# Enabling Asynchronous Coupled Data Intensive Analysis Workflows on GPU-accelerated Platforms via Data Staging

Daihou Wang (student author)
Rutgers Discovery Informatics Institute
Rutgers University
Piscataway, NJ 08854

David J. Foran
Xin Qi
Rutgers Cancer Institute of New Jersey
New Brunswick, NJ 08903

Manish Parashar
Rutgers Discovery Informatics Institute
Rutgers University
Piscataway, NJ 08854

## ABSTRACT

Enabled by the advanced network techniques as Infiniband and RDMA, data staging and in-situ/in-transit techniques are emerging as an attractive approach for large scale data intensive workflows. At the same time, accelerator based heterogeneous platforms are widely adopted for data-intensive analysis applications due to their superior computation and power performances. However, the complex memory hierarchies and programming directives prevent former data-staging libraries and middlewares from performing efficiently on accelerator-based platforms.

Here, we present our work enabling asynchronous coupled data-intensive analysis workflow through the combination of *CUDA-aware DataSpaces (cwDS)*, an asynchronous data management library for large scale data insensitive applications on heterogeneous CPU/GPU platforms, and *ExtendedAcc (ExtAcc)*, an event-based asynchronous runtime for coupled in-situ analysis workflows on GPU-clusters.

We demonstrate the performance evaluation of the combined framework on two histopathological image analysis workflows on Stampede platform at TACC, and Comet at SDSC.

## KEYWORDS

data staging, in-situ analysis, asynchronous execuation, GPU cluster

**ACM Reference format:**
Daihou Wang (student author), David J. Foran, Xin Qi, and Manish Parashar. 2016. Enabling Asynchronous Coupled Data Intensive Analysis Workflows on GPU-accelerated Platforms via Data Staging. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 2 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

Data staging, in-situ/in-transit techniques have been widely adopted into extreme-scale scientific computation and simulation workflows to manage the large volumes of data produced. Data and partial results are transfered off the computation nodes to in-memory staging area, for efficient communication between coupled workflows running on analysis nodes.

To enable efficient data tranfer, retrieval and sharing between different stages of the same of application and between different applications, *data staging* technics were proposed. Data are transfered off the computation nodes to seperated nodes called *staging area*, where they were further analized or shared between different workflows. Projects like DataSpaces [6] and DataStager [1] have provided solutions for efficient data staging and sharing within/between scientific computation workflows, by enabling asynchronous data transfer and efficient indexing for data query and retrieval.

After *data staging*, post-processing were conducted on the simulation/computation results in the staging area. *In-situ*, *in-transit* and hybird *in-situ in-transit* processing frameworks were developed for post-processing on the data either on the staging nodes (*in-situ*), or offloaded to secondanry resources (*in-transit*). *In-situ* processing were proposed to optimize the post-processing efficiency by minimizing the data relocation, and mappping the post-processing task to the corresponding staging area.

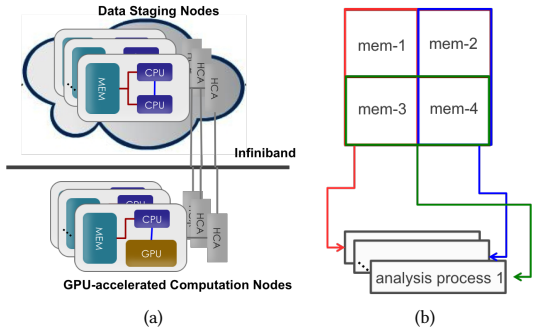


**Figure 1: Illustration of data staging based coupled analysis workflows on GPU-accelerated platforms.**

Though previous proposed frameworks [2, 5, 7] provided efficient post-processing, as the data scale and algorithm develops, we're facing a new challenge that traditional CPU-based post-processing frameworks cannot provide the expected throughput for extreme-scale applications. Meanwhile, as the great computation performance and power cost, acclerators represented by GPGPUs are becoming the major supporting computing units in both supercomputers and commercial computing, and has been used for more and more popular for data-intensive scentific analysis applications. However the complexed memory hierarchies, and programming models create challenges for efficiently deploying staging enabled coupled analysis workflows on these platforms. In this study we proposed to use accelerator-based platform as the secondary resources for post-processing (illustrated in Figure 1), via the combined *cwDS* and *ExtendedAcc* framework.

*Conference'17, Washington, DC, USA*
2017. 978-x-xxxx-xxxx-x/YY/MM…$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

# 2 METHODOLOGY

## 2.1 CUDA-aware DataSpaces

In CUDA-aware DataSpaces, we design PCIe based and GPUDirect RDMA based protocals to accomandate data transfer on/off the seperated GPU memory. Pipelined PCIe protocal is designed to maximize the CPU-GPU data transfer rate with host CPU involved. With GPUDirect RDMA (GDR) supported, GDR protocal enables data transfering without host GPU involvment. To accommodate various data access patterns, CUDA-aware DataSpaces introduce non-contiguous data sharing with strided data type. *cwDS strided data type* enables non-contiguous data sharing with user defined strided data patterns. To further support coupled data-intensive workflows on GPU-clusters, in *cwDS* we introduce *volatile cwDS variables* to enable dynamic *cwDS* variable allocation and release on the staging area.

