de calor. O escalonador aloca a VM no primeiro servidor da lista com capacidade para recebê-la. Se o escalonador não encontrar servidor com capacidade suficiente para receber a VM, ela permanece na fila de escalonamento até que algum servidor fique disponível. Os valores 34 e 0,001, da Equação 5.8, são a maior temperatura e o maior índice de recirculação de calor de um servidor.

Algoritmo 7: ALGORITMO 2D - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores*, *ListaVMs*, *HRF*, *pesoTemp*, *pesoHRF*, *pesoCPU*
Saída: Alocação VM

```

1 início
2   para cada n em NUMERO_DE_CHASSI faça
3     para cada m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
4       ListaServidores[n][m].ranking =
5         (pesoTemp * (ListaServidores[n][m].temperatura/34)) + (pesoHRF *
6           (HRF[n]/0.001)) + (pesoCPU * (ListaServidores[n][m].utilizacaoCPU));
7     fim
8   fim
9   Ordenar.ranking(ListaServidores);
10  para cada vm em ListaVMs faça
11    para cada servidor em ListaServidores faça
12      se servidor tem recursos suficientes para vm então
13        Aloca vm no servidor;
14        listaVMs.remove(vm);
15      parar;
16    fim
17  fim

```

5.1.6 Escalonador 2D_PREDICTION

O escalonador 2D_PREDICTION, mostrado no Algoritmo 8, é uma variação do escalonador 2D que inclui predição para estimar a temperatura futura do servidor e a sobrecarga de CPU. A temperatura futura substitui a atual no cálculo do *ranking* (Linha 6) e a predição da sobrecarga restringe a utilização do servidor (Linhas 12 a 14). A predição evita que uma VM recém escalonada gere sobrecarga em um servidor. Caso isso ocorra, o controlador teria que selecionar uma das VMs alocadas e transferi-la para reduzir a utilização de CPU. Este processo aumentaria o número de migrações desnecessariamente. O preditor pode, também, auxiliar na restrição dos servidores caso a temperatura futura prevista seja superior à temperatura de emergência (Linhas 12 a 14). Um bom escalonador deve selecionar um servidor para receber uma VM de forma que ela não seja migrada logo em seguida. A ideia principal do algoritmo 2D é utilizar duas variáveis (CPU e temperatura) no cálculo do *ranking* que são responsáveis por uma grande quantidade de migrações no centro de dados. As modificações ocorrem nas Linhas 4, 5 e nas Linhas 12 a 14 do Algoritmo 8, isto é, as alterações devem ser inseridas após as Linhas 3 e 9 do Algoritmo 7, respectivamente.

5.1.7 Escalonador 2D_POOL

O escalonador 2D_POOL, apresentado no Algoritmo 9, é uma modificação do Algoritmo 7 que inclui o banco de servidores para atender rapidamente por demandas de processamento quando não há servidores com recursos disponíveis (Linhas 18 a 21). O algoritmo inicialmente calcula o *ranking* de todos os servidores ativos (Linhas de 2 a 6) e em seguida ordena do menor para maior *ranking* (Linha 7). As VMs são alocadas nos servidores que possuem o menor *ranking*, ou seja, menor temperatura, recirculação de calor e utilização de CPU no instante T . Caso

Algoritmo 8: ALGORITMO 2D_PREDICTION - COMPLEXIDADE $O(n^4)$.

Entrada: *ListaServidores, ListaVMs, HRF, pesoTemp, pesoHRF, pesoCPU*
Saída: Alocação VM

```

1 início
2   para cada n em NUMERO_DE_CHASSI faça
3     para cada m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
4       ListaServidores[n][m].temperaturaFutura ←
5         preverTemperaturaFutura(ListaServidores[n][m]);
6       ListaServidores[n][m].preveSobrecarga ←
7         preveSobrecargaFutura(ListaServidores[n][m]);
8       ListaServidores[n][m].ranking =
9         (pesoTemp * (ListaServidores[n][m].temperaturaFutura/34)) + (pesoHRF *
10          (HRF[n]/0.001)) + (pesoCPU * (ListaServidores[n][m].utilizacaoCPU));
11      fim
12    fim
13  Ordenar.ranking(ListaServidores);
14  para cada vm em ListaVMs faça
15    para cada servidor em ListaServidores faça
16      se servidor.temperaturaFutura > TEMPERATURA_EMERGENCIA ou
17        servidor.preveSobrecarga então
18        continue;
19      fim
20      se servidor tem recursos suficientes para vm então
21        Aloca vm no servidor;
22        listaVMs.remove(vm);
23      parar;
24    fim
25  fim
26 fim

```

alguma VM não consiga ser alocada em um servidor por falta de recursos, um novo servidor é retirado do banco de servidores e colocado em operação para receber a VM. As modificações, as quais ocorrem nas Linhas de 18 a 21 do Algoritmo 9, devem ser inseridas após a Linha 15, do Algoritmo 7.

5.1.8 Escalonador 2D_POOL_AND_PREDICTION

O escalonador 2D_POOL_AND_PREDICTION, descrito no Algoritmo 10, é uma fusão dos escalonadores 2D_POOL e 2D_PREDICTION. Nas Linhas 4 e 5 são feitas previsões para as variáveis temperatura e utilização de CPU. Na Linha 6 é calculado o *ranking* dos servidores. Após o cálculo ser executado para todos os servidores, é gerada uma lista ordenada (Linha 9). As VMs são alocadas nos servidores que possuem o menor *ranking*. Servidores com a temperatura futura maior que a temperatura de emergência e com indicação de sobrecarga de CPU são desconsiderados (Linhas 13 a 15). Caso alguma VM não consiga ser alocada em um servidor por falta de recursos, um novo servidor é retirado do banco de servidores e colocado em operação para receber a VM. As modificações ocorrem nas Linhas 4 e 5, 13 a 15 e 23 a 26 do Algoritmo 10, isto é, as alterações devem ser inseridas após as Linhas 3, 9 e 15 do Algoritmo 7, respectivamente.

Algoritmo 9: ALGORITMO 2D_POOL - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores*, *ListaVMs*, *HRF*, *pesoTemp*, *pesoHRF*, *pesoCPU*
Saída: Alocação VM

```

1 início
2   para cada n em NUMERO_DE_CHASSI faça
3     para cada m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
4       ListaServidores[n][m].ranking =
5         (pesoTemp * (ListaServidores[n][m].temperatura/34)) + (pesoHRF *
6           (HRF[n]/0.001)) + (pesoCPU * (ListaServidores[n][m].utilizacaoCPU));
7     fim
8   fim
9   Ordenar.ranking(ListaServidores);
10  para cada vm em ListaVMs faça
11    escalonado ← falso;
12    para cada servidor em ListaServidores faça
13      se servidor tem recursos suficientes para vm então
14        Aloca vm no servidor;
15        escalonado ← verdadeiro;
16        listaVMs.remove(vm);
17      parar;
18    fim
19  se !escalonado então
20    ListaServidores ← removeServidordoBancodeServidores();
21    Aloca vm no servidor.ultimo();
22  fim
23 fim

```

5.1.9 Escalonadores do SimWare

Os escalonadores BEST_PERFORMANCE, UNIFORM_TASK, LOW_TEMP_FIRST, RANDOM, MIN_HR e CENTER_RACK_FIRST foram desenvolvidos originalmente para o simulador SimWare, proposto por Yeo *et al.* (YEO; LEE, 2012). Esses escalonadores foram mantidos no SimDC3D com o objetivo de disponibilizar um maior número de estratégias que envolvam diferentes variáveis durante o processo de escalonamento. Eles são descritos neste capítulo para contrastar suas abordagens com às propostas nesta tese. Em particular, os escalonadores originais do SimWare consideram apenas uma ou nenhuma das dimensões de maior consumo para fazer alocação de máquinas virtuais.

5.1.9.1 Escalonador BEST_PERFORMANCE

O escalonador BEST_PERFORMANCE, reproduzido no Algoritmo 11, aloca uma máquina virtual em um servidor com a menor *utilização de CPU*. Em situações de sobrecarga, uma máquina virtual poderá ser alocada em servidores sobrecarregados, não existindo qualquer condição no algoritmo para impedir isso. Outro ponto observado nesse algoritmo é que ele não analisa as temperaturas dos servidores, podendo selecionar um servidor que esteja com sobrecarga térmica.

Algoritmo 10: ALGORITMO 2D_POOL_AND_PREDICTION - COMPLEXIDADE
 $O(n^4)$.

Entrada: *ListaServidores, ListaVMs, HRF, pesoTemp, pesoHRF, pesoCPU*
Saída: Alocação VM

```

1 início
2   para cada n em NUMERO_DE_CHASSI faça
3     para cada m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
4       ListaServidores[n][m].temperaturaFutura ←
5         preverTemperaturaFutura(ListaServidores[n][m]);
6       ListaServidores[n][m].preveSobrecarga ←
7         preveSobrecargaFutura(ListaServidores[n][m]);
8       ListaServidores[n][m].ranking =
9         (pesoTemp * (ListaServidores[n][m].temperaturaFutura/34)) + (pesoHRF * (HRF[n]/0.001)) + (pesoCPU * (ListaServidores[n][m].utilizacaoCPU));
10      fim
11     fim
12   Ordenar.ranking(ListaServidores);
13   para cada vm em ListaVMs faça
14     escalonado ← falso;
15     para cada servidor em ListaServidores faça
16       se servidor.temperaturaFutura > TEMPERATURA_EMERGENCIA ou
17         servidor.preveSobrecarga então
18         continue;
19       fim
20       se servidor tem recursos suficientes para vm então
21         Aloca vm no servidor;
22         escalonado ← verdadeiro;
23         listaVMs.remove(vm);
24         parar;
25       fim
26     fim
27   se !escalonado então
28     ListaServidores ← removeServidoresdoBancodeServidores();
29     Aloca vm no servidor.ultimo();
30   fim
31 fim
  
```

5.1.9.2 Escalonador LOW_TEMP_FIRST

O escalonador LOW_TEMP_FIRST, mostrado no Algoritmo 12, seleciona o servidor que tem a menor temperatura de entrada, como mostra as Linhas 11 a 15, dentre todos os servidores em operação no centro de dados. Servidores sobrecarregados são descartados durante o processo de escolha (Linhas 8 a 10). Esse algoritmo tem como objetivo distribuir as cargas de trabalho inicialmente nas regiões mais frias do centro de dados, gerando economias de energia no sistema de refrigeração com controle dinâmico de temperatura. Ele considera os servidores que estão em locais mais quentes somente quando não há mais opções. Esse algoritmo evita servidores sobrecarregados, porém permite a alocação de VMs em servidores cuja temperatura seja maior que *EMERGENCY_TEMPERATURE*.

5.1.9.3 Escalonador RANDOM

O escalonador RANDOM, mostrado no Algoritmo 13, escolhe aleatoriamente um servidor entre todos os servidores em operação para receber uma VM. Uma distribuição aleatória de

Algoritmo 11: ALGORITMO BEST_PERFORMANCE - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores, ListaVMs*
Saída: Alocação VM

```

1 início
2 enquanto !ListaVMs.vazia() faça
3     melhor ← 2; /* Utilização de CPU é sempre menor ou igual a 1 */
4     para cada n em NUMERO_DE_CHASSI faça
5         para cada m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
6             utilCPU ← ListaServidores[n][m].VMsRequeremCPU();
7             se utilCPU < melhor então
8                 melhor ← utilCPU;
9                 melhorN ← n;
10                melhorM ← m;
11            fim
12        fim
13    fim
14    ListaServidores[melhorN][melhorM].AtribuiUmaVM(listaVMs.inicio());
15    listaVMs.remove(ListaVMs.inicio());
16 fim
17 fim

```

Algoritmo 12: ALGORITMO LOW_TEMP_FIRST - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores, ListaVMs*
Saída: Alocação VM

```

1 início
2 enquanto !ListaVMs.vazia() faça
3     melhor ← 100000;
4     para cada n em NUMERO_DE_CHASSI faça
5         para cada m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
6             temperaturaServidor ← (ListaServidores)[n][m].TemperaturaAtualEntrada();
7             utilCPU ← ListaServidores[i][j].VMsRequeremCPU();
8             se utilCPU = 100 então
9                 continue;
10            fim
11            se temperaturaServidor < melhor então
12                melhor ← temperaturaServidor;
13                melhorN ← n;
14                melhorM ← m;
15            fim
16        fim
17    fim
18    ListaServidores[melhorN][melhorM].AtribuiUmaVM(listaVMs.inicio());
19    listaVMs.remove(ListaVMs.inicio());
20 fim
21 fim

```

VMs normalmente produz uma distribuição balanceada, porém esse algoritmo não leva em consideração a carga de processamento que cada VM gera em um servidor. Portanto, servidores recebendo a mesma quantidade de VMs terão utilizações de CPU, memória e tráfego de rede diferentes entre si. Isso pode levar a sobrecargas de CPU e temperatura, além de provocar congestionamento no enlace do servidor. Não foram implementados requisitos de utilização de CPU, memória, temperatura e tráfego que restrinjam a alocação de uma VM em um servidor, como pode ser visto na Linha 8.

Algoritmo 13: ALGORITMO RANDOM - COMPLEXIDADE $O(n)$.

Entrada: *ListaServidores, ListaVMs*
Saída: Alocação VM

```

1 início
2   enquanto !ListaVMs.vazia() faça
3     melhorN ← sortear()%NUMERO_DE_CHASSI;
4     melhorM ← sortear()%NUMERO_DE_SERVIDORES_NO_CHASSI;
5     se (melhorN < 0) ou (melhorM < 0) ou (melhorN > NUMERO_DE_CHASSI) ou
6       (melhorM > NUMERO_DE_SERVIDORES_NO_CHASSI) então
7       continuar;
8     fim
9     ListaServidores[melhorN][melhorM].AtribuiUmaVM(listaVMs.inicio());
10    listaVMs.remove(ListaVMs.inicio());
11 fim

```

5.1.9.4 Escalonador MIN_HR

O escalonador MIN_HR, mostrado no Algoritmo 14, aloca uma máquina virtual no servidor que provoca a menor recirculação de calor dentre todos os servidores com capacidade para receber uma VM. Esse algoritmo tem como parâmetro de entrada um vetor ordenado (*HRFSorted*) que contém os chassis que produzem a menor recirculação de calor. A lógica ao alocar as VMs em servidores com baixa recirculação de calor (Linhas 9 e 10) é produzir menos calor na saída dos servidores. Consequentemente, haverá menos ar quente recirculando pelo centro de dados, impactando positivamente na temperatura de entrada dos demais servidores. O algoritmo impõe restrições a servidores sobrecarregados durante a busca (Linhas 3 a 13). Caso nenhum servidor não seja selecionado (Linha 15), o algoritmo escolhe aleatoriamente um servidor para receber uma máquina virtual (Linhas 20 a 23), podendo, neste caso, selecionar um servidor com sobrecarga de temperatura ou de CPU, uma vez que não existe restrições após a escolha aleatória.

5.1.9.5 Escalonador CENTER_RACK_FIRST

O escalonador CENTER_RACK_FIRST, descrito no Algoritmo 15, tem como objetivo alocar uma máquina virtual nos servidores que estão no mesmo *rack* e em uma posição na qual a recirculação de calor gerada por cada *rack* tenha um efeito menor sobre os demais. Os *racks* estão dispostos em fileiras, sendo que o corredor frio fica entre duas fileiras de racks, como pode ser visto na Figura 2.1.

A estratégia do algoritmo é alocar as VMs inicialmente nos dois *racks* centrais, alternado entre os chassis de cada *rack*. Uma vez preenchidos os dois *racks* centrais, os próximos serão os vizinhos do lado direito e do lado esquerdo. Novamente, há uma alternância entre todos os chassis de cada *rack*. Por último, serão escolhidos os *racks* das extremidades. A disposição escolhida é baseada em uma análise da recirculação de calor gerada por todos os chassis em um único *rack*. A sequência dos *chassis* é um parâmetro de entrada do algoritmo (*machineNumber*).

Algoritmo 14: ALGORITMO MIN_HR - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores, ListaVMs, HRFSorted*
Saída: Alocação VM

```

1 início
2   enquanto !ListaVMs.vazia() faça
3     para cada n em NUMERO_DE_CHASSI faça
4       para cada m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
5         utilCPU = ListaServidores[HRFSorted[n]][m].VMsRequeremCPU
6         se utilCPU = 100 então
7           continuar;
8         fim
9         melhorN ← HRFSorted[n];
10        melhorM ← m;
11        ir para FIM;
12      fim
13    fim
14    FIM:
15    se melhorN ≠ 0 e melhorM ≠ 0 então
16      ListaServidores[melhorN][melhorM].AtribuiUmaVM(listaVMs.inicio());
17      listaVMs.remove(ListaVMs.inicio());
18    fim
19    senão
20      melhorN ← sortear()%NUMERO_DE_CHASSI;
21      melhorM ← sortear()%NUMERO_DE_SERVIDORES_NO_CHASSI;
22      ListaServidores[melhorN][melhorM].AtribuiUmaVM(listaVMs.inicio());
23      listaVMs.remove(ListaVMs.inicio());
24    fim
25  fim
26 fim

```

Nas Linhas 2 a 4, é calculada a sequência dos chassis e nas Linhas 5 a 16 é feita a alocação de acordo com a ordem estabelecida anteriormente. O algoritmo impõe restrições a servidores sobrecarregados durante a busca (Linhas de 9 a 11), caso nenhum servidor seja selecionado (Linha 18), o algoritmo escolhe aleatoriamente um servidor para receber uma máquina virtual (Linhas 23 a 26), podendo, neste caso, selecionar um servidor com sobrecarga de temperatura ou de CPU, uma vez que não existe restrições após a escolha aleatória.

5.2 Algoritmos de Realocação de Máquinas Virtuais

A realocação e o escalonamento de máquinas virtuais têm como objetivo comum selecionar servidores para alocar as VMs que chegam ou estão sendo migradas no centro de dados. Os dois problemas são NP-Difíceis (JUNG et al., 2010; OKADA et al., 2015). Entretanto, esses dois processos são executados em momentos e com restrições diferentes no SimDC3D. No processo de escalonamento, todos os servidores em operação são considerados para a alocação de uma nova VM. Além disso, as VMs que chegam ao centro de dados não possuem estatísticas de utilização de recursos. Desse modo, o escalonador seleciona um servidor com base em critérios preestabelecidos, por exemplo, como descrito na Seção 5.1.1, mas ele não tem como avaliar o impacto que a VM causará no servidor. A realocação de VMs, por outro lado, é mais restritiva, pois somente os servidores ativos e que não tenham VMs a migrar fazem parte da busca. Além disso, as VMs que estão sendo migradas possuem estatísticas dos recursos

Algoritmo 15: ALGORITMO CENTER_RACK_FIRST - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores, ListaVMs, HRF, machineNumber*
Saída: Alocação VM

```

1 início
2   para cada k em MATRIZ_HR faça
3     | HRF[k] = machineNumber[k] - 1;
4   fim
5   enquanto !ListaVMs.vazia() faça
6     para cada n em NUMERO_DE_CHASSI faça
7       para cada m em NUMERO_DE_SERVIDORES_NO_CHASSI faça
8         | utilCPU = ListaServidores[HRF[n]][m].VMsRequeremCPUEscala() +
9           | (ListaVMs.inicio()).retornarCargaCPU();
10        | se utilCPU >= NUMERO_DE_CORE_EM_UM_SERVIDOR então
11          |   continuar;
12          fim
13          | melhorN ← HRF[n];
14          | melhorM ← j;
15          ir para FIM;
16        fim
17      fim
18      se melhorN ≠ 0 e melhorM ≠ 0 então
19        | ListaServidores[melhorN][melhorM].AtribuiUmaVM(listaVMs.inicio());
20        | listaVMs.remove(ListaVM.inicio());
21      fim
22      senão
23        | melhorN ← sortear()%NUMERO_DE_CHASSI;
24        | melhorM ← sortear()%NUMERO_DE_SERVIDORES_NO_CHASSI;
25        | ListaServidores[melhorN][melhorM].AtribuiUmaVM(listaVMs.inicio());
26        | listaVMs.remove(ListaVMs.inicio());
27      fim
28    fim
29 fim

```

utilizados e possibilita ao escalonador avaliar o impacto que causarão no servidor. Um servidor, neste caso, pode ser desconsiderado se alguma restrição atual ou futura (baseada em predição) for violada.

Uma realocação eficiente tem como objetivo diminuir o número de servidores ativos e não sobrecarregar os demais. Grandes quantidades de economia de energia podem ser obtidas desligando servidores com baixo processamento e movendo as máquinas virtuais para outros servidores que possam receber novas cargas. A desvantagem dessa técnica é a sobrecarga que a consolidação de máquinas virtuais pode causar nos servidores.

Todas as dimensões sofrem algum tipo de instabilidade no processo de consolidação. Mais especificamente, servidores podem ficar sobrecarregados com as demandas exigidas pelas novas máquinas virtuais. Uma concentração maior de processamento em um número menor de servidores gera pontos quentes em determinadas regiões do centro de dados, fazendo com que o sistema de refrigeração reaja ajustando a temperatura para baixo. E, por último, em alguns casos, enlaces podem sofrer sobrecarga de tráfego devido a um número maior de VMs em um número menor de servidores.

Para estabilizar o centro de dados, o controlador central coleta informações dos servidores a procura de oportunidades de otimização e também para resolver problemas momentâneos, como descrito na Seção 4.3.2.1. Caso seja identificado algum desses casos, o controlador identifica

os servidores e seleciona as VMs a serem migradas para outro servidor. Uma vez definidas as VMs a serem migradas, é necessário escolher os servidores que as receberão. O SimDC3D permite que seja utilizada uma única estratégia para todos os casos citados na Seção 4.3.2.1 ou uma estratégia específica para cada caso. Esta flexibilidade permite testar diversas estratégias para avaliar o consumo energético em todas as dimensões, a estabilidade do centro de dados e a disponibilidade dos recursos.

O SimDC3D disponibiliza atualmente cinco algoritmos de realocação de máquinas virtuais, sendo que os três primeiros foram propostos neste trabalho:

1. 3DMOBFD (3 Dimensional Multiple Objectives Best Fit Decreasing);
2. 2D (2 Dimensional) ;
3. HIGH_TEMPERATURE;
4. FFD (The First Fit Decreasing);
5. PABFD (Power Aware Best Fit Decreasing).

Os algoritmos de realocação 3DMOBFD e 2D são propostas que utilizam as mesmas estratégias dos algoritmos de escalonamento com algumas modificações, como pode ser visto nos Algoritmos 16 e 17. O algoritmo HIGH_TEMPERATURE, Algoritmo 18, procura um servidor que tem a menor temperatura no centro de dados. Já as estratégias FFD, Algoritmo 19, e PABFD, Algoritmo 20, foram propostos por Beloglazov (BELOGLAZOV, 2013) e implementados no SimDC3D. O objetivo é testar a eficiência desses algoritmos em um ambiente de simulação 3D em comparação com uma simulação unidimensional ou bidimensional.

5.2.1 Algoritmo 3DMOBFD

Inicialmente, o algoritmo 3DMOBFD, Algoritmo 16, calcula o *ranking* dos servidores ativos de acordo com a Equação 5.1 e que não farão nenhuma migração (Linha 2). Em seguida, as VMs são ordenadas por ordem decrescente de utilização de CPU (Linha 3) e os servidores, por ordem crescente de *Ranking* (Linha 4). O servidor com o maior *ranking* em um determinado instante possui a maior quantidade de variáveis em condições mínimas. Contudo, esse método não garante que o servidor tenha todas as variáveis em condições ideais. O próximo passo do algoritmo é avaliar o impacto que a primeira VM da lista causa no primeiro servidor, analisando a carga de CPU, a memória, a temperatura e o tráfego. Isso é feito nas Linhas 8, 9, 11 e 12, respectivamente. Caso o servidor atenda aos requisitos, a VM é alocada nele (Linha 13). Caso contrário, um servidor é removido do banco de servidores quando habilitado (Linhas de 20 a 24). Se não for localizado nenhum servidor, a VM não é migrada e permanece no servidor até a próxima tentativa.

Algoritmo 16: ALGORITMO 3DMOBFD - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores, ListaVMs*
Saída: Realocação das VM

```

1 início
2   Calcula ranque dos Servidores;
3   Ordena ListaVMs em ordem decrescente de utilização CPU;
4   Ordena ListaServidores em ordem decrescente de ranque;
5   para cada vm em ListaVMs faça
6     para cada servidor em ListaServidores faça
7       escalonado = falso;
8       se servidor.utilizacaoCPU + vm.cargaMediaCPU <
          LIMITE_SUPERIOR_DE_USO_CPU então
9         se servidor.memoria + vm.memoria <= servidor.memoriaTotalServidor então
10        NovaTemp =
11          servidor.calcTempFut(servidor.utilizacaoCPU + vm.cargaMediaCPU);
12        se NovaTemp < TEMPERATURA_EMERGENCIA então
13          se (vm.avr_trafego + servidor.trafego) * 0,80 < capacidade_enlace então
14            Aloca vm no servidor;
15            escalonado = verdadeiro;
16            parar;
17          fim
18        fim
19      fim
20    se não escalonado então
21      remove servidor do banco de servidores;
22      Aloca vm no servidor;
23      RecalculaRanking(servidorSelecionado);
24    fim
25  fim
26 fim

```

5.2.2 Algoritmo 2D

O algoritmo 2D, Algoritmo 17, inicia a sua execução com o cálculo do *ranking* dos servidores ativos usando a Equação 5.8 e que não farão nenhuma migração (Linha 2). Em seguida, as VMs são ordenadas por ordem decrescente de utilização de CPU (Linha 3) e os servidores, por ordem decrescente de *Ranking* (Linha 4). O servidor com o menor *ranking* em um determinado instante possui a maior quantidade de variáveis em condições mínimas. O próximo passo do algoritmo é avaliar o impacto que a primeira VM da lista causa no primeiro servidor, analisando a carga de CPU, a memória, a temperatura e o tráfego. Isso é feito nas Linhas 8, 9, 11 e 12, respectivamente. Caso o servidor atenda aos requisitos, a VM é alocada nele (Linha 13). Caso contrário, um servidor é removido do banco de servidores quando habilitado (Linhas de 20 a 24). Se não for localizado nenhum servidor, a VM não é migrada e permanece no servidor até a próxima tentativa.

5.2.3 Algoritmo HIGH_TEMPERATURE

O algoritmo HIGH_TEMPERATURE, Algoritmo 18, inicialmente ordena as VMs em ordem decrescente de utilização de CPU (Linha 2) e os servidores, que estão aptos a receber uma VM, em ordem ascendente de temperatura (Linha 3). Essa estratégia move uma máquina virtual

Algoritmo 17: ALGORITMO 2D - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores, ListaVMs*
Saída: Realocação das VM

```

1 início
2   Calcula ranque dos Servidores;
3   Ordena ListaVMs em ordem decrescente de utilização CPU;
4   Ordena ListaServidores em ordem crescente de ranque;
5   para cada vm em ListaVMs faça
6     para cada servidor em ListaServidores faça
7       escalonado = falso;
8       se servidor.utilizacaoCPU + vm.cargaMediaCPU <
          LIMITE_SUPERIOR_DE_USO_CPU então
9         se servidor.memoria + vm.memoria <= servidor.memoriaTotalServidor então
10          NovaTemp =
11            servidor.calcTempFut(servidor.utilizacaoCPU + vm.cargaMediaCPU);
12          se NovaTemp < TEMPERATURA_EMERGENCIA então
13            se (vm.avr_trafego + servidor.trafego) * 0,80 < capacidade_enlace então
14              Aloca vm no servidor;
15              escalonado = verdadeiro;
16            parar;
17          fim
18        fim
19      fim
20    se não escalonado então
21      remove servidor do banco de servidores;
22      Aloca vm no servidor;
23      RecalculaRanking(servidorSelecioneado);
24    fim
25  fim
26 fim

```

para o primeiro servidor que tenha a temperatura mais baixa no centro de dados, desde que os requisitos de utilização de CPU, memória, temperatura e rede sejam atendidos, Linhas 7, 8, 10 e 11 respectivamente. Esse algoritmo deve ser utilizando nos casos em que os servidores precisam migrar máquinas virtuais que estão com sobrecarga de temperatura, realocando as VMs das regiões mais quentes para as mais frias do centro de dados.

5.2.4 Algoritmo FFD (*The First Fit Decreasing*)

O algoritmo FFD, Algoritmo 19, proposto por Beloglazov (BELOGLAZOV, 2013), tem como objetivo alocar uma máquina virtual no primeiro servidor disponível que suporte a carga de processamento da VM a ser migrada.

Inicialmente, o algoritmo ordena as VMs em ordem decrescente de utilização de CPU (Linha 2) e tenta alocar no primeiro servidor da lista (Linhas 4 a 9). Esse algoritmo tem como objetivo alocar o maior número de VMs em um único servidor até que todos os seus recursos estejam esgotados.

Algoritmo 18: ALGORITMO HIGH_TEMPERATURE - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores, ListaVMs*
Saída: Realocação das VM

```

1 início
2   Ordena ListaVMs em ordem decrescente de utilização CPU;
3   Ordena ListaServidores em ordem crescente de temperatura;
4   para cada vm em ListaVMs faça
5     para cada servidor em ListaServidores faça
6       escalonado = falso;
7       se servidor.utilizacaoCPU + vm.cargaMediaCPU <
          LIMITE_SUPERIOR_DE_USO_CPU então
8         se servidor.memoria + vm.memoria <= servidor.memoriaTotalServidor então
9           NovaTemp =
10            servidor.calcTempFut(servidor.utilizacaoCPU + vm.cargaMediaCPU);
11            se NovaTemp < TEMPERATURA_EMERGENCIA então
12              se (vm.avr_trafego + servidor.trafego) * 0,80 < capacidade_enlace então
13                Aloca vm no servidor;
14                escalonado = verdadeiro;
15                parar;
16            fim
17          fim
18        fim
19      se não escalonado então
20        remove servidor do banco de servidores;
21        Aloca vm no servidor;
22        RecalculaRanking(servidor.Selecionado);
23      fim
24    fim
25  fim
26 fim

```

Algoritmo 19: ALGORITMO FFD - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores, ListaVMs*
Saída: Realocação das VM

```

1 início
2   Ordena ListaVMs em ordem decrescente de utilização;
3   para cada vm in ListaVMs faça
4     para cada servidor in ListaServidores faça
5       se servidor tem recursos suficientes para vm então
6         Aloca vm no servidor;
7         parar;
8       fim
9     fim
10  fim
11  para cada servidor in ListaServidores faça
12    se servidor não foi alocado vm então
13      suspende servidor;
14    fim
15  fim
16 fim

```

5.2.5 Algoritmo PABFD (Power Aware Best Fit Decreasing)

O algoritmo PABFD, Algoritmo 20, proposto por Beloglazov (BELOGLAZOV, 2013), tem como objetivo alocar as VMs no servidor que tenha o menor consumo energético, levando em conta a demanda energética da nova VM. Primeiro, o algoritmo coloca todos os servidores em hibernação (Linhas 2 a 4). Depois disso, as VMs são ordenadas em ordem decrescente de utilização de CPU (Linha 5). Para cada VM que será alocada, o algoritmo percorre todos os

servidores e identifica o servidor com o menor consumo energético (Linhas 7 a 19). Com isso, uma vez identificado o menor consumo, a VM é alocada (Linha 20 a 23). O objetivo do PABFD é distribuir as VMs de forma a manter o consumo energético dos servidores uniforme.

Algoritmo 20: ALGORITMO PABFD - COMPLEXIDADE $O(n^2)$.

Entrada: *ListaServidores, ListaVMs*
Saída: Realocação das VM

```

1 início
2   para cada servidor em ListaServidores faça
3     | Assumir que servidor esteja Suspenso;
4   fim
5   Ordena ListaVMs em ordem decrescente de utilização;
6   para cada vm em ListaVMs faça
7     | minConsumo ← ∞;
8     | servidorAlocado ← Null;
9     para cada servidor em ListaServidores faça
10      | se servidor tem recursos suficientes para vm então
11        | se servidor esta Suspenso então
12          | servidor foiSuspenso ← verdadeiro;
13          | Assumir Acorde-via-Lan o servidor;
14        fim
15        consumo ← estimarConsumoEnergia(servidor, vm);
16        se servidor foiSuspenso = verdadeiro então
17          | Assumir servidor esteja suspenso;
18        fim
19      fim
20      se consumo < minConsumo então
21        | alocaServidor ← servidor;
22        | minConsumo ← consumo;
23      fim
24    fim
25    se alocaServidor ≠ Null então
26      | Aloca vm no servidor;
27    fim
28  fim
29 fim

```

5.3 Algoritmos de Controle de Temperatura do Sistema de Refrigeração

O sistema de refrigeração de um centro de dados é responsável por manter as temperaturas dos servidores sempre abaixo de limiares de segurança para evitar danos a eles. Para isso, ele recebe informações de sensores de temperatura e diminui a temperatura do ar fornecido para o centro de dados para evitar que os servidores aqueçam. Porém, ao fazer isso, seu consumo de energia aumenta. Portanto, para que o sistema de refrigeração seja efetivo, ele deve reagir para manter a temperatura sempre abaixo dos limiares de segurança, mas ao mesmo tempo ele não pode abaixar muito a temperatura para evitar consumo de energia excessivo e deve até mesmo aumentar a temperatura quando possível para se economizar energia.

O SimDC3D oferece dois algoritmos para controle do sistema de refrigeração. O primeiro adota uma estratégia conservadora para fazer mudanças na temperatura do ar fornecido pelo

sistema de refrigeração e foi importado do SimWare. O segundo foi desenvolvido neste trabalho e adota uma estratégia mais agressiva, pois o ajuste conservador do algoritmo padrão do SimWare pode provocar superaquecimento nos servidores.

5.3.1 Algoritmo Padrão do SimWare

O algoritmo padrão do SimWare foi proposto em (YEO; LEE, 2012) e está reproduzido no Algoritmo 21. A estratégia do algoritmo é subir a temperatura em $0,01^{\circ}C$ a cada segundo, até que um servidor ativo tenha a temperatura de entrada superior a temperatura definida no parâmetro *EMERGENCY_TEMPERATURE*. A partir desse momento, a temperatura é ajustada para descer de $0,01^{\circ}C$ por segundo até que nenhum servidor esteja acima da temperatura de emergência. Após suprir a demanda por ar frio, o processo se inicia novamente subindo a temperatura a cada segundo. A taxa de incremento e decremento pode ser ajustada no parâmetro *CRAC_DISCHARGE_CHANGE_RATE_0_00x*.

Algoritmo 21: ALGORITMO PADRÃO DO SIMWARE - COMPLEXIDADE $O(n^2)$.

Entrada: *temperaturaArAtual, taxa de Ajuste*
Saída: *temperaturaAr*

```

1 início
2   maiorTemperaturaEntrada ← 0
3   temperaturaAr = temperaturaArAtual
4   para cada i em NUMERO_DE_CHASSI faça
5     para cada j em NUMERO_DE_SERVIDORES_NO_CHASSI faça
6       temperaturaServidor = servidor[i][j].temperaturaAtualEntrada();
7       se temperaturaServidor > maiorTemperaturaEntrada então
8         maiorTemperaturaEntrada ← temperaturaServidor
9       fim
10    fim
11  fim
12  se TEMPERATURA_EMERGENCIA > maiorTemperaturaEntrada então
13    | temperaturaAr = temperaturaAr + taxa de Ajuste
14  fim
15  senão
16    | temperaturaAr = temperaturaAr - taxa de Ajuste
17  fim
18  retorna temperaturaAr
19 fim

```

5.3.2 Algoritmo Agressivo do SimDC3D

O algoritmo agressivo do SimDC3D, mostrado no Algoritmo 22, fornece ar frio inicialmente com uma temperatura de $10^{\circ}C$ e aumenta em $0,1^{\circ}C$ a cada segundo até que a maior temperatura de um servidor seja de $28^{\circ}C$. A partir daí, o algoritmo aumenta a temperatura em $0,01^{\circ}C$ até que haja algum servidor com temperatura igual a $29,5^{\circ}C$. Desse ponto em diante, o algoritmo não aumenta mais a temperatura. Caso algum servidor atinja a temperatura de emergência (*EMERGENCY_TEMPERATURE*) definida em $29,99^{\circ}C$, o algoritmo começa a diminuir a temperatura do ar condicionado em $0,2^{\circ}C$ até que todos os servidores atinjam temperaturas abaixo da temperatura de emergência.

O objetivo desse algoritmo é aumentar aceleradamente a temperatura do ar condicionado para economizar energia ou reduzir a temperatura do ar rapidamente para combater pontos quentes até um determinado limite, isto é, $28^{\circ}C$. No intervalo entre $28^{\circ}C$ e $29,5^{\circ}C$, o algoritmo muda para ajustes mais finos, impedindo que modificações abruptas no termostato elevem a temperatura de um servidor acima da temperatura máxima permitida. Um servidor ao atingir uma temperatura no intervalo entre $29,5^{\circ}C$ e $29,9^{\circ}C$ faz com que o algoritmo pare de ajustar o termostato, devido a proximidade da temperatura de emergência (*EMERGENCY_TEMPERATURE*). O ajuste para baixo os valores utilizados no algoritmo ($28^{\circ}C$ e $29,5^{\circ}C$) produz duas consequências: aumento no consumo energético no sistema de refrigeração e redução do tempo de sobrecarga térmica nos servidores. Desta forma, é possível realizar ajustes que acabe completamente com sobrecarga térmica.

Algoritmo 22: ALGORITMO AGRESSIVO DO SIMDC3D - COMPLEXIDADE $O(n^2)$.

```

Entrada: temperaturaArAtual, taxa de Ajuste
Saída: temperaturaAr
1 início
2   maiorTemperaturaEntrada  $\leftarrow$  0
3   temperaturaAr = temperaturaArAtual
4   para cada  $i$  em NUMERO_DE_CHASSI faça
5     para cada  $j$  em NUMERO_DE_SERVIDORES_NO_CHASSI faça
6       temperaturaServidor = servidor[i][j].temperaturaAtualEntrada();
7       se temperaturaServidor > maiorTemperaturaEntrada então
8         maiorTemperaturaEntrada  $\leftarrow$  temperaturaServidor
9       fim
10    fim
11  fim
12  se maiorTemperaturaEntrada < 28 e
13    maiorTemperaturaEntrada < TEMPERATURA_EMERGENCIA então
14    temperaturaAr = temperaturaAr + (taxa de Ajuste * 10)
15  fim
16  senão
17    se maiorTemperaturaEntrada  $\geq$  28 e maiorTemperaturaEntrada < 29.5 e
18      temperaturaAr < TEMPERATURA_EMERGENCIA então
19      temperaturaAr = temperaturaAr + taxa de Ajuste
20    fim
21    senão
22      se maiorTemperaturaEntrada  $\geq$  29.5 e
23        maiorTemperaturaEntrada < TEMPERATURA_EMERGENCIA então
24        temperaturaAr = temperaturaAr + 0
25      fim
26    senão
27      temperaturaAr = temperaturaAr - (taxa de Ajuste * 20)
28    fim
29  retorna temperaturaAr
fim

```

5.4 Algoritmo de Otimização da Topologia da Rede

O Algoritmo 23 de otimização da topologia da rede do centro de dados foi desenvolvido neste trabalho e está implementado no SimDC3D. Inicialmente, o algoritmo percorre cada comutador

de acesso para verificar se há mais de um enlace interligando a camada de acesso com a camada superior (agregação, núcleo). Caso haja, o próximo passo é verificar se um único enlace suporta todo o tráfego entre as duas camadas. Se o enlace suportar, os fluxos de dados são roteados para o enlace escolhido e os demais enlaces são desativados. O algoritmo percorre todos os comutadores de acesso e posteriormente os de agregação movendo o maior número possível de fluxos para o menor número possível de enlaces e comutadores. Após concluir essa etapa, o controlador central procura comutadores que não estão sendo utilizados para serem desligados.

O processo de otimização descrito acima é executado periodicamente com o intervalo de tempo definido pelo parâmetro *PERIODIC_INTERVAL_EXECUTION_TOPOLOGY*. Para evitar congestionamentos prolongados causados pela otimização da topologia, o controlador monitora a utilização dos enlaces a cada segundo. Caso o tráfego se aproxime da capacidade máxima do enlace, definida na configuração da topologia, e existam enlaces e comutadores desligados, o controlador os ativa para suprir a demanda por uma maior capacidade de banda e dissipar rapidamente os pontos de congestionamento entre as camadas.

5.5 Considerações Finais

Neste capítulo, foram descritos os algoritmos de escalonamento e consolidação de máquinas virtuais, controle de temperatura e otimização de topologia. Os escalonadores descritos dispõem de diferentes estratégias de escalonamento para alocar uma máquina virtual em um servidor. No próximo capítulo, serão reportados os resultados de simulações gerados para avaliar alguns escalonadores descritos nesse capítulo. Também foram apresentadas as estratégias utilizadas pelo controlador para consolidar máquinas virtuais em um conjunto menor de servidores. Esses algoritmos também serão avaliados no próximo capítulo. Além disso, foram descritos os algoritmos de ajuste de temperatura do ar condicionado e uma estratégia de otimização de topologia de rede. Todos os algoritmos descritos neste capítulo estão implementados no SimDC3D e disponíveis para experimentação.

Algoritmo 23: ALGORITMO OTIMIZAÇÃO TOPOLOGIA DE REDE -
 COMPLEXIDADE $O(n^2)$.

Entrada: *LComutadoresAcesso, LComutadoresAgregacao, LComutadoresCore,*
LEnlacesAcessoAgregacao, LEnlacesAgregacaoCore

Saída: *TopologiaOtimizada*

```

1 início
2   para cada comutador em LComutadoresAcesso faça
3     para cada enlace em LEnlacesAcessoAgregacao faça
4       | trafego = trafego + comutador.enlace.trafego;
5     fim
6     para cada enlace em LEnlacesAcessoAgregacao faça
7       | se trafego < comutador.enlace.velocidade então
8         | mover todo trafego para comutador.enlace;
9       fim
10    fim
11    para cada enlace em LEnlacesAcessoAgregacao faça
12      | se trafego = 0 então
13        | desativar comutador.enlace;
14      fim
15    fim
16  fim
17  para cada comutador em LComutadoresAgregacao faça
18    para cada enlace em LEnlacesAgregacaoNucleo faça
19      | trafego = trafego + comutador.enlace.trafego;
20    fim
21    para cada enlace em LEnlacesAgregacaoNucleo faça
22      | se trafego < comutador.enlace.velocidade então
23        | mover todo trafego para comutador.enlace;
24      fim
25    fim
26    para cada enlace em LEnlacesAgregacaoNucleo faça
27      | se trafego = 0 então
28        | desativar comutador.enlace;
29      fim
30    fim
31  fim
32  para cada comutador em LComutadoresAgregacao faça
33    | se todas portas desativas então
34    | desligar comutador;
35  fim
36  fim
37  para cada comutador em LComutadoresNucleo faça
38    | se todas portas desativas então
39    | desligar comutador;
40  fim
41  fim
42 fim
  
```

CAPÍTULO 6

RESULTADOS EXPERIMENTAIS NO SIMULADOR

SIMDC3D

Este capítulo apresenta os resultados de sete estudos de caso que avaliam diferentes aspectos de um centro de dados. Em particular, são discutidos resultados que permitem analisar as diferentes funcionalidades implementadas no simulador SimDC3D, detalhar o comportamento energético das três dimensões do centro de dados, as relações existentes entre as otimizações que ocorrem em cada uma das dimensões, avaliar os resultados dos algoritmos de predição e, por último, analisar e comparar o desempenho dos algoritmos de otimização propostos no Capítulo 5.

O próximo ponto é discutir os resultados obtidos em termos dos seguintes itens: consumo energético global e específico de cada dimensão do centro de dados, temperatura média, SLA, tempo de inatividade das VMs durante o processo de migração (*Downtime*), total de migrações por política de realocação de VMs, estatísticas das filas de escalonamento, estatísticas dos traços utilizados nas simulações, estatísticas dos preditores e, principalmente, o *tradeoff* entre consumo de energia versus qualidade de serviço.

O restante deste capítulo está organizado da seguinte forma. A Seção 6.1 descreve os parâmetros do centro de dados simulado. Na seção 6.2, são detalhados os demais parâmetros utilizados nas simulações, como algoritmos de otimização, políticas de controle de temperatura, uso ou não de predição, entre outros. A Seção 6.3 descreve indicadores de desempenho que são analisados nos estudos de caso. Na Seção 6.4, são discutidos os resultados de sete estudos de caso. E a Seção 6.5 apresenta as considerações finais deste capítulo.

6.1 Parâmetros do Centro de Dados

O ambiente simulado consiste em um centro de dados com as seguintes características: 13 racks; 50 chassis; e 10 servidores por chassi (totalizando 500 servidores), cada um com 10 núcleos, 128GB de memória e interface de rede de 1 Gbps. Cada servidor está configurado

para consumir no máximo 260 Watts (100% utilização) e no mínimo 130 Watts (ocioso). O consumo de energia de um processador é a somatória do consumo de energia em modo ocioso mais o consumo dinâmico que depende da carga de processamento em um instante t .

Dependendo do estudo de caso, os servidores podem ser ajustados para desligar/ligar ou hibernar/acordar. Quando parametrizado para desligar/ligar, o tempo estipulado é de 40 segundos para desligar e 70 segundos para ligar. Os tempos para hibernar/acordar são 20 e 10 segundos, respectivamente. Nesses períodos, os servidores não processam carga de trabalho. O consumo energético de cada servidor em hibernação é de 5 Watts. Além disso, os ventiladores dos processadores foram parametrizados para terem rotações e consumos energéticos variáveis de acordo com a carga de processamento.

Os parâmetros de limites de utilização da CPU $THRESHOLD_TOP_USE_OF_CPU$ e $THRESHOLD_BOTTOM_USE_OF_CPU$ foram definidos como 0,90 e 0,30, respectivamente. Esses parâmetros são utilizados para determinar se um servidor está sobrecarregado ou subutilizado. O banco de servidores possui tamanho variável no Cenário 1 e 30 servidores nos demais cenários que o usam.

As máquinas virtuais podem assumir três configurações de memória (1, 2 e 4 GB) escolhidas aleatoriamente, sendo que cada máquina virtual possui apenas um núcleo de processamento. O parâmetro $IMAGE_SIZE_VIRTUAL_MACHINE$ especifica o tamanho do arquivo de imagem de uma máquina virtual. Esse parâmetro possui valor 200 MB nos estudos de caso deste capítulo. Durante o processo de migração, a imagem de uma VM deve ser enviada do servidor de origem para o servidor destino. A cópia da memória da VM é iniciada somente após a transferência da imagem da VM. O tempo de inatividade (*Downtime*) de uma VM é calculado de acordo com (LUČANIN et al., 2014) e como descrito na Seção 4.4.1. Já o tempo de migração de uma VM é calculado somando-se o tamanho de sua imagem mais o tamanho de sua memória e dividindo-se esse resultado pela velocidade do enlace entre o servidor de origem e o de destino.

