Spectrum Scale

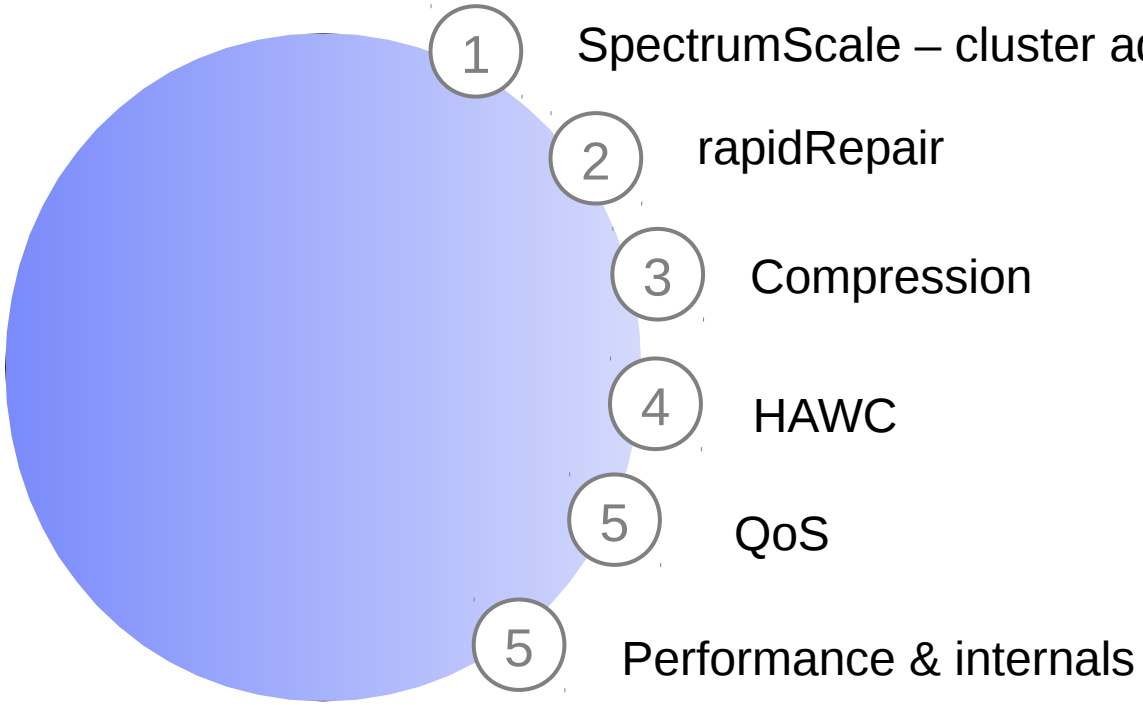
workshop Ehningen 2016

olaf.weiser@de.ibm.com
IBM Deutschland
SpectrumScale Support Specialist

Spectrum Scale v4.2 adds

- Improved provisioning and administrator productivity with a new GUI
- Management of resource contention with **Quality of Service (QoS)** prioritization and monitoring
- Storage portfolio visibility with IBM Spectrum Control integration
- Unified file and object storage with integrated data analytics
- Policy-based migration to the optimum storage pool and active archive
- Storage space savings with **compression**
- Asynchronous Disaster Recovery remote replication capability
- **Improved design for faster**, easier, and more reliable workflows
- **Hadoop support**, multiprotocol NAS, and proven enterprise capabilities
- **H**ighly **a**vailable read and **w**rite **c**aching to accelerate applications

Agenda



1. SpectrumScale – cluster administration w/o root
2. rapidRepair
3. Compression
4. HAWC
5. QoS
5. Performance & internals

Agenda - sudo/ssh wrapper

1) Background information and motivation
2) Commication with sudo wrapper enabled
3) SpectrumScale – configuration example

SpectruMScale – sshwrapper Background

- Currently GPFS administrators use the “root” login to the nodes and perform the operations (using ssh, say for ex: mmcrcluster, mmcrnsd, mmcrfs etc)

- Many customers raised concerns and not allow “root” login which they felt as a security risk

- Work of Sudo Wrappers is based on the “GPFS and Remote shell”. Will allow customers admin to perform the operations using a non-root user

White paper :

https://www.ibm.com/developerworks/community/wikis/form/anonymous/api/wiki/fa32927c-e904-49cc-a4cc-870bcc8e307c/page/f0cc9b82-a133-41b4-83fe-3f560e95b35a/attachment/7d4dd845-cee1-4a3d-ba8e-d67b4beffe6b/media/GPFS%20and%20remote%20shell.pdf

passwordless access – consider....

- Category 1: "Arbitrary remote program execution":
  - ssh
  - mmdsh …
  - mmaddcallback
- Category 2: "User specified remote program execution"
  - Existing policy feature "EXEC" for migrating files to/from external pools
- Category 3: "Remote GPFS command execution"
  - being able to run a command from a list of executables in /usr/lpp/mmfs/bin
- Category 4: "Remote GPFS code execution":
  - able to run some code compiled into mmfsd or mmsdrserv and has control over the parameters passed to that code

passwordless access – consider....

pro's & con's:

- Customers expect to have full “control” over root account activity
  - who is root , what is root doing
- Customer expect a software to work properly automatically, with no manual intervention
  - → *e.g. what if creating a new file system(GPFS) and the need for changing the /etc/fstab*
- Using well known open-source and maintained software ssh , or proprietary software / RPC calls (seen as back doors ?) for need remote executions

Agenda - sudo/ssh wrapper

1.) Background information and motivation

2.) Commication with sudo wrapper enabled

3.) SpectrumScale – configuration example

SpectrumScale: wrapper – communication flow

mmgetstate –N nodeB

root

mmremote mmgetstate

root

ssh gpfsadmin@NodeB sudo mmremote mmGetState

root

named user

sudo mmgetstate –N nodeB

gpfsadmin

NodeA

sudo mmremote mmGetState

gpfsadmin

NodeB

SpectrumScale - needed sudo configurations

1. Create user and group.

   In our example, username is “gpfsadmin” and group is “gpfs”

2. Allow root user to login as gpfsadmin via ssh without prompt for password

3. Add the following lines to /etc/sudoers

   - # Preserve GPFS env vars
     Defaults env_keep += "MMMODE environmentType GPFS_rshPath GPFS_rcpPath mmScriptTrace GPFSCMDPORTRANGE GPFS_CIM_MSG_FORMAT“

   - ## Allows members of the gpfs group to run all commands but only selected commands without a password
     %gpfs ALL=(ALL) PASSWD: ALL, NOPASSWD: /usr/lpp/mmfs/bin/mmremote, /usr/bin/scp, /bin/echo, /usr/lpp/mmfs/bin/mmsdrrestore

   - ## tty

     Defaults:%gpfs !requiretty

SpectrumScale: sudo setup - cont()

Disable root login (sshd)

- Edit /etc/ssh/sshd_config and make sure

  PermitRootLogin no

- Restart sshd daemon

  service sshd daemon

- Make sure you can't do ssh localhost /bin/date as root id

Sample file: /usr/lpp/mmfs/samples/sudoers.sample

Agenda - sudo/ssh wrapper

1.) Background information and motivation

2.) Commication with sudo wrapper enabled

3.) SpectrumScale – configuration example

SpectrumScale with sudo wrapper enabled

- sudo controlled access
- Specify a named user, who is capable to get root privileges locally on every node

Example:

```
laff@node1:~> mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         clusterA.site
  GPFS cluster id:           15425310929724810145
  GPFS UID domain:           clusterA.site
  Remote shell command:      sudo wrapper in use
  Remote file copy command:  sudo wrapper in use
  Repository type:           CCR

 Node  Daemon node name  IP address  Admin node name  Designation
-----------------------------------------------------------------
   1   node1.site        10.0.0.111  node1.site       manager-gateway
   2   node2.site        10.0.0.112  node2.site       quorum-gateway

laff@node1:~>
```

wrapper scripts are effictive

16.03.16

Creating a cluster and license (Demo)

- Cmd with sudo

```
sudo /usr/lpp/mmfs/bin/mmcrcluster --use-sudo-wrapper -N
c12c4apv10:quorum,c12c4apv11
mmcrcluster: Performing preliminary node verification ...
mmcrcluster: Processing quorum and other critical nodes ...
mmcrcluster: Processing the rest of the nodes ...
mmcrcluster: Finalizing the cluster data structures ...
mmcrcluster: Command successfully completed
mmcrcluster: Warning: Not all nodes have proper GPFS license
designations.
    Use the mmchlicense command to designate licenses as needed.
mmcrcluster: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

- Check

```
mmlslicense
   -bash: /usr/lpp/mmfs/bin/mmlslicense: Permission
denied

ls -lrt /usr/lpp/mmfs/bin/mmlslicense
   -r-x------ 1 root root 3041 Sep 28 14:50 /usr/lpp/mmfs/bin/mmlslicense
```

Changing a license (Demo)

Now, take sudo

```
sudo /usr/lpp/mmfs/bin/mmchlicense server
--accept -N c12c4apv10,c12c4apv11
```

```
The following nodes will be designated as possessing server
licenses:
        c12c4apv10.gpfs.net
        c12c4apv11.gpfs.net
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

Enable sudo wrapper for existing cluster

- reconfiguring existing SpectrumScale cluster is supported
- flip between use / no use of sudo wrapper is possible
- parallel administration scenarios is possible as well

Mini howto:

(1) bring your cluster to gpfs 4.2 level

```
node1:~ # mmlsconfig minReleaseLevel
minReleaseLevel 4.2.0.0
node1:~ #
```

(2) configure sudo

Enable sudo wrapper for existing cluster

- (3) Verify your sudo configuration

```
laff@node2:~> id
uid=1000(laff) gid=100(users) groups=100(users),1000(gpfs)
laff@node2:~> sudo /usr/lpp/mmfs/bin/mmcommon test sshwrap node2
mmcommon test sshwrap: Command successfully completed
laff@node2:~> sudo /usr/lpp/mmfs/bin/mmcommon test sshwrap node1
mmcommon test sshwrap: Command successfully completed
laff@node2:~>
```

- (4) Change existing cluster***

```
laff@node2:~> sudo mmchcluster --use-sudo-wrapper
mmsetrcmd: Command successfully completed
mmsetrcmd: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
laff@node2:~>
```

***Changing cluster works for offline and active clusters

Enable sudo wrapper for existing cluster

```
mmcommon test sshwrap: Command successfully completed
laff@node2:~> sudo mmlscluster
laff's password:

GPFS cluster information
========================
  GPFS cluster name:         clusterA.site
  GPFS cluster id:           15425310929724810145
  GPFS UID domain:
  Remote shell command:
  Remote file copy comman
  Repository type:

 Node  Daemon node name
-------------------------
   1   node1.site
   2   node2.site

laff@node2:~> sudo /usr/l
```

```
laff@node2:~> sudo /usr/lpp/mmfs/bin/mmchcluster --use-sudo-wrapper
mmsetrcmd: Command successfully completed
mmsetrcmd: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
laff@node2:~> sudo mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         clusterA.site
  GPFS cluster id:           15425310929724810145
  GPFS UID domain:           clusterA.site
  Remote shell command:      sudo wrapper in use
  Remote file copy command:  sudo wrapper in use
  Repository type:           CCR

 Node  Daemon node name  IP address  Admin node name  Designation
-----------------------------------------------------------------
   1   node1.site        10.0.0.111  node1.site       manager-gateway
   2   node2.site        10.0.0.112  node2.site       quorum-gateway

laff@node2:~>
```

Enable sudo wrapper for existing cluster

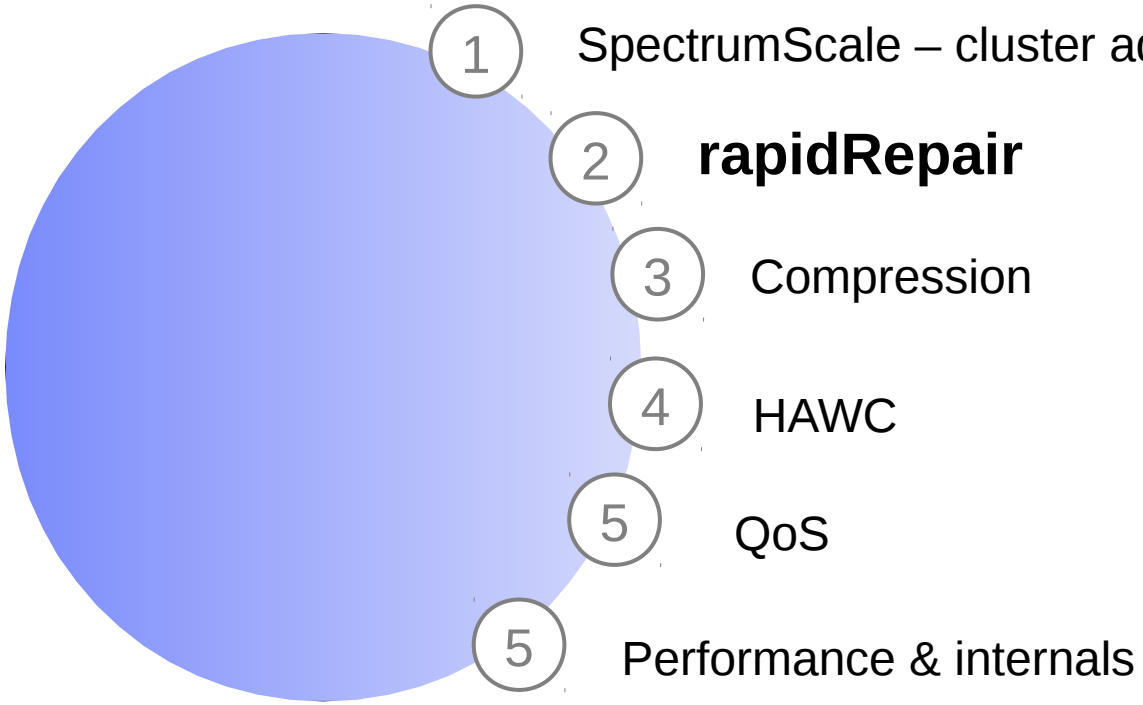
- Sometimes it is a good idea to be able to work as root directly
- No need to change cluster configuration
  from „sudo wrapper in use“ mode
- you can use root user in parallel*** ,

***so far it is allowed in your environment

Running mmcmd directly as root

1) export SUDO_USER=gpfsadmin # your admin user
2) allow root login and add root user to gpfs group or disable requiretty for root.

