Solving Linux File System Pain Points

Steve French
Samba Team & Linux Kernel CIFS VFS maintainer
Principal Software Engineer Azure Storage

Legal Statement

- This work represents the views of the author(s) and does not necessarily reflect the views of Microsoft Corporation
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.

Who am I?

- Steve French smfrench@gmail.com
- Author and maintainer of Linux cifs vfs (for accessing Samba, Windows and various SMB3/CIFS based NAS appliances)
- Also wrote initial SMB2 kernel client prototype
- Member of the Samba team, coauthor of SNIA CIFS Technical Reference,former SNIA CIFS Working Group chair

  (and formerly IBM Linux File System Architect)
- Principal Software Engineer, Azure Storage: Microsoft

Outline

- General Linux File System Status – Linux FS and VFS Activity
- What are the goals?
- Some Key Pain Points
- Testing

A year ago … and now … kernel (including kernel file systems) improving

- 14 months ago we had Linux version 4.11 ie “Fearless Coyote”

A few days ago ago we got 4.20-rc2 “People’s Front”

Discussions driving some of the FS development activity ?

- New mount API, new fsinfo API
- Additional security features
- Many of the high priority, evolving storage features are critical:
  - Better support for faster storage
    - RDMA and low latency ways to access VERY high speed storage (NVMe etc.)
    - Faster (and cheaper) network adapters (10Gb→40Gb->100Gb ethernet … and RDMA)
    - I/O priority
  - Now that statx (extended stat) is in, adding more metadata flags
  - Locks, locks and more locks
  - Recent changes for dedupe …
  - Larger block sizes (block size > page size e.g. in XFS)
  - Broadening use of copy offload (e.g. “copy_file_range” syscall, cross fs copy)
    - In rsync, cp etc.
  - Shift to Cloud (longer latencies, object & file coexisting, stronger sec required)