Dependendo do estudo de caso, a topologia pode ser de duas camadas, três camadas ou três camadas com enlaces redundantes. A topologia de duas camadas possui quatro comutadores de núcleo de 24 portas de 10 Gbps e 13 comutadores de acesso com 48 portas de 1 Gbps e 4 portas de 10 Gbps. Os enlaces entre os comutadores de acesso e núcleo são de 10 Gbps e dos servidores com os comutadores de acesso são de 1 Gbps. Cada comutador de acesso é conectado com todos os comutadores de núcleo provendo redundância nas conexões, como pode ser visto na Figura 6.1(a). A topologia de três camadas possui dois comutadores de núcleo de 24 portas de 10 Gbps, quatro comutadores de agregação de 24 portas de 10 Gbps e 13 comutadores de acesso com 48 portas de 1 Gbps e quatro portas de 10 Gbps. Os enlaces entre os comutadores de acesso e agregação são de 10 Gbps. Todos os comutadores de agregação possuem um enlace para cada comutador de núcleo e os comutadores de acesso estão ligados somente a um dos quatro comutadores de agregação, como mostra a Figura 6.1(b). E, por último, a topologia de três camadas com enlaces redundantes têm as mesmas quantidades e característica da topologia

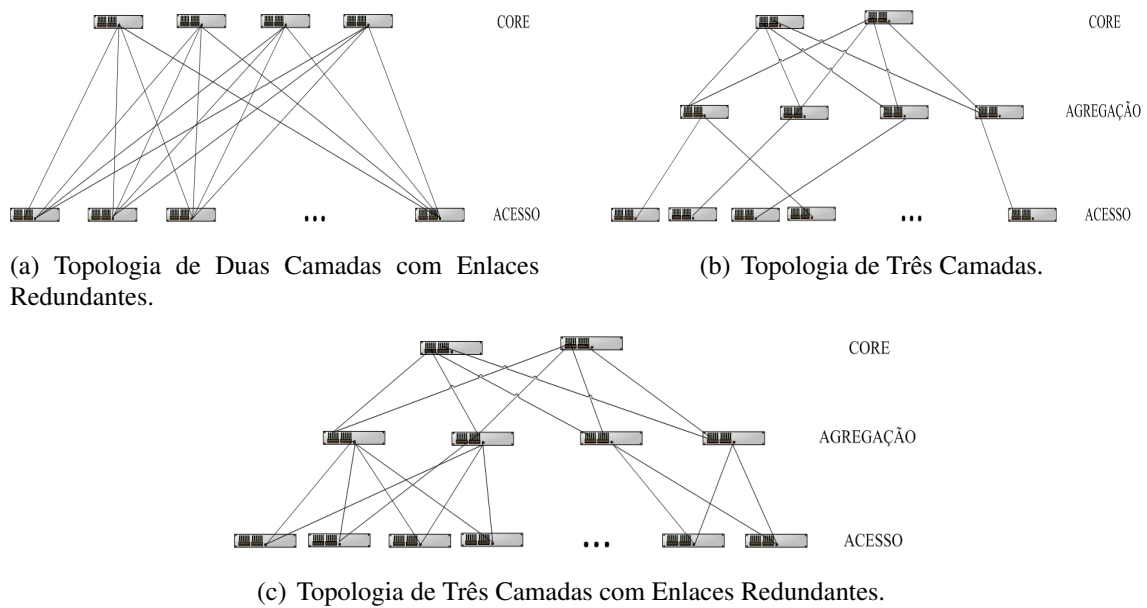


Figura 6.1 – Topologias Simuladas nos Estudos de Caso.

de três camadas, porém um comutador de acesso é conectado a dois comutadores de agregação, como pode ser visto na Figura 6.1(c).

O chassi dos comutadores de acesso e agregação foram parametrizados para consumir 70W e as portas 0,20, 0,30, 0,65 e 1,30 Watts nas velocidades entre 10 Mbps a 10 Gbps. Já o chassi do comutador de núcleo consome 185W e as portas 0,12, 0,18, 0,89 e 1,60 Watts nas velocidades entre 10 Mbps a 10 Gbps. O consumo energético de cada comutador é calculado conforme a Equação (4.6).

Em todas os estudos de caso, o parâmetro *DYNAMIC_WORKLOAD_VMS* está habilitado. Esse parâmetro faz com que a utilização de CPU seja variável em cada VM de acordo com a carga de processamento definida no arquivo de traço. Lembrando que o Algoritmo 1 definido na Seção 4.3.1.2 é utilizado para o cálculo de utilização de CPU em cada segundo de simulação.

Para executar uma simulação no SimDC3D, é necessário fornecer um traço com as cargas de trabalho que as máquinas virtuais devem executar. O traço deve estar no formato SWF e conter os tempos de chegada das cargas de trabalho no centro de dados, o tamanho da memória e a utilização média de CPU que cada carga exige. As características de cada traço utilizado nos estudos de caso estão resumidas na Tabela 6.1.

Tabela 6.1 – Características dos traços utilizados nos estudos de caso do SimDC3D.

	LANL-CM5	METACENTRUM	UNILU-GAIA	SHARCNET	RICC
Tempo total (hora)/(dia)	834 / 34	815 / 33	2114 / 88	499 / 25	323 / 13
Tempo médio de chegada das VMs (s)	360,50	41,47	183,05	75,36	150,29
Desvio padrão (s)	1.356,55	114,90	1.385,60	204,68	1.047,21
Tempo médio de execução das VMs (s)	3.540,27	23.316,70	13.500,50	33.808,40	10.395,40
Desvio padrão de execução (s)	6.330,79	91.946,40	52.209,00	150.671,00	29.903,20
Total de VMs que chegaram no centro de dados	696.256	54.488	516.422	102.846	206.858
Média de utilização do centro de dados	10,67%	8,68%	3,61%	21,58%	37,15%

De acordo com o estudo de caso, pode-se utilizar um ou mais traços e uma combinação de algoritmos para demonstrar uma propriedade específica sob uma determinada carga de processamento. Os traços estão disponíveis em (HEBREW, 2018) e foram extraídos dos seguintes *clusters*: Grid do Laboratório Nacional Los Alamos (LANL); Grid Nacional da República Tcheca (METACENTRUM); Grid Gaia da Universidade de Luxemburgo (UniLU-Gaia); Grid de instituições acadêmicas de Ontário no Canadá (SHARCNET); e *RIKEN Integrated Cluster of Clusters do Japão* (RICC).

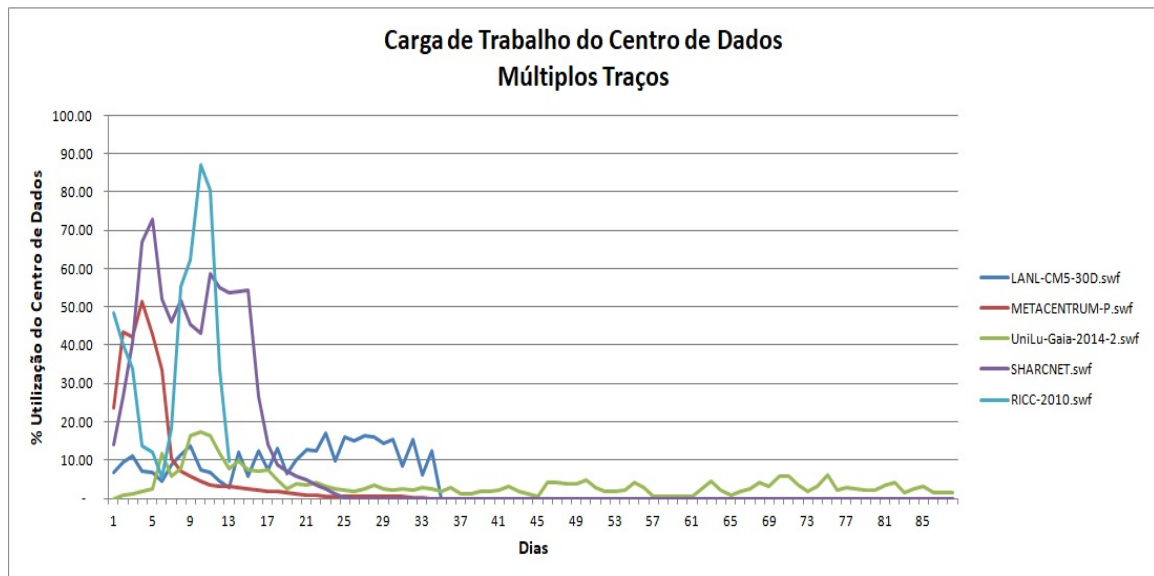


Figura 6.2 – Cargas de trabalho aplicadas no centro de dados.

Cada traço aplica uma carga de processamento diferente no centro de dados, como mostra a Figura 6.2. Os traços também têm características bem distintas, como mostra a Tabela 6.1, em relação ao total de VMs que chegam ao centro de dados, o tempo médio de chegada entre VMs e, principalmente, o tempo médio de execução de cada VM.

6.2 Parâmetros de Simulação

Para avaliar as principais características do SimDC3D e dos algoritmos propostos, foram realizados sete estudos de caso com diferentes parâmetros e cargas de processamento. Atualmente, o SimDC3D disponibiliza 137 parâmetros subdivididos em 14 categorias, como descrito no Capítulo 4 e no Apêndice B. Os parâmetros da Categoria 1 são fixos para todos os estudos de caso e definem as configurações gerais, como sementes, nomes dos traços, número de *threads*, entre outros. Os demais parâmetros utilizados são descritos abaixo.

1. **Categoria 2 – HIBERNATING_SERVER:** define se os servidores hibernam ou desligam durante a simulação. Servidores em hibernação consomem uma pequena quantidade de energia que é definida pelo parâmetro *POWER_CONSUMPTION_HIBERNATING*. Um estudo de caso no Capítulo 6 compara as duas estratégias.

2. **Categoria 4 – POLICY_OF_TEMPERATURE_CONTROL:** determina a política de controle de temperatura do ar condicionado. O SimDC3D possui dois algoritmos de controle de temperatura que foram descritos na Seção 4.3.2.3 e ambos foram utilizados nos estudos de caso.
3. **Categoria 5 – SCHEDULING_ALGORITHM:** indica o algoritmo utilizado no escalonamento das VMs que chegam ao centro de dados. O SimDC3D possui 14 algoritmos de escalonamento que foram descritos na Seção 5.1. Cada estudo de caso descreve o algoritmo utilizado.
4. **Categoria 5 – SCHEDULING_WITH_PREDICTION:** habilita o monitoramento e armazenamento de séries temporais que são coletadas de uma ou mais variáveis ordenadas no tempo. Escalonadores que utilizam preditores devem habilitar o monitoramento das variáveis que desejam utilizar para fazer previsões.
5. **Categoria 5 – PREDICTION_ALGORITHM:** estipula qual algoritmo de previsão é utilizado nos algoritmos de escalonamento. O parâmetro *SCHEDULING_WITH_PREDICTION* deve ser verdadeiro para ativar esta funcionalidade e o parâmetro *VARIABLE_PREDICTED* define qual variável é utilizada na previsão. Dois algoritmos de previsão estão disponíveis, o *RBF* e o *Polynom*. Eles estão descritos nos Anexos C e D, respectivamente, e foram utilizados no estudo de caso que avalia os preditores.
6. **Categoria 10 – OPTIMIZATION_ALGORITHM_LOW_UTILIZATION:** especifica o algoritmo de otimização utilizado quando o servidor está com baixa utilização. O parâmetro *THRESHOLD_BOTTOM_USE_OF_CPU* determina o limite inferior de utilização da CPU. Os algoritmos de otimização foram descritos na Seção 5.2. Para cada estudo de caso é indicado o algoritmo utilizado.
7. **Categoria 10 – OPTIMIZATION_ALGORITHM_OVERLOAD_UTILIZATION:** define qual algoritmo de otimização é usado quando o servidor está sobrecarregado. O parâmetro *THRESHOLD_TOP_USE_OF_CPU* determina o limite superior de utilização da CPU. Para cada estudo de caso é indicado o algoritmo utilizado.
8. **Categoria 10 – OPTIMIZATION_ALGORITHM_OVERHEATING_TEMPERATURE:** determina qual algoritmo de otimização é utilizado quando o servidor está superaquecido. O parâmetro *EMERGENCY_TEMPERATURE* define o limite superior da temperatura para considerar um servidor superaquecido. Os algoritmos são descritos na Seção 4.3.2.3.
9. **Categoria 10 – OPTIMIZATION_ALGORITHM_LINK_OVERLOAD:** define qual algoritmo de otimização é executado quando os enlaces do servidor com os comutadores de acesso estão sobrecarregados. O algoritmo de otimização de topologia está descrito na Seção 5.4.

10. **Categoria 10 – USING_PREDICTION_TO_OVERLOAD:** habilita a utilização do preditor de CPU para prevenir sobrecargas da CPU.
11. **Categoria 10 – PREDICTION_ALGORITHM_OVERLOAD:** define qual algoritmo é utilizado para estimar a utilização de CPU em um servidor.
12. **Categoria 10 – VMS_ALGORITHM_SELECTION:** define como as máquinas virtuais são ordenadas em um servidor para serem selecionadas durante o processo de migração. Os critérios de ordenação estão definidos na Seção 5.2.
13. **Categoria 12 – SIMULATES_NETWORK:** habilita ou desabilita a simulação da infraestrutura de rede.
14. **Categoria 13 – SIMULATE_TRAFFIC_MATRIX:** define se o tráfego de rede deve ser simulado. Caso seja habilitada a simulação da rede e não seja habilitada a simulação do tráfego, o consumo energético de cada comutador é calculado utilizando as velocidades dos enlaces definidas no FNSS. Neste caso, os algoritmos de escalonamento e migração de máquinas virtuais não usam informações de utilização dos enlaces. As funcionalidades de monitoramento e adaptação das velocidades nas portas dos comutadores são desprezadas mesmo que estejam ativas.
15. **Categoria 13 – DYNAMIC_SPEED_ADAPTATION:** determina se o controlador muda as velocidades das portas dos comutadores de acordo com o volume de tráfego do enlace, como descrito na Seção 4.3.2.4.
16. **Categoria 14 – POOL_SIZE:** define o tamanho do banco de servidores.

6.3 Indicadores de Desempenho do Centro de Dados

Diversos fatores interferem no consumo energético do centro de dados. Portanto, análises unidimensionais não são adequadas, pois podem indicar ganhos ou perdas irrealistas. Além disso, economias excessivas de energia podem levar a degradações em parâmetros de qualidade de serviço. Esta seção apresenta as variáveis que são calculadas pelo SimDC3D para se avaliar o impacto de algoritmos de otimização e os efeitos que eles geram nos indicadores de qualidade do centro de dados. Esses indicadores fornecem subsídios para se avaliar a relação entre economia de energia e qualidade de serviço. Comprometer a qualidade de serviço em detrimento de economia de energia não é desejável para operadores de centro de dados, pois violações de acordos de nível de serviço podem levar a multas contratuais muito maiores que os montantes obtidos com economia de energia.

Para que um operador de centro de dados possa compreender melhor as relações existentes nos diversos subsistemas de um centro de dados, o SimDC3D disponibiliza as seguintes variáveis:

1. **Total de migrações de máquinas virtuais:** indica o total de migrações de máquinas virtuais que ocorrem no centro de dados durante uma simulação. Economias de energia na dimensão servidores são alcançadas, principalmente, quando se mantém uma quantidade mínima de servidores ligados para atender à demanda de processamento. As combinações de um conjunto de variáveis dos algoritmos de otimização podem levar a um número elevado de migrações e comprometer os indicadores de *Downtime* e *SLA*. Este indicador é influenciado por diversos fatores, como quantidade de servidores ligados, temperatura do centro de dados e estratégias de migração de máquinas virtuais.
2. **Servidores Desligados:** indica a quantidade de servidores que foram desligados por baixa utilização durante a simulação. Esta variável possui duas interpretações. Primeiro, quanto mais servidores desligados, menor é o calor gerado em um determinado instante, gerando economia no sistema de refrigeração e servidores. Segundo, esta variável pode indicar que o escalonamento e otimização de máquinas virtuais estão mantendo um conjunto muito pequeno de servidores ligados e, conseqüentemente, aumentando o número de migrações de máquinas virtuais.
3. **SLA:** especifica o percentual de máquinas no centro de dados que sofreram sobrecarga de CPU. Minimizar o número de servidores físicos diminui o consumo energético. Entretanto, o aumento de máquinas virtuais em um servidor pode gerar sobrecargas da CPU e quebrar os acordos de nível de serviço.
4. **Downtime:** determina o percentual de tempo em que uma máquina sofreu interrupção durante o processo de migração de uma máquina virtual, ou seja, quanto maior o *Downtime*, maior é a interrupção sofrida durante todas as migrações de um servidor. O número total de migrações que ocorre em um servidor influencia o *Downtime*. Minimizar o número de migrações gera uma menor indisponibilidade das máquinas virtuais. O cálculo de *Downtime* foi proposto por Luvcanin *et al.* (LUČANIN *et al.*, 2014) e é dado pela Equação 4.8.
5. **Temperatura Média do Centro de Dados:** indica a temperatura média do centro de dados. Manter o centro de dados com temperaturas mais altas reduz o consumo energético do sistema de refrigeração. Porém, manter o centro de dados com temperaturas elevadas pode causar superaquecimento nos servidores em determinados instantes. Estratégias de controle de temperatura permitem equilibrar a relação da temperatura do ar condicionado com a temperatura dos servidores, proporcionando economia sem ultrapassar o limite de temperatura definido pelo parâmetro *EMERGENCY_TEMPERATURE*.
6. **Tempo Médio com Servidores Super Aquecidos:** especifica o tempo médio, não necessariamente consecutivo, que os servidores ficaram com temperaturas acima da temperatura máxima permitida. Esta variável é afetada diretamente pelo controle de temperatura. Uma das conseqüências de temperaturas altas é a diminuição da vida útil do

hardware, o que leva a prejuízos financeiros para o operador do centro de dados. Além disso, temperaturas altas podem causar falhas de *hardware* e interrupções dos serviços.

7. **Consumo Energético das três dimensões:** detalha o consumo energético de cada dimensão do centro de dados, exibindo valores consolidados e detalhados de cada subsistema simulado.
8. **Tempo médio das VMs na Fila de Escalonamento:** indica o tempo médio que as VMs permaneceram na fila aguardando para serem escalonadas. Um tempo médio alto indica que o centro de dados está operando com alta carga ou um número reduzido de servidores ligados.

6.4 Estudos de Caso

Esta seção apresenta resultados de simulação do SimDC3D para avaliar vários aspectos de um centro de dados, incluindo o desempenho dos algoritmos propostos neste trabalho: 3DMOBFDF, 2D, controle de temperatura (agressivo) e otimização de topologia. Além dos resultados do 3DMOBFDF e 2D, serão apresentados resultados de outros dois algoritmos descritos em Beloglazov *et al.* (BELOGLAZOV *et al.*, 2010): o FFD (*First Fit Decreasing*), que aloca uma máquina virtual no primeiro servidor disponível com capacidade para recebê-la; e o PABFD (*Power-Aware Best Fit Decreasing*), que foi descrito na Seção 5.2.5.

O algoritmo 3DMOBFDF permite a parametrização do *ranking* para privilegiar uma determinada dimensão sobre as demais. Para avaliar a efetividade dessa estratégia, foram criadas duas variantes no cálculo do *ranking*. Na primeira, chamada de 3DMOBFDF(a), o *ranking* (Equação 5.1) é calculado com as variáveis de temperatura, consumo energético e CPU, fazendo $\alpha = \beta = \gamma = 1$ e $\delta = \epsilon = 0$, $e_t = 3$, $e_{CPU} = 1$, $e_E = 1$, $W_{Temp} = 0.20$, $W_{CPU} = 0.40$ e $W_{Power} = 0.40$. A segunda variante, chamada de 3DMOBFDF(b), utiliza todas as variáveis da Equação 5.1, fazendo $\alpha = \beta = \gamma = \delta = \epsilon = 1$, $e_t = e_{CPU} = e_E = e_M = e_{Tr} = 3$ e $W_{Temp} = W_{CPU} = W_{Power} = W_{Memory} = W_{Traffic} = 1$.

Os resultados de simulação estão divididos em sete cenários distintos, discutidos nas seções subsequentes. Os cenários construídos analisam os impactos dos seguintes componentes nos algoritmos de alocação de máquinas virtuais: tamanho de um banco de servidores de reserva (Cenário 1); controle de temperatura (Cenário 2); impacto de diferentes cargas de processamento no centro de dados (Cenário 3); otimizações em diferentes topologias de rede (Cenário 4); preditor de CPU (Cenário 5); estratégias de predição em escalonadores de VMs (Cenário 6); e centro de dados tradicional versus centro de dados verde (Cenário 7). Os resultados apresentados são médias de 10 execuções independentes e com intervalo de confiança de 95%.

6.4.1 Cenário 1: Avaliação do Impacto do Banco de Servidores no Consumo Energético e no Desempenho do Centro de Dados

Os principais trabalhos existentes na literatura assumem que sempre há um servidor disponível para receber uma máquina virtual. Entretanto, essa suposição é pouco realista, pois se não há um servidor com capacidade suficiente, a máquina virtual deve esperar até que um novo servidor seja ligado ou acordado (*i.e.*, retirado do modo de hibernação) para que ela seja alocada. O fato de uma máquina virtual ter de esperar pela disponibilidade de algum servidor faz com que o seu tempo de espera de entrada em produção aumente ou sobrecarregue o servidor onde ela está rodando em caso de migrações, podendo levar a violações de acordos de nível de serviço.

O SimDC3D apresenta duas contribuições importantes para eliminar essa hipótese pouco realista. Primeiro, ele não aloca uma máquina virtual imediatamente quando não há servidores disponíveis, simulando realisticamente o tempo de espera até que um servidor esteja disponível. Segundo, ele introduz um banco de servidores de reserva que ficam ociosos a espera de novas cargas de trabalho.

O uso de um banco de servidores no SimDC3D tem a desvantagem de aumentar ligeiramente o consumo de energia do centro de dados, mas pode eliminar completamente o tempo de espera das máquinas virtuais e reduzir o número de migrações por sobrecarga de CPU, pois o banco diminui significativamente o tempo que um servidor em operação fica sobrecarregado. Este trabalho usa como medida de violação de SLA o percentual de tempo que um servidor fica com utilização de CPU igual a 100% (a definição formal de SLA está na Seção 4.4.1). Esse nível de utilização de CPU leva a aumentos dos tempos de resposta das máquinas virtuais hospedadas no servidor e, conseqüentemente, a degradações de qualidade de serviço. Quando há necessidade de servidores para atender novas demandas, o controlador central faz a remoção de servidores do banco e os coloca em operação. Os servidores retirados do banco são substituídos por outros servidores que estavam desligados ou hibernando.

A Tabela 6.2 mostra os efeitos de diferentes tamanhos do banco de servidores no número de migrações, consumo energético, violações de SLA e tempo médio de espera das máquinas virtuais na fila de escalonamento. Este cenário utiliza o traço da SHARCNET e as estratégias de desligar ou hibernar os servidores do centro de dados que não estão no banco de reservas (os servidores do banco estão sempre ligados). O algoritmo de otimização é o 3DMOBFD(a). A temperatura do ar condicionado está fixada em 19°C, *i.e.*, não há controle dinâmico da temperatura. Por fim, a topologia de rede é fixa e não é feita nenhuma otimização durante o experimento.

A Tabela 6.2 também detalha os resultados para os quatro tipos de migrações: sobrecarga, quando um servidor está sobrecarregado; baixa utilização, quando um servidor está com pouca carga de trabalho e suas máquinas virtuais podem ser migradas para que ele seja desligado; alta temperatura, quando um servidor está com a temperatura elevada; e predição, quando o preditor de utilização de CPU indica que máquinas virtuais devem ser migradas para se evitar sobrecarga futura (neste experimento foi utilizada a regressão linear para o preditor).

Tabela 6.2 – Desempenho do centro de dados em função da redução do número de servidores no banco.

	SEM BANCO DE SERVIDORES	BANCO DE SERVIDORES							
		SERVIDORES FORA DO BANCO ESTÃO HIBERNANDO				SERVIDORES FORA DO BANCO ESTÃO DESLIGADOS			
		TAMANHO DO BANCO				TAMANHO DO BANCO			
		Servidores Hibernando	50	40	30	20	50	40	30
Total de Migrações									
Sobrecarga	47.765	13.460	14.749	12.744	13.129	12.932	12.275	11.276	10.784
Predição	255.524	54.676	60.035	49.460	54.342	56.461	55.216	53.811	58.382
Baixa utilização	7.992	11.376	11.359	11.269	10.976	11.332	10.942	10.291	9.929
Altas temperaturas	12	80	79	81	77	79	76	90	96
Enlaces sobrecarregados	7	1	1	0	1	3	2	3	1
TOTAL	311.301 ± 7.068	79.593 ± 4.873	86.224 ± 3.023	73.555 ± 1.064	78.525 ± 3.086	80.808 ± 4.195	78.510 ± 2.757	75.471 ± 1.233	79.191 ± 1.059
Consumo Energético (GW)									
Servidor	70,67	84,90	82,89	80,98	78,98	82,72	80,50	78,67	76,54
Sistema de refrigeração	24,14	29,00	28,31	27,66	26,98	28,25	27,49	26,87	26,14
Computadores	3,41	3,41	3,41	3,41	3,41	3,13	3,12	3,11	3,10
TOTAL	98,22 ± 0,24	117,31 ± 0,25	114,61 ± 0,23	112,06 ± 0,31	109,37 ± 0,35	114,10 ± 0,14	111,12 ± 0,19	108,66 ± 0,23	105,78 ± 0,24
Parâmetros dos Níveis de Serviço (%)									
Violações de SLA	2,89 ± 0,14	0,06 ± 0,00	0,06 ± 0,01	0,06 ± 0,01	0,07 ± 0,01	0,08 ± 0,00	0,10 ± 0,00	0,11 ± 0,01	0,18 ± 0,01
<i>Downtime</i>	4,54 ± 0,47	0,03 ± 0,00	0,07 ± 0,07	0,02 ± 0,00	0,03 ± 0,00	0,03 ± 0,00	0,03 ± 0,00	0,06 ± 0,06	0,18 ± 0,12
Fila de Escalonamento (s)									
Tempo médio	0,38 ± 0,01	-	-	0,00 ± 0,00	0,01 ± 0,00	0,02 ± 0,00	0,06 ± 0,01	0,21 ± 0,01	1,13 ± 0,05
Desvio padrão	0,85	-	-	0,03	0,08	0,11	0,31	0,97	3,59

Os resultados mostram que há um aumento expressivo no número de migrações, tempo médio de espera de máquinas virtuais na fila de escalonamento e no percentual de quebra de SLA nas simulações que não utilizam o banco de servidores. Isso significa que em determinados momentos não há servidores disponíveis e o escalonador precisa esperar até que um servidor seja ligado ou tirado do modo de hibernação. Porém, o consumo energético é menor, devido a uma quantidade menor de servidores ligados/hibernando durante toda a simulação.

Os experimentos também mostram que a variação do número de servidores no banco implica diretamente nos resultados. Pode-se concluir que diferentes tamanhos influenciam no SLA, *Downtime*, tempo na fila de escalonamento e no número de migrações. A melhor relação custo-benefício para esse centro de dados é de 30 servidores em modo de hibernação. Essa configuração privilegia a qualidade de serviço em detrimento do consumo energético e resulta na redução das violações de SLA e tempo médio de espera na fila de escalonamento em comparação com os servidores desligados. Entretanto, a utilização do banco aumenta o consumo de energia em 14,09% em relação a simulação sem a utilização do banco e com os servidores em modo de hibernação e 3,13% em comparação com um banco de 30 servidores considerando que os servidores fora do banco estão desligados.

Tabela 6.3 – Desempenho do centro de dados x controle de temperatura dinâmico.

	CONTROLE ESTÁTICO TEMPERATURA			CONTROLE DINÂMICO DE TEMPERATURA					
	19(°C)			ALGORITMO PADRÃO			ALGORITMO AGRESSIVO		
	PABFD	3DMOBFDF (a)	3DMOBFDF (b)	PABFD	3DMOBFDF (a)	3DMOBFDF (b)	PABFD	3DMOBFDF (a)	3DMOBFDF (b)
Total de Migrações									
Sobrecarga	13.486	12.996	19.456	22.340	18.657	36.679	12.878	12.958	17.467
Predição	52.439	50.597	71.141	80.998	71.432	160.069	49.526	55.104	68.174
Baixa utilização	12.652	11.408	16.204	10.861	10.083	14.545	12.541	11.134	15.422
Altas temperaturas	74	78	81	17.980	12.401	19.825	357	186	217
Enlaces sobrecarregados	1	0	1	2	2	1	1	2	3
TOTAL	78.651 ± 4.660	75.079 ± 4.945	106.883 ± 6.886	132.182 ± 7.322	112.575 ± 8.540	231.119 ± 37.658	75.303 ± 2.309	79.384 ± 4.773	101.284 ± 5.780
Consumo Energético (GW)									
Servidor	82,20	81,22	80,37	77,00	76,78	75,41	82,93	81,71	80,64
Processador	80,54	79,58	78,66	73,58	73,36	71,69	80,51	79,42	78,16
Ventoinhas	1,66	1,63	1,71	3,43	3,42	3,72	2,42	2,29	2,48
Sistema de refrigeração	28,07	27,74	27,45	16,27	16,29	15,89	19,94	20,28	19,61
Comutadores	3,41	3,41	3,41	3,41	3,41	3,41	3,41	3,41	3,41
TOTAL	113,69 ± 0,11	112,37 ± 0,21	111,23 ± 0,21	96,69 ± 0,42	96,49 ± 0,79	94,72 ± 0,15	106,28 ± 0,25	105,40 ± 0,17	103,67 ± 0,20
Temperatura									
Temperatura média do termostato (°C)	19,00 ± 0,00	19,00 ± 0,00	19,00 ± 0,00	27,75 ± 0,02	27,74 ± 0,04	27,83 ± 0,01	25,60 ± 0,06	25,21 ± 0,01	25,47 ± 0,05
Total de chassis com sobrecarga térmica	1 ± 0	1 ± 0	1 ± 0	32 ± 0	33 ± 1	32 ± 0	9 ± 1	9 ± 1	8 ± 1
Tempo médio com sobrecarga térmica por chassi (s)	58 ± 8,00	23,04 ± 0,06	63,80 ± 7,21	13,530 ± 1,652	12,822 ± 1,569	10,507 ± 1,217	51 ± 4	38 ± 5	39 ± 2
Parâmetros dos Níveis de Serviço (%)									
Violações de SLA	0,06 ± 0,01	0,06 ± 0,00	0,06 ± 0,00	0,08 ± 0,01	0,08 ± 0,01	0,10 ± 0,02	0,06 ± 0,01	0,06 ± 0,00	0,07 ± 0,01
Downtime	0,02 ± 0,00	0,10 ± 0,13	0,03 ± 0,00	1,08 ± 0,20	0,90 ± 0,18	1,01 ± 0,29	0,02 ± 0,00	0,03 ± 0,00	0,15 ± 0,13

6.4.2 Cenário 2: Avaliação do Controle de Temperatura nos Indicadores de Desempenho do Centro de Dados e na Temperatura dos Servidores

O SimDC3D permite que a temperatura do ar condicionado seja mantida constante, como no experimento anterior, ou controlada dinamicamente. Nesta seção, serão analisados os impactos de dois algoritmos de controle de temperatura nos algoritmos de alocação de máquinas virtuais PABFD, 3DMOBFDF(a) e 3DMOBFDF(b) utilizando o traço SHARCNET com um banco de 30 servidores ativos e os demais em modo de hibernação. O primeiro algoritmo (*padrão*), descrito na Seção 5.3.1, e o segundo algoritmo (*agressivo*), descrito na Seção 5.3.2.

A Tabela 6.3 mostra que há uma redução no consumo de energético do sistema de refrigeração de 42,04% (28,07GW para 16,27GW) para o PABFD, 41,28% (27,74GW para 16,29GW) para o 3DMOBFDF(a) e de 42,11% (27,45GW para 15,89GW) para o 3DMOBFDF(b) com o controle dinâmico de temperatura *padrão* em relação ao ajuste fixo de temperatura em 19°C. Diferentes percentuais de economias são obtidos porque as otimizações utilizam quantidades diferentes de servidores para executar o mesmo traço, impactando diretamente no consumo energético do sistema de refrigeração. Porém, o algoritmo *padrão* produz sobrecarga térmica em aproximadamente 32 chassis (≈320 servidores), com os servidores ficando com temperatura acima da temperatura de emergência em média de 13.530s (PABFD), 12.822s (3DMOBFDF(a)) e 10.507s (3DMOBFDF(b)). Esses tempos não são consecutivos e representam as somas dos períodos de sobrecarga térmica ao longo de toda a simulação.

Nesse ambiente, o 3DMOBFDF(b) produz o menor tempo de sobrecarga térmica nos servidores (10.507s). Como a temperatura faz parte do cálculo do *ranking*, o 3DMOBFDF(b)

seleciona servidores mais frios. Já o PABFD resulta no maior tempo de sobrecarga (13.530s) por não utilizar a temperatura na sua política de seleção de servidores entre as três políticas. A tabela também mostra que há uma diminuição no consumo de energia de 111.23GW para 94.72GW, mas um aumento nas migrações de 106.883 para 231.119. Esse comportamento é esperado, pois o controle de temperatura permite que o ar condicionado trabalhe com temperaturas mais elevadas e reduza seu consumo de energia.

O algoritmo *agressivo*, por outro lado, diminui significativamente a sobrecarga térmica dos servidores de 32 para 8 chassis, mantém o centro de dados mais frio, em torno de $25^{\circ}C$ em comparação com os $27^{\circ}C$ do algoritmo padrão, e reduz os períodos de sobrecarga em mais de duas ordens de grandeza (38s a 51s). Muito embora o consumo energético seja ligeiramente superior ao do algoritmo *padrão*, os ganhos em relação ao consumo energético com a temperatura do ar condicionado constante são expressivos. Mais especificamente, as reduções são 28,96% (PABFD), 26,89% (3DMOBFD(a)) e 28,56% (3DMOBFD(B)). Além disso, o algoritmo melhora significativamente as métricas de desempenho, como número de migrações, SLA e *Downtime*.

O algoritmo 3DMODBFD (a) obteve o menor consumo energético em relação ao PABFD na dimensão servidor, porém teve um aumento no consumo energético do sistema de refrigeração. Esse aumento ocorre porque ele mantém a temperatura média do centro de dados mais baixa em relação ao PABFD. Os algoritmos 3DMODBFD (a) e (b) obtiveram o menor consumo energético na dimensão servidor e refrigeração em comparação com o PABFD, porém o número de migrações é maior por sobrecarga e predição devido a menor quantidade de servidores alocados.

A utilização do 3DMOBFD(b) com controle de temperatura agressivo reduz o consumo de energia global do centro de dados produzido pelo PABFD em 2,46%. Além disso, as reduções de consumo das estratégias PABFD, 3DMODBFD (a) e 3DMODBFD com controle de temperatura agressivo em comparação com o controle de temperatura estático são de 6,52%, 6,20% e 6,80%, respectivamente.

6.4.3 Cenário 3: Avaliação de Diferentes Cargas de Processamento nos Indicadores de Desempenho

Este cenário avalia o desempenho dos algoritmos FFD, PABFD, 2D e duas variantes do algoritmo 3DMOBFD sob cinco cargas diferentes de processamento representadas pelos traços reais descritos na Tabela 6.1. O traço Unilu-Gaia produz a menor utilização do centro de dados (3,61%), porém ele é o mais longo e cobre um período de 88 dias. O traço da SHARCNET tem uma utilização intermediária (21,58%). Já o traço da LANL-CM5 gera o maior pico de utilização do centro de dados (88%) e o RICC a maior utilização média (37,15%). Cabe salientar, entretanto, que em alguns traços há picos de utilização como mostra a Figura 6.2. O

centro de dados foi configurado com um banco de 30 servidores de reserva, ar condicionado fornecendo ar frio com controle dinâmico de temperatura agressivo e sem otimizações de rede.

As Tabelas 6.4, 6.5 e 6.6 mostram os resultados para os cinco traços e para os algoritmos avaliados. O FFD possui o menor consumo de energia sob as três cargas de trabalho, mas causa mais migrações e violações de SLA em todas as situações. Isso ocorre porque o FFD sobrecarrega os servidores sequencialmente ao alocar uma máquina virtual ao primeiro servidor disponível sem se preocupar com os impactos futuros dessa alocação no servidor.

O algoritmo 2D tem o segundo menor consumo energético, porém as quebras de SLA são em média 43% maiores em comparação ao PABFD e ao 3DMOBFD(a) e (b) para o traço LANL-CM5, 1100% para METACENTRUM, 117% para SHARCNET, 36% para UNILU-GAIA e 110% para RICC. O número de migrações do algoritmo 2D é inferior ao FFD, mas maior em relação às outras estratégias na maioria dos casos.

O algoritmo PABFD reduz em média mais de 72% o número de migrações e em 69% as violações de SLA em comparação ao FFD. Entretanto, ele aumenta, por exemplo, com o traço da LANL-CM5 o consumo energético dos servidores de 66,79GW (FFD) para 68,77GW, METACENTRUM de 59,17GW para 61,58GW. O PABFD mantém mais servidores ativos e distribui melhor as máquinas virtuais nos servidores para evitar migrações constantes, produzindo um resultado mais estável para o centro de dados. Um ponto interessante a ser observado é a redução no consumo de energia das ventoinhas dos servidores. Isso ocorre porque as CPUs dos servidores não ficam tão sobrecarregadas como no FFD e, portanto, exigem menos trabalho das ventoinhas para resfriá-las.

Nas Tabelas 6.4, 6.5 e 6.6, pode-se observar que tanto o 3DMOBFD(a) quanto o 3DMOBFD(b) reduzem o consumo energético em relação ao PABFD. Além disso, o número de migrações é menor em quase todos os traços simulados, demonstrando que as duas versões são mais estáveis e reduzem o número de migrações em relação ao PABFD e ao FFD.

A configuração do 3DMOBFD(a) é menos agressiva em relação à economia de energia, obtendo, por exemplo, uma redução de 1,00% em relação ao PABFD no traço da SHARCNET e no traço da UNILU-GAIA. Já a configuração do 3DMOBFD(b) reduz o número de servidores ligados e obtém maiores economias de energia. Por exemplo, no traço da SHARCNET a redução foi de 2,32% e no da UNILU-GAIA foi de 1,23%. Quando o centro de dados é submetido a maiores cargas, por exemplo, pelo traço da RICC e do LANL-CM5, o desempenho dos algoritmos apresentam resultados semelhantes porque as cargas utilizam praticamente todos os recursos do centro de dados.

Nos parâmetros de níveis de serviço, os traços da UNILU-GAIA e da LANL-CM5 resultaram em pequenas vantagens para o PABFD em relação ao 3DMOBFD. Nos demais traços, o algoritmo 3DMOBFD teve desempenho melhor que o PABFD. Já os tempos de *Downtime* dos três algoritmos foram próximos e apresentaram pouca variabilidade.

Tabela 6.4 – Desempenho do centro de dados x estratégias de alocação de VMs para os traços LANL-CM5 e METACENTRUM.

	LANL-CM5					METACENTRUM				
	FFD	PABFD	2D	3DMOBFDF (a)	3DMOBFDF (b)	FFD	PABFD	2D	3DMOBFDF (a)	3DMOBFDF (b)
Total de Migrações										
Sobrecarga	1.154.987	382.568	535.781	412.419	429.379	24.384	2.158	13.747	1.771	2.056
Predição	1.794.933	409.630	576.414	450.655	489.525	95.952	11.706	31.588	12.843	13.747
Baixa utilização	135.206	129.665	125.791	107.747	115.879	7.597	3.498	4.522	2.936	3.060
Altas temperaturas	1.253	1.956	1.334	1.869	1.735	1.071	617	557	304	241
Enlaces sobrecarregados	17	10	29	9	7	1	3	1	1	1
TOTAL	3.086.396 ± 5.318	923.828 ± 12.542	1.239.349 ± 9.864	972.699 ± 15.890	1.036.525 ± 13.229	129.004 ± 693	17.982 ± 322	50.414 ± 376	17.855 ± 520	19.105 ± 448
Consumo Energético (GW)										
Servidor	66,79	68,77	67,58	68,04	67,92	59,17	61,84	60,96	61,98	61,59
Processador	63,36	65,61	64,21	64,86	64,74	56,93	60,00	58,89	60,20	59,80
Ventoinhas	3,43	3,16	3,37	3,18	3,18	2,24	1,84	2,07	1,78	1,79
Sistema de refrigeração	12,70	12,47	12,11	12,32	12,34	13,47	13,90	13,14	13,74	13,69
Comutadores	5,80	5,80	5,80	5,80	5,80	5,63	5,63	5,63	5,63	5,63
TOTAL	85,29 ± 0,01	87,04 ± 0,05	85,49 ± 0,01	86,17 ± 0,02	86,06 ± 0,02	78,28 ± 0,09	81,37 ± 0,09	79,74 ± 0,08	81,35 ± 0,11	80,92 ± 0,10
Temperatura										
Total de chassis com sobrecarga térmica	15 ± 1	15 ± 1	15 ± 0	15 ± 1	14 ± 1	5 ± 1	7 ± 1	9 ± 0	8 ± 1	9 ± 1
Tempo médio com sobrecarga térmica por chassis (s)	434 ± 18	554 ± 38	684 ± 11	540 ± 22	529 ± 31	82 ± 18	29 ± 3	36 ± 6	21 ± 2	19 ± 2
Parâmetro dos Níveis de Serviço (%)										
Violações de SLA	2,63 ± 0,01	0,73 ± 0,01	1,12 ± 0,01	0,80 ± 0,01	0,81 ± 0,01	0,06 ± 0,01	0,01 ± 0,00	0,12 ± 0,01	0,01 ± 0,00	0,01 ± 0,00
Downtime	0,77 ± 0,01	0,24 ± 0,00	0,36 ± 0,00	0,26 ± 0,00	0,27 ± 0,00	0,12 ± 0,06	0,13 ± 0,19	0,07 ± 0,06	0,05 ± 0,06	0,09 ± 0,08
Fila de Escalonamento (s)										
Tempo médio	0,15 ± 0,00	0,08 ± 0,00	0,09 ± 0,00	0,08 ± 0,00	0,08 ± 0,00	90,60 ± 1,87	96,99 ± 0,98	97,85 ± 0,85	95,84 ± 0,74	95,85 ± 0,57
Desvio padrão	1,12	0,84	0,88	0,86	0,85	382,32	406,62	405,83	402,44	402,89
Máquinas Virtuais										
Número máximo de VMs simultâneas	6.506 ± 6	6.550 ± 10	6.515 ± 13	6.525 ± 35	6.500 ± 34	4.628 ± 19	4.540 ± 4	4.601 ± 12	4.554 ± 5	4.545 ± 7

6.4.4 Cenário 4: Avaliação de Otimizações de Topologias de Rede

Esta seção avalia o desempenho do centro de dados utilizando três topologias de rede (duas camadas, três camadas e três camadas com enlaces redundantes) e o impacto da otimização na topologia em relação às estratégias de alocação de máquinas virtuais. O algoritmo de otimização escolhe um conjunto mínimo de elementos de rede que devem permanecer ativos para atender à demanda de tráfego e em seguida desliga o máximo possível de enlaces e comutadores, como visto no Algoritmo 5.4. O SimDC3D também simula a adaptação dinâmica das taxas dos enlaces (ALR - *Adaptive Link Rate*) descrita na Seção 4.3.2.4.

O traço utilizado é do SHARCNET e as características do centro de dados são as mesmas detalhadas na Seção 6.1.

Para este estudo de caso foram configurados os seguintes parâmetros: tempo de adaptação dinâmica ($RUNTIME_LINK_ADAPTATION=180s$), intervalo de execução da otimização de topologia ($PERIODIC_INTERVAL_EXECUTION_TOPOLOGY=600s$), valor de referência ($TRAFFIC_LIMIT_FOR_ADAPTATION=80%$), percentual de comunicação interna ($PROPORTION_OF_INTERNAL_COMMUNICATION_BETWEEN_VMS=30%$), percentual de comunicação externa ($PROPORTION_OF_EXTERNAL_COMMUNICATION=70%$). Os significados desses parâmetros estão detalhados na Seção 4.3.2.4.

Tabela 6.5 – Desempenho do centro de dados x estratégias de alocação de VMs para os traços SHARCNET e UNILU-GAIA.

	SHARCNET					UNILU-GAIA				
	FFD	PABFD	2D	3DMOBFD (a)	3DMOBFD (b)	FFD	PABFD	2D	3DMOBFD (a)	3DMOBFD (b)
Total de Migrações										
Sobrecarga	163.902	14.230	16.095	12.445	18.704	152.963	38.200	61.271	37.357	39.794
Predição	563.484	55.895	39.425	53.720	77.484	387.334	55.603	94.723	58.007	62.271
Baixa utilização	8.777	12.311	11.230	10.393	15.866	145.358	165.269	202.512	159.854	164.699
Altas temperaturas	189	307	263	202	193	170	216	230	248	194
Enlaces sobrecarregados	2	2	2	2	4	39	13	47	22	21
TOTAL	736.353 ± 26.830	82.745 ± 7.446	67.015 ± 3.516	76.761 ± 7.185	112.251 ± 7.083	685.865 ± 7.919	259.302 ± 19.764	358.784 ± 14.871	255.489 ± 15.578	266.979 ± 4.925
Consumo Energético (GW)										
Servidor	76,01	82,90	80,71	81,61	80,69	94,88	97,58	95,08	96,43	96,26
Processador	73,10	80,48	78,24	79,30	78,22	91,83	94,86	92,10	93,66	93,50
Ventoinhas	2,91	2,42	2,47	2,30	2,47	3,05	2,72	2,98	2,77	2,75
Sistema de refrigeração	19,43	19,90	19,70	20,17	19,66	16,14	16,42	15,86	16,16	16,15
Comutadores	3,41	3,41	3,41	3,41	3,41	15,02	15,02	15,02	15,02	15,02
TOTAL	98,85 ± 0,11	106,22 ± 0,40	103,83 ± 0,27	105,19 ± 0,05	103,76 ± 0,34	126,05 ± 0,08	129,02 ± 0,66	125,97 ± 0,06	127,61 ± 0,35	127,43 ± 0,12
Temperatura										
Total de chassis com sobrecarga térmica	5 ± 0	9 ± 1	9 ± 1	10 ± 1	9 ± 1	12 ± 1	12 ± 1	12 ± 1	13 ± 1	12 ± 1
Tempo médio com sobrecarga térmica por chassis (s)	66 ± 7	44 ± 7	56 ± 4	33 ± 6	36 ± 2	82 ± 5	80 ± 5	96 ± 5	70 ± 2	71 ± 3
Parâmetro dos Níveis de Serviço (%)										
Violações de SLA	0,37 ± 0,03	0,06 ± 0,01	0,13 ± 0,01	0,06 ± 0,01	0,06 ± 0,01	0,33 ± 0,01	0,13 ± 0,01	0,21 ± 0,00	0,16 ± 0,02	0,18 ± 0,02
Downtime	0,29 ± 0,08	0,06 ± 0,06	0,02 ± 0,00	0,02 ± 0,00	0,03 ± 0,00	0,10 ± 0,01	0,04 ± 0,00	0,05 ± 0,00	0,12 ± 0,13	0,08 ± 0,07
Fila de Escalonamento (s)										
Tempo médio	0,01 ± 0,00	0,01 ± 0,00	0,01 ± 0,00	0,00 ± 0,00	0,00 ± 0,00	0,06 ± 0,00	0,05 ± 0,00	0,05 ± 0,00	0,05 ± 0,00	0,05 ± 0,00
Desvio padrão	0,10	0,13	0,14	0,10	0,09	0,91	0,85	0,85	0,85	0,83
Máquinas Virtuais										
Número máximo de VMs simultâneas	11.781 ± 3	11.803 ± 4	11.778 ± 2	11.801 ± 5	11.807 ± 4	3.091 ± 2	3.094 ± 7	3.093 ± 4	3.085 ± 5	3.085 ± 3

A Tabela 6.7 apresenta o consumo energético da infraestrutura de rede sem qualquer otimização para cada uma das topologias com os algoritmos PABFD, 3DMOBFD(a) e 3DMOBFD(b).

A Tabela 6.8 mostra os resultados com a otimização habilitada para todas as topologias. A topologia de duas camadas oferece possibilidades de otimização nos enlaces redundantes entre os comutadores de acesso e núcleo e nos próprios comutadores de núcleo, que podem ser desligados. A Tabela 6.8 mostra que a redução do consumo energético nos comutadores de núcleo foi de 74,65%, pois o algoritmo de otimização reduziu de quatro para um os comutadores de núcleo. A economia total neste caso foi de 31,34% em comparação com a Tabela 6.7. Esse percentual aumenta para 40,33% quando a adaptação dinâmica nas velocidades dos enlaces (ALR) é habilitada, como mostra a Tabela 6.9. Neste caso, a ALR é responsável por 16,67% de economia.

Na topologia de três camadas, o consumo energético dos comutadores foi de 3,36 GW, como mostra a Tabela 6.7. A otimização de topologia produziu uma redução no consumo energético de 10,12%. A redução energética dos comutadores de núcleo foi de 50%, pois o algoritmo de otimização reduziu pela metade a quantidade desses comutadores na topologia. Com a ALR, a redução do consumo energético saltou para 22,92%, sendo que os comutadores

Tabela 6.6 – Desempenho do centro de dados x estratégias de alocação de VMs para o traço RICC.

