

Spectrum Scale Erasure Code Edition

New Storage Options for Spectrum Scale

March, 2020

Olaf Weiser - olaf.weiser@de.ibm.com

Special thanks for contributing material to:

Bill Owen, Senior Technical Staff Member, Stephen Edel
Spectrum Scale Development

Lin Feng Shen, Senior Engineer
Spectrum Scale Development

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.



Intro ...

GPFS native Raid / GNR

**What is Spectrum Scale Erasure Code Edition?
Architectural Comparison with ESS**

ECE Hardware Requirements

Real example

Network is important

Technology Trends

- high speed networks
 - 25GbE, 40 GbE, 50 GbE, 100 GbE, 200 GbE ..
 - infiniband FDR (56) , EDR (100) , HDR (200)
- new „disk“ technology brings > 3 GB/s bandwidth and 500.000 IOPS per drive
- new interfaces to the block layer ...

...challenges...

- scale over fabrics
- disk: MTBF, TBW*, DWPD*
- silent data corruption

*Terra Bytes Written

** Dive Writes per Day

Software defined RAID in SpectrumScale



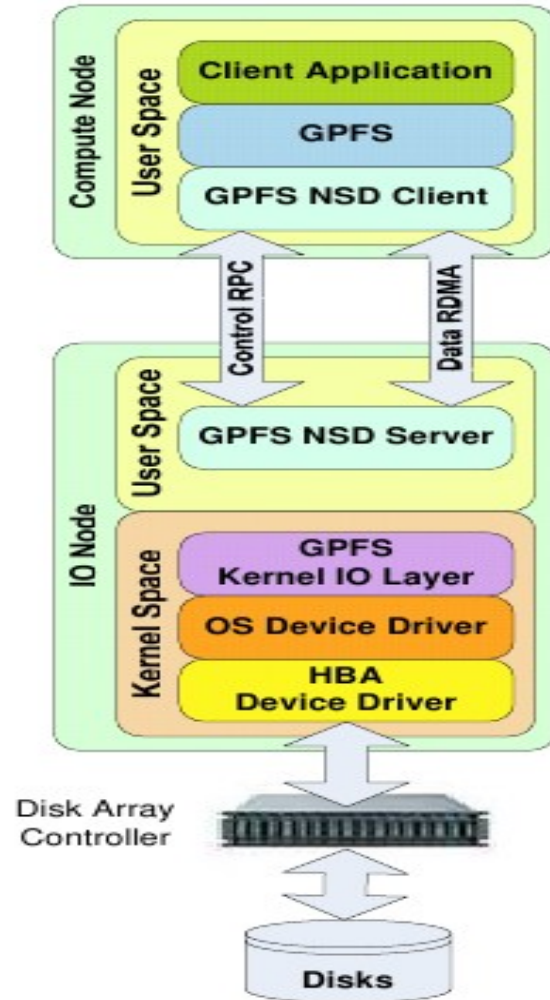
SSD MTBF is ~ 1,2 million hours

Lets say we use the drive intensively office hours .. so 8h / day
Lets say we have 1000 SSDs

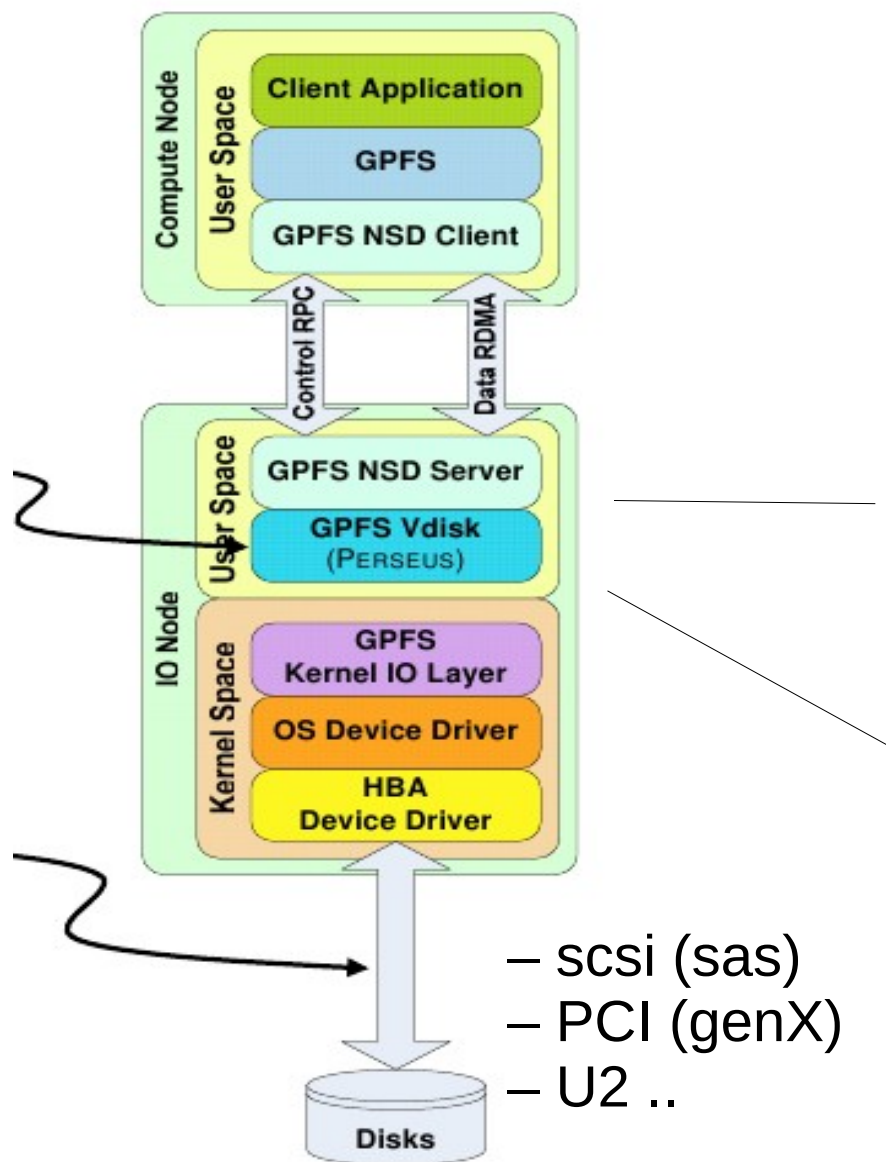
$$1.200.000 / 8000 = 150 \text{ days} \quad ;-)$$

GPFS native Raid / G N R
as ECE

Software defined RAID in SpectrumScale



Software defined RAID in SpectrumScale



check sum
protected

Software RAID / erasure coding

- 2,3,4 Way replication
- $8+2p$
- $8+3p$
- $4+2p$

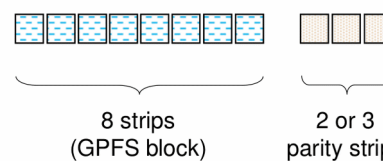
2-fault
tolerant
codes

$8 + 2p$ Reed Solomon



3-fault
tolerant
codes

$8 + 3p$ Reed Solomon



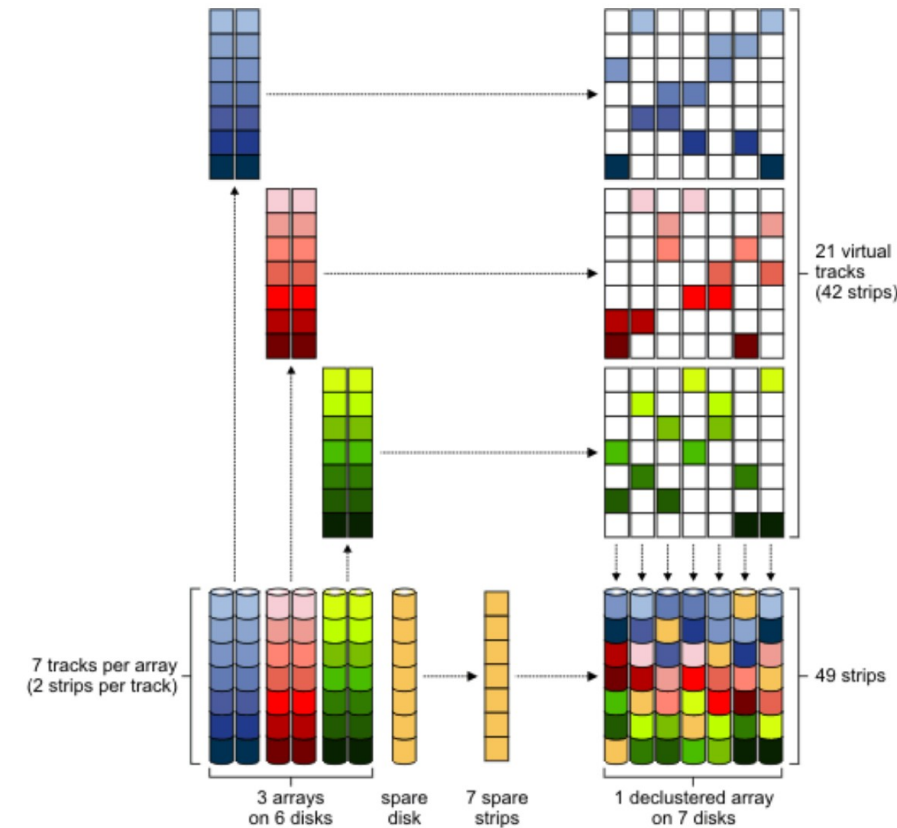
check sum
protected
AND! sequence number

high speed
networks

high speed
disk access

Spectrum Scale Erasure Code Advantages

- Declustered erasure coding provides for data and parity to be distributed over all the disks and nodes in the declustered Array for fastest performance out of the chosen media
 - Faster and more intelligent rebuild operations, using more drives in parallel
 - Prioritize normal vs critical conditions to better use node resources
 - Spare capacity is also distributed across all drives and nodes, so no dedicated spare disks are needed
- Improved storage efficiency and performance
 - 8+2P and 8+3P utilize less overhead vs 100% - 200% for 2X-3X replication
 - Patented algorithms optimize I/O data paths, read and multi-layer write caching



Rebuilding concurrent disk failures – 3 faults

...for a 8+3p example.. up to 512 drive within one DA

- Stripes with greatest number of failures have highest rebuild overhead, but are a smallest fraction of total stripes.
- Failure rate*:
 - 2.1% of all stripes affected by concurrent failure of 1 disk
 - 0.04 % of all stripes affected by concurrent failure of 2 disks
 - 0.000074 % of all stripes affected by concurrent failure of 3 disks

Only 3/8 out of the full stripe need to be relocated



* Example assumes

8+3p stripes (M=11), and N = 512 HDDs:

Fraction of stripes with 1 failure = $M/N = 11/512 = 2,1\%$

Fraction of stripes with 2 failure = $M/N * (M-1)/(N-1) = 11/512 * 10/511 = 0.0420437866928 \%$

Fraction of stripes with 3 failures = $M/N * (M-1)/(N-1) * (M-2)/(N-2) = 11/512 * 10/511 * 9/510 = 7.41949176931e-04 \%$

Spectrum Scale Erasure Code - Disk Hospital

- Identify device problems before hard drive failure:
 - Dead or misbehaving disks
 - Connectivity issues
 - Media errors
 - Slow drives
- Attempt corrective action to revive sick or failing devices:
 - Power cycle non-responsive drives
 - Recompute and rewrite corrupted data
 - Rediscover disk connectivity
- Maintain “health record” for each device
 - If device is accumulating too many errors, remove from service
 - If device is persistently slow, remove from service



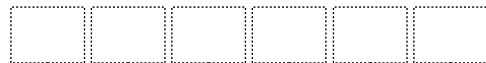
Spectrum Scale Erasure Code - Integrity Management

- Every IO has a checksum added to data trailer
- For writes, verify data integrity when data passes from
 - Client (compute node) to storage node
 - Storage node to storage media
 - Writes also include a sequence number in the metadata to detect
 - dropped/skipped writes
- For reads, verify data integrity when data passes from
 - Storage media to storage node
 - Storage node to client
- A background scrub task periodically detects and fixes silent data corruption on the storage devices
- Automatic data rebuild on failure, automatic rebalance on recovery or when new storage is added
- Rebuild has minimal impact on system performance
 - Rebuild is distributed across disks and nodes
 - Rebuild can be deferred with sufficient protection
- Failure domain for high hardware failure tolerance



Looking closer ;-)- RAID LEVEL and check sum

GPFS

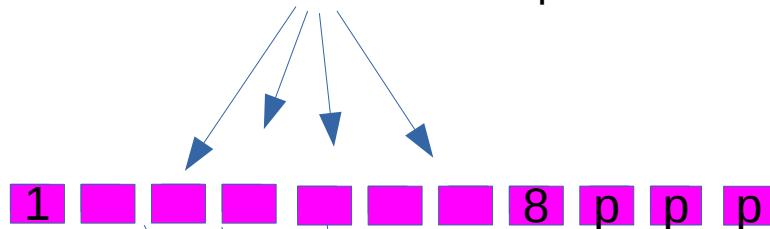


blocksize=4M

```
># dd if=/dev/zero bs=4M of=/net/gpfs1/einfile count=1
1+0 records in
1+0 records out
4194304 bytes (4.2 MB) copied, 0.003221 s, 1.3 GB/s
```



8+3p /BS=4M



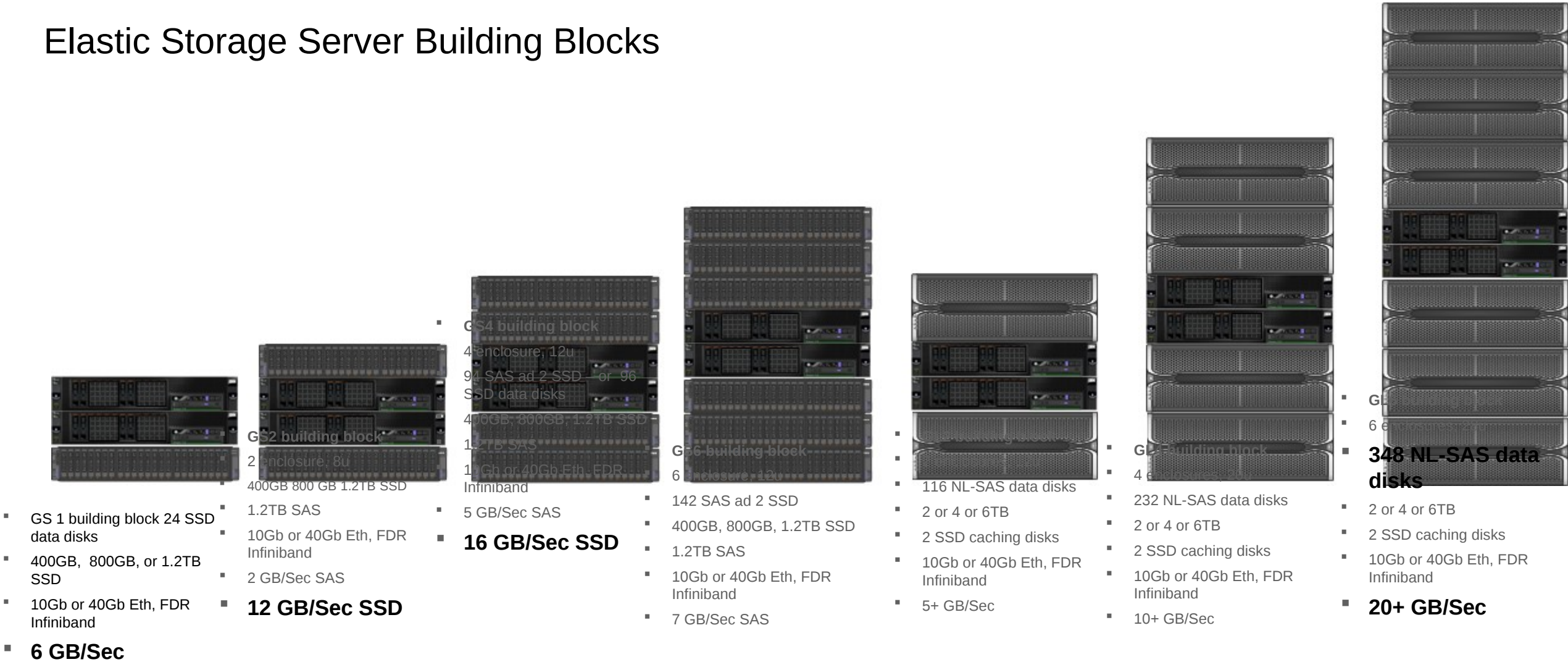
IO size: 1032 sectors x 512 Byte ~ 512K

```
TRACE_IO: FIO: write data tag 0 16 buf 0x4016DA8000 disk 80D0 da 8:5757872 nSectors 1032 err 0
TRACE_IO: FIO: write data tag 1 16 buf 0x4016E30000 disk 42B0 da 9:7986000 nSectors 1032 err 0
TRACE_IO: FIO: write data tag 2 16 buf 0x4016EB8000 disk 46A0 da 10:00476464 nSectors 1032 err 0
TRACE_IO: FIO: write data tag 3 16 buf 0x4016F40000 disk 8700 da 0:0008576 nSectors 1032 err 0
TRACE_IO: FIO: write data tag 4 16 buf 0x4016FC8000 disk 46E0 da 1:8248040 nSectors 1032 err 0
TRACE_IO: FIO: write data tag 5 16 buf 0x4017050000 disk 8110 da 2:2218928 nSectors 1032 err 0
TRACE_IO: FIO: write data tag 6 16 buf 0x40170D8000 disk 8120 da 3:14408624 nSectors 1032 err 0
TRACE_IO: FIO: write data tag 7 16 buf 0x4017160000 disk 8530 da 4:15981488 nSectors 1032 err 0
TRACE_IO: FIO: write data tag 8 16 buf 0x40171E8000 disk 8740 da 5:12442544 nSectors 1032 err 0
TRACE_IO: FIO: write data tag 9 16 buf 0x40171F0000 disk 4670 da 6:7592880 nSectors 1032 err 0
TRACE_IO: FIO: write data tag 10 16 buf 0x40171F8000 disk 8260 da 7:5626800 nSectors 1032 err 0
```


