

IMPLEMENTAÇÃO DO ARTIGO: ESTRATÉGIAS PARA ALOCAÇÃO DINÂMICA DE RECURSOS EM UM AMBIENTE HÍBRIDO DE COMPUTAÇÃO EM NUVEM

Ana Carolina Lopes da Silva¹, Thiago Meira Bernardes¹

¹Faculdade do Gama - Universidade de Brasília

Abstract. *This article describes the implementation of the article Estratégia para Alocação Dinâmica de Recursos em um Ambiente Híbrido de Computação em Nuvem, using the cloud tool Openstack. Will be describe the steps to install cloud tool, based on the tutorial [2] and how the implementation of the article was done.*

Resumo. *Este artigo descreve a implementação do artigo Estratégia para Alocação Dinâmica de Recursos em um Ambiente Híbrido de Computação em Nuvem [1], utilizando a ferramenta Openstack. Serão descritos os passos para instalação da ferramenta de nuvem, baseado no tutorial [2] e como foi feita a implementação do artigo.*

1. Introdução

Este artigo é uma proposta de trabalho prático da disciplina de Computação em Nuvem, ministrada na Universidade de Brasília - Faculdade do Gama. O trabalho propõe implementar um artigo, utilizando uma ferramenta de computação em nuvem. Aqui foi implementado o artigo Estratégia para Alocação Dinâmica de Recursos em um Ambiente Híbrido de Computação em Nuvem, utilizando o software Openstack.

A computação em nuvem busca deslocar toda infraestrutura computacional para a rede, assim o custo de software e hardware podem ser reduzidos. Existem quatro modelos de implementação na computação em nuvem: a nuvem privada, pública, comunitária e híbrida.

Sua arquitetura possui três camadas a de infraestrutura, plataforma e aplicação. A Figura 1, mostra a arquitetura da computação em nuvem (Zhang et al, 2010). Openstack é um exemplo de serviço da camada IaaS.

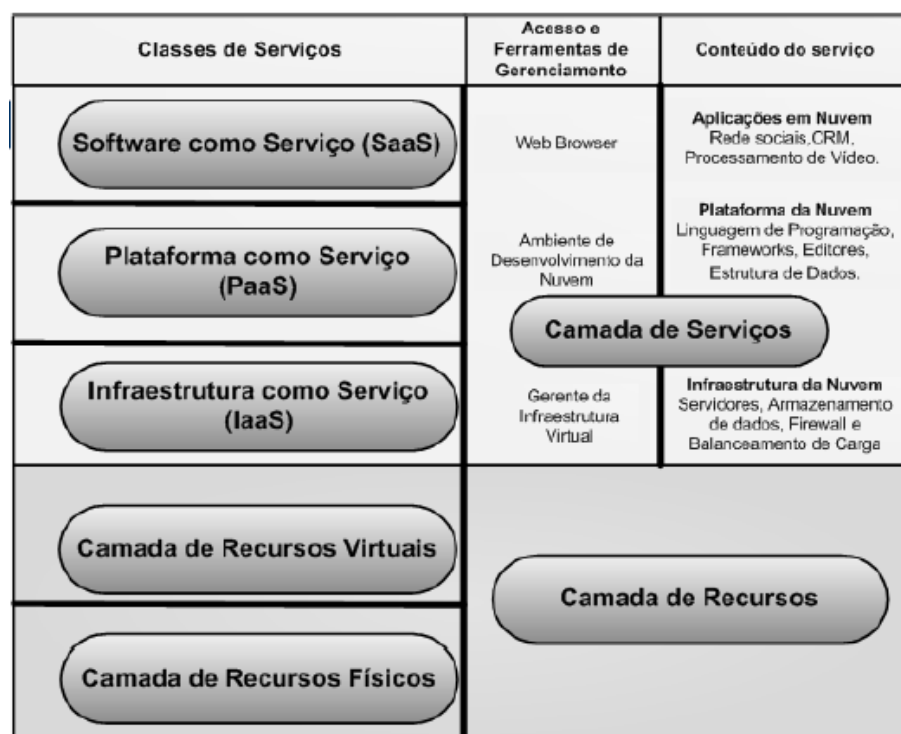


Figura 1 - Arquitetura da Computação em Nuvem.

