



Lenovo HPC Storage Solutions Update



Spectrum Scale User Group, Singapore, 11-Mar-2019

Lenovo™

Michael Hennecke | HPC Chief Technologist

Lenovo DSS-G

Lenovo's Spectrum Scale RAID Solutions



+ Lenovo Distributed Storage Solution for IBM Spectrum Scale

The Hardware Components:

SR650 Servers

D1224 JBODs

D3284 JBODs

The Solution:

DSS-G2xx



Lenovo **LENOVO PRESS**

Lenovo ThinkSystem SR650 Server
Product Guide

Lenovo ThinkSystem SR650 is an Intel Xeon Scalable 2U rack server for small businesses up to large enterprises that need industry-leading reliability, management, and security, as well as maximizing performance and flexibility for future growth. The SR650 server is designed to handle a wide range of workloads, such as databases, virtualization and cloud computing, virtual desktop infrastructure (VDI), enterprise applications, collaboration, and business analytics and big data.

Featuring the Intel Xeon Processor Scalable Family, the SR650 server offers scalable performance, storage capacity, and I/O expansion. The SR650 server supports up to two processors, up to 1.5 TB (support for up to 3 TB is planned for future 2018 Intel Turbo Boost memory), up to 24x 2.5-inch or 12x 3.5-inch drive bays with an extensive choice of SATA HDDs, SAS/SATA HDDs, and SAS/SATA HDDs, and flexible I/O expansion options with the LSI HBA, the dedicated storage controller slot, and up to 16 PCIe slots.

The SR650 server offers basic or advanced hardware RAID protection and a wide range of networking options, including selectable LOM, ML2, and PCIe network adapters. The next-generation Lenovo XClarity Controller, which is built into the SR650 server, provides advanced server processor control, monitoring, and alerting functions.

The following figure shows the ThinkSystem SR650.



Figure 1. Lenovo ThinkSystem SR650

Did you know?

The SR650 server features a unique AnyBay design that allows a choice of drive interface types in the same drivebay: SAS drives, SATA drives, or L2700 SATA HDDs.

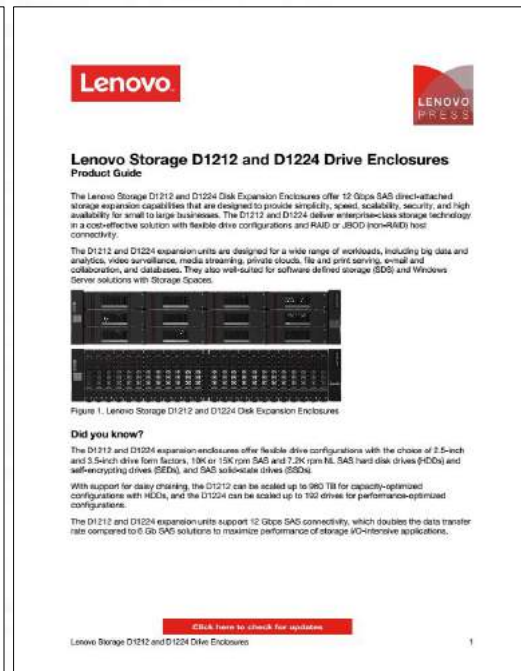
The SR650 server offers onboard NVMe PCIe ports that allow direct connections to the L2700 NVMe SSDs, which frees up I/O slots and helps lower NVMe solution acquisition costs.

The SR650 server offers impressive compute power per watt, featuring 80 PLUS Titanium and Platinum redundant power supplies that can deliver 90% (Titanium) or 94% (Platinum) efficiency at 90% load when connected to a 100V-240V AC power source.

The SR650 server is designed to meet ASHRAE A1 standards (up to 48 °C [113 °F]) in select configurations, which enable customers to lower energy costs, while still maintaining world-class reliability.

[Click here to check for updates](#)

Lenovo ThinkSystem SR650 Server 1



Lenovo **LENOVO PRESS**

Lenovo Storage D1212 and D1224 Drive Enclosures
Product Guide

The Lenovo Storage D1212 and D1224 Disk Expansion Enclosures offer 12 Gbps SAS direct-attached storage expansion capabilities that are designed to provide simplicity, speed, scalability, security, and high availability for small to large businesses. The D1212 and D1224 deliver enterprise-class storage technology in a cost-effective solution with flexible drive configurations and RAID or JBOD (non-RAID) host connectivity.

The D1212 and D1224 expansion units are designed for a wide range of workloads, including big data and analytics, video surveillance, media streaming, private clouds, file and print serving, email and collaboration, and databases. They are also well-suited for software-defined storage (SDS) and Windows Server solutions with Storage Spaces.




Figure 1. Lenovo Storage D1212 and D1224 Disk Expansion Enclosures

Did you know?

The D1212 and D1224 expansion enclosures offer flexible drive configurations with the choice of 2.5-inch and 3.5-inch drive form factors, 10K or 15K rpm SAS and 7.2K rpm NL SAS hard disk drives (HDDs) and self-encrypting drives (SEDs), and SAS hot-swappable drives (SSDs).

With support for data chaining, the D1212 can be scaled up to 960 TB for capacity-optimized configurations with HDDs, and the D1224 can be scaled up to 752 drives for performance-optimized configurations.

The D1212 and D1224 expansion units support 12 Gbps SAS connectivity, which doubles the data transfer rate compared to 6 Gb SAS solutions to maximize performance of storage I/O-intensive applications.

[Click here to check for updates](#)

Lenovo Storage D1212 and D1224 Drive Enclosures 1



Lenovo **LENOVO PRESS**

Lenovo Storage D3284 External High Density Drive Expansion Enclosure
Product Guide

The Lenovo Storage D3284 High Density Expansion Enclosure offers 12 Gbps SAS direct-attached storage expansion capabilities that are designed to provide density, speed, scalability, security, and high availability for medium to large businesses. The D3284 delivers enterprise-class storage technology in a cost-effective dense solution with flexible drive configurations of up to 84 drives in 5U rack space and RAID or JBOD (non-RAID) host connectivity.

The D3284 expansion unit is designed for a wide range of workloads, including big data and analytics, video surveillance, private and hybrid clouds, file and print serving, and backup and archiving. The D3284 is also well-suited for software-defined storage (SDS) and Windows Storage Spaces.



Figure 1. Lenovo Storage D3284 HD Expansion Enclosure

Did you know?

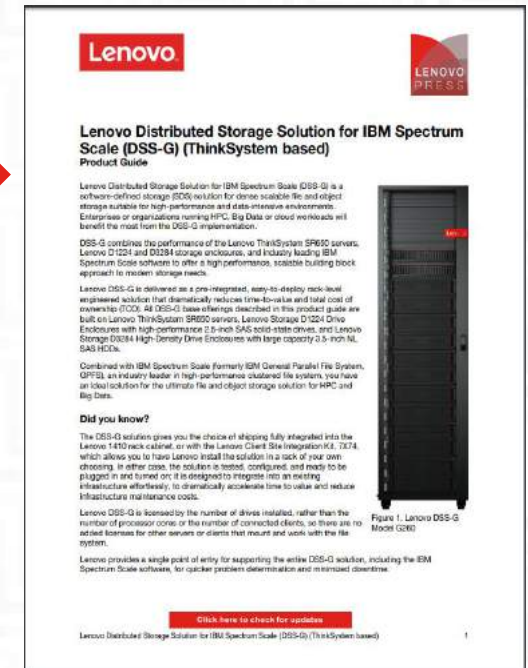
The D3284 expansion enclosures support 12 Gbps SAS connectivity, which doubles the data transfer rate compared to 6 Gb SAS solutions to maximize performance of storage I/O-intensive applications.

With support for data chaining, the D3284 expansion enclosures can be scaled up to 3.36 PB for capacity-optimized configurations.

The D3284 expansion enclosures allow data chaining with D1212 and D1224 expansion enclosures: Up to two D3284 and two D1212 or one D1224 drive enclosures is supported in a single chain.

[Click here to check for updates](#)

Lenovo Storage D3284 External High Density Drive Expansion Enclosure 1



Lenovo **LENOVO PRESS**

Lenovo Distributed Storage Solution for IBM Spectrum Scale (DSS-G) (ThinkSystem based)
Product Guide

Lenovo Distributed Storage Solution for IBM Spectrum Scale (DSS-G) is a software-defined storage (SDS) solution for dense scalable file and object storage suitable for high-performance and data-intensive environments. Enterprises or organizations running HPC, Big Data or cloud workloads will benefit the most from the DSS-G implementation.

DSS-G continues the performance of the Lenovo ThinkSystem SR650 servers, Lenovo D1212 and D1224 storage enclosures, and industry leading IBM Spectrum Scale software to offer a high performance, scalable building block approach to modern storage needs.

Lenovo DSS-G is delivered as a pre-integrated, easy-to-deploy rack-level engineered solution that dramatically reduces time-to-value and total cost of ownership (TCO). All DSS-G hardware elements described in this product guide are built on Lenovo ThinkSystem SR650 servers, Lenovo Storage D1212 Drive Enclosures with high-performance 2.5-inch SAS solid-state drives, and Lenovo Storage D3284 High Density Drive Enclosures with large capacity 2.5-inch NL SAS HDDs.

Combined with IBM Spectrum Scale Formerly IBM General Parallel File System, GPFS, an industry leader in high-performance clustered file system, you have an ideal solution for the ultimate file and object storage solution for HPC and Big Data.

Did you know?

The DSS-G solution gives you the choice of shipping fully integrated into the Lenovo 1400 rack cabinet, or with the Lenovo Client Storage System 14.7U, which allows you to have Lenovo install the solution in a rack of your own choosing. In either case, the solution is tested, configured, and ready to be plugged in and turned on; it is designed to integrate into an existing infrastructure effortlessly, to dramatically accelerate time to value and reduce infrastructure maintenance costs.

Lenovo DSS-G is licensed by the number of drives installed, rather than the number of processor cores or the number of connected clients, so there are no add-on licenses for other servers or clients that mount and write with the file system.

Lenovo provides a single point of entry for supporting the entire DSS-G solution, including the IBM Spectrum Scale software, for quicker problem determination and minimized downtime.




Figure 1. Lenovo DSS-G Model D260

[Click here to check for updates](#)

Lenovo Distributed Storage Solution for IBM Spectrum Scale (DSS-G) (ThinkSystem based) 1

<https://lenovopress.com/lp0644-lenovo-thinksystem-sr650-server>

<https://lenovopress.com/lp0512-lenovo-storage-d1212-d1224-drive-enclosures>

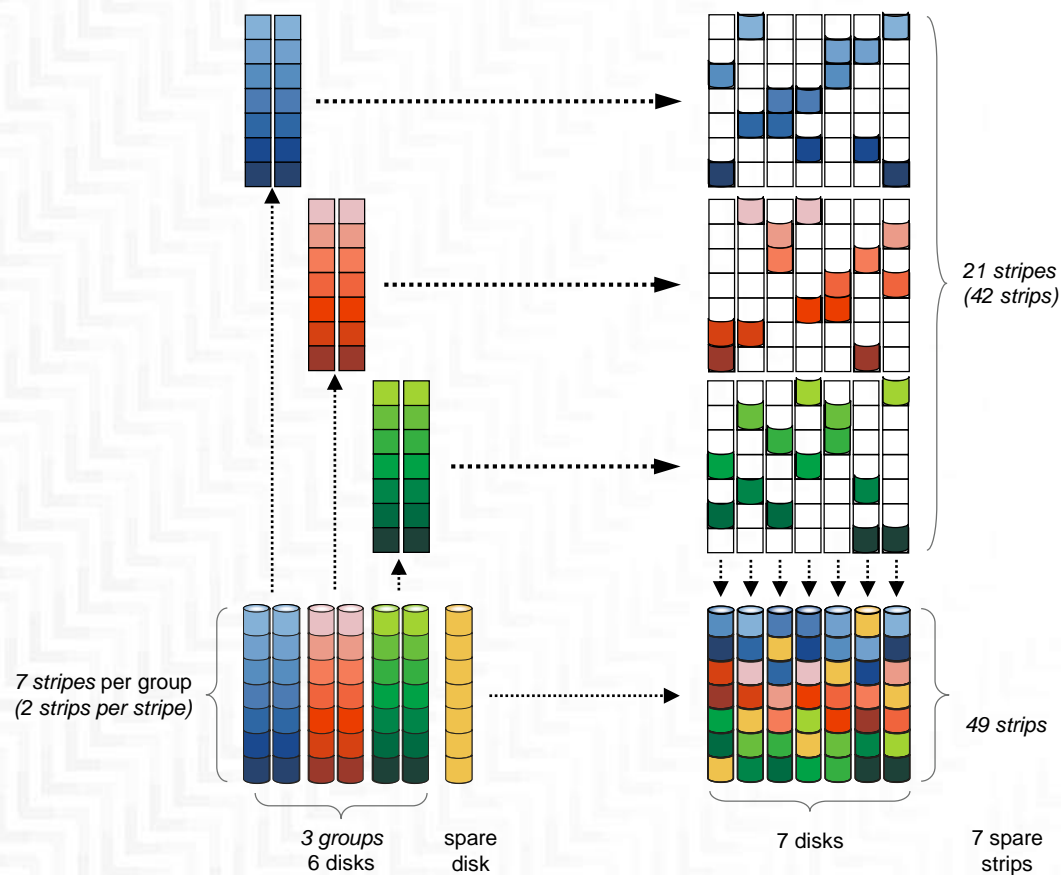
<https://lenovopress.com/lp0513-lenovo-storage-d3284-external-high-density-drive-expansion-enclosure>

