When Your Business Depends On It

Development and Deployment of a Global File System for a Global Enterprise

W. Phillip Moore

wpm@ms.com

Vice President

Morgan Stanley

Overview

- AFS in Aurora (MS Environment)
- VMS (Volume Management System)
- PTS Database Synchronization
- 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**
  - 6000+ hosts in 30+ sites on 6 continents, sites ranging in size from 1500 down to 3
- **Topology and Bandwidth of Network**
  - Metropolitan WANs at multiple T3 bandwidth
  - Intercontinental WANs as low as 32K
- **System Criticality and Availability**
  - 24 x 7 System Usage
  - Near-zero or Zero Downtime Requirement

AFS in Aurora • Kerberos Environment

- **Single, Global Kerberos Realm**
- **Commercial Kerberos 5 product (Cybersafe Challenger), not AFS Kerberos.**
- **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**
- **Most Production Applications run from AFS**
- **No AFS? No UNIX**

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 and Transact-SQL
  - Uses Perl API for fs/pts/vos/bos commands

VMS • Authentication and Authorization

- Server Process runs as root, authenticated as system:administrators member in all AFS cells
- Client performs KRB4 Mutual Authentication with server to authenticate user identity
- Server uses PTS group membership or user to determine authorization

VMS • Automated Filesystem Operations

- **Users/admins are not required to think at the volume/mtpt/ACL level.**
- **VMS allows users/admins to work with large macroscopic filesystemstructures.**
- **e.g.,** `vms create project somemeta someproj somerelease` **will run 1000's of** `vos/fs/pts` **commands to create** `/ms/dev/somemeta/someproj`**, make it visible in the namespace globally, create the initial release, etc. This will create 10 or more volumes in one AFS cell, and distribute changes to** `/ms/dev` **to most of the cells around the world.**

VMS • Automated Filesystem Operations

Currently supported VMS operations

	create	destroy	move
user	yes	no	yes
group	yes	no	no
project	yes	yes	no

e.g., 'vms move user' was an early necessity once real people started living in AFS.

VMS • The /ms Namespace - Top Level

- **Important: we are implementing a namespace (/ms), NOT a filesystem technology (NFS, AFS, DFS).**

```
/ms/.global  \____  System Directories
    .local   /
    dev      \
    dist      \___  User Directories
    group     /
    user     /
```

  - /ms/dev Development Environment (source code)
  - /ms/dist Distributed Data (applications, data, etc)
  - /ms/user Human User Home Directories
  - /ms/group None of the Above (i.e. everything else)
  - dev, user and group are all non-replicated RW data
  - dist is always RO replicated data

VMS • The /ms Namespace - Global Visibility

```
/ms/.global/<short cell name>
            sa.a
            cw.a
            cw.b
            eb.a
            etc...
```

- The "short cell name" is an abbreviation of the official cell names. The cell name syntax is:

  `[a-z].<building>.ms.com`

- Where the "building" is a unique 2-character building identifier, and the first field simply distinguishes multiple cells in large buildings. e.g., a.sa.ms.com, b.cw.ms.com, etc.

VMS • The /ms Namespace - Canonical Names

- **The 4 top level "user" directories make up the canonical namespace.**
- **Location of RW data is hidden through symlinks which indirect the data location through /ms/.global.**
- **RW Paths:**
  - /ms/user/w/wpm -> /ms/.global/sa.a/user/w/wpm
  - /ms/group/it/afs -> /ms/.global/sa.a/group/it/afs
  - /ms/dev/somemeta/someproj -> /ms/.global/hq.a/dev/somemeta/someproj
- **RO Paths:**
  - /ms/dist is a hierarchy of mount points for volumes which are replicated, and assumed to come from the clients local AFS cell.

