

UNIVERSIDADE ESTADUAL DO PIAUÍ - UESPI
CAMPUS PROFESSOR ALEXANDRE ALVES DE OLIVEIRA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

SÁVIO GEORGE DE ALMEIDA ARAÚJO

**ANÁLISE DE DESEMPENHO DE SERVIDORES EM NUVEM COM OPENSTACK E
SOFTWARE TRADICIONAL DE VIRTUALIZAÇÃO**

Biblioteca UESPI PNB
Registro Nº _____
CDD 004
CUTTER A658a
V _____ EX 1
Data 28/01/2018
Visto _____

PARNAÍBA-PI

2015

SÁVIO GEORGE DE ALMEIDA ARAÚJO

**ANÁLISE DE DESEMPENHO DE SERVIDORES EM NUVEM COM OPENSTACK E
SOFTWARE TRADICIONAL DE VIRTUALIZAÇÃO**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para obtenção do título
de Bacharel em Ciências da Computação,
Universidade Estadual do Piauí, Área da
Computação.

Orientador: Prof. Dr. Sérgio Barros de Sousa.

PARNAÍBA-PI

2015

A658a

Araújo, Sávio George de Almeida.

Análise de desempenho de servidores em nuvem com openstack e software tradicional de virtualização / Sávio George de Almeida Araújo. Parnaíba: UESPI, 2015.

43 f. : il.

Orientador: Dr. Sérgio Barros de Sousa.

Monografia, (Graduação em Ciência da Computação) - Universidade Estadual do Piauí, 2015.

1. Computação em nuvem - 2. Virtualização 3. Desempenho.

I. Sousa, Sérgio Barros de II. Universidade Estadual do Piauí III. Título

CDD 004

SÁVIO GEORGE DE ALMEIDA ARAÚJO

**ANÁLISE E DESEMPENHO DE SERVIDORES EM NUVEM COM OPENSTACK E
SOFTWARE TRADICIONAL DE VIRTUALIZAÇÃO**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual do Piauí – UESPI, Campus Prof. Alexandre Alves de Oliveira, como parte das exigências da disciplina de Estágio Supervisionado, requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Dr. Sérgio Barros de Sousa

Monografia Aprovada em: **24 de julho de 2015.**

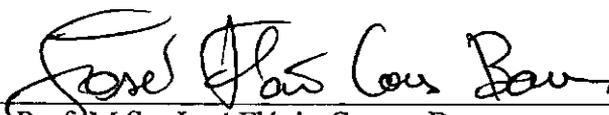
Banca Examinadora:



Prof. Dr. Sérgio Barros de Sousa
UESPI/Parnaíba – Orientador



Prof. M.Sc. Francisco das Chagas Rocha
UESPI/Parnaíba – Avaliador



Prof. M.Sc. José Flávio Gomes Barros
FACEMA/Caxias-MA – Avaliador

DEDICATÓRIA

Este trabalho é dedicado primeiramente a Deus, o autor e consumidor da minha fé que me inspirou, em entendimento e deu-me forças para suportar as noites sem dormir (risos).
À minha querida, linda, maravilhosa e incomparável esposa pela compreensão e apoio nesta reta final. Aos meus familiares e amigos pela torcida, apoio e pelas orações.

AGRADECIMENTOS

Agradeço primeiramente a Deus, minha fonte de expiração, pela conclusão deste trabalho.

Ao Prof. Dr. Sérgio Barros pelas orientações e inspiração para continuação na carreira acadêmica.

Aos amigos Gylvanne Viera e Isaque Alves que muito apoiaram e ajudaram na descoberta de meus limites e a superá-los.

A todos muito obrigado.

EPÍGRAFE

*Pois a sabedoria entrará no teu coração, e o conhecimento
será aprazível à tua alma.
(Salomão)*

RESUMO

Ao longo de algumas experiências obtidas na área de desenvolvimento de software, foi possível observar a rotina e as etapas necessárias para se criar um ambiente estável de programação. Percebeu-se que muito tempo era empregado na preparação e configuração do ambiente, com instalação de bibliotecas, pacotes não nativos do sistema, padronização de uma linguagem, melhoramento no desempenho de virtualizações e resolução de incompatibilidades. Com isso, buscou-se ferramentas que auxiliassem na criação de ambientes de desenvolvimento e que pudesse comprovar sua viabilidade. Nesse trabalho é realizado um estudo comparativo entre ambientes virtualizados através de testes de desempenho. Consiste na implementação de um ambiente de nuvem e suas configurações básicas e na realização de tarefas concernentes a um ambiente em produção. O desempenho durante as atividades foi comparado com o número de máquinas virtuais utilizadas em cada caso e suas respectivas variáveis de desempenho. Verificou-se uma considerável economia de recursos da máquina física na implementação realizada com o software Vagrant, concretizando, através dos resultados, as vantagens de utilizá-lo.

Palavras-chave: Computação em Nuvem. Desenvolvimento. Virtualização. Recursos. Desempenho.

ABSTRACT

Along some experiences gained in the software development area, we observed the routine and the steps required to create a stable programming environment. It was noticed that a long time was employed in the preparation and environment configuration, with installation libraries, non-native packages system, standardization of a language, improving the performance of virtualization and resolving incompatibilities. Thus, it sought to tools that could help in the creation of development environments and that could prove their viability. This work is carried out a comparative study of virtualized environments through performance tests. Is to implement a cloud environment and its basic settings and carrying out concerning an environment where production tasks. The performance in activities was compared with the number of virtual machines used in each case and their performance variables. There was a considerable saving features of the physical machine in implementing held with Vagrant software, implementing, through the results, the advantages of using it.

Keywords: Cloud Computing. Development. Virtualization. Features. Performance.

LISTA DE FIGURAS

Figura 1 - Internet como metáfora de computação em nuvem.....	16
Figura 2 - Exemplo de virtualização.....	18
Figura 3 - Representação de virtualização de servidores.....	18
Figura 4 - Modelo de rede virtualizada.....	19
Figura 5 - Funcionamento Desktop Virtual.....	20
Figura 6 - Demonstração de aplicativos virtualizados.....	21
Figura 7 - Modelo de Serviços da Nuvem.....	21
Figura 8 - Instalação do virtualbox e obtenção do instalador vagrant (a). Instalação do vagrant via terminal (b).....	26
Figura 9 - Adição da box e criação do Vagrantfile criação da configuração (a). Inicialização do vagrant (b).....	27
Figura 10 - Acesso ao nó chef (a) e baixando instalador (b).....	27
Figura 11 - Obtendo cookbooks e habilitando knife e fazendo upload (a). Listando o conteúdo do livro de receitas (b).....	28
Figura 12 - Processo de geração da keygen (a) e cópia da ID da keygen para o nó controlador via SSH (b).....	29
Figura 13 - Registrando o chef-client em cada nó e adicionando as regras do OpenStack (a); instalação do OpenStack em cada nó através do chef (b).....	29
Figura 14 - Obtendo credenciais das API's e criação da imagem cirros (a). Acesso ao painel horizon através do navegador (b).....	30
Figura 15 - Criação das instâncias e atribuição dos IP's flutuantes.....	31
Figura 16 - Visão geral das instâncias criadas (a). Resumo de todos os projetos criados (b)..	31
Figura 17 - Estado 'zero' da máquina.....	34
Figura 18 - Desempenho com virtualbox em funcionamento.....	34
Figura 19 - Desempenho virtualbox com 1vm em funcionamento (a). Captura virtualbox 1vm (b).....	35
Figura 20 - Desempenho virtualbox com 2vm em funcionamento (a). Captura virtualbox 2vm (b).....	36
Figura 21 - Desempenho virtualbox com 3vm em funcionamento (a). Captura virtualbox com 3vm (b).....	37
Figura 22 - Desempenho do Vagrant com Virtualbox e usando interface web (a). Captura	

Virtualbox com 5 vm criadas pelo Vagrant (b).....38

Figura 23 - Desempenho com duas instâncias sendo executadas..... 39

Figura 24 - Variáveis de desempenho..... 40

LISTA DE TABELAS

Tabela 1 - Eventos relacionados ao ambiente.....	32
Tabela 2 - Descrição do hardware utilizado.....	33

LISTA DE ABREVIATURAS E SIGLAS

Admin	Administrador (tradução)
AMD	Advanced Micro Devices
API	Application Programming Interface
APU	Accelerated Processing Unit
ARPANet	Advanced Research Projects Agency Network
ATI	Tecnologia de Placas de Vídeo
CD	Compact Disc
CD-ROM	Compact Disc Read-Only
DDR	Double Data Rate
DVD	Digital Versatile Disc
E/S	Entrada/Saída
HD	Hard Disc
IaaS	Infrastructure as a Service
ID	Identity Digital
IP	Internet Protocol
I/O	Input / Output
LTS	Long Term Support
PaaS	Platform as a service
PC	Personal Computer
Power-off	Desligar
RAM	Random Access Memory
SaaS	Software as a service
SSH	Secure Shell
Swap	Management virtual memory
TI	Tecnologia da Informação
USB	Universal Serial Bus
VM	Virtual Machine
VMM	Virtual Machine Manager
Vmware	Software de Virtualização
Web	World Wide Web

SUMÁRIO

1 INTRODUÇÃO.....	13
2 REVISÃO DA LITERATURA.....	15
2.1 Início da computação em nuvem.....	15
2.2 A nuvem e os benefícios da virtualização.....	16
2.3 O funcionamento da virtualização.....	17
2.3.1 Tipos de virtualização.....	18
2.3.1.1 Virtualização de servidores.....	18
2.3.1.2 Virtualização total (completa) e paravirtualização.....	18
2.3.1.3 Virtualização de redes.....	19
2.3.1.4 Virtualização de Desktops.....	20
2.3.1.5 Virtualização de Aplicativos.....	20
2.4 Tipos de Computação em Nuvem.....	21
2.4.1 Modelos de Implantações em Nuvem.....	22
2.5 Backup e Segurança.....	23
2.5.1 Tipos de Backup.....	24
2.6. Desenvolvimento com Vagrant.....	25
3 MATERIAIS E MÉTODOS.....	26
3.1 Demonstrando o ambiente.....	26
3.2 Preparação do ambiente de teste de desempenho.....	30
4 RESULTADOS E DISCUSSÃO.....	34
4.1. Testes de Desempenho.....	34
4.1.1 Análise de Desempenho com VirtualBox.....	34
4.1.2 Análise de Desempenho com Vagrant.....	37
4.2 Análise comparativa dos ambientes.....	39
4.3 Definindo variáveis de desempenho.....	39
5 CONCLUSÃO.....	42
5.1 Sugestão para trabalhos futuros.....	42
REFERÊNCIAS.....	43

1 INTRODUÇÃO

Ter um ambiente de desenvolvimento bem consistente e ágil, trás inúmeras vantagens perante o mercado de software. O uso de virtualização para a criação desses ambientes tem sido cada vez mais comum, entretanto manter e criá-los pode tornar-se uma tarefa muito demorada e por vezes cara. Uma das tarefas mais difíceis é acrescentar novos recursos e manter as configurações compatíveis, incluindo aí a necessidade de trabalhar com várias linguagens, várias versões de uma mesma biblioteca, um novo membro na equipe, e um considerável uso do tempo para compatibilizar a infraestrutura de rede.

