

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Guilherme Schmitt

**DEMÉTER – UM ESTUDO DE ALGORITMOS DE ESCALONAMENTO DE
TAREFAS APLICADOS EM AMBIENTES DE NUVEM DE TRÊS CAMADAS**

Florianópolis

Guilherme Schmitt

**DEMÉTER – UM ESTUDO DE ALGORITMOS DE ESCALONAMENTO DE
TAREFAS APLICADOS EM AMBIENTES DE NUVEM DE TRÊS CAMADAS**

Trabalho de conclusão de curso
submetido ao curso de Sistemas de
Informação para a obtenção do Grau
de Bacharel em Sistemas de
Informação. Orientador: Prof. Dr.
Mario Antônio Ribeiro Dantas
Co-orientador: Eduardo Camilo Inácio
Universidade Federal de Santa
Catarina

Florianópolis

Guilherme Schmitt

**DEMÉTER – UM ESTUDO DE ALGORITMOS DE ESCALONAMENTO DE
TAREFAS APLICADOS EM AMBIENTES DE NUVEM DE TRÊS CAMADAS**

Este trabalho de conclusão de curso foi julgado aprovado para a obtenção do Título de “Bacharel em Sistemas de Informação”, e aprovado em sua forma final pelo Sistemas de Informação.

Florianópolis, Novembro de 2016

Prof. Dr. Frank Siqueira
Universidade Federal de Santa Catarina
Coordenador do Curso

Prof. Dr. Mario Antonio Ribeiro Dantas
Universidade Federal de Santa Catarina
Orientador

Eduardo Camilo Inácio
Universidade Federal de Santa Catarina
Co-orientador

Banca Examinadora:

Prof. Dr. Patrícia Della Méa Plentz
Universidade Federal de Santa Catarina

Prof. Dr. Márcio Bastos Castro
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Agradeço a todos, que de uma forma ou de outra, me ajudaram, mesmo que não sabendo, na elaboração deste trabalho.

A minha família pelo apoio nos momentos de estresse sofridos por causa deste projeto.

Ao professor Mario Dantas e ao Eduardo Camilo, que me deram a oportunidade e liberdade de trilhar o meu caminho neste trabalho.

Aos meus colegas de aula e trabalho, por me ajudarem em momentos de incerteza passados no desenvolvimento deste trabalho. Especialmente aos meus amigos Bruno Crocomo, André Matias e Jhonata da Rocha, o primeiro por ter me dado o melhor conselho ao começar o trabalho de conclusão de curso, o segundo por ter tirado todas as dúvidas que eu sempre tive durante a graduação e o terceiro por ter sido um grande amigo nesses tempos de estudos, dividindo os trabalhos e tarefas das aulas.

Enfim, grato a todos que se dispuseram a ler ou, por ventura, utilizarão este trabalho para algo bom.

“When does a man die? Does he die when he's been shot? No.

Does he die when he's ravaged with disease? No.

Does he die when he's been poisoned? NO!

A man dies when, and only when, he is FORGOTTEN!”

Dr. Hiriluk – One Piece

RESUMO

O crescimento constante da utilização de recursos de *Cloud Computing* (Computação em nuvem), gera um aumento contínuo de consumo energético nos *data center*. Para suportar tal crescimento, empresas apostam em diversas técnicas para diminuir os gastos com energia, seja utilizando virtualização, melhorando o fluxo de ar, unificando servidores que não são utilizados, ou também utilizando algoritmos de escalonamento de tarefas.

Neste trabalho de graduação foi realizado um estudo de alguns algoritmos de escalonamento de tarefas, com ênfase em consumo de energia, aplicados nas arquiteturas de nuvem mais utilizadas atualmente, as configurações de *data centers* de três camadas. Configurações essas onde temos *switches* de acesso, agregação e *core*, além de diversos servidores. Desta forma, cenários foram propostos e simulados de forma a verificar a eficácia dos algoritmos e comparando-os, para melhor entender seus desempenhos nos cenários propostos.

Após a realização dos estudos experimentais nossos resultados indicam que nem sempre o melhor algoritmo de consumo energético é aquele que deve ser usado. Por exemplo, em alguns cenários, apesar da economia energética, constatou-se um nível de falhas da ordem de cerca de 30% as tarefas. Em outros casos, o simulador, também, apresentou falhas ao utilizarmos outras técnicas como DVFS (*Dynamic Voltage and Frequency Scaling* / Escala de tensão e frequência dinâmicas) e virtualização.

Palavras-chave: TI Verde, *Data center*, *Green Cloud*, Eficiência energética, Algoritmos de escalonamento de tarefas.

ABSTRACT

The steady growth in the use of Cloud Computing resources generates a continuous increase in energy consumption in the data center. To support such growth, companies rely on various techniques to reduce energy costs, either by using virtualization, improving air flow, unifying servers that are not being used or using task scheduling algorithms.

In this work, a study of some task scaling algorithms with emphasis on energy consumption applied to the most usable cloud architectures, the three layer data centers, was carried out. Settings for which there are access, aggregation, and core switches, as well as several servers. In this way, scenarios were proposed and simulated in order to verify the effectiveness of the algorithms and to compare them, in order to better understand their performance in the proposed scenarios.

After the experimental studies, the results indicate that not always the best algorithm of energy consumption is the one that should be used. For example, in some scenarios, despite the energy economy, about 30% of the tasks order has failed. In other cases, the simulator also presented faults when using other techniques such as DVFS (Dynamic Voltage and Frequency Scaling) and virtualization.

Key-words: Green IT, Data center, Green cloud, Energy efficient, Task scheduling algorithms.

LISTA DE ABREVIATURAS

TI	Tecnologia da Informação
kWh	Quilowatt-hora
CPU	<i>Central Processing Unit</i>
NORMA	<i>No Remote Memory Access</i>
MPP	<i>Massively Parallel Processors</i>
COW	<i>Cluster of Workstations</i>
PC	<i>Personal Computer</i>
LAN	<i>Local Area Network</i>
SMP	<i>Symmetric Multi-Processing</i>
HPCC	<i>High Performance Computing Cluster</i>
HAC	<i>High Availability Cluster</i>
LBC	<i>Load Balance Computing</i>
SLA	<i>Service Level Agreement</i>
IaaS	<i>Infrastructure as a Service</i>
PaaS	<i>Plataform as a Service</i>
SaaS	<i>Software as a Service</i>
EC2	<i>Elastic Compute Cloud</i>
VM	<i>Virtual Machine</i>
IDC	<i>Interdomain Domain Controller</i>
PUE	<i>Power Usage Effectiveness</i>

LISTA DE QUADROS

Quadro 1: Configuração das tarefas dos ambientes	58
Quadro 2: Configuração dos <i>links</i> dos ambientes	58
Quadro 3: Configuração do primeiro cenário	59
Quadro 4: Configuração do segundo cenário	65
Quadro 5: Configuração do terceiro cenário	73

LISTA DE FIGURAS

Figura 01: Organização de um sistema distribuído	22
Figura 02: Arquitetura de multicomputadores	23
Figura 03: Convergências de elementos na contribuição da computação em nuvem	27
Figura 04: Arquitetura de uma nuvem segundo Foster	28
Figura 05: Arquitetura de uma nuvem verde proposta por Liu	32
Figura 06: Desempenho com a DeepMind ligada e desligada	37
Figura 07: Round-Robin	42
Figura 08: Data center de 3 camadas	50
Figura 09: Interface WEB do simulador Green Cloud	56
Figura 10: Consumo dos aparelhos do cenário 1 com o Random	59
Figura 11: Consumo total do cenário 1	60
Figura 12: Consumo dos switches do cenário 1	61
Figura 13: Consumo dos servidores do cenário 1	61
Figura 14: Gráficos de carga por computador no cenário 1	63
Figura 15: Gráficos de distribuição de tarefas no primeiro cenário	63
Figura 16: Consumo dos aparelhos do cenário 2 com o Random	65
Figura 17: Consumo total do cenário 2	66
Figura 18: Consumo dos switches do cenário 2	67
Figura 19: Consumo dos servidores do cenário 2	67
Figura 20: Carga do Data center do cenário 2	68
Figura 21: Gráficos de carga por computador no cenário 2	69
Figura 22: Quantidade de tarefas que falharam do cenário 2	70

Figura 23: Tarefas que falharam por servidor no cenário 2	71
Figura 24: Consumo dos aparelhos do cenário 3 com o Random	72
Figura 25: Consumo total do cenário 3	73
Figura 26: Consumo dos switches do cenário 3	74
Figura 27: Consumo dos servidores do cenário 3	75
Figura 28: Relação energia e tarefas realizadas pelos servidores no algoritmo BestDENS	76
Figura 29: Relação energia e tarefas realizadas pelos servidores no algoritmo HEROS	76
Figura 30: Uso dos switches no algoritmo BestDENS	77
Figura 31: Uso dos switches no algoritmo HEROS	78
Figura 32: Tarefas que falharam no cenário 3	78

SUMÁRIO

1. INTRODUÇÃO	15
1.1. OBJETIVOS	17
1.1.1. OBJETIVO GERAL	17
1.1.2. OBJETIVOS ESPECÍFICOS	18
1.1.3. METODOLOGIA	18
1.2. ESTRUTURA DO TRABALHO	19
2. FUNDAMENTAÇÃO TEÓRICA	21
2.1. SISTEMAS PARALELOS E DISTRIBUÍDOS	21
2.2. MULTICOMPUTADORES	22
2.3. CLUSTER	23
2.4. GRID	24
2.5. CLOUD COMPUTING	25
2.5.1. MODELOS DE NÍVEIS	28
2.5.2. MODELOS DE SERVIÇOS	29
2.5.3. TIPOS DE NUVEM	30
2.5.4. GREEN CLOUD COMPUTING	31
2.6. DYNAMIC VOLTAGE AND FREQUENCY SCALING	32
2.7. VIRTUALIZAÇÃO	33
3. TRABALHOS RELACIONADOS	35
4. PROPOSTA DEMÉTER	40
4.1. PROPOSTA DOS ALGORITMOS	40
4.1.1. RANDOM	41
4.1.2. ROUND-ROBIN	41

4.1.3. BestDENS	42
4.1.4. HEROS	46
4.2. PROPOSTA DE AMBIENTES E ARQUITETURA DE TRÊS CAMADAS	49
4.3. PROPOSTA DE SIMULAÇÃO	51
4.3.1. COMPARAÇÃO COM OUTROS SIMULADORES	51
4.3.2. GREEN CLOUD SIMULATOR	52
4.3.1. INTERFACE WEB DO SIMULADOR GREEN CLOUD	55
5. CENÁRIOS E RESULTADOS EXPERIMENTAIS	57
5.1. CENÁRIO 1	59
5.1.1. SIMULAÇÃO DO CENÁRIO 1	60
5.2. CENÁRIO 2	65
5.2.1. SIMULAÇÃO DO CENÁRIO 2	66
5.3. CENÁRIO 3	72
5.3.1. SIMULAÇÃO DO CENÁRIO 3	74
5.4. CONSIDERAÇÕES FINAIS	80
6. CONCLUSÃO	83
6.1. TRABALHOS FUTUROS	84
REFERÊNCIAS	85

1. INTRODUÇÃO

Segundo um estudo da Gartner (2014), uma das principais tendências estratégicas de TI em 2015 era a convergência para as redes nuvem como uma grande prioridade das empresas. Segundo Cearly: "(...) A nuvem é o novo estilo de computação de autoatendimento e elasticamente escalável, (...) a coordenação e o gerenciamento (dos aplicativos) serão baseados na nuvem". Segundo um estudo de caso do ITA (2016), o Brasil é o maior utilizador deste tipo de serviço na América Latina. De acordo com Frost e Sullivan (2014) o mercado de computação em nuvem no Brasil movimentou US\$ 217 milhões em 2012, e a previsão é que movimente US\$ 1.1 bilhões até 2017. É interessante notar que não há um tipo de implementação mais popular nos ambientes de nuvem. Segundo Capgemini (2016), um em cada quatro pesquisados afirmou que utiliza uma nuvem pública, um em cada seis utiliza uma nuvem privada, um em cada seis utiliza uma nuvem híbrida e um em cada quatro respondeu não ter nenhuma preferência.

Segundo a IDC, o crescimento contínuo da utilização de recurso de *Cloud Computing*, se deve pela necessidade das empresas em reduzir custos, aumentar a produtividade e melhorar sua eficiência internamente. Mas o crescimento passa além da redução de custos, ele serve também para mitigar os riscos com falta de energia e a garantia de alta disponibilidade e continuidade na operação do código (PIMENTEL, 2016).

Com a alta demanda de serviço em nuvem e sua constante expansão, é necessário cada vez mais recursos para suprir a demanda de serviços, ou seja, são necessários mais servidores e *data centers* para manter a qualidade no fluxo de

informações e dados, aumentando drasticamente o consumo de energia. Um computador gera cerca de uma tonelada de CO²/ano, além de seus componentes conterem materiais tóxicos. Segundo pesquisa realizada por Cupertino (2009), a TI (Tecnologia da Informação) Verde tornou-se essencial e faz parte do planejamento das empresas. O estudo revela que 45% dos executivos entrevistados adotam iniciativas em redução de impactos ambientais, principalmente no que se refere a redução do consumo energético e custos de resfriamento de equipamentos.

Segundo um artigo publicado por Bawden (2016) no Independent, os *data centers* consomem cerca de 3% do fornecimento mundial de energia elétrica e são responsáveis por cerca de 2% da emissão de gases poluentes do efeito estufa. Para efeito de comparação, os 416,2 *terawatts*/hora de energia consumida pelos *data centers* do mundo no último ano foi maior que o consumo total do Reino Unido no respectivo período, algo em torno de 300 *terawatts*/hora. Os gases poluentes equivalem ao que é emitido por toda a indústria aérea. Ian Bitterlin, professor visitante na universidade de Leeds e especialista em *data centers* no Reino Unido, afirma que o consumo de energia usado pelos *data centers* dobra a cada quatro anos. Ele ainda cita um estudo focado no Japão que prevê que toda a oferta de energia do país seria utilizada para o funcionamento dos seus *data centers*, caso o crescimento continue como hoje.

Segundo um estudo de Meisner, Gold e Wenisch (2009), a utilização média dos servidores em um *data center* é entre 20-30%. O desperdício de energia em equipamentos ociosos é agravado pelas perdas na infra-estrutura de fornecimento

de energia e resfriamento, que podem aumentar o consumo de energia entre 50-100%. Em implementações típicas, os servidores ociosos são responsáveis por cerca de 60% do consumo de energia do data center.

Visto as previsões para o aumento do consumo de energia, a *Energy Star* (2012) propôs doze alternativas para diminuir a energia consumida em *data centers*. Dentre as opções apresentadas, aparecem algumas como: Virtualização de servidor, que reduz o custo de energia entre 10% e 40%; investir em tecnologias que são mais eficientes; melhorar o fluxo de ar geral e unificar os servidores que são pouco usados. A tecnologia que será abordada neste trabalho de graduação não é citada como uma das opções propostas pela *Energy Star*, que é a implementação de algoritmos de escalonamento de tarefas em data centers.

1.1. OBJETIVOS

Nesta seção serão apresentados os objetivos do trabalho realizado, tanto os gerais quanto os específicos.

1.1.1. OBJETIVO GERAL

Entender o consumo energético nos ambientes de *cloud* e desenvolver um estudo de análise de algoritmos orientados ao menor consumo de energia.

É esperado obter como resultado, redução no consumo de energia dos ambientes com os algoritmos usados. Por meio destes resultados, será possível avaliar qual algoritmo funciona melhor para cada cenário.

1.1.2. OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

1. Revisar o referencial teórico sobre os algoritmos de escalonamento de tarefas;
2. Identificar o estado da arte de trabalhos relacionados com as mesmas técnicas.
3. Adoção de algoritmos de escalonamento de tarefas em um ambiente de um simulador.
4. Verificar simuladores existentes e escolher o mais apropriado para o desenvolvimento do estudo.
5. Analisar como os algoritmos aplicados, trabalham junto com outras técnicas de redução no consumo, como Virtualização, DVFS e *Dynamic Shutdown*.
6. Comparar quantitativamente os resultados dos algoritmos, comparados a um estado de partida.

1.1.3. METODOLOGIA

Neste trabalho de graduação é realizado uma pesquisa exploratória a fim de comparar algoritmos de escalonamento de tarefa aplicados em ambientes de nuvem de três camadas e avaliá-los em termos de eficiência energética.

A metodologia de desenvolvimento deste trabalho é dividida em três etapas:

Etapa 1. Nesta etapa serão levantadas informações sobre os conceitos chaves. Será analisada a literatura na área de algoritmos de escalonamento voltados ao consumo de energia, e posteriormente enfatizando a eficiências dos algoritmos em ambientes de *cloud computing*.

Etapa 2. Pesquisa de trabalhos relacionados à algoritmos de escalonamento de tarefa e *energy-aware* aplicados em redes *cloud computing*.

Etapa 3. Será feita uma análise de simuladores, para saber qual que atende melhor os requisitos requeridos para as simulações.

Etapa 4. Aplicação prática e análise dos algoritmos no ambiente de simulação.

Atividade 4.1: Aplicação dos algoritmos (Random, BestDENS, HEROS e Round-Robin) nos ambientes propostos.

Atividade 4.2: Análise dos resultados obtidos das simulações.

1.2. ESTRUTURA DO TRABALHO

Este trabalho está dividido em 6 capítulos. No capítulo dois é apresentado a fundamentação teórica do trabalho, são apresentados fundamentos da computação distribuída e da computação em nuvem e detalhes do simulador utilizado.

O capítulo três é apresentado os trabalhos relacionados a pesquisa feita, apresentando estudos que se focam na mesma área desta pesquisa: A redução de consumo de energia por meio de algoritmos, tanto de escalonamento de tarefa, quando de *energy-aware*. Também são detalhados os algoritmos que serão utilizados.

No capítulo 4 é apresentado a proposta da pesquisa a ser feita neste trabalho e no capítulo 5 demonstrado os cenários e os resultados obtidos das simulações feitas. No capítulo 6 ficam as conclusões da pesquisa junto aos trabalhos propostos a serem realizados no futuro.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados conceitos básicos utilizados no trabalho, para contextualizar as diversas áreas que podem ser utilizadas no decorrer do projeto.

2.1. SISTEMAS PARALELOS E DISTRIBUÍDOS

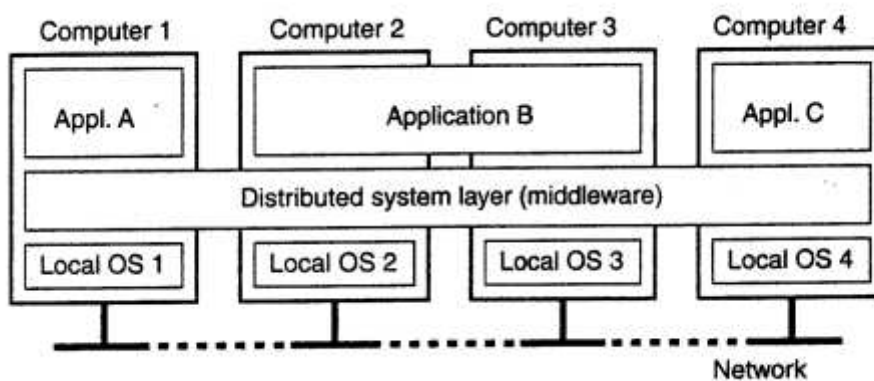
Temos várias definições para o que são sistemas distribuídos, mas podemos defini-la de uma forma geral, como: “Um sistema distribuído é uma coleção de computadores independentes que aparece aos seus usuários como um único sistema coerente.” (TANENBAUM; VAN STEEN, 2007).

Um sistema distribuído é composto por componentes autônomos que colaboram entre si, dando a impressão ao usuário que ele está conectado a um único computador. As diferenças dos computadores e como eles se comunicam são ocultos do usuário, assim como sua organização interna. Usuários conseguem interagir com as aplicações e com o sistema distribuído de maneira consistente e uniforme, independente de quando ou onde a interação ocorre (TANENBAUM; VAN STEEN, 2007).

Como um sistema distribuído é composto por vários computadores individuais e independentes, ele é fácil de escalar/expandir, além disso, mesmo que algumas partes do sistema esteja indisponível, ele continuará disponível para o usuário pelo motivo que mesmo que alguns computadores estejam desabilitados, outros estarão

habilitados. Sistemas distribuídos são organizados por meio de uma camada de software, o qual é colocado no meio de outras duas, a camada-superior sendo composta por usuários e aplicações, e a camada inferior composta por sistemas operacionais e meio de comunicação, como mostrado na Figura 1, por essas características os sistemas distribuídos muitas vezes são chamadas de *middleware* (TANENBAUM; VAN STEEN, 2007).

Figura 1: Organização de um sistema distribuído.



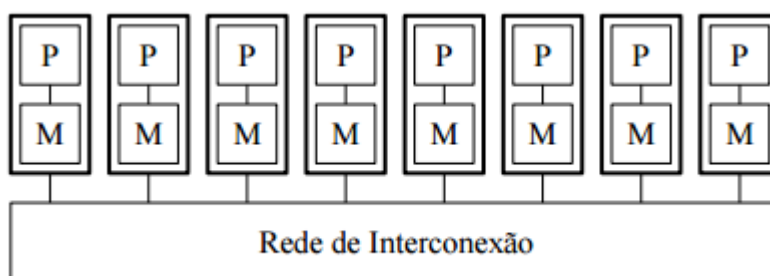
(TANENBAUM; VAN STEEN, 2007)

2.2. MULTICOMPUTADORES

Na arquitetura de multicomputadores, cada processador (P) tem acesso apenas a sua memória local (M), como é mostrado na Figura 2. A comunicação com outras máquinas é feita por envio de mensagens através da rede de interconexão, por isso, estas máquinas são chamadas de sistemas de troca de mensagens, elas podem ser implementadas por máquinas comuns, o compartilhamento de memória é

classificado como NORMA (*NO Remote Memory Access*) e podem ser classificados em duas categorias: MPP (*Massively Parallel Processors*) e COW (*Cluster of Workstations*). MPP são multicomputadores com multiprocessadores, eles são fortemente acoplados por uma rede de interconexão de alta velocidade, COW são máquinas construídas a partir de PCs (*Personal Computers*), sendo ligados por redes de interconexão tradicionais (DE ROSE; NAVAU, 2002) e (CATABRIGA, 2016).

Figura 2: Arquitetura de multicomputadores



(DE ROSE; NAVAU, 2002)

2.3. CLUSTER

Cluster é um sistemas que relaciona dois ou mais computadores, para que eles trabalhem juntos, dividindo uma tarefa e processando-a simultaneamente. Segundo Baker, Buyya e Hyde (2000), um cluster é um tipo de sistema de processamento paralelo ou distribuído, que consiste de uma coleção de computadores *stand-alone* interligados cooperativamente trabalhando juntos como um recurso de computação único e integrado, ele é composto por todos os

componentes disponíveis em uma LAN (*Local Area Network*) com PC's ou *Workstations*.

Sua arquitetura pode mudar dramaticamente, por exemplo, temos os *clusters* com base em componentes *commercial off-the-shelf* e aqueles baseados em processadores SMP (*Symmetric Multi-Processing*) *high-end* e tecnologias de rede personalizada. As configurações de *clusters* físicos também varia bastante, desde um grupo de PC's localizados em uma sala até placas mães armazenadas em um *rack* localizadas em uma sala de serviços. Clusters estão preenchendo o nicho de processamento antes ocupadas por máquinas autônomas super poderosas (BAKER, BUYYA, HYDE; 2000).

Alguns autores classificam *clusters* em três categorias: HPCC (*High Performance Computing Cluster*) são aqueles *clusters* de alto desempenho, indicados para aplicações que exijam bastante processamento; HAC (*High Availability Cluster*) são aqueles focados em manter a aplicação em pleno funcionamento, mantendo a disponibilidade em 99,999% do tempo em cada ano; LBC (*Load Balance Computing*) onde as tarefas são distribuídas o mais uniformemente o possível entre os computadores (BAKER, BUYYA, HYDE; 2000).

2.4. GRID

O GRID tem uma ideia parecida com a dos *clusters*, combinar o poder de processamento de vários computadores ligados em rede para executar uma determinada tarefa, ele irá explorar as potencialidades da rede de computadores e

software para interligar e gerenciar uma grande variedade de recursos computacionais distribuídos em um ambiente já existente (FOSTER, KESSELMAN; 2010).

O conceito de GRID apareceu na década de noventa, segundo Foster e Kesselman (2010), o conceito foi inspirado pela disponibilidade de redes de alta velocidade e pelo desafio dos requisitos exigidos por novas tecnologias, com estes problemas, os pesquisadores começaram a pensar em uma infraestrutura de computação que pudesse fornecer acesso a computação *on-demand* e permitir o compartilhamento seguro, flexível e coordenado de recursos entre coleções dinâmicas de indivíduos, recursos e instituições. Este conceito refere-se às redes de energia elétrica, que forneciam energia por demanda.

Para se identificar um GRID, Foster *et al.* (2008) definiu três características básicas para o sistema:

- Recursos coordenados que não se sujeitam a um controle centralizado: “Um GRID integra e coordena recursos e usuários que vivem em diferentes controle de domínios”
- Utilizar padrões abertos, interfaces e protocolos de propósito geral: “Um GRID é construído a partir de protocolos multi-usos e interfaces que abordam questões tão fundamentais como autenticação, autorização, descoberta de recursos e acesso a recursos, eles devem ser abertos”
- Prover o mínimo em qualidade de serviços: “A utilidade do sistema combinado é significativamente maior do que a da soma das suas partes.”