VMS • The /ms Namespace - Development Environment

- **Key: All replicated data in /ms/dist has a canonical "source" in /ms/dev.**

```
/ms/dev/metaproj/project/release/src
                                 build
                                 install/common
                                         exec -> .exec/@sys
                                         .exec/<sysname values>
```

- **src** ⇒ **Source Code, Makefiles, Documentation**
- **build** ⇒ **Temporary Compilation Space**
- **install** ⇒ **Location of data to be installed/replicated**

VMS • The /ms Namespace - Development Environment (cont)

- **Namespace Security is implemented via a simple PTS group naming scheme, and standard ACL entries:**

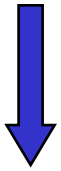
```
(wpm@zappa) fs la /ms/dev/metaproj/project
Access list for /ms/dev/metaproj/project is Normal rights:
  metaproj:dev rlidwka
  metaproj:project rlidwka
  metaproj rlidwka
  system:administrators rlidwka
  system:anyuser rl
```

In addition, VMS operations depend on similarly named groups, such as metaproj:dist and metaproj:project-dist. All groups are owned by "metaproj" by default, thus we can delegate the management of the entire metaproject to a non-administrative group.

VMS • The /ms Namespace - Development Environment (cont)

- **Clean, deterministic mapping between /ms/dev and /ms/dist pathnames.**

```
/ms/dev/metaproj/project/release/install/common
                                        exec -> .exec/@sys
                                        .exec/<sysname values>
```



```
/ms/dist/metaproj/PROJ/project/release/common
                                      exec -> .exec/@sys
                                      .exec/<sysname values>
```

VMS • The /ms Namespace - Default Symlinks

- **No, you don't need N metaproj x M project $PATH entries. One $PATH entry per metaproj is supported via sets of default symlinks.**

```
/ms/dist/somemeta/bin/someapp ->
        ../PROJ/someproj/somerelease/exec/bin/someapp
/ms/dist/somemeta/man/man1/someapp.1 ->
        ../PROJ/someproj/somerelease/common/man/man1/someapp.1
```

- **The contents of /ms/dist/somemeta are *all* symlinks of the above form, providing a composite namespace for many projects.**

- **Each project can have one and only one default release.**

VMS • Incremental/Parallel Volume Distribution

- For every distributed volume found under /ms/dist, there is a unique source (or "canonical") volume for it somewhere in the global environment. There is one copy of each distributed volume in each cell.
- e.g., cn.459.aurora, in the a.sa.ms.com cell, is the "canonical" source volume for all of the dt.460.aurora distributed volume found in every cell, mounted on /ms/dist/aurora.
- The canonical volume is incrementally dumped from and restored to the distributed volume in each cell.

VMS • Incremental/Parallel Volume Distribution (cont)

- VMS distributes up to 4 volumes at once, forking one child process per volume.
- Each volume is then dumped to a set of files
- VMS then restores to multiple cells simultaneously, again forking one child per cell.
- Note that this produces a flat, non-heirarchical distribution scheme.
- With 10 AFS cells on the other side of a single Trans-Atlantic WAN link, this is suboptimal.

VMS • Incremental/Parallel Volume Distribution (cont)

Distribution using vos dump/restore does have some problems.

- **Incremental vos restore doesn't properly remove data from RW**
- **vnode version incrementing during vos release can result in apparently stale client AFS caches. i.e., new data won't necessarily be seen after an incremental distribution.**
- **Interrupted vos restores can corrupt the RW volume and leave it offline, requiring a salvage to recover.**

PTS Database Synchronization

- **pts Wrapper Script**
  - Query commands run against the local cell (i.e. examine, membership, listowned, etc)
  - Changes are applied first to a central AFS cell in NY, then locally in the foreground, and to all remaining cells globally in the background.
  - Limitation: groups can only be created centrally by normal users (unable to specify id unless system:administrators)
- **ptsdumpd/ptssync Mechanism**
  - ptssync script runs periodically, compares PTS database in remote cells against central cell, applies necessary changes
  - ptsdumpd runs on all DB servers, accepts queries to dump entire PTS database in ASCII format