Diante desse aspecto, este trabalho intenciona demonstrar as funcionalidades e vantagens de utilizar softwares que permitam criar ambientes de desenvolvimento de infraestruturas totalmente virtualizados usando várias máquinas virtuais distintas baseadas em virtualização de servidores. E apresentar a viabilidade de uso do software Vagrant unido ao servidor Chef junto com o ambiente de virtualização em detrimento ao uso de técnicas tradicionais de virtualização utilizando exclusivamente o software Virtualbox. Para demonstração dos recursos e análise de eficiência do ambiente demonstrado, foi utilizado software de virtualização com quantidade de processadores e desempenho de memória pré-configurada; E em seguida o mesmo software sendo gerenciado por outro, para o desenvolvimento da infra-estrutura.

Nesse aspecto, o presente trabalho tem como objetivo apresentar uma alternativa de ambiente integrado de virtualização de servidores na nuvem, avaliando e otimizando o uso de recursos financeiros e de rede frente ao modelo tradicional de organização de máquinas virtuais.

Foi utilizado o software Vagrant para automatizar a criação do ambiente de desenvolvimento e uma ferramenta de automação e gerenciamento de configurações o Chef que usa dados que podem ser aplicados a vários projetos definidos em nós. Através de testes realizados num ambiente virtualizado foi implementado uma infra-estrutura de nuvem com Openstack e feito testes de desempenho para avaliar o consumo de recursos da máquina física, usando o Vagrant com Virtualbox e somente Virtualbox entre as máquinas virtuais.

Para apresentação desta pesquisa, este trabalho descreverá no capítulo 2 uma revisão bibliográfica sobre virtualização, Computação em Nuvem, backup e segurança e

desenvolvimento com Vagrant. No capítulo 3 serão demonstrados os métodos de preparação do ambiente virtual de produção e ambiente de teste para analisar as características e o eficiente funcionamento do ambiente criado.

Em sequência no capítulo 4 será apresentado uma análise do desempenho da máquina física a partir do funcionamento e adição das máquinas virtuais em seu funcionamento padrão e gerenciadas pelo Vagrant e por fim são demonstrados os dados obtidos e analisado os resultados descritos na conclusão contida no capítulo 5 deste trabalho.

2 REVISÃO DA LITERATURA

2.1 Início da computação em nuvem

Segundo Gomes (2012), no fim dos anos 60, a agência *Advanced Research Projects Agency Network* (ARPANet) concebeu a criação de uma rede intergaláctica de computadores. A idéia consistia em interagir e acender a qualquer recurso na Internet, a partir de qualquer local e qualquer momento.

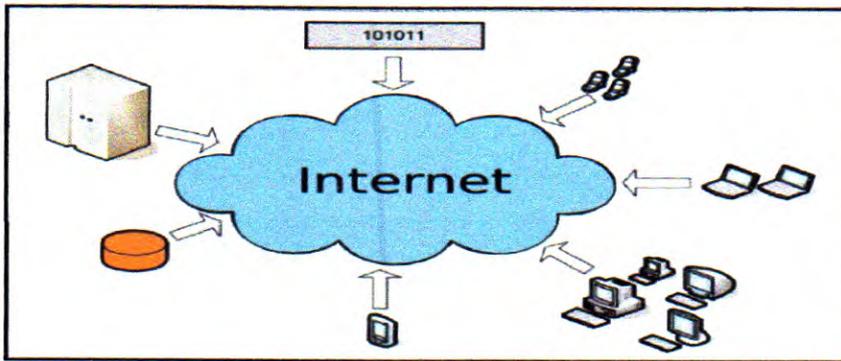
A Computação em Nuvem é um termo usado para descrever um ambiente de computação baseado em uma imensa rede de servidores, sejam estes virtuais ou físicos. Uma definição simples pode então ser um conjunto de recursos como capacidade de processamento, armazenamento, conectividade, plataformas, aplicações e serviços disponibilizados na internet (TAURION, 2009).

Segundo Parreira (2012), a primeira vez que surgiu o termo Computação em Nuvem foi em 1997, durante uma palestra em Dallas de Ramnath Chellappa, um professor de sistemas de informação, chamada os Intermediários em Computação em Nuvem: Um Novo Paradigma Computacional.

No final da década de 90, início dos anos 2000, a Amazon se notabilizou como empresa de comércio eletrônico oferecendo uma variada gama de mercadorias como livros, CDs e DVDs, artigos de beleza, eletrodomésticos etc. Para realizar esse serviço de forma satisfatória, a Amazon investiu pesadamente em um parque computacional e, posteriormente, em decorrência de períodos de ociosidade, percebeu que poderia aproveitar essa infraestrutura para oferecer serviços de computação para empresas e usuários domésticos. Nascia assim, a computação em nuvem na Amazon (CARISSIMI, 2008).

Segundo Buyya e colaboradores (2009), a computação em Nuvem está se tornando uma das palavras chaves da indústria de Tecnologia da Informação (TI). A nuvem é uma metáfora para a Internet ou infraestrutura de comunicação entre os componentes arquiteturais, baseada em uma abstração que oculta à complexidade da infraestrutura. Cada parte desta infraestrutura é provida como um serviço os quais normalmente são alocados em centros de dados, utilizando hardware compartilhado para computação e armazenamento. A Figura 1 demonstra a internet como exemplo de uma grande nuvem.

Figura 1 - Internet como metáfora de computação em nuvem



Fonte: RUSCHEL; ZANOTTO; MOTA, 2010.

Muitos programas e aplicações, assim como arquivos e outros dados relacionados, não precisam mais estar instalados ou armazenados no computador do usuário. Dois exemplos de serviços disponibilizados são o Google Drive do provedor Google, um serviço de armazenamento que integra o Google Docs, um pacote de aplicativos que disponibiliza programas de edição de texto, planilhas e apresentações. E o IBM SmartCloud Enterprise do provedor IBM, serviço de infraestrutura projetado para fornecer acesso rápido a ambientes seguros de servidores virtuais corporativos (PEREIRA, 2013).

2.2 A nuvem e os benefícios da virtualização

Segundo Siqueira (2008), a virtualização tem um potencial extremamente grande, conta com inúmeras vantagens que chega a constituir um novo campo da informática, permitindo a simulação de aplicativos, ferramentas e demais recursos. Facilita a transformação de ambientes físicos complexos em ambientes simplificados e fáceis de gerenciar. No mundo da computação tudo se faz de forma virtual, com ferramentas de acompanhamento da entrega de aplicações armazenadas no próprio computador, na rede local ou num servidor remoto. Os softwares de entrega de aplicações podem ser empregados em pelo menos três ambientes: na internet, nas estruturas cliente-servidor e em desktops.

Segundo Carissimi (2008), virtualização é a técnica que permite particionar um único sistema computacional em vários outros denominados de máquinas virtuais. Cada máquina virtual oferece um ambiente completo muito similar a uma máquina física. Com isso, cada máquina virtual pode ter seu próprio sistema operacional, aplicativos e serviços de rede, ou

seja, na Internet.

A virtualização é uma tecnologia que permite uma gestão eficiente dos recursos, otimizando a infraestrutura por meio da consolidação de equipamentos e gerenciamento centralizado, gerando economia (BOSING; KAUFMANN, 2012)

Segundo Veras (2011), as organizações buscam construir plataformas digitais mais estáveis que permitam o crescimento mesmo em ambientes turbulentos. Parece contraditório mais não é, espera-se plataformas mais estáveis e flexíveis. Pode-se pensar em arquiteturas e infraestrutura que suportem o crescimento do negócio. A virtualização, neste contexto, é uma peça chave, pois permite alterar a infraestrutura rapidamente utilizando instrumentos lógicos e não físicos, ao mesmo tempo que estabiliza o ambiente tornando as aplicações independentes do hardware.

2.3 O funcionamento da virtualização

Segundo Dantas (2008), o significado original de máquina virtual, também por vezes chamada de máquina virtual de hardware, é o conjunto de ambientes de execução díspares e independentes, correndo cada um deles um determinado Sistema Operacional. Isto permite que aplicações escritas para um Sistema Operacional sejam utilizáveis em máquinas que correm Sistemas Operacionais diferentes.

Segundo More e Tapaswi (2014), a técnica de virtualização é usada para criar um ambiente virtual para computação pela virtualização de hardware, I/O e processadores. Este ambiente virtual é possível com a ajuda de uma camada especial de software chamado um monitor de VM (VMM) ou hypervisor.

Para entender seu funcionamento, precisamos ter em mente alguns conceitos básicos sobre virtualização. Dantas (2008) afirma que o host é também apelidado de servidor de máquinas virtuais, refere-se ao sistema operacional principal que gere (todo) o hardware e controla as máquinas virtuais. Teixeira (2010) fala que o guest é executado sobre o sistema computacional virtualizado, sendo que vários sistemas virtualizados podem coexistir, estando em execução simultânea no mesmo sistema real. Segundo Jacobi e colaboradores (2012), o hypervisor é responsável pelo gerenciamento e controle dos recursos compartilhados pela máquina física (Figura 2).

