

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Uso Eficiente De Energia Para Cloud Computing Através De Simulações

Rafael Rodrigues de Freitas

Florianópolis – SC

2011/2

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO – CTC
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA – INE
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Uso Eficiente De Energia Para Cloud Computing Através De Simulações

Rafael Rodrigues de Freitas

Trabalho de conclusão de curso
apresentado como parte dos
requisitos para obtenção do grau
de bacharel em Ciências da Computação.

Florianópolis – SC

2011/2

Sumário

1. INTORODUÇÃO.....	8
1.1 Objetivos.....	9
1.1.1 Objetivos Gerais.....	9
1.1.2 Objetivos específicos.....	9
1.2 Motivação.....	10
1.3 Organização do Trabalho.....	10
2 . FUNDAMENTAÇÃO TEÓRICA.....	11
2.1 Computação Utilitária.....	11
2.2 Acordo de Nível de Serviço (SLA).....	11
2.3 Virtualização.....	12
2.4 Computação em Nuvem.....	13
2.4.1 Origem do termo.....	13
2.4.2 Histórico.....	13
2.4.3 Visão Geral.....	14
2.4.4 Modelos de Serviço.....	15
2.4.5 Modelos de Implantação.....	16
2.4.6 Computação Verde.....	17
3. CloudSim.....	19
3.1 Definição.....	19
3.2 Arquitetura do CloudSim.....	20
3.2.1 Modelando a nuvem.....	21
3.2.2 Modelando a Alocação das Máquinas Virtuais.....	21
3.2.3 Modelando Cargas de Trabalho Dinâmicas.....	23
3.2.4 Modelando o Consumo de Energia de Data Center.....	23
3.3 Módulos do CloudSim.....	24
3.3.1 Núcleo do CloudSim.....	24
3.3.2 Processamento Interno do Datacenter.....	24
3.3.3 Comunicação Entre Entidades.....	25
3.4 Implementação do CloudSim.....	25
3.4.1 BwProvisioner.....	26
3.4.2 CloudCoordinator.....	26
3.4.3 Cloudlet.....	26

3.4.4 CloudletScheduler.....	26
3.4.5 Datacenter.....	27
3.4.6 DatacenterBroker.....	27
3.4.7 DatacenterCharacteristics.....	27
3.4.8 Host.....	27
3.4.9 NetworkTopology.....	27
3.4.10 RamProvisioner.....	27
3.4.11 SanStorage.....	28
3.4.12 Sensor.....	28
3.4.13 Vm.....	28
3.4.14 VmAllocationPolicy.....	28
3.4.15 VmScheduler.....	28
4. Proposta e Desenvolvimento.....	29
4.1 Cenários.....	29
4.2 Descrição da Implementação.....	30
4.2.1 Alocação/desalocação de máquinas virtuais.....	31
4.2.2 Ativação/desativação de máquinas físicas.....	32
4.2.3 Violações de SLA.....	33
4.2.4 Modelagem do consumo de energia.....	33
4.2.5 Modelagem da carga de requisições na nuvem.....	33
4.2.6 Alocação das requisições nas máquinas virtuais.....	34
4.3 Extensões e Implementações Feitas.....	34
4.3.1 NewBroker.....	35
4.3.2 VmAllocationPolicy.....	35
4.3.2 ConfigHosts e ConfigVms.....	35
4.3.3 PowerDatacenterNonPowerAwareExtended.....	35
4.3.4 Package Experimentos.....	35
4.3.5 CloudletWaiting.....	36
4.3.6 CloudletScheduler.....	36
4.3.7 VmScheduler.....	36
4.3.8 SensorGlobal.....	36
4.3.9 HostMonitor.....	36
4.3.10 VmMonitor.....	37
5. Resultados e Trabalhos Futuros.....	38
5.1 CloudSim.....	38
5.2 Objetivos do Desenvolvimento.....	38
5.3 Conclusões e Trabalhos Futuros.....	39

LISTA DE FIGURAS

Lista de Figuras

Figura 1: Arquitetura do CloudSim.....	20
Figura 2: Principais classes implementadas no CloudSim.....	26
Figura 3: Função utilizada.....	34

LISTA DE REDUÇÕES

SLA – Service Level Agreement (Acordo de Nível de Serviço)

SaaS – Software as a Service (Software como Serviço)

PaaS – Platform as a Service (Plataforma como Serviço)

IaaS – Infrastructure as a Service (Infraestrutura como Serviço)

EC2 – Elastic Compute Cloud

RRD – Round Robin Database

XML – Extensible Markup Language

RESUMO

O ambiente de Computação em Nuvem é um assunto amplo e vasto para o desenvolvimento de pesquisa científica. Neste trabalho de conclusão de curso é apresentada a contribuição realizada para validar uma abordagem de Computação em Nuvem Verde desenvolvida no tema de uma bolsa científica. Para isto foi necessário estudar, entender, usar e aperfeiçoar o simulador *CloudSim*, realizando os diversos experimentos necessários para comprovar o correto funcionamento de uma nova abordagem de Computação em Nuvem Verde.

Palavras-chave: computação em nuvem, nuvem verde, CloudSim, simulação

1. INTORODUÇÃO

Computação em nuvem nos últimos anos tem se tornado um dos termos mais comentados na área de tecnologia, principalmente por ter várias empresas grandes apostando nessa área.

Um ponto em questão a ser abordado em se tratando de computação em nuvem é o gasto com consumo de energia no centro de dados, assim como no sistema para climatizar o ambiente. É neste ponto que entra a computação em nuvem verde, onde o propósito é de usar estratégias para diminuir o processamento das máquinas, conseqüentemente diminuindo o consumo de energia elétrica por parte das máquinas e diminuindo o calor dissipado das mesmas, portanto diminuindo também a energia gasta com climatizadores de ar.

Pesquisas nesta área se tornam de difícil realização por causa da necessidade de um centro de dados de grande porte para a realização de testes e experimentos. Para ajudar a realização de testes e experimentos foram sendo criadas ferramentas para contornar o problema citado anteriormente. Tais ferramentas ajudam a simular um ambiente de computação em nuvem compatível em tamanho com o ambiente a ser simulado.

O objetivo deste trabalho é mostrar a computação em nuvem, como é organizada, como funciona e quais as suas finalidades, descrever o que é computação verde, os problemas energéticos encontrados e como podemos solucionar através de modelos de

simulação utilizando a ferramenta de simulação em computação em nuvem *CloudSim*, tendo seu princípio servir de suporte para simulações de ambientes de computação em nuvem.

Será também descrito neste trabalho o que é e como funciona o *CloudSim*, assim como suas características e a arquitetura do mesmo. Será mostrada as funcionalidades necessárias no *CloudSim* para a realização dos cenários propostos para que a simulação execute conforme o planejado e por consequência as implementações feitas estendendo a ferramenta para atingir os objetivos.

1.1 Objetivos

1.1.1 Objetivos Gerais

Este trabalho de conclusão de curso tem como objetivo fazer um estudo geral na área de computação em nuvem assim como ter um conhecimento mais aprofundado em computação em nuvem verde e na ferramenta de simulação de computação em nuvem *CloudSim*. Também tem como objetivo adaptar a ferramenta para simular os cenários que serão vistos neste trabalho.

