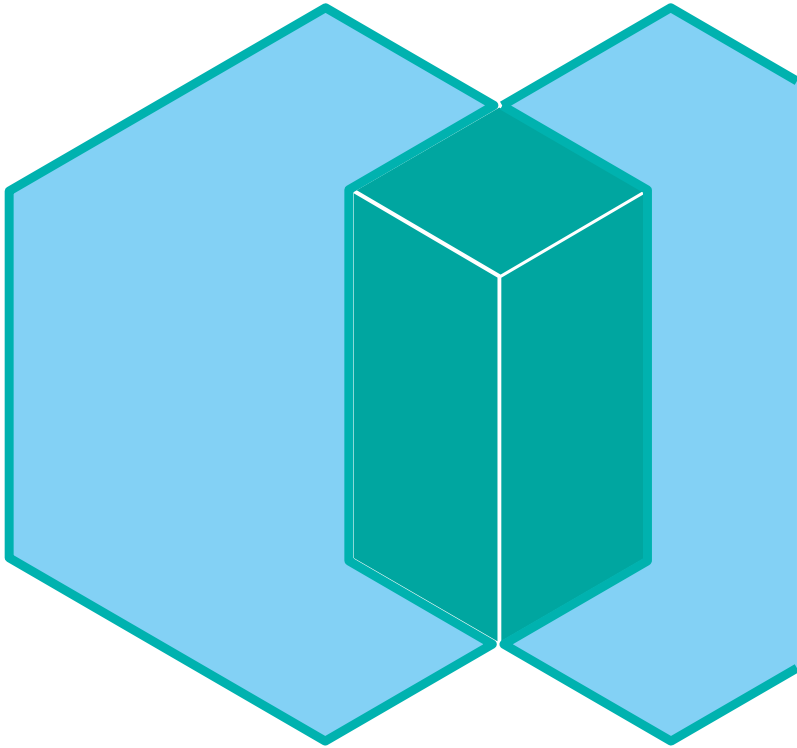
“Field Update”

Achim Rehor, Spectrum Scale and ESS Support EMEA



IBM Spectrum Scale Support Global Time Zone Coverage

- Global team locations
  - Poughkeepsie, NY USA
  - Raleigh, NC USA
  - Toronto, ON Canada
  - Kelsterbach, Germany
  - Pune, Bangalore, India
  - Beijing, China

Agenda

- Interesting cases
  - Huge Clusters and their specific issues
  - `No space left on device` with just 4% occupied?
  - expels over and over ;)
  - ESS Power LE node hang due to oom
- Quality Guild
  - What is that ? What is our approach with that Quality Guild
  - Outcome : some examples
- Enhancements, Improvements , Tipps ...
  - Enhancements in 5.0.0 – 5.0.2
  - Links and Questions

Interesting cases 1: Fileset creation slowness

- **Background**
  - Usage Modell Change: independent filesets
  - More granular management
- **problem(s)**
  - # of filesets x runtime : increasing : #1 ~25 sec , #322 (out of 920) > 30 min
  - gpfs.snap runtime
  - Asserts
- **findings and trials to resolve**
  - expand inodes takes longer and longer
  - Enlarge pagepool ➔ no improvement
  - Disk issues in the DSS ➔ wasn't the culprit
  - lessen # mounts ➔ no improvement

```
2018-08-27_06:36:26.775+0200: [I] Command: tscrfileset /dev/project cvsk21 --inode-space new --inode-limit 3720000
                                           -t Fileset for project cvsk21
==========================================
2018-08-27_06:36:26.978+0200: [N] Expanding project inode space 322 current 0 inodes (0 free) by 3723264
2018-08-27_07:06:08.778+0200: [N] Expanded project inode space 322 from 0 to 3723264 inodes (on-demand).
==========================================
expanding time: 30 min
2018-08-27_07:06:14.016+0200: [I] Fileset cvsk21 created with id 322 root inode 345744867331.
2018-08-27_07:06:14.016+0200: [I] Command: successful tscrfileset
```