2018 Linux FS/MM summit (in April)

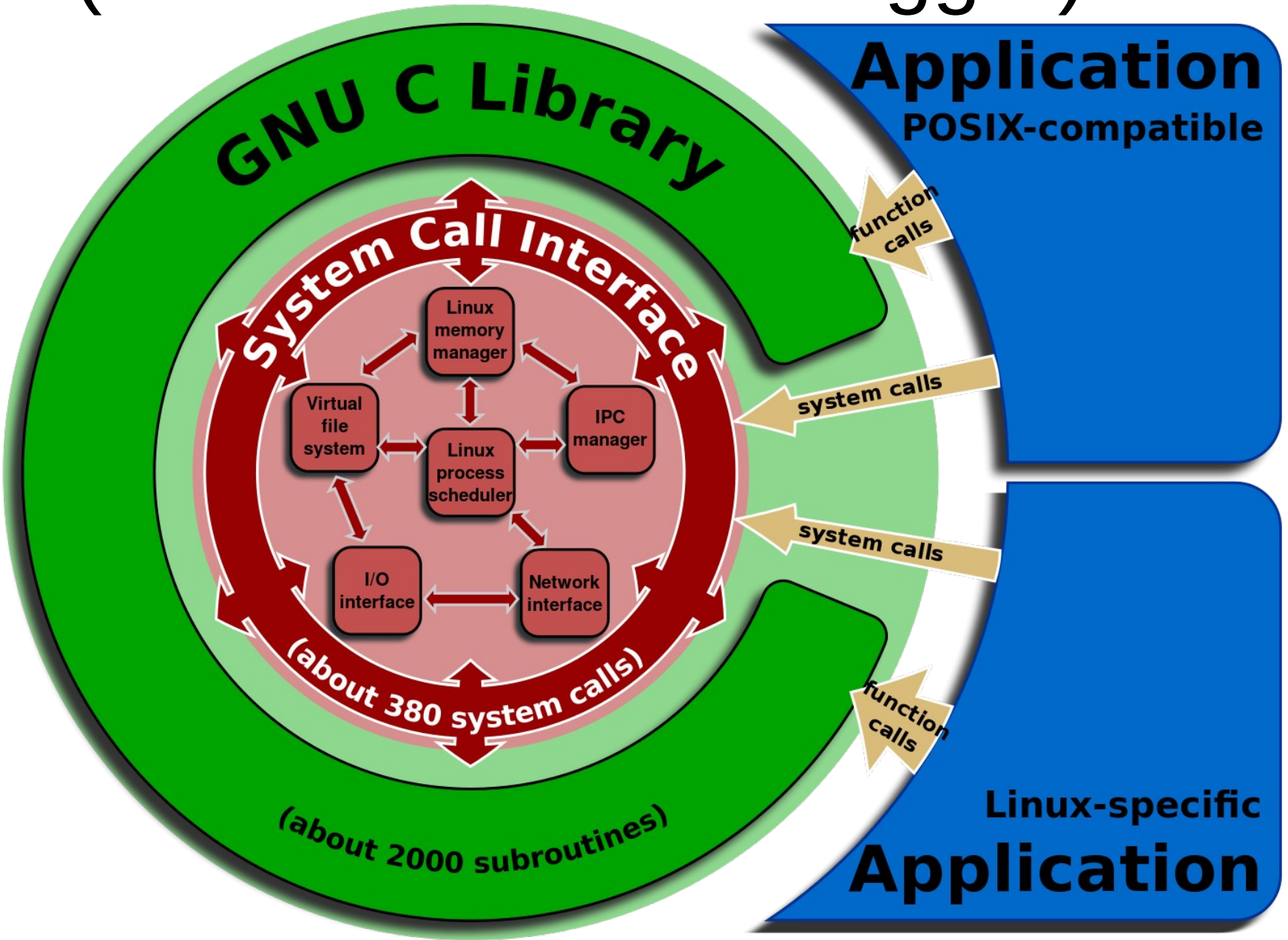
- Great group of talented developers

Most Active Linux Filesystems this year (January ie 4.15 to now ie 4.20-rc2)

- 4644 kernel file system changes since 4.15 kernel released, 6.8% of kernel overall (up). FS are important to Linux!
- Kernel is now 17.6 million lines of source code (measured today with sloccount tool)
- 60+ Linux file systems. cifs.ko (cifs/smb3 client) among more active (#4 out of 60 and growing). More activity is good!
- BTRFS 927 changesets (up), most changesets of any fs related compoenent
- VFS (overall fs mapping layer and common functions) 633
- XFS 564 (up)
- F2FS 378
- **cifs.ko** (CIFS/SMB2/SMB3 client) 344 (**up more than 100%**! And continuing to increase)
- Has 49,600 lines of kernel code (not counting user space helpers and samba userspace tools)
- NFS client 245 (down)
- NFS server 92 (down). Linux NFS server is **MUCH** smaller than Samba server (or even CIFS or NFS clients).
- And various other file systems: EXT4 200, Ceph 138, GFS2 123, AFS 118 …
- NB: Samba is as active as all Linux file systems put together (>4000 changesets per year) - broader in scope (by a lot) and also is user space not kernel. 3.4Million Lines of Code. **100x larger than the NFS server in Linux!**

- POSIX != Linux

(Linux API is much bigger)



Linux is BIG

- Currently 293 Linux syscalls!
- vs
- About 100 POSIX API calls

What are the goals?

- Make Linux File Systems fastest, most stable way to access general purpose data
- Make it easy, predictable for app writers to achieve great performance, security, integrity for their data

What about presentations at this conference that relate to fs … ?

- When eBPF meets FUSE: Improving Performance of User File Systems
- Zero copy UserMode File System
- Filename encoding and case-insensitive filesystems
- Untrusted File Systems
- Shiftfs
- FS checkpointing
- Optimizing network i/o in VMs for cluster/network fs
  - Virtio
- And more ...

Samba's favorite complaints

- Poll of a few Samba team guys (e.g Jeremy Allison of Google, Samba server, confirmed today ...)
- Data can't be migrated to Linux (not just Samba) without a richer ACL/security model
  - Whether "RichACLs" or ZFS ACLs or NFSv4.1 ACLs or SMB/NTFS ACLs or … (they are all 'close enough')
    - Can't get around need for Deny ACEs in migration and key workloads (Linux is only major OS that lacks this)
- O_NOFOLLOW semantics almost useless … need a new "O_NO_FULL_PATH_FOLLOW" for the whole path not just the last component (security hole otherwise)

The Copy Problem

- Copy offload (server side copy, cluster copy)
- Copy parallelization
  - And pegging the CPU on one processor …
- Cross mount copy (when the same fs)
- Option for limiting extending writes
- Better sparse file support in tools (not just in cp)
- Larger i/o sizes (NFS defaults to 1MB and SMB3 defaults to even larger – 4MB now, and can go to 8MB)

The RichACL problem

- ‘POSIX ACLs’ (used by some Linux fs) don’t have deny ACEs, mode bits are worse
- DENY Access Control Entries matter for real workloads, real government, regulatory, customer requirements
- ‘Claims Based’ ACLs can often be emulated but Apache (and also Windows flie systems) support this, which can also be helpful for modern privacy/security requirements

The statx and extra metadata problem ..

- Not just for the cloud ...

Theme: Let the File system do more if it knows more ...

- Sometimes the file system can do operations more efficiently than the VFS since it has access to more information (not just “copy from mount a to b”)
- Too much layering (virtual block layers) can also hurt file system’s ability to do its job by abstracting information that the FS much know

Exciting year!!

- LOTS of new features

  ...

Testing … testing … testing

- See README in xfstest git tree
- xfstesting page for individual file systems e.g.in cifs wiki https://wiki.samba.org/index.php/Xfstesting-cifs
- Easy to setup, exclude file for slow tests or failing ones
- XFSTEST status update
  - Bugzillas
  - Features in progress
  - Automating improvements (see e.g. how ext4 automates xfstest)
- Add more testcases to the FS layer (xfstests)!

Thank you for your time

- Future is very bright!