1.1.2 Objetivos específicos

Este trabalho tem como pretensão:

- Apresentar um conceito geral sobre computação em nuvem;
- Conceituar e levantar os problemas da computação em nuvem verde;
- Demonstrar o *CloudSim* mostrando sua arquitetura e funcionalidades;
- Demonstrar cenários exigidos para as simulações;
- Apresentar extensões feitas no *CloudSim* para os cenários vistos; e
- Mostrar resultados alcançados.

1.2 Motivação

Conforme citado anteriormente, pesquisas na área de computação em nuvem verde são de difícil realização por seu alto custo de implementação, por isso uma alternativa é o uso de ferramentas que ajudam a simular o ambiente a ser experimentado e testado.

Portanto o foco deste trabalho é o estudo e adaptação do *CloudSim* para realizar simulações de ambientes de computação em nuvem verde, assim evitando o alto custo que é ter um centro de dados para se fazer pesquisas.

1.3 Organização do Trabalho

Este trabalho está organizado em 4 capítulos. O capítulo 2 tem como serventia dar a base teórica sobre computação em nuvem para os capítulos seguintes. O capítulo 3 um uma introdução ao leitor do que é o *CloudSim*, mostra sua arquitetura interna e é descrita de uma forma geral o seu funcionamento, dando uma visão geral das principais classes. O capítulo 4 mostra os cenários que foram simulados, suas características e as extensões feitas no *CloudSim* para que os cenários possam ser simulados. Por fim o capítulo 5 mostra alguns resultados obtidos das simulações e terminando o capítulo é dedicado às considerações finais e trabalhos futuros.

2 . FUNDAMENTAÇÃO TEÓRICA

2.1 Computação Utilitária

Computação utilitária é o empacotamento de recursos computacionais, como processamento armazenamento de dados e a disponibilização de serviços, como os serviços tradicionais (como eletricidade, água, telefone) (WIKIPEDIA; *Utility Computing*, 2011).

No início a computação utilitária era apenas uma ideia, mas haviam os que acreditavam que no futuro seria possível que isso fosse real, que serviços além dos já tradicionais como água e luz prestados para a população teriam que dividir a cena com outros tipos de serviços, como o proposto pela computação utilitária, usar de recursos computacionais para prover serviços à população. Leonard Kleinhock, um dos chefes da ARPA-NET, em 1969 disse:

“As redes de computadores ainda estão na sua infância, mas, assim que estiverem crescendo e ficarem mais sofisticadas, provavelmente veremos a expansão da computação utilitária, que como energia elétrica e telefone, serão serviços fornecidos nas casas e escritórios pelo país.” (apud KLEINROCK, 2005).

2.2 Acordo de Nível de Serviço (SLA)

O acordo de nível de serviço, no geral chamado de SLA (que é acrônimo para *Service Level Agreement*), é termo do contrato assinado entre duas ou mais entidades onde no qual o nível da prestação do serviço é definido formalmente.

A definição do SLA no contrato pode ter várias informações, variando com a área e abrangência do contrato, entre elas podemos ter: disponibilidade, segurança, incidência

de erros, prioridades, performance e entre outros.

O gerenciamento do SLA acontece em duas fases, em uma primeira fase aonde se dá a negociação do contrato, onde o cliente e a prestadora do serviço irão acordar sobre garantias e preços, e na fase seguinte, se dá a monitoração do SLA propriamente dita, que serve para ver se as cláusulas do contrato estão sendo cumpridas.

No âmbito da computação em nuvem o acordo de nível de serviço se mostra de essencial importância, pois como os recursos computacionais são oferecido por terceiros, é de extrema importância que tenha um nível na qualidade da prestação do serviço e um controle na forma que se consiga garantir o nível exigido.

2.3 Virtualização

Em computação virtualização de hardware é a virtualização de computadores ou virtualização de sistemas operacionais. Esta técnica esconde a camada de *hardware* para o usuário do sistema, fazendo com que o que apareça para o usuário seja uma outra camada de *hardware*, essa abstrata. O software que controla a virtualização no início era chamado de *control program*, hoje em dia é comumente chamado de *hypervisor* ou VMM (que é o acrônimo para *Virtual Machine Monitor*) (WIKIPEDIA; *Hardware Virtualization*, 2011).

Esta camada abstrata de *hardware* apresentada ao usuário tem sempre algum sistema operacional por cima, e este conjunto da camada abstrata de *hardware* em conjunto com o sistema operacional se dá o nome de máquinas virtuais (*Virtual Machines*).

A virtualização vem sendo empregada desde muito tempo. Nos anos 70 a IBM utilizou virtualização em seus mainframes. A grande popularização pelo interesse da virtualização vem devido ao grande poder de processamento em conjunto com a capacidade de memória nos computadores pessoais nos dias de hoje, fazendo com que a virtualização seja possível de ser utilizada em vários ambientes, tanto profissional quanto

familiar.

2.4 Computação em Nuvem

2.4.1 Origem do termo

O termo computação em nuvem veio do termo *Cloud* ou nuvem usado na década de 90 para designar malha de circuitos nas redes de telecomunicações. Essas empresas de telecomunicações ofereciam serviços de Rede Privada Virtual (VPN – *Virtual Private Network*) ou de transmissão de dados que eram feitas por circuitos dedicados garantindo taxa de tráfego com um custo bem menor, fazendo assim um roteamento entre seus circuitos que eram redundantes e que eram pouco usados. As empresas garantiam a taxa de tráfego e evitavam congestionamento, mas a contra ponto, elas não sabiam por onde um pacote trafegava na rede, esse caminho se passou a chamar “nuvem”.

Para fazer uma comparação com o tema deste trabalho, Computação em Nuvem se assemelha à nuvem do parágrafo anterior devido a não se saber aonde estão os recursos computacionais utilizados.

2.4.2 Histórico

O primeiro registro que se têm de uma definição acadêmica formal relacionada com as definições mais atuais sobre Computação em Nuvem foi a de Chellappa (CHELLAPPA, 1997), que em sua definição de computação em nuvem, escreveu: “*Um paradigma computacional, onde as fronteiras da computação serão determinadas por lógicas econômicas ao invés de limites técnicos*”.

Em 1999, uma empresa chamada Salesforce foi fundada por Marc Benioff e se utilizava das tecnologias de empresas como a Yahoo! e a Google para prover soluções de negócios. Tais soluções tinham como conceito o software “sob demanda”, o que é hoje conhecido como SaaS (*Software As A Service*, ou *Software Como Serviço*). Trabalhando em cima desse modelo de negócios a empresa teve resultados muito bons graças à

flexibilidade e a velocidade, fazendo com que outras empresas se voltassem para este conceito.

Uma empresa que desempenhou grande papel no contexto da computação em nuvem foi a Amazon, tanto que seus serviços viraram referência. Durante 2001 a empresa noticiou que menos de 10% da capacidade computacional de todo o seu centro tecnológico era usada na maior parte do tempo e o grande restante servia para suprir os picos de demanda que aconteciam durante o ano em datas importantes (feriados, Natal, etc...). A partir de 2005 a Amazon começou a “alugar” a grande parte ociosa de recurso computacional para terceiro e a partir daí surgiu o serviço *Amazon Elastic Compute Cloud* (EC2), serviço que oferece máquinas virtuais sob demanda e cobra usando como base o tráfego na rede e/ou o processamento utilizado.