2.5. CLOUD COMPUTING

A computação em nuvem está sendo utilizada nos tempos atuais para prestar serviços antes conhecidos como “*utility computing*” e “*on-demand*”, ela se baseia em separar os aplicativos dos usuários usando virtualização, oferecendo agregação de recursos e visão única do sistema, permite que os sistemas operacionais de cada usuário possam ser executados de uma forma isolada dos outros sistemas, além de permitir uma gama de aplicações legadas para serem utilizadas. Um relatório da Universidade da Califórnia, resumiu as principais características da computação em nuvem como: "(1) a ilusão de recursos de computacionais infinitos; (2) a eliminação de um compromisso *up-front* por usuários da nuvem; e (3) a capacidade de pagamento para uso como necessário "

A computação em nuvem tem sido confundida muitas vezes com “*utility computing*”, uma categoria de serviços *on-demand*, segundo Buyya, Yeo e Venugopal (2008), *cloud computing* é “um sistema de computação distribuído orientado ao consumidor, que consiste em uma coleção de computadores virtualizados e interconectados que são fornecidos dinamicamente e apresentados como um ou mais recursos computacionais unificados”. Já Vaquero (2009), diz que:

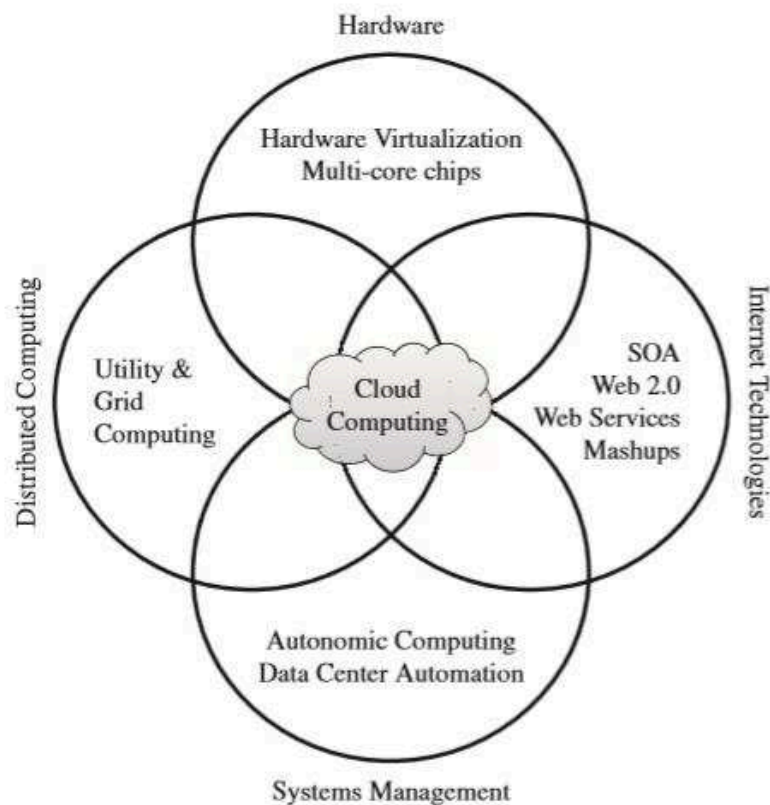
“Nuvens são um grande conjunto de recursos virtualizados facilmente utilizáveis e acessíveis (como hardware, plataformas de desenvolvimento e / ou serviços). Esses recursos podem ser dinamicamente re-configurados para ajustar a uma carga variável, permitindo também uma utilização de recursos otimizada. Este conjunto de recursos é normalmente

explorada por um modelo de pay-per-use em que garantias são oferecidas pelo fornecedor de infraestrutura por meio de SLA (Service Level Agreement) personalizadas.”

Uma das principais razões para as nuvens serem tão populares entre empresas é o conceito de “*anytime*” e “*anywhere*”, o usuário não precisa estar em um local específico para utilizar o serviço, ele pode acessá-lo de qualquer lugar a qualquer hora. Além disso, sistemas baseados em nuvem possuem a capacidade de otimizar os recursos utilizados, como reduzir a quantidade de energia consumida. Porém, migrar cargas de trabalho e dados para um ambiente compartilhado, como é uma nuvem, aumenta o potencial de acesso e exposição a eles, por isso, nuvens precisam ser consistentes em torno de autenticação, gerenciamento de identidade, conformidade e tecnologias relacionadas ao acesso.

Segundo Buyya *et al.* (2011), podemos rastrear as raízes da *cloud computing*, observando o avanço de várias tecnologias, especialmente *hardware* (virtualização, chips *multi-core*), tecnologias da internet (*Web Services*, *Web 2.0*), computação distribuída (*grid*, *clusters*), e gerenciamento de sistemas (computação autônoma, automação de centro de dados), a Figura 3 mostra como a convergência destes elementos contribuíram para a *cloud computing*.

Figura 3: Convergências de elementos na contribuição da computação em nuvem.



(BUYYA; BROBERG; GOSCINSKI, 2010)

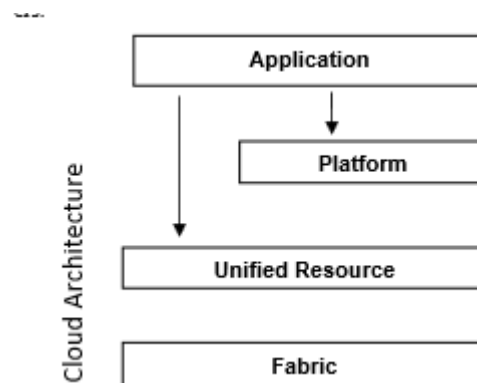
2.5.1. MODELO DE NÍVEIS

Há várias formas de definir as camadas de uma arquitetura de computação em nuvem, mas Foster *et al.* (2008), definiram quatro camadas como mostrado na Figura 4:

- **Nível de Ambiente:** Contém os recursos de hardware, como os recursos computacionais, os de armazenamento e os de rede.

- **Nível de Recurso Unificado:** “Contém os recursos que foram captados/encapsulados de modo que eles podem ser expostos a camadas superiores e usuários finais, como recursos integrados.” (FOSTER *et al.*, 2008).
- **Nível de Plataforma:** Agrega uma coleção de ferramentas especializadas, middleware e serviços para proporcionar desenvolvimento e/ou uma plataforma de implementação.
- **Nível de Aplicação:** Contém os aplicativos que serão executados na nuvem.

Figura 4: Arquitetura de uma nuvem segundo Foster.



(FOSTER *et al.*, 2008)

2.5.2. MODELOS DE SERVIÇOS

Existem três modelos de serviços prestados por nuvens, estes são: *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) e *Software as a Service* (SaaS).

- **IaaS:** *Infrastructure as a Service* são recursos virtualizados (computação, armazenamento e comunicação) sob demanda, uma nuvem permite o fornecimento sob demanda de servidores, que executam diversos sistemas operacionais e *softwares* personalizados. Segundo Nurmi *et. al* (2009), IaaS é considerado como a camada de fundo de sistemas de computação em nuvem. Como exemplo de empresas que oferecem este tipo de serviço, temos a *Amazon Web Service* com o EC2 (*Elastic Compute Cloud*), que oferece principalmente VMs (*Virtual Machines*) com uma pilha de *softwares* que podem ser personalizados, assim como um servidor físico poderia ser, além disso, o usuário possui o poder de realizar diversas tarefas, como : Iniciar e parar o servidor, personalizá-lo com pacotes de *softwares*, configurar permissões de acesso e as regras do *firewall* (BUYYYA; BROBERG; GOSCINSKI, 2010).
- **PaaS:** *Platform as a Service* são nuvens mais robustas e flexíveis para utilização de recursos tecnológicos, elas possuem um nível maior de abstração, tornando-as facilmente programáveis, oferecem um ambiente que os desenvolvedores podem usar para criar e implementar suas próprias aplicações sem se preocupar com a quantidade de processadores ou memória que os aplicativos estão utilizando. Como exemplo de plataforma temos o *Microsoft Azure*, que além de oferecer serviço PaaS, também oferece IaaS e SaaS, além disso ela é aberta e flexível, suporta uma ampla seleção de sistemas operacionais, linguagens de programação, *frameworks*,

ferramentas, bases de dados e dispositivos (BUYYA; BROBERG; GOSCINSKI, 2010).

- **SaaS:** *Software as a Service* são serviços prestados por uma nuvem, onde a aquisição de um software não está relacionado a compra de licenças, o usuário final tem acesso a esse serviço por meio de portais *webs* e é pago pelo o que foi usado. Consumidores estão mudando de programas instalados em *desktops* para serviços *on-line* de *software* que oferecem a mesma funcionalidade, com isso a manutenção de *software* de clientes e o desenvolvimento e teste de *softwares* para prestadores se torna mais simples (BUYYA; BROBERG; GOSCINSKI, 2010).

2.5.3. TIPOS DE NUVEM

A computação em nuvem pode ser dividida em quatro modelos básicos, segundo Mell e Grance (2009):

- **Pública:** Uma nuvem pública pertence a uma organização que disponibiliza e vende os seus serviços para o público ou uma grande parte da indústria. Ela é uma nuvem padrão, onde o prestador disponibiliza recursos através da Internet, este tipo de nuvem pode ser livre ou oferecida em um modelo *pay-per-usage*. Segundo Ambrust (2009), uma nuvem pública é: “Uma nuvem disponibilizada no modelo *pay-as-you-go* para o público em geral.”
- **Privada:** Uma nuvem privada é uma nuvem própria ou alugada por uma empresa e utilizada por apenas ela. Ela, tal como as outras, irá fornecer um

ambiente virtualizado junto com outros recursos computacionais. Segundo Ambrust (2009), uma nuvem privada é: “Um *data center* interno de uma empresa ou organização, não colocados a disposição do público em geral.”

- **Comunitária:** Segundo Mell e Grance (2009) uma nuvem comunitária é: “Uma nuvem compartilhada por várias organizações e suporta uma específica comunidade que tem preocupações em comum (ex: Missão, requisitos de segurança, considerações políticas e conformidades)”.
- **Híbridas:** São duas ou mais nuvens (privada, pública ou comunidade), que permanecem sendo uma única, porém, trabalham juntas por meio de tecnologias padronizadas ou proprietárias, que permitem a portabilidade de aplicações e dados (PELL; GRANCE, 2009).

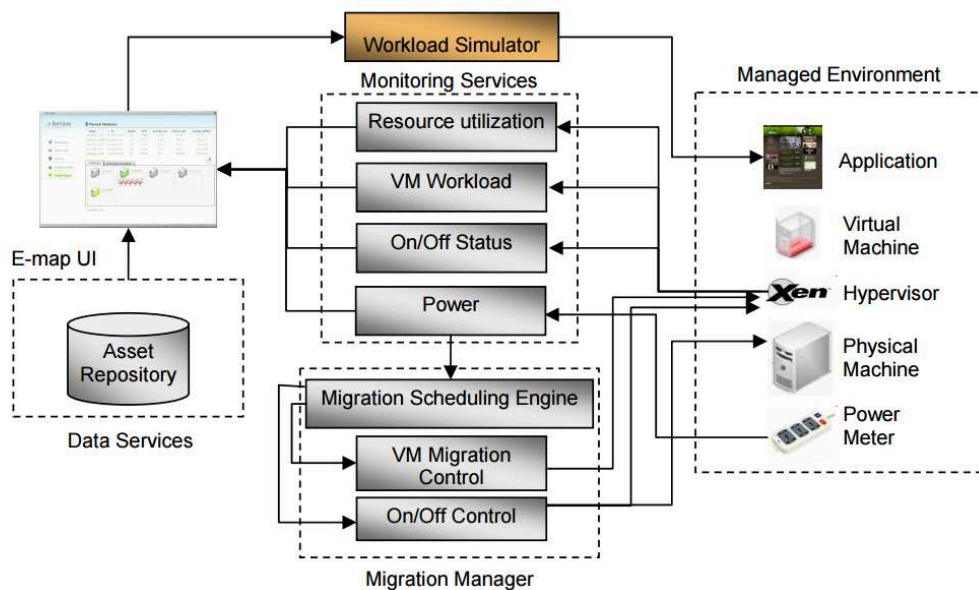
2.5.4. GREEN CLOUD COMPUTING

Green Cloud é uma arquitetura IDC (*Interdomain Domain Controller*) que visa reduzir o consumo de energia em um *data center* sem afetar seu desempenho no ponto de vista do usuário, para isso ele utiliza a tecnologia de migração de máquinas virtuais, esta tecnologia é um dos fatores chaves para nuvens eco-eficientes. O modelo de nuvem verde coloca o foco no gerenciamento dos recursos no *data center*, na economia de energia, ele se baseia fundamentalmente na aplicação de critérios de posicionamento, alocação, redimensão e migração de VMs. Um dos desafios dessa arquitetura é fazer a decisão automaticamente do escalonamento

das migrações das VMs entre os servidores físicos, visando atender as cargas de trabalho com o menor custos de energia possível (LIU *et al.* 2009).

Liu (2009) propõe uma nova arquitetura e um novo sistema exploratório para nuvens verdes. O sistema exploratório irá monitorar uma variedade de fatores e medidas de desempenho do sistema, como, a carga de trabalho, o consumo de energia e a utilização de recursos da nuvem, com isso o sistema conseguirá se adaptar dinamicamente a carga de trabalho e a utilização dos recursos através de migração de VMs. Enquanto a arquitetura reduzirá o consumo desnecessário de energia em um ambiente de nuvem. A Figura 5 mostra a arquitetura proposta e os relacionamentos dos componentes com ela.

Figura 5: Arquitetura de uma nuvem verde proposta por Liu.



(LIU *et al.*, 2009)

2.6. DYNAMIC VOLTAGE AND FREQUENCY SCALING

Dynamic Voltage and Frequency Scaling (DVFS), segundo Le Sueur e Heiser (2010), é uma técnica de gerenciamento de energia comumente usado, onde a frequência de clock de um processador é reduzida para permitir uma redução na tensão de alimentação. Esta redução de consumo de energia, pode ser significativa para a computação, principalmente para cargas de trabalho ligadas a memória. Ele é capaz de reduzir o consumo de energia de um circuito integrado CMOS, tal como um processador de computador moderno, a função utilizada para reduzir a frequência que opera estes circuitos é a seguinte (LE SUEUR, HEISER; 2010):

$$P = CfV^2 + P_{static}$$

Onde C é a capacitância das portas dos transistores (que depende do tamanho da função), f é a frequência operacional e V é a tensão de alimentação (LE SUEUR, HEISER; 2010).

2.7. VIRTUALIZAÇÃO

A virtualização surgiu como uma proposta de solução para a escolha cara e de difícil gerenciamento de colocar cada serviço em um computador para garantir a tolerância a falhas. Ela permite que cada computador hospede diversas máquinas virtuais, onde cada máquina possui um determinado sistema operacional. Caso uma máquina venha a falhar, as outras não serão afetadas, igual o modelo adotado com

multicomputadores, só que com um custo muito mais baixo e de muito mais fácil manutenção (TANENBAUM, 2009).

Além do forte isolamento, as máquinas virtuais apresentam outras vantagens, como: Menos máquinas físicas, gerando economia em *hardware* e em eletricidade; Cada aplicação pode levar consigo seu próprio ambiente; Criação de *checkpoints* e a migração são muito mais fáceis; Podemos fazer simulações de aplicações, ferramentas e demais recursos; Também é disponibilizado serviços de redes virtualmente através de interfaces de rede, switches, roteadores e firewalls virtuais (TANENBAUM, 2009).

3. TRABALHOS RELACIONADOS

Nesta seção irei apresentar alguns trabalhos recentes focados na área de economia de recursos em *cloud computing*, esta pesquisa tem como objetivo apresentar as tecnologias encontradas mais atuais na área. O que se percebeu é que a maioria das pesquisas em eficiência energética tem focado na otimização dos elementos de processamento. Abaixo serão comentados alguns trabalhos nessa área.

Observa-se a forte tendência de dispositivos clientes *front-end* consumir cada vez mais energia em função da interação mais evasiva em *data centers* (*back-end*). Para contornar estes problemas, recursos da nuvem precisam ser alocados não só para satisfazer a qualidade de *software*, mas também para reduzir o consumo de energia. Para isso, Beloglazov, Abawajy e Buyya (2012), propuseram algumas políticas e algoritmos eficientes para *data centers* virtualizados. O algoritmo de alocação de recursos com *energy-aware* proposto utiliza a consolidação dinâmica de VMs, eles propõem três políticas de seleção, *The Minimization of Migrations*, *Highest Potential Growth*, *The Random Choice*:

- *The Minimization of Migrations*: Seleciona o número mínimo de VMs necessárias para migrar de um *host* para reduzir a utilização de CPU (*Central Processing Unit*).
- *Highest Potential Growth*: Quando o limite superior de utilização é utilizado, esta política é utilizada, ela migra as VMs com menor utilização de CPU relativo a capacidade de CPU definido por parâmetros, ele faz este processo

a fim de minimizar o potencial aumento de utilização do *host* e impedir alguma violação de SLA.

- *The Random Choice*: Seleciona um número aleatório de um número de VMs necessárias para diminuir a utilização de CPU de um *host*.

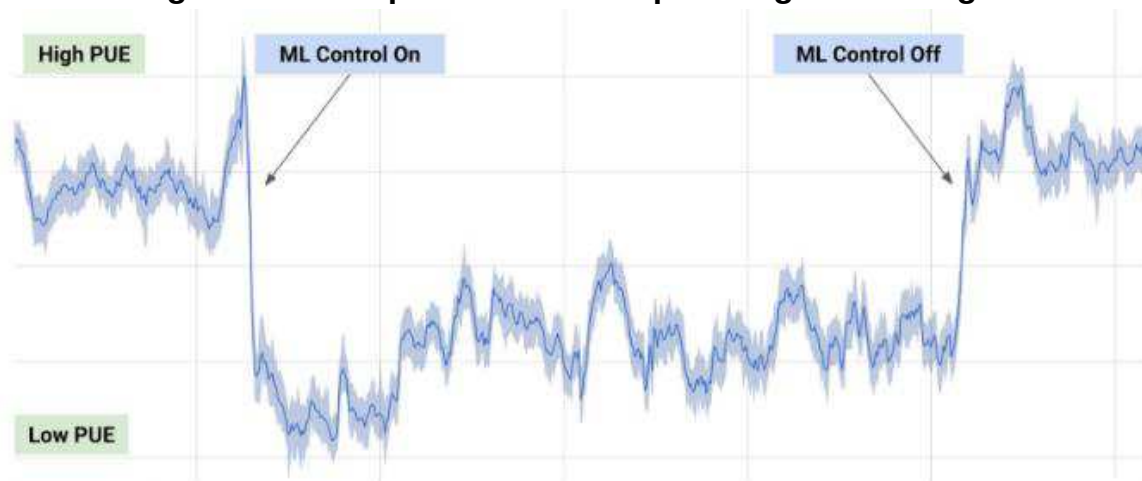
Os resultados mostram que o algoritmo proposto leva uma redução substancial do consumo de energia comparados a algoritmos de alocação de recursos estáticos em ambientes de data centers em nuvem (BELOGLAZOV, ABAWAJY e BUYYA; 2012).

Massimo *et. al* (2015), propõe um algoritmo que analisa possíveis destinos para a migração de VMs e mapeá-los com o custo de mantê-los no destino específico, permitindo que nuvens federadas possam aplicar uma estratégia de gerenciamento de energia dinâmica. Assume-se que o custo de hospedagem está relacionado ao consumo de energia em um local específico durante um intervalo de tempo, assim ele implementa políticas específicas para encontrar o melhor destino para migração.

O algoritmo proposto no artigo é capaz de processar a migração de uma carga de trabalho genérica, o destino selecionado para a migração pode ser o local com espaço disponível e custo de energia mais baixo, para isso o algoritmo cria uma matriz de destino/granularidade que mostra os custos a cada intervalo de destino e de tempo. Com esta matriz podemos aplicar as políticas para escolher o melhor destino, para isso, é proposto um algoritmo que extrai o local de destino com um custo mínimo para cada granularidade e armazena-os em uma matriz com os custos de energia no local e o destino específico (MASSIMO *et al*; 2015).

O Google utilizou uma *machine learning*, a DeepMind, em seu próprio *data center*, conseguindo reduzir a quantidade de energia utilizada na refrigeração em até 40%. Em qualquer ambiente de consumo de energia em larga escala, esse desempenho já seria uma grande melhora, além de ajudar o próprio Google, a máquina também irá ajudar outras empresas que rodam os seus serviços na nuvem do Google. Para treinar a máquina, utilizaram dados históricos que já haviam sido recolhidos por milhares de sensores dentro do *data center*. Também foram treinados dois conjuntos de redes neurais para prever a temperatura e pressão do *data center* na hora seguinte. O objetivo da previsão é simular as ações recomendadas a partir do modelo PUE (*Power Usage Effectiveness*), para garantir que não vão além de quaisquer restrições operacionais (EVANS, GAO; 2016). A Figura 6, nos mostra o desempenho da máquina em um cenário típico de testes:

Figura 6: Desempenho com a DeepMind ligada e desligada



(EVANS, GAO; 2016)

Monteiro e Dantas (2012) propõem um cenário com um *data center* de pequeno porte, em que é considerado o consumo relativo à transmissão e os custos relativos dos dispositivos de redes, indicando a melhor configuração para tal situação. Eles utilizam uma arquitetura de duas camadas, com *switches* de Acesso e de Agregação, dois servidores SMB (*Small Business*) e um roteador *gateway*. Após os experimentos, os resultados indicaram que a energia necessária para o transporte de dados influencia no consumo de energia, além disso existe uma grande vantagem quando utilizamos diversos usuários simultaneamente, por causa que os gastos com o transporte de dados é diluído entre os usuários. Com a simulação do cenário, conseguiram uma economia de cerca de 37% da energia consumida no total. Porém, ao analisar os modelos propostos, percebe-se que o modelo de computação em nuvem tenham uma gerência de energia melhor não perdendo desempenho, a métrica proposta não garante eficiência energética de um modelo sobre o outro, pois, vários fatores são ignorados, como o uso de energias renováveis e o reaproveitamento da energia utilizada, ela não pode ser usada sozinha, pois é uma métrica inicial de medição de eficiência energética, ela demonstra somente o total de energia fornecido ao *Data center*.

Faria (2015) propõe uma abordagem genérica de alocação de recursos energeticamente eficiente, que tende a melhorar o desempenho energético de ambientes de alto desempenho heterogêneos, a partir da seleção de recursos com menores custos na transferência de dados, além dos estados dos nodos e as respectivas eficiências energéticas, para isso foi desenvolvido um algoritmos de escalonamento de recursos, o ENA (*Energy and Network Aware Algorithm*). Foi

provado que esta abordagem é eficiente em alguns casos, como quando os tamanhos dos arquivos de entrada iam aumentando, a proposta conseguiu reduzir consideravelmente o consumo de energia em comparação com outras abordagens existentes.

Bignatto Junior (2016) propõe um modelo que lhe permite estimar o custo energético na virtualização de *desktops*, ele tem como objetivo auxiliar a tomada de decisão sobre o provisionamento dos recursos, a fim de melhorar a eficiência energética. Segundo o modelo proposto, é melhor utilizarmos *desktops* virtuais para aplicações com taxa de transmissão, de quadros e comandos, baixa ou regular.

4. PROPOSTA DEMETER

A proposta deste trabalho é entender e analisar o funcionamento de algoritmos de escalonamento de tarefas em um ambiente de nuvem computacional de três camadas sob a perspectiva do consumo energético. Por meio da análise de métricas de consumo de energia em servidores e equipamentos de transmissão, coletadas via experimentos realizados em um simulador de nuvens computacionais, a influência de quatro diferentes algoritmos de escalonamento de tarefas no consumo energético em três ambientes com escala de complexidade diferentes é avaliada.

Para esta comparação será utilizado um estado base, que sempre será o ambiente proposto utilizando o algoritmo de escalonamento Random. Os outros algoritmos propostos a serem estudados são: Round-Robin, BestDENS e HEROS. Os ambientes e as simulações serão feitas utilizando o *Online Green Cloud Simulation*.

4.1. PROPOSTA DOS ALGORITMOS

Primeiro foi feito um estudo e análise dos algoritmos a serem utilizados neste trabalho, o primeiro algoritmo escolhido foi o Random, caso um ambiente não possua nenhum tipo de escalonador, ele irá distribuir as tarefas aleatoriamente entre os servidores, tornando-o uma perfeita opção de cenário base para comparação com os resultados dos outros algoritmos. O próximo a ser escolhido foi o

Round-Robin, um algoritmo de escalonamento muito conhecido por ser amplamente utilizado em sistemas operacionais que também pode ser aplicado na distribuição de tarefas em *data centers*. Os últimos a serem escolhidos foram: BestDENS e HEROS. O HEROS é baseado no DENS e o BestDENS trata-se de uma implementação aprimorada do DENS que sempre pega o melhor servidor de acordo com os melhores ajustes, ou seja, possuem uma forma de trabalho um pouco parecida, ambos tentam sobrecarregar os servidores que possuem o melhor indicador, de acordo com suas funções, e os outros tendem a ficar em um estado ocioso, ao utilizarmos técnicas auxiliares (como o *Dynamic Shutdown*, por exemplo) poderíamos ter uma noção real da economia nos cenários. Nas seções abaixo, os algoritmos serão melhores abordados, mostrando como é calculado as funções de seleções e como eles funcionam.