Interesting cases 1: Fileset creation slowness

- **findings and trials to resolve**
  - 're-initializing the inode manager‘ hotspot
  - ”Inode Allocatio Map File” size ~555GB in inode 2

```
InodeNum:2 (Inode Allocation Map):
...
 file size 555141300224 (0x8141000000), indirectionLevel 2, nFullBlocks 66178 inode clean
```

  - -n numnodes = max (16k)
  - Why is that causing grief ? ➔ InodeManagerInitialization
    - upto now: resetInodeManager() reads entire file under “ro” lock
    - changed : avoid reading the inode map file to update the inode manager free counts.
      Instead new inodes are calculated for each segment and counted towards free space
- **Fix** ➔ Defect 1070443 : 5.0.1.1 efix11 (APAR IJ11105)
  - Fileset create time (including inode expansion) < 4 min

Interesting cases 1: Fileset creation slowness

CMVC Defect 1070443 :

TS001340628: inode expansion is very slow on large cluster

As part of inode expansion v2, after adding new segments to the inode and inode map file, SGInodeSpaceMap::Expand() calls resetInodeManager() which essentially reads the entire inode map file under ro lock to update the inode manager inode free counts.

Since inode map file can be very large for large clusters or for mmcrfs with large -n option, and since resetInodeManager() reads individual inode map records which constitute small read I/Os, this is a performance bottleneck.

Fix is to **avoid reading the inode map file** to update the inode manager free counts. Instead the newly expanded inodes are calcualated for each segment and counted towards free space.

To avoid this value from being overwritten by inode free count updates from nodes that have not yet seen the expanded inodes, **allocMsgUpdateISegment** RPC is modified to also send the maximum inodes seen by the sending node for a inode map segment.

When running with back level nodes in the cluster, this fix can cause inode manager to report less free inodes than actual. This is a temporary condition that gets corrected during subsequent updates to the inode map segment or during inode manager recovery.

Interesting cases 1: Fileset creation slowness

- **And now ? …**
- **More problems: mmchmgr takes too long → waiters for SGInodeMapMutex**
- **improving inode manager initialization times**
  - SGmgr move → reset inode allocation manager
    - same root cause: Read of InodeAllocMapFile at initialization under ro lock by Segment
    - now: reading full blocks, no locks
      - → quickly initialized inode manager with free inode count per segment
      - → may contain stale or incorrect accounting (note: read with no locks)
        asynchronously updated during inode alloc map scan while async recovery
- **Fix** → Defect 1074974 Speed up inode alloc manager initialization: 5.0.1.1 efix17
- Randproblem 1: Long recovery and ACL related waiters after daemon assert
  Defect 1074954 fixed in efix 20
- Randproblem 2: potential deadlock beim Mount und QoS Initialisierung
  Defect 1074382 fixed in efix 22
- **Preview**: SGAsyncRecovery during mmchmgr still ‘slow
  → doesnt cause hangs anymore, though runtime is inacceptable

Interesting cases 2 : `No space left on device` with only 4% occupied?

```
[root@pfecl1n1]/gpfs/cl1fs1/kai: mmlsfs cl1fs1
flag                value                    description
------------------- ------------------------ -----------------------------------
 -f                 8192                     Minimum fragment size in bytes
 -i                 4096                     Inode size in bytes
 -I                 32768                    Indirect block size in bytes
 -m                 1                        Default number of metadata replicas
 -r                 1                        Default number of data replicas
 -j                 scatter                  Block allocation type
 -B                 262144                   Block size
 -V                 17.00 (4.2.3.0)          File system version
 -L                 4194304                  Logfile size
 --inode-limit      8388608                  Maximum number of inodes
 --is4KAligned      Yes                      is4KAligned?
 --subblocks-per-full-block 32               Number of subblocks per full block
 -P                 system;data              Disk storage pools in file system
 -d                 SYS3;SYS4;SYS5;DAT3;DAT4;DAT5  Disks in file system
 -T                 /gpfs/cl1fs1             Default mount point
```

