UNIVERSIDADE FEDERAL DE SANTA CATARINA

ESTUDO E IMPLEMENTAÇÃO DE UM SERVIDOR DE VIRTUALIZAÇÃO GERENCIÁVEL PARA FINS ACADÊMICOS

Guilherme Celmer Balz

Florianópolis - SC

2011/2

Universidade Federal de Santa Catarina

Departamento de Informática e Estatística

Curso de Sistemas de Informação

ESTUDO E IMPLEMENTAÇÃO DE UM SERVIDOR DE VIRTUALIZAÇÃO GERENCIÁVEL PARA FINS ACADÊMICOS

Guilherme Celmer Balz

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do
grau de Bacharel em Sistemas de Informação
pela Universidade Federal de Santa Catarina.

Florianópolis - SC

2011/2

Guilherme Celmer Balz

ESTUDO E IMPLEMENTAÇÃO DE UM SERVIDOR DE VIRTUALIZAÇÃO GERENCIÁVEL PARA FINS ACADÊMICOS

Trabalho de conclusão de curso apresentado como parte dos requisitos de obtenção do grau de Bacharel em Sistemas de Informação pela Universidade Federal de Santa Catarina.

_____

Frank Augusto Siqueira

Orientador

Banca Examinadora

_____

Antônio Carlos Mariani

_____

Mário Antônio Ribeiro Dantas

RESUMO

A virtualização de sistemas operacionais é uma tendência global, motivada pelos enormes benefícios que traz em sua essência. Apesar de não ser novidade, vem ganhando muita atenção por agregar recursos computacionais de forma otimizada, buscando proporcionar ganhos em poder de processamento que estendem a capacidade de parques computacionais em prover novas vias e mecanismos de informação. Surgida recentemente, a expressão Computação em Nuvem identifica esta tecnologia, originada da essência da virtualização, cuja evolução está em ascensão muito antes da popularização do termo, que se refere aos serviços de computação disponibilizados em rede e pela Internet. Estes serviços, providos por estruturas dinâmicas e distribuídas geograficamente, abstraem o ambiente físico para o usuário, oferecendo tecnologias de informação *online* de processamento com recursos e benefícios antes inimagináveis.

Dentro deste cenário, é realizado um estudo das tecnologias de virtualização de sistemas disponíveis, com o objetivo de encontrar ferramentas que permitam disponibilizar um serviço de máquinas virtuais para alunos, professores e pesquisadores da universidade, de forma simplificada e autônoma. Neste contexto, são apresentadas algumas ferramentas de código livre, dando ênfase ao funcionamento e benefícios oferecidos por elas, a fim de identificar as mais apropriadas para viabilizar a concretização da proposta.

Palavras-chave: computação em nuvem, virtualização, software livre, openvz

ABSTRACT

The virtualization of operating systems is a global trend, driven by the enormous benefits that it brings in its essence. Although it's not a recent concept, it has gained many attention by optimizing computing resources, seeking to provide gains in processing power that extend the computational capacity of datacenters to provide new technics and mechanisms for information technology. Emerged recently, the term Cloud Computing – which originates from the essence of virtualization – is the identification of this new technology whose evolution is on a rise for a long time before the popularization of the term, by services that are available in network computing and the Internet. These services, provided by geographically distributed and dynamic structures, are abstracted physically to the end-user, providing online information technology resources with features and benefits that were previously unimaginable.

Within this scenario, was conducted a study of system virtualization technologies available in order to find tools to provide a service of virtual machines for students, teachers and university researchers, in a simplified form and unattended. Trough this context, are presented some open source tools, with emphasis on the functionalities and benefits offered by them in order to identify the most appropriate technologies to enable the implementation of this proposal.

Keywords: virtualization, cloud computing, open source, openvz

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

VE	*Virtual Environment (ambiente virtual)*
VM	*Virtual Machine (máquina virtual)*
x86	*Sigla referente à Arquitetura de Processadores Intel*
VMM	*Virtual Machine Manager (gerenciador de máquinas virtuais)*
I/O	*Instruções de processamento de entrada e saída (input/output)*
CPU	*Central Processing Unit (unidade central de processamento)*
API	*Application Programming Interface (interface de programação de aplicativos)*
WSDL	*Web Service Definition Language (padrão de linguagem para web services)*
VPS	*Virtual Private Server (servidor virtual privado)*
GPL	*General Public License (licença pública geral)*
RPM	*Recursive Package Manager (repositório de arquivos recursivos)*
Yum	G*erenciador de instalação de pacotes para distribuições Linux.*
GUI	*Graphical User Interface (interface gráfica do utilizador)*
MD5	*Message-Digest Algorythm (algoritmo de criptografia unidirecional)*

GLOSSÁRIO E TRADUÇÕES

Mainframe	Computador de grande porte.
Hypervisor	Gerenciador de máquinas virtuais situado em uma camada acima do *supervisor* (o núcleo do sistema operacional).
Driver	Controlador de dispositivos, responsável por traduzir instruções para um componente ou software específico.
Kernel	Núcleo do sistema operacional, responsável pela intercomunicação do sistema operacional com o hardware.
Container	Referente a um espaço reservado e isolado.
Workstation	Estação de trabalho (geralmente individual).
Guest	Referente a máquinas virtuais em um sistema virtualizador.
Supervisor	Referência ao gerenciador de processos que trabalha em conjunto com o núcleo do sistema operacional.
Live Migration	Referente à migração ativa de máquinas virtuais entre servidores.
Overhead	Sobrecarga de processamento, seja do tempo de computação, uso de memória ou outros recursos requeridos para executar uma determinada tarefa.
Framework	Espaço de trabalho ou conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação.
Web Service	Solução utilizada na integração de sistemas e na comunicação entre diferentes aplicações, geralmente realizada através da Internet.
Entrepreneurs	Investidores de risco, que apostam em uma idéia ou novo negócio.
Creative Commons	Organização não governamental sobre direitos de propriedade que oferece licenças que permitem a cópia e compartilhamento com menos restrições.
Template	Modelo ou padrão, pronto para uso.
Cluster	Conjunto de computadores interligados para somar poder computacional.
Front-end	Refere-se à interface que permite interagir com o sistema, seja para o usuário ou para outro sistema.
Back-end	Referente à estrutura base de um sistema.
Buffer	Região de memória temporária para dados pré-processados.

Cache	Memória ou dispositivo de acesso rápido.
Tarball	Formato de arquivo que agrupa diversos outros, geralmente compactados.
Chroot	Operação em Linux que isola um diretório raiz de outro, no caso, o principal.
Script	Conjunto de instruções de linguagem de programação executável.
Metadata	Refere-se a “dados de outros dados”, no caso, inteligível por um sistema.
Loop	Ciclo, em repetição. Refere-se a um programa em execução infinita.
Hash	Sequência (pedaço) de bits, geradas por um algoritmo de dispersão.

1. Introdução

Nos últimos anos, a evolução tecnológica, em especial na área da computação, tem apresentado mudanças e inovações a cada momento. O acesso a meios computacionais e o crescimento do poder de processamento dos computadores vêm alcançando níveis que jamais foram imaginados, permitindo difundir a tecnologia de informação para as mais variadas áreas de estudo, entretenimento, conhecimento e comunicação. Esta crescente demanda trouxe também a necessidade de aperfeiçoar o uso dos sistemas computacionais, procurando disponibilizá-los de forma a reduzir o sub-aproveitamento e buscando minimizar os mais variados desperdícios, como o excesso de equipamentos e o consumo de energia, procurando alinhar-se a uma das maiores preocupações da sociedade contemporânea: o uso consciente e responsável dos recursos.

Com a evolução do *hardware* dos computadores, é muito comum a subutilização dos recursos computacionais em praticamente todos os parques computacionais. São feitos investimentos pouco calculados em máquinas com relação aos sistemas para os quais elas são destinadas, criando ambientes de processamento ociosos, principalmente pela inexperiência de profissionais na administração destes recursos.

A virtualização já é um assunto bastante difundido desde o final da década de 60, onde os *mainframes* eram extremamente caros e procurava-se aproveitar ao máximo dos recursos que dispunham. Porém, com a evolução e a conseqüente redução dos custos, pensou-se demais na eficiência e pouco na eficácia. Máquinas poderosas, com vários núcleos de processamento, são disponibilizadas para gerir ambientes com baixas demandas, sendo que estas poderiam comportar o ambiente de várias outras máquinas físicas em uma única, através da virtualização.

Na computação, o termo “virtualização” refere-se a uma forma de esconder as características físicas de uma plataforma para os seus usuários, simulando a existência de um único *hardware* para mais de um sistema operacional. Esta camada de virtualização, configurada

inicialmente ou dentro de um sistema operacional, isola um “computador virtual” do outro, procurando evitar a interferência entre eles e também preservar seu desempenho, permitindo que cada um subentenda que os recursos são únicos e exclusivos. Dentro deste paradigma, surgiu uma enorme variedade de cenários e meios de virtualização, oferecendo vantagens econômicas de processamento e recursos extremamente atrativas, sendo difundido em praticamente todos os ambientes computacionais.

Utilizando-se da alta disponibilidade de recursos computacionais e do advento das redes de computadores em escala mundial, foram criadas diversas oportunidades de expandir este universo, muito além do compartilhamento de dados e informações. Uma delas, intimamente ligada a virtualização, é denominada *computação em nuvem*. A computação em nuvem, ou *cloud computing*, engloba em sua definição o fornecimento ou uso de recursos computacionais sem depender de sua localização física, sendo abstraída aos seus usuários.

De uma forma simplificada, o conceito de computação em nuvem pode ser descrito como um conjunto de recursos computacionais, distribuídos geograficamente e acessíveis por meio da Internet, sendo disponíveis para praticamente qualquer lugar do mundo e oferecidos em tempo integral. Estes recursos geralmente não necessitam a instalação de programas no computador do usuário, pois são oferecidos na forma de aplicações web, processando e armazenando dados remotamente. Por este motivo tem-se a alusão à nuvem, que simboliza a rede mundial de computadores.

Serviços oferecidos em ambientes de computação em nuvem estão cada dia mais populares, sendo oferecidos por diversas companhias e também trazidos para dentro de muitas organizações como forma de otimizar o uso dos recursos computacionais, migrando seus servidores e aplicações para plataformas virtualizadoras disponíveis na nuvem, atraídos pelas vantagens que a tecnologia oferece. Grandes empresas como o Google, Oracle, Amazon, Microsoft

e tantas outras fazem pesados investimentos em desenvolvimento de tecnologias e aprimoramento dos seus meios computacionais, oferecendo soluções e serviços para os mais distintos usuários, principalmente competindo por grandes clientes na incessante busca pela referência no mercado.

Da mesma forma que as tecnologias de informação evoluem a passos largos, com constantes mudanças e influenciando nossa vida em praticamente todas as atividades, profissionais e acadêmicos dos cursos de computação e tecnologia de informação devem ser preparados para aproveitar os benefícios destes novos recursos. Como forma de incentivo, é necessário oferecer meios que estendam a utilização, estudo e desenvolvimento destas ferramentas, disponibilizando ambientes onde possam ser exercitados fundamentos de programação e tecnologias de informação de forma altamente didática, consolidando conhecimentos através da prática e verificação dos resultados.

Dentro deste cenário, serão explorados os conceitos e definições das tecnologias de virtualização, abordando os diferentes cenários e suas características. Também serão abordados os conceitos de computação em nuvem, apresentando informações relevantes que os fundamentam com relação ao cenário de serviços. Foram realizados testes e experimentos com ferramentas de software livre que utilizam diferentes técnicas de virtualização, buscando encontrar uma solução ágil, robusta e de alto desempenho que possa ser aplicada e disponibilizada para oferecer um serviço de virtualização de sistemas operacionais para a comunidade acadêmica.

Sendo assim, os resultados deste estudo poderão servir de referência para avaliar os benefícios da utilização do recurso no meio acadêmico, bem como para a implantação do serviço.

1.1 Objetivos

1.1.1 Objetivo Geral

Apresentar uma proposta para oferecer um serviço de ambientes de virtualização estruturado em software livre, dedicado a utilização em estudos e pesquisas por alunos, professores e laboratórios da Universidade.

1.1.2 Objetivos Específicos

Através deste trabalho pretende-se:

- Efetuar um estudo teórico e prático das tecnologias de virtualização e suas ferramentas, contextualizando os tipos e as vantagens que cada técnica oferece atualmente, buscando aprofundar os conhecimentos nas áreas de computação em nuvem, sistemas de alta disponibilidade e de virtualização de servidores;
- Apresentar as principais técnicas de virtualização disponíveis e seus derivados, assim como algumas das ferramentas desenvolvidas em software livre existentes no mercado, compatíveis com o objetivo proposto;
- Levantar os requisitos, ferramentas e requisitos necessários para fundamentar um modelo de implantação e disponibilização do serviço;
- Analisar as ferramentas e sistemas a serem utilizados, compatíveis com o objetivo da proposta;
- Preparar um ambiente de testes que possa servir para simulações e avaliações;
- Pesquisar e analisar de ferramentas disponíveis para gerenciamento e configuração das máquinas virtuais;
- Apresentar uma análise dos resultados obtidos com a pesquisa.

1.2 Motivação

A motivação deste trabalho surgiu da necessidade de oferecer um ambiente de virtualização de servidores de fácil acesso a alunos, professores e pesquisadores da Universidade, visando encontrar uma solução disponível em software livre que ofereça a oportunidade de preparar e implantar um serviço de criação de máquinas virtuais temporárias para estudos e testes, sendo facilmente gerenciada e administrada.

Da mesma forma, estimular o estudo e desenvolvimento de tecnologias de informação que utilizem o conceito de computação em nuvem no meio acadêmico, permitindo que alunos, professores e pesquisadores tenham acesso a um recurso computacional que possibilite realizar testes práticos e experimentos em um ambiente estável e de alta disponibilidade.

Como adicional, promover a cultura e uso de softwares de código livre, permitindo que ela seja empregada e desenvolvida tanto no meio acadêmico como também estimular a adoção por empresas e corporações. Da mesma forma, promover a utilização de sistemas que otimizem parques computacionais, proporcionando uma melhor utilização dos seus recursos e encorajando a adoção de novas tecnologias aliadas a sustentabilidade.

1.3 Metodologia

A preparação deste trabalho iniciou através do levantamento bibliográfico de informações referentes à virtualização de sistemas operacionais e suas metodologias, buscando dar ênfase às aplicações e técnicas que surgiram nos últimos anos, com novas ideologias e serviços baseados em sua estrutura. Dentre estes, serão abordados os conceitos de computação em nuvem e computação utilitária, assim como as tecnologias que viabilizam de forma consolidada a implantação de ferramentas de virtualização.

Devido à variada gama de assuntos relacionados à virtualização, o trabalho restringe-se a focar apenas na virtualização de sistemas operacionais, seus derivados e soluções, explorando seus recursos e mecanismos. Apesar de possuir outros tipos relacionados diretamente, as referências a virtualização neste trabalho estão vinculadas apenas ao cenário 'computadores e sistemas operacionais'.

Sendo assim, a construção deste trabalho foi feita de acordo com as seguintes etapas:

- Detalhamento teórico das tecnologias de virtualização e suas aplicações, apontando suas finalidades e diferenciais;
- Seleção de ferramentas de virtualização baseadas nestas tecnologias, com recursos que permitam criar estruturas de virtualização de servidores;
- Realização de testes, levantamento de informações e demandas para elaboração de uma solução que possa atender as necessidades apontadas;
- Apresentação dos resultados obtidos com a pesquisa, revelando um balanço das tecnologias estudadas e suas vantagens referentes às necessidades apontadas.

1.4 Estrutura do Trabalho

Este trabalho está organizado em seis capítulos. O primeiro capítulo apresenta uma introdução sobre o tema abordado no trabalho, apresentando os objetivos e a metodologia utilizada. O segundo capítulo apresenta um estudo sobre as tecnologias de virtualização desenvolvidas, apresentando seus conceitos, modelos e principais características identificadas. No terceiro capítulo são apresentados os conceitos de computação utilitária e computação em nuvem, além de serviços em que são baseados e também trabalhos relacionados. No quarto

capítulo são abordadas as ferramentas de virtualização e computação em nuvem, apresentando a estruturação e vantagens de cada uma, assim como algumas aplicações que oferecem recursos de gerência de ambientes virtualizados. O quinto capítulo descreve o desenvolvimento do projeto, com os resultados obtidos a partir dos estudos e experimentos realizados. O sexto capítulo destaca as considerações finais sobre o trabalho, bem como sugestões para trabalhos futuros relacionados com a tecnologia de virtualização e com sua utilização no ambiente acadêmico.

2. Conceitos, Definições e Trabalhos Relacionados

2.1 Virtualização de Sistemas Operacionais

A virtualização de sistemas operacionais vem ganhando espaço nos últimos anos por ser uma técnica eficiente de partilhar recursos computacionais de um sistema computacional, permitindo a execução de vários ambientes virtuais de forma simultânea e isolados entre si, realizando um maior aproveitamento dos recursos computacionais e reduzindo a ociosidade. Cada máquina virtual pode oferecer um sistema computacional completo muito similar a uma máquina física, com seu próprio sistema operacional, aplicativos e ser capaz de oferecer serviços de rede. É possível ainda interconectar cada uma dessas máquinas através de interfaces de rede (CARISSIMI, 2008).

De uma forma simples, a virtualização possibilita ter dois ou mais ambientes computacionais em execução utilizando apenas um *hardware*, como mostra a figura 1. Por exemplo, é possível ter uma máquina virtual Linux e outra Microsoft Windows, em execução simultânea, utilizando um único servidor físico (*GOLDEN & SCHEFFY,* 2008).

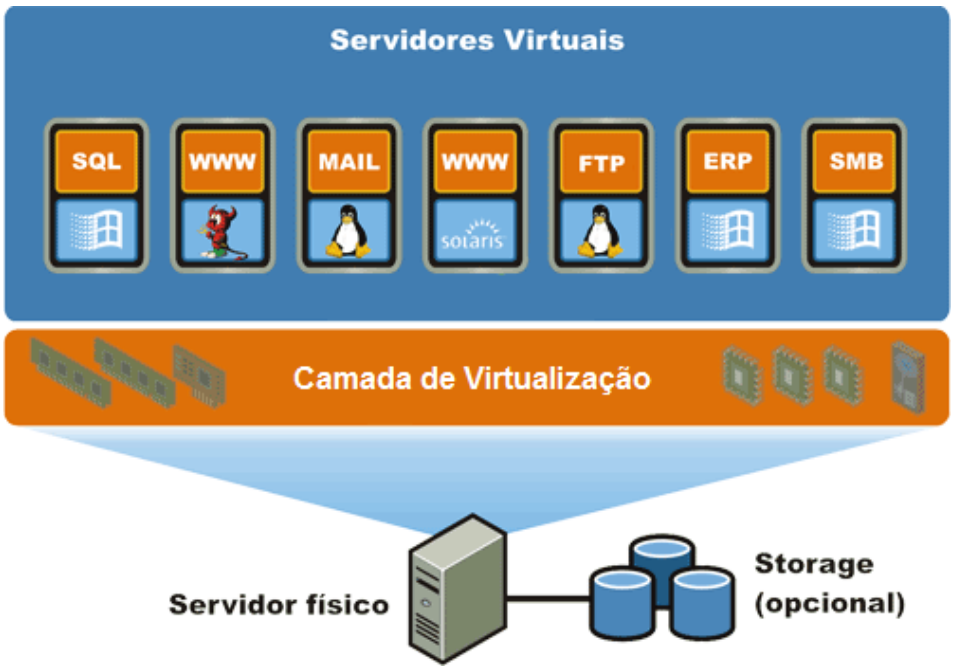

Figura 1 – Exemplo de um servidor físico virtualizando sistemas operacionais distintos.

Para permitir isto, são utilizados sistemas operacionais e softwares com técnicas avançadas de abstração e emulação, sempre mantendo esforços para prover o máximo de segurança, desempenho, estabilidade e confiabilidade dos meios envolvidos. Além disso, é uma proposta para consolidação de servidores e parques computacionais, permitindo que organizações reduzam o número de servidores físicos através da alocação dos sistemas operacionais em máquinas virtuais, resultando em um menor consumo energético, espaço físico e redução de custos com hardware e manutenção (KOLYSHKIN, 2006).

Segundo Silva (2007), as máquinas virtuais referem-se a sistemas virtuais sobre uma máquina real. Prova disso é que, em uma mesma máquina, elas compartilham atributos comuns, como:

- Compatibilidade do software: a máquina virtual fornece uma abstração compatível de modo que todo o software escrito para ela funcione;
- Isolamento: garante que os softwares executados em cada uma das máquinas virtuais e os da máquina real estejam totalmente isolados entre si;
- Encapsulamento: é usado para manipular e controlar a execução do software na máquina virtual;
- Desempenho: adicionar uma camada de software a um sistema pode afetar o desempenho do software que funciona na máquina virtual, mas os benefícios de uso de sistemas virtuais devem compensar.

Para os usuários, a virtualização pode oferecer diversos benefícios, entre eles:

- Ambientes de desenvolvimento ("*sandboxes*") ágeis, onde desenvolvedores de sistemas operacionais e softwares podem realizar testes, modificações, avaliar o desempenho, etc. em um ambiente que pode ser restaurado imediatamente após panes ou falhas.
- A capacidade de recuperar rapidamente um sistema afetado por ataques maliciosos deliberados ou falhas acidentais de funcionamento, através da restauração do sistema por uma cópia estável.
- A disponibilidade de realocar as máquinas virtuais quando um servidor se torna indisponível em um ambiente composto por mais de uma máquina física, a fim de balancear a carga dinamicamente sem que o usuário perceba, usando os recursos agregados de forma mais eficiente;
- A capacidade de consolidar múltiplos serviços em uma única máquina física ou cluster, permitindo que cada um seja administrado independentemente de maneira autônoma, onde cada um pode fazer suas próprias modificações e escolhas sem necessidade de consultar os proprietários dos demais sistemas;
- Melhor aproveitamento do poder computacional que uma única máquina dispõe, principalmente em plataformas multiprocessadas;
- Planejamento e uso consciente dos recursos computacionais como um todo, reduzindo desperdícios e otimizando sua utilização.

Em uma perspectiva empresarial, a virtualização pode oferecer reduções de custos com a melhor utilização de hardware, reduzindo a quantidade de equipamentos necessários para uma infraestrutura de computação que necessita vários sistemas diferentes. Custos de treinamento para empregados podem ser reduzidos quando se utiliza virtualização porque permite que muitas configurações de treinamento diferentes (de sistemas operacionais e aplicativos) coexistam numa

plataforma única, reduzindo dessa forma a quantidade de computadores necessários para treinamento, e a reconfiguração também será mínima entre diferentes sessões. Além disso, a virtualização pode também oferecer a vantagem de executar sistemas ou aplicativos antigos em hardware moderno, sem temer problemas de compatibilidade ou desempenho (MATHEWS; DOW, 2006).

Já em uma perspectiva educacional, o emprego de virtualização pode trazer inúmeras vantagens, tornando o processo de ensino mais vantajoso em muitas disciplinas. Pode ser proporcionado um ambiente prático para programar e lecionar, onde tanto alunos quanto professores têm acesso a recursos que um ambiente virtualizado oferece sem maiores preocupações quanto à disponibilidade de equipamentos ou utilização de laboratórios físicos. Também reduz a dependência de manutenção, reinstalação de sistema e aplicativos, otimizando e agilizando o processo de ensino.

Pela finalidade proposta de ser um sistema para promover o aprendizado, é comum acontecerem erros durante o processo em experimentos práticos, muitas vezes resultando em danos para o sistema ou para as ferramentas instaladas que o usuário teria dificuldades em recuperar. A virtualização promove a facilidade de contornar estes problemas, oferecendo meios de restaurar um sistema corrompido em poucos minutos, evitando também que os demais sejam afetados.

2.1.1 Tipos de Virtualização de Sistemas Operacionais

Muitos autores identificam as soluções de virtualização de várias maneiras, classificando as técnicas de acordo com suas arquiteturas ou metodologias.

Segundo Silva (2007), nas soluções existentes atualmente, a virtualização de sistemas operacionais é implementada por duas metodologias, sendo a primeira envolvendo somente software e a segunda combinando hardware e software.

A virtualização por software tem suas origens no começo da história dos computadores, por volta dos anos 60, onde grandes *mainframes* demandavam um aproveitamento otimizado em razão do seu alto custo e concorrência de uso, necessitando mecanismos que permitissem gerenciar seus recursos de forma dinâmica e disponibilizá-los para uso por múltiplos usuários (SILVA, 2007).

A virtualização por software consiste em múltiplas instâncias isoladas em nível de usuário em vez de apenas uma, em um único sistema operacional compatível (SILVA, 2007) ou utilizando um software de virtualização de hardware. Diferentemente da metodologia combinada por hardware e software, não requer assistência do hardware para executar as máquinas virtuais (VMs, do inglês *Virtual Machine*) de maneira eficiente.

Já a metodologia de virtualização combinando hardware e software, conhecida como virtualização assistida por hardware, apresenta vantagens em relação ao desempenho por utilizar instruções de processadores especializados que realizam o gerenciamento de diversos recursos do computador, otimizando ciclos de processamento e consequentemente trazendo ganhos de performance. O auxílio de um software intermediário, o *hypervisor*, realiza o controle de separação destas instruções e monitoramento, sendo responsável por organizá-las e por efetuar o tratamento necessário.

Cada técnica de virtualização efetua um balanceamento entre nível de isolamento e aumento de compartilhamento de recursos entre as máquinas virtuais. Em tese, quanto maior o isolamento, maior o custo de desempenho para gerenciar estes mecanismos, assim como a

complexidade de implementá-los. Do contrário, técnicas de isolamento mais fraco podem ter como benefício um melhor desempenho.

Dentro destas características, cada metodologia possui técnicas e conceitos bem difundidos, com vantagens de acordo com seus propósitos, conforme as principais abordagens descritas a seguir.

2.1.2 Virtualização por Software

A virtualização ou simulação de hardware consiste em um software especializado que habilita a hospedagem de máquinas virtuais em um sistema operacional. É classificada como virtualização por software, onde dentro de sua estrutura, o virtualizador fornece todos os requisitos necessários para simular um hardware físico, baseando-se nos recursos disponibilizados pelo sistema hospedeiro (memória, disco, interface de rede, vídeo, processamento) e também simulando *drivers* para os dispositivos. Basicamente, torna o sistema operacional virtualizado praticamente idêntico a uma aplicação em execução dentro de outro sistema, como mostra a figura 2.

