

Using Darshan and CODES to Evaluate Application I/O Performance

Harsh Khetawat
Department of Computer Science
North Carolina State University
Raleigh, North Carolina
Email: hkhetaw@ncsu.edu

Christopher Zimmer
Oak Ridge National Laboratory
Oak Ridge, Tennessee
Email: zimmercj@ornl.gov

Frank Mueller
Department of Computer Science
North Carolina State University
Raleigh, North Carolina
Email: mueller@cs.ncsu.edu

Sudharshan Vazhkudai
Oak Ridge National Laboratory
Oak Ridge, Tennessee
Email: vazhkudaiss@ornl.gov

Scott Atchley
Oak Ridge National Laboratory
Oak Ridge, Tennessee
Email: atchleyes@ornl.gov

Abstract—Burst buffers have become increasingly popular in HPC systems, allowing bursty I/O traffic to be serviced faster without slowing down application execution. The ubiquity of burst buffers creates opportunities for studying their ideal placement in the HPC topology. Furthermore, the topology of the network interconnect can also affect the performance of the storage hierarchy for different burst buffer placement schemes. To that end we create a reproducible framework that allows individual centers to develop their own models and evaluate performance based on their workload characteristics. We use CODES to create models that simulate the network and storage layers of an HPC system and Darshan traces for I/O replay. We augment the Darshan traces with synchronization primitives, and allow multi-dimensional scaling of traces to represent future workload characteristics. Finally, we evaluate the effect of network topology, storage architecture, and application I/O patterns on overall I/O performance.

Keywords—HPC, burst buffers, simulation, I/O.

I. INTRODUCTION

With the increasing scale of computation and data in High Performance Computing (HPC), existing storage systems are becoming a bottleneck for large-scale scientific and data-intensive applications [1]. The ratio of flops to I/O bandwidth on Titan is 27:1, on Summit that ratio is increasing to 80:1. Data generation rates are increasing faster than traditional parallel file system (PFS) ingestion capabilities. To alleviate this bottleneck, the traditional PFS is being augmented with a tier of intermediate, high-bandwidth flash-based storage devices called burst buffers. These burst buffers sit between the compute nodes and the parallel file system and are designed to absorb the periodic bursts of I/O by HPC applications.

Various hierarchies exist for the placement of burst buffers in HPC systems:

- Burst buffers co-located with compute nodes, e.g., used in Summit, the next-generation OLCF system [2];
- Burst buffers co-located with I/O nodes, e.g., used in Cori, an HPC system at NERSC [3];

- Global burst buffers, e.g., used in Konrad, an HPC system at ZIB, Germany [4].

Moreover, modern HPC systems use a range of network topologies, e.g., fat-tree [5], dragonfly [6], torus, etc. These network topologies have different blocking characteristics, diameters, bisection bandwidths, and costs. The performance of burst buffers depends on the network and storage contention that I/O traffic experiences during execution of a typical application workload.

Using a Parallel Discrete Event Simulator (PDES), we simulate a full HPC system with different network topologies and burst buffer architectures. The simulator is then used to replay I/O traces from real-world applications, which then mirrors the expected performance of a real system. Using these simulations and models, we then evaluate various architectures and determine the ideal choices based on the application workload, performance requirements, and cost constraints.

The primary contributions of this work are the following:

- We use the CODES simulation suite to simulate a full HPC system. We use the network and burst buffer models available in CODES in order to simulate different HPC deployments.
- We evaluate the performance of these models on real-world application workloads using Darshan I/O traces. We extend the Darshan logs and the simulator to scale the traces to reflect their size and distribution on a different HPC system from where they are collected in support of both strong and weak scaling.
- We create a framework with the above models and extensions that will allow HPC centers to assess expected performance of HPC systems based on their respective workloads.
- The framework can facilitate development of parallel file systems, data staging schemes, traffic shaping algorithms and resiliency techniques by performing sensitivity studies to parametric variations.