Creation loop

```
i=0
while true
do
  i=$[$i+1]
  dd if=/dev/zero of=/gpfs/cl1fs1/kai/$i bs=4K count=1 2>/dev/null
  echo $? $i written
done
```

Interesting cases 2 : No space left on device with only 4% occupied?

```
[root@pfecl1n1]/gpfs/cl1fs1/kai: mmdf cl1fs1
disk                disk size  failure holds    holds              free KB             free KB
name                    in KB    group metadata data        in full blocks        in fragments
--------------- ------------- -------- -------- ----- -------------------- -------------------

Disks in storage pool: system (Maximum disk size allowed is 288 GB)
SYS3                 20971520        1 Yes      No           9494784 ( 45%)           472 ( 0%)
SYS4                 20971520        2 Yes      No           9499392 ( 45%)           696 ( 0%)
SYS5                 20971520        3 Yes      No           9497344 ( 45%)           632 ( 0%)
             -------------                         -------------------- -------------------
(pool total)         62914560                           28491520 ( 45%)          1800 ( 0%)

Disks in storage pool: data (Maximum disk size allowed is 288 GB)
DAT3                 20971520        1 No       Yes                 0 (  0%)          7696 ( 0%)
DAT4                 20971520        2 No       Yes                 0 (  0%)          6104 ( 0%)
DAT5                 20971520        3 No       Yes                 0 (  0%)          7584 ( 0%)
             -------------                         -------------------- -------------------
(pool total)         62914560                                0 (  0%)         21384 ( 0%)
             =============                         ==================== ===================

(data)               62914560                                0 (  0%)         21384 ( 0%)
(metadata)           62914560                           28491520 ( 45%)          1800 ( 0%)
             =============                         ==================== ===================

(total)             125829120                           28491520 ( 23%)         23184 ( 0%)

Inode Information
-----------------
Number of used inodes:          7841056
Number of free inodes:           547552
Number of allocated inodes:     8388608
Maximum number of inodes:       8388608
```

```
[root@pfecl1n1]/gpfs/cl1fs1/kai: df /gpfs/cl1fs1/
Filesystem     1K-blocks     Used Available Use% Mounted on
cl1fs1          62914560 62893312     21248 100% /gpfs/cl1fs1

[root@pfecl1n1]/gpfs/cl1fs1/kai: df -h /gpfs/cl1fs1/
Filesystem      Size  Used Avail Use% Mounted on
cl1fs1           60G   60G   21M 100% /gpfs/cl1fs1

[root@pfecl1n1]/gpfs/cl1fs1/kai: ll 7836989
-rw-r--r-- 1 root root 4096 Apr 28 01:19 7836989
[root@pfecl1n1]/gpfs/cl1fs1/kai: ll 7836990

ls: cannot access 7836990: No such file or directory
```

Deletion loop

```
#!/bin/bash
# Script to delete files on subblocks 2-32

f=0
counter=1

while [ $f -le 7836989 ]; do
        counter=1
        f=$[$f+1]

                while [ $counter -lt 32 ]
                do
                ((f++))
#               echo will delete file: $f
                rm -rf $f
                ((counter++))
        done
done
```

Interesting cases 2 : `No space left on device` with only 4% occupied?

