

Understanding, Discussing and Analyzing the OpenNebula's and OpenStack's IaaS Management Layers

Demétrius Roveda¹, Adriano Vogel¹, Dalvan Griebler^{1,2}

¹Laboratório de Pesquisas Avançadas para Computação em Nuvem (LARCC)
Faculdade Três de Maio (SETREM) – Três de Maio – RS – Brasil

²Programa de Pós-Graduação em Ciência da Computação Pontifícia Universidade
Católica do Rio Grande do Sul (PUCRS), Porto Alegre – RS – Brasil

roveda.demetrius@gmail.com, adrianovogel103@gmail.com,
dalvan.griebler@acad.pucrs.br

Abstract. *The OpenNebula and OpenStack tools have been used for large corporations and research centers to implement IaaS clouds. The management layer is an important element for the user and administrator because it deals with the resources monitoring, development support, orchestration, and integration with other cloud platforms and services. The goal of this paper is to discuss and analyze the differences in the management layer for pointing out advantages and disadvantages of the tools. The results demonstrated that OpenNebula is more restrict and focused on simplicity in almost all comparisons while OpenStack is fragmented, complex and robust.*

Resumo. *As ferramentas OpenNebula e OpenStack vem sendo utilizadas em grandes corporações e centros de pesquisa para implantação de nuvens IaaS. A camada de gerenciamento é um dos elementos importantes para o usuário e administrador, pois ela trata do monitoramento dos recursos, suporte ao desenvolvimento, orquestração e a integração com outras plataformas e serviços em nuvem. O objetivo desse trabalho é discutir e analisar as diferenças na camada de gerenciamento para apontar as vantagens e desvantagens das ferramentas. Os resultados demonstraram que a OpenNebula é mais restrita e focada na simplicidade em quase todas as comparações, enquanto a OpenStack é mais fragmentada, complexa e robusta.*

1. Introdução

A computação em nuvem é um paradigma que emergiu a partir da combinação de tecnologias consolidadas (virtualização, clusters computacionais, grids, redes de computadores) [Chandrasekaran 2014]. Ferramentas de computação em nuvem do modelo de infraestrutura como serviço (IaaS), oferecem recursos computacionais às camadas de plataforma como serviço (PaaS) e software como serviço (SaaS). A computação em nuvem vem recebendo uma crescente atenção e exploração pelo meio acadêmico e por organizações, motivados pelas principais vantagens oferecidas: elasticidade, flexibilidade, eficiência energética, fácil gerenciamento e customização, redução de custos e baixo investimento inicial [Buyya et al. 2013].

As ferramentas gerenciadoras de infraestrutura abstraem a complexidade computacional dos seus clientes através da provisão de serviços e recursos computacionais sob demanda [Buyya et al. 2013]. Porém, no ponto de vista de desenvolvedores, arquitetos e administradores de nuvem, as ferramentas envolvem um elevado nível de complexidade e desafios a serem superados. Neste artigo, a proposta é discutir e analisar duas ferramentas que são referência em nuvem de modelo IaaS, que é

segmentado em camadas e são responsáveis por controlar a infraestrutura virtual (hardware, rede, armazenamento). O modelo IaaS é a base da arquitetura da nuvem, que gerencia a infraestrutura e provê recursos para as camadas acima (PaaS e SaaS), as quais são extremamente dependentes.

No estudo de [Dukaric and Juric 2013], foram criadas sete camadas conceituais para uma ferramenta gerenciadora de infraestrutura de nuvem. Dentre essas camadas, a responsável pelo gerenciamento é formada pelos componentes (CLI *Tools*, APIs, *Dashboard* e *Orchestrator*), que executam funções essenciais (instanciar, excluir e gerenciar VMs, monitorar recursos, gerenciamento de energia e APIs), para usuários e administradores de ambientes de nuvem. Todos esses aspectos são extremamente importantes para implantar e utilizar um ambiente de nuvem. Porém, na literatura, não há estudos que se aprofundam especificamente na camada de gerenciamento (descrito na Seção 3), com um viés para o usuário e administrador de nuvem. O objetivo do trabalho também é analisar e discutir a arquitetura, funcionamento das ferramentas, interface com usuário, flexibilidade de integração através das APIs, suporte ao desenvolvimento na camada de gerenciamento e tecnologias para implantação. Sendo assim, as contribuições deste artigo são as seguintes:

- Uma extensão da metodologia de estudo sobre a camada de gerenciamento proposta por [Dukaric and Juric 2013].
- Uma análise detalhada das interfaces de gerenciamento das ferramentas OpenNebula e OpenStack.

O artigo está organizado em 5 seções. Na seção 2 é apresentada a base teórica de IaaS. Na subseção 2.3 é descrita a ferramenta OpenStack e na subseção 2.4 OpenNebula é evidenciada. A seção 3 relaciona o presente estudo com demais trabalhos encontrados na literatura. Além disso, a seção 4 demonstra as comparações e contribuições. Finalizando, na seção 5 são realizadas as conclusões do estudo e indicação trabalhos futuros.

2. Infraestrutura como Serviço (IaaS)

O conceito de IaaS (infraestrutura como serviço) tem como objetivo abstrair a complexidade computacional da infraestrutura (hardware, virtualização), controlando e gerenciando máquinas bem como servidores virtuais, chamados de instâncias. Dessa forma, os tradicionais *data centers* baseados em infraestrutura física, deram lugar a IaaS e infraestrutura virtual (computador, armazenamento e rede) [Chandrasekaran 2014]. Além disso, IaaS segue ideologias da computação em nuvem, com serviços sob demanda, pagamento de acordo com a utilização, elasticidade e flexibilidade. Também conta com características exclusivas como: acesso web aos recursos, escalabilidade dinâmica, infraestrutura compartilhada e máquinas virtuais com sistema operacional pré-instalado.

