Considerations when Choosing a Backup System for AFS

By Kristen J. Webb
President and CTO
Teradactyl LLC.

June 18, 2005

The Andrew File System® has a proven track record as a scalable and secure network file system. Organizations everywhere depend on AFS for the sharing of data both internally and externally. The most telling sign of the longevity of AFS is the adoption of the open source version of the software (OpenAFS). One of the important challenges facing AFS system administrators is the need to provide efficient, cost-effective backup strategies that can scale with ever increasing cell sizes. This paper provides an overview of the issues to consider when putting together a long term backup and recovery strategy. The topics include: Methods available for data backup and recovery; Impact of the backup function on cell performance and scale; Volume management; Centralized vs. Distributed backups; Disk-to-Tape, Disk-to-Disk and Disk-to-Disk-to-Tape comparisons; Disaster Recovery requirements; Advanced Solutions. The reader should have a basic understanding of backup and recovery concepts and familiarity with AFS architecture and cell administration.

Introduction

The purpose of this paper it to examine the general behavior of backup systems in relation to working with the Andrew File System (AFS). It is written as a guide for system administers to help make them aware of all of the considerations for backup, restore, and disaster recovery that they need to take into account when putting together a backup and recovery strategy for AFS. The paper includes discussions on the features and limitations of the native backup and recovery programs provided with AFS. It is hoped that the information presented here can help administrators consider tradeoffs in cost, performance, and reliability for different types of solutions.

Methods Available for Data Backup and Recovery

Backup solutions for AFS can be divided into two major classes. The first class of solutions includes the native backup and recovery programs provided by AFS. These solutions backup volume data from file servers using the volserver interface. They work at the AFS volume level and use a backup format specific to AFS. We will refer to this class of solutions as “intrinsic” solutions since they use the native backup format provided by AFS for backup and recovery of volume data.

The second class of solutions relies on the AFS file system. These solutions take data from a file server by accessing the /afs/cell tree in the same way that users access their data. We will refer to this class of solutions, those which do not store data in the native AFS backup format, as "extrinsic" solutions.

We will now consider some of the important differences between these two basic solution classes. We assume at this time a comparable feature set exists between intrinsic and extrinsic solutions. That is to say, both solutions are assumed to be able to back up data from a file server volume to a backup location (disk or tape) using similar backup schedules, and to restore the data from a backup location back to a location in the cell. We will talk more about advanced solutions later.

Cell Impact: An important distinction between intrinsic and extrinsic solutions is the manner in which they obtain data, and therefore, the impact that these mechanism can have on a cell. To understand the difference we review the processes that run under AFS and how each solution class impacts each process.

> **bosserver**: The basic overseer process. There is no measurable difference concerning the impact on this process between intrinsic and extrinsic solutions.
>
> **kaserver**: The authentication server process. There is no measurable difference concerning the impact on this process between intrinsic and extrinsic solutions.
>
> **ptserver**: The protection server process. This is the process that communicates information about users and groups to the fileserver process (below). Since extrinsic solutions access data through the fileserver process, they will have some impact on this process.
>
> **vlserver**: The volume location server process. This processes serves queries to the Volume Location Database (VLDB). Intrinsic solutions typically query the database in advance of backup operations to determine the volume set on each file server that will be backed up. Extrinsic solutions request information about files through the AFS client, which contacts the vlserver to determine the file server that contains the volume for the file.
>
> **fileserver**: The file server process. Extrinsic solutions need to access all file and directory information through this process. Passing backup data through the fileserver causes the backup process to compete directly for fileserver process resources. This can have an impact on normal user access to the cell.
>
> **volserver**: The volume server process: Intrinsic solutions access volume information through this process. The volserver process can impact file server performance (especially on single CPU servers) by competing with the fileserver process for server resources. Extrinsic solutions have no impact on this process.

buserver: The backup server process. The native AFS backup system uses this process to store and retrieve information about tape volumes. A backup application may optionally store tape volume information in the same way. Since backup and recovery operations are the primary functions of the buserver, it's use is not a performance issue.

upserver: The update server process. There is no measurable difference concerning the impact on this process between intrinsic and extrinsic solutions.

afsd: The client cache manager. Extrinsic processes access file and directory information using the AFS client. In most cases, this would be from dedicated backup servers, or possibly the AFS client located on each file server. This does not impact other AFS clients directly, but can impact overall client performance through the additional load placed on the fileserver process (above). Accessing data through the AFS client cache may be a bottleneck when accessing data for backup from many file servers.

Access to Volume Data

The native AFS backup utilities provide a mechanism for backup rights to be given to authenticated operators (not necessarily admin users) to initiate backup and restore requests. This feature allows operators who do not necessarily have admin rights to initiate backup and restore requests.

