Procedia Computer Science
Volume 80, 2016, Pages 1927–1939
ICCS 2016. The International Conference on Computational Science

Community Science Exemplars in SEAGrid Science Gateway: Apache Airavata based Implementation of Advanced Infrastructure

Sudhakar Pamidighantam*, Supun Nakandala, Eroma Abeysinghe, Chathuri Wimalasena, Shameera Rathnayaka Yodage, Suresh Marru, Marlon Pierce

Research Technologies Division, University Information Technology Services, Indiana University

Abstract

We describe the science discovered by some of the community of researchers using the SEAGrid Science gateway. Specific science projects to be discussed include calcium carbonate and bicarbonate hydrochemistry, mechanistic studies of redox proteins and diffraction modeling of metal and metal-oxide structures and interfaces. The modeling studies involve a variety of ab initio and molecular dynamics computational techniques and coupled execution of a workflows using specific set of applications enabled in the SEAGrid Science Gateway. The integration of applications and resources that enable workflows that couple empirical, semi-empirical, ab initio DFT, and Moller-Plesset perturbative models and combine computational and visualization modules through a single point of access is now possible through the SEAGrid gateway. Integration with the Apache Airavata infrastructure to gain a sustainable and more easily maintainable set of services is described. As part of this integration we also provide a web browser based SEAGrid Portal in addition to the SEAGrid rich client based on the previous GridChem client. We will elaborate the services and their enhancements in this process to exemplify how the new implementation will enhance the maintainability and sustainability. We will also provide exemplar science workflows and contrast how they are supported in the new deployment to showcase the adoptability and user support for services and resources.

Keywords: SEAGrid Science Gateway, Cyberinfrastructure, Apache Airavata, Computational Chemistry, Workflows, Community Infrastructure

1. Introduction

Science Gateways are web portals and desktop applications that enable scientific communities to more simply and more powerfully access geographically distributed computational and data resources

*Corresponding Author. Tel.: 1-217-333-5831
Email Address: pamidigs@iu.edu

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.535

(Soddemann, 2007; Wilkins-Diehr, 2008). Gateways also benefit computing resource providers by broadening the providers' user communities to include new an non-traditional supercomputer users in a controlled fashion: the gateway encodes best practices for running jobs, using resources optimally, moving data off shared resources and onto archival storage, and generally helping its users be good users of valuable shared systems. Consequentley, the number of users that science gateways bring to national cyberinfrastructure systems such as XSEDE now exceed users accessing the systems by traditional logins (Lawrence, 2015).

Science gateways provide a single point of access to many resources and can help executing workflows, or processing pipelines, across resources in a seamless fashion for the users. Science itself is multidisciplinary, and problem solving environments need to provide user interfaces to easily access critical applications from across disciplines in a transparent manner for efficient scheduling of the computations. The Science and Engineering Applications (SEAGrid.org) science gateway is such an environment. It has evolved from Computational Chemistry Grid infrastructure (Dooley, 2006) that initially provided computational chemistry tools and later integrated engineering applications such as Abaqus and Nek5000. This manuscript describes a productive SEAGrid Science gateway that is celebrating 10 years of service to the community by presenting scientific discoveries from the community of users and by describing SEAGrid's redesign and reimplementation using Apache Airavata science gateway middleware, which will enable sustainability and expansion in the future.

This paper is organized as follows. In Section 2 we describe the capabilities of the current infrastructure for handling the community needs. In Section 3, we present specific science discoveries made by the SEAGrid community of researchers that exemplify SEAGrid's capabilities. In Section 4, the new implementation of the middleware using a hosted Apache Airavata gateway infrastructure is described. Section 5 surveys related work, and Section 6 provides conclusions from current implementations and outlook for additional advance services.

2 SEAGrid Systems, Organization and Current Capabilities

2.1 SEAGrid System Overview

SEAGrid provides a locally running desktop application portal (client) and hosted middleware to run simulations through batch scheduling systems at various High Performance Computing (HPC) sites. Each user has a SEAGrid allocation award, or project, that is tracked by the gateway's middleware. The SEAGrid desktop client is a platform independent application that is developed in Java and includes third-party helper applications for visualization and data movement. The desktop client provides functions such as the authentication, job submission, transfer of data and other services. The functions are routed through a remote server running the Grid Middleware Service (GMS) (Dooley, 2006; Shen, 2014). The GMS is a secure web service using both SSL and X.509 security for communication. Through the use of a MySQL database, the GMS acts as the central "hub" of the SEAGrid. Client functionality provides pre-processing and input generation, job creation and editing, job and output monitoring and post-processing functions.