Agenda

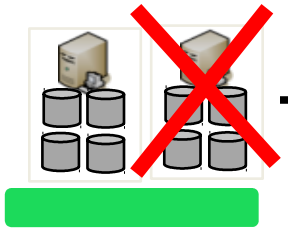


1. SpectrumScale – cluster administration w/o root
2. **rapidRepair**
3. Compression
4. HAWC
5. QoS
5. Performance & internals

Rapid Repair - background

- When the fs has down disks and replicas are 2 or 3, gpfs will still allocate blocks on the down disks. And marked files with blocks allocated in down disks as "missing update"
- When we start down disks, gpfs will scan the whole fs, check files with "missing update" flag, and try to repair the whole file.
- When the file is very large, and have many blocks allocated in "down disks", gpfs will repair all these blocks.
- So the issue is that in disk down period, if only have a small write and only a few blocks are changed, gpfs will still try to repair the whole file.
- "Rapid Repair" is aim to fast repair process

SpectrumScale – restripeOnDiskFailure



nodeLeave,diskFailure

```
mmchconfig metadataDiskWaitTimeForRecovery=seconds   (default 2400 sec.)
mmchconfig dataDiskWaitTimeForRecovery=seconds       (default 3600 sec.)
mmchconfig minDiskWaitTimeForRecovery=seconds        (default 1800 sec.)
mmchconfig maxDownDisksForRecovery=disks          (default 16 disks)
mmchconfig maxFailedNodesForRecovery=nodes      (default 3 nodes)
```

gpfsRecoverFailedDisk

tschdisk start -a / restripefs -r

nodeJoin

gpfsRestartDownDisks

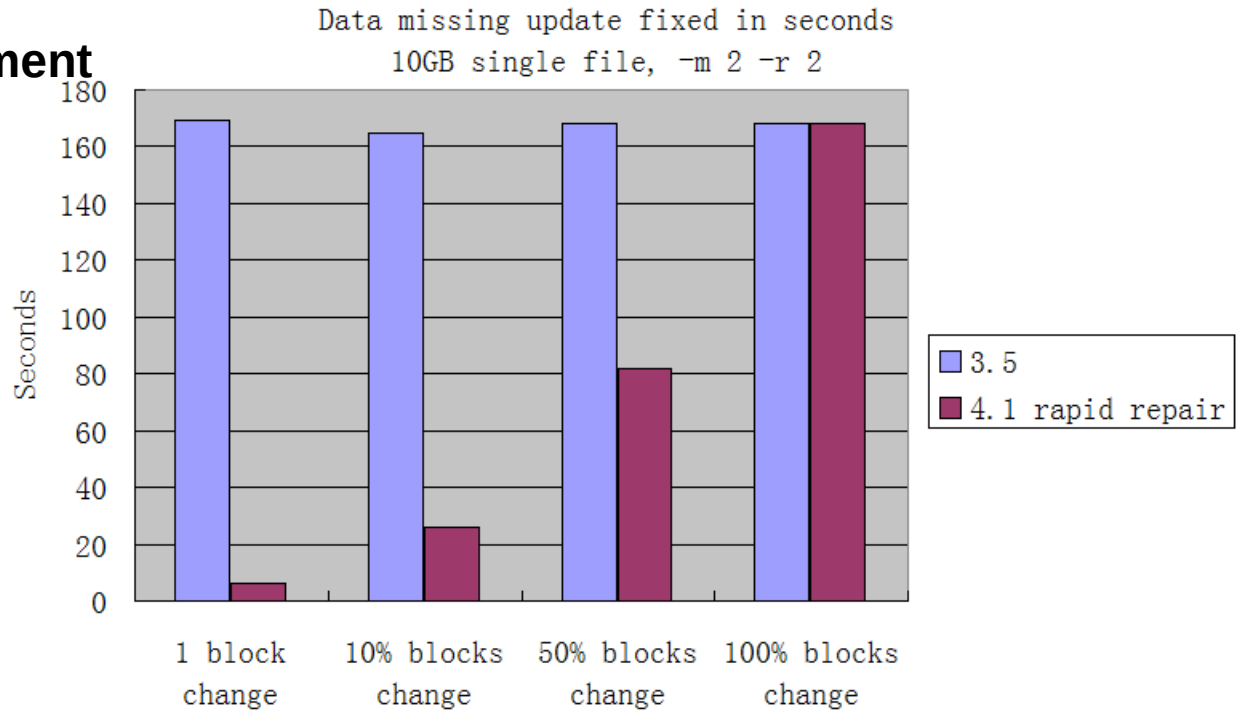
Logfiles:

/var/adm/ras/mmfs.log.latest
/var/adm/ras/restripefsOnDiskFailure.log.

Improvements in 4.1

repair

- **Changes in DiskAddress**
  - A new (former unused)bit in DiskAddress indiated if this data block is "missing update"
  - Only need to repair "missing update" blocks in repair process
- **Performance Improvement**



External Changes

- mmlsfs device --rapid-repair
- mmchfs device --(no)rapid-repair

```
# mmlsfs foofs
 --rapid-repair     yes                 rapidRepair enabled?
```

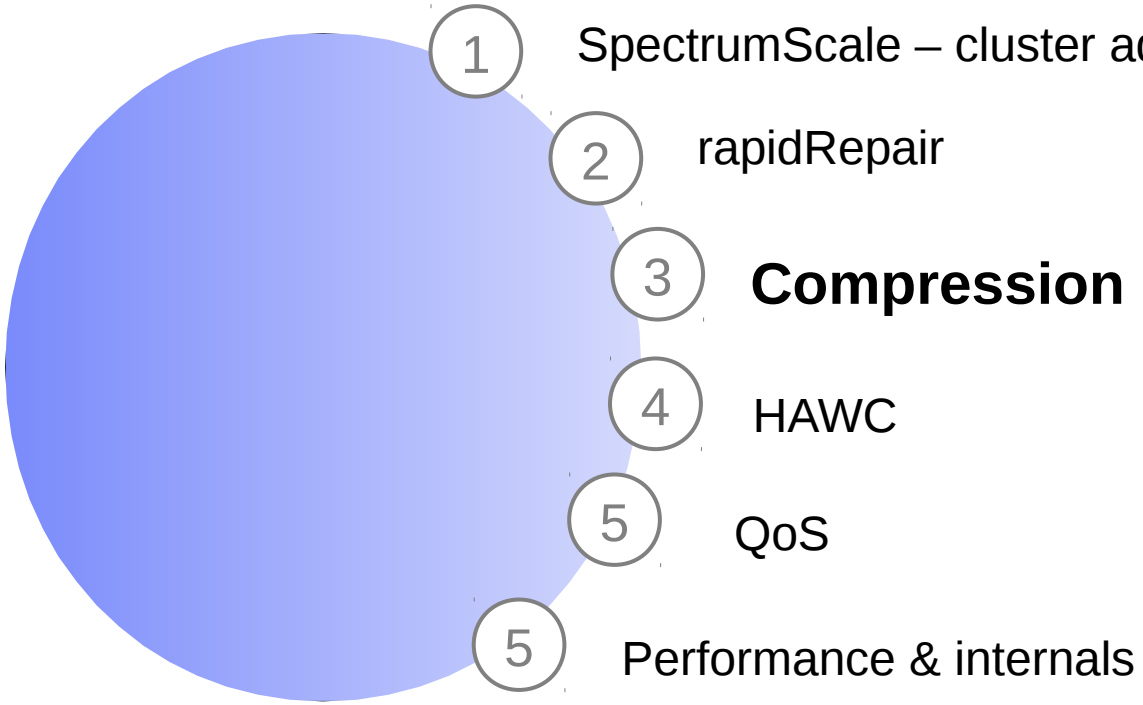
```
# mmchfs foofs --norapid-repair
hs21n07:/tmp/mmfs [Wed Nov 06 07:43:57] # mmlsfs foofs
--rapid-repair
flag              value                 description
------------------ ---------------------- --------------------------------
 --rapid-repair     no                    rapidRepair enabled?
```

```
# mmchfs foofs --rapid-repair
# mmlsfs foofs --rapid-repair
flag              value                 description
------------------ ---------------------- --------------------------------
 --rapid-repair     yes                   rapidRepair enabled?
```

Co - exists with pre-4.1

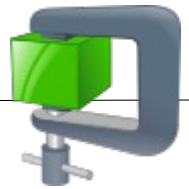
	minReleaseLevel < 4.1	minReleaseLevel >=4.1 && disable rapid-repair	miReleaseLevel >= 4.1 && enable rapid-repair
fs version < 4.1	works in old way	works in old way	"mmchfs" will fail
fs updated to 4.1	fs cannot be mounted	works in old way	works in rapid_repair way
new created 4.1 fs	fs cannot be mounted	works in rapid_repair way	works in rapid_repair way

Agenda



1. SpectrumScale – cluster administration w/o root
2. rapidRepair
3. **Compression**
4. HAWC
5. QoS
5. Performance & internals

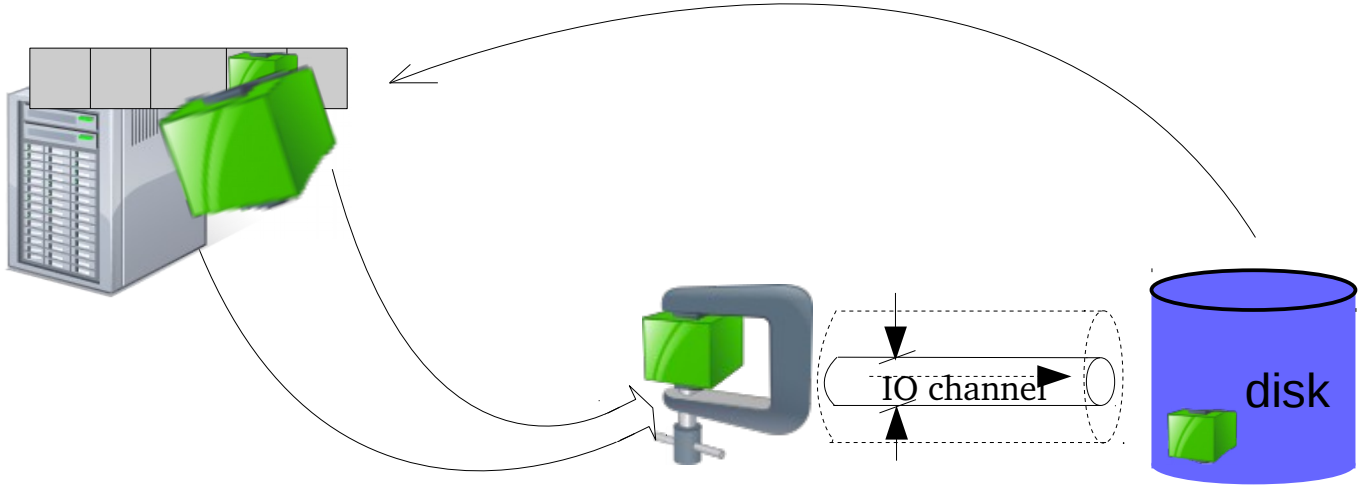
Agenda - compression



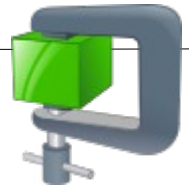
1) SpectrumScale/compression - overview
2) SpectrumScale/compression – current implementation
3) SpectrumScale/compression – demo
4) SpectrumScale/compression – closer look
5) additional considerations

Compression – goals

- Improved storage efficiency
  - Compression ratio is typically > 2x for compressable data
- Reduced IO bandwidth consumption
  - Read/write compressed data reduces load on storage backend
- Improved client-side caching
  - Caching compressed data increases apparent cache size



Compression – why in the file system



- Reduction in either the file system or the underlying storage will reduce the storage utilization
- Reduction in the file system reduces the bandwidth requirements
- Reduction in the file system increases the effective size of the cache
- File system has additional semantics to determine what, when & how to reduce
- File system can export reduced data to improve remote copy, backup, data distribution, etc

Compression: Density vs Speed

The benchmark uses the Open-Source Benchmark program by m^2 (v0.14.3) compiled with GCC v4.8.2 on Linux Mint 64-bits v17. The reference system uses a Core i5-4300U @1.9GHz. Benchmark evaluates the compression of reference Silesia Corpus in single-thread mode.

Compressor	Ratio	Compression	Decompression
memcpy	1.000	4200 MB/s	4200 MB/s
LZ4 fast 17 (r129)	1.607	**690 MB/s**	**2220 MB/s**
LZ4 default (r129)	**2.101**	**385 MB/s**	**1850 MB/s**
LZO 2.06	2.108	350 MB/s	510 MB/s
QuickLZ 1.5.1.b6	2.238	320 MB/s	380 MB/s
Snappy 1.1.0	2.091	250 MB/s	960 MB/s
LZF v3.6	2.073	175 MB/s	500 MB/s
zlib 1.2.8 -1	2.730	59 MB/s	250 MB/s
LZ4 HC (r129)	**2.720**	22 MB/s	**1830 MB/s**
zlib 1.2.8 -6	3.099	18 MB/s	270 MB/s

From https://github.com/Cyan4973/lz4

Compression – why zlib

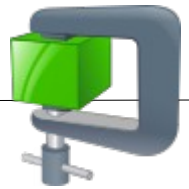
There are many compression algorithms, but

- zlib has become the defacto standard used in gzip, pkzip, png, http, ssh, among others.

- It is a lossless compression algorithm that is optimized for decompression speed and compression ratio (typically 4.5:1).

- More importantly, it is free, general purpose and legally unencumbered

Agenda - compression



1) SpectrumScale/compression - overview
2) **SpectrumScale/compression – current implementation**
3) SpectrumScale/compression – demo
4) SpectrumScale/compression – closer look
5) additional considerations

Compression – current implementation

- Reading compressed data is -*usually*- slower than reading raw data due to on-the-fly zlib uncompression overhead
  (as long no bottle neck is in the IO path)
- to be addressed by using faster algorithm (such as lz4), hardware acceleration, and other optimizations
- reduced disk IO can benefit read performance more in a busier system
- Primarily for “cold” Data Compression with this release
- Platform support: Linux, AIX, Windows

SpectrumScale/compression – how it works

two choices, to compress data

changing file attribute directly

mmchattr –compression yes file*

Policy engine

```
RULE 'shrink' MIGRATE FROM
        POOL 'datapool'
        COMPRESS('yes')
            WHERE NAME LIKE '%file%'
```

data is compressed ,
when command/policy returns

Note:

You can specify ***-I defer*** , to mark the data to be compressed,
but tell GPFS to do the compression later

SpectrumScale/compression – file attributes

...a completely compressed file looks like this:

```
node1:/gpfs/beer # mmlsattr -L 1Gfile.01.out
file name:            1Gfile.01.out
metadata replication: 1 max 2
data replication:     1 max 2
immutable:
appendOnly
flags:
storage po
fileset na
snapshot r
creation t
Misc attri
Encrypted:
node1:/gpf
```

```
node1:/# mmchattr --compression no 1Gfile.01.out
node1:/# mmchattr --compression yes -I defer 1Gfile.01.out
node1:/# mmlsattr -L 1Gfile.01.out
file name:            1Gfile.01.out
metadata replication: 1 max 2
data replication:     1 max 2
immutable:            no
appendOnly:           no
flags:                illcompressed
storage pool name:    system
fileset name:         root
snapshot name:
creation time:        Sun Feb 21 12:39:40 2016
Misc attributes:      ARCHIVE COMPRESSED
Encrypted:            no
node1:/gpfs/beer #
```

not fully compressed

SpectrumScale/compression – file attributes

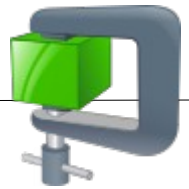
Table 4. Significance of COMPRESSED and illCompressed indicators displayed by **mmlsattr**

COMPRESSED	illCompressed	Significance
Not displayed	Not displayed	The file is not compressed. Any pending decompression is completed.
Not displayed	Displayed	The file is marked for decompression, which is deferred.
Displayed	Not displayed	The file is compressed. Any pending compression is completed.
Displayed	Displayed	The file is marked for compression. Compression is deferred, or part of the file is decompressed due to a file update or memory mapping.

use restripe facilities from GPFS to compress data

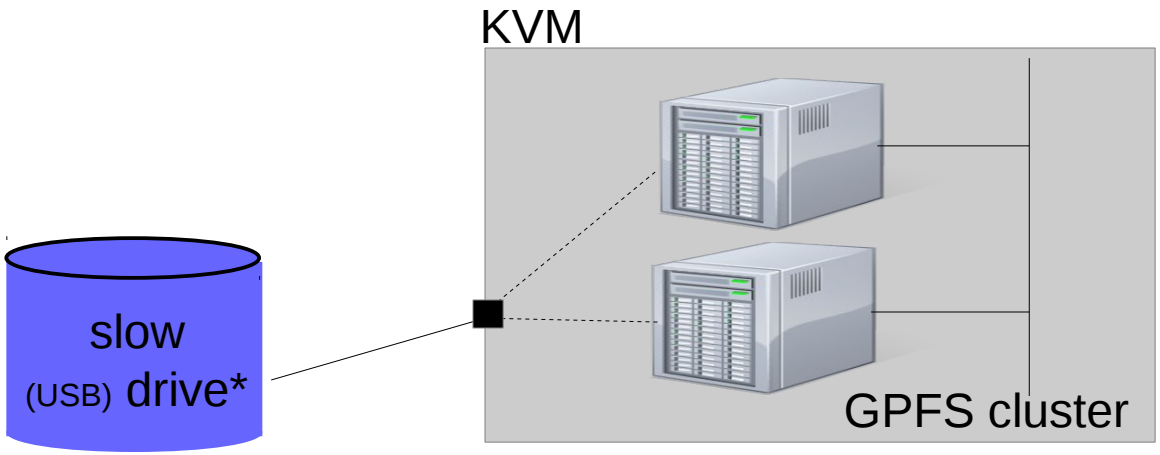
mmrestripfile -z
mmrestripfs -z

Agenda - compression



1) SpectrumScale/compression - overview
2) SpectrumScale/compression – current implementation
3) **SpectrumScale/compression – demo**
4) SpectrumScale/compression – closer look
5) additional considerations