Intrinsic solutions usually require admin level access. Backup automation can be achieved using the –localauth option to vos from a file server. This right can be extended to dedicated backup servers by copying the cell keys to the appropriate location. From a security standpoint, since backup information for AFS (and possible other data) resides on backup servers, they need to be as or more secure than the file servers themselves.

Extrinsic solutions require an admin token to access data through the file system interface. There are additional requirements and challenges for extrinsic solutions:

Negative ACLs: Users may write Access Control List rules to block even administrators from viewing data through the file system. In order to properly back up this data, the backup application must be intelligent enough to work within these protections. One way to grant access to admin users for backing up protected data is to use the fs setaccess command. This will provide read and list access to protected directories. The backup application may then optionally restore or remove the appropriate rights once the information is obtained. Alternatively, the backup application may report what directories it failed to backup because access was not granted. The ability for users to block data from backup vs. the need to be able to protect the data from loss is an import tradeoff when deciding on an extrinsic backup solution for AFS.

Mount Points: Extrinsic solutions must be able to detect and properly handle AFS mount points during backup and recovery operations. Otherwise the application may be

exposed to potential problems such as circular mount points (a volume mounted to itself) or entering a volume of another cell that has been mounted within a user's volume.

Completeness of Data

Intrinsic solutions can backup and restore all volume information by communicating directly with the volserver or through the use of vos dump and vos restore commands. This includes meta data such as volume quotas and all directory ACLs.

While simple extrinsic solutions may properly backup and restore file and directory information, additional effort is required to obtain data specific to AFS:

> **Directory ACLs**: The vos command ensures that intrinsic solutions backup all ACLs for every directory, even during an incremental backup. A challenge for extrinsic solutions is the correct backup and recovery of directory Access Control List (ACL) meta data. Since there is no easy way to detect updates to this information, it must be collected for every directory during each backup. One way to obtain this information is to use the fs la command on each directory.
>
> **Additional Volume Information**: The volserver interface backs up additional information such as a volume's quota. Solutions that do not use this interface must be able to track such information in order to properly restore an individual volume, vice partition, file server, or an entire cell.
>
> **Unmounted Volumes**: Systems that are based on file system traversal may miss volumes that are not mounted. One workaround for this is to obtain a list of volumes from the VLDB, and then mount each volume in a well known location before performing the backup operation.
>
> **Changing Mount Points**: If a volume's mount point is moved to a new location, the view through the file system of the volumes data will change to a new absolute pathname. Extrinsic solutions must be careful not to detect this as new data. They must also be able to coordinate new incremental backups of the volume's data with previously existing backups. The best way to workaround this issue is to mount each volume to a known location for backup.

Read-Only Volumes: A backup solution for AFS should provide for the option to backup the .readonly volumes in a cell. Read only volumes contain a version of the data from the last release of a corresponding read/write volume. The read/write volume can subsequently change. A recovery scenario may require the restore of the read/write volume and any related read-only volumes. The read/write volume can be restored from the last backup and used to re-create the read-only volumes. In this case, the new read-only volumes are not the same as before the recovery. If data from read-only volumes is backed up, then the state of these volumes can be properly restored.

Summary on Intrinsic vs. Extrinsic Solutions

Backup solutions that communicate with the volserver, either directly or through the vos command have several advantages over solutions that rely on data access through the AFS file system. The impact on a cell, ability to access data, and ensuring that all data is properly being backed up are all issues to consider when deciding on a backup solution.

Volume Management Considerations

A unique feature of AFS is flexible volume management. AFS volumes can be dynamically resized or relocated to any file server within a cell without the need to reconfigure the AFS clients. This allows system administrators to balance the file servers within a cell based on size, activity, or other factors. AFS cells can be very dynamic, with volume creation and deletion happening on a daily basis. Load balancing programs such as balance can move many volumes on a frequent basis. A backup solution should provide for appropriate tracking mechanisms.

Volume Existence: The backup application must be able to know what volumes exist within a cell. New full backups must be taken for any newly created volumes. Optional archive actions may be performed for volumes that have been removed. Most applications will look to the VLDB to obtain list of volumes needed to perform backup operations.

File Server and VLDB States: To ensure all volumes are protected, it is necessary to compare file server and VLDB states on a regular basis. The vos syncser and vos syncvldb commands can be used to automatically synchronize state information between database and file servers. The volume state on each file server (vos listvol) can also be compared with the current state of the VLDB (vos listvldb).

