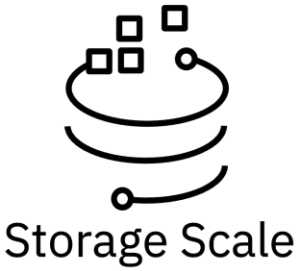
Lessons from the Field

IBM Storage Scale Days 2024

March 5-7, 2024 | Stuttgart Marriott Hotel Sindelfingen

Achim Rehor

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.

IBM Storage Scale Days 2024

Agenda

- TCP Connection Resets
- Expel refresher
- (No route to host)
- ESS3500 and Protocols VMs
- GNR timeout handling on RG recovery
- Survey: What can we do to improve your support experience?

TCP Connection Resets

- MCOT (MultiConnectionOverTcp)
  since 5.1.1
  maxTcpConnsPerNodeConn = 2 (default)
  may help to saturate network links better (small clusters)
  doubles (quadruples, …) the number of sockets (socketMaxListenConnections, net.core.somaxconn)

- MROT (MultiRailOverTcp)
  since 5.1.5
  MCOT still in place
  more than one interface (more than one gpfs subnet) connect to other nodes (subnets='192.168.1.0 192.168.2.0')
  resiliency over adapter/port failures
  tscConnMode (0=neither MCOT nor MROT, 1=MCOT, 2=MROT)
  IpPair Table for deciding on 'next good interface', if one fails
  (detectIpPairAggressiveness 2 …)
  Very often only one single interface exists, so MROT deals with just one IpPair!

```
Inter-node communication configuration:
  tscConnMode      mrot
  tscTcpPort       1191
  my address       9.155.5.31/24 (ens192) <c0n0>
...
Connection details:
  <c0n1> 9.155.5.35/0 (swsaix35)
...
    IpPair Table (offset 0 [2626/0/1]):
      idx iface           status ping_cnt source          destination     subnet
        0 ens192              up        0 9.155.5.31      9.155.5.35      9.155.5.0/24

    <c0n1>:[0] (9.155.5.31:1191 <-> 9.155.5.35:37075)
...
    <c0n1>:[1] (9.155.5.31:46381 <-> 9.155.5.35:1191)
```

- But what are now Connection Resets ?
  - network error (err 73 ~= E_CONNRESET = 73) (Connection reset by peer)
  - partner node mmshutdown, expel, lost quorum
    [E] Close connection to 10.200.42.121 fscc-sr655v3-1 <c0n0>:[1] (Connection reset by peer). Attempting reconnect.
    [E] Close connection to 10.200.42.121 fscc-sr655v3-1 <c0n0>:[0] (Connection reset by peer). Attempting reconnect.

TCP Connection Resets

- packet loss

  look for "[W] The TCP connection to IP address ..." warnings, that tell about retransmits or lost=[1,2]
  if these losses happen on specific packets (reconnect for example), they can lead to expels or Connection resets

- ARP tuning

  in large clusters (>100 nodes) ARP traffic can cause a substantial amount of network traffic
  some sysctl tunings are recommended to minimize this effect.
  - net.ipv4.conf.all.arp_filter=1 - will define which interface responds to an ARP request
  - net.ipv4.conf.all.arp_ignore=1 - reply only if the target IP address is local address configured on the incoming interface
  - net.ipv4.neigh.default.gc_thresh1= - default is 128, minimum number of ARP cache entries, no garbage collector purges, if less than these entries exist, choose s/t like 1024
  - net.ipv4.neigh.default.gc_thresh2= - default is 512, garbage collector kicks in and removes entries older than 5 sec, when this value is reached, choose s/t like 4096
  - net.ipv4.neigh.default.gc_thresh3= - default is 1024, hard maximum entries to keep before GC runs immediately, choose s/t like 8192