SpectrumScale/compression – demonstration




```
node1:/gpfs/beer # dd if=/dev/zero bs=1M count=1000 of=1Gfile.01.out
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 43.1289 s, 24.3 MB/s

node1:/gpfs/beer # du -h 1Gfile.01.out
1000M   1Gfile.01.out
node1:/gpfs/beer # mmchattr --compression yes 1Gfile.01.out
node1:/gpfs/beer # du -h 1Gfile.01.out
100M    1Gfile.01.out
```

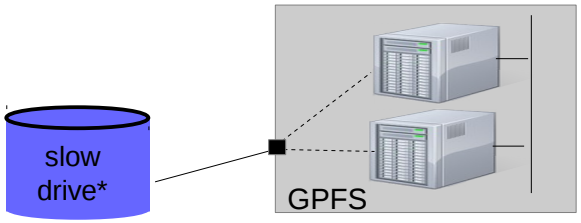
generate a file (non compressed)

Verify disk occupancy

Now, compress it

See, the result

SpectrumScale/compression – demonstration

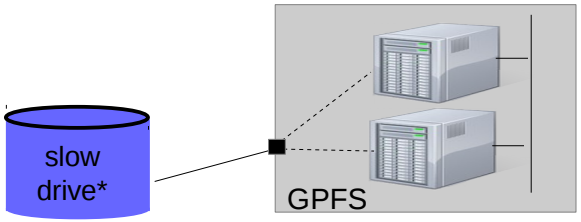


created some more files (non compressed)

node1:/gpfs/beer # dd if=/dev/zero bs=1M \
count=1000 of=1Gfile.02.out

```
node1:/gpfs/beer # ll
total 1126401
-rw-r--r-- 1 root root 1048576000 Feb 21 12:40 1Gfile.01.out
-rw-r--r-- 1 root root 1048576000 Feb 21 12:45 1Gfile.02.out
dr-xr-xr-x 2 root root      32768 Jan  1  1970 .snapshots

node1:/gpfs/beer # du -h *
100M    1Gfile.01.out
1000M   1Gfile.02.out
node1:/gpfs/beer #
```

SpectrumScale/compression – demonstration



Read performance uncompressed file:

node1:/gpfs/beer # time dd if=./1Gfile.02.out bs=1M of=/dev/null
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 37.8568 s, **27.7 MB/s**

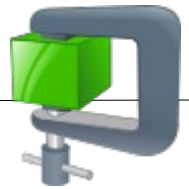
factor 8 !

Read performance compressed file:

node1:/gpfs/beer # time dd if=./1Gfile.01.out bs=1M of=/dev/null
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 4.81812 s, **218 MB/s**

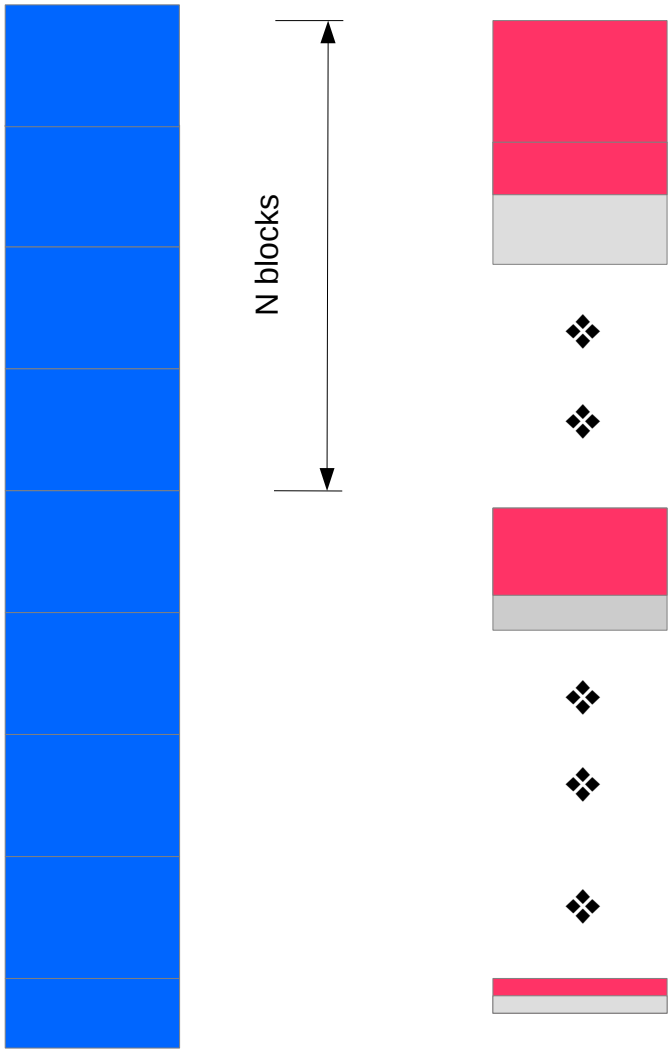
** this (poor) USB drive can deliver up to 30 MB/s , so pretending a poor designed storage back end

Agenda - compression



1) SpectrumScale/compression - overview
2) SpectrumScale/compression – current implementation
3) SpectrumScale/compression – demo
4) **SpectrumScale/compression – closer look**
5) additional considerations

Compression in GPFS



- N original data blocks (“compression group”)
  compress to 1 to N-1 compressed blocks

  Compressed blocks in middle of file
  are always full blocks (ok to have unused subblocks)

  Unused full blocks are set to NULL*

  *All compressed disk addresses include bit to indicate compression

  Any read within N blocks get BR over entire range & reads first block

  Last block can be compressed to fewer subblocks

A Closer Look At a Compressed File

```
[root@green01 doc]# ls –li  almaden_IgorRef.ps
214822 -rw-r--r-- 1 root root 1107724 Nov  2 12:55 almaden_IgorRef.ps
[root@green01 doc]# tsdbfs fs0 listda 214822
Inode 214822 snap 0 address 5:134067760 status=USERFILE indirection=1
Inode Disk Pointers:
  0:  1:2842645760    1: (null)     2:  2:3899032960    3: (null)
  4:  3:1920707840    5: (null)     6:  4:1110170752    7: (null)
  8:  1:3872143104    9: (null)    10:  2:1309923968   11: (null)
 12:  3:1536567040   13: (null)    14:  4:737554176    15: (null)
 16:  1:199757056    17: (null)    18:  2:2577588608   19: (null)
 20:  3:2397042432   21: (null)    22:  4:3541782016   23: (null)
 24:  1:1686381952   25: (null)    26:  2:1820831232   27: (null)
 28:  3:472497024    29: (null)    30:  4:3311297536   31: (null)
 32:  1:810540928    33: (null)       ...      329: (null)
[root@green01 doc]# mmchatttr --compress yes almaden_IgorRef.ps
[root@green01 doc]# tsdbfs fs0 listda 214822
Inode 214822 snap 0 address 5:134067760 status=USERFILE indirection=1
Inode Disk Pointers:
  0: C 1:837431424    1: (null)     2: C 2:3511051392   3: (null)
  4: C 3:2704355712   5: (null)     6: (znull)     7: (null)
  8: (znull)     9: (null)    10: (znull)    11: (null)
 12: (znull)    13: (null)    14: (znull)    15: (null)
 16: (znull)    17: (null)    18: (znull)    19: (null)
 20: C 3:710664960   21: (null)    22: C 4:1275351936   23: (null)
 24: (znull)    25: (null)    26: (znull)    27: (null)
 28: (znull)    29: (null)    30: (znull)    31: (null)
 32: (znull)    33: (null)       ...      329: (null)
[root@green01 doc]#
```

Compression group 1 (10 blocks)

Compression group 2 (7 blocks)

Before compression

After compression

Compression flag: compressed data block

Compression group 1 (3 blocks)

Compression group 2 (2 blocks)

ZNULL: deallocated data block

Update Driven Uncompression

[root@green01 doc]# tsdbfs fs0 listda 214822
Inode 214822 snap 0 address 5:134067760 status=USERFILE indirection=1
Inode Disk Pointers:

```
 0: C 1:837431424    1: (null)     2: C 2:3511051392   3: (null)
 4: C 3:2704355712   5: (null)     6: (znull)     7: (null)
 8: (znull)     9: (null)    10: (znull)    11: (null)
12: (znull)    13: (null)    14: (znull)    15: (null)
16: (znull)    17: (null)    18: (znull)    19: (null)
20: C 3:710664960   21: (null)    22: C 4:1275351936   23: (null)
24: (znull)    25: (null)    26: (znull)    27: (null)
28: (znull)    29: (null)    30: (znull)    31: (null)
32: (znull)    33: (null)       ...       329: (null)
```

Compression group 1 (3 blocks)

Compression group 2 (2 blocks)

Before update-in-place

[root@green01 doc]# dd if=/dev/zero of=almaden_IgorRef.ps bs=32K count=1 conv=notrunc
1+0 records in
1+0 records out
32768 bytes (33 kB) copied, 0.128166 s, 256 kB/s

After update in place

illcompressed bit is set

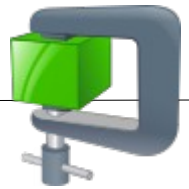
[root@green01 doc]# tsdbfs fs0 listda 214822
Inode 214822 snap 0 address 5:134067760 status=USERFILE indirection=1
Inode Disk Pointers:

```
 0:  1:1271530240    1: (null)     2:  2:2888763008    3: (null)
 4:  3:2581450368    5: (null)     6:  4:3207599872    7: (null)
 8:  1:2616023040    9: (null)    10:  2:2012921984   11: (null)
12:  3:1455917824   13: (null)    14:  4:1978349312   15: (null)
16:  1:1840058624   17: (null)    18:  2:1955300864   19: (null)
20: C 3:710664960   21: (null)    22: C 4:1275351936   23: (null)
24: (znull)    25: (null)    26: (znull)    27: (null)
28: (znull)    29: (null)    30: (znull)    31: (null)
32: (znull)    33: (null)       ...       329: (null)
```

Compression group got DECOMPRESSED (10 blocks)

Compression group 1 (2 blocks)

Agenda - compression



1) SpectrumScale/compression - overview
2) SpectrumScale/compression – current implementation
3) SpectrumScale/compression – demo
4) SpectrumScale/compression – closer look
5) **additional considerations**

DIO, Mapped Files, mmrestorefs, etc.

Direct IO

- Direct IO is turned into buffered IO on compressed files
- To help enforce direct IO, Spectrum Scale does not compress a file that is already opened Direct IO. An error is returned the compression command.

Memory Mapped File

- If the file is already compressed, upon page-in request, Spectrum Scale automatically decompresses the paged-in region and set the illcompressed flag of the file.
- Spectrum Scale does not compress a file that is already memory mapped. An error is retu in this case.

mmrestorefs: Restored data from snapshot is decompressed

- Do not compress files system (using policy, mmchattr, or mmrestripefs) during mmrestoref avoid a data loss risk (this will be fixed in PTF1).

HSM, backup, ADR, AFM: Not compression-aware

- HSM migrate/recall and backup/restore decompressed data and cause files to become illcompressed. Use mmrestripefs or mmrestripefile command to recompress the recalled/restored files.

Current Limitations

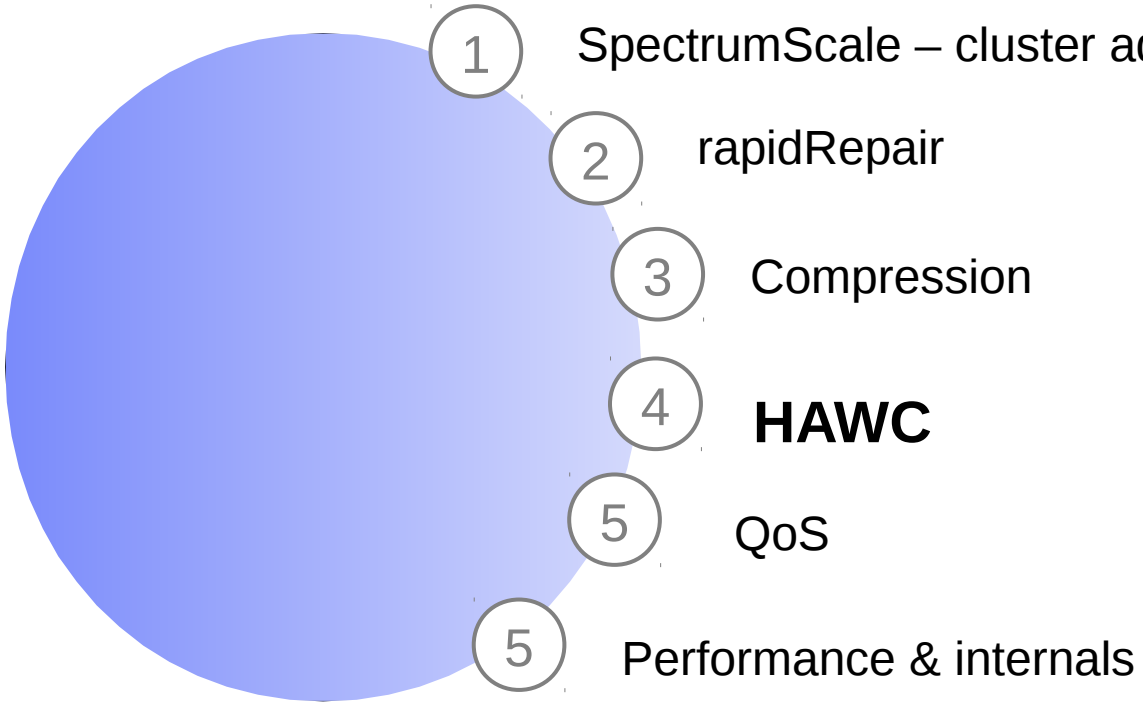
The following operations are not supported:

- Compressing files in snapshots
- Compressing a clone or clone parent
- Compressing files in an AFM cache site
- Compressing files in an FPO environment or horizontal storage pools
- Cloning a compressed file

On Windows:

- GPFS **mmapplypolicy** command is not available. Mmchattr command is available to compress/decompress.
- Compression of files in Windows hyper allocation mode is not supported.
- The following Windows APIs are not supported:
  - FSCTL_SET_COMPRESSION to enable/disable compression on a file
  - FSCTL_GET_COMPRESSION to retrieve compression status of a file
- In Windows Explorer, in the Advanced Attributes window, the compression feature is not supported.

