

Redefining CephFS bridge with the new VFS module for Ceph

`vfs_ceph_new`

Anoop C S

IBM / Samba Team

April 8, 2025



Agenda

1. Overview
2. LibCephFS
3. The design
4. Testbed
5. Conclusion
6. Future
7. Q&A

Agenda

1. Overview
2. LibCephFS
3. The design
4. Testbed
5. Conclusion
6. Future
7. Q&A

- Free software *re*-implementation of **SMB** networking protocol
- File and print services and more...
- Integration of AD (Active Directory)
 - DC (Domain Controller)
 - Domain member
- Stackable **VFS** interface
 - Local - btrfs, ext4, xfs etc.
 - Clustered - cephfs, glusterfs, gpfs etc.
- Modern **VFS**
 - File handles
 - Path reference `fsp`
 - `O_PATH` in Linux

Ceph and CephFS

- Unified system for object, block and **file** storage
- Highly reliable, easy to manage and free
- CephFS
 - **POSIX** complaint file system
 - Built on top of Ceph's distributed object store **RADOS**
 - **MDS** (MetaData Server)
 - Data and Metadata pools
- Convenient client facing libraries

Agenda

1. Overview
2. LibCephFS
3. The design
4. Testbed
5. Conclusion
6. Future
7. Q&A

Userspace CephFS API

- `libcephfs` - native C API
- Easy to use
- Correspondence to standard file systems calls
 - `ceph_openat()` = `openat(2)`
 - `ceph_readdir()` = `readdir(2)`
- Included as `libcephfs.h`
 - Versioned
 - Provides *pkgconfig* file: `cephfs.pc`
- API behaviour changes not properly versioned

`libcephfs` versioning

Currently versioned at **2.0.0**

Two variants for LibCephFS

High level API

- Most commonly used
- Matches standard function signatures
- **Path** and **File descriptor** based
- Support for *at() calls

Examples

```
ceph_openat(), ceph_close(),  
ceph_getxattr()
```

Low level API

- More fine grained
- Notion of **inodes** and **file handles**
- Support for user permissions (per call) and supplementary groups
- **Asynchronous IO**

Examples

```
ceph_ll_lookup(), ceph_ll_open(),  
ceph_ll_put()
```


Agenda

1. Overview
2. LibCephFS
3. **The design**
4. Testbed
5. Conclusion
6. Future
7. Q&A

Previously..

- Existing vfs module for ceph: `vfs_ceph`
 - Long standing and it works !
- Source: `source3/modules/vfs_ceph.c`
- Consumes **high level** `libcephfs` APIs
- Implements most of the `SMB_VFS_XXXX` interfaces
 - Including path based fallback mechanism
- Statically linked against `libcephfs.so`
- Mandatory to be loaded as the **last module** in `vfs` objects list
- File descriptors maintained on `libcephfs` client side
 - fds are passed over as it is to upper layers

The trigger

- [BUG 14053](#) *vfs_ceph (libcephfs) denies access when permitted via supplementary group membership*
 - [Shachar Sharon](#) created [MR 3466](#)
 - Proposed changes addressed the direct issue
 - But the ability for `smbd` to switch credentials was implicitly ignored
- Further discussions and looking for alternatives

Thoughts outlined

1. Shift to **low level** `libcephfs` APIs
2. Additionally **optimize** CPU and memory consumption due to separate independent `libcephfs` client stacks

A name?

- Should we make changes to `vfs_ceph`?
 - not minimally invasive
- Separate module? perfect.
- What about a name? uh-oh !
 - `vfs_ceph_ll`
 - `vfs_ceph2`
 - `vfs_ceph_experimental`
 - `vfs_ceph_test`
 - `vfs_ceph_next`
 - `vfs_ceph_ng`
- Finally settled on `vfs_ceph_new`

vfs_ceph_new structure

- Source: `source3/modules/vfs_ceph_new.c`
- Initially crafted out of `vfs_ceph` [MR 3718](#)
- CephFS client initialization with locally **cached mounts**
 - Helpful for access to multiple shares configured with same ceph user and underlying ceph file system from a unique client.
- Operations on **inodes** via **file handles**
 - Leverages `vfs_fsp` extensions to hold `struct vfs_ceph_fh`
 - Proactively adds credentials using `get_current_utok()`
- No longer statically linked against `libcephfs.so`

struct vfs_ceph_fh

```
/* Ceph file-handles via fsp-extension */  
struct vfs_ceph_fh {  
    struct vfs_ceph_dirp dirp;  
    struct cephmount_cached *cme;  
    struct UserPerm *uperm;  
    struct files_struct *fsp;  
    struct vfs_ceph_config *config;  
    struct vfs_ceph_iref iref;  
    struct Fh *fh;  
    struct dirent *de;  
    int fd;  
    int o_flags;  
};
```