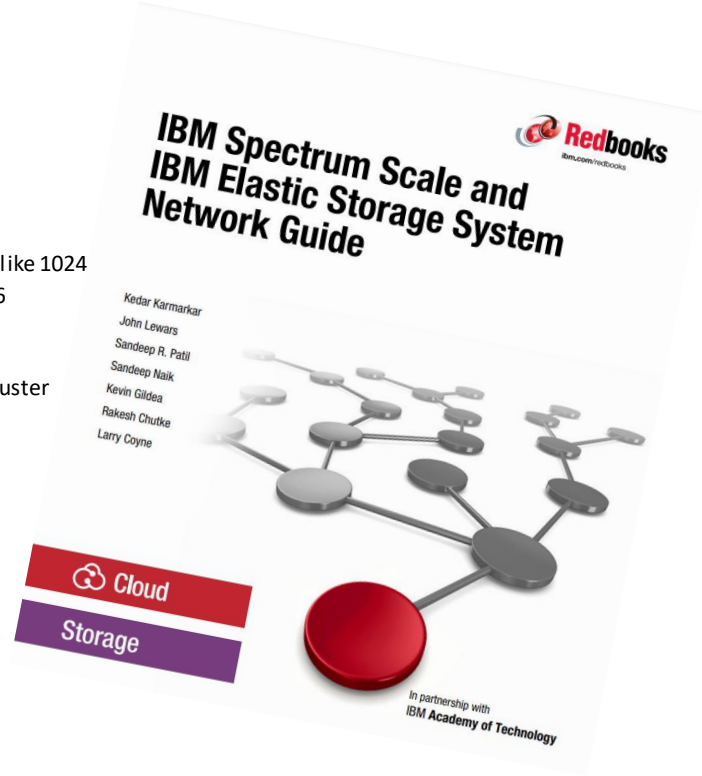
  these gc_thresh values should be based on how many ARP cache entries you do expect
  - net.ipv4.neigh.default.gc_stale_time= - default is 60, effects how often the ARP cache is checked for stale entries, choose a huge value, if you don't use IPAT in this cluster
  - net.ipv4.neigh.default.gc_interval= - default is 30, interval between GC runs, choose a huge value, if you don't use IPAT in this cluster

  https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt for reference

- Buffer tuning

  - net.core.netdev_max_backlog = 250000, net.core.netdev_budget = 600 (default is 300)
  - HW Ring buffers : (ethtool -g ib0)

```
Ring parameters for ib0:
Pre-set maximums:
RX:             8192
RX Mini:        n/a
RX Jumbo:       n/a
TX:             8192
Current hardware settings:
RX:             512
RX Mini:        n/a
RX Jumbo:       n/a
TX:             1024
```



for details and much more, see:
https://www.redbooks.ibm.com/redpapers/pdfs/redp5484.pdf

Expel refresher

What is an expel, and why do we need that ?

- SpectrumScale relies on every node being responsive and a low latency network in order to achieve the wanted high bandwidth

- Nodes not being responsive can block transactions, causing filesystem access to freeze

- Expels are the GPFS way to remove (temporarily) such nodes from the cluster, freeing the blocked resources

- an expelled node leaves the cluster (thus loses all filesystem access) and rejoins immediately

Expel refresher

Types of expels

- lease renewal expels

The clustermgr keeps track of all nodes 'leaseRenewal' times
if a node misses to renew its disk lease in time, the clstrmgr starts to check its availabilty and then decides wheter to expel it

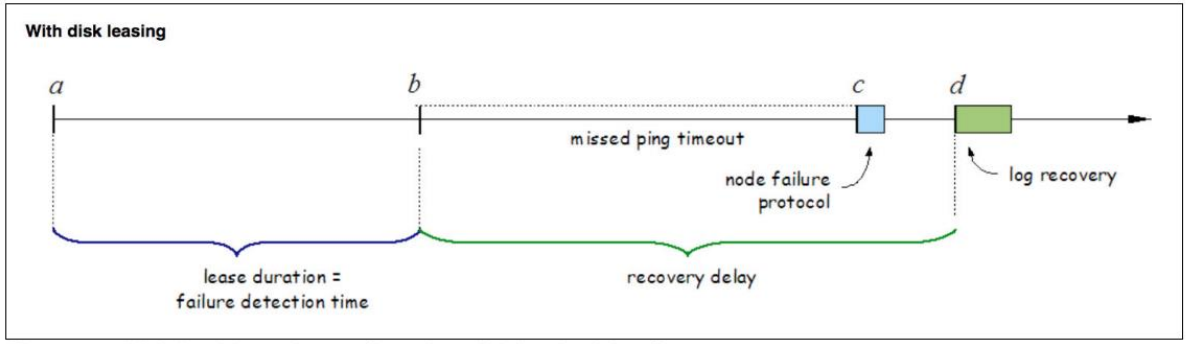


Figure 4-2 High-level lease timeout flow when disk leasing is in effect

well known picture, to better understand the tunables and the flow for lease expiration expels.
Tunables on this are:
failureDetectionTime --> will modify leaseDuration accordingly
minMissedPingTimeout, maxMissedPingTimeout, and leaseRecoveryWait

