HFlush is a pull-based data flusher that implements a continuous data eviction mechanism. Initial results have shown HFlush to be a promising solution to the growing challenge of extreme scale data generation, especially in case of workloads with periodic I/O or in systems that make use of modern hardware with high concurrency. The ability of HFlush to amortize the I/O stall time allows applications using it to significantly increase the CPU usage by over 50%. The near real-time nature of the eviction provides an improved overall latency on the data flushing with a 7X latency reduction and a 2X bandwidth increase over batch-based flushing solutions, which reflects in a lower I/O stall time for the application.

HFlush
Realtime Flushing for Modern Storage Environments
Jaime Cernuda, Hugo Trivino, Hariharan Devarajan, Anthony Kougkas, Xian-He Sun

OVERVIEW
- A disparity in speed between CPU and storage access time has created what is known as the I/O bottleneck.
- To solve this issue, traditional solutions have involved data buffering and aggregations on fast storage mediums.
- However, faster tiers of data storage, such as RAM, have lower storage capacity which eventually require eviction of data to a lower tier, typically a Parallel File System (PFS).

History

- Current eviction solutions are event-based and fail I/O when performing evictions.
- Evictions are initiated by individual nodes, without reliability, and provide writing patterns not favorable to the PFS.
- Enhanced capabilities of the new storage devices (e.g., NVMe SSD) such as increased hardware concurrency are not taken into account by existing system software.

Approach
- Amortize the cost of evictions into small continuous flushing operations instead of irregularly calling IO operations.
- Globally coordinate all evictions to provide better writing patterns to the PFS and match the demand by offering elastic resources.
- Leverage the hardware concurrency to perform device-specific eviction optimizations.
- Move to a server pull eviction model to provide a continuous stream of evictions.

HFLUSH DESIGN

Backlog
- Makes flushed data available in the memory.
- Data is written to the tail.
- Data between the tail and head is safe in the PFS and can be overridden.

Active Operators
- Can perform operations that make use of the global nature of HFlush
  - deduplication
  - sorting
  - compression

Data Collector
- Pulls objects from the head of the backlog.

Data Dispatcher
- Splits the I/O to write objects into the PFS.

HFlush Performance

WORKLOADS

INITIAL EVALUATIONS

CONCLUSIONS

- HFlush is a pull-based data flusher that implements a continuous data eviction mechanism.
- Initial results have shown HFlush to be a promising solution to the growing challenge of extreme scale data generation, especially in case of workloads with periodic I/O or in systems that make use of modern hardware with high concurrency.
- The ability of HFlush to amortize the I/O stall time allows applications using it to significantly increase the CPU usage by over 50%.
- The near real-time nature of the eviction provides an improved overall latency on the data flushing with a 7X latency reduction and a 2X bandwidth increase over batch-based flushing solutions, which reflects in a lower I/O stall time for the application.

RELATED WORK

Leveraging a data streaming paradigm enables:
- Autoscaling
- Data durability
- Pipelined flushing in multi-tiered environments
- Matching hardware properties from source and destination

Related Work

- CPU: Dual Intel(R) Xeon Scalable Silver 4114 @ 2.20GHz (40 nodes)
- RAM: 96 GB RAM
- Network: 10GbE Ethernet with RoCE
- Storage: local 512GB NVMe SSD.

- CPU: Dual Intel(R) Xeon Scalable Silver 4114 @ 2.20GHz (40 nodes)
- RAM: 32GB DDR2-667
- Storage: 1 250GB Samsung 860 Evo SATA SSD, 1TB Seagate 7200K SATA hard drive

- CPU: Two quad-core Opteron 2376 @ 2.3GHz (40 nodes)
- RAM: 32GB DDR2-667
- Network: 10GbE Ethernet with RoCE
- Storage: 1 250GB Samsung 860 Evo SATA SSD, 1TB Seagate 7200K SATA hard drive


Jaime Cernuda
Illinois Institute of Technology
jcernudagarcia@hawk.iit.edu

Hugo Trivino
Illinois Institute of Technology
hhernandeztrivino@hawk.iit.edu