# CS550 "Advanced Operating Systems"

## Instructor: **Professor Xian-He Sun**

- Email: sun@iit.edu
- Office: SB235C
- Class time: Monday, Wed., 3:15pm-4:30pm, SB113
- Office hour: Monday, Wednesday, 4:45-5:45pm
- http://www.cs.iit.edu/~sun/cs550.html

- TA: Mr. Hua Xu, Email: hxu40@hawk.iit.edu
- Office Hour: 11am - 12pm, Tuesday
- meet.google.com/kfp-pysg-cat
- Office Hour: 12pm - 1pm, Tuesday & Thursday
  meet.google.com/bnn-eqao-htg

- Blackboard*:*
  - http://blackboard.iit.edu
- Substitute lecturer:
  - Anthony Kougkas, assistant research professor
  - akougkas@hawk.iit.edu

# What This Course is About

- Understanding the *fundamental concepts* of distributed systems, in particular, distributed operating system
- Mastering *distributed programming* techniques
  - Multithreading, RPC, RMI, Sockets, etc.
- Understanding the *general principles* of distributed paradigms
  - object-based systems, file systems, web-based systems

# Prerequisite

- CS450 “Operating Systems”
  - CS430 “Intro to Algorithms”
- Familiar with
  - Java or C/C++ programming
  - Networking programming
    - Sockets
    - Multithreaded
    - RPC, Java RMI
  - Basic concepts of computer architecture

# Course Schedule

- Week 1: Introduction
- Week 2: Architecture
- Week 3: Processes and threads
- Week 4: Communication: RPC,RMI,…
- Week 5: Naming
- Week 6: Synchronization and coordination
- Week 7: Consistency and replication
- Week 8: Fault tolerance
- Week 9: Security
- Week 10: Distributed file systems
- Week 11: Exam
- Week 12-15: Final project presentation

Subject to change

# Course Materials

- Required:
  - Maarten van Steen and Andrew S. Tanenbaum “Distributed Systems” (3rd edition) 2017
  - Soft copy is available online. Hard copy is available via Amazon.
- Supplemental readings:
  - Andrew S. Tanenbaum and Maarten van Steen Distributed Systems: Principles and Paradigms (2nd edition) Prentice Hall, 2007.
  - George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair Distributed Systems: Concepts and Design (fifth edition) Pearson, 2011.

# Misc. Course Details

- Grading
  - 33% -- Homework, Programming Assignment, and Participation
  - 37% -- Exam
  - 30% -- Term Project and Presentation
- Use the course blackboard
  - Announcements
  - Lecture notes
  - Assignments
  - Discussion
  - …

# Term Project

- See http://www.cs.iit.edu/~sun/html/report2.html
- A two-page project proposal due by Jan. 29, 2024
- Final project report is due on April 25, 2024

- Example topics
  - Study and practice of some middleware programming-environment, software packages, applications.
  - Study and analyze some distributed environment, architectures, and network structures.
  - Study the distributed solution of certain application package, algorithm, and system software.
  - Performance metric, measurement, and benchmark.
  - Study and practice of some visualization tools.
  - Survey of certain topics.
  - Any other topics that are relevant to this course.

Will have more on the topics in Jan. 24 lecture

# Misc. Course Details

- The course has 4 sections:
  - 01 for main campus (in person)
  - 02 for Ph.D. Systems Qualifier Exam
  - 03 for Internet
  - 04 for Beacon students

- Final grading:
  - Based on curve
  - For students in section 02
    - To get a grade of A, weighted total >= the 80$^{th}$ percentile

# Policies

- Collaboration policy
  - Encouraged for high level concepts and understanding the courses materials
  - but …..

- Cheating policy
  - Copying all or part of another student's homework
  - Allowing another student to copy all or part of your homework
  - Copying all or part of code found in a book, magazine, the Internet, or other resource

# Policies

- IIT Code of Academic Honesty [link]
- All violations of academic integrity will be reported to academichonesty@iit.edu
- Sanctions for violations of academic integrity
  - **Expulsion from a course.** The student is assigned a punitive failing grade of 'E' for the course and can no longer participate in the course or receive evaluation of coursework from the instructor.
  - **Suspension.** Suspension is a status assigned for various periods of time in which a student's enrollment is interrupted. A suspended student may not attend day or evening classes, participate in student activities, or live in campus housing. A suspended student may apply for reinstatement at the end of the period of suspension. If reinstated, the student may be placed on disciplinary probation for a period of time designated by the DDAD.
  - **Expulsion**. Expulsion is the complete severance of association with the University. Notation of the violation of the Code is made on the student's transcript

# Policies

- You can report sexual harassment at iit.edu/incidentreport, anonymously.
- IIT Sexual Harassment & Discrimination Info

  - Illinois Tech prohibits all sexual harassment, sexual misconduct, and gender discrimination by any member of our community. This includes harassment among students, staff, or faculty. Sexual harassment of a student by a faculty member or sexual harassment of an employee by a supervisor is particularly serious. Such conduct may easily create an intimidating, hostile, or offensive environment.

  - Illinois Tech encourages anyone experiencing sexual harassment or sexual misconduct to speak with the Office of Title IX Compliance for information on support options and the resolution process.

# AI Policies

- A guideline on academic honesty and generative AI
- A guide on assigning writing and generative AI
- Galvin Library's guide to AI

# Any Questions?

# Personal Introduction

- Research interests
  - Parallel and Distributed Processing
  - Memory and I/O system (Big Data Systems)
  - Performance Analysis and Modeling

- Research group:
  - Gnosis Research Center (GRC) for accelerating data-driven discovery
  - http://grc.iit.edu
  - Weekly Research seminar

