



**Universidade Federal de Santa Catarina – UFSC**  
**Departamento de Informática e Estatística**  
**Curso de Sistemas de Informação**

# **Computação em Nuvem em Data Centers**

Autor

Frederico Luciano Galler de Magalhães Gomes

Orientador: Prof. Carlos Westphall  
Banca: Carla Westphall e Fernando Cruz

Florianópolis, 2010.2

## **Agradecimentos**

Agradeço primeiramente à minha família. À minha mãe que me ensinou a sempre valorizar o lado positivo das coisas, mesmo quando não vemos esperança. Ao meu pai que despertou em mim a curiosidade pela ciência e a honestidade entre os homens. À minha amada esposa que suportou (não com tanta paciência assim) a minha ausência durante os últimos meses em função deste trabalho, porém nunca me faltou com amor e carinho.

Agradeço também à raça: Donadel, Gabriel, Mohamed e Neto pelo companheirismo nestes quase cinco anos de facul, pelas parceiras na sinuca e no boteco, e pelos ensinamentos sobre truco e coisas afins.

Finalmente agradeço ao professor Westphall pela paciência de ter um orientando que apesar de não ser muito presente, procurou fazer um trabalho de qualidade e que venha a contribuir com os estudos do LRG e da UFSC.

**“O segredo é não correr atrás das borboletas...  
É cuidar do jardim para que elas venham até você.”**

*- Mário Quintana*

# Conteúdo

Resumo.....	5
1. Introdução.....	6
1.1 Justificativas / Motivação.....	7
1.2 Definição do Problema .....	7
1.3 Objetivos .....	7
1.4 Estrutura do trabalho.....	8
2. Fundamentação Teórica.....	9
2.1 Redes de Computadores .....	9
2.2 Sistemas Distribuídos .....	12
2.2.1 Virtualização.....	13
2.2.2 Grids .....	15
2.2.3 Nuvens .....	16
2.3 Arquiteturas de TI.....	23
2.4 Data Center .....	25
2.4.1 Topologia e Classificação (TIA-942) .....	27
2.4.2 Arquitetura .....	31
3. Desenvolvimento .....	35
3.1 Requisitos de infraestrutura para computação em nuvem .....	35
3.2 Data Center .....	38
3.2.1 Análise de tráfego de rede no data center .....	39
3.2.2 Limitações das arquiteturas tradicionais.....	42
3.2.3 Requisitos de redes para data centers em nuvem.....	44
3.3 Propostas de arquiteturas para data centers.....	46
3.3.1 DCell.....	46
3.3.2 VL2.....	50

3.3.3	PortLand.....	55
3.3.4	Comparativo .....	58
4.	Conclusão .....	62
4.1	Trabalhos Futuros .....	63
5.	Bibliografia .....	64

## **Resumo**

Arquiteturas tradicionais de data center apresentam limitações de escalabilidade, de capacidade de comunicação entre servidores e de agilidade de migração de máquinas virtuais. Estes fatores são essencialmente necessários para a implantação de uma nuvem computacional que atenda com plenitude o seu conceito baseado na dinamicidade de serviços e recursos. Neste trabalho será discutido computação em nuvem, data centers e as novas propostas de redes que visam a evolução das arquiteturas tradicionais de data centers.

## 1. Introdução

O principal insumo das organizações modernas é a informação. É dela que se extrai o substrato para as decisões tomadas sobre os processos organizacionais, desde os operacionais até os estratégicos. Estes mesmos processos geram um volume crescente e não esgotável de dados. Data centers são projetados para suportar o processamento e armazenamento de grandes quantidades de informação através de uma infraestrutura que é determinante para o desempenho e disponibilidade das aplicações que trabalham com estas informações. A tendência de consolidação aliada ao crescente uso e desenvolvimento do multiprocessamento e da virtualização de computadores, além do barateamento de equipamentos de computação tradicionais, estão transformando o data center no principal elemento computacional de uma organização.

O mais recente modelo de computação distribuída cunhado como computação em nuvem promete mudanças profundas na forma de entrega de serviços computacionais. Traz consigo um novo modelo de negócio para a computação. Mais simples para o usuário final, que agora se torna um consumidor e paga apenas pelo consumido. Do outro lado, melhor gerenciável para as empresas fornecedoras de serviço (hardware, software e comunicação). Seus maiores benefícios são a redução de custos de capital e operacional, a elasticidade e integração da infraestrutura.

O modelo tradicional de data center possui limitações, impostas principalmente pelo modelo de rede IP/Ethernet, que enrijece a estrutura do data center, limita sua escalabilidade e causa a fragmentação de recursos. Este trabalho discute os requisitos para estes data centers operarem neste novo modelo de computação em nuvem. Como objetivos, pretende-se estudar as características das arquiteturas de data centers tradicionais, identificar os requisitos de infraestrutura para computação em nuvem, e, apresentar três modelos inovadores de arquitetura de data centers (DCell, VL2 e PortLand) que buscam atender estes requisitos.

O escopo deste trabalho se limita a discussão dos requisitos baseado em bibliografia especializada, não havendo implementação e validação de modelo.

### **1.1 Justificativas / Motivação**

A adoção do modelo de computação em nuvem por data centers, além de oferecer elasticidade na disponibilização do serviço, reduz gastos na aquisição e operacionalização da infraestrutura e ainda diminui a quantidade de poluentes liberado para manter o sistema. Economia de gastos, disponibilidade e preservação ecológica são metas também presentes nos objetivos das organizações.

O grau de evolução do conhecimento humano observado nas últimas décadas nunca foi registrado antes. A computação, além de consequência, também dá suporte a esta evolução. Participar e contribuir para este processo através do desenvolvimento científico e tecnológico é a grande força motivacional deste trabalho.

### **1.2 Definição do Problema**

O problema a ser tratado neste trabalho é a identificação dos requisitos de computação em nuvem que devem ser tratados para uma arquitetura de data centers que dê suporte à computação em nuvem. As soluções tradicionais possuem limitações para a escalabilidade, agilidade e economia, porém já existem propostas atuais que atendem estes requisitos.

### **1.3 Objetivos**

O objetivo geral deste trabalho é identificar os requisitos que devem estar presentes em arquiteturas de data centers que garantam as características e necessidades de infraestrutura do modelo de computação em nuvem.

Como objetivos específicos podemos relacionar o estudo das arquiteturas de data centers e de computação em nuvem, a análise das

propostas de arquiteturas mais significativas disponíveis no mercado e no meio acadêmico que buscam atender o problema proposto neste trabalho.

#### **1.4 Estrutura do trabalho**

O trabalho é estruturado em quatro capítulos, sendo este a (1) Introdução, seguido do capítulo (2) de Fundamentação Teórica, onde se busca definir conceitos de redes, sistemas distribuídos e data centers, que irão dar embasamento para a análise das arquiteturas propostas. No capítulo de (3) Desenvolvimento serão definidos os requisitos em função de uma análise sobre os fluxos de dados e comportamentos de data centers tradicionais. Em seguida serão apresentadas três propostas de redes para data centers de nova geração. Baseado nos resultados desta análise tem-se o capítulo de (4) Conclusão.

## **2. Fundamentação Teórica**

Por se tratar de um estudo da infraestrutura de sistemas distribuídos, este trabalho está fundamentado na disciplina de Redes de Computadores, mas conceitos de tecnologias recentes e de data centers são indispensáveis para a compreensão do problema.

### **2.1 Redes de Computadores**

As redes de computadores surgiram das necessidades de interligação entre computadores e sua evolução confunde-se com o desenvolvimento do hardware como um todo. Data de 1969 a primeira rede de computadores por comutação de pacotes, a ARPANET, interligando as universidades de Los Angeles, Santa Bárbara, Stanford e Utah. A partir de então a proliferação das redes e suas interconexões formaram uma confederação que evoluiu para a Internet de hoje. Dentro das corporações as redes locais (LANs) começaram a permitir o compartilhamento de discos e impressoras. A década de 90 foi marcada pela grande informatização dos departamentos, a tecnologia da informação como um todo revolucionava processos e negócios. A Internet se popularizou nesta época graças ao surgimento da World Wide Web (www), que, através do HTML, HTTP, servidor Web e do browser, permitiu que diversos produtos e serviços fossem criados para a Internet. Paralelamente surge nos ambientes corporativos a Intranet, rede privada baseada no funcionamento da Internet. Ela permitiu que as tecnologias e benefícios da Internet fossem instalados internamente nas corporações. A Internet é hoje um componente central da infraestrutura mundial de telecomunicações.

As redes de computadores são baseadas em protocolos dispostos num modelos em camadas. Os protocolos encapsulam os dados a serem enviados, definem as regras da comunicação, o formato de mensagens e fazem a detecção e o tratamento de erros. Hardwares, softwares e protocolos de redes são organizados em camadas, e cada camada fornece serviços para a camada superior, através do encapsulamento. Tal modelo fornece modularidade, o que facilita a atualização de componentes do sistema. Cada protocolo pertence a

uma camada e quando tomados em conjunto são denominados pilha de protocolo.

OSI	TCP/IP	PROTOCOLOS
Aplicação	Aplicação	HTTP, SMTP, FTP, RPC, DNS, SNMP, outros
Apresentação		
Sessão	Transporte	TCP                      UDP
Transporte		
Rede	Rede	IP
Enlace	Enlace	Ethernet    PPP    SLIP    outros
Física	Física	protocolos dependem do meio físico

Figura 2.1 – O modelo OSI e o TCP/IP com alguns de seus protocolos

O modelo referência para redes de computadores é o modelo OSI (*Open Systems Interconnection*), porém o mais utilizado é o modelo TCP/IP (*Transmission Control Protocol/Internet Protocol*). O TCP/IP surgiu em 1979 na rede ARPANET com o objetivo de interconectar as diversas redes (com diferentes tipos de enlaces) que se formavam, mas apenas na década de 90, com o surgimento da Web foi adotado como padrão da Internet. A figura 2.1 compara o modelo OSI com o TCP/IP e relaciona alguns protocolos com suas camadas. O modelo TCP/IP foi extraído de [Kurose, Ross 2006] e apresenta cinco camadas, porém existem referências na literatura do mesmo modelo com quatro camadas, sem a camada física.

Cada camada do modelo é responsável por prover um serviço que será utilizado pela camada acima. A camada física depende do meio físico (fios de cobre, fibra ótica, cabo coaxial), pois é responsável em movimentar cada bit que está dentro do pacote fornecido pela camada de enlace, que transmite o pacote de nó em nó (roteador, switch, computador). Em função dos diferentes tipos de enlaces entre a origem e o destino, os datagramas são manuseados por diferentes tipos de protocolos da camada de enlace. A camada de rede é responsável pelas funções de repasse fim-a-fim (máquina de origem para a máquina destino) e roteamento dos pacotes entre os diversos nós da rede. O protocolo IP implementa o repasse e o endereçamento na Internet. A camada

de transporte fornece serviços de comunicação (entrega confiável ou não, controle de congestionamento, multiplexação e demultiplexação) diretamente aos processos de aplicativos que rodam em máquinas diferentes. Por fim, a camada de aplicação define como os processos de um aplicativo trocam mensagens entre si – tipos de mensagens (requisição, resposta), sintaxe, semântica e regras de comportamento.

Quanto à classificação das redes de acordo com a sua dispersão geográfica, podemos identificar as PANs, LANs, MANs e WANs.

PANs (*Personal Area Network*) – são redes pessoais e têm alcance de poucos metros. Utiliza conexões USB, wireless, Bluetooth, etc.

LANs (*Local Area Networks*) – são redes locais limitadas a centenas de metros (ou no máximo poucos quilômetros), utilizada nas organizações para prover conexão de alta velocidade entre processadores, periféricos, terminais e dispositivos de comunicação de uma forma geral em um único prédio (ou campus) [Dantas 2005]. Quando comparada às MANs e WANs, apresenta alta taxa de transmissão, baixa latência (tempo de resposta) e baixa taxa de erros. A tecnologia que predomina em redes locais é a Ethernet, que apesar de não ser orientada à conexão e fornecer um serviço não confiável à camada de rede, é barata e oferece velocidades variadas de 10Mbps, 100Mbps, 1Gbps e até 10 Gbps em Backbones.

MANs (*Metropolitan Area Networks*) – as redes metropolitanas provêm a interligação das redes locais à Internet em uma área metropolitana de uma determinada região. Oferecem a conexão entre centrais telefônicas, redes de acesso à internet, transmissão de sinais de TV, dados e voz. Dentre outras tecnologias existentes para as MANs, existe uma baseada no padrão Ethernet das LANs, a Metro Ethernet<sup>1</sup>, que permitem a criação de VPNs (*Virtual Private Networks*). Redes virtuais privadas utilizam a infraestrutura da Internet, mas

---

<sup>1</sup> Metro Ethernet é um termo genérico para descrever uma tecnologia de rede Ethernet em uma área metropolitana. É utilizada para conexão com a Internet pública ou entre corporações dispersas geograficamente.

mantém o comportamento de uma Intranet, garantido, assim, a segurança dos dados.

WANs (*Wide Area Networks*) – redes geograficamente distribuídas abrangem distâncias de países, continentes ou do planeta inteiro. Interligam várias redes metropolitanas diferentes através de tecnologias que dão suporte à QoS (*Quality of Service*), engenharia do tráfego e VPN (ex. MPLS - *MultiProtocol Label Switching*). Com este suporte é possível gerenciar a rede baseado especificamente no cliente e no tipo de dados transferidos.

## 2.2 Sistemas Distribuídos

Já no início da história das redes, as pessoas da área “*visualizavam que as aplicações pudessem se mover de um computador para o outro visando à utilização dos ciclos ociosos das máquinas que formavam a rede*” [Dantas, 2005]. Um sistema distribuído é, segundo [Tanenbaum 1995], “*uma coleção de computadores independentes que parecem um sistema único para o usuário*”. Coulouris [Coulouris 1994] define como “*um sistema onde os componentes de hardware e software, localizados em computadores interligados por uma rede, comunicam e coordenam suas ações somente através de troca de mensagens*”.

Sistemas distribuídos devem apresentar duas características inerentes: (1) a transparência na sua utilização, ou seja, a capacidade de apresentar-se aos seus usuários como uma entidade única, e (2) o alto grau de tolerância a falhas. Middlewares são as ferramentas da camada de aplicação que têm o objetivo de transparecer a complexidade e diversidade da infraestrutura de ambientes distribuídos aos usuários. É responsável pelo gerenciamento das tarefas (escalonamento) e da utilização de recursos dos aplicativos distribuídos.

Dentre as vantagens que sistemas distribuídos podem oferecer, podem-se destacar (1) a disponibilidade e a confiabilidade, já que a falha em um componente (rede, máquina ou programa) não implica na falha do sistema como um todo; (2) a flexibilidade, que é a facilidade de expansão e retração

dos componentes; (3) a velocidade e o desempenho (em padrões proprietários); e (4) a economia, ou seja, melhor relação custo/desempenho.

As mensagens trocadas entre aplicações distribuídas, dependendo da rede podem atrasar ou se perder. Em função disto, algumas desvantagens também são observadas, como (1) a maior dificuldade na garantia de segurança, (2) a complexidade no desenvolvimento de software distribuído, (3) a complexidade na gerência de recursos, e (4) o alto custo para implementar aplicações colaborativas (aplicações que utilizam a Internet como meio de comunicação entre os diversos clientes que estão colaborando - ex. e-mail, blogs, aplicações corporativas).

Sistemas distribuídos são formados por uma infraestrutura que engloba produtos e padrões que podem ser abertos (livres de royalties) ou proprietários. Padrões abertos oferecem várias facilidades, como a portabilidade e independência de fornecedores, já os proprietários oferecem velocidade e desempenho. Em um ambiente heterogêneo e desconectado como a Internet (MAN e WAN), a necessidade de portabilidade e novas tecnologias de segurança e transações definem os padrões abertos utilizados. Já num ambiente interno homogêneo de um datacenter (LAN), por exemplo, onde servidores devem manter uma relação fortemente acoplada, padrões proprietários possuem diferenciais que, dependendo do projeto, podem ser relevantes.

### **2.2.1 Virtualização**

Virtualização é basicamente o uso de software para simular hardware. Suas primeiras utilizações datam de 1967, dos mainframes /360 modelo 67 da IBM [Taurion 2009]. Esta tecnologia permite construir múltiplos servidores virtuais em uma única máquina física. Através disto possibilita um uso mais eficiente dos recursos, pois compartilha e distribui a carga de vários servidores virtuais em um único servidor físico.