II. SOLUTION APPROACH

We use a Parallel Discrete Event Simulator (PDES) [7] called CODES [8][9] to simulate various HPC deployments. The CODES framework builds on ROSS to allow simulations of exascale network and storage architectures in HPC environments. CODES abstracts the network models as components to create packet-level simulations of the most popular HPC network topologies. It supports dragonfly, slimfly, torus and fat-tree topologies. It also has support for storage models including a local storage, and a CODES store model, which can be used to simulate burst buffers.

In order to model real world HPC workloads we use Darshan [10] traces collected from application executions at the Oak Ridge Leadership Computing Facility (OLCF). We use the traces collected by these systems to resemble the I/O behavior of scientific applications in our simulation. Since these traces are collected from current executions of the application, they do not accurately represent future workload sizes so we augment the Darshan traces in the following ways:

- We add the ability to weak scale the application by increasing the number of ranks while maintaining the same I/O behavior.
- We add synchronization information to the traces to accurately capture application behavior. This allows us to not have to rely on absolute timing information from the Darshan trace.
- We can increase the number of I/O phases of the application to model future runtimes.
- We can also scale the application I/O sizes to represent larger future workloads.

To simulate a real HPC environment, our framework also supports striping I/O across multiple storage devices, client-side buffering of I/O and tracking individual I/O phases of the application. Since applications are seldom allocated contiguous nodes in an HPC system, the framework also allows for fine-grained allocation of compute nodes to applications, and for simulating multiple applications running simultaneously.

Finally, we evaluate the effect of different workloads on the I/O performance of the system with HPC applications widely used in leadership computing facilities. GTC [11] is a highly scalable scientific application that simulates billions of plasma particles inside a reactor. S3D [12] is a direct numerical simulation (DNS) code used in computational fluid dynamics, which solves the Navier-Stokes equations. HACC or Hardware/Hybrid Accelerated Cosmology Code [13] is used to conduct high resolution simulations of cosmological structure for modern-day galactic surveys.

Figure 1 shows the I/O phases of the applications across various burst buffer and network configurations. The choice of network topology has no effect on the performance of the node-local burst buffers. GTC exhibits highly bursty I/O causing significant variation in performance across configurations. The grouped burst buffer performs much better than global for the dragonfly network while the performance is similar for the fat-tree networks. The node-local burst buffers significantly outperform the global and grouped for highly bursty traffic. HACC has scattered I/O behavior. This causes individual ranks

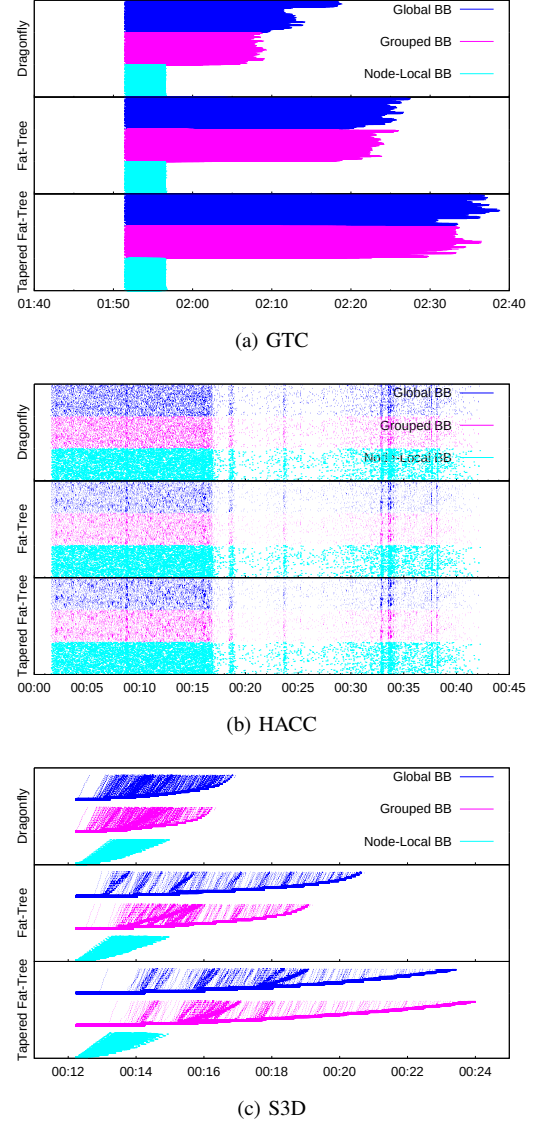


Fig. 1. I/O phases of each application across network topologies and burst buffer architectures.