Erasure Code shipped in ESS (IBM Elastic Storage Server)



Elastic Storage Server Building Blocks

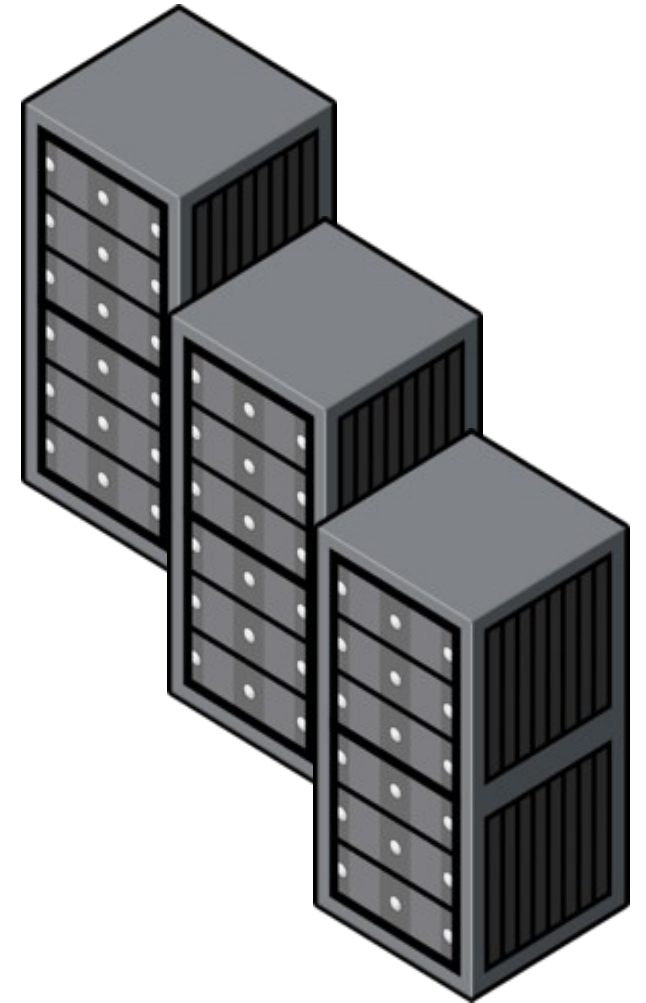


What is Spectrum Scale Erasure Code Edition?

What is Spectrum Scale Erasure Code Edition?

A new Spectrum Scale offering that brings all of the benefits of “Data Management Edition” *plus* **Spectrum Scale RAID**

- Spectrum Scale running in storage rich servers connected to each other with a high speed network infrastructure
- Bring your own hardware – select any hardware that meets minimum requirements
 - Provides Storage devices can be HDD, SSD, NVMe or a mixture
- features of an Enterprise Storage Controller all in software
 - Enterprise ready storage software used in Spectrum Scale Elastic Storage Server (ESS)
- Restricted GA June 2019 *



Architectural details

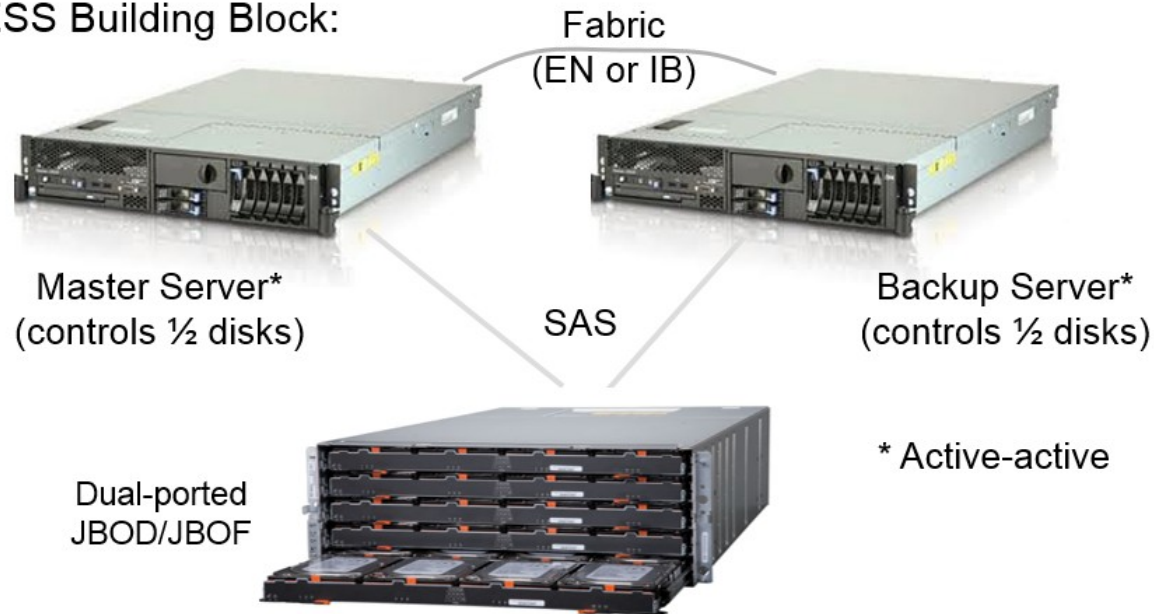
Comparison with ESS

Hardware Architecture Comparison

ESS

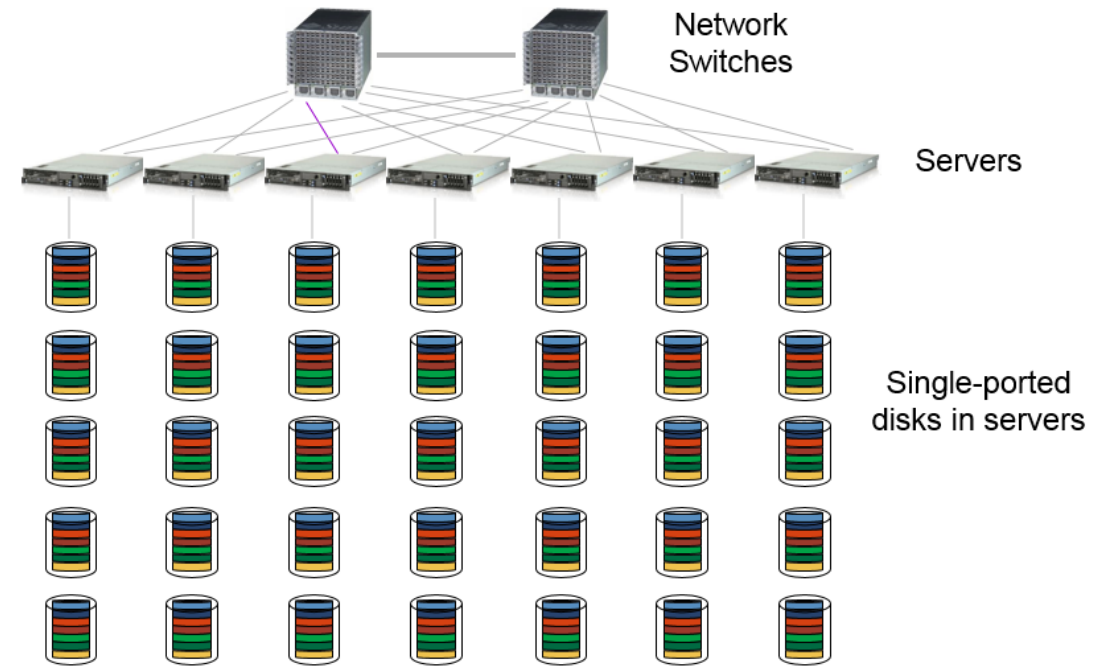
- Twin-tailed disks, dual servers – provide very high availability
- However, in case when a failure of both the master and backup servers happens it results in data unavailability

ESS Building Block:



ECE

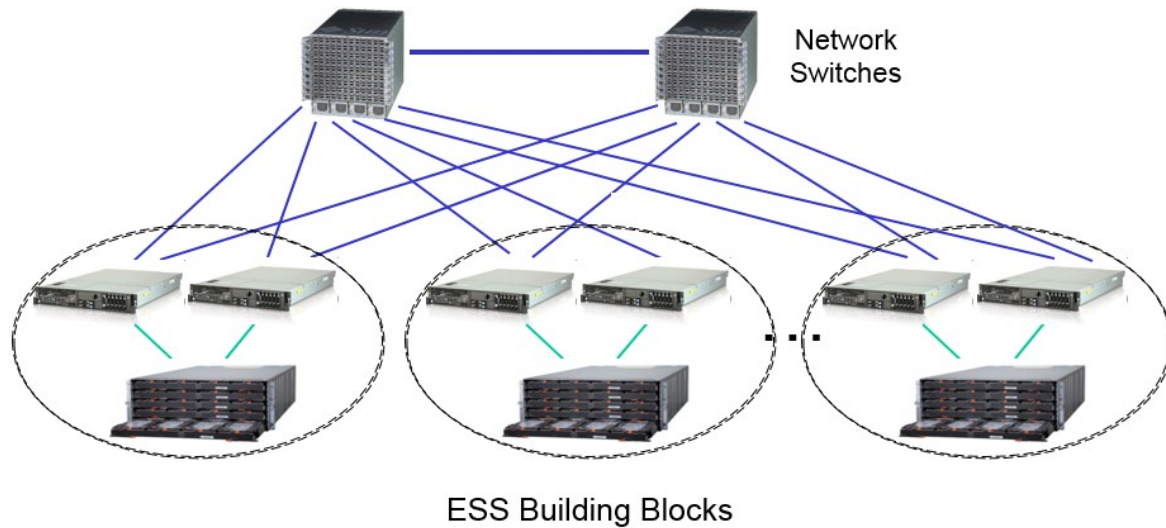
- Network RAID Internal disk rich commodity servers
- Tolerates concurrent failure of an arbitrary pair of servers (or 3 servers if 8+3p erasure code) and disks



Scale-out Cluster with Multiple Building Blocks

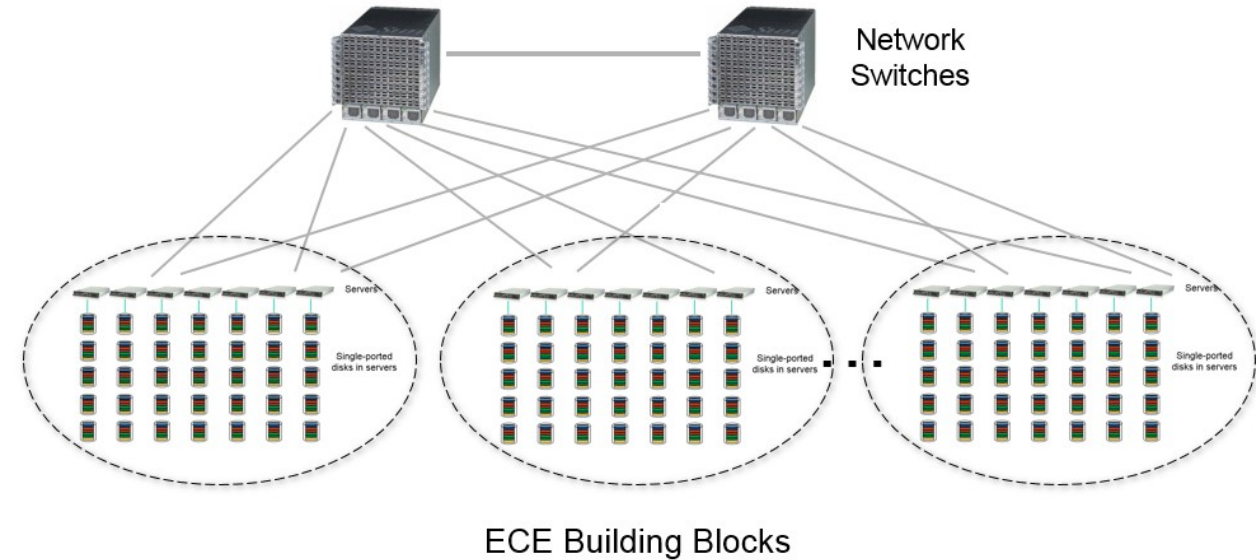
ESS

- Twin-tailed disks, dual servers building block
- Multiple ESS building blocks in the same Scale cluster



