Experiences in Building a Data Packaging Pipeline for Tomography Beamline

Lavanya Ramakrishnan Richard S. Canon
Lawrence Berekely National Lab
Berkeley, CA 94720
Email:{lramakrishnan,scanon}@lbl.gov

***Abstract*—Light source and neutron source facilities have seen a tremendous increase in data volumes and data rates due to recent improvements in detector resolution. Facilities such as the Advanced Light Source (ALS) are expecting to produce a couple of petabytes per year within the next few years. As a result, we are rapidly exceeding the capacity of the resources and tools available to users currently. Thus, there is a need for new computing, storage and analyses capabilities for supporting these sciences such as those provided at High Performance Computing (HPC) centers. Beamline 8.3.2 at the Advanced Light Source is a Synchrotron-based Hard X-ray Micro-Tomography instrument that allows non-destructive 3-Dimensional imaging of solid objects. In this paper, we outline a system architecture for end-to-end data packaging movement and management for data from the ALS Tomography beamline. The data and metadata are stored and available for processing at the National Energy Research Computing Center (NERSC). We detail our experiences, challenges and perspective on designing similar data-intensive pipelines.**

I. Introduction

Light source and neutron source facilities have seen unprecedented data rates due to recent improvements in detector resolution and speed and luminosity. For example, the Advanced Light Source (ALS) has seen their data volumes grow from 65 TB/year to 312 TB/year and are expected to be producing a couple of petabytes per year within the next few years. The data volume exceeds the capabilities of end workstations that are often attached to the beamlines and used for analyses. The problem is only expected to get worse as detector technologies improve. The need for better computational, storage and analyses capabilities have also been recognized in the larger context of other Basic Energy Science facilities [1].

Beamline 8.3.2 at the Advanced Light Source is a Synchrotron-based Hard X-ray Micro-Tomography instrument. The beamline allows non-destructive 3-Dimensional imaging of solid objects by collecting X-ray images of a sample as it is rotated through 180 degrees. The beamline's users are varied and it has been used to study water transport in plants, time-resolve studies for geologic carbon sequestration, dendrite growth in batteries, brittle bone disease and materials failures at high temperatures [2]. The beamline is representative of the data challenges faced by other science groups [3]. The beamline has to handle large volumes and velocity of data due to higher resolutions and fast scans now possible in the detectors. In this environment, the sample changing and setup time dominates the experiment time. The beamline has previously handled the large data volumes by adding local hard drives. The beamline operators have also spent multiple hours per week copying and deleting data manually. The volume of the data is exceeding the capacity of the processing power and tools available on the workstation attached to the beamline. Thus, data processing and down-stream analyses, visualization and simulation are bottle-necked and heavily impacted by the data capture process and the resources available at the experiment facility.

Data-intensive scientific applications need for large-scale storage and compute resources. Traditionally, supercomputing centers have been used for tightly coupled applications. However, scientific computing facilities are increasingly being used for data-intensive sciences [4]. However, there is still a gap in management of these frameworks in supercomputing facilities. There is a need for an end-to-end solution for data packaging, movement and data management for effective and efficient downstream processing.

In this paper, we detail a system architecture for end-to-end data packaging, movement and data access. We describe the challenges and experiences while managing the data for the ALS beamline 8.3.2 at the National Energy Research Scientific Computing Center (NERSC). Specifically, our contributions are:

- We detail a system architecture for data packaging and movement from the beamline to NERSC.
- We detail our system design for metadata and data management and data access at NERSC
- We detail our experiences and show the usage over a period of few months.

The rest of this paper is organized as follows. In Section II we detail the system architecture and our experiences in Section III. We present a discussion on pipelines for future data-intensive applications and our conclusions in Sections III and V respectively.