<https://lenovopress.com/lp0837-lenovo-dss-g-thinksystem>

+ The Magic of Declustered RAID

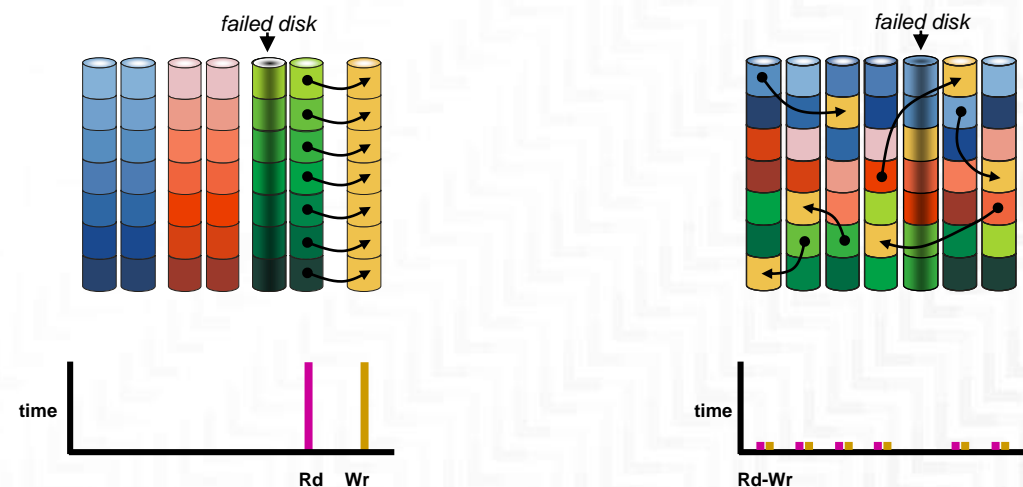
How does Declustered RAID work?

- Distributing Data and Parity information as well as Spare Capacity across all disks



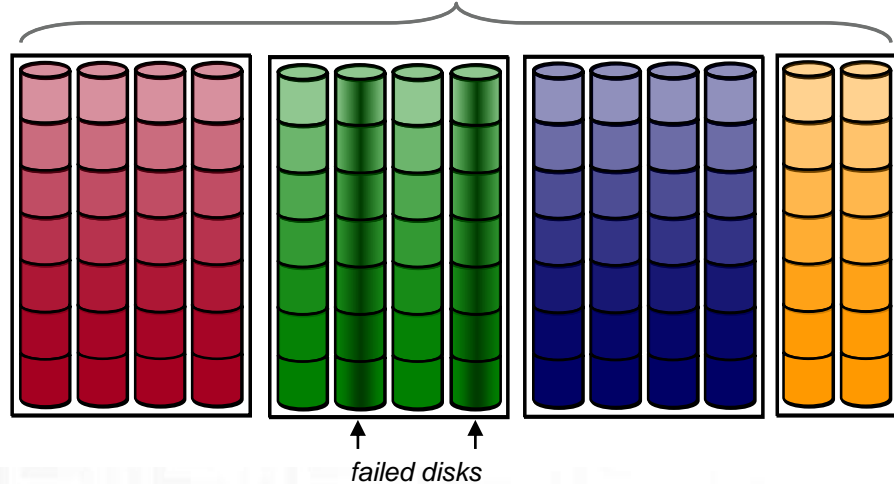
Rebuild with Declustered RAID1

- Traditional RAID would have one LUN (logical unit number) fully busy resulting in slow rebuild and high impact overall
- **Declustered RAID** rebuild activity spreads the load across many disks resulting in **faster rebuild** and **less disruption** to user programs
- **Declustered RAID minimizes** critical data exposed to **data loss** in case of a **second failure**.



+ Declustered RAID6 Rebuild Example – Two disk faults

14 physical disks / 3 traditional RAID6 arrays / 2 spares



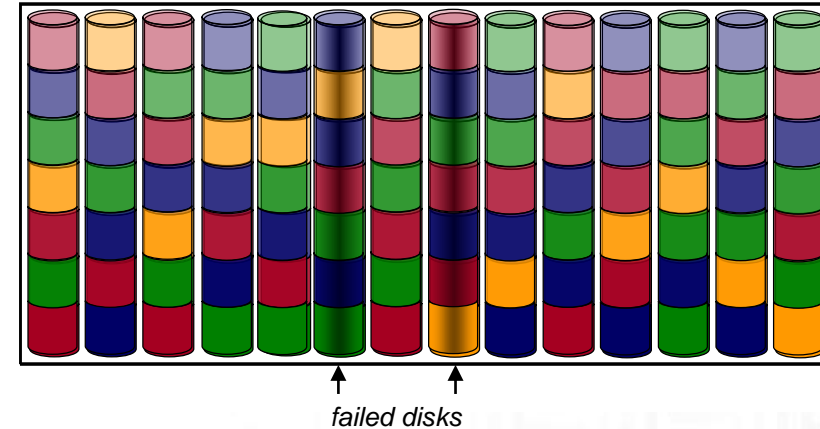
failed disks

Number of faults per stripe		
Red	Green	Blue
0	2	0
0	2	0
0	2	0
0	2	0
0	2	0
0	2	0
0	2	0

Number of stripes with 2 faults = 7

14 physical disks / 1 declustered RAID6 array / 2 spares

Decluster data, parity and spare



failed disks

Number of faults per stripe		
Red	Green	Blue
1	0	1
0	0	1
0	1	1
2	0	0
0	1	1
1	0	1
0	1	0

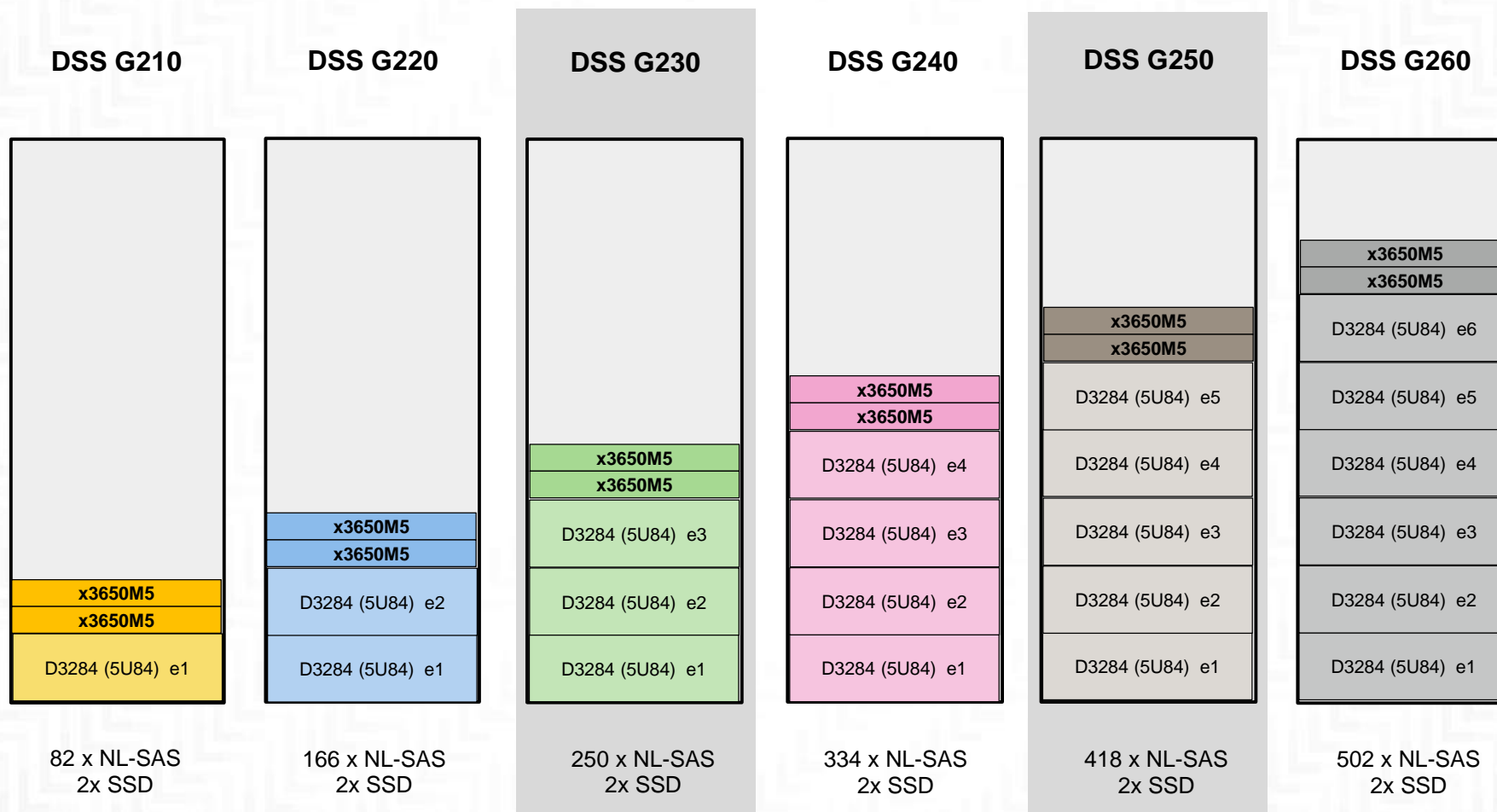
Number of stripes with 2 faults = 1

DSS-G220 8+3p → 3 faults in a 83-disk array: $(11/83) \cdot (10/82) \cdot (9/81) = 0.18\%$

DSS-G260 8+3p → 3 faults in a 251-disk array: $(11/251) \cdot (10/250) \cdot (9/249) = 0.006\%$

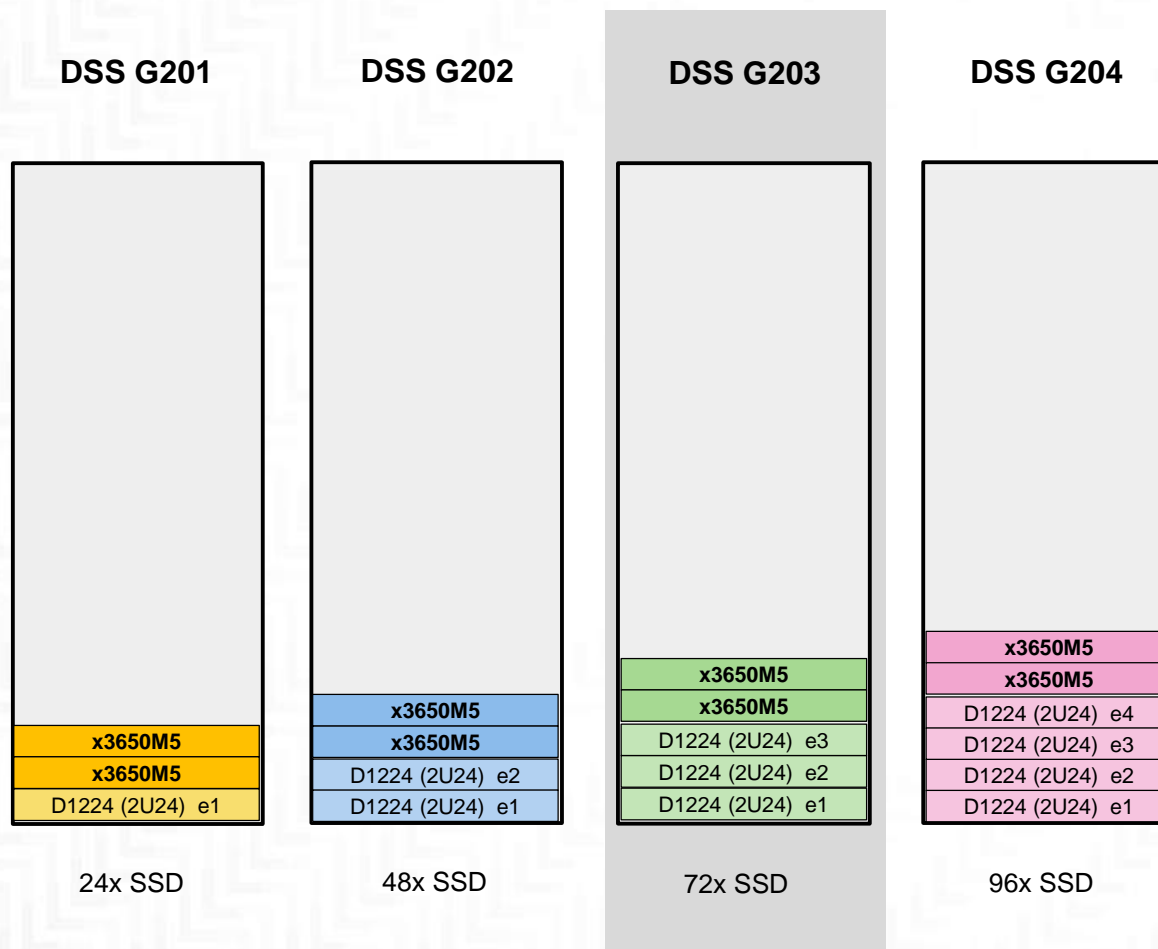
+ DSS for Spectrum Scale RAID building blocks