# The Gnosis Research Center

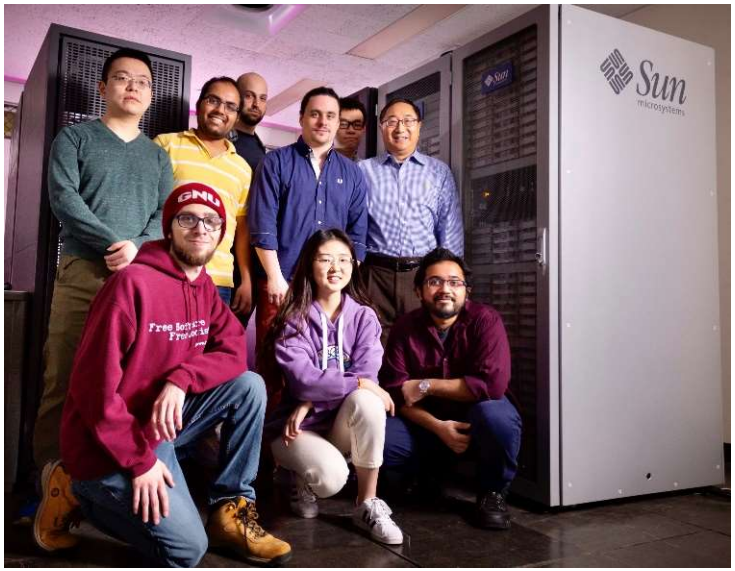
***http://grc.iit.edu***

**Specialize** in high performance software systems for big data applications

*(System Group, GRC Center)*

**Supported** by:

- NSF, DoE, NASA, and industry




# Core Members

**Dr. Kun Feng**
Research Software Engineer

**Eneko Gonzalez**
Research Software Engineer

**Dr. Aparna Sasidharan**
Research Software Engineer

Anthony Kougkas, Research Professor

**Wiam Amine**
PhD Student

**Keith Bateman**
PhD Candidate

**Vadim Biryukov**
PhD Student

**Jaime Cernuda**
PhD Candidate

**Luke Logan**
PhD Candidate

**Xiaoyang Lu**
PhD Candidate

**Neeraj Rajesh**
PhD Candidate

**Meng Tang**
PhD Candidate

**Hua Xu**
PhD Student

**Jie Ye**
PhD Candidate

**Izzet Yildirim**
PhD Candidate

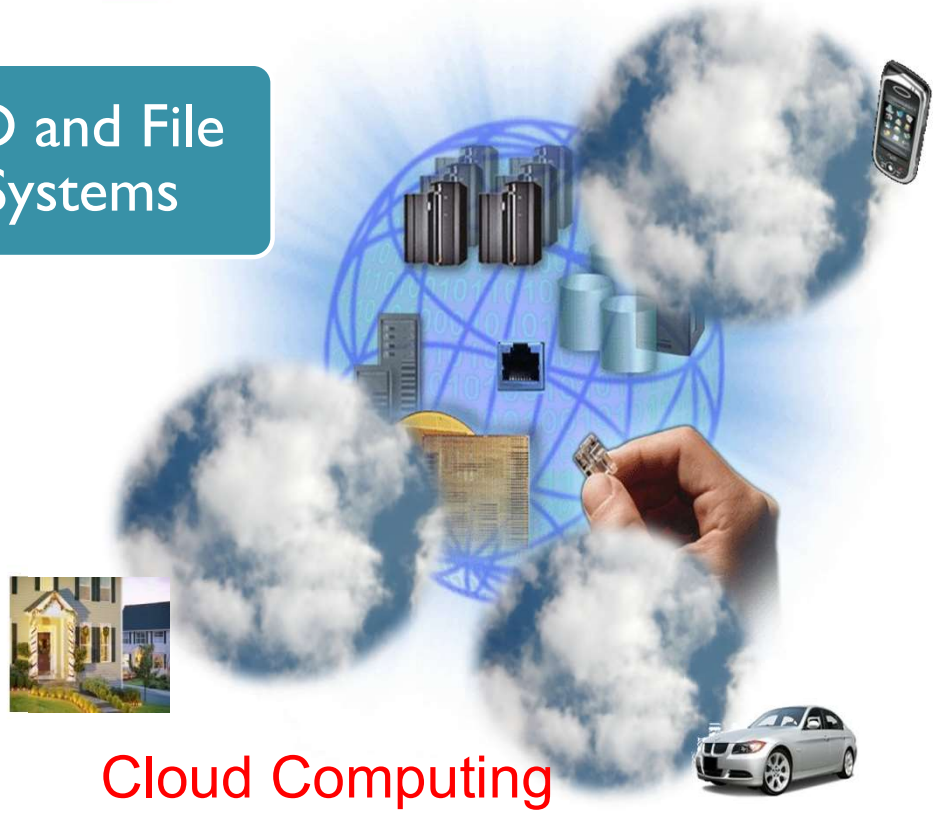
# Advanced Computing Computing at SCS

Data intensive computing

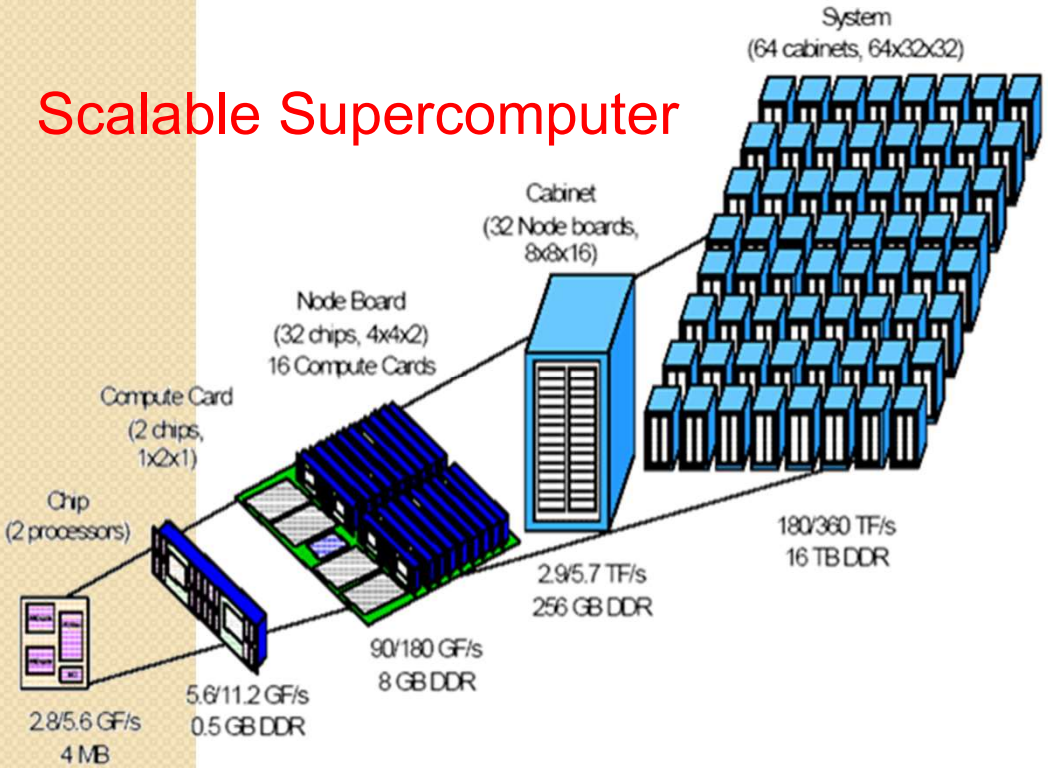
Memory Systems