SEAGrid aims to simplify access to HPC systems. These systems may be part of a larger grid (like XSEDE), but they may also be unaffiliated local campus HPC resources. Providing a unified allocation mechanism is thus one of SEAGrid's core services. The current user community consists of researchers who use the SEAGrid desktop client through SEAGrid allocated grants administered by the virtual organization (called "community users"). Individually funded allocations from the HPC resource providers or from the XSEDE's XRAC allocation process are also supported. PI user accounts and projects in SEAGrid are controlled and vetted through web registration and an allocation request/review process. SEAGrid allocation blocks from XSEDE are used to support the community allocations requests. PIs themselves can manage resource allocations of their students, postdocs, and other participants under the SEAGrid provided (allocated) projects.

In addition to the desktop client, a new site, SEAGrid.org, has been developed and is now available. This site provides a browser-based science gateway interface for job and data management as well as gateway administrative functions for managing users, applications, and resources. Both the SEAGrid desktop and browser clients use a common middleware based on Apache Airavata (Marru et al, 2011; Smith et al., 2015), so users can switch between the two client types as desired. The client and middleware will also allow existing users to access an XSEDE or local allocation they have acquired on their own by registering explicitly. Consulting services and issue reporting in SEAGrid is provided through the JIRA service desk application. The grid middleware and client software stacks are Java-based code, maintained on the Apache Airavata Git repository, with supplemental, SEAGrid-specific code maintained on GitHub under the SciGaP project (https://github.com/scigap). The compiled desktop application is distributed via the middleware server and updated immediately on the researcher's desktop through Java Web Start technology. A team of programmers and scientists is responsible for debugging, updating, and releasing coordinated versions of the client and middleware software stacks.

Wherever possible, SEAGrid leverages advanced services provided by resource providers. For example, SEAGrid makes use of advanced information for users of XSEDE resources. We have integrated job start time estimation on XSEDE resources both for jobs to be submitted and for jobs that are already in queue. To help users move large data files to and from community accounts, we have protoyped the integration of Globus based data sharing capabilities in the SEAGrid desktop client from an XSEDE compute resource where sharing is enabled. In a collaborative project with XSEDE staff, we integrated XSEDE's resource information services into SEAGrid gateway. This work (Smith et al., 2015) involved AMQP based messages that are aggregated and transfer HPC resource data into SEAGrid middleware database. The general system status and queue level load information is now available in SEAGrid desktop client as a result of this project. The resource information for job and queue status is collected by Karnak prediction and data services (Fan et al, 2014). The services have been implemented as REST services and interfaces in HTML and command line and java clients are made available. The software components developed during this project are openly distributed from github. The services are generic for science gateways and currently integration of these services in Apache Airavata and SciGaP projects are underway.

The middleware architecture design provides an abstraction model wherein the worker services are hidden from the client by a management layer. The middleware uses a set of common interfaces to aggregate and organize the information and capabilities. This approach allows easy addition, removal, replacement, and upgrading of services without forcing an update to the client software. Currently, the SEAGrid middleware relies on multiple third party services and an integrated database. Services provide basic information on resources (software, networks, HPC systems), users (projects, profiles, preferences), and jobs (cost, resource); the information is maintained persistently in a relational database that establishes relationships that are meaningful to both users and administrators.

