IBM Spectrum Scale
Field Update

Spectrum Scale User Group Meeting 19.-20. October ‘22

Achim Rehor & Bianca Schmidt
Spectrum Scale/ESS Support EMEA

Disclaimer

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- 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.

Agenda

- IBM Spectrum Scale /ESS Solution Support
  - ESS/Scale Support in EMEA
  - ESS/Scale Reference Guides
  - New Feature: IBM Technical Support Chat - LiveChat
  - New Feature: Escalation Button
- Aus dem Feld – interessante Probleme
  - LazyNodes – hanging cluster
  - workerThreads – viel hilft viel (nicht immer ;))
  - Mitigation for double node I/O hang crash
  - InodeAlloc – Verteilung freier Inodes
  - mmsdrfs/Filesystem Recovery

Agenda

- **IBM Spectrum Scale /ESS Solution Support**
  - ESS/Scale Support in EMEA
  - ESS/Scale Reference Guides
  - New Feature: IBM Technical Support Chat - LiveChat
  - New Feature: Escalation Button
- Aus dem Feld – interessante Probleme
  - LazyNodes – hanging cluster
  - workerThreads – viel hilft viel (nicht immer ;))
  - Mitigation for double node I/O hang crash
  - InodeAlloc – Verteilung freier Inodes
  - mmsdrfs/Filesystem Recovery

ESS/Scale Support in EMEA

Spectrum Scale

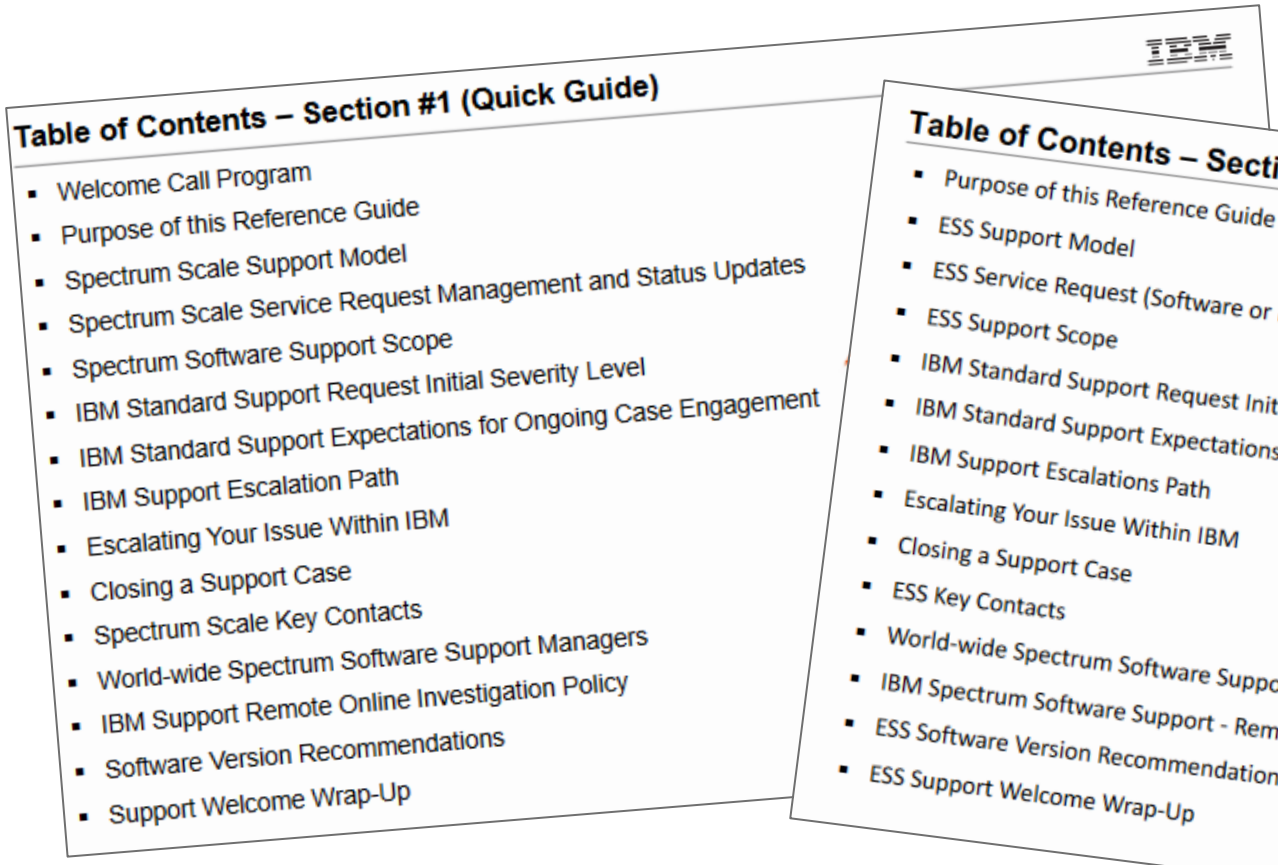
- IBM Spectrum Scale /ESS Solution Support EMEA
  - Das Team besteht aus Kollegen in Deutschland, Italien, Großbritannien, Belgien, Slowenien, Bulgarien, Kroatien
  - Abdeckung der EMEA Tagesschicht von Mo-Fr (Sev 1-4) + Sa-So (Sev 1)
  - Sev 1 Cases werden nach dem “Follow-the-sun” Modell bearbeitet

ESS/Scale Reference Guides

Spectrum Scale

ESS Reference Guide: https://www.ibm.com/support/pages/node/6252477

Spectrum Scale Reference Guide: https://www.ibm.com/support/pages/node/6252403



IBM

Table of Contents – Section #1 (Quick Guide)

- Welcome Call Program
- Purpose of this Reference Guide
- Spectrum Scale Support Model
- Spectrum Scale Service Request Management and Status Updates
- Spectrum Software Support Scope
- IBM Standard Support Request Initial Severity Level
- IBM Standard Support Expectations for Ongoing Case Engagement
- IBM Support Escalation Path
- Escalating Your Issue Within IBM
- Closing a Support Case
- Spectrum Scale Key Contacts
- World-wide Spectrum Software Support Managers
- IBM Support Remote Online Investigation Policy
- Software Version Recommendations
- Support Welcome Wrap-Up