II. System Architecture

Figure 1 shows the system architecture of the end-to-end data packaging, movement and management pipeline. The data packaging process starts at the Advanced Light Source (ALS) when new data arrived from the detector. The data conversion and packaging program is launched for each arriving data set. The data is packaged and converted to an HDF5 file. HDF5 is

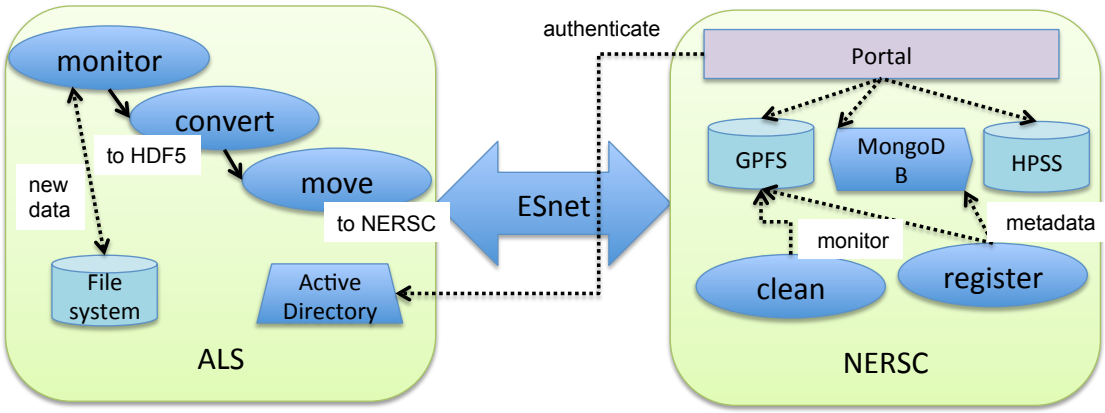

Fig. 1. The figure shows the system architecture at the Advanced Light Source (ALS) and National Energy Research Scientific Computing Center (NERSC). We monitor, convert and move the data on the ALS side. The data is registered and available for download on the NERSC side. Users can access the data through a portal interface. Data is cleaned out from the file system periodically.

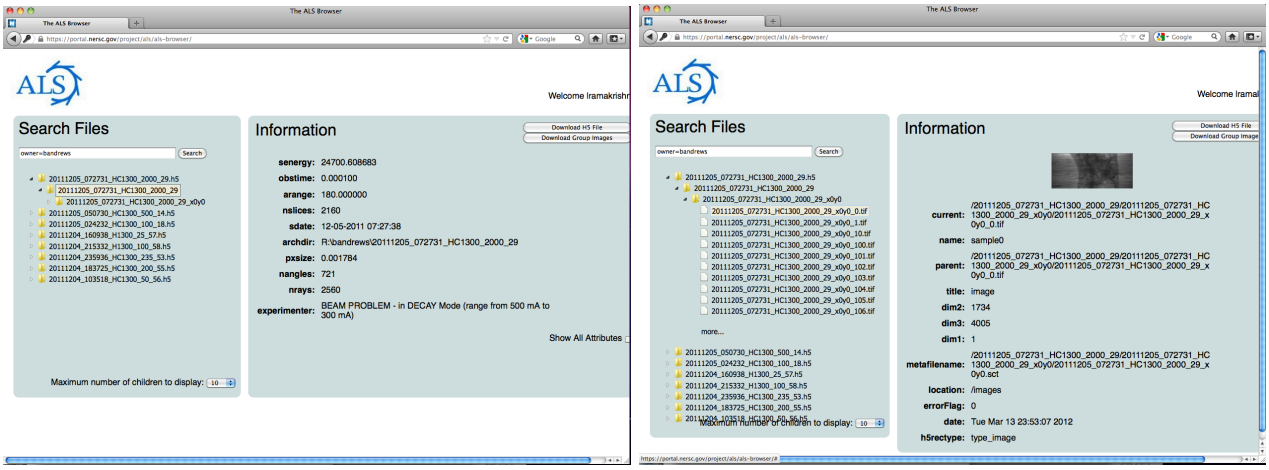

Fig. 2. The figures show the ALS user's data workspace. The search bar allows the user to select and search for particular files. The list of data sets are available for the user's to browse and more details including a thumbnail of the image and metadata can be viewed on the main screen.

a portable and extensible data model and file format. HDF5 is extensively used for scientific data since it provides efficient parallel I/O.

Next, the HDF5 file is moved using Globus Online over the Energy Sciences Network (ESnet). ESnet provides high-speed networking to Department of Energy (DOE) scientists and their collaborators worldwide. Currently, the original data set and the HDF5 data set are moved to NERSC since the HDF5 format has only recently been adopted by this user group. Moving the original and packaged data allows us to build confidence in the new packaging process and gives the users time to adapt their tool chain to the new format.

