SCA/HPCAsia 2026 Tutorial: Accelerating HPC Application I/O with Fast Node-Local Storage

Chen Wang

chen.wang@ntu.edu.sg

Assistant Professor

Nanyang Technological University

Jae-Sung Yeom

yeom1@llnl.gov

Computer Scientist

Lawrence Livermore National Laboratory

Contributers: Hariharan Devarajan, Cameron Stanavige, Michael Brim, Kathryn Mohror, Michela Taufer, Ian Lumsden

Tutorial link: https://wangchen.io/unifyfs-dyad-tutorial

Schedule

Tutorial Link:
https://wangchen.io/unifyfs-dyad-tutorial

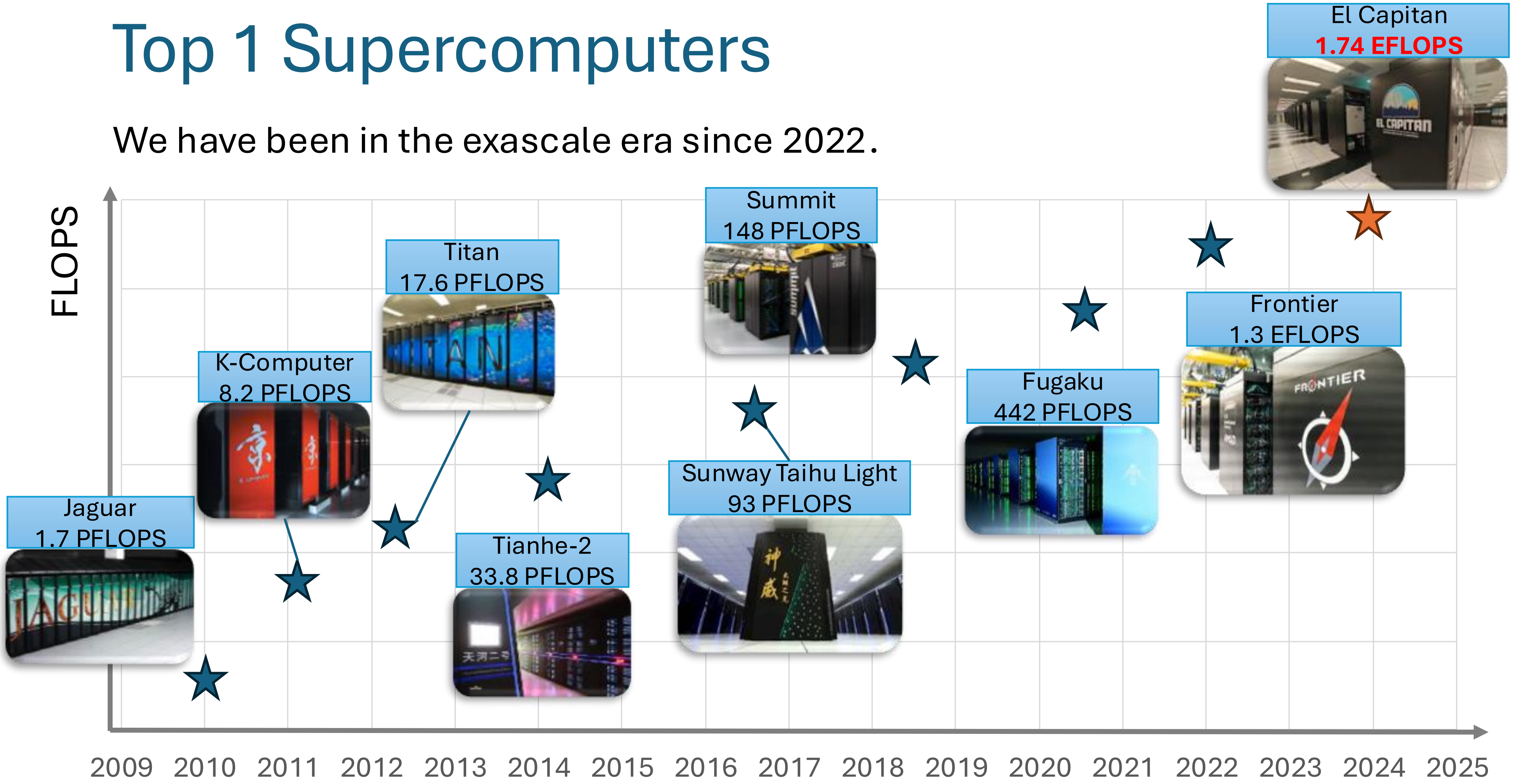
Time	Event
13:30 - 14:00	Introduction and Motivation
14:00 - 14:30	Deep Dive: UnifyFS
14:30 - 14:50	Break
14:50 - 15:20	Deep Dive: DYAD
15:20 - 15:50	DYAD Walkthrough
15:50 - 16:15	UnifyFS Walkthrough
16:15 - 16:30	Wrap-up and Q&A

Cluster account request:
https://forms.gle/Ya1WRVn9cxdRSMmXA

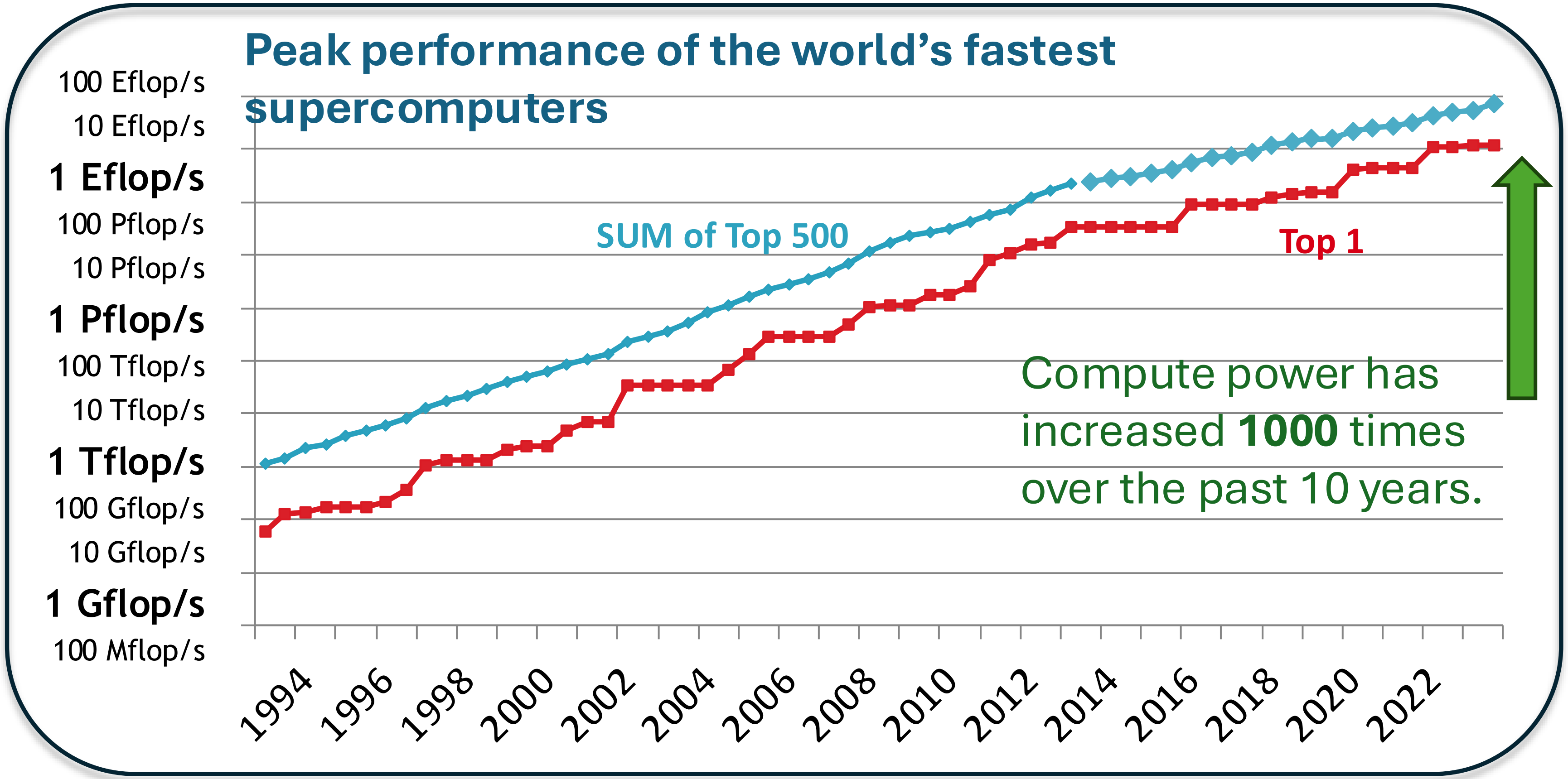
ssh -p 2223 username@13.215.163.223
password: P@ssw0rd

Top 1 Supercomputers

We have been in the exascale era since 2022.