Auditing and Reporting • Cell Auditing

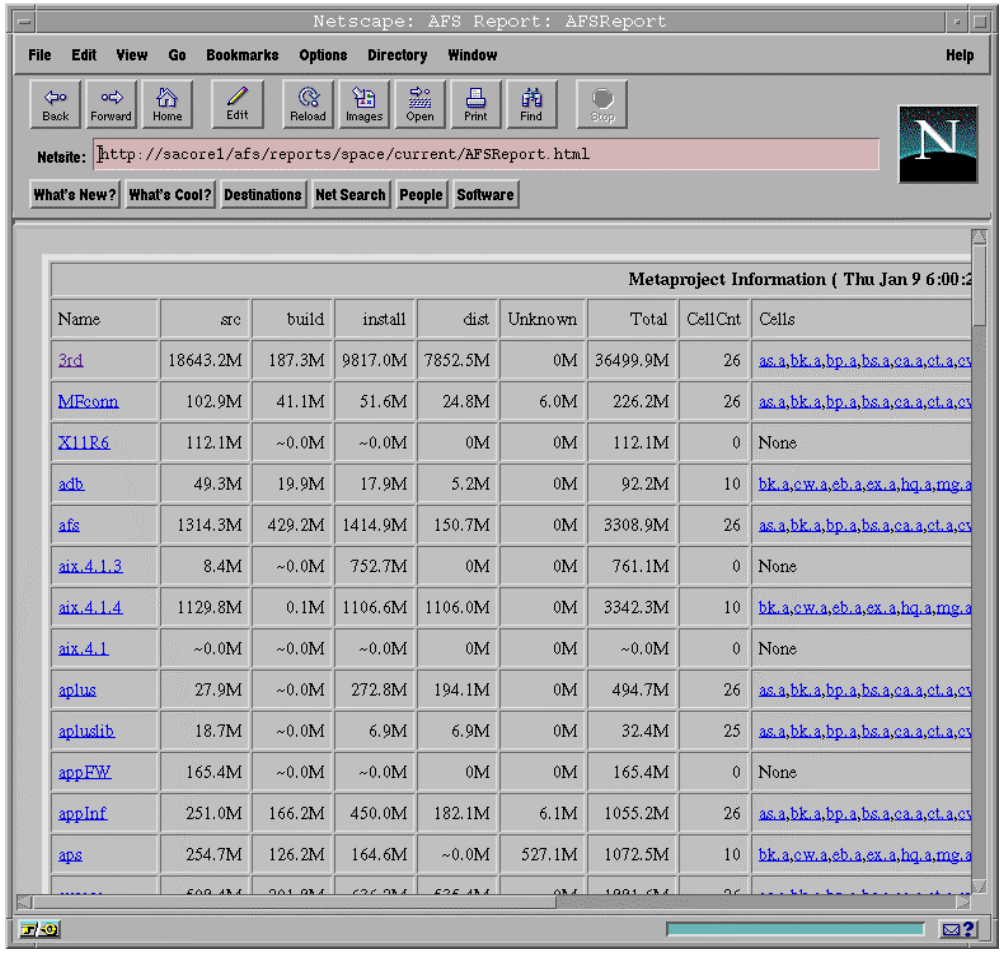
- **'bosaudit' checks the status of all the AFS database and file servers cell-wide. Some of the key auditing features include:**
  - All Ubik services have quorom, uptodate database versions, and a single Ubik sync site
  - All Encryption keys are identical
  - Consistent server CellServDB configurations
  - 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 fileservers 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)

