

GFS

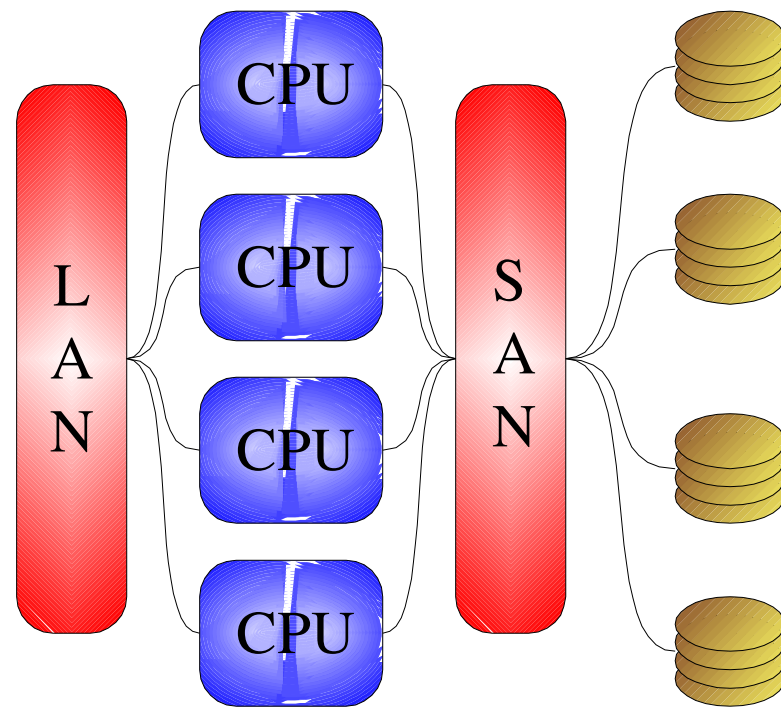
Ken Preslan
Red Hat, Inc.