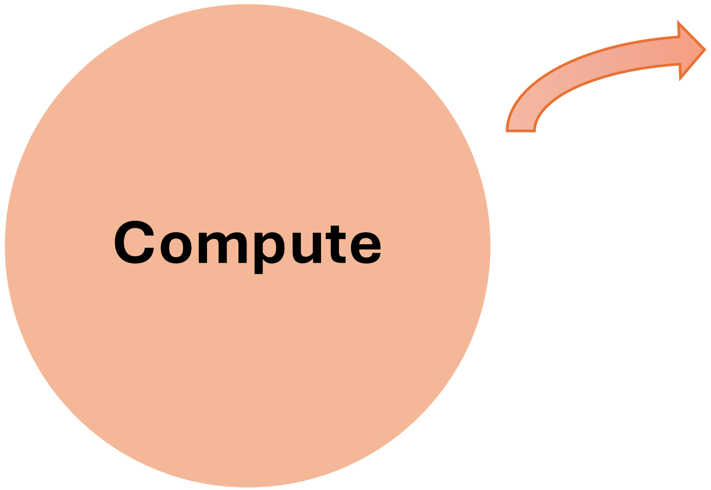


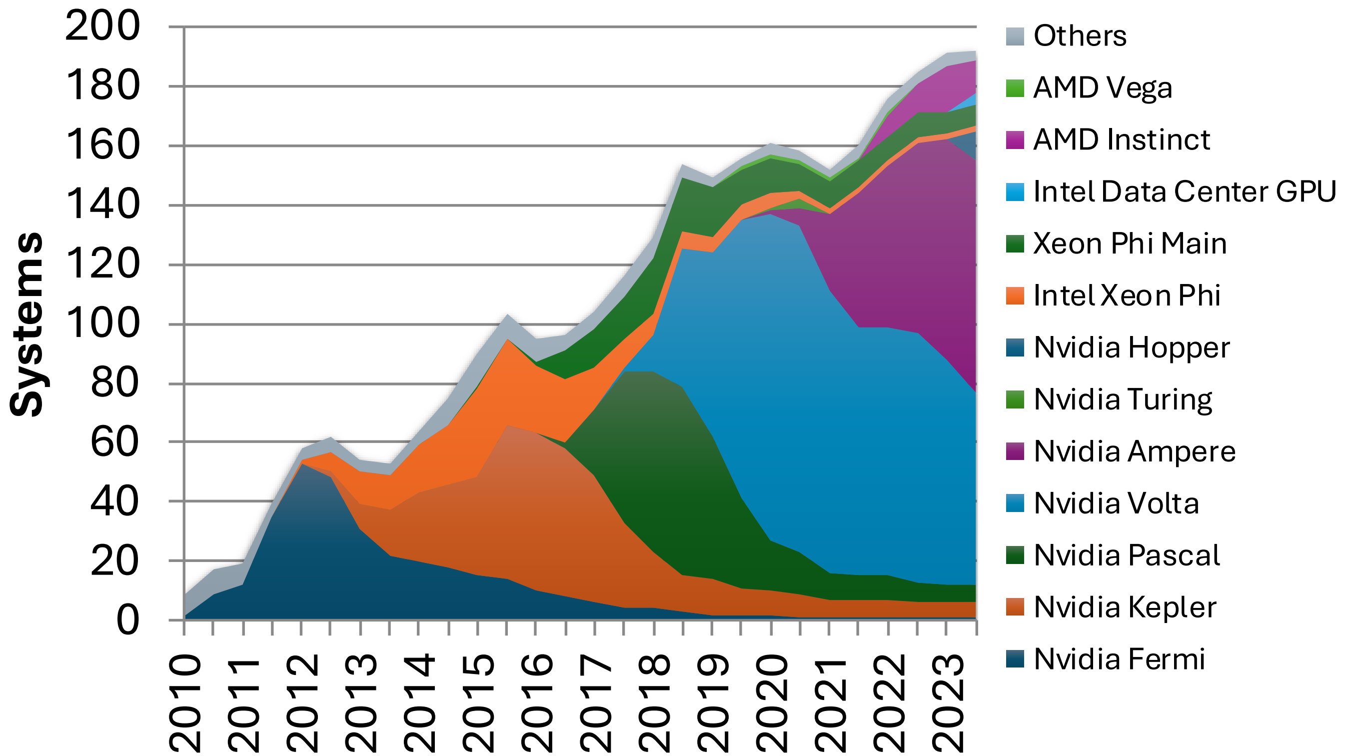
Compute Power Has Increased Significantly



