Data at Scale

For the most demanding HPC and AI infrastructures

Jean-Thomas Acquaviva
jtacquaviva@ddn.com
18 Oct. 2024

ensiie
PARIS-ÉVRY

ddn

The AI Data Company


DDN is the World's Largest Privately Held Storage Company
Market Leader With Global Presence
10,000
Customers
1,000
Employees
20 Years
of Industry Leadership
10
Technology Centers Around the World

DDN is a Leading Global Provider of Data Storage and Data Management Solutions

Broad Customer Base
11,000 Customers
Strategic Enterprise, Government, Academia

Broad Acceptance
AI, Healthcare, Finance, Cloud, Auto, Research, Manufacturing, Defense

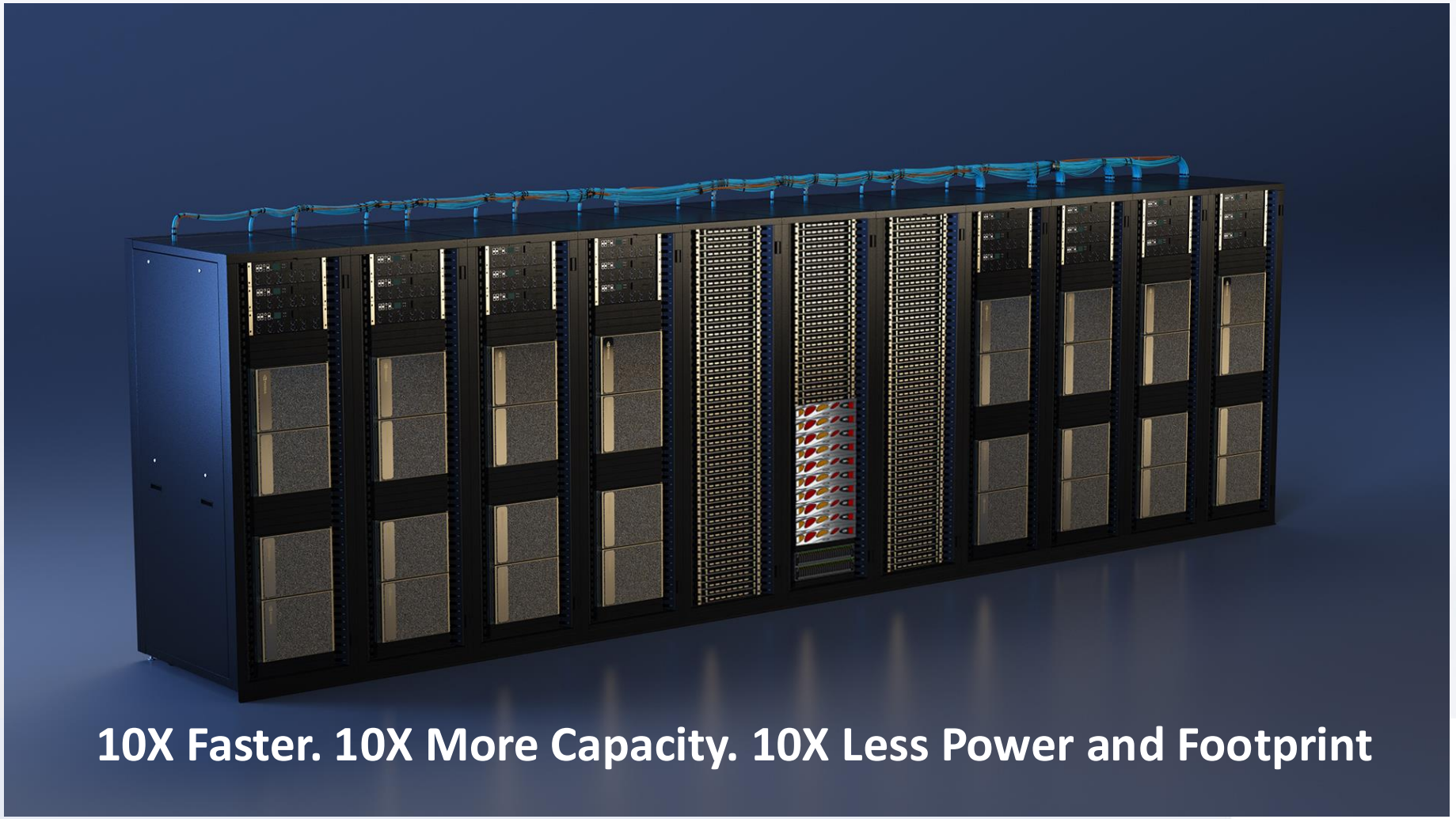
One Powerful Platform
Proven for AI and HPC At Scale in 1000's of Data Centers, Edge, Cloud

World Class Team
1000 Team Members
600 Engineers

Sustained Stability
20+ Years in Business
Long Term Growth

Global Presence
US, Europe, Asia, ME
Customers, Labs, Offices

DDN Makes the Storage Which Drives Advances in AI, HPC, Life Sciences, Finance, Autonomous Cars at Any Scale



We Power More AI Across More Markets Than Anyone Else

More than ~~200,000~~ 500,000 GPUs are Accelerated and Enabled by DDN Storage

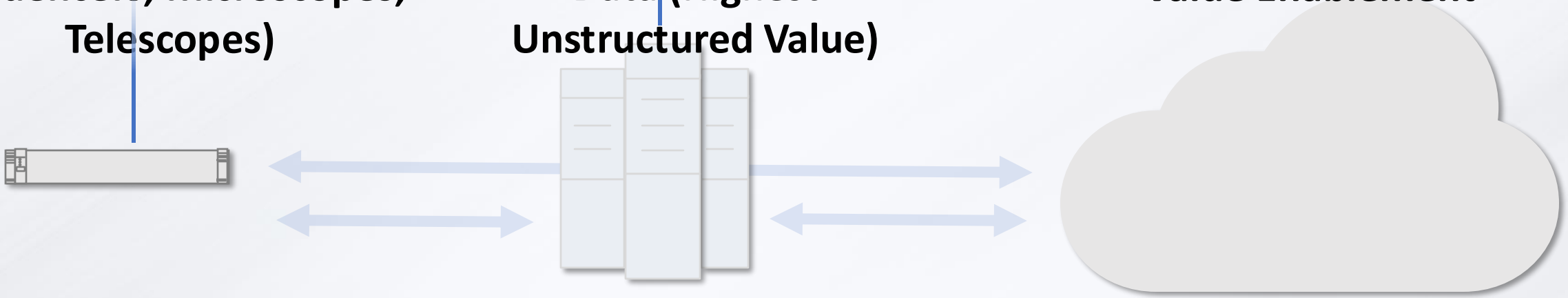
- ✓ Autonomous driving
- ✓ AI chatbots and personal assistants
- ✓ Healthcare
- ✓ Financial Services
- ✓ Manufacturing
- ✓ Energy
- ✓ Defense
- ✓ Public sector and research
- ✓ Generative AI and data analytics

DDN Optimizes Workflows: Edge to Data Center to Cloud

Data Generated at the EDGE in Real Time (Sensors, Cameras, Sequencers, Microscopes, Telescopes)

DATA CENTER Ingests, Processes, Accelerates and Delivers Insight from Data (Highest Unstructured Value)

The CLOUD Completes the Overall Customer Value Enablement



EDGE **DATACENTER** **CLOUD**

HPC is the meeting point between science, software and hardware

At Scale

Real-Time, Many-Core RAID Engine
Full Implementation of Real-Time Storage OS, Interrupt-Free; Massively Parallel I/O Delivery System

High Density Flash
24 QLC SSDs per enclosure
30TB initially, moving to 60TB

Parallel multi-channel rebuilds
Super fast rebuilds for large SSDs

ReACT Cache Management
Real-Time I/O Data-Aware Cache Optimizer, Small IOs use Mirrored Battery Backed Cache (Write-Back), Large & Streaming I/Os Dynamically use Write-Through

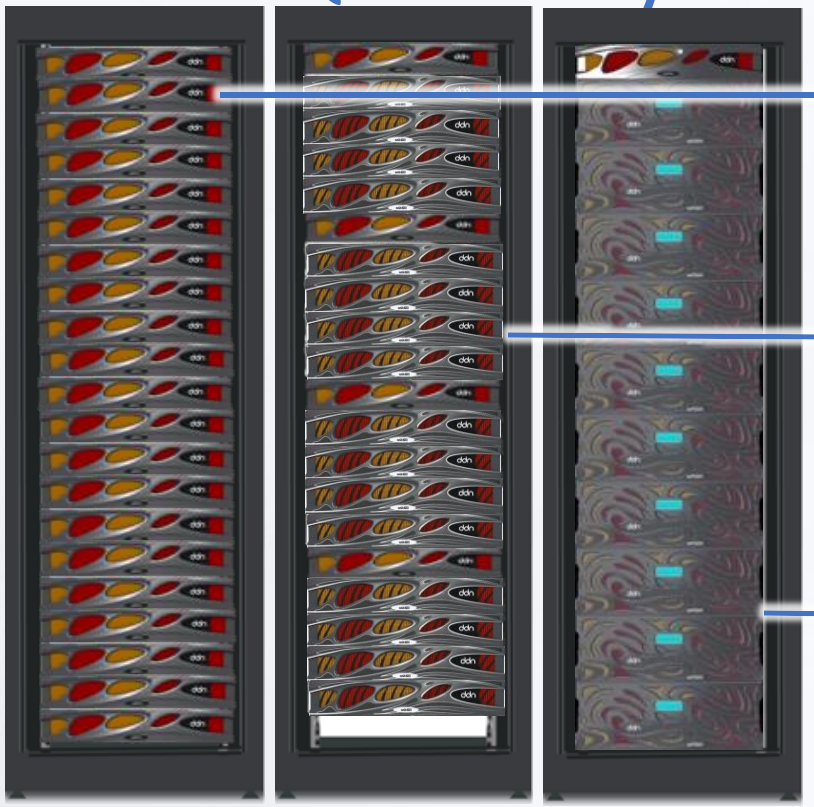
No Single Point of Failure,
Zero Midplane Design

No External Switching - Storage Fusion NMVe Fabric
Highly-Over-Provisioned Backend RAIDed Fabric Withstands Failures of Drives, Enclosures, Cables, etc

Data Protection on disk
T10 PI Type 2

World's Most Performing and Flexible Storage At Any Scale

TLC Flash QLC Flash Hybrid



Best IOPS & Throughput per rack

Up to 70M IOPs in a single rack
1.8 TB/s of read throughput
1.4 TB/s of write throughput
16PB of Flash

Best Price per Flash TB

Up to 14M IOPs in a single rack
360 GB/s of read throughput
240 GB/s or write throughput
Up to 26PB of Flash

Best Price per TB

20PB per Rack
90 GB/s of read throughput
65 GB/s of write throughput

GTC 2024

“And the Best Storage is ?...”

