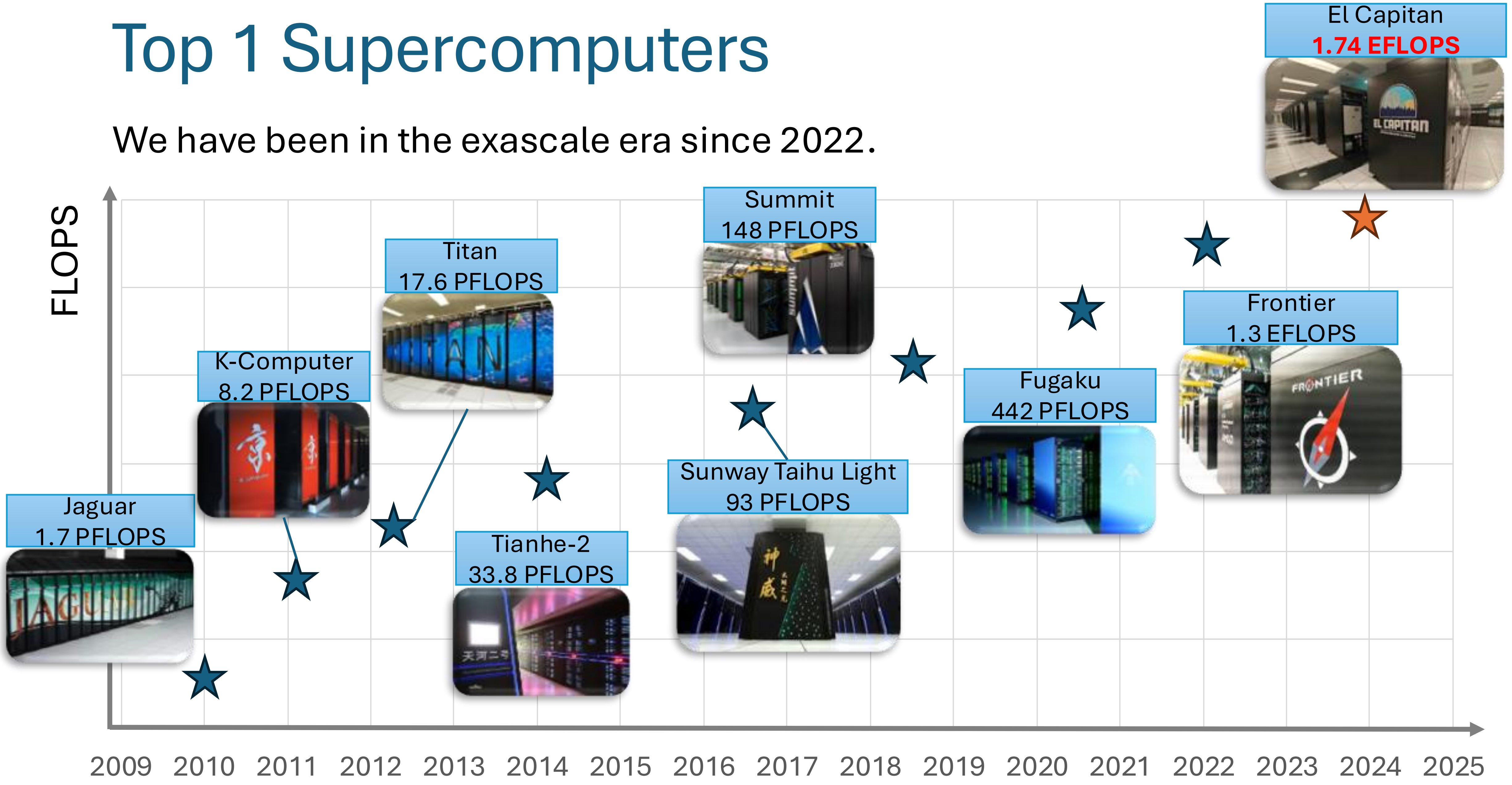
HPC Storage Systems in the Exascale Era: Trends, Challenges, and Opportunities

Chen Wang
chen.wang@ntu.edu.sg
College of Computing and Data Science, NTU
2025-09-25

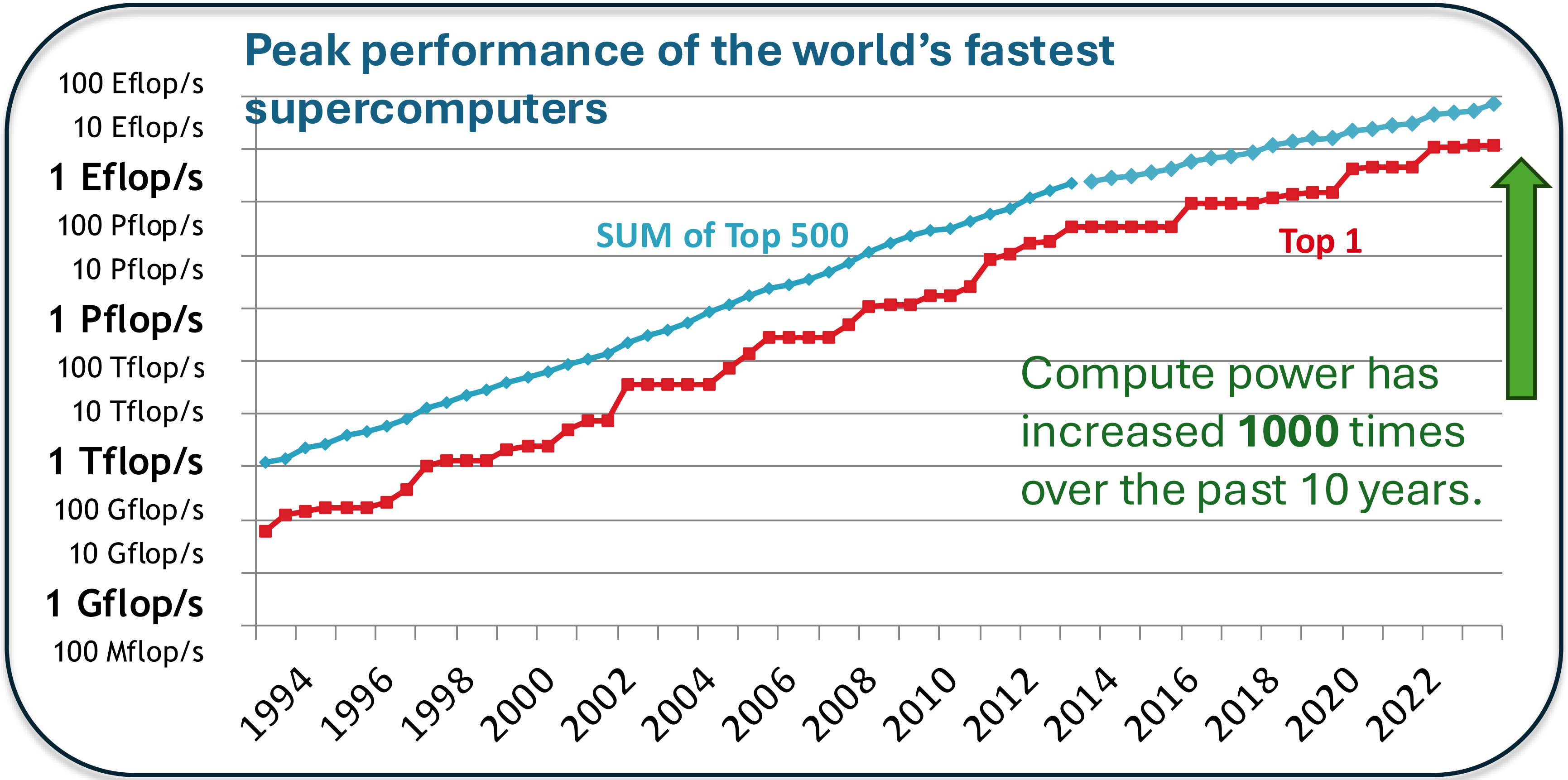
2025 HPC User Group Symposium

Top 1 Supercomputers

We have been in the exascale era since 2022.