Data Source: Top500.org

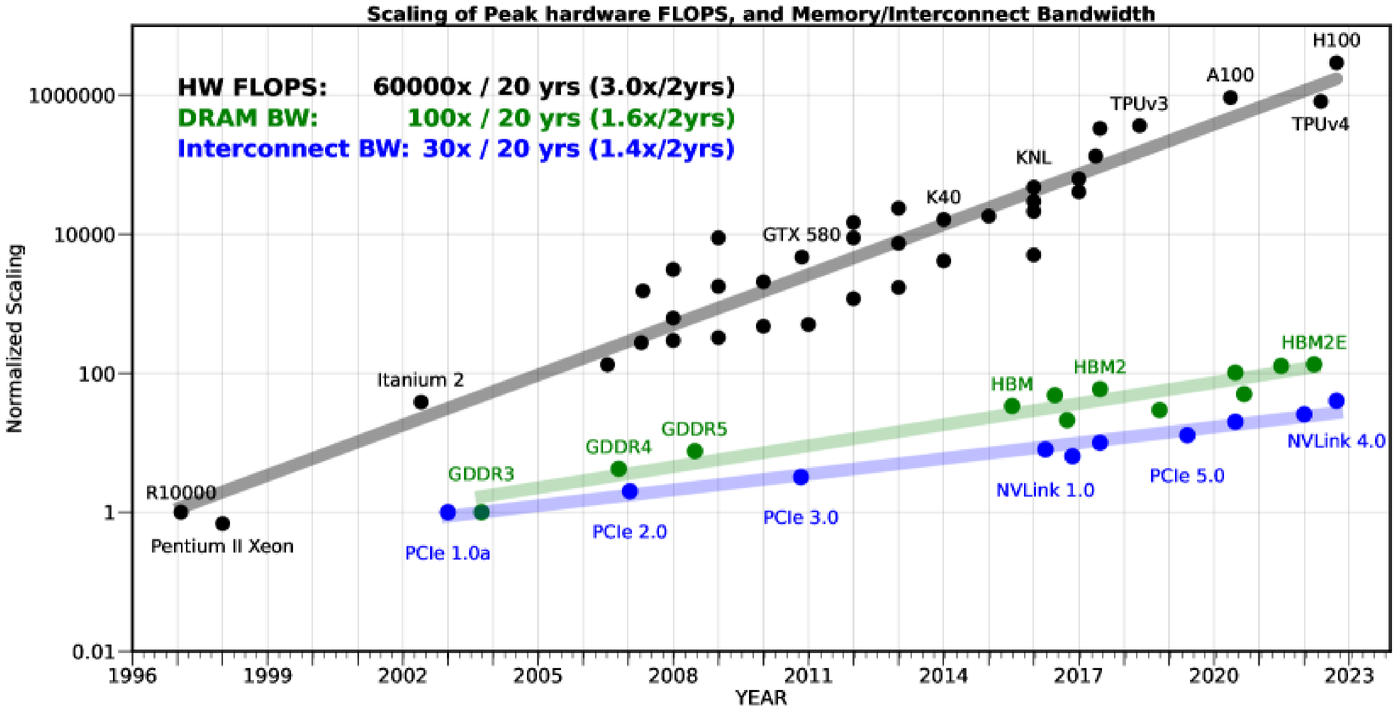
GPUs Have Become the Primary Workhorse





Over 1/3 of top 500 systems have accelerators

AI and Memory Wall

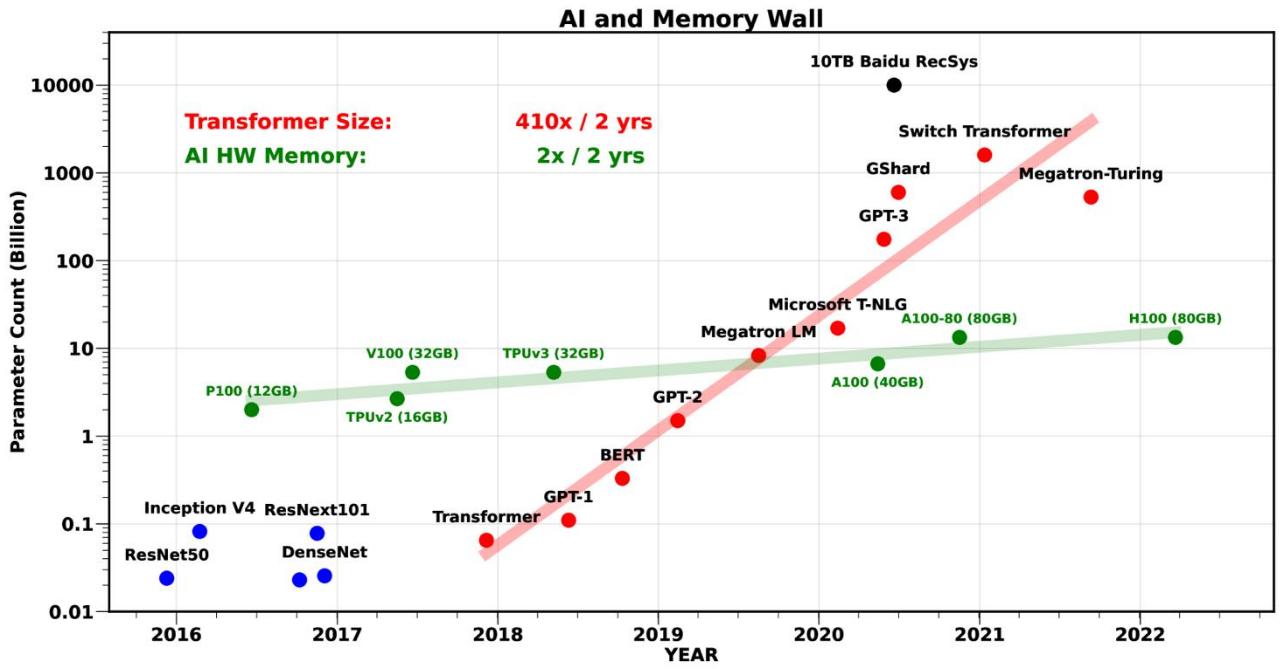


GPU FLOPS vs. Memory Bandwidth

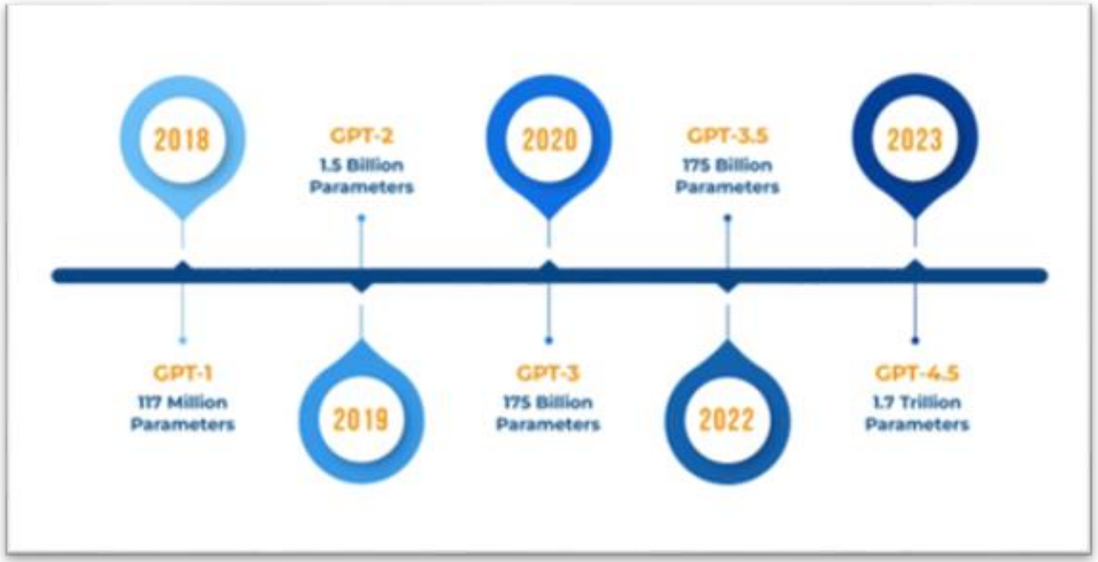
The performance gap is expected to grow at 50% per year.

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.

AI and Memory Wall



Transformer Size vs. Memory Capacity

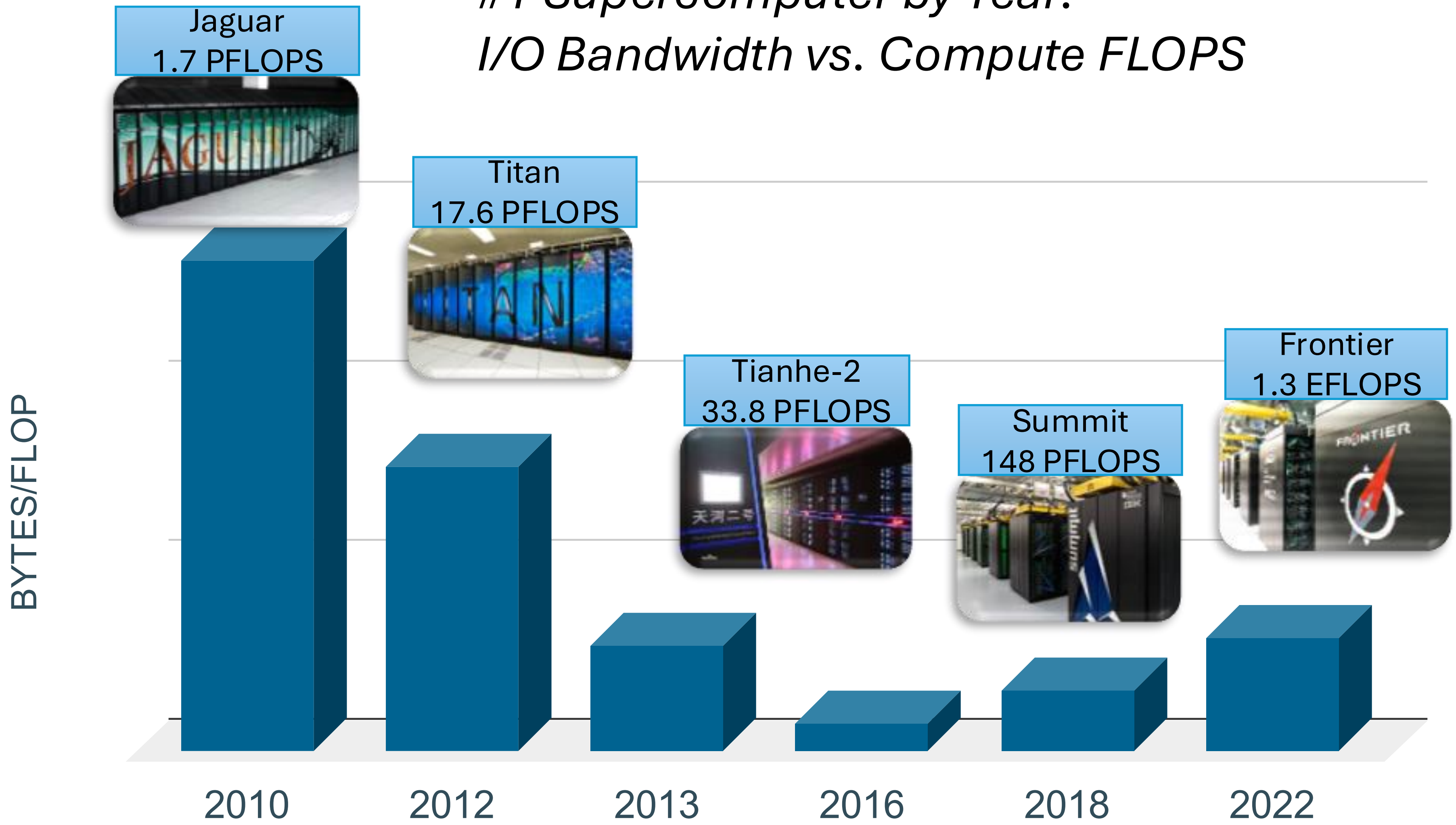


The Evolution of GPT Models

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.

