

SIMMYCLOUD, simulando o gerenciamento de recursos virtualizados em plataformas de Computação em Nuvem

Cássio P. Alkmin, Daniel Cordeiro*

¹Instituto de Matemática e Estatística
Universidade de São Paulo (IME-USP)
Rua do Matão, 1010 – Cidade Universitária
São Paulo – SP – Brazil – 05508-090

{cassiop, danielc}@ime.usp.br

Abstract. *Cloud Computing platforms have changed the Information Technology industry, allowing on-demand rent of computational resources. There are, nevertheless, several infrastructure challenges that need to be tackled by Cloud Computing providers in order to offer an efficient service, while guaranteeing quality of service for their users. The management of computational resources is one of their main challenges, mainly because of the variety of users' performance goals and due to the constant changes on the demand for resources. Simulation tools are essential to evaluate different strategies on large scale scenarios. In this work, we present the SIMMYCLOUD, a framework to simulate Cloud Computing environments, that allows the use of real-world workload traces to allow the evaluation of new strategies to manage computational resources on a Cloud Computing platform.*

Resumo. *Plataformas de Computação em Nuvem revolucionaram a indústria de Tecnologia da Informação ao permitir a locação sob demanda de grandes quantidades de recursos computacionais. Porém, ainda há muitos desafios infraestruturais para que tais recursos sejam oferecidos de modo mais eficiente, e sem comprometer a qualidade do serviço contratado pelos usuários. Dentre tais desafios, o gerenciamento dos recursos computacionais se destaca, frente à diversidade de usos que os usuários farão dos recursos e à demanda volátil dos mesmos. Ferramentas de simulação tornam-se, assim, importantes aliadas na avaliação de diferentes estratégias de gestão de recursos computacionais em cenários de grande escala. Neste trabalho apresentamos o SIMMYCLOUD, um arcabouço para a simulação de ambientes de Computação em Nuvem que permite o uso de traços de execução reais para a avaliação e comparação de novas estratégias de gestão de recursos em plataformas de Computação em Nuvem.*

1. Introdução e motivação

Plataformas de Computação em Nuvem (*Cloud Computing*) revolucionaram a indústria de Tecnologia da Informação ao permitir que grandes quantidades de recursos computacionais fossem oferecidos aos usuários como um serviço sob demanda.

*Este trabalho foi parcialmente financiado pela CAPES e pela FAPESP (proc. #2012/03778-0).

Tais plataformas proveem diversas vantagens competitivas para a indústria. Por exemplo, uma empresa *startup* pode colocar uma nova aplicação em produção sem se preocupar com os custos de aquisição e manutenção de servidores. Outra vantagem de sua utilização é a escalabilidade que pode ser atingida com tais plataformas. Por exemplo, um pesquisador com uma simulação composta por diversas tarefas do tipo *batch* poderia executá-la em menos tempo em uma plataforma de Computação em Nuvem. Em tais plataformas, o custo de utilizar 1.000 servidores durante uma hora é, tipicamente, o mesmo que utilizar um servidor durante 1.000 horas.

A Computação em Nuvem não é classificada como um novo paradigma em termos de pesquisa em Ciência da Computação. Computação em Nuvem é, na verdade, a evolução e a convergência de pesquisas realizadas em diferentes áreas da Computação, tais como aglomerados (*clusters*) e grades (*grids*) de computadores, computação autônoma, computação utilitária, etc. Entretanto, seu sucesso com aplicações comerciais em grande escala indica seu potencial para a execução de aplicações científicas de alto desempenho.

Para tal, é necessário que a gestão de recursos computacionais seja controlada pelo desenvolvedor e realizada de tal forma que a plataforma forneça garantias sobre a qualidade do desempenho que pode ser obtido. Contudo, o problema de gestão de recursos em plataformas de Computação em Nuvem— diretamente relacionado ao problema clássico de combinatória conhecido como *Vector Bin Packing* — é sabidamente NP-difícil e pouco se conhece sobre os métodos de gestão utilizados em plataformas comerciais (que são mantidos como segredo industrial).