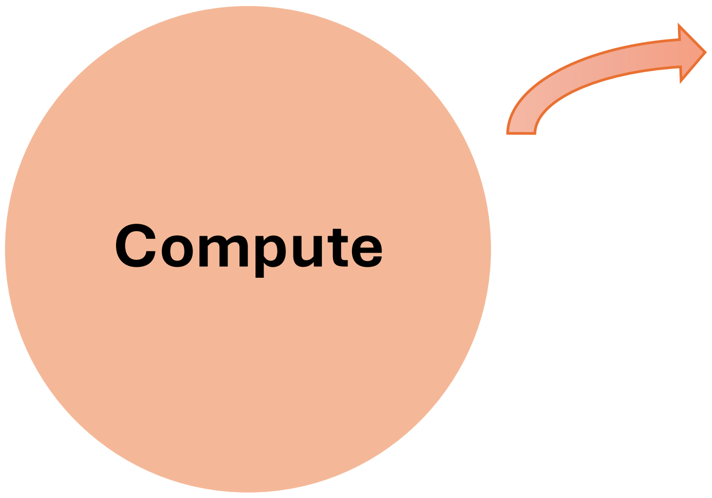


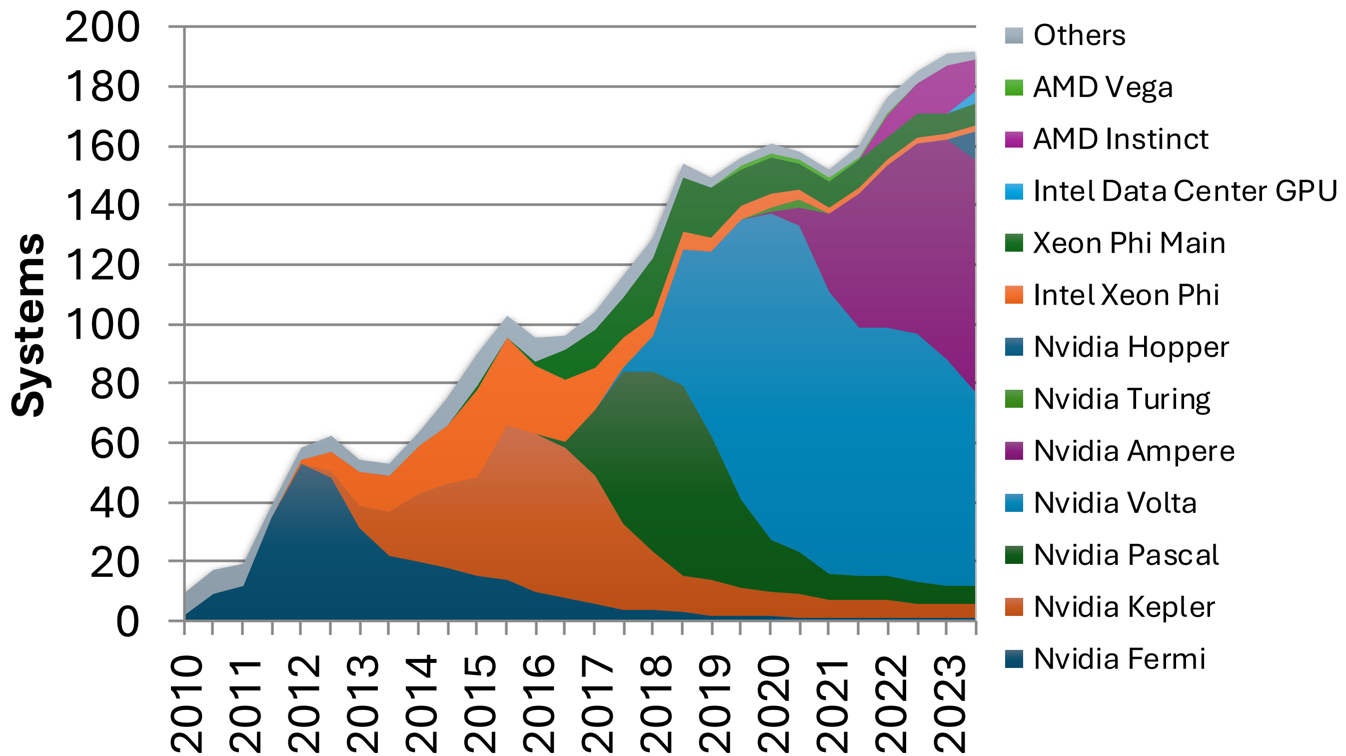
Compute Power Has Increased Significantly



Data Source: Top500.org

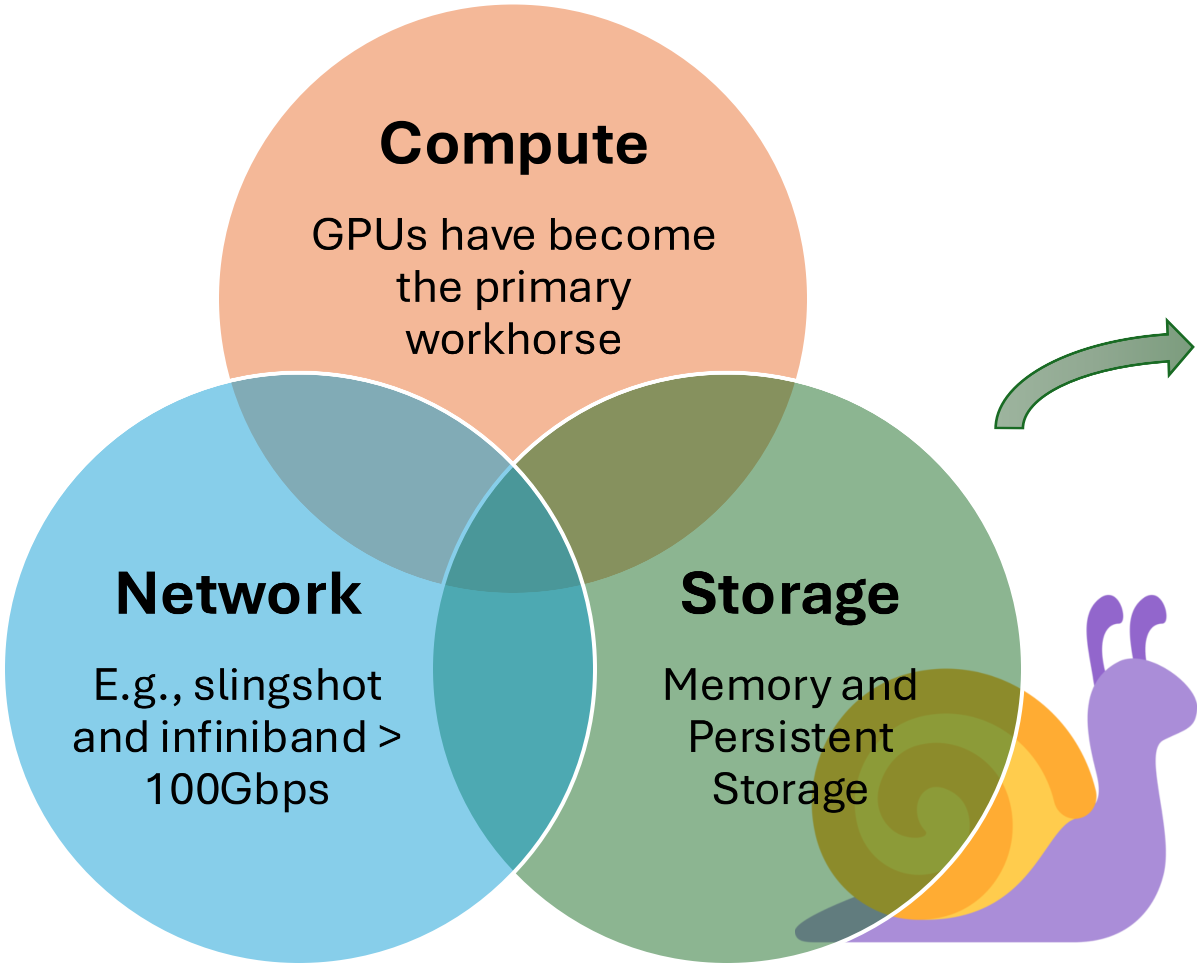
GPUs Have Become the Primary Workhorse

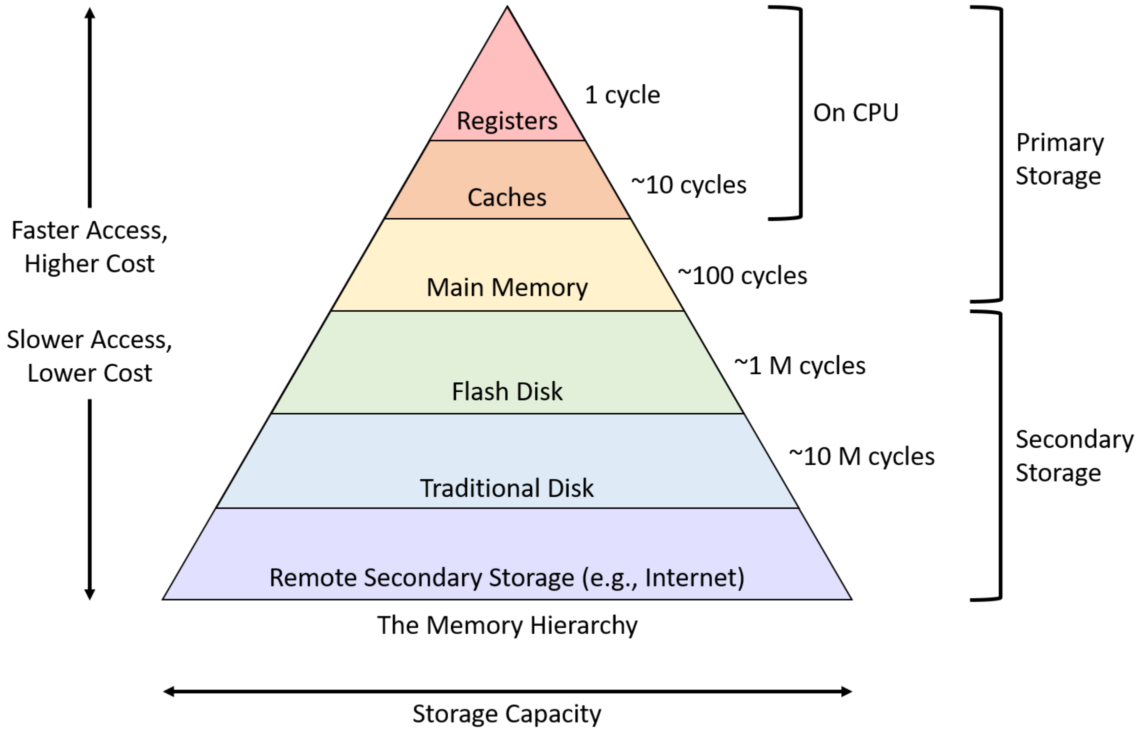




Over 1/3 of top 500 systems have accelerators

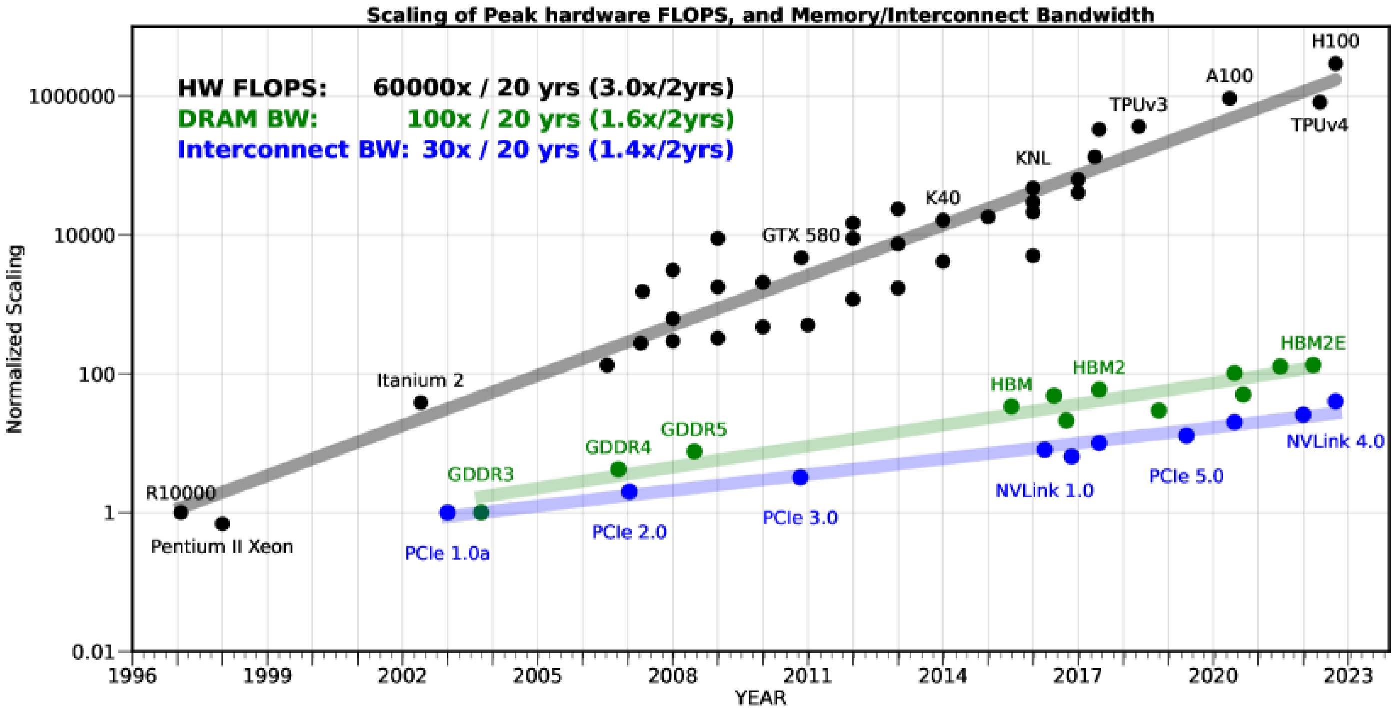
The Deep Storage Hierarchy





The entire storage hierarchy is getting deeper and more complex, and the boundary between memory and storage is steadily blurring.

