Uma Avaliação da Robustez Intra Data Centers Baseada na Topologia da Rede

Rodrigo S. Couto, Miguel Elias M. Campista e Luís Henrique M. K. Costa *

[1] Grupo de Teleinformática e Automação
PEE/COPPE - DEL/POLI
Universidade Federal do Rio de Janeiro

{souza, miguel, luish}@gta.ufrj.br

Resumo. *A crescente utilização de aplicações em data centers, que requerem comunicação intensa entre os servidores devido ao uso de sistemas de computação e armazenamento distribuídos, tornam a infraestrutura de rede um dos principais gargalos. Além disso, as redes formadas intra data centers devem ser escaláveis, tolerantes a falhas e serem capazes de se comunicar a altas taxas de transmissão. Para isso, diversas arquiteturas foram propostas com topologia e protocolos próprios. Este trabalho apresenta uma análise comparativa da robustez a falhas das topologias utilizadas pelas principais arquiteturas de data center da literatura corrente. A análise é baseada em falhas nos principais componentes da rede (enlace, servidor e comutador) provocadas para mostrar os compromissos existentes em cada arquitetura. A análise considera a disposição topológica dos nós e sua relação com o nível de tolerância a falhas. Os resultados mostram que é possível modificar parâmetros de algumas topologias de forma que a rede dependa menos de um determinado tipo de componente.*

Abstract. *The growing use of data center applications requiring intensive communications between servers as a consequence of distributed computing systems and storage facilities, makes the network infrastructure a relevant bottleneck. Moreover, intra data center networks must be scalable, fault tolerant, and also be able to communicate at high transmission rates. For this end, several architectures have been proposed employing their own topology and protocols. This paper presents a comparative analysis considering the impact onto the topologies used by the main data center architectures found on the current literature. The analysis is based on failures on the main networking components (link, server and switch) induced to show the existing architectures tradeoffs. The analysis considers the topological node positioning and its relation to the level of fault tolerance. Results show that it is possible to modify parameters of some topologies in order to make less dependent of a given component.*

1. Introdução

Atualmente, o tempo necessário para concluir qualquer transação na Internet vem se tornando um diferencial cada vez mais valorizado entre as empresas que oferecem serviços a usuários finais como sites de busca de conteúdo, de operações bancárias e de

*Este trabalho foi realizado com recursos do CNPq, CAPES, Faperj e FINEP. Os autores gostariam de agradecer ao professor Daniel Ratton Figueiredo pelas contribuições ao trabalho.

compras de passagens aéreas. A solução comum adotada é aumentar o poder da base computacional para que as requisições sejam prontamente atendidas com o menor tempo de resposta possível. Tal poder computacional vem explodindo em números cada vez mais contundentes. Dados estimados revelam que a Google, por exemplo, possui uma base computacional formada por quase um milhão de servidores concentrados em dezenas de *data centers* espalhados ao redor do mundo [Miller, 2011].

A efetividade do poder computacional dos *data centers* está intimamente ligada à sua capacidade de comunicação entre os diferentes servidores que os compõem. Modelos de programação, como o *MapReduce* [Dean e Ghemawat, 2008], são utilizados para processamento distribuído que requerem transporte maciço de dados entre as suas diferentes fases de execução (mapeamento e redução). Aplicações executadas em servidores de *data centers* podem precisar de dados armazenados em sistemas de arquivos remotos antes de dar continuidade a qualquer outra operação. Por fim, atualizações de sistemas de indexação e ainda caches de conteúdo também podem necessitar o uso intenso de redes de comunicação. Todas essas tarefas envolvem uma quantidade crescente de dados e requerem operações intensas de rede.

As arquiteturas para *data centers* possuem tipicamente três objetivos principais: escalabilidade, baixo custo e compatibilidade com tecnologias legadas, e tolerância a falhas [Verdi et al., 2010]. O primeiro se refere ao aumento da rede sem prejuízo das taxas de transmissão e do poder de processamento dos dados. Dependendo da topologia intra *data center*, a taxa de dados entre dois nós topologicamente distantes pode estar limitada à taxa do nó hierarquicamente superior, ou seja do nó raiz, se considerada uma topologia em árvore. O segundo objetivo está ligado ao emprego de equipamentos de prateleira que possuem custos reduzidos já que são produzidos em escala e, provavelmente, compatíveis com tecnologias legadas. A solução trivial para o problema da limitação da taxa de transmissão vai de encontro à redução de custos. A aquisição de equipamentos de rede, que podem transmitir a taxas suficientemente grandes para suprir a deficiência da topologia, não resolve o problema fundamentalmente já que o tamanho dos *data centers* cresce dia após dia. Por outro lado, essa mesma limitação de taxa poderia ser atacada a partir de protocolos de comunicação próprios. Entretanto, o emprego de tecnologias difundidas como o IP e o Ethernet facilita a interoperabilidade com outras redes. Por fim, considerando que o tamanho dos *data centers* é da ordem de centenas de milhares de nós, é de se esperar que a falha de componentes seja frequente. Falhas em discos rígidos, problemas de suprimento elétrico, quedas de comutadores podem ocorrer diariamente e mesmo assim o *data center* deve se manter operacional sem provocar abalos perceptíveis aos usuários.

Dependendo do objetivo principal de cada uma das arquiteturas para *data centers*, seja escalabilidade, custo e compatibilidade, ou tolerância a falhas, um dos três componentes principais do sistema – servidor, comutador ou enlace - pode ser utilizado predominantemente. A arquitetura proposta pela Fat-Tree [Al-Fares et al., 2008] preza o uso de comutadores de prateleira e a garantia da escalabilidade da rede. Para tal, foca a sua arquitetura na presença de comutadores e não requer dos servidores nenhuma atividade de rede. Já a arquitetura DCell [Guo et al., 2008] atribui maior importância aos servidores já que eles participam do encaminhamento de dados. O DCell possui uma estrutura em células conectadas recursivamente para aumentar a sua capacidade de expansão, na qual os servidores possuem um papel fundamental para conectar as células e garantir a conec-

tividade intra célula. O BCube [Guo et al., 2009] é outra arquitetura que se assemelha ao DCell em seu aspecto de uso dos servidores para as tarefas de rede. O BCube foi desenvolvido para ser embutido em contêineres selados para facilitar o seu deslocamento. O BCube também possui arquitetura modular na qual, diferente do DCell, seus módulos são interconectados por comutadores.