2. Resumo do Artigo Implementado

O artigo *Estratégia para Alocação Dinâmica de Recursos em um Ambiente Híbrido de Computação em Nuvem*, procurou trabalhar na parte de prover recursos sob demanda para garantir a qualidade como serviço. Para isso foi proposta uma solução autônômica, que modifica dinamicamente a quantidade de recursos de CPU e instancia/remove Máquinas Virtuais. Para obtenção dos resultados, foram realizados experimentos em uma nuvem híbrida, com um balanceador de carga que distribui as requisições de uma aplicação web, a qual foi alinhada a diferentes estratégias para alocação dinâmica de recursos.

A solução proposta, adiciona a arquitetura FairCPU, que utiliza o conceito de Unidade de Processamento para alocar recursos de CPU a máquinas virtuais, o conceito de sensor e loop de controle, que fornece a capacidade de computação autônômica.

A Figura 2 mostra o elemento básico do loop de controle.



Figura 2 - Loop de Controle.

Na Figura 3, é possível visualizar a solução proposta, e sua relação com a arquitetura FairCPU. A abordagem foi dividida em duas partes: Agente e Controlador Autônômico.

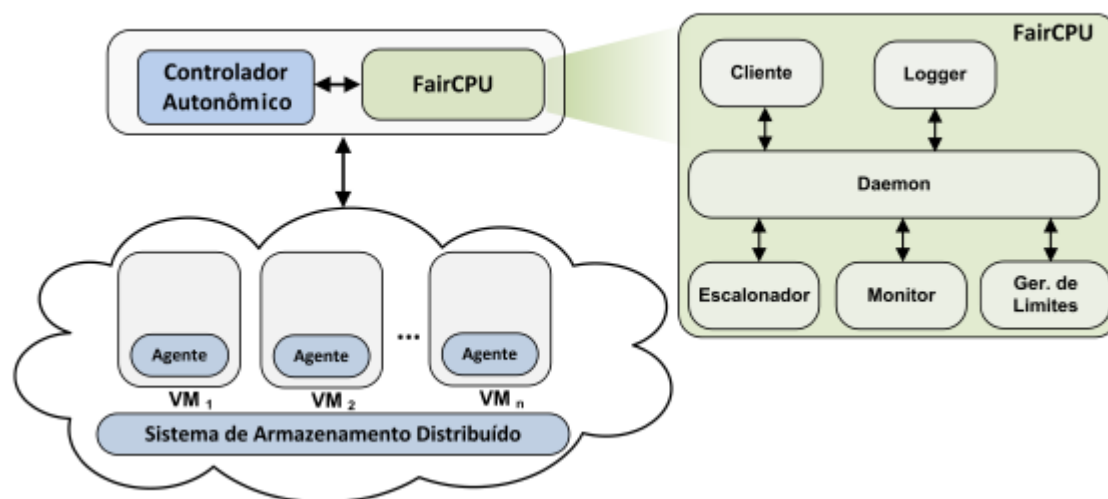


Figura 3 - Visão geral da solução proposta.

Para testar a solução foram realizados três experimentos, executados em dois ambientes, uma nuvem privada construída com o OpenNebula e uma nuvem pública da Amazon.

3. Openstack

Openstack é um software opensource, utilizado para controlar grandes pools de computação, armazenamento e recursos de rede através de um datacenter [3]. É usado para configurar e operar infraestrutura de computação e armazenamento em nuvem.

O gerenciamento é feito através de um dashboard ou da API do Openstack. A Figura 4 mostra como Openstack controla esses componentes.