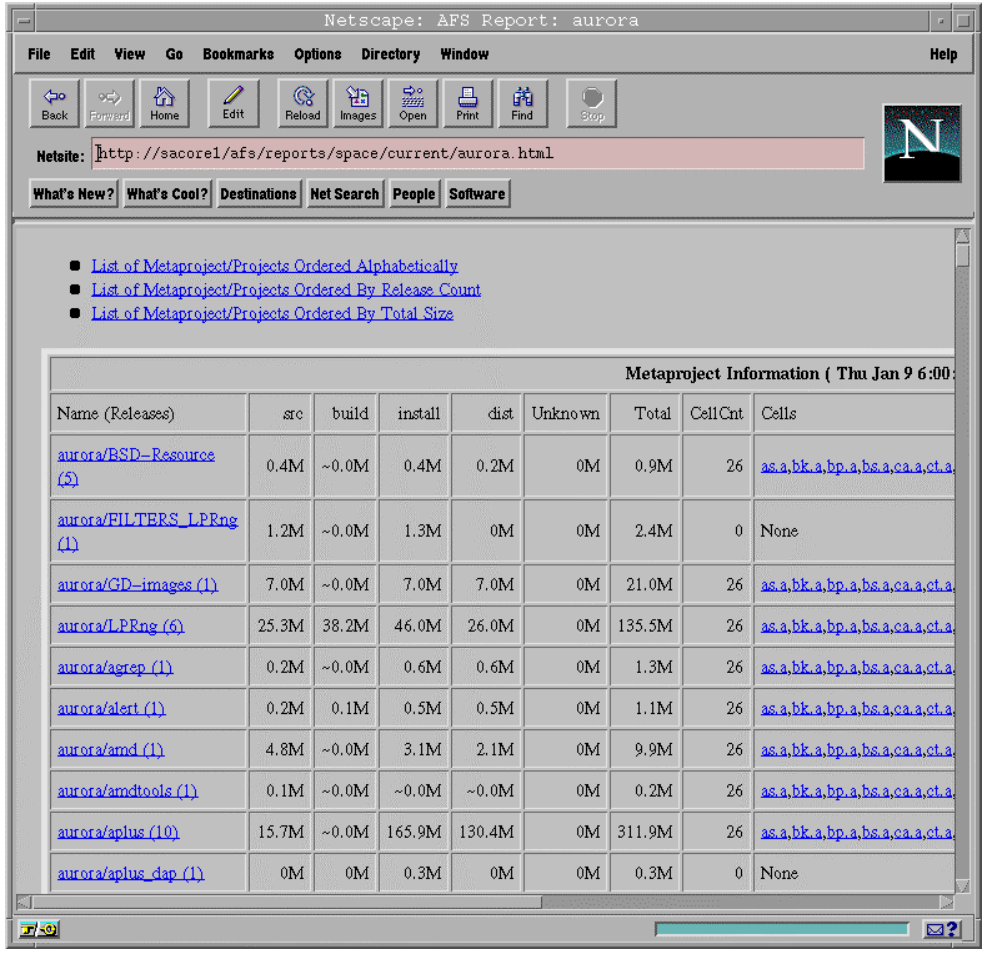
Auditing and Reporting • Metaproject Reporting



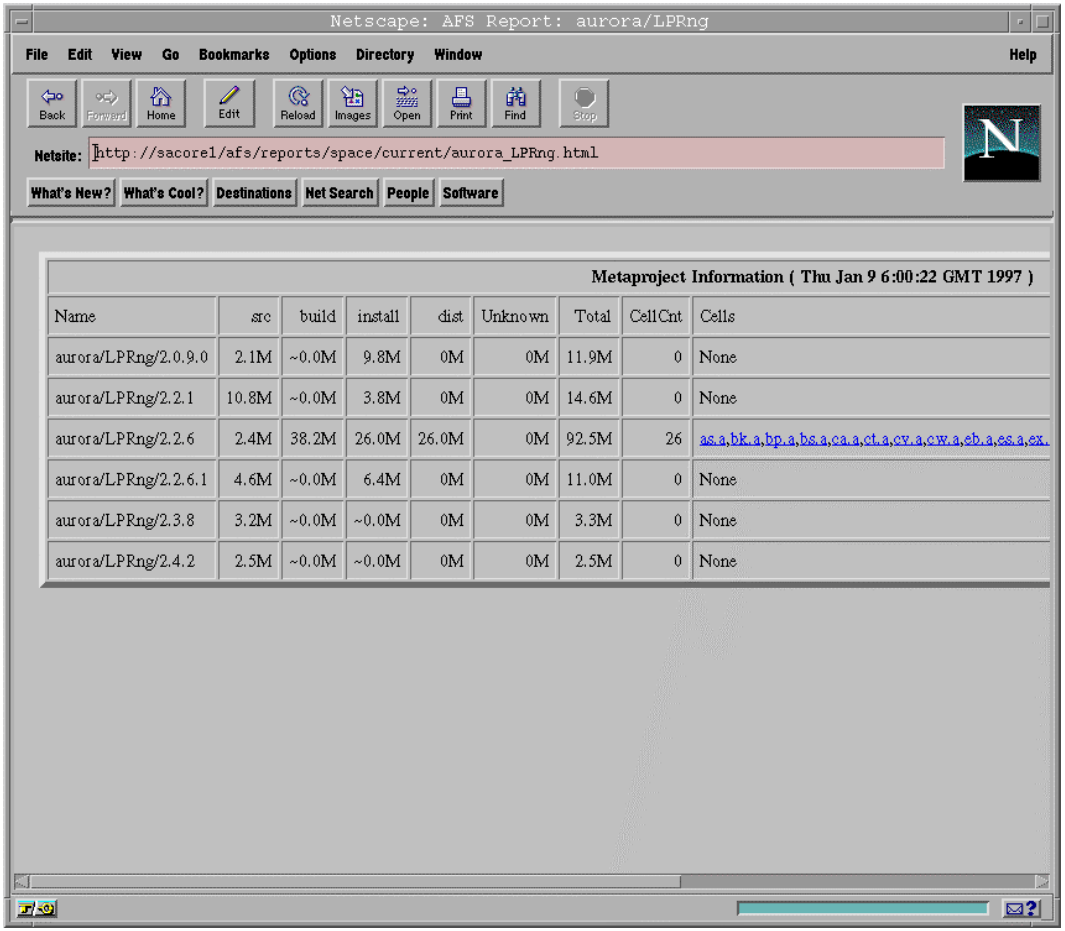
Metaproject Information (Thu Jan 9 6:00:2