Até o momento, poucos trabalhos comparam as diferentes arquiteturas sob o ponto de vista de falhas de cada um dos seus componentes principais. Nota-se que, dependendo da arquitetura, a rede intra *data center* é composta apenas por comutadores ou por comutadores e servidores. Essas diferenças tornam a comparação complexa porque depende da métrica de interesse. Em [Gill et al., 2011] são realizadas medidas em enlaces e equipamentos de redes de *data centers* reais para quantificar falhas nessas redes, não considerando as novas topologias devido à escassez de soluções reais que as utilizem atualmente. Popa *et al.* [Popa et al., 2010] comparam diferentes arquiteturas em termos de custo e consumo energético. Para isso, é definido um nível alvo para a capacidade e latência da rede que atinjam um desempenho satisfatório. Na comparação, os custos das diferentes arquiteturas são calculados para alcançar o alvo estabelecido. Em [Guo et al., 2009] a robustez das topologias são comparadas considerando um padrão de tráfego específico e o funcionamento dos protocolos das arquiteturas que as empregam. Este trabalho, diferentemente de [Guo et al., 2009], compara a robustez da topologia das principais arquiteturas intra *data center* existentes (FatTree, DCell e BCube) do ponto de vista da teoria dos grafos. Para isso, modela-se todos equipamentos da rede como nós e os cabos como enlaces. A partir do modelo, neste trabalho avalia-se o impacto na topologia em consequência de falhas em cada um dos componentes principais de rede (servidor, comutador e enlace). Os resultados obtidos permitem verificar o grau de comprometimento de cada uma das arquiteturas com os seus componentes de rede. As métricas utilizadas na avaliação são o tamanho dos componentes conexos da rede e a relação entre o diâmetro de seu maior componente conexo e seu diâmetro original. A análise da robustez considerando apenas a disposição topológica é importante pois é independente da aplicação e dos algoritmos de roteamento e engenharia de tráfego das arquiteturas. Assim, fornece um estudo de base assumindo que novos algoritmos possam ser incorporados às arquiteturas em questão. O estudo deste trabalho considera então que todos os algoritmos de roteamento e engenharia de tráfego funcionam de forma ótima, reagindo de forma adequada às falhas de componentes e possibilitando o uso eficiente dos múltiplos caminhos.

Este trabalho está organizado da seguinte forma: a Seção 2 detalha os critérios de formação das topologias de rede intra *data center*. A Seção 3 apresenta a metodologia utilizada na análise de robustez bem como descreve as métricas empregadas na avaliação. Já a Seção 4 apresenta os resultados obtidos. Por fim, a Seção 5 conclui este trabalho e indica as direções futuras.

2. Topologias de redes intra *Data Center*

Neste trabalho foram consideradas três das topologias mais referenciadas na literatura, e já utilizadas em trabalhos recentes como [Raiciu et al., 2011], e as variações de seus parâmetros, como número de portas de comutadores e servidores. Considera-se ainda que os enlaces possuem a mesma capacidade em todas as topologias abordadas.

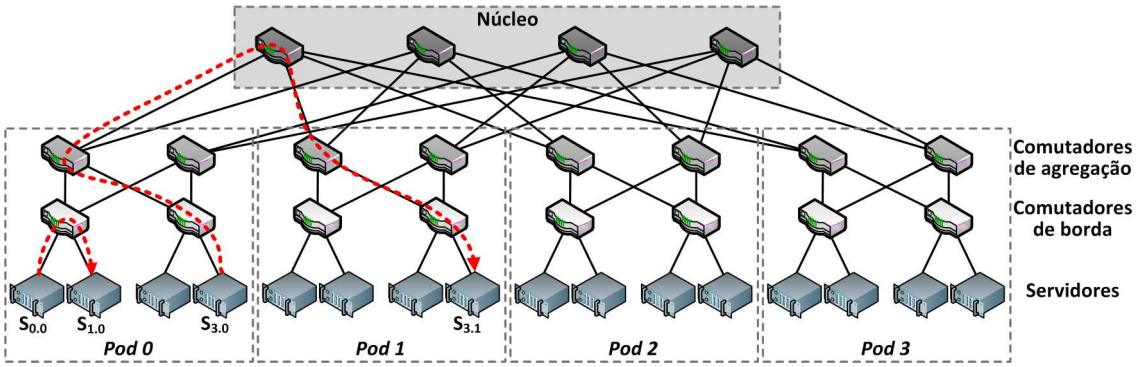


Figura 1. Topologia utilizada pela arquitetura Fat-Tree.

2.1. Arquitetura Fat-Tree

Neste trabalho denomina-se como Fat-Tree a topologia proposta por Al-Fares *et al.* em [Al-Fares et al., 2008]. Al-Fares *et al.* utilizam o conceito de fat-tree, que é um caso especial de uma rede de Clos, para definir uma topologia de *data center* organizada na forma de uma árvore k-nária. Como mostrado na Figura 1, a árvore possui dois tipos de conjuntos, o núcleo e os *Pods*. O núcleo é formado por comutadores que possuem cada uma de suas portas conectada a um *Pod* diferente. O *Pod* é formado por comutadores de agregação e de borda, e também pelos servidores do *data center*. Os comutadores de agregação realizam a conexão entre o *Pod* e o núcleo e possuem conexão com os comutadores de borda e os de núcleo. Já os comutadores de borda possuem ligações com um conjunto diferente de servidores. Todos os comutadores da rede são idênticos e possuem k portas. Assim, a rede possui k *Pods*, sendo que cada *Pod* possui $\frac{k}{2}$ comutadores de agregação e outros $\frac{k}{2}$ de borda. Em um *Pod*, cada comutador de agregação está ligado a todos os comutadores de borda, que estão individualmente ligados a $\frac{k}{2}$ servidores diferentes. Assim, a topologia Fat-Tree possui capacidade para $\frac{k}{2} * \frac{k}{2} * k = \frac{k^3}{4}$ servidores. A Figura 1 mostra uma Fat-Tree para $k = 4$. Note que o Servidor de índice 0 no *Pod 0* ($S_{0.0}$) está se comunicando com o Servidor 1 ($S_{1.0}$) no seu mesmo *Pod*, sendo que ambos estão conectados através do mesmo comutador de borda. Já o Servidor 3 do *Pod 0* ($S_{3.0}$) se comunica com um servidor em outro *Pod*, $S_{3.1}$, e por isso mesmo precisa utilizar os comutadores de núcleo. A Fat-Tree permite que todos os servidores se comuniquem ao mesmo tempo utilizando a capacidade total de suas interfaces de rede. Além disso, nessa topologia, todos os elementos de rede são idênticos, não necessitando de comutadores caros com grande número de portas em níveis mais altos da hierarquia da árvore.