Agenda

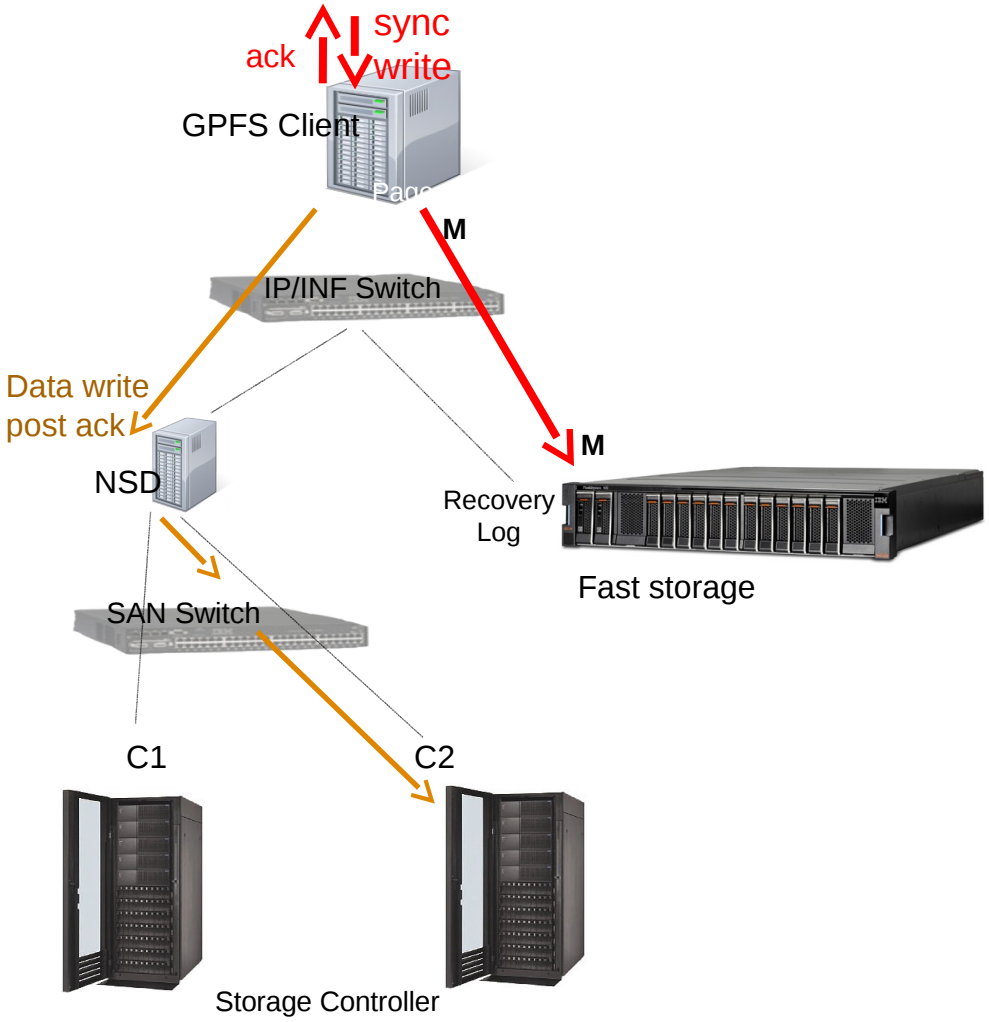


1. SpectrumScale – cluster administration w/o root
2. rapidRepair
3. Compression
4. **HAWC**
5. QoS
5. Performance & internals

HAWC - fundamentals

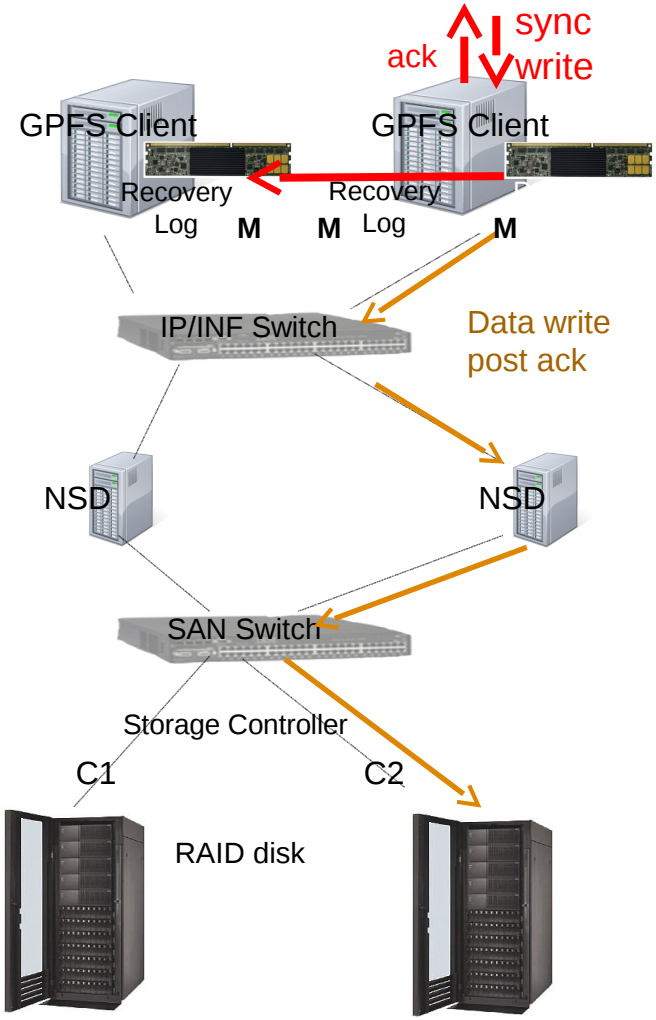
- Store recovery log in fast storage (NVRAM)
  - E.g.,Flash-backed DIMMs, Fast SSDs (Fusion-IO, etc)
- Place both metadata and small writes in recovery log
  - Log small (during fsync) writes and send large writes directly to disk
  - Allow aggregation of small writes into more efficient large write requests to storage
- Per-FS Write Size Threshold-based write policy
  - Only log write requests <= threshold
- Sparse files
  - Log writes of ‘0’s to initialize data blocks (Log compact form of offset and range)
- Use new system.log pool if faster storage available than for metadata
- Either **replicate in pairs** or store on **shared storage**
  - On node failure, quickly run recovery log
- Never read from log in critical path
  - Only read from log upon failure
- For sequential synchronous writes, allow data to be gathered and flush to storage as full blocks
- GPFS Flash optimizations
  - HAWC and LROC together resolve many latency issues with write and read path
  - Creating a “Hot File” policy with a fast storage pool improves HAWC steady state performance
    1. Write data to HAWC
    2. Write data to fast storage
    3. Down tier to slow storage

HAWC – shared reliable storage



- HAWC I/O Path
- Writeback I/O Path

HAWC – replicated across nodes



- HAWC I/O Path
- Writeback I/O Path

General Information and Limitations

- Only use when log on faster storage than primary data store
  - Otherwise data is just written twice with no benefit
- Recovery log can be with metadata or in system.log pool
- data in inode is not logged
- directory blocks are not logged
- No data or metadata is currently encrypted in the recovery log

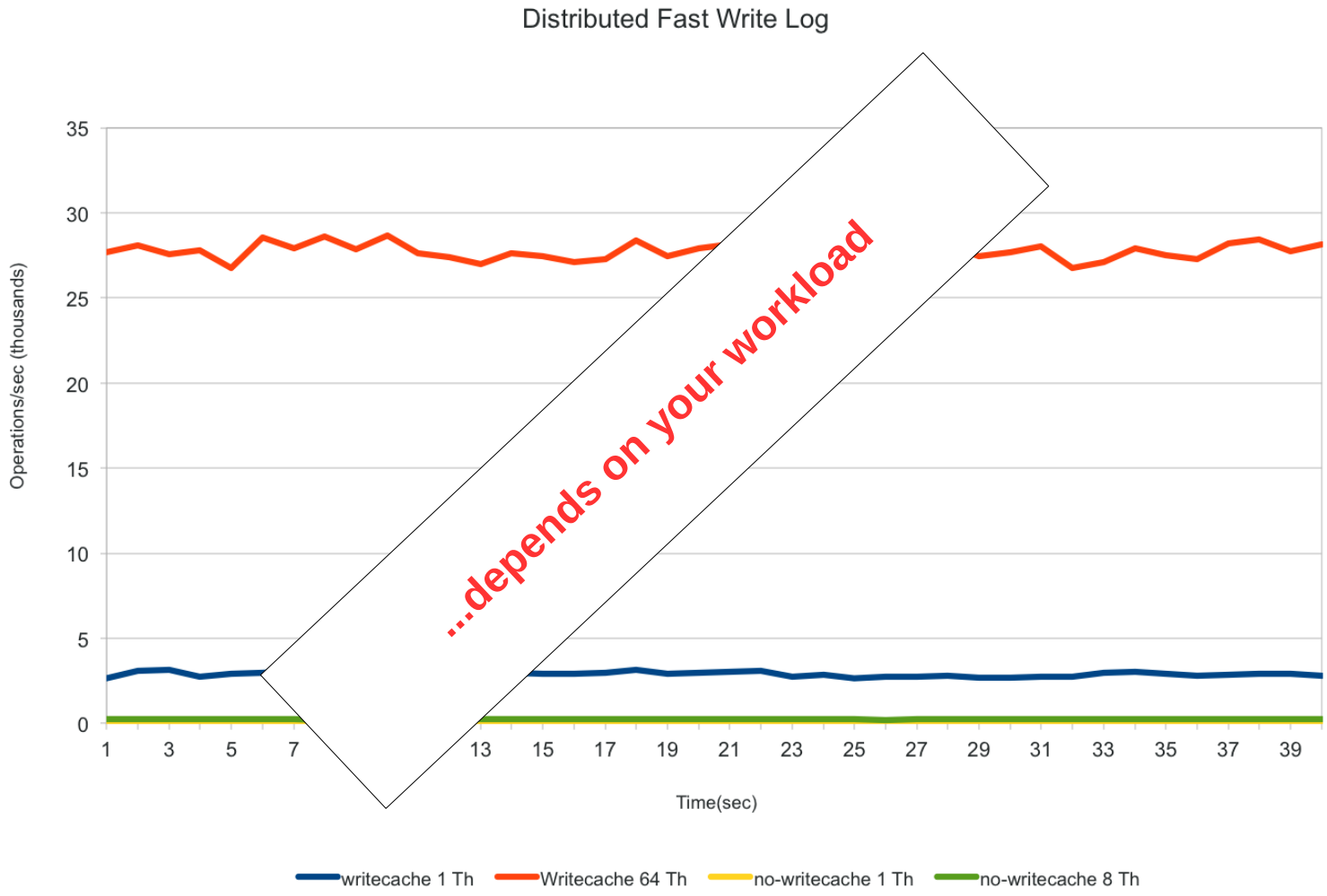
HAWC – external changes

- Created new --write-cache-threshold FS parameter
  - Can be changed during runtime (up to 64KB)
    - Increased - Values not logged will now be logged
    - Decreased - Writes will now not be logged and conflict with logged updates, which will cause a forced done record to log

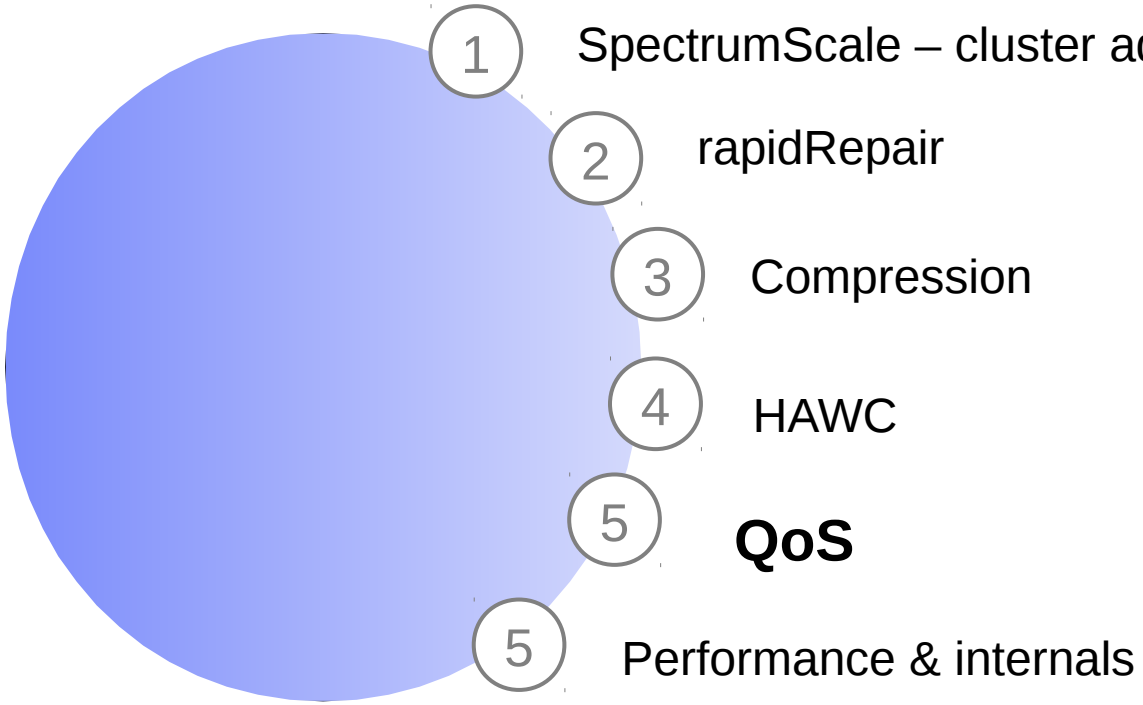
```
[root@xcat01 ~]# mmlsfs beer  --write-cache-threshold
flag                value                    description
------------------- ------------------------ -----------------------------------
 --write-cache-threshold 0                    HAWC Threshold (max 65536)
[root@xcat01 ~]#
```

```
[root@xcat01 ~]# mmlsfs beer --log-replicas
flag                value                    description
------------------- ------------------------ -----------------------------------
 --log-replicas     0                        Number of log replicas
[root@xcat01 ~]#
```

8K write performance improvement



Agenda



1. SpectrumScale – cluster administration w/o root
2. rapidRepair
3. Compression
4. HAWC
5. **QoS**
5. Performance & internals

SpectrumScale – QoS

1) QoS – new cmds / external changes

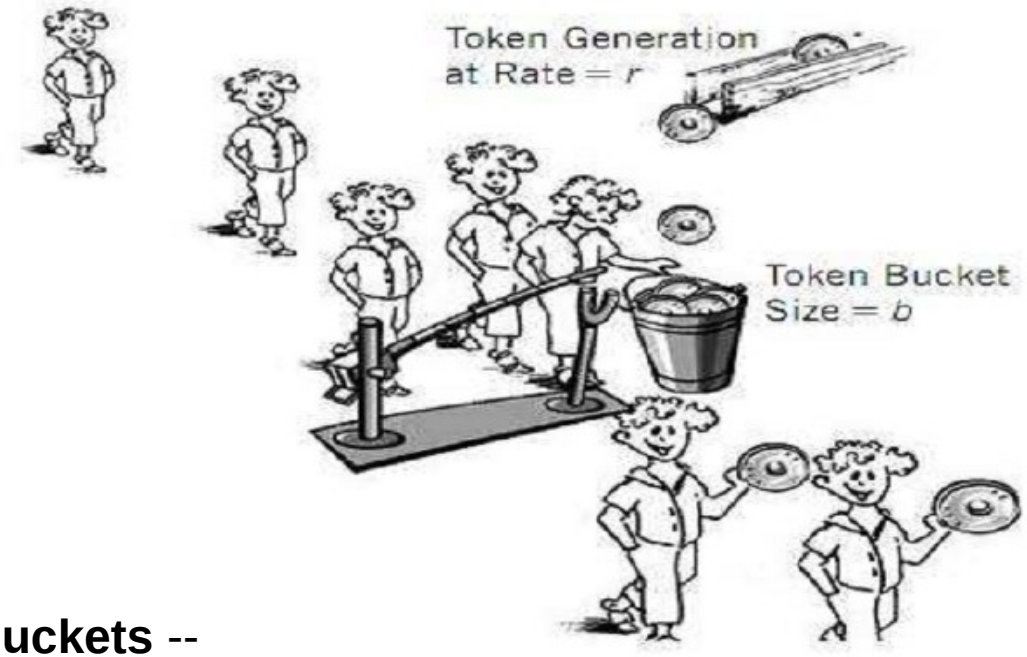
2) How it works…

3) a little example

Quality of Service – for GPFS

- GPFS has great performance, efficiency, etc, etc, but …
- Without QOS - no way to control performance of competing task/jobs:
- Restripe, backup, policy scan/ILM/HSM, rcopy
- Real Work: near-real-time decision support, data crunching

SpectrumScale – Qos , how it works



multiple token buckets --

one token bucket for each combination of:

- disk pool,
- QOS class,
- and node.

SpectrumScale/Qos: consideration

- may be used to prevent maintenance tasks from "dominating" file system performance

- Two classes of programs are currently defined: class maintenance includes long running programs like mmrestripefs or mmapplypolicy

It is perfectly okay to issue mmchqos at any time. It will affect IO completion rates but it will not “break anything”.

