



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ
CAMPUS LUIZ MENEGHEL - CENTRO DE CIÊNCIAS TECNOLÓGICAS
SISTEMAS DE INFORMAÇÃO

JULIO CESAR JARDIM PEREIRA

ESTUDO DE CASO:
Análise e Proposta de Implantação de Computação
em Nuvem para uma Editora Gráfica

Bandeirantes

2015

JULIO CESAR JARDIM PEREIRA

ESTUDO DE CASO:
**Análise e Proposta de Implantação de Computação
em Nuvem para uma Editora Gráfica**

Trabalho de Conclusão de Curso submetido à
Universidade Estadual do Norte do Paraná,
como requisito parcial para obtenção do grau
de Bacharel em Sistemas de Informação.

Orientador: Prof. Me. Luiz Fernando Legore
do Nascimento

Bandeirantes

2015

JULIO CESAR JARDIM PEREIRA

ESTUDO DE CASO:

**Análise e Proposta de Implantação de Computação
em Nuvem para uma Editora Gráfica**

Trabalho de Conclusão de Curso submetido à
Universidade Estadual do Norte do Paraná,
como requisito parcial para obtenção do grau
de Bacharel em Sistemas de Informação.

COMISSÃO EXAMINADORA

Prof. Me. Luiz Fernando Legore do
Nascimento
UENP – *Campus* Luiz Meneghel

Prof. Me. Glauco Carlos Silva
UENP – *Campus* Luiz Meneghel

Prof. Me. Rodrigo Tomaz Pagno
UENP – *Campus* Luiz Meneghel

Bandeirantes, 1º de Setembro de 2015.

A todos aqueles que de alguma forma contribuíram para que a conclusão deste trabalho se tornasse possível.

AGRADECIMENTOS

Primeiramente a Deus que me ajudou nos momentos mais difíceis, nas madrugadas, nos momentos de angústia.

Aos meus amigos em especial, Guilherme, José Mário, Paulo Cesar, Alexandre e Andressa que me auxiliaram durante o curso, nos momentos tristes e felizes.

A banca avaliadora, em especial meu orientador Luiz Fernando, pela dedicação, paciência e atenção em todos os momentos, sendo o principal responsável pelos créditos obtidos pelo projeto.

A minha família em especial, minha mãe Cleonice e meu pai João Lobo (*in memoriam*) e meus irmãos Lígia e João Carlos que sempre estiveram do meu lado nos momentos mais difíceis de minha vida.

"A vitória se alcança com a conjugação de esforço e com o equilíbrio da mente com o corpo."

(William Douglas)

RESUMO

Quando se fala em computação em nuvem, fala-se na possibilidade de acessar arquivos e executar diferentes tarefas via rede Internet. Considerando que não haja problemas com a conexão de rede internet, a disponibilidade dos serviços tem sido fortemente ampliada. Além disso, tornou-se possível obter maior flexibilidade e escalabilidade dos serviços oferecidos. Para muitas empresas, a computação em nuvem é um excelente meio o qual permite distribuir suas aplicações e dados com constantes atualizações tecnológicas. Para que a computação em nuvem seja funcional é necessário que a escolha pelo tipo de nuvem a ser contratada ou implantada esteja de acordo com as necessidades das empresas ou pessoas físicas. Neste trabalho é apresentada uma proposta para a construção de uma nuvem computacional privada a qual atenda diretamente as necessidades de empresa específica do setor gráfico.

Palavras-chave: Computação em Nuvem; Ferramentas de Construção de Nuvem.

ABSTRACT

When it comes to cloud computing, speaks on the ability to access files and perform different tasks via Internet. Whereas there are no problems with the internet network connection, the availability of services has been greatly expanded. In addition, it became possible to obtain greater flexibility and scalability of services. For many companies, cloud computing is an excellent way which allows you to distribute your applications and data with constant technological updates. For cloud computing to be functional it is necessary to choose the cloud type to be hired or deployed conforms to the needs of companies or individuals. This paper presents a proposal to build a private cloud computing which directly meets the specific business needs of the printing industry.

Keywords: *Cloud computing; Cloud Construction Tool.*

LISTA DE ABREVIATURAS E SIGLAS

AMQP	Protocolo Avançado de Filas de Mensagens.
API	Interface de Programação de Aplicativos.
CPU	Unidade Central de Processamento.
DAAS	Banco de Dados como Serviço.
EC2	Plataforma Central de Elasticidade.
IAAS	Infraestrutura como Serviço.
IETF	Força Tafera de Engenheiro da Internet.
IPSEC	Protocolo de Segurança IP.
OVF	Formato de Virtualização Aberto.
PAAS	Plataforma como Serviço.
PDF	formato de documento portátil.
RAW	Formatos de Arquivos de Imagens Digitais.
RNP	Rede Nacional de Ensino e Pesquisa.
SIFT	Sistema Desenvolvimento de Tolerância a Falha.
SSL	Segurança da Camada de Transporte.
S3	Simple Armazenamento de Serviço.
TAAS	Ensaio como Serviço.
TI	Tecnologia da Informação.
VDI	Imagem de Disco Virtual.
VHD	Disco Rígido Virtual.
VMDK	Disco Virtual.
VPN	Rede Privada Virtual.
WSDL	Descrição de Linguagem XML.

LISTA DE ILUSTRAÇÕES

Figura 2.1: Modelo de 3 universos: falha, erro e defeito.....	16
Figura 2.2: Clusters em alta disponibilidade.....	20
Figura 2.3: Modelo de bando de dados distribuídos.....	23
Figura 2.4: Topologia de redes VPN.....	29
Figura 3.1: Arquitetura do <i>OpenStack</i>	42
Figura 3.2: Arquitetura do <i>Eucalyptus</i>	46
Figura 5.1: Cenário atual da empresa.....	50
Figura 5.2: Tunelamento entre filiais e matriz.....	51
Figura 5.3: Cenário de cópia dos sistemas e arquivos.....	54
Figura 5.4: Cenário proposto.....	55

LISTA DE TABELAS

Tabela 2.1: Resumo dos principais atributos de dependabilidade.....	18
Tabela 2.2: Comparativo entre protocolos de segurança.....	28
Tabela 3.1: Tabela comparativa das ferramentas <i>clouds</i>	47

SUMÁRIO

1.	INTRODUÇÃO	13
1.1	Formulação e Escopo do Problema	14
1.2	Justificativa	14
1.3	Objetivo.....	15
1.3.1	Objetivos Específicos.....	15
1.4	Organização do Trabalho	15
2.	FUNDAMENTAÇÃO TEÓRICA.....	16
2.1	Conceitos Clássicos: Falha, Erro e Defeito.....	16
2.2	Tolerância a falhas	17
2.3	Dependabilidade	18
2.4	Cluster	19
2.5	Sistemas Distribuídos.....	21
2.6	Aplicações de Sistemas Tolerantes a Falhas.....	23
2.7	Tolerância a Falhas em Sistemas Distribuídos	24
2.8	Recuperação em sistemas distribuídos	25
2.9	Rede Privada Virtual	27
2.10	Computação em Nuvem.....	29
2.11	Serviços da Nuvem	31
2.12	Software as a Service (SaaS)	32
2.13	Tipos de Nuvens.....	34
2.14	Nuvem privada	34
2.15	Nuvem Híbrida	35
2.16	Nuvem Pública	36
2.17	Nuvem Comunidade	36
2.17	Considerações Finais.....	37
3	CONSTRUÇÃO DA NUVEM.....	38
3.1	Sistemas Construtoras de Nuvens.....	38
3.1.1	<i>OpenStack</i>	38
3.1.2	<i>CloudStack</i>	42
3.1.3	<i>OpenNebula</i>	44
3.1.4	<i>Eucalyptus</i>	45

3.2	Diferenças entre as ferramentas	47
3.3	Considerações Finais	48
4	METODOLOGIA.....	49
5	DESENVOLVIMENTO.....	50
5.1	Cópia das Máquinas Virtuais e Arquivos.....	52
5.2	Localidade das Cópias	54
5.3	Backup Fora do Ambiente da Nuvem	54
5.4	Diagrama Proposto	55
6	ANÁLISE DO DIAGRAMA PROPOSTO.....	56
6.1	Análise de possíveis falhas	57
7	CONCLUSÃO	60
7.1	Trabalhos Futuros	60
	REFERÊNCIAS.....	61

1. INTRODUÇÃO

Quando se fala em computação em nuvem (*cloud computing*), fala-se na possibilidade de acessar arquivos e executar diferentes tarefas pela internet, isto é, os mais diferentes serviços ficam disponíveis de forma online. Assim, os arquivos e aplicações não mais precisam ficar restritos aos computadores domésticos ou aos grandes servidores presentes em uma empresa. Esse modelo foi desenvolvido tendo como seu principal objetivo entregar serviços de fácil acesso, custo baixo com a garantia que haja disponibilidade e escalabilidade.

A computação em nuvem é utilizada atualmente para reduzir os custos operacionais e revertê-los em novos investimentos. As áreas de negócio podem tirar vantagem desta mudança na alocação de novos recursos da TI, para atividades de valor agregado que tragam benefícios para o negócio e servir como base para inovação de negócios e redução de riscos potenciais.

Uma nuvem computacional remete à utilização de serviços de infra-estrutura, ou seja, de *software* e *hardware* provindos de um ambiente capaz de fornecer recursos virtualizados ilimitados e de forma escalável. Esses recursos são, no entanto transparente ao usuário final, ou seja, não é visível. O usuário tem a impressão de estar utilizando um serviço local ou disponibilizado em determinado local específico, contudo o que realmente ocorre é que esses recursos estão espalhados em diversos equipamentos disponíveis na rede mundial, compondo uma nuvem.

Essa composição feita por um conjunto de *datacenters* que, juntos, tornam o possível os recursos, processamento e armazenamento, que trabalham de forma conjunta, permitindo grande elasticidade e disponibilidade, sendo essas as principais características da computação em nuvem.

A computação em nuvem, no entanto é um tema que ainda gera muitas dúvidas no momento da escolha da melhor tecnologia a ser empregada para a hospedagem e armazenamento de dados dos sistemas utilizados pelas empresas. Tanto a nuvem (*Cloud*) quanto o uso do Software como Serviço (*SaaS*) poderão utilizar-se de *datacenters* próprios ou de terceiro para o armazenamento do seu software.

1.1 Formulação e Escopo do Problema

Para que uma empresa de grande porte que possui inúmeras filiais possa crescer e seus serviços possam ser entregues com agilidade, necessita de ambientes que facilitam essa troca de informação entre qualquer localidade.

Hoje o ambiente da empresa analisada fica preso à troca de arquivos com seus clientes e suas filiais através do protocolo FTP, mesmo sendo uma ferramenta útil, o serviço às vezes atrasa por depender do tempo dos funcionários em disponibilizar os arquivos. Os serviços das filiais, por exemplo, podem atrasar por não ter acesso aos arquivos enviados pelos clientes, pois hoje para que eles tenham acesso às imagens o funcionário responsável na matriz necessita disponibilizar tais os arquivos no FTP. E também outro ponto importante é se uma falha na matriz acontecer as filiais não terão acessos os arquivos de trabalho.

1.2 Justificativa

A empresa Alpha Utrapress é uma empresa voltada ao seguimento de fotos, álbuns de formatura e outras editorações gráficas. Trabalha com imagens e arquivos de grande porte e já possui algumas filiais espalhadas pelo território nacional, onde pretende ampliar seus serviços.

Preocupada em garantir a disponibilidade dos serviços, a escalabilidade de suas possível novas filiais, tal empresa busca um modelo em nuvem a qual possa garantir a prestação de serviços à seus clientes mesmo em caso das possíveis falha e catástrofes.

Esse trabalho tem, portanto como justificativa um estudo o qual possa direcionar uma futura implantação para a construção de uma nuvem privada a qual possa atender a demanda crescente da referida empresa.

1.3 Objetivo

Apresentar uma proposta de um ambiente em rede a partir de um estudo de caso, a qual permita a integração em nuvem de todos os pontos de uma empresa do ramo de editoração gráfica espalhadas no território nacional.

1.3.1 Objetivos Específicos

- Analisar o parque tecnológico atual da empresa Alpha Ultrapress;
- Apresentar uma proposta de computação em nuvem para futura implantação de um ambiente em nuvem;
- Testar possíveis falhas de topologia.

1.4 Organização do Trabalho

Esse trabalho é apresentado da seguinte forma: O Capítulo 2 e 3 são apresentados às fundamentações teóricas para embasamento e melhor compreensão do leitor. O capítulo seguinte, 4, é apresentada a metodologia utilizada nesse estudo. No capítulo 5, traz o desenvolvimento do trabalho, apresenta uma proposta para construção de uma nuvem privada elenca as principais ferramentas para a construção de uma nuvem privada. A análise de possíveis falhas as quais permitem a validação da proposta está no Capítulo 6. E por fim as conclusões do trabalho, vista no Capítulo 7.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados primeiramente são descritos as características de uma falha em sistemas, principais características de um cluster e de um sistema distribuído e suas falhas. Por fim, apresentando a descrição de uma rede privada e descrito tipos de nuvens e ferramentas para sua implantação.