Multiple Volume Detection: It is possible for the same read-write volume to exist in more than one location in a cell. Backup applications must be able to detect this situation and determine which volume is the correct one to backup.

Off-line volumes: Backup applications should be aware of and report any currently off-line volumes that are not being backed up.

Orphaned Vnodes: Orphaned vnodes are a result of a file on an AFS volume that no longer has a parent vnode. Extrinsic solutions cannot detect this situation through the file system. The volinfo –orphaned command can be used to detect volumes with orphaned vnodes. Intrinsic solutions can detect orphaned vnodes if they look closely at the information in the native backup stream. Their existence in a volume consumes disk space. This can result in a higher than expected quota for the volume. Volumes with orphaned vnodes can be cleaned up using the bos salvage command. Salvaging a volume frees up the disk space used by orphaned vnodes.

Distributed vs. Centralized Backups

The backup application provided with AFS allows administrators a high degree of flexibility when configuring backups. We examine the most common configurations and look at the advantages and disadvantages of each. The native AFS backup system is classified as a Disk to Tape solution.

Distributed Backups: This method is implemented by providing locally attached backup hardware (usually standalone tape drives) to each file server. Each file server is configured to backup the volumes located on that server.

Advantages:

Scale: Distributed solutions ensure that backup operations can scale with cell growth by providing massively parallel backup capabilities.

Backup Windows: The time to complete backups can be controlled by providing enough backup hardware on each file server to get the job done.

Networks: Dumping data to locally attached tape drives keeps backup traffic off of the LAN.

Disadvantages:

Cost: Dedicating backup hardware to each file server adds a significant cost to the deployment of new file servers and the growth of the cell.

Volume Relocation: Backup performance degrades as volumes are moved from one server to another. The appearance of a new volume on a server (one that has really been moved from another server) can result in the taking of a new full backup ahead of schedule.

Tape Libraries: Traditionally, the distributed approach made it difficult to take advantage of automating backup and recovery functions to a single tape library. Many tape library manufacturers now provide library partitioning that allows many file servers to share a single library.

Centralized Backups: This method is implemented by dedicating one or more systems to the backup function. These could be a subset of the file servers. However,of many sites need to also protect data that is not stored in AFS. When possible, these systems should be completely dedicated to backup operations to reduce the impact of backups on cell performance.

Advantages:

Cost: Centralized backup servers can reduce the cost of backup hardware required to deploy new file servers. Additional backup hardware may not be required as each new file server is added.

Volume Relocation: Backup performance does not necessarily degrade as volumes are moved from one server to another. New full backups should not be required on relocated volumes.

Tape Libraries: Dedicated backup servers can make full use of tape library automation for applications that backup AFS as well as other site data.

Disadvantages:

Scale: Backup of volume information to centralized backup servers is typically much slower than backup to locally attached tape devices on file servers.

Backup Windows: Available backup windows may limit the total amount of data that can be supported. This can limit cell growth or add additional backup hardware costs.

Networks: Centralized backups for AFS impact the LAN. A secondary network may be used for backups, but at an increase in cost per file server.

Mixed Solutions

One way to mitigate the pros and cons of distributed and centralized approaches is to consider something in between. Dedicating backup hardware to a single file server, to provide the backup function for several additional file servers, can lower backup hardware costs over distributed solutions. This also helps to keep backup windows under control and reduces the network impact over a centralized approach.

Summary

The native AFS backup utilities provide a flexible solution space for configuration of backups and use of backup hardware. As new file servers are brought online, there are many tradeoffs to consider when configuring backup hardware to continue to scale with cell growth.

Disk to Tape Solutions

The native backup application provided with AFS implements a Disk to Tape backup solution. Data from the file server hard disks is transferred to either locally attached tape devices or over the network to a centralized backup server. Backup applications that manage AFS data in this way are subject to the same tradeoffs of centralized vs. distributed approaches described above.

Tape Throughput and Compression

Disk to Tape solutions may not perform optimally with modern tape devices. Currently, SAIT-1 has sustained transfer rates of 30 MB/s native and 78 MB/s compressed. LTO-3 has sustained transfer rates of 68 MB/s native and 136 MB/s compressed. If the backup application cannot supply data fast enough, throughput can drop dramatically. This can result in a phenomenon called shoe-shining. The tape stops and starts a great deal while the drive waits for more data. This leads to head wear on the tape drive and can also reduce the realized compression ratio if tape compression is used. Attaching tape drives locally to the files servers will usually result in better throughput and compression than backing up data through a network to a centralized backup server.

Disk to Disk Solutions

As hard disk costs continue to decline, disk only solutions for backup and recovery are becoming more common. This method eliminates the need for tapes, tape devices, and robotic tape libraries by storing all backup information entirely on disk.