O autor [Buyya et al. 2013] destaca que a nuvem IaaS pode ser implantada em 3 diferentes modelos, como nuvem pública, privada e híbrida. A nuvem pública está disponível para todos, desde que se tenha um cartão de crédito para pagar de acordo com a utilização do serviço. Nesse modelo, o modo tradicional são grandes corporações (Amazon, Google, IBM, HP, Rackspace, Microsoft, etc.) vendendo máquinas virtuais pela internet. Por outro lado, a nuvem privada geralmente fica alocada dentro da estrutura física da organização, no próprio *hardware* e sem dependência de terceiros. Já a nuvem híbrida, é formada a partir de uma mesclagem entre os modelos privado e público. No

modo tradicional, determinados serviços (vitais) são executados na nuvem privada, enquanto os demais são executados na nuvem pública, atraídos pelo baixo investimento.

2.1. Ferramentas para Gerenciamento de IaaS

Ferramentas de gerenciamento de IaaS controlam os recursos da infraestrutura (computador, rede, armazenamento), oferecem ao administrador de nuvem diversos meios para melhorar a provisão de recursos, suporte aos usuários, automatizam rotinas e disponibilizam recursos de forma inteligente (alocação sob demanda e aumento da eficiência). Elas são extremamente relevantes para a nuvem como um todo e conseqüentemente para as camadas que estão acima dela (PaaS e SaaS). Essas ferramentas vão além da virtualização, pois elas também possuem e gerenciam diversos recursos (isolamento de recursos, VLANs, Banco de dados, segurança lógica do ambiente). Sendo assim, a virtualização é voltada para o hardware, enquanto a computação em nuvem utiliza a virtualização para alcançar vantagens (flexibilidade, melhor utilização do hardware, elasticidade, etc) e oferecer recursos computacionais compartilhados na forma de serviços através da internet.

Dentre diversas soluções de código aberto para IaaS, destacam-se OpenStack, pela aceitação corporativa e OpenNebula pela simplicidade e eficiência. Essas ferramentas possuem um gerenciamento nativo relacionado com a demanda que cada plataforma busca preencher. Com isso, existe uma grande variedade de formas de implantações (APIs, linguagem de programação, arquitetura, interface) e gerenciamento (recursos, serviços, usuários, entre outros), se compararmos entre elas. As ferramentas possuem uma diversificada gama de funcionalidades, suporte e opções de implantação, o que pode resultar em complexidades na instalação e gerenciamento. Essas ferramentas precisam oferecer uma interface (cli, web portal) que acompanhe a evolução dos recursos e seja eficiente nas interações com usuários e administradores de nuvem.

Em demandas específicas, o gerenciamento nativo de cada ferramenta não supre todas as necessidades, quando é necessário alcançar um alto nível do gerenciamento ou recursos avançados, como por exemplo: monitoramento avançado dos recursos ou venda de serviços (pague pelo uso). Quando isso acontece, pode-se utilizar ferramentas de terceiros, que ofereçam um exclusivo gerenciamento de nuvem [Shroff 2010].

2.2. Camada de Gerenciamento

A camada de gerenciamento na arquitetura de ferramentas IaaS foi introduzida na taxonomia unificada proposta por [Dukaric and Juric 2013]. Voltando-se para ferramentas provedoras de infraestrutura de nuvem. Percebe-se uma grande demanda pelo controle otimizado dos recursos e também nas interações que oferecem poder computacional (CPUs, memória, redes, armazenamento) aos usuários.

Por essa razão, soluções de gerenciamento contam principalmente com APIs e orquestradores que interagem com serviços para oferecerem recursos. Por outro lado, temos as interfaces (CLI, *dashboard*), que oferecem interações entre os usuários e o sistema para a alocação e provisionamento de recursos de forma flexível e customizável.

• CLI (Interface de linha de comando)

A CLI (*Command Line Interface*) ou simplesmente linha de comando é uma interface que oferece uma série de comandos para o administrador de nuvem interagir, controlar e

monitorar o sistema [Petersen 2006]. Em implantações clássicas, o administrador tem acesso a um terminal, no qual é possível a criação, modificação e exclusão de um ambiente virtual (máquinas, volumes, redes, usuários, grupos, cotas, entre outros). As CLIs das ferramentas são basicamente os seus componentes, que são usados para executar tudo o que é oferecido na *Dashboard* (interface gráfica), que de fato é apenas uma interface gráfica que abstrai a complexidade da parametrização de todas as CLIs da ferramenta. Teoricamente, tudo o que se pode fazer na CLI, poderia ser realizado pela interface gráfica (*Dashboard*), porém, opções avançadas são executadas unicamente na CLI (ex: recuperar VMs, restaurar volumes, configurações específicas de rede, etc.).

• APIs (Interface de Programação de Aplicações)

Um conceito que vem ganhando destaque pela necessidade de interoperabilidade entre diversos serviços é a API (Application Programming Interface), pois facilita interações de forma transparentes aos usuário. Em outras palavras, isso ocorre de uma forma abstrata à visão do usuário. Através dela, é possível que serviços e aplicações trabalhem de forma unificada, assim integrando ambientes e componentes distintos. APIs podem ser vistas como rotinas e protocolos que agem como um meio de campo na conexão entre dois sistemas (sistema operacional, web site) ou serviços específicos que conversam entre si para realizarem tarefas e processarem requisições, tanto de usuários, quanto de outros sistemas ou serviços independentes [Jacobson et al. 2011].

• *Dashboard*

As ferramentas estudadas (OpenNebula e OpenStack) possuem interface gráfica (*Dashboard*) que centraliza o monitoramento e gerenciamento dos recursos computacionais da nuvem. Ela tem a função de integrar e controlar os diversos componentes abstraindo a complexidade do gerenciamento dos serviços e recursos através de menus gráficos [Sabharwal:2013]. O administrador de nuvem tem o máximo de privilégios no gerenciamento na *Dashboard*, sendo assim, ele controla e define as cotas de recursos para os usuários do ambiente de nuvem. Além disso, possibilita ao administrador monitorar a alocação e utilização dos recursos computacionais, serviços e componentes. Por outro lado, um usuário de nuvem tem acesso a uma interface exclusiva na qual ele pode visualizar os recursos designados a ele, instanciar máquinas virtuais, tendo controle total desde a alocação de recursos (infraestrutura) até a execução de aplicações na nuvem.

