

OVERVIEW: COMPUTAÇÃO NA NUVEM, OPENSTACK E APLICAÇÕES DO TIPO BAG OF TASKS

FERNANDO ANGELIN¹; GERSON GERALDO H. CAVALHEIRO²

¹Universidade Federal de Pelotas – fangelin@inf.ufpel.edu.br

²Universidade Federal de Pelotas – gerson.cavalheiro@inf.ufpel.edu.br

1. INTRODUÇÃO

A utilização da computação em nuvem vem crescendo exponencialmente nos últimos anos, permitindo utilizar computadores de uma forma eficiente e remota. Neste tipo de modelo computacional, o uso dos recursos pode ser otimizado dinamicamente conforme as demandas de seus usuários. As mudanças na alocação dos recursos podem ser feitas em tempo real permitindo assim sistemas mais eficientes, evitando o superdimensionamento e/ou subdimensionamento dos recursos. Durante os últimos anos, foram consideráveis os progressos que aconteceram na computação em nuvem. Muitos fornecedores, como Microsoft, Google, Amazon e IBM, estão oferecendo serviços na nuvem, que incluem: Infrastructure as a Service (IaaS), Software as a Service (SaaS) e Plataforma as a Service (PaaS) (KIM, 2011).

A arquitetura de serviços em nuvem consiste em um hardware físico como base para um ambiente de virtualização, este que dará suporte ao SaaS e PaaS. O hardware físico compreende CPU, memória RAM, discos rígidos, interface de rede, entre outros. O ambiente de virtualização consiste em máquinas virtuais e dispositivos de entrada e saída virtualizadas, que são gerenciadas por um hypervisor. São exemplos de hypervisor: KVM, QEMU, Hyper-V e Xen. Existem dois tipos de hypervisor: o nativo e o hóspede. Um hypervisor nativo, também chamado de Tipo 1, executa diretamente no hardware físico, no modo mais privilegiado na execução das instruções. Um hypervisor hóspede, também chamado de Tipo 2, executa diretamente em um sistema operacional nativo, que executa no hardware físico, tendo menores privilégios na execução (KIM, 2011).

IaaS torna possível ao usuário acessar computação, armazenamento e infraestrutura de rede mantida em data centers provedores de IaaS. Com o IaaS, torna-se desnecessário que usuários individuais mantenham servidores, equipamentos de armazenamento de dados e equipamentos de rede, ou seja, os usuários contratam estes recursos nos servidores de nuvens. IaaS pode alocar diretamente hardware físico para os clientes ou alocar hardware virtualizado. Para hardware virtualizado, a IaaS oferece uma API para gerenciar as necessidades do cliente. Por meio do SaaS, o usuário tem acesso a softwares hospedados em um data center, como por exemplo, uma conta de e-mail. Dessa forma, o cliente não precisa se preocupar com atualizações de software. SaaS é provido em IaaS. O SaaS utiliza-se do IaaS por meio de uma API fornecida, para proteger e gerenciar a infraestrutura necessária para executar o SaaS e armazenamento de dados. PaaS tem por objetivo prover desenvolvimento de aplicações e ferramentas de execução, de modo que os usuários possam desenvolver e testar aplicações de forma mais rápida. PaaS também é provido em IaaS. Também utiliza-se da API fornecida pela IaaS. Na maioria das vezes, as aplicações desenvolvidas utilizando PaaS acabam ficando hospedados em IaaS.

O Openstack é um sistema operacional em nuvem que controla grandes conjuntos de computação, armazenamento e recursos de rede em um datacenter, tudo gerenciado por meio de um painel, com interface web, que dá aos

administradores o controle dos recursos que cada usuário necessita (OPENSTACK, 2014). Uma das suas principais características é a de ter código aberto e uma comunidade ativa no seu desenvolvimento. O Openstack permite a criação e a gestão de uma nuvem de forma simplificada, visando o melhor aproveitamento dos recursos disponíveis. O ideal deste sistema é prover escalabilidade, ou seja, prover computação em nuvem de forma elástica, tanto para nuvens públicas quanto para nuvens privadas, independente também do seu tamanho. O Openstack fornece IaaS para que SaaS ou PaaS sejam executados a partir dessa IaaS.