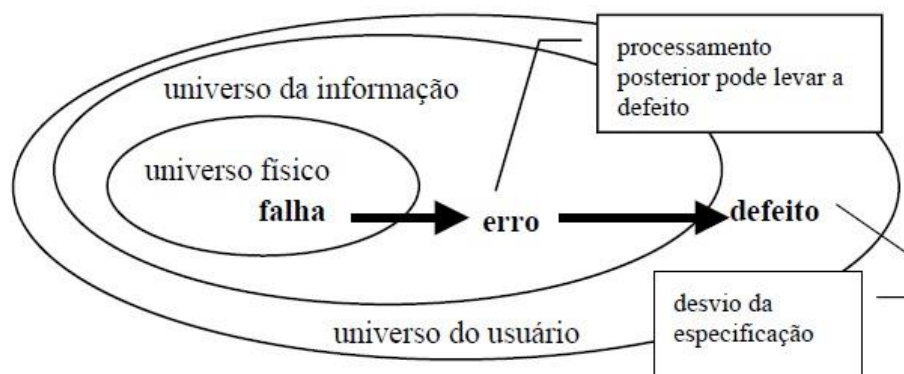
2.1 Conceitos Clássicos: Falha, Erro e Defeito

Importante estar atento a possíveis falhas, erros e defeitos que podem ocorrer em sistemas, onde um defeito (*failure*) é definido como um desvio da especificação. Defeitos não podem ser tolerados, mas devem ser evitados.

Define-se que um sistema está em estado errôneo, ou em erro, se o processamento posterior a partir desse estado pode levar a um defeito e finalmente define-se falha ou falta (*fault*) como a causa física ou algorítmica do erro. (JOHNSON, 1984)

Na Figura 2.1 é apresentada uma simplificação para os conceitos de falha, erro e defeito, onde falhas são associadas ao universo físico, erro ao universo da informação e defeito ao universo do usuário.

Figura 2.1: Modelo de 3 universos: falha, erro e defeito



Fonte: Adaptado de Barry W. Johnson, (1984)

2.2 Tolerância a falhas

Segundo (PEREIRA, 2004), uma rede confiável e sempre disponível são cada vez mais importantes e necessárias em ambientes computacionais. Isso se deve à grande necessidade de recursos disponíveis a todo o momento. As falhas em sistemas computacionais são inevitáveis, porém, um dos principais objetivos na utilização dos *clusters* (agrupamento de computadores) é justamente proporcionar meios de tolerância a estas falhas.

Na indústria o termo nunca teve boa aceitação, sendo que desenvolvedores de sistemas de controle preferem usar o termo sistemas redundantes para seus equipamentos. Na comercialização de sistemas computacionais como *mainframes* e servidores de rede, o termo usual é alta disponibilidade, designando a principal qualidade desses sistemas.

Sistemas redundantes e sistemas de alta disponibilidade apresentam técnicas comuns, mas alcançam resultados diferentes, uns visam alta confiabilidade e outros continuidade de serviço. Para englobar essas qualidades embaixo de um único chapéu, frequentemente aparece o termo segurança de funcionamento. Com a popularidade do termo segurança computacional, relacionado aos aspectos de segurança contra intrusos e mal-intencionados e que engloba criptografia, autenticação e vários tipos de proteção de sistemas, o termo segurança de funcionamento relacionado a tolerância a falhas caiu em desuso.

O próprio termo tolerância a falhas como designação de área sofre várias críticas, não apenas no Brasil, mas também internacionalmente (WEBER, 2003). A maior crítica é a possibilidade de entender o termo como uma propriedade absoluta. Nessa visão distorcida, um sistema tolerante a falhas conseguiria prever toda e qualquer falha em qualquer situação, o que realmente é uma promessa irrealizável e pode conduzir a falsas expectativas entre usuários. Assim, aos poucos o termo dependabilidade vem substituindo tolerância a falhas no meio acadêmico.

2.3 Dependabilidade

O termo dependabilidade é uma tradução literal do termo inglês *dependability*, que indica a qualidade do serviço fornecido por um dado sistema e a confiança depositada no serviço fornecido. Segundo Avizienis (2004) a dependabilidade de um sistema de computadores deve ser entendida como a habilidade de se evitar falhas em serviços que são mais frequentes e mais severas do que seria tolerável.

Tolerância a falhas e dependabilidade não são propriedades de um sistema a que se possam atribuir diretamente valores numéricos. Mas todos os atributos da dependabilidade correspondem a medidas numéricas. Principais atributos são confiabilidade, disponibilidade, segurança de funcionamento (*safety*), segurança (*security*), manutenibilidade, testabilidade e comprometimento do desempenho (*performability*). Um resumo dos principais atributos é mostrado na Tabela 2.1.

Tabela 2.1: Resumo dos principais atributos de dependabilidade

Atributos	Significado
Dependabilidade (<i>dependability</i>)	Qualidade do serviço fornecido por um dado sistema.
Confiabilidade (<i>reliability</i>)	Capacidade de atender a especificação, dentro de condições definidas, durante certo período de funcionamento e condicionado a estar operacional no início do período.
Disponibilidade (<i>availability</i>)	Probabilidade de o sistema estar operacional num instante de tempo determinado; alternância de períodos de funcionamento e reparo.
Segurança (<i>safety</i>)	Probabilidade do sistema ou estar operacional e executar sua função corretamente ou descontinuar suas funções de forma a não provocar dano a outros sistema ou pessoas que dele dependam.
Segurança (<i>security</i>)	Proteção contra falhas maliciosas, visando privacidade, autenticidade, integridade e irrepudiabilidade dos dados.

Fonte: WEBER (2003)

2.4 Cluster

Cluster (ou *clustering*) é o nome dado a um sistema que relaciona dois ou mais computadores para que estes trabalhem de maneira conjunta no intuito de processar uma tarefa. Estas máquinas dividem entre si as atividades de processamento e executam este trabalho de maneira simultânea. (BUYYYA, 1999).

Cada computador que faz parte do cluster recebe o nome de nó (ou nodo). Teoricamente, não há limite máximo de nós, mas independentemente da quantidade de máquinas que o compõe, o cluster deve ser "transparente", ou seja, ser visto pelo usuário ou por outro sistema que necessita deste processamento como um único computador.

Cada nó de um cluster pode ser um único computador ou um computador com múltiplos processadores simétricos (memória compartilhada). Cada nó tem a sua própria memória, dispositivos de entrada/saída e sistema operacional.

O seu surgimento remonta ao início da década de 90, motivado pelos baixos custos dos seus componentes - processador, redes de interconexão e memória, assim como do surgimento de ferramentas de alto desempenho para computação distribuída e do aumento da necessidade de poder de computação das aplicações científicas e comerciais.

Há vários tipos de *cluster*, mas os principais são: de alto desempenho, de alta disponibilidade e de balanceamento de carga.

Os *clusters* de alto desempenho (*High Performance Computing Cluster*) são direcionados a aplicações bastante exigentes no que diz respeito ao processamento. Sistemas utilizados em pesquisas científicas, por exemplo, podem se beneficiar deste tipo de *cluster* por necessitarem analisar uma grande variedade de dados rapidamente e realizar cálculos bastante complexos.

Os clusters de alta disponibilidade (*High Availability Computing Cluster*) têm como finalidade manter um sistema ativo durante um longo período, evitando quedas em serviços que deixariam sistemas inativos e causando prejuízos incalculáveis para as empresas. Nos *clusters* de alta disponibilidade o que garante um serviço

ininterrupto é o trabalho das máquinas configuradas para realizar em conjunto a replicação de aplicações e servidores. (PITANGA, 2008)

A alta disponibilidade é alcançada com utilização de *hardware* específico, como o é caso de sistemas os quais utilizam tanto a redundância da informação, quanto a conexão com a rede Internet. A Figura 2.2 ilustra um modelo de *cluster* de alta disponibilidade.

Figura 2.2: *Clusters* em Alta Disponibilidade



Fonte: Freekernel Security Systems

Está ligada diretamente a crescente dependência dos sistemas informatizados. Os equipamentos que hospedam esses sistemas possuem um papel crítico. O exemplo disso pode-se citar empresas as quais possuem como atividade a prestação de serviços relacionados ao comércio eletrônico (*e-business*), noticiários, banco de dados, dentre outros.

Clusters para Balanceamento de Carga (*Load Balancing*) as tarefas de processamento são distribuídas o mais uniformemente possível entre os nós. O foco aqui é fazer com que cada computador receba e atenda a uma requisição e não, necessariamente, que divida uma tarefa com outras máquinas.

A disponibilidade alcançada em *clusters* é bastante significativa. Por exemplo, 99% de disponibilidade significa que o *cluster* pode rodar 24 horas por dia, 7 dias por

semana, 52 semanas por ano com uma possível indisponibilidade de aproximadamente 5 dias, para manutenção planejada ou não.

Para muitas empresas, isso é satisfatório. Mas alguns sistemas de missão crítica (como transações bancárias, bolsas de valores, algumas aplicações médicas, guia e monitoramento militar ou controle de tráfego aéreo) exigem um grau mais elevado de disponibilidade. Nesses casos, 99,999% de disponibilidade significa uma indisponibilidade de cerca de 5 minutos por ano.

A dependabilidade é mais alta em um *cluster* do que a alcançada por meio de vários servidores independentes. Uma vez que o *cluster* é visto como uma única máquina, existe apenas um ambiente para monitorar e gerenciar. Os vários usuários e sistemas de armazenamento estão localizados no mesmo ambiente, tornando mais fácil a localização de problemas e mais rápido o reparo.

2.5 Sistemas Distribuídos

Um sistema distribuído pode ser definido como aquele "formado por componentes de *hardware* e *software*, situados em redes de computadores, e que se comunicam e coordenam suas ações apenas através de trocas de mensagens" (Coulouris, 2001).

A separação espacial dos computadores que formam esses sistemas distribuídos pode variar desde poucos metros, em uma rede local, até quilômetros de distância, em uma rede largamente dispersa. Da mesma maneira, podem variar bastante os desafios enfrentados pelos projetistas desses sistemas - frente aos problemas de heterogeneidade, segurança e confiabilidade (tolerância à faltas).

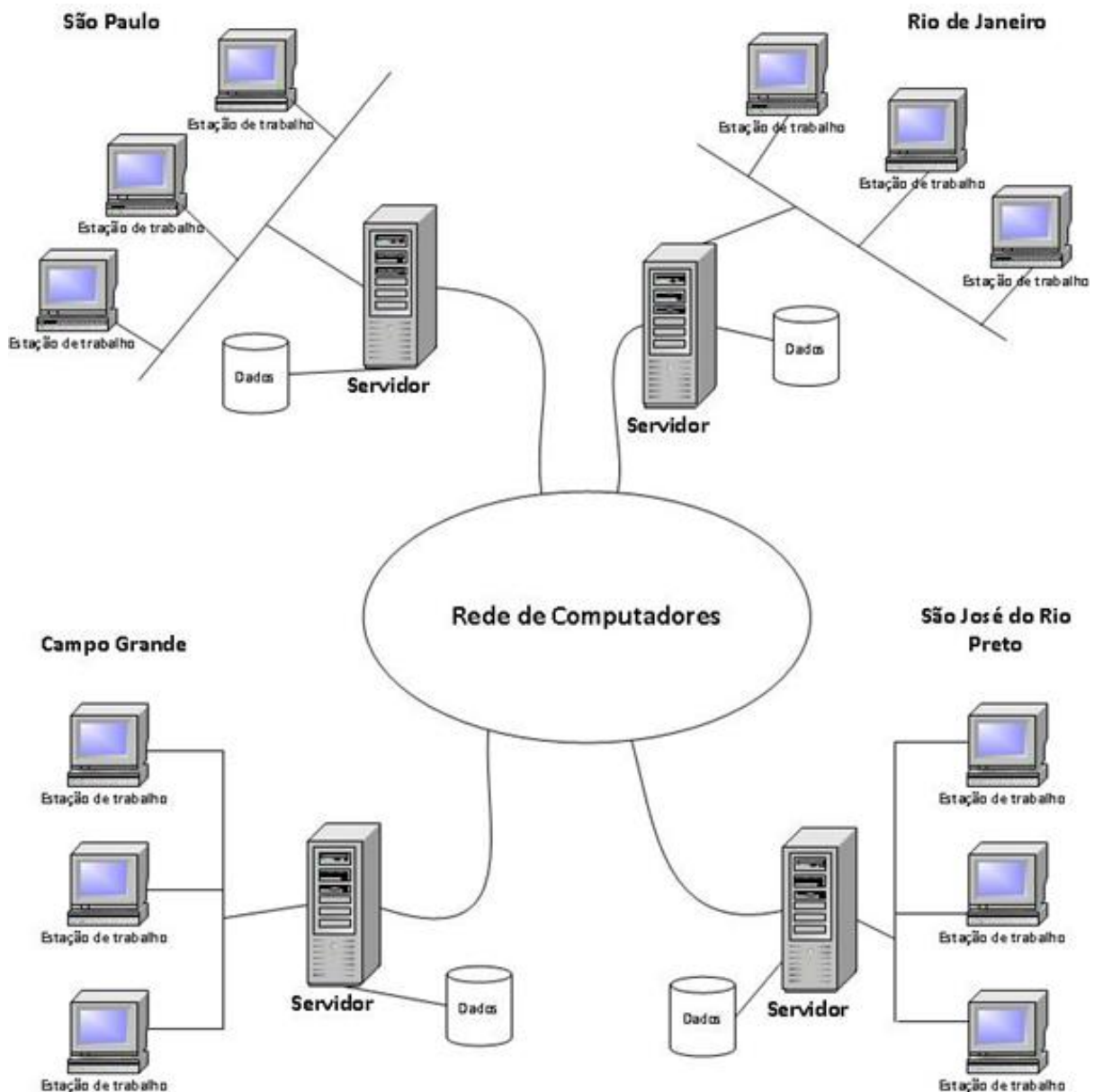
Características do sistema distribuído:

- **Compartilhamento:** uso de recursos de *hardware* e *software* compartilhados;
- **Interoperabilidade:** equipamentos e *software* de fabricantes diferentes;
- **Concorrência:** processamento concorrente para aumentar o desempenho;

- **Escalabilidade:** Capacidade ampliada pela adição de novos recursos;
- **Tolerância a falhas:** capacidade de continuar em operação após a ocorrência de uma falha;
- **Complexidade:** Tipicamente, sistemas distribuídos são mais complexos que sistemas centralizados. São mais difíceis de compreender e testar;
- **Segurança:** Mais suscetível a ataques externos;
- **Gerenciamento:** Os computadores podem ser de tipos diferentes e podem operar em versões diferentes de sistemas operacionais. Defeitos em uma máquina podem se propagar a outras máquinas;
- **Imprevisibilidade:** São imprevisíveis em suas respostas. A resposta depende da carga total do sistema, sua organização e a carga de rede, e isso pode mudar de uma hora para outra.

