

Um estudo comparativo de simuladores para sistemas distribuídos

Gabriel F. C. de Queiroz^{1,2}, Rodrigo de Souza Couto¹, Alexandre Sztajnberg¹

¹PEL/DETEL/FEN – Universidade do Estado do Rio de Janeiro (UERJ)
Rua São Francisco Xavier, 524 – 5º andar – Rio de Janeiro – RJ – Brasil

²DETEL/DEPES – Unidade: Maracanã
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)
g_queiroz@ymail.com, rodrigo.couto@uerj.br, alexszt@ime.uerj.br

Abstract. *As the computer networks have become wider and more complex, always dealing with greater information traffic and high computational effort applications, the distributed systems appeared as a way to share the resources needed by different entities and execute tasks in different stations. Many solutions were developed, such as cloud computing, grids, clusters and peer-to-peer networks (P2P). Nevertheless, planning such networks involves greater costs, what makes it more interesting to use simulators to verify its planning decisions. Therefore, this survey addresses to different simulation environments and their features, performing a comparative study of their functionalities.*

Resumo. *Conforme as redes de computadores se tornaram maiores e mais complexas, sempre lidando com maiores tráfegos e aplicações de alto esforço computacional, os sistemas distribuídos surgiram como uma forma de partilhar os recursos necessários por diversas entidades e executar tarefas em diferentes estações. Muitas soluções foram desenvolvidas, como a computação em nuvem, as grades, os clusters e as redes par-a-par (P2P). No entanto, projetar tais redes envolve custos maiores, o que torna interessante a utilização de simuladores para verificar as decisões de planejamento. Sendo assim, este trabalho se endereça a diferentes ambientes de simulação e suas características, realizando um estudo comparativo de suas funcionalidades.*

1. Introdução

Com o desenvolvimento das redes de computadores, diversos equipamentos e aplicações foram criados para atender aos mais diversos requisitos por parte dos usuários e dos administradores de rede. Cada vez mais, a necessidade por taxas de transmissão maiores, recursos de multimídia e processamento impulsionam pesquisas na Engenharia e na Ciência da Computação, de forma a atender as necessidades humanas.

Depois de décadas de evolução constante, os sistemas de computação se tornaram mais acessíveis, baratos e rápidos de tal forma que a criação de sistemas compostos por diversos computadores, ou estações, conectados se tornou possível, nos quais diferentes tarefas podem ser terceirizadas para um subconjunto de estações e evitar que uma única estação fique sobrecarregada por uma aplicação computacionalmente difícil. Tais sistemas são conhecidos como sistemas distribuídos, já que o processamento não se encontra mais centralizado em uma máquina apenas ou

em um processador único, sendo também considerados uma coletânea de computadores independentes que se mostram ao usuário final como um sistema único e coerente [Tanenbaum and Van Steen, 2007].

Dada a importância dos sistemas distribuídos e sua ampla difusão, este artigo se destina ao estudo de diferentes ambientes de simulação disponíveis para este tipo de sistema, como as grades, os *clusters*, as nuvens e as redes P2P. A simulação é uma importante ferramenta de trabalho para analisar o desempenho de um sistema ou projetar uma rede, por exemplo. Como a realização de um experimento real pode ser inviável por questões financeiras, complexidade dos sistemas ou número de variáveis envolvidas, os ambientes de simulação são capazes de fornecer resultados coerentes sem incluir os riscos de uma experimentação. Dessa forma, este estudo reúne parte das principais ferramentas de simulação amplamente disponíveis na *Internet* para analisar suas características de funcionamento suportadas, considerando o tipo de sistema ao qual se destina, e promover uma análise comparativa entre esses simuladores.

Este artigo está estruturado da seguinte forma. A Seção 2 trata dos conceitos básicos acerca dos sistemas distribuídos e as semelhanças e diferença entre grades, *clusters*, nuvens e redes P2P. A Seção 3 apresenta os ambientes de simulação estudados e suas características, enquanto a Seção 4 apresenta trabalhos relacionados que foram desenvolvidos nesta área. A Seção 5 conclui o trabalho e indica direções futuras.

2. Conceitos básicos

Antes de descrever as características dos ambientes de simulação estudados, é importante rever alguns conceitos fundamentais sobre os próprios sistemas distribuídos, sejam estes grades, *clusters*, nuvens ou redes P2P. Embora todos esses sistemas partam do mesmo princípio de processamento distribuído por diferentes nós, eles se diferenciam pelos objetivos envolvidos, a homogeneidade dos nós, as abstrações utilizadas e os modelos de negócio.

Os sistemas distribuídos são formados por diferentes computadores que interagem entre si de maneira colaborativa para desempenhar tarefas computacionais. Cada uma das máquinas desse sistema possui seu próprio núcleo de processamento e espaço de memória, realizando trocas de mensagens entre si através da rede.

Algumas das preocupações mais comuns em sistemas distribuídos envolvem o escalonamento das tarefas [Legrand, Marchal and Casanova, 2003], a concorrência [Killian *et al.*, 2007] e a escalabilidade [Rohloff and Schantz, 2010]. O escalonamento de tarefas é o processo de organização das tarefas a serem realizadas pelo processador, levando em conta aspectos como a prioridade dos processos e justiça, evitando que o processador fique ocioso ou dedicado apenas a uma tarefa. O principal problema decorrente de um escalonamento mal formulado é a inanição (*starvation*) [Tanenbaum and Van Steen, 2007], que ocorre quando um processo nunca consegue ser executado pelo processador e processos concorrentes conseguem acessá-lo frequentemente e possivelmente terminar suas execuções em tempos bastante reduzidos. Já a concorrência é a principal característica de sistemas concorrentes, nos quais diferentes processos competem pelo tempo de execução do processador e precisam lidar com problemas como *deadlock* [Tanenbaum and Van Steen, 2007], que ocorre quando um processo entra em um estado que não o leva a nenhum outro, e a quebra de exclusão mútua, que

ocorre quando dois processos tentam acessar o mesmo recurso e os resultados de suas operações se sobrepõem. Contudo, é importante notar que, embora muitos dos sistemas distribuídos também sejam concorrentes, esses são sistemas distintos, já que é possível haver concorrência apenas com uma máquina e também conceber um sistema distribuído livre de competição. Já a escalabilidade é a capacidade que um sistema tem de se expandir sem que isso incorra no aumento excessivo da complexidade do sistema ou tenha seu desempenho reduzido, de modo que o sistema esteja apto ao crescimento da carga de trabalho ou do número de pares na rede [Tanenbaum and Van Steen, 2007].

Pela sua característica distribuída, esses sistemas podem conter uma quantidade ampla e variável de estações, destacando sua escalabilidade e a disponibilidade dos recursos presentes nas estações, como no caso das redes par-a-par, que podem lidar constantemente com a questão da rotatividade de nós na rede. No entanto, nem todos os sistemas distribuídos apresentam uma quantidade variável de estações participantes, como é o caso dos *clusters* de estações, no qual a adição de um servidor ao grupo não é tão frequente quanto, por exemplo, a adição de um membro a uma rede baseada em BitTorrent [Cohen, 2003].

Dado o exposto, é possível notar que os sistemas distribuídos possuem uma concepção geral em comum, mas também podem se diferenciar em diversos fatores, dependendo do serviço que se pretende oferecer em nível de aplicação. Há sistemas de computação distribuídos voltados para o processamento de tarefas computacionalmente difíceis ou para o provimento ininterrupto de um determinado serviço, como no caso dos *clusters* e grades, que possuem topologias fixas ou que se alteram com pouca frequência. Outro exemplo de sistema distribuído são as nuvens, sendo um assunto de interesse acadêmico e comercial, que tem ganhado atenção nos últimos anos, visto que a computação em nuvem provê serviços e capacidade de processamento em grande escala através da *Internet* [Zhang, Cheng and Boutaba, 2010]. As nuvens também não apresentam mudanças de topologia constantes, já que não é necessário deslocar constantemente os centros de dados pela rede. Porém, nem todos os sistemas distribuídos são pautados em topologias fixas, como no caso das redes P2P, na qual os nós estão distribuídos por uma determinada região e diferentes usuários entram e saem da rede frequentemente. Essa variação de topologia presente nas redes P2P é ainda maior no caso das redes sem fio *ad hoc* [Kurose and Ross, 2009], nas quais os dispositivos estão em constante movimento e não há um ponto de acesso para realizar tarefas de roteamento.