Plataformas de Computação em Nuvem podem utilizar desde estratégias de escalonamento estático (onde máquinas virtuais são designadas às máquinas físicas que as executarão no início da execução), quanto dinâmicas (onde máquinas virtuais são migradas à medida que novas informações sobre a plataforma de execução são obtidas e analisadas. Para estudar novas estratégias de gestão de recursos, desenvolvemos o simulador SIMMYCLOUD.

O SIMMYCLOUD foi projetado para que novos algoritmos de gestão de recursos sejam avaliados utilizando-se traços de execução de plataformas de Computação em Nuvem reais. Abstraímos os tipos de eventos (lógicos e físicos) presentes em tais traços e fornecemos uma API que permite não só a criação de novas heurísticas de escalonamento, mas também acesso à estatísticas de desempenho em tempo de execução e métricas consolidadas de desempenho ao final da simulação.

O restante deste artigo é organizado como se segue. A Seção 2 discute os trabalhos relacionados ao estudo da gestão de recursos e consolidação de servidores em plataformas de Computação em Nuvem. Nossa proposta para a simulação de novos algoritmos de gestão utilizando-se traços de execução reais é apresentada na Seção 3, e sua arquitetura e detalhes de implementação na Seção 4. Na Seção 5 apresentamos experimentos realizados com nosso simulador para a avaliação de algoritmos clássicos de *Vector Bin Packing* aplicados no problema da gestão de recursos de uma nuvem. Detalhes sobre a documentação e demonstração de uso do simulador são apresentados na Seção 6. Finalmente, apresentamos nossas considerações finais na Seção 7.

2. Trabalhos relacionados

Sabe-se que a consolidação de servidores é uma estratégia efetiva para maximizar a utilização dos recursos, ao mesmo tempo que diminui o consumo energético dos *datacenters* em que é empregada [Zhang et al. 2010]. A migração de máquinas virtuais é outra estratégia que tem sido adotada em ambientes virtualizados. Com o uso de *live migration*, sistemas operacionais completos podem ser migrados de uma máquina física para outra, com interrupção de disponibilidade do serviço por intervalo menor que um segundo.

A alocação das máquinas virtuais na menor quantidade possível de máquinas físicas, considerando mais de um recurso computacional, pode ser modelado como o problema de otimização combinatória conhecido como *Vector Bin Packing*, que é NP-difícil [Panigrahy et al. 2011]. É necessário o uso de heurísticas para encontrar soluções em tempo viável para este problema. Avaliamos algumas ferramentas que propõem simular um ambiente de infraestrutura de Computação em Nuvem, de modo a permitir comparações dos efeitos de se empregar diferentes heurísticas para alocação de máquinas virtuais, bem como estratégias para predição de desempenho e migração de máquinas virtuais.

A ferramenta CloudSched [Tian et al. 2013] simula uma plataforma de computação considerando quatro recursos: *rede*, *disco*, *memória* e *CPU*. A ferramenta permite realizar simulações com *workloads* gerados artificialmente ou com traços de execução preexistentes. A simulação considera que a execução é realizada em um ambiente virtualizado, e a alocação de máquinas virtuais ocorre dinamicamente. A ferramenta não foi projetada para analisar a migração de máquinas virtuais entre máquinas físicas.

Um arcabouço de simulação de ambientes de Computação em Nuvem precursor do CloudSched é o CloudSim [Calheiros et al. 2011]. Ele permite a modelagem de fatos tanto técnicos quanto de negócios, inerentes a todo provedor de computação como serviço. Com a arquitetura direcionada à modelagem de comportamentos, a ferramenta não permite a utilização de um traço de execução preparado previamente. O CloudAnalyst [Wickremasinghe et al. 2010] provê uma interface gráfica e outras extensões para o CloudSim, de modo a facilitar a visualização dos resultados de suas simulações.