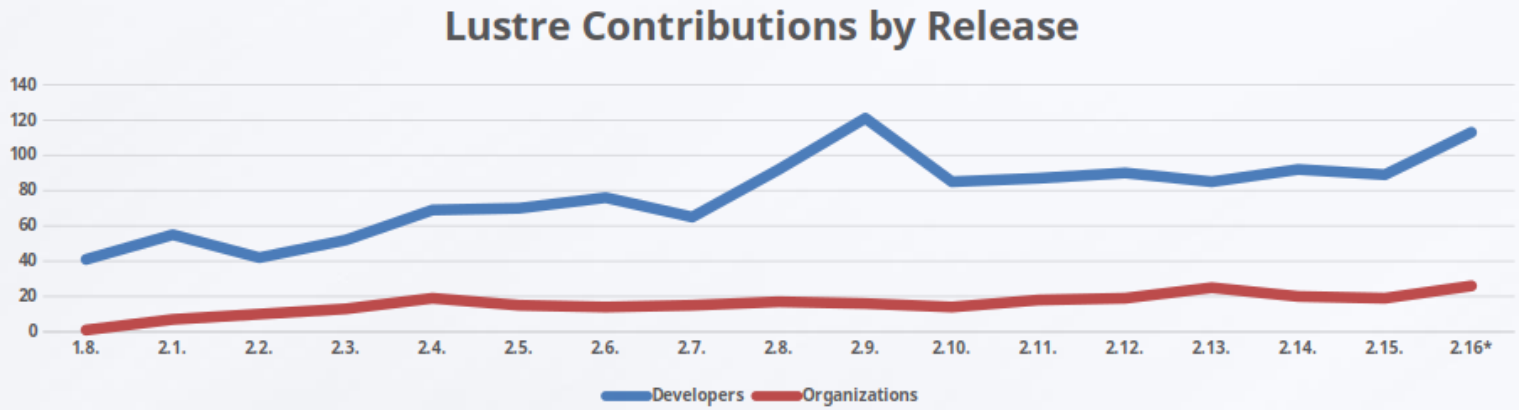
- Paul, President & Co-Founder ddn

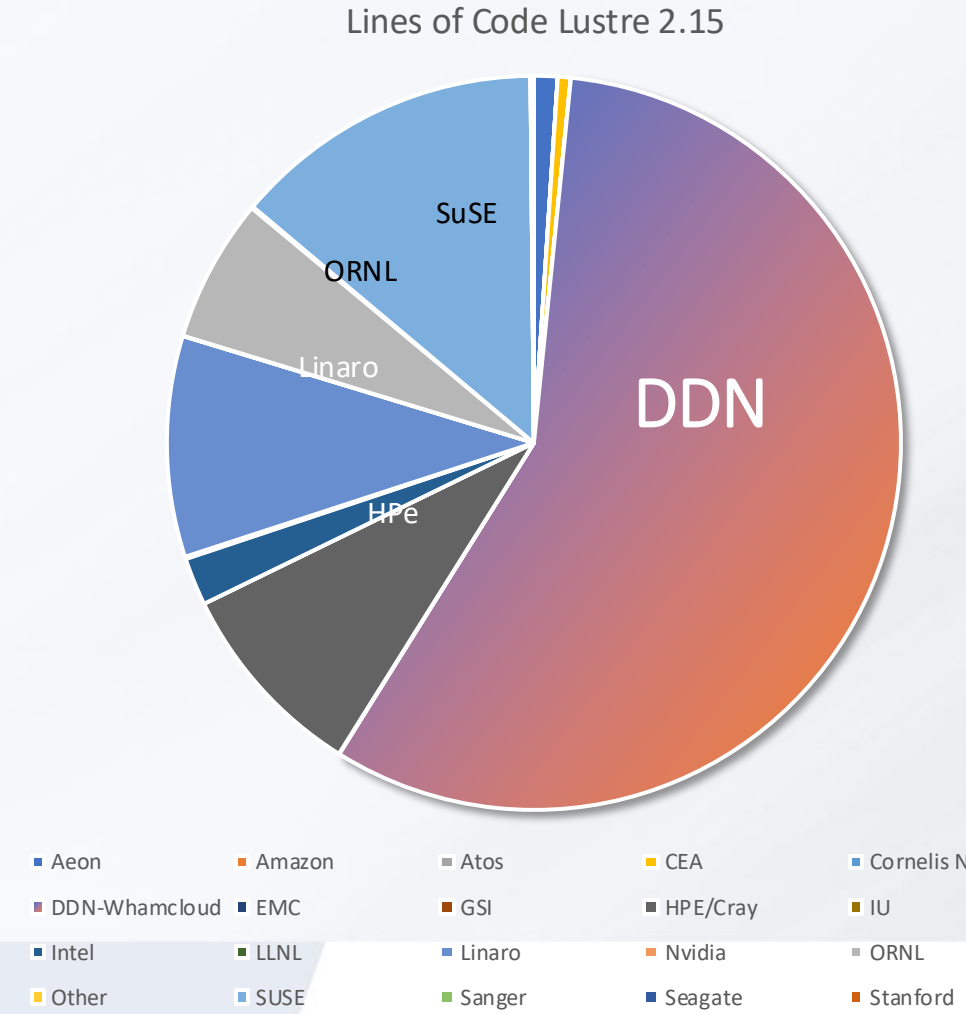
“DDN !”

- Jensen Huang - President NVIDIA


Select DDN European Customers at Scale
DDN OPERATES IN EUROPE SINCE 2001

Lustre Open-Source Parallel FRile system (OpenSFS)





- Designed for HPC: data extension of the compute platform
- OpenSFS provides overall directions and a forum for discussion among users
- DDN is the lead contributor to Lustre
- User meetings in Europe organized by EOFS
- User meetings in Asia organized by DDN

DDN European Collaboration Framework

A – EuroHPC Infrastructures powered by DDN

Leonardo


Meluxina

Discoverer

Vega


Deucalion

B – EuroHPC Research Programs and DDN

- EuroHPC design of next-Gen IO system
- AI-automated features extractions from Satellite Images
- Nuclear Fusion code optimization

C – DDN R&D Spending in Europe

- Significant portion of our WW turnover is spent on R&D
- 25 persons R&D in EU
- France focused on Software Platforms

HPC for science: AI at cnrs

Centre National de la Recherche Scientifique

Jean Zay - World Class AI system

a BigScience initiative

BLOOM

176B params · 59 languages · Open-access

1000,000 GPU hours of learning

IDRIS is driving excellence in scientific research for modeling and intensive numerical computation which requires seamless scaling and enterprise resilience.

Supplied as a service infrastructure, this is next generation class systems used for the refinement of AI algorithms at large scale.

EOS NVIDIA Flagship System

576 DXH: 4608 H100 GPU

NDR400 IB Compute and Storage

Storage:

48 AI400NVX2

EXAScaler 6

12 PB flash

4.3 TB/s Read

3.1 TB/s Write

DDN **Hot node** for Accelerated AI Training

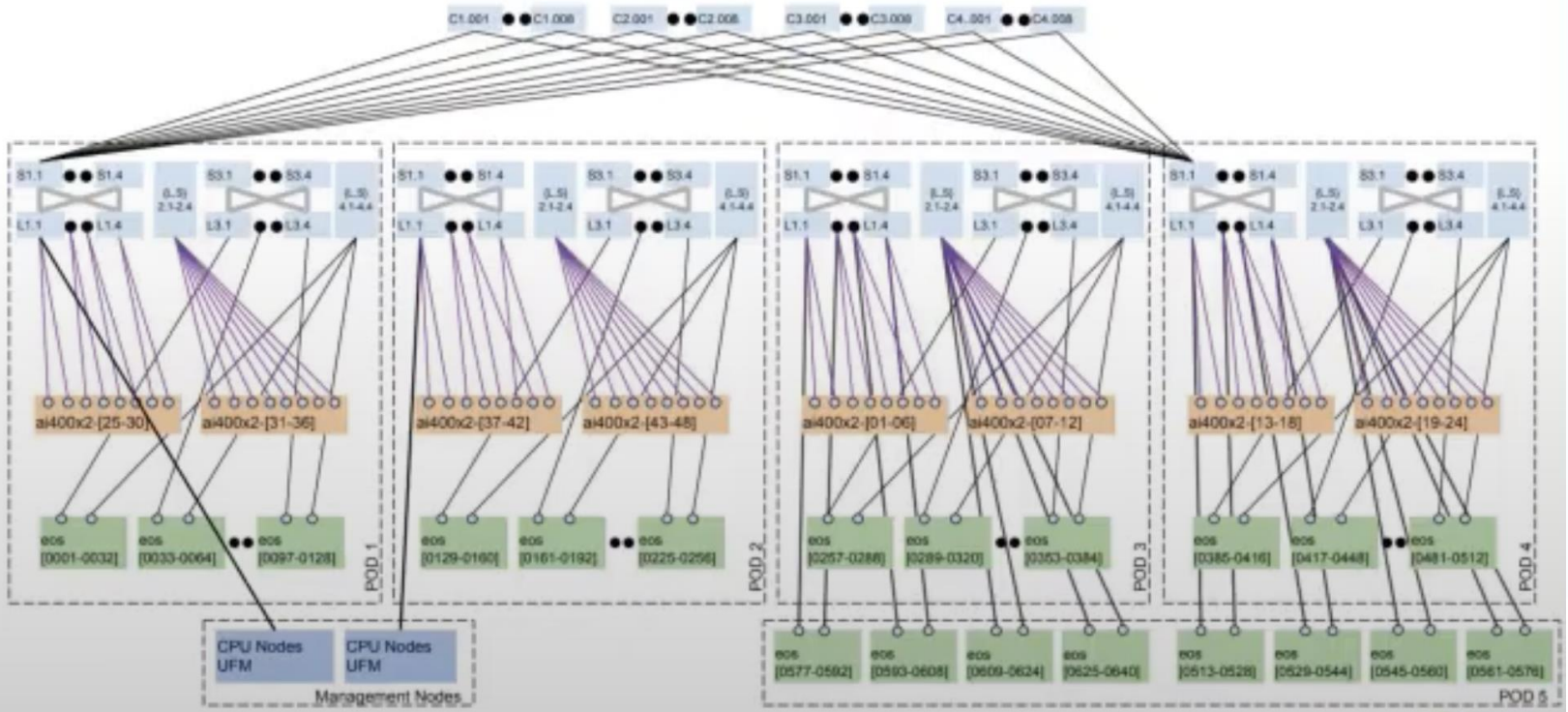
EOS IS THE THIRD-GENERATION FLASGHIP DGX SUPERPOD

EOS looks very much like a fat-node fat-tree HPC system

Hierarchical Design: 32N Scalable units, 128N PODs

- DGX H100: 8xNDR400 ports for compute and 2x NDR400 port storage
- Separated, non-blocking fabrics for both compute and storage
- Three-level fat tree topologies

- 48 DDN AI400X2 storage appliances connected with HDR IB
- Appliance IB connections are interleaved across 4 PODs
- Target a minimum performance of 2 TB/s (read) to support DL training at scale



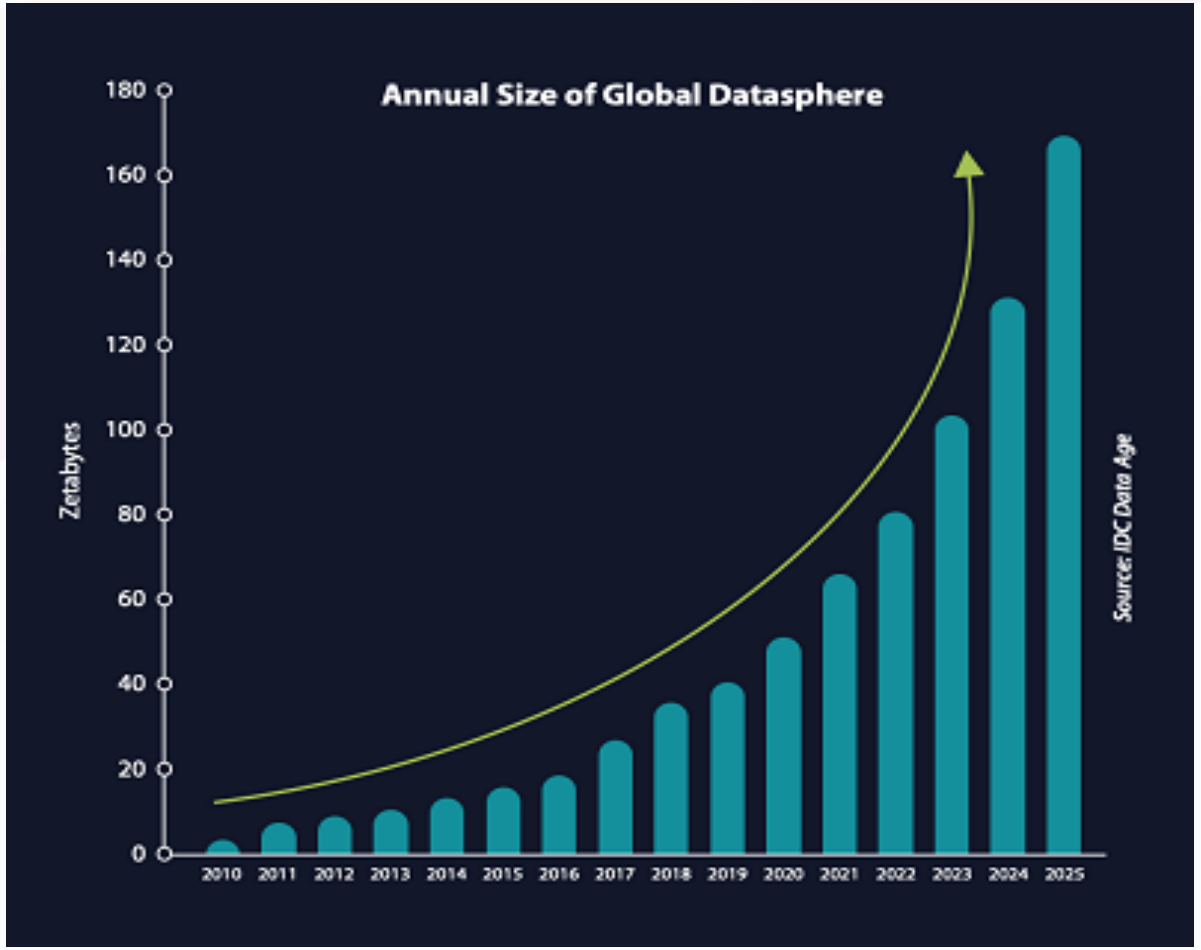
Understanding Data at Scale: Commodity and Scarcity

2024: Data acquisition devices are ubiquitous

2024: Computing capabilities are ubiquitous

2024+: Data management at scale is scarce

As computing is becoming a commodity, the bottleneck to time-to-science has shifted from compute to data management.



