A Dynamic Multi-Tiered Storage System for Extreme Scale Computing

SCALABLE COMPUTING SOFTWARE LABORATORY

Hariharan Devarajan Anthony Kougkas Xian-He Sun
hdevarajan@hawk.iit.edu, akougkas@iit.edu, and sun@iit.edu

ILLINOIS INSTITUTE OF TECHNOLOGY

ABSTRACT

In the era of data explosion, where data analysis is essential for scientific discoveries, the slow storage system has led to the research conundrum known as I/O bottleneck. Additionally, the explosion of data has led to proliferation of application as well as storage technologies. This has created a complex matching problem between diverse *application requirements* and *storage technology features*.

In this proposal, we introduce Jal, a dynamic, re-configurable, and heterogeneous-aware storage system. Jal utilizes a layered approach including application model, data model, and storage model. Our evaluations have shown these models, can accelerate I/O for the application while transparently and efficiently utilizing the diverse storage systems.

Poster QR

Website QR

CHALLENGES

Challenge 1: *How to understand and characterize the cause of application I/O behavior?*

1. Understanding the applications' I/O behavior is cumbersome and the research has been focused on understanding "What Happened".
2. However, they have to provide manual analysis and heuristics to determine its causal relationship for the observed I/O behavior.

Challenge 2: *How to match diverse application requirements with storage configurations?*

1. Scientific workflows require a diverse set of performance requirements to perform I/O.
2. However, modern storage system are not re-configurable to adapt to conflicting I/O requirements and complex heterogeneous storage hierarchy.

Challenge 3: *How to design a dynamically re-configurable multi-tiered storage system?*

1. Modern system are multi-tenant and run a variety of workflows that have multiple conflicting requirements.
2. However, software stack is designed for static, homogeneous and fixed software deployments and fail to cope in this dynamic environment.