O SIMMYCLOUD foi idealizado como um *framework* capaz de simular plataformas de Computação em Nuvem, com foco em avaliar diferentes soluções para desafios de infraestrutura desses ambientes, e quais os efeitos dessas soluções em conjuntos de dados previamente coletados. O fluxo da simulação permite avaliar efeitos de diferentes estratégias para alocação dinâmica de máquinas virtuais, predição de desempenho das mesmas, migração de máquinas virtuais e critérios para desligar máquinas físicas ou colocá-las em estado de baixo consumo de energia.

3. SIMMYCLOUD e suas funcionalidades

Neste trabalho propomos a ferramenta SIMMYCLOUD. O SIMMYCLOUD provê um arcabouço para a simulação e análise de diferentes estratégias de gerenciamento de recursos computacionais em ambientes virtualizados. O SIMMYCLOUD foi modelado como um *sistema a eventos discretos* [Medina and Chwif 2010], onde a simulação ocorre de acordo com o processamento de uma sequência de eventos ordenados de acordo com um relógio lógico. O estado do sistema só pode ser alterado durante o processamento de um evento.

A alocação dos recursos computacionais ocorre em tempo de execução e utiliza

dados como a quantidade de recursos necessários à execução da máquina virtual, o tempo pelo qual ela deverá ser executada, etc. Os dados necessários em cada requisição são apresentados em maior detalhe na Seção 5.

Com o objetivo de comparar os efeitos no uso de diferentes estratégias de gerenciamento de recursos computacionais, quatro classes de estratégias podem ser implementadas e substituídas em arquivo de configuração:

Predição de desempenho algoritmo que determina (utilizando-se de alguma estratégia de predição) a quantidade estimada de recursos computacionais que uma dada máquina virtual irá utilizar no próximo intervalo de tempo da simulação;

Alocação de máquinas virtuais estratégia que determina qual máquina física será designada para a execução de uma máquina virtual que ainda não iniciou sua execução;

Migração de máquinas virtuais estratégia que utiliza-se de dados do preditor de desempenho para decidir se uma dada máquina virtual deve continuar sua execução em uma máquina física ou se deve ser migrada para uma máquina com mais recursos.

Desligamento (ou suspensão) de máquinas físicas determina quais máquinas físicas ociosas podem ser desligadas ou configuradas para um estado de menor consumo de energia.

Módulos de estatísticas são os responsáveis por computar dados relevantes para a comparação do emprego de diferentes estratégias. Novos módulos de estatísticas podem ser adicionados para medir e analisar novas métricas de desempenho. O SIMMYCLOUD atualmente permite a análise das seguintes métricas:

falhas de QoS número de máquinas virtuais que executaram em uma máquina física sobrecarregada.

máquinas físicas ativas quantidade de máquinas físicas ativas, ou seja, não estão em um estado de baixo consumo de energia.

migrações quantidade de migrações de máquinas virtuais que ocorreram no último intervalo de tempo.

máquinas virtuais em espera quantidade de máquinas virtuais que não foram executadas por falta de recursos disponíveis ¹.

Capacidade residual total soma da capacidade residual das máquinas físicas ativas. Veja definição a seguir.

% capacidade residual em cada máquina física porcentagem da capacidade residual de cada máquina física.

Stretch de cada máquina virtual – razão entre o tempo que a máquina virtual demora para ser executada (contado a partir de sua submissão no sistema) pelo tempo realmente utilizado para a sua execução.

Cabe aqui elucidar o que é considerado uma falha de QoS e a capacidade residual de uma máquina física. Uma falha de QoS (*Quality of Service*) ocorre quando uma máquina virtual está alocada em uma máquina física sobrecarregada, ou seja, a quantidade de recursos disponíveis na máquina física é inferior à soma das demandas das máquinas virtuais alocadas nessa máquina física. A *capacidade residual* [Panigrahy et al. 2011] de uma máquina física é definida como a norma do vetor residual \vec{r} , cujas componentes são determinadas pela quantidade ainda disponível de cada recurso na máquina física, como ilustrado na Figura 1.

¹ Uma máquina virtual em espera não é considerada uma falha de QoS.

