



Bacula

The Network Backup Solution

Apresentado por João Henrique F. de Freitas
8 Outubro de 2008 em Campinas

Bacula – a Ferramenta de Backup em Rede para *BSD,
Linux, Mac, Unix and Windows

It comes by night and sucks the vital essence from your computers.

Projeto Open Source

Bacula é uma solução de backup em rede, projetada para os sistemas *BSD, Linux, Mac OS X, Unix and Windows.

Os objetivos originais do projeto eram:

- backup de qualquer cliente de um Palm a um computador mainframe
- prover funcionalidades “Enterprise” similares aos aplicativos comerciais
- garantir a compatibilidade por 30 anos (desde que tenha hardware apropriado)
- usar uma licença Free and Open Source (GPL v2)

História do Projeto

Bacula = Backup + Dracula

- Janeiro 2000 – Project started
- 14 Abril 2002 – First release to Source Forge (version 1.16)
- 29 Junho 2006 – Release 1.38.11
- Janeiro 2007 – Release 2.0.0
- Agosto 2007 – Release 2.2.0 (current 2.2.8)

■ Downloads

670,013 all versions 4.2 TB

Introdução

Você faz backups?

- Não
- Sim, eu fiz um no mês passado
- Sim, arquivos .tar toda semana
- De vez em quando eu faço um rsync ...
- Sim, CDs toda semana
- Eu uso scripts próprios

Problemas:

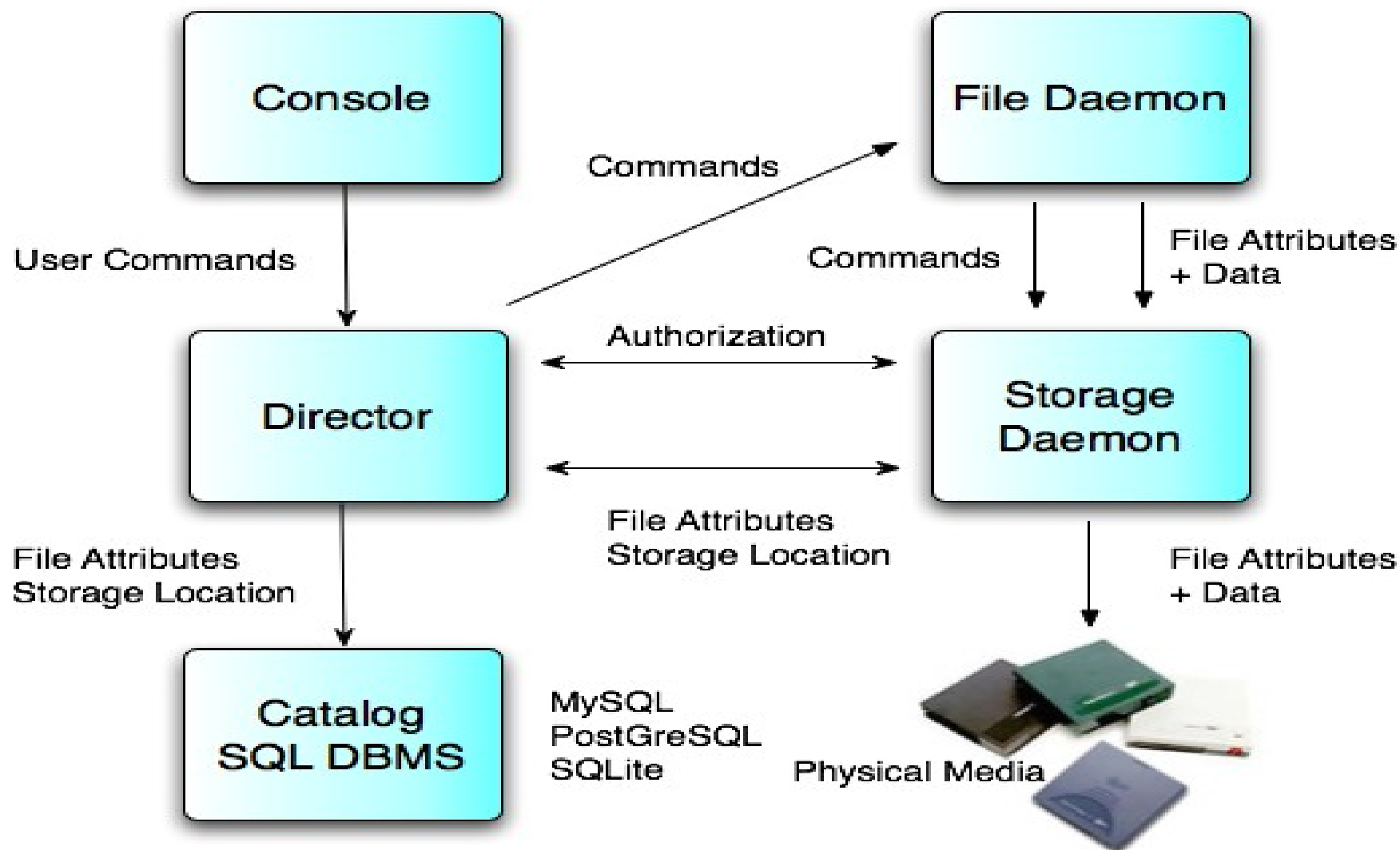
- Como você procura os arquivos que precisa restaurar?
- Com você restaura para um determinado ponto (point in time)?
- O que tem na media?
- Como você gerencia 2000 máquinas?
- Leis governamentais

