



OpenAFS Backup Strategies

Marcus Watts
University of Michigan
mdw@umich.edu

Matt Benjamin
Linux Box Corporation
matt@linuxbox.com



Objectives

- Review first principles
- Consider varying site requirements/issues
 - small vs. large
 - catalog-based, per-file restores
 - network backup software integration
 - New hardware environment (SATA drives, declining cost of on-line storage)
- Discuss backup with respect to AFS architecture
- Summarize and categorize *free-software* backup strategies applicable to OpenAFS
- Take flack



Why Do Backups?

Restores

- Disks, servers, controllers faulty, corrupted or destroyed
- Data-centers/sites damaged, compromised, off-line, or destroyed
- Files lost, changed, damaged, destroyed

More generally, time travel. Recollect data as it was at an arbitrary point in time.



Expectations: Backup

- Whole system, infrequently
- Incremental, on a frequent basis
- Some off-site capability, usually



Expectations: Restore

- Restore a whole server
- Restore a whole volume
- Restore individual file(s)

Would like, where possible, to give individual users the capability to locate and recover previous generations of (their own) files, themselves.



Futures/Trends

- Declining cost/MB of on-line storage (SATA)
- No comparable reduction in cost of tape media and tape automation
- Who wants sequential-access media unless it is
 - Much cheaper
 - Much smaller
 - Lasts much longer in a vault
- High-capacity laser disk still interesting—need options for off-line/archival storage
- Unique value to be had in on-line, AFS-accessible storage, especially taking advantage of COW



OpenAFS Architecture

- Volumes
- Volume Replication
- Volume Backup (Copy-On-Write)
- Volume Shadows, Copies, and Dumps
- Cache Manager and Mounts
- Authentication based on Kerberos tickets



Traditional Backup Concept (eg, dump)

3-Dimensions:

- Filesystems
- Files
- Time



AFS Backup Concept

4-Dimensions:

- Server
- Volumes
- Files
- Time

Must map either volumes or files into non-AFS aware backup system, possibly losing volume dimension.



Volumes

- Volume is unit of storage management
- Volume is unit of distribution
- Volume operations support precursors to backup operations (ie, clones)
- Volume replication and backup operations provide several advanced features (eg, snapshot backup, COW)



AFS Backup Possibilities

Backup tools for AFS use one or more of the available access paths for AFS data, volume, mount, protection information:

- AFS Backup Suite
- Native filesystem interface (naïve or enhanced to handle AFS features [ACLs, system:administrators rules, mounts])
- Vos command suite
- Ibid, with knowledge of AFS dump format
- AFS library interface
- Direct access to on-disk structures



Native Filesystem

Not true that *nobody* sees this as acceptable, but this is basically how not to back up AFS (cf. *afs-bacula*)

- Unix file attributes can be set by an AFS system administrator
- Could get/set ACLs, too, if you taught your backup software to read/write AFS ACLs (not hard), also permission swizzling (no default access for admins)
- Serious volume and mount-traversal issues
- Foolish not to coordinate with replica creation, volume backup, and use the appropriate paths



AFS Backup System

OpenAFS inherits a complete backup architecture, most Transarc AFS sites have used it

- Reflects insight into mainframe tape backup procedures
 - Incrementals and hierarchy, tape management/labeling, dump history and expiry, client/server
 - Admin interface assumes presence of tape operators, but is scriptable (mount/unmount commands)
- Understands AFS volumes, one backup includes a selection of 1+ volumes (a volset), and large cell size
- Built on vos backup, inherits its features, uses the backup volume index
- Automatically spans "tapes"



AFS Backup System (Files)

AFS backup works cleanly with UNIX files, so one "tape virtualization" strategy is built-in.

- A file is a special tape device
 - A symlink to a UNIX file
 - Normally is labeled as if a tape
 - Tape device is a UNIX path, so mount command is usually scripted to manage handling of backup files
- The AFS backup database doesn't know what you called your backup files, just their labels (naming convention, location, off-line storage left to you)



AFS Backup System Issues

Reasons why some sites don't any longer use it.

- IBM migrated customers to TSM, available as XBSA tape device (still looks like AFS backup)
- Backup database is an AFS database using Ubik
 - wants to be on a standalone server, on traditional AFS hardware
 - Opaque data format (expect)
- Want to leverage the network backup you already have, and its not TSM, and it does admin-guided per-file restores
- You'd prefer to use cheap on-line storage, maybe even continuously accessible in AFS
- You would like an HSM



Tape-Oriented Systems

Scripts exist to integrate AFS with various tape systems, most automating vos dump/restore operations.

