SCALABLE COMPUTING
SOFTWARE LABORATORY

ILLINOIS INSTITUTE
OF TECHNOLOGY

Hermes

High Performance Data Access: the DMSH approach

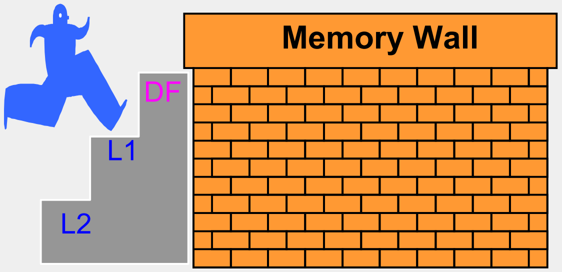
Xian-He Sun
sun@iit.edu

KEYNOTE: HPC-CHINA 2019

Hot Issues

- AI and Deep Learning
- Big Data
- High Performance Computing and Cloud Computing

The Issue is Data Processing

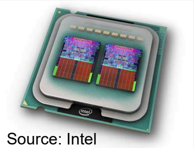


Hermes

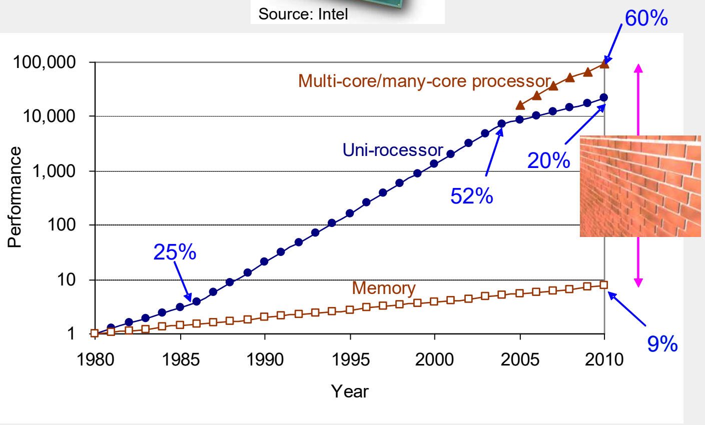
Background and Motivation

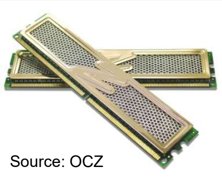
Problem: The Memory-wall Problem

- Processor performance increases rapidly
  - Uni-processor: ~52% until 2004
  - Aggregate multi-core/many-core processor performance even higher since 2004
- Memory: ~9% per year
- I/O: ~6% per year
- Processor-memory speed gap keeps increasing



Source: Intel





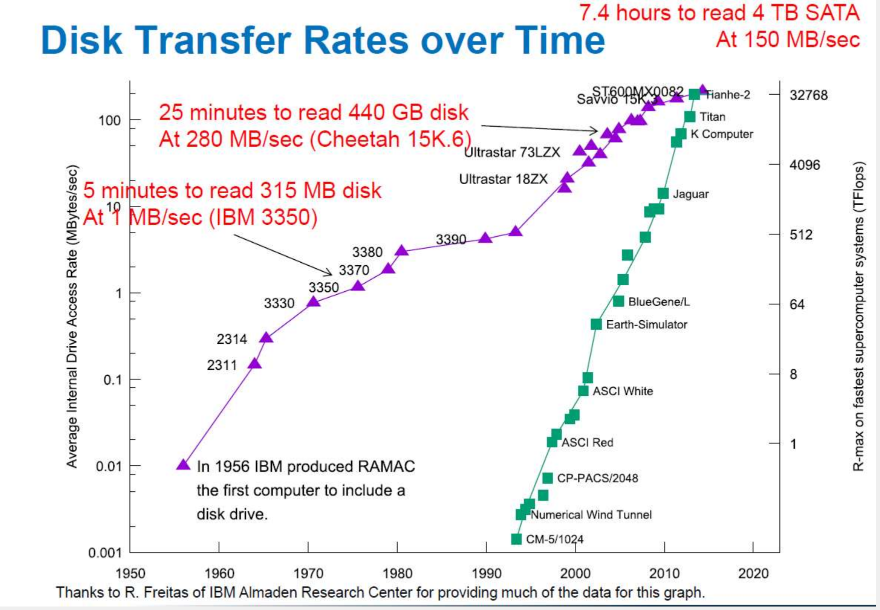
Source: OCZ

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

I/O becomes the Bottleneck of HPC

Project	On-line Data (TB)	Off-line Data (TB)
Combustion in Reactive Gases	100	1000
Seismic Hazard Analysis	204	125
Climate Science	60	200
Nuclear Structure and Reactions	52	27
Reactor Thermal Hydraulic Modeling	100	200
Quantum Chromodynamics	2000	1000
Plasma Physics	1333	200
Turbulent Combustion	600	1000
Physical Chemistry	512	1000

[1] R. Latham, R. Ross, B. Welch, and K. Antypas, "Parallel I/O in Practice," Tutorial of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2015



Disk Transfer Rates over Time

7.4 hours to read 4 TB SATA
At 150 MB/sec

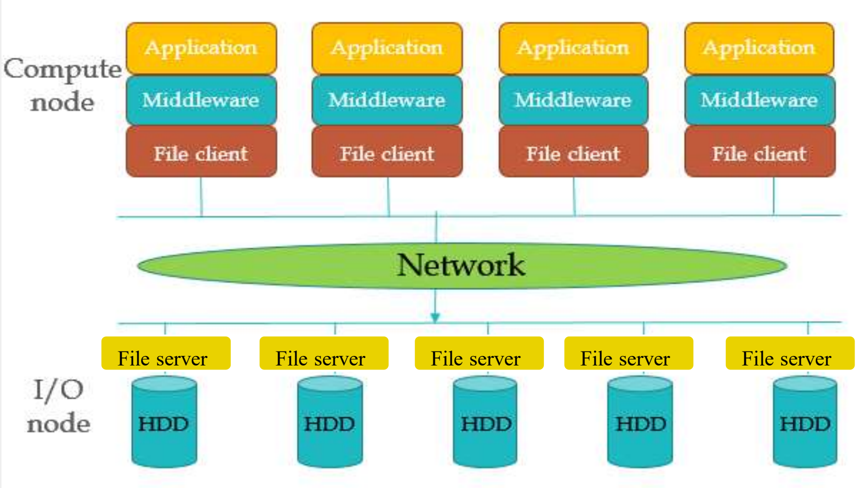
25 minutes to read 440 GB disk
At 280 MB/sec (Cheetah 15K.6)

5 minutes to read 315 MB disk
At 1 MB/sec (IBM 3350)

In 1956 IBM produced RAMAC the first computer to include a disk drive.

Thanks to R. Freitas of IBM Almaden Research Center for providing much of the data for this graph.

Solution: Parallel I/O Systems

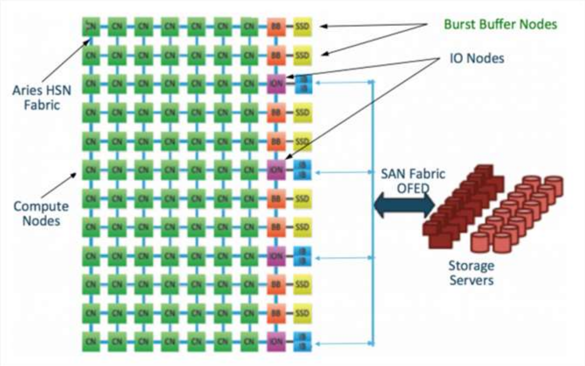


Architecture of a typical parallel I/O system

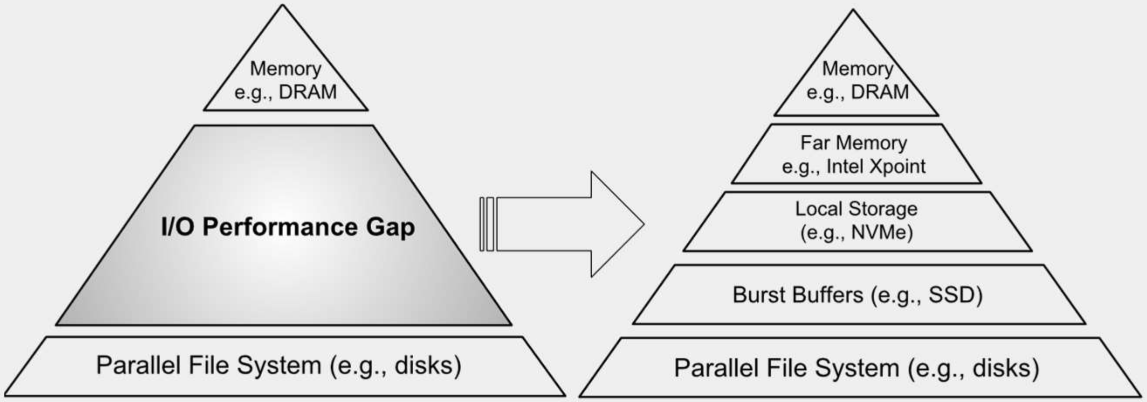
Not Good Enough

New: Deep Memory and Storage Hierarchy(DMSH)

- HPC storage systems with burst buffers (BB) have been installed at several HPC sites.
- Several new non-volatile technologies and remote data access mechanisms are developed in recent years
- A multi-(performance)levels memory and storage in a hierarchy is built, called **DMSH**
- **Inclusion is undefined (multi-tiered)**