I/O and File Systems

- Big data management



Scalable Supercomputer



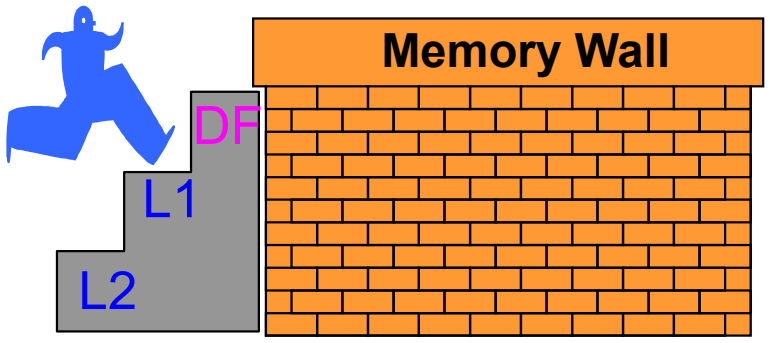

Cloud Computing

**System Research**

# Hot Issues

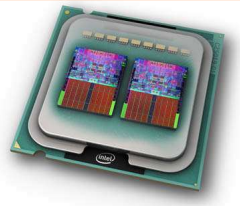
- AI and Deep Learning
- Big Data
- *High Performance and Could Computing*

COMPUTING POWER



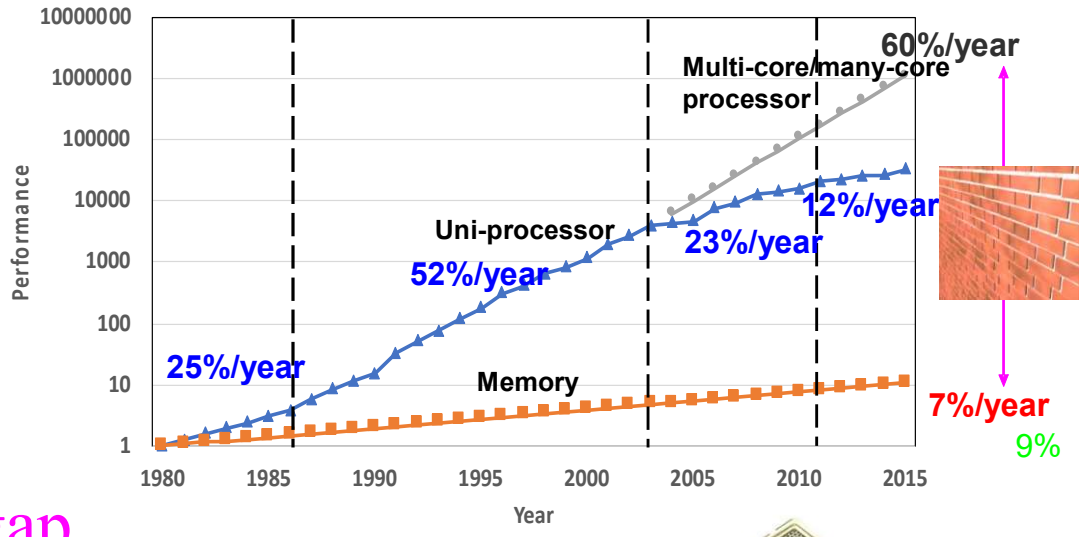
# Why Data Centric ? The Memory-wall Problem

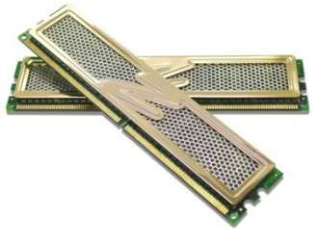
- Processor performance increases rapidly
  - Uni-processor: ~52% until 2004
  - Aggregate multi-core/many-core processor performance even higher since 2004
- Memory: ~7% per year
  - Storage: ~6% per year
- Processor-memory speed gap keeps increasing



Source: Intel

Leiserson, Charles E., et al. "There's plenty of room at the Top: What will drive computer performance after Moore's law?." *Science* 368.6495 (2020).



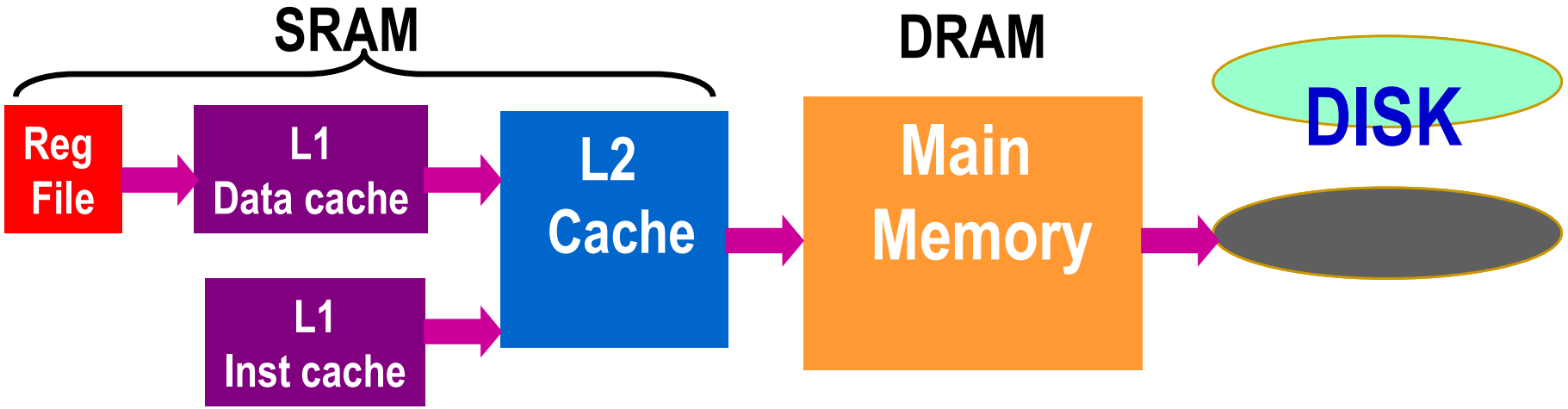



Source: OCZ

Memory-bounded speedup (1990), Memory wall problem (1994)