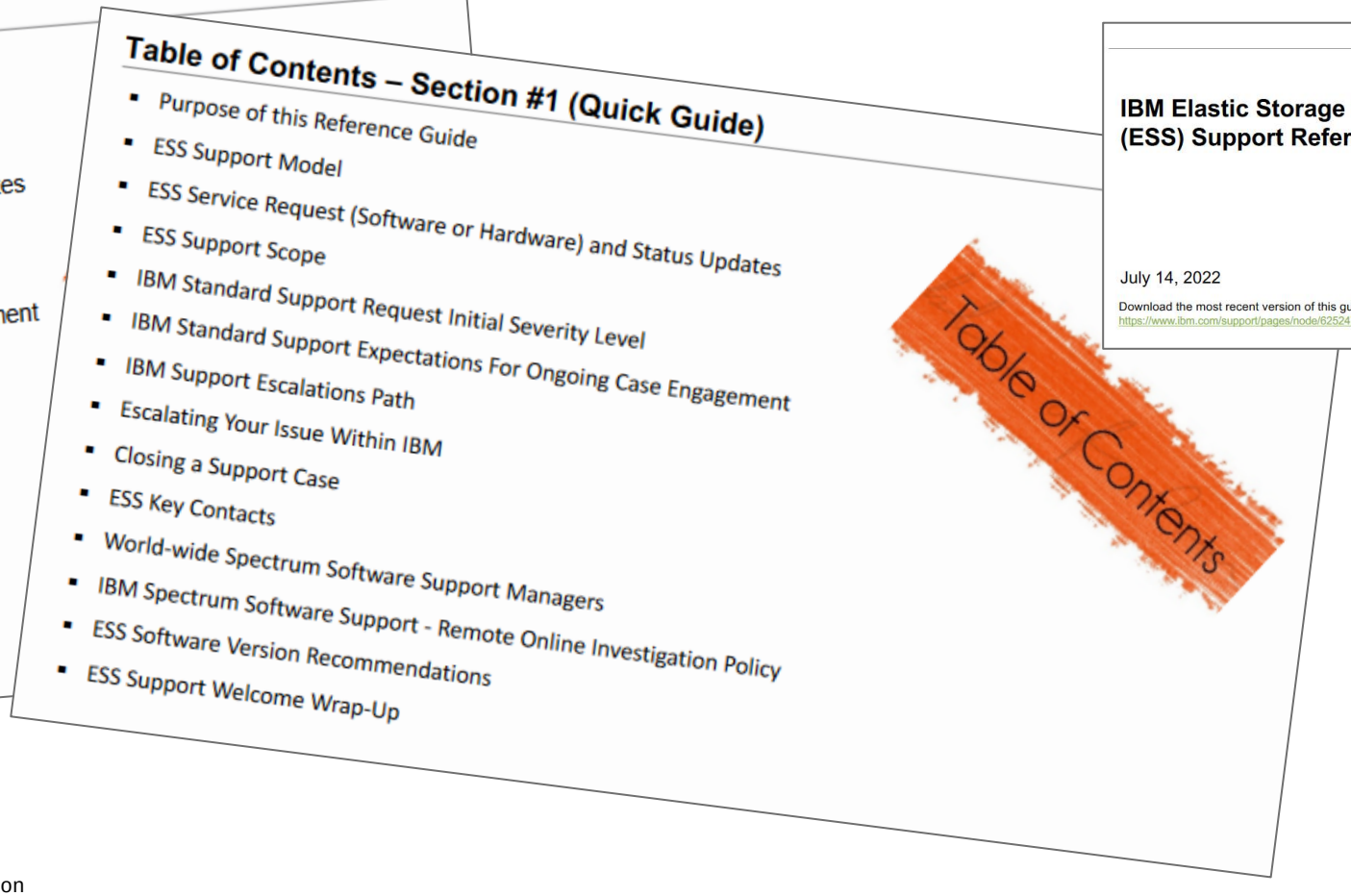
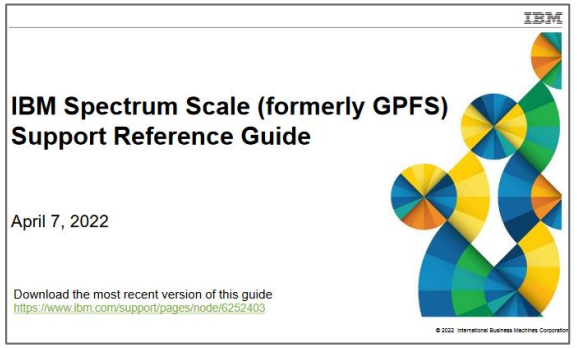


Table of Contents – Section #1 (Quick Guide)

- Purpose of this Reference Guide
- ESS Support Model
- ESS Service Request (Software or Hardware) and Status Updates
- ESS Support Scope
- IBM Standard Support Request Initial Severity Level
- IBM Standard Support Expectations For Ongoing Case Engagement
- IBM Support Escalations Path
- Escalating Your Issue Within IBM
- Closing a Support Case
- ESS Key Contacts
- World-wide Spectrum Software Support Managers
- IBM Spectrum Software Support - Remote Online Investigation Policy
- ESS Software Version Recommendations
- ESS Support Welcome Wrap-Up

Table of Contents

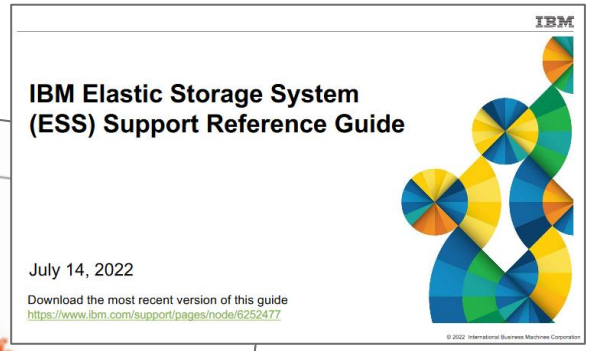


IBM

IBM Spectrum Scale (formerly GPFS) Support Reference Guide

April 7, 2022

Download the most recent version of this guide
https://www.ibm.com/support/pages/node/6252403



IBM

IBM Elastic Storage System (ESS) Support Reference Guide

July 14, 2022

Download the most recent version of this guide
https://www.ibm.com/support/pages/node/6252477

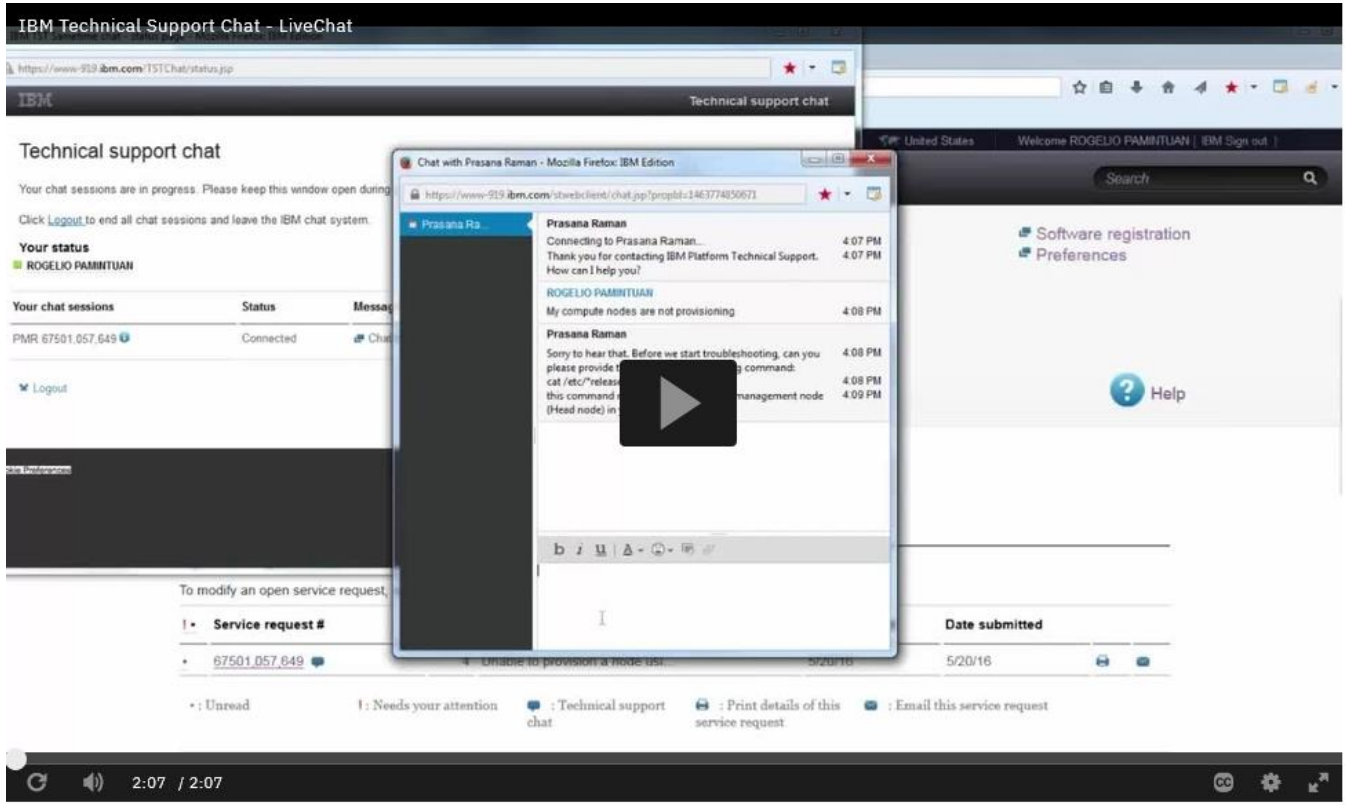
New Feature: IBM Technical Support Chat - LiveChat