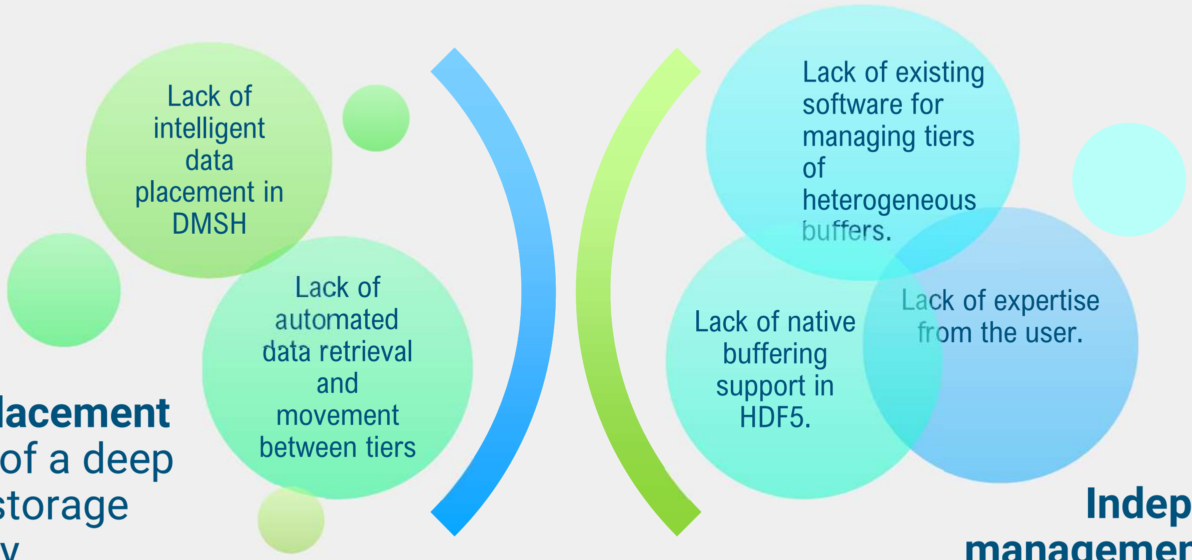


Cori, a Cray XC40 system at NERSC uses Cray's DataWarp BB technology


Memory e.g., DRAM
I/O Performance Gap
Parallel File System (e.g., disks)
Memory e.g., DRAM
Far Memory e.g., Intel Xpoint
Local Storage (e.g., NVMe)
Burst Buffers (e.g., SSD)
Parallel File System (e.g., disks)

Ideally, the presence of multiple layers of storage should be **transparent** to applications without having to sacrifice **I/O performance**.

DMSH Challenges



Lack of intelligent data placement in DMSH

Lack of automated data retrieval and movement between tiers

Lack of existing software for managing tiers of heterogeneous buffers.

Lack of native buffering support in HDF5.

Lack of expertise from the user.

Complex data placement among the tiers of a deep memory and storage hierarchy

Independent management of each tier of the DMSH

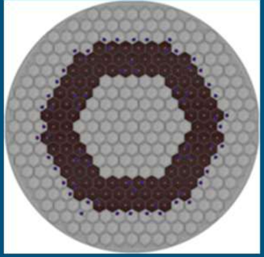
Exist: Hierarchical Data Format (HDF)

The HDF Group

- Users use Hierarchical Data Format 5 (HDF5) to provide data information
- HDF5 library and core technologies address the problems of
  - organizing, storing, discovering, accessing, analyzing, sharing, and preserving data in the face of enormous growth in size and complexity.
- HDF offers a flexible format and powerful API backed by over 25 years of development history.
- HDF stores data in binary files organized for high-performance access, using a machine-independent, **self-describing format**.
- Considers memory-to-disk I/O endpoints.

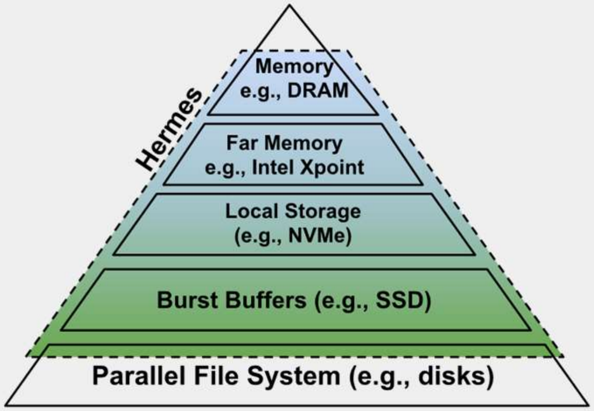
Big Opportunity

- **A combination of DMSH I/O Buffering and the HDF5 data format is a solution that can efficiently support scientific discovery**
- Hermes: a software system which seamlessly and transparently supports accesses to the tiers of the DMSH to boost applications' I/O performance



Objective: Application-Specific Optimization

- Utilizing diverse hardware devices
- Utilizing deep, non-inclusive memory hierarchy (tiers)
- Utilizing application-specific information
- Utilizing what memory & file systems can provide
- Developing a I/O system to materialize these utilizations automatically



Hermes

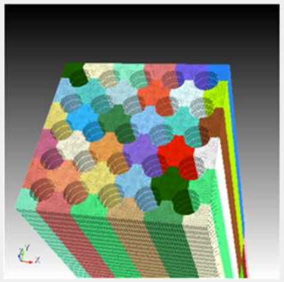
Memory e.g., DRAM

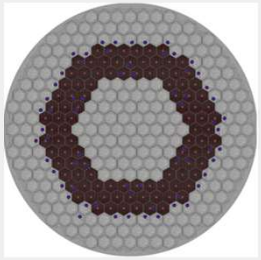
Far Memory e.g., Intel Xpoint

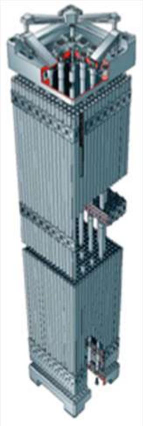
Local Storage (e.g., NVMe)

Burst Buffers (e.g., SSD)

Parallel File System (e.g., disks)







Ideally, the presence of multiple layers of storage should be **handle** the **model** and **scale complexity** of applications without sacrificing **I/O performance**.

What do We Have (in addition to the Hermes design)?

1 Theoretic Foundation: The Concurrent-AMAT memory model and Pace-Matching Data Transfer Mechanism

2 Method Development: Smart, selective I/O cache and I/O optimization in heterogenous environments

3 Software Practice: The optimization and integration of HPC and Cloud/Big Data file systems

4 Industrial and DoE Support: The collaboration with the HDF group and the LBNL

SCALABLE COMPUTING
SOFTWARE LABORATORY

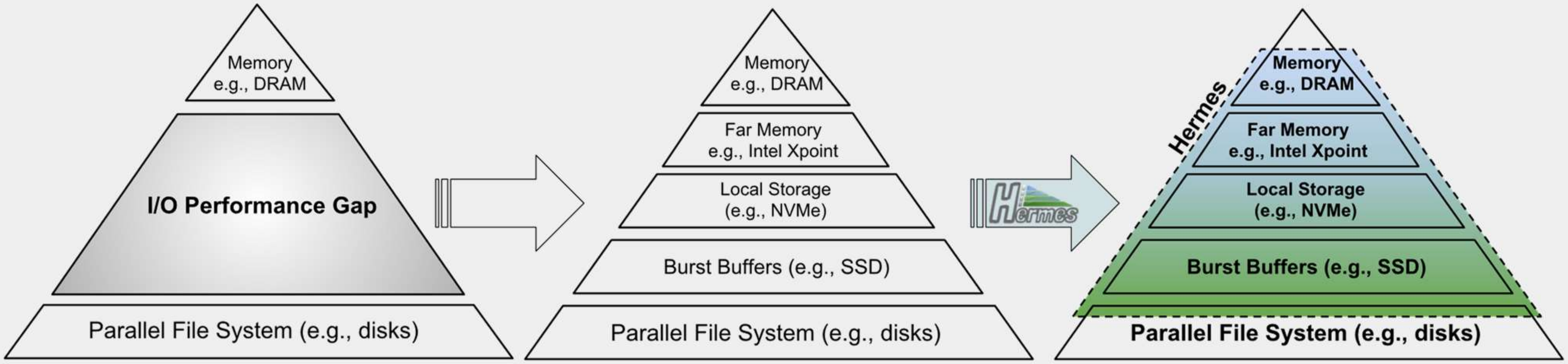
ILLINOIS INSTITUTE
OF TECHNOLOGY

Hermes

Overview

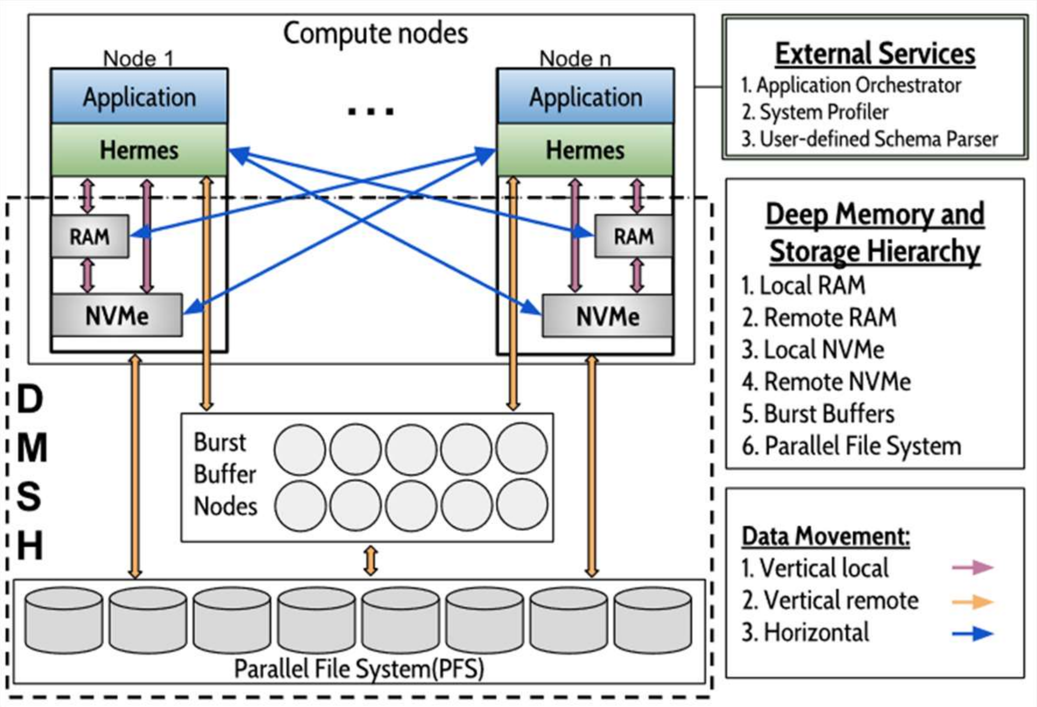
Hermes Overview

- 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.,).