X.-H. Sun, and L. Ni, "Another View of Parallel Speedup," Proc. of ACM-IEEE Supercomputing'90, NY, Nov. 1990

# Memory-wall Solution: Memory Hierarchy

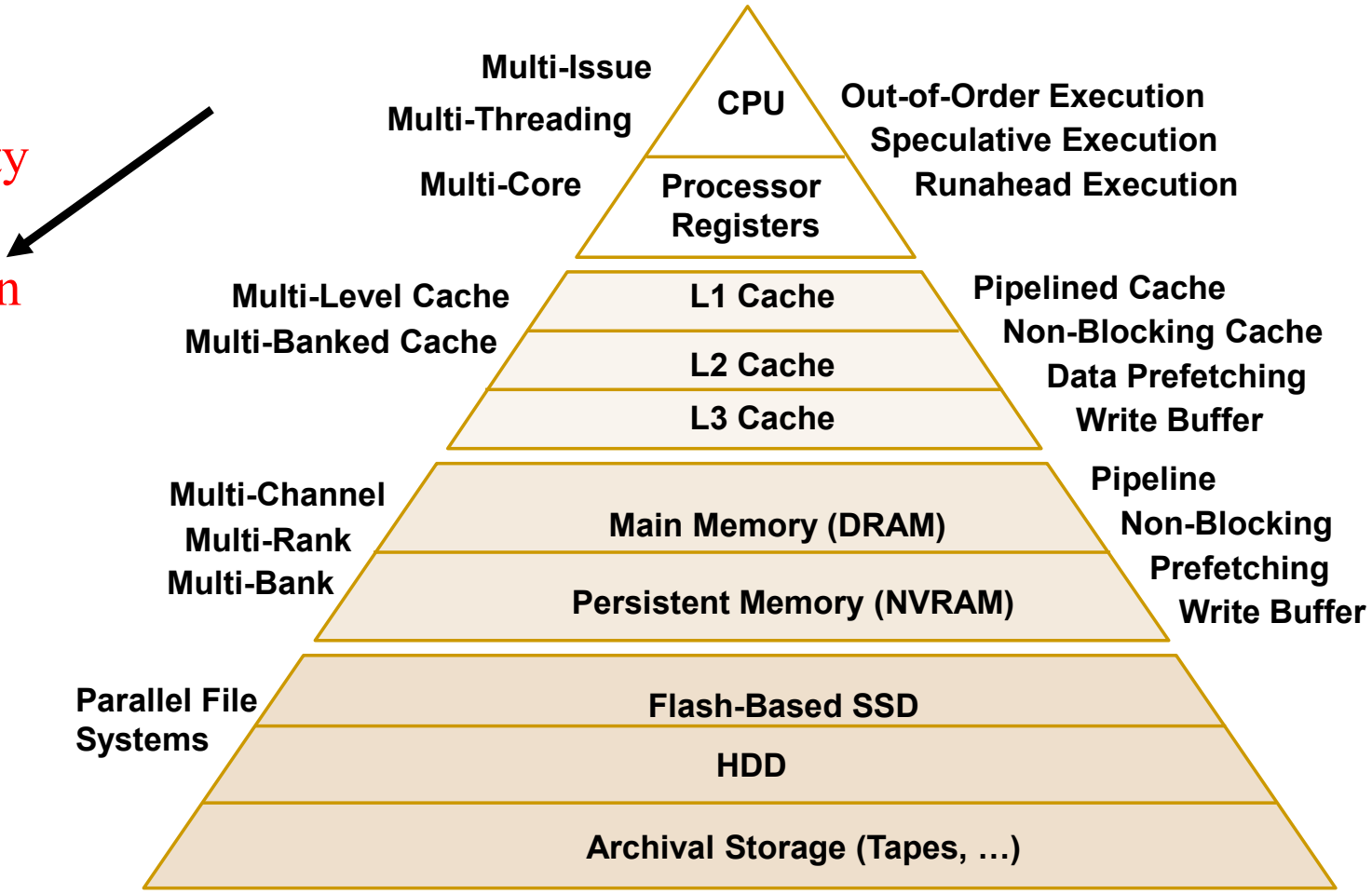




Data Locality

# Advanced Solution: Deep Hierarchy & Concurrency

## Assumptions

- Memory Hierarchy: Locality
- Concurrence: Data access pattern
  - Data stream



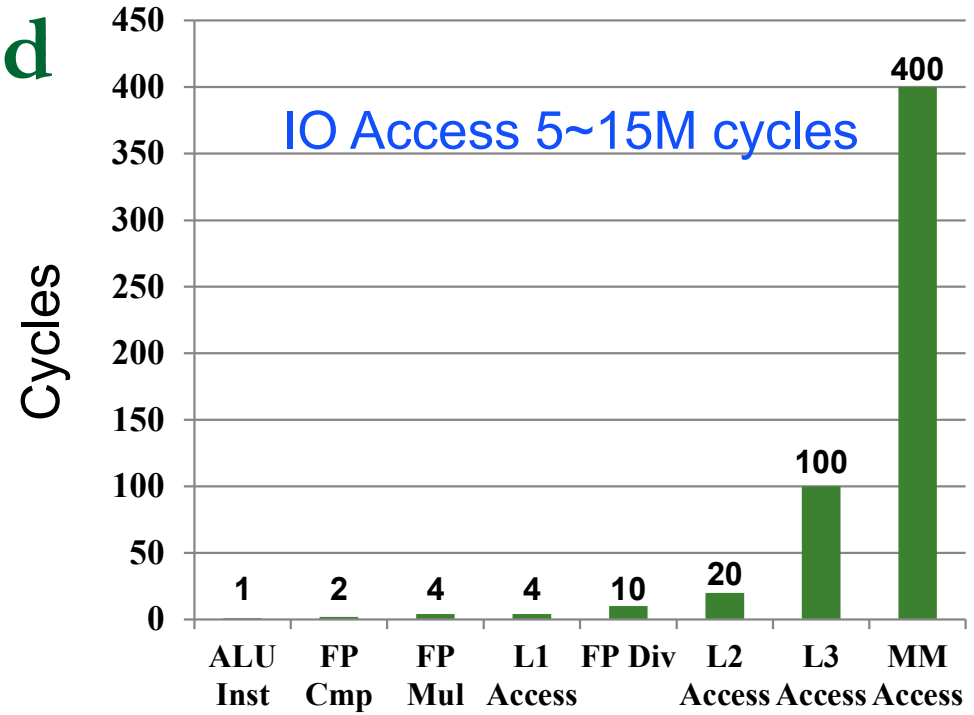

**Deep Memory-Storage Hierarchy with Concurrence**

# Assumption of Current Solutions

- Memory Hierarchy: Locality
- Concurrence: Data access pattern
  - Data stream

**Extremely Unbalanced Operation Latency**

**Performances vary largely**



# Advanced Solution: ASIC from CPU side

- GPU, DSP, AI Chip
  - GPU is a chip tailored to graphics processing, DSP is for signal processing, and AI chip is designed to do AI tasks
- Limited solution
  - Assume data are on the chip
- Limited application
  - *Computation Accelerator*
  - Please recall our memory-bound results for multicore




