

LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES

FELIPE LEONAN LUPIA CARDOSO

**BACULA: APLICAÇÃO DE UM SISTEMA DE GESTÃO DE BACKUP LIVRE EM UMA
ORGANIZAÇÃO MILITAR**

NATAL/RN
2012

FELIPE LEONAN LUPIA CARDOSO

BACULA: APLICAÇÃO DE UM SISTEMA DE GESTÃO DE BACKUP LIVRE EM UMA ORGANIZAÇÃO MILITAR

Monografia apresentada ao curso de Especialização em Redes de Computadores do Centro Universitário do Rio Grande do Norte como requisito parcial para a obtenção do título de Especialista em Redes de Computadores.

Orientador: Prof. Ms. Aluizio F. Rocha Neto.

NATAL/RN
2012

Catálogo na Publicação – Biblioteca do UNI-RN
Setor de Processos Técnicos

Cardoso, Felipe Leonan Lupia.

Bacula : aplicação de um sistema de gestão de backup livre em uma organização militar / Felipe Leonan Lupia. – Natal, 2012.
24 f.

Orientador: Prof. Ms. Aluizio Ferreira da Rocha Neto.

Monografia (Especialização em Redes de Computadores) – Centro Universitário do Rio Grande do Norte.

1. Programa de Sistemas – Monografia. 2. Backup – Monografia.
3. Gestão – Backup – Monografia. I. Rocha Neto, Aluizio Ferreira da. II. Título.

RN/UNI-RN/BC

CDU 004.451.3

FELIPE LEONAN LUPIA CARDOSO

BACULA: APLICAÇÃO DE UM SISTEMA DE GESTÃO DE BACKUP LIVRE EM UMA ORGANIZAÇÃO MILITAR

Monografia apresentada ao curso de Especialização em Redes de Computadores do Centro Universitário do Rio Grande do Norte como requisito parcial para a obtenção do título de Especialista em Redes de Computadores.

Aprovado em: __/__/__

BANCA EXAMINADORA

Aluizio Ferreira da Rocha Neto, M.Sc.
Orientador

Membro examinador

Membro examinador

RESUMO

Este trabalho apresenta um estudo de caso sobre a implantação de um sistema de backup em uma organização militar, com o objetivo de criar um ambiente centralizado de backup em uma rede heterogênea de computadores. Baseia-se em um caso real do Centro de Lançamento da Barreira do Inferno em Parnamirim, município do Estado do Rio Grande do Norte, onde até a finalização deste trabalho não possuía um sistema eficiente de backup e utilizava scripts adaptados não correspondendo à importância e o sigilo das informações que trafegam em sua rede. A continuidade na utilização desse processo requereria uma contínua documentação detalhada sobre o funcionamento interno e execução dos scripts. Em virtude deste cenário e dos elevados preços das ferramentas proprietárias, o trabalho demonstra a factibilidade técnica e econômica no uso de um sistema de gerência de backup literalmente livre, tendo inclusive respostas similares às ferramentas pagas.

Palavras-chave: Backup. Bacula.

ABSTRACT

This paper presents a case study on the deployment of a backup system for a military organization with the goal of creating a centralized backup in a heterogeneous network of computers. It is based on a real case of the Launch Center Barreira do Inferno in Melbourn, municipality of Rio Grande do Norte, where until the completion of this work did not have an efficient system backup and used scripts not adapted corresponding to the importance and secrecy of information that travels on its network. The continued use of this process would require a continuous detailed documentation on the inner workings and execution of scripts. Given this scenario and the high prices of proprietary tools, this work demonstrates the technical and economic feasibility of using a management system free up literally, also having similar responses to the paid tools.

Keywords: Backup. Bacula.

LISTA DE SIGLAS ABREVIATURAS

CLBI	Centro de Lançamento da Barreira do Inferno
FSF	Free Software Foundation
GPL	General Public Licence

LISTA DE FIGURAS

Figura 1: Diagrama de operação em rede do Bacula.	14
Figura 2: Interação entre os componentes do Bacula.	18

SUMÁRIO

1 INTRODUÇÃO	9
1.1 OBJETIVOS	10
1.2 MOTIVAÇÃO	11
1.3 PROBLEMA DE PESQUISA	11
2 REFERENCIAL TEÓRICO	12
2.1 SOFTWARE LIVRE	12
2.2 CÓPIA DE SEGURANÇA (BACKUP)	12
2.3 O BACULA	14
2.3.1 Principais características	15
2.3.2 Componentes	16
2.3.2.1 Bacula Director	16
2.3.2.2 Bacula File	16
2.3.2.3 Bacula Storage	16
2.3.2.4 Catálogo	17
2.3.2.5 Console ..	17
3 DESENVOLVIMENTO DA SOLUÇÃO	19
3.1 CONFIGURANDO O DIRECTOR DAEMON	19
3.2 CONFIGURANDO STORAGE DAEMON	21
3.3 CONFIGURANDO O <i>FILE DAEMON</i>	21
4 CONCLUSÃO	23
REFERÊNCIAS	24
APÊNDICE A – bacula-dir.conf	25
APÊNDICE B – bacula-sd.conf	32
APÊNDICE C – bacula-fd.conf	36

1 INTRODUÇÃO

Para se armazenar qualquer informação, seja ela pessoal, comercial ou corporativa, era necessário a existência de arquivos, cujo tamanho podiam variar desde uma pequena pasta até prédios inteiros, onde toneladas de papéis eram guardadas, catalogados e tratados por especialistas. Quando um documento era muito importante, a cópia dele era criada, têm-se aí uma das primeiras formas de backup.

A virtualização de informações já é realidade na maior parte do mundo, a maioria das empresas e instituições têm hoje seus dados trafegando em redes, sejam elas locais, metropolitanas ou até mesmo globais, onde a diversidade de terminais de usuários, servidores e sistemas operacionais é enorme, nasce assim, a necessidade de um aplicativo de backup que consiga trabalhar ao mesmo tempo com todas essas diferenças e ainda de forma remota pela rede.