Imagina-se a virtualização como uma camada de abstração entre o hardware e o software, conforme ilustrado na figura 2.2 extraída de [Veras 2009]. Esta camada é conhecida como *hypervisor* e entrega para o SO

(Sistema Operacional) um conjunto de instruções de máquinas equivalente a um processador físico. O servidor físico (virtualizado) passa a executar vários servidores virtuais - cada qual com seu SO e aplicações locais - chamados de MVs (Máquinas Virtuais ou *virtual machines*). O *hypervisor* é responsável pelo agendamento, gerenciamento da memória, e manutenção da máquina virtual. Também cria e garante o isolamento de partições nas MVs.

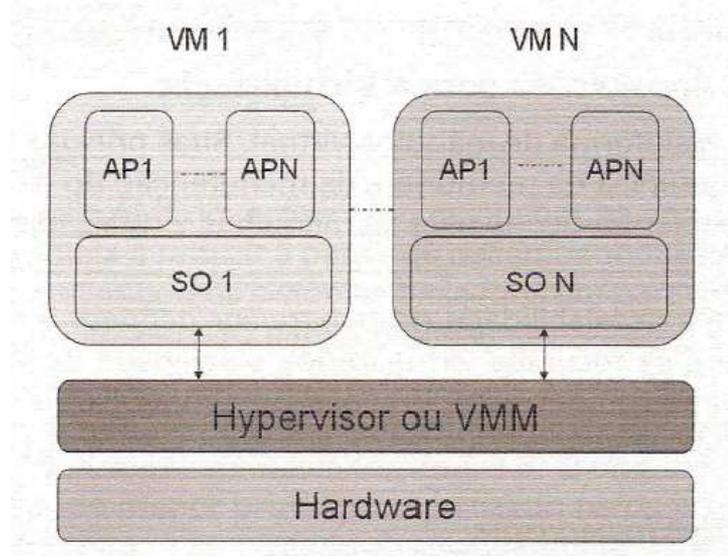


Figura 2.2 – Virtualização com hypervisor

O armazenamento e processamento das informações nas grandes organizações são geralmente realizados em servidores com arquitetura x86 – arquitetura comum em que são baseados todos os processadores intel e seus compatíveis, utilizados em residências e organizações. “A *virtualização de servidores trata da consolidação de vários servidores físicos e subutilizados em um único servidor com alto grau de utilização, reduzindo a complexidade do gerenciamento, o espaço físico e os requisitos de energia e refrigeração*” [Veras 2009].

A tecnologia de virtualização em arquiteturas x86 teve grande avanço a partir de 2005 com o desenvolvimento de processadores com extensões que facilitam a virtualização. Estas extensões, chamadas de *Hardware Assist*, melhoraram o desempenho dos sistemas virtualizados através da alteração dos níveis de acessos do processador aos seus recursos. Estes níveis são tradicionalmente chamados de anéis de proteção (*rings*) e são identificados de

0 a 3. O *ring 0* é o de maior privilégio e é utilizado pelo SO. O *ring 3* é o de menor privilégio e é utilizado pelos processos de usuários.

Ao utilizarmos um *hypervisor* num processador x86 comum, o *ring 0* fica sob sua utilização. O SO utiliza o *ring 1*, já que está rodando acima da camada de virtualização. Neste caso chamamos de paravirtualização e há perdas de desempenho, pois nativamente o SO utiliza instruções de nível *ring 0*. Para resolver esta questão, o *Hardware Assist* cria o *ring -1*, com novas instruções para utilização exclusiva do *hypervisor*, que passa a ter total prioridade sobre o hardware e não altera a interação da máquina com o SO, que continua utilizando o *ring 0*.

Esta tecnologia de virtualização de hardware fornece a (1) base para maximizar a eficiência e capacidade de virtualização de software; (2) aumenta o desempenho de MVs em um único sistema físico; e (3) aumenta a capacidade de usuários por máquina virtual.

A virtualização permite virtualizar toda a infraestrutura de TI. Além dos servidores, *storage*<sup>2</sup> e redes também são virtualizados através do *hypervisor*. Com a virtualização de redes, é possível criar redes virtuais dentro de um servidor ou através de múltiplos servidores. Uma rede virtual é formada de switches virtuais e placas de rede virtuais. Os switches virtuais podem se comportar como switches da camada 2, suportando inclusive VLANs (LANs Virtuais) com controle pelas portas. As placas de rede virtuais se comportam exatamente como uma placa real, com endereço IP e MAC próprios.

### 2.2.2 Grids

Dentre as diversas arquiteturas distribuídas, "*grid computing habilita organizações a trocar e compartilhar informações e recursos computacionais entre departamentos e organizações de forma segura e eficiente*" [Foster 2002]. Esta abordagem surgiu há cerca de quinze anos e é utilizada principalmente em projetos de governos e no meio acadêmico, sem fins lucrativos. Sua dimensão pode ser global, empresarial ou se confundir com um pequeno cluster.

---

<sup>2</sup> Storage se refere a dispositivos especializados em armazenamento. Podem ser discos (ópticos ou mecânicos) ou fitas.

Ian Foster [Foster 2002] elenca 3 características que são inerentes aos Grids: (1) coordena recursos que não estão vinculados a um controle central, ou seja, unifica recursos computacionais de organizações diferentes através da Internet; (2) utiliza protocolos e interfaces de padrões abertos que permitem negociar e gerenciar o compartilhamento dos recursos; e (3) seus serviços (segurança, confiabilidade e performance) podem ser configurados a fim de oferecer ao usuário determinado nível de qualidade de serviço (QoS).

Por unificar recursos de organizações diferentes, Grids são, em sua maioria, comandados por um gerenciador de recursos locais (ex. Condor, Globus) que permitem a virtualização dos recursos computacionais e executam processamento em lotes. Em outras palavras, os usuários submetem os lotes de tarefas à MV entregue pelo gerenciador, indicando alguns parâmetros, como o local de origem dos dados, a quantidade de processadores necessários, o tempo de duração e o local de saída para o resultado. O gerenciador de recursos coloca o pedido numa fila e aguarda a disponibilidade dos recursos, para então submeter a tarefa. A maioria dos Grids não permite aplicativos interativos nativamente, e aí se concentram os esforços da comunidade para oferecer recursos com latências mais baixas [Foster et al. 2008].

Baseado nos parâmetros de reserva dos lotes de tarefas entregue por cada usuário, é possível configurar determinados níveis QoS (segurança, confiabilidade e performance) para os serviços do grid. Em adição, SLAs (*Service Level Agreement*) podem ser editados para definir o que é compartilhado, quem é autorizado a compartilhar, e em que condições ocorrem o compartilhamento.

### **2.2.3 Nuvens**

O crescimento da internet nos últimos dez anos e o desenvolvimento de tecnologias e aplicações fizeram com que a quantidade de informação disponível na internet aumentasse de forma gigantesca. Segundo o EMC Corporation (multinacional americana de infraestrutura de TI) são criados por dia mais de 180 peta bytes (180.000.000.000.000.000 bytes) de informações na internet [EMC 2010]. Para armazenar e processar este volume crescente de dados, as empresas que trabalham com conteúdo web utilizam computação

distribuída justamente por sua característica de escalabilidade e economia. Em uma palestra no ano de 2006 [Zhang et al. 2010], Eric Schmidt do Google usou o termo “*cloud computing*” para descrever como gerenciava o imenso parque computacional de sua empresa, composto de 450 mil servidores interligados em 24 fazendas de servidores ao redor do mundo.

A partir de então a ideia incorporou vários conceitos e tecnologias – virtualização, grid, *utility*<sup>3</sup> e *autonomic*<sup>4</sup> *computing*, *web services* e SOA (*Service Oriented Architecture*). Computação em nuvem promete uma modificação profunda no mundo da tecnologia, permitindo a entrega de softwares de uma maneira mais atrativa, transformando a internet no repositório de aplicações e arquivos, permitindo que estes possam interagir um com os outros e conosco pela rede através de um modelo comercial, com garantia de serviços. Permitirá também que esta interação seja realizada de qualquer dispositivo conectado à Internet (celular, PC, notebook, PDA, smartphones, porta retrato digital, etc.). “*Computação em nuvem tem este nome como metáfora à Internet. A imagem da nuvem representa ‘todas as outras coisas’ que fazem a rede funcionar*” [Velte et al. 2010].

Esta transformação da computação em serviço também é comparada com a transformação da eletricidade em serviço nos primórdios da Revolução Industrial. Antes do surgimento da distribuição pública de eletricidade as próprias indústrias geravam sua energia através geradores próprios. Depois da instalação da rede pública de eletricidade, as indústrias não tinham mais conhecimento de como e onde a energia era produzida, apenas a consumiam. Comparando a geração elétrica com processamento e armazenamento digital e as redes de distribuição elétrica com a internet, podemos visualizar um futuro onde os usuários (pessoas e empresas) consomem processamento e armazenamento assim como consomem eletricidade – pagando apenas o que

---

<sup>3</sup> Utility computing se refere ao modelo de entrega dos serviços de computação (armazenamento e processamento) sob demanda.

<sup>4</sup> Autonomic computing são sistemas computacionais capazes de se auto configurar, de acordo com o ambiente.

foi consumido – com a vantagem de terem uma “tomada móvel”, representada em qualquer dispositivo ligado à Internet.

Segundo o *National Institute of Standards and Technology* [Mell, Grace 2009], computação em nuvem pode ser definida como “*um modelo que permite o acesso através da rede, de maneira conveniente e sob demanda, a um conjunto configurável de recursos compartilhados de computação (ex. redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados e liberados com o mínimo esforço do gerenciador de recursos ou do prestador de serviços*”.

Conforme [Taurion 2009], a implantação da nuvem pode ocorrer de três maneiras distintas:

- Nuvem pública – a infraestrutura da nuvem pertence a uma empresa (provedor) que disponibiliza para o público em geral no modo *pay-as-you-go* (paga o quanto consome).

- Nuvem privada – a infraestrutura da nuvem privada é interna a uma organização, não disponível ao público em geral. O objetivo principal de uma nuvem privada não é vender capacidade pela Internet, mas sim dar aos usuários locais uma infraestrutura ágil e flexível para suportar solicitações de serviços dentro de seu próprio domínio administrativo.

- Nuvem híbrida – composta em parte por uma nuvem privada e em parte por uma nuvem pública. Uma complementa a outra. A nuvem híbrida também pode permitir acesso remoto através de interfaces remotas, como a de Web services que a Amazon EC2 utiliza. A tecnologia de VPN permite a criação de máquinas em nuvens públicas se comportarem como sendo da rede interna (com o IP da empresa). Os dados críticos podem ser mantidos na nuvem interna e os dados leves em nuvens públicas.

“A Nuvem é um tipo de sistema distribuído e paralelo composto de uma coleção de computadores interconectados e virtualizados que são dinamicamente provisionados e apresentados como um ou mais recursos computacionais unificados baseado em acordos de níveis de serviço (SLAs) estabelecidos através de provedores de serviços e consumidores” [Buya et al. 2009]. Quanto aos tipos de serviços

entregues aos consumidores finais neste modelo podemos classificá-los de três maneiras:

- SaaS (*Software as a Service*) – é a entrega da aplicação através de serviços baseados em web. O grande diferencial está no modelo de pagamento por utilização mensal e não por licença, como é feito tradicionalmente. Além disto, não há custo para o usuário na aquisição de equipamentos, como servidores para a hospedagem da aplicação. SaaS retira da organização a gerência e manutenção da aplicação e dos equipamentos envolvidos, que passam a ser efetuados pelo provedor de serviços.

- PaaS (*Platform as a Service*) – facilita a criação, desenvolvimento e testes de aplicações web através da entrega de plataformas de desenvolvimento de software pela Internet.

- IaaS (*Infrastructure as a Service*) – os clientes alugam processamento, armazenamento, serviços de rede e qualquer outro recurso fundamental de computação para qualquer propósito.

Computação em nuvem é um conceito/tecnologia em desenvolvimento. Existem vários desafios a serem explorados e vencidos. Destacam-se quatro:

- A segurança e confiança da nuvem ainda não estão maduras. A ideia de deixar os dados e as aplicações em uma infraestrutura de terceiros não agrada muita gente. É necessário adquirir confiança no modelo, por isso, esforços nas áreas de consistência de autenticação, gerência de identidades, *compliance*<sup>5</sup>, acesso à tecnologia estão sendo feitos para garantir aos usuários garantia e transparência nas operações.

- Os padrões de conexão de sistemas e softwares para computação em nuvem ainda não estão definidos (como é o caso dos grids). Isto faz com que cada provedor crie as suas próprias tecnologias, ocasionando incompatibilidade entre aplicações de diferentes provedores. Porém existem esforços bem fundamentados para a definição dos padrões sendo os mais importantes o *Open Cloud Manifesto* [Open Cloud Manifesto 2010] e o *Cloud*

---

<sup>5</sup> *Compliance* se refere à regulamentação de acordo com as normas oficiais.

Standards [Cloud Standards 2010]. Um dos desafios principais é a padronização do formato das imagens virtuais e APIs (*Application Program Interface*) de migração.

- Apesar do grande crescimento das comunicações por banda larga, ainda existem altas latências na comunicação da Internet e aplicações que necessitam de curto tempo de resposta ainda não estão prontas para rodar na nuvem. O cenário é promissor para o mercado de infraestrutura de redes: a banda larga está cada vez mais barata e aceleradores de rede e outras tecnologias para aumentar o desempenho da rede tendem a se popularizar.

- A adoção de um novo modelo de TI baseado na nuvem exigem novas posturas e políticas na gerência dos recursos. Em um ambiente onde os recursos são virtuais, a governança e gerência de contratos e licenças precisam ser revistas e redimensionadas.

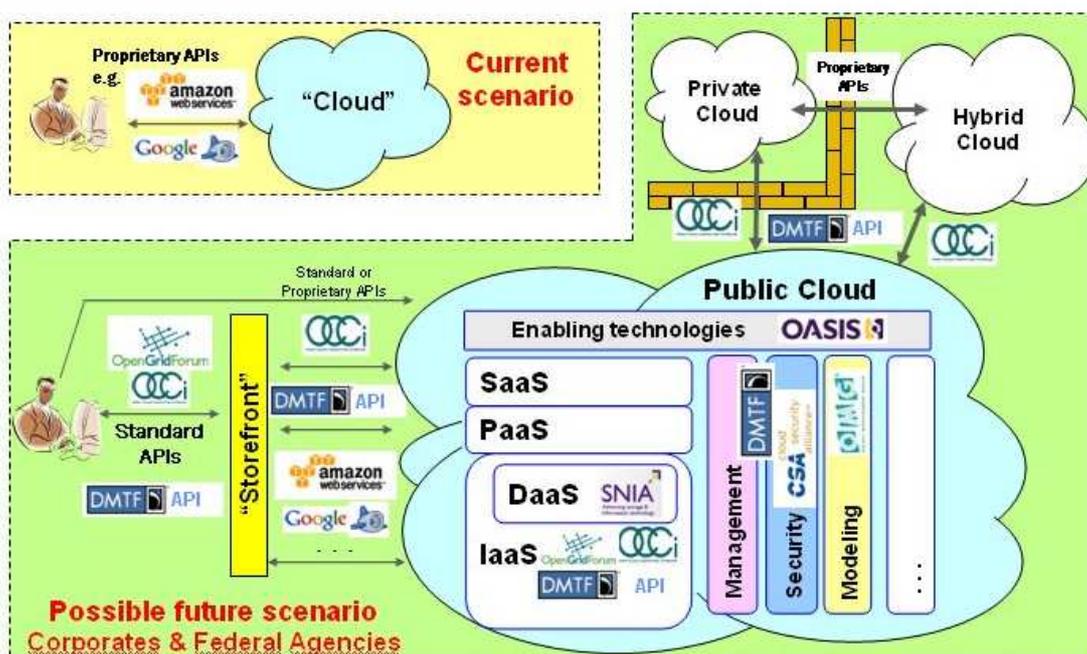


Figura 2.3 – Estágio atual e um futuro desejável para computação em nuvem

No canto superior esquerdo da figura 2.3 [Cloud Standards 2010], está representado o atual estágio de computação em nuvem, com poucos provedores, cada qual utilizando seus padrões proprietários e incompatíveis. No resto da figura está um possível e desejável futuro estágio onde toda a

comunicação e interação de serviços são feitas através de padrões abertos, com regras e políticas federais, com provedores especializados em infraestrutura e serviços, e com a interação entre nuvens públicas e privadas.