A Figura 2.3 ilustra o exemplo de um sistema distribuído em banco de dados que está totalmente transparente para os usuários que utilizam aplicações nessa arquitetura distribuída, isto é, para o usuário, os dados estão em um único servidor, mas na verdade eles podem estar distribuídos em vários locais fisicamente separados.

Figura 2.3: Modelo de Banco de Dados Distribuídos



Fonte: (Rosário, 2003)

2.6 Aplicações de Sistemas Tolerantes a Falhas

A redundância inerente à tolerância a falhas aumenta consideravelmente o custo de um sistema de computação. Só esse acréscimo de custo já implica no estabelecimento de fronteiras claras ao emprego indiscriminado de técnicas de tolerância a falhas. (Weber, 2003)

Para a escolha adequada de um sistema de computação tolerante a falhas, as características especiais da aplicação e as suas exigências quanto a confiabilidade e disponibilidade devem ser conhecidas em detalhe. Só obedecendo

critérios às exigências essenciais de cada aplicação torna-se possível contornar o custo associado ao emprego de técnicas de tolerância a falhas.

As áreas tradicionais onde são empregados sistemas tolerantes a falhas são:

- Aplicações críticas de sistemas de tempo real como medicina, controle de processos e transportes aéreos;
- Aplicações seguras de tempo real como transportes urbanos; a
- Aplicações em sistemas de tempo real de longo período de duração sem manutenção, como em viagens espaciais, satélites e sondas;
- Aplicações técnicas como telefonia e telecomunicações;
- Aplicações comerciais de alta disponibilidade como sistemas de transação e servidores de redes.

Essas áreas não abrangem todo o universo de aplicações onde tolerância a falhas pode ser empregada com vantagens para o usuário de sistemas de computação. Exigências quanto à disponibilidade e confiabilidade são encontradas em qualquer área. Usuários, que inicialmente se mostram satisfeitos em contar apenas com a simples automação de serviços, logo passam a desejar que esses serviços sejam prestados corretamente e sem interrupções.

Sistemas tolerantes a falhas são caros e, portanto empregados apenas naquelas situações em que a sua não utilização acarretaria prejuízos irrecuperáveis.

2.7 Tolerância a Falhas em Sistemas Distribuídos

Sistemas distribuídos são construídos por vários nodos processadores independentes. Esses sistemas se diferenciam de computadores paralelos pelo acoplamento fraco entre os nodos, ou seja, os elementos de um sistema distribuído não tem acesso a uma memória comum. Toda a interação deve ser feita por troca de mensagens através de canais de comunicação. Nodos de um sistema distribuído também não tem acesso a um relógio global, portanto não possuem uma base de tempo comum para ordenação de eventos. Além disso, sistemas distribuídos são

geralmente construídos com elementos não homogêneos e assíncronos.

Sistemas distribuídos não são sinônimos de redes de computadores. Uma rede de computadores pode fornecer a infraestrutura computacional para um sistema distribuído, mas nem toda aplicação de rede é necessariamente distribuída. Por exemplo, uma rede onde cada servidor roda uma aplicação e os demais nodos são clientes dessa aplicação não apresenta os problemas de consistência de dados e sincronização típicos de problemas distribuídos.

Por outro lado, um *cluster* de alto desempenho com centenas e até milhares de nodos pode ser considerado um sistema distribuído, sem necessariamente ser suportado por uma rede.

Sistemas distribuídos apresentam uma redundância natural, extremamente proveitosa para o emprego de técnicas de tolerância a falhas. Defeito em um nodo processador ou na rede de comunicação não precisa provocar necessariamente a queda de todo o sistema, e o sistema pode ser reconfigurado usando apenas os nodos disponíveis.

Garantir dependabilidade envolve solucionar problemas de consenso, ordenação e atomicidade na troca de mensagens entre grupos de processos, sincronizar relógios quando necessário, implementar réplicas consistentes de objetos, garantir resiliência de dados e processos num ambiente sujeito a quedas de estações tanto clientes como servidoras, particionamento de redes, perda e atrasos de mensagens e, eventualmente, comportamento arbitrário dos componentes do sistema.

Algumas características básicas devem ser consideradas para alcançar dependabilidade em sistemas distribuídos. Dependabilidade pode ser necessária, em maior ou menor grau, dependendo da aplicação. Mas todos os sistemas distribuídos devem suprir um mínimo de confiabilidade para permitir operação continuada do sistema mesmo com queda de um ou mais de seus nodos e canais.

2.8 Recuperação em sistemas distribuídos

Para que o sistema não sofra pane devido a falhas em seus componentes, erros devem ser detectados o mais rapidamente possível, o nodo faltoso deve ser

identificado através de diagnóstico apropriado e finalmente isolado através de reconfiguração do sistema. Essa reconfiguração se faz realocando processos e escolhendo caminhos alternativos de comunicação entre os processos. Assim, para aumentar a dependabilidade de sistemas distribuídos, detecção de erros, diagnóstico e reconfiguração são técnicas essenciais, que devem ser incorporadas de forma transparente à aplicação e ao seu ambiente de suporte.

Dependendo do tipo de sistema, da especificação de sua resposta no tempo e dos custos de implementação, falhas podem ser mascaradas ou erros podem ser recuperados.

Mascaramento é mais efetivo em sistemas de tempo real, podendo também ser usado em sistemas convencionais, mas o custo em hardware é maior. Nesse caso falhas não se propagam e o sistema continua seu processamento sem queda de desempenho.

As técnicas de recuperação por retorno não usam tanta redundância como as técnicas de mascaramento, mas pressupõe um estado global anterior seguro armazenado de alguma forma no sistema e a possibilidade de *rollback* para esse estado, o que inviabiliza a sua aplicação a uma vasta gama de sistemas de tempo real usados em controle de processos contínuos. As alternativas à recuperação por retorno são, em geral, de difícil projeto e implantação e extremamente vinculadas a casos bem modelados.

A recuperação por retorno, que é simples em um sistema com um único processo, pode se tornar bastante complexa em um sistema distribuído. A recuperação nesses sistemas envolve o retorno não apenas dos processos no nó falho, mas de todos os processos envolvidos direta ou indiretamente na comunicação com os primeiros, e pode provocar efeito dominó. Efeito dominó significa o retorno sucessivo, e em cascata, de todos os processos do sistema ao início da computação, ou próximo ao início, desfazendo grande quantidade de processamento.

Após a localização da falha por procedimentos de diagnóstico, o sistema distribuído pode ser reconfigurado. A reconfiguração envolve a determinação de uma nova configuração para a rede, o isolamento dos nós faltosos, redistribuição dos recursos restantes para as aplicações, realocação dos processos aos nós e reinicialização ou recuperação do sistema. Reconfiguração também é necessária

quando um novo nodo é integrado à rede.

Para a determinação de uma nova configuração para a rede é necessário consenso, todos os novos nodos devem reconhecer a mesma configuração (ou seja, quais são os nodos perfeitos que restaram e qual a topologia de sua interconexão). Após a migração dos processos e sua recuperação, o processamento pode ser reiniciado. Todas essas operações devem ser realizadas no menor intervalo de tempo possível e considerando, que mesmo nesse pequeno intervalo, novos nodos podem falhar na rede.

Deve-se considerar também, no caso de isolamento de nodos, a queda de desempenho gradativa a qual o sistema está sujeito a cada nova reconfiguração. Essa queda de desempenho pode inviabilizar algumas aplicações. A reconfiguração está também relacionada ao gerenciamento e manipulação de grupos de processos no momento da falha.

2.9 Rede Privada Virtual

É uma rede de acesso restrito, onde algumas partes são conectadas utilizando a rede pública da Internet e assim substituindo a tecnologia de links dedicados ou rede de pacotes para a conexão de redes remotas. Tendo motivo o lado financeiro, onde os links dedicados são caros, e do outro lado está a Internet, por ser uma rede de alcance mundial tem pontos de presença espalhados por todo o mundo (Martins, 2000).

Conexões com a Internet podem ter um custo mais baixo que *links* dedicados, principalmente quando as distâncias são grandes, esse tem sido o motivo pelo qual as empresas cada vez mais utilizam a infra-estrutura da Internet para conectar a rede privada. Com toda essa infra-estrutura de conexão entre *hosts* da rede privada é uma ótima solução em termos de custos, mas, não em termos de privacidade, pois a Internet é uma rede pública, onde os dados em trânsito podem ser lidos por qualquer equipamento.

Um das formas de interligar redes em pontos geograficamente distantes deve-se é a implantação de rede VPN, essas redes são construídas sobre a infra-estrutura de uma rede pública, mas é possível implantar métodos que mantêm as informações seguras e integras.

Estes túneis virtuais habilitam o tráfego dos dados criptografados através da Internet e esses dispositivos, são capazes de entender os dados criptografados formando uma rede virtual segura sobre a rede Internet.

Uma VPN deve oferecer mecanismos que garantam a segurança da comunicação em um meio inseguro como a Internet. Os principais protocolos adotados e difundidos em VPNs para garantir a segurança nas comunicações através da Internet são o IPSEC (*Internet Protocol Security*) e o SSL (*Secure Socket Layer*)(Fontes, 2011).

Na Tabela 2.2, é realizado uma comparação entre dois dos principais protocolos de segurança para VPN.

Tabela 2.2: Comparativo entre protocolos de segurança.

	SSL	IPSec
Aplicação	Aplicações que suportam <i>web browser</i> , <i>e-mail</i> e compartilhamento de arquivo.	Serviços baseados em IP
Criptografia	Forte, porem variável – depende de como a <i>web browser</i> foi configurado.	Forte e consistente – Depende da aplicação
Autenticação	É variável – pode se usar uma ou duas formas de autenticação. <i>tokens</i> e certificação digital.	É forte – pode se usar duas formas de autenticação, utilizando <i>tokens</i> e certificação digital.
Segurança	Moderada – pois com qualquer computador é possível estabelecer o túnel VPN	Forte – devido ao fato da aplicação ter de estar vinculada a um computador/usuário que irá conectar a VPN.
Acessibilidade	Em qualquer lugar a qualquer hora acesso a base de usuários	Acesso limitado à base de usuários bem definidos e controlados

	amplamente distribuídos	
Complexidade	Modesta	Alta
Custo	Baixo – pois não requer software específico	Alto – Requer <i>software</i> de cliente gerenciado
Escalabilidade	Alta – Facilmente implantada e escalonável	Modesta – Escalonável no lado do servidor

Fonte: Fontes (2011).

O importante dessa ferramenta é que ela utiliza um padrão de criptografia mundial, criado pelo IETF (*Internet Engineering Task Force*) tornando o tráfego das informações pelo “túnel” (é a denominação do caminho lógico percorrido pelos pacotes encapsulados) mais seguro.

Na Figura 2.4 é representando um modelo de redes VPN.

Figura 2.4: Topologia de redes VPN



Fonte: CAMPINHOS, E; BARCELLOS, R. (2007).