DSS-G2x0 „NL-SAS“ models with Lenovo D3284 JBODs

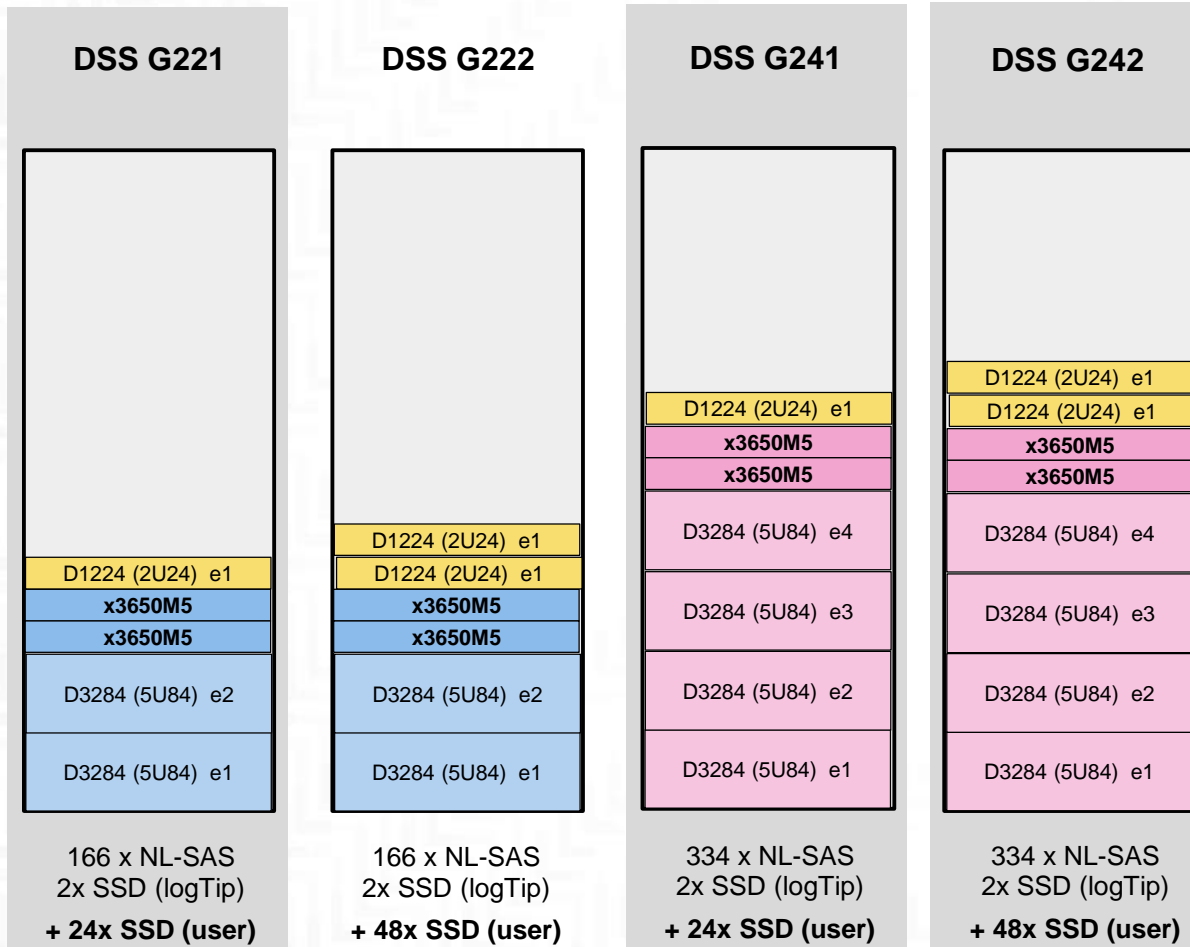


+ DSS for Spectrum Scale RAID building blocks

DSS-G20x „SSD“ models with Lenovo D1224 JBODs



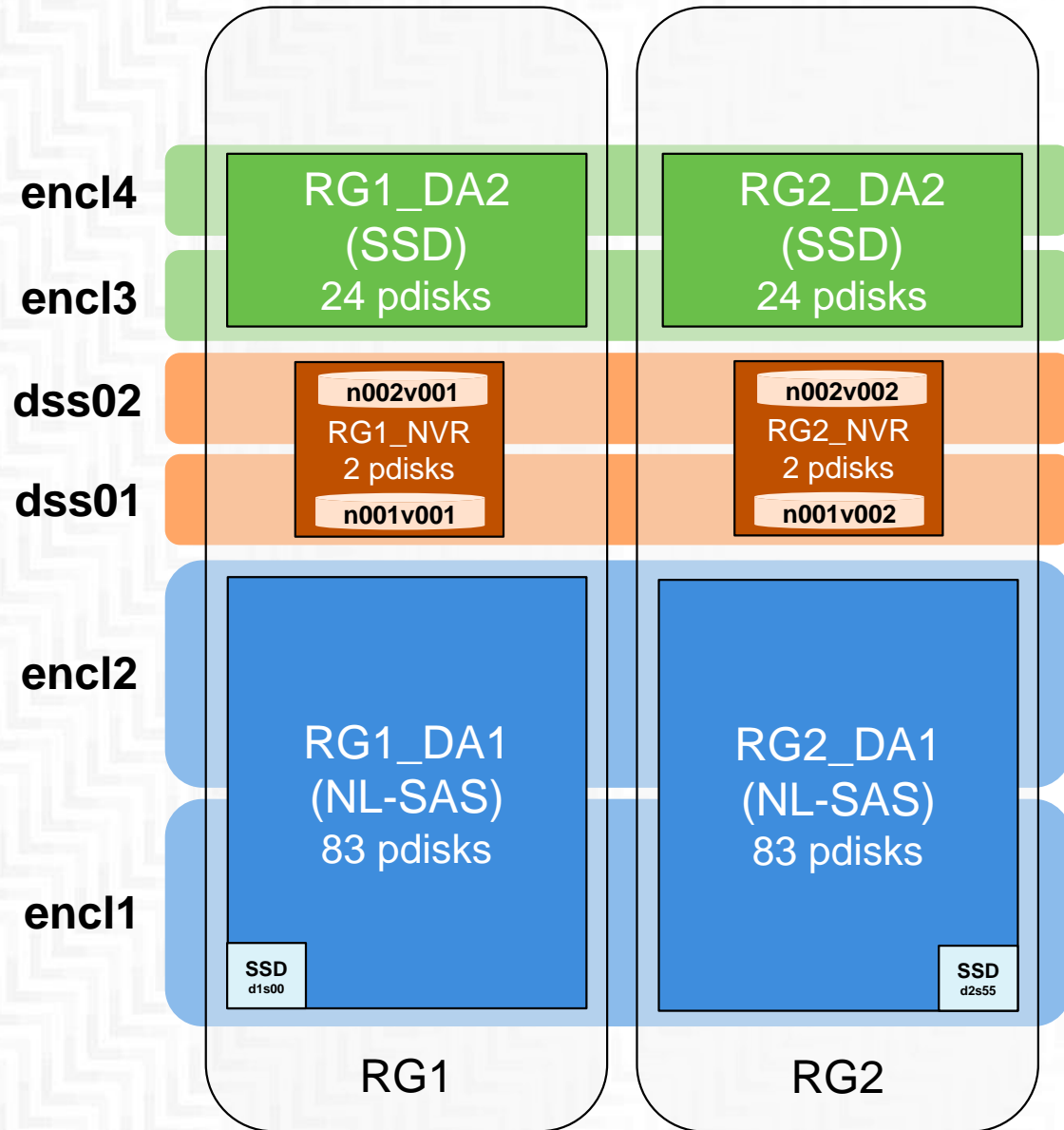
DSS-G2xy „hybrid“ models with NL-SAS JBODs and SSD JBODs



2019 Lenovo. All rights reserved.

Greyed-out items are not yet generally available in DSS-G 2.2, but can be supported as special bids.

+ The DSS-G222 „Hybrid“ Model – GA in DSS-G 2.2 (Nov/2018)



- Two separate Distributed Arrays for NL-SAS and for SSD
 - logTip still on SSD in the NL-SAS encl1
- Customer choice how to use the vdisks in those two DA's:
 - metadataOnly on SSD
 - if metadata performance is limited by HW performance, e.g small reads
 - dataonly on SSD for a GPFS tier, and using ILM to NL-SAS
 - Completely separate GPFS filesystems for different usage

+ Lenovo Top100 HPC Systems with IBM Spectrum Scale



CINECA Marconi

(Top #18 Skylake + KNL 16.2PF/s peak)

(Top #98 Broadwell 2.0PF/s peak)

Lenovo GSS @ OPA100



BSC Mare Nostrum

(Top #22 Skylake 10.3PF/s peak)

OPA100; IBM ESS @ 40GbE



SciNet Niagara

(Top #53 Skylake 4.6PF/s peak)

Lenovo DSS-G + Excelerio NVMe

@ EDR Dragonfly+



LRZ SuperMUC1/2

(Top #58 Haswell 3.6PF/s peak)

(Top #57 Sandybridge 3.2PF/s peak)

IBM/Lenovo GSS @ FDR10 / FDR14

+ More Lenovo DSS-G References



Univ. Birmingham Research Data Store:

First **DSS-G** systems shipped (04/2017)
Several SC17 HPCWire Reader's Choice Awards



LRZ „SuperMUC-NG“ Storage:

2x **DSS G220** @ 4TB + 2x **G202-SSD** (HOME)
4x **DSS G240** @ 12TB + 1x **G202-SSD** (PROJECT)
44x **DSS G220** @ 10TB + 1x **G202-SSD** (SCRATCH)
All connected to **OPA100**; Home/Project also to **40GbE**



NIMR / KMA Project Storage:

12x **DSS G260** @ 10TB
FDR → cNFS Ethernet
connecting to KMA's Cray XC40:
Top#**76** „Miri“ and Top#**77** „Nuri“



FZJ „JUST5“ Storage Cluster:

1x **DSS G260** @ 10TB (ARCH)
3x **DSS G240** @ 10TB (HOME)
18x **DSS G240** @ 10TB (WORK/DATA)
Large **100GbE** fabric, connecting to FZJ's
Top#**23** „JUWELS“ and Top#**38** „JUREKA“