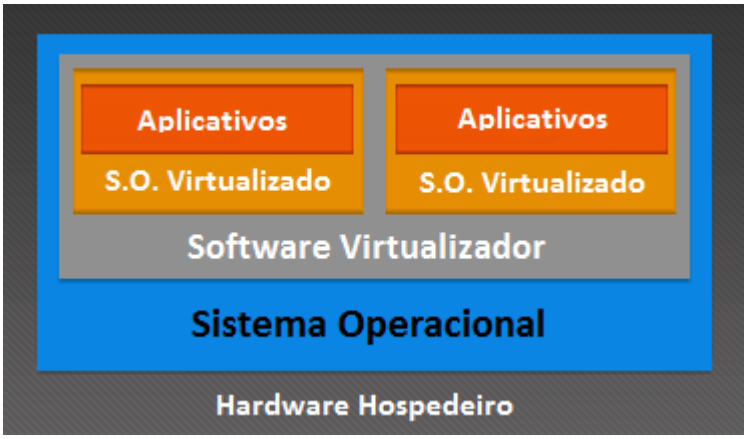

Figura 2 – Simulação de hardware para máquinas virtuais em um sistema hospedeiro.

Conhecida também como virtualização x86, a técnica consolidou-se no final dos anos 90, onde devido à ausência de suporte nativo para virtualização nos processadores fabricados na época para esta arquitetura, tornou-se necessário buscar um maior aproveitamento dos recursos. Foram desenvolvidas complexas técnicas de software que permitiram atingir bons níveis de desempenho na execução (WORLD CONGRESS ON ENGINEERING, 2009).

As ferramentas mais conhecidas que oferecem soluções baseadas em virtualização por software ou que fazem uso de técnicas relacionadas a ela são: VMWare, Xen, QEMU, KVM, VirtualBox, Microsoft Virtual PC, Microsoft Virtual Server, Solaris Zones e FreeBSD Jails.

Vale reforçar que a virtualização de hardware não deve ser confundida com emulação de hardware, apesar da semelhança dos conceitos. Um emulador consiste em um agente escrito que permite a interação de sistemas de arquiteturas distintas e fisicamente incompatíveis, intermediando o processo de tradução de instruções. A virtualização utiliza-se de alguns recursos de emulação que, acrescidos a um conjunto completo de recursos, permite múltiplas execuções visando o máximo de desempenho (SILVA, 2007).

A virtualização por software possui como vantagens a portabilidade e o baixo custo de implementação, permitindo ser facilmente instalada em um sistema hospedeiro, pois geralmente não requer alterações na sua estrutura. Ela habilita, por exemplo, executar um sistema operacional Linux em uma máquina física hospedeira com Windows XP, assim como o contrário, sem modificações na instalação.

Apesar de ser uma técnica bastante difundida e consolidada, a virtualização de hardware não é uma técnica recomendada para ambientes e sistemas críticos, pois em determinadas situações ela pode vir a ‘sugar’ ou esgotar os recursos do sistema hospedeiro, comprometendo as demais VMs ou o sistema como um todo. É uma excelente alternativa para criar ambientes de testes rápidos em um computador *desktop*, visando, por exemplo, avaliar a compatibilidade de um

aplicativo em diferentes versões de sistemas operacionais, testar acesso a bancos de dados, executar um aplicativo sem suporte a um sistema atual, simular ataques de segurança, entre outros.

2.1.3 Virtualização no Nível de Sistema Operacional

A virtualização no nível de sistema operacional é um modelo de virtualização baseada em software onde o *kernel* de um sistema operacional nativo é modificado para hospedar múltiplas instâncias isoladas no nível de usuário. Estas instâncias, chamadas de *containers* ou ambientes virtuais (*virtual environments* – VEs), assemelham-se a um servidor real e também se comportam como um ao ponto de vista do usuário. (WIKIPEDIA; Operating System-Level Virtualization, 2011). Uma máquina virtual no nível de sistema operacional pode ser alocada em poucos segundos a partir do sistema operacional base, que particiona seus recursos a partir do redirecionamento das requisições de acesso a recursos, provendo isolamento entre as instâncias.

Esta técnica permite criar múltiplos ambientes virtuais seguros e isolados em um único servidor, com semelhança total a um sistema operacional completo. Podem ser reiniciadas independentemente e podem executar diferentes distribuições, com acesso administrativo e recursos dedicados, tornando cada instância um ambiente exclusivo e seguro.

É otimizada de forma a garantir o melhor desempenho, gestão de recursos e eficiência, pois sua arquitetura possui baixas perdas de desempenho pelo fato de utilizar um único *kernel* na sua estrutura, o que ajuda a maximizar o uso dos recursos físicos do servidor, permitindo a execução de múltiplos *containers* isolados entre si em um mesmo hardware, conforme ilustra a figura 3. Além disso, possui implementações que tornam possíveis migrações ativas – transferência de um servidor virtualizado de uma máquina entre nós físicos – habilitando o balanceamento de carga sem interromper a operação do servidor.

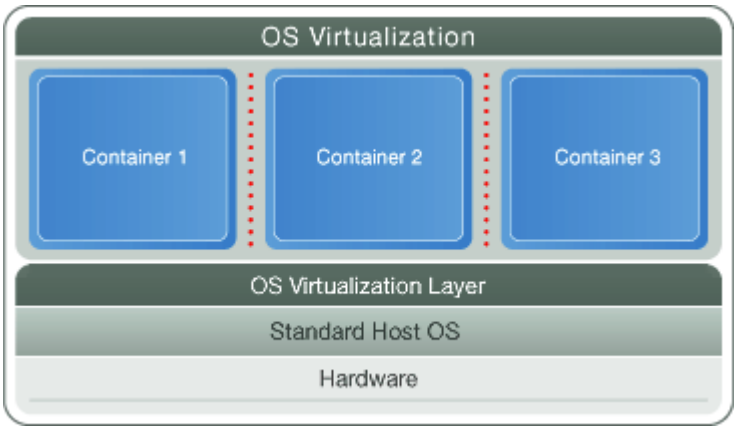

Figura 3 – Ilustração da arquitetura de virtualização no nível de sistema operacional (PARALLELS VIRTUOZZO, 2011).

O principal fator que favorece o desempenho da virtualização no nível de sistema operacional é o fato de cada instância utilizar a mesma interface de chamadas de sistema para requisitar serviços do *kernel* do sistema hospedeiro, eliminando a necessidade de uma camada intermediária para emular ou tratar as requisições, diferindo do que ocorre nas virtualizações completas e parciais (WIKIPEDIA; Operating System-Level Virtualization, 2011).

Diferentemente das demais técnicas, a virtualização no nível de sistema operacional não oferece a mesma flexibilidade, restringindo a utilização de sistemas operacionais diferentes do hospedeiro. Em um sistema hospedeiro Linux, por exemplo, é possível utilizar diferentes distribuições, sendo estas totalmente compatíveis com o *kernel*, porém máquinas virtuais de outros sistemas, como Windows, não são capazes de serem hospedadas. Pode parecer uma grande barreira para sua utilização, mas em diversas situações ela se torna vantajosa, principalmente em ambientes homogêneos, que prezam pela cultura de utilizar softwares de código livre e também economizando em capacitação de profissionais em sistemas distintos.

Algumas implementações conhecidas que utilizam esta técnica são: OpenVZ, LXC (Linux Containers) e FreeBSD Jails.

2.1.4 Virtualização Assistida por Hardware

A virtualização assistida por hardware é uma abordagem de virtualização de plataformas que permite a virtualização completa de forma eficiente, utilizando o auxílio de recursos de hardware do sistema hospedeiro. A técnica foi inicialmente concebida pela IBM nos anos 70, que desenvolveu o *mainframe* IBM System/370 e o sistema operacional VM/370, primeiro a utilizar o conceito de combinar um software capaz de gerenciar o hardware para ambientes virtuais (FISHER-OGDEN, 2011). Posteriormente, a mesma técnica foi adicionada aos processadores da arquitetura x86 (Intel VT-x e AMD-V), em 2006.

Quando a arquitetura x86 foi concebida, o conceito de virtualização foi abordado sem muita consideração, pois se destinava a pequenos computadores, sofrendo influências pelo declínio dos *mainframes* e a ascensão das *workstations* (FISHER-OGDEN, 2011). Sendo assim, ela não cumpre com os requerimentos de virtualização apontados por Popek e Goldberg, descritos em um artigo escrito em 1973, que formaliza os requisitos de tornar virtualizáveis arquiteturas de terceira geração (POPEK; GOLDBERG, 1973). Mesmo que os requisitos sejam descritos apoiados por hipóteses simplificadoras, eles representam uma forma conveniente de delimitar se uma determinada arquitetura pode suportar virtualização de forma eficiente e ser capaz de fornecer orientações para a concepção de arquiteturas de computadores virtualizados.

Os requisitos propostos por Popek e Goldberg resumem-se em três aspectos principais:

- Equivalência: um programa em execução sobre um gerenciador de virtualização deve comportar-se de forma idêntica à execução direta em uma máquina real;
- Segurança: o gerenciador de máquinas virtuais deve ter controle total sobre os recursos virtualizados;

- Eficiência: uma fração estatisticamente dominante das instruções de máquina deve ser executada sem intervenção do gerenciador.

Segundo Fisher-ogden (2011), a dificuldade em cumprir estes requisitos na implementação de um gerenciador de virtualização assistida por hardware, foi a limitação encontrada na arquitetura *x86* que não permite interceptar algumas instruções privilegiadas.

Para compensar esta lacuna, duas técnicas de virtualização da arquitetura *x86* foram concebidas: a virtualização completa e a paravirtualização. Ambas são baseadas na concepção de um gerenciador (ou monitor) de máquinas virtuais intermediário, também conhecido como *hypervisor*.

O *hypervisor*, ou monitor de máquinas virtuais (VMM – *Virtual Machine Manager*) é uma das técnicas mais populares de virtualização que permite múltiplos sistemas operacionais (chamados *guests*) serem executados concorrentemente em um computador hospedeiro (WIKIPEDIA; Hypervisor, 2011). Sua concepção é baseada na metodologia que combina hardware e software, sendo que sua nomenclatura deriva do fato de estar acima do *supervisor*, o conhecido *kernel* dos sistemas operacionais modernos. O *hypervisor* provê aos sistemas operacionais *guest* uma plataforma de operação virtual, gerenciando seus recursos e execução.

Para gerenciar as camadas de virtualizações, o *hypervisor* fica responsável por dividir o hardware entre os sistemas hospedados, controlando o hardware subjacente e permitindo que seja usado por muitos sistemas virtualizados ao mesmo tempo, dando a eles a ilusão de que estão sendo executados em um hardware privativo. Ele abstrai os recursos físicos disfarçando-os como correspondentes virtuais discretos, que podem ser alocadas para uso por máquinas virtuais individuais, garantindo, através do encapsulamento, que cada um trate sua porção de hardware como se fosse real.

Os *hypervisors* têm como funções principais: definir o ambiente de máquinas virtuais; alterar o modo de execução (privilegiado, convidado, etc.); emular as instruções e escalonar a CPU para uso das VM's; gerenciar o acesso a blocos de memória e disco; e intermediar as chamadas por acesso a dispositivos (USB, rede, drives, etc.). Os monitores devem possuir três características fundamentais:

- Eficiência de processamento, onde a participação do processador real deve executar a maioria das instruções sem intervenções, deixando ao monitor o tratamento das instruções que não puderem ser tratadas.
- Integridade, onde os recursos de hardware devem ser alocados explicitamente.
- Equivalência à máquina real, onde o monitor deve oferecer um comportamento de execução semelhante ao da máquina real.

Além destas características, o *hypervisor* deve levar em consideração características como a compatibilidade, que é importante para a execução do legado de software, alto desempenho para a execução dos sistemas virtualizados e também a simplicidade, evitando falhas no monitor que possam repercutir para todas as máquinas virtualizadas. Também devem ter mobilidade, permitindo que seus sistemas hospedados sejam deslocados de uma máquina física para outra, sem interrupções, chamado de "*live migration*".

Na virtualização completa, o *hypervisor* é responsável por simular um hardware completo para executar sistemas operacionais *guests* sem qualquer modificação anterior. Já na paravirtualização, apenas parte das operações é simulada pelo *hypervisor*, sendo o restante delas gerenciada pelo próprio hardware.

2.1.4.1 Virtualização Completa

A virtualização completa, ou nativa, refere-se à técnica de virtualização que se baseia em prover uma espécie de ambiente virtual responsável por fazer uma simulação completa do hardware subjacente (*GOLDEN & SCHEFFY,* 2008). Por sua essência, faz com que cada particularidade do hardware do sistema seja refletida para as máquinas virtuais, interceptando as operações de cada sistema operacional virtualizado (ex.: instruções de I/O, alterações de estado, acesso a memória, CPU), emulando as operações em software e retornando as instruções consistentes para cada VM, idênticas a um hardware físico (*GOLDEN & SCHEFFY,* 2008).

A principal vantagem da virtualização completa é a possiblidade de virtualizar de sistemas operacionais não modificados, igualmente na emulação. Contudo, neste caso o sistema virtualizado deve suportar o hardware da máquina onde a virtualização completa for empregada (REIS, 2009).

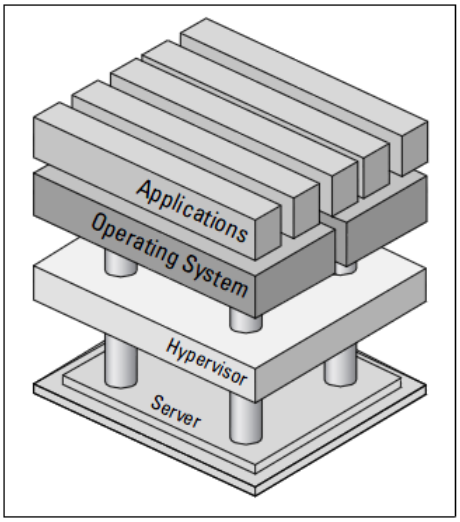

Figura 4 –O *hypervisor* intercepta as operações do hardware, emulando as operações e repassando para as máquinas virtuais.

Como desvantagem, a técnica de interceptar e emular as instruções pode reduzir drasticamente o desempenho do sistema em ambientes com muita atividade. A camada criada pelo software de virtualização requer que todas as operações passem por ela, e como elas

consomem tempo, o desempenho é afetado em situações de sobrecarga (*GOLDEN & SCHEFFY,* 2008).

2.1.4.2 Paravirtualização

A paravirtualização, também conhecida como virtualização parcial, é uma técnica que elimina parcialmente a interceptação das operações (como é feito na virtualização completa), delegando ao sistema operacional hospedeiro as funções de gerenciar os recursos de hardware tanto para si mesmo quanto para as máquinas virtuais hóspedes. Nesta abordagem, de acordo com Fisher-Odgen (2001), modificações no código-fonte do sistema hospedeiro são aplicadas para contornar os desafios da virtualização de arquitetura *x86*, onde a tradução binária das instruções problemáticas ajuda a filtrá-las e contorná-las.

A técnica permite que o sistema virtualizado acesse recursos do hardware diretamente, porém, com restrições que são administradas pelo sistema hospedeiro. Ela requer que o sistema hospedeiro seja modificado de acordo com a *API* provida pelo sistema paravirtualizador, reduzindo a quantidade e o tempo de operações que o sistema virtualizado executaria em um ambiente de virtualização completa. Como benefício, oferece um maior desempenho que a virtualização completa, otimizando a utilização dos recursos e reduzindo o *overhead* (SILVA, 2007).

Pelas vantagens oferecidas com referência ao desempenho, a técnica de paravirtualização foi largamente aplicada em *frameworks* de virtualização como o Xen, KVM e VMWare na implementação de seus *hypervisors* como forma de orientar os sistemas virtualizados a apontar chamadas de sistema para ele, estreitando a comunicação com o *hypervisor* e cooperando na obtenção de um melhor desempenho.

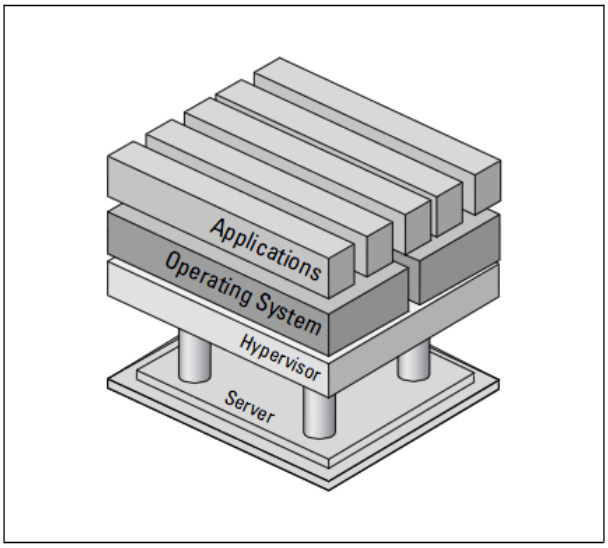

Figura 5 –Técnica de paravirtualização, ilustrando o estreitamento da entre o *hypervisor* e o sistema.

2.1.5 Visão Geral sobre as Técnicas de Virtualização

Como visto anteriormente, cada técnica de virtualização possui suas características, o que nos permite avaliar a melhor utilização para cada uma de acordo com nossas necessidades. Na tabela abaixo estão resumidas as vantagens e desvantagens das técnicas apresentadas:

Tipo	Descrição	Vantagens	Desvantagens
Virtualização Por Software	Oferece uma estrutura de virtualização simulada, permitindo sistemas e aplicativos incompatíveis com o sistema nativo serem executados.	Permite execução diferentes versões de múltiplos sistemas operacionais.	Baixo desempenho, consumindo muitos recursos dependendo a atividade em operação.
Virtualização Completa	O hypervisor fornece uma máquina virtual completa da mesma arquitetura hospedeira, permitindo que máquinas virtuais sem modificações executem isoladamente.	Permite execução diferentes versões de múltiplos sistemas operacionais.	O sistema não sabe que está virtualizado, podendo afetar o desempenho do sistema como um todo, particularmente em aplicativos com intensas operações E/S.
Paravirtualização	O hypervisor fornece uma máquina virtual completa, porém, especializada para cada hóspede, que precisa ser modificado para execução em isolamento.	Desempenho elevado, permitindo cooperação do SO com o hypervisor, melhorando a E/S e a alocação de recursos.	Exige alterações nos sistemas operacionais virtualizados para que possam utilizar as hipercamadas. O impacto é significativo em sistemas operacionais antigos e fechados, sem extensões implementadas para a paravirtualização.
Virtualização no Nível de SO	Um único SO é modificado para permitir vários processos servidores de	Melhor desempenho de todas as técnicas, com uma camada de virtualização leve e rápida,	Restringe o uso a sistemas operacionais compatíveis com o sistema hospedeiro.

	espaço de usuário, executados isoladamente, porém, na mesma plataforma de hardware.	com gerenciamento dinâmico de recursos.	Fraco isolamento, onde o uso de recursos de uma pode influenciar nas outras caso não sejam adotadas políticas de gerenciamento.

Tabela 1 – Resumo das técnicas de virtualização descritas, apresentando suas vantagens e desvantagens.

2.2 Computação Utilitária

O conceito de computação utilitária refere-se a um agregado de recursos computacionais, como processamento, armazenamento de dados e serviços, oferecidos como um modelo de negócio da mesma forma que os tradicionais serviços públicos (telefonia, eletricidade, etc.), ou seja, sob demanda (WIKIPEDIA; Utility Computing, 2011).

Este modelo de negócio foi previsto e teve seus primeiros passos por volta dos anos 60, onde grandes companhias como a IBM proviam serviços de computação utilitária para bancos e grandes corporações com seus enormes *mainframes*, restringindo uma maior escala de uso motivada pelo alto custo e pela escassez de provisionamento de recursos computacionais na época. Possuir um *mainframe* era privilégio de poucos, pois não compensava pelo alto custo. As poucas companhias que detinham e dominavam a tecnologia, ofereciam contratos de utilização para terceiros, sendo considerado o início da computação utilitária.

Com o advento dos microcomputadores a partir dos anos 70, muitas empresas finalmente puderam ter acesso a essa tecnologia, criando seus próprios *datacenters* e quase extinguindo o conceito, sendo ressurgido nos anos 90 com a popularização das redes de computadores e a Internet. Neste período, grandes companhias começaram a abrir seus *datacenters* para o mundo, oferecendo serviços de computação a outras empresas, encorajadas pelas vantagens oferecidas, principalmente econômicas.

Foi uma questão de tempo até a computação estar tecnologicamente madura o suficiente para voltar à atenção para o desenvolvimento de serviços computacionais com o mesmo conceito,

porém agora, sendo providos globalmente graças aos avanços das telecomunicações. Esta evolução gerou novas tecnologias para computação utilitária, sendo englobados em um paradigma identificado como computação em nuvem.

2.3 Computação em Nuvem

Computação em nuvem – ou *cloud computing* – é considerada como o paradigma atual da computação moderna por fazer uma analogia à computação em grade, também chamada *grid computing*, somado com a computação utilitária. A computação em grade é um modelo computacional capaz de alcançar altas taxas de processamento ao dividir as tarefas entre diversas máquinas, podendo ser em redes geograficamente distribuídas ou redes de longa distância, que juntas formam uma máquina virtual (WIKIPEDIA; Grid Computing, 2011). No caso, a grade em questão é a Internet, que abriga incalculáveis recursos computacionais em oferta para os usuários.

O termo nuvem já vem sendo utilizado como metáfora para referenciar as grandes redes de telecomunicações desde os anos 90, identificadas com uma ilustração em diagramas de redes, mapas e esquemas técnicos como forma de abstrair os recursos que o meio em questão representa, generalizando-os (WIKIPEDIA; Cloud Computing, 2011). Naturalmente, a mesma ilustração foi sendo utilizada para identificar a Internet, fazendo um paralelo das redes de telecomunicações com as redes de computadores, tornando-se símbolo e expressão da moda no mundo computacional quando o assunto é de serviços de computação nela oferecidos.

Em um sistema computacional disponível na Internet, a partir de um computador ou dispositivo conectado a ela, é possível ter acesso a diversos serviços, arquivos, informações e programas na forma de um sistema integrado, independente da plataforma, acessíveis a qualquer hora. O processamento de informações, acesso e armazenamento é realizado remotamente, muitas vezes sem a necessidade de o usuário instalar algum software para sua utilização.

A computação em nuvem descreve toda uma nova maneira de suplementar, consumir e oferecer serviços de tecnologia de informação via protocolos de Internet, onde geralmente envolve o provisionamento de recursos dinamicamente escaláveis e virtualizados. Na base de sua estrutura está o conceito de convergência de infraestruturas e compartilhamento de serviços, que habilita grandes instituições a oferecerem aplicações e sistemas ágeis com gerenciamento centralizado e menor manutenção, permitindo aos administradores destes sistemas alocarem recursos de forma dinâmica, conforme a demanda exigida.

2.3.1 Origens da Computação em Nuvem

As primeiras referências a um conceito de computação em nuvem foram descritas em meados dos anos 60, quando o cientista da computação John McCarthy sugeriu que "*a computação poderá algum dia ser organizada como uma utilidade pública*", onde o compartilhamento de recursos computacionais pode conduzir um futuro onde poder computacional e aplicações específicas poderão ser oferecidas na forma de produtos ou serviços, como eletricidade e água (WIKIPEDIA; John McCarthy, 2011). A idéia da computação como uma utilidade pública foi bastante popular na época, mas praticamente desapareceu em meados dos anos 70 ao tornar-se claro que as tecnologias disponíveis não ofereciam condições de consolidá-la. No entanto, desde o início do século 21, o modelo computacional ressurgiu em novas formas, basicamente com os provedores de serviços de aplicação, computação em grade e computação utilitária.

Na década de 90, no mesmo período do advento da Internet e das grandes redes de telecomunicações, importantes serviços de computação utilitária com o conceito de oferta sob demanda (*pay-per-use*) começaram a embrionar, desenvolvidos por companhias como a HP, IBM, Microsoft, Yahoo, Sun, Bell Labs e Google. Neste meio, surgiram os primeiros *web services*,

soluções de integração de sistemas e aplicações que se comunicam de forma transparente com outras através das redes de telecomunicações (WIKIPEDIA; Utility Computing, 2011).

Em 1997, Ramnath K. Chellappa, PhD, estabeleceu o que para muitos é a primeira definição acadêmica para computação em nuvem, que pode ser descrita como "*um paradigma computacional onde os limites da computação serão definidos por fronteiras econômicas ao invés de limites técnicos*". Ele explica que o termo surgiu como referência à abstração de quais seriam os limites e onde as coisas realmente iriam acontecer, fazendo uma analogia com a nuvem em razão da indefinição destes meios (IT COMPASS, 2010).

Uma das grandes contribuintes ao desenvolvimento das tecnologias de computação em nuvem foi, sem dúvidas, a Amazon. Com um grande parque computacional para operar seus negócios, a empresa observou que menos de 10% da capacidade de processamento era utilizada, sendo o restante usado apenas em situações de maior fluxo. Inspirada pela emergente arquitetura de computação em nuvem, que promovia em sua essência uma utilização mais eficiente dos recursos, a Amazon iniciou um esforço de pesquisa e desenvolvimento de serviços computacionais a serem oferecidos para clientes externos, lançando em 2006 a Amazon Web Services, mais precisamente o conhecido Amazon EC2 (*Elastic Compute Cloud*). Apesar de ainda experimental, o serviço atraiu imediatamente a atenção de programadores e *entrepreneurs* do mundo todo, permitindo que experimentassem no serviço as suas idéias e criações. Em 2007, o serviço entra efetivamente no ar, oferecendo recursos de computação em nuvem cobrados de acordo com a utilização (*pay-per-use*), sendo um dos primeiros serviços consolidados do gênero que colaborou fortemente com o início de uma nova tendência na computação moderna.

2.3.2 Características Gerais

A computação em nuvem oferece um conjunto de recursos computacionais – processamento, softwares, armazenamento, acesso a dados – que podem ser considerados tanto as aplicações entregues como serviços através da Internet quanto os sistemas de software nos *datacenters* que oferecem estes serviços. Ela não requer que os usuários finais tenham conhecimento sobre a localização física e a configuração do sistema que provê os serviços, sendo um conjunto de sistemas computacionais logicamente ou virtualmente dispostos formando uma nuvem, acessíveis aos usuários através da Internet (WIKIPEDIA; Cloud Computing, 2011). Os provedores de computação em nuvem oferecem aplicações pela Internet acessadas por navegadores ou softwares em computadores e dispositivos móveis, enquanto os softwares gerenciadores e os dados são armazenados em servidores remotos.

A computação em nuvem é uma evolução da computação utilitária, onde explora de forma mais aprofundada os conceitos de infraestruturas convergentes e de serviços compartilhados. Na computação utilitária tradicional, os serviços computacionais como armazenamento, processamento e largura de banda são oferecidos na forma de produtos, disponibilizados por provedores especializados que alugam seus recursos a terceiros, eliminando preocupações com escalabilidade e disponibilidade. Na computação em nuvem, o propósito maior é utilizar os mesmos recursos, porém de forma conveniente e sob demanda, ampliando o aproveitamento através de uma gerência facilitada com provisionamento de forma dinâmica (SOUZA; MOREIRA; MACHADO, 2009).