After Deletion:

```
[root@pfecl1n1]/gpfs/cl1fs1/kai: mmdf cl1fs1
disk                disk size  failure holds    holds              free KB             free KB
name                    in KB    group metadata data        in full blocks        in fragments
--------------- ------------- -------- -------- ----- -------------------- -------------------
Disks in storage pool: system (Maximum disk size allowed is 288 GB)
SYS3                 20971520        1 Yes      No          9656064 ( 46%)           472 ( 0%)
SYS4                 20971520        2 Yes      No          9657088 ( 46%)           472 ( 0%)
SYS5                 20971520        3 Yes      No          9656320 ( 46%)           632 ( 0%)
                -------------                         -------------------- -------------------
(pool total)         62914560                           28969472 ( 46%)          1576 ( 0%)

Disks in storage pool: data (Maximum disk size allowed is 288 GB)
DAT3                 20971520        1 No       Yes           56576 (  0%)      20196056 (96%)
DAT4                 20971520        2 No       Yes           56576 (  0%)      20196152 (96%)
DAT5                 20971520        3 No       Yes           58624 (  0%)      20194312 (96%)
                -------------                         -------------------- -------------------
(pool total)         62914560                          171776 (  0%)      60586520 (96%)
                =============                         ==================== ===================
(data)               62914560                          171776 (  0%)      60586520 (96%)
(metadata)           62914560                        28969472 ( 46%)          1576 ( 0%)
                =============                         ==================== ===================
(total)             125829120                        29141248 ( 23%)      60588096 (48%)
Inode Information
-----------------
Number of used inodes:          248973
Number of free inodes:         8139635
Number of allocated inodes:    8388608
Maximum number of inodes:      8388608


171776 / 256
671
[root@pfecl1n1]/gpfs/cl1fs1/kai: dd if=/dev/zero of=./wastedspace bs=256K count=671
671+0 records in
671+0 records out
175898624 bytes (176 MB) copied, 2.26565 s, 77.6 MB/s
```

Interesting cases 2 : No space left on device with only 4% occupied?

```
[root@pfecl1n1]/gpfs/cl1fs1/kai: mmdf cl1fs1
disk                disk size  failure holds    holds              free KB             free KB
name                    in KB    group metadata data        in full blocks        in fragments
--------------- ------------- -------- -------- ----- -------------------- -------------------
Disks in storage pool: system (Maximum disk size allowed is 288 GB)
SYS3                 20971520        1 Yes      No          9656064 ( 46%)           472 ( 0%)
SYS4                 20971520        2 Yes      No          9656832 ( 46%)           696 ( 0%)
SYS5                 20971520        3 Yes      No          9656320 ( 46%)           632 ( 0%)
                -------------                         -------------------- -------------------
(pool total)         62914560                           28969216 ( 46%)          1800 ( 0%)

Disks in storage pool: data (Maximum disk size allowed is 288 GB)
DAT3                 20971520        1 No       Yes               0 (  0%)      20196056 (96%)
DAT4                 20971520        2 No       Yes               0 (  0%)      20196152 (96%)
DAT5                 20971520        3 No       Yes               0 (  0%)      20194312 (96%)
                -------------                         -------------------- -------------------
(pool total)         62914560                               0 (  0%)      60586520 (96%)
                =============                         ==================== ====================
(data)               62914560                               0 (  0%)      60586520 (96%)
(metadata)           62914560                           28969216 ( 46%)          1800 ( 0%)
                =============                         ==================== ====================
(total)             125829120                           28969216 ( 23%)      60588320 (48%)

Inode Information
-----------------
Number of used inodes:          248974
Number of free inodes:         8139634
Number of allocated inodes:    8388608
Maximum number of inodes:      8388608
```

```
[root@pfecl1n1]/gpfs/cl1fs1/kai: df /gpfs/cl1fs1/
Filesystem     1K-blocks    Used Available Use% Mounted on
cl1fs1          62914560 2328064  60586496   4% /gpfs/cl1fs1

[root@pfecl1n1]/gpfs/cl1fs1/kai: df -h /gpfs/cl1fs1/
Filesystem      Size  Used Avail Use% Mounted on
cl1fs1           60G  2.3G   58G   4% /gpfs/cl1fs1
```

```
[root@pfecl1n1]/gpfs/cl1fs1/kai: dd if=/dev/zero of=./cantcreate bs=4K count=1
dd: error writing ‘./cantcreate’: No space left on device
1+0 records in
0+0 records out
0 bytes (0 B) copied, 0.276065 s, 0.0 kB/s

[root@pfecl1n1]/gpfs/cl1fs1/kai: ll cantcreate
-rw-r--r-- 1 root root 0 Apr 29 08:37 cantcreate
```

Interesting cases 3: Expels over and over;)

- **One node gets expelled over and over!**
  - most of the time a network setup issue (DNS, Firewall, …)

Scenario: one node joins, tries to mount a FS and gets expelled, GPFS recycles, the node joins again and … gets expelled again.