As redes P2P são uma arquitetura de sistemas distribuídos na qual os pares desempenham funções de cliente e servidor, realizando solicitações ou respondendo às requisições de outros pares. Como os nós podem desempenhar ambas as funções, não há um nó central responsável pelo processamento de todas as funções de rede, confirmando a característica distribuída das redes P2P.

Muitos dos objetivos da computação em *cluster*, grade e nuvem são relativamente os mesmos e partem do mesmo princípio, buscando melhorar a capacidade computacional através da utilização de muitos nós.

Sendo assim, é oportuno destacar algumas de suas diferenças antes de conhecer os ambientes de simulação. A Tabela 1 resume a discussão a seguir. Os *clusters* são formados basicamente por máquinas com alto grau de homogeneidade, tanto em termos de *hardware* quanto de sistema operacional e aplicações, as quais tendem a estar

localizadas próximas umas das outras em uma rede local (LAN – *Local Area Network*). Outro aspecto importante dos *clusters* é que as máquinas envolvidas compartilham seus recursos para executar uma tarefa, seja para oferecer alta confiabilidade e tolerância a falhas, no caso da oferta ininterrupta de um recurso, ou para prover maior capacidade computacional na sua resolução. Um dos sistemas de *cluster* mais conhecidos é o do projeto Beowulf, baseado na infraestrutura Linux [Gropp, Lusk and Sterling, 2003].

Tabela 1. Características dos tipos de sistemas distribuídos comparados

Tipo	<i>Cluster</i>	Grade	Nuvem	P2P
Número de nós	Até a faixa de 100 nós	Até a faixa de 1000 nós	Variável, de dezenas a milhares de máquinas virtuais	Variável, depende do tamanho do sistema
Homogeneidade	Alta	Baixa	Baixa	Baixa
Mudança de topologia	Raramente	Pouco frequente	Pouco frequente	Muito frequente
Extensão	Rede local	Rede de longa distância	Rede de longa distância	Variável, pode ir de uma rede de área pessoal até uma rede de longa distância
Escalonamento	Gerenciamento centralizado, único domínio administrativo	Gerenciamento distribuído, vários domínios administrativos	Gerenciamento distribuído	Gerenciamento distribuído
Uso de técnicas de virtualização	Baixo	Baixo	Alto	Médio, redes sobrepostas em nível de aplicação

As Grades Computacionais (*Computer Grids*) são sistema de computação distribuído, com características diferentes dos *clusters*. Podem apresentar um alto grau de heterogeneidade, tanto de *hardware* e *software* quanto de domínios administrativos, se forem orientadas ao aproveitamento de recursos ociosos. Por outro lado, grades para processamento científico podem ser formadas por federações de *clusters*, integrando diferentes entidades, que colaboram entre si para implementar um sistema com amplos recursos de processamento e armazenamento. Neste caso, mesmo havendo heterogeneidade entre os *clusters* federados, existe uma hierarquia administrativa de alocação e escalonamento que organizam o processamento no conjunto de recursos. Dessa forma, os nós em uma grade podem não estar sempre na mesma LAN, geralmente espalhada em área geográfica maior do que os *clusters* [Sadashiv and Kumar, 2011].

Além disso, as máquinas em uma grade podem desempenhar tarefas diferentes, oferecendo compartilhamento de recursos e capacidade de processamento por requisição sob um controle que pode estar descentralizado nas várias organizações administrativas. Outra diferença entre clusters e grades está na ordem de grandeza desses sistemas. Enquanto os clusters podem envolver até centenas de máquinas, as grades podem chegar a milhares de máquinas [Hanamakkanavar and Handur, 2015].

Com relação ao conceito de computação em nuvem, ainda não há um consenso quanto à sua definição, mas ela compartilha das mesmas características das grades, evoluindo a partir destas. No entanto, o contexto econômico e tecnológico no qual as nuvens estão inseridas não é o mesmo que as grades encontraram quando foram propostas, de modo que ambas apresentam diferenças em termos de segurança, modelos de negócio, aplicações e abstrações [Foster et al., 2008]. Embora as grades também promovam o compartilhamento de recursos, as nuvens o fazem através do uso intensivo de técnicas de virtualização, não havendo compartilhamento direto dos recursos. A própria virtualização de recursos é um diferencial, sendo muito mais explorada pelas nuvens do que pelas grades, até mesmo para realizar o provisionamento dinâmico dos serviços [Zhang, Cheng and Boutaba, 2010]. Sendo assim, a computação em nuvem pode ser compreendida como um sistema de computação distribuído de grande escala, no qual serviços e recursos, como armazenamento e capacidade de processamento, são oferecidos, sob demanda, à consumidores externos através da *Internet* [Foster et al., 2008].

3. Ambientes de simulação

Considerando os conceitos básicos acerca dos sistemas distribuídos e os diferentes modelos de sistemas existentes, tais como grades, *clusters*, nuvens e redes P2P, esta seção descreve as características e funcionalidades oferecidas pelos ambientes de simulação disponíveis para estes sistemas.

A utilização de ferramentas de simulação é recorrente em estudos e projetos acadêmicos como forma de validar soluções obtidas ou avaliar o desempenho de uma nova proposta. Essas ferramentas são empregadas porque pode não ser possível a realização do experimento em um sistema real por questões diversas, como a falta de equipamentos para uma implementação ou medição e custo financeiro envolvido.

Sendo assim, os ambientes de simulação são uma ferramenta livre dos riscos de uma implementação real para o pesquisador, pois não envolve custos de equipamento para instalação de uma rede e também não oferece riscos ao funcionamento de uma rede preexistente. No entanto, as ferramentas de simulação também apresentam a desvantagem de ser cenário hipotético, podendo não considerar todos os aspectos de uma implementação real. Entretanto, determinados simuladores podem apontar com certa precisão para um resultado próximo do que se obteria em um experimento real. Além disso, cada simulador apresenta vantagens e desvantagens relacionadas às suas próprias funcionalidades e limitações. Simuladores de eventos discretos, por exemplo, modelam de forma simples a sucessão de eventos no tempo, o que não retrata com realismo a passagem contínua de tempo. Outro *trade-off*, presente nos simuladores estudados, é a relação entre o realismo nas simulações ao modelar, por exemplo, a troca de pacotes na rede e ambos o tempo de execução e o consumo de memória.

As subseções, a seguir, tratam individualmente de cada ambiente de simulação estudado durante esta pesquisa, sendo esses o GridSim [Buyya and Murshed, 2002] [Sulistio et al., 2008], o GroudSim [Ostermann et al., 2010], o CloudSim [Calheiros et al., 2011] [Devi and Sujana, 2014], o GreenCloud [Kliazovich, Bouvry and Khan, 2012], o SimGrid [Casanova, Legrand and Quinson, 2008] [Legrand, Marchal and Casanova, 2003], o PeerSim [Montresor and Jelasity, 2009] [Kazmi and Bukhari, 2011] e o PlanetSim [García et al., 2004] [Ahulló et al., 2008], retratando as características principais, funcionalidades e demais aspectos para cada uma delas, como o tipo de simulador, linguagem de programação, usabilidade e documentação e a modelagem em nível de rede.

3.1 GridSim

O GridSim¹ é um ambiente de simulação de eventos discretos para computação em grade, escrito em Java e desenvolvido pelo laboratório CLOUDS da University of Melbourne, na Austrália.

Um sistema discreto é aquele no qual suas variáveis de estado apenas mudam em pontos discretos no tempo, chamados eventos, cuja sequência cronológica descreve o comportamento do sistema [Ostermann et al., 2010]. Portanto, um simulador de eventos discretos consiste em uma lista de eventos futuros e um algoritmo de avanço de tempo.

O GridSim segue uma abordagem baseada em processos, ou seja, cria uma linha de execução (*thread*) para cada entidade no sistema. Por criar múltiplos processos na simulação, o GridSim apresenta alto consumo de memória e possui dificuldades para simular algumas milhares de grades [Ostermann et al., 2010], reduzindo sua escalabilidade.