2.10 Computação em Nuvem

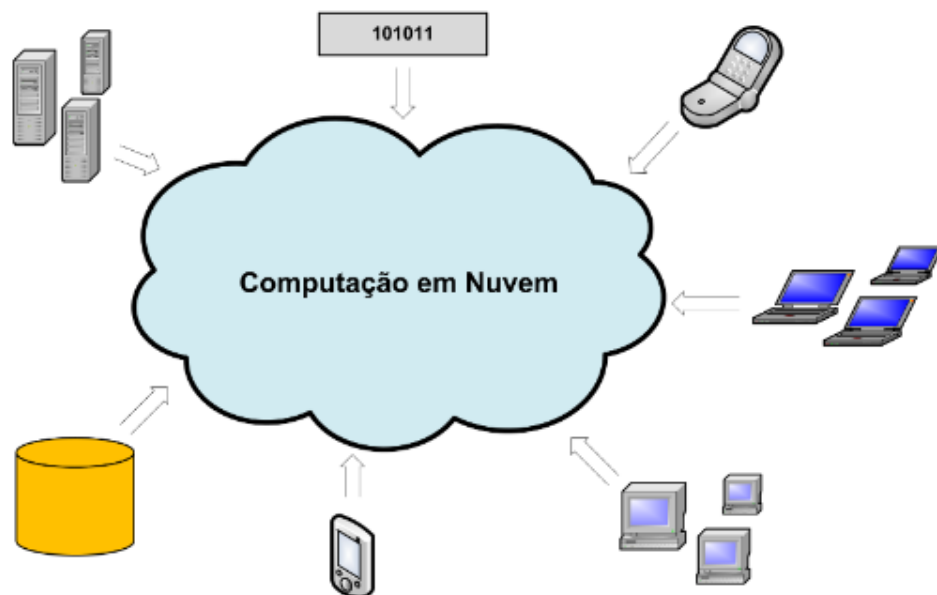
A expressão *cloud computing* começou a ganhar força em 2008, mas, conceitualmente, as ideias por trás da denominação existem há muito mais tempo. Também conhecida no Brasil como computação nas nuvens ou computação em nuvem, a *cloud computing* se refere, essencialmente, à noção de utilizarmos, em qualquer lugar e independente de plataforma, as mais variadas aplicações por meio da internet com a mesma facilidade de tê-las instaladas em computadores locais.

Segundo o Instituto Nacional de Padrões e Tecnologia do Departamento de Comércio norte-americano, 2011 (NIST) a “Computação em nuvem (*Cloud*

Computing) é um modelo desenvolvido para permitir acesso ubíquo, conveniente e sob demanda via rede a um agrupamento compartilhado e configurável de recursos computacionais (por exemplo, redes, servidores, equipamentos de armazenamento, aplicações e serviços), que pode ser rapidamente fornecido e liberado com esforços mínimos de gerenciamento ou interação com o provedor de serviços.”

Na Figura 2.5 é ilustrada uma visão geral de uma nuvem computacional onde usuários estão enviados seus dados e aplicações para a nuvem e assim podendo acessá-las de qualquer local do mundo de forma fácil e ágil.

Figura 2.5 - Visão Geral de uma Computação em Nuvem



Fonte: Souza et al. 2009

Segundo o NIST, são enumeradas cinco características essenciais do modelo de computação em nuvem:

- 1) **AUTOSERVIÇO SOB DEMANDA:** o consumidor deve ser capaz de alocar novos recursos automaticamente, sem interação humana com o provedor de serviços;
- 2) **ACESSO AMPLO VIA REDE:** os recursos devem estar disponíveis através

da rede e devem ser acessíveis por mecanismos padrão, permitindo seu uso por diferentes dispositivos, tais como computadores pessoais, *smartphones*, *tablets*, etc;

3) AGRUPAMENTO DE RECURSOS: os recursos computacionais do provedor de serviços devem ser agrupados para servir a múltiplos consumidores, com recursos físicos e virtuais sendo arranjados e rearranjados dinamicamente conforme a demanda desses consumidores. Deve haver um senso de independência de localização, no qual o consumidor não tem um controle exato de onde os recursos utilizados estão localizados, mas deve ser possível especificar esse local em alto nível de abstração (país, unidade federativa ou *datacenter*);

4) ELASTICIDADE RÁPIDA: os recursos devem ser alocados e liberados de forma elástica, e de forma automática em alguns casos, permitindo a rápida adaptação à demanda. Para o consumidor, os recursos disponíveis devem parecer ser ilimitados, sendo possível alocar a quantidade desejada desses recursos a qualquer momento;

5) SERVIÇOS MENSURADOS: serviços de computação em nuvem devem controlar e otimizar os recursos de maneira automática, disponibilizando mecanismos para medir esses recursos utilizando um sistema de medida apropriado para o tipo de recurso sendo utilizado (por exemplo, quantidade de espaço de armazenamento, velocidade de comunicação, capacidade de processamento, número de usuários ativos, etc.). Deve ser possível monitorar, controlar e consultar o uso dos recursos, provendo transparência para o consumidor e para o provedor dos serviços.

2.11 Serviços da Nuvem

Os serviços de computação em nuvem fornecem a Tecnologia da Informação (TI) como serviço pela Internet ou por uma rede dedicada, com entrega sob

demanda e pagamento com base no uso. Os serviços de computação em nuvem variam de aplicativos completos e plataformas de desenvolvimento até servidores, armazenamento e *desktops* virtuais.

2.12 Software as a Service (SaaS)

Intimamente ligado à computação em nuvem está o conceito de *Software as a Service* (SaaS) ou, em bom português, *Software como Serviço*. Em sua essência, trata-se de uma forma de trabalho em que o software é oferecido como serviço, assim, o usuário não precisa adquirir licenças de uso para instalação ou mesmo comprar computadores ou servidores para executá-lo. Nessa modalidade, no máximo, paga-se um valor periódico - como se fosse uma assinatura - somente pelos recursos utilizados e/ou pelo tempo de uso.

Para entender melhor os benefícios do SaaS, suponha que uma empresa que tem 20 funcionários necessita de um *software* para gerar folha de pagamento. Há várias soluções prontas para isso no mercado, no entanto, a empresa terá que comprar licenças de uso do *software* escolhido e, dependendo do caso, até mesmo *hardware* para executá-lo. Muitas vezes, o preço da licença ou mesmo dos equipamentos pode resultar em custo alto e não compatível com a condição de porte pequeno da empresa.

Se, por outro lado, a companhia encontrar um fornecedor de software para folha de pagamento que trabalha com o modelo SaaS, a situação pode ficar mais fácil: essa empresa poderá, por exemplo, oferecer esse serviço por meio de *cloud computing* e cobrar apenas pelo número de funcionários e/ou pelo tempo de uso. Com isso, a contratante paga um valor baixo pelo uso da aplicação. Além disso, *hardware*, instalação, atualização, manutenção, entre outros, são tarefas que ficam por conta do fornecedor.

Outro exemplo, é o *Skype* da *Microsoft*, que é um software de comunicação permite vários tipos de utilização, que pode ser a simples troca de mensagens até uma videoconferência em grupo. Nesse, é possível contratar apenas este recurso, ou combinar vários recursos oferecidos e adaptáveis a sua necessidade. É um modelo flexível, que lhe permite controlar o que se necessita, pagando apenas pelo que utiliza em determinado período ou situação pontual, isto é, o usuário paga o

serviço e não o produto (Alecrim 2011).

Também é importante levar em conta que o intervalo entre a contratação do serviço e o início de sua utilização é extremamente baixo, o que não aconteceria se o software tivesse que ser instalado nos computadores do cliente - este só precisa se preocupar com o acesso ao serviço (no caso, uma conexão à Internet) ou, se necessário, com a simples instalação de algum recurso mínimo, como um *plugin* no navegador de Internet de suas máquinas.

Há também conceitos derivados do SaaS que são utilizados por algumas companhias para diferenciar os seus serviços. São eles:

- **Platform as a Service (PaaS):** Plataforma como Serviço. Trata-se de um tipo de solução mais amplo para determinadas aplicações, incluindo todos (ou quase todos) os recursos necessários à operação, como armazenamento, banco de dados, escalabilidade (aumento automático da capacidade de armazenamento ou processamento), suporte a linguagens de programação, segurança e assim por diante;
- **Database as a Service (DaaS):** Banco de Dados como Serviço. O nome já deixa claro que essa modalidade é direcionada ao fornecimento de serviços para armazenamento e acesso de volumes de dados. A vantagem aqui é que o detentor da aplicação conta com maior flexibilidade para expandir o banco de dados, compartilhar as informações com outros sistemas, facilitar o acesso remoto por usuários autorizados, entre outros;
- **Infrastructure as a Service (IaaS):** Infraestrutura como Serviço. Parecido com o conceito de PaaS, mas aqui o foco é a estrutura de hardware ou de máquinas virtuais, com o usuário tendo inclusive acesso a recursos do sistema operacional;
- **Testing as a Service (TaaS):** Ensaio como Serviço. Oferece um ambiente apropriado para que o usuário possa testar aplicações e sistemas de maneira remota, simulando o comportamento destes em nível de execução.

2.13 Tipos de Nuvens

O desenvolvimento de um ambiente na nuvem irá depender da necessidade da aplicação a ser oferecida e do tipo de contrato de prestação de serviço. Apesar da aparência dos serviços serem disponibilidades de forma pública, onde qualquer usuário tem acesso a todo o conteúdo da nuvem, os modelos de negócios tem promovido o desenvolvimento de modelos de implantação que garantam um adequado nível de controle da informação a ser disponibilizada (tipo e conteúdo) e visibilidade da nuvem (NIST, 2011).

Atualmente os tipos de modelo de desenvolvimento são Público, Privado, Comunidade e Híbrido.

2.14 Nuvem privada

Até agora, foi tratado à computação nas nuvens como um sistema composto de duas partes: o provedor da solução e o utilizador, que pode ser uma pessoa, uma empresa ou qualquer outra organização. Entender-se que esse contexto como um esquema de nuvem pública. No entanto, especialmente no que diz respeito ao segmento corporativo, é possível também o uso do que se conhece como nuvem privada.

Do ponto de vista do usuário, a nuvem privada (*private cloud*) oferece praticamente os mesmos benefícios da nuvem pública. A diferença está, essencialmente, nos "bastidores" os equipamentos e sistemas utilizados para constituir a nuvem ficam dentro da infraestrutura da própria corporação(CHIRIGATI, 2009).

Em outras palavras, a empresa faz uso de uma nuvem particular, construída e mantida dentro de seus domínios. Mas o conceito vai mais além: a nuvem privada também considera a cultura corporativa, de forma que políticas, objetivos e outros aspectos inerentes às atividades da companhia sejam respeitados.

A necessidade de segurança e privacidade é um dos motivos que levam uma organização a adotar uma nuvem privada. Em serviços de terceiros, cláusulas contratuais e sistemas de proteção são os recursos oferecidos para evitar acesso não autorizado ou compartilhamento indevido de dados. Mesmo assim, uma

empresa pode ter dados críticos por demais para permitir que outra companhia responda pela proteção e disponibilização de suas informações. Ou, então, a proteção oferecida pode simplesmente não ser suficiente. Em situações como essas é que o uso de uma nuvem privada se mostra adequado.

Uma nuvem privada também pode oferecer a vantagem de ser "moldada" com precisão às necessidades da companhia, especialmente em relação a empresas de grande porte. Isso porque o acesso à nuvem pode ser melhor controlado, assim como a disponibilização de recursos pode ser direcionada de maneira mais eficiente, aspecto capaz de impactar positivamente a rotina corporativa.

Empresas como *Microsoft*, *IBM* e *HP* oferecem soluções para nuvens privadas. As organizações interessadas devem, todavia, contar com profissionais ou mesmo consultoria especializada na criação e manutenção da nuvem, afinal, uma implantação mal executada pode interferir negativamente no negócio.

Os custos de equipamentos, sistemas e profissionais da nuvem privada poderão ser elevados no início. Por outro lado, os benefícios obtidos a médio e longo prazo, como ampla disponibilidade, agilidade de processos e os já mencionados aspectos de segurança compensarão os gastos, especialmente se a implantação for otimizada com virtualização, padronização de serviços e afins.

2.15 Nuvem Híbrida

Para a flexibilização de operações e até mesmo para maior controle sobre os custos, as organizações podem optar também pela adoção de nuvens híbridas. Nelas, determinadas aplicações são direcionadas às nuvens públicas, enquanto outras, normalmente mais críticas, permanecem sob a responsabilidade de sua nuvem privada. Pode haver também recursos que funcionam em sistemas locais (*on premise*), complementando o que está nas nuvens.

Perceba que nuvens públicas e privadas não são modelos incompatíveis entre si. Não é preciso abrir mão de um tipo para usufruir o outro. Pode-se aproveitar o "melhor dos dois mundos", razão pela qual as nuvens híbridas são uma tendência muito forte nas corporações (CHIRIGATI, 2009).

A implantação de uma nuvem híbrida pode ser feita tanto para atender a uma demanda contínua quanto para dar conta de uma necessidade temporária. Por exemplo, uma instituição financeira pode integrar à sua nuvem privada um serviço público capaz de atender a uma nova exigência tributária. Ou então, uma rede de lojas pode adotar uma solução híbrida por um curto período para atender ao aumento das vendas em uma época festiva.

É claro que a eficácia de uma nuvem híbrida depende da qualidade da sua implantação. É necessário considerar aspectos de segurança, monitoramento, comunicação, treinamento, entre outros.

Esse planejamento é importante para avaliar inclusive se a solução híbrida vale a pena. Quando o tempo necessário para a implantação é muito grande ou quando há grandes volumes de dados a serem transferidos para os recursos públicos, por exemplo, seu uso pode não ser viável.

2.16 Nuvem Pública

No modelo de implantação público, a infra-estrutura de nuvens é disponibilizada para o público em geral, sendo acessado por qualquer usuário que conheça a localização do serviço. Neste modelo de implantação não podem ser aplicadas restrições de acesso quanto ao gerenciamento de redes, e menos ainda, aplicar técnicas de autenticação e autorização (CHIRIGATI, 2009).

