GSoC 2022: OpenAFS

Project: Per volume reverse name lookup index

Mentors: Andrew Deason, Michael Meffie, Mark Vitale, Benjamin Kaduk
Contributor: Vikramraj Sitpal

Overview

OpenAFS is a distributed file system. Currently, it has "File ID" or FID to "entry" or name lookup functionality, AKA, reverse lookup and also normal lookup, which is “name” for a given FID. Inverse Lookup is currently computationally expensive because the algorithm scans a hash index data structure embedded within a "Directory object" (which has key:name, value: FID) looking for a name that matches the given FID. It is not performant as it's merely brute-forcing the search.

So my code changes add a completely new feature, a key-value database built on top of LMDB, and along with it low-level unit tests, and overall feature tests using the Robotest framework.

Project

Three GitHub repositories have my work:

1. Feature Code: https://github.com/vikramrajsitpal/openafs/tree/gsoc-22-ri
    a. Tag (3 commits starting this tag): **gsoc-2022-ri-final**
2. Test Library Code:
    a. Repo: https://github.com/vikramrajsitpal/robotframework-openafslibrary/tree/gsoc-ri
    b. PRs:
        i. https://github.com/openafs-contrib/robotframework-openafslibrary/pull/3
        ii. https://github.com/openafs-contrib/robotframework-openafslibrary/pull/4
3. Test scripts:
    a. Repo: https://github.com/vikramrajsitpal/openafs-robotest/tree/gsoc-ri
    b. PR: https://github.com/openafs-contrib/openafs-robotest/pull/31

Future Work

All thanks to Andrew Deason for giving me some pointers for future work of this project during the initial project design phase.

1. Bulk convert entire volumes at once: The base project is just to handle the incremental create/rename/delete operations in the DB. But we'd also need to be able to populate the whole database of a pre-existing volume. One could start off by writing this as a standalone tool that gets its data from the *volscan* utility maybe. Ideally, we could have something run in the *fileserver* or *volserver* that would convert volumes "on-line", but that's harder to do.
2. Copying Reverse Index DB along with volumes: OpenAFS has a lot of functionality for moving volumes around on different servers, and cloning volumes and maintaining RO copies of volumes and such. We've been ignoring that when talking about the base project, but we could try to figure out how to make the FID->name work with this. (That is, the DB gets moved along with the volume data when the volume is moved, cloned, etc.)
3. Add OpenAFS client code to use the new RPC: The project adds a new RPC for inverse lookup, but it uses a bare-bones Python script to call the RPC, not OpenAFS client.