Expanding further

- Introduction of **proxy** mode [MR 3792](#)
 - Structural reorganization, new `struct vfs_ceph_config`
 - Abstraction of all module specifications (both internal and external)
 - Dynamic loading of libraries (`libcephfs.so` or `libcephfs_proxy.so`)
- Switch to use `ceph_readdir_r()` from `ceph_readdir()` [MR 3833](#)
- Making use of low level **non blocking** API for asynchronous IO [MR 3857](#)
 - Detection of API during compile/build time
 - `struct tevent_threaded_context` added
- Taking advantage of **case insensitive** CephFS subvolumes [MR 3992](#)
 - Identify and conditionally add `FILE_CASE_SENSITIVE_SEARCH` flag

struct vfs_ceph_config

```
struct vfs_ceph_config {
    #if HAVE_CEPH_ASYNCIO
        struct tevent_threaded_context *tctx;
    #endif
    const char *conf_file;
    const char *user_id;
    const char *fsname;
    struct cephmount_cached *mount_entry;
    struct ceph_mount_info *mount;
    enum vfs_cephfs_proxy_mode proxy;
    void *libhandle;
    uint32_t capabilities;
    // ...
};
```


Agenda

1. Overview
2. LibCephFS
3. The design
4. **Testbed**
5. Conclusion
6. Future
7. Q&A

- Modular ansible based testing framework
 - Samba Integration Testing (SIT)
 - Refer [sambaXP 2024 talk](#)
- Possible integration for Samba GitLab CI?
- Includes a subset of core *smb2* torture test suites
- Various share combinations for CephFS backend
 - ceph kernel client based shares
 - `vfs_ceph`
 - `vfs_ceph_new`
 - `vfs_ceph_new` under proxy mode

GitLab integration (projected view)

vfs_ceph_snapshots: Prepend connectpath to full snapshot path from dirfsp

Edit Code

Open Anoop C S requested to merge [anoopcs-vfs-ceph-snapshots...](#) into [master](#) 14 hours ago

Overview 0 Commits 2 Pipelines 1 Changes 1

Add a to-do item

There is a situation with `ceph_snap_gmt_convert_dir()` where we might end up opening the wrong snapdir for reading the snapshots. This is due to the fact that we calculate the parent as empty string via `ceph_snap_get_parent_path()` for a file inside a directory under share root. In addition to that we follow the logic from `shadow_copy2` module to return the converted snapshot path corresponding to file share's root.

BUG: https://bugzilla.samba.org/show_bug.cgi?id=15819

0 0

Pipeline #1745211649 failed

Pipeline failed for [8a935339](#) on [anoopcs-vfs-ceph-sn...](#) 10 hours ago

Download

Approve Approval is optional

Ready to merge!

Delete source branch Squash commits Edit commit message

• 2 commits and 1 merge commit will be added to master.

Merge...

0 Assignees

None - assign yourself

Edit

0 Reviewers

None - assign yourself

Edit

Labels

ci/cephfs

Edit

Stage: external

Failed jobs

ci/cephfs

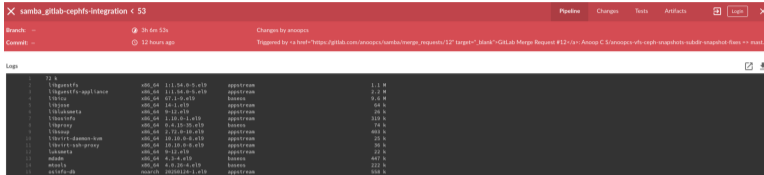
spent

1 Participant



Sample MR

Jenkins redirected (projected view)



samba_gitlab-cephfs-integration < 53

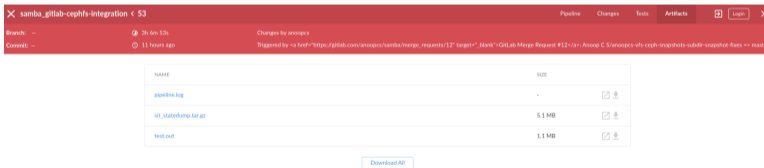
Branch: -- 2h 4m 53s Changes by anooops

Commit: -- 12 hours ago Triggered by ra hael~https://gitlab.com/anooops/samba/merge_requests/12/target*, Nank~GitLab Merge Request #12~/ra~ Anoop C S/anooops-vm-ceph-snapshots-subdir-snapshot-files == mast...