HIGH-LEVEL DESIGN

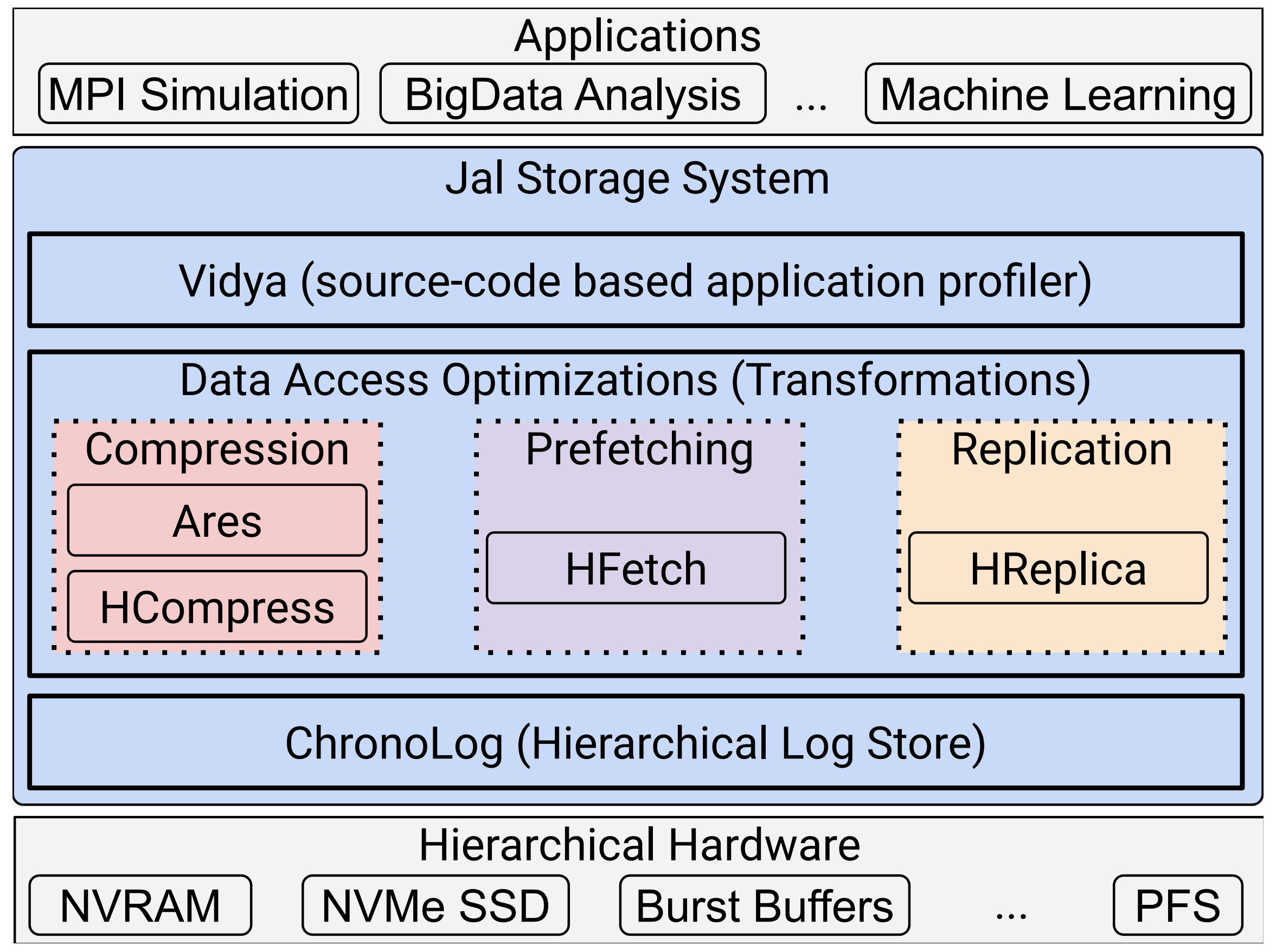

Figure 1: High-Level architecture of Jal Storage System

Jal storage system is a dynamic re-configurable multi-tiered storage system which can achieve perfect matching between application requirements and diverse storage technologies.

- **Application Model using Vidya** It uses a source-code based profiler which identifies the cause of the I/O behavior of applications. Using this approach, Vidya can enable automated optimization and insights on application's I/O behavior.
- **Storgae Model using ChronoLog** It builds a heterogeneous-aware storage system which can be dynamically re-configured to different storage configurations during runtime.
- **Data Model using Optimization** Each optimization translates different application's I/O requirement to underlying storage configuration to extract maximum performance of each applications. We develop novel data compression, data prefetching, and data replication engines that can transform different application requirements into storage configuration for optimizing I/O.

COMPONENT DESIGN

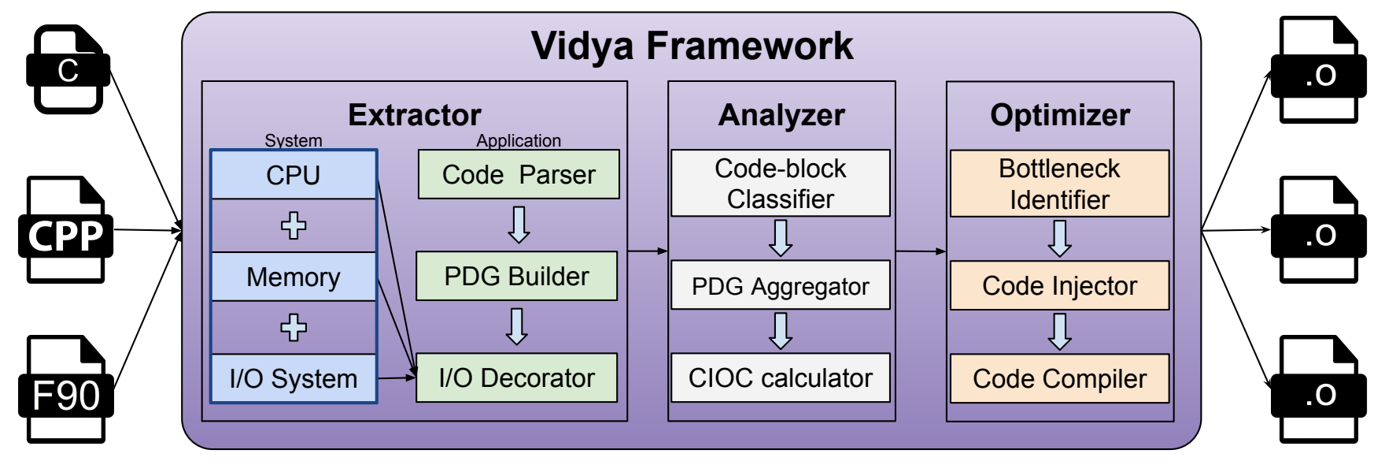

Figure 2: Vidya design

- **Profile:** extraction of code features.
- **Analyze:** transforming code features to app features.
- **Optimize:** use app features for automatic optimization.