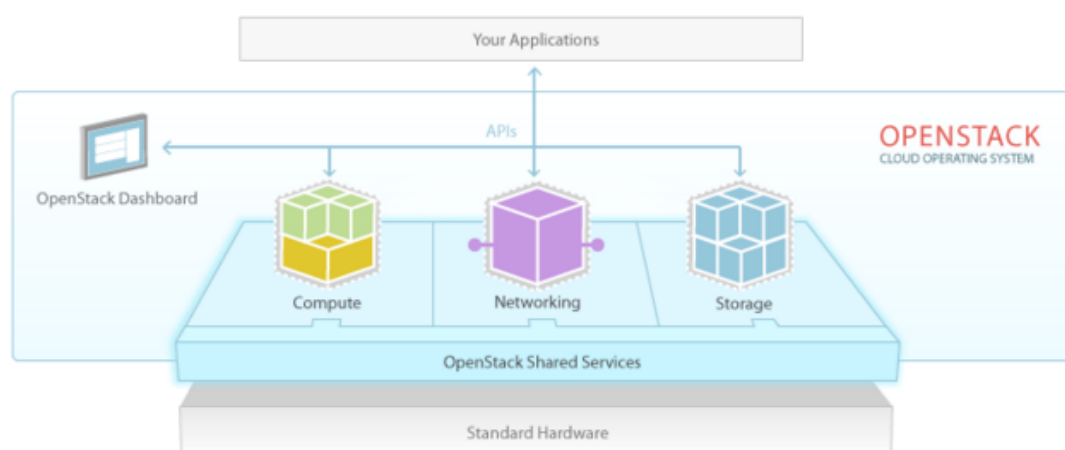


Figura 4 - Gerenciamento do Openstack.

4. Instalação do Openstack

A ferramenta foi instalada seguindo o tutorial [2]. Esse tutorial explica como instalar o Openstack, no Ubuntu Server 14.04. A instalação foi feita em uma máquina virtual criada no VirtualBox.

Para o Openstack foi utilizado o Canonical Openstack Autopilot, que é uma maneira fácil de construir e gerenciar uma nuvem privada OpenStack (<http://www.ubuntu.com/download/cloud/install-openstack-with-autopilot>).

Ubuntu Server:

- Download: <http://www.ubuntu.com/download/server>
- Siga o vídeo: Ubuntu Openstack install step 2: Ubuntu Linux server install. (<https://www.youtube.com/watch?v=vPPTNQ5dW1A>)

Após a instalação do sistema - Ubuntu Openstack install step 3: Ubuntu Linux MAAS software install

- Atualize os repositórios: `sudo apt update`, `sudo apt upgrade`, `sudo apt dist-upgrade`.
- Instale no ubuntu: `sudo apt install openssh-server`
- Instale o putty (SSH tool): <http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe> (no seu sistema operacional).
- Add stable juju repository: `sudo add-apt-repository ppa:juju/stable`
- Add stable MAAS repository: `sudo add-apt-repository ppa:maas-maintainers/stable`
- Add the openstack cloud stable repository: `sudo add-apt-repository ppa:cloud-installer/stable`
- `sudo apt update`
- `sudo apt install maas`
- Acesse o MAAS, use o IP do ubuntu (`ifconfig -a`): `http://<your_server_IP>/MAAS`
- `sudo maas-region-admin createadmin`: login como root e crie uma senha.

Instale o MAAS network and etherwake config - Ubuntu Openstack install step 4: Ubuntu Linux MAAS network and etherwake config (<https://www.youtube.com/watch?v=tQwcRmjJijI>).

- Edite a configuração da rede: `sudo nano /etc/network/interfaces`

*# This file describes the network interfaces available on your system
and how to activate them. For more information, see interfaces(5).*

*# The loopback network interface
auto lo
iface lo inet loopback*

*# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.150
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 192.168.1.1*

*auto eth1
iface eth1 inet static
address 10.1.1.100
netmask 255.255.255.0*

- Salve as alterações e reinicie o sistema: `sudo reboot now`
- Volte no tutorial do ubuntu e configure o cluster network interface
- Crie um chave SSH para o MAAS no ubuntu: `ssh-keygen -t rsa`
- `cat ~/.ssh/id_rsa.pub`
- Na interface do MAAS cole a chave gerada em Root>Preferences
- Instale o etherwake: `sudo apt install etherwake`
- Edite o etherwake template para remover o wakeonlan e adicionar o sudo access: `sudo nano /etc/maas/templates/power/ether_wake.template`