	RICC				
	FFD	PABFD	2D	3DMOBFD (a)	3DMOBFD (b)
Total de Migrações					
Sobrecarga	37.352	13.870	27.945	13.985	14.795
Predição	78.154	34.316	45.792	33.295	34.667
Baixa utilização	18.986	15.116	17.607	12.781	12.920
Altas temperaturas	960	900	417	735	587
Enlaces sobrecarregados	1	1	3	1	4
TOTAL	135.454 ± 841	64.203 ± 395	91.763 ± 909	60.796 ± 880	62.973 ± 574
Consumo Energético (GW)					
Servidor	73,45	75,24	74,41	75,30	75,20
Processador	69,66	71,73	70,71	71,82	71,73
Ventoinhas	3,79	3,51	3,71	3,47	3,47
Sistema de refrigeração	21,23	21,44	20,90	21,37	21,39
Comutadores	2,22	2,22	2,22	2,22	2,22
TOTAL	96,91 ± 0,03	98,90 ± 0,10	97,53 ± 0,05	98,88 ± 0,07	98,81 ± 0,08
Temperatura					
Total de chassis com sobrecarga térmica	14 ± 1	13 ± 1	12 ± 1	13 ± 1	13 ± 1
Tempo médio com sobrecarga térmica por chassi (s)	153 ± 10	143 ± 11	154 ± 9	133 ± 7	124 ± 5
Parâmetro dos Níveis de Serviço (%)					
Violações de SLA	2,89 ± 0,06	1,55 ± 0,03	3,16 ± 0,06	1,46 ± 0,04	1,51 ± 0,04
<i>Downtime</i>	0,04 ± 0,00	0,02 ± 0,00	0,03 ± 0,00	0,02 ± 0,00	0,02 ± 0,00
Fila de Escalonamento (s)					
Tempo médio	889,69 ± 2,33	996,25 ± 3,70	914,59 ± 5,01	1.005,39 ± 2,09	995,52 ± 6,75
Desvio padrão	1.171,78	2.612,06	2.400,71	2.629,29	1.305,28
Máquinas Virtuais					
Número máximo de VMs simultâneas	5.913 ± 22	5.957 ± 24	6.013 ± 20	5.933 ± 14	5.956 ± 23

Tabela 6.7 – Consumo energético da infraestrutura de rede sem otimização.

	TOPOLOGIAS								
	2 Camadas			3 Camadas			3 Camadas com Enlaces Redundante		
	PABFD	3DMOBFD (a)	3DMOBFD (b)	PABFD	3DMOBFD (a)	3DMOBFD (b)	PABFD	3DMOBFD (a)	3DMOBFD (b)
Consumo Energético (GW)									
Servidor	82,86 ± 0,11	81,73 ± 0,11	80,64 ± 0,13	82,73 ± 0,06	81,62 ± 0,19	80,84 ± 0,15	82,90 ± 0,27	81,61 ± 0,03	80,69 ± 0,24
Sistema de Refrigeração	19,92 ± 0,09	20,30 ± 0,09	19,60 ± 0,04	19,83 ± 0,04	20,19 ± 0,11	19,71 ± 0,06	19,90 ± 0,13	20,17 ± 0,07	19,66 ± 0,12
Comutadores	3,67 ± 0,00	3,67 ± 0,00	3,67 ± 0,00	3,36 ± 0,00	3,36 ± 0,00	3,36 ± 0,00	3,41 ± 0,00	3,41 ± 0,00	3,41 ± 0,00
Acesso	2,25 ± 0,00	2,25 ± 0,00	2,25 ± 0,00	2,16 ± 0,00	2,16 ± 0,00	2,16 ± 0,00	2,19 ± 0,00	2,19 ± 0,00	2,19 ± 0,00
Agregação	-	-	-	0,53 ± 0,00	0,53 ± 0,00	0,53 ± 0,00	0,56 ± 0,00	0,56 ± 0,00	0,56 ± 0,00
Núcleo	1,42 ± 0,00	1,42 ± 0,00	1,42 ± 0,00	0,66 ± 0,00	0,66 ± 0,00	0,66 ± 0,00	0,66 ± 0,00	0,66 ± 0,00	0,66 ± 0,00
TOTAL	106,45 ± 0,12	105,70 ± 0,14	103,92 ± 0,16	105,92 ± 0,10	105,17 ± 0,26	103,91 ± 0,17	106,22 ± 0,40	105,19 ± 0,05	103,76 ± 0,34

Tabela 6.8 – Desempenho da infraestrutura de rede com otimização x estratégias de alocação de VMs.

	TOPOLOGIAS								
	2 Camadas			3 Camadas			Topologia 3 Camadas com Enlaces Redundante		
	PABFD	3DMOBFD (a)	3DMOBFD (b)	PABFD	3DMOBFD (a)	3DMOBFD (b)	PABFD	3DMOBFD (a)	3DMOBFD (b)
Total de Migrações									
Sobrecarga	15.815	12.791	19.479	12.920	12.189	19.123	12.758	12.746	19.189
Predição	62.643	55.375	75.338	49.952	51.064	78.232	48.911	54.672	75.315
Baixa utilização	12.767	10.716	15.853	12.753	11.102	15.477	12.748	11.025	16.300
Altas temperaturas	307	216	225	344	194	237	278	192	194
Enlaces sobrecarregados	1	2	3	2	3	1	1	3	1
TOTAL	91.532 ± 7.349	79.099 ± 6.535	110.898 ± 6.994	75.972 ± 1.894	74.552 ± 3.309	113.069 ± 4.793	74.696 ± 576	78.638 ± 3.616	110.998 ± 4.842
Consumo Energético (GW)									
Servidor	82,72	81,61	80,67	82,68	81,72	80,61	82,90	81,71	80,70
Sistema de Refrigeração	19,90	20,11	19,58	19,91	20,23	19,60	19,98	20,17	19,68
Comutadores	2,52	2,52	2,52	3,02	3,02	3,02	2,77	2,77	2,77
Acesso	2,16	2,16	2,16	2,16	2,16	2,16	2,16	2,16	2,16
Agregação	-	-	-	0,52	0,52	0,52	0,28	0,28	0,28
Núcleo	0,36	0,36	0,36	0,33	0,33	0,33	0,33	0,33	0,33
TOTAL	105,13 ± 0,06	104,23 ± 0,23	102,77 ± 0,23	105,60 ± 0,18	104,96 ± 0,16	103,22 ± 0,13	105,65 ± 0,09	104,65 ± 0,27	103,14 ± 0,22
Parâmetros dos Níveis de Serviço (%)									
Violações de SLA	0,07 ± 0,01	0,07 ± 0,00	0,07 ± 0,01	0,06 ± 0,01	0,06 ± 0,00	0,07 ± 0,00	0,06 ± 0,01	0,06 ± 0,01	0,07 ± 0,00
DownTime	0,07 ± 0,06	0,07 ± 0,06	0,03 ± 0,00	0,06 ± 0,06	0,06 ± 0,00	0,03 ± 0,00	0,02 ± 0,00	0,06 ± 0,06	0,03 ± 0,00
Fila de Escalonamento									
Tempo Médio (s)	0,01 ± 0,00	0,01 ± 0,00	0,00 ± 0,00	0,00 ± 0,00	0,00 ± 0,00	0,01 ± 0,00	0,01 ± 0,00	0,00 ± 0,00	0,01 ± 0,00
Desvio Padrão (s)	0,16	0,14	0,07	0,10	0,07	0,10	0,15	0,05	0,14

Tabela 6.9 – Desempenho da infraestrutura de rede com otimização e ALR x estratégias de alocação de VMs.

	TOPOLOGIAS								
	2 Camadas			3 Camadas			Topologia 3 Camadas com Enlaces Redundante		
	PABFD	3DMOBFD (a)	3DMOBFD (b)	PABFD	3DMOBFD (a)	3DMOBFD (b)	PABFD	3DMOBFD (a)	3DMOBFD (b)
Total de Migrações									
Sobrecarga	13.121	12.705	18.170	14.046	12.637	20.136	12.647	12.930	20.497
Predição	51.481	54.026	72.968	55.900	52.735	81.165	50.336	56.685	80.086
Baixa utilização	12.664	10.871	15.984	12.643	11.093	15.593	12.342	10.767	16.028
Altas temperaturas	323	237	262	271	224	240	262	218	180
Enlaces sobrecarregados	2	1	7	3	0	1	0	1	4
TOTAL	77.591	77.841	107.390	82.863	76.689	117.135	75.588	80.601	116.794
	± 3.300	± 7.780	± 8.391	± 7.531	± 5.213	± 10.682	± 2.199	± 7.953	± 5.592
Consumo Energético (GW)									
Servidor	82,83	81,65	80,79	82,76	81,53	80,65	82,95	81,69	80,63
Sistema de Refrigeração	19,91	20,21	19,61	19,89	20,18	19,55	19,86	20,27	19,58
Comutadores	2,10	2,10	2,10	2,59	2,59	2,59	2,36	2,36	2,36
Acesso	1,77	1,77	1,77	1,77	1,77	1,77	1,77	1,77	1,77
Agregação	-	-	-	0,50	0,50	0,50	0,27	0,27	0,27
Núcleo	0,33	0,33	0,33	0,32	0,32	0,32	0,32	0,32	0,32
TOTAL	104,84	103,95	102,49	105,24	104,30	102,79	105,17	104,32	102,57
	± 0,15	± 0,23	± 0,24	± 0,51	± 0,25	± 0,08	± 0,12	± 0,15	± 0,23
Parâmetros dos Níveis de Serviço (%)									
Violações de SLA	0,07	0,07	0,07	0,05	0,07	0,07	0,07	0,07	0,06
	± 0,01	± 0,00	± 0,00	± 0,00	± 0,01	± 0,01	± 0,00	± 0,00	± 0,01
DownTime	0,06	0,02	0,03	0,06	0,07	0,07	0,14	0,03	0,04
	± 0,06	± 0,00	± 0,00	± 0,06	± 0,07	± 0,07	± 0,08	± 0,00	± 0,00
Fila de Escalonamento									
Tempo Médio (s)	0,01	0,00	0,00	0,01	0,00	0,00	0,01	0,00	0,00
	± 0,00	± 0,00	± 0,00	± 0,00	± 0,00	± 0,00	± 0,00	± 0,00	± 0,00
Desvio Padrão (s)	0,16	0,03	0,10	0,18	0,11	0,07	0,16	0,05	0,10

de acesso são responsáveis pela maior parte da economia (18.06%). As estatísticas extraídas do SimDC3D indicam que as conexões entre os servidores e os comutadores de acesso ficaram entre as velocidades de 10 a 100Mbps, não sendo necessário enlaces de 1GB para o tráfego gerado.

Na topologia de três camadas com enlaces redundantes, há um aumento no consumo energético nos comutadores de acesso e agregação em comparação com a topologia de três camadas. Isso é provocado pelo acréscimo dos enlaces redundantes de *uplink* da camada de acesso para a camada de agregação. Como nessa topologia há dois caminhos interligando um comutador de acesso com outros dois comutadores de agregação, o algoritmo de otimização reduziu os comutadores de agregação pela metade. A redução total de energia dos comutadores nesta topologia foi de 18,77% e de 30,79% quando a ALR é habilitada, como mostra a Tabela 6.9.

Entre todas as topologias, a de duas camadas foi a que obteve a maior redução (40,33%) no consumo energético da infraestrutura de rede, seguido de 30,79% da topologia de três camadas com enlaces redundantes e 22,92% da topologia de três camadas.

As estratégias de alocação de máquinas virtuais não foram influenciadas pela topologia de rede e pelo uso de ALR, pois não foram observadas alterações significativas nos resultados produzidos pelos algoritmos PABFD, 3DMOBFD(a) e 3DMOBFD(b).

Tabela 6.10 – Desempenho do preditor de CPU.

	Sem Preditor CPU		Com Preditor CPU	
	3DMOBFD (a)	3DMOBFD (b)	3DMOBFD (a)	3DMOBFD (b)
Total de Migrações				
Sobrecarga	898.089	959.868	412.419	429.379
Predição	-	-	450.655	489.525
Baixa utilização	96.175	101.771	107.747	115.879
Altas temperaturas	1.684	1.797	1.869	1.735
Enlaces sobrecarregados	4	7	9	7
TOTAL	995.951 ± 10.938	1.063.443 ± 8.114	972.699 ± 15.890	1.036.525 ± 13.229
Consumo Energético (GW)				
Servidor	67,01 ± 0,02	66,91 ± 0,02	68,04 ± 0,02	67,92 ± 0,02
Sistema de Refrigeração	12,13 ± 0,01	12,16 ± 0,00	12,32 ± 0,01	12,34 ± 0,01
Comutadores	5,80 ± 0,00	5,80 ± 0,00	5,80 ± 0,00	5,80 ± 0,00
TOTAL	84,95 ± 0,03	84,87 ± 0,02	86,17 ± 0,02	86,06 ± 0,02
Parâmetro dos Níveis de Serviço (%)				
Violações de SLA	1,43 ± 0,01	1,47 ± 0,01	0,80 ± 0,01	0,81 ± 0,01
DownTime	0,27 ± 0,01	0,28 ± 0,00	0,26 ± 0,00	0,27 ± 0,00
Fila de Escalonamento				
Tempo Médio (s)	0,09 ± 0,00	0,10 ± 0,00	0,08 ± 0,00	0,08 ± 0,00
Desvio Padrão (s)	0,92	0,94	0,86	0,85

6.4.5 Cenário 5: Avaliação de Preditor de Uso de CPU na Prevenção de Sobrecargas

Esta seção avalia o desempenho do preditor de uso de CPU nos algoritmos de alocação de máquinas virtuais 3DMOBFD(a) e 3DMOBFD(b) utilizando o traço da LANL-CM5 com banco de 30 servidores, controle de temperatura dinâmico agressivo e sem otimizações de rede.

Neste experimento, o SimDC3D utiliza regressão linear para prever o comportamento futuro da utilização de CPU de um servidor. A análise de regressão linear estuda a relação entre uma variável dependente e outras variáveis independentes. A relação entre elas é representada por um modelo matemático que associa a variável dependente com as variáveis independentes. Esse

modelo é designado por modelo de regressão linear simples (MRLS) que define uma relação linear entre a variável dependente e uma variável independente.

O modelo de regressão linear simples é dado por:

$$Y = E(Y|X = x) + \epsilon = \alpha + \beta X + \epsilon \quad (6.1)$$

Em que:

Y : variável dependente aleatória. No SimDC3D, a variável Y indica a utilização de CPU de um servidor;

X : variável independente medida sem erro (não aleatória). No SimDC3D, a variável X indica tempo;

α : coeficiente de regressão, que representa o intercepto (parâmetro desconhecido do modelo (a estimar));

β : coeficiente de regressão, que representa o declive (parâmetro desconhecido do modelo (a estimar));

ϵ : erro aleatório ou estocástico, que inclui todas as influências no comportamento da variável Y que não podem ser explicadas linearmente pelo comportamento da variável X .

Uma aplicação importante de um modelo de regressão é a previsão de observações futuras de Y , ($Y_f(x)$) correspondente a um dado valor da variável X , x_f , no qual o estimador é dado por:

$$\hat{Y}_f = \hat{y}_f = \hat{\alpha} + \hat{\beta}x_f \quad (6.2)$$

Os principais indicadores de eficiência do preditor de uso de CPU são o SLA e o número de migrações por sobrecarga de CPU. A migração por sobrecarga ocorre no momento que a utilização de CPU de um servidor ultrapassa o valor estipulado no parâmetro *THRESHOLD_TOP_USE_OF_CPU*, que foi definido em todas as simulações em 90%.

A Tabela 6.10 mostra os resultados dos algoritmos 3DMOBFDA(a) e 3DMOBFDA(b) com e sem o preditor de uso de CPU. Quando o preditor está desabilitado, há uma grande quantidade de migrações por sobrecarga de CPU e um aumento nas violações de SLA. Porém, há uma redução no consumo energético em comparação com as simulações com o preditor habilitado. Essa redução no consumo energético ocorre devido à redução no número de servidores ativos, gerando, como consequência, sobrecargas nesses servidores.

Ao habilitar o preditor de uso de CPU, há redução no número de migrações por sobrecarga de CPU, além de uma diminuição significativa nas violações de SLA no centro de dados em mais de 44%, como pode ser visto na Tabela 6.10. As simulações com preditores deixam o centro de dados estável, melhorando a previsibilidade de sobrecarga de CPU. Isso reflete nos indicadores de qualidade, porém exige uma quantidade maior de servidores para lidar com

a carga de processamento que as VMs exigem. Dessa forma, há um aumento no consumo energético dos servidores e consequentemente no sistema de refrigeração.

A utilização do preditor de uso de CPU é fundamental para auxiliar na redução das sobrecargas de CPU nos servidores do centro de dados e para manter os indicadores SLA e número de migrações por sobrecarga dentro de valores aceitáveis. O SimDC3D permite que outras técnicas de predição sejam implementadas estendendo a classe *PredictionAlgorithms*.

6.4.6 Cenário 6: Avaliação das Estratégias de Predição do SimDC3D

Para examinar os resultados e a eficácia dos preditores, o SimDC3D fornece dados estatísticos, como raiz do erro quadrático médio, média de erro, variância, desvio padrão e coeficiente de variação dos valores preditos em relação aos valores reais. Esses dados são calculados e apresentados por servidor ao final de cada simulação.

Um conjunto de parâmetros auxilia na configuração dos dois algoritmos de predição. Os parâmetros permitem definir desde o tamanho da janela para armazenar o histórico da variável, até parâmetros específicos como dimensão, atraso, número de centros e ordem polinomial, que são específicos dos dois algoritmos.

No arquivo de configuração, o parâmetro *VARIABLE_PREDICTED* define qual variável será predita, dentro de um conjunto de variáveis possíveis.

Devido à grande quantidade de servidores, são apresentados os resultados de somente 250 servidores, o que equivale a 50% do total de equipamentos do centro de dados simulado, e do traço da SHARCNET. A seguir, serão apresentados os resultados de predição da variável temperatura. Essa variável foi utilizada no escalonamento de máquinas virtuais pelo algoritmo 3DMOBF(b).

A Tabela 6.11 apresenta a raiz do erro quadrático médio (RMSE) de 250 servidores. O algoritmo de predição configurado é o *RBF* e a variável a ser predita é a temperatura do servidor. O tamanho da janela foi parametrizado em 1.200 elementos, com coleta de uma amostra a cada 60 segundos. O parâmetro *CLENGTH* determina o intervalo de previsão e foi fixado em cinco. Consequentemente, o algoritmo de escalonamento utilizará a temperatura prevista para cinco minutos no futuro (*CLENGTH* * tempo coleta).

O servidor que obteve o melhor resultado foi o servidor 1 do chassi 24 (0,016) e o maior erro quadrático médio observado foi no servidor 10 do chassi 21 (1,912). Alguns servidores não tiveram valores suficientes para calcular a raiz do erro quadrático médio, desta forma as células foram preenchidas com o símbolo (-). A média da RMSE entre todos os 250 servidores foi de 0,60, com um desvio padrão de 0,36.

Os resultados obtidos indicam que o preditor utilizado tem uma acurácia alta na variável temperatura, como pode ser observado nas Tabelas 6.11 e 6.12.

Tabela 6.11 – Raiz do erro quadrático médio da variável temperatura.

	Servidor 1	Servidor 2	Servidor 3	Servidor 4	Servidor 5	Servidor 6	Servidor 7	Servidor 8	Servidor 9	Servidor 10
Chassi 0	1,077	1,117	1,029	0,571	0,559	0,908	0,532	1,110	0,542	1,247
Chassi 1	0,534	0,566	0,828	0,630	0,706	0,600	0,657	0,528	0,588	1,060
Chassi 2	0,497	0,881	0,481	1,318	0,512	0,489	0,335	0,602	0,477	0,483
Chassi 3	1,384	0,262	0,147	0,278	0,250	0,238	0,241	0,249	0,145	0,261
Chassi 4	0,436	0,156	0,144	0,144	0,166	0,266	0,171	0,155	0,263	0,258
Chassi 5	0,805	0,576	0,544	0,785	0,851	0,854	-	0,641	0,853	0,579
Chassi 6	1,059	1,066	0,592	0,542	0,649	0,917	0,811	0,945	0,635	0,957
Chassi 7	0,858	1,145	0,570	0,800	0,806	0,597	0,822	0,650	0,640	0,898
Chassi 8	0,243	0,447	0,428	0,639	0,430	0,623	0,355	0,624	0,450	0,600
Chassi 9	0,341	0,215	0,131	0,150	0,192	0,439	0,156	0,143	0,186	0,184
Chassi 10	0,623	0,997	0,667	0,584	0,757	0,929	0,999	0,916	0,605	0,617
Chassi 11	0,601	0,646	0,818	0,606	0,855	0,774	0,864	0,570	0,696	0,587
Chassi 12	1,083	1,412	1,232	0,909	0,967	1,217	1,475	1,116	1,272	1,021
Chassi 13	0,504	0,169	0,196	0,178	0,190	0,199	-	0,161	0,185	0,386
Chassi 14	0,172	0,180	0,050	0,097	0,164	0,052	1,630	0,165	0,131	-
Chassi 15	1,194	0,664	1,231	1,354	1,222	1,181	1,140	1,220	0,645	1,086
Chassi 16	0,620	0,824	0,594	0,717	0,831	0,569	0,857	0,657	0,579	0,692
Chassi 17	0,569	0,776	0,652	0,663	0,749	0,572	0,586	0,600	0,662	0,639
Chassi 18	0,455	0,308	0,332	0,326	0,366	0,527	0,485	0,467	0,319	0,480
Chassi 19	0,550	0,478	0,438	0,119	0,636	0,535	0,683	0,124	0,109	0,129
Chassi 20	0,631	0,573	0,535	0,555	0,634	0,693	0,641	0,673	0,825	0,738
Chassi 21	1,794	0,605	0,700	1,641	1,444	0,591	0,605	0,618	0,654	1,912
Chassi 22	0,420	1,249	0,507	1,199	0,449	0,486	0,403	0,445	0,483	0,427
Chassi 23	0,155	0,162	0,140	0,158	0,630	0,639	0,115	0,134	0,139	-
Chassi 24	0,016	0,222	0,221	0,136	0,096	-	-	0,222	0,101	0,079

6.4.6.1 Erro Médio e Desvio Padrão

Para calcular o erro médio, o SimDC3D armazena o valor predito de um determinado servidor e aguarda o avanço do tempo até o ponto que o valor predito e real estão no mesmo tempo da simulação. O tempo que será aguardado para coletar o valor real simulado depende da multiplicação de dois parâmetros, *CLENGTH* e *MONITORINGTIME*. Após a coleta, é calculada a razão entre os dois valores e o resultado é armazenado para que se possa calcular o erro médio do servidor ao final da simulação.

As Tabelas 6.12 e 6.13 mostram as estatísticas de erro médio e desvio padrão de cada servidor. O servidor que obteve o menor erro médio foi o servidor 01 do chassi 24, com 0,04% e desvio padrão de 0,0004. O maior erro médio foi o do servidor 10 do chassi 21, com 2,58% e desvio padrão de 0,0748.

A Tabela 6.14 exhibe os resultados do coeficiente de variação (C_v) referente ao preditor RBF. Por exemplo, no servidor 0 do chassi 0, o coeficiente de variação atinge 197% do valor da média. Já no servidor 6 do chassi 13, a relação foi de 99%. O menor C_v é de 73% no servidor 3 do chassi 14 e o maior C_v foi de 401% no servidor 7 do chassi 14.

6.4.6.2 Desempenho dos Preditores *RBF* e *Polynom*

Na seção anterior, foram discutidos os resultados estatísticos da utilização do algoritmo de predição *RBF* com o parâmetro *CLENGTH* igual a cinco.

Para compreender a eficácia dos algoritmos *RBF* e *Polynom* e a influência do parâmetros *CLENGTH* sobre os resultados, foi realizado um conjunto específico de simulações em

Tabela 6.12 – Erro médio dos valores preditos em relação ao real da variável temperatura.

	Servidor 1	Servidor 2	Servidor 3	Servidor 4	Servidor 5	Servidor 6	Servidor 7	Servidor 8	Servidor 9	Servidor 10
Chassi 0	2,14%	2,07%	1,87%	1,55%	1,57%	1,84%	1,53%	2,01%	1,61%	2,31%
Chassi 1	1,59%	1,64%	1,67%	1,88%	1,91%	1,73%	2,03%	1,45%	1,69%	2,07%
Chassi 2	1,34%	1,87%	1,34%	2,14%	1,38%	1,26%	0,70%	1,51%	1,35%	1,31%
Chassi 3	1,79%	0,66%	0,37%	0,72%	0,66%	0,58%	0,61%	0,61%	0,36%	0,65%
Chassi 4	0,32%	0,37%	0,32%	0,34%	0,39%	0,49%	0,42%	0,37%	0,47%	0,49%
Chassi 5	1,85%	1,71%	1,61%	1,87%	1,96%	1,96%	-	1,90%	1,97%	1,57%
Chassi 6	2,20%	2,08%	1,72%	1,59%	1,98%	1,99%	1,84%	2,08%	1,56%	2,12%
Chassi 7	1,79%	1,92%	1,56%	1,65%	1,78%	1,74%	1,87%	1,88%	1,93%	2,01%
Chassi 8	0,73%	1,03%	1,20%	1,39%	1,21%	1,32%	1,00%	1,36%	1,31%	1,34%
Chassi 9	0,56%	0,42%	0,30%	0,36%	0,45%	0,71%	0,36%	0,33%	0,40%	0,40%
Chassi 10	1,86%	2,06%	1,72%	1,72%	1,91%	2,03%	2,16%	2,17%	1,81%	1,82%
Chassi 11	1,75%	1,94%	1,86%	1,71%	1,87%	1,76%	1,98%	1,56%	1,48%	1,67%
Chassi 12	1,78%	1,91%	2,03%	1,40%	1,62%	1,91%	2,08%	1,94%	2,07%	1,54%
Chassi 13	0,91%	0,44%	0,53%	0,42%	0,51%	0,55%	-	0,43%	0,45%	0,82%
Chassi 14	0,43%	0,39%	0,14%	0,19%	0,39%	0,13%	1,35%	0,39%	0,27%	-
Chassi 15	2,35%	1,97%	2,40%	2,34%	2,46%	2,11%	2,22%	2,21%	1,66%	2,07%
Chassi 16	1,77%	1,97%	1,83%	2,22%	1,95%	1,74%	2,05%	1,98%	1,73%	2,17%
Chassi 17	1,62%	1,75%	1,64%	1,65%	1,65%	1,41%	1,70%	1,59%	1,91%	1,84%
Chassi 18	0,97%	0,79%	0,88%	0,89%	1,04%	1,12%	1,02%	1,05%	0,81%	1,00%
Chassi 19	0,61%	0,56%	0,53%	0,29%	0,73%	0,63%	0,72%	0,29%	0,28%	0,31%
Chassi 20	1,81%	1,64%	1,54%	1,34%	1,79%	2,14%	1,88%	1,45%	1,88%	1,48%
Chassi 21	2,53%	1,51%	1,82%	2,45%	2,15%	1,48%	1,72%	1,72%	1,63%	2,58%
Chassi 22	1,15%	1,65%	1,44%	1,54%	1,11%	1,35%	1,06%	1,14%	1,34%	1,19%
Chassi 23	0,38%	0,38%	0,33%	0,37%	0,75%	0,79%	0,29%	0,34%	0,33%	-
Chassi 24	0,04%	0,41%	0,42%	0,29%	0,20%	-	-	0,41%	0,22%	0,16%

Tabela 6.13 – Desvio padrão dos valores preditos em relação ao real da variável temperatura.

	Servidor 1	Servidor 2	Servidor 3	Servidor 4	Servidor 5	Servidor 6	Servidor 7	Servidor 8	Servidor 9	Servidor 10
Chassi 0	0,0423	0,0441	0,0410	0,0246	0,0238	0,0358	0,0222	0,0439	0,0223	0,0492
Chassi 1	0,0204	0,0220	0,0321	0,0242	0,0279	0,0235	0,0247	0,0205	0,0231	0,0402
Chassi 2	0,0186	0,0342	0,0175	0,0494	0,0191	0,0187	0,0131	0,0225	0,0172	0,0179
Chassi 3	0,0472	0,0076	0,0043	0,0079	0,0069	0,0071	0,0071	0,0075	0,0043	0,0077
Chassi 4	0,0038	0,0039	0,0039	0,0037	0,0042	0,0078	0,0042	0,0039	0,0078	0,0074
Chassi 5	0,0316	0,0245	0,0232	0,0304	0,0336	0,0340	-	0,0274	0,0337	0,0257
Chassi 6	0,0421	0,0427	0,0256	0,0232	0,0274	0,0364	0,0357	0,0376	0,0273	0,0376
Chassi 7	0,0327	0,0470	0,0242	0,0308	0,0312	0,0245	0,0316	0,0269	0,0259	0,0343
Chassi 8	0,0079	0,0165	0,0148	0,0221	0,0149	0,0217	0,0122	0,0215	0,0152	0,0205
Chassi 9	0,0104	0,0063	0,0035	0,0038	0,0051	0,0133	0,0041	0,0038	0,0052	0,0050
Chassi 10	0,0265	0,0403	0,0287	0,0250	0,0318	0,0365	0,0395	0,0355	0,0256	0,0264
Chassi 11	0,0238	0,0251	0,0305	0,0244	0,0322	0,0323	0,0324	0,0232	0,0268	0,0235
Chassi 12	0,0440	0,0541	0,0500	0,0372	0,0392	0,0498	0,0567	0,0442	0,0518	0,0420
Chassi 13	0,0164	0,0050	0,0055	0,0056	0,0054	0,0055	-	0,0046	0,0056	0,0127
Chassi 14	0,0041	0,0047	0,0010	0,0028	0,0040	0,0013	0,0540	0,0041	0,0036	-
Chassi 15	0,0476	0,0284	0,0496	0,0551	0,0489	0,0484	0,0458	0,0491	0,0282	0,0439
Chassi 16	0,0264	0,0318	0,0242	0,0294	0,0323	0,0232	0,0329	0,0273	0,0239	0,0281
Chassi 17	0,0219	0,0283	0,0252	0,0264	0,0287	0,0217	0,0224	0,0240	0,0254	0,0246
Chassi 18	0,0156	0,0107	0,0113	0,0109	0,0119	0,0180	0,0166	0,0164	0,0112	0,0165
Chassi 19	0,0179	0,0154	0,0141	0,0030	0,0206	0,0173	0,0223	0,0032	0,0026	0,0032
Chassi 20	0,0264	0,0239	0,0222	0,0245	0,0267	0,0278	0,0266	0,0266	0,0320	0,0286
Chassi 21	0,0697	0,0248	0,0282	0,0634	0,0565	0,0241	0,0234	0,0242	0,0269	0,0748
Chassi 22	0,0144	0,0464	0,0170	0,0448	0,0157	0,0165	0,0137	0,0159	0,0164	0,0143
Chassi 23	0,0041	0,0044	0,0039	0,0043	0,0209	0,0211	0,0030	0,0035	0,0038	-
Chassi 24	0,0004	0,0063	0,0062	0,0035	0,0026	-	-	0,0063	0,0026	0,0021

Tabela 6.14 – Coeficiente de variação dos valores preditos em relação ao real da variável temperatura.

	Servidor 1	Servidor 2	Servidor 3	Servidor 4	Servidor 5	Servidor 6	Servidor 7	Servidor 8	Servidor 9	Servidor 10
Chassi 0	1,974	2,133	2,190	1,589	1,520	1,951	1,448	2,180	1,383	2,127
Chassi 1	1,287	1,339	1,921	1,289	1,463	1,360	1,219	1,418	1,364	1,945
Chassi 2	1,388	1,831	1,309	2,307	1,382	1,484	1,870	1,489	1,277	1,364
Chassi 3	2,644	1,151	1,166	1,102	1,053	1,213	1,166	1,228	1,184	1,182
Chassi 4	1,186	1,048	1,203	1,082	1,074	1,581	1,011	1,054	1,658	1,490
Chassi 5	1,713	1,429	1,442	1,624	1,718	1,740	-	1,447	1,713	1,641
Chassi 6	1,917	2,053	1,488	1,462	1,387	1,827	1,938	1,805	1,752	1,774
Chassi 7	1,824	2,452	1,554	1,874	1,751	1,412	1,688	1,427	1,342	1,706
Chassi 8	1,085	1,603	1,233	1,590	1,227	1,642	1,224	1,582	1,161	1,529
Chassi 9	1,852	1,491	1,192	1,059	1,133	1,885	1,152	1,153	1,302	1,246
Chassi 10	1,426	1,963	1,663	1,452	1,660	1,802	1,825	1,635	1,417	1,456
Chassi 11	1,362	1,293	1,637	1,429	1,727	1,838	1,634	1,487	1,810	1,405
Chassi 12	2,476	2,833	2,465	2,659	2,413	2,612	2,722	2,276	2,498	2,730
Chassi 13	1,814	1,136	1,038	1,335	1,072	0,989	-	1,084	1,256	1,544
Chassi 14	0,956	1,209	0,733	1,433	1,034	0,995	4,011	1,031	1,308	-
Chassi 15	2,023	1,438	2,068	2,357	1,989	2,289	2,059	2,220	1,705	2,120
Chassi 16	1,495	1,616	1,322	1,325	1,656	1,336	1,608	1,379	1,379	1,295
Chassi 17	1,349	1,621	1,537	1,599	1,741	1,536	1,318	1,504	1,327	1,338
Chassi 18	1,613	1,355	1,281	1,222	1,151	1,606	1,628	1,563	1,382	1,649
Chassi 19	2,937	2,736	2,665	1,013	2,806	2,753	3,106	1,094	0,915	1,022
Chassi 20	1,460	1,460	1,447	1,830	1,496	1,298	1,416	1,830	1,704	1,932
Chassi 21	2,759	1,649	1,555	2,585	2,630	1,629	1,363	1,403	1,649	2,901
Chassi 22	1,253	2,804	1,179	2,907	1,420	1,224	1,295	1,390	1,229	1,196
Chassi 23	1,070	1,152	1,186	1,148	2,788	2,675	1,068	1,034	1,151	-
Chassi 24	0,914	1,527	1,484	1,208	1,330	-	-	1,529	1,209	1,264

que pretende-se demonstrar o comportamento dos preditores aumentando o tempo de previsibilidade da variável temperatura.

As Tabelas 6.15 a 6.18 mostram os resultados variando o parâmetro *CLENGTH* no intervalo de 5 a 20, com o tempo de coleta a cada 60 segundos. Como a previsibilidade futura é dada por $CLENGTH*60$, o menor tempo futuro é 5 minutos e o maior 20 minutos.

A Tabela 6.15 exhibe o RMSE das simulações, as Tabelas 6.16 e 6.17 mostram os erros médios e o desvio padrão e, por último, a Tabela 6.18 mostra o coeficiente de variação. Esses resultados mostram que o algoritmo *Polynom* obteve resultados melhores do que o algoritmo *RBF* para a variável temperatura.

A raiz do erro quadrático médio demonstra uma precisão melhor para um tempo futuro maior. O algoritmo *Polynom* com o parâmetro *CLENGTH* igual a 20 tem menor RMSE em comparação ao algoritmo *RBF*. Essa comparação é mais perceptível analisando-se a Tabela 6.16, com os valores do erro médio e a Tabela 6.17, com o desvio padrão. As diferenças se tornam mais evidentes com os valores 15 e 20 para *CLENGTH*. O algoritmo *Polynom* mantém uma variação menor do erro médio conforme aumenta-se o tempo futuro que deverá ser predito.

O coeficiente de variação, independente do valor utilizado no *CLENGTH*, continua demonstrando uma grande variabilidade na comparação entre os dois algoritmos. Na maioria dos casos, o algoritmo *Polynom* apresenta um menor coeficiente de variação em relação ao algoritmo *RBF* como pode ser observado na tabela.

Com as estatísticas disponíveis no SimDC3D, é possível analisar o comportamento dos preditores em relação a uma variável a ser predita. Nesta seção, procurou-se evidenciar as estatísticas geradas pelo SimDC3D e a eficácia dos dois algoritmos de predição na predição

Tabela 6.15 – Raiz do erro quadrático médio da variável temperatura para CLENGTH de 5 a 20 para os algoritmos RBF e Polynom.

	RBF				POLYNOM			
	5	10	15	20	5	10	15	20
Chassi 0 - 1	1,077	0,622	0,708	2,262	0,896	0,643	0,712	0,959
Chassi 0 - 2	1,117	1,185	1,816	0,869	0,543	1,004	1,159	1,019
Chassi 0 - 3	1,029	1,104	1,518	0,869	0,842	0,687	1,122	0,903
Chassi 0 - 4	0,571	1,075	1,235	0,852	0,944	1,035	0,933	0,921
Chassi 0 - 5	0,559	0,524	0,656	0,864	0,886	0,653	1,086	0,737
Chassi 5 - 6	0,854	1,388	0,764	0,854	0,905	0,710	0,759	0,949
Chassi 5 - 7	0,000	1,127	1,855	0,781	0,943	0,618	0,802	0,857
Chassi 5 - 8	0,641	0,701	0,702	1,637	0,929	0,861	0,860	0,763
Chassi 5 - 9	0,853	0,759	1,192	0,847	0,713	0,928	0,794	0,815
Chassi 5 - 10	0,579	1,042	0,757	2,059	0,976	0,933	0,864	0,876
Chassi 10 - 1	0,623	2,035	0,647	0,864	0,700	0,678	0,910	0,852
Chassi 10 - 2	0,997	0,725	0,741	1,729	0,704	0,630	0,985	0,982
Chassi 10 - 3	0,667	1,564	1,837	0,769	0,647	0,677	0,834	0,891
Chassi 10 - 4	0,584	0,593	1,603	0,000	0,953	0,911	0,883	0,851
Chassi 10 - 5	0,757	0,681	0,131	0,856	0,725	0,899	0,717	0,937
Chassi 15 - 6	1,181	0,830	1,764	0,637	0,866	0,618	0,909	1,005
Chassi 15 - 7	1,140	1,086	0,723	0,667	0,923	0,699	0,725	1,049
Chassi 15 - 8	1,220	1,209	1,588	1,600	0,705	0,000	0,771	1,036
Chassi 15 - 9	0,645	1,202	0,619	1,338	0,822	0,856	0,844	0,807
Chassi 15 - 10	1,086	1,224	1,700	1,461	0,925	0,642	0,701	0,748
Chassi 20 - 1	0,631	1,057	1,473	1,072	0,902	0,642	0,795	0,911
Chassi 20 - 2	0,573	1,385	1,079	2,211	0,908	0,676	1,014	0,777
Chassi 20 - 3	0,535	1,066	0,755	0,891	0,575	0,667	0,739	0,964
Chassi 20 - 4	0,555	0,696	0,744	0,910	0,902	0,806	0,747	0,899
Chassi 20 - 5	0,634	0,735	0,965	2,218	0,714	0,848	0,763	0,936
Chassi 25 - 6	0,640	1,034	0,633	2,101	0,641	0,937	0,689	0,925
Chassi 25 - 7	0,882	0,920	0,747	0,839	0,650	0,707	0,000	0,725
Chassi 25 - 8	0,563	0,719	0,569	2,420	0,649	0,603	1,224	0,837
Chassi 25 - 9	0,630	0,989	1,249	2,130	0,964	0,670	0,811	0,713
Chassi 25 - 10	0,578	0,642	0,620	0,776	0,966	1,020	0,765	0,839

da variável temperatura. A temperatura predita foi utilizada nos algoritmos de escalonamento de máquinas virtuais. Neste caso, o escalonador calcula o ranking dos servidores utilizando a temperatura futura e não mais a temperatura atual do servidor.

Tabela 6.16 – Erro médio da variável temperatura para CLENGTH de 5 a 20 para os algoritmos RBF e Polynom.

	RBF				POLYNOM			
	5	10	15	20	5	10	15	20
Chassi 0 - 1	2,14%	2,01%	2,46%	4,14%	2,02%	1,79%	2,37%	2,66%
Chassi 0 - 2	2,07%	2,61%	3,65%	2,89%	1,49%	2,43%	2,72%	2,86%
Chassi 0 - 3	1,87%	2,29%	3,36%	2,83%	1,80%	2,08%	2,56%	2,35%
Chassi 0 - 4	1,55%	2,47%	2,49%	2,77%	1,99%	2,49%	2,22%	2,50%
Chassi 0 - 5	1,57%	1,58%	2,21%	2,86%	2,05%	1,99%	2,51%	2,36%
Chassi 5 - 6	1,96%	3,00%	2,71%	2,56%	1,99%	2,19%	2,57%	2,59%
Chassi 5 - 7	0,00%	2,41%	3,77%	2,18%	2,19%	1,80%	2,65%	2,88%
Chassi 5 - 8	1,90%	2,31%	2,25%	3,61%	2,13%	2,17%	2,27%	2,40%
Chassi 5 - 9	1,97%	2,51%	2,63%	2,50%	2,24%	2,38%	2,59%	2,66%
Chassi 5 - 10	1,57%	2,30%	2,65%	4,18%	2,19%	2,39%	2,31%	3,05%
Chassi 10 - 1	1,86%	3,84%	2,20%	2,74%	1,56%	2,15%	2,45%	3,13%
Chassi 10 - 2	2,06%	2,03%	2,59%	3,84%	2,15%	1,99%	2,50%	2,77%
Chassi 10 - 3	1,72%	2,95%	3,62%	2,41%	1,87%	2,15%	2,88%	2,08%
Chassi 10 - 4	1,72%	1,87%	3,91%	0,00%	2,21%	2,29%	2,18%	2,87%
Chassi 10 - 5	1,91%	2,31%	0,61%	2,66%	2,25%	2,26%	2,25%	3,02%
Chassi 15 - 6	2,11%	2,12%	3,39%	2,34%	1,94%	1,75%	2,12%	2,86%
Chassi 15 - 7	2,22%	3,06%	2,45%	2,51%	2,06%	2,21%	2,26%	2,96%
Chassi 15 - 8	2,21%	2,71%	3,47%	3,65%	2,23%	0,00%	2,56%	2,98%
Chassi 15 - 9	1,66%	2,59%	2,10%	2,92%	1,89%	2,17%	2,91%	2,56%
Chassi 15 - 10	2,07%	2,69%	3,72%	3,17%	2,12%	1,92%	2,27%	2,54%
Chassi 20 - 1	1,81%	2,32%	3,60%	2,97%	1,99%	2,02%	2,64%	2,22%
Chassi 20 - 2	1,64%	3,18%	2,75%	4,02%	2,05%	1,94%	2,25%	2,44%
Chassi 20 - 3	1,54%	2,31%	2,57%	2,60%	1,60%	2,08%	2,29%	2,64%
Chassi 20 - 4	1,34%	2,23%	2,46%	2,84%	2,04%	1,93%	2,30%	2,32%
Chassi 20 - 5	1,79%	2,59%	2,27%	4,03%	1,86%	2,08%	2,42%	2,53%
Chassi 25 - 6	1,83%	2,58%	2,12%	3,91%	1,84%	2,21%	2,06%	2,47%
Chassi 25 - 7	2,04%	2,22%	2,59%	2,67%	1,90%	1,87%	0,00%	2,36%
Chassi 25 - 8	1,63%	2,30%	1,68%	4,37%	1,85%	1,73%	3,55%	2,18%
Chassi 25 - 9	1,79%	2,39%	2,98%	3,98%	2,13%	2,07%	2,76%	2,25%
Chassi 25 - 10	1,65%	2,02%	2,07%	2,15%	2,22%	2,46%	2,40%	2,66%

6.4.7 Cenário 7: Centro de Dados Tradicional versus Centro de Dados Verde

Esta seção compara os resultados de economia de energia de um centro de dados tradicional, no qual não há qualquer tipo de otimização, com um centro de dados verde, que utiliza diversos mecanismos de otimização em todas as dimensões utilizando o traço da SHARCNET.

No centro de dados tradicional, as máquinas virtuais são escalonadas em um servidor e permanecem até o final da execução. O escalonador utilizado é o 3DMOBF(b), o sistema

Tabela 6.17 – Desvio padrão da variável temperatura para CLENGTH de 5 a 20 para os algoritmos RBF e Polynom.

	RBF				POLYNOM			
	5	10	15	20	5	10	15	20
Chassi 0 - 1	0,0423	0,0246	0,0262	0,0871	0,0348	0,0261	0,0271	0,0323
Chassi 0 - 2	0,0441	0,0458	0,0685	0,0334	0,0231	0,0372	0,0437	0,0339
Chassi 0 - 3	0,0410	0,0437	0,0557	0,0338	0,0331	0,0276	0,0425	0,0352
Chassi 0 - 4	0,0246	0,0414	0,0480	0,0332	0,0358	0,0382	0,0358	0,0313
Chassi 0 - 5	0,0238	0,0211	0,0248	0,0333	0,0337	0,0267	0,0411	0,0290
Chassi 5 - 6	0,0340	0,0611	0,0287	0,0355	0,0359	0,0291	0,0296	0,0305
Chassi 5 - 7	0,0000	0,0445	0,0703	0,0340	0,0363	0,0260	0,0322	0,0341
Chassi 5 - 8	0,0274	0,0279	0,0285	0,0635	0,0364	0,0324	0,0337	0,0311
Chassi 5 - 9	0,0337	0,0306	0,0509	0,0357	0,0295	0,0349	0,0322	0,0330
Chassi 5 - 10	0,0257	0,0413	0,0289	0,0788	0,0379	0,0347	0,0339	0,0339
Chassi 10 - 1	0,0265	0,0938	0,0251	0,0353	0,0310	0,0279	0,0348	0,0304
Chassi 10 - 2	0,0403	0,0295	0,0282	0,0657	0,0296	0,0257	0,0372	0,0330
Chassi 10 - 3	0,0287	0,0627	0,0717	0,0315	0,0280	0,0279	0,0324	0,0337
Chassi 10 - 4	0,0250	0,0242	0,0735	0,0000	0,0369	0,0338	0,0339	0,0337
Chassi 10 - 5	0,0318	0,0263	0,0030	0,0353	0,0302	0,0326	0,0291	0,0363
Chassi 15 - 6	0,0484	0,0341	0,0686	0,0229	0,0341	0,0262	0,0342	0,0333
Chassi 15 - 7	0,0458	0,0477	0,0282	0,0235	0,0363	0,0289	0,0298	0,0351
Chassi 15 - 8	0,0491	0,0487	0,0595	0,0611	0,0290	0,0000	0,0303	0,0345
Chassi 15 - 9	0,0282	0,0486	0,0240	0,0510	0,0327	0,0325	0,0328	0,0333
Chassi 15 - 10	0,0439	0,0489	0,0636	0,0553	0,0364	0,0274	0,0281	0,0291
Chassi 20 - 1	0,0264	0,0409	0,0621	0,0454	0,0347	0,0250	0,0305	0,0364
Chassi 20 - 2	0,0239	0,0585	0,0396	0,0850	0,0348	0,0269	0,0391	0,0309
Chassi 20 - 3	0,0222	0,0411	0,0281	0,0367	0,0242	0,0267	0,0297	0,0327
Chassi 20 - 4	0,0245	0,0274	0,0284	0,0362	0,0344	0,0304	0,0301	0,0347
Chassi 20 - 5	0,0267	0,0269	0,0363	0,0841	0,0296	0,0319	0,0302	0,0318
Chassi 25 - 6	0,0267	0,0381	0,0237	0,0800	0,0268	0,0353	0,0276	0,0341
Chassi 25 - 7	0,0341	0,0347	0,0274	0,0328	0,0270	0,0279	0,0000	0,0279
Chassi 25 - 8	0,0232	0,0283	0,0228	0,1094	0,0272	0,0252	0,0479	0,0316
Chassi 25 - 9	0,0265	0,0369	0,0444	0,0808	0,0368	0,0271	0,0297	0,0280
Chassi 25 - 10	0,0241	0,0253	0,0233	0,0313	0,0365	0,0377	0,0306	0,0333

de refrigeração fornece ar a 19°C e a topologia de rede é a de três camadas com enlaces redundantes. O SimDC3D foi parametrizado para não executar a adaptação das velocidades dos enlaces e nem a otimização da topologia da rede.

No centro de dados verde foram configurados o algoritmo 3DMOBF(b) para escalonamento e otimização de máquinas virtuais, o controle de temperatura é dinâmico (agressivo) e, por último, foram habilitadas a otimização da topologia da rede e a ALR.