• *Orquestrador*

Conforme [O'Neil 2014], o orquestrador surgiu da necessidade das ferramentas de nuvem do modelo IaaS que gerenciam nuvens públicas, para executar funções básicas de uma nuvem privada, mas em larga escala. Dessa forma, ele oferece um gerenciamento automatizado das tarefas de controle e utilização do ambiente de nuvem nas VMs (criação, monitoramento, atualização e exclusão). Dependendo da arquitetura da ferramenta, ele pode ser visto como uma API ou uma CLI.

2.3. Openstack

A ferramenta OpenStack é difundida e utilizada no cenário de nuvem IaaS de código aberto, devido ao seu projeto ser relativamente robusto [OpenStack:15]. Grandes empresas investem no desenvolvimento de novos recursos. Como resultado, esses

funcionam melhor utilizando produtos dessas empresas que são chamadas “stacks” [Llorente:15].

OpenStack pode atingir elevados níveis de flexibilidade, pois existem diversos serviços independentes, que são realizados através de APIs individuais. Consequentemente, OpenStack é definido ainda como uma ferramenta completa e customizável, porém, isso resulta em aumento da complexidade para implantação. Dentre vários serviços, destaca-se o nova (controle dos recursos computacionais), keystone (autenticação entre usuários e serviços), Cinder e Swift (controle do armazenamento), Neutron (serviços de Rede), Horizon (interface gráfica para interações), SQL (Banco de dados permanente) e RabbitMQ (comunicação e notificação entre os serviços). Toda a comunicação entre os serviços é feita através de APIs específicas [OpenStack:15].

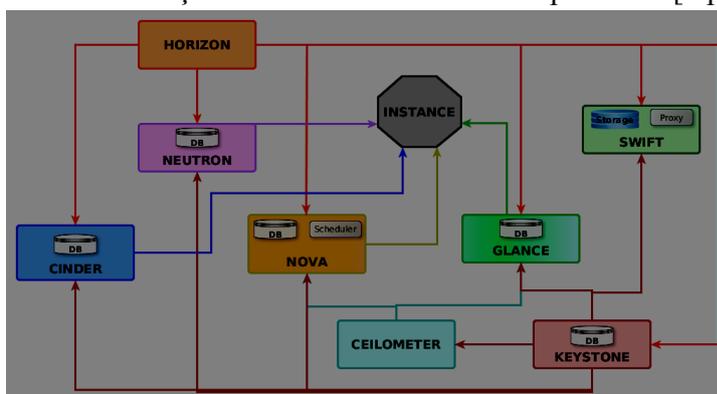


Figura1. ArquiteturaOpenStack. Adaptado de [OpenStack 2015].

Conforme a Figura 1, a robustez da OpenStack no processo de criação de instâncias é apresentado a seguir:

1. Autenticação do usuário é feita pela interface Web ou pela CLI do nova (cliente envia uma requisição HTTP POST para o keystone, que responde via Token, autentica, autoriza e faz o controle de acesso).
2. O nova é acionado, verifica se foi autenticada a requisição (keystone), e chama o banco de dados e o agendador via mensageiro (RabbitMQ).
3. Nova-schedule (agendador) encontra um nodo para alocar a instância através de filtros e balanceamento de carga, então chama o nova via mensageiro.
4. Nova recebe a requisição, inicia a instância e contata (RabbitMQ) o componente Neutron para serviços de rede.
5. O Neutron recebe a requisição e associa um IP e MAC para a instância.
6. O Nova contata via RabbitMQ o componente Cinder para criação de volumes. Após resposta, passa a instrução ao virtualizador para montar o volume na VM.
7. O Nova requisita uma imagem ao serviço glance e transfere para a instância.
8. Finalizando, o Nova inicia a VM, passando um comando XML ao virtualizador que é responsável pela execução. Ainda, o nova verifica periodicamente o status da instância e atualiza no banco de dados.

2.4. OpenNebula

A ferramenta OpenNebula é também *open source*. Ela surgiu em 2008 como uma ferramenta que pode ser utilizada para o gerenciamento de infraestrutura de nuvens públicas, privadas e híbridas. Ela é similar a ferramenta OpenStack, pois as duas gerenciam infraestrutura de nuvem. Entretanto, cada ferramenta foi desenvolvida para resolver problemas distintos. Esse projeto foi desenvolvido com um foco maior nas necessidades do cliente, principalmente com vantagens para ambientes de nuvem privada [OpenNebula:15].

A OpenNebula é uma ferramenta bem integrada, por isso, a sua instalação é consideravelmente "fácil", bem como a utilização da mesma. Ela não possui tantos recursos como a OpenStack, que foi desenvolvida para suprir outros tipos de necessidades. No entanto, OpenNebula é mais simples de ser gerenciada, pois não possui tantos componentes, resultando em perda de flexibilidade. Para implementações específicas, é possível utilizar ferramentas de terceiros para efetuar determinadas tarefas. Por exemplo, com o OpenVSwitch é possível criar redes e roteadores virtuais bem como implantar regras de *firewall* para as VMs.

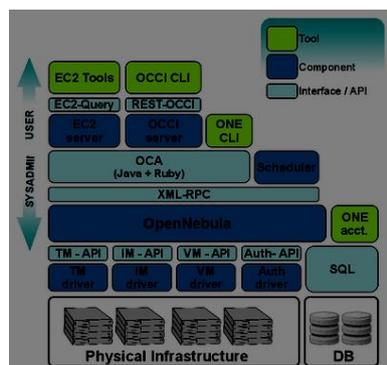


Figura 2. Arquitetura OpenNebula. Extraído [OpenNebula:15].

Como se pode ver na Figura 2, a estrutura do OpenNebula é dividida em camadas, componentes e APIs. Na base, tem-se representada a infraestrutura física (hardware), sob esta camada tem (TM Driver – Transfer Manager Driver, IM Driver – Information Manager Driver, VM Driver – Virtual Machine Driver e Auth Driver – Authentication Driver), que interpretam as funções (eg., criação de volumes, VMs, entre outros) para a camada principal da OpenNebula. Tais instruções são passadas através de APIs da camada de abstração de recursos (TM-API, IM-API, VM-API e Auth-API).

