STORAGE DEVELOPER CONFERENCE

SDC 21

BY Developers FOR Developers

Virtual Conference
September 28-29, 2021

The perspective of today's storage architectures.

How did we get here?

Where we are today?

Where are we going?

Presented by Supermicro

SUPERMICRO

intel

Powered by Intel®

Timeline

1980	1985	1990	1995	2000	2005	2010	2015	2020	2025
Fixed Geometry	SCSI - Block addressing			Thin Provisioning - Capacity Abstraction			NVMe		
Proprietary Interfaces	Parallel - 5 MB/s - 320 MB/s			Serial Attached SCSI (SAS) (3 - 24 Gb/S)					
			Fibre Channel (1 -64 Gb/s)	iSCSI	Parallel File Systems	Object Storage		PCI-e (Gen4/5)	
Block Storage				Ethernet as a storage interface (iSCSI)		SDS (Ceph)		NVMeoF	

- The Journey
  - In addition to the timeline shown above many other initiatives were introduced along the way
    - Some more longer lasting than others!

1980
Fixed Geometry
Proprietary Interfaces

Fixed Geometries

- O/S - Device Drivers required knowledge of the disk's head, track and sector information
- New devices could not easily be added with this constraint
- Storage Interfaces (mainly) proprietary across hardware vendors.
  - As an example the *Storage Module Device* (SMD) interface used two interface cables (A cable for control and B cable for data) and was used with removable and non-removable disk drives.
    - Popular with Mini-computer manufacturers such as Digital Equipment Corporation (DEC)
    - Registers were loaded with the cylinder, head and sectors parameters prior to reading and writing
  - Heavy burden on the O/S and device drivers

1985

SCSI - Block addressing

The path to storage abstraction

- Small Computer Systems Interface (SCSI)
  - SCSI evolved from Shugart Associates System Interface (SASI)
  - During the mid eighties (SCSI), was adopted by *ANSI*
    - Referred to as SCSI-1
  - Initially this was a *parallel* 8-bit data transfer implementation with a 5 MBps transfer speed
  - Up to eight devices could be connected
    - Typically one Host Bus Adapter and up to seven peripheral units
  - Maximum cable length of six meters for *single ended* devices and 25 meters for *differential* devices

The path to storage abstraction

- Later versions used a single *low voltage differential (LVD)* interface capable of around 12 meter cable lengths
  - Over time hardware implementations increased the device connectivity to 16 devices with speeds increasing geometrically from 5 MBps to 640 MBps
  - After this *Serial Attached SCSI* (SAS) replaced parallel SCSI allowing for higher transfer speeds due to the nature of *serial architecture*

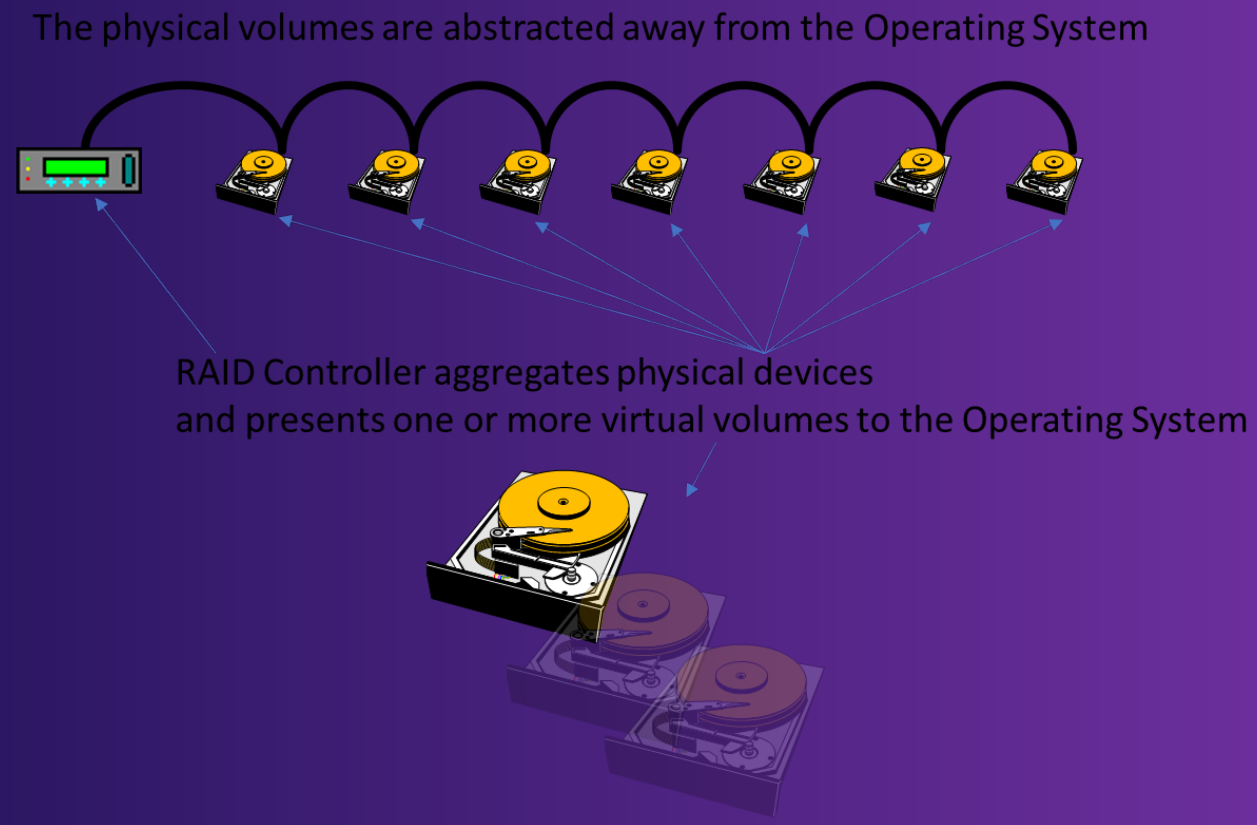
SCSI Interface

- Geometries were abstracted
  - Data seen as a sequence of contiguous blocks
- Devices were more intelligent and relieved the burden from the controllers/device drivers
  - Addressable sectors are presented as a sequence of contiguous blocks
  - Data transfers involve parameters such as:
    - Opcode (Read/Write).
    - The starting block from where to initiate the operation
    - How many blocks were involved in the transfer
  - Commands encapsulated within a *Command Descriptor Block* (CDB)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Operation Code							
Byte 1	Reserved			Logical Block Address MSB				
Byte 2	Logical Block Address							
Byte 3	Logical Block Address							(LSB)
Byte 4	Transfer Length							
Byte 5	Control Byte							

Hardware RAID

- While SCSI disk devices provided a degree of abstraction
  - Generally there was generally a one to one correspondence between the physical devices and logical devices as viewed by the Operating System
- *RAID* Controllers have the ability to aggregate disks into larger virtual volumes
  - Present the volumes as individual devices (LUNs) to the O/S