O GridSim é um dos principais simuladores de grades utilizados na literatura [Casanova et al., 2014], contando com uma extensa comunidade e ampla documentação. O GridSim permite a simulação e modelagem de sistemas distribuídos, aplicações, recursos e escalonadores. O simulador oferece suporte à modelagem de recursos heterogêneos, como computadores de um ou mais núcleos de processamento e *clusters*, cada elemento com diferentes capacidades e configurações, o que também inclui a opção de ajustar o fuso horário para simulações de grades globalmente distribuídas.

O escalonamento pode ser feito através de uma política de compartilhamento no tempo ou de compartilhamento no espaço. No primeiro, as tarefas são alocadas aos processadores e podem ser executadas concorrentemente em um esquema multitarefa, no qual cada processo é executado durante uma fatia de tempo. No segundo, as tarefas são escalonadas a partir de filas, das quais são dedicadas a um processador que a executa completamente, assim não há duas tarefas alocadas a um processador ao mesmo tempo e elas também não são atribuídas imediatamente a um processador, devendo esperar até que algum deles termine sua execução [Sulistio et al., 2008]. No entanto, não é capaz de oferecer as abstrações de serviços utilizadas pela computação em nuvem, já que não é capaz de modelar ambientes e recursos virtuais.

¹ <http://www.cloudbus.org/gridsim/>

No GridSim, os recursos são máquinas e *clusters*, com seus processadores e memórias, possuindo desempenho e configurações diferentes. Simulações de *clusters* geralmente são feitas em simuladores de grades, como o GridSim, que permite a migração de processos e replicação de dados entre *clusters*. O GridSim também inclui uma política de alocação de *clusters* extensível, conforme mencionado em [Grudenic and Bogunović, 2009].

Segundo [Buyya and Murshed, 2002], as funcionalidades oferecidas pelo GridSim permitem que ele seja utilizado para diferentes sistemas distribuídos além das grades, como os *clusters* e as redes P2P. Por ser um ambiente de simulação de computação em grade, o GridSim também pode ser utilizado para simulações de *clusters*, alocando os recursos sob um mesmo domínio administrativo e adaptando o escalonamento das aplicações. Em *clusters*, o escalonamento é feito de forma a melhorar o desempenho global do sistema, já que os escalonadores nas grades visam melhorar o desempenho das aplicações às quais se referem.

É possível definir valores de qualidade de serviço (*Quality of Service - QoS*) no GridSim, priorizando o envio de pacotes de aplicações com requerimentos de latência e entrega rápida de dados. A QoS é um parâmetro utilizado em redes para diferenciar fluxos de tráfego de aplicações com requerimentos críticos de atraso, *jitter* ou taxa constante, como aplicações de áudio e vídeo [Sulistio et al., 2008].

O GridSim não considera aspectos de rede, como o controle de congestionamento, mas faz uma alusão à fragmentação de fluxo em pacotes que ocorre em redes reais, o que pode torná-lo muito lento [Casanova, Legrand and Quinson, 2008]. Sendo assim, o GridSim também não oferece suporte a padrões de rede sem fio em suas especificações de topologia de rede. Muitos dos simuladores de sistemas distribuídos não introduzem a geração de pacotes em suas funcionalidades, pois a simulação em nível de pacote é computacionalmente intensa e compromete a escalabilidade da simulação, visto que há um compromisso entre uma simulação mais realista e uma escalável [Kliazovich, Bouvry and Khan, 2012].

3.2 GroudSim

O GroudSim² é um simulador de eventos discretos para computação em grade e nuvem voltado para a escalabilidade das simulações obtidas, utilizando apenas uma linha de execução. A ferramenta foi desenvolvida como parte do projeto ASKALON, na University of Innsbruck.

O GroudSim é escrito em Java e também oferece suporte à modelagem de recursos, escalonamento de tarefas e transferência de arquivo. Diferente de simuladores como o GridSim, o GroudSim não segue uma abordagem baseada em processos, que cria uma linha de execução para cada entidade na simulação. Ao invés disso, ele segue uma abordagem baseada em um único processo, o que diminui o consumo de memória necessário às simulações [Ostermann et al., 2010].

A ferramenta pode ser utilizada tanto para grades quanto para nuvens, já que permite a utilização de máquinas virtuais e demais características da computação em nuvem. O GroudSim permite a modelagem de custo e é dedicado principalmente ao modelo IaaS (*Infrastructure as a Service*), modelo de negócio da computação em

² <http://dps.uibk.ac.at/projects/groudsim/>

nuvem no qual o usuário solicita máquinas virtuais, servidores ou demais infraestruturas, mas pode ser estendido para suportar modelos adicionais [Kumar, 2013]. O GroudSim possui dois modelos de escalonamento, um para grades e outro para nuvens. Para grades, ele utiliza um sistema de filas, como o compartilhamento no espaço do GridSim, em que as tarefas esperam até que uma unidade de processamento esteja disponível. Para nuvens, os recursos, ou instâncias, são primeiramente adquiridos, uma política de compartilhamento de recursos é aplicada na chegada de uma tarefa, nenhum mecanismo de fila é empregado [Ostermann et al., 2010].

Com relação à modelagem do nível de rede, o GroudSim não lida com a transmissão de pacotes, assim como grande parte dos simuladores de sistemas distribuídos, mas é capaz de modelar latência e largura de banda. Esse simulador também não considera mecanismos de controle de congestionamento na rede, assumindo que as únicas limitações de largura de banda de cada *host* são relativas apenas à sua conexão de rede [Casanova et al., 2014].

Embora o simulador não conte com uma grande comunidade, o GroudSim é um projeto ativo, o que inclui a adição de novos modelos de energia e de tarifação em versões mais recentes, e está documentado em artigos publicados ao longo dos últimos anos [Ostermann et al. 2010] [Kecskemeti, Ostermann and Prodan, 2014]. O GroudSim está disponível gratuitamente para fins de ensino e pesquisa mediante requisição aos autores do simulador.

3.3 CloudSim

O CloudSim³ é um ambiente de simulação de eventos discretos para computação em nuvem também desenvolvido pelo laboratório CLOUDS. Com o crescimento da computação em nuvem, o projeto desenvolvido se destinava à criação de um ambiente de simulação geral e extensível que oferecesse suporte a modelagem, simulação e experimentação integradas para serviços de nuvem em grande escala.

Pela relação intrínseca entre a computação em nuvem e a computação em grade, o CloudSim foi desenvolvido sobre o simulador GridSim, aproveitando a base de suas implementações e inserindo novas abstrações para adaptar suas funcionalidades à computação em nuvem e seu novo modelo de negócio, já que os ambientes de simulação para sistemas distribuídos, até então disponíveis, não se aplicavam à computação em nuvem, pois não oferecem suporte à simulação de infraestruturas virtualizadas e nem à modelagem de centros de dados [Calheiros et al., 2011].

Por ter sido baseado no GridSim, o CloudSim também é escrito em Java e cria uma linha de execução para cada entidade no sistema, apresentando os mesmos problemas de escalabilidade e consumo de memória. No entanto, as versões mais recentes do CloudSim deixaram de utilizar o SimJava, uma estrutura com bibliotecas para a definição e execução de simulações, passando a utilizar uma estrutura própria. Como o SimJava cria múltiplas linhas de execução e insere alta sobrecarga (*overhead*) devido às trocas de mensagens entre cada linha, o uso de memória é alto e prejudica a escalabilidade do sistema. Dessa forma, as últimas versões do CloudSim apresentam melhorias quanto ao *overhead* e à escalabilidade [Calheiros et al., 2011].

³ <http://www.cloudbus.org/cloudsim/>

Como a computação em nuvem é um tópico de bastante interesse acadêmico e comercial, o CloudSim possui uma extensa comunidade de suporte e a documentação pode ser facilmente encontrada na *Internet*, havendo tutoriais, cursos e exemplos disponíveis [Calheiros et al., 2011].

Dentre as principais funcionalidades oferecidas pelo CloudSim, é possível citar o suporte à modelagem e simulação de ambientes de nuvem em grande escala, incluindo centros de dados, máquinas virtuais e corretores de serviço (*brokers*), além de políticas de escalonamento e alocação de recursos e tarefas. Também é possível utilizar o CloudSim para simulações de nuvens que lidam com recursos de domínio público e privado e balanceamento de carga. Assim como no GridSim, é possível utilizar tanto o compartilhamento no tempo quanto o compartilhamento no espaço para realizar a alocação dos processadores aos serviços virtualizados.