Quote: "IBM Spectrum Scale expels
In the context of this publication, it is important to understand expels in a IBM Spectrum Scale cluster. As the name indicates, expels in an IBM Spectrum Scale cluster is a scenario in which a specific node that is a part of the cluster is expelled from the cluster. Many scenarios exist where expel of node or nodes occurs that might directly affect the workload; therefore, it is important to understand the reason for the expel. *Field experience shows that many expel scenarios relate to the underlying network.* "

```
root:~# mmfsadm dump cfgmgr | grep -A 2 "lease config:"
lease config: dynamic yes failureDetectionTime 35.0 usePR no recoveryWait 35 dmsTimeout 23
 leaseDuration 35.0/23.3 renewalInterval 30.0/11.7 renewalTimeout 5.0 fuzz 3.00/1.17
 missedPingTimeout 15x2.0=30.0 totalPingTimeout 60x2.0=120.0
```

```
2024-02-13_11:41:12.376+0000: [I] Recovery: data, delay 40 sec. for safe recovery.
```

for details, again see : https://www.redbooks.ibm.com/redpapers/pdfs/redp5484.pdf

Expel refresher

Types of expels (continued)

- RPC timeouts to other nodes (or more general 'communication failures')

  if a node communicates to another node (RPC), for example for a token request, and does not receive a reply in time,
  it will ask the clustermgr to expel that other node (because it does not adhere to SpectrumScale's communication protocol)
  The Clustermgr then has to decide, which of the 2 (requestor or target) shall be expelled. net.core.netdev_budget = 600

  Rules to decide on which one to expel :
  1. Quorum nodes are favored over non-quorum nodes
  2. Local (home file system) cluster nodes are preferred over nodes accessing a remote file system
  3. Nodes having a management role are favored over non-manager nodes
  4. NSD servers are favored over non-NSD server nodes
  5. Nodes that were active longer in the cluster are preferred over nodes that were part of the cluster for shorter periods of time.

  2 upcoming additions:
  a) in cases where a higher priority node has an issue where clients ask the cluster mgr to expel it, an 'expel-storm' can occur, where clients are expelled over and over, while an expel of the 'higher prio' node would have resolved the situation.
  From **5.1.9.0 on the clustermgr keeps track of an expel history**, and decides upon that.
  Internal Story 302079: Network resiliency enhancement - CM should examine history of recent requests for expel candidates in the case of an RPC time out related expel

  ```
  expelHistoryTimeout 60
  expelHistoryWaitInterval 5
  ```

  b) If NSDs are served by GNR nodes (ESS or ECE), then avoid to expel both ESS IO nodes at the same time (or in ECE case, avoid to expel >FT number of vdisk server nodes)
  Internal Story 312832 : Network resiliency enhancement - CM should try to avoid expelling the last remaining server serving an ESS building block on an RPC time out related expel
  Planned for the 5.2.0 timeframe

Expel refresher

Types of expels (continued)

- Planned expels

  An admin can decide to keep a node from joining the cluster with the
  # mmexpelnode command.
  That node will be expelled and marked as expelled, so it cannot rejoin, unless the admin resets that mmexpelnode flag

```
Usage: mmexpelnode [-o | --once] [-f | --is-fenced] [-w | --wait] -N Node[,Node...]
            mmexpelnode {-l | --list}
            mmexpelnode {-r | --reset} -N {all | Node[,Node...]}
```

(No route to host) <--- possibly a Firewall Issue ?