Já em 2007, grandes companhias como a Google, a IBM e até Universidades começaram a investir maciçamente em pesquisas sobre computação em nuvem e até hoje é tema que atrai grandes investimentos e já possui vários provedores de serviços diferenciados.

2.4.3 Visão Geral

Nos últimos anos o termo computação em nuvem está sempre entre os mais citados no contexto da área tecnológica, muita gente prevê que é o momento certo para a evolução das nuvens.

Como os recursos de hardware estão cada vez mais acessíveis por causa do preço, as gigantes de software como a Amazon, Google e Microsoft precisam manter enormes *data centers* para não terem problemas em picos de uso, mas na maioria das vezes não usam todo o potencial disponível, tanto que terceirizam estes recursos que não estão sendo aproveitados.

E qual a definição de computação em nuvem? Não há uma definição padrão e única para definir computação em nuvem, mesmo não sendo um conceito novo, não

existe uma definição formal e universalmente aceita. Pode-se dizer que a definição de computação em nuvem envolve diversas outras definições, como virtualização, acordo de níveis de serviço e computação em grade, fazendo com que usando esses conceitos se torne a base da computação utilitária.

2.4.4 Modelos de Serviço

Na computação em nuvem existem várias formas de prover serviços através dos recursos dos provedores aos seus usuários. Existem três principais categorias divididas de acordo com o uso desses recursos. Estas denominações das categorias não foram criadas junto com a computação em nuvem e nem são relacionadas estritamente à computação em nuvem, podendo portanto serem usadas em outros contextos. Apenas estas denominações foram apenas incorporadas pela computação em nuvem. Abaixo seguem os três principais modelos de serviço:

- **Software como Serviço** (*Software as a Service* ou SaaS): É uma forma de distribuição de software ou prestação de serviço onde um desenvolvedor de software produz, opera, mantém e disponibiliza para clientes remotos. Consequentemente estes clientes usufruem e pagam pelo software de acordo com as suas necessidades, isto é, sendo sob demanda. Geralmente é fornecido através da internet podendo ser acessado de qualquer lugar tendo acesso à mesma.

- **Infraestrutura como Serviço** (*Infrastructure as a Service* ou IaaS): É o serviço onde é disponibilizado ao cliente um ou vários recursos computacionais fundamentais, como processamento, armazenamento, rede e outros, fazendo com que o mesmo use software próprio. O cliente não detém o controle a infraestrutura da nuvem, mas tem o controle sobre o sistema operacional assim como armazenamento e componentes de rede. O que o provedor do serviço presta faz neste modelo é entregar ao cliente um conjunto de uma ou várias máquinas. A virtualização é uma das principais tecnologias utilizadas nesse modelo de serviço.

- **Plataforma como Serviço** (*Platform as a Service* ou PaaS): Fornece uma plataforma para o desenvolvimento de softwares (principalmente aplicações para web) que facilita o trabalho do desenvolvedor pois abstrai os requisitos de hardware e muito provavelmente outras camadas de softwares necessárias, como servidor web, base de dados.

Com esses três modelos citados é possível identificar o tipo de cliente alvo para cada modelo, sendo o Software como Serviço para o usuário final, a Plataforma como Serviço destinado à desenvolvedores e por último a Infraestrutura como Serviço sendo destinado a engenheiros.

2.4.5 Modelos de Implantação

Como visto anteriormente no capítulo 2.4.4 em Modelos de Serviço, também é possível categorizar em relação a capacidade de abrangência da nuvem computacional. As três categorias mais utilizadas são nuvem pública, nuvem privada e nuvem híbrida.

- **Nuvem Pública:** É uma infraestrutura na qual é destinada a suprir as necessidades do público em geral. É disponibilizada por uma organização que vende ou disponibiliza o serviço. Este tipo de nuvem propicia às organizações de economia de escala, mas por outro lado diminui nível de customização e aumenta o cuidado com a segurança dos dados, pois os mesmos estão espalhados em lugares desconhecidos e são de difícil recuperação.

- **Nuvem Privada:** A nuvem faz parte (é acessada) apenas de uma organização, sendo administrada por ela ou por terceiros. Serve para dar suporte aos usuários locais fornecendo uma infraestrutura ágil e flexível para suportar cargas de trabalho dentro da própria organização. É aonde ocorre os menores riscos devido a esta característica de ser privada.

- **Nuvem Híbrida:** É a complementação da nuvem privada com uma nuvem pública, por isto o nome de nuvem híbrida.

2.4.6 Computação Verde

Computação Verde fornece modelos e técnicas para a gestão integrada de dispositivos de computação e sistemas de suporte ambiental, visando à qualidade do serviço, robustez e eficiência energética. Além disso, a computação em nuvem verde ajuda a resolver os problemas que têm urgência de solução nos *data centers*, como por exemplo, atender à demanda de pico de carga. Apesar de que o uso médio da capacidade de processamento continua sendo muito inferior em quase todo tempo, comparando como os picos de carga, conforme descrito em Valancius *et al.* [Valancius et al. 2009].

É possível atingir o objetivo de economizar mais energia elétrica através do controle integrado de alocação de recursos no funcionamento de servidores, por exemplo, através da aplicação de técnicas para o uso de máquinas virtuais (MV), previsão de distribuição da carga e a operação das unidades de suporte que controlam a refrigeração do *data center*. Por exemplo, o sistema de gerência deve ser capaz de desligar os servidores não utilizados e reduzir também, proporcionalmente, o consumo de energia usada para refrigerar o ambiente, desativando equipamentos, em resposta das características da carga do sistema em um determinado período de tempo. Assim sendo, a eficiência do uso de energia no *data center* está comprometida sem controlar estes dispositivos, atingindo ganhos significativos de eficiência energética quando o controle integrado for realizado [Lefèvre and Orgerie, 2010].

No artigo [Liu et al. 2009] os autores apresentam a arquitetura de computação em nuvem verde para reduzir o consumo de energia em *data center*, garantindo o desempenho dos serviços dos usuários. Para verificar a eficiência do funcionamento desta arquitetura os autores usaram uma aplicação de jogo (Tremulous) como uma aplicação que usa máquinas virtuais. Os resultados mostraram que eles conseguiram economizar até 27% de energia, usando a arquitetura de computação em nuvem verde.

A arquitetura para computação em nuvem verde proposta neste relatório pode gerar uma economia de consumo de energia de até 70%, comparando com os *data centers* que utilizam o modelo de "uma infraestrutura por aplicação" (chamada de "modelo de Arquitetura Tradicional") e até 40% de economia de consumo de energia em relação ao uso de modelos com base em computação em nuvem (o que é chamado de "modelos de computação em nuvem").

3. CloudSim

3.1 Definição

CloudSim é uma ferramenta de simulação feita na linguagem Java com o propósito de simular ambientes de computação em nuvem nos quais seria muito oneroso colocar em prática o ambiente real, fazer experimentações em ambientes de computação em nuvem e fazer a modelagem de um *data center* no *CloudSim* antes de implantar no mundo real, além de várias outras abordagens.