Figura 2 - Exemplo de virtualização



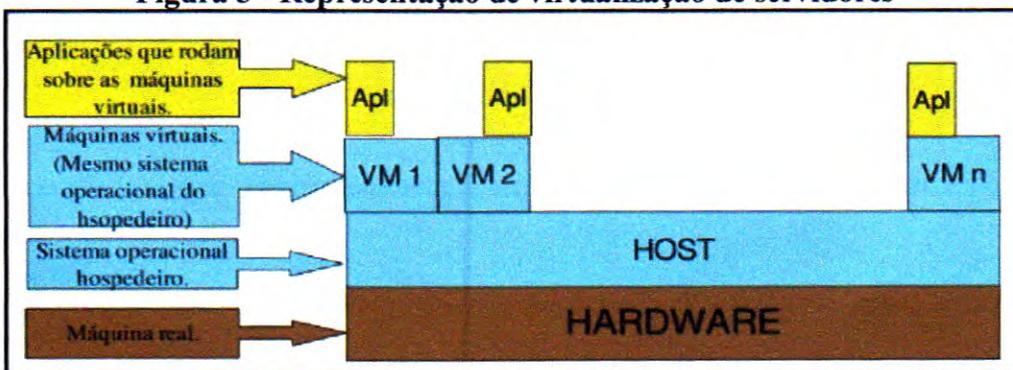
Fonte: DANTAS, 2008.

2.3.1 Tipos de virtualização

2.3.1.1 Virtualização de servidores

De acordo com Carissimi (2008), uma pesquisa realizada pela Network World (www.networkworld.com), a virtualização de servidores é motivada, em ordem de importância, pelas seguintes razões: aumentar a taxa de utilização de servidores, reduzir os custos operacionais de data centers, melhorar os procedimentos de recuperação de desastres e de backup, criar ambientes mais flexíveis para desenvolvimento e teste de software e reduzir custos de administração de TI (Figura 3).

Figura 3 - Representação de virtualização de servidores



Fonte: POLLON, 2008 apud LAUREANO, 2004.

2.3.1.2 Virtualização total (completa) e paravirtualização

Segundo Soares (2010), em ambientes de virtualização completa as máquinas

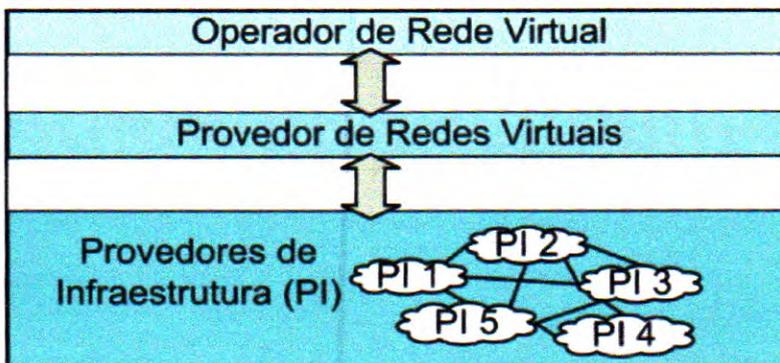
virtuais executam sistemas operacionais sem quaisquer modificações. Com isso, capturar todas as instruções privilegiadas executadas pelas máquinas virtuais requer a aplicação de métodos complexos de análise dinâmica de código, como a tecnologia usada pelo VMware para reescrever partes do código que devem ser repassadas para o VMM em tempo de execução. Isto impõe um custo adicional na execução das máquinas virtuais, que embora possa ser atenuado usando artifícios como caching para as porções de código mais utilizadas, ainda é significativa.

Segundo Choinacki (2012), na paravirtualização o sistema a ser virtualizado sofre modificações para que a interação com o monitor das máquinas virtuais seja mais eficiente. Por sofrer modificações o sistema virtualizado através de paravirtualização se torna pouco portátil, ela permite o acesso aos recursos de hardware diretamente. O acesso é monitorado pelo monitor de máquinas virtuais, que fornece ao sistema convidado todos os limites do sistema, tais como endereços de memória que podem ser utilizados e endereçamento de disco.

2.3.1.3 Virtualização de redes

A virtualização de redes é uma técnica promissora para a Internet do futuro, pois facilita a adição de novas funcionalidades nos elementos de rede e diminui os custos das organizações. Uma das questões que ainda precisa ser investigada para a implantação eficiente desta tecnologia é a alocação de recursos físicos para as redes virtuais (Figura 4) (ALKMIM; BATISTA; FONSECA, 2011).

Figura 4 - Modelo de rede virtualizada



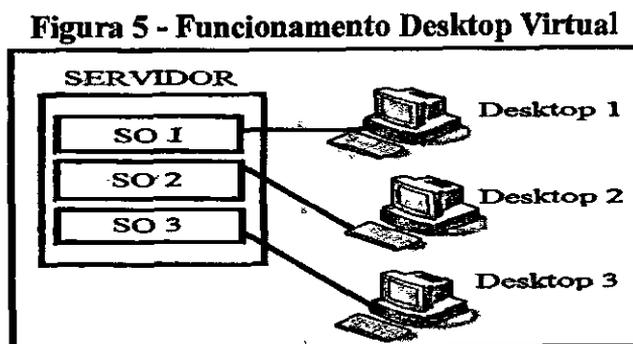
Fonte: FERNANDES; DUARTE, 2011.

Segundo Fernandes e Duarte (2011), o controle e o gerenciamento das redes virtuais são discutidos para as principais plataformas de virtualização existentes e incluem o controle global e local da rede. O controle global inclui operações como a instanciação de nós e enlaces virtuais. Já o controle local monitora os recursos de cada nó físico que são atribuídos a cada rede virtual, tratando o isolamento entre redes.

2.3.1.4 Virtualização de Desktops

A consolidação de servidores é o uso clássico de virtualização, mas ela também traz uma série de benefícios quando empregada em desktops. Inicialmente, a virtualização oferece uma forma simples para testar novas configurações ou executar programas que foram feitos para sistemas operacionais diferentes do nativo. Dessa forma, um usuário que deseja testar um software, ou abrir um programa, que não foi desenvolvido para seu sistema operacional pode lançar mão desse recurso (CARISSIMI, 2008).

As técnicas de virtualização têm grande importância para projetos de sistemas computacionais, utilizadas com o objetivo principal de desacoplar os recursos físicos de um hardware da sua representação lógica (Figura 5). (BARHAM; DRAGOVIC, 2003 apud AGUIAR, 2008).



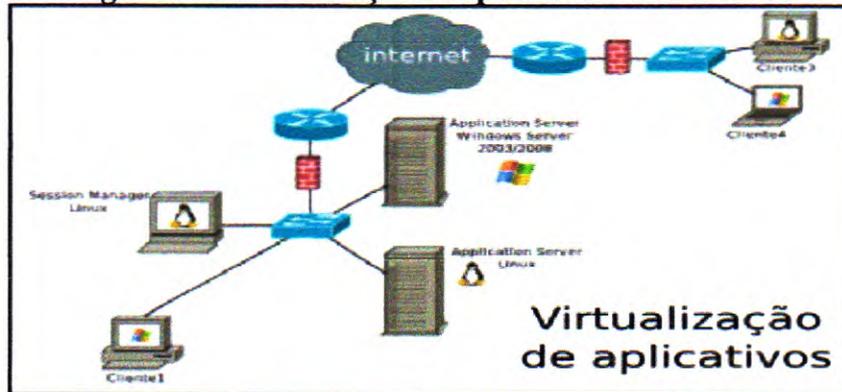
Fonte: BOSING; KAUFMANN, 2012.

2.3.1.5 Virtualização de Aplicativos

Segundo Teixeira (2010), a aplicação virtualizada é executada sobre a máquina local usando os seus recursos, mas sua execução não irá modificar o sistema hospedeiro. Ao invés disso, ele é executado em um pequeno ambiente virtual que contém arquivos necessários,

entradas de registros e outros recursos que ele necessita para ser executado. Este ambiente virtual atua como uma camada entre a aplicação e o sistema operacional. A camada virtual deve ser leve e geralmente ocupar pouco espaço em memória e ser carregado apenas no momento exato em que a aplicação a ser virtualizada será executada (Figura 6).

Figura 6 - Demonstração de aplicativos virtualizados

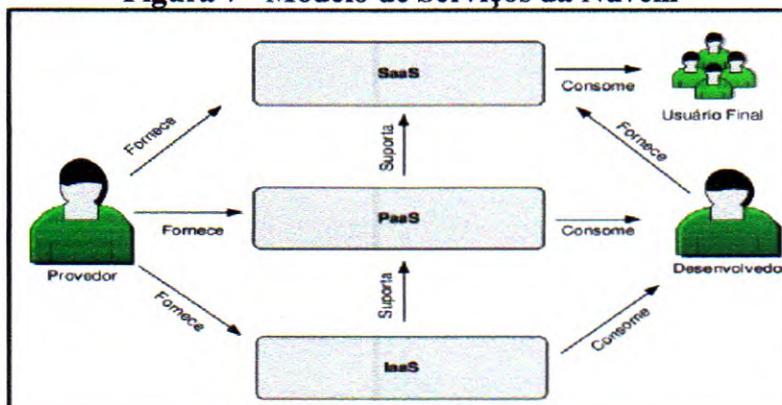


Fonte: CORDEIRO, 2011.

2.4 Tipos de Computação em Nuvem

Segundo Ogura (2011), na computação a dependência do equipamento pode ser solucionado pela tecnologia de virtualização possibilitando que ele fosse executado por meio de uma máquina virtual (MV). Com essa progressão da tecnologia, as infraestruturas foram divididas em três pilares conceituais encontrados em forma de infraestrutura (IaaS), plataforma (PaaS) e aplicação (SaaS) (Figura 7).

Figura 7 - Modelo de Serviços da Nuvem



Fonte: RUSCHEL; ZANOTTO; MOTA, 2010.