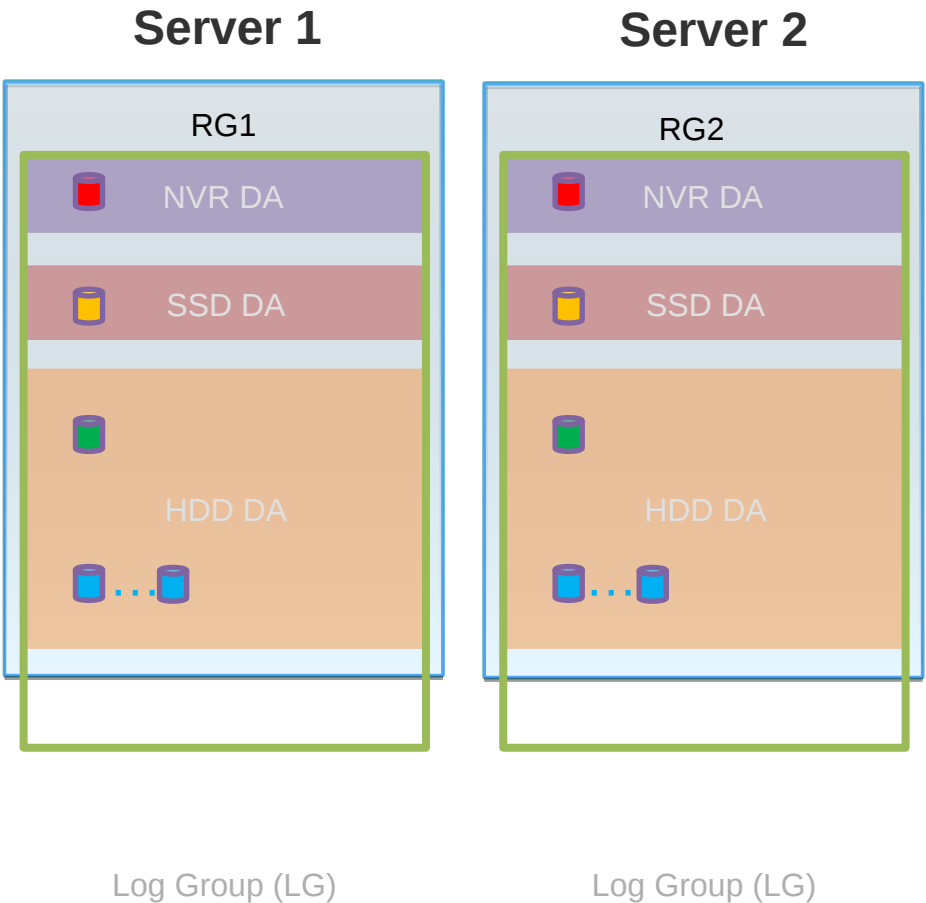
ECE

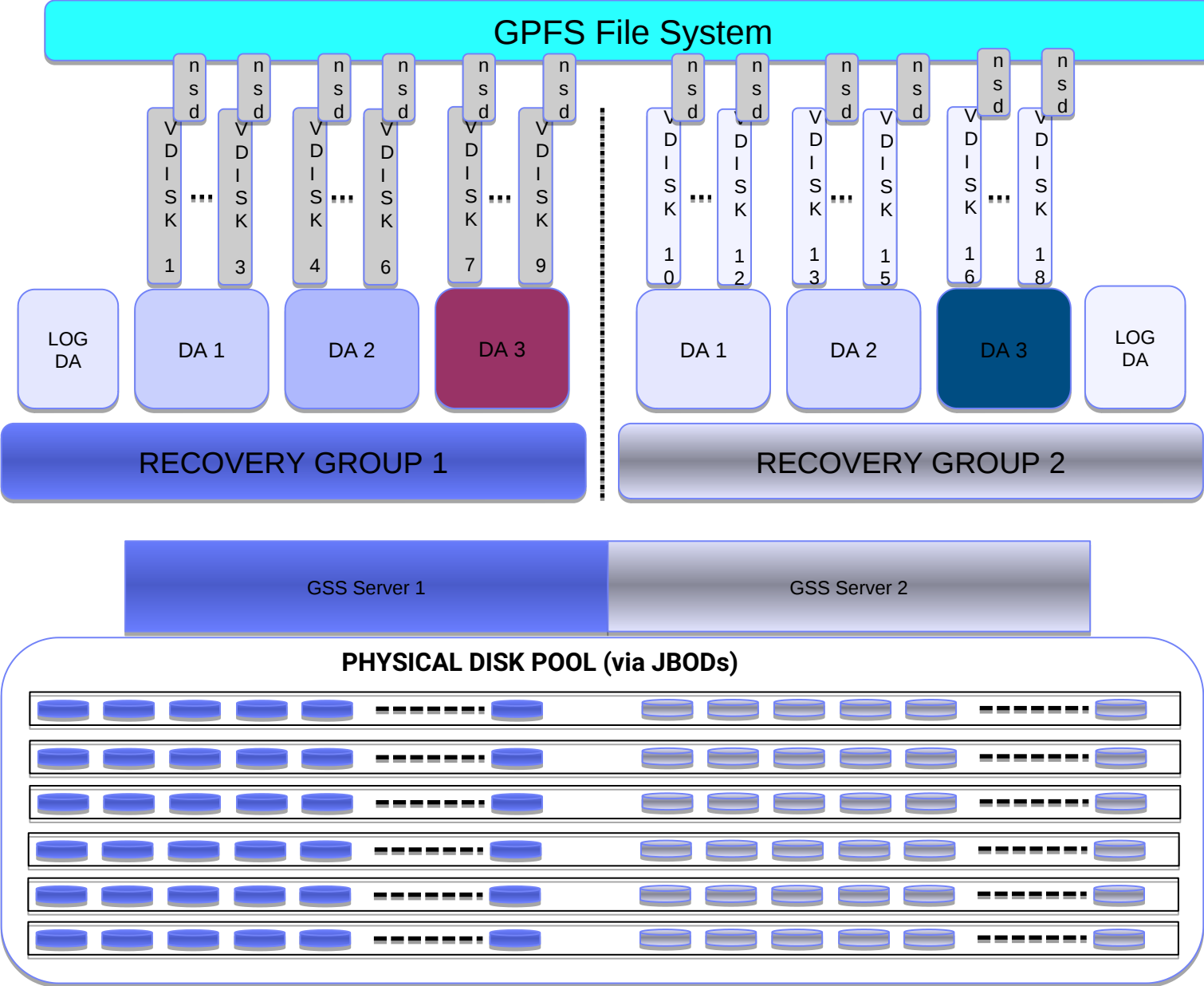
- Commodity servers based ECE cluster (building block)
- Multiple ECE clusters in the same Scale cluster