Foi projetado para preencher a lacuna que faltava para simular cenários de computação em nuvem, pois os simuladores disponíveis eram voltados para computação em grade, não oferecendo as funcionalidades necessárias que envolve um ambiente de computação em nuvem, como o isolamento das camadas de serviço (SaaS, PaaS e IaaS) e a modelagem de ambiente de centro de dados que consiste de centenas de milhares de máquinas.

O simulador oferece suporte para modelagem e simulação em larga escala de ambientes de computação em nuvem, incluindo *data centers*, em um simples computador, oferece também uma plataforma para modelar usuários dos serviços, políticas de alocações e configurar a descrição do ambiente a ser simulado.

É possível também simular conexões de rede através dos mais variados elementos do sistemas simulados. Permite simular ambiente de nuvens federadas em diferentes redes.

Algumas das características do *CloudSim* é a disponibilidade de um motor de virtualização que ajuda na criação e gerenciamento, de múltiplos e independentes serviços virtualizados em um *data center*. Outra característica é a flexibilidade de troca entre os tipos de alocação dos serviços virtualizados nos núcleos de processamento.

3.2 Arquitetura do *CloudSim*

A arquitetura o *CloudSim* pode ser dividida em 3 camadas principais:

- Camada do núcleo da simulação: É aonde é feito o controle da simulação propriamente dito, faz a gerência da execução dos eventos que acontecem na simulação e faz o controle da troca de mensagens entre as entidades presentes na simulação.
- Camada Principal do *CloudSim*: É aonde fica todo o controle da simulação em se tratando da nuvem, também é aonde são definidos os objetos nos quais vão ser simulados, como rede, recursos da nuvem, serviços da nuvem.
- Camada do usuário: É a camada aonde é definido o ambiente em qual vai ser simulado, bem como algumas configurações são feitas nesta camada, também aqui é definida algumas políticas do lado do usuário do *data center*.

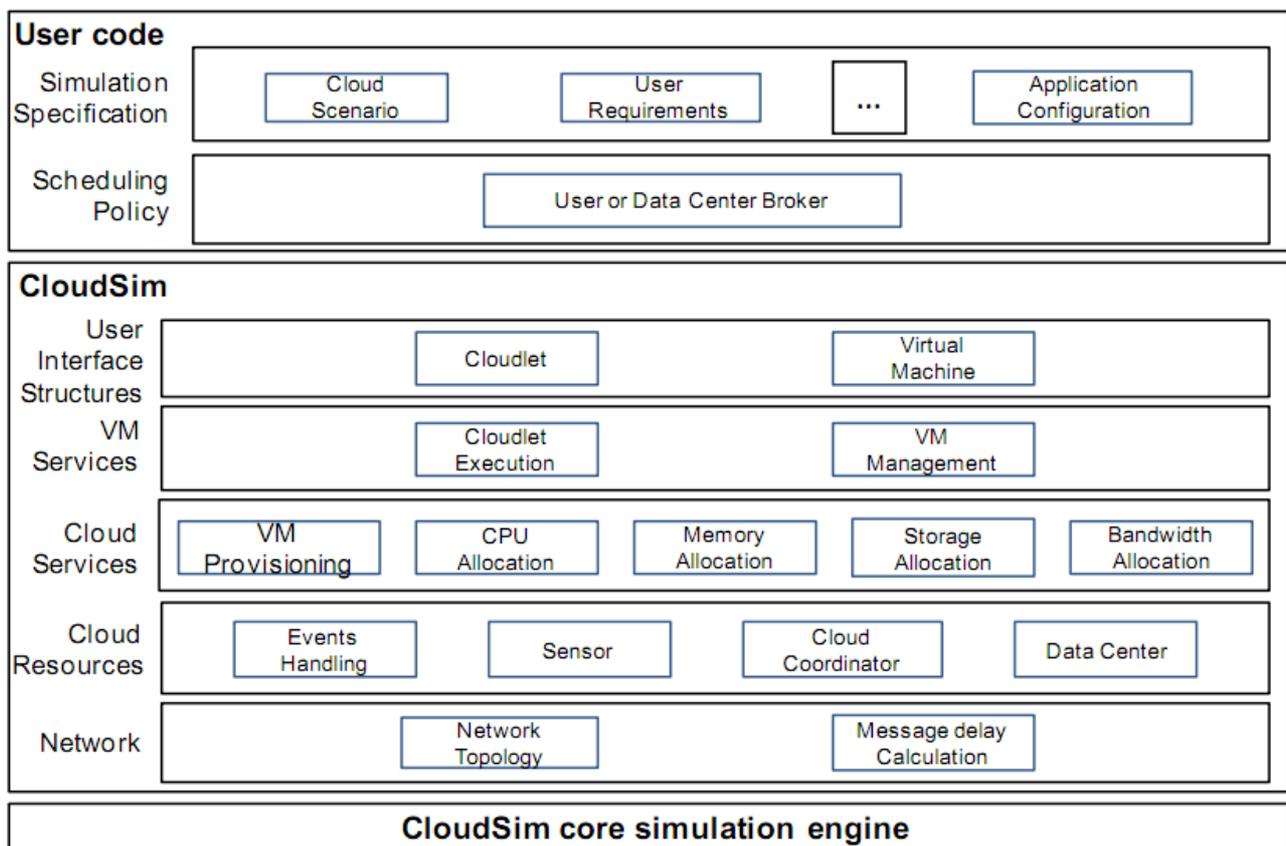


Figura 1: Arquitetura do *CloudSim*

A Figura 1 mostra o conteúdo das camadas principais do CloudSim citadas anteriormente.

3.2.1 Modelando a nuvem

A entidade *data center* no *CloudSim* gerencia uma quantidade de máquinas físicas, que no simulador é chamado de *host*. Para as máquinas físicas são designadas uma ou mais máquinas virtuais dependendo na política de alocação de máquinas virtuais que foi definido pelo provedor do serviço da nuvem. Aqui a política serve para as operações de controle relacionados ao ciclo de vida das máquinas virtuais, como designar uma máquina física à uma máquina virtual, criação de máquina virtual, a destruição das mesmas e a migração das máquinas virtuais.

Um *data center* pode gerenciar diversas máquinas físicas, que por sua vez gerencia os ciclos de vidas das máquinas virtuais. Uma máquina física é um componente dentro do simulador que representa um servidor computacional em uma nuvem, é atribuído uma capacidade de processamento (expressa no simulador em MIPS – Milhões de Instruções Por Segundo), quantidade de memória, quantidade de espaço de armazenamento e uma política de alocação de núcleos de processamento para as máquinas virtuais. É implementado no componente que representa a máquina física no simulador interfaces que suportam a modelagem e a simulação de máquinas com um núcleo ou com vários núcleos (*single-core* e *multi-core*, respectivamente).

A alocação de máquinas virtuais é o processo de criar instâncias em máquinas físicas que casam com as características críticas (como armazenamento e capacidade de memória), configurações (ambiente de software) e requerimentos do provedor de serviço.

3.2.2 Modelando a Alocação das Máquinas Virtuais

Um dos principais aspectos que fazem a computação em nuvem diferente da computação em grade é a implantação massiva de ferramentas e tecnologias de virtualização. Esta virtualização é tratada como uma camada a parte na arquitetura da

computação em nuvem que age como um executor, gerenciador e um ambiente para as aplicações serem executadas.