Segundo Pedrosa e Nogueira (2011), o modelo IaaS servidores capazes de executar softwares customizados e operar em diferentes sistemas operacionais. Possui uma aplicação que funciona como uma interface única para a administração da infraestrutura, promovendo a comunicação com hosts, switches, roteadores e o suporte para a inclusão de novos equipamentos. Por se tratar da camada inferior, esta também é responsável por prover a infraestrutura necessária para as camadas intermediária e superior.

Foscarini e Ribeiro (2014) diz que no modelo PaaS é concedido ao usuário acesso a uma plataforma previamente configurada pelo provedor de Computação em Nuvem. Questões relativas ao gerenciamento do sistema operacional sobre o qual a plataforma está configurada ficam a cargo do provedor de Computação em Nuvem.

Ogura (2011) diz que o modelo SaaS é a camada mais alta da arquitetura da computação na nuvem tem a responsabilidade de disponibilizar aplicações completas ao usuário final. Este acesso é provido pelos prestadores de serviço através de portais web, sendo completamente transparente ao usuário, o que permite a execução de programas que executam na nuvem a partir de uma máquina local. Para oferecer esta transparência, o SaaS utiliza-se das duas camadas inferiores, o PaaS e o IaaS.

2.4.1 Modelos de Implantações em Nuvem

Segundo Sousa e colaboradores (2010), quanto ao acesso e à disponibilidade, há diferentes tipos de modelos de implantação para os ambientes de computação em nuvem. A restrição ou abertura de acesso depende do processo de negócios, do tipo de informação e do nível de visão desejado.

Segundo Techio e Misaghi (2014 apud NIST, 2011), no modelo de nuvem pública a infraestrutura de nuvem é disponibilizada para o público em geral, desde usuários domésticos à grandes corporações, o acesso a informação é fácil, basta saber o endereço lógico o fornecedor tem o controle dos recursos aos quais são de fácil acesso e disponibilizados para o público em geral, dando a impressão de ser infinita devida a grande elasticidade. O controle da nuvem pública é exercido pelo provedor que disponibiliza o serviço, sendo este o responsável pela instalação, manutenção, gerenciamento e a atualização dos recursos necessários. O usuário pode fazer personalizações básicas e limitadas. Alguns autores não recomendam que sejam armazenadas informações sigilosas neste tipo de nuvem, por questões

de segurança.

Segundo Sousa, Moreira e Machado (2009), no modelo de nuvem privada a infraestrutura de nuvem é utilizada exclusivamente para uma organização, sendo esta nuvem local ou remota e administrada pela própria empresa ou por terceiros. Neste modelo de implantação são empregados políticas de acesso aos serviços. As técnicas utilizadas para prover tais características podem ser em nível de gerenciamento de redes, configurações dos provedores de serviços e a utilização de tecnologias de autenticação e autorização. Um exemplo deste modelo seria o cenário de uma universidade e seus departamentos. A universidade pode estar interessada em disponibilizar serviços para seus departamentos e outros órgãos desta instituição não devem ter acesso a esses serviços.

Simões e Kamienski (2014) afirmam que as nuvens comunitárias são aquelas que permitem uso compartilhado por usuários que colaboram para obter um ambiente computacional de maior capacidade.

Martins (2010) fala sobre no modelo de nuvem híbrida a infraestrutura é uma composição de duas ou mais nuvens, que podem ser do tipo privada, pública ou comunidade e que continuam a ser entidades únicas, mas conectadas por meio de tecnologia proprietária ou padronizada que permite a portabilidade de dados e aplicações.

2.5 Backup e Segurança

Uma política de segurança de uma empresa define as normas e procedimentos que melhor atendam ao propósito da mesma, minimizados os riscos com perdas e violações de qualquer um de seus bens. Em toda política de segurança da informação faz-se necessário ter uma ideia clara daquilo que se quer proteger, contra quem queremos defender e quais obstáculos que essa política oferece para funcionamento normal do sistema. Podemos assumir que todos os dados referentes a uma empresa fazem parte do seu patrimônio. Esta Política de Segurança da Informação restringe-se à defesa das informações e sistemas computacionais de software e hardware da empresa. Assim como ao acesso físico e lógico às informações (MELO JÚNIOR, 2007).

De acordo com Casquinha (2013), backup é a cópia de dados de um dispositivo de armazenamento a outro (CD-ROM, DVD, disco rígido, disco rígido externo (compatíveis com USB), fitas magnéticas e a cópia de segurança externa (online) para que possam ser

restaurados em caso da perda dos dados originais, o que pode envolver desaparecimentos acidentais ou corrupção de dados.

Segundo a Blatt [2011], cerca de 13.000 servidores e 50.000 posições de títulos foram perdidos nos edifícios destruídos. Durante os atentados as torres gêmeas; empresas que estavam ao redor do WTC (World Trade Center) também foram afetadas, pois faziam seus backup's nas torres, muitas delas deixaram até mesmo de existir.

Segundo Melo Júnior (2007), a política de backup é uma importante preocupação da estratégia de segurança física. Mas os backups apresentam vários perigos, pois se um visitante tiver acesso ao backup poderá por em risco a informação.

2.5.1 Tipos de Backup

Segundo Cardoso Neto e colaboradores (2012), um podemos realizar um backup completo que consiste na cópia de todos os arquivos para o dispositivo de armazenamento de destino do backup. Sua característica principal é que não há verificação se o arquivo foi alterado desde a última cópia ou não, independente de qualquer filtro de verificação a cópia completa dos dados é feita, mesmo que seja igual a última. Apesar de oferecer nível de segurança alto e ser um ponto rápido de restauração, esse tipo de backup possui grande desvantagem ao seu favor, pois se torna muito grande ocupando espaços desnecessários na mídia de backup com arquivos duplicados, mas, no entanto, ele é ponto de partida para restauração usando outros métodos que validam somente as alterações após o último backup completo devendo ser realizado periodicamente.

De acordo com Domingues (2012), quando se copia somente os arquivos criados ou alterados desde o último backup normal ou incremental e os marca como arquivos que passaram por backup, então o atributo de arquivo é desmarcado. Se você utilizar uma combinação desses backup's, precisará do último conjunto de backup normal e de todos os conjuntos de backups incrementais para restaurar os dados.

Tomé e colaboradores (2012) afirmam quando se copia todos os dados que foram alterados desde o último backup total encontramos um backup diferencial. Nesta estratégia, o atributo de arquivamento não é desmarcado, desta forma todos os arquivos que foram modificados após um backup completo serão copiados para o backup. Para executar uma restauração completa, é preciso de uma mídia de backup total e da mídia do backup

diferencial mais recente.

2.6. Desenvolvimento com Vagrant

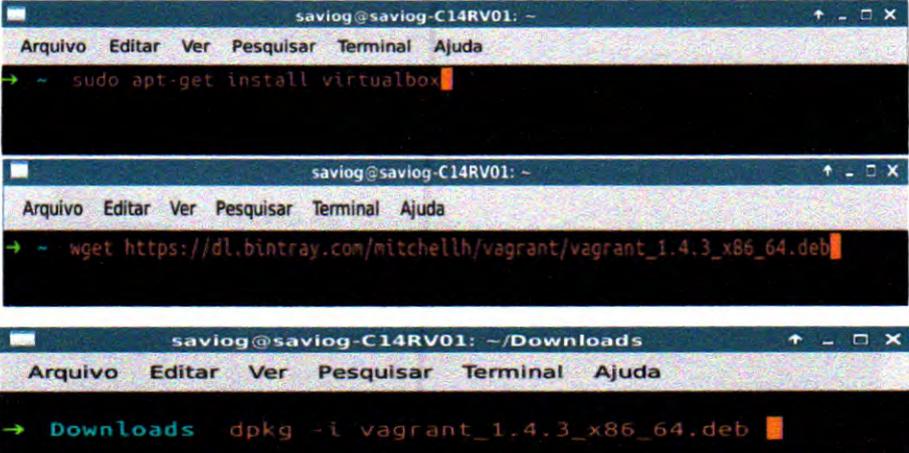
Segundo Peacock (2013), o Vagrant permite gerenciar e apoiar um ambiente de desenvolvimento virtualizado. Em vez de executar todos seus projetos localmente em sua própria máquina, o Vagrant permite que todos os projetos sejam executados individualmente em seu próprio ambiente virtualizado, utilizando diferentes requisitos e dependências para cada projeto. Vagrant fornece uma interface de linha de comando e uma linguagem de configuração simples que permite definir e controlar máquinas virtuais, sendo executadas facilmente em seus próprios sistemas, de forma integrada, o que lhe permite definir a forma como o sua máquina vai interagir com a máquina virtual.

3 MATERIAIS E MÉTODOS

3.1 Demonstrando o ambiente

Para a implementação do ambiente, foi escolhido o Openstack, um software de computação em nuvem no modelo de infraestrutura (IaaS) usando o software Virtualbox, uma ferramenta de emulação de sistemas operacionais, que permite a virtualização de componentes físicos de um computador gerenciado pelo Vagrant um software que cria e gerencia ambientes de desenvolvimento dinamicamente virtualizados e configuráveis. Outra ferramenta com papel importante nesta montagem foi o Chef que permite transformar toda infraestrutura, automatizando sua construção e realizando um provisionamento baseado em nós. A instalação deu-se por meio de um terminal (Figura 8).

