

UNIVERSIDADE LUTERANA DO BRASIL

ULBRA – *CAMPUS* GUAÍBA

CURSO DE SISTEMAS DE INFORMAÇÃO



**IMPLEMENTAÇÃO DE ALTA  
DISPONIBILIDADE PARA APLICAÇÕES COM  
RECURSOS GEOGRAFICAMENTE  
DISPERSOS  
TRABALHO DE CONCLUSÃO DE CURSO I**

ALEXANDRE FAGUNDES BISCAIO

ANDRÉ PERES  
Orientador

Guaíba, Dezembro de 2008.

## **DADOS DE IDENTIFICAÇÃO**

Acadêmico: Alexandre Fagundes Biscaio

E-mail: [biscaio@terra.com.br](mailto:biscaio@terra.com.br)

Professor Orientador: André Peres

E-mail: [peres@guaiba.ulbra.tche.br](mailto:peres@guaiba.ulbra.tche.br)

Título do Projeto: Implementação de Alta Disponibilidade para Aplicações com Recursos Geograficamente Dispersos

Período de realização: 01/08/2008 à 28/11/2008

## SUMÁRIO

|   |  |    |
|---|--|----|
| 1 | INTRODUÇÃO .....   | 4  |
| 2 | MECANISMOS DE ALTA DISPONIBILIDADE.....                    | 6  |
| 3 | CENÁRIO PROPOSTO .....                                     | 16 |
| 4 | APLICAÇÃO DE ALTA DISPONIBILIDADE E TESTES PROPOSTOS ..... | 23 |
| 5 | CONCLUSÕES .....   | 27 |
| 6 | TRABALHO DE CONCLUSÃO DE CURSO - II .....                  | 27 |
| 7 | REFERÊNCIAS .....  | 28 |

## 1 INTRODUÇÃO

*Storage*, ou área de armazenamento, é onde os dados são armazenados de forma não volátil, ou seja, mesmo que um computador seja desligado, ao ligá-lo novamente os dados armazenados estarão acessíveis, pois os mesmos são gravados de forma magnética (ou óptica) a dispositivos chamados discos rígidos (ou HD- *Hard Drives*) ou discos ópticos, como mídias de DVD. Sua importância para sistemas de informação está na capacidade de manter os dados providos por esses sistemas aos usuários que os utilizam a fim de suportar uma determinada área de negócio de uma empresa ou mesmo como usuários finais no caso, por exemplo, de sistemas online, como a própria *Internet* (rede mundial de computadores).

Este Trabalho de Conclusão de Curso apresenta uma proposta para o desenvolvimento de uma solução capaz de fornecer alta disponibilidade à camada de acesso aos dados (*storage*) por aplicações que necessitem de alta disponibilidade. Considerando que todas as demais camadas de infra-estrutura são facilmente contingenciáveis, apenas a camada de *storage* possui poucas soluções com custo acessível que possam ser implementadas por pequenas ou médias empresas devido ao custo envolvido em tais soluções, já que usualmente são considerados *storages* e *softwares* dos maiores *players* desta área no mercado.

Alinhado a essa necessidade agrega-se outros motivadores para a busca de uma solução que possa prover redundância à camada de *storage*: a) o percentual de empresas que já sofreram alguma interrupção dos serviços de Tecnologia da Informação e b) os motivos que levam as interrupções do funcionamento de aplicações por falhas de infra-estrutura.

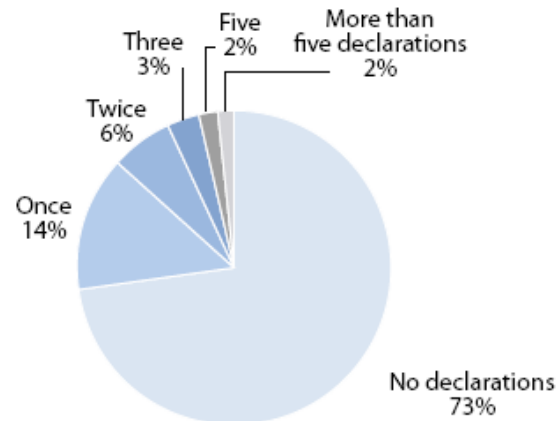
De acordo com o relatório *Disaster Recovery Journal-October 2007-Global Disaster Recovery Preparedness Online Survey*, da *Forrester Research, Inc* (BALAOURAS, 2008), mais de um quarto das empresas entrevistadas já experimentaram interrupções em seus serviços de TI nos últimos 5 anos de operação, ocasionando conseqüentemente interrupções nos negócios da empresa, trazendo danos ao faturamento, imagem e serviços oferecidos (muitas vezes, serviços essenciais a sociedade), conforme demonstrado na Figura 1.

---

**Figure 1** More Than A Quarter Of Companies Have Declared A Disaster In The Past Five Years

---

**“How many times have you had to declare a ‘disaster’ and recover operations at your recovery site in the past five years?”**



Base: 250 disaster recovery decision-makers and influencers at businesses globally  
(percentages may not total 100 because of rounding)

Source: Forrester/*Disaster Recovery Journal* October 2007 Global Disaster Recovery Preparedness Online Survey

Source: Forrester Research, Inc.

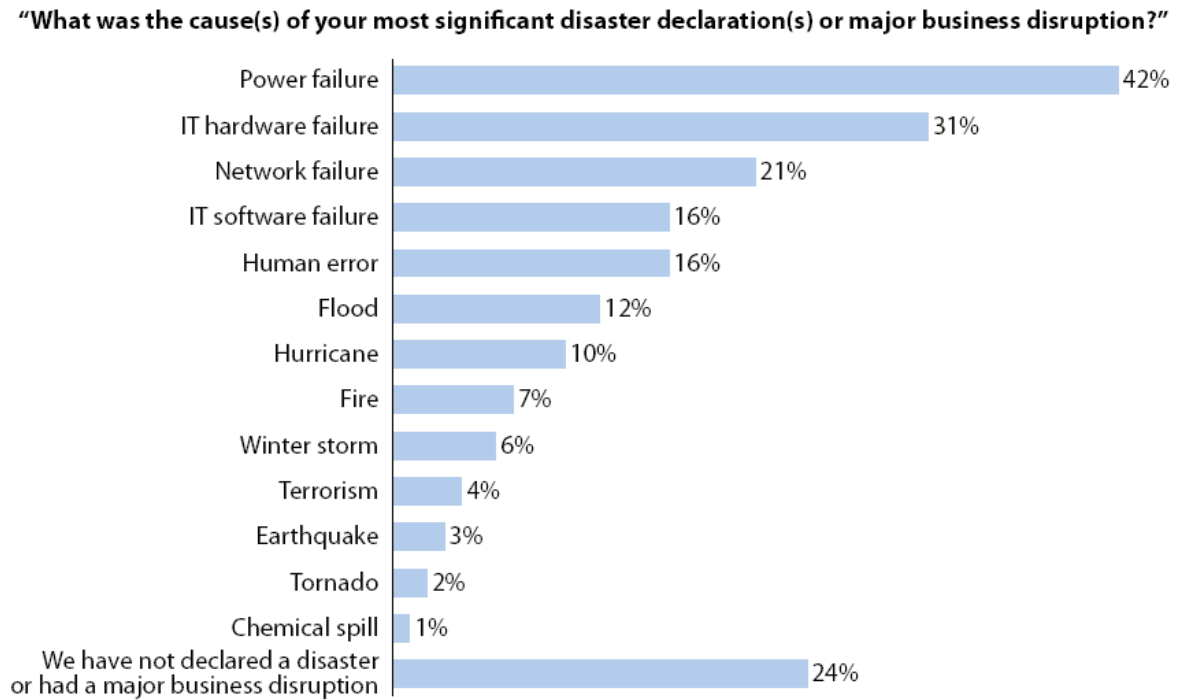
Figura 1: Mais de um quarto das companhias declararam um desastre nos últimos cinco anos

Agregado a isso temos também os motivos destas falhas e o quanto cada uma representa, percentualmente, como pontos a serem tratados a fim de diminuir-se essas falhas, ou propor cenários de infra-estrutura que possam suportar a maioria destas falhas melhorando assim a disponibilidade de sistemas de Tecnologia de Informação. Tais fatores são mostrados abaixo na Figura 2. (BALAOURAS, 2008)

---

**Figure 2** Power Failures Are The Most Common Cause Of Declared Disasters And Downtime
 

---



Base: 250 disaster recovery decision-makers and influencers at businesses worldwide (multiple responses accepted) (Does not include those who answered “Other” or “Don’t know”)

Figura 2: motivos das falhas em infra-estrutura que interromperam serviços de Tecnologia da Informação nas empresas pesquisadas

## 2 MECANISMOS DE ALTA DISPONIBILIDADE

Alta Disponibilidade é um conjunto de técnicas e tecnologias combinadas para entregar a usuários ou clientes acesso aos recursos de que necessitam sempre que necessário, contando com elementos redundantes ou seja, duplicados, para que em caso de falha de parte dos recursos, os demais que são redundantes possam assumir as funcionalidades exercidas pelos recursos que falharam.

