

A UTILIZAÇÃO DO POSICIONAMENTO DE MÁQUINAS VIRTUAIS PARA A QUALIDADE DE APLICAÇÕES DE CLIENTES EM NUVEM

Daniel Freitas de Jesus – Universidade federal do ABC

Daniel.jesus@ufabc.edu.br

Prof. Dr. Carlos Alberto Kamienski – Universidade Federal do ABC

Carlos.kamienski@ufabc.edu.br

Resumo

Com a evolução das tecnologias de virtualização, das redes e dos hipervisores de computação, o compartilhamento de recursos nas atuais infraestruturas como serviços, tornou-se mais flexível. Toda a implementação de uma nuvem, depende de desempenho e padronizações para clientes; tendo em vista que o ambiente é compartilhado, todos querem obter o melhor desempenho. As qualidades de serviços são definidas, protocolos são padronizados, e há a preocupação com a segurança entre os clientes, em que são criadas quotas de recursos e, na teoria, a plataforma está preparada para a demanda. Porém o desempenho para os clientes varia e, em pouco tempo, a degradação dos recursos torna-se uma barreira para o uso eficiente da infraestrutura da nuvem. O comprometimento da escalabilidade e da disponibilidade, acontece pela falta de uma estratégia para a heterogeneidade de *hardware*, variações dos fatores de rede, computação e cargas de trabalho, que são as características de um ambiente compartilhado, sendo necessária uma tecnologia capaz de satisfazer de maneira automatizada das regras propostas pelo modelo da nuvem. Estas características são estudadas neste artigo, assim como os conceitos chaves da computação em nuvem, as redes virtuais e os algoritmos de *placement*.

Palavras-Chaves: Computação em Nuvem; Posicionamento de Máquinas Virtuais; *OpenStack*; Virtualização; Avaliação de Desempenho.

Abstract

With the evolution of virtualization technologies, networks and computing hypervisors, resource sharing in today's infrastructures as services has become more flexible. The entire implementation of a cloud depends on performance and standardization for customers; Given that the environment is shared, everyone wants to get the best performance. Service qualities are defined, protocols are standardized, and there is concern for security among customers, where resource quotas are created and, in theory, the platform is ready for demand. But performance for customers varies, and resource degradation soon becomes a barrier to the efficient use of cloud infrastructure. The compromise of scalability and availability is due to the lack of a strategy for hardware heterogeneity, variations of network factors, computing and workloads, which are the characteristics of a shared environment, requiring a technology capable of satisfying Automation of the rules proposed by the cloud model. These characteristics are studied in this article, as well as the key concepts of cloud computing, virtual network, data center models and placement algorithms.

Keys-Words: Cloud computing; Virtual Machines Placement; OpenStack; Virtualization; Performance evaluation.

Introdução

São inúmeros os exemplos de empresas com sistemas *Web*, que migraram as aplicações ao ambiente de nuvem, entre elas, aplicações que fazem parte do dia-dia da sociedade informatizada, como as redes sociais, canais de vídeos, programação de televisão, serviços de transportes, da sociedade rural para o monitoramento, computação científica e o uso na Internet das Coisas no cenário urbano.

Sob uma visão administrativa, o motivo desta adesão ao ambiente em nuvem é a ausência de um custo inicial em um investimento de infraestrutura de TI, que pode ou não trazer o retorno esperado devido aos efeitos de variação de clientes (o mesmo efeito de *Flash Crowd*¹) [11].

¹ *Flash Crowd* é um termo criado em 1971 por Larry Niven, representa um aumento anormal de procura por um site por causa de algum evento, neste processo pode ocorrer indisponibilidades e atrasos [11].

Empresas grandes e médias que já possuem *datacenters* utilizam a nuvem privada e a nuvem híbrida, obtendo capacidade terceirizada ajustada para sua demanda [17]; [18].

O provedor é o responsável pela infraestrutura da nuvem, mantendo os recursos acessíveis de maneira transparente aos clientes, independentemente da demanda, através do aumento de recursos virtuais (*CPU*, memória, rede e disco), sempre quando necessário [11].

Para o provedor quanto mais o mesmo utiliza os recursos físicos para alocar máquinas virtuais e aplicações, melhor será a sua receita de vendas, enquanto ao cliente, quanto melhor a qualidade da oferta destes recursos computacionais, melhor será o desempenho de sua aplicação [11].

Neste contexto, surge a necessidade de ferramentas que garantam ao provedor a oferta de recursos de forma maximizada, enquanto mantenha a qualidade dos recursos ao cliente da nuvem e seus usuários da aplicação.

