

Ferramentas *Open Source* e Redundância de *Backups*

Diego Dummer Vaz¹, Msc. Eduardo Maroñas Monks¹

¹Tecnologia em Redes de Computadores Faculdade de Tecnologia Senac
Pelotas – RS – Brazil

diegodvaz@gmail.com, emmonks@gmail.com

Resumo. *Este projeto analisa duas ferramentas que gerenciam servidores de backups, são elas Bacula e BackupPC. Foram analisadas características, funcionalidades e particularidades destas ferramentas para gerenciamento de backups, gerando um comparativo entre elas. Foi aplicada uma solução de redundância, visando maior disponibilidade e integridade dos dados.*

Abstract. *This project examines two tools that manage backup servers, are they Bacula and BackupPC. Features, functionality and characteristics of these tools for managing backups were analyzed, generating a comparison between them. A redundancy solution, aiming higher availability and data integrity was implemented.*

1. Introdução

Apesar da importância de manter-se cópias de segurança dos dados, nem sempre é uma atividade de fácil resolução a escolha da ferramenta que será utilizada.

Tão importante quanto a escolha da ferramenta que gerenciará o servidor, é implementar um sistema de redundância dos *backups* (*backup* do *backup*), pois o local de armazenamento dos *backups* também está sujeito a falhas. Todas as possibilidades tem de ser levadas em consideração, pois a função dos *backups* é ser uma alternativa segura contra todos os tipos de sinistros.

Como exemplo de situação em que seria essencial um planejamento completo de *backups*, pode-se citar a queda das torres do complexo *World Trade Center*. Segundo [Ribeiro 2010], cada torre utilizava a outra para armazenar seus *backups*. A queda das duas torres causou a perda dos dois pontos de *backup*. Uma política bem aplicada de *backups off-site* teria salvado os dados.

2. Backups

Segundo [Adrenaline 2013], na década de 50, quando pesquisadores desenvolviam os primeiros computadores digitais da história, a forma predominante de armazenar arquivos consistia nos cartões perfurados. Apesar de ter mais de um século de idade, o cartão perfurado foi utilizado nas eleições presidenciais americanas, no ano 2000. Na Figura 1 pode-se analisar uma máquina de perfuração do papel, utilizada na década de 50.

No ano de 1956, a IBM lançou os primeiros discos com 5MB (*megabytes*) de capacidade de armazenamento e fisicamente possuíam o tamanho de dois refrigeradores.

Na década de 60, surgiram as fitas magnéticas, muito mais rápidas e eficientes, pois um rolo de fita poderia guardar informações equivalentes a dez mil cartões perfurados. Fitas são utilizadas até hoje para armazenamento de grandes volumes de dados, existem modelos capazes de armazenar mais de 3TB (*terabytes*) de dados sem compressão. Segundo [HP 2013], atualmente a utilização de fitas é tão fácil, flexível, portátil e intuitiva quanto usar outras mídias removíveis e compartilháveis, como uma unidade USB.

Os discos rígidos foram sendo aprimorados ao longo das décadas seguintes, em 1982 a empresa Hitachi apresentou ao mercado um disco rígido com capacidade de armazenamento de até 1GB (*gigabytes*), sendo este o real ponto inicial das tecnologias de discos rígidos que existem atualmente. Ainda de acordo com [Adrenaline 2013], a partir do advento dos discos rígidos, várias outras formas de armazenamento foram desenvolvidas se valendo do mesmo princípio de tecnologia, um exemplo é o desenvolvimento de *storages*, que são equipamentos gerenciáveis com um ou mais discos rígidos.



Figura 1. Cartões perfurados.

3. Sistemas de *Backups* Analisados

Nesta seção observa-se a estrutura das plataformas de *backups* que são objeto de estudo deste artigo. Foi analisada a funcionalidade dos principais arquivos de configuração de ambas as plataformas, além de uma breve descrição das mesmas.

3.1. Bacula

Bacula é uma plataforma, *opensource* (código-aberto), para gerenciamento de *backups*. Segundo [Bacula 2013], esta ferramenta consiste em um conjunto de programas que permitem efetuar *backups* e restaurações de dados contidos em computadores com variados sistemas operacionais. O Bacula é relativamente fácil de usar e eficiente, oferece diversos recursos avançados de gerenciamento e armazenamento, que tornam mais ágil a tarefa de encontrar e recuperar arquivos perdidos ou danificados.