Este capítulo abordará as tecnologias e técnicas existentes para a implementação de Alta Disponibilidade para aplicações, considerando todas as camadas de infra-estrutura, normalmente aplicada no mercado, partindo da camada mais alta de conectividade de borda e segurança até a camada mais baixa que é a camada de armazenamento (ou *Storage*), conforme mostrado na Figura 3.

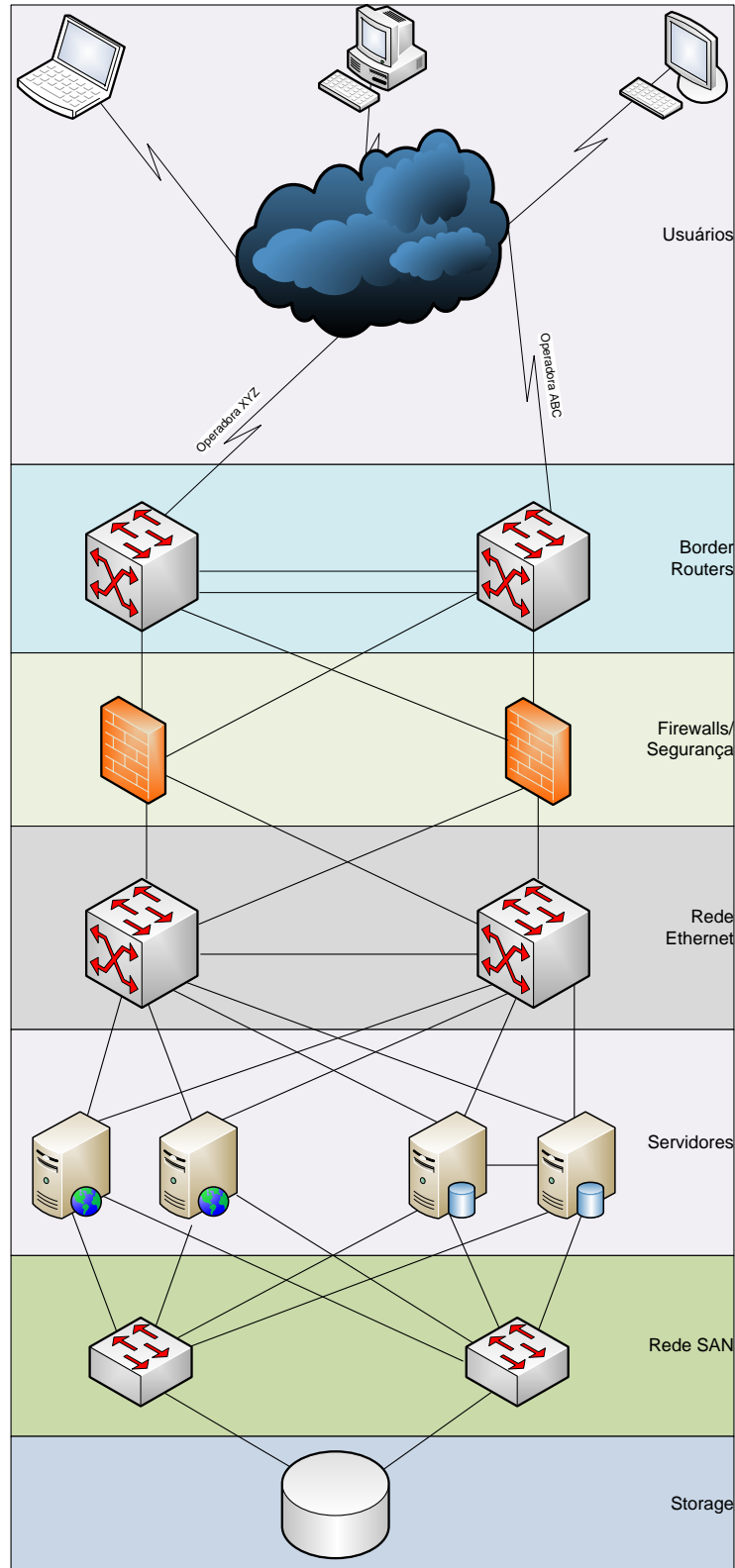


Figura 3: Infra-estrutura redundante normalmente utilizada

Como este trabalho está voltado à Tecnologia da Informação, estão sendo desconsideradas camadas ainda inferiores como alimentação elétrica

redundante, refrigeração redundante e controles físicos de segurança (acesso físico à infra-estrutura).

Os próximos itens do texto descrevem cada uma das camadas apresentadas na Figura 1.

## **2.1 BORDER ROUTERS**

Esta camada normalmente é tornada redundante colocando-se mais de um meio de acesso à *Internet*. Este meio de acesso deve ser adquirido de operadoras de Telecomunicações diferentes. Para garantia do serviço em caso de falhas de disponibilidade em uma prestadora do serviço cada um dos links deve ser conectado a mais de um *switch/router*, os quais através de protocolos de roteamento e aplicações embutidas de balanceamento de carga, provêm redundância para o tráfego de entrada de dados e de saída. Da mesma forma, as conexões com a camada imediatamente inferior também são redundantes, onde devem haver conexões entre cada equipamento desta camada com os *Firewalls* e outros *appliances* de segurança (como *IPS-Intruder Protection System*, Sistemas Antivírus, etc). (KLOTH, 2005)

## **2.2 FIREWALLS/SEGURANÇA**

A camada de segurança é composta por segurança de perímetro, segurança de *gateway* e segurança de *endpoints*.

A segurança de perímetro que normalmente é composta por *Firewall* e *IPS (Intruder Protection System)*, tem sua redundância implementada geralmente no modo *active/standby* ou *failover*(onde um elemento redundante passivo assume as funcionalidades do elemento que falhou) e *active/active* (onde elementos redundantes compartilham os acesso de maneira ativa e em caso de falha de um os demais devem suportar, além da sua, a carga do que falhou), onde um equipamento assume plenamente as funções do outro que entrou em uma condição de erro ou falha.

A infra-estrutura de segurança de perímetro tem por objetivo garantir que somente o tráfego autorizado pelas políticas de segurança da empresa ocorra, assim como a análise e detecção de falhas de segurança neste tráfego sejam apontadas, podendo ser bloqueadas ou somente arquivadas (dependendo do nível de risco das

mesmas para a empresa). Por se tratar de uma camada crítica na infra-estrutura, esta camada não somente está voltada a garantir a conectividade redundante, mas também visa garantir a disponibilidade, integridade e confidencialidade (que são mecanismos para a aplicação das políticas de acesso e uso), que pode ser provido através de VPNs (*virtual private networks*). (NORTHCUTT, 2005)

Ainda em caso de falha total, pode-se fazer uso dos *borders routers* para manter a disponibilidade, no entanto, a segurança de acesso e uso estará em risco, pois os *borders routers* normalmente não refletem todas as regras de segurança impostas no *Firewall* e *IPS*.

Esses ativos devem também prover conectividade redundante à camada inferior (Rede *Ethernet* que atende os servidores, e por conseqüência os serviços de TI por eles oferecidos).

A segurança de *gateway* e *endpoints* têm por objetivo restringir, monitorar, analisar e permitir o acesso a conteúdo condizentes com a política de segurança da empresa e livres de riscos, como vírus, spams, páginas com conteúdo malicioso e *malwares* diversos.

A alta disponibilidade desses gateways, a exemplo da infra-estrutura de segurança de perímetro, é realizada normalmente no modo *active/active* e *active/standby*.