- Symptom : Client nodes join the cluster without any issue,
  once load is generated, i.e. IO on the FS, they are getting expelled
- Finding : these nodes were recently added to the cluster.
  As the existing Nodes (Quorum and NSD servers) do have a valid firewall config
  (either firewalld disabled, or port 1191 open) the new client nodes 'join' went just fine.

```
# nmap clientN1 -p 1191

Starting Nmap 6.40 ( http://nmap.org ) at 2024-02-17 17:03 CET
Nmap scan report for ClientN1 (192.168.45.32)
Host is up (0.00017s latency).
rDNS record for 192.168.45.32: ClientN1.gpfs.net
PORT     STATE    SERVICE
1191/tcp filtered gpfs     vs. 1191/tcp open  gpfs

Nmap done: 1 IP address (1 host up) scanned in 0.66 seconds
```

- As soon as any connect is required (token traffic, other RPCs …) the Server initiates the connection, cannot talk to the client, and hence asks for an expel of the 'unreachable' node

IBM Storage Scale Days 2024

ESS3500 and Protocols VMs

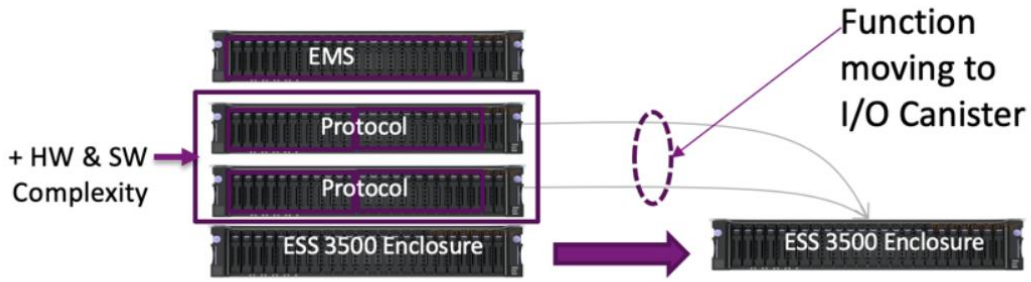
Table 1. Overview of IBM Storage Scale System 3500

Product	Number of DIMMs per server canister	Total memory per server canister	Total storage (raw capacity) per IBM Storage Scale System 3500	Server Canister features
IBM Storage Scale System 3500	8 (64 GB DIMM)	512 GB	Up to 360 TB per IBM Storage Scale System 3500 unit	Single socket AMD Intel 7642 48-core processor.
	8 (128 GB DIMM)	1024 GB	Up to 720 TB per IBM Storage Scale System 3500 unit	Dual 960 GB NVMe boot drives.

ESS3500 and Protocols VMs

From ESS 6.1.3.1, you can run a virtual machine (VM) on an ESS 3500 I/O node canister to support protocol services such as NSF and SMB, which are enabled for an ESS cluster.

Figure 1. Protocol VM deployment on ESS 3500 I/O nodes



ESS 3500

- Optimized for entry configuration
- Hardware simplification and cost reduction
- Eliminate dedicated protocol node
- Only NFS and SMB (no Object)

- Management via bridged network from host management network. No BMC network.
- Fixed VM Configuration
  - VM Create/Destroy via script or Ansible

- One VM per canister
- 8 cores (out of 48) {1 socket, 8 threads}
- 64 GB mem (out of 512)
- Max 8 building blocks
- Number of client connections
  - NSF – 1000
  - SMB – 512
- 195710 MiB Boot drive mapped to dedicated mirrored partition on the host
- 1 or 2 Network Adapters via PCIe-Passthrough – adapters are not shared with host (own IOMMU_group)

