Simulating the Burst Buffer Storage Architecture on an IBM BlueGene/Q Supercomputer

Jian Peng[1], Sughosh Divanji[1], Ioan Raicu[1,2] ,Michael Lang[3]
[1] Illinois Institute of Technology, Chicago, IL, USA
[2] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA
[3] Los Alamos Scientific Laboratory, Los Alamos, NM, USA

Goal

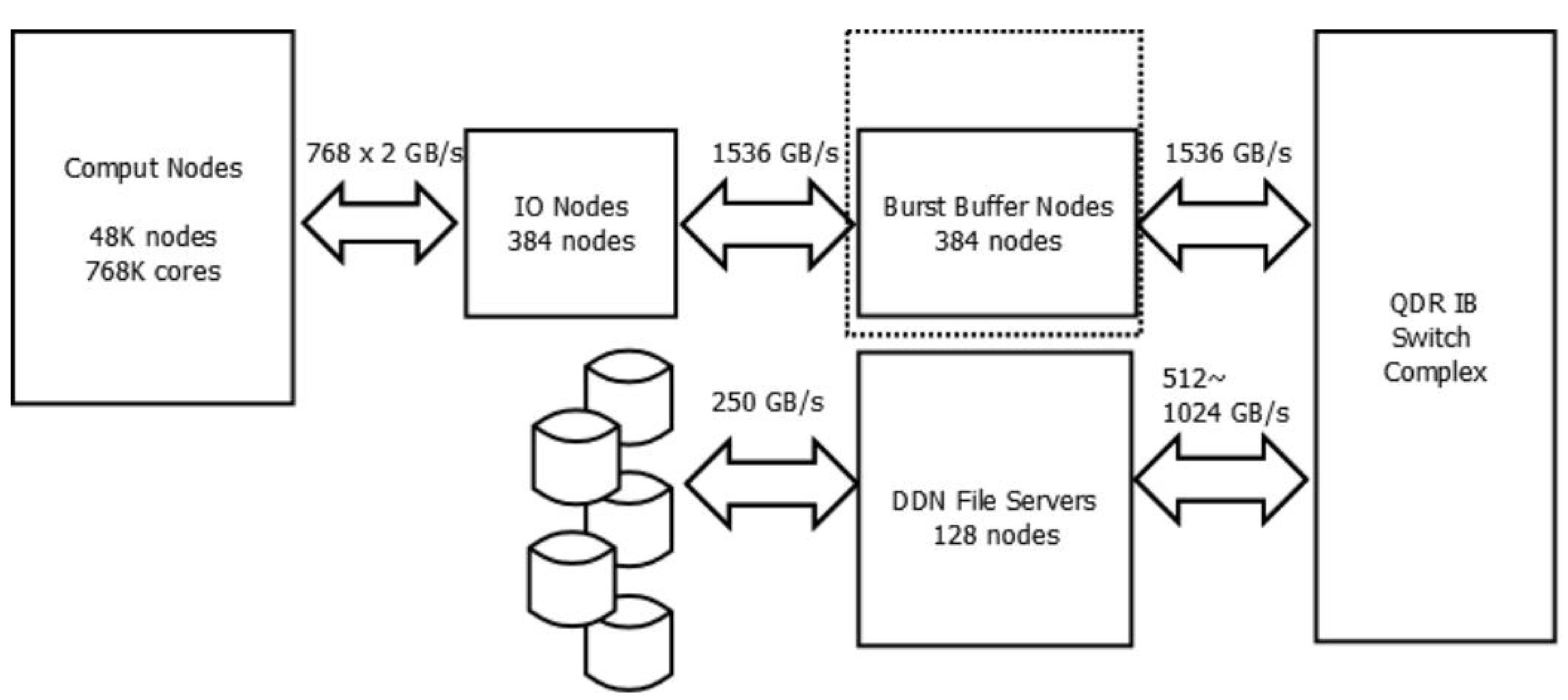
In this poster, we simulated the burst buffer storage architecture on an IBM BlueGene/Q supercomputer (with a 5D Torus network interconnect) using the CODES/ROSS simulation framework to study the potential I/O performance improvement with burst buffers. These results are a stepping stone towards studying this new storage architecture on future dragon-fly network based supercomputers, including resource management covering both storage and job scheduling.