A orientação da computação em nuvem para atender o mercado é traduzida em sua arquitetura orientada ao mercado [Buyya et al. 2009]. Neste modelo existe a necessidade de atender o cliente de maneira confiável e legal. Para isso os mesmos conceitos de QoS e SLAs utilizados na computação em grid são adaptados e reformulados para atender os diversos serviços oferecidos pela nuvem com dinamicidade, permitindo alterações de acordo com as variações impostas pelo mercado.

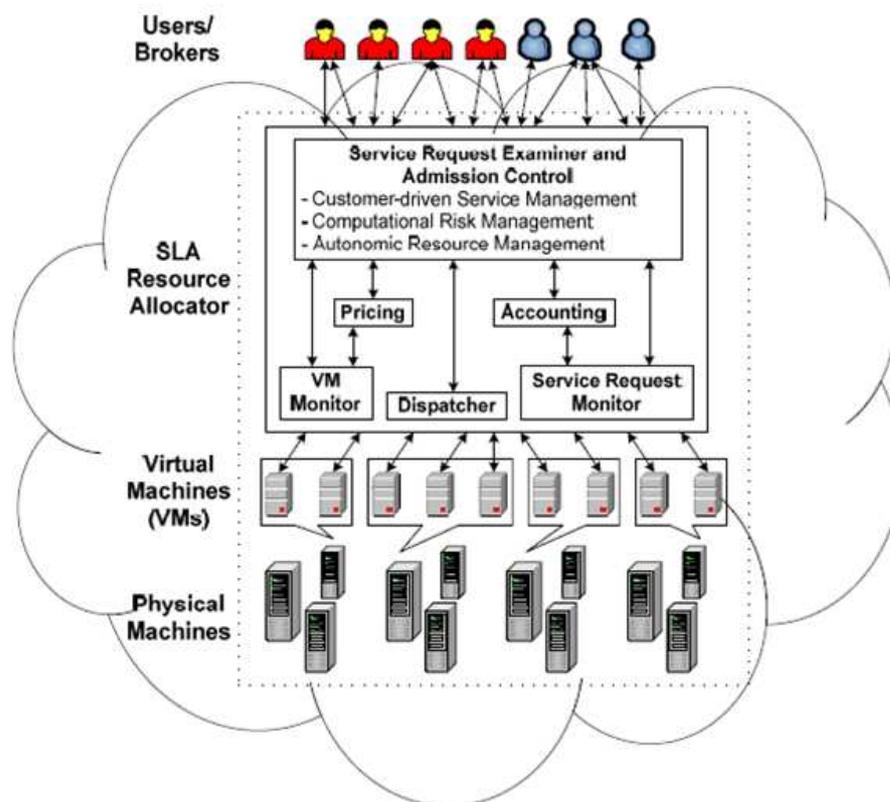


Figura 2.4 – Arquitetura de computação em nuvem orientada ao mercado

Na figura 2.4 [Buyya et al. 2009] os usuários são os *users/brokers* que solicitam e consomem serviços da nuvem através de seus dispositivos. As regras de como estes serviços são disponibilizados são definidas no contrato de serviço entre cliente e provedor, representado na figura pelo *SLA Resource*

*Allocator*. As MVs (Máquinas Virtuais) fornecem a flexibilidade de recursos e aplicações necessárias às variações de mercado, indispensável à computação em nuvem. As máquinas físicas são a infraestrutura que atende todo o modelo, composta de servidores e data centers distribuídos geograficamente.

Nuvens e grids são ambos sistemas distribuídos e compartilham diversos fundamentos: (1) transparência (abstração dos recursos), (2) disponibilidade (alto grau de tolerância a falhas), (3) flexibilidade (facilidade de expansão e retração dos componentes); (4) desempenho (várias máquinas possuem maior capacidade do que apenas uma) e (5) economia. Porém, é importante destacar as principais diferenças entre os dois modelos [Foster et al. 2008]: (1) processamento *on-line* nas nuvens e em lotes nos grids; (2) a virtualização na computação em nuvem é o elemento central, já em grids seu uso é mais discreto; (3) a segurança em computação em nuvem é obtida pelo isolamento dos serviços, em grids utilizam-se credenciais e perfis de usuários; (4) a centralização do controle nas nuvens e a descentralização nos grids.

Computação em nuvem é orientada ao mercado e em função disto os recursos são compartilhados por todos ao mesmo tempo, reduzindo a latência e permitindo que aplicações executem nativamente nas nuvens. Os grids não estão tão baseados em virtualização quanto as nuvens, pois cada organização deve manter o controle sobre seus recursos. Já na nuvem, os recursos são exclusivos do provedor, que utiliza integralmente a virtualização para entregar os serviços aos seus usuários.

Através da virtualização, computação em nuvem consegue o aumento da (1) eficiência na utilização de recursos, da (2) configurabilidade das aplicações, da (3) disponibilidade das aplicações, e da (4) maior capacidade de resposta dos recursos – acompanhamento e manutenção automatizados. Estas características são essenciais para a manutenção do SLA entre provedor e cliente.

Quanto maior a virtualização, maior a dificuldade de monitoramento. Em nuvens, a monitoração dos recursos requer um entendimento delicado entre o monitoramento da aplicação de negócio, do servidor físico, da máquina virtual e da manutenção do hardware. Quando maior a nuvem, maior o desafio de

monitoramento. Por outro lado, uma das grandes vantagens da nuvem é sua capacidade de gerenciamento automático, onde a qualidade de serviço é garantida pela abstração dos recursos.

## 2.3 Arquiteturas de TI

O entendimento das diferenças e das relações entre as arquiteturas de TI é fundamental para a compreensão da TI como um todo. Segundo [Veras 2009] a arquitetura de TI pode ser classificada em quatro grandes grupos interdependentes:

- Arquitetura Estratégica – é a lógica organizacional dos processos de negócio e da infraestrutura de TI. Reflete os requisitos de integração e padronização do modelo operacional da empresa.
- Arquitetura dos Processos – é o mapa das atividades que compõe os grandes processos de negócios.
- Arquitetura de Aplicação – define as aplicações individuais e suas interfaces.
- Arquitetura Tecnológica – define os padrões tecnológicos em que os serviços de infraestrutura se baseiam. Responde pelo nível de serviço entregue pela infraestrutura de TI às aplicações.

A forma de construir os processos define a arquitetura da aplicação a ser utilizada, que por sua vez define a arquitetura tecnológica. Portanto, a infraestrutura sempre vai depender da arquitetura utilizada pelas aplicações. Existem diversas arquiteturas de aplicação, as mais convencionais são a arquitetura cliente/servidor e a arquitetura de aplicação multicamada.

O modelo cliente servidor utiliza duas camadas (cliente e servidor) e ambas executam partes da aplicação, sobrecarregando o cliente. Já o modelo multicamada com adoção do padrão web na camada de *front-end* (interface

com o usuário), faz com que o cliente apenas execute o browser. Isto alivia a carga de computação do lado do cliente e conseqüentemente melhora o desempenho da aplicação.

As camadas do modelo multicamada podem ser identificadas como: (1) Camada Cliente – browser que permite a interação entre usuário e aplicação; (2) Camada de Apresentação – conjunto de servidores que formata as informações e dados oferecidos pela aplicação ao usuário. Para aplicações web a camada de aplicação é implementada por servidores web; (3) Camada de Aplicação – servidores que possuem as regras de negócio e acessam a camada de dados. É implementada por servidores de aplicação; (4) Camada de dados – *storage* de armazenamento dos dados da aplicação, utiliza diferentes SGDBs (Sistemas de Gerenciamento de Banco de Dados).

A figura 2.5 [Veras 2009] compara as duas arquiteturas de aplicação. Os servidores do modelo de camadas podem ser implementados individualmente, em vários servidores ou ainda em alguns servidores virtualizados que rodam múltiplas máquinas virtuais.

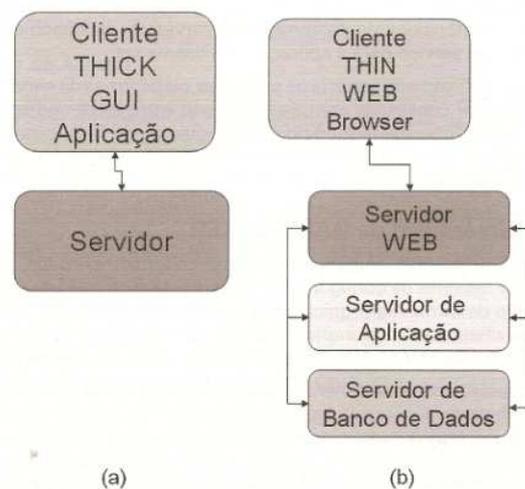


Figura 2.5 – Arquiteturas cliente/servidor (a) e multicamadas (b)

Portanto, as formas de se definir processos e aplicações definem a infraestrutura a ser utilizada. O alinhamento da TI com o negócio é realizado no nível da arquitetura estratégica e é repassada horizontalmente a todas as

demais camadas da arquitetura. Alterações em uma destas camadas influenciam todas as outras.

## 2.4 Data Center

*“O data center é um conjunto integrado de componentes de alta tecnologia que permite fornecer serviços de infraestrutura de valor agregado, tipicamente processamento e armazenamento de dados, em larga escala, para qualquer tipo de organização. É o elemento central da infraestrutura de TI de qualquer organização...”* [Veras 2009].

A história do data center começa em 1964 com o lançamento do mainframe IBM System 360 que centralizava todo o processamento da organização. O acesso ao mainframe era realizado através de terminais burros e grande quantidade de calor era gerada para a manutenção do sistema. Posteriormente, foi lançado o IBM VM/370 que introduziu o multiprocessamento que permitiu várias CPUs trabalharem como uma só. Era o início do conceito de virtualização. Outras inovações trazidas por este modelo foram o uso de memória virtual para abstração e mapeamento da memória real e utilização do sistema operacional VM (*Virtual Machine*), que já utilizava o conceito de *hypervisor*, muito difundido atualmente.

A década de 80 foi marcada por certa popularização da computação através do surgimento do IBM PC (*Personal Computer*), MS-DOS (Microsoft), UNIX da SUN (*Stanford University Network*) e seu protocolo de rede TCP/IP que rodava no topo da rede Ethernet. Agora, os acessos ao *mainframe* nas organizações não eram realizados apenas através dos terminais. Os PCs ou Estações de Trabalho emulavam o terminal de acesso e ainda permitiam rodar aplicações locais. A comunicação era entre estação de trabalho e servidor ou entre as próprias estações de trabalho. Era o início da era cliente/servidor. Para garantir a integridade dos dados, que agora sofriam múltiplos downloads por

usuário, surgiu o protocolo APPC (*Advanced Program-to-Program Communications*) que gerenciava melhor a comunicação entre mainframe-PC e permitia o coprocessamento<sup>6</sup> – primeiro passo para o processamento distribuído.

As conexões entre PCs fizeram surgir as LANs, com predominância das topologias Token-Ring e Ethernet. O barateamento dos equipamentos tornou viável à organização fornecer uma estação de trabalho para cada funcionário e a descentralização do gerenciamento deste parque marcava sua época. Servidores de arquivos x86 foram introduzidos às LANs, garantindo a integridade e compartilhando arquivos entre vários usuários da rede. Diversas soluções surgiram no paradigma cliente-servidor e caracterizavam-se pela descentralização das aplicações e dos equipamentos.

A partir de 1990, com o surgimento da Web, inicia-se nas organizações o uso da Internet como principal meio de conexão com o mundo externo. Sendo a Internet uma rede pública, aspectos de segurança mais sérios começaram ser considerados.

A TI trouxe enormes benefícios às organizações e a dependência dos recursos computacionais aumentou rapidamente. A dificuldade de gerenciar o crescente parque tecnológico trouxe de volta a necessidade de centralização da gerência dos recursos. Os servidores baseados em tecnologia x86 e os processadores multinucleados permitiram a recentralização ou consolidação dos recursos, trazendo de volta o data center, agora baseado em padrões abertos.

Neste processo várias tecnologias foram criadas e adaptadas para dar suporte à consolidação. Pode-se destacar (1) os servidores *blades*<sup>7</sup> e servidores em RACK baseados em processadores x86, (2) as tecnologias de virtualização e (3) o aumento da oferta de banda. Além disto, tendências em

---

<sup>6</sup> Coprocessamento se refere a utilização de um processador secundário (PC) para suplementar as funções do processador principal (mainframe) com o objetivo de acelerar a performance do sistema.

<sup>7</sup> Servido blade é um tipo de servidor desenhado para ocupar pouco espaço e economizar energia. São placas encaixadas em um chassi que provê energia, resfriamento e conexão.

modelos mais ágeis, como as novas arquiteturas de aplicações, a necessidade de redução do consumo de energia e refrigeração e a utilização de materiais que não degradem o meio ambiente (*Green IT*) facilitaram a consolidação do data center.

Os principais benefícios da consolidação, segundo [Hornby 2002] são: (1) melhorar os níveis de serviço; (2) aumentar a disponibilidade das aplicações; e (3) melhorar a gerenciabilidade do ambiente. O *International Data Corporation* [Cisco IDC 2009] inclui entre os benefícios a (4) redução no custo dos serviços e (5) a habilidade da organização de focar em novos mercados e oportunidades.

#### **2.4.1 Topologia e Classificação (TIA-942)**

A TIA (*Telecommunications Industry Association*) lançou a norma TIA-942 (*Telecommunications Infrastructure for Data Centers*) em 2005 e já está em sua quinta revisão. Esta norma indica os requisitos desde a construção até a pronta ativação do data center [TIA 2010]. É baseada em um conjunto de outras normas e é a principal norma existente para qualquer projeto de data center.

TIA 568 - Cabeamento

TIA 569 - Encaminhamentos e Espaço

TIA 606 - Administração

TIA 607 - Aterramento

ASHRAE - refrigeração HVAC

IEEE 1100 - ITE Aterramento

##### **2.4.1.1 Topologia**

O data center é um ambiente crítico tanto pelo custo dos equipamentos instalados como pelos dados e aplicações hospedados. Este ambiente deve ser seguro e automatizado. É comum o uso de “salas seguras” construídas com paredes anti-fogo, sistemas de detecção e prevenção de incêndios,

nobreaks para garantir energia elétrica, sistemas de controle de acesso, controle de temperatura e humidade, etc. A Topologia Básica de um data center, segundo a norma TIA-942, é a mostrada na figura 2.6 [IBD 2010].