Introdução

Bacula para ajudar:

- Open Source (GPLv2)
- Gerenciamento centralizado
- Backup/restore pela rede
- Muitas plataformas (*BSD, Linux, Mac OS X, Unix Win32, ...)
- Medias diferentes (Tape, disk, USB, CD/DVD)
- Confiável
- Sabe o que, quando e onde foi feito backup
- Permite restaurar os arquivos que você quer (Catalog + GUI)
- Restaura em um determinado ponto no tempo
- Escalável para manipular 10,000 máquinas

Cinco Componentes Principais



Os Cinco Componentes do Bacula

1. Director (DIR)

- Controle e administração de tudo está centralizado
- Unidade básica é o Job (um cliente, um conjunto de arquivos, ...)
- Agendamentos, inicializam e supervisionam todos os Jobs
- Mantem o catálogo (SQL database)
- Tipicamente um Director, exceto em lugares muito grandes

Os Cinco Componentes do Bacula

2. File daemon or Client (FD)

- Fazem o backup de arquivos, restore e verificação solicitados pelo Director
- Instalado em cada máquina como um serviço (daemon)
- Comunica via rede com o Director e Storage daemon
- Necessita de acesso a todos os arquivos que irão ser backed up (root, SYSTEM)
- Tipicamente muitos File daemons por Director; um para cada máquina

Os Cinco Componentes do Bacula

3. Storage daemon (SD)

- Lê e escreve dados para a mídia física
 - Disk, Tape, CD/DVD, USB, ...
- Aceita ordens e autorizações vindas do Director
- Aceita e retorna dados para/de File daemons (FD)
- Envia localização dos arquivos para o Director -> Catalog
- Tipicamente um por Director mas com múltiplos devices

Os Cinco Componentes do Bacula

4. Console

- Permite o usuário ou administrador controlar o Bacula
- Comunicação com o Director via rede
- Inicia jobs, revisão dos Job, consulta/modifica o catalog
- Meios de acesso
 - TTY (bconsole)
 - bat a Qt 4 (GUI) – mais abrangente
 - wxWidgets (GUI) – Linux, Unix, Win32
 - Gnome (GUI) – deprecated
 - Diversas interfaces web (bweb é a mais abrangente)
- Restrição nos consoles permitem os usuários restaurarem seus próprios arquivos

Os Cinco Componentes do Bacula

5. Catalog database

- Único componente não escrito pelo Bacula team
- SQL database (MySQL, PostgreSQL, or SQLite) - único
- Armazena Jobs executados, Volumes usados, localização de Arquivos, ...
- Permite restores rápidos
- Permite inquirir quando e onde os arquivos foram backed up
- Dados antigos automaticamente são prunados pelo Director
- Suporte a múltiplos database para melhor escalabilidade

Características

- Um servidor e catálogo central com backup distribuído
- Todos os componentes se comunicam via rede
- Agendamento interno para automação e execução de job simultâneos com prioridades.
- Restore interativo com muitas opções, por exemplo:
 - current backup (mais comum)
 - backup anterior com data e tempo
 - lista de arquivos/diretórios para restaurar
 - restaurar por JobId
 - ...