AI and Memory Wall

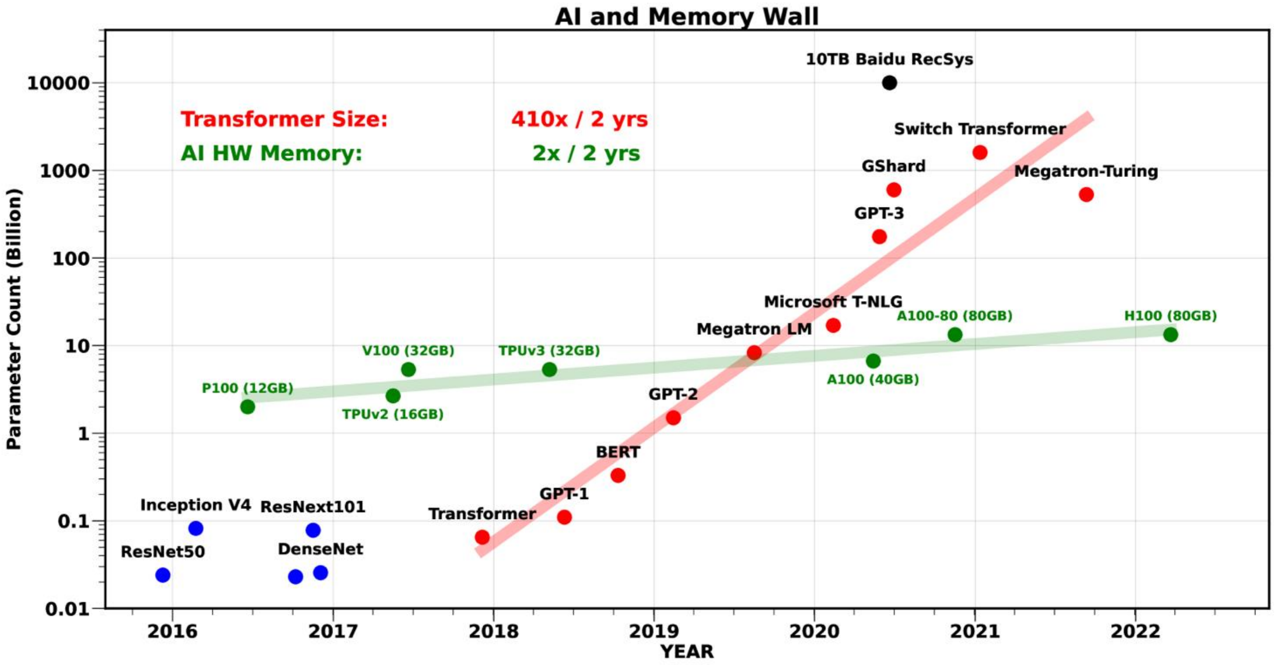


GPU FLOPS vs. Memory Bandwidth

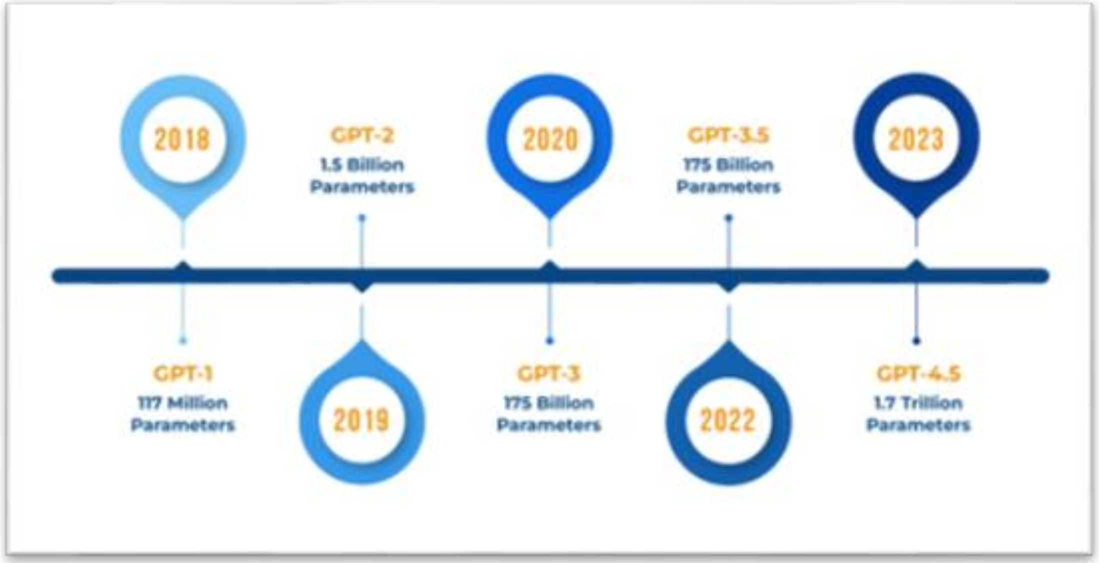
Gholami, Amir, Zhewei Yao, Sehoon Kim, Coleman Hooper, Michael W. Mahoney, and Kurt Keutzer. "Ai and memory wall." IEEE Micro 44, no. 3 (2024): 33-39.

The performance gap is expected to grow at 50% per year.

AI and Memory Wall



Transformer Size vs. Memory Capacity



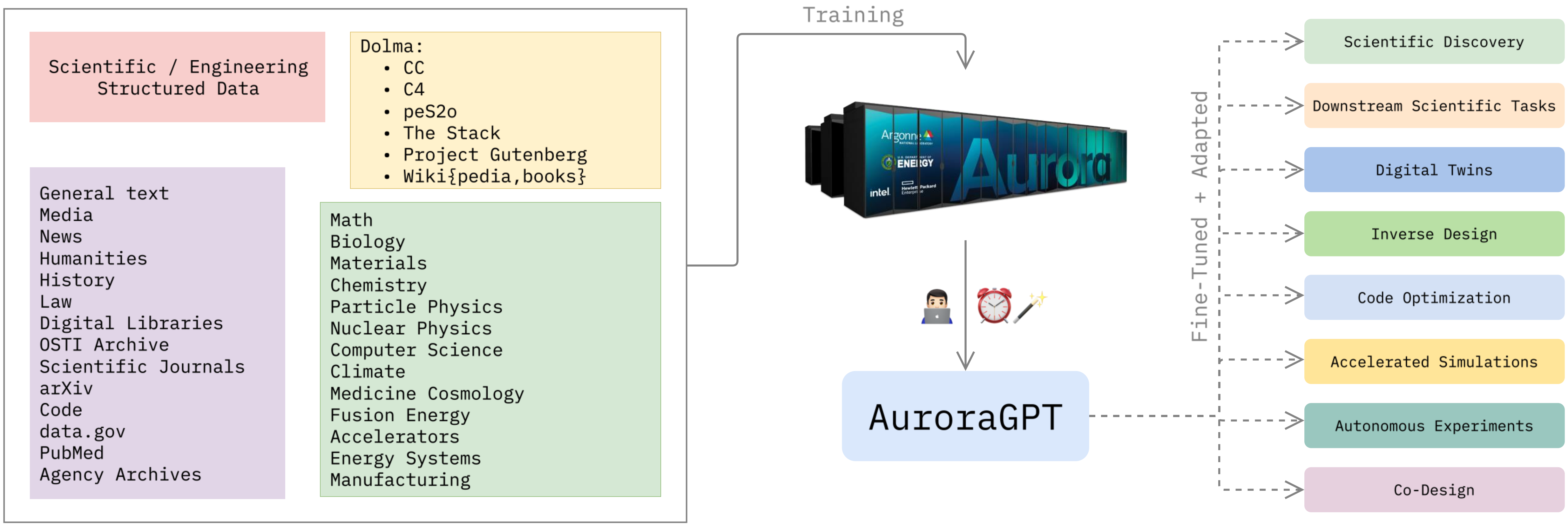
The Evolution of GPT Models

Gholami, Amir, Zhewei Yao, Sehoon Kim, Coleman Hooper, Michael W. Mahoney, and Kurt Keutzer. "Ai and memory wall." IEEE Micro 44, no. 3 (2024): 33-39.

Case Study: AuroraGPT

AuroraGPT*: *General purpose scientific LLM*. Broadly trained on a general corpora plus scientific {papers, texts, data}

*named after the Aurora Supercomputer at Argonne.



Source: Sam Foreman. 2025. "Scientific AI at Scale: AuroraGPT". *https://samforeman.me/talks/openskai25/ai4science/*

Case Study: AuroraGPT

AuroraGPT Goals:

- Explore pathways towards a **“Scientific Assistant”** model
- Multilingual – English, 日本語, Français, Deutsche, Español, Italiana
- Multimodal – images, tables, equations, proofs, time-series, graphs, fields, etc.
- **A series of LLMs: 7B, 70B, 200B, 1T**, etc. params.

Dataset	Format	Size
CORE	Full text collection of scientific papers	>2TB
peS2o	Jsonl (40M open access academic papers)	259GB
PMC-OA	markdown+pdf	202GB
Arxiv	pdf+figures	2.2TB
Biorxiv	xml+pdf+figures	9.7TB
Medrxiv	xml+pdf+figures	542GB
chemrxiv	pdf	
ACM	XML	16GB
NIH_LITARCH	xml+pdf+figures	153GB

Scientific dataset: 20T tokens ~100M papers.

Case Study: AuroraGPT

Challenges:

- GPU-agnostic: NVIDIA, Intel, AMD, etc.
- Parallel Computing: Identify the right level of parallelism for Exascale machines.
- Data Handling: Converting PDF (math formula, figures) to texts; De-duplication to avoid memorization and bias.