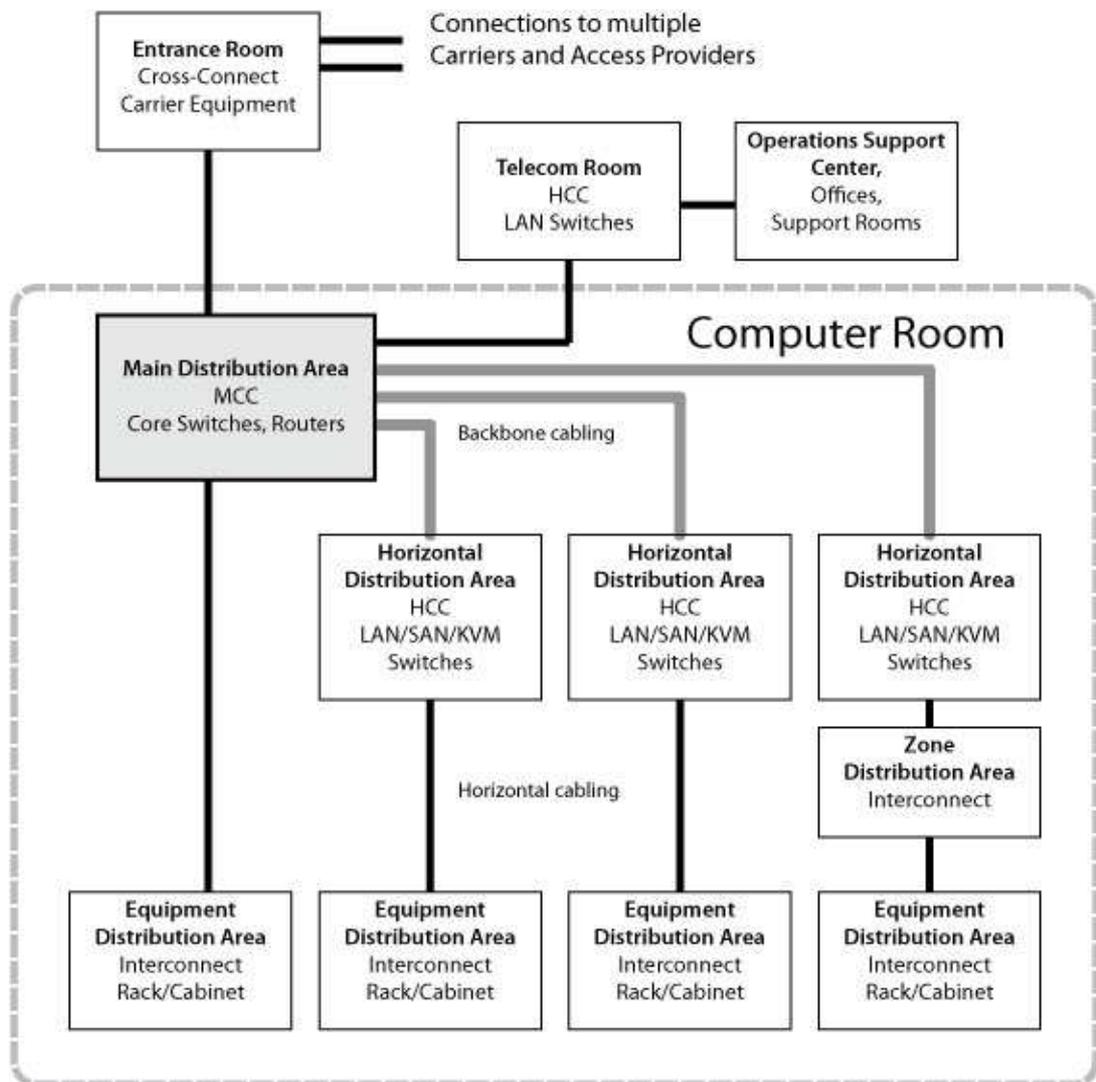


Figura 2.6 – Topologia de um data center, segundo TIA-942

- *Entrance Room* (ER) – Sala de entrada. Espaço de interconexão entre o cabeamento estruturado do data center e o cabeamento vindo das operadoras de telecomunicações.

- *Main Distribution Área* (MDA) – Área onde se encontra a conexão central do data center e de onde se distribui o cabeamento estruturado.

- *Horizontal Distribution Área* (HDA) – Área utilizada para conexão com área de equipamentos. Inclui o *cross connect*<sup>8</sup> horizontal e equipamentos intermediários.

- *Zone Distribuitor Área* (ZDA) – Ponto de interconexão opcional do cabeamento horizontal. Provê flexibilidade para o data center. Fica entre o HDA e o EDA.

- *Equipment Distribution Área* (EDA) – Área para equipamentos terminais (servidores, storage, unidades de fita) e equipamentos de rede.

#### **2.4.1.2 Classificação**

A norma TIA-942 define um padrão de classificação para os aspectos básicos do data center: (1) Layout e espaço físico; (2) Infraestrutura de cabeamento; (3) Condições ambientais; (4) Nível de disponibilidade. Este padrão é dividido em 4 camadas (*tiers*), de acordo com o risco de falhas:

- *Tier 1* – Básico: neste modelo não existe redundância nas rotas físicas e lógicas. Prevê uma distribuição de energia elétrica para atender a carga elétrica sem redundância. A interrupção parcial ou total das operações pode ser causada por falha elétrica, tanto da concessionária como da operadora de telecomunicações, ou por falha em equipamentos de rede (switch, roteadores, cabos). Possui um *downtime* (tempo de indisponibilidade) de 28,8 hr/ano.
- *Tier 2* – Componentes Redundantes: Os equipamentos de telecomunicações do data center e também os equipamentos de telecomunicações da operadora assim como os dispositivos da LAN-SAN devem ter módulos redundantes. O cabeamento do backbone principal LAN e SAN das áreas de distribuição horizontal para os switches do backbone deve ter cabo de cobre ou fibras redundantes. No *Tier 2* deve-se ter duas caixas de

---

<sup>8</sup> Nome genérico para qualquer sistema ou dispositivo que realiza uma conexão cruzada, isto é, inverte as posições de entrada e saída.

acesso de telecomunicações e dois caminhos de entrada até o ER. Devem-se prover módulos de no-break redundantes. É necessário um sistema de gerador elétrico para suprir toda a carga. Não é necessário redundância na entrada do serviço de distribuição de energia. Os sistemas de ar condicionado devem ser projetados para operação contínua 24 x 7 e incorporam redundância N+1. Possui um *downtime* de 22 hr/ano.

- Tier 3 – Sistema Auto Sustentado: Deve ser atendido por no mínimo duas operadoras de telecomunicações com cabos distintos. Devem-se ter duas salas de entrada (ER) com no mínimo 20 metros de separação. Estas salas não podem compartilhar equipamentos de telecomunicações e devem estar em zonas de proteção contra incêndios, sistemas de energia e ar condicionado distintos. Devem-se prover caminhos redundantes entre as salas de entrada (ER), as salas MDA e as salas HDA. Nestas conexões devem-se ter fibras ou pares de fios redundantes. Devem-se ter uma solução de redundância para os elementos ativos considerados críticos como o storage. Deve-se prover pelo menos uma redundância elétrica N+1. Possui um *downtime* de 1,6 hr/ano.
- Tier 4 – Sem Tolerância a Falhas: Todo o cabeamento deve ser redundante, além de protegido por caminhos fechados. Os dispositivos ativos devem ser redundantes e devem ter alimentação de energia redundante. O sistema deve prover comutação automática para os dispositivos de backup. Recomenda-se uma MDA secundária, desde que em zonas de proteção contra incêndio, devendo ser separadas e alimentadas por caminhos diferentes. Não é necessário um caminho duplo até o EDA. Deve-se prover disponibilidade elétrica 2(N+1). O prédio deve ter pelo menos duas alimentações de energia de empresas públicas a partir de diferentes subestações. O sistema de HVAC deve incluir múltiplas unidades de ar condicionado com a capacidade de resfriamento combinada para manter a

temperatura e a umidade relativa de áreas críticas nas condições projetadas. Esta camada é tolerante à catástrofes como enchentes, falta de energia, falta de água, queda de pequenas aeronaves, etc. Possui um *downtime* de apenas 0,4 hr/ano.

	Ano de Surgimento	Downtime (hr/ano)	Uptime (%)	Custo (\$/W)
Tier 1	1965	28.8	99.671	10
Tier 2	1970	22	99.749	11
Tier 3	1985	1.6	99.982	20
Tier 4	1995	0.4	99.995	22

Tabela 2.1 – *Downtime* por camada, segundo a TIA-942

A tabela 2.1 [Veras 2009] resume o *downtime* para cada camada da TIA-942 com um indicativo de custo de construção nos Estados Unidos. O data center padrão é o Tier 2, porém quanto mais críticas aplicações e os dados armazenados no data center, maior a necessidade de uma camada mais segura.

#### 2.4.2 Arquitetura

Antes de explorar a arquitetura de data center, vamos relembrar a arquitetura de TI, onde a forma de construir os processos define a arquitetura da aplicação a ser utilizada, que por sua vez define a arquitetura tecnológica. Portanto a arquitetura do data center sempre vai depender da arquitetura utilizada pelas aplicações. Considerando que os modelos de arquitetura de aplicação mais utilizados são o cliente/servidor e multicamadas, vamos estudar o modelo hierárquico de data center em camadas. Este modelo busca simplificar a complexidade de configuração e das tecnologias envolvendo o data center e também mantê-lo escalonável, confiável e com baixo custo. Pode ser dividido nas seguintes camadas:

(1) Camada de Núcleo – representa o núcleo da rede que suporta o data center, transporta grandes quantidade de dados de forma confiável e rápida. É o nível mais alto do modelo hierárquico. É implementada com dois switches de alto desempenho e escalabilidade. Utiliza protocolos de roteamento como o OSPF (*Open Shortest Path First*) e o EIGRP (*Enhanced Interior Gateway*

*Routing Protocol*). Faz o balanceamento de carga entre a camada de núcleo e agregação.

(2) Camada de Agregação – fornece os serviços de roteamento, filtros e acesso à rede WAN. Determina o melhor caminho para atender uma requisição de um determinado serviço da rede e encaminha-o ao núcleo que providencia o transporte. Realiza tanto a comunicação servidor-servidor quanto a comunicação servidor-cliente. Processa o protocolo Spanning Tree e é o default gateway. Também é chamada de camada 3 (modelo OSI).

(3) Camada de Acesso – é a camada de conectividade que controla o acesso aos recursos de data center para os usuários. É a mais baixa do modelo hierárquico e onde estão os servidores que suportam as camadas do modelo de aplicação. Também é chamada de camada 2 (modelo OSI).

Além destas camadas, dependendo do número de servidores do data center podem existir outros switches ToRs (*Top of Rack*) entre a da camada de acesso e todos os servidores de um único *rack*. Também existem equipamentos exclusivos para armazenamento que são conectados através de switches SAN (*Storage Area Network*) abaixo dos servidores da camada de acesso, que os controlam. São formados por unidades de armazenamento (servidores, discos e fitas) que suportam protocolos como iSCSI<sup>9</sup> (*Internet Small Computer System Interface*) e fibre channel<sup>10</sup>.

Este modelo em camadas permite maior gerenciamento, escalabilidade e independência entre as camadas e se torna mais resiliente à medida que cada uma destas camadas pode ser construída observando aspectos de alta disponibilidade.

O data center geralmente é construído com componentes redundantes para a obtenção de altos níveis de serviço para as aplicações. É comum o uso de duas conexões com o provedor de acesso e da utilização de switches aos pares a fim de aumentar a disponibilidade. Até mesmo o próprio data center

---

<sup>9</sup> Padrão baseado em IP para conectar periféricos de armazenamento e transferir dados baseados em comandos SCSI (interface paralela para conexão de periféricos a taxas de até 80Mbytes/s).

<sup>10</sup> Arquitetura serial de transferência de dados com alta largura de banda, utilizando fibras ópticas.

pode ser construído aos pares, em locais distantes, garantindo os dados através da replicação em caso de desastre. Estes fatores vão de encontro com a classificação em Tiers, segundo a norma TIA-942.

Para representação e separação das camadas da aplicação são utilizadas VLANs (redes locais virtuais). Uma VLAN é um domínio broadcast de rede configurável através de software. Usuários são agrupados em um mesmo domínio de broadcast independente de cabeamento físico. Este modelo garante a resiliência fazendo o balanceamento de carga entre as camadas e garantindo a segurança através da adoção de firewalls entre as camadas. Porém, como veremos mais à frente no capítulo 3, este modelo impõe certas limitações de tráfego.

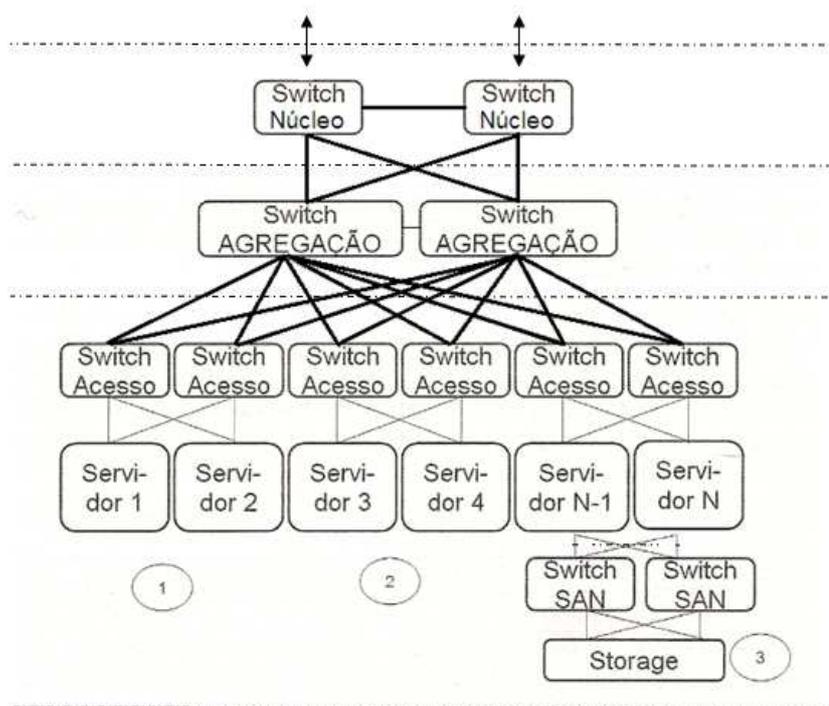


Figura 2.7 – Arquitetura de data center

A figura 2.7, adaptada de [Veras 2009] representa o modelo hierárquico de data center utilizando uma topologia colapsada, ou seja, o switch de núcleo centraliza a rede e a separação entre as camadas da aplicação é feita através de VLANs. As linhas grossas representam conexões feitas a 10 Gigabit Ethernet e as finas em 1 Gigabit Ethernet. As camadas verticais 1, 2, 3

representam respectivamente a camada de apresentação, camada de aplicação e camada de dados. As camadas horizontais representam a camada de núcleo, de agregação e de acesso.

Cabe comentar que existem outros tipos de modelos de arquiteturas para data center. Como já foi comentado, a arquitetura do data center depende da arquitetura das aplicações. Aplicações que utilizam alto poder de processamento, como computação científica ou análises financeiras utilizam supercomputadores. Neste caso o modelo de data center baseado em clusters utilizando servidores x86 é uma opção cada vez mais utilizada principalmente pelo baixo custo, escalabilidade e disponibilidade.

### **3. Desenvolvimento**

A proposta deste trabalho consiste em identificar os requisitos que devem estar presentes em arquiteturas de data centers que garantam as características e necessidades da infraestrutura do modelo de computação em nuvem.

Primeiramente vamos identificar os requisitos de infraestrutura para computação em nuvem, em seguida analisaremos o tráfego e definiremos os pontos críticos das arquiteturas de redes tradicionais para data center, levantaremos os requisitos para redes de data centers e finalmente estudaremos três propostas de arquiteturas (DCell, VL2 e PortLand) que buscam melhorar o desempenho do data center.

#### **3.1 Requisitos de infraestrutura para computação em nuvem**

Do ponto de vista do hardware [Armbrust et al 2009], computação em nuvem introduz três novos aspectos: (1) A ilusão de recursos computacionais infinitos para o usuário, disponíveis sob demanda sem necessidade de provisionamento prévio muito antecipado; (2) A independência da aquisição de infraestrutura de hardware por parte dos usuários, permitindo aumentar os recursos de hardware apenas quando há um aumento nas suas necessidades; (3) A possibilidade de pagar o uso de recursos computacionais baseado em consumo (processador/hora, armazenamento/dia) e liberá-los assim que necessário.

A elasticidade para obter e liberar recursos é um dos aspectos chaves da computação em nuvem, sendo uma das principais diferenças quando comparada com computação em grid. Este fator exige recursos abundantes – elevado número de servidores e infraestrutura de rede qualificada – que garantam os picos das demandas de diversos serviços ao mesmo tempo. *Supervisors* alocam dezenas de MVs com endereços fixos de IP e MAC em cada máquina física. Num data center de 100 mil de servidores físicos cada qual suportando 20 MVs, há necessidade de endereçamento para 2 milhões de

IPs e MACs, que, num ambiente em nuvem, necessitam interagir entre si e entre solicitações externas.

A elasticidade encontrada na computação em nuvem deve ser acompanhada de agilidade, a fim de diminuir o tempo de pré-reserva dos recursos. Isto é, a solicitação do usuário por mais recursos deve ser atendida imediatamente pela nuvem. Agilidade também é fundamental e alvo de muitos estudos. Segundo [Greenberg et al. 2009 apud Verdi et al. 2010], *“agilidade é a capacidade de realocar dinamicamente servidores entre os serviços oferecidos pela nuvem com o propósito de otimizar a alocação da carga de trabalho”*.

Somado aos fatores expostos acima, os processos de negócios encontrados nas organizações estão cada vez mais dinâmicos e demandam aplicações mais ágeis, capazes de reconfiguração dinâmica. SOA (*Service Oriented Architecture*) é um estilo de arquitetura de aplicação que está surgindo como alternativa aos modelos cliente-servidor e multicamada. SOA pressupõe que as funcionalidades do negócio ou lógica da aplicação são tornadas disponíveis aos seus usuários ou a outras aplicações como um conjunto de serviços reutilizáveis.