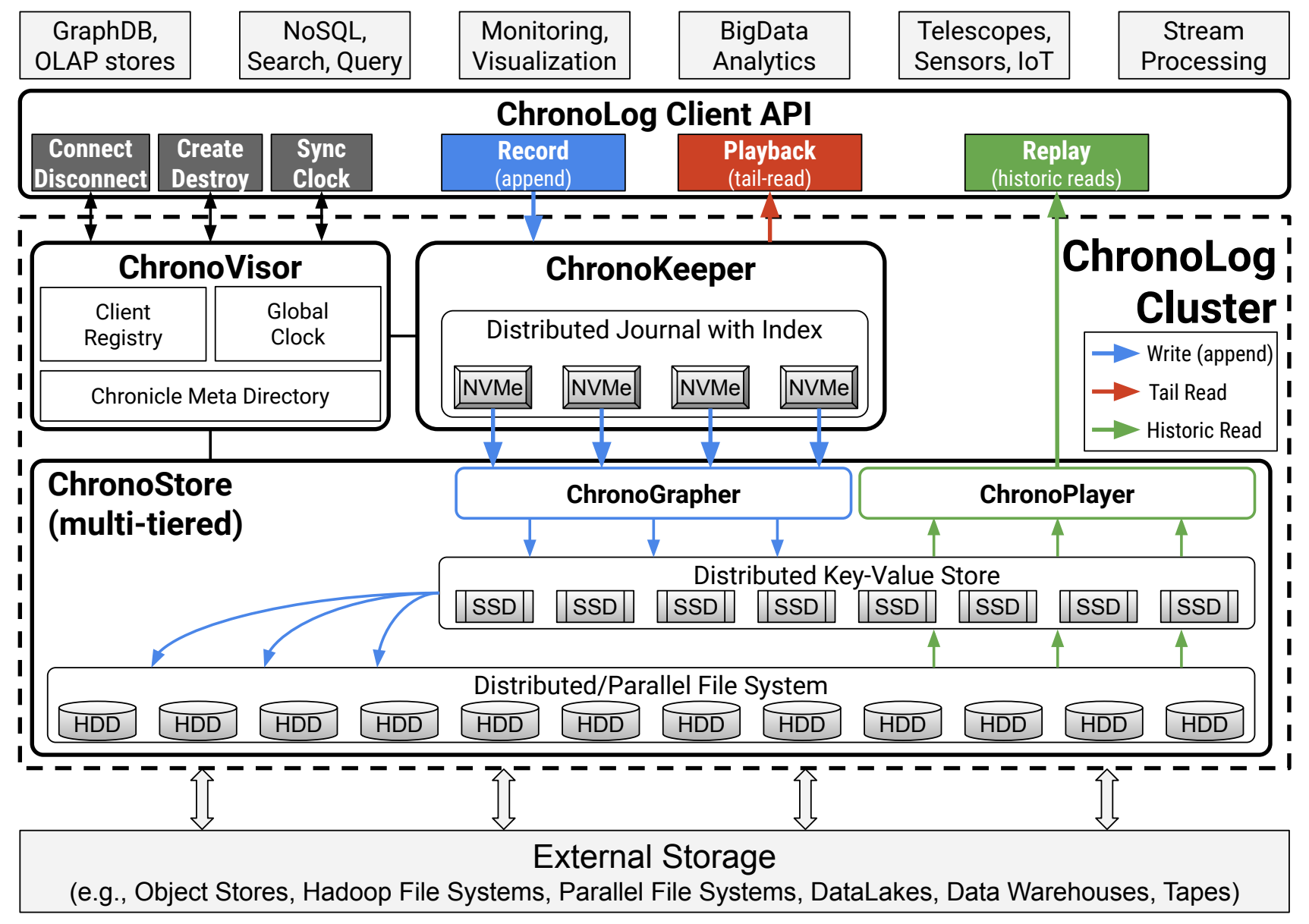

Figure 3: Chronolog design

- **Unify:** heterogenous hardware under single platform.
- **Chrono:** Timebased data ordering to avoid synchronization.
- **Stream:** paradigm to consume and retrieve data.
- **MWMR** clients do not disctate I/O parallelism.
- **Tunable** based on configuration different combination of hardware is utilized.