- to completely disable the QOS feature, issue the command `mmchqos fsname –disable`

- Valid for all FS traffic, but need to configure on the Cluster that owns the file system

Quality of Service – external changes

- To cap GPFS maintenance to 300 IOPs:

```
mmchqos <fsname> --enable
maintenance=300iops,other=unlimited,pool=*
```

```
currently supported classes:
        maintenance
        other
```

- To enable or disable QOS:

```
mmchqos <fsname> {--enable|--disable}
```

Example / outlook:

```
[root@n1 ~]# mmchqos beer --enable –classes  \
other=unlimited,maintenance=unlimited,gold=100iops,platin=1000iops,laff=1000000iops
QOS configuration has been installed and broadcast to all nodes.
[root@n1 ~]#
[root@n1 ~]# mmlsqos beer
QOS cfg::       enabled -- :pool=*,gold=100Iops,other=inf,maintenance=inf
QOS status::    throttling active, monitoring active
[root@n1 ~]#
```

Quality of Service – external changes

- To see current QOS configuration

mmlsqos <fsname>

- To see configuration and recent performance Statistics

mmlsqos <fsname> --seconds 60 –sum-nodes {yes | no}

- To plot recent performance with real-time updates

samples/charts/qosplot.pl <fsname> --trim 600 –sum-nodes { yes | no }

(requires gnuplot)

Quality of Service – new option --qos

- All potentially long running GPFS commands should accept a –**qos**

- class-of-service option and if not specified, operate as if **--qos** maintenance were specified.

- As of this writing, the following commands are treated as long running GPFS commands:

mmadddisk, mmapplypolicy, mmcheckquota, mmdefragfs, mmdeldisk, mmdelfileset, mmdelsnapshot, mmdf, mmfileid, mmfsck, mmfsctl/tsreclaim, mmlssnapshot, mmrestripefs, mmrpldisk

```
[root@n2 ~]# mmcheckquota
mmcheckquota: Missing arguments.
Usage:
  mmcheckquota [-v] [-N {Node[,Node...] | NodeFile |
NodeClass}]
                [--qos QosClass] {-a | Device [Device ...]}
```

Note: If --qos is specified explicitly on a command line, but QOS is disabled, a warning message will be issued.

SpectrumScale – QoS

1) QoS – new cmds / external changes

2) How it works…

3) a little example

Quality of Service – example

Check status of QoS :

```
node1:~ # mmlsqos beer
QOS cfg::         disabled
QOS status::      throttling inactive, monitoring inactive
```

enable QoS :

```
node1:~ # mmchqos beer --enable
Adjusted QOS Class specification: :pool=*,other=inf,maintenance=inf
QOS configuration has been installed and broadcast to all nodes.
```

Check/verify QoS :

```
node1:~ # mmlsqos beer
QOS cfg::         enabled -- :pool=*,other=inf,maintenance=inf
QOS status::      throttling active, monitoring active
```

Quality of Service – example (cont.)

Check/monitor existing IO capabilities

```
node1:~ # mmlsqos beer --seconds 30
QOS cfg::         enabled -- :pool=*,other=inf,maintenance=inf
QOS status::      throttling active, monitoring active
=== for pool system
18:48:50 other iops=100.4 ioql=0.0 qsdl=0.0 et=5
18:48:55 other iops=120.2 ioql=1.3 qsdl=0.0 et=5
[...]
```

- ...generate additional load to find out IO / IOPS limit of your Environment / Storage back ends
- Consider IO bandwidth vs IOPS

Enable throttling….

```
node1:~ # mmchqos beer --enable pool=*,maintenance=100IOPS
QOS configuration has been installed and broadcast to all
nodes.
```

Quality of Service – example (cont.)

Verify …

```
[root@n1 ~]# mmlsqos beer --seconds 60 --sum-nodes yes
QOS cfg::        enabled -- :pool=*,other=inf,maintenance=100Iops
QOS status::     throttling active, monitoring active
```

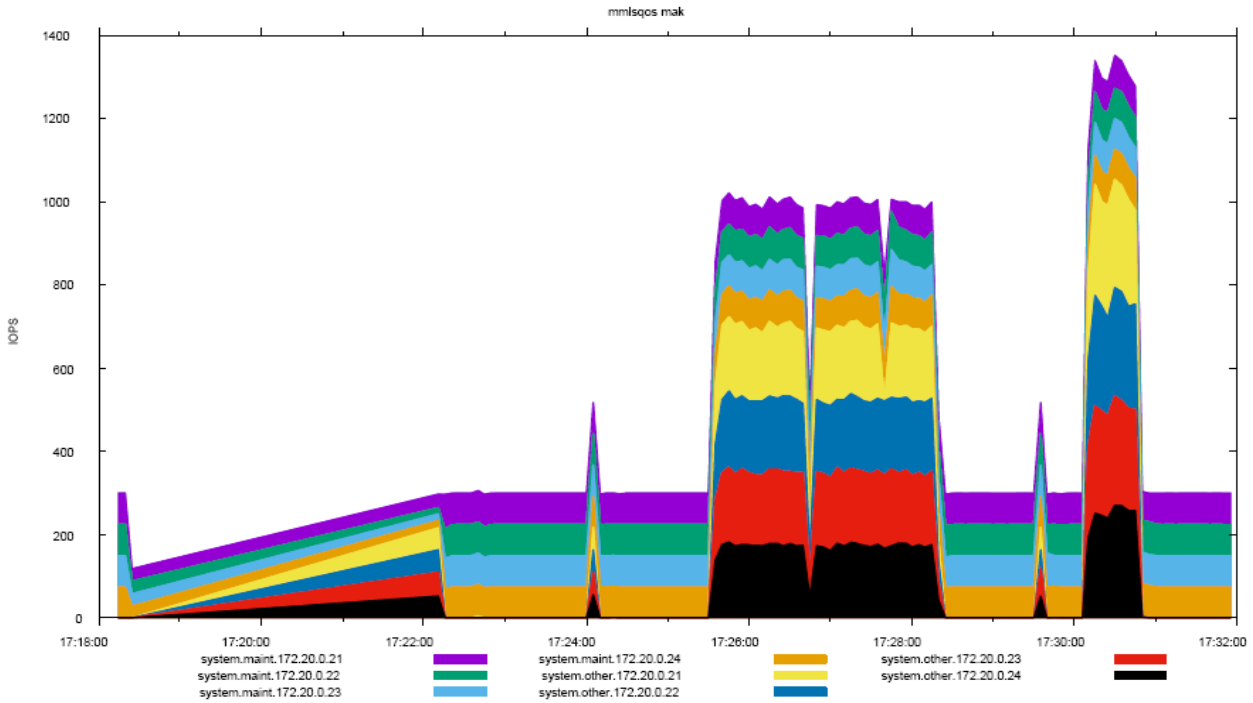
```
=== for pool system
14:34:35 maint iops=48.0 ioql=0.0 qsdl=30.5 et=5
14:34:40 maint iops=48.0 ioql=0.0 qsdl=33.0 et=5
14:34:45 maint iops=48.0 ioql=0.0 qsdl=30.5 et=5
14:34:50 maint iops=48.0 ioql=0.0 qsdl=33.0 et=5
14:34:55 maint iops=48.0 ioql=0.0 qsdl=31.3 et=5
14:35:00 maint iops=48.0 ioql=0.0 qsdl=32.1 et=5
14:35:05 maint iops=48.0 ioql=0.0 qsdl=31.3 et=5
14:35:10  gold iops=28.4 ioql=0.0 qsdl=9.9 et=5
14:35:10 other iops=319.4 ioql=0.0 qsdl=0.0 et=5
14:35:10 maint iops=48.0 ioql=0.0 qsdl=31.6 et=5
14:35:15 maint iops=90.0 ioql=0.1 qsdl=32.2 et=5
14:35:20 maint iops=50.8 ioql=0.0 qsdl=27.9 et=5
14:35:25 maint iops=48.0 ioql=0.0 qsdl=24.4 et=5
14:35:30 maint iops=7.2 ioql=0.0 qsdl=2.4 et=5
[root@n1 ~]# mmlsqos beer --seconds 60 --sum-nodes yes
QOS cfg::        enabled -- :pool=*,other=inf,maintenance=100Iops
QOS status::     throttling active, monitoring active
=== for pool datapool
09:04:15 other iops=6.2 ioql=68.0 qsdl=0.0 et=5
09:04:25 other iops=0.2 ioql=0.0 qsdl=0.0 et=5
=== for pool system
14:34:35 maint iops=48.0 ioql=0.0 qsdl=30.5 et=5
14:34:40 maint iops=48.0 ioql=0.0 qsdl=33.0 et=5
14:34:45 maint iops=48.0 ioql=0.0 qsdl=30.5 et=5
14:34:50 maint iops=48.0 ioql=0.0 qsdl=33.0 et=5
14:34:55 maint iops=48.0 ioql=0.0 qsdl=31.3 et=5
14:35:00 maint iops=48.0 ioql=0.0 qsdl=32.1 et=5
14:35:05 maint iops=48.0 ioql=0.0 qsdl=31.3 et=5
14:35:10  gold iops=28.4 ioql=0.0 qsdl=9.9 et=5
14:35:10 other iops=319.4 ioql=0.0 qsdl=0.0 et=5
14:35:10 maint iops=48.0 ioql=0.0 qsdl=31.6 et=5
14:35:15 maint iops=90.0 ioql=0.1 qsdl=32.2 et=5
14:35:20 maint iops=50.8 ioql=0.0 qsdl=27.9 et=5
14:35:25 maint iops=48.0 ioql=0.0 qsdl=24.4 et=5
14:35:30 maint iops=7.2 ioql=0.0 qsdl=2.4 et=5
[root@n1 ~]#
```

```
(root) tiger15 - Konsole <5>
File  Edit  View  Bookmarks  Settings  Help
laff@linux-e2ef:~> tig
Last login: Sun Mar  6 14:25:23 2016 from sig-9-83-62-9.evts.uk.ibm.com
[root@tiger15 ~]# ssh n1
Last login: Sun Mar  6 14:26:01 2016 from tiger15.frozen
[root@n1 ~]# mmrestripfes beer -b
-bash: mmrestripfes: command not found
[root@n1 ~]# mmrestripefs beer -b
Scanning file system metadata, phase 1 ...
Scan completed successfully.
Scanning file system metadata, phase 2 ...
Scanning file system metadata for datapool storage pool
Scan completed successfully.
Scanning file system metadata, phase 3 ...
Scan completed successfully.
Scanning file system metadata, phase 4 ...
Scan completed successfully.
Scanning user file metadata ...
   0.37 % complete on Sun Mar  6 14:33:46 2016  (    221185 inodes with total       1008 MB data
   0.70 % complete on Sun Mar  6 14:34:07 2016  (    444417 inodes with total       1880 MB data
   1.09 % complete on Sun Mar  6 14:34:29 2016  (    716801 inodes with total       2944 MB data
   1.48 % complete on Sun Mar  6 14:34:51 2016  (    987137 inodes with total       4000 MB data
   1.88 % complete on Sun Mar  6 14:35:12 2016  (   1261568 inodes with total       5071 MB data
 100.00 % complete on Sun Mar  6 14:35:25 2016  (   1532928 inodes with total       6131 MB data
Scan completed successfully.
[root@n1 ~]#
```

Quality of Service – gnuplot

node1:/usr/lpp/mmfs/samples/charts # ll
total 4
-r-xr--r-- 1 root root 3575 Nov 13 23:53 qosplot.pl
node1:/usr/lpp/mmfs/samples/charts #

mmrestripefs @300iops vs gpfsperf-mpi



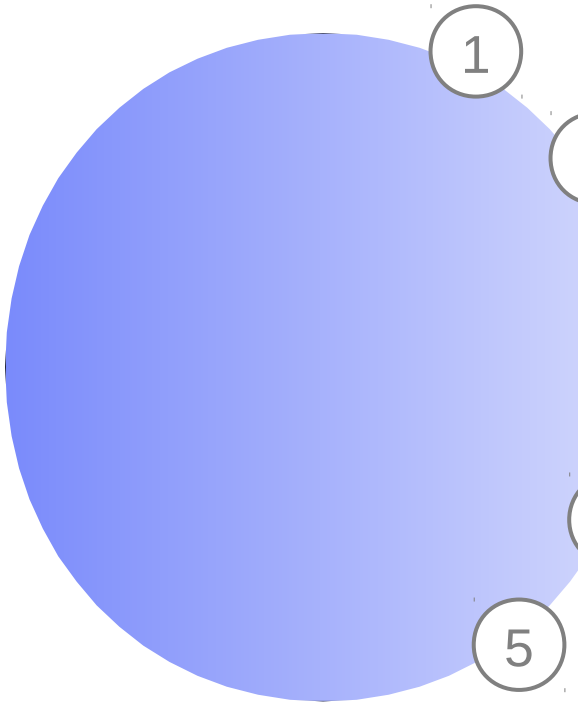
Quality of Service – other considerations

- If you have multiple storage pools….

  you should make sure the “virtual” storage LUNs of one pool do not map to the same physical devices as the LUNs of another pool.
  – You can specify different QOS IO limits for different pools using the pool keyword and appropriate syntactic delimiters of the mmchqos command.

- IOPs limits specified in a mmchqos command apply to all IO operations issued from **ALL** the nodes that have the given filesystem mounted….
  – If maintenance task(s) are not spread evenly over the nodes, then the maintenance tasks may not fully reach or exploit the specified limit.

Agenda



1. SpectrumScale – cluster administration w/o root
2. rapidRepair
3. Compression
4. HAWC
5. QoS
5. **Performance & internals**

Agenda - performance monitoring

1) overview

2) dstat gpfsops

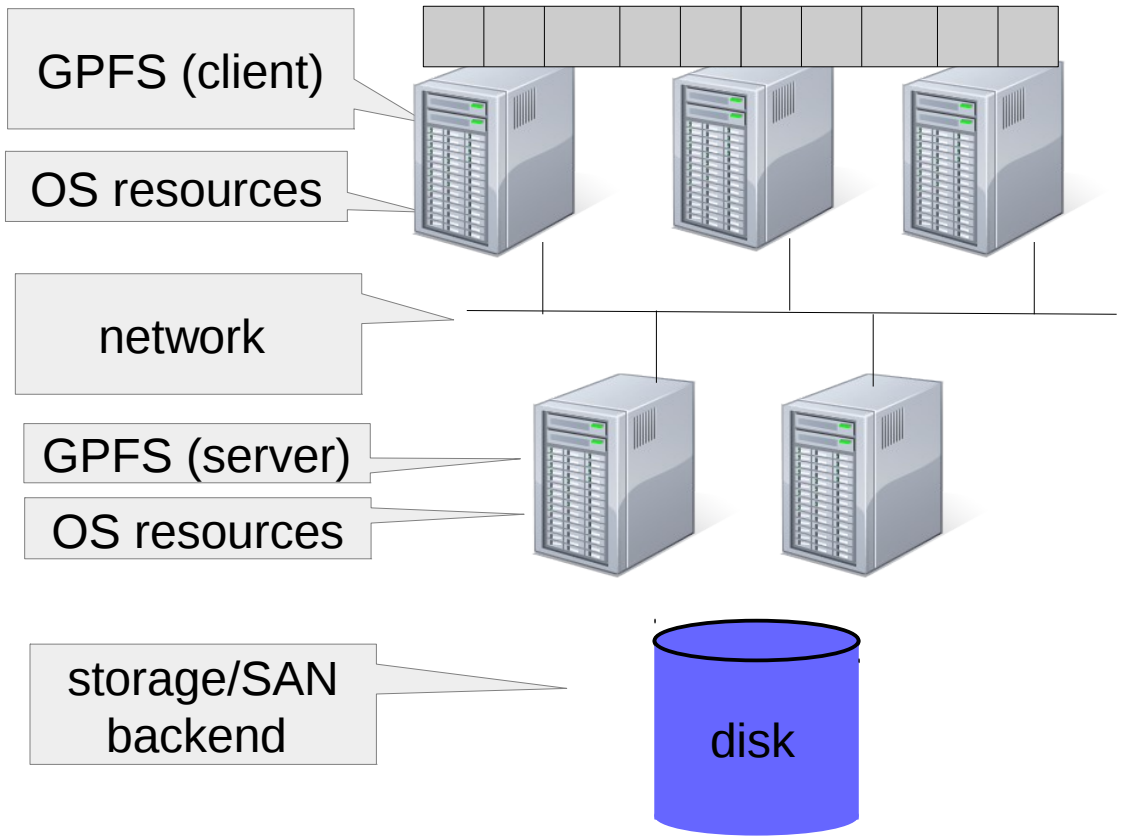
3) Zimon - implementation

4) Monitoring Performance – examples

5) miscellaneous

Performance Monitoring – goals

- identify hot spots
- take care about SLA
- find failures
- sizing



Agenda - performance monitoring

1) overview
2) dstat gpfsops
3) Zimon – introduction
4) Zimon - implementation
5) Monitoring Performance – examples
6) miscellaneous