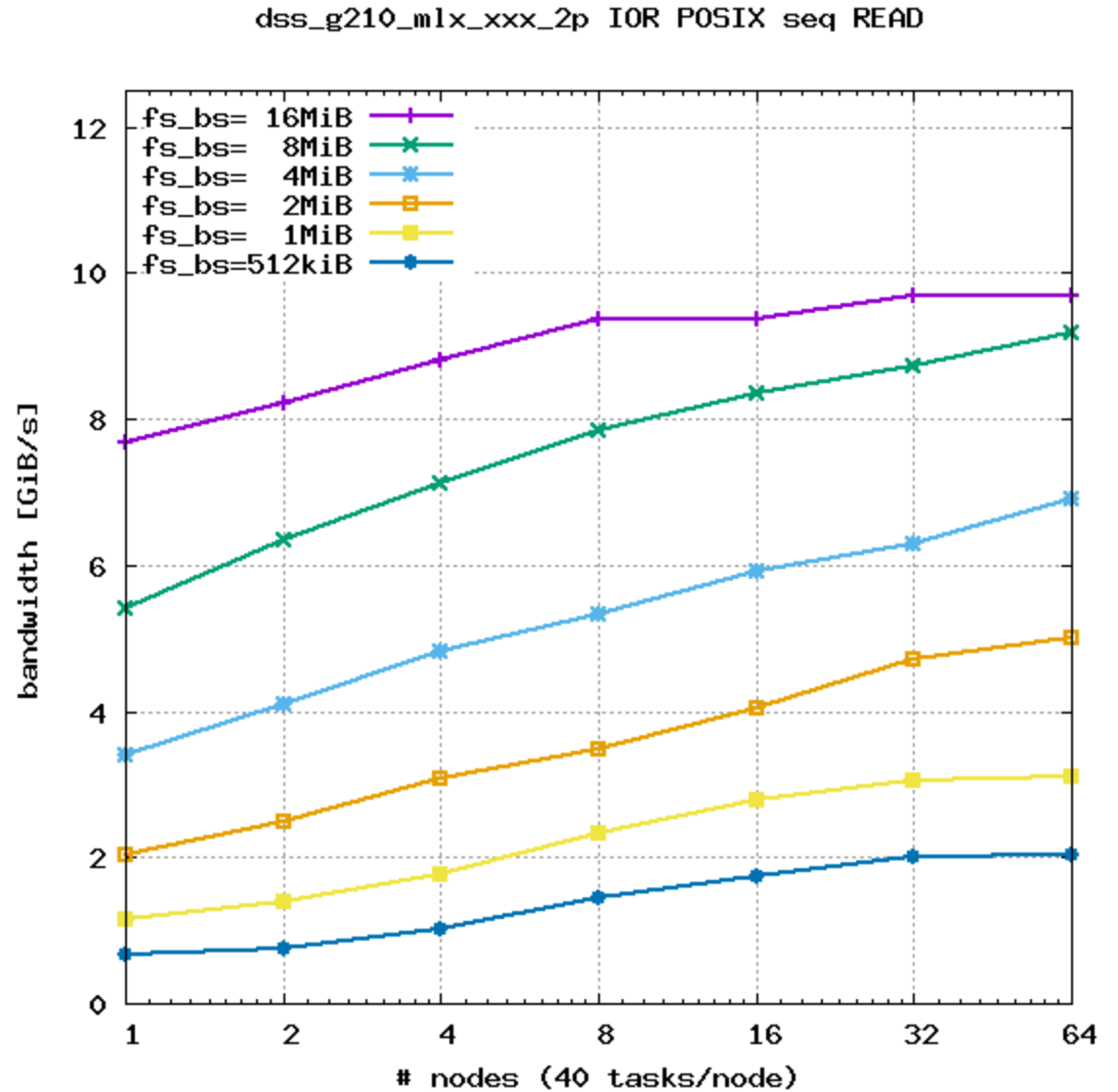
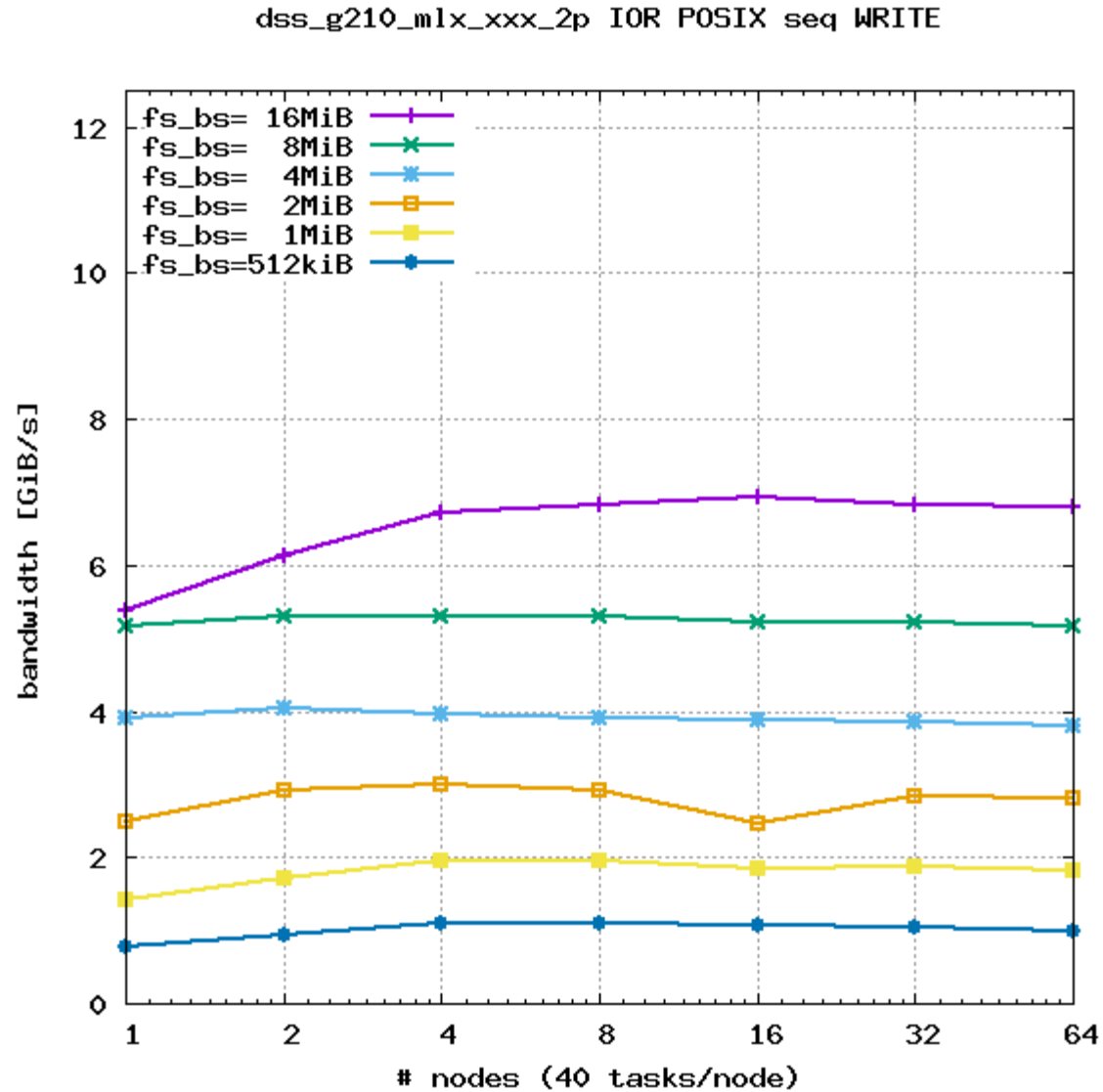
+ Spectrum Scale 5.0: More Sub-Blocks

- Scale 5.0 supports up to 1024 subblocks
- No longer need to trade-off performance (streaming I/O bandwidth scales with file system blocksize) and space efficiency
- Metadata improvements for shared directory (file creates, etc)
- Requires new file system format, cannot do online migration from old 32-subblock format to new format

new default FS blocksize in 5.0

Blocksize	Subblock size	# Subblocks
64 KiB	2 KiB	32
128 KiB	4 KiB	32
256 KiB	8 KiB	32
512 KiB	8 KiB	64
1 MiB	8 KiB	128
2 MiB	8 KiB	256
4 MiB	8 KiB	512
8 MiB	16 KiB	512
16 MiB	16 KiB	1024

+ DSS-G210 (NL-SAS) Filesystem Blocksize Dependence (gpfs-5.0.2)



+ Lenovo DSS-G100 NVMe Server



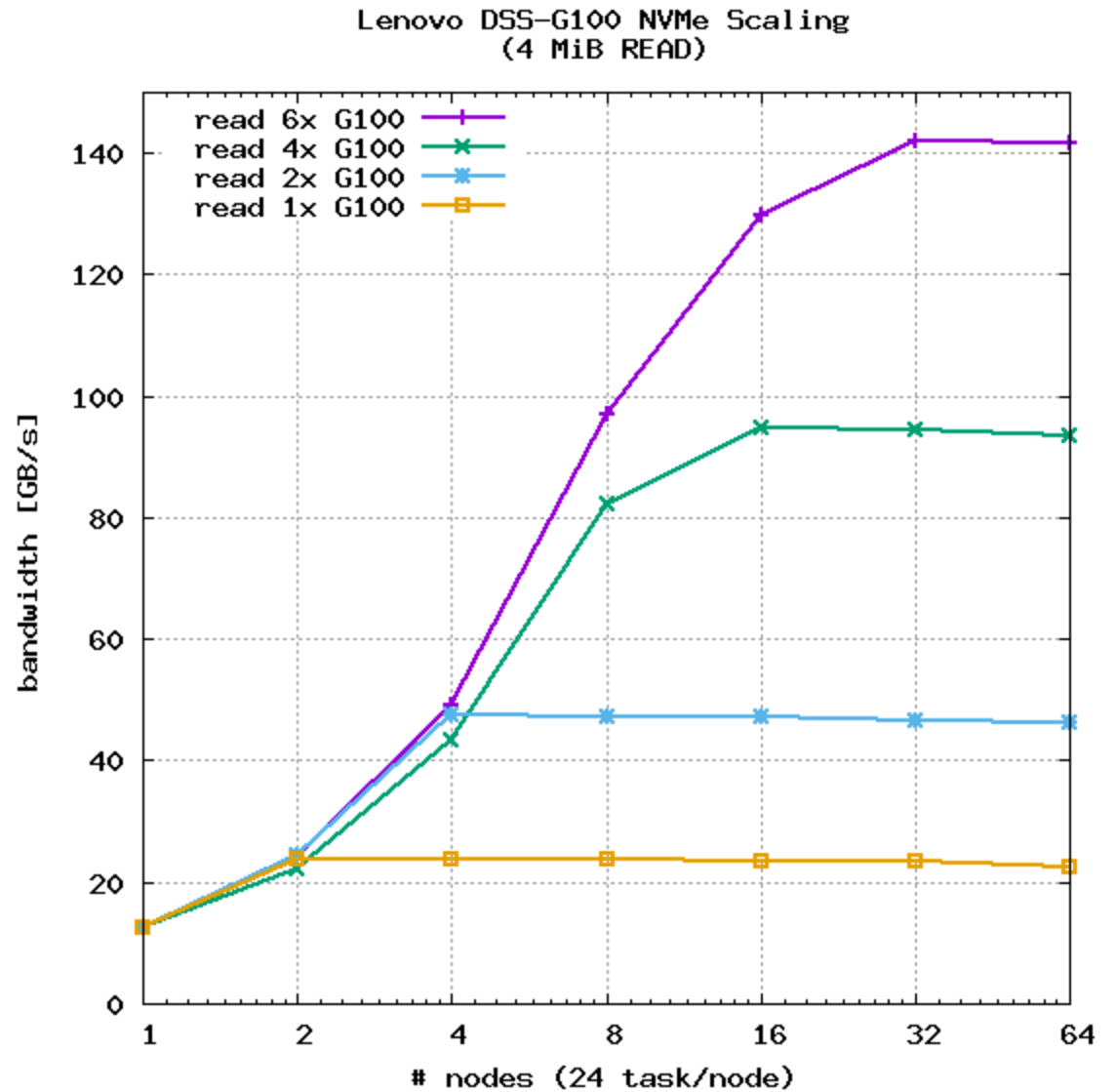
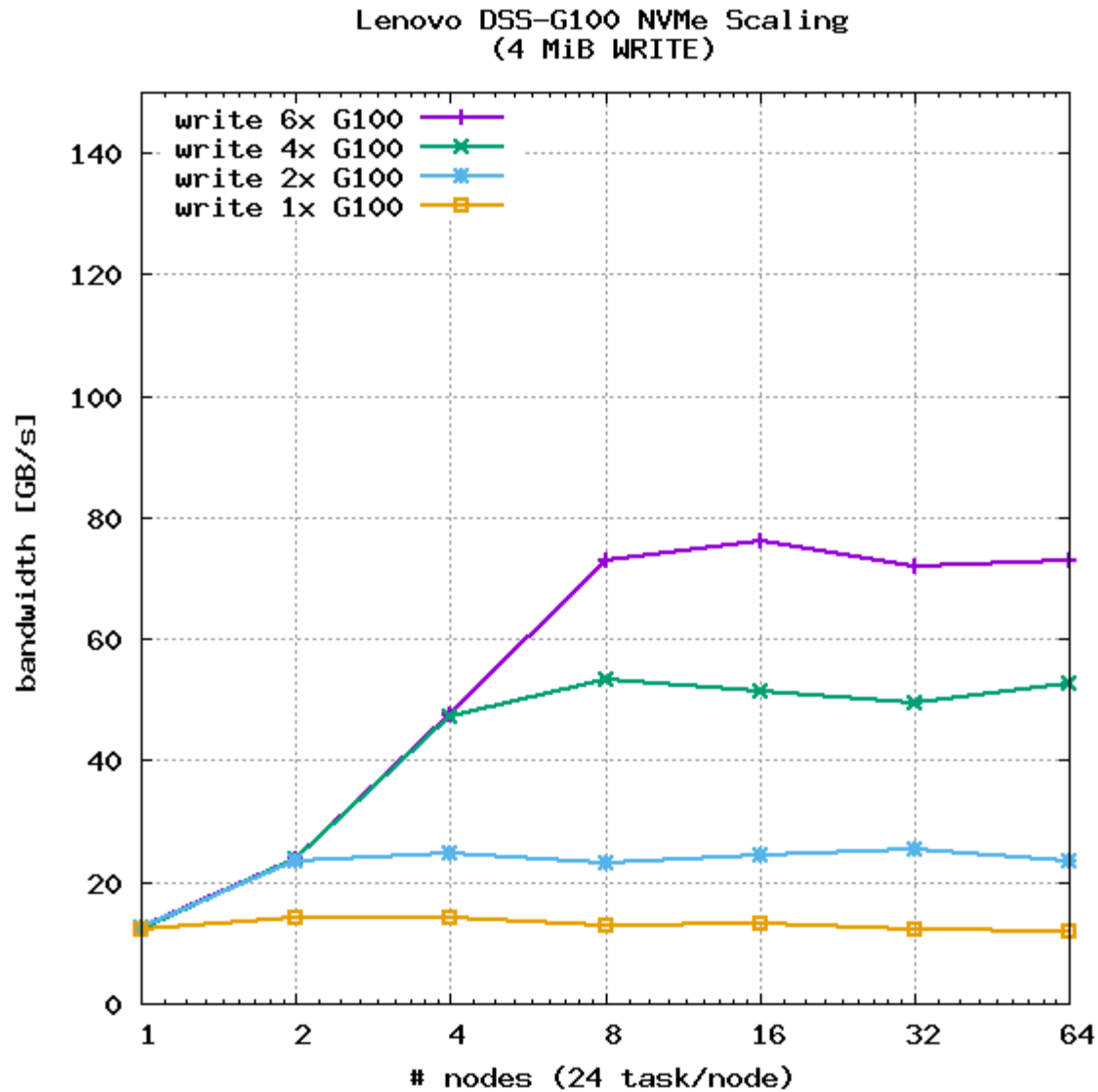
- Lenovo ThinkSystem SR630 server
 - **SR630** (1U) or SR650 (2U)
 - 2x SkyLake CPUs (4114); 192GB default
 - Up to **8x U.2 NVMe** drives in AnyBay slots
 - All NVMe drives from ThinkSystem portfolio...
 - Networking options:
 - 2x Mellanox ConnectX-5 2-port (VPI)
 - 2x Intel OPA100 1-port
 - Ethernet options: 10 / 25 / 40 / 100GbE
- Software stack options:
 - „Classical“ **IBM Spectrum Scale** (in Lenovo BOM)
 - **Excelero NVMesh** (in Lenovo BOM)
 - JBOF and RAID0/10 today, MeshProtect coming in v2.0
 - **E8-Storage** software certified on SR630 (VLS)
 - RAID within the server (4+1p; 8+2p)
 - **IBM „Mestor“**, if & when it becomes a product...