The data arrives at the data transfer node at the National Energy Research Scientific Computing Center (NERSC). NERSC is a leading scientific computing facility supporting research within the Department of Energy's Office of Science. NERSC provides high-performance computing (HPC) resources to approximately 4,000 researchers working on about 400 projects.

When the data arrives at NERSC, the metadata is extracted and the metadata and the data are registered in a MongoDB database. The users use a web front end to download the data sets. The user is able to view the metadata and thumbnails of the images before selecting the data set to download. The HDF5 file or the raw files can be downloaded from the web front end.

A copy of the data is also archived in HPSS (High Performance Storage Systems) at NERSC for long-term access. The files on the file system are cleaned out periodically.

A. *Data Packaging*

The data packaging process is initiated on the data transfer node attached to the beamline. A process is continuously monitoring the directory in which the data from the beamline arrives. We know a given "experiment" data set is ready to be packaged when a *done.txt* file is written out in the directory. An experiment source directory might have a set of images or a set of tile directories that have images. Application metadata is stored in each directory and/or sub-directory as a text file by the control program that writes the experiment data to disk.

The data packaging process reads and packages the data and metadata into an HDF5 file. The data packaging uses PyTables, a Python library that is built on top of the HDF5 library and uses the NumPy package. Table II-A shows the structure

```
GROUP ``/'' {
   ATTRIBUTE ``errorFlag'' {
      DATATYPE  H5T_STD_I64LE
      DATASPACE  SCALAR
      DATA {
      (0): 0
      }
   }
   ATTRIBUTE ``exptime'' { ... }
   ATTRIBUTE ``owner'' { ... }
   ATTRIBUTE ``sourcefile'' { ... }
   ATTRIBUTE ``sourcehost'' { ... }
   GROUP ``set1'' {

      ATTRIBUTE ``senergy'' { ... }
      ....
      DATASET ``image0.tif'' {
         DATATYPE  H5T_STD_U16LE
         DATASPACE  SIMPLE { ...}
         ATTRIBUTE ``date'' { ... }
         ATTRIBUTE ``dim1'' { ... }
         ATTRIBUTE ``dim2'' { ... }
         ATTRIBUTE ``dim3'' { ... }
      }
      DATASET "image1.tif'' {

        ...
      }
```

TABLE I. STRUCTURE OF THE HDF5 FILE. THE ROOT LEVEL HAS SOME BASIC METADATA (I.E., ATTRIBUTE) ABOUT THE ENTIRE DATA SET. EACH SET OF IMAGES ARE PLACED IN A GROUP AND EACH IMAGE IS A DATASET. METADATA IS AVAILABLE AT BOTH THE GROUP AND IMAGE LEVEL.

of the HDF5 file. The metadata is attached as attributes to the root level group. The images are stored as data sets and image-level attributes are associated at this level. Images are grouped together under groups. The data sets contain the original image data. Each HDF5 might contain one or more groups and each group contains one or data sets. Our current hierarchy captures the structure of the file structure organization. Future work will consider alternate structures based on the I/O patterns of the analyses processes.

B. Metadata

The user data has three types of application metadata. Table II-A shows the structure of the packaged HDF5 file and sample metadata.

The root level contains metadata about the whole experiment. This has attributes such as experiment time, owner of the experiment, the original experiment data directory and the original experiment host. The application level metadata at this level contains information about the experiment setup (e.g., angles, exposure, etc). The metadata at the image level contains information about the dimensions of the images and timestamp when the image was collected.

The application metadata is parsed and stored into a MongoDB database when the data arrives at NERSC. In addition to the application metadata, we also associate system level metadata with each data set in the database. If an error was encountered during packaging (e.g., corrupt images, zero length files), the image and all the groups in the chain are marked with an error flag. Additionally, we also register where a file is stored in the filesystem and/or archive so they can be accessed efficiently and easily.

C. Data Transfer

We use Globus Online [5] to transfer data from the beamline to NERSC. Globus Online is a data movement service that provides robust file transfer capabilities. We use Globus Onlines command-line interface (CLI). CLI allows users to interact with Globus Online via secure shell and thus clients don't need to install custom client software.