Hardware RAID

- RAID (in its strictest sense) involved redundancy for error recovery
  - As a result not all of the physical device's capacity was available to the logical units
  - The diagram opposite shows six physical disks (PDs) with interleaved data and recovery data
  - The capacity of 4 drives are used for data storage and the capacity of 2 drives are required for the RAID6 algorithm
  - Usable capacity is therefore 2/3 of the total capacity

RAID Level 6
(Left Symmetric Parity Placement)

PD0	PD1	PD2	PD3	PD4	PD5
0	1	2	3	P0	Q0
6	7	P1	Q1	4	5
P2	Q2	8	9	10	11
12	13	14	15	P3	Q3
18	19	P4	Q4	16	17

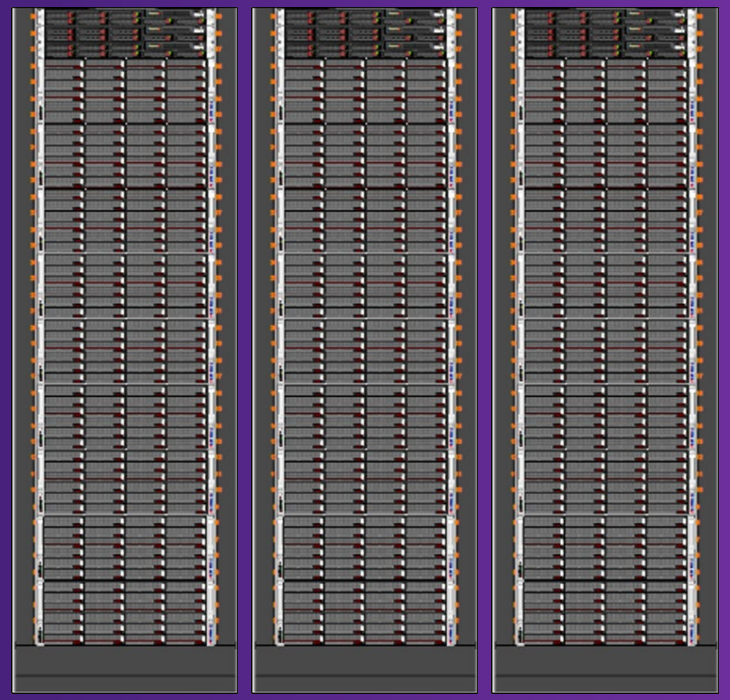
Limitations of locally attached storage

- SCSI led to a level of *commoditization* of devices as vendors began to provide support for these more open devices
- RAID controllers offered enhanced performance and reliability but were still hardware based often requiring expensive *ASICs* to reach the performance required for the complex error correction algorithms
  - Software based RAID was slow when using parity based RAID levels
    - Mirrored RAID levels fairly common

Limitations of locally attached storage

- Storage Solutions were still monolithic in that they were mainly *scale up* with the ability to daisy chain additional storage within the same domain.
- These islands of storage did not allow for easy sharing of data
  - The speed difference between storage interfaces and networking meant that the two interfaces were kept distinct and dedicated to their own tasks

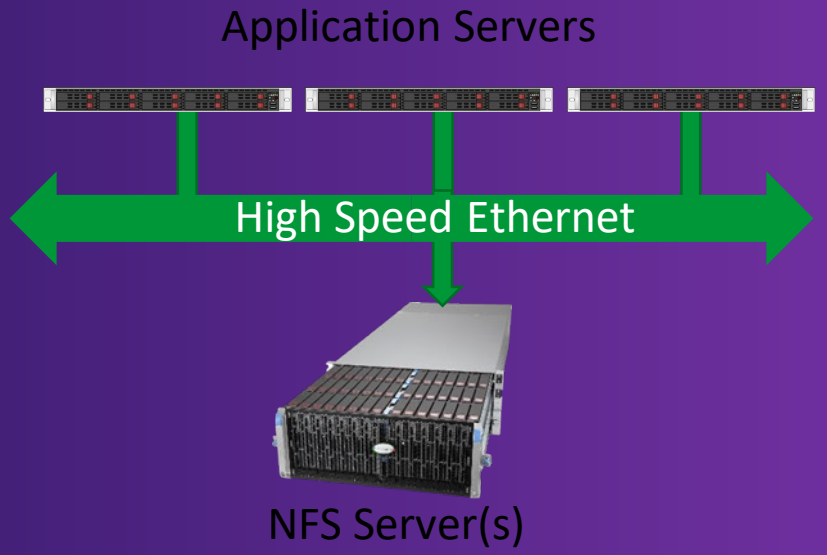
Independent Data Silos



1995	2000	2005
	Fibre Channel	iSCSI

Ethernet as a storage interface

- As Ethernet networks became faster, data could be shared from a dedicated storage device to multiple systems
- File sharing in the form of *SMB* and *NFS* became more common and in the late 1990s/2000s block sharing with *iSCSI* using ubiquitous Ethernet began to take hold
- Again, vendors could take advantage of commodity interfaces which were much more cost effective than high speed *SANs* using *Fibre Channel*



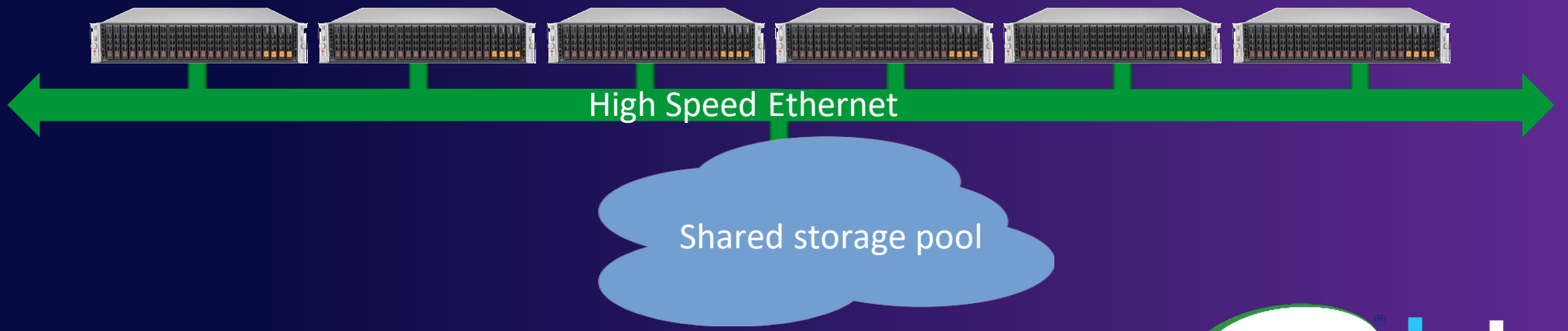
Scale out Models

- During this time computing and storage nodes began to converge
  - Compute nodes became powerful enough to house and maintain their own locally attached storage
  - PCI-e based RAID was common (and still is)
  - SAS Host Bus Adapters combined with software RAID was more cost effective

2005	2010	2015
Parallel File Systems	Object Storage	

Parallel File Systems

- With scale out files could be distributed across nodes
- Data could be accessed in parallel leading to performance enhancements
- Dedicated servers for Metadata were often used to avoid file system bottlenecks
- Examples are IBM Spectrum Scale (formerly GPFS) and Lustre



2005	2010	2015
Parallel File Systems	Object Storage	

Object Storage

- Objects are unstructured in that they do not possess the tabular structure required for a traditional database
- No fixed record sizes
- The objects could be documents such as PDFs, pictures in the form of JPEG files, or other media types such a video files or collected news feeds.
- Unstructured data will have its own associated metadata such as when and where a file was created facilitating searches without the need for a database.