A arquitetura VL2 [Greenberg et al., 2009] também é uma rede de Clos e não foi utilizada neste trabalho por se assemelhar com a Fat-Tree [Popa et al., 2010].

2.2. Arquitetura BCube

A topologia BCube [Guo et al., 2009] foi proposta para utilização em *data centers* modulares (MDC - *Modular Data Centers*), que são geralmente montados em contêineres, possuindo maior facilidade em sua migração para outras localidades geográficas. Essa migração é útil em termos de economia de energia, possibilitando o transporte do *data center* para regiões com menor custo de refrigeração em uma determinada época ou para atender melhor a demanda por um serviço em uma localidade específica. Por serem mon-

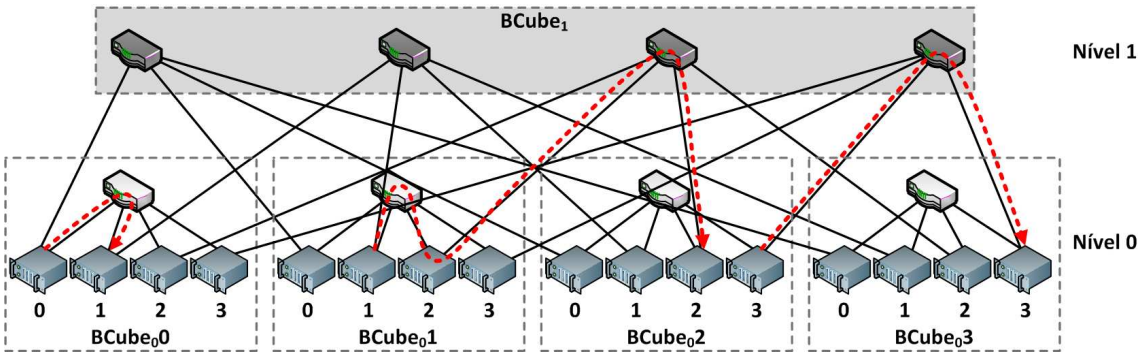


Figura 2. Topologia utilizada pela arquitetura BCube.

tadas em contêineres selados e com alta densidade de equipamentos, essas redes possuem manutenção difícil, necessitando de uma alta tolerância a falhas. Assim, é desejável que o desempenho diminua lentamente com o aumento do número de falhas de seus equipamentos. Além disso, como no caso da Fat-Tree, a rede precisa ter alta capacidade de transferência de bits e baixo custo. Para tal, a topologia BCube emprega mini-comutadores e servidores que participam do encaminhamento. Esses servidores devem, então, possuir mais de uma interface de rede, que tipicamente não é maior que cinco [Guo et al., 2009]. A topologia BCube é definida de forma recursiva, sendo uma rede $BCube_n$ constituída por n redes do tipo $BCube_{n-1}$. O componente fundamental da topologia é uma rede $BCube_0$, composta por um único comutador de n portas ligado a n servidores. Para a construção de uma $BCube_1$ utiliza-se n redes $BCube_0$ e n comutadores. Cada comutador é conectado a todas as redes $BCube_0$ através da conexão com um dos servidores de cada $BCube_0$. A Figura 2 ilustra uma rede $BCube_1$. De forma mais geral, uma rede $BCube_k$ ($k \leq 1$) é formada por n $BCube_{k-1}$s e n^k comutadores de n portas. Para construir uma $BCube_k$ numera-se as n $BCube_{k-1}$s de 0 a $n-1$ e os servidores de cada uma delas são numerados de 0 até n^k-1. Em seguida, conecta-se a porta de nível k do i-ésimo servidor ($i \in [0, n^k-1]$), situado no j-ésimo $BCube_k$ ($j \in [0, n-1]$), à j-ésima porta do i-ésimo comutador de nível k. Uma rede $BCube_k$ possui capacidade de n^{k+1} servidores. Note na Figura 2 que no $BCube_0 0$, o Servidor 0 se comunica através de um comutador com o Servidor 1 na mesma rede. Já o Servidor 1 do $BCube_0 1$ utiliza o comutador da rede para encaminhar os seus dados para o Servidor 2 que é quem possui o enlace até a rede do destino, no caso, o $BCube_0 2$. Entretanto, as comunicações entre $BCube$s diferentes em um mesmo nível podem se comunicar envolvendo também apenas um comutador de nível superior, como é o caso do Servidor 3 do $BCube_0 2$ com o Servidor 3 do $BCube_0 3$. Logo, diferente da Fat-Tree, os servidores podem participar do encaminhamento de dados dependendo das suas posições na topologia completa.

2.3. Arquitetura DCell

Assim como as topologias acima, a DCell [Guo et al., 2008] foi proposta para prover alta capacidade de transferência e tolerância a falhas. Da mesma forma que a BCube, a DCell é definida recursivamente e utiliza encaminhamento pelos servidores e mini-comutadores. O componente fundamental da topologia é a rede $DCell_0$ constituída, assim como a $BCube_0$, por um comutador ligado a n servidores. Constrói-se uma $DCell_1$ utilizando $n+1$ redes $DCell_0$, na qual duas $DCell_0$ estão conectadas entre si através de um enlace formado por um de seus servidores. A $DCell_1$ está ilustrada na Figura 3.