Como mostra a Tabela 6.19, um centro de dados tradicional que não executa qualquer otimização tem um consumo energético 99,53% maior em relação a um centro de dados verde. Aplicando técnicas simples, como adaptação dos enlaces e controle do termostato, juntamente

Tabela 6.18 – Coeficiente de variação da variável temperatura para CLENGTH de 5 a 20 para os algoritmos RBF e Polynom.

	RBF				POLYNOM			
	5	10	15	20	5	10	15	20
Chassi 0 - 1	1,974	1,225	1,064	2,105	1,726	1,457	1,142	1,213
Chassi 0 - 2	2,133	1,753	1,876	1,155	1,554	1,532	1,606	1,183
Chassi 0 - 3	2,190	1,913	1,660	1,196	1,841	1,328	1,662	1,500
Chassi 0 - 4	1,589	1,677	1,926	1,196	1,799	1,534	1,617	1,252
Chassi 0 - 5	1,520	1,338	1,124	1,165	1,638	1,344	1,637	1,228
Chassi 5 - 6	1,740	2,035	1,060	1,383	1,800	1,326	1,152	1,178
Chassi 5 - 7	0,000	1,850	1,863	1,559	1,659	1,442	1,217	1,186
Chassi 5 - 8	1,447	1,206	1,264	1,759	1,710	1,491	1,483	1,298
Chassi 5 - 9	1,713	1,222	1,934	1,430	1,314	1,464	1,241	1,237
Chassi 5 - 10	1,641	1,796	1,091	1,884	1,732	1,454	1,468	1,111
Chassi 10 - 1	1,426	2,443	1,140	1,290	1,990	1,298	1,422	0,972
Chassi 10 - 2	1,963	1,449	1,088	1,712	1,374	1,288	1,490	1,194
Chassi 10 - 3	1,663	2,123	1,981	1,306	1,495	1,296	1,125	1,618
Chassi 10 - 4	1,452	1,296	1,879	0,000	1,673	1,479	1,556	1,177
Chassi 10 - 5	1,660	1,137	0,498	1,324	1,344	1,444	1,295	1,200
Chassi 15 - 6	2,289	1,613	2,022	0,978	1,758	1,499	1,615	1,166
Chassi 15 - 7	2,059	1,559	1,149	0,935	1,759	1,310	1,319	1,184
Chassi 15 - 8	2,220	1,795	1,715	1,675	1,301	0,000	1,186	1,155
Chassi 15 - 9	1,705	1,874	1,139	1,748	1,728	1,499	1,126	1,303
Chassi 15 - 10	2,120	1,815	1,708	1,745	1,715	1,423	1,239	1,147
Chassi 20 - 1	1,460	1,764	1,726	1,527	1,742	1,240	1,158	1,639
Chassi 20 - 2	1,460	1,841	1,437	2,116	1,698	1,384	1,741	1,268
Chassi 20 - 3	1,447	1,778	1,095	1,410	1,515	1,286	1,298	1,237
Chassi 20 - 4	1,830	1,227	1,155	1,273	1,688	1,574	1,308	1,494
Chassi 20 - 5	1,496	1,040	1,600	2,089	1,588	1,536	1,250	1,255
Chassi 25 - 6	1,457	1,474	1,122	2,047	1,456	1,595	1,341	1,379
Chassi 25 - 7	1,674	1,561	1,056	1,228	1,420	1,494	0,000	1,183
Chassi 25 - 8	1,422	1,230	1,358	2,500	1,467	1,457	1,349	1,453
Chassi 25 - 9	1,482	1,544	1,491	2,031	1,723	1,310	1,075	1,245
Chassi 25 - 10	1,464	1,253	1,125	1,454	1,642	1,531	1,275	1,255

com algoritmos de otimização mais eficientes, a redução nos servidores é de 51,04%, no sistema de refrigeração é de 61,70% e na infraestrutura de rede 38,86%, chegando a um consumo final 49,88% menor em relação a um centro de dados tradicional.

No centro de dados verde, alguns chassis sofreram por alguns segundos uma sobrecarga térmica. Durante os 25 dias simulados do traço da SHARCNET, nove chassis permaneceram em média 25,51 segundos não consecutivos com temperaturas acima de 30°C, sendo esse

Tabela 6.19 – Desempenho de um centro de dados tradicional x centro de dados verde.

	Centro de Dados Tradicional	Centro de Dados Verde	Porcentagem Redução
Consumo Energético (GW)			
Servidor	149,68 $\pm 6,47$	80,63 $\pm 0,09$	51,04%
Processador	146,43 $\pm 6,54$	78,14 $\pm 0,10$	46,64%
Ventoinhas	3,25 $\pm 0,16$	2,49 $\pm 0,02$	23,38%
Sistema de Refrigeração	51,12 $\pm 2,21$	19,58 $\pm 0,08$	61,70%
Comutadores	3,86 $\pm 0,22$	2,36 $\pm 0,08$	38,86%
Acesso	2,48 $\pm 0,14$	1,77 $\pm 0,00$	28,63%
Agregação	0,63 $\pm 0,04$	0,27 $\pm 0,08$	57,14%
Núcleo	0,75 $\pm 0,04$	0,32 $\pm 0,00$	57,33%
TOTAL	204,66 $\pm 8,90$	102,57 $\pm 0,23$	49,88%
Temperatura			
Temperatura média do fornecimento ar ($^{\circ}C$)	19	25,51 $\pm 0,06$	
Total de Chassis com Sobrecarga Térmica	0	9 ± 1	
Tempo médio com Sobrecarga Térmica por Chassi (s)	0	29,53 $\pm 0,02$	

tempo desprezível em comparação com os benefícios da economia de energia no sistema de refrigeração.

6.5 Considerações Finais

Este capítulo apresentou os resultados de um conjunto de simulações que demonstram as características do SimDC3D. Inicialmente, descreveu-se um conjunto restrito de parâmetros, dentre os 137 disponíveis, que foram parametrizados nas simulações. Optou-se por combinar um conjunto reduzido de especificações, devido à maior influência que exercem no consumo energético e no desempenho do centro de dados. Além de descrever os parâmetros mais relevantes, foram descritas as variáveis analisadas para determinar o desempenho e os níveis de qualidade de serviço do centro de dados.

A avaliação experimental consistiu em sete estudos de caso, em que cada um apresentou algumas características do simulador e avaliou os impactos que as estratégias de otimização causaram em todas as dimensões do centro de dados. Também demonstrou-se como os parâmetros de simulação influenciam tanto no desempenho energético como nos indicadores de

qualidade do centro de dados. Foram evidenciados um conjunto de fatores que podem melhorar alguns indicadores de qualidade, porém deve-se considerar o custo energético para isso.

O SimDC3D também foi testado usando diversos tipos de carga de processamento, exigindo em certos momentos a capacidade de processamento máxima do centro de dados. Contudo, em alguns momentos, as cargas excessivas de processamento exigidas por alguns traços afetam o desempenho do centro de dados, sendo um resultado esperado.

Por fim, foram descritas as estatísticas dos preditores utilizados no SimDC3D e como eles podem auxiliar na melhoria dos indicadores de desempenho.

CAPÍTULO 7

CONCLUSÃO E TRABALHOS FUTUROS

A computação em nuvem é uma das principais tecnologias disruptivas do nosso tempo. A nuvem tem o potencial de permitir uma capacidade infinita de computação e armazenamento, maior conectividade e, principalmente, uma computação onipresente. Portanto, é fundamental que a computação em nuvem possa se expandir de forma sustentável, minimizando os impactos ambientais e os custos operacionais causados pela grande quantidade de energia utilizada pelos centros de dados que a suportam.

Para dar suporte à computação na nuvem, novos centros de dados precisam ser construídos e antigos expandidos. Mas, mais importante do que construções e expansões, eles precisam ser gerenciados adequadamente para evitar desperdícios na utilização de recursos e principalmente para reduzir os altos custos operacionais e ambientais.

O escalonamento e a realocação de máquinas virtuais possibilitam que um centro de dados seja gerenciado para melhorar seu desempenho energético mediante a reorganização da infraestrutura na qual são identificados e desativados recursos ociosos. Este trabalho mostrou com resultados de vários cenários estudados que estratégias de escalonamento e realocação de máquinas virtuais, controle dinâmico de temperatura e otimização da topologia de rede podem reduzir o consumo energético em mais de 50% no centro de dados.

Devido à dificuldade de se testar e avaliar estratégias de economia de energia em um ambiente real, este trabalho desenvolveu o simulador SimDC3D que considera todas as dimensões de maior consumo (servidores, sistema de refrigeração e infraestrutura de rede), principalmente por não haver atualmente nenhum simulador disponível que incorpore adequadamente todas essas dimensões. O SimDC3D é um simulador que reproduz realisticamente essas três dimensões, podendo, assim, prover um ambiente de simulação para analisar o impacto das estratégias adotadas em todas as dimensões do centro de dados. O SimDC3D, portanto, difere significativamente de outros simuladores que disponibilizam análises apenas em uma ou, no máximo, em duas dimensões, deixando de relacionar as decisões tomadas em cada dimensão em relação às outras.

A arquitetura do SimDC3D foi idealizada para permitir inúmeras combinações de parâmetros, diferentes cargas de processamento e matrizes de tráfego e várias topologias de redes. Essas características permitem avaliar o impacto de estratégias de economia de energia em um centro de dados diante de diferentes cargas de trabalho e com a possibilidade de mesclar diversos algoritmos e métodos de economia de energia. O SimDC3D possibilita, também, a implementação de novas estratégias, estendendo suas principais classes. Além disso, ele simula recursos de *hardware* como: processadores, ventoinhas, memória, interface de redes, sistema de refrigeração e a infraestrutura de rede.

O simulador SimDC3D foi utilizado em simulações de diferentes cenários para demonstrar os impactos de algoritmos desenvolvidos neste trabalho na economia de energia e em parâmetros de qualidade de serviço. Inicialmente, demonstrou-se que a decisão de colocar servidores ociosos em hibernação ou desligá-los afeta os acordos de nível de serviço. A escolha de aumentar a economia de energia desligando os servidores em vez de colocá-los em modo de hibernação aumenta a quebra do SLA em mais de 57%. Ainda, o resultados experimentais mostram um conflito entre qualidade de serviço e economia de energia. A utilização de um banco de servidores, em um período de 30 dias, elevou o consumo energético em mais de 14% em comparação com os mesmos parâmetros sem a utilização do banco. Porém, com o banco de servidores, houve uma redução de mais de 97% na quebra dos acordos de nível de serviço.

Diferentes estratégias de alocação de VMs geram consumos energéticos distintos nos servidores, sistema de refrigeração e na infraestrutura da rede como mostra a Tabela 6.9. O 3DMOBFD(b) reduziu em 2,5% o consumo energético em comparação ao do PABFD com um traço de carga de trabalho de 30 dias. Em um período de um ano, essa economia salta para 30% devido a escolha de um estratégia mais agressiva ou 10% caso utilize uma estratégia menos agressiva (3DMOBFD(b)). Todas as estratégias mantiveram o mesmo percentual de quebra de SLA (0,07). Porém, observa-se que, devido à redução da quantidade de servidores utilizados pelos algoritmos 3DMOBFD(a) e 3DMOBFD(b), houve um aumento no número de migrações por sobrecarga e pelo preditor. Contudo, esse aumento não afetou o SLA das duas estratégias em comparação ao PABFD.

Centros de dados com controle estático de temperatura são ineficientes e geram um desperdício de energia. Ajustar a temperatura do centro de dados dinamicamente de acordo com a demanda e que combata rapidamente os pontos quentes do centro de dados reduz o consumo energético do sistema de refrigeração em mais de 27%, em média. Esse resultado é importante porque ele independe dos algoritmos de escalonamento e realocação de máquinas virtuais.

Diferentes topologias de redes proporcionam ao centro de dados diversos níveis de tolerância a falhas na infraestrutura de comunicação. Entretanto, desligar os enlaces e comutadores ociosos pode afetar diretamente a capacidade do centro de dados de reagir a falhas. Para evitar isso, a adaptação dinâmica é uma alternativa. Os estudos de caso demonstraram uma redução de mais de 10% no consumo energético da infraestrutura de rede sem prejudicar os

níveis de redundância implementados. Todavia, maiores economias podem ser obtidas caso sejam reduzidos a quantidade de enlaces e comutadores desnecessários em períodos de baixa utilização. Além disso, foram observadas economias em torno de 18% ao se reduzir enlaces e comutadores e 30% ao se aplicar a otimização juntamente com ALR para uma topologia de três camadas com enlaces redundantes. Essa economia pode chegar a 40% em uma topologia de duas camadas. Esses percentuais estão consistentes com outros resultados reportados em outros trabalhos da área.

Os preditores utilizados neste trabalho demonstraram ser uma importante ferramenta para alocação mais eficiente de cargas de trabalho. O preditor de uso de CPU utilizando regressão linear reduziu em mais de 44% a quebra nos acordos de nível de serviço com um pequeno aumento no consumo energético de apenas 1,44%. Também, foram avaliados dois algoritmos de predição (RBF e POLYNOM) para prever o comportamento da variável temperatura. Os resultados mostraram que o algoritmo RBF teve uma margem de erro menor em relação ao *POLYNOM* para predições de até 5 min.

Para previsões com tempos futuros maiores, o algoritmo *POLYNOM* demonstrou ter uma margem de erro menor do que o *RBF*. O erro médio para uma janela de 20 minutos variou entre 2,08% até 3,13% com uma média de 2,61%, enquanto o RBF teve uma média de 3,01%.

Muito embora este trabalho tenha apresentado uma avaliação experimental extensiva de vários parâmetros de um centro de dados, ainda há muitas questões importantes a serem investigadas. A quantidade de parâmetros que pode interferir no consumo energético e na qualidade de serviço de um centro de dados é muito alta e testar todas as combinações levaria a um número exorbitante de possíveis cenários. Por isso, uma maneira de se mitigar esse problema, mas que possui alta complexidade, seria o desenvolvimento de um modelo matemático que incorpore as três dimensões de maior consumo para se avaliar analiticamente a qualidade de algoritmos de escalonamento e realocação de máquinas virtuais.

O SimDC3D também apresenta algumas possibilidades de trabalhos futuros. Uma extensão particularmente importante seria a simulação de múltiplos centros de dados, permitindo que cargas de trabalho fossem realocadas não somente internamente, mas também entre centros de dados. Com isso, ele possibilitaria avaliar, por exemplo, algoritmos que consideram horários distintos para alocar cargas de trabalho. Realocar cargas entre centros de dados em fusos horários distantes pode possibilitar a transferência de uma carga de trabalho de um país que está em seu horário comercial para outro país que está em um horário noturno e com baixa utilização. Evidentemente, nesse tipo de simulação os aspectos de qualidade de serviço se tornam ainda mais importantes por causa de possíveis latências que seriam inseridas.

REFERÊNCIAS

- ABTS, D. et al. Energy proportional datacenter networks. *SIGARCH Comput. Archit. News*, ACM, New York, NY, USA, v. 38, n. 3, p. 338–347, jun. 2010.
- ABU-LIBDEH, H. et al. Symbiotic routing in future data centers. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 40, n. 4, p. 51–62, ago. 2010.
- AKHTER, N.; OTHMAN, M. Energy Aware Resource Allocation of Cloud Data Center: Review and Open Issues. *Cluster Computing*, v. 19, n. 3, p. 1163–1182, Sep 2016.
- AL-FARES, M.; LOUKISSAS, A.; VAHDAT, A. A scalable, commodity data center network architecture. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 38, p. 63–74, August 2008.
- ALGER, D. *Grow a Greener Data Center*. 1st. ed. [S.l.: s.n.], 2009.
- ASHRAE. *American Society of Heating, Refrigerating and Air-Conditioning Engineers*. 2014. <https://www.ashrae.org>. Last checked March 25, 2014.
- AVETISYAN, A. I. et al. Open cirrus: A global cloud computing testbed. *Computer*, Institute of Electrical and Electronics Engineers, Inc., 3 Park Avenue, 17 th Fl New York NY 10016-5997 United States, v. 43, n. 4, p. 35–43, 2010.
- BARROSO, L.; HOLZLE, U. The case for energy-proportional computing. *Computer*, v. 40, n. 12, p. 33–37, 2007.
- BELOGLAZOV, A. *Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing*. Tese (Doutorado) — The university of Melbourne, 2013.
- BELOGLAZOV, A.; BUYYA, R. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. In: *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*. New York, NY, USA: ACM, 2010. (MGC '10), p. 4:1–4:6.
- BELOGLAZOV, A.; BUYYA, R. Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers. *Concurrency and Computation: Practice and Experience*, Wiley Online Library, v. 24, n. 13, p. 1397–1420, 2012.

- BELOGLAZOV, A.; BUYYA, R. Energy efficient allocation of virtual machines in cloud data centers. In: *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*. [S.l.: s.n.], May. p. 577–578.
- BELOGLAZOV, A. et al. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *CoRR*, abs/1007.0066, 2010.
- BENSON, T. et al. Understanding Data Center Traffic Characteristics. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 40, p. 92–99, jan. 2010.
- BIANZINO, A. P. et al. A survey of green networking research. *Communications Surveys Tutorials, IEEE*, PP, n. 99, p. 1–18, 2010.
- BOLLA, R. et al. Energy-aware performance optimization for next-generation green network equipment. In: *Proceedings of the 2nd ACM SIGCOMM workshop on Programmable routers for extensible services of tomorrow*. New York, NY, USA: ACM, 2009. (PRESTO '09), p. 49–54.
- BOX, G. E.; JENKINS, G. M.; REINSEL, G. C. *Time series analysis: forecasting and control*. 3rd. ed. [S.l.]: Prentice Hall, 1994.
- BUYYA, R.; RANJAN, R.; CALHEIROS, R. N. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. In: *IEEE. High Performance Computing & Simulation, 2009. HPCS'09. International Conference on*. [S.l.], 2009. p. 1–11.
- CALHEIROS, R. N. et al. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. *Software: Practice and Experience*, Wiley Online Library, v. 41, n. 1, p. 23–50, 2011.
- CLOS, C. A study of non-blocking switching networks. *Bell System Technical Journal*, Blackwell Publishing Ltd, v. 32, n. 2, p. 406–424, 1953.
- COLE, G. Estimating drive reliability in desktop computers and consumer electronics systems. *Seagate Technology Paper TP*, v. 338, 2000.
- COOLSIM. *CoolSim*. 2019. <https://www.coolsimsoftware.com/>. Last checked September 5, 2019.
- COUTO, R. S.; CAMPISTA, M. E. M.; COSTA, L. H. M. Uma avaliação da robustez intra data centers baseada na topologia da rede. In: *SBRC 2012*. Ouro Preto, MG: SBRC 2012, 2012.
- FITTKAU, F.; FREY, S.; HASSELBRING, W. Cloud user-centric enhancements of the simulator cloudsim to improve cloud deployment option analysis. In: _____. *Service-Oriented and Cloud Computing: First European Conference, ESOC 2012, Bertinoro, Italy, September 19-21, 2012. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 200–207.
- GARG, S.; BUYYA, R. Networkcloudsim: Modelling parallel applications in cloud simulations. In: *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. [S.l.: s.n.], 2011. p. 105–113.
- GARTNER. *The Data Center Is Dead, and Digital Infrastructures Emerge*. 2018. <https://www.equinox.com.br/resources/analyst-reports/gartner-emerging-digital-infrastructures/>.

- GE, C.; SUN, Z.; WANG, N. A Survey of Power-Saving Techniques on Data Centers and Content Delivery Networks. *IEEE Communications Surveys Tutorials*, v. 15, n. 3, p. 1334–1354, Third 2013.
- GILBERT, R. R. *Validation of Computational Fluid Dynamics Based Data Center Cyber-Physical Models*. Dissertação (Mestrado) — Arizona State University, 2012.
- GREENBERG, A. et al. V12: a scalable and flexible data center network. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 39, p. 51–62, ago. 2009.
- GROSSMAN, R. L. et al. The open cloud testbed: A wide area testbed for cloud computing utilizing high performance network services. *CoRR*, abs/0907.4810, 2009. Disponível em: <http://arxiv.org/abs/0907.4810>.
- GUNARATNE, C. et al. Reducing the energy consumption of ethernet with adaptive link rate (alr). *IEEE Transactions on Computers*, IEEE Computer Society, Los Alamitos, CA, USA, v. 57, p. 448–461, 2008.
- GUO, C. et al. Bcube: a high performance, server-centric network architecture for modular data centers. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 39, p. 63–74, ago. 2009.
- GUO, C. et al. Dcell: a scalable and fault-tolerant network structure for data centers. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 38, p. 75–86, August 2008.
- GUPTA, S. et al. Gdcsim: A tool for analyzing green data center design and resource management techniques. In: *Green Computing Conference and Workshops (IGCC), 2011 International*. [S.l.: s.n.], 2011. p. 1–8.
- GUSTEDT, J.; JEANNOT, E.; QUINSON, M. Experimental methodologies for large-scale systems: A survey. *Parallel Processing Letters*, v. 19, n. 03, p. 399–418, 2009. Disponível em: <http://www.worldscientific.com/doi/abs/10.1142/S0129626409000304>.
- HEBREW. *Parallel Workloads Archive*. 2018. <https://www.cs.huji.ac.il/labs/parallel/workload/logs.html>.
- HELLER, B. et al. ElasticTree: Saving Energy in Data Center Networks. In: *Proceedings of the 7th USENIX NSDI*. Berkeley, CA, USA: USENIX Association, 2010. (NSDI'10), p. 17–17.
- HU, L. et al. Hmdc live virtual machine migration based on hybrid memory copy and delta compression. In: . [S.l.: s.n.], 2013.
- ISHII, R. P. *Otimização de operações de entrada e saída visando reduzir o tempo de resposta de aplicações distribuídas que manipulam grandes volumes de dados*. Tese (Doutorado) — Universidade de São Paulo, 2010.
- ISSARIYAKUL, T.; HOSSAIN, E. *Introduction to Network Simulator NS2*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 1441944125, 9781441944122.
- JARARWEH, Y. et al. Teachcloud: a cloud computing educational toolkit. *International Journal of Cloud Computing*, v. 2, n. 2-3, p. 237–257, 2013.
- JUNG, G. et al. Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures. In: IEEE. *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*. [S.l.], 2010. p. 62–73.

- KANT, K.; MURUGAN, M.; DU, D.-C. Willow: A control system for energy and thermal adaptive computing. In: *Parallel Distributed Processing Symposium (IPDPS), 2011 IEEE International*. [S.l.: s.n.], 2011. p. 36–47.
- KLIAZOVICH, D.; BOUVRY, P.; KHAN, S. GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers. *The Journal of Supercomputing*, Springer US, v. 62, n. 3, p. 1263–1283, 2012.
- KLIAZOVICH, D.; BOUVRY, P.; KHAN, S. U. Dens: data center energy-efficient network-aware scheduling. In: IEEE. *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*. [S.l.], 2010. p. 69–75.
- LAGO, D. G.; MADEIRA, E. R.; BITTENCOURT, L. F. Escalonamento com prioridade na alocação ciente de energia de máquinas virtuais em nuvens. *Anais do XXX SBRC*, 2012.
- LEFURGY, C.; WANG, X.; WARE, M. Server-level power control. In: *Autonomic Computing, 2007. ICAC '07. Fourth International Conference on*. [S.l.: s.n.], 2007. p. 4–4.
- LEISERSON, C. E. Fat-trees: universal networks for hardware-efficient supercomputing. *Computers, IEEE Transactions on*, IEEE, v. 100, n. 10, p. 892–901, 1985.
- LLC, M. *The Green Grid*. 2014. Disponível em: <<http://www.thegreengrid.org/>>.
- LUČANIN, D. et al. Energy-aware cloud management through progressive sla specification. In: *Economics of Grids, Clouds, Systems, and Services*. [S.l.]: Springer, 2014. p. 83–98.
- MAHADEVAN, P.; BANERJEE, S.; SHARMA, P. Energy proportionality of an enterprise network. In: *Proceedings of the first ACM SIGCOMM workshop on Green networking*. New York, NY, USA: ACM, 2010. (Green Networking '10), p. 53–60.
- MAHADEVAN, P.; SHAH, A.; BASH, C. Reducing lifecycle energy use of network switches. In: *Sustainable Systems and Technology (ISSST), 2010 IEEE International Symposium on*. [S.l.: s.n.], 2010. p. 1–6.
- MAHADEVAN, P. et al. Energy aware network operations. In: *INFOCOM Workshops 2009, IEEE*. [S.l.: s.n.], 2009. p. 1–6.
- MAHADEVAN, P. et al. A power benchmarking framework for network devices. In: FRATTA, L. et al. (Ed.). *NETWORKING 2009*. [S.l.]: Springer Berlin / Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5550). p. 795–808.
- MARTELLO, S.; TOTH, P. *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990.
- MENG, X.; PAPPAS, V.; ZHANG, L. Improving the scalability of data center networks with traffic-aware virtual machine placement. In: IEEE. *INFOCOM, 2010 Proceedings IEEE*. [S.l.], 2010. p. 1–9.
- MOORE, J. et al. Making Scheduling Cool: Temperature-Aware Workload Placement in Data Centers. In: *Proceedings of The USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2005. (ATC '05), p. 5–5.

- MORETTIN, P. A.; TOLOI, C. *Análise de séries temporais*. 2nd. ed. [S.l.]: Editora Edgard Blucher, 2006.
- MORO, M. P. *SimDC3D: A Data Center Simulator*. 2018. <https://github.com/simdc3d>.
- MYSORE, R. N. et al. Portland: A scalable fault-tolerant layer 2 data center network fabric. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 39, n. 4, p. 39–50, ago. 2009.
- NATHAN, S.; KULKARNI, P.; BELLUR, U. Resource availability based performance benchmarking of virtual machine migrations. In: *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*. New York, NY, USA: ACM, 2013. (ICPE '13), p. 387–398.
- NEDEVSKI, S. et al. Reducing network energy consumption via sleeping and rate-adaptation. In: *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2008. (NSDI'08), p. 323–336.
- NUCCI, A.; SRIDHARAN, A.; TAFT, N. The Problem of Synthetically Generating IP Traffic Matrices: Initial Recommendations. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 35, n. 3, p. 19–32, jul. 2005. ISSN 0146-4833.
- NÚÑEZ, A. et al. New techniques for simulating high performance mpi applications on large storage networks. *The Journal of Supercomputing*, v. 51, n. 1, p. 40–57, 2010.
- OKADA, T. K. et al. Consolidation of vms to improve energy efficiency in cloud computing environments. In: *2015 XXXIII Brazilian Symposium on Computer Networks and Distributed Systems*. [S.l.: s.n.], 2015. p. 150–158.
- OSTERMANN, S. et al. Groudsim: An event-based simulation framework for computational grids and clouds. In: _____. *Euro-Par 2010 Parallel Processing Workshops: HeteroPar, HPCC, HiBB, CoreGrid, UCHPC, HPCF, PROPER, CCPI, VHPC, Ischia, Italy, August 31–September 3, 2010, Revised Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 305–313.
- OXLEY, M. A. et al. Rate-based thermal, power, and co-location aware resource management for heterogeneous data centers. *Journal of Parallel and Distributed Computing*, v. 112, p. 126 – 139, 2018. Parallel Optimization using/for Multi and Many-core High Performance Computing.
- PATEL, C. D. et al. Smart cooling of data centers. In: AMERICAN SOCIETY OF MECHANICAL ENGINEERS. *ASME 2003 International Electronic Packaging Technical Conference and Exhibition*. [S.l.], 2003. p. 129–137.
- PATEL, S.; MAKWANA, R. M. Optimized energy efficient virtual machine placement algorithm and techniques for cloud data centers. *Journal of Computer Science*, v. 12, n. 9, p. 448–454, 2016.
- REPORTER, D. M. *Google uses MORE power than Salt Lake City as vast data farms suck up electricity*. 2014. <http://www.dailymail.co.uk/news/article-2035382/Google-discloses-energy-consumption-power-Salt-Lake-City.html>. Last checked March 3, 2014.

- RICCIARDI, S. et al. Saving Energy in Data Center Infrastructures. In: *Data Compression, Communications and Processing (CCP), 2011 First International Conference on*. [S.l.: s.n.], 2011. p. 265–270.
- RONG, H. et al. Optimizing Energy Consumption for Data Centers. *Renewable and Sustainable Energy Reviews*, v. 58, n. C, p. 674–691, 2016.
- SAINO, L.; COCORA, C.; PAVLOU, G. A Toolchain for Simplifying Network Simulation Setup. In: *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*. ICST, Brussels, Belgium, Belgium: ICST, 2013. (SIMUTOOLS '13).
- SAMS, S. L. Discovering Hidden Costs in your Data Centre—a CFO Perspective. *IBM White Paper*, IBM, set. 2010.
- SHANG, Y.; LI, D.; XU, M. A comparison study of energy proportionality of data center network architectures. p. 1–7, June 2012.
- SHIRAYANAGI, H.; YAMADA, H.; KONO, K. Honeyguide: A vm migration-aware network topology for saving energy consumption in data center networks. In: *Computers and Communications (ISCC), 2012 IEEE Symposium on*. [S.l.: s.n.], 2012. p. 000460–000467.
- SINGH, S. et al. Eh-gc: An efficient and secure architecture of energy harvesting green cloud infrastructure. *Sustainability*, v. 9, n. 4, 2017. ISSN 2071-1050. Disponível em: <https://www.mdpi.com/2071-1050/9/4/673>.
- SINHA, P.; ZOLTNERS, A. A. The multiple-choice knapsack problem. *Operations Research*, v. 27, n. 3, p. 503–515, 1979.
- SRIRAM, I. Speci, a simulation tool exploring cloud-scale data centres. In: _____. *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 381–392. Disponível em: http://dx.doi.org/10.1007/978-3-642-10665-1_35.
- SULLIVAN, R. F. Alternating cold and hot aisles provides more reliable cooling for server farms. *Uptime Institute*, 2000.
- TANG, Q.; GUPTA, S. K. S.; VARSAMOPOULOS, G. Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach. *Parallel and Distributed Systems, IEEE Transactions on*, v. 19, n. 11, p. 1458–1472, Nov 2008.
- TIGHE, M. et al. DCSim: A Data Centre Simulation Tool for Evaluating Dynamic Virtualized Resource Management. In: *CNSM 2012 Workshop on Systems Virtualization Management*. [S.l.: s.n.], 2012. p. 385–392.
- UDDIN, M. et al. Measuring efficiency of tier level data centers to implement green energy efficient data centers. *Middle - East Journal of Scientific Research*, International Digital Organization for Scientific Information, v. 15, n. 2, p. 200–207, 2013.
- VALIANT, L. G. A scheme for fast parallel communication. *SIAM Journal on Computing*, v. 11, n. 2, p. 350–361, 2008.

- VOORSLUYS, W. et al. Cost of virtual machine live migration in clouds: A performance evaluation. In: *Cloud Computing*. [S.l.]: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5931). p. 254–265.
- WANG, X.; CHEN, M. Cluster-level feedback power control for performance optimization. In: IEEE. *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*. [S.l.], 2008. p. 101–110.
- WANG, X. et al. CARPO: Correlation-Aware Power Optimization in Data Center Networks. In: *2012 Proceedings IEEE INFOCOM*. [S.l.: s.n.], 2012. p. 1125–1133.
- WU, Q. et al. Formal control techniques for power-performance management. *Micro, IEEE*, v. 25, n. 5, p. 52–62, 2005.
- XU, D.; WANG, K. Stochastic modeling and analysis with energy optimization for wireless sensor networks. *International Journal of Distributed Sensor Networks*, v. 10, n. 5, 2014.
- Xu, F. et al. Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions. *Proceedings of the IEEE*, v. 102, n. 1, p. 11–31, Jan 2014.
- YEO, S.; LEE, H.-H. S. SimWare: A Holistic Warehouse-Scale Computer Simulator. *Computer*, Institute of Electrical and Electronics Engineers, Inc., 3 Park Avenue, 17 th Fl New York NY 10016-5997 United States, v. 45, n. 9, p. 48–55, 2012.
- YUE, M.; ZHANG, L. A Simple Proof of the Inequality $MFFD(L) \leq 71/60 OPT(L) + 1$, L for the MFFD Bin-Packing Algorithm. *Acta Mathematicae Applicatae Sinica*, v. 11, n. 3, p. 318–330, Jul 1995.

APÊNDICE A

MODELAGEM DO PROBLEMA DE MAPEAMENTO DE MÁQUINA VIRTUAL

Um cenário desejado é que o centro de dados seja gerenciado por um controlador de energia com capacidade de gerenciar os servidores, máquinas virtuais, sistema de refrigeração e a rede. Tal controlador coletará informações de temperatura, tráfego entre os *switches*, utilização de CPU, memórias e rede em cada servidor do centro de dados.

Com base nessas informações coletadas em intervalos de tempo regulares, o controlador determinará quais são os servidores ideais que receberão novas máquinas virtuais. É de sua responsabilidade detectar variações nas temperaturas dos servidores ou da rede, estabelecendo uma nova realocação para um conjunto de máquinas virtuais. Outra função do controlador é definir os melhores caminhos para os vários fluxos dos dados na rede do centro de dados, tendo como objetivo minimizar a utilização dos ativos de rede. É desejável que o controlador gerencie portas não utilizadas, e adapte a velocidade de cada porta conforme a demanda de tráfego.

Uma solução para resolver o problema do mapeamento de um conjunto de v ($v \in \{1, \dots, v\}$) máquinas virtuais para serem alocadas em um conjunto de m ($m \in \{1, \dots, m\}$) servidores é um algoritmo guloso. Para cada máquina virtual, busca-se em todos os servidores para determinar a melhor alocação, esse algoritmo tem uma complexidade na ordem de $O(v \times m)$. Para centro de dados com uma grande quantidade de servidores e máquinas virtuais, o tempo necessário para achar uma solução eficiente é inviável, sendo um problema NP-Completo.

Encontrar um mapeamento ótimo que obtenha o menor consumo energético nas três dimensões é um problema de otimização combinatória multiobjetivo. Dado um conjunto de v máquinas virtuais e m servidores, há m^v mapeamentos teoricamente possíveis, resultando em um grande espaço de busca. As soluções devem ser avaliadas pelos seguintes critérios: custos em termos de consumo energético, número de migrações, número de *boots/shutdown* de servidores, etc.

O problema de otimização combinatória multiobjetivo produz um mapeamento de máquinas virtuais em máquinas físicas que obtenha o menor consumo energético nos servidores, sistema de refrigeração e rede. Ele pode ser estendido como um problema de múltiplas mochilas que foi proposto para problemas de tolerância a falhas (SINHA; ZOLTNER, 1979). Na próxima seção definiremos o problema da mochila para um único objetivo, e na seção seguinte o problema é estendido para múltiplas mochilas (múltiplos objetivos).

A.1 Problema da Mochila

O problema da mochila constitui um grupo de problemas de otimização combinatória que vêm sendo intensamente estudados há várias décadas, atraindo teóricos e práticos. O interesse dos teóricos deve-se, principalmente, à estrutura simples do problema, o que permite, além da exploração de diversas propriedades do mesmo, a sua resolução como parte de problemas mais complexos. Já do ponto de vista prático, estes problemas aparecem em diversas aplicações reais, incluindo, entre outros, carregamento de veículos, problemas de corte e empacotamento.

Neste problema, existe uma determinada quantidade de itens, cada qual como o seu peso e valor, onde deseja-se colocar esses itens em uma mochila com uma capacidade predefinida. O objetivo é colocar os itens que somem o maior valor, sendo que a soma dos pesos não ultrapasse a capacidade da mochila.

O problema da mochila pode ser descrito da seguinte forma: dado um conjunto de n itens, no qual cada item $j = \{1, \dots, n\}$ possui um peso w_j e um lucro v_j associados. A mochila possui uma capacidade finita C , determinando o peso máximo suportado por ela. Martello et al. (MARTELLO; TOTH, 1990) demonstraram matematicamente uma solução para esse problema como mostra as equações a seguir:

$$\text{Maximizar } z = \sum_{j=1}^n v_j x_j \quad (\text{A.1})$$

sujeito a

$$\sum_{j=1}^n w_j x_j \leq C, \quad (\text{A.2})$$

$$x_j = 0 \text{ ou } 1, \quad j \in N = \{1, \dots, n\} \quad (\text{A.3})$$

A Equação (A.1) define o objetivo do problema, que é maximizar o lucro obtido pelos itens inseridos na mochila. A Restrição (A.2) garante que os itens inseridos não ultrapasse a capacidade da mochila e a Restrição (A.3) especifica os valores que a variável x_j pode receber, informando se o item foi ou não inserido, respectivamente.

Uma extensão do problema da mochila é o problema de múltiplas mochilas. A diferença entre estes problemas estão no número de mochilas utilizadas: enquanto no primeiro problema utilizamos apenas uma mochila, no problema de múltiplas mochilas são utilizadas várias

mochilas. Esse problema foi proposto para ser aplicado em problemas de tolerância a falhas, esquema de criptografia pública e problemas de escalonamento. A formulação matemática proposto por Martello et al. (MARTELLO; TOTH, 1990) é descrito a seguir:

$$\text{Maximizar } z = \sum_{i=1}^m \sum_{j=1}^n v_j x_{ij}, \quad (\text{A.4})$$

sujeito a

$$\sum_{j=1}^n w_j x_{ij} \leq c_i, \quad i \in M = \{1, \dots, m\}, \quad (\text{A.5})$$

$$\sum_{i=1}^m x_{ij} \leq 1, \quad j \in N = \{1, \dots, n\}, \quad (\text{A.6})$$

$$w_j \leq \max_{i \in M} \{c_i\} \quad j \in N, \quad (\text{A.7})$$

$$c_i \geq \min_{j \in N} \{w_j\} \quad i \in M, \quad (\text{A.8})$$

$$v_j \text{ e } c_i \geq \text{inteiros positivos}, \quad (\text{A.9})$$

$$x_{ij} = 0 \text{ ou } 1, \quad i \in M, j \in N, \quad (\text{A.10})$$

Em que:

n é a quantidade total de itens;

m é a quantidade total de mochilas;

w_j e v_j representam, respectivamente, o peso e o valor de item j ;

c_i representa a capacidade da mochila i ;

x_{ij} indica se o item j esta na mochila i ($x_{ij} = 1$) ou não ($x_{ij} = 0$).

Como mostra a Equação (A.4) o objetivo do problema é maximizar o valor total obtido por inserir os elementos nas mochilas. A Restrição (A.5) garante que os itens inseridos não ultrapasse o peso da mochila, a Restrição (A.6) não permite que um item seja inserido em mais de uma mochila, as Restrições (A.7) e (A.8) garante que os itens não terão o peso maior que a maior capacidade de uma mochila e a capacidade das mochilas deve ser maior ou igual ao menor peso entre os itens. As Restrições (A.9) e (A.10) limita o problema a só trabalhar com valores inteiros e binários.

Para centro de dados heterogêneos, as mochilas tem capacidades diferentes, e para centro de dados homogêneos todas as mochilas têm a mesma capacidade.

O problema de múltiplas mochilas pode ser estendido para resolver o problema de mapeamento de VMs para servidores, neste caso uma mochila é um servidor, e os itens de diferentes capacidades são as VMs.

O problema de alocação de VMs para máquinas físicas pode ser descrito da seguinte forma:

- **Máquinas Virtuais (VM):** são os itens a serem inseridos nas mochilas. Temos V máquinas virtuais, e os recursos necessários para a VM i com $1 \leq i \leq V$ são

representados por um vetor de D -dimensões \vec{v}_i , em que $v_{i,d}$ representa a demanda de recursos da VM i no componente d , em que $1 \leq d \leq D$. As VMs não têm lucro ou valor, apenas pesos em dimensões D . No âmbito deste trabalho, o D assume valor 4 que representa: o processador, demanda de memória, largura de banda e temperatura, que são os recursos mais importantes para se obter economia de energia;

- **Servidores Físicos (SF):** os servidores físicos são as mochilas que abrigam as VMs (itens). Temos S servidores físicos, a capacidade de recurso do SF j é representado por um vetor D -dimensional \vec{r}_j , com $1 \leq j \leq R$, onde $r_{j,d}$ representa a capacidade de recursos do Servidores Físicos (SF) j no componente d , com $1 \leq d \leq D$.
- **Mapeamento de VMs para SFs:** uma matriz de mapeamento binário X é usada, com R linhas e V colunas, em que $X_{ji} \in \{0, 1\}$ e $X_{ji} = 1 \Leftrightarrow v_i$ é executada no r_j , $X_{ji} = 0$ caso contrário.

Um exemplo que ilustra o conceito de uma matriz de mapeamento é dado por:

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (\text{A.11})$$

A Matriz (A.11) é um exemplo de mapeamento no qual o servidor r_0 hospeda duas máquinas virtuais v_0 e v_3 , o servidor r_1 hospeda v_1 e v_2 , e por último o servidor r_2 não hospeda nenhuma VM, e poderia ser desligado. A Equação (A.12) garante que uma VM deve pertencer somente a uma SF em um determinado momento:

$$\forall i, 1 \leq i \leq V : \sum_{j=1}^R X_{ji} = 1 \quad (\text{A.12})$$

A.2 Avaliação do Mapeamento

Dada uma solução válida para o problema combinatório representado por uma matriz de mapeamento, a utilização de todos os servidores produzirá um mapeamento que deverá ser avaliado. A utilização dos recursos serão comparados com limites configuráveis, que serão utilizados para identificar os servidores sobrecarregados.

O cálculo da utilização de um servidor é feito pela multiplicação da linha correspondente, na matriz de mapeamento com um vetor que representa os requisitos de recursos de cada máquina

virtual numa categoria de recurso específico. Com base na matriz de mapeamento da Equação (A.11), segue um exemplo de requisito de memória:

$$\vec{v}_{mem} = \begin{pmatrix} 300 \\ 650 \\ 1140 \\ 470 \end{pmatrix}, \quad (\text{A.13})$$

a utilização de memória do servidor \vec{r}_{mem} é calculado por:

$$\begin{aligned} \vec{r}_{mem} &= X \cdot \vec{v}_{mem} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 300 \\ 650 \\ 1140 \\ 470 \end{pmatrix} = \\ &= \begin{pmatrix} 1 \cdot 300 + 0 \cdot 650 + 0 \cdot 1140 + 1 \cdot 470 \\ 0 \cdot 300 + 1 \cdot 650 + 1 \cdot 1140 + 0 \cdot 470 \\ 0 \cdot 300 + 0 \cdot 650 + 0 \cdot 1140 + 0 \cdot 470 \end{pmatrix} = \\ &= \begin{pmatrix} 770 \\ 1790 \\ 0 \end{pmatrix} \end{aligned} \quad (\text{A.14})$$

A.3 Função Objetivo

O problema de otimização combinatória multiobjetivo deve ser resolvido pela formulação de uma função alvo ou de custo que será minimizado. A função de custo é composta por várias partes que representam diferentes categorias e questões. Cada parte contribui para o custo total com um peso configurável (denotado como penalidade), para ser capaz de examinar diferentes prioridades. Por exemplo, a disposição para economizar energia de uma forma extremamente agressiva com possíveis efeitos colaterais em um curto período de tempo em servidores sobrecarregados irá variar entre os centro de dados. Para formalizar, a função objetivo é minimizar o custo total em um mapeamento C

$$C = \sum_k c_k \cdot w_k, \quad (\text{A.15})$$

Em que:

c_k é o custo na categoria k ;

w_k é o peso para o custo correspondente na categoria k .

A função alvo e seus parâmetros serão descritos posteriormente, usando-se uma abordagem orientada para a consolidação dinâmica do modelo de custo.

O escalonador deverá consolidar máquinas virtuais em um menor número possível de servidores físicos, reordenar a topologia da rede quando possível e escolher máquinas físicas localizadas em regiões estratégicas do centro de dados para aumentar a eficiência do sistema de refrigeração.

De maneira formal, tem-se as seguintes notações para o modelo de energia do problema proposto:

$X_{i,j}$: indica máquina virtual i no servidor j ;

$u(X_{i,j})$: normaliza a utilização da CPU pela máquina virtual i no servidor j ;

S_j^{ocioso} : consumo de energia estático do servidor j ;

S_j^{din} : consumo máximo de energia dinâmica do servidor j ;

mv_p_i : quantidade de processadores contratado para a máquina virtual i ;

mv_m_i : quantidade de memória contratado para a máquina virtual i ;

mv_r_i : quantidade de largura de banda contratado para a máquina virtual i ;

sv_p_j : quantidade de processadores do servidor j ;

sv_m_j : quantidade de memória do servidor j ;

sv_r_j : quantidade de largura de banda do servidor j ;

TE_j : temperatura de entrada do servidor j ;

TS_j : temperatura de saída do servidor j ;

L_j^{rack} : determina qual *rack* está localizado o servidor j ;

$L_j^{posicao}$: determina qual posição no *rack* está localizado o servidor j ;

w_j : número de máquinas virtuais atribuído ao servidor j ;

C_k^{chassi} : energia consumida pelo chassi do comutador k ;

$C_{l,k}^{porta}$: energia consumida da porta l no comutador k ;

P_k : número de portas ativas do comutador k ;

M : número total de servidores;

V : número total de máquinas virtuais;

K : número total de comutadores;

E : número total de enlaces no centro de dados;

f_a : número total de fluxos no *enlace* a ;

$t_f^{enlace_a}$: a taxa do fluxo de dados f no *enlace* a ;

C^{enlace_a} : capacidade do enlace a .

Como exemplo: $u(X_{i,j}) = 0,5$ significa que a VM i está utilizando 50% da CPU no servidor

j .

O calculo do consumo energético do servidor j é baseado no modelo proposto por Lefurgy et al. (LEFURGY; WANG; WARE, 2007), no qual é considerado o percentual de utilização do processador como mostra a Equação (A.16).

$$P_j^{servidor} = \begin{cases} S_j^{ocioso} + \sum_{j=1}^{w_j} u(X_{ij}) \cdot S_j^{din}, & w_j > 0; \\ 0, & w_j = 0 \end{cases} \quad (\text{A.16})$$

O consumo energético de um servidor j é calculado pela soma do consumo de energia do servidor em modo ocioso, que é uma constante, mais o consumo de energia dinâmico que é linear e depende da quantidade de VMs alocadas no servidor. Quando não existir nenhuma VM atribuída ao servidor, o servidor pode ser desligado e seu consumo é zero.

O consumo energético de um comutador k é calculado pelo modelo apresentado por Wang et al. (WANG et al., 2012), como mostra a Equação (A.17).

$$P_k^{comutador} = \begin{cases} C_k^{chassi} + \sum_{l=1}^{P_k} C_{l,k}^{porta}, & \text{se comutador } k \text{ está ligado;} \\ 0, & \text{se comutador } k \text{ está desligado} \end{cases} \quad (\text{A.17})$$

A energia total do comutador k é a soma da energia consumida pelo chassi, que é constante, mais a energia consumida por todas as portas ativas.