Como objetivo geral, esta pesquisa busca o estudo de algoritmos de posicionamento na plataforma *OpenStack*, por meio da análise estatística de recursos disponíveis para virtualização e rede, identificando cargas de trabalho e o comportamento gerado dentro do ambiente de nuvem.

Este trabalho contribui no campo de compreensão do comportamento dos algoritmos de posicionamento em ambientes homogêneos e heterogêneos, com variações de cargas de trabalho, identificando que a forma como os recursos são utilizados implica no desenvolvimento da elasticidade e na escalabilidade da nuvem.

Computação em nuvem

A Computação em Nuvem pode ser compreendida como o conjunto de aplicações entregues como um serviço pela internet, onde fisicamente servidores e aplicações estão alocados em *datacenters* terceirizados, provendo serviços sobre demanda [9], em regiões geograficamente distribuídas.

De acordo com o *NIST*², existem 5 características essenciais para a computação em nuvem [19]:

- Serviços sob demanda: os recursos são mensurados por utilização (horas de *CPU*, uso de largura de banda, entre outras métricas). Elas podem ser mensuradas sem a intervenção humana;
- Amplo acesso aos recursos: protocolos fornecem acesso a recursos heterogêneos como celulares, *notebooks*, *desktop*, entre outros;
- *Resource pooling*: este termo define um grupo de recursos físico e virtuais disponíveis em um ambiente multiusuário e abstrato em sua localização, ou seja, são recursos disponíveis para um cliente na nuvem;
- Elasticidade: a variação da escalabilidade com o aumento da demanda é uma tarefa da elasticidade. O cliente pode rapidamente ganhar mais recursos para a nuvem, escalando seus recursos fornecidos pelo provedor;
- Medição de serviços: a infraestrutura é constantemente monitorada para otimização do uso de recursos da nuvem, para a garantia de *QoS*,³ segurança e disponibilidade.

Máquinas virtuais utilizam melhor os recursos computacionais porque permitem que um servidor físico trabalhe com várias cargas de trabalho de clientes de maneira a isolar as aplicações, reduzindo a ociosidade dos servidores físicos [19].

Virtualização

A virtualização é um paradigma que permite o compartilhamento de recursos físicos em um ambiente virtual [8]. As máquinas virtuais podem ser de dois tipos [2]:

- Tipo I (*Guest System* – sistema convidado): o *hipervisor*⁴ é executado entre o hardware e os sistemas convidados, o *hardware* é compartilhado para cada máquina virtual. São exemplos deste tipo o *hipervisor Xen Server*,

² *National Institute of Standards and Technology*, é o órgão responsável pelo desenvolvimento de normas, e padronizações para as operações em computação em nuvem [19].

³ A qualidade de serviço (*QoS Quality of Service*), pode ser entendida como atender as expectativas perceptíveis e mensuradas do cliente a um serviço prestado na nuvem [19].

⁴ São os monitores que criam e gerenciam recursos para as máquinas virtuais, fornecendo uma interface de acesso entre o servidor físico e a máquina virtual.

- Tipo II (Host System): o hipervisor é um processo executando dentro do sistema real, simulando as operações reais, cada nova instância recebe um novo *hipervisor*.

As técnicas de virtualização podem ser classificadas em [8]; [3]:

- Virtualização total: a virtualização é completa, sem modificações no convidado, nesta classe, o hipervisor faz o gerenciamento de recursos de forma imperceptível para a máquinas virtuais e cada máquina virtual executa sua arquitetura nativa;
- Paravirtualização: ocorre a modificação do núcleo do sistema operacional e todas as referências feitas pelo hóspede ao hardware precisam ser monitoradas pelo hipervisor. As chamadas de sistemas são substituídas por chamadas ao hipervisor que proporciona um hardware virtual;
- Recompilação dinâmica: ocorre em 7 passos.

A virtualização é utilizada no atual contexto computacional para aplicações em nuvem, este paradigma evita aos donos de aplicações cliente-servidor um desperdício de investimentos feitos com a previsão da carga de trabalho (picos, ociosidade e crescimento imprevisto), e ao mesmo tempo traz aos provedores a oportunidade de oferecer aos clientes um ambiente elástico de recursos virtuais ajustado a demanda da aplicação [13].

Redes Definidas por Softwares

Devido a popularização e presença diária das redes de computadores na sociedade urbana, governos e população mundial, fabricantes e pesquisadores de estruturas de redes abordam os desafios da conectividade através da comutação da rede física de forma inovadora, programando o roteamento e a comutação via software [15].