Um serviço, do ponto de vista da arquitetura SOA, é uma função de um sistema computacional que é disponibilizada para outro sistema por meio de uma implementação encapsulada de software, acionada por uma ou mais interfaces, satisfazendo certas obrigações pactuadas que asseguram regras de interação bem definidas. Um serviço pode ser reutilizado em outros processos de negócios, aplicações compostas ou mesmo por outros serviços. O conjunto de serviços fica disponível para acesso externo através de um “barramento de serviço” e as interconexões entre as aplicações são feitas através da interface do serviço. A figura 3.1 [Veras 2009] esquematiza a SOA.

SOA é a evolução da arquitetura de software para abordar modularização, reutilização, extensibilidade e flexibilidade. Para a construção escalável de plataformas de computação em nuvem, é preciso aproveitar SOA na construção de componentes reutilizáveis, interfaces baseadas em padrões e soluções extensíveis de arquiteturas [Zhang et al. 2009].



Figura 3.1 – SOA Service Oriented Architecture

Segundo a interdependência entre as camadas de arquitetura de TI (seção 2.3), a orientação a serviços é repassada para a camada seguinte da arquitetura de aplicação, exigindo que a arquitetura tecnológica (infraestrutura) atenda os requisitos demandados pelas aplicações. Passou a ser chamado pela Cisco de SOI (*Service Oriented Infrastructure*) os serviços de processamento, armazenamento, rede e segurança fornecidos sob demanda à SOA.

Os serviços de nuvem devem ser construídos [HPC in the Cloud, 2010] sobre uma infra-estrutura altamente resiliente (elástica) - tanto de hardware e software - que deve ser configurada para manter alta disponibilidade. As altas velocidades de ligações entre servidores físicos devem estar associadas com a tecnologia de balanceamento de carga para garantir que mesmo as mais complexas aplicações atendam a demanda de negócios.

As plataformas em nuvem também devem possuir inteligência embutida. A instrumentação dinâmica, por exemplo, permite que a infraestrutura de nuvem possa reagir às mudanças nas exigências da aplicação. Com a utilização de agentes para monitorar o desempenho da aplicação, a infraestrutura torna-se escalável para cima ou para baixo – automaticamente – a fim de atender às necessidades imediatas da aplicação.

A utilização de recursos físicos geograficamente dispersos melhora a recuperação de desastres e a continuidade dos negócios, e deve ser parte integrante de um serviço na nuvem. Os usuários devem avaliar as capacidades de recuperação de desastres do provedor de serviços e receber garantias – sob a forma de acordos de nível de serviço (SLA) – que, mesmo nas piores situações possíveis, serão reinicializados rapidamente. Os clientes corporativos devem poder escolher um datacenter regional específico para implantar sua nuvem, permitindo reduzir os custos de largura de banda e que satisfaça as demais exigências regulamentares.

Resumindo os fatores explanados acima, teremos os seguintes requisitos de infraestrutura para atender computação em nuvem:

- ✓ elasticidade: suportar e entregar um grande número de equipamentos;
- ✓ alta disponibilidade: suportar falhas nos equipamentos;
- ✓ agilidade: suportar liberdade de migração de MVs e comunicação entre todos os servidores;
- ✓ cobrança: deve ser passível de mensuração para posterior cobrança dos clientes;

### **3.2 Data Center**

*“As principais nuvens estão sendo construídas em cima da infraestrutura tecnológica de empresas que dispõem de um vastíssimo parque computacional”* [Taurion 2009]. O modelo de computação em nuvem pretende transformar o data center em um ambiente auto ajustável, eficiente na alocação e escalabilidade de serviços e econômico no consumo de matéria prima (energia e equipamentos).

À medida que as tecnologias de virtualização evoluem, a importância das redes aumenta. O advento dos servidores *blades* e da virtualização estão fazendo com que a infraestrutura de redes suporte uma alta densidade de configurações. A virtualização está remodelando os produtos de rede e a forma

como eles são entregues. Um data center que suporte computação em nuvem – caracterizado pela habilidade de ser ágil, rápido e seguro – será atendido por uma tecnologia de virtualização com suporte à mobilidade das MVs. Em tal ambiente, a virtualização será implantada em toda a infraestrutura de TI, incluindo servidores, armazenamento e redes. O novo modelo será hábil a prover a migração de máquinas virtuais de missão crítica dentro do data center e ao redor do mundo.

Há duas opções para a construção de uma infraestrutura de rede para servidores em grande escala. Uma opção de hardware aproveita protocolos de comunicação especializados, tais como InfiniBand ou Myrinet. Esta opção permite a escalabilidade em milhares de nós com alta largura de banda, porém seus equipamentos são caros, sua tecnologia é proprietária, e não são naturalmente compatíveis com as aplicações TCP/IP. Outra opção, mais adequada para a escala de equipamentos que garantam a elasticidade da computação em nuvem, é a utilização de switches e roteadores Ethernet comoditizados. Esta abordagem aproveita-se do custo dos equipamentos, do suporte já familiar de gerenciamento de infra-estrutura, e da compatibilidade com aplicações, sistemas operacionais e hardware legados.

### **3.2.1 Análise de tráfego de rede no data center**

De acordo com o tipo de aplicação, um data center típico possui dois tipos de tráfego: (1) o tráfego que entra e sai do data center e (2) o tráfego que é gerado e flui internamente no data center. Por exemplo, aplicações de busca (ex. Google) através de *MapReduce*<sup>11</sup>, produzem elevado fluxo interno para realizar indexações e sincronizações. Por outro lado aplicações de vídeo (ex. YouTube) causam elevado fluxo externo através da transferência de arquivos entre data center e usuário.

A matriz de tráfego é utilizada para determinar quem envia a quantidade de dados pra quem e quando dentro de um data center. Um estudo [Greenberg et al. 2009] baseado nos protocolos de gerenciamento de rede *Netflow* e *SNMP*

---

<sup>11</sup> Algoritmo patenteado pelo Google para o processamento de grandes conjuntos dados de forma distribuída.

(*Simple Network Management Protocol*) em data centers operando em nuvem demonstrou 4 tendências:

1 – A taxa de tráfego entre servidores dentro do data center é quatro vezes maior do que o tráfego que entra e sai do data center.

2 – Devido a configurações de desempenho (*oversubscription*), a distribuição de carga de trabalho que envolve a troca de um volume alto de dados é preferencialmente realizada nas áreas onde a largura de banda disponível entre servidores é maior (dentro do mesmo *rack*, depois dentro da mesma VLAN e, de menor preferência, em servidores mais distantes). Quanto mais distribuídos os dados pior o desempenho das aplicações.

3 – A demanda de largura de banda entre servidores dentro do data center está crescendo mais rápido do que a demanda entre hosts externos.

4 – A rede é um gargalo à computação. Frequentemente foram encontrados switches ToR (*Top of Rack*) com uplinks acima de 80% de utilização.

Este estudo foi instrumentalizado em um cluster altamente utilizado com 1500 nós. Os servidores foram distribuídos de forma uniforme através de 75 switches ToRs conectados hierarquicamente. Foram coletados logs ao nível de soquete de todas as máquinas por mais de dois meses.

Outros estudos revelaram que a matriz do tráfego é altamente variável e dominada por rajadas. Observando o tráfego em switches de 19 data centers, [Benson et al. 2009] concluiu-se que a maior carga média encontra-se nos switches do núcleo e diminui progressivamente à medida que descemos em direção aos switches de acesso. Entretanto, as maiores perdas são encontradas em alguns switches ToRs, indicando o comportamento típico de rajadas nos switches de acesso. Isto revela que a descoberta de rotas por caminhos alternativos poderiam evitar a maior parte das perdas resultantes de *switches* congestionados.

Para avaliar o congestionamento em uma rede de data center [Kandula et al. 2009] foi utilizado um cluster com 1500 servidores interligados por 150

switches com fluxos de aplicações *MapReduce*. O resultado demonstrou que 86% destes switches sofreram congestionamento de pelo menos 10 segundos e 15% apresentaram períodos de congestionamento de pelo menos 100 segundos. A figura 3.2 [Kandula et al. 2009 apud Verdi et al. 2010] ilustra esta situação. Através dela pode-se perceber que os períodos de congestionamento de até 10 segundos (círculos azuis) estão concentrados em vários switches em determinados períodos do dia, indicando picos de demanda originados das aplicações. Já os longos períodos de congestionamento (quadrados vermelhos) tendem a ser localizados em um conjunto menor de switches.

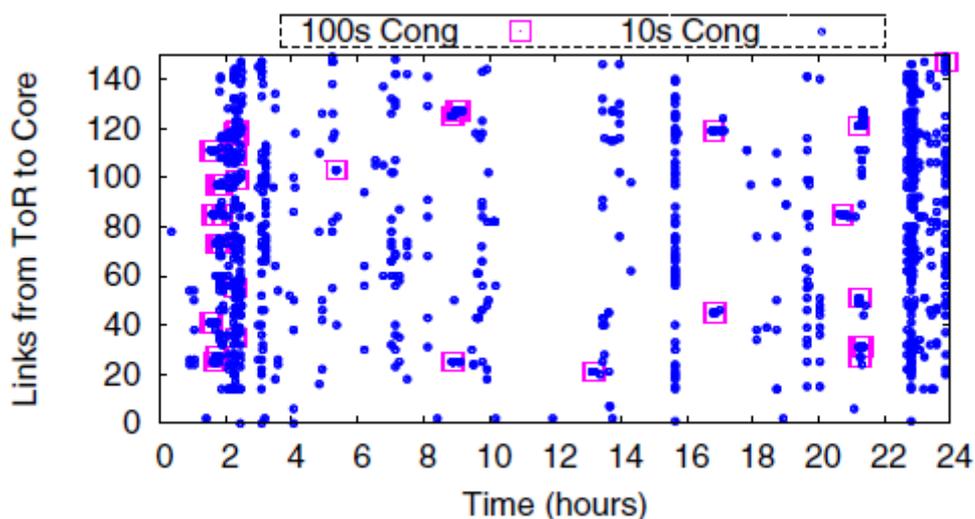


Figura 3.2 – Congestionamento da rede

A figura 3.3 [Kandula et al. 2009] mostra que a maioria dos congestionamentos tende a ter curta duração. De todos os congestionamentos com mais de um segundo, 90% não passa de dois segundos. Dos 10 % restantes, houve 665 com mais de 10 segundos.

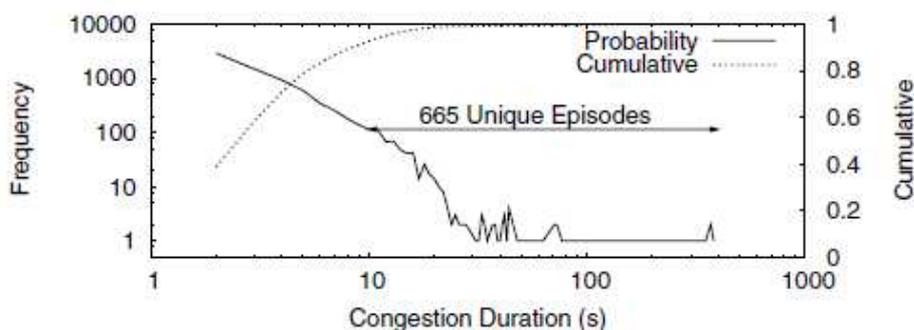


Figura 3.3 – Tamanho dos congestionamentos.

Os fatores acima dificultam a previsão do comportamento do tráfego e o uso de técnicas tradicionais de engenharia de tráfego, demandando novos mecanismos para reduzir os problemas de congestionamento e prover garantias de uniformidade nas taxas de transferência entre qualquer par de servidores, independente de sua localização.

### **3.2.2 Limitações das arquiteturas tradicionais**

Considerando o modelo hierárquico de data center (Seção 2.4), a configuração dos elementos de rede (roteadores e switch) impõe limitações à agilidade do sistema, funcionando como um gargalo na distribuição dos serviços. A arquitetura atual se limita na capacidade de mapear as solicitações de aplicações arbitrariamente em um subconjunto de recursos físicos disponíveis. Para evitar overheads (por exemplo, inundações de pacotes e requisições ARP) e para isolar diferentes serviços ou grupo de servidores lógicos (por exemplo, e-mail, busca, servidores front-end, back-end), os servidores são divididos em *Virtual LANs* (VLANs).

Um domínio de camada 2 é definido por um par de roteadores de acesso, conforme a figura 3.4 [Greenberg et al. 2008 apud Verdi et al 2010], e é dividido em várias VLANs, sendo uma subrede IP por VLAN. Quando um servidor de origem deseja se comunicar com outro servidor destino na mesma LAN (ou VLAN), ele precisa descobrir o endereço físico (MAC - *Media Access Control*) da placa de rede deste servidor destino. Se a origem conhece o endereço lógico (IP) do destino, ela envia, antes do pacote, uma mensagem *broadcast* utilizando o protocolo ARP (*Address Resolution Protocol*) a todos os nós da rede. Aquele nó que identificar o seu IP na mensagem, responderá para a origem informando o seu endereço MAC. A origem armazena este MAC numa memória cache e o utiliza sempre que precisar para encaminhar pacotes IP através de *frames* Ethernet. O ARP é frequentemente utilizado para traduzir endereços entre as camadas 2 e 3. Este *broadcast* gerado pelo ARP limita o número de servidores nas VLANs criadas no domínio de camada 2 do data center à algumas centenas de máquinas.

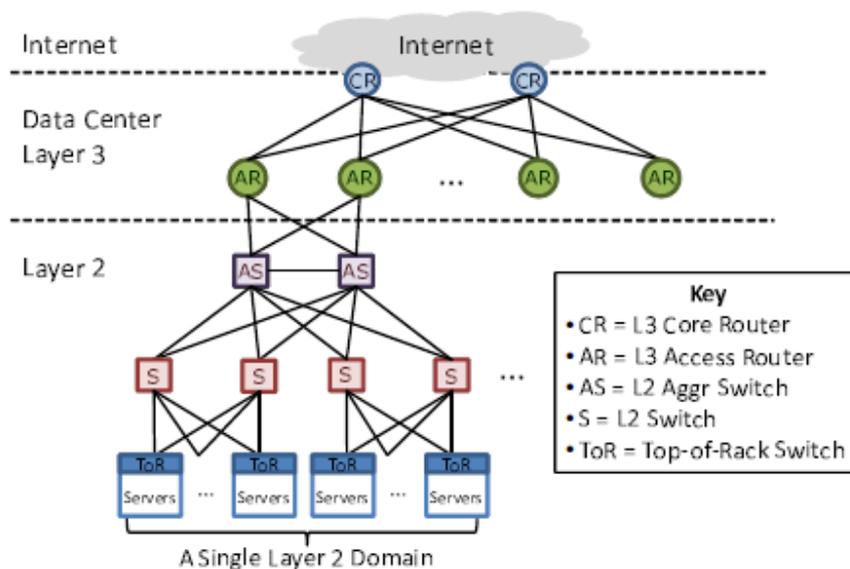


Figura 3.4 – Organização tradicional de um data center

A migração de uma MV para uma VLAN diferente exige a atribuição de um novo endereço IP com base no número do novo switch de subrede, uma operação que quebra todas as ligações TCP abertas para o host e invalida qualquer estado da sessão mantido pelo data center. Isto impede a migração de MVs entre VLANs e causa a fragmentação de recursos

Além disso, quando as subredes hospedam múltiplos serviços, a arquitetura atual não previne que a sobrecarga de um serviço influencie em outro serviço. Todos os serviços que rodam numa subrede são atingidos pela sobrecarga de apenas um destes serviços. Isto é chamado de alocação estática de serviços.