Existem no mercado soluções proprietárias que funcionam no ambiente descrito acima, mas, devido a sua origem, são muito dispendiosas tanto em termos de aquisição, quanto manutenção e treinamento de pessoal. Isto faz com que instituições que talvez não disponham de capital para realizar tal investimento não tenham acesso a esta tecnologia e sejam obrigados a utilizar outras soluções menos eficazes ou a até mesmo ignorar a cópia de segurança correndo o risco de perder muito em caso de algum tipo de pane ou sinistro.

De acordo Sêmola (2003, p.2), informações do tipo: “segredos de negócio, análise de mercado e da concorrência, dados operacionais históricos e pesquisas, são informações fundamentais e se revelam como um importante diferencial competitivo ligado ao crescimento e a continuidade do negócio”.

No Centro de Lançamento da Barreira do Inferno (CLBI), não encontramos as informações descritas por Sêmola, mas sim, dados de pesquisas com foguetes que envolvem inclusive a segurança nacional com níveis de sigilo muito altos que não podem ser perdidos e que requerem um tratamento adequado em relação a sua segurança.

O CLBI, criado pela Portaria nº S-139/GM3, de 12 de outubro de 1965, tem por finalidade executar e prestar apoio às atividades de lançamento e rastreo de engenhos aeroespaciais e de coleta e processamento de dados de suas cargas úteis, bem como

executar os testes, experimentos, pesquisa básica ou aplicada e outras atividades de desenvolvimento tecnológico de interesse da Aeronáutica, relacionados com a Política da Aeronáutica para Pesquisa e Desenvolvimento e com a Política Nacional de Desenvolvimento das Atividades Espaciais.

Desde a sua criação, o número de computadores conectados à rede institucional, bem como o número de servidores, serviços disponibilizados e o fluxo de dados tem aumentado de forma acelerada, mesmo assim, o Centro estava atrasado em relação aos procedimentos necessários a se garantir a segurança da informação.

O conceito de segurança da informação está relacionado à garantia de que as informações, em qualquer formato (mídias eletrônicas, papel, conversações pessoais) estejam protegidas contra o acesso por pessoas não autorizadas, bem como estejam sempre disponíveis quando necessárias, e que sejam confiáveis.

De acordo com Alves (2006, p.2), para se atingir tais objetivos é necessário um conjunto de medidas que visam proteger e preservar as informações e sistemas de informações. Dentre esses, estão a integridade, confidencialidade e disponibilidade. O conceito de integridade está relacionado à prevenção à alteração não autorizada das informações. A confidencialidade garante que os dados não sejam acessados por usuários sem privilégio. Por fim, a disponibilidade refere-se à acessibilidade da informação sempre que solicitada.

Pode-se observar os seguintes óbices em relação ao backup dos dados e dificuldades que a estruturação da rede atual impõem para o pleno funcionamento do novo sistema de backup:

- Utilização de scripts adaptados, sem documentação sobre o seu desenvolvimento para a realização do backup;
- Estrutura heterogênea da rede, múltiplas plataformas;
- Decisão sobre uso de software livre para o novo sistema de backup;

1.1 OBJETIVOS

O objetivo geral deste trabalho é substituir as ferramentas atuais de backup do Centro de Lançamento da Barreira do Inferno que atualmente são baseados em scripts, por um sistema seguro, automatizado que forneça uma real segurança aos dados de lançamento e rastreo do Centro.

1.2 MOTIVAÇÃO

Tudo isto, motivou a realização deste trabalho, pois, viu-se à necessidade do Centro em ter uma rede de computadores mais segura, contra possíveis danos a seus dados armazenados. O presente trabalho tem também como objetivo servir como base de consulta para os futuros administradores de rede da instituição.

1.3 PROBLEMA DE PESQUISA

O problema de pesquisa abordado por este trabalho consiste em demonstrar a implementação de um ambiente centralizado de backup através do conjunto de softwares livres Bacula, para substituir os scripts de backup ora utilizados nessa Organização Militar.

2 REFERENCIAL TEÓRICO

Será apresentado a seguir, os conceitos que nortearam esta pesquisa.

2.1 SOFTWARE LIVRE

Software Livre, ou Free Software, conforme a definição de software livre criada pela *Free Software Foundation*¹, é o software que pode ser usado, copiado, estudado, modificado e redistribuído sem restrição. A forma usual de um software ser distribuído livremente é sendo acompanhado por uma licença de software livre (como a GPL ou a BSD), e com a disponibilização do seu código-fonte.

Software Livre se refere à existência simultânea de quatro tipos de liberdade para os usuários do software, definidas pela FSF.

As 4 liberdades básicas associadas ao software livre são:

- A liberdade de executar o programa, para qualquer propósito (liberdade nº 0).
- A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades (liberdade nº 1).
- A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade nº 2).
- A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade nº 3).

Para que essas liberdades sejam reais, elas tem de ser irrevogáveis desde que não se faça nada errado; caso o desenvolvedor do software tenha o poder de revogar a licença, mesmo que o usuário não tenha dado motivo, o software não é livre.

2.2 CÓPIA DE SEGURANÇA (BACKUP)

Em seu livro, FARIA (2010, p.1) define backup como sendo: “uma cópia segura das informações, de forma a garantir a continuidade do negócio em caso de sinistros. A

¹ A **Free Software Foundation (FSF, Fundação para o Software Livre)** é uma organização sem fins lucrativos, que se dedica a eliminação de restrições sobre a cópia, redistribuição, estudo e modificação de programas de computadores.

cópia pode permanecer no mesmo dispositivo de origem ou armazenado em mídia externa. Os dispositivos de armazenamento mais comuns são o CD, DVD, HD externo, fitas magnéticas, mídias flash e cópia de segurança externa (online)”.

Em um sistema de gestão de backup para ser melhor compreendido o seu funcionamento, alguns conceitos devem ser esclarecidos, tais conceitos inclusive irão auxiliar no momento de realizar a configuração deste sistema.

Job – Ação (backup, restauração, verificação). Opera com *Pools*.

Volume - Faria (2010, p.6) em seu livro, define como sendo:

Divisão lógica (arquivo), projetada para o armazenamento dos dados. Os volumes podem ser criados em qualquer dispositivo de backup. Na utilização de fitas, somente é possível a criação de um único volume lógico. Todo dispositivo precisa ser “etiquetado” junto ao sistema, com isso a mídia estará pronta para a gravação.