Foreword and Context Framing

> A file system lasts 5 years, data live forever

Choose carefully :-)

Data Solution: Requirements

Storage systems have to meet to multiple technical requirements. HDD, SDD and PCM offers order of magnitude of difference in latency and bandwidth. Some of these technologies have limited endurance.

Performance capabilities

- Capacity, amount of data to be stored
- Bandwidth, rate for the data ingestion / egest
- IOps, control operation (metadata), creation, deletion, search

- **Functional aspects**
  - Data protection, what happens if a device fails?
  - Data security, what happens in case of inappropriate behavior
  - Interoperability, will this work with the existing set-up / future extension

Architecture of a Data Solution: Relevance

Relevance of a storage solutions ultimately depends on users 'needs and working habits

- **Diversity in data usage**
  - RMWM – Read Many, Write Many
    - Hot data and active data
  - WORM – Write Once, Read Many
    - Hot data could be moved to a capacity tier if prefetching
  - WORO – Write Once, Read Once
    - Cold data, can be archived
  - IO patterns (small, large accesses, random, linear accesses, shared, private accesses)
- **Data API**
  - Object store / file systems
  - High level libraries
  - Specific data format (SEGY, GDAL, DICOM, GRID2)

Data Solution: Landscape

Data systems belong to a technical landscape, with its natural shift, trends and opportunities

- **CAPEX,** Cost per GB, full flash, heterogeneous architectures
- **OPEX**, Cost of operating and managing the solution
- **Forward looking**, betting on the wrong set of technologies can lead to missed opportunities, accelerated obsolescence (swift vs S3)

Characteristics of Storage for AI & HPC

How can organizations accelerate the implementation of large-scale systems?

Capacity	Throughput	Latency	24x7
Data expands to meet capacity, especially for **AI LLM**	We need to eliminate system bottlenecks to **maximize data throughput**	We need to streamline data paths to **reduce data latency** within systems	AI & HPC systems are expensive, and we need to **maximize utilization**

Differences between AI & HPC workloads?

AI & HPC Workloads have different characteristics

HPC	AI	Commercial
• Write oriented • Very large datasets • Very high throughput • Regular access patterns • Datasets are uniform • Only a few data files • Task-oriented	• ~~Read oriented~~ • Actually*, it is 50/50 • Very large datasets • Very high throughput • Irregular access patterns • Sampling and Labelling • Many small files or streams • Workflow oriented	• Relatively small datasets • Interactive throughput • Irregular access patterns • Sampling and Labelling • Wide range of files & types • Interactivity oriented

Machine Learning is Write and Read Intensive

- Analysis of over 23,000 Machine Learning Jobs

"Most ML jobs are perceived to be read-intensive with a lot of small reads while a few ML jobs also perform small writes."

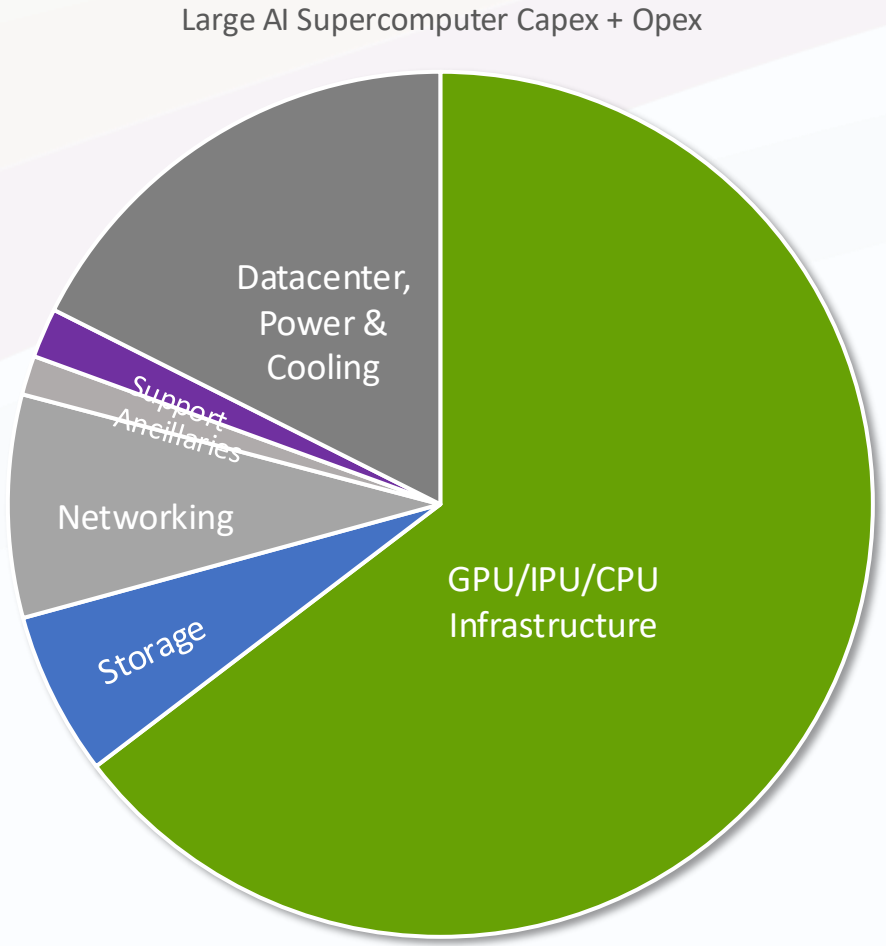
"Our study showed that ML workloads generate a **large number of small file reads and writes**…"

Characterizing Machine Learning I/O Workloads on Leadership Scale HPC Systems
https://arnabkrpaul.github.io/publications/mascots21.pdf

Zoom on AI data path

Spend is Large, Risks are High. How can Storage Help?

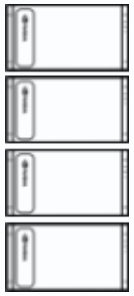
- A Storage System typically represents 5% of the 3 year Capex & Opex budget of an AI system for Deep Learning/LLM training
- IO Wait and associated elements of the training process can consume up to 43%[1] of runtime
- How can the efficiency architecture and consumption of storage resources impact overall productive output of this System?



1 (Maeng et al., 2021).

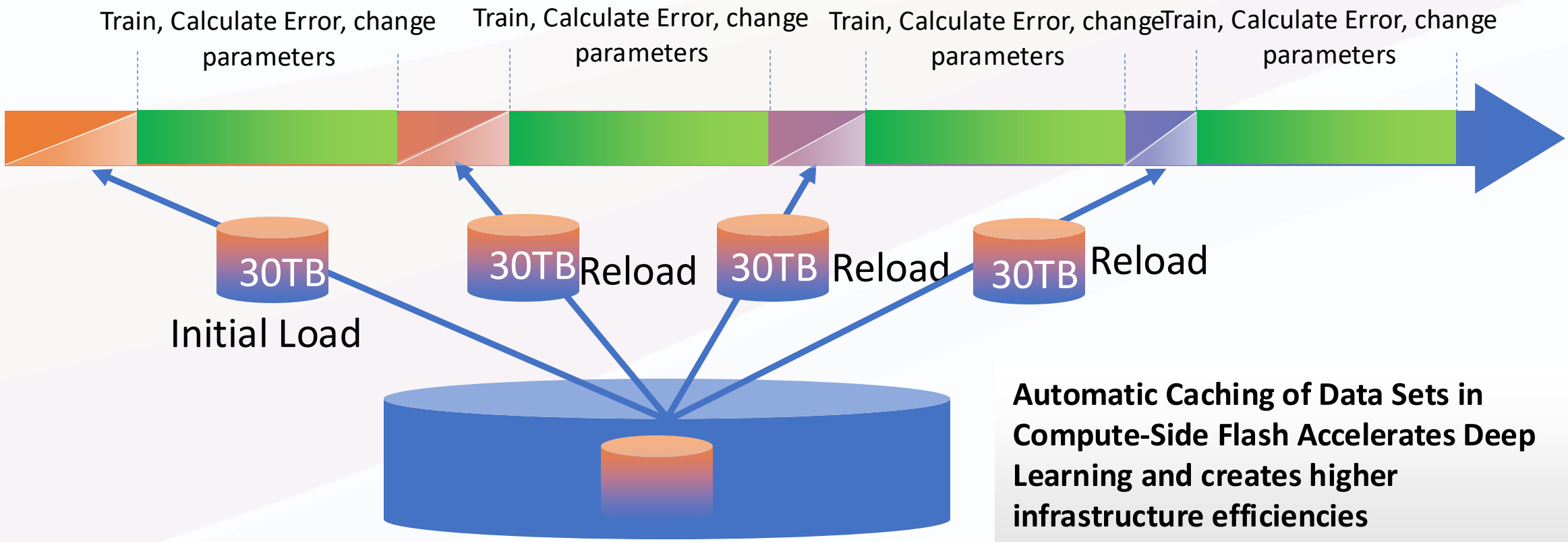
What does the End-to-end data journey involve?

Example: Life Sciences – Drug Discovery Application

CAPTURE	PREPARATION	PROCESS	ANALYZE	ARCHIVE
Millions of digital images captured	Cleansing and labeling of data	 Model training Computation and analysis	Examine & Explore results for action	Store data for long-term compliance

End-to-end data journey

Accelerates Multi-Epoch Training and Improves Efficiency



Automatic Caching of Data Sets in Compute-Side Flash Accelerates Deep Learning and creates higher infrastructure efficiencies

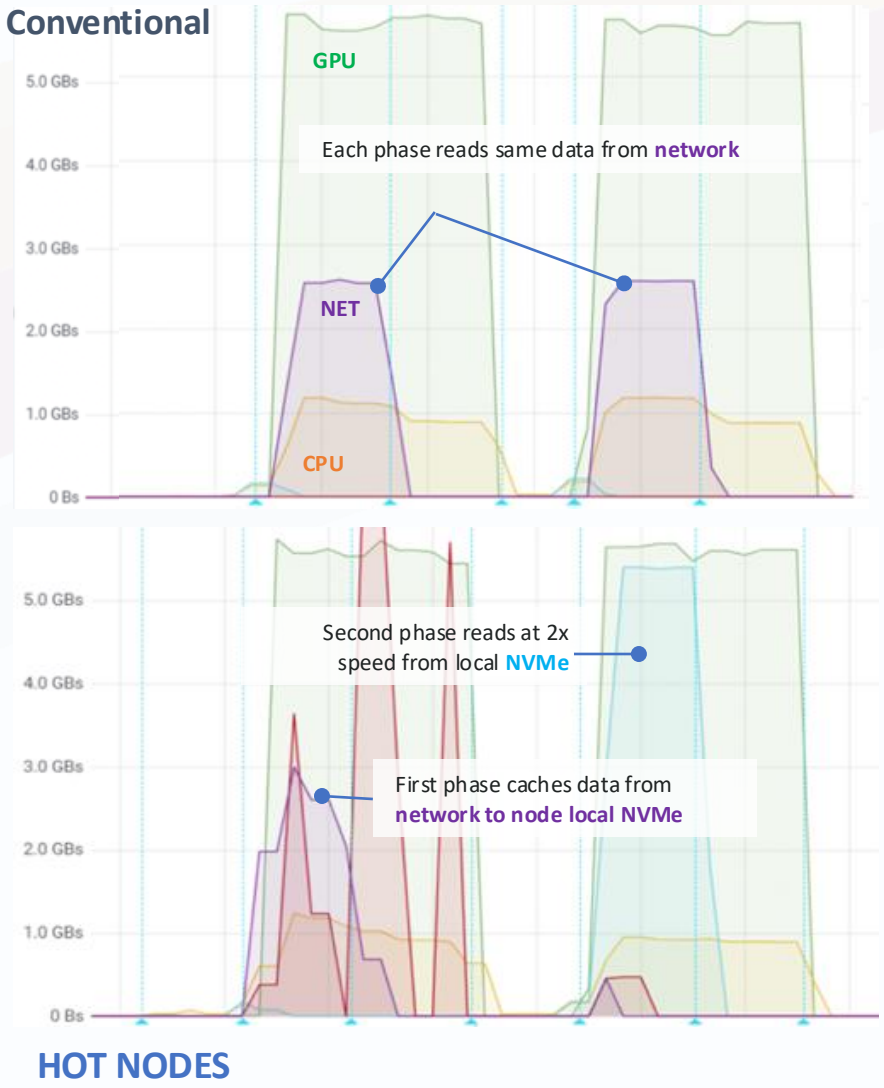
Removing IO bottleneck Improve Time to Solution

By reducing storage wait time for Load, hugely reducing wait time for checkpoints, and enabling more checkpointing to eliminate lost time of recomputation, DDN reduces data center runtime by 5-12%, delivering higher productivity to LLM's and Generative AI.