Cada máquina virtual precisa ser alocada em um servidor de forma que haja uma distribuição de calor uniforme no centro de dados. Assumindo-se que o sistema de refrigeração fornece fluxo de ar constante no interior dos servidores, o calor gerado a partir de cada posição pode ser descrita por uma matriz distribuição de calor \mathbf{D} , onde cada elemento $d_{x,y} \in \mathbf{D}$ indica a fração de calor gerado pelo servidor no $rack$ $x \in L^{rack} = 1, \dots, M$ na posição $y \in L^{posicao} = 1, \dots, M$ como mostra a Equação (A.18).

$$\sum_{y=1}^M d_{x,y} \leq 1 \quad \forall x, 1 \leq x \leq M \quad (\text{A.18})$$

O consumo de energia é completamente transformado em calor, em que a quantidade total de calor (ou energia) recebida pela posição $L_y^{posicao}$ pode ser expressa pela Equação (A.19).

$$U_y^{saida} = \sum_{x=1}^M d_{x,y} \cdot P_x^{servidor} \quad (\text{A.19})$$

A temperatura de entrada TE_y , e a temperatura de saída TS_y é dado pela Equação (A.20).

$$TS_y^{saida} = TE_y + g(U_y^{saida}) \quad (\text{A.20})$$

A função $g(U)$ converte a quantidade de calor U (em Watt) em aumento de temperatura. A temperatura do ar é calculado em função da densidade do ar ρ (em kg/m^3), a velocidade do

fluxo de ar Q (em m^3/s), e a capacidade de calor do ar C (em $Joule/(C \cdot kg)$), como mostra a Equação (A.21) (TANG; GUPTA; VARSAMOPOULOS, 2008).

$$g(U) = \frac{U}{\rho \cdot Q \cdot C} \quad (\text{A.21})$$

A função objetivo é:

$$\text{Minimizar } z = \sum_{j=1}^M P_i^{server} + \sum_{k=1}^K P_k^{switch} + \max_{y=1 \dots M} (TS_y + g(d_{x,y} \cdot E_j^{din})), (\text{A.22})$$

sujeito a

$$\sum_{j=1}^M X_{ij} \leq 1, \quad \forall i \in \{1, V\} \quad (\text{A.23})$$

$$\sum_{i=1}^V mv_p_i \cdot X_{ij} \leq sv_p_j, \quad \forall j \in \{1, M\} \quad (\text{A.24})$$

$$\sum_{i=1}^V mv_m_i \cdot X_{ij} \leq sv_m_j, \quad \forall j \in \{1, M\} \quad (\text{A.25})$$

$$\sum_{i=1}^V mv_r_i \cdot X_{ij} \leq sv_r_j, \quad \forall j \in \{1, M\} \quad (\text{A.26})$$

$$\sum_{j=1}^M w_j = V, \quad (\text{A.27})$$

$$\sum_{i=1}^{w_j} u(X_{i,j}) \leq 1, \quad \forall j \in \{1, M\} \quad (\text{A.28})$$

$$TS_j + \sum_{i=1}^V + g(U_i^{saida}) \cdot X_{ij} \geq 15, \quad (\text{A.29})$$

$$\sum_{j=1}^M TS_j + g(U_j^{saida}) \leq 32, \quad (\text{A.30})$$

$$\sum_{f=1}^{f_a} t_f^{enlace_a} \leq C^{enlace_a}, \quad \forall a \in \{1, E\} \quad (\text{A.31})$$

A função objetivo (A.22) é minimizar o consumo energético dos servidores, comutadores e temperatura de saída. A Restrição (A.23) impede que uma máquina virtual seja hospedada em mais de um servidor. As Restrições (A.24), (A.25) e (A.26) garantem que a alocação das máquinas virtuais não extrapole os recursos físicos do servidor. A Restrição (A.28) impõe que a utilização do servidor deve ser inferior a 100%. As Restrições (A.29) e (A.30) impõem as temperaturas mínimas e máximas de saída de um servidor conforme recomendação da ASHRAE (ASHRAE, 2014). A Restrição (A.31) assegura que em qualquer enlace a taxa total de dados de todos os fluxos usados nestes enlaces não devem exceder a capacidade do enlace.

APÊNDICE B

PARÂMETROS DO SIMULADOR SIMDC3D

O arquivo de parâmetro do SimDC3D possui 137 parâmetros dividido em 14 categorias como pode-se observar no código abaixo. Cada categoria é um agrupamento de parâmetros correlacionados que parametriza uma determinada funcionalidade do simulador. A seguir, descreve-se a função de cada parâmetro do arquivo de configuração.

```

1 // 1 - Settings General
2 PERIODIC_LOG_INTERVAL = 86400;
3 SKIP_x_SECONDS = 0;
4 FINISHES_AT_DAY = 0;
5 PRINTSCREEN = false;
6 RUN_MULTITHREAD = true;
7 NUMBER_OF_THREADS = 9;
8 SEED = 12345;
9 SUBMIT_TIME_SEC = 31526918;
10
11 // 2 - Setting Server
12 NUMBER_OF_CHASSIS = 50;
13 NUMBER_OF_SERVERS_IN_ONE_CHASSIS = 10;
14 NUMBER_OF_CORES_IN_ONE_SERVER = 10;
15 TOTAL_OF_MEMORY_IN_ONE_SERVER = 128000000;
16 BANDWIDTH_IN_ONE_SERVER = 1000000;
17 CONSTANT_FAN_POWER = false;
18 FAN_RPM_NO_LIMIT = false;
19 POWER_ON = 70;
20 POWER_OFF = 30;
21 HIBERNATING_SERVER = true;
22 POWER_CONSUMPTION_HIBERNATING = 5.0;
23 HIBERNATING_POWER_ON = 10;
24 HIBERNATING_POWER_OFF = 20;
25 MODEL_OF_ENERGY_CONSUMPTION = linear_model;
26
27 // 3 - Setting Virtual Machines
28 MEMORY_VIRTUAL_MACHINE = 1000000,2000000,4000000;
29 IMAGE_SIZE_VIRTUAL_MACHINE = 204800;
30 V_TRESHOLD = 1000000;
31 MEMORY_DIRTYING = 100000;
32 T_RESUME = 5;
33

```



```
34 // 4 - Setting CRAC
35 CRAC_SUPPLY_TEMP_IS_CONSTANT_AT = 0;
36 SUPPLY_TEMPERATURE_OFFSET_ALPHA = 0;
37 INSTANT_COOL_AIR_TRAVEL_TIME = false;
38 INSTANT_CHANGE_CRAC_SUPPLY_AIR = false;
39 CRAC_DISCHARGE_CHANGE_RATE_0_00x = 10;
40 POLICY_OF_TEMPERATURE_CONTROL = Aggressive_Mode;
41 EMERGENCY_TEMPERATURE = 29.999;
42
43 // 5 - Setting Scheduling Algorithm
44 SCHEDULING_ALGORITHM = 3DMOBFD_POOL_AND_PREDICTION;
45 SCHEDULING_WITH_PREDICTION = true;
46 OPTIMIZATION_WITH_PREDICTION = false;
47 PREDICTION_ALGORITHM = RBF;
48 VARIABLE_PREDICTED = temperature;
49
50 SCHEDULER_2D_WEIGHT_TEMPERATURE = 0.40;
51 SCHEDULER_2D_WEIGHT_HEAT_RECIRCULATION = 0.1;
52 SCHEDULER_2D_WEIGHT_LOAD_CPU = 0.50;
53
54 // 6 - Parameters Prediction Algorithm RBF and POLYNOM
55 SIZE_WINDOWDOWN_PREDICTION = 1200;
56 MONITORINGTIME = 60;
57
58 // 7 - parameters prediction algorithm RBF
59 DIM = 2;
60 DELAY = 1;
61 CENTER = 10;
62 STEP = 1;
63 COLUMN = 1;
64 CLENGTH = 5;
65 MAKECAST = 1;
66
67 // 8 - Parameters Prediction Algorithm POLYNOM
68 N = 2;
69 CAST = 1;
70
71 // 9 - Setting Workload
72 DYNAMIC_WORKLOAD_VMS = true;
73 WORKLOAD_DISTRIBUTION_MODEL = uniform_real_distribution;
74
75 // 10 - Setting Migration Parameters
76 SIMULATES_MIGRATION_VMS = true;
77
78 OPTIMIZATION_ALGORITHM_LOW_UTILIZATION = 3DMOBFD;
79 PERIODIC_INTERVAL_EXECUTION_LOW = 30;
80
81 OPTIMIZATION_ALGORITHM_OVERLOAD_UTILIZATION = 3DMOBFD;
82 PERIODIC_INTERVAL_EXECUTION_OVERLOAD = 10;
83
84 OPTIMIZATION_ALGORITHM_OVERHEATING_TEMPERATURE = High_Temperature;
85
86 PERIODIC_INTERVAL_EXECUTION_TEMPERATURE = 15;
87
88 OPTIMIZATION_ALGORITHM_CORRELATION = DEFAULT;
89 PERIODIC_INTERVAL_EXECUTION_CORRELATION = 60;
90
91 OPTIMIZATION_ALGORITHM_LINK_OVERLOAD = DEFAULT;
92 PERIODIC_INTERVAL_EXECUTION_LINK_OVERLOAD = 15;
```

```
92 VMS_ALGORITHM_SELECTION_OVERLOADED_LINK = higher_IO_network;
93
94 VMS_ALGORITHM_SELECTION =
    HIGHER_UTILIZATION_AVERAGE_CPU/LESS_MEMORY;
95
96 THRESHOLD_TOP_USE_OF_CPU = 0.90;
97 THRESHOLD_BOTTOM_USE_OF_CPU = 0.30;
98
99 USING_PREDICTION_TO_OVERLOAD = true;
100 PREDICTION_ALGORITHM_OVERLOAD = LR;
101 SIZE_WINDOWDOWN_PREDICTION_CPU = 60;
102
103 // 11 - Setting 3DMOBFD Algorithm Parameters
104 ALPHA_3DMOBFD = 1;
105 BETA_3DMOBFD = 1;
106 GAMMA_3DMOBFD = 1;
107 DELTA_3DMOBFD = 1;
108 EPSILON_3DMOBFD = 0;
109
110 WEIGHT_TEMPERATURE = 0,40;
111 WEIGHT_CPU = 0,25;
112 WEIGHT_POWER = 0,25;
113 WEIGHT_MEMORY = 0,1;
114 WEIGHT_TRAFFIC = 0,0;
115
116 E_TEMPERATURE = 3;
117 E_CPU = 1;
118 E_POWER = 1;
119 E_MEMORY = 1;
120 E_TRAFFIC = 2;
121
122 // 12 - Setting Network Parameters
123 SIMULATES_NETWORK = true;
124
125 TYPE_TOPOLOGY = 3LAYER;
126 NAME_FILE_TOPOLOGY = 3camadasfull.xml;
127
128 NUMBER_CHASSI_RACK = 4;
129 NUMBER_OF_PORTS_SWITCH_ACCESS = 48;
130 NUMBER_OF_PORTS_SWITCH_AGGREGATION = 24;
131 NUMBER_OF_PORTS_SWITCH_CORE = 24;
132
133 EACCE_CHASSIS = 70;
134 EACCE_LINECARD = 0;
135 EACCE_PORT10 = 0.20;
136 EACCE_PORT100 = 0.30;
137 EACCE_PORT1000 = 0.65;
138 EACCE_PORT10000 = 1.30;
139
140 EAGGR_CHASSIS = 70;
141 EAGGR_LINECARD = 0;
142 EAGGR_PORT10 = 0.20;
143 EAGGR_PORT100 = 0.30;
144 EAGGR_PORT1000 = 0.65;
145 EAGGR_PORT10000 = 1.30;
146
147 E_CORE_CHASSIS = 147;
148 E_CORE_LINECARD = 38;
149 E_CORE_PORT10 = 0.12;
```

```
150 Ecore_PORT100 = 0.18;
151 Ecore_PORT1000 = 0.89;
152 Ecore_PORT10000 = 1.60;
153
154 // 13 - Parameters Traffic Matrix
155 SIMULATE_TRAFFIC_MATRIX = true;
156 NAME_FILE_MATRIX_TRAFFIC =
    /mnt/f/matriztrafego/cyclostationary48.xml;
157
158 CALCULATE_TRAFFIC_CORRELATION = false;
159 EXECUTE_LOAD_BALANCING = true;
160
161 NUMBER_NODES_TRAFFIC_MATRIX = 500;
162 NUMBER_OF_SAMPLES_TRAFFIC_MATRIX = 48;
163 NUMBER_OF_PERIODS = 1;
164 K = 3;
165
166 SEED_PERIOD_ON = 12345;
167 SEED_PERIOD_OFF = 56789;
168
169 PROPORTION_OF_INTERNAL_COMMUNICATION_BETWEEN_VMS = 30;
170 PROPORTION_OF_EXTERNAL_COMMUNICATION = 70;
171
172 DYNAMIC_SPEED_ADAPTATION = true;
173 RUNTIME_LINK_ADAPTATION = 180;
174 TRAFFIC_LIMIT_FOR_ADAPTATION = 80;
175
176 SIMULATE_TOPOLOGY_OPTIMIZATION = true;
177 OPTIMIZATION_ALGORITHM_TOPOLOGY = CARPO;
178 PERIODIC_INTERVAL_EXECUTION_TOPOLOGY = 600;
179
180 // 14 - Setting SERVER POOL
181 POOL_SIZE = 30;
182 INCREASE_POOL_SIZE_DYNAMICALLY = false;
183
```

- Categoria 01: **Settings General**

1. **PERIODIC_LOG_INTERVAL:** (INTEIRO) - Determina em **segundos** o período no qual as informações serão consolidadas no final da simulação, por padrão 86400s (1 dia).
2. **SKIP_x_SECONDS:** (INTEIRO) - Especifica o tempo em **segundos** que será pulado do tempo inicial em uma simulação.
3. **FINISHES_AT_DAY:** (INTEIRO) - Define a quantidade em **dias** que uma simulação será executada. Caso seja 0, a simulação executa todo o traço.
4. **PRINTSCREEN:** (BOOLEANO) - Caso seja *TRUE*, habilita modo de depuração. Caso contrário, desabilita.
5. **RUN_MULTITHREAD:** (BOOLEANO) - Caso seja *TRUE*, o SimDC3D executará algumas rotinas em paralelo durante uma simulação, a quantidade de *Threads* que serão

criadas para executar essas rotinas é definida no parâmetro (NUMBER_OF_THREADS). Caso contrário, a simulação será executada em modo *single thread*.

6. **NUMBER_OF_THREADS:** (INTEIRO) - Se a simulação for executada em modo *Multithread* esse parâmetro define a quantidade de *Threads* que serão criadas para cada uma das sub-rotinas que serão executadas em paralelo.
7. **SEED:** (INTEIRO) - Determina um número que será somando com o tempo do sistema para gerar uma semente para um gerador de número aleatórios. Os número são gerados a cada segundo e são usados no cálculo para determinar a utilização de CPU de uma máquina virtual.
8. **SUBMIT_TIME_SEC:** (INTEIRO) - O tempo de chegada da primeira carga de trabalho extraído do traço precisa ser ajustado com o tempo de simulação. O tempo da chegada da primeira carga de trabalho deverá ser subtraído de 1 e informado nesse parâmetro.

- Categoria 02: **Setting Server**

1. **NUMBER_OF_CHASSIS:** (INTEIRO) - Determina o número total de chassis do centro de dados.
2. **NUMBER_OF_SERVERS_IN_ONE_CHASSIS:** (INTEIRO) - Define o total de servidores em cada chassi.
3. **NUMBER_OF_CORES_IN_ONE_SERVER:** (INTEIRO) - Estabelece o número total de processadores em cada servidor.
4. **TOTAL_OF_MEMORY_IN_ONE_SERVER:** (INTEIRO) - Indica o total de memória em **Kbytes** de um servidor.
5. **BANDWIDTH_IN_ONE_SERVER:** (INTEIRO) - Determina a velocidade da interface de rede em **Kbps**. Os valores possíveis para esse parâmetro são: 10000, 100000, 1000000, 10000000.
6. **CONSTANT_FAN_POWER:** (BOOLEANO) - Caso seja *TRUE*, indica que o ventilador do processador terá rotação constante, independente da carga de processamento. Caso contrário, a rotação do ventilador será ajustado de acordo com a carga de processamento.
7. **FAN_RPM_NO_LIMIT:** (BOOLEANO) - Caso seja *FALSE*, a rotação máxima dos ventiladores é de 3000 rpm. Caso contrário, o parâmetro **SUPPLY_TEMPERATURE_OFFSET_ALPHA** tem que ser maior que 0.

8. **POWER_ON**: (INTEIRO) - Define o tempo em **segundos** para que um servidor seja ligado.
9. **POWER_OFF**: (INTEIRO) - Estabelece o tempo em **segundos** para que um servidor seja desligado.
10. **HIBERNATING_SERVER**: (BOOLEANO) - Se for *TRUE*, durante a simulação um servidor entrará em hibernação, quando for selecionado para desligar consumindo uma pequena quantidade de energia definido no parâmetro *POWER_CONSUMPTION_HIBERNATING*. Caso contrário, o servidor será desligado e não consumirá nenhuma energia.
11. **POWER_CONSUMPTION_HIBERNATING**: (REAL) - Determina o consumo energético do servidor em **Watts**, quando está em modo de hibernação.
12. **HIBERNATING_POWER_ON**: (INTEIRO) - Define o tempo em **segundos** para que um servidor saia de hibernação.
13. **HIBERNATING_POWER_OFF**: (INTEIRO) - Estabelece o tempo em **segundos** para que um servidor entre em hibernação.
14. **MODEL_OF_ENERGY_CONSUMPTION**: (STRING) - Indica o modelo que será utilizado para calcular o consumo energético do servidor. Atualmente, existem dois modelos, "*Linear_Model*" e "*PowerModelSpecPower*".

- Categoria 03: **Setting Virtual Machines**

1. **MEMORY_VIRTUAL_MACHINE**: (CONJUNTO INTEIRO) - Define um conjunto de valores em **Kbytes** que serão sorteados para determinar a quantidade de memória de uma máquina virtual. Caso o arquivo de traço traga informações do quantitativo de memória utilizado pela carga de trabalho, o valor será somado com o valor sorteado.
2. **IMAGE_SIZE_VIRTUAL_MACHINE**: (INTEIRO) - Determina o tamanho da imagem em **Kbytes** das máquinas virtuais. Esse valor fará parte do cálculo do tempo de migração de um VM entre dois servidores.
3. **V_TRESHOLD**: (INTEIRO) - Define o limite da memória em **Kbytes** na pré-cópia. A utilização desse parâmetro e dos dois próximos são descritos na Seção 4.4.1.
4. **MEMORY_DIRTYING**: (INTEIRO) - Estabelece a taxa de memória suja em **Kbytes**.
5. **T_RESUME**: (INTEIRO) - Determina o tempo em **segundos** necessário para reiniciar uma máquina virtual.

- Categoria 04: **Setting CRAC**

1. **CRAC_SUPPLY_TEMP_IS_CONSTANT_AT:** (INTEIRO) - Caso seja informado **zero**, a temperatura do ar condicionado será determinado pelo algoritmo indicado no parâmetro *POLICY_OF_TEMPERATURE_CONTROL*. Se o valor for diferente de zero o ar condicionado será ajustado com a temperatura em °C definida nesse parâmetro.
2. **SUPPLY_TEMPERATURE_OFFSET_ALPHA:** (INTEIRO) - Define a temperatura em °C que será utilizado como *alpha* para calcular o valor de gatilho. $T_{trigger} = T_{emergency} + alpha$. Esse parâmetro permite somente valores entre -10 a 50.
3. **INSTANT_COOL_AIR_TRAVEL_TIME:** (BOOLEANO) - Caso seja *TRUE*, o ar frio fornecido pelo sistema de refrigeração chegará instantaneamente nos servidores. Caso contrário, o SimDC3D calcula um tempo de propagação em **segundos** para que o ar gelado chegue nos servidores.
4. **INSTANT_CHANGE_CRAC_SUPPLY_AIR:** (BOOLEANO) - Caso seja *TRUE*, a temperatura do ar condicionado será ajustado como sendo, $EMERGENCY_TEMPERATURE - MaiorTemperaturaCentroDados$. Caso contrário, a temperatura do ar condicionado será determinado pela política definido no parâmetro *POLICY_OF_TEMPERATURE_CONTROL*.
5. **CRAC_DISCHARGE_CHANGE_RATE_0_00x :** (INTEIRO) - Indica o percentual que a temperatura será ajustada para cima ou para baixo pelo algoritmo de controle de temperatura *DEFAULT*. O intervalo de valores aceito é (> 1 e ≤ 50).
6. **POLICY_OF_TEMPERATURE_CONTROL:** (STRING) - Determina a política de controle do termostato do ar condicionado. Atualmente, estão disponíveis duas políticas, *Aggressive_Mode* e *Default*.
7. **EMERGENCY_TEMPERATURE:** (REAL) - Estabelece uma temperatura em °C que será utilizado como referência para determinar se um servidor está com sobrecarga térmica. Um servidor com temperatura maior que *EMERGENCY_TEMPERATURE* é considerado superaquecido.

- Categoria 05: **Setting Scheduling Algorithms**

1. **SCHEDULING_ALGORITHM:** (STRING) - Determina o algoritmos que será utilizado pelo escalonador de máquinas virtuais. Os algoritmos disponíveis estão descritos na Seção 5.1.
2. **SCHEDULING_WITH_PREDICTION:** (BOOLEANO) - Caso seja *TRUE*, o escalonador utiliza previsão no escalonamento de máquinas virtuais, por meio desse parâmetro, o simulador armazena a série histórica da variável que deverá ser predita. Caso contrário, o escalonador não utiliza previsão.

3. **OPTIMIZATION_WITH_PREDICTION:** (BOOLEANO) - Caso seja *TRUE*, os algoritmos de consolidação de máquinas virtuais implementa predição. Caso contrário, não utiliza-se predição.
4. **PREDICTION_ALGORITHM:** (STRING) - Define o algoritmo de predição, atualmente o SimDC3D disponibiliza o algoritmo RBF e Polynom.
5. **VARIABLE_PREDICTED:** (STRING) - Define a variável que será predita. O SimDC3D disponibiliza temperatura, memória e tráfego de rede.
6. **SCHEDULER_2D_WEIGHT_TEMPERATURE:** (REAL) - Especifica o peso da temperatura que será utilizado para calcular o ranking do servidor. Esse parâmetro e os demais são específicos do escalonador 2D. O intervalo de valores é entre (≥ 0 e ≤ 1). A somatória do peso dos três parâmetros deverá ser igual a 1.
7. **SCHEDULER_2D_WEIGHT_HEAT_RECIRCULATION:** (REAL) - Determina o peso da recirculação de calor que será utilizado para calcular o ranking do servidor.
8. **SCHEDULER_2D_WEIGHT_LOAD_CPU:** (REAL) - Define o peso da CPU que será utilizado para calcular o ranking do servidor. O intervalo de valores é entre (≥ 0 e ≤ 1).

- Categoria 06: **Parameters Prediction Algorithm RBF and POLYNOM**

1. **SIZE_WINDOWDOWN_PREDICTION :** (INTEIRO) - Estabelece o tamanho do vetor que armazenará os valores da variável que será predita.
2. **MONITORINGTIME:** (INTEIRO) - Define o intervalo em **segundos** no qual a variável que será predita deverá ser lida e armazenada no vetor.

- Categoria 07: **Parameters Prediction Algorithm RBF**

1. **DIM:** (INTEIRO) - Define a dimensão.
2. **DELAY:** (INTEIRO) - Indica o atraso.
3. **CENTER:** (INTEIRO) - Estabelece o número de centros.
4. **STEP:** (INTEIRO) - Define o número de passos para o preditor.
5. **COLUMN:** (INTEIRO) - Define a coluna que será lida.
6. **CLENGTH:** (INTEIRO) - Indica a posição do vetor predito.
7. **MAKECAST:** (INTEIRO) - Valor constante (1), como descrito no algoritmo.

- Categoria 08: **Parameters Prediction Algorithm POLYNOM**

1. **N:** (INTEIRO) - Define a ordem polinomial.

2. **CAST**: (INTEIRO) - Valor constante (1), como descrito no algoritmo.

- Categoria 09: **Setting Workload**

1. **DYNAMIC_WORKLOAD_VMS**: (BOOLEANO) - Caso seja *TRUE*, uma máquina virtual terá uma variabilidade de utilização de CPU conforme descrito na Seção 4.3.1.2. Caso contrário, as máquinas virtuais terão uma utilização de CPU constante, sendo o valor extraído do traço.
2. **WORKLOAD_DISTRIBUTION_MODEL**: (STRING) - Define o modelo de distribuição do gerador de números aleatórios que é utilizado para calcular a variabilidade de utilização de CPU, como descrito na Seção 4.3.1.2.

- Categoria 10: **Setting Migration Parameters**

1. **SIMULATES_MIGRATION_VMS**: (BOOLEANO) - Caso seja *TRUE*, o SimDC3D simula a migração de máquinas virtuais e executa as otimizações parametrizadas. Caso contrário, as máquinas virtuais são escalonadas em um servidor quando chegam no centro de dados e ficam até o fim da execução da carga de trabalho.
2. **OPTIMIZATION_ALGORITHM_LOW_UTILIZATION**: (STRING) - Define o algoritmo que será utilizado na otimização de máquinas virtuais quando um servidor está subutilizado, ($\% \text{ Processamento} < \text{THRESHOLD_BOTTOM_USE_OF_CPU}$).
3. **PERIODIC_INTERVAL_EXECUTION_LOW**: (INTEIRO) - Indica de quanto em quanto tempo em **segundos** será executado a otimização.
4. **OPTIMIZATION_ALGORITHM_OVERLOAD_UTILIZATION**: (STRING) - Define o algoritmo que será utilizado na otimização de máquinas virtuais quando um servidor está sobrecarregado, ($\% \text{ Processamento} \geq \text{THRESHOLD_TOP_USE_OF_CPU}$).
5. **PERIODIC_INTERVAL_EXECUTION_OVERLOAD**: (INTEIRO) - Indica de quanto em quanto tempo em **segundos** será executado a otimização.
6. **OPTIMIZATION_ALGORITHM_OVERHEATING_TEMPERATURE**: (STRING) - Define o algoritmo que será utilizado na otimização de máquinas virtuais quando um servidor está com sobrecarga térmica. ($\text{TemperaturaServidor} > \text{EMERGENCY_TEMPERATURE}$).
7. **PERIODIC_INTERVAL_EXECUTION_TEMPERATURE**: (INTEIRO) - Indica de quanto em quanto tempo em **segundos** será executado a otimização.
8. **OPTIMIZATION_ALGORITHM_CORRELATION**: (STRING) - Define o algoritmo que será utilizado na otimização de máquinas virtuais quando um servidor está com sobrecarga térmica. ($\text{TemperaturaServidor} > \text{EMERGENCY_TEMPERATURE}$).

9. **PERIODIC_INTERVAL_EXECUTION_CORRELATION:** (INTEIRO) - Indica de quanto em quanto tempo em **segundos** será executado a otimização.
10. **OPTIMIZATION_ALGORITHM_LINK_OVERLOAD:** (STRING) - Define o algoritmo que será utilizado na otimização de máquinas virtuais quando um servidor está com a interface de rede congestionado (Tráfego > CapacidadeMáximaInterface).
11. **PERIODIC_INTERVAL_EXECUTION_LINK_OVERLOAD :** (INTEIRO) - Indica de quanto em quanto tempo em **segundos** será executado a otimização.
12. **VMS_ALGORITHM_SELECTION_OVERLOADED_LINK :** (STRING) Define qual o método de ordenação para selecionar um ou mais VMs que irão migrar para outro servidor quando houver enlaces congestionados.
13. **VMS_ALGORITHM_SELECTION :** (STRING) - Define qual o método de ordenação para selecionar um ou mais VMs que irão migrar para outro servidor.
14. **THRESHOLD_TOP_USE_OF_CPU:** (REAL) - Determina o limite superior para que um servidor seja considerado sobrecarregado na utilização de CPU. Os valores deverão ser entre (> 0.0 e ≤ 1.0). Um servidor é considerado sobrecarregado se o (% Processamento \geq THRESHOLD_TOP_USE_OF_CPU).
15. **THRESHOLD_BOTTOM_USE_OF_CPU:** (REAL) - Estipula o limite inferior para definir se um servidor está subutilizado. Os valores deverão ser entre (> 0.0 e ≤ 1.0). Um servidor é considerado subutilizado se o (% Processamento $<$ THRESHOLD_BOTTOM_USE_OF_CPU).
16. **USING_PREDICTION_TO_OVERLOAD:** (BOOLEANO) - Caso seja *TRUE*, o simulador utiliza predição para prevenir sobrecarga de processamento. Uma vez que o preditor indicar que haverá uma sobrecarga futura o SimDC3D seleciona e migra uma máquina virtual conforme a estratégia definida em VMS_ALGORITHM_SELECTION. Caso contrário, não utiliza predição para prevenir sobrecarga de utilização de CPU.
17. **PREDICTION_ALGORITHM_OVERLOAD:** (STRING) - Define o algoritmo utilizado para calcular a predição.
18. **SIZE_WINDOWDOWN_PREDICTION_CPU:** (INTEIRO) - Determina o tamanho da janela que será utilizado para armazenar as séries temporais da variável utilização de CPU. Esses valores são coletados a cada segundo em cada servidor que esteja ligado no centro de dados.

- Categoria 11: **Setting 3DMOBFD Algorithm Parameters**

1. **ALPHA_3DMOBFD:** (INTEIRO) - Caso seja 1, a variável temperatura fará parte do cálculo do ranking. Caso contrário (0) não.

2. **BETA_3DMOBFD:** (INTEIRO) - Caso seja 1, a variável utilização de CPU fará parte do cálculo do ranking. Caso contrário (0) não.
3. **GAMMA_3DMOBFD:** (INTEIRO) - Caso seja 1, a variável consumo de energia fará parte do cálculo do ranking. Caso contrário (0) não.
4. **DELTA_3DMOBFD:** (INTEIRO) - Caso seja 1, a variável memória fará parte do cálculo do ranking. Caso contrário (0) não.
5. **EPSILON_3DMOBFD:** (INTEIRO) - Caso seja 1, a variável tráfego da rede fará parte do cálculo do ranking. Caso contrário (0) não.
6. **WEIGHT_TEMPERATURE:** (REAL) - Caso seja 1, todas as variáveis terão o mesmo peso. No caso do peso atribuído for entre (> 0 e < 1) a variável temperatura terá o peso escolhido, sendo que a somatória de todos os pesos deverá ser igual a 1.
7. **WEIGHT_CPU:** (REAL) - Caso seja 1, todas as variáveis terão o mesmo peso. No caso do peso atribuído for entre (> 0 e < 1) a variável utilização de CPU terá o peso escolhido, sendo que a somatória de todos os pesos deverá ser igual a 1.
8. **WEIGHT_POWER:** (REAL) - Caso seja 1, todas as variáveis terão o mesmo peso. No caso do peso atribuído for entre (> 0 e < 1) a variável consumo energético terá o peso escolhido, sendo que a somatória de todos os pesos deverá ser igual a 1.
9. **WEIGHT_MEMORY:** (REAL) - Caso seja 1, todas as variáveis terão o mesmo peso. No caso do peso atribuído for entre (> 0 e < 1) a variável memória terá o peso escolhido, sendo que a somatória de todos os pesos deverá ser igual a 1.
10. **WEIGHT_TRAFFIC:** (REAL) - Caso seja 1, todas as variáveis terão o mesmo peso. No caso do peso atribuído for entre (> 0 e < 1) a variável tráfego da rede terá o peso escolhido, sendo que a somatória de todos os pesos deverá ser igual a 1.
11. **E_TEMPERATURE:** (INTEIRO) - Define o expoente que será utilizado na variável temperatura para calcular o ranking do servidor.
12. **E_CPU:** (INTEIRO) - Define o expoente que será utilizado na variável utilização de CPU para calcular o ranking do servidor.
13. **E_POWER:** (INTEIRO) - Define o expoente que será utilizado na variável consumo de energia para calcular o ranking do servidor.
14. **E_MEMORY:** (INTEIRO) - Define o expoente que será utilizado na variável memória para calcular o ranking do servidor.
15. **E_TRAFFIC:** (INTEIRO) - Define o expoente que será utilizado na variável tráfego da rede para calcular o ranking do servidor.

- Categoria 12: **Setting Network Parameters**

1. **SIMULATES_NETWORK**: (BOOLEANO) - Caso seja *TRUE*, o SimDC3D simula a infraestrutura de rede. Caso contrário, não simula.
2. **NAME_FILE_TOPOLOGY**: (STRING) - Indica a localização (diretório) é o nome do arquivo no formato XML que contém a topologia de rede gerado pelo FNSS.
3. **NUMBER_CHASSI_RACK**: (INTEIRO) - Determina a quantidade que *chassis* que são alocados em cada *rack*. Esse parâmetro é utilizado para calcular a quantidade de *racks* e comutadores de acesso do centro de dados.
4. **NUMBER_OF_PORTS_SWITCH_ACCESS**: (INTEIRO) - Define o número de portas de um comutador de acesso.
5. **NUMBER_OF_PORTS_SWITCH_AGGREGATION**: (INTEIRO) - Estipula o número de portas de um comutador de agregação.
6. **NUMBER_OF_PORTS_SWITCH_CORE**: (INTEIRO) - Indica o número de portas de um comutador Core.
7. **EACCE_CHASSIS**: (REAL) - Determina o consumo energético em **Watts** do *chassi* nos comutadores de acesso.
8. **EACCE_LINECARD**: (REAL) - Define o consumo energético em **Watts** dos *linecards* nos comutadores de acesso.
9. **EACCE_PORT10**: (REAL) - Estabelece o consumo energético em **Watts** das portas dos comutadores de acesso na velocidade de 10 Mbps.
10. **EACCE_PORT100**: (REAL) - Estipula o consumo energético em **Watts** das portas dos comutadores de acesso na velocidade de 100 Mbps.
11. **EACCE_PORT1000** : (REAL) - Indica o consumo energético em **Watts** das portas dos comutadores de acesso na velocidade de 1000 Mbps.
12. **EACCE_PORT10000**: (REAL) - Define o consumo energético em **Watts** das portas dos comutadores de acesso na velocidade de 10000 Mbps.
13. **EAGGR_CHASSIS**: (REAL) - Estabelece o consumo energético em **Watts** do *chassi* nos comutadores de agregação.
14. **EAGGR_LINECARD**: (REAL) - Estipula o consumo energético em **Watts** dos *linecards* nos comutadores de agregação.
15. **EAGGR_PORT10**: (REAL) - Determina o consumo energético em **Watts** das portas dos comutadores de agregação na velocidade de 10 Mbps.

16. **EAGGR_PORT100**: (REAL) - Define o consumo energético em **Watts** das portas dos comutadores de agregação na velocidade de 100 Mbps.
17. **EAGGR_PORT1000**: (REAL) - Estabelece o consumo energético em **Watts** das portas dos comutadores de agregação na velocidade de 1000 Mbps.
18. **EAGGR_PORT10000**: (REAL) - Estipula o consumo energético em **Watts** das portas dos comutadores de agregação na velocidade de 10000 Mbps.
19. **ECORE_CHASSIS**: (REAL) - Determina o consumo energético em **Watts** do *chassi* nos comutadores de núcleo.
20. **ECORE_LINECARD**: (REAL) - Define o consumo energético em **Watts** dos *linecards* nos comutadores de núcleo.
21. **ECORE_PORT10**: (REAL) - Estabelece o consumo energético em **Watts** das portas dos comutadores de núcleo na velocidade de 10 Mbps.
22. **ECORE_PORT100**: (REAL) - Estipula o consumo energético em **Watts** das porta dos comutadores de núcleo na velocidade de 100 Mbps.
23. **ECORE_PORT1000**: (REAL) - Determina o consumo energético em **Watts** das portas dos comutadores de núcleo na velocidade de 1000 Mbps.
24. **ECORE_PORT10000**: (REAL) - Indica o consumo energético em **Watts** das portas dos comutadores de núcleo na velocidade de 10000 Mbps.

- Categoria 13: **Parameters Traffic Matrix**

1. **SIMULATE_TRAFFIC_MATRIX**: (BOOLEANO) - Caso seja *TRUE*, será habilitado a simulação da transmissão de dados utilizando uma matriz de tráfego sintética gerado no FNSS. Caso contrário, a simulação não terá tráfego na rede, zerando qualquer informação de tráfego nos enlaces e nas interfaces de rede.
2. **NAME_FILE_MATRIX_TRAFFIC**: (STRING) - Indica a localização (diretório) é o nome do arquivo no formato XML que contém a matriz de tráfego gerado pelo FNSS.
3. **CALCULATE_TRAFFIC_CORRELATION**: (BOOLEANO) - Caso seja *TRUE*, calcula a correlação de tráfego entre todas as máquinas virtuais do centro de dados. Caso contrário, não calcula.
4. **EXECUTE_LOAD_BALANCING**: (BOOLEANO) - Caso seja *TRUE*, o SimDC3D realiza o balanceamento de carga quando houver dois ou mais enlaces entre as camadas da topologia de rede. Caso contrário, não realiza o balanceamento de carga.

5. **NUMBER_NODES_TRAFFIC_MATRIX:** (INTEIRO) - Determina o número de nós da matriz de tráfego, esse parâmetro deve ser o mesmo informado no FNSS.
6. **NUMBER_OF_SAMPLES_TRAFFIC_MATRIX:** (INTEIRO) - Estipula a quantidade de matrizes de tráfego que serão geradas por período, esse parâmetro deve ser o mesmo informado no FNSS.
7. **NUMBER_OF_PERIODS:** (INTEIRO) - Indica o número de dias dentro do arquivo que contém as matrizes de tráfego, esse parâmetro deve ser o mesmo informado no FNSS.
8. **K:** (INTEIRO) - Determina o valor de K que será utilizado para gerar os números aleatórios da distribuição log-normal dos fluxos de dados.
9. **SEED_PERIOD_ON:** (INTEIRO) - Estipula um número que será usado como semente para inicializar o gerador de números aleatórios utilizado para calcular o tempo das rajadas de transmissão de uma máquina virtual. Esse número aleatório é somado juntamente com uma variável de tempo.
10. **SEED_PERIOD_OFF:** (INTEIRO) - Define um número que será usado como semente para inicializar o gerador de números aleatórios utilizado para calcular o tempo de inatividade depois de uma rajada de transmissão de uma máquina virtual. Esse número aleatório é somado juntamente com uma variável de tempo.
11. **PROPORTION_OF_INTERNAL_COMMUNICATION_BETWEEN_VMS:** (INTEIRO) - Indica entre todos os fluxos simulados qual o percentual que será entre máquinas virtuais.
12. **PROPORTION_OF_EXTERNAL_COMMUNICATION:** (INTEIRO) - Estipula entre todos os fluxos simulados qual o percentual que será entre as máquinas virtuais e o mundo exterior. A somatória o percentual de e comunicação interna e externa deverá ser igual a 100.
13. **DYNAMIC_SPEED_ADAPTATION:** (BOOLEANO) - Caso seja *TRUE*, o controlador executará adaptação de velocidades das portas de todos os comutadores de acordo com o tráfego de dados. Caso contrário, o controlador não ajustará a velocidade das portas dos comutadores.
14. **RUNTIME_LINK_ADAPTATION:** (INTEIRO) - Define de quanto em quanto tempo em **segundos** que ocorrerá a execução da adaptação de velocidades nos comutadores.
15. **TRAFFIC_LIMIT_FOR_ADAPTATION:** (INTEIRO) - Define um percentual que será utilizado para calcular o valor de referência. Esse valor será utilizado como referência para comparar com o tráfego atual, caso o tráfego seja menor que o valor de referência é modificado para baixo a velocidade do enlace. Uma descrição detalhada está descrito na Seção 6.2.

16. **SIMULATE_TOPOLOGY_OPTIMIZATION**: (BOOLEANO) - Caso seja *TRUE*, o SimDC3D executa um algoritmo que otimizará a topologia de rede. Caso contrário, não otimiza a topologia.
17. **OPTIMIZATIONZ_ALGORITHM_TOPOLOGY**: (STRING) - Define qual é o algoritmo que será executado para otimizar a topologia.
18. **PERIODIC_INTERVAL_EXECUTION_TOPOLOGY**: (INTEIRO) - Define de quanto em quanto tempo em **segundos** a otimização da topologia ocorrerá.

- Categoria 14: **Setting POOL**

1. **POOL_SIZE**: (INTEIRO) - Define a quantidade de servidores que serão reservados para serem utilizados no banco de servidores. Caso o valor seja zero, a simulação não utilizará o banco de servidores.
2. **INCREASE_SIZE_POOL_DYNAMICALLY**: (BOOLEANO) - Caso seja *TRUE*, o tamanho do banco de servidores será ajustado automaticamente. Caso contrário, o tamanho do banco é fixo e o tamanho é estipulado no parâmetro **SIZE_POOL**.

APÊNDICE C

IMPLEMENTAÇÃO DO ALGORITMO DE PREDIÇÃO RBF

```

1
2 /*PREDICTION ALGORITHM RBF
3  *
4  *This file is part of TISEAN
5  *
6  *Copyright (c) 1998-2007 Rainer Hegger, Holger Kantz,
7  *Thomas Schreiber
8  *
9  *TISEAN is free software; you can redistribute it and/or modify
10 *it under the terms of the GNU General Public License as published by
11 *the Free Software Foundation; either version 2 of the License, or
12 *(at your option) any later version.
13 *
14 *TISEAN is distributed in the hope that it will be useful,
15 *but WITHOUT ANY WARRANTY; without even the implied warranty of
16 *MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
17 *GNU General Public License for more details.
18 *
19 *You should have received a copy of the GNU General Public License
20 *along with TISEAN; if not, write to the Free Software
21 *Foundation, Inc., 51 Franklin St, Fifth Floor, Boston,
22 *MA 02110-1301 USA
23 *
24 *Author: Rainer Hegger. Last modified: Mar 11, 2002
25 *Author of the modifications : Marcos Paulo Moro
26 *Changes: 31/03/2015
27 */
28 #include <stdio.h>
29 #include <stdlib.h>
30 #include <string.h>
31 #include <limits.h>
32 #include "tisean/tsa.h"
33 #include <math.h>
34 #include <iostream>
35 #include <vector>
36 #include <time.h>
37
38 #include "SimDC3D-Constants.h"
39
40 using namespace std;
41
42 extern void rescale_data(double *x, unsigned long l, double *min,
43 double *interval);
44 extern void variance(double *s, unsigned long l, double *av, double *var);
45 extern void check_alloc(void *pnt);
46 extern void solvele(double **mat, double *vec, unsigned int n);
47

```

```

48 extern bool PRINTSCREEN;
49 extern int DIM;
50 extern int DELAY;
51 extern int CENTER;
52 extern int STEP;
53 extern unsigned int COLUMN;
54 extern unsigned long LENGTH;
55 extern long CLENGTH;
56 extern char MAKECAST;
57 extern int N;
58 extern char CAST;
59
60 #define WID_STR "Fits a RBF-model to the data"
61
62 char stdo=1;
63 char setdrift=1;
64
65 unsigned int verbosity=0xff;
66 unsigned long INSAMPLE=ULONG_MAX, exclude=0;
67
68 double *series,*coefs;
69 double varianz, interval, minimo;
70 double **center;
71
72 char sinsample=0;
73 unsigned int pars=1, hpar;
74 long *coding;
75 long maxencode;
76 double *results;
77 double std_dev;
78
79 vector<double> PREDICTION;
80
81 double avdistanceRBF(void)
82 {
83     int i, j, k;
84     double dist=0.0;
85
86     for (i=0; i<CENTER; i++) {
87         for (j=0; j<CENTER; j++) {
88             if (i != j) {
89                 for (k=0; k<DIM; k++) {
90                     dist += sqr(center[i][k]-center[j][k]);
91                 }
92             }
93         }
94     }
95     return sqrt(dist/(CENTER-1)/CENTER/DIM);
96 }
97
98 double rbf(double *act, double *cen)
99 {
100     static double denum = 0.00;
101     double r = 0.00;
102     int i = 0;
103
104     denum = 2.0*varianz*varianz;
105
106     for (i=0; i<DIM; i++){
107         r += sqr(*(act-i*DELAY)-cen[i]);
108     }
109     return exp(-r/denum);
110 }
111
112 void driftRBF(void)
113 {
114     double *force = NULL;
115     double h = 0.00;
116     double h1 = 0.00;
117     double step=1e-2;

```



```

118 double step1 = 0.00;
119 int i, j, k, l, d2=DIM;
120
121 check_alloc(force=(double*)malloc(sizeof(double)*d2));
122
123 for (l=0; l<20; l++) {
124     for (i=0; i<CENTER; i++) {
125         for (j=0; j<d2; j++) {
126             force[j]=0.0;
127             for (k=0; k<CENTER; k++) {
128                 if (k != i) {
129                     h=center[i][j]-center[k][j];
130                     force[j] += h/sqr(h)/fabs(h);
131                 }
132             }
133         }
134         h=0.0;
135         for (j=0; j<d2; j++) {
136             h += sqr(force[j]);
137         }
138         step1 = step/sqrt(h);
139         for (j=0; j<d2; j++) {
140             h1 = step1*force[j];
141             if (((center[i][j]+h1) > -0.1) &&
142                 ((center[i][j]+h1) < 1.1)) {
143                 center[i][j] += h1;
144             }
145         }
146     }
147 }
148 free(force);
149 }
150
151 void make_fitRBF(void)
152 {
153     double **mat, *hcen;
154     double h = 0.00;
155     int i, j, n, nst;
156
157     check_alloc(mat=(double**)malloc(sizeof(double)*(CENTER+1)));
158
159     for (i=0; i<=CENTER; i++) {
160         check_alloc(mat[i]=(double*)malloc(sizeof(double)*(CENTER+1)));
161     }
162
163     check_alloc(hcen=(double*)malloc(sizeof(double)*CENTER));
164
165     for (i=0; i<=CENTER; i++) {
166         coefs[i]=0.0;
167         for (j=0; j<=CENTER; j++) {
168             mat[i][j]=0.00;
169         }
170     }
171
172     for (n=(DIM-1)*DELAY; n<INSAMPLE-STEP; n++) {
173         nst=n+STEP;
174         for (i=0; i<CENTER; i++) {
175             hcen[i]=rbf(&series[n], center[i]);
176         }
177         coefs[0] += series[nst];
178         mat[0][0] += 1.0;
179         for (i=1; i<=CENTER; i++) {
180             mat[i][0] += hcen[i-1];
181         }
182         for (i=1; i<=CENTER; i++) {
183             coefs[i] += series[nst]*(h=hcen[i-1]);
184             for (j=1; j<=i; j++) {
185                 mat[i][j] += h*hcen[j-1];
186             }
187         }

```

```

188 }
189
190 h=(double) (INSAMPLE-STEP-(DIM-1)*DELAY);
191
192 for (i=0;i<=CENTER;i++) {
193     coefs[i] /= h;
194     for (j=0;j<=i;j++) {
195         mat[i][j] /= h;
196         mat[j][i]=mat[i][j];
197     }
198 }
199
200 solvele(mat,coefs,(unsigned int) (CENTER+1));
201
202 for (i=0;i<=CENTER;i++){
203     free(mat[i]);
204 }
205 free(mat);
206 free(hcen);
207 }
208
209 double forecast_errorRBF(unsigned long i0,unsigned long i1)
210 {
211     int i,n;
212     double h,error=0.0;
213
214     for (n=i0+(DIM-1)*DELAY;n<i1-STEP;n++) {
215         h=coefs[0];
216         for (i=1;i<=CENTER;i++)
217             h += coefs[i]*rbf(&series[n],center[i-1]);
218         error += (series[n+STEP]-h)*(series[n+STEP]-h);
219     }
220
221     return sqrt(error/(i1-i0-STEP-(DIM-1)*DELAY));
222 }
223
224 void make_castRBF()
225 {
226     double *cast,new_el = 0.00;
227     int i,n,dim = 0;
228
229     dim=(DIM-1)*DELAY;
230
231     check_alloc(cast=(double*)malloc(sizeof(double)*(dim+1)));
232
233     for (i=0;i<=dim;i++) {
234         cast[i]=series[LENGTH-1-dim+i];
235     }
236
237     for (n=0;n<CLENGTH;n++) {
238         new_el=coefs[0];
239         for (i=1;i<=CENTER;i++) {
240             new_el += coefs[i]*rbf(&cast[dim],center[i-1]);
241         }
242
243         if (PRINTSCREEN) {
244             cout << "# " << new_el*interval+minimo << endl;
245         }
246
247         PREDICTION.push_back(new_el*interval+minimo);
248
249         for (i=0;i<dim;i++) {
250             cast[i]=cast[i+1];
251         }
252         cast[dim]=new_el;
253     }
254
255     free(cast);
256 }
257

