Bacula
The Network Backup Solution

Presented by Kern Sibbald at UKUUG

20 February 2008 in London

Bacula – the Network Backup Tool for Linux, Mac, Unix and Windows

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

Open Source Project

Bacula is a network backup solution, designed for Linux, Mac OS, Unix and Windows systems.

Project goals are to:

- backup any client from a Palm to a mainframe computer
- provide “Enterprise” features similar to the largest commercial applications
- assure data compatibility for 30 years (providing you have the appropriate hardware)
- use a Free and Open Source (GPL v2) license

Project History

Bacula = Backup + Dracula

- January 2000 – Project started
- 14 April 2002 – First release to Source Forge (version 1.16)
- 29 June 2006 – Release 1.38.11
- January 2007 – Release 2.0.0
- August 2007 – Release 2.2.0 (current 2.2.8)

- Downloads

183,461 all versions 1.1 TB

Introduction

Do you do backups?

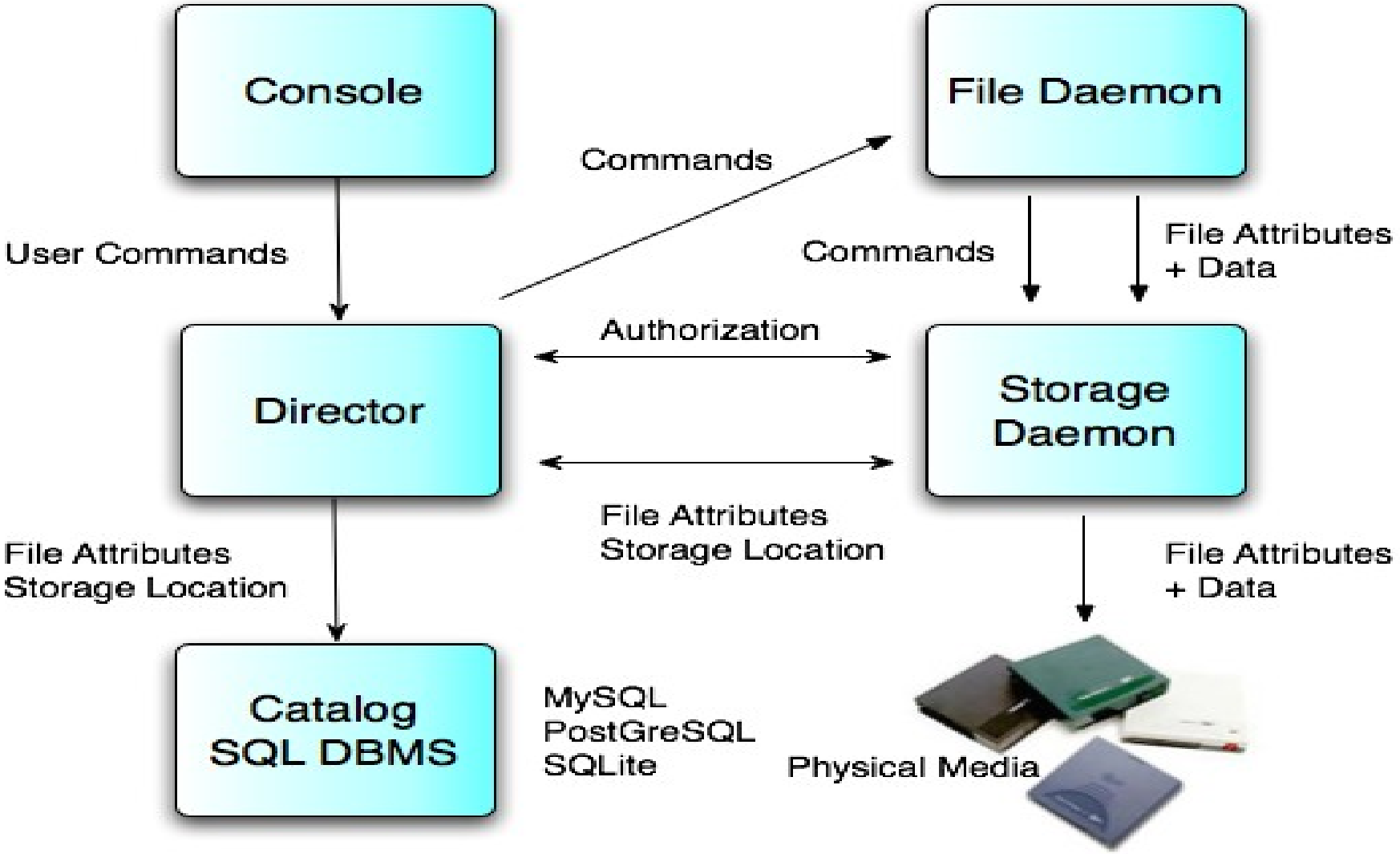
- No
- Yes, I did one last month
- Yes, tarballs every week
- Sometimes I rsync ...
- Yes, CDs every week
- Custom Scripts