Aurora (#3 in Top500)
166 Racks
10,624 Nodes
21,248 CPUs (Intel Xeon Max)
63,744 GPUs (Intel Data Center GPU Max)
84,992 NICs (Slingshot-11)
8 PB HBM
10 PB DDR5

AMD MI250X

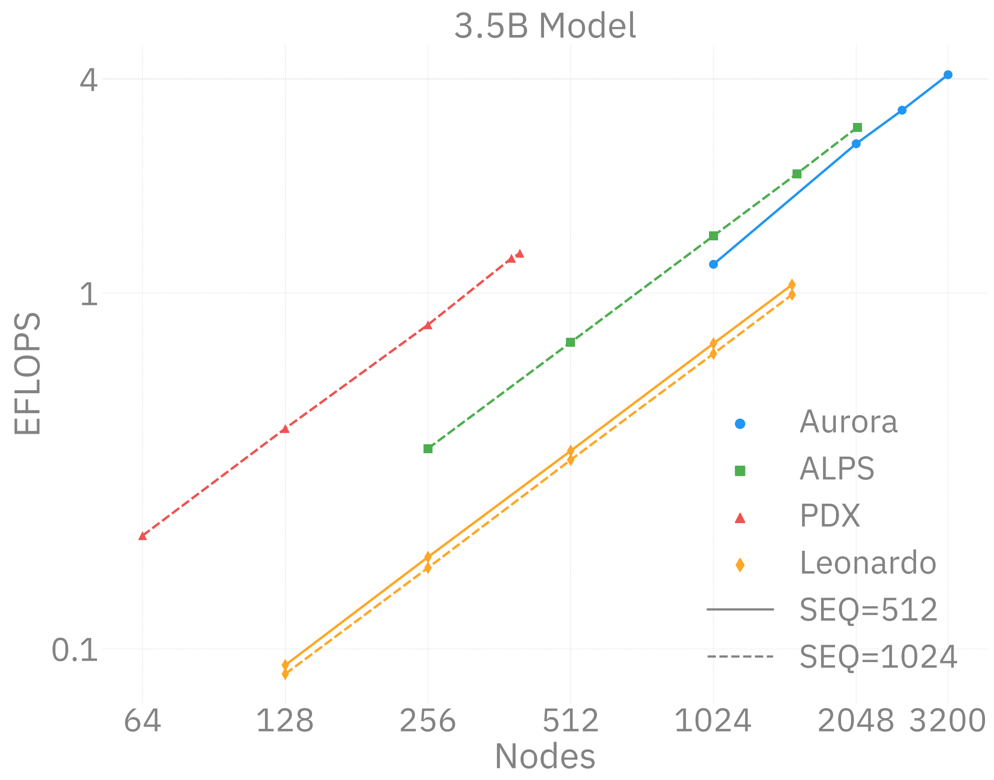
NVIDIA GH200

Case Study : AuroraGPT

- Forked implementation of *Megatron-Deepseed*
  - 3D parallelism-based implementation: tensor, pipeline, and data parallelism
  - https://github.com/argonne-lcf/Megatron-DeepSpeed
- Package management tools (e.g., Conda) can have tens of thousands of small files and Python imports can iterate over many of them.
  - Over 30,000 files opened on 38,400 ranks; > 900 million metadata operations

    → File system stalls for all users.
- After tokenization, have to manually split the dataset and ensure balanced loading
- No system-side solutions. Users now must deal with these issues.

MProt-DPO: Scaling Results



Dharuman, Gautham, Kyle Hippe, Alexander Brace, Sam Foreman, Väinö Hatanpää, Varuni K. Sastry, Huihuo Zheng et al. "MProt-DPO: Breaking the ExaFLOPS barrier for multimodal protein design workflows with direct preference optimization." SC'24, pp. 1-13, 2024.

Trends, Challenges, and Opportunities

Trends

- GPU performance outpaces memory (capacity & bandwidth)
  - Distributed training for large-scale models.
  - **Deeper memory/storage hierarchy**
- I/O optimizations are performed at the application level.

Challenges

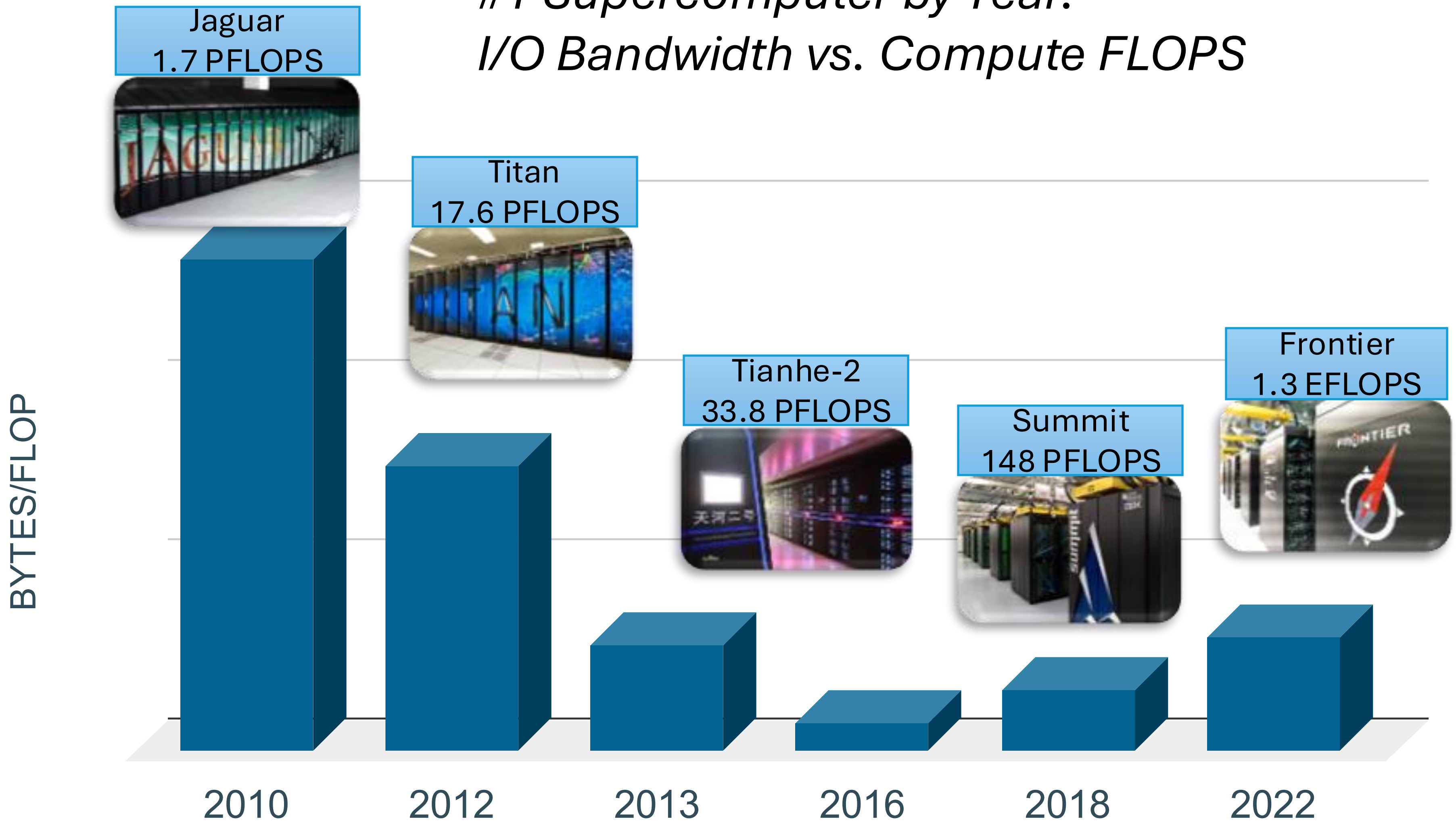
- Current storage architectures and software systems are not optimized for AI/LLM data access.
- Hardware-agnostic AI remains difficult.
- I/O issues often only occur at large scales.

Opportunities

- Need for deep storage-hierarchy-aware designs in future architectures.
  - Intelligent management of deep storage hierarchies is key to efficient AI/LLM
  - **Data locality is everything**
- Requires collaboration between storage system designers & AI researchers.

I/O is even Slower

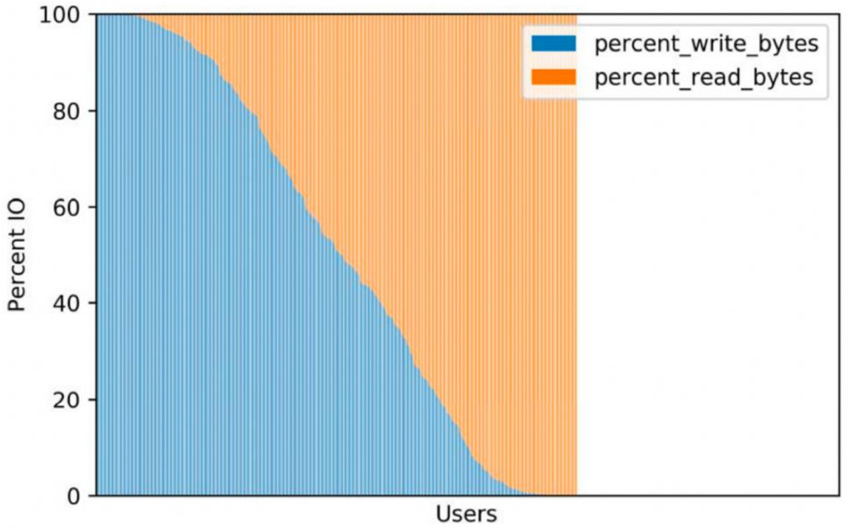
#1 Supercomputer by Year:
I/O Bandwidth vs. Compute FLOPS



I/O Subsystem Inefficiency

A study of 4 million jobs over four years on two LLNL systems shows that

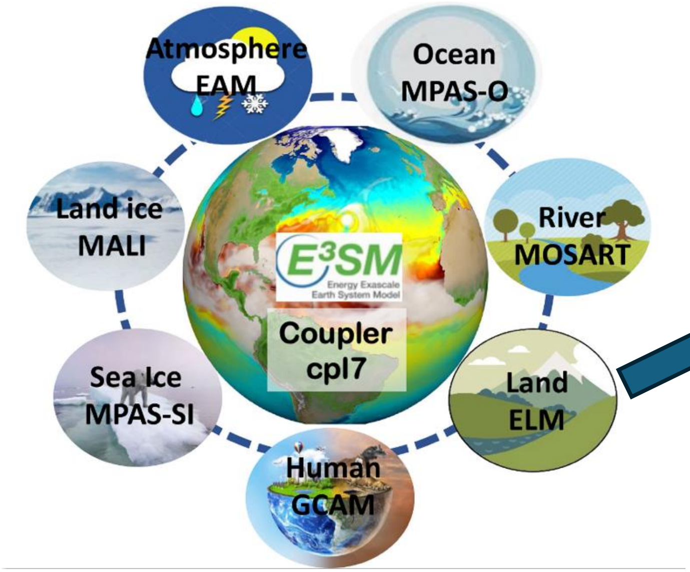
- on average, jobs which performing I/O spread I/O activities across 78.8% of their runtime.
- less than 22% write-intensive jobs perform efficient writes.
- HPC jobs are no longer write dominated