Mesmo estando em um contexto isoladas, as máquinas virtuais estão dividindo processamento e barramento do sistema com outras máquinas virtuais alocadas na mesma máquina física. Este fator crítico deve ser levado em conta quando se faz a alocação de máquinas virtuais para que não se aloque mais capacidade de processamento do que a máquina física pode oferecer.

Por este ponto de vista tentando simular diferentes políticas de provisionamento tentando variar o nível de isolamento da performance das máquinas virtuais nas máquinas físicas, o CloudSim suporta dois níveis de provisionamento. O primeiro é a nível das máquinas físicas, onde é possível quantificar o processamento para cada máquina virtual em específico. No segundo nível, que é na máquina virtual, é atribuído uma quantidade fixa de processamento para cada aplicação executando na máquina virtual em específico.

Em cada nível o CloudSim implementa duas políticas de provisionamento usando compartilhamento por tempo e compartilhamento por espaço. Portanto existem quatro combinações possíveis para serem usadas com as políticas implementadas por padrão no CloudSim. São elas:

- Compartilhamento por espaço na máquina virtual e na máquina física: As aplicações serão executadas uma por uma dentro da máquina virtual e as máquinas virtuais também serão executadas uma por uma, exemplificando, existe uma fila de máquinas virtuais para executar suas aplicações que por sua vez estão em uma fila de execução dentro de sua máquina virtual.

- Compartilhamento por espaço na máquina virtual e compartilhamento por tempo na máquina física: As aplicações serão executadas uma a uma nas máquinas virtuais mas as máquinas virtuais serão executadas dentro de uma máquina física paralelamente.

- Compartilhamento por tempo na máquina virtual e compartilhamento por espaço

na máquina física: As aplicações executam paralelamente dentro da máquina virtual mas as máquinas virtuais executam em ordem dentro das máquinas físicas.

- Compartilhamento por tempo na máquina virtual e na máquina física: Ambas as aplicações e máquinas virtuais são executadas em paralelo.

Os exemplos descritos para as combinações são consideradas apenas usando um núcleo de processamento.

3.2.3 Modelando Cargas de Trabalho Dinâmicas

Desenvolvedores e provedores de serviços desenvolvem aplicações que possuem um comportamento dinâmico em termos de carga, disponibilidade e escalabilidade.

Em se tratando de computação em nuvem, é importante que existam vários tipos de configurações de máquinas virtuais para prover o máximo de compatibilidade às aplicações em se tratando de cargas dinâmicas.

Um ponto importante para o tratamento desta carga dinâmica é um modelo de utilização implementado no simulador para que se consiga determinar a necessidade de processamento da aplicação. Por padrão vem dois modelos de utilização (um aleatório e outro utilizando carga máxima) podendo ser estendível para outros tipos.

Também é possível a modelagem de violações de SLA fazendo com que políticas descrevam em qual situação pode ocorrer a violação do acordo.

3.2.4 Modelando o Consumo de Energia de Data Center

Ambientes de computação em nuvem são construídos em um largo número (centenas de milhares) de máquinas para fornecer os tipos de serviços descritos anteriormente. Tais ambientes junto com o conjunto de aparelhos para manter o clima em uma temperatura aceitável para as máquinas consome um número muito alto de energia elétrica resultando em um custo operacional alto.

No CloudSim existe por padrão modelos básicos e entidades para validar e avaliar as técnicas empregadas para evitar o consumo excessivo de energia. Existem vários tipos

de implementações feitas para gerenciar e modelar o consumo de energia nos mais variados níveis de um ambiente de computação em nuvem.

3.3 Módulos do CloudSim

3.3.1 Núcleo do CloudSim

O núcleo do CloudSim é composto principalmente pela sua classe principal onde ocorre a simulação, as filas dos eventos que foram adiadas e os que ainda vão ocorrer e as entidades do núcleo, que trocam mensagens entre si para fazer a simulação ser executada.

A classe principal gerencia as filas de eventos e controla passo a passo a execução destes eventos, cada evento quando criado é enviado para a fila de eventos futuros com o tempo em que será executado.

As entidades são as principais peças no CloudSim em se tratando de execução da simulação, pois cada mensagem enviada entre as entidades se transforma em um evento que será executado pela classe principal do simulador. Todas as entidades possuem um método que se chama `processEvent()` que como o próprio nome do método diz faz o processamento do evento que nada mais é uma mensagem enviada para esta entidade.

Faz parte da troca de mensagens entre as entidades um código que serve para a entidade saber sobre o que se trata a mensagem. Existem vários códigos padrões no CloudSim usados para tal finalidade.

A mensagem enviada, que logo depois se transforma em evento a ser executado, tem como seus atributos o tipo, tempo de envio, tempo no qual o evento deve ser executado, quem enviou, qual é o destinatário e o código para a identificação do evento pelo destinatário.

3.3.2 Processamento Interno do *Datacenter*

O processamento das aplicações dentro das máquinas virtuais fica a cargo do

Datacenter na simulação. Para cada máquina física presente no *Datacenter*, é feita a atualização das aplicações dentro de cada máquina virtual presente na máquina física, cada máquina virtual retorna o tempo final da aplicação que falta menos tempo para ser completada a execução. Em seguida a máquina física retorna para o *Datacenter* o menor tempo retornado pelas máquinas virtuais.

Com o menor tempo em mãos, o *Datacenter* envia uma mensagem para ele próprio passando como atributo da mensagem o tempo mínimo conseguido nas máquinas virtuais, que nada mais é do que o tempo de fim de execução da aplicação que está mais perto de terminar a execução, que servirá de tempo para o próximo processamento das aplicações no *Datacenter*.

3.3.3 Comunicação Entre Entidades

A comunicação entre entidades funciona por causa de uma classe chamada *Cloud Information Service*, que é na qual é guardada todas as entidades da simulação em questão. Essa classe guarda tanto entidades criadas no início da simulação (entidades estáticas) tanto entidades criadas em tempo de execução.

Através deste serviço é possível enviar mensagem entre entidades sabendo apenas o nome da entidade, sem precisar saber o número de identificação da entidade ou outro dado qualquer.

3.4 Implementação do *CloudSim*

O *CloudSim* é implementado onde cada classe possui uma função específica dentro da simulação, seja para representar um objeto físico na simulação (por exemplo, uma máquina física) tanto para representar uma política de alocação de máquinas virtuais, a seguir será descrito as principais classes (como consta na Figura 2) que compõem a camada principal do *CloudSim* mostrada no capítulo 3.2.