Figura 8 - Instalação do virtualbox e obtenção do instalador vagrant (a). Instalação do vagrant via terminal (b)

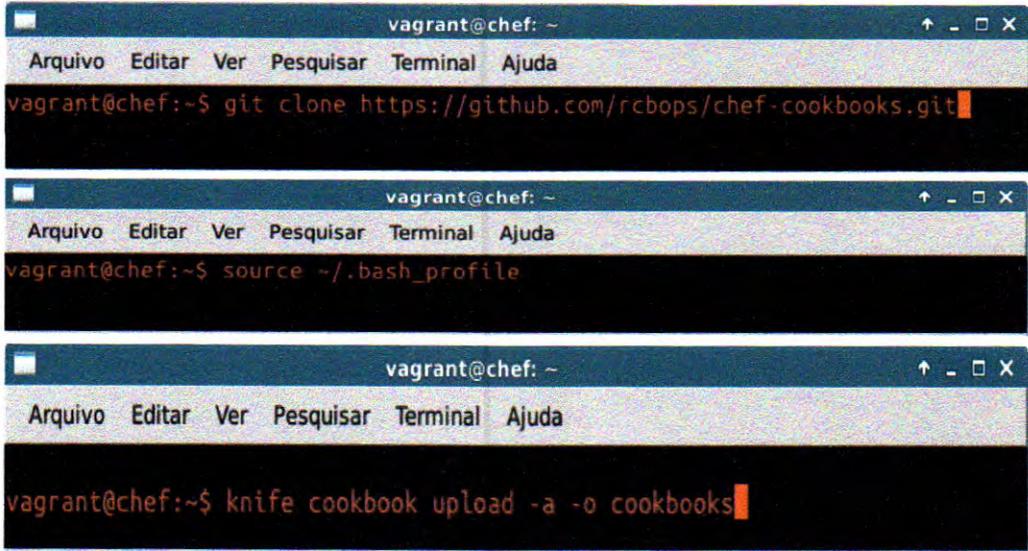


The figure consists of three terminal window screenshots. The first screenshot (a) shows the command `sudo apt-get install virtualbox` being entered in a terminal window titled `saviog@saviog-C14RV01: ~`. The second screenshot shows the command `wget https://dl.bintray.com/mitchellh/vagrant/vagrant_1.4.3_x86_64.deb` being entered in a terminal window titled `saviog@saviog-C14RV01: ~`. The third screenshot (b) shows the command `dpkg -i vagrant_1.4.3_x86_64.deb` being entered in a terminal window titled `saviog@saviog-C14RV01: ~/Downloads`.

Após a instalação foi importado uma *Box* para o Vagrant a fim de criar uma instância de máquina virtual. As *Boxes* são pacotes que possuem imagens pré-configuradas de um sistema operacional, servem como base para a criação de uma VM, simplificando sua configuração. Para concepção deste cenário foi escolhida uma imagem do Ubuntu 12.04 precise 64bits, com intento de ser executada em cada nó criado pelo *Chef*. Posteriormente foi gerado o *Vagrantfile* por meio de um comando chamado 'vagrant init', neste arquivo foi

configuração a partir dos *cookbooks* e atualizá-los (Figura 11).

**Figura 11 - Obtendo cookbooks e habilitando knife e fazendo upload (a).
Listando o conteúdo do livro de receitas (b)**

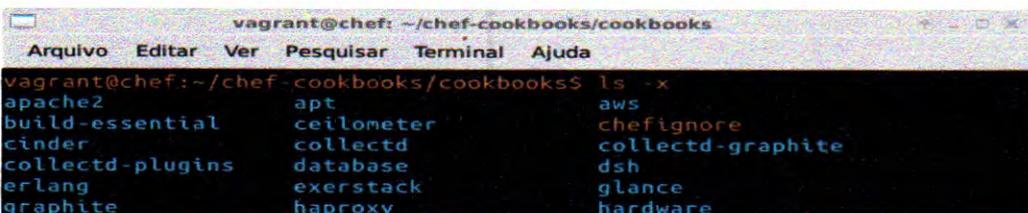


(a)

```
vagrant@chef: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
vagrant@chef:~$ git clone https://github.com/rcbops/chef-cookbooks.git
```

```
vagrant@chef: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
vagrant@chef:~$ source ~/.bash_profile
```

```
vagrant@chef: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
vagrant@chef:~$ knife cookbook upload -a -o cookbooks
```

(b)

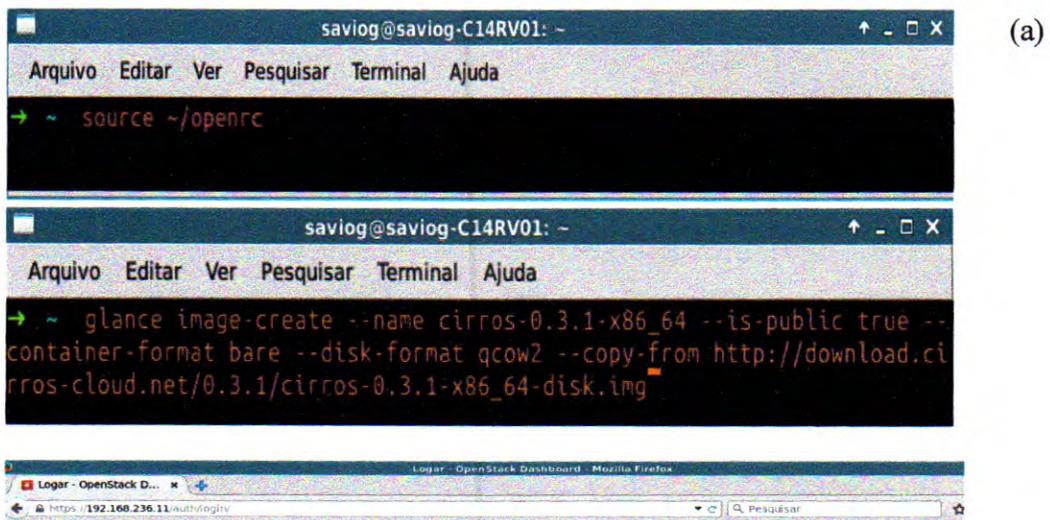
```
vagrant@chef: ~/chef-cookbooks/cookbooks
Arquivo Editar Ver Pesquisar Terminal Ajuda
vagrant@chef:~/chef-cookbooks/cookbooks$ ls -x
apache2          apt              aws
build-essential ceilometer      chefignore
cinder           collectd        collectd-graphite
collectd-plugins database        dsh
erlang           exerstack      glance
graphite        haproxy        hardware
```

Para criar o ambiente de nuvem foi definido uma implementação básica de 5 nós, onde um nó sobreveio para o *Chef-Servidor*, um para o *nó controlador* que serve para controlar outros nós e realizar as principais tarefas nas VM's, dois nós de *nova computação* usado para permitir o gerenciamento escalável de grandes quantidades de servidores e máquinas virtuais, e um *nó cinder* utilizado para armazenamento baseado em blocos e gerenciamento centralizado de serviços.

Para configurar o ambiente do OpenStack, primeiramente foi criado uma chave SSH privada sem senha pública gerenciada pelo Vagrant e copiada para todos os nós. Esta medida permite ligar qualquer aplicação com o ambiente virtual através de uma pasta compartilhada entre o host e o guest (Figura 12).

Glance, uma API que permite criar, reativar, atualizar e destruir uma imagem de máquina virtual. Posteriormente foi efetuado acesso ao *horizon dashboard*, uma interface web usada para administração da infraestrutura do OpenStack, através do endereço do nó controlador. A (Figura 14) demonstra estas etapas.

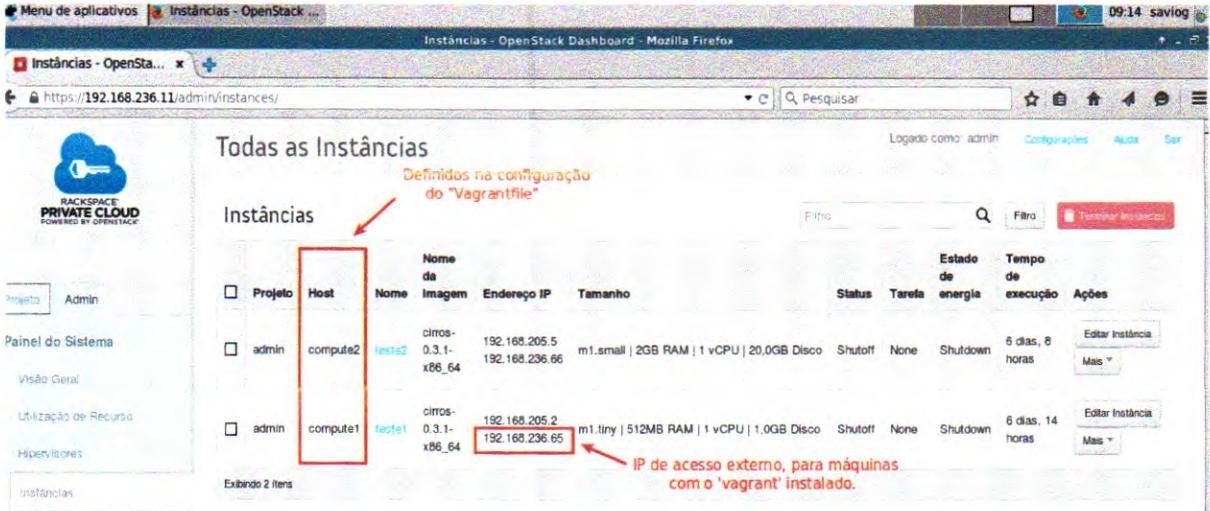
Figura 14 - Obtendo credenciais das API's e criação da imagem cirros (a). Acesso ao painel horizon através do navegador (b)



3.2 Preparação do ambiente de teste de desempenho

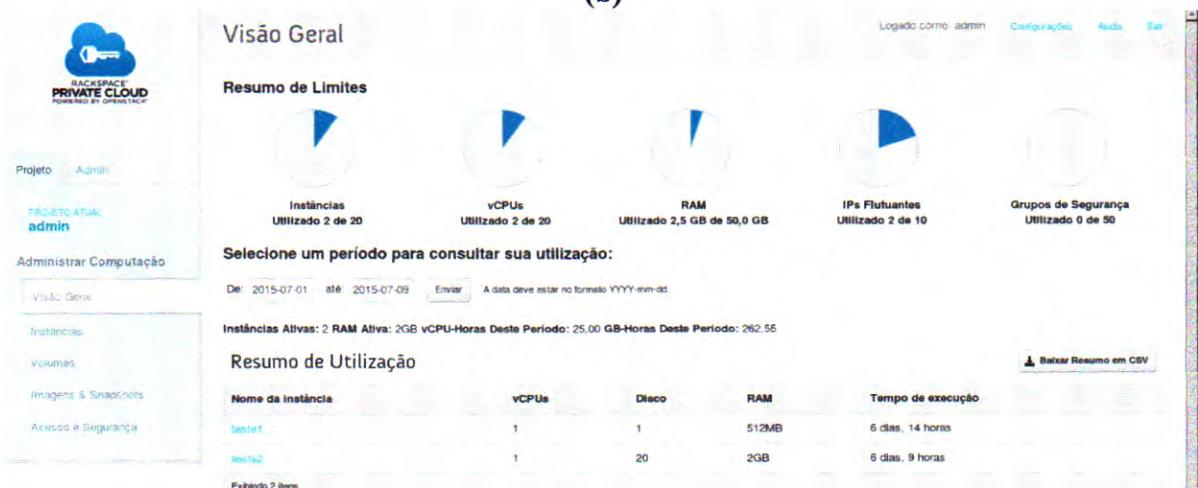
Para obter um ambiente básico para testes com OpenStack foi criado duas instâncias para simular um ambiente de produção, tendo como base a imagem baixada anteriormente. Logo depois foi atribuído um IP flutuante (externo) para acesso dos usuários, através de SSH com '`nome_imagem@IP_flutuante`' onde foi definida as configurações de cada *host* (Figura 15).