Introduction

Burst buffer architecture has been proposed to handle bursty I/O patterns in HPC systems[1]. Burst buffers are high-throughput, low-capacity storage devices that act as a staging area or a write-behind cache for HPC storage systems. The approach we follow to incorporate burst buffers is to place these buffers on I/O nodes that connect to the external storage system and to manage these buffers as part of the I/O forwarding services. If burst buffers are sufficiently large and fast, they can absorb I/O bursts [2]. Meanwhile, burst buffer layer can provide some distributed file systems [7] more opportunities to improve data locality which is crucial for application performance [8].

We have implemented our simulation using CODES (Co-Design of Multilayer Exascale Storage Architectures)[3] and ROSS (Rensselaer Optimistic Simulation System)[4] simulation frameworks. ROSS is a parallel discrete event simulator which uses time warp protocols[5] to simulate discrete events in parallel.

Figure 1 Modeling Blue Gene/Q (Mira)



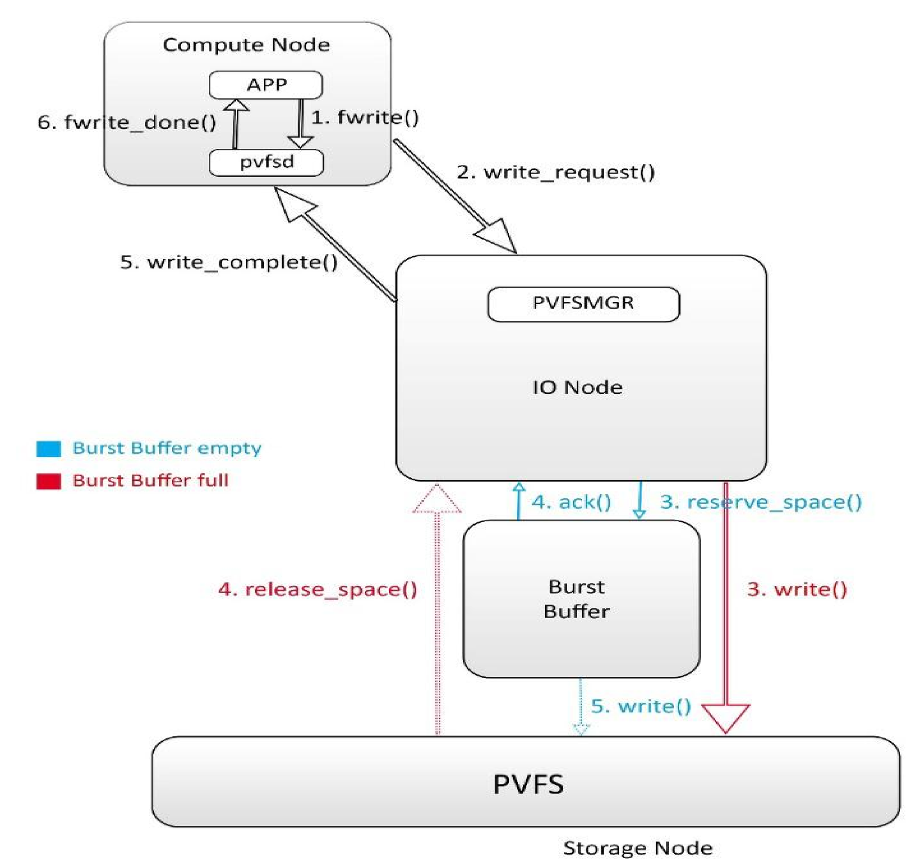
We measure the performance of our simulation for I/O operations of in Intrepid and Mira at ALCF and compare the performance of the burst buffer architecture to architecture without burst buffers through simulations. Figure 1 shows the Blue Gene/Q (Mira) modeling details