Logs

Step	Duration	Start Time	End Time	Agent	Status
1	72 s				
1	1h00m53s	000_04	1:1:04.0-0-0-10	appotriam	1:1 M
2	1h00m53s	000_04	1:1:04.0-0-0-10	appotriam	2:2 M
3	1h00m53s	000_04	07:1:0-w50	boson	9:0 M
4	1h00m53s	000_04	18-1-e10	appotriam	66 s
5	1h00m53s	000_04	9-10-e10	appotriam	26 s
6	1h00m53s	000_04	1:10-00-0-0-10	appotriam	310 s
7	1h00m53s	000_04	0-0-15-00-0-10	boson	76 s
8	1h00m53s	000_04	2:72-0-10-0-10	appotriam	408 s
9	1h00m53s	000_04	10:00-0-0-0-10	appotriam	26 s
10	1h00m53s	000_04	10:10-0-0-0-10	appotriam	56 s
11	1h00m53s	000_04	9-10-e10	appotriam	22 s
12	1h00m53s	000_04	4-2-0-0-10	boson	447 s
13	1h00m53s	000_04	4-0-20-0-0-10	boson	222 s
14	1h00m53s	000_04	20200120-1-e10	appotriam	500 s

Jenkins job landing page



samba_gitlab-cephfs-integration < 53

Branch: -- 2h 4m 53s Changes by anooops

Commit: -- 11 hours ago Triggered by ra hael~https://gitlab.com/anooops/samba/merge_requests/12/target*, Nank~GitLab Merge Request #12~/ra~ Anoop C S/anooops-vm-ceph-snapshots-subdir-snapshot-files == mast...

NAME	SIZE	Actions
pipeline.log	-	📄 ⬇
uit_statecomp.tar.gz	5.1 MB	📄 ⬇
test.out	1.1 MB	📄 ⬇

Download All

Jenkins job artifacts page

Agenda

1. Overview
2. LibCephFS
3. The design
4. Testbed
5. **Conclusion**
6. Future
7. Q&A

Functionality wise

- Test failures especially smbtoriture failures
- Interesting ones
 1. smb2.charset test suite failure [BUG 15716](#)
 - Caused by *re-organization* around retrieving FS capabilities
 2. Ceph MDS crash [tracker 69059](#)
 - Somehow triggered a different code path by *smbd* cleanups/changes
 3. LibCephFS crash [tracker 69624](#)
 - Related to snapshot lookups when stacked with *vfs_ceph_snapshots*
 4. smb2.rw.rw1 torture test failure [tracker 70726](#)
 - Incorrect behaviour during asynchronous IO, still open

Note

smbtoriture was found to be a good bug detector !

Case (in)sensitive CephFS subvolumes

What is the difference?

Server versions: [Samba 4.21.4](#) + [Ceph 19.2.1](#)

Case sensitive	Latency	Throughput	Ops rate
<i>Enabled</i>	12.637 ms	2189.222 KB/s	316.540 ops/s
<i>Disabled</i>	12.261 ms	2256.935 KB/s	326.219 ops/s

Workload: [SWBUILD](#)

Note

- Benchmark utility: [SPECstorage](#)
- Case sensitivity set at subvolume level using `ceph fs subvolume charmap set`

What about performance?

Server versions: [Samba 4.21.4](#) + [Ceph 19.2.1](#)

IO mode	Latency	Throughput	Ops rate
<i>Sync IO</i>	12.261 ms	2256.935 KB/s	326.219 ops/s
<i>Async IO</i>	11.877 ms	2332.388 KB/s	336.736 ops/s

Workload: [SWBUILD](#)

Note

- Benchmark utility: [SPECstorage](#)
- Backed by [case insensitive](#) CephFS subvolume
- `aio read size` and `aio write size` parameters from `smb.conf` controls IO mode

Agenda

1. Overview
2. LibCephFS
3. The design
4. Testbed
5. Conclusion
6. **Future**
7. Q&A

- Optimized caching of ceph mounts
 - Integrate struct `cephmount_cached` with **memcache**?
- **Zero-copy** interface
 - Under active development [PR 62003](#)
- Profile memory usage, probably detect leaks
- Optional **extra pipeline** job for upstream GitLab MRs
 - Run tests on multiple share configurations backed by CephFS
 - Propose to `samba-technical`

Agenda

1. Overview
2. LibCephFS
3. The design
4. Testbed
5. Conclusion
6. Future
7. Q&A

Thank you

Anoop C S

anoopcs@samba.org

Questions?