*Oversubscription* é um método utilizado para fazer um dimensionamento adequado da rede economizando na quantidade de enlaces e equipamentos. Parte do princípio de que as conexões de dados tendem a ser utilizadas de forma intermitente, ou seja, de que os dados fluem em rajadas. Refere-se à taxa de largura de banda garantida alocada por usuário (conexão) e é calculada baseada na capacidade da placa de rede do servidor, na quantidade

e capacidade dos links e *uplinks*<sup>12</sup> do switch. Independentemente da capacidade dos equipamentos, ao encontrarmos um fator de *oversubscription* superior a 1, muito comum em data centers tradicionais [Al-Fares et al. 2008], teremos um desempenho reduzido na vazão e latência entre os servidores. A rede torna-se, então, um gargalo para matrizes de tráfego onde uma grande parte dos servidores transmite no máximo da sua capacidade, causando congestionamento nos switches e o consequente descarte de pacotes. Existem diversas arquiteturas sem *oversubscription* (*oversubscription* = 1), que garantem a probabilidade de bloqueio zero para qualquer tráfego de entrada: *Clos* e *Fat Tree* são exemplos e serão discutidas mais adiante.

### 3.2.3 Requisitos de redes para data centers em nuvem

A viabilidade de um data center na nuvem, depende da alocação dinâmica de recursos entre um vasto parque de servidores. A rede, em particular, deve ser capaz de atribuir qualquer servidor a qualquer serviço, independente da sua localização e natureza, com alta disponibilidade. No cerne desta questão está a necessidade de desacoplar as aplicações da própria infraestrutura e transformar o data center em um sistema ágil, de alto desempenho e eficiente.

Para atender os requisitos de um data center operando em nuvem, o seu design deve ser direcionado para alcançar as seguintes metas [Al-Fares et al. 2008]:

- Largura de banda de interconexão escalável: cada servidor deve poder se comunicar com qualquer outro na rede através da capacidade total de sua placa de rede. Ou seja, não pode existir gargalo na rede.
- Economia de escala: a tecnologia de rede Ethernet existente comumente (switches e roteadores) deve ser utilizada em todos os níveis da rede para reduzir o custo capital e operacional.
- Compatibilidade: o novo sistema deve manter a compatibilidade com os sistemas legados. A tecnologia de rede Ethernet e o

---

<sup>12</sup> Utilizado para interligar switches.

endereçamento IP devem ser beneficiados com o novo sistema de interconexão.

No contexto de arquiteturas orientadas a serviço, o data center dinâmico (DDC – *Dynamic Data Center*) é uma proposta de modelo de data center [Cisco, IDC 2009] que poderá suportar integralmente a arquitetura SOI fornecendo os recursos sob demanda da arquitetura SOA e da computação em nuvem. Mais especificamente, o DDC permitirá que:

- As imagens do sistema operacional (SO) e das aplicações sejam armazenadas centralmente;
- O provisionamento de servidores seja feito *on-demand*;
- Uma única ferramenta de gerenciamento controle todos os recursos tecnológicos;
- Os serviços sejam automaticamente reinicializados em caso de falha do recurso de hardware;
- As flutuações de carga de trabalho (*workloads*) sejam automaticamente atendidas pelos servidores.

O DDC ainda é uma promessa, mas o avanço da virtualização e das suas funcionalidades poderá viabilizar o modelo rapidamente.

Os requisitos básicos de confiabilidade e desempenho devem estar alinhados com o aumento da escalabilidade e agilidade inerentes à computação em nuvem. Somados a estes requisitos, a redução de custos e o gerenciamento centralizado completam os 6 requisitos identificados para uma arquitetura de rede de data center que suporte computação em nuvem [Verdi et al. 2010]:

1. Agilidade: a migração de MVs deve ocorrer entre quaisquer servidores dentro do data center independente de sua localização;
2. Desempenho: o limite da capacidade de comunicação entre servidores deve ser determinado pela placa de rede, ou seja, não podem existir gargalos na rede. Todas as conexões devem possuir vazão e latência constantes, além de isolamento. Este fator também ajuda na simplificação do desenvolvimento de aplicações;

3. Escalabilidade: o objetivo é gerenciar milhões de IPs e MACs ao mesmo tempo. Portanto, as tabelas de encaminhamento devem ter tamanho gerenciável e *broadcasts* devem ser evitados;
4. Controle: controle flexível, capaz de introduzir com facilidade novos nós na rede. Este controle deve prover monitoramento, resolução de problemas e segurança, através de um software de rede (plano de controle e pilhas de protocolos) simplificado;
5. Confiabilidade: o grande número de equipamentos aumenta as chances de falha. O sistema deve garantir tolerância às falhas comuns em data centers – de hardware, software e/ou energia.
6. Custo: O sistema deve utilizar equipamentos comoditizados para redução do custo de capital. O consumo de energia proporcionalmente ao processamento é um desafio que promete economia no consumo mensal.

Existem diversos estudos e arquiteturas que buscam atender os requisitos identificados acima, porém atendê-los de uma maneira global, ou seja, todos os requisitos ao mesmo tempo é um desafio, pois a implantação de algumas soluções causa o detrimento de outras.

### **3.3 Propostas de arquiteturas para data centers**

O desenvolvimento de pesquisas para novas propostas de arquiteturas de redes para data center tem tido grande crescimento nos últimos anos. A solução das atuais limitações que impedem o pleno desenvolvimento da computação em nuvem é a grande motivação destes trabalhos. Neste trabalho foram identificados diversos estudos, tais como SEATTLE [Kim et al. 2008], Moonson [Greenberg et al. 2008], BCube [Guo et al. 2009], porém vamos estudar mais profundamente as arquiteturas DCell [Guo et al. 2008], VL2 [Greenberg et al. 2009] e PortLand [Mysore et al. 2009].

#### **3.3.1 DCell**

DCell é uma estrutura de rede criada para interconectar um número exponencialmente alto de servidores em um ambiente de data center [Guo et

al. 2008]. Seus objetivos são três: escala, tolerância a falhas e alta capacidade de transmissão. Sua estrutura é recursiva, ou seja, um DCell de alto nível é composto de outros DCells de mais baixo nível. Todos os DCells de mesmo nível são conectados uns aos outros. A escala de DCell dobra exponencialmente a medida que o grau do nó aumenta. DCell é tolerante a falhas, pois não possui pontos únicos de falha e quando ocorre uma falha em um link ou nó, utiliza um protocolo de roteamento distribuído que escolhe o caminho mais curto. DCell também provê alta capacidade de rede comparada à estrutura tradicional em árvore.

Uma rede de data center baseada em DCell possui quatro componentes que trabalham em conjunto: (1) a própria estrutura de rede DCell; (2) um algoritmo de roteamento eficiente e distribuído que explora esta estrutura de rede; (3) roteamento tolerante a falhas; e (4) um esquema de expansão física incremental.

A estrutura de rede utiliza servidores com múltiplas portas de rede e mini-switches de baixo custo para construir sua arquitetura recursiva. Cada servidor é conectado a outro servidor e a um mini-switch através de links bidirecionais. A conotação  $DCell_k$  (sendo  $k \geq 0$ ) indica o nível  $k$  do DCell. A figura 3.5 [Guo et al. 2008] demonstra como DCells em diferentes níveis são construídos. Neste exemplo  $DCell_0$  é a pedra fundamental para a construção de um DCell maior. Ele possui  $n$  servidores (neste caso  $n=4$ ) e um mini-switch que conecta todos os servidores do mesmo  $DCell_0$ . Num nível acima,  $DCell_1$  é o conjunto construído com  $n+1$   $DCell_0$  (no caso do exemplo,  $DCell_1 = 5 DCell_0$ ). Cada servidor é identificado por uma tupla  $[a_1, a_2]$  onde  $a_1$  representa o nível 1 e  $a_0$  o nível 0. Se houverem mais níveis, mais identificadores são adicionados à tupla. Dois servidores com tuplas  $[i, j-1]$  e  $[j, i]$  são conectados através de um link, além de serem conectados ao mini-switch dos seus respectivos DCell. Se tomarmos cada DCell com um único nó virtual, cada nó está diretamente conectado com todos os outros, porém, na verdade a conexão é feita através dos links entre os servidores.

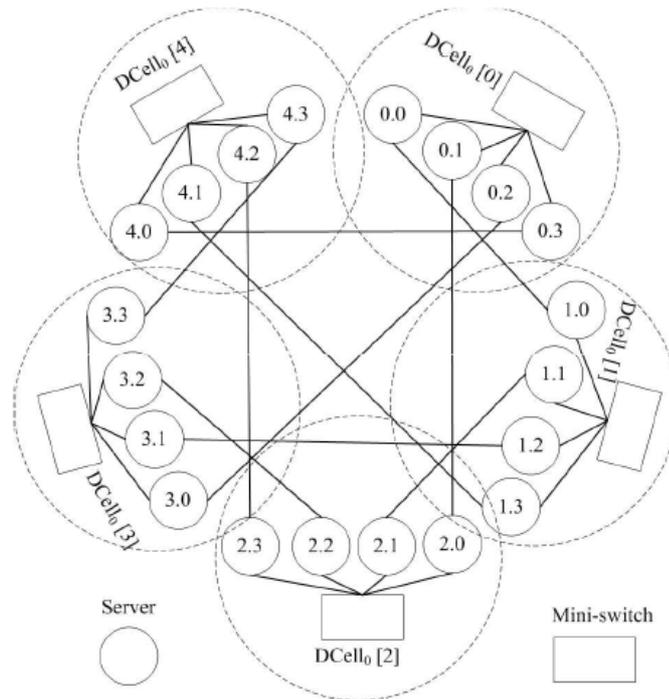


Figura 3.5 – Estrutura de rede DCell

O número de servidores ( $t_k$ ) suportados por um DCell dobra exponencialmente a medida que o grau do nó aumenta, conforme fórmula abaixo. Segundo ela, quando  $k=3$  e  $n=6$ , o número total de servidores pode chegar à marca de 3.26 milhões de máquinas.

$$\left(n + \frac{1}{2}\right)^{2^k} - \frac{1}{2} < t_k < (n + 1)^{2^k} - 1$$

O roteamento em uma rede DCell não pode utilizar o esquema de estado global dos links, já que existem milhões de interconexões entre os servidores, tampouco o protocolo de roteamento hierárquico OSPF, pois necessita de uma área de backbone para interconectar as outras áreas, resultando em congestionamento e criação de pontos únicos de falha. Por isso foi desenvolvido o protocolo DFR (*DCell Fault-tolerant Routing protocol*), uma solução robusta de roteamento descentralizado que fornece tolerância a falhas comuns em data centers (de hardware, software e/ou energia). O roteamento DFR utiliza um algoritmo de roteamento simples unicast (*DCellRouting*) para explorar a estrutura recursiva da rede. O DFR mantém o estado dos links

vizinhos da subrede dentro do  $DCell_k$  e sua abordagem pode ser resumida em ‘dividir e conquistar’, ou seja, como todos os DCells de um mesmo nível estão conectados, primeiro o algoritmo identifica o link que conecta os dois DCells para então identificar o link entre a origem e o link inter-DCells, e entre este e o destino.

A performance do *DCellRouting* é comparável ao roteamento *shortest-path*. A tabela 1 [Guo et al. 2008] compara as médias e desvios padrões entre os dois com diferentes  $n$  e  $k$ . Observamos que a diferença é pequena, porém, neste caso o uso do *DCellRouting* é vantajoso devido a sua simplicidade.

$n$	$k$	$t_k$	Shortest-path		DCellRouting	
			mean	stdev	mean	stdev
4	2	420	4.87	1.27	5.16	1.42
5	2	930	5.22	1.23	5.50	1.33
6	2	1,806	5.48	1.18	5.73	1.25
4	3	176,820	9.96	1.64	11.29	2.05
5	3	865,830	10.74	1.59	11.98	1.91
6	3	3,263,442	11.31	1.55	12.46	1.79

Tabela 3.1 – Comparação entre Shortes-path e DCellRouting.

Quando necessita realizar um broadcast, um servidor em um  $DCell_k$  envia o pacote para todos os seus  $k+1$  vizinhos. O receptor checa se já recebeu o pacote anteriormente, caso positivo descarta o pacote duplicado. Se o pacote for novo, encaminha via broadcast aos seus outros  $k$  links. O diâmetro de uma  $DCell_k$  é de no máximo  $2^{k+1} - 1$  que representa o mesmo número de saltos para a mensagem atingir todos os nós do  $DCell_k$ . O limite de alcance do broadcast é atingido codificando um valor de alcance  $k$  na mensagem de broadcast.

0	4	8	16	31
Version	DHL	Protocol	Payload Length	
Identification				
Retry	Flags	TTL	Header Checksum	
Source DCN Address				
Destination DCN Address				
Proxy DCN Address				

Figura 3.6 - Cabeçalho do protocolo DCN.

O DFR faz parte de um conjunto de protocolos implementados entre a camada 3 e 2, que de um lado oferece uma camada Ethernet virtual à camada IP, e do outro, gerenciam as interfaces físicas Ethernet. Isto permite às aplicações funcionarem normalmente com TCP/IP, utilizando o IP apenas para endereçamento dos servidores de aplicação, sem participar do roteamento. Há um mapeamento direto entre IP e o endereço DCN (figura 3.6 [Guo et al. 2008]). Este conjunto de protocolos é implementado nos servidores, que efetuam todo o roteamento e encaminhamento de pacotes. Um módulo de encaminhamento rápido recebe os pacotes de todas as portas físicas e decide se aceita os pacotes localmente ou encaminha-os para outros servidores. Este módulo mantém uma tabela de encaminhamento que é consultada assim que o pacote é recebido. Se não existir o registro correspondente para o endereço do pacote, o DFR será acionado para calculá-lo e adicioná-lo na cache da tabela. Quando algum estado de link muda, toda a tabela é invalidada e recalculada pelo DFR.

### 3.3.2 VL2

VL2 é uma solução de rede proposta pela equipe da Microsoft que visa atender grandes data centers através de uma rede escalável, confiável, ágil e sem o uso da prática de *oversubscription* [Greenberg et al. 2009]. Consegue isto aproveitando o limite das tecnologias de rede de baixo custo disponíveis no mercado e utilizando uma estrutura de nomes simples, ou seja, sem significado topológico, ao contrário dos nomes IPs hierárquicos. Com isto, as aplicações legadas podem continuar trabalhando da mesma forma e o VL2 consegue entregar aos serviços a abstração de que todos os servidores estão interconectados através de um grande switch Ethernet livre de interferência – uma camada 2 virtual. Não ocorre fragmentação dos recursos, mesmo que o tamanho de cada serviço varie de 1 servidor a 100.000. Assim, VL2 alcança a agilidade – qualquer serviço pode ser atribuído a qualquer servidor – além de manter a alta largura de banda entre servidores e o isolamento de desempenho entre serviços.

Os três principais objetivos da arquitetura VL2 são:

- Uniformizar e manter alta a capacidade entre servidores: a taxa máxima de comunicação servidor-servidor deve ser limitada somente pela capacidade das placas de rede dos servidores e a alocação de um servidor a um serviço deve ser independente da topologia de rede;

- Performance isolada: o tráfego de um serviço não deve influenciar no tráfego de outro serviço, como se cada serviço fosse conectado por um switch físico;

- Semântica de camada 2: o gerenciador do data center deve ser capaz de atribuir facilmente qualquer servidor para qualquer serviço e configurar este servidor com qualquer endereço IP esperado pelo serviço. MVs devem ser capazes de migrar para qualquer servidor, mantendo o mesmo endereço IP, e a configuração de rede dos servidores deve se comportar como se fosse uma LAN.

VL2 organiza seus switches comoditizados através da topologia Clos, conforme a figura 3.7 [Greenberg et al. 2009], com alta conexão entre switches intermediários e de agregação atendidos pelo balanceador de carga VLB (*Valiant Load Balancing*). Todos os switches de agregação se conectam com os intermediários e os ToRs se conectam a apenas dois de agregação, porém o elevado número de links do switch de agregação oferece milhares de caminhos diferentes, de modo que se há  $n$  switches intermediários, a falha em um destes reduz a largura de banda total em apenas  $1/n$ .

VL2 utiliza (1) instanciação de serviços através de nomes simples (não hierárquico como o IP) para migrar MVs a qualquer lugar da rede; (2) VLB (*Valiant Load Balancing*) para distribuir o tráfego uniformemente nos caminhos da rede; e (3) resolução de endereços baseada em sistemas finais para escalar um grande conjunto de servidores, sem introduzir complexidade ao gerenciamento da rede.