Both ALS and NERSC are on ESnet (Energy Sciences Network) and hence data transfers happen over ESnet. Currently, we do not leverage any bandwidth provisioning. ESnet provides a bandwidth reservation feature that will let us leverage this in the future to serve the needs of real-time analyses.

D. Data Management at NERSC

Today, ALS users typically conduct their experiments at the beamline and are responsible for copying and taking the data with them. Thus, typically a user needs to download the data to run analyses shortly after the experiment. Rarely, users might return to download previous data sets. In order to balance access time and file system limits for users, we use both file system and HPSS archive. The data arrives at NERSC in project space and the HPSS archive simultaneously. The project space allows us to provide users efficient data access for immediate needs. HPSS provides longer term archive of the data that can be accessed at a future time. However, access of data from HPSS is likely to have slightly longer access times. The data in the project file system is cleaned periodically.

E. User Interface

The user interface (Figure 2) allows users to browse and download their data sets. The users login using their active directory identity and are presented with a view of their data sets. The interface allows them to view the hierarchy in the data sets and download the entire or partial data sets associated with the experiment. The original images or the HDF5 version can be downloaded.

The user interface and back end services are based on NEWT [6]. NEWT provides a web service that allows you to access resources at NERSC through a simple REST API. The user interface queries the MongoDB system to determine the list of files the user has permission to view. The application metadata is also pulled from MongoDB. The portal interacts with the file system or archive to prepare the files for download. If the user requests the raw image files, the files are converted back from HDF5 to raw images to make them available to the user.

F. Security

The users of the tomography beamline have an account in the ALS active directory. The users use the same account to login into the user interface. This enables local control at the ALS of the users accessing the system. NEWT was expanded to support remote active directory authentication. The portal uses a group username to access the file systems on NERSC.

The authorization at the file level, when a user accesses a file through the portal, is handled through the MongoDB system. A user at the interface level has a view to just his or her files. A super user has the ability to see all user files.

III. Experiences

In this section, we detail some of the statistics from the first few months of the system in operation and discuss our experiences. We discuss implications for designing and deploying data-intensive pipelines in the future.

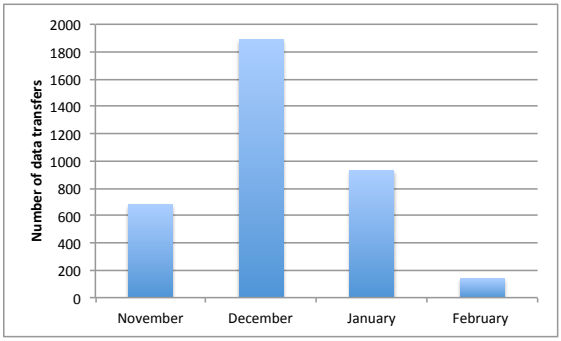

Fig. 3. The figure shows the number of data transfers by month over the period of early November 2012 to February 2013.

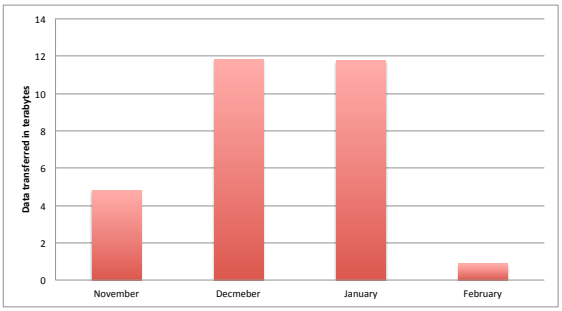

Fig. 4. The figure shows the data transferred in terabytes per month.

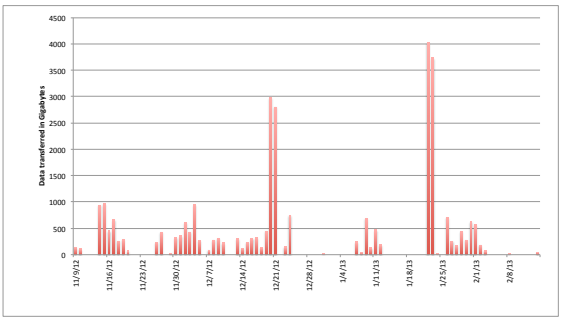

Fig. 5. The figure shows the data transferred per day in gigabytes.

