



BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

AuriStor's next generation AFS and the Linux kernel AFS

Jeffrey Eric Altman
AuriStor, Inc

David Howells

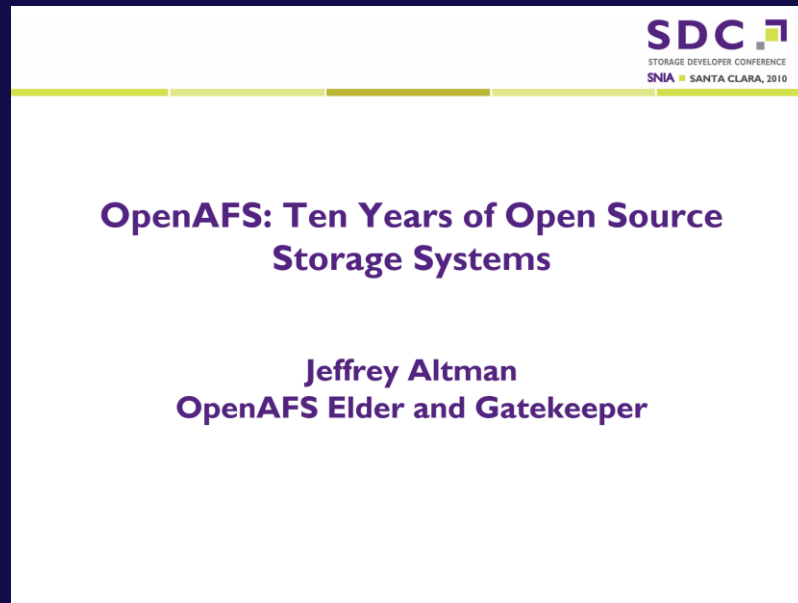


Welcome

This talk is a follow-up to the 2010 SDC talk which focused on the importance of AFS family file systems

and why they are still in use 35 years after inception and 20 years after IBM released OpenAFS 1.0 under the IBM Public License 1.0

This talk describes the future of AFS, as represented by AuriStorFS and the Linux kernel.



https://www.snia.org/sites/default/orig/sdc_archives/2010_presentations/monday/JeffreyAltman-Open_AFS.pdf

AFS Operational Goals - Review

Location Independence

Authentication, Integrity Protection, & Privacy

Geographic Replication of Critical Data

Atomic Publishing Model

One File System for all

Fine Grained Access Control

Federated Authentication

Platform specific path resolution (@sys)

Platform Architecture Independence

Distributed Administration

Still Here - Even Ten Years Later



**For the work flows that AFS
excels at there are no clear
alternatives**



Transition Costs are Huge



**Legacy deployments are
not enough**

AuriStorFS: The Next Generation AFS

Zero configuration clients

Improved Security Model

Client cache poisoning attack prevention

Performance

Scale

Enhanced File System Functionality

Out-of-the-box /afs access



Zero configuration

Cell Discovery for /afs

In modern clients, the contents of /afs can be populated on-demand via DNS SRV queries

```

root@bullfrog:/afs# ls -la
total 22
drwxr-xr-x  8 root          root    2048 Dec 31  1969 .
drwxr-xr-x 23 root          root    4096 Mar  6  2018 ..
> set type=SRV
> _afs3-vlserver._udp.your-file-system.com.
Server: router-nyc.auristor.com
Addresses: 208.125.0.254
           50.75.226.158
Aliases: nyc-router.auristor.com
_afs3-vlserver._udp.your-file-system.com SRV service location:
  priority = 10
  weight   = 0
  port     = 7003
  svr hostname = atwater-block.auristor.com
_afs3-vlserver._udp.your-file-system.com SRV service location:
1969 .@cell -> .your-file-system.com
1969 @cell -> your-file-system.com
1969 .:mount
1969 .@mount
1969 .your-file-system.com
2019 your-file-system.com