2.2 SEAGrid Applications and Workflows

SEAGrid provides access to several well-known, well-supported community and commercial packages that are maintained by XSEDE and other service providers. The most popular SEAGrid applications include Gaussian, NWChem, GAMESS, Molcas, and the molecular dynamics (MD) packages Amber, NAMD and Gromacs. SEAGrid gateway currently uses a site-specific script to manage jobs submissions and provide job information to the GMS. The site-specific scripts are also used to provide workflow capabilities by conditional execution of coupled applications in a single site, in multiple queues on the same site as dependent jobs, or even multiple sites as dependent jobs, keeping the single original job handle for the user to monitor. These site-specific scripts are also very handy in terms of deploying new and modifying existing applications and for integrating with post processing modules at a particular site. As described in section 3.3 below, several workflows were deployed that couple different functions in LAMMPS MD with LAMMPS compute modules such as XRD and SAED to generate diffraction patterns and consequently visualized using VisIt. Workflows that couple QuantumEspresso SCF calculations with subsequent Bands runs are provided. Other workflows include a Nek5000 based generation of Vortex Street based on user provided code integration for a flow simulation.

2.3 Data Analytics for Runtime Prediction and Data Sharing

SEAGrid infrastructure archives user simulation data and job metadata for its uses. Viewed globally, this wealth of data can be used to compute and provide an estimate of total runtime for new simulations. We compute the total run time prediction by correlating the input attributes with runtime based on machine learning techniques. Using input and output data from the Gaussian application, the current mean error in prediction of run time is about 56%. The dataset consists of more than 40 input attributes to be correlated and the prediction includes restarted jobs. Data for jobs with very small run time (5min) were not considered during the training. We have also created a data analytics platform to present user generated simulation data by engaging simulation output parsers present in the client and the middleware that generate metadata as described further in section 4.3

2.4 User Support

Maintaining the continued support and involvement of an extended SEAGrid user community requires that users are fully supported in their use of SEAGrid applications. This includes the timely addressing of user questions and problems, as well as providing training and educational support in the use of the application software. Several new software suites were integrated since 2014 including DFTB+, Tinker, Abinit, Quantum Espresso, CP2K, Molcas, and Nek5000. We also have recently tested the large memory nodes of SDSC Comet by executing Abaqus software with large memory requirements exceeding 1TB. New workflows using modules in Quantum Espresso are being prototyped based on research needs in a nano-photonics project driven by PI Prashant Jain from UIUC. Users are supported through a consultation system (http://www.gridchem.org/consult), the SEAGrid/GridChem help mailing list (help@gridchem.org), and e-mail and telephone contact. More than 520 tickets were answered using the consulting portal since inception and some direct email and phone support has also been provided to novice and new users.

3. Community Research Highlights

SEAGrid community has been active in research and teaching some recent highlights from these activities that benefited from SEAGrid gateway are presented below. These are only a few out of many accomplishments and are provided only to illustrate the capabilities and by no means an exhaustive set. An extensive list of SEAGrid-enabled publications is available from SEAGrid.org.

3.1 Comparative Enzymatic Hydroxylation Thermodynamics

Bach and co-workers (Badieyan, 2015), supported by SEAGrid allocations and services, provided an elucidation of the mechanism of N-hydroxylation by flavin-dependent monooxygenase enzymes that are important in xenobiotic catabolism and the biosynthesis of sterols, fatty acids, and siderophores. In this DFT analysis they showed how ornithine is selectively hydroxylated over lysine by the enzyme as depicted in Figure 1. SEAGrid provided checkpoint based restart capability used by the scientists for critical frequency calucations required for obtaining the transition structures (TSs) in the reaction coordinate.