to spend different amounts of time for different configurations but the overall I/O phase of the application remains unaffected. Finally, S3D exhibits staggered I/O behavior but has localized bursty traffic, resulting in node-local burst buffers outperforming other configurations.

III. CONCLUSION

From our experimental results we can see the effect of burst buffer placement, network topology and application I/O patterns on overall application performance. Certain applications such as HACC have I/O patterns that are not affected by the placement of storage resources or the HPC network topology. On the other hand GTC and S3D which have bursty I/O has a significant performance impact depending on the choice of burst buffer architecture and network topology. Furthermore, we are looking at how various storage architectures and network topologies impact the I/O performance of the application when multiple applications run simultaneously as well as alongside adversarial jobs.

REFERENCES

- [1] Bing Xie, Jeffrey Chase, David Dillow, Oleg Drokin, Scott Klasky, Sarp Oral, and Norbert Podhorszki. Characterizing output bottlenecks in a supercomputer. In *High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for*, pages 1–11. IEEE, 2012.
- [2] Oak Ridge National Laboratory. Summit. <https://www.olcf.ornl.gov/olcf-resources/compute-systems/summit/>.
- [3] National Energy Research Scientific Computing Center. Cori. <http://www.nersc.gov/users/computational-systems/cori/>.
- [4] Zuse Institute Berlin. Konrad. <http://www.zib.de/features/supercomputing-zib>.
- [5] Charles E Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE transactions on Computers*, 100(10):892–901, 1985.
- [6] John Kim, Wiliam J Dally, Steve Scott, and Dennis Abts. Technology-driven, highly-scalable dragonfly topology. In *Computer Architecture, 2008. ISCA'08. 35th International Symposium on*, pages 77–88. IEEE, 2008.
- [7] Richard M Fujimoto. Parallel discrete event simulation. *Communications of the ACM*, 33(10):30–53, 1990.
- [8] Misbah Mubarak, Christopher D Carothers, Robert B Ross, and Philip Carns. Enabling parallel simulation of large-scale hpc network systems. *IEEE Transactions on Parallel and Distributed Systems*, 28(1):87–100, 2017.
- [9] Jason Cope, Ning Liu, Sam Lang, Phil Carns, Chris Carothers, and Robert Ross. Codes: Enabling co-design of multilayer exascale storage architectures. In *Proceedings of the Workshop on Emerging Supercomputing Technologies*, volume 2011, 2011.
- [10] Philip Carns, Robert Latham, Robert Ross, Kamil Iskra, Samuel Lang, and Katherine Riley. 24/7 characterization of petascale i/o workloads. In *Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on*, pages 1–10. IEEE, 2009.
- [11] Zhihong Lin, Taik Soo Hahm, WW Lee, William M Tang, and Roscoe B White. Turbulent transport reduction by zonal flows: Massively parallel simulations. *Science*, 281(5384):1835–1837, 1998.
- [12] David Lignell, Chun Sang Yoo, Jacqueline Chen, Ramanan Sankaran, Mark Fahey, Evatt Hawkes, Tianfeng Lu, and Chung Law. S3d: Petascale combustion science, performance, and optimization collaborators. 06 2018.
- [13] Salman Habib, Adrian Pope, Hal Finkel, Nicholas Frontiere, Katrin Heitmann, David Daniel, Patricia Fasel, Vitali Morozov, George Zagaris, Tom Peterka, et al. Hacc: Simulating sky surveys on state-of-the-art supercomputing architectures. *New Astronomy*, 42:49–65, 2016.