

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

**LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.**

www.theredhatsummit.com

GFS2: DEBUGGING PERFORMANCE ISSUES

Carlos Eduardo Maiolino

Software Maintenance Engineer, Red Hat

05-05-2011

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Agenda

- On-Disk format
- Glocks
- Getting useful data
- Analyzing data
- Performance considerations
- How to help performance
- Where can I find more information?
- Red Hat Subscriptions
- Contact

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



On-disk format

- A Gap of 64KB
- Superblock
- Resource Groups (up to 2GB)
 - Rgrp headers
 - Bitmaps blocks
 - Blocks available for use



On-disk format #2

- Superblock
 - Pointer to the 'root' directory location and the root of the metafs
 - Block size information
 - Lock protocol
 - Lock table



On-disk format #3

- Resource Groups
 - Splits filesystem into slices (similar to block groups)
 - Contains a resource group header
 - rg_free
 - rg_dinodes
 - Bitmaps (2 bits per block)
 - Allocated/free
 - Data/metadata
 - The data/metadata blocks

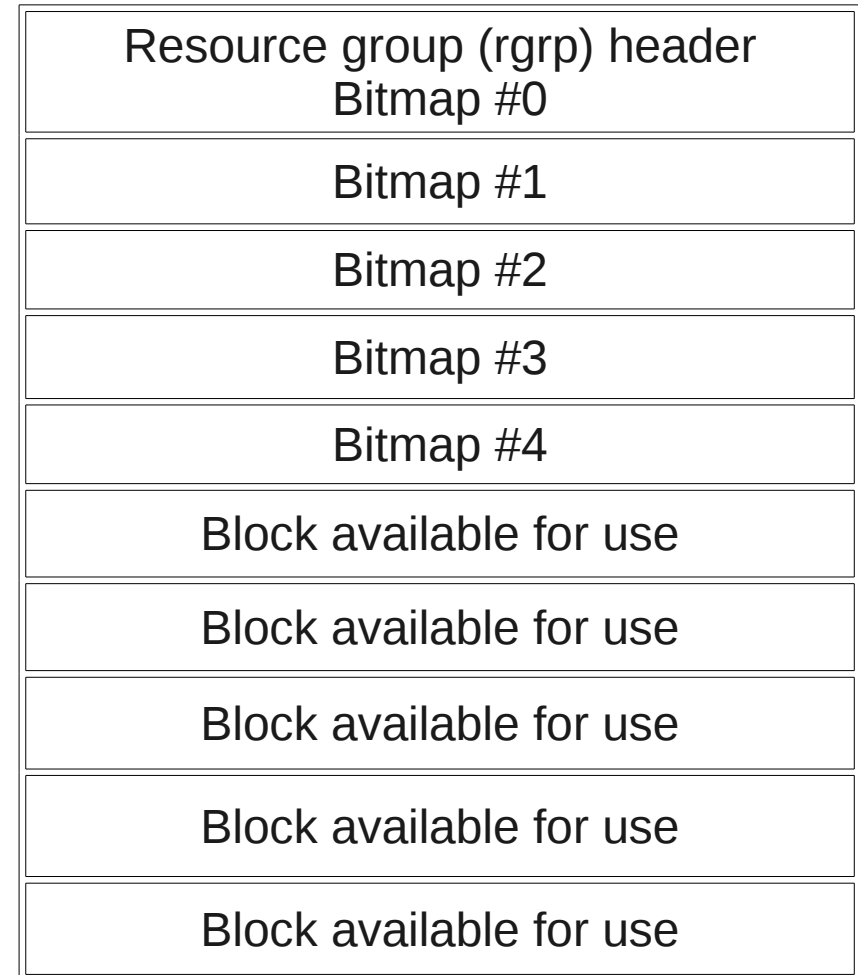


On-disk format #4

General on-disk view



Internal rgrp organization



Glocks

- GFS2 concept used to manage cache
 - 1:1 mapping from glock to dlm lock
 - Manage data and metadata caches of inodes
 - The lock state determines what can be cached
- Most performance problems occur when the implication of the glock system are not understood
- Easily viewed by a debugfs interface
- Holders: requests to grant a lock (granted or not)



