


/AFS

I LOVE YOU, I LOVE YOU NOT ...

AT FIRST ...

Organizations often choose AFS for one or more features or combinations of features that are not available with other available file systems

- Flexible permissions
 - Kerberos integration
 - Replication
 - Ease of use
 - Scalability
 - Integration of geographically distributed servers
 - Cross platform capabilities
 - Native, familiar user interface (GUI, command line)
 - And more – and besides – it just works ...
- 

TIME PASSES ...

AFS is adopted and, over time, “just works,” and ... recedes into the background.

“What a great solution” becomes “what does AFS do?”

Organizations lose track of what AFS is doing and what AFS can do.

Believing that AFS does “just X” (shares files) or is “just Y” (cross platform), niche solutions are implemented (SharePoint, other web based tools, drag and drop file transfer, home grown file system synchronization, NFS ...)



... LET'S GET RID OF AFS!

AFS falls victim to its own success.

- Few people are involved in maintenance and ongoing operation of AFS
 - Often one person or one FTE
- File systems are not “sexy” and not well understood by management
 - Compared to data base systems, e.g., or web-based “equivalents”
- Once implemented and integrated into the IT environment, AFS need rarely be discussed
- Implementors of other technologies may believe their niche solution replaces AFS
- Fewer and fewer IT folk understand the totality of what AFS provides
- Management finally decides to “do it in.”

AFS IS NO LONGER LOVED ...

Instead of asking “Do we already have a system that provides X?” we ask “What can we implement that provides X?”

AFS has lost its critical mass of champions ... and is declared obsolete by edict ... often with no formal analysis.

Since AFS “only provides file sharing,” e.g., then it must be easy to do without.

I mean, really, my desktop has a file system




PICK A FILESYSTEM, ANY FILESYSTEM ...

- https://en.wikipedia.org/wiki/Comparison_of_file_systems#Features



THEN DECIDE WHAT TO GIVE UP ...

- Cross platform ease of use?
 - User friendly, flexible ACLs?
 - Global, single name space?
 - Applications integrated around single name space ...
 - Institutional file system?
 - Maintained by one or two admins ...
 - Ease of scalability?
 - Flexibility of administration?
 - Live data moves/no downtime for server maintenance?
 - Cost effective adjustment of numbers of file servers?
 - Ease of adding/subtracting file servers?
 - Kerberos integration?
 - Snapshots/.backup volumes?
- 

AND LET THE WAILING BEGIN ... PIECEWISE REPLACEMENT

- Piecewise replacement of AFS is an inadequate approach
 - Abstract out one feature, X, of AFS
 - Replace X with equivalent using technology A
 - Feature X is replaced!
 - Example: move www data to NFS (easy)
 - Abstract out different feature, Y, of AFS
 - Replace Y with equivalent, using technology B
 - Feature Y is replaced!
 - Example: move some shared files to Sharepoint!

PROLIFERATION OF SOLUTIONS

- **File transfer solutions for file “sharing”**
 - Web-fronted, filesystem backed
 - Email
 - Sharepoint
 - ...
- **While some special purpose solutions do add value ...**
 - Each solution requires
 - Implementation
 - Administration
 - Maintenance

PROLIFERATION OF COSTS

- **Baseline/AFS costs:** Institutional file system maintained for institution by one or two AFS admins
- **“Disintegrated” costs**
 - Per organization solutions, sometimes > 1
 - Per organization administration/hardware/VM costs
 - Per organization skillsets
- **Difficult to assess “disintegrated” costs**
 - Organizations may or may not be able to assess costs accurately
 - Institution may or may not bother

ADDITIONAL COMPLEXITY

- **Some solutions encourage proliferation of file versions**
 - No longer one shared file in one location ...
- **Different organizations may prefer different solutions**
 - Users must learn each approach
- **Security mechanisms are likely to differ across technologies**
 - Users are unlikely to be fluent in all

APPLICATIONS AND APPLICATION DATA IN /AFS

- Applications reside in /afs
- Some depend on single name space/run on multiple nodes and require consistent paths to data
- Moving these out of /afs is painful
 - Preserve paths
 - Preserve permissions
- One current “solution” (I am not making this up ...)
 - For applications integrated around /afs single name space
 - Bring up an AFS cell, per application!
 - That is – to get rid of AFS, bring up more cells!

GETTING ON WITH IT ... MOVING DATA OUT OF /AFS

- Management pressure to retire AFS may lead to security risks
- Metric is often “how many bytes did we get out of AFS this week?”
- But:
 - Millions or billions of files
 - Per-file decisions re disposition impossible
 - Ask user “anything sensitive?”
 - Years of history in /afs space – does user know?

RUSH TO REMOVE DATA FROM /AFS ...

- Throw it over the fence
 - Recently – zip files of AFS data –
 - No check for embedded mount points
 - So no certain knowledge of archive content
 - Social engineering security risk
 - Create mount point to HR volume
 - Ask for archive
 - Get HR data as well ...
 - No access control in archive
 - Email to user

We've now distributed an archive of unsecured data ... WT?

RESULT: LOSS OF INSTITUTIONAL CONTROL OF DATA

- It's insufficient to simply remove data from AFS into some other form or system
- AFS mount points are likely to be “seen” as directories – so care must be taken when using recursive processing to move AFS data to other systems/formats
- Institutional and legal considerations require that existing access restrictions/ACLs are respected in the target system/format
- AFS ACLs don't “travel” from one file system to another – so must be translated to equivalent on target file system
 - If possible ...
- The authorization mechanism of a target system should be at least as robust as Kerberos/AFS

RECOMMENDATIONS ... IF YOU MUST ~~/AFS~~

- Ask questions first, shoot later ...
 - Why get rid of /afs?
 - There are legitimate reasons, of course ...
 - All of these are addressed in AuriStor
 - (And that suggestion Lead balloon.)
 - What are the “disintegrated” costs?
 - What functionality is lost? What do we give up?
 - Cost of migration of data?
 - Who takes care of target systems?
 - Will some organizations choose to continue using AFS?
 - What’s the time frame? How long does /afs exist in parallel with new?
 - How do we maintain data security throughout the migration!!!

OR FAIL ...

Prediction ... given that no formal analysis has been performed where I am ...

I will still be “getting rid of it” many years from now ... and we may come full circle and realize that, after all, an institutional file system isn’t the root of all evil ... and move data back where it belongs ...