<code>bacula-dir-agendamento.conf</code>	Arquivo onde são criados os agendamentos;
<code>bacula-dir-clients-and-jobs.conf</code>	Arquivo onde são associados clientes e tarefas de backups;
<code>bacula-dir.conf</code>	Principal arquivo de configuração do Bacula;
<code>bacula-dir-filesets.conf</code>	Diretórios que serão backupeados em cada host;
<code>bacula-fd.conf</code>	Configurações do backup local;
<code>bacula-sd.conf</code>	Configurações de dispositivos de armazenamento;

Figura 2. Arquivos de configuração do Bacula.

Na Figura 2 observa-se as funções dos arquivos envolvidos na configuração dos servidores Bacula utilizados. O Bacula tem sua configuração quase que totalmente personalizável, sendo assim, nem todos os arquivos aqui demonstrados existem por padrão. Conforme foi definido no arquivo `"/etc/bacula/bacula-sd.conf"`, os *backups* estão sendo salvos no diretório `"/backups"`.

O Bacula pode utilizar os bancos de dados PostgreSQL [Group 2013] ou MySQL [Corporation 2013], para a execução deste projeto foi escolhido o PostgreSQL. O catálogo do Bacula é salvo em base de dados PostgreSQL que fica no diretório `"/var/lib/postgresql/8.4/main/"`.

3.2. BackupPC

Segundo [Barratt 2013], BackupPC é um sistema de alta performance. Este sistema efetua *backups* de sistemas UNIX, Linux, Windows e MacOSX. BackupPC é altamente configurável, de fácil manutenção e administração.

De acordo com [Morimoto 2013], dado o custo cada vez menor dos *HDs* (*Hard Disks*), tanto domésticos quanto corporativos, os sistemas RAID (*Redudant Array of Independet Disks*) tornaram-se cada vez mais viáveis, do ponto de vista financeiro. Ficou mais barato desenvolver soluções de *backups* distribuídas entre vários discos e manter *backups* em discos locais no servidor ou mesmo em *storages* com várias cópias segmentas por períodos (dias, semanas, meses) e a plataforma BackupPC pode ser configurada para trabalhar em todos estes cenários.

Na Figura 3 pode-se visualizar as funções desempenhadas pelos principais arquivos de configuração do BackupPC.

<code>192.168.1.101.pl</code>	Arquivo atual de configuração de cliente de backup;
<code>192.168.1.101.pl.old</code>	Arquivo anterior de configuração de cliente de backup;
<code>192.168.140.141.pl</code>	
<code>192.168.140.141.pl.old</code>	
<code>apache.conf</code>	Arquivo de configuração do Apache;
<code>backuppc</code>	
<code>config.pl</code>	Arquivo de configurações gerais do BackupPC;
<code>config.pl.old</code>	Arquivo anterior de configurações gerais do BackupPC;
<code>hosts</code>	Arquivo onde ficam listados os hosts cadastrados;
<code>hosts.old</code>	Arquivo anterior de hosts cadastrados;
<code>htgroup</code>	
<code>htpasswd</code>	Arquivo de autenticação do Apache;
<code>localhost.pl</code>	Arquivo atual de configuração de backup local;
<code>localhost.pl.old</code>	Arquivo anterior de configuração de backup local;

Figura 3. Arquivos de configuração do BackupPC.

Nos arquivos `"nomedohost.pl"` somente constam as configurações que foram personalizadas pelo administrador para este *host* especificamente, caso o *host* seja adicionado ao BackupPC e nenhuma configuração for modificada, as configurações utilizadas serão do arquivo `"config.pl"`. Os *backups* são salvos no diretório `"/var/lib/backuppc"`.

4. *Clusters*

Com a difusão do uso da Internet, ao longo dos anos aumentou exponencialmente a necessidade de obter-se informação cada vez mais rápido e igualmente a necessidade desta informação estar disponível sempre que necessária. De acordo com [Trigo 2007], seguindo esta tendência foram desenvolvidas ferramentas que possibilitam agrupar diversos computadores ou dispositivos em pró de uma atividade, este conceito denomina-se *cluster*.