Não existe limite de gravação no volume, o processo ocorre até o fim da mídia, desta forma, é necessário se definir o tempo de duração do uso do volume, pois, ao final deste é iniciado o período de retenção.

Período de Retenção - Barros (2007, p.5) conceitua como sendo:

Tempo em que as informações são retidas na mídia de armazenamento. Os dados não podem ser sobrescritos pelo sistema, a não ser por intervenção humana. Um tempo prolongado aumenta a quantidade de restaurações de dados específicos, contudo haverá um maior custo em mídias de armazenamento.

Expiração - Prazo final de retenção.

Pool – Volumes que compartilham as mesmas características, por exemplo, vários *Pools* que possuem o mesmo período de retenção.

Fileset – Descrição dos arquivos, diretórios e/ou partições que serão salvas pelo backup.

Catálogo – Conjunto de informações armazenadas em um banco de dados que facilita a busca dos atributos das informações que foram armazenadas no backup que tem como objetivo localizar dados de backup que serão restaurados. Com ele é possível buscar por exemplo apenas um arquivo em específico que talvez tenha sido corrompido ou apagado, sem a necessidade de se restaurar o backup por completo.

Tape Virtualization – Backup realizado em disco rígido, graças ao aumento da

sua confiabilidade e diminuição de seu preço, em substituição as fitas magnéticas.

Purge – Ação de apagar informações do catálogo em referência a um certo volume. Em relação aos dados armazenados na mídia, nenhuma alteração e/ou remoção ocorre, eles são preservados.

Prune – Eliminar informações do catálogo desnecessárias.

Política de Backup – Documento que contém os procedimentos diários de backup e restore incluindo o papel do administrador e operadores de backup, a estratégia de backup e resolução das ocorrências.

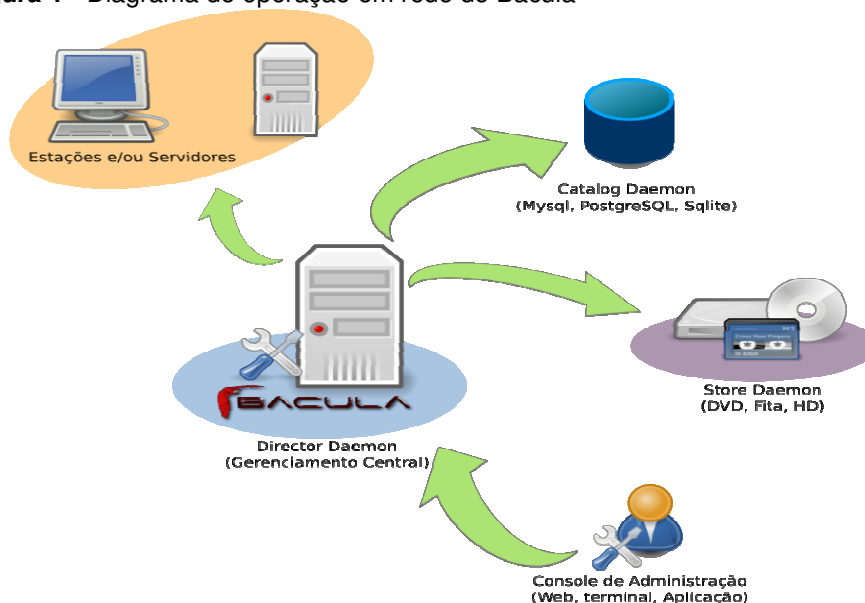
2.3 O BACULA

Bacula é um conjunto de programas que permite ao administrador de sistema, administrar backup, restauração e verificação dos dados de computadores em uma rede de sistemas mistos.

Possui características modulares que funcionam de maneira independente em máquinas diversas, inclusive com sistemas operacionais distintos.

Em termos técnicos, o Bacula, é um Programa de Backup em rede, conforme mostra o diagrama da Figura 1.

Figura 1 - Diagrama de operação em rede do Bacula



Fonte: (<http://www.bacula.com.br/?p=60>)

2.3.1 Principais características

Dentre as características do Bacula, pode-se destacar:

- Estrutura cliente/servidor
- Estrutura modular independente (director, client, database, administration console).
- GPL – economia de custos com licenças, conhecimento e possibilidade de customização da ferramenta.
- Inúmeros canais de suportes pela comunidade (mailing lists, forums, IRC channel, etc.)
- Farta documentação disponível na Internet.
- Portabilidade (módulos para diferentes sistemas operacionais – Windows, Linux, MAC, etc. – são compatíveis.
- Infinitude de recursos para a customização de backups.
- Funcionalidade que permite a execução de scripts (ou executáveis) antes/depois do início de jobs (backup/restore), tanto no cliente quanto servidor Bacula.
- Operação via linha de comando ou GUI (inclusive, com diferentes interfaces web desenvolvidas pela comunidades. Destques: webacula e o bacula-web – ferramentas de visibilidade gerencial, com gráficos, etc., sendo que a primeira ainda possibilita operações de backup, restore...)
- Suporte a maioria dos dispositivos de storage do mercado (inclusive mídias ópticas).
- Funcionalidade para o envio de mensagens de log dos trabalhos de backup/restore ou ainda instruções para o operador de backup (diferentes perfis).
- 100% compatível com o esquema GFS.
- Única ferramenta de backup multi-banco-de-dados.
- Pelo fato de ser livre, permite o desenvolvimento de uma série de “addons”, por terceiros inclusive, potencializando os recursos da ferramenta. Inclusive, já existe plugin para o Nagios (monitoração).

O Bacula pode, perfeitamente, substituir as ferramentas proprietárias mais comuns (como, por exemplo, o ArcServe da Computer Associates e o TCM, da IBM).

2.3.2 Componentes

Abaixo segue um breve resumo de todos os softwares que compõem o Bacula, sendo apresentado a função de cada um no sistema de backup.

2.3.2.1 Bacula Director

De acordo o site (BACULA, 2011) define o sub-tópico acima como: “Serviço responsável por supervisionar todos os backups, restaurações, verificações e as operações de arquivo”.