- Automate VOS dump for backup in arbitrary system
 - *Afsdump*, others
- Amanda-AFS (Mitch Collingsworth, Cornell)
- Afsdump NJIT
- NetWorkerAFS (UWisc)



Amanda

Amanda is a well-known, large-site backup system. Complex, many features, some issues:

- Dump images defined by disklist entries (DLEs), Amanda automation revolves around organizing appropriate disklists
- DLEs may not span tape images, sets maximum backup image size, but may now be RAIT
- Amanda uses a dynamic “dump cycle” in place of static dump hierarchy, relies on dump size estimates to know when to do next full, incremental dump
- Convenient tape virtualization (chg-disk)
- Slow evolution



Amanda-AFS Cornell

Mature concept, takes ideas from AFS backup volsets into Amanda.

- Uses AFS API rather than vos command suite, for dump estimates
- Supports per-file restores using *amrestore*, ACLs not (never) preserved, not sure about efficiency
- Volset syntax limited to one "server, partition, volume" triple, seems to be intention to enhance
- Not sure how backup volumes exploited



Amanda-AFS Cornell Cont

Patch for DflushVolume define, etc, trivia (eg, OpenAFS default paths):

https://secure.linuxbox.com/tiki/tiki-download_file.php?fileId=75

(Get it next week, and I'll fix large-file-server issues.)



Amanda-AFS FNAL

Initial work by Marc Mengel of FNAL on a somewhat different Amanda-AFS integration. Didn't implement it, but found these points noteworthy:

- Uses vos command suite for dump operations
- Agent runs on a fileserver machine, dumps single volumes using vos dump, parses the dump for TOC
- Dump estimates done using Unix du
- Supports per-file restores using *amrestore*, restores to a scratch partition and carries out files
- Encourages use of backup volumes, adds some assumptions for restores to r/w volume



Bacula

Recent free-software effort with distributed architecture, Unix/Windows native agents. Has momentum.

- Multi-platform support (ie, Windows)
- GUI tools (well...)
- SQL-based catalog, knows every generation of every file backed up (good or bad)
- Distributed, per-file restores
- Easy to set up
- Currently a network backup, probably has not scaled to anything like the size of a large AFS cell



Bacula and AFS

Naïve

- Runs as a POSIX client (transits CM)
- Proxies AFS credentials on a client machine
- No ACL support, or consideration of them (some paths may not be backed up)
- No knowledge of AFS volumes
- Better to ship AFS dumps to tape



A Bacula Agent for AFS

A native Bacula agent for AFS would use the same approaches used in the Amanda agents (ie, dump parsing and/or AFS APIs).

Would be nice to see:

- AFS volume sets, as in Cornell Amanda-AFS
- Proper file catalogs
- Proper per-file restores, with ACL support
- Selection of best mechanism vos dump/parse, APIs



Online Backup

The reason restores painful for users is serialized dump (tapes). If we use disk, no reason not to make backups available on-line.

- Accessible to users directly
- With correct permissions
- Administrators don't need to do per-file restores for users



AFS Online Backup

Online backup for AFS should:

- Provide on-line access, with protection, for free
- Take advantage of incremental dumps
- Provide multiple backup generations
- With space-sharing (COW)



Shadow, Copies, Clones

AFS 1.3 vos suite gets quite far towards implementing on-line, distributed snapshot backup.

Vos copy/shadow

Remote copies of volumes, which may be incremental

Vos clone

Gives you COW



Shadow, Copies, Clones: Issues

A few limitations and incompleteness, currently.

- Not enough COW (limited to 4 unassigned volume indexes [vos clone])
- VLDB integration is non-existent, overlaps with traditional clone operations in unpleasant ways
- Cache manager does not know about the non-existent VLDB integration
- Vos copy and shadow may be used safely, with correct VLDB representation
- Use of vos clone considered harmful, for now



Take Flack

What did we forget?

What should we have said about HSM (Ken?)?

Questions?

Talk available (tonight) at:

[/afs/umich.edu/user/m/d/mdw/afsbackup](afs.umich.edu/user/m/d/mdw/afsbackup)