2.17 Nuvem Comunidade

O modelo comunidade é caracterizado pelo fato da infra-estrutura de nuvem ser compartilhada por várias organizações e suporta uma comunidade específica que partilha as mesmas preocupações como missão, requisitos de segurança, política e considerações de conformidade. Pode ser gerenciado pelas organizações ou por terceiros e podem existir localmente ou remotamente (Souza et al. 2009)

2.17 Considerações Finais

Nesse capítulo foi apresentada uma revisão de literatura, em alguns desses trabalhos é apresentado os tipos de sistemas distribuídos, suas características e possíveis falhas. Apresentaram-se também alguns tipos de nuvens, seus atributos e uma descrição de redes privadas virtuais e um comparativo de qual melhor protocolo ser utilizado nessas redes.

3 Construção da Nuvem

A construção da nuvem é feita através de ferramentas próprias, ferramentas essas que auxiliam em toda a configuração para que o ambiente *cloud* funcione de forma correta e segura.

3.1 Sistemas Construtoras de Nuvens

Os Sistemas Construtoras de Nuvens constituem um novo paradigma e um conjunto de novas metodologias e processos de gestão de infraestruturas informáticas.

Enquanto num sistema tradicional os recursos são provisionados e configurados manualmente por pessoal qualificado, os sistemas de operação em nuvem estão disponibilizados em um ambiente *multi-tenancy* onde os recursos são provisionados automaticamente (de forma *self-service*) a partir de um *pool* de recursos de forma fácil e intuitiva (geralmente através de aplicações *web*) ficando imediatamente disponíveis para utilização.

Embora existam inúmeras ferramentas de computação em nuvem, optou-se por apresentar apenas as ferramentas código aberto para o modelo de infraestrutura como serviço.

3.1.1 *OpenStack*

É um software de código aberto, capaz de gerenciar os componentes de múltiplas infraestruturas virtualizadas, assim como o sistema operacional gerencia os componentes de nossos computadores, o *OpenStack* é chamado de Sistema Operacional da Nuvem, por cumprir o mesmo papel em maior escala.

É considerada uma plataforma de *software*, por fornecer APIs que em conjunto são capazes de controlar todos os recursos disponíveis na oferta dessa infraestrutura: máquinas virtuais, rede, armazenadores, balanceadores de carga, até mesmo um painel de controle *web* está presente entre os softwares do *OpenStack*, a maior parte escrita em *Python*.

O *OpenStack* é um conjunto de projetos de *software* de código aberto usados para configurar e operar infraestrutura de computação e armazenamento em nuvem.

A *Rackspace* (provedor de infraestrutura americano) e a NASA (agência espacial americana) foram os principais contribuidores iniciais para o projeto.

A *Rackspace* forneceu sua plataforma "*Cloud Files*" para implantar o aspecto de armazenamento (*Objeto Store*) do *OpenStack*, enquanto que a NASA entrou com o "*Nebula*" para implementar o lado computacional (*Nova Compute*).

O consórcio *OpenStack* desde então agregou mais de 100 membros em menos de um ano, incluindo a Canonical (responsável pelo *Ubuntu*), *Dell*, *Citrix*, *Red Hat*, *IBM*, *Cisco*, *Dell*, *Hewlett-Packard*, *SUSE*, *VMware*, *Yahoo!*, etc.

O *OpenStack* apresenta seus serviços através de APIs compatíveis com os serviços EC2 e S3 da *Amazon AWS*, e portanto aplicações escritas para estes serviços do AWS podem ser usados com *OpenStack* também.

Segundo (Lopes, 2012) os principais serviços no *OpenStack*, são:

- Computação (*Nova Compute*);
- Objeto Store (*Swift*);
- Imagem (*Glance*);
- Dashboard (*Horizon*);
- Identidade (*Keystone*).

Computação (*Nova Compute*): é um controlador de nuvem, usado para inicializar instâncias virtuais para o usuário ou um grupo. Pode-se ser utilizado também para configurar a rede para cada instância ou projeto que contenha várias instâncias de um projeto particular.

Na *Nova Compute*, existem componentes específicos para a realização de diversas tarefas. São eles:

- **API Server:** responsável por traduzir chamadas do mundo externo para ações dentro da nuvem por meio de chamadas *Web Services* (é uma solução utilizada para a integração de sistemas onde *software* ou *hardware* podem enviar ou receber mensagens);

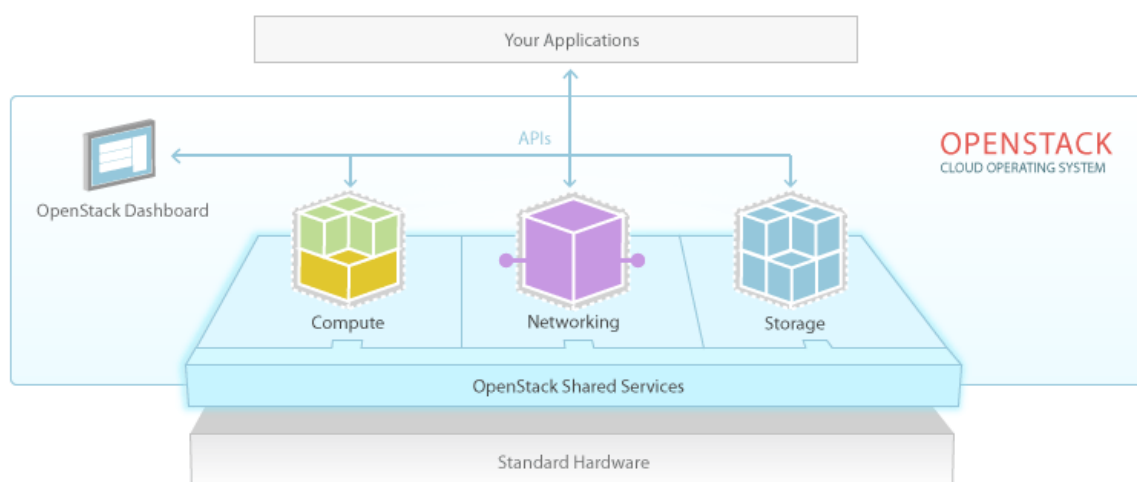
- **Message queue:** sistema que implementa o protocolo AMQP (Protocolo Avançado de Filas de Mensagens) e controla a troca de mensagens entre os componentes Nova;
- **Compute worker:** é o componente que dá origem a um nó de computação, sendo gerenciador do ciclo de vida das instâncias virtuais;
- **Network controller:** componente responsável por implementar os controles de rede dentro da infraestrutura de nuvem, gerenciando tanto operações nos nós da nuvem quanto nas máquinas virtuais;
- **Volume worker:** gerencia os discos virtuais para armazenamento permanente. No *OpenStack*, uma instância possui armazenamento volátil. Isto significa que, quando ela for deslocada da nuvem, os dados de seu sistema de arquivos serão perdidos. Para prover armazenamento persistente são utilizados volumes lógicos que são montados nas instâncias virtuais. Esse componente é quem gerencia estes volumes, permitindo que instâncias armazenem dados em um sistema de arquivos permanente;
- **Scheduler:** esse componente escalona ações dentro da nuvem. Uma ação importante é a definição de qual nó físico da nuvem executará uma instância virtual. Ele possui algoritmos que permitem regular como a escolha ocorrerá, considerando fatores como carga de processamento, memória disponível e arquitetura da CPU;
- **Objeto Store (Swift):** é um sistema para armazenar objetos em um sistema de grande capacidade, altamente escalável e redundante. O *Object Storage* tem uma variedade de aplicações, tais como *backup* ou arquivamento de dados, servidor de gráficos ou vídeos (*streaming* de dados para o navegador do usuário), armazenamento secundário ou terciário de dados estáticos, desenvolvendo novas aplicações com integração de dados de armazenamento. O serviço *Object Storage* usa uma arquitetura distribuída, sem um ponto central de controle, para prover maior escalabilidade e redundância. Objetos são replicados em múltiplos nós, sendo o serviço

responsável por essa replicação e por garantir a integridade dos dados através do cluster. No caso de falha de um nó, o serviço refaz seu conteúdo em outro nó ativo;

- **Imagem (*Glance*):** é um sistema de pesquisa e recuperação de imagens de máquinas virtuais. Ele pode ser configurado de três maneiras:
 - Usando *OpenStackObjectStore* para armazenar imagens.
 - Utilizando solução da *Amazon Simple Storage (S3)* de armazenamento diretamente.
 - Utilizando a solução *Simple Storage Service* da *Amazon (S3)* para através armazenamento S3, com armazenamento de objetos como o intermediário para o acesso S3.
- **Dashboard (*Horizon*):** provê para o usuário uma *interface web* para todos os serviços *OpenStack*. Com esta interface web, o usuário pode realizar a maioria das operações em sua nuvem, como o lançamento de uma instância, a atribuição de endereços IP e definir controles de acesso, ou seja, é um front-end web para os outros serviços *OpenStack*;
- **Identidade (*Keystone*):** fornece uma política de acesso, autenticação e autorização, para todos os serviços *OpenStack*. Ele também oferece um catálogo de serviços dentro de uma nuvem *OpenStack*.

Imagens públicas ou privadas (acessíveis por um grupo específico de usuários), de diferentes formatos, podem ser catalogadas nesse serviço. Atualmente é oferecido suporte aos seguintes formatos de imagens: RAW (formatos de arquivos de imagens digitais), VHD (Disco Rígido Virtual), VDI (Imagem de Disco Virtual), VMDK (Disco Virtual *Machine*) e OVF (Formato de Virtualização Aberto).

Outra característica interessante do *OpenStack* é a possibilidade de realizar *scale-up* (ampliar) e *scale-down on demand* (diminuir gradualmente a demanda), possibilitando expandir e recolher recursos de uma instância de forma transparente.

Figura 3.1: Arquitetura do *OpenStack*

Fonte: *OpenStack* (2013)

Para melhor entendimento da arquitetura do *OpenStack*, na Figura 3.1 está representada os principais blocos de construção de seus componentes. Camadas relacionadas a aplicações e administração de acesso estão no topo da pilha, com os elementos padrões de recursos para computação, redes e armazenamento vindo imediatamente abaixo deles. Tudo repousa sobre os serviços compartilhados e, no fim, em plataformas padrão específicos de hardware de código livre.

3.1.2 *CloudStack*

É uma ferramenta *Open-Source* que gerencia todos os componentes de múltiplas infraestruturas virtualizadas. Permite a criação de nuvens privadas, híbridas e públicas que podem fornecer infraestrutura como um serviço para os usuários.

- **Suporte a múltiplos hipervisores**

O *CloudStack* trabalha com uma variedade de *hipervisores*. Uma única implantação de nuvem pode conter múltiplas implementações de hipervisores. O release atual do *CloudStack* suporta soluções empresariais *pre-packaged* como o *Citrix XenServer* e o *VMware vSphere*, assim como *KVM* ou *Xen* executando no *Ubuntu* ou *CentOS*.

- **Gestão de infraestrutura altamente escalável**

O *CloudStack* pode gerenciar dezenas de milhares de servidores instalados em múltiplos centros de computação geograficamente distribuídos. O servidor de gerenciamento centralizado é linearmente escalável, eliminando a necessidade de servidores de gerenciamento de cluster intermediários. Nenhuma falha de componente único pode causar uma interrupção geral na nuvem. A manutenção periódica do servidor de gerenciamento pode ser executada sem afetar o funcionamento de máquinas virtuais que são executadas na nuvem.

- **Gerenciamento automático de configuração**

O *CloudStack* automaticamente configura os parâmetros de rede e armazenamento de cada máquina virtual hóspede. O *CloudStack* gerencia internamente um pool de dispositivos virtuais ("virtual appliances") para suporte à nuvem. Estes dispositivos oferecem serviços como *firewall*, roteamento, *DHCP*, acesso *VPN*, *console proxy*, acesso a *storage* e replicação de *storage*. O uso extensivo de dispositivos virtuais simplifica a instalação, configuração e gerenciamento contínuo de uma implantação de nuvem.

- **Interface gráfica do usuário**

O *CloudStack* oferece uma *interface web* para o administrador, usado para provisionamento e gestão da nuvem, assim como uma *interface web* do usuário final, usada para executar máquinas virtuais e gerenciar modelos (*templates*) de máquinas virtuais. A interface de usuário pode ser customizada para refletir os padrões de visuais de apresentação do provedor de serviços ou empresa.

- **Alta disponibilidade**

A plataforma do *CloudStack* tem um número de recursos para aumentar a disponibilidade do sistema. O próprio servidor de gerenciamento pode ser

implantado em um ambiente com múltiplos nós onde é feito balanceamento de carga entre os servidores. *MySQL* pode ser configurado para usar replicação, provendo uma método manual de recuperação em caso de perda do *database*. Para os hosts, a plataforma *CloudStack* suporta *NIC bonding* e o uso de redes isoladas de *storage*, assim como *iSCSI Multipath*.

3.1.3 *OpenNebula*

É um projeto *Open-Source* que compreende um conjunto de ferramentas que viabilizam a gerência de *datacenters* e a criação de nuvens privadas ou híbridas virtualizadas segundo o modelo de infraestrutura como serviço.