Interesting cases 4: ESS Power LE node hang due to oom

- **Symptom**
  - System hang. Unable to ssh into the node. No console login possible.
  - Will most likely still respond to ping over network.
  - Free memory is significantly higher than vm.min_free_kbytes.
- **Cause**
  - RHEL on Power LE systems is allocating 5% of the installed memory for kvm_cma by default.
  - This memory is reported as free memory but can't be used by normal applications.
- **Resolving The Problem**
  - kvm_cma allocation during boot can be disabled by adding the kernel option: kvm_cma_resv_ratio=0
  - `vi /etc/default/grub`
  - `GRUB_CMDLINE_LINUX="crashkernel=auto console=hvc0 kvm_cma_resv_ratio=0"`
  - `grub2-mkconfig -o /boot/grub2/grub.cfg`
  - `reboot`
- ESS Power LE node hang due to oom while free memory > vm.min_free_kbytes

Quality Guild

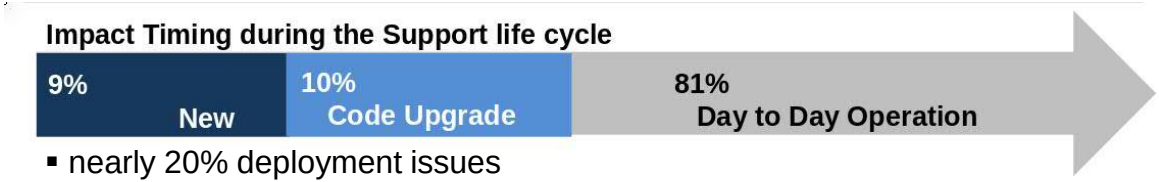
- What is that ? What is our approach with that Quality Guild
- critical field issue analysis and pattern identification
- Findings:
  - ESS as a SONAS Replacement ➔ Protocol support simplification
  - Systematic approach to push customers to latest releases ➔ 4.1.X EOS 30.4. 2019 !!!
  - Focus on Deployment and Upgrade squad continues this year.
  - RAS enhancements
  - Network squad

Quality Guild : Analysis

Top Spectrum Scale Categories of 2018

- Filesystem Core shows highest percentage on counted issues
- CES component is growing waecst
- ESS/GNR thirdmost category
- Naturally these are the most often used product features

Quality Guild : Analysis



- nearly 20% deployment issues

- RAS
  - Protocol resiliency
  - Upgrade challenges
  - Performance tuning, configuration, parameters
  - Monitoring and Serviceability
- Networking Issues
- Product Quality – improved
  - Release Currency - known defects
  - File access related problems
  - Gaps in Error injection testing

Quality Guild : Analysis

- Approx 57% of CIEs are on recent vintage release (V423/v500/v501) but still not at the latest PTFs equivalent to V4238 or better recommended level

Quality Guild : Actions

http://www-01.ibm.com/support/docview.wss?uid=ssg1S1009703

IBM Spectrum Scale Software Version Recommendation Preventive Service Planning

Preventive Service Planning

Abstract

IBM Spectrum Scale Software Version Recommendation

Content

This generalized recommendation is made available to assist clients in implementing a code update strategy. It is a full field perspective, and as such, a customized recommendation which takes into account specifics such as business upgrade windows, length of time since last update, decommission plans, etc. may require assistance from local support teams. In general, recommendations assume planning updates annually.