Spectrum Scale

- Introduction to IBM Technical Support Chat – LiveChat

  https://www.ibm.com/support/pages/introduction-ibm-technical-support-chat-livechat

  https://mediacenter.ibm.com/media/IBM+Technical+Support+Chat+-+LiveChat/1_35b47mp0



New Feature: Escalation Button

Spectrum Scale

- Escalation Button in Support Case Interface
  - This escalation method requires an open support case.
  - Expect a case owner, team lead or manager to respond to this escalation.
  - Datails: https://www.ibm.com/support/pages/node/6554766

Escalating a Case

1. Log into community, then select **Cases > View Cases** from the header, and then click the case you want to escalate.
2. In the Status area of the case details page, click **Escalate case.** The escalation window displays.

Note: If Escalate Case is not available, see the If You Cannot Escalate a Case section.

IBM Support | Offerings | Cases | Downloads | Products | Forums | Monitoring | FAQ | Search support

CSPDev2 Planned Maintenance: On Sunday, November 1 from 0200 UTC to 0500 UTC, the IBM Support site will be in read- only mode. If you need to monitor or open a case during this time, you can call IBM directly: IBM Directory of wor

Back to all cases | Actions

Security Guardium Insights

Case number

Description
Security Guardium Insights

Status
New Case

Close case | Escalate case

Case history

Add a comment

Post comment

Search case history

3 Nov 2020

3 Nov 2020 10:07 PM
[IBM) created this case

Product
Security Guardium Insights

Product attributes

Severity
2 - Significant impact (any system is down)

Created
3 Nov 2020 (1 days old)
10:07 PM

Upload files

Before providing problem information and/or diagnostic attachments, please read and understand Exchanging diagnostic data with IBM.

Customer

Chat with an Agent

Escalate case

*Preferred method for IBM to contact you
Email | Phone | Other

*Email

*Escalation reason
Increased business impact/deadline

*Description of concern/escalation
More than half of the company is now impacted. Need quick resolution before EOD 11/5.

Cancel | Submit

Agenda

- IBM Spectrum Scale /ESS Solution Support
  - ESS/Scale Support in EMEA
  - ESS/Scale Reference Guides
  - New Feature: IBM Technical Support Chat - LiveChat
  - New Feature: Escalation Button
- **Aus dem Feld – interessante Probleme**
  - LazyNodes – hanging cluster
  - workerThreads – viel hilft viel (nicht immer ;))
  - Mitigation for double node I/O hang crash
  - InodeAlloc – Verteilung freier Inodes
  - mmsdrfs/Filesystem Recovery

LazyNodes – hanging cluster

Spectrum Scale

Problem:

Node leaves verursachen Cluster hangs
- verteilte Resourcen muessen reorganiziert werden
  (Block Allocation, Inode Allocation, ccMsgGroupLeave, tokenManagement)
- Ein einzelner 'LazyNode' kann diese Update Prozesse behindern, und Fehlverhalten erzeugen
  (z.B. auf CES nodes, NFS NULL check, no progress ==> failover)

Mitigation:

Script Implementierung (kundenspezifisch), welche periodisch waiters checked, falls bestimmte RPC waiters 'zu alt' werden, wird der Admin angemailed

Solution:

Integration in mmhealth:

Ab 5.1.5.1(+efix) wird ein event in mmhealth generiert. (rpc_waiters)
Spaeter wird ein expel des Problem-Knoten (optional) moeglich sein.

```
[root@c6n2 ~]# mmhealth cluster show

Component           Total         Failed       Degraded        Healthy          Other
-------------------------------------------------------------------------------------
NODE                    6              0              0              6              0
GPFS                    6              0              0              6              0
NETWORK                 6              0              0              6              0
FILESYSTEM              2              0              0              2              0
DISK                    9              0              0              9              0
CES                     2              0              0              2              0
CESIP                   1              0              0              1              0
FILESYSMGR              1              0              0              1              0
GUI                     2              0              0              2              0
PERFMON                 6              0              0              6              0
THRESHOLD               6              0              0              6              0
```

```
Component           Total         Failed       Degraded        Healthy          Other
-------------------------------------------------------------------------------------
NODE                    6              0              5              1              0
GPFS                    6              0              5              1              0
NETWORK                 6              0              0              6              0
FILESYSTEM              2              0              0              2              0
DISK                    9              0              0              9              0
CES                     2              0              0              2              0
CESIP                   1              0              0              1              0
FILESYSMGR              2              0              0              2              0
GUI                     2              0              0              2              0
PERFMON                 6              0              0              6              0
THRESHOLD               6              0              0              6              0
```

```
[root@c6n2 ~]# mmhealth node show -N all

Node name:      c6n1
Node status:    DEGRADED
Status Change:  Now

Component       Status        Status Change     Reasons & Notices
-----------------------------------------------------------------------
GPFS            DEGRADED      Now               ccr_quorum_nodes_warn
NETWORK         HEALTHY       6 hours ago       -
FILESYSTEM      HEALTHY       6 hours ago       -
DISK            HEALTHY       6 hours ago       -
FILESYSMGR      HEALTHY       5 hours ago       -
GUI             HEALTHY       6 hours ago       -
PERFMON         HEALTHY       6 hours ago       -
THRESHOLD       HEALTHY       6 hours ago       -
```

LazyNodes – hanging cluster

Spectrum Scale

Implementierung:

mmsysmon analysiert regelmaessig `mmdiag --network`
nach pending RPC replies
Wenn diese einen Grenzwert ueberschreiten `(msg_revoke_timeout (default 180))`
wird ein health event generiert `rpc_waiters`

Customizing:

/var/mmfs/mmsysmon/mmsysmonitor.conf : `msg_revoke_timeout = 180`

Component	Total	Failed	Degraded	Healthy	Other
NODE	6	1	5	0	0
GPFS	6	1	5	0	0
NETWORK	6	0	0	6	0
FILESYSTEM	2	0	0	2	0
DISK	9	0	0	9	0
CES	2	0	0	2	0
CESIP	1	0	0	1	0
FILESYSMGR	2	0	0	2	0
GUI	2	0	0	2	0
PERFMON	6	0	0	6	0
THRESHOLD	6	0	0	6	0

```
[root@c6n2 ~]# mmhealth cluster show node

Component      Node      Status        Reasons
-----------------------------------------------------------------------------------------
NODE           c6n3      FAILED        gpfs_unresponsive,ccr_local_server_warn,ccr_comm_dir_warn
NODE           c6n2      DEGRADED      ccr_quorum_nodes_warn
NODE           c6n1      DEGRADED      rpc_waiters,ccr_quorum_nodes_warn
NODE           c6n4      DEGRADED      ccr_quorum_nodes_warn
NODE           c6n5      DEGRADED      ccr_quorum_nodes_warn
NODE           c6n6      DEGRADED      ccr_quorum_nodes_warn
```

```
[root@c6n2 ~]# mmhealth node eventlog -N c6n1 | grep "2022-10-14 15"
2022-10-14 15:39:49.357152 CEST   ccr_quorum_nodes_warn      WARNING   At least one quorum node is non-reachable. Item=PC_QUORUM_NODES,ErrMsg='Ping CCR quorum nodes failed',Failed='10.0.0.99'
2022-10-14 15:41:49.859626 CEST   rpc_waiters                WARNING   Pending RPC messages were found for those nodes: c6n3.
2022-10-14 15:46:09.893850 CEST   ccr_quorum_nodes_ok        INFO      All quorum nodes are reachable PC_QUORUM_NODES.
2022-10-14 15:47:40.051812 CEST   no_rpc_waiters             INFO      No pending RPC messages were found.
```

workerThreads – viel hilft viel (nicht immer ;))