O Instituto Nacional de Padrões e Tecnologia (NIST, do inglês "*National Institute of Standards and Technology*") apresenta a seguinte definição para computação em nuvem: "*Computação em nuvem é um modelo que possibilita acesso de modo conveniente, sob demanda e onipresente a um conjunto de recursos computacionais configuráveis que podem ser rapidamente*

adquiridos e liberados com mínimo esforço gerencial ou interação com o provedor de serviços" (MELL; GRANCE, 2009).

Este modelo de computação em nuvem sustenta-se em três pilares básicos (SOUZA; MOREIRA; MACHADO, 2009):

- Reduzir custos na aquisição e composição da infraestrutura requerida, podendo ser composta sob demanda, com recursos heterogêneos;
- Flexibilidade na adição e substituição de recursos computacionais, escaláveis tanto em nível de hardware quanto de software;
- Prover uma abstração dos meios envolvidos, oferecendo facilidades de acesso aos usuários destes serviços.

2.3.3 Características Essenciais

Mell e Grance (2009) classificaram em cinco as características essenciais que contextualizam as soluções de computação em nuvem, formalizando o paradigma:

- Autoserviço sob demanda – O usuário pode adquirir unilateralmente recursos computacionais como tempo de processamento e armazenamento na rede, de forma que lhe for conveniente e sem depender de recursos humanos na interação com o prestador de serviço;
- Amplo acesso de rede – Os recursos são disponibilizados pela rede, acessíveis através de meios padronizados e de plataformas computacionais heterogêneas;
- Alocação de Recursos – Os provedores de recursos computacionais devem ser provisionados de forma a servir múltiplos usuários, utilizando diferentes meios físicos e virtuais ajustáveis de forma dinâmica e de acordo com a demanda. A localização física e

quantidade dos recursos computacionais é abstraída para o usuário, sendo independente de seu conhecimento;

- Elasticidade – Os recursos devem ser rapidamente provisionados e liberados, em alguns casos de forma automática, buscando suprir demandas de maior ou menor fluxo. Para o usuário, estes recursos devem transparecer de forma ilimitada, podendo ser adquiridos em qualquer quantia a qualquer momento;
- Mensuração do serviço – Sistemas de computação em nuvem devem automaticamente controlar e otimizar o provisionamento de recursos por meios de medição do uso, com níveis de abstração apropriadas de acordo com o tipo de serviço. A utilização pode ser monitorada, controlada e exibida para os usuários, informando sobre a utilização tanto para o provedor quanto para o usuário.

Dentro destas características, a disponibilização de serviços de computação em nuvem é classificada em três modelos. A abrangência de implantação em que estes modelos são disponibilizados que é classificada em quatro técnicas de implantação.

2.3.4 Modelos de Serviço

O ambiente de computação em nuvem é composto por três modelos de serviços, definindo um padrão de arquitetura para as soluções oferecidas, classificados de acordo com o público alvo. Estas três definições foram englobadas no paradigma a partir de referências a modelos já existentes, consolidados em estruturas de computação utilitária, mas que diferem dos conceitos de computação em nuvem. Os modelos de serviço são classificados como:

- Software como Serviço (SaaS – do inglês *Software as a Service*) – São sistemas ou softwares de serviços disponibilizados na da Internet, mantidos e desenvolvidos por prestadoras de serviços, acessíveis através de dispositivos remotos, sendo estes normalmente um navegador *web* ou um aplicativo. Os usuários não tem acesso à infraestrutura por trás do serviço, que inclui rede, servidores, sistemas operacionais, armazenamento ou mesmo características da aplicação, salvo algumas exceções específicas. Essa abstração da camada subjacente favorece os desenvolvedores, permitindo que se concentrem apenas na inovação e criação de novas aplicações, ignorando a infraestrutura. Softwares como serviço geralmente são acessíveis de qualquer lugar a qualquer momento, permitindo uma maior integração e incorporação de recursos. Favorece também o aprimoramento e atualização destas aplicações, já que independem do meio físico que o usuário utiliza, permitindo manutenções transparentes.
- Plataforma como Serviço (PaaS – do inglês *Platform as a Service*) – São infraestruturas de alto nível de integração oferecidas para implementar, desenvolver e testar aplicações em nuvem. Estas geralmente são oferecidas pelo provedor na forma de *APIs*, sendo de linguagens de programação, bibliotecas de arquivos, serviços de bancos de dados e ferramentas, proporcionando agilidade ao desenvolvimento e reduzindo a complexidade. O usuário não administra ou controla a infraestrutura subjacente, mas têm controle sobre sistemas implantados, permitindo configurá-los para atender suas necessidades.
- Infraestrutura como Serviço (IaaS – do inglês *Infraestructure as a Service*) – São infraestruturas em nuvem que disponibilizam ao usuário recursos de processamento, armazenamento, rede e demais requisitos fundamentais para o cliente, permitindo

execução de softwares arbitrários, sendo estes desde sistemas operacionais a aplicativos específicos. O cliente não administra a nuvem, mas tem acesso a um sistema operacional completo, com possibilidade de demandar recursos e configurar alguns componentes de rede e segurança (*firewall*). De forma simplificada, a IaaS faz referência a infraestruturas baseadas em virtualização, com mecanismos de que permitem escalar recursos dinamicamente, aumentando ou diminuindo de acordo com as necessidades das instâncias.

A ilustração a seguir (figura 6) demonstra de forma simplificada a inter-relação das metodologias descritas, identificando os papéis dos usuários envolvidos nas soluções de computação em nuvem.

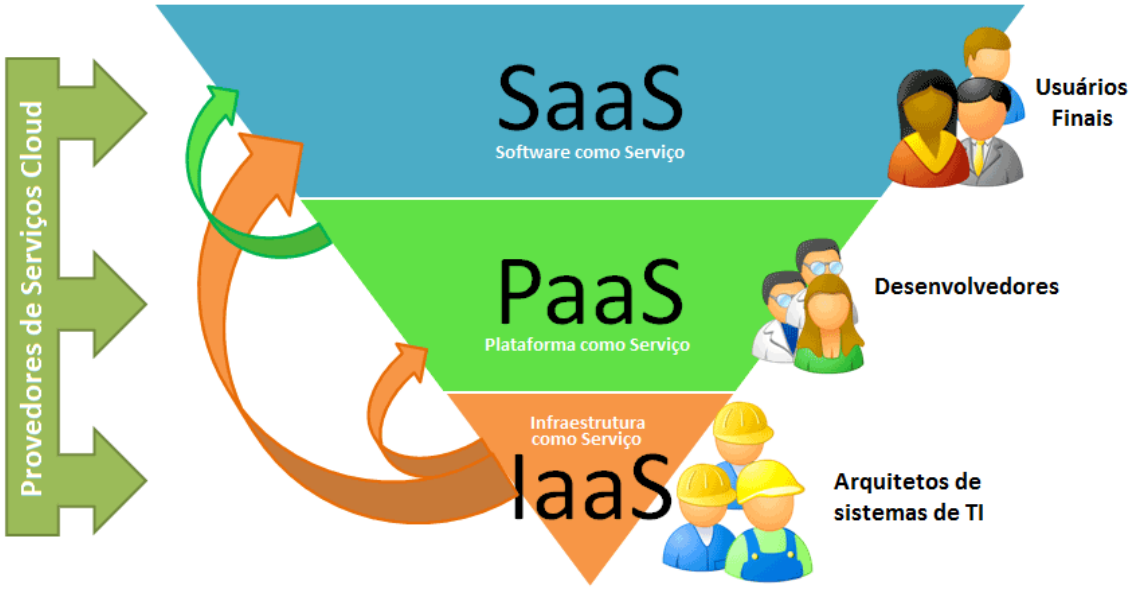

Figura 6 – Modelos de serviço de computação em nuvem e seus usuários. Adaptado de (SCHULLER, 2008).

Em uma forma simplificada, os SaaS atendem os usuários finais que procuram satisfazer aplicações de negócios como: gestão de relacionamento com clientes (CRM), sistemas integrados de gestão empresarial (ERP), sistemas de gerenciamento de conteúdo (CMS), processos em geral e

também entretenimento. As PaaS, no entanto, buscam prover um ambiente para desenvolvedores de aplicativos e empresas de software, fornecendo-lhes ferramentas para construir aplicações SaaS, enquanto as IaaS buscam oferecer recursos de baixo nível para administradores de sistemas e redes que precisam suprir demandas de PaaS e SaaS particulares (SCHULLER, 2008) A figura 7 ilustra uma alusão a nuvem, serviços oferecidos e meios de acesso.

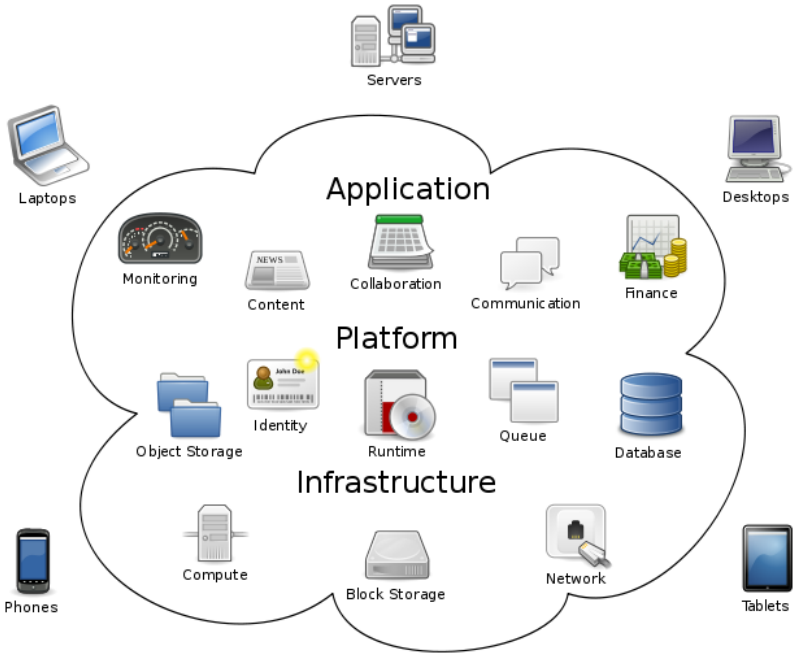

Figura 7– Ilustração do conceito de Computação em Nuvem, referenciando os serviços oferecidos em seu contexto. Adaptado de (WIKIPEDIA; Cloud Computing, 2011).

Serviços providos nestes modelos geralmente são oferecidos por *datacenters* especializados, que são identificados como uma Nuvem. Quando esta disponibiliza serviços sob a forma de demanda, cobrando pelos recursos que são utilizados (*pay-per-use*), é identificada como uma nuvem pública (do inglês *Public Cloud*). Diversas empresas prestam serviços de computação utilitária de forma pública, sendo destacadas a Amazon Web Services, Windows Azure e Google AppEngine (ARMBRUST et al., 2009). Além das nuvens públicas, existem outras três técnicas de implantação, classificadas em nuvens privadas, híbridas e comunitárias.

2.3.5 Técnicas de Implantação

As técnicas de implantação diferem basicamente pela acessibilidade e disponibilidade, variando de acordo com o processo de negócio, tipo de informação e nível de visão. A restrição de acesso é uma medida que muitas organizações e provedoras de serviços aplicam em seus ambientes computacionais, onde somente alguns usuários tem acesso aos serviços providos. De acordo com a NIST (2011), as técnicas são classificadas como:

- Nuvens Públicas – a infraestrutura é provisionada para uso geral, podendo ser fornecida por um provedor que vende o serviço ou disponibiliza de forma livre. Pode ser operada e gerenciada por uma empresa privada, instituição acadêmica ou organização governamental. Nuvens públicas proporcionam economias de escalabilidade, pois permite o uso compartilhado de recursos, porém, oferece limitações de customização e preocupações com a segurança dos dados, cumprimento de acordos de níveis de serviço (SLAs) e políticas de acesso, já que muitas vezes a localização física é desconhecida e falhas do serviço podem ser extremamente prejudiciais.
- Nuvens Privadas – a infraestrutura é fornecida para uso exclusivo de uma única organização, compreendendo vários consumidores ou unidades de negócio. Ela pode ser operada e gerenciada pela própria organização ou por terceiros, veiculada ou não na Internet. O objetivo das nuvens privadas é prover aos usuários da organização uma infraestrutura ágil e flexível, capaz de oferecer serviços dentro do seu domínio administrativo. Oferece maiores garantias de segurança e estabilidade em detrimento de sua natureza privada.
- Nuvens Comunitárias – a infraestrutura é oferecida para uso exclusivo de uma comunidade específica de usuários que compartilha interesses semelhantes, além de regras de

utilização. É semelhante às nuvens privadas na disponibilização, podendo ser operada e gerenciada por uma ou mais organizações na comunidade, ou também providas por terceiros.

- Nuvens Híbridas – são compostas de duas ou mais infraestrutura distintas (pública, privada ou comunitária) mantidas como entidades únicas, porém unidas através de tecnologias padronizadas ou proprietárias que habilitam a portabilidade de dados e aplicações, geralmente utilizando esta técnica como forma de balancear ou transferir o uso para outras nuvens.

As técnicas de implementação e utilização de infraestruturas em nuvem não devem ser analisadas apenas pelo contexto de serem internas ou externas com relação à localização física dos seus ativos, recursos e informações, mas também por quem são consumidos e por quem são os responsáveis por sua administração, segurança e conformidade com as políticas e normas (CLOUD SECURITY ALLIANCE, 2011).

2.3.6 Vantagens e Desvantagens

Estruturas de computação em nuvem, quando aplicadas de forma correta, podem oferecer diversas vantagens aos usuários que vão além da possibilidade de utilizar aplicações que não requerem instalação no computador. Algumas delas podem ser exemplificadas:

- Mínima ou nenhuma preocupação com o sistema operacional ou hardware do computador pessoal para utilizar os recursos de nuvens, da mesma forma com relação a atualizações ou manutenções do software utilizado na nuvem;
- Portabilidade de informações, facilitando o trabalho corporativo e o compartilhamento de arquivos, sendo acessíveis em qualquer lugar e necessitando apenas acesso a Internet.

Sistemas e softwares não mais se restringem ao ambiente local de computação, permitindo também que sejam acessados nos mais variados dispositivos;

- Elasticidade e flexibilidade na utilização dos recursos, tanto na utilização quanto na contratação dos serviços, pagando somente pelo que for utilizado. Esta vantagem estende-se tanto para ambientes privados como públicos, onde os usuários podem alocar equipamentos ou recursos caso precise redimensionar, inclusive contratar serviços de forma a suprir demandas;
- Economia na implantação de infraestruturas baseadas em computação em nuvem, geralmente mais enxutas que as soluções tradicionais por otimizar o uso do hardware, promovendo menor consumo de energia, menor utilização de espaço físico e refrigeração, contribuindo com o uso racional e sustentável;
- Economia na contratação de serviços baseados na nuvem, reduzindo ou eliminando despesas com manutenção de infraestruturas locais e TI, assim como aquisição e atualização de softwares.

Embora as vantagens sejam atrativas por promoverem principalmente a economia em recursos computacionais, existem diversas preocupações, principalmente a nível técnico, que devem ser consideradas, além de uma avaliação real das necessidades ao empregar o uso desta tecnologia. Algumas destas podem ser listadas, como:

- Disponibilidade de Acesso – o acesso aos recursos computacionais em nuvem depende da Internet, salvo algumas exceções em que as nuvens abrangem redes locais ou geograficamente próximas, atendidas por infraestruturas de rede preferencialmente ágeis. Assim como a disponibilidade é importante em todos os casos, um fator crucial para decisões de utilização é a largura de banda, muitas vezes provida de forma limitada e

irregular. Para situações que demandam transferências volumosas de dados, as vantagens promovidas podem não surtir os resultados esperados.

- Segurança da Informação – em nuvens públicas, a segurança da informação é uma das maiores preocupações dos usuários dos serviços, pois as garantias de integridade, confiabilidade e confidencialidade dos dados nem sempre são atestada pelos provedores. O desconforto com as questões de segurança impedem uma maior adoção pelos setores público e privado, principalmente pelo envolvimento de terceiros na gestão dos serviços. Políticas de segurança vêm ganhando espaço, oferecendo segurança através da criptografia e utilização de chaves públicas, porém ainda dependem de uma maior homologação e padronização entre as prestadoras de serviços.
- Padronização – geralmente, as prestadoras de serviços possuem APIs padronizadas, dentro de licenças *Creative Commons* para integração e desenvolvimento de aplicações, da mesma forma que muitas ainda insistem em utilizar tecnologias restritas e proprietárias, dificultando a interoperabilidade. Apesar das APIs proprietárias prometerem vantagens atraentes aos consumidores, com justificativas de prover maior segurança, políticas contratuais e suporte, elas dificultam a migração entre os serviços, tornando os usuários dependentes destas prestadoras. Existem diversas entidades que apresentam conceitos de padronização, adotados inclusive por grandes companhias, mas ainda demandam esforços para convencer uma maior adoção.

Entre estas desvantagens listadas, muitas delas podem ser contornadas ou minimizadas através de cláusulas contratuais entre as organizações e as prestadoras de serviços em nuvem. Nestes contratos, entram os acordos de níveis de serviço (SLAs), regulamentações que os provedores oferecem como garantias de disponibilidade e cumprimento, buscando atrair clientes e também elevar a confiança na prestação de seus serviços.

2.3.7 Acordos de Níveis de Serviço

Os acordos de níveis de serviço (SLAs – do inglês *Service Level Agreement*) são instrumentos de gestão das expectativas do cliente. A meta desses acordos é definir um conjunto de métricas para oferecer garantias de qualidade e quantidade dos serviços prestados, buscando atender a demanda dos clientes com um entendimento claro do conjunto de compromissos. Estes acordos servem como uma ferramenta formalizada de prevenção de conflitos, sendo atualizados e revisados constantemente, buscando garantir que ambas as partes usarão os mesmos critérios para avaliar a qualidade do serviço (GOMES; FALBO; MENEZES, 2005).

A elaboração de um contrato SLA considera duas fases bem diferenciadas: a negociação do contrato, onde as partes estabelecem as garantias de prestação do serviço e preços, e o acompanhamento da sua realização em tempo real. A gestão do SLA engloba a definição do contrato nos quatro passos seguintes: elaboração de esquema básico com os parâmetros de qualidade de serviço (QoS – do inglês *Quality of Service*); negociação de acordos e valores no contrato SLA; acompanhamento do contrato SLA e aplicação do acordo com as regras estabelecidas (IT-TUDE, 2011).

O ponto principal é a construção de uma nova camada sobre a plataforma de serviços, capaz de criar um mecanismo de negociação entre provedores e consumidores procurando delimitar as capacidades oferecidas e o cumprimento das demandas do cliente. Esta camada deve ser capaz de atender requerimentos como (IT-TUDE, 2011):

- Otimização na seleção de recursos – o provisionamento deve ser processado de forma a buscar as melhores fontes dentre as disponíveis para atender a demanda;

- Monitoramento em tempo real – capaz de monitorar as execuções instanciadas no contrato, identificando ameaças de violações de acordo e realizando ações de remediação, visando garantir ou minimizar as consequências em caso de descumprimento;
- Negociação de condições – as duas partes devem concordar com todas as condições e termos do acordo antes de firmarem, baseadas nas necessidades do cliente e nas capacidades do provedor em atendê-las;
- Divulgação e descoberta – tanto o cliente quanto o provedor necessitam de um contato "virtual" para se introduzirem. O provedor deve divulgar seus serviços em mecanismos de publicação baseados em seus modelos de SLA, enquanto o cliente deve buscar e encontrar quem possa suprir suas necessidades.
- *Templates* de Acordos de Níveis de Serviço – os provedores podem ter dificuldades em descrever e delimitar suas capacidades de recursos de forma clara e segura, incluindo aspectos legais e de utilização. A utilização de modelos de serviços (*templates*) colabora com a identificação e avaliação das capacidades oferecidas para os clientes, facilitando o entendimento e criação de acordos;
- Capacidade de contabilizar – provedores de serviços devem sumarizar os recursos utilizados pelos usuários para cobrá-los, comparando com as condições especificadas no acordo. O monitoramento deve ser constantemente analisado, contabilizado e divulgado ao cliente, incluindo possibilidades de penalizações ou violações de acordo.

Da forma com que a maioria dos serviços de computação em nuvem são prestados, compartilhando recursos e infraestrutura com diversos clientes, acordos de nível de serviço se

tornam essenciais para oferecer garantias de disponibilidade e uma maior segurança a usuários em missões críticas, promovendo e agregando vantagens a sua utilização.

2.4 Trabalhos Relacionados

Diversas instituições de ensino fazem uso das vantagens de implementações de virtualização e computação em nuvem, atendendo seus parques computacionais, *clusters*, laboratórios de pesquisa e sistemas internos com objetivo de otimizar o uso dos recursos computacionais e resolver problemas de sobrecarga dos servidores. Os benefícios são inúmeros, partindo desde a redução da subutilização dos equipamentos, consumo de energia, manutenção, refrigeração e espaço físico, como também um maior controle por parte dos administradores dos sistemas na gestão e no provimento de novas demandas. Apesar de a tecnologia estar consolidada, poucas instituições promovem uma maior utilização por parte da comunidade acadêmica, sendo nestes casos oferecida por iniciativa de professores ou pesquisadores, geralmente disponibilizadas em âmbito limitado ou experimental. A idéia de prover um ambiente de computação virtualizado para ser utilizado por professores e alunos não é recente, sendo citada em diversos trabalhos sobre virtualização de servidores que sugerem a perspectiva de implantar recursos de virtualização para promover o ensino, bem como em experimentos onde que a virtualização foi peça fundamental ao agregar benefícios ao processo de aprendizado e capacitação dos alunos.

Silva (2007) descreve que a virtualização de servidores oferece excelentes vantagens se empregada na área de ensino, oferecendo mecanismos para otimizar o uso de equipamentos e reduzir a manutenção. Também sugere duas abordagens de aplicação, como: utilizar um servidor central para armazenar máquinas virtuais dos alunos, acessíveis através de conexões remotas em

máquinas clientes de pequeno porte, permitindo recuperação rápida do sistema em casos de comprometimento do sistema pelas mais variadas situações; e a aplicação de sistemas virtuais com o conteúdo dos cursos, sendo disponibilizados aos alunos para utilização e modificação, restritos ao espaço virtual, permitindo no final de cada período restaurar rapidamente os sistemas a seu estado original.

Estas duas abordagens citam como principal vantagem o fato de permitirem a criação máquinas virtuais customizadas, instalando o sistema operacional e as ferramentas necessárias para cada perfil de disciplina ou curso, sendo disponibilizadas em um estado onde todo o sistema esteja pronto para uso e permitindo que os mesmos sejam restaurados em pouco tempo em casos de falhas, sem maiores transtornos aos alunos e professores. Em processos de aprendizagem e ensino, é comum ocorrerem danos ao sistema ou ferramentas instaladas que comprometem o ambiente como um todo, sendo um excelente recurso na remediação destas ocorrências.

Em um estudo voltado à preparação de gestores de sistemas de informação, Marshall (2011) aborda as experiências resultantes da utilização de sistemas de virtualização como plataforma para aplicação em dois cursos: Infraestrutura de TI, que aborda arquitetura de sistemas e redes de comunicações com uma abrangência sobre as soluções de TI em um contexto organizacional; e Arquiteturas Corporativas, que se concentra em questões a nível organizacional relacionadas ao planejamento, elaboração, projeção e implementação de soluções baseadas em infraestruturas de TI. Os dois cursos utilizam ambientes computacionais que requerem privilégios de acesso administrativo, diferente dos demais cursos que compartilham o mesmo laboratório, oferecendo riscos à integridade dos sistemas. A virtualização possibilitou a oferta de ambientes seguros para a realização de experimentos, sem maiores riscos para as máquinas físicas, promovendo um cenário onde os estudantes podem praticar não apenas programação, mas

também desenvolver conhecimentos sobre modificações na *kernel* do sistema operacional, configurações de rede, atualização de recursos, entre outros.

No mesmo estudo, os autores inicialmente avaliaram técnicas de virtualização nas máquinas locais utilizando softwares de virtualização baseados na simuação de hardware instalados nos computadores dos laboratórios, onde os alunos salvavam as instâncias virtuais em dispositivos de armazenamento removíveis ou em servidores locais. Porém, estas atividades demandavam muito tempo das disciplinas ao baixar o ambiente para utilização e salvar os resultados após o término cada aula, resultando também sobrecargas na rede. Com a utilização de ambientes hospedados em um servidor principal, foi possível contornar esse problema, sendo necessário apenas efetuar o acesso à máquina remota para iniciar as atividades. Também minimizou o uso e dependência do *hardware* local para executar o ambiente de virtualização, pois as atividades se concentram no servidor, configurado com um sistema especializado de virtualização capaz de atender múltiplas instâncias de máquinas virtuais.

Marshall (2011) argumenta outras vantagens percebidas ao se utilizar um sistema de virtualização de servidores como ferramenta de ensino, principalmente ao tornar possível a utilização remota, permitindo aos alunos desenvolverem suas atividades fora da universidade e em horários compatíveis com suas rotinas, sem depender da disponibilidade dos laboratórios.

Em um estudo semelhante, Bower (2010) cita os desafios em promover o estudo em ambientes computacionais restritos, que obriga os alunos a utilizarem computadores pessoais para atividades que poderiam ser realizadas em laboratório. Os desafios apontados foram em questão ao armazenamento e utilização das máquinas virtuais, já que as estações dos laboratórios não permitiam acesso administrativo e não preservavam alterações ou novos softwares instalados

no sistema. As máquinas virtuais necessitavam ser salvas em dispositivos de armazenamento móveis para posterior utilização, com as mesmas dificuldades que Marshall (2011) apontou.

Muitos trabalhos abordam principalmente a utilização de ferramentas e soluções privadas, como no caso dos estudos descritos, onde os experimentos foram desenvolvidos utilizando o VMware e seus componentes. Também abordam a utilização de softwares de virtualização instalados nos computadores dos laboratórios, atestando que não é uma medida eficiente para remediar os problemas de limitações de uso, pois consomem muito tempo até estarem disponíveis. A utilização de um servidor centralizado para hospedar as máquinas virtuais mostrou-se uma solução favorável, tornando ágil a disponibilização dos ambientes e também reduzindo gargalos de desempenho e tráfego de rede nas instalações.

Estes estudos apontam excelentes resultados na aplicação de técnicas de virtualização voltada ao ensino, permitindo ampliar as experiências e cenários em diversas situações, com riscos menores para as infraestruturas de ensino e também capacitando futuros profissionais a utilizarem estes recursos, cada vez mais comuns nas organizações.