Cluster Summit
July 29th, 2004

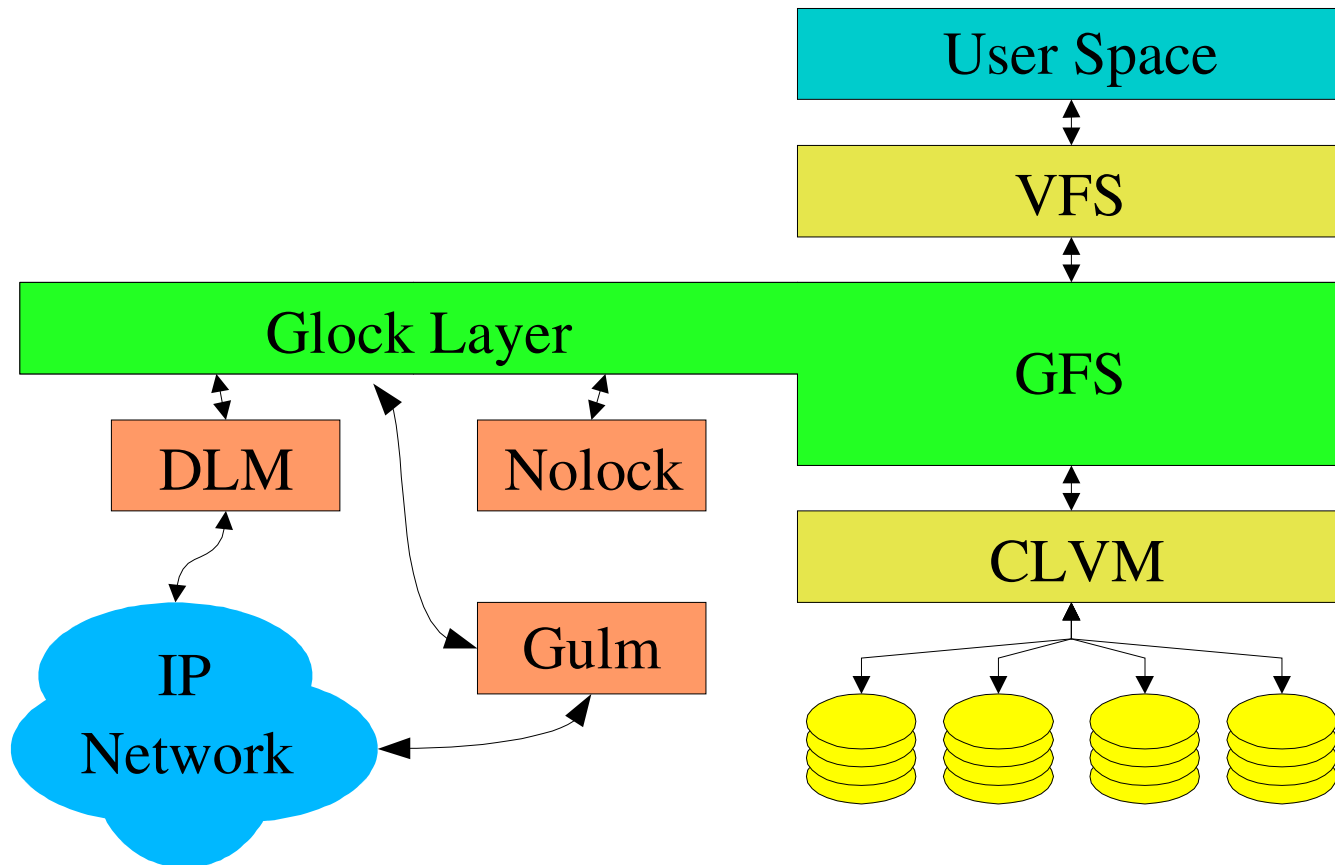
GFS Overview

- Symmetric Shared Disk Filesystem
- Similar to a local filesystem with intermachine locking
- Written from the ground up to be a CFS
- Aiming for posix semantics
- Large files and filesystems
- Journalled

GFS Cluster Layout



GFS Stack



Lock Module Interface

- Lock_harness – lightweight, GFS-specific CI switch
- Very lock-centric
- Minimal cluster management
- Mount, unmount, lock/LVB operations, plock operations, a callback (completion, blocking, recover-journal), recover-journal-done
- Maps Journal ID to nodes
- Handles all GFS inter-node communication

Older Features

- Asynchronous journaling
- Multiple journals – one per node
- ExHash directories
- Online Growable (data space and journals)
- Lock caching
- Full read and write-back caching
- Dynamic Inodes
- 64-bits everywhere
- Deadlock avoidance through lock sorting

New Features since 4.2

- Asynchronous locking
- Quotas
- Extended Attributes
- ACLs
- Shared locks are shared between processes
- Multi-writer Direct I/O
- Improved unlink/deallocation
- Improved allocation algorithms
- Improved plock code

New Features since 4.2

- Journalled data
- FS quiese support
- Better response to memory pressure
- Better transaction/log code
- Ability to convert metadata blocks back to data blocks
- Better NFS support
- Coherent shared mmap() support

New Features since 4.2

- Lots of bug fixes
- Lots of cleanup

Asynchronous Locking

- Lock modules and glock layer rewritten to support async locking
- Glock layer calls into the LM with request
- LM issues a callback with result
- Infrastructure still there to work with synchronous lock modules

Async Locking (Glock)

- Two options:
 - Prefetch
 - The calling code passes in a structure that defines the request. That structure can be polled or slept on
- Main users:
 - Prefetch inode locks on readdir
 - Statfs
 - Optimization acquiring multiple locks

Quotas

- Quotas are fuzzy
- Overruns are tunable
- Trade-off: More accuracy means more contention
- User and Group quotas
- Usage limit and Warn limit

Quotas

- Current quota values are cached in lock LVBs (to minimize quota file reads)
- Quota changes are cached in the filesystem in per-machine areas
- Changes are synced back to the quota file periodically
- Changes are also synced more often when the user gets closer to their limit
- Idea is to decouple quota handling as much as possible from the quota file

Future Work

- Shared Root
- Graceful error handling (no panics)
- Local storage utilization
- File Versioning
- Filesystem Snapshotting
- File locality controls
- Block based file backup support
- Metadata device locality
- File cache control

Future Work

- File System shrink
- Dynamic hotspot elimination
- DMAPI
- Failure recovery performance improvements
- Forced Unmount
- I/O load balancing
- Range-level locking
- Dynamic Journal reconfiguration

Future Work

- FSCK speed improvements
- Buffer Passing
- Operation Passing