Memory e.g., DRAM
I/O Performance Gap
Parallel File System (e.g., disks)
Memory e.g., DRAM
Far Memory e.g., Intel Xpoint
Local Storage (e.g., NVMe)
Burst Buffers (e.g., SSD)
Parallel File System (e.g., disks)
Hermes
Hermes
Memory e.g., DRAM
Far Memory e.g., Intel Xpoint
Local Storage (e.g., NVMe)
Burst Buffers (e.g., SSD)
Parallel File System (e.g., disks)

Software Architecture

- Hermes machine model:
  - Large amount of RAM
  - Local NVMe and/or SSD device
  - Shared Buffer at each layer
  - Remote disk-based PFS
- Hierarchy is based on media:
  - Access Latency
  - Data Throughput
  - Capacity
- Two data paths:
  - Vertical (within node)
  - Horizontal (across nodes)



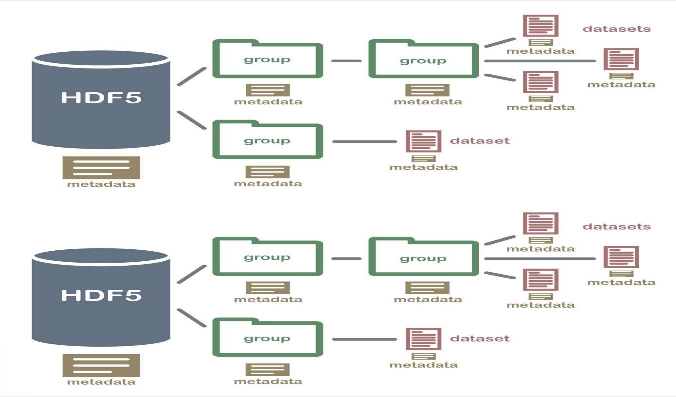
External Services
1. Application Orchestrator
2. System Profiler
3. User-defined Schema Parser

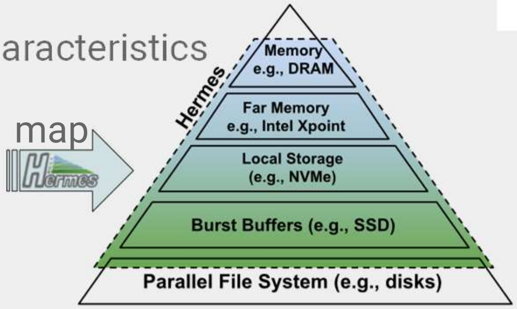
Deep Memory and Storage Hierarchy
1. Local RAM
2. Remote RAM
3. Local NVMe
4. Remote NVMe
5. Burst Buffers
6. Parallel File System

Data Movement:
1. Vertical local →
2. Vertical remote →
3. Horizontal →

Hermes + HDF5

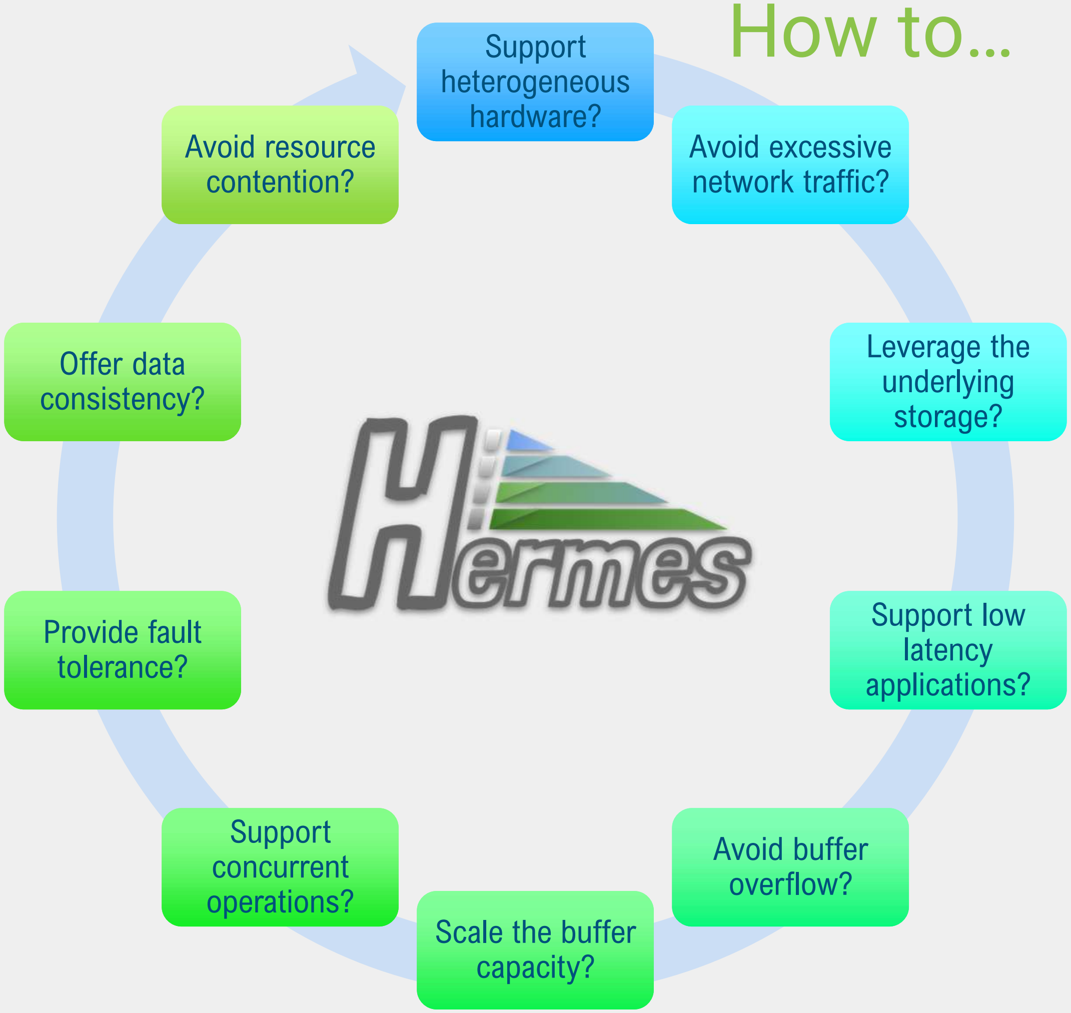
- HDF5 is a self-describing hierarchical data format which makes it ideal for Hermes
  - Utilize the rich metadata offered by HDF5 to efficiently place data in the hierarchy.
  - Leverage HDF5 characteristics
    - files,
    - groups,
    - datasets,
    - chunked I/O





Hermes attacks several technical challenges in multi-tiered storage systems

How to…



Support heterogeneous hardware?

Avoid excessive network traffic?

Leverage the underlying storage?

Support low latency applications?

Avoid buffer overflow?

Scale the buffer capacity?

Support concurrent operations?

Provide fault tolerance?

Offer data consistency?

Avoid resource contention?

How to use Hermes?

- Data Buffering
  - Persistent or not, temporary storage
- Data Caching for Buffering
  - Prefetching, exclusive cache
- Data Streaming
- Integration of various storage pools
  - File systems, Object stores, DBMS, HDFS
- ML and AI workloads (focused application)

SCALABLE COMPUTING SOFTWARE LABORATORY

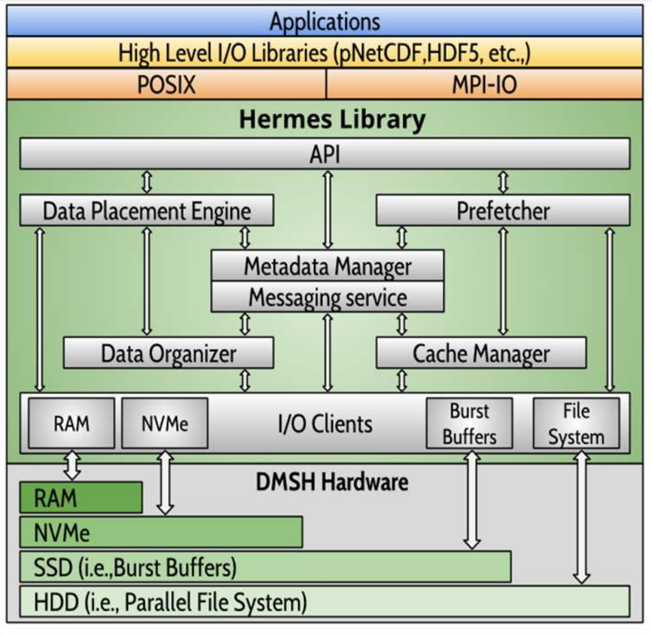
ILLINOIS INSTITUTE OF TECHNOLOGY

Hermes

Technical Details

Design Details

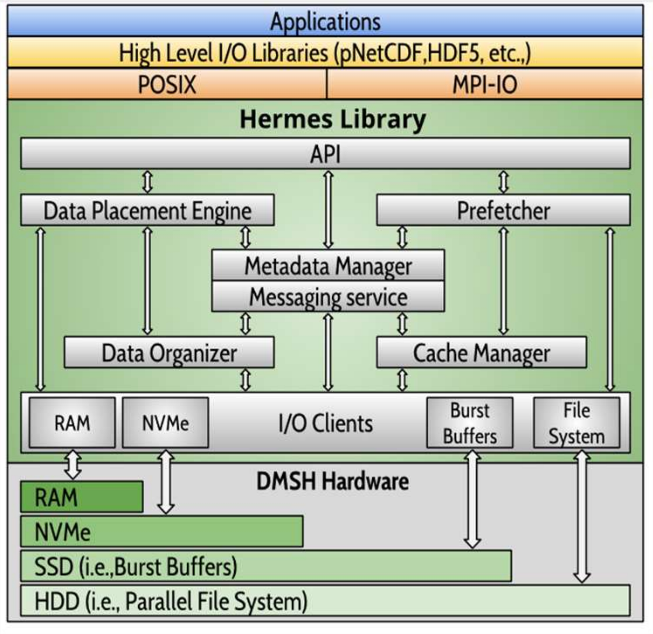
- Hermes API
  - intercept all I/O calls from applications
  - calculates the operations to be carried out in case of an active buffering scenario.
- Hermes Data Placement Engine (DPE)
  - calculates the data destination, i.e., where in the hierarchy should the data be placed.
  - uses various data placement policies.
- Hermes Data Organizer
  - event-based component
  - carries out all data movements
    - E.g., for prefetching reasons, evictions, lack of space, or hotness of data etc.