## 2.3 REDE ETHERNET

Esta é a camada que deve garantir conectividade entre os servidores (camada inferior) e a camada de *Firewalls*/Segurança (camada superior), fornecendo meios de disponibilizar as aplicações (providas pelos servidores) aos seus usuários. Devido a questões de implementação de segurança ou mesmo de melhora no tráfego ethernet existem diversos cenários possíveis para a implementação desta camada. A mais tradicional segue o padrão das já apresentadas, onde dois ou mais equipamentos (no caso, *Switches*) interconectam os equipamentos das camadas superiores e inferiores com conexões que atendem todos os equipamentos ou seja, existindo dois *Firewalls*, A e B, e dois *Switches* X e Y, o *Switch* X terá conexões físicas com os *Firewalls* A e B, assim como o *Switch* Y as terá. Outro ponto comum é que existe interconexão entre os dois ou mais *Switches*, provendo comunicação

entre os Servidores (camada inferior) e o restante da topologia (camadas superiores ou inferiores) sem que o tráfego necessite subir uma camada na infra-estrutura necessitando de tratamento (roteamento na maioria dos casos) para voltar a mesma camada em que se originou (no caso outros servidores que são atendidos por outros *Switches*). Este tipo de implementação visa minimizar o impacto causado por falhas de conexões em cabos, ou mesmo *Tranceivers* (portas do *Switch*) que atendam as conexões entre as camadas superiores e inferiores, bem como minimizar o custo de conectividade com as camadas superiores, que normalmente possuem menor capacidade de conexão (portas disponíveis), conforme apresentado na Figura 4.(SCHMIDT, 2006)

Com conexões redundantes a implementação do protocolo STP (*spanning-tree protocol*) se faz obrigatória, pois um gerenciamento dos links entre os equipamentos se faz necessário para evitar *loops* na topologia de rede. Este protocolo, desenvolvido pela *Digital Equipaments Corporation*, criando a especificação IEEE 802.1d, tem como função determinar o melhor caminho para o tráfego, já que mais de um caminho existe (*loop*).

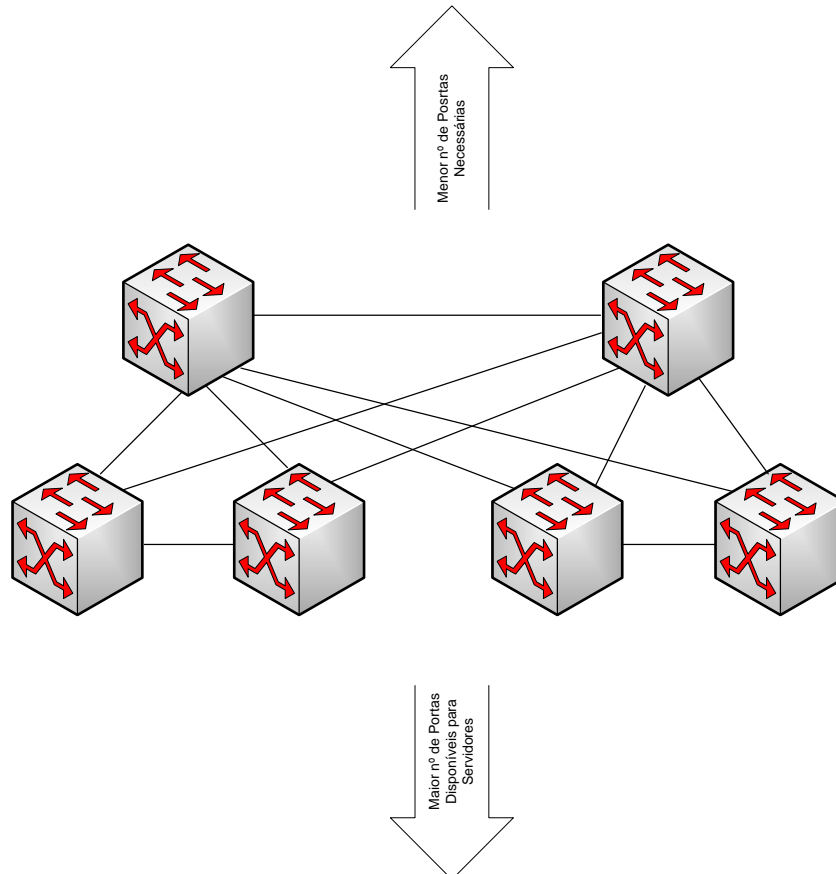


Figura 4: Infra-estrutura de Rede redundante

## 2.4 SERVIDORES

A camada de servidores possui algumas características mais peculiares do que as já apresentadas. A questão de manter servidores disponíveis e contingentes é mais abrangente do que somente recursos físicos, já que não podemos deixar de considerar que disponibilizar um servidor, que possui a instalação e configuração de um sistema operacional, preparação e configuração dos dados a serem disponibilizados e outras questões deste nicho, podem ser muito mais dispendiosas do que a configuração de um *Switch* de rede (onde um simples *restore* de um arquivo de configuração em um equipamento sem configuração alguma pode disponibilizar os serviços em um tempo mínimo).

Para este fim, componentes como placas controladoras de discos, discos rígidos, pentes de memória e placas de rede devem ser contingentes.

Mesmo a questão de configuração de discos que suportarão o Sistema Operacional (apesar de não possuírem dados) devem desde o início contar com tecnologias como *RAID (Redundant Array of Independent Disks)* que dependendo da necessidade podem ser disponibilizados de algumas formas ainda mais peculiares, considerando a segurança na falha de um disco rígido (considerando os tipos de proteção mais comumente utilizados)(GOUVEIA, 2005).

Seguindo a linha de redundância de componentes, alguns fabricantes de *Mother Boards* implementam em seus produtos redundância para os pentes de memória, onde estes disponibilizados em pares trabalham com a mesma tecnologia do RAID 1 implementado para discos (*mirror*) onde são espelhados de forma a prover contingência um para um.

Uma questão importante e já citada implicitamente na camada de Rede (*Switches* que atendem as conexões com os servidores) é que deve existir placas de redes – *NIC's (Network Interface Cards)* também redundantes, de forma que cada segmento a que o servidor esteja conectado sejam atendidas por duas *NIC's*, conectadas a *Switches* diferentes, provendo assim alta disponibilidade a conectividade do mesmo.

Assim como as *NIC's*, as *HBA's (Host Buss Adapter)* que tem a função de conectar os servidores aos *Switches SAN (Storage Area Network)* para prover acesso ao Storage (onde estão armazenados os dados disponibilizados aos

usuários pelas aplicações), que estão nas últimas camadas da infra-estrutura. As HBA's devem estar cada uma conectada a um *Switch SAN* diferente, e uma aplicação instalada em cada servidor (*driver*) garante o acesso aos recursos de Storage em caso de falha da HBA's, cabos, ou outro elemento que impeça o acesso ao Storage por um caminho, tornando assim possível o acesso aos dados por outro caminho. Esta implementação recebe o nome de *Multipath* e normalmente é disponibilizado pelo fabricante do Sistema Operacional instalado, ou pelo próprio fabricante da *HBA* instalada.

Exemplos clássicos destas implementação são o *Nonstop* (da empresa *Tandem*) e o *Continuous Processing* (da *Stratus*) no final da década de 70, onde a primeira, *Tandem*, implementou tolerância a falhas por *software* em seus sistemas e a *Stratus* implementou tolerância a falhas por *hardware*, tornando esses mecanismos transparentes às aplicações. Modelos destes sistemas foram comercializados por outras empresas, com os nomes de *Olivetti CPS32* e *IBM/88*, assim como *VAX 8600*, *IBM 3090*, *VAXft 3000* (1990), *Teradata* (baseados na arquitetura *Intel 80x86*) e *Sequoia* (baseados na arquitetura *Motorola 68000*).

## **2.5 REDE SAN(Storage Area Network)**

Esta é a camada da infra-estrutura responsável por prover conectividade aos *Storages* (onde os dados que as aplicações disponibilizam ficam armazenados de forma centralizada). Conceitualmente esta camada é formada por *Switches Fibre Channel – FC* (composto por portas ópticas, onde os cabos são fibras ópticas), onde cada *Storage Processor* (CPU existente no *Storage* dedicada a tratar o I/O do mesmo) deve possuir conectividade com cada segmento da Rede SAN, ou, contar com a implementação de *Inter-switch link* (ISL) onde se passa a contar com um único *Fabric* (segmento interconectado com caminhos redundantes entre os switches e os Storages sem, no entanto, gerar um *loop* na rede SAN). Atualmente, com a disseminação da utilização do *iSCSI* (onde o protocolo *SCSI* é encapsulado no IP- *internet protocol*), esta topologia está sendo migrada para redes *Ethernet* de alta velocidade, onde combinações de 1 *Gigabit Ethernet* ou 10 *Gigabit Ethernet*, associados ao aumento do MTU (*maximum transmit unit*) para utilização de *Jumbo Frames*, podem em muitos cenários dispensar o acesso ao *Storage* através de fibra ótica, minimizando investimentos e aumentando a convergência

para padrões únicos (rede *Ethernet*), com melhor administração, tornando assim a própria camada de Redes *Ethernet* a camada de acesso ao *Storage*.