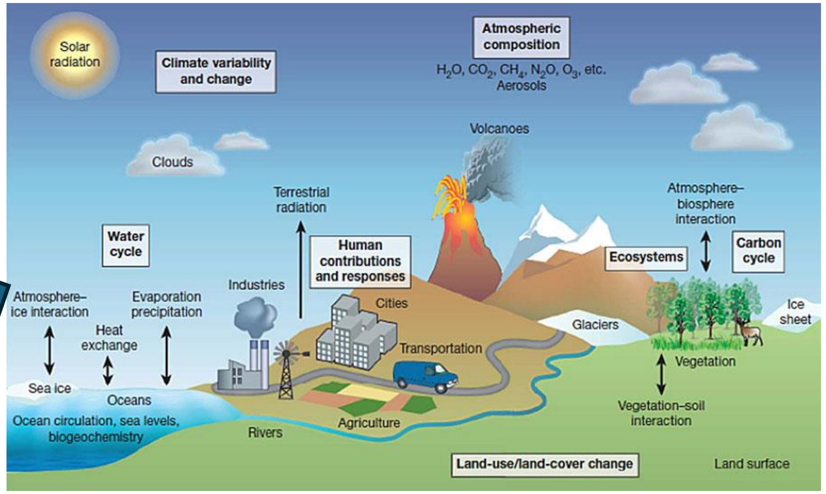


Percentage I/O (Write vs. Read) by User

Paul, Arnab K., et al. "Understanding HPC Application I/O behavior using system level statistics." 2020 IEEE HiPC.

Case Study: Energy Exascale Earth System Model (E3SM)

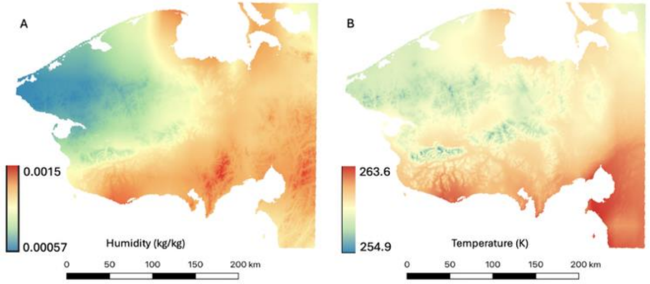




File	Size
Surface (I)	188 GB
Forcing (I)	1.4 TB
History (O)	134 GB
Restart (O)	4.2 TB

I:Input O:Output

A high-resolution (1kmx1km, previously 10kmx10km) land simulation over Alaska (21.6 Million land grid cell)
Used three supercomputers: Perlmutter, Summit (#1 from 2018-2020), and Frontier (#1 2022-2023, #2 now)

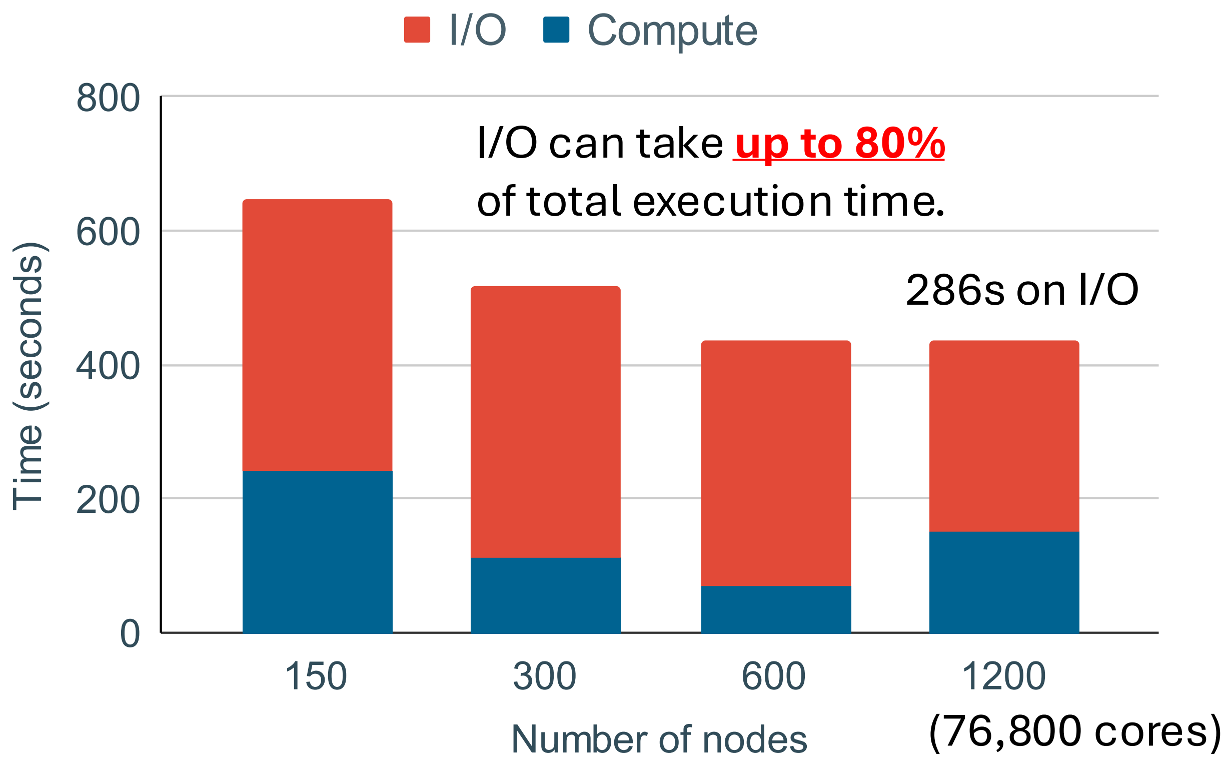


2025 CCGRID SCALE Challenge Finalist: Dali Wang, Chen Wang, Qinglei Cao, and et.al. "Scaling Ultrahigh-Resolution E3SM Land Model for Leadership-Class Supercomputers".

Case Study: Energy Exascale Earth System Model (E3SM)

Strong scaling results on Frontier.

- Up to 1200 nodes with I/O.
  - **Bottleneck**: 76,800 processes concurrently write to a single file.
- Up to 4000 nodes (nearly half of the Frontier) without I/O.
  - Bottleneck: initialization phase.
- *Note this is only a 5-day test simulation.*
- *We encountered both the scalability issue and the I/O bottleneck.*



2025 CCGRID SCALE Challenge Finalist: Dali Wang, Chen Wang, Qinglei Cao, and et.al. "Scaling Ultrahigh-Resolution E3SM Land Model for Leadership-Class Supercomputers".

How I/O Works in HPC

HPC Application

NWCHEM HIGH-PERFORMANCE COMPUTATIONAL CHEMISTRY SOFTWARE

LAMMPS

WCRP World Climate Research Programme CMIP6

OpenAI

deepseek

E3SM Energy Exascale Earth System Model

I/O Library and middleware

HDF5

netCDF

NVIDIA DALI

MPI

HPC File System

lustre File System

BeeGFS

IBM Spectrum Scale

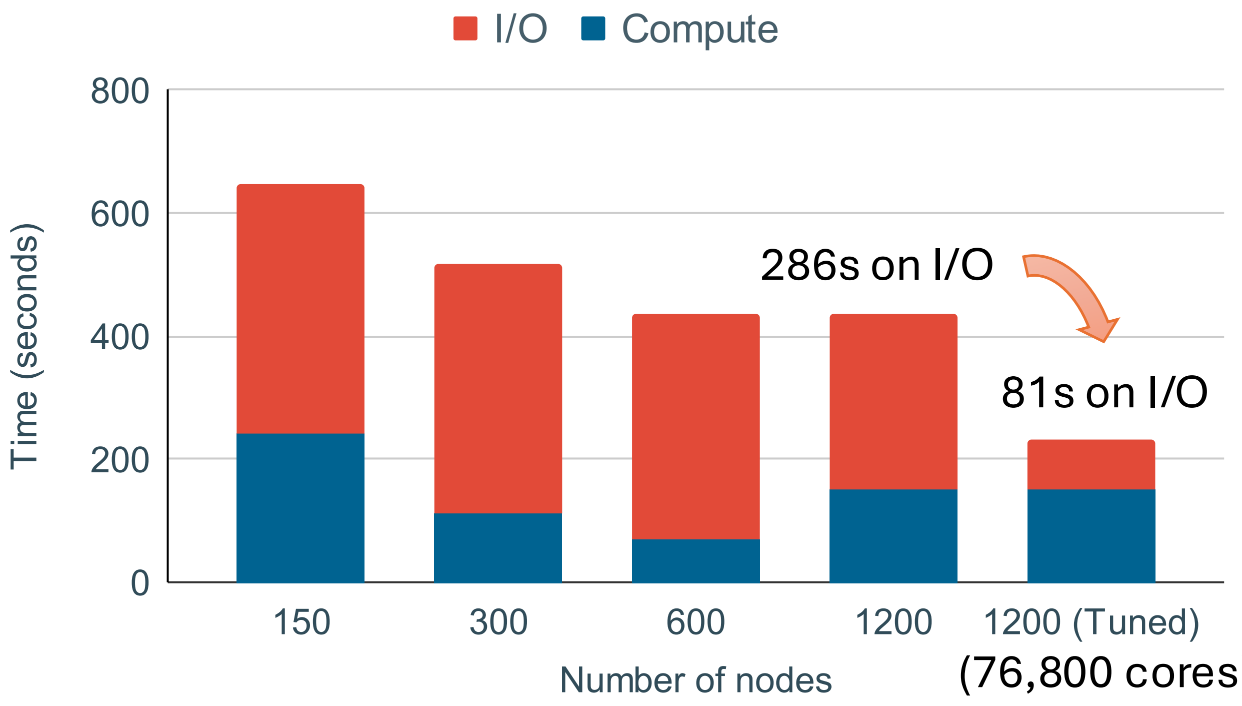
GlusterFS

The E3SM Example:

E3SM
- SCORPIO
  - PnetCDF
    - MPI-IO
      - File system

SCORPIO, PnetCDF, MPI-IO: I/O library and middleware

Case Study: Energy Exascale Earth System Model (E3SM)



(76,800 cores)

76,800 processes concurrently open/read/write a single file, causing significant congestions.
→ Delegate all I/O to one aggregator per node. Operate on one file per node.

After tuning:

- I/O time: 286s → 81s. (3.5x)
- Write bandwidth → ~300GB/s. This is still far away from the system peak performance (5TB/s).