Kougkas, Anthony, Hariharan Devarajan, and Xian-He Sun. "Hermes: a heterogeneous-aware multi-tiered distributed I/O buffering system." In Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing (HPDC2018), pp. 219-230, ACM, 2018.

Design Details

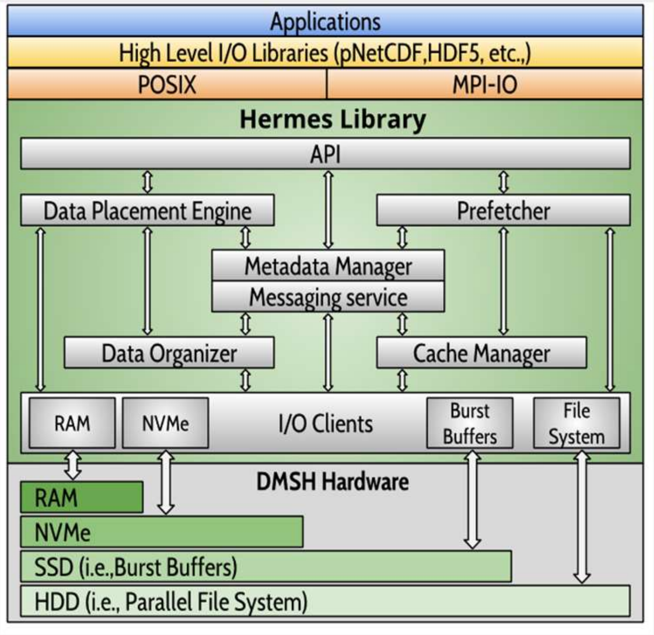
- Metadata Manager
  - maintains two types of metadata:
    - user’s metadata operations (e.g., files, directories, permissions etc.),
    - Hermes library's internal metadata (e.g., locations of all buffered data and internal temporary files that contain user files).
- Cache Manager
  - handles all buffers inside Hermes
  - equipped with several data replacement policies (e.g., least recently used (LRU) and least frequently used (LFU)).



Kougkas, Anthony, Hariharan Devarajan, and Xian-He Sun. "IRIS: I/O Redirection via Integrated Storage." In Proceedings of the 2018 International Conference on Supercomputing (ICS2018), pp. 33-42. ACM, 2018.

Design Details

- Messaging Service
  - enables horizontal buffering
  - provides an infrastructure to pass instructions to other nodes to perform operations on data or facilitate its movement
- Prefetcher
  - implements several typical prefetching algorithms
    - sequential data access,
    - strided access,
    - random access,
    - user defined prefetching



H. Devarajan, A. Kougkas, X.-H. Sun. "An Intelligent, Adaptive, and Flexible Data Compression Framework," IEEE/ACM International Symposium in Cluster, Cloud, and Grid Computing (CCGrid'19), Larnaca, Cyprus, May, 2019.

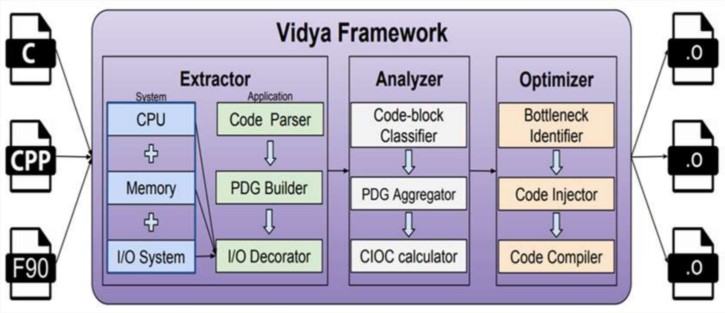
External Services

- Application Orchestrator
  - offers support in a multiple-application environment
  - manages access to the shared layers of the hierarchy
  - minimizes interference between different applications sharing a layer
  - coordinates the flushing of the buffers to achieve maximum I/O performance

Anthony Kougkas, Hariharan Devarajan, Xian-He Sun, and Jay Lofstead. "Harmonia: An Interference-Aware Dynamic I/O Scheduler ", in Proceedings of the IEEE International Conference on Cluster Computing 2018 (Cluster'18), Sept. 2018

External Services

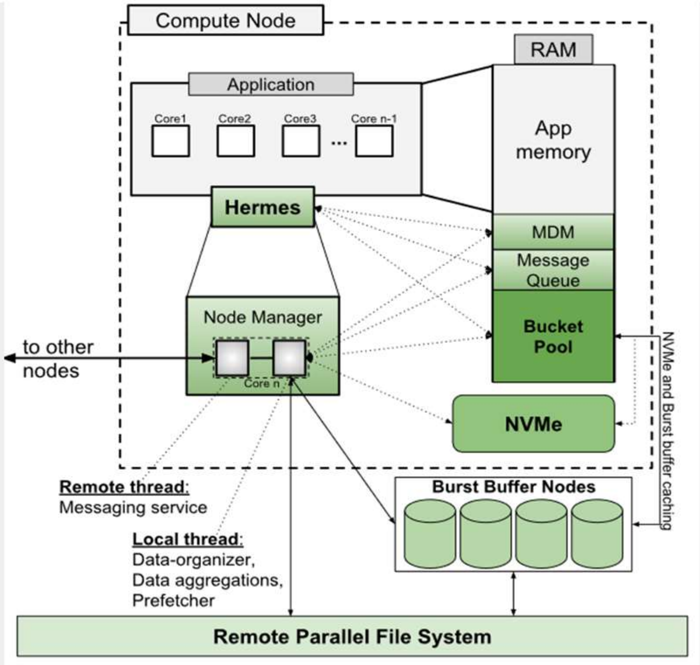
- System Profiler
  - runs the profiler once during the application initialization
  - performs a profiling of the underlying system in terms of hardware resources
  - detects the availability of DMSH and measures each layer's respective performance
  - profiles the applications and identifies incoming I/O phases
  - works together with the application coordinator (Harmonia) to detect access conflicts



Hariharan Devarajan, Anthony Kougkas, P. Challa, Xian-He Sun
*“Vidya: Performing Code-Block I/O Characterization for Data Access Optimization”, i*n Proceedings of the IEEE International Conference on High Performance Computing, Data, and Analytics 2018 (HiPC'18), Dec. 2018

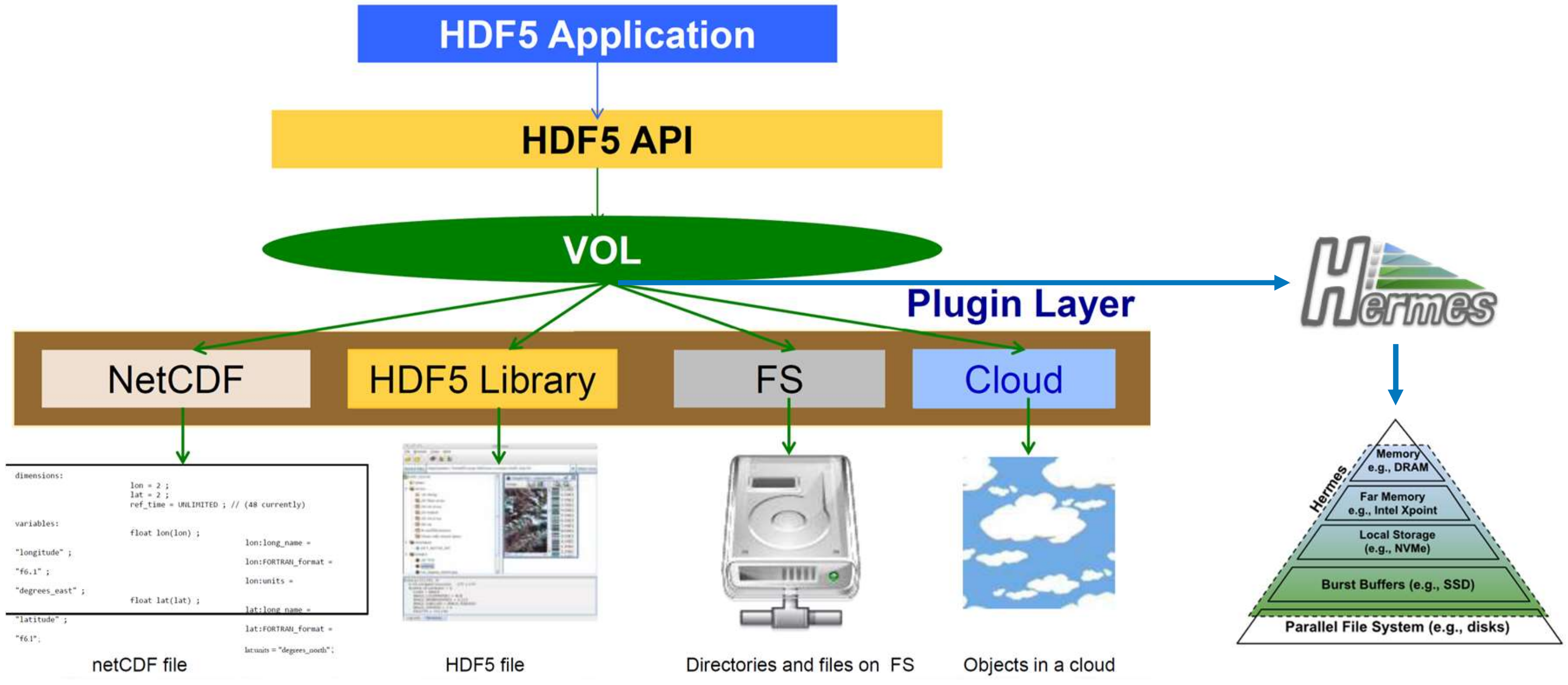
Deployment

- Dedicated core for Hermes
- Node Manager
  - Dedicated multithreaded core per node
  - MDM (MetaData Manager)
  - Data Organizer
  - Messaging Service
  - Memory management
  - Prefetcher
  - Cache manager
- RDMA-capable communication
- Can also be deployed in I/O Forwarding Layer (I/O FL)