Apesar de *clusters* sempre fazerem referência ao fato de disponibilizar equipamentos compatíveis em torno de um mesmo objetivo, há algumas classificações de *clusters*, cada uma delas voltada a formas diferentes de disponibilizar serviços com melhor qualidade.

Os *clusters* podem ser divididos basicamente nos sub-itens a seguir:

4.1. *Cluster de Alto Desempenho*

O *cluster* de processamento paralelo tem por objetivo dividir o processamento de determinada atividade entre os membros do *cluster*, visando menor tempo total de processamento e mais rápida resolução da tarefa.

4.2. *Balanceamento de Carga*

Este tipo de *cluster* visa a análise de carga em cada membro do *cluster*, sempre que é recebida uma nova tarefa. O membro com menor carga recebe a tarefa, dessa forma evitando que algum membro do *cluster* fique sobrecarregado, a divisão de tarefas promovida pelo *cluster* visa principalmente a redução do tempo de resposta para os serviços disponibilizados.

4.3. *Alta Disponibilidade*

Os *clusters* de alta disponibilidade visam prover acesso a determinado serviço na totalidade do tempo. Os serviços são disponibilizados igualmente em todos os membros do *cluster*, promovendo redundância do serviço, dessa forma quando o nó principal estiver indisponível o nó secundário assume a tarefa sem que o usuário perceba a troca ocorrida.

5. *Tipos de Backups*

Há *backups* de tipos diferentes, para aplicações diferentes, pois cada tipo de *backup* se encaixa melhor a cada ambiente ou necessidade. A análise do tipo de *backup* a ser utilizado deve ser efetuada na etapa de planejamento das rotinas de *backup*, mas para isso deve-se conhecer conceitualmente os tipos de *backup* existentes.

5.1. *Backup Full*

Neste tipo de *backup*, é efetuada a cópia de todos os arquivos, sem exceções. Geralmente utiliza-se este tipo de *backup* na primeira vez em que a tarefa é executada, para servir de referência aos outros tipos de *backups* que venham a ser utilizados posteriormente. Também comumente é utilizada após o término do último dia de expediente da semana, em casos corporativos, visando fazer proveito da janela do fim de semana visto que o volume de dados demanda mais tempo para ser executada.

5.2. Diferencial

Neste tipo de *backup*, é efetuada a cópia somente dos arquivos que foram modificados ou criados desde a execução do último *backup full*.

5.3. Incremental

Nesta modalidade somente são copiados os arquivos que foram modificados ou criados após o último *backup* diferencial. É o método mais rápido de efetuar *backups*, porém, o *restore* pode ser uma tarefa bem complexa se o volume de dados for grande e houver um extenso histórico de *backups*.

5.4. Cópia

Pode-se dizer que é a redundância de um volume de *backups*, muito utilizado na modalidade "Off-Site" (dados retirados do local físico onde os dados são originados, normalmente são armazenados em nuvem).

5.5. Migração

Quando diagnosticado problema de leitura ou escrita, utiliza-se esta técnica para copiar os dados para outro dispositivo e efetuar o descarte do dispositivo original.

6. Melhores práticas com *Backups*

De acordo com [de Faria 2010], as melhores práticas ao lidar com *backups* seriam:

- Estratégia de *backup* condizente com a natureza dos dados armazenados;
- Testes de *restore* periódicos;
- Operador de fitas, administrador de *backup* e administrador de *restore* devem ser sempre pessoas diferentes;
- Documentação atualizada constantemente, inclusive com uso de banco de soluções;
- Alta disponibilidade;
- Planejamento e simulações de "disaster recovery";
- Existência de procedimento padrão para descarte de mídias (Por exemplo: trituração, incineração, dentre outras);
- Adequação à norma **ISO 27002**[Directory 2013].

6.1. Esquema GFS - *Grandfather Father Son Backup*

De acordo com [de Faria 2010], **GFS** é a mais popular estratégia de *backups*. Esta estratégia, em princípio, foi criada com objetivo de ser utilizada com fitas, no entanto com o passar do tempo percebeu-se a possibilidade de aplicação em qualquer sistema de *backup* hierárquico, pois consiste em ciclos diários, semanais e mensais.