Um dos padrões intermediários desta mudança é o padrão *Openflow*. O encaminhamento do padrão *Openflow* segue o princípio do encaminhamento baseado em pacotes, onde cada pacote inspecionado gera uma consulta na tabela de encaminhamento do comutador. Para algumas plataformas de nuvem, o *Openflow* é utilizado para criar as redes virtuais isolando o tráfego entre os clientes [15].

Datacenters Verdes

Em todo o mundo, os efeitos de gases de efeito estufa (GEE), que envolvem a penetração da tecnologia no cotidiano da sociedade, rapidamente propagam-se, e este efeito atualmente representa 2,2 % do total da emissão feita pelo homem [1].

O consumo de energia também impulsiona o segmento das redes verdes, segundo o autor [4], o consumo de energia dos datacenters varia entre 0,1 e 1,5% a nível mundial e a projeção é de um acréscimo de 2,2% ao ano [20].

Dentro do Datacenter, as máquinas virtuais podem consumir mais energia enquanto estiverem mais distribuídas no servidor físico, isso ocorre porque a cada máquina virtual criada, a mesma irá consumir recursos de um servidor físico. Porém máquinas menos distribuídas podem acarretar em mais sobrecargas ao servidor físico, gerando mais variação entre os usuários do sistema implementado na nuvem [1].

Posicionamento de Máquinas Virtuais

O posicionamento de máquinas virtuais é o responsável pelo processo de seleção de quais máquinas virtuais devem ser alocadas nas máquinas físicas dentro do *Datacenter* [7].

As máquinas virtuais de uma nuvem podem ser de duas formas: homogêneas (de um mesmo tipo e tamanho dimensional⁵) ou heterogênea (variando em recursos, hipervisores e sistemas. Um exemplo é a nuvem da *Amazon* possuindo 50 tipos diferentes com suporte a diversas características) [12].

A seleção errada do *host* (servidor físico), pode causar desequilíbrio aos parâmetros de qualidade da aplicação, desperdício de recursos e aumentar o consumo de energia do *Datacenter*, isso implica em despesas extras tanto ao provedor da nuvem, que não consegue atender as necessidades das aplicações do cliente, como para os usuários que recebem uma aplicação de baixa qualidade [16].

⁵ Dimensões de recursos são os recursos disponíveis ou solicitados para a criação da máquina virtual, esta pode ser CPU, memória, Disco, rede e em outros algoritmos de posicionamento também pode agregar a afinidade do usuário ao conjunto de servidores físicos.

Podemos classificar em duas categorias os problemas de posicionamento de máquinas virtuais [7]:

- Problema de alocação do servidor estático: as máquinas virtuais são atribuídas sem levar em consideração a variação de carga de trabalho ao longo do tempo, os recursos inicialmente fixados não sofrem alteração e o servidor que a hospeda só é trocado manualmente;
- Problema de alocação de servidor dinâmico: útil no balanceamento de carga, as máquinas virtuais são migradas em servidores periodicamente, acompanhando a variação da carga de trabalho.

Os algoritmos utilizados para o posicionamento de máquinas virtuais podem ser classificados [7] nos seguintes grupos:

- Algoritmo *FirstFit* (primeiro a aplicar): é otimizado localmente, possui forma gananciosa, o primeiro nó disponível no início é escolhido, devido à complexidade baixa, é considerado um algoritmo rápido;
- Algoritmo *NextFit* (próximo a aplicar): usa uma variável que pesquisa por máquinas físicas, logo após a alocação, a pesquisa começará da próxima máquina física disponível;
- Algoritmo *BestFit* (melhor ajuste) de uma dimensão: visita os servidores físicos em ordem decrescente de capacidade em uma única dimensão, posicionando a máquina virtual para a primeira máquina que possui recursos suficientes;
- Algoritmo Randômico: inicia a pesquisa de forma aleatória para cada máquina virtual. Se o servidor selecionado não satisfizer o pedido, escolhe outra máquina aleatoriamente;
- Algoritmo *BinPacking*: consideramos os servidores físicos e as VMs como objetos tridimensionais (*CPU*, memória, e largura de banda), com o objetivo de colocar um número de itens em um único recipiente possível;
- Algoritmo com Método de restrição: existem restrições aplicáveis como limitações de capacidade (*CPU*, memória, largura de banda física) com o objetivo de colocar um número de itens em um único recipiente possível;

- Método estocástico de programação de número inteiro: utiliza-se este método quando envolvem dados desconhecidos, objetivando encontrar um servidor físico que consuma menos energia;
- Algoritmo multiobjetivo de otimização por colônia de formigas (ACO – Ant Colony Optimization): é um algoritmo distribuído, em que a melhor solução é alcançada por meio do feromônio na trilha de cada formiga. Este método propõe minimizar a violação do *SLA*⁶ através da consolidação e da migração de máquinas virtuais.