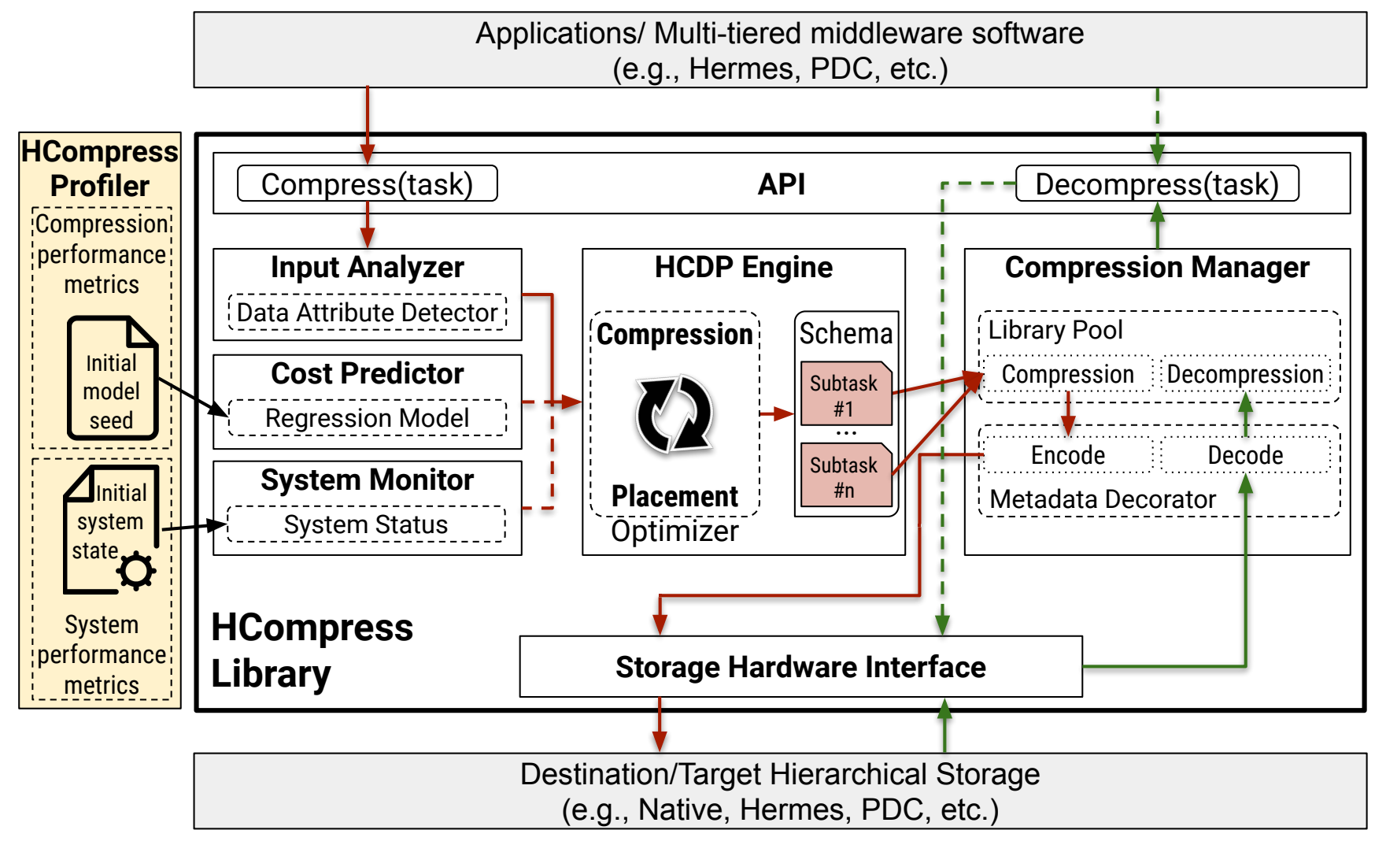

Figure 4: HCompress design

- **Adaptive** compression engine which can choose for different app requirement the ideal compression.
- **Dynamic** compressors can be changed on runtime to provide different performance characteristics.
- **Intelligent:** Feedback loop to improve prediction of performance counters for efficient placement.