Figura 15 - Criação das instâncias e atribuição dos IP's flutuantes

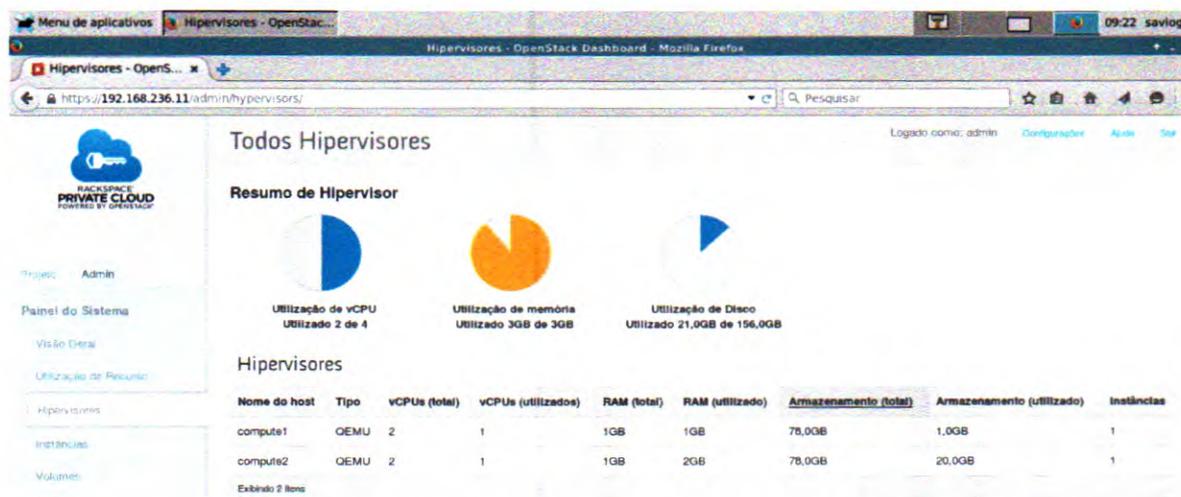


Para o acompanhamento das instâncias foi utilizado o próprio painel do dashboard na guia 'projetos'. E foi obtida uma percepção dos limites estabelecidos para cada projeto e seus resumos que podem ser baixados. Na guia 'Admin' tivemos acesso a todos os hipervisores que demonstram a configuração dos espaços virtualizados e seus limites (Figura 16).

Figura 16 - Visão geral das instâncias criadas (a). Resumo de todos os projetos criados (b)



(a)



(b)

Em sequência foi implementado um processo de análise do desempenho da máquina, durante a virtualização da nuvem, empregando o Virtualbox e utilizando o Vagrant. Para a obtenção dos resultados de desempenho da máquina, foi recorrido ao aplicativo *Monitor de Sistema* disponível no próprio Sistema Operacional Ubuntu 14.04LTS. Os teste foram realizados seguindo uma metodologia de ampliação de uso de recursos computacionais, incluindo máquinas virtuais (em uso) no Virtualbox e ações aplicadas no Vagrant (Tabela 1).

Tabela 1 - Eventos relacionados ao ambiente

Ambiente	Vm's do virtualbox	Ações do vagrant e virtualbox
1	0 Vm's funcionando	0 ações
2	0 Vm's funcionando, Virtualbox iniciado	0 ações
3	1 Vm funcionando	0 ações
4	2 Vm's funcionando	0 ações
5	3 Vm's funcionamento	0 ações
6	0 Vm's funcionando	Vagrant funcionando + interface web Openstack
7	0 Vm's funcionando	Vagrant funcionando + inter. Web + 2 instâncias

A concepção de todo este meio suas implementações e testes, foram realizadas em um único hardware com suas características descritas (Tabela 2).

Tabela 2 - Descrição do hardware utilizado

Descrição	Características e especificações
Notebook	Positivo 1110m Sim+
Sistema Operacional	Ubuntu 14.04.2 LTS
Processador	Dual Core AMD C-60 APU with Radeon(tm) HD Graphics
Memória RAM	6 GB DDR3
VGA	[AMD/ATI] Radeon HD 6290

4 RESULTADOS E DISCUSSÃO

4.1. Testes de Desempenho

4.1.1 Análise de Desempenho com VirtualBox

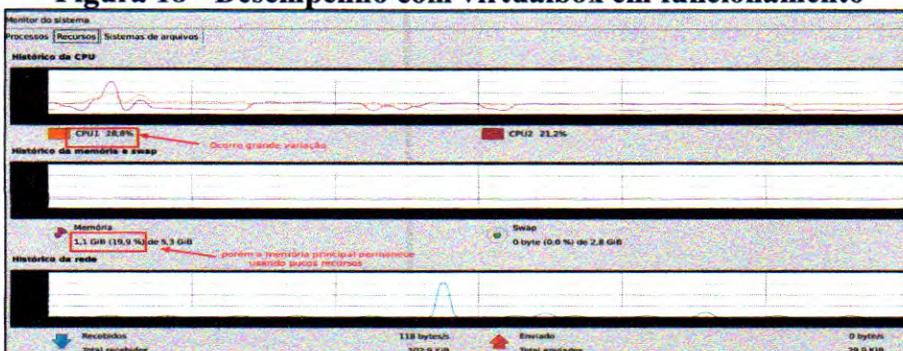
Para analisar o desempenho da máquina foi utilizado o Monitor de Sistema do Ubuntu 14.04, com geração dos dados em tempo real. Com finalização somente com o total consumo dos recursos da máquina. Primeiramente foi definido o estado “zero” de funcionamento da máquina, fixado como o momento posterior á inicialização do sistema e finalização de todos os processos carregados após iniciar. Esta providência previne contra possíveis influencias por parte de atividades executadas em segundo plano (Figura 17).

Figura 17 - Estado 'zero' da máquina



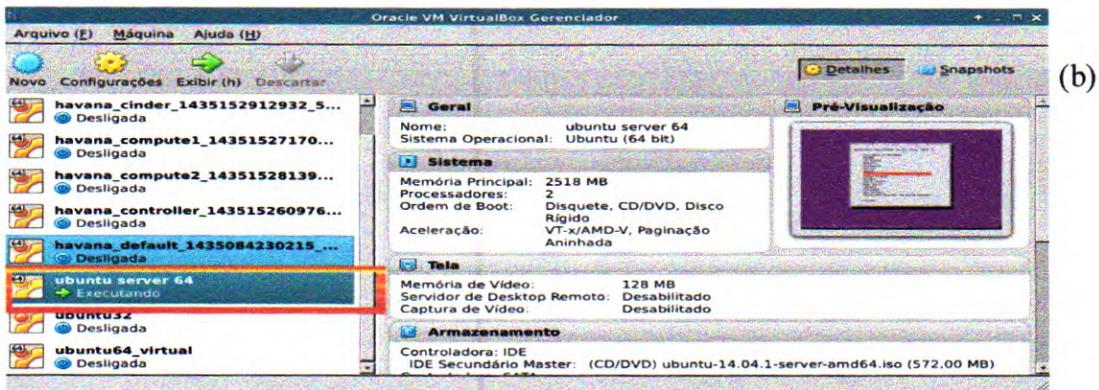
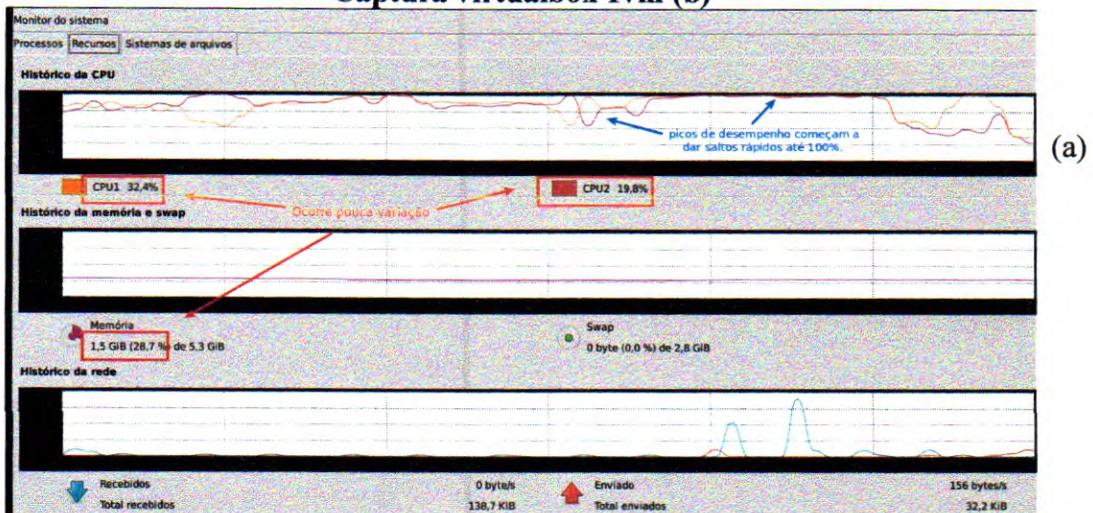
A fim de analisar variações de desempenho do sistema, a próxima etapa foi executada somente com o funcionamento do Virtualbox, sem nenhuma VM iniciada. Durante o teste foi percebido a ocorrência de grande variação em um dos núcleos do processador enquanto o estado inicial da memória RAM permaneceu estável (Figura 18).

