Backups with Bacula

— Managing backups in networks with Bacula —

Stefan Schumacher

www.kaishakunin.com
stefan.schumacher@kaishakunin.com
PGP:0xB3FBAE33

Version 1.5 vom 2008/02/10
LinuxDays Luxemburg
February 19th, 2008

解釈人

About me

- Student of Educational Science and Psychology, Otto-von-Guericke-University Magdeburg
- Softwaredeveloper for the new scientific Russian-German Dictionary
- Admin of institute's network
- Technical writer, freelancer, IT security consultant (www.kaishakunin.com)
- Linux 1997-2002, NetBSD since 2001, NetBSD developer since 2008

解釈人

ToC

解釈人

Bibliography

- http://www.net-tex.de/netbsd/backup.html
  Slides and Article (130 pages, in German)
- W. Curtis Preston
  „*Unix Backup & Recovery*“, O'Reilly 1999
- www.backupcentral.com (Storage Mountain)
- Æleen Frisch
  „*Unix System-Administration*“, O'Reilly 2003, (online)
- http://www.bacula.org
- http://www.postgresql.org/docs/

解釈人

Problems of Backups

- complexity rises
- heterogenous Networks $\rightsquigarrow$ different Operating Systems, architectures, file systems
- Amount of data $\uparrow$, timewindow $\downarrow$
- high managerial effort
- finding data (files) is complex
- Managing backups is more complex than making simple backups
- Management tools

解釈人

Strategy

1. create a strategy for the backup system, algorithmize it, write documentation
2. base for the whole backup process
3. must be updated on every change
4. basic structure works in general

解釈人

Solutions
What do I need?

- client/server backup systems instead of single backups of each client
- central management of backups
- central catalog of meta data
- central media changer (stacker, silo, robot . . .)
- free software : Amanda, Bacula
- commercial software: Veritas NetBackup, IBM TSM ...

解釈人

Strategy, Steps

1. Plan: generate an ideal strategy, cut it down to reality
2. Do: implement strategy
3. Check: check the strategy in action, identify problems
4. Act: solve problems, recycle to *Plan*

解釈人

Strategy, Steps

1. take stock (inventory)
2. identify disasters
3. identify important data
4. backup everything
5. write documentation
6. Testing, Testing, Testing

解釈人

possible disasters

- **user error**
  backup / mirror with archive
- **administrator error**
  backup / mirror whole system with archive
- **harddrive broken**
  backup, mirroring, RAID

解釈人

Inventory

- list of hardware *and* software
- use a database
- list all hardware, operating systems, programms/services
- amount of data to be backed up

解釈人

possible disasters

- **file system corruption**
- **electronic intrusion**
- **intrusion, theft, loss**
- **natural disasters**
  backup, mirror of the whole system
  (hot) backup system at different place

解釈人

What? When?

What?

- non-reproducable data (eg. logs)
- data that costs money
- operating systems, configuration data
- seperate data (code ↔ photos)
- backup sources instead of binaries

When?

- after updates (non regular, started by user)
- at useful times (cron, anacron)
- at once (replication/synchronisation)

解釈人

Documentation and Testing

- document everything, also administration tasks
- write documentation for someone else ⇝ let it check by someone else
- instuct other admins
- The System has to be tested under *real* conditions.
- Non-instructed admins starts a restore just equipped with documentation.

解釈人

backup everything?!

- Makes it sense to *not* backup a whole system?
  - Whole system can be restored.
  - Memory consumption can be ignored.
- Backup the backup systems
  Catalog (Database, logfile)
- backup everything and exclude useless data

解釈人

Backup Modes

- **Point-in-Time**: standard in database world
  all transactions are logged, logs can be restored to a defined point in time
- **Replication** copies data to another system, hot standby
- **Full**: back up everything
- **Incremental**: back up everything that has been changed since the last full backup
- **Differential**: back up everything that has been changed since the last differential backup

解釈人

Compression

- saves storage, network traffic, costs CPU time
- hardware support in tape devices
- option of bacula or via pipe to a compression programm

解釈人

Which Media?