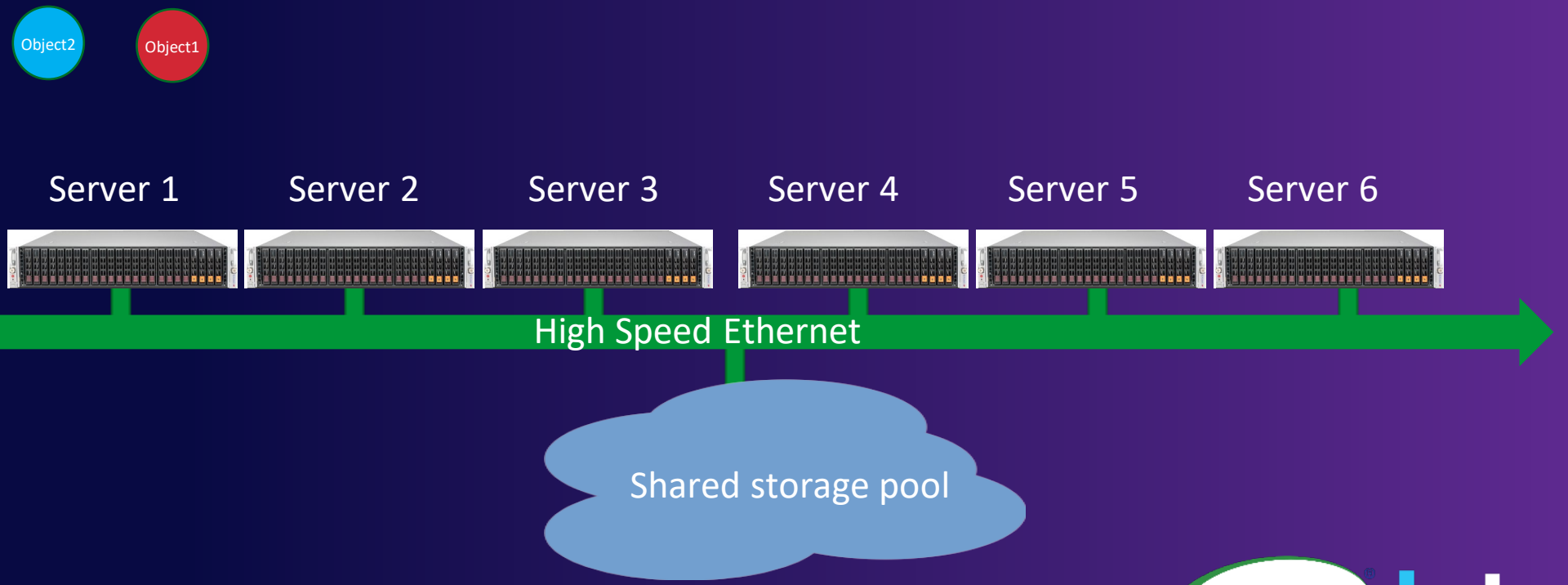
Table1 | Table1

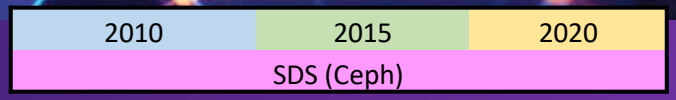
ID	Code	Status	Open Date	Close date
1	1345	Open	6/1/2020	
2	1260	Open	5/29/2020	
3	14530	Closed	5/28/2020	6/4/2020
(New)	0			

2005	2010	2015
Parallel File Systems	Object Storage	

Object Storage

- Data Resiliency
  - Many SDS systems use replication to ensure data availability
    - 3 X replication would store data on three distinct servers
      - Here an object could be written to server 3, server 5 and server 1 while a second object could be written to server 1, server 4 and server 6





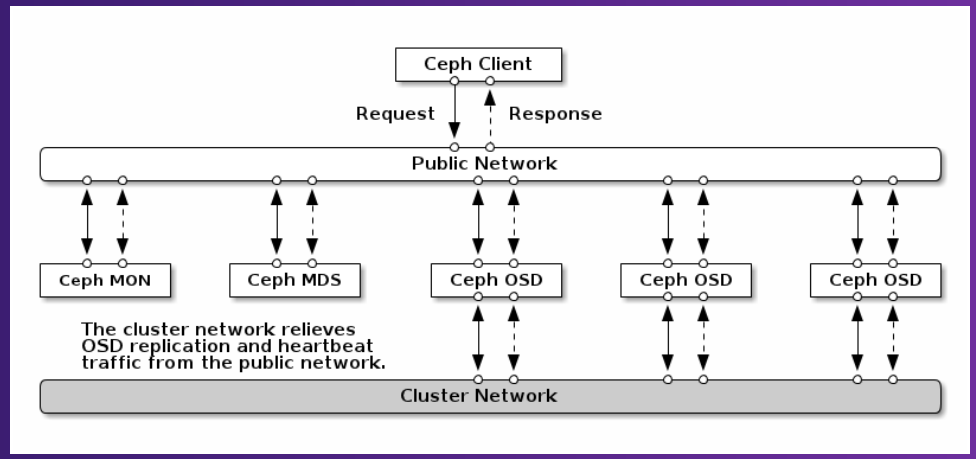
Object Storage - Ceph

- An Example of Open Source Private Cloud Software
  - Ceph
    - Supports Block, File and Object Based Storage
    - Community and Enterprise supported versions exist.
      - As of May 2021 the community has released V16 (Pacific) of Ceph.
    - Ceph can function as persistent storage for an OpenStack/Kubernetes environment
    - Ceph provides an Object Gateway which is compatible with Amazon S3 APIs and also OpenStack Swift.