Hardware Resource Partitioning

ESS Building Block





1 RAID level per vdisk
(8+2P, 8+3P, 2-way, 3-way, 4-way)
-> Allows very small
file system on all disks!

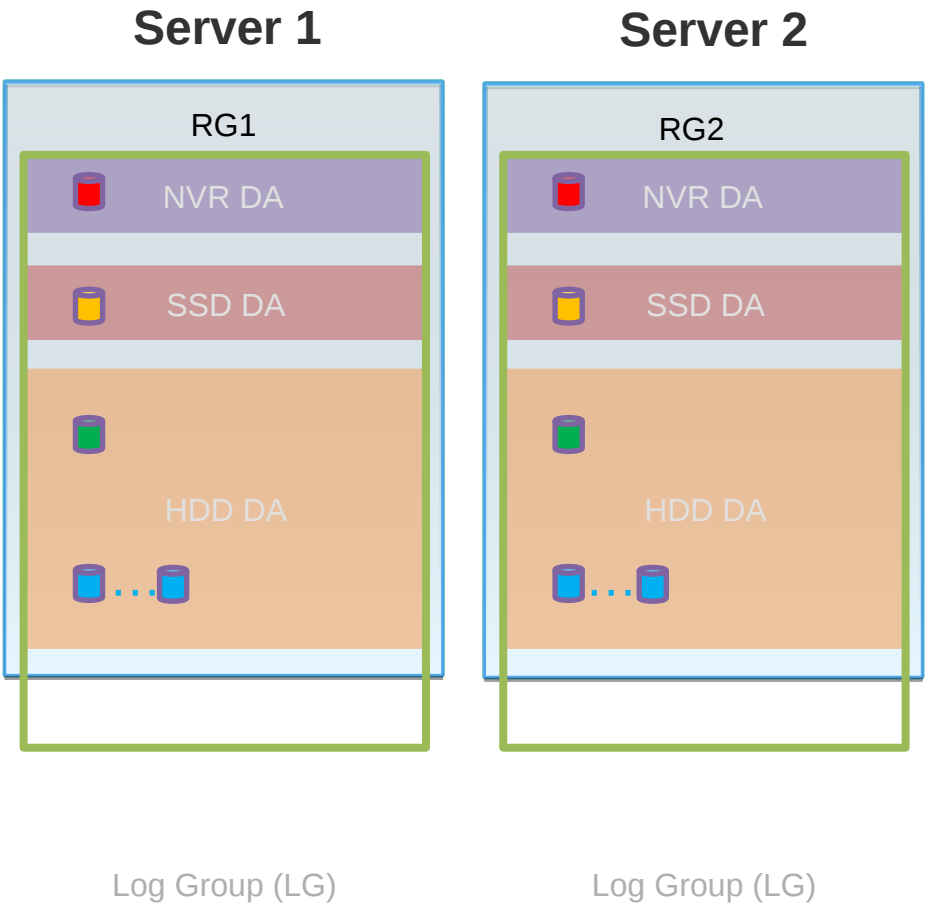
1-N Vdisk inside 1 DA

No multipathing, 1:1
mapping

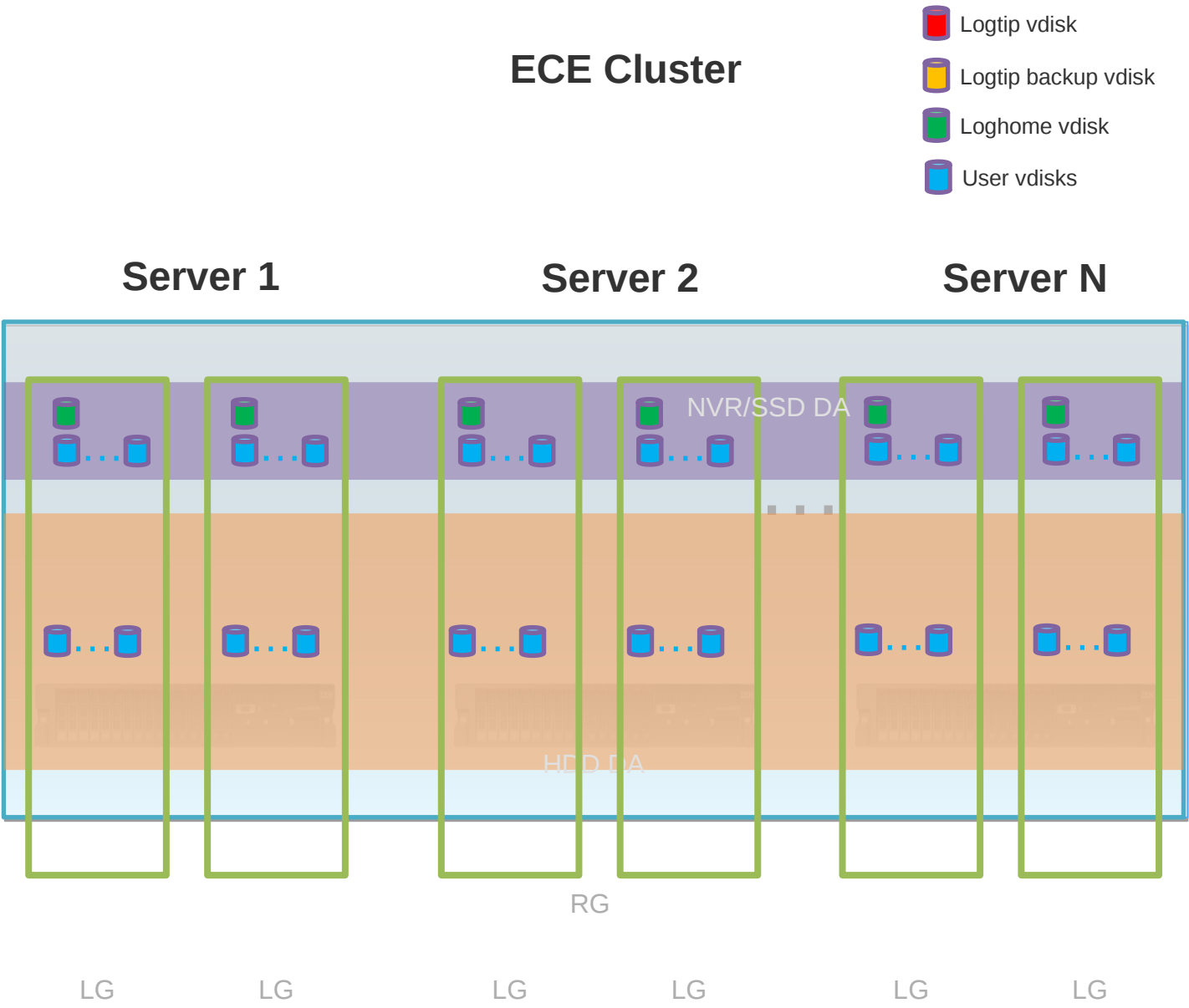
Failover on node
failure only

Hardware Resource Partitioning

ESS Building Block

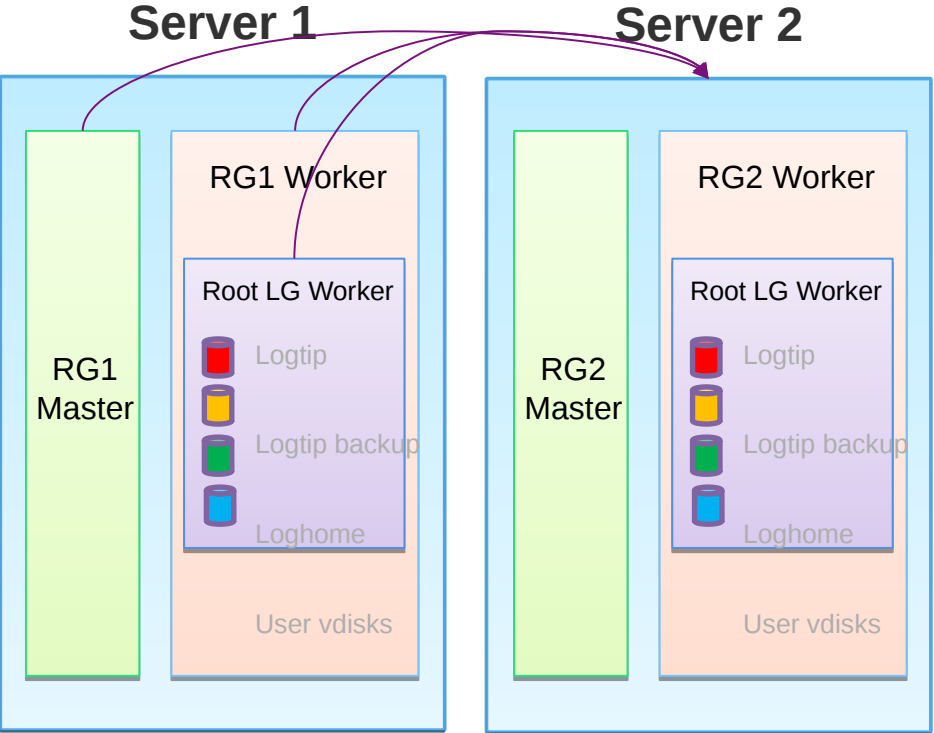


ECE Cluster

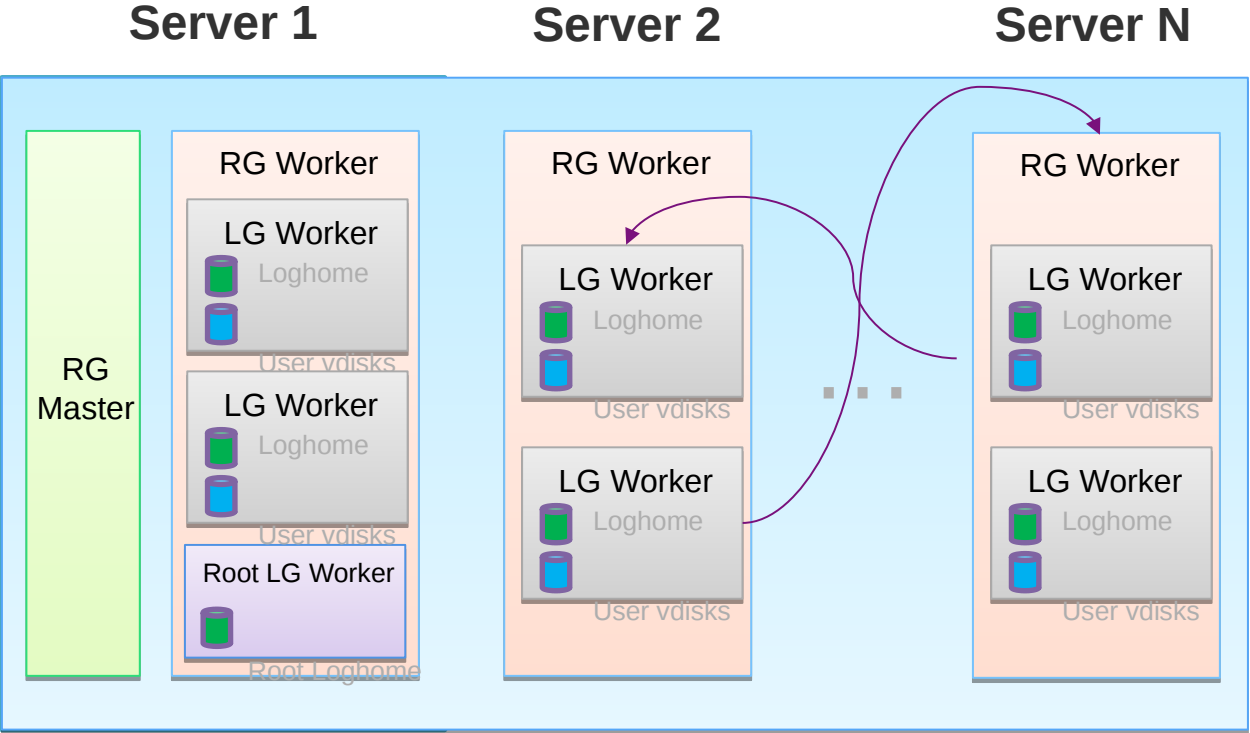


Software Failover Architecture

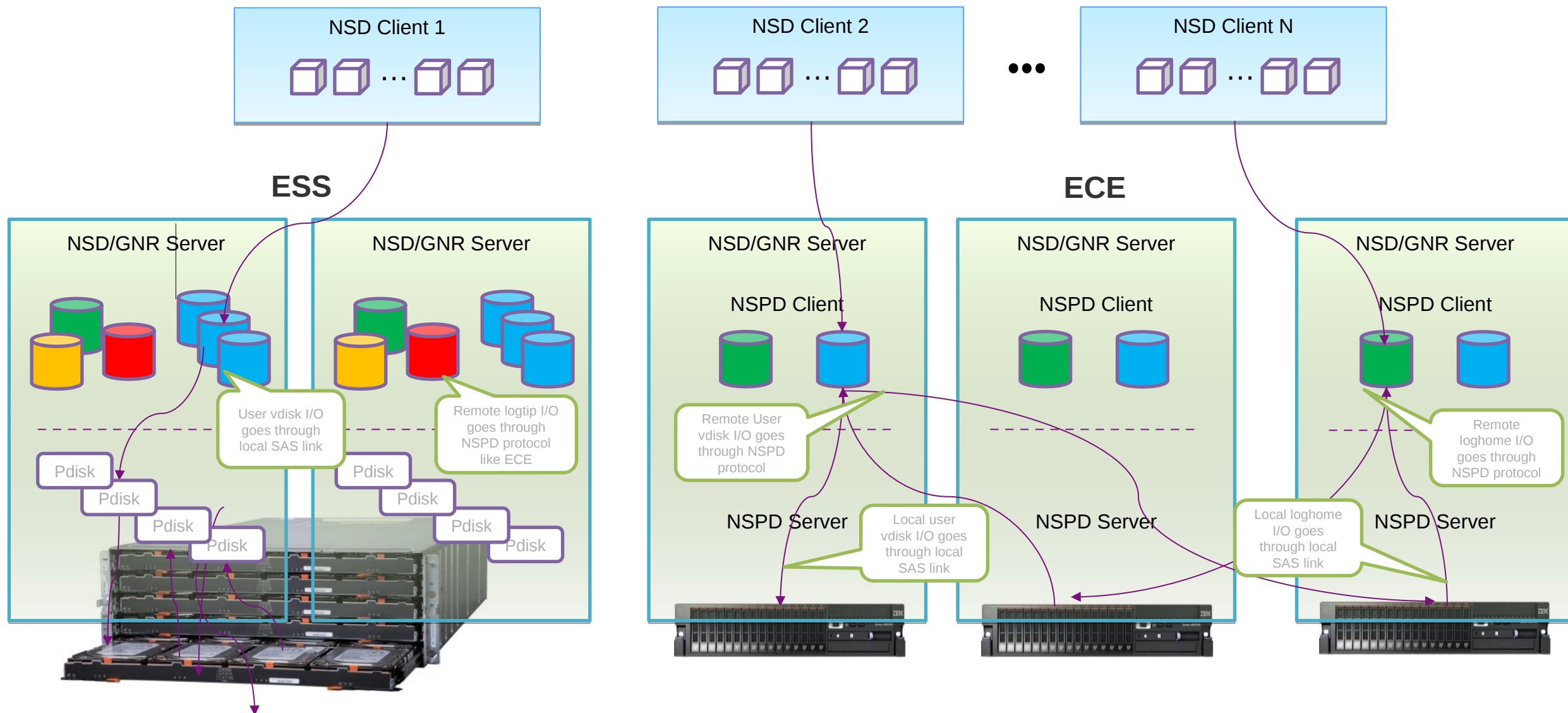
ESS



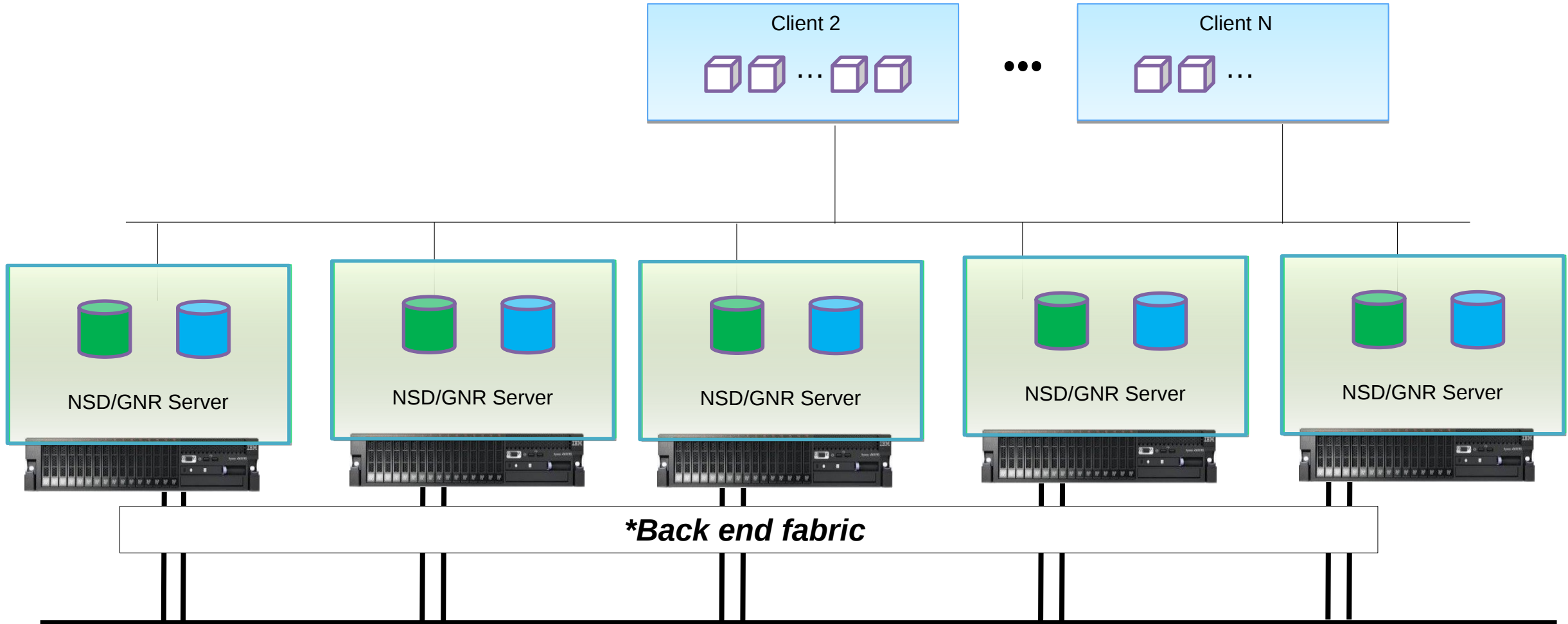
ECE



NSD I/O Path



design the network – network is the key



*optionally , but adds a lot of performance,
* can be ETH or infiniband or RoCE

* optionally, use multicluster and subnet parameter

ECE Requirements

ECE Hardware Requirements for each Storage Server



CPU architecture	x86 64 bit processor with 8 or more processor cores per socket. Server should be dual socket with both sockets populated
Memory	<ul style="list-style-type: none">• 64 GB or more for configurations with up to 24 drives per node. For NVMe configurations, it is recommended to utilize all available memory DIMM sockets to get optimal performance.• For server configurations with more than 24 drives per node, contact IBM® for memory requirements.
Server packaging	Single server per enclosure. Multi-node server packaging with common hardware components that provide a single point of failure across servers is not supported at this time.
System drive	A physical drive is required for each server's system disk. Recommend RAID1 protected and have a capacity of 100 GB or more.
SAS Host Bus Adapter*	LSI SAS HBA, models SAS3108, SAS3216 or SAS3516.
SAS Data Drives	SAS or NL-SAS HDD or SSDs in JBOD mode. SATA drives are not supported at this time.
NVMe Data Drives	Enterprise class NVMe drives with U.2 form factor.
Fast Media Req.	At least one SSD or NVMe drive is required in each server for IBM Spectrum Scale Erasure Code Edition logging.

ECE OS & Network Requirements for each Storage Server



Operating system	RHEL 7.5 or 7.6. See IBM Spectrum™ Scale FAQ for details of supported versions.
Network Adapter	Mellanox ConnectX-4 or ConnectX-5, (Ethernet or InfiniBand)
Network Bandwidth	25 Gbps or more between storage nodes. Higher bandwidth may be required depending on the workload requirements.
Network Latency	Average latency must be less than 1 msec between any storage nodes.
Network Topology	To achieve the maximum performance for a workload, a dedicated storage network is recommended. For other workloads, a separate network is recommended but not required.

Erasure Code Options and Failure Tolerance



Erasure Code	4+2P	4+3P	8+2P	8+3P
Number of Nodes in RG				
4-5	Not Recommended 1 Node	1 Node + 1 Device	Not Recommended 0 Nodes	Not Recommended 1 Node
6-8	2 Nodes	2 Nodes* (Limited by RG descriptors)	Not Recommended 1 Node	Not Recommended 1 Node + 1 Device
9	2 Nodes	3 Nodes	Not Recommended 1 Node	Not Recommended 1 Node + 1 Device
10	2 Nodes	3 Nodes	2 Nodes	2 Nodes
11+	2 Nodes	3 Nodes	2 Nodes	3 Nodes

Installation and Hardware Pre-check



The IBM Spectrum Scale Erasure Code Edition precheck, integrated in the installation toolkit installation, deployment or upgrade precheck. The ECE check: standalone, publicly available, open source

For IBM Spectrum Scale Erasure Code Edition, the pre-check includes the following on all scale-out nodes:

- Check CPU requirements (Server cpu type/number of sockets/number of cores)
- Check memory requirements (Server memory & DIMM utilization)
- Confirm consistent, allowable disk topology
- Check OS and firmware levels
- Check whether the networking requirements including the required NIC and SAS adapters are met
- Check whether the required syscall parameters are set correctly

Installation toolkit-related prerequisites

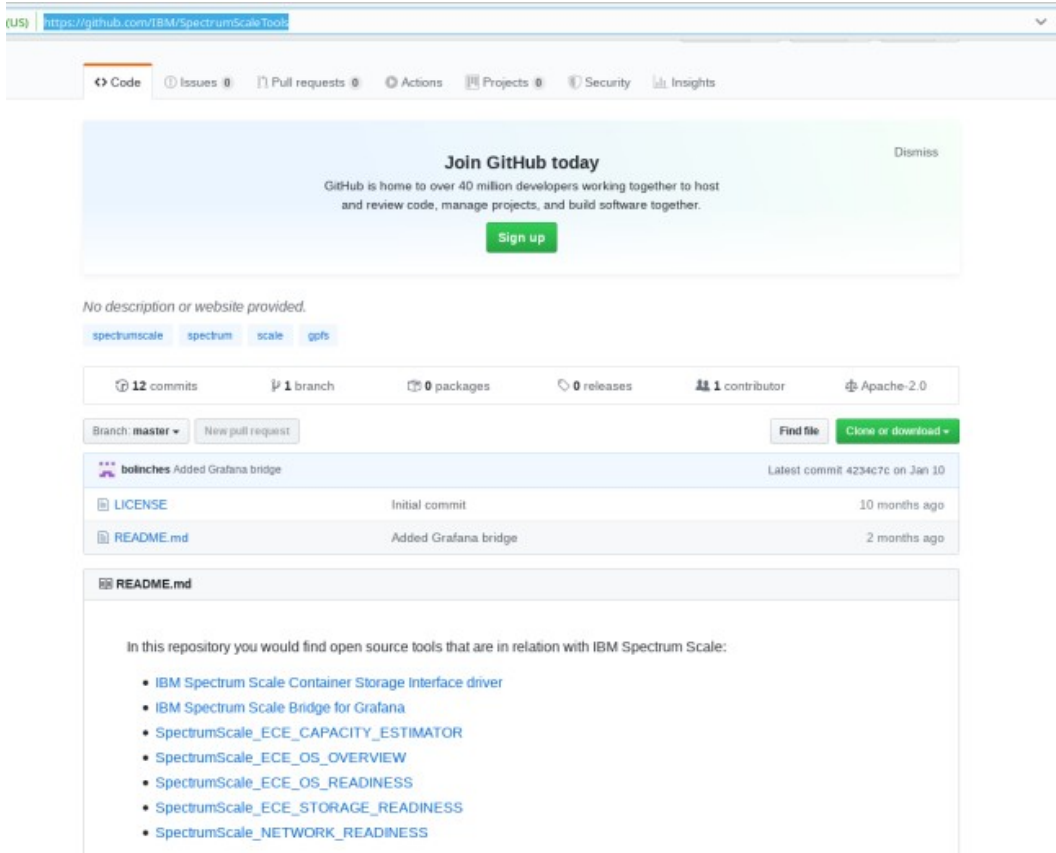
- Ensure that networking is set up in one of the following ways.
- DNS is configured such that all host names, either short or long, are resolvable.
- All host names are resolvable in the /etc/hosts file. The host entries in the /etc/hosts file must be in the following order:<IP address> <Fully qualified domain name> <Short name>
- Passwordless SSH must be set up using the FQDN and the short name of the node

Hardware precheck - verify minimum levels and consistency across Recovery Groups via toolkit

Test results saved with install log for installation record

https://www.ibm.com/support/knowledgecenter/en/STXKQY_ECE_5.0.4/com.ibm.spectrum.scale.ece.v5r04.doc/b1lece_min_hwrequirements.htm

Installation and Hardware Pre-check



- ssh keys verteilen
- fping

<https://github.com/IBM/SpectrumScaleTools>

High Level / prechecks

- Network precheck between every ECE storage node
 - Average latency < 1 msec
 - Maximum latency < 2 msec
 - Standard Deviation < 0.33 msec
- Network KPI check for network assessment
 - Standalone (based on nsdperf)
 - Publicly available
 - Open source (nsdperf becomes opensource software)

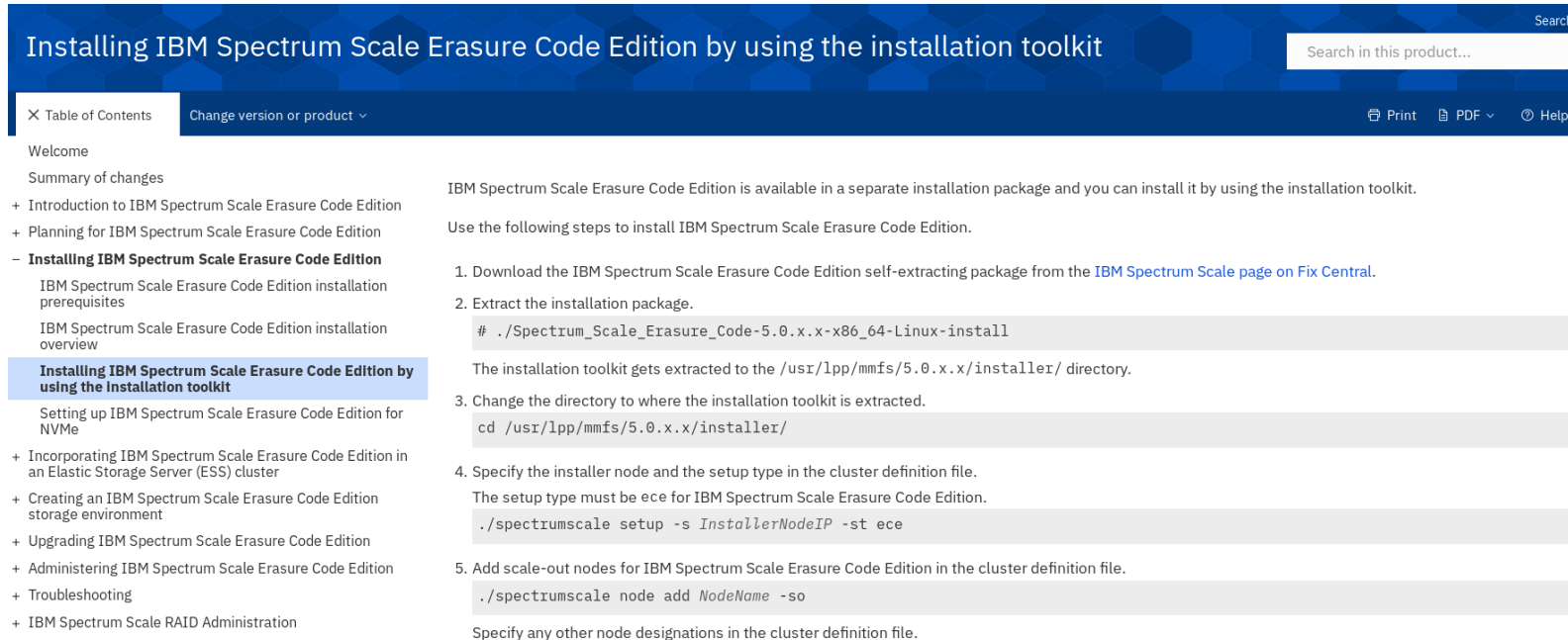
```
[root@c8n1 tmp]# ll
total 152
drwxr-xr-x. 2 root root    195 Jan  3 03:41 SpectrumScale_ECE_OS_READINESS-master
-rw-r--r--. 1 root root 22906 Feb  5 15:32 SpectrumScale_ECE_OS_READINESS-master.zip
drwxr-xr-x. 2 root root    167 Feb  3 03:36 SpectrumScale_ECE_STORAGE_READINESS-master
-rw-r--r--. 1 root root 18905 Feb  5 15:32 SpectrumScale_ECE_STORAGE_READINESS-master.zip
drwxr-xr-x. 3 root root    4096 Feb  5 15:58 SpectrumScale_NETWORK_READINESS-master
-rw-r--r--. 1 root root 102852 Feb  5 15:32 SpectrumScale_NETWORK_READINESS-master.zip
[root@c8n1 tmp]#
```

Setup and Install

ECE Installation Steps

The install process

https://www.ibm.com/support/knowledgecenter/STXKQY_ECE_5.0.4/com.ibm.spectrum.scale.ece.v5r04.doc/bl1ece_installingscaleecewithtoolkit.htm



The screenshot shows the 'Installing IBM Spectrum Scale Erasure Code Edition by using the installation toolkit' page. The page has a blue header with a search bar and navigation links. The left sidebar contains a table of contents with the following items: Welcome, Summary of changes, Introduction to IBM Spectrum Scale Erasure Code Edition, Planning for IBM Spectrum Scale Erasure Code Edition, **Installing IBM Spectrum Scale Erasure Code Edition** (highlighted), IBM Spectrum Scale Erasure Code Edition installation prerequisites, IBM Spectrum Scale Erasure Code Edition installation overview, **Installing IBM Spectrum Scale Erasure Code Edition by using the Installation toolkit** (highlighted), Setting up IBM Spectrum Scale Erasure Code Edition for NVMe, Incorporating IBM Spectrum Scale Erasure Code Edition in an Elastic Storage Server (ESS) cluster, Creating an IBM Spectrum Scale Erasure Code Edition storage environment, Upgrading IBM Spectrum Scale Erasure Code Edition, Administering IBM Spectrum Scale Erasure Code Edition, Troubleshooting, and IBM Spectrum Scale RAID Administration. The main content area contains the following text and steps:

IBM Spectrum Scale Erasure Code Edition is available in a separate installation package and you can install it by using the installation toolkit.