Características (cont.)

- Administração simples com consoles (linha de comando, GUI, e web)
- Volumes identificados, para prevenir sobreposição acidental
- Suporte para labels ANSI / IBM
- Formato dos dados do Volume independênte de Máquina - extensivo
- Suporte para Unicode no Win32; UTF-8 no Unix
- CDRROM de rescue para “bare metal” recovery (muito complicado)

Bacula – Características de Hardware

- Backups podem expandir em múltiplos volumes
- Múltiplos backups (jobs, clients, OSes) por volume
- Suporte a maioria dos tape drives com recursos configuráveis
- Suporte a múltiplos drive autochangers (libraries)
- Suporte a tape com leitores de código de barras
- Biblioteca de gerenciamento extensa para Pool e Volume
- Restauração rápida de arquivos individuais (um usuário reportou 4 a 6 horas com tar e 3 a 4 minutos com Bacula!).

Bacula – Características de Segurança

- Autorização de Daemon com CRAM-MD5
- Daemon Director e Storage podem ser executados como non-root
- MD5, SHA1, ... assinaturas para cada arquivo
- CRC checksum para cada bloco no Volume
- Restrição de consoles e tray-monitors
- Comunicação cifrada (TLS)
- Dados cifrados (PKI)
- Detecção de intruso (Tripwire like)

Bacula Jobs -- quem, oque, onde, quando

Jobs são a estrutura básica

- Name – nome único (quem)
- Type – o que fazer: backup, Backup, Migrate, Admin, Restore
- Level – nível de detalhes do tipo: Full, Differential, Incremental
- FileSet – quais arquivos fazer backup
- Client – onde pegar os arquivos (nome da máquina)
- Storage – onde colocar os arquivos (em qual hardware)
- Pool – qual conjunto de Volumes (tapes, disk) é para usar
- Schedule – quando fazer isto

Bacula – Director Arquivo de Configuração

```
Director {  
  Name = bacula-dir  
  Query File = "/usr/local/etc/query.sql"  
  Working Directory = "/var/bacula"  
  PID Directory = "/var/run"  
  Maximum Concurrent Jobs = 20  
  Password = "secret"  
  Messages = Standard  
}
```

Bacula – Director Arquivo de Configuração

```
Job {          # who, what, where, when
  Name = "Server1"
  Type = Backup
  Client = server1-fd
  FileSet = "Full Set"
  Storage = File
  Schedule = "Weekly Cycle"
  Pool = Standard
  Messages = Standard
  Write Bootstrap = "/var/bacula/server1.bsr"
}
```

Bacula – Director Arquivo de Configuração

```
Client {  
  Name = server1-fd  
  Address = server1.example.org  
  Catalog = MyCatalog  
  Password = "secret-fd"  
  File Retention = 30 days  
  Job Retention = 6 months  
  AutoPrune = yes  
  Maximum Concurrent Jobs = 20  
}
```

Bacula Configuração – FileSet

- Include/Exclude arquivos e/ou diretórios
- Regex ou wildcard para selecionar nomes em arquivos/diretórios
- Compressão usando critério similar de seleção
- Qual o tipo de filesystem para backup
- Backup de dados Access Control List (permissões)
- Manipulação de arquivos sparse
- Assinatura (MD5, SHA1, ...)