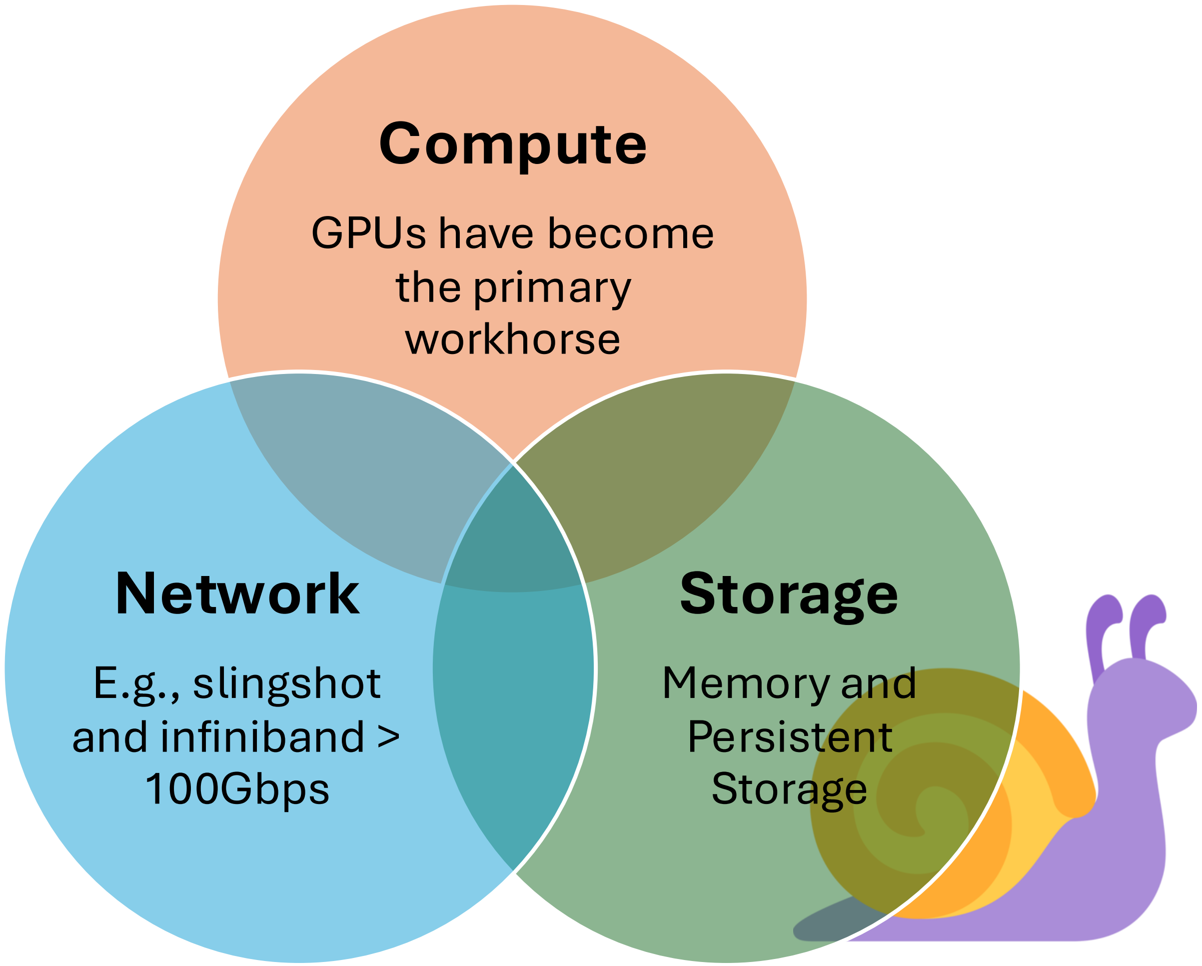
I/O is even Slower

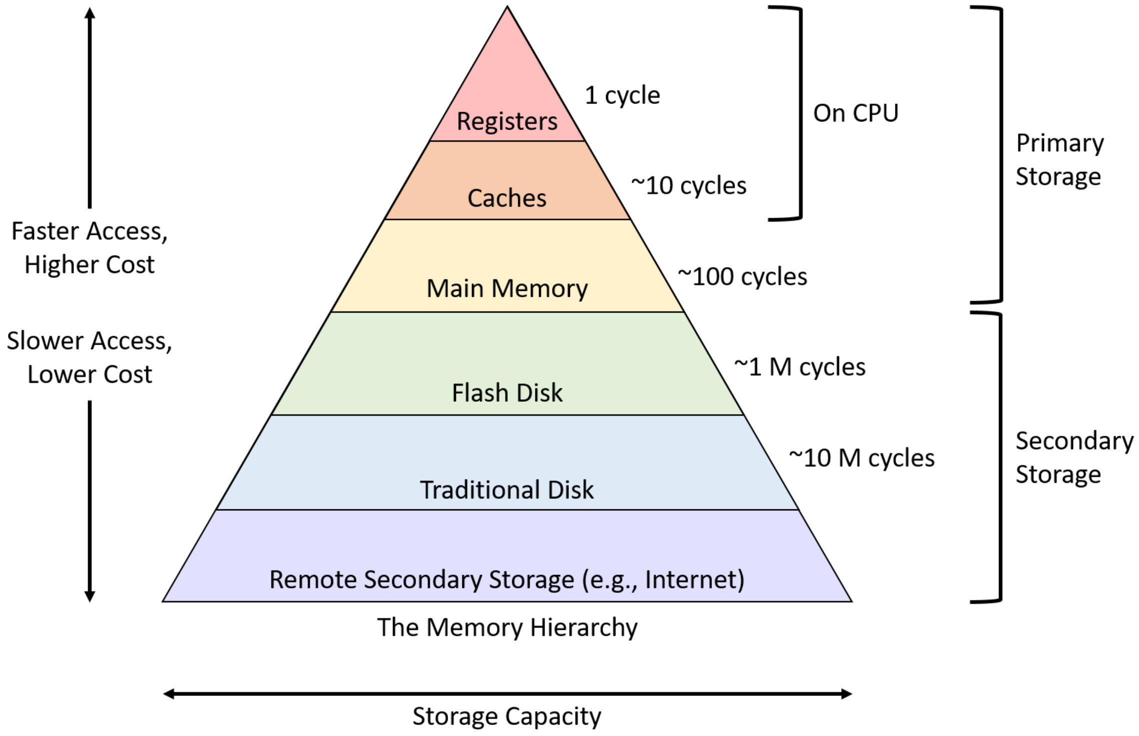
#1 Supercomputer by Year:
I/O Bandwidth vs. Compute FLOPS



Tutorial link: https://wangchen.io/unifyfs-dyad-tutorial

The Deep Storage Hierarchy



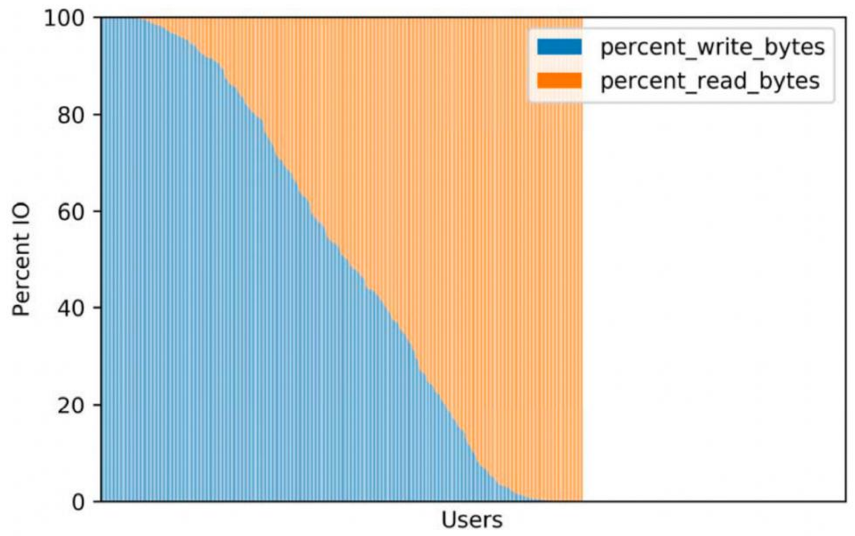


The entire storage hierarchy is getting deeper and more complex, and the boundary between memory and storage is steadily blurring.