A. Kougkas, H. Devarajan, J. Lofstead, X.-H. Sun. "LABIOS: A Distributed Label-Based I/O System," The 28th International Symposium on
B. High-Performance Parallel and Distributed Computing(HPDC'19), Phoenix, USA, June, 2019 (Best Paper award).

Hermes VOL plugin for HDF5 coming…



```
dimensions:
                        lon = 2 ;
                        lat = 2 ;
                        ref_time = UNLIMITED ; // (48 currently)

variables:
                        float lon(lon) ;
                                                lon:long_name =
"longitude" ;
                                                lon:FORTRAN_format =
"f6.1" ;
                                                lon:units =
"degrees_east" ;
                        float lat(lat) ;
                                                lat:long_name =
"latitude" ;
                                                lat:FORTRAN_format =
"f6.1";
                                                lat:units = "degrees_north";
```

netCDF file

HDF5 file

Directories and files on FS

Objects in a cloud

Buffering Modes

1

Persistent

- Synchronous
  - write-through cache,
  - stage-in
- Asynchronous
  - write-back cache,
  - stage-out

2

Non-Persistent

- Temporary scratch space
- Intermediate results
- In-situ analysis and visualization

3

Bypass

- Write-around cache

Hermes Data Placement Policies

1

Maximum Application Bandwidth (MaxBW): this policy aims to maximize the bandwidth applications experience when accessing Hermes.

2

Maximum Data Locality: this policy aims to maximize buffer utilization by simultaneously directing I/O to the entire DMSH.

3

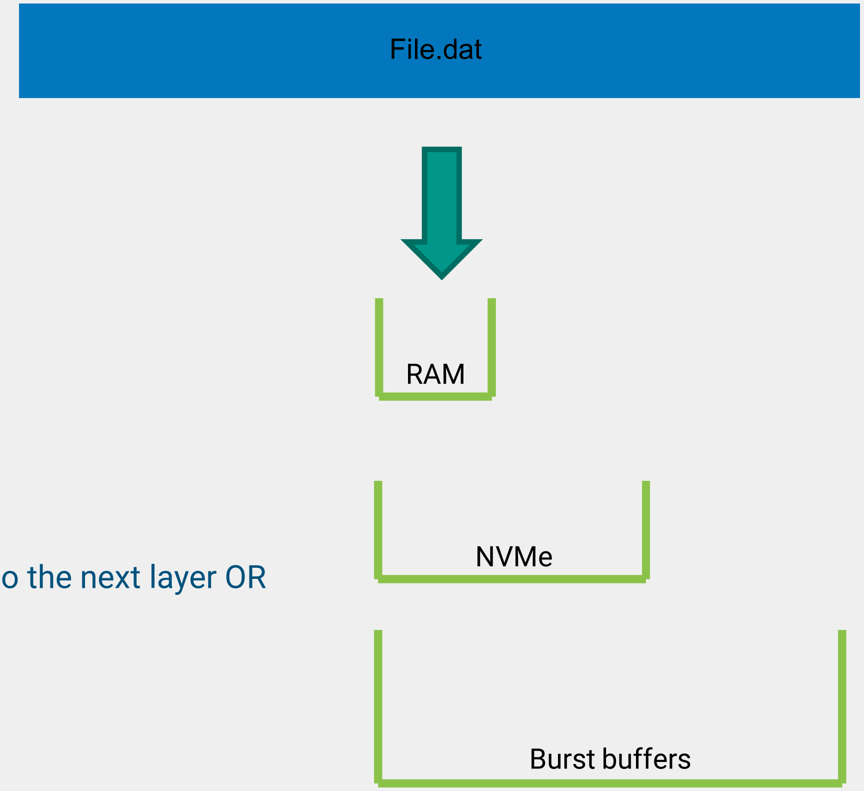
Hot-data: this policy aims to offer applications a fast cache for frequently accessed data (i.e., hot-data).

4

User-defined: this policy aims to support user-defined buffering schemas. Users are expected to submit an XML file with their preferred buffering requirements.

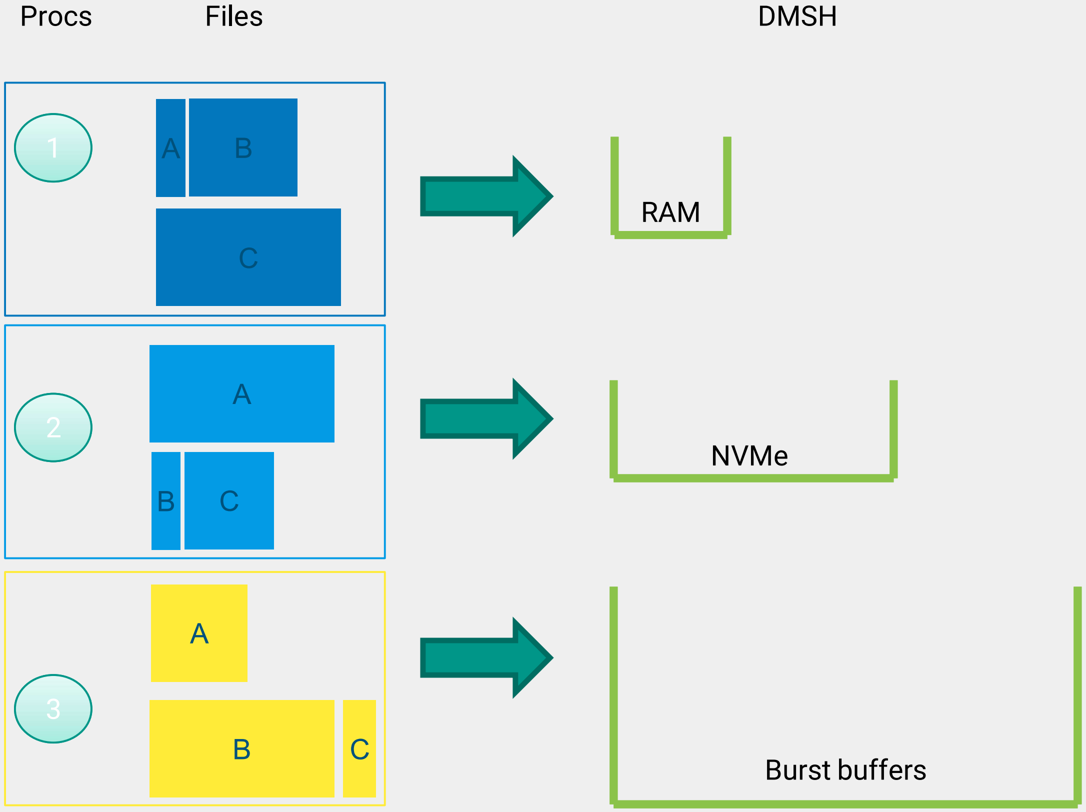
1 Maximum Bandwidth

- Start from the top layer
  - If free space > request size
    - place data here
  - If not, choose the best between
    1. Place as much data as possible here and the rest to the next layer OR
    2. Skip this layer and place data to the next one OR
    3. First flush top layer and then place data
- Recursive process



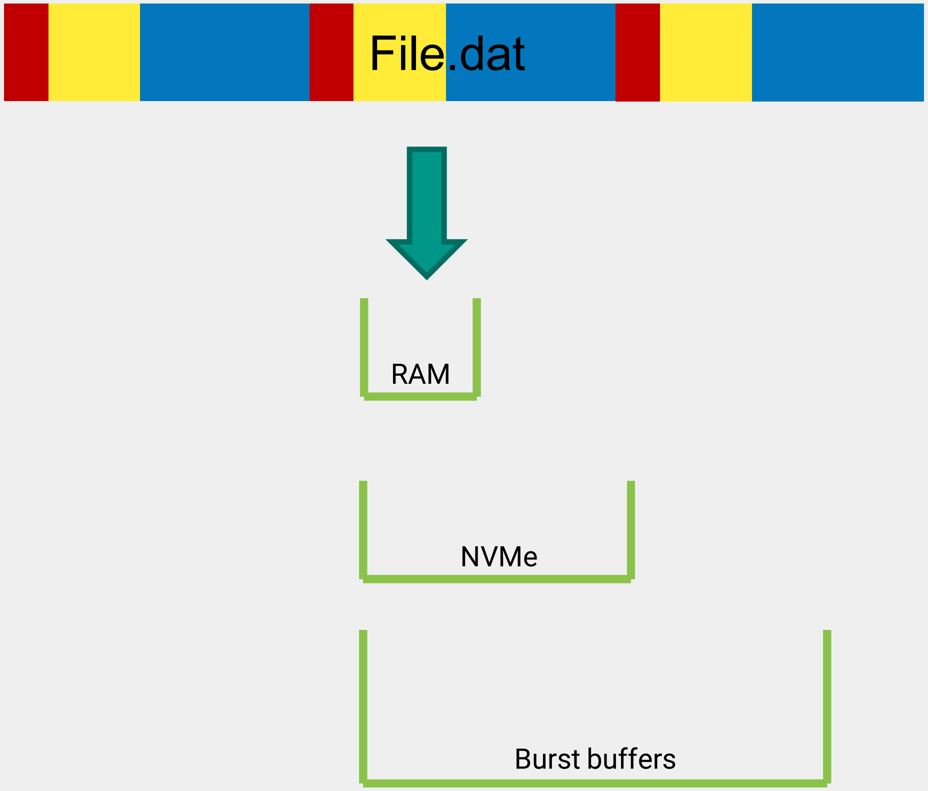
2 Data Locality

- Data dispersion unit:
  - POSIX files
  - HDF5 datasets
  - Etc.
- Place data based on:
  - Location of previously buffered data
  - Ratio between layers