Use the following steps to install IBM Spectrum Scale Erasure Code Edition.

1. Download the IBM Spectrum Scale Erasure Code Edition self-extracting package from the [IBM Spectrum Scale page on Fix Central](#).
2. Extract the installation package.

```
# ./Spectrum_Scale_Erasure_Code-5.0.x.x-x86_64-Linux-install
```

The installation toolkit gets extracted to the `/usr/lpp/mmfs/5.0.x.x/installer/` directory.
3. Change the directory to where the installation toolkit is extracted.

```
cd /usr/lpp/mmfs/5.0.x.x/installer/
```
4. Specify the installer node and the setup type in the cluster definition file.
The setup type must be `ece` for IBM Spectrum Scale Erasure Code Edition.

```
./spectrumscale setup -s InstallerNodeIP -st ece
```
5. Add scale-out nodes for IBM Spectrum Scale Erasure Code Edition in the cluster definition file.

```
./spectrumscale node add NodeName -so
```

Specify any other node designations in the cluster definition file.

Phase 1: regular install / cluster deploy

Phase 2: check/add GNR rpms

Phase 3: deploy ECE with mmvdisk

ECE Installation Steps

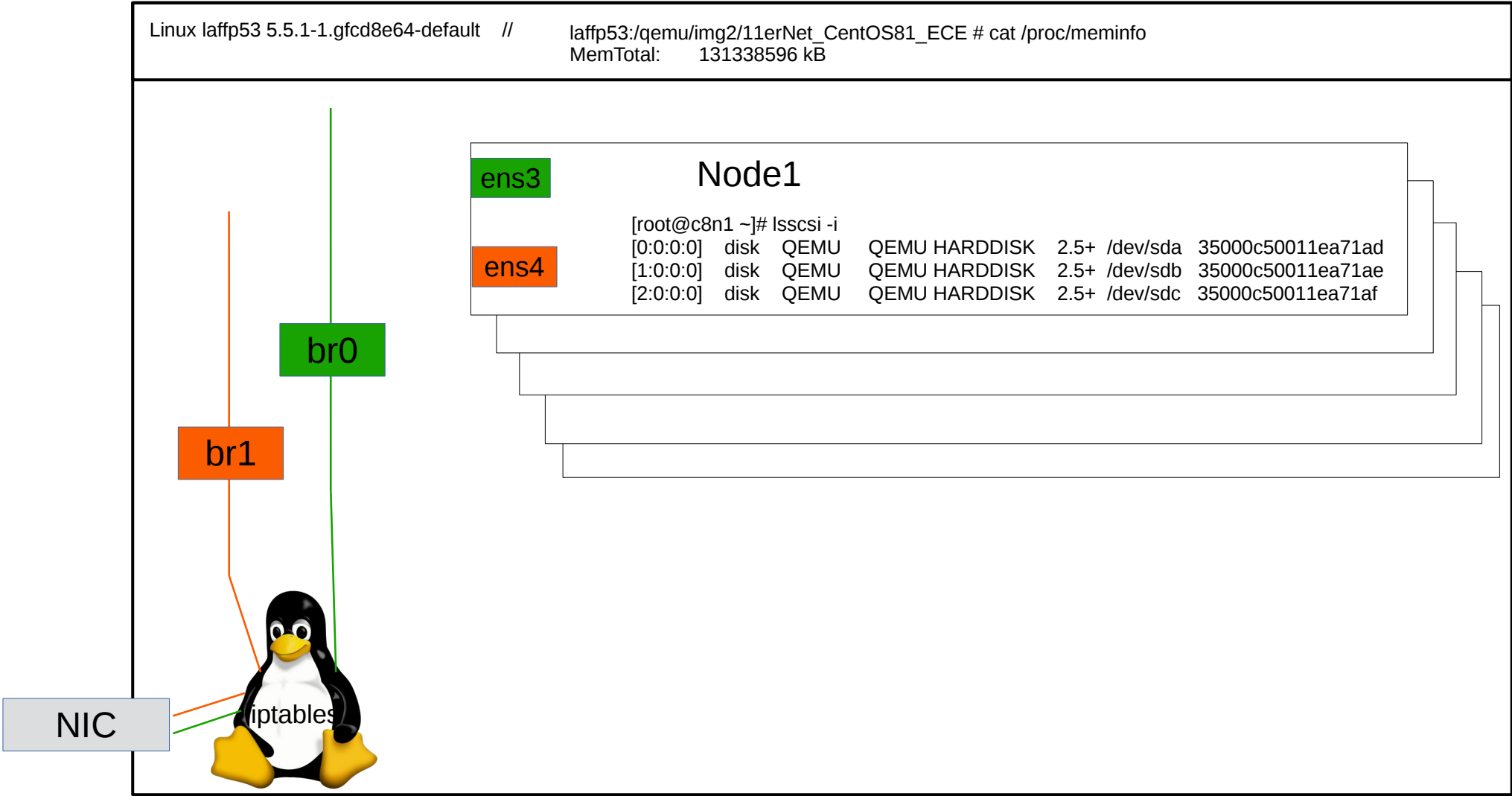
Phase 1: regular install / cluster deploy

Phase 2: check/add GNR rpms

Phase 3: deploy ECE with mmvdisk

- prepare the nodes
- create a node class (group all nodes)
- configure this nodeclass for GNR .. (set and check settings)
- create RG
- create vdisks + NSDs
- create file system

a nice KVM test setup



Qemu config / example



```
laffp53:/qemu/img2/11erNet_CentOS81_ECE # cat start_qemu_11er
```

```
#network
```

```
ip link add br0 type bridge
ip a add 10.0.11.1/24 dev br0
ip link set br0 up
```

```
ip link add br1 type bridge
```

```
/usr/bin/qemu-kvm -hda /qemu/img2/11erNet_CentOS81_ECE/c8n1_snap.img -m 12G -vnc :11 \
-net nic,macaddr=12:20:01:12:34:11 -net tap,script=/etc/qemu-ifup.br0 \
-net nic,macaddr=12:20:01:12:34:12 -net tap,script=/etc/qemu-ifup.br1 \
-drive if=scsi,id=hd,file=/qemu/img2/11erNet_CentOS81_ECE/c8n1_d1.omg -device virtio-scsi-pci,id=scsi0 --enable-kvm -device scsi-hd,wwn=0x5000c50011ea71ad,serial=S1WNG0M507681D,drive=hd,id=sluo,logical_block_size=4096,physical_block_size=4096 \
-drive if=scsi,id=hc,file=/qemu/img2/11erNet_CentOS81_ECE/c8n1_d2.omg -device virtio-scsi-pci,id=scsi1 --enable-kvm -device scsi-hd,wwn=0x5000c50011ea71ae,serial=S1WNG0M507682D,drive=hc,id=slup,logical_block_size=4096,physical_block_size=4096 \
-drive if=scsi,id=he,file=/qemu/img2/11erNet_CentOS81_ECE/c8n1_d3.omg -device virtio-scsi-pci,id=scsi2 --enable-kvm -device scsi-hd,wwn=0x5000c50011ea71af,serial=S1WNG0M507683D,drive=he,id=sluq,logical_block_size=4096,physical_block_size=4096 &
```

```
/usr/bin/qemu-kvm -hda /qemu/img2/11erNet_CentOS81_ECE/c8n2_snap.img -m 12G -vnc :12 \
-net nic,macaddr=12:20:02:12:34:11 -net tap,script=/etc/qemu-ifup.br0 \
-net nic,macaddr=12:20:02:12:34:12 -net tap,script=/etc/qemu-ifup.br1 \
-drive if=scsi,id=hf,file=/qemu/img2/11erNet_CentOS81_ECE/c8n2_d1.omg -device virtio-scsi-pci,id=scsi3 --enable-kvm -device scsi-hd,wwn=0x5000c50012ea72ad,serial=S2WNG0M507681D,drive=hf,id=slbo,logical_block_size=4096,physical_block_size=4096 \
-drive if=scsi,id=hg,file=/qemu/img2/11erNet_CentOS81_ECE/c8n2_d2.omg -device virtio-scsi-pci,id=scsi4 --enable-kvm -device scsi-hd,wwn=0x5000c50012ea72ae,serial=S2WNG0M507682D,drive=hg,id=slbp,logical_block_size=4096,physical_block_size=4096 \
-drive if=scsi,id=hh,file=/qemu/img2/11erNet_CentOS81_ECE/c8n2_d3.omg -device virtio-scsi-pci,id=scsi5 --enable-kvm -device scsi-hd,wwn=0x5000c50012ea72af,serial=S2WNG0M507683D,drive=hh,id=slbq,logical_block_size=4096,physical_block_size=4096 &
```

```
laffp53:/qemu/img2/11erNet_CentOS81_ECE # cat /etc/qemu-ifup.br0
#!/bin/sh
echo "ip link set $1 up"
ip link set $1 up
ip link set $1 master br0
```

ECE deploy

```
[root@c8n1 ~]# mmlscluster
```

```
GPFS cluster information
```

```
=====
```

```
GPFS cluster name:      myBeer.c8n1
GPFS cluster id:        15550968977049095596
GPFS UID domain:        myBeer.c8n1
Remote shell command:    /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:         CCR
```

```
Node  Daemon node name
```

```
-----
1     c8n1
2     c8n2
3     c8n3
4     c8n4
5     c8n5
```

```
[root@c8n1 ~]#
```

```
[root@c8n1 ~]# mmvdisk server list --disk-topology -N all
```

node number	server	needs attention	matching metric	disk topology
1	c8n1	no	100/100	ECE 3 HDD
2	c8n2	no	100/100	ECE 3 HDD
3	c8n3	no	100/100	ECE 3 HDD
4	c8n4			
5	c8n5			

```
[root@c8n1 ~]#
```

```
[root@fscc-sr650-13 SpectrumScale_NETWORK_READINESS-master]# mmvdisk server list --disk-topology -N all
```

node number	server	needs attention	matching metric	disk topology
1	ece_13.localnet.com	no	100/100	ECE 2 SSD/NVMe
2	ece_14.localnet.com	no	100/100	ECE 2 SSD/NVMe
3	ece_15.localnet.com	no	100/100	ECE 2 SSD/NVMe
4	ece_16.localnet.com	no	100/100	ECE 2 SSD/NVMe
5	ece_17.localnet.com	no	100/100	ECE 2 SSD/NVMe
6	ece_18.localnet.com	no	100/100	ECE 2 SSD/NVMe

```
[root@fscc-sr650-13 SpectrumScale_NETWORK_READINESS-master]# mmvdisk
```

command:

```
mmvdisk server list --disk-topology -N all
```

ECE deploy

```
[root@c8n1 ~]# mmlsnodeclass
Node Class Name      Members
-----
beer                 c8n1,c8n2,c8n3,c8n4,c8n5
[root@c8n1 ~]# mmvdisk server ^C
[root@c8n1 ~]# mmvdisk nc ^C
[root@c8n1 ~]# mmvdisk nc list
```

```
node class  recovery groups
-----
beer        -
```

```
[root@c8n1 ~]#
```

command

```
mmvdisk nc create --node-class beer -N all
```

```
[root@c8n1 ~]# mmvdisk server configure --node-class beer
mmvdisk: Checking resources for specified nodes.
mmvdisk: Node 'c8n2' has a scale-out recovery group disk topology.
mmvdisk: Node 'c8n1' has a scale-out recovery group disk topology.
mmvdisk: Node 'c8n4' has a scale-out recovery group disk topology.
mmvdisk: Node 'c8n5' has a scale-out recovery group disk topology.
mmvdisk: Node 'c8n3' has a scale-out recovery group disk topology.
mmvdisk: Node class 'beer' has a scale-out recovery group disk topology.
mmvdisk: Setting configuration for node class 'beer'.
mmvdisk: Node class 'beer' is now configured to be recovery group servers.
mmvdisk: For configuration changes to take effect, GPFS should be restarted
mmvdisk: on node class 'beer'.
[root@c8n1 ~]# █
```

pay attention „- - “

command

```
mmvdisk server configure --node-class beer --recycle all
```


command

```
mmvdisk server configure --node-class beer --recycle all
```

mmfsconfig

```
[beer]
nsdRAIDTracks 131072
nsdSmallThreadRatio 1
nsdMinWorkerThreads 3842
nsdMaxWorkerThreads 3842
nsdRAIDEventLogToConsole all
nsdRAIDBlockDeviceMaxSectorsKB 0
nsdRAIDBlockDeviceNrRequests 0
nsdRAIDBlockDeviceQueueDepth 0
nsdRAIDBlockDeviceScheduler off
nsdRAIDMaxPdiskQueueDepth 128
nsdRAIDSmallThreadRatio 2
nsdRAIDDefaultGeneratedFD yes
nsdRAIDMasterBufferPoolSize 2G
nsdRAIDThreadsPerQueue 16
nsdRAIDDiskCheckVWCE yes
panicOnIOHang yes
maxMBps 24000
maxFilesToCache 128k
maxStatCache 128k
ignorePrefetchLUNCount yes
prefetchPct 50
pitWorkerThreadsPerNode 32
nspdBuflerMemPerQueue 24m
nspdThreadsPerQueue 2
nspdQueues 64
```

– attention , when add/removing nodes to existing nodeclass
– recycle 1

– some of those parameters can be customized

– customized settings will be checked, when **mmvdisk** is used

ECE deploy

```
[root@c8n1 ~]# mmvdisk recoverygroup create --recovery-group rgBeer --node-class beer
mmvdisk: Checking node class configuration.
mmvdisk: Checking daemon status on node 'c8n1'.
mmvdisk: Checking daemon status on node 'c8n2'.
mmvdisk: Checking daemon status on node 'c8n3'.
mmvdisk: Checking daemon status on node 'c8n4'.
mmvdisk: Checking daemon status on node 'c8n5'.
mmvdisk: Node 'c8n1' has a scale-out recovery group disk topology.
mmvdisk: Node 'c8n4' has a scale-out recovery group disk topology.
mmvdisk: Node 'c8n2' has a scale-out recovery group disk topology.
mmvdisk: Node 'c8n3' has a scale-out recovery group disk topology.
mmvdisk: Node 'c8n5' has a scale-out recovery group disk topology.
mmvdisk: Creating recovery group 'rgBeer'.
mmvdisk: Formatting log vdisks for recovery group.
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001R00TLOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG001LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG002LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG003LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG004LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG005LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG006LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG007LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG008LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG009LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG010LOGHOME
mmvdisk: Created recovery group 'rgBeer'.
```

Command

```
mmvdisk recoverygroup create --recovery-group rgBeer --node-class beer
```

Volatile write cache detection

- IBM Spectrum Scale Erasure Code Edition now has the ability to test if volatile write caching mode is enabled on the physical disks.
- SCSI / NVMe drives support a volatile write caching mode
- IBM Spectrum Scale Erasure Code Edition **cannot be used with drives operating in this mode** because on power failure, the cached data is lost, causing already committed data to revert to an older version.
- If IBM Spectrum Scale Erasure Code Edition detects a drive with volatile write caching mode enabled, it puts the pdisk into a new volatile write cache enabled (VWCE) state and drains all data from the drive. If IBM Spectrum Scale Erasure Code Edition detects a large number of drives with volatile write caching enabled, it stops service of the recovery group and waits for volatile write caching mode to be disabled on the drives.
- The volatile write cache detection feature is enabled for all new IBM Spectrum Scale Erasure Code Edition installations starting from version 5.0.4. On previous installations, the feature is disabled by default and must be manually enabled in order to take advantage of the check.