Segundo [Calheiros et al., 2011], o CloudSim suporta o provisionamento de máquinas virtuais em dois níveis: primeiro, no nível do *host*; segundo, no nível de VM. No nível de *host*, é possível especificar quanto do poder de processamento geral de cada núcleo vai ser alocado a cada VM. No nível de VM, as aplicações recebem e utilizam uma parcela fixa do poder de processamento atribuído à VM. As VMs são alocadas aos *hosts* por ordem de chegada, mas é possível modelar e simular outras políticas de alocação.

O CloudSim oferece suporte aos três principais modelos de negócio da computação em nuvem: IaaS, PaaS (*Platform as a Service*) e SaaS (*Software as a Service*). Enquanto o primeiro modelo oferece acesso a VMs e armazenamento, o segundo fornece ambientes computacionais para o desenvolvimento de aplicações e o terceiro provê acesso a aplicações, como editores de texto e correio eletrônico, instaladas nos centros de dados da nuvem através da *Internet*. Sendo assim, o CloudSim suporta a modelagem de custos para cada modelo de negócio, como uso de memória, armazenamento, largura de banda e requisições de aplicação.

Apesar da computação em nuvem estar associada à *Internet*, o CloudSim não apresenta componentes de rede, como roteadores e comutadores, de forma que as comunicações de rede são feitas a partir de abstrações. Por exemplo, a latência da rede pode ser simulada com base nas informações de uma matriz de latência, usada para simular o atraso na transmissão de mensagens entre duas entidades em unidades de tempo de simulação, como milissegundos. Além da latência, o CloudSim também é capaz de incluir informações de largura de banda.

Versões mais recentes do CloudSim ainda incluem modelos de consumo de energia para simulação de ambientes nos quais a energia é um fator limitante e suporte a um novo modelo de negócio, conhecido como CaaS (*Container as a Service*), que fornece ambientes de computação isolados para as aplicações implementadas, situando-se entre o IaaS e o PaaS.

3.4 GreenCloud

O GreenCloud⁴ é um simulador de computação em nuvem sofisticado que modela a transmissão de pacotes na rede e foi desenvolvido para simulações de centros de dados cientes dos níveis de energia. Ele oferece uma modelagem detalhada do consumo de

⁴ <https://greencloud.gforge.uni.lu/>

energia pelos servidores, comutadores de rede e enlaces de comunicação. O GreenCloud foi desenvolvido como parte dos projetos GreenIT e ECO-CLOUD, na University of Luxembourg.

A falta de simuladores detalhados e sem qualquer sistema de provisionamento para a análise da eficiência de energia de nuvens foi a motivação por trás do desenvolvimento do GreenCloud para interagir e medir o desempenho das nuvens [Alam, Pandey and Rautaray, 2015], embora versões mais recentes do CloudSim também se enderecem à questão da eficiência em energia.

O GreenCloud é considerado uma extensão do NS-2⁵, um simulador de rede escrito em C++ e OTcl [Meenaghan and Delaney, 2004] com suporte a diferentes protocolos, ambientes cabeados e sem fio, topologias e roteamento de pacotes. Como o NS-2 simula a transmissão de pacotes, o GreenCloud possui modelos muito mais precisos quanto à comunicação na rede e tem suporte completo à arquitetura TCP/IP, com pacotes e cabeçalhos. A maioria dos ambientes de simulação para sistemas distribuídos opta por não modelar a transmissão de pacotes devido ao aumento no tempo de simulação, utilizando abstrações e modelos mais simples para considerar apenas, por exemplo, largura de banda e latência. O GreenCloud foi projetado para capturar os detalhes do consumo de energia pelos componentes de um centro de dados e também os padrões de comunicação em nível de pacote entre eles. Sendo assim, o GreenCloud possui tempos de simulação maiores e menor escalabilidade em comparação aos demais simuladores de nuvem, ao passo que apresenta maior precisão em suas simulações por permitir a modelagem da rede [Kliazovich, Bouvry and Khan, 2012].

O GreenCloud é um projeto ativo e está disponível gratuitamente na *Internet* e também possui uma documentação detalhada, além de estarem disponíveis tutoriais, manuais de usuário e guias para desenvolvedores na página do simulador na *Internet*. Embora não conte com uma comunidade tão grande quanto a do CloudSim, o GreenCloud se faz bastante presente em publicações na sua área, haja vista o crescente interesse acadêmico tanto na computação em nuvem quanto em tecnologias verdes e ecologicamente viáveis.

Diferente do CloudSim, o GreenCloud oferece apenas políticas de escalonamento simples e servidores com apenas um núcleo de processamento, ou seja, não é possível simular sistemas multiprocessados.

3.5 SimGrid

O SimGrid⁶ é um ambiente de simulação de eventos discretos para escalonamento de aplicações baseado em C, inicialmente desenvolvido pelo grupo de pesquisa AppLeS na University of California. Hoje, conta com o apoio de diferentes instituições de ensino e centros de pesquisa ao redor do mundo, principalmente na França. O SimGrid permite a simulação de aplicações em sistemas distribuídos de grande escala, como grades, *clusters* e redes P2P. Sendo bastante versátil, o SimGrid também inclui algumas funcionalidades de um simulador de rede, oferecendo suporte a redes locais e de longas

⁵ <http://www.isi.edu/nsnam/ns/>

⁶ <http://simgrid.gforge.inria.fr/>

distâncias. O SimGrid também pode ser utilizado como um simulador para computação em nuvem e permite a criação de máquinas virtuais e alocação de processos a elas, porém esse suporte ainda se encontra em fase experimental.

Apesar de ser um simulador que utilize a linguagem C em seu núcleo, o SimGrid possui vários *bindings*, ligações que permitem escrever a simulação em C ou Java, por exemplo. Em uma recente atualização do simulador, grande parte do código em C foi reescrito em C++, atualizando a tecnologia, permitindo a compatibilidade com outros ambientes orientados a objetos e aprimorando a estruturação orientada a objetos, o que torna o sistema interno mais modular.

O SimGrid é um projeto ativo e está disponível gratuitamente na rede. O simulador conta com uma extensa documentação na *Internet*, incluindo guias de usuário e tutoriais, além de uma grande comunidade envolvida em seu uso e desenvolvimento [Xu et al., 2011].

O SimGrid é um simulador de eventos capaz de simular a concorrência de processos e modelar não apenas os elementos da computação distribuída, como servidores e políticas de escalonamento, mas também a rede em nível de pacote, simulando fluxos de pacotes, embora não crie pacotes como os simuladores de rede. Tais modelos são fundamentais devido ao longo tempo de simulação necessário aos simuladores de rede, melhorando a escalabilidade e reduzindo a precisão das simulações [Casanova et al., 2014].

O SimGrid não inclui mecanismos reais de controle de congestionamento, como os do protocolo de transporte (TCP), e também não considera o descarte de pacotes quando as filas dos roteadores estão cheias, levando a resultados mais otimistas e, portanto, menos precisos, para diminuir os tempos de simulação [Casanova, Legrand and Quinson, 2008]. Para lidar com a transmissão de pacotes, o SimGrid modela os pacotes individuais de uma comunicação fim-a-fim como um fluxo, que é caracterizado por sua taxa de transferência ou largura de banda. Essa largura de banda depende da topologia da rede e da interação com os demais fluxos de rede, com os quais compete pelo meio. No entanto, o SimGrid considera que os fluxos estão sempre em estado estacionário, desconsiderando, por exemplo, o estado inicial de transmissões TCP com o algoritmo de partida lenta, no qual as transmissões começam com o envio de apenas um pacote. Para simulações feitas com pacotes maiores, esse efeito se torna pouco relevante, pois a fração de tempo relativa a esse curto período é muito menor do que a relativa ao estado estacionário [Casanova et al., 2014].