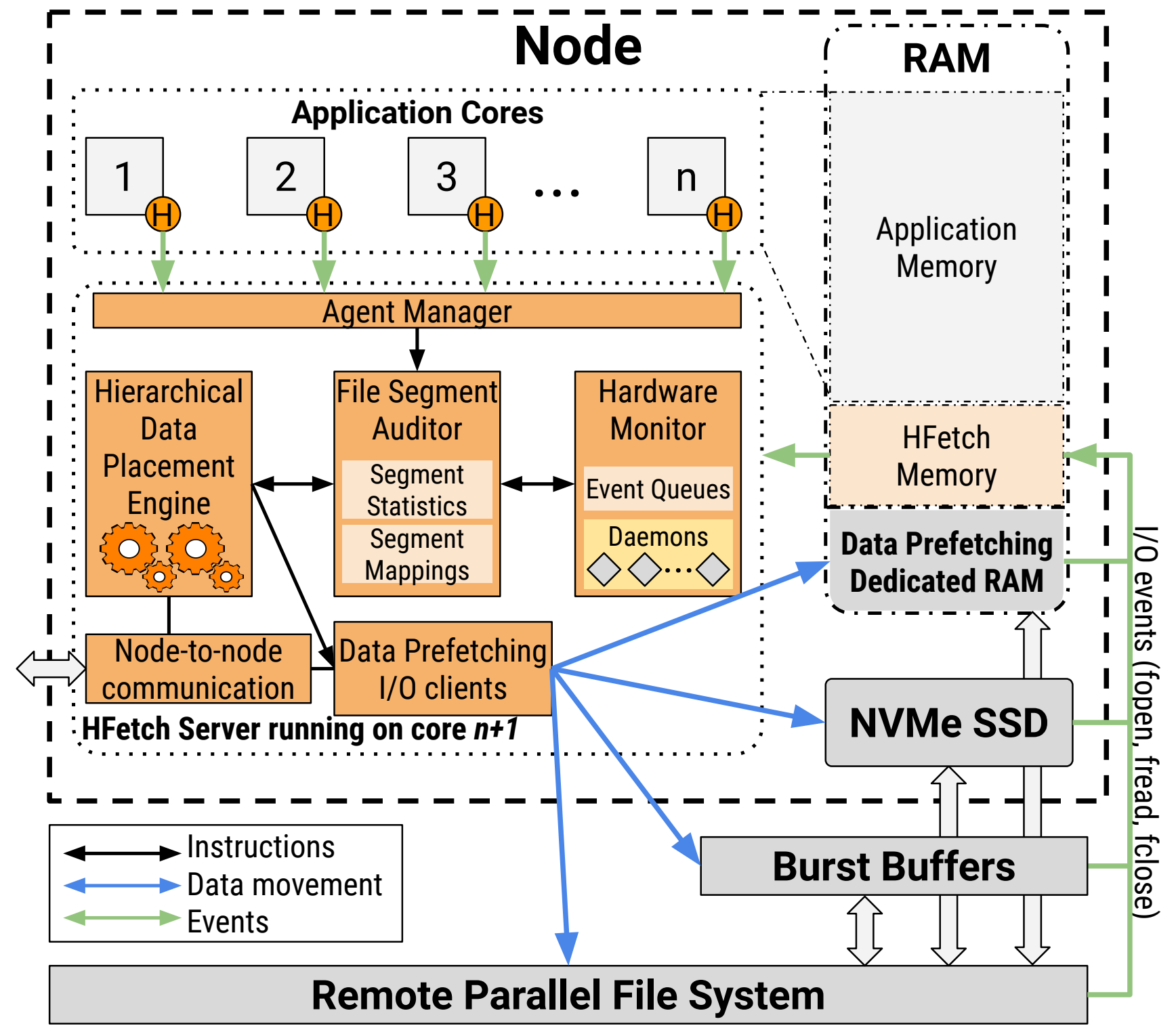

Figure 5: HFetch design

- **Pipeline** Enable hierachical pipeline for prefetching optimization.
- **Server-Push** utilize inotify to capture asynchronous I/O events and prefetch data.
- **Dynamic** Change to changing hotness of dataset and prefetch relevent pieces accordingly.
- **Global** Data-centric prefetching enables a global view of data instead of per-process/app view.

