Openstack Lightwatch, um módulo para coleta de informações de consumo de cpu e memória para nuvens Openstack

Julio Machado[1], João Vítor V. T. de Oliveira[1], Vitor A. Ataídes[1], Maurício L. Pilla[1], Renata H. S. Reiser[1], Laércio L. Pilla[2]

[1]Centro de Desenvolvimento Tecnológico – UFPel
Pelotas – RS – Brasil

[2]Departamento de Informática e Estatística – UFSC
Florianópolis – SC – Brasil

{jmdsneto,jvvtdoliveira,vataides,pilla,reiser}@inf.ufpel.br, laercio.pilla@ufsc.br

Resumo. *Neste artigo é apresentado o Openstack Lightwatch, um módulo coletor de informações para nuvens Openstack que permite a coleta, em tempo real, de informações sobre consumo de CPU e memória das máquinas físicas e virtuais. O módulo foi avaliado em termos do atraso de chegada das informações coletadas ao banco de dados.*

1. Introdução

Com o rápido desenvolvimento das tecnologias de processamento e armazenamento, os recursos computacionais ficaram mais baratos e de fácil acesso. Essa evolução permitiu o desenvolvimento de um novo modelo computacional chamado de Computação em Nuvem [Mell and Grance 2011], no qual recursos computacionais são oferecidos como serviços e podem ser contratados ou desalocados. A Computação em Nuvem implementa um modelo de negócio orientado a serviços. Em outras palavras, os recursos de *hardware* e recursos no nível de plataforma são oferecidos como serviços sob demanda.

Openstack [Openstack 2015] é um Software Livre para computação em nuvem, implementando IaaS (*Infrastructure as a Service*). O Openstack oferece a capacidade de controlar uma grande quantidade de recursos de computação, de rede e de armazenamento, provendo também recursos sob demanda ao usuário. Ele permite que qualquer organização crie e ofereça serviços de Computação em Nuvem. O Openstack divide os nós em dois tipos: nó de Controle, responsável por executar os módulos que mantém o funcionamento do Openstack; e nó de Computação, responsável por executar as máquinas virtuais da Nuvem.

O objetivo desse trabalho é apresentar um módulo do escalonador de nuvens *Openstack Maestro*, o *Lightwatch*, o qual permite a coleta de informações do consumo de CPU e memória de máquinas virtuais (VMs).

Este artigo está organizado da seguinte forma. A Seção 2 apresenta os módulos e serviços do Openstack Maestro relevantes para este trabalho. A Seção 3 descreve a metodologia experimental e os resultados obtidos. Finalmente, a Seção 4 apresenta as conclusões e trabalhos futuros.

2. Openstack Maestro

Em [Ataídes et al. 2016] é apresentado um escalonador de VMs para nuvens Openstack, o *Openstack Maestro*. A Figura 1 apresenta a arquitetura do *Openstack Maestro*, composta

por serviços e módulos e também a distribuição entre os nós. Os serviços e módulos que constituem o *Openstack Maestro* são:

- **Htop-as-a-service** (serviço): Tem como objetivo disponibilizar as informações presentes no $software$ htop como um serviço;
- **Lightwatch** (módulo): Agregador de informações dos nós de computação; e
- **Lightwatch-API** (serviço): Armazena as informações coletadas pelo *Lightwatch* e as mantém acessíveis através de rotas de busca de informações.