## 2.6 STORAGE

Esta camada da infra-estrutura, foco deste trabalho, tem por objetivo disponibilizar dados a diversas aplicações, distribuídos em diversos servidores, com um tempo de resposta adequado, com segurança e com disponibilidade. O início destas premissas se dá pela redundância da alimentação elétrica, disposição de discos em grupos utilizando técnicas RAID (normalmente RAID 5) provendo redundância física a cada grupo de discos (Figura 5), utilização de *hot spares* (normalmente utilizando um disco *hot spare* por *enclosure*), garantindo ainda mais segurança para os grupos de discos (*RAID Groups*), já que este assumiria o papel de um disco em falha em qualquer grupo de disco, podendo-se contar com até dois discos em falha por *enclosure*, conforme mostrado na Figura 6.

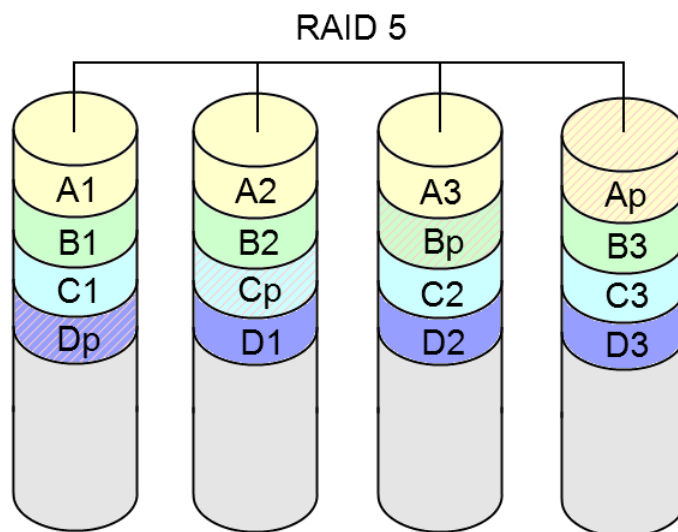


Figura 5: RAID 5-disposição dos discos, dados e paridade (sigla p)

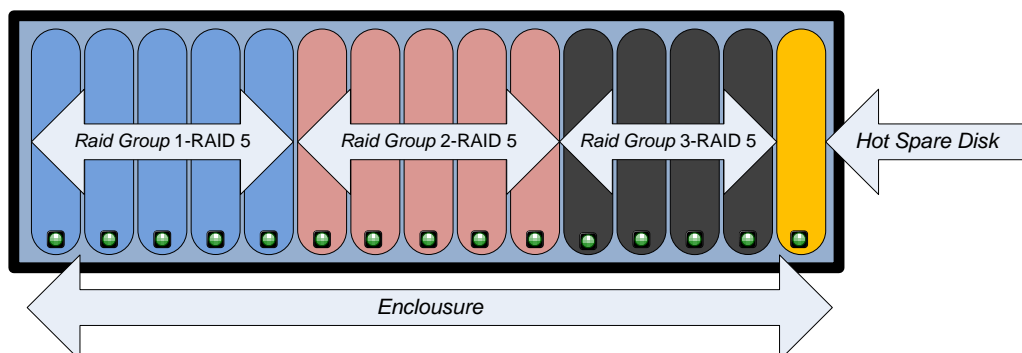


Figura 6: *Enclosure* e *Raid Groups* atendidos por *Hot Spare*

Mesmo com estas técnicas e implementações redundantes, existe uma questão: o *Storage* continua sendo um ponto único de falha, já que não há implementação de *Storages* em *clusters* (de forma que um dado ao ser gravado pudesse ser distribuído em *Storages* distintos). Para minimizar este problema, existem soluções de replicação de *Storages* onde os dados de um *Storage* de produção são replicados para um *Storage* secundário. Essa replicação pode ter duas formas: síncrona, onde os dados são replicados no menor espaço de tempo possível de um *Storage* ao outro, minimizando em muito as perdas de dados em caso de ter-se que contar com os dados do *Storage* replicado, e assíncrona, onde os dados são replicados após um período maior de tempo (normalmente utilizado para aplicações de menor impacto para as áreas de negócio). Normalmente, o que determina a utilização de uma técnica ou outra são os dados de *RPO* (*recovery point objective*) e *RTO* (*recovery time objective*) (DOLEWSKI, 2008) de cada Serviço de TI, já que a escolha por replicações síncronas são mais onerosas em termos financeiros e de utilização da infra-estrutura de TI (conectividade entre os *Storages* e ferramentas de replicação são os principais itens que tornam esta implementação mais onerosa). Porém, mesmo com *Storages* replicados, não há uma forma de disponibilizar os servidores e as próprias aplicações a se utilizarem dos dados que foram replicados, sendo necessários nestes casos que conexões físicas (fibras ópticas, cabos de rede, etc) tenham que ser refeitas. Outro ponto de atenção são as *LUNs* (*logical unit number*) (que são a menor porção de área em disco disponibilizada por um *Storage* a um servidor) e/ou *Meta LUNs* (que são um conjunto de *LUNs* entregues a um Sistema Operacional como se fosse somente uma *LUN*) precisem ser novamente associadas a cada Host. Ainda relacionado a *Storages* os *zonings* (associação entre os hosts e os *Storages*) precisarão ser refeitos e uma série de outros procedimentos em nível de sistemas operacionais, bancos de dados e as próprias aplicações para que sejam possíveis a estas re-estabelecer a comunicação com o *Storage* replicado, como se fosse o *Storage* principal.

O dispêndio de tempo para execução destas alterações na infra-estrutura (apesar de ser mais rápida do que um procedimento de restore de dados, por exemplo, de mídia magnética) normalmente não atende o *RPO* e o *RTO*

(DOLEWSKI, 2008) necessários para suportar uma área de negócio crítica (por exemplo: a central de assinantes de uma empresa de revistas, ou o setor de televendas de uma empresa que vende medicamentos, ou mesmo um hospital). A área de negócio (que financia os investimentos em TI) seria eminentemente afetada em caso de sinistro com o *Storage* principal, mesmo com todas as demais camadas de infra-estrutura sendo redundantes.

Algumas soluções mais robustas (e conseqüentemente mais onerosas em nível de aquisição e manutenção) provêm acesso à camada de *Storage* através da virtualização da mesma (CLARK, 2005), onde *apliances* anexados à SAN gerenciam esta virtualização. Desta forma a existência de *Storages* fisicamente duplicados (independentes de fabricantes) provêm alta disponibilidade para esta camada, conforme apresentado na Figura 7. Obviamente este nível de investimento deverá estar associado ao impacto causado às aplicações no caso de um *Storage*, associado ao RPO e RTO de cada aplicação que se utiliza desta estrutura (DOLEWSKI, 2008).

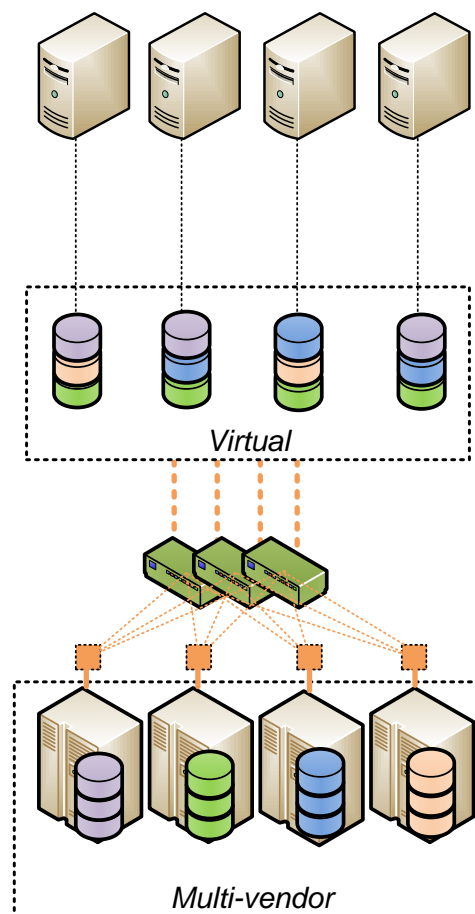


Figura 7: Alta Disponibilidade de Storage por Virtualização

### 3 CENÁRIO PROPOSTO

Neste capítulo serão mostrados os problemas existentes para a disponibilização de uma aplicação em ambiente de Alta Disponibilidade associados aos cenários existentes para contornar estes problemas. Serão elencados os motivos relacionados à camada de *Storage* que precisam ser contornados a fim de atingir este objetivo.

A fim de desenvolver um cenário que possa disponibilizar alta disponibilidade a uma aplicação, deve-se primeiro considerar uma questão sobre o conteúdo existente em cada servidor, qualificando assim os dados existentes nos discos alocados em cada um destes: sistema operacional, aplicações e dados.