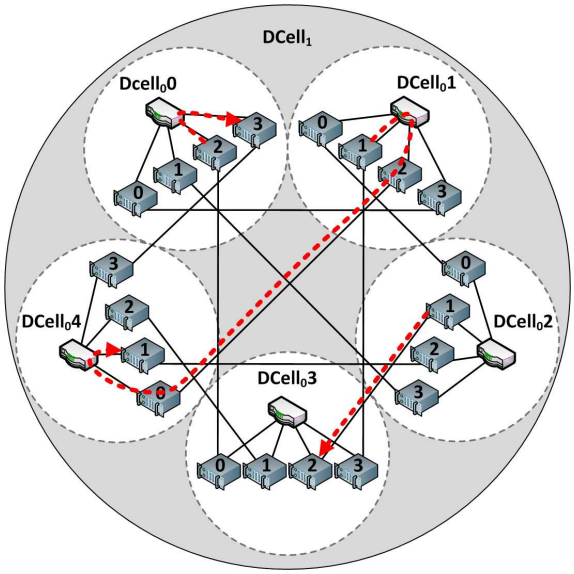


Figura 3. Topologia utilizada pela arquitetura DCell.

Note que as comunicações intra célula são realizadas localmente através do comutador, como visto na comunicação entre o Servidor 2 e 3 da $DCell_0 0$. Já a comunicação entre servidores em células diferentes ou são realizadas de maneira direta, como a comunicação entre o Servidor 1 na $DCell_0 2$ e o Servidor 2 na $DCell_0 3$, ou através de uma combinação de servidores e comutadores, como visto na comunicação entre o Servidor 1 na $DCell_0 1$ e o Servidor 1 na $DCell_0 4$. Note que, neste último exemplo, o caminho total percorrido é o maior já visto entre as topologias apresentadas, englobando dois comutadores e dois servidores. Esse comportamento é confirmado nos resultados deste trabalho.

Observa-se na arquitetura DCell que, diferentemente da BCube, os comutadores apenas estão conectados aos servidores de seu DCell e a ligação direta entre diferentes redes DCell é sempre realizada através dos servidores. Para a construção de uma $DCell_k$ são necessárias $n + 1$ redes $DCell_{k-1}$. Cada servidor em uma rede $DCell_k$ possui $k + 1$ enlaces, sendo que em cada servidor, o primeiro enlace (enlace de nível 0) é conectado ao comutador da $DCell_0$ que ele faz parte, o segundo enlace conecta o servidor a um nó de uma mesma $DCell_1$, mas em uma $DCell_0$ vizinha. Genericamente, o enlace de nível i de um servidor o conecta a uma $DCell_{i-1}$ vizinha dentro de uma mesma $DCell_i$. O procedimento de construção é mais complexo que o da rede BCube, sendo executado a partir de um algoritmo de formação proposto por Guo *et al.* [Guo et al., 2008].

A capacidade da rede em número de servidores pode ser calculada de forma recursiva utilizando as seguintes relações:

$$g_k = t_{k-1} + 1, \tag{1}$$

$$t_k = g_k \times t_{k-1}, \tag{2}$$

onde g_k é o número de redes $DCell_{k-1}$ no $DCell_k$ e t_k é o número de servidores no $DCell_k$. A $DCell_0$ é um caso especial na qual $g_0 = 1$ e $t_0 = n$.

3. Análise de Robustez

Neste trabalho, a análise de robustez das topologias intra *data center* leva em consideração falhas de comutadores, servidores e enlaces. Para tal, a topologia foi modelada como um grafo $G = (V, E)$ não direcionado, no qual V é o conjunto de vértices sendo $V = S \cup C$, onde S é o conjunto formado pelos servidores e C pelos comutadores. O conjunto E é formado pelas arestas entre os pares de vértices em V, modelando os enlaces da rede. No grafo G, todos os enlaces possuem peso unitário, devido à homogeneidade da taxa de bit das interfaces de rede proposta pelas topologias em questão. Para avaliar a robustez do grafo G de cada uma das redes consideradas, foram retirados S', C' ou E' de G, onde $S' \subset S$, $C' \subset C$ e $E' \subset E$, gerando o subgrafo G'. A remoção de componentes da rede foi realizada de forma a avaliar o impacto individual. Assim, os componentes retirados de G foram escolhidos aleatoriamente de apenas um dos conjuntos de componentes originais por vez, gerando o subgrafo G'. No caso de remoção de um componente, ele não fará parte do grafo G' e, se o componente for um vértice, todas as suas arestas também são eliminadas. Os grafos do experimento foram gerados e analisados a partir da ferramenta NetworkX [Hagberg et al., 2008]. Essa análise verificou qual o comportamento da rede quando não há manutenção dos dispositivos até o ponto de existirem uma determinada fração de componentes danificados. Este estudo é importante, por exemplo, para *data center* modulares (MDC) nos quais a manutenção de componentes é de difícil realização.

Este trabalho avalia a robustez do ponto de vista do tamanho da rede e da variação do comprimento de seus caminhos quando a rede é submetida a falhas. O primeiro identifica o número de servidores conectados, enquanto a segunda identifica o número de saltos necessários para os servidores se comunicarem. A seguir, essas métricas são detalhadas.

3.1. Tamanho da rede

A análise de tamanho da rede leva em consideração o tamanho relativo máximo e médio dos componentes conexos resultantes após falhas de dispositivos da rede, semelhante às métricas utilizadas no trabalho de Albert *et al.* [Albert et al., 2000]. Na Equação 3, o tamanho relativo do maior componente conexo em relação ao número de servidores existentes, dado por TR_{max}, é dado por:

$$TR_{max} = \frac{\max_{1 \leq i \leq n} |R_i|}{\sum_{i=1}^{n} |R_i|}, \tag{3}$$

onde R_i é o conjunto dos servidores presentes em cada componente conexo i e n é o número de componentes conexos existentes no grafo resultante G'. Já a segunda métrica, tamanho médio dos componentes conexos de G' isolados, desconsidera o componente com o maior número de servidores. O tamanho médio, dado por T_{med}, é dado por:

$$T_{med} = \frac{1}{n-1}(\sum_{i=1}^{n} |R_i| - \max_{1 \leq i \leq n} |R_i|). \tag{4}$$