Spectrum Scale

Problem:
User sehen schlechte ReadPerformance, wenn im remotecluster
ein IO intensiver Job auf ein anderes Filesystem laeuft

Analyse:
(NSD Server Seite)
dump nsd – zeigt NSD queuing an …
Pagepool Calculation: nsdBufspace 30 (default)

```
Config data for NsdQueueTraditional (NsdQueueUseNsdIO):
 Input config parms:
  nsdThreadsPerDisk: 3, nsdMinWorkerThreads: 16, nsdMaxWorkerThreads 1024, nsdThreadsPerQueue 3,

                 numServerDisksForType 76, nsdThreadMethod 1
  nsdSmallBufferSize 65536, nsdBigBufferSize: 0, nsdSmallThreadRatio 0, maxNumNsdQueues 256
 Derived config parms:
  threadRatio: 7, threadsPerQueue: 3, numLargeQueues 31, numSmallQueues: 224
  largeBufferSize: 4194304, smallBufferSize: 65536, desiredThreadsForType 228
 Thread Summary
  Large Threads 93, Small Threads 672
  Large Buff Size 390070272, Small Buff Size 44040192, Total Buff Size 434110464  <== ~420MB
```

Queue Auslastung (zum Beispiel):

```
 Queue NSD type NsdQueueTraditional : SMALL, threads started 3, active 0, highest 1, deferred 0, chgSize 0, draining 0, is_chg 0
  requests pending 0, highest pending 1, total processed 1158          <== out of 3 threads 0 are active, and max 1 was active in the past …idling..
…
 Queue NSD type NsdQueueTraditional : LARGE, threads started 3, active 3, highest 3, deferred 0, chgSize 0, draining 0, is_chg 0
  requests pending 25, highest pending 208, total processed 13227       <== currently 25 IOs are pending, highest pending was 208
```

deutliche Ueberlastung der LARGE IOs

Empfehlung:
- workerThreads 48 (default) auf 512 erhoehen ==> ‘autotuning’
- nsdBufspace ausreichend (60GB von 192GB pagepool)
- mmchconfig nsdSmallThreadRatio=1, damit 1:1 large:small threads
- nsdMaxWorkerThreads=2048
- maxNumNsdQueues=512
- nsdThreadsPerQueue=4

= ergibt dann 1024 Small/1024 Large Threads (vorher 672/93)

Testen mit :

```
# mmfsadm test nsd qcalc nsdSmallThreadRatio=1 nsdMaxWorkerThreads=2048 nsdMultiQueue=512 nsdThreadsPerQueue=4
```

workerThreads – viel hilft viel (nicht immer ;))

Spectrum Scale

Problem:
viel besser, aber noch nicht gut.

Analyse: (Client Seite)
Immer noch performance Einbruch, wenn der IO intense Job laeuft
Auch, wenn der auf einem anderen Filesystem arbeitet!

Empfehlung:
Weitere Load Reduzierung (fuer die NSD server), durch Einschraenken der Client IOs

- workerThreads=64 (statt 512, wie auf den Servern)

Fairere IO Verteilung, durch Nichtueberlasten der NSD Server Queues

Alternativ: QoS (bedarf allerdings ein wenig Planung und Konfiguration)

Mitigation for double node I/O hang crash

Spectrum Scale

Problem:

ESS System verliert Zugang zum Filesystem

Analyse:

Beide IO nodes reboot innerhalb kurzester Zeit. Pdisk Analyse zeigt eine auffaellige pdisk, bevor der Node crashed.
syslog zeigt : [X] Forcing kernel panic to clear hung I/O <== panicOnIOHang gesetzt

1. Problem : Disk Hospital erkennt die disk einige Male als 'ok' an und reintegriert sie.
Kurz darauf, erneuter Fehler ==> Disk Hospital
'toggle' – diagnosing / ok

```
ioHangDetectorTimeout (default of 300
seconds)
```

2. Problem : nach RG takeover verwaltet der 2. Node die pdisk des ersten zusaetzlich.
Der IO auf die Daten ist aber beim Hang/Panic nicht erfuellt worden.
Hohe Wahrscheinlichkeit, dass der Partner Knoten dieselbe Disk lesen moechte.
Wenn sie dort dasselbe Verhalten zeigt ==> panicOnIOHang, IO node crash.

Grosses Problem : Kein Node mehr da zum RG uebernehmen, wenn der erste noch nicht wieder voll aktiv ist
(reboot/ GPFS start / RG takeback dauert halt ein bisschen)

Mitigation for double node I/O hang crash

Spectrum Scale

Fix Phase 1:

Heuristic zeigt, kurz vor 'Hang' sieht man etliche 'timeouts on write' und sich wiederholende:
`[E] Pdisk eXsY of RG rg_ESS5K_01 is not reachable; changing state to missing.` messages

Script: Zaehlen dieser Nachrichten ==> wenn mehr als 3 in 30 Minuten auftauchen
Warnung an Admin : "pdisk eXsY nearly dead, please think about –simulate-failing this disk "

Fix Phase 2:

Automatisches 'failen' der pdisk kommt nicht in Frage. Im Zweifelsfall zuviele gleichzeitig

Code Fix:
- new pdisk state 'iohang state' , 2 (pre)configured times for ioHangDetectorTimeouts
- 'throttle pdisk queue depth' <== Wahrscheinlichkeit fuer hanging IOs reduzieren

```
diskState_t::isIOHang
ioHangDetectorTimeoutWarning (300)
ioHangDetectorTimeout (600)

nsdRAIDThrottlePdiskQueueDepth
```

Wenn ein iohang diagnostiziert wird, wird nach `ioHangDetectorTimeoutWarning` Sekunden fuer diese pdisk `nsdRAIDThrottlePdiskQueueDepth` gesetzt, und erst nach `ioHangDetectorTimeout` Sekunden der panicOnIOHang getriggered.
Die pdisk hat dann diese Parameter bereits gesetzt, wenn der RG takeover stattfindet, sodass die Wahrscheinlichkeit den IOHang dort ebenso zu sehen geringer ist.
Plus .... es gibt mehr Zeit fuer den reboot/retakeover des crashed Nodes

Fix is in ESS 6.1.4 (Spectrum Scale 5.1.5.0)

InodeAlloc – Verteilung freier Inodes

Background

The inode allocation map file:

The inode allocation map (IMap) file is a system file that is created during filesystem creation time and contains bits representing the allocation state of every inode in the file system. A new inode allocator on any node in the cluster looks up the IMap file for an unused inode, marks the corresponding bit as BeingCreated, updates the corresponding inode and directory entry structures and marks the bit as InUse. Since inode allocations may occur on any mounted node, to enable parallelism, the IMap file is divided into independently lockable records. Further, the total inodes in a file system can be partitioned into disjoint inode spaces which facilitate snapshotting individual inode spaces. Such inode spaces form the backbone of independent filesets. The IMap file reflects the inode space partition by grouping records into a fixed number of regions and a variable number of segments where each region set represents one inode space. In general, a file system starts with **one segment per inode space**. So in general, **for any inode space, the number of regions determines the inode allocation node parallelism**. It is important that each mounted node is able to get its own IMap region. If there are less regions than mounted nodes, the inode allocation requests will have to go through the network to another node adding significant delay. Hence, during file system creation time, the number of IMap regions is calculated to be larger than the maximum number of nodes that will mount the file system. Specifically, it will be 4 times the NumNodes argument to mmcrfileset. The reason for the 4x multiplier has to do with inode expansion.