Os desafios para implantar, capacitar e adaptar os usuários a utilizarem estas ferramentas foram identificados nestes trabalhos, principalmente pela dificuldade em introduzir novos sistemas e cenários para os alunos, muitas vezes acostumados a ambientes gráficos e pouca interação com o funcionamento dos softwares e sistemas operacionais. Estes desafios não se concentram apenas nos usuários, mas também na disponibilização e preparação de serviços de virtualização para a comunidade, que demandam atenção redobrada nas questões de segurança e estabilidade.

A virtualização em universidades encontra-se em uma fase bastante consolidada, permitindo que novas aplicações sejam desenvolvidas para benefício da comunidade acadêmica, incentivando

o uso de tecnologias que aperfeiçoem o poder computacional e tornem os meios de aprendizado mais eficientes.

3. Tecnologias de Virtualização e Computação em Nuvem

A virtualização, praticamente esquecida após o advento dos microcomputadores e da arquitetura *x86*, ressurge fortemente em meados dos anos 90 graças aos esforços pela busca de otimizar os recursos computacionais. Atualmente, impulsionada pela computação em nuvem, estas duas tecnologias geraram um número sem fim de ferramentas, oportunidades e serviços que formaram uma nova definição para a computação moderna.

Muitas destas ferramentas se destacaram nos últimos anos, principalmente por facilitarem o gerenciamento de máquinas virtuais e oferecerem mecanismos rápidos de provisionamento de recursos, habilitando usuários e profissionais de TI a explorar melhor seus meios computacionais.

Dentro do enfoque deste trabalho foram escolhidas algumas ferramentas a serem abordadas, levando em consideração suas características, para posterior avaliação e adoção na homologação da proposta.

3.1 Eucalyptus

Inicialmente concebido como um projeto universitário no departamento ciências da computação da Universidade de Santa Bárbara, na Califórnia, EUA, o Eucalyptus, acrônimo para "*Elastic Utility Computing Architecture Linking Your Programs To Useful Systems*", ou em uma tradução livre, "Arquitetura de Computação Utilitária e Elástica para Vincular Seus Programas a Sistemas Úteis" é um conjunto de ferramentas que permite virtualizar todos os componentes de hardware como um único conjunto de recursos, permitindo a criação de nuvens locais privadas, sem necessitar alterações físicas na infraestrutura de TI ou introdução de hardware especializado.

Ela implementa uma nuvem privada IaaS (Infraestrutura como Serviço) acessível por uma *API* compatível os serviços EC2 e S3 da Amazon, estendendo suas funcionalidades e permitindo sua execução em forma híbrida, utilizando recursos de nuvens públicas (EUCALYPTUS, 2010).

A Canonical, empresa responsável pelo sistema operacional Ubuntu e seus derivados, vem acolhendo tecnologias de computação em nuvem como um importante segmento dos seus negócios, o *Ubuntu Enterprise Cloud Service*. A partir da versão 9.04, o sistema operacional Ubuntu trouxe junto em sua estrutura o suporte nativo para utilizar a ferramenta, incentivando a sua popularização.

Em sua versão de código aberto, o Eucalyptus traz as seguintes funcionalidades (EUCALYPTUS, 2010):

- Interface compatível com o Amazon Web Services (AWS);
- Clusterização flexível;
- Infraestrutura como Serviço (IaaS) de alta disponibilidade;
- Gerenciamento de rede, segurança de grupos, isolamento de tráfego;
- Elasticidade e capacidade de auto-serviço;
- Armazenamento abstrato baseada em balde (compatível com o Amazon S3);
- Armazenamento abstrato baseado em bloco (compatível com EBS);
- Suporte aos *hypervisors* Xen e KVM;
- Suporte a virtualização de sistemas operacionais Windows;
- Gerenciamento de cotas de uso;
- Gerenciamento de grupos de usuários e permissões de uso;

A versão *enterprise* (comercial) difere-se da versão de código aberto por oferecer suporte ao *hypervisor* VMWare e seus demais recursos, além de integração direta com SAN, tecnologia privada de armazenamento de dados em rede.

Dentre as características que fizeram o Eucalyptus tornar-se uma das ferramentas mais utilizadas na concepção de nuvens privadas, podemos citar (EUCALYPTUS, 2010):

- Ferramenta de código aberto, com participação ativa de contribuidores no desenvolvimento e correções de erros;
- Arquitetura modular, com interfaces bem definidas e que permite substituí-las facilmente por componentes customizados;
- Permite que seus componentes sejam instalados de forma distribuída e descentralizada;
- Foi projetado desde o princípio para ser escalável e capaz de obter o máximo desempenho em diversos ambientes, inclusive sobrepor uma infraestrutura existente;
- Flexibilidade de arquiteturas, podendo ser instalada em configurações distintas;
- Compatível com as APIs da Amazon, como EC2 e S3;
- Projetado para ter suporte aos *hypervisors* atuais e futuros, em especial o KVM e o Xen;
- Possibilidade de integrar com nuvens privadas.

3.1.1 Arquitetura

A arquitetura do Eucalyptus foi projetada sobre cinco componentes de alto nível, cada um responsável por delegar funções específicas, gerenciáveis através de uma interface *web* e implementados como um serviço individual. Este último aspecto oferece vantagens por incluir

uma *API* bem definida, conforme mostra a figura 8, formalizada em um documento *WSDL*, que abrange as operações que cada serviço realiza e as estruturas de entrada e saída dos dados.

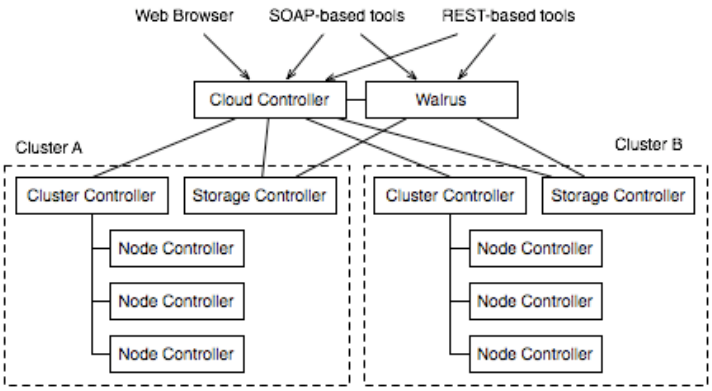

Figura 8 – Arquitetura do Eucalyptus (EUCALYPTUS, 2011).

Cada componente de sua estrutura pode ser brevemente descrito, conforme ilustrado na Figura 9 (WIKIPEDIA; Eucalyptus (computing), 2011):

- Controlador de Nó (NC – *Node Controller*) – através de um *hypervisor*, é responsável por gerenciar todas as atividades das máquinas virtuais no respectivo nó em que estiver instalado, reportando as atividades em execução e as informações de estado ao controlador de cluster (CC);
- Controlador de Cluster (CC – *Cluster Controller*) – responsável por controlar as máquinas virtuais em execução nos nós gerenciados, além de administrar as conexões de rede entre elas e o meio exterior;
- Controlador de Armazenamento (SC – *Storage Controller*) – oferece um serviço de armazenamento de blocos dinâmico em rede que pode ser vinculado as máquinas virtuais para armazenar dados de usuário e cópias de estado. Integrável com a API S3 (*Simple Storage Service*) da Amazon.

- Controlador de Nuvem (CLC – *Cloud Controller*) – É responsável por fornecer informações da nuvem para os usuários, além de exibir os recursos virtualizáveis (servidores, rede, armazenamento) através de APIs de alto nível e permitir administrá-los.
- Walrus – Habilita armazenamento escalável com uma interface implementada compatível com o Amazon S3, provendo mecanismos persistentes de armazenamento que podem ser utilizados como partições pelas máquinas virtuais.

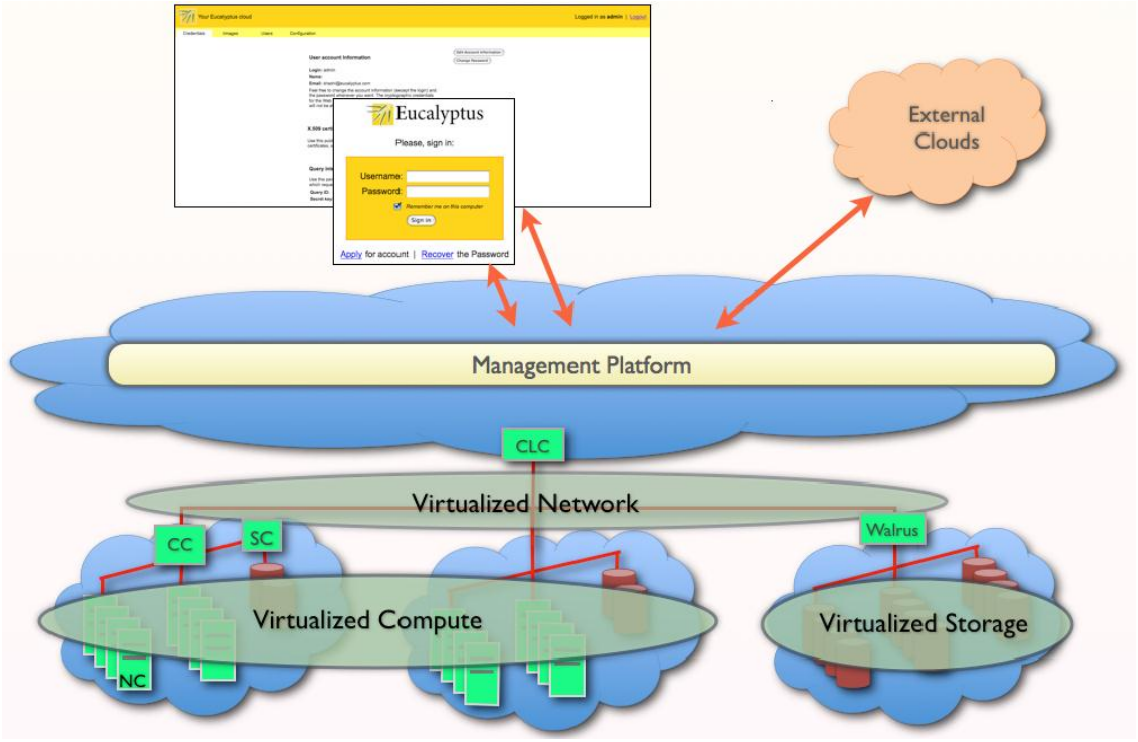

Figura 9 – Representação conceitual dos componentes do Eucalyptus e suas correlações. (EUCALYPTUS, 2010).

Contextualizando a figura 9, o CLC (Controlador de Nuvem) virtualiza os recursos subjacentes (servidores, armazenamento e rede). O Controlador de Cluster (CC) forma o *front-end* para cada *cluster* definido na nuvem. Cada Controlador de Nó (NC) administra uma máquina física, gerenciando as instâncias de máquinas virtuais em um *cluster*. O Controlador de Armazenamento (SC) fornece o serviço de armazenamento de bloco, semelhante ao Amazon EBS e o Walrus oferece um recurso escalável de armazenamento, semelhante ao Amazon S3. A plataforma de

gestão (*Management Platform*) fornece um terminal para gerenciar a nuvem, oferecendo interfaces para administradores, gerentes de projeto, desenvolvedores e usuários.

O projeto Eucalyptus oferece um curso completo e gratuito para treinamento e aprendizado da plataforma, com excelente material didático dividido em módulos e com diversos testes experimentais. Também disponibiliza diversos artigos e casos utilizando o Eucalyptus.

A versão do Eucalyptus, examinada no desenvolver deste trabalho, foi a 2.0. Encontrava-se disponível a versão *beta* do Eucalyptus 3.0. Outro aspecto relevante é que o *Ubuntu Enterprise Cloud*, em 2011, substituiu o conjunto de ferramentas de virtualização do projeto pelo OpenStack.

Em sua essência, o Eucalyptus é um *framework* completo de computação em nuvem, habilitando ao utilizador uma série de recursos integrados que possibilitam administrar grandes nuvens privadas e integrá-las com serviços contratados como o Amazon EC2.

3.2 OpenVZ

O OpenVZ é uma ferramenta de virtualização em nível de sistema operacional, baseada na arquitetura *Unix* e no *kernel* do Linux. Ela habilita um servidor físico criar e executar múltiplas instâncias do sistema operacional, conhecidas como *containers*, servidores virtuais privados (VPSs) ou ambientes virtuais (VEs), sendo similar a tecnologias como o FreeBSD Jails e o Solaris Zones (OPENVZ, 2011). Cada *container* tem seu funcionamento independente, com direito a acesso administrativo, controle de usuários, interface de rede IP, porção de memória, processos, arquivos, aplicações, arquivos de configuração e bibliotecas de sistema (OPENVZ, 2011).

O OpenVZ é base do *Parallels Virtuozzo Containers*, um software proprietário da Paralells Inc., licenciado sob a versão 2 da *GPL*, sendo mantido e patrocinado pela companhia.

Ao ser comparado com tecnologias baseadas em hypervisors, como o VMWare, Xen e KVM, a limitação que o OpenVZ impõe é o fato de que tanto o sistema virtualizador quanto as máquinas virtuais sejam distribuições Linux, porém, permite que sejam utilizadas diferentes distribuições nos ambientes virtualizados distintos, como por exemplo, CentOS, Debian e Fedora.

O diferencial do OpenVZ com relação às demais tecnologias de virtualização é seu excepcional desempenho, sendo afetada em cerca de 1 a 3% do desempenho se comparado à execução direta em um servidor físico (WIKIPEDIA; OpenVZ, 2010).

Cada *container* possui uma porção individual de recursos providos pela *kernel* do sistema hospedeiro. Dentro da *kernel*, estes recursos são ou isolados ou virtualizados, dependendo das características de disponibilidade do hardware físico, como memória RAM e núcleos do processador.

Da mesma forma, cada ambiente possui seu próprio conjunto de objetos (KOLYSHKIN, 2006):

- Arquivos – bibliotecas do sistema, softwares, diretórios "*/proc*" e *"/sys*" virtualizados, *locks* virtualizados, etc.;
- Árvore de Processos – cada *container* apenas visualiza seu próprio conjunto de processos, a começar pelo *init*, o inicializador dos processos *daemons*, programas que rodam de forma independente no background do sistema sem necessitarem intervenção do usuário. Os PIDs (identificadores de processos) são virtualizados, inclusive o PID 1, "pai" de todos os processos (WIKIPEDIA; Daemon_(computing), 2010);
- Interface de rede – Cada *container* comunica-se através de um dispositivo de rede virtual, recebendo um endereço de IP próprio e permitindo ter seu conjunto de regras (*iptables*) e regras de roteamento;

- Dispositivos – alguns são virtualizados, mas caso necessário, qualquer máquina virtual pode ter acesso a dispositivos reais (até de forma exclusiva), como interfaces de rede, porta serial, partições de disco, etc.;
- Objetos *IPC* (comunicação inter-processos por múltiplos threads) – Memória compartilhada, semáforos, troca de mensagens e chamadas remotas de procedimentos.

3.2.1 Gerenciamento de Recursos

O gerenciamento de recursos em ambientes de virtualização no nível de sistema operacional é de importância fundamental, pois há um número limitado de recursos dentro de um único *kernel* a ser compartilhado para vários ambientes virtuais ao mesmo tempo. Todos estes recursos devem ser compartilhados de forma a permitir que muitas máquinas virtuais possam coexistir em um único sistema e, principalmente, não se influenciarem mutuamente (KOLYSHKIN, 2006).

O gerenciador de recursos do OpenVZ consiste em três componentes (KOLYSHKIN, 2006):

- Gerenciador de cotas de disco de dois níveis – No primeiro nível, o administrador de um servidor OpenVZ pode definir cotas individuais de acordo com o espaço disponível ou número de instâncias. No segundo nível, o proprietário de cada máquina virtual pode usar ferramentas padrões do sistema para definir regras de cota por usuário ou por grupo de usuário;
- Escalonador de CPU “leal” – Também subdividido em dois níveis, o escalonador de CPU decide a qual ambiente virtualizado disponibilizar tempos de uso, levando em consideração o nível de prioridade da máquina e as regras de limitação. No segundo nível, o escalonador

padrão do Linux decide para quais processos no ambiente virtualizado será oferecido o tempo de uso, seguindo as regras de prioridades dos processos.

- *User Beancounters* – Um conjunto de parâmetros de instrução por *container*, composto por contadores, limites e garantias de disponibilidade. É formado por cerca de 20 parâmetros que foram cuidadosamente selecionados para cobrirem todos os aspectos de operação de uma máquina virtual, de forma que nenhum *container* abuse de algum recurso que seja limitado no sistema hospedeiro e que poderia vir a afetar o desempenho dos demais. Estes recursos são calculados e controlados, geralmente de memória, além de objetos do *kernel*, como memória compartilhada inter-processos, *buffer* de rede, entre outros.

3.2.2 Pontos de Verificação e Migração Ativa

A criação de pontos de verificação pelo OpenVZ oferece a possibilidade de executar migrações ativas de um ambiente virtual para outro servidor físico. Seu estado de execução é suspendido e todo o ambiente virtualizado é salvo em um arquivo, podendo então ser transferido para outra máquina e então reiniciado. Todo o procedimento leva apenas alguns segundos, dando a impressão de que não foi uma interrupção que ocorreu, mas sim um atraso de processamento até que as conexões de rede sejam reestabelecidas (KOLYSHKIN, 2006).

3.2.3 Utilitários do OpenVZ

O OpenVZ traz um poderoso utilitário de gerência dos ambientes virtuais, intitulado *vzctl*. Com uma interface em linha de comando em alto nível, permite criar e inicializar *containers* com poucos comandos, assim como modificar diversos parâmetros das máquinas virtuais, inclusive em tempo de execução. É possível, por exemplo, alterar o tamanho da memória ou espaço de disco da

máquina virtual sem precisar parar sua execução, geralmente impraticável em tecnologias de virtualização que utilizam emulação e paravirtualização.

A criação dos ambientes virtuais utiliza *templates*, versões pré-criadas e configuradas de imagens de sistemas com uma estrutura pronta para ser executada e gerenciada pelo OpenVZ. Os *templates* são disponibilizados e armazenados em forma de *cache*, em arquivos *tarball*, sendo cada um deles preparado e customizado com um conjunto de pacotes instalados, incluindo o acesso administrativo por *chroot*. No processo de criação de um *container* a partir de um *template*, o arquivo é descompactado e colocado em execução em poucos segundos, permitindo assim a implantação de ambientes de forma eficiente.

O utilitário *vzpkg* é o conjunto de ferramentas que facilita a criação dos *templates*, com suporte a repositórios *RPM* e *Yum*. Para criar, por exemplo, um *template* do Fedora Core 5, é preciso selecionar um conjunto de repositórios que tenha os pacotes desta distribuição e um conjunto de pacotes que serão instalados. Além disso, podem ser utilizados *scripts* antes e depois da instalação que permitem aplicar modificações e customizações no *template*. Todos estes recursos reunidos (repositórios, lista de pacotes, scripts, chaves GPG, etc.) compõem o *metadata* do *template,* permitindo que o utilitário *vzpkgcache* crie um novo template automaticamente. Ele irá fazer o download e a instalação dos pacotes listados em uma máquina virtual temporária e criar um novo *template* a partir do resultado desta instalação.

Diversos *templates* pré-configurados são disponibilizados no site do projeto, além de *templates* enviados por contribuintes com variadas opções de ambientes.

3.2.4 Desempenho

O OpenVZ oferece uma plataforma que, assim como outras tecnologias de virtualização, atende soluções de consolidação de servidores, segurança, ambientes de testes, entre tantas

outras. No seu diferencial dominante está a vantagem do desempenho elevado, com um menor *overhead* ao ser comparada com técnicas que utilizam *hypervisors*, sendo um fator crítico para tomada de decisões.

Segundo um estudo apresentado pela HP, onde os autores fazem um comparativo entre as tecnologias de virtualização no nível de sistema operacional (OpenVZ) e paravirtualização (Xen), há ganhos de desempenho consideráveis na virtualização por *containers* quando comparada a virtualização por *hypervisors*, onde o tempo de resposta para uma aplicação com múltiplos threads varia de 400 a 600% na plataforma Xen, enquanto na plataforma OpenVZ, variou apenas 100%, como mostra na figura 10 (HEWLETT PACKARD, 2008).

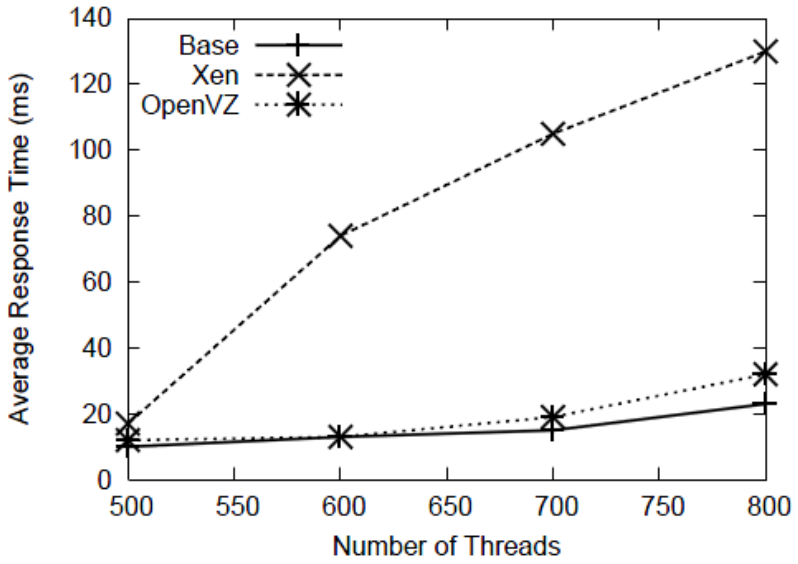

Figura 10 – Tempo de resposta é significativamente maior com o incremento de operações (threads) em uma aplicação virtualizada no Xen, comparando com o OpenVZ (HEWLETT PACKARD, 2008).

A diferença fundamental entre as duas técnicas de virtualização concentra-se no equilíbrio entre desempenho e isolamento. Além disso, elas diferem em seus propósitos, sendo que a tecnologia de *hypervisor* Xen habilita a execução de sistemas operacionais distintos, algo que não pode ser concretizado pela virtualização em nível de sistema operacional por ser geralmente executada em um *kernel* único (HEWLETT PACKARD, 2008).

O Xen proporciona um maior isolamento, separando as máquinas virtuais em regiões de memória distintas, onde apenas um conjunto selecionado de chamadas do *hypervisor* permite ao sistema operacional o acesso a recursos de hardware. Já o OpenVZ apenas organiza um conjunto de domínio de processos que são monitorados através de *beancounters* e mantém o isolamento em nível de recurso. Como resultado, sempre que houver trocas de uso de recursos entre as VMs, o Xen precisa reorganizar a tabela de gerência de memória, gerando *overheads* de tempo de acesso (HEWLETT PACKARD, 2008).

Outro aspecto relevante é o gerenciamento massivo, onde o usuário administrador do servidor físico tem acesso a todos os processos e arquivos, permitindo aplicar atualizações de segurança massivas a todos os *containers*, da mesma forma que uma atualização do *kernel* irá afetar automaticamente todos eles. É extremamente conveniente em ambientes que atualizações massivas são impraticáveis ou demandam muito esforço por parte dos administradores.

Na versão 8.04 do Ubuntu, versão entitulada *Hardy Heron*, o sistema trouxe suporte nativo à instalação e virtualização com o OpenVZ, embutidos no *kernel* da versão. Outras distribuições também o ofereciam, porém, nas atualizações posteriores, o suporte não fora incluido. Parte acredita-se em razão do despreparo dos desenolvedores, que não conseguiram entregar versões estáveis a tempo de a nova *kernel* ser lançada, disponibilizando no site do projeto versões *beta* e *stable* da recompilação dos núcleos mais recentes por quem desejar.

3.3 Proxmox VE

O *Proxmox Virtual Environment* é um *framework* de virtualização de código livre, desenvolvido e mantido pela *Proxmox Server Solutions* com apoio financeiro da *Internet Foundation Austria* (IPA). Oferece um conjunto de recursos completo de virtualização, agregando

as tecnologias virtualização baseada em *kernel* (KVM) e de virtualização baseada em sistema operacional (OpenVZ) em uma única ferramenta (PROXMOX, 2011).

É uma solução de virtualização customizada e preparada sobre o Linux Debian Lenny de 64 bits, oferecida na forma de uma imagem de instalação direta em um servidor vazio, sem necessidade de configurações ou instalação de um sistema operacional pré-configurado. Disponibiliza, em poucos minutos, um ambiente completo e pronto para entrar em produção. Também pode ser instalado em uma distribuição Lenny a partir do repositório, no caso de reaproveitar um sistema já existente, sem resultar em perda de dados.

O Proxmox VE oferece (PROXMOX, 2011):

- Sistema operacional completo (Debian Lenny) de 64 bits;
- Particionamento do disco gerenciado por LVM2 (Logical Volume Management);
- Kernel customizada, com KVM e OpenVZ integrados;
- Ferramentas de recuperação e backup;
- Interface web para administração;
- Recursos de clusterização que permitem a migração de máquinas virtuais de um servidor físico para o outro.

Como requisitos, o Proxmox VE necessita de um sistema físico de 64 bits com suporte a virtualização por hardware (Intel VT ou AMD-V) para permitir o uso do KVM. Para utilização da aplicação web, pede como requisitos o navegador Firefox e o Java.

3.3.1 Recursos

A interface de administração (figura 11) traz praticamente todos os recursos necessários para gerenciar e monitorar um ambiente de virtualização, com recursos de manutenção das

máquinas virtuais, configuração, criação de imagem de sistemas, download de *templates*, backup e gráficos de utilização dos recursos, entre outros. Disponibiliza também um recurso de acesso ao terminal das máquinas virtuais em Linux, idêntica a um acesso por SSH e também o acesso por terminais VNC para sistemas baseados em interface gráfica.