3.2. Tamanho dos caminhos

A análise do tamanho dos caminhos considera a relação entre o diâmetro do maior componente conexo do grafo G', dado por D^*_{max}, e o diâmetro do grafo original G, dado

por D_{max}. Os valores de diâmetro consideram apenas a distância entre os servidores no grafo. Calcula-se o diâmetro relativo do maior componente conexo como:

$$DR_{max} = \frac{D^*_{max}}{D_{max}}. \quad (5)$$

Além disso, analisou-se a relação da distância média entre os servidores do maior componente conexo do grafo G', dada por D^*_{med} e a distância média entre os nós de G, dada por D_{med}. A distância média é calculada utilizando o comprimento de apenas um dos menores caminhos entre todos os pares de servidores rede. Assim, a distância média relativa do maior componente conexo é definida como:

$$DR_{med} = \frac{D^*_{med}}{D_{med}}. \quad (6)$$

4. Resultados

Os resultados foram obtidos a partir das topologias ilustradas na Seção 2, com algumas variações de seus parâmetros para atingir um número de servidores próximo a 3.400. Esse número foi escolhido pela quantidade elevada de nós que ainda permitem simulações em tempo hábil. Como essas topologias possuem estrutura regular, os resultados podem ser estendidos para números distintos de servidores, tal que respeitem a regra de formação de cada topologia. A Tabela 1 apresenta o nome associado a cada uma das topologias neste trabalho e os seus respectivos parâmetros como o número correspondente de portas por comutador e por servidor. A tabela também apresenta as propriedades das redes que são dadas pelo número de servidores resultantes, porcentagem do número de comutadores em relação ao número total de nós, diâmetro e comprimento médio dos menores caminhos entre servidores. Esses valores são referentes ao grafo G, ou seja, a rede original antes da falha de seus componentes.

Vale observar que, apesar de algumas dessas topologias poderem operar parcialmente completas, considerou-se apenas o uso da rede com topologia completa, ou seja, com todas as interfaces de rede de servidores e comutadores em uso, para melhor avaliar os aspectos fundamentais de cada uma das arquiteturas. Além disso, o número de portas de comutadores não foi limitado em relação aos comutadores disponíveis comercialmente, de forma a ser atingido um número próximo de servidores entre as topologias, o que aumenta o grau de justiça da comparação realizada. Essa justiça ocorre pois um dos principais objetivos de um data center é oferecer capacidade de processamento ou redundância de armazenamento, o que pode ser aumentado com um número maior de servidores. Assim, topologias com mesmo número de servidores atingiriam seus objetivos de forma semelhante se não existissem gargalos ou falhas na rede.

4.1. Falhas de enlace

A Figura 4 mostra os resultados de robustez segundo as métricas de tamanho relativo máximo (TR_{max}) e médio (T_{med}) em relação a porcentagem do total de enlaces retirados do grafo G. De acordo com os valores de TR_{max} da Figura 4(a), a degradação das topologias dependente de servidores pode ser dividida em duas fases. As fases se

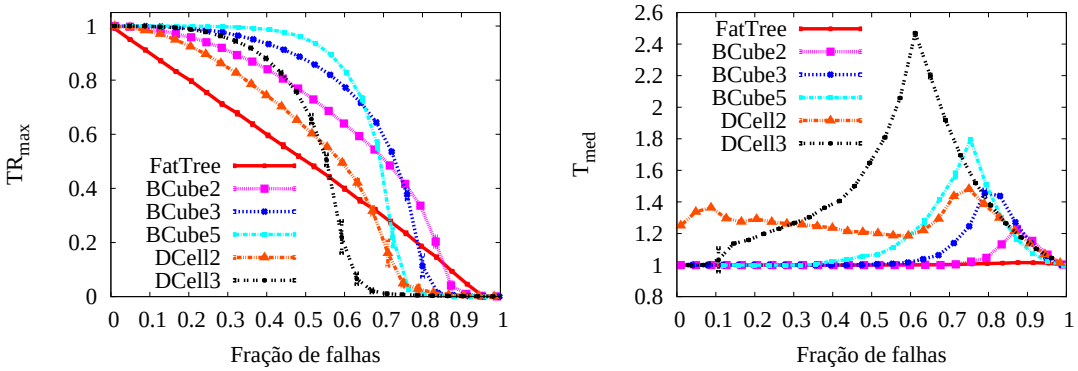
Tabela 1. Configurações das redes utilizadas.

Nome	Portas por comutador	Portas por servidor	Servidores	Porcentagem de comutadores	Diâmetro	Comprimento Médio
Fat-Tree	24	1	3456	17,2	6,0	5,9
BCube2	58	2	3364	3,3	4,0	3,9
BCube3	15	3	3375	16,6	6,0	5,6
BCube5	5	5	3125	50,0	10,0	8,0
DCell2	58	2	3422	1,7	5,0	4,9
DCell3	7	3	3192	12,5	11,0	8,2

diferenciam pelo comportamento da curva, sendo a Fase 1 a região de decaimento aproximadamente logarítmico e a Fase 2, a região de decaimento aproximadamente exponencial. A topologia Fat-Tree, entretanto, possui apenas uma fase de queda de desempenho. Assim, em relação à primeira fase, ou seja, para valores menores de falhas, a topologia Fat-Tree apresenta uma degradação mais rápida em relação ao número de componentes conectados. Todavia, apresenta desempenho superior às outras topologias a partir, aproximadamente, do ponto de mudança de transição de fase devido ao decaimento exponencial das outras topologias. Esse comportamento exponencial será explicado mais adiante.

Um resultado interessante ao comparar topologias do mesmo tipo formadas com diferentes parâmetros, como a BCube e a DCell, mostra que as topologias com número maior de portas de servidores possuem degradação mais suave na primeira fase. Esse comportamento pode ser explicado pelos resultados da Figura 4(b), que mostra a avaliação do tamanho médio (T_{med}) das topologias. Nessa análise, são desconsideradas as amostras do experimento realizado que contém apenas um único componente conexo. Observa-se nesses resultados que o tamanho médio dos componentes conexos, desconsiderando o maior deles, é baixo para todas as redes, possuindo maior valor de pico de 2,5 nós, no caso da topologia DCell2. Isso indica que o maior componente conexo se desconecta em componentes com um número pequeno de servidores, semelhante ao caso de redes com distribuição exponencial de grau submetidas a falhas aleatórias de seus nós [Albert et al., 2000]. Esse comportamento pode ser explicado pelo fato das redes de *data center* consideradas possuírem uma fração significativa de enlaces com nós servidores, existindo uma probabilidade maior da aresta escolhida para falhar ser um enlace conectado diretamente a um servidor. Como a degradação da rede é representada em boa parte por desconexões de conjuntos de servidores, topologias com maior número de portas de servidor serão mais tolerantes a falhas de enlaces.