I/O Subsystem Inefficiency

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

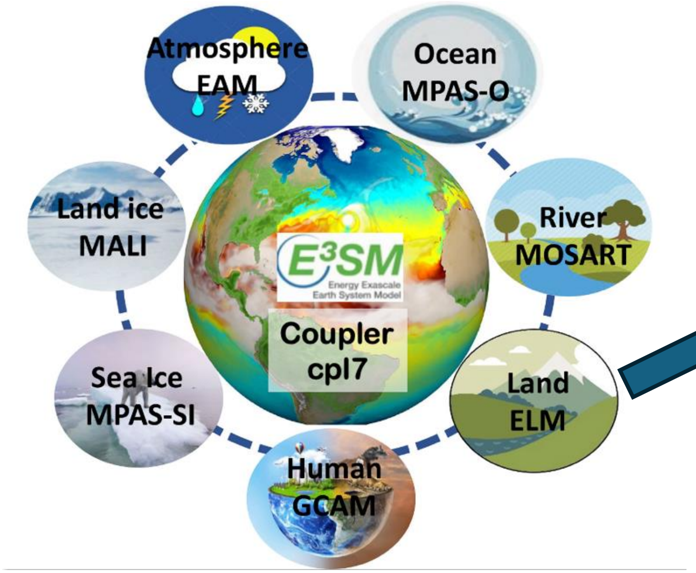
- on average, jobs which performing I/O spread I/O activities across 78.8% of their runtime.
- less than 22% write-intensive jobs perform efficient writes.
- HPC jobs are no longer write dominated

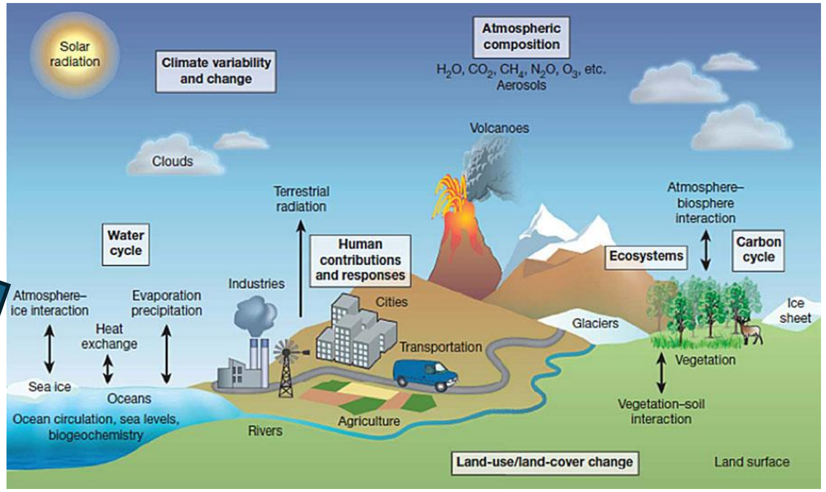


Percentage I/O (Write vs. Read) by User

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)

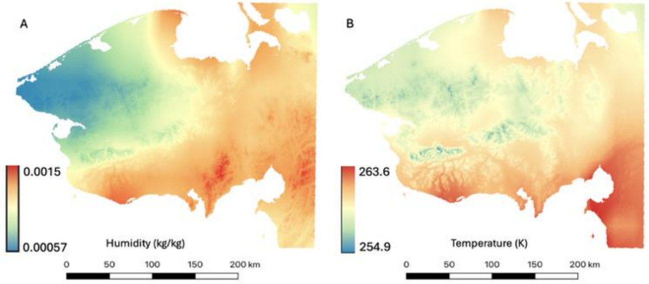




File	Size
Surface (I)	188 GB
Forcing (I)	1.4 TB
History (O)	134 GB
Restart (O)	4.2 TB

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)

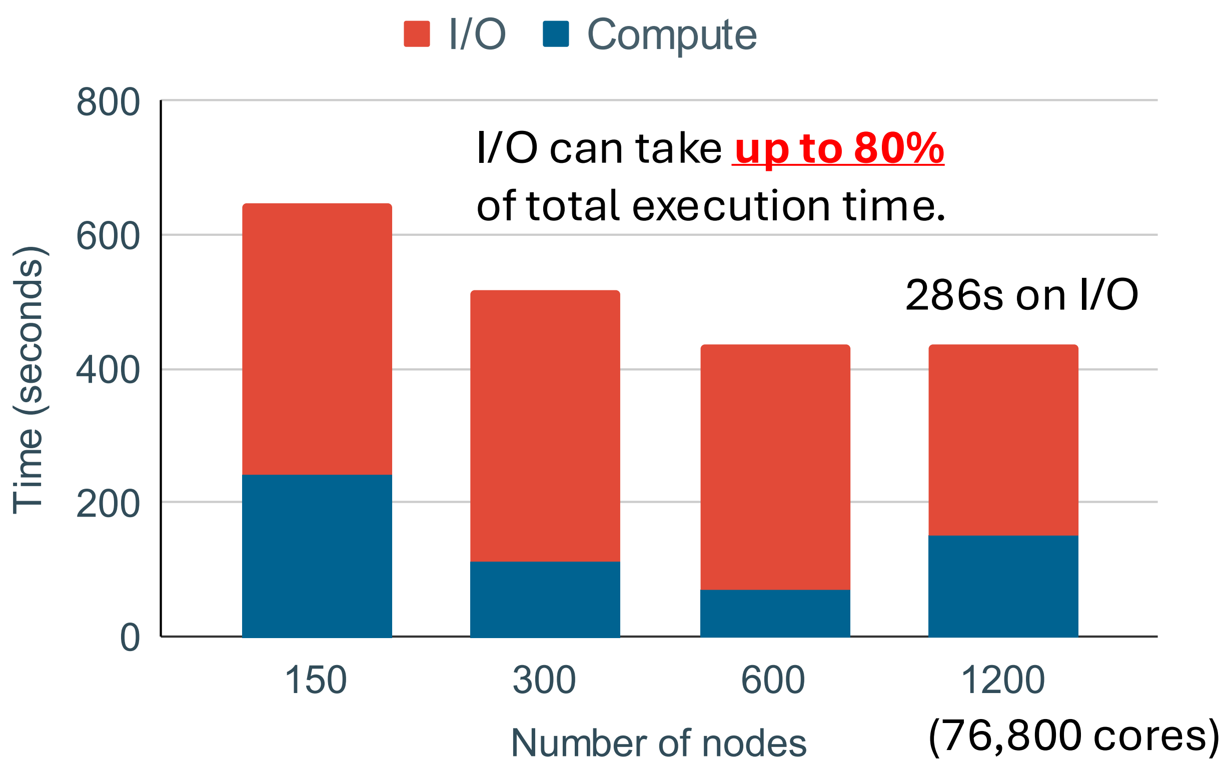


2025 CCGRID SCALE Challenge Finalist: Dali Wang, Chen Wang, Qinglei Cao, and et.al. "Scaling Ultrahigh-Resolution E3SM Land Model for Leadership-Class Supercomputers".

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

The E3SM Example:

E3SM

↳ SCORPIO

↳ PnetCDF

↳ MPI-IO

↳ File system (POSIX)