Glocks #2

- Three lock modes:

GLOCK	DLM
Shared	Protected Read
Exclusive	Exclusive
Deferred	Concurrent Write

- Two or more processes can grab a **shared lock** at the same time
- **Exclusive locks** are not compatible with any other lock type
- NULL lock mode is used to maintain a ref count on LVBs (currently used only for quota data)



Glocks #3

- Dumping glocks:
 - Mount debugfs interface:
`#mount -t debugfs debugfs /sys/kernel/debug`
 - Read glocks file (gfs2 module and filesystem should be already mounted) at:
 - `/sys/kernel/debug/gfs2/<locktable>/glocks`
example: `#cat /sys/kernel/debug/gfs2/rhel\gfs2/glocks`



Glocks #4

- Glock dump output example:

```
G: s:SH n:2/1c9e9d f:lq t:SH d:EX/0 a:0 r:3  
  l: n:2149/1875613 t:4 f:0x00 d:0x00000001 s:3864  
  H: s:EX f:W e:0 p:3124 [flush-253:2] gfs2_glock_nq_init+0x16/0x37 [gfs2]  
G: s:EX n:2/18aaba f:yflq t:EX d:EX/0 a:0 r:5  
  H: s:EX f:H e:0 p:3798 [dd] gfs2_write_begin+0x64/0x3b4 [gfs2]  
  H: s:EX f:W e:0 p:1540 [flush-253:2] gfs2_glock_nq_init+0x16/0x37 [gfs2]  
G: s:SH n:5/1dca0b f:lq t:SH d:EX/0 a:0 r:3  
  H: s:SH f:EH e:0 p:3122 [(ended)] gfs2_glock_nq_init+0x16/0x37 [gfs2]  
G: s:SH n:5/1eaaba f:lq t:SH d:EX/0 a:0 r:3  
G: s:SH n:5/1ca475 f:lq t:SH d:EX/0 a:0 r:3  
G: s:SH n:5/ee5f0 f:lq t:SH d:EX/0 a:0 r:3  
G: s:EX n:3/119252 f:yflq t:EX d:EX/0 a:2 r:4  
  H: s:EX f:tH e:0 p:3400 [dd] gfs2_glock_nq_init+0x16/0x37 [gfs2]  
  R: n:1151570 f:30000000 b:36263/36263 i:9
```



Glocks #5

- Glock types

Type	Lock Type	Usage
2	Inode	Inode data and metadata
3	Resource Group	Resource group metadata
5	lopen	Inode last closer detection
6	flock	flock() syscall
8	Journal	Journal Mutexes

G: s:EX n:2/18aaba f:ylq t:EX d:EX/0 a:0 r:5

H: s:EX f:H e:0 p:3798 [dd] gfs2_write_begin+0x64/0x3b4 [gfs2]

H: s:EX f:W e:0 p:1540 [flush-253:2] gfs2_glock_nq_init+0x16/0x37 [gfs2]

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Glocks #6

- Glock Flags

Flag	Name	Meaning
d	Pending demote	A remote demote request
D	Demote	A demote request (local or remote)
i	Invalidate in progress	Invalidating pages under that glock
l	Locked	The glock is in the process of changing state
y	Dirty	Data needs to be flushed before release glock

G: s:EX n:2/18aaba **f:ylq** t:EX d:EX/0 a:0 r:5

H: s:EX f:H e:0 p:3798 [dd] gfs2_write_begin+0x64/0x3b4 [gfs2]

H: s:EX f:W e:0 p:1540 [flush-253:2] gfs2_glock_nq_init+0x16/0x37 [gfs2]

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Glocks #7

- Holder flags

Flag	Name	Meaning
a	Async	Do not wait for glock result (will poll for result later)
A	Any	Accepts any compatible lock
c	No cache	When unlocked, demote DLM lock immediately
H	Holder	Requested lock is granted
W	Wait	Waiting for a lock to be granted

G: s:EX n:2/18aaba f:ylq t:EX d:EX/0 a:0 r:5

H: s:EX **f:H** e:0 p:3798 [dd] gfs2_write_begin+0x64/0x3b4 [gfs2]