Os resultados da Figura 4(a) também mostram que, para um mesmo número de portas de servidores, a topologia BCube possui desempenho superior à DCell. Isso ocorre, pois a topologia DCell utiliza servidores para realizar a principal conexão entre sub-redes DCell, enquanto a BCube utiliza comutadores para essa função. Assim, como a falha de enlaces de servidores é mais frequente, o número de servidores a se desconectarem do maior componente conexo é maior no caso da DCell para um mesmo número de falhas de enlace. Essa afirmação é comprovada pelos resultados da Figura 4(b), na qual o tamanho médio dos componentes conexos da DCell é maior do que a BCube para um número igual de portas por servidor. A topologia Fat-Tree possui o pior desempenho por possuir apenas uma porta de servidor, como visto na Figura 4(a).



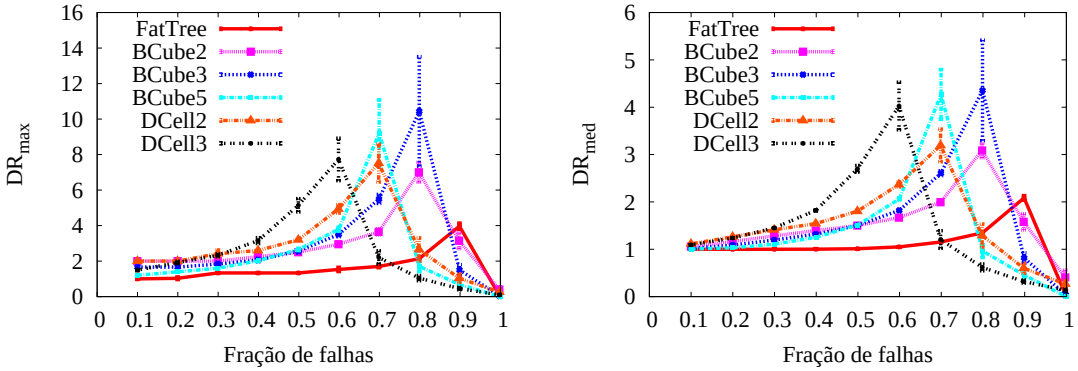
(a) Maior componente conexo. (b) Outros componentes.

Figura 4. Tamanho da rede resultante - falhas de enlace.

Apesar das conclusões da análise anterior, de acordo com a Figura 4(a), a partir do ponto próximo à sua transição de fase, um mesmo tipo de topologia possui desempenho inferior se utilizar um maior número de portas. Esse caso é uma generalização da situação na qual, a partir de um determinado número de falhas, a Fat-Tree possui desempenho superior às topologias baseadas em servidores. Esse comportamento ocorre, pois a partir de certo número de desconexões de servidores do maior componente conexo, o número de enlaces entre comutadores é comparável ao número restante dos de servidores, aumentando a probabilidade de enlaces de comutadores falharem. Como a exclusão de enlaces de comutadores desconecta um maior número de servidores da rede, a velocidade de degradação aumenta, impactando proporcionalmente o número de comutadores, explicando a diminuição exponencial. Essa exclusão de um maior número de servidores pode ser mostrada pelo comportamento da Figura 4(b), na qual o tamanho médio dos componentes conexos cresce até atingir um valor de pico, quando passa a decrescer da mesma forma que os resultados de trabalhos reconhecidos da literatura para redes exponenciais [Albert et al., 2000]. O decrescimento ocorre, pois o número de componentes aumenta, diminuindo o tamanho do maior componente e, consequentemente, afetando a probabilidade de remoção das arestas dos componentes isolados. O ponto de pico coincide com o ponto de mudança de concavidade do gráfico da Figura 4(a), pois é o ponto no qual o tamanho do maior componente decresce com maior rapidez, aumentando a probabilidade de ocorrência de falha nos componentes isolados.

A partir da análise apresentada, conclui-se que quanto maior a dependência do número de portas extras nos servidores, que é nula no caso da Fat-Tree, mais lenta é a degradação de tamanho até um determinado ponto no qual a situação se inverte. Entretanto, esse ponto ocorre para valores altos de falha e pode não representar uma situação real. Além disso, mostra-se que a topologia DCell possui maior degradação do que a BCube em relação ao tamanho da rede devido à sua necessidade de utilizar servidores para interconectar diferentes sub-redes DCell.

A Figura 5 mostra a evolução do tamanho dos caminhos na rede. A curva do diâmetro da rede, vista na Figura 5(a), é semelhante ao do tamanho médio dos caminhos (Figura 5(b)), havendo apenas diferença de seus valores. Os resultados mostram que todas as curvas possuem um pico de diâmetro, podendo ser observado que a falha de



(a) Diâmetro Relativo. (b) Comprimento Médio Relativo.

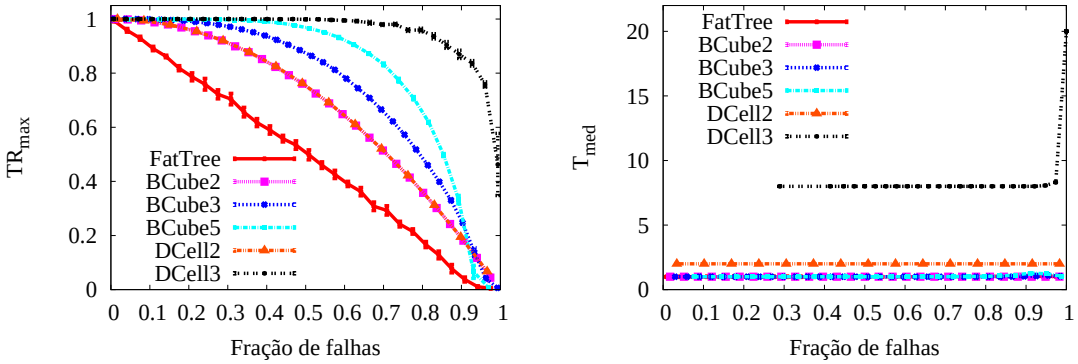
Figura 5. Tamanho dos caminhos da rede resultante - falhas de enlace.

enlace elimina caminhos mais curtos até um ponto no qual os caminhos se encurtam devido à diminuição da rede. Outra característica desses resultados é o fato de o aumento dos caminhos ser elevado, chegando até a quatro vezes o tamanho médio no visto na Figura 5(b). A topologia com o melhor desempenho nessa análise é a Fat-Tree, apesar de apresentar o pior desempenho em relação ao tamanho da rede. Essa observação mostra a necessidade de avaliar a robustez considerando mais de uma característica da rede.