Introduction

Bacula does:

- Network backup/restore
- Centrally managed
- Many platforms (Unix, *BSD, Linux, Mac OS X, Win32, ...)
- Different media (Tape, disk (files, usb, ...), CD/DVD
- Reliable
- Open Source
- Knows what was backed up when
- Allows restoring files you want (Catalog + GUI)
- Restores to a point in time
- Scales to handle 2000 machines
- Provides a bare-metal recovery (non-trivial)

Main Components



The Six Bacula Components

1. Director (DIR)

- Control and administration for everything is centralized
- Basic unit is a Job (one client, one set of files, ...)
- Schedules, initiates and supervises all Jobs
- Maintains the catalog
- Typically one Director except in very large shops

The Six Bacula Components

2. File daemon or Client (FD)

- Does file backup, restore and verification requested by Director
- Installed on each machine as a service (daemon)
- Communicates over network with Director and Storage daemon
- Needs access to all files to be backed up (root)
- Common code but adapted specifically to each OS
- Typically multiple File daemons per Director; one for each machine

The Six Bacula Components

3. Storage daemon (SD)

- Reads and writes data to the physical medium
  - Disk, Tape, CD/DVD, USB, ...
- Accepts orders and authorization from the Director
- Accepts and returns data to/from File daemons (FD)
- Sends file storage location to Director -> Catalog
- Typically one per Director but with multiple devices

The Six Bacula Components

4. Console

- Allows user or administrator to control Bacula
- Communicates with Director via network
- Start jobs, review Job output, query/modify catalog
- Consoles available
  - TTY (bconsole)
  - bat a Qt 4 (GUI) – most comprehensive
  - wxWidgets (GUI) – Linux, Unix, Win32
  - Gnome (GUI)
  - Several web interfaces
- Restricted consoles (ACL) for security

The Six Bacula Components

5. Catalog database

- Only component not written by Bacula team
- SQL database (MySQL, PostgreSQL, or SQLite)
- Tracks Jobs run, Volumes used, File locations, ...
- Permits rapid restores
- Allows inquiry of when and where files were backed up
- Old data automatically pruned by Director
- Supports multiple databases of same vendor - scaling

The Six Bacula Components

6. Tray Monitor

- Gnome/KDE/Win32 GUI tray applet
- Monitors Director, File daemons, Storage daemon
- Near real-time display of activity

Features

- A central server and catalog with distributed backup
- All components communicate via the network and are deployed separately.
- Internal scheduler for automatic and simultaneous job execution with priorities.
- Interactive restore of one or more files from:
  - current backup
  - prior backup of time and date
  - list of files/directories to restore
  - restore by JobId
  - ...
- Simple administration with consoles (command line, GUI, and web)

Features (cont.)

- Labeled Volumes, to prevent accidental overwriting
- Support for ANSI / IBM labels
- Machine independent Volume data format - extensible
- Support for Unicode on Win32; UTF-8 on Unix
- Rescue CDROM for “bare metal” recovery.