EVALUATIONS

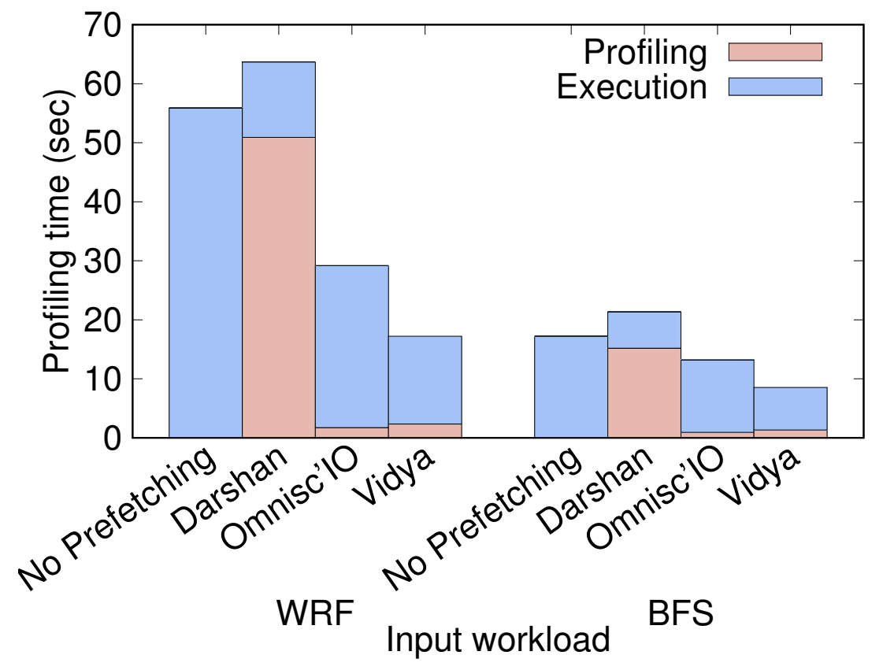

Figure 6: Vidya: Prefetching On/Off

Vidya profiling balances the trade-off bet. offline and online profiling techniques.

- **Profiling** more expensive than online
- **Prediction** less accurate than offline
- **Overall** Overall performance is better, due to a middle ground on trade-off.