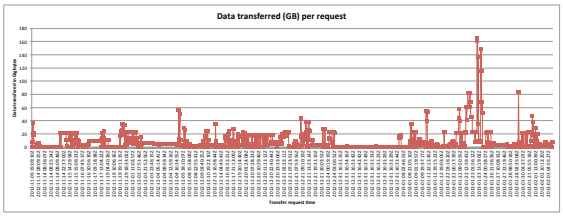

Fig. 6. The figure shows the data transferred in Gigabytes per request.

A. Statistics

The data shown her is the data collected from 2012-11-09 19:09:10Z to 2013-02-14 21:41:58Z roughly a period of three months covering the deployment of the discussed system. The data is collected from Globus Online statistics and included 3643 transfer records that includes retry attempts of failed transfers. A total of 19.39 TB of HDF5 files and 9.9 TB of the tar of the image files were moved. Note that currently three copies of the data are moved a) the raw images for archival b) the HDF5 file for immediate access on the file system and c) the HDF5 file in archive. Approximately 12% of the data transfers failed.

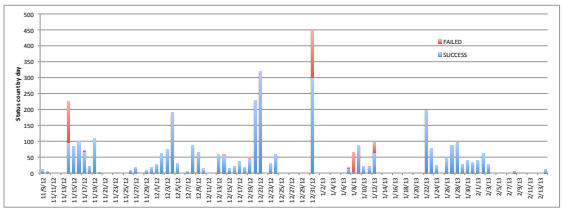

Fig. 7. The figure shows the number of successful and failure transfers by day.

Figure 3 shows the number of data transfers by month. The month of December had the maximum data transfers because there was also some catch up on failed transfers from the days of initial deployment.

Figure 4 shows the data transferred in terabytes in each of the months. In December and January, almost 12 TB data was transferred. On average about 10TB of data is transferred every month.

Figure 5 shows the distribution of amount of data transferred by day. The peaks in the distribution in late December and late January are from manual remediation of earlier failures. The data transferred per day can vary from a few Gigabytes to close to thousands of Gigabytes in a day.

Figure 6 shows the amount of data requested per request. Other than a few outliers, most data transfer requests are less than 60GB and often much lower.

Figure 7 shows the success and failures as detected by Globus Online. These do not account for failures that Globus Online was able to automatically rectify. Failures tend to be clustered to a small number of days indicating problems that often included human intervention. In addition, there were a small number of failures that were outside of the Globus Online system. For example, in one case, the beamline computer shut down resulted in a few hours of no transfers.

B. Transfer challenges

A core component of our framework is the data movement from the ALS to NERSC. We use Globus Online's command-line tools to setup an automated data transfer on data arrival. Globus Online is able to detect and recover from transient errors through a simple retry mechanism. However, we did face a few challenges.

Transfer speeds sometimes fluctuated to lower levels than expected. This might be acceptable for routine transfers. However, there is a need for more real-time analyses at which point there will be a need to consider provisioning required bandwidth using technologies such as OSCARS [7].

Ideally, we would like to schedule retry of failed transfers during a period of low activity on the beamlimes. However, Globus Online does not currently offer a way to schedule transfers at specific times of the day. Additionally, in case of failures during bulk transfers, a catalog of successful versus failed transfers needs to be generated manually to repeat the failed transfers.

The endpoint at the ALS is a Windows based machine with a Samba server that the scientist uses to configure and launch their experiments. The Samba partition is mounted on a Linux data transfer node from where the data transfers are initiated. The files names are derived from user inputs that initially allowed spaces which caused a problem with Globus Online transfers. Subsequently, this was fixed by a Globus server upgrade on the data transfer node and also the user program was changed to restrict spaces.

C. Data Packaging and Analysis Pipelines

The large data volumes associated with the beamline necessitate the need to consider large-scale infrastructure such as those available at HPC systems. Currently scientists take their raw data and either use the beamline computer and/or other desktop resources to analyze the data. However, as data volumes grow there is a need to scale the analyses process. There are trade-offs associated with locality versus scalability for processing in cases where the real-time processing might inform further experimentation. Our current work only focuses on the data packaging pipeline that is a foundation for the analysis pipeline.