Volatile write cache detection

```
laffp53:~ # which nvme
```

```
/usr/sbin/nvme
```

```
laffp53:~ # rpm -qf /usr/sbin/nvme
```

```
nvme-cli-1.8.1-lp151.5.9.1.x86_64
```

```
laffp53:~ #
```

```
laffp53:~ # nvme id-ctrl -H /dev/nvme1n1 | head -10
```

```
NVME Identify Controller:
```

```
vid      : 0x144d
```

```
ssvid    : 0x144d
```

```
sn       : S4EWNG0M507688D
```

```
mn       : Samsung SSD 970 EVO Plus 1TB
```

```
fr       : 1B2QEXM7
```

```
rab      : 2
```

```
ieee     : 002538
```

```
cmic     : 0
```

```
[3:3] : 0      ANA not supported
```

```
laffp53:~ # nvme id-ctrl -H /dev/nvme1n1 | grep -A 2 vwc
```

```
vwc      : 0x1
```

```
[0:0] : 0x1    Volatile Write Cache Present
```

```
laffp53:~ #
```

ECE deploy



```
[root@c8n1 ~]# mmvdisk vs define --vdisk-set vs100g4M --rg rgBeer --code 8+2P --block-size 4M --set-size 100g
mmvdisk: Vdisk set 'vs100g4M' has been defined.
mmvdisk: Recovery group 'rgBeer' has been defined in vdisk set 'vs100g4M'.
```

		member vdisks					
vdisk set	count	size	raw size	created	file system and attributes		
vs100g4M	10	9 GiB	12 GiB	no	-, DA1, 8+2p, 4 MiB, dataAndMetadata, system		

		declustered		capacity			all vdisk sets defined
recovery group	array	type	total raw	free raw	free%	in the declustered array	
rgBeer	DA1	HDD	518 GiB	390 GiB	75%	vs100g4M	

		vdisk set map memory per server		
node class	available	required	required per vdisk set	
beer	4096 MiB	3585 MiB	vs100g4M (768 KiB)	

```
[root@c8n1 ~]# mmvdisk vs create
```

Command

mmvdisk vs define --vdisk-set vs100g4M --rg rgBeerBeer --code 8+2p --block-size 4M --set-size 100g
mmvdisk vs define --vdisk-set vs100g4M --rg rgBeer --code 8+2P --block-size 4M --set-size 100g

ECE deploy

```
[root@c8n1 ~]# mmvdisk vs create --vdisk-set vs100g4M
mmvdisk: 10 vdisks and 10 NSDs will be created in vdisk set 'vs100g4M'.
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG001VS001
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG002VS001
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG003VS001
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG004VS001
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG005VS001
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG006VS001
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG007VS001
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG008VS001
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG009VS001
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG001LG010VS001
mmvdisk: Created all vdisks in vdisk set 'vs100g4M'.
mmvdisk: (mmcrnsd) Processing disk RG001LG001VS001
mmvdisk: (mmcrnsd) Processing disk RG001LG002VS001
mmvdisk: (mmcrnsd) Processing disk RG001LG003VS001
mmvdisk: (mmcrnsd) Processing disk RG001LG004VS001
mmvdisk: (mmcrnsd) Processing disk RG001LG005VS001
mmvdisk: (mmcrnsd) Processing disk RG001LG006VS001
mmvdisk: (mmcrnsd) Processing disk RG001LG007VS001
mmvdisk: (mmcrnsd) Processing disk RG001LG008VS001
mmvdisk: (mmcrnsd) Processing disk RG001LG009VS001
mmvdisk: (mmcrnsd) Processing disk RG001LG010VS001
mmvdisk: Created all NSDs in vdisk set 'vs100g4M'.
[root@c8n1 ~]#
```

Command

```
mmvdisk vs create --vdisk-set vs100g4M
```


ECE deploy

```
[root@c8n1 ~]# mmvdisk fs create --file-system beer --vdisk-set vs100g4M --mmcrfs -T /gpfs/beer
mmvdisk: Creating file system 'beer'.
mmvdisk: The following disks of beer will be formatted on node c8n1:
mmvdisk:      RG001LG001VS001: size 10204 MB
mmvdisk:      RG001LG002VS001: size 10204 MB
mmvdisk:      RG001LG003VS001: size 10204 MB
mmvdisk:      RG001LG004VS001: size 10204 MB
mmvdisk:      RG001LG005VS001: size 10204 MB
mmvdisk:      RG001LG006VS001: size 10204 MB
mmvdisk:      RG001LG007VS001: size 10204 MB
mmvdisk:      RG001LG008VS001: size 10204 MB
mmvdisk:      RG001LG009VS001: size 10204 MB
mmvdisk:      RG001LG010VS001: size 10204 MB
mmvdisk: Formatting file system ...
mmvdisk: Disks up to size 153.24 GB can be added to storage pool system.
mmvdisk: Creating Inode File
mmvdisk:   78 % complete on Fri Feb 28 09:52:41 2020
mmvdisk:  100 % complete on Fri Feb 28 09:52:42 2020
mmvdisk: Creating Allocation Maps
mmvdisk: Creating Log Files
mmvdisk:    3 % complete on Fri Feb 28 09:52:59 2020
mmvdisk:  100 % complete on Fri Feb 28 09:53:01 2020
mmvdisk: Clearing Inode Allocation Map
mmvdisk: Clearing Block Allocation Map
mmvdisk: Formatting Allocation Map for storage pool system
mmvdisk: Completed creation of file system /dev/beer.
[root@c8n1 ~]#
```

Command

```
mmvdisk fs create --file-system beer --vdisk-set vs100g4M --mmcrfs -T /gpfs/beer
```

- add storage / nodes to ECE
- who serves my diskXYZ ?
- delete a node (storage) from ECE

ECE manage - find my disk



```
[root@c8n1 beer]# mmlsdisk beer
disk      driver  sector  failure holds  holds  status  availability  storage
name      type    size    group metadata data      availability pool
-----
RG001LG001VS001 nsd      512      1 yes    yes    ready    up    system
RG001LG002VS001 nsd      512      2 yes    yes    ready    up    system
RG001LG003VS001 nsd      512      1 yes    yes    ready    up    system
RG001LG004VS001 nsd      512      2 yes    yes    ready    up    system
RG001LG005VS001 nsd      512      1 yes    yes    ready    up    system
RG001LG006VS001 nsd      512      2 yes    yes    ready    up    system
RG001LG007VS001 nsd      512      1 yes    yes    ready    up    system
RG001LG008VS001 nsd      512      2 yes    yes    ready    up    system
RG001LG009VS001 nsd      512      1 yes    yes    ready    up    system
RG001LG010VS001 nsd      512      2 yes    yes    ready    up    system
[root@c8n1 beer]#
```

mmlsdisk <dev>

```
[root@c8n1 beer]# mmlsnsd
File system  Disk name  NSD servers
-----
beer         RG001LG001VS001 vdisk server
beer         RG001LG002VS001 vdisk server
beer         RG001LG003VS001 vdisk server
beer         RG001LG004VS001 vdisk server
beer         RG001LG005VS001 vdisk server
beer         RG001LG006VS001 vdisk server
beer         RG001LG007VS001 vdisk server
beer         RG001LG008VS001 vdisk server
beer         RG001LG009VS001 vdisk server
beer         RG001LG010VS001 vdisk server
```

mmlsnsd <dev>

ECE manage - find my disk



```
[root@c8n1 beer]# mmvdisk rg list --rg rgBeer --server
```

node number	server	active	remarks
1	c8n1	yes	serving rgBeer: LG005, LG010
2	c8n2	yes	serving rgBeer: root, LG001, LG006
3	c8n3	yes	serving rgBeer: LG002, LG007
4	c8n4	yes	serving rgBeer: LG003, LG008
5	c8n5	yes	serving rgBeer: LG004, LG009

```
[root@c8n1 beer]#
```

mmvdisk rg list -rg RG --server

(2) – shows the node

(1) – find log group

```
[root@c8n1 beer]# mmvdisk vdisk list --vs all | grep LG00[2,7]
```

vdisk	vs	server	rg	size
RG001LG002VS001	vs100g4M	beer	rgBeer	DA1, 4+2p, 4 MiB
RG001LG007VS001	vs100g4M	beer	rgBeer	DA1, 4+2p, 4 MiB

```
[root@c8n1 beer]#
```

mmvdisk vdisk list -vs all

ECE manage - shrink / delete a node

Node	Daemon node name	IP address	Admin node name	Designation
1	c8n1	10.0.11.11	c8n1	quorum-manager
2	c8n2	10.0.11.12	c8n2	quorum
3	c8n3	10.0.11.13	c8n3	quorum
4	c8n4	10.0.11.14	c8n4	
5	c8n5	10.0.11.15	c8n5	

```
[root@c8n1 ~]# mmvdisk rg delete --recovery-group rgBeer -N c8n5
```

```
mmvdisk:
```

```
mmvdisk: The file system 'beer' requires all servers from recovery group 'rgBeer'.
```

```
mmvdisk: Remove the nodes from the file system first with the command:
```

```
mmvdisk: mmvdisk filesystem delete --file-system beer --vdisk-set vs100g4M --recovery-group rgBeer -N c8n5
```

```
mmvdisk: Command failed. Examine previous error messages to determine cause.
```

```
[root@c8n1 ~]#
```

Command

```
mmvdisk rg delete --recovery-group rgBeer -N c8n5
```

```
[root@c8n1 ~]# mmvdisk filesystem delete --file-system beer --vdisk-set vs100g4M --recovery-group rgBeer -N c8n5
```

```
mmvdisk: This will run the GPFS mmdeldisk command on file system 'beer'
```

```
mmvdisk: which may take hours to complete and should not be interrupted.
```

```
mmvdisk: Do you wish to continue (yes or no)? yes
```

```
mmvdisk: Reducing recovery group 'rgBeer' of file system 'beer', vdisk set 'vs100g4M'.
```

```
mmvdisk: Deleting disks ...
```

```
mmvdisk: Scanning file system metadata, phase 1 ...
```

```
mmvdisk: 100 % complete on Sat Feb 29 03:04:15 2020
```

```
mmvdisk: Scan completed successfully.
```

```
mmvdisk: Scanning file system metadata, phase 2 ...
```

```
mmvdisk: 100 % complete on Sat Feb 29 03:04:15 2020
```

```
mmvdisk: Scan completed successfully.
```

```
mmvdisk: Scanning file system metadata, phase 3 ...
```

```
mmvdisk: Scan completed successfully.
```

```
mmvdisk: Scanning file system metadata, phase 4 ...
```

```
mmvdisk: 100 % complete on Sat Feb 29 03:04:15 2020
```

```
mmvdisk: Scan completed successfully.
```

ECE manage

```
[root@c8n1 ~]# mmvdisk filesystem delete --file-system beer --vdisk-set vs100g4M --recovery-group rgBeer -N c8n5

mmvdisk: This will run the GPFS mmdeldisk command on file system 'beer'
mmvdisk: which may take hours to complete and should not be interrupted.

mmvdisk: Do you wish to continue (yes or no)? yes

mmvdisk: Reducing recovery group 'rgBeer' of file system 'beer', vdisk set 'vs100g4M'.
mmvdisk: Deleting disks ...
mmvdisk: Scanning file system metadata, phase 1 ...
mmvdisk: 100 % complete on Sat Feb 29 03:04:15 2020
mmvdisk: Scan completed successfully.
mmvdisk: Scanning file system metadata, phase 2 ...
mmvdisk: 100 % complete on Sat Feb 29 03:04:15 2020
mmvdisk: Scan completed successfully.
mmvdisk: Scanning file system metadata, phase 3 ...
mmvdisk: Scan completed successfully.
mmvdisk: Scanning file system metadata, phase 4 ...
mmvdisk: 100 % complete on Sat Feb 29 03:04:15 2020
mmvdisk: Scan completed successfully.
mmvdisk: Scanning file system metadata, phase 5 ...
mmvdisk: 100 % complete on Sat Feb 29 03:04:15 2020
mmvdisk: Scan completed successfully.
mmvdisk: Scanning user file metadata ...
mmvdisk: 100.00 % complete on Sat Feb 29 03:04:18 2020 ( 102400 inodes with total 1660 MB data processed)
mmvdisk: Scan completed successfully.
mmvdisk: Checking Allocation Map for storage pool system
mmvdisk: tsdeldisk completed.

mmvdisk: Attention: There are incomplete vdisk set or file system changes.
mmvdisk: Members of vdisk set 'vs100g4M' are orphaned from file system 'beer'.
mmvdisk: Complete any vdisk set or file system changes to dismiss this notice.
[root@c8n1 ~]# █
```

Command

```
mmvdisk fs delete --file-system beer --vdisk-set vs100g4M --recovery-group rgBeer -N c8n5
```

ECE manage

```
[root@c8n1 ~]# mmvdisk vdisk list --vdisk-set all
```

vdisk	vdisk set	file system	recovery group	declustered array, RAID code, block size	remarks
RG001LG001VS001	vs100g4M	beer	rgBeer	DA1, 8+2p, 4 MiB	
RG001LG002VS001	vs100g4M	beer	rgBeer	DA1, 8+2p, 4 MiB	
RG001LG003VS001	vs100g4M	beer	rgBeer	DA1, 8+2p, 4 MiB	
RG001LG004VS001	vs100g4M	beer	rgBeer	DA1, 8+2p, 4 MiB	
RG001LG005VS001	vs100g4M	beer	rgBeer	DA1, 8+2p, 4 MiB	
RG001LG006VS001	vs100g4M	beer	rgBeer	DA1, 8+2p, 4 MiB	
RG001LG007VS001	vs100g4M	beer	rgBeer	DA1, 8+2p, 4 MiB	
RG001LG008VS001	vs100g4M	beer	rgBeer	DA1, 8+2p, 4 MiB	
RG001LG009VS001	vs100g4M	-	rgBeer	DA1, 8+2p, 4 MiB	orphan
RG001LG010VS001	vs100g4M	-	rgBeer	DA1, 8+2p, 4 MiB	orphan

```
[root@c8n1 ~]#
```

```
[root@c8n1 ~]# mmvdisk rg delete --rg rgBeer -N c8n5
```

```
mmvdisk: This command will delete 'c8n5' and its storage  
mmvdisk: and server capacity from recovery group 'rgBeer'.
```

```
mmvdisk: Do you wish to continue (yes or no)? yes
```

```
mmvdisk: 2 vdisks and 2 NSDs will be deleted in vdisk set 'vs100g4M'.
```

```
mmvdisk: Deleted all node 'c8n5' NSDs in vdisk set 'vs100g4M'.
```

```
mmvdisk: (mmdelvdisk) [I] Processing vdisk RG001LG009VS001
```

```
mmvdisk: (mmdelvdisk) [I] Processing vdisk RG001LG010VS001
```

```
mmvdisk: Deleted all node 'c8n5' vdisks in vdisk set 'vs100g4M'.
```

```
mmvdisk: (mmdelvdisk) [I] Processing vdisk RG001LG001VS001
```

```
mmvdisk: (mmdelvdisk) [I] Processing vdisk RG001LG002VS001
```

```
mmvdisk: Updating parameters for declustered array
```

```
mmvdisk: Updating pdisk list for recovery group
```

```
mmvdisk: This could take a long time.
```

```
mmvdisk: Removing node 'c8n5' from node class 'beer'.
```

Command

```
mmvdisk rg delete --rg rgBeer -N c8n5
```

ECE manage

```
[root@c8n1 ~]# mmvdisk rg list --rg rgBeer --server
```

node number	server	active	remarks
1	c8n1	yes	serving rgBeer: LG004, LG005
2	c8n2	yes	serving rgBeer: root, LG001, LG006
3	c8n3	yes	serving rgBeer: LG002, LG007
4	c8n4	yes	serving rgBeer: LG003, LG008

```
[root@c8n1 ~]# █
```

Command

```
mmvdisk rg delete --rg rgBeer -N c8n5
```

Network ..
network ...

