Discovery Engines: Connecting X-ray Experiments to HPC Analysis

Justin M. Wozniak,[*][†] Hemant Sharma,[*] Michael Wilde,[*][†] Ian T. Foster[*][†][‡]

[*]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA
[†]Computation Institute, University of Chicago and Argonne National Laboratory, Chicago, IL, USA
[‡]Dept. of Computer Science, University of Chicago, Chicago, IL, USA

I. OVERVIEW

High-energy diffraction microscopy (HEDM) is an important method for determining the grain structure of metals. It is performed at specialized light sources such as the Advanced Photon Source (APS) at Argonne National Laboratory (ANL). In the typical scientific workflow, the scientist applies for a week of beam time and spends that time gathering data. Over the next several months, the data is processed using custom-built tools. High-performance computing (HPC) is rarely if ever applied during the week of beam time. Applying HPC to quickly analyze the data produced by the experiment detectors has the potential to greatly reduce errors. It may provide feedback to the scientist during the run, to improve the quality of results and the utility of precious beam time. Most importantly, it will speed the process of scientific discovery, even if the HPC is applied after beam time. In this work, we present an HEDM application that spans a small cluster and the Blue Gene/Q at ANL. It uses up to 64K cores of the BG/Q to provide near real-time feedback to APS beam users.

II. HIGH-ENERGY DIFFRACTION MICROSCOPY AT THE APS

APS beamline scientists H. Sharma and J. Almer, performing experiments at APS Sector 1-ID, are developing new experimental techniques and algorithms for determining the grain structure of polycrystalline material samples, via high-energy diffraction microscopy (HEDM). In close collaboration with DEBD, they are applying new tools and methods to parallelize (and dramatically speed up) their new algorithms; enhance them with data and metadata management techniques; and make them available to the broader APS and scattering communities.

A. Scientific overview

HEDM is used to non-destructively determine the grain structure of polycrystalline materials. It's a valuable technique for the analysis of grain defects in advanced alloy materials, such as those used in turbine blades for both energy (e.g. wind turbines) and engine applications (e.g. jet engine turbines). The technique allows a material sample, or even a manufactured part, to be studied within an APS beamline across a dynamically varying range of concurrent thermal and tensile stresses. A polycrystalline material sample (typically a metal alloy) is positioned in the X-ray beam, and the diffraction pattern of the beam as it passes through the sample is collected by a detector.

The diffraction patterns are then analyzed by software tools to reconstruct the granular structure of the material, with a focus on determining the granular structure of material defects that can cause component failure in fabricated parts.

A line-focus X-ray beam is used to collect diffraction data from a 2D cross-section ("layer") of a rotating polycrystalline sample using a high-resolution detector (1.5 micron pixel size) placed in close proximity to the sample (less than 10 mm away). 2D TIFF format images, 8MB in size, are collected from the detector. At each angle of rotation, typically 360 to 1440 angles per layer, 8MB per file, are stored on the s1idserv file server. A new layer (3 to 12 GB of raw data) is collected every 3-5 minutes. As in the case of FF-HEDM, the acquisition is performed using a set of "SPEC" scripts. The metadata of each layer is stored in a separate file.

B. Workflow details

The raw TIFF files are reduced to binary files containing only information about the diffraction signal of the detector. Due to the sparse nature of the data, each 8 MB raw file can be reduced to a binary file of about 1 MB. This reduction is performed using 320 CPUs on Orthros, each using about 2 min of CPU time. These binary files are fed into an orientation detection program to map the orientation of each point in a grid in the 2D cross-section of the sample. Two approaches are possible for this step. Using 320 CPUs on Orthros, at about 30s for each grid point (a total of 100k grid points), we can obtain results in 3 hours. Using 10,000 CPUs on the Mira BG/Q or a similarly parallel resource, we can scale this workflow to obtain results in less than 5 min.

The output is a text or binary file (selectable by the user), about 10 MB in size, containing information about the orientation of each point in a grid in the 2D cross-section of the sample. All analysis software has been developed in-house in Sector 1, implemented in C. Parallel processing of the analysis routines is implemented using the Swift parallel scripting language in collaboration with the DEBD project.

A detailed diagram of the workflow for NF-HEDM is shown in Figure 1. In this figure, the detector produces data on an NFS installation at the APS (1). A large batch of data reduction jobs are run on the local cluster, named Orthros (2). The data is moved via Globus to Argonne Leadership Computing Facility (ALCF) storage (3), and stored in a metadata catalog (4) [4]. Finally, the HPC component begins (5)- a large
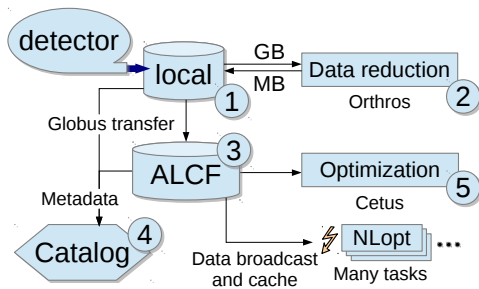
Fig. 1: Cross-lab APS to ALCF workflow.

batch of hundreds of thousands of optimization operations are performed rapidly across tens of thousands of CPU cores of a Blue Gene/Q.

III. MANY-TASK PARALLELIZATION FOR HEDM

The HPC component of the workflow was implemented using Swift/T [6], a dataflow programming system implemented atop the Turbine runtime [5] and the Asynchronous Dynamic Load Balancer [3]. In this model, C analysis code developed for this work is linked to the NLopt optimizer library and the GNU scientific library. These C code tasks are grouped into a large Swift script, which compiles into an MPI program for execution on a large HPC resource. Swift/T has been shown to scale to over a billion tasks/second on hundreds of thousands of cores [1] or GPUs [2].

REFERENCES

[1] T. G. Armstrong, J. M. Wozniak, M. Wilde, and I. T. Foster. Compiler techniques for massively scalable implicit task parallelism. In *Proc. SC*, 2014.

[2] S. J. Krieder, J. M. Wozniak, T. G. Armstrong, M. Wilde, D. S. Katz, B. Grimmer, I. T. Foster, and I. Raicu. Design and evaluation of the GeMTC framework for GPU-enabled many task computing. In *Proc. HPDC*, 2014.

[3] E. L. Lusk, S. C. Pieper, and R. M. Butler. More scalability, less pain: A simple programming model and its implementation for extreme computing. *SciDAC Review*, 17, 2010.

[4] T. Malik, K. Chard, and I. Foster. Benchmarking cloud-based tagging services. In *Proc. CloudDB*, 2014.

[5] J. M. Wozniak, T. G. Armstrong, K. Maheshwari, E. L. Lusk, D. S. Katz, M. Wilde, and I. T. Foster. Turbine: A distributed-memory dataflow engine for high performance many-task applications. *Fundamenta Informaticae*, 28(3), 2013.

[6] J. M. Wozniak, T. G. Armstrong, M. Wilde, D. S. Katz, E. Lusk, and I. T. Foster. Swift/T: Large-scale application composition via distributed-memory data flow processing. In *Proc. CCGrid*, 2013.