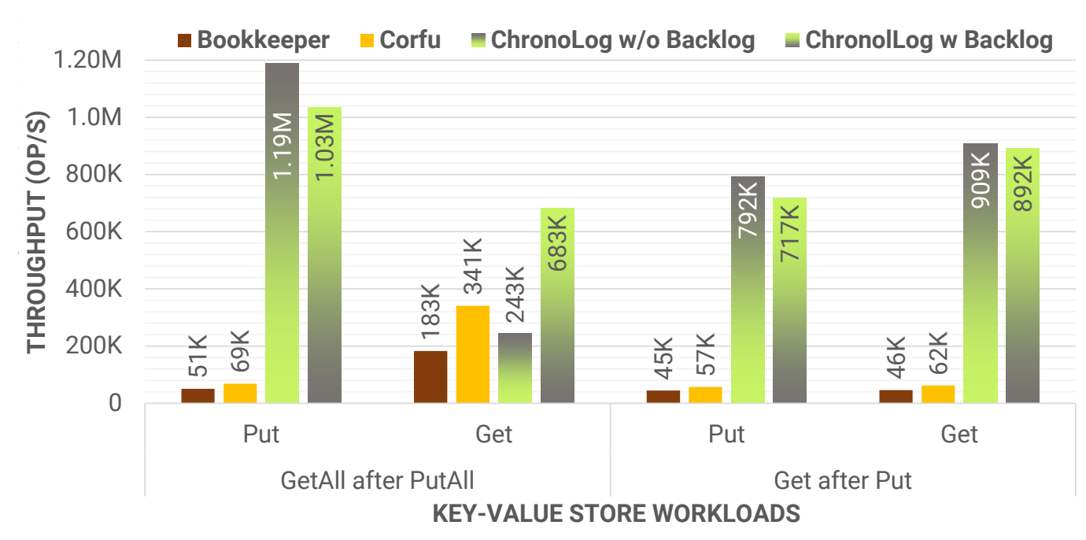

Figure 7: ChronoLog backend for KVS

Chronolog can be efficiently used as a backend for Key-Value store such as Redis.

- **Chrono** enables time based ordering and hence no synchronization during tail operations.
- **Hierarchical** Uses NVMe as a fast catch for recent data and hence get after put is more stable than get all after put all.

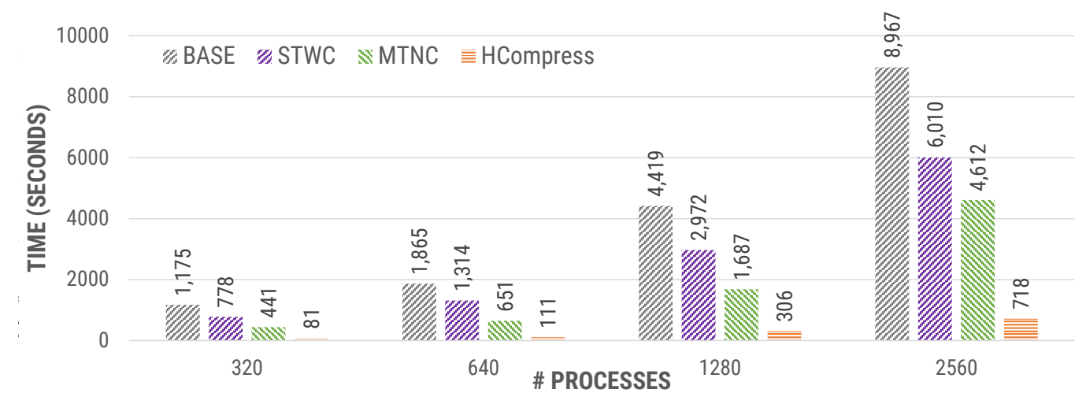

Figure 8: HCompress: Scientific workflow

Optimizes complex workflow of Simulation (VPIC) + Analysis (BD-CATS).

- **System** Based on weights system is configured to use the most appropriate compression.
- **Hierarchical** Engine utilizes better layers and save more data on higher layers.

CONCLUSIONS

We presented Jal storage system, a dynamic, multi-tiered storage system which can be reconfigured for different application requirement on runtime. We discussed our layered methodology along with of extracting application requirements, converting them into various storage configuration, and finally a building a malleable log store which can deploy those storage configurations. We showcased through evaluations that this approach can lead to 7x better performance over existing storage systems by utilizing several automated layers of source-code based profiler, data access optimizations, and a distributed Log store. Additionally, we showcased that this approach is viable for modern multi-tenant super computers with good performance benefits.

PAPERS

[1] Devarajan, H., Kougkas, A., Challa, P., and Sun, X.-H. (2018). Vidya: Performing code-block io characterization for data access optimization. In *2018 IEEE 25th International Conference on High Performance Computing, Data, and Analytics*. IEEE.

[2] Devarajan, H., Kougkas, A., Logan, L., and Sun, X.-H. (2020a). Hcompress: Hierarchical data compression for multi-tiered storage environments. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*.

[3] Devarajan, H., Kougkas, A., and Sun, X.-H. (2019). An intelligent, adaptive, and flexible data compression framework.

[4] Devarajan, H., Kougkas, A., and Sun, X.-H. (2020b). Hfetch: Hierarchical data prefetching for scientific workflows in multi-tiered storage environments. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 62–72.

[5] Kougkas, A., Devarajan, H., Bateman, K., Cernuda, J., Rajesh, N., and Sun, X.-H. (2020). Chronolog: A distributed shared tiered log store with time-based data ordering. In *Proceedings of the 36th International Conference on Massive Storage Systems and Technology (MSST 2020)*.