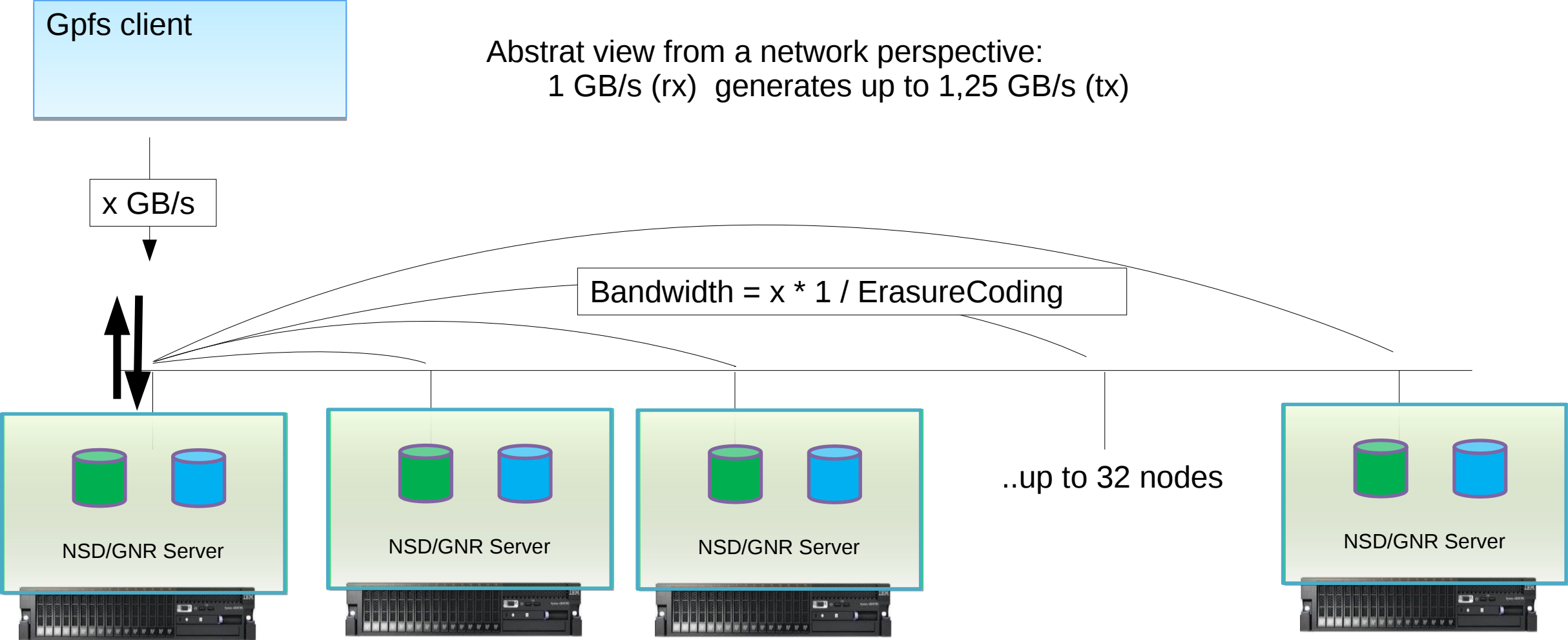
ECE, it's all about the network – consider ...

- Spectrum Scale ECE is highly network dependent
- NSD servers receive a request (ex. Write), and will need to send the write data and parity data to pdisks on other nodes
- Latency on the network plays a large role in performance
 - A High speed, low latency storage network is essential
- Keep CES, AFM, TCT and other services on separate networks
- Ensure storage network (backend) is as fast as or faster than client network (frontend)
- Use the mmnetverify connectivity all option in the mmnetverify command in the IBM Spectrum Scale: Command and Programming Reference to ensure that your network is configured for use by IBM Spectrum Scale

ECE performance – SNC architecture

Write full block: ~ 25% overhead for 8+2p

Abstrat view from a network perspective:
1 GB/s (rx) generates up to 1,25 GB/s (tx)



ECE / SNC architecture - tune your network



Quick check:

```
[root@fscs-sr650-13 beer]# /usr/lpp/mmfs/samples/perf/gpfsperf read seq /gpfs/beer/myFile20G-5 -n 20g -r 4M -th 16
/usr/lpp/mmfs/samples/perf/gpfsperf read seq /gpfs/beer/myFile20G-5
recSize 4M nBytes 20G fileSize 20G
nProcesses 1 nThreadsPerProcess 16
file cache flushed before test
not using direct I/O
offsets accessed will cycle through the same file segment
not using shared memory buffer
not releasing byte-range token after open
```

Data rate was 1759119.21 Kbytes/sec, Op Rate was 419.41 Ops/sec, Avg Latency was 31.450 milliseconds, thread utilization 0.8

```
[root@fscs-sr650-13 beer]#
```

```
[root@fscs-sr650-13 beer]# mmdiag --network | grep -A 1 devicename
devicename      speed      mtu      duplex      rx_dropped  rx_errors  tx_dropped  tx_errors
enp134s0f0      50000      1500     full        75434       223        0           0
```

nominal bandwidth

Ouch !!

ECE / SNC architecture - tune your network



- ibdev2netdev
- ib_read_bw

```
[root@fscc-sr650-13 beer]# ibdev2netdev
mlx5_0 port 1 ==> enp134s0f0 (Up)
mlx5_1 port 1 ==> enp134s0f1 (Down)
[root@fscc-sr650-13 beer]# ib_send_bw -d mlx5_0 -i 1
```

* Waiting for client to connect... *

Send BW Test			
Dual-port	: OFF	Device	: mlx5_0
Number of qps	: 1	Transport type	: IB
Connection type	: RC	Using SRQ	: OFF
RX depth	: 512		
CQ Moderation	: 100		
Mtu	: 1024[B]		
Link type	: Ethernet		
GID index	: 5		
Max inline data	: 0[B]		
rdma_cm QPs	: OFF		
Data ex. method	: Ethernet		

local address: LID 0000 QPN 0x00d5 PSN 0x6ebcf4
GID: 00:00:00:00:00:00:00:00:00:00:255:255:10:00:12:13
remote address: LID 0000 QPN 0x00d6 PSN 0x9059e6
GID: 00:00:00:00:00:00:00:00:00:00:255:255:10:00:12:14

#bytes	#iterations	BW peak[MB/sec]	BW average[MB/sec]	MsgRate[Mpps]
65536	1000	0.00	5476.26	0.087620

```
[root@fscc-sr650-13 beer]#
```

```
[root@fscc-sr650-14 ~]# ib_send_bw -d mlx5_0 -i 1 ece_13.localnet.com
```

Send BW Test			
Dual-port	: OFF	Device	: mlx5_0
Number of qps	: 1	Transport type	: IB
Connection type	: RC	Using SRQ	: OFF
TX depth	: 128		
CQ Moderation	: 100		
Mtu	: 1024[B]		
Link type	: Ethernet		
GID index	: 4		
Max inline data	: 0[B]		
rdma_cm QPs	: OFF		
Data ex. method	: Ethernet		

local address: LID 0000 QPN 0x00d6 PSN 0x9059e6
GID: 00:00:00:00:00:00:00:00:00:00:255:255:10:00:12:14
remote address: LID 0000 QPN 0x00d5 PSN 0x6ebcf4
GID: 00:00:00:00:00:00:00:00:00:00:255:255:10:00:12:13

#bytes	#iterations	BW peak[MB/sec]	BW average[MB/sec]	MsgRate[Mpps]
65536	1000	5461.50	5461.36	0.087382

```
[root@fscc-sr650-14 ~]#
```

ECE / SNC architecture - tune your network

```
[root@fscc-sr650-13 beer]# /usr/lpp/mmfs/samples/perf/gpfsperf read seq /gpfs/beer/myFile20G-5 -n 20g -r 4M -th 16 | grep "rate"
Data rate was 2510987.80 Kbytes/sec, Op Rate was 598.67 Ops/sec, Avg Latency was 22.746 milliseconds, thread utilization 0.
```

```
[root@fscc-sr650-13 beer]# mmdiag --network | grep -B 1 enp134s0f0 | tail -2
```

devicename	speed	mtu	duplex	rx_dropped	rx_errors	tx_dropped	tx_errors
enp134s0f0	50000	1500	full	273815	223	0	0

```
[root@fscc-sr650-13 beer]# /usr/lpp/mmfs/samples/perf/gpfsperf read seq /gpfs/beer/myFile20G-5 -n 20g -r 4M -th 16 | grep "rate"
Data rate was 2779710.47 Kbytes/sec, Op Rate was 662.73 Ops/sec, Avg Latency was 22.320 milliseconds, thread utilization 0.
```

```
[root@fscc-sr650-13 beer]# mmdiag --network | grep -B 1 enp134s0f0 | tail -2
```

devicename	speed	mtu	duplex	rx_dropped	rx_errors	tx_dropped	tx_errors
enp134s0f0	50000	1500	full	281227	223	0	0

```
[root@fscc-sr650-13 beer]#
```

ECE / SNC architecture - tune your network



```
[root@fscc-sr650-13 beer]# /usr/lpp/mmfs/samples/perf/gpfsperf read seq /gpfs/beer/myFile20G-5 -n 20g -r 4M -th 16 | grep "rate"
Data rate was 2510987.80 Kbytes/sec, Op Rate was 598.67 Ops/sec, Avg Latency was 22.746 milliseconds, thread utilization 0.
[root@fscc-sr650-13 beer]# mmdiag --network | grep -B 1 enp134s0f0 | tail -2
devicename      speed      mtu        duplex    rx_dropped rx_errors tx_dropped tx_errors
enp134s0f0      50000     1500      full      273815     223       0         0
[root@fscc-sr650-13 beer]# /usr/lpp/mmfs/samples/perf/gpfsperf read seq /gpfs/beer/myFile20G-5 -n 20g -r 4M -th 16 | grep "rate"
Data rate was 2779710.47 Kbytes/sec, Op Rate was 662.73 Ops/sec, Avg Latency was 22.320 milliseconds, thread utilization 0.
[root@fscc-sr650-13 beer]# mmdiag --network | grep -B 1 enp134s0f0 | tail -2
devicename      speed      mtu        duplex    rx_dropped rx_errors tx_dropped tx_errors
enp134s0f0      50000     1500      full      281227     223       0         0
[root@fscc-sr650-13 beer]#
```

```
[root@fscc-sr650-13 beer]# mmdsh -N all "ethtool -G enp134s0f0 tx 8192"
```

```
[root@fscc-sr650-13 beer]# mmdsh -N all "ethtool -G enp134s0f0 rx 8192"
```

ECE / SNC architecture - tune your network



```
[root@fscc-sr650-13 beer]# /usr/lpp/mmfs/samples/perf/gpfsperf read seq /gpfs/beer/myFile20G-5 -n 20g -r 4M -th 16 | grep "rate"
Data rate was 2510987.80 Kbytes/sec, Op Rate was 598.67 Ops/sec, Avg Latency was 22.746 milliseconds, thread utilization 0.
[root@fscc-sr650-13 beer]# mmdiag --network | grep -B 1 enp134s0f0 | tail -2
devicename      speed      mtu        duplex    rx_dropped rx_errors tx_dropped tx_errors
enp134s0f0      50000     1500      full      273815     223       0          0
[root@fscc-sr650-13 beer]# /usr/lpp/mmfs/samples/perf/gpfsperf read seq /gpfs/beer/myFile20G-5 -n 20g -r 4M -th 16 | grep "rate"
Data rate was 2779710.47 Kbytes/sec, Op Rate was 662.73 Ops/sec, Avg Latency was 22.320 milliseconds, thread utilization 0.
[root@fscc-sr650-13 beer]# mmdiag --network | grep -B 1 enp134s0f0 | tail -2
devicename      speed      mtu        duplex    rx_dropped rx_errors tx_dropped tx_errors
enp134s0f0      50000     1500      full      281227     223       0          0
[root@fscc-sr650-13 beer]#
```

```
[root@fscc-sr650-13 beer]# mmdsh -N all "ethtool -G enp134s0f0 tx 8192"
```

```
[root@fscc-sr650-13 beer]# mmdsh -N all "ethtool -G enp134s0f0 rx 8192"
```

```
[root@fscc-sr650-13 beer]# mmdsh -N all "ethtool -G enp134s0f0 tx 8192"
[root@fscc-sr650-13 beer]# mmdsh -N all "ethtool -G enp134s0f0 rx 8192"
[root@fscc-sr650-13 beer]#
[root@fscc-sr650-13 beer]# /usr/lpp/mmfs/samples/perf/gpfsperf read seq /gpfs/beer/myFile20G-5 -n 20g -r 4M -th 16 | grep "rate"
Data rate was 4211753.14 Kbytes/sec, Op Rate was 1004.16 Ops/sec, Avg Latency was 14.411 milliseconds, thread utilization 0.
[root@fscc-sr650-13 beer]# mmdiag --network | grep -B 1 enp134s0f0 | tail -2
devicename      speed      mtu        duplex    rx_dropped rx_errors tx_dropped tx_errors
enp134s0f0      50000     1500      full      281227     223       0          0
[root@fscc-sr650-13 beer]#
```

ECE / SNC architecture - udev rules



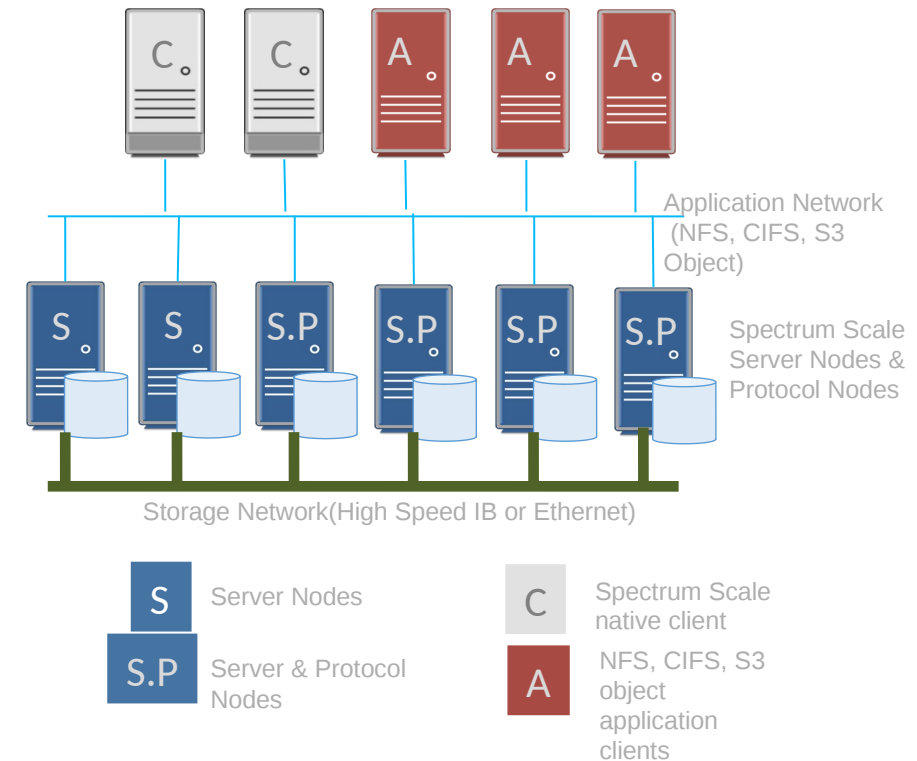
```
# tx queue length  
cat /etc/udev/rules.d/99-ibm-queue lenght.rules
```

```
KERNEL=="enP*", RUN+="/sbin/ip link set %k txqueuelen 10000" , RUN+="/sbin/ethtool -G %k rx 8192" , RUN+="/sbin/ethtool -G %k tx 8192"
```

ECE USE CASES

ECE Use Case: Dedicated High Performance file serving IBM Spectrum Scale

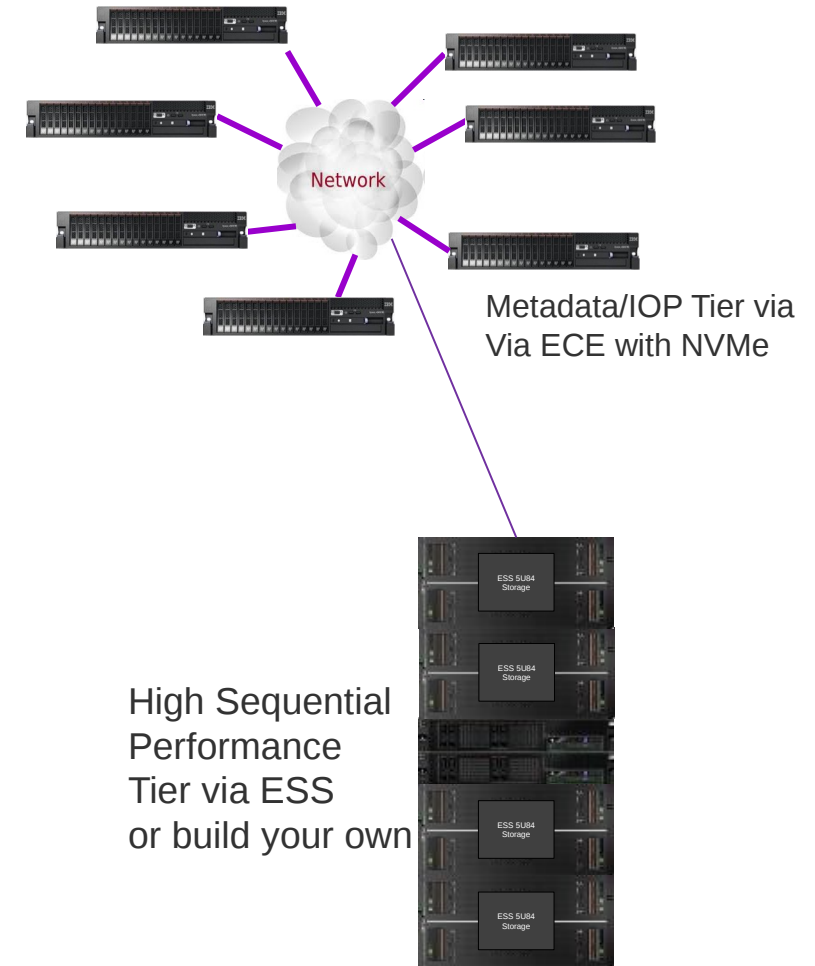
- Spectrum Scale Server high performance file services deployed on storage rich nodes communicating to native Spectrum Scale clients
- Deploy IBM Spectrum Scale Protocol services to allow customers to access ECE with NFS, SMB and Object.
- Dedicate High speed IB or Ethernet for NFS/SMB/storage communication
- Accelerate data processing by leveraging enterprise NVMe drives to deliver high throughput and low latency
- Each ECE storage server is typically configured with several NVMe drives to store and accelerate Spectrum Scale metadata and small data I/O, combined with a number of HDD drives to store user data.
- With the high performance design of ECE, it can deliver high performance file serving to the customer workloads.