Trabalhos Relacionados

No trabalho [5], os autores apresentaram o gerenciador de *SLA* usado pelo *OpenStack* e, suas estratégias de alocação de máquinas virtuais, para migração online. O objetivo do trabalho é usar um protótipo para o gerenciamento de *SLA* para o *OpenStack*, para ser conectado entre as camadas de *SaaS* e *PaaS*, fornecendo escalabilidade automática.

Segundo os autores do trabalho [14], o posicionamento soluciona problemas de balanceamento de carga, utilização de recursos, consolidação de servidores e o tempo de resposta. O autor trabalha o problema da dimensionalidade, em que procura otimizar a latência de rede, melhorar a recuperação de um banco de dados e, a melhoria na utilização de recursos. Os três parâmetros de desempenho são: *CPU*, memória, e largura de banda.

Para os autores [10], o desempenho obtido em relação ao posicionamento da VM, determina o impacto na *NFV*, devido a atrasos de alta ou baixa taxa de transferência entre máquinas virtuais. A afinidade significa que elas devem ser instanciadas na mesma máquina física e, a anti-afinidade, o inverso, que é uma regra. Porém em grandes conjuntos de máquinas virtuais para o mesmo rack ou domínio, seria complexo e difícil a depuração, pois a granularidade do servidor físico é parte dos requisitos da aplicação.

OpenStack

⁶ O *SLA* [5] é um contrato com regras para negócios, acompanhamento, monitoramento e execução de serviços. Inicialmente são definidas as responsabilidades de cada parte no serviço provido, logo após ocorre o acompanhamento e a execução cria o relacionamento com o cliente.

A plataforma *OpenStack* é um projeto que foi desenvolvido pela *NASA* e a *Rackspace* em 2010, consiste em módulos que representam um serviço para cada recurso da nuvem (computação, virtualização, segurança, rede e armazenamento), interagindo entre si através de uma *API*⁷ pública [21].

A comunicação entre cada módulo ou *daemon* (processos) ocorre através de um protocolo corretor de mensagens assíncronas com um serviço de *broker* que é o responsável pela classificação e o endereçamento das mensagens [22].

Para verificar os estados dos serviços e a utilização da plataforma, o *OpenStack* usa uma base de dados atualizada pelos serviços e monitores de recursos, cada módulo é desenvolvido independentemente, recebendo cada, um nome, dentre eles os principais são:

- *Keystone*: é o serviço responsável pela autenticação aos serviços da plataforma, cada serviço deve possuir uma entrada válida de usuário e senha;
- *Horizon*: possui uma interface Web administrativa para os serviços *OpenStack*, um painel de controle, um painel de usuário e um painel de configurações, também é responsável por um conjunto de *API* abstraídas dos módulos do *OpenStack* para desenvolvedores;
- *Nova-Compute*: este serviço é o responsável por fazer a comunicação direta com a *API* do hipervisor para a criação de máquinas virtuais e *containers*;
- *Neutron*: é o responsável pelo gerenciamento dos serviços de rede para o ambiente multiusuário, fornecendo conectividade entre as máquinas virtuais e o plano de controle para as redes virtuais.

O *OpenStack* possui dois tipos de programadores para alocação de máquinas virtuais [21]:

- *Nova-Scheduler-FilterScheduler*: é o algoritmo habilitado por padrão. O administrador pode usar filtros para selecionar servidores que tenham recursos especiais, como dispositivos, memória física disponível e outras restrições impostas ao posicionamento, a alocação acontece em forma circular;

⁷ *Application Programming Interface*: é a interface disponível pelo serviço que permite a comunicação de uma camada de nível mais baixo com outra de nível mais elevado.

- *ChanceScheduler*: o algoritmo utiliza servidores habilitados aleatoriamente sem regras de prioridades, com a seleção uniforme de um intervalo de sequências (listas de servidores disponíveis) de forma determinista e não vazia.

Relatos circunstanciados

Este trabalho utiliza a pesquisa experimental como método de análise dos algoritmos de posicionamento, verificando a influência nos tempos e processamento da aplicação.

O ambiente desenvolvido para testes consiste em 8 servidores físicos, Sistema Operacional Ubuntu Server 15, conforme mostra a Tabela 1, dispostos em uma rede árvore nível 2, representando um *rack* de servidores em uma nuvem privada.