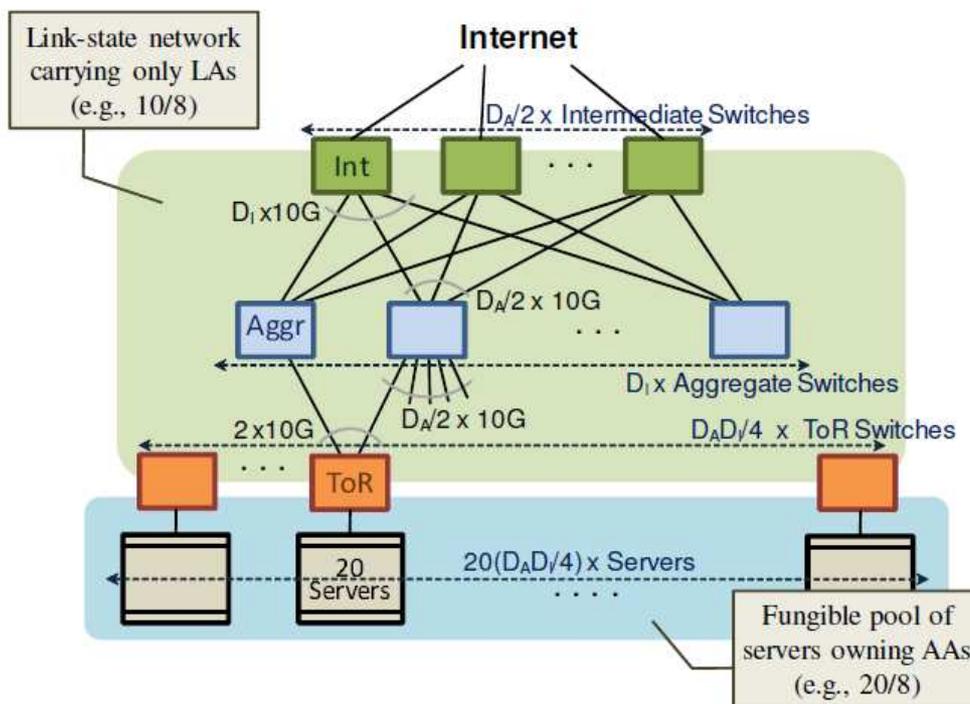


Figura 3.7 – Exemplo da arquitetura VL2 com topologia de rede Clos

O esquema de endereçamento VL2 separa os nomes utilizados pelas aplicações, denominados endereços IP de aplicação específica (AA - *Application-specific Adresses*), dos seus locais físicos, denominados endereços IP de local específico (LA - *Location-specific Adresses*). Toda a infraestrutura de rede utiliza os LAs e um protocolo de estado de conexão baseado em IP (camada 3) que dissemina estes LAs pela rede. Isto permite aos switches alcançar toda a topologia, e encaminhar os pacotes encapsulados com LAs pelo caminho mais curto. Do outro lado, as aplicações utilizam os AAs que permanecem inalterados enquanto a localização dos servidores é alterada pela migração das máquinas virtuais. Cada AA está associado a um LA, que identifica o ToR ao qual o servidor está conectado. São os switches ToRs que fazem o mapeamento LA/AA, conforme a figura 3.8 [Greenberg et al. 2009].

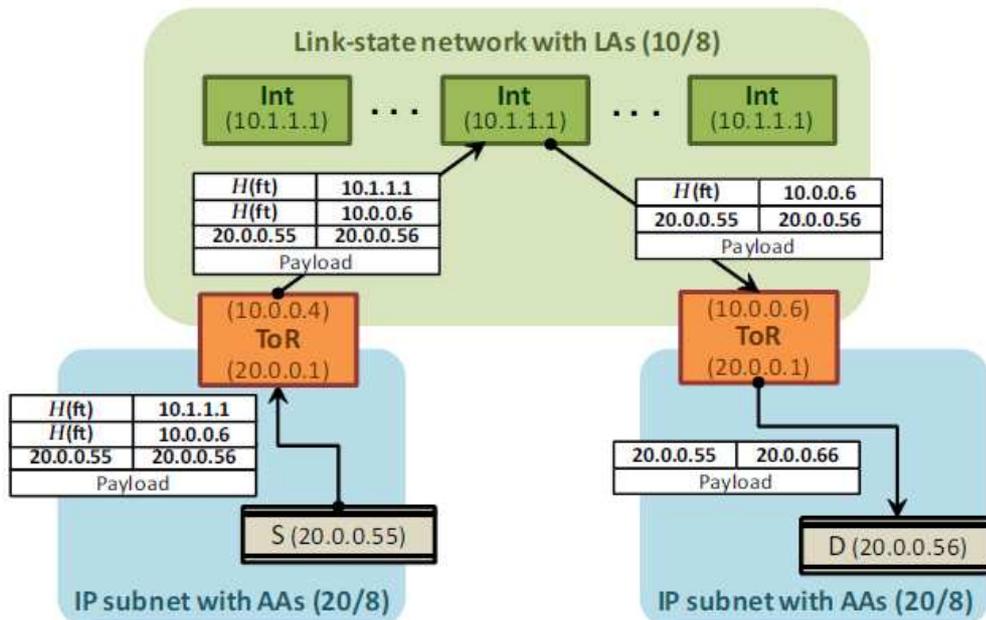


Figura 3.8 – Roteamento VL2.

O ponto principal da oferta de uma camada 2 virtual é ter servidores que, através da implementação de uma pilha de protocolos, acreditam que compartilham uma única sub-rede IP de grande porte (isto é, o espaço inteiro AA) com outros servidores no mesmo serviço. Assim, quando uma aplicação envia um pacote AA pela primeira vez, a pilha de rede no host de origem gera um *broadcast* ARP para o destino AA. O agente VL2 rodando neste host de origem intercepta esta solicitação ARP e converte em uma solicitação *unicast* ao sistema de diretório e este responde com o LA do ToR correspondente. Finalmente o pacote é tunelado até o destino. O agente VL2 mapeia estes endereços, similarmente ao cache ARP, sendo que a próxima conexão não precisará de consulta ao sistema de diretórios. Esta estrutura impõe limite à fonte de *broadcast* evitando transbordo na rede. Faz isso transferindo o ARP para o serviço de diretório e interceptando mensagens DHCP nos ToRs.

Para manter controle das informações do servidor, o sistema de diretório provê 3 funções: (1) consulta de endereços, (2) atualizações do mapeamento de AA para LA, e (3) um mecanismo cache de atualização reativa para que as atualizações sensíveis à latência (por exemplo, a atualização do mapeamento de AA a LA para uma máquina virtual submetidos a migração ao vivo) aconteçam rapidamente.

O sistema de diretório VL2 busca prover escalabilidade, confiança e alta performance, através de baixa latência, alta taxa de transferência e alta disponibilidade. Utiliza duas camadas de servidores: num universo de 100 mil servidores, de 50 a 100 são destinados à pesquisa de nomes (exige baixo tempo de resposta); e 5 a 10 para atualizações de mapeamento (exige confiabilidade) através do algoritmo Paxos.

O serviço de diretório também pode utilizar políticas de acesso através do controle da requisição de tradução de endereços AA, desde que ele conheça o servidor que faz a requisição. Com isto consegue isolamento entre servidores pertencentes ao mesmo serviço. Os pacotes fluem diretamente da origem ao destino, sem interferência e atraso de um gateway IP necessário entre duas VLANs na arquitetura convencional.

O roteamento entre servidores neste ambiente é feito de maneira simples e resiliente: o *Valiant Load Balancing* (VLB) espalha o tráfego uniformemente nos caminhos da rede – cada servidor, independentemente, escolhe uma conexão ao acaso através da rede para cada um dos fluxos enviado para os outros servidores no data center. O uso do VLB é limitado ao modelo funil (*hose model*), onde os fluxos entre o nó de ingresso e egresso devem ser convergidos em uma mesma alocação no fluxo de dados, até que seja necessária a sua divergência. A multiplexação dos fluxos na rede possibilita uma otimização do fluxo dos dados, diminuindo a banda de passagem reservada na rede.

VL2 utiliza dois mecanismos que se completam para distribuir o tráfego e evitar áreas de concentração: VLB e ECMP (Equal Cost Multiple Path). O VLB distribui o tráfego entre um conjunto de switches intermediários e o ECMP distribui o tráfego entre conexões de mesmo custo. Quando há o encapsulamento de pacotes para um determinado switch intermediário e ocorre uma falha neste switch, é necessário atualizar um número potencialmente grande de agentes VL2 nos servidores. Em função disto VL2 designa o mesmo endereço LA (ex. 10.1.1.1) para todos os switches intermediários, e o sistema de diretório retorna este endereço *anycast* aos agentes que realizam consultas. Todos os switches intermediários possuem exatamente três saltos de distância

do servidor de origem, cenário ideal para o ECMP, que cuida de entregar os pacotes encapsulados com o endereço anycast para qualquer um dos switches intermediários ativos. No caso de falhas no switch, o ECMP reage, enviando os pacotes para um switch operacional, eliminando a necessidade de notificar todos os agentes VL2 e assim, garante a escalabilidade.

### 3.3.3 PortLand

PortLand é uma proposta de rede que se propõe a ser eficiente, escalável, facilmente gerenciável e tolerante a falhas. Seu funcionamento é baseado no conhecimento sobre o estado da topologia da rede e em protocolos de roteamento compatíveis com Ethernet. PortLand alcança seus objetivos através da construção de uma camada 2 mais robusta sobre uma topologia de rede *Fat Tree*.

*Fat Tree* deriva da tradicional topologia em árvore e é composta de switches comoditizados para suportar toda a comunicação entre os clusters. Seu diferencial está na oferta de múltiplos caminhos entre quaisquer dois nodos. Essa redundância fornece subsídios para a tolerância a falhas. Sua dimensão é calculada baseada no número de portas ( $k$ ) de cada switch da rede. Uma *Fat Tree* pode suportar, sem bloquear, a comunicação entre  $k^3/4$  servidores finais utilizando  $5k^2/4$  switches de  $k$  portas. A árvore como um todo é dividida em  $k$  *pods* individuais, com cada *pod* suportando comunicação não bloqueante entre  $k^2/4$  servidores. A figura 3.9 [Mysore et al. 2009] esquematiza uma *Fat Tree* com  $k=4$ . Também podemos dividir a árvore em três camadas, chamadas de borda, agregação e núcleo.

Um gerenciador de infraestrutura de rede centralizado mantém informações sobre o estado da rede, incluindo sua topologia. Pode ser implementado na mesma rede do data center ou em separado. A quantidade de informações é restrita ao estado, simplificando o protocolo e promovendo robustez à rede. Além disto, o gerenciador realiza resolução de requisições ARP, operações multicast e participa do mecanismo de tolerância a falhas.

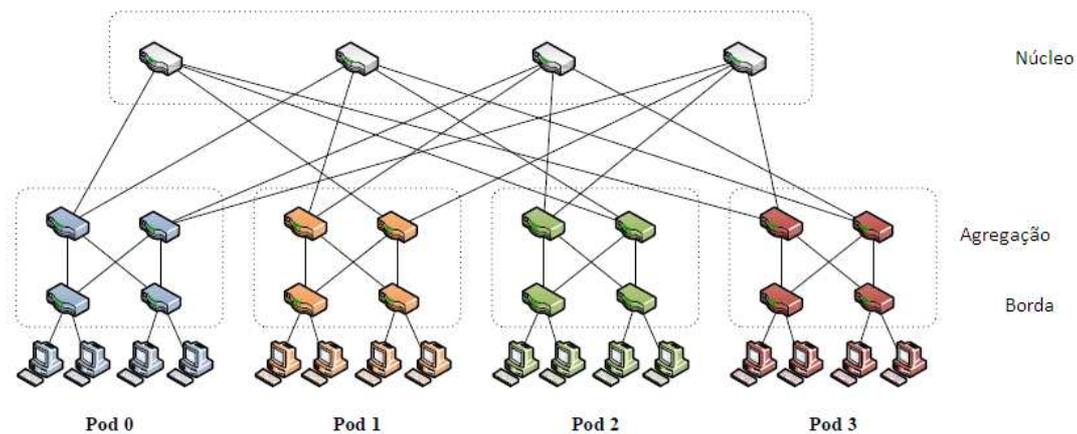
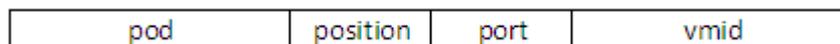


Figura 3.9 - Topologia *Fat Tree* utilizada por PortLand

Para um eficiente encaminhamento, roteamento e migração de MVs, PortLand utiliza um design de endereços chamado Pseudo MAC. Cada servidor físico possui um pseudo-MAC (PMAC) único com o mesmo prefixo entre *pods*. Na verdade, cada servidor físico permanece inalterado, acreditando que continua com seu endereço MAC original (AMAC – *actual MAC*). O ARP utiliza o PMAC para mapeamento da rede. Todos os pacotes utilizam o PMAC, resultando em pequenas tabelas de encaminhamento. O responsável pela tradução PMAC/AMAC são os switches de borda (figura 3.9), mantendo a ilusão de um MAC original para o servidor final.

O formato de PMAC possui 48 bits com a seguinte estrutura:



O campo *pod* possui 16 bits e identifica o *pod* do servidor dentro do data center. Todos os switches do mesmo *pod* possuem o mesmo número *pod*. Este campo é utilizado pelo switch de núcleo para encaminhar o pacote à porta atribuída ao referido *pod*, conforme sua tabela de localização. *Position* têm 8 bits e é utilizado pelo switch de agregação para localizar a posição do switch de borda dentro do *pod*. O campo *port* também têm 8 bits e identifica a porta do switch de borda que conecta-se ao servidor final. *Vmid* possui 16 bits e é utilizado para multiplexar MVs em uma única máquina física.

Para reduzir os efeitos indesejáveis do broadcast ARP, PortLand utiliza o gerenciador de infraestrutura da seguinte maneira: (1) o switch de borda intercepta uma requisição ARP de tradução de IP para MAC e encaminha para o gerenciador de infraestrutura; (2) O gerenciador consulta a tabela PMAC e retorna o endereço PMAC associado ao referido IP; (3) baseado neste retorno, o switch de borda cria uma resposta para o ARP e devolve ao servidor de origem.

No caso do gerenciador não encontrar o PMAC para um IP de uma requisição ARP, esta requisição seguirá em frente e será encaminhada por broadcast a todos os switches e servidores do data center. O servidor alvo retornará o seu AMAC para o switch de ingresso que encaminhará a nova relação PMAC/AMAC para o gerenciador e o host de origem. A partir de então todo encaminhamento de pacote é feito utilizando o PMAC, e a tradução PMAC/AMAC é realizada apenas no último switch de borda.

Para auxiliar no suporte a migração de máquinas virtuais o gerenciador sempre recebe uma atualização via ARP da MV migrada, com o novo mapeamento dos endereços IP/MAC do servidor hospedeiro. Porém, todas as máquinas que se comunicavam com a MV migrada manterão o PMAC anterior, conforme o seu cache ARP, e serão incapazes de continuar a comunicação até sua cache ARP vencer e ser atualizada. Para resolver esta limitação, o gerenciador encaminha uma mensagem de invalidação para o switch anterior da MV migrada. Esta mensagem define uma entrada na tabela de fluxo para o tratamento correto de pacotes subsequentes destinadas ao switch inválido. O switch transmite um ARP gratuito unicast de volta para qualquer host de origem para definir o novo endereço PMAC na sua cache ARP. O switch invalidado pode, opcionalmente, transmitir o pacote para o destino real, para evitar perda de pacotes.

Portland utiliza o protocolo LDP (*Location Discovery Protocol*) para descobrir a posição global de cada switch na topologia e tornar o encaminhamento de pacotes mais eficiente. Este protocolo não necessita de configuração e não deixa o switch trabalhar até saber sua localização. Periodicamente os switches encaminham uma mensagem LDM (*Location*

*Discovery Message*) por todas as suas portas para definir a sua posição e monitorar o estado das suas conexões físicas. A mensagem LDM contém as seguintes informações: (1) identificador único de switch, (2) número de *pod* do switch (com exceção para switches do núcleo), (3) posição do switch de borda dentro do *pod*, (4) nível do switch na árvore e (5) direção do fluxo na porta do switch. A identificação do nível do switch é simples: o switch de borda recebe mensagens LDMs apenas nas portas conectadas aos switches de agregação, pois os servidores não emitem LDM; o switch de agregação recebe LDMs em todas as portas, pois todas as portas estão conectadas a um switch de borda ou a um switch de núcleo; e o switch de núcleo recebe mensagens em todas as suas portas apenas do switch de agregação. Os switches só podem apresentar três estados: (1) desconectado, (2) conectado a um servidor ou (3) conectado a outro switch.