Ainda de acordo com [de Faria 2010], no esquema **GFS** os *backups* são divididos em três grupos (diário, semanal e mensal), os *backups* "filho" ou diários são rotacionados diariamente, os *backups* "pai" ou semanais são rotacionados semanalmente, assim como o *backup* "avô" ou mensal é rotacionado mensalmente. O rotacionamento sempre acontece entre *backups* semelhantes, ou seja, diário com diário, semanal com semanal e mensal com mensal.

7. Testes Efetuados

Utilizou-se oito VMs (*Virtual Machines*), quatro no cenário Bacula e quatro no cenário BackupPC, todas com sistema operacional Debian Linux Squeeze, formando dois *clusters* de alta disponibilidade com dois nós, cada cenário com dois *clusters*. Na *Tabela 1*, pode-se visualizar detalhadamente o *hardware* das VMs utilizadas como servidores de *backups* (Bacula e BackupPC). Na *Tabela 2* analisa-se o *hardware* utilizado nos servidores NFS, onde são armazenados os *backups* e dados dos servidores Bacula e BackupPC.

Tabela 1. Hardware dos servidores de backups.

Servidor de Backup	
Processador:	1 Core
Disco Rígido:	20GB
Memória:	1GB
Sistema Operacional:	Debian Linux Squeeze 6.0.7 i386
Interface de Rede 1:	10/100 Mb/s (Modo Bridge)
Interface de Rede 2:	10/100 Mb/s (Modo Host Only)

Tabela 2. Hardware dos servidores NFS.

Servidor NFS	
Processador:	1 Core
Disco Rígido 1:	20GB
Disco Rígido 2:	20GB
Memória:	512MB
Sistema Operacional:	Debian Linux Squeeze 6.0.7 i386
Interface de Rede 1:	10/100 Mb/s (Modo Bridge)
Interface de Rede 2:	10/100 Mb/s (Modo Host Only)

7.1. Cenários analisados

Efetuiu-se testes em um ambiente de alta disponibilidade, utilizando-se de técnicas para prover redundância não só dos *backups* efetuados, mas também das próprias plataformas de gerenciamento das tarefas de *backups*.

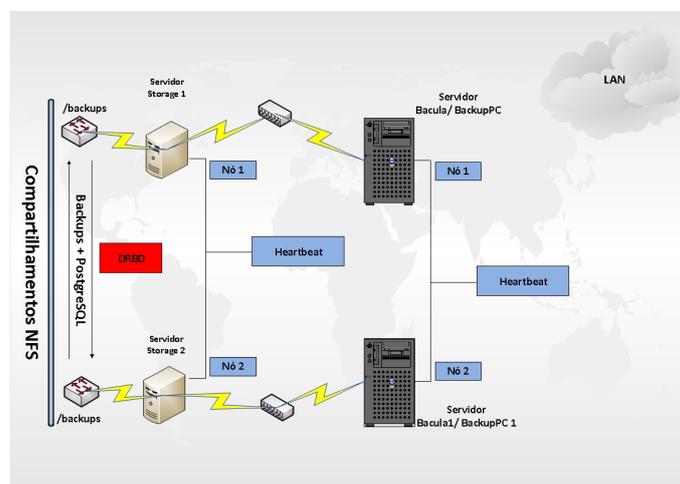


Figura 4. Redundância de servidores e backups.

As duas plataformas de *backup* analisadas Bacula e BackupPC foram submetidas a configurações e adaptações, a fim de trabalharem de forma redundante nos cenários adotados. Neste cenário é analisado ambiente com redundância total de servidores de *backups* e de dados coletados pelas tarefas de *backups*.

Na Figura 4 percebe-se a comunicação entre servidores NFS, que efetuam o armazenamento dos *backups*, e dos servidores de gerenciamento do serviço de *backups*. Segundo [Ferreira 2008], através da interface virtual gerada pelo Heartbeat [Heartbeat 2013], onde constantemente os nós do *cluster* comunicam-se, buscando a disponibilidade do nó secundário na ausência do nó principal. Os dados contidos nos compartilhamentos NFS são sincronizados em tempo real através do **DRBD** [DRBD 2013], gerando redundância dos *backups* armazenados nestes compartilhamentos.