O Openstack tem uma arquitetura modular, ou seja, ele é composto por uma coleção de projetos desenvolvidos separadamente, mas todos sobre o mesmo padrão. Os componentes, como são chamados, abrangem os três pilares da computação na nuvem, que é a computação extensiva, armazenamento e recursos de rede, tornando-o mais fácil de implementar e operar a nuvem criada com Openstack. Cada componente tem um codinome referente, para fácil distinção. Cada componente é responsável por algum serviço que será provido.

Os principais componentes são: Nova, que provê máquinas virtuais sob demanda; Swift: provê um sistema de armazenamento escalável que suporta armazenamento de objetos; Cinder: provê blocos de armazenamento persistentes para as máquinas virtuais convidadas; Glance: provê um catálogo e um repositório para imagens de disco virtuais; Keystone: provê autenticação e autorização para todos os serviços Openstack; Horizon: provê uma interface de usuário baseada em web para todos os serviços Openstack; Neutron: provê conectividade de rede entre a interface de dispositivos geridos pelo Openstack; Ceilometer: provê um único ponto de contato para sistemas de faturamento; Heat: provê a organização para várias aplicações em nuvens compostas; Trove: fornece o serviço de banco de dados, tanto para bancos relacionais como não relacionais; Sahara: provê serviços de processamento de dados para os recursos gerenciados pelo Openstack (OPENSTACK, 2014). Ainda pode-se instalar outros componentes, conforme a demanda do usuário, como por exemplo o Neat, que provê migração de máquinas virtuais dentro de clusters, para fins de reduzir o consumo energético dos mesmos (BELOGLAZOV, 2013).

Como o projeto do Openstack é aberto, a comunidade é livre para criar novos componentes ou aprimorar componentes existentes, tanto para facilitar a gerência da nuvem criada com o Openstack quanto para testar a eficiência do sistema como um todo. Um exemplo disso, é a existência de um simulador de carga, que está presente no componente Neat (BELOGLAZOV, 2013). O Neat atua juntamente com o Nova, ou seja, no fornecimento de computação extensiva com as máquinas virtuais. A entrada deste simulador consiste em um “traço de execução”, onde é informado o trabalho que está chegando para essas máquinas virtuais, o tempo em que ele chegou e a quantidade de processamento que ele ocupa até a chegada do próximo trabalho. Desta forma é testado se o algoritmo de escalonamento aplicado pelo Nova atende à demanda das aplicações.

Um grande número de aplicações podem se beneficiar da computação em nuvem, dentre elas estão presentes as aplicações do tipo Bag of Tasks, que demandam alto poder de processamento. Bag of Tasks é um método de processamento paralelo em que as tarefas são colocadas em um “saco” de tarefas, e este é compartilhado por processadores clientes, chamados de “workers”. Cada worker, repetitivamente, pega uma tarefa e executa-a. Possivelmente, essa execução gera novas tarefas, que também são colocadas dentro do “saco” de tarefas inicial. A computação encerra quando o “saco” de tarefas estiver vazio e os workers estiverem ociosos (ANDREWS, 1999).

O padrão Bag of Tasks é aplicável nos casos em que nenhum subprograma dependa da computação de outro subprograma para ter a sua execução realizada, como por exemplo, aplicações que envolvam buscas por força bruta e, também, para a avaliação da função fitness em algoritmos genéticos (LUKE, 2003).

A arquitetura geral de uma abordagem Bag of Tasks consiste em ter um servidor central de tarefas que gerencia uma fila de tarefas que serão distribuídas para os processadores que estiverem disponíveis. Além de servir os processadores clientes com as tarefas, o servidor é responsável por integrar os vários resultados em uma solução final.