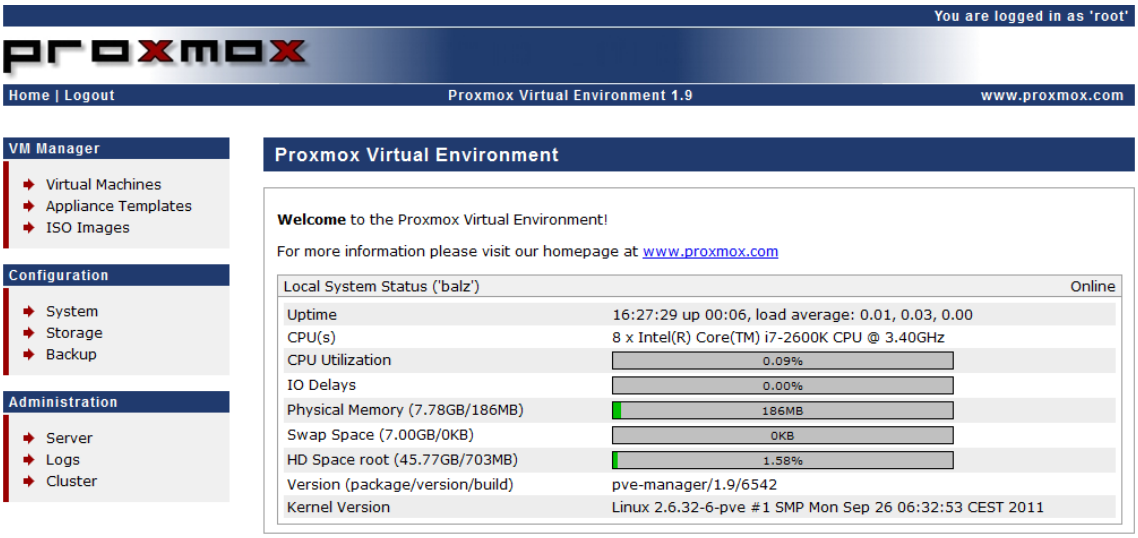

Figura 11 – Interface do Proxmox VE

O Proxmox VE oferece meios para integrar outros servidores configurados com o Proxmox VE, permitindo migrar máquinas virtuais de um sistema ao outro em questão de minutos, inclusive de forma ativa.

Permite criar, em poucos minutos, ambientes de virtualização baseados em *templates* OpenVZ (figura 12), assim como sistemas não baseados em *kernel* Linux como o Windows, instanciando-os em virtualizações pelo KVM.

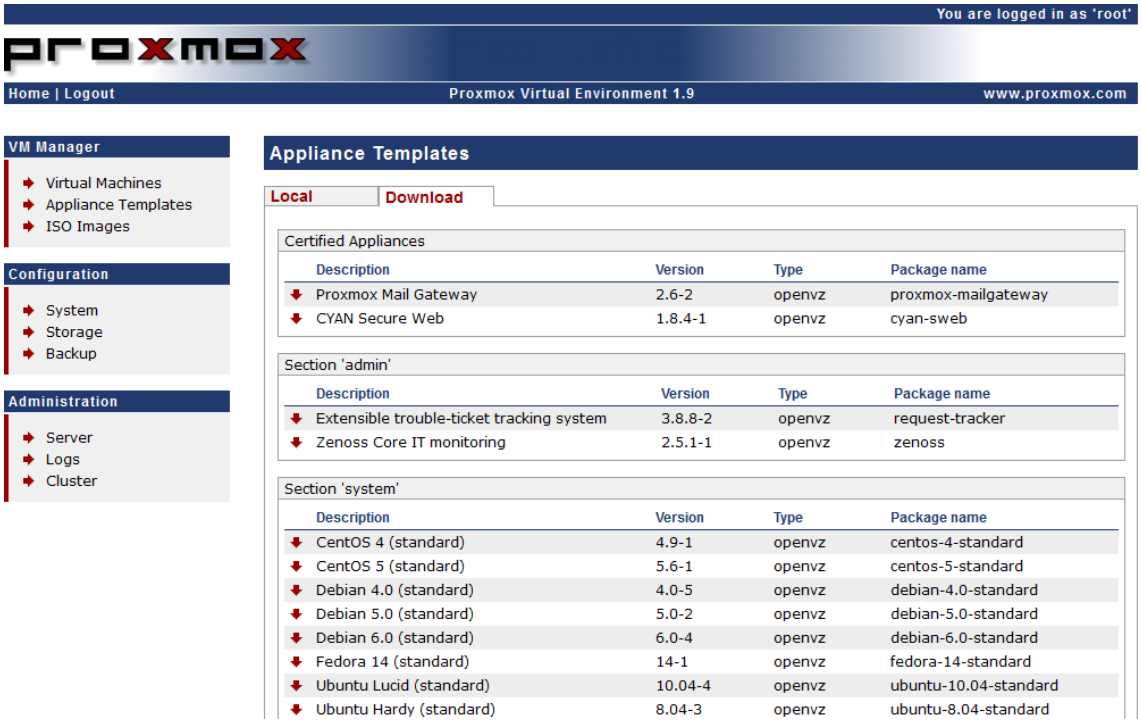

Figura 12 – Opções de template oferecidas na aplicação para download e utilização.

3.3.2 Diferenciais

A ferramenta Proxmox VE permite oferecer um sistema de virtualização com elevada otimização de recursos, oferecendo uma estrutura híbrida que possibilita instanciar sistemas operacionais que o OpenVZ não suporta, sendo uma solução livre e gratuita com possibilidades interessantes para diversas aplicações.

A eficiência da virtualização por *containers* que o OpenVZ oferece, conforme comentado no tópico anterior, incentiva a virtualização de sistemas de código livre, deixando a virtualização por KVM como um recurso adicional da ferramenta para ser utilizada em casos que demandam a utilização de um sistema comercial.

3.4 OVZ Web Panel

O OVZ Web Panel é *front-end* desenvolvido na forma de uma interface gráfica *web* para administrar recursos e máquinas virtuais de um sistema com suporte a tecnologia de virtualização OpenVZ (OpenVZ Web Panel, 2011).

Permite aperfeiçoar o uso do OpenVZ oferecendo a possibilidade de administrar praticamente todos os recursos da tecnologia de virtualização com mecanismos de configuração de ajustes mínimos do servidor, gerenciamento de servidores secundários centralizado e habilitar usuários a utilizar máquinas virtuais particulares.

Na sua versão 2.0, oferece uma suíte de administração completa e facilmente gerenciável, composta por recursos como:

- Criação e manutenção de *containers* passo-a-passo pela interface gráfica;
- Configurações dos *containers* (rede, acesso administrativo, usuário responsável, etc.);
- Criar usuários e associá-los a máquinas virtuais como responsáveis;
- Gerenciar recursos como memória, espaço em disco, uso de CPU, etc.;
- Download, upload e instalação de *templates*;
- Agregar outros servidores OpenVZ como nós de um *cluster*;
- Logs de registro de eventos;
- Interface de usuário com limitações;

Possui compatibilidade com os sistemas operacionais Debian, CentOS, Ubuntu e diversos outros, exigindo apenas algumas APIs como Ruby, Rails, Javascript e ExtJs, além, claro, do *framework* OpenVZ configurado.

3.4.1 Recursos

Semelhante ao Proxmox VE, ele permite descarregar diversos *templates* de ambientes oferecidos no site do projeto, simplesmente selecionando os sistemas de interesse e aguardando a sua disponibilização (figura 13).

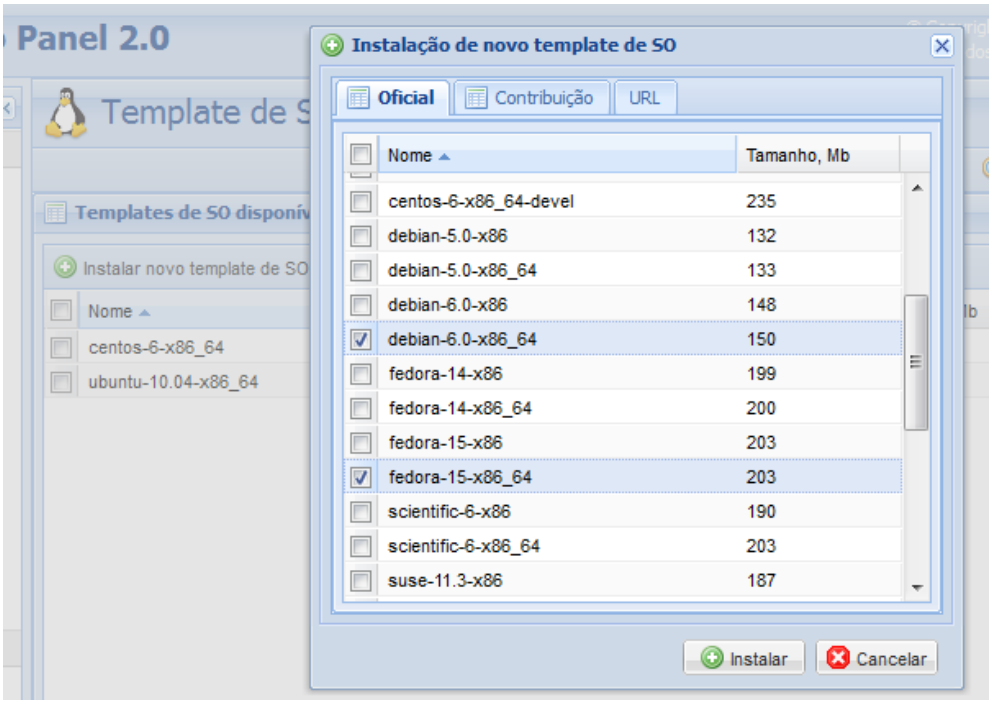

Figura 13 – Templates para download no OVZ-Web-Panel.

Oferece importantes recursos ao administrador para configurações de execução das máquinas (figura 14), assim como clonar, salvar cópias seguras do estado atual da máquina para posterior restauração e também gerar um novo *template* a partir de uma máquina em execução, preservando uma configuração de ambiente e mantendo-a pré-estabelecida para uso futuro. O recurso de clonar as máquinas virtuais é outra praticidade que a ferramenta oferece, pois permite agilizar a disponibilização de um ambiente configurado em múltiplas instâncias em poucos minutos.

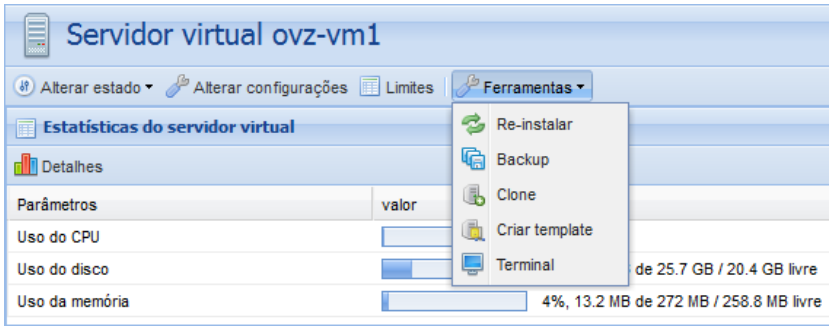

Figura 14 – Opções de administração de uma máquina virtual diretamente na interface.

Para um melhor gerenciamento dos recursos do servidor e também oferecer qualidade de serviço, é possível criar *templates* de configurações para as máquinas a serem criadas, permitindo simplificar as especificações.

Como ferramentas de monitoramento, possui recursos que apresentam estatísticas do servidor hospedeiro e também das máquinas virtuais individualmente, permitindo observar, coletar informações e detectar gargalos, como mostra na figura 15. Traz também um registro de eventos para monitorar as principais atividades no servidor, como o acesso dos usuários, atividades de estado das máquinas virtuais, falhas de sincronização, entre outras, permitindo observar e detectar situações anormais.

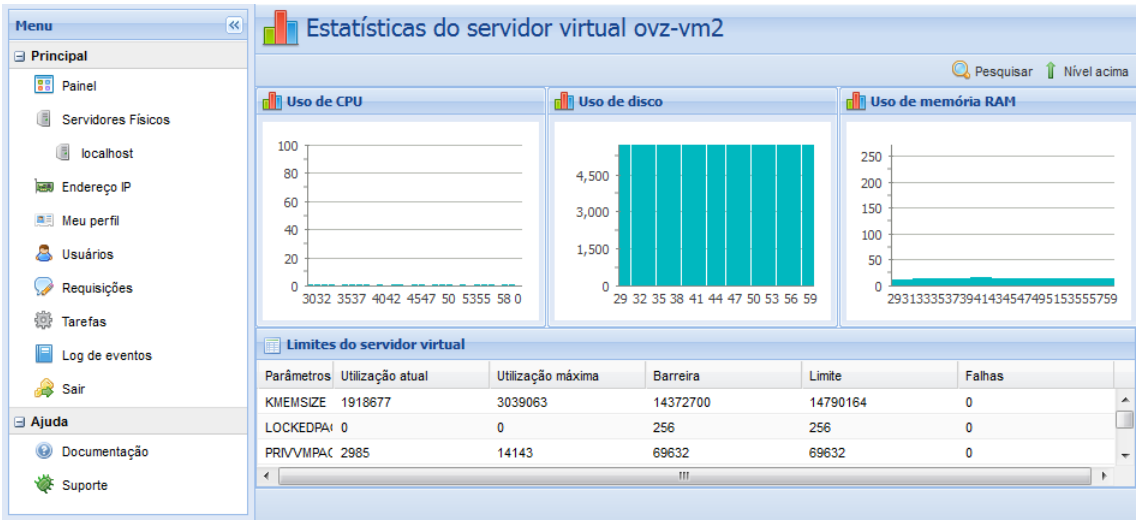

Figura 15 – Monitor de estatísticas de uma máquina virtual em execução.

Possui um utilitário de acesso ao terminal de comando que permite um acesso direto a máquina virtual, muito útil para verificação de alguns recursos e testes, porém é bastante limitada se comparado a um acesso por SSH.

3.4.2 Diferenciais

Um dos diferenciais mais importantes do OVZ-Web-Panel é a possibilidade de criar contas de usuário, vinculando-os a uma ou mais máquinas virtuais e permitindo controle total sobre os sues, o que torna a ferramenta compatível com o propósito fundamentado neste trabalho. Os usuários também podem enviar mensagens pedindo suporte aos administradores, solicitando alguma atividade que não está ao seu alcance.

4. Estudo de Caso

4.1 Descrição do Estudo de Caso

O principal interesse deste trabalho é estudar a possibilidade de oferecer um serviço de computação em nuvem, capaz de proporcionar aos usuários um sistema gerenciador de máquinas virtuais o mais completo possível, com recursos administrativos, utilizando ferramentas e sistemas disponíveis de código aberto.

O estudo de caso descrito a seguir tem como objetivo apresentar os desafios e soluções encontradas na preparação e escolha das ferramentas para o serviço proposto.

4.2 Sistemas e Ferramentas Utilizadas

Com base nas plataformas e ferramentas de apoio a virtualização que foram descritas no Capítulo 3, o ambiente de experimentos proposto foi definido com os seguintes recursos, sustentados pelas suas justificativas.

4.2.1 Sistema Operacional

O sistema operacional escolhido foi o Ubuntu Server 10.04 LTS (*Lucid Lynx*) por oferecer suporte necessário para os testes das ferramentas e ser uma versão classificada como *LTS* (suporte a longo prazo) para a qual, de acordo com os mantenedores do sistema, são oferecidas atualizações de software, segurança e compatibilidade de hardware em um período estendido por pelo menos cinco anos. Também oferece a segurança de ser considerada estável e robusta, sendo moderadamente livre de falhas, pois todos os recursos oferecidos na versão passam por rigorosos testes antes de serem incorporados no *release* final.

O Ubuntu traz nativamente o *hypervisor* KVM como *back-end* de virtualização e a biblioteca *libvirt* como interface de programação de aplicativos (*API*), permitindo virtualizar sistemas distintos. Por si só, o KVM traz apenas suporte a virtualização de sistemas sem uma interface gráfica, sendo utilizado o QEMU-KVM, uma versão híbrida, que utiliza recursos de simulação de hardware gráfico para a interface, mantendo as vantagens da aceleração de virtualização pelo KVM.

Outro aspecto que favoreceu a escolha do sistema operacional Ubuntu na versão 10.04 foi a questão da compatibilidade da tecnologia de virtualização no nível de sistema operacional OpenVZ, apoiado pelo fato de ser a versão mais atual com suporte estável para a recompilação do *kernel* com as configurações que habilitam a técnica de virtualização por sistema operacional.

4.2.2 Plataformas de Virtualização

A plataforma de virtualização Eucalyptus é intimamente relacionada com a versão 10.04 do Ubuntu Server, sendo oferecida em sua estrutura como plataforma de computação em nuvem apoiada pelo projeto. Desenvolvida em código aberto, possui extensa documentação de apoio e compatibilidade com diversas distribuições Linux. Foi concebido com base tanto na virtualização com o *hypervisor* Xen quanto KVM.

Como ferramenta de administração das máquinas virtuais na tecnologia OpenVZ, foi utilizado o OVZ-Web-Panel, uma ferramenta de interface web desenvolvida para gerenciar e aprovisionar os recursos de maneira facilitada, sem necessitar conhecimentos avançados em Unix e com mecanismos de provisionamento de VMs para terceiros. A versão utilizada nos experimentos foi a 2.0.

Outra ferramenta de virtualização escolhida foi o Proxmox VE por oferecer recursos híbridos de virtualização tanto com o QEMU-KVM quanto OpenVZ. No desenvolver deste

trabalho, foi utilizada uma imagem de instalação oferecida na página *web* do projeto, baseada no Linux Debian Lenny, na versão 1.9 e também na versão 2.0, que até a conclusão deste trabalho, esta mais recente encontrava-se na versão *beta*. O sistema oferece uma interface *web* para gerenciar o servidor de virtualização completa, com todos os recursos necessários, sendo facilmente instalada e disponibilizada em poucos minutos. É possível instalar o Proxmox VE no sistema Ubuntu, porém, optou-se por utilizar o Debian em razão das diferenças serem pequenas, pois o *kernel* é o mesmo nas duas distribuições.

4.2.3 Critérios para Definição e Escolha dos Meios Virtualizadores

A escolha das plataformas e ferramentas de virtualização anteriormente descritas baseou-se nos seguintes critérios:

- Tecnologia desenvolvida em código aberto – Ser desenvolvida e mantida por comunidades e projetos que apóiam o software livre, permitindo a participação de colaboradores e a utilização irrestrita.
- Desempenho – Oferecer elevado desempenho, permitindo um melhor aproveitamento do poder computacional e consequentemente resultando em um número maior de máquinas virtuais em um único sistema.
- Mecanismos de controle – Possuir ferramentas administrativas que facilitem a gestão do serviço, oferecendo uma interface completa para criação e manutenção das máquinas virtuais.

Com base nestes aspectos, foram definidas as tecnologias de virtualização no nível de sistema operacional OpenVZ e a de virtualização completa com o *hypervisor* KVM para servirem de base ao trabalho. Ambas são desenvolvidas com código livre e são compatíveis, permitindo oferecer um recurso híbrido para hospedar máquinas virtuais.

A virtualização no nível de sistema operacional com o OpenVZ permite não apenas ter uma maior performance que as outras tecnologias, mas também oferecer a possibilidade de restringir o usuário a utilizar apenas sistemas operacionais de código livre, evitando complicações legais com o uso de softwares proprietários.

A virtualização completa com KVM oferece a vantagem de possibilitar a virtualização de sistemas proprietários, sendo desencorajada a sua oferta de serviço livremente por depender de uma maior administração dos mantenedores para controlar sua utilização e verificar as conformidades. Sua inclusão na proposta foi motivada por considerá-la um recurso de trazer vias para virtualizar sistemas em situações diferenciadas, que obrigam a necessidade de utilizar softwares comerciais ou softwares livres que demandam recursos que a virtualização com o OpenVZ não permite.

Através destas duas técnicas, buscou-se configurar um sistema virtualizador capaz de atender a proposta e prover um ambiente híbrido para criação de máquinas virtuais.

4.3 Premissas do Cenário Proposto

Para o desenvolvimento do ambiente proposto, foram definidos critérios específicos para avaliar a disponibilização do serviço. Estes foram apontados como sendo requisitos fundamentais para viabilizar a sua implantação de forma segura e funcional:

- Interface de acesso – Disponibilizado na forma de um *web service* com uma interface intuitiva, de fácil entendimento ao usuário;
- Autonomia – Acesso simplificado aos usuários, com meios que o permitam cadastrar-se no serviço e requisitar a criação de uma máquina virtual. Como adicional, ter meios de gerenciá-la remotamente sem necessitar a intervenção dos responsáveis pelo serviço para operá-la.

- Regras de Utilização – Para utilizar o serviço, devem ser impostas regras e limitações a serem definidas pelos gestores do serviço. Ao cadastrar-se, o usuário deve concordar com as regras de utilização;
- Perfis de usuário – Cada usuário pode ter uma ou mais máquinas associadas a sua conta, sendo possível conferir informações sobre ela e alterar algumas configurações básicas. Também com recursos para iniciar, parar, reiniciar, salvar e restaurar a VM de um estado anterior, se possível reinstalar por completo em caso de alguma pane;
- Segurança – Deve ser configurado um ambiente de rede ao sistema para não oferecer risco de segurança aos demais recursos da universidade. Todas as máquinas estarão em um *range* de endereços estáticos, sendo facilmente associáveis aos usuários, o que facilita a identificação em caso de infrações de uso. Deve utilizar certificados de segurança SSL e chaves criptográficas na comunicação.

4.4 Recursos Utilizados

Para a implementação desta proposta foram utilizados dois computadores de hardware comum, não especializado, cada um com as seguintes configurações:

- Computador 1 – Processador Intel i7 2600k, com 8Mb de *cache*, *clock* de 3.4Ghz, 4 núcleos (8 *threads*) e 64 bits; 8Gb de RAM DDR3 1800 mhz; Disco rígido SATA II de 200Gb.
- Computador 2 – Processador AMD Athlon II X2, com 2Mb de *cache*, *clock* de 2.8Ghz, 2 núcleos e 64 bits; 2Gb de RAM DDR2 800mhz; Disco rígido SATA II de 200Gb.

O Computador 1 foi preparado para servir de plataforma principal de testes das tecnologias de clusterização, sendo utilizado como um nó de nuvem privada com a ferramenta Eucalyptus. Nos testes utilizados com o OpenVZ e o Proxmox VE, o mesmo foi configurado para operar como

servidor principal, pois os recursos de administração das tecnologias são disponibilizados no próprio sistema e não requerem um nó de administração (*front-end*).

O computador 2 foi inicialmente designado como nó de administração (*front-end*), responsável por coordenar os serviços da tecnologia Eucalyptus, encarregando-se de gerenciar a rede, monitorar as máquinas virtuais em execução nos nós e abrigar o repositório de máquinas virtuais. Nos testes com OpenVZ e Proxmox VE, foi configurado como um segundo servidor a ser integrado como nó ao principal, sendo utilizado para testes de *backup* e migração de máquinas virtuais.

Ambas as máquinas são interconectadas por um *Switch* de 8 portas *gigabit*, servidos de conexão com a Internet através de um roteador/gateway.

4.5 Processo de Instalação e Configuração do Ambiente

A preparação dos cenários para avaliação com os recursos computacionais descritos utilizou as ferramentas de virtualização apresentadas, utilizando o sistema operacional proposto. O processo dividiu-se em duas fases de experimentos, descritas a seguir.

4.5.1 Eucalyptus e OpenVZ

Na primeira fase de experimentos, foi definido um laboratório a ser preparado com o objetivo de avaliar o seguinte sistema:

- Sistema operacional Linux Ubuntu Server LTS na versão 8.04 (*Hardy Heron*);
- *Framework* de virtualização OpenVZ;
- Ferramenta de administração de nuvens privadas Eucalyptus.

A proposta deste primeiro laboratório foi de integrar os recursos que o Eucalyptus oferece para administrar nuvens privadas de virtualização com o ambiente de virtualização OpenVZ, configurando de forma que pudessem ser gerenciadas pela ferramenta. O OpenVZ teve seus recursos integrados no *kernel* Linux que o Ubuntu 8.04 utiliza, sendo esta o motivo de realizar o experimento com um sistema já defasado. Nas edições seguintes, o *kernel* não trouxe os recursos de virtualização do ambiente, o que forçou uma recompilação posterior, sem muitas garantias de estabilidade e confiabilidade.

Pelo fato de o Eucalyptus oferecer uma interface gráfica *web*, com recursos de cadastro de usuários e disponibilização de máquinas virtuais de forma facilitada, cogitou-se que a mesma teria algum suporte de compatibilidade com a técnica de virtualização que o OpenVZ utiliza, diferentemente da técnica com *hypervisors* que a ferramenta opera.

Após preparar o sistema operacional com suporte a virtualização do OpenVZ e também do *hypervisor* KVM, o passo seguinte era preparar a instalação do Eucalyptus. Pelo fato do Eucalyptus ter sido desenvolvido para versões posteriores do Ubuntu, foi necessário portar a instalação, convertendo os arquivos-fonte. Durante esta fase de preparação, diversas dependências do Eucalyptus não eram possíveis de aplicar pelo fato de não terem sido compatibilizadas com a versão do *kernel* do Ubuntu 8.04. Isto é um fator considerado natural, pois na medida em que surgem novas versões do sistema (atualmente está na versão 11.04), não são preparadas atualizações para os sistemas antigos, também como forma de incentivar o usuário a manter-se atualizado. Sendo assim, o experimento inicialmente proposto não foi possível de ser concretizado.

Em uma revisão desta primeira fase de experimentos, foi trocado o sistema operacional Linux Ubuntu Server da versão 8.04 LTS para a versão 10.04 LTS, a mais recente com suporte a longo prazo. Este possuía compatibilidade total com o Eucalyptus, que inclusive é embarcado na

instalação e identificado como Ubuntu Enterprise Cloud, sendo oferecido como uma solução de computação em nuvem apoiada pelos projetos da Canonical.

Após instalado e configurado o Eucalyptus, foi iniciado o processo de instalação do OpenVZ. Como o *kernel* do Ubuntu 10.04 não possui os recursos necessários para suporte ao *framework,* foi necessário recompilar o mesmo com as configurações e dependências necessárias. O processo está documentado no Anexo 1 do apêndice deste trabalho.

Com os dois sistemas de virtualização em operação, foi pesquisada uma solução que tornasse possível integrar e gerenciar máquinas virtuais do OpenVZ dentro do Eucalyptus, buscando estender para ele a possibilidade de gerenciar máquinas virtuais de duas técnicas distintas. Um dos fatores que motivou o experimento foi o fato de ser utilizada a biblioteca de virtualização *libvirt* no cerne do Eucalyptus. O *libvirt* é uma *API* de código livre que concentra rotinas e funções de programação para diversas técnicas de virtualização, inclusive o OpenVZ, oferecendo um conjunto de recursos completo para construir ferramentas de gestão de máquinas virtuais. Apesar do potencial da biblioteca, os desenvolvedores do Eucalyptus focaram em utilizar apenas os recursos providos pelo *libvirt* para trabalhar com os *hypervisors* Xen e KVM, deixando de fora o OpenVZ. Até o fim desta pesquisa, não foram encontradas soluções que contornassem ou viabilizassem a integração do OpenVZ ao Eucalyptus, tornando impraticável a união pretendida dos dois recursos de virtualização.

4.5.2 OpenVZ e Proxmox VE

Nesta segunda fase de experimentos, após ter sido abandonada a proposta de utilizar o *framework* Eucalyptus, foi mantido o uso do OpenVZ e também mantido o *hypervisor* KVM. Como o KVM tem suporte nativo no *kernel* utilizado pelo Ubuntu, a solução de virtualização completa foi

considerada como um recurso secundário por viabilizar a virtualização de máquinas não compatíveis com o OpenVZ, até mesmo sistemas privativos, proporcionando uma solução híbrida.

