

# When Your Business Depends On It

---

## The Evolution of a Global File System for a Global Enterprise

**Phillip Moore**

**Phil.Moore@MorganStanley.com**

**Executive Director, UNIX Engineering**

**Morgan Stanley and Co.**

**Member, OpenAFS Council of Elders**

**(AKA: OpenAFS Advisory Board)**

---

# Overview

---

- **AFS in Aurora (MS Environment)**
  - **VMS (Volume Management System)**
  - **Auditing and Reporting**
  - **AFS Growing Pains**
  - **Future Directions**
-

## AFS in Aurora ( MS Environment )

---

- **For Aurora Project information see LISA '95 paper:**
    - <http://www.usenix.org/publications/library/proceedings/lisa95/gittler.html>
  - **Definition of Enterprise/Scale**
  - **Kerberos Environment**
  - **AFS Environment**
-

## AFS in Aurora • Definition of Enterprise/Scale

---

**"Enterprise" unfortunately means "Department" or "Workgroup" to many vendors. "Scale" is often simply assumed to mean "number of hosts". It's not that simple:**

- **Machines: How Many and Where**
    - 25000+ hosts in 50+ sites on 6 continents, sites ranging in size from 1500 down to 3
  - **Topology and Bandwidth of Network**
    - Metropolitan WANs, very high bandwidth
    - Intercontinental WANs, as low as 64K
  - **System Criticality and Availability**
    - 24 x 7 System Usage
    - Near-zero or Zero Downtime Requirement
-

## AFS in Aurora • Kerberos Environment

---

- **Single, Global Kerberos Realm**
  - **Currently migrating from Cybersafe Challenger to MIT**
  - **All AFS cells share same KeyFile**
  - **All UNIX Authentication Entry Points are Kerberized, and provide**
    - Kerberos 5 tickets
    - Kerberos 4 tickets
    - AFS tokens (for all cells in CellServDB)
  - **Many Applications/Systems use Kerberos credentials for authentication**
-

# AFS in Aurora • AFS Environment

---

- **AFS is the Primary Distributed Filesystem for all UNIX hosts**
  - **Most UNIX hosts are dataless AFS clients**
    - Exceptions: AFS servers (duh), Backup servers
  - **Most Production Applications run from AFS**
  - **No AFS? No UNIX**
-

# AFS in Aurora • Why AFS

---

- **Superior client/server ratio**
    - NFSv1 servers (circa 1993) topped out at 25:1
    - AFS went into the 100s
  - **Robust volume replication**
    - NFS servers go down, and take their clients with them
    - AFS servers go down, no one notices (OK, for RO data only)
  - **WAN File sharing**
    - NFS just couldn't do it reliably
    - AFS worked like a charm
  - **Perhaps surprisingly, Security was NEVER a serious consideration**
    - However, had there been no pre-existing Krb4 infrastructure, AFS may have never been considered, due to the added integration challenges
-

# VMS (Volume Management System)

---

- **VMS :: Features**
    - Authentication and Authorization
    - Automated Filesystem Operations
    - The /ms Namespace
    - Incremental/Parallel Volume Distribution Mechanism
  - **VMS :: Implementation**
    - Uses RDBM (Sybase) for Backend Database
    - Coded in perl5 (but architected in perl4), SQL
    - Uses Perl API for fs/pts/vos/bos commands
-

# VMS: The Global Filesystem

---

- One top-level AFS “mount point (`/ms` instead of `/afs`)
- Choice of `/ms` stresses *namespace*, not filesystem technology or protocol
- Original plan was to migrate `/ms` from AFS to DFS/DCE
- Traditional `/afs` namespace exposes individual AFS cells, `/ms` hides them.