3 Hot data

- Place data based on:
  - Spectrum of hot – cold data
- Higher layers hold hotter data



SCALABLE COMPUTING
SOFTWARE LABORATORY

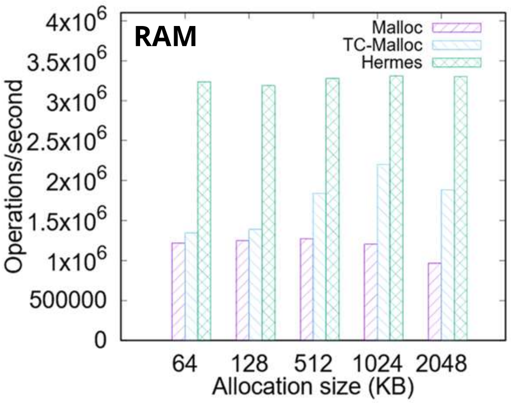
ILLINOIS INSTITUTE
OF TECHNOLOGY

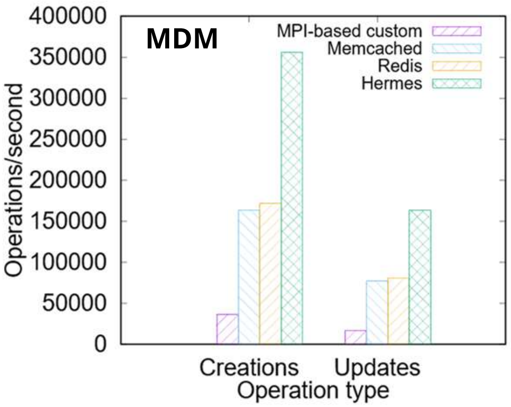
Hermes

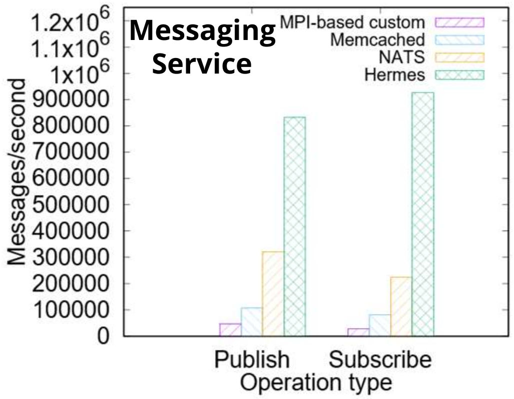
Some Initial Results

Hermes Components

- MPI shared dynamic memory window exposed in all nodes
- MPI_Put(), MPI_Get()
  - If RDMA is present, MPI uses it
- No need for dedicated server
- Indexing of windows for fast querying
- Complex data structures
- Update operations use MPI_EXCLUSIVE which ensure FIFO consistency
- Entire window with its index is mmap’ed for fault tolerance



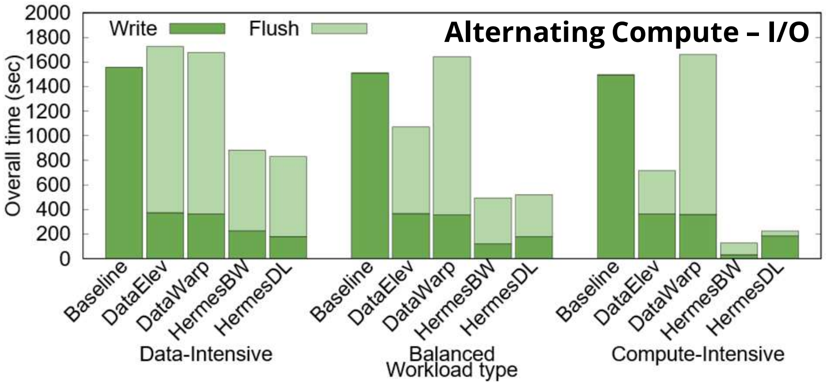




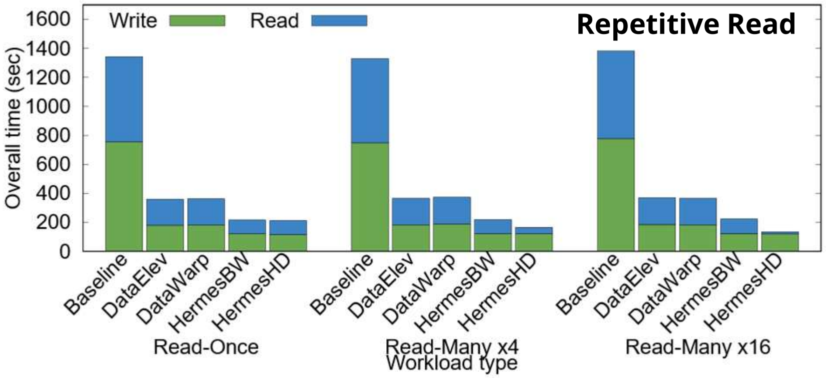
- 1 million fwrite() of various size and measured memory ops/sec
- 1 million metadata operations and measure MDM throughput ops/sec
- 1 million queue operations and measure messaging rate msg/sec

Benchmarks

- File-per-process
- 1024 ranks each 64MB
- 16 phases resulting 1TB total I/O
- Alternating Compute – I/O :
  - Data need to persisted
  - Workloads:
    - Data-intensive
    - Balanced
    - Compute-intensive
  - Metric:
    - Overall I/O time (write + flush)
- Repetitive Read:
  - Temporary data
  - Workloads:
    - Read-once: 32MB read 1x time
    - Readx4: 8MB read 4x times
    - Readx16: 2MB read 16x times
  - Metric:
    - Overall I/O time (write + read)



Hermes offers **8x and 2x** higher write performance on average when compared to No Buffering and state-of-the-art buffering platforms



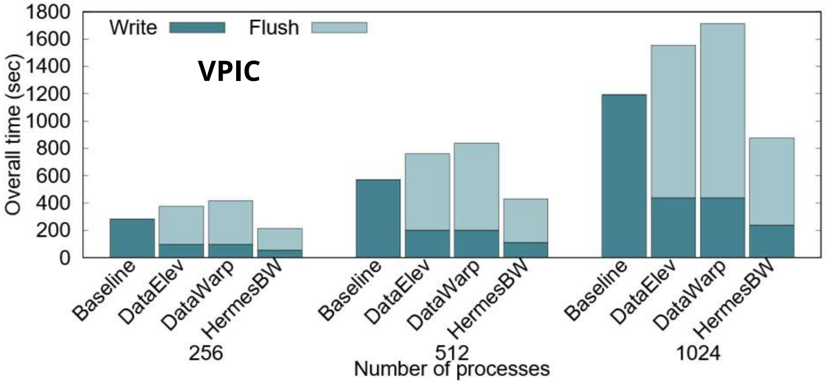
Hermes offers **38x and 11x** higher read performance for repetitive patterns when compared to No Buffering and state-of-the-art buffering platforms

- Hermes hides flushing behind compute (similar to Data Elevator)
- Hermes also hides data movement between layers behind compute
- Hermes leverages the extra layers of the DMSH to offer higher BW

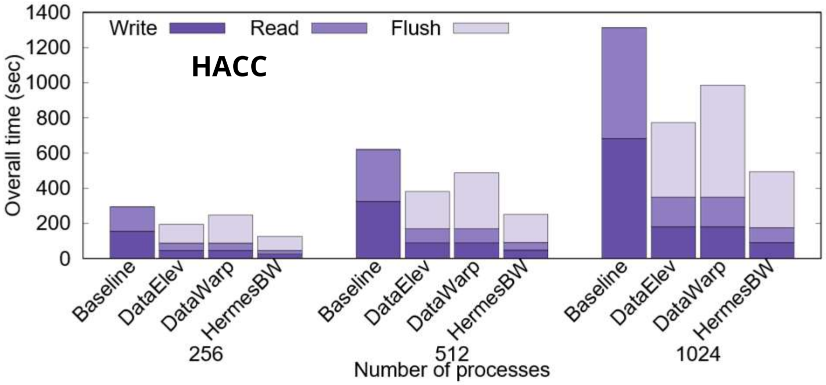
Anthony Kougkas, Hariharan Devarajan, and Xian-He Sun. *Hermes: A Heterogeneous-Aware Multi-Tiered Distributed I/O Buffering System,* In Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing, pp. 219-230. ACM, 2018.