Também conhecido como Director Daemon, este serviço é responsável pela administração de todos os processos de backup, restauração, verificação e arquivamento. O Administrador de Sistema usa o Director Daemon para efetuar agendamentos de backup e para recuperar arquivos.

2.3.2.2 Bacula File

Ou *File Daemon*, é o software cliente instalado na máquina que irá ser “backupeada”. As informações (dados) são enviadas diretamente ao Bacula Storage. Existem versões clientes, para diversos sistemas operacionais: linux, BSD, Unix, Windows (9x, NT, 2000, XP, 2003) e Macintosh (OSX).

Ele também é responsável em administrar a gravação dos arquivos de restauração comandados pelo Director Daemon.

2.3.2.3 Bacula Storage

Serviço que administra a gravação e restauração dos dados em mídias apropriadas. Dos quais, as mais utilizadas são: HD, Fita dat, DVD, Cd e etc.

2.3.2.4 Catálogo

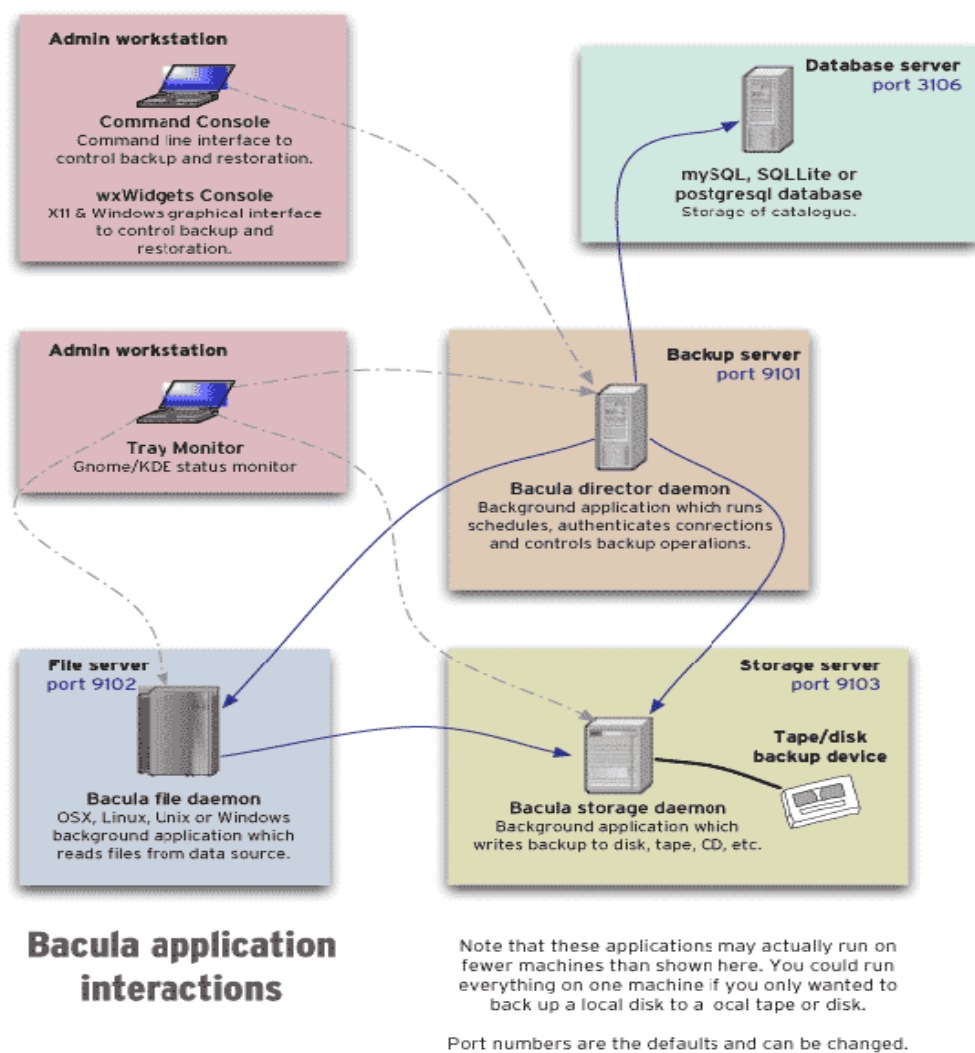
Programa responsável em manter um registro, em banco de dados, de todos os volumes gravados em mídia. Está indexação dos arquivos agiliza os processos, como por exemplo, a busca por um arquivo, entre vários volumes é consideravelmente mais rápida.

2.3.2.5 Console

Software para administração do bacula. Possui uma versão em texto puro (TTY) e versões voltadas para o ambiente web. A versão gráfica mais conhecida é o webacula e foi desenvolvida em PHP. Entre outras funcionalidades destacamos: restauração total ou parcialmente arquivos de um job, mostrar a condição dos volumes, visualizar todos os job em execução.

Na figura 2, pode-se observar uma ilustração que apresenta a interação entre os aplicativos que compõem o Bacula.

Figura 2: Interação entre os componentes do Bacula.



Fonte: (http://www.bacula.org/en/dev-manual/main/main/What_is_Bacula.html)

3 DESENVOLVIMENTO DA SOLUÇÃO

Para idealização do servidor de backup com BACULA, fez-se a instalação de um servidor Linux (Debian Lenny 6.0), sendo configurado o Bacula Director Daemon, Bacula Storage Daemon, Catalog, Console e o Banco de Dados (MySQL).

O Bacula Storage poderia ser instalado em um servidor separado, mas, optou-se para que fosse o mesmo do Bacula Director.

Nas estações e servidores que serão “backupeados” foi instalado o Bacula File Daemon.

Não apresentarei o processo de instalação de um servidor Linux, nem dos aplicativos envolvidos, será levado em consideração os servidores já se encontrarem com todos os aplicativos instalados e funcionando.

3.1 CONFIGURANDO O DIRECTOR DAEMON

O Director é o programa de controle central para todos os outros daemons. Ele agenda e monitora todos os *backups* a serem realizados.

O arquivo de configuração Director é encontrado no diretório especificado na opção *-sysconfdir* que especificou no comando *./configure*. Normalmente o arquivo de configuração do Director é nomeado *bacula.dir.conf*.