Novamente foi preparado um ambiente com o sistema operacional Ubuntu 10.04, preparando-o com suporte a virtualização pelo OpenVZ, conforme o mostra o Apêndice 1. Como *front-end* para gerenciar a virtualização, foi escolhida a ferramenta OVZ-Web-Panel. Para a virtualização com o *hypervisor* KVM, foi realizado um experimento utilizando o ambiente de virtualização Proxmox VE.

4.5.2.1 OVZ-Web-Panel

O OVZ-Web-Panel foi escolhido por oferecer uma interface gráfica de usuário na forma de aplicação *web*, com praticamente todos os mecanismos de controle necessários para gerenciar o sistema de virtualização OpenVZ diretamente sobre o navegador de Internet. Além disso, oferece um recurso para cadastro de usuários, vinculando-os a máquinas virtuais que o mesmo requisitar, permitindo disponibilizar uma maior autonomia para utilização. A tela principal do OVZ-Web-Panel pode ser conferida na figura 16.

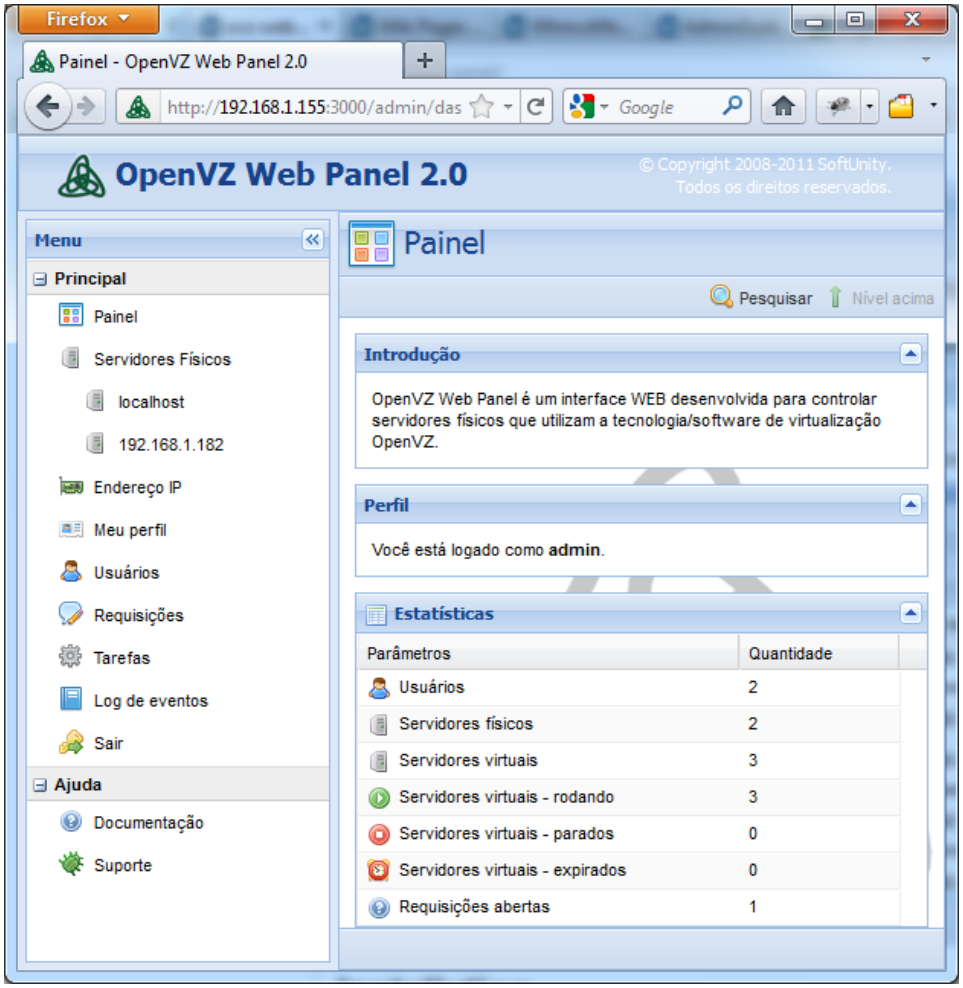

Figura 16 – Tela inicial do OVZ-Web-Panel, no modo administrador.

A inclusão de um nó secundário requer apenas que o sistema esteja pré-configurado com o *framework* OpenVZ e as bibliotecas de programação *Ruby* como suporte para os *scripts* de intercomunicação (figura 17). Oferece ainda como configuração adicional recursos de encriptação por SSL para tornar seguras as comunicações entre os nós do cluster. Como o OVZ-Web-Panel foi instalado no mesmo servidor (localhost) que está configurado o OpenVZ, ele já inclui o próprio como um servidor hospedeiro de máquinas virtuais. A ferramenta permite ser instalada em um servidor administrativo (*front-end*) caso o administrador opte por fazer.

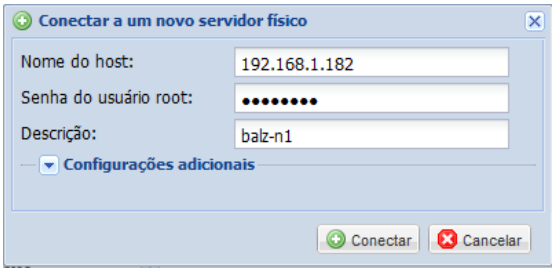

Figura 17 – Incluindo um servidor físico para ser administrado pelo OVZ-Web-Panel.

Com dois ou mais nós, é possível realizar a migração de máquinas virtuais em poucos segundos. Para isso, os dois nós devem trocar chaves públicas de forma a habilitar a transferência segura (figura 18). A migração é um recurso que depende de intervenção manual para configurar as chaves e é exclusiva para o administrador.

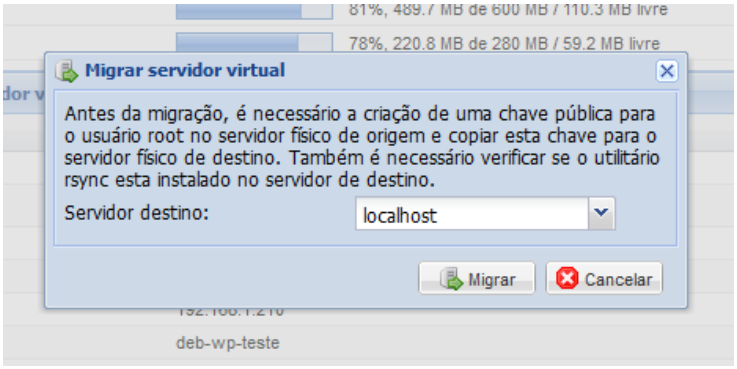

Figura 18 – Migração de uma máquina virtual de um nó para outro.

Para criação de máquinas virtuais, o OVZ-Web-Panel sugere diversos *templates* prontos, como comentado na descrição da ferramenta no Capítulo 3. Estes templates podem ser descarregados no servidor e disponibilizados para a criação de máquinas diretamente na interface *web*, sem demandar linhas de comando para a operação.

Antes de criar uma máquina virtual, é necessário escolher ou definir as configurações do *container* que irá abrigá-la. Isso permite ao administrador uma melhor utilização dos recursos físicos do servidor hospedeiro, otimizando o provisionamento e viabilizando a criação de um número maior de máquinas virtuais, assim como dedicar mais recursos a uma máquina específica que prevê um maior uso. Podem ser configurados *templates* de servidor com configurações

específicas, como espaço em disco, memória RAM, uso do processador, configurações de rede e também opções avançadas de diversos parâmetros (figura 19).

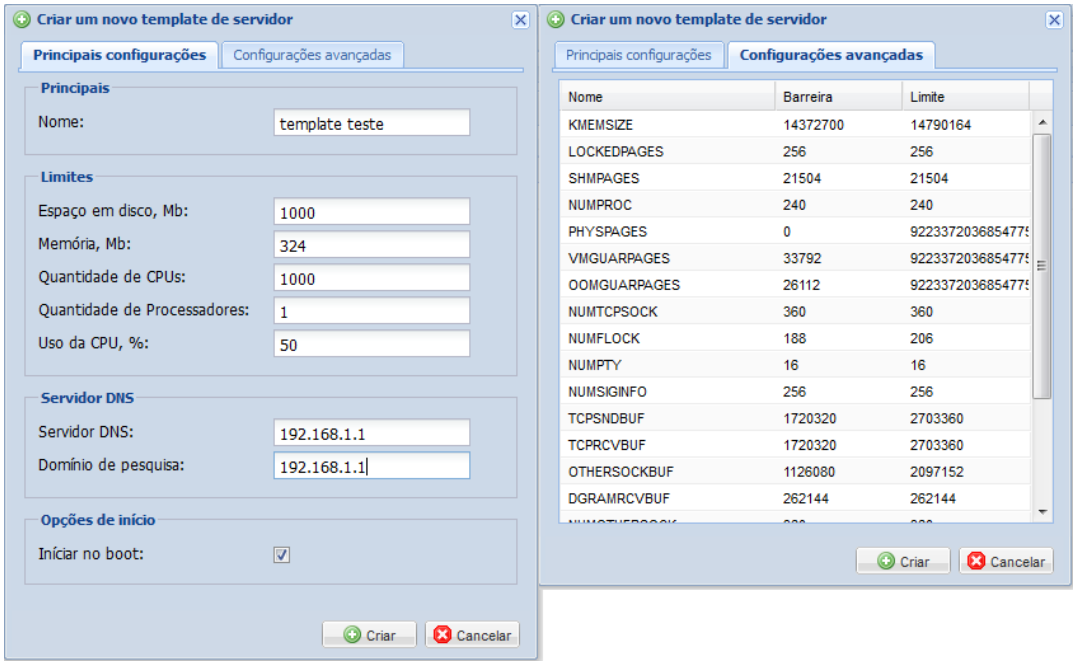

Figura 19 – Opções para criação de templates de containers para abrigar máquinas virtuais.

A alocação de recursos de processamento é realizada através de uma implementação de um escalonamento de dois níveis pelo OpenVZ, como forma de oferecer uma estratégia de compartilhamento de CPU de forma justa, baseada em dois parâmetros: *cpuunits* e *cpulimit*. No primeiro nível, o escalonador aloca o tempo de uso da CPU para os *containers* baseado nas Unidades de Processamento (*cpuunits*) que foram destinadas. No segundo nível, o escalonador do Linux entra em cena, decidindo quais processos devem rodar nos *containers* de acordo com os padrões de prioridades de processo. Isso habilita o administrador a definir diferentes valores de Unidades de CPU para distintos *containers* de acordo com as necessidades de uso. Já o parâmetro de Uso da CPU (*cpulimit*) é uma maneira de limitar o uso da CPU, assim como também é possível indicar o número de processadores (núcleos) que ela pode utilizar.

É possível também alocar diversos limites avançados, como número de máximo de processos em execução, entre outros, permitindo ajustes finos na configuração para aperfeiçoar o desempenho do sistema.

4.5.2.2 Proxmox VE

O Proxmox VE, como descrito no Capítulo 3, é uma ferramenta que integra a virtualização por *containers* utilizando o OpenVZ e pelo *hypervisor* KVM. A plataforma é oferecida sobre o sistema operacional Debian Lenny pelo fato desta distribuição manter o suporte ao OpenVZ, diferentemente do Ubuntu, no qual foi necessário recompilar o *kernel*. É possível também instalar a ferramenta para ser utilizada no Ubuntu, mas por questões de estabilidade, a versão utilizada nos testes foi a instalação a partir da ISO baseada no Debian Lenny 5.

A instalação do Proxmox VE é feita de forma rápida e facilitada, com algumas poucas configurações necessárias na instalação, como configurações regionais, teclado, senha de administrador e interface de rede. O restante é automático, sendo concluído em poucos minutos (figura 20).

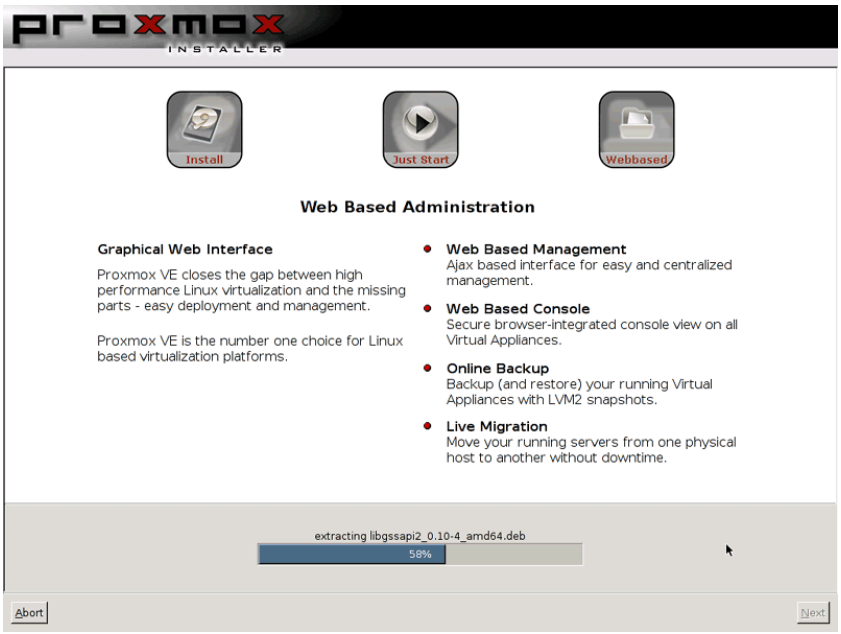

Figura 20 – Andamento da Instalação do Proxmox VE.

Depois de concluída a instalação e inicializado, o sistema está pronto para entrar em produção, restando acessar o seu endereço IP em um navegador de Internet para utilizar o console de gerenciamento (figura 11 do Capítulo 3).

Um dos grandes benefícios que o Proxmox VE oferece é a possibilidade de instanciar tanto máquinas virtuais baseadas em *container* quanto máquinas em virtualização completa, através da utilização do *hypervisor* KVM. Este cenário híbrido possibilita atender a demandas específicas, como no caso a utilização de um sistema incompatível com a virtualização por *containers* ou um sistema proprietário.

Da mesma forma que o OVZ-Web-Panel, existem diversos *templates* prontos para a criação de *containers* que são disponibilizados, bastando efetuar o *download* para instanciar as máquinas. Para criação de máquinas virtuais baseadas em virtualização completa com o KVM, é necessário fazer o envio de imagens de instalação dos sistemas para o servidor, sendo realizado através da própria interface *web* de forma facilitada, mantendo-as em um repositório para futuras necessidades (figura 21).

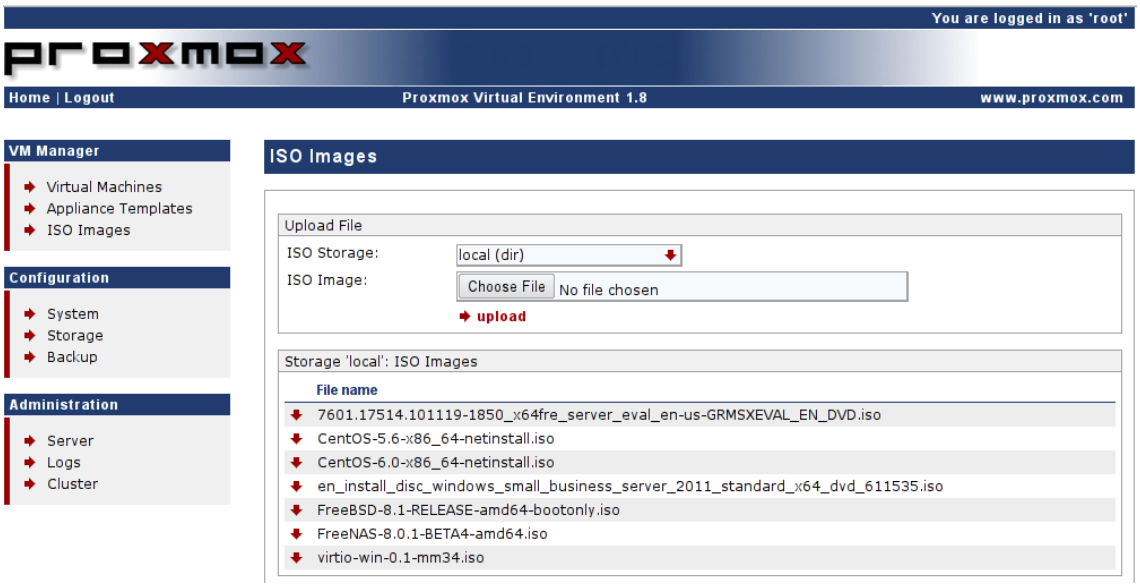

Figura 21 – Repositório de imagens de sistemas operacionais para criação de máquinas virtuais pelo KVM.

Como descrito no Capítulo 3, o Proxmox VE possibilita a integração com outros servidores, agregando-os como nós de um *cluster*, permitindo a migração de máquinas virtuais de um servidor ao outro, balanceando a carga de utilização dos servidores. A migração pode ser feita de forma ativa, ou seja, com um curto período de indisponibilidade até que o sistema seja movido por completo. Todo o processo de migração é realizado através da interface gráfica, sem depender de intervenção manual no processo.

O acesso às máquinas virtuais pode ser feito através de uma interface de acesso remoto utilizando o protocolo VNC com uma ferramenta disponibilizada na própria interface *web*, permitindo a execução de terminais para as máquinas virtuais. No caso de sistemas com interface gráfica, a exemplo do Windows, é possível acessá-la sem precisar algum software pré-instalado na máquina virtual, permitindo configurar através dela um recurso para ser acessado apropriadamente por outras vias, como o próprio sistema de conexão remota que ele oferece. No caso de máquinas sem interface gráfica, o acesso é realizado com a mesma ferramenta ou através de conexão SSH.

O Proxmox VE permite a realização de cópias de segurança das máquinas instanciadas, oferecendo vias de restaurar instâncias comprometidas, porém, não oferece recursos de clonagem como o OVZ-Web-Panel.

Na criação das máquinas virtuais, as opções de configuração são diferenciadas, permitindo customizar diversos recursos. Uma característica observada é a diferença entre os recursos de rede que a ferramenta apresenta, com opções de configuração de *bridge* e placas de rede, permitindo escolher dispositivos virtualizados compatíveis e também dispositivos físicos. Apesar destes benefícios, a ferramenta não oferece recursos de definir uma faixa de endereços IP para as máquinas, dependendo de uma definição manual. No OVZ-Web-Panel, esta faixa de endereços é configurável, onde cada máquina nova consome um IP da faixa reservada.

Diferente do OVZ-Web-Panel, é provida apenas uma interface administrativa, sem a possibilidade de criar contas de usuários com perfil limitado. A versão 2.0 do Proxmox VE, que durante este trabalho encontrava-se na versão *beta*, um dos objetivos do cronograma para a versão é possibilitar esse recurso, com vias de integração a bases de usuários, o que seria um mecanismo interessante para facilitar a gestão dos usuários do serviço.

5. Projeto, Desenvolvimento e Resultados

5.1 Projeto e Implementação

Para preparar o cenário e avaliar as ferramentas escolhidas, foram levantados critérios separados em três categorias para julgar as necessidades que as ferramentas devem suprir para viabilizar o projeto, sendo eles: fundamentais, essenciais e complementares.

Critérios fundamentais são considerados os meios cruciais para viabilizar o serviço, que podem ser descritos como:

- Desenvolvido em código livre, permitindo uso irrestrito e modificações;
- Oferecer uma interface gráfica do utilizador (*GUI*) para administrá-la;
- Permitir criar e controlar máquinas virtuais Linux, com acesso remoto por SSH e perfil administrativo (*chroot*) para o usuário;
- Oferecer recursos de isolamento, segurança e gerenciamento de desempenho;
- Permitir o gerenciamento de usuários do sistema, os quais não devem possuir privilégios administrativos sobre o serviço.

Os critérios essenciais são considerados mecanismos facilitadores na administração do serviço, permitindo gerenciar os recursos por meio da interface e agilizar o provisionamento por parte do administrador. Podem ser descritos como:

- Oferecer opções de configuração dos ambientes das máquinas virtuais (processador, memória, espaço em disco, rede);
- Oferecer uma lista de opções de modelos de máquinas virtuais prontos para utilização;

- Mecanismos de monitoramento do uso dos recursos;
- Recursos de “autosserviço” para os usuários, onde os mesmos possam ligar, desligar, reiniciar e fazer *backup* das máquinas virtuais de forma autônoma, sem necessitar intervenção do administrador.

Os recursos complementares, como diz o significado do termo, são considerados adicionais à utilização do serviço, não essencialmente necessários, mas que possam agregar funcionalidades para a ferramenta e trazer maiores benefícios. Como recursos complementares, podemos definir:

- Documentação e tutoriais de utilização;
- Utilitários de acesso ao terminal da máquina virtual pela interface gráfica;
- Permitir replicar máquinas virtuais.

5.2 Implantação e Resultados

Nesta etapa, foram realizados testes de desempenho para analisar o comportamento das máquinas virtuais com relação aos recursos providos, buscando compreender o funcionamento do escalonador de processos com um número considerável de execuções simultâneas. Na sequência, foram analisadas as duas ferramentas e suas características, de forma a coletar informações sobre o funcionamento também e comparar os mecanismos que cada uma oferece para viabilizar a implantação e disponibilização do sistema.

5.3 Testes de Desempenho

Para avaliar o comportamento do servidor com um número elevado de máquinas virtuais em execução simultânea, foi proposto um ambiente com uma quantidade de 15 máquinas com alguma atividade em andamento de forma a consumir 100% dos recursos de processamento.

Cada máquina virtual foi criada contendo 700 MB de espaço físico e 256 MB de memória RAM, utilizando um template do Ubuntu 10.04 x86_64. Foi definida também uma quantia de 500 unidades de CPU, buscando dividir o total de 8000 unidades que o servidor provisiona, de forma a aproximar o uso de 100% dos recursos e sobrar uma porção para o sistema.

Neste *template*, foi preparado um *script* baseado em um *loop* infinito *md5sum* para estressar o uso de CPU pelas máquinas virtuais. O *md5sum* é um programa que calcula uma *hash MD5* de um arquivo para ser utilizado como assinatura digital, forma popular de verificar a sua integridade de arquivos em validações de segurança. Inicialmente, foi criado um arquivo de conteúdo aleatório, com 50 MB de tamanho, através do seguinte comando:

```
dd if=/dev/urandom of=arquivo count=50 bs=1024k
```

O *script* utilizado para calcular o *hash* MD5 do arquivo contém o seguinte código:

```
#!/bin/sh
#Script de stress do CPU - stress.sh
i=0
while [ 1 ]
do
   md5sum arquivo
```

```
    i=`expr $i + 1`
    echo "Iteration: $i"
done
```

Estes dois arquivos foram salvos na pasta */etc/init.d/* e o *script* foi configurado para ser carregado na inicialização do sistema. Em seguida, o *template* foi replicado em um total de 15 máquinas virtuais, com o objetivo de colocar todas em execução simultânea para observar o comportamento do escalonador de processos.

Com apenas uma máquina em execução, o consumo de CPU que a máquina virtual obtém em uso máximo varia de 11 a 12% (figura 22).

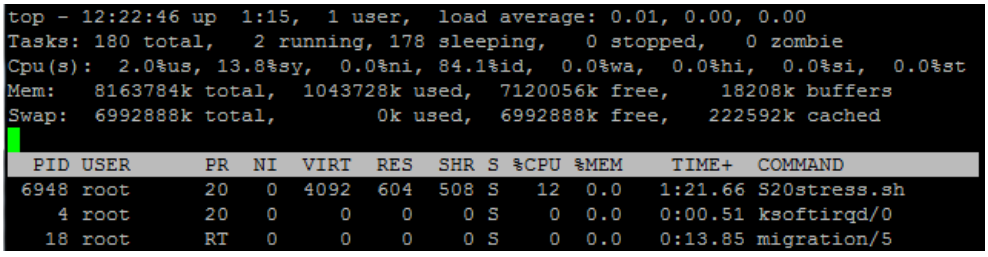

Figura 22 – Uso de processamento com uma instância em execução.

Ao ser analisado, percebeu-se que o parâmetro não foi seguido como o esperado, pois fora especulado que iria trabalhar em torno 6%. No caso, está sendo utilizando 100% do uso de um núcleo físico.

Após revisar o funcionamento dos parâmetros, constatou-se que o valor medido de 12% corresponde à utilização de um núcleo por um único processo, justificado pelo fato de que, independente das Unidades de CPU definidas com uma única máquina em execução, ela utiliza o máximo de recursos disponibilizados. Se forem executadas múltiplas instâncias deste mesmo processo na mesma máquina virtual, estas irão consumir todos os recursos disponíveis, apenas considerando os limites fixados de unidades de CPU em caso de uso concorrente. A figura 23

mostra múltiplas instâncias, sendo estas executadas gradualmente na mesma máquina, consumindo os recursos disponíveis a cada novo processo até saturar a disponibilidade de recursos.

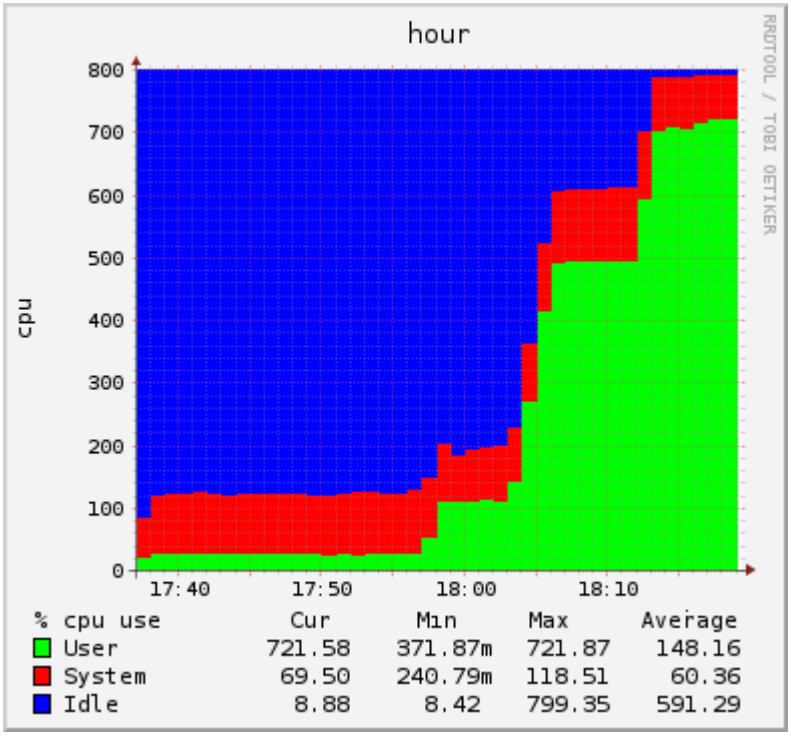

Figura 23 – Execução gradual de múltiplas instâncias do mesmo script em uma única máquina virtual.