Inode expansion:

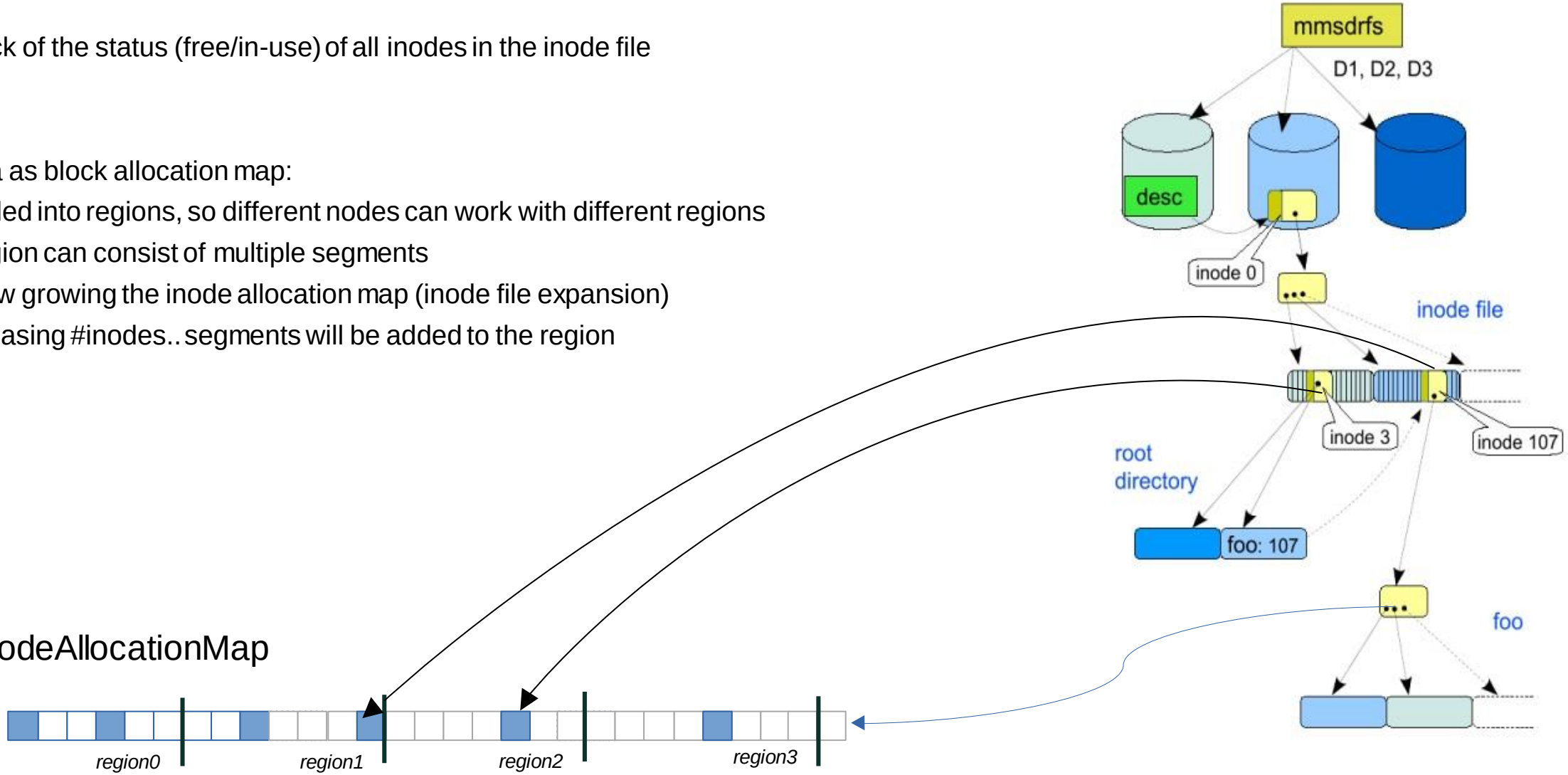
For any inode space (independent fileset), the user may specify the maximum number of inodes allowed in that inode space. However, to save time, only a subset of the maximum number of inodes are created which are called the allocated inodes. When the number of free allocated inodes fall below 25% of the total allocated inodes, then the file system performs an automatic inode expansion upto MIN(25% of allocated inodes, 4M, maximum number of inodes for the inode space). Since the automatic inode expansion is triggered only when 75% of the allocated inodes are in use, GPFS tries to ensure that even if 75% of allocated inodes are in use, each node will have at least one region that still has free inodes. This is assured by setting IMap regions to 4 times the max number of mounted nodes.

Note that inode expansion can also be triggered explicitly by user by increasing the number of pre-allocated inodes for any inode space.

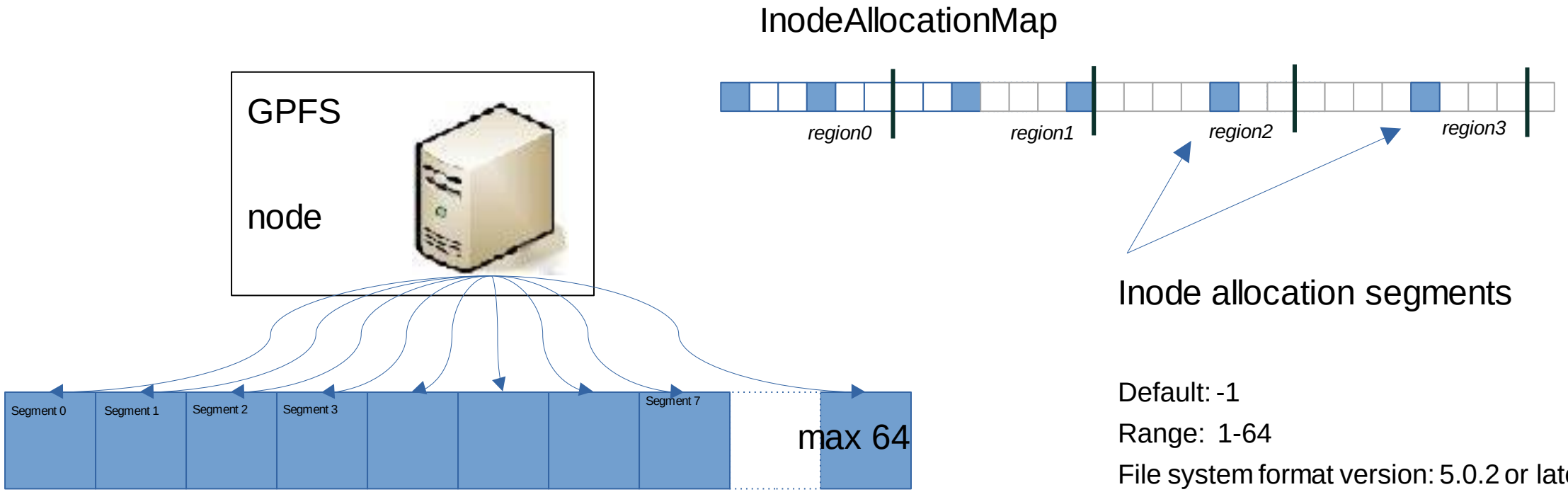
Inode Allocation Map

- Keeps track of the status (free/in-use) of all inodes in the inode file

- Same idea as block allocation map:
  - Divided into regions, so different nodes can work with different regions
  - A region can consist of multiple segments
    to allow growing the inode allocation map (inode file expansion)
  - increasing #inodes.. segments will be added to the region



Active Inode Allocation regions and segments



Default: -1

Range: 1-64

File system format version: 5.0.2 or later

mmchconfig maxActiveIallocSegs=default (which is 8 or 4 , depeding on -n)

Specifies the number of active inode allocation segments that are maintained on the specified nodes. The valid range is 1 - 64. A value greater than 1 can significantly improve performance in the following scenario:

A single node has created a large number of files in multiple directories.