Começando pelo terceiro ponto, fica explícito que a boa prática de colocar os dados existentes em um servidor (seja ele um servidor de arquivos, de banco de dados, de páginas web ou com outra função) em uma unidade de disco distinta da unidade utilizada para o sistema operacional pode reduzir alguns riscos: em caso de o disco que contém o sistema operacional sofrer algum dano, os dados estarão resguardados, assim como alterações no sistema operacional, ataque por vírus ou outras alterações que impeçam o sistema operacional de conseguir entregar os dados referentes ao seu papel em uma rede, não ocasionarão danos aos dados por este armazenado, principalmente se esta unidade de disco não for local a um servidor e sim disponibilizada por um *Storage*. Desta forma, seria possível reinstalar o servidor preservando intactos os dados entregues por este, não necessitando, dependendo do caso, realizar-se um restore de dados, o que normalmente é muito dispendioso (devido à baixa velocidade de unidades de fitas magnéticas).

Sobre o primeiro e o segundo ponto (sistema operacional e aplicações) podemos considerar como principal problema a possibilidade de ter dois servidores (*hardwares*) distintos instalados e mantidos com a mesma configuração (cenário este exigido por aplicações dispostas em ambiente de alta disponibilidade). Isso, além de ser um desafio em tempo de instalação e configuração, também é desafiado pelo processo de manterem-se as mesmas versões de binários, bibliotecas e arquivos de configuração em servidores diferentes ao longo da vida útil destes. Nestes casos quanto mais servidores (para aumentar a disponibilidade) maior o

custo de administração. Outro ponto sobre esse assunto é que, via de regra, a instalação de sistemas operacionais e aplicações que este deva suportar são realizadas em discos locais (Figura 8), contando-se com tecnologias de RAID para maximizar a disponibilidade pela redundância de discos rígidos que contém esta instalação, o que dificulta a disponibilização dos serviços providos por um servidor em casos de execução de procedimentos de recuperação (em caso de perda do servidor por falhas de *hardware*, por exemplo).

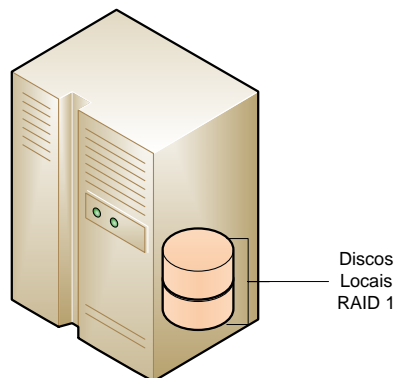


Figura 8: discos locais: sistema operacional e aplicações

Para cada caso de falha, abaixo será citado algumas possibilidades de contorno para prover alta disponibilidade para aplicações disponibilizadas por um servidor:

**Falha de disco rígido local** : partição ou *virtual disk* deve estar configurada sobre RAID (normalmente RAID 1 ou 5), desta forma o acesso ao sistema de arquivos não é interrompido, apesar de a performance ser sensivelmente afetada pelo cálculo da paridade para acesso aos arquivos (inerentes à tecnologia RAID).

**Falha da controladora que atende os discos:** neste caso, para prover-se redundância da controladora, deverá haver duas controladoras que suportem tecnologia RAID e poderá ser configurado RAID 1, 0+1, 10,0+5 ou 50 para atender os *virtual disks* ou partições onde ficarão armazenado o sistema de arquivos (usualmente o mais aplicado é a disposição dos discos em RAID 1 por ser menos oneroso). Esta técnica é conhecida como *Duplexing*, pois duplica o hardware

envolvido no acesso ao disco. Na Figura 9 há um diagrama demonstrando a disposição de um RAID 1 com *duplexing*.

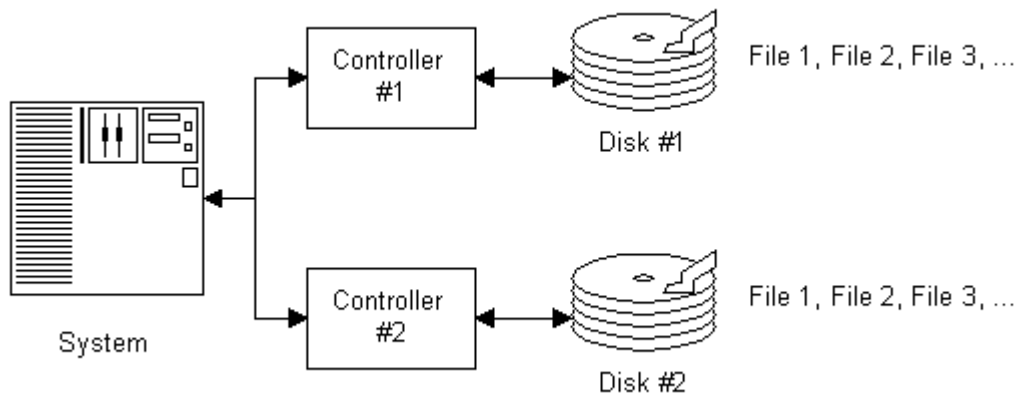


Figura 9: disposição de RAID utilizando duplexing

Outro risco importante a ser considerado é o fato de um servidor (*hardware*) ter um tempo de vida útil limitado assim como peças de reposição para este tornam-se menos prováveis de ser adquiridas, trazendo à tona o problema citado anteriormente em manterem-se versões compatíveis de binários e bibliotecas (e aí deve ser considerado os *drivers* dos novos periféricos substituídos) já que novos *hardwares* possuirão novas versões de drivers e aplicações (assim como sistemas operacionais) suportadas.

Estes problemas supracitados são contornados (de certa forma) mudando a técnica de instalação para não utilizar discos locais, e sim discos (LUN's) em *Storages*, já que nestes a segurança é aumentada pela própria questão da especialidade do equipamento e redundância que o mesmo oferece. Isso pode ser feito com técnicas de *Boot* por iSCSI ou *Boot* por *Fibre Channel*, onde a partida do sistema operacional se dá através de acessos a discos remotos (LUN's), realizados pelas controladoras HBA's.

Outra possibilidade bastante viável é a adoção de técnicas de virtualização onde um servidor de aplicação (seja qual for sua função) contará com um ambiente (BIOS, *drivers*, etc) igual independentemente do *hardware* que a hospede, uma vez que o hardware será tratado pelo Sistema de Virtualização (onde são alguns exemplos o VMWare, Virtual Server e Xen Server) minimizando assim o

impacto na manutenção dos servidores de aplicações ao longo de toda sua vida útil, já que contando com sistemas de virtualização as mesmas pouco sofrerão com alterações de *hardware* (que será tratada somente no nível de sistema hospedeiro).

Mesmo com os passos citados acima, ainda permanece a questão de o *Storage* ser o ponto único de falha. Para isso torna-se necessário abordar dois cenários: Replicação e Virtualização de Storage.

Replicação pode ser um cenário interessante tanto para servidores físicos quanto virtuais e estão relacionados diretamente com o RPO e RTO de cada aplicação suportada pelo servidor em questão. Quando se fala em replicação, está implícito que, no caso da ocorrência de um desastre, algum tempo de indisponibilidade haverá para tornar as aplicações operacionais novamente. Esse tempo (delta) está relacionado com as técnicas empregadas para replicação (síncrona ou assíncrona) considerando o RPO (já que alguma perda de dados ocorrerá) para cada área de negócio. Neste caso, quanto maior o RPO maior será o esforço para repetição de tarefas de reprocessamento e entradas nos sistemas (como lançamentos de pagamentos, notas fiscais, baixas em estoques, etc.). (WESLEY, 2005)

A opção de replicação basicamente baseia-se em soluções de *hardware* ou *software* que contando com dois ou mais *storages* copiam periodicamente uma LUN (ou MetaLUN) de um *storage* A para outro *storage* B. Essa cópia periódica (replicação) pode ser síncrona, ou seja, a cada gravação no *storage* A a mesma informação é também gravada no *storage* B, ou, assíncrona, ou seja, após um delta de tempo, todas operações de gravação (que ficam armazenadas em uma área de controle de transações) executadas no *storage* A são executadas no *storage* B, mantendo assim o conteúdo entre os dois storages bastante similar. Na Figura 10 abaixo, há um diagrama representando graficamente a replicação entre *storages* diferentes.