Tabela 1 Ambiente de Desenvolvimento

Hypervisor		Compute Host						
Nome do host	Tipo	vCPUs (utilizados)	vCPUs (total)	RAM (utilizado)	RAM (total)	Armazenamento Local (usado)	Armazenamento Local (total)	Instâncias
sd1	QEMU	0	8	512MB	15,7GB	0Bytes	277GB	0
sd6	QEMU	0	8	512MB	15,7GB	0Bytes	213GB	0
sd8	QEMU	0	16	512MB	29,4GB	0Bytes	263GB	0
sd2	QEMU	0	8	512MB	15,7GB	0Bytes	277GB	0
sd7	QEMU	0	16	512MB	23,5GB	0Bytes	205GB	0
sd3	QEMU	0	16	512MB	29,4GB	0Bytes	263GB	0
sd5	QEMU	0	8	512MB	15,7GB	0Bytes	213GB	0

Displaying 7 items

As métricas avaliadas são o tempo, a taxa de alocação em cada servidor físico e a influência do posicionamento nos resultados obtidos.

Para o experimento 1, envio e recebimento de arquivos, as Vms possuem 4GB de RAM, 1 *Vcpu* e 20 GB de disco.

Os arquivos possuem 4 tamanhos diferentes, conforme mostra a Tabela 2. Os arquivos são transferidos utilizando-se o protocolo SCP (Security Copy Protocol), com uma criptografia de 128 bits, um script em python que escaneia IPs na rede enviando e recebendo os arquivos com tamanhos ordenados em escala crescente.

Tabela 2 Fatores e Níveis do experimento 1

#	Fator	Níveis
1	Algoritmo usado	<i>Chance-scheduler</i> <i>Filter-scheduler</i>
2	Tamanho de arquivos (SCP – 128 bit)	10 kb, 10 MB, 100MB, 512 MB
3	Rede	1GB
4	Tamanho da memória das VMs	4GB
5	Quantidade de VMs	15
6	PMs	Servidores
7	Tipo de protocolo	TCP
8	Tipo de comunicação	nXn
9	VCPUs	1
1 0	Tipo de PMs	Homogênea

Após 5 horas de testes, as VMs são destruídas aleatoriamente, mesmo ainda em execução. Em um cenário real, as VMs dos clientes se comunicam entre si cada uma fazendo o suporte a uma parte da aplicação (Servidor *Web*, Banco de Dados, etc.), estes módulos estão presentes na infraestrutura compartilhada e devido aos vários clientes, cada um com sua aplicação, a mesma sofre influência do posicionamento e da multitenacidade.

Para a avaliação do experimento 2, um tráfego experimental é modelado como o tráfego Web real utilizando a metodologia de Downey [6], em que os arquivos seguem uma distribuição log-normal, com média de 22,125 e desvio padrão de 40,07.

Os fatores e níveis estão representados na Tabela 3. A cada novo envio, o programa escolhe um novo arquivo uniformemente, o enviando com um tempo determinado em uma exponencial em segundos, com média de 5 segundos.

Tabela 3 Fatores e Níveis do experimento 2

#	Fator	Níveis
1	Algoritmo usado	Chance-scheduler Filter-scheduler
2	Tamanho de arquivos	Lognormal
3	Rede	1GB
4	Tamanho das VMs	4GB
5	Quantidade de VMs	15
6	PMs	Servidores
7	Tipo de protocolo	TCP
8	Tipo de comunicação	1X n
9	VCPUs	1
10	Tipo de PMs	Homogênea, Heterogênea

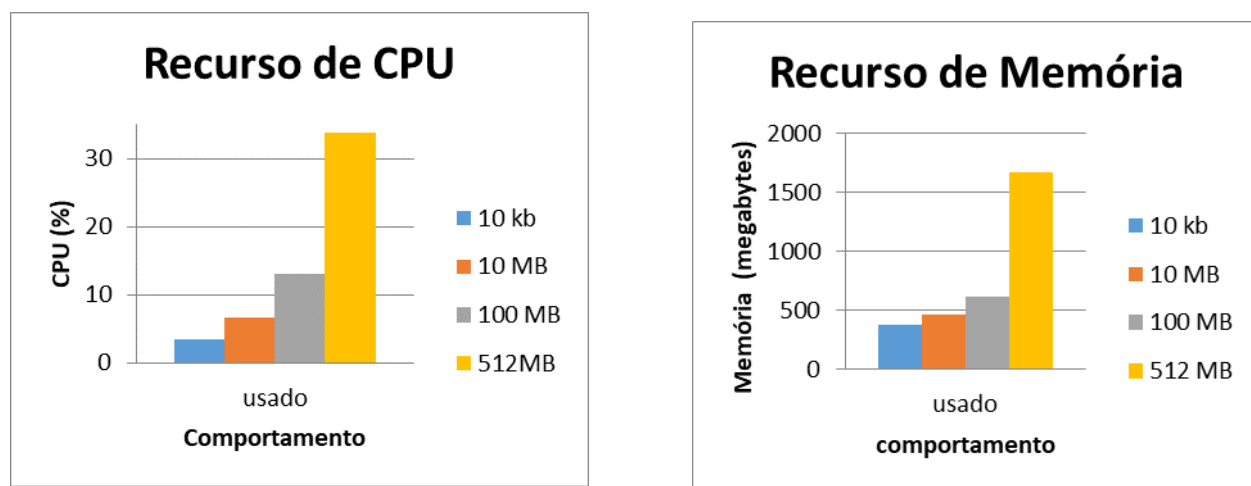
Para os dois algoritmos, foi solicitado a criação de 20 VMs idênticas, que fazem transferências com uma VM escolhida aleatoriamente.