4.1.1. RANDOM

Como o próprio nome sugere, ele seleciona um nodo aleatoriamente para executar determinada tarefa.

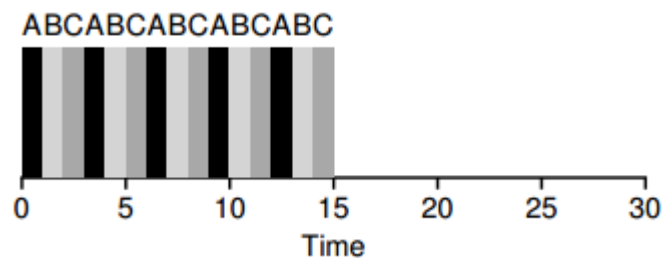
4.1.2. ROUND-ROBIN

O algoritmo Round-Robin executa um processo em uma determinada fração de tempo, após esse tempo acabar ele parte para o próximo processo na fila de

execução. Este comportamento é feito repetidamente até os processos que estão na fila serem concluídos (ARPACI-DUSSEAU, ARPACI-DUSSEAU; 2014).

Na Figura 7 temos três processos (A, B e C) sendo executados pelo algoritmos Round-Robin, cada processo deve ser executado por 5 segundos, o algoritmo possui um time-slice (tempo de execução da tarefa até ser pego a próxima da fila) de 1 segundo.

Figura 7: Round-Robin



(ARPACI-DUSSEAU, ARPACI-DUSSEAU; 2014)

4.1.3. BestDENS

A metodologia DENS, cujo é um acrônimo para *Data center Energy-efficient Network-aware Scheduling*, objetiva alcançar o equilíbrio entre o desempenho individual do trabalho, requisitos de qualidade de serviço (QoS) definidos no *Service Level Agreement (SLA)*, demandas de tráfego e a energia consumida pelo *data center*. O *network-aware* refere-se à capacidade do algoritmo DENS receber e analisar um *feedback* em tempo de execução do *switches* e *links* do *data center*, bem como tomar decisões e ações com base no *feedback* da rede. A abordagem de

escalonamento utilizada pelo algoritmo é projetada para evitar *hotspots* dentro de um *data center*, enquanto minimiza o número de servidores de computação necessários para a execução de tarefas (KLIAZOVICH, BOUVRY, KHAN; 2013).

O algoritmo DENS minimiza o consumo total de energia de um *data center* selecionando os recursos de computação mais adequados para a execução de tarefas, baseando-se no nível de carga e no potencial de comunicação dos componentes do *data center*. O potencial de comunicação é definido como a quantidade de largura de banda de ponta a ponta, fornecida a servidores individuais ou a um grupo pela arquitetura de *data center* (KLIAZOVICH, BOUVRY, KHAN; 2013).

Para um *data center* de três camadas, definimos a métrica M do algoritmo DENS como a combinação ponderada do nível do servidor (f_s), nível do rack (f_r) e nível do módulo (f_m) (KLIAZOVICH, BOUVRY, KHAN; 2013):

$$M = \alpha \cdot f_s + \beta \cdot f_r + \gamma \cdot f_m$$

Onde α , β e γ são coeficientes ponderados que definem o impacto dos correspondentes componentes (servidores, racks, e/ou módulos) sobre o comportamento métrico. A medida que o valor de α aumenta, maior será a carga do servidor selecionado. Valores que possuem β alto, tendem a priorizar racks com

baixa atividade de tráfego de rede. Ao elevarmos o valor de Y , maior são as chances de serem selecionados módulos carregados (KLIAZOVICH, BOUVRY, KHAN; 2013).

O fator relacionado a escolha dos servidores, é uma combinação entre a carga dos servidores $L_s(l)$ e seu potencial de comunicação $Q_r(q)$ (KLIAZOVICH, BOUVRY, KHAN; 2013):

$$f_s(l, q) = L_s(l) \cdot \frac{Q_r(q)^\varphi}{\delta_r}$$

Onde $L_s(l)$ é um fator dependente da carga dos servidores individuais l . $Q_r(q)$ define a carga de *uplink* do rack analisando o nível de congestionamento na fila de saída do *switch*. ζ_r é uma largura de banda sobre os *switches* dos racks e φ é um coeficiente que define as proporções entre L_s e Q_r na métrica. Como no caso das escolhas de servidores, os fatores que afetam os racks e os módulos podem ser descritos pelas seguintes fórmulas (KLIAZOVICH, BOUVRY, KHAN; 2013):

$$f_r(l, q) = L_r(l) \cdot \frac{Q_m(q)^\varphi}{\delta_m} = \frac{Q_m(q)^\varphi}{\delta_m} \cdot \frac{1}{n} \sum_{i=1}^n L_s(l)$$

$$f_m(l) = L_m(l) = \frac{1}{k} \sum_{j=0}^k L_r(l)$$

Onde $L_r (l)$ é uma carga do rack que foi obtida por meio de uma soma normalizada de todas as cargas dos servidores individuais de determinado rack. L_m é uma carga do módulo obtida pela soma normalizada de todas as cargas do rack deste módulo, n e k são o número de servidor em um rack e o número de racks em um módulo respectivamente, $Q_m (q)$ é uma carga de tráfego proporcional aos *switches* presentes no módulo de entrada, ζ_m é a largura de banda sobre o fator de provisionamento dos *switches* do módulo, o fator de nível do módulo f_m inclui apenas um componente de carga l (KLIAZOVICH, BOUVRY, KHAN; 2013).

O fator de carga no DENS é definido como a soma de duas funções sigmóides, apresentada na função abaixo:

$$L_s(l) = \frac{1}{1 + e^{-10(l - \frac{1}{2})}} - \frac{1}{1 + e^{-\frac{10}{\varepsilon}(l - (1 - \frac{\varepsilon}{2}))}}.$$

A primeira função define a forma, enquanto a segunda visa a convergência dos servidores para a obtenção do valor máximo de carga. ζ define o tamanho e a inclinação descendente. A carga do servidor (l) está entre um raio de 0 e 1, para tarefas que possuem carga determinística, l pode ser calculada como a soma das cargas computacionais de todas as tarefas em execução. Abaixo é apresentado o algoritmo DENS em forma de pseudocódigo (KLIAZOVICH, BOUVRY, KHAN; 2013):

DENS Algorithm

Initialization

set weighted coefficient $\alpha = 0.7, \beta = 0.2, \gamma = 0.1$
set proportional coefficient $\varphi = 2$
get server load l
get queue size at access and aggregate switches q

Server selection**FOR** all servers **DO**

compute server load $L_s(l)$, rack load $L_r(l)$, and
module load $L_m(l)$
compute communications potentials of rack $Q_r(q)$
and module $Q_m(q)$
compute metric factors related to servers $f_s(l, q)$,
racks $f_r(l, q)$, and modules $f_m(l)$
compute DENS metric as a weighted sum of
 $f_s(l, q)$, $f_r(l, q)$, and $f_m(l)$

ENDFOR

Select server with highest DENS metric

4.1.4. HEROS

Heterogeneous Energy-efficient Resource allocation Optimizing Scheduler (HEROS) é um algoritmo para o balanceamento de cargas em alocações de recursos, focando na eficiência energética em sistemas heterogêneos, durante o processo de tomada de decisão é considerado a heterogeneidade do sistema, utilizando uma representação holística do mesmo. Servidores que possuem diversos tipos de recursos (processamento, armazenamento, rede, entre outros) e têm estruturas internas variáveis de seus componentes, podem ser utilizados de forma

mais eficiente. O algoritmo é baseado no DENS - apresentado na seção anterior - e no e-STAB (GUZEK, KLIAZOVICH, BOUVRY; 2015).

Um escalonador inteligente e energeticamente eficiente combina duas características contraditórias: Energia consumida e tempo média de resposta. A diminuição do consumo é obtida ao colocarmos servidores ociosos no estado de suspensão e pela consolidação da carga. A minimização do tempo de resposta é operada por uma ampla distribuição de carga de trabalho. E é isso que o HEROS faz, da mesma forma que o DENS, HEROS aloca tarefas para os servidores com as pontuações máximas, esta pontuação é calculada por uma função de decisão, compostas por uma função de seleção de servidor e uma função de potencial de comunicação (GUZEK, KLIAZOVICH, BOUVRY; 2015).

A métrica *Performance per Watt (PpW)* é usada para frisar a eficiência energética e muitas vezes, pode ser usada diretamente para fazer a seleção do servidor mais eficiente em energia, a função é a seguinte (GUZEK, KLIAZOVICH, BOUVRY; 2015):

$$PpW_s(l) = Perf_s(l)/P_s(l),$$

Onde $Perf_s(l)$ é a função de performance e $P_s(l)$ é a função de consumo de energia. Devido a heterogeneidade, é necessário expressar l na mesma escala para todos os servidores. Para isso precisamos usar uma unidade padrão, como MIPS ou

um número de pedidos processados por segundo. Ao utilizarmos somente a função de PpW , temos uma grande desvantagem, pois, os servidores se tornam os mais eficientes em termos energéticos quando estão totalmente carregados, o que leva a uma sobrecarga e redução drástica do desempenho e da eficiência energética. Caso o desenvolvedor queira evitar isso, a função de seleção do servidor é definida como (GUZEK, KLIASOVICH, BOUVRY; 2015):

$$H_s(l) = PpW_s(l) \cdot \left(1 - \gamma \cdot \frac{1}{1 + e^{-\frac{\alpha}{\max l_s} (l - \beta \cdot \max l_s)}}\right)$$

Onde $\max l_s$ é a carga máxima do servidor. O segundo termo da função de seleção é um sigmoid escalado para o domínio L_s e para o intervalo da função PpW para valores altos de carga. O coeficiente α determina a nitidez da inclinação descendente, enquanto β é baseado na carga máxima aceitável do servidor (GUZEK, KLIASOVICH, BOUVRY; 2015).

O potencial de comunicação $Q(u)$ é baseado no utilizado pelo DENS, porém, ao invés de usarmos o tamanho da fila do *buffer*, no HEROS é utilizado a carga de *link* real, a função é definida da seguinte forma (GUZEK, KLIASOVICH, BOUVRY; 2015):

$$Q(u) = e^{-\left(\frac{2u}{U_{max}}\right)^2}$$

Onde u é a carga do *link* atual e U_{\max} é a carga máxima do *link*. Por possuir apenas um componente, esta função faz o HEROS ser aplicável a outros tipos de topologias além da de três camadas. A função de decisão final é obtida pela multiplicação da função de seleção do servidor e da função do potencial de comunicação (GUZEK, KLIAZOVICH, BOUVRY; 2015):

$$F_s(l, u) = H_s(l) \cdot Q_s(u).$$

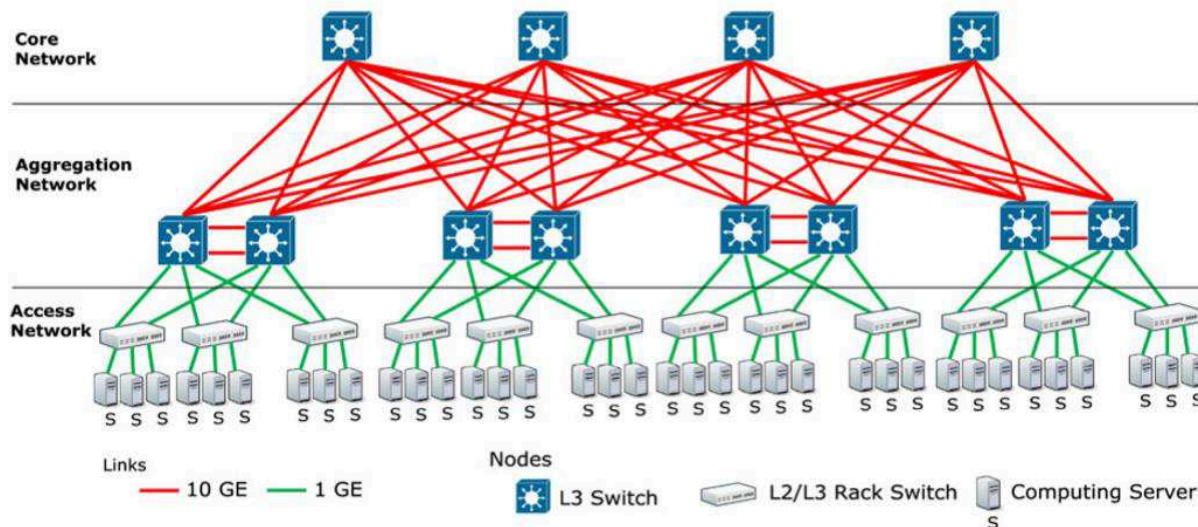
O servidor escolhido para a execução de uma tarefa é aquele com o valor da função de decisão mais alto, em caso de empate, o servidor é escolhido aleatoriamente entre aqueles com os melhores desempenhos. Caso um servidor ocioso seja escolhido, o PpW máximo é multiplicado pelo potencial de comunicação, para fazer uma escolha equilibrada entre economia de energia potencial e balanceamento de carga de trabalho entre os racks (GUZEK, KLIAZOVICH, BOUVRY; 2015).

4.2. PROPOSTA DE AMBIENTES E ARQUITETURA DE TRÊS CAMADAS

Após o estudo do algoritmos, foram feitas as análises dos ambientes a serem simulados, após um tempo de estudo, foram propostos três ambientes. O primeiro possui poucos recursos, visando ter uma melhor análise do desempenho dos algoritmos BestDENS e HEROS, que tendem a sobrecarregar servidores e deixar outros ociosos, como eles comportam em ambientes com poucos recursos como este proposto? No segundo ambiente aumentamos a complexidade dos componentes, para conseguirmos analisar o comportamento dos dois algoritmos citados anteriormente, principalmente se conseguem identificar aparelhos ociosos, qual a economia adquirida e se conseguem garantir a qualidade de serviço. O terceiro ambiente é muito parecido com o segundo, porém, com *links* mais poderosos, para analisarmos se as configurações dos *links* influenciam no desempenho dos algoritmos propostos.

A arquitetura de três camadas foi escolhida por ser a mais comum nos dias de hoje, e como o próprio nome sugere, é dividida em três camadas: Acesso, Agregação e *Core*. A arquitetura pode ser vista na Figura 8. Ela é muito popular por causa da disponibilidade da camada de agregação, que facilita a expansão do número de nodos nos servidores, para mais de 100 mil *hosts*, e ainda mantém os *switches Layer-2* baratos na rede de acesso, o que fornece uma topologia sem *loop* (KLIASOVICH, BOUVRY, KHAN; 2013).

Figura 8: Data center de 3 camadas



(KLIAZOVICH, BOUVRY, KHAN; 2011)

A camada *core* é a raiz da “árvore”, ela é responsável por gerenciar o tráfego entre os roteador da camada de agregação e a entrada/saída do *data center*, esta camada utiliza normalmente um *Layer 3-terminated 10 GigE link*, capaz de suportar um alto nível de *throughout*, desempenho e atender outros requerimentos. É necessário utilizar um *Layer 3 links*, pois eles adquirem a escalabilidade da largura de banda, rápida convergência e conseguem evitar o bloqueio do caminho ou o risco de problemas de transmissão que teriam caso fosse utilizado domínios de *Layer 2* (CISCO, 2016). A camada de agregação, possui muitos *uplinks* da camada de acesso conectado a si, por isso, ele tem como responsabilidade primária agregar as milhares de sessões de entrada e saída no *data centers*.

A camada de agregação também é responsável por fornecer outros tipos de serviços, como: Balanceamento de carga do servidor, *firewall* e *SSL offloading* para

todos os servidores da camada de acesso. A camada de agregação deve ser a mais importante da arquitetura do *data center*, pois, possui uma grande densidade de portas, grande processamento na CPU e os módulos de serviço introduzem implicações únicas no *design* como um todo (CISCO, 2016).

A camada de Acesso é a responsável por manter o *pool* de servidores/*host*, ele desempenha papéis críticos no cumprimento de determinados requisitos, como o agrupamento de NICs, *clustering* e *broadcast containment* (CISCO, 2016).

4.3. PROPOSTA DE SIMULAÇÃO

Para o desenvolvimento dos experimentos deste trabalho foi escolhida a abordagem de simulação, devido à complexidade/custo de um ambiente real de três camadas. Foram pesquisados e estudados diversos simuladores (CloudSim, MDCSim) e optou-se pelo *Green Cloud*. A escolha desse simulador se baseou em sua característica de possuir facilidade de alteração dos valores dos enlaces, a quantidade de *switches* em cada camada, a quantidade de *hosts* e o tamanho das tarefas. Um outro fator diferencial foi a possibilidade de utilizarmos os algoritmos a serem investigados. O simulador permite simular um ambiente composto por milhares de nodos e, também, contabiliza os erros de conexão ou perdas relacionadas ao congestionamento em *switches* de rede. Quanto aos modelos de energia são implementados para cada componente do *data center*.

4.3.1. COMPARAÇÃO COM OUTROS SIMULADORES

O *Green Cloud*, é um simulador ao nível de pacotes, diferente do *CloudSim* e *MDCSim* que são simuladores baseados em eventos. No *Green Cloud* sempre que uma mensagem é transmitida entre servidores, uma estrutura de pacotes com seus cabeçalhos é alocada na memória e todos o processamento de protocolo associado é executado. Também conseguimos simular um *data center* composto por milhares de nodos, enquanto que nos outros dois citados anteriormente nós podemos simular milhões de computadores (KLIAZOVICH *et. al.*, 2010).

Tanto *CloudSim* quando o *MDCSim* implementam modelos de comunicação limitados, contabilizando apenas os *delays* de comunicação e a largura de banda, no *Green Cloud* um dos seus pontos fortes são os detalhes que ele nos oferece enquanto modela os aspectos de comunicação no *data center*, com ele conseguimos captar a dinâmica de protocolos de comunicação amplamente utilizados, tais como IP, TCP, UDP, etc. Diferente do *CloudSim* e *MDCSim*, o *Green Cloud* também contabiliza os erros de conexão ou as perdas relacionadas ao congestionamento em *switches* de rede (KLIAZOVICH *et. al.*, 2010).

Quanto aos modelos de energia, no *Green Cloud* eles são implementados para cada componente do *data center* (servidores, *switches* e racks), além disso eles podem operar no nível de pacote também, permitindo atualizar os níveis de consumo de energia sempre que um novo pacote sai ou chega do *link*, ou sempre que uma nova execução de tarefa é iniciada ou completada no servidor. Além disso, ele possui três diferentes modos de economia de energia, sendo eles: DVFS, DNS e DVFS + DNS (KLIAZOVICH *et. al.*, 2010).

4.3.2. GREEN CLOUD SIMULATOR

O *Green Cloud* é uma extensão do simulador Ns2, ele extrai, agrega e disponibiliza informações sobre a energia consumida pelos elementos de computação e comunicação do *data center* de uma forma sem precedentes. Neste simulador, possuímos três componentes de consumo de energia: os componentes relacionados a computação e comunicação, além dos componentes energéticos relacionados à infra-estrutura física de um *data center* (KLI AZOVICH *et. al.*, 2010).

A eficiência do *data center* pode ser definida em termos do desempenho fornecido por *watt*, este desempenho pode ser quantificado utilizando algumas métricas, como as duas a seguir: *Power Usage Effectiveness* (PUE) e *Data Center Infrastructure Efficiency* (DCiE) (KLI AZOVICH *et. al.*, 2010).

No *Green Cloud*, os componentes do servidor implementam nodos *single core* que possuem um valor predefinido em um limite de potência de processamento em MIPS (milhões de instruções por segundo) ou FLOPS (operações em ponto flutuante por segundo), o tamanho está associado aos recursos de memória/armazenamento. O modelo de energia seguido pelos componentes do servidor depende do estado do servidor e da utilização da CPU. No simulador o consumo de energia aumenta linearmente com o nível de carga da CPU e da frequência operacional. A função que expressa o consumo médio de energia de um servidor, é a seguinte (KLI AZOVICH *et. al.*, 2010):

$$P = P_{\text{fixed}} + P_f \cdot f^3$$

Onde P_{fixed} é a potência consumida que não escala com a frequência de operação f , enquanto P_f é o consumo de energia da CPU dependente da frequência.

Switches e *Links* formam uma camada de interconexão fornecendo *workloads* para qualquer um dos servidores, para execução em tempo hábil. A qualidade do sinal de transmissão em um determinado cabo determina um *tradeoff* entre a taxa de transmissão e a distância do *link*, que são os fatores que definem o custo e o consumo de energia dos transmissores. A proposta de otimização de consumo energético utilizados na rede foram baseados em links DVS, que introduziu um elemento de controle para cada porta do *switch*, que dependendo do tráfego padrão e dos níveis atuais de utilização dos *links*, poderia diminuir as taxas de transmissão. Porém, o efeito do DVS é limitado, uma vez que apenas uma porção (3-15%) da energia consumida está ligada a taxa do *link*. A energia consumida por um *switch* e todos os seus transmissores pode ser definida como (KLIAZOVICH *et. al.*, 2010):

$$P_{\text{switch}} = P_{\text{chassis}} + n_{\text{linecards}} + P_{\text{linecard}} + \sum_{i=0}^R n_{\text{ports},r} + P_r$$

Onde P_{chassis} está relacionado com a energia consumida pelo *hardware* do *switch*, P_{linecard} é a energia consumida por qualquer *active network line card*, P_r corresponde à energia consumida por uma porta (transmissor) executando à taxa r . Nesta função, apenas o último componente é dependente da taxa do *link*, enquanto

outros componentes, como o $P_{chassis}$ e o $P_{linecard}$ permanecem fixos durante toda a duração da operação do *switch*. Portanto, o $P_{chassis}$ e o $P_{linecard}$ podem ser evitados desligando o *hardware* do *switch* ou colocando-o no modo de suspensão. No *Green Cloud* possuímos três algoritmos para a economia de energia em *switches* implementados: DVS, somente o DNS (*Domain Name System*) ou DVS+DNS (KLIAZOVICH *et. al.*, 2010).

Workloads são tipicamente modelados como uma sequência de tarefas que podem ser divididas em um conjunto de tarefas. As tarefas podem ser dependentes, exigindo uma saída de outra tarefa para iniciar a execução, ou independentes. A execução bem sucedida de um *workload* depende de uma conclusão bem sucedida de seus dois componentes principais: processamento e comunicação. O primeiro define a quantidade de processos que devem ser executados antes de um determinado prazo em uma escala de tempo, no caso, utilizamos este *deadline* para a garantia de QoS. O segundo componente define a quantidade e o tamanho das transferências de dados que devem ser executadas antes, durante e após a execução do *workload* (KLIAZOVICH *et. al.*, 2010).

O componente de comunicação é composto de três partes: O tamanho do *workload* e o tamanho interno e externo das comunicações do *data center*. O tamanho do *workload* define os números de *bytes* que devem ser transmitidos aos servidores antes que uma execução de *workload* possa ser iniciada. O tamanho das comunicações externas, corresponde ao resultado da execução da tarefas, são os dados transmitidos para fora da rede do *data center* no momento da conclusão da tarefa. O tamanho das comunicações internas do *data center* define a quantidade de

dados que podem ser trocados com outro *workload* que pode ser executado no mesmo servidor ou em um diferente. No *Green Cloud*, a comunicação interna é realizada com um *workload* escolhido aleatoriamente (KLIAZOVICH *et. al.*, 2010).

4.3.3. INTERFACE WEB DO SIMULADOR GREEN CLOUD

A interface WEB do simulador Green Cloud foi lançada em 2016, com o intuito de deixar o simulador mais acessível, abstraindo a complexidade do código utilizado para as simulações e deixando o usuário preocupado apenas com os parâmetros a serem utilizados em determinada simulação. É possível modificar diversas características da arquitetura e tarefas a serem simuladas, como mostrado na Figura 9.

Figura 9: Interface WEB do simulador Green Cloud

✔ Run GreenCloud Simulation

Simulation Setup

Datacenter Load ⓘ ▬

Task Scheduler ⓘ Green

Energy Management

Hosts

DVFS ⓘ

Dynamic shutdown ⓘ

Network Switches

DVFS ⓘ

Dynamic shutdown ⓘ

Shutdown/wake up time (s)

Virtualization

DVFS ⓘ

VM static configuration ⓘ

Data Center Topology

Topology ⓘ three-tier small

Topology Parameters

Core switches

Aggregation switches

Access switches per pod

Servers per rack

Total servers

Network links

Core to aggregation Gb/s ms

Aggregation to access Gb/s ms

Access to host Gb/s ms

Task Description

Task size (MIPS) ⓘ

(< <http://greencloud.uni.lu/> >)

É possível escolhermos a carga do *data center*, quais tecnologias queremos utilizar (DVFS, *Dynamic Shutdown*, Virtualização), quantos *switches* e servidores teremos, além dos *links* que conectam os respectivos componentes. Também escolhemos o tamanho da tarefas (em MIPS), o tempo de *deadline* entre outras coisas.

Além disso, após a execução da simulação, gráficos com os resultados são fornecidos pela ferramenta, deixando a análise dos resultados um pouco mais fácil de ser entendido.

5. CENÁRIOS E RESULTADOS EXPERIMENTAIS

Neste capítulo serão propostos três cenários a serem utilizados nas simulações, todos seguirão a topologia de três camadas, já apresentada anteriormente. Devido às limitações do simulador, não temos como alterar as configurações de *hardware* dos servidores, porém, podemos fazer alterações nos *links*, nas tarefas e na topologia da rede, como mencionado no último parágrafo da proposta.