Bacula – Director Arquivo de Conf. (cont)

```
FileSet {
  Name = "Full Set"
  Include {
    Options {
      signature=SHA1;
      regex = ".*\.bak$";
      exclude = yes
    }
    File = /
    File = /usr
    File = /var
  }
  Exclude {
    File = /proc; File = /tmp; File = /sys; File = /.journal
  }
}
```

Bacula – Director Arquivo de Conf (cont)

```
Schedule {  
    Name = "Weekly Cycle"  
    Run = Level=Full 1st sun at 2:05  
    Run = Level=Differential 2nd-5th sun at 2:05  
    Run = Level=Incremental mon-sat at 2:05  
}
```

Total directives per resource:

Director=27 Client=21 Storage=21 Job=60 Schedule=3, Device=52, ...

Bacula – Storage Arquivo de Configuração

```
Device {  
  Name = File  
  Archive Device = /var/bacula/backups  
  Device Type = File    # DVD, FIFO, Tape  
  Media Type = File  
  Label Media = yes  
  Random Access = yes  
  ...  
}
```

Bacula – Storage Arquivo de Conf. (cont)

```
AutoChanger {
  Name = LTO-Changer
  Device = Drive-0, Drive-1
  Changer Device = /dev/sg0
  ...
}
Device {
  Name = Drive-0
  Archive Device = /dev/nst0
  Device Type = Tape    # DVD, File, FIFO
  Media Type = LTO-2
  Autochanger = yes
  ...
}
```


Instalações Reais

- 53TB, 150,000,000 arquivos, 90 clientes, Linux
- 40TB, 40,000,000 arquivos, 30 clientes, Solaris
- LTO-3 biblioteca com muitos drives
- Grande biblioteca com 100's slots para tape
- Bibliotecas e drives conectados com FC SAN
- 20GB, 200,000 arquivos, 1 cliente, Linux disco e tape

Desenvolvimento do Projeto

Site : <http://www.bacula.org/>

Estilo de desenvolvimento:

- Projeto no SourceForge
- Guia de Desenvolvedores com guias gerais de desenvolvimento
- Acesso ao SVN. Atualmente 16 desenvolvedores podem fazer commits
- Patches e commits revisados por K. Sibbald
- Teste de código usando uma suite de regressão
- Lista de email para desenvolvedores (bacula-devel)

Licença:

- GPL 2 copyright destinado para FSFE.
- Freedom Task Force (FTF)

Materials

For users and system administrators

- Manual: <http://www.bacula.org/en/rel-manual/index.html>
- Lista de OS e compatibilidade de Hardware (no manual)
- Bugs reports: <http://bugs.bacula.org/>
- Lista de email para suporte: bacula-users@lists.sourceforge.net

Para desenvolvedores

- Docs: <http://www.bacula.org/en/developers/index.html>
- Lista de email: bacula-devel@lists.sourceforge.net,
bacula-commits@lists.sourceforge.net
- Repositório SVN no Source Forge

Direções Futuras

- Solicitação de funcionalidades
 - Enviados pelos usuários
- Voto comunitários
 - No início do ciclo de desenvolvimento
- Projetos em desenvolvimento
 - Restauração Accurate de arquivos renomeados ou deletados
 - Merge de múltiplos backups (Backups sintéticos ou Consolidação)
 - Adição de Plugins para a diretiva FileSet Include
 - LIBDBI database driver para suportar mais SQL engines
 - Autenticação baseada em certificados
 - Melhoramento no agendamento dos jobs e resolução de conflitos

Direções Futuras (cont)

- Serviços Profissionais necessários para penetrar no mundo enterprises
- Como estruturar esforço comercial vs. Open Source?
- Código Comunitário e Enterprise geralmente são diferentes
- Soluções Enterprise necessitam funcionar com software proprietary
- GPL e código proprietário criam problemas de licença

Bacula Systems SA para salvar

- Código Community == Código Enterprise (exceto para branding)
- Suporte Profissional
- Treinamento
- Serviços de consultoria

Créditos

Agradecimentos

- Dan Langille quem criou a apresentação original
- Karl Cunningham quem atualizou
- Kern Sibbald mantenedor do projeto
- Esta apresentação foi traduzida para o Português

Uma cópia .pdf desta apresentação pode ser encontrada em:

<http://www.bacula.org> -> Presentations -> ...