7.1.1. Bacula - Alta Disponibilidade

Neste cenário foram utilizadas as ferramentas Bacula, Heartbeat e DRBD para prover alta disponibilidade, praticamente total, na tarefa de efetuar *backups* e *restores* via Bacula. Ao iniciar-se uma janela de *backups*, o *host* negocia a conexão com a interface virtual criada pelo Heartbeat do servidor Bacula. No servidor Bacula, durante o processo de *boot*, são montados os diretórios `"/etc/bacula"` e `"/var/lib/postgresql"`, que correspondem ao diretório dos arquivos de configuração do Bacula e o diretório onde fica a base PostgreSQL que contém o catálogo do Bacula, nesta ordem respectivamente. Na Figura 5 pode-se analisar o cenário adotado na utilização do "Bacula redundante com armazenamento distribuído".

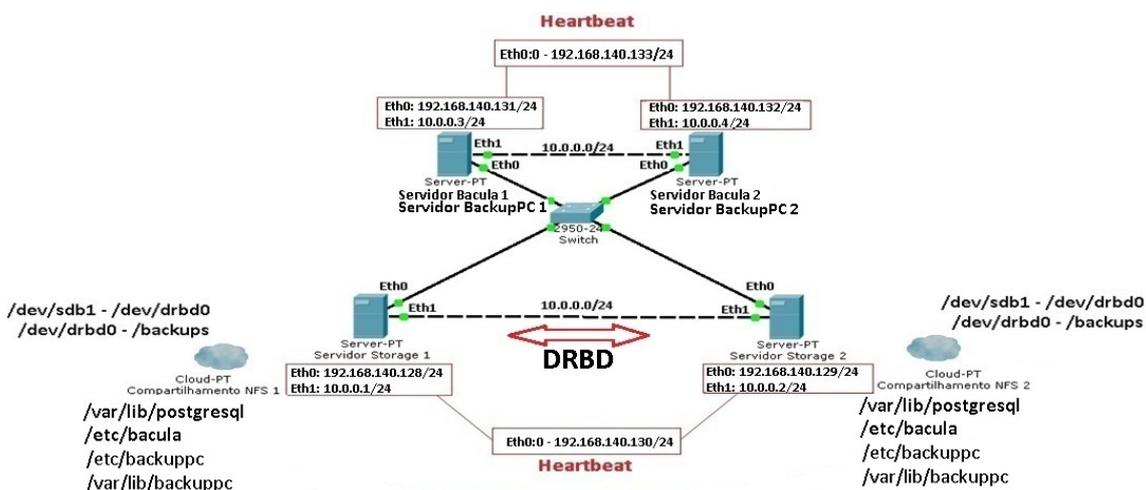


Figura 5. Cenário Bacula de alta disponibilidade.

O Heartbeat monitora constantemente a disponibilidade do nó primário (servidor Bacula 1), caso esse servidor tenha algum problema que cause sua indisponibilidade, o nó secundário (servidor Bacula 2) assume de forma transparente ao usuário do Bacula. Isso é possível devido ao sincronismo, em tempo real, efetuado pelo DRBD nos diretórios do Bacula e PostgreSQL que estão hospedados nos compartilhamentos NFS. Quando o nó secundário assume, são montados os compartilhamentos NFS, iniciado o PostgreSQL

e os serviços do Bacula (**bacula-dir**, **bacula-sd** e **bacula-fd**, nesta respectiva ordem). Como os dois servidores Bacula utilizam a mesma base de dados, a troca de nós passa a ser praticamente transparente. Somente ocorre interrupção do serviço de *backup* se no momento da queda do nó primário, alguma tarefa de *backup* estiver sendo executada. A tarefa corrente é interrompida, mas será dado seguimento a partir da próxima tarefa, conforme agendamento previsto na janela de *backups*.