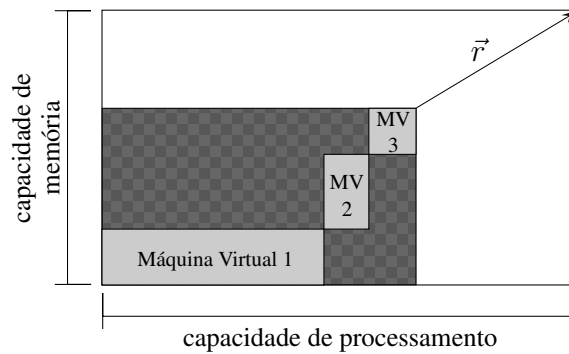


Figura 1. Vetor residual de uma máquina física

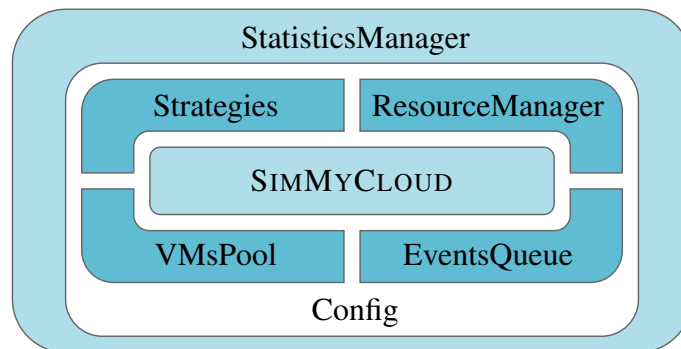


Figura 2. Arquitetura do SIMMYCLOUD

4. Arquitetura do SIMMYCLOUD

O SIMMYCLOUD foi desenvolvido seguindo alguns princípios, a saber:

- *Extensibilidade* – Regras e fluxos que podem variar entre simulações devem ser componentizados, de forma a permitir fácil adição ou substituição de regras de negócio.
- *Softcoding* – Variáveis e módulos não devem ser *hard-coded*; para executar simulações diferentes não devem ser necessárias alterações no código fonte.
- *Duck typing* – A fim de evitar uso de computação para verificar constantemente se estratégias e módulos em uso são de um tipo específico, assume-se que o objeto possui os métodos e propriedades necessários de acordo com sua semântica.
- *Agnosticismo perante componentes* – Cada componente tem sua responsabilidade e sabe de que dados ou outros componentes da simulação precisa, assim, a injeção de dependências é minimizada e todos os componentes são, de modo padronizado, acessíveis.

Tais princípios foram úteis para que a arquitetura ficasse enxuta e permitisse que simulações com diferentes módulos e variáveis fossem facilmente configuradas.

A Figura 2 ilustra os componentes principais do SIMMYCLOUD, cujas responsabilidades são:

- SIMMYCLOUD – executar o fluxo principal do simulador, coordenando o acesso aos demais componentes a fim de que executar o comportamento esperado a cada evento.

- *Strategies* – tomar decisões quanto ao gerenciamento dos recursos computacionais, quando necessário.
- *ResourceManager* – gerenciar os recursos computacionais em si, desde ligar e desligar máquinas físicas a alocar e liberar recursos dessas máquinas.
- *VMsPool* – gerenciar as máquinas virtuais que não estão alocadas, o que geralmente ocorre quando nenhuma máquina física é capaz de prover os recursos requisitados por uma ou mais máquinas virtuais.
- *EventsQueue* – gerenciar os eventos de simulação agendados, priorizando-os para execução.
- *StatisticsManager* – gerenciar os módulos de estatísticas ativos na simulação, notificando-os quando ocorre eventos no gerenciamento dos recursos computacionais.
- *Config* – gerenciar os componentes configurados para a simulação atual, sendo uma interface para acesso a todos os componentes.

5. Avaliação experimental

Para avaliar o simulador, utilizamos um traço de execução disponibilizado publicamente pelo Google, o *ClusterData2011_1* [Wilkes and Reiss 2013], que possui informações detalhadas sobre todos os eventos que ocorreram em uma “célula”² de um *datacenter*, com aproximadamente 12 mil máquinas físicas. Os dados coletados representam um período de 29 dias do mês de maio de 2011, anonimizados para fins de confidencialidade.