O *OpenNebula* é composto por cinco componentes principais:

- Interfaces e APIs, que permitem a interação de um usuário com uma infraestrutura estabelecida com o *OpenNebula*;
- Grupos e usuários, responsáveis pela criação de contas e grupo e pelos mecanismos de autenticação e autorização que o sistema fornece;
- Controle de rede, subsistemas responsáveis gerenciar a integração com outras redes ou datacenter;
- Host e virtualizações, que suporta gerenciadores de virtualização para gerência do ciclo de vida; e
- Monitoramento das máquinas virtuais e o componente para armazenamento que pode ser configurado para suportar um sistema de arquivos compartilhado ou não. (Laszewski, et al 2012)

Consegue lidar com picos de utilização através do *Cloud Bursting*, possui também uma arquitetura flexível que pode acomodar hardwares múltiplos e combinações diferentes de *softwares*.

A arquitetura interna do *OpenNebula* engloba vários componentes especializados em diferentes aspectos para o gerenciamento da infraestrutura virtual e pode ser dividida em três camadas. *Core* (núcleo) - Responsável por controlar o ciclo de vida das máquinas virtuais, o núcleo do *OpenNebula* possui três diferentes áreas de gerenciamento, são elas:

- Tecnologias de imagens e armazenamento, para a preparação de

imagens de disco para as máquinas virtuais;

- Tecnologias de rede, para o provimento de ambientes de rede para as máquinas virtuais;
- *Hipervisores*, para a criação e gerenciamento das máquinas virtuais.

3.1.4 *Eucalyptus*

É uma infraestrutura de *software* de código aberto que utiliza diversos recursos computacionais disponível aos pesquisadores, como *clusters* e *workstation farms*. A fim de promover a exploração da comunidade científica de sistemas de computação em nuvem, o *Eucalyptus* foi desenvolvido totalmente modularizado, permitindo os pesquisadores realizarem experimentos de segurança, escalabilidade, confiabilidade e implementação de interfaces. (Nurmi et al. 1998-2000)

É compatível com varias ferramentas de monitoramento como por ex: *Nagios* e o *Ganglia*. Nenhuma ferramenta para o controle e monitoramento de seus componentes e das máquinas virtuais é disponibilizada. O *Eucalyptus* possui duas versões:

- *Eucalyptus Enterprise Edition*
- *Eucalyptus Open-Source*

Na versão Enterprise as organizações tem a possibilidade de implementar nuvens privadas com suas próprias políticas, de forma bastante intuitiva utilizando todo o aparato desenvolvido pela plataforma *Eucalyptus*.

Já na versão *Open-Source*, que também conta com todo aparato da plataforma, qualquer configuração pode ser visualizada por toda comunidade, além das seguintes restrições que são impostas, visando, de acordo com a própria Equipe *Eucalyptus*: “maximizar o número de membros da comunidade que se beneficiam do sistema e para minimizar os efeitos negativos dos experimentos que deram errados”. Os usuários do *Eucalyptus* tem acesso a diversos recursos, como pequenos clusters, conjunto de *workstations* e várias máquinas servidoras ou *desktop*.

Como os endereços de IP são escassos e o acesso público a toda ramificação dos hosts a nuvem interna pode ser perigosa, geralmente os administradores de rede elegem um nó (conhecido como nó *front-end*) responsável

por fazer um roteamento das informações públicas para os *hosts* internos.

Estes *hosts* internos somente podem se comunicar com seu respectivo nó *front-end* ou com os demais *hosts* internos através de uma rede privada.

O *cluster controller* é o componente responsável por realizar o roteamento de pacotes entre as redes virtualizadas externa (pública) com a interna (privada). Desta forma, todas as manipulações realizadas nos recursos físicos alocados nos *Node Controllers* devem ser necessariamente, gerenciadas pelo *Cluster Controller*.

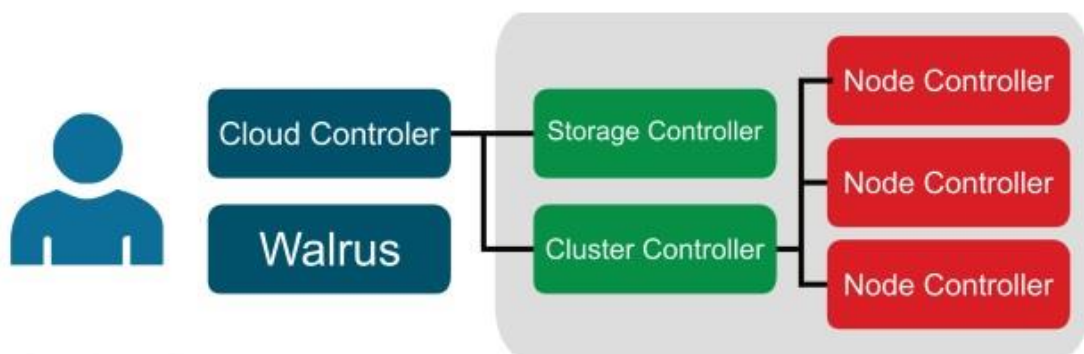
Para tal, o *Cluster Controller* armazena um arquivo *WSDL* com todas as definições das interfaces para realizar as devidas operações. Este arquivo *WSDL* é bastante similar ao do *Node Controller*, porém este possibilita realizar as operações tanto para um único *Node Controller* quanto para um conjunto de *Nodes Controllers*.

Quando um *Cluster Controller* recebe uma requisição de *runInstance*, ele avalia qual o melhor *Node Controller* para executar esta nova instância a partir de critérios como a quantidade de recursos disponíveis e o consumo de energia da máquina física associada ao *Node Controller*.

O *Walrus* é um serviço de armazenamento em *Eucalyptus* responsável pelo armazenamento de imagens de máquinas, armazenamento de *snapshots* (formato de ficheiros) e servir arquivos usando S3 API. Deve ser considerado como um sistema de armazenamento de ficheiros simples.

Na Figura abaixo é apresentada a arquitetura do *Eucalyptus*.

Figura 3.2: Arquitetura do *Eucalyptus*.



Fonte: *Eucalyptus Documentation* (Julho, 2015).

3.2 Diferenças entre as ferramentas

Na Tabela abaixo apresenta cada particularidade das ferramentas e suas principais diferenças.

Tabela 3.1: Tabela comparativa das ferramentas *clouds*.

	OpenStack	CloudStack	OpenNebula	Eucalyptus
Ambiente de desenvolvimento	Código Aberto	Código Aberto	Código Aberto	Código Aberto
Ambiente ideal	Foi desenvolvido para ambiente que utilizam o disco maciçamente	Foi desenvolvido para implantar e gerenciar grandes redes de máquinas virtuais	Melhor em ambientes que usam rede, processador e memória	Foi desenvolvido para ambiente que utilizam sistemas mais antigos
Vantagens	Utiliza principais <i>Hypervisores</i> do mercado	Excelente interface possui assistente gráfico	Voltadas para atender aos requisitos de empresas que utilizavam a ferramenta em versões anteriores	Bem visto entre os especialistas no ramo
Desvantagens	Quando a queda de um dos nodos é necessária à reconstrução inteira da estrutura	Baixa quantidade de colaboradores	Poucos <i>Hypervisores</i> suportados	Poucos <i>Hypervisores</i> suportados

Fonte: o autor.

3.3 Considerações Finais

Nesse capítulo foi apresentado 4 tipos de ferramentas para construção de nuvens privadas, suas principais características e suas arquiteturas. Foi criada uma tabela apresentando diferenças entre as ferramentas.

4 METODOLOGIA

A natureza da pesquisa é exploratória a qual foi utilizada para determinar o uso de ferramentas para a construção de nuvem. As técnicas utilizadas para o desenvolvimento do trabalho são pesquisas bibliográficas e apresentação de cenários para entendimento melhor do ambiente.

Com relação à abordagem do problema este trabalho é desenvolvido seguindo uma abordagem quantitativa, uma vez que o foco está em determinar o melhor conjunto de software para se trabalhar em determinado cenário.

Para alcançar os objetivos propostos nesta pesquisa foi necessário os seguintes passos metodológicos:

- a) Fundamentação Teórica: Abordagem de alguns temas como: funcionamento de ambiente virtualizados, contextualização de ambientes distribuídos.
- b) Estudo dos trabalhos relacionados para determinar qual ferramenta a ser utilizada no ambiente da empresa analisada.
- c) Realização de avaliações nos cenários a partir de uso de linguagens formais, utilizando a teoria de conjuntos.

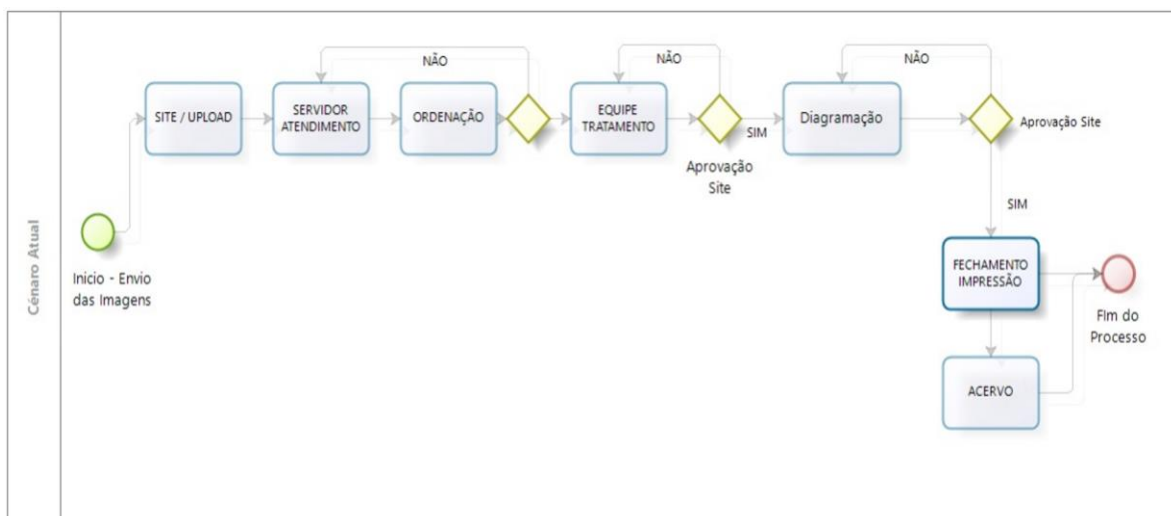
5 DESENVOLVIMENTO

A empresa em análise, Alpha Ultrapress localizada em Bandeirantes – PR, trabalha com arquivos gráficos, dos *softwares* da *Adobe®* e *CorelDraw®*, o tamanho desses arquivos são em média de 5 a 7 *Gbytes* cada, criando um ambiente de dados entorno de 30 *TBytes* entre fotos e arquivos.

Um dos cenários da empresa que geram esse grande volume de dados é o recebimento diário de arquivos de seus clientes (formandos), eles enviam a empresa um número elevado de fotos, tais imagens são inseridas nos projetos dos convites, o tamanho desses arquivos variam muito, podendo chegar até 5 *GBytes* cada pasta de foto.

Como ilustrado na Figura 5.1, os arquivos são enviados pelos clientes através de uma ferramenta disponível no site da empresa.

Figura 5.1: Cenário atual da empresa.



Fonte: o autor

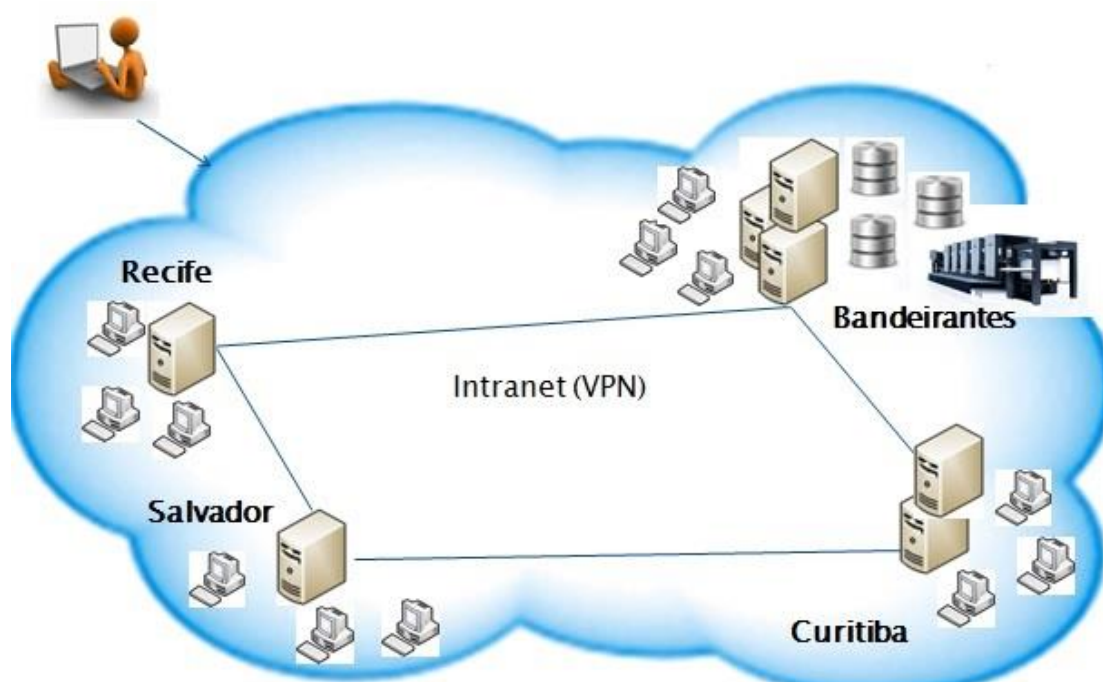
Uma vez que as imagens são recebidas na nuvem da empresa, o setor de atendimento confere e direciona os arquivos ao setor de ordenação. As imagens são então verificadas segundo os critérios de conformidade pré-estabelecidas. Em caso negativo, o setor de atendimento retorna ao cliente solicitando a correção e o reenvio. Em caso positivo, a organização das imagens é feita e separada pelo número de contrato. Para cada pasta de contrato há a separação por nome de curso, nome de turma e nome do formando.