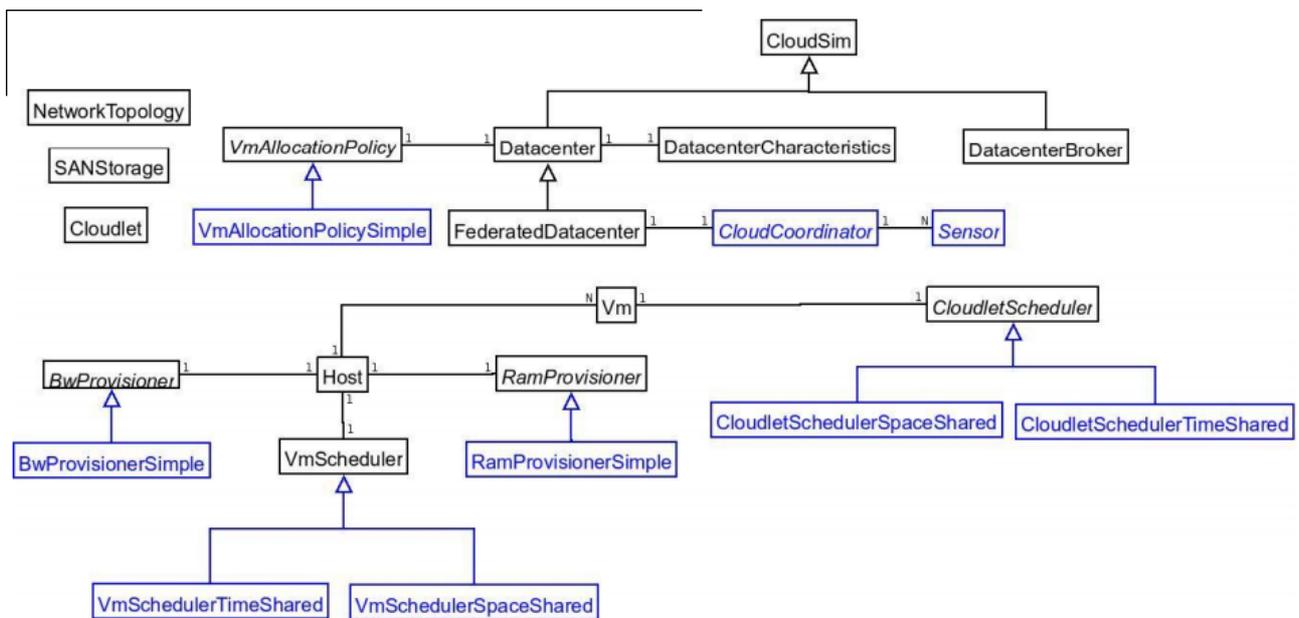


Figura 2: Principais classes implementadas no CloudSim

3.4.1 BwProvisioner

Classe abstrata que modela as políticas de largura de banda para as máquinas virtuais. A principal função desta classe é alocar a largura de banda para um conjunto de máquinas virtuais concorrentes em um centro de dados. Uma classe que implementa os métodos desta classe é a BwProvisioningSimple, onde se permite alocar a quantidade de banda necessária, porém é diminuída da banda disponível pela máquina física.

3.4.2 CloudCoordinator

Classe abstrata que monitora os múltiplos centro de dados caso seja necessário na simulação e toma decisões dependendo da monitoração.

3.4.3 Cloudlet

Esta classe modela as aplicações que irão ser executadas na nuvem. Possui atributos necessários para a sua execução como tamanho e quantidade de núcleos necessários.

3.4.4 CloudletScheduler

Classe abstrata que é estendida para implementar diferentes políticas que determinam o processamento das *Cloudlets* dentro de uma máquina virtual. Os dois tipos

de políticas que são implementadas no CloudSim são de compartilhamento de espaço (CloudletSchedulerSpaceShared) e de tempo (CloudletSchedulerTimeShared).

3.4.5 Datacenter

Classe que modela o nível de Infraestrutura como Serviço oferecido por provedores como Amazon, Google, Microsoft. Encapsula um conjunto de computadores (máquinas físicas) que podem ser homogêneos ou heterogêneos .

3.4.6 DatacenterBroker

Classe que modela o agente que age como o mediador entre o usuário e o provedor da nuvem, é através dele que são enviados os *Cloudlets* para serem executados e também a alocação das máquinas virtuais.

3.4.7 DatacenterCharacteristics

Classe que contém as informações referentes aos recursos do centro de dados.

3.4.8 Host

Classe que modela as máquinas físicas. Possui informações sobre a quantidade de processamento, armazenamento, largura de banda e quantidade de memória disponível. Responsável por compartilhar recursos entre as máquinas virtuais.

3.4.9 NetworkTopology

Classe que contém a informação para o comportamento da rede na simulação.

3.4.10 RamProvisioner

Classe abstrata que representa a política de alocação de memória principal para as máquinas virtuais. Na simulação quando uma máquina virtuais é alocada em uma máquina física essa classe é responsável por verificar se há memória principal suficiente na máquina física para alocar a máquina virtual. A classe RamProvisionerSimple é uma implementação desta classe onde a máquina virtual pode pedir a alocação de qualquer

quantidade de memória principal da máquina física.

3.4.11 SanStorage

Classe que modela uma rede de armazenamento comum em centro de dados utilizados para armazenamentos (como Amazon S3). Esta classe implementa uma interface simples que pode ser usada para simular armazenamento e leitura de qualquer quantidade de dados, sujeito a disponibilidade da largura de banda da rede.

3.4.12 Sensor

Interface usada para ser implementada por um sensor que por sua vez é usado para o CloudCoordinator tirar conclusões e fazer decisões a partir das informações que os sensores monitoram.

3.4.13 Vm

Classe que modela uma máquina virtual, que é gerenciada e alocada em uma máquina física (classe Host).

3.4.14 VmAllocationPolicy

Classe abstrata que representa a política de alocação das máquinas virtuais nas máquinas físicas. A principal funcionalidade desta classe é verificar se uma máquina física tem requisitos suficientes para alocar uma máquina virtual em específico.

3.4.15 VmScheduler

Classe abstrata que implementa as políticas de distribuição de processamento da máquina física entre as máquinas virtuais alocadas nesta máquina física.

4. Proposta e Desenvolvimento

O *CloudSim* possui várias classes implementando políticas e funcionalidades simples para que se consiga uma simulação sem grande detalhamento no cenário simulado. Com estas classes nativas no *CloudSim* é possível ter apenas um tipo de alocação de máquinas virtuais nas máquinas físicas, apenas dois tipos de escalonamento de processos nas máquinas virtuais (compartilhamento de tempo ou espaço) e dois tipos de escalonamento de máquinas virtuais nas máquinas físicas (como nos processos nas máquinas virtuais, compartilhamento de tempo ou espaço).

Com isto vemos que caso se deseje um comportamento diferente do descrito acima, é necessário fazer extensões para se obter o resultado esperado nas simulações.

A seguir é descrito os cenários para que após seja feita a descrição da implementação.

4.1 Cenários

Ao todo neste trabalho foram implementadas seis cenários para serem avaliadas e validadas. Estes cenários fazem parte no desenvolvimento de outros trabalhos como visto no de Werner (Werner, 2011). Abaixo as quatro primeiras:

1 – Sem nenhuma política de economia de energia: Não há políticas para desligamentos das máquinas físicas, migração de máquinas virtuais e tampouco de realocação das mesmas.

2 – Aqui ocorre a migração de máquinas virtuais quando uma máquina física estiver com a carga abaixo de 20% para que se concentre a carga em apenas em um número reduzido de máquinas físicas.

3 – Neste cenário há a realocação da capacidade de processamento da máquina virtual caso sua carga de trabalho ultrapasse os 90%.

4 – Neste cenário é o cenário 2 junto com o cenário 3.

Todos os cenários acima tem o mesmo ambiente de simulação, que são dois centros de dados com 100 máquinas físicas cada, cada máquina física possui 10GB de memória, 1TB de armazenamento, processador de 4 núcleos executando 3000 MIPS (milhões de instruções por segundo) cada.