ESS3500 and Protocols VMs

```
# mmlscluster
GPFS cluster information
========================
  GPFS cluster name:         test.gpfs.net
  GPFS cluster id:           7959738745332852934
  GPFS UID domain:           test.gpfs.net
  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   Myess1CanS1-hs.gpfs.net  10.0.14.71   Myess1CanS1-hs.gpfs.net   quorum-manager-perfmon
   2   Myess2CanS1-hs.gpfs.net  10.0.14.72   Myess2CanS1-hs.gpfs.net   quorum-manager-perfmon
   3   MyemsS1-hs.gpfs.net      10.0.14.70   MyemsS1-hs.gpfs.net       perfmon
   4   MyPnCanS11-hs.gpfs.net   10.0.14.73   MyPnCanS11-hs.gpfs.net    perfmon
   5   MyPnCanS12-hs.gpfs.net   10.0.14.74   MyPnCanS12-hs.gpfs.net    perfmon
   6   Myess1CanS2-hs.gpfs.net  10.0.15.71   Myess1CanS2-hs.gpfs.net   quorum-manager-perfmon
   7   Myess2CanS2-hs.gpfs.net  10.0.15.72   Myess2CanS2-hs.gpfs.net   quorum-manager-perfmon
   8   MyemsS2-hs.gpfs.net      10.0.15.70   MyemsS2-hs.gpfs.net       perfmon
   9   MyPnCanS21-hs.gpfs.net   10.0.15.73   MyPnCanS21-hs.gpfs.net    perfmon
  10   MyPnCanS22-hs.gpfs.net   10.0.15.74   MyPnCanS22-hs.gpfs.net    perfmon
  11   MyQuS3.gpfs.net          10.0.13.1    MyQuS3.gpfs.net           quorum-perfmon
```

```
# mmlscluster --ces
GPFS cluster information
========================
  GPFS cluster name:         test.gpfs.net
  GPFS cluster id:           7959738745332852934

Cluster Export Services global parameters
-----------------------------------------
  Shared root directory:                /gpfs/cesSharedRoot
  Enabled Services:                     NFS
  Log level:                            0
  Address distribution policy:          even-coverage

Node   Daemon node name       IP address    CES IP address list
-----------------------------------------------------------------
   4   MyPnCanS11-hs.gpfs.net 10.0.14.73    10.0.15.207
   5   MyPnCanS12-hs.gpfs.net 10.0.14.74    10.0.15.205
   9   MyPnCanS21-hs.gpfs.net 10.0.15.73    10.0.15.204
  10   MyPnCanS22-hs.gpfs.net 10.0.15.74    10.0.15.206
```

```
$ grep essvm *mas*/etc/hosts
192.168.45.14  MyPnCanS11.gpfs.net.g  MyPnCanS11 Myess1CanS1 essvm
192.168.45.15  MyPnCanS12.gpfs.net.g  MyPnCanS12 Myess2CanS1 essvm
192.168.45.24  MyPnCanS21.gpfs.net.g  MyPnCanS21 Myess1CanS2 essvm
192.168.45.25  MyPnCanS22.gpfs.net.g  MyPnCanS22 Myess2CanS2 essvm
```

ESS3500 and Protocols VMs

- **Note:** Only one protocol VM can be run on a single I/O node canister. Therefore, for one building block of ESS 3500, you can run two VMs on this ESS building block.
- Limited to NFS and SMB protocol (no Object, due to performance limitation, number of cores, memory)
- Our standard recommendations for protocol nodes don't really fit to these VMs (maxFilesToCache 1000000-4000000, 128 GB for SMB/NFS protocol nodes)
- The documentation states : The protocol VM implementation does not support a highly scalable NFS and SMB workload. However, this implementation supports small NFS and SMB workloads
- **Tip:** If you want to run a massive workload by using an NFS protocol and an SMB protocol, you can use the Power 64 LE version of the protocol node implementation instead the VM implementation.

See here: Protocol virtual machine deployment on ESS 3500 I/O nodes

Note: you are free to choose any other powerful HW to establish your own physical Protocol Node (you just lose the ESS protocol deployment playbook facility ;))

GNR timeout handling on RG recovery

GNR's Fault Tolerance

- Node failure: other node(s) takeover the RG or the LogGroups
- Disk failures: Reed-Solomon codes (8(4)+2P, 8(4)+3P, 3-way-, 4-way-replication)
- IOM, Drawer or even Enclosure Failures: depending on number of failed (inaccessible) disks and number of enclosures, vdisk FT can tolerate that.
- BuildingBlock failures: Only Scale replication can tolerate this (in principle)