Advantages:

Fast Restore Times: Disk based restores are much faster than restores from tape. Many restores may be performed in parallel w/o the need for inter process sharing of tapes or tape devices.

Backup Windows: Network backups can be run in parallel from several file servers at once to backup server hard disks. This can result in higher network bandwidth utilization which should lead to shorter backup windows. This allows each backup server to support more data in the same available backup window.

Ease of Management: Tape mount requests for backup and restore, and tape library management are no longer required in a disk only solution.

Disadvantages:

Cost: The amount of disk space required to store backups must keep pace with data growth. If data retention requirements are long, the amount of disk space required can exceed the size of the data being protected by many factors.

Offsite Management: If backup data is required to be stored offsite, tape is still an inexpensive method to achieve this. Disk only based solutions typically require a second, offsite storage location and a method to electronically transfer backup data to that location.

Disk Failure: RAID solutions can greatly increase the reliability of data storage but they are still not 100% reliable. Malicious or accidental deletion of backup data can result in a real loss of data.

More on Data Retention Requirements

Site policy for how long backup data is retained can very greatly. Traditionally, commercial entities tended to keep information for shorter periods of time. Their primary focus for backing up data was for disaster recovery. Today, primarily for legal reasons, companies tend to keep data for longer periods of time. Education and Research institutions tend to keep data for longer periods of time (sometimes forever). They place a higher value on the ability to go back in time to retrieve data. Our experience in helping administrators choose backup retention policies seems to indicate that more is better. The ability to restore data to any point in time in the last 90 days is better than just the last 30. In addition to the pure disaster recovery nature of a backup system, these retention policies allow for a greater probability of success when restoring older lost data.

As an example scenario, consider a researchers hasty cleanup before a long two week vacation. Some important data is accidentally removed during the cleanup. After vacation, and after the researcher has had a chance to "get back on top of things" the critical data loss is discovered. If the site policy is to remove incremental backups after only two weeks, the researcher may get some version of his data back, but it may not be the most recent working copy. Since several weeks have past, it will be even more difficult to remember what the important changes were between the version of the data that was restored and what the user really needs.

This is just one example of the time lag that can occur between when data is lost and when it is actually recovered. A limitation of Disk to Disk solutions is the high cost that longer data retention policies can impose.

Disk to Disk to Tape Solutions

Many of the problems with a centralized approach to backup can be solved by introducing hard disk caching on the dedicated backup server(s). This method obtains backup data from a cell (either intrinsically or extrinsically) and stores it on disk partitions that are locally attached to a backup server. There are many advantages of this approach over simpler Disk to Tape solutions:

Tape Compression: Data can be written to tape from locally attached disk storage on the backup server faster that it can be taken over a network. This can lead to much better tape compression.

Backup Windows: Network backups can be run in parallel from several file servers at once to backup server hard disks. This results in higher network bandwidth utilization and shorter backup windows.

The limiting factor for most solutions of this type is the need to backup all data from file severs over the network. This can ultimately result in the network and the speed at which data can be sent over it as a limiting factor for the total size of a cell.

Restore and Disaster Recovery Considerations

File/Directory Restore

AFS provides fast, short term recovery via the .backup volume for read write volumes. This allows users to retrieve files and directories unassisted from a previous snapshot of their data. However, if some time has passed before the user realizes that the data they need is missing, they cannot retrieve the last version from their .backup volume. The data will need to be restored from a previous point in time from the backup system.

The native AFS backup system provides an interface to restore volume data to any point in time that a backup was taken. It may take several attempts at a restore before the data that is needed is actually found.

Solutions that provide for an on-line lookup directory can reduce the time and overhead for restoring individual files and directories by allowing operators (or users) to determine which tape(s) or disk volume(s) are needed to complete a restore properly on the first attempt.

Extrinsic solutions, by dealing with data at the file system level, may be able to save directory and file information for an online lookup directory. They may also have the capability to restore a sub set of data such as a single file or directory without having to restore an entire volume. It is possible to restore regular file and directory information (no ACLs) to a location outside of AFS.

Intrinsic solutions can provide an online lookup database by analyzing data from the native backup stream and storing useful information such as pathnames, file sizes, and modify times in an online lookup database. Restoring individual files and directories is challenging for these volume based solutions so a full volume restore may still be required. AFS now provides a restorvol program which can be used to restore data in the native backup format to a location outside of AFS.

Volume Restore

Some reasons that a complete volume restore may be required include a volume that has been accidentally deleted or one which has become corrupted beyond repair. The native AFS backup system provides plenty of useful options for performing individual volume restores using the backup volrestore command. Alternative backup solutions should be able to provide most if not all of the same restore capabilities.