2025 CCGRID SCALE Challenge Finalist: Dali Wang, Chen Wang, Qinglei Cao, and et.al. "Scaling Ultrahigh-Resolution E3SM Land Model for Leadership-Class Supercomputers".

Trends, Challenges, and Opportunities

Trends

- Many legacy code still relies on CPU. Running them on GPU nodes is wasteful.
- The I/O bottleneck often manifests only at large scale and is hard to get away once occurred.
- HPC I/O is no longer dominated by large writes; read volume is increasing rapidly due to AI and ML tasks.

Challenges

- It is difficult and time-consuming to port legacy HPC code to modern languages, frameworks, or GPUs.
- Tuning parallel I/O is hard due to the complex and deep hierarchy.
  - Default I/O configurations and optimizations are no longer fit for exascale runs.
  - Manual tuning approach requires expertise on both applications and storage systems and may not be always feasible.

Opportunities

- Application side:
  - LLM-guided code translation may be worth exploring.
  - Auto-tuning and learning-based tuning mechanisms for I/O optimization at large scale.
- System side:
  - Co-allocating GPU-heavy and CPU-heavy jobs with smart scheduling algorithms
    - Disaggregated memory and storage can help.
- Performance engineering knowledge and talents are need especially as system scales up.

Optimizing HPC File Systems?

HPC Application

NWCHEM
HIGH-PERFORMANCE COMPUTATIONAL CHEMISTRY SOFTWARE

LAMMPS

WCRP CMIP6
World Climate Research Programme

OpenAI

deepseek

E3SM
Energy Exascale Earth System Model

I/O Library and middleware

HDF5

netCDF

NVIDIA DALI

MPI

HPC File System

lustre File System

BeeGFS®

IBM Spectrum Scale

GlusterFS

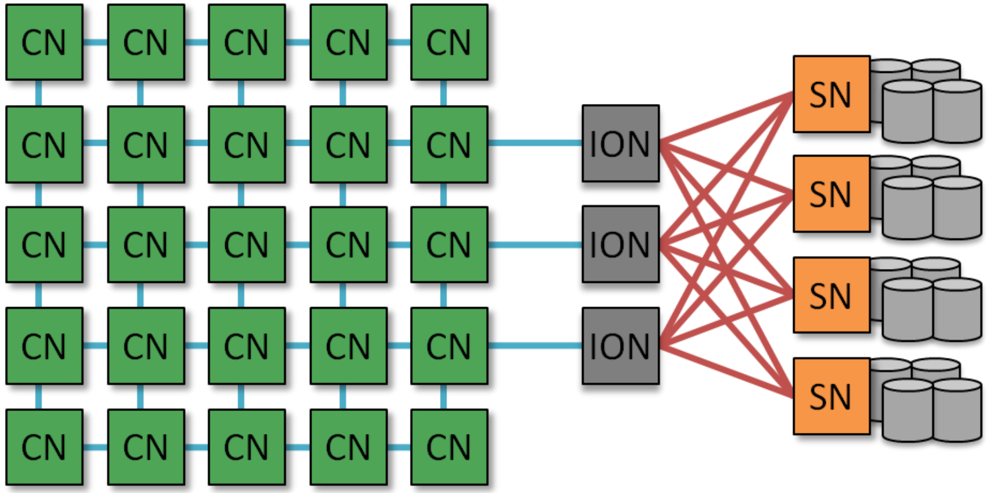
Optimizing HPC File Systems?

Traditional HPC file systems are **global resources shared by all users and jobs**. They are static and unable to adapt to different workloads, making it basically impossible to optimize for a single job. This limitation affects all applications.

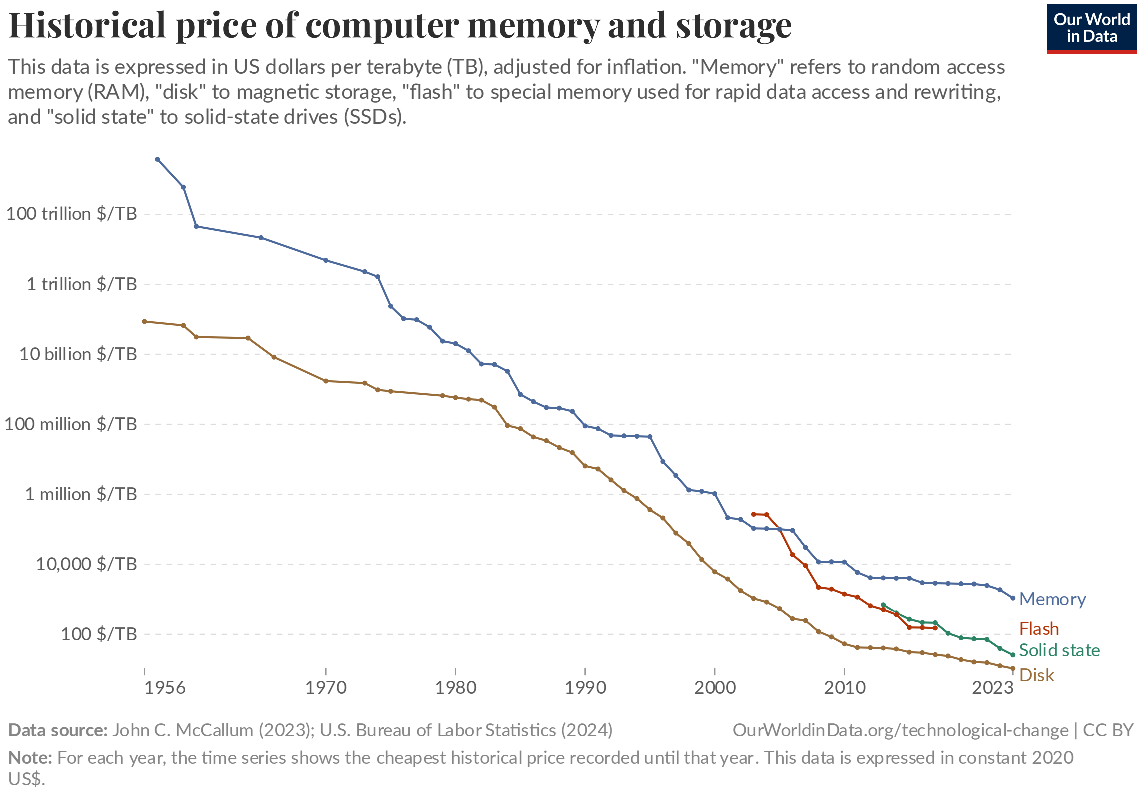
Burst Buffers: Yet Another Storage Layer

"Tape is Dead. Disk is Tape. Flash is Disk." (at CIRD'07)

Jim Gray
(1998 Turing Award Winner)



CN: Compute Node; ION: I/O Node; SN: Storage Node

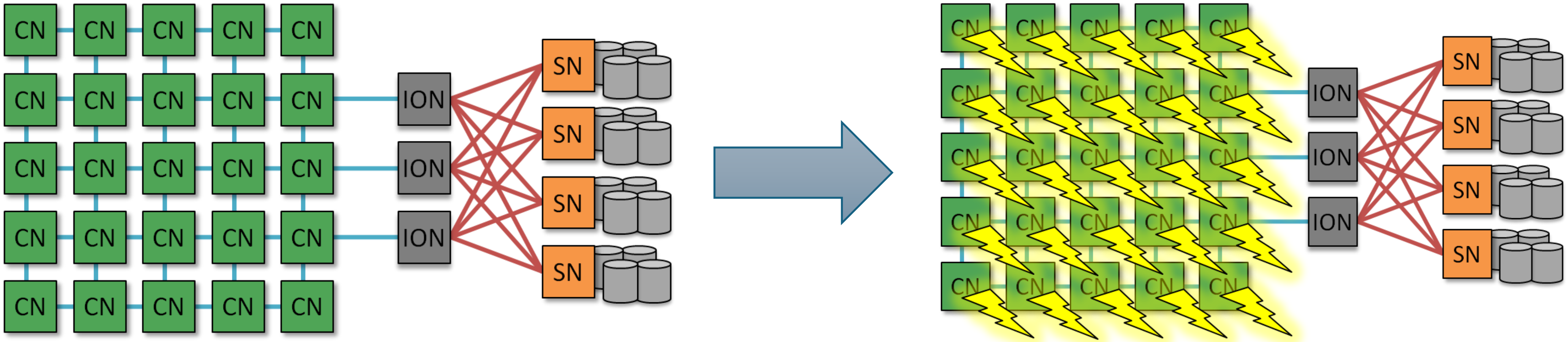


Figures courtesy of Glenn K. Lockwood.
https://blog.glennklockwood.com/2017/03/reviewing-state-of-art-of-burst-buffers.html

Burst Buffers: Yet Another Storage Layer

Node-local burst buffer:

- Attach one SSD to each compute node.
  - Scales linearly.
- **Only accessable from the attached node.**
  - Users need to manage data transfers across layers and between nodes.



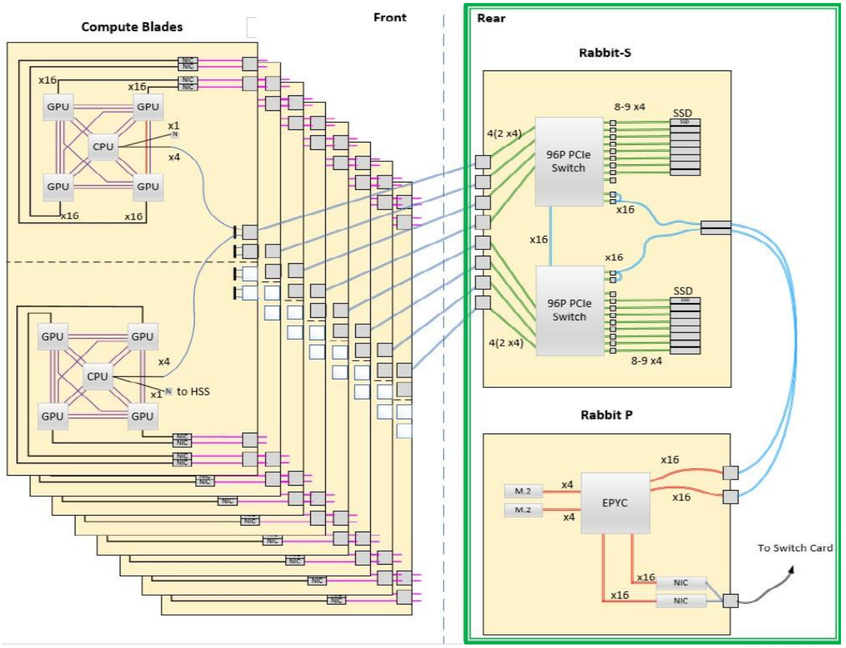
CN: Compute Node; ION: I/O Node; SN: Storage Node