Implementation

Figure 1 is our modeling details of Mira. Compute nodes(CN) are located in a 5D torus network. There are 16 computing cores on each CN. One midplane contains 512 CNs in 4 × 4 × 4 × 4 × 2 structure. One rack contains 2 midplanes and Mira has 48 racks which sums up to 49,152 CNs, 786,432 cores. The bandwidth of the torus link is 2GB/s in both directions. There are 384 IO nodes(ION) connected to the torus network through specialized compute nodes called "bridge nodes". The CN/ION ratio in our simulation is 128: 1.The bandwidth of each interconnection link between ION and DDN file system server is 4GB/s, which sums up to 1536 GB/s. There are 128 DDN file servers connected to disk arrays with total bandwidth of 250GB/s [6].

We have implemented a write protocol simulating the file write procedure on Mira as Figure 2.

Figure 2 Mira simulated file write protocol



Results

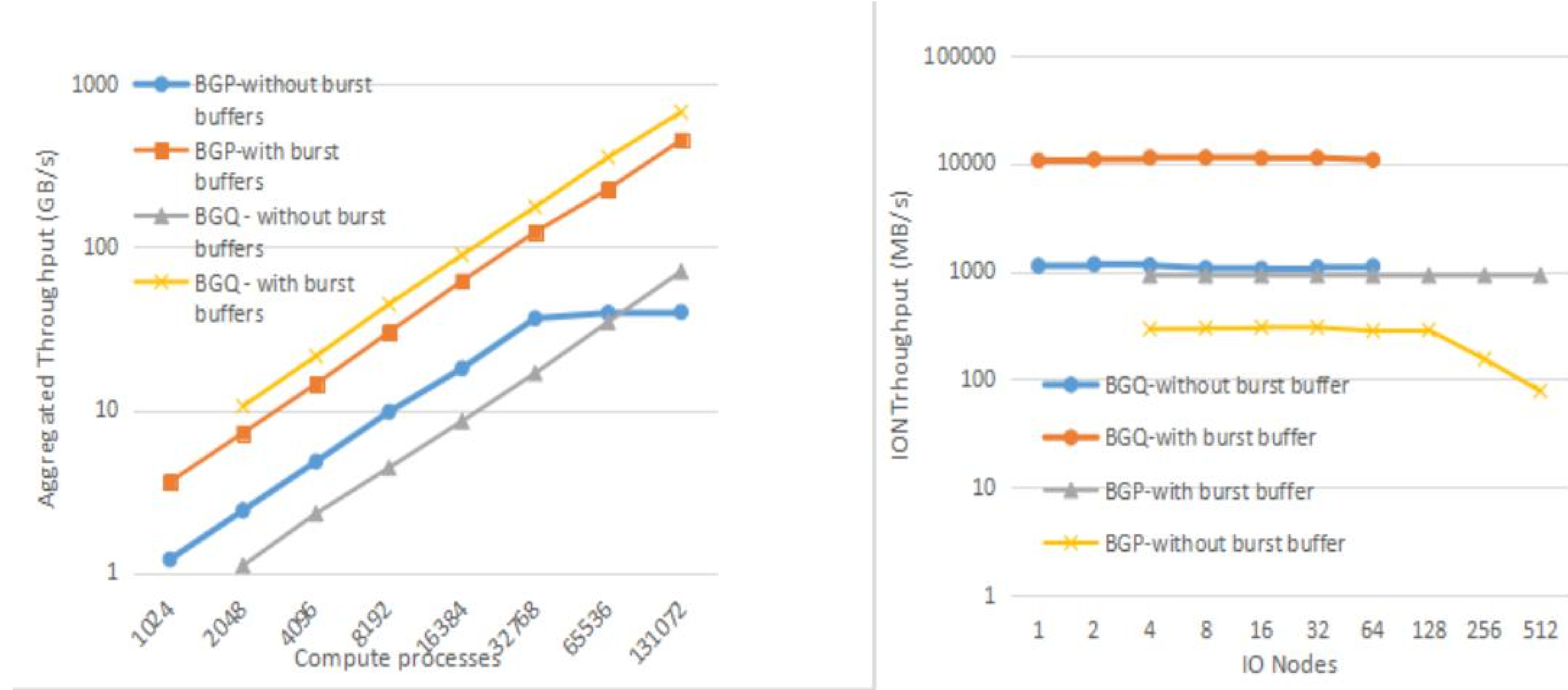
We conducted several sets of experiments with configurations of Intrepid and Mira. The largest scale is up to 131,072 compute processes. Figure 3 (left) shows the aggregated write throughput with burst buffer enabled or not. Figure 3 (right) shows the per IO node throughput with burst buffer enabled or not.