Os ambientes apresentados nas seções abaixo, foram baseados no modelo de três camadas, são ambientes fictícios, criados apenas para simulações e não condizem com arquiteturas reais. O modelo de três camadas foi escolhido por ser um dos mais utilizados atualmente. Como já informado em seções anteriores, as características dos ambientes foram escolhidas para termos uma certa variação de componentes, para analisarmos como cada algoritmos funciona com uma determinada quantia de componentes.

Utilizaremos o algoritmo Random para obtermos resultados que serão utilizadas nas comparações dos algoritmos na próxima seção, vamos utilizar as seguintes métricas:

- Consumo Total;
- Consumo dos *switches* do *core*;

- Consumo dos *switches* de Agregação;
- Consumo dos *switches* de Acesso;
- Consumo dos servidores;
- Entre outras;

Nos quadros abaixo, são exibidos as configurações de *links* e tarefas utilizados nos ambientes, a configuração utilizada para os *links* é baseada no trabalho de Baliga *et al*, (2011).

Quadro 1: Configuração das tarefas dos ambientes

Configuração das tarefas	
Tamanho da tarefa (MIPS):	600000
Memória alocada durante a execução da tarefa (<i>task memory</i>):	3000000
Armazenamento disponível para execução da tarefa (<i>task storage</i>):	600000
Bytes a serem enviados para o servidor, afim de executar a tarefa (<i>task description size</i>):	10000
Deadline da tarefa (s):	7
Bytes enviados quando a tarefa é realizada com sucesso (<i>task output</i>):	350000

Quadro 2: Configuração dos *links* dos ambientes

Configuração dos <i>links</i>		
	Cenários 1 e 2	Cenário 3
Core para Agregação	10 Gb/s	100 Gb/s
Agregação para Acesso	1 Gb/s	10 Gb/s
Acesso para os <i>Hosts</i>	1 Gb/s	1 Gb/s

5.1 CENÁRIO 1

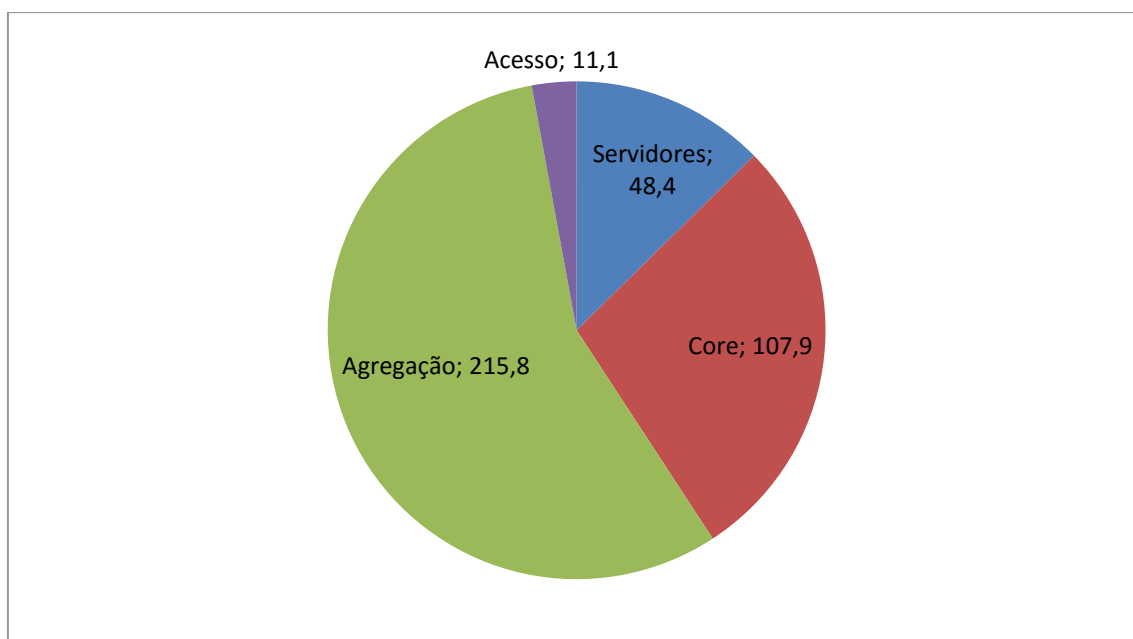
O cenário apresentado no quadro 3, é classificado como um ambiente de pequeno porte, pois, possui poucos *switches* e servidores. O propósito de colocar um ambiente desta magnitude é ver como os algoritmos se comportam com determinadas características.

Quadro 3: Configuração do primeiro cenário

CENÁRIO 1	
<i>Core Switches</i>	2
<i>Switches</i> de Agregação	4
<i>Switches</i> de Acesso	2
Servidores por <i>rack</i>	4
Total de servidores	16
Número de Tarefas executadas	4283

A Figura 10 exibe os consumos energéticos do cenário 1, para realizar tal simulação, foram utilizados o algoritmo Random e o simulador *online Green Cloud Simulation*. O consumo total deste cenário é de 383,2 watt/hora, sendo o principal consumidor os dispositivos de rede, principalmente os da camada de Agregação. Apenas uma tarefa foi detectada como não podendo ser concluída antes do prazo final, estipulado pelo *deadline*, então ela foi descartada e sinalizada como uma tarefa falha. Esta simulação foi feita usando uma carga de 70% no *data center*.

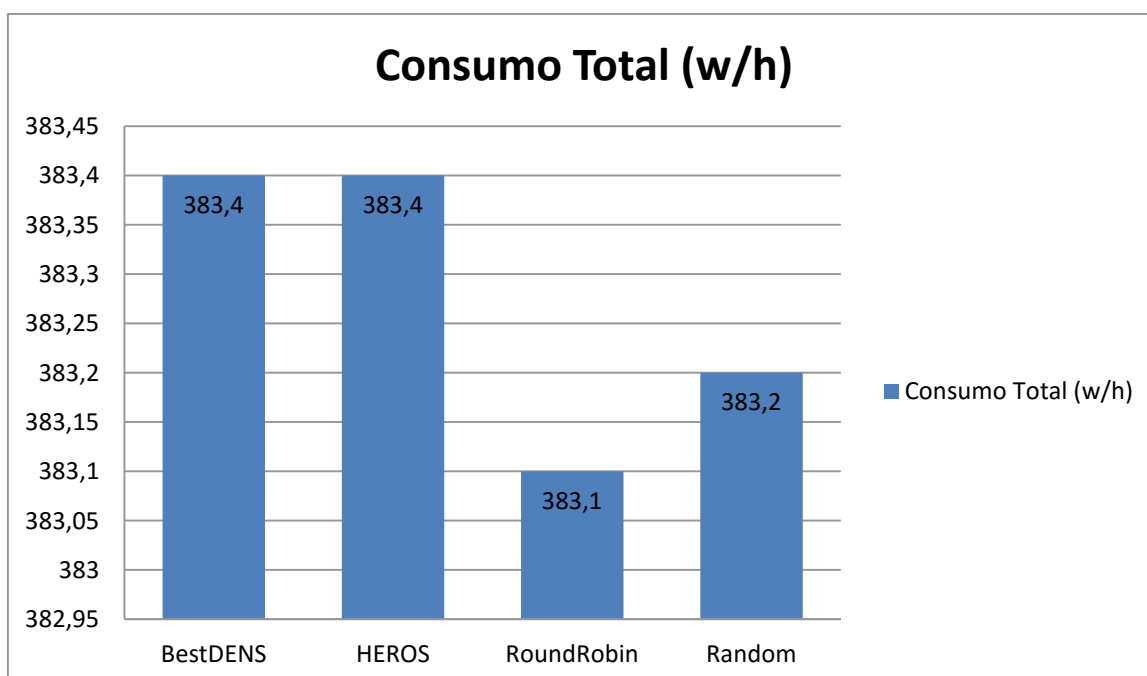
Figura 10: Consumo dos aparelhos do cenário 1 com o Random



5.1.1. SIMULAÇÃO DO CENÁRIO 1

O primeiro cenário a ser simulado, possui poucos servidores e dispositivos de rede, apesar de sua baixa complexidade, é importante analisarmos o desempenho dos algoritmos estudados em arquiteturas menores, como essa. A Figura 11 já nos surpreende com seus resultados, os algoritmos BestDENS e HEROS possuem um desempenho energético pior que o apresentado pelo Random, além disso, o ganho com o Round-Robin é mínimo: 0,1 watts/hora.

Figura 11: Consumo total do cenário 1



Quando analisamos o impacto dos algoritmos no consumo dos dispositivos de rede, fica evidente que eles não são voltados para ambientes pequenos. A Figura 12 apresenta os números obtidos após a simulação, todos possuem o mesmo consumo nos *switches core* e de acesso. O BestDENS e o HEROS conseguem piorar o consumo nos *switches* de Agregação, comparado com o nosso cenário proposto na seção 5.1, pois eles aumentam a carga de determinados *switches*, mas os que ficam ociosos não são colocados no modo *sleep*. O mesmo ocorre ao analisarmos os números da Figura 13, ambos os algoritmos citados anteriormente consomem mais que o estado de partida, o único que consegue uma redução mínima é o Round-Robin, além disso, nenhuma tarefa executada pelos seus servidores falharam, diferente do Random, que possui uma falha.

Figura 12: Consumo dos switches do cenário 1

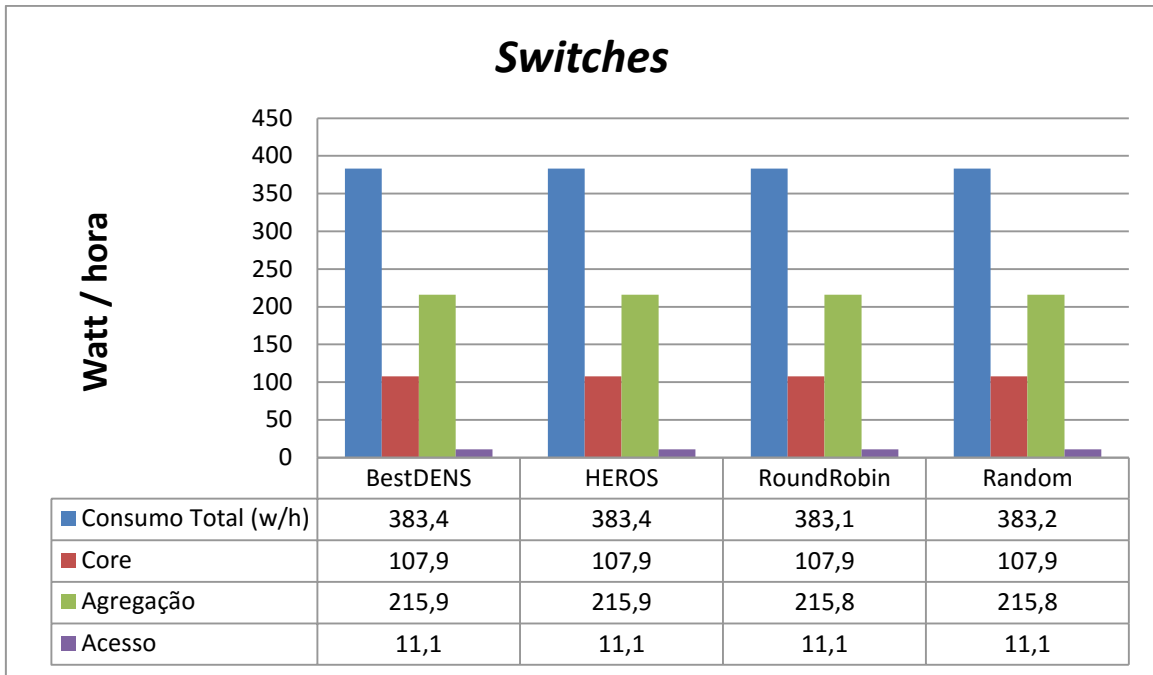
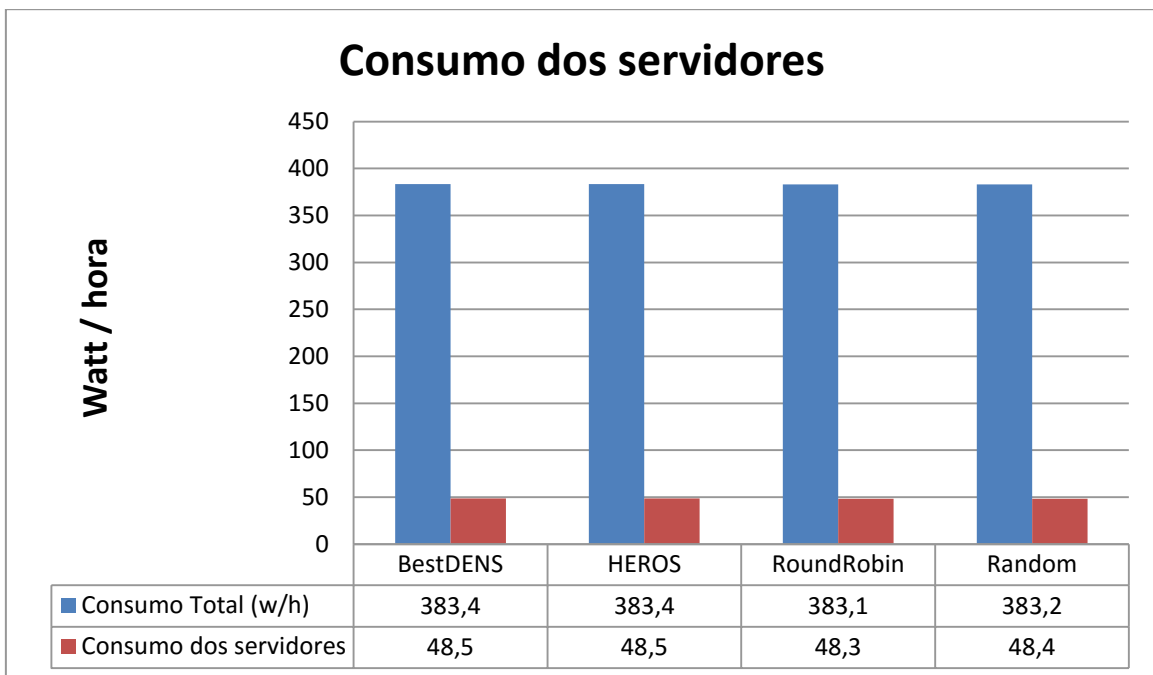


Figura 13: Consumo dos servidores do cenário 1



Partindo para a análise da carga dos servidores e da distribuição de tarefas, nós conseguimos perceber grandes diferenças entre os quatro algoritmos estudados neste trabalho. A Figura 14, mostra a carga média (Normalizada, no eixo Y, sendo 0 igual a nenhuma carga e 1 utilizando 100% dos recursos disponíveis) por servidor (eixo X) no *data center*. Analisando os gráficos apresentados, podemos perceber que o BestDENS e o HEROS apresentam comportamentos opostos em alguns momentos, por exemplo: O primeiro ocupa mais os primeiros computadores do servidor, enquanto o segundo os últimos. Porém, ambos apresentam irregularidades na hora de distribuir a carga, enquanto alguns computadores quase chegam ao 100%, outros são poucos utilizados. Os algoritmos Random e Round-Robin são os que fazem a melhor distribuição, sendo que o segundo distribui de forma igual em todos os computadores. Random e Round-Robin tomam decisões uniformizadas, o primeiro seleciona uma máquina de distribuição aleatória, o segundo atribui ciclicamente tarefas a máquinas, uniformizando a carga por padrão.

O comportamento de distribuição igual de tarefas do Round-Robin fica mais evidente ao analisarmos a Figura 15, que nos mostra gráficos com a distribuição média de tarefas (eixo Y) entre os servidores do ambiente (eixo X). Analisando a figura, ela só comprova o que já havia sido dito anteriormente, BestDENS e HEROS não distribuem regularmente as tarefas entre os computadores, alguns chegam a executar aproximadamente 400 tarefas, enquanto outros ficam próximos a zero. O Random distribui de uma maneira aleatória, como o próprio nome sugere, e o Round-Robin comprova que ele distribui de maneira igualitária entre os computadores, onde todos executam a mesma quantidade de tarefas.

Figura 14: Gráficos de carga por computador no cenário 1.

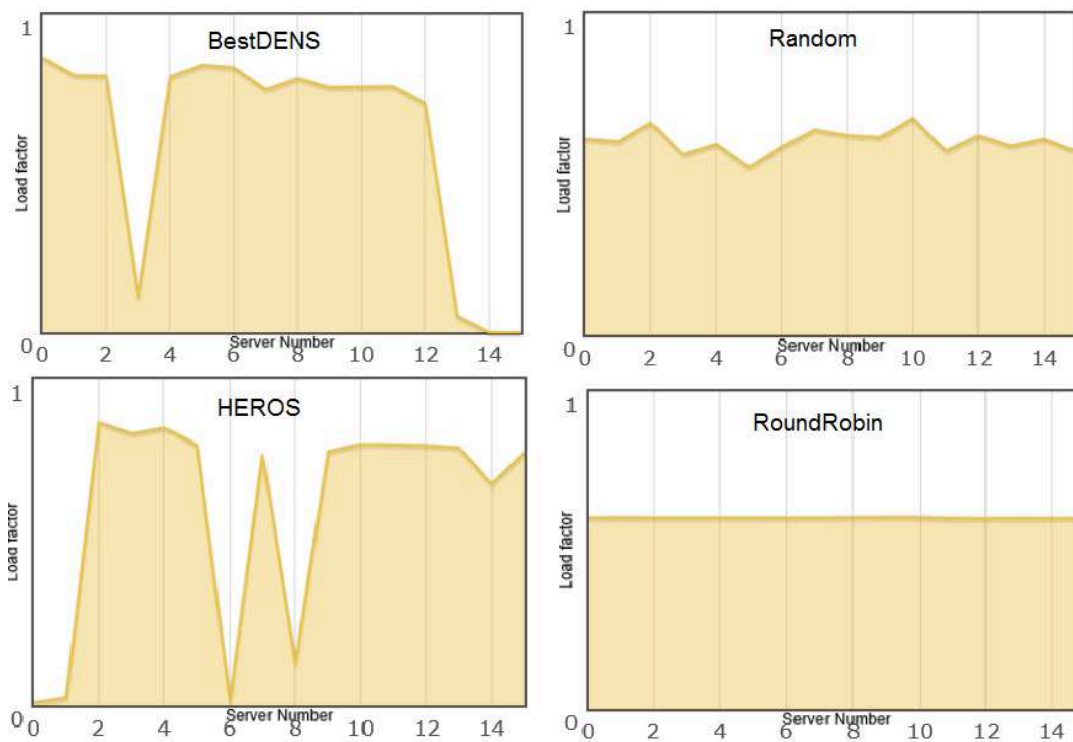
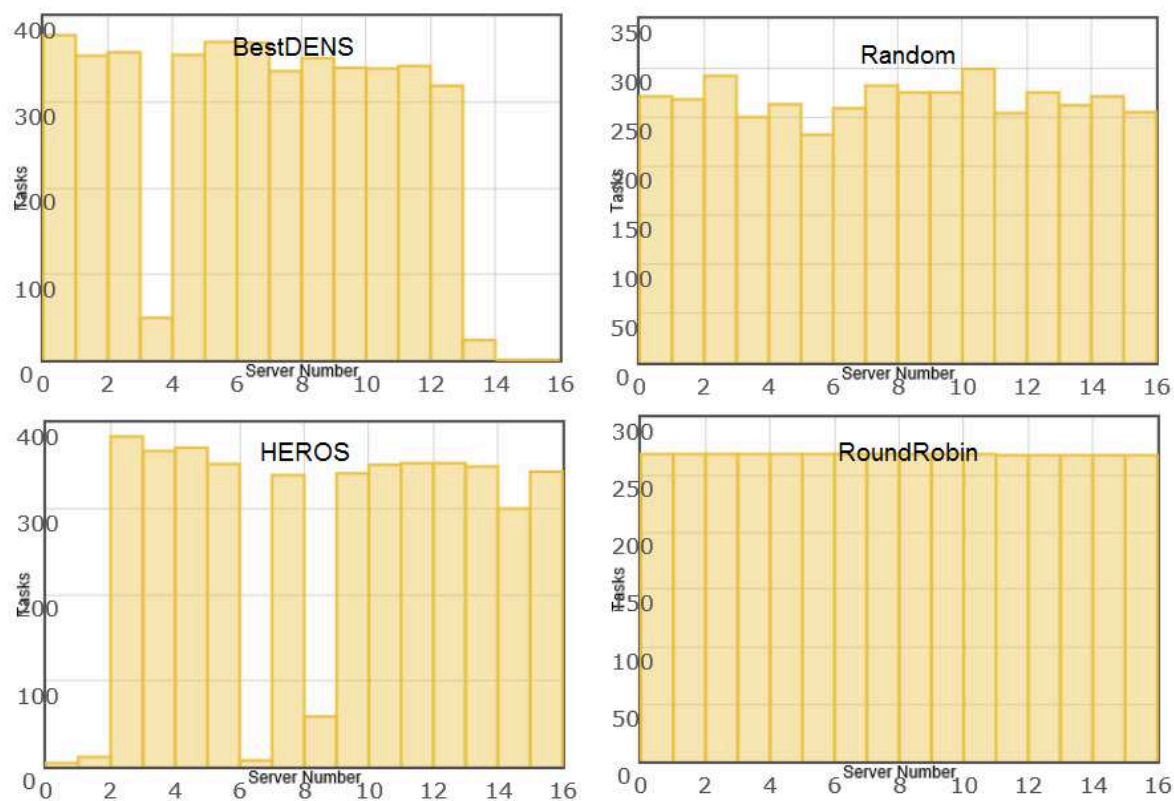


Figura 15: Gráficos de distribuição de tarefas no primeiro cenário.



5.2 CENÁRIO 2

O cenário apresentado no quadro 4, possui configurações de um ambiente de pequeno/médio porte, se levarmos em consideração o ambiente padrão dos simulador.

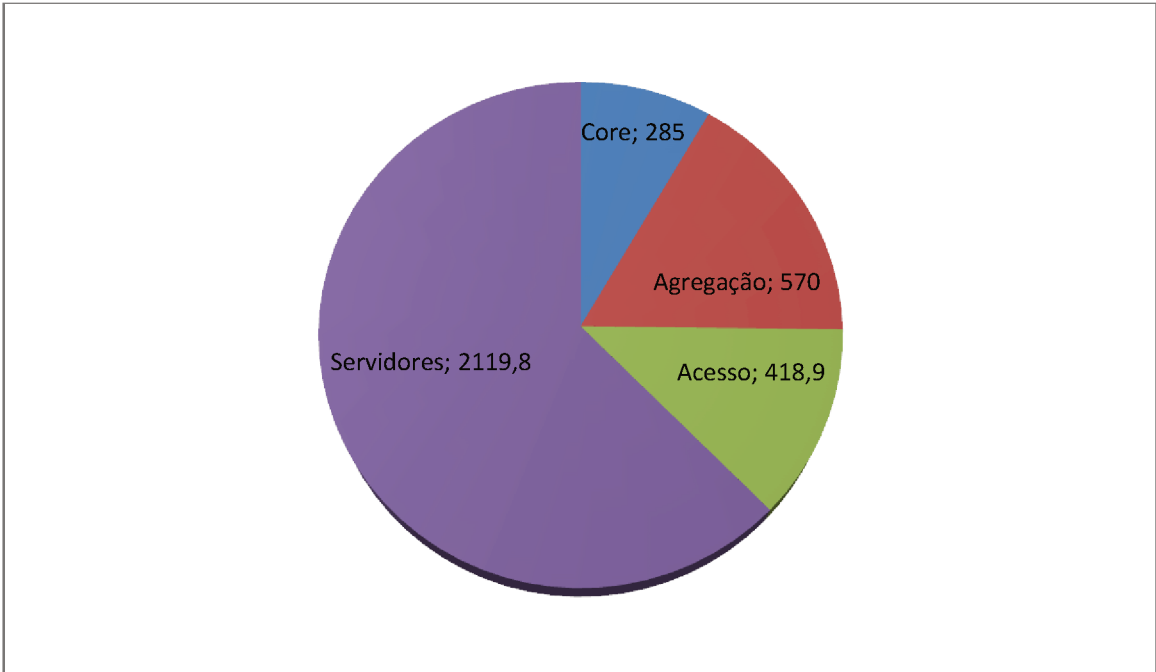
Quadro 4: Configuração do segundo cenário

CENÁRIO 2	
Core Switches	5
Switches de Agregação	10

Switches de Acesso	30
Servidores por rack	6
Total de servidores	900
Número de Tarefas executadas	100654

A Figura 16 exibe os consumos energéticos do cenário 2, para realizar tal simulação, foram utilizados o algoritmo Random e o simulador *online Green Cloud Simulation*. O cenário 2 tem um consumo de 3393,7 watt/hora e não possui nenhuma falha de tarefa. Esta simulação foi feita usando uma carga de 70% no *data center*.