O experimento 3 consiste em fazer processamento *OLTP*⁸ em cada máquina alocada para dois clientes independentes na nuvem. Cada cliente possui 7 máquinas virtuais que possuem um banco de dados Mysql-Server e um script que executa benchmark no SysBench.

Resultados e Discussões

Conforme mostra o Gráfico 1, os arquivos consomem recursos com média crescente em relação ao aumento do tamanho do arquivo, gerando uma variação de carga de necessidade de recursos, o crescimento é linear, as VMs são iguais entre si, conforme apresentado na Tabela 2.

Gráfico 1 - Recursos de CPU e Memória



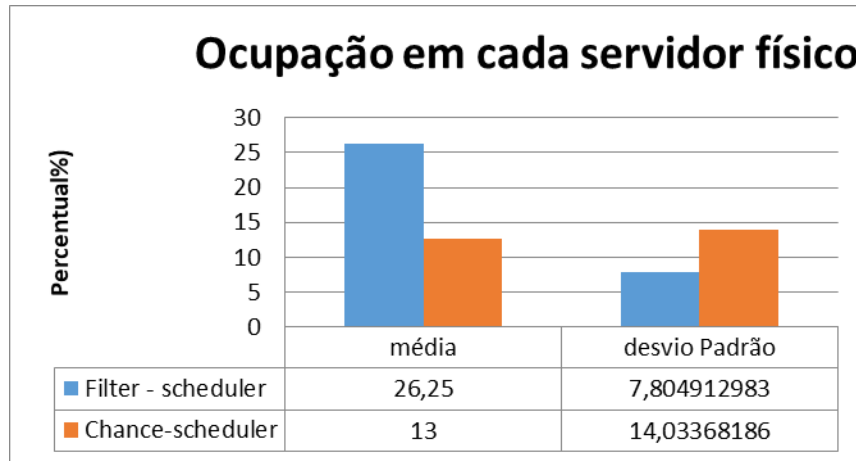
Ao aumentarmos a quantidade de VM alocada em cada máquina física, o tempo de transferência tende a aumentar, conforme reduzimos o número de VMs o tempo de transferência torna-se menor. Esta variação pode ser entendida como o custo no posicionamento. Como as máquinas físicas estão em um mesmo rack, não há custo de geodistribuição.

Segundo o autor [13], os grandes datacenters utilizam em média de 5 a 20 % de sua capacidade de processamento, considerando um cenário ótimo quando utilizam 30 % da capacidade total do servidor físico. Conforme mostra o Gráfico 2, o

⁸ OLTP: *Online Transaction Processing*. Processamento de transações em tempo real no banco de dados da empresa em seu dia-dia

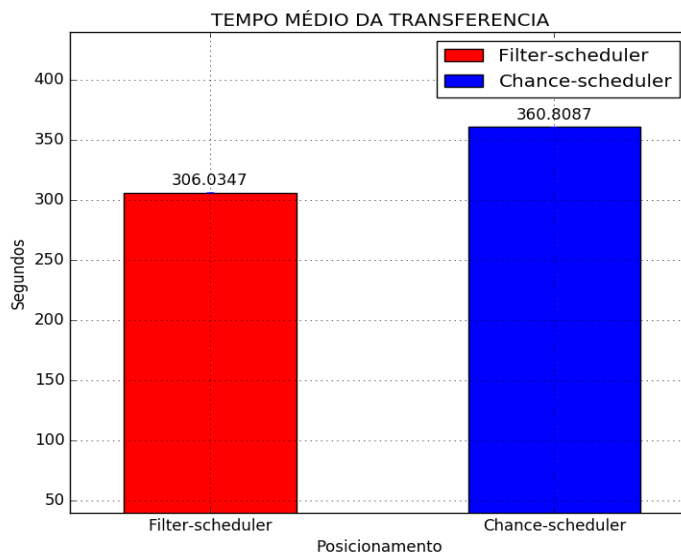
FilterScheduler possui maior ocupação do servidor físico cerca de 27 %, enquanto o ChanceScheduler possui uma ocupação menor (13 % por servidor físico), para 20 máquinas virtuais.