Performance monitoring – simple entry with dstat (½)

- dstat and stat gpfsops

```
[root@gss01 util]# ls -l $PWD/dstat_gpfsops*
[]  /usr/lpp/mmfs/samples/util/dstat_gpfsops.py.dstat.0.7
[root@gss01 util]#
```

- copy it to dstat's share directory

```
# cp /usr/lpp/mmfs/samples/util/dstat_gpfsops.py.dstat.0.7  \
                                        /usr/share/dstat/dstat_gpfsops.py
```

  - Configure your environment
    DSTAT_GPFS_WHAT=vfs,ioc,nsd,vio

Performance monitoring – simple entry with dstat (2/2)

- regular dstat output ...

```
You did not select any stats, using -cdngy by default.
----total-cpu-usage---- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai hiq siq| read  writ| recv  send|  in   out | int   csw
  0   0 100   0   0   0|1213k  128k|   0     0 |   0     0 | 150   283
  0   0 100   0   0   0|   0     0 |  87k 2360B|   0     0 | 165   459
  0   0 100   0   0   0|   0     0 |  87k 1740B|   0     0 | 173   404
  0   0 100   0   0   0|   0     0 |  87k 1806B|   0     0 | 159   400
  0   0 100   0   0   0|   0     0 |  87k 1608B|   0     0 | 186   441
  0   0 100   0   0   0|   0     0 |  87k 2018B|   0     0 | 160   398
  1   0  99   0   0   0|   0   298M|  87k 1872B|   0     0 |3271  7492
  2   0  98   0   0   0|   0   585M|  87k 1674B|   0     0 |5689    13k
  2   0  98   0   0   0|   0   663M|  87k 1740B|   0     0 |4899    11k
  2   0  98   0   0   0|   0   666M|  87k 1796B|   0     0 |5187    11k
  2   0  98   0   0   0|   0   634M|  87k 1716B|   0     0 |5047    11k
```

no IB statistics here

- dstat output with gpfsops...

gpfs-vfs-ops													gpfs-disk-i/o																			gpfs-vio															
cr	del	op/cl	rd	wr	trunc	fsync	looku	gattr	sattr	other	oth r	oth w	mb rd	mb wr	steal	clean	sync	logwr	revok	pref	wrbeh	logda	rns r	rns w	lns r	lns w	vd r	vd w	pd r	pd w	logti	ClRea	ClShW	ClMdW	ClPFT	ClFTW	FlUpW	FlPFT	Migrt	Scrub	LgWr	4sCon	FixIt	LTRea	LHRea	RGDsc	Meta
0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	8	31	0	0	8	31	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	104	0	0	0	0	0	0	0	0	0	0	134	108	0	0	134	107	1072	1280	0	0	0	0	0	108	0	0	0	0	104	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	122	0	0	0	0	0	0	0	0	0	0	122	122	0	0	122	123	976	1476	0	0	0	0	0	122	0	0	0	0	124	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	129	0	0	0	0	0	0	0	0	0	0	132	132	0	0	132	132	1056	1587	0	0	0	0	0	132	0	0	0	0	130	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	151	0	0	0	0	0	0	0	0	0	0	148	148	0	0	148	148	1184	1777	0	0	0	0	0	149	0	0	0	0	151	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	141	0	0	0	0	0	0	0	0	0	0	141	140	0	0	141	140	1128	1684	0	0	0	0	0	140	0	0	0	0	139	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	139	0	0	0	0	0	0	0	0	0	0	139	140	0	0	139	140	1112	1680	0	0	0	0	0	139	0	0	0	0	140	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	145	0	0	0	0	0	0	0	0	0	0	146	146	0	0	146	146	1168	1756	0	0	0	0	0	146	0	0	0	0	147	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	93	0	0	0	0	0	0	0	0	0	0	62	88	0	0	62	88	496	1060	0	0	0	0	0	88	0	0	0	0	92	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 ^C

```
[root@gss01 ~]#
```

Agenda - performance monitoring

1) overview

2) dstat gpfsops

3) Zimon - implementation

4) Monitoring Performance – examples

5) miscellaneous

Zimon - Introduction

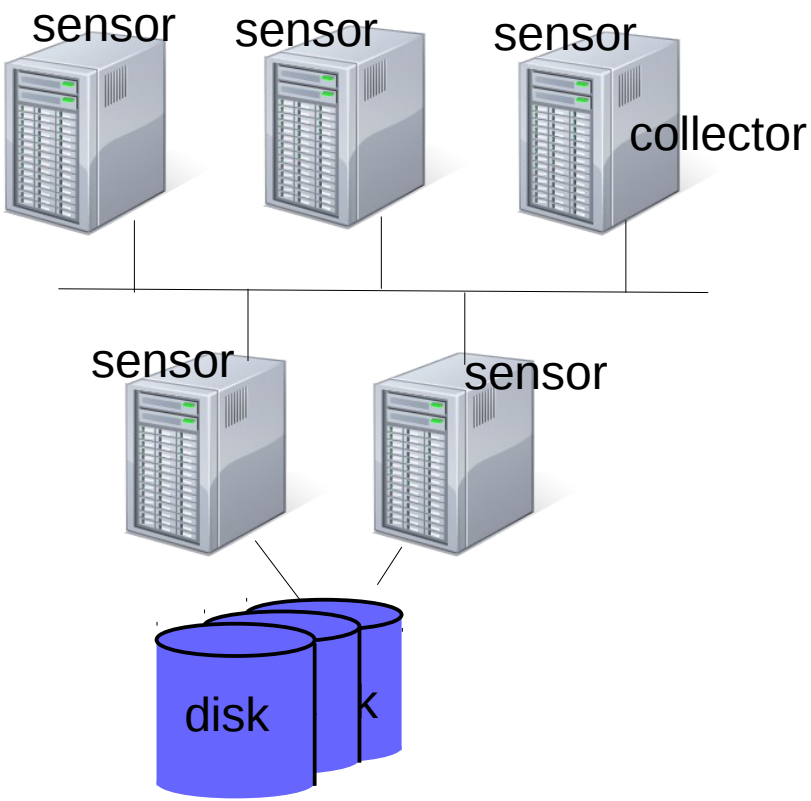
- ZIMon collectors (the back--end database of ZIMon) run independently of GPFS.
  - collect system statistics and GPFS
  - the ZIMon packages,both collector and sensors, are part of an extended GPFS distribution.
  - supported on Linux (ppc, x86_64)

- The ZIMon configuration controls individual sensors.By changing the configuration, sensors can be activated or disabled
  - parameters can be changed through a CLI
  - Changes in the configuration will be propagated to all perfmon nodes.

- Configuration is shared among all nodes in the perfmon class
  - Sections in configuration can be used for finer granularity e.g. CES, SMB, NFS ...
  - Sensors for cluster wide data need to run on a single node

- Starting, stopping and reconfiguring of ZIMon sensors is done through existing GPFS administrative commands.

Spectrum Scale – embedded monitoring and reporting



- ZIMon comes with GPFSpackage
- “all” relevant metrics
  - OS data like CPU, load
  - filesystemstatistics
  - disk statistics
- SpectrumScale 4.2
  - RHEL 6.x , RHEL7, SLES,
- ready to use w/o GUI
- Consists of sensors and collectors

SpectrumScale – performance monitoring

- Install sensor on all nodes

[root@n2 ~]# mmdsh -N all "rpm -ihv /tmp/gpfs.gss.pmsensors-4.2.0-1.el7.x86_64.rpm"

```
laff@linux-e2ef:/archive/projects/req/GPFS/4.2/perf> ls
gpfs.gss.pmcollector_4.2.0-1.D6.0.10_amd64.deb  gpfs.gss.pmcollector-4.2.0-1.SLES12.ppc64le.rpm  gpfs.gss.pmsensors-4.2.0-1.el7.ppc64le.rpm
gpfs.gss.pmcollector_4.2.0-1.D7.6_amd64.deb     gpfs.gss.pmcollector-4.2.0-1.SLES12.x86_64.rpm   gpfs.gss.pmsensors-4.2.0-1.el7.ppc64.rpm
gpfs.gss.pmcollector_4.2.0-1.D8.3_amd64.deb     gpfs.gss.pmcollector_4.2.0-1.U12.04_amd64.deb    gpfs.gss.pmsensors-4.2.0-1.el7.x86_64.rpm
gpfs.gss.pmcollector-4.2.0-1.el6.ppc64.rpm      gpfs.gss.pmcollector_4.2.0-1.U14.04_amd64.deb    gpfs.gss.pmsensors-4.2.0-1.SLES11.ppc64.rpm
gpfs.gss.pmcollector-4.2.0-1.el6.x86_64.rpm     gpfs.gss.pmsensors_4.2.0-1.D6.0.10_amd64.deb     gpfs.gss.pmsensors-4.2.0-1.SLES12.ppc64le.rp
gpfs.gss.pmcollector-4.2.0-1.el7.ppc64le.rpm    gpfs.gss.pmsensors_4.2.0-1.D7.6_amd64.deb        gpfs.gss.pmsensors-4.2.0-1.SLES12.x86_64.rpm
gpfs.gss.pmcollector-4.2.0-1.el7.ppc64.rpm      gpfs.gss.pmsensors_4.2.0-1.D8.3_amd64.deb        gpfs.gss.pmsensors_4.2.0-1.U12.04_amd64.deb
gpfs.gss.pmcollector-4.2.0-1.el7.x86_64.rpm     gpfs.gss.pmsensors-4.2.0-1.el6.ppc64.rpm         gpfs.gss.pmsensors_4.2.0-1.U14.04_amd64.deb
gpfs.gss.pmcollector-4.2.0-1.SLES11.ppc64.rpm   gpfs.gss.pmsensors-4.2.0-1.el6.x86_64.rpm
laff@linux-e2ef:/archive/projects/req/GPFS/4.2/perf>
```

- Install collector on at least one (or 2) node(s)

[root@n2 ~]# rpm -ihv /tmp/gpfs.gss.pmcollector-4.2.0-1.el7.x86_64.rpm
Preparing... ## [100%]
Updating / installing...
   1:gpfs.gss.pmcollector-4.2.0-1.el7 ## [100%]
[root@n2 ~]#

SpectrumScale – performance monitoring

- generate initial configuration

[root@n2 ~]# mmperfmon config generate --collectors n2

```
mmperfmon: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
[root@n2 ~]# Sat Feb 27 01:34:41 PST 2016: mmcommon pushSdr_async: mmsdrfs propagation started
Sat Feb 27 01:34:43 PST 2016: mmcommon pushSdr_async: mmsdrfs propagation completed; mmdsh rc=0
[root@n2 ~]#
```

 # maybe check before generate new config
[root@n2 # mmperfmon config show
mmperfmon: There is no performance monitoring configuration data.
[root@gss01 GUI]#

- activate sensores

[root@n2 ~]# mmchnode --perfmon -N all

```
Sat Feb 27 01:35:47 PST 2016: mmchnode: Processing node n2.frozen
Sat Feb 27 01:35:47 PST 2016: mmchnode: Processing node n4.frozen
Sat Feb 27 01:35:47 PST 2016: mmchnc
Sat Feb 27 01:35:47 PST 2016: mmchnc
Restarting pmsensors (via systemctl):
mmchnode: Propagating the cluster conf
  affected nodes.  This is an asynchronou
[root@n2 ~]# Sat Feb 27 01:35:48 PST 2
n5.frozen:  Restarting pmsensors (via sy
n4.frozen:  Restarting pmsensors (via sy
n3.frozen:  Restarting pmsensors (via sy
Sat Feb 27 01:35:51 PST 2016: mmcom

[root@n2 ~]#
```

```
[root@n2 corefs]# mmlscluster
[...]

 Node  Daemon node name  IP address   Admin node name  Designation
-------------------------------------------------------------------
   1   n2.frozen        172.20.0.21  n2.frozen        quorum-manager-perfmon
   3   n4.frozen        172.20.0.23  n4.frozen        manager-perfmon
   4   n5.frozen        172.20.0.24  n5.frozen        manager-perfmon
   6   n3.frozen        172.20.0.22  n3.frozen        manager-perfmon
```

SpectrumScale – collector node – consider...

[root@gss01 zimon]# ls -l $PWD/ZIMonCollector.cfg

-rw-r--r-- 1 root root 4266 Feb 27 00:46 /opt/IBM/zimon/ZIMonCollector.cfg

```
domains = {
        # this is the raw domain
        aggregation = 0         # aggregation factor for the raw domain is always 0.
#       ram = "500m"            # amount of RAM to be used
        duration = "12h"        # amount of time that data with the highest precision is kept.
        filesize = "1g" # maximum file size
        files = 5               # number of files.
},
{
        # this is the first aggregation domain that aggregates to 30 seconds
        aggregation = 30
#       ram = "800m"            # amount of RAM to be used
        duration = "1w"         # keep 30 second aggregates for 1 week.
        filesize = "1g"         # maximum file size
        files = 5               # number of files.
},
{
        # this is the second aggregation domain that aggregates to 30*30 seconds =
        aggregation = 30
#       ram = "800m"            # amount of RAM to be used
```

minimum domain

Be aware of sufficiant:

- MEM,
- CPU
- diskspace

- activate collector

[root@n2 ~]# systemctl start pmcollector

[root@n2 ~]# systemctl status pmcollector

pmcollector.service - LSB: Start the ZIMon performance monitor collector.
   Loaded: loaded (/etc/rc.d/init.d/pmcollector)
   Active: active (running) since Sat 2016-02-27 01:39:22 PST; 3s ago
  Process: 3794 ExecStart=/etc/rc.d/init.d/pmcollector start (code=exited, status=0/SUCCESS)
 Main PID: 3797 (ZIMonCollector)
   CGroup: /system.slice/pmcollector.service
           └─3797 /opt/IBM/zimon/ZIMonCollector -C /opt/IBM/zimon/ZIMonCollector.cfg -R /var/run

SpectrumScale – Zimon further configs (outlook)

- to avoid to much „expensive“ workload, restrict some measurements to single nodes… and once a day

$ mmperfmon config update GPFSDiskCap.restrict=<one node in cluster>
$ mmperfmon config update GPFSDiskCap.period=86400

- ….. if you forget it, it is disabled by default .. ;-)

```
mmperfmon config show

  {
          name = "GPFSDiskCap"
          period = 0
  },
```

SpectrumScale – zimon summary

- # install sensor

```
mmdsh -N all "rpm -ihv /tmp/gpfs.gss.pmsensors-4.2.0-1.el7.x86_64.rpm"
```

- # install collector

```
rpm -ihv /tmp/gpfs.gss.pmcollector-4.2.0-1.el7.x86_64.rpm
```