Optimizing Multi-Epoch Training

- Without DDN Hot Nodes technology, Multi-Epoch Training consumes storage and network bandwidth with every GPU systems repeatedly pulling data.

- With DDN Hot Nodes, we automatically cache data sets on internal NVMe devices, freeing the network and storage from load and accelerating the whole training process



DDN Accelerates the Thousands of Checkpoints Needed in AI

Prediction Accuracy – Improve accuracy by lowering learning rate from a checkpoint

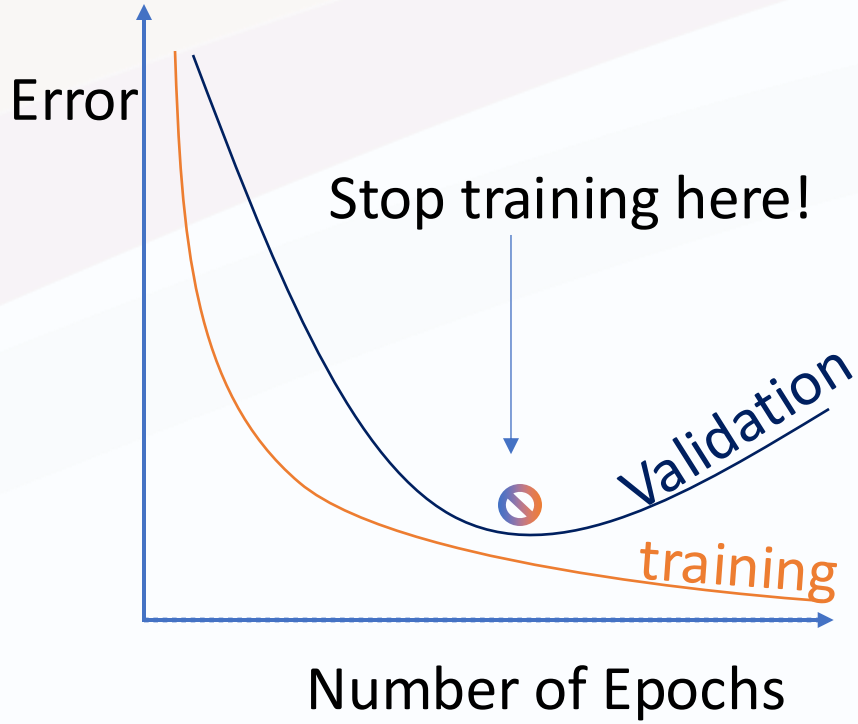
Multi-System Training - continue training model across different nodes or clusters/cloud

Transfer Learning – if goals change, start afresh from a checkpoint

Better Fine Tuning - pick out less trained states to restart new experiments

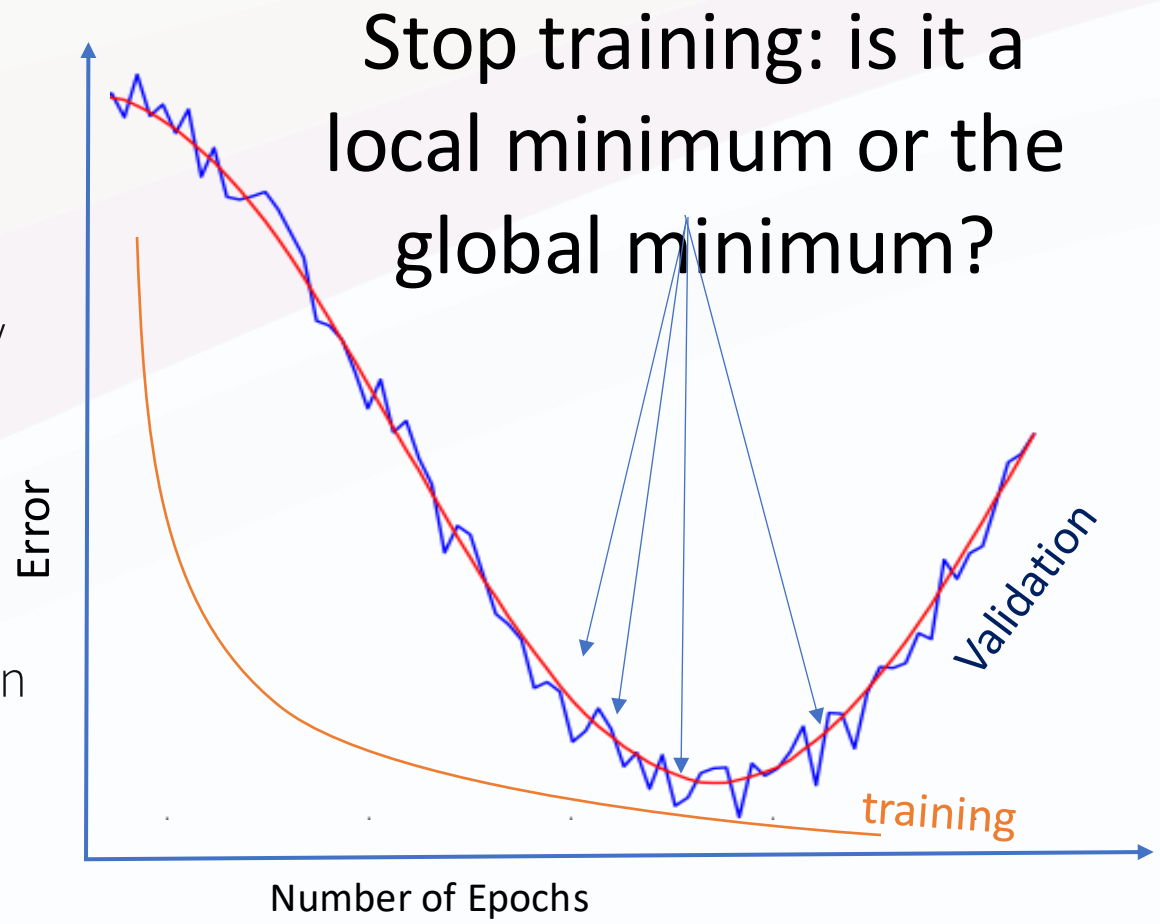
Early Stopping - For large models, without sufficient regularization, the error on the evaluation dataset can start to increase.

- → need to go back and export the model that had the best validation error.



Checkpoints is intrinsic to Deep Learning Training

- **Non-linear convergence** – local minimum exist where the next epoch degrades accuracy but on the long run accuracy can still be improved
- **Over-fitting to detect Sweet-Spot –** some overfitting is mandatory to ensure that the global minimal has been reached
- **Rolling-back to the Global minimum** – once the detection of the global minimum has been assessed, rolling back to correct model state requires parsing the checkpoint history



Some DGX H100 Customers Reference

THE AI DATA COMPANY

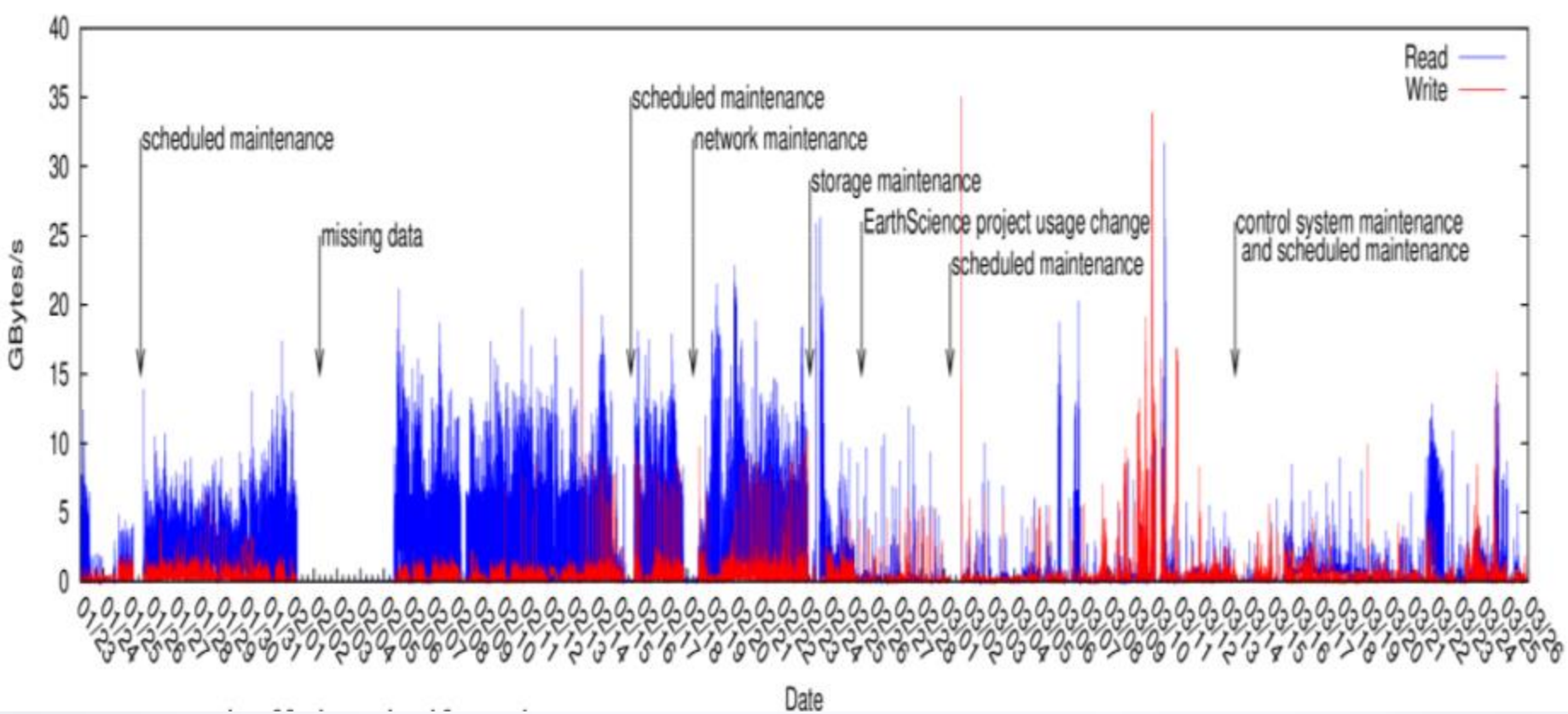
Example deployments at different scale

	NVIDIA - EOS (US)	Customer B (EMEA)	Customer C (EMEA)	Customer D (ASIA)	Customer E (US)	Customer F (US)
DGX Systems	**512 DGX H100**	**160 DGX H100**	**127 DGX H100**	**64 DGX H100**	**32 DGX H100**	**16 DGX H100**
DDN Appliances	**48 AI400X2**	**12 AI400X2**	**30 AI400X2**	**9 AI400X2**	**3 AI400X2**	**2 AI400X2**
Useable NVME Capacity	12 PB	6 PB	7.5 PB	2.25 PB	150 TB	500 TB
Aggregate Shared Read	4.3 TB/s	1 TB/s	2.7 TB/s	810 GB/s	270 GB/s	180 GB/s
Aggregate Shared Write	3.1 TB/s	780 GB/s	1.95 TB/s	585 GB/s	195 GB/s	130 GB/s
Per GPU Shared Read	1 GB/s	781 MB/s	2.6 GB/s	1.5 GB/s	1 GB/s	1.4 GB/s
Per GPU Shared Write	750 MB/s	609 MB/s	1.8 GB/s	1.2 GB/s	750 MB/s	1 GB/s
Per DGX Shared Read	8.5 GB/s	6.25 GB/s	21.25 GB/s	12.7 GB/s	8.5 GB/s	11.25 GB/s
Per DGX Shared Write	6 GB/s	4.8 GB/s	15.4 GB/s	9 GB/s	6 GB/s	8 GB/s
Per DGX Useable NVME Capacity	23 TB	37.5 TB	59 TB	35 TB	4.7 TB	31 TB

HPC workload used to be different from AI workload

99% of time IO system stressed less than 33% of its peak bandwidth
70% of time IO system stressed less than 5% its peak bandwidth

Mesures au Argone National Lab.