Bacula – Hardware Features

- Backups can span multiple volumes
- Multiple backups (jobs, clients, OSes) per volume
- Supports most tape drives with configurable Device resources
- Support for multiple drive autochangers (libraries)
- Supports tape barcode readers
- Extensive Pool and Volume library management
- Rapid restoration of individual files (one user reported 4 to 6 hours with tar and 3 to 4 minutes with Bacula!).

Bacula – Security Features

- Daemon authorization with CRAM-MD5
- Director and Storage daemon can be run non-root
- MD5, SHA1, ... signatures for each file
- CRC checksum for each Volume block
- Restricted consoles and tray-monitors
- Communications (TLS) encryption
- Data (PKI) encryption
- Tripwire like intrusion detection (Verify)

Technical Highlights

- OS support : Linux (all versions including the zSeries), Win32, Solaris, *BSD, Mac OS X, Irix, Tru64, AIX, HP-UX
- Backup has disk spooling capability to avoid “shoe-shine” on tapes
- Backup/restore of POSIX Access Control Lists (ACL), Mac resource forks, Win32 permissions
- Support for large files (>2GB) and 64 bit architectures
- Multi-thread implementation
- Originally written in C, now converted to a subset of C++

Bacula – Director Configuration File

```
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
}
Console {
  Name = Monitor
  Password = “monitor-secret”
  CommandACL = status, .status
  CatalogACL = BackupDB
}
```

Bacula Configuration - Job

Jobs are the basic unifying structure

- Name – unique name
- Type – what to do: backup, Backup, Migrate, Admin, Restore
- Level – level of detail of type: Full, Differential, Incremental
- FileSet – what to files to backup
- Client – where to get the files (machine name)
- Storage – where to put the files (which hardware)
- Pool – which set of Volumes (tapes, disk) to use
- Schedule – when to do it

Bacula – Director Configuration File

```
Job {
  Name = “Laptop”
  Type = Backup
  Client = laptop-fd
  FileSet = “Full Set”
  Schedule = “Weekly Cycle”
  Storage = File
  Messages = Standard
  Pool = Standard
  Write Bootstrap = “/var/bacula/laptop.bsr”
  Priority = 10
}
```

Bacula – Director Configuration File

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

Bacula Configuration – FileSet

- Include/Exclude files and/or directories
- Regex or wildcard for file/directory name selection
- Compression using similar selection criteria
- Which filesystem types to backup
- ACL support
- Sparse file handling
- Signature (MD5, SHA1, ...)

Bacula – Director Configuration File (cont)

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

Bacula – Director Configuration File (cont)

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

Bacula – File daemon Configuration File

```
FileDaemon {
  Name = laptop-fd
  Working Directory = /var/bacula
  PID Directory = /var/run
}

Director {
  Name = bacula-dir
  Password = “secret-fd”
}
```

Bacula – Storage Configuration File

```
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 Configuration File (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
   ...
}
```

Future Directions

- Feature Requests
- Community Voting
- Projects
  - Accurate restoration of renamed/deleted files
  - Merge multiple backups (Synthetic Backup or Consolidation)
  - Add Plugins to the FileSet Include statements.
- Bacula Systems SA
  - Professional support
  - Training
  - Consulting

Project development

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

Development style:

- SourceForge project
- Developer's guide with code style guidelines
- Developer SVN access. Currently 13 developers may commit
- Patches and commits reviewed by K. Sibbald
- All code tested using a regression test suite
- Email list for developers (bacula-devel)

License:

- GPL 2 copyright assigned to FSFE.
- Freedom Task Force (FTF)

Resources

For users and system administrators

- Manual: http://www.bacula.org/en/rel-manual/index.html
- OS and Hardware compatibility lists (in manual)
- Bugs reports: http://bugs.bacula.org/
- Email support list: bacula-users@lists.sourceforge.net

For developers

- Docs: http://www.bacula.org/en/developers/index.html
- Email list: bacula-devel@lists.sourceforge.net,
  bacula-commits@lists.sourceforge.net
- SVN at Source Forge

Credits

Thanks

- Dan Langille who created the original presentation
- Karl Cunningham who updated it
- This presentation draws heavily on their work

A .pdf copy of this presentation can be found at:

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