2010	2015	2020
SDS (Ceph)		

Object Storage - Ceph

- Ceph architecture consists of:
  - Monitor Nodes (MON)
  - Object Storage Nodes (OSD)
  - MetaData server nodes (MDS)
    - (CephFS only)
  - Gateway nodes (RGW)
    - (Not shown on diagram)
  - Client nodes

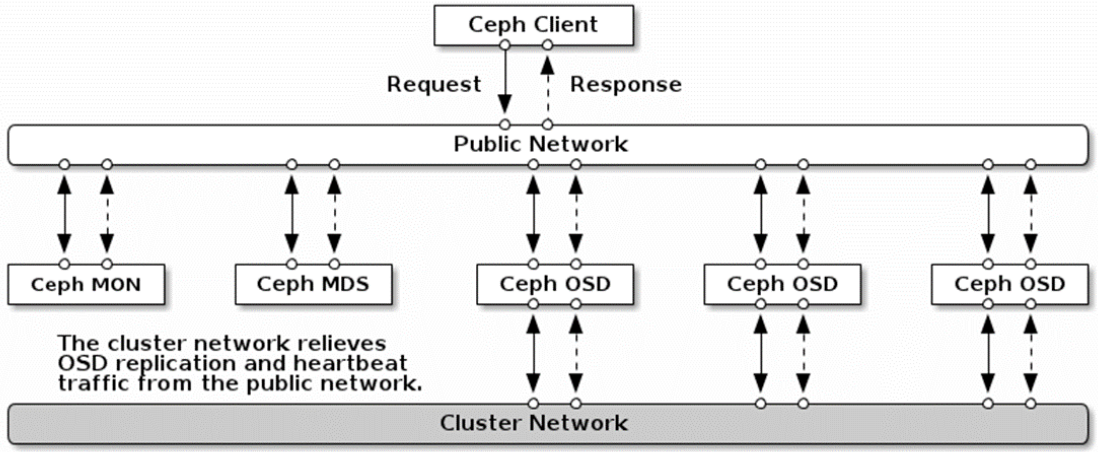


Reference http://docs.ceph.com/docs/master/rados/configuration/network-config-ref/

2010	2015	2020
SDS (Ceph)		

Object Storage - Ceph

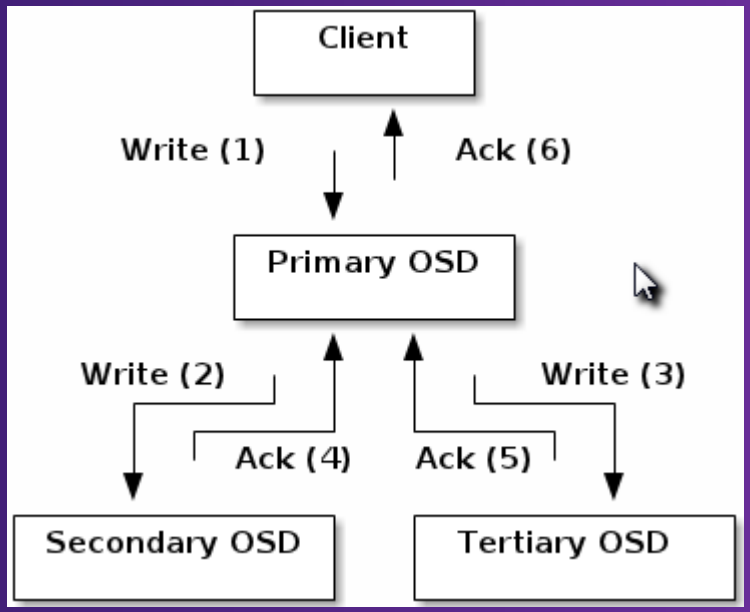
- Server roles can be converged
- All nodes communicate across a Ceph Public Network
  - An optional Cluster Network can be implemented which can be used to offload OSD replication tasks
- Data replication and erasure coding is fully supported
  - Default is replication at the server level but can be modified.



2010	2015	2020
SDS (Ceph)		

Object Storage - Ceph

- A closer look at Ceph
  - Ceph Replication
    - By default three copies are kept
      - Can be changed
    - Synchronous replication – strong consistency
  - Erasure Coding
    - Objects are stored in k+m chunks where k = # of data chunks and m = # of recovery or coding chunks
    - Example k=7, m= 2 would use 9 OSDs – 7 for data storage and 2 for recovery
  - Pools are created with an appropriate replication scheme



STORAGE DEVELOPER CONFERENCE
SDC 21
BY Developers FOR Developers

Virtual Conference
September 28-29, 2021

Evolution of HPC file systems

What it was like in the early days?

Progression towards today?

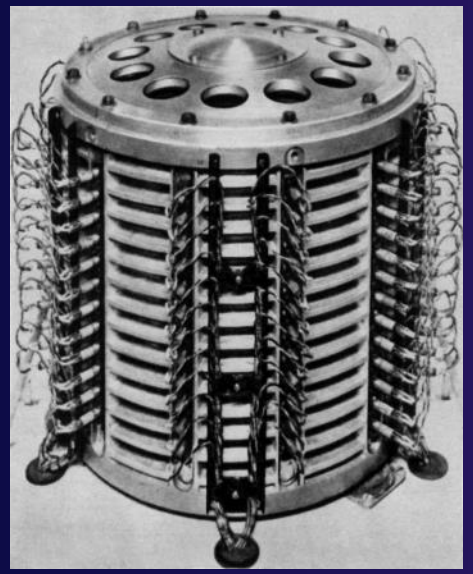
What does the future look like?

Presented by Randy Kreiser, Supermicro

Early days of storing data

Before computers were developed to function on disk operating systems:

- Computer built to run a single proprietary application
- Application had complete and exclusive control of the entire machine
- Application would write its persistent data directly to a disk, or drum, by sending commands directly to the device controller
- Application was responsible for managing the absolute locations on disk, making sure that it was not overwriting already-existing data
- Additionally, writing sequentially to tape drives was used as well