Em [Velho and Legrand, 2009] e [Fujiwara and Casanova, 2007], são propostas melhorias ao modelo de fluxo de pacotes, permitindo, por exemplo, a simulação de parte do algoritmo de partida lenta em conexões TCP e sobrecarga de protocolo. Essas melhorias ajudaram a tornarem a ferramenta bastante versátil, aproximando a precisão de suas simulações dos resultados obtidos com simuladores de rede, mas ainda mantendo os tempos de simulação reduzidos.

Ainda segundo os estudos de [Casanova et al., 2014], a proposta de um simulador abrangente, envolvendo simulações de grades, *clusters* e redes P2P, é capaz de produzir resultados até melhores do que os obtidos com simuladores específicos. O foco na versatilidade levou o SimGrid a investir tanto na escalabilidade de suas simulações quanto na precisão. Sendo assim, as simulações no SimGrid apresentaram

melhor escalabilidade do que as do GridSim, obtendo tempos de simulação mais baixos e sendo mais eficiente em termos de utilização de memória [Casanova et al., 2014].

Com relação às redes P2P, o SimGrid permite simulações altamente escaláveis, com alguns milhões de nós em uma única máquina, considerando latência e largura de banda e também a rotatividade dos nós na rede, mas ainda não há considerações acerca do consumo de energia nas simulações. O SimGrid é capaz de realizar execuções em paralelo sem diminuir a escalabilidade, sendo mais escalável que os mais conhecidos simuladores de redes P2P [Quinson, Rosa and Thiéry, 2012].

3.6 PeerSim

O PeerSim⁷ é um ambiente de simulação para redes P2P escrito em Java e desenvolvido inicialmente como parte dos projetos BISON e DELIS. O principal foco do simulador é a escalabilidade de suas simulações. Dessa forma, o PeerSim não se endereça à modelagem da infraestrutura de rede e nem à transmissão de pacotes, utilizando modelos de simulação simples e escaláveis.

O PeerSim possui dois modos de simulação, um baseado em eventos discretos e um baseado em ciclos. Na simulação baseada em ciclos, os pares que formam a rede são visitados uma vez por ciclo, seguindo uma ordem sequencial ou aleatória, e nesta oportunidade eles podem enviar ou receber mensagens. Com isso, o modo baseado em ciclos possui maior escalabilidade que o modo de eventos discretos, pois não lida com a complexidade computacional inserida ao manter uma fila de eventos e escaloná-los para sua execução, como no caso da simulação de eventos discretos. No entanto, a abordagem do modo baseado em ciclos não oferece o mesmo nível de realismo que o outro método proporciona, já que um par que nada tem a enviar ou receber não precisa ser consultado. Além disso, apenas o modo de eventos discretos oferece suporte à concorrência de processos [Ebrahim, Khan and Mohani, 2014].

Além disso, o PeerSim permite a simulação de redes de sobreposição estruturadas e não estruturadas. Uma rede de sobreposição é aquela que possui um conjunto de pares e suas conexões em nível lógico, construída sobre a infraestrutura física, cujos enlaces e processo de roteamento são transparentes para a rede sobreposta. Nas redes estruturadas, são utilizados algoritmos que definem a organização dos pares e recursos disponíveis na rede, o que não ocorre nas redes não estruturadas, nas quais as conexões dos pares e o armazenamento dos dados acontecem arbitrariamente, já que um par pode ingressar na rede e apenas copiar as conexões de um vizinho.

O PeerSim oferece suporte à simulação de redes dinâmicas, permitindo a rotatividade de pares através do ingresso, egresso e falha dos mesmos. No modo baseado em ciclos, é possível alterar a quantidade de nós na rede entre um ciclo e outro [Xu et al., 2011]. As características dos componentes do PeerSim podem ser estendidas e adaptadas pelos pesquisadores. No entanto, não é possível realizar simulações distribuídas no PeerSim.

O PeerSim é um dos simuladores de redes P2P mais utilizados na comunidade e oferece alguns tutoriais, exemplos e ampla documentação para o modo baseado em

⁷ <http://peersim.sourceforge.net/>

ciclos [Naicken et al., 2006]. No entanto, de acordo com [Ebrahim, Khan and Mohani, 2014], destaca-se a falta de material referente ao modo de eventos discretos, não sendo intuitiva a sua utilização. O simulador faz parte de um projeto ativo e está disponível na *Internet*, porém sua comunidade é menos ativa e não há uma versão nova do simulador lançada nos últimos anos.

Pela simplicidade no modelo de simulação do PeerSim, as informações de rede não estão presentes, apenas as abstrações da rede de sobreposição estão incluídas. O simulador não modela a rede subjacente, ou seja, não há representação de cabeçalhos, comportamento de protocolos TCP/IP e demais detalhes. É possível aplicar atrasos que simulem o efeito da latência, mas não é possível configurar a largura de banda. As mensagens trocadas no PeerSim não inserem a carga adicional dos cabeçalhos dos protocolos de rede. Sendo assim, é preciso considerar se essas informações são necessárias ao estudo realizado ao escolher o PeerSim.

3.7 PlanetSim

O PlanetSim⁸ é um simulador de eventos discretos para redes P2P escrito em Java e desenvolvido no projeto Planet, da Rovira i Virgili University. Assim como o PeerSim, o PlanetSim também visa a escalabilidade de suas simulações. O simulador também oferece suporte a redes estruturadas e não estruturadas.

Ao contrário do PeerSim, o PlanetSim permite a realização de simulações distribuídas. O PlanetSim também apresenta um modo baseado em ciclos e, dentro de cada ciclo, todos os processos são executados separadamente, viabilizando a simulação distribuída [Quinson, Rosa and Thiéry, 2012].

O PlanetSim não possui atualizações ou versões novas introduzidas nos últimos anos, porém suas funcionalidades estão amplamente documentadas e disponíveis na *Internet*, incluindo um extenso manual para usuários e desenvolvedores [Naicken et al., 2006].

De acordo com [García et al., 2004], o PlanetSim utiliza uma abordagem estruturada em sua arquitetura, que é composta de três camadas independentes. A camada mais básica é a camada de rede, que define a troca de mensagens entre pares na rede subjacente e pode ser usada para modelar latência, custos e topologia, embora a versão disponível do simulador apresente apenas topologias simples de rede, como a rede em anel, e nenhum modelo de latência. A camada intermediária é a camada de sobreposição, na qual são definidos os algoritmos das redes estruturadas e o comportamento dos pares da rede para envio e recebimento de mensagens, bem como as conexões na rede de sobreposição. A camada mais alta é a camada de aplicação, que permite a modelagem de aplicações e serviços a serem executadas em cada um dos pares.

Ainda com relação à camada de sobreposição, é possível recorrer à maneira tradicional de definir os protocolos a partir de algoritmos para reger a comunicação entre os pares, mas o PlanetSim também permite a implementação de protocolos baseada em comportamento. Essa abordagem permite que o pesquisador associe cada ação que um par pode tomar em diferentes comportamentos e definir quando tais comportamentos se aplicam [Ahulló et al., 2008]. Por exemplo, é possível utilizar a

⁸ <http://ants.etse.urv.es/planet/planetsim/>

definição de comportamentos para realizar a simulação de redes P2P cujos pares podem ser cooperativos e distribuir mensagens ou maliciosos e apenas recebê-las de seus vizinhos.

Similarmente, também é possível representar o comportamento malicioso de pares em uma rede P2P no PeerSim, apesar de que é necessário realizar algumas extensões no simulador, conforme o estudo de [Oliveira et al., 2015] para distribuição de partes (*chunks*) de arquivos de vídeo em um sistema P2P.

Com relação à modelagem do nível de rede, o PlanetSim oferece algum suporte à simulação da infraestrutura de rede subjacente, o que não ocorre no PeerSim, que modela apenas a rede de sobreposição. Por isso, em [Xu et al., 2011], destaca-se que o PeerSim é mais escalável que o PlanetSim, demonstrando o compromisso entre a escalabilidade e o nível de realismo das simulações.

3.8 Resumo

As Tabelas 2 e 3 consolidam as principais características dos simuladores comparados, permitindo uma referência rápida aos mesmos.

A Tabela 2 apresenta as características gerais, incluindo o tipo de sistema-alvo, o tipo de simulador e a linguagem utilizada para descrever a simulação. Ainda estão elencadas características sobre a disponibilidade, uso e o suporte de comunidade de desenvolvedores.