IBM Spectrum Scale	Minimum Recommended Level	Field proven level	Latest Level
IBM Spectrum Scale	4.1.1.22 (**Alert**: Version 4.1.x is going End Of Support on 4/30/2019. Clients are advised to upgrade to one of the versions in the "field proven" column as soon as possible). [1]	4.x stream 4.2.3.12 [Nov 15, 2018] 5.x stream 5.0.2.2 [Dec 13, 2018]	4.x stream 4.2.3.13 [Jan 2019] 5.x stream 5.0.2.3 [Feb 2019]
IBM Spectrum Scale for ESS	ESS 4.6 [Mar 2017]	4.x stream: ESS 5.2.5 [Dec 2018] 5.x stream: ESS 5.3.2 [Nov 2018]	4.x stream ESS 5.2.5 [Dec 2018] 5.x stream: ESS 5.3.2.1 [Feb 2019]

[1]For information on upgrade, see the Upgrade section in the IBM Spectrum Scale Knowledge Center. The IBM Spectrum Scale supported upgrade

Quality Guild : RAS enhancements

- Continuation of 2018 RAS efforts
  - Network Reconnect GA (Phase 2)
  - Call Home reports and health checker rules
- Ganesha & NFS improvements for locking and scalability of large directories
  - Better locking handling in NFS and improved stress testing
  - Extend test cases to include additional locking scenarios and I/O drivers.
  - Architecture Review Board evaluates design and recommends changes
- Avoid long outages
  - Monitor local disk contention
  - Prevent outages caused by maintenance operations
  - Warn about back-level Spectrum Scale versions
  - Run health monitoring everywhere - deprecate Non-CCR cluster setup
- Collect the right data to better understand our customers and improve the product
  - Call Home system heartbeat (minimum data set) and critical events
  - Crash detection and reporting through Call Home
  - Measure Install/Upgrade Toolkit usage
  - Tool to collect detailed Network data

Quality Guild

- Outcome : Network squad
  - Improvements in 5.0.2 for Avoiding and Debugging Expels
    - Prioritize the commMsgCheckMessages RPC to avoid RPC Time-out (critical RPCs)
    - Network PD improvement - dump the TCP_INFO to mmfs.log when disk lease is overdue

```
[N] Node 192.168.41.31 (node11) lease renewal is overdue. Pinging to check if it is alive
[I] The TCP connection to IP address 192.168.41.31 node11 <c0n1> (socket 63) state: state=1 ca_state=0
snd_cwnd=10 snd_ssthresh=43 unacked=0 backoff=0 retransmits=0 rto=209000 rcv_ssthresh=924512 rtt=8847
rttvar=14163 retrans=0 reordering=3 lost=0
```

    - New Pro-active Reconnect Feature code added in 5.0.2

```
# mmfsadm dump config | grep -i proactive
 proactiveReconnect 0
```

```
# mmchconfig proactiveReconnect=yes
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
 affected nodes.  This is an asynchronous process.
```

  - IBM Spectrum Scale Network Related Flows and Troubleshooting

Enhancements, Improvements, Tipps ...

5.0.0 Enhancements:

- more than 32 subblocks
- mmrestripefs faster (-b / --strict)
- mmchnsd command: Change NSDs without unmounting the file system
- mmtracectl –status added
- mmfsck –status-report added
- Assert consolidation LOGASSERT / DBGASSERT / DYNASSERT / disableAssert feature

Enhancements, Improvements, Tipps ...