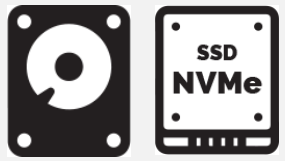
"Understanding and Improving Computational Science Storage Access through Continuous Characterization"
PHILIP CARNS et al.
Argonne National Laboratory
2011, Journal Proceedings of 27th IEEE Conference on Mass Storage Systems and Technologies

Introduction to Storage Technologies

The Storage Landscape is Wide and Deep

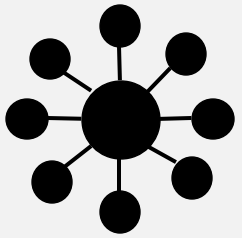
MEDIA & INTERCONNECT

- Magnetic media
  - Disk, Tape
- Solid State
  - SDD, NVMe, etc
- Interface
  - SATA, SAS, SCSI, PCIe



INTERCONNECT

- DAS vs SAN vs NAS
- Network Attached
  1. Ethernet Switches
  2. InfiniBand Switches
  3. RDMA (Rocke)



PROTOCOLS

- Enterprise vs Performance
  - SMB
  - NFS
  - S3
  - LNET

FILESYSTEMS

- Serial
- Parallel

lustre®

exa6

Recap - Characteristics of Storage for AI & HPC

How can organizations accelerate the implementation of AI systems at scale?

Capacity	Throughput	Latency	24x7
Data expands to meet capacity, especially for **multi-dimensional models**	Eliminate system bottlenecks to **maximize data throughput**	streamline data paths to **reduce data latency** within systems	AI & HPC systems are expensive: **maximize utilitization**

Capacity – Is Scaling Easy?

Not always as simple as just adding more storage devices

Capacity

Data expands to meet capacity – Capacity as you grown

More storage capacity may also need more:

- More network infrastructure
- More floorspace
- More power & cooling
- More time to transfer
- More capacity for archive…

Throughput – More than just Device Performance

Each element of the datapath needs to be considered to avoid bottlenecks

Throughput

We need to eliminate system bottlenecks to **maximize data throughput**

Parallelism is a key multiplier

Devices	Networks	Protocols	Clients

Increasing **device speed** is **expensive**,
and may not deliver the desired impact

Increasing **parallelism** can deliver throughput, using **cost-effective** devices and media.

Latency – How do we streamline data paths?

Latency is additive – and dominated by the slowest latency in the data path

Latency

We need to streamline data paths to **reduce data latency** within systems

Traditional IO Path

CLIENT → HCA/NIC → SWITCH → HCA/NIC → SERVER → FILESYSTEM → HBA → SAN SWITCH → HBA → STORAGE

Streamlined IO Path

CLIENT → HCA/NIC → SWITCH

Latency can also be **DECREASED** by:

- Server-side data caching
- Client-side data caching
- RDMA to bypass internal buffers

Latency can also be **INCREASED** by:

- Additional data hops
- Internal replication
- Additional processing

24x7 – Continuous, Concurrent Access

How do we ensure that all application can access simultaneously, with ***no performance impact?***

24x7

AI & HPC systems are expensive, and we need to **maximize utilization**

For a $100M system lasting 5 years what is the cost of 1 day of downtime?

$20M / year = $55K / day

Parallel Filesystem Architecture

What is a Filesystem?

Controls how data stored and retrieved

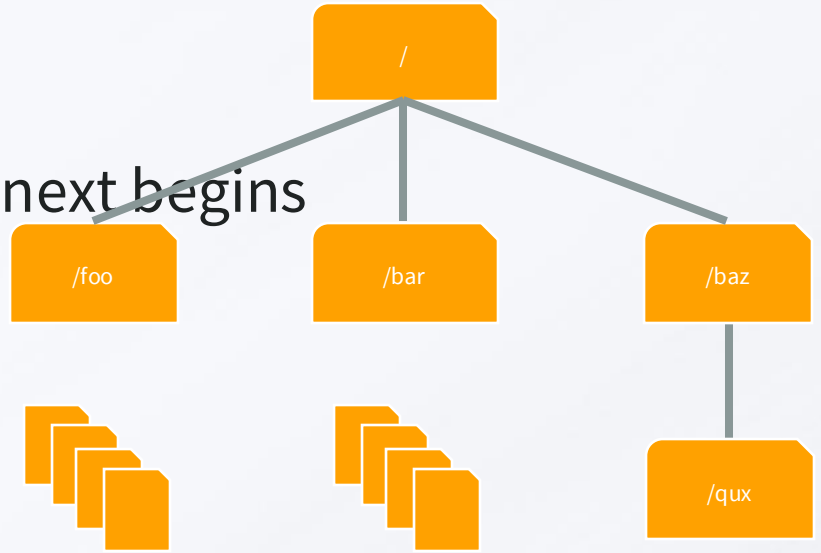
Without a Filesystem, data would have no structure

No way to tell where one data object stops and the next begins

Filesystem separates data into named objects

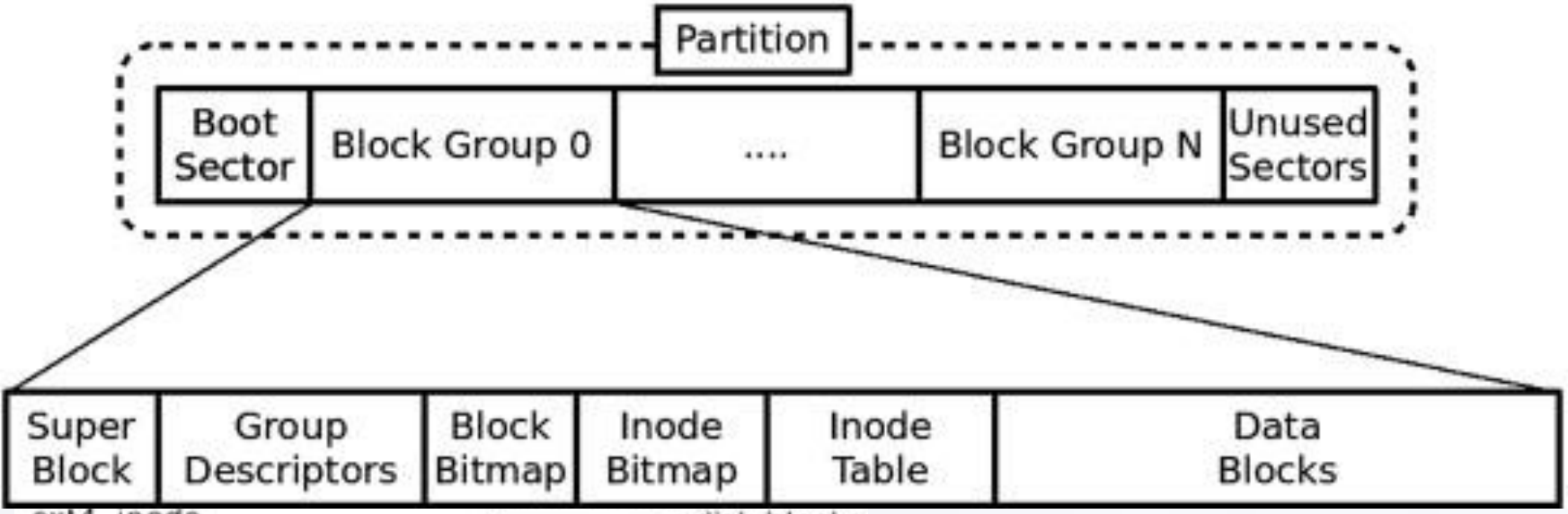
Data easily isolated and identified

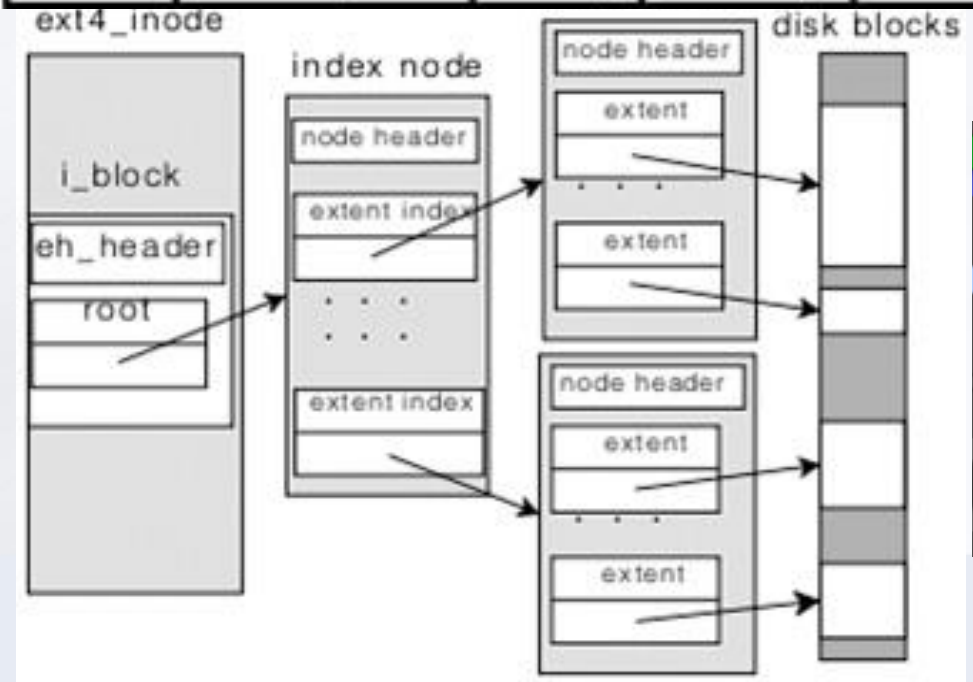
Each group of data called a “file”



Structure and logic rules used to manage groups of data and names called “Filesystem”