Name	src	build	install	dist	Unknown	Total	CellCnt	Cells
3rd	18643.2M	187.3M	9817.0M	7852.5M	0M	36499.9M	26	as,a,bk,a,bp,a,bs,a,ca,a,ct,a,cv
MFconn	102.9M	41.1M	51.6M	24.8M	6.0M	226.2M	26	as,a,bk,a,bp,a,bs,a,ca,a,ct,a,cv
X11R6	112.1M	~0.0M	~0.0M	0M	0M	112.1M	0	None
adb	49.3M	19.9M	17.9M	5.2M	0M	92.2M	10	bk,a,cw,a,eb,a,ex,a,hq,a,mg,a
afs	1314.3M	429.2M	1414.9M	150.7M	0M	3308.9M	26	as,a,bk,a,bp,a,bs,a,ca,a,ct,a,cv
aix.4.1.3	8.4M	~0.0M	752.7M	0M	0M	761.1M	0	None
aix.4.1.4	1129.8M	0.1M	1106.6M	1106.0M	0M	3342.3M	10	bk,a,cw,a,eb,a,ex,a,hq,a,mg,a
aix.4.1	~0.0M	~0.0M	~0.0M	0M	0M	~0.0M	0	None
aplus	27.9M	~0.0M	272.8M	194.1M	0M	494.7M	26	as,a,bk,a,bp,a,bs,a,ca,a,ct,a,cv
apluslib	18.7M	~0.0M	6.9M	6.9M	0M	32.4M	25	as,a,bk,a,bp,a,bs,a,ca,a,ct,a,cv
appFW	165.4M	~0.0M	~0.0M	0M	0M	165.4M	0	None
appInf	251.0M	166.2M	450.0M	182.1M	6.1M	1055.2M	26	as,a,bk,a,bp,a,bs,a,ca,a,ct,a,cv
aps	254.7M	126.2M	164.6M	~0.0M	527.1M	1072.5M	10	bk,a,cw,a,eb,a,ex,a,hq,a,mg,a

Auditing and Reporting • Metaproject Reporting (cont)