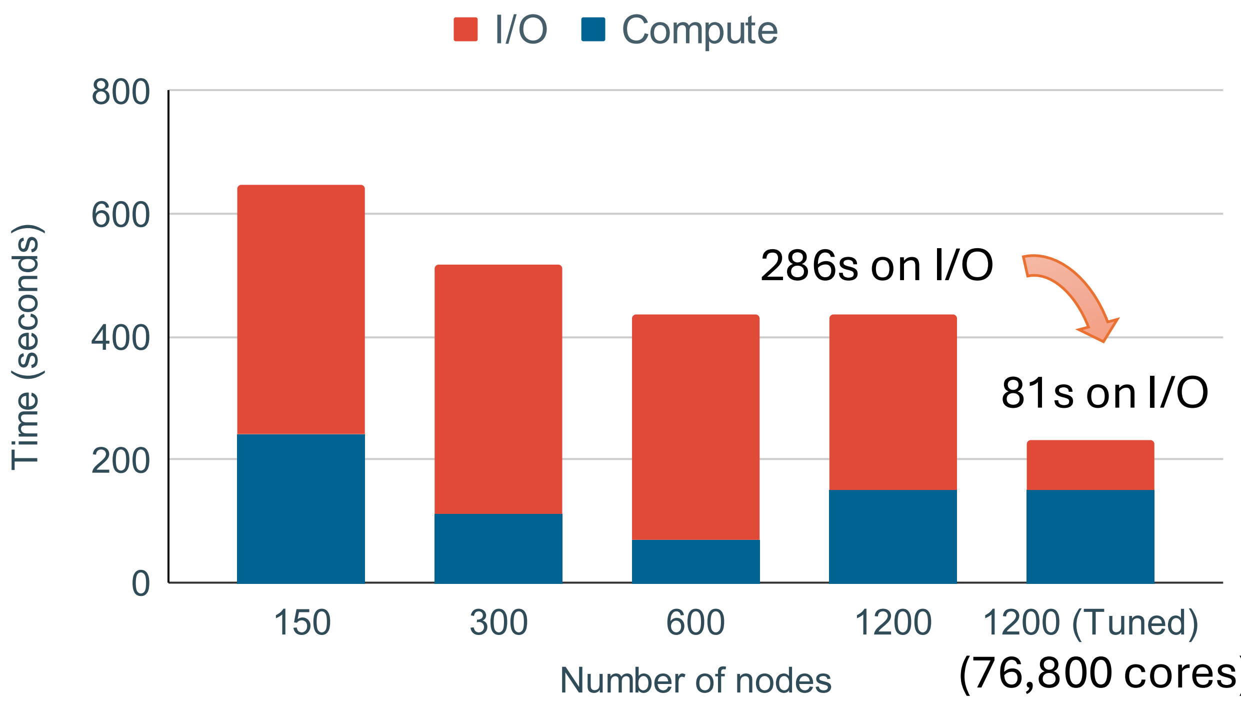
I/O library and middleware

I/O Library and middleware

HPC File System

Tutorial link: https://wangchen.io/unifyfs-dyad-tutorial

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".

Optimizing HPC File Systems?

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

Tutorial link: https://wangchen.io/unifyfs-dyad-tutorial

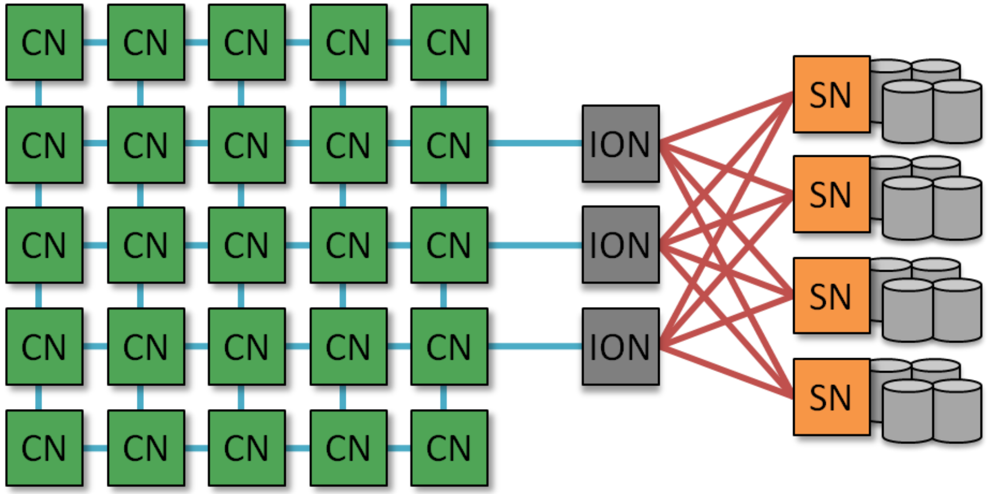
Optimizing HPC File Systems?

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.

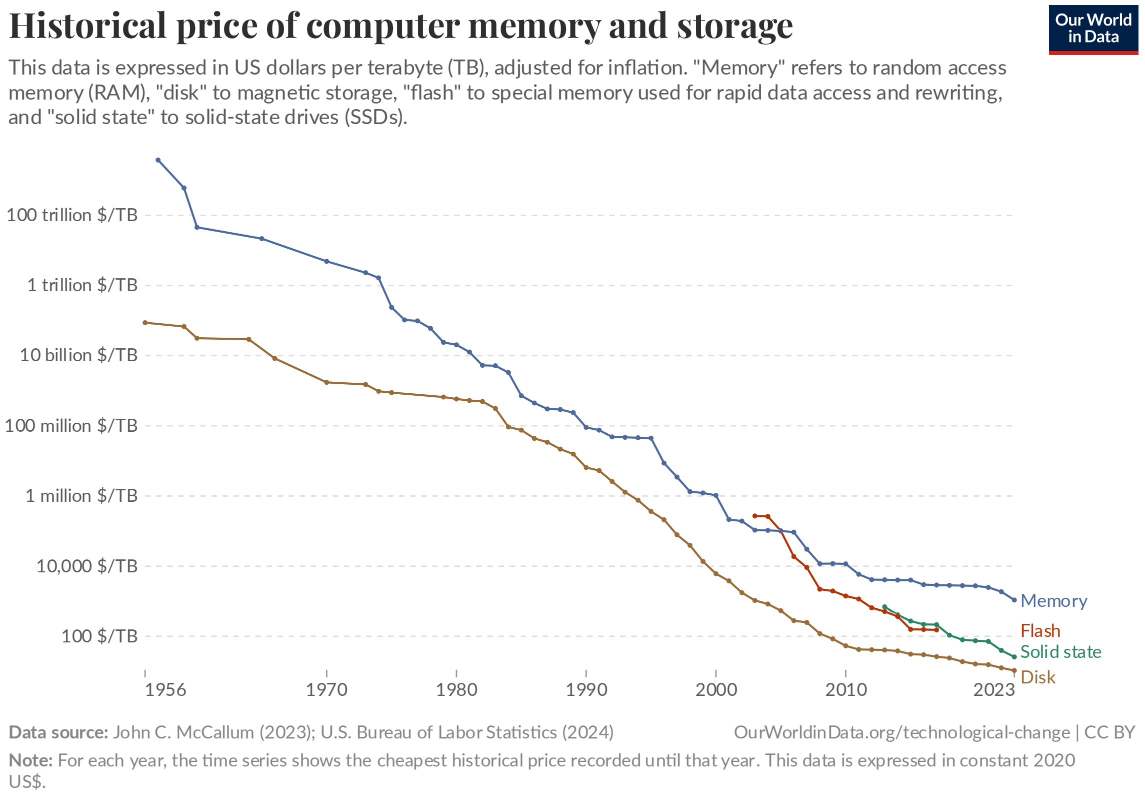
Burst Buffers: Yet Another Storage Layer

"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



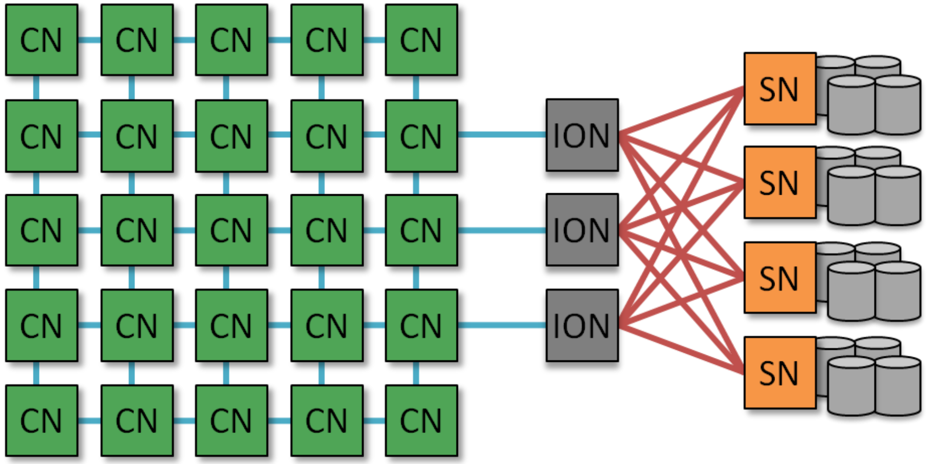
Figures courtesy of Glenn K. Lockwood.
https://blog.glennklockwood.com/2017/03/reviewing-state-of-art-of-burst-buffers.html