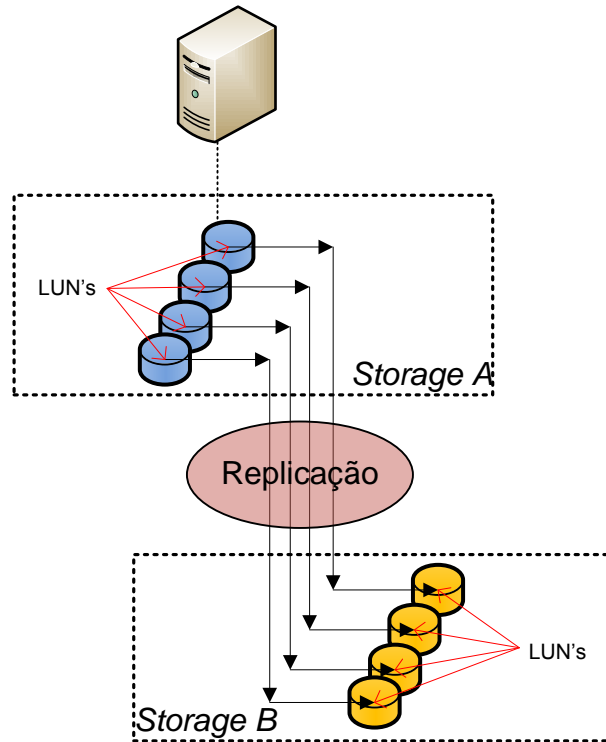


Figura 10: replicação de LUNs entre *storages*

Com o diagrama acima, fica mais evidenciado o que fora citado anteriormente sobre o tempo necessário para disponibilizar um servidor com acesso ao *storage B*, caso o *storage A* tenha sofrido alguma indisponibilidade: a reconfiguração do acesso do servidor às LUN's do outro *storage (B)* precisará ser refeita (*zoning*). Isso, dependendo do número de LUNs existentes pode levar vários minutos, cruciais para muitas aplicações e áreas de negócios.

Um cenário para esta solução é o apresentado na Figura 11, onde os discos (LUNs) utilizados pelo servidor, fornecidos pelo *storage A*, são replicados para o *storage B*, com uma solução por *software*, que pode rodar no mesmo servidor ou em um servidor a parte somente com esta função.

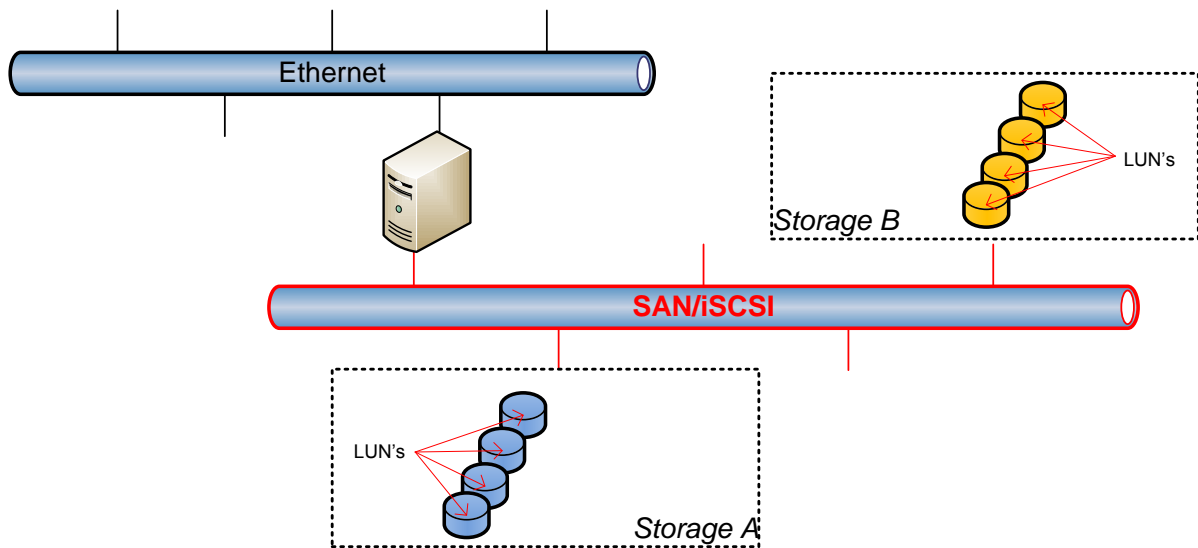


Figura 11: replicação de dados por software (dados)

Virtualização de *storage* é a técnica existente atualmente que pode ser utilizada para minimizar ou extinguir o tempo gasto em reconfiguração dos *zonnings* necessária para fazer um servidor acessar outro storage que possui as LUNs replicadas a partir de um storage de produção (*storage A* como mostrado anteriormente). Com isso aplicado, mesmo que um storage deixe de funcionar, os dados fornecidos ao servidor continuarão disponíveis, já que eles não estão armazenados unicamente em um storage e sim, em mais de um. Neste cenário o sistema operacional não acessa fisicamente os discos (LUNs) e sim de uma camada de software que provê acesso virtual aos discos (LUNs ou Meta LUNs), conforme mostrado na Figura 12, onde esta deverá também ser redundante conforme demonstrado na Figura 13.(WESLEY, 2005)