ECE Use Case: High Performance Compute tier



- ECE's high performance erasure coding provides the capability of being a tier 1 storage device that can then tier to different storage medias (e.g. flash drives, spinning disks, tape, cloud storage, etc.) with different performance and cost characteristics.
- The policy based Information Life Cycle management feature makes it very convenient to manage data movement among different storage tiers.
- In this example, the ECE high performance compute tier is composed of NVMe drives to store and accelerate Spectrum Scale metadata and the set of hot data for high performance computing and analytics. The second tier can consist of NL SAS drives for lower \$/TB and fast sequential performance.



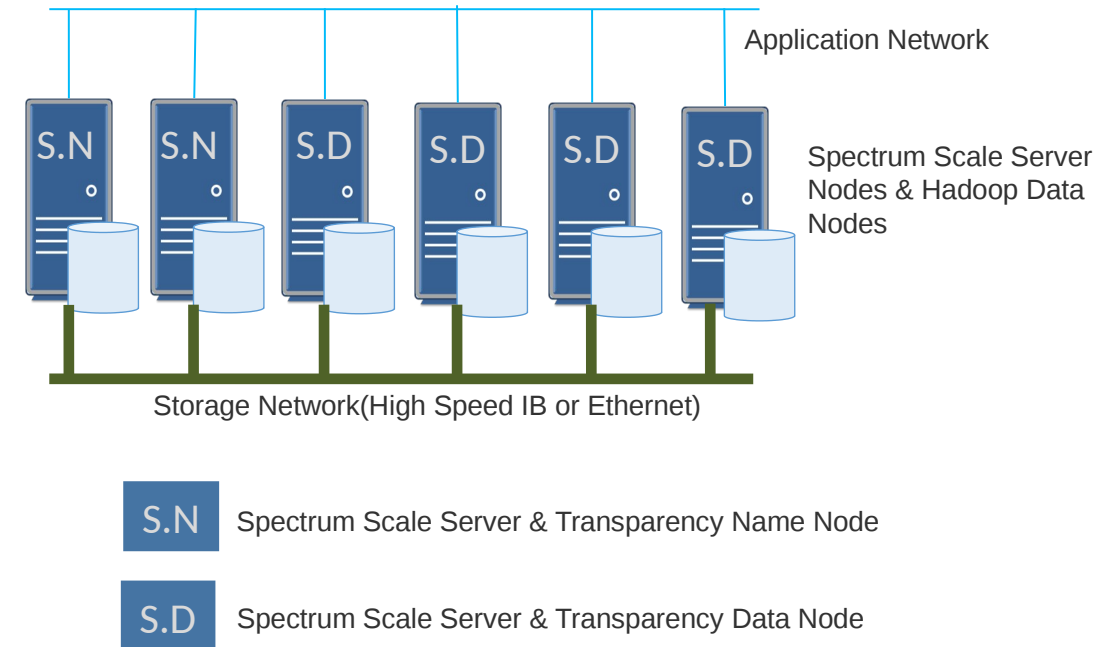
ECE Use Case: Analytics

Deployment Model:

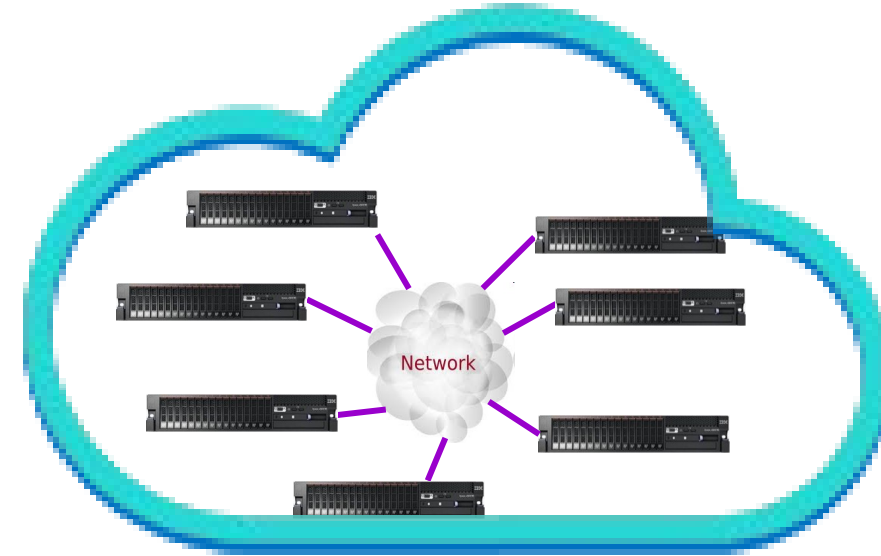
- Spectrum Scale Server and Transparency nodes (Name Node and Data Node) are deployed in storage rich server
- Dedicate High speed IB or Ethernet for storage communication (optional but highly recommended)

Use Case:

- Analytics workload based on HDP (or even Cloudera)
- Enterprise storage of HDFS alternative



- With space efficient erasure coding and extreme end-to-end data protection design and implementation, ECE can deliver the essential cost effective and data reliability value-adds to large scale cloud storage systems.
- The ECE storage system for high capacity cloud storage may be composed of an NVMe storage pool to store and accelerate GPFS metadata and small data I/O's, or all high capacity drives for lowest \$/TB
- An ECE storage system can also be low cost cloud storage connected to an on-site Spectrum Scale cluster



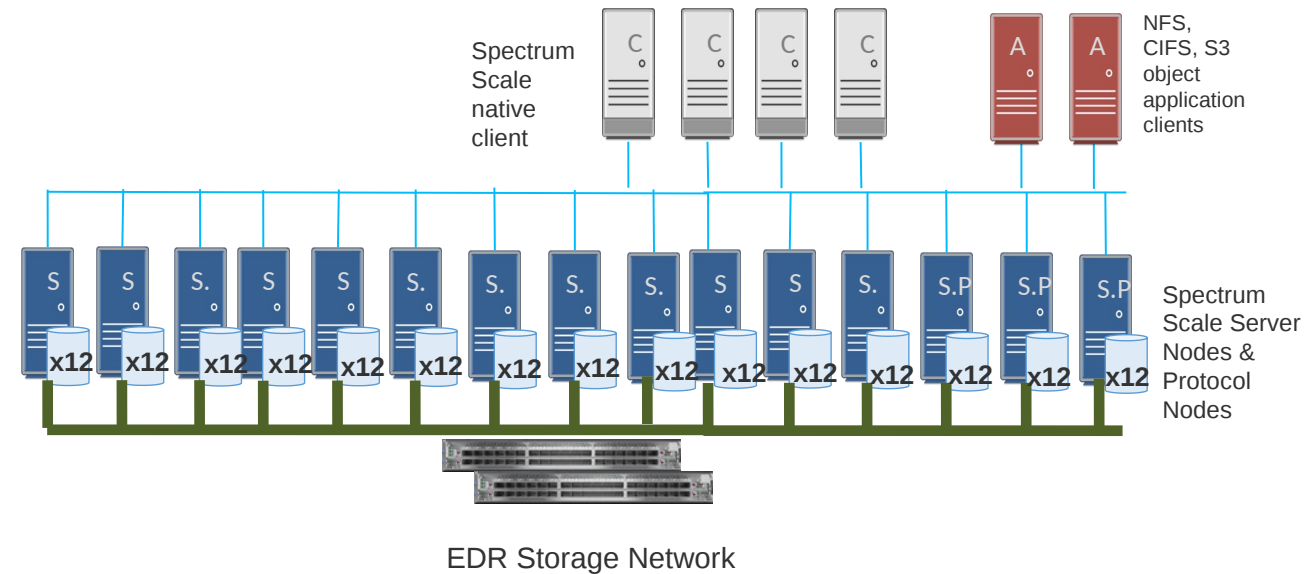
REQUIREMENT:

- 200TB Usable
- HPC -100GB/sec throughput
- SMB and NFS Access

DESIGN

- # Nodes needed – 15
- 12 drives per node
- 1.5TB drive size (e.g. Intel 4800)
- 2 x 100Gb/sec network cards per node
- Usable capacity per node – 35TB
- Usable Network BW per node – 10GB/sec

- Overall est. Read BW – 160GB/sec
- Overall est. Write BW – 135GB/sec

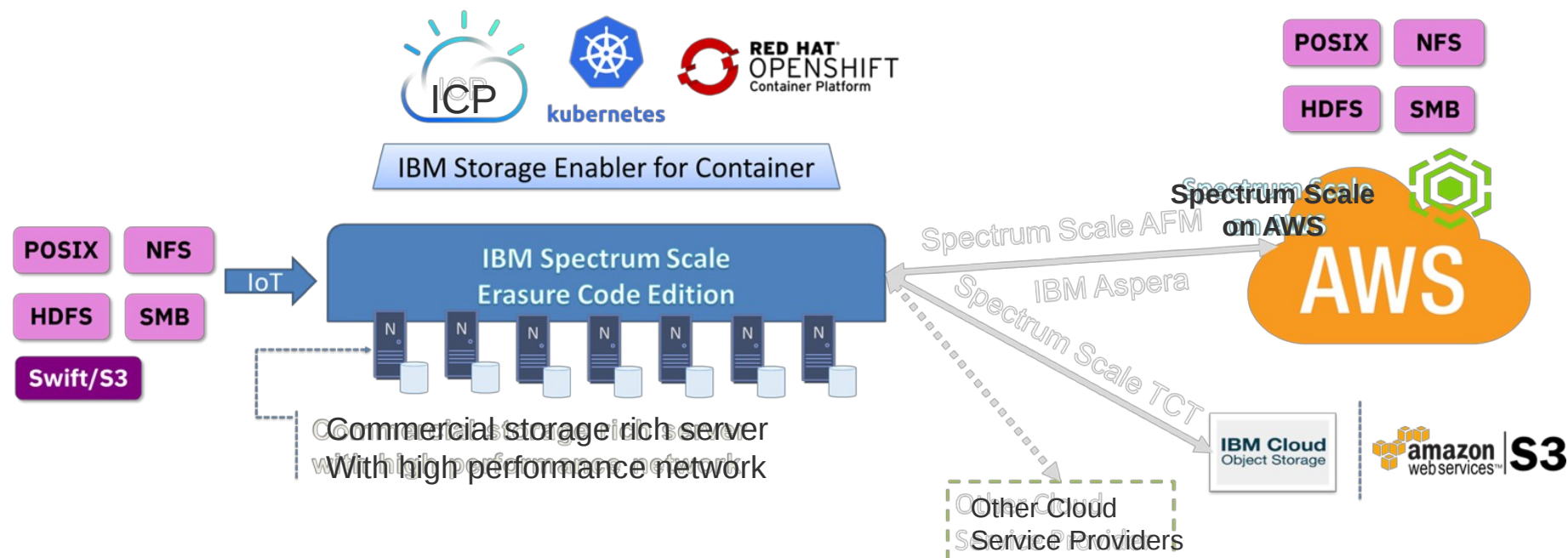


ECE Use Cases: Hybrid Multi-Cloud Storage including Containers



IBM Spectrum Scale

- ECE can provide a high performance on-prem Scale Out Filesystem and leverage containers and Kubernetes to support IBM Cloud Private, Red Hat OpenShift as well as leveraging AFM and TCT to a multitude of private/hybrid/public clouds
- Data comes from both data center and public cloud which need to be stored in a single name space to provide storage service for container
- IBM Spectrum Scale runs in both on-prem data center and AWS public cloud are connected by Spectrum Scale AFM
- Spectrum Scale with IBM Storage Enabler for Container providers storage service for container



Licensing & FAQs

- Spectrum Scale ECE is licensed by the usable TiB
 - usable capacity defined as the capacity presented to Linux, before applying erasure coding.
 - Thus the license pricing is independent of any choice of Error Correction width.
- Spectrum Scale ECE can also be licensed by the usable PiB with a discount
- ECE licenses ordered via Passport Advantage
 - The parts are Restricted initially pending review of a client's requirements and design

- **Can I buy ECE and use the licenses both with and without the erasure coding capability, i.e. both internal disk and SAN configurations?**
- Yes. An ECE license can be applied to ECE, DME or DAE. You will be compliant with licensing provided the total TBs deployed across all Editions does not exceed your ECE entitlement.
- **Will an ECE client be allowed (if the need arises in the future) to use its licenses within an IBM ESS appliance? If so, will the required capacity continue to be calculated as it is today with DAE/DME/Suite, i.e. as the net / RAIDed capacity reported by the ESS GUI?**
- Yes. ECE licenses will be treated the same as DME licenses.
- **Can ECE be offered within an ELA, provided the client meets OM's technical & business prerequisites?**
- Yes. Please contact Spectrum Scale offering management or Finance for approval of the Restricted part.
- **Can ECE be offered within an ESA, provided the client meets OM's technical & business prerequisites?**
- Yes. Note that inclusion of any edition of Scale within an ESA is subject to approval by Offering Management
- **Which options will IBM offer to trade-up existing Scale licensees to ECE from (a) DME/Suite, (b) DAE, (c) Advanced, (d) Standard?**
- Existing licenses for any Edition of Scale can be traded up. Trade-ups from DME or DAE will be "one TB for one TB" and based on the difference in price between the editions. Trade-ups from Advanced or Standard are similar to the existing process for trading up to DME
- **Are clients allowed to deploy ECE in the same Scale cluster as (a) DAE, (b) DME, (c) ESS DAE, (d) ESS DME? How are the current rules governing Multi-Clustering of different Scale Editions affected by the introduction of ECE?**
- The same rules apply to ECE as to DME. Different editions cannot exist in the same cluster. Multi-clustering is supported, with the same limitations as for DME. See the Knowledge Center.
- **Will the FPO feature in Scale DAE and DME continue to be supported, and if so for how long?**
- IBM does not currently plan to deprecate or remove this capability in a subsequent release of the product; it will remain supported and current with updates to the operating systems. Customers do not need to change any of their existing applications and scripts that use FPO at this time. They should not expect significant new functionality or enhancements to FPO.
- **Can ECE (a) be fully managed and accessed from containers through SEC/CSI, (b) be run itself in container mode?**
- ECE will support containerized workloads today and containerization in the future in the same way as other editions. See the Knowledge Center for details.
- **For more information, on Slack, join the #scale-ece channel in the IBM Storage Systems workspace**



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