GNR timeout handling on RG recovery **Node Fault tolerance**

Node Failure

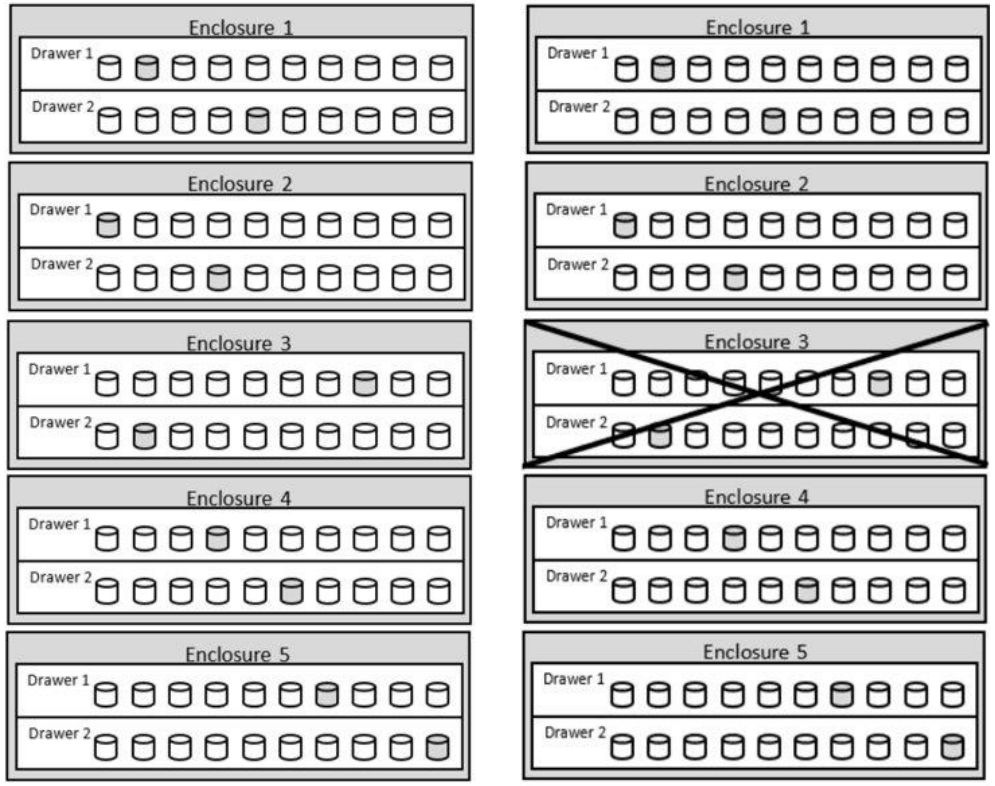
rg_essio1

rg_essio2

rg_essio2/rg_essio1

GNR timeout handling on RG recovery

pdisk group failures



Fault tolerance

```
GNR server disk topology: ESS GL2S (match: 100/100)
GNR configuration: 2 enclosures, 2 SSDs, 0 empty slots,
168 disks total, 6 NVRAM partitions
```

```
# mmvdisk rg list --rg rg_gssio1-hs --ft

configuration data  declustered    VCD spares
                    array          configured  actual  remarks
------------------  -----------    ----------  ------  -------
relocation space    DA1                    44      67  must contain VCD

configuration data  disk group fault tolerance                  remarks
------------------  ------------------------------------------  -------
rg descriptor       1 drawer + 1 pdisk                          limiting fault tolerance
system index        1 drawer + 1 pdisk                          limited by rg descriptor

vdisk               RAID code        disk group fault tolerance                  remarks
------------------  ---------------  ------------------------------------------  -------
rg_gssio1_hs_LOGHOME  4WayReplication  1 drawer + 1 pdisk                        limited by rg descriptor
rg_gssio1_hs_LOGTIP   2WayReplication  1 drawer                                   limited by rg descriptor
rg_gssio1_hs_LOGTIPBACKUP  Unreplicated  0 pdisk                                 limited by rg descriptor
RG001VS001          4WayReplication  1 drawer + 1 pdisk
RG002VS012U1        8+2p             2 pdisk
RG002VS013          8+3p             1 drawer
RG002VS014          8+3p             1 drawer
...
```

8+2p = 10 strips
FT will survive
a full enclosure failure

GNR timeout handling on RG recovery **Spectrum Scale Replication**

```
 -m                2                    Default number of metadata replicas
 -M                2                    Maximum number of metadata replicas
 -r                2                    Default number of data replicas
 -R                2                    Maximum number of data replicas
```

Filesystem defined with 2 replicas,
spread over 2 Failuregroups,
Preferrably over 2 sites (or even better: 3 sites with a desc quorum disk on site 3)