Em geral, a única mudança que deve ser feita é a modificação do recurso *FileSet* para que a diretiva de configuração contenha pelo menos uma linha com um nome válido de um diretório (ou arquivo) para ser salvo.

É provável também querer se alterar o endereço de e-mail de notificação a partir da raiz padrão para o um endereço de e-mail válido.

Finalmente, se houverem vários sistemas para ser realizado *backup*, é preciso de um *File Daemon* separado com especificações do cliente para cada sistema, especificando o seu nome, endereço e senha. Como forma de padronização e para facilitar a depuração, a comunidade que utiliza e desenvolve o Bacula nomeia seus *daemons* com o mesmo nome do sistema acrescido do sufixo *-fd* para *File Daemon*, *-dir* para Director Daemon e *-sd* para Storage Daemon. Desta forma garante-se que cada componente do Bacula terá um nome único, melhor identificação das mensagens que

podem ser enviadas por cada um dos *daemons* diferente e no caso destes compartilharem o mesmo diretório de trabalho, os arquivos temporários terão também seus nomes únicos, diferente disso poderia ocasionar muitas falhas estranhas. No apêndice A, se encontra o arquivo de configuração do Director Daemon chamado de *bacula-dir.conf* utilizado no desenvolvimento da solução.

Nesta arquivo *bacula-dir.conf*, as únicas alterações foram em relação a identificação do *Director*, *Jobs* e *FileSet*. Abaixo, segue um resumo do que cada seção deste arquivo influencia em todo o sistema.

- *Director*: para definir o nome do Director e sua senha de acesso utilizada para autenticar o programa *Console*. Só uma definição única do recurso *Director* pode aparecer no arquivo de configuração.
- *Job*: para definir os Jobs de backup e/ou restore e para unir os recursos *FileSet*, *Client* e *Schedule* a serem utilizados para cada trabalho. Normalmente, utiliza-se de nomes diferentes, correspondentes a cada cliente (ou seja, um *Job* por *Client* configurado).
- *JobDefs*: recurso opcional para fornecer padrões de recursos para os *Jobs* (*trabalhos de backup*).
- *Schedule*: define quando um trabalho (*Job*) é para ser executado automaticamente. É permitido criar qualquer número de agendamentos, mas cada trabalho terá apenas um agendamento (*Schedule*).
- *FileSet*: define o conjunto de arquivos a serem copiados de cada cliente, da mesma forma que o *Schedule* podem existir vários *FileSet* configurados mas apenas um poderá ser direcionado por cliente.
- *Client*: define os clientes do sistema Bacula, onde deverá estar configurado o *Bacula File Daemon*. Cada trabalho possui apenas um Cliente.
- *Storage*: permite configurar em qual dispositivo físico os volumes devem ser montados.
- *Pool*: para definir o conjunto de volumes que podem ser utilizados para um trabalho específico. A maioria das pessoas usa um pool de padrão único. Pools permitem a restrição de um trabalho (ou um cliente) para usar apenas um conjunto particular de Volumes.

- Catalog: definição do banco de dados para ir manter a lista de arquivos e os nomes dos volumes onde são salvos.
- Messages: para definir onde as mensagens de erro e as informações devem ser enviadas.

3.2 CONFIGURANDO STORAGE DAEMON

O *Storage Daemon* é responsável, a pedido do *Director Daemon*, para aceitar a gravação de dados enviados pelo *File Daemon* e colocá-lo em uma mídia de armazenamento, ou no caso de um pedido de restauração, para encontrar os dados e enviá-lo para o servidor de arquivos.

O arquivo de configuração do daemon de armazenamento é encontrado no diretório especificado na opção `-sysconfdir` que especificou no comando `./configure`. Por padrão, o arquivo do *Storage Daemon* é chamado *bacula-sd.conf*. No Apêndice B, está o conteúdo do arquivo configurado no sistema de backup do CLBI.

Nesta arquivo *bacula-sd.conf*, a seção alterada foi a *Device* na qual foi especificado em qual diretório seria salvo o backup e qual o tipo de mídia que seria utilizado, que no caso é *File*, o que significa que estamos usando um disco rígido como forma de armazenamento dos dados, além das modificações previstas como nome do *Director* que obrigatoriamente deve ser o mesmo que configurado no arquivo *bacula-dir.conf*.

3.3 CONFIGURANDO O FILE DAEMON

O *File Daemon* é um programa que é executado em cada máquina (cliente). A pedido do *Director Daemon*, encontra os arquivos a serem copiados e envia-os (seus dados) para o servidor de armazenamento (*Storage Daemon*).

O diretório de configuração do *daemon* é encontrado no diretório especificado na opção `-sysconfdir` que especificou no comando `./configure`. Por padrão, o arquivo de configuração do daemon de arquivo é chamado *bacula-fd.conf*. Normalmente, para usuários de primeira vez, nenhuma mudança é necessária para este arquivo. Padrões razoáveis estão definidos. No entanto, se você estiver indo para fazer backup de mais

de uma máquina, você vai precisar instalar o daemon de arquivo com uma configuração única em cada máquina cliente para fazer o backup. As informações sobre cada daemon do arquivo deve aparecer no arquivo do Director Daemon de configuração. No Apêndice C há um exemplo do arquivo de configuração `bacula-fd.conf` de uma das máquinas de onde está se gerando o backup.

Nesta arquivo `bacula-fd.conf`, não existe muito o que modificar, apenas deve-se modificar o nome do *FileDaemon* e do *Director* que deve ser o mesmo configurado no `bacula-dir.conf`.

4 CONCLUSÃO

Este trabalho objetivou a implantação de um Sistema de Backup em um ambiente de rede heterôgeneo, utilizando apenas ferramentas de Software Livre capazes de executar as tarefas de backup de uma organização, com total controle e segurança, sendo disposto para o usuário de forma clara a forma a qual os processos ocorrem. Ficou constatado também que não é necessário altos investimentos para se conseguir alcançar um nível de confiança alta no que tange a salva-guarda de informações, pois, o Bacula apresentou-se como uma plataforma de backup confiável e que está em constante evolução.

A utilização do Bacula encerrou uma preocupação que assolava os administradores da rede do Centro de Lançamento da Barreira do Inferno, onde não havia segurança alguma nos backups que eram realizados.

