



Bacula

The Network Backup Solution

Presented by Kern Sibbald at BSDCan

17 May 2008 in Ottawa

Bacula – the Network Backup Tool for *BSD, 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 *BSD, Linux, Mac OS X, Unix and Windows systems.

Original project goals were 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

670,013 all versions 4.2 TB

Introduction

Do you do backups?

- No
- Yes, I did one last month
- Yes, tarballs every week
- Sometimes I rsync ...
- Yes, CDs every week
- I use custom scripts

Problems:

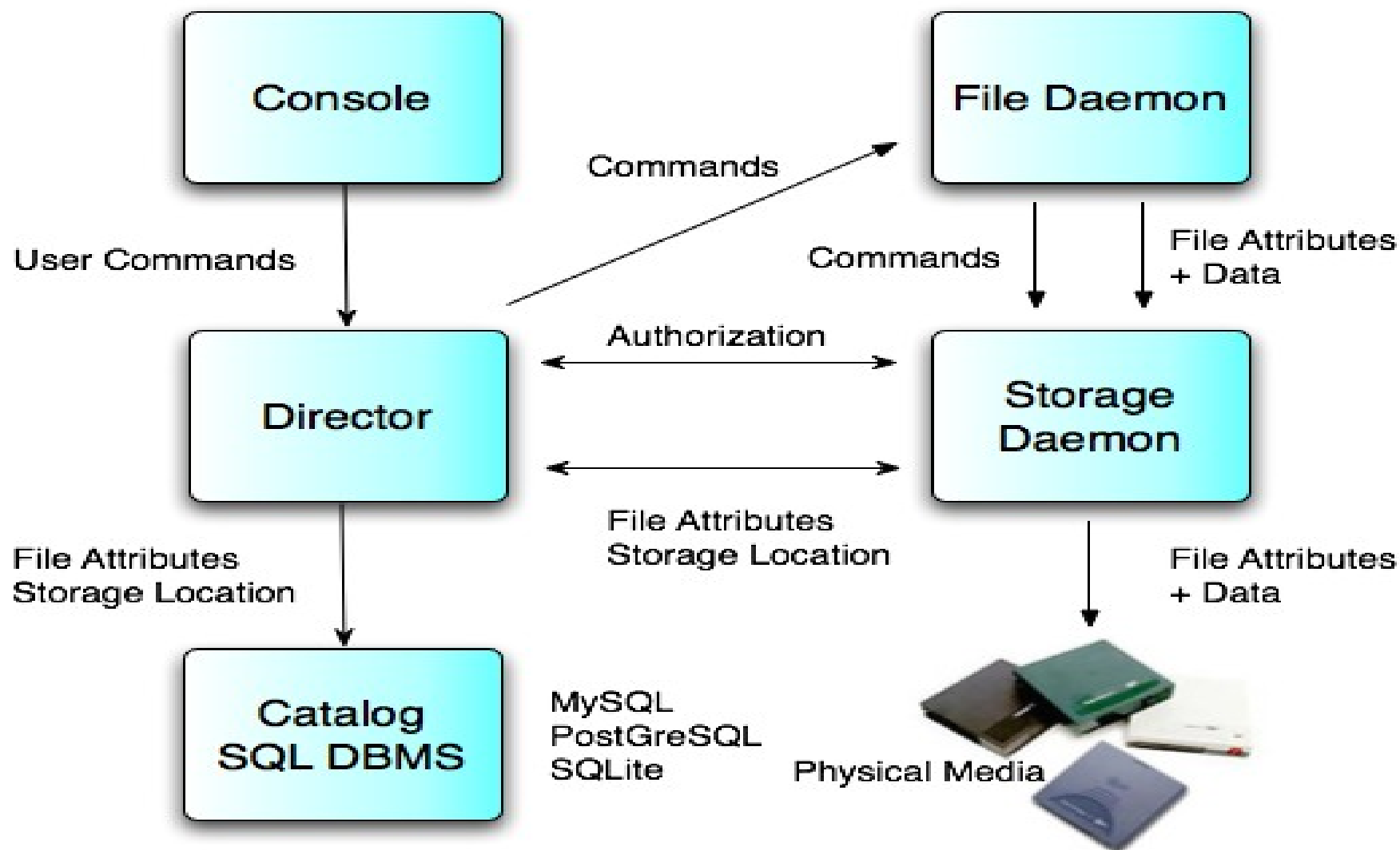
- How do you find the files you need to restore?
- How do you restore to a point in time?
- What is on what medium?
- How do you handle 2000 machines?
- Government regulations

Introduction

Bacula to the rescue:

- Open Source (GPLv2)
- Centrally managed
- Network backup/restore
- Many platforms (*BSD, Linux, Mac OS X, Unix Win32, ...)
- Different media (Tape, disk, USB, CD/DVD)
- Reliable
- Knows what was backed up when and where
- Allows restoring files you want (Catalog + GUI)
- Restores to a point in time
- Scales to handle 10,000 machines

Five Main Components



The Five 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 (SQL database)
- Typically one Director except in very large shops

The Five 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, SYSTEM)
- Typically multiple File daemons per Director; one for each machine

The Five 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 Five 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) – deprecated
 - Several web interfaces (bweb is most comprehensive)
- Restricted consoles permit users to restore their own files

The Five Bacula Components

5. Catalog database

- Only component not written by Bacula team
- SQL database (MySQL, PostgreSQL, or SQLite) - unique
- 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 for scaling

Features

- A central server and catalog with distributed backup
- All components communicate via the network.
- Internal scheduler for automatic and simultaneous job execution with priorities.
- Interactive restore with many options, for example:
 - current backup (most common)
 - prior backup of time and date
 - list of files/directories to restore
 - restore by JobId
 - ...

Features (cont.)

- Simple administration with consoles (command line, GUI, and web)
- 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 (very complicated)

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)

Bacula Jobs -- who, what, where, when

Jobs are the basic unifying structure

- Name – unique name (who)
- 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

```
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 Configuration File

```
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 Configuration File

```
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 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
- Backup OS Access Control List data (permissions)
- Sparse file handling
- Signature (MD5, SHA1, ...)

Bacula – Director Configuration File (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 Configuration File (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 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
  ...
}
```


Real Installations

- 53TB, 150,000,000 files, 90 clients, Linux
- 40TB, 40,000,000 files, 30 clients, Solaris
- LTO-3 libraries with several drives
- Large libraries with 100's of tape slots
- Libraries and drives connected with FC SAN
- 20GB, 200,000 files, 1 client, Linux disk and tape

Project development

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

Development style:

- SourceForge project
- Developer's guide with code style guidelines
- Developer SVN access. Currently 16 developers may commit
- Patches and commits reviewed by K. Sibbald
- 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

Future Directions

- Feature Requests
 - Submitted by users
- Community Voting
 - At beginning of development cycle
- Current development projects
 - Accurate restoration of renamed/deleted files
 - Merge multiple backups (Synthetic Backup or Consolidation)
 - Add Plugins to the FileSet Include statements
 - LIBDBI database driver to support more SQL engines
 - Certificate based authentication
 - Better job scheduling conflict resolution

Future Directions (cont)

- Professional services necessary to penetrate enterprises
- How to structure a commercial effort vs Open Source?
- Community and Enterprise code are often different
- Enterprise solutions must work with proprietary software
- GPL and proprietary code create licensing problems

Bacula Systems SA to the rescue

- Community code == Enterprise code (except for branding)
- Professional support
- Training
- Consulting services

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 -> ...