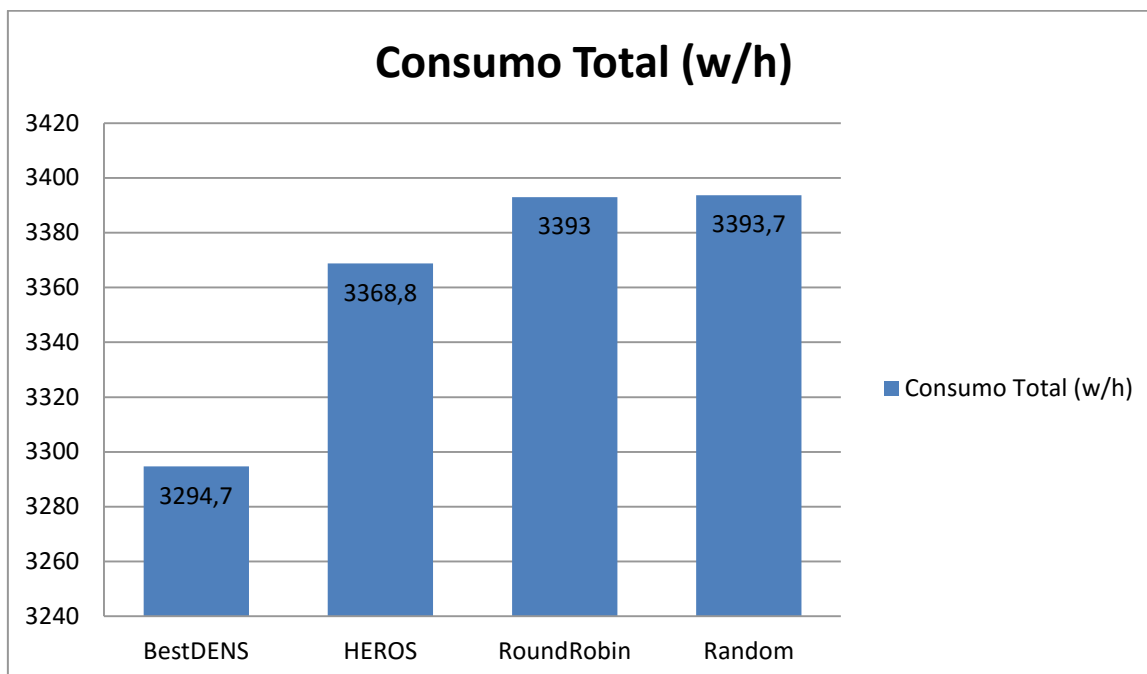
Figura 16: Consumo dos aparelhos do cenário 2 com o Random



5.2.1. SIMULAÇÃO DO CENÁRIO 2

Após a simulação do segundo cenário, temos como resultado os gráficos abaixo, analisando-os podemos tirar algumas conclusões. A que fica mais evidente é mostrada logo na Figura 17, onde podemos ver o consumo total de watts por hora do *data center* com o algoritmo proposto. Se analisarmos apenas o consumo total, o melhor escalonador a ser utilizado para o ambiente 2 é o BestDENS, tendo uma economia de 99 watts por hora, comparado ao ponto de partida (com algoritmo Random).

Figura 17: Consumo total do cenário 2



Porém, os algoritmos não possuem um real impacto no consumo dos dispositivos de rede, como observado na Figura 18. Em todos os ambientes simulados o desempenho é igual, mas há uma explicação para isso, nenhum dos algoritmos estudados neste projeto, altera o comportamento direto dos dispositivos de rede. Eles trabalham diretamente com os servidores, por isso, a diferença apresentada no desempenho é exibida quando comparamos a energia consumida por eles, como pode ser vistos na Figura 19.

Figura 18: Consumo dos switches do cenário 2

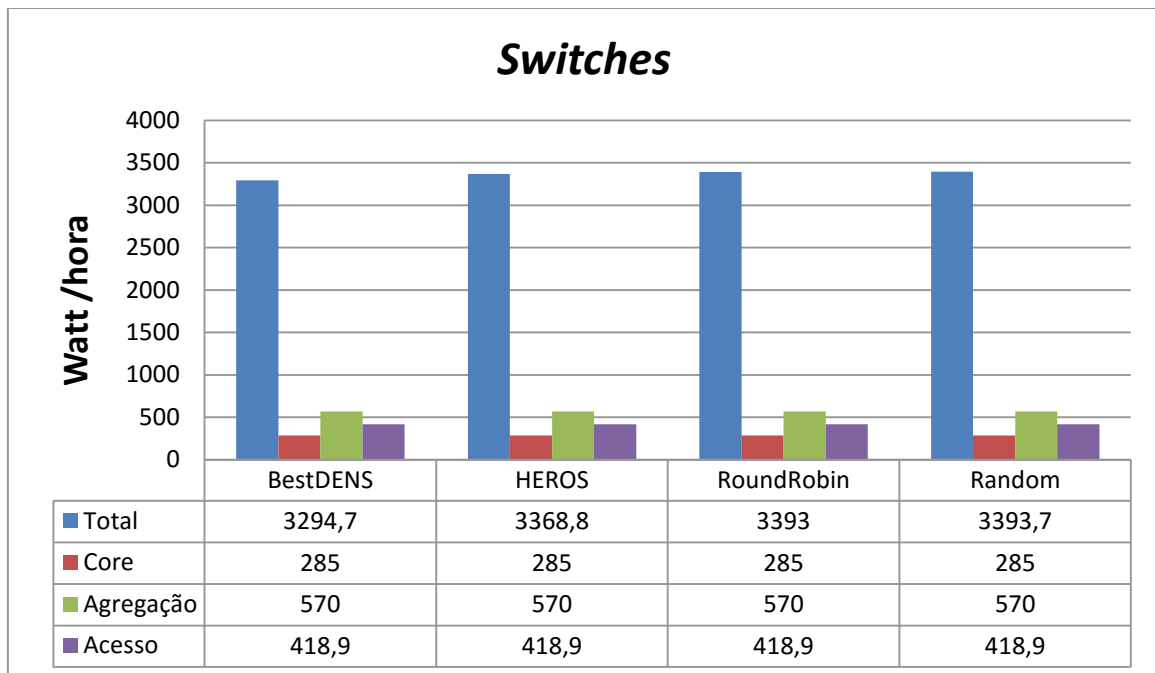
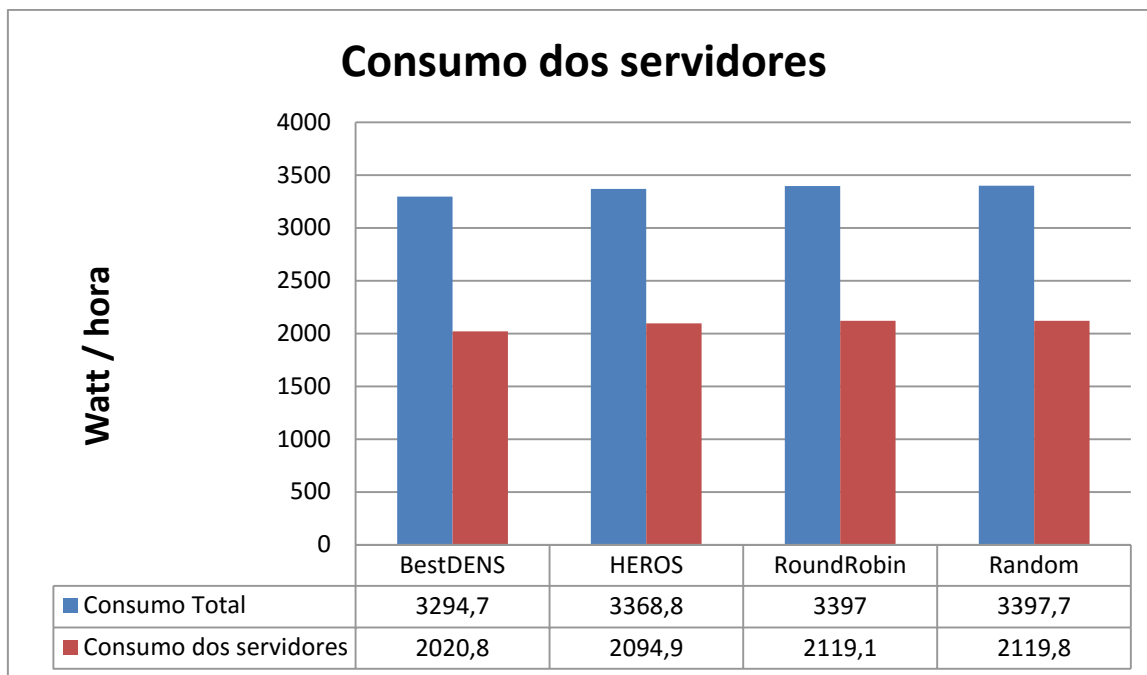
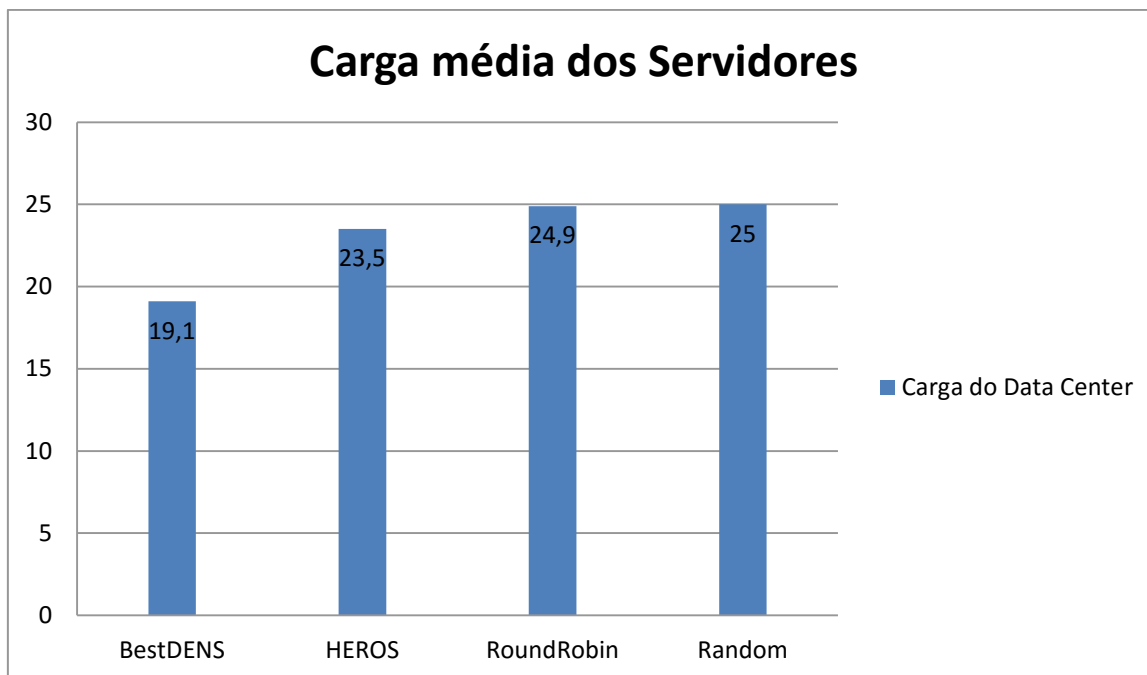


Figura 19: Consumo dos servidores do cenário 2



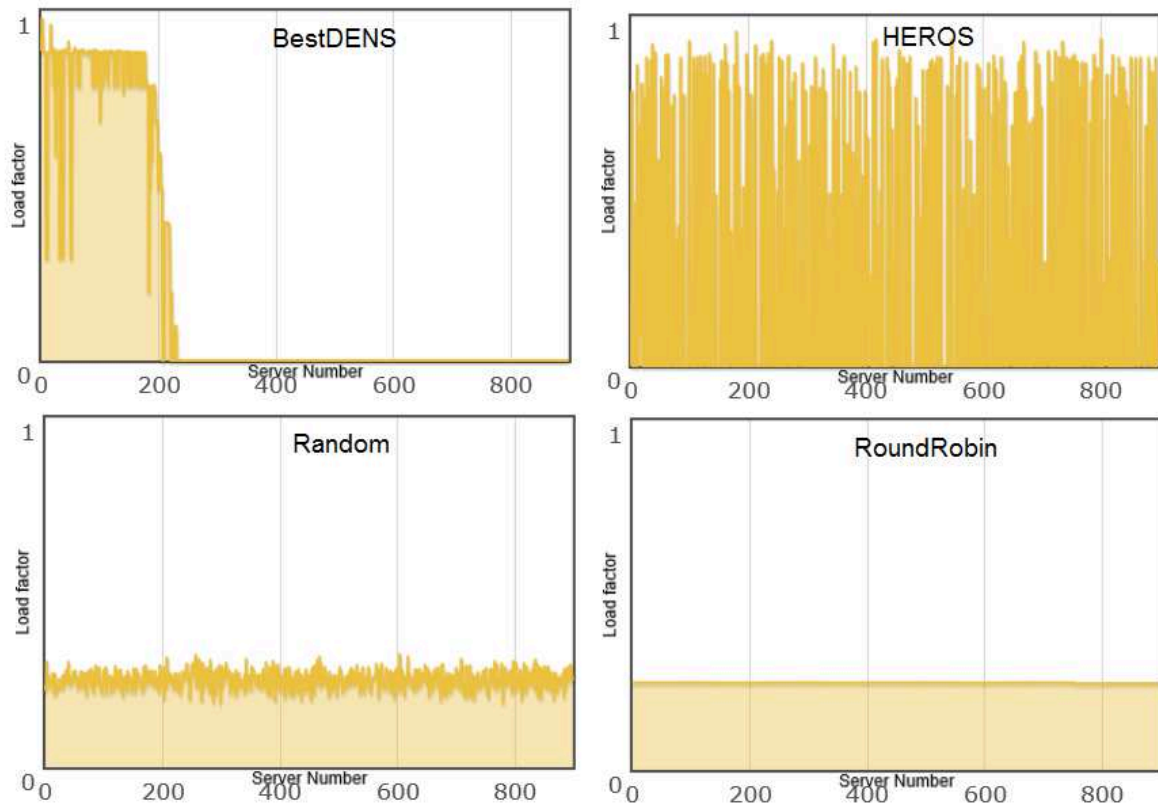
Analisando apenas o desempenho energético dos algoritmos, o BestDENS é o que se sai melhor perante os outros, porém, não podemos levar apenas isso na hora de escolhermos qual utilizarmos. Uma questão importante é saber se todas as tarefas estão sendo executadas pelo servidor, além da carga utilizada no *Data Center*. Em relação a média da carga no servidores(Figura 20), o BestDENS continua sendo o melhor, tendo a menor média de ocupação (19,1, sendo 100 a carga total e 0 nenhuma carga no servidor.).

Figura 20: Carga do *Data center* do cenário 2



Ao analisarmos a Figura 21, podemos notar que os primeiros computadores do cenário simulado com o BestDENS, utilizam quase todo o fator de carga de seus processadores. No HEROS nós temos uma maior distribuição, porém, podemos notar que muitos computadores utilizam diversos recursos enquanto outros utilizam poucos ou quase nenhum. Nós temos uma distribuição mais igual quando olhamos os gráficos dos algoritmos Random e Round-Robin, sendo que o segundo distribui de forma igual entre todos os computadores. Round-Robin e Random são os escalonadores mais ineficientes em termos de energia, quando conseguimos colocar os servidores e *switches* ociosos em modo *sleep* ou são desligados, pois, eles não permitem que nenhum dos componentes seja desligado durante a execução das tarefas.

Figura 21: Gráficos de carga por computador no cenário 2.

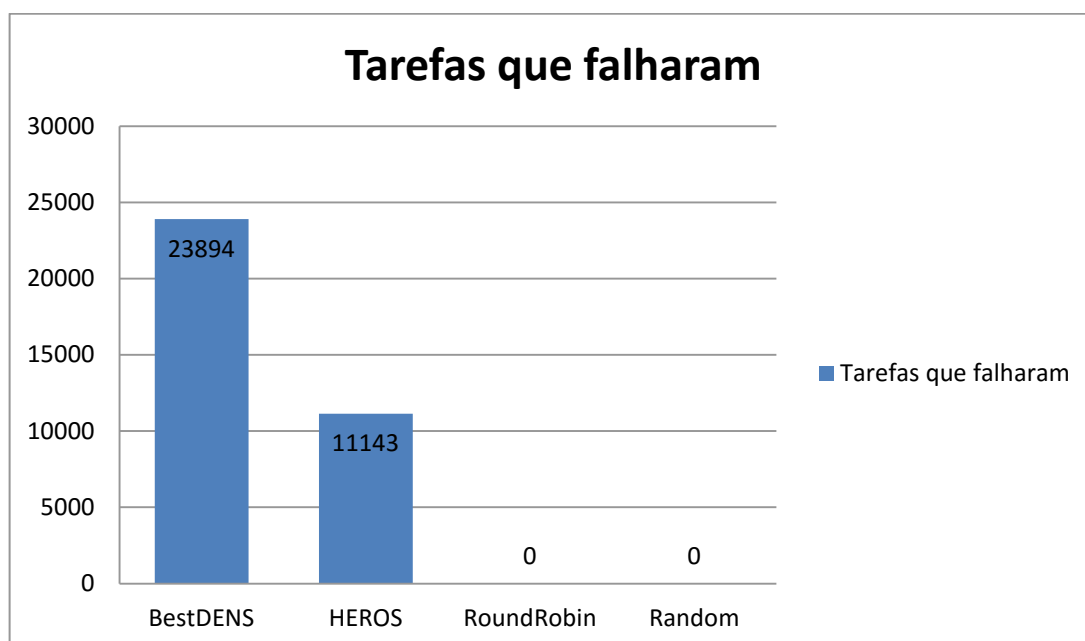


Na teoria o BestDENS consegue uma melhor consolidação da carga de trabalho, impedindo o sobrecarregamento de servidores e *switches*, com isso possuímos uma melhor eficiência energética. A carga média de um servidor é de aproximadamente 0,9 (em uma normalização de 0 a 1), isso garante que nenhum atraso adicional seja colocado nas comunicações feitas pelos servidores, não causando congestionamentos.

Quando analisamos a quantidade de tarefas que falharam durante a simulação(Figura 22), notamos que tanto o BestDENS quanto o HEROS, os dois com melhores desempenhos até então, possuem um alto nível de falhas. Das

100654 tarefas que devem ser executadas, obtivemos um índice de falha de cerca de 24% e 10%, nos algoritmos citados anteriormente respectivamente, são valores bem altos pros padrões recomendados na computação, o recomendado seria algo em torno de 0,1% ou menos. Apesar de não ter um grande desempenho na redução do consumo de energia, o algoritmo de Round-Robin não falha em nenhuma tarefa e reduz o consumo total. Podemos indicar o grande desempenho energético dos algoritmos HEROS e BestDENS ao grande número de tarefas que falharam, pois, elas não chegam a serem processadas pelos servidores, não aumentando a carga dos servidores e consequentemente a energia consumida por eles.

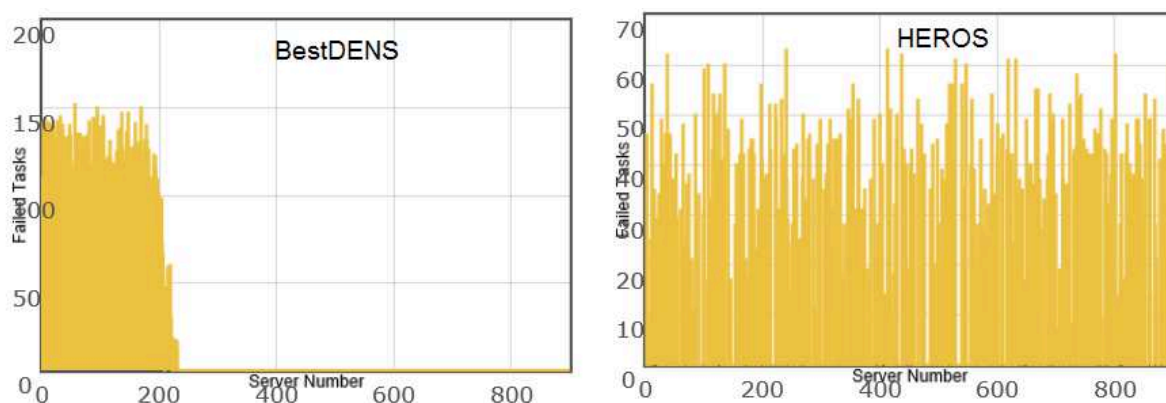
Figura 22: Quantidade de tarefas que falharam do cenário 2



Na Figura 23, podemos notar a quantidade de tarefas que falharam em cada servidor. Ao compararmos com a Figura 21, podemos notar que a distribuição de falhas é igual a de carga nos servidores, ambos os algoritmos possuem

computadores com altos índices de carga, o que acarretaram em diversas falhas, logo, quanto menor o fator de *load* do servidor, menor a chance de haver falhas.

Figura 23: Tarefas que falharam por servidor no cenário 2



5.3 CENÁRIO 3

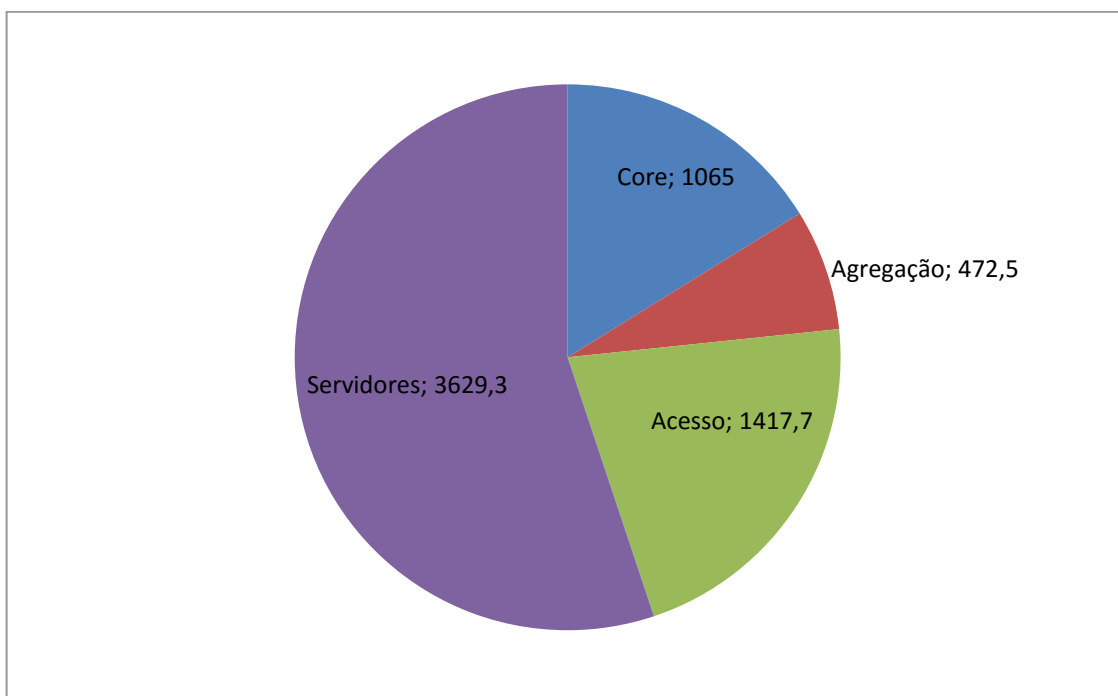
O cenário apresentado no quadro 5, apresenta uma arquitetura de alta velocidade, com *links* de até 100Gb/s, possuindo também mais *switches* de acesso que os outros cenários apresentados até agora. Além disso, possuímos mais servidores, o que nos dará uma melhor avaliação no desempenho dos algoritmos aplicados.

Ao simularmos o cenário com o algoritmo Random como ponto de partida, nós temos os consumos apresentados na Figura 24. Podemos perceber um alto consumo nos servidores e *switches* de acesso, além disso, não há nenhuma falha de tarefa. Esta simulação foi feita usando uma carga de 30% no *data center*.