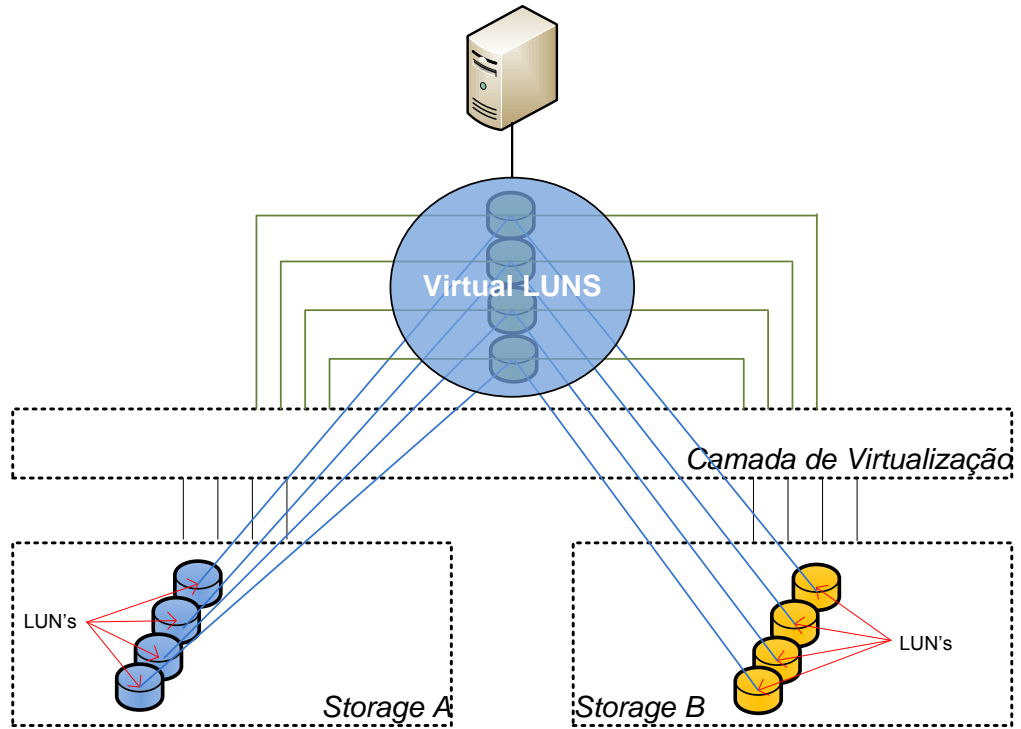


Figura 12: virtualização de *storage*

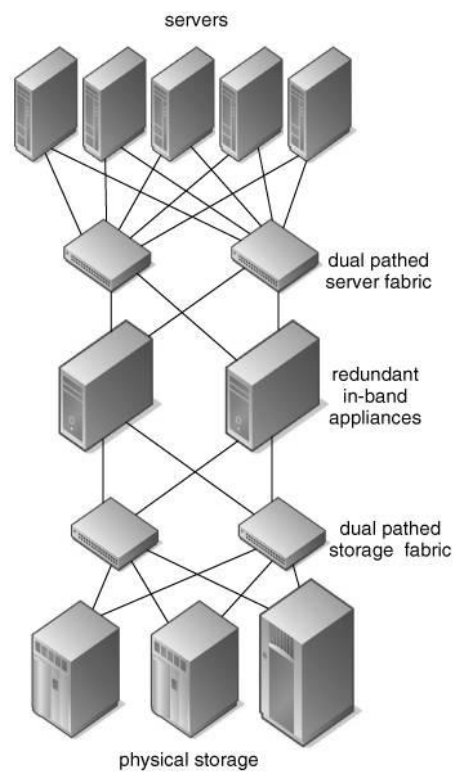


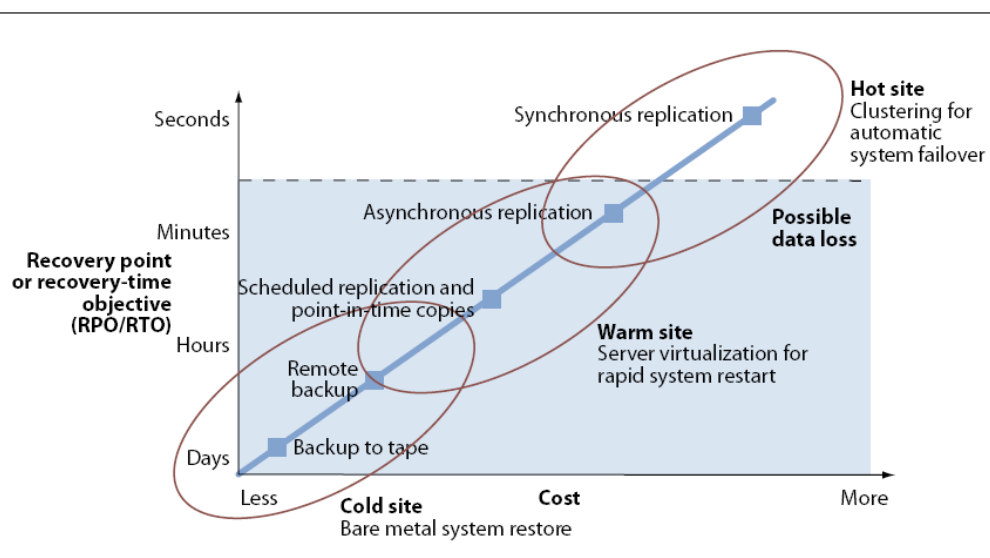
Figura 13: provimento de alta disponibilidade para camada de virtualização de *storage* (*redundant in-band appliances*)

Com essa solução, mesmo que o *Storage A* venha a ficar indisponível, seja qualquer o motivo, os dados continuarão acessíveis ao servidor a partir da camada de virtualização que acessa os dados necessários do *storage B*.

Com estes dois cenários expostos temos soluções aplicáveis a diferentes níveis de RPO e RTO (DOLEWSKI, 2008), para diferentes níveis de investimento, onde o que balizará o investimento será o custo agregado à solução associado ao retorno financeiro suportado pelas aplicações disponibilizadas pela infra-estrutura envolvida em cada solução. Desta forma, aplicações que não são tão críticas ao negócio (visto o retorno financeiro que provêem) podem ter sua disponibilidade de *storage* mantida por uma solução menos onerosa do que a solução adotada às aplicações mais críticas (visto também o retorno financeiro).

#### 4 APLICAÇÃO DE ALTA DISPONIBILIDADE E TESTES PROPOSTOS

De acordo com o contexto apresentado e analisando as requisições existentes no mercado de minimizar investimentos maximizando disponibilidade, não se pode deixar de realizar primeiramente uma análise de cada aplicação, associado a cada área de negócios que as utiliza a fim de criar uma matriz em que seja explicitada a real necessidade de alta disponibilidade, indicando em qual nível esta deva ser provida. Conforme a Figura 14 deve ser considerada em que tempo (RTO) e que possíveis perdas de informações (RPO) essas aplicações podem suportar (assim como suas áreas usuárias). (BALOURAS, 2008)



Source: Forrester Research, Inc.

Figura 14: Matriz de recuperação contínua

Esse gráfico evidencia que quanto menos informações são perdidas e onde a recuperação é mais rápida (ou o *down-time* inexistente) mais cara é a solução. Baseado nisso, a primeira coisa a ser feita é determinar que aplicações devam utilizar qual técnica de recuperação ou alta disponibilidade. Como o tema deste trabalho está focado em alta disponibilidade, será abordada apenas a situação mais crítica, ou seja: RPO e RTO próximos a zero. (DOLEWSKI, 2008)

Como primeiro cenário será proposto um ambiente para a camada de *storage*, de baixo custo, que proverá redundância através de replicação de dados. Para este primeiro cenário a proposta é utilizar como plataforma de *hardware* servidores de baixo custo ou ainda estações de trabalho, porém com bom desempenho, principalmente no acesso a discos (atualmente existem soluções com discos de 15000 rpms). Um dimensionamento de memória em torno de 4 Gb de RAM ou mais e uma placa de rede compatível com a solução de rede existente na infra-estrutura deve ser considerada a fim de maximizar o acesso aos recursos do *hardware* evitando gargalos. (SCHIMIDT, 2006)

Para montar este primeiro cenário deverão ser dispostos estes dois servidores com um sistema operacional que disponha de soluções de replicação de dados, assim como uma solução de compartilhamento de arquivos (CIFS, NFS, etc.) ou *software* que disponibilize um *target* iSCSI a fim de prover aos servidores de aplicação acesso (compartilhado ou não) ao sistema de arquivos. Os servidores de aplicação poderão estar dispostos sob forma de *clusters* ou de balanceamento de carga (onde o tráfego pode ser balanceado entre mais de um servidor com, por exemplo, *Round Robin DNS* no caso de balanceamento de carga). (GOUVEIA, 2005)

Para prover um melhor tempo de recuperação para servidores de aplicações nos casos de problemas de *hardware* destes a unidade de armazenamento que suportará o sistema operacional (além é claro da unidade de armazenamento que proverá acesso aos dados e a própria aplicação) deverá estar armazenada nesta solução de *storage* e no caso de falha, uma simples troca de *hardware* e reconfiguração do iSCSI *initiator* tornará operacional novamente a aplicação e a disponibilidade dos serviços providos por este servidor. A Figura 15, abaixo, mostra através de um diagrama a arquitetura de *hardware* e a topologia a ser adotada nesta solução de replicação. A camada de Dados abaixo dos servidores

de aplicação são na verdade *targets* iSCSI ou compartilhamentos NFS ou CIFS que são providos aos servidores pela camada de servidores de dados (camada inferior).

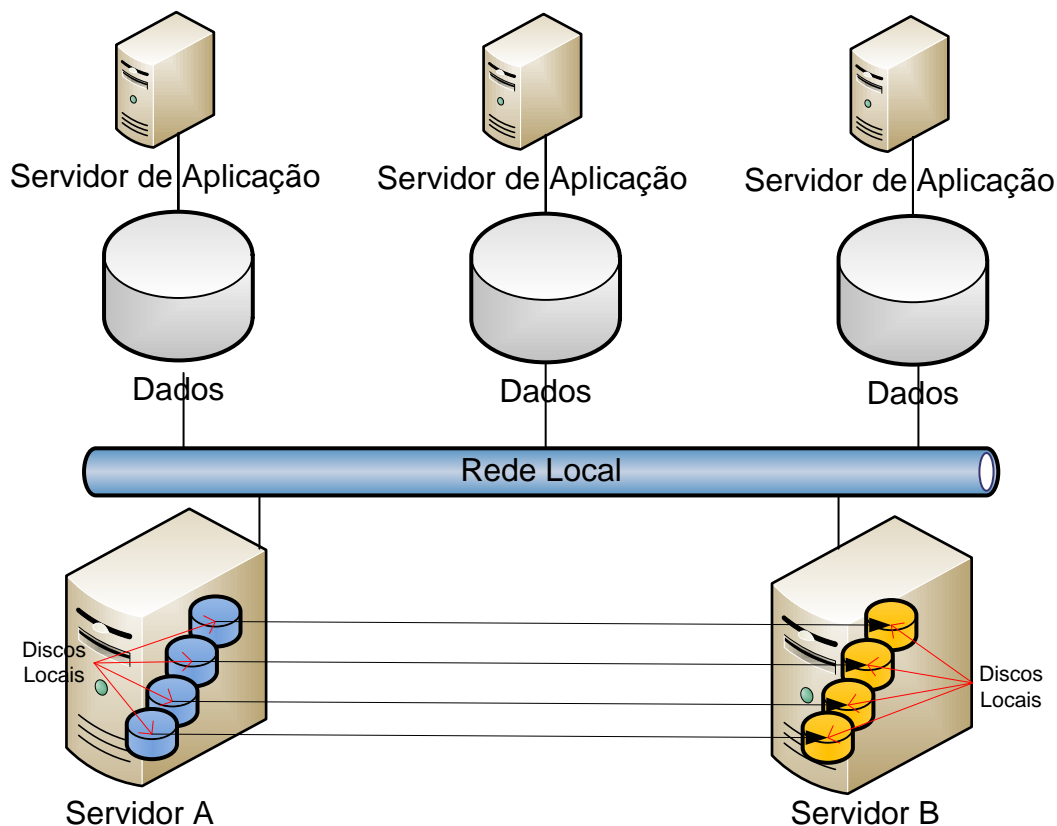


Figura 15: solução de replicação de dados para vários servidores

Na ocorrência de problemas que impeçam o correto funcionamento de um dos servidores de dados (servidores A ou B), que possuem os dados replicados entre si, a única manutenção necessária para disponibilizá-los novamente aos servidores de aplicação seria a reconfiguração dos *shares* (no caso de utilizar-se CIFS), *exports* (no caso de NFS) ou *initiators* iSCSI (no caso de *targets* iSCSI).