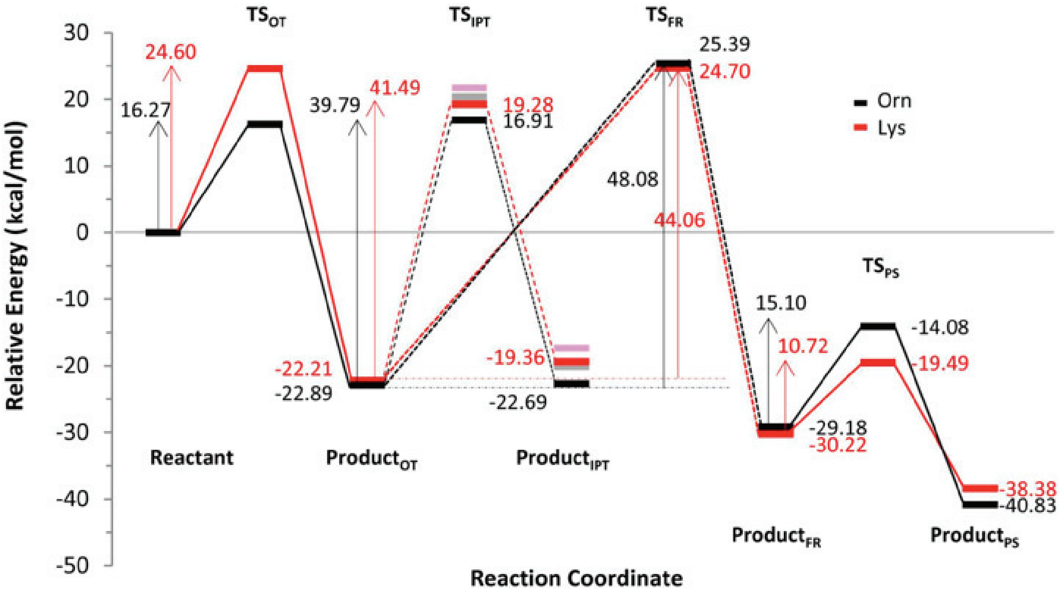

Figure 1. Relative energy profiles for the hydroxylation reactions of Orn and Lys catalyzed by SidA, calculated at the B3LYP/6-311+G(d,p) level (reprinted from J. Org. Chem. 2015, 80, 2139−2147)

3.2 Calcium Carbonate Hydration

In a bio-mineralization simulation work Espinoza-Marzal and co-workers involved multiple software tools and techniques to ascertain the free energies of hydration of calcium carbonate for various hydration models. The computation involved Monte Carlo searches using classical force fields followed by semi-empirical quantum chemical calculations using DFTB+ software and subsequently ab initio and DFT computations for energetics. This study clarified the hydration environment for calciun carbonate and defined the energetics of hydration in first and second shells around the central calcium carbonate moiety (Lopez-Barganza, 2015).

3.3 Diffraction Workflows

The research of Coleman and Spearot (Coleman, 2014; Coleman, 2015) provided the diffraction characteristics of alumina polymorphs using an efficient implementation of diffraction compute as part of LAMMPS software. The polymorphs are otherwise difficult to distinguish by methods such as centrosymmetry analysis or radial distribution functions (Coleman, 2015) while the current diffraction based methods work very well. This research involved the extension of the LAMMPS package to support diffraction simulations, which we integrated into the SEAGrid gateway. To further enable the research, we also developed infrastructure in the SEAGrid gateway to distribute the computing and the visualization tasks on multiple resources available in XSEDE to exploit specialized hardware and software available on different systems. This workflow is depicted in Figure 2. SEAGrid support for both customized LAMMPS deployments, optimal use of XSEDE resources, and simulation-visualization coupling through workflows expedited Coleman's Ph. D. work (Coleman, 2014).