A implantação de um novo sistema de backup incentivou os usuários da rede a armazenarem suas informações nos servidores de arquivos, coisa que anteriormente era desencorajada devido ao risco da perda dessas informações.

A confiabilidade trouxe melhorias nos serviços do Centro, onde o tráfego de informação pela rede aumentou, pois serviços como: protocolo e encaminhamento de documentos, controle de estoques, dentre outros, hoje se encontram disponíveis na rede e protegidos pelo Bacula.

Muitas possibilidades se tornaram viáveis de serem alcançadas, como a virtualização de toda a documentação existente com o uso de assinaturas digitais, possibilidade que nem ao menos era cogitada antes da implementação de um sistema de backup de alta confiabilidade como este que foi implementado.

REFERÊNCIAS

ALVES, Gustavo Alberto. **Segurança da Informação**: uma visão inovadora da gestão. Rio de Janeiro: Ciência Moderna, 2006.

BARROS, Euriam. **Entendendo os conceitos de backup, restore e recuperação de desastres**. Rio de Janeiro: Ciência Moderna, 2007.

FARIA, Heitor Medrado. **Bacula**: ferramenta livre de backup. Rio de Janeiro: Brasport, 2010.

BACULA. Disponível em: <http://pt.wikipedia.org/wiki/Free_Software_Foundation>. Acesso em: 09 ago. 2012.

APÊNDICE A – bacula-dir.conf

```
# Arquivo de Configuração do Bacula Director.
# Criado por Sgt Felipe Leonan em Ago/2012

Director {                                # define myself
    Name = clbi20101-dir
    DIRport = 9101                        # where we listen for UA connections
    QueryFile = "/root/bacula/bin/query.sql"
    WorkingDirectory = "/root/bacula/bin/working"
    PidDirectory = "/root/bacula/bin/working"
    Maximum Concurrent Jobs = 1
    Password = "password"                # Console password
    Messages = Daemon
}

JobDefs {
    Name = "DefaultJob"
    Type = Backup
    Level = Incremental
    Client = clbi20101-fd
    FileSet = "Full Set"
    Schedule = "WeeklyCycle"
    Storage = File
    Messages = Standard
    Pool = File
    Priority = 10
    Write Bootstrap = "/root/bacula/bin/working/%c.bsr"
}

JobDefs {
    Name = "SondaIVJob"
    Type = Backup
    Level = Incremental
    Client = sondaiv-fd
    FileSet = "listadearquivos"
    Schedule = "WeeklyCycle"
    Storage = File
    Messages = Standard
    Pool = File
    Priority = 10
    Write Bootstrap = "/root/bacula/bin/working/%c.bsr"
}

# Define the main nightly save backup job
# By default, this job will back up to disk in /tmp

Job {
    Name = "BackupClient1"
    JobDefs = "DefaultJob"
}

Job {
    Name = "BackupSondaIV"
    JobDefs = "SondaIVJob"
}
```

```

Job {
    Name = "BackupLeonan"
    JobDefs = "DefaultJob"
}

#Job {
#   Name = "BackupClient2"
#   Client = clbi201012-fd
#   JobDefs = "DefaultJob"
#}

# Backup the catalog database (after the nightly save)

Job {
    Name = "BackupCatalog"
    JobDefs = "DefaultJob"
    Level = Full
    FileSet="Catalog"
    Schedule = "WeeklyCycleAfterBackup"

    # This creates an ASCII copy of the catalog
    # Arguments to make_catalog_backup.pl are:
    # make_catalog_backup.pl <catalog-name>
    RunBeforeJob = "/root/bacula/bin/make_catalog_backup.pl MyCatalog"

    # This deletes the copy of the catalog
    RunAfterJob = "/root/bacula/bin/delete_catalog_backup"
    Write Bootstrap = "/root/bacula/bin/working/%n.bsr"
    Priority = 11                # run after main backup
}

# Standard Restore template, to be changed by Console program
# Only one such job is needed for all Jobs/Clients/Storage ...

Job {
    Name = "RestoreFiles"
    Type = Restore
    Client=clbi20101-fd
    FileSet="Full Set"
    Storage = File
    Pool = Default
    Messages = Standard
    Where = /tmp/bacula-restores
}

# List of files to be backed up
FileSet {
    Name = "Full Set"
    Include {
        Options {
            signature = MD5
        }
    }

    # Put your list of files here, preceded by 'File =', one per line
    # or include an external list with:

```

```

#   File = <file-name
#   Note: / backs up everything on the root partition.
#   if you have other partitions such as /usr or /home
#   you will probably want to add them too.
#   By default this is defined to point to the Bacula binary
#   directory to give a reasonable FileSet to backup to
#   disk storage during initial testing.

    File = /home/leonanfillc/teste
}

# If you backup the root directory, the following two excluded
#   files can be useful

Exclude {
    File = /root/bacula/bin/working
    File = /tmp
    File = /proc
    File = /tmp
    File = /.journal
    File = /.fsck
}
}

FileSet {
    Name = "listadearquivos"
    Include {
        Options {
            signature = MD5
        }

        File = /mnt/dados/netlogon
    }

    Exclude {
        File = /mnt/dados/antigos
    }

}

# When to do the backups, full backup on first sunday of the month,
#   differential (i.e. incremental since full) every other sunday,
#   and incremental backups other days
Schedule {
    Name = "WeeklyCycle"
    Run = Full 1st sun at 23:05
    Run = Differential 2nd-5th sun at 23:05
    Run = Incremental mon-sat at 23:05
}

# This schedule does the catalog. It starts after the WeeklyCycle
Schedule {
    Name = "WeeklyCycleAfterBackup"
    Run = Full sun-sat at 23:10
}

# This is the backup of the catalog