```

```

258 vector<double> runRBF(vector<double> vetorPredicao)
259 {
260     char stdi=0;
261     int i=0;
262     int j=0;
263     int cstep=0;
264     double sigma = 0.00;
265     double av = 0.00;
266
267     // initializing global variables
268     varianz = 0;
269     interval = 0;
270     minimo = 0;
271     INSAMPLE = ULONG_MAX;
272
273     //erase prediction vetor
274     PREDICTION.erase(PREDICTION.begin(), PREDICTION.end());
275
276     series=(double*)malloc(vetorPredicao.size()*sizeof(double));
277
278     for (int i=0; i < vetorPredicao.size(); i++) {
279         series[i] = vetorPredicao[i];
280     }
281
282     LENGTH = vetorPredicao.size();
283
284     rescale_data(series, LENGTH, &minimo, &interval);
285
286     variance(series, LENGTH, &av, &varianz);
287
288     if (INSAMPLE > LENGTH) {
289         INSAMPLE=LENGTH;
290     }
291
292     if (CENTER > LENGTH) {
293         CENTER = LENGTH;
294     }
295
296     if (MAKECAST) {
297         STEP=1;
298     }
299
300     check_alloc(coefs=(double*)malloc(sizeof(double)*(CENTER+1)));
301     check_alloc(center=(double**)malloc(sizeof(double*)*CENTER));
302     for (i=0; i<CENTER; i++){
303         check_alloc(center[i]=(double*)malloc(sizeof(double)*DIM));
304     }
305
306     cstep=LENGTH-1-(DIM-1)*DELAY;
307
308     for (i=0; i<CENTER; i++) {
309         for (j=0; j<DIM; j++) {
310             center[i][j] = series[(DIM-1)*DELAY-j*DELAY+(i*cstep) /
311                                 (CENTER-1)];
312         }
313     }
314     if (setdrift){
315         driftRBF();
316     }
317
318     varianz=avdistanceRBF();
319
320     make_fitRBF();
321
322
323     if (PRINTSCREEN) {
324         cout << "#Center points used: " << endl;
325         for (i=0; i<CENTER; i++) {
326             cout << "# ";
327             for (j=0; j<DIM; j++) {

```

```

328     cout << center[i][j]*interval+minimo << " ";
329     }
330     cout << endl;
331     }
332     cout << endl;
333     cout << "#variance= " << varianz*interval << endl;
334     cout << endl;
335     cout << "#Coefficients: " << endl;
336     cout << coefs[0]*interval+minimo << endl;
337     for (i=1; i<=CENTER; i++) {
338         cout << coefs[i]*interval << endl;
339     }
340     }
341
342     av = sigma = 0.0;
343
344     for (i=0;i<INSAMPLE;i++) {
345         av += series[i];
346         sigma += series[i]*series[i];
347     }
348
349     av /= INSAMPLE;
350
351     sigma=sqrt(fabs(sigma/INSAMPLE-av*av));
352
353     if (PRINTSCREEN) {
354         cout << "#insample error= " <<
355         forecast_errorRBF(OLU,INSAMPLE)/sigma << endl;
356     }
357
358     if (INSAMPLE < LENGTH) {
359         av=sigma=0.0;
360         for (i=INSAMPLE;i<LENGTH;i++) {
361             av += series[i];
362             sigma += series[i]*series[i];
363         }
364         av /= (LENGTH-INSAMPLE);
365         sigma=sqrt(fabs(sigma/(LENGTH-INSAMPLE)-av*av));
366         if (PRINTSCREEN) {
367             cout << "#out of sample error= " <<
368             forecast_errorRBF(INSAMPLE,LENGTH)/sigma << endl;
369         }
370     }
371
372     if (MAKECAST) {
373         make_castRBF();
374     }
375
376     free(series);
377
378     free(coefs);
379
380     for (i=0;i<CENTER;i++){
381         free(center[i]);
382     }
383
384     free(center);
385
386     return(PREDICTION);
387 }
388
389
390 void check_alloc(void *pnt)
391 {
392     if (pnt == NULL) {
393         fprintf(stderr,"check_alloc: Couldn't allocate enough memory.
394             Exiting\n");
395         exit(CHECK_ALLOC_NOT_ENOUGH_MEMORY);
396     }
397 }

```

```

398
399 void variance(double *s,unsigned long l,double *av,double *var)
400 {
401     unsigned long i=0;
402     double h=0.0;
403
404     *av=0.0;
405     *var=0.0;
406
407     for (i=0;i<l;i++) {
408         h=s[i];
409         *av += h;
410         *var += h*h;
411     }
412
413     *av /= (double)l;
414     *var=sqrt(fabs((*var)/(double)l-(*av)*(*av)));
415
416     if (*var == 0.0) {
417         fprintf(stderr,"Variance of the data is zero. Exiting!\n\n");
418         exit(VARIANCE_VAR_EQ_ZERO);
419     }
420 }
421
422
423 void solvele(double **mat,double *vec,unsigned int n)
424 {
425     double vswap=0.0,*mswap,*hvec,max=0.0,h=0.0,pivot=0.0,q=0.0;
426     int i=0,j=0,k=0,maxi=0;
427
428     for (i=0;i<n-1;i++) {
429         max=fabs(mat[i][i]);
430         maxi=i;
431         for (j=i+1;j<n;j++)
432             if ((h=fabs(mat[j][i])) > max) {
433                 max=h;
434                 maxi=j;
435             }
436         if (maxi != i) {
437             mswap=mat[i];
438             mat[i]=mat[maxi];
439             mat[maxi]=mswap;
440             vswap=vec[i];
441             vec[i]=vec[maxi];
442             vec[maxi]=vswap;
443         }
444
445         hvec=mat[i];
446         pivot=hvec[i];
447         if (fabs(pivot) == 0.0) {
448             fprintf(stderr,"Singular matrix! Exiting!\n");
449             exit(SOLVELE_SINGULAR_MATRIX);
450         }
451         for (j=i+1;j<n;j++) {
452             q= -mat[j][i]/pivot;
453             mat[j][i]=0.0;
454             for (k=i+1;k<n;k++) {
455                 mat[j][k] += q*hvec[k];
456             }
457             vec[j] += q*vec[i];
458         }
459     }
460     vec[n-1] /= mat[n-1][n-1];
461     for (i=n-2;i>=0;i--) {
462         hvec=mat[i];
463         for (j=n-1;j>i;j--) {
464             vec[i] -= hvec[j]*vec[j];
465         }
466         vec[i] /= hvec[i];
467     }

```


APÊNDICE D

IMPLEMENTAÇÃO DO ALGORITMO DE PREDIÇÃO POLYNOM

```

1
2 /*PREDICTION ALGORITHM POLYNOM
3 *
4 *This file is part of TISEAN
5 *
6 *Copyright (c) 1998-2007 Rainer Hegger, Holger Kantz,
7 *Thomas Schreiber
8 *
9 *TISEAN is free software; you can redistribute it and/or modify
10 *it under the terms of the GNU General Public License as published by
11 *the Free Software Foundation; either version 2 of the License, or
12 *(at your option) any later version.
13 *
14 *TISEAN is distributed in the hope that it will be useful,
15 *but WITHOUT ANY WARRANTY; without even the implied warranty of
16 *MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
17 *GNU General Public License for more details.
18 *
19 *You should have received a copy of the GNU General Public License
20 *along with TISEAN; if not, write to the Free Software
21 *Foundation, Inc., 51 Franklin St, Fifth Floor, Boston,
22 *MA 02110-1301 USA
23 *
24 *Author: Rainer Hegger
25 *Changes:
26 *06/30/2006: Norm of the errors was wrong
27 *
28 *Author of the modifications : Marcos Paulo Moro
29 *Changes: 31/03/2015
30 */
31
32 double polynom(int act,int dim,long cur,long fac)
33 {
34     int j=0,n=0,hi=0;
35     double ret=1.0;
36
37     n=cur/fac;
38     hi=act-(dim-1)*DELAY;
39     for (j=1;j<=n;j++) {
40         ret *= series[hi];
41     }
42     if (dim > 1) {
43         ret *= polynom(act,dim-1,cur-n*fac,fac/(N+1));
44     }

```

```

45 return ret;
46 }
47
48 int number_parsPOLYNOM(int ord,int start)
49 {
50     int i=0,ret=0;
51
52     if (ord == 1) {
53         for (i=start; i<=DIM; i++) {
54             ret += 1;
55         }
56     }
57     else {
58         for (i=start; i<=DIM; i++) {
59             ret += number_parsPOLYNOM(ord-1,i);
60         }
61     }
62
63     return ret;
64 }
65
66 void make_codingPOLYNOM(int ord,int d,int fac,int cur)
67 {
68     int j=0;
69
70     if ( d == -1) {
71         coding[hpar++]=cur;
72     }
73     else {
74         for (j=0;j<=ord;j++) {
75             make_codingPOLYNOM(ord-j,d-1,fac*(N+1),cur+j*fac);
76         }
77     }
78 }
79
80 void make_fitPOLYNOM(void)
81 {
82     int i=0,j=0,k=0;
83     double **mat,*b;
84
85     check_alloc(b=(double*)malloc(sizeof(double)*pars));
86
87     check_alloc(mat=(double**)malloc(sizeof(double)*pars));
88
89     for (i=0; i<pars; i++) {
90         check_alloc(mat[i]=(double*)malloc(sizeof(double)*pars));
91     }
92
93     for (i=0; i<pars; i++) {
94         b[i]=0.0;
95         for (j=0;j<pars;j++) {
96             mat[i][j]=0.0;
97         }
98     }
99     for (i=0; i<pars; i++) {
100         for (j=i; j<pars; j++) {
101             for (k=(DIM-1)*DELAY; k<INSAMPLE-1; k++) {
102                 mat[i][j] += polynom(k,DIM,coding[i],maxencode) *
103                             polynom(k,DIM,coding[j],maxencode);
104             }
105         }
106     }
107     for (i=0; i<pars; i++) {
108         for (j=i; j<pars; j++) {
109             mat[j][i] = (mat[i][j] /= (INSAMPLE-1-(DIM-1)*DELAY));
110         }
111     }
112
113     for (i=0; i<pars; i++) {
114         for (j=(DIM-1)*DELAY; j<INSAMPLE-1; j++) {

```



```

115     b[i] += series[j+1] * polynom(j,DIM,coding[i],maxencode);
116 }
117     b[i] /= (INSAMPLE-1-(DIM-1)*DELAY);
118 }
119
120 solvele(mat,b,pars);
121
122 for (i=0; i<pars; i++) {
123     results[i]=b[i];
124 }
125
126 free(b);
127
128 for (i=0; i<pars; i++){
129     free(mat[i]);
130 }
131
132 free(mat);
133 }
134
135 void decodePOLYNOM(int *out,int dimen,long cur,long fac)
136 {
137     int n=0;
138     n=cur/fac;
139     out[dimen]=n;
140     if (dimen > 0) {
141         decodePOLYNOM(out, (dimen-1), (cur-((long)n*fac)), (fac/(N+1)));
142     }
143 }
144 }
145
146 double make_errorPOLYNOM(unsigned long i0,unsigned long i1)
147 {
148     int j=0,k=0;
149     double h=0.0,err=0.0;
150
151     err=0.0;
152     for (j=i0+(DIM-1)*DELAY;j<(long)i1-1;j++) {
153         h=0.0;
154         for (k=0;k<pars;k++) {
155             h += results[k]*polynom(j,DIM,coding[k],maxencode);
156         }
157         err += (series[j+1]-h)*(series[j+1]-h);
158     }
159
160     return err /= ((long)i1-(long)i0-(DIM-1)*DELAY);
161 }
162 }
163
164 void make_castPOLYNOM()
165 {
166     int i=0,j=0,k=0,hi=0;
167     double casted=0.0;
168
169     for (i=0;i<=(DIM-1)*DELAY;i++) {
170         series[i]=series[LENGTH-(DIM-1)*DELAY-1+i];
171     }
172
173     hi=(DIM-1)*DELAY;
174
175     for (i=1;i<=CLENGTH;i++) {
176         casted=0.0;
177         for (k=0;k<pars;k++) {
178             casted += results[k] * polynom((DIM - 1) * DELAY,DIM,
179                 coding[k], maxencode);
180         }
181         if (PRINTSCREEN) {
182             cout << casted*std_dev << endl;
183         }
184         PREDICTION.push_back(casted*std_dev);

```

```

185     for (j=0;j<(DIM-1)*DELAY;j++) {
186         series[j]=series[j+1];
187     }
188     series[hi]=casted;
189 }
190 }
191
192 vector<double> runPolynom(vector<double> vetorPredicao)
193 {
194     char stdi=0;
195     int i=0,j=0,k=0;
196     int *opar, sumpar=0;
197     double in_error=0.0,out_error=0.0,av=0.0;
198
199     // initializing global variables
200     pars = 1;
201     varianz = 0;
202     interval = 0;
203     minimo = 0;
204     sinsample=0;
205     hpar = 0;
206     maxencode = 0;
207     exclude = 0;
208     std_dev = 0;
209
210     //erase prediction vetor
211     PREDICTION.erase(PREDICTION.begin(),PREDICTION.end());
212
213     series=(double*)malloc(vetorPredicao.size()*sizeof(double));
214
215     for(int i=0; i < vetorPredicao.size(); i++) {
216         series[i] = vetorPredicao[i];
217     }
218
219     LENGTH = vetorPredicao.size();
220
221     variance(series,LENGTH,&av,&std_dev);
222
223     for (i=0;i<LENGTH;i++) {
224         series[i] /= std_dev;
225     }
226
227     if (!sinsample || (INSAMPLE > LENGTH)) {
228         INSAMPLE=LENGTH;
229     }
230
231     maxencode=1;
232
233     for (i=1;i<DIM;i++) {
234         maxencode *= (N+1);
235     }
236
237     for (i=1;i<=N;i++) {
238         pars += number_parsPOLYNOM(i,1);
239     }
240
241     if (PRINTSCREEN) {
242         cout << "#number of free parameters " << pars << endl << endl;
243     }
244
245     check_alloc(coding=(long*)malloc(sizeof(long)*pars));
246
247     hpar=0;
248
249     make_codingPOLYNOM(N,DIM-1,1,0);
250
251     check_alloc(results=(double*)malloc(sizeof(double)*pars));
252
253     make_fitPOLYNOM();
254

```

```

255 check_alloc(opar=(int*)malloc(sizeof(int)*DIM));
256
257 if (PRINTSCREEN) {
258     cout << "#used norm for the fit " << std_dev << endl;
259 }
260
261 for (j=0; j<pars; j++) {
262     decodePOLYNOM(opar,DIM-1,coding[j],maxencode);
263     if (PRINTSCREEN) {
264         cout << "#";
265     }
266     sumpar=0;
267     for (k=0; k<DIM; k++) {
268         sumpar += opar[k];
269         if (PRINTSCREEN) {
270             cout << opar[k] << " ";
271         }
272     }
273     if (PRINTSCREEN) {
274         cout << results[j]/pow(std_dev,(double)(sumpar-1)) << endl;
275     }
276 }
277
278 in_error=make_errorPOLYNOM((unsigned long)0,INSAMPLE);
279
280 if (PRINTSCREEN) {
281     cout << endl;
282     cout << "#average insample error " << sqrt(in_error) << endl;
283 }
284
285 if (INSAMPLE < LENGTH) {
286     out_error=make_errorPOLYNOM(INSAMPLE,LENGTH);
287     cout << "#average out of sample error " << sqrt(out_error) << endl;
288 }
289
290 if (CAST) {
291     make_castPOLYNOM();
292 }
293
294 free(opar);
295 free(series);
296 free(coefs);
297 free(center);
298 free(coding);
299 free(results);
300
301 return(PREDICTION);
302 }

```

Algorithm D.1 – Algoritmo de Predição POLYNOM

APÊNDICE E

RESULTADOS DE SIMULAÇÃO FORNECIDOS PELO SIMDC3D

O SimDC3D disponibiliza uma grande quantidade de informações e estatísticas após a conclusão de uma simulação. A seguir, apresenta-se os dados gerados de uma simulação que utilizou-se de todos os recursos disponíveis do simulador.

Nas linhas 1 e 2, exibem-se as informações referentes ao tempo de simulação do traço e o tempo de processamento do traço. Nas Linhas 6 a 8, apresentam-se as estatísticas de chegadas das máquinas virtuais no centro de dados. Nas linhas 10 a 12, expõem-se as estatísticas dos tempos de execução das máquinas virtuais no centro de dados.

Na linha 15, é apresentada a quantidade máxima de VMs executada, simultaneamente, no centro de dados. Nas linhas 17 a 19, exibem-se as estatísticas da fila de escalonamento. Nas linhas 21 a 293, expõem-se as estatísticas dos preditores utilizados no algoritmo de escalonamento. Essas estatísticas são apresentadas quando a simulação utiliza essa funcionalidade no escalonamento de máquinas virtuais.

Nas linhas 298 a 368, exibem-se as estatísticas por categoria de migração de máquinas virtuais. Nas linhas 373 a 426, apresentam-se as estatísticas de SLA da simulação, e nas linhas 429 a 481, expõem-se as informações do cálculo de *Downtime*. Nas linhas 485 a 489, são descritos os dados do total de VMs processados no centro de dados.

Na linha 494, mostra-se a temperatura média e, na linha 496, o maior pico do consumo energético do centro de dados. Nas linhas 498 a 509, é exibido a média do consumo energético de todas as dimensões do centro de dados. Nas linhas 513 a 524, apresenta-se o consumo total de energia do centro de dados e, nas linhas 623 a 663, exibe-se a média e o consumo de todas as dimensões a cada 24 horas. Nas linhas 526 e 527, são mostrados o PUE e tPUE do centro de dados.

A utilização média do centro de dados e a utilização diária são exibidas nas linhas 529 a 531. O consumo diário de energia, o consumo por chassi e a média do consumo por chassi são vistos nas linhas 533 a 548.

A sobrecarga térmica por chassi e por servidor está disponibilizado nas linhas 557 a 614. Nas linhas 668 a 690, exibe-se o tempo que os servidores ficaram na escala de temperatura do SimDC3d (10 a 33°C). Nas linhas 692 a 792, expõe-se a quantidade de tempo que os servidores ficaram na escala de utilização de CPU (0 a 100%).

As estatísticas de conexão e desconexão das portas dos comutadores nas respectivas velocidades do *Ethernet* podem ser vistas nas linhas 796 a 817. Essas estatísticas são vistas se o parâmetro que ativa a adaptação de velocidade dos enlaces for verdadeiro. O tráfego total, a média do tráfego, o maior e menor tráfego de cada enlace estão expostos nas linhas 820 a 1373. Essas estatísticas só são apresentadas quando a simulação de tráfego está habilitada.

E, por último, nas linhas 1376 a 1401, são vistos as estatísticas de latência de execução das cargas de trabalho que são extraídos do traço simulado.

```

1 Total time clock running simulator seg / min / hours / day : 1796419 / 29940 / 499 / 20
2 Total Computing time seg / min / hours : 6275 / 104.583 / 1.74306
3
4 Trace Statistics
5
6 Average time between arrivals of VMs: 75.3677
7 Variance between arrivals of VMs: 41893.4
8 Standard deviation arrivals of VMs: 204.679
9
10 Average execution time of the VMs: 33808.4
11 Variance execution time of the VMs: 2.27016e+10
12 Standard deviation arrivals of VMs: 150671
13
14
15 Maximum number of simultaneous VMs: 11801
16
17 Average time in the scheduling queue: 1.05598
18 Variance in the scheduling queue: 39.9193
19 Standard deviation in the scheduling queue: 6.31817
20
21 Prediction Statistics
22
23 Total of hits in the Prediciton Range: 752048
24 Total of error in the Prediciton Range: 1830
25
26 Root Mean Squared Error (RMSE)
27
28 Chassi 0 1.38746 1.36055 1.50502 1.36727 1.54138 1.67385 1.31013 1.36762 1.62291 0.000000
29 Chassi 1 1.67543 1.69901 1.28017 1.69682 1.27793 1.27914 1.69691 0.959853 1.70759 1.67917
30 Chassi 2 1.03877 0.806066 0.921667 1.09445 0.885309 1.12728 1.16143 1.1601 1.05302 1.05384
31 Chassi 3 0.682079 0.707392 0.615888 0.649429 0.569191 0.623371 0.556807 1.30987 0.536826 0.664207
32 Chassi 4 0.309591 0.421132 0.296885 0.297579 0.280508 0.417239 0.150789 0.434674 0.285801 0.274689
33 Chassi 5 1.28213 0.000000 0.000000 1.0494 1.10033 1.50912 1.40722 1.38158 1.26757 1.34934
34 Chassi 6 1.0751 0.000000 1.42849 1.34262 1.10499 0.000000 1.21806 1.0824 1.02128 1.16735
35 Chassi 7 0.197064 1.54263 1.67304 1.73731 1.66161 1.55383 1.284 1.73388 0.000000 1.0449
36 Chassi 8 0.86517 0.967091 0.2128 0.844659 1.08044 0.933583 1.76518 0.000000 1.03422 1.01616
37 Chassi 9 0.579098 0.381067 0.456928 0.317986 0.737433 0.287153 0.446898 0.340018 0.312475 0.299225
38 Chassi 10 1.16624 1.07569 1.03536 1.256 1.12705 1.39331 1.12029 1.1165 1.10785 1.14581
39 Chassi 11 0.979136 1.29169 0.946672 1.35002 0.000000 1.21304 1.40973 1.36958 0.000000 1.00589
40 Chassi 12 1.04305 1.07022 0.000000 0.125564 0.979092 0.143274 0.000000 0.0283962 0.831728 1.07084
41 Chassi 13 0.800758 0.726765 0.000000 0.650346 0.58549 0.581889 0.635659 0.674542 0.720511 0.842361
42 Chassi 14 1.03022 0.315068 0.327598 0.925834 0.340805 0.86674 0.729 0.340919 0.301534 0.330754
43 Chassi 15 0.673931 1.10809 1.09746 1.24684 1.28287 1.01506 1.30108 1.11561 1.22293 1.16449
44 Chassi 16 0.000000 0.129017 0.000000 1.02074 1.23241 1.34173 0.000000 1.0241 1.09813 1.31044
45 Chassi 17 0.995428 1.11012 0.0974476 1.22659 1.08924 1.25234 1.12373 1.02578 1.13373 1.25603
46 Chassi 18 1.01139 0.986367 0.778767 0.750854 1.03918 0.855491 0.000000 0.868744 1.01699 1.00125
47 Chassi 19 0.523974 1.47005 0.259547 0.285937 1.50672 1.69339 0.251564 0.351303 1.44832 1.5003
48 Chassi 20 1.36337 1.05401 1.33675 1.36572 1.07675 2.25043 1.37321 0.713263 0.992927 0.985968
49 Chassi 21 1.14213 1.10304 1.60398 1.1134 0.000000 1.25102 0.920879 1.13904 1.04054 1.09713
50 Chassi 22 0.741921 0.887623 1.17476 0.15361 0.959366 1.37201 0.819234 1.15318 0.814465 0.000000
51 Chassi 23 0.783274 0.292455 0.838339 0.872857 0.413948 0.496441 0.389381 0.000000 0.420996 0.106552
52 Chassi 24 0.793089 0.328597 0.293312 0.306351 0.472778 0.525198 0.528641 0.555075 0.308889 0.384835
53 Chassi 25 1.08959 1.00359 1.36934 1.09865 0.924392 1.10325 1.14073 1.36628 1.01564 1.81797
54 Chassi 26 0.000000 1.09142 1.26974 0.180659 1.11345 0.943543 1.22329 1.1893 1.55427 1.06837
55 Chassi 27 0.104097 3.192 1.08038 1.13487 2.00836 1.13814 1.15122 1.23236 0.952796 0.911448
56 Chassi 28 0.745329 0.214241 0.609351 0.756853 0.684869 0.000000 0.000000 0.585141 0.593283 0.705719
57 Chassi 29 0.398664 0.425808 0.484318 0.429038 0.404838 0.176196 0.425484 0.408388 0.398615 0.382002
58 Chassi 30 1.29477 1.55163 1.28186 0.96201 1.11633 1.08054 0.89752 1.21405 1.46836 1.11531
59 Chassi 31 1.10376 0.000000 1.39303 0.000000 1.18949 0.111503 1.28857 1.11183 0.000000 0.164559
60 Chassi 32 1.01612 1.11062 1.19618 0.000000 0.983346 1.19002 0.000000 0.976217 1.34633 1.25368
61 Chassi 33 1.37384 1.14141 1.16631 0.919843 0.990221 1.52927 1.28904 0.843493 1.07019 1.25128
62 Chassi 34 0.451251 0.450765 0.456941 0.452952 0.553085 0.444365 0.332386 0.585821 0.431267 0.479167
63 Chassi 35 0.000000 1.39259 1.05889 1.38997 1.46088 1.09525 1.31082 1.42935 1.50978 1.14042

```


226 Chassi 36 0.0549478 0.0524675 0.0550788 0.0549261 0.00205801 0.055032 0.0542021 0.0442745 0.0450317 0.00543317
227 Chassi 37 0.00547735 0.00494083 0.0440595 0.0663982 0.0339321 0.0404716 0.0396608 0.0404662 0.0000000 0.0353851
228 Chassi 38 0.0261402 0.0258453 0.0233544 0.0255896 0.0695095 0.0261331 0.0247792 0.0271816 0.0301861 0.0286504
229 Chassi 39 0.0127188 0.0119423 0.0501212 0.0125566 0.0390608 0.0112196 0.0111914 0.0113329 0.0116799 0.0117787
230 Chassi 40 0.0493002 0.0492793 0.0540753 0.0417474 0.0499502 0.0656024 0.0511512 0.0474052 0.0485886 0.0508345
231 Chassi 41 0.058526 0.0542621 0.054466 0.0730725 0.0910956 0.0958281 0.0896158 0.0916721 0.0518648 0.0650801
232 Chassi 42 0.0545727 0.0602011 0.0547573 0.0546822 0.0487274 0.0771292 0.050559 0.0581394 0.011909 0.0567815
233 Chassi 43 0.0432292 0.041488 0.0376846 0.0000000 0.038987 0.0416562 0.0481656 0.0324937 0.0436748 0.0148827
234 Chassi 44 0.0146285 0.0116732 0.0108803 0.0108826 0.0119443 0.00454467 0.01273 0.0133159 0.0119993 0.0103386
235 Chassi 45 0.0591793 0.0567888 0.0465726 0.0446798 0.0361671 0.0441668 0.047331 0.0424095 0.0468167 0.0554653
236 Chassi 46 0.0544368 0.055514 0.052751 0.0528232 0.0442715 0.0542048 0.0568824 0.0542595 0.0467283 0.0437286
237 Chassi 47 0.0378837 0.0477773 0.0475494 0.0000000 0.00557268 0.0443957 0.0579458 0.0459315 0.0000000 0.00552911
238 Chassi 48 0.0232046 0.023557 0.0000000 0.02026 0.025315 0.0258052 0.0263024 0.0203692 0.0219993 0.0267723
239 Chassi 49 0.0111252 0.0154824 0.0115129 0.0103321 0.0113948 0.0135671 0.0144316 0.0271236 0.0143698 0.0317004
240
241
242 Coefficient of variation of predicted values and real
243
244 Chassi 0 1.50242 1.5345 1.75077 1.81349 1.81926 1.56221 1.62251 1.51115 1.6579 0.00000
245 Chassi 1 1.87163 1.8372 1.49982 1.83404 1.50253 1.5036 1.92941 1.6562 1.8537 1.83458
246 Chassi 2 1.52462 1.82718 1.77377 1.48696 1.9782 1.46172 1.48482 1.48678 1.49597 1.55128
247 Chassi 3 1.62913 1.91362 1.72476 1.50956 1.54058 1.49888 1.5303 2.16816 1.58699 1.43145
248 Chassi 4 1.44107 1.80299 1.42713 1.45927 1.57935 1.83845 1.1698 1.67694 1.39297 1.6448
249 Chassi 5 1.6301 0.00000 0.00000 1.8332 1.79848 1.58591 1.49306 1.53515 1.67063 1.5833
250 Chassi 6 1.7 0.00000 1.45453 1.57947 1.58502 0.00000 1.59071 1.7699 1.79615 1.59783
251 Chassi 7 1.11719 2.06386 1.77532 1.89182 1.79998 1.85278 1.56307 1.69128 0.00000 1.77704
252 Chassi 8 1.76031 1.83236 1.14832 1.76841 1.54117 1.61291 2.72279 0.00000 1.60382 1.65053
253 Chassi 9 1.56076 1.27649 1.79396 1.60802 2.17362 1.46245 1.7514 1.36134 1.41107 1.39523
254 Chassi 10 1.57455 1.63918 1.84019 1.68561 1.59189 1.49235 1.59042 1.60434 1.60897 1.51698
255 Chassi 11 1.76351 1.52021 1.89875 1.44726 0.00000 1.65423 1.97655 2.00436 0.00000 1.77051
256 Chassi 12 1.53761 1.48664 0.00000 0.856506 1.71199 0.91107 0.00000 0.491977 1.73428 1.48547
257 Chassi 13 1.65652 1.55492 0.00000 1.68492 1.8075 1.74722 1.6465 1.51205 1.85617 1.90301
258 Chassi 14 2.59539 1.34293 1.27896 2.4725 1.42939 2.85731 2.45671 1.39032 1.22001 1.44133
259 Chassi 15 1.62841 1.65956 1.72471 1.68396 1.61638 1.81697 1.61264 1.57507 1.69449 1.65166
260 Chassi 16 0.00000 0.505517 0.00000 1.80657 1.60717 1.51968 0.00000 1.84969 1.60076 1.57573
261 Chassi 17 1.73519 1.8098 0.337875 1.53858 1.52125 1.49648 1.78794 1.92444 1.48653 1.49749
262 Chassi 18 1.49995 1.51279 1.78426 1.82938 1.39439 1.54156 0.00000 1.90255 1.423 1.44957
263 Chassi 19 1.63558 3.65792 1.31862 1.28747 3.57599 3.68487 1.34596 1.20771 3.72389 3.61853
264 Chassi 20 1.50378 1.82397 1.54546 1.50072 1.64618 2.04422 1.50003 1.56334 1.76743 1.75981
265 Chassi 21 1.64672 1.5782 1.59526 1.74006 0.00000 1.47189 1.94409 1.64114 1.54414 1.55142
266 Chassi 22 1.86756 1.77647 1.6487 0.892089 1.63365 1.79158 1.73056 1.6688 1.75188 0.00000
267 Chassi 23 2.41752 1.44624 2.35893 2.38864 1.49157 1.36013 1.58337 0.00000 1.64531 0.996723
268 Chassi 24 3.01007 1.52211 1.40054 1.34214 2.16359 2.26504 2.667 2.35866 1.32325 2.42363
269 Chassi 25 1.76222 1.85283 1.4783 1.71445 1.8617 1.6616 1.69877 1.47409 1.77494 1.94531
270 Chassi 26 0.00000 1.66606 1.52547 0.864537 1.59844 1.87116 1.61339 1.63576 1.86051 1.70279
271 Chassi 27 0.493426 2.1593 1.63146 1.48536 2.47161 1.56223 1.56524 1.62069 1.70671 1.76729
272 Chassi 28 1.72474 1.63666 2.20987 1.42632 1.66508 0.00000 0.00000 1.76025 1.81809 1.58759
273 Chassi 29 1.56057 1.54906 1.58829 1.38014 1.52948 1.01131 1.59396 1.36369 1.53021 1.49874
274 Chassi 30 1.73995 1.40645 1.39264 1.90698 1.66722 1.8854 1.85477 1.57373 1.42156 1.83487
275 Chassi 31 1.82418 0.00000 1.47516 0.00000 1.59919 0.477998 1.60392 1.76632 0.00000 0.931611
276 Chassi 32 1.73825 1.62804 1.69361 0.00000 1.89732 1.7025 0.00000 1.50136 1.42354 1.56087
277 Chassi 33 2.04757 1.57772 1.55463 1.75279 1.68825 1.89241 2.17033 1.82126 1.67386 1.45464
278 Chassi 34 1.73066 1.74041 1.79574 1.76911 2.58113 1.64028 1.53962 1.96819 1.6497 1.65312
279 Chassi 35 0.00000 1.62461 1.87357 1.78957 1.85285 1.76296 1.65886 1.47927 1.80872 1.72189
280 Chassi 36 1.48529 1.62327 2.07541 1.48616 0.728297 1.51155 1.52089 1.76758 1.52972 0.929651
281 Chassi 37 0.827642 0.982655 1.32699 2.47567 1.68148 1.61836 1.69676 1.6245 0.00000 1.59668
282 Chassi 38 1.62973 1.61783 1.8386 1.6591 2.72553 1.84048 1.79925 1.67173 1.59048 1.78633
283 Chassi 39 1.50783 1.48344 3.45474 1.60679 3.5739 1.541 1.48581 1.52085 1.46405 1.46813
284 Chassi 40 1.62801 1.66361 1.64391 1.69948 1.62644 1.5088 1.81323 1.64549 1.57995 1.75479
285 Chassi 41 2.09477 1.66535 1.72942 2.04916 2.16652 2.31602 1.99391 2.40637 1.67827 1.36708
286 Chassi 42 1.67472 1.69209 1.67217 1.67571 1.74366 1.58732 1.63044 1.51043 1.5879 1.62701
287 Chassi 43 1.55234 1.50454 1.61324 0.00000 1.71725 1.67079 1.60511 1.88596 1.5207 1.70694
288 Chassi 44 1.4948 1.43811 1.30017 1.33476 1.3499 1.0049 1.55404 1.59023 1.89503 1.36544
289 Chassi 45 1.47785 1.65128 1.82472 1.88578 1.65129 1.60002 1.77731 1.64342 1.85243 1.77961
290 Chassi 46 1.45877 1.65321 1.65504 1.58162 1.81594 1.5946 1.8281 1.46585 1.92419 1.81315
291 Chassi 47 1.83483 1.50185 1.53512 0.00000 1.0211 1.82771 2.01567 1.76344 0.00000 0.873658
292 Chassi 48 1.66071 1.55657 0.00000 1.62582 1.61208 1.54049 1.53431 1.7331 1.56607 1.45695
293 Chassi 49 1.34231 1.70823 1.26871 1.32547 1.39059 1.64992 1.76277 2.10141 1.57334 2.17086
294
295
296 Statistics of Migration Policies
297
298 Total Migrations Overloaded Utilization: 1889
299 Total Migrations Overloaded Utilization (every 86400 secs):
300 672 127 655 347 10 25 10 13 10 0 4 0 0 0 14 2 0 0 0
301
302 Total Migrations Prediction: 44238
303 Total Migrations Prediction (every 86400 secs):
304 5522 2073 10350 25149 41 541 61 125 20 43 47 0 4 0 24 136 96 6 0 0
305
306 Total Migrations Low Utilization: 10048
307 Total Migrations Low Utilization (every 86400 secs):
308 2697 1350 2348 189 48 834 69 351 487 66 116 0 7 3 328 989 148 18 0 0
309
310 Total Migrations Hight Temperature: 89
311 Total Migrations Hight Temperature (every 86400 secs)
312 10 5 54 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0
313


```

314 Total idle servers turn off: 2104
315 963 185 326 42 4 163 15 88 24 20 24 0 2 1 106 70 66 5 0 0
316
317
318 Total Migrations per server
319 Chassi 0 276 286 620 354 314 303 98 84 251 57
320 Chassi 1 313 267 76 232 152 255 200 238 156 130
321 Chassi 2 255 251 110 103 133 160 34 31 165 106
322 Chassi 3 121 112 65 29 50 82 77 74 96 111
323 Chassi 4 97 96 75 117 121 139 62 99 117 98
324 Chassi 5 353 96 34 194 168 211 62 65 43 22
325 Chassi 6 442 48 18 24 181 33 134 296 170 306
326 Chassi 7 79 98 69 49 88 118 21 148 11 103
327 Chassi 8 65 74 41 26 101 113 45 27 96 22
328 Chassi 9 142 65 128 165 48 75 173 62 86 70
329 Chassi 10 298 226 197 38 223 20 204 218 199 160
330 Chassi 11 111 19 41 101 30 24 133 138 11 34
331 Chassi 12 17 26 28 39 36 15 19 28 36 15
332 Chassi 13 192 21 37 140 159 31 44 30 82 59
333 Chassi 14 85 104 64 92 60 165 170 120 227 56
334 Chassi 15 53 77 156 52 40 75 37 105 29 195
335 Chassi 16 19 55 40 38 39 50 52 69 77 31
336 Chassi 17 51 76 60 36 208 11 47 32 38 15
337 Chassi 18 141 166 164 238 192 158 44 203 38 162
338 Chassi 19 50 244 77 57 74 77 77 197 78 62
339 Chassi 20 15 43 26 28 41 47 15 35 61 56
340 Chassi 21 33 199 187 40 50 27 38 50 41 76
341 Chassi 22 82 78 252 135 124 89 127 215 120 23
342 Chassi 23 265 84 60 65 212 187 186 36 41 74
343 Chassi 24 64 67 223 245 78 300 85 63 79 128
344 Chassi 25 205 72 23 69 112 259 57 34 77 66
345 Chassi 26 48 63 59 35 37 72 25 67 230 153
346 Chassi 27 51 73 43 48 67 23 177 50 67 83
347 Chassi 28 241 92 113 171 200 42 26 68 217 212
348 Chassi 29 93 131 200 201 245 72 74 243 261 79
349 Chassi 30 56 217 272 137 308 114 157 274 70 261
350 Chassi 31 66 21 48 55 46 39 30 203 49 34
351 Chassi 32 50 31 65 19 154 53 34 63 38 74
352 Chassi 33 137 151 244 61 205 191 122 106 42 209
353 Chassi 34 90 239 269 250 95 225 117 102 74 211
354 Chassi 35 19 67 184 297 289 79 42 25 280 190
355 Chassi 36 20 31 39 53 32 19 27 50 95 27
356 Chassi 37 49 32 181 39 46 184 43 39 46 49
357 Chassi 38 225 267 77 151 61 226 152 54 33 85
358 Chassi 39 64 245 111 82 243 237 96 252 158 244
359 Chassi 40 353 355 254 184 298 190 297 307 329 251
360 Chassi 41 473 344 53 383 390 124 376 146 74 230
361 Chassi 42 47 70 27 37 188 44 140 25 133 39
362 Chassi 43 39 164 177 50 210 20 206 198 11 51
363 Chassi 44 75 196 193 128 57 162 253 203 78 197
364 Chassi 45 29 177 71 187 106 219 162 147 115 100
365 Chassi 46 52 58 43 27 29 41 38 29 48 60
366 Chassi 47 43 10 24 19 46 29 62 35 18 19
367 Chassi 48 53 21 21 52 15 25 17 55 110 21
368 Chassi 49 56 55 48 75 47 131 75 109 65 66
369
370
371 Statistics of Migration Policies
372
373 SLA Violation
374 Chassi 0 0.00536767 0.00734367 0.0163068 0.00991895 0.0110925 0.00235686 0.00533707 0.0104736 0.00400494 0.0180355
375 Chassi 1 0.0147038 0.0113966 0.00220596 0.00907316 0.00901297 0.00846795 0.000881388 0.000757976 0.016818 0.0126403
376 Chassi 2 0.000697553 0.000936828 0.0175783 0.011452 0.000451573 0.000679623 0 0.000779329 0.00958655 0.0145451
377 Chassi 3 0.00134213 0.00116475 0.00112154 0.000747278 0.00279126 0.0111036 0.00733334 4.90227e-05 0.0142161 0.000279782
378 Chassi 4 0.0308126 0.0122295 0.000866607 0.00717322 0.0309642 0.014925 0.0574235 0.000863615 0.000253492 0.00194882
379 Chassi 5 0.00399042 0.000976172 0.000505401 0.00114836 0.00419274 0.000395984 8.29687e-05 5.58673e-05 0.000612001
0.000220171
380 Chassi 6 0.0125975 0.0109288 0 0.000993954 0.00114851 5.73745e-05 0.0011841 0.00612546 0.000415137 0.00548766
381 Chassi 7 0.000805463 0.000892441 0.000778365 0.000254459 0.00121024 0.00182159 0.00012394 0.00473192 0 0.00170205
382 Chassi 8 2.42405e-05 0.0027549 0 3.93726e-05 0.000656989 1.84328e-05 0.00315391 0.00741959 0.000298783 0.000194325
383 Chassi 9 0.000834453 0.000300643 0.000326205 0.00132385 0.000281339 0.000650152 0.00421771 0.000341319 0.000449186
0.000691786
384 Chassi 10 0.00178139 0.000672648 0.000367645 0.000201464 0.000399202 1.87897e-06 0.000139702 0.000599615 0.000623251
0.000743978
385 Chassi 11 0.00049099 0.000225422 0.000569526 0.000296451 0.00245128 0.000181915 0.000798142 0.000676872 5.63062e-05
0.000594735
386 Chassi 12 0.000455709 5.62977e-05 0 0.000774789 0 2.08816e-05 0 0.000357479 0 0.000128853
387 Chassi 13 0.0022885 0.000647633 0 0.00186039 0.00428144 0 0.00137351 6.63975e-06 0.00261983 0.000225216
388 Chassi 14 0.000894642 0.000707535 0.000275323 0.000514574 0.000546284 0.00161933 0.00106859 0.000758393 0.000235895
0.000511512
389 Chassi 15 0.000575364 0.00011379 0.00053781 0.000234038 0.000554529 4.05711e-05 3.15766e-05 0.0003248 0.000247535
0.000243157
390 Chassi 16 2.35427e-06 9.30979e-05 6.24875e-06 8.97922e-07 0 0.000691248 0.00236062 0.000660104 0.000539949 0.00053609
391 Chassi 17 8.94327e-07 0.000441605 0.00019742 0.000747193 0.00131098 0 0.000611976 2.65405e-06 0.000912381 0
392 Chassi 18 6.15596e-05 0.00104641 0.000236966 0.00216564 0.00158504 0.0002963 0.00389013 0.000865609 0.00081899 0.000878677
393 Chassi 19 0.00134899 0.00168613 0.000471874 0.000715914 0.0003998 0.00165184 0.000344874 0.00136279 0.00210123 0.00109668
394 Chassi 20 0 0.000224936 0.000309526 0.00163452 0 0 0.00166964 0.000426297 0.000624299 0.00117051
395 Chassi 21 0 7.79487e-06 0.000367456 0.00113694 0.00179319 0.000581035 0.000295115 0.00343113 0 0.000697024

```



```

477 Chassi 48 99.9891 99.9756 99.9938 99.9731 99.9622 99.992 99.9304 99.9872 99.9567 99.9918
478 Chassi 49 99.9904 99.9775 99.9882 99.9837 99.9925 99.9915 99.9988 99.9969 99.9966 99.9886
479
480
481 Average SLA DownTime per Active Host 99.7819
482
483 Statistics of the Virtual Machines
484
485 Total VMs arrived in the Data Center: 102846
486 Total VMs scheduling in the Data Center: 102846
487 Total VMs finish in the Data Center: 102680
488 Total VMs that have not completed the migration because finish: 155
489 Total VMs Migrates: 56264
490
491
492 Statistics from the average energy consumption of data centers
493
494 Average Supply Temperature: 25.3716
495
496 peakPower: 180386
497
498 Average Power Consumption (Servers = Fans + Computing components): 43808.7
499 Average Power Consumption (Fans): 1499.7
500 Average Power Consumption (Computing components): 42309
501
502 Average Power Consumption (CRAC): 10876
503
504 Average Power Consumption (Switch Access): 755.716
505 Average Power Consumption (Switch Aggregation): 0
506 Average Power Consumption (Switch Core): 749.048
507 Average Power Consumption (Total = Switch Access + Switch Aggregation + Switch Core): 1504.76
508
509 Average Power Consumption (Total = CRAC + Servers + Switches): 56189.4
510
511 Statistical Data Center Energy Consumption
512
513 Energy (Servers = Fans + Computing components): 7.57014e+10
514 Energy (Fans): 2.59148e+09
515 Energy (Computing components): 7.31099e+10
516
517 Energy (CRAC): 1.87937e+10
518
519 Energy (Total = Switch Access + Switch Aggregation + Switch Core): 2.60023e+09
520 Energy (Switch Access): 1.30588e+09
521 Energy (Switch Aggregation): 0
522 Energy (Switch Core): 1.29436e+09
523
524 Energy (Total = CRAC + Servers + Switches): 9.70953e+10
525
526 PUE: 1.24002
527 tPUE: 1.29251
528
529 Average Utilization (Data Center Level): 0.215024
530 Utilization of the data center (every 86400 secs):
531 0.139119 0.266019 0.409045 0.668404 0.685727 0.409881 0.33224 0.312891 0.218923 0.195988 0.170264 0.165239 0.163967 0.158843
    0.153687 0.0139622 0.00255787 0.0022247 0.000954062 0.000655086
532
533 Total computing power (every 86400 secs):
534 132.23 232.184 324.28 469.531 500 399.117 349.058 332.962 239.306 219.932 190.952 187 186.807 184.012 179.314 40.5956 27.5
    26.9968 26 26
535
536 Per Chassis Avg Power:
537 77.3676 114.636 115.701 101.554 96.422 130.909 133.852 104.317 67.2212 103.965
538 147.471 83.9777 88.5825 70.293 87.1908 131.05 86.4803 90.602 57.2813 89.4027
539 81.0419 78.0642 72.2803 67.0886 67.8496 59.072 76.1474 52.0309 61.8362 70.4477
540 124.861 83.2968 58.7923 59.8539 65.1249 124.511 69.1561 59.8919 68.1285 80.7365
541 105.39 130.697 63.3355 49.4338 71.7142 69.7652 76.9887 60.9825 60.4427 79.7238
542
543 Per Chassis Avg Computing Power (max utilization):
544 0.334443 0.511791 0.523683 0.470486 0.451702 0.601367 0.60553 0.497103 0.326292 0.518143
545 0.689948 0.406337 0.45821 0.366733 0.4517 0.629561 0.410045 0.445757 0.294201 0.446543
546 0.411626 0.387939 0.335607 0.32197 0.335132 0.276836 0.396669 0.244371 0.308931 0.373455
547 0.594672 0.399713 0.277312 0.319125 0.354305 0.594867 0.335525 0.283777 0.360636 0.385824
548 0.57474 0.625457 0.309105 0.254568 0.360486 0.372103 0.3925 0.317461 0.311084 0.39868
549
550 Per Chassis Avg Inlet Temperature:
551 25.7388 26.1073 26.5823 28.3135 29.4394 25.58 25.5679 25.7989 26.8073 29.1762
552 25.5858 25.9824 26.8403 27.7622 29.0462 25.587 25.6702 26.1252 26.9671 28.9506
553 25.7797 26.2135 26.9351 28.6325 29.2905 25.7638 26.086 26.5295 27.8403 28.6294
554 25.5751 25.674 26.127 26.4851 28.1847 25.5743 25.928 26.7044 27.3885 28.4848
555 25.5705 25.5577 25.7581 26.654 28.6306 25.7255 26.0392 26.3879 27.7439 28.6784
556
557 Per Chassis Seconds (avg) Over Emergency Temp:
558 0 0 0 0 91.3 0 0 0 0 12
559 0 0 0 0 53.8 0 0 0 0 15.9
560 0 0 0 0 107 0 0 0 0 29.9
561 0 0 0 0 0 0 0 0 0 9.9
562 0 0 0 0 9 0 0 0 0 11.8