7.1.2. BackupPC - Alta Disponibilidade

Neste cenário foi utilizada a plataforma BackupPC como servidor de *backups*. Assim como no item 7.1.1, foram utilizadas as ferramentas Heartbeat e DRBD para prover alta disponibilidade, praticamente total, na tarefa de efetuar *backups* e *restores*. Ao iniciar-se uma janela de *backups*, o *host* negocia a conexão com a interface virtual criada pelo Heartbeat do servidor BackupPC. No servidor BackupPC, durante o processo de *boot*, são montados os diretórios **"/etc/backuppe"** e **"/var/lib/backuppe"**, que correspondem ao diretório dos arquivos de configuração do BackupPC e o diretório onde ficam os dados copiados, nesta ordem respectivamente. Na Figura 6 pode-se analisar o cenário adotado na utilização do "BackupPC redundante com armazenamento distribuído".

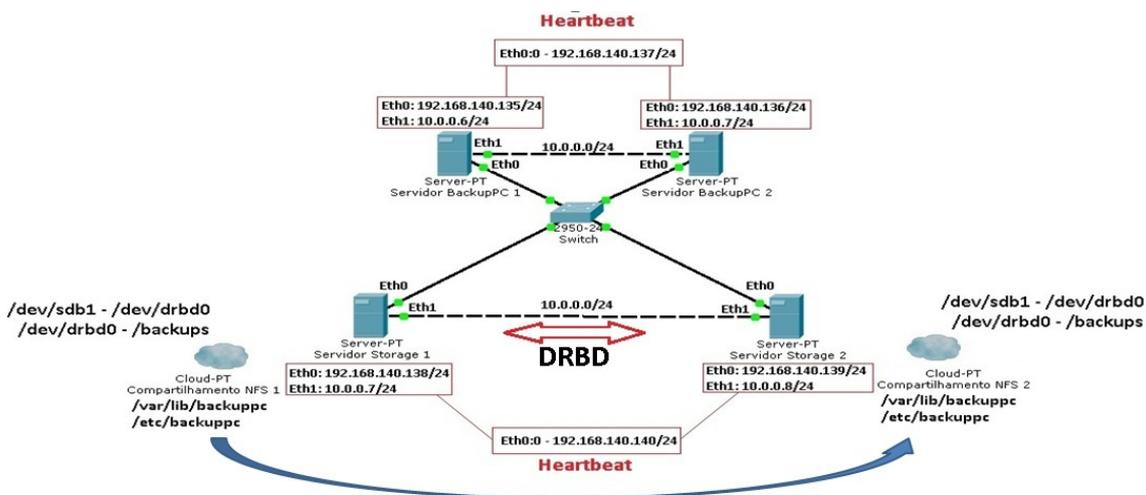


Figura 6. Cenário BackupPC de alta disponibilidade.

O Heartbeat monitora constantemente a disponibilidade do nó primário (servidor BackupPC 1), caso esse servidor tenha algum problema que cause sua indisponibilidade, o nó secundário (servidor BackupPC 2) assume de forma transparente ao usuário. Isso é possível devido ao sincronismo, em tempo real, efetuado pelo DRBD nos diretórios do BackupPC que estão hospedados nos compartilhamentos NFS. Quando o nó secundário assume, são montados os compartilhamentos NFS e o BackupPC. Como os dois servidores BackupPC utilizam a mesma base de dados, a troca de nós passa a ser praticamente transparente. Somente ocorre interrupção do serviço de *backup* se no momento da queda do nó primário, alguma tarefa de *backup* estiver sendo executada. A tarefa corrente é interrompida, mas será dado seguimento a partir da próxima tarefa, conforme agendamento previsto na janela de *backups*.

8. Comparativo entre Bacula e BackupPC

Na Tabela 3, serão analisadas características, funcionalidades e particularidades das ferramentas Bacula e BackupPC, gerando um comparativo entre elas.

Tabela 3. Comparativo de recursos entre Bacula e BackupPC.

Comparativo de funcionalidades		
Recursos	Bacula	BackupPC
Estrutura Cliente/Servidor	X	X
Estrutura Modular Independente	X	
Versão Licenciada sob GPL	X	X
Versão Proprietária	X	
Catálogo de Backups	X	X
Agendamento de Job	X	X
Backup/Restore	X	X
Multiplataforma	X	X
Suporte a Variados Tipos de Mídias	X	X
Documentação Web	X	X
Comunidade Atuante	X	X
Gerenciamento via Browser	X	X
Gerenciamento via Terminal	X	
RunBeforeJob/RunAfterJob	X	X
Envio de E-mails	X	X
Desduplicação de Dados	X	
Backup Full	X	X
Backup Diferencial	X	
Backup Incremental	X	X

9. Futuros Trabalhos

No decorrer dos testes e estudos realizados, percebeu-se a necessidade da existência de monitoramento e emissão de alertas via *email*, que serão enviados para a pessoa responsável pela manutenção do serviço de cópia de segurança dos dados. A implementação desta melhoria no projeto servirá como objeto de estudo em trabalho futuro a ser desenvolvido.