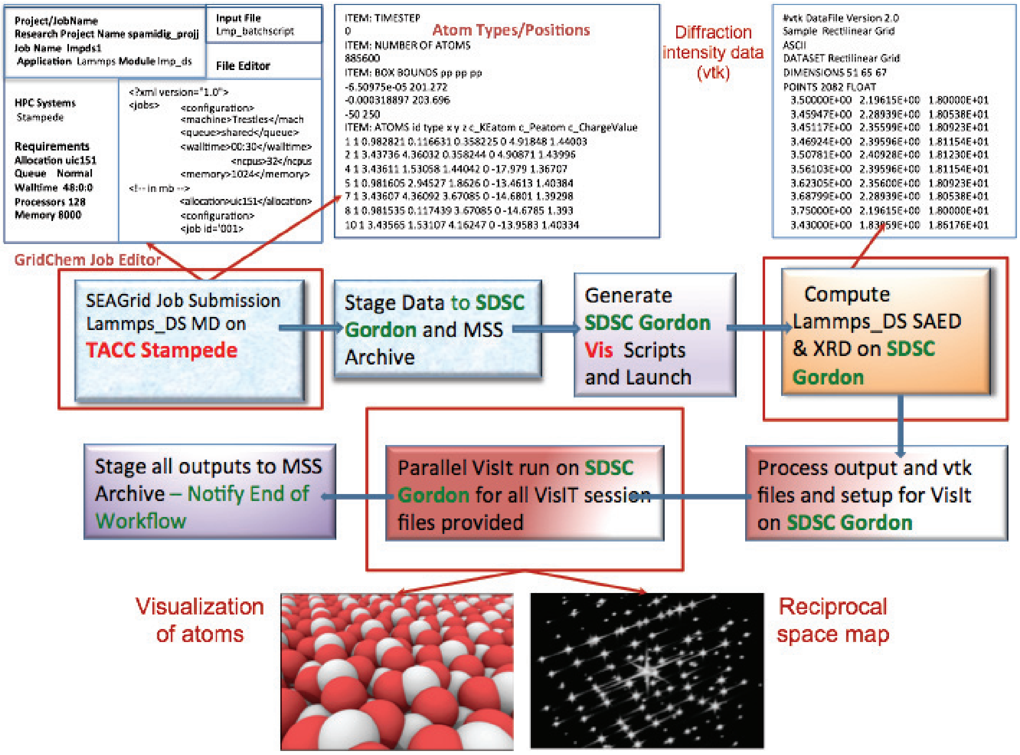

Figure 2. Schematic workflow depicting diffraction computation and visualization tasks that are orchestrated from GridChem client of SEAGrid Science Gateway.

3.4 Vortex Shedding Simulation Gateway

This project builds on SEAGrid infrastructure to provide a gateway that illustrates the fluid dynamics vortex shedding phenomenon for educational users. As part of this project, we deployed Nek5000 spectral element fluid dynamics software on SEAGrid to run on the Comet XSEDE resource and use the GenAPP application wrapping software (Brookes, 2015) to provide limited user inputs. The fully configured gateway is now accessible from http://gw165.iu.xsede.org/vortexshedding/. This deployment takes a user subroutine and recompiles Nek5000, prepares additional input files on the fly, runs Nek5000 fluid dynamics, analyzes the resulting scalar field data using VisIt software, and prepares movies using ffmpeg. The final movie is then delivered to the gateway. An example set of results from this project is presented in Figure 3. The project lead, Prof. Arne Pearlstein at the University of Illinois, uses NERSC systems for production calculations, and we plan to integrate NERSC resources into SEAGrid for their group during this allocation period. We will explore integrating SeedMe.org for sharing the visualizations (images and movies) from the vortex shedding gateway during this project as well.

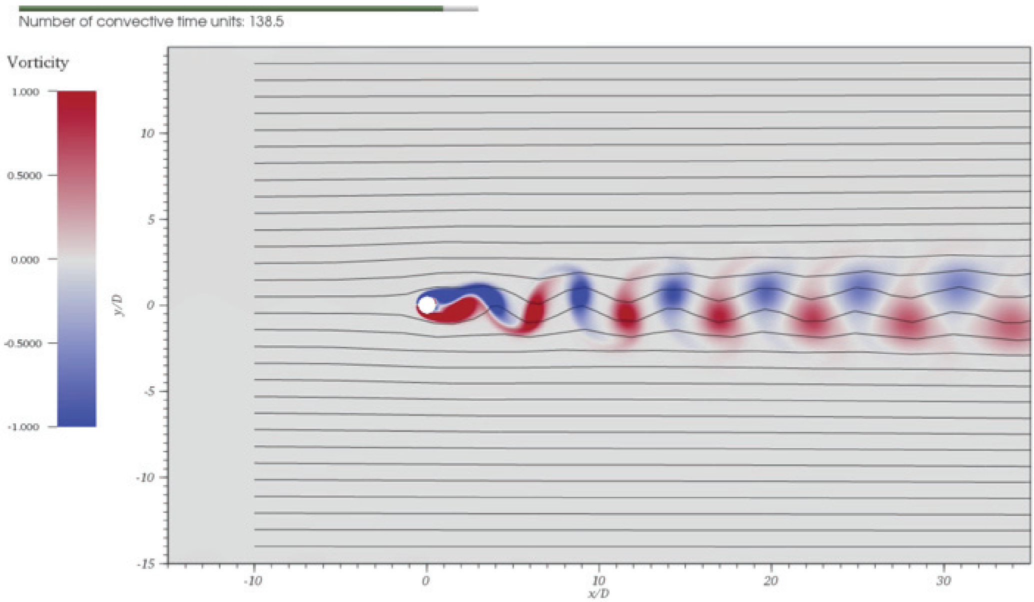

Figure 3. Vortex Street snapshot produced for flow around a spring mounted cylinder simulated using Nek5000 fluid dynamics code visualized using VisIt executed on comet.