Conclusion

From current results, we can conclude that:

- At the scale we run our experiments (upto 131,072 cores), burst buffers can scale very well in current Blue Gene/P and Blue Gene/Q storage architecture.
- The interconnection network between IO nodes and burst buffer nodes will be the bottleneck of full utilization of buffer buffers.
- bringing storage closer to the compute nodes can have a significant positive impact on performance

Figure 3 Simulation results



Related work

- Burst Buffer simulation on Intrepid
- Parallel File System simulators
- CODES network modeling framework
- ROSS massive parallel discrete-event simulator

Future work

Simulating supercomputers with dragon-fly network topologies, to study the effectiveness of the burst buffer architecture. We also plan to explore real application IO traces. Furthermore, we will also study burst buffer aware job scheduling, where data movement between burst buffers and persistent storage, and among the burst buffer resources, can be minimized

Acknowledgments

We would like to thank Jonathan Jenkins at ANL for his help in CODES. We also would like to thank Ning Liu for his experience from previous work.

References

[1] Liu, N., Carothers, C., Cope, J., Carns, P., Ross, R., Crume, A., & Maltzahn, C. (2011, September). Modeling a leadership-scale storage system. InProceedings of the 9th international conference on Parallel Processing and Applied Mathematics-Volume Part I (pp. 10-19).
[2] Liu, Ning, et al. "On the role of burst buffers in leadership-class storage systems." Mass Storage Systems and Technologies (MSST), 2012 IEEE 28th Symposium on. IEEE, 2012.
[3] Cope, Jason, et al. "Codes: Enabling co-design of multilayer exascale storage architectures." Proceedings of the Workshop on Emerging Supercomputing Technologies. Vol. 2011. 2011.
[4] Carothers, Christopher D., David Bauer, and Shawn Pearce. "ROSS: A high-performance, low-memory, modular Time Warp system." Journal of Parallel and Distributed Computing 62.11 (2002): 1648-1669.
[5] Jefferson, David, et al. Time warp operating system. Vol. 21. No. 5. ACM, 1987.
[6] Chen, Dong, et al. "The IBM Blue Gene/Q interconnection network and message unit." 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC). IEEE, 2011.
[7] Alex Szalay, Julian Bunn, Jim Gray, Ian Foster, Ioan Raicu. "The Importance of Data Locality in Distributed Computing Applications", NSF Workflow Workshop 2006
[8] Dongfang Zhao, Ioan Raicu. "Distributed File Systems for Exascale Computing, Doctoral Showcase, IEEE/ACM Supercomputing/SC 2012

Contact Information:

Jian Peng jpeng10@hawk.iit.edu | Sughosh Divanji sdivanji@hawk.iit.edu | Ioan Raicu iraicu@cs.iit.edu | Michael Lang mlang@lanl.gov