Tabela 2. Características gerais dos simuladores comparados

Simulador	Tipo de sistema	Tipo de simulador	Linguagem	Disponibilidade e usabilidade
GridSim	Cluster e Grade	Eventos discretos	Java	Código aberto; documentação extensa; comunidade numerosa e ativa
GroudSim	Grade e Nuvem	Eventos discretos	Java	Código aberto (requer solicitação); documentação extensa; comunidade não muito numerosa, porém ativa
CloudSim	Nuvem	Eventos discretos	Java	Código aberto; documentação extensa; comunidade numerosa e ativa
GreenCloud	Nuvem	Pacote (baseado no ns-2)	C++/OTcl	Código aberto; documentação extensa; comunidade não muito numerosa, porém ativa

SimGrid	<i>Cluster, Grade, P2P</i>	Eventos discretos	C/C++ e Java	Código aberto; documentação extensa; comunidade numerosa e bastante ativa
PeerSim	P2P	Eventos discretos e baseado em ciclos	Java	Código aberto; documentação extensa (exceto para o modo de eventos discretos); comunidade numerosa e ativa
PlanetSim	P2P	Eventos discretos	Java	Código aberto; documentação extensa; comunidade não muito numerosa e ativa

A Tabela 3 apresenta características mais específicas dos simuladores avaliados. Os modelos de redes suportados vão desde um mecanismo de comunicação abstrato, ou por eventos, onde é apenas possível configurar a latência e a banda disponível, até a modelagem completa da troca de pacotes da arquitetura TCP/IP. O modelo de custo inclui a possibilidade de tarifação ou a possibilidade de configurar modelos de negócios, importante para o caso dos simuladores de nuvens, nos quais o custo dos serviços oferecidos é parte do modelo. Os simuladores também são avaliados quanto ao uso de modelos para avaliar o desempenho energético e, por último, características julgadas relevantes agrupadas no item detalhes adicionais.

Tabela 3. Características específicas dos simuladores comparados

Simulador	Modelo de rede	Modelo de custo	Modelo de energia	Detalhes adicionais
GridSim	Abstrações para latência e banda	Não	Não	Alto consumo de memória; baixa escalabilidade; não suporta virtualização; diferentes políticas de escalonamento; concorrência
GroudSim	Abstrações para latência e banda	Sim (IaaS)	Não	Uma única <i>thread</i> ; baixo consumo de memória; escalabilidade média; políticas simples de escalonamento
CloudSim	Abstrações para latência e banda	Sim (IaaS, PaaS, SaaS e CaaS)	Sim	Melhorias quanto ao <i>overhead</i> e à escalabilidade em relação ao GridSim; nuvens com recursos públicos e privados; diferentes políticas de escalonamento

GreenCloud	Troca de pacotes e TCP/IP (sobre o simulador de rede ns-2)	Não	Sim	Modo de economia de energia; alta precisão e realismo nas simulações; escalabilidade baixa
SimGrid	Modela fluxos de pacote, latência, banda, redes locais e de longa distância, comportamento do TCP etc.	Parcial	Não	Bastante versátil; consumo de memória reduzido; suporte à concorrência; suporte à execução paralela; escalabilidade média; interface para o simulador de rede ns-3
PeerSim	Abstrações para latência	Não	Não	Redes dinâmicas; suporte à concorrência no modo de eventos discretos; não há suporte para simulações distribuídas; escalabilidade muito alta
PlanetSim	Simulação limitada da troca de pacotes na rede	Sim	Não	Simulação modular e independente de rede, sobreposição e aplicações; suporte para simulações distribuídas; escalabilidade alta

4. Outros simuladores

Como a área de sistemas distribuídos é aberta a muitos temas de pesquisa, em particular pela computação em nuvem, diversos artigos já se endereçaram a tais questões, fazendo uso de diferentes ambientes de simulação para verificar resultados propostos ou realizando estudos sobre essas ferramentas.

Em [Buyya and Murshed, 2002], o GridSim, um ambiente de simulação para computação em grade e *clusters*, foi proposto como forma de simular e modelar o gerenciamento de recursos distribuídos. Em [Ostermann et al., 2010], é apresentado o simulador GroudSim, um ambiente de simulação para grades e nuvens, descrevendo suas funcionalidades e destacando a escalabilidade de suas simulações baseadas em eventos.

Já em [Devi and Sujun, 2014], são apresentadas diversas formas de se usar a ferramenta CloudSim em trabalhos de pesquisa sobre computação em nuvem, destacando-se que a maioria das pesquisas o utiliza como ferramenta de simulação em trabalhos sobre provimento de recursos em nuvem. A computação em nuvem também é estudada em [Kumar, 2013] e em [Zhao et al., 2012], nos quais se descrevem alguns ambientes de simulação para nuvens, como o CloudSim, o GreenCloud e o GroudSim. Em [Alam, Pandey and Rautaray, 2015], esses simuladores também são estudados,

incluindo não apenas estudos sobre ferramentas de simulação, mas também serviços, aplicações e demais assuntos relacionados à computação em nuvem.

Em [Casanova, Legrand and Quinson, 2008], o simulador SimGrid é apresentado, ressaltando sua versatilidade por ser um ambiente de simulação para redes P2P, grades e *clusters*. Com relação às redes P2P, pesquisas sobre simuladores de redes par-a-par são apresentadas em [Naicken et al., 2006] e em [Zhou, Zhou and Luo, 2013], realizando um estudo comparativo de diversos simuladores, como o PeerSim e o PlanetSim.

Além dos simuladores estudados neste artigo, há outros ambientes de simulação presentes na literatura, embora menos utilizados ou ainda relativamente recentes [Malhotra and Jain, 2013], como o DCSim para a simulação de centros de dados no modelo IaaS para computação em nuvem. Há ainda simuladores que se originaram de alguns ambientes descritos neste artigo, como o CloudAnalyst, o EMUSIM, o NetworkCloudSim e o MDCSim [Alam, Pandey and Rautaray, 2015] [Ettikyala and Devi, 2015] [Kumar and Rai, 2014], que são extensões do CloudSim para, dentre outras modificações, inclusão de interface gráfica, emulação, aplicações de alto desempenho e características de *hardware* de centros de dados, respectivamente. Com relação às simulações de nuvens cientes dos níveis de energia, a eficiência de energia é tratada em [Sohrabi and Moser, 2015], que inclui os simuladores GreenCloud, CloudSim, MDCSim e GSSIM. Outros simuladores para computação em nuvem, como o iCanCloud, MR-CloudSim, SmartSim, SimIC, CDOSim, TeachCloud e SPECI são estudados em [Ahmed and Sabyasachi, 2014] [Malhotra and Jain, 2013].

Além do GridSim, do GroudSim e do SimGrid, os demais simuladores de grades encontrados na literatura são opções que não conseguiram manter uma comunidade ativa de usuários ao longo do tempo [Casanova et al., 2014]. Outros simuladores de grades menos utilizados são o OptorSim [Bell et al., 2002] [Bell et al., 2003] [Cameron et al., 2004], o ChicSim [Ranganathan and Foster, 2002], o DGSim [Iosup, Sonmez and Epema, 2008], o MicroGrid [Xia et al., 2004], o GangSim [Dumitrescu and Foster, 2005] e o Beosim [Jones and Pang, 2003] [Jones et al., 2004].

Com relação às redes P2P, também há uma grande quantidade de simuladores disponíveis, embora muitos deles apresentem problemas relacionados à baixa escalabilidade, como o P2PSim, o GPS e o PeerThing [Ebrahim, Khan and Mohani, 2014], ou até mesmo devido a uma documentação precária, seja pelos poucos comentários nos códigos-fonte ou pela falta de detalhes em manuais de usuário, sendo esse o caso de simuladores como o NeuroGrid e o Query Cycle, respectivamente [Ebrahim, Khan and Mohani, 2014]. Dentre outros simuladores de redes P2P, estão incluídos o OverSim, o Agent, o D-P2PSim, o DHTSim, o Brite, o Narses, o P2PRealm, o P2PAM, o ChunkSim e o Overlay Weaver [Kazmi and Bukhari, 2011] [Naicken et al., 2006] [Xu et al., 2011].