Partition Recovery

The loss of an entire disk partition may require many volumes to be restored at once. While advances in RAID technology have reduced the frequency of full partition restores, a backup solution for AFS should be ready to perform this type of restore if it becomes necessary. The native AFS backup system provides for easy partition recovery using the backup diskrestore command.

File Server Recovery

The need to recover an entire file server from backup is an unlikely but not impossible event. The native AFS backup system provides for parallel recovery of multiple vice partitions using the backup diskrestore command. Alternative solutions that can perform partition recovery should be able to handle a complete server recovery. For fast recovery, the ability to perform partition and/or volume restores in parallel is a plus.

Cell Disaster Recovery

Up to this point in our discussion on recovery, we have only been concerned with recovering AFS volume data. In the worst case scenario, all that may remain of a cell's existence is the last set of tapes that were sent to an offsite location.

The ability for any solution to recover an entire cell requires careful planning, documentation, and testing. The complete details of how to plan for and perform a complete cell recovery are beyond the scope of this document.

Offsite Data Requirements for AFS

In the worst case scenario, disaster recovery for AFS will be most successful if the appropriate steps are taken to ensure the proper backup and offsite management of all data relevant to the cell. Backing up the data in all of the volumes located on the file servers is not enough to perform a complete cell recovery. The native AFS backup system provides the backup dbadd utility as a mechanism for storing the backup data on tape for offsite management. It does not provide for the backup and recovery of additional data that is critical in rebuilding a cell. Some of the most import information related to AFS that is needed to complete a cell recovery is summarized below:

/usr/afs/db: The location of the UBIK databases (database servers only)

/usr/afs/etc: The cell information directory
CellServDB: The cell database server listing
BosConfig: The bosserver configuration file
/etc: Contains files for startup (e.g. /etc/init.d/afs for Solaris)
/usr/afs/bin: The binaries for AFS

Most alternative backup solutions provide for a UNIX backup client that can be used to back up these most critical areas of the database and file servers so that the information can be sent offsite as part of the disaster recovery requirements.

Advanced Backup Solutions

Our research and discussions with many groups seeking alternative solutions for AFS backup and recovery shows that there are not many alternatives to choose from. Support from most major backup software vendors for AFS is limited if at all. The most basic support typically includes an ability to create, store, and retrieve vos dump images within the vendor's framework. This provides for the centralization of backups across an enterprise, including AFS, but does not address issues of scale, or ease of use.

Advanced Maryland Automatic Network Disk Archiver (AMANDA)

Amanda includes an extension that can backup AFS volumes. It is an intrinsic solution with a few added features. Amanda stores file and directory information in an online lookup database. It also has the ability to restore individual files and directories outside of AFS (ACLs not preserverd).

Tivoli Storage Manager (TSM)

Volume Mode

TSM provides an intrinsic solution for AFS known as a volume mode client. It is an integrated interface to the native AFS backup system that replaces the native butc tape cooridinator with a buta tape corridinator. This allows vos dump images to be stored within TSM. Backups and restores are done using the native AFS restore programs. This allows TSM to restore partitions and servers using native AFS restore commands. This interfaces also works with the AFS buserver.

File Mode

TSM also provides extrinsic solutions for AFS known as file mode clients. These clients are ACL and mount point aware. Mount point management is complex. There is an option to ignore mount points to back up an entire tree (e.g. /afs/cell/user) as a complete dataset. The file mode clients take advantage of TSM's incremental forever technology and backup server disk caching capabilities. The admin user must have rl access on all directories that are backed up. Restore of individual files or directories is available

through an online lookup database. Restore of multiple volumes must be done carefully if mount points are to be preserved.

True incremental Backup System (TiBS)

TiBS is an intrinsic backup solution for AFS. Data can be stored in multiple backup levels. The products TeraMerge technology can generate lower level backups from data previously backed up. This removes periodic full backups from AFS file servers. Nightly backups from file servers are performed by backing up only the volumes that have changed, and only the file data that has changed within a volume. Volume management is automatic. An online lookup database is available. Restores are typically done for complete volumes, but restore from a single incremental backup is possible. Partition, server, and cell recovery are available. TiBS currently supports large file (>2GB) backup available in OpenAFS 1.3.

Summary

The Andrew File System is a specialty file system with equally special requirements for backup and recovery. Smaller AFS cells have a lot of flexibility in how to manage day to day backups. As cell and data sizes grow, considerations must be taken into account to ensure that the backup system can keep pace. Each strategy has unique costs and benefits that may be important for a particular sites needs.