O gerenciador de banco de dados SQL (MySQL ou SQLite) através da API SQL (pertencente à camada de suporte), armazena informações através de um contador do consumo de recursos (ONE accounting), que é aplicado em ambientes de nuvem pública. Analisando o taxonomia [Dukaric and Juric 2013], possibilita-se elencar ainda as APIs pertencentes à camada de gerenciamento (XML-RPC, OCA, EC2-Query e REST-OCCI), que serão abordadas na seção 4. O componente *Scheduler* (agendador) é responsável por alocar as VMs nos nodos. Isso pode ser feito de duas formas: manualmente: o usuário escolhe em qual nodo a VM será instanciada, ou dinamicamente: ocorre quando a OpenNebula escolhe o nodo que está com a menor carga para instanciá-la. A CLI da ferramenta será discutida na seção 4. Já a ferramenta EC2 Tools é disponibilizada pela Amazon, é utilizada para integração com a nuvem OpenNebula. A OCCI CLI é uma ferramenta para demandas específicas no monitoramento e gerenciamento remoto de

serviços na nuvem OpenNebula.

3. Trabalhos Relacionados

A computação em nuvem vem recebendo muita atenção da literatura nos últimos anos, diversos autores pesquisam e publicam materiais sobre. No estudo de [Alam et al. 2015], é feito uma pesquisa sobre ferramentas e soluções para nuvem, onde destaca-se ferramentas e softwares de código aberto. Ainda, este estudo concluiu que a nuvem traz diversas vantagens e facilidades quando implantada seguindo as melhores práticas, agregando valor aos negócios.

No estudo de [Wen et al. 2012], compara-se ferramentas de código aberto gerenciadoras de infraestrutura de nuvem (OpenStack e OpenNebula). Analisando a arquitetura e o suporte que elas possuem para virtualizadores e recursos de segurança. Ainda apresenta recomendações de acordo com o tipo de demanda de cada cliente. Esse estudo concluiu que OpenStack é uma solução para implantações complexas e robustas e ambientes de grandes corporações, enquanto OpenNebula é mais indicado para *data centers*, instituições públicas e privadas.

Outro estudo interessante é apresentado por [Dukaric and Juric 2013], o qual a partir de uma consistente pesquisa bibliográfica propôs uma taxonomia unificada para ferramentas gerenciadoras de infraestrutura de nuvem. Tal pesquisa elenca sete camadas conceituais e fundamentais que deveriam estar presentes em todas as soluções para teoricamente serem consideradas completas. Ainda, as principais soluções de nuvem pública e soluções de código aberto são comparadas através da taxonomia sugerida colaborando dessa forma com informações precisas e úteis para a implantação e utilização de ambientes de nuvem.

No estudo de [Thome et al. 2013] foram elencadas ferramentas de código aberto para a construção de nuvens IaaS, considerando nas soluções de nuvem o suporte para implantação (virtualização, armazenamento e rede), e características das ferramentas (monitoramento, segurança, gerenciamento de energia e balanceamento de carga). Nesse estudo, as características das soluções são comparadas, buscando encontrar diferenças entre as ferramentas bem como pontos fortes e fracos de cada uma.

As pesquisas relacionadas deixam ainda aspectos importantes das ferramentas de nuvem a serem investigados. O estudo de [Thome et al. 2013], [Wen et al. 2012] e [Alam et al. 2015] são restritos às características e arquitetura das soluções, apresentando uma visão geral delas. Em [Dukaric and Juric 2013] é apresentado o estado da arte com uma visão geral de IaaS, sem um nível elevado de detalhes sobre o gerenciamento das ferramentas.

Dessa forma, este estudo é voltado exclusivamente para serviços e componentes da camada de gerenciamento (APIs, CLIs, *Dashboard* e Orquestradores) das ferramentas OpenNebula e OpenStack. Com base na taxonomia proposta por [Dukaric and Juric 2013], foi analisada a camada de gerenciamento das ferramentas, incluindo uma comparação da usabilidade e suporte a interações para usuários e administradores de nuvem. Outro aspecto estendido, foi o suporte a tecnologias de infraestrutura em cada ferramenta.

4. Discussão e Análise

Nesta seção é apresentada a discussão e comparação entre as ferramentas OpenNebula e

OpenStack, levando-se em conta o suporte e flexibilidade de controle e acesso. O levantamento realizado baseou-se na taxonomia unificada de IaaS [Dukaric and Juric 2013]. Para a coleta de informações, fontes de documentação oficial de cada ferramenta foram consideradas. Com as pesquisas de [Thome et al. 2013], [Wen et al. 2012], [Alam et al. 2015] e experiências em trabalhos anteriores de [Maron et al. 2014], [Maron et al. 2015], foi possível fundamentar o que é importante para a camada de gerenciamento em soluções de nuvem IaaS. Para sumarizar o conhecimento, foram tabuladas as características de cada ferramenta e classificadas com detalhes voltados a usabilidade e relevância do gerenciamento do ponto de vista do usuário e administrador de IaaS.

Os usuários de IaaS são geralmente administradores de redes e sistemas, pois eles irão determinar qual é a infraestrutura necessária para rodar suas plataformas e, se for o caso, suas aplicações. Utilizando um ambiente de nuvem, pode-se ter vantagens como: liberdades na customização e elasticidade para seus servidores, que é de grande importância para usuários de IaaS. Por outro lado, dependendo do cenário e da ferramenta utilizada, o administrador de IaaS irá preparar a estrutura computacional de nuvem, gerenciar as cotas de usuários, monitorar e gerenciar os recursos computacionais do hardware. Uma visão geral é ilustrada e representada na Figura 3.



Figura3. Visão geral do usuário e administrador de IaaS.