```

```

FileSet {
    Name = "Catalog"
    Include {
        Options {
            signature = MD5
        }

        File = "/root/bacula/bin/working/bacula.sql"
    }
}

# Client (File Services) to backup
Client {
    Name = clbi20101-fd
    Address = localhost
    FDPort = 9102
    Catalog = MyCatalog
    Password = "password"                # password for FileDaemon
    File Retention = 30 days             # 30 days
    Job Retention = 6 months             # six months
    AutoPrune = yes                     # Prune expired Jobs/Files
}

Client {
    Name = sondaiv-fd
    Address = 192.168.40.15
    FDPort = 9102
    Catalog = MyCatalog
    Password = "password"                # password for FileDaemon
    File Retention = 30 days             # 30 days
    Job Retention = 6 months             # six months
    AutoPrune = yes                     # Prune expired Jobs/Files
}

# Second Client (File Services) to backup
# You should change Name, Address, and Password before using
#Client {
#    Name = clbi201012-fd
#    Address = localhost2
#    FDPort = 9102
#    Catalog = MyCatalog
#    Password = "password"                # password for FileDaemon 2
#    File Retention = 30 days             # 30 days
#    Job Retention = 6 months             # six months
#    AutoPrune = yes                     # Prune expired Jobs/Files
#}

# Definition of file storage device
Storage {
    Name = File
    # Do not use "localhost" here
    Address = localhost                 # N.B. Use a fully qualified name
    here
    SDPort = 9103
    Password = "password"
    Device = FileStorage
    Media Type = File

```

```

}

# Definition of DDS tape storage device
#Storage {
#   Name = DDS-4
#   Do not use "localhost" here
#   Address = localhost           # N.B. Use a fully qualified name here
#   SDPort = 9103
#   Password = "password"        # password for Storage daemon
#   Device = DDS-4               # must be same as Device in Storage daemon
#   Media Type = DD              # must be same as MediaType in Storage daemon
#   Autochanger = yes           # enable for autochanger device
#}

# Definition of 8mm tape storage device
#Storage {
#   Name = "8mmDrive"
#   Do not use "localhost" here
#   Address = localhost           # N.B. Use a fully qualified name here
#   SDPort = 9103
#   Password = "password"
#   Device = "Exabyte 8mm"
#   MediaType = "8mm"
#}

#}

# Definition of DVD storage device

#Storage {
#   Name = "DVD"
#   Do not use "localhost" here
#   Address = localhost           # N.B. Use a fully qualified name
here
#   SDPort = 9103
#   Password = "password"
#   Device = "DVD Writer"
#   MediaType = "DVD"
#}

# Generic catalog service
Catalog {
    Name = MyCatalog
#   Uncomment the following line if you want the dbi driver
#   dbdriver = "dbi:@DB_TYPE@"; dbaddress = 127.0.0.1; dbport =
    dbname = "bacula"; dbuser = "bacula"; dbpassword = ""
}

# Reasonable message delivery -- send most everything to email address
#   and to the console
Messages {
    Name = Standard
#
# NOTE! If you send to two email or more email addresses, you will need
#   to replace the %r in the from field (-f part) with a single valid
#   email address in both the mailcommand and the operatorcommand.
#   What this does is, it sets the email address that emails would display

```

```

# in the FROM field, which is by default the same email as they're being
# sent to. However, if you send email to more than one address, then
# you'll have to set the FROM address manually, to a single address.
# for example, a 'no-reply@mydomain.com', is better since that tends to
# tell (most) people that its coming from an automated source.

mailcommand = "/root/bacula/bin/bsmtp -h email.clbi.intraer -f
\"\\(Bacula\\) \\<%r\\>\" -s \"Bacula: %t %e of %c %l\" %r"
operatorcommand = "/root/bacula/bin/bsmtp -h email.clbi.intraer -f
\"\\(Bacula\\) \\<%r\\>\" -s \"Bacula: Intervention needed for %j\" %r"
mail = sti@clbi.intraer = all, !skipped
operator = sti@clbi.intraer = mount
console = all, !skipped, !saved

#

# WARNING! the following will create a file that you must cycle from
# time to time as it will grow indefinitely. However, it will
# also keep all your messages if they scroll off the console.
#

append = "/root/bacula/bin/working/log" = all, !skipped
catalog = all
}

# Message delivery for daemon messages (no job).
Messages {
    Name = Daemon
    mailcommand = "/root/bacula/bin/bsmtp -h email.clbi.intraer -f
\"\\(Bacula\\) \\<%r\\>\" -s \"Bacula daemon message\" %r"
    mail = sti@clbi.intraer = all, !skipped
    console = all, !skipped, !saved
    append = "/root/bacula/bin/working/log" = all, !skipped
}

# Default pool definition
Pool {
    Name = Default
    Pool Type = Backup
    Recycle = yes # Bacula can automatically recycle
Volumes
    AutoPrune = yes # Prune expired volumes
    Volume Retention = 365 days # one year
}

# File Pool definition
Pool {
    Name = File
    Pool Type = Backup
    Recycle = yes # Bacula can automatically recycle
Volumes
    AutoPrune = yes # Prune expired volumes
    Volume Retention = 365 days # one year
    Maximum Volume Bytes = 50G # Limit Volume size to something
reasonable

```

```
Maximum Volumes = 100           # Limit number of Volumes in Pool
}

# Scratch pool definition

Pool {
    Name = Scratch
    Pool Type = Backup
}

# Restricted console used by tray-monitor to get the status of the
director
Console {
    Name = clbi20101-mon
    Password = "password"
    CommandACL = status, .status
}
```