Later, we analysis and modeled the computation and power performance of the various data placement strategies adopted in the DataSpaces framework, on heterogeneous CPU-GPU platforms. And proposed a performance optimized data placement scheme. Later we combined the data access pattern of analysis kernels with this optimized data placement scheme, and generates power-performance optimized data placements for coupled analysis kernels.

## 2.2 ExtendedAcc

In addition to the *cwDS* library, here, we also present the runtime layer of the combined framework, ExtendedAcc (ExtAcc), an event-based asynchronous runtime for coupled in-situ analysis workflows on GPU-clusters.

ExtAcc supports asynchronously data transfer and task execuation using light-weigth events. Working together, the ExtAcc runtime provide a hierarchical support for: (1) unified data representation based programming model, ExtAcc provide the higher level of interface of data/variables in the staging area; (2) latency hiding, ExtAcc analysis the data and control dependency of the post processing kernels, and optimizedly schedules the data sharing and data updating to hide the latency by overlapping the data updating to the staging area with computation on the post processing nodes.

# 3 EVALUATION

To evaluate the performance of the proposed combined *cwDS* and *ExtendedAcc* framework, we detailed a case study on a coupled data-intensive computer-aided diagnosis (CAD) workflow of histopathological WSI images. This CAD analysis workflow is part of the prostate carcinoma histopathology image automatic grading study [3, 4] at Rutgers Cancer Institute of New Jersey.

As WSI imaging include multiple layers of high resolution image into one single file, WSI images generated for current diagnosis and research usage range from 1 to 8 GB per image, which made the WSI-based analysis workflows unable to fit in memory for single node. To avoid the intensive disk I/O, we adopt data staging into the workflow to enable efficient data analysis and sharing within different stages of an analysis workflow or between different anlysis workflows.

To evaluate the proposed combined frame, we compare the performance of 2 CAD workflows: LSH based sub-image retrieval and local feature based automatic Gleason grading, with their PFS based staging implementations. We compare the experiment results using 64,128,256 data staging cores, with testing WSI data set upto 100GB on Stampede at TACC and Comet at SDSC. Results showed that: (1) The proposed *cwDS* library speed up to 23.2% and 18.9% for the LSH-based retrieval and automatic Gleason grading workflow, comparing to their disk-based staging implementations. (2) With the additional wraping scripts, the proposed *ExtendedAcc* runtime further accelerate the testing workflows for 5.4% and 3.7%, comparing to their naive *cwDS* based implementations.

# 4 CONCLUSION

To overcome the I/O cost in disk based data staging for coupled in-situ data analysis workflows, in this paper, we present our design and evaluation of an combined *cwDS* and *ExtendedAcc* framework for coupled data intensive analysis workflows on GPU-cluster.

By introducing CUDA awareness to the DataSpaces library, the *CUDA-aware DataSpaces* library enables direct data staging and updating from GPU memory to the data staging area. With kernel-driven *cwDS*, the new library can optimize over the data access granduality, data placement schemes for the best computation and power performance. To further enabling the asynchronous execuation, we proposed *ExtendedAcc* runtime. With light-weight events, *ExtAcc* supports the unified data representation, as well as latency hiding for data staging and sharing. Experiment results of 2 CAD analysis workflows showed the proposed framework achieves both excellent performance and scalability.

# REFERENCES


[1] H. Abbasi, M. Wolf, G. Eisenhauer, S. Klasky, K. Schwan, and F. Zheng. 2009. DataStager: Scalable Data Staging Services for Petascale Applications. In *Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing*. 39–48.

[2] J. C. Bennett, H. Abbasi, P. T. Bremer, R. Grout, A. Gyulassy, T. Jin, S. Klasky, H. Kolla, M. Parashar, V. Pascucci, P. Pebay, D. Thompson, H. Yu, F. Zhang, and J. Chen. 2012. Combining In-situ and In-transit Processing to Enable Extreme-Scale Scientific Analysis. In *Proceeding of IEEE 26th International Parallel and Distributed Processing Symposium (SC'12)*. 1–9.

[3] J. Diaz-Montes I. Rodero L. Yang M. Parashar D. J. Foran D. Wang, M. Diaz-Granados and X. Qi. 2014. High-throughput automatic Gleason-grading system for prostate histopathology images using CometCloud. In *Proceeding of High Performance Computing in Biomedical Image Analysis Associated with MICCAI (MICCAI-HPC '14)*. 21–30.

[4] J. Ren-H. Zhong I. Kim D. Wang, D. J. Foran and X. Qi. 2015. Exploring automatic prostate histopathology image Gleason grading via local structure modeling. In *Proceeding of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 15fi)*. 2649–2652.

[5] C. Docan, M. Parashar, J. Cummings, and S. Klasky. 2011. Moving the Code to the Data, Dynamic Code Deployment using ActiveSpaces. In *Proceeding of 25th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2011)*. 758–769.

[6] C. Docan, M. Parashar, and S. Klasky. 2010. DataSpaces: an interaction and coordination framework for coupled simulation workflows. In *Proceeding of 19th International Symposium on High Performance and Distributed Computing (HPDC 2010)*. 25–36.

[7] F. Zhang, C. Docan, M. Parashar, S. Klasky, N. Podhorszki, and H. Abbasi. 2012. Enabling In-situ Execution of Coupled Scientific Workflow on Multi-core Platform. In *Proceeding of IEEE 26th International Parallel and Distributed Processing Symposium (IPDPS 2012)*. 1352–1363.