Filesystem Internal Structure



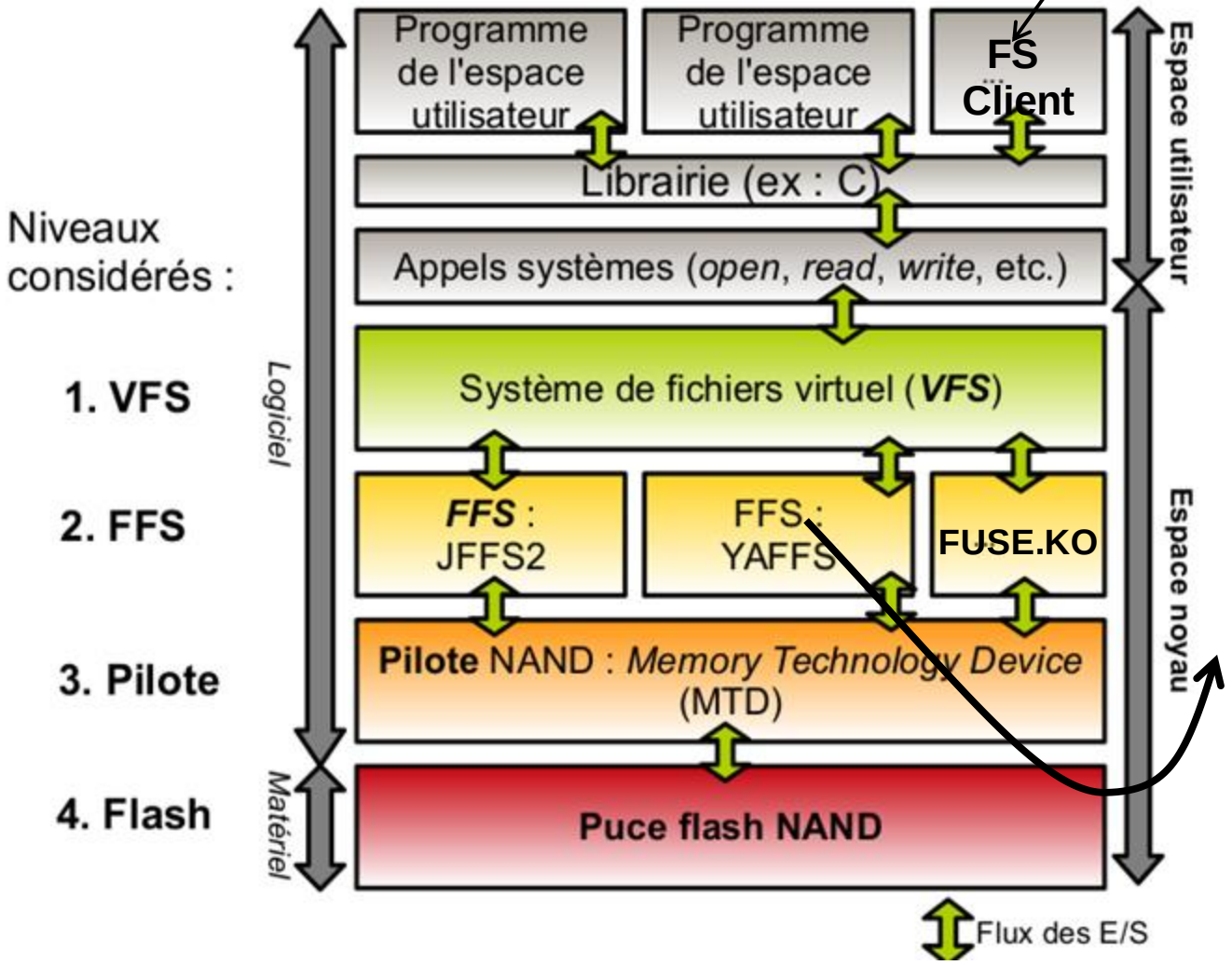


Inode Unique identifier
Limited number: When a file system is created, a fixed number of inodes are allocated.
No name = The inode itself does not store the filename. Filenames are stored in directory entries, which link the name to the inode.

```
j.acquaviva@c3sfr1:~$ ls -li script.stdout
12624386236935445807 -rw-r--r-- 1 j.acquaviva j.acquaviva Group 1126 Sep  1  2023 script.stdout
j.acquaviva@c3sfr1:~$ find . -inum 12624386236935445807 -exec stat {} \;
  File: ./script.stdout
  Size: 1126            Blocks: 3          IO Block: 524288 regular file
Device: 38h/56d Inode: 12624386236935445807  Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1397/j.acquaviva)   Gid: ( 1397/j.acquaviva Group)
Access: 2023-09-01 10:31:46.000000000 +0200
Modify: 2023-09-01 10:31:49.000000000 +0200
Change: 2023-09-01 10:31:49.000000000 +0200
 Birth: -
j.acquaviva@c3sfr1:~$
```

Call Stack

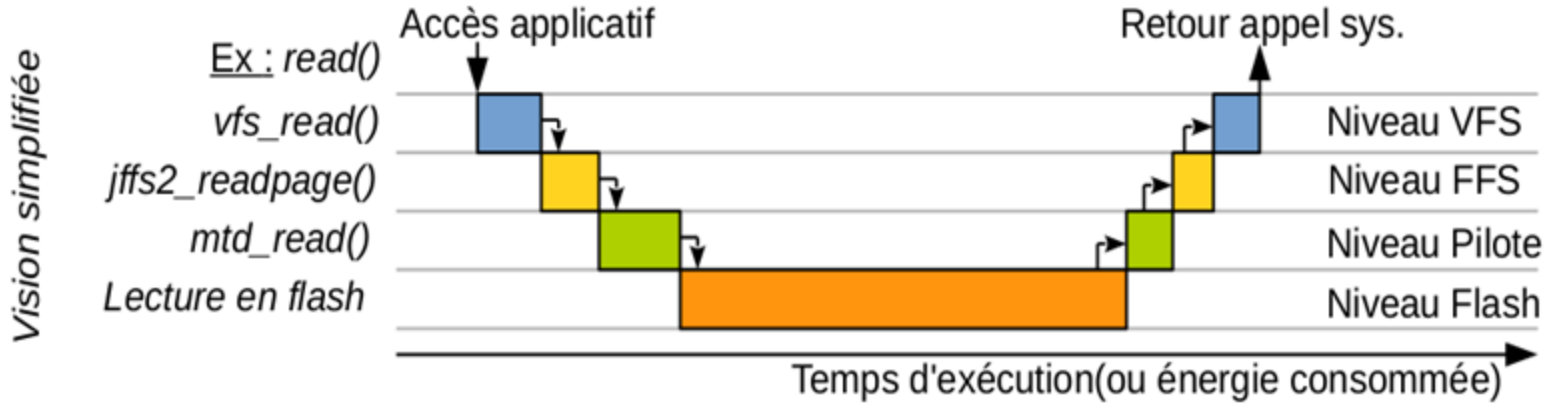
Management of memory cache and network FS communication protocol



Source: P. Olivier et J. Boukhobza

Overhead associated to Call Stack

▶ Gestion du stockage : **pile d'appels**



Source: P. Olivier et J. Boukhobza

Data and Metadata

Metadata : data describing data

- data attributed (and extended attribute)

→ size, user access rights, date of modification (ls –l)

!= pointer (does not allow to locate data on the storage system)

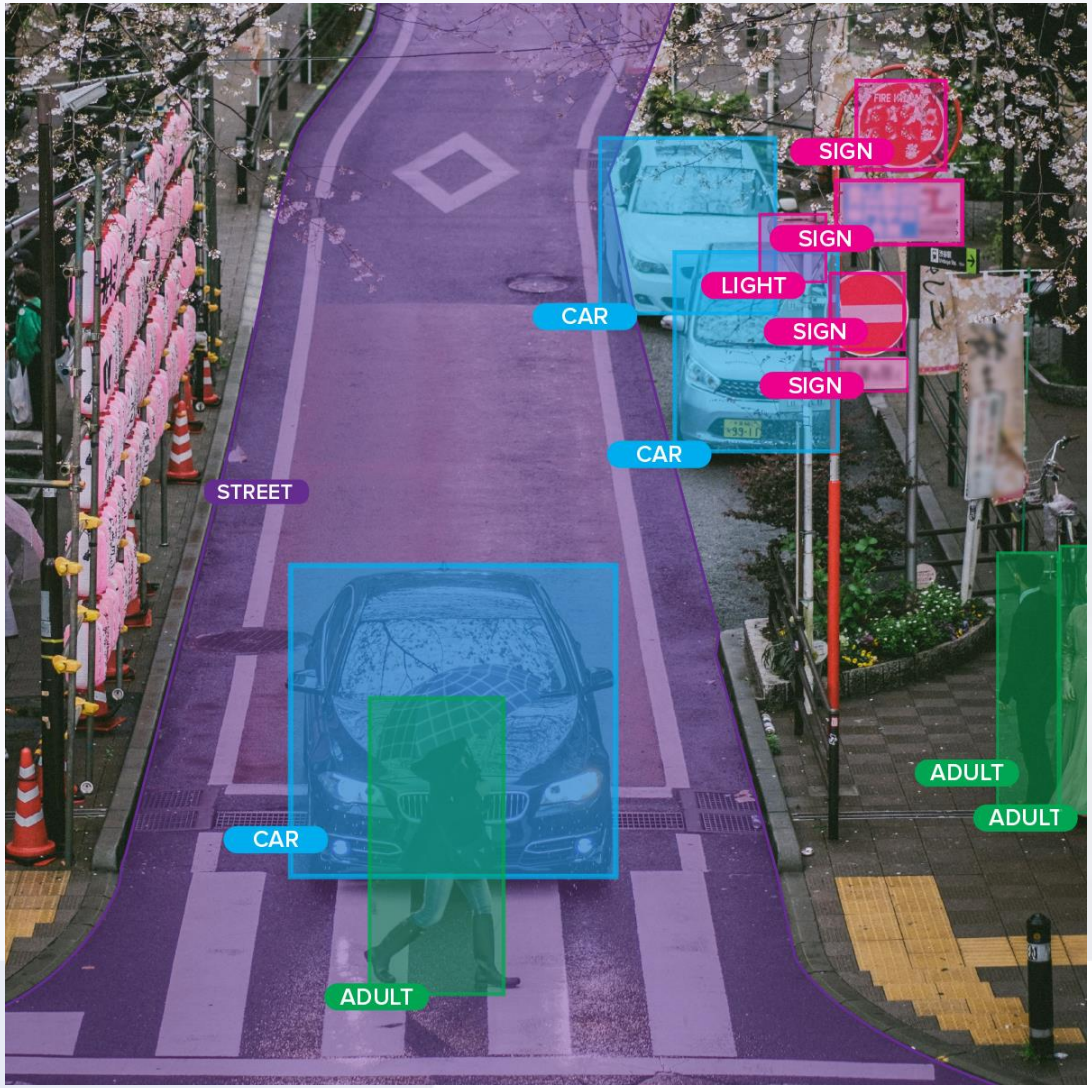
Metadata are at the core of the scalability challenge

→ metadata are accessed frequently and massively, e.g. *ls* –l in a directory with many file. No data access but many metadata accesses

Blurring Borders: Metadata & Data

THE AI DATA COMPANY

- AI Data tend to be metadata heavy
  - Every frame of an autonomous car is annotated by 100s of metadata
- Metadata allow to structure the Data-lake
  - Prevent Data-lake to turn in Data-Swamp
- Query-able Metadata: Data-LakeHouse
  - Data Lake + Data Warehouse



POSIX norm

Definition of the file access protocol Posix : 1988

→ At the time the world was sequential

→ accessing metadata was not an issue: no concurrent accesses, no risk of inconsistency

→ caching metadata was a straightforward solution

- List all the names of all the files hosted within a directory:

  Readdir

- Check the attributes the files hosted within a directory:

  Fstat for each and every file!

Questions:

What happens if the content of the same directory is modified by multiple distributed processes?

Performance and consecutive accesses?

Key Performance metrics

Bandwidth: volume of data read or written per second

→ throughput

IOPs: number of IO requests per second

→ difference between latency and throughput

Some order of magnitude

→ 1 AMD Roma CPU-DRAM: 200 GB/s

→ 1 network link IB HDR200: 20 GB/s

→ 1 Hard Drive: ~200 MB/s

→ #Top500: Frontiers: 9500 AMD Trento

→ An HPC system needs to be connected in parallel to thousands of storage devices

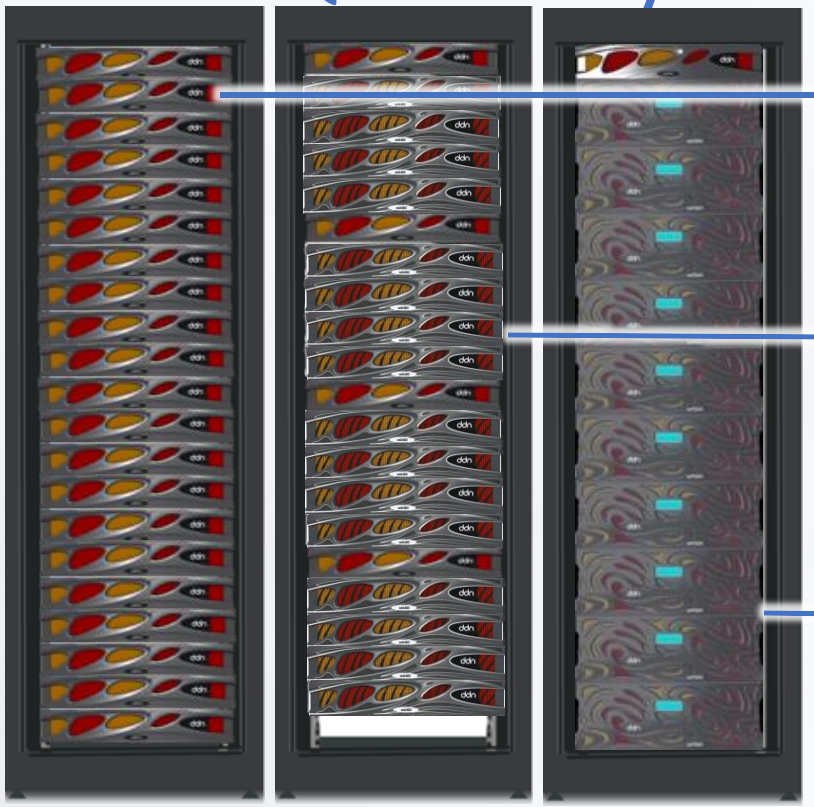
Key Performance metrics: devices

Metrics	Hard Drive (HDD)	Flash (NMVe) PCI gen4
Bandwidth	0.2 GB/s	8 GB/s
Latency	4 ms	0.02ms
Capacity	22 TB	60 TB (QLC)
Price	$14.3 / TB	$50 / TB