Additionally, there is a need for provenance and metadata capture at each step in the analysis pipeline to allow users to understand where and how the data was generated and used for downstream analyses.

D. Data and Metadata Management

Our data and metadata management framework relies on two technologies: HDF5 and MongoDB. HDF5 provides a flexible data model and file format for storing and managing data. It supports a variety of data types and is designed for flexible and efficient I/O. Our framework itself however does not rely on HDF5 and users can plug in various other data packaging formats as desired. Additionally, in the future we will explore the possibility of the beamline data capturing process itself writing out the HDF5 file rather than the raw image format.

MongoDB is a document oriented NoSQL database that we use as a catalog of the files and their metadata. The flexibility of the NoSQL database allows us to use the same database for all ALS beamlines capturing a wide variety of data and metadata. MongoDB meets our current needs and performance requirements. However, the NoSQL space is expanding rapidly and it is possible some of the other offerings might provide better capabilities in the future.

E. HPC resources

There has been limited use of HPC resources for data-intensive pipelines from experiment facilities. We outline some of our challenges and experiences with using NERSC resources for ALS data.

Our initial quota on the file system was not sufficient for the ALS data. We had to get the quota increase and manage data between the archival HPSS system and the parallel file system. The data consists of a lot of small files and we crossed the inode quota. Also, we need to package the data carefully before moving data to HPSS since the small files can cause the migration to tape to fall behind.

A majority of the supercomputing centers today do not provide many services (e.g., databases, catalog, curation) to support data. These services will be essential at large-scale centers that support data collection and data analyses pipelines.

Currently, we use the active directory at ALS to authenticate users to the data. This provides a low barrier of entry for the users using the system. However, as users start to use NERSC resources for analyses, there is a need to bridge the gap between authentication and authorization models at both facilities.

F. End-to-end QoS

Ensuring end-to-end Quality of Service for the data packaging pipeline is non-trivial. Our experience demonstrates the need for verification and validation at each step in the pipeline to verify the step occurred and validate that data wasn't corrupted or other errors weren't encountered. We are exploring the use of frameworks such as Spade to manage the data packaging pipeline.

Errors can occur during the data collection, packaging, transfer or cataloging phases. Also semi-automated failure recovery is critical in such a distributed environment. Often, user intervention is required to rectify the error. However, after the error has been rectified it is often necessary to just perform a set of tasks on all failed tasks.

Performance fluctuations and scalability challenges also need to be considered at every point in the framework. Additionally, we have to consider other operational issues in such a distributed environment. For example, the maintenance windows of NERSC and ALS follow different schedules and so there is a need to coordinate and perform remedial actions on both ends.

IV. Related Work

In this paper, we detail our system architecture for end-to-end data packaging, movement and management of light source data. We detail related work in the area of workflow tools, end-to-end pipelines and use of databases for metadata management.

Workflow tools have provided support for running data-intensive computing pipelines on HPC systems [8] [9] [10] [11] and geographically distributed resources [12] [13]. These tools focus on composing and executing workflows but do not provide specific tools required to package, move and manage experimental data.

CAMP uses an Advanced Messaging Queuing Protocol (AMQP) based messaging service to manage the tasks for procurement from NASA FTP servers and processing at

NERSC of MODIS data [14]. CAMP's focus is on the analyses pipeline for MODIS data in HPC environment and uses FTP for downloading pre-packaged data from NERSC. Our work focuses on the data packaging, movement and management prior to data processing.

Previous work has looked at the use of object oriented database for storing experimental data [15] and evaluated the use of NoSQL databases for scientific use [16].

V. Conclusion

In this paper, we detail a system architecture for data packaging, movement and management for the tomography beamline at the Advanced Light Source. Our pipeline packages data into HDF5 format and uses Globus Online to move the data to NERSC. At NERSC the data is cataloged in a NoSQL database and managed across the file system and archive. Users interact with a web interface to download the data. In future work, the pipeline will be expanded to run real-time and offline analyses workflows at NERSC and allow users to share data with other users.

Acknowledgements

This work was supported by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. The authors would like to thank Dula Parkinson, Elif Dede, Eli Dart, Craig Tull, Abdelliah Essari and Jack Deslippe.