# New Solution: PIM chip

- PIM
  - Processing in memory (also called processor in memory) is the integration of a processor with RAM on a single chip.
  - NDP (Near-memory Data Processing)
  - ISP (In-Storage Processing)
- Computer power is weak
  - A full kitchen needs a refrigerator
- Limited application
  - *Data movement reducer*
  - A helper/mitigator

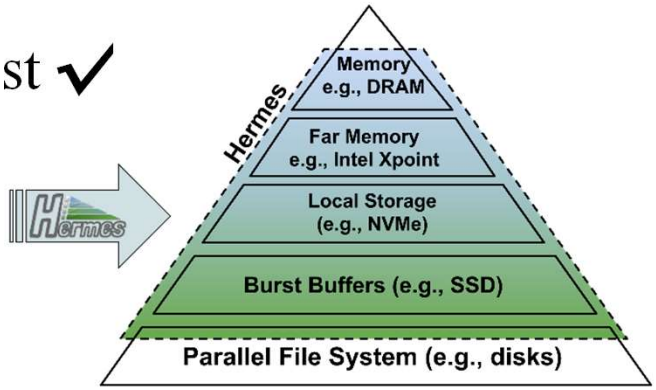
**How to use it?**

# $Dataflow_v$ Implementation

I/O level

- Storage is the last level of the memory hierarchy (DMSH)√
- Start at where the data is
- Advantage
  - Can be implemented and verified
- ***Challenges***
  - Data management ✓
  - Network impact √
  - Passing operation demands with data request ✓

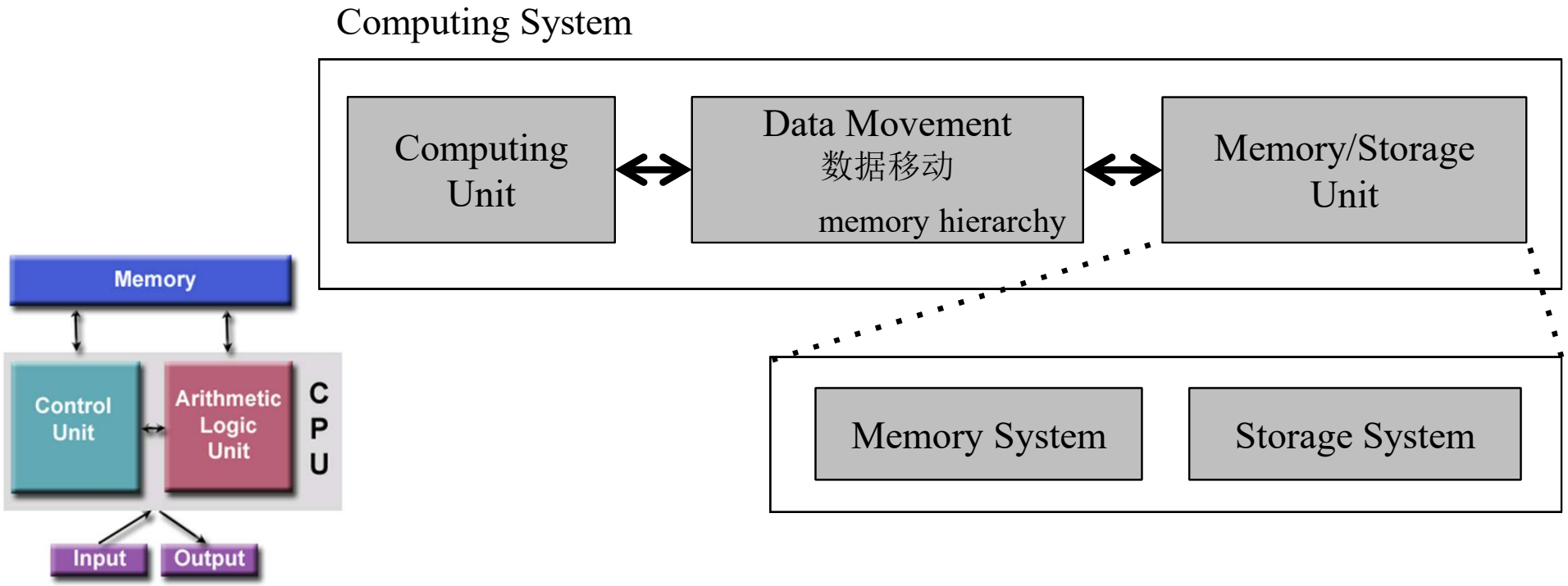
***Let us do it ?***




Anthony Kougkas, Hariharan Devarajan, and Xian-He Sun. "Hermes: a heterogeneous-aware multi-tiered distributed I/O buffering system," ACM, HPDC18, Tempe, Arizona, USA, June 2018

# Re-examine the von Neumann Arch.

- Can we make von Neumann more data centric or compute and data equal ?
- **Yes**: focus on data and data access delay
- **How**: *Advance current memory-wall solutions*