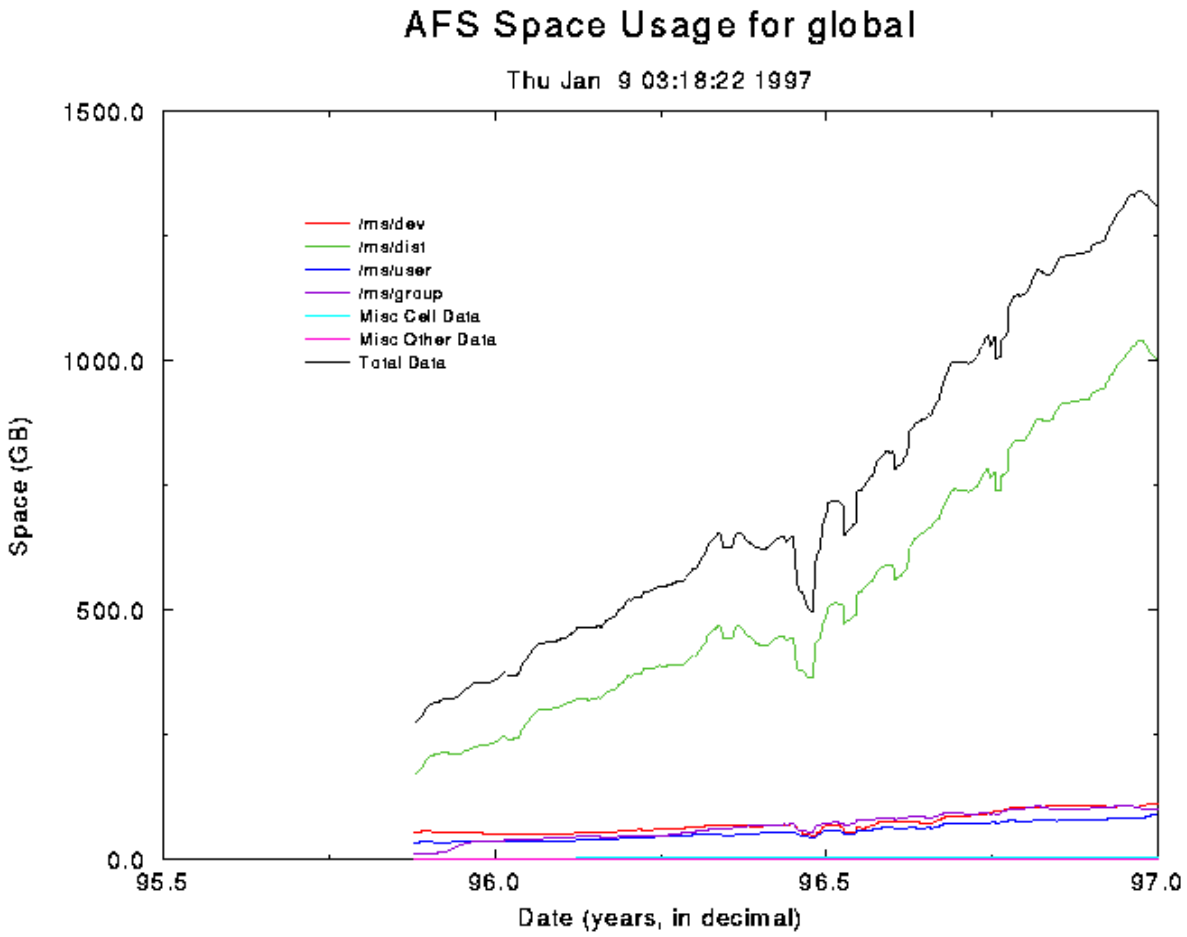
Auditing and Reporting • Metaproject Reporting (cont)