Na Tabela 1 são comparados itens como interface do usuário (UI), segurança da ferramenta, se possuem painel de notificações e qual o nível de customização referente ao gerenciamento de VMs (máquinas virtuais).

O item Dashboard refere-se ao portal web disponível para interações. Dessa forma, a ferramenta OpenNebula utiliza o Sunstone como interface gráfica e a OpenStack usa o componente Horizon para essa finalidade. Dentre suas particularidades, foi comparado a segurança de cada ferramenta, bem como é tratado o acesso externo a elas. As VMs da ferramenta OpenNebula podem ser acessadas de forma segura através de um websocket SSL, que encripta com certificados (OpenSSL), enquanto na ferramenta OpenStack o acesso seguro pode ser habilitado utilizando SSL e certificados de autoridade. As ferramentas suportam também, a autenticação (método em que os usuários se logam no sistema), a autorização de cada usuário (limite de acesso aos recursos da ferramenta), e se possuem um painel de notificações (centralizador de alertas e informações), que é bem difundido em portais web de soluções de nuvem.

Outro ponto imprescindível referente à execução de VMs na nuvem é o gerenciamento e controle dos recursos. Tal tópico é apresentado em forma de comparação

entre as soluções na Tabela 1. Se as ferramentas possuem suporte a *Snapshots* (salvar o estado da VM) e suporte à migração de VMs (alterar o host onde a VM está alocada em tempo real e sem interrupções). Um fator que conta a favor da OpenNebula é que na primeira tela após o login do usuário, a *dashboard* apresenta uma visão geral do ambiente, como informações referentes a quantidade de recursos que está em uso (CPU, memória, armazenamento), quantidade de VMs instanciadas entre outros. Já a OpenStack não possui esse recurso que é muito útil ao administrador de IaaS.

Tabela1. Comparação dos elementos da Dashboard.

<i>Dashboard (UI)</i>	<i>OpenNebula (Sunstone)</i>	<i>OpenStack (Horizon)</i>
Acesso Seguro	noVNC encryption socket	Secure HTTPS proxy certificate authority (CA)
Autenticação	usuário/senha	usuário/senha
Autorização	Permissões por grupos de segurança	Permissões por grupos de segurança
Painel de Controle	/	/
Escolha da hospedeiro para instanciar a VM	Nativo Sunstone	/
Suporte a <i>Snapshots</i>	Nativo Sunstone	Nativo Horizon
Suporte à migração de VMs	Nativo do Sunstone (Implementação habitual do OpenNebula)	Nativo Horizon (utilizando armazenamento compartilhado)

Na comparação dos elementos de cada ferramenta, nota-se que as duas possuem recursos que atendem aos pontos levantados, mas apenas a ferramenta OpenNebula oferece além da escolha automática, a flexibilidade da escolha de qual servidor a VM será instanciada, o que aumenta os níveis de flexibilidade e customização. Enquanto a ferramenta OpenStack, faz esse processo de forma transparente ao usuário, de tal forma que o servidor é selecionado baseado no que tiver a menor carga (menos instâncias e recursos alocados no host). Ambas as ferramentas não possuem painel de notificações.

Na OpenNebula, a divisão da quantidade de recursos e tipos de serviços é implementada para cada cliente de nuvem na forma de quotas, para limitar a quantidade de recursos computacionais (processadores, memória, armazenamento). Adicionalmente, é definido se o usuário pode ou não explorar serviços disponíveis (orquestrador, redes, entre outros). Em contraste, na ferramenta OpenStack, todo o controle ocorre na forma de projetos (*data center* virtual), ou seja, projetos são criados para delimitar a quantidade máxima de recursos (quotas), e os usuários são criados e vinculados à projetos para competir e compartilhar recursos. A Tabela 2 demonstra também o controle exercido pelo usuário e administrador de IaaS através da interface gráfica das ferramentas, nas duas o administrador tem controle e permissão total nos tópicos comparados, e as permissões dos usuários são relativamente parecidas.

Nota-se que a ferramenta OpenNebula é mais flexível aos usuários, a criação de instâncias também é mais eficiente, pois possui uma interface exclusiva para os usuários, tendo um alto nível de customização dos recursos (redes virtuais, volumes, quantidade de CPUs e memória), e há ainda possibilidade de monitorar a utilização dos recursos individualmente, porém precisa-se de um controle rigoroso nas permissões (quotas) para evitar eventuais problemas de segurança. O mesmo cuidado é necessário na ferramenta OpenStack, pois o usuário de nuvem pode escolher em qual Cluster computacional a VM será instanciada e também tem privilégios para gerenciar os volumes e controle das redes virtuais, o que não é possível com a outra ferramenta comparada. O usuário da OpenNebula tem mais flexibilidade com as *templates* (modelo de configuração das VMs), pois consegue editá-las.

Dessa forma, OpenNebula é mais indicada para nuvens privadas e *data centers* por ser centralizada e simplificada enquanto a OpenStack é mais indicada para grandes corporações construírem grandes e complexas nuvens ou para provedores de nuvem pública devido a flexibilidade que oferece ao suportar diversos componentes.

Tabela2. Controle da interface gráfica. Ações possíveis como administrador e usuário na interface são: Criar (C), Acessar (A), Editar (E), Deletar (D).

Ações	OpenNebula		OpenStack	
	Administrador	Usuário	Administrador	Usuário
Usuários	C A E D	- A - -	C A E D	- A - -
Grupos	C A E D	- - - -	C A E D	- A - -
ACLs	C A E D	- A - -	C A E D	- A - -
Nodos	C A E D	- A - -	C A E D	- A - -
Clusters/Agregador de Nodos	C A E D	- A - -	C A E D	- A - -
Discos/Volumes	C A E D	- A - -	C A E D	C A E D
Redes Virtuais	C A E D	- A - -	C A E D	C A E D
Zonas de disponibilidade	C A E D	- A - -	C A E D	- A - -
Imagens	C A E D	- A - -	C A E D	- A - -
Templates/Flavors	C A E D	- A E -	C A E D	- A - -
Serviços de Infraestrutura	C A E D	- A - -	C A E D	- A - -
Cotas	C E	- -	C E	- -
Monitorar recursos (por nodo e VMS)	A	A	- -	- -

A Tabela 3 evidencia todas as APIs da ferramenta OpenNebula e conforme sua arquitetura, ela possui a XML-RPC API que fornece métodos para interações entre o Sunstone (interface gráfica e as CLIs). Ela faz uma RPC (Remote Procedure Call), pois chama remotamente as CLIs que invocam outros métodos remotos nos hosts. A OpenNebula possui duas APIs OCA (OpenNebula Cloud API) para desenvolvimento do core da ferramenta que são: Ruby OpenNebula Cloud API e Java OpenNebula Cloud API, essas suportam as linguagens de programação Ruby e Java.

