



Updates on distributed file system access via the new VFS

Anoop C S

sambaXP 2023



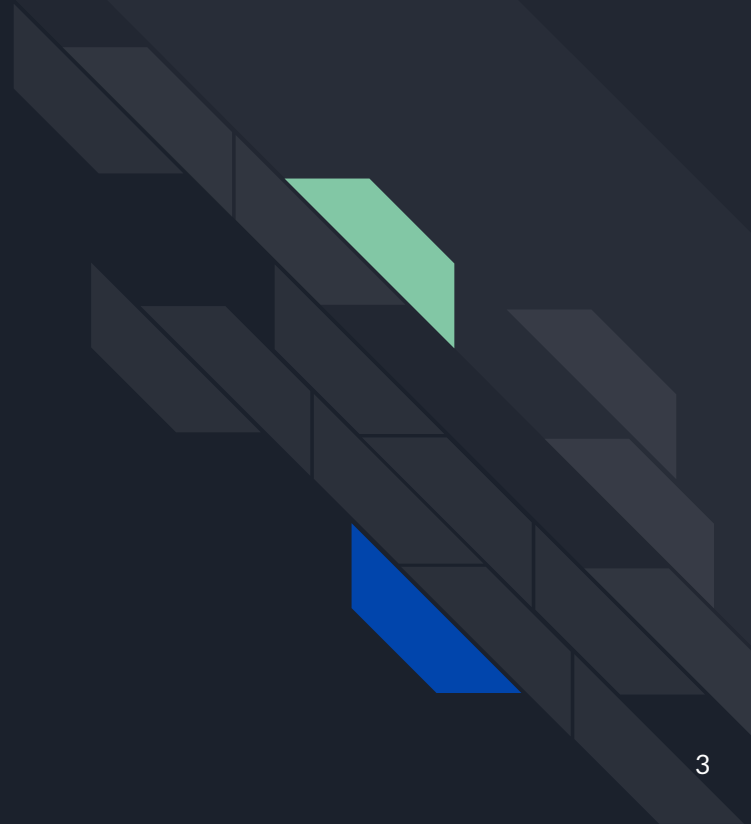
Agenda

sambaXP 2023

- Samba and new VFS
- Distributed File system interaction
- *at () variant compliance
- Performance impact
- Integration testing
- Takeaways



The new VFS(recap)



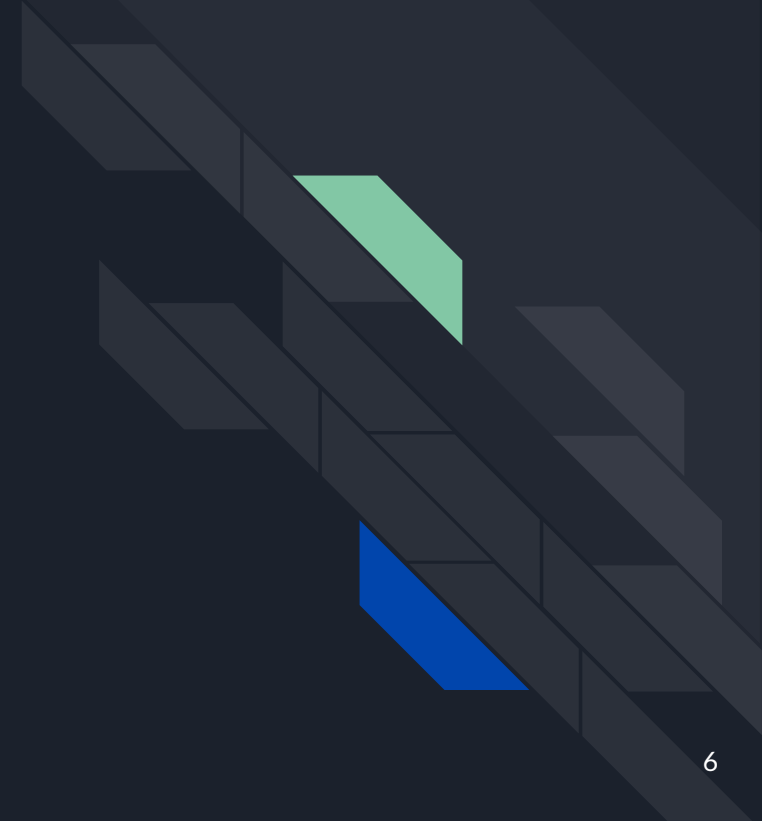
VFS all this while

- A mechanism to extend the functionality of Samba
- Completely based on paths
 - pervasive use of paths on SMB1
- Path processing
 - complex and slow
- Internal file handle structure `struct file_struct{}`
 - `SMB_VFS_CREATE_FILE()`