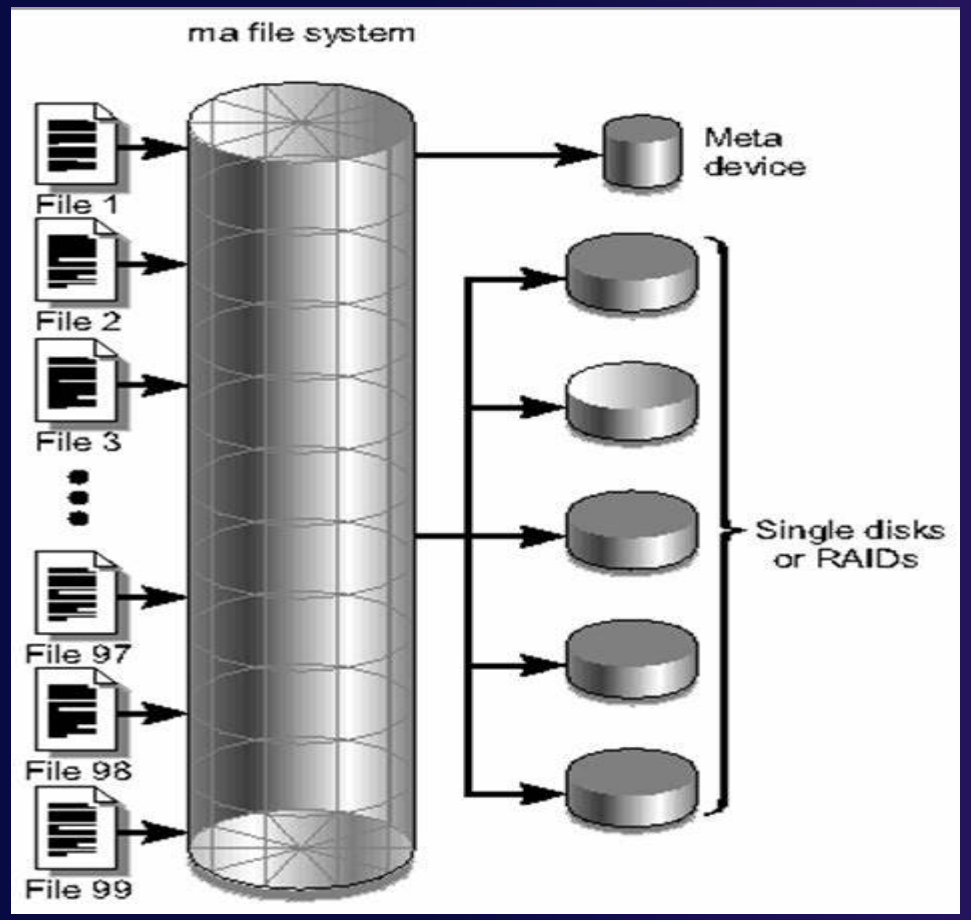


Along comes disk OS & early file systems

Computer systems that could run more than one application required:

- A mechanism to ensure that applications did not write over each other's data
- File systems addressed the over writing problem by adopting standards
- Disk logical blocks were used to track which blocks were free from those that were in use and marking them accordingly
- File systems freed applications from having to deal directly with the storage medium
- File systems allowed applications to create data hierarchies through an abstraction known as a directory
- Applications simply told the file system to write blocks of data to the disk and let the file system worry about how to do it
- A directory could contain not only files but other directories
- Directories could contain their own files and directories and so on

Striped file system progression



Up until now file systems were created on single devices.

Striped file systems and RAID features:

- Ability to use multiple disks for resiliency by adding parity drives to RAID groups
- Added file system performance
- Separating out metadata for better metadata performance and overall file system performance

Progression to parallel file systems

HPC parallel file systems:

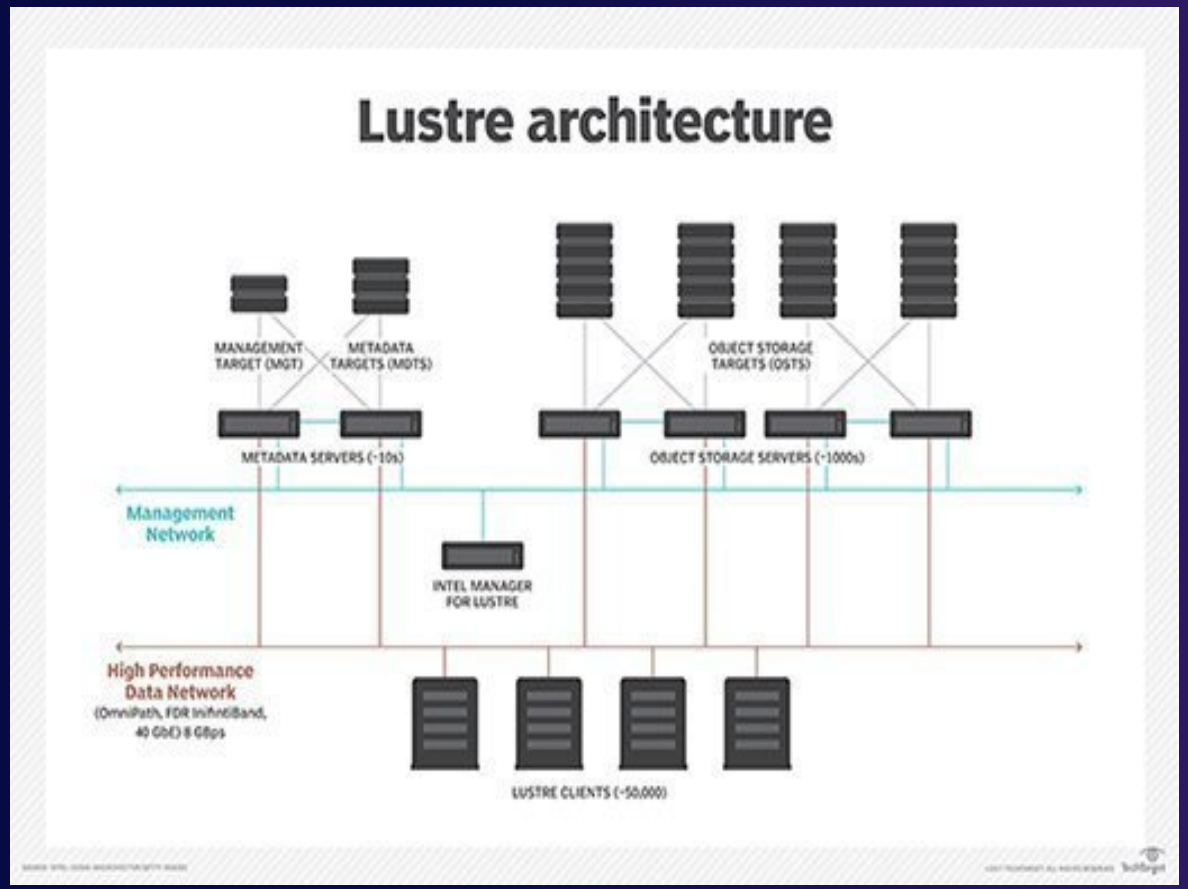
1. Spectrum Scale
   (previously known as GPFS)
2. Lustre
3. BeeGFS