Para orquestração é utilizada o OneFlow Server API, que controla e monitora serviços ou aplicação compostos por diversas VMs, ou seja, serviço de controle e gerenciamento em larga escala a nível de infraestrutura (ligar e desligar diversas VMs de um grupo ou adicionar um vCPU). Como APIs de integração, a OpenNebula possui a EC2-Query API que é utilizada para integração da nuvem OpenNebula (privada) com a nuvem da Amazon (pública), formando uma nuvem híbrida. A OpenNebula suporta também a OCCI¹ (Open Cloud Computing Interface), que é uma API padrão para utilização em nuvens públicas e trata do faturamento, gerenciamento de tarefas e monitoramento remoto de serviços baseados em IaaS. Diversas APIs podem ser desenvolvidas e integradas com a OpenNebula, como no estudo apresentado por [Righi et al. 2015].

Tabela 3. OpenNebula APIs

API/Componente	Funções
	API para gerenciamento na Interface Gráfica
XML-RPC API	Fornecer métodos para interações com as CLIs, pois o Sunstone (<i>Dashboard</i>) utiliza essa API para executar as tarefas na nuvem.
	OpenNebula Cloud API (OCA) – para desenvolvimento do core OpenNebula
Ruby OpenNebula Cloud API e Java OpenNebula Cloud API	Essas duas APIs fazem parte do OCA, que foi criada com o viés para desenvolvedores, possibilitando a criação de outras APIs para o OpenNebula
OneFlow Server API	Controlar e monitorar aplicações ou serviços compostos por VMs conectadas.
	APIs de Integração
EC2 Query API	API de integração com a nuvem da Amazon e recursos de gerenciamento de imagens, instancias, volumes, <i>snapshots</i> , endereçamento IP, etc.
	API externa para gerenciamento remoto de serviços baseado em IaaS, a última versão

¹ <http://wiki.opennebula.org/ecosystem:occi>

OCCI (Open Cloud Computing Interface)	dessa API foi desenvolvida com ênfase na interoperabilidade, também utilizada para gerenciamento de nuvem, ainda possui recursos para gerenciar outros modelos de nuvem como PaaS e SaaS.
---------------------------------------	---

Do ponto de vista de desenvolvedores, a XML-RPC API é utilizada para desenvolver bibliotecas especializadas para aplicações em nuvem, ou também quando se vê necessário uma interface de baixo nível com o core da OpenNebula. Já as OCA, são úteis para desenvolver serviços avançados de IaaS que precisam de acesso completo à funcionalidades da OpenNebula.

A Tabela 4 apresenta as APIs da documentação oficial do OpenStack², pois a ferramenta possui uma arquitetura única, sendo totalmente fragmentada em APIs, de tal forma que cada uma responde por uma função específica e ainda, as APIs funcionam totalmente interligadas entre os componentes. Cada serviço na ferramenta OpenStack é composta por pelo menos uma API. Também é apresentada a EC2 compatibility API que é utilizada para integração da nuvem OpenStack com nuvens da Amazon, possibilitando a utilização de nuvens híbridas. Embora não esteja claro na documentação da OpenStack, a API OCCI (Open Cloud Computing Interface), pode ser utilizado em conjunto com a ferramenta OpenStack para um gerenciamento mais efetivo dos recursos computacionais. Através do desenvolvimento de novas APIs, diversas demandas podem ser alcançadas, um exemplo é uma API da Cisco³ para melhorar os serviços de rede.

Tabela 4. OpenStack APIs

<i>API</i>	<i>Função</i>
Block Storage	Gerencia volumes e <i>Snapshots</i> de armazenamento em bloco
Compute	Lançar instancias e controlar o armazenamento
Database Service	Gerencia instancias e serviços de banco de dados
Data Processing	Produz operações de dados (mineração, tratamento, análise, estatísticas)
Identity	Controla a autenticação e autorização
Image Service	Controla as permissões dos usuários nas interações com imagens
Networking	Gerencia as redes do ambiente virtual de nuvem
Object Storage	Gerencia o sistema de armazenamento de objetos
Telemetry	Gerencia o controle da utilização dos recursos
Orchestration	Idealiza a orquestração na nuvem
EC2 compatibility	Oferece suporte para cargas de trabalho serem executadas na nuvem da Amazon

A Tabela 5, compara as CLIs das ferramentas OpenNebula e OpenStack, evidenciando as diferenças das arquiteturas. A OpenNebula possui diversas CLIs, cada uma com sua função e todas são manipuladas pelo serviço ONE através do usuário oneadmin (usuário administrador padrão do sistema). Na ferramenta OpenStack, também é encontrado diversas CLIs, porém, nessa ferramenta cada uma é usada para controlar um projeto de API, ou seja, comandos com controle total e flexível sobre as APIs. Por outro lado, a OpenNebula é uma ferramenta mais centralizada, conservadora e conseqüentemente com controle de permissões mais rigoroso, e as CLIs não oferecem tantas opções quanto na OpenStack. Como mostrado na Tabela 4, existem diversas APIs no OpenStack que são utilizadas para oferecer serviços. Na Tabela 5, a parte com fundo cinza serve para diferenciar, expressar que existe contraste entre as ferramentas que impossibilitam a comparação igualitária.

