

2024 OpenAFS Workshop

Toward an OpenAFS Roadmap

June 11, 2024



About me

Michael Meffie

Principal Software Engineer

Sine Nomine Associates

Commercial support for OpenAFS

mmeffie@sinenomine.net

Table of Contents

1. | Project Goals

We will review our project goals
and priorities.

Table of Contents

1. | Project Goals

We will review our project goals and priorities.

2. | Near Term: 6-12 Months

The next few months we will be focusing on integrating the backlog of changes already in code review.

Table of Contents

1. | Project Goals

We will review our project goals and priorities.

2. | Near Term: 6-12 Months

The next few months we will be focusing on integrating the backlog of changes already in code review.

3. | Med Term: 12-24 Months

In the medium term, we would like to complete larger projects already underway and need refocus.

Table of Contents

1. | Project Goals

We will review our project goals and priorities.

2. | Near Term: 6-12 Months

The next few months we will be focusing on integrating the backlog of changes already in code review.

3. | Med Term: 12-24 Months

In the medium term, we would like to complete larger projects already underway and need refocus.

4. | Discussion

What are the things on your wish list?

01

Project Goals

Reviewing our general goals

We will review our project goals and priorities in this section.



Security

Addressing vulnerabilities is our highest priority.

The OpenAFS code base has a long history and many legacy features.

Improvements have been made over the years, but much more will be done modernize and harden the codebase.

Objectives

Unsafe interfaces will be hardened or disabled by default.

Correcting unsafe threading and memory management.

Obsolete and unused code will be removed at a faster pace.

Applying principles of least privilege, such as non-root file servers.

Reliability

Correcting and reducing bugs is a high priority.

The OpenAFS code base has a long history and many legacy features.

Improvements have been made over the years, but much more will be done modernize and harden the codebase.

Objectives

Obsolete and unused code will be removed at a faster pace.

Clear the outstanding change and ticket backlog.

Performance

Performance and scalability changes need to be a focus.

The OpenAFS code base has a long history. Many of the limits are due to outdated assumptions.

Objectives

Modernize internal hashing in the file server and the cache manager.

Modernize database storage to vastly improve throughput.

Remove networking bottlenecks without breaking compatibility.

Usability

Improving the experience without compromising compatibility.

The OpenAFS code base has a long history.

Objectives

Modernize documentation for setup and usage.

Improve deployment experience.

Improve logging and error messages.

Introduce new administration commands.




02

Near term

**Objectives for the next 6
to 12 months**

Our next few months will be focusing
on integrating the backlog of changes
already in code review.



Security

More frequent Security Releases.

Rx rxgk security class phase 2 for strong encryption on the wire for authenticated connections (rxkad replacement only)

Audit code and tickets to identify and resolve vulnerabilities.

Restrict **fs flushall** to root

Resolve static analysis warnings.

Continue to remove obsolete code and fix compiler warnings.

Reliability

Fix more rx connection object leaks.

Improve **vos restore** robustness by fixing start and end dump tag checks and other related issues.

Update integration of the **cache manager** with **systemd** for more reliable startup and shutdown processes.

Databases: Various fixes and improvements.

Unit tests: TAP Tests for file and database servers.

Performance

Cache Manager: Use DV hints for dentry re-validation optimization (e.g., improves performance when doing builds in /afs)

Rx: Multiple listener threads

Ubik: Avoid delaying database reads during internal operations.

Fileserver: Improved hashing for large number of callbacks

Usability

Day one support for new Linux and macOS versions.

Integrate FreeBSD and AIX contributions.

Features for containerization (@sys resolution)

macOS prepane updates.

Name pipe audit logging interface.



02

Medium Term




03

Medium Term

**Development of new code in the
next 12 to 24 months**

Some projects are already under way
in either gerrit or public github repos.

We should refocus our efforts on them
after more urgent tasks are completed.



Fileserver Snapshot

Suspend a fileserver to snapshot all of the vice partitions.

To take advantage of filesystems that provide snapshots, such as ZFS.

Targeted to read-only heavy workloads.

Performance and Scale

Improved DR on platforms that support snapshots.

Administration

Includes new tooling for managing the fileserver freeze.

Ubik Key-Value

Leverage modern Key-Value storage for database backend

LMDB can replace the legacy on-disk hashes while retaining the Ubik protocol for database replication.

New database datatypes will be needed to support IPv4 and Rxgk features.

Performance and Scale

Initial testing shows large performance gains and high requests per second.

Extensibility

The new key-value scheme allows for new data types to be stored (still limited by RPCs)

Administration

Includes new tooling for database conversions, up and down.

IPv6 Support

Complete support for IPv6 servers and clients.

Much more work needs to be done to prepare the code base for IPv6.

Most of the work is IPv4 to UUID server id conversion.

Remove obsolete IPv4 code

Older IPv4 code is deprecated and will be slated to be removed.

Convert Server Numbers

Server IDs will be migrated away from IPv4 32 bit numbers.

Generic Socket Address

Internal overhaul to change out 32-bit addresses with generic addresses.

04

Discussion

What are the items on your wish list?

What would be useful and important at your site?



Thank you

Michael Meffie
mmeffie@sinenomine.net