Quadro 5: Configuração do terceiro cenário

CENÁRIO 3	
Core Switches	2
Switches de Agregação	4
Switches de Acesso	256
Servidores por rack	3
Total de servidores	1536
Número de Tarefas executadas	174449

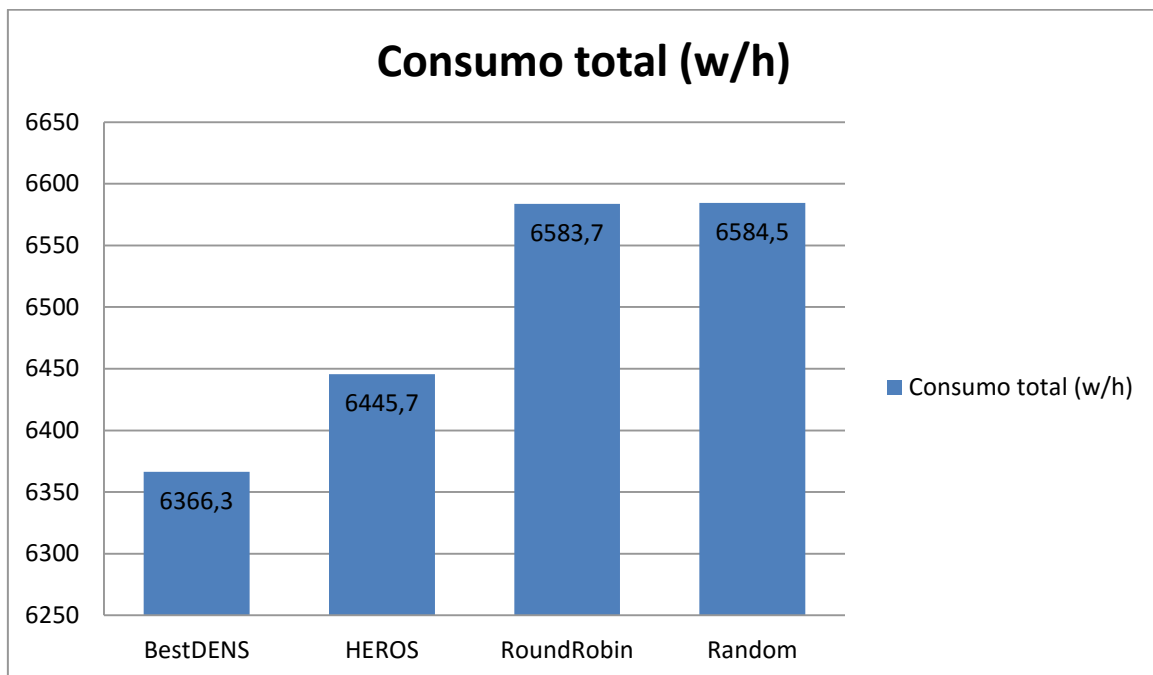
Figura 24: Consumo dos aparelhos no cenário 3 com o Random



5.3.1. SIMULAÇÃO DO CENÁRIO 3

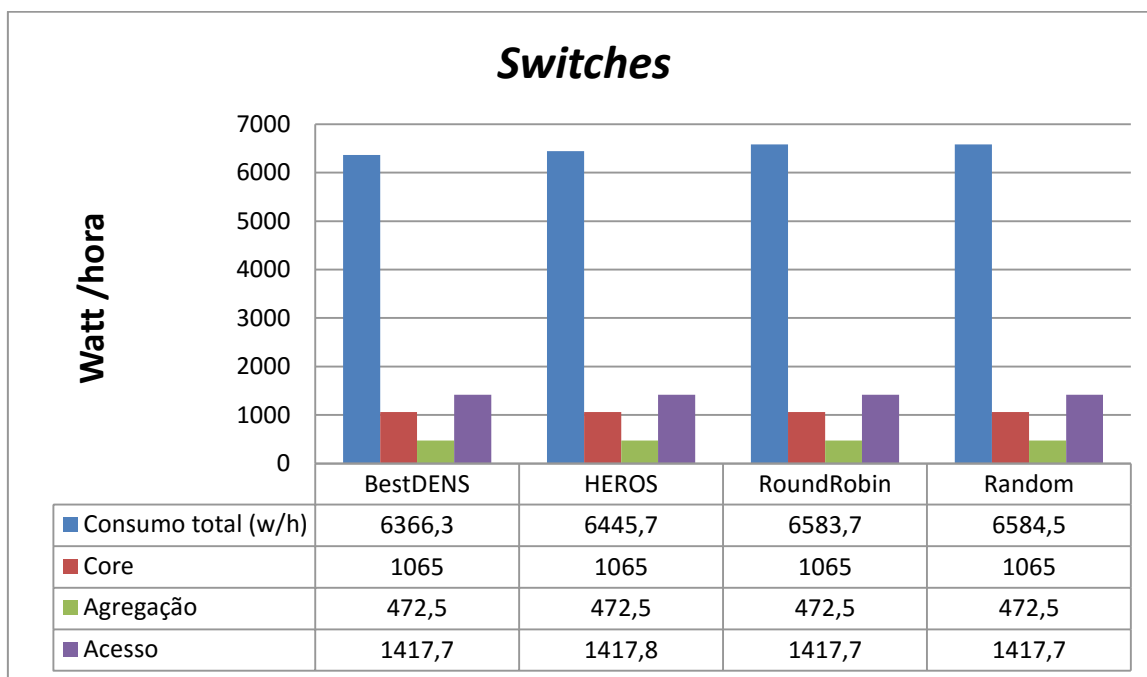
Após a simulação do último cenário com os algoritmos propostos no início deste projeto, obtivemos alguns resultados, que serão exibidos nos gráficos abaixo. Neste cenário, aumentamos o número de servidores e de *switches* de acesso consideravelmente, conseqüentemente uma economia maior de energia quando aplicamos os algoritmos BestDENS e HEROS, vide o gráfico de consumo total (Figura 25), apresentado abaixo:

Figura 25: Consumo total do cenário 3



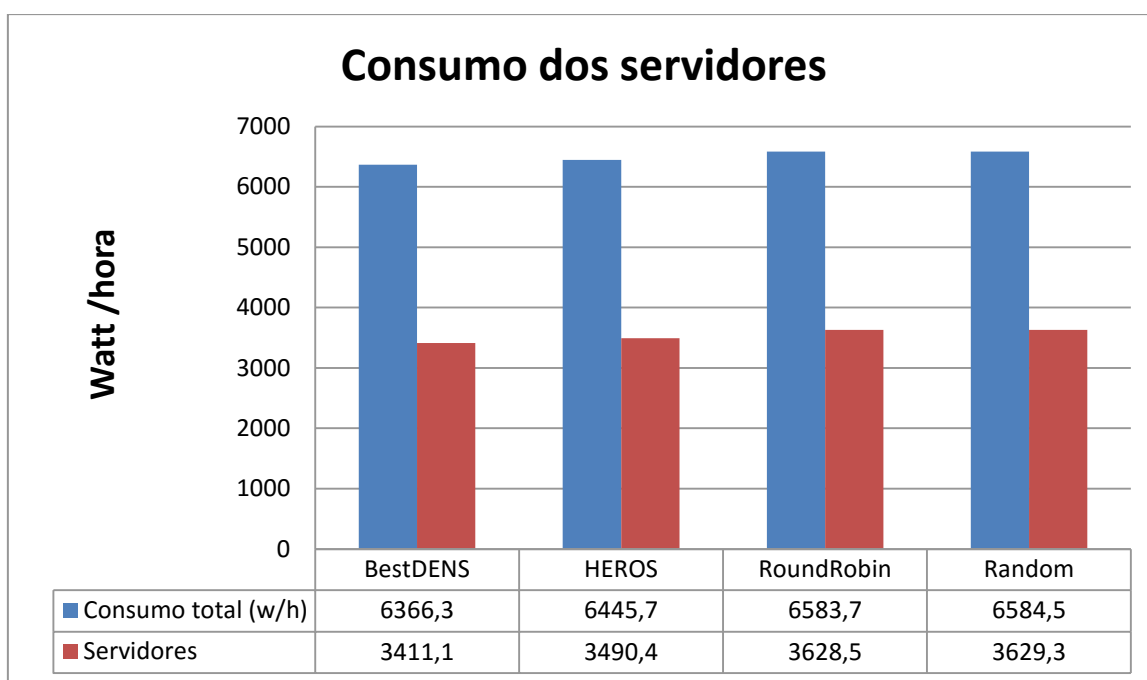
A economia apresentada na simulação do BestDENS chega a 5% comparada ao ponto de partida, a economia do HEROS é de 2%, o que nos mostra que quanto maior a quantidade de servidores o cenário tiver, provavelmente maior será a economia que obteremos utilizando ambos os algoritmos. Porém, o comportamento apresentado nas simulações anteriores, também ocorre nesta, não obtivemos nenhuma vantagem no consumo dos dispositivos de rede quando comparamos os desempenhos dos algoritmos aplicados ao nosso ponto de partida, aliás, quando utilizamos o HEROS obtemos um pequeno aumento no consumo dos *switches* de Acesso, vide a Figura 26.

Figura 26: Consumo dos *switches* no cenário 3



A grande queda no consumo está relacionada ao desempenho dos servidores, ao observarmos a Figura 27 podemos ver o consumo deles. A economia de energia quando comparamos os cenários aplicando os algoritmos BestDENS e Random, é de 6% para o primeiro. Essa economia poderia ser maior, caso fosse aplicado um algoritmo para desligar equipamentos ociosos.

Figura 27: Consumo dos servidores no cenário 3



Na Figura 28 nós temos a quantidade de energia consumida numa escala de 0 a 4 (watt/hora), por cada servidor utilizando o algoritmo BestDENS, além de

apresentar quantas tarefas cada servidor executou. Foram utilizados um pouco mais de 250 servidores, de um ambiente com 1536, o restante estava gastando energia de forma ociosa. O mesmo não se aplica aos outros cenários, a Figura 29 mostra o exemplo da simulação com o HEROS. Aparentemente todos os servidores executaram ao menos uma tarefa na simulação, não tínhamos nenhum computador ocioso como apresentado no caso anterior. O BestDENS detecta *hotspots* de congestionamento na rede do *data center* e ajusta a sua metodologia de trabalho. Para trabalho intensivo com dados, o algoritmo é uma escolha realmente relevante, pois, este tipo de trabalho é mais limitado pela disponibilidade de recursos de comunicação do que pelas capacidades de processamento disponíveis.

Figura 28: Relação energia e tarefas realizadas pelos servidores no algoritmo

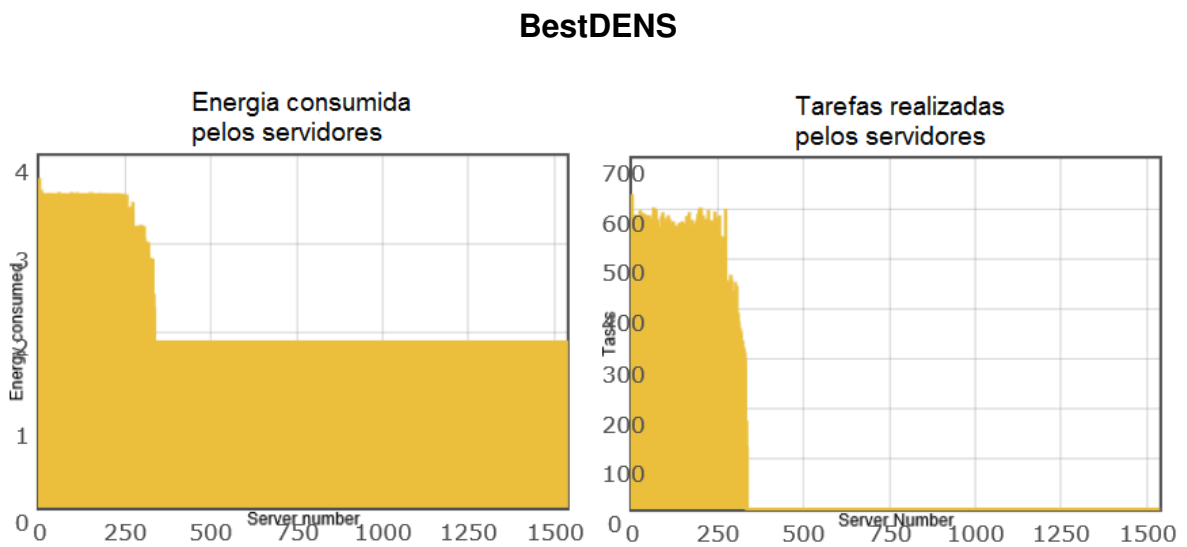
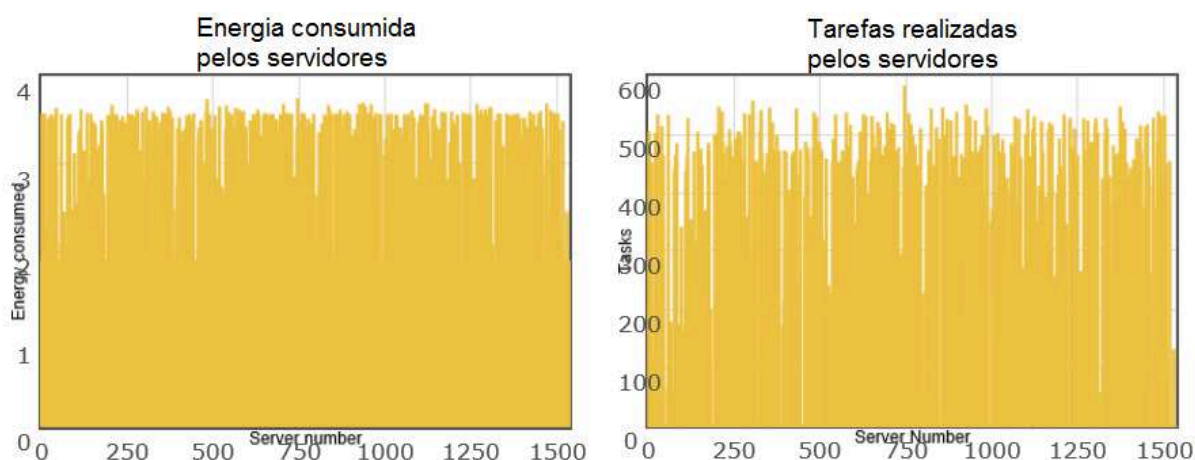


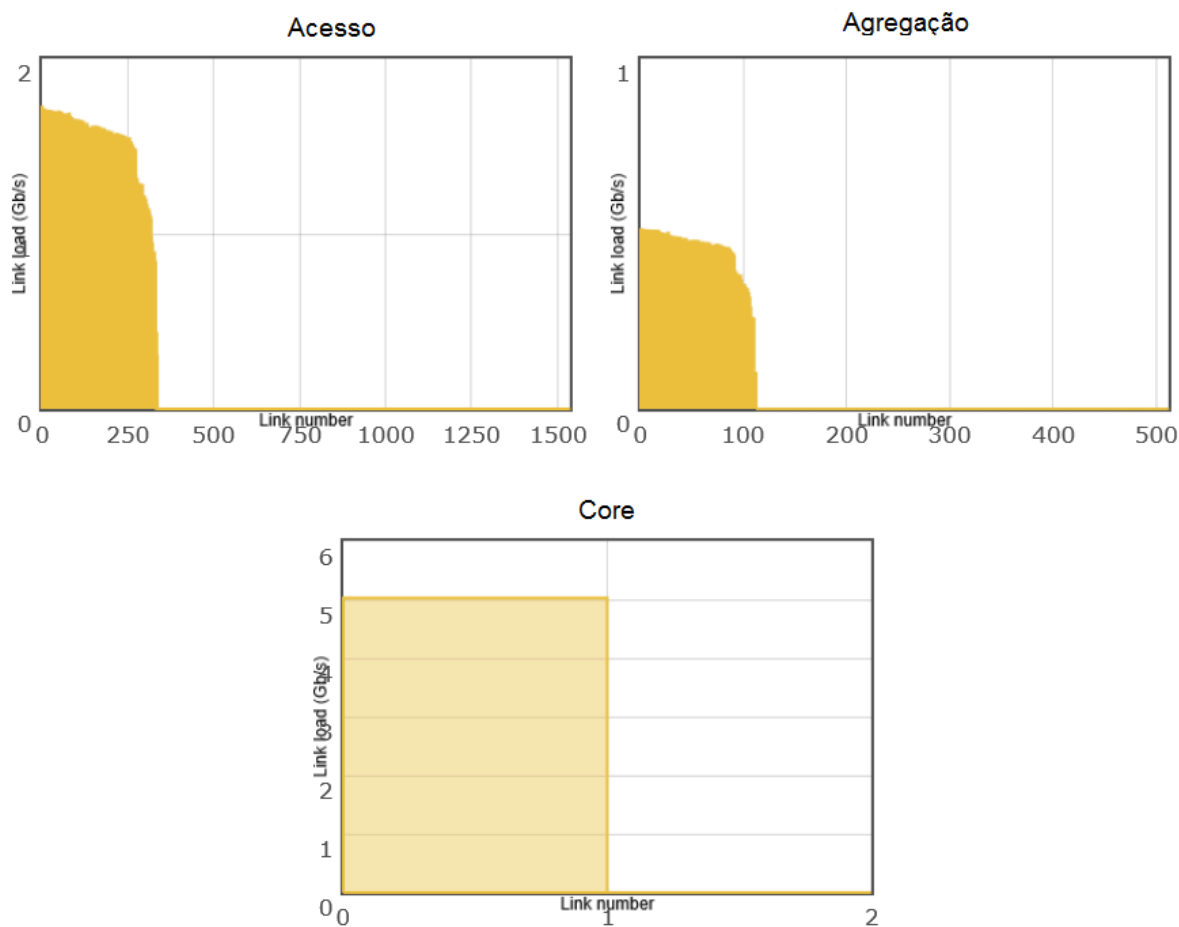
Figura 29: Relação energia e tarefas realizadas pelos servidores no algoritmo

HEROS



O mesmo caso pode ser utilizado para os dispositivos de rede, utilizando um algoritmo que detecta e desligue os equipamentos ociosos, teríamos uma grande economia de energia no cenário em que aplicamos o BestDENS. Além de economizarmos com a energia nos aparelhos, também poderíamos diminuir o consumo dos equipamentos de refrigeração, afinal, teríamos menos equipamentos superaquecendo a sala. As Figuras 30 e 31 exemplificam bem o comportamento dos algoritmos BestDENS e HEROS em relação ao uso dos *switches*, sendo o eixo Y a carga dos links (em Gb/s) que foram utilizados .

Figura 30: Uso dos *switches* no algoritmo BestDENS



Apesar de apresentar bons resultados na economia de energia, BestDENS e HEROS repetem a mesma falha apresentada na simulação anterior, possuem um alto número de falhas em tarefas, chegando a índices de 30% e 22% de falhas respectivamente, como mostrado na Figura 32. Nos cenários em que foram aplicados os algoritmos Round-Robin e Random, ambos obtiveram 0 falhas. Portanto, este pode ser o grande modificador de desempenho dos dois primeiros algoritmos, pois, com menos tarefas, teoricamente há menos carga nos servidores, diminuindo o consumo.

Figura 31: Uso dos switches no algoritmo HEROS

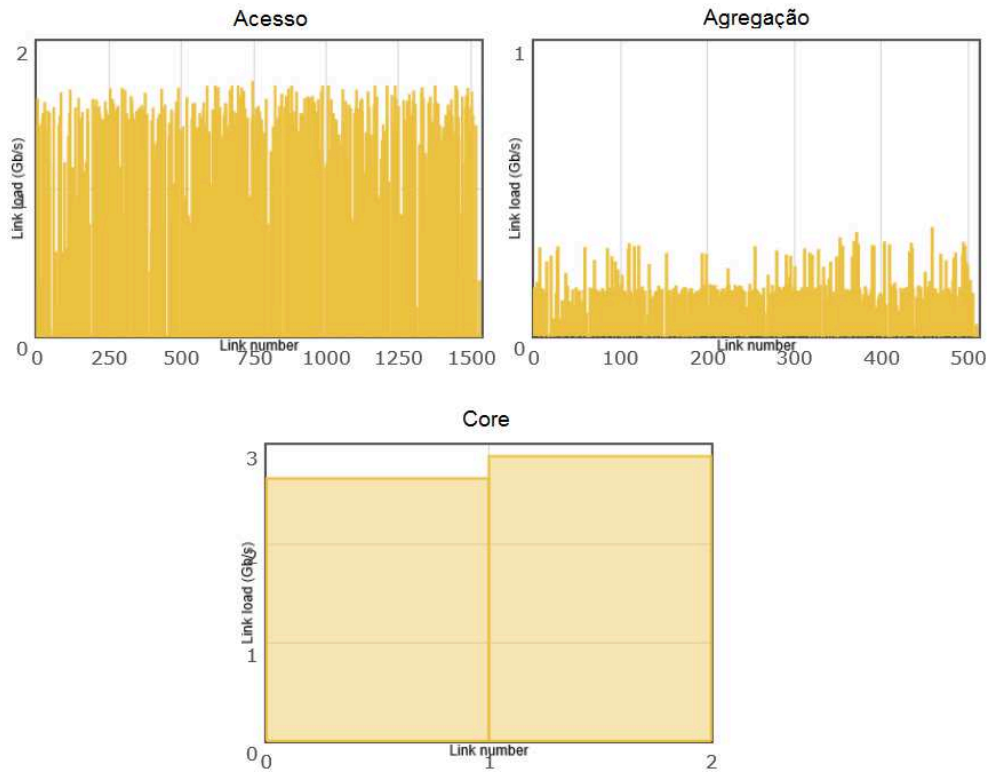
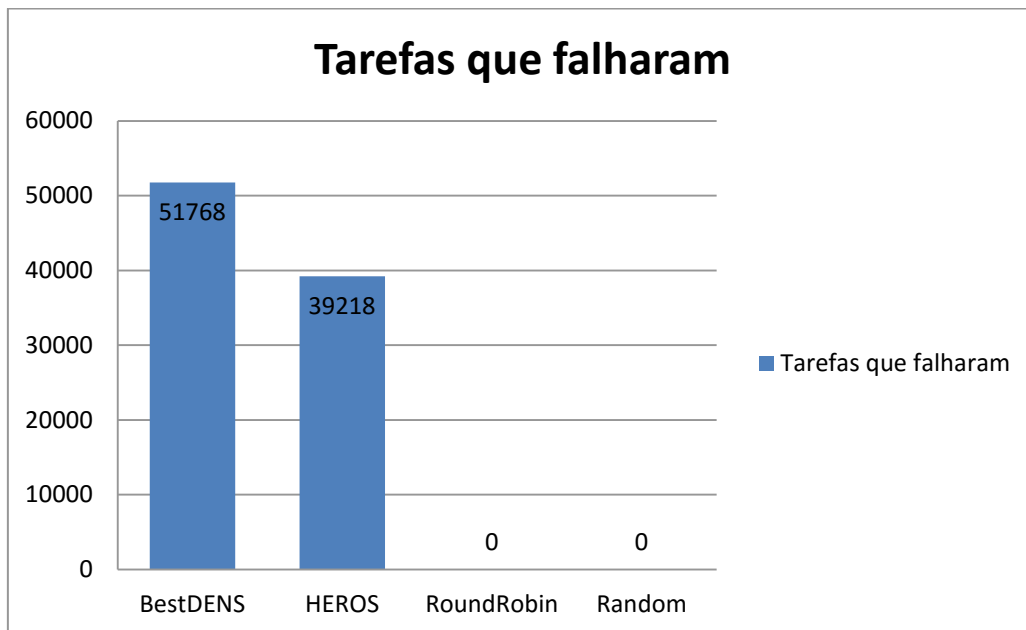


Figura 32: Tarefas que falharam no cenário 3



5.4. CONSIDERAÇÕES FINAIS

A interface *web* do simulador *Green Cloud* se mostrou de fácil execução, porém, apresentou diversas falhas nas execuções feitas durante este projeto. Uma das mais significativas foi o não funcionamento de outras tecnologias que estão disponíveis para serem utilizadas pelo simulador, como: DVFS, *Dynamic Shutdown* e virtualização. Um dos objetivos específicos, o quarto para ser exato, era saber como os algoritmos estudados se comportam quando trabalham com estas técnicas, o que não foi possível ser feito, diante de tal problema.

Por causa do problema com técnicas auxiliares, não é possível fazer uma análise mais detalhada do funcionamento dos algoritmos aplicados. Principalmente do BestDENS e do HEROS, pois, eles atuam diretamente na distribuição de tarefas para determinados servidores, deixando outros dispositivos ociosos, podendo obter um desempenho energético cada vez melhor.

Outro problema encontrado foi a dificuldade de analisar os dados. Após a simulação ser executada é exibidos os dados de consumo total de energia de cada aparelho, os resultados nos mostram os consumo na métrica de watts por hora, mas não é informado se a simulação foi feita executando durante uma hora.

A interface é amigável, o simulador é fácil de se usar, até é um pouco didático, tendo pequenas explicações sobre cada parâmetro, algoritmo e gráfico que é usado, porém, são explicações rasas, falta um local com maior informação, principalmente dos algoritmos. Por exemplo, existem diversos algoritmos chamados *Green* ou com alguma parte do seu nome de *Green*, mas qual é o utilizado na *interface*? Poderia haver um *link* com algum artigo explicando detalhadamente o

algoritmo, não precisa estar necessariamente na descrição. Utilizando esta interface também não conseguimos colocar novos algoritmos para simulação, o que nos deixa com poucas opções, 6 no total.

Há poucas opções de customização do ambiente, basicamente só podemos mexer na topologia de rede do *data center*. Seria interessante podermos escolher qual o modelo de *switch* utilizar, as características de cada servidor, o tipo de cabeamento utilizado para a transmissão de dados, os equipamentos de resfriamento, além do tipo da topologia. Com esta interface só podemos simular cenário de três camadas, quando tentei simular um cenário de duas camadas, colocando zero *switches* no *core* o simulador acusou um erro.

Também não foi possível realizar simulações com muito servidores, na primeira vez que criei os cenários, haviam ambientes com até 170 mil servidores, porém, ao simularmos um cenário com apenas 10 mil servidores, a simulação já falhava, por dois motivos:

1. Para criar cerca de 40 mil servidores, levava aproximadamente uma hora;
2. Quando os servidores eram criados, logo em seguida as configurações das tarefas eram carregadas, porém, a simulação travava nesta parte e abortava.

Apesar das falhas apontadas, a interface *web* do *Green Cloud* se mostrou um simulador útil para pequenos cenários. Pode ser utilizado para mostrar o desempenho de algoritmos de escalonamento de tarefas em ambientes de três camadas e como eles podem influenciar no consumo energético dos servidores.

Esta interface *web* é “nova”, como dito no capítulo 4, foi lançada em 2016, é normal ter erros e limitações, no futuro provavelmente estes bugs serão corrigidos e novas ferramentas serão adicionadas, o que a deixaria bem útil para ser utilizada no ensino de aluno, simular qual algoritmo e tecnologia utilizar nos servidores próprios, entre outras finalidades.

6. CONCLUSÃO

Neste trabalho foram realizados três testes em cenários fictícios, no primeiro cenário possuímos poucos servidores, no segundo temos algumas centenas e no terceiro alguns milhares. Todos eles foram simulados utilizando a *interface web* do simulador *Green Cloud*, utilizando uma arquitetura de três camadas. Todas as simulações foram feitas com o intuito de medir o consumo de energia do *data center* em determinado cenário.