1. reliability
2. speed
3. time to data)
4. storage place
5. costs

解釈人

Encryption

- pseudo devices: mount and backup clear text
- container/filesystems: backup encrypted
- encrypt backup with mcrypt, GnuPG, OpenSSL
- use common programms
- Verify!
- backup/restore more complex
- better: compression and encryption on file level

解釈人

Backups & Archives

- proper storage of medias (see manual)
- protect against theft/manipulation/fire/water
- use off site storage
- create a catalog of used media
  Pkey, Name, Use, Type, Files, Place, Data ...
- archive tapes (friday tape)
- archive data within the System (CVS/SVN/...)
- rotate media (avoid media rot)

解釈人

Problems

- media breaks ⇝ use multiple rotating sets
- filesystems are in use ⇝ use time frame with inactivity (night, lunch) or use filesystem snapshots
- instruct users to use directories that are getting backed up
- consolidate systems
- read log files
- verify media

解釈人

Bacula
Introduction

- backup system developed for networks
- client/server architecture with dæemons
- central config on server
- journalling meta data in a database
- supports streamer, DVD writer, hard disks, files
- stacker, silo etc. incl. bar codes

解釈人

Filesystem Snapshots

image of a filesystem:

1. stop all write processes
2. sync all blocks
3. create image
4. use system and image

backups with FSS

- create snapshot
- backup snapshot
- destroy snapshot
- real filesystem can be used

解釈人

Bacula
Introduction

- communication via TCP/IP (Cram-MD5, TSL/SSL)
- TUI and GUIs are available
- installable as client-only
- clients are available for MS Windows, MacOS X and almost every Unix
- uses *full*, *incremental* and *differential* strategy

解釈人

Bacula
Components

- **Director** (bacula-dir, Director)
- **Storage** (bacula-sd, Writes data to the media)
- **File** (bacula-fd, sends data to be backed up from client to server)
- **Catalog** (PostgreSQL, Catalog)
- **Console** (bconsole, Console to access Bacula)

解釈人

Bacula: Options

Fileset:

- Include / Exclude
- Signature: MD5/SHA1
- Compression: Gzip/Bzip
- Verify: inodes, modes, permissions, size, MD5

解釈人

Bacula
Configuration

- Configuration in .confs
- FileSet: what shall be backed up
- Client: which host shall be backed up
- Schedule: when shall be backed up
- Pool: where shall be backed up to
- Job: sums all options up

解釈人

Bacula: Options

JobDef:

- Type: Backup, Restore, Verification, Administration
- Level: Full, Incremental, Differential
- Pool: Mediapool, eg. archive tape, set of tapes
- Client, FileSet
- Storage: device (DVD, HDD, tapes)
- RunBeforeJob, RunAfterJob: clean up and external backups (Pg etc.)
- Schedule:
- administration via console

解釈人

Bacula: Restore

- JobDefs for Restore
- Restore by Hand possible (LiveCD)
- pre-defined Actions
- pure SQL in PostgreSQL ;-)

解釈人

backing up PostgreSQL

- dumping database cluster
- logical backup (pg_dump)
- Point in Time Recovery with Write Ahead Logs
- Replication (Pgpool)

解釈人

Bacula: Restore

1. Restore the last 20 Jobs
2. List all Jobs that included a special file. One of the Jobs can be restored.
3. A list of comma seperated JobIDs will be restored
4. Perform a SQL query to find a JobId. The matching ID can be restored via 3.)
5. Restores the last status of a Clients.
6. Like 5.), but a date can be specified
7. A list of files which shall be restored is read in from a file
8. Specify a file and date for that file

解釈人

Coda

Coda

解釈人

Coda

- complex networks require a complex backup system
- accurate planning required
- exhaustive testing required
- backups are a managed process, not a wild hack!

解釈人

The End

Questions?

Contact: kaishakunin.com
stefan.schumacher@kaishakunin.com

October 11th, 2008: Magdeburg, Germany:
Magdeburger Open-Source-Tag
www.open-source-tag.de

解釈人