Figura 18 - Desempenho com virtualbox em funcionamento



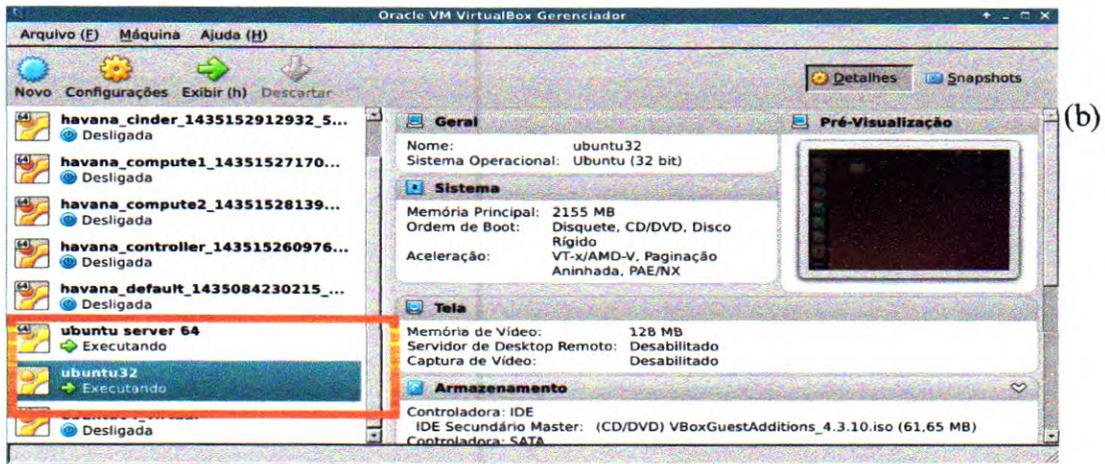
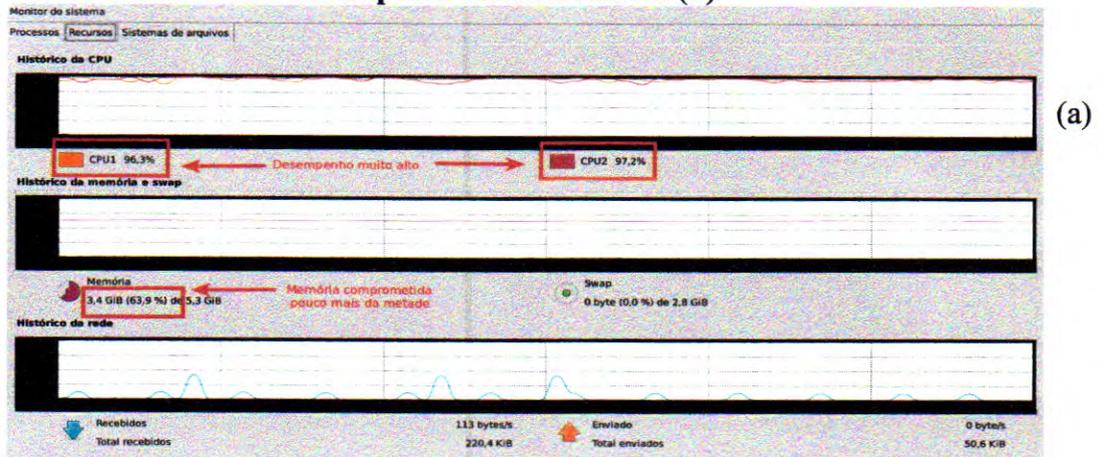
Posteriormente foi iniciada uma única VM com Ubuntu Server 64bits para analisar os dados. Durante o teste foi percebido pouca variação no estado do processador e da memória em relação ao teste anterior, em contra partida, os gráficos dos núcleos realizavam ligeiros saltos até a marca de 100% e retornavam aos valores então capturados (Figura 19).

**Figura 19 - Desempenho virtualbox com 1vm em funcionamento (a).
Captura virtualbox 1vm (b)**



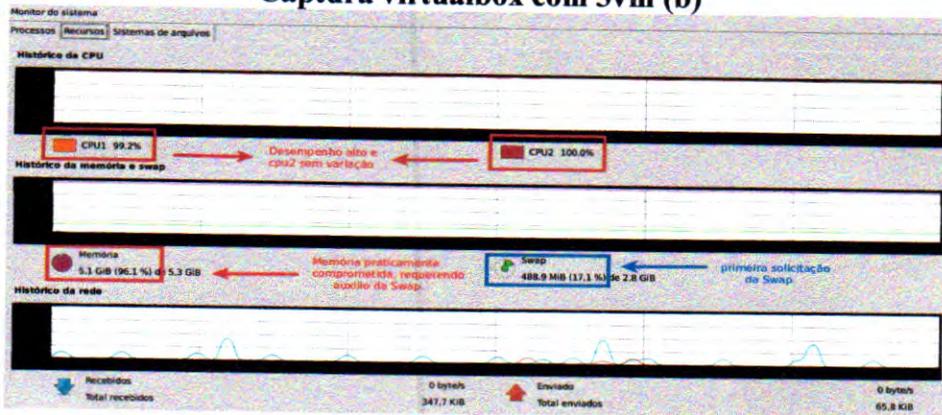
Posteriormente foi iniciada uma segunda VM com Ubuntu 32bits para analisar os dados de desempenho e foi percebido o alto desempenho do processador, chegando aproximadamente a 95% em ambos os núcleos. O estado da memória RAM ficou com pouco mais de 60% de seu desempenho comprometido (Figura 20).

**Figura 20 - Desempenho virtualbox com 2vm em funcionamento (a).
Captura virtualbox 2vm (b)**

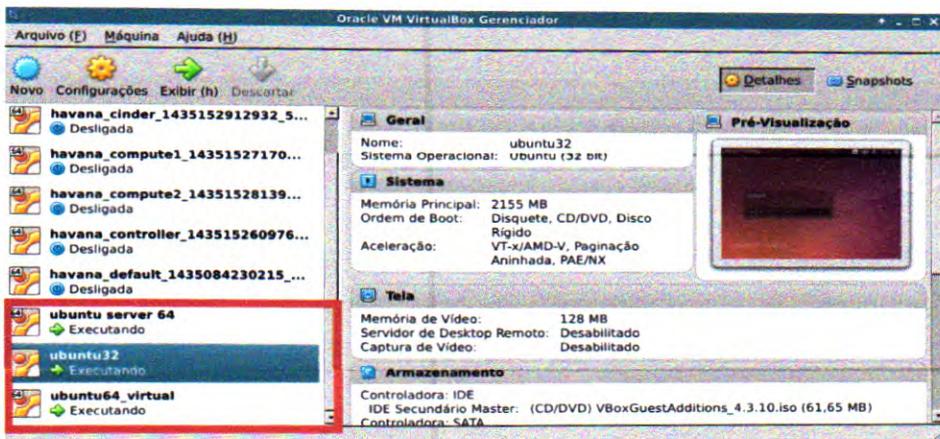


Posteriormente foi iniciado uma terceira VM com Ubuntu 64bits para analisar os dados de desempenho. Durante o teste foi verificado no estado do processador, o desempenho máximo em um de seus núcleos sem qualquer variação, enquanto o outro núcleo tinha variações de desempenho entre 90% e 100%. O estado da memória RAM encontrava-se praticamente comprometida com 96% de sua capacidade, requerendo auxílio de 17% da memória Swap, até antes não utilizada nos testes anteriores (Figura 21).

**Figura 21- Desempenho virtualbox com 3vm em funcionamento (a).
Captura virtualbox com 3vm (b)**



(a)



(b)

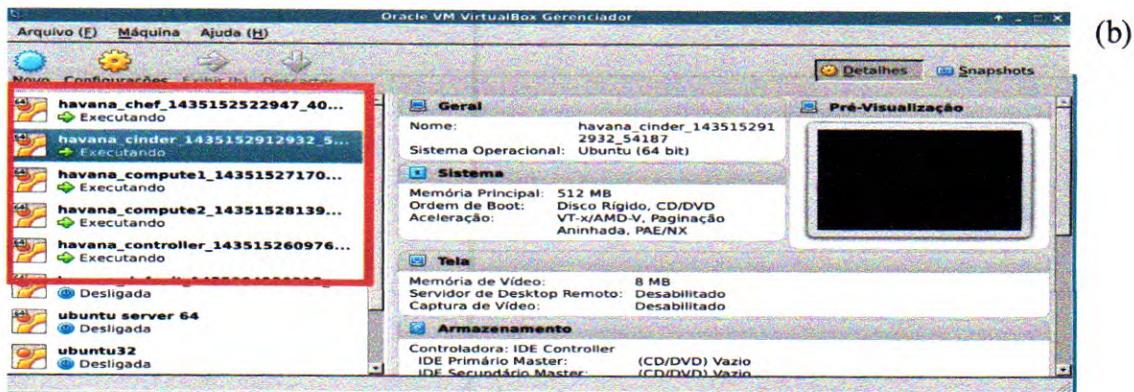
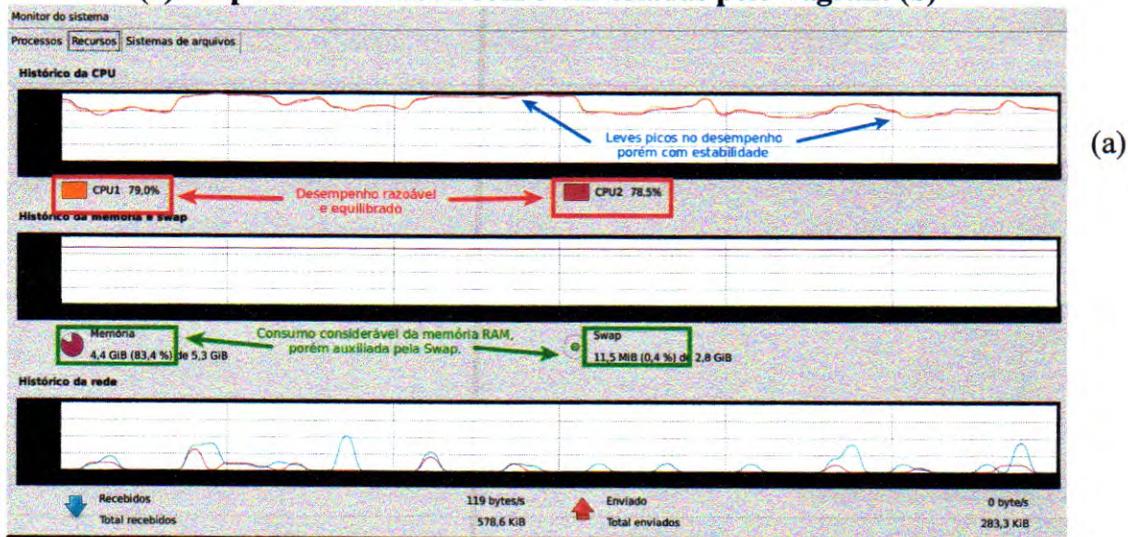
Em seguida foi criada uma quarta VM para ser executada, com o intuito de verificar a permanência do Virtualbox em seu estado de funcionamento. Durante o teste, ao iniciar a VM todos os recursos do PC foram consumidos e a máquina permaneceu parada sem qualquer resposta a comandos ou ações até a realização de um desligamento por meio do 'Power-off'.