APÊNDICE B – bacula-sd.conf

```
# Arquivo de Configuração do Bacula Storage.
# Criado por Sgt Felipe Leonan em Ago/2012

Storage {                                # definition of myself
    Name = clbi20101-sd
    SDPort = 9103                        # Director's port
    WorkingDirectory = "/root/bacula/bin/working"
    Pid Directory = "/root/bacula/bin/working"
    Maximum Concurrent Jobs = 20
}

# List Directors who are permitted to contact Storage daemon
Director {
    Name = clbi20101-dir
    Password = "password"
}

# Restricted Director, used by tray-monitor to get the
# status of the storage daemon
Director {
    Name = clbi20101-mon
    Password = "password"
    Monitor = yes
}

# Note, for a list of additional Device templates please
# see the directory <bacula-source>/examples/devices
# Or follow the following link:
http://bacula.svn.sourceforge.net/viewvc/bacula/trunk/bacula/examples/devices/
# Devices supported by this Storage daemon
# To connect, the Director's bacula-dir.conf must have the
# same Name and MediaType.
Device {
    Name = FileStorage
    Media Type = File
    Archive Device = /root/bacula/testebackup
    LabelMedia = yes;                    # lets Bacula label unlabeled media
    Random Access = yes;
    AutomaticMount = yes;                # when device opened, read it
    RemovableMedia = no;
    AlwaysOpen = yes;
}

# An autochanger device with two drives
#Autochanger {
#    Name = Autochanger
#    Device = Drive-1
#    Device = Drive-2
#    Changer Command = "/root/bacula/bin/mtx-changer %c %o %S %a %d"
#    Changer Device = /dev/sg0
#}

#Device {
```

```

# Name = Drive-1                                #
# Drive Index = 0
# Media Type = DLT-8000
# Archive Device = /dev/nst0
# AutomaticMount = yes;                        # when device opened, read it
# AlwaysOpen = yes;
# RemovableMedia = yes;
# RandomAccess = no;
# AutoChanger = yes
#
# # Enable the Alert command only if you have the mtx package loaded
# # Note, apparently on some systems, tapeinfo resets the SCSI controller
# # thus if you turn this on, make sure it does not reset your SCSI
# # controller. I have never had any problems, and smartctl does
# # not seem to cause such problems.
# #
# Alert Command = "sh -c 'tapeinfo -f %c |grep TapeAlert|cat'"
# If you have smartctl, enable this, it has more info than tapeinfo
# Alert Command = "sh -c 'smartctl -H -l error %c'"
#}

#Device {
# Name = Drive-2                                #
# Drive Index = 1
# Media Type = DLT-8000
# Archive Device = /dev/nst1
# AutomaticMount = yes;                        # when device opened, read it
# AlwaysOpen = yes;
# RemovableMedia = yes;
# RandomAccess = no;
# AutoChanger = yes
# # Enable the Alert command only if you have the mtx package loaded
# Alert Command = "sh -c 'tapeinfo -f %c |grep TapeAlert|cat'"
# If you have smartctl, enable this, it has more info than tapeinfo
# Alert Command = "sh -c 'smartctl -H -l error %c'"
#}

# A Linux or Solaris LTO-2 tape drive
#

#Device {
# Name = LTO-2
# Media Type = LTO-2
# Archive Device = /dev/nst0
# AutomaticMount = yes;                        # when device opened, read it
# AlwaysOpen = yes;
# RemovableMedia = yes;
# RandomAccess = no;
# Maximum File Size = 3GB
## Changer Command = "/root/bacula/bin/mtx-changer %c %o %S %a %d"
## Changer Device = /dev/sg0
## AutoChanger = yes
# # Enable the Alert command only if you have the mtx package loaded
## Alert Command = "sh -c 'tapeinfo -f %c |grep TapeAlert|cat'"
## If you have smartctl, enable this, it has more info than tapeinfo
## Alert Command = "sh -c 'smartctl -H -l error %c'"
#}

```

```

# A Linux or Solaris LTO-3 tape drive
#Device {
#  Name = LTO-3
#  Media Type = LTO-3
#  Archive Device = /dev/nst0
#  AutomaticMount = yes;                # when device opened, read it
#  AlwaysOpen = yes;
#  RemovableMedia = yes;
#  RandomAccess = no;
#  Maximum File Size = 4GB
## Changer Command = "/root/bacula/bin/mtx-changer %c %o %S %a %d"
## Changer Device = /dev/sg0
## AutoChanger = yes
#  # Enable the Alert command only if you have the mtx package loaded
## Alert Command = "sh -c 'tapeinfo -f %c |grep TapeAlert|cat'"
## If you have smartctl, enable this, it has more info than tapeinfo
## Alert Command = "sh -c 'smartctl -H -l error %c'"
#}

# A Linux or Solaris LTO-4 tape drive
#Device {
#  Name = LTO-4
#  Media Type = LTO-4
#  Archive Device = /dev/nst0
#  AutomaticMount = yes;                # when device opened, read it
#  AlwaysOpen = yes;
#  RemovableMedia = yes;
#  RandomAccess = no;
#  Maximum File Size = 5GB
## Changer Command = "/root/bacula/bin/mtx-changer %c %o %S %a %d"
## Changer Device = /dev/sg0
## AutoChanger = yes
#  # Enable the Alert command only if you have the mtx package loaded
## Alert Command = "sh -c 'tapeinfo -f %c |grep TapeAlert|cat'"
## If you have smartctl, enable this, it has more info than tapeinfo
## Alert Command = "sh -c 'smartctl -H -l error %c'"
#}

# A FreeBSD tape drive
#Device {
#  Name = DDS-4
#  Description = "DDS-4 for FreeBSD"
#  Media Type = DDS-4
#  Archive Device = /dev/nsa1
#  AutomaticMount = yes;                # when device opened, read it
#  AlwaysOpen = yes
#  Offline On Unmount = no
#  Hardware End of Medium = no
#  BSF at EOM = yes
#  Backward Space Record = no
#  Fast Forward Space File = no
#  TWO EOF = yes

#  If you have smartctl, enable this, it has more info than tapeinfo

```

```
# Alert Command = "sh -c 'smartctl -H -l error %c'"
#}

# Send all messages to the Director,
# mount messages also are sent to the email address
Messages {
    Name = Standard
    director = clbi20101-dir = all
}
```

APÊNDICE C – bacula-fd.conf

```
# Arquivo de Configuração do Bacula File Daemon.
# Criado por Sgt Felipe Leonan em Ago/2012

# List Directors who are permitted to contact this File daemon
#

Director {
    Name = clbi20101-dir
    Password = "password"
}

#
# Restricted Director, used by tray-monitor to get the
# status of the file daemon
#

Director {
    Name = clbi20101-mon
    Password = "password"
    Monitor = yes
}

# "Global" File daemon configuration specifications
FileDaemon {                                # this is me
    Name = clbi20101-fd
    FDport = 9102                            # where we listen for the director
    WorkingDirectory = /root/bacula/bin/working
    Pid Directory = /root/bacula/bin/working
    Maximum Concurrent Jobs = 20
}

# Send all messages except skipped files back to Director
Messages {
    Name = Standard
    director = clbi20101-dir = all, !skipped, !restored
}
```