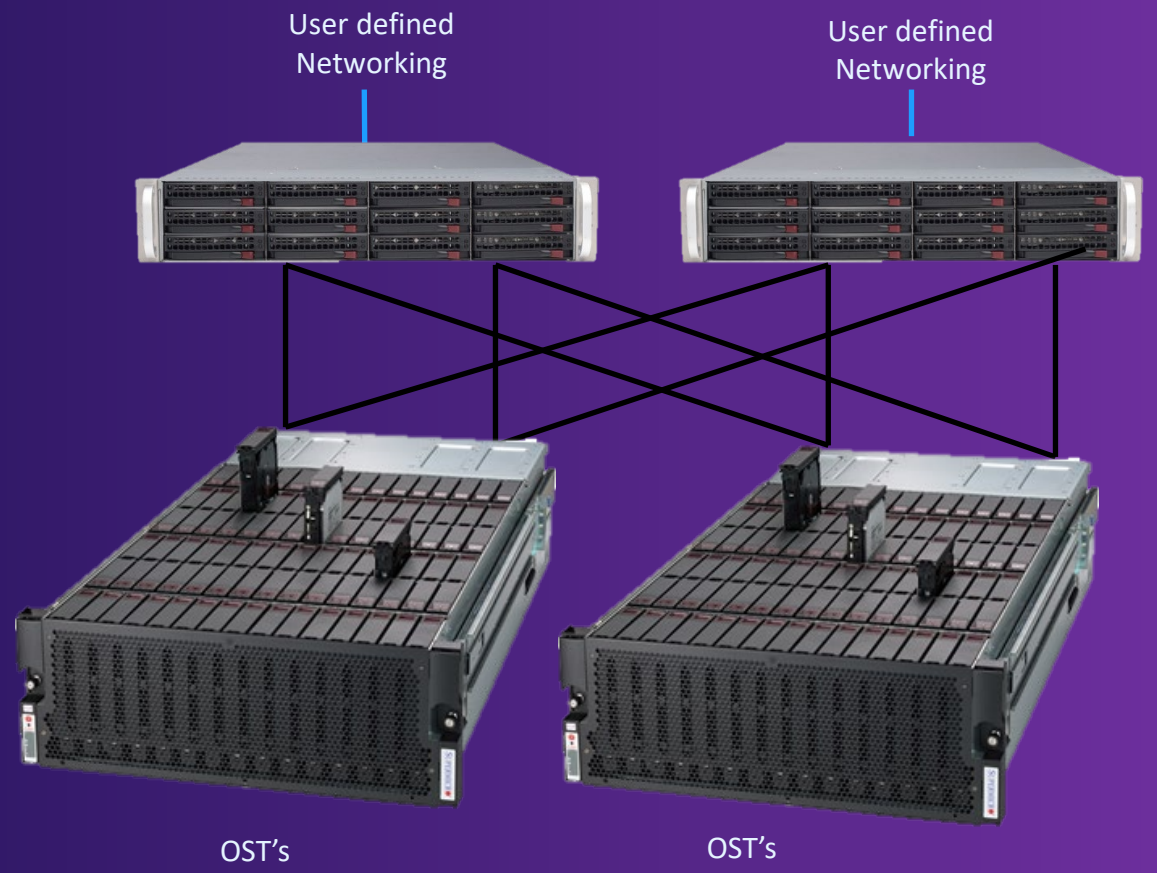
A parallel file system is:

- Software component designed to store data across multiple networked servers
- Facilitate high-performance access through simultaneous, coordinated input/output operations between clients and storage nodes.

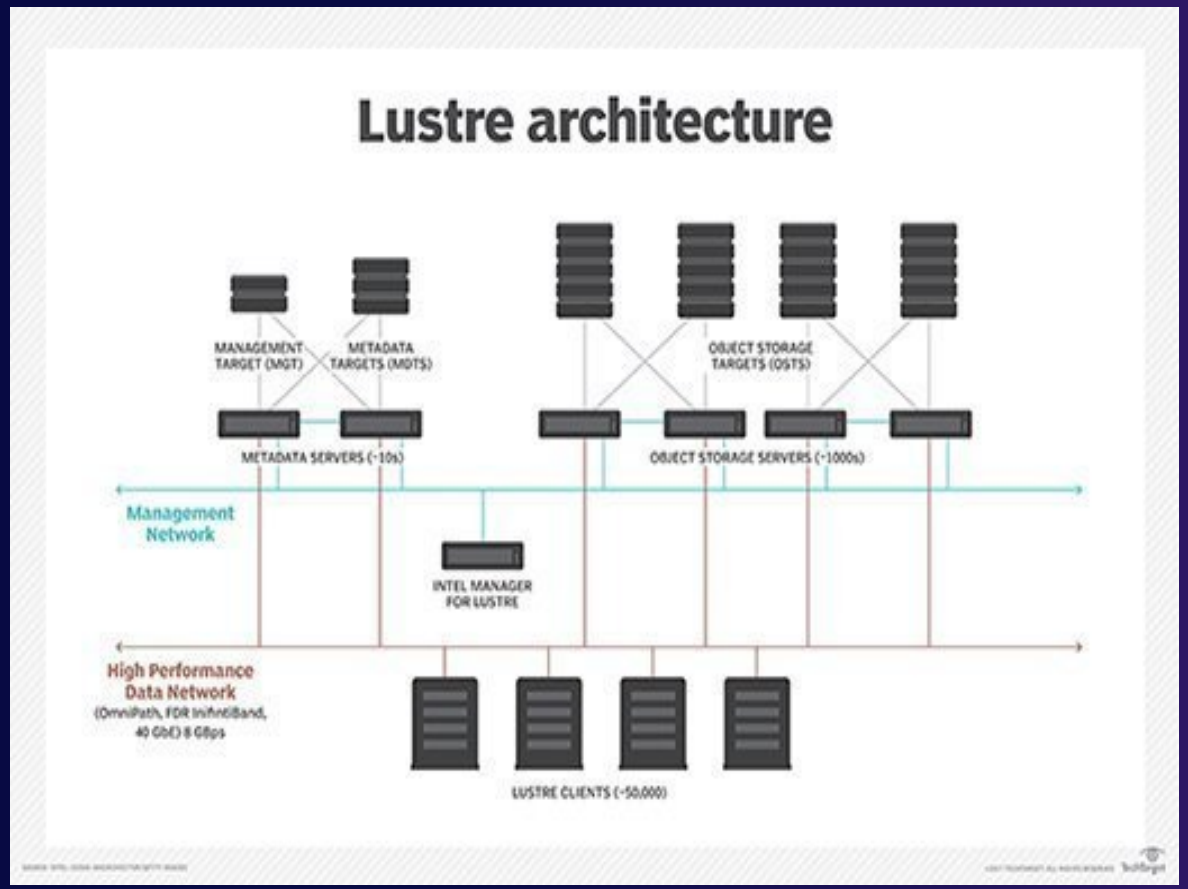
Storing/accessing striped data across storage nodes takes performance to a whole new level.