Tutorial link: https://wangchen.io/unifyfs-dyad-tutorial

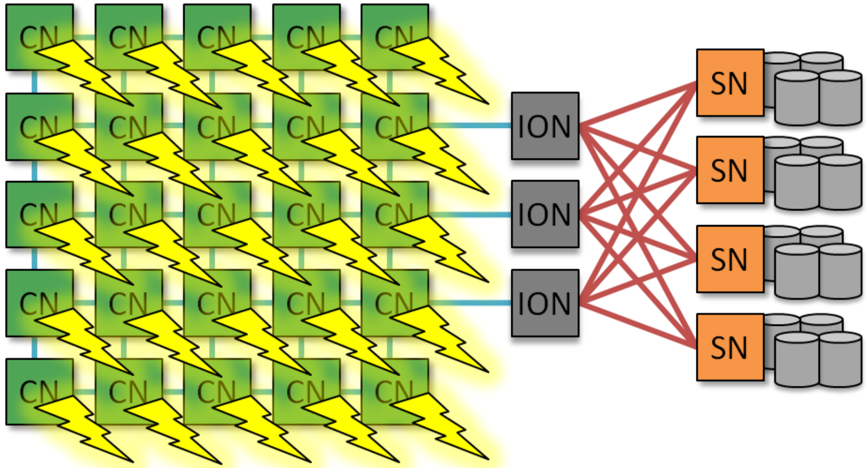
Burst Buffers: Yet Another Storage Layer

Node-local burst buffer:

- Attach one SSD to each compute node.
    - Scales linearly.
- **Only accessiable 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

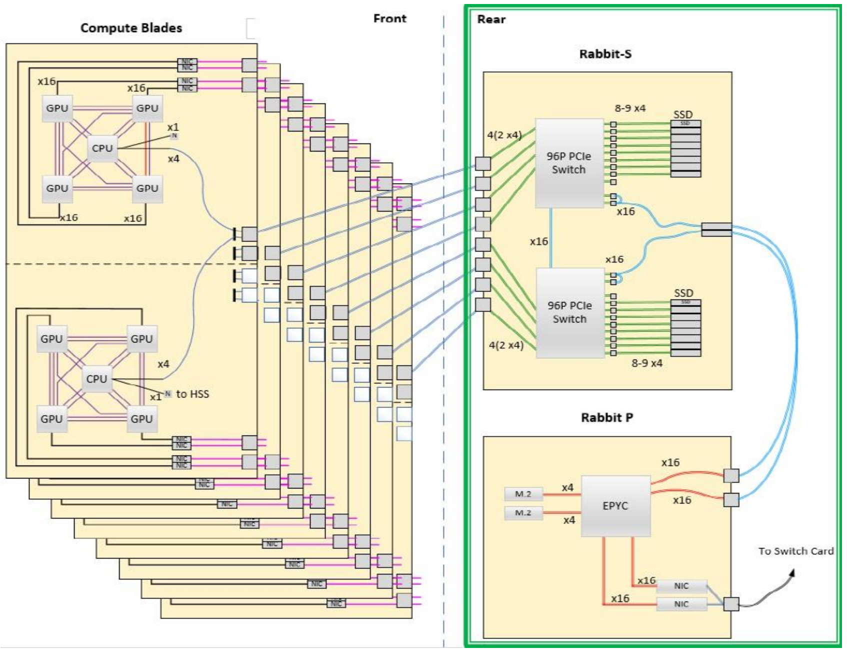


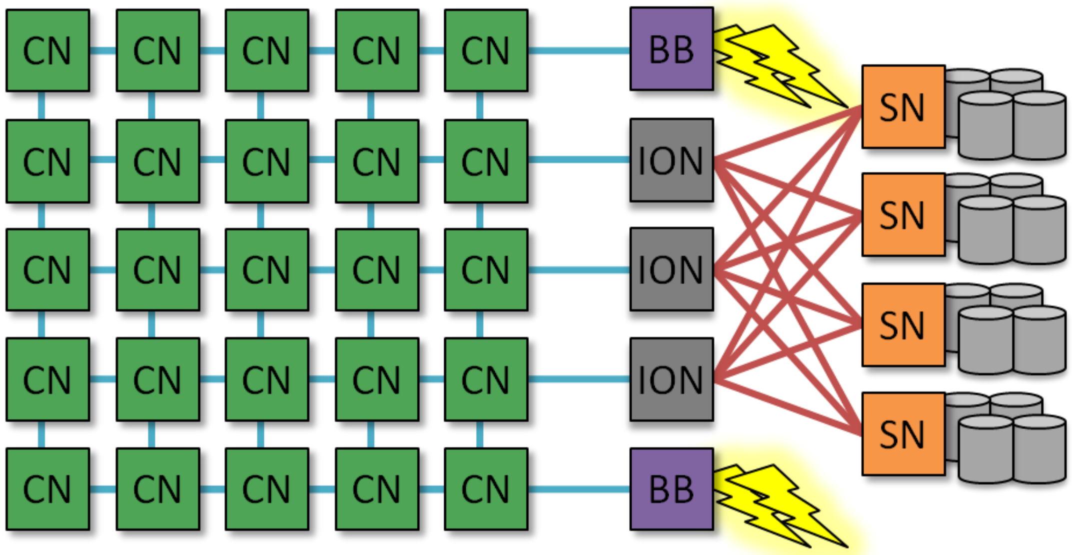
Node-local Burst Buffer

Burst Buffers: Yet Another Storage Layer

The “Rabbit” way:

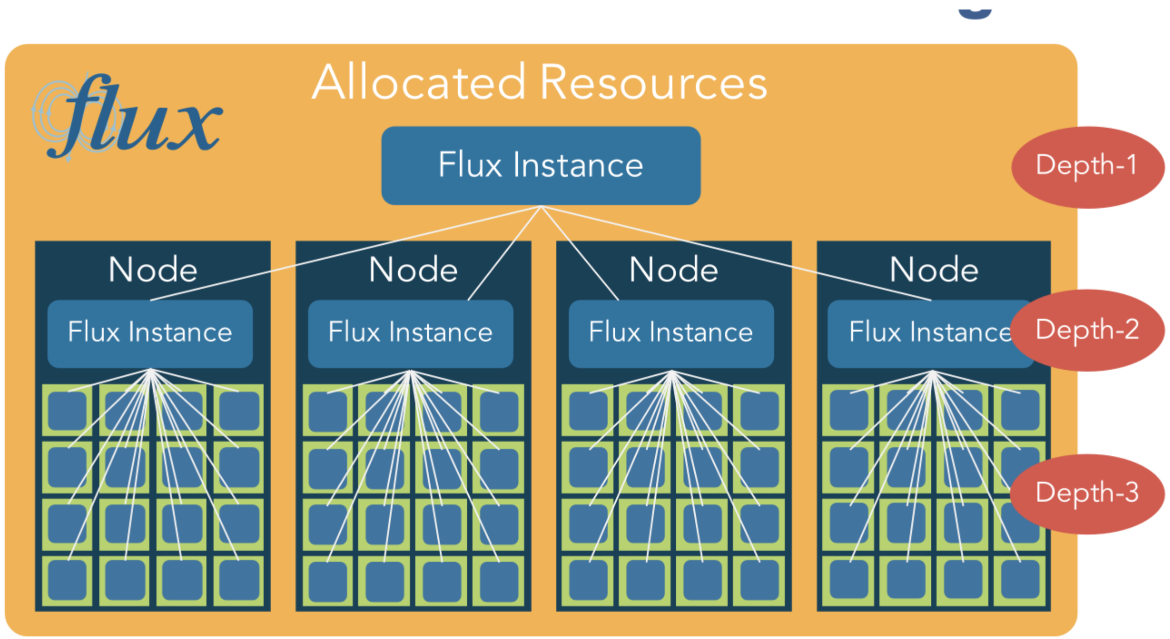
- Each Rabbit node consists of *N* SSDs and one processor.
- Two Rabbit nodes sit in each rack; Each is directly connected to all compute nodes within the same rack.
- Provide a shared address space to all compute nodes.