Secure, Enhanced and Modern VFS

- SMB1 is deprecated and disabled by default
- SMB2+ is purely handle based
- New helper function which is called `openat_pathref_fsp()`
 - skips NTFS emulation
 - handle restricted internally for path reference
- `O_PATH` instead of plain `O_RDONLY`
 - limited number of syscalls
 - can read but not modify inode metadata
- Use of `*at()` variants
- Fallback mechanisms

Distributed File System access with GlusterFS



GlusterFS: in a nutshell

- Scalable distributed network file system in user space
- Runs on any file system with extended attributes support
- Replication, quotas, snapshots, geo replication and more
- Bricks - building block for a volume
- Volume types
 - *Distributed*
 - *Arbiter*
 - *Disperse*
 - *Replicated*
- Supports access via NFS and SMB
- CTDB for HA with shared volume

Samba integration

vfs_glusterfs

- consumes libgfapi
- `glfd` and `fsp` extension
- basic configurations
 - volume name
 - path
 - log level, log location
- samba VFS compatible APIs
- additional logic
- memory consumption

vfs_glusterfs_fuse

- works on local FUSE mount
- similar to local file system access
- no additional configurations

Big brother `openat()`
with others



*at () calls from *libgfapi*

- Available with v11
- Implemented with basic test cases
- one-to-one correspondence
- Conditionally included within `vfs_glusterfs`
 - `#ifdef HAVE_GFAPI_VER_7_11`

Changes at `vfs_glusterfs`

- `source3/modules/vfs_glusterfs.c`
- Consume `glfs_*at()` calls in `vfs_glusterfs` [#2682](#)
- `vfs_gluster_openat()`
 - invoking `glfs_openat()`
 - `if` conditions
 - `O_PATH`
 - `O_CREATE`
 - backward compatibility
- Fetching parent `glfd`
 - `mkdirat()`, `renameat()`, `linkat()`, `fstatat()` etc..
- Discover and modify handle based calls
 - `fstat()`, `fchmod()`, `fsetxattr()` etc..

Bugs, fixes and improvements

- `SMB_VFS_GET_REAL_FILENAME` [#2524](#)
 - well before `glfs_*at()` APIs were introduced
 - fixed with older full name path construction
- `O_CREAT` special case [#2743](#)
 - bug inside `glfs_openat()` [#3838](#)
 - failure to link inode in file create
- Series of `EBADF` errors ! [#2745](#)
 - again `smb_vfs_get_real_filename`
 - now that we use `glfs_openat()` , back to handle based approach
 - `SMB_VFS_GETXATTR`
 - differentiate the usage of `glfs_getxattr()` and `glfs_fgetxattr()`
 - `SMB_VFS_FNTIMES` [#2748](#)

Operational impact



Performance

- Use cases
 - Creation of large number of files (10k empty normal files)
 - Listing large directory (10k empty normal files)
- Versions
 - GlusterFS: latest `master`
 - Samba: `4.18.2`, `4.17.7`, `4.16.10`, `4.15.13` and `4.14.14`
- Type of volume
 - Distributed-Replicated with Arbiter ($2 \times (2 + 1) = 6$)
- Number of nodes
 - 2 nodes: non-clustered Samba
- Client
 - `smbclient`
- Statistics tool
 - `smbcontrol` and `smbstatus`