Standard parallel file system architecture





Standard parallel file system metadata architecture



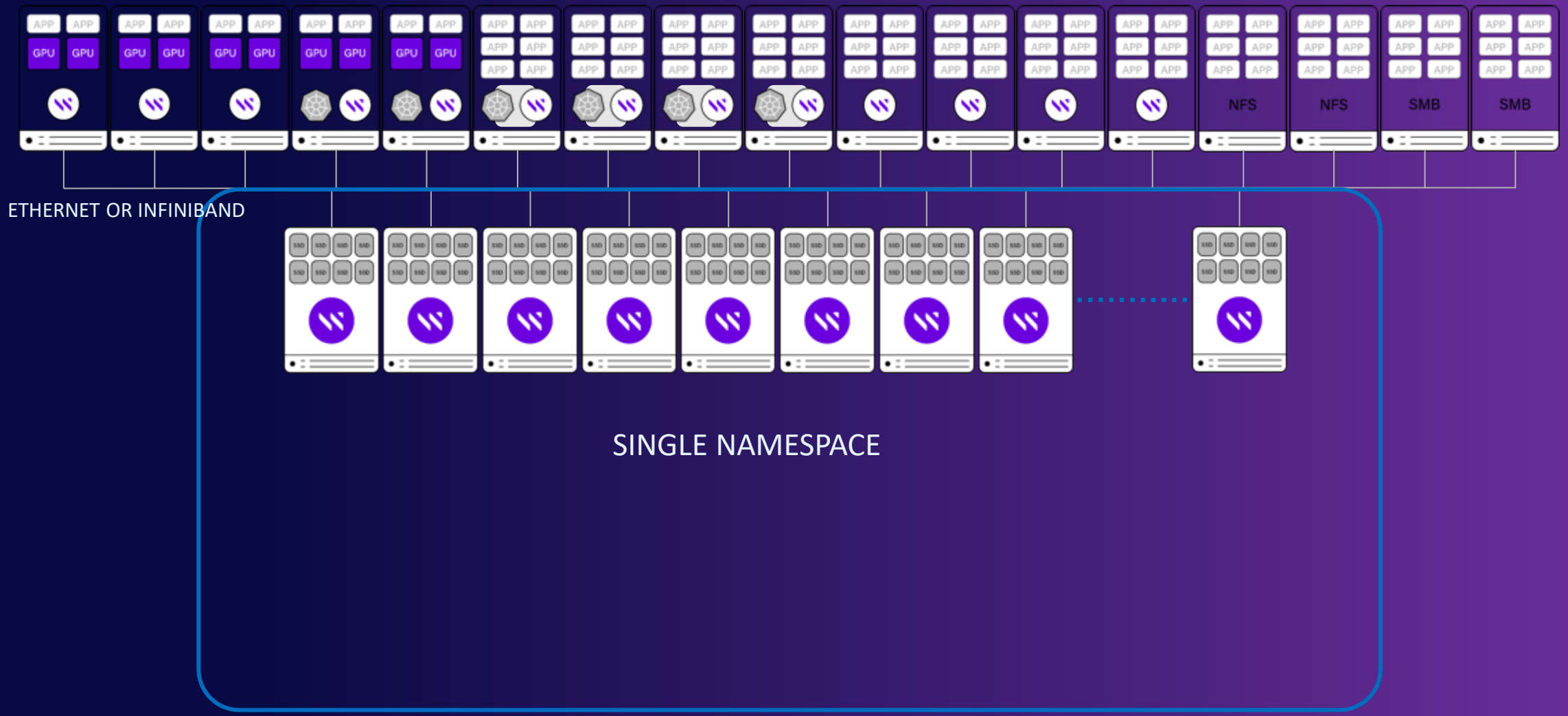
Forward to NVMe-oF parallel file systems

Most legacy parallel file systems overlay file management software on top of block storage, creating a layered architecture that impacts performance.

New high-performance file-based storage solution features must include:

- Highly scalable and easy to deploy, configure, manage, and expand
- Specifically take advantage of high performance NVMe SSD's
- Leverage existing technologies in new ways and augmenting them with engineering innovations within the file system
- Deliver a more powerful and simpler solution that would have traditionally required several disparate storage systems
- Deliver a file system solution with high performance for all workloads (big and small files, and writes, random, sequential, and metadata heavy)
- Designed to run on commodity server infrastructure, not relying on any specialized hardware

NVMe-oF parallel file system architecture



NVMe-oF & Cloud in a single namespace

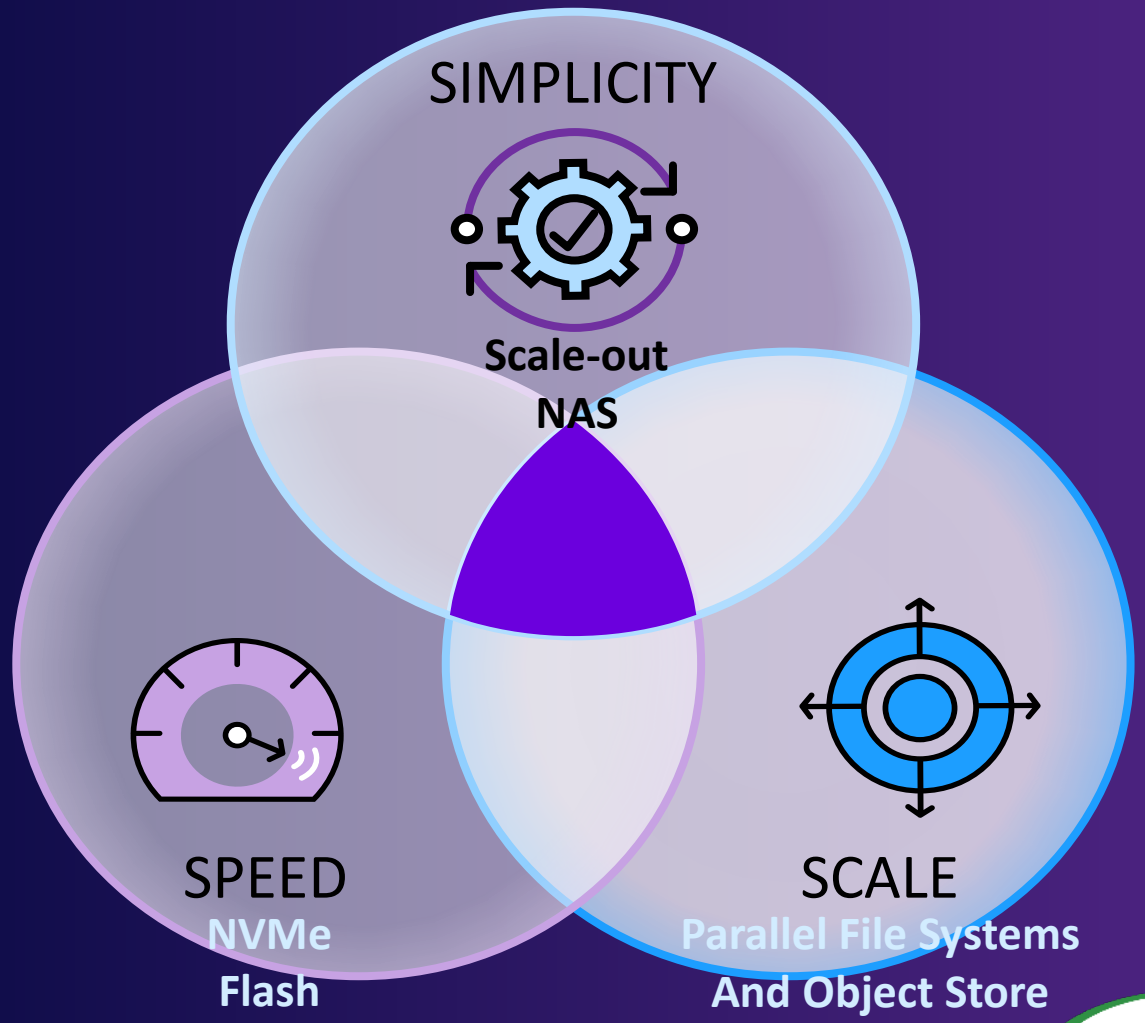
When designing a modern storage solution, a key consideration is to account for the continuing evolution and improvement of technology.