```



```

641 Total Power Draw (Computing + CRAC + Switches) (every 86400 secs)
642 34572.7 61517.7 92296.5 144529 152092 103863 87138 82642.1 58378.6 53001.8 46123.4 44969.3 44814.1 43891.5 42699.5 8842
    5823.12 5716.46 5452.72 5425.56
643
644 Total Energy Draw (Computing) (every 86400 secs):
645 2.38235e+09 4.31317e+09 6.23312e+09 9.4764e+09 9.86871e+09 6.9776e+09 5.93559e+09 5.63574e+09 4.02504e+09 3.66827e+09
    3.18966e+09 3.11402e+09 3.10329e+09 3.03949e+09 2.95526e+09 5.44632e+08 3.26031e+08 3.18102e+08 2.98453e+08 2.96425e+08
646
647 Total Energy Draw (CRAC) (every 86400 secs):
648 4.7366e+08 8.68826e+08 1.60679e+09 2.87291e+09 3.13494e+09 1.86322e+09 1.46115e+09 1.37284e+09 8.89084e+08 7.81823e+08
    6.66818e+08 6.42826e+08 6.40146e+08 6.24297e+08 6.05634e+08 9.34103e+07 5.14265e+07 5.01489e+07 4.70511e+07 4.67316e+07
649
650 Total Energy Draw (Switch Access) (every 86400 secs):
651 6.60885e+07 6.78436e+07 6.92922e+07 7.26228e+07 7.18736e+07 6.81992e+07 6.72613e+07 6.69722e+07 6.51333e+07 6.46751e+07
    6.40762e+07 6.40051e+07 6.40018e+07 6.39365e+07 6.3848e+07 6.14232e+07 6.11798e+07 6.1171e+07 6.11366e+07 6.11366e+07
652
653 Total Energy Draw (Switch Aggregation) (every 86400 secs):
654 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
655
656 Total Energy Draw (Switch Core) (every 86400 secs):
657 6.4989e+07 6.52822e+07 6.52122e+07 6.53267e+07 6.52244e+07 6.47595e+07 6.47216e+07 6.47204e+07 6.46497e+07 6.45803e+07
    6.45019e+07 6.45011e+07 6.45011e+07 6.44959e+07 6.44956e+07 6.44832e+07 6.44803e+07 6.44803e+07 6.44751e+07 6.44751e+07
658
659 Total Energy Draw (Total = Switch Access + Switch Aggregation + Switch Core) (every 86400 secs):
660 1.31077e+08 1.33126e+08 1.34504e+08 1.3795e+08 1.37098e+08 1.32959e+08 1.31983e+08 1.31693e+08 1.29783e+08 1.29255e+08
    1.28578e+08 1.28506e+08 1.28503e+08 1.28432e+08 1.28344e+08 1.25906e+08 1.2566e+08 1.25651e+08 1.25612e+08 1.25612e+08
661
662 Total Energy Draw (Computing + CRAC + Switches) (every 86400 secs):
663 2.98708e+09 5.31513e+09 7.97442e+09 1.24873e+10 1.31407e+10 8.97379e+09 7.52872e+09 7.14027e+09 5.04391e+09 4.57935e+09
    3.98506e+09 3.88535e+09 3.87194e+09 3.79222e+09 3.68924e+09 7.63949e+08 5.03117e+08 4.93903e+08 4.71115e+08 4.68769e+08
664
665 Supply Temperature (every 86400 secs):
666 27.2865 25.7402 23.6064 20.5489 19.8343 22.1931 22.9644 23.1033 24.4522 24.895 25.1664 25.3389 25.3491 25.4094 25.4751
    28.6379 29.3517 29.36 29.36 29.36
667
668 Inlet Air Temperature Distribution of the server at the highest temperature (seconds):
669 2690 10
670 3090 11
671 3055 12
672 2266 13
673 1715 14
674 2436 15
675 1903 16
676 2661 17
677 45362 18
678 8569287 19
679 20844730 20
680 17490681 21
681 21718428 22
682 49778539 23
683 30460190 24
684 79862784 25
685 38325379 26
686 21644395 27
687 44254821 28
688 38015456 29
689 3124 30
690 220 31
691
692 Utilization Distribution (seconds) :
693 41591455 0
694 1229 1
695 1070 2
696 1077 3
697 1239 4
698 1234 5
699 1302 6
700 1448 7
701 1426 8
702 1498 9
703 282177 10
704 695 11
705 725 12
706 695 13
707 849 14
708 878 15
709 994 16
710 1148 17
711 1245 18
712 1517 19
713 43551 20
714 2175 21
715 2768 22
716 3331 23
717 4014 24
718 4798 25
719 5693 26
720 6553 27
    
```

721	7908	28
722	9061	29
723	770236	30
724	62937	31
725	72479	32
726	84145	33
727	97756	34
728	113039	35
729	130558	36
730	151381	37
731	172336	38
732	198344	39
733	58878466	40
734	1045053	41
735	1122493	42
736	1213434	43
737	1314336	44
738	1197495	45
739	1212321	46
740	1291209	47
741	1272479	48
742	1352146	49
743	68335125	50
744	1884973	51
745	2025052	52
746	2164831	53
747	2298283	54
748	2277541	55
749	2322773	56
750	2396656	57
751	2330455	58
752	2336264	59
753	42020852	60
754	2757923	61
755	2790632	62
756	2854776	63
757	2914187	64
758	2834485	65
759	2829139	66
760	2847849	67
761	2783156	68
762	2713802	69
763	40151298	70
764	2857180	71
765	2814229	72
766	2763904	73
767	2664897	74
768	2332068	75
769	2126507	76
770	1950828	77
771	1724909	78
772	1530497	79
773	34111530	80
774	962255	81
775	759430	82
776	613044	83
777	496979	84
778	392481	85
779	323226	86
780	272585	87
781	233318	88
782	190935	89
783	957095	90
784	124564	91
785	109758	92
786	99199	93
787	89360	94
788	79515	95
789	72666	96
790	67664	97
791	63458	98
792	55036	99
793		
794	Statistic Switches / Links	
795		
796	SWITCHES EDGE	Connect 10 100 1000 10000 Disconnect 10 100 1000 1000
797	Switch Edge 4	46754 51945 5380 52 46693 51961 5419 52
798	Switch Edge 5	50694 54137 3631 27 50632 54154 3671 27
799	Switch Edge 6	59418 62058 2823 26 59349 62077 2865 26
800	Switch Edge 7	57141 61314 4391 34 57096 61340 4397 34
801	Switch Edge 8	58805 61459 2837 15 58767 61478 2850 15
802	Switch Edge 9	61754 65754 4207 55 61702 65775 4234 55
803	Switch Edge 10	62088 65343 3454 21 62018 65368 3494 21
804	Switch Edge 11	59131 62335 3410 21 59068 62354 3450 21
805	Switch Edge 12	63067 66419 3553 21 62998 66440 3594 21
806	Switch Edge 13	62786 65977 3395 18 62717 65998 3437 18
807	Switch Edge 14	64256 67236 3130 13 64202 67246 3170 13
808	Switch Edge 15	64597 67722 3331 16 64533 67741 3371 16

809	Switch Edge 16	34230	37549	3437	17	34193	37561	3457	17
810									
811	SWITCHES AGGREGATION								
812									
813	SWITCHES CORE								
814	Switch Core 0	1305	17320	16309	294	1292	17320	16309	294
815	Switch Core 1	13	13	13	13	0	13	13	13
816	Switch Core 2	13	13	13	13	0	13	13	13
817	Switch Core 3	13	13	13	13	0	13	13	13
818									
819									
820	LINKS	Sum Traffic	Average Traffic	Higher Traffic	Lower Traffic				
821									
822	[EDGE 0 -> 10]	Traffic Total 4.46358e+07	Average 34.7277	Standard Deviation 46.0902	Higher Traffic 650.183	Lower Traffic 3.84308e-08			
823	[EDGE 0 -> 11]	Traffic Total 5.0803e+07	Average 39.3484	Standard Deviation 37.4521	Higher Traffic 818.235	Lower Traffic 3.86154e-07			
824	[EDGE 0 -> 12]	Traffic Total 3.69897e+07	Average 27.744	Standard Deviation 36.4618	Higher Traffic 1000	Lower Traffic 6.49129e-07			
825	[EDGE 0 -> 13]	Traffic Total 3.95521e+07	Average 30.767	Standard Deviation 36.1278	Higher Traffic 1000	Lower Traffic 1.6909e-07			
826	[EDGE 0 -> 14]	Traffic Total 5.62903e+07	Average 42.9698	Standard Deviation 48.3154	Higher Traffic 745.783	Lower Traffic 2.83889e-07			
827	[EDGE 0 -> 15]	Traffic Total 3.84022e+07	Average 30.178	Standard Deviation 40.4167	Higher Traffic 1000	Lower Traffic 2.25262e-07			
828	[EDGE 0 -> 16]	Traffic Total 2.60219e+07	Average 20.4071	Standard Deviation 32.7784	Higher Traffic 898.247	Lower Traffic 3.86154e-07			
829	[EDGE 0 -> 4]	Traffic Total 2.16822e+08	Average 121.019	Standard Deviation 108.042	Higher Traffic 918.464	Lower Traffic 3.82521e-07			
830	[EDGE 0 -> 5]	Traffic Total 1.6431e+08	Average 123.749	Standard Deviation 71.4983	Higher Traffic 703.701	Lower Traffic 9.04894e-07			
831	[EDGE 0 -> 6]	Traffic Total 1.1517e+08	Average 88.5422	Standard Deviation 56.4989	Higher Traffic 777.478	Lower Traffic 3.80206e-08			
832	[EDGE 0 -> 7]	Traffic Total 4.65201e+07	Average 36.0356	Standard Deviation 37.7832	Higher Traffic 762.9	Lower Traffic 1.6909e-07			
833	[EDGE 0 -> 8]	Traffic Total 5.17696e+07	Average 39.7045	Standard Deviation 40.4864	Higher Traffic 771.3	Lower Traffic 8.76107e-07			
834	[EDGE 0 -> 9]	Traffic Total 5.57394e+07	Average 43.3334	Standard Deviation 49.6696	Higher Traffic 792.139	Lower Traffic 1.79393e-07			
835	[EDGE 1 -> 10]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
836	[EDGE 1 -> 11]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
837	[EDGE 1 -> 12]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
838	[EDGE 1 -> 13]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
839	[EDGE 1 -> 14]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
840	[EDGE 1 -> 15]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
841	[EDGE 1 -> 16]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
842	[EDGE 1 -> 4]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
843	[EDGE 1 -> 5]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
844	[EDGE 1 -> 6]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
845	[EDGE 1 -> 7]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
846	[EDGE 1 -> 8]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
847	[EDGE 1 -> 9]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08			
848	[EDGE 10 -> 257]	Traffic Total 817510	Average 0.693288	Standard Deviation 2.61809	Higher Traffic 410.701	Lower Traffic 1.50348e-05			
849	[EDGE 10 -> 258]	Traffic Total 344214	Average 1.1507	Standard Deviation 2.65588	Higher Traffic 366.27	Lower Traffic 9.23065e-06			
850	[EDGE 10 -> 259]	Traffic Total 539827	Average 2.01693	Standard Deviation 3.6305	Higher Traffic 246.899	Lower Traffic 1.65055e-05			
851	[EDGE 10 -> 260]	Traffic Total 1.37109e+06	Average 3.81817	Standard Deviation 4.03805	Higher Traffic 259.304	Lower Traffic 1.88046e-05			
852	[EDGE 10 -> 261]	Traffic Total 1.09829e+06	Average 2.6832	Standard Deviation 4.05492	Higher Traffic 379.217	Lower Traffic 3.2423e-06			
853	[EDGE 10 -> 262]	Traffic Total 3.58987e+06	Average 9.1539	Standard Deviation 5.92569	Higher Traffic 438.818	Lower Traffic 1.92333e-05			
854	[EDGE 10 -> 263]	Traffic Total 1.06732e+06	Average 3.27947	Standard Deviation 4.48052	Higher Traffic 347.724	Lower Traffic 3.84308e-08			
855	[EDGE 10 -> 264]	Traffic Total 883958	Average 2.70029	Standard Deviation 3.95054	Higher Traffic 244.159	Lower Traffic 5.46522e-05			
856	[EDGE 10 -> 265]	Traffic Total 645792	Average 2.24447	Standard Deviation 3.84036	Higher Traffic 255.037	Lower Traffic 2.60328e-05			
857	[EDGE 10 -> 266]	Traffic Total 990778	Average 3.00305	Standard Deviation 4.40753	Higher Traffic 437.182	Lower Traffic 1.79393e-07			
858	[EDGE 10 -> 267]	Traffic Total 410121	Average 1.42892	Standard Deviation 3.12205	Higher Traffic 434.674	Lower Traffic 2.3052e-05			
859	[EDGE 10 -> 268]	Traffic Total 3.66435e+06	Average 5.49824	Standard Deviation 2.82241	Higher Traffic 169.861	Lower Traffic 2.78558e-06			
860	[EDGE 10 -> 269]	Traffic Total 433729	Average 0.759021	Standard Deviation 2.36178	Higher Traffic 285.443	Lower Traffic 9.54194e-05			
861	[EDGE 10 -> 270]	Traffic Total 1.44778e+06	Average 4.13896	Standard Deviation 5.5738	Higher Traffic 534.684	Lower Traffic 2.388e-05			
862	[EDGE 10 -> 271]	Traffic Total 2.52737e+06	Average 3.92613	Standard Deviation 5.55709	Higher Traffic 268.108	Lower Traffic 2.41022e-06			
863	[EDGE 10 -> 272]	Traffic Total 3.39383e+06	Average 7.58477	Standard Deviation 4.81034	Higher Traffic 543.03	Lower Traffic 5.06422e-05			
864	[EDGE 10 -> 273]	Traffic Total 659654	Average 2.53913	Standard Deviation 3.75562	Higher Traffic 441.465	Lower Traffic 7.26437e-05			
865	[EDGE 10 -> 274]	Traffic Total 278560	Average 1.39765	Standard Deviation 3.17129	Higher Traffic 293.474	Lower Traffic 2.88244e-05			

866	[EDGE 10 -> 275] Traffic Total 3.96432e+06 Average 5.74674 Standard Deviation 9.12383 Higher Traffic 324.04 Lower Traffic 1.8605e-06
867	[EDGE 10 -> 276] Traffic Total 429183 Average 2.02615 Standard Deviation 4.01839 Higher Traffic 157.529 Lower Traffic 7.48217e-06
868	[EDGE 10 -> 277] Traffic Total 751320 Average 0.739962 Standard Deviation 2.46675 Higher Traffic 338.075 Lower Traffic 1.14729e-06
869	[EDGE 10 -> 278] Traffic Total 1.1126e+06 Average 2.22847 Standard Deviation 3.70268 Higher Traffic 227.613 Lower Traffic 1.89728e-05
870	[EDGE 10 -> 279] Traffic Total 556690 Average 0.901257 Standard Deviation 3.21894 Higher Traffic 588.286 Lower Traffic 2.51787e-05
871	[EDGE 10 -> 280] Traffic Total 866869 Average 1.92335 Standard Deviation 2.46964 Higher Traffic 464.299 Lower Traffic 0.000162972
872	[EDGE 10 -> 281] Traffic Total 512585 Average 1.22697 Standard Deviation 3.41473 Higher Traffic 568.149 Lower Traffic 2.38409e-05
873	[EDGE 10 -> 282] Traffic Total 1.57556e+06 Average 2.03276 Standard Deviation 4.88405 Higher Traffic 294.133 Lower Traffic 1.61363e-06
874	[EDGE 10 -> 283] Traffic Total 770144 Average 3.68247 Standard Deviation 4.96935 Higher Traffic 568.149 Lower Traffic 2.45719e-05
875	[EDGE 10 -> 284] Traffic Total 690773 Average 1.60583 Standard Deviation 3.28218 Higher Traffic 650.183 Lower Traffic 5.19564e-05
876	[EDGE 10 -> 285] Traffic Total 2.82592e+06 Average 7.96545 Standard Deviation 14.3015 Higher Traffic 300.389 Lower Traffic 3.76259e-06
877	[EDGE 10 -> 286] Traffic Total 507585 Average 0.464346 Standard Deviation 1.3362 Higher Traffic 135.549 Lower Traffic 1.35099e-05
878	[EDGE 10 -> 287] Traffic Total 426868 Average 2.27196 Standard Deviation 3.85908 Higher Traffic 266.501 Lower Traffic 5.85855e-05
879	[EDGE 10 -> 288] Traffic Total 547196 Average 2.43683 Standard Deviation 3.82366 Higher Traffic 185.639 Lower Traffic 5.36926e-05
880	[EDGE 10 -> 289] Traffic Total 563639 Average 1.16489 Standard Deviation 2.60703 Higher Traffic 129.401 Lower Traffic 2.09657e-05
881	[EDGE 10 -> 290] Traffic Total 774112 Average 1.41756 Standard Deviation 2.84995 Higher Traffic 251.102 Lower Traffic 6.59658e-06
882	[EDGE 10 -> 291] Traffic Total 792241 Average 1.51135 Standard Deviation 3.7934 Higher Traffic 382.344 Lower Traffic 1.2751e-05
883	[EDGE 10 -> 292] Traffic Total 507459 Average 1.38909 Standard Deviation 2.78544 Higher Traffic 372.533 Lower Traffic 5.84623e-05
884	[EDGE 10 -> 293] Traffic Total 1.0577e+06 Average 2.19051 Standard Deviation 2.84582 Higher Traffic 358.153 Lower Traffic 6.47575e-05
885	[EDGE 10 -> 294] Traffic Total 1.03582e+06 Average 3.20952 Standard Deviation 4.45232 Higher Traffic 362.508 Lower Traffic 7.81143e-06
886	[EDGE 10 -> 295] Traffic Total 340755 Average 1.29062 Standard Deviation 2.79646 Higher Traffic 200.245 Lower Traffic 1.98341e-05
887	[EDGE 10 -> 296] Traffic Total 708376 Average 1.77115 Standard Deviation 4.31069 Higher Traffic 265.769 Lower Traffic 7.79705e-06
888	[EDGE 11 -> 297] Traffic Total 2.57466e+06 Average 4.35476 Standard Deviation 7.80916 Higher Traffic 378.883 Lower Traffic 7.428e-06
889	[EDGE 11 -> 298] Traffic Total 537254 Average 2.09277 Standard Deviation 4.07197 Higher Traffic 301.152 Lower Traffic 4.93352e-06
890	[EDGE 11 -> 299] Traffic Total 1.26412e+06 Average 3.0572 Standard Deviation 4.52002 Higher Traffic 399.871 Lower Traffic 2.95762e-06
891	[EDGE 11 -> 300] Traffic Total 494715 Average 1.78858 Standard Deviation 3.57667 Higher Traffic 196.919 Lower Traffic 1.02278e-05
892	[EDGE 11 -> 301] Traffic Total 828353 Average 3.49782 Standard Deviation 4.59214 Higher Traffic 382.627 Lower Traffic 1.20923e-05
893	[EDGE 11 -> 302] Traffic Total 187905 Average 1.30226 Standard Deviation 2.45872 Higher Traffic 172.562 Lower Traffic 2.19816e-05
894	[EDGE 11 -> 303] Traffic Total 625777 Average 0.610332 Standard Deviation 1.26365 Higher Traffic 157.176 Lower Traffic 8.00073e-05
895	[EDGE 11 -> 304] Traffic Total 1.44806e+06 Average 2.69622 Standard Deviation 3.73725 Higher Traffic 538.957 Lower Traffic 5.30813e-06
896	[EDGE 11 -> 305] Traffic Total 1.15263e+06 Average 4.01907 Standard Deviation 5.08007 Higher Traffic 417.195 Lower Traffic 1.34327e-05
897	[EDGE 11 -> 306] Traffic Total 569896 Average 2.30949 Standard Deviation 4.08306 Higher Traffic 214.017 Lower Traffic 5.2005e-06
898	[EDGE 11 -> 307] Traffic Total 1.05799e+06 Average 1.37184 Standard Deviation 3.65626 Higher Traffic 383.298 Lower Traffic 2.23451e-06
899	[EDGE 11 -> 308] Traffic Total 1.02405e+06 Average 2.75213 Standard Deviation 4.57314 Higher Traffic 344.414 Lower Traffic 2.65307e-06
900	[EDGE 11 -> 309] Traffic Total 1.56403e+06 Average 3.52857 Standard Deviation 4.57774 Higher Traffic 591.446 Lower Traffic 5.81126e-06
901	[EDGE 11 -> 310] Traffic Total 552171 Average 1.90485 Standard Deviation 3.58276 Higher Traffic 273.22 Lower Traffic 5.56378e-06
902	[EDGE 11 -> 311] Traffic Total 1.51932e+06 Average 3.57272 Standard Deviation 3.40026 Higher Traffic 403.229 Lower Traffic 2.89011e-06
903	[EDGE 11 -> 312] Traffic Total 314198 Average 1.71446 Standard Deviation 4.26408 Higher Traffic 818.235 Lower Traffic 2.33996e-05
904	[EDGE 11 -> 313] Traffic Total 654089 Average 2.13395 Standard Deviation 3.80314 Higher Traffic 300.646 Lower Traffic 3.86154e-07
905	[EDGE 11 -> 314] Traffic Total 1.50539e+06 Average 2.62484 Standard Deviation 3.82128 Higher Traffic 468.823 Lower Traffic 9.70397e-06
906	[EDGE 11 -> 315] Traffic Total 1.28365e+06 Average 2.06973 Standard Deviation 4.03643 Higher Traffic 512.2 Lower Traffic 1.81907e-05
907	[EDGE 11 -> 316] Traffic Total 483864 Average 1.92807 Standard Deviation 3.79631 Higher Traffic 437.735 Lower Traffic 5.70923e-05
908	[EDGE 11 -> 317] Traffic Total 757453 Average 0.701169 Standard Deviation 1.69321 Higher Traffic 262.931 Lower Traffic 8.4551e-06
909	[EDGE 11 -> 318] Traffic Total 1.94283e+06 Average 2.17655 Standard Deviation 2.05032 Higher Traffic 330.342 Lower Traffic 3.4874e-06

910 [EDGE 11 -> 319] Traffic Total 1.15372e+06 Average 1.00708 Standard Deviation 1.89534 Higher Traffic 357.363 Lower Traffic 1.90927e-05

911 [EDGE 11 -> 320] Traffic Total 1.65359e+06 Average 4.09265 Standard Deviation 5.13591 Higher Traffic 291.849 Lower Traffic 5.14003e-06

912 [EDGE 11 -> 321] Traffic Total 3.3754e+06 Average 2.66417 Standard Deviation 4.17147 Higher Traffic 352.398 Lower Traffic 4.23892e-05

913 [EDGE 11 -> 322] Traffic Total 6.01162e+06 Average 5.37284 Standard Deviation 6.80729 Higher Traffic 220.55 Lower Traffic 1.93333e-05

914 [EDGE 11 -> 323] Traffic Total 2.82466e+06 Average 2.3043 Standard Deviation 5.75167 Higher Traffic 484.205 Lower Traffic 6.63165e-06

915 [EDGE 11 -> 324] Traffic Total 1.9156e+06 Average 1.55516 Standard Deviation 2.29652 Higher Traffic 193.09 Lower Traffic 1.7592e-06

916 [EDGE 11 -> 325] Traffic Total 731464 Average 0.691075 Standard Deviation 2.35158 Higher Traffic 299.664 Lower Traffic 2.12417e-06

917 [EDGE 11 -> 326] Traffic Total 805350 Average 0.874407 Standard Deviation 1.883 Higher Traffic 226.377 Lower Traffic 3.82609e-05

918 [EDGE 11 -> 327] Traffic Total 685027 Average 0.626352 Standard Deviation 1.53399 Higher Traffic 756.84 Lower Traffic 5.54194e-06

919 [EDGE 11 -> 328] Traffic Total 1.5232e+06 Average 2.54716 Standard Deviation 2.69673 Higher Traffic 510.894 Lower Traffic 0.000141144

920 [EDGE 11 -> 329] Traffic Total 1.87109e+06 Average 3.06301 Standard Deviation 3.60414 Higher Traffic 442.532 Lower Traffic 8.8225e-05

921 [EDGE 11 -> 330] Traffic Total 223434 Average 1.49158 Standard Deviation 2.40696 Higher Traffic 218.897 Lower Traffic 5.1203e-05

922 [EDGE 11 -> 331] Traffic Total 1.21063e+06 Average 1.86631 Standard Deviation 1.75268 Higher Traffic 243.809 Lower Traffic 1.61937e-05

923 [EDGE 11 -> 332] Traffic Total 1.38946e+06 Average 1.33637 Standard Deviation 1.13478 Higher Traffic 360.733 Lower Traffic 2.63839e-05

924 [EDGE 11 -> 333] Traffic Total 342522 Average 0.744112 Standard Deviation 2.29024 Higher Traffic 179.948 Lower Traffic 8.3214e-05

925 [EDGE 11 -> 334] Traffic Total 1.86319e+06 Average 1.67226 Standard Deviation 1.23932 Higher Traffic 271.365 Lower Traffic 5.96818e-06

926 [EDGE 11 -> 335] Traffic Total 481094 Average 3.17313 Standard Deviation 5.2863 Higher Traffic 525.879 Lower Traffic 2.45189e-05

927 [EDGE 11 -> 336] Traffic Total 1.01888e+06 Average 0.963826 Standard Deviation 1.53508 Higher Traffic 427.83 Lower Traffic 4.07759e-05

928 [EDGE 12 -> 337] Traffic Total 740504 Average 2.56954 Standard Deviation 3.70353 Higher Traffic 253.071 Lower Traffic 1.21861e-05

929 [EDGE 12 -> 338] Traffic Total 397748 Average 1.777 Standard Deviation 3.5295 Higher Traffic 334.087 Lower Traffic 2.79244e-05

930 [EDGE 12 -> 339] Traffic Total 585943 Average 2.07154 Standard Deviation 4.05465 Higher Traffic 418.788 Lower Traffic 4.20571e-05

931 [EDGE 12 -> 340] Traffic Total 945219 Average 0.921741 Standard Deviation 2.17942 Higher Traffic 1000 Lower Traffic 7.788e-05

932 [EDGE 12 -> 341] Traffic Total 536875 Average 0.480883 Standard Deviation 1.9301 Higher Traffic 351.336 Lower Traffic 1.82568e-05

933 [EDGE 12 -> 342] Traffic Total 485419 Average 2.16123 Standard Deviation 3.9499 Higher Traffic 654.131 Lower Traffic 6.85666e-05

934 [EDGE 12 -> 343] Traffic Total 326609 Average 0.767665 Standard Deviation 2.23855 Higher Traffic 147.418 Lower Traffic 7.89102e-06

935 [EDGE 12 -> 344] Traffic Total 588873 Average 2.42728 Standard Deviation 4.04104 Higher Traffic 482.778 Lower Traffic 1.26797e-05

936 [EDGE 12 -> 345] Traffic Total 1.22111e+06 Average 2.6896 Standard Deviation 2.62506 Higher Traffic 373.413 Lower Traffic 3.42361e-05

937 [EDGE 12 -> 346] Traffic Total 638749 Average 2.82319 Standard Deviation 3.06621 Higher Traffic 301.528 Lower Traffic 2.43997e-05

938 [EDGE 12 -> 347] Traffic Total 1.23041e+06 Average 3.62244 Standard Deviation 5.33749 Higher Traffic 218.836 Lower Traffic 1.14669e-05

939 [EDGE 12 -> 348] Traffic Total 528658 Average 1.06443 Standard Deviation 2.86764 Higher Traffic 257.16 Lower Traffic 7.62049e-07

940 [EDGE 12 -> 349] Traffic Total 560864 Average 2.3139 Standard Deviation 4.26364 Higher Traffic 257.539 Lower Traffic 5.82966e-06

941 [EDGE 12 -> 350] Traffic Total 589440 Average 2.31826 Standard Deviation 3.29819 Higher Traffic 253.402 Lower Traffic 1.7946e-05

942 [EDGE 12 -> 351] Traffic Total 775829 Average 3.129 Standard Deviation 3.97193 Higher Traffic 309.74 Lower Traffic 4.25028e-05

943 [EDGE 12 -> 352] Traffic Total 2.46434e+06 Average 8.07217 Standard Deviation 10.0589 Higher Traffic 270.574 Lower Traffic 1.0827e-05

944 [EDGE 12 -> 353] Traffic Total 2.20361e+06 Average 2.88473 Standard Deviation 6.74912 Higher Traffic 638.084 Lower Traffic 2.51918e-06

945 [EDGE 12 -> 354] Traffic Total 1.38321e+06 Average 3.25438 Standard Deviation 4.43854 Higher Traffic 237.719 Lower Traffic 7.0377e-06

946 [EDGE 12 -> 355] Traffic Total 749288 Average 3.4012 Standard Deviation 3.60227 Higher Traffic 285.634 Lower Traffic 5.70877e-06

947 [EDGE 12 -> 356] Traffic Total 379660 Average 1.655 Standard Deviation 3.11641 Higher Traffic 121.572 Lower Traffic 3.03133e-05

948 [EDGE 12 -> 357] Traffic Total 533450 Average 2.09036 Standard Deviation 3.95116 Higher Traffic 443.843 Lower Traffic 1.02783e-05

949 [EDGE 12 -> 358] Traffic Total 507991 Average 1.85835 Standard Deviation 2.96412 Higher Traffic 241.487 Lower Traffic 1.74996e-05

950 [EDGE 12 -> 359] Traffic Total 718056 Average 2.62219 Standard Deviation 4.11175 Higher Traffic 307.165 Lower Traffic 3.5279e-05

951 [EDGE 12 -> 360] Traffic Total 547359 Average 2.41493 Standard Deviation 4.32945 Higher Traffic 369.412 Lower Traffic 6.49129e-07

952 [EDGE 12 -> 361] Traffic Total 699250 Average 2.6078 Standard Deviation 4.18156 Higher Traffic 337.945 Lower Traffic 2.33132e-06

953 [EDGE 12 -> 362] Traffic Total 1.58086e+06 Average 2.92376 Standard Deviation 3.9379 Higher Traffic 538.957 Lower Traffic 2.63774e-06

954	[EDGE 12 -> 363] Traffic Total 653321 Average 2.70773 Standard Deviation 3.57463 Higher Traffic 314.446 Lower Traffic 1.03465e-05
955	[EDGE 12 -> 364] Traffic Total 1.07114e+06 Average 3.00032 Standard Deviation 5.00093 Higher Traffic 309.652 Lower Traffic 3.02396e-06
956	[EDGE 12 -> 365] Traffic Total 1.61458e+06 Average 2.85077 Standard Deviation 3.88696 Higher Traffic 476.485 Lower Traffic 4.30045e-05
957	[EDGE 12 -> 366] Traffic Total 394576 Average 1.50459 Standard Deviation 3.06329 Higher Traffic 188.171 Lower Traffic 3.17859e-05
958	[EDGE 12 -> 367] Traffic Total 746234 Average 0.728632 Standard Deviation 1.40478 Higher Traffic 179.2 Lower Traffic 3.51415e-06
959	[EDGE 12 -> 368] Traffic Total 773017 Average 1.66211 Standard Deviation 1.87784 Higher Traffic 212.692 Lower Traffic 1.25736e-05
960	[EDGE 12 -> 369] Traffic Total 1.028e+06 Average 0.912205 Standard Deviation 2.09983 Higher Traffic 191.603 Lower Traffic 2.06577e-05
961	[EDGE 12 -> 370] Traffic Total 2.74913e+06 Average 2.26525 Standard Deviation 2.2098 Higher Traffic 286.314 Lower Traffic 3.79009e-05
962	[EDGE 12 -> 371] Traffic Total 3.00648e+06 Average 2.39453 Standard Deviation 3.99279 Higher Traffic 443.263 Lower Traffic 1.78221e-05
963	[EDGE 12 -> 372] Traffic Total 1.1256e+06 Average 0.995533 Standard Deviation 2.17324 Higher Traffic 423.421 Lower Traffic 7.73189e-06
964	[EDGE 12 -> 373] Traffic Total 921417 Average 1.0132 Standard Deviation 2.00808 Higher Traffic 703.701 Lower Traffic 2.86893e-06
965	[EDGE 12 -> 374] Traffic Total 179404 Average 0.331251 Standard Deviation 1.37836 Higher Traffic 187.372 Lower Traffic 8.16499e-05
966	[EDGE 12 -> 375] Traffic Total 975512 Average 0.79823 Standard Deviation 2.10654 Higher Traffic 313.785 Lower Traffic 9.08198e-06
967	[EDGE 12 -> 376] Traffic Total 870713 Average 0.91994 Standard Deviation 2.5323 Higher Traffic 616.027 Lower Traffic 1.84689e-06
968	[EDGE 13 -> 377] Traffic Total 1.28342e+06 Average 2.53786 Standard Deviation 1.87852 Higher Traffic 217.778 Lower Traffic 1.11151e-05
969	[EDGE 13 -> 378] Traffic Total 362897 Average 0.764289 Standard Deviation 2.03572 Higher Traffic 198.346 Lower Traffic 8.25139e-06
970	[EDGE 13 -> 379] Traffic Total 998995 Average 3.14524 Standard Deviation 4.53898 Higher Traffic 293.401 Lower Traffic 1.71592e-05
971	[EDGE 13 -> 380] Traffic Total 365645 Average 1.71065 Standard Deviation 3.05544 Higher Traffic 222.95 Lower Traffic 1.79214e-05
972	[EDGE 13 -> 381] Traffic Total 438140 Average 0.42511 Standard Deviation 1.0476 Higher Traffic 171.732 Lower Traffic 1.6964e-05
973	[EDGE 13 -> 382] Traffic Total 338197 Average 0.730585 Standard Deviation 1.92589 Higher Traffic 261.64 Lower Traffic 7.61657e-05
974	[EDGE 13 -> 383] Traffic Total 392868 Average 2.38494 Standard Deviation 3.58173 Higher Traffic 193.765 Lower Traffic 1.91739e-05
975	[EDGE 13 -> 384] Traffic Total 2.20388e+06 Average 4.3989 Standard Deviation 3.22697 Higher Traffic 389.297 Lower Traffic 1.6909e-07
976	[EDGE 13 -> 385] Traffic Total 1.53374e+06 Average 2.80322 Standard Deviation 6.83546 Higher Traffic 282.3 Lower Traffic 1.23454e-05
977	[EDGE 13 -> 386] Traffic Total 297915 Average 0.281681 Standard Deviation 1.13999 Higher Traffic 375.9 Lower Traffic 3.5586e-05
978	[EDGE 13 -> 387] Traffic Total 3.30454e+06 Average 3.04982 Standard Deviation 1.52573 Higher Traffic 242.052 Lower Traffic 3.77262e-05
979	[EDGE 13 -> 388] Traffic Total 438835 Average 0.888804 Standard Deviation 2.11289 Higher Traffic 295.712 Lower Traffic 2.2069e-05
980	[EDGE 13 -> 389] Traffic Total 697009 Average 2.96469 Standard Deviation 2.93074 Higher Traffic 623.16 Lower Traffic 1.00409e-05
981	[EDGE 13 -> 390] Traffic Total 1.07115e+06 Average 2.70788 Standard Deviation 4.06885 Higher Traffic 1000 Lower Traffic 1.13024e-05
982	[EDGE 13 -> 391] Traffic Total 839689 Average 1.61844 Standard Deviation 2.16761 Higher Traffic 373.574 Lower Traffic 2.46437e-05
983	[EDGE 13 -> 392] Traffic Total 396498 Average 2.12031 Standard Deviation 3.68076 Higher Traffic 374.406 Lower Traffic 1.58164e-05
984	[EDGE 13 -> 393] Traffic Total 488456 Average 2.25592 Standard Deviation 3.81025 Higher Traffic 346.169 Lower Traffic 7.15916e-06
985	[EDGE 13 -> 394] Traffic Total 412434 Average 1.75824 Standard Deviation 3.35984 Higher Traffic 270.06 Lower Traffic 6.92804e-06
986	[EDGE 13 -> 395] Traffic Total 281037 Average 1.87836 Standard Deviation 4.12574 Higher Traffic 178.047 Lower Traffic 0.000143844
987	[EDGE 13 -> 396] Traffic Total 945440 Average 1.87993 Standard Deviation 3.50564 Higher Traffic 226.969 Lower Traffic 1.29033e-05
988	[EDGE 13 -> 397] Traffic Total 630693 Average 0.999135 Standard Deviation 2.37057 Higher Traffic 207.743 Lower Traffic 4.34239e-05
989	[EDGE 13 -> 398] Traffic Total 2.59291e+06 Average 7.42619 Standard Deviation 5.02939 Higher Traffic 254.728 Lower Traffic 1.60417e-05
990	[EDGE 13 -> 399] Traffic Total 742328 Average 3.06903 Standard Deviation 5.25617 Higher Traffic 407.868 Lower Traffic 9.99115e-07
991	[EDGE 13 -> 400] Traffic Total 1.95171e+06 Average 1.74172 Standard Deviation 1.76142 Higher Traffic 370.114 Lower Traffic 1.25184e-05
992	[EDGE 13 -> 401] Traffic Total 786317 Average 3.61615 Standard Deviation 3.68801 Higher Traffic 211.889 Lower Traffic 1.35181e-05
993	[EDGE 13 -> 402] Traffic Total 1.31333e+06 Average 3.65142 Standard Deviation 5.06336 Higher Traffic 422.012 Lower Traffic 1.8932e-05
994	[EDGE 13 -> 403] Traffic Total 1.45255e+06 Average 4.28488 Standard Deviation 8.29094 Higher Traffic 444.206 Lower Traffic 2.83889e-07
995	[EDGE 13 -> 404] Traffic Total 819485 Average 1.67474 Standard Deviation 3.40452 Higher Traffic 660.45 Lower Traffic 1.76138e-06
996	[EDGE 13 -> 405] Traffic Total 468370 Average 2.07133 Standard Deviation 4.10158 Higher Traffic 705.053 Lower Traffic 4.98447e-06
997	[EDGE 13 -> 406] Traffic Total 664419 Average 1.41294 Standard Deviation 3.29372 Higher Traffic 239.264 Lower Traffic 1.19206e-05

998 [EDGE 13 -> 407] Traffic Total 559999 Average 1.994 Standard Deviation 3.5953 Higher Traffic 517.035 Lower Traffic 2.31014e-05

999 [EDGE 13 -> 408] Traffic Total 1.68244e+06 Average 1.42214 Standard Deviation 3.62599 Higher Traffic 305.288 Lower Traffic 1.37391e-05

1000 [EDGE 13 -> 409] Traffic Total 1.22749e+06 Average 1.5443 Standard Deviation 3.11633 Higher Traffic 422.133 Lower Traffic 2.54381e-05

1001 [EDGE 13 -> 410] Traffic Total 378155 Average 1.48557 Standard Deviation 3.15793 Higher Traffic 524.793 Lower Traffic 3.62515e-05

1002 [EDGE 13 -> 411] Traffic Total 1.57517e+06 Average 1.85672 Standard Deviation 3.60843 Higher Traffic 455.341 Lower Traffic 2.22645e-06

1003 [EDGE 13 -> 412] Traffic Total 1.53726e+06 Average 2.86836 Standard Deviation 3.75505 Higher Traffic 289.621 Lower Traffic 1.07603e-05

1004 [EDGE 13 -> 413] Traffic Total 1.40681e+06 Average 2.29461 Standard Deviation 3.18126 Higher Traffic 419.711 Lower Traffic 8.32802e-06

1005 [EDGE 13 -> 414] Traffic Total 1.70717e+06 Average 2.38572 Standard Deviation 3.90637 Higher Traffic 447.329 Lower Traffic 1.12073e-05

1006 [EDGE 13 -> 415] Traffic Total 1.03033e+06 Average 1.77954 Standard Deviation 3.34523 Higher Traffic 347.01 Lower Traffic 4.40558e-06

1007 [EDGE 13 -> 416] Traffic Total 531334 Average 0.952641 Standard Deviation 2.66416 Higher Traffic 218.584 Lower Traffic 1.20094e-06

1008 [EDGE 14 -> 417] Traffic Total 5.84974e+06 Average 4.72992 Standard Deviation 4.76546 Higher Traffic 382.593 Lower Traffic 1.1967e-05

1009 [EDGE 14 -> 418] Traffic Total 2.7165e+06 Average 7.96353 Standard Deviation 7.2123 Higher Traffic 435.82 Lower Traffic 2.23451e-06

1010 [EDGE 14 -> 419] Traffic Total 505806 Average 1.96523 Standard Deviation 3.52779 Higher Traffic 419.082 Lower Traffic 4.19469e-05

1011 [EDGE 14 -> 420] Traffic Total 1.72048e+06 Average 5.21747 Standard Deviation 6.43225 Higher Traffic 355.747 Lower Traffic 1.20547e-06

1012 [EDGE 14 -> 421] Traffic Total 2.37788e+06 Average 1.92283 Standard Deviation 3.18127 Higher Traffic 352.03 Lower Traffic 1.54337e-05

1013 [EDGE 14 -> 422] Traffic Total 804644 Average 3.8352 Standard Deviation 4.58858 Higher Traffic 238.283 Lower Traffic 8.9928e-05

1014 [EDGE 14 -> 423] Traffic Total 952996 Average 1.05554 Standard Deviation 2.41505 Higher Traffic 491.06 Lower Traffic 1.64343e-06

1015 [EDGE 14 -> 424] Traffic Total 1.96139e+06 Average 5.00892 Standard Deviation 4.67397 Higher Traffic 384.505 Lower Traffic 1.56022e-05

1016 [EDGE 14 -> 425] Traffic Total 2.04823e+06 Average 6.59858 Standard Deviation 5.11059 Higher Traffic 245.302 Lower Traffic 2.26661e-05

1017 [EDGE 14 -> 426] Traffic Total 1.47346e+06 Average 1.32934 Standard Deviation 2.19641 Higher Traffic 340.226 Lower Traffic 8.01767e-06

1018 [EDGE 14 -> 427] Traffic Total 4.63034e+06 Average 3.83367 Standard Deviation 5.20416 Higher Traffic 433.817 Lower Traffic 2.83889e-07

1019 [EDGE 14 -> 428] Traffic Total 1.76066e+06 Average 1.57431 Standard Deviation 3.41424 Higher Traffic 524.675 Lower Traffic 2.80951e-05

1020 [EDGE 14 -> 429] Traffic Total 1.33282e+06 Average 1.35484 Standard Deviation 1.77239 Higher Traffic 263.628 Lower Traffic 1.6076e-05

1021 [EDGE 14 -> 430] Traffic Total 2.56331e+06 Average 3.24334 Standard Deviation 3.85695 Higher Traffic 245.243 Lower Traffic 3.19539e-06

1022 [EDGE 14 -> 431] Traffic Total 4.26816e+06 Average 3.38144 Standard Deviation 4.07646 Higher Traffic 343.698 Lower Traffic 5.23157e-06

1023 [EDGE 14 -> 432] Traffic Total 1.99524e+06 Average 1.72999 Standard Deviation 1.99447 Higher Traffic 249.48 Lower Traffic 7.80468e-06

1024 [EDGE 14 -> 433] Traffic Total 2.87076e+06 Average 5.06363 Standard Deviation 3.52934 Higher Traffic 252.451 Lower Traffic 6.30537e-06

1025 [EDGE 14 -> 434] Traffic Total 1.15321e+06 Average 3.55754 Standard Deviation 4.95403 Higher Traffic 521.881 Lower Traffic 8.30022e-06

1026 [EDGE 14 -> 435] Traffic Total 652444 Average 0.57684 Standard Deviation 2.03015 Higher Traffic 343.467 Lower Traffic 8.06799e-06

1027 [EDGE 14 -> 436] Traffic Total 662051 Average 0.624379 Standard Deviation 1.26368 Higher Traffic 301.616 Lower Traffic 4.41625e-06

1028 [EDGE 14 -> 437] Traffic Total 366780 Average 0.903171 Standard Deviation 2.36732 Higher Traffic 157.285 Lower Traffic 1.36337e-06

1029 [EDGE 14 -> 438] Traffic Total 1.07349e+06 Average 1.46212 Standard Deviation 2.66868 Higher Traffic 343.467 Lower Traffic 1.20681e-06

1030 [EDGE 14 -> 439] Traffic Total 488639 Average 1.01872 Standard Deviation 2.52617 Higher Traffic 357.204 Lower Traffic 1.12997e-06

1031 [EDGE 14 -> 440] Traffic Total 520991 Average 0.776249 Standard Deviation 2.14266 Higher Traffic 526.23 Lower Traffic 5.31511e-05

1032 [EDGE 14 -> 441] Traffic Total 1.26586e+06 Average 1.12897 Standard Deviation 2.82478 Higher Traffic 392.741 Lower Traffic 2.70875e-05

1033 [EDGE 14 -> 442] Traffic Total 644297 Average 1.21182 Standard Deviation 2.70179 Higher Traffic 264.354 Lower Traffic 3.03316e-06

1034 [EDGE 14 -> 443] Traffic Total 1.08271e+06 Average 4.09634 Standard Deviation 8.87123 Higher Traffic 745.783 Lower Traffic 1.9126e-05

1035 [EDGE 14 -> 444] Traffic Total 538916 Average 2.60227 Standard Deviation 4.34982 Higher Traffic 444.923 Lower Traffic 2.07422e-05

1036 [EDGE 14 -> 445] Traffic Total 1.81118e+06 Average 3.21843 Standard Deviation 6.70759 Higher Traffic 268.625 Lower Traffic 1.56122e-05

1037 [EDGE 14 -> 446] Traffic Total 656043 Average 2.23529 Standard Deviation 3.99015 Higher Traffic 371.057 Lower Traffic 9.40558e-06

1038 [EDGE 14 -> 447] Traffic Total 395521 Average 2.71522 Standard Deviation 3.86767 Higher Traffic 200.992 Lower Traffic 3.43222e-07

1039 [EDGE 14 -> 448] Traffic Total 566852 Average 2.66573 Standard Deviation 3.23947 Higher Traffic 281.605 Lower Traffic 2.57967e-05

1040 [EDGE 14 -> 449] Traffic Total 623105 Average 2.97722 Standard Deviation 4.44248 Higher Traffic 297.41 Lower Traffic 5.35066e-06

1041 [EDGE 14 -> 450] Traffic Total 160814 Average 1.00218 Standard Deviation 2.24039 Higher Traffic 89.1963 Lower Traffic 5.70877e-06