Traditional AFS	MS Namespace
<code>/afs/transarc.com</code> <code>ibm.com</code> <code>cmu.edu</code> <code>nasa.gov</code> <code>...</code> <code>...</code>	<code>/ms/.global/ny.a</code> <code>ny.b</code> <code>...</code> <code>.local</code> <code>dev</code> <code>dist</code> <code>group</code> <code>user</code>

# VMS: The Top Level Namespace

---

- Six Top Level Directories under /ms

Type	Directory	Function
Special	.global	Cell-specific, globally visible data
	.local	Local view of cell-specific data
Readonly	dist	Replicated, distributed data
Readwrite	dev	MSDE Development Area
	group	Arbitrary RW Data
	user	<i>Human</i> User Home Dirs

# ReadWrite Namespace

---

- **Three top level paths for globally visible, readwrite data**
  - `/ms/dev`
  - `/ms/group`
  - `/ms/user`
- **Location Independent Paths, symlinks that redirect into the cell-specific .global namespace**
  - `/ms/dev/perl5/AFS-Command -> ../../global/ny.u/dev/perl5/AFS-Command/`
  - `/ms/user/w/wpm -> ../../global/ny.w/user/w/wpm/`
  - `/ms/group/it/afs -> ../../global/ny.u/group/it/afs/`
- **Use of “canonical” location independent paths allows us to easily move data from one cell to another**
- **Data in RW namespace is NOT replicated**

# Global Cell Distribution

---

- **Limits on Scalability**
  - Fileservers scale infinitely
  - Database server do NOT (Ubik protocol limitations)
- **Boundaries between cells determined by bandwidth and connectivity.**
  - Originally, this meant one or two cells per building
    - Two cells per building in large sites (redundancy)
    - One cell per building in small sites (cost)
  - Today, large sites implement the Campus Model, some small sites have no local cell, and depend on the nearest campus.
- **As of December 2003, we have 43 AFS cells**
  - 21 Cells in 4 Campuses (NY, LN, HK, TK)
    - 17 Production, 4 Dev/QA
  - 20 Standalone Cells in Branch Offices
  - 2 Engineering/Test cells (NY)

# MSDE Namespace (dev, dist)

---

- **MPR = Metaproj/Project/Release**
  - **Metaproj:** Group of related Projects
  - **Project:** typically a single software “product”
  - **Release:** typically a software version, such as 1.0, 2.1, etc.
- **RW data for a single project lives in only one AFS cell**
  - `/ms/dev/afs/vms -> ../../global/ny.v/dev/afs/vms/`
- **RW data for a *metaproj* can be distributed globally by placing different projects in different AFS cells.**
  - `/ms/dev/perl5/jcode -> ../../global/tk.w/dev/perl5/jcode/`
  - `/ms/dev/perl5/core -> ../../global/ny.v/dev/perl5/core/`
  - `/ms/dev/perl5/libxml-perl -> ../../global/in.w/dev/perl5/libxml-perl/`
- **Projects should be located “near” the primary developers, for performance reasons, but they are *still visible globally*.**

# MSDE Namespace (dist)

---

- **/ms/dev is:**
  - Not replicated
  - Not distributed (data lives in ONE AFS cell)
  - Readwrite
  - Obviously *not* suitable for use in production (obvious, right?)
- **/ms/dist is:**
  - Replicated
  - Distributed
  - Readonly
- **WARNING: Existence in /ms/dist does NOT automatically imply production readiness**
  - A *necessary* but not a *sufficient* condition
  - “Production” status of applications is *not* managed by VMS (yet...)