Para essa avaliação experimental, utilizamos os dados referentes às primeiras duas horas desse traço de execução, período suficiente para uma comparação preliminar das estratégias. Importamos os dados do traço para o simulador utilizando *scripts* implementados no SIMMYCLOUD. A entrada do simulador é composta pelos seguintes dados para cada máquina virtual:

- *momento da submissão* – tempo de simulação em que é requisitada a execução da máquina virtual;
- *identificador* – identificador da máquina virtual, que pode conter letras, números e pontuação, excetuando a “,” (vírgula);
- *CPU e memória solicitadas* – recursos de processamento e memória RAM solicitadas para a execução desta máquina virtual;
- *tempo de execução* – tempo estimado para a execução completa da máquina virtual.

A simulação foi configurada para execução com as estratégias de gerenciamento de recursos computacionais descritas a seguir. A estratégia *FirstFit* foi escolhida para realizar alocação das máquinas virtuais, a qual as aloca na primeira máquina física capaz de prover a quantidade de recursos requisitados. O algoritmo de predição de desempenho estima o uso de recursos necessários baseado na média das últimas cinco medições de cada máquina virtual. A estratégia de migração *MigrateIfOverloaded* migra máquinas virtuais de máquinas físicas sobrecarregadas para outras. Caso uma máquina física fique ociosa, a estratégia *PowerOffIfEmpty* a desliga (ou a suspende).

Utilizamos o SIMMYCLOUD para comparar o desempenho da estratégia *FirstFit* com e sem o uso de um algoritmo de predição de desempenho para amparar a gestão de

²Uma célula é um conjunto de máquinas que compartilham um mesmo sistema de gerenciamento de *clusters*, geralmente presentes em um único *cluster*.

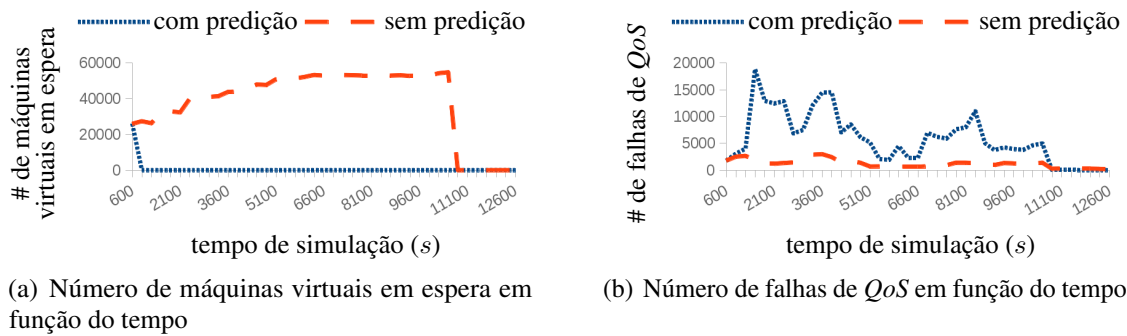


Figura 3. Número de falhas de QoS e quantidade de máquinas virtuais ociosas obtidas com e sem o uso de informações de predição de desempenho.

recursos. A não utilização de um algoritmo de predição de desempenho implica que as decisões de gerenciamento de recursos serão baseadas apenas na estimativa dada pelo desenvolvedor. A comparação é apresentada na Figura 3.

Como pode ser observado na Figura 3(a), o uso de um algoritmo de predição de desempenho é capaz de evitar a presença de máquinas virtuais na fila de espera por uma máquina física na maior parte do tempo. Sem o uso de predição, elas precisam esperar o término de outras máquinas virtuais antes de serem alocadas a uma máquina física, o que só ocorre a partir do tempo $t = 11.100$ segundos.

Uma análise aprofundada (Figura 3(b)) mostra, entretanto, que a diminuição da fila de espera implica em um aumento significativo do número no número de falhas de QoS. Isso ocorreu pois a estratégia de predição de desempenho subestimou, em alguns casos, a quantidade de recursos que as máquinas virtuais necessitariam nos intervalos de tempo seguintes, o que levou à sobrecarga de algumas máquinas físicas.

