POSTER: Leveraging Deep Memory Hierarchies for Data Staging in Coupled Data-Intensive Simulation Workflows

Tong Jin*, Fan Zhang*, Qian Sun*, Hoang Bui*, Norbert Podhorszki†, Scott Klasky†,
Hemanth Kolla‡, Jacqueline Chen‡, Robert Hager§, Choong-Seock Chang§, Manish Parashar*
*NSF Cloud and Autonomic Computing Center, Rutgers University, Piscataway, NJ 08854, USA
†Oak Ridge National Labortory, P.O. Box 2008, Oak Ridge, TN, 37831, USA
‡Sandia National Labortory, Livermore, CA 94550, USA
§Princeton Plasma Physics Laboratory, Princeton, NJ 08543, USA

***Abstract*—Next generation in-situ/in-transit data processing has been proposed for addressing data challenges at extreme scales. However, further research is necessary in order to understand how growing data sizes from data intensive simulations coupled with limited DRAM capacity in High End Computing clusters will impact the effectiveness of this approach. In this work, we propose using deep memory levels for data staging, utilizing a multi-tiered data staging method with both DRAM and solid state disk (SSD). This approach allows us to support both code coupling and data management for data intensive simulations in cluster environment. We also show how an application-aware data placement mechanism can dynamically manage and optimize data placement across DRAM and SSD storage levels in staging method. We present experimental results on Sith - an Infiniband cluster at Oak Ridge, and evaluate its performance using combustion (S3D) and fusion (XGC) simulations.**

I. Introduction

Scientific simulation workflows running at scale on High End Computing platforms can provide dramatic insights into natural and engineering systems. These simulations however can also generate very large amounts of data, which must be processed and analyzed before new insights from the simulations can be realized. In-situ/in-transit data processing using in-memory data-staging approaches [1] have been proposed to address challenges due to increasing data sizes and related costs. However, architectural trends indicate that emerging systems will have increasing numbers of cores per node, and a correspondingly decreasing amounts of DRAM memory per core as well as memory bandwidth. These trends can significantly impact the effectiveness of in-memory staging solutions and their ability to enable data-intensive simulation workflows.

Fortunately, non-volatile memory devices such as solid state devices (SSD) are becoming more pervasive and can may help address these challenges. These technologies offer several benefits over magnetic hard disk due to lower data access latency, lower power consumption and stability. In this poster, we explore how a SSD-based deep memory hierarchy can be used for data staging to address the challenges outlined above. Specifically, we present the design of a multi-tiered data staging runtime that leverages both DRAM and SSD to support dynamic data staging for coupled data-intensive simulation workflows. The hybrid staging approach allows us to accommodate larger data volumes, which may exceed available DRAM main memory within the staging area, and to support runtime data sharing, data coupling and data management required by simulation workflows. We also introduce a credit-based application-aware data placement mechanism that can dynamically manage and optimize data placement across the DRAM and SSD storage levels using data locality and user provided hints about data access patterns. We implemented and evaluated our multi-tiered data-staging approach on Sith Infiniband cluster at Oak Ridge and we summarize results in this poster.

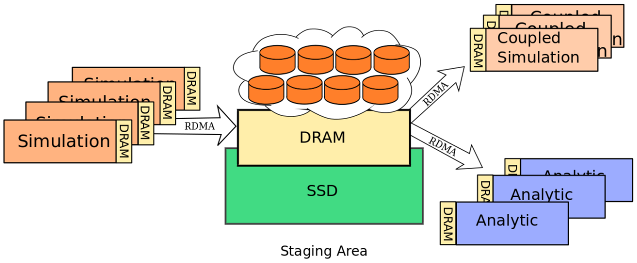

Fig. 1: Conceptual overview of a framework for multi-tiered staging to support coupled data-intensive simulation workflows.

II. Methodology of Multi-tiered Staging

A. *Multi-tiered Staging Framework*

This research focuses on enabling data intensive coupled simulation workflows, such as those illustrated in Figure 1. A typical workflow consists of the coupled scientific simulations and one or more analytics components. These components exchange data at runtime at different scales, produce and consume data at different rates, and progress at different speeds. Orchestrating these workflows and efficient managing the data exchanges between the components is a key challenge, and in-memory data-staging based solutions have been effectively used to address these challenges [1]. However as data volumes increase and data no longer fits into main memory on the staging nodes these solutions can break down. The multi-tiered staging framework presented in this poster extends staging to deeper memory hierarchies by utilizing two types of local memory storage resources - DRAM and SSD.

978-1-4799-5548-0/14/$31.00 ©2014 IEEE

Case #	# of readers	Data domain	Read frequency	Analytics techniques
1	One	Whole	Every single time step	Visualization [2]
2	One	Whole	Twenty consecutive time steps out of thirty time steps	Feature tracking [3], data trajectories visualization [4]
3	One	Partial	Every ten time steps	Interactive visualization, descriptive statistic [5]
4	Two	Whole	Every ten time steps for one, five time steps for the other	Visualization and topology analysis simultaneously [5]

TABLE I: A summary of the characteristic of the four test cases.

B. *Application-aware data placement mechanism*

Our multi-tiered data staging framework uses an application-aware credit-based data placement mechanism that leverages user-provided hints regarding data access patterns, and uses these hints along with data locality to dynamically manage the placement of data within the staging area. The goal is to optimize placement of data objects across DRAM and SSD storage levels by leveraging knowledge of access patterns and prefetching data required by read requests. At the end of each time step, the progress of the simulation is not impacted by where the data is placed within the staging area. As a result, our data placement mechanism primarily focuses on data movement across the levels of the memory hierarchy on the staging nodes in order to optimize data retrieval requests. Furthermore, it ensures that data is moved to DRAM prior to a read request in order to avoid the penalty of reading data from a lower level of the memory hierarchy, i.e., the SSD level, and improving the overall performance of data retrieval.