# MSDE Namespace (default namespace)

---

- The “default” namespace merges the relative pathnames from numerous projects into a single, virtual directory structure

- Fully qualified, release-specific paths:

```
/ms/dist/foo/PROJ/bar/1.0/common/etc/bar.conf
                               man/man1/bar.1
                               exec/bin/bar
/ms/dist/foo/PROJ/baz/2.1/common/man/man1/baz.1
                               exec/bin/baz
/ms/dist/foo/PROJ/lib/1.1/common/include/header.h
                               exec/lib/libblah.so
```

- Default symlinks:

```
/ms/dist/foo/bin/bar           -> ../PROJ/bar/1.0/exec/bin/bar
bin/baz                         -> ../PROJ/baz/2.1/exec/bin/baz
etc/bar.conf                    -> ../PROJ/bar/1.0/common/etc/bar.conf
include/header.h               -> ../PROJ/lib/1.1/common/include/header.h
lib/libblah.so                 -> ../PROJ/lib/1.1/exec/lib/libblah.so
man/man1/bar.1                 -> ../../PROJ/bar/1.0/common/man/man1/bar.1
man/man1/baz.1                 -> ../../PROJ/baz/2.1/common/man/man1/baz.1
```

# MSDE Namespace (default namespace, cont'd)

---

- Each distinct project can have ***ONE AND ONLY ONE*** default release
- **Relative pathname conflicts are not allowed**
  - If both *foo/bar/1.0* and *foo/baz/2.1* have a *bin/configure*, then only one of them can be made default.
- **Defaults make it easier to configure the environment**
  - prepend PATH */ms/dist/foo/bin*
  - prepend MANPATH */ms/dist/foo/man*
- **Defaults are useful, but not ever production releases has to be made default.**
  - Change Control is covered in Day Two

# Auditing and Reporting • Cell Auditing

---

- **'bosaudit' checks the status of all the AFS database and file servers cell-wide. Some of the key auditing features include:**
    - All Ubik services have quorum, uptodate database versions, and a single Ubik sync site
    - All Encryption keys are identical
    - Consistent server CellServDB configurations
    - Reports on Missing or Incorrect BosConfig entries
    - Disabled or temporarily enabled processes
    - Presence of core files
-

## Auditing and Reporting • Cell Auditing (cont)

---

**'vldbaudit' queries the entire VLDB and listvol output from all file servers in the cell and does a full 2-way sanity check, reporting on:**

- Missing volumes (found in VLDB, not on specified server/partition)
  - Orphan volumes
  - Offline volumes
  - Incorrectly replicated volumes (missing RO clone, too few RO sites)
-

## Auditing and Reporting • LastAccess Data

---

- **Question: when was the last time someone accessed an AFS volume**
    - vos commands won't tell you
    - volinfo will
  - **Batch jobs collect cell-wide volinfo data**
  - **Data is correlated with VMS namespace, and per-release, per-project rollups are possible**
  - **Time for a demo...**
-

# AFS Horror Stories

---

- **Cell Wide Outages and other unpleasant disasters**
    - vos delentry root.afs
    - Busy/abort floods
  - **Slow disks (or a slow SAN), can mean client hangs**
  - **RW Cluster recovery**
  - **A RW server hangs in New York, and a VCS cluster in Tokyo panics**
-

# AFS Architectural Problems

---

- **Single Threaded Client**
  - **Single Threaded volserver**
    - Solution is on the way
  - **Windows client SMB “hack”**
  - **“vos” is WAY too smart**
  - **PAGs, or the lack thereof, in Linux 2.6**
-

# AFS Politics and Culture

---

- **Not a modern, sexy, technology anymore**
  - **Taken for granted**
  - **Every two years we have the “How can we get rid of AFS” department offsite**
    - Same conclusion every time: we’re stuck with it.
  - **Huge IT investment in storage technologies (SAN, NAS, appliances, etc), but... The Storage Engineering group doesn’t manage AFS**
    - Politics, not technology
-

# AFS at Morgan Stanley: The Future

---

- **Its here to stay: as goes AFS, so goes Aurora**
  - **Use of RW data being actively discouraged**
    - But wait until they find out how insecure NFS is, even V4.
  - **Windows clients are about to explode**
    - OK, *usage* is going to explode, not the clients (I can dream...)
  - **No plans to replace AFS/VMS for managing software distribution**
    - VMS desperately needs a complete rewrite
-