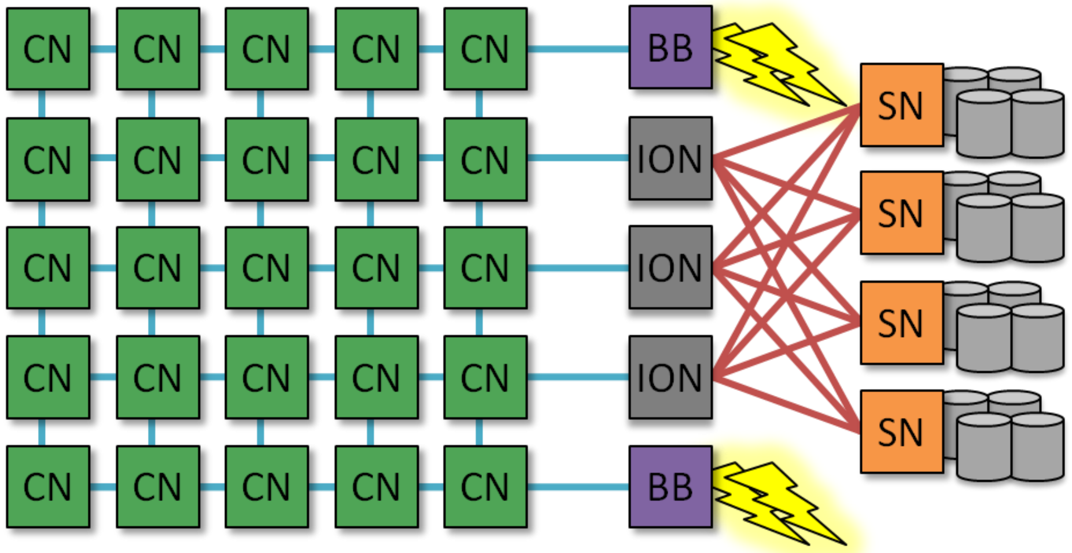
Node-local Burst Buffer

Burst Buffers: Yet Another Storage Layer

The “Rabbit” way:

- Each Rabbit node consists of *N* SSDs and one processor.
- Two Rabbit nodes sit in each rack; Each is directly connected to all compute nodes within the same rack.
- Provide a shared address space to all compute nodes.





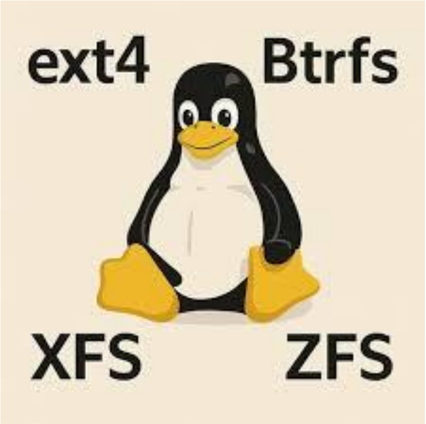
BB: Burst Buffer Node

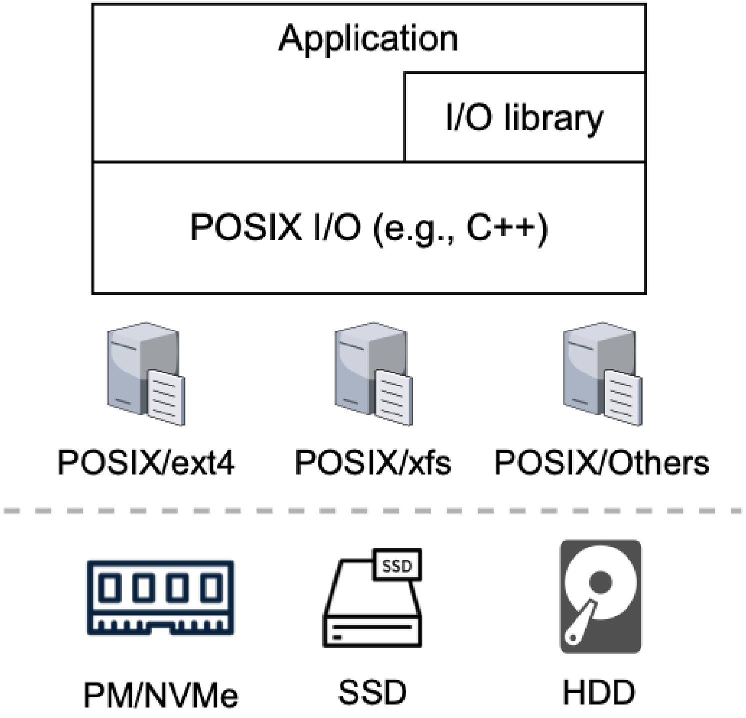
Burst Buffer File System

- Need file systems to manage these burst buffer devices.
  - Node-local burst buffer is managed using local file systems.
  - Rabbit is managed by Lustre.
- Nearly all widely-deployed file systems are **POSIX-complaint**. However, POSIX is not a good fit for HPC.

lustre File System

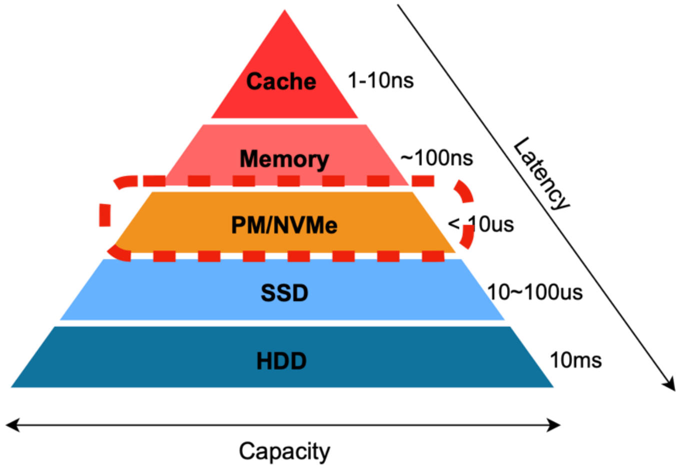
IBM Spectrum Scale

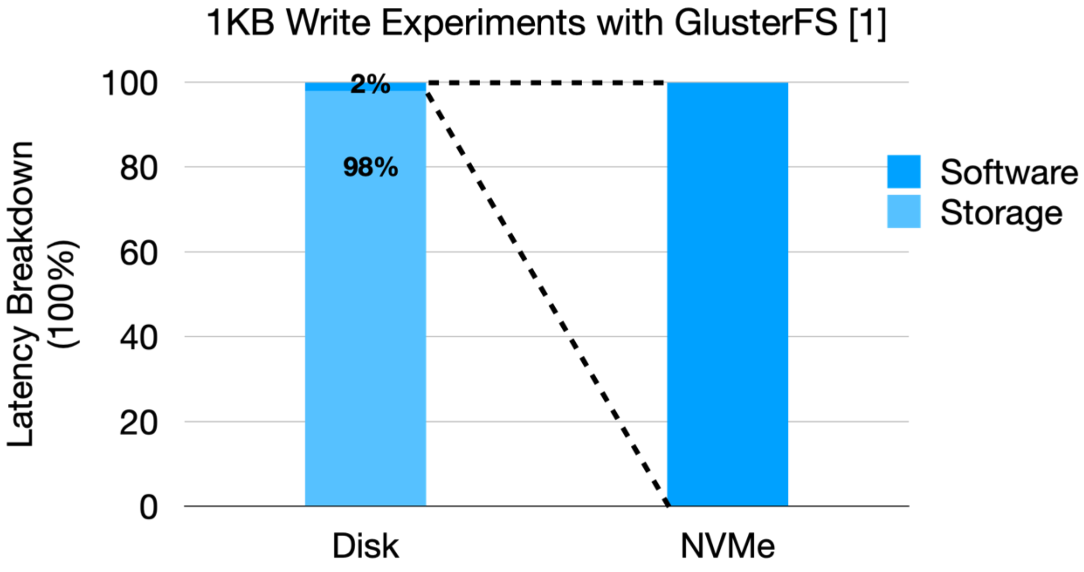




POSIX – What's So Bad About It?

- Designed decades ago (early 1970s) for use **by a single machine with a single storage device**.
  - Designed for compatibility, not performance.
- The primary limitation is its **strict consistency semantics requirements.**
  - Write needs to become immediately visible to any subsequent reads.
  - It is extremely expensive to maintain POSIX consistency at scale.





Applications do not need POSIX semantics

Study of 17 HPC Applications:

- None of them requires the strict POSIX semantics.

- *Chen Wang, Kathryn Mohror, and Marc Snir. "File System Semantics Requirements of HPC Applications", HPDC, 2021.*
- *Chen Wang, Kathryn Mohror, and Marc Snir. "Formal Definitions and Performance Comparison of Consistency Models for Parallel File Systems", TPDS, 2024.*

Applications do not need POSIX semantics

Users can run their applications on file systems with weaker models for a better I/O performance.

Applications do not need POSIX semantics

Great! Now let's ditch POSIX and find a better alternative.

The Rise of Non-POSIX Systems

Non-POSIX File System	Desgin Goal	Institution
UnifyFS	Improving scientific simulation checkpointing performance	Lawrence Livermore National Lab, USA
Spectral	Improving scientific simulation checkpointing performance	Oak Ridge National Lab, USA
LLIO	Fugaku I/O accleration layer	RIKEN, Japan
Gfarm/BB	Node-local burst buffer system	University of Tsukuba, Japan
SuperFS	Accelerating I/O using NVMe and RDMA.	Tsinghua University, China

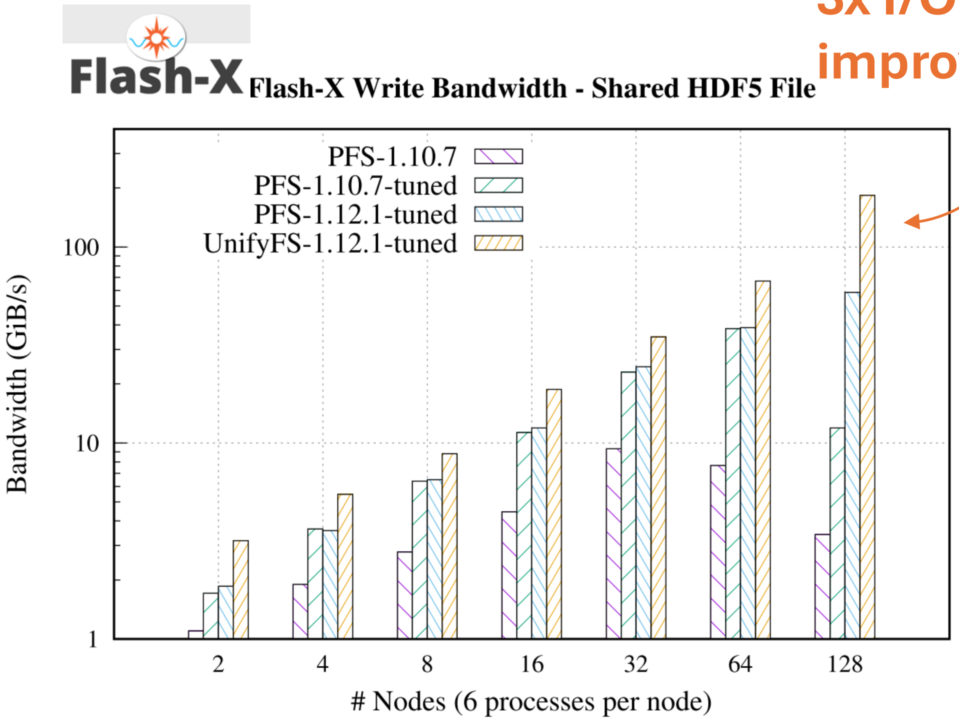
Example Non-POSIX Systems:

UnifyFS: A specialized burst buffer parallel file system for supercomputers. Designed **for write-heavy HPC applications**.