- activate config /start demons

```
mmperfmon config generate --collectors n2
mmchnode --perfmon -N all
```

on collectornode: `systemctl status pmcollector`

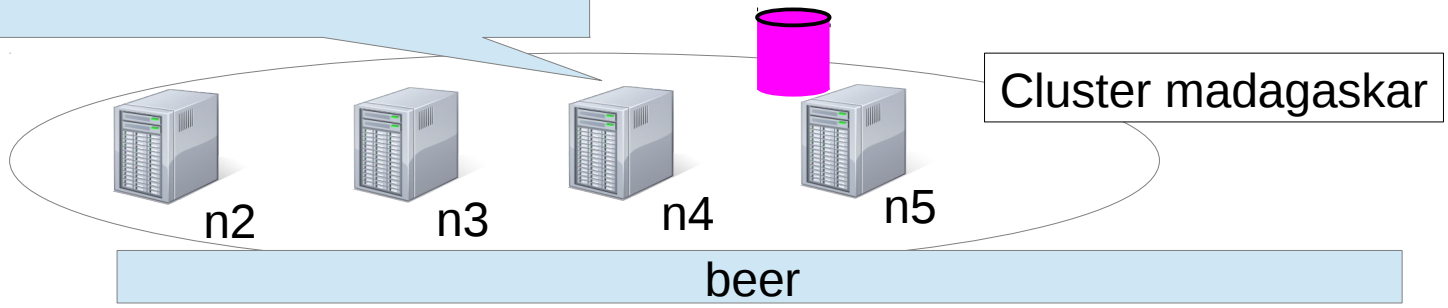
- **Ready to use**

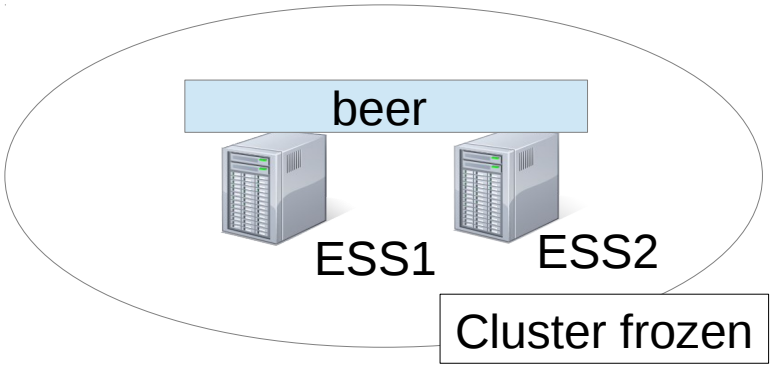
Agenda - performance monitoring

1) overview
2) dstat gpfsops
3) Zimon – introduction
4) Zimon - implementation
5) Monitoring Performance – examples
6) miscellaneous

SpectrumScale – Zimon , examples

```
[root@n4 beer]# cp file10G.1 file10G.2
[root@n4 beer]#
```






```
ou did not select any stats, using -
---total-cpu-usage---- -dsk/total- -
sr sys idl wai hiq siq| read  writ|_
 0   0 100   0   0   0|1212k  130k|
 2   0  97   0   0   0| 321M  793M|
 3   0  97   0   0   0| 707M  597M|
 2   0  97   0   0   0| 289M  773M|
 3   1  97   0   0   0| 731M  623M|
 2   0  97   0   0   0| 265M  797M|
 3   0  97   0   0   0| 771M  621M|
```

**mmperfmon query compareNodes **
gpfs_fs_bytes_written
gpfs_fs_bytes_read
…
…
…

SpectrumScale – Zimon , examples

[root@n2 ~]# mmperfmon query compareNodes gpfs_fs_bytes_written,gpfs_fs_bytes_read 20 --filter gpfs_fs_name=beer

```
File  Edit  View  Bookmarks  Settings  Help
 10 2016-02-27-08:58:41  0  0 1174405120  0  0  0  452984832  0
[root@n2 ~]# mmperfmon query compareNodes gpfs_fs_bytes_written,gpfs_fs_bytes_read 20 --filter gpfs_fs_name=beer

Legend:
 1:     n2.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_written
 2:     n3.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_written
 3:     n4.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_written
 4:     n5.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_written
 5:     n2.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_read
 6:     n3.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_read
 7:     n4.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_read
 8:     n5.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_read

Row           Timestamp n2 n3         n4 n5 n2 n3         n4 n5
  1 2016-02-27-08:58:28  0  0          0  0  0  0          0  0
  2 2016-02-27-08:58:29  0  0          0  0  0  0          0  0
  3 2016-02-27-08:58:30  0  0          0  0  0  0          0  0
  4 2016-02-27-08:58:31  0  0          0  0  0  0          0  0
  5 2016-02-27-08:58:32  0  0  285212672  0  0  0  637534208  0
  6 2016-02-27-08:58:33  0  0 1107296256  0  0  0 1946157056  0
  7 2016-02-27-08:58:34  0  0 1224736768  0  0  0 1107296256  0
  8 2016-02-27-08:58:35  0  0 1207959552  0  0  0 1157627904  0
  9 2016-02-27-08:58:36  0  0 1241513984  0  0  0 1107296256  0
 10 2016-02-27-08:58:37  0  0 1207959552  0  0  0 1124073472  0
 11 2016-02-27-08:58:38  0  0 1258291200  0  0  0 1140850688  0
 12 2016-02-27-08:58:39  0  0 1224736768  0  0  0 1124073472  0
 13 2016-02-27-08:58:40  0  0 1174405120  0  0  0  452984832  0
 14 2016-02-27-08:58:41  0  0  553648128  0  0  0          0  0
 15 2016-02-27-08:58:42  0  0          0  0  0  0          0  0
 16 2016-02-27-08:58:43  0  0          0  0  0  0          0  0
 17 2016-02-27-08:58:44  0  0          0  0  0  0          0  0
 18 2016-02-27-08:58:45  0  0          0  0  0  0          0  0
 19 2016-02-27-08:58:46  0  0          0  0  0  0          0  0
 20 2016-02-27-08:58:47  0  0          0  0  0  0          0  0
[root@n2 ~]# ^C
```

Customize output for a better overview

Show last 20 entries

can be adjusted to greater intervalls / buckets

SpectrumScale – Zimon , examples

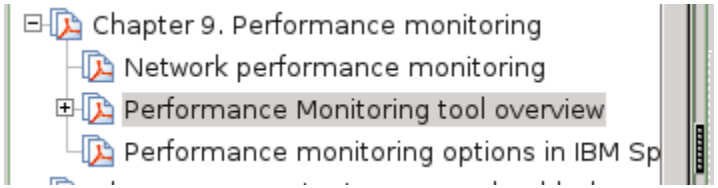
[root@n2 ~]# mmperfmon query compareNodes gpfs_fs_bytes_written,gpfs_fs_bytes_read -n 5 -b 30 --filter gpfs_fs_name=beer

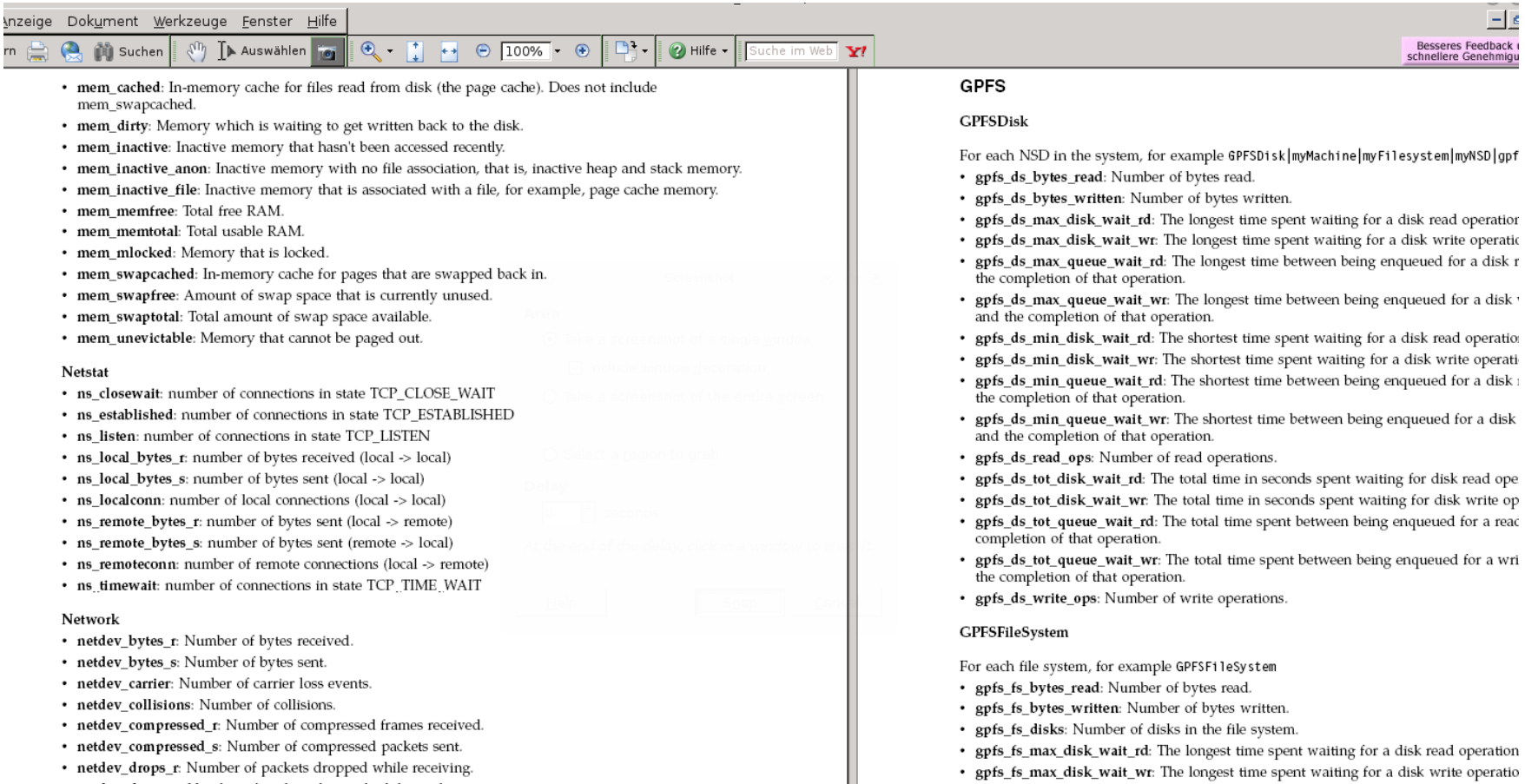
```
[root@n2 ~]# mmperfmon query compareNodes gpfs_fs_bytes_written,gpfs_fs_bytes_read -n 5 -b 3

Legend:
 1:     n2.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_written
 2:     n3.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_written
 3:     n4.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_written
 4:     n5.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_written
 5:     n2.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_read
 6:     n3.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_read
 7:     n4.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_read
 8:     n5.frozen|GPFSFilesystem|frozen.frozen|beer|gpfs_fs_bytes_read

Row           Timestamp n2         n3          n4 n5 n2         n3         n4 n5
  1 2016-02-27-09:12:30  0          0           0  0  0          0          0  0
  2 2016-02-27-09:13:00  0          0           0  0  0          0          0  0
  3 2016-02-27-09:13:30  0          0           0  0  0    1380352          0  0
  4 2016-02-27-09:14:00  0 1174413312 10486009856  0  0 2097369088 9059958784  0
  5 2016-02-27-09:14:30  0  906043392           0  0  0   83886080          0  0
[root@n2 ~]#
```

SpectrumScale – Zimon , List of queries

SpectrumScale 4.2 – AdvancedAdministration Guide …

Chapter





- **mem_cached**: In-memory cache for files read from disk (the page cache). Does not include mem_swapcached.
- **mem_dirty**: Memory which is waiting to get written back to the disk.
- **mem_inactive**: Inactive memory that hasn't been accessed recently.
- **mem_inactive_anon**: Inactive memory with no file association, that is, inactive heap and stack memory.
- **mem_inactive_file**: Inactive memory that is associated with a file, for example, page cache memory.
- **mem_memfree**: Total free RAM.
- **mem_memtotal**: Total usable RAM.
- **mem_mlocked**: Memory that is locked.
- **mem_swapcached**: In-memory cache for pages that are swapped back in.
- **mem_swapfree**: Amount of swap space that is currently unused.
- **mem_swaptotal**: Total amount of swap space available.
- **mem_unevictable**: Memory that cannot be paged out.

Netstat

- **ns_closewait**: number of connections in state TCP_CLOSE_WAIT
- **ns_established**: number of connections in state TCP_ESTABLISHED
- **ns_listen**: number of connections in state TCP_LISTEN
- **ns_local_bytes_r**: number of bytes received (local -> local)
- **ns_local_bytes_s**: number of bytes sent (local -> local)
- **ns_localconn**: number of local connections (local -> local)
- **ns_remote_bytes_r**: number of bytes sent (local -> remote)
- **ns_remote_bytes_s**: number of bytes sent (remote -> local)
- **ns_remoteconn**: number of remote connections (local -> remote)
- **ns_timewait**: number of connections in state TCP_TIME_WAIT

Network

- **netdev_bytes_r**: Number of bytes received.
- **netdev_bytes_s**: Number of bytes sent.
- **netdev_carrier**: Number of carrier loss events.
- **netdev_collisions**: Number of collisions.
- **netdev_compressed_r**: Number of compressed frames received.
- **netdev_compressed_s**: Number of compressed packets sent.
- **netdev_drops_r**: Number of packets dropped while receiving.

GPFS

GPFSDisk

For each NSD in the system, for example `GPFSDisk|myMachine|myFilesystem|myNSD|gpf`

- **gpfs_ds_bytes_read**: Number of bytes read.
- **gpfs_ds_bytes_written**: Number of bytes written.
- **gpfs_ds_max_disk_wait_rd**: The longest time spent waiting for a disk read operatior
- **gpfs_ds_max_disk_wait_wr**: The longest time spent waiting for a disk write operatic
- **gpfs_ds_max_queue_wait_rd**: The longest time between being enqueued for a disk r the completion of that operation.
- **gpfs_ds_max_queue_wait_wr**: The longest time between being enqueued for a disk and the completion of that operation.
- **gpfs_ds_min_disk_wait_rd**: The shortest time spent waiting for a disk read operatio
- **gpfs_ds_min_disk_wait_wr**: The shortest time spent waiting for a disk write operati
- **gpfs_ds_min_queue_wait_rd**: The shortest time between being enqueued for a disk the completion of that operation.
- **gpfs_ds_min_queue_wait_wr**: The shortest time between being enqueued for a disk and the completion of that operation.
- **gpfs_ds_read_ops**: Number of read operations.
- **gpfs_ds_tot_disk_wait_rd**: The total time in seconds spent waiting for disk read ope
- **gpfs_ds_tot_disk_wait_wr**: The total time in seconds spent waiting for disk write op
- **gpfs_ds_tot_queue_wait_rd**: The total time spent between being enqueued for a reac completion of that operation.
- **gpfs_ds_tot_queue_wait_wr**: The total time spent between being enqueued for a wri the completion of that operation.
- **gpfs_ds_write_ops**: Number of write operations.

GPFSFileSystem

For each file system, for example `GPFSFileSystem`

- **gpfs_fs_bytes_read**: Number of bytes read.
- **gpfs_fs_bytes_written**: Number of bytes written.
- **gpfs_fs_disks**: Number of disks in the file system.
- **gpfs_fs_max_disk_wait_rd**: The longest time spent waiting for a disk read operation
- **gpfs_fs_max_disk_wait_wr**: The longest time spent waiting for a disk write operatio

Agenda - performance monitoring

1) overview

2) dstat gpfsops

3) Zimon - implementation

4) Monitoring Performance – examples

5) miscellaneous

mmfind (1/2)

- Problem
  - mmapplypolicy good
  - interface “bad”
- Result
  - Weak adoption by customers
- Solution
  - mmfind

mmapplypolicy's wasted potential

mmapplypolicy enables the admin to run a multi-node, multi-thread scan of a file system, and to execute arbitrary searches or commands

– Super-powerful, BUT...
– Advanced admin guide: 50 pages
– /usr/lpp/mmfs/samples/ilm: 3000 lines of sample policies

mmfind (2/2)

- Every sysadmin knows the find command
- mmfind == “drop-in replacement for find”
- Not a perfect match
  - supports most of posix find and GNU find
  - some extra GPFS-specific flags

mmfind – details..