A ferramenta OpenStack utiliza o Ceilometer para medir a utilização dos recursos por usuário, na OpenNebula, esse monitoramento é feito por host (oneacct). Já o Heat é o orquestrador da OpenStack, dessa forma, não sendo possível a comparação com as CLIs da OpenNebula. Considerando que as CLIs são únicas de cada ferramenta, a OpenNebula possui CLIs para: distribuição em clusters (onecluster) e zonas de disponibilidade

² <http://developer.openstack.org/api-ref.html>

³ <http://www.cisco.com/docs/switches/api/openstack>

(onezone), gerenciamento de *data center* (onevdv), controle de ACLs (oneacl) e criação de *templates* para usuários (onetemplate).

Do lado OpenStack, a robustez da ferramenta fica evidenciada também nas CLIs, exemplos específicos que se destacam nessa solução são: CLI para um cluster Hadoop (tratamento de grandes volumes de dados), suporte para nuvem híbrida (euca2ools), para infraestrutura Bare metal (Ironic), CLIs específicas para armazenamento de objetos (Swift) e o orquestrador (Heat).

Tabela 5. CLIs presentes nas ferramentas

<i>OpenNebula</i>		<i>OpenStack</i>		
<i>CLI</i>	<i>Descrição</i>	<i>CLI</i>	<i>Descrição</i>	<i>Serviço</i>
oneimage	Gerencia as imagens	Glance	Gerenciar imagens	Imagem
onedatastores	gerencia volumes virtuais	Cinder	Gerenciar volumes	Armazenamento em blocos
onedb	Ferramenta de banco de dados	Trove	Criar e gerenciar bancos de dados	Banco de dados
onevnet	Redes virtuais	Neutron	Configurar e gerenciar redes	Redes virtuais
onehost	Gerencia os hosts da OpenNebula	Nova	Interagir e controlar recursos computacionais	Compute
oneuser onegroup	Gerencia os usuários e grupos da OpenNebula	keystone	Gerenciar usuários, regras e credenciais	Serviços de identidade
onecluster	Gerencia Clusters	Sahara	Gerenciar Cluster de Hadoop	Processamento de dados
onevdv	Gerencia Data Center virtual na OpenNebula	Bare Metal	Controle da infraestrutura bare metal	Ironic
oneacl	Gerencia as ACLs	Euca2ools	Interação com Amazon EC2	EC2 API
onevm onezone	Gerencia as VMs da OpenNebula Gerencia o segregamento entre zonas na ferramenta	Swift	Administração de objetos (acesso, tratamento, atualizações)	Armazenamento de objetos
		Heat	Controlar pilhas de: templates, VMs, redes, volumes	Orquestrador
onetemplate	Gerencia as templates	OpenStack	Interface de interações para o cliente da OpenStack	Cliente
oneacct	Monitora o status dos nodos da nuvem	Ceilometer	Medidor da utilização dos recursos	Telemetria

Além do suporte à APIs, CLIs, e UI, as soluções de nuvem precisam apresentar suporte para uma implantação robusta e flexível, pois torna-se uma tarefa demorada e complexa escolher a melhor solução para atender melhor às demandas de acordo com os processos internos e necessidades da TI. Também é necessário explorar as soluções que a nuvem oferece, principalmente em soluções de código aberto. O desenvolvimento de ferramentas são altamente flexíveis e robustos, com amplo suporte para virtualizadores, sistemas de armazenamento, redes de computadores e serviços de integração são um diferencial entre diversas soluções para a nuvem.

Conforme a Tabela 6, a Openstack se mostra mais flexível para implantações de ambientes de nuvem. O suporte à plugins de rede é mais amplo na OpenStack, trazendo aos administradores de nuvem mais alternativas e controle. No armazenamento, a OpenStack é mais robusto, mais tecnologias são suportadas e o serviço de armazenamento apresenta uma melhor integração. O suporte à virtualizadores também é mais completo com a OpenStack, o que pode trazer ganho de desempenho em determinadas aplicações devido ao impacto entre as diferentes soluções de virtualizadores [Gupta and Kalé 2013]. Além disso, a OpenStack suporta virtualização a nível de sistema operacional (contêiner), o que pode trazer um melhor isolamento de recursos e desempenho para aplicações executadas na nuvem [Gupta et al. 2013].

Tabela 6. Comparação do suporte e flexibilidade para implantação

<i>Soluções</i>	<i>OpenNebula</i>	<i>OpenStack</i>
Suporte a Virtualizador	Xen, KVM, Vmware	Hyper-V, VMware, XenServer, Xen, KVM, docker, LXC (contêiner)
Armazenamento	NFS, ssh, qcow2, vmfs, ceph, LVM, fslvm, dev	LVM, Ceph, Gluster, NFS, ZFS, Sheepdog

5. Conclusões

A pesquisa realizada teve como objetivo apresentar uma análise da usabilidade em duas ferramentas de gerenciamento de IaaS (OpenNebula e OpenStack), levando-se em conta o ponto de vista do usuário e administrador de IaaS. Os resultados das comparações mostram que existem divergências entre as ferramentas (concepções, funcionalidades, arquitetura), pois cada uma é voltada para superar desafios específicos.

A interface gráfica é o *middleware* entre o usuário de IaaS e os recursos que a ferramenta possui. Dessa forma, foi visto que administradores da OpenNebula, possuem uma maior flexibilidade para instanciar VMs, podendo selecionar qual nodo a mesma será instanciada, tal opção não é encontrada na OpenStack. Ainda, pode-se visualizar a utilização dos recursos do sistema de uma forma simples (primeira tela após o login do usuário). O usuário da OpenNebula também consegue editar *templates* (modelo de configuração da VM), também consegue visualizar todas as redes virtuais e selecionar qual será utilizada, o que é uma vulnerabilidade do ponto de vista da segurança. Na Openstack, os usuários pertencem a projetos, e esse grupo de usuários possuem acesso total aos volumes e redes virtuais, que de fato são mais completos e robustos, comparado aos da OpenNebula.