References

[1] "Data and communications in basic energy sciences: Creating a pathway for scientific discovery," http://science.energy.gov/ /media/ascr/pdf/research/scidac/ASCR_BES_Data_Report.pdf, 2012.

[2] H. A. Bale, A. Haboub, A. A. MacDowell, J. R. Nasiatka, D. Y. Parkinson, B. N. Cox, D. B. Marshall, and R. O. Ritchie, "Real-time quantitative imaging of failure events in materials under load at temperatures above 1,600 c," *Nature materials*, 2012.

[3] "Tomography beamline at advanced light source," http://microct.lbl.gov.

[4] D. Gunter, S. Cholia, A. Jain, M. Kocher, K. Persson, L. Ramakrishnan, S. P. Ong, and G. Ceder, "Community accessible datastore of high-throughput calculations: Experiences from the materials project," in *5th workshop on Many-Task Computing on Grids and Supercomputers (MTAGS)*.

[5] B. Allen, J. Bresnahan, L. Childers, I. Foster, G. Kandaswamy, R. Kettimuthu, J. Kordas, M. Link, S. Martin, K. Pickett *et al.*, "Globus online: Radical simplification of data movement via saas," *Preprint CI-PP-5-0611, Computation Institute, The University of Chicago*, 2011.

[6] S. Cholia, D. Skinner, and J. Boverhof, "Newt: A restful service for building high performance computing web applications," in *Gateway Computing Environments Workshop (GCE), 2010.* IEEE, 2010, pp. 1–11.

[7] C. Guok, E. N. Engineer, and D. Robertson, "Esnet on-demand secure circuits and advance reservation system (oscars)," *Internet2 Joint*, 2006.

[8] B. Ludscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. Lee, J. Tao, and Y. Zhao, "Scientific Workflow Management and the Kepler System," 2005, citeseer.ist.psu.edu/ludscher05scientific.html.

[9] T. Oinn, M. Greenwood, M. Addis, M. N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M. R. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe, "Taverna: Lessons in Creating a Workflow Environment for the Life Sciences: Research Articles," *Concurr. Comput. : Pract. Exper.*, vol. 18, no. 10, pp. 1067–1100, 2006.

[10] E. Deelman, G. Singh, M. hui Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz, "Pegasus: a framework for mapping complex scientific workflows onto distributed systems," *SCIENTIFIC PROGRAMMING JOURNAL*, vol. 13, pp. 219–237, 2005.

[11] M. Wilde, M. Hategan, J. M. Wozniak, B. Clifford, D. S. Katz, and I. Foster, "Swift: A language for distributed parallel scripting," *Parallel Computing*, vol. 37, no. 9, pp. 633–652, 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167819111000524

[12] R. Buyya, D. Abramson, and J. Giddy, "Nimrod/g: An architecture for a resource management and scheduling system in a global computational grid," in *High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fourth International Conference/Exhibition on*, vol. 1. IEEE, 2000, pp. 283–289.

[13] I. Raicu, Y. Zhao, C. Dumitrescu, I. Foster, and M. Wilde, "Falkon: a fast and light-weight task execution framework," in *Supercomputing, 2007. SC'07. Proceedings of the 2007 ACM/IEEE Conference on*. IEEE, 2007, pp. 1–12.

[14] V. Hendrix, L. Ramakrishnan, Y. Ryu, C. van Ingen, K. R. Jackson, and D. Agarwal, "Camp: Community access {MODIS} pipeline," *Future Generation Computer Systems*, no. 0, pp. –, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X13002021

[15] A. Adesanya, T. Azemoon, J. Becla, A. Hanushevsky, A. Hasan, W. Kroeger, A. Trunov, D. L. Wang, I. Gaponenko, S. Patton, and D. R. Quarrie, "On the verge of one petabyte - the story behind the babar database system," *CoRR*, vol. cs.DB/0306020, 2003.

[16] E. Dede, M. Govindaraju, D. Gunter, R. Canon, and L. Ramakrishnan, "Semi-structured data analysis using mongodb and mapreduce: A performance evaluation," in *Proceedings of the 4th international workshop on Scientific cloud computing*, 2013.