Scientific Applications

- Strong scaled up to 1024 ranks
- 16-time steps
- Metric:
  - Total I/O time (write + read + flush)
- Vector Particle-In-Cell (VPIC):
  - Uses HDF5 files
- Hardware Accelerated Cosmology Code (HACC):
  - MPI - I/O Independent



Hermes offers **5x and 2x** higher write performance on average when compared to
No Buffering and state-of-the-art buffering platforms



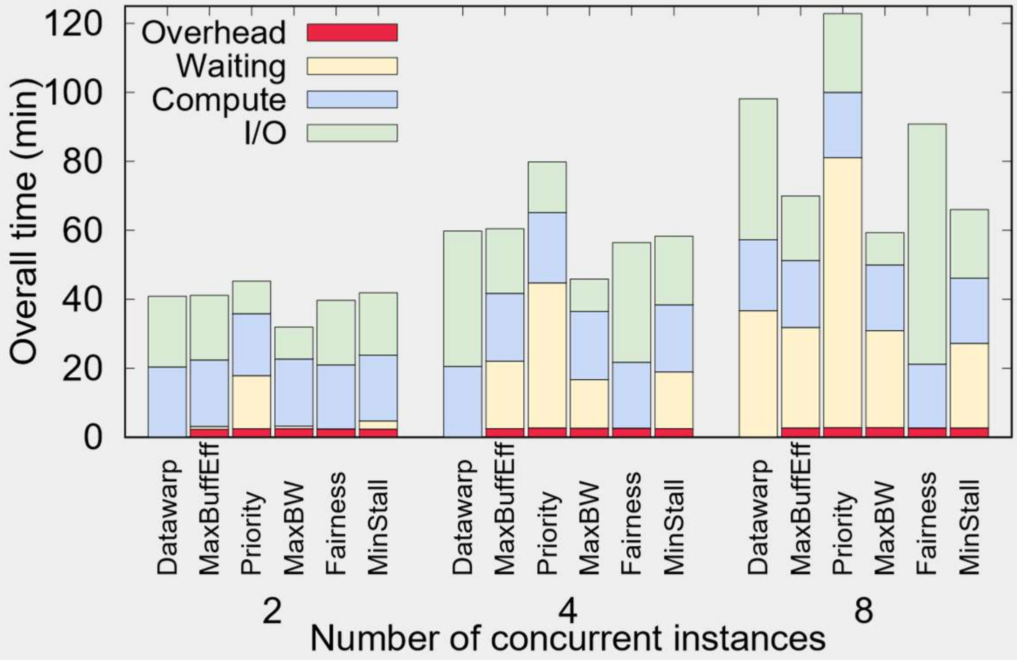
Hermes offers **7.5x and 2x** higher read performance for repetitive patterns when compared to
No Buffering and state-of-the-art buffering platforms

- Hermes hides data movement between tiers behind compute
- Hermes leverages the extra layers of the DMSH to offer higher BW
- Hermes utilizes a concurrent flushing overlapped with compute

Anthony Kougkas, Hariharan Devarajan, and Xian-He Sun. *Hermes: A Heterogeneous-Aware Multi-Tiered Distributed I/O Buffering System,* In Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing, pp. 219-230. ACM, 2018.

Application Orchestrator

- Synthetic benchmark
  - Balanced workload (compute-I/O)
- Average completion time
  - Waiting time
  - Computation time
  - I/O time
  - Overheads
- Concurrent execution scaling
  - 2-8 instances
  - Buffer can hold data up to 4 instances before they flush
- Compared to DataWarp scheduling



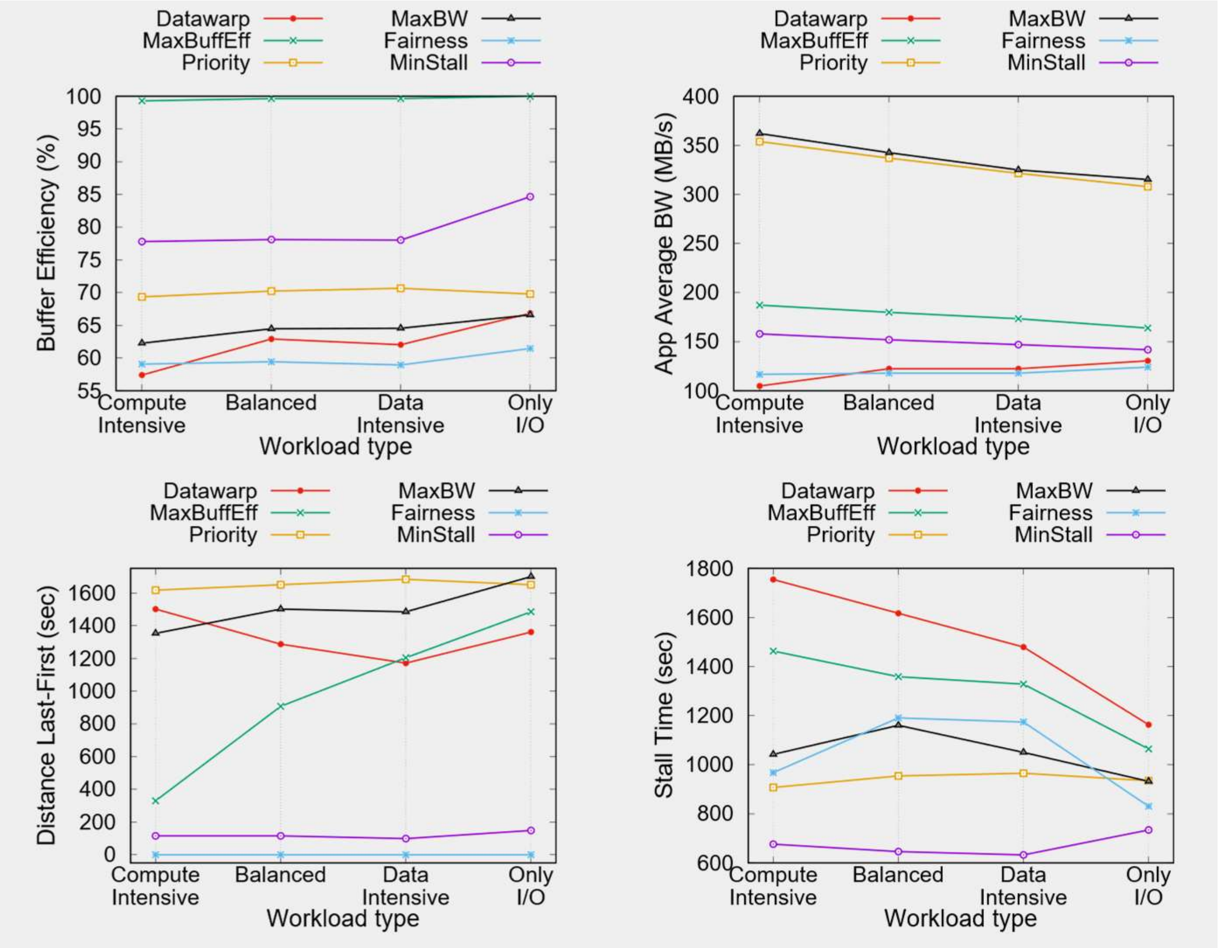
Anthony Kougkas, Hariharan Devarajan, Jay Lofstead, and Xian-He Sun. "*Harmonia: An Interference-Aware Dynamic I/O Scheduler for Shared Non-Volatile Burst Buffers*." In 2018 IEEE International Conference on Cluster Computing (CLUSTER), pp. 290-301. IEEE, 2018

- 40% faster execution than DataWarp for 8 concurrent instances
- 4% overhead on average to perform I/O phase detection offline
- MaxBW offers the best I/O time whereas Fairness the slowest I/O
- Harmonia's scheduling policies offer greater flexibility to the system

Scheduling Metrics

- Max Buffer Efficiency:
  - Harmonia can be **2x** more efficient
- Maximum Buffer Bandwidth:
  - Harmonia can offer **3x** higher average bandwidth
- Application Fairness:
  - Harmonia can achieve **10x** higher fairness
- Minimum Stall Time (waiting time):
  - Harmonia can minimize stall time for application by **3x**

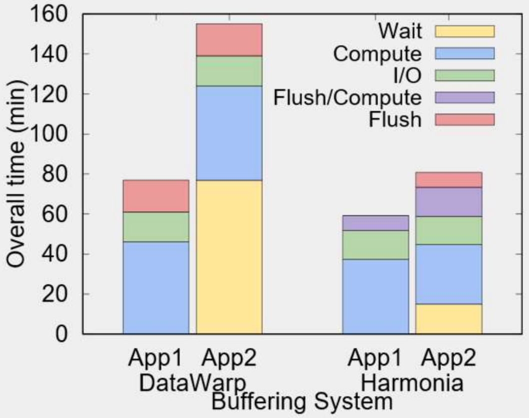
Anthony Kougkas, Hariharan Devarajan, Jay Lofstead, and Xian-He Sun. "*Harmonia: An Interference-Aware Dynamic I/O Scheduler for Shared Non-Volatile Burst Buffers*." In 2018 IEEE International Conference on Cluster Computing (CLUSTER), pp. 290-301. IEEE, 2018



- Harmonia’s policies can better adapt to workloads than other buffering systems

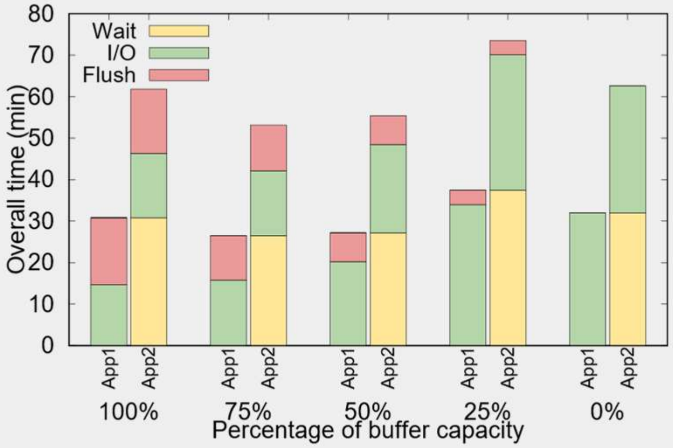
Buffer Draining

- Buffer draining: flushing of data from buffers to the persistent layer(i.e.,PFS)
- 2 instances of VPIC:
  - Buffer can hold data only for 1 instance
  - In each step:
    - Computation phase
    - Writing data to buffers
  - Harmonia leverages computation phases to drain the buffers
- **2x** better performance than DataWarp
- Flushing threshold initiates flushing:
  - 100% case same behavior as DataWarp
  - 0% case incoming I/O conflicts with flush
  - 50-75% threshold offers the best overlapping of incoming I/O and flushing