Reminder on Storage Architecture

THE AI DATA COMPANY

TLC Flash QLC Flash Hybrid



Best IOPS & Throughput per rack

Up to 70M IOPs in a single rack
1.8 TB/s of read throughput
1.4 TB/s of write throughput
16PB of Flash

Best Price per Flash TB

Up to 14M IOPs in a single rack
360 GB/s of read throughput
240 GB/s or write throughput
Up to 26PB of Flash

Best Price per TB

20PB per Rack
90 GB/s of read throughput
65 GB/s of write throughput

Technologies tend to stack

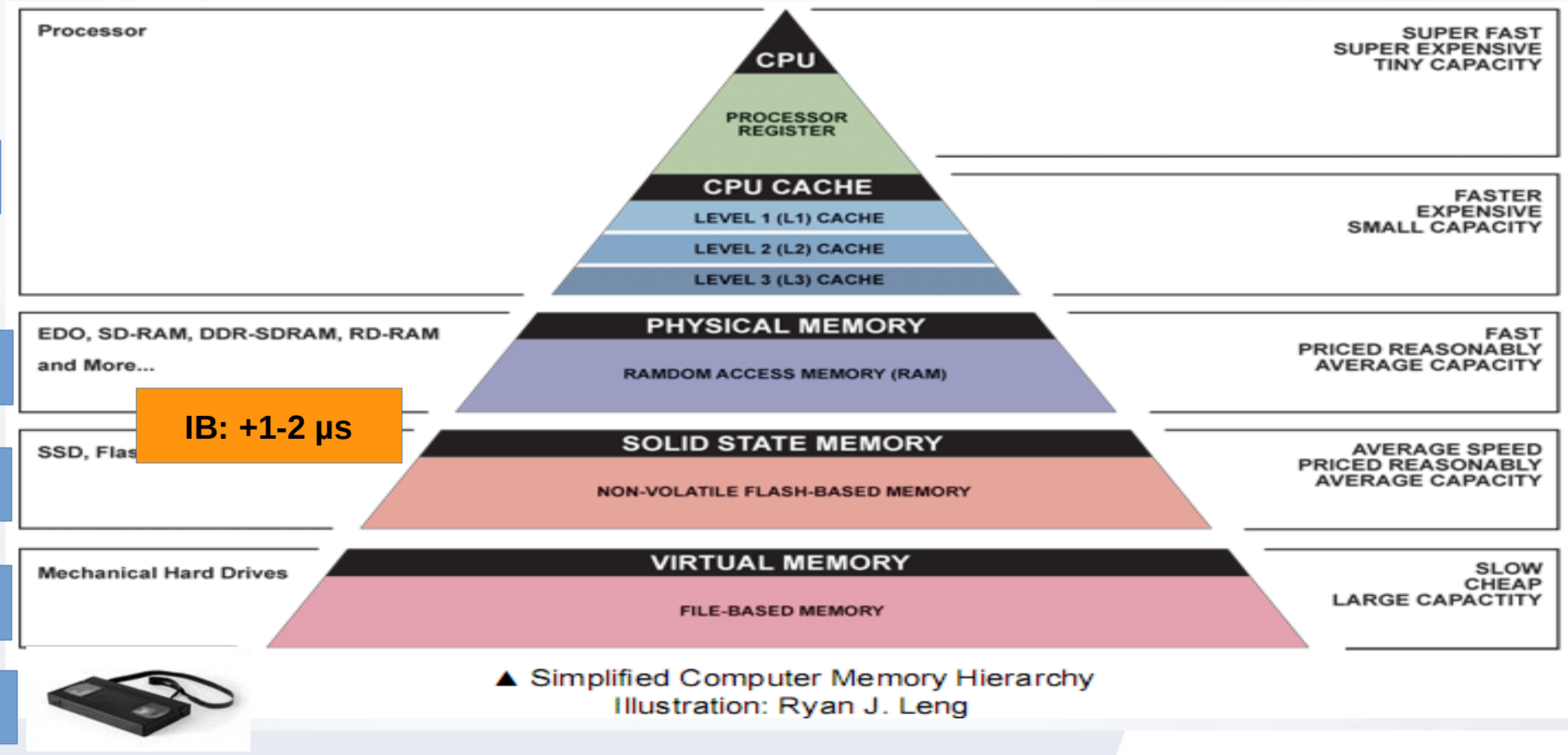
1-10 ns

50-100 ns

100 µs

5 ms

50 sec



IB: +1-2 µs

▲ Simplified Computer Memory Hierarchy
Illustration: Ryan J. Leng

Non-Parallel Filesystem Architecture (NAS)

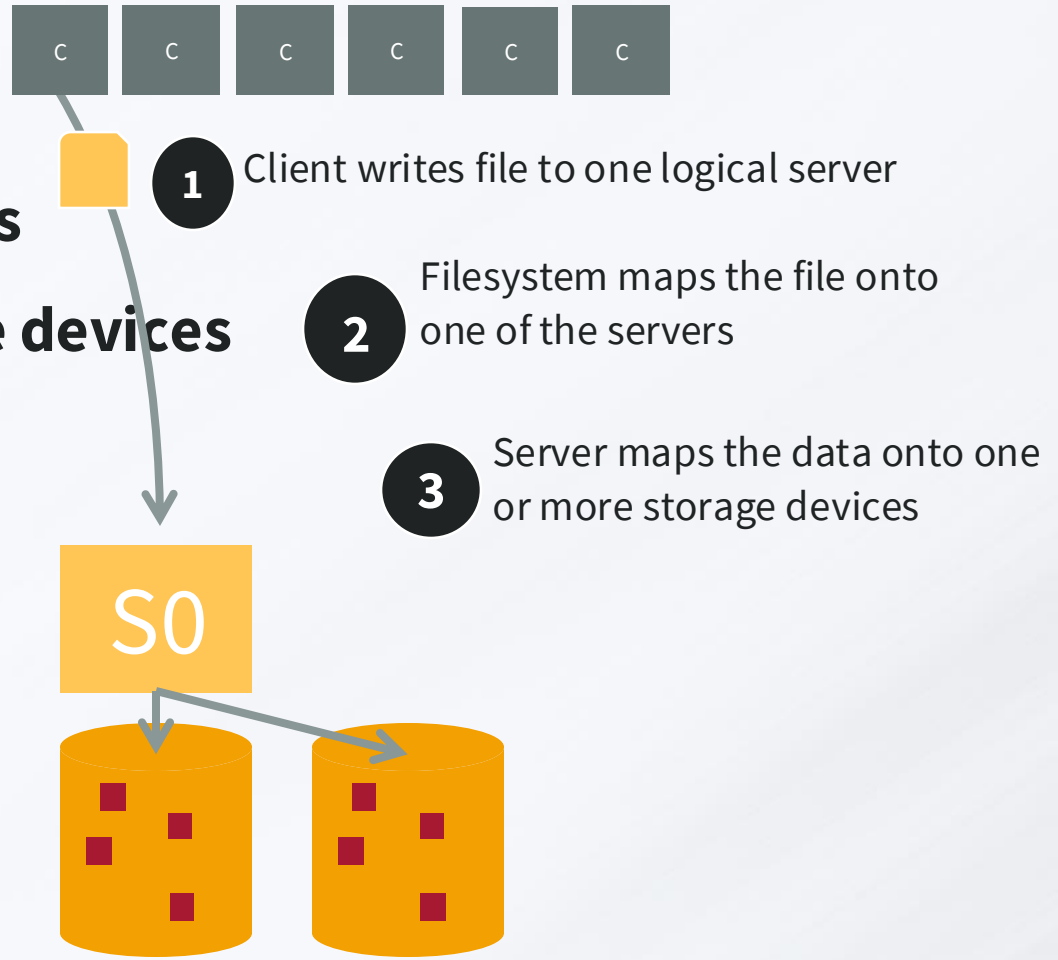
Start by looking at a simple SERIAL filesystem

Each client accesses only one *logical* server

Filesystem maps the file onto one of the servers

Server maps the data onto one or more storage devices

Performance is limited by single data path



Parallel Filesystem Architectures Scale Out NAS

Scale Out NAS Systems do not parallelize in granular way from client

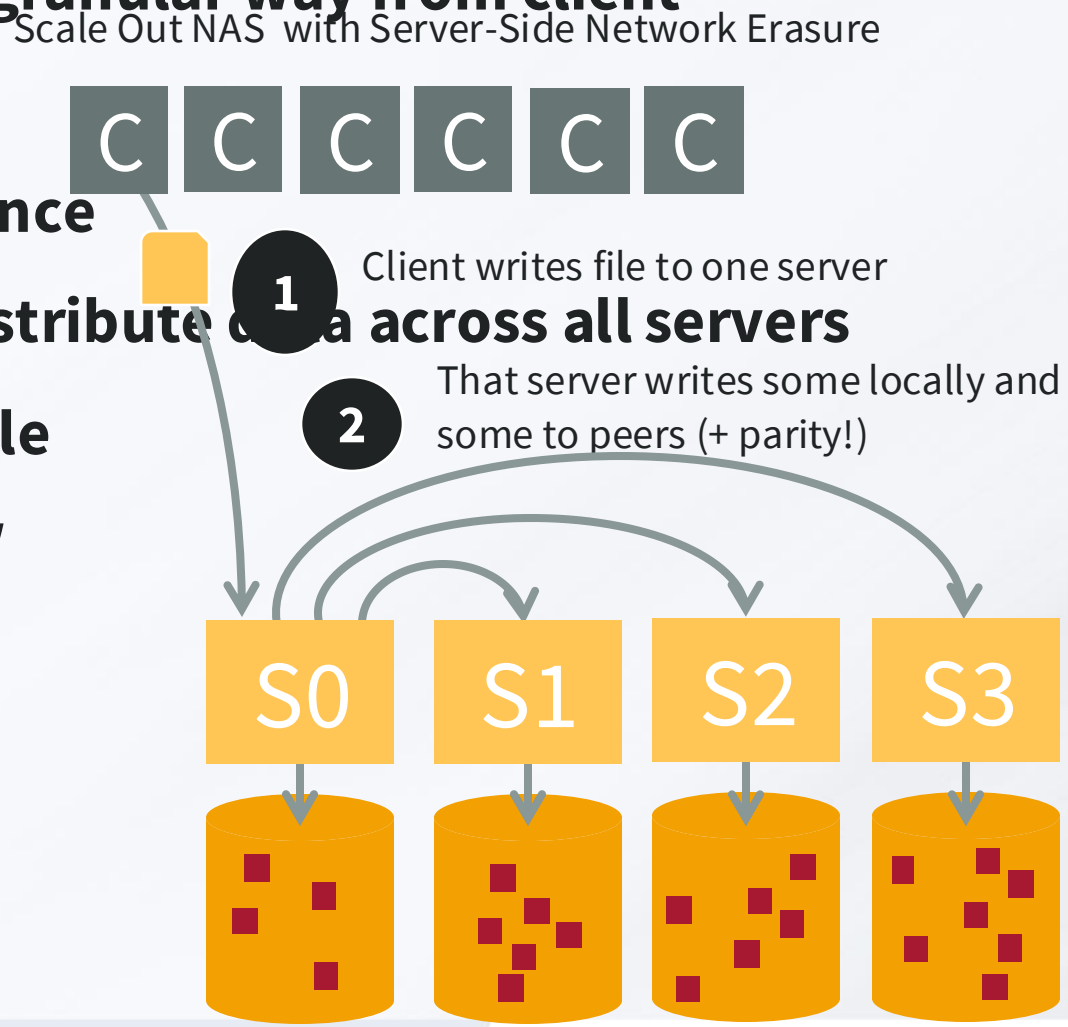
Each client accesses only one server

All clients access all servers, try to load-balance

Servers have to move data at back-end to distribute data across all servers

pNFS partly solves this issue, but not scalable

Note complexity and latency within back end!