4 Sustainable Infrastructure through Apache Airavata Integration

With over a decade of operational experience and understanding of the challenges for enabling a diverse set of scientific applications on a wide range of computing platforms, SEAGrid has much to offer to other gateway systems. Conversely, SEAGrid, as a single gateway working from a single code base, would benefit from leveraging general purpose gateway middleware and hosted services in use by other gateways. This would allow SEAGrid to concentrate on supporting and growing its user community, improving the scientific user's experience, and other innovations while devoting less effort to operations and maintenance.

Based on these mutually beneficial goals, SEAGrid is moving toward replacing its middleware with hosted Apache Airavata middleware and supporting services that are made available through the NSF-funded SciGaP.org project. We provide an overview of Apache Airavata in this section and describe some of the specific ways that SEAGrid integrates with the gateway.

4.1 Apache Airavata Overview

Apache Airavata is open source, community governed middleware designed to support gateway and workflow clients. It is written in Java but provides programming language independent APIs and data models for client integration and internal communication between components. Apache Thrift provides the interface and data model definition language as well as tools for binding these to programming language-specific software development kits (SDKs). Apache Airavata's API and data models are described in greater detail in (Pierce, 2014).

Apache Airavata APIs provide mechanisms for gateways to enable users to create, execute, and monitor computational experiments and retrieve results. The API also provides extensive methods for gateway operators to manage the metadata descriptions of scientific applications and computing resources used by the gateway. This is prescriptive metadata that is used in the implementation to construct submission requests to HPC resources. Given the size and comprehensiveness of the API, the Apache Airavata team provides a reference client implementation as a PHP Web application, dubbed the PHP Gateway for Airavata (PGA). The PGA is included under the Apache Airavata project and has an Apache v2 open source license but is available as a separate download and Git repository. The PGA serves as the basis for the new SEAGrid Web gateway front end.

While individual gateways can download and run their own instances of Apache Airavata, the related Science Gateway Platform as a service (SciGaP.org) project provides hosted instances of the services that can serve multiple gateway tenants. This multiple-tenant requirement informs the API design and implementation: gateways can create, update and delete their own user, computing resource, and application metadata through API calls, but they do not have the permission to do these operations globally. Global operations are reserved for a few “superadmin” API methods. General Apache Airavata security requirements and user authentication methods are described in (Kanewala, 2014; Heiland, 2015), and a more detailed implementation paper is in preparation.

4.2 SEAGrid Integration with Airavata

In this section, we review specific steps for transforming SEAGrid into an Apache Airavata-hosted gateway. As described in previous sections, SEAGrid already follows the multi-tiered architecture that is typical of gateways. Our tasks were to a) replace SEAGrid’s middleware layer with Apache Airavata while retaining SEAGrid’s functionality, b) simultaneously support the current production desktop client and a new web client with integrated user management, c) provide SEAGrid admins with a comprehensive administrator’s dashboard for easy administration, and d) redesign the desktop client to resolve minor design incompatibilities between the SEAGrid client’s assumptions and Apache Airavata’s API assumptions about the organization of user-created computational experiments. The latter step is not strictly necessary for gateways wishing to adopt the Airavata API and hosted services. The new JavaFX-based rich desktop client, based on the previous production Java Web Start client, is an MVC based application that was developed with extensibility in mind based. The current version allows gateway users to create, run, and view experiments; to upload, download, and browse user files; and to use third party features such as the NanoCAD system for molecular structure creation and specialized experiment editors for GAMESS and Gaussian application.