Com estas medições, podemos perceber que o comportamento dos algoritmos estudados é influenciado pela quantidade de servidores no cenário. No primeiro ambiente, todos os algoritmos que foram aplicados não apresentaram grande eficiência, sendo o Round-Robin o que consegue uma pequena redução.

No segundo cenário, tanto BestDENS quanto HEROS, conseguem uma redução significativa no consumo de energia, porém, quando analisamos as tarefas executadas, vemos que ambos possuem um alto índice de falhas, influenciando diretamente no custo/benefício da aplicação. Voltamos a olhar o algoritmo Round-Robin, ele consegue uma pequena economia e além disso, executa todas as tarefas enviadas aos servidores.

O mesmo acontece no terceiro cenário, torna-se evidente que a quantidade de servidores presente no ambiente influencia tanto BestDENS quanto HEROS, tanto é que seus desempenhos em questões energéticas, comparado aos outros dois, são melhores. Porém, ao analisarmos a quantidade de falhas, os dois voltam a

apresentar diversas. Como nos dois ambientes anteriores, o que apresenta melhor resultado em relação energia x tarefas executadas, é o Round-Robin.

Percebeu-se que o algoritmo BestDENS poderia ter um desempenho energético ainda melhor, caso fosse utilizado alguma tecnologia auxiliar que desligasse os aparelhos ociosos. Além disso, por os ambientes possuírem as mesmas configurações para todos os servidores e tarefas, fica evidente que o melhor algoritmo sempre, para o balanceamento de carga e conseqüentemente a economia de energia, seria o Round-Robin. Caso, tivéssemos servidores com configurações diferentes entre si, ou tarefas com tamanhos variados, provavelmente o BestDENS seria o melhor algoritmo para ser escolhido, aplicando outras técnicas junto a ele, como *Dynamic shutdown*, DVFS ou virtualização. Também foram notadas diversas falhas apresentadas pelo simulador utilizado, visto o seu recente lançamento.

6.1. TRABALHOS FUTUROS

Nesta seção, serão apresentados sugestões para trabalhos futuros:

- Implementação e análise dos algoritmos utilizados no trabalho em outros tipos de arquitetura;
- Utilização de tecnologias adicionais como DVFS e virtualização nos cenários, e analisar o comportamento dos algoritmos;

- Parceria com a universidade desenvolvedora da interface web do simulador Green Cloud, a fim de arrumar as falhas encontradas e implementar as sugestões encontradas na seção 5.4 deste trabalho;
- Implementação e análise dos algoritmos e cenários estudados em outros simuladores, comparando a precisão dos resultados de ambos;
- Desenvolvimento de um algoritmo de escalonamento de tarefas próprio;
- Implementação dos algoritmos em ambientes reais, para analisar o real impacto.
- Poderia ser estendido a área de estudo, para englobar outras áreas como o consumo no resfriamento dos ambiente.

REFERÊNCIAS

2016 TOP MARKETS REPORT CLOUD COMPUTING COUNTRY CASE STUDY, 2016, Aaa. 2016 ITA Cloud Computing Top Markets Report. 2016.

ARPACI-DUSSEAU, Remzi H.; ARPACI-DUSSEAU, Andrea C. **Operating systems: Three easy pieces**. Arpaci-Dusseau, 2014.

BAKER, Mark; BUYYA, Rajkumar; HYDE, Dan. **Cluster computing: A high-performance contender**. arXiv preprint cs/0009020, 2000.

BALIGA, J.; AYRE, R.W.A.; HINTON, K.; TUCKER, R.S.; **Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport**. Proceedings of the IEEE , vol.99, no.1, pp.149-167, Jan. 2011

BAWDEN, Tom. **Global warming: Data centres to consume three times as much energy in next decade, experts warn. 2016**. Disponível em: <<http://www.independent.co.uk/environment/global-warming-data-centres-to-consume-three-times-as-much-energy-in-next-decade-experts-warn-a6830086.html>>. Acesso em: 25 mar. 2016.

BELOGLAZOV, Anton; ABAWAJY, Jemal; BUYYA, Rajkumar. **Energy-aware resource allocation heuristics for efficient management of data centers for**

cloud computing. Future generation computer systems, v. 28, n. 5, p. 755-768, 2012.

BIGNATTO JUNIOR, Pedro Wilson. **Alocação de recursos ciente de energia para entrega de desktops virtuais em um ambiente de computação em nuvem.** 2016. 85 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de São Carlos, São Carlos, 2016.

BUYYA, Rajkumar; BROBERG, James; GOSCINSKI, Andrzej M. (Ed.). **Cloud computing: principles and paradigms.** John Wiley & Sons, 2010.

BUYYA, Rajkumar; YEO, Chee Shin; VENUGOPAL, Srikumar. **Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities.** In: IEEE International Conference on High Performance Computing and Communications, 10., Proceedings...”, 2008

CAPGEMINI. **Business Cloud in Brazil: Research Report 2014: At the tipping point of accelerated adoption.** Disponível em: <https://www.br.capgemini.com/resource-file-access/resource/pdf/business_cloud_in_brazil_20140827_v15.pdf>. Acesso em: 13 set. 2016.

CATABRIGA, Lucia. **Arquiteturas MIMD.** Disponível em: <<http://www.inf.ufes.br/~luciac/fem/MIMD.pdf>>. Acesso em: 03 fev. 2016.

CISCO. **Data Center Multi-Tier Model Design**. Disponível em: <http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCInfra_2.html>. Acesso em: 27 set. 2016.

CUPERTINO, Calif. **Symantec Study Reveals Green IT Now Essential IT Practice**. 2009.

DE ROSE, C.; NAVAU, P. **Fundamentos de processamento de alto desempenho**. Anais: 2a Escola Regional de Alto Desempenho, p. 3-29, 2002.

ENERGY STAR PROGRAM. **Top 12 Ways to Decrease the Energy Consumption of Your Data Center**. 2012. Disponível em : <https://www.energystar.gov/ia/products/power_mgt/downloads/DataCenter-Top12-Brochure-Final.pdf>. Acesso em: 20 out. 2015

EVANS, Rich; GAO, Jim. **DeepMind AI reduces energy used for cooling Google data centers by 40%**. 2016. Disponível em: <<https://green.googleblog.com/2016/07/deepmind-ai-reduces-energy-used-for.html>>. Acesso em: 01 ago. 2016.

FARIA, Izaias de. **Uma abordagem de seleção de recursos consciente de consumo de energia baseada em topologia de rede, tamanho de arquivos e**

potência de equipamentos. 2015. 102 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2015.

FOSTER, Ian et al. **Cloud computing and grid computing 360-degree compared.** In: Grid Computing Environments Workshop, 2008. GCE'08. Ieee, 2008. p. 1-10.

FOSTER, Ian; KESSELMAN, Carl. **The history of the grid. computing**, v. 20, n. 21, p. 22, 2010.

FROST; SULLIVAN. **Analysis of the Brazilian Cloud Computing Market**, 2014.

GARTNER. **Gartner Identifies the Top 10 Strategic Technology Trends for 2015.** 2014. Disponível em: <<http://www.gartner.com/newsroom/id/2867917>>. Acesso em: 13 set. 2016.

GUZEK, Mateusz; KLIAZOVICH, Dzmitry; BOUVRY, Pascal. **HEROS: Energy-efficient load balancing for heterogeneous data centers.** 2015 IEEE 8th International Conference on Cloud Computing. IEEE, 2015.

KLIAZOVICH, Dzmitry, et al. **GreenCloud: a packet-level simulator of energy-aware cloud computing data centers.** Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE. IEEE, 2010.

KLIAZOVICH, Dzmitry; BOUVRY, Pascal; KHAN, Samee Ullah. **DENS: data center energy-efficient network-aware scheduling.** Cluster computing 16.1 (2013): 65-75.

LE SUEUR, Etienne; HEISER, Gernot. **Dynamic voltage and frequency scaling: The laws of diminishing returns.** In Proceedings of the 2010 international conference on Power aware computing and systems 2010 Oct 3 (pp. 1-8).

LIU, Jie et al. **Challenges towards elastic power management in internet data centers.** In: Distributed Computing Systems Workshops, 2009. ICDCS Workshops' 09. 29th IEEE International Conference on. IEEE, 2009. p. 65-72.

LIU, Liang et al. **GreenCloud: a new architecture for green data center.** In: Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session. ACM, 2009. p. 29-38.

MEISNER, David; GOLD, Brian T. and WENISCH, Thomas F. "**PowerNap: eliminating server idle power.**" ACM Sigplan Notices. Vol. 44. No. 3. ACM, 2009.

MELL, Peter; GRANCE, Tim. **The NIST definition of cloud computing.** National Institute of Standards and Technology, v. 53, n. 6, p. 50, 2009.

MONTEIRO, Rogério Castellar; DANTAS, Mario Antonio Ribeiro. **Green Cloud Computing: A Study Case of Resources and Network Energy Consumption**. 2012.

NURMI, Daniel et al. **The eucalyptus open-source cloud-computing system**. In: Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on. IEEE, 2009. p. 124-131.

PIMENTEL, João Alfredo. **Nuvem reduz consumo de energia. Como usar o conceito em tempos de crise energética?** Disponível em: <<http://computerworld.com.br/nuvem-reduz-consumo-de-energia-como-usar-o-conceito-em-tempos-de-crise-energetica>>. Acesso em: 13 set. 2016.

TANENBAUM, Andrew S.; VAN STEEN, Maarten. **Distributed Systems: Principles and Paradigms**. 2. ed. Amsterdam: Pearson, 2007.

TANENBAUM, Andrew S.. **Sistemas operacionais modernos**. 3. ed. São Paulo: Pearson Prentice Hall, 2009.

VAQUERO, Luis M. et al. **A break in the clouds: towards a cloud definition**. ACM SIGCOMM Computer Communication Review, v. 39, n. 1, p. 50-55, 2008.

APÊNDICE

DEMÉTER – UM ESTUDO DE ALGORITMOS DE ESCALONAMENTO DE TAREFAS APLICADOS EM AMBIENTES DE NUVEM DE TRÊS CAMADAS

Guilherme Schmitt¹

¹Departamento de Informática – Universidade Federal de Santa Catarina (UFSC)

guischmitt@gmail.com

Abstract. *In this work, a study of tasks scheduling algorithms with emphasis on energy consumption, applied to the three layer cloud architectures os carried out. In this way, scenarios were proposed and simulated in order to verify the effectiveness of the algorithms and to compare them, in order to better understand their performance in the proposed scenarios.*

Resumo. *Neste trabalho é realizado um estudo de algoritmos de escalonamento de tarefas, com ênfase em consumo de energia, aplicados em arquiteturas de nuvem de três camadas. Cenários foram propostos e simulados de forma a verificar a eficácia dos algoritmos e comparando-os, para melhor entender seus desempenhos nos cenários propostos.*

1. Introdução

O Brasil é o maior utilizador dos serviços de nuvem na América Latina, segundo um estudo de caso do ITA (2016). De acordo com Frost e Sullivan (2014) o mercado de computação em nuvem no Brasil movimentou US\$ 217 milhões em 2012, e a previsão é que movimente US\$ 1.1 bilhões até 2017. Com a alta demanda de serviço em nuvem e sua constante expansão, é necessário cada vez mais recursos para suprir a demanda de serviços, ou seja, são necessários mais servidores e data centers para manter a qualidade no fluxo de informações e dados, aumentando drasticamente o consumo de energia.

Segundo um artigo publicado por Bawden (2016) no Independent, os data centers consomem cerca de 3% do fornecimento mundial de energia elétrica e são responsáveis por cerca de 2% da emissão de gases poluentes do efeito estufa. Para efeito de comparação, os 416,2 terawatts/hora de energia consumida pelos data centers do mundo no último ano foi maior que o consumo total do Reino Unido no respectivo período, algo em torno de 300 terawatts/hora. Os gases poluentes equivalem ao que é emitido por toda a indústria aérea.

Segundo um estudo de Meisner, Gold e Wenisch (2009), a utilização média dos servidores em um data center é entre 20-30%. O desperdício de energia em equipamentos ociosos é agravado pelas perdas na infra-estrutura de fornecimento de energia e resfriamento, que podem aumentar o consumo de energia entre 50-100%. Em implementações típicas, os servidores ociosos são responsáveis por cerca de 60% do consumo de energia do data center.

Visto os problemas apresentados pelo alto consumo de energia de redes nuvens e principalmente por aparelhos desperdiçando energia em modo ocioso, a proposta deste trabalho é entender e analisar o funcionamento de algoritmos de escalonamento de tarefas em um ambiente de nuvem computacional de três camadas sob a perspectiva do consumo energético. Por meio da análise de métricas de consumo de energia em servidores e equipamentos de transmissão, coletadas via experimentos realizados em um simulador de nuvens computacionais, a influência de quatro diferentes algoritmos de escalonamento de tarefas no consumo energético em três ambientes com escala de complexidade diferentes é avaliada.

2. Fundamentação Teórica

2.1. Random

Como o próprio nome sugere, ele seleciona um nodo aleatoriamente para executar determinada tarefa.

2.2. Round-Robin

O algoritmo Round-Robin executa um processo em uma determinada fração de tempo, após esse tempo acabar ele parte para o próximo processo na fila de execução. Este comportamento é feito repetidamente até os processos que estão na fila serem concluídos (ARPACI-DUSSEAU, ARPACI-DUSSEAU; 2014).

2.3 Best-DENS

O Best-DENS minimiza o consumo total de energia de um data center selecionando os recursos de computação mais adequados para a execução de tarefas, baseando-se no nível de carga e no potencial de comunicação dos componentes do data center. O pseudo-código do algoritmo é apresentado abaixo (KLIAZOVICH, BOUVRY, KHAN; 2013):

DENS Algorithm

Initialization
 set weighted coefficient $\alpha = 0.7$, $\beta = 0.2$, $\gamma = 0.1$
 set proportional coefficient $\varphi = 2$
 get server load l
 get queue size at access and aggregate switches q

Server selection
FOR all servers **DO**
 compute server load $L_s(l)$, rack load $L_r(l)$, and
 module load $L_m(l)$
 compute communications potentials of rack $Q_r(q)$
 and module $Q_m(q)$
 compute metric factors related to servers $f_s(l, q)$,
 racks $f_r(l, q)$, and modules $f_m(l)$
 compute DENS metric as a weighted sum of
 $f_s(l, q)$, $f_r(l, q)$, and $f_m(l)$
ENDFOR
 Select server with highest DENS metric

2.4 HEROS

Heterogeneous Energy-efficient Resource allocation Optimizing Scheduler (HEROS) é um algoritmo para o balanceamento de cargas em alocações de recursos, focando na eficiência energética em sistemas heterogêneos. HEROS aloca tarefas para os servidores com as pontuações máximas, esta pontuação é calculada por uma função de decisão, compostas por uma função de seleção de servidor e uma função de potencial de comunicação (GUZEK, KLIAZOVICH, BOUVRY; 2015).

3. Estado da arte

Na busca do estado da arte, foram encontrados alguns trabalhos que abordam o mesmo problema, a diminuição do alto consumo de energia em *data centers*:

Google utilizou uma machine learning, a DeepMind, em seu próprio data center, conseguindo reduzir a quantidade de energia utilizada na refrigeração em até 40%. Para treinar a máquina, utilizaram dados históricos que já haviam sido recolhidos por milhares de sensores dentro do data center. Também foram treinados dois conjuntos de redes neurais para prever a temperatura e pressão do data center na hora seguinte. O objetivo da previsão é simular as ações recomendadas a partir do modelo PUE (Power Usage Effectiveness), para garantir que não vão além de quaisquer restrições operacionais. A Figura 1, mostra o desempenho da máquina em um cenário típico de testes (EVANS, GAO; 2016):

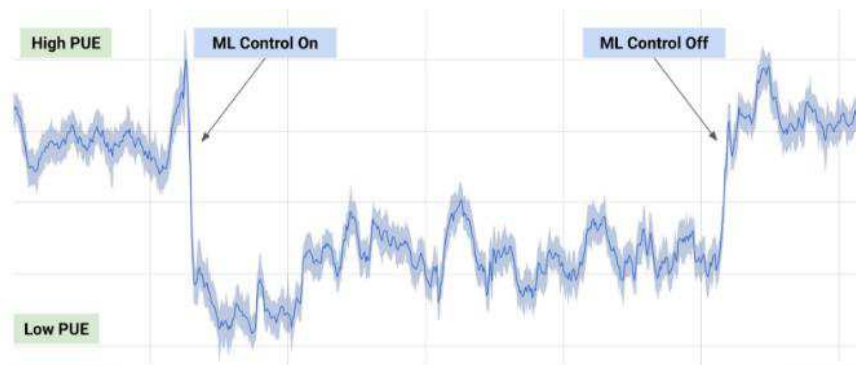


Figura 1 – Desempenho com o DeepMind ligado e desligado

Faria (2015) propõe uma abordagem genérica de alocação de recursos energeticamente eficiente, que tende a melhorar o desempenho de ambientes de alto desempenho heterogêneos, a partir da seleção de recursos com menores custos na transferência de dados, além dos estados dos nodos e as respectivas eficiências energéticas, para isso foi desenvolvido um algoritmos de escalonamento de recursos, o ENA (Energy and Network Aware Algorithm). A proposta conseguiu reduzir consideravelmente o consumo de energia em comparação com outras abordagens existentes.

Monteiro e Dantas (2012) propõem um cenário com um data center de pequeno porte, em que é considerado o consumo relativo à transmissão e os custos relativos dos dispositivos de redes, indicando a melhor configuração para tal situação. Após os experimentos, os resultados indicaram que a energia necessária para o transporte de dados influencia no consumo de energia, além disso existe uma grande vantagem quando utilizamos diversos usuários simultaneamente. Uma economia de cerca de 37% da energia consumida no total foi conseguida após as simulações.

4. Escolha dos cenários e do simulador

Já que o objetivo da proposta deste trabalho é fazer simulações em ambientes de três camadas, para sabermos qual algoritmo é melhor para determinado cenário, na sequência são apresentados o simulador escolhido e os cenários propostos, assim como as justificativas para os mesmos.

4.1 Simulador Green Cloud

Para o desenvolvimento dos experimentos deste trabalho foi escolhida a abordagem de simulação, devido à complexidade/custo de um ambiente real de três camadas. Foram pesquisados e estudados diversos simuladores (CloudSim, MDCSim) e optou-se pelo Green Cloud. A escolha desse simulador se baseou em sua característica de possuir facilidade de alteração dos valores dos enlaces, a quantidade de switches em cada camada, a quantidade de hosts e o tamanho das tarefas. Um outro fator diferencial foi a possibilidade de utilizarmos os algoritmos a serem investigados. O simulador permite simular um ambiente composto por milhares de nodos e, também, contabiliza os erros de

conexão ou perdas relacionadas ao congestionamento em switches de rede. Quanto aos modelos de energia são implementados para cada componente do data center.

O Green Cloud é uma extensão do simulador Ns2, ele extrai, agrega e disponibiliza informações sobre a energia consumida pelos elementos de computação e comunicação do data center de uma forma sem precedentes. Neste simulador, possuímos três componentes de consumo de energia: os componentes relacionados a computação e comunicação, além dos componentes energéticos relacionados à infra-estrutura física de um data center (KLIAZOVICH et. al., 2010).

A eficiência do data center pode ser definida em termos do desempenho fornecido por watt, este desempenho pode ser quantificado utilizando algumas métricas, como as duas a seguir: Power Usage Effectiveness (PUE) e Data Center Infrastructure Efficiency (DCiE) (KLIAZOVICH et. al., 2010).

O modelo de energia seguido pelos componentes do servidor depende do estado do servidor e da utilização da CPU. No simulador o consumo de energia aumenta linearmente com o nível de carga da CPU e da frequência operacional. . A qualidade do sinal de transmissão em um determinado cabo determina um *tradeoff* entre a taxa de transmissão e a distância do *link*, que são os fatores que definem o custo e o consumo de energia dos transmissores.

4.2 Cenários

Após o estudo do algoritmos, foram feitas as análises dos ambientes a serem simulados, após um tempo de estudo, foram propostos três ambientes, suas configurações de tarefas e *links* podem ser vistas nos quadros 1, 2:

Configuração das tarefas	
Tamanho da tarefa (MIPS):	600000
Memória alocada durante a execução da tarefa (<i>task memory</i>):	3000000
Armazenamento disponível para execução da tarefa (<i>task storage</i>):	600000
Bytes a serem enviados para o servidor, afim de executar a tarefa (<i>task description size</i>):	10000
Deadline da tarefa (s):	7
Bytes enviados quando a tarefa é realizada com sucesso (<i>task output</i>):	350000

Quadro 1: Configuração das tarefas dos ambientes

Configuração dos <i>links</i>		
	Cenários 1 e 2	Cenário 3
Core para Agregação	10 Gb/s	100 Gb/s
Agregação para Acesso	1 Gb/s	10 Gb/s
Acesso para os <i>Hosts</i>	1 Gb/s	1 Gb/s

Quadro 2: Configuração dos *links* dos ambientes

O primeiro cenário possui poucos recursos, visando ter uma melhor análise do desempenho dos algoritmos BestDENS e HEROS, que tendem a sobrecarregar servidores e deixar outros ociosos. No segundo ambiente aumentamos a complexidade dos componentes, para conseguirmos analisar o comportamento dos dois algoritmos citados anteriormente, principalmente se conseguem identificar aparelhos ociosos, qual a economia adquirida e se conseguem garantir a qualidade de serviço. O terceiro ambiente é muito parecido com o segundo, porém, com links mais poderosos, para analisarmos se as configurações dos links influenciam no desempenho dos algoritmos propostos. As configurações da topologia de rede dos cenários, podem ser vistas no quadro 3:

	Cenário 1	Cenário 2	Cenário 3
Core Switches	2	5	2
Switches de Agregação	4	10	4
Switches de Acesso	2	30	256
Servidores por rack	4	6	3
Total de servidores	16	900	1536
Número de Tarefas executadas	4283	100654	174449

Quadro 3: Configuração dos cenários

5. Avaliação e resultados

Utilizaremos o algoritmo Random para obtermos resultados que serão utilizadas nas comparações dos algoritmos na próxima seção, vamos utilizar as seguintes métricas:

- Consumo Total;
- Consumo dos switches do core;
- Consumo dos switches de Agregação;
- Consumo dos switches de Acesso;
- Consumo dos servidores;
- Entre outras;

5.1 Cenário 1

A Figura 2 exibe os consumos energéticos do cenário 1, os algoritmos BestDENS e HEROS possuem um desempenho energético pior que o apresentado pelo Random, além disso, o ganho com o Round-Robin é mínimo: 0,1 watts/hora. A Figura 3 nos mostra o desempenho nos dispositivos de rede, e podemos reparar que todos os algoritmos possuem

desempenho semelhante, isso se deve ao fato de nenhum deles atuar diretamente no consumo da rede.

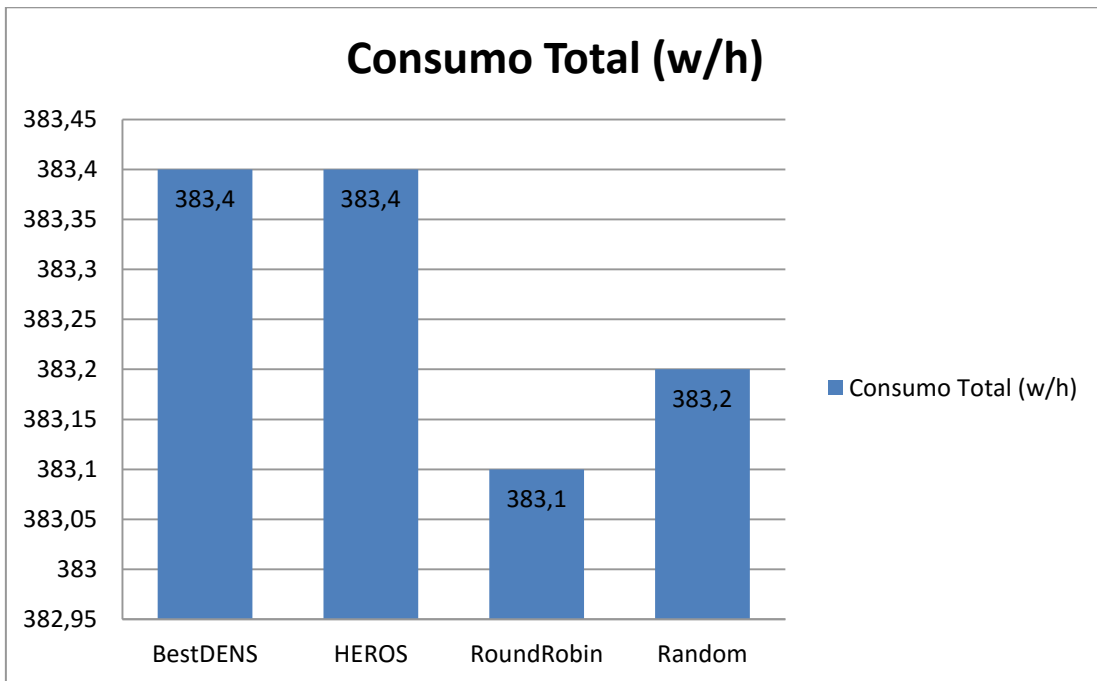


Figura 2: Consumo total do cenário 1.