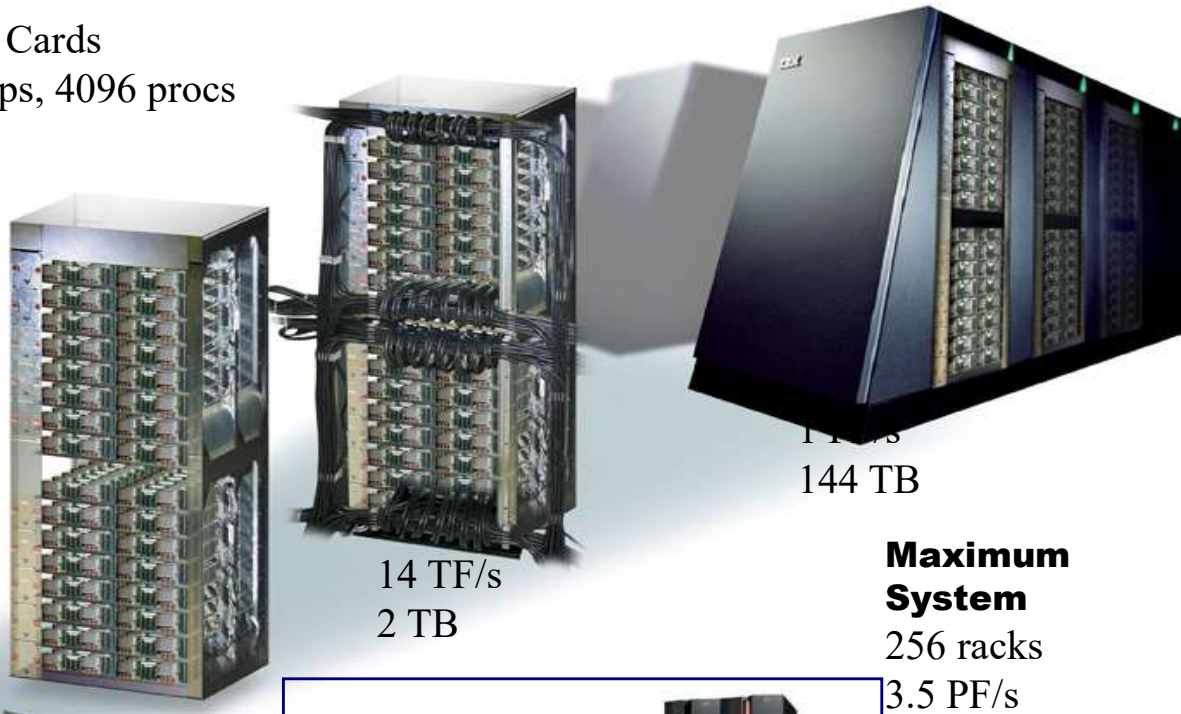
# Evolution of Computing:
## The biggest machine becomes even bigger

**Petaflops System**
72 Racks

**Rack** Cabled 8x8x16

IBM BG/P

32 Node Cards
1024 chips, 4096 procs



144 TB

Source: ANL ALCF

**Node Board**
(32 chips 4x4x2)
32 compute, 0-2 IO cards

14 TF/s
2 TB

**Maximum System**
256 racks
3.5 PF/s
512 TB

**Compute Card**
1 chip, 20 DRAMs

435 GF/s
64 GB

**Chip**
4 cores

13.6 GF/s
2.0 GB DDR
Supports 4-way SMP

850 MHz
8 MB EDRAM

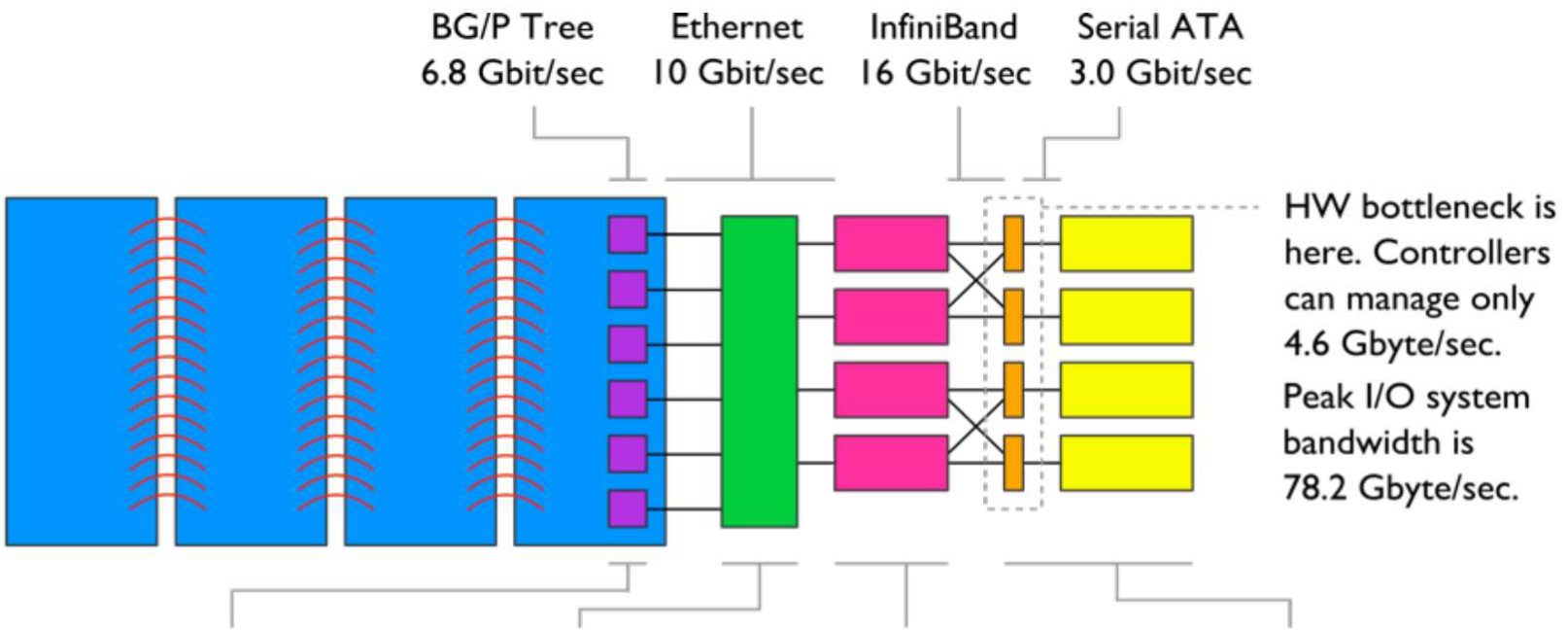
Front End Node / Service Node
System p Servers
Linux SLES10

HPC SW:
Compilers
GPFS
ESSL
Loadleveler

# Frontier: the World Fastest Computer


- 1.194 exaFLOPS (Rmax, 10^18) / 1.67982 exaFLOPS (Rpeak)
- 9,472 AMD Epyc 7453s "Trento" 64 core 2 GHz CPUs (606,208 cores)
- 37,888 Radeon Instinct MI250X GPUs (8,335,360 cores).
- 74 19-inch (48 cm) rack cabinets. Each cabinet hosts 64 blades, each consisting of 2 nodes.