Após essa separação os arquivos são disponibilizados no servidor para a equipe de tratamento, cada foto é verificada para corrigir imperfeições caso haja necessidade. Após esse procedimento as fotos são publicadas no site para aprovação dos formandos, caso não estejam em conforme, os arquivos são novamente retornados ao setor para correção das imperfeições. Se após a publicação estiverem em conforme as imagens são disponibilizadas para a equipe de diagramação.

No setor de diagramação as fotos são utilizadas para a criação de cada projeto de convite, nesse processo são gerados os arquivos com maior tamanho, após a criação dos projetos os modelos são publicados no site para aprovação dos clientes, se os modelos estiverem corretos e com os itens que o cliente solicitou eles são fechados gerando arquivos em *pdf* (formato de documento portátil) no tamanho e dimensões corretas para então ser enviado para impressão.

Como é apresentado na Figura 5.2, para que a comunicação entre as filiais aconteça é necessário à configuração em cada localidade de um servidor *gateway*, esse equipamento será responsável em realizar o tunelamento entre as filiais, de forma ágil e segura.

Figura 5.2: Tunelamento entre filiais e matriz



Fonte: o autor.

Neste cenário, os usuários das filiais terão acessos ao servidor da Matriz Bandeirantes podendo copiar, editar e excluir arquivos, facilitando o acesso às

informações necessárias.

Para que essa comunicação seja segura e confiável deve-se levar em consideração inúmeros aspectos, já descritos na Tabela 2.2, existem vários protocolos de segurança, os principais são o IPSec e o SSL.

Como apresentado na Tabela 2.2, os protocolos de segurança apresentam suas vantagens e desvantagens, observa-se que o protocolo IPSec leva vantagem em alguns aspectos como na autenticação, o IPSec é associado ao *software* de VPN instalado no computador para conectar-se no túnel, com isso restringindo o acesso a equipamentos não autorizados, ou seja, os que não possuem o *software* IPSec VPN instalado e configurado.

Em relação à criptografia o IPSec, durante o estabelecimento do túnel VPN, o usuário deve concordar com as políticas de segurança impostas pela corporação, com isso assegurando o administrador de rede que o usuário aderiu a política de segurança.

Já em relação utilização do usuário o SSL leva vantagem em relação ao IPSec sendo mais prático a implantação e também em relação ao custo da implantação do SSL é mais baixo que do IPSec.

Segundo Fontes (2011), o protocolo IPSec possui características que o torna viável em aplicações site-to-site, conectando através de um dispositivo, geralmente um roteador, ou *firewall*, estabelecendo uma conexão confiável entre dois pontos através de dispositivos e softwares que adotam o IPSec.

No entanto, o IPSec, não demonstra pontos fortes em ambientes em que se exige um alto grau de mobilidade. Para os casos onde se exige um alto fator de mobilidade e escalabilidade, o SSL apresenta benefícios que o torna mais adequado a este tipo de ambiente.

Depois de realizadas análises nos dois protocolos de segurança é aconselhável a utilização nesse ambiente o protocolo SSL por apresentar características mais indicadas a realidade da empresa gráfica.

5.1 Cópia das Máquinas Virtuais e Arquivos

O funcionamento da cópia dos sistemas e servidores são diretamente ligados a configurações da ferramenta escolhida para a implantação, como é apresentado no capítulo anterior existem inúmeras ferramentas que dispõem de

configurações e APIs que facilitam a construção do ambiente, tais como: *OpenStack*, *OpenNebula*, *Eucalyptus* e *CloudStack*,

O *OpenStack* como já apresentado no capítulo anterior a ferramenta apresenta inúmeros serviços, é possível gerenciar toda a estrutura da nuvem, como por exemplo, provê uma *interface web* para outros serviços, armazena e recupera discos virtuais, gerencia as redes virtuais para nuvem e pode armazenar os arquivos reais de discos virtuais.

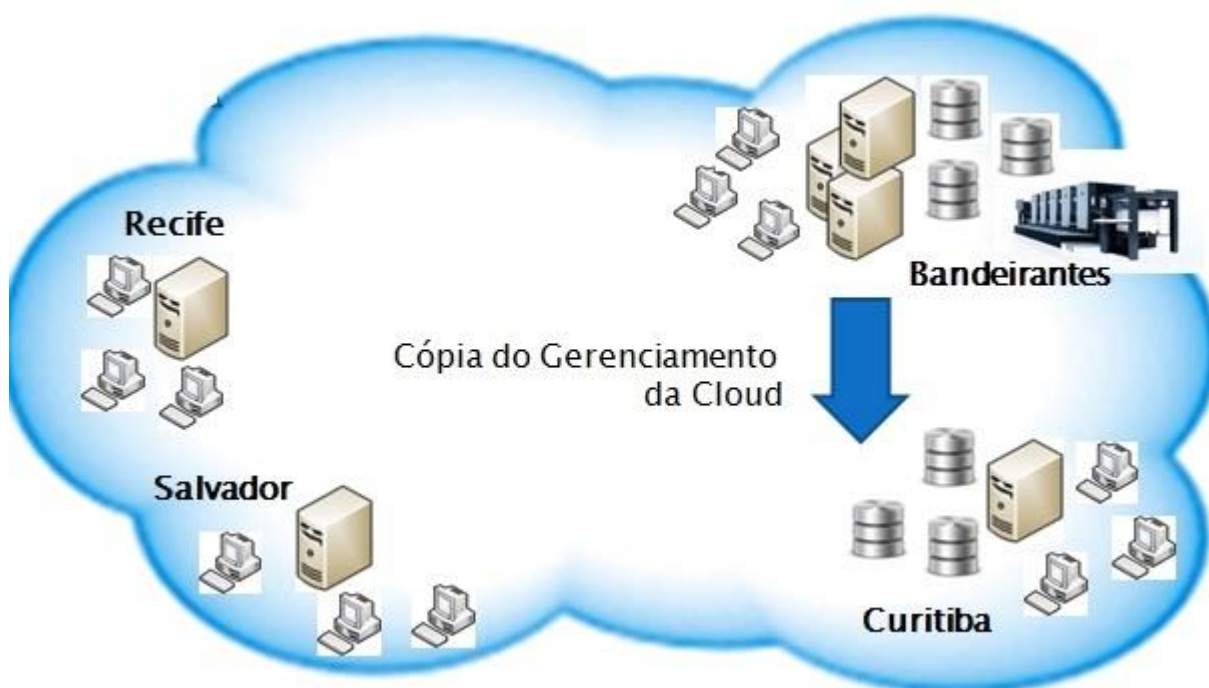
Em relação ao *backup* dos sistemas, o *OpenStack* tem um item muito importante que é a *OpenStack Swift* é o mais adequado para fazer *backup* e arquivar dados não estruturados, como documentos, imagens, arquivos de áudio e vídeo, e-mail e imagens de máquinas virtuais. Os objetos são gravados em varias unidades, o software Swift garante que os dados são replicados em um *cluster* servidor.

Por padrão o Swift coloca três cópias de cada objeto em locais exclusivos, levando em consideração a ordem, primeiro por região, seguido por zona, servidor e unidade. Se uma unidade de disco falhar ou servidor, o *OpenStack Swift* replica seu conteúdo a partir de nós ativos para novos locais da *cloud*.

Como a lógica de replicação é toda feita através de *software*, não necessitando de técnicas por *hardware* como o *RAID*, servidores e discos convencionais e baratos podem ser utilizados.

Como se pode ver na Figura 5.3 a cópia dos sistemas e arquivos é realizada através desse ambiente de tunelamento, as cópias são sincronizadas automaticamente. Toda alteração realizado nos servidores na *cloud* na copia é alterada também por ser um sistema sincronizado.

Figura 5.3: Cenário de cópia dos sistemas e arquivos.



Fonte: o autor.

5.2 Localidade das Cópias

Importante salientar o que a escolha dessa localidade para manter a cópia dos arquivos e sistemas em sincronismo foi em relação à qualidade de serviço de Internet, rápida resposta na necessidade de suporte técnico e mão de obra local.

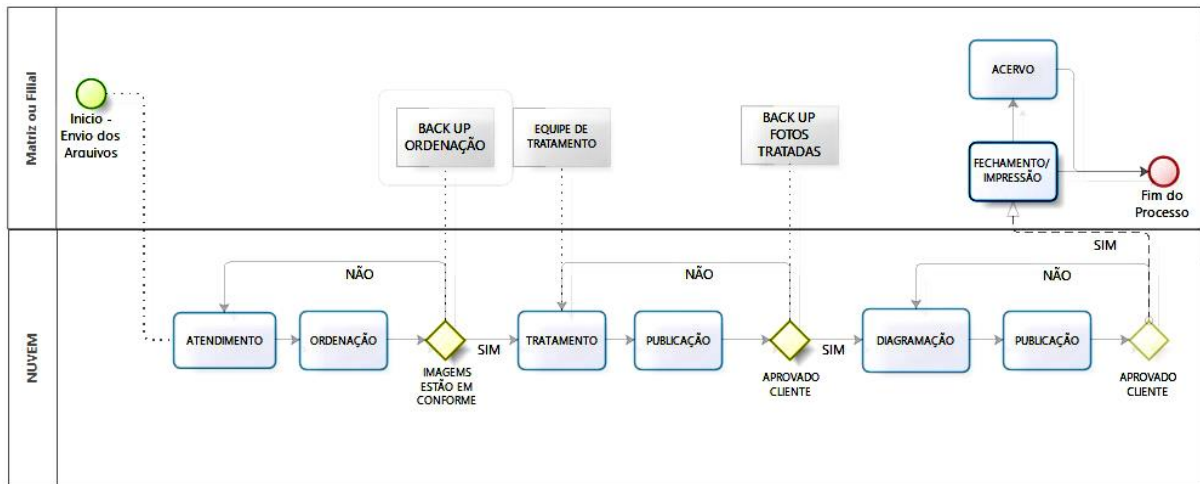
E principalmente por que a localiza esta em dos pontos mapeados pela RNP (Rede Nacional de Ensino e Pesquisa), pois foi construído para garantir não só a largura de banda necessária ao tráfego Internet usual (navegação *web*, correio eletrônico, transferência de arquivos), mas também o uso de serviços e aplicações avançadas e a experimentação.

5.3 Backup Fora do Ambiente da Nuvem

Na Figura 5.4 é ilustrado o *backup* das imagens sendo realizada fora do ambiente da nuvem, isso se faz necessário quando ocorrer falhas humanas e/ou técnicas no ambiente. Esse *backup* seria automático e em pontos importante do processo.

Se a falha a acontecer, o funcionário da matriz tem acesso ao *backup* para agilizar a recuperação dos arquivos e dar continuidade no processo.

Figura 5.4: Cenário Proposto.



Fonte: o autor.

5.4 Diagrama Proposto

Em um cenário proposto é importante apresentar todas as características, suas falhas e também suas soluções. Nessa sessão apresenta-se o modelo que é proposto para empresa em análise.

Na Figura 5.4, é ilustrado o cenário proposto, nela a imagem é enviado pelos clientes através de uma ferramenta disponível no site da empresa. Uma vez que as imagens são recebidas na nuvem da empresa, o setor de atendimento confere e direciona os arquivos ao setor de ordenação. As imagens são então verificadas segundo os critérios de conformidade pré-estabelecidas.

Caso haja a negativa dessas conformidades, o setor de atendimento retorna ao cliente solicitando a correção e o reenvio. Em caso positivo, a organização das imagens é feita e separada pelo número de contrato.

Para cada pasta de contrato, há a separação por nome de curso, nome de turma e nome do formando. Após esse procedimento o *backup* é realizado localmente das imagens ordenadas.

Ao mesmo tempo em que se realiza o *backup* das imagens ordenadas, elas ficam disponíveis para a equipe de tratamento realizar as devidas às correções.

Após as correções, as imagens são publicadas no site aguardando a

aprovação do cliente. Em caso negativo as imagens são novamente tratadas retornam ao setor de tratamento. Em caso positivo, as imagens são disponibilizadas ao setor de diagramação.

Nesse setor as imagens são inseridas junto dos projetos dos convites, esses arquivos também são publicados para aprovação ou não dos clientes. Em caso positivo, as imagens são disponibilizadas ao setor de fechamento. Caso contrário, os projetos são refeitos.

No setor de fechamento os arquivos já aprovados são configurados nas dimensões das impressoras. E por fim os arquivos fechados são copiados no servidor de acervo, encerrado o processo de confecção dos produtos.

Para que o ambiente descrito anteriormente funcione de forma correta, é necessário a utilização de ferramentas de construção de nuvem, como o *OpenStack*. Essa ferramenta fará todo o gerenciamento do cenário apresentado, podendo criar e configurar todos os sistemas que ficaram disponíveis e compartilhados na nuvem para todas as filiais. Para gerenciar as máquinas virtuais utiliza-se o *OpenStack Swift* como já descrito anteriormente, ele é responsável pela criação novas máquinas virtuais e também realização do *backup* das imagens.