4.1.2 Análise de Desempenho com Vagrant

Para analisar o desempenho do Vagrant com Virtualbox foi considerado as ações que mais consomem recursos da máquina a partir de sua inicialização. Para este primeiro teste foi iniciado o Vagrant e feito o acesso a interface web do Openstack, em sequência invocado o modo gráfico do Virtualbox via terminal, apenas para coleta de dados e em seguida fechado. Durante o teste foi verificado um desempenho considerável e equilibrado entre os núcleos do

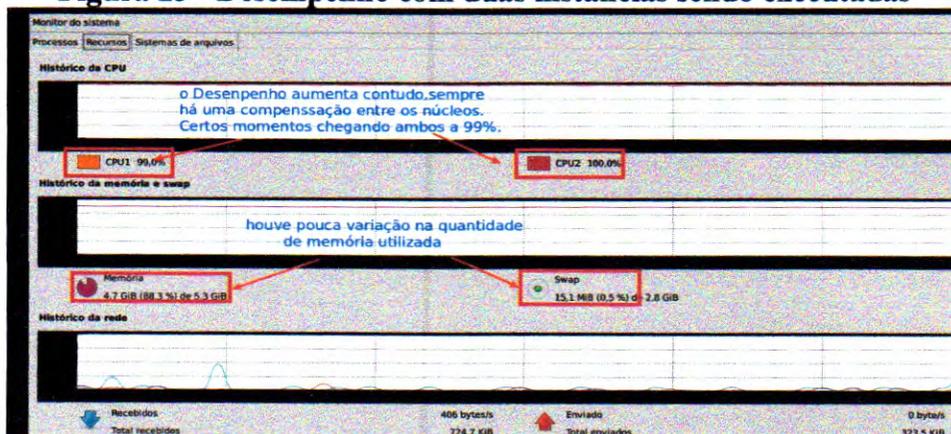
processador e leves picos em seu desempenho, porém mantendo sua estabilidade. Na memória RAM foi observado um consumo considerável e equilibrado com o auxílio da memória Swap (Figura 22).

Figura 22 - Desempenho do Vagrant com Virtualbox e usando interface web
(a). Captura Virtualbox com 5 vm criadas pelo Vagrant (b)



Em seguida foi realizado um segundo teste para análise de desempenho, durante a execução de duas instâncias do OpenStack. Essas instâncias são imagens virtualizadas de servidores, que dentre todas as operações, são as que mais consomem recursos. Durante o teste foi observado o alto desempenho do processador, sem travamento de um dos seus núcleos, foi registrado variações de 85% a 100%. No desempenho da memória RAM e Swap houve pouca variação em relação ao teste anterior (Figura 23).

Figura 23 - Desempenho com duas instâncias sendo executadas



4.2 Análise comparativa dos ambientes.

Diante dos testes realizados observa-se que o ambiente virtual utilizando o Vagrant obteve melhores resultados por usar VM's de configuração básica, gerenciadas pelo mesmo, ocasionando uma economia considerável de recursos à máquina física.

Da mesma forma, o ambiente virtualizado com uso apenas do Virtualbox, obteve menor desempenho econômico, de recursos e tempo por isso, o máximo de máquinas virtuais suportadas não superou o valor 3.

Por conseguinte, observa-se que o ambiente utilizando o software Virtualbox gerenciado pelo Vagrant, otimiza a virtualização de ambientes neste caso, na construção de um ambiente de nuvem pois é melhor que o Virtualbox nas características de isolamento e definição para cada máquina virtual como nós, compartilhamento do ambiente, manutenção rápida e desempenho.

4.3 Definindo variáveis de desempenho

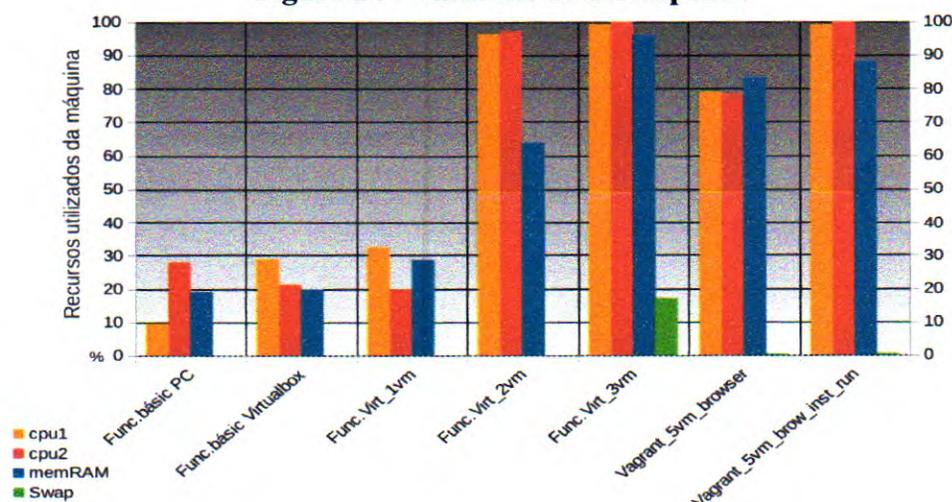
No início, para criar o ambiente de testes na nuvem foi utilizado o software virtualbox, no entanto, devido as características do hardware utilizado e os complexos métodos para a construção deste cenário a implementação tornou-se inviável somente utilizando uma máquina virtual, ocasionando um grande aumento na quantidade de tempo exercido para concluir cada estágio. Após a realização dos testes de ambiente, uma solução foi encontrada no software Vagrant uma ferramenta que permite virtualizar o ambiente de

desenvolvimento, com o auxílio do virtualbox, permitindo criar uma infraestrutura para cada projeto através de várias máquinas virtuais funcionando de forma reduzida e invisível ao usuário.

Ressalta-se que o Vagrant possui um conjunto de ferramentas que têm adicionadas à sua estrutura, uma série de configurações e programas denominados *Box* que funcionam como uma imagem pré-configurada do sistema desejado. Outra ferramenta indispensável para a construção dessa infraestrutura foi o Chef-Server um framework de automação de sistemas e infraestrutura de nuvens, utilizado para implantar servidores e aplicativos tanto no meio físico quanto virtual. Cada estação de trabalho criada pelo Chef-Servidor é mantida e configurada pelo chef-cliente que utiliza cookbooks (livro de receitas) para direcionar como o chef-cliente vai configurar cada nó durante a configuração do ambiente virtualizado.

Durante os processos utilizados para análise dos testes de desempenho, verificou-se o baixo tempo de resposta da máquina nos ambientes de testes 4 e 5. Com isso percebeu-se uma relação entre *consumo* e *tempo de resposta*, quanto mais VM's eram adicionadas ao cenário, mais recursos eram comprometidos acarretando uma diminuição no tempo de resposta em outras ações da máquina. O que levou a realizar constantes acessos aos processos executados na máquina e efetuar o cancelamento daqueles que poderiam influenciar nas variáveis adquiridas nos testes. Esta ação foi realizada entre o intervalo de cada ambiente de teste e no entremeio dos ambientes 5 e 6 foi realizado o *reboot* da máquina. Após este procedimento, listou-se todos os processos atualmente em execução e foi realizado o descarte daqueles que tinham maior preponderância para execução deste cenário. Durante os testes foram coletadas diversas variáveis de desempenho e registradas (Figura 24).

Figura 24 - Variáveis de desempenho



Foi observado que a infraestrutura montada com o Vagrant, apesar de utilizar o Virtualbox demonstrou-se mais eficiente na locação de recursos da máquina, não comprometendo as atividades realizadas no Openstack, ambiente escolhido para a implementação. Com os resultados devidamente registrados verificou-se que ao iniciar o virtualbox, o consumo dos recursos permanece mínimos desde a inicialização da máquina até a adição de 1 VM. Após a inclusão de mais uma VM houve uma considerável corrida por recursos, começando a afetar outras funções da máquina. Ao chegar no 5º ambiente de teste quase todos os recursos haviam sido consumidos, um dos núcleos do processador já estava em seu consumo máximo, a memória RAM praticamente comprometida e cerca de quase 20% da memória Swap utilizada (na cor verde, mostrado na Figura 4.8).

Portanto, verifica-se com os resultados obtidos nos dois últimos testes, as vantagens de realizar uma implementação utilizando o Vagrant, o consumo dos recursos é aceitável uma vez que não prejudica a manipulação de elementos do projeto (a nuvem). Nos dois testes realizados com o Vagrant é utilizado uma pequena parte da memória Swap (em verde) o que trás um alívio á memória RAM não sobrecarregando o funcionamento do ambiente. Outros elementos importantes levados em consideração são os benefícios em realizar etapas de configurações tão complexas de forma automatizada e concisa.