1042	[EDGE 14 -> 451]	Traffic Total	887350	Average	3.79389	Standard Deviation	3.66275	Higher Traffic	300.239	Lower Traffic	5.5669e-06
1043	[EDGE 14 -> 452]	Traffic Total	327447	Average	1.61212	Standard Deviation	2.88783	Higher Traffic	215.922	Lower Traffic	2.89875e-05
1044	[EDGE 14 -> 453]	Traffic Total	816413	Average	1.36769	Standard Deviation	2.79381	Higher Traffic	374.406	Lower Traffic	5.01655e-06
1045	[EDGE 14 -> 454]	Traffic Total	691374	Average	3.67207	Standard Deviation	5.43269	Higher Traffic	593.478	Lower Traffic	5.68534e-06
1046	[EDGE 14 -> 455]	Traffic Total	885796	Average	1.94179	Standard Deviation	3.75552	Higher Traffic	464.501	Lower Traffic	3.66174e-05
1047	[EDGE 14 -> 456]	Traffic Total	1.21157e+06	Average	4.73881	Standard Deviation	6.06369	Higher Traffic	602.831	Lower	Traffic 6.15763e-06
1048	[EDGE 15 -> 457]	Traffic Total	1.82491e+06	Average	5.00514	Standard Deviation	6.47319	Higher Traffic	478.213	Lower	Traffic 3.97135e-06
1049	[EDGE 15 -> 458]	Traffic Total	600793	Average	2.15502	Standard Deviation	3.40988	Higher Traffic	285.361	Lower Traffic	8.45488e-06
1050	[EDGE 15 -> 459]	Traffic Total	698713	Average	2.04703	Standard Deviation	3.40876	Higher Traffic	261.664	Lower Traffic	1.57385e-05
1051	[EDGE 15 -> 460]	Traffic Total	1.54164e+06	Average	1.97775	Standard Deviation	3.94095	Higher Traffic	349.018	Lower	Traffic 4.88453e-06
1052	[EDGE 15 -> 461]	Traffic Total	374603	Average	1.96887	Standard Deviation	3.83933	Higher Traffic	505.671	Lower Traffic	2.73897e-05
1053	[EDGE 15 -> 462]	Traffic Total	846677	Average	3.72852	Standard Deviation	9.19839	Higher Traffic	550.78	Lower Traffic	8.70949e-06
1054	[EDGE 15 -> 463]	Traffic Total	923110	Average	2.13968	Standard Deviation	3.17563	Higher Traffic	303.478	Lower Traffic	3.33308e-05
1055	[EDGE 15 -> 464]	Traffic Total	1.71721e+06	Average	4.25235	Standard Deviation	6.1742	Higher Traffic	1000	Lower Traffic	1.32817e-05
1056	[EDGE 15 -> 465]	Traffic Total	1.50224e+06	Average	5.21545	Standard Deviation	5.35676	Higher Traffic	315.932	Lower	Traffic 1.07104e-05
1057	[EDGE 15 -> 466]	Traffic Total	2.31626e+06	Average	1.97629	Standard Deviation	2.06823	Higher Traffic	271.645	Lower	Traffic 4.09488e-06
1058	[EDGE 15 -> 467]	Traffic Total	678195	Average	1.91278	Standard Deviation	2.66962	Higher Traffic	181.217	Lower Traffic	7.5023e-07
1059	[EDGE 15 -> 468]	Traffic Total	562029	Average	2.51944	Standard Deviation	4.55837	Higher Traffic	351.488	Lower Traffic	2.14598e-05
1060	[EDGE 15 -> 469]	Traffic Total	648238	Average	2.24706	Standard Deviation	3.6825	Higher Traffic	201.047	Lower Traffic	1.39282e-05
1061	[EDGE 15 -> 470]	Traffic Total	1.55174e+06	Average	5.29132	Standard Deviation	4.83065	Higher Traffic	362.679	Lower	Traffic 2.7088e-05
1062	[EDGE 15 -> 471]	Traffic Total	1.86979e+06	Average	3.80668	Standard Deviation	5.89899	Higher Traffic	251.388	Lower	Traffic 1.18662e-05
1063	[EDGE 15 -> 472]	Traffic Total	2.2713e+06	Average	5.73314	Standard Deviation	4.30064	Higher Traffic	372.991	Lower	Traffic 6.29247e-07
1064	[EDGE 15 -> 473]	Traffic Total	616161	Average	2.30327	Standard Deviation	4.59729	Higher Traffic	346.241	Lower Traffic	2.64006e-05
1065	[EDGE 15 -> 474]	Traffic Total	891498	Average	2.41873	Standard Deviation	4.10029	Higher Traffic	273.292	Lower Traffic	1.47728e-05
1066	[EDGE 15 -> 475]	Traffic Total	1.1691e+06	Average	1.65844	Standard Deviation	3.01019	Higher Traffic	321.707	Lower	Traffic 1.71381e-05
1067	[EDGE 15 -> 476]	Traffic Total	576274	Average	2.39544	Standard Deviation	3.59821	Higher Traffic	170.274	Lower Traffic	1.46647e-05
1068	[EDGE 15 -> 477]	Traffic Total	415251	Average	1.9096	Standard Deviation	2.97497	Higher Traffic	541.445	Lower Traffic	8.42985e-06
1069	[EDGE 15 -> 478]	Traffic Total	709113	Average	1.31666	Standard Deviation	2.56005	Higher Traffic	192.732	Lower Traffic	2.25262e-07
1070	[EDGE 15 -> 479]	Traffic Total	471653	Average	1.02349	Standard Deviation	2.74068	Higher Traffic	223.747	Lower Traffic	4.3252e-05
1071	[EDGE 15 -> 480]	Traffic Total	553195	Average	0.518604	Standard Deviation	2.20349	Higher Traffic	378.883	Lower Traffic	1.79513e-06
1072	[EDGE 15 -> 481]	Traffic Total	2.16306e+06	Average	1.97297	Standard Deviation	2.44491	Higher Traffic	444.841	Lower	Traffic 1.13024e-05
1073	[EDGE 15 -> 482]	Traffic Total	640709	Average	0.990723	Standard Deviation	2.35543	Higher Traffic	243.009	Lower Traffic	1.74856e-05
1074	[EDGE 15 -> 483]	Traffic Total	928593	Average	2.68066	Standard Deviation	5.39893	Higher Traffic	604.513	Lower Traffic	1.75821e-05
1075	[EDGE 15 -> 484]	Traffic Total	510899	Average	2.29492	Standard Deviation	4.15325	Higher Traffic	351.453	Lower Traffic	5.08753e-05
1076	[EDGE 15 -> 485]	Traffic Total	707363	Average	2.0626	Standard Deviation	3.08356	Higher Traffic	227.122	Lower Traffic	1.7617e-05
1077	[EDGE 15 -> 486]	Traffic Total	832367	Average	0.879848	Standard Deviation	2.39968	Higher Traffic	293.337	Lower Traffic	2.33999e-05
1078	[EDGE 15 -> 487]	Traffic Total	734200	Average	1.0636	Standard Deviation	2.62111	Higher Traffic	467.357	Lower Traffic	8.5672e-07
1079	[EDGE 15 -> 488]	Traffic Total	450614	Average	0.987029	Standard Deviation	2.94387	Higher Traffic	306.06	Lower Traffic	1.92839e-05
1080	[EDGE 15 -> 489]	Traffic Total	478655	Average	2.10525	Standard Deviation	3.71584	Higher Traffic	254.063	Lower Traffic	2.2257e-05
1081	[EDGE 15 -> 490]	Traffic Total	416880	Average	2.62486	Standard Deviation	3.15272	Higher Traffic	169.035	Lower Traffic	3.29068e-06
1082	[EDGE 15 -> 491]	Traffic Total	337877	Average	1.98978	Standard Deviation	3.76468	Higher Traffic	289.037	Lower Traffic	3.77469e-05
1083	[EDGE 15 -> 492]	Traffic Total	935954	Average	1.93344	Standard Deviation	3.79417	Higher Traffic	379.528	Lower Traffic	2.07077e-06
1084	[EDGE 15 -> 493]	Traffic Total	1.45423e+06	Average	3.27595	Standard Deviation	3.88211	Higher Traffic	700.137	Lower	Traffic 2.34099e-05
1085	[EDGE 15 -> 494]	Traffic Total	453967	Average	2.05068	Standard Deviation	3.6671	Higher Traffic	497.074	Lower Traffic	7.9655e-06

1086	[EDGE 15 -> 495]	Traffic Total 185728	Average 1.29595	Standard Deviation 2.56815	Higher Traffic 143.495	Lower Traffic 0.000111884
1087	[EDGE 15 -> 496]	Traffic Total 2.62593e+06	Average 5.31066	Standard Deviation 2.72213	Higher Traffic 202.705	Lower Traffic 1.57414e-05
1088	[EDGE 16 -> 497]	Traffic Total 2.50711e+06	Average 4.25173	Standard Deviation 8.61676	Higher Traffic 682.452	Lower Traffic 9.67804e-06
1089	[EDGE 16 -> 498]	Traffic Total 482136	Average 1.90695	Standard Deviation 2.9031	Higher Traffic 208.333	Lower Traffic 8.17256e-06
1090	[EDGE 16 -> 499]	Traffic Total 327634	Average 1.88869	Standard Deviation 3.0973	Higher Traffic 216.926	Lower Traffic 2.1507e-05
1091	[EDGE 16 -> 500]	Traffic Total 919264	Average 2.7203	Standard Deviation 4.52134	Higher Traffic 340.134	Lower Traffic 1.85201e-06
1092	[EDGE 16 -> 501]	Traffic Total 1.62553e+06	Average 3.37046	Standard Deviation 2.72897	Higher Traffic 617.656	Lower Traffic 7.32298e-06
1093	[EDGE 16 -> 502]	Traffic Total 375805	Average 0.355479	Standard Deviation 1.51431	Higher Traffic 251.944	Lower Traffic 3.97348e-05
1094	[EDGE 16 -> 503]	Traffic Total 586149	Average 2.65021	Standard Deviation 4.46763	Higher Traffic 385.889	Lower Traffic 2.47678e-05
1095	[EDGE 16 -> 504]	Traffic Total 705783	Average 2.4583	Standard Deviation 4.17496	Higher Traffic 391.092	Lower Traffic 5.84507e-06
1096	[EDGE 16 -> 505]	Traffic Total 3.59362e+06	Average 7.70189	Standard Deviation 5.13374	Higher Traffic 399.556	Lower Traffic 1.05165e-06
1097	[EDGE 16 -> 506]	Traffic Total 777193	Average 3.39845	Standard Deviation 4.11201	Higher Traffic 363.963	Lower Traffic 1.54541e-05
1098	[EDGE 16 -> 507]	Traffic Total 665926	Average 0.955918	Standard Deviation 2.55544	Higher Traffic 338.625	Lower Traffic 1.24618e-05
1099	[EDGE 16 -> 508]	Traffic Total 1.45132e+06	Average 1.99798	Standard Deviation 2.83801	Higher Traffic 505.863	Lower Traffic 3.86154e-07
1100	[EDGE 16 -> 509]	Traffic Total 825556	Average 2.49019	Standard Deviation 3.48702	Higher Traffic 395.836	Lower Traffic 1.61379e-05
1101	[EDGE 16 -> 510]	Traffic Total 1.12883e+06	Average 3.24081	Standard Deviation 4.21677	Higher Traffic 357.204	Lower Traffic 1.49776e-05
1102	[EDGE 16 -> 511]	Traffic Total 556162	Average 2.06334	Standard Deviation 3.53745	Higher Traffic 389.297	Lower Traffic 1.29196e-06
1103	[EDGE 16 -> 512]	Traffic Total 1.39866e+06	Average 4.76781	Standard Deviation 5.02232	Higher Traffic 535.742	Lower Traffic 1.74793e-05
1104	[EDGE 16 -> 513]	Traffic Total 1.78162e+06	Average 6.27783	Standard Deviation 8.49945	Higher Traffic 301.152	Lower Traffic 1.23454e-05
1105	[EDGE 16 -> 514]	Traffic Total 2.52562e+06	Average 6.96727	Standard Deviation 10.1436	Higher Traffic 898.247	Lower Traffic 4.88453e-06
1106	[EDGE 16 -> 515]	Traffic Total 1.40574e+06	Average 2.30326	Standard Deviation 3.75637	Higher Traffic 302.342	Lower Traffic 1.6711e-05
1107	[EDGE 16 -> 516]	Traffic Total 3.54843e+06	Average 5.98201	Standard Deviation 10.7388	Higher Traffic 433.857	Lower Traffic 3.97135e-06
1108	[EDGE 2 -> 10]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1109	[EDGE 2 -> 11]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1110	[EDGE 2 -> 12]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1111	[EDGE 2 -> 13]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1112	[EDGE 2 -> 14]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1113	[EDGE 2 -> 15]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1114	[EDGE 2 -> 16]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1115	[EDGE 2 -> 4]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1116	[EDGE 2 -> 5]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1117	[EDGE 2 -> 6]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1118	[EDGE 2 -> 7]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1119	[EDGE 2 -> 8]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1120	[EDGE 2 -> 9]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1121	[EDGE 3 -> 10]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1122	[EDGE 3 -> 11]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1123	[EDGE 3 -> 12]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1124	[EDGE 3 -> 13]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1125	[EDGE 3 -> 14]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1126	[EDGE 3 -> 15]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1127	[EDGE 3 -> 16]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1128	[EDGE 3 -> 4]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1129	[EDGE 3 -> 5]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1130	[EDGE 3 -> 6]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1131	[EDGE 3 -> 7]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1132	[EDGE 3 -> 8]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1133	[EDGE 3 -> 9]	Traffic Total 0	Average -nan	Standard Deviation -nan	Higher Traffic 0	Lower Traffic 1e+08
1134	[EDGE 4 -> 17]	Traffic Total 841114	Average 2.05993	Standard Deviation 2.8028	Higher Traffic 203.533	Lower Traffic 3.62288e-05
1135	[EDGE 4 -> 18]	Traffic Total 4.67382e+06	Average 6.70454	Standard Deviation 3.25857	Higher Traffic 339.856	Lower Traffic 2.35611e-05
1136	[EDGE 4 -> 19]	Traffic Total 1.0843e+07	Average 16.2986	Standard Deviation 6.92243	Higher Traffic 370.826	Lower Traffic 6.93703e-06
1137	[EDGE 4 -> 20]	Traffic Total 4.09808e+06	Average 6.57212	Standard Deviation 5.16256	Higher Traffic 308.347	Lower Traffic 9.78247e-06
1138	[EDGE 4 -> 21]	Traffic Total 5.60581e+06	Average 9.05811	Standard Deviation 5.55798	Higher Traffic 748.331	Lower Traffic 5.99454e-07
1139	[EDGE 4 -> 22]	Traffic Total 3.39309e+06	Average 8.68635	Standard Deviation 9.74468	Higher Traffic 365.542	Lower Traffic 1.55459e-05
1140	[EDGE 4 -> 23]	Traffic Total 2.55254e+06	Average 5.20635	Standard Deviation 2.75322	Higher Traffic 544.275	Lower Traffic 3.21515e-05
1141	[EDGE 4 -> 24]	Traffic Total 1.91967e+06	Average 2.98151	Standard Deviation 2.21003	Higher Traffic 275.104	Lower Traffic 9.40558e-06
1142	[EDGE 4 -> 25]	Traffic Total 6.19147e+06	Average 7.68891	Standard Deviation 6.42926	Higher Traffic 336.354	Lower Traffic 6.78105e-06

1187	[EDGE 5 -> 70]	Traffic Total	2.09982e+06	Average	1.8464	Standard Deviation	3.61141	Higher Traffic	563.944	Lower Traffic	1.25534e-05
1188	[EDGE 5 -> 71]	Traffic Total	9.17247e+06	Average	8.05897	Standard Deviation	5.04277	Higher Traffic	394.293	Lower Traffic	2.06436e-05
1189	[EDGE 5 -> 72]	Traffic Total	1.00966e+07	Average	8.24009	Standard Deviation	4.3344	Higher Traffic	798.041	Lower Traffic	1.68745e-06
1190	[EDGE 5 -> 73]	Traffic Total	424669	Average	0.388106	Standard Deviation	1.64463	Higher Traffic	281.779	Lower Traffic	4.73155e-06
1191	[EDGE 5 -> 74]	Traffic Total	2.92254e+06	Average	2.6824	Standard Deviation	1.915	Higher Traffic	406.591	Lower Traffic	8.00725e-06
1192	[EDGE 5 -> 75]	Traffic Total	1.0448e+06	Average	1.27856	Standard Deviation	2.84074	Higher Traffic	278.457	Lower Traffic	1.39728e-05
1193	[EDGE 5 -> 76]	Traffic Total	2.26418e+06	Average	4.79102	Standard Deviation	2.43209	Higher Traffic	275.696	Lower Traffic	6.23343e-05
1194	[EDGE 5 -> 77]	Traffic Total	6.81337e+06	Average	10.6176	Standard Deviation	5.21734	Higher Traffic	372.36	Lower Traffic	1.67937e-05
1195	[EDGE 5 -> 78]	Traffic Total	894030	Average	0.899944	Standard Deviation	1.46879	Higher Traffic	85.3311	Lower Traffic	2.27024e-05
1196	[EDGE 5 -> 79]	Traffic Total	754154	Average	0.851031	Standard Deviation	2.20057	Higher Traffic	213.215	Lower Traffic	0.000151175
1197	[EDGE 5 -> 80]	Traffic Total	2.96496e+06	Average	2.69766	Standard Deviation	1.48504	Higher Traffic	304.245	Lower Traffic	5.41515e-06
1198	[EDGE 5 -> 81]	Traffic Total	2.85612e+06	Average	2.29619	Standard Deviation	4.73105	Higher Traffic	505.863	Lower Traffic	2.29607e-06
1199	[EDGE 5 -> 82]	Traffic Total	3.38811e+06	Average	4.3948	Standard Deviation	2.34256	Higher Traffic	212.079	Lower Traffic	9.15422e-06
1200	[EDGE 5 -> 83]	Traffic Total	3.06619e+06	Average	2.74026	Standard Deviation	1.7867	Higher Traffic	418.788	Lower Traffic	6.74008e-05
1201	[EDGE 5 -> 84]	Traffic Total	5.66207e+06	Average	4.99738	Standard Deviation	3.18932	Higher Traffic	362.515	Lower Traffic	4.72228e-06
1202	[EDGE 5 -> 85]	Traffic Total	3.75652e+06	Average	3.01884	Standard Deviation	5.06414	Higher Traffic	479.753	Lower Traffic	1.02574e-05
1203	[EDGE 5 -> 86]	Traffic Total	3.49513e+06	Average	2.93565	Standard Deviation	3.84168	Higher Traffic	399.556	Lower Traffic	2.26625e-05
1204	[EDGE 5 -> 87]	Traffic Total	2.04618e+06	Average	1.93035	Standard Deviation	2.09725	Higher Traffic	408.432	Lower Traffic	7.75562e-06
1205	[EDGE 5 -> 88]	Traffic Total	4.2622e+06	Average	3.35039	Standard Deviation	4.27205	Higher Traffic	610.785	Lower Traffic	5.56378e-06
1206	[EDGE 5 -> 89]	Traffic Total	1.3291e+07	Average	10.571	Standard Deviation	3.74219	Higher Traffic	582.51	Lower Traffic	5.15633e-06
1207	[EDGE 5 -> 90]	Traffic Total	3.31189e+06	Average	2.74182	Standard Deviation	2.99434	Higher Traffic	340.134	Lower Traffic	3.08829e-05
1208	[EDGE 5 -> 91]	Traffic Total	8.87822e+06	Average	7.07199	Standard Deviation	4.64393	Higher Traffic	425.577	Lower Traffic	5.35066e-06
1209	[EDGE 5 -> 92]	Traffic Total	5.8935e+06	Average	6.67852	Standard Deviation	5.07262	Higher Traffic	561.443	Lower Traffic	7.81512e-06
1210	[EDGE 5 -> 93]	Traffic Total	426513	Average	1.83859	Standard Deviation	3.87185	Higher Traffic	703.701	Lower Traffic	4.42661e-05
1211	[EDGE 5 -> 94]	Traffic Total	3.62805e+06	Average	9.40884	Standard Deviation	6.69406	Higher Traffic	364.894	Lower Traffic	9.45254e-06
1212	[EDGE 5 -> 95]	Traffic Total	1.59497e+06	Average	1.55737	Standard Deviation	1.29532	Higher Traffic	117.656	Lower Traffic	4.82109e-05
1213	[EDGE 5 -> 96]	Traffic Total	729532	Average	2.61243	Standard Deviation	4.17719	Higher Traffic	422.133	Lower Traffic	2.47678e-05
1214	[EDGE 6 -> 100]	Traffic Total	2.93845e+06	Average	10.1968	Standard Deviation	5.74801	Higher Traffic	363.506	Lower Traffic	9.74283e-07
1215	[EDGE 6 -> 101]	Traffic Total	1.17118e+06	Average	4.73444	Standard Deviation	6.03419	Higher Traffic	380.064	Lower Traffic	1.51197e-05
1216	[EDGE 6 -> 102]	Traffic Total	463717	Average	2.58788	Standard Deviation	4.50263	Higher Traffic	309.065	Lower Traffic	2.14535e-05
1217	[EDGE 6 -> 103]	Traffic Total	588347	Average	1.1164	Standard Deviation	2.94777	Higher Traffic	487.398	Lower Traffic	4.16639e-06
1218	[EDGE 6 -> 104]	Traffic Total	1.06378e+07	Average	9.87776	Standard Deviation	4.65019	Higher Traffic	636.745	Lower Traffic	3.80206e-08
1219	[EDGE 6 -> 105]	Traffic Total	1.18742e+06	Average	4.37931	Standard Deviation	3.80348	Higher Traffic	378.605	Lower Traffic	1.18736e-06
1220	[EDGE 6 -> 106]	Traffic Total	652760	Average	2.83951	Standard Deviation	4.64418	Higher Traffic	313.975	Lower Traffic	8.00363e-07
1221	[EDGE 6 -> 107]	Traffic Total	415717	Average	2.77565	Standard Deviation	4.5349	Higher Traffic	269.092	Lower Traffic	2.65307e-06
1222	[EDGE 6 -> 108]	Traffic Total	1.2348e+06	Average	1.15547	Standard Deviation	1.95207	Higher Traffic	255.589	Lower Traffic	1.25425e-06
1223	[EDGE 6 -> 109]	Traffic Total	580647	Average	0.528561	Standard Deviation	2.03011	Higher Traffic	373.413	Lower Traffic	1.61824e-05
1224	[EDGE 6 -> 110]	Traffic Total	1.69952e+06	Average	2.38601	Standard Deviation	3.93864	Higher Traffic	280.07	Lower Traffic	2.11025e-06
1225	[EDGE 6 -> 111]	Traffic Total	1.38903e+06	Average	1.21213	Standard Deviation	1.76038	Higher Traffic	211.634	Lower Traffic	1.74856e-05
1226	[EDGE 6 -> 112]	Traffic Total	1.69479e+06	Average	2.96435	Standard Deviation	4.32404	Higher Traffic	484.881	Lower Traffic	6.63795e-06
1227	[EDGE 6 -> 113]	Traffic Total	2.28e+06	Average	3.32274	Standard Deviation	4.90638	Higher Traffic	660.45	Lower Traffic	6.61415e-06
1228	[EDGE 6 -> 114]	Traffic Total	3.5371e+06	Average	3.02437	Standard Deviation	2.27755	Higher Traffic	362.595	Lower Traffic	8.61768e-06
1229	[EDGE 6 -> 115]	Traffic Total	1.23978e+06	Average	2.07609	Standard Deviation	4.49767	Higher Traffic	395.593	Lower Traffic	3.50312e-06
1230	[EDGE 6 -> 116]	Traffic Total	1.37858e+06	Average	3.51909	Standard Deviation	4.93723	Higher Traffic	467.357	Lower Traffic	7.428e-06

1231	[EDGE 6 -> 117] Traffic Total 1.9393e+06 Average 1.6695 Standard Deviation 2.2051 Higher Traffic 370.826 Lower Traffic 3.4874e-06
1232	[EDGE 6 -> 118] Traffic Total 1.33591e+06 Average 1.11906 Standard Deviation 3.31487 Higher Traffic 316.718 Lower Traffic 6.43992e-07
1233	[EDGE 6 -> 119] Traffic Total 1.03908e+06 Average 2.97938 Standard Deviation 4.38354 Higher Traffic 777.478 Lower Traffic 1.32614e-05
1234	[EDGE 6 -> 120] Traffic Total 2.40137e+06 Average 2.02489 Standard Deviation 2.70866 Higher Traffic 193.768 Lower Traffic 9.50579e-06
1235	[EDGE 6 -> 121] Traffic Total 3.47506e+06 Average 2.69407 Standard Deviation 3.67667 Higher Traffic 422.177 Lower Traffic 4.38609e-06
1236	[EDGE 6 -> 122] Traffic Total 2.20513e+06 Average 1.97572 Standard Deviation 1.68904 Higher Traffic 226.82 Lower Traffic 3.57868e-05
1237	[EDGE 6 -> 123] Traffic Total 4.52506e+06 Average 4.07071 Standard Deviation 1.84217 Higher Traffic 578.086 Lower Traffic 7.23585e-06
1238	[EDGE 6 -> 124] Traffic Total 1.06088e+07 Average 8.40888 Standard Deviation 6.52217 Higher Traffic 395.593 Lower Traffic 9.80151e-06
1239	[EDGE 6 -> 125] Traffic Total 812504 Average 0.766381 Standard Deviation 1.66702 Higher Traffic 358.912 Lower Traffic 1.91739e-05
1240	[EDGE 6 -> 126] Traffic Total 2.37287e+06 Average 1.88638 Standard Deviation 4.09645 Higher Traffic 698.234 Lower Traffic 4.07189e-06
1241	[EDGE 6 -> 127] Traffic Total 1.03487e+07 Average 8.36186 Standard Deviation 7.37912 Higher Traffic 302.342 Lower Traffic 3.84308e-08
1242	[EDGE 6 -> 128] Traffic Total 1.1561e+07 Average 9.31684 Standard Deviation 4.24201 Higher Traffic 460.828 Lower Traffic 7.66818e-06
1243	[EDGE 6 -> 129] Traffic Total 1.08093e+07 Average 8.7588 Standard Deviation 5.80131 Higher Traffic 356.265 Lower Traffic 1.56064e-05
1244	[EDGE 6 -> 130] Traffic Total 980005 Average 1.91731 Standard Deviation 1.86959 Higher Traffic 656.896 Lower Traffic 6.37231e-06
1245	[EDGE 6 -> 131] Traffic Total 640769 Average 0.60564 Standard Deviation 1.74167 Higher Traffic 194.96 Lower Traffic 2.91061e-05
1246	[EDGE 6 -> 132] Traffic Total 2.78293e+06 Average 2.4868 Standard Deviation 3.15462 Higher Traffic 762.769 Lower Traffic 6.09579e-05
1247	[EDGE 6 -> 133] Traffic Total 786331 Average 0.7897 Standard Deviation 1.86402 Higher Traffic 272.581 Lower Traffic 7.57615e-05
1248	[EDGE 6 -> 134] Traffic Total 390221 Average 2.86685 Standard Deviation 2.79485 Higher Traffic 120.809 Lower Traffic 0.000122075
1249	[EDGE 6 -> 135] Traffic Total 1.22506e+06 Average 1.15163 Standard Deviation 0.970672 Higher Traffic 190.785 Lower Traffic 4.18813e-05
1250	[EDGE 6 -> 136] Traffic Total 6.49442e+06 Average 10.2293 Standard Deviation 4.54795 Higher Traffic 449.688 Lower Traffic 1.3369e-06
1251	[EDGE 6 -> 97] Traffic Total 2.69854e+06 Average 4.25337 Standard Deviation 3.82799 Higher Traffic 705.053 Lower Traffic 1.69217e-05
1252	[EDGE 6 -> 98] Traffic Total 520477 Average 1.22185 Standard Deviation 1.6097 Higher Traffic 372.697 Lower Traffic 3.17657e-05
1253	[EDGE 6 -> 99] Traffic Total 7.69137e+06 Average 15.062 Standard Deviation 5.14004 Higher Traffic 500.592 Lower Traffic 2.001e-05
1254	[EDGE 7 -> 137] Traffic Total 1.46299e+06 Average 1.36442 Standard Deviation 2.10874 Higher Traffic 444.841 Lower Traffic 3.76742e-05
1255	[EDGE 7 -> 138] Traffic Total 631856 Average 1.37547 Standard Deviation 2.50615 Higher Traffic 345.833 Lower Traffic 4.51547e-05
1256	[EDGE 7 -> 139] Traffic Total 220948 Average 1.52594 Standard Deviation 2.76031 Higher Traffic 209.692 Lower Traffic 1.49539e-06
1257	[EDGE 7 -> 140] Traffic Total 571820 Average 2.61723 Standard Deviation 4.1907 Higher Traffic 175.886 Lower Traffic 6.37231e-06
1258	[EDGE 7 -> 141] Traffic Total 557794 Average 1.15409 Standard Deviation 2.31757 Higher Traffic 376.134 Lower Traffic 6.12184e-06
1259	[EDGE 7 -> 142] Traffic Total 343386 Average 0.767613 Standard Deviation 2.04986 Higher Traffic 263.829 Lower Traffic 2.32698e-05
1260	[EDGE 7 -> 143] Traffic Total 981521 Average 0.95837 Standard Deviation 0.77901 Higher Traffic 182.653 Lower Traffic 1.64798e-05
1261	[EDGE 7 -> 144] Traffic Total 191956 Average 1.11819 Standard Deviation 2.73077 Higher Traffic 118.188 Lower Traffic 3.28173e-05
1262	[EDGE 7 -> 145] Traffic Total 323974 Average 1.16557 Standard Deviation 2.65797 Higher Traffic 317.964 Lower Traffic 7.81761e-05
1263	[EDGE 7 -> 146] Traffic Total 370157 Average 2.743 Standard Deviation 5.04224 Higher Traffic 286.314 Lower Traffic 1.822e-05
1264	[EDGE 7 -> 147] Traffic Total 1.78931e+06 Average 3.84538 Standard Deviation 5.80239 Higher Traffic 287.745 Lower Traffic 3.54829e-05
1265	[EDGE 7 -> 148] Traffic Total 509528 Average 1.07293 Standard Deviation 3.51658 Higher Traffic 500.592 Lower Traffic 7.5605e-06
1266	[EDGE 7 -> 149] Traffic Total 558674 Average 3.23364 Standard Deviation 5.2818 Higher Traffic 344.427 Lower Traffic 1.85201e-06
1267	[EDGE 7 -> 150] Traffic Total 627673 Average 2.08464 Standard Deviation 3.65097 Higher Traffic 226.377 Lower Traffic 1.6909e-07
1268	[EDGE 7 -> 151] Traffic Total 765110 Average 3.05261 Standard Deviation 5.22386 Higher Traffic 203.357 Lower Traffic 9.97418e-06
1269	[EDGE 7 -> 152] Traffic Total 610041 Average 1.21746 Standard Deviation 2.39812 Higher Traffic 251.102 Lower Traffic 5.8225e-05
1270	[EDGE 7 -> 153] Traffic Total 789661 Average 1.55264 Standard Deviation 2.63851 Higher Traffic 244.248 Lower Traffic 7.1088e-05
1271	[EDGE 7 -> 154] Traffic Total 331151 Average 0.758672 Standard Deviation 2.23394 Higher Traffic 160.712 Lower Traffic 3.26901e-07
1272	[EDGE 7 -> 155] Traffic Total 2.16353e+06 Average 5.38154 Standard Deviation 6.58019 Higher Traffic 596.087 Lower Traffic 3.37507e-06
1273	[EDGE 7 -> 156] Traffic Total 2.37211e+06 Average 2.12376 Standard Deviation 2.06522 Higher Traffic 762.9 Lower Traffic 6.97017e-06
1274	[EDGE 7 -> 157] Traffic Total 1.0927e+06 Average 2.36178 Standard Deviation 4.25187 Higher Traffic 204.232 Lower Traffic 1.57204e-06

1275 [EDGE 7 -> 158] Traffic Total 648117 Average 1.82906 Standard Deviation 3.4289 Higher Traffic 324.305 Lower Traffic 1.57745e-06

1276 [EDGE 7 -> 159] Traffic Total 1.06893e+06 Average 4.10734 Standard Deviation 3.72108 Higher Traffic 262.481 Lower Traffic 1.01138e-05

1277 [EDGE 7 -> 160] Traffic Total 1.11842e+06 Average 0.943279 Standard Deviation 2.7602 Higher Traffic 457.613 Lower Traffic 8.74506e-06

1278 [EDGE 7 -> 161] Traffic Total 1.98453e+06 Average 2.6853 Standard Deviation 3.29495 Higher Traffic 308.508 Lower Traffic 2.46342e-05

1279 [EDGE 7 -> 162] Traffic Total 1.16835e+06 Average 3.56952 Standard Deviation 7.29944 Higher Traffic 544.301 Lower Traffic 7.00941e-06

1280 [EDGE 7 -> 163] Traffic Total 4.35004e+06 Average 7.12124 Standard Deviation 4.8846 Higher Traffic 357.092 Lower Traffic 3.60152e-05

1281 [EDGE 7 -> 164] Traffic Total 1.32957e+06 Average 4.36368 Standard Deviation 6.44318 Higher Traffic 261.736 Lower Traffic 4.30624e-06

1282 [EDGE 7 -> 165] Traffic Total 698212 Average 0.938707 Standard Deviation 4.1791 Higher Traffic 316.622 Lower Traffic 1.18383e-05

1283 [EDGE 7 -> 166] Traffic Total 528057 Average 1.70396 Standard Deviation 3.2182 Higher Traffic 282.687 Lower Traffic 1.20681e-06

1284 [EDGE 7 -> 167] Traffic Total 624362 Average 0.59247 Standard Deviation 2.2463 Higher Traffic 242.516 Lower Traffic 1.01836e-05

1285 [EDGE 7 -> 168] Traffic Total 2.66134e+06 Average 6.31024 Standard Deviation 13.416 Higher Traffic 424.892 Lower Traffic 1.92641e-06

1286 [EDGE 7 -> 169] Traffic Total 3.12132e+06 Average 2.76001 Standard Deviation 4.02403 Higher Traffic 413.527 Lower Traffic 4.3827e-06

1287 [EDGE 7 -> 170] Traffic Total 621966 Average 0.571227 Standard Deviation 1.83061 Higher Traffic 271.377 Lower Traffic 4.25807e-05

1288 [EDGE 7 -> 171] Traffic Total 1.61065e+06 Average 1.50417 Standard Deviation 1.58417 Higher Traffic 222.922 Lower Traffic 1.75048e-05

1289 [EDGE 7 -> 172] Traffic Total 2.03529e+06 Average 1.80198 Standard Deviation 8.02685 Higher Traffic 266.642 Lower Traffic 5.14454e-07

1290 [EDGE 7 -> 173] Traffic Total 2.35242e+06 Average 2.19153 Standard Deviation 2.89364 Higher Traffic 369.46 Lower Traffic 3.16791e-05

1291 [EDGE 7 -> 174] Traffic Total 611832 Average 0.536954 Standard Deviation 1.76317 Higher Traffic 214.511 Lower Traffic 4.72209e-05

1292 [EDGE 7 -> 175] Traffic Total 822715 Average 0.76774 Standard Deviation 1.34768 Higher Traffic 267.019 Lower Traffic 8.54844e-06

1293 [EDGE 7 -> 176] Traffic Total 2.73047e+06 Average 2.21019 Standard Deviation 2.22321 Higher Traffic 293.428 Lower Traffic 6.79029e-05

1294 [EDGE 8 -> 177] Traffic Total 589536 Average 1.40008 Standard Deviation 2.00539 Higher Traffic 246.659 Lower Traffic 1.04998e-05

1295 [EDGE 8 -> 178] Traffic Total 397179 Average 1.79164 Standard Deviation 3.36294 Higher Traffic 771.3 Lower Traffic 1.20955e-05

1296 [EDGE 8 -> 179] Traffic Total 398274 Average 2.62174 Standard Deviation 3.46015 Higher Traffic 423.421 Lower Traffic 3.90652e-05

1297 [EDGE 8 -> 180] Traffic Total 877493 Average 0.788515 Standard Deviation 1.50507 Higher Traffic 432.183 Lower Traffic 1.8605e-06

1298 [EDGE 8 -> 181] Traffic Total 682466 Average 0.577128 Standard Deviation 2.32677 Higher Traffic 372.36 Lower Traffic 8.30022e-06

1299 [EDGE 8 -> 182] Traffic Total 402400 Average 0.877592 Standard Deviation 2.49105 Higher Traffic 293.428 Lower Traffic 2.56369e-06

1300 [EDGE 8 -> 183] Traffic Total 504383 Average 1.31021 Standard Deviation 2.33377 Higher Traffic 336.539 Lower Traffic 0.000119647

1301 [EDGE 8 -> 184] Traffic Total 2.95706e+06 Average 2.64547 Standard Deviation 2.77456 Higher Traffic 484.205 Lower Traffic 4.6425e-05

1302 [EDGE 8 -> 185] Traffic Total 5.00437e+06 Average 4.74156 Standard Deviation 5.07633 Higher Traffic 703.458 Lower Traffic 2.38884e-05

1303 [EDGE 8 -> 186] Traffic Total 639457 Average 0.601293 Standard Deviation 1.80924 Higher Traffic 421.952 Lower Traffic 2.27243e-05

1304 [EDGE 8 -> 187] Traffic Total 783366 Average 0.711296 Standard Deviation 1.41634 Higher Traffic 177.992 Lower Traffic 2.01031e-05

1305 [EDGE 8 -> 188] Traffic Total 2.04296e+06 Average 1.70551 Standard Deviation 3.10408 Higher Traffic 325.39 Lower Traffic 2.33132e-06

1306 [EDGE 8 -> 189] Traffic Total 1.80915e+06 Average 1.75124 Standard Deviation 1.14877 Higher Traffic 98.1526 Lower Traffic 2.77499e-05

1307 [EDGE 8 -> 190] Traffic Total 235131 Average 1.15133 Standard Deviation 2.39708 Higher Traffic 261.071 Lower Traffic 2.6722e-06

1308 [EDGE 8 -> 191] Traffic Total 695521 Average 0.961985 Standard Deviation 3.12912 Higher Traffic 745.783 Lower Traffic 3.94191e-06

1309 [EDGE 8 -> 192] Traffic Total 348345 Average 1.66709 Standard Deviation 3.32669 Higher Traffic 697.029 Lower Traffic 2.65842e-05

1310 [EDGE 8 -> 193] Traffic Total 1.3141e+06 Average 2.26524 Standard Deviation 2.54794 Higher Traffic 371.408 Lower Traffic 1.76138e-06

1311 [EDGE 8 -> 194] Traffic Total 944818 Average 0.843659 Standard Deviation 1.81109 Higher Traffic 351.029 Lower Traffic 3.2976e-05

1312 [EDGE 8 -> 195] Traffic Total 465556 Average 1.8351 Standard Deviation 3.61378 Higher Traffic 372.19 Lower Traffic 2.88659e-05

1313 [EDGE 8 -> 196] Traffic Total 505758 Average 0.476576 Standard Deviation 1.84518 Higher Traffic 541.445 Lower Traffic 8.76107e-07

1314 [EDGE 8 -> 197] Traffic Total 955559 Average 2.46756 Standard Deviation 3.05994 Higher Traffic 371.454 Lower Traffic 1.68745e-06

1315 [EDGE 8 -> 198] Traffic Total 433161 Average 2.03792 Standard Deviation 3.95149 Higher Traffic 443.843 Lower Traffic 4.70463e-05

1316 [EDGE 8 -> 199] Traffic Total 932313 Average 1.27818 Standard Deviation 3.51137 Higher Traffic 259.478 Lower Traffic 1.01938e-06

1317 [EDGE 8 -> 200] Traffic Total 1.28874e+06 Average 3.90904 Standard Deviation 4.39518 Higher Traffic 253.135 Lower Traffic 8.00725e-06

1318 [EDGE 8 -> 201] Traffic Total 4.42082e+06 Average 7.65075 Standard Deviation 3.64079 Higher Traffic 188.597 Lower Traffic 3.64021e-05

1319	[EDGE 8 -> 202] Traffic Total 749090 Average 2.61905 Standard Deviation 3.96811 Higher Traffic 501.262 Lower Traffic 4.42491e-05
1320	[EDGE 8 -> 203] Traffic Total 569186 Average 3.75435 Standard Deviation 4.07183 Higher Traffic 195.403 Lower Traffic 2.61372e-05
1321	[EDGE 8 -> 204] Traffic Total 1.15531e+06 Average 3.20664 Standard Deviation 4.72046 Higher Traffic 433.817 Lower Traffic 5.14003e-06
1322	[EDGE 8 -> 205] Traffic Total 861721 Average 4.01941 Standard Deviation 4.34527 Higher Traffic 247.19 Lower Traffic 5.69428e-05
1323	[EDGE 8 -> 206] Traffic Total 625694 Average 1.53215 Standard Deviation 2.9456 Higher Traffic 429.915 Lower Traffic 9.51873e-06
1324	[EDGE 8 -> 207] Traffic Total 2.041e+06 Average 6.80278 Standard Deviation 6.34239 Higher Traffic 260.937 Lower Traffic 4.7651e-06
1325	[EDGE 8 -> 208] Traffic Total 6.39491e+06 Average 8.09116 Standard Deviation 4.95671 Higher Traffic 725.758 Lower Traffic 1.05329e-05
1326	[EDGE 8 -> 209] Traffic Total 1.06492e+06 Average 0.911596 Standard Deviation 2.39042 Higher Traffic 469.605 Lower Traffic 5.46831e-06
1327	[EDGE 8 -> 210] Traffic Total 437377 Average 1.13322 Standard Deviation 2.53369 Higher Traffic 250.804 Lower Traffic 1.40067e-05
1328	[EDGE 8 -> 211] Traffic Total 1.31243e+06 Average 1.08318 Standard Deviation 2.96263 Higher Traffic 457.512 Lower Traffic 1.89343e-06
1329	[EDGE 8 -> 212] Traffic Total 1.2811e+06 Average 4.14346 Standard Deviation 5.18311 Higher Traffic 444.288 Lower Traffic 2.24903e-05
1330	[EDGE 8 -> 213] Traffic Total 1.13111e+06 Average 1.02962 Standard Deviation 2.89847 Higher Traffic 362.169 Lower Traffic 4.47772e-06
1331	[EDGE 8 -> 214] Traffic Total 1.75478e+06 Average 3.54175 Standard Deviation 2.35401 Higher Traffic 236.095 Lower Traffic 4.0108e-05
1332	[EDGE 8 -> 215] Traffic Total 2.02052e+06 Average 4.06682 Standard Deviation 5.35069 Higher Traffic 386.461 Lower Traffic 1.08474e-05
1333	[EDGE 8 -> 216] Traffic Total 1.46183e+06 Average 3.14805 Standard Deviation 4.58212 Higher Traffic 351.953 Lower Traffic 1.37049e-05
1334	[EDGE 9 -> 217] Traffic Total 322645 Average 0.711494 Standard Deviation 2.53049 Higher Traffic 369.46 Lower Traffic 1.9669e-05
1335	[EDGE 9 -> 218] Traffic Total 503762 Average 0.454652 Standard Deviation 1.89765 Higher Traffic 646.378 Lower Traffic 2.22866e-05
1336	[EDGE 9 -> 219] Traffic Total 731954 Average 0.689686 Standard Deviation 1.58265 Higher Traffic 179.108 Lower Traffic 1.32067e-05
1337	[EDGE 9 -> 220] Traffic Total 283889 Average 1.36226 Standard Deviation 3.01673 Higher Traffic 186.375 Lower Traffic 1.49254e-05
1338	[EDGE 9 -> 221] Traffic Total 749531 Average 2.75173 Standard Deviation 5.05352 Higher Traffic 614.729 Lower Traffic 1.06801e-05
1339	[EDGE 9 -> 222] Traffic Total 623740 Average 1.23542 Standard Deviation 2.3094 Higher Traffic 232.911 Lower Traffic 2.33824e-05
1340	[EDGE 9 -> 223] Traffic Total 352741 Average 1.64558 Standard Deviation 3.04231 Higher Traffic 249.657 Lower Traffic 7.87949e-06
1341	[EDGE 9 -> 224] Traffic Total 1.54532e+06 Average 2.93619 Standard Deviation 3.5018 Higher Traffic 379.648 Lower Traffic 6.67769e-05
1342	[EDGE 9 -> 225] Traffic Total 2.95362e+06 Average 4.80104 Standard Deviation 8.97855 Higher Traffic 274.87 Lower Traffic 2.22645e-06
1343	[EDGE 9 -> 226] Traffic Total 819626 Average 1.39522 Standard Deviation 3.13608 Higher Traffic 253.901 Lower Traffic 8.99503e-06
1344	[EDGE 9 -> 227] Traffic Total 429163 Average 1.8797 Standard Deviation 4.3112 Higher Traffic 792.139 Lower Traffic 3.75562e-05
1345	[EDGE 9 -> 228] Traffic Total 9.23719e+06 Average 7.20099 Standard Deviation 8.20295 Higher Traffic 744.923 Lower Traffic 7.07961e-06
1346	[EDGE 9 -> 229] Traffic Total 3.47102e+06 Average 4.72655 Standard Deviation 5.90987 Higher Traffic 231.267 Lower Traffic 6.12184e-06
1347	[EDGE 9 -> 230] Traffic Total 2.21523e+06 Average 9.64189 Standard Deviation 11.5818 Higher Traffic 368.509 Lower Traffic 8.31077e-06
1348	[EDGE 9 -> 231] Traffic Total 558823 Average 3.19795 Standard Deviation 4.30747 Higher Traffic 202.137 Lower Traffic 5.84341e-05
1349	[EDGE 9 -> 232] Traffic Total 319414 Average 0.700554 Standard Deviation 2.08104 Higher Traffic 184.121 Lower Traffic 1.44929e-05
1350	[EDGE 9 -> 233] Traffic Total 678066 Average 0.586712 Standard Deviation 1.88607 Higher Traffic 271.772 Lower Traffic 4.07337e-05
1351	[EDGE 9 -> 234] Traffic Total 868306 Average 3.43074 Standard Deviation 3.82555 Higher Traffic 330.499 Lower Traffic 1.92839e-05
1352	[EDGE 9 -> 235] Traffic Total 1.27219e+06 Average 1.96652 Standard Deviation 2.71878 Higher Traffic 467.357 Lower Traffic 2.42242e-05
1353	[EDGE 9 -> 236] Traffic Total 2.94123e+06 Average 2.46529 Standard Deviation 5.8791 Higher Traffic 422.272 Lower Traffic 1.79393e-07
1354	[EDGE 9 -> 237] Traffic Total 2.39908e+06 Average 3.40341 Standard Deviation 10.2583 Higher Traffic 315.889 Lower Traffic 1.01401e-05
1355	[EDGE 9 -> 238] Traffic Total 2.46787e+06 Average 4.20263 Standard Deviation 11.0993 Higher Traffic 216.973 Lower Traffic 8.82852e-06
1356	[EDGE 9 -> 239] Traffic Total 1.15236e+06 Average 2.86963 Standard Deviation 3.28199 Higher Traffic 269.789 Lower Traffic 8.18023e-06
1357	[EDGE 9 -> 240] Traffic Total 610584 Average 0.574028 Standard Deviation 2.0591 Higher Traffic 317.585 Lower Traffic 4.05392e-06
1358	[EDGE 9 -> 241] Traffic Total 611252 Average 2.60216 Standard Deviation 4.33102 Higher Traffic 316.622 Lower Traffic 1.92641e-06
1359	[EDGE 9 -> 242] Traffic Total 2.46871e+06 Average 6.41155 Standard Deviation 12.9276 Higher Traffic 671.972 Lower Traffic 1.30722e-06
1360	[EDGE 9 -> 243] Traffic Total 1.27764e+06 Average 4.45427 Standard Deviation 8.27038 Higher Traffic 638.084 Lower Traffic 1.08137e-05
1361	[EDGE 9 -> 244] Traffic Total 3.05685e+06 Average 4.82725 Standard Deviation 3.48247 Higher Traffic 744.923 Lower Traffic 1.66782e-05
1362	[EDGE 9 -> 245] Traffic Total 1.28264e+06 Average 4.81438 Standard Deviation 8.27066 Higher Traffic 194.96 Lower Traffic 1.08678e-05

```

1363 [EDGE 9 -> 246] Traffic Total 326978 Average 0.562298 Standard Deviation 2.00953 Higher Traffic 200.913 Lower Traffic
6.40638e-05
1364 [EDGE 9 -> 247] Traffic Total 3.03212e+06 Average 4.65228 Standard Deviation 2.98969 Higher Traffic 285.634 Lower
Traffic 1.36474e-05
1365 [EDGE 9 -> 248] Traffic Total 716346 Average 3.02631 Standard Deviation 4.24236 Higher Traffic 238.873 Lower Traffic
4.4537e-05
1366 [EDGE 9 -> 249] Traffic Total 850061 Average 0.737975 Standard Deviation 2.53956 Higher Traffic 525.422 Lower Traffic
2.46084e-05
1367 [EDGE 9 -> 250] Traffic Total 1.39938e+06 Average 1.18422 Standard Deviation 2.72336 Higher Traffic 333.842 Lower
Traffic 6.16906e-06
1368 [EDGE 9 -> 251] Traffic Total 695175 Average 1.56923 Standard Deviation 2.97412 Higher Traffic 424.33 Lower Traffic
1.41699e-05
1369 [EDGE 9 -> 252] Traffic Total 675386 Average 1.05859 Standard Deviation 2.04981 Higher Traffic 344.326 Lower Traffic
7.24212e-06
1370 [EDGE 9 -> 253] Traffic Total 363308 Average 1.7368 Standard Deviation 3.38069 Higher Traffic 271.589 Lower Traffic
1.49546e-05
1371 [EDGE 9 -> 254] Traffic Total 1.42106e+06 Average 3.49297 Standard Deviation 2.11939 Higher Traffic 243.572 Lower
Traffic 2.4048e-05
1372 [EDGE 9 -> 255] Traffic Total 585583 Average 2.33114 Standard Deviation 3.60214 Higher Traffic 282.463 Lower Traffic
5.08753e-05
1373 [EDGE 9 -> 256] Traffic Total 645105 Average 2.75854 Standard Deviation 4.59484 Higher Traffic 351.953 Lower Traffic
3.94191e-06
1374
1375
1376 Average Latency: 55226.6
1377
1378 Average Latency (every 86400 secs):
1379 31588.2 42956.2 46348.2 81262.5
1380
1381 Average Latency in % (every 86400 secs):
1382 1436.46 100.823 101.601 100.696
1383
1384 Number of Jobs (every 86400 secs)
1385 1808 1253 3384 3386 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1386
1387 --- Latency for every job ---
1388 Job Number Latency numCPUs Physical Node Number
1389 1082077 1703 1 14.1
1390 1082078 1966 1 17.3
1391 1082079 2152 1 17.5
1392 1082080 178710 20 14.3 14.2 17.4 13.8 13.4 13.3 13 13.2 18.1 14.4 13.6 13.9 17.6 18 17.8 13.1 13.7 13.5 17.7 17.9
1393 1082081 54366 2 18.5 18.2
1394 1082083 65 8 13.1 13.9 14.1 14.2 14.3 14.4 18 18.1
1395 1082084 65 8 14.1 14.1 17.3 17.5 18.2 18.2 18.5 18.5
1396 1082085 65 8 14.1 14.1 17.3 17.5 18.2 18.2 18.5 18.5
1397 1082086 76064 20 17.5 17.5 17.5 17.5 18.2 18.2 18.2 18.2 18.5 18.5 18.5 18.5 14.1 14.1 14.1 14.1 17.3 17.3 17.3 17.3
1398 1082087 41 8 14.1 14.1 17.3 17.5 18.2 18.2 18.5 18.5
1399 1082088 65 8 14.1 14.1 17.3 17.5 18.2 18.2 18.5 18.5
1400 1082090 35 2 17.5 17.5
1401 1082091 94 1 17.5

```