Ao retomar o padrão inicialmente proposto de um processo por máquina virtual, foi iniciada uma segunda instância, onde o consumo medido manteve-se em torno de 12% para cada *container* ativo (figura 24), confirmando que o escalonador desconsidera o parâmetro quando há sobra de recursos.

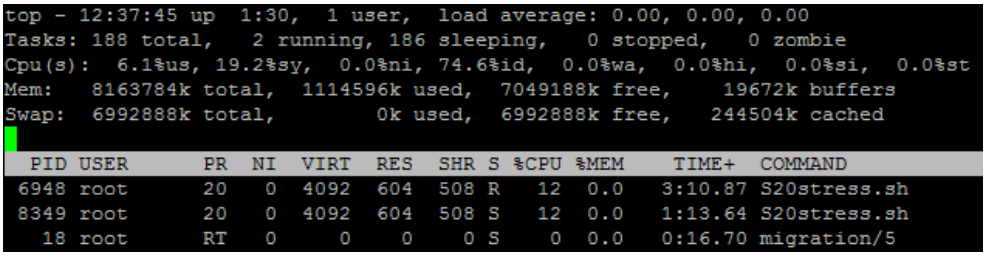

Figura 24 – Execução de duas máquinas virtuais, cada uma com uma execução do script.

As máquinas virtuais restantes foram gradualmente inicializadas, conforme mostra a figura 25. A cada máquina ativada, o uso de CPU eleva 100 pontos, indicando que um núcleo inteiro foi dedicado para o *container*. Ao serem ativadas oito instâncias, o gráfico de uso atinge os 800 pontos, indicando que no momento está sendo alocado um núcleo para cada *container*. Ao ser iniciada a 9ª máquina virtual, os recursos começam a ser redistribuídos, resultando em uma utilização aproximada de 94% do total e considerando enfim a alocação de Unidades de Processamento. Manteve-se nessa proporção até a 12ª máquina ser iniciada aproximadamente às 23h47min horas no gráfico, quando elevou o uso para 97% e 98% quando a 13ª foi ativada. Manteve-se em torno de 99% com a ativação da 14ª e da 15ª.

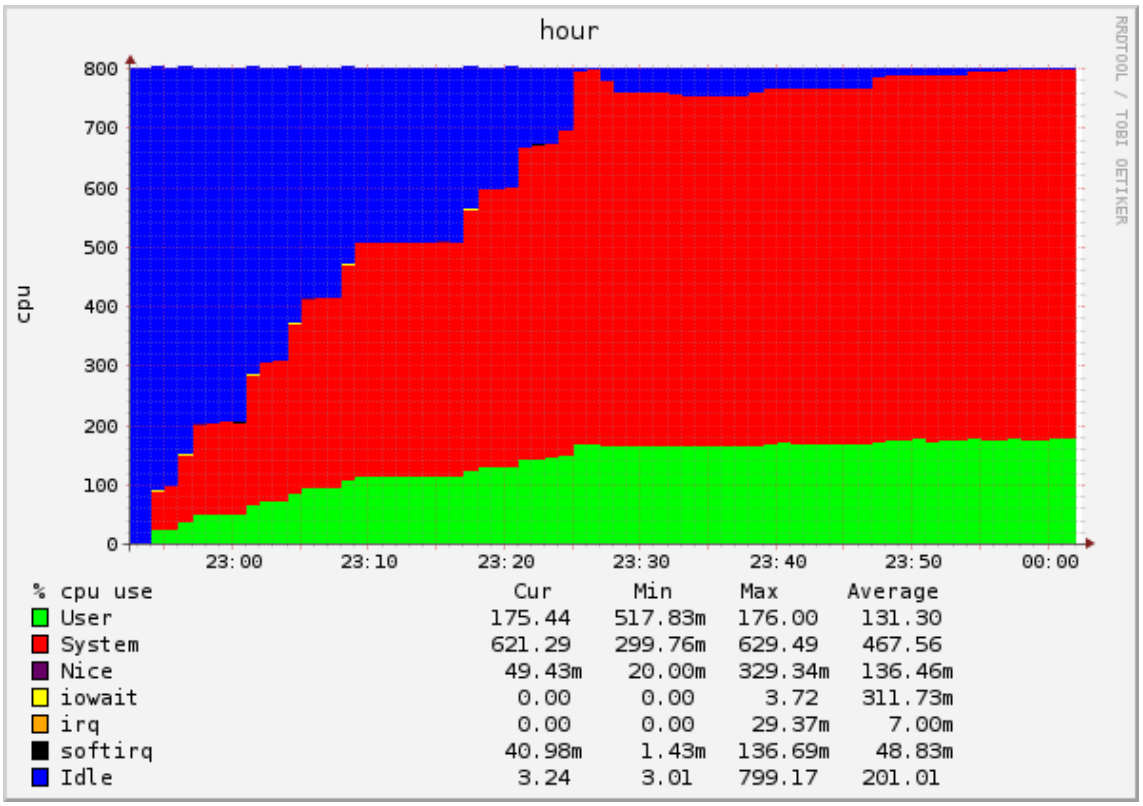

Figura 25 – Execução gradual das máquinas virtuais com recursos idênticos.

Com as 15 máquinas em execução e o uso de processamento saturado, o escalonador é forçado a dividir o consumo, conforme mostra a figura 26:

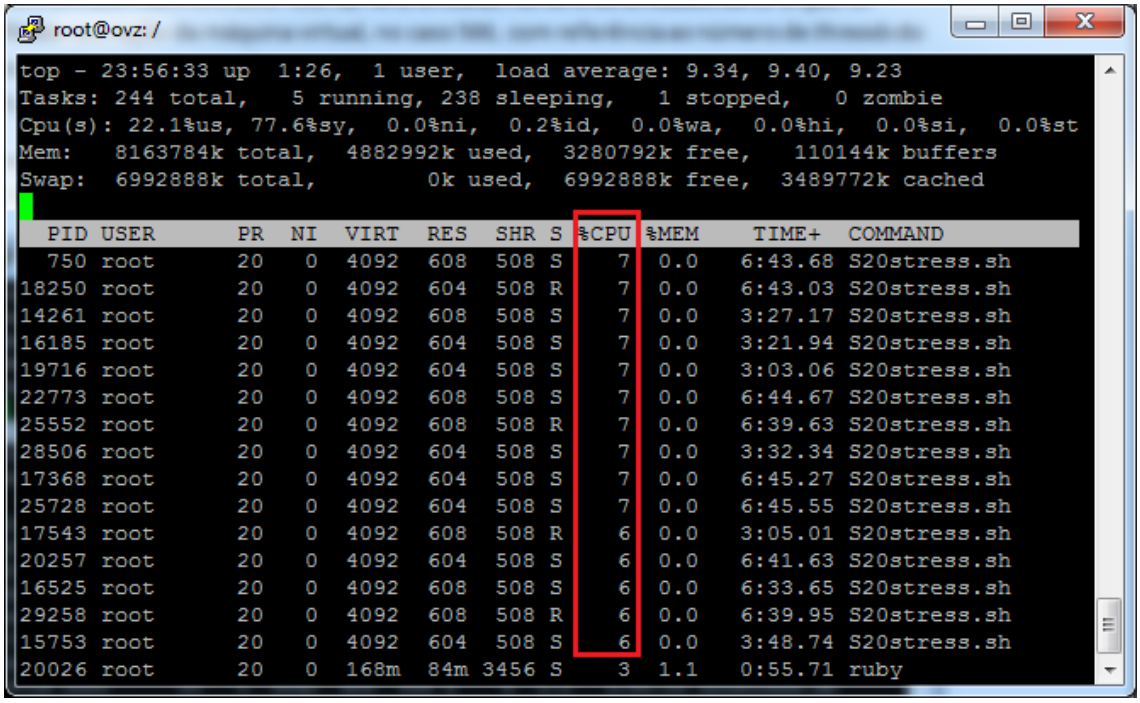

Figura 26 – O processamento é partilhado o mais igualmente possível entre os processos em execução, buscando manter o desempenho máximo para cada máquina.

O escalonador de CPU divide os recursos da melhor maneira possível, sem beneficiar uma máquina específica, justamente pelo valor de unidades de CPU definido na configuração inicial ser igual para todas as máquinas. Mesmo saturando o uso de processamento, a distribuição é preservada de forma a evitar que alguma máquina tenha mais benefícios que a outra, mesmo a 100% de uso. Para uma reserva mínima de 50% da CPU física destinada a uma única máquina virtual, por exemplo, deve ser utilizado o campo "Uso da CPU" do gerenciador, correspondente ao parâmetro *cpulimit*. Porém, da mesma forma que o *cpuunits*, só tem validade na concorrência de uso em um mesmo núcleo.

Para atestar esta conclusão, foram colocadas 16 máquinas virtuais com um total de 3000 unidades de CPU. A primeira ficou com 1500 unidades, enquanto as outras 15 com 100 unidades cada, com 50% de reserva mínima de CPU. No resultado (figura 27), a máquina com o privilégio de 1500 unidades utilizou um núcleo por inteiro, representado por 12%. As outras 15 máquinas dividiram-se entre os sete núcleos restantes utilizando em torno 6% cada. Ainda restou uma

pequena sobra de 3% de recursos ociosos disponíveis, mesmo considerando o uso pelo gerenciador OpenVZ.

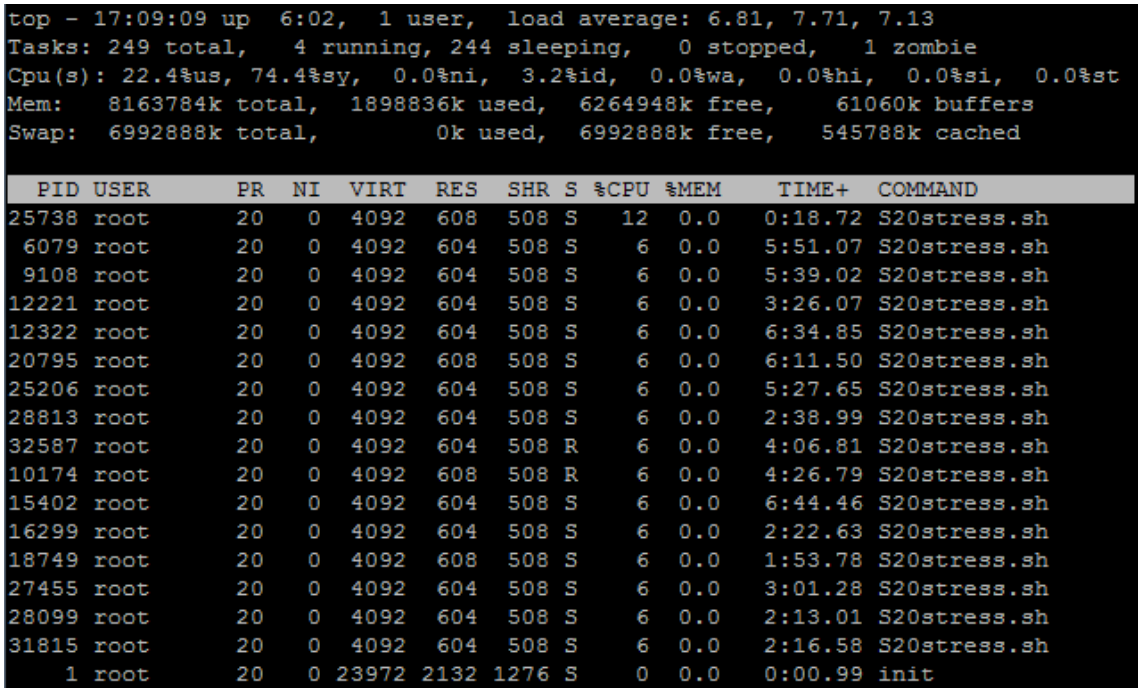

Figura 27 – Medição de uso dos recursos de CPU divididos entre 16 VMs, considerando 50% de uso para uma única máquina virtual.

De acordo com as propriedades do escalonador, caso fosse disponibilizado apenas um único núcleo físico, a divisão de processamento de 50% para a primeira máquina e o restante dividido entre as outras 15 ocorreria. Como a sua estrutura do escalonador de processos baseia-se prioritariamente em utilizar primeiro os núcleos físicos, somente depois que exceder o uso individual pelas máquinas virtuais é que a utilização começa a ser efetivamente repartida.

Em um segundo teste, as unidades de CPU por máquina virtual foram redistribuídas, definidas segundo a tabela 1, em uma forma de utilizar 90% da CPU física:

Máquina Virtual	Unidades de CPU	Porcentagem
Vm01	1050	15%
Vm02	1050	15%
Vm03	350	5%
Vm04	350	5%
Vm05	350	5%

Vm06	350	5%
Vm07	350	5%
Vm08	350	5%
Vm09	350	5%
Vm10	350	5%
Vm11	140	2%
Vm12	140	2%
Vm13	140	2%
Vm14	140	2%
Vm15	140	2%
Total:	**7000**	**90%**

Tabela 2 – Distribuição de Unidades de CPU entre as máquinas virtuais.

Com esta redistribuição, 10% foram calculados como sobra para o sistema e os outros 90% distribuídos entre as 15 máquinas virtuais, onde cinco delas tem direito a apenas 2% do processador, oito máquinas com direito a 5% e duas com direito a 15%. A mudança surtiu efeito, alocando os recursos das máquinas seguindo a dinâmica esperada, como mostra o resultado da figura 28:

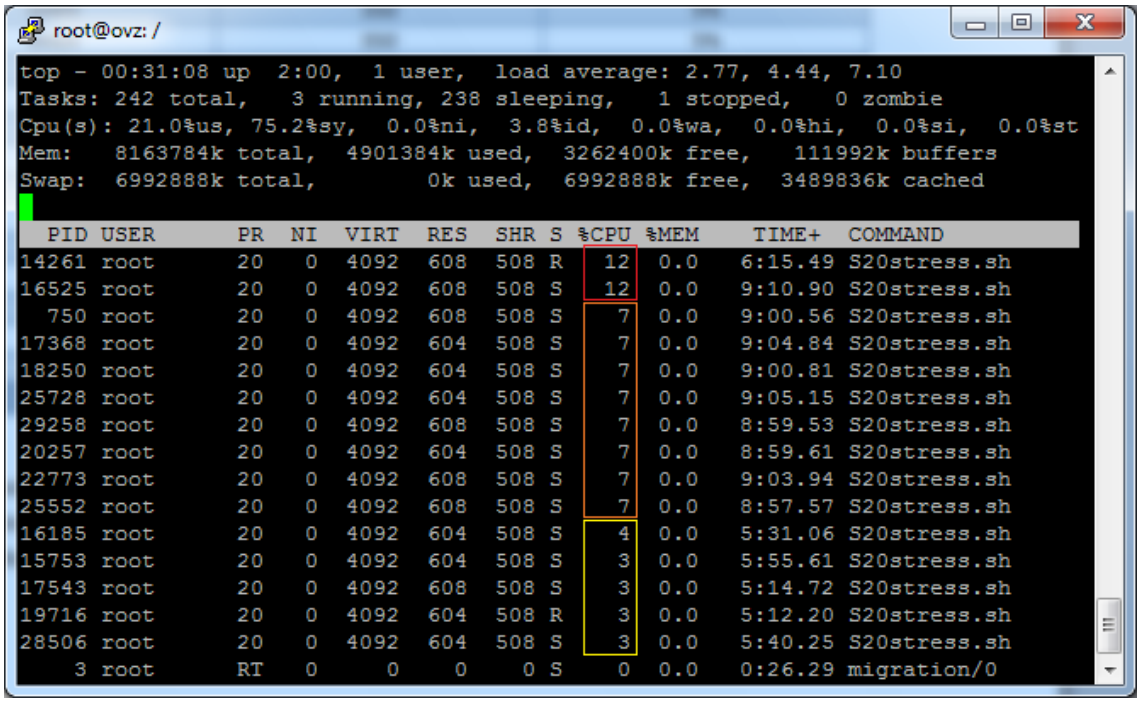

Figura 28 – Redistribuição das unidades de CPU de forma a atender os recursos reservados para cada máquina virtual. O somatório resulta em aproximadamente 96%, sendo que os outros 4% foram usados por outros recursos do servidor.

Após a alocação de recursos para cada máquina virtual de acordo com a tabela, o sistema comportou-se como o esperado, permitindo uma sobra de recursos para o sistema hospedeiro próxima aos 10% reservados. Como o OpenVZ atesta que o sistema virtualizador consome entre 1 e 3%, restaram aproximados 5% de recursos disponíveis. O gráfico abaixo mostra a variação do uso conforme mostra o gráfico na figura abaixo (figura 29):

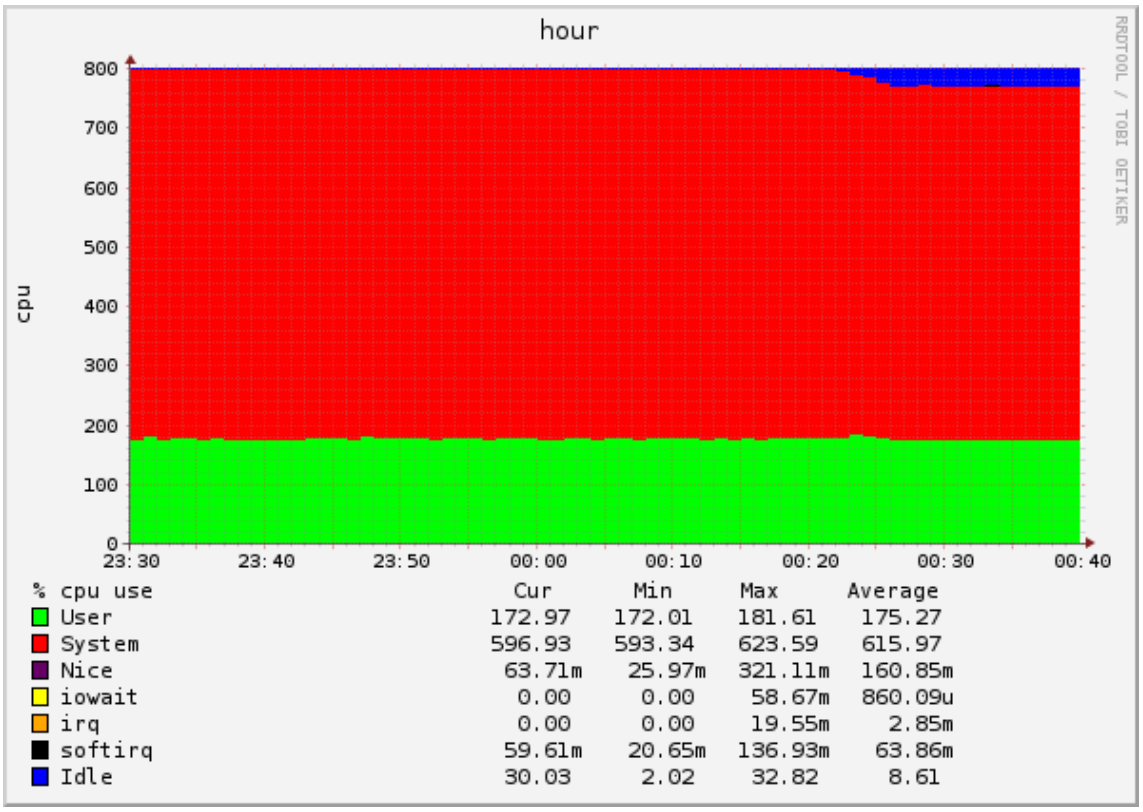

Figura 29 – Medição de uso do CPU pelas 15 máquinas virtuais com alocação de Unidades de CPU iguais em todas; Após realocar as unidades conforme a Tabela 1, a mudança surtiu efeito, resultando na sobra de recursos de CPU (cor azul).

Através da aplicação destes testes, foi possível entender o comportamento dos mecanismos de alocação de recursos de processamento, permitindo observar diferentes cenários. O escalonador de processos busca ocupar ao máximo os recursos disponibilizados, destinando-os de forma dinâmica e otimizada, respeitando as situações em que a utilização é concorrente e policiada.

Apesar dos cenários de testes serem situações bastante incomuns, foi possível ter uma noção de como o sistema se comporta no cumprimento da distribuição dos recursos, permitindo que os administradores tenham o poder de controlar e organizar a alocação de acordo com a demanda.

Os resultados medidos também afirmam que a virtualização em nível de sistema operacional possui um maior desempenho com relação às demais tecnologias, além de permitir um maior controle sobre os sistemas operacionais utilizados.

5.4 Implantação e Disponibilização do Serviço

O serviço de virtualização de sistemas operacionais deve oferecer aos usuários um sistema robusto e estável, permitindo utilizarem ambientes virtualizados com privilégios administrativos sem oferecer riscos à infraestrutura dos laboratórios e da universidade. Estes ambientes devem possibilitar os usuários praticarem diferentes atividades em um ambiente seguro, permitindo atividades práticas que envolvam permissões administrativas nos sistemas sem se preocupar em comprometer a integridade física das máquinas e sistemas operacionais.

Para atender este cenário, foi configurado um sistema mesclando as duas ferramentas: Proxmox VE e OVZ-Web-Panel. Com base nas análises feitas sobre os recursos de cada ferramenta, foram coletadas as seguintes informações com relação aos critérios apontados para estabelecer o serviço, identificando o que cada sistema oferece (tabela 2). Também foram identificadas outros mecanismos que podem ser considerados recursos complementares, observados durante a análise.

Critério	OVZ-Web-Panel	Proxmox VE
Desenvolvido em código livre	X	X
Interface gráfica para o utilizador	X	X

Permitir a criação de VMs Linux com acesso SSH e perfil administrativo	X	X
Recursos de isolamento, segurança e gerenciamento de performance	X	X
Opções de configuração dos ambientes de máquinas virtuais (processador, memória, disco, etc.)	X	X
Opções de modelos de máquinas virtuais prontos para utilização	X	X
Perfis de ambientes para instalação das maquinas virtuais pré-configurados	X	
Mecanismos de monitoramento do uso dos recursos	X	parcial
Recursos de autosserviço para o usuário	X	parcial
Documentação e tutoriais	X	X
Utilitários de acesso ao terminal da máquina virtual por uma interface gráfica	parcial	X
Perfil administrativo e de usuário	X	
Possibilidade de associar máquinas virtuais a determinados usuários	X	
Possibilidade de definir um 'tempo de vida' para as máquinas virtuais	X	
Mecanismo de atendimento e suporte ao usuário, com envio de mensagens ao administrador	X	
Criação de máquinas virtuais com *hypervisor* KVM		X
Migração de máquinas virtuais entre nós de servidores	parcial	X
Clonagem de máquinas virtuais	X	
Configuração de recursos de rede na interface administrativa (endereços IP, DNS, etc.)	X	X

Tabela 3 – Avaliação dos critérios das ferramentas.

Os critérios identificados como 'parciais' fazem comparação entre as duas ferramentas, sendo considerado completo em uma e limitado na outra. Por exemplo, o recurso de migração de máquinas virtuais no OVZ-Web-Panel demanda intervenção manual, enquanto no Proxmox VE é realizada de forma automatizada. Também foi observada uma vantagem maior na ferramenta de interface com a máquina virtual que o Proxmox VE oferece, permitindo um acesso facilitado tanto a máquinas sem interface gráfica quanto a máquinas somente acessíveis por uma interface

gráfica. No OVZ-Web-Panel, este acesso é oferecido com uma ferramenta simples, na forma de um terminal de comando, bastante limitada.

Como a ferramenta OVZ-Web-Panel possui compatibilidade com o Proxmox VE, foi possível configurar um cenário utilizando as duas ferramentas, permitindo que as vantagens oferecidas fossem unidas em um único sistema. O Proxmox VE oferece vários recursos que se destacam com relação ao OVZ-Web-Panel, porém, a ausência de um mecanismo de criação e acesso para usuários sem privilégios administrativos torna a ferramenta incompatível para servir de *front-end* para a proposta do sistema, dependendo do suporte desta para possibilitar o cenário.

Os *templates* de *containers* para virtualização no nível de sistema operacional com o OpenVZ são listados nas duas ferramentas, pois elas identificam o mesmo local do repositório. O mesmo ocorre com as máquinas virtuais criadas ou em execução, sendo exibidas e também gerenciadas nas duas interfaces. A compatibilidade se estende inclusive para os nós de servidores, onde é possível enxergar em qual nó se encontra uma determinada VM.

O OVZ-Web-Panel permite ao administrador criar perfis de usuários, delegando máquinas virtuais a cada usuário e estabelecendo também um tempo de vida para a mesma. A limitação desta ferramenta é de apenas prover máquinas virtuais Linux, virtualizadas pelo OpenVZ, pois a mesma foi desenvolvida apenas com este propósito. De certa forma, é uma excelente vantagem, pois restringe a utilização de sistemas e softwares privativos, o que resulta em uma maior dependência administrativa ao controlar estas instâncias.

A criação de máquinas virtuais de sistemas privativos ou incompatíveis com a virtualização pelo OpenVZ fica ao encargo do Proxmox VE, que suporta estas instâncias, virtualizando através do KVM. Apesar de a ferramenta não possuir perfis de usuário, este recurso pode ser oferecido, porém, disponibilizado de forma restrita, dependendo de um administrador para configurar o ambiente. Um administrador poderá atender a uma requisição, preparando uma máquina virtual

do Windows, por exemplo, e configurando uma ferramenta de acesso remoto para o usuário dentro do sistema. Infelizmente esta não é uma maneira simplificada de providenciar uma máquina virtual deste tipo, mas a possibilidade de atender estas requisições permite agregar esta vantagem ao serviço.

5.5 Premissas Complementares

Para viabilizar a implantação do serviço, é essencial desenvolver e avaliar alguns aspectos importantes antes de disponibilizá-lo para a comunidade. Estas devem ser observadas e trabalhadas em conjunto com os órgãos reguladores de serviços e infraestrutura de redes da universidade para estar de acordo com as normas estabelecidas.

5.5.1 Solicitação de Máquinas Virtuais

Como ambas as ferramentas não oferecem um sistema de cadastro externo, apenas com intervenção do administrador – no caso o OVZ-Web-Panel – é necessário elaborar uma página de apresentação sobre o serviço separada da interface de acesso. Esta deve conter informações para o usuário de como o serviço funciona e os procedimentos necessários para solicitar a criação de uma conta. Para isso, um formulário de cadastro pode ser desenvolvido, onde os usuários devem preencher um cadastro que será avaliado e aprovado pelo administrador. Este deverá ser feito utilizando um email válido da universidade, ou matrícula. Depois de cadastrado, o usuário deve fazer a requisição da máquina virtual através do serviço de requisições do OVZ-Web-Panel, informando detalhes como, por exemplo, a finalidade ou o vínculo com algum professor/disciplina. Informação como cota de espaço em disco e tipos de máquinas (classificadas por ‘poder de processamento’) podem também ser definidos como critérios para a requisição, todos necessitando aprovação pelo administrador.

