Lessons from Post-processing Climate Data on Modern Flash-based HPC Systems

[Extended Abstract]

Adnan Haider
Illinois Institute of Technology
3300 South Federal Street
Chicago, United States
ahaider3@hawk.iit.edu

Sheri Mickelson (Advisor)
National Center of
Atmospheric Research
1850 Table Mesa Drive
Boulder, United States
mickelso@ucar.edu

John Dennis (Advisor)
National Center of
Atmospheric Research
1850 Table Mesa Drive
Boulder, United States
dennis@ucar.edu

Xian-He Sun (Advisor)
Illinois Institute of Technology
3300 South Federal Street
Chicago, United States
sun@iit.edu

1. INTRODUCTION

Post-processing climate data applications are heavily relied upon for facilitating scientific discovery. Because these applications are I/O bound, flash-based HPC systems are compelling for running post-processing applications. HPC systems incorporate flash devices as *local* storage for compute nodes, a separate *pooled* flash-only filesystem, or as a burst buffer. However, the tradeoffs associated with these different architectures is currently unclear. Thus, with the goal of matching multiple varying I/O workloads with different flash storage architectures, we analyze the performance of a local and pooled flash architecture to clearly quantify their tradeoffs.

2. APPLICATION WORKLOAD

The post-processing applications, PyAverager and PyReshaper, were evaluated on flash architectures. The applications use posix I/O to read/write data in parallel [3] with small request sizes. We tested five datasets from the Community Earth System Model (CESM), each with varying request sizes. In Figure 1, I/O time accounts for, on average, 90% of total execution time. By using workloads from both post-processing climate data applications and IOR [1], a popular I/O benchmark, we can extend our results to many other I/O intensive workloads.

3. GORDON RESULTS

Gordon has a local I/O architecture, meaning each compute node can read/write to a single solid state drive (SSD). SSDs are located on I/O nodes, which are connected to compute nodes via an infiniband interconnect [4]. As seen in Figure 2, the flash devices *increased* runtime by a factor of 2 compared to Gordon's disk file system for the more I/O intensive datasets. The single SSD for each compute node and single infiniband interconnect resulted in scalability and latency issues due to contention over the network and accesses becoming queued on the SSD.

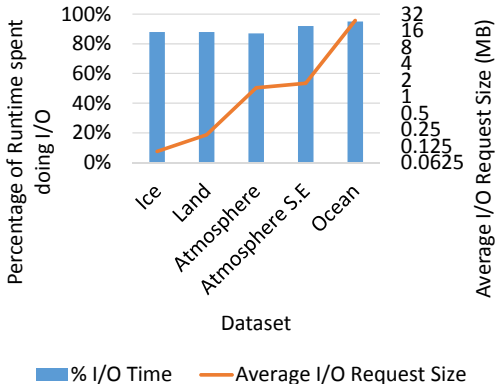

Figure 1: The I/O characteristics of PyReshaper for five different datasets from the Community Earth System Model. The average I/O request size is the average amount of data being read or written on a single I/O call. In addition to these workloads, we use the IOR benchmark which can be used for comparison with other scientific I/O workloads.

From running IOR (data on poster), we found that local flash architectures offer fast I/O access for applications which output small intermediate data or have a small number of parallel I/O requests. Also, we found that as the number of processes, data amount, and request size increase, the performance advantage of SSDs decreases due to the single flash device per compute node.

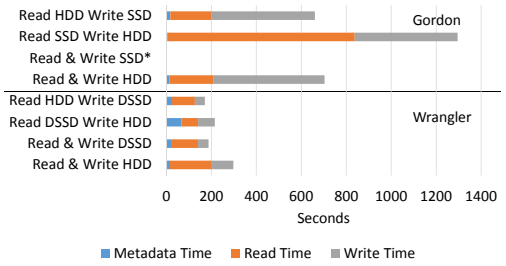

Figure 2: Gordon's (top) and Wrangler's (bottom) PyReshaper timings using 16 processes running on the Ocean dataset. Gordon's flash increases runtime by 2X since its using a single SSD. Reading and writing to SSD causes the SSD to reach capacity and the application fails. Wrangler's pooled DSSD architecture allows parallel access to multiple flash devices decreasing runtime by half compared to Wrangler's read & write HDD.