Embora os simuladores de distribuídos não necessariamente simulem a troca de pacotes em nível de rede, existem simuladores específicos para tais funções. O ns-2 e o ns-3 [Henderson, Lamage and Riley, 2008] são dois dos principais simuladores de rede utilizados pela comunidade. O OMNeT++ também é um simulador para redes de comunicação bastante utilizado [Varga and Hornig, 2008]. Esses simuladores de rede permitem a criação de diferentes tipos de rede, como as de comunicação cabeada ou sem-fio, sendo essa em modo infraestruturado ou ad hoc. Por isso, esses ambientes

podem ser utilizados para simulações de redes tanto no modelo cliente/servidor quanto no P2P [Kazmi and Bukhari, 2011], embora não sejam simuladores criados especificamente para redes P2P, como o PeerSim e o PlanetSim.

5. Considerações finais

Tendo em vista a crescente necessidade de recursos de *hardware* e *software* para lidar com aplicações computacionalmente intensas e com altos requisitos de banda e latência, os sistemas distribuídos representam uma área de interesse acadêmico e comercial para se endereçar a tais questões. Como é possível que a realização da experimentação prática seja inviável na maioria dos casos, dados os altos custos envolvidos e a própria estrutura distribuída dos pares em uma rede P2P e dos centros de dados e servidores de nuvens, grades e *clusters*, o uso de ferramentas de simulação tem se mostrado um dos principais meios para testar e validar projetos.

Neste trabalho, foram listados diferentes ambientes de simulação para sistemas distribuídos amplamente difundidos na literatura, livres para uso e disponíveis gratuitamente. Com relação aos *clusters*, é possível constatar que, em geral, os simuladores destinados à computação em grade, como o GridSim e o SimGrid, também podem ser utilizados para a simulação dos mesmos.

Para a abordagem de computação em grade, foram analisados tanto o GridSim e o SimGrid quanto o GroudSim. É possível concluir que a escolha de um desses simuladores deve observar principalmente duas características, a escalabilidade da rede que se pretende simular e a relevância das informações relativas à pilha de protocolos TCP/IP. Para experimentos cujo realismo das simulações em nível de pacote sejam relevantes, o SimGrid e seu modelo de fluxos de pacotes se tornam mais indicados. Caso seja necessário alcançar soluções mais escaláveis, tanto o baixo consumo de memória quanto a abordagem baseada em uma única linha de execução do GroudSim se tornam mais indicados. No entanto, é importante ressaltar que o GridSim ainda é o principal simulador de grades utilizado dentre os trabalhos referentes à simulação de grades [Casanova et al., 2014] [Hao, Liu and Wen, 2012].

Com relação à computação em nuvem, foram estudados os simuladores GroudSim, CloudSim e GreenCloud. Como a computação em nuvem constitui um tema de interesse elevado, é viável concluir o motivo de o CloudSim, dentre todos os simuladores pesquisados, ser o que possui a maior comunidade e alta frequência de atualização para novas versões. Novamente, a escolha do simulador de nuvem deve se adequar aos objetivos da pesquisa, sendo necessário observar três características: o modelo de negócios, a escalabilidade e o nível de realismo necessário. O CloudSim é o que apresenta melhores resultados no aspecto geral, já que os problemas de escalabilidade, aos quais o GroudSim se endereça, foram tratados com a mudança da estrutura SimJava. Além disso, o CloudSim é o único que se endereça a todos os principais modelos de negócio da computação em nuvem, IaaS, PaaS e SaaS, perdendo apenas para o GreenCloud em termos de precisão e realismo da simulação em nível de pacote. O foco do GroudSim é apenas para simulações de IaaS, sendo necessário expandi-lo para tratar dos demais modelos, porém seu ponto forte é a escalabilidade de suas soluções, sendo indicado também a projetos que integrem o uso de grades e nuvens, já que o simulador se endereça a ambas. Embora o GreenCloud tenha surgido

com o objetivo de tratar de questões ambientais e do consumo de energia, o CloudSim também passou a incluir tais características, de forma que o principal diferencial do GreenCloud é o seu suporte completo à pilha de protocolos TCP/IP e a modelagem real da transmissão de pacotes na rede, sendo importante considerar que esse tipo de simulação é computacionalmente mais intensa, o que reduz a escalabilidade para conferir precisão aos resultados.

Os ambientes de simulação estudados para redes P2P foram o SimGrid, o PeerSim e o PlanetSim. Embora o PeerSim seja considerado o principal simulador de estado da arte para redes P2P [Casanova et al., 2014], o SimGrid também é capaz de fornecer simulações altamente escaláveis e com a vantagem de se endereçar à modelagem da rede em nível de pacote. Para os simuladores de redes P2P, é importante considerar, novamente, a escalabilidade das simulações e a relevância da infraestrutura de rede subjacente, mas também a concorrência de processos e as simulações distribuídas. O PeerSim, no geral, é o simulador de redes P2P mais completo, pois apresenta alta escalabilidade em seu modo baseado em ciclos e confere maior realismo e suporte à concorrência no modo de eventos discretos, porém não é capaz de realizar simulações distribuídas. O SimGrid possui o diferencial da versatilidade, provendo alta escalabilidade, embora menor que o modo baseado em ciclos do PeerSim, e maior realismo e precisão em suas simulações, já que faz uso de seu modelo de fluxos de pacotes na rede subjacente, além de também ser capaz de simular a concorrência entre processos. O PlanetSim se destaca por ser capaz de simular o processamento distribuído realizado em cada par localmente no seu modo baseado em ciclos, além de possuir uma estrutura modular que permite simular apenas aplicações, algoritmos de sobreposição e topologias de rede. O PlanetSim permite a simulação da camada de rede, diferentemente do PeerSim, mas a modelagem em nível de rede do PlanetSim ainda é mais simples que do que a encontrada no SimGrid.

6. Referências

- AHMED, Arif; SABYASACHI, Abadhan Saumya. Cloud computing simulators: A detailed survey and future direction. In: **Advance Computing Conference (IACC), 2014 IEEE International**. IEEE, 2014. p. 866-872.
- AHULLÓ, Jordi Pujol; LÓPEZ, Pedro García. PlanetSim: An extensible framework for overlay network and services simulations. Em: **Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops**. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008. p. 10.
- ALAM, Md Imran; PANDEY, Manjusha; RAUTARAY, Siddharth S. A comprehensive survey on cloud computing. **International Journal of Information Technology and Computer Science (IJITCS)**, v. 7, n. 2, p. 68-79, 2015.
- BELL, William H. et al. Optorsim: A grid simulator for studying dynamic data replication strategies. **The International Journal of High Performance Computing Applications**, v. 17, n. 4, p. 403-416, 2003.
- BELL, William H. et al. Simulation of dynamic grid replication strategies in optorsim. **International Workshop on Grid Computing**. Springer, 2002. p. 46-57.

- BUYA, Rajkumar; MURSHED, Manzur. Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. **Concurrency and Computation: Practice and Experience**, v. 14, n. 13-15, p. 1175-1220, 2002.
- CALHEIROS, Rodrigo N. et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. **Software: Practice and Experience**, v. 41, n. 1, p. 23-50, 2011.
- CAMERON, David G. et al. OptorSim: a grid simulator for replica optimisation. **UK e-science all hands conference**. p. 1-2, 2004.
- CASANOVA, Henri; LEGRAND, Arnaud; QUINSON, Martin. Simgrid: A generic framework for large-scale distributed experiments. **Tenth Intl. Confr. on Computer Modeling and Simulation, 2008. UKSIM 2008**. IEEE, 2008. p. 126-131.
- CASANOVA, Henri et al. Versatile, scalable, and accurate simulation of distributed applications and platforms. **Journal of Parallel and Distributed Computing**, v. 74, n. 10, p. 2899-2917, 2014.
- COHEN, Bram. Incentives build robustness in BitTorrent. **Workshop on Economics of Peer-to-Peer systems**. 2003. p. 68-72.
- DEVI, R. K.; SUJAN, S. A Survey on Application of Cloudsim Toolkit in Cloud Computing. **International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)**, v. 3, n. 6, p. 13146-13153, 2014.
- DUMITRESCU, Catalin L.; FOSTER, Ian. GangSim: a simulator for grid scheduling studies. In: **Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on**. IEEE, 2005. p. 1151-1158.
- EBRAHIM, Mansoor; KHAN, Shujaat; MOHANI, Syed Sheraz Ul Hasan. Peer-to-Peer Network Simulators: an Analytical Review. 2014.
- ETTIKYALA, Kalpana; DEVI, Y. Rama. A study on cloud simulation tools. **International Journal of Computer Applications**, v. 115, n. 14, 2015. p. 18-21.
- FOSTER, Ian et al. Cloud computing and grid computing 360-degree compared. Em: **Grid Computing Environments Workshop, 2008. GCE'08**. Ieee, 2008. p. 1-10.
- FUJIWARA, Kayo; CASANOVA, Henri. Speed and accuracy of network simulation in the simgrid framework. Em: **Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools**. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007. p. 12.
- GARCÍA, Pedro et al. **Planetsim: a New Overlay Network Simulation Framework**. Springer Berlin Heidelberg, 2004.
- GROPP, William; LUSK, Ewing; STERLING, Thomas. Beowulf Cluster Computing with Linux, 2003.
- GRUDENIĆ, Igor; BOGUNOVIĆ, Nikola. Computer Cluster and Grid Simulator. Em: **MIPRO 2009**. 2009.