Current NVMe drive portfolio:

- Intel P4800X (Optane): 375/750 GB (30DWD)
- Intel P46x0: 1.6 / 3.2 / 6.4 TB (3DWD)
- Intel P45x0: 1 / 2 / 4 TB (<=1 DWD)
- Toshiba PX04PMB:
 - 800 GB / 1600 GB (10DWD)
 - 960 GB / 1920 GB (3DWD)
- Samsung PM963: 1.92 / 3.84TB (1DWD)

+ DSS-G100 (NVMe) Bandwidth – 8x P4610 1.6TB per server, unreplicated



Excelero NVMe

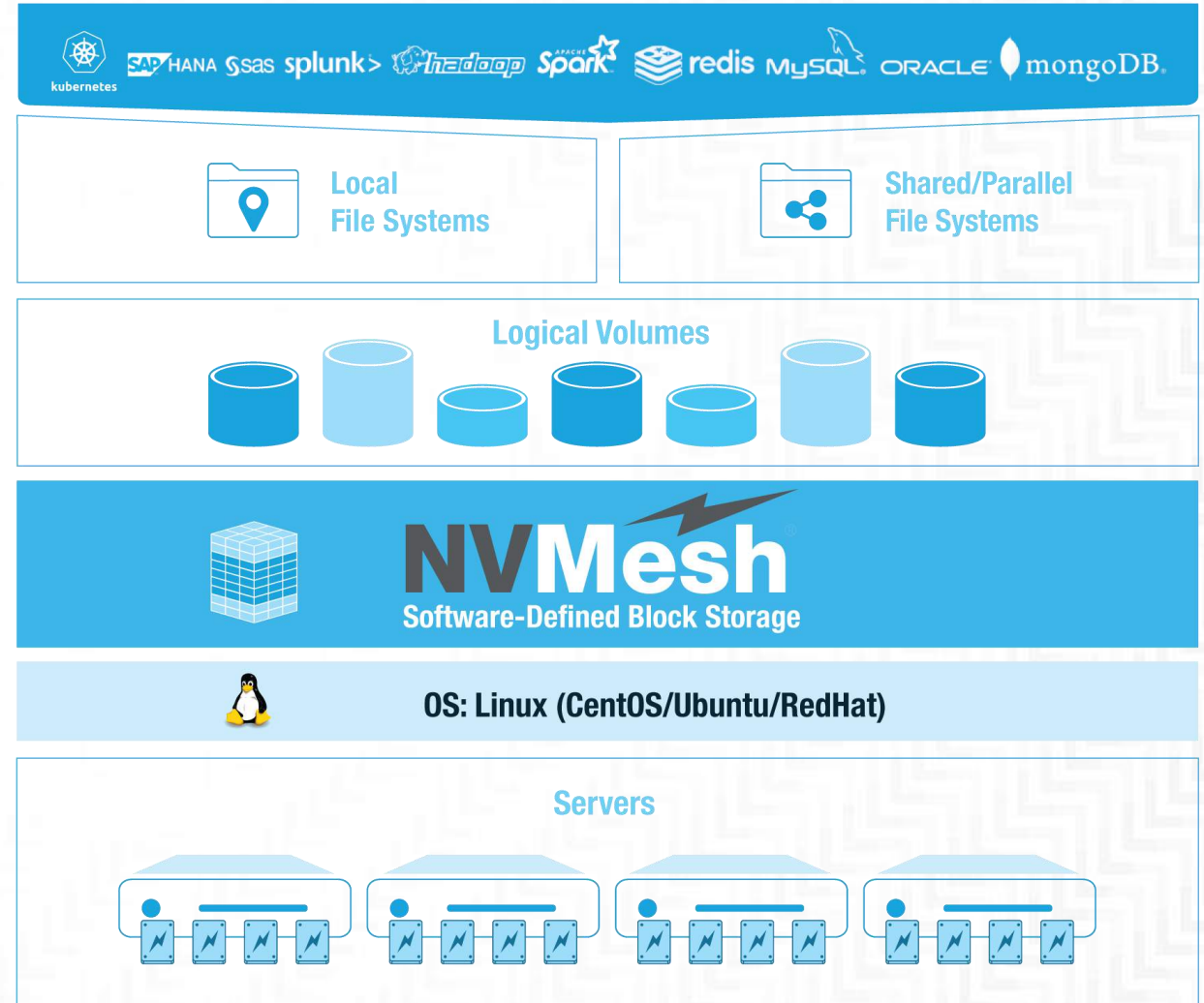
The „Server SAN“ for NVMe



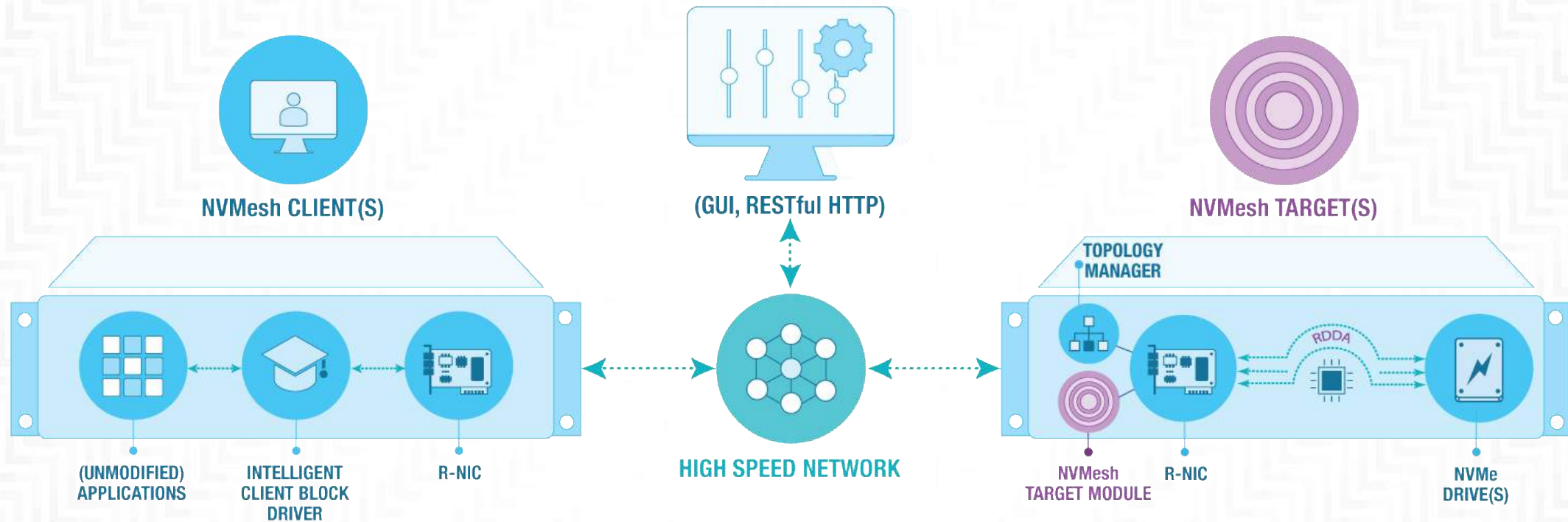
+ Excelero NVMesh Server SAN

NVMesh allows unmodified applications to utilize pooled NVMe storage devices across a network at local speeds and latencies.

Distributed NVMe storage resources are pooled with the ability to create arbitrary, dynamic block volumes that can be utilized by any host running the NVMesh block client.



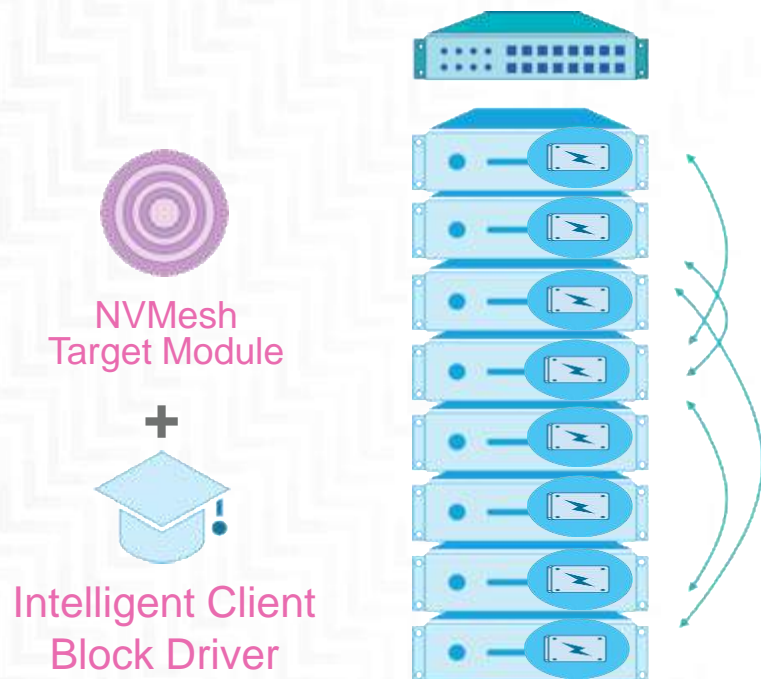
+ Excelero NVMesh Software and Hardware Components



- All server hardware must support PCIe 3.0
- Supported Linux OS distributions include RHEL/CentOS, Ubuntu and SLES
- NVMe devices are supported in PCIe adapter as well as U.2 and M.2 drive form-factors

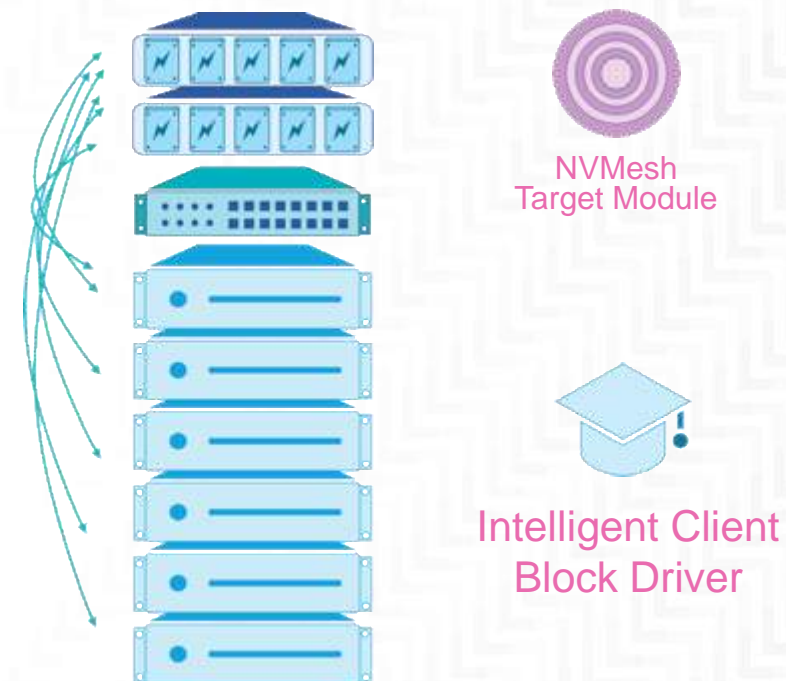
+ NVMesh Deployment Options

Local Storage in Application Server



- Storage is unified into one pool
- NVMesh Target Module & Intelligent Client Block Driver run on all nodes
- NVMesh bypasses server CPU
- Linearly scalable

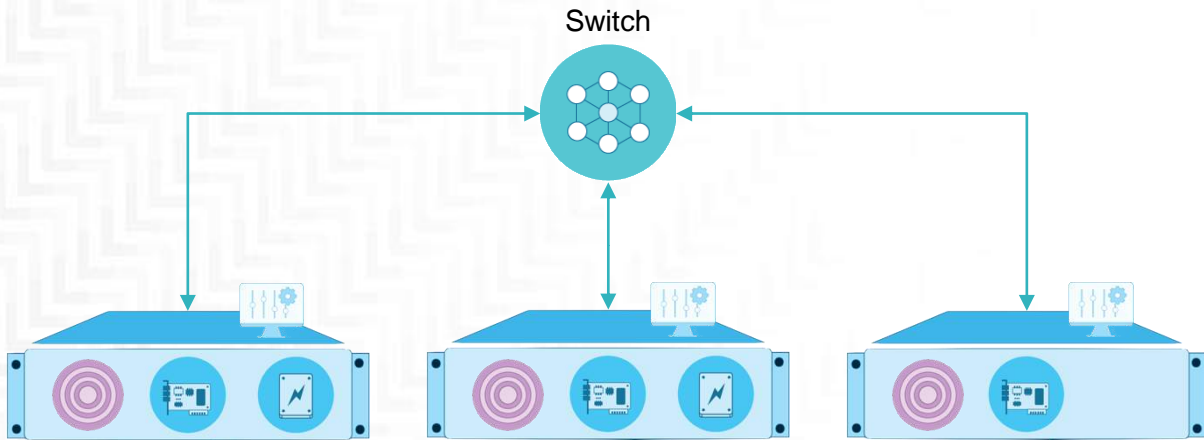
Centralized Storage





- Storage is unified into one pool
- NVMesh Target Module runs on storage nodes
- Intelligent Client Block Driver runs on compute nodes
- Applications get performance of local storage