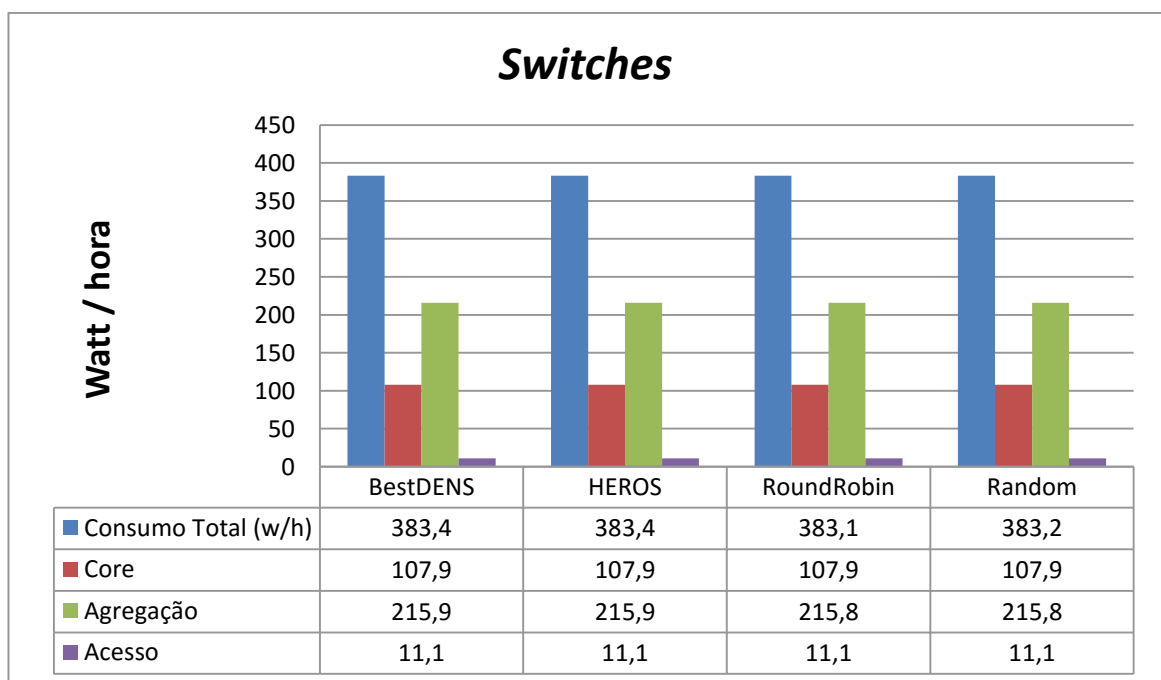


Figura 3: Consumo dos switches do cenário 1

5.2 Cenário 2

A Figura 4 exibe os consumos energéticos do cenário 4. Se analisarmos apenas o consumo total, o melhor escalonador a ser utilizado para o ambiente 2 é o BestDENS, tendo uma economia de 99 watts por hora, comparado ao ponto de partida (com algoritmo Random).

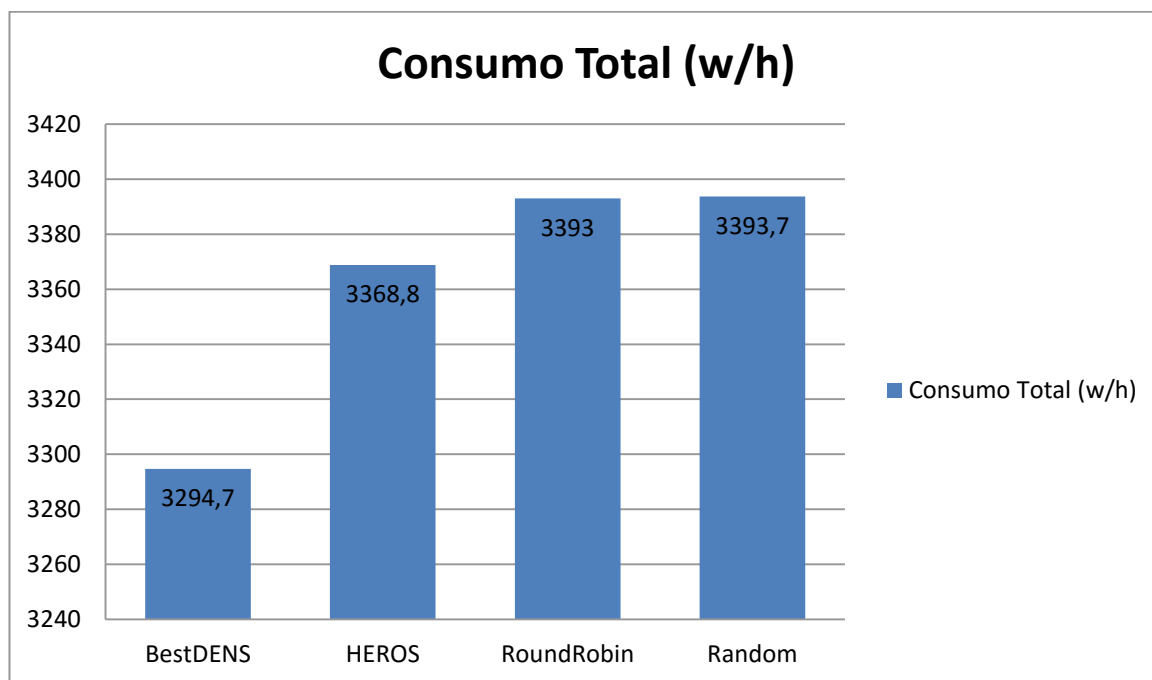


Figura 4:

Quando analisamos a quantidade de tarefas que falharam durante a simulação (Figura 5), notamos que tanto o BestDENS quanto o HEROS, os dois com melhores desempenhos até então, possuem um alto nível de falhas, cerca de 24% e 10% das tarefas, respectivamente. Podemos indicar o grande desempenho energético dos algoritmos HEROS e BestDENS ao grande número de tarefas que falharam, pois, elas não chegam a serem processadas pelos servidores, não aumentando a carga dos servidores e consequentemente a energia consumida por eles.

O gráfico de carga por computador pode ser visto na Figura 6, ao analisarmos a figura, podemos notar que os primeiros computadores do cenário simulado com o BestDENS, utilizam quase todo o fator de carga de seus processadores. Isso se deve pois, o BestDENS consegue uma melhor consolidação da carga de trabalho, impedindo o sobrecarregamento de servidores e switches, com isso possuímos uma melhor eficiência energética. A carga média de um servidor é de aproximadamente 0,9 (em uma normalização de 0 a 1), isso garante que nenhum atraso adicional seja colocado nas comunicações feitas pelos servidores, não causando congestionamentos.

Nós temos uma distribuição mais igual quando olhamos os gráficos dos algoritmos Random e Round-Robin, sendo que o segundo distribui de forma igual entre todos os computadores. Round-Robin e Random são os escalonadores mais ineficientes em termos de energia, quando conseguimos colocar os servidores e switches ociosos em modo sleep ou são desligados, pois, eles não permitem que nenhum dos componentes seja desligado durante a execução das tarefas.

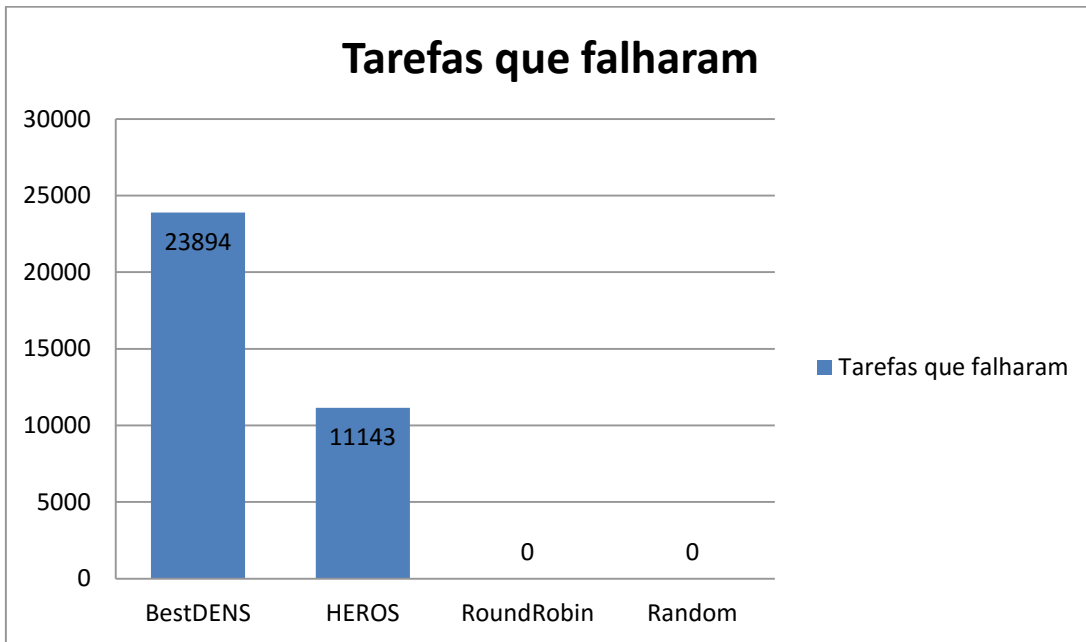


Figura 5: Quantidade de tarefas que falharam por simulação no cenário 2

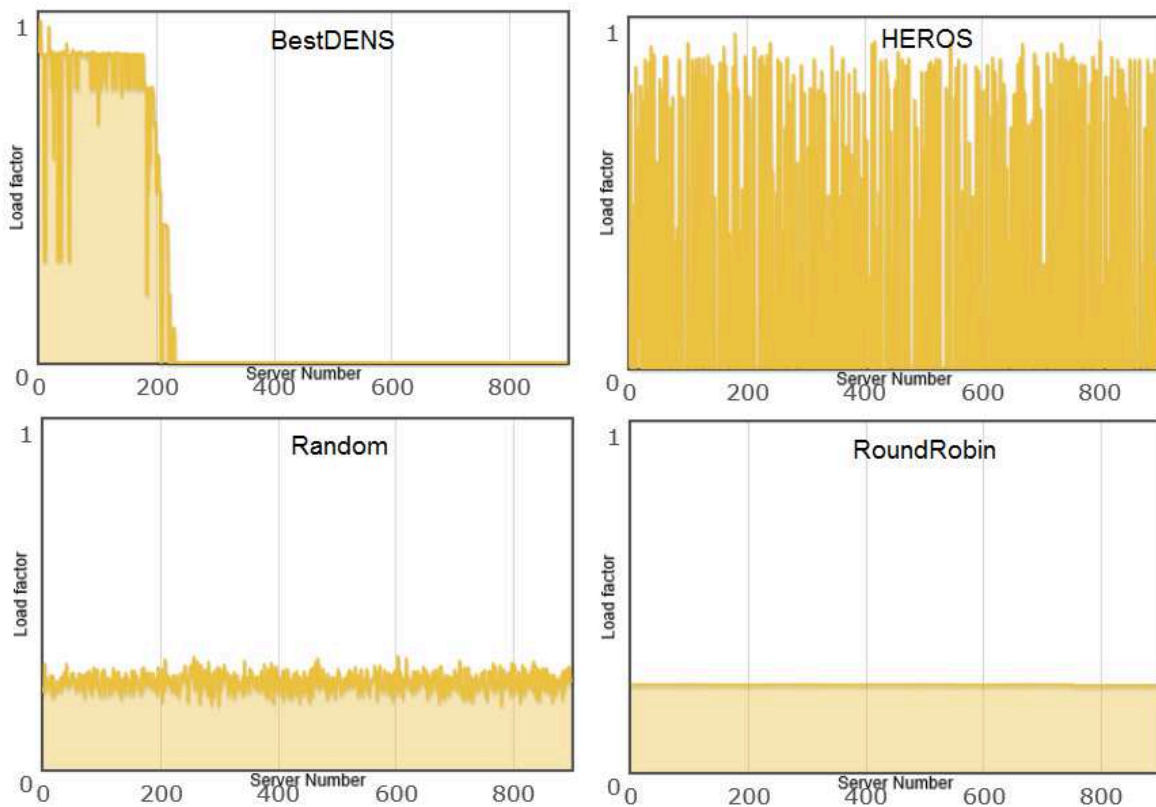


Figura 6: Gráficos de carga por computador no cenário 2

5.3 Cenário 3

A Figura 7 apresenta o consumo total dos aparelhos nas simulações. A economia apresentada na simulação do BestDENS chega a 5% comparada ao ponto de partida, a economia do HEROS é de 2%, o que nos mostra que quanto maior a quantidade de servidores o cenário tiver, provavelmente maior será a economia que obteremos utilizando ambos os algoritmos. Porém, esse ganho também se deve ao fato de obtermos um grande índice de tarefas que falharam em sua execução.

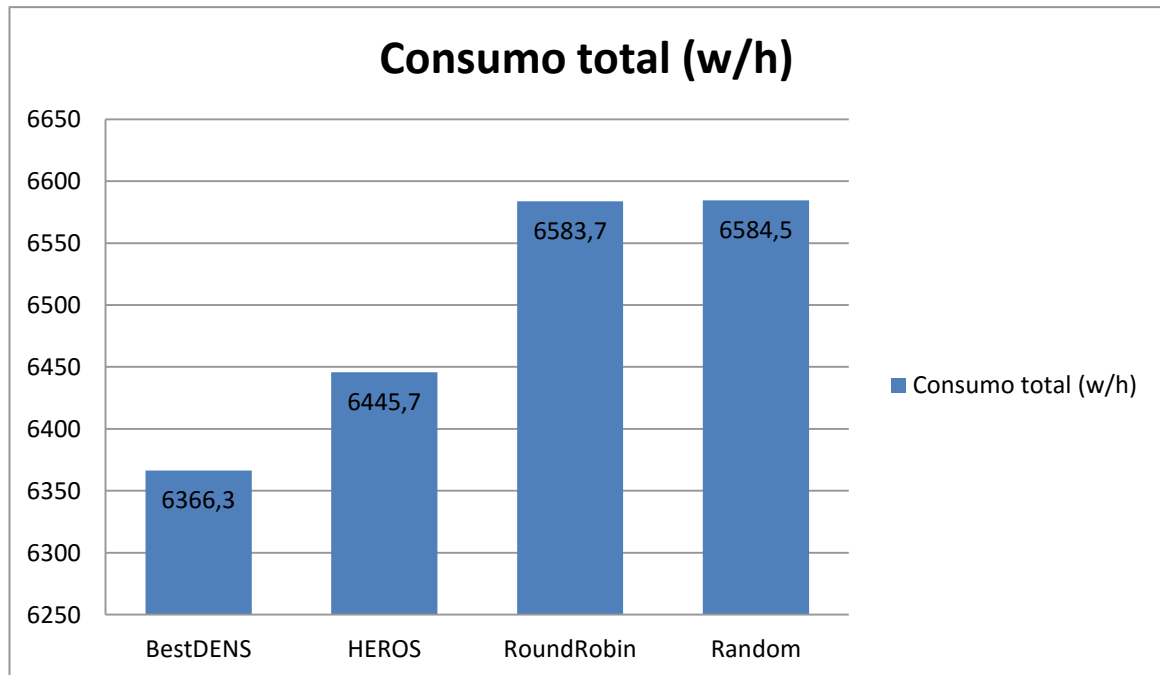


Figura 7: Consumo total do cenário 3

Nas Figuras 8 e 9 podemos perceber como o desempenho do algoritmo BestDENS poderia ser melhorado caso utilizássemos alguma tecnologia auxiliar, como o *Dynamic Shutdown*, para encontrar os aparelhos ociosos e os colocassem em modo de hibernação. Além de economizarmos com os aparelhos de computação e comunicação, teríamos a economia dos aparelhos de refrigeração. O BestDENS detecta hotspots de congestionamento na rede do data center e ajusta a sua metodologia de trabalho. Para trabalho intensivo com dados, o algoritmo é uma escolha realmente relevante, pois, este tipo de trabalho é mais limitado pela disponibilidade de recursos de comunicação do que pelas capacidades de processamento disponíveis.

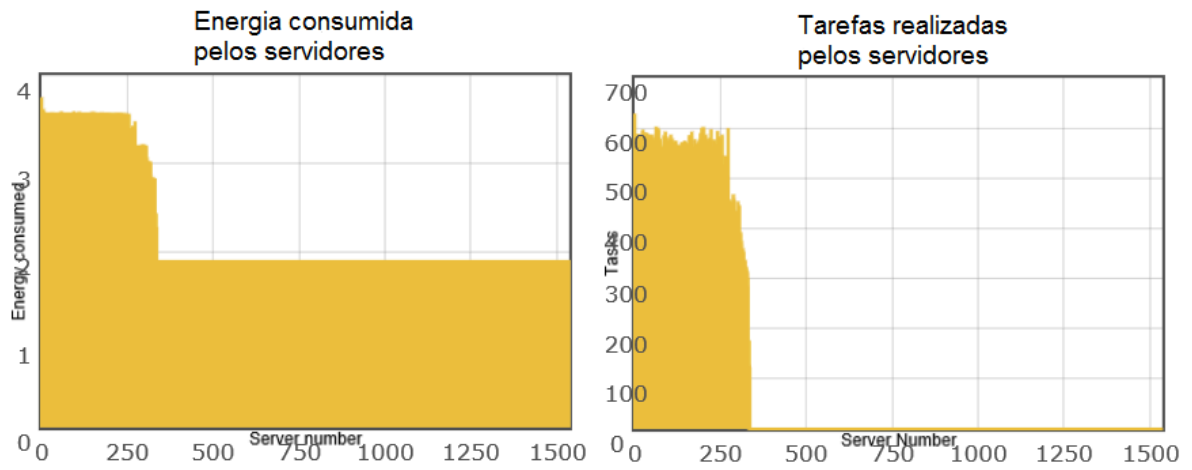


Figura 8: Relação energia e tarefas realizadas pelos servidores no algoritmo BestDENS

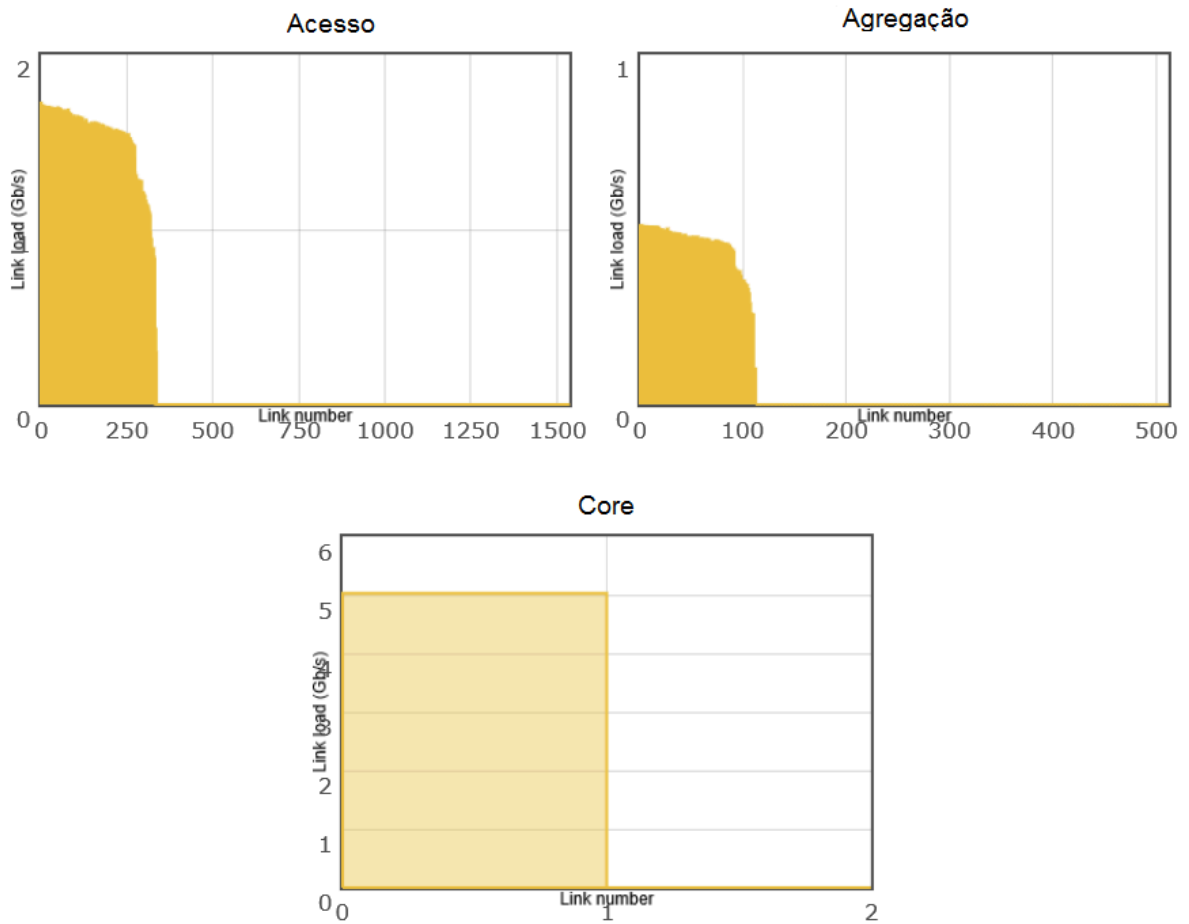


Figura 9: Uso dos switches no algoritmo BestDENS

6. Conclusões e Trabalhos futuros

Com estas medições, podemos perceber que o comportamento dos algoritmos estudados é influenciado pela quantidade de servidores no cenário. No primeiro ambiente, todos os algoritmos que foram aplicados não apresentaram grande eficiência, sendo o Round-Robin o que consegue uma pequena redução.

No segundo e terceiro cenário, tanto BestDENS quanto HEROS, conseguem uma redução significativa no consumo de energia, porém, quando analisamos as tarefas executadas, vemos que ambos possuem um alto índice de falhas, influenciando diretamente no custo/benefício da aplicação. Ao olharmos o desempenho do algoritmo Round-Robin, ele consegue uma pequena economia e além disso, executa todas as tarefas enviadas aos servidores. O algoritmo que apresenta o melhor desempenho é o Round-Robin.

Percebeu-se que o algoritmo BestDENS poderia ter um desempenho energético ainda melhor, caso fosse utilizado alguma tecnologia auxiliar que desligasse os aparelhos ociosos. Além disso, por os ambientes possuírem as mesmas configurações para todos os servidores e tarefas, fica evidente que o melhor algoritmo sempre, para o balanceamento de carga e conseqüentemente a economia de energia, seria o Round-Robin. Caso, tivéssemos servidores com configurações diferentes entre si, ou tarefas com tamanhos variados, provavelmente o BestDENS seria o melhor algoritmo para ser escolhido, aplicando outras técnicas junto a ele, como Dynamic shutdown, DVFS ou virtualização. Também foram notadas diversas falhas apresentadas pelo simulador utilizado, visto o seu recente lançamento. Como trabalhos futuros espera-se:

- Implementação e análise dos algoritmos utilizados no trabalho em outros tipos de arquitetura;
- Utilização de tecnologias adicionais como DVFS e virtualização nos cenários, e analisar o comportamento dos algoritmos;
- Implementação e análise dos algoritmos e cenários estudados em outros simuladores, comparando a precisão dos resultados de ambos;
- Implementação dos algoritmos em ambientes reais, para analisar o real impacto.
- Desenvolvimento de um algoritmo de escalonamento de tarefas próprio;

8. Referências

2016 TOP MARKETS REPORT CLOUD COMPUTING COUNTRY CASE STUDY,

2016, Aaa. 2016 ITA Cloud Computing Top Markets Report. 2016.

ARPACI-DUSSEAU, Remzi H.; ARPACI-DUSSEAU, Andrea C. **Operating systems: Three easy pieces.** Arpaci-Dusseau, 2014.

BAWDEN, Tom. **Global warming: Data centres to consume three times as much energy in next decade, experts warn. 2016.** Disponível em:

<<http://www.independent.co.uk/environment/global-warming-data-centres-to-consume-three-times-as-much-energy-in-next-decade-experts-warn-a6830086.html>>. Acesso em: 25 mar. 2016.

EVANS, Rich; GAO, Jim. **DeepMind AI reduces energy used for cooling Google data centers by 40%.** 2016. Disponível em: <<https://green.googleblog.com/2016/07/deepmind-ai-reduces-energy-used-for.html>>. Acesso em: 01 ago. 2016.

FARIA, Izaias de. **Uma abordagem de seleção de recursos consciente de consumo de energia baseada em topologia de rede, tamanho de arquivos e potência de equipamentos.** 2015. 102 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2015.

FROST; SULLIVAN. **Analysis of the Brazilian Cloud Computing Market,** 2014.

GUZEK, Mateusz; KLIAZOVICH, Dzmitry; BOUVRY, Pascal. **HEROS: Energy-**

efficient load balancing for heterogeneous data centers. 2015 IEEE 8th

International Conference on Cloud Computing. IEEE, 2015.

KLIAZOVICH, Dzmitry, et al. **GreenCloud: a packet-level simulator of energy-aware cloud computing data centers**. Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE. IEEE, 2010.

KLIAZOVICH, Dzmitry; BOUVRY, Pascal; KHAN, Samee Ullah. **DENS: data center energy-efficient network-aware scheduling**. Cluster computing 16.1 (2013): 65-75.

MEISNER, David; GOLD, Brian T. and WENISCH, Thomas F. "**PowerNap: eliminating server idle power**." ACM Sigplan Notices. Vol. 44. No. 3. ACM, 2009.

MONTEIRO, Rogério Castellar; DANTAS, Mario Antonio Ribeiro. **Green Cloud Computing: A Study Case of Resources and Network Energy Consumption**. 2012.