-- mode: shell-script -*-

Control node power through WOL, via `wakeonlan` or `etherwake`.
#*

*mac_address={{mac_address}}
power_change={{power_change}}*

*if ["\${power_change}" != 'on']
then
 echo "There is no way to power down a node through etherwake." >&2
 exit 1
#elif [-x /usr/bin/wakeonlan]
#then
/usr/bin/wakeonlan \$mac_address*

```

elif [ -x /usr/sbin/etherwake ]
then
    sudo /usr/sbin/etherwake -i eth1 $mac_address
else
    echo "No wakeonlan or etherwake program found." >&2
fi

exit 0

```

- Adicione o etherwake para o sudoers: `sudo nano /etc/sudoers.d/99-maas-sudoers`
- Adicione: `maas ALL= NOPASSWD: /usr/sbin/etherwake`
- Reinicie o sistema: `sudo reboot now`

A partir daí, o tutorial segue para adição de nós a sua nuvem, e instalação de outros recursos fornecidos pela ferramenta.

5. Implementação do Artigo

Como já foi citado anteriormente, para testar a solução proposta foram realizados três experimentos pelos autores da proposta, o primeiro foi uma nuvem privada e elasticidade vertical, o segundo experimento foi uma nuvem privada com elasticidade horizontal e o terceiro, uma nuvem híbrida e elasticidade horizontal.

No trabalho prático proposto na disciplina de Computação em Nuvem, teria que ser implementada a solução proposta no artigo escolhido pela dupla, para a comparação de resultados, sendo que as ferramentas utilizadas foram diferentes. Porém foram encontradas muitas dificuldades pela dupla para que essa tarefa fosse realizada. Uma delas foi a instalação do OpenStack utilizando uma máquina virtual. São muitos erros encontrados, mesmo seguindo um tutorial, e a maioria ocorre por causa da utilização de Virtual Machines ao invés de uma infraestrutura real física. Outros problemas vinham da Instalação de ambiente, configuração de outras ferramentas necessárias, como MAAS e juju que varia muita de acordo com as versões gerando uma documentação complexa.

Na implementação a instalação do ambiente foi concluída com sucesso. Foram configuradas três máquinas virtuais. Uma que é o servidor do openstack e outros dois nós que são instanciados e tem seus recursos alocados dinamicamente. O problema consistiu na hora de consumir os recursos dos nós subsequentes, o gerenciador encontra os nós, instala o serviço nos mesmos, mas na hora de fazer o comissionamento do serviço ocorrem erros.

Foram seguidos até outros tutoriais, como o que instala o Openstack pelo devstack, que é uma produção da comunidade OpenStack, onde a instalação é mais rápida, porém a dificuldade em acrescentar e manipular os nós e mexer com a ferramenta, são bem parecidos com o que foi seguido no outro tutorial, que utiliza o Canonical's OpenStack Autopilot, do Ubuntu.

6. Conclusão

Apesar da dupla não ter conseguido implementar de fato o artigo [1], o aprendizado adquirido sobre como se configurar uma infraestrutura computação em nuvem, tal como seu funcionamento, foi de grande valia. Essa parte prática da disciplina, deu a oportunidade de ter um contato real com a área de computação nuvem e como tudo o que foi estudado na parte teórica se relaciona na prática.

A maior dificuldade encontrada realmente foi a de ter que utiliza uma máquina virtual, sabe-se que é possível realizar o trabalho nessas condições, mas existem muitas limitações quando se trata de uma VM em computadores com recursos limitados.

7. Referências

- [1] Rego, P. A., Coutinho, E. F. e Sousa, J. N. *Estratégia para Alocação Dinâmica de Recursos em um Ambiente Híbrido de Computação em Nuvem*. XI Workshop de Computação em Nuvem e Aplicações.
- [2] <http://www.openstackbasement.com/>
- [3] <https://www.openstack.org/>