5.0.1 Enhancements:

- mmcachectl : pagepool usage:

```
[root@kbnode1 ~]# mmcachectl show --device fs1
FSname       Fileset  Inode    SnapID   FileType     NumOpen    NumDirect  Size         Cached       Cached
             ID                                     Instances  IO         (Total)      (InPagePool) (InFileCache)
-------------------------------------------------------------------------------------------------------------------
fs1          0        2059     0        special      1          0          524288       524288       F
fs1          0        4        0        special      1          0          262144       262144       F
fs1          0        2056     0        special      0          0          262144       262144       F
fs1          0        2057     0        special      1          0          524288       524288       F
fs1          0        2058     0        special      1          0          524288       524288       F
fs1          0        3        0        directory    0          0          262144       0            F

File count: 6
```

- mmap read enhancement → performance improvements
  - “mmapOptimizations” set by default
  - +8% sequential read bandwidth with mmap() over 5.0.0
- metadata enhancement → metadata create/delete performance improvements
  - New configuration parameter fileCacheSoftLimitPct : percentage below mFtC to start background steal thread
  - +10% file metadata performance improvements for deletes, when deleting huge # of files, over 5.0.0
- Samba winbind → queuing and signal handling improved
- Ganesha → now dumps stack backtrace on a crash (if coredump is setup)

Enhancements, Improvements, Tipps ...

5.0.2 Enhancements:

- mmfsck improvements --estimate-only runs with the exact same params , tests disk stat, RPC stat, bytes to scan

```
Estimating fsck run time
Measuring disk stat...
Measuring RPC stat...
Estimating bytes to scan...
Fsck will complete in 2 hours 6 minutes (+/- 7 minutes)
Note that this estimate does not factor in any CPU processing overhead and assumes balanced scan workload across all threads and nodes
```

- mmchfs maintenance-mode option , if “yes” → prevents mounting
- Improve stat cache handling in Linux (performance improvements over 5.0.1)
- panicOnIOHang (ioHangDetectorTimeout -- (default 300 seconds))
- diskIOHang event callback
- autoBuildGPL → if mmstartup notices failing modules, calls mmbuildgpl on its own
- Network improvements :
  - mmnetverify checks connectivity of remote clusters
  - prioritize the commMsgCheckMessages RPC
  - lease renewal pings are taking subnets variable into account
  - proactive reconnect socjet
  - dump TCP_INFO in mmfs.log when disk lease overdue recognized

Enhancements, Improvements, Tipps ...

5.0.2 Enhancements:

- Samba improvements :
  - idmap lookups improved
  - graceful behavior of ctdb OOM : avoids crash, going unhealthy if swap > 95% used, logs memory usage
  - speedup wbinfo –p → reduces likelihood of winbindd monitor timeouts causing failovers

- Auth improvements
  File auth configuration in AD base authentication schemes are validating network config now.
  Checking if DNS server can be contacted to fetch AD domain controllers
  On a working cluster configured with AD authentication scheme, run:

```
# mmuserauth service check --data-access-method file --nodes cesNodes
```

Enhancements, Improvements, Tipps ...

mmfsck tipps :

1. --patch won't work if there is reserved file corruptions, like allocation map. To fix such errors, we need mmfsck –y
But full -y needs to scan FS again and it could be very slow. So when mmfsck with patch-file showed reserved file corruption, you can use workarounds like aborting -y run after reserved file repair and then using patch-file to complete the rest repair

2. If customer can't run offline mmfsck, we could run online fsck in read only mode(-onV) - it will tell us if there are any inode corruptions. Online fsck will not check reserved files, duplicate block corruptions, directories and orphan inodes. Basically it has to read all inodes and indirect blocks to find lost blocks and during that time if it finds any inode or indirect block corruption, it will report it

3. You can use -xsc to skip expensive check for directory cycles, which means cross hard links between directories. It never happened in real cases, so we can skip to speed up offline mmfsck

4. -xc is to suppress replica compare(data blocks) in case log recovery fails. So if there is no log recovery failure, you can skip it too. So -xc is just a way to make sure to not do expensive data replica compare under any condition

Useful Links

- Spectrum Scale FAQ : everything you may want to know about supported releases, Upgrade Paths, etc.
- Release currency, field approved level
- IBM Spectrum Scale Network Related Flows and Troubleshooting
- IBM Spectrum Scale Wiki

Questions?

Do you have any issue with the Spectrum Scale/ESS support ?

Where can we improve ?