A partir dessa análise, podemos planejar, por exemplo, nova simulação com a mesma estratégia de predição, porém com uma estratégia de alocação de máquinas virtuais que não busque utilizar 100% da capacidade de recursos computacionais disponíveis nas máquinas físicas. Desse modo, teríamos uma tolerância a erros da estratégia de predição, ainda que com possível aumento na quantidade de máquinas virtuais em espera.

6. Documentação e demonstração no SBRC

Na página da ferramenta, disponibilizada publicamente em <http://www.ime.usp.br/~cassiop/simmycloud/>, é possível encontrar uma breve descrição do SIMMY-CLOUD, sua documentação e seu código fonte (escrito na linguagem Python e disponibilizado como software livre). O Guia do Usuário apresenta com mais detalhes o funcionamento da ferramenta, seus componentes e meios para extendê-los.

A demonstração da ferramenta no SBRC mostrará todos os passos necessários para sua utilização, desde o *download* do traço de execução real disponibilizado pelo Google até a simulação e análise das métricas de desempenho calculadas automaticamente pela ferramenta. A demonstração apresentará todas as estratégias implementadas pelo SIMMYCLOUD e realizará, ao vivo, uma comparação de duas execuções configuradas com estratégias de gestão de recursos distintas. Apresentaremos como analisar graficamente os dados gerados pelo SIMMYCLOUD.

7. Considerações finais

Neste trabalho apresentamos o SIMMYCLOUD, um novo simulador para amparar a criação e análise de métodos de gerenciamento de recursos em plataformas de Computação em Nuvem.

O SIMMYCLOUD implementa diversas estratégias para a consolidação de servidores baseadas no problema clássico conhecido como *Vector Bin Packing*. Além disso, sua arquitetura foi projetada para armazenar e fornecer informações e métricas de desempenho relevantes, que podem ser facilmente utilizadas para o desenvolvimento de novos algoritmos.

Neste artigo ilustramos a utilização da ferramenta utilizando como exemplo a análise comparativa entre o uso ou não de um algoritmo simples para predição de desempenho. A análise foi realizada utilizando-se um traço de execução de um provedor de Computação em Nuvem comercial bastante conhecido.

O código fonte da ferramenta e instruções para sua utilização estão disponíveis publicamente em <http://www.ime.usp.br/~cassiop/simmycloud/>. O código é disponibilizado sob a licença permissiva MIT, e seu guia do usuário sob a licença Creative Commons - Atribuição 4.0 Internacional.

Referências

- [Calheiros et al. 2011] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., and Buyya, R. (2011). Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1):23–50.
- [Medina and Chwif 2010] Medina, A. C. and Chwif, L. (2010). *Modelagem e Simulação de Eventos Discretos*. Editora Campus, 3 edition.
- [Panigrahy et al. 2011] Panigrahy, R., Talwar, K., Uyeda, L., and Wieder, U. (2011). Heuristics for vector bin packing. Technical report, Microsoft Research. Disponível em: <http://research.microsoft.com/apps/pubs/default.aspx?id=147927>.
- [Tian et al. 2013] Tian, W., Zhao, Y., Xu, M., Zhong, Y., and Sun, X. (2013). A toolkit for modeling and simulation of real-time virtual machine allocation in a cloud data center. *IEEE Transactions on Automation Science and Engineering*.
- [Wickremasinghe et al. 2010] Wickremasinghe, B., Calheiros, R. N., and Buyya, R. (2010). CloudAnalyst: A CloudSim-based visual modeller for analysing cloud computing environments and applications. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 446–452. IEEE.
- [Wilkes and Reiss 2013] Wilkes, J. and Reiss, C. (2013). ClusterData2011_1 - Details of the ClusterData-2011-1 trace. Google Project Hosting – http://code.google.com/p/googleclusterdata/wiki/ClusterData2011_1. [Online; acessado em 23/02/2014].
- [Zhang et al. 2010] Zhang, Q., Cheng, L., and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1):7–18.