Quando um pacote começa a percorrer um caminho, não é possível para ele viajar de volta na topologia. Isto pode causar perda de conectividade, mas PortLand prefere correr este risco e garantir que loops transitórios e tempestades de broadcast não ocorram.

### **3.3.4 Comparativo**

Diferentes soluções são utilizadas pelas três arquiteturas estudadas para resolver os requisitos levantados na seção 3.2.3. Algumas arquiteturas privilegiam certos requisitos em detrimento de outros. Porém, verificou-se que basicamente a definição de uma topologia robusta e a inserção de uma camada de endereçamento entre a camada 3 (IP) e 2 (Ethernet), além, claro, de outros mecanismos, é que permitem um data center comoditizado trabalhar com agilidade, desempenho, segurança e gerenciabilidade.

Dentro de cada arquitetura estudada, as soluções encontradas para atender determinado requisito acabam influenciando a solução de outro requisito. Por exemplo, a separação de endereços lógicos e físicos ajuda na migração de MVs (requisito agilidade) e diminui o tamanho das tabelas de

roteamento (requisito escalabilidade). Portanto, as soluções devem ser pensadas globalmente a fim de melhorar o sistema como um todo.

Na tabela 3.2, apresentamos os mecanismos de balanceamento de carga e as modificações nos nós de cada proposta estudada. Em seguida, na tabela 3.3 temos um comparativo das soluções propostas relacionadas a cada requisito identificado.

	Balanceamento de carga	Modificação	
		nos nós finais	nos switches
<b>DCell</b>	Não especificado	sim	não
<b>VL2</b>	VLB + ECMP	sim	não
<b>PortLand</b>	Não especificado	não	sim

Tabela 3.2 – Propostas estudadas e suas características.

	Requisito 1	Requisito 2	Requisito 3	Requisito 4	Requisito 5	Requisito 6
	Agilidade	Desempenho	Escalabilidade	Controle e roteamento	Confiabilidade	Custo
<b>DCell</b>	Não	Depende	Estrutura recursiva; a escalabilidade dobra exponencialmente a medida que o grau aumenta;	Controle descentralizado com o uso de protocolo (DRF) hierárquico	Alta tolerância a falhas; protocolo específico e topologia recursiva	Commodity + cabeamento + placas de rede extras
<b>VL2</b>	Sim	Alto	Controle de broadcasts de rede; tabelas de roteamento pequenas (tipicamente 16K de entradas)	Sistema de Diretório centralizado com tunelamento IP-in-IP	Ampliação do nível superior da rede; topologia Clos	Commodity
<b>PortLand</b>	Sim	Alto	Conhecimento da topologia, controle de broadcasts e tabelas pequenas de Pseudo MAC	Gerenciador de Infraestrutura centralizado com Pseudo MAC baseado na posição hierárquica	Múltiplos caminhos; topologia Fat-Tree	Commodity

Tabela 3.3 – Propostas estudadas e requisitos de redes para computação em nuvem.

1. Agilidade: a capacidade do data center em realocar dinamicamente os servidores, geralmente é alcançada a partir da separação dos endereços lógicos dos físicos - VL2 utiliza os não hierárquicos AAs/LAs e PortLand utiliza o Pseudo MAC hierárquico. DCell não possui mecanismos específicos para promover a agilidade.
2. Desempenho: Apesar de DCell ter sido construído para suportar a taxa de transmissão das placas de rede dos servidores, acaba criando gargalos nas camadas inferiores (os DCells de mais nível baixo) que prejudicam o desempenho de todo o conjunto. VL2 e PortLand utilizam uma taxa de oversubscription de 1:1, ou seja, no limite da placa de rede. VL2 ainda utiliza o VLB e o ECMP para distribuir o tráfego e evitar áreas de concentração de fluxo, porém em alguns casos [Al-Fares et al 2010] pode haver perda de larguras de banda devido à colisões de longo prazo.
3. Escalabilidade: o desafio de gerenciar milhões de IPs e MACs ao mesmo tempo deve ser resolvido através de tabelas de encaminhamento com tamanho gerenciável e controle de *broadcasts*. DCell utiliza uma estrutura recursiva que permite que sua escalabilidade dobre exponencialmente a medida que o grau do DCell aumente; VL2 e PortLand controlam as mensagens de broadcasts e possuem tabelas de roteamento pequenas (tipicamente 16K de entrada).
4. Controle: deve ser flexível, capaz de introduzir com facilidade novos nós na rede. Em DCell é descentralizado e utiliza o protocolo DRF para roteamento. Em VL2 e PortLand é centralizado e em todas as três arquiteturas, o controle atua através de uma pilha de protocolos que trabalha entre a camada 3 (IP) e 2 (Ethernet).
5. Confiabilidade: a topologia do data center influencia muito neste aspecto pois possuir ou não múltiplos caminhos depende de como a rede é estruturada.

6. Custo: todas as três propostas utilizam equipamentos comoditizados, porém DCell necessita de cabeamento e placas de rede extras.

## 4. Conclusão

Dado o tamanho, custo e importância dos data centers de serviços em nuvem emergentes, é imprescindível repensar os componentes e arquiteturas destes ambientes. Computação em nuvem demanda uma nova geração de data centers especializados em virtualização e mobilidade de MVs. A TI deverá ser hábil a prover a migração de MVs de missão crítica dentro do datacenter e ao redor do mundo. As solicitações de serviços críticos somente migrarão para uma arquitetura virtual se as portas virtuais possuírem a mesma consistência, segurança, gerência e políticas oferecidas pelas portas físicas.

A identificação dos requisitos para arquiteturas de data centers que suportem computação em nuvem se concentram primeiramente em requisitos de redes. As limitações impostas pelo modelo tradicional IP/Ethernet devem ser solucionadas para prover ao sistema os requisitos identificados neste trabalho: agilidade, desempenho, escalabilidade, controle, confiabilidade e baixo custo.

As arquiteturas estudadas utilizam diferentes mecanismos para atender os requisitos acima, porém a utilização de topologias robustas de redes e a introdução de uma camada de endereçamento entre a camada 3 (IP) e 2 (Ethernet) é comum a todas elas, configurando-se como as principais soluções encontradas.

Porém, além de requisitos de redes, existem muitos outros desafios. A definição de padrões abertos que permitam a interoperabilidade entre as diferentes nuvens é essencial. A redução do tempo de acesso aos discos através da adição de uma memória cache entre a RAM e os discos melhora o desempenho das aplicações e vários projetos (FAWN, Gordon e RAMCloud) estão sendo desenvolvidos nesta área. Sistemas de armazenamento NoSQL permitem a escalabilidade horizontal necessária para a computação em nuvem. *Green Cloud* (nuvem verde) traz um forte apelo ecológico, permitindo que toda a energia elétrica dispendida na manutenção do data center varie em função da carga de trabalho dos serviços locados no data center. Máquinas físicas, equipamentos de rede e infraestrutura de refrigeração terão seu funcionamento acionados gradualmente, de acordo com a demanda de máquinas virtuais por

parte das aplicações. Segundo estudos [Liu et al 2009], a economia pode chegar em torno de 30% do consumo de um data center tradicional.

O modelo de computação em nuvem atrai o interesse de diversos setores como indústrias de hardware, software, comunicação e serviços. Ainda estamos no início da evolução, tentando adaptar as nossas tecnologias a este novo paradigma, e com certeza novas soluções e modelos irão surgir para suportar uma computação de escala mundial que atenda nossas necessidades e transforme o mundo como o conhecemos.

#### **4.1 Trabalhos Futuros**

Propõe-se como trabalhos futuros o estudo de redução de acesso aos discos, proporcionando maior desempenho ao data center; e a implementação em um data center ou cluster de alguma das arquiteturas estudadas, a fim de comprovar sua eficiência. Tem-se, então, em mais um desafio: encontrar uma infraestrutura escalável e disponível, com capacidade de centenas ou milhares de servidores, para efetuar testes consistentes.

## 5. Bibliografia

[Al-Fares et al. 2008] *A Scalable Commodity Data Center Network Architecture* / Al-Fares M., Loukissas A. e Vahdat A.. — Proceedings of the ACM SIGCOMM 2008 conference on Data communication, Seattle, EUA, 2008.

[Al-Fares et al 2010] *Hedera: dynamic flow scheduling for data center networks* / Al-Fares M., Radhakrishnan S., Raghavan B., Huang N., e Vahdat A. — Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, San Jose, CA, EUA, 2010.

[Armbrust et al. 2009] *Above the Clouds: A Berkeley View of Cloud Computing* / Armbrust M., Fox A., Griffith R., Joseph A. D., Katz R. H., Konwinski A., Lee G., Patterson D. A., Rabkin A., Stoica I. e Zaharia M. — Technical Report UCB/EECS, EECS Department, University of California, Berkeley, EUA, 2009.

[Benson et al. 2009] *Understanding Data Center Traffic Characteristics* / Benson T., Anand A., Akella A. and Zhang M. — ACM SIGCOMM Computer Communication Review, SESSION: Best workshop papers from ACM SIGCOMM 2009 Pages: 92-99, Nova York, EUA, 2010

[Buyya et al. 2009] *Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility* / Buyya R., Yeo C. S., Venugopal S., Broberg J., e Brandic I. — Grid Computing and Distributed Systems (GRIDS) Laboratory, Department of Computer Science and Software Engineering - The University of Melbourne, Australia 2009.

[Cisco, IDC 2009] *The Next Phase of Datacenter Development* – Adaptado de *Datacenter Network Customer Requirements* de Lucinda Borovick. Disponível em: [http://www.cisco.com/en/US/prod/collateral/switches/ps9441/idc\\_paper.pdf](http://www.cisco.com/en/US/prod/collateral/switches/ps9441/idc_paper.pdf)

[Cloud Standards 2010] Página principal Cloud Standarts Wiki / Acessado em 08/08/10: [http://cloud-standards.org/wiki/index.php?title=Main\\_Page](http://cloud-standards.org/wiki/index.php?title=Main_Page)

[Coulouris 1994] *Distributed Systems: Concepts and Desig* / Coulouris G., Dollimore J., Kindberg T. – Addison-Wesley, 1994.

[Dantas 2005] *Computação Distribuída de Alto Desempenho: Redes, Clusters e Grids Computacionais* / Dantas M. – AXCEL, Brasil, 2005

[EMC 2010] *Teletipo digital do crescimento das informações mundiais* – acessado em 06/07/2010: <http://brazil.emc.com/leadership/digital-universe/expanding-digital-universe.htm>

[Foster et al. 2002] *What is the Grid? A Three Point Checklist* / Ian Foster – Argonne National Laboratory & University of Chicago, 2002 – artigo disponível em [http://www.ogf.org/UnderstandingGrids/resource\\_articles.php](http://www.ogf.org/UnderstandingGrids/resource_articles.php)

[Foster et al. 2008] *Cloud Computing and Grid Computing 360-Degree Compared* / Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu – Grid Computing Environments Workshop 2008, Texas, EUA, 2008.

[Greenberg et al. 2009] *VL2: A Scalable and Flexible Data Center Network* / Greenberg A., Hamilton J. R., Jain N., Kandula S., Kim C., Lahiri P., Maltz D. A., Patel P., Sengupta S. – Microsoft Research, ACM SIGCOMM 2009 Conference. Barcelona, Espanha, 2009.

[Greenberg et al. 2008] *Towards a Next Generation Data Center Architecture: Scalability and Commoditization* / Greenberg A., Lahiri P., Maltz D. A., Patel P., e Sengupta S. – Proceedings of the ACM Workshop on Programmable Routers For Extensible Services of Tomorrow, Seattle, USA, 2008.

[Guo et al. 2008] *Dcell: A Scalable and Fault-tolerant Network Structure for Data Centers* / Guo C., Wu H., Tan K., Shi L., Zhang Y., e Lu S. – SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008, New York, EUA 2008.

[Guo et al. 2009] *BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers* / Guo C., Lu G., Li D., Wu H., Zhang X., Shi Y., Tian C., Zhang Y., e Lu S. – Proceedings of the ACM SIGCOMM 2009, Barcelona, Spain, 2009

[Hornby 2002] *Consolidation in the Data Center: Simplifying IT Environments to Reduce Total Cost of Ownership* / Hornby D. & Pepple K. – Prentice Hall-PTR, 2002.

[HPC in the Cloud, 2010] *Cloud Computing: Improving Enterprise IT Efficiency and Agility. HPC in the Cloud* – Sponsored Content, 2010. Disponível em: <http://www.hpcinthecloud.com/features/Cloud-Computing-Improving-Enterprise-IT-Efficiency-and-Agility-102021873.html?viewAll=y>

[IBD 2010] *Intelligent Building Dictionary* - Fig. 16: TIA-942 Data Center Cabling. Disponível em: <http://www.intelligent-building-dictionary.com/figure.php/id/27/>

[Kandula et al. 2009] *The Nature of Data Center Traffic: Measurements and Analysis*. Kandula S., Sengupta S., e Patel P. – Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference Chicago, Illinois, EUA 2009.

[Kim et al. 2008] *Floodless in seattle: a scalable ethernet architecture for large enterprises* / Kim C., Caesar M. e Rexford J. – SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008, Seattle, USA, 2008.

[Kliegl et al. 2010] *Generalized DCell Structure for Load-Balanced Data Center Networks* / Kliegl M., Lee J., Zhang X., Guo C. e Rincon D. – Grupo de Pesquisa Asiático da Microsoft, Proceedings IEEE Infocom 2010, San Diego, CA, EUA, 2010.

[Kurose, Ross 2006] *Redes de computadores e a Internet: uma abordagem top-down* / Kurose J. F., Ross K. W. – Pearson Addison Wesley 3. ed., São Paulo, Brasil, 2006.

[Liu et al 2009] *GreenCloud: a new architecture for green data center* / Liu L., Wang H., Liu X., Jin X., He W. B., Wang Q. B., e Chen Y. – Proceedings of the 6th international Conference industry Session on Autonomic Computing and Communications industry Session, Barcelona, Spain, 2009.

[**Mysore et al. 2009**] *PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric*. Mysore R. N., Pamboris A., Farrington N., Huang N., Miri P., Radhakrishnan S., Subramanya V., e Vahdat A. – Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication, Barcelona, Espanha, 2009.

[**Mell, Grace 2009**] NIST - *The NIST Definition of Cloud Computing* / Mell P. e Grance T. – The National Institute of Standards and Technology, Information Technology Laboratory 2009. Disponível em: [csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc](http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc)

[**Open Cloud Manifesto 2010**] Open Cloud Manifesto (2010) — Acessado em 10/05/2010 : <http://opencloudmanifesto.org/Open%20Cloud%20Manifesto.pdf>

[**Tanenbaum 1995**] *Distributed Operating Systems* / Tanenbaum A. S. — Prentice-Hall, 1995.

[**Taurion 2009**] *Cloud Computing: computação em nuvem: transformando o mundo da tecnologia da informação* / Taurion C.— Brasport, Rio de Janeiro, Brasil, 2009.

[**TIA 2010**] *TIA-942 - Telecommunications Infrastructure Standard for Data Centers*. Disponível em: <http://www.tiaonline.org/>

[**Velte et al. 2010**] *Cloud Computing: A Practical Approach* / Velte A. T., Velte T. J., Elsenpeter R. – The McGraw-Hill Companies, EUA, 2010.

[**Veras 2009**] *Datacenter: componente central da infraestrutura de TI* / Veras M. — Brasport, Rio de Janeiro, Brasil, 2009.

[**Verdi et al. 2010**] *Novas Arquiteturas de Data Center para Cloud Computing* / Verdi F. L., Rothenberg C. E., Pasquini R. e Magalhães M. F. — SBRC 2010 - Minicursos, Brasil, 2010.

[**Wazlawick 2008**] *Metodologia De Pesquisa Em Ciência Da Computação* / Wazlawick R. S. – Elsevier, Rio de Janeiro, Brasil, 2008.

[**Zhang et al. 2009**] *CCOA: Cloud Computing Open Architectur* / Zhang L. J. e Zhou Q. – IEEE International Conference on Web Services, IBM T.J. Watson Research Center, Nova York, EUA 2009.

[**Zhang et al. 2010**] *Cloud computing: state-of-the-art and research challenges* / Zhang Q., Lu Cheng L., Boutaba R. – Journal of Internet Services and Applications, Springer, London, 2010.