Description	DataWarp	Harmonia
Min Completion Time	76.9	59.2
Max Completion Time	155.1	80.8
Average Completion Time	116	70
Average Wait Time	38.45	7.45

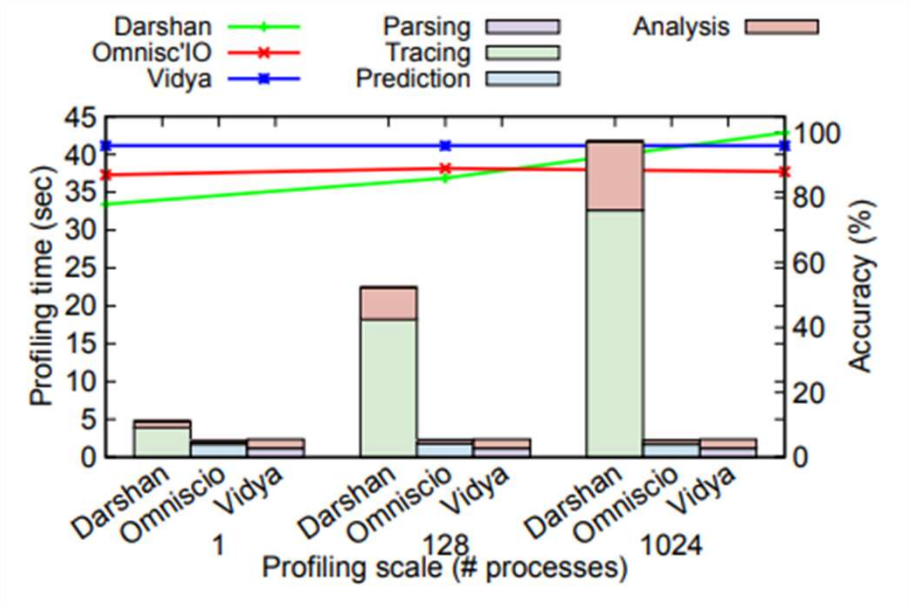
Test Metrics in Minutes



- Harmonia leverages computation to “hide” flushing

Vidya Profiling

- Profiling scale
  - Application: **CM1**
  - Goal: Predict I/O intensity
  - Sensitive for Darshan
- Vidya and Omnisc'IO are not affected by scale.
- Darshan's accuracy is better **but** that is a trade-off of profiling cost



Hariharan Devarajan, Anthony Kougkas, P. Challa, and Xian-He Sun. "*Vidya: Performing Code-Block I/O Characterization for Data Access Optimization*", Proc. of the IEEE International Conference on High Performance Computing, Data, and Analytics 2018 (HiPC'18)

SCALABLE COMPUTING SOFTWARE LABORATORY

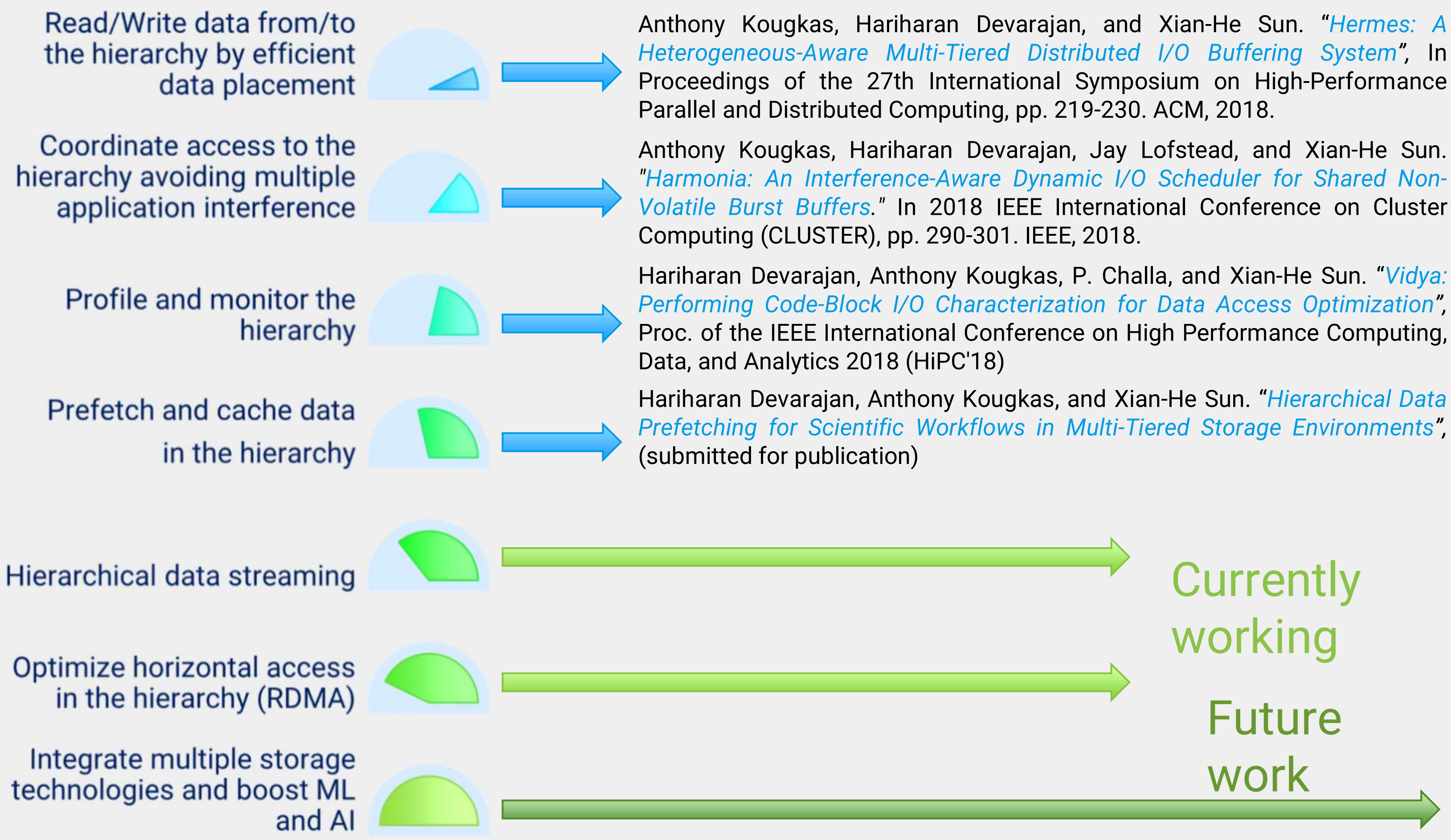
ILLINOIS INSTITUTE OF TECHNOLOGY

Hermes

Current state & Conclusion

Hermes

Major Steps



Read/Write data from/to the hierarchy by efficient data placement

Anthony Kougkas, Hariharan Devarajan, and Xian-He Sun. “*Hermes: A Heterogeneous-Aware Multi-Tiered Distributed I/O Buffering System*”, In Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing, pp. 219-230. ACM, 2018.

Coordinate access to the hierarchy avoiding multiple application interference

Anthony Kougkas, Hariharan Devarajan, Jay Lofstead, and Xian-He Sun. "*Harmonia: An Interference-Aware Dynamic I/O Scheduler for Shared Non-Volatile Burst Buffers*." In 2018 IEEE International Conference on Cluster Computing (CLUSTER), pp. 290-301. IEEE, 2018.

Profile and monitor the hierarchy

Hariharan Devarajan, Anthony Kougkas, P. Challa, and Xian-He Sun. “*Vidya: Performing Code-Block I/O Characterization for Data Access Optimization*”, Proc. of the IEEE International Conference on High Performance Computing, Data, and Analytics 2018 (HiPC'18)

Prefetch and cache data in the hierarchy

Hariharan Devarajan, Anthony Kougkas, and Xian-He Sun. “*Hierarchical Data Prefetching for Scientific Workflows in Multi-Tiered Storage Environments*”, (submitted for publication)

Hierarchical data streaming

Optimize horizontal access in the hierarchy (RDMA)

Currently working

Integrate multiple storage technologies and boost ML and AI

Future work

Hermes Impact

- Accelerate applications’ I/O access by transparently leveraging the DMSH.
  - Data are moved through the hierarchy effortlessly.
  - Applications have a scalable middleware software to navigate the I/O challenges.
- Leverage the HDF5 ecosystem to reach a wide scientific audience:
  - HDF5 is already used by the majority of users in the scientific community (95% market).
  - HDF5 is a building block for many other high-level I/O level libraries such as pNetCDF, MOAB, CGNS, and Silo
- Merging HPC, big data, and AI technologies
  - Utilize self-learning techniques to discern the application’s I/O behavior and configure the system accordingly
  - Merging memory and storage
- A foundation for data-centric system design
  - Forward thinking results: LABIOS, Compression
  - The next step: file systems, memory systems, OS

Conclusion

- Data Access becomes the bottleneck of computing
- Many new technologies are developed, but not well utilized
- Hermes is proposed to address I/O issues of DMSH
- It is workable accelerator and requests system enhancement
- It builds a foundation for next generation data-centric system design

TOO MANY THINGS NEED TO DO
FROM FOUNDATION TO SOFTWARE

SCALABLE COMPUTING
SOFTWARE LABORATORY

ILLINOIS INSTITUTE
OF TECHNOLOGY

Hermes

Q&A