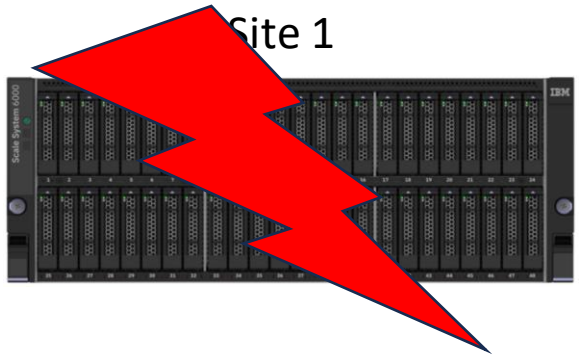
Vdisks from BB on Site 1

Vdisks from BB on Site 2

```
############################################################################
Tue Jan 16 11:36:36 EST 2024: Output for /usr/lpp/mmfs/bin/mmlsdisk MyFs -L on MyNode1 - timeout=60
############################################################################
disk         driver   sector     failure holds    holds                                     storage
name         type       size       group metadata data  status        availability disk id pool         remarks
------------ -------- ------ ----------- -------- ----- ------------- ------------ ------- ------------ ---------
RG003VS004   nsd         512           1 yes      yes   ready         up                 1 system       desc
RG004VS004   nsd         512           1 yes      yes   ready         up                 2 system
RG005VS004   nsd         512           2 yes      yes   ready         up                 3 system       desc
RG006VS004   nsd         512           2 yes      yes   ready         up                 4 system
desc_MyFsQB  nsd         512           3 no       no    ready         up                 5 system       desc
Number of quorum disks: 3
Read quorum value:      2
Write quorum value:     2
```

GNR timeout handling on RG recovery **Spectrum Scale Replication**

Site 1



Site 2

Site 3

GNR timeout handling on RG recovery

Default behaviour: Try to bring a RG up, no matter what … and how long it takes
Clients will retry as long, as the RG is not 'failed'. Once it is 'failed', nsd client will stop the retry and return E_IO to upper layer, allowing to go for a replica, if there is one.

In case of vdisk servers (stateful servers) we try **nsdClientMaxRetries** often to get an IO RPC done.
- nsdClientMaxRetries defaults to 1000 (retries start with 0.5 sec, but grow to 5 sec fast)
  ==> ~5000 sec (83 min) before failing.

In a non-replicated setup, this is perfectly fine, since … there is no other server for the data served by that RG anyway
If replicated, it might make sense to set this lower … (nsdClientMaxRetries = 20 ?)

```
[root@essio1 ~]#
[root@essio1 ~]# mmlsconfig nsdClientMaxRetries
nsdClientMaxRetries 1000
[root@essio1 ~]# mmchconfig nsdClientMaxRetries=20 -i
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
[root@essio1 ~]# mmlsconfig nsdClientMaxRetries
nsdClientMaxRetries 20
[root@essio1 ~]# mmfsadm dump config | grep nsdClientMaxRetries
 * nsdClientMaxRetries 20
[root@essio1 ~]#
```

IBM Storage Scale Days 2024

Survey: What can we do to improve your support experience?

- Feel free to email me:
  Mail: Achim.Rehor@de.ibm.com

- or hit the escalate button on a ticket, where you think we did not meet your expectations, or our goals to ==>

Your candid feedback is extremely valuable as we strive to deliver the best technical support possible and exceed your expectations.

Thank you for using IBM

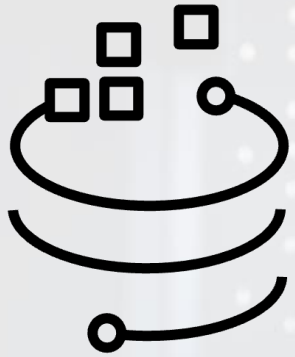
Name of Speaker

Mail: Achim.Rehor@de.ibm.com

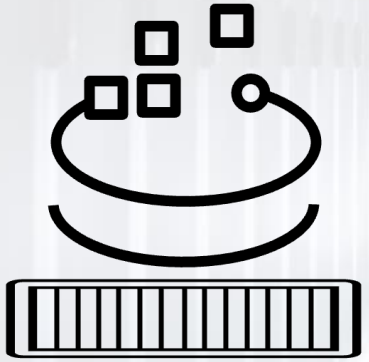
Mobil: (49) 170 452 1194

LinkedIn: https://www.linkedin.com/in/achim-rehor-5908a280/

Thank you for using



Storage Scale



Storage Scale System