Nesse ponto, através das comparações foi visto que a OpenNebula possui uma interface gráfica mais amigável ao usuário e administradores de IaaS, pois é uma solução mais centralizada e com gerenciamento simplificado. Por isso, ela é indicada para ambientes de nuvem privada, ou *data centers*. Em demandas específicas que necessitem de recursos avançados de nuvem (VLANs, *firewall*, roteador virtual), o administrador pode recorrer a ferramentas de terceiros compatíveis com a OpenNebula.

Por outro lado, a ferramenta OpenStack é mais complexa, robusta e fragmentada, pois diversos "stacks" (investidores externos), desenvolvem novos projetos para ser utilizados com ele. Em instalações típicas, são oferecidos recursos avançados, como o componente de rede Neutron, que oferece suporte completo em serviços de rede (roteadores virtuais, VLANs, segmentações, etc). As CLIs da OpenStack possuem recursos mais avançados que as da OpenNebula, e com essa gama diversificada de recursos oferecidos, os administradores de IaaS necessitam de um alto conhecimento do sistema operacional e da própria ferramenta de infraestrutura para implantação e gerenciamento. A análise evidencia que para um ambientes de nuvem avançado, escalável e flexível (amplo suporte à virtualizadores), como um provedor de nuvem pública ou um conjunto de *data centers*, OpenStack tende a ser vantajoso.

Como trabalhos futuros, o objetivo é continuar investigando a camada de gerenciamento de IaaS, especificamente sobre o gerenciamento de energia e suporte às SLAs (Acordo de Nível de Serviço) nas ferramentas OpenNebula e OpenStack.

Agradecimentos

Esta pesquisa foi realizada com o apoio do projeto HiPerfCloud⁴. Os autores agradecem o suporte financeiro da Abase Sistemas⁵ e da Sociedade Educacional Três de

⁴ <http://hiperfcloud.setrem.com.br/>

⁵ <http://www.abase.com.br/>

Maio (SETREM)⁶.

Referências

- [Alam et al. 2015] Alam, M. I., Pandey, M., and Rautaray, S. S. (2015). A comprehensive survey on cloud computing.
- [Buyya et al. 2013] Buyya, R., Vecchiola, C., and Selvi, S. (2013). *Mastering Cloud Computing: Foundations and Applications Programming*. Mastering Cloud Computing: Foundations and Applications Programming. Elsevier Science.
- [Chandrasekaran 2014] Chandrasekaran, K. (2014). *Essentials of Cloud Computing*. Taylor & Francis.
- [Dukaric and Juric 2013] Dukaric, R. and Juric, M. B. (2013). Towards a unified taxonomy and architecture of cloud frameworks. *Future Generation Computer Systems*, 29(5):1196–1210.
- [Gupta and Kale 2013] Gupta, A. and Kale, L. V. (2013). Towards efficient mapping, scheduling, and execution of hpc applications on platforms in cloud. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, 2013 IEEE 27th International, pages 2294–2297. IEEE.
- [Gupta et al. 2013] Gupta, A., Kale, L. V., Gioachin, F., March, V., Suen, C., Lee, B., Faraboschi, P., Kaufmann, R., and Milojicic, D. (2013). The who, what, why and how of high performance computing applications in the cloud. In *Proceedings of the 5th IEEE International Conference on Cloud Computing Technology and Science*.
- [Jacobson et al. 2011] Jacobson, D., Woods, D., and Brail, G. (2011). *APIs: A Strategy Guide*. O'Reilly and Associate Series. O'Reilly Media.
- [Llorente 2015] Llorente, I. M. (2015). OpenNebula vs. OpenStack: User Needs vs. Vendor Driven (Official Page) <<http://opennebula.org/opennebula-vs-openstack-user-needs-vs-vendor-driven/>>. Last access in May, 2015.
- [Maron et al. 2014] Maron, C. A. F., Griebler, D., Vogel, A., and Schepke, C. (2014). Avaliação e Comparação do Desempenho das Ferramentas OpenStack e OpenNebula. In *12th Escola Regional de Redes de Computadores (ERRC)*, pages 1–5, Canoas. Sociedade Brasileira de Computação.
- [Maron et al. 2015] Maron, C. A. F., Griebler, D., Vogel, A., and Schepke, C. (2015). Em Direção à Comparação do Desempenho das Aplicações Paralelas nas Ferramentas OpenStack e OpenNebula. In *15th Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS)*, Gramado, RS, Brazil. Sociedade Brasileira de Computação.
- [O'Neil 2014] O'Neil, M. (2014). *The Death of Core Competency: A Management Guide to Cloud Computing and the Zero Friction Future*. InsightaaS Press.
- [OpenNebula 2015] OpenNebula (2015). OpenNebula (Official Page) <<http://opennebula.org/>>. Last access in May, 2015.
- [OpenStack 2015] OpenStack (2015). OpenStack roadmap <<http://openstack.org/>>. Last access May, 2015.

⁶ <http://www.setrem.com.br/>

- [Petersen 2006] Petersen, R. (2006). *Introductory Command Line Unix for Users*. Surfing Turtle Press.
- [Righi et al. 2015] Righi, R., Rodrigues, V., Andre daCosta, C., Galante, G., Bona, L., and Ferreto, T. (2015). Autoelastic: Automatic resource elasticity for high performance applications in the cloud. *Cloud Computing, IEEE Transactions on*, PP(99):1–1.
- [Sabharwal 2013] Sabharwal, N. (2013). *Apache CloudStack Cloud Computing*. Community experience distilled. Packt Publishing.
- [Shroff 2010] Shroff, G. (2010). *Enterprise Cloud Computing: Technology, Architecture, Applications*. Cambridge University Press.
- [Thome et al. 2013] Thome, B., Hentges, E., and Griebler, D. (2013). Computação em Nuvem: Análise Comparativa de Ferramentas Open Source para IaaS. In 11th *Escola Regional de Redes de Computadores (ERRC)*, page 4, Porto Alegre, RS, Brazil. Sociedade Brasileira de Computação.
- [Wen et al. 2012] Wen, X., Gu, G., Li, Q., Gao, Y., and Zhang, X. (2012). Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 2457–2461. IEEE.