```

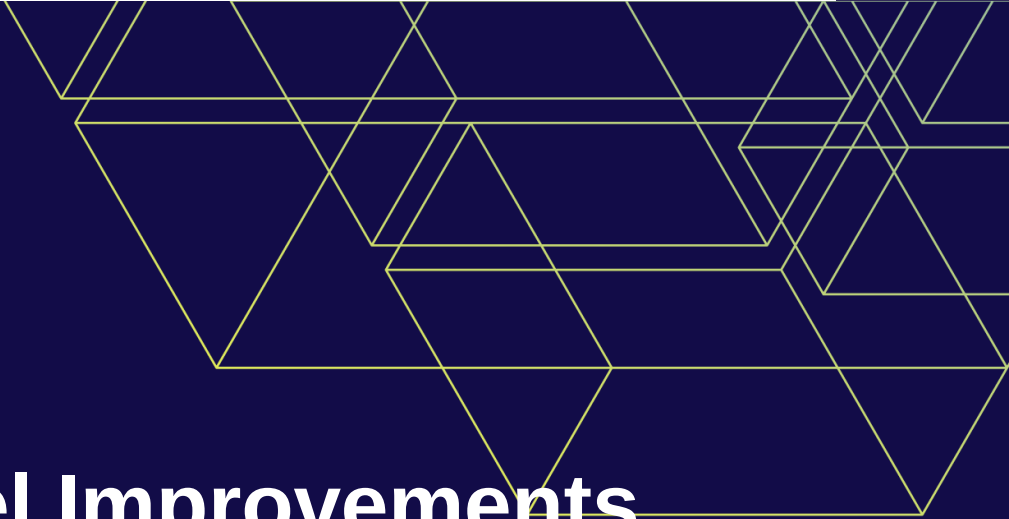
Federated Authentication

Likewise, cell mapping to authentication domains is also performed via DNS

GSS-Kerberos v5 resolves the realm from the service ticket host component

yfs-rxgk/_afs.your-file-system.com

```
> set type=TXT
> _kerberos.your-file-system.com.
Server: router-nyc.auristor.com
Addresses: 208.125.0.254
           50.75.226.158
           [redacted]
           [redacted]
kerberos.your-file-system.com text =
           [redacted]
           "YOUR-FILE-SYSTEM.COM"
>
```

Security Model Improvements

GSS-API RX Security Class



Multiple authentication services

Kerberos v5 (Active Directory)

GSS Secure Anonymous



Modern crypto (e.g. AES, Camellia)



Larger key sizes



Reduced key exposure

Rekeying

Token combining

Combined Identity Authentication

Authenticated User on an Authenticated Machine

1. `user@AD.YOUR-CELL-NAME.COM`
2. `DESKTOP$@AD-YOUR-CELL-NAME.COM`

Anonymous User on an Authenticated Machine

1. `WELLKNOWN/ANONYMOUS@WELLKNOWN:ANONYMOUS`
2. `DESKTOP$@AD-YOUR-CELL-NAME.COM`

Key Combination protects against cache poisoning attacks

https://www.auristor.com/documentation/man/linux/7/auristorfs_acls.html

User-Centric, Constrained Elevation Access Control

*Combined Identity Authentication, and
Multi-factor Access Control Lists*

- grant more rights, not fewer, when used to evaluate an ACL's Normal entries
- revoke more rights, not fewer, when used to evaluate an ACL's Negative entries

```
anonymous,machine-identity      l  
system:authuser,machine-identity lrk  
group-of-users,group-of-machines lrw
```

https://www.auristor.com/documentation/man/linux/7/auristorfs_acls.html

RX Security Class Capabilities

	yfs-rxgk	rxkad-k5	rxnull
Authentication	GSS-API	Kerberos v5	None
Integrity Protection	Yes	Yes	No
Privacy	AES256-CTS-HMAC-SHA1-96 AES256-CTS-HMAC-SHA512-384	fcrypt	No
Rekeying	Yes	No	No
Max Pkts / Call	2^64	2^30	unlimited
# identities	one or more (ordered)	one	none
Combined Key	Yes	No	No