III. Experimental Evaluation

A. *Impact of Application-aware Data Placement*

Our experiments evaluated the data reading performance of our multi-tiered staging approach with the application-aware data placement mechanism. We used four test cases with typical data reading patterns derived from real scientific simulation workflows, as listed in Table I. In our experiments, two synthetic application codes were used to write datasets of different sizes of data into staging area, which was then read by coupled components, to emulate the end-to-end data movement behavior in a real coupled simulation workflow. We compared our approach with three other data placement mechanisms, i.e., DRAM only placement (place data in DRAM all the time), SSD only placement (place data in SSD all the time, no data prefetching), and conventional data locality based placement (predict and prefetch data from SSD to DRAM based on data temporal and spatial localities without hints). Our experiments demonstrated that our multi-tiered staging approach with application-aware credit-based data placement can effectively support evaluated data reading patterns providing improved data reading performance. Specifically, our approach provided better performance for (1) low frequency read to small data sets, and (2) cases with multiple readers.

B. *Performance Evaluation with Application Drivers*

To validate the performance of the multi-tiered staging approach for real applications, we integrated our framework with two applications with different coupling behaviors: (1) a *tightly coupled combustion DNS-LES simulation workflow* [6] with high frequency one-way end-to-end data exchange, and (2) a *loosely coupled plasma fusion XGC1-XGCa simulation workflow* [7], [8] with multiple coupling steps and bidirectional data exchange. For comparison purposes, we evaluated BP-file [9] based staging, in-memory staging and our multi-tiered staging to temporarily cache the data for these scenarios, and measured the cumulative time of reading data for different data and systems scales. The experiments demonstrated that our approach improves read performance by 10% to 31.5% over that of file-based staging. Furthermore, as the number of cores increased, the performance of multi-tiered staging approached that of in-memory staging.

IV. Conclusion

This poster presented the design of a multi-tiered data staging framework that leverages multiple levels of the memory hierarchy for data staging to support data intensive coupled simulation workflows. Our framework utilizes both DRAM and SSD storage to implement a virtual shared-space abstraction to support coupling and data exchange. Furthermore, it leverages user-provided hints about data access pattern to improve data placement. Our experiments demonstrates that our framework can dynamically manage and optimize data placement across deep memory hierarchy levels to support coupled data intensive simulation workflows in an efficient and scalable manner.

References

[1] C. Docan, M. Parashar, and S. Klasky, “Dataspaces: an interaction and coordination framework for coupled simulation workflows,” *Cluster Computing*, vol. 15, no. 2, pp. 163–181, 2012.

[2] K.-L. Ma, “In Situ Visualization at Extreme Scale: Challenges and Opportunities,” *IEEE Computer Graphics and Applications*, vol. 29, no. 6, pp. 14–19, 2009.

[3] T. Jin, F. Zhang, M. Parashar, S. Klasky, N. Podhorszki, and H. Abbasi, “A scalable messaging system for accelerating discovery from large scale scientific simulations,” in *Proc. IEEE International Conference on High Performance Computing (HiPC)*, December 2012.

[4] J. Wei, H. Yu, R. Grout, J. Chen, and K.-L. Ma, “Dual space analysis of turbulent combustion particle data,” in *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, 2011, pp. 91–98.

[5] J. C. Bennett, H. Abbasi, P.-T. Bremer, R. W. Grout, A. Gyulassy, T. Jin, S. Klasky, H. Kolla, M. Parashar, V. Pascucci, P. Pbay, D. Thompson, H. Yu, F. Zhang, and J. Chen, “Combining in-situ and in-transit processing to enable extreme-scale scientific analysis,” in *Proceedings of IEEE/ACM Supercomputing Conference (SC)*, November 2012.

[6] J. H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E. R. Hawkes, S. Klasky, W. K. Liao, K. L. Ma, J. Mellor-Crummey, N. Podhorski, R. Sankaran, S. Shende, and C. S. Yoo, “Terascale direct numerical simulations of turbulent combustion using S3D,” *Comp. Sci. Disc.*, vol. 2, pp. 1–31, 2009.

[7] E. F. D’Azevedo, J. Lang, P. H. Worley, S. A. Ethier, S.-H. Ku, and C. Chang, “Hybrid mpi/openmp/gpu parallelization of xgc1 fusion simulation code,” in *Supercomputing Conference 2013*, 2013.

[8] S. Ku, C. Chang, and P. Diamond, “Full-f gyrokinetic particle simulation of centrally heated global itg turbulence from magnetic axis to edge pedestal top in a realistic tokamak geometry,” *Nuclear Fusion*, vol. 49, no. 11, p. 115021, 2009.

[9] Q. Liu, J. Logan, Y. Tian, H. Abbasi, N. Podhorszki, J. Y. Choi, S. Klasky, R. Tchoua, J. Lofstead, R. Oldfield, M. Parashar, N. Samatova, K. Schwan, A. Shoshani, M. Wolf, K. Wu, and W. Yu, “Hello adios: the challenges and lessons of developing leadership class i/o frameworks,” *Concurrency and Computation: Practice and Experience*, pp. n/a–n/a, 2013. [Online]. Available: http://dx.doi.org/10.1002/cpe.3125