Processes and threads on multiple nodes are now concurrently attempting to delete or unlink files in those directories.

Active Inode Allocation regions and segments

– number of regions depends on -n value from mmcrfs

– ‘-n’ value can be changed for existing file system but it has no effect for the inode allocation map

```
-n NumNodes
        Changes the number of nodes for a file system but does
        not change the existing system metadata structures. This
        setting is just an estimate and can be used only to
        affect the layout of the system metadata for storage
        pools created after the setting is changed.
```

– shortly after creating the file system, you’ll get 8x or 4x the number of ‘-n’ regions

– inodes retrieved then from segments,

– design/plan accordingly, that enough regions are available (‘-n’)

– the more inodes are used/needed, the more segments will be needed (done automatically)

– segments are distributed equally across all regions

InodeAlloc – Verteilung freier Inodes

Spectrum Scale

Parameter:
-n # of nodes mounting the FS <== bestimmt # Regions

iAllocMap am Beispiel:
maxActiveIallocSegs 4 (bei n=512)
n=512 : next_prime(512*4) = 2053
independent filesets=12

Klassen von Segmenten und deren Stati

I = HasInodes
i = ProbablyHasInodes
N = NoInodes
n = ProbablyNoInodes

pro independent Fileset ein 'inodeSpaceInd'

```
# mmfsadm dump ialloc
…
Inode allocation dump for stripe group fsn512:
  total inodes 6393856 alloc 1723392 max 2804736 nRegions 2053
 inodeSpaceMask  0x0000000000001E00 inodeBlockMask   0x00000000000001FF iallocSpaceMask 0x000000000000000F iallocSegmentMask
0x0000000000000000
  free_inodes 1518162 local_free 0 total 1518162 (88% of nInodes)
  nInodeSpaceMaps 13

inodeSpaceInd 0 state 2 holdCount 0 suspendCount 0
  nAllocatedInodes 396288 activeInodesPerSegment 32512 nSegmentsPerIRegion 1
  maxActiveIallocSegs 4 maxDeallocHistorySize 50
  activeIAllocSegsP[0]: regNo 666 segNo 0 deallocPrefetchCursor 0
  activeIAllocSegsP[1]: regNo 182 segNo 0 deallocPrefetchCursor 0
  activeIAllocSegsP[2]: regNo 232 segNo 0 deallocPrefetchCursor 0
  activeIAllocSegsP[3]: regNo 1942 segNo 0 deallocPrefetchCursor 0
  inodeFileExpansionRunning 0 inodeFileExpansionBlocked 0 maxInodeThresholdWarningIssued 0
  nBackgroundExpansions 0 nOnDemandExpansions 0 nFailedExpansions 0
            0 inodes allocated from list of prefetched/deleted inodes
            0 inodes allocated from current ialloc segment
            0 inodes allocated from random 'HasInodes' segment ('I')
            0 inodes allocated from random 'ProbablyHasInodes' segment ('i')
            0 inodes allocated from random 'ProbablyNoInodes' segment ('n')
            0 inodes allocated on re-try of local busy segment
            0 inodes allocated on re-try of remote busy segment
            0 inodes allocated from additional reserved range
            0 inodes allocated specified by caller
            0 inodes on list of prefetched/deleted inodes that were not free
            0 segment in active use by another slot
            0 segment in use on another node
            0 no segment with free inodes found
            0 start-over after new active segment seen during search
            0 inode specified by caller already in use
  segment hint: I=36 i=2017 n=0 N=0         <=== 36 Segmente mit Inodes,2017 Segmente mit wahrscheinlich Inodes, 0 Segemente mit keinen
  segment search: active threads 0 phase 0

      0: IIIIIIIIII IIIIIIIIII IIIIIIIIII IIIIIIiiii iiiiiiiiii
     50: iiiiiiiiii iiiiiiiiii iiiiiiiiii iiiiiiiiii iiiiiiiiii
```

InodeAlloc – Verteilung freier Inodes

Spectrum Scale

```
inodeSpaceInd # mmlsfileset fsn512 -i
Collecting fileset usage information ...
Filesets in file system 'fsn512':
Name          Status    Path                        InodeSpace  MaxInodes  AllocInodes  UsedInodes
root          Linked    /gpfs/fsn512                   0          396288       396288        4006
independn1    Linked    /gpfs/fsn512/indeplinkn1       1          200704       102400           1
independn2    Linked    /gpfs/fsn512/indeplinkn2       2          200704       102400           1
independn3    Linked    /gpfs/fsn512/indeplinkn3       3          200704       102400           1
independn4    Linked    /gpfs/fsn512/indeplinkn4       4          200704       102400           1
independn5    Linked    /gpfs/fsn512/indeplinkn5       5          200704       102400           1
independn6    Linked    /gpfs/fsn512/indeplinkn6       6          200704       102400           1
independn7    Linked    /gpfs/fsn512/indeplinkn7       7          200704       102400           1
independn8    Linked    /gpfs/fsn512/indeplinkn8       8          200704       200704      200701
independn9    Linked    /gpfs/fsn512/indeplinkn9       9          200704       102400           1
…

inodeSpaceInd 8 state 2 holdCount 0 suspendCount 0
  nAllocatedInodes 200704 activeInodesPerSegment 32512 nSegmentsPerIRegion 1
…
  segment hint: I=1 i=0 n=0 N=2052
  segment search: active threads 0 phase 2

      0: NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN
     50: NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN
…
   1400: INNNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN
...
   2050: NNN
```

InodeAlloc – Verteilung freier Inodes

Spectrum Scale

Neuer FS parameter (5140):
man mmcrfs/mmchfs
[--auto-inode-limit | --noauto-inode-limit]

```
[root@kbnode2 ~]# mmlsfileset fsn512 -i | grep independn8
Name                     Status    Path                                        InodeSpace      MaxInodes    AllocInodes     UsedInodes
independn8               Linked    /gpfs/fsn512/indeplinkn8                       8              200704         200704         200704   <== Full


[root@kbnode2 ~]# mmlsfs fsn512 --auto-inode-limit
flag                value                    description
------------------- ------------------------ ------------------------------------
 --auto-inode-limit No                       Increase maximum number of inodes per inode space automatically?


[root@kbnode2 ~]# touch /gpfs/fsn512/indeplinkn8/fileIn8n200704
touch: cannot touch '/gpfs/fsn512/indeplinkn8/fileIn8n200704': No space left on device


[root@kbnode2 ~]# mmchfs fsn512 --auto-inode-limit


[root@kbnode2 ~]# mmlsfs fsn512 --auto-inode-limit
flag                value                    description
------------------- ------------------------ ------------------------------------
 --auto-inode-limit Yes                      Increase maximum number of inodes per inode space automatically?


[root@kbnode2 ~]# touch /gpfs/fsn512/indeplinkn8/fileIn8n200704


[root@kbnode2 ~]# mmlsfileset fsn512 -i | grep independn8
Name                     Status    Path                                        InodeSpace      MaxInodes    AllocInodes     UsedInodes
independn8               Linked    /gpfs/fsn512/indeplinkn8                       8              267264         267264         200705   <== expanded
```

```
mmfs.log:
2022-10-17_12:03:28.022+0200: [N] Expanding fsn512 inode space 8 current 200704 inodes (0 free) by 66560 (auto-inode-limit yes)
kbnode3.mainz.de.ibm.com:  2022-10-17_12:03:28.782+0200: [N] Expanded fsn512 inode space 8 from 200704 to 267264 inodes (on-demand).
vs.
2022-10-16_16:43:37.993+0200: [N] Expanding fsn512 inode space 8 current 168960 inodes (41383 free) by 31744 (auto-inode-limit no)
2022-10-16_16:43:38.394+0200: [N] Expanded fsn512 inode space 8 from 168960 to 200704 inodes (background).
```