Volume Security Policies and Maximum ACLs



AFS Volumes are relocatable, replicable object stores

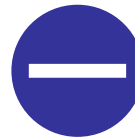


Security Policies mandate the use of

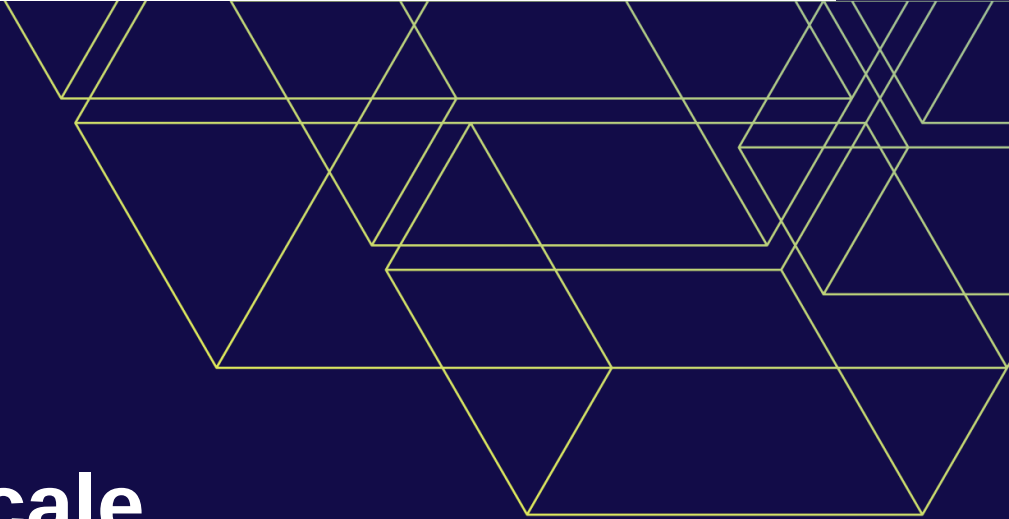
RX Security Classes, and

Data Protection levels (integrity protection / wire privacy)

Also, determine which servers are permitted to store which objects



Maximum ACLs restrict the access rights that can be granted via per-object ACLs



Scale

32-bits just isn't enough

Maximum ~2GiB or ~4GiB file size

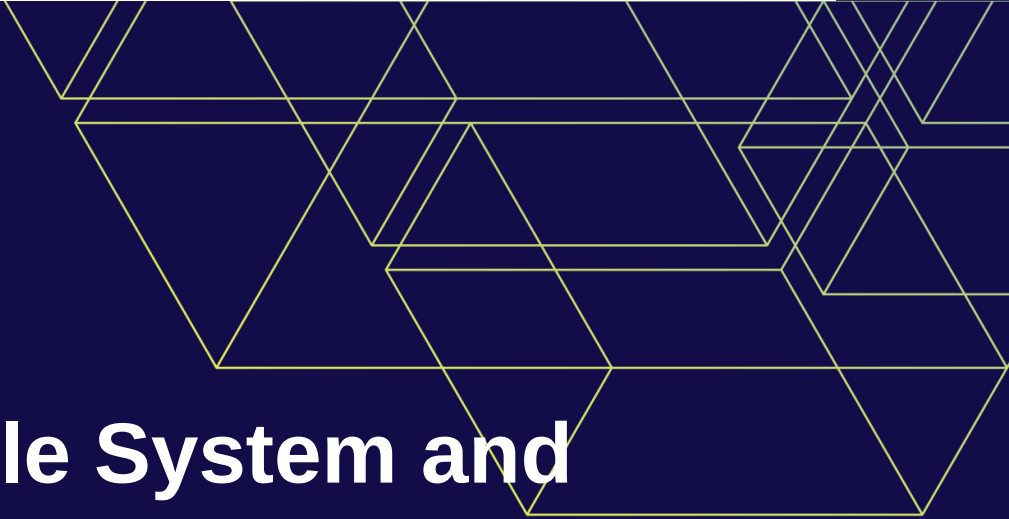
2038-problem, 2106-problem

100ns granularity to match NTFS

UNIX Epoch

Scaling to Meet Demand

	AuriStorFS	AFS
Year 2038 Ready	Yes	No
Timestamp Granularity	100ns (UNIX Epoch)	1s (UNIX Epoch)
Rx Listener Thread Throughput	<= 8.2 gbit/second	<= 2.4 gbits/second
Rx Listener Threads per Service	up to 16	1
Rx Window Size (default)	128 packets / 180.5KB	32 packets / 44KB
Rx Window Size (maximum)	65535 packets / 90.2MB	32 packets / 44KB
Rx Congestion Window Validation	Yes	No
Volume IDs per Cell	2^{64}	2^{31}
Object IDs per Volume	2^{95} directories and 2^{95} files	2^{30} directories and 2^{30} files
Objects per Volume	2^{90} directories or files	2^{26} directories or files
Objects per Directory	Up to 2,029,072	up to 64,447
Maximum Distributed DB Size	16 exabytes (2^{64} bytes)	2 gigabytes (2^{31} bytes)
Maximum Assignable Quota	16 zettabytes (2^{74} bytes)	2 terabytes (2^{41} bytes)
Maximum Reported Volume Size	16 zettabytes (2^{74} bytes)	2 terabytes (2^{41} bytes)
Maximum Volume Size	16 zettabytes (2^{74} bytes)	16 zettabytes (2^{74} bytes)
Maximum Transferable Volume Size	16 zettabytes (2^{74} bytes)	5.639 terabytes
Maximum Partition Size	16 zettabytes (2^{74} bytes)	16 zettabytes (2^{74} bytes)



Enhanced File System and Network Functionality

Networking Improvements



IPv6



Pipeline data engine

SACK based loss recovery as documented in RFC6675 with elements of

New Reno from RFC6582 on top of TCP-style congestion control elements as documented in RFC5681