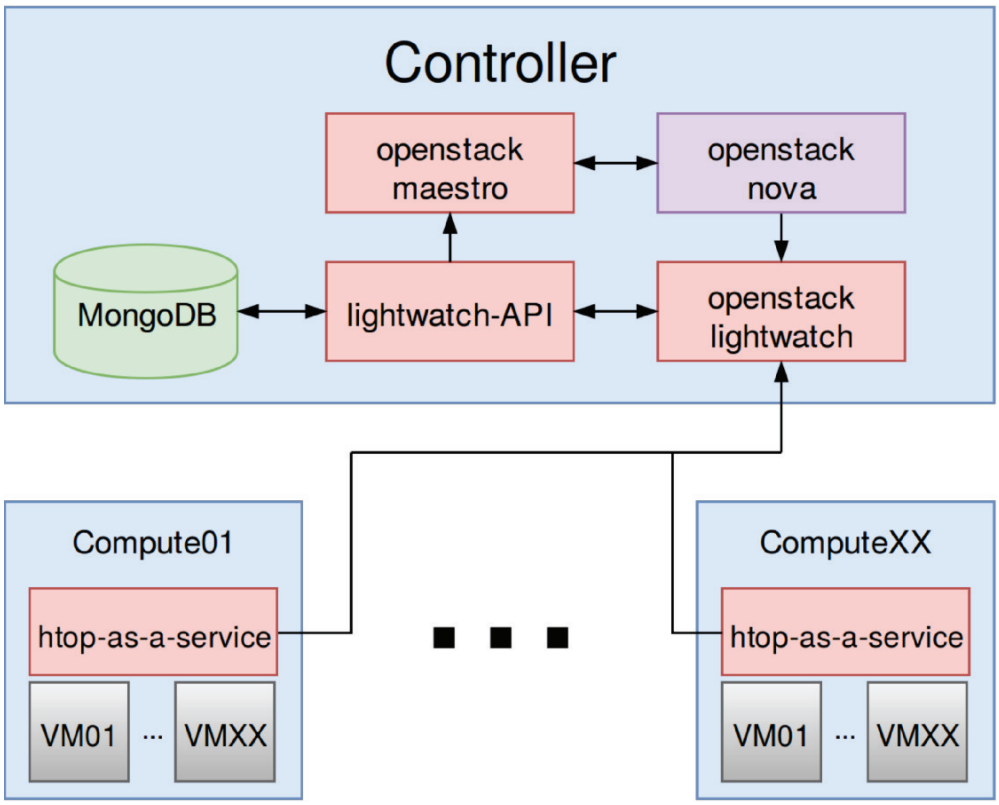

Figura 1. Arquitetura do *Openstack Maestro*

A seguir, os componentes acima são detalhados.

2.1. *Htop-as-a-service*

É um serviço tipo REST (*Representational State Transfer*) que tem como objetivo disponibilizar as informações presentes no *htop*. O desenvolvimento desse serviço foi feito utilizando-se *Node.js* com módulo *expressjs*. O serviço *Htop-as-a-service* possui duas rotas desenvolvidas, raiz e virt-top.

- Raiz: Essa rota é responsável por disponibilizar as informações de consumo de memória e CPU. O acesso a essas informações é feito através de uma requisição HTTP GET para o endereço IP das máquinas físicas onde o serviço *Htop-as-a-service* está sendo executado.
- Virt-top: De modo similar à rota Raiz, também disponibiliza informações de CPU e consumo de memória, mas das máquinas virtuais e também é nessecário uma requisição HTTP GET para o IP da máquina onde o serviço esta sendo executado, sendo necessário também adicionar a rota $url\ virt - top$.

2.2. *Lightwatch*

Através do *Htop-as-a-service*, coleta informações sobre o consumo dos nós e envia para o *Lightwatch-API*. A coleta e envio de informações é efetuada por uma subrotina que verifica VMs que foram instanciadas a cada 8 s. As coletas são feitas de modo constante para que se obtenha um histórico de uso da nuvem. Essa coleta de informações funciona de modo análogo ao Ceilometer do Openstack, com uma vantagem de armazenar as informações no banco de dados.

A arquitetura do *Lightwatch* é composta por seis classes:

- **Controller**: É a classe central da arquitetura, representa o nó *controller*. Tem como objetivo instanciar objetos de outras classes, centralizar e enviar as informações dos computes para uma API.
- **Compute**: Representa os nós de computação. Possui o objetivo de manter as informações atuais dos nós de computação.
- **VM**: Representa as VMs que estão sendo executadas dentro dos nós de computação. Tem por objetivo manter as informações das VMs atualizadas.
- **Htop**: É responsável por fazer a comunicação entre o controller com cada nó de computação do serviço *Htop-as-a-service*.
- **Lightwatch-API**: Responsável por se comunicar com a API do *Lightwatch*.