- Stage 1 translate find request to policy file (tr_findToPol.pl)
- Stage 2 run mmapplypolicy (the long part)
- Stage 3 convert the mmapplypolicy output (mmfindUtil*c)

- Performance

– Should be an inode count “tipping point” for raw “print all files”, have not found it

– Much faster if you are using -exec
(Would love some customer feedback on this)

– /usr/lpp/mmfs/samples/ilm/mmfind.README has lots of notes, examples, performance suggestions, and so on

mmfind – examples...

```
mmfind /fs1 -size +1G -a \( -user pfs001  -o \( -uid +100 -uid -200 \) \)

mmfind /p7_smallFS -ls 2>/dev/null | sort

mmfind /fs1 -nouser

mmfind /fs1 -type f -exec /bin/editFile {} \;
```

SpectrumScale – deadlock detection

- Monitor waiters in core GPFS code
- Configurable cutoff levels
- Skip waiters which can be legitimately long
  for e.g. PIT worker

```
waiting on ThCond 0x1110CDD60 (0x1110CDD60) (PitCondvar), reason 'Waiting until pit work is complete' (Long)
```

- some waiters to detect overload
  - “NSD I/O completion”
  - “waiting for exclusive use of connection for sending msg”

GPFS – debug data …

simple investigating waiters

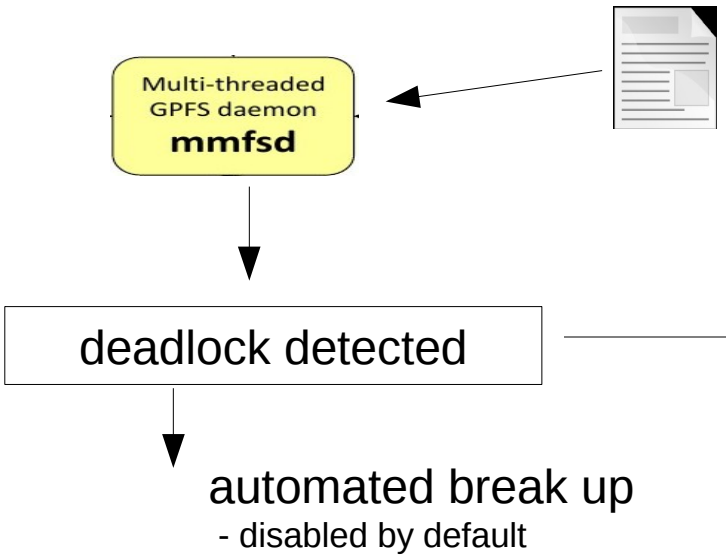
- Pending RPCs - Waiters

In this example, 192.168.1.4's communications are disrupted. 192.168.1.5 is trying to send a mount RPC to this:

```
   8.032398  17278 TRACE_TS: sgm_rpc_start(origErr 0, flags 0x0): sgMgr 192.168.1.4 err 0

   8.032401  17278 TRACE_TS: tscSend: service 00020001 msg 'sgmMsgSGMount' n_dest 1 data_len 8 msg_id 32
                            msg 0x85AC368 mr 0x85AC298

   8.032402  17278 TRACE_TS: llc_send_msg: cl 0, dest 192.168.1.4, msg_id 32, type 1, len 8

   8.032403  17278 TRACE_TS: acquireConn enter: addr 192.168.1.4 nodeidx 2 add 1
   8.032404  17278 TRACE_TS: acquireConn exit: err 0
   8.032420  17278 TRACE_TS: llc_send_msg: returning 0

   8.032421  17278 TRACE_MUTEX: Thread 0x13C02 (Mount handler) waiting on condvar 0x85AC350
                               (0x85AC350) (MsgRecord): waiting for RPC replies

===== dump waiters =====
0x855E3B0 waiting 8.269320000 seconds, Mount handler: on ThCond 0x85AC350 (0x85AC350) (MsgRecord),
          reason 'waiting for RPC replies' for sgmMsgSGMount on node 192.168.1.4

===== dump tscomm =====
Pending messages:
  msg_id   32, service 2.1, msg_type  1 'sgmMsgSGMount', n_dest 1, n_pending 1
  this 0x85AC298, n_xhold 1, ccP 0x905BB548 cbFn 0x0
    sent by 'Mount handler' (0x855E3B0)
    dest 192.168.1.4     status pending   , err 0, reply len 0
```

SpectrumScale – deadlock detection

- DeadlockDetectionThreshold
  - 0 to disable automated deadlock detection
  - !!! to enable … set a value in **seconds (default 300)**
- configurable dynamically
- Adjust according to workload to avoid false alarms in deadlock detection




```
mmfsadm dump config |grep dead
   deadlockBreakupDelay 0
   deadlockDataCollectionDailyLimit 10
   deadlockDataCollectionMinInterval 600
   deadlockDetectionThreshold 300
   deadlockDetectionThresholdIfOverloaded 1800
   [...]
```

automated debug collection
- enabled by default
- tuneable

```
on each node
internaldump.140312.22.20.24.deadlock.hs22
kthreads.140312.22.20.24.deadlock.hs22n65.
trcrpt.140312.22.20.24.deadlock.hs22n65.gz
```

Reduced PIT workload – deleting disks

1) A new option 'empty' option is introduced in the mmchdisk command, indicates that the disk will be emptied by next mmrestripefs -r/-R/-b, all data blocks allocated in the disk will be migrated off.

2) After "mmchdisk empty" or the old fashion "mmchdisk suspend", the disk status will be displayed as 'to be emptied',

3) will be displayed as 'being emptied' during the mmrestripefs -r/-R/-b, to be 'emptied' after the mmrestripefs.

4) "mmchdisk suspend" or "mmchdisk empty" on 'emptied' disks will do nothing,

5) "mmchdisk resume" will change the disk back to 'ready'.

Disk status display:

-- 'to be emptied'
--'being emptied'
-- 'emptied'

deleting disks - example

```
ps7n04:/tmp/mmfs [Fri Feb 06 04:26:54] # mmchdisk foofs empty -d gpfs1nsd
ps7n04:/tmp/mmfs [Fri Feb 06 04:27:10] # mmchdisk foofs suspend -d gpfs2nsd
ps7n04:/tmp/mmfs [Fri Feb 06 04:27:52] # mmlsdisk foofs -L
disk         driver   sector     failure holds    holds                                    storage
name         type       size       group metadata data  status        availability disk id pool          remarks
------------ -------- ------ ----------- -------- ----- ------------- ------------ ------- ------------ ---------
gpfs1nsd     nsd         512          -1 Yes      Yes   to be emptied up                  1 system
gpfs2nsd     nsd         512          -1 Yes      Yes   to be emptied up                  2 system         desc
gpfs3nsd     nsd         512          -1 Yes      Yes   ready         up                  3 system         desc
gpfs4nsd     nsd         512          -1 Yes      Yes   ready         up                  4 system         desc
Number of quorum disks: 3
Read quorum value:      2
Write quorum value:     2
Attention: Due to an earlier configuration change the file system
may contain data that is at risk of being lost.
```

```
==== start the mmrestripefs -r ====
ps7n03:/tmp/mmfs [Fri Feb 06 04:36:36] # mmrestripefs foofs -r
Scanning file system metadata, phase 1 ...
Scan completed successfully.
Scanning file system metadata, phase 2 ...
Scan completed successfully.
Scanning file system metadata, phase 3 ...
Scan completed successfully.
Scanning file system metadata, phase 4 ...
Scan completed successfully.
Scanning user file metadata ...
```

```
==== check the disk status ====
ps7n04:/tmp/mmfs [Fri Feb 06 04:37:20] # mmlsdisk foofs -L
disk         driver   sector     failure holds    holds
name         type       size       group metadata data  status        availability
------------ -------- ------ ----------- -------- ----- ------------- ------------
gpfs1nsd     nsd         512          -1 Yes      Yes   being emptied up
gpfs2nsd     nsd         512          -1 Yes      Yes   being emptied up
gpfs3nsd     nsd         512          -1 Yes      Yes   ready         up
gpfs4nsd     nsd         512          -1 Yes      Yes   ready         up
Number of quorum disks: 3
Read quorum value:      2
Write quorum value:     2
Attention: Due to an earlier configuration change the file system
may contain data that is at risk of being lost.
```

deleting disks - example

```
==== suspend or empty 'emptied' disks will do nothing ====
ps7n03:/tmp/mmfs [Fri Feb 06 05:07:26] # mmchdisk foofs suspend -d gpfs1nsd
ps7n03:/tmp/mmfs [Fri Feb 06 05:07:35] # mmchdisk foofs empty -d gpfs2nsd
ps7n03:/tmp/mmfs [Fri Feb 06 05:07:42] # mmlsdisk foofs -L
disk         driver   sector     failure holds    holds                                     storage
name         type       size       group metadata data  status        availability disk id pool         remarks
------------ -------- ------ ----------- -------- ----- ------------- ------------ ------- ------------ ---------
gpfs1nsd     nsd         512          -1 Yes      Yes   emptied       up                 1 system
gpfs2nsd     nsd         512          -1 Yes      Yes   emptied       up                 2 system        desc
gpfs3nsd     nsd         512          -1 Yes      Yes   ready         up                 3 system        desc
gpfs4nsd     nsd         512          -1 Yes      Yes   ready         up                 4 system        desc
Number of quorum disks: 3
Read quorum value:      2
Write quorum value:     2
```

```
==== resume 'emptied' disks will change the disk back to 'ready' ====
ps7n03:/tmp/mmfs [Fri Feb 06 05:07:48] # mmchdisk foofs resume -d gpfs1nsd
ps7n03:/tmp/mmfs [Fri Feb 06 05:08:31] # mmlsdisk foofs -L
disk         driver   sector     failure holds    holds                                     storage
name         type       size       group metadata data  status        availability disk id pool         remarks
------------ -------- ------ ----------- -------- ----- ------------- ------------ ------- ------------ ---------
gpfs1nsd     nsd         512          -1 Yes      Yes   ready         up                 1 system        desc
gpfs2nsd     nsd         512          -1 Yes      Yes   emptied       up                 2 system
gpfs3nsd     nsd         512          -1 Yes      Yes   ready         up                 3 system        desc
gpfs4nsd     nsd         512          -1 Yes      Yes   ready         up                 4 system        desc
Number of quorum disks: 3
Read quorum value:      2
Write quorum value:     2
```

Inode expansion – optimized...

– Inode expansion When you use up enough inodes (i.e. files), GPFS allocates more.

```
#08:03:24# hs22n56:~ # mmdf expFS -F
Inode Information
-----------------
Total number of used inodes in all Inode spaces:  99999
Total number of free inodes in all Inode spaces: 1
Total number of allocated inodes in all Inode spaces: 100000
Total of Maximum number of inodes in all Inode spaces: 200000
```

– GPFS dynamic Inode expansion

```
Sat May 30 00:55:45.945 2015: [N] Expanding inoexpLoop_time1432751044_pid17301796 inode space 4
current 52096 inodes (0 free) by 12800
Sat May 30 00:55:48.756 2015: [N] Expanded inoexpLoop_time1432751044_pid17301796 inode space 4 from
52096 to
64896 inodes.
```

– Trigger expansion via:

```
mmchfs -inode-limit X:NumToPrealloc
```

Inode expansion – optimized...

– “Old” inode expansion was expensive
  Requires very strong locks, blocking current users of the inode ranges in question until it completes...

– Large inode counts == long waiters

Solution and effects

- Try to decrease cost and duration of locks
- New FS automatically has the new version
- Old FS requires mmchfs -V full;
  the next expansion will trigger an implicit (i.e. invisible) conversion from old-style inode map to new-style inode map

–
This will be “somewhat expensive” the first time

Agenda - performance monitoring

1) overview

2) dstat gpfsops

3) Zimon – introduction

4) Zimon - implementation

5) Monitoring Performance – examples

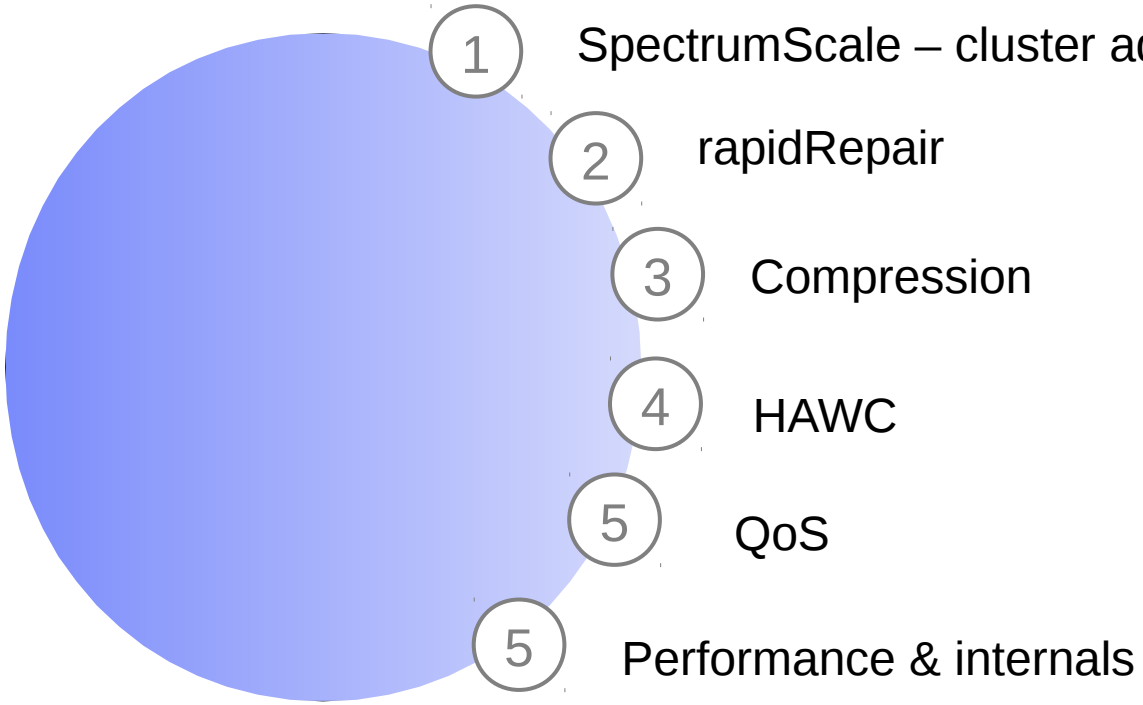
6) Miscellaneous

    ….

    ….

    ….

Agenda



1. SpectrumScale – cluster administration w/o root
2. rapidRepair
3. Compression
4. HAWC
5. QoS
5. Performance & internals

olaf.weiser@de.ibm.com
IBM Deutschland
SpectrumScale Support Specialist

Disclaimer

- The information in this document may be **IBM CONFIDENTIAL**.
- This information is provided on an "AS IS" basis without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Some jurisdictions do not allow disclaimers of express or implied warranties in certain transactions; therefore, this statement may not apply to you.
- This information is provided for information purposes only as a high level overview of possible future products. PRODUCT SPECIFICATIONS, ANNOUNCE DATES, AND OTHER INOFORMATION CONTAINED HEREIN ARE SUBJECT TO CHANGE AND WITHDRAWAL WITHOUT NOTICE.
- USE OF THIS DOCUMENT IS LIMITED TO SELECT IBM PERSONNEL AND TO BUSINESS PARTNERS WHO HAVE A CURRENT SIGNED NONDISCLUSURE AGREEMENT ON FILE WITH IBM. THIS INFORMAITON CAN ALSO BE SHARED WITH CUSTOMERS WHO HAVE A CURRENT SIGNED NONDISCLOSURE AGREEMENT ON FILE WITH IBM, BUT THIS DOCUMENT SHOULD NOT BE GIVEN TO A CUSTOMER EITHER IN HARDCOPY OR ELECTRONIC FORMAT.
- Important notes:
- IBM reserves the right to change product specifications and offerings at any time without notice. This publication could include technical inaccuracies or typographical errors. References herein to IBM products and services do not imply that IBM intends to make them available in all countries.
- IBM makes no warranties, express or implied, regarding non-IBM products and services, including but not limited to Year 2000 readiness and any implied warranties of merchantability and fitness for a particular purpose. IBM makes no representations or warranties with respect to non-IBM products. Warranty, service and support for non-IBM products is provided directly to you by the third party, not IBM.
- All part numbers referenced in this publication are product part numbers and not service part numbers. Other part numbers in addition to those listed in this document may be required to support a specific device or function.
- MHz / GHz only measures microprocessor internal clock speed; many factors may affect application performance. When referring to storage capacity, GB stands for one billion bytes; accessible capacity may be less. Maximum internal hard disk drive capacities assume the replacement of any standard hard disk drives and the population of all hard disk drive bays with the largest currently supported drives available from IBM.
- IBM Information and Trademarks
- The following terms are trademarks or registered trademarks of the IBM Corporation in the United States or other countries or both: the e-business logo, IBM, xSeries, pSeries, zSeries, iSeries.
- Intel, Pentium 4 and Xeon are trademarks or registered trademarks of Intel Corporation. Microsoft Windows is a trademark or registered trademark 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.