Maximum window size increased to 65535 packets from 32

Filesystem Operation Enhancements

Locking

- Advisory Locks
- Mandatory Locks
- Atomic Create and Lock

Rename Variants

- Replace existing target
- NoReplace – fail if existing target
- Exchange

Append Data

- Return offset at completion

Store Data Variants

- Reserve
- Reserve and Truncate
- Reserve and ZeroFill

Server Side Operations

- Silly Rename
- Copy Object
- Copy Range
- Move Object

Client Caching and Callback



**Data Version
incremented for
each object's data
change**



**Secure Callback
notifications**

Issued to all subscribed
clients for data and metadata
change

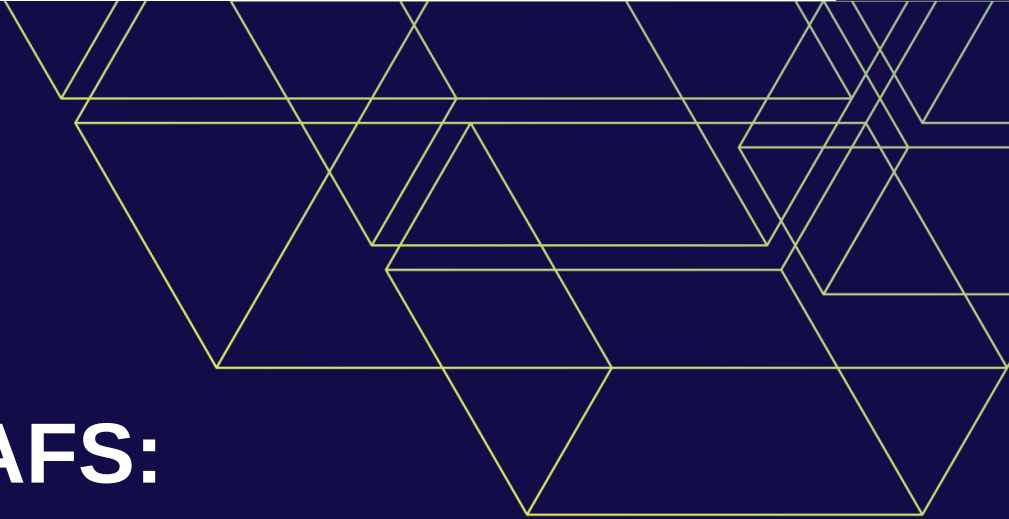
Delivered before completion
of the RPC to ensure
serialization of object
updates and subsequent
communications via other
channels



**Cached data is not
considered valid
unless an
outstanding
callback promise
exists**



**Callback promises
expire after a
period of time
determined by the
object server.**



KAFS: Out of the Box AFS

KAFS – Linux Native AFS Client



Part of the Linux kernel



Compiled as a module by Fedora, Debian

Module signed by build process



kafs-client package (rpm, deb)

Sets home cell

Handles DNS cell lookups

/afs started by *systemd* during boot

Basic Kerberos authentication

Fedora 32 Accessing /afs

```
dnf install krb5-workstation kafs-client
```

[optional] Create `/etc/kafs/client.d/defaults.conf`

- `[defaults]`
- `thiscell = <cellname>`
- `sysname = <sysname>`

```
systemctl start afs.mount
```

```
Systemctl enable afs.mount
```




Contrasting AuriStorFS and KAFS

AuriStorFS Linux Client Design

Cross Platform

Common code is shared across Linux, Solaris, AIX, macOS, and various BSD flavors; and a userland library

Monolithic kernel module

Combines the filesystem, rx network stack, and data/metadata cache management.

The internal locking model for inodes and dentries is an imperfect match for all vfs implementations.

One Mounted Device For All Volumes

The entire file namespace is exposed to the vfs as a single device traditionally mounted on /afs

KAFS Linux Client Design

Single platform

- Only exists on Linux, but will work with any arch

Modular design

- Userspace-accessible AF_RXRPC network protocol driver

- FS-Cache caching, shared with NFS, CIFS, ...

