Developments in GFS2

Andy Price <anprice@redhat.com>
Software Engineer, GFS2

OSSEU 2018

GFS2 recap

- Shared storage cluster filesystem
- High availability clusters
- Uses glocks (“gee-locks”) based on DLM locking
- One journal per node
- Divided into resource groups
- Requires clustered lvm: clvmd/lvmlockd
- Userspace tools: gfs2-utils

Not-so-recent developments

- Full SELinux support (linux 4.5)
  - Previously no way to tell other nodes that cached labels are invalid
  - Workaround was to mount with -o context=<label>
  - Relabeling now causes security label invalidation across cluster

Not-so-recent developments

- Resource group stripe alignment (gfs2-utils 3.1.11)
  - mkfs.gfs2 tries to align resource groups to RAID stripes where possible
  - Uses libblkid topology information to find stripe unit and stripe width values
  - Tries to spread resource groups evenly across disks

Not-so-recent developments

- Resource group LVBs (linux 3.6)
  - -o rgrplvb (all nodes)
  - LVBs: data attached to DLM lock requests
  - Allows resource group metadata to be cached and transferred with glocks
  - Decreases resource group-related I/O

Not-so-recent developments

- Location-based readdir cookies (linux 4.5)
  - -o loccookie (all nodes)
  - Uniqueness required by NFS
  - Previously 31 bits of filename hash
  - Problem: collisions with large directories (100K+ files)
  - Cookies now based on the location of the directory entry within the directory

Not-so-recent developments

- gfs_controld is gone (linux 3.3)
  - dlm now triggers gfs2 journal recovery via callbacks
  - gfs2-utils now required on all nodes for withdraw scripts triggered by a udev event handler.

Recent developments

- iomap writes
  - Multi-page writes
  - Overheads amortized across the range of pages
  - SEEK_HOLE and SEEK_DATA also added using iomap
  - Initial results show good performance improvements in workloads such as SAS calibration

Recent developments

- New resource group header fields
  - Checksum (crc32)
    - Provides instant, reliable error detection
  - Location and size data from rindex
  - Distance to the next resource group
    - Reduces reliance on the rindex and allows more efficient iteration over resource groups.

Recent developments

- Expanded journal log header information
  - statfs changes
    - Allows global statfs updates to be done on journal flush instead of from separate statfs change file
  - Debugging information
    - Pointers back to journal inode, statfs inode, quota inode
  - Timestamp (nanosecond granularity)
  - New crc32 to check the new fields

gfs2 on IBM z Systems

- s390x (64-bit) virtual environment
  - Usual virt challenges
    - Low entropy pool
    - Resource overcommitment
- Cluster fencing provided by SMAPI
  - fence_zvmip
- Multi-LPAR config (Single System Image) possible

Future developments

- Faster fsck.gfs2
  - Currently reads block-by-block
  - Use aio, larger reads
  - Makes much larger filesystems practical
- Process-shared resource group locking
- trusted.* xattrs
- Deprecate ‘meta’ gfs2 filesystem fork
  - Replace with better interfaces (sysfs/ioctl)

Future developments

- Filesystem versioning
  - Currently no way to prevent mounting if features aren’t supported or mount options conflict
  - Important for new option defaults where all nodes must use the same ones, e.g. rgrplvb
  - Required to prevent new on-disk data being written by one node and treated as corruption by another
  - fsck.gfs2 must have the same version semantics of gfs2

Future developments

- Journal flush-related performance improvements
  - Journal flushing can be slow, blocks new transactions, increases glock release latency
  - Data must be written back before journal is flushed
  - Tricky problem but potential solutions are being investigated.
  - Plenty of scope for modifying journal log header structures if necessary

Thanks!

Questions?

Mailing list: cluster-devel@redhat.com

gfs2: https://git.kernel.org/pub/scm/linux/kernel/git/gfs2/linux-gfs2.git

gfs2-utils: https://pagure.io/gfs2-utils