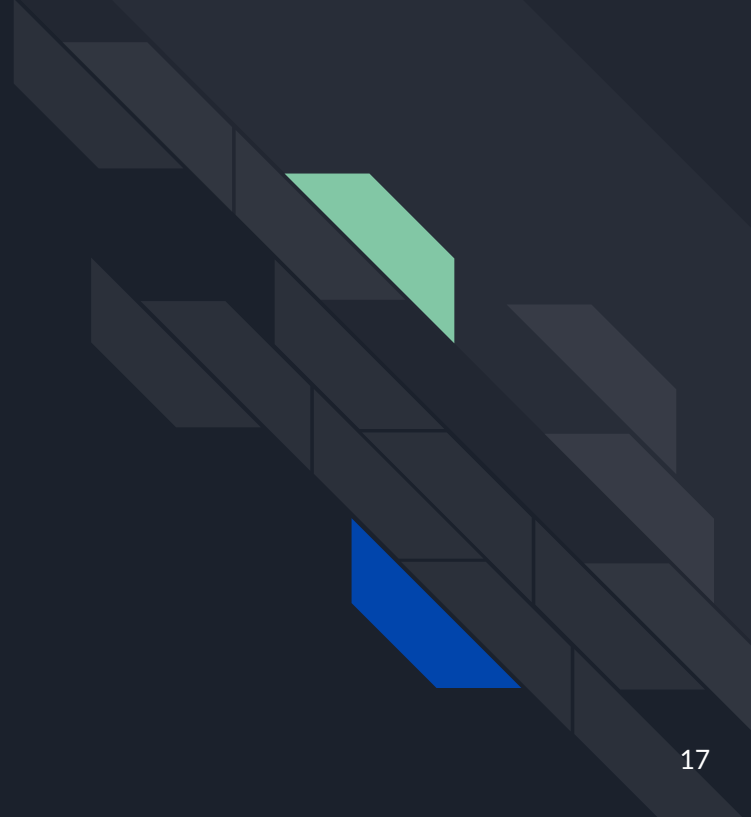
Create

	4.18.2	4.17.7	4.16.10	4.15.13	4.14.14
syscall_openat_count / syscall_openat_time	30002 / 585507996	30002 / 570539715	20002 / 809108188	20002 / 810024474	10002 / 13379790
syscall_close_count / syscall_close_time	20002 / 163096034	20002 / 150176480	20002 / 451371544	20002 / 462214346	10002 / 3956460
syscall_stat_count / syscall_stat_time	67 / 338	63 / 264	170075 / 46404701	170075 / 48753135	100025 / 10203590
syscall_fstat_count / syscall_fstat_time	20003 / 211337	20003 / 195341	20004 / 207866	20004 / 242882	10003 / 137310
syscall_realpath_count / syscall_realpath_time	5 / 35	5 / 30	50007 / 695390	50007 / 707193	40007 / 441254
syscall_chdir_count / syscall_chdir_time	34 / 13792	32 / 17900	40036 / 282802	40036 / 279825	20010 / 162245
total time taken	~28m 20s	~26m 30s	~27m 30s	~27m 20s	~1m 30s

List

	4.18.2	4.17.7	4.16.10	4.15.13	4.14.14
<code>syscall_openat_count / syscall_openat_time</code>	10009 / 8399074	10009 / 8302566	10009 / 7956770	10009 / 8251093	10008 / 6579450
<code>syscall_close_count / syscall_close_time</code>	10008 / 4580979	10008 / 4430866	10008 / 3896773	10008 / 4197262	10007 / 3414322
<code>syscall_stat_count / syscall_stat_time</code>	14 / 238	14 / 89	52 / 2444	52 / 2460	58 / 3156
<code>syscall_fstat_count / syscall_fstat_time</code>	10012 / 55752	10012 / 48500	10015 / 48166	10015 / 52721	10015 / 45150
<code>syscall_realpath_count / syscall_realpath_time</code>	8 / 342	8 / 220	14 / 78	14 / 359	13 / 68
<code>syscall_readdir_count / syscall_readdir_time</code>	10003 / 63917	10003 / 58672	10003 / 51169	10003 / 51819	10003 / 45446
<code>total time taken</code>	~20s	~20s	~20s	~20s	~20s

Build and test



Testing integration

- Fast moving upstream
 - Keeping up with any breaking changes
- Automated testing
- Jenkins based pipeline on CentOS CI infrastructure
 - sponsored for vibrant and active community projects
 - backed by OCP(Openshift Cloud Platform)
 - separate namespace
 - none or minimal downtime
- Act on failure notification
 - analyze and debug
 - initiate discussion or propose fixes

Nightly runs

Samba RPM [build jobs](#)

- Configured to run on current [supported release branches + master](#)
- Packages built are uploaded to [artifacts server](#)
- Easily available for public consumption

GlusterFS [integration job](#)

- Very basic chosen subset of [smb2](#) torture tests
- Run against replicated and disperse volumes

Testing integration: updates & future

New home

- from [gluster/samba-integration](#) to [sink/sit-environment](#) + [sink/sit-test-cases](#)
- supporting openshift specs and build repositories also moved
 - [sink/samba-centosci](#)
 - [sink/samba-build](#)

Add CephFS

- Include CephFS integration with `vfs_cephfs`
 - Few fixes for `vfs_cephfs` already landed upstream [#2939](#)
- Preparatory work already started
 - Modularize GlusterFS related tasks [#6](#)

The gist and takeaways



Yes, it works !

- Handle based new VFS approach is now robust and secure
- Distributed file systems like GlusterFS are now nearly compatible with new VFS structure
- *at () call variants were introduced and proven to work flawlessly
- Various bugs discovered are now fixed with further enhancements
- Performance penalty is a concern and are to be looked at in future
- Established automated process to find breakages with distributed file system integration

Thank you !

Questions ?

Anoop C S

✉ anoopcs@samba.org

[matrix] [anoopcs:matrix.org](https://matrix.org/join/anoopcs:matrix.org)

🗨 [anoopcs:Libera.Chat](https://libera.chat/#anoopcs)