Software-defined storage solutions should accommodate these changes:

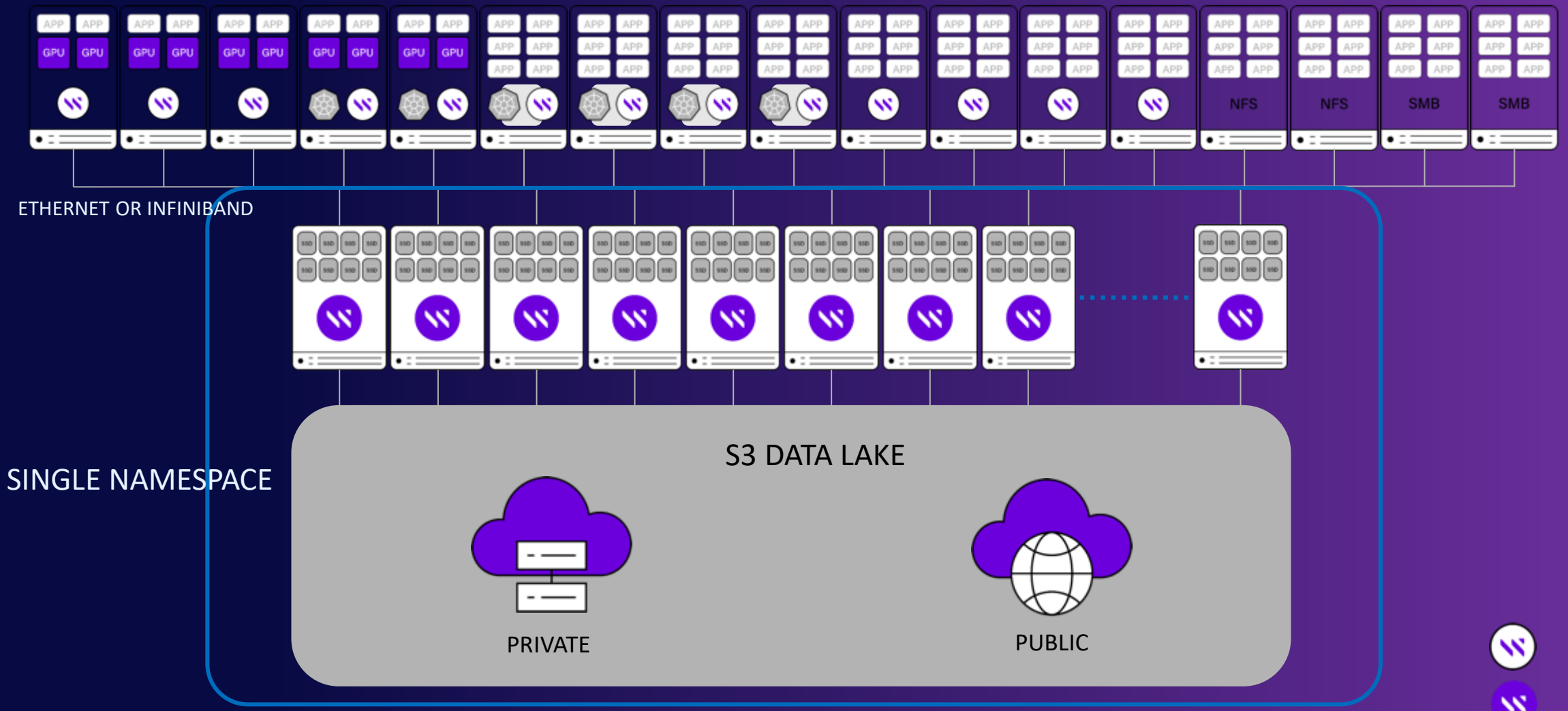
- Must be able to run on commodity server hardware
- Adapt to customer environments
- Add cloud-like agility, scalability, and on-demand performance
- Be simple to deploy and expand fluidly without incurring the typical procurement delays associated with traditional external storage appliances
- Provide a new file system to deliver the performance of all-flash arrays
- Provide the simplicity of scale-out NAS, and the scalability of the cloud in a single architecture

Note: The figure on the next slide provides an overview of the file system value proposition.

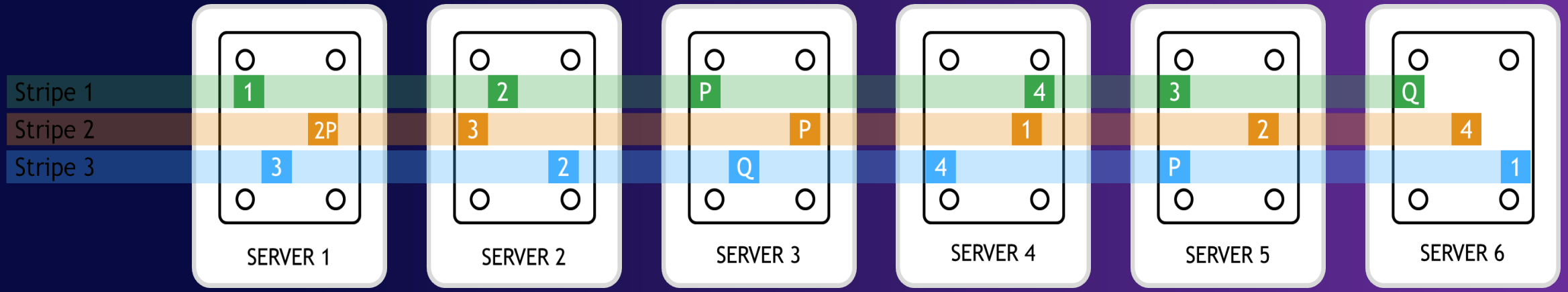
NVMe-oF & Cloud in a single namespace



NVMe-oF & Cloud file system architecture



Erasure coding is the new RAID



- Data protected across storage nodes
- No Performance impact during rebuilds
- Data protected at file level, so only need to rebuild small part of failed SSD/Server
- Smart rebuilds – dramatically faster than traditional RAID rebuilds
- Bigger the cluster, faster the rebuild

What does the future look like

New technologies will continue to change the landscape.

PMEM/Optane will become tier 0 in front of spinning drives and maybe even NVMe.

Dual actuator drives could be an interesting addition to the cloud solution.

400Gb networking is right around the corner which will contribute to the next step up in HPC file system performance.

PCIe Gen5 lurks over the horizon which will provide the opportunity to extract even more performance from each server, especially when combined with 400Gb networking.

Please take a moment to rate this session.

Your feedback is important to us.

Visit us at www.supermicro.com