4.2. Falhas de comutador

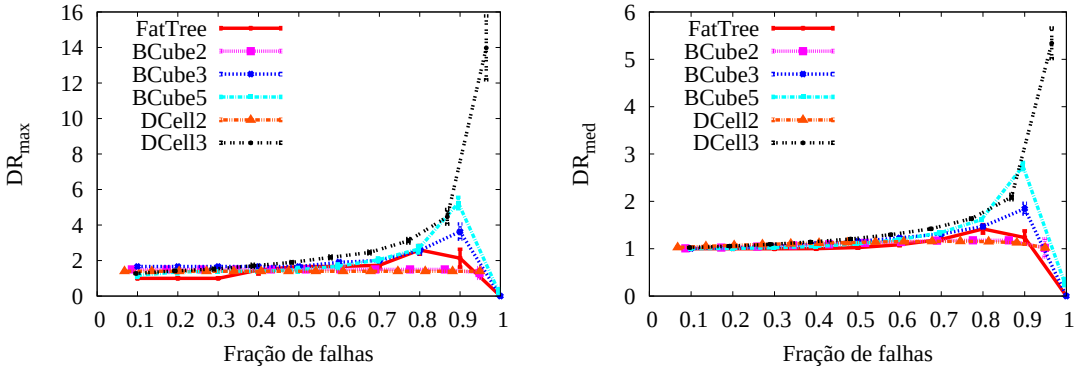
Os resultados da Figura 6 mostram a evolução do tamanho da rede considerando falhas nos comutadores. A avaliação de TR_{max}, na Figura 6(a) possui curva com comportamento parecido ao caso de falha de enlaces da Figura 4(a). Entretanto, a região relativa à segunda fase é insignificante. Consequentemente, o gráfico de T_{med}, na Figura 6(b), não possui picos significativos, à exceção da rede DCell3 que aumenta em aproximadamente 20 vezes o seu tamanho médio após a extinção da maior parte dos comutadores. Desconsiderando esse comportamento, é observado que a falha de comutadores produz componentes isolados de tamanho constante. É interessante notar a robustez da rede DCell3 em relação ao tamanho da rede. A Figura 6 mostra que o desempenho da DCell3 em relação ao seu tamanho apresenta queda apenas para valores maiores que 60% de falhas. Comparado ao desempenho das redes Fat-Tree e BCube3, que possuem porcentagem de comutadores de mesma ordem de grandeza (Tabela 1), a DCell3 possui desempenho superior. Além disso, a Figura 6(b) mostra que a rede não possui pontos para valores pequenos de falhas devido à inexistência de componentes isolados. O desempenho superior da DCell3 será melhor entendido na próxima seção. A comparação dos resultados da Figura 6 entre as topologias BCube e Fat-Tree mostra a dependência do nível de tolerância e do número de portas utilizadas pelos servidores, no qual a Fat-Tree apresenta o pior desempenho por não considerar o encaminhamento entre servidores. Entre as configurações da topologia DCell, também pode ser observada essa relação. A DCell2 possui o mesmo desempenho da BCube2, possivelmente por utilizar o mesmo número de portas de comutador.

Os resultados da Figura 7 mostram que a falha de comutador em geral dobra o diâmetro da rede, possuindo picos apenas em valores próximos de 90% de falha na maioria das redes. Além disso, o tamanho médio dos caminhos pouco se altera para valores menores que 90% de remoções de comutador. Com isso, conclui-se que a falha de comu-



(a) Maior componente conexo. (b) Outros componentes.

Figura 6. Tamanho da rede resultante - falhas de comutador.



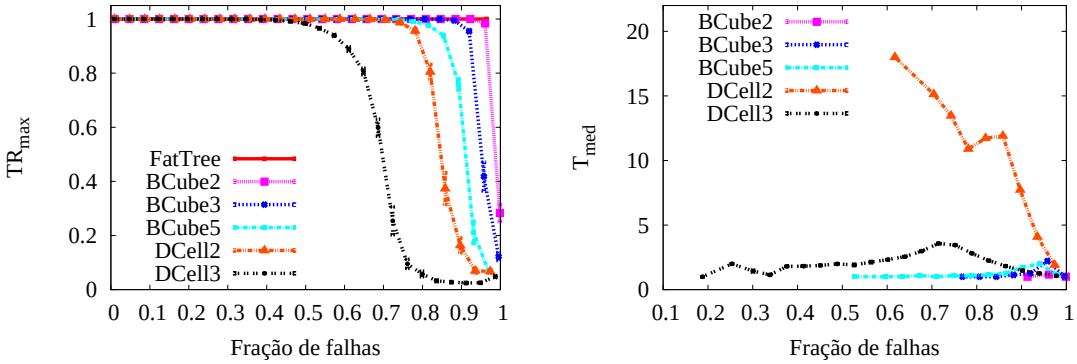
(a) Diâmetro Relativo. (b) Comprimento Médio Relativo.

Figura 7. Tamanho dos caminhos da rede resultante - falhas de comutador.

tador possui um impacto baixo no tamanho dos caminhos da rede.