Gráfico 2 - Ocupação em servidor físico



Os resultados do experimento 2 são apresentados no Gráfico 3, em que a média de envios ficou em 306 segundos para o *FilterScheduler* e 360,80 para o *ChanceScheduler* evidenciando um melhor desempenho na transferência com o uso do *FilterScheduler*. Como as VMs são distribuídas de forma homogênea, houve um melhor ajuste nos servidores físicos, beneficiando o conjunto de VMs devido ao posicionamento.

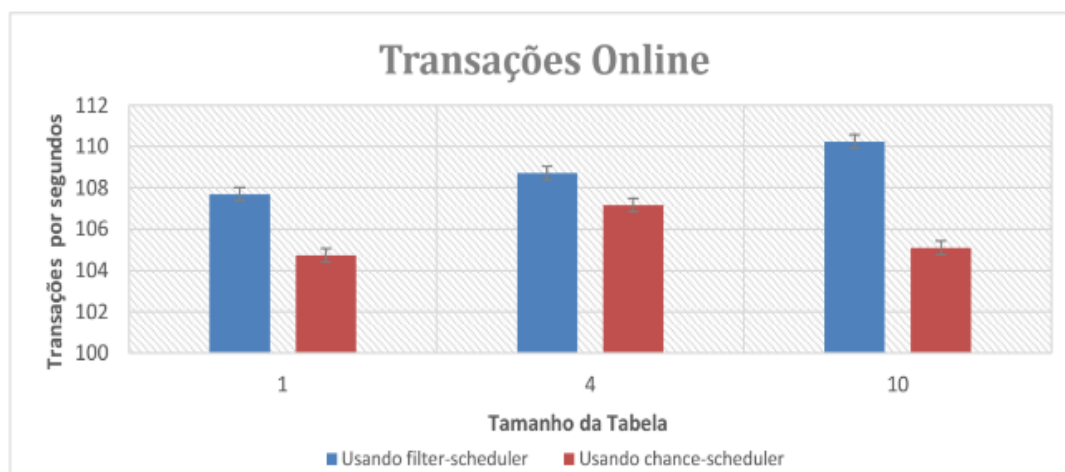
Gráfico 3 - Tempo médio da transferência



O algoritmo *FilterScheduler* consegue uma melhor distribuição da carga de trabalho em relação ao *ChanceScheduler* que usa aleatoriedade, causando desproporção entre as máquinas físicas. Em relação a um algoritmo de *binpacking* (que trata requisições em lotes) ele consegue manter a prioridade de clientes na fila de requisições.

O resultado do experimento 3 é apresentado no Gráfico 4, representando a quantidade de transações *OLTP* nas *VMs*. Ao usar tabelas de tamanho 10, o *FilterScheduler* continua a crescer, enquanto o *ChanceScheduler* teve uma queda em seu desempenho.

Gráfico 4 - Transações online



3.

Conclusões

A computação em nuvem possibilita o rápido desenvolvimento de ambiente virtualizados para clientes, com monitoramento centralizado, sendo suas aplicações presentes nos mais diversos setores, como bancos, hospitais, em computação científica, entre outros.

Para obter desempenho e provisionamento enxuto, os algoritmos de agendamento são os recursos presentes para esta tarefa, sendo estudados por suas limitações atuais, levando a comunidade científica a modelar novas formas de caracterização de uso e novas formas de agendamento.

Esta primeira parte da pesquisa buscou uma revisão dos agendadores utilizados atualmente na plataforma OpenStack, pois eles são os mais presentes, mesmo em outras plataformas, como no Opennebula, em que filtros do OpenStack tem um trabalho similar as políticas do OpenNebula.

A próxima etapa será estudar aplicações para a nuvem, estudar e propor um modelo de agendamento eficiente, e utilizar os conceitos chaves para melhorar o desempenho da aplicação.