6 Análise do diagrama proposto

Apresentam-se abaixo algumas premissas para que sejam analisadas levando em consideração o cenário proposto.

- Cloud.Alpha é o nome atribuído a nuvem privada da editora.
- B, C são empresas filiais da editora as quais possuem compõem a “Cloud.Alpha” hardware e software capazes de armazenamento de dados.
- R e S são atualmente empresas filiais de menor porte (escritórios) as quais ainda não possuem servidores de armazenamento, no entanto compartilham sistemas e dados de controle dos contratos.
- Cloud.Alpha contém { B, C, R e S }

- B, C, R e S são conexas entre si através de um link de rede internet. Os dados trafegados entre elas ocorre através de um tunelamento VPN, utilizando o protocolo SSL.
- Os *links* de redundância estão presentes apenas nos setores onde há o maior armazenamento de dados. Isto, devido a dependabilidade do acesso à linha de produção. Nesse caso B e C.
- Os arquivos ou pastas de contrato (a), são aqui tratados como uma única variável, podendo eles estarem completos, incompletos ou com erros de armazenamento (corrompidos).
- A variação Δt para sincronismos de (a) entre B e C depende principalmente do meio, isto é, da rede internet contratada, porém este procedimento é transparente para o setor de produção, devido a falta de referencial.

6.1 Análise de possíveis falhas

Caso 1. Detectado a ausência de arquivos/contratos (a) em B, (a não pertence B).

a) Se a não pertence B, mas a pertence a C então existe uma falha de *hardware* ou *software* da *Cloud*.Alpha a qual não permite que haja sincronismo dos dados. Como procedimento, deve-se solicitar o reparo e efetuar transferência direta e imediata para o setor de produção, caso haja urgência de contrato.

b) Se a= não existe, então

- não houve abertura de contratos;
- a foi sumariamente excluído;

Se a = não existe, então uma verificação no sistema de gerenciamento de contratos deve ser feita. Caso necessário e ainda mais profundo, efetuar

análise.

Se a foi sumariamente excluído. Então se deve efetuar auditoria para analisar o erro ocorrido. Somente o responsável pelo arquivamento dos contratos tem a prerrogativa e permissão para exclusão dos contratos e arquivos.

c) Caso o intervalo entre a inserção dos dados e a sua exclusão sejam maior que 1 dia, então uma cópia de segurança (backup fora da nuvem) deverá estar disponível para restauração.

Caso 2. Imagem (a) erroneamente alterada.

Uma imagem foi erroneamente alterada, isto é, não foi devidamente tratada conforme os procedimentos pré-especificados pela empresa. Além disso, e como houve o salvamento desse na Cloud.Alpha, B e C já possuem o arquivo/contrato (a) atualizados de forma errada. Durante o processo de aprovação foi possível perceber a falha. Nesse caso, somente através da restauração do backup é possível recuperar uma versão anterior do arquivo/contrato para novo tratamento de imagem.

Caso 3. As unidades deixam de ser temporariamente conexas entre si.

Por serem *datacenters* B e C utilizam link de redundância para disponibilidade dos sistemas e arquivos entre si e entre as demais. Essas falhas podem estar associadas a:

- Falta de fornecimento de energia elétrica;
- Falta no fornecimento da rede internet;
- Falha nos equipamentos de a fronteira (roteadores).

Assim como em uma nuvem privada ou pública contratada, a falta de qualquer um dos itens acima mencionados irá acarretar na desconexão temporária.

Como procedimento, deve-se efetuar a abertura de uma OS (ordem de serviço) junto as empresas contratantes, e aguarda os reparos.

Parcialmente, os setores de produção (B e C) poderão ainda continuar

operando em função de manterem seus arquivos/contratos em seus servidores locais.

Caso 4. Falha ou manutenção nos servidores de produção locais

Caso haja falha de hardware e/ou armazenamento dos dados locais, uma outra máquina (servidora) deverá ser temporariamente alocada para otimização do processo de reparação da falha ou manutenção programada. Nesse caso, utiliza-se o mesmo procedimento descrito no Caso 1 (c).

Caso 5. Falha no gerenciador da Cloud.Alpha (*OpenStack*)

Durante a construção do ambiente principal Cloud.Alpha é importante que seja preparado paralelamente um ambiente pré-configurado igual ao ambiente principal, pois caso haja necessidade o ambiente é ativo para o pleno funcionamento do ambiente.

Caso 6. Escalabilidade e flexibilidade do setor de produção.

Caso a empresa entenda que os custos de produção podem ser menores em outro local ou um novo produto/serviço seja agregado em uma outra empresa filial, então não haverá grande dificuldade em criar um novo parque de impressão. Isso devido a possibilidade dos arquivos estarem disponibilizados em uma *Cloud*. Nesse caso torna-se necessário um máquina local para que haja sincronismo de somente alguns ou todos os contratos destinados a produção. Essa máquina será portanto a mesa de trabalho local.

7 CONCLUSÃO

Após análises realizadas e que foram apresentadas no decorrer desse trabalho optou-se por uma nuvem privada por atender uma necessidade ou demanda específica de negócio. Os estudos indicaram que a ferramenta de código livre, *OpenStack* foi a que se mostrou mais condizente com as atividades da empresa voltada ao ramo de editoração de imagens. Essa por permitir um gerenciamento de disco e arquivos de grande porte de forma mais eficiente. Além disso, a ferramenta permite que se possa ter maior flexibilidade e escalabilidade, uma vez que essa é também a preocupação da empresa.

É possível concluir que há a possibilidade de efetuar outras mudanças no cenário atual da empresa, isto é, possibilita a realização de mudanças de todo setor de diagramação e tratamento para outra e qualquer filial.

Durante a análise do parque de TI da empresa observou-se que há também uma necessidade em se modificar os processos de negócios atualmente apresentados afim de serem adequados ao cenário de uma nuvem privada.

Outra conclusão em que se pode chegar e referente ao tempo de reparação quando há a ocorrência de falhas. Em uma nuvem pública, o tempo de reparação da falha é superior a de uma nuvem privada.

7.1 Trabalhos Futuros

Como trabalho futuro pretende-se:

- Realizar todas as simulações sobre a ferramenta de construção de nuvem;
- Analisar os custos de implantação da Cloud.Alpha.

REFERÊNCIAS

Alecrim, E. (2011). **O que é cloud computing (computação em nuvem)?** Disponível em <http://www.infowester.com/cloudcomputing.php/>. Acessado em: 08/08/2015

Openstack - Arquitetura Openstack. (2003) http://docs.openstack.org/admin-guide-cloud/content/ch_getting-started-with-openstack.html. Acessado em 15/08/15.

AVIZIENIS, A.; LAPRIE, J. C.; RANDELL, B.; LANDWEHR C.. **Basic concepts and taxonomy of dependable and secure computing.** In: IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. Anais. p.11-33, 2004.

BACELLAR H V. Cluster: **Computação de Alto Desempenho. Instituto de Computação.** Disponível em: <http://www.ic.unicamp.br/~ducatte/mo401/1s2010/T2/107077-t2.pdf> - Brasil

BUY YA, R., “**Parallel Programming Models and Paradigms**” In: “**High Performance Cluster Computing**”, v.1, pp 4-26, Prentice Hall, 1999.

BUY YA, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. **Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility.** Future Gener. Comput. Syst., 25(6):599–616, 2009b.

CACHO, Nélio. Lopes, Frederico. Batista, Thais. **Avaliando o Aprisionamento entre Várias Plataformas de Computação em Nuvem.** 31º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos– Brasil, 2013.

CARDOSO, Rogério N. Pereira – VPN – **Programa Cisco 1 a 1** – Disponível em <http://www.mier.com/reports/cisco/MPLS-VPN.pdf>, 2003.

CAMPINHOS, E.; BARCELOS, R. **TOPOLOGIA DE VPN: otimizando eficiência e segurança.** Vitoria, 2007.

CONTI, F. de. **Grandes computacionais para processamento de alto desempenho**. Universidade Federal de Santa Maria, 2009.

CHIRIGATI, Fernando Seabra. **Computação em Nuvem**. Rio de Janeiro, RJ. 2009.
Disponível em: <http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2009_2/seabra/>.
Acessado em 18/08/2015.

D. Nurmi, R. W. C. Grzegorzcyk, G. Obertelli, S. Soman, L. Youselff, and D. Zagorodnov, “**Eucalyptus: A technical report on an elastic utility computing architecture linking your programs to useful systems**,” UCSB Computer Science Technical, 2008-2010.

EUCALYTUS. Eucalyptus Documentation. Disponível em:
<https://www.eucalyptus.com/docs/> Acessado em 22/07/2015.

Exemplo objeto de armazenamento durante a Instalação:
<http://docs.openstack.org/grizzly/openstack-object-storage/admin/content/example-object-storage-installation-architecture.html>, 2013.

Estudo e caracterização de plataformas de Computação em Nuvem. Programa CTIC – Rede Nacional de Ensino e Pesquisa - AltoStratus: Soluções de Middleware para Composição, Execução e Gerenciamento de Serviços em Nuvens Híbridas e Heterogêneas, 2011.

LG. COULOURIS, J. Dollimore, T. Kindberg, **Distributed Systems: Concepts and Design**, Third Edition, Addison-Wesley, 2001.

JOHNSON, D. The Intel 432: **a VLSI architecture for fault-tolerant computer systems**. Computer, New York, 17(8):40-48, 1984.

LASZEWSKI, G.; DIAZ, J.; WANG, F.; FOX, G. **Comparison of Multiple Cloud Frameworks**. In 5th International Conference on Cloud Computing. IEEE, 2012.

LOPES F.; Almeida A.; Batista T.; Cavalcante E.; Gondim R.; Diniz T.; Cássio A.; Cesar Thiago. **Introdução a Plataformas de Computação em Nuvem – Uma Abordagem Prática.** <http://www.ppgsc.ufrn.br/~evertonrsc/publications/2012-EPOCA.pdf>, 2012.

MARTINS, D; **Redes Privadas Virtuais com IPSec.** Universidade de Brasília Ciência da Computação. <http://www.cic.unb.br/~rezende/trabs/vpn.pdf>, 2010.

MORAN, J. M. **O que aprendi sobre avaliação em cursos semipresenciais.** Especialista em projetos inovadores na educação presencial e a distância. Publicado em: SILVA, Marco & SANTOS, Edméa (Orgs). Avaliação da Aprendizagem em Educação. Disponível em: <http://www.eca.usp.br/prof/moran/aprendi.htm>, 2006.

MARON, C. A. F; Griebler, D; Vogel, A; Schepke, C; **Avaliação e Comparação do Desempenho das Ferramentas OpenStack e OpenNebula.** Universidade Católica do Rio Grande do Sul (PUCRS) – Porto Alegre – RS – Brasil. http://www.researchgate.net/profile/Dalvan_Griebler/publication/269872669_Avaliao_e_Comparao_do_Desempenho_das_Ferramentas_OpenStack_e_OpenNebula/links/54bcdd9f0cf24e50e940a3ef.pdf, 2014.

NIST (201) "**National Institute of Standards and Technology Draft Definition of Cloud Computing**". <http://csrc.nist.gov/groups/SNS/cloud-computing>. Acessado em: 27/08/2015

PRADHAN, D. K., Fault-Tolerant System Design. Prentice Hall, New Jersey, 1996.

PEREIRA, N. A. **Serviços de pertinência para clusters de alta disponibilidade.** Disponível em: <http://www.teses.usp.br/teses/disponiveis/45/45134/tde-04102004-104700/publico/dissertacao.pdf> , 2004.

RAPOPORT E.; VPN - **Virtual Private Network Rede Privada Virtual.** Departamento de Engenharia Eletrônica e de Computação (DEL) Escola

Politécnica/Universidade Federal do Rio de Janeiro. Disponível em <http://www.gta.ufrj.br/~rezende/cursos/eel879/trabalhos/vpn/>. Julho de 2003.

SOUSA, F. R. C., Moreira, L. O, Machado, J. C. (2009). **Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios** - Universidade Federal do Ceará, Capítulo 7, 2009.

TANENBAUM, Andrew S. Redes de Computadores. Tradução: Vandenberg D. de Souza. – 4ª Edição. Rio de Janeiro: Elsevier, ISBN: 85-352-1185-3, 2003.

VIEIRA, Rodrigo C.; Guardieiro, Paulo R.; **Proposta e Avaliação de uma Política de Preempção de LSPs Implementada com Lógica Fuzzy e Algoritmos Genéticos em um Ambiente de Rede DiffServ/MPLS**. Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, 2005.

LASZEWSKI V., J. Diaz, F. Wang e G. C. Fox. **Comparison of Multiple Cloud Frameworks**. IEEE Fifth International Conference on Cloud Computing. Washington. 2012. p.734-741, 2002.

WEBER, T. **Um roteiro para exploração de conceitos básicos de tolerância a falhas**. Instituto de Informática – UFRGS, 2003.

ZAGARI, E., T. Badan, R. Prado, E. Cardozo, M. Magalhães. **Uma Plataforma para Engenharia de Tráfego com Qualidade de Serviço em Redes MPLS**. In 20 Simpósio Brasileiro de Redes de Computadores - SBRC'2002, Búzios, R.J., Maio 2002.