Parallel Filesystem Architectures Scale Out NAS (#2)

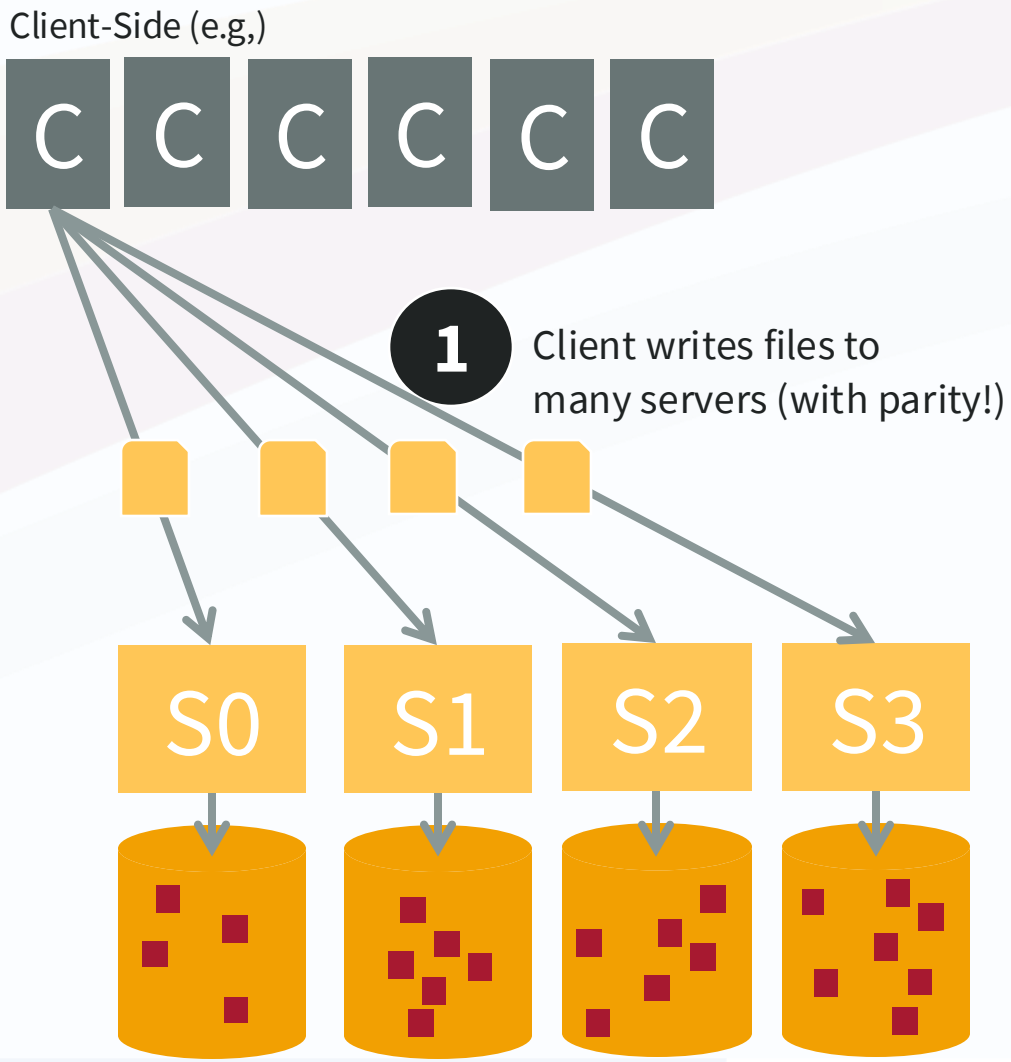
Client sends data in parallel to many servers across network

High bandwidth from client

Good scaling since as system gets bigger

- Every component (client, server) contributes to additional performance

BUT it consumes some resources on client and network can be congested



Shared Parallel Filesystem Lustre

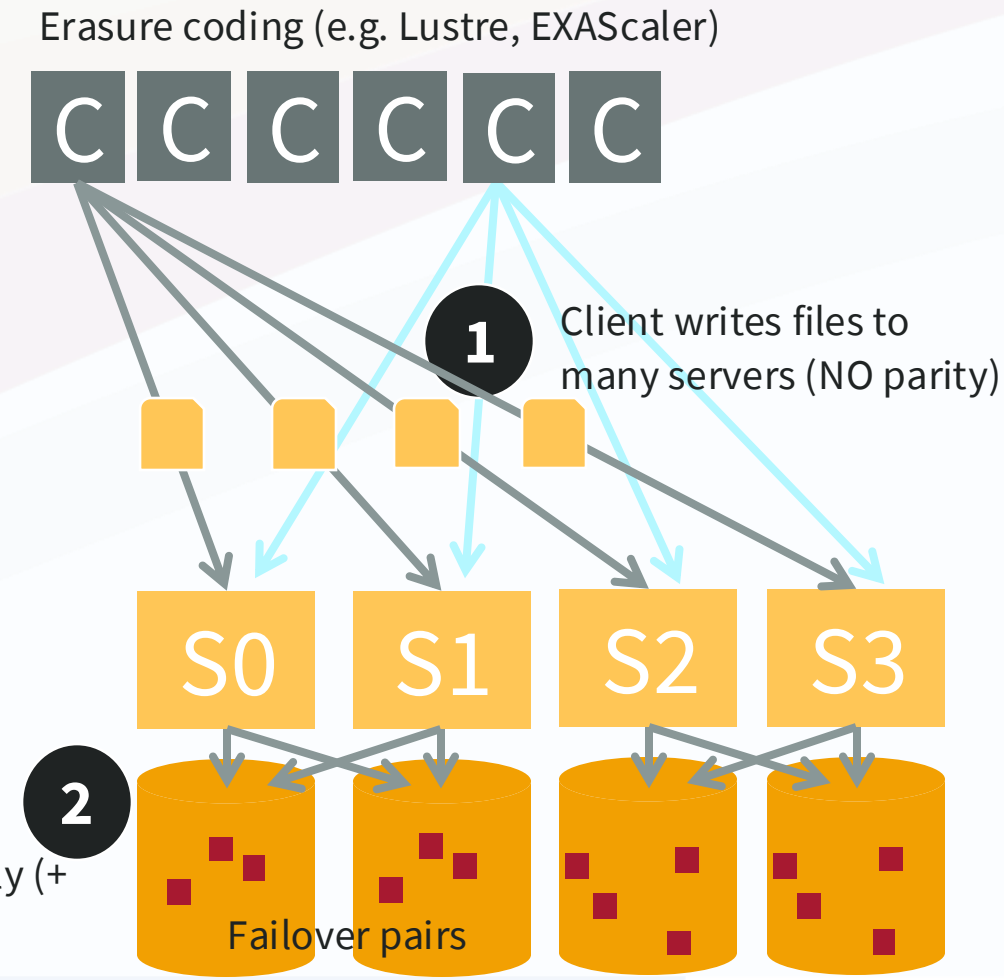
Each client can read/write to any and all servers, striping file across multiple servers

Clients can write SIMULTANEOUSLY

Servers responsible for concurrency and failover in event of issue on server side

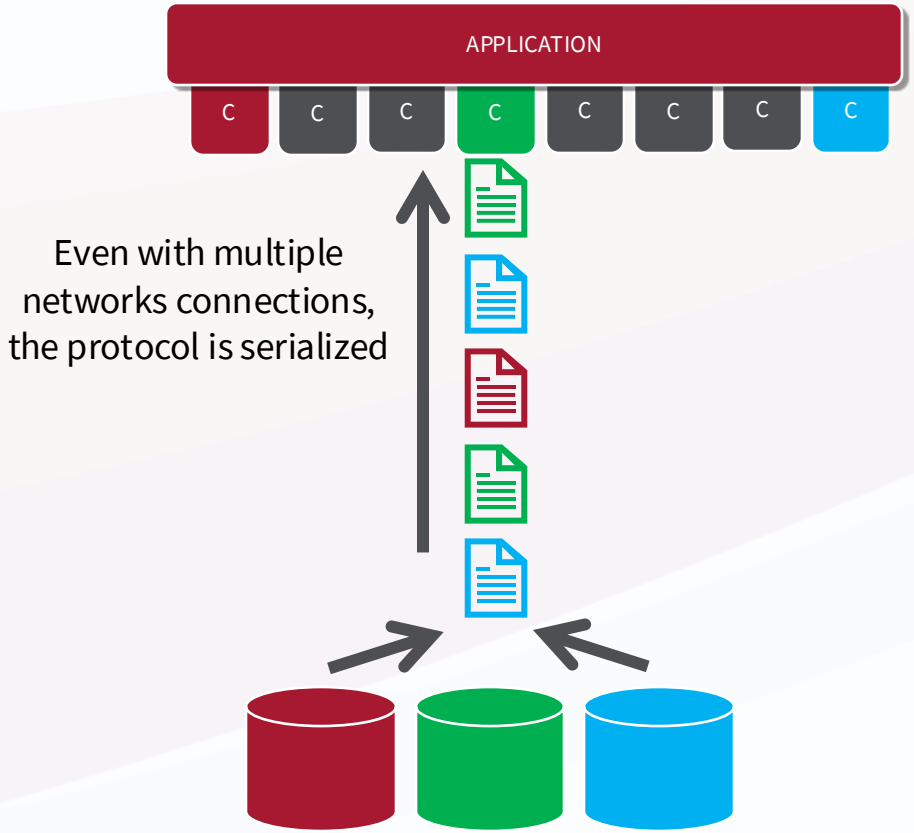
SHARED PARALLEL allows full concurrency

E.g. DDN EXAScaler, Lustre



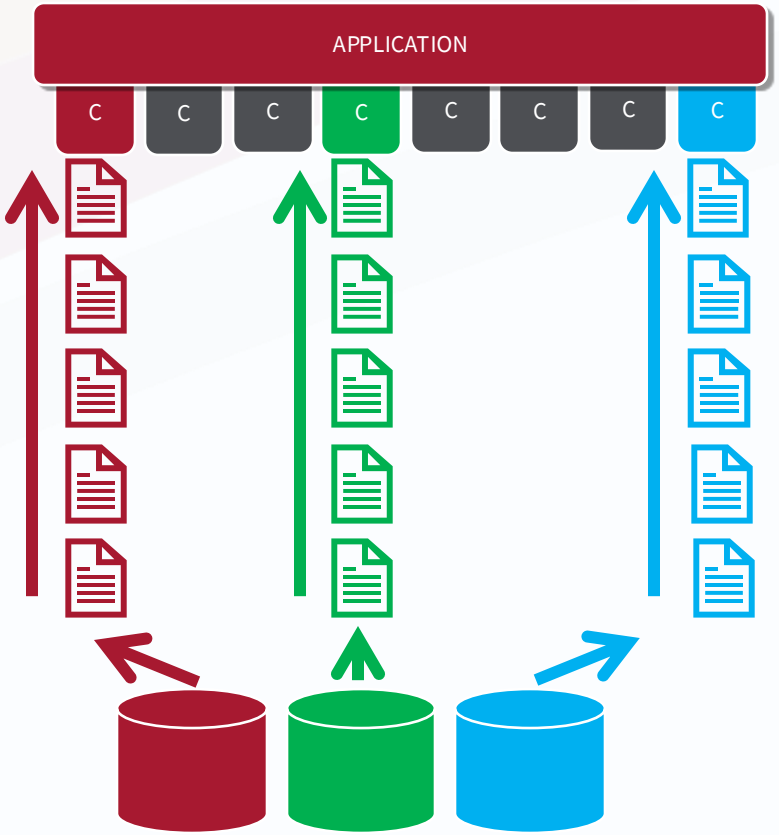
Key Enabler – Parallel Filesystem Architectures

Performance is driven by device speed – limit is around 2-4GB/sec for today's Gen4 NMVe



Even with multiple networks connections, the protocol is serialized

Parallelism is hard, but brings much more performance



File System vs Object Storage

Key properties of Object: Write Once – Key properties of Objects Storage: no hierarchy

File System

- **Definition of a tree structure based on a hierarchy of directories and files**
- Access to data requires tree parsing O(log*n*)

Object Store

- Hash table
- Access to data is constant at any scale O(1)

Object Storage

Quality of the hash table de la table de hachage

- Identification of the object owner or object metadata owner
  - Metadata management is a crucial point in object storage
- Uniform distribution across multiple object storage servers
- Definition of primary and secondary object servers
- → Fault tolerance
- Hashing speed
  → *cache-aware hashing function*
- CRUSH *(UC Santa Cruz)*

Take away
Storage systems are parallel systems

Thank You!

ddn