InodeAlloc – Verteilung freier Inodes

Spectrum Scale

Vorher:

```
inodeSpaceInd 8 state 2 holdCount 0 suspendCount 0
  nAllocatedInodes 200704 activeInodesPerSegment 32512 nSegmentsPerIRegion 1
…
  segment hint: I=1 i=0 n=0 N=2052
  segment search: active threads 0 phase 2

      0: NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN
     50: NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN NNNNNNNNNN
…
```

N = NoInodes

Die zusaetzlichen ‘free inodes’ werden allen Regions hinzugefuegt

Nachher

```
:
:

…
inodeSpaceInd 8 state 2 holdCount 0 suspendCount 0
  nAllocatedInodes 267264 activeInodesPerSegment 32512 nSegmentsPerIRegion 1
  maxActiveIallocSegs 4 maxDeallocHistorySize 50
…
  segment hint: I=1 i=2051 n=0 N=1
  segment search: active threads 0 phase 2

      0: iiiiiiiiii iiiiiiiiii iiiiiiiiii iiiiiiiiii iiiiiiiiii
     50: iiiiiiiiii iiiiiiiiii iiiiiiiiii iiiiiiiiii iiiiiiiiii
```

i = ProbablyHasInodes

Agenda

- IBM Spectrum Scale /ESS Solution Support
  - ESS/Scale Support in EMEA
  - ESS/Scale Reference Guides
  - New Feature: IBM Technical Support Chat - LiveChat
  - New Feature: Escalation Button
- Aus dem Feld – interessante Probleme
  - LazyNodes – hanging cluster
  - workerThreads – viel hilft viel (nicht immer ;))
  - Mitigation for double node I/O hang crash
  - InodeAlloc – Verteilung freier Inodes
  - **mmsdrfs/Filesystem Recovery**

mmsdrfs/Filesystem Recovery

Spectrum Scale

- Ausgangssituation
  - Loss of the Cluster → alle VMs sind gelöscht worden
  - Der Backendstorage der alle Daten/Metadaten hosted wurde nicht angefasst
  - Kein Backup von den Daten/Metadaten
  - Kein Backup der Clusterkonfiguration/Filesystemkonfiguration

- Kann der Support da etwas tun? Ja - ABER....
  - **Keine Abdeckung über einen Standardsupport Vertrag**
  - Abschluss eines separaten Vertrags auf Best-effort Basis war nötig

mmsdrfs/Filesystem Recovery

Spectrum Scale

- Lösungsansätze
  - Aufsetzen eines neuen Spectrum Scale Clusters auf neuem Level ohne jegliche weitere Konfiguration
  - Sammeln von Daten auf allen ehemaligen NSD Servern:

    # tspreparedisk -s , #ls -il /dev/* , #mmfsadm test readdescraw

  - Aus den gesammelten Daten müssen Informationen für jedes Filesytem und jede Disk/NSD: NSD name, NSD Server, NSD ID, Storage Pool, FS Descriptor, Storage Pool,...extrahiert werden

    Beispielzeilen für vollständige Informationen aus einem /var/mmfs/gen/mmsdrfs file:

```
%%home%%:40_SG_ETCFS:fsl_rivendell:2:   dev             = /dev/fsl_rivendell
%%home%%:40_SG_ETCFS:fsl_rivendell:3:   vfs             = mmfs
%%home%%:40_SG_ETCFS:fsl_rivendell:4:   nodename        = -
%%home%%:40_SG_ETCFS:fsl_rivendell:5:   mount           = mmfs
%%home%%:40_SG_ETCFS:fsl_rivendell:6:   type            = mmfs
%%home%%:40_SG_ETCFS:fsl_rivendell:7:   account         = false
%%home%%:50_SG_MOUNT:fsl_rivendell::rw:mtime:relatime:::::::::::::::::::::
%%home%%:60_SG_DISKS:fsl_rivendell:1:desc_only_disk1:104857600:3:descOnly:0A00006D60421C78:nsd:c8n3::other::generic:user:::::system:c8n3:::::
%%home%%:60_SG_DISKS:fsl_rivendell:2:m_only_disk1:5368709120:1:metadataOnly:0A00006B60421C79:nsd:c8n1,c8n2::other::generic:user:::::system:c8n1,c8n2::::
%%home%%:60_SG_DISKS:fsl_rivendell:3:m_only_disk2:5368709120:2:metadataOnly:0A00006B60421C7B:nsd:c8n1,c8n2::other::generic:user:::::system:c8n1,c8n2::::
%%home%%:60_SG_DISKS:fsl_rivendell:4:d_only_disk1:5368709120:1:dataOnly:0A00006B60421C7D:nsd:c8n1,c8n2::other::generic:user:::::data:c8n1,c8n2:::::
%%home%%:60_SG_DISKS:fsl_rivendell:5:d_only_disk2:5368709120:2:dataOnly:0A00006B60421C7F:nsd:c8n1,c8n2::other::generic:user:::::data:c8n1,c8n2:::::
```

mmsdrfs/Filesystem Recovery

Spectrum Scale

Anlegen eines Blankoeintrags um später ein mmimportfs File zu erzeugen

%%home%%:60_SG_DISKS:????: ????: ????: ????: ????: ????: ????:nsd: ????::other::????:user::::: ????: ????:::::

– In allen Ausgaben von #mmfsadm test readdescraw /dev/* die höchste Version finden und darauf basierend die weiteren Daten sammeln:

FS descriptor in sector 2048 of /dev/dm-44
uid C0A8FDFA:5AC38852, name "gpfs", config version 68373

→ Filesystem name: gpfs

→ %%home%%:60_SG_DISKS:gpfs:????:????:????:????:????:????:nsd:????::other::????:user::::: ????:????:::::

mmsdrfs/Filesystem Recovery

Spectrum Scale

- Storage Pool information:

```
storage pools: max 8
        alloc map type 'scatter'
                0: name 'system' Valid nDisks 21 nInUse 21 id 0 poolFlags 0 thin inode -1 nBlocks 0
                regns 2441 segs 3 size 8192 FBlks 0 MBlks 4096000 subblock size 32768
                [...]
                defaultDataReplicas -1 maxFutureDiskBlocks -1 defBGF 1 defWAD 0
                data 0 nFGs 0 disks: Bitmap size 129 1s 0 : 00000000 00000000 00000000
                meta 1 nFGs 2 disks: Bitmap size 129 1s 20 : 6060007E 60000787 80000000
        Number of metadata disks in pool system: 20
                1: name 'data01' Valid nDisks 108 nInUse 108 id 65537 poolFlags 0 thin inode -1 nBlocks 0
                        regns 3251 segs 6 size 262144 FBlks 566231040 MBlks 0 subblock size 32768
                        [...]
                        defaultDataReplicas -1 maxFutureDiskBlocks -1 defBGF 1 defWAD 0
                        data 1 nFGs 2 disks: Bitmap size 129 1s 108 : 1F9FFF81 9FFFF878 7FFFFFFF FFFFFFFF
                        meta 0 nFGs 0 disks: Bitmap size 129 1s 0 : 00000000 00000000 00000000 00000000
        Number of data disks in pool data01: 108
                2: not in use
                3: not in use
```

→ **Daraus ergeben sich zwei Pools:**

Pool ID 0 -> name: system -> containing 20 metadataonly Disks

Pool ID 65537 -> name: data01 -> containing 108 dataonly Disks

mmsdrfs/Filesystem Recovery

Spectrum Scale

- NSD information:

2: nsd2: uid C0A8FDFA:5AC38843, status InUse, availability OK,
created on node <IP>, Tue Apr 3 15:57:23 2018
type 'nsd', sector size 512, failureConfigVersion 28912
quorum weight {1,1}, failure group: id 1, fg index 1
locality group: id 1, lg index 1
failureGroupStrP: (1), rackId 1, locationId 0, extLgId 0
nSectors 419430400 (0:19000000) (204800 MB), inode0Sector 131072
alloc region: no of bits 2688, seg num 0, offset 456, len 744
suballocator 0x1800469EBDC type 1 nBits 32 subSize 2688 dataOffset 460
storage pool: 0
holds metadata only
[...]