As configurações das máquinas virtuais tem processador de 2 núcleos com 1000 MIPS cada, 1GB de memória e 10GB de armazenamento.

Os dois cenários descritos a seguir diferem nas características do ambiente de simulação das quatro anteriores, serão simulados ambientes usando apenas um centro de dados, com apenas 8 máquinas físicas. Cada máquina física apresenta um processador de quatro núcleos executando 3000 MIPS cada, com 10GB de memória e 1TB de armazenamento, as máquinas virtuais possuem 2.5GB de memória, processador de quatro núcleos executando 3000 MIPS cada e tendo um armazenamento de 100GB. São elas:

1 – Cenário sob demanda, onde o ambiente (alocação/desalocação de máquinas virtuais, ativação/desativação de máquinas físicas) vai se moldando de acordo com as necessidades que vão aparecendo durante o espaço de tempo.

2 – Cenário usando *sparcs*, onde sempre existe uma máquina virtual e uma máquina física ocioso esperando em algum momento serem utilizados para um eventual pico de carga.

4.2 Descrição da Implementação

As características básicas do ambiente simulado, representando máquinas físicas e máquinas virtuais foram obtidas a partir dos recursos computacionais usados na nossa universidade, ou seja, do data center em uso na Universidade Federal de Santa Catarina.

Estes acessos serão chamados daqui em diante de requisições que chegam no centro de dados, que na simulação chegam em grupos de 3 em 3 segundos, para cada

requisição é gerada uma mensagem ao centro de dados que por sua vez responde indicando que foi iniciada a execução da requisição com sucesso, só então é possível dar entrada à próxima requisição ao centro de dados.

Para controlar a carga nas máquinas virtuais e máquinas físicas foi implementado uma extensão às máquinas virtuais e máquinas físicas que permitem às mesmas serem monitoradas por um sensor externo, que faz com que caso algum evento diferente do esperado aconteça ele tome as medidas necessárias de acordo com cada cenário em execução.

Este sensor externo assim como as requisições que chegam à nuvem, também é acionado em intervalos de 3 em 3 segundos para checar as condições do ambiente. Assim como acontece com o processamento interno da classe Datacenter, aqui também acontece o mesmo, depois de realizar suas tarefas tanto o sensor externo quanto a entidade que envia as requisições para a nuvem enviam uma mensagem para si mesmos avisando que em 3 segundos eles terão que executar esta mesma tarefa.

4.2.1 Alocação/desalocação de máquinas virtuais

Todos os cenários exceto o cenário que utiliza máquinas *saves* apresentam o mesmo comportamento para a criação de mais máquinas virtuais durante a execução da simulação. Caso a média de todas as máquinas virtuais em execução estiver acima de 90% então é criada uma nova máquina virtual, que dependendo de qual cenário estiver sendo executado depende aonde será alocada, se for algum dos quatro primeiros apresentados, será alocada em qualquer máquina física ligada, caso o cenário em execução seja algum o quinto cenário apresentado, então a máquina física escolhida é a que tiver mais máquinas virtuais desde que seja cumpridos os requisitos de recursos disponíveis para a máquina virtual.

Caso seja o cenário usando *saves*, a máquina virtual será criada a partir que a máquina virtual que esteja sendo a *save* seja utilizada para executar alguma requisição

e portanto ela perde este posto, passando a ser a máquina virtual *spare* a máquina que acabou de ser criada.

Em todas os cenários o tempo para ativar uma máquina virtual é de dez segundos.

Em se tratando de desalocação de máquinas virtuais, todos os cenários seguem um mesmo padrão, a máquina virtual deve estar ociosa por pelo menos um certo período de tempo para depois ser desalocada, para todos os cenários o tempo é de dois minutos, apenas no cenário que utiliza *sparers* a máquina virtual que ocupa esta função não deve ser desalocada.

4.2.2 Ativação/desativação de máquinas físicas

Na questão de ativação e desativação de máquinas físicas para cada cenário é diferente, no primeiro cenário apresentado como não existe nenhuma política para diminuir o consumo de energia então não há política de desativação e muito menos de ativação de máquinas físicas.

No segundo cenário apresentado para cada vez que ocorre a migração de máquinas virtuais a máquina física que tinha a carga baixa e teve suas máquinas virtuais migradas para outro lugar é desligada. Só volta a ser ligada quando não há mais espaço para alocação de máquinas virtuais e é necessário alguma máquina física.

O terceiro cenário apresentado também não há nenhuma política de ativação e desativação de máquinas físicas como acontece no primeiro cenário apresentado pois só se trata de realocação de máquinas virtuais.

O quarto cenário apresentado possui a mesma política do segundo cenário apresentado por ter a mesma política de migração de máquinas virtuais.

O cenário sob demanda possui uma política de ativação de máquinas físicas conforme diz a própria descrição do cenário, quando não há mais lugar para alocação para máquinas virtuais e isso se faz necessário, então é ativada uma máquina física. A desativação se faz presente quando uma máquina física não possuir mais máquinas

virtuais por mais de cinco minutos.

O cenário que faz uso de *sparers* se baseia na mesma política tanto para alocar uma nova máquina virtual tanto para ativar uma máquina física, quando uma máquina física está como *spare* recebe uma máquina virtual então esta deixa de ser *spare* e então é ativada uma máquina física, que por sua vez se torna a *spare*. O processo de desativação é o mesmo do cenário anterior, excetuando-se que a máquina física que é *spare* não deve ser desativada.

Nos últimos dois cenários apresentados o tempo para ligar uma máquina física é de 2 minutos.

4.2.3 Violações de SLA

As violações de SLA são determinadas pelo tempo em que uma requisição é processada no centro de dados. Em todos os cenários o tempo para que uma requisição seja executada sem ocorrer violação de SLA é de dez segundos.

4.2.4 Modelagem do consumo de energia

Em todos os cenários o consumo de energia é definido no consumo dos núcleos dos processadores, onde consome 140 watts.

4.2.5 Modelagem da carga de requisições na nuvem

Em se tratando de número de requisições que chegam ao centro de dados em um instante de tempo, foram usadas várias abordagens, são elas:

- **Carga aleatória:** Onde não é definido um padrão para a quantidade de requisições, a cada iteração em que é enviada as requisições, é gerado uma quantidade aleatória.

- **Carga fixa:** É fixado uma quantidade que sempre será usada em cada iteração do envio de requisições para o centro de dados.

- **Carga baseada em RRD/XML:** RRD é um banco de dados que armazena uma

série de dados de um determinado tipo de variável, como tráfego de uma rede, temperatura de um processador, ou no caso deste trabalho, carga de trabalho de uma máquina. É gerado a partir deste banco de dados um documento em XML para assim fazer a leitura dos dados e pegar a quantidade de requisições a ser usada em um instante de tempo.

- **Carga baseada em função matemática:** A partir de uma função matemática mostrada na Figura 3 é possível adquirir valores usados na quantidade de requisições em cada iteração.

$$\left| \sin\left(\frac{x}{2}\right) \cos\left(\frac{x^2}{2}\right) \right|$$

Figura 3: Função utilizada

Vale destacar que o *CloudSim* não possui nativamente nenhuma modelagem para a carga usada, cabendo ao usuário implementar as políticas necessárias.