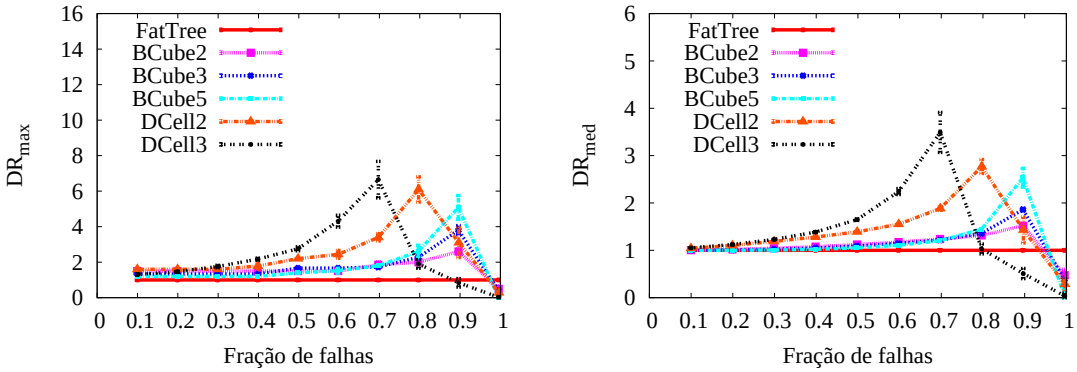
4.3. Falhas de servidor

A Figura 8 mostra os resultados para remoção de servidores. É importante notar que nas análises anteriores o termo $\sum_{i=1}^{n} |R_i|$ das Equações 3 e 4 é constante e representa o número total de servidores da rede pois não há exclusão desse tipo de componente. Nesta análise, porém, $\sum_{i=1}^{n} |R_i|$ reduz-se de uma unidade a cada remoção de servidor. Assim, como mostra a Figura 8(a), a rede Fat-Tree apresenta robustez máxima ($TR_{max} = 1$). Isso ocorre, pois a remoção não induz componentes desconexos, visto que a rede não depende de interconexão por servidores. Os resultados mostram também que as demais redes, à exceção da DCell3, possuem melhor robustez a esse tipo de falhas do que a observada para os casos de remoção de comutadores e enlaces. Assim, mostra-se que do ponto de vista de robustez pelo tamanho da rede, as redes baseadas em servidores são viáveis. No caso da rede DCell3 mostrou-se na Seção 4.2 que ela possui robustez alta para falhas de comutadores, o que induz a hipótese que parte significativa da rede se mantém conectada através de servidores. Os resultados da Figura 8(a) comprovam essa hipótese pois essa rede é a que apresenta o maior decaimento do TR_{max} de acordo com o aumento das falhas de servidor. Os resultados de T_{med} da Figura 8(b) mostram que os componentes isolados são



(a) Maior componente conexo. (b) Outros componentes.

Figura 8. Tamanho da rede resultante - falhas de servidor.



(a) Diâmetro Relativo. (b) Comprimento Médio Relativo.

Figura 9. Tamanho dos caminhos da rede resultante - falhas de servidor.

pequenos em relação ao número total de nós, que é aproximadamente de 3.400, e possui o mesmo comportamento de pico do caso de falha de enlaces, induzindo também mudança de concavidade no gráfico de TR_{max}. A ausência de pontos na Figura 8(b) para falhas até um certo valor, específico para cada tipo de rede, mostra que as topologias demoram a possuir componentes isolados, à exceção da DCell3 que possui pior desempenho. Os resultados da Figura 9 para tamanho de caminho, possuem o mesmo comportamento da Seção 4.1, apresentando picos e mostrando o impacto significativo da remoção de servidores em topologias que dependem do encaminhamento desses componentes.

5. Conclusões

Neste trabalho avaliou-se o comportamento de topologias intra *data center* propostas na literatura quando são submetidas a falhas de seus componentes. A partir da análise foi mostrado o compromisso na escolha de uma topologia em termos de robustez, referente ao tipo de falha que a topologia é mais sensível. Observou-se que a degradação da rede ocorre pela desconexão de um pequeno número de servidores e que, para topologias dependentes de encaminhamento por servidores, o tamanho dos caminhos aumenta consideravelmente em consequência das falhas induzidas. Além disso, mostrou-se que isoladamente o tamanho da rede ou o tamanho dos caminhos não fornecem uma maneira

eficaz de avaliar o desempenho da topologia e devem ser utilizadas de maneira combinada. Assim, como trabalho futuro, pretende-se propor uma métrica de robustez que leve em consideração as diferentes medidas utilizadas neste trabalho. A importância de cada medida dependerá da principal aplicação utilizada no *data center*, podendo ser ajustada através da atribuição de pesos para cada uma. Por exemplo, uma aplicação com requisitos maiores de latência poderá possuir maior dependência em relação ao tamanho de seus caminhos, enquanto uma aplicação com nível alto de paralelismo poderá precisar de redes com melhor robustez em relação ao tamanho de seus componentes conexos. A partir da métrica de robustez proposta será possível combiná-la com outras métricas, como de análise de capacidade de transferência de bits da rede ou seu custo financeiro, de forma a escolher a melhor topologia a ser utilizada para uma determinada aplicação.

Referências

Al-Fares, M., Loukissas, A. e Vahdat, A. (2008). A scalable, commodity data center network architecture. Em *ACM SIGCOMM*, p. 63–74.

Albert, R., Jeong, H. e Barabási, A. (2000). Error and attack tolerance of complex networks. *Letters to Nature*, 406(6794):378–382.

Dean, J. e Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51:107–113.

Gill, P., Jain, N. e Nagappan, N. (2011). Understanding network failures in data centers: measurement, analysis, and implications. Em *ACM SIGCOMM*, p. 350–361.

Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D. A., Patel, P. e Sengupta, S. (2009). VL2: a scalable and flexible data center network. Em *ACM SIGCOMM*, p. 51–62.

Guo, C., Lu, G., Li, D., Wu, H., Zhang, X., Shi, Y., Tian, C., Zhang, Y. e Lu, S. (2009). BCube: a high performance, server-centric network architecture for modular data centers. Em *ACM SIGCOMM*, p. 63–74.

Guo, C., Wu, H., Tan, K., Shi, L., Zhang, Y. e Lu, S. (2008). DCell: a scalable and fault-tolerant network structure for data centers. Em *ACM SIGCOMM*, p. 75–86.

Hagberg, A., Swart, P. e S Chult, D. (2008). Exploring network structure, dynamics, and function using NetworkX. Relatório técnico, Los Alamos National Laboratory (LANL).

Miller, R. (2011). Google uses about 900,000 servers. Acessado em `http://www.datacenterknowledge.com/archives/2011/08/01/report-google-uses-about-900000-servers/`.

Popa, L., Ratnasamy, S., Iannaccone, G., Krishnamurthy, A. e Stoica, I. (2010). A cost comparison of datacenter network architectures. Em *ACM Co-NEXT '10*, p. 16:1–16:12.

Raiciu, C., Barre, S., Pluntke, C., Greenhalgh, A., Wischik, D. e Handley, M. (2011). Improving datacenter performance and robustness with multipath TCP. Em *ACM SIGCOMM*, p. 350–361.

Verdi, F., Rothenberg, C., Pasquini, R. e Magalhães, M. (2010). Novas arquiteturas de data center para cloud computing. Em *Minicursos do XXVIII SBRC*, p. 103–152.