2.3. *Lightwatch-API*

A informação que é armazenada no banco é chamada de *log*. Um *log* do Lightwatch-API contém informações de CPU e memória de todos os nós de computação em determinado momento. No *log* também está presente a informação das VMs que estão sendo executadas em cada nó de computação e o consumo de cada VM.

Além disso, este serviço contém duas rotas, uma de busca e uma de inserção.

- **Inserção**: Esta rota é responsável por transformar a informação recebida pelo módulo *Lightwatch* em *log* e salva-lo no banco de dados. É necessária a operação de uma requisição HTTP POST para o ip da máquina onde o serviço está sendo executado.
- **Busca**: Esta rota é responsável pela busca de *logs* no banco de dados, utilizando um filtro que deve ser passado junto da requisição. Também é necessário realizar uma requisição HTTP GET para o ip da máquina onde o serviço está sendo executado.

3. Avaliação

Atualmente, a nuvem de experimentos utilizada no LUPS (*L*aboratory of Ubiquitous and Parallel Systems) é composta por nove servidores, sendo um deles designado *controller* e oito nós de computação. A configuração dos servidores é homogênea:

- Processador: Intel Core i5 2310 @2.9GHz
- Memória: 16 GiB
- Sistema Operacional: Ubuntu 14.04.3 LTS

O objetivo dos experimentos realizados foi avaliar o atraso do *Lightwatch* entre a captura de informações e a disponibilização em seu banco de dados.

Foram realizados 10 testes, consistindo cada um de alocação de 10 VMs na Nuvem. No momento da requisição da alocação, o tempo é salvo e após as 10 requisições os tempos são comparados com o tempo de chegada da informação no banco de dados. Como se pode observar na Tabela 1, a maior média dos testes chegou a 12,67 s e a menor em 9,67 s. A maior diferença encontrada foi de 20 s e a menor diferença foi de 7 s. O desvio padrão ficou entre 1,73 e 3,89 s.

Tabela 1. Dados sobre o atraso do *Lightwatch*.

Teste	Média (s)	Mediana (s)	Desvio Padrão
1	12,67	14	3,08
2	11,11	12	2,47
3	10,33	12	2,29
4	10,67	12	2,29
5	10,67	11	3,57
6	9,67	9	1,73
7	11,56	10	2,46
8	11,56	8	2,7
9	11,56	11	3,21
10	11,89	15	3,89

Como pode ser observado, a média dos testes em relação a diferença do tempo em que uma VM foi instanciada para o tempo que o banco de dados levou para ter percebido variou de 9,67 s a 12,67 s. Com o tempo de verificação das VMs instanciadas setado em 8 s, não foi necessário realizar verificações constantes, reduzindo então o consumo, pois o tempo que as instanciações foram feitas foi relativamente constante(o desvio padrão ficou entre 1,73 s e 3,89 s).

4. Conclusão e Trabalhos Futuros

Neste artigo, o módulo *Lightwatch* foi apresentado e seu atraso avaliado. Este módulo permite um maior controle do modo como as informações podem ser obtidas, devido à flexibilidade de sua implantação, assim, torna-se crucial para a obtenção de um melhor desempenho e redução de consumo em nuvens Openstack.

Na continuidade, propõe-se a otimização do *Lightwatch*, desenvolvendo-se uma memória *cache* para armazenamento dos *logs*, reduzindo-se então o tempo de acesso aos mesmos.

Referências

Ataídes, V. A., Pilla, M. L., and Pilla, L. L. (2016). Openstack maestro, um escalonador de máquinas virtuais para nuvens openstack. Universidade federal de Pelotas.

Mell, P. M. and Grance, T. (2011). Sp 800-145. the nist definition of cloud computing. Technical report, Gaithersburg, MD, United States.

Openstack (2015). Openstack. `http://openstacl.org`. Acessado em: 8/12/2016.