+ Minimum (RAID 10) Configuration Logical Topology

- **3 Servers, 3 RDMA NICs, 2 NVMe Drives**



- (2) Servers with at least one NVMe in each
- (1) Server runs Target software but no NVMe drives*
- **NVMesh-management** deployed in HA on all 3 servers 
- All servers will run **NVMesh-client & Target Software** 
- All servers require at least one Mellanox RDMA NIC
- Network switch must be Data Center Bridging (DCB) capable with Global Pause (Tx + Rx flow-control) enabled for all storage network ports
- More advanced network configurations for flow-control are possible, but not required for small POCs

* NVMesh-target will be installed on the system for Topology Manager (RAFT) quorum

+ Coming with NVMesh2: MeshProtect



MeshProtect™

More Data Protection

NVMesh¹

NVMesh²

No Redundancy

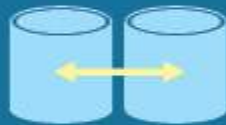
Mirroring (N+1)

Parity-based (N+M)

Concatenated



Mirrored



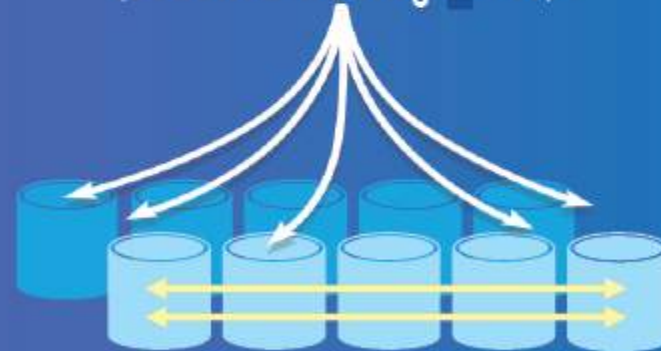
Striped



Striped & Mirrored



N+M Parity-based
(Erasure Coding N+M)



New!
90+% Usable

+ Lenovo ThinkSystem Servers Certified for NVMesh

Excelero NVMesh is certified on the following Lenovo servers (refer to the online reference at <https://www.excelero.com/nvmesh-interoperability-matrix/1.2.1/en/topic/servers> for the latest information):

Vendor	Model(s)	Form Factor	CPUs	NVMe Drives	x16 NIC Slots	Max. Read Bandwidth
Lenovo®	ThinkSystem SR630	1RU, 1 node, 19" Rack	Dual socket Intel Xeon® Scalable	8x U.2 (max 10x)	2	23 GB/s
Lenovo	ThinkSystem SR650	2RU, 1 node, 19" Rack	Dual socket Intel Xeon® Scalable	8x U.2 (max 24x)	2	23 GB/s
Lenovo	ThinkSystem SD530	2RU, 4 nodes, 19" Rack	4 xDual socket Intel Xeon® Scalable	16x U.2 (4x per node)	4 (1 per node)	46 GB/s

Other Lenovo server models can be certified/validated as needed...

+ For more Information...



POOLING NVME WITHIN GPFS NSDS ENABLES EFFICIENT BURST BUFFER

CASE STUDY



USE CASE
Large-scale modeling, simulation, analysis and visualization

CHALLENGE
Complete checkpoints within 16 minutes to meet availability SLA's

SOLUTION
NVMesh enables a petabyte-scale unified pool of distributed high-performance NVMe flash as burst buffer for checkpointing

RESULTS

- 80 pooled NVMe devices
- 145 GB/s of write burst (device limited)
- 280GB/s read throughput (network limited)
- Well over 20M random 4k IOPS

BENEFITS

- Meets the 16 minute checkpoint window
- Extremely cost effective
- Unheard of burst buffer bandwidth

SciNet is Canada's largest supercomputer center, providing Canadian researchers with computational resources and expertise necessary to perform their research at massive scale. The center helps power work from the biomedical sciences and aerospace engineering to astrophysics and climate science. SciNet is currently building what will be the fastest supercomputer in Canada. As the project is funded by government branches, such as the University of Toronto and Compute Canada, this new supercomputer needs to meet high levels of availability to ensure high ROI for the supercomputer. One way to increase availability is by using a burst buffer for checkpointing. This case study lays out how Excelero's NVMesh enables SciNet to build a petabyte-scale unified pool of distributed high-performance NVMe as a burst buffer for checkpointing. The NVMe pool delivers 230GB/s of throughput and well over 20M random 4k IOPS and enables SciNet to meet its availability SLA's.

High-performance computing applications consist of complex sets of processes that sometimes run for weeks. When any of these processes is interrupted, this could destroy the results of the entire compute job. This problem becomes worse as supercomputers become more powerful – imagine the challenge for Canada's soon to be largest supercomputer. Therefore, parallel computing applications use the concept of checkpoint-restart. This technique allows compute jobs to be restarted from the most recently saved checkpoint in case of an interruption.

Checkpoints are typically saved in a shared, parallel file system; SciNet has chosen GPFS. But as clusters become larger and the amount of memory per node increases, each individual checkpoint becomes larger and either takes more time to complete or requires a higher-performance file system. When a system is checkpointing it's not computing, which reduces the availability score of the system. To shorten those moments of unavailability, SciNet decided to implement a burst buffer leveraging Excelero's NVMesh.

COPYRIGHT © 2017 | INFO@EXCELERO.COM | WWW.EXCELERO.COM

1



NVMesh²

LOWEST-LATENCY DISTRIBUTED BLOCK STORAGE FOR SHARED NVMe

DATA SHEET

INTRODUCTION

The quest for zero-latency storage is real. In this era where technology is ubiquitous, the multitudinous latency-sensitive applications that surround us require fast and efficient processing of data at massive-scale. Providing near-zero latency at such scale is the remaining storage challenge and by extension, the most pressing technology challenge for web-scale data centers.



New-generation flash media, such as NVMe, are moving the bar on storage latency. Single-digit microseconds latency is a reality when used locally. This is setting expectations for application developers, who now get much better performance from one local NVMe flash device than an entire enterprise-grade all flash array.

Excelero delivers the lowest-latency (5µs) distributed block storage for web-scale applications: NVMesh enables shared NVMe across any network and supports any local or distributed file system. Customers benefit from the performance of local flash with the convenience of centralized storage while avoiding proprietary hardware lock-in and reducing the overall storage TCO.

100% SOFTWARE-DEFINED	LOWEST OVERHEAD	BLOCK STORAGE
Use any Hardware	Local Flash Latency across the Network	Use any File System

COPYRIGHT © 2018 | INFO@EXCELERO.COM | WWW.EXCELERO.COM

1

+ NVMesh Scratch on Demand: SLURM Integration

Using the <https://github.com/excelero/nvmesh-shell> command line tool (nvmesh)...



- **Sysadmin:** defines NVMesh volume provisioning group (VPG) **SCRATCH_8**, and defines its capacity as SLURM licenses (count=max capacity):

```
# in slurm.conf:  
Licenses=nvmesh_gb:12800
```

- **User:** requests desired NVMesh capacity as a SLURM license:

```
# in user's batch job:  
#SBATCH --nodes=4  
#SBATCH --licenses=nvmesh_gb:4000
```

- SLURM prolog for NVMesh (on each node):

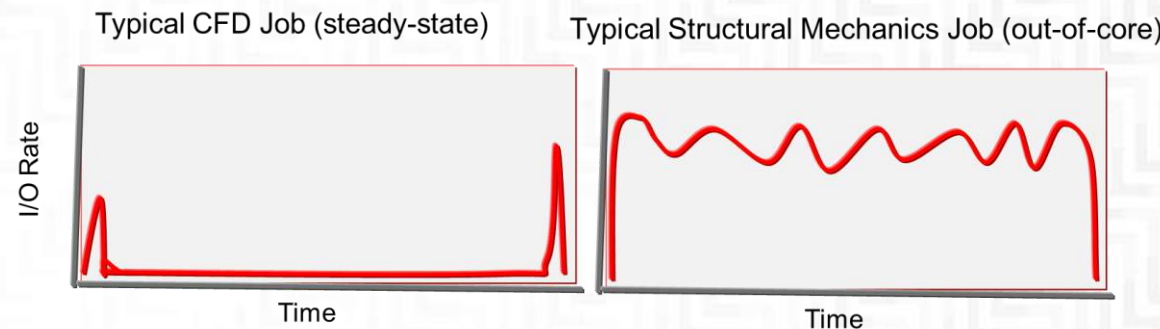
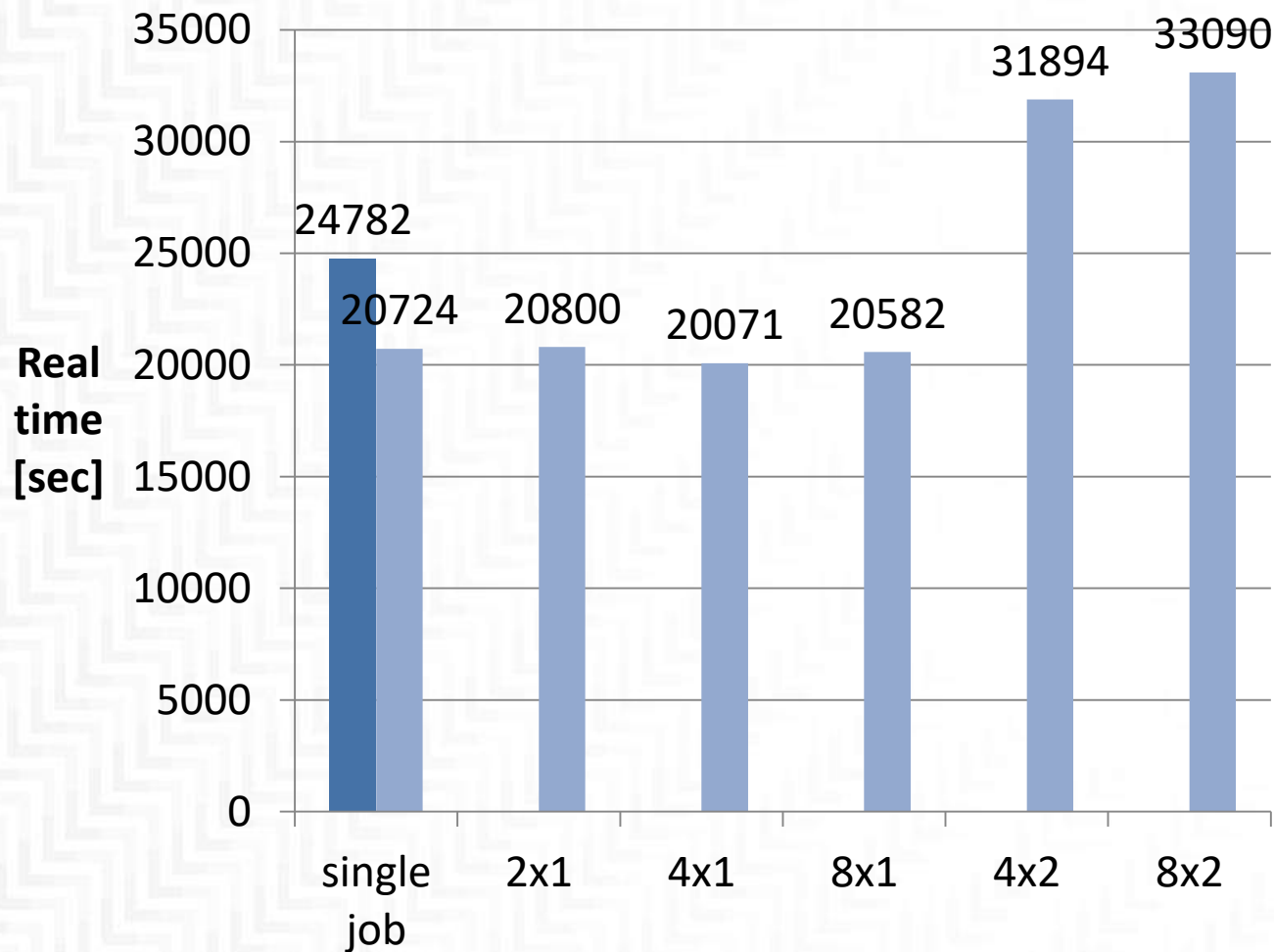
```
nvmesh add volume -n `hostname`  
-v SCRATCH_8 -S $GB_PER_NODE  
nvmesh_attach_volumes --wait_for_attach `hostname`  
mkfs -t xfs -f /dev/nvmesh/`hostname` # TRIMS...  
mkdir -p /nvmesh/scratch  
mount /dev/nvmesh/`hostname` /nvmesh/scratch
```

- SLURM epilog for NVMesh (on each node):

```
umount -f /nvmesh/scratch  
nvmesh_detach_volumes --force `hostname`  
nvmesh delete volume -v `hostname` --force --yes
```