→ **Daraus ergeben sich folgende Infos:**

NSD Number in the FS / mmsdrfs File: 2
Failuregroup: 1
Disk Size: 204800 MB = 214748364800 Byte (to be used)
Storage Pool: 0 = system (to be used)
Designation: metadataOnly

%%home%%:60_SG_DISKS:gpfs:2:nsd2:214748364800:1:metadataOnly:????:nsd:????::other::????:user::::::system:????::::

mmsdrfs/Filesystem Recovery

Spectrum Scale

- NSD Server Namen:
  - Diese Informationen konnten nicht aus Daten rekonstruiert werden, waren aber bekannt:

    nsd1,nsd2 -> serving Failuregroup 1

    nsd3,nsd4 -> serving Failuregroup 2

    %%home%%:60_SG_DISKS:gpfs:2:nsd2:214748364800:1:metadataOnly:????:nsd:nsd1,nsd2::other::????:user::::::system:nsd1,nsd2:::::

- NSD IDs:

  2: nsd2: uid C0A8FDFA:5AC38843, status InUse, availability OK,
  created on node <IP>, Tue Apr 3 15:57:23 2018
  type 'nsd', sector size 512, failureConfigVersion 28912
  quorum weight {1,1}, failure group: id 1, fg index 1
  locality group: id 1, lg index 1
  […]

  - Die FS UID C0A8FDFA:5AC38843 ist NICHT dir NSD ID! Suche mit der FS UID in allen /tmp/$line.<nodde_name>readdescraw.out Files.

    → Genau eine File hat die NSD ID

mmsdrfs/Filesystem Recovery

Spectrum Scale

*NSD descriptor in sector 64 of /dev/mapper/dm**
NSDid: C0A8FDF65ABC9751 format version: 1403 Label:
Paxos sector: -2061383299 number of sectors: 8192 isPdisk: 0
Comment: NSD descriptor for <unknown> Thu Mar 29 09:35:45 2018
*Disk descriptor in sector 96 of /dev/mapper/dm**
*FS disk unique id: C0A8FDFA:5AC38843 created on node *****, Tue Apr 3 15:57:23 2018*
format version: 1 last updated: Tue Apr 3 15:57:20 2018
FS descriptor at sector 2048 for 4082 sectors
id C0A8FDFA:5AC38843 is disk nsd2 in FS gpfs

FS UID (C0A8FDFA:5AC38843) -> NSD ID (C0A8FDF65ABC9751).

Die stanza line vervollständigt sich:

%%home%%:60_SG_DISKS:gpfs:2:nsd2:214748364800:1:metadataOnly:C0A8FDF65ABC9751:nsd:nsd1,nsd2::other::????:user:::::system:nsd1,nsd2:::::

mmsdrfs/Filesystem Recovery

Spectrum Scale

- Disk Type:
  - dmm - Device-Mapper Multipath (DMM), showing up as

%%home%%:60_SG_DISKS:gpfs:2:nsd2:214748364800:1:metadataOnly:C0A8FDF65ABC9751:nsd:nsd1,nsd2::other::dmm:user:::::system:nsd1,nsd2:::::

- Anlegen des stanza files zum Importieren des Filesystem Informationen
  - Aus dem neu angelegten Cluster die erste Zeile aus dem /var/mmfs/gen/mmsdrfs File einfügen:

%%9999%%:00_VERSION_LINE::2600:3:999::lc:nsd1:::/usr/bin/ssh:/usr/bin/scp:<cluster ID>:lc2:1656327475::gpfs_cluster_1:1:1:1:3:A:::central:0.0:

  - Filesystem Information:

%home%%:30_SG_HEADR:gpfs::150:::0:::no::no:::::::::::
%%home%%:40_SG_ETCFS:gpfs:1:%2Fgpfs:
%%home%%:40_SG_ETCFS:gpfs:2: dev = /dev/gpfs
%%home%%:40_SG_ETCFS:gpfs:3: vfs = mmfs
%%home%%:40_SG_ETCFS:gpfs:4: nodename = -
%%home%%:40_SG_ETCFS:gpfs:5: mount = mmfs
%%home%%:40_SG_ETCFS:gpfs:6: type = mmfs
%%home%%:40_SG_ETCFS:gpfs:7: account = false
%%home%%:50_SG_MOUNT:gpfs::rw:mtime:relatime:userquota;groupquota;filesetquota;perfileset:::perfileset:::::::::::::::

Die letzte Zeile 50_SG_MOUNT muss den vorher verwendeten Settings (Quota, filesetdf,….) angepasst werden

mmsdrfs/Filesystem Recovery

Spectrum Scale

- NSD Sektion (für jedes einzelne NSD nach Disknummer sortiert (grün))

%%home%%:60_SG_DISKS:gpfs:2:nsd2:214748364800:1:metadataOnly:C0A8FDF65ABC9751:nsd:nsd1,nsd2::other::dmm:user:::::system:nsd1,nsd2:::::

– Import
  - File auf einem NSD Knoten unter /tmp ablegen und mit #mmimportfs all -i /tmp/<filename> importieren
    – Prüfen:

      #mmlsconfig (see last few lines if the Filesystem exists)

      #mmlsnsd -X

      #mmlsdisk <fs_name> -L

    – Mount testen:

      1) read-only via #mmmount <fs_name> -o ro -N <one_node_name> (Lesen)

      2) read/write via #mmmount <fs_name> -N <one_node_name> (Schreiben)

      3) #mmmount <fs_name> -a (to make it available to any Spectrum Scale Node)

    Remark: Ist etwas schief gegangen, kann die Config gelöscht werden #mmexportfs <fs_name> -o /tmp/export.file (und dann löschen von /tmp/export.file).

mmsdrfs/Filesystem Recovery

Spectrum Scale

- Fallstrick descOnly Disk
  - disks: version 370, no of items 129, in use 129

    → Aber nur Informationen über 128 Disks (Data + Metadata)

  - Eine Disk übrig aus Failure Group 3, eine file descriptor Disk

1: nsd1: uid C0A8FDFA:5AC38841, status InUse, availability OK,
created on node <IP>, Tue Apr 3 15:57:21 2018
type 'nsd', sector size 512, failureConfigVersion 28859
quorum weight {1,1}, failure group: id 3, fg index 0
...
storage pool: 0
holds nothing
...

» Da dies eine lokale Disk an einer VM war, die verloren war, wurde das Filesystem mit einer orphaned disk importiert und später eine neue descOnly Disk in FG 3 angelegt

mmsdrfs/Filesystem Recovery

Spectrum Scale

– Lessons Learned → Backup!!!

1) mmexportfs/mmimportfs command to backup Filesystem relevant configuration.
https://www.ibm.com/docs/en/spectrum-scale/5.1.2?topic=reference-mmexportfscommand
https://www.ibm.com/docs/en/spectrum-scale/5.1.2?topic=reference-mmimportfscommand#Mmimportfs

2) Backup CCR relevant data.
https://www.ibm.com/docs/en/spectrum-scale/5.1.2?topic=ccr-recovering-from-existingbackup
This can be automated.
https://www.ibm.com/docs/en/spectrum-scale/5.0.5?topic=issues-automatic-backup-gpfscluster-data

3) For Spectrum Scale CES/Protocols refer here:
https://www.ibm.com/docs/en/spectrum-scale/5.1.1?topic=recovery-backing-up-restoring-protocols-ces-configuration-information

4) mmbackup and Spectrum Protect can be used to backup Filesystem content (user data).
https://www.ibm.com/docs/en/spectrum-scale/5.1.2?topic=reference-mmbackupcommand

Thank you for using
IBM Spectrum Scale/ESS!