# Intrepid Parallel Storage System




**Gateway nodes** run parallel file system client software and forward I/O operations from HPC clients.

*640 Quad core PowerPC 450 nodes with 2 Gbytes of RAM each*

**Commodity network** primarily carries storage traffic.

*900+ port 10 Gigabit Ethernet Myricom switch complex*

**Storage nodes** run parallel file system software and manage incoming FS traffic from gateway nodes.

*136 two dual core Opteron servers with 8 Gbytes of RAM each*

**Enterprise storage** controllers and large racks of disks are connected via InfiniBand or Fibre Channel.

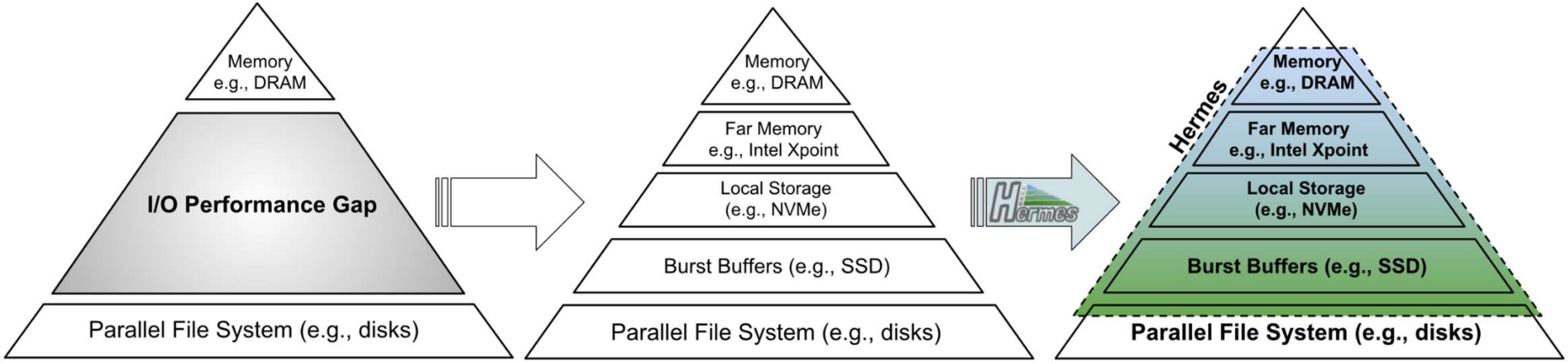
*17 DataDirect S2A9900 controller pairs with 480 1 Tbyte drives and 8 InfiniBand ports per pair*

Architectural diagram of the 557 TFlop IBM Blue Gene/P system at the Argonne Leadership Computing Facility.

# Current Project: Hermes

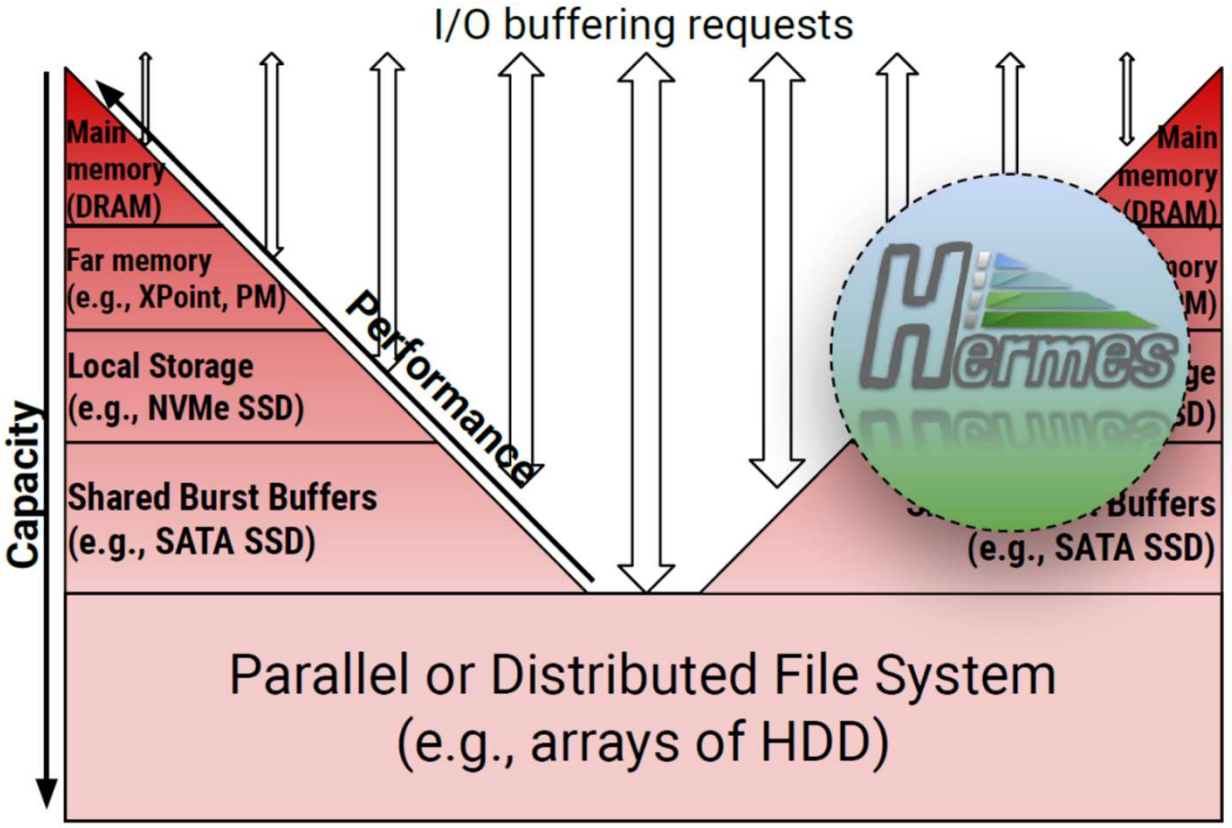


- A new, multi-tiered, distributed caching platform that:
  - Enables, manages, and supervises I/O operations in the Deep Memory and Storage Hierarchy (DMSH).
  - Offers selective and dynamic layered data placement/replacement
  - Is modular, extensible, and performance-oriented.
  - Supports a wide variety of applications (scientific, BigData, etc.,).