+ Nastran SOL 400 Customer Case

Reference run on GPFS/GSS is 419min



Application: MSC Nastran 2016

- SMP=4, MEM=24gb
- Max disk usage: 62 GB
- I/O transferred: 8.6 TB
- 1 or 2 jobs per node

Excelero

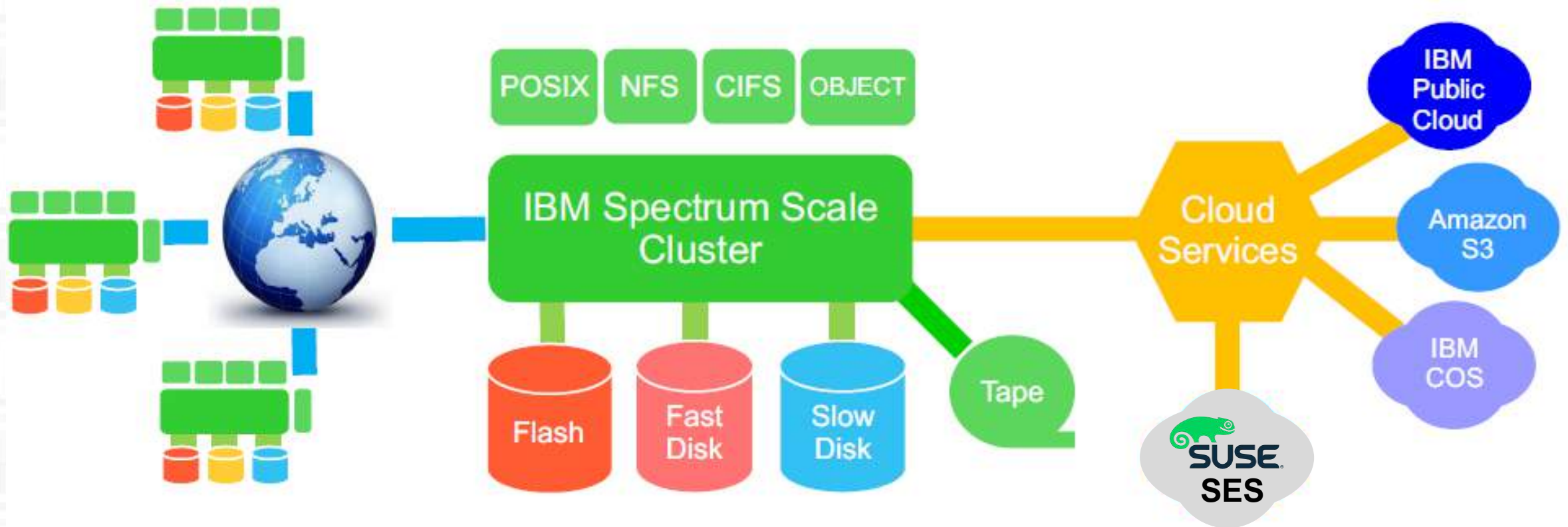
- 3 file server
- 2x 800 GB NVMe each
- RAID0 across 6 NVMe devices
- Up to 10 GB/s write (single node)
- Up to 8.5 GB/s write (8 clients)

Ceph / SUSE SES as a Spectrum Scale Tier

Lenovo / SUSE Proof-of-Concept

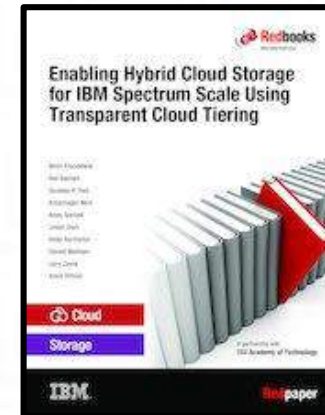
+ Spectrum Scale Data Management, incl. Cloud Tiering

Spectrum Scale's Information Lifecycle Management (ILM) rules can be used to control initial placement, migration, and deletion of data across storage pools.

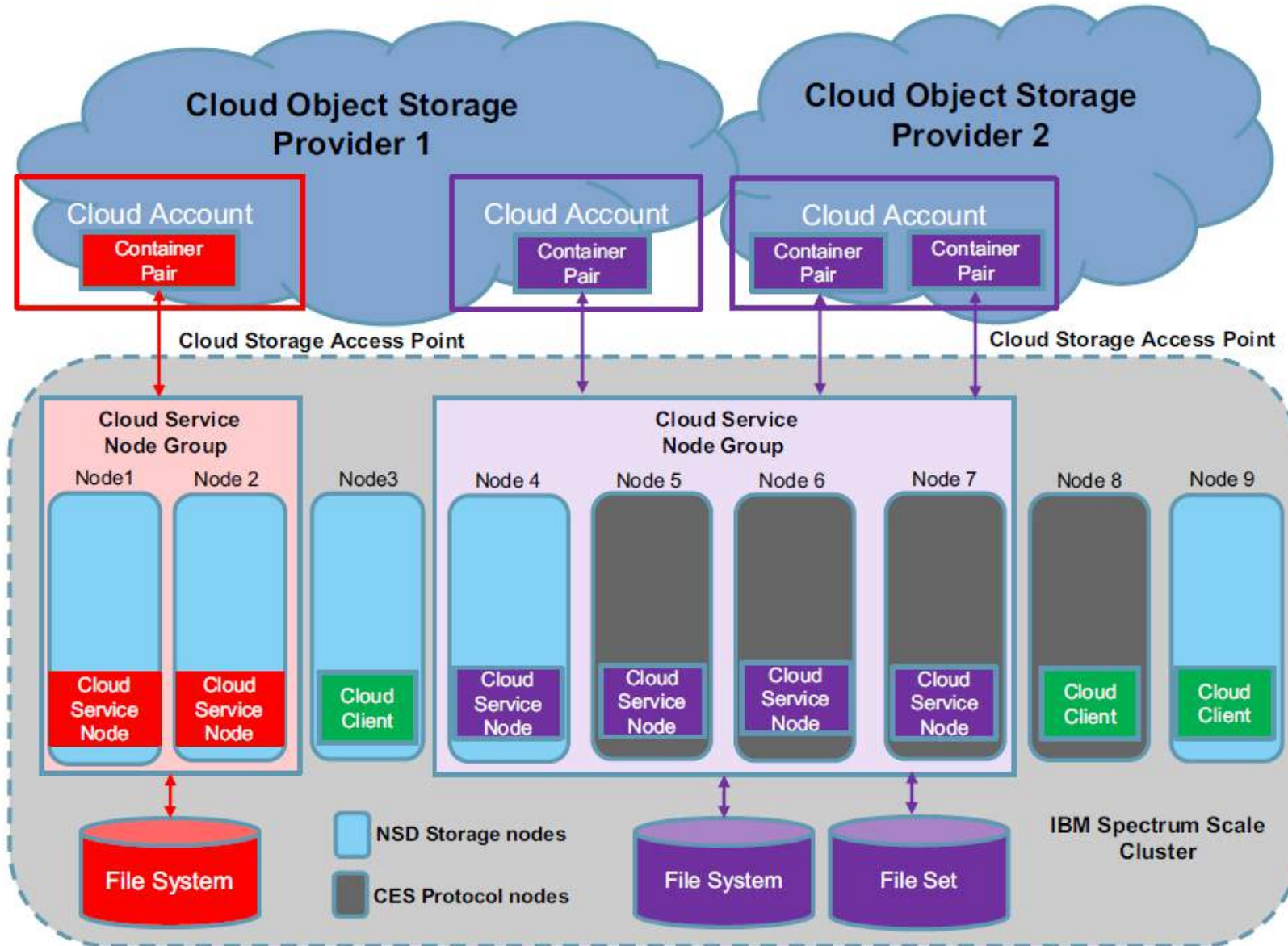


+ Transparent Cloud Tiering (TCT) – Documentation

- Spectrum Scale 5.0.2 Product Documentation:
 - Installation Guide:
Chapter 7. Installing Cloud services on IBM Spectrum Scale nodes. pp387-392
 - Administration Guide:
Chapter 6. Configuring and tuning your system for Cloud services. pp91-122
 - Command and Programming Reference: mmcloudgateway command. pp239-266
- Spectrum Scale Knowledge Center: „Planning for Cloud services”
 - https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.2/com.ibm.spectrum.scale.v5r02.doc/bl1ins_planning_MCS.htm
- IBM Redbooks (one for **tiering**, one for data sharing):
 - **Enabling Hybrid Cloud Storage for IBM Spectrum Scale Using Transparent Cloud Tiering**
<http://www.redbooks.ibm.com/abstracts/redp5411.html?Open>
 - Cloud Data Sharing with IBM Spectrum Scale
<http://www.redbooks.ibm.com/abstracts/redp5419.html?Open>
- Spectrum Scale User Group Talks → www.spectrumscale.org



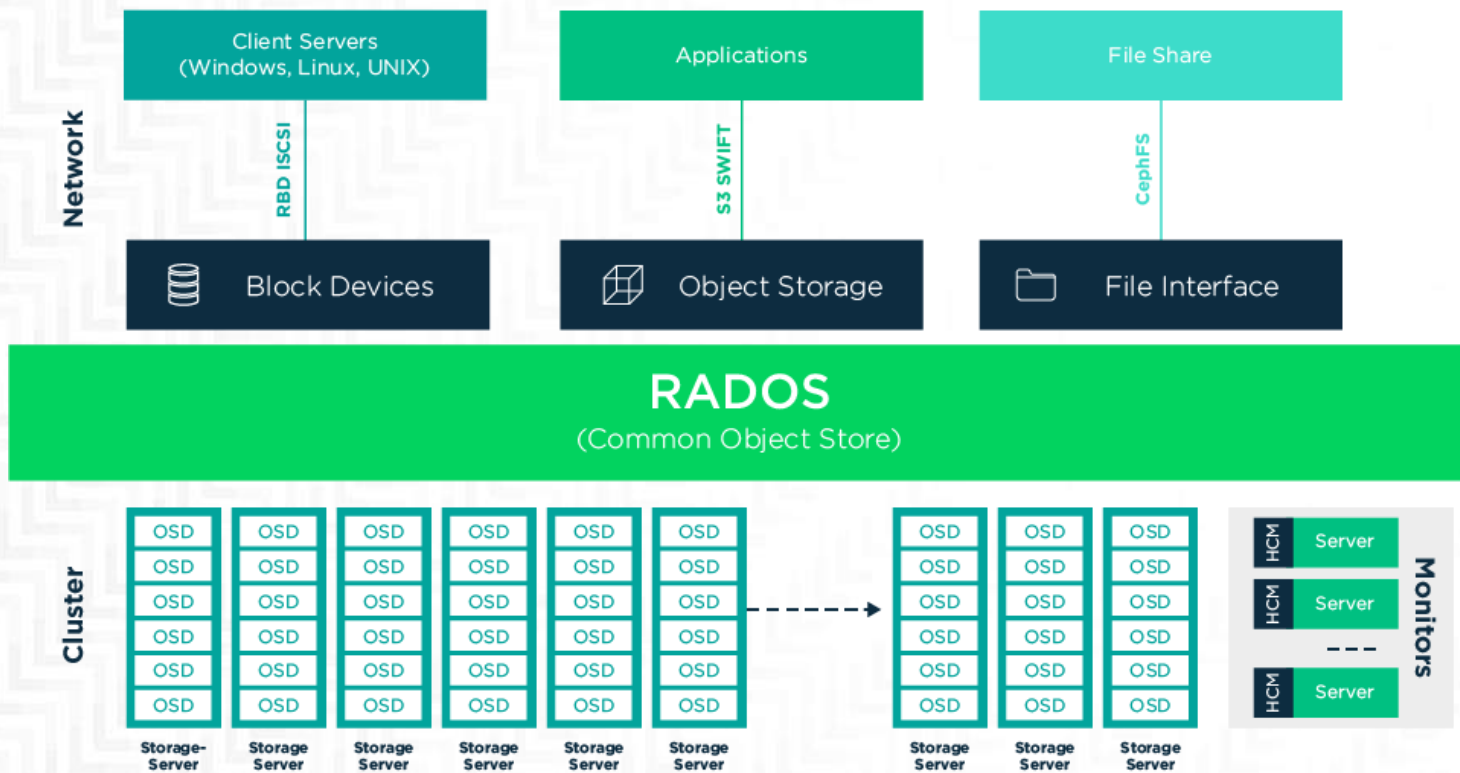
+ Transparent Cloud Tiering: Components and Terminology



+ Tiering to SUSE Enterprise Storage (powered by Ceph)



- Minimum 4x Object Storage Nodes and 1x Management Node
- Ceph monitors (3x), gateway nodes, metadata servers can reside on OSD servers, but for demanding workloads they should run on separate nodes