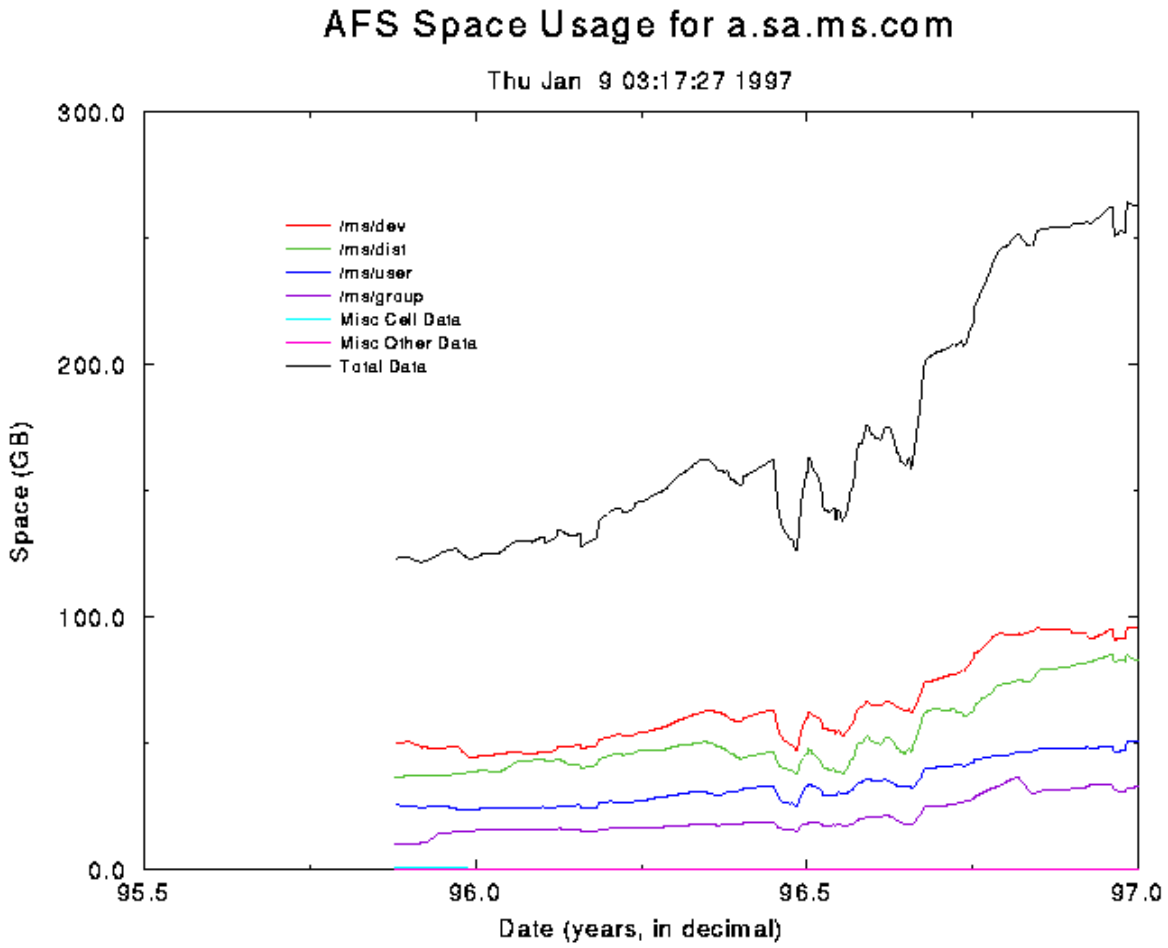
Metaproject Information (Thu Jan 9 6:00:22 GMT 1997)

Name	src	build	install	dist	Unknown	Total	CellCnt	Cells
aurora/LPRng/2.0.9.0	2.1M	~0.0M	9.8M	0M	0M	11.9M	0	None
aurora/LPRng/2.2.1	10.8M	~0.0M	3.8M	0M	0M	14.6M	0	None
aurora/LPRng/2.2.6	2.4M	38.2M	26.0M	26.0M	0M	92.5M	26	as.a,bk.a,bp.a,bs.a,ca.a,ct.a,cv.a,cw.a,eb.a,es.a,ex.
aurora/LPRng/2.2.6.1	4.6M	~0.0M	6.4M	0M	0M	11.0M	0	None
aurora/LPRng/2.3.8	3.2M	~0.0M	~0.0M	0M	0M	3.3M	0	None
aurora/LPRng/2.4.2	2.5M	~0.0M	~0.0M	0M	0M	2.5M	0	None

Auditing and Reporting • Capacity and Usage Reporting



Auditing and Reporting • Capacity and Usage Reporting (cont)



AFS Growing Pains

- Cell Wide Outages and other unpleasant disasters
- Missing Functionality (hard mounts, multihomed support)
- Limits of Scalability (e.g., Large SMP hosts)
- Limited Maneuverability (delayed deployment of new OS releases)
- Lack of a real scalable backup solution

Future Directions

- **AFS 3.5**
- **AFS on NT**
- **VMS Enhancements**
- **DCE/DFS**