# Hermes: A Multi-tiered I/O Buffering System

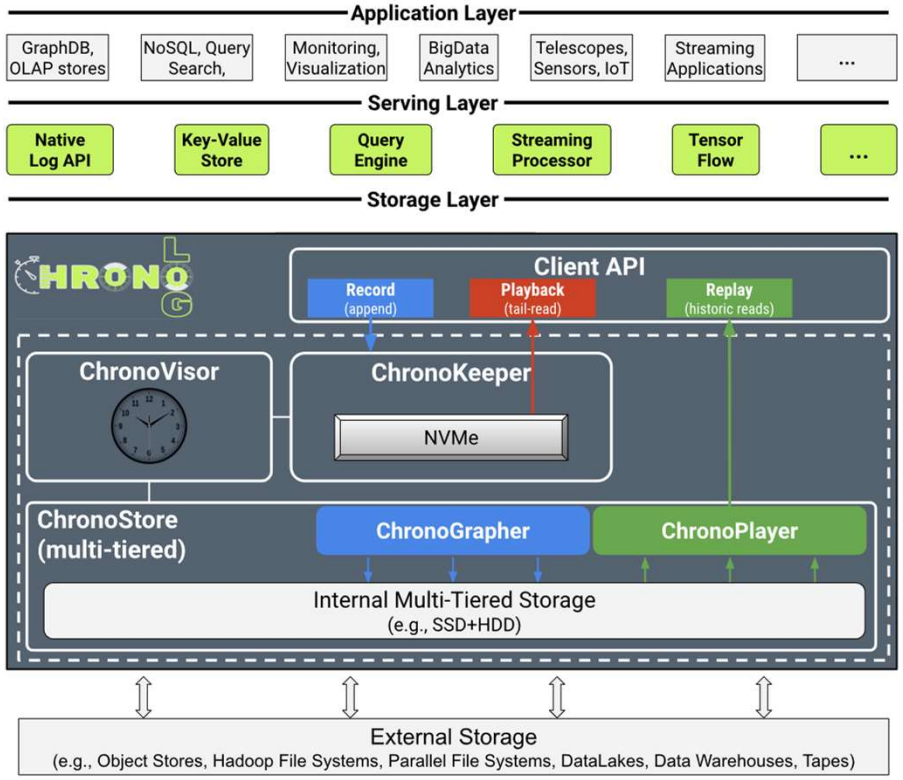
- Selective cache, concurrent, matching
- Independent management of each tier




A. Kougkas, H. Devarajan, and X.-H. Sun, "I/O Acceleration via Multi-Tiered Data Buffering and Prefetching," Journal of Computer Science and Technology, vol. 35, no. 1, pp. 92-120, Jan. 2020

# ChronoLog: A High-Performance Storage Infrastructure for Activity and Log Workloads

- Unprecedented huge activity (or log) data
  - Activity data describe things that happen rather than things that are
- Unparalleled importance of activity/log data
  - traditional database systems, non-traditional data management systems, decision making, information retrieval, data mining, deep learning, etc.
- *ChronoLog* is a distributed shared log storage ecosystem
  - Supports a wide variety of applications with different requirements under a single platform
  - Offers total ordering, high concurrency, and capacity scaling
- Challenges:
  - Imposing total ordering of distributed events
  - Scaling under a global log order
- Key techniques:
  - A log ordering based on a physical time (i.e., a globally accessible clock)
  - A dynamic tiered data management




Proposed ecosystem: including a core library & collection of plugins

**COEUS**: Accelerating Scientific Insights Using Enriched Metadata

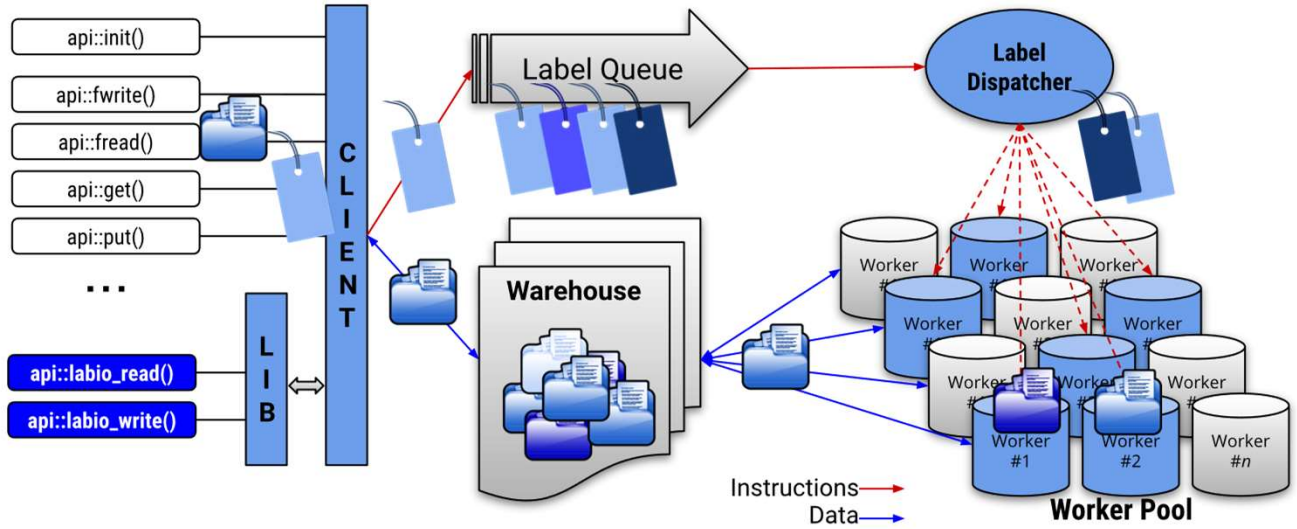
THE UNIVERSITY OF CHICAGO

Funded by the National Science Foundation under Award Number (FAIN): 2104013

ILLINOIS INSTITUTE OF TECHNOLOGY

# dLabel: Data Operation with Label

- Data requests are transformed into (data) ***Label*** units
  - A label is a tuple of an operation and a pointer to the data
- A dispatcher distributes labels to the workers
- Workers execute labels independently (i.e., fully decoupled)




A. Kougkas, H. Devarajan, J. Lofstead, X.-H. Sun; "*LABIOS: A Distributed Label-Based I/O System", i*n Proceedings of ACM HPDC '19 (Best Paper Award)

# Work Opportunities

- Research opportunities for graduate students:
  - Always look for self-motivated and hard-working grad students
  - Ph.D. students: CS597 and CS691
  - MS students: CS591 "Research and Thesis for MS Degree"
  - Take CS546 & CS550, check my research projects, send me your CV
- Research opportunities for undergrad students:
  - NSF REU (Research Experiences for Undergraduates) with Prof. Xian-He Sun
    - Various project topics, including development of scheduling simulator, analysis of system logs, ….
    - If interested, contact Prof. Sun (sun@iit.edu)
- Research opportunities for both graduate & undergrad:
  - Programmer

# Any Questions?