4.2.6 Alocação das requisições nas máquinas virtuais

Nos quatro primeiros cenários apresentados a política de seleção de máquinas virtuais para a execução de requisições é feita de forma aleatória dentre as máquinas virtuais alocadas nas máquinas físicas do centro de dados.

Já nos outros dois cenários a estratégia adotada é de executar a requisição em uma máquina virtual que esteja com sua carga de trabalho alta, mas abaixo de 90% de sua capacidade máxima. Com isso faz-se com que o processamento de requisições fique mais concentrado.

4.3 Extensões e Implementações Feitas

Foram feitas várias extensões e implementadas várias funcionalidades para com que o *CloudSim* se encaixasse nas necessidades dos cenários, a seguir será mostrada quais implementações foram feitas para atender essas necessidades.

4.3.1 NewBroker

Foi feita uma nova classe em substituição ao DatacenterBroker já presente no CloudSim. Nesta nova classe é encarregada da gerencia de grande parte da simulação, é onde é gerenciado o envio de requisições para o centro de dados, a contabilidade de violações de SLA, onde sai as mensagens para o centro de dados tanto para alocar/desalocar máquinas virtuais quanto ligar/desligar máquinas físicas. É nesta classe onde é calculado a quantidade de requisições a ser enviadas para o centro de dados em um certo instante quanto a tomar decisões de quando a média das máquinas virtuais estiver alta.

É também responsável pelo fim da simulação.

4.3.2 VmAllocationPolicy

Foram feitas duas extensões a esta classe para implementar os dois tipos diferentes de alocação de máquinas virtuais nas máquinas físicas. Uma classe para aplicar a política dos quatro primeiros cenários apresentados e outra classe para implementar os outros dois cenários, já que ambos usam a mesma política, diferenciando-se apenas na questão da máquina física *spare*.

4.3.2 ConfigHosts e ConfigVms

Foram criadas estas duas classes para guardar a configuração das máquinas físicas e máquinas virtuais para facilitar a implementação.

4.3.3 PowerDatacenterNonPowerAwareExtended

Foi feita uma extensão da classe PowerDatacenterNonPowerAware para fazer o calculo da energia consumida durante a simulação diferente do que já vem implementado no *CloudSim*.

4.3.4 Package Experimentos

Foi criado um pacote que contém as classes que contém as características de cada

cenário bem como a implementação de métodos para desalocar máquinas virtuais e alguns específicos de cada cenário.

4.3.5 CloudletWaiting

Classe estendida da classe *Cloudlet* que implementa a verificação da quebra de SLA, foram implementados atributos que verificam se a requisição ultrapassou o valor limite de tempo máximo antes de violar o SLA.

4.3.6 CloudletScheduler

Extensão da classe *CloudletSchedulerTimeShared* que implementa a execução das requisições dentro de uma máquina virtual, foram adicionadas políticas para verificar durante a execução das requisições se as mesmas tinham seu limite de tempo máximo de execução estourado.

4.3.7 VmScheduler

Extensão da classe *VmSchedulerTimeShared* que suprime algumas políticas nativas do *CloudSim* em relação a alocação de processamento para as máquinas virtuais alocadas em uma mesma máquina física.

4.3.8 SensorGlobal

Classe criada como uma entidade no *CloudSim*, tendo em mãos o poder de enviar mensagens durante a simulação para outras entidades. Tem como função principal monitorar as máquinas físicas e máquinas virtuais e caso algum gatilho ocorra (por exemplo, média de processamento maior que 90% nas máquinas virtuais). Caso esta hipótese seja confirmada envia uma mensagem para o *NewBroker* para tratar do caso.

4.3.9 HostMonitor

Classe estendida da classe *PowerHost* que implementa os três estados da máquina física, o desligado, em uso e em stand by, sendo que para cada estado o

consumo de energia é diferente. Também é implementado nessa classe a máquina física usando *spare*, pois o cenário usando spares utiliza esta classe.

4.3.10 VmMonitor

Classe estendida da classe Vm, aonde é implementada a carga de utilização da máquina virtual e também os atributos que verificam se a máquina virtual está ociosa ou se ela funciona como *spare*.

5. Resultados e Trabalhos Futuros

Foi feita uma breve introdução nos conceitos que envolvem a computação em nuvem, como surgiu, modelos de implantação e de serviço e por fim uma introdução no que norteia a computação em nuvem verde.

5.1 CloudSim

A ferramenta *CloudSim* foi apresentada neste trabalho no Capítulo 3 mostrando sua arquitetura, suas camadas e suas principais classes que fazem parte principal no que se diz respeito a simulação dos cenários.

5.2 Objetivos do Desenvolvimento

Foi apresentado no Capítulo 4 os tipos de cenários que foram implementados, suas características que diferem um dos outros e suas políticas para as diferentes áreas dentro do centro de dados.

Os cenários necessitavam de funcionalidades que o *CloudSim* não possuía por padrão, assim foram feitas várias extensões no código da ferramenta para atingir o grau de satisfação desejado de cada cenário.

Foram feitos diversos testes nas simulações, em cada cenário descrito anteriormente, foram feitas baterias de testes usando como parâmetros diversas variáveis, como tamanho da requisição, carga a ser empregada e um multiplicador para aumentar o tipo de carga usada.

Por fim a ferramenta se tornou bastante simples de ser estendida pois por ser bastante modular, com isso facilitou o trabalho de adaptar a ferramenta aos cenários em questão.

5.3 Conclusões e Trabalhos Futuros

Com este trabalho se constatou a facilidade de fazer experiências e testes em ambientes de computação em nuvem verde usando simulações através da extensão e aprimoramento do *CloudSim*.

Verificando os resultados se constata a falta de uma base relacionando os resultados apresentados neste trabalho com um resultado adquirido em um ambiente real, portanto um dos caminhos a ser visto é a questão da validação dos cenários simulados, até que ponto eles são fiéis em um ambiente real.

Outro ponto interessante para um trabalho futuro seria de usar outras ferramentas ou simuladores como o *CloudSim* para implementar estes mesmos cenários e fazer a comparação de resultados entre os diferentes tipos de implementações.

6. REFERÊNCIAS

CHELLAPPA, R. **Intermediaries in cloud-computing: A new computing paradigm**, In: INFORMS Dallas 1997 Cluster: Eletronic Commerce, Dallas, Texas, 1997

Valancius, V., Laoutaris, N., Massoulié, L., Diot, C., and Rodriguez, P. (2009). Greening the internet with nano data centers. In CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies, pages 37–48, New York, NY, USA. ACM.

Lefèvre, L. and Orgerie, A.-C. (2010). Designing and evaluating an energy efficient cloud. *The Journal of Supercomputing*, 51:352–373. 10.1007/s11227-010-0414-2

Liu, L., Wang, H., Liu, X., Jin, X., He, W. B., Wang, Q. B., and Chen, Y. (2009). Greencloud: a new architecture for green data center. In ICAC-INDST '09: Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session, pages 29–38, New York, NY, USA. ACM.

Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, *CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms*, Software: Practice and Experience (SPE), Volume 41, Number 1, Pages: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011.

Werner, Jorge. (2011). Uma Abordagem para Alocação de Máquinas Virtuais em Ambientes de Computação em Nuvem Verde. Dissertação de Mestrado. PPGCC- UFSC. 2011.