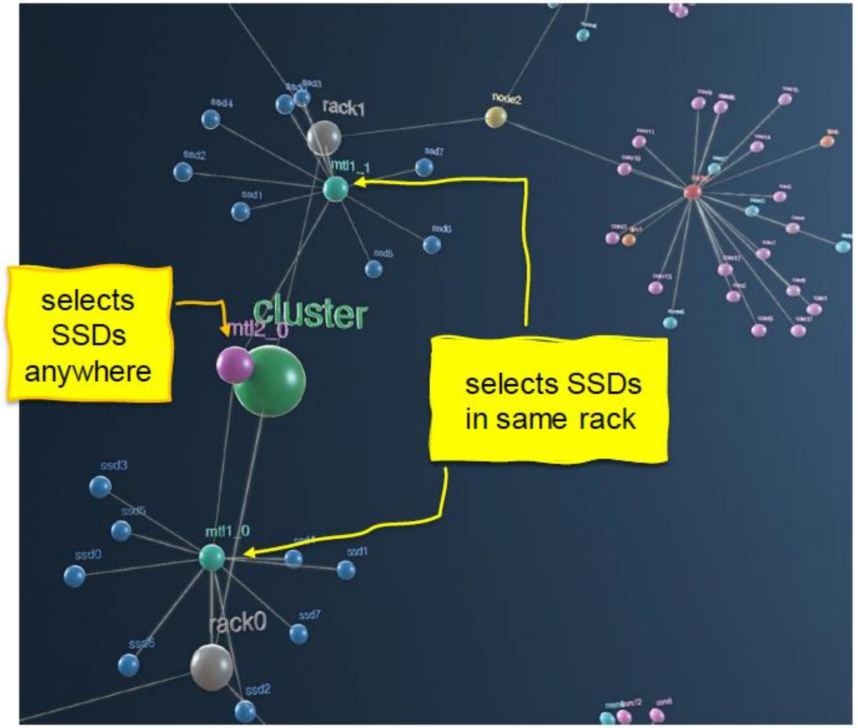
BB: Burst Buffer Node

Flux hierarchically manages of resources and jobs for scalability



- Hierarchical and modular management
- Mitigates the centralized scheduler bottleneck
- Deployed at El Capitan
  - Ranked 1st in top500 supercomputers

Flux's graph-based resource representation addresses challenges with Rabbit storage system



- Better suited for dynamic and heterogenous resources
- Rabbit is a near-compute disaggregate storage system that can be dynamically allocated per job as either shared or node-local.

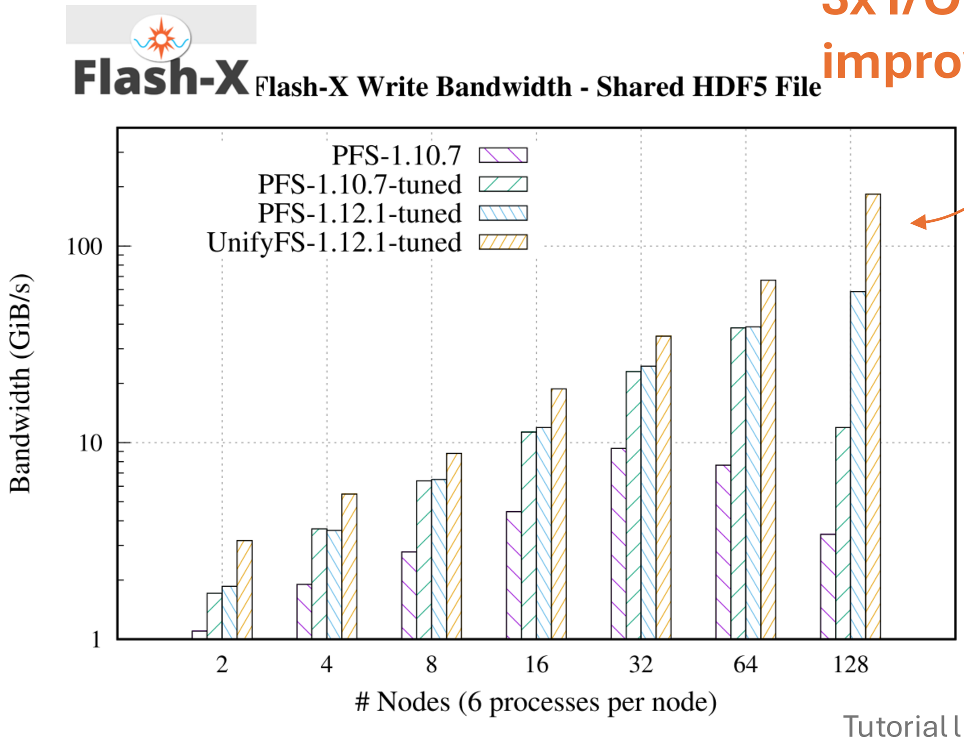
Tutorial link: https://wangchen.io/unifyfs-dyad-tutorial

UnifyFS and DYAD

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

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

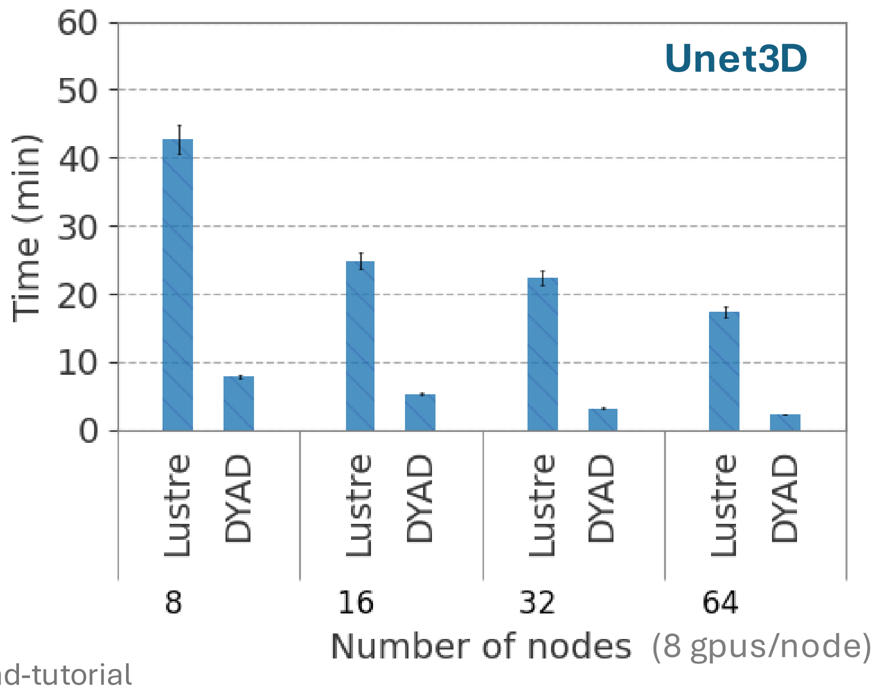
3x I/O bandwidth improvement



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

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

Up to 8x training time improvement



Tutorial link: https://wangchen.io/unifyfs-dyad-tutorial

Thank You!

UnifyFS: A specialized burst buffer parallel file system for supercomputers. Our goal is to provide easy, portable, and fast support for I/O-intensive applications.

- https://github.com/LLNL/UnifyFS
- *Michael Brim, Adam Moody, Seung-Hwan Lim, Ross Miller, Swen Boehm, Cameron Stanavige, Kathryn Mohror, Sarp Oral, "UnifyFS: A User-level Shared File System for Unified Access to Distributed Local Storage," 37th IEEE International Parallel & Distributed Processing Symposium (IPDPS), May 2023.*

unifyFS

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

- https://github.com/flux-framework/dyad
- DYAD data movement coordination strategy reduces network contention and thus improving data movement
- DYAD employs a hierarchical sample discovery technique to isolate metadata accesses, and that improves lookup throughput
- DYAD's novel streaming RPC over RDMA protocol boosts inter-node access
- DYAD optimizes large-scale DL workloads by 8.2x as compared to state-of-the-art solutions.

Call for Papers

7th International Workshop on Extreme-Scale Storage and Analysis (ESSA 2026), held in conjunction with IPDPS (New Orleans, May 2026)!

<u>Deadline: Feb. 6, 2026.</u>

Special Issue

New Advances in Parallel and Distributed Computing

Message from the Guest Editors

This Special Issue focuses on emerging advances that enhance the performance, scalability, and intelligence of parallel and distributed computing systems. It emphasizes new technologies and paradigms in high-performance computing (HPC) in the era of AI, particularly in the context of the convergence of HPC, cloud, and edge computing. Topics of interest include, but are not limited to, the following:

- Advanced architectures for the HPC–cloud–edge computing continuum.
- Parallel and distributed algorithms for large-scale AI and large language models (LLMs).
- Data-driven optimization techniques for complex scientific and AI workflows.
- Cross-facility resource management and scheduling.
- Workflow management frameworks for distributed and federated infrastructures.
- Performance modeling, optimization, and benchmarking of distributed systems.
- Disaggregated storage and high-throughput data flow optimization.
- Green and sustainable computing strategies for data-intensive workloads.
- Integration of HPC, quantum, and AI accelerators into distributed platforms.
- Emerging applications, including digital twins, autonomous systems, IoT, and 6G networking.

Electronics

an Open Access Journal by MDPI

Impact Factor 2.6
CiteScore 6.1

mdpi.com/si/267658

Electronics
Editorial Office
MDPI, Grosspeteranlage 5
4052 Basel, Switzerland
Tel: +41 61 683 77 34
electronics@mdpi.com

mdpi.com/journal/electronics