- https://github.com/LLNL/UnifyFS

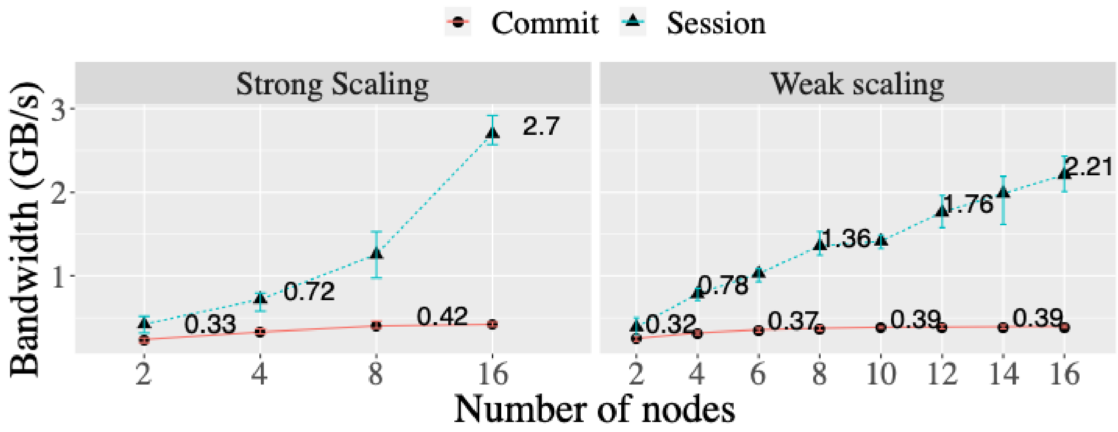
DYAD: is a data streamer optimized for deep learning (DL) training.

- https://github.com/flux-framework/dyad



3x I/O bandwidth improvement

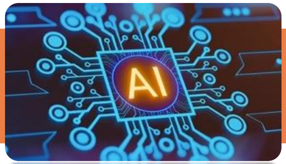
Impact of Consistency Models on I/O Performance in DL Training.



5x I/O bandwidth improvement.

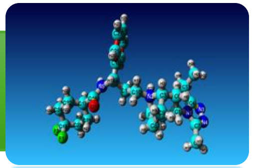
I/O Demands in AI vs Traditional HPC Workloads

AI Workloads



- Often treat the underlying I/O subsystem as a block box.
    - Not a big issue for small-scale models.
    - **Prefer memory semantics**
- **Primary focus is data throughput**
    - Can data be fed/dumped fast enough to keep GPUs busy?
- Random read is the major data access pattern.
- Don’t need POSIX semantics.

HPC Workloads

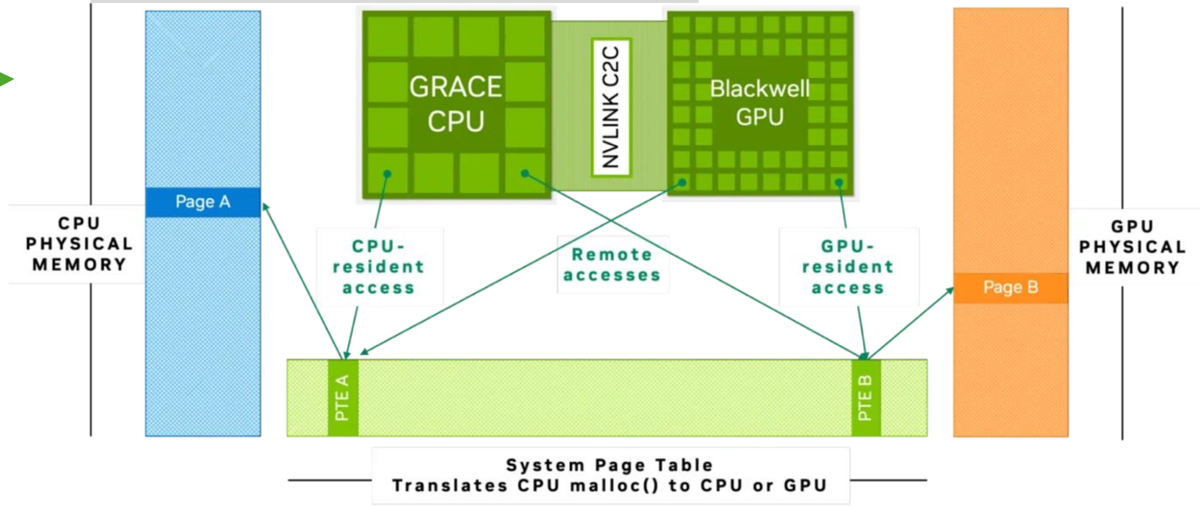


- **Filesystem-aware**: read input files, periodically write snapshot & checkpoint files.
    - Actively interacting with I/O libraries and the underlying persistent storage
    - **Storage semantics**
- **Larger write volume** than AI workloads
- Care about persistency, capacity, and bandwidth
- Don’t need POSIX semantics.

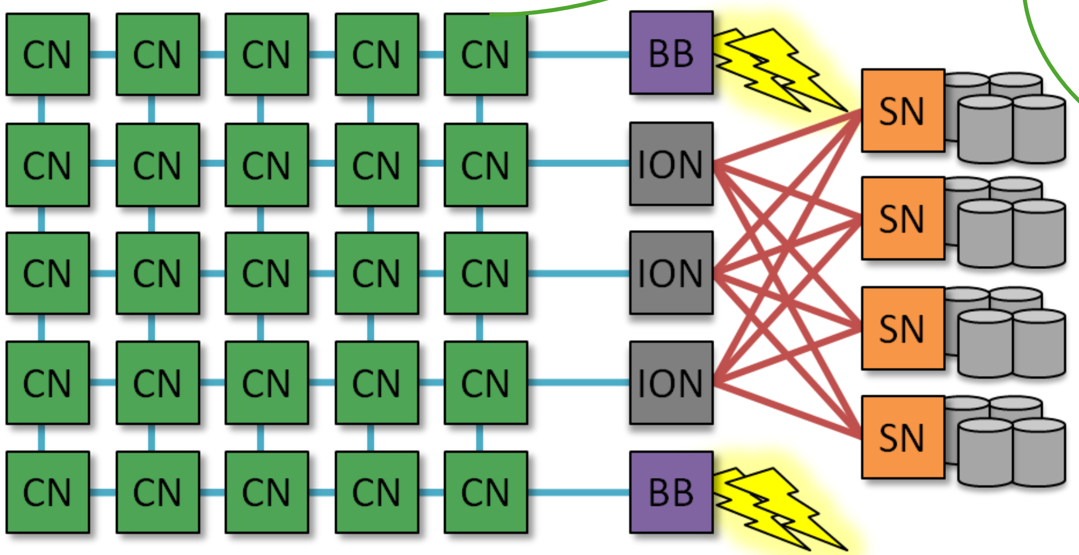
From Memory to Storage: A Deepening Hierarchy

- GPU HBM
- CPU DRAM
- CXL Memory/SSD
- NVMe/SSD
  - GPUDirect Storage
  - NVMe over Fabric
- Hard Drive

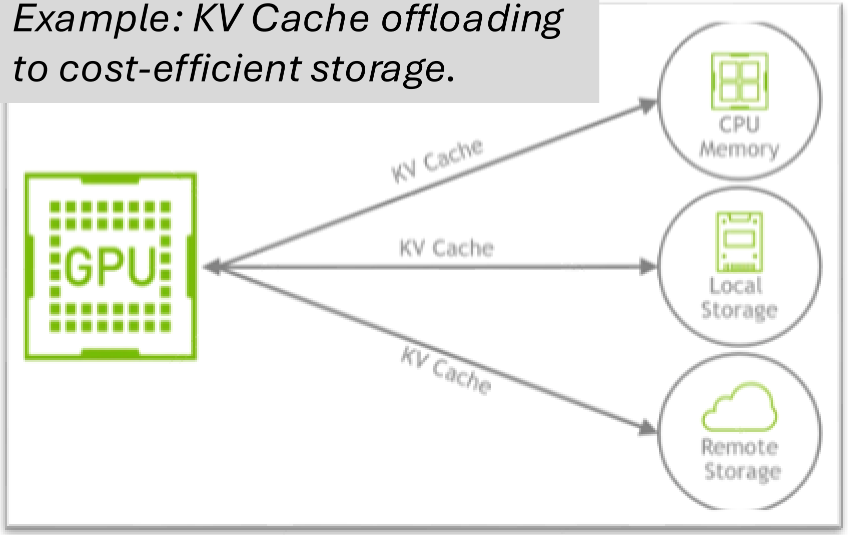
Unified CPU GPU memory



Burst Buffer Architecture



Example: KV Cache offloading to cost-efficient storage.



Key Takeaways

- Emerging AI and HPC workloads are placing increasing pressure on storage systems.
- HPC systems continue to scale rapidly; they have begun integrating new storage devices (e.g., NVMe and CXL SSD) as an additional capacity expansion and I/O acceleration layer.
  - Such new technologies are blurring the boundary between memory and storage, yet they are still programmed separately.
  - The utilization of new storage layers remains low.
- AI workloads treat HPC storage system as block box, while this black box was not originally built for optimizing AI workloads.
  - A better collaboration and understanding is required.
- For the increasingly deep storage architecture, POSIX is no longer sufficient to manage them.
  - Exploring alternative models may provide a short-term remedy.

Who drives the next generation of HPC Storage design? Should we wait for big nations?

Thank You & Questions?

Chen Wang
chen.wang@ntu.edu.sg