We first examine SEAGrid’s use of Apache Airavata’s administrator functions. Apache Airavata provides gateway administrators with an API for managing prescriptive metadata about scientific applications. The PGA reference gateway provides a starting point implementation, which we reused for SEAGrid’s Web client. Airavata’s Application Catalog API methods and data models (part of the Registry component) are used to add applications to the SEAGrid gateway. The application catalog has two tiers for interface description and deployment. Both these abstraction layers are coupled through an application module specification. Each of above has a separate user interface in the SEAGrid Admin Dashboard, which the gateway administrator can use to configure the application. Adding an application consists of three steps: 1) defining an Application Module, which may be versioned; 2) defining inputs and outputs of the application (the Application Interface in Airavata’s data model); and 3) for execution entering all deployment related information for each resource it exists (the Application Deployment description in Airavata’s data model).

The Airavata Admin Dashboard is specially designed for gateway administrator operations. After integrating with Airavata, SEAGrid administrators have access to the admin dashboard through the web based SEAGrid gateway. Gateway administrators can manage their compute resource preferences, storage preferences, applications and users through admin dashboard. Gateway administrators can monitor their experiment (submitted job) traffic using Experiment Statistics interface. Here the administrator can monitor gateway activities for a desired period of time and also drill-down to examine specific experiments to get logs, outputs and error files that may be useful for debugging.

A major feature of the new SEAGrid gateway is its dual web and desktop user interfaces. For light users, the Web-based gateway interface provides a simple entry point, whereas for users who need lot of pre- and post-processing the desktop application is the better option. In either case, a user can access all data and metadata through either interface. The Web interface is developed using responsive design principles so that it can scale to different screen resolutions, so power users may still find the Web client useful for monitoring their experiments.

In the new user interfaces, users can create experiments that can be launched later, and they can also clone existing experiments to create new ones with altered input parameters. For example, a user may want to run experiments with slightly different input parameters. Once an experiment is launched, users can monitor the job's progress and also have the option of cancelling or deleting experiments. Experiments can be organized into related projects. Search features allow users to find experiments matching several search criteria.

Airavata framework supports pre and post job command execution trivially and this capability is available both programmatically and in the test portal. The Airavata work distribution and management system is capable of handling conditionally executed coupled tasks in a directed acyclic graph (DAG). Workflow design was enabled through Xbaya and orchestration and enactment is suported through Airavata infrastructure described in (Marru et al , 2007; Marru et al., 2015; Pierce, 2015). The workflows such as the ones shown in the vortex shedding exmaple have been already implemented but more elaborate DAGs will be deployed after the Xbaya refactoring is completed this (2016) summer.

4.3 Data Management

SEAGrid allows users to execute a wide range of scientific applications and download the resulting raw outputs. Additionally we are also building capabilities that allow users to manage their experimental results within the gateway and gain more insights into the data. This schema independent framework consists of an agent that automatically identifies new data products and extracts metadata from them. A server that indexes the metadata using a NoSQL database and a REST API for querying the indexed data is provided. This prototype uses Apache Solr for indexing parsed data from the simulation outputs and organizes the metadata into a non-relational (NoSQL) database. The data can then be provided through a Web browser or via the SEAGrid desktop client; user interfaces enable users to search, organize and share the data with other collaborators or make the data public if desired. This effort is described in more detail in (Nakandala, 2015). This infrastructure will automatically runs data postprocessing pipelines, which includes running application-specific parsers to extract attributes, variables, and metadata. The parsed data are indexed in a data store so that they are queriable by the end users. We have integrated the prototype version of this is feature to the SEAGrid Gateway, which runs within the larger Airavata ecosystem as shown in Figure 4.