Além disso, adotar uma solução de *backup* que tem como origem os servidores de dados tende a maximizar a utilização dos recursos de *backup* e minimizar as janelas para o mesmo (tempo necessário para o *backup*).

Como segundo cenário, a fim de atender ininterruptamente o fornecimento de dados a servidores a proposta é a virtualização da camada de acesso a dados (*storage*), estendendo a topologia anterior com uma camada a mais de *software* e *hardware*. Desta maneira os dados não estarão presentes unicamente em um *storage* (servidor ou ainda em um *desktop*) e sim distribuídos em mais de um *hardware*. (WESLEY, 2005)

A solução de virtualização está fundamentada em prover através de serviços de cluster acesso aos *targets* iSCSI ou ainda a compartilhamentos CIFS ou NFS, dependendo da necessidade, porém a solução baseada em iSCSI tende a atender melhor qualquer cenário, visto a melhoria a nível de tratamento do protocolo SCSI dentro do protocolo de transporte, já que o tráfego iSCSI está baseado em blocos de acesso a disco e não em arquivos isolados. Na Figura 16 está um diagrama que demonstra como uma infra-estrutura para este fim deverá ser configurada. (WESLEY, 2005)

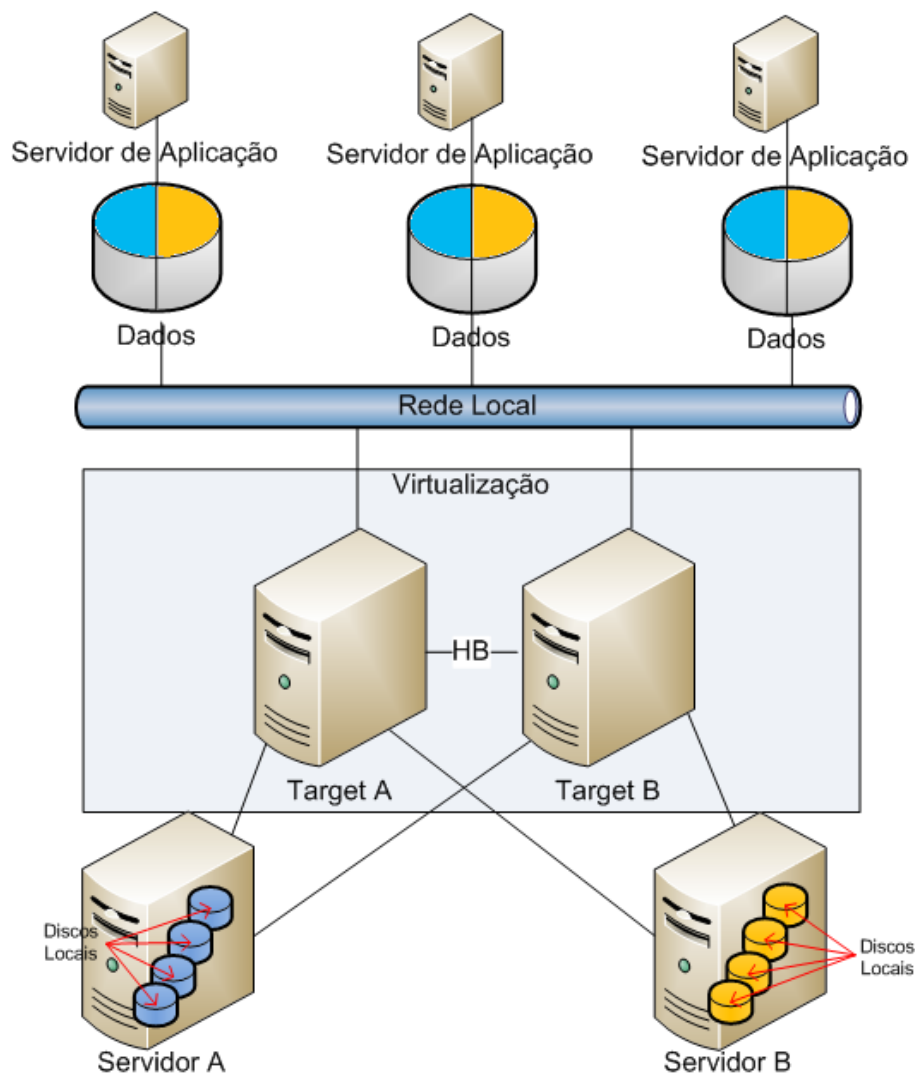


Figura 16: cluster para virtualização de *storage*

Deve-se considerar que, quanto maior a necessidade do negócio frente à tecnologia, melhores *hardwares* deverão ser considerados para suportar esta proposta, a fim de garantir os níveis de disponibilidades exigidos pelo negócio. Para

esta análise, indicadores como MTBF (*mean time between failures*) e MTTR (*mean time to repair*) de cada componente devem ser fortemente considerados. (OGGERINO, 2001)

## 5 CONCLUSÕES

Com os estudos realizados, bem como os cenários propostos, pode-se concluir que é viável atualmente disponibilizar serviços de TI com alta disponibilidade a empresas que não possuem recursos suficientes para a aquisição de plataformas especializadas para este fim, onde somente os grandes *players* de mercado possuem tais ofertas.

Conclui-se também que a utilização do protocolo iSCSI trouxe muitos benefícios para este tipo de solução, já que permite a construção de soluções de *storages* a partir de servidores ou mesmo estações a um custo bem menor. Obviamente a performance obtida com tais soluções não podem ser comparadas a soluções construídas especificamente para este fim, no entanto, no quesito disponibilidade, elas podem ser em muitos casos empregadas.

## 6 TRABALHO DE CONCLUSÃO DE CURSO - II

Para o Trabalho de Conclusão de Curso II a ser apresentado no próximo semestre, espera-se construir duas soluções para a implementação dos cenários aqui propostos. O primeiro, voltado a replicação de *storages*, onde utilizar-se-á dois servidores assumindo o papel de *storage* (virtuais) onde os dados entre os mesmos estarão sendo sincronizados, onde um terceiro servidor fará uso dos mesmos. O segundo cenário, voltado a virtualização de *storage*, deverá contar com dois servidores assumindo o papel de *storage* (virtuais) onde um terceiro assumirá o papel de *appliance* a fim de entregar recursos de *storage* a um quarto e quinto servidores que farão uso destes recursos.

Estes cenários propostos visarão provar os conceitos aqui implementados para validar em uma escala pequena os conceitos e técnicas apresentadas, sem contudo se preocupar com a performance que, esperadamente, não será a mais adequada para cenários de produção. Com isso proposto, as

soluções disponíveis para maximizar a disponibilidade de aplicações para pequenas e médias empresas estarão bastante evidenciadas.

## 7 REFERÊNCIAS

BALAOURAS, Stephanie. **Building The Business Case For Disaster Recovery Spending, Forrester** – 03/04/2008. 18p.

KLOTH, Axel K. **Advanced Router Architectures**: Arquiteturas avançadas para Roteadores. ISBN: 978-0849335501, 2005. 240p.

CLARK, Tom. **Storage Virtualization: Technologies for Simplifying Data Storage and Management**: Tecnologias para simplificar o Gerenciamento e o Armazenamento de Dados. ISBN: 978-0321262516, 2005. 264p.

NORTHCUTT, Stephen. **Inside Perimeter Security (2nd Edition)**: por dentro da segurança de perímetro. ISBN: 978-0672327377, 2005. 768p.

DOLEWSKI, Richard. **System i Disaster Recovery Planning**. Auxilia no mapeamento de informações para apoio a decisão de Planejamento de Recuperação em Desastres. ISBN: 978-1583470671, 2008. 350p.

SCHMIDT, Klaus. **High Availability and Disaster Recovery: Concepts, Design, Implementation**. Alta Disponibilidade e Recuperação de Desastres: conceitos, Design e implementação. ISBN: 978-3540244608, 2006. 410p.

GOUVEIA, José. **Redes de Computadores**. Locais e Wireless. ISBN: 9789727224739, 2005. 312p.

WESLEY, Adilson. **Storage Virtualization: Technologies for Simplifying Data Storage and Management**. ISBN: 0321262514, 2005. 264p.

OGGERINO, Chris. **High Availability Network Fundamentals**. ISBN: 9781587130175, 2001. 250p.