Esse padrão, tem uma limitação: quando o servidor realiza 10% do trabalho, é impossível alcançar um aumento de velocidade superior a 10. Este modelo simples de avaliação de desempenho com base na fração sequencial de um programa é chamado de Lei de Amdahl (AMDAHL, 1967). O resultado mais fundamental da Lei de Amdahl é a relação entre um limite superior de aceleração dada pela fração sequencial (HILL; MARTY, 2008). Para o servidor de tarefas não sofrer com gargalos de execução, ele deve fazer uma pequena fração do trabalho, incluindo o envio e recebimento de mensagens e a computação do resultado final.

2. METODOLOGIA

Foram estudados os conceitos, o funcionamento e os serviços providos pela Computação Nuvem. Também foi estudado o padrão Bag of Tasks: as características deste tipo de aplicação e como são executadas. A partir disso, está sendo estudado o Openstack: sua arquitetura modular e seus componentes.

3. RESULTADOS E DISCUSSÃO

O andamento deste trabalho se dará na construção de um parser que receberá como entrada informações de aplicações do tipo Bag of Tasks, como id, tempo de chegada, por quanto tempo ocupará a CPU e a porcentagem de uso da CPU e, terá como saída um “traço de execução” que será usado na entrada do simulador de carga do Neat. Também será feita a análise de desempenho com diferentes tipos de escalonadores presentes no Nova para este “traço de execução”.

4. CONCLUSÕES

Neste trabalho foi apresentado o conceito de computação em nuvem, sua arquitetura e os tipos de serviços que podem ser providos pela nuvem (IaaS, PaaS e SaaS). Também, foi explicado o funcionamento de um sistema operacional para controlar uma nuvem: o Openstack, que é composto por componentes que podem ser instalados conforme a necessidade do usuário. Dentro destes componentes adicionais, temos o Neat, que apresenta o simulador de carga para o Openstack. Também, foi conhecido o padrão Bag of Tasks: suas principais características de funcionamento e a principal limitação. A principal contribuição deste trabalho se dará no parser que transformará as informações das aplicações Bag of Tasks em um “traço de execução”, que será utilizado para entrada do simulador e assim, poder analisar o desempenho de diferentes tipos de escalonadores presentes no Nova.

5. REFERÊNCIAS BIBLIOGRÁFICAS

KIM, W. On Cloud Computing Adoption. **Journal of Web and Grid Services**, vol.7, no. 4, 2011.

KIM, W. Cloud Architecture: A Preliminary Look. **Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia**, Ho Chi Minh City, Vietnam, pages 2-6, 2011.

OPENSTACK. **Openstack: The Open Source Cloud Operational System**. About Openstack. Acessado em 20 de Julho de 2014. Online. Disponível em <https://www.openstack.org/software/>.

OPENSTACK. **Openstack Roadmap**. Openstack Capabilities. Acessado em 20 de Julho de 2014. Online. Disponível em <https://www.openstack.org/software/roadmap/>.

IBM. **Computação em Nuvem e Armazenamento com Openstack**. Recém-chegado da computação em nuvem: Openstack. Acessado em 21 de Julho de 2014. Online. Disponível em <https://www.ibm.com/developerworks/br/cloud/library/cl-openstack-cloud/>.

BELOGLAZOV, A. **Energy-Efficient Management of Virtual Machines in Datacenters for Cloud Computing**. 2013. Tese (Doutorado em Ciência da Computação) – THE UNIVERSITY OF MELBOURNE. Melbourne-Australia.

ANDREWS, G.R. **Foundations of Multithread, Parallel and Distributed Programming**. Tucson, AZ, USA: PEARSON, 1999.xx + 664p.

LUKE, E. A. **CS4163: Parallel Algorithm Analysis & Design**. Acessado em 20 de Junho de 2014. Online. Disponível em <http://www.cse.msstate.edu/~luke/Courses/sp03/CS4163/notes1.ps>

AMDAHL, G. M. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities. April 18-20, 1967, SPRING JOINT COMPUTER CONFERENCE, 1967, New York, NY, USA. **Proceedings...** ACM, 1967. P.483-485. (AFIPS'67(Spring)).

HILL, M.; MARTY, M. Amdahl's Law in the Multicore Era. **Computer**, [S.l.], v.41, n.7, p.33-38, July 2008.