4. WRANGLER RESULTS

Wrangler, released in 2015, provides 500TB of DSSD storage, an order of magnitude lower latency flash device. Wrangler is a pooled flash architecture, meaning each compute node has access to all 500TB of flash storage via a PCI Express connection [2]. Since all flash devices are underneath a parallel file system, scalability issues seen in a local flash architecture are non-existent. Since the interconnect is PCI Express as compared to Infiniband, interconnect latency was not a significant factor for the applications. Wrangler provided up to a 6x *reduction* in execution time (data on poster) when reading from hard disk and writing to DSSD (hybrid I/O) compared to when reading and writing to Wrangler's hard disks. For all datasets, hybrid I/O provided the best performance with only half the flash storage consumption. When running IOR (shown on poster), the pooled DSSD devices provided consistent improvements for all request sizes and process counts, illustrating that a pooled architecture can provide significant improvement for many diverse I/O workloads, unlike Gordon's flash architecture which degraded in performance with larger request sizes and process counts.

5. COMPARISON OF I/O ARCHITECTURES

There are many local flash architectures deployed in current systems, so we tested (data on poster) how many flash devices are needed for sufficient performance using a local architecture. We discovered different datasets, depending on their I/O intensity, needed a different amount of SSDs. Thus, allocating a configurable amount of SSDs during job configuration can provide better resource utilization. After discussion with the Gordon architecture team, a hybrid configuration, meaning a configurable amount of flash devices per job, is possible on Gordon but not available as default.

We compared the speedup provided by flash for Wrangler and Gordon. Wrangler provided up to 2x more speedup which can be attributed to the DSSD devices and high throughput of PCI Express, since the impact of other hardware differences between the two systems was eliminated (details on poster). For some datasets, the performance provided by Wrangler is not significantly better than Gordon, leading us to conclude that a cheaper, more prevalent local architecture can be sufficient for post-processing applications. However, multiple flash devices per compute node, a high throughput interconnect, and the ability of processes to access data not on their own compute node's flash device will be essential.

We conducted tests (data on poster) which eliminated the performance impact of other hardware so we could compare the benefits of flash in isolation. From these tests, we found that running on three years of newer hardware (more memory, larger caches, faster interconnects, etc), without using flash provides more improvement than running on flash with all other hardware constant for our workloads. This means that although flash devices provide significant improvement (6x reduction in execution time), other hardware could be just as important when trying to accelerate I/O intensive workloads.

Overall from our experience of running on flash-based HPC systems, we observed:

- An incorrect matching between storage architecture and I/O workload can hide the benefits of flash by increasing runtime by 2x.
- Hybrid I/O decreases flash storage consumption by half while decreasing runtime by 6x.
- A local flash architecture is a cost-effective alternative to a pooled architecture if scalability and interconnect bottlenecks are alleviated.
- For our I/O workload, there are three main criteria which determine performance on flash-based HPC systems. 1)The number of flash devices in a job. 2)The interconnect between the CPU and flash device. 3)A scalable filesystem allowing all data to be seen by all processes.
- Using hardware that is three years newer provides more speedup than using flash devices for some datasets.

6. ACKNOWLEDGEMENTS

We thank XSEDE, TACC, and SDSC for their resources and help. TG-ASC150025

7. REFERENCES

[1] Nersc ior. https://www.nersc.gov/users/computational-systems/cori/nersc-8-procurement/trinity-nersc-8-rfp/nersc-8-trinity-benchmarks/ior/.

[2] N. Gaffney, C. Jordon, T. Minyard, and D. Stanzione. Building wrangler. In *IEEE International Coneference on Big Data*, 2014.

[3] S. Mickelson. Optimizing workflow for cesm. www.cesm.ucar.edu/events/ws.2015/presentations/sewg/, 2015.

[4] S. Strande, P. Cicotti, R. Sinkovits, W. Young, R. Wagner, M. Tatineni, E. Hocks, A. Snavely, and M. Norman. Gordon: design, performance, and experiences deploying and supporting a data intensive supercomputer. In *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond*, 2012.