



GFS Best Practices and Performance Tuning
Curtis Zinzilieta, RHCE
Regional Services Manager
Red Hat

GFS Best Practices and Performance Tuning

Agenda

- GFS Overview
- Best Practices
- Performance Tuning

GFS Best Practices and Performance Tuning

A highly available, clustered filesystem

- Standalone through hundreds of nodes
- When built with correct infrastructure, can sustain any single point of failure
- POSIX compliant, supporting ACL's, locking, quotas and extended attributes
- Oracle certified cluster filesystem
- Supports root filesystem and boot from SAN
- With shared storage, supports live virtual machine migration, and cluster of clusters virtualization support.
 - *See Thomas Cameron & Lon Hohberger's Cluster Failover and Demonstration session for details*

GFS Best Practices and Performance Tuning

Ongoing enhancement and feature development

- Improvements to statfs performance
- Conga cluster manager and configuration utility
- GFS1 stable and available, GFS2 in tech preview
 - Performance improvements
 - Improvements in small file/large directory performance
 - Journals as special files, add journals after storage creation
 - Uses less kernel memory
- GFS2 GA scheduled for RHEL5.3
 - gfs2_convert, unmount->run->remount

GFS Best Practices and Performance Tuning

Configuration and Setup

- Shared Storage
 - Includes SAN, iSCSI, GNBD
- Fencing
 - Power, SAN Fabric, Management Console, IPMI, Virtual System, SCSI Reservation
 - Scripts to shutdown access, found in `/sbin/fence_<methodname>`
- System prep
 - Setup ntpd
 - Bonded network and MPIO SAN connections

GFS Best Practices and Performance Tuning

Configuration and Setup

- system-config-cluster, X based setup
- Conga cluster manager and configuration utility
 - web interface with client and server daemons
 - Integrates setup of CLVM, GFS and Red Hat Cluster Suite.
- edit /etc/cluster/cluster.conf
- Easy setup of a single-node system
 - Create storage using CLVM (pv/vg/lvcreate)
 - `gfs_mkfs -j 1 -p lock_nolock /dev/vg00/mygfs`
- Easily change to a clustered filesystem configuration

GFS Best Practices and Performance Tuning

Configuration and Setup

- Quorum
 - One vote per node, majority division is in control of storage
 - Quorum disk partition
 - Raw, 10mb partition, best on its own LUN
 - Allows for single-node, minority survival
 - Additional voting and quorum selection

```
<cman expected_votes="1" two_node="0" />  
<quorumd interval="1" tko="10" votes="3" device="/dev/foo">  
  <heuristic program="ping A -c1 -t1" score="1"  
    interval="2" tko="3" />  
</quorumd>
```

GFS Best Practices and Performance Tuning

Configuration and Setup

- On-disk setup and configuration
 - Block Size
 - Volume Size and recovery considerations
 - Journals
 - Resource Group size (256M)
 - `gfs_mkfs -p lock_nolock -r 2048 /dev/vg/mygfs`
 - `gfs_mkfs -b 4096 -J <SIZE> -j 1 -p lock_nolock /dev/vg/mygfs`

GFS Best Practices and Performance Tuning

Configuration and Setup

- Locking, and single to clustered filesystem changes
- Distributed lock management
- Switching from lock_nolock to a clustered filesystem:

- Display the superblock:

```
•gfs_tool sb /dev/vg00/mygfs all
```

- Change Locking:

```
•gfs_tool sb /dev/vg00/mygfs sb_lockproto lock_dlm
```

- Change Cluster:

```
•gfs_tool sb /dev/vg00/mygfs sb_locktable mycluster:mygfs
```

GFS Performance Tuning

GFS Best Practices and Performance Tuning

Performance Tuning

- Fast Failover
 - Quick failure detection and recovery
- Throughput and Filesystem Performance
 - On-disk format
 - Parameters and tuning: `gfs_tool`
 - Kernel, readahead and other system tuning

GFS Best Practices and Performance Tuning

Performance Tuning for Faster Failover

- Configuration file changes to `/etc/cluster/cluster.conf`
- Heartbeat pause and timeout seconds
 - (RHEL4)
 - `<cman hello_timer="5" deadnode_timeout="21">` *(seconds)*
 - (RHEL5)
 - `<totem token="10000"/>` *(millisec)*
 - `<totem retransmits_before_loss="4"/>` *(count)*
 - **See 'man openais.conf' for more detail**
- `post-fail-delay`
 - `<fence_daemon post_fail_delay="0" post_join_delay="12"/>`
- Also change `post-fail-delay` when capturing cores, `sysrq-t`, etc
 - `<fence_daemon post_fail_delay="30" post_join_delay="12"/>`

GFS Best Practices and Performance Tuning

Performance Tuning for Faster Failover

- Filesystem settings
 - Dead node journal recovery
 - `gfs_tool settune recoverd_secs 60`
- Fast Fencing
 - Network configuration: private interconnect between nodes and fencing devices as appropriate
 - Fencing device selection
 - Fencing recovery action and requirements

GFS Best Practices and Performance Tuning

Performance Tuning for Throughput

- Filesystem layout
 - Journal
 - By default, journal only metadata. Journaling all data when working with small files can improve performance due to faster fsync return. Set flag for a file or directory:
 - `gfs_tool setflag inherit_jdata /gfs1/directory`
 - Mount options: 'noatime', 'noquota'
 - Breakup systems into multiple subdirectories
 - Resource Group sizing
 - `gfs_mkfs -p lock_nolock -r 2048 /dev/vg/mygfs`
 - RAID and filesystem growth considerations

GFS Best Practices and Performance Tuning

Performance Tuning for Throughput

- `gfs_tool`
 - `gfs_tool gettune /mountpoint`
- Locking Management: `ilimit`, etc, no longer needed
- `scand_seconds`, `glock_purge`, `demote_secs`,
 - `gfs_tool settune /mntpoint glock_purge 50 (default 0 percent)`
 - `gfs_tool settune /mntpoint demote_secs 100 (default 300)`
 - `gfs_tool settune /mntpoint scand_secs 3 (default 5)`

GFS Best Practices and Performance Tuning

Performance Tuning for Throughput

- gfs_tool
 - max_readahead
 - `gfs_tool settune /mntpoint max_readahead 262144` (bytes)
 - **Enable fast statfs if possible**
 - `gfs_tool settune /mntpoint statfs_fast 1` (enabled)

GFS Best Practices and Performance Tuning

Performance Tuning for Throughput

- System configuration options
 - I/O Scheduler Selection: CFQ -vs- Deadline
 - Readahead and other system level tuning
 - `/sys/block/<dev>/queue/nr_requests`
 - `/sys/block/<dev>/queue/read_ahead_kb`
 - `/sys/block/<dev>/queue/scheduler`
 - Use Direct I/O where appropriate
 - HBA device tuning

GFS Best Practices and Performance Tuning

Performance Tuning for Throughput

- Testing tools
 - Standard Linux/Unix tools
 - vmstat, iostat, sar, nfsstat
 - Benchmarks and load tools
 - IOZone
 - Bonnie++
 - Postal
 - others...

GFS Best Practices and Performance Tuning

Where to go next?

- Red Hat Training: RH436, Enterprise Storage Management
 - <https://www.redhat.com/training/architect/courses/rh436.html>
- Web Sites
 - <http://sources.redhat.com/cluster/wiki/>
- Red Hat documentation
 - <https://www.redhat.com/docs/manuals/csgfs/>
- This presentation:
 - <https://people.redhat.com/czinzili/>