5.5.2 Conexão e Segurança

A oferta do serviço pode ser limitada a utilização dentro da rede universitária, restringindo a conexão externa e oferecendo maiores garantias de segurança. Por outro lado, impõe barreiras quanto à disponibilidade do aluno para acessar fora da rede do campus em situações que se torna necessário. O OVZ-Web-Panel oferece recursos de conexão segura via SSL, além da própria segurança das conexões SSH para as máquinas virtuais, podendo ser disponibilizado externamente. Como segurança dos demais sistemas da universidade, é sugerida a utilização de uma faixa de endereços IP reservada e preferencialmente fechada, impedindo tentativas de intrusão.

5.5.3 Políticas de Utilização do Serviço

Para este serviço ser oferecido dentro da comunidade acadêmica, devem ser definidas regras explícitas de utilização aos usuários finais, a fim de conscientizá-los sobre os propósitos do recurso e os riscos que o mesmo pode vir a sofrer em caso de constatação de mau uso. Dentre estas informações, devem estar descritas as possíveis circunstâncias com referência as infrações que um usuário mal intencionado possa vir a cometer, como por exemplo: utilizar o recurso para disseminar pragas virtuais, emails em massa (*spam*), ataques e tentativas de invasão com propósitos prejudiciais, download de arquivos ilegais, entre outras. Estas informações devem ser apresentadas a cada usuário que solicitar uma máquina virtual, sendo obrigatório concordar com as políticas de utilização do serviço.

Outras necessidades para melhorar a oferta do serviço podem ser pesquisadas e apontadas durante a utilização, levantando informações e observações que possam contribuir com a sua consolidação.

5.6 Cenários e Vantagens da Utilização

Existem diversas atividades em que as máquinas virtuais podem ser utilizadas, envolvendo as mais variadas disciplinas ou aplicações no meio acadêmico, oferecendo vantagens em promover um ensino prático e mais qualificado.

Um suposto cenário para aplicar do uso de máquinas virtuais é a disponibilização de um *web service* em uma disciplina que envolva programação com intercomunicação entre objetos, como, por exemplo, Computação Distribuída, Desenvolvimento de Sistemas Orientados a Objetos ou Programação Web. O professor poderia configurar uma aplicação hospedada em uma máquina virtual para responder a determinadas requisições, como por exemplo, a consulta e o registro de informações em um banco de dados, simulando as mais variadas situações, buscando ampliar a experiência dos alunos com exercícios práticos de alto nível.

Em disciplinas como a de Sistemas de Informação Cliente/Servidor, grupos de alunos podem dispor de uma máquina virtual para desenvolver um *webservice*, instalando e configurando as ferramentas necessárias para desenvolver as atividades propostas, com resultados transparentes e claros sobre os desafios encontrados. O mesmo pode ser considerado em disciplinas de segurança da informação, com testes de validação de certificados digitais, experiências com ferramentas de quebra de segurança, ataques, interceptação de pacotes, entre outros, permitindo aos alunos presenciarem as mais variadas situações, estimulando o aprendizado.

Para desenvolver estudos com relação ao funcionamento dos sistemas operacionais e seus mecanismos, podem ser promovidas aulas práticas para analisar a arquitetura e comportamento, ampliando o entendimento a respeito dos elementos envolvidos. Podem ser utilizadas máquinas virtuais para estender a noção de como funciona o escalonamento de processos, gerência de

memória, estruturas e sistemas de arquivos, realizando atividades práticas onde os arquivos do sistema podem ser examinados profundamente e até modificados. Disciplinas como a de Sistemas Operacionais pode ser ramificada com conhecimentos mais específicos, por exemplo, o ensino de sistemas Unix/Linux, focando na utilização de interfaces de linha de comando. Em uma etapa mais avançada, os alunos podem aprender a modificar a *kernel* de um sistema operacional adicionando e removendo módulos, ampliando a compreensão sobre a estrutura e o funcionamento.

Para os laboratórios da universidade, as máquinas virtuais permitem que os alunos não fiquem dependentes dos computadores para atividades que requerem uma maior autonomia, habilitando a customização e instalação de ferramentas sem restrição nos ambientes virtuais, eliminando preocupações em prejudicar o sistema dos equipamentos. Essa vantagem aperfeiçoa a disponibilização dos laboratórios, permitindo que as estações fiquem disponíveis para todas as disciplinas sem contratempos, além de reduzir as manutenções por parte dos administradores dos laboratórios.

Na comunidade acadêmica, inúmeras aplicações podem ser desenvolvidas utilizando sistemas virtualizados, beneficiados pela facilidade em que estes recursos podem ser providos. Projetos científicos de pesquisa e experimentos podem ser embrionados utilizando sistemas virtualizados, agilizando processos e reduzindo a dependência de *hardware* ou espera por recursos. Novas disciplinas podem ser elaboradas e incluídas nos cursos, abrangendo a utilização de ferramentas e softwares em ambientes virtualizados, promovendo a cultura e os benefícios que sistemas de código livre podem oferecer.

5.7 Desafios e Obstáculos

Para a implantação de um serviço de virtualização, existem diversos critérios a serem analisados rigorosamente, com excepcional importância a questão de segurança e isolamento. O

sistema, em princípio, não deve oferecer riscos para a infraestrutura da universidade em todos os aspectos, principalmente com questão a sobrecarga de utilização da rede e a ameaças de segurança aos demais sistemas. Naturalmente, os sistemas virtualizados são oferecidos de forma isolada, providos e concentrados em um ou mais servidores que não compartilham outros sistemas, moderadamente inofensivos aos demais sistemas que se encontram concentrados em um mesmo parque computacional.

Toda cautela é válida na concepção do serviço, avaliando principalmente as regras de *firewall* e a utilização de certificados de segurança. O ideal é que o serviço seja disponibilizado em uma faixa de endereços IP separada, com restrições quanto ao acesso e uso de portas específicas aos servidores da universidade.

Outro obstáculo a ser considerado é a preparação dos alunos para utilizarem sistemas em Linux, e também para compreenderem a lógica de funcionamento de um ambiente virtualizado. A grande maioria pode nunca ter utilizado um sistema Unix, ainda mais em linha de comando, sendo muitas vezes um fator que desencoraja o uso por estarem habituados a utilizar ambientes com interface gráfica. A dependência de disciplinas ou planos de ensino que capacitem os alunos a trabalhar com sistemas do tipo é um fato a ser considerado, assim como os conceitos e o funcionamento de sistemas virtualizadores.

5.8 Conclusões e Resultados Obtidos

Durante a etapa de instalação e experimentos com as ferramentas propostas, foram encontradas dificuldades em preparar um ambiente com o *framework* Eucalyptus em condições de ser avaliado apropriadamente, com poucos resultados passíveis de serem analisados e comparados com outros sistemas. Boa parte dos recursos que o Eucalyptus oferece é voltada para

oferecer ambientes de computação em nuvem para grandes corporações, vinculados com softwares privativos e limitados para uso livre, desviando o foco com relação a um dos critérios abordados como base deste estudo. Com isso, optou-se por não utilizar a tecnologia, apesar de atender em boa parte os requisitos do ambiente proposto. Outro ponto a considerar é que o Eucalyptus não oferece recursos de integrar, em seu ambiente, as técnicas de virtualização no nível de sistema operacional, desmotivando o seu uso.

A busca por outras ferramentas como alternativa para a proposta inicial permitiu identificar excelentes utilitários, permitindo conceber um ambiente capaz de atender de maneira muito satisfatória os requisitos apontados, demonstrando com argumentos válidos para encorajar a implantação de um serviço de virtualização. Alguns aspectos, como o controle e registro de usuários, um acesso simplificado ao terminal de comando e também às máquinas virtualizadas por KVM podem ser aprimorados, visto que as ferramentas são desenvolvidas em código livre e também possuem uma comunidade de colaboradores altamente ativa.

A metodologia de virtualização no nível de sistema operacional comandada pelo OpenVZ correspondeu às expectativas nos testes de desempenho, apresentando resultados satisfatórios quanto à alocação de recursos entre as máquinas virtuais em várias situações, além de oferecer mecanismos para partilhar e regular a utilização em condições em que é demandado. Foi possível também perceber o baixo consumo do gerenciador, atestando à alta escalabilidade que a técnica possui. Como um dos objetivos deste estudo concentrou-se na utilização de um sistema de alto desempenho, a escolha da técnica também trouxe o benefício de restringir o uso de sistemas operacionais Linux, desenvolvidos em código livre, assim como o próprio *framework*.

6. Considerações Finais e Trabalhos Futuros

A consolidação de servidores em ambientes virtuais e a oferta de recursos e serviços de computação em nuvem modificaram praticamente por completo os rumos da computação moderna. Cenários previstos nos primórdios da computação por mentes visionárias tornaram-se uma nova realidade, com ofertas de serviços computacionais por demanda acessíveis em larga escala e disponibilidade. Neste meio, diversas tecnologias e conceitos embrionaram, proporcionando novos modelos de serviços onde às preocupações com os recursos físicos se abstraem praticamente por completo, permitindo desenvolver novos paradigmas computacionais.

Foi possível, no decorrer deste trabalho, analisar ferramentas de virtualização robustas e maduras que nos permitem prover ambientes de alto desempenho e escalabilidade com pouco investimento e resultados extremamente satisfatórios. Os inúmeros atrativos que estas tecnologias oferecem fazem tanto da virtualização quanto da computação em nuvem insumos fundamentais para desenvolvimento tecnológico em diversas organizações, especialmente em instituições de ensino.

Com os resultados deste estudo, foi possível reunir informações e recursos em busca de tornar viável um serviço de virtualização em nuvem capaz de oferecer meios de incentivar o ensino prático em cursos de computação e tecnologia da informação, proporcionando experiências construtivas e um aprendizado diferenciado. Não apenas buscando promover o ensino, mas também para servir a comunidade científica com meios para facilitar o desenvolvimento de trabalhos que demandam recursos computacionais supríveis com sistemas virtualizados.

Através da utilização de duas metodologias de virtualização integradas, foi possível organizar um sistema versátil e poderoso, unindo o desempenho sólido da virtualização no nível

de sistema operacional para instâncias em Linux e a possibilidade de utilizar máquinas virtuais de sistemas privados por *hypervisors*. As ferramentas estudadas para promover o serviço mostraram-se desenvolvidas e capazes de atender a proposta, com poucas ressalvas quanto à facilidade em tornar o serviço mais acessível e autônomo. Estas não demonstraram ser um obstáculo para a implantação do serviço, pois os mesmos podem ser contornados inicialmente com medidas simples, permitindo que ele possa ser avaliado e aprimorado conforme o crescimento.

Como trabalhos futuros, existem condições de desenvolver as ferramentas do ambiente proposto, buscando agregar vias de consolidar o serviço de forma robusta e autônoma, minimizando a dependência do administrador para gerenciar os recursos. Apesar do OpenVZ-Web-Panel oferecer uma interface rica em recursos de customização, são visíveis algumas modificações e implementações que podem adaptar melhor a ferramenta para utilização na Internet, permitindo que os usuários possam se cadastrar diretamente, minimizando a intervenção de um responsável. Como o Proxmox oferece uma solução híbrida de virtualização, com excelentes recursos, mostrando-se compatível com o OpenVZ-Web-Panel, a fusão das ferramentas pode oferecer um sistema poderoso e flexível, capaz de atender demandas por sistemas não suportados pelo OpenVZ, ampliando a oferta de serviços.

7. Referências Bibliográficas

CARISSIMI, Alexandre. **Virtualização: da Teoria a Soluções.** Universidade Federal do Rio Grande do Sul (UFRGS), Instituto de Informática, Porto Alegre – RS, 2008. Disponível em: < http://www.gta.ufrj.br/ensino/CPE758/artigos-basicos/cap4-v2.pdf>. Acesso em: 5 mar. 2011.

GOLDEN, Bernard; SCHEFFY, Clark. **Virtualization for Dummies, Sun AMD Special Edition.** Indianapolis: Wiley Publishing INC, 2008. Disponível em: <http://www.amd.com/us/Documents/Virt_for_Dummies.pdf>. Acesso em 23 abr. 2011.

KOLYSHKIN, Kirill. **Virtualization in Linux**. 2006. 5 f. Disponível em: <http://download.openvz.org/doc/openvz-intro.pdf>. Acesso em: 23 abr. 2011.

SILVA, Rodrigo Ferreira da. **Virtualização de Sistemas Operacionais.** 2007. 114 f. Monografia (Graduação em Tecnologia da Informação e da Comunicação) - ISTCC, Petrópolis - Rj, 2007. Disponível em: <http://www.lncc.br/~borges/doc/Virtualizacao de Sistemas Operacionais.TCC.pdf>. Acesso em: 23 abr. 2010.

MATHEWS, Jeanna N.; DOW, Eli M.. **Executando o Xen.** Alta Books, 2006. 616 p. Disponível em: <http://altabooks.tempsite.ws/capitulos_amostra/xen.pdf>. Acesso em: 23 abr. 2011.

WORLD CONGRESS ON ENGINEERING, 2009, London, U.K.. **Implementation of a Purely Hardware-assisted VMM for x86 Architecture** [S.I.], 2009. 5 p. Disponível em: <http://www.iaeng.org/publication/WCE2009/WCE2009_pp136-140.pdf>. Acesso em: 23 abr. 2011.

WIKIPEDIA, THE FREE ENCYCLOPEDIA. **Operating system-level virtualization.** Disponível em: <http://en.wikipedia.org/wiki/Operating_system-level_virtualization>. Acesso em: 15 mai. 2011.

Parallels Virtuozzo Containers. Parallels Holdings Ltd.. Disponível em: <http://www.parallels.com/products/pvc/>. Acesso em: 14 ago. 2011.

FISHER-OGDEN, John. **Hardware Support for Efficient Virtualization.** University of California, San Diego. Disponível em: <http://cseweb.ucsd.edu/~jfisherogden/hardwareVirt.pdf>. Acesso em: 25 jun. 2011.

POPEK, Gerald J.; GOLDBERG, Robert P.. Formal Requirements for Virtualizable Third Generation Architectures. **Communications of the ACM.**, New York, n. , p. 412-421, 15 out. 1973. Disponível em: <http://www-users.cselabs.umn.edu/classes/Spring-2010/csci5105/papers/popek-virt-reqmts.pdf>. Acesso em: 25 jun. 2011.

WIKIPEDIA, THE FREE ENCYCLOPEDIA. **Hypervisor.** Disponível em: < http://en.wikipedia.org/wiki/Hypervisor>. Acesso em: 15 mai. 2011.

REIS, Wanderson Santiago dos. **Virtualização de Serviços Baseado em Contêineres**: Uma Proposta para Alta Disponibilidade de Serviços em Redes Linux de Pequeno Porte. 2009. 54 f. Monografia (Pós-graduação) - UFLA, Lavras - MG, 2009.

WIKIPEDIA, THE FREE ENCYCLOPEDIA. **Utility Computing.** Disponível em: < http://en.wikipedia.org/wiki/Utility_Computing>. Acesso em: 19 mai. 2011.

WIKIPEDIA, THE FREE ENCYCLOPEDIA. **Grid Computing.** Disponível em: < http://en.wikipedia.org/wiki/Grid_Computing>. Acesso em: 12 ago. 2011.

WIKIPEDIA, THE FREE ENCYCLOPEDIA. **Cloud Computing.** Disponível em: < http://en.wikipedia.org/wiki/Cloud_computing>. Acesso em: 12 ago. 2011.

WIKIPEDIA, THE FREE ENCYCLOPEDIA. **Cloud Computing.** Disponível em: < http://en.wikipedia.org/wiki/John_McCarthy_(computer_scientist)>. Acesso em: 12 ago. 2011.

IT COMPASS – Fujitsu (Australia). **Three questions for the man who coined the term "cloud computing". 2010.** Disponível em: <http://it-compass.au.ts.fujitsu.com/_#Three-questions-for-the-man-who-coined-the-term-cloud-computing-3024860.html>. Acesso em: 12 ago. 2011.

MELL, Peter; GRANCE, Timothy. **The NIST Definition of Cloud Computing:** Recommendations of the National Institute of Standards and Technology. 07 out. 2009. Disponível em: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>. Acesso em: 19 ago. 2011.

SOUSA, Flávio R.C.; MOREIRA, Leonardo O.; MACHADO, Javam C. **Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios.** Universidade Federal do Ceará. ERCEMAPI 2009. Disponível em: < http://www.es.ufc.br/~flavio/files/Computacao_Nuvem.pdf>. Acesso em 12 ago. 2011.

SCHULLER, Sinclair. **Demystifying The Cloud: Where Do SaaS, PaaS and Other Acronyms Fit In?** 2008. Disponível em: <http://www.saasblogs.com/saas/demystifying-the-cloud-where-do-saas-paas-and-other-acronyms-fit-in/>. Acesso em: 12 ago. 2011.

ARMBRUST, Michael et al. **Above the Clouds: A Berkeley View of Cloud Computing.** University Of California, Berkeley: EECS Department, 2009. 25 p. Disponível em: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>. Acesso em: 19 mai. 2011.

CLOUD SECURITY ALLIANCE. **Cloud Computing Architectural Framework.** Disponível em: <https://wiki.cloudsecurityalliance.org/guidance/index.php/Cloud_Computing_Architectural_Framework>. Acesso em: 12 ago. 2011.

GOMES, Silvia Bogéa; FALBO, Ricardo de Almeida; MENEZES, Crediné Silva de. **Um Modelo para Acordo de Nível de Serviço em TI.** 2005. 15 f. Artigo (Mestrado) - Ufes, Vitória - Es, 2005. Disponível em: <http://www.inf.ufes.br/~falbo/download/pub/2005-Sbqs.pdf>. Acesso em: 19 ago. 2011.

IT-TUDE. **Service Level Agreements.** Disponível em: <http://www.it-tude.com/sla-article.html>. Acesso em: 04 set. 2011.

MARSHALL, Bryan. Using VmWare vCenter to Teach System Administration in a Lab. **Issues In Information Systems**, [s.i.], v. 12, n. 2, p.153-171, 2011. Disponível em: <http://iacis.org/iis/2011_iis/number_2/153-161_AL2011_1681.pdf>. Acesso em: 19 ago. 2011.

BOWER, Timothy. **EXPERIENCES WITH VIRTUALIZAION TECHNOLOGY IN EDUCATION**. The Journal Of Computing Sciences In Colleges, Kansas, n. , p.311-318, 05 maio 2010. Disponível em: <http://www.salina.k-state.edu/faculty/tim/scholarship/using_virtualization.pdf>. Acesso em: 26 set. 2011.

EUCALYPTUS, The Open Source Cloud Platform. Eucalyptus Inc.. Disponível em: < http://open.eucalyptus.com/>. Acesso em: 13 abr. 2011.

WIKIPEDIA, THE FREE ENCYCLOPEDIA. **Eucalyptus (Computing).** Disponível em: < http://en.wikipedia.org/wiki/Eucalyptus_(computing)>. Acesso em: 13 abr. 2011.

OpenVZ, Linux Containers. Parallels Holdings Ltd.. Disponível em: < http://openvz.org> Acesso em: 6 jun 2011.

WIKIPEDIA, THE FREE ENCYCLOPEDIA. **Daemon (Computing).** Disponível em: < http://en.wikipedia.org/wiki/Daemon_(computing)>. Acesso em: 23 abr. 2011.

HEWLETT PACKARD. **Performance Evaluation of Virtualization Technologies for Server Consolidation.** University Of Michigan, 2008. 13 p. Disponível em: <http://www.hpl.hp.com/techreports/2007/HPL-2007-59R1.pdf>. Acesso em: 29 jul. 2011.

PROXMOX, Virtual Environment. Proxmox Server Solutions GmbH. Disponível em: < http://pve.proxmox.com>. Acesso em: 14 ago. 2011.

PROXMOX, Virtual Environment. Proxmox Server Solutions GmbH. Disponível em: < http://pve.proxmox.com>. Acesso em: 14 ago. 2011.

OpenVZ Web Panel. Disponível em: < http://code.google.com/p/ovz-web-panel/>. Acesso em 6 jun. 2011.

ANEXO I – Configurando o OpenVZ no Ubuntu Server 10.04

Como o OpenVZ não foi mantido nas versões posteriores a 8.04 do Ubuntu (Hardy Heron), é necessário compilar um *kernel* estável fornecido pela comunidade Wiki do OpenVZ. A versão 8.04 do Ubuntu também não oferece suporte ao Eucalyptus, o que fez necessária a preparação do ambiente na versão 10.04 para a realização do experimento. Outro aspecto a observar é a instalação do sistema utilizando o *filesystem Ext3*, pois o *Ext4* pode apresentar instabilidade na gerência de cotas dos containers do OpenVZ.

A instalação do ambiente foi realizada sobre um sistema x86_64.

1. Após instalar o Ubuntu 10.04 Server, atualize o sistema. Após, reinicialize-o caso tenha atualizações no kernel:

```
# sudo apt-get update
# sudo apt-get upgrade
# sudo apt-get upgrade
```

2. Em modo *root*, atualize a shell atual. Ao perguntar para instalar a dash como /bin/sh, selecione “no”:

```
# dpkg-reconfigure dash
```

3. Desabilitar a extensão de segurança “AppArmor” (opcional):

```
# /etc/init.d/apparmor stop
# update-rc.d -f apparmor remove
# apt-get remove apparmor apparmor-utils
```

4. Instalar os pacotes necessários para recompilar a Kernel:

```
# apt-get install kernel-package libncurses5-dev fakeroot
wget bzip2 module-assistant debhelper build-essential
```

5. Verificar a versão da Kernel atualmente instalada:

```
# uname -r
```

6. Montar as dependências da Kernel atual no sistema utilizando como parâmetro a exata versão instalada (exibida no comando anterior):

```
# apt-get build-dep --no-install-recommends linux-image-
2.6.32-30-server
```

7. Preparar os *headers* e criar a configuração de compilação do Kernel

```
# sudo m-a prepare
# sudo kernel-packageconfig
```

8. Otimizar o uso de múltiplos núcleos (o número deve ser o total de núcleos físicos + 1). O *script* abaixo realiza o necessário:

```
# Cores=$(Nr () { echo $#; }; Nr $(grep "processor" /proc/cpuinfo |
cut -f2 -d":"))
# echo "CONCURRENCY_LEVEL := $(($Cores + 1))" | sudo tee -a
/etc/kernel-pkg.conf
```

9. Fazer o download do código fonte da Kernel do sistema atual:

```
# cd /usr/src
# wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-
2.6.32.tar.bz2
```

10. Fazer o download dos patches do OpenVZ para preparar a Kernel (lembrando que deve ser utilizada a versão "current", considerada a mais estável, localizada no endereço http://download.openvz.org/kernel/branches/2.6.32/current/patches/):

```
# wget
http://download.openvz.org/kernel/branches/2.6.32/current/pat
ches/patch-feoktistov.1-combined.gz
```

11. Fazer o download dos arquivos de configuração relacionados ao ambiente em que está sendo realizada a instalação, localizado no endereço http://download.openvz.org/kernel/branches/2.6.32/current/configs/ :

```
# wget
http://download.openvz.org/kernel/branches/2.6.32/current/con
figs/kernel-2.6.32-x86_64.config.ovz
```

12. Descompactar o código-fonte do Kernel baixado e prepará-lo para ser compilado, aplicando o patch do OpenVZ e o arquivo de configuração:

```
# tar –xpf linux-2.6.32.tar.bz2
# mv linux-2.6.32       linux-2.6.32-openvz
# rm –f linux
# ln –s linux-2.6.32-openvz   linux
# cd linux
# unzip –dc ../patch-feoktistov.1-combined.gz | patch –p1
# cp –rf ../kernel-2.6.32-x86_64.config.ovz  .config
# make oldconfig
```

13. Corrigir um bug, alterando a uma linha no arquivo “Documentation/lguest/Makefile”, evitando um erro de compilação:

```
all: lguest
clean:
```

Altere para:

```
all:
clean:
```

14. Executar a compilação do Kernel (levará algum tempo até concluir):

```
# cd /usr/src/linux
# sudo make-kpkg --initrd --append-to-version=-openvz --
revision=1 kernel_image kernel_headers
```

15. Verificar os pacotes .deb gerados. Se corretos, exibirá dois arquivos conforme a seguir:

```
# cd ..
# ls  - *.deb

linux-headers-2.6.32.30-openvz_1_amd64.deb
linux-image-2.6.32.30-openvz_1_amd64.deb
```

16. Instalar a nova Kernel:

```
# sudo dpkg –i linux-image-2.6.32.30-openvz_1_amd64.deb
# sudo dpkg –i linux-image-2.6.32.30-openvz_1_amd64.deb
```

17. Atualizar o *initrd* e a configuração de *boot* do GRUB para definir a nova Kernel na inicialização (será definida como a Kernel principal no menu de inicialização):

sudo mkinitramfs -k 2.6.32.30-openvz -o /boot/initrd.img-2.6.32.30-openvz
sudo update-grub

18. Alterar a configuração do arquivo /etc/sysctl.conf para uso do OpenVZ, inserindo os parâmetros conforme a seguir:

```
( ... )
net.ipv4.conf.default.forwarding=1
net.ipv4.conf.default.proxy_arp = 1
net.ipv4.ip_forward=1
net.ipv4.conf.all.rp_filter = 1
kernel.sysrq = 1
#net.ipv4.tcp_ecn = 0
net.ipv4.conf.default.send_redirects = 1
```

```
net.ipv4.conf.all.send_redirects = 0
```
(fim do arquivo sysctl.conf)

19. Aplicar as mudanças feitas no sysctl.conf e instalar as ferramentas de gerência do OpenVZ:

```
# sudo sysctl -p
# sudo apt-get install --no-install-recommends vzctl vzquota
vzdump
```

20. Crie um link simbólico para eliminar problemas de compatibilidade e atualize o arquivo de inicialização do sistema para iniciar o OpenVZ:

```
# ln –s /vz    /var/lib/vz
```

21. No arquivo /etc/vz/vz.conf , altere o parâmetro de “NEIGHBOUR_DEVS” para ‘all’, evitando problemas de rede com as VM’s. Caso tenha feito a instalação sobre o sistema de arquivos Ext4, altere o parâmetro de “DISK_QUOTA” para ‘no’, evitando problemas de criação dos containers.

22. Finalmente, reinicialize o sistema com a nova Kernel.

```
# sudo reboot
```

23. Se tudo estiver OK, verifique a Kernel em uso. Deverá exibir conforme a seguir:

```
# sudo uname –r

2.6.32.30-openvz
```

24. Verifique se o serviço do OpenVZ está ativo. Se exibir a mensagem “openvz is running...”, está tudo OK:

```
# service vz status
```

25. Concluída a instalação do OpenVZ no Ubuntu 10.04.