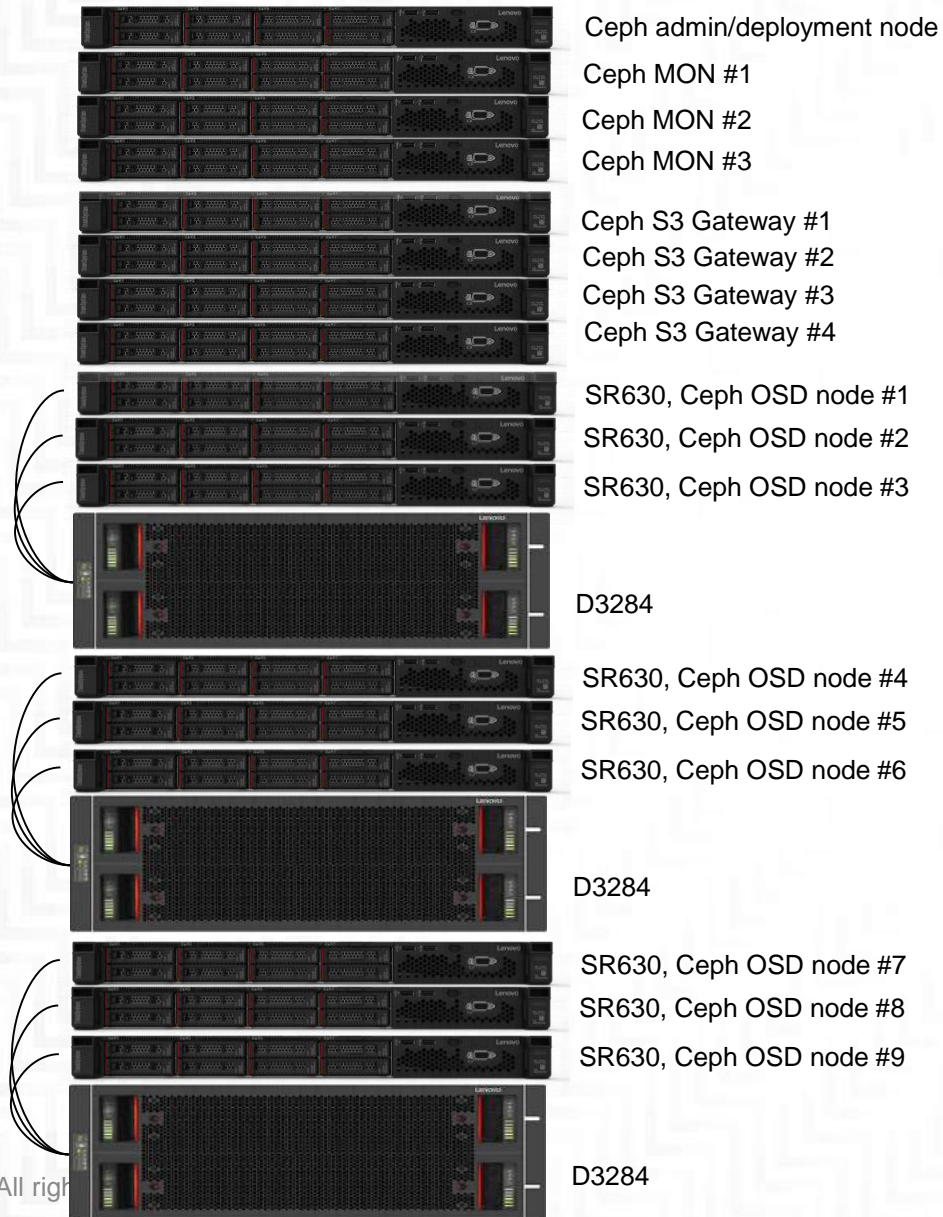
DSS-C Hardware used for the POC:

- **6x Ceph OSD nodes – “Capacity”**
 - SR650, single-socket (4110)
 - **12x HDD 3.5” SATA 4TB**
 - 4-port 10GbE
 - 128GB memory
 - ➔ Ceph storage pool is 3WayReplicated
- **3x Ceph MON nodes**
 - SR530, single-socket (3104)
 - 2-port 10GbE
 - 16GB memory
- **2x radosgw roles**, colocated on 2x OSDs
 - Dedicated 1x 10GbE links to „WAN“
- Maximum „**WAN**“ **bandwidth** between SES cluster and Spectrum Scale TCT nodes:
 - **2x 10GbE Ethernet**, plan to test 100GbE

+ Why Ceph as an Engineered Solution?

- SDS cost savings & flexibility:
 - Avoid large markup by storage vendor on hardware
 - Share hardware resources between storage and application; increases utilization; get more work out of less hardware
 - More customer flexibility in choosing the best hardware parts for their needs
- SDS disadvantages, when done by yourself:
 - Customer is responsible for selecting and installing hardware; may not provision adequately for the needs of the software
 - **Lenovo Solution Architecture & Support, with joint testing efforts of Lenovo and SUSE**
 - Customer is responsible to debug problems and then work with server, storage, OS, or networking vendor
 - **Lenovo can be the single point of contact for support, with Level3 SUSE support for SES**
 - Software vendor has to be prepared to support their software running on almost any reasonable hardware
 - **SUSE and Lenovo have defined a portfolio using few building blocks only**

+ Example: Ceph solution for Spectrum Scale TCT Tiering



G8052 switch for 1G XCC and OS management
NE10032 switch for Ceph-internal comms and client uplinks

TOTAL CAPACITY:

Using 10 TB drives

9x 28x 10 TB = 2.5 PB raw
with 6+2 erasure coding = **1.9 PB usable capacity**
This fits in 32U.

Another SR630/D3284 block brings total to
12x 28x 10 TB = 3.4 PB raw
with 9+2 erasure coding = **2.8 PB usable capacity**
This fits in 40U.

+ TCT Setup for Ceph, 1 of 2

```
mmcrnodeclass tct_group1 -N tct2201  
mmchnode --cloud-gateway-enable -N tct_group1  
mmcloudgateway service start -N tct_group1
```

Ask an IBM (or Lenovo) technical specialist for instructions how to enable TCT. It is not enabled by default, to ensure a review of the intended use case is done before using TCT.

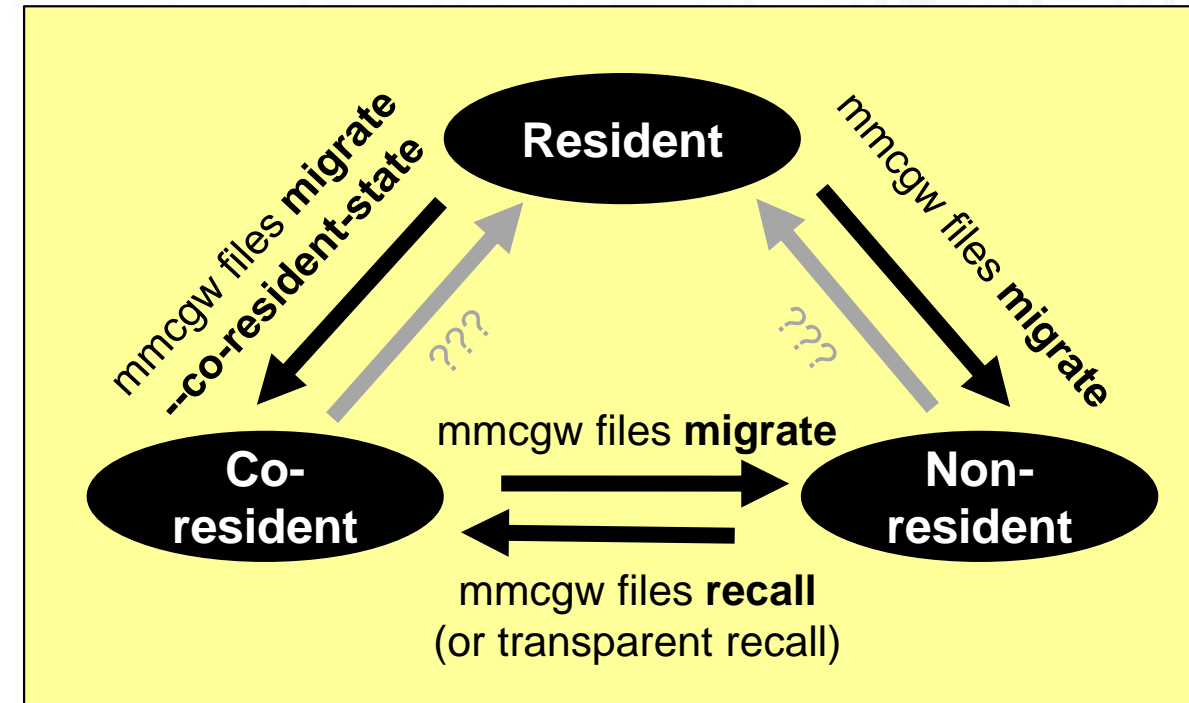
```
echo "odgmAbQLZTwXS1h2EKKjaoo00KCFXhOZjYYkY4w1" > ~/ceph_user1_password.txt  
mmcloudgateway account create  
  --cloud-nodeclass tct_group1 \  
  --account-name ceph_user1 \  
  --cloud-type CLEVERSAFE-NEW \  
  --username 91ESX930CJ8D1SL5S4DG --pwd-file ~/ceph_user1_password.txt  
  
mmcloudgateway cloudStorageAccessPoint create \  
  --cloud-nodeclass tct_group1 \  
  --account-name ceph_user1 \  
  --cloud-storage-access-point-name csap1-ceph_user1 \  
  --url "http://is-sr650a-tct.ses.eu.lenovo.com:7480"
```

+ TCT Setup for Ceph, 2 of 2

```
mmcloudgateway cloudService create \  
  --cloud-nodeclass tct_group1 \  
  --cloud-service-name tct_tiering1-ceph_user1 \  
  --cloud-service-type Tiering \  
  --account-name ceph_user1
```

```
mmcloudgateway containerPairSet create \  
  --cloud-nodeclass tct_group1 \  
  --container-pair-set-name tct_fs1_container1 \  
  --cloud-service-name tct_tiering1-ceph_user1 \  
  --scope-to-filesystem --path /gpfs/tct_fs1 \  
  --enc DISABLE \  
  --transparent-recalls ENABLE
```

```
mmcloudgateway maintenance list
```



```
mmcloudgateway files \  
  { migrate | recall | restore |  
    delete | destroy | reconcile |  
    list | cloudList | ... }
```

+ Performance Observations ... Work in Progress ;-)

- Always test the networking layer first ... using **iperf3** or similar tools
- Basic testing of the the Ceph **radosgw** performance can be done with **s3cmd**
 - Beware the default **--multipart-chunk-size-mb=15**, increase it to up to **5120** (5 GiB)
 - **s3cmd put** performance is unreliable: delay in beginning, stalls ... watch the network (**dstat**)
 - **s3cmd get** is more reliable; single **get** achieves ~300MiB/s, 4x in parallel saturate 10GbE link
- Explicit **migrate** and **recall** tests with **mmcloudgateway files**
 - **recall** runs at 10GbE wirespeed, even if only a single file is recalled (threaded multipart get)
 - **migrate** is slower: ~100MiB/s for single file, scales up to ~300MiB/s for parallel migrates
 - Beware the CSAP's **--mpu-parts-size** max=default of 128MiB ... this needs to increase!
- Usage of a second **cloudStorageAccessPoint** is erratic
 - Single file always seems to go through single CSAP
 - For multiple files (e.g. parallel **recall** calls), 2nd CSAP is sometimes used, but unpredictable

+ mmccloudgateway Command Categories

mmccloudgateway

node

list

account

{create,update,delete,list}

CloudStorageAccessPoint

{create,update,delete,list}

cloudService # Tiering or Sharing

{create,update,delete,list}

[keymanager

{create,update,rotate,list}]

containerPairSet

{create,test,update,delete,list}

maintenance

{create,update,delete,list,
setState,setFrequency,status}

mmccloudgateway

[config

{set,list}]

service

{start,stop,status,version,
backupConfig,restoreConfig}

files

{migrate,recall,restore,
delete,destroy,reconcile,
list,cloudList,
rebuildDB,defragDB,
backupDB,checkDB,
import,export}

mmccloudgateway <category> # show usage

+ mmcloudgateway files Command

```
migrate    [-v] [--co-resident-state] [--cloud-service-name CloudServiceName] [--] File[ File ...]
recall     [-v] [--local-cluster LocalCluster] [--owning-cluster OwningCluster] [--] File[ File ...]
restore     [-v] [--overwrite] [--restore-stubs-only]
            {-F FileListFile | [--dry-run] [--restore-location RestoreLocation] [--id Id] [--] File}
delete     [--delete-local-file | --recall-cloud-file | --require-local-file] [--keep-last-cloud-file] [--] File [File ...]
destroy     [--cloud-retention-period-days CloudRetentionPeriodDays [--preview]] [--timeout Minutes]
            --container-pair-set-name ContainerPairSetName --filesystem-path FilesystemPath
reconcile   --container-pair-set-name ContainerPairSetName Device
cloudList   { --path Path [--recursive [--depth Depth]] [--file File] | --file-versions File |
            --files-usage --path Path [--depth Depth] | --reconcile-status --path Path |
            --path Path --start YYYY-MM-DD[-HH:mm] --end YYYY-MM-DD[-HH:mm] }
backupDB    --container-pair-set-name ContainerPairSetName
checkDB     --container-pair-set-name ContainerPairSetName
rebuildDB   --container-pair-set-name ContainerPairSetName Device
defragDB    --container-pair-set-name ContainerPairSetName
list        [--] File[ File ...]
import      [--cloud-service-name CloudServiceName] [--container Container] [--import-only-stub] [--import-metadata]
            { [--directory Directory] | [--directory-root DirectoryRoot] | [--target-name TargetName] } File[ File]
export      [--cloud-service-name CloudServiceName] [--target-name TargetName] [--container Container]
            [--manifest-file ManifestFile [--tag Tag]] [--export-metadata [--fail-if-metadata-too-big]]
            [--strip-filesystem-root] File[ File]
```

thanks.

mhennecke @ lenovo.com