- Uses native Linux VFS locking model

Each volume has its own superblock

- Vnode IDs map to inode numbers

- df* works correctly

- Linux VFS automounter handles AFS mountpoints

- Individual volumes can be mounted anywhere



Contrasting Development Processes

Advantages of an External Project

Kernel-independent

Already ported to many systems

Features developed there

Faster rollout

Point of focus for developers interested in
AFS

Disadvantages of Being Out-of-tree

Kernel interfaces change often

Parts of kernel API not accessible to non-GPL

Massive continuous building effort to support many kernels across distributions

User must find and install the packages

Advantages of Being In-Kernel (1)

Kernel-reserved interfaces

RCU, container features

Shared components

FS-Cache, keyrings

Crypto

Tracepoints

Debugging facilities

Advantages of Being In-Kernel (2)

Developers making big changes in-kernel must make the changes throughout

Automated testing

Distributions build the filesystem modules with distribution kernels

Much more widely available

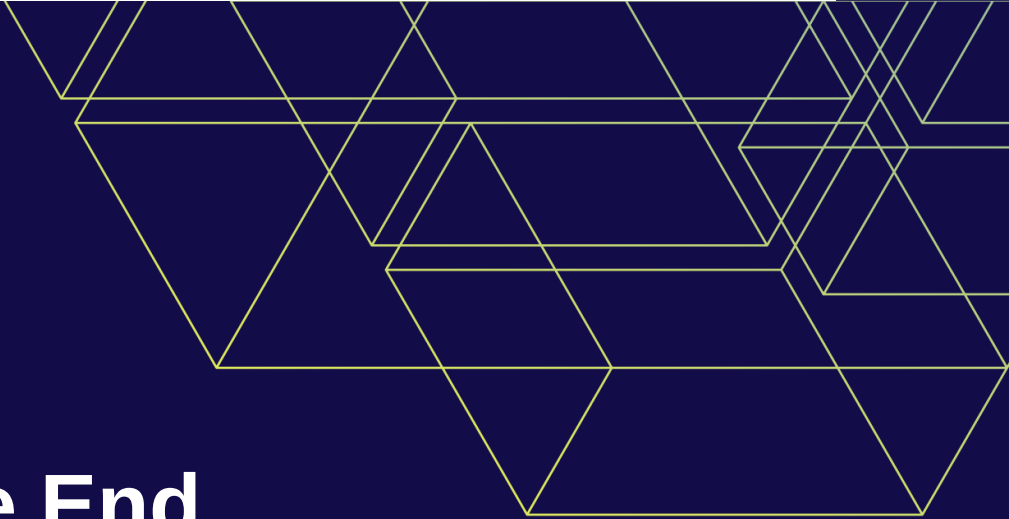
Note: Not in any enterprise distros yet

Future KAFS Developments

Direct-I/O local caching

Per-container keyring

Advanced AuriStorFS features



The End

References

<https://www.auristor.com/filesystem/>

<https://www.infradead.org/~dhowells/kafs/>

https://www.auristor.com/documentation/man/linux/7/auristorfs_acs.html

<https://tools.ietf.org/html/draft-wilkinson-afs3-rxgk>

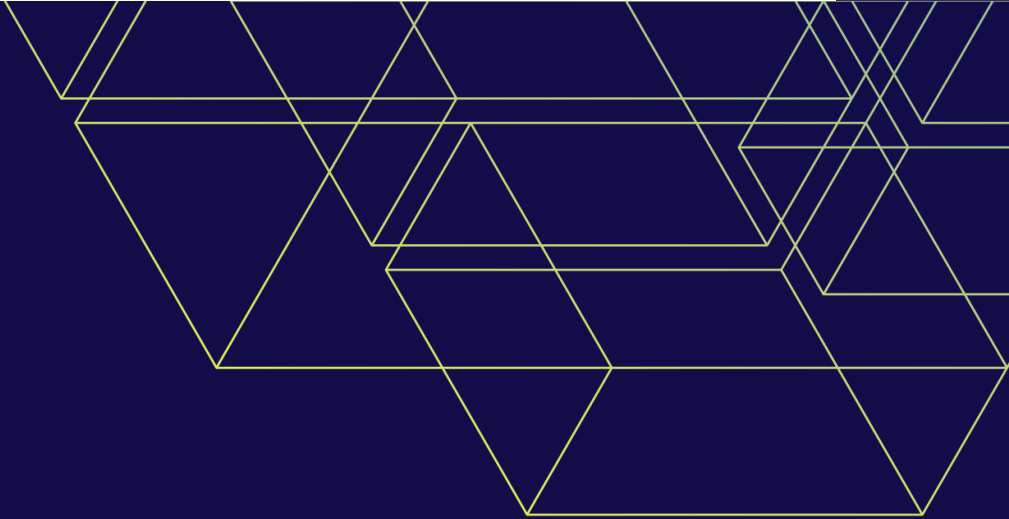
<https://tools.ietf.org/html/draft-wilkinson-afs3-rxgk-afs>

<https://tools.ietf.org/html/draft-howard-gss-sanon>

Contact Info

David Howells
dhowells at redhat dot com

Jeffrey Altman
jaltman at auristor dot com



**Please take a moment
to rate this session.**

Your feedback matters to us.