10. Conclusões

Em virtude da análise das plataformas de *backup* e dos testes realizados, conclui-se que foram estudadas duas das mais conceituadas e completas ferramentas de gerenciamento de *backups* do mercado. Tanto Bacula quanto BackupPC possuem quase que a totalidade das funcionalidades necessárias a uma boa plataforma desta categoria, porém, nota-se que o Bacula é mais completo que o BackupPC. Pode-se citar alguns aspectos levados em consideração: Possui mais de uma interface *Web* de gerência, possui comunidade de desenvolvimento e colaboração bem mais ampla e difundida, possui livros publicados sobre o assunto e principalmente por arquivar o catálogo de tarefas em um banco de dados robusto como o PostgreSQL.

Nos testes executados, foi possível concluir com ampla certeza, que o benefício da utilização de *clusters* de alta disponibilidade na gerência de *backups* é imensurável, devido principalmente, ao fato de não perder toda a janela de *backups* em caso de indisponibilidade, mas sim somente tarefa a corrente, pois a tarefa seguinte prossegue normalmente conforme agendamento do catálogo.

Analisando a infraestrutura de redundância implementada em ambos os cenários testados, conclui-se que é essencial prover alta disponibilidade e armazenamento distribuído redundante de serviços com o grau de importância altíssimo como *backups*.

Referências

- Adrenaline (2013). O dia do backup - a história do procedimento e como fazer o seu. Disponível em <<http://adrenaline.uol.com.br/tecnologia/artigos/142/o-dia-do-backup—a-historia-do-procedimento-e-como-fazer-o-seu.html?pg=2>>. Acesso em: set 2013.
- Bacula (2013). Bacula the open source network backup solution. Disponível em <<http://www.bacula.org/en/>>. Acesso em: out 2013.
- Barratt, C. (2013). Backuppc. Disponível em <<http://backuppc.sourceforge.net/index.html>>. Acesso em: set 2013.
- Corporation, O. (2013). Mysql - the world's most popular open source database. Disponível em <<http://www.mysql.com/>>. Acesso em: nov 2013.
- de Faria, H. M. (2010). *Bacula - Ferramenta Livre de Backup*. Brasport.
- Directory, T. I. . (2013). Introduction to iso 27002. Disponível em <<http://www.27000.org/iso-27002.htm>>. Acesso em: nov 2013.
- DRBD (2013). Drbd software development for high availability clusters. Disponível em <<http://www.drbd.org/>>. Acesso em: set 2013.
- Ferreira, R. E. (2008). *Linux - Guia do Administrador do Sistema*. Novatec.
- Group, P. G. D. (2013). Postgresql - the world's most advanced open source database. Disponível em <<http://www.postgresql.org/>>. Acesso em: nov 2013.
- Heartbeat (2013). Heartbeat subsistema para linux de alta disponibilidade. Disponível em <<http://packages.debian.org/pt/squeeze/heartbeat>>. Acesso em: set 2013.
- HP (2013). Hp. Disponível em <<http://www8.hp.com/br/pt/products/storage-media/product-detail.html?oid=4153086!tab=features>>. Acesso em: nov 2013.
- Morimoto, C. E. (2013). Guiadohardware.com.br - tudo sobre os hds, flash e armazenamento. Disponível em <<http://www.hardware.com.br/guias/hds/>>. Acesso em: nov 2013.
- Ribeiro, G. (2010). Veezor. Disponível em <<http://www.veezor.com/sem-categoria/porque-solucao-de-backup-corporativo/>>. Acesso em: nov 2013.
- Trigo, C. H. (2007). *OpenLDAP - Uma Abordagem Integrada*. Novatec.