- HAO, Yongsheng; LIU, Guanfeng; WEN, Na. An enhanced load balancing mechanism based on deadline control on GridSim. **Future Generation Computer Systems**, v. 28, n. 4, p. 657-665, 2012.
- HANAMAKKANAVAR, Amit S.; HANDUR, Vidya S. Load Balancing in Distributed Systems: A survey. **International Journal of Emerging Technology in Computer Science & Electronics**, v. 14, n. 2, p. 961-966, 2015.
- HENDERSON, Thomas R.; LACAGE, Mathieu; RILEY, George F. Network simulations with the ns-3 simulator. **SIGCOMM demonstration**, v. 14, n. 14, p. 527, 2008.
- IOSUP, Alexandru; SONMEZ, Ozan; EPEMA, Dick. DGSim: Comparing grid resource management architectures through trace-based simulation. **European Conference on Parallel Processing**. Springer, Berlin, Heidelberg, 2008. p. 13-25.
- JONES, William M.; PANG, Louis W. Beowulf Mini-grid Scheduling. **Удаленный ресурс CiteSeer**. – Режим доступа: <http://citeseer.ist.psu.edu/696342.html> –25 P, 2003.
- JONES, William M. et al. Job communication characterization and its impact on meta-scheduling co-allocated jobs in a mini-grid. **Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International**. IEEE, 2004. p. 253.
- KAZMI, Irum; BUKHARI, Syed Fahim Yousaf. PeerSim: An efficient & scalable testbed for heterogeneous cluster-based P2P network protocols. **Computer Modelling and Simulation (UKSim), 2011 Uksim 13th International Conference on**. IEEE, 2011. p. 420-425.
- KECSKEMETI, Gabor; OSTERMANN, Simon; PRODAN, Radu. Fostering energy-awareness in simulations behind scientific workflow management systems. Em: **Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on**. IEEE, 2014. p. 29-38.
- KILLIAN, Charles Edwin et al. Mace: language support for building distributed systems. Em: **ACM SIGPLAN Notices**. ACM, 2007. p. 179-188.
- KLIAZOVICH, Dzmitry; BOUVRY, Pascal; KHAN, Samee Ullah. GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. **The Journal of Supercomputing**, v. 62, n. 3, p. 1263-1283, 2012.
- KUMAR, Parveen; RAI, Anjandeeep Kaur. An overview and survey of various cloud simulation tools. **Journal of Global Research in Computer Science**, v. 5, n. 1, p. 24-26, 2014.
- KUMAR, Rohit. Cloud Computing: An Introspection. **International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)**, v. 2, n. 11, p. 4413-4417, 2013.
- KUROSE, James F.; ROSS, Keith W. **Computer Networking: a Top-down Approach**. Boston: Addison Wesley, 6th ed., 2013.
- LEGRAND, Arnaud; MARCHAL, Loris; CASANOVA, Henri. Scheduling Distributed Applications: the SimGrid Simulation Framework. Em: **Proceedings of the 3rd**

- International Symposium on Cluster Computing and the Grid.** IEEE Computer Society, 2003. p. 138-145.
- MALHOTRA, Rahul; JAIN, Prince. Study and comparison of various cloud simulators available in the cloud computing. **International Journal of Advanced Research in Computer Science and Software Engineering**, v. 3, n. 9, 2013. p. 347-350.
- MEENEGHAN, Paul; DELANEY, Declan. An introduction to NS, Nam and OTcl scripting. **National University of Ireland**, 2004.
- MONTRESOR, Alberto; JELASITY, Márk. PeerSim: A scalable P2P simulator. **Peer-to-Peer Computing, 2009. P2P'09. IEEE Ninth International Conference on.** IEEE, 2009. p. 99-100.
- NAICKEN, Stephen et al. A survey of peer-to-peer network simulators. **Proceedings of The Seventh Annual Postgraduate Symposium**, Liverpool, UK. 2006.
- OLIVEIRA, José Antonio et al. Improving peer neighborhood on P2P video distribution networks using Push/Pull protocol. **Computer Communications**, v. 61, p. 17-33, 2015.
- OSTERMANN, Simon et al. Groudsim: An event-based simulation framework for computational grids and clouds. Em: **Euro-Par 2010 Parallel Processing Workshops.** Springer Berlin Heidelberg, 2010. p. 305-313.
- QUINSON, Martin; ROSA, Cristian; THIÉRY, Christophe. Parallel simulation of peer-to-peer systems. Em: **Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on.** IEEE, 2012. p. 668-675.
- RANGANATHAN, Kavitha; FOSTER, Ian. Decoupling computation and data scheduling in distributed data-intensive applications. In: **null.** IEEE, 2002. p. 352.
- ROHLOFF, Kurt; SCHANTZ, Richard E. High-performance, massively scalable distributed systems using the MapReduce software framework: the SHARD triple-store. Em: **Programming Support Innovations for Emerging Distributed Applications.** ACM, 2010. p. 1-5.
- SADASHIV, Naidila; KUMAR, SM Dilip. Cluster, grid and cloud computing: A detailed comparison. Em: **Computer Science & Education (ICCSE), 2011 6th International Conference on.** IEEE, 2011. p. 477-482.
- SULISTIO, Anthony et al. A toolkit for modelling and simulating data Grids: an extension to GridSim. **Concurrency and Computation: Practice and Experience**, v. 20, n. 13, p. 1591-1609, 2008.
- TANENBAUM, Andrew S.; VAN STEEN, Maarten. Distributed systems-principles and paradigms, New Jersey: Prentice Hall, 2nd ed., 2007.
- VARGA, András; HORNIG, Rudolf. An overview of the OMNeT++ simulation environment. In: **Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops.** ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008. p. 60.

- VELHO, Pedro; LEGRAND, Arnaud. Accuracy study and improvement of network simulation in the simgrid framework. Em: **Proceedings of the 2nd International Conference on Simulation Tools and Techniques**. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009. p. 13.
- XIA, Huaxia et al. The microgrid: Using online simulation to predict application performance in diverse grid network environments. In: **Challenges of Large Applications in Distributed Environments, 2004. CLADE 2004. Proceedings of the Second International Workshop on**. IEEE, 2004. p. 52-61.
- XU, He et al. A survey of peer-to-peer simulators and simulation technology. **Journal of Convergence Information Technology (JCIT)**, v. 6, n. 5, p. 260-272, 2011.
- ZHANG, Qi; CHENG, Lu; BOUTABA, Raouf. Cloud computing: state-of-the-art and research challenges. **Journal of Internet Services and Applications**, v. 1, n. 1, p. 7-18, 2010.
- ZHAO, Wei et al. Modeling and simulation of cloud computing: A review. Em: **Cloud Computing Congress (APCloudCC), 2012 IEEE Asia Pacific**. IEEE, 2012. p. 20-24.
- ZHOU, Shijie; ZHOU, Yanghong; LUO, Jiaqing. Research on simulators for peer-to-peer systems. Em: **Computer Science and Network Technology (ICCSNT), 2013 3rd International Conference on**. IEEE, 2013. p. 726-731.