H: s:EX **f:W** e:0 p:1540 [flush-253:2] gfs2_glock_nq_init+0x16/0x37 [gfs2]

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Getting useful data

- 'glocks' file in debugfs are realtime information
- Is the filesystem stuck or just slow?
 - Collecting two or more glock dumps of the filesystem will help to answer this question
- Each node has its own glock table, so **always** collect glock dumps from **all nodes** accessing the filesystem
- Triggering a sysrq-t (stack traces) is often useful
- The 'hangalyzer' application can help to collect data



Analyzing data

- Start looking for holders in waiting state
- What type of glock these holders are waiting?
 - Inode(2) and rgrp(3) glocks usually are the most important
- Did the holders list change between the glock dumps?
- Which processes are related to these holders?
 - The 'p:' field on the Holder line will tell you
- **The holder holding the glock can be on a different node**



Analyzing data #2

- What if there are changes on the holders list but the filesystem is slow?
 - The node holding that glock is a master dlm lock or a remote copy?
 - Lots of remote copies can result in many cache invalidations
- Find out the file related with the glock (inode glocks)
 - It can give a clue about why the filesystem is slow
 - **Be careful, running `find` command on a filesystem with performance problems can make the issue worse.**



Analyzing data #3

- What if glocks/holders didn't change between glock dumps?
- Found a deadlock?
 - An application deadlock?
 - Better look at the application code
 - A possible gfs2 deadlock?
 - Please open a support case immediately



Performance considerations

- A shared filesystem involves much more variables which we should take care than a local filesystem
 - Network
 - Red Hat cluster-suite
 - Applications I/O patterns
 - An application who knows to be issuing I/O to a shared filesystem will perform much better than an application who doesn't
- We can't expect the same performance as a local filesystem



How to help performance

- GFS2 is “self-tuning”
 - Most performance options are already enabled by default
- atime and diratime
 - Mount the filesystem with 'noatime' and 'nodiratime'
- Avoid stat() the files
 - Disable ls –color alias on your system
- Don't use slow_statfs unless absolutely necessary
 - slow_statfs is disabled by default



How to help performance #2

- Prefer many smaller filesystems than a larger one
- Be careful with the Resource Groups size
 - When made, gfs2 tries to pick an optimal size, but it might not be the best to your environment
- Pre-allocate files if possible



How to help performance #3

- Avoiding lock contention
 - Keep each node on its own piece of the filesystem
 - Avoid two or more nodes touching the same data at the same time whenever is possible
 - Have each node allocating its own files if possible
 - Locking a lock on the lock master is much faster.



How to help performance #4

- General applications
 - Prefer flocks instead of posix locks
 - Use DLM in user-space is welcomed
- NFS
 - Be careful, NFS is not cluster-aware
 - active/passive only
 - Use “localflocks” mount option



Red Hat Subscriptions

- How valuable is a Red Hat subscription when using GFS2 ?
 - Most gfs2 upstream contributions
 - Upstream maintainers works for Red Hat
 - Red Hat is always “on top” of gfs2 issues
 - GFS2 needs cluster infrastructure also developed by Red Hat



Where can I find more information?

- Red Hat documentation
 - <http://docs.redhat.com/>
- How do I debug a gfs/gfs2 performance problem
 - <https://access.redhat.com/kb/docs/DOC-41485>
- Cluster and GFS deployment best practices
 - <https://access.redhat.com/kb/docs/DOC-40821>
- How does file locking works
 - <https://access.redhat.com/kb/docs/DOC-41609>
- How can I gather gfs2 lockdump information (hangalyzer)
 - <https://access.redhat.com/kb/docs/DOC-34401>



Contact

- Red Hat customer portal
 - <http://access.redhat.com>
- Knowledge base
 - <https://access.redhat.com/kb/knowledgebase>
- Red Hat online user groups
 - <https://access.redhat.com/groups/groups-dashboard>

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Questions

Questions?

Dúvidas?

Fragen?

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



LIKE US ON FACEBOOK

www.facebook.com/redhatinc

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#redhat

READ THE BLOG

summitblog.redhat.com

GIVE US FEEDBACK

www.redhat.com/summit/survey

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