Referências Bibliográficas

[1] BIANZINO; CHAUDET; ROSSI; ROUGIER. A Survey of Green Networking Research. Institut TELECOM, TELECOM ParisTech, CNRS LTCI UMR 5141, Paris, France 2010. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[2] CALLEGATI, Franco; CERRONI, Walter; CONTOLI, Chiara. Virtual Networking Performance in OpenStack Platform for Network Function Virtualization. Journal of Electrical and Computer Engineering Volume 2016, Article ID 5249421, Received 19 October 2015; Revised 19 January 2016; Accepted 30 March 2016. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[3] CARISSIMI, Alexandre. Virtualização: da teoria a soluções; 26º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos Minicursos do Simpósio Brasileiro de Redes de Computadores—SBRC 2008, 173-207. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[4] CHANDRAKAR, Vallari; PATEL, Punyaban; Roy, MANOJEET. REVIEW ON VIRTUAL MACHINE PLACEMENT ALGORITHMS. Ijirse.com; International Journal of Innovative Research in Science and Engineering, may 2016. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[5] CHEN, Kuan-yin; XU, Yang; XI, Kang; CHAO, H. Jonathan. Intelligent Virtual Machine Placement for Cost Efficiency in Geo-Distributed Cloud Systems, Communica-

tions (ICC), 2013 IEEE International Conference on IEEE. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[6] DOWNEY, A. B. "The structural cause of file size distributions" Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on 2001. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[7] JIN, Hao; PAN, Deng; XU, Jing; PISSINOU, Niki. Efficient VM Placement with Multiple Deterministic and Stochastic Resources in Data Centers. Global Communications Conference (GLOBECOM), 2012 IE. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[8] LAUREANO, Marcos. Máquinas Virtuais e Emuladores, Novatec Editora.

[9] MASTELIC, Toni; OLEKSIK, Ariel; CLAUSSEN, Holger; BRANDIC, Ivona; CLAUSSEN, Jean; PIERSON, Marc; VASILAKOS, Athnasios V. Cloud Computing: Survey on Energy e Efficiency. ACM Computing Surveys, Association for Computing. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[10] RASOULI, N.; MEYBODI, M. R.; MORSHEDLOU, H. Virtual Machine Placement in Cloud Systems Using Learning Automata. 2013 13th, Iranian Conference on Fuzzy Systems (IFSC). "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[11] RODRIGUES, Guilherme Da Cunha; CALHEIROS, Rodrigo N.; GUIMARAES, Vinicius Tavares; SANTOS, Glederson Lessa dos. Monitoring of Cloud Computing Environments: Concepts, Solutions, Trends, and Future Directions. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[12] SHI, Lei; BUTLER, Bernard; BOTVICH, Dmitri; JENNINGS, Brendan. Provisioning of Requests for Virtual Machine Sets with Placement Constraints in IaaS Clouds. Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on, IEEE. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[13] VERDI, F. L.; ROTHENBERG, C. Esteve; MAGALHÃES, R. Pasquiniand M. F. "Novas Arquiteturas de Data Center para CloudComputing". In 28th Brazilian Symposium on Computer Networks (SBRC), Shourt-course, Gramado, Brazil, survey May 2010. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[14] ZHENG, Qinghua; LI, Rui; LI, Xiuqi; SHAH, Nazaraf; ZHANG, Jianke; TIAN, Feng. Virtual Machine Consolidated Placement Based on Multi-Objective Biogeography-Based Optimization. Future Generation Computer Systems, Volume 54, January 2016, Pages 95–122, Science Direct. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[15] ZHOU; SHENG; LI." XtratuM/PPC: A Hypervisor for Partitioned System on PowerPC Processors". Springer Science + Business Media, New York, 27 Out. 2012. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[16] RAJEEV; KUMAR; GUPTA; PATERIYA, R. K. Survey on Virtual Machine Placement Techniques in Cloud Computing Environment, International Journal on Cloud Computing: Services and Architecture (IJCCSA),Vol. 4, No. 4, August 2014. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[17] ARMBRUST, Michael; FOX, Armando; GRIFFITH, Rean; JOSEPH, Anthony D.; KATZ, Randy H.; KONWINSKI, Andrew; LEE, Gunho; PATTERSON, David A.; RABKIN, Ariel; STOICA, Ion; ZAHARIA, Matei. Above the Clouds: A Berkeley View of Cloud Computing. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[18] BRUIN, Boudewijn de; FLORIDI, Luciano. The Ethics of Cloud Computing. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[19] CREASY, R. J. The Origin of the VM/370 Time-Sharing System. IBM J. RES. DEVELOP. VOL. 2 483 0 NO. 5 SEPTEMBER 1981. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[20] BIANZINO; CHAUDET; ROSSI; ROUGIER. A Survey of Green Networking Research. Institut TELECOM, TELECOM Paris Tech, CNRS LTCI UMR 5141, Paris, France 2010. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[21] OpenStack API Complete Reference, docs.openstack.org, acesso em 20 de Março, 2015. "O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)".

[22] OpenStackCommand-Line Interface Reference, docs.openstack.org, acesso em 20 de Março, 2015. "O conteúdo expresso