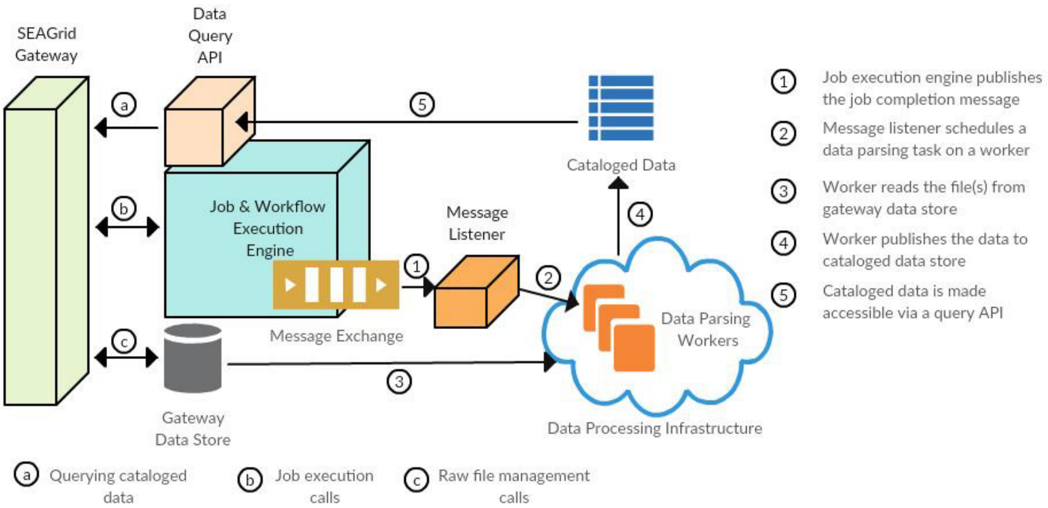

Figure 4: Data post processing pipelines in SEAGrid

Data postprocessing occurs when the Airavata middleware detects that an experiment has completed its execution on an HPC resource, as described in (Marru, 2015; Pierce, 2015). In the data post processing and cataloging pipeline a message listener is listening to the Airavata message exchange which gets notified about successful completion of running an experiment. When such an event is notified to the message listener it will then schedule a data post processing task on a worker that is running on a data processing infratructure. This data postprocessing worker then reads the required file(s) from the gateway data store and runs a set of application and file specific parsers to mine and extract important fields, metadata, attributes and variables. This extracted data is then published to a cataloged data store and those are made accessible to the gateway users via a query API.

In the initial version of this feature we cataloged simulation output of several computational chemistry applications including Gaussian09, Gamess, NWChem and Molpro. Some of the queries that we have facilitated in this version are simple field matching queries including InChI substring matching, molecular formula matching, common name matching, SMILES matching, energy matching and enthalpy matching. Going further, we plan to implement features that can enable users to search existing molecular structures and replace some atoms or the structure and resubmit as a new simulation experiment. In adition to this we also plan to support integrating external molecular databases and enable complex structure filter queries.

5 Related Work

Science gateways have been developed to support a diverse set of scientific communities. Recent overviews of the community are available in (Gesing, 2015; Wilkins-Diehr, 2013). Notable gateways include the CIPRES Workbench for phylogenetics research (Miller, 2010), NanoHUB for nano-engineering (Klimeck, 2008), and the Galaxy Portal for bioinformatics research (Goecks, 2010), all of which have user bases in the thousands. Science gateways for digital arts and humanities are described in (Craig, 2015). Gateways to support computational chemistry and material science include MOSGrid (Kruger, 2014), NanoHub, and Diagrid. Given the popularity of science gateways by some user communities, there are several efforts to provide general purpose frameworks. The software for the Galaxy Portal, for example, can be freely downloaded and used to set up new instances that are otherwise unaffiliated with the main Galaxy site and can be applied to other scientific domains (Madduri, 2015). HUBzero (McLennan, 2010) software is a generalization of the nanoHUB framework and has been used to build other successful gateways. The iPlant Agave framework (Goff, 2011) and WS-PGRADE (Kacsuk, 2012) frameworks provide software and hosted services that are similar in overall design to the Apache Airavata software and hosted services described here.

6. Conclusions and Future Work

As discussed in this paper, SEAGrid's goal is to be responsive to the needs of its user community in order to enable scientific research. Serving a community of users also provides the SEAGrid team with a perspective on global needs and priorities that may not be obvious to individual users. Thus we have both explicit and implicit requirements that guide our future work. In the near term, we are investigating simplified ways for creating graphical interface for MD simulations. We will add (sub)structure searching in databases and provide metadata to the users. We also see longer term opportunities for enabling users to explore over a decade's worth of computational experiments. We would like to enable users to share their data and provide users a search based access to their own and shared data. Finally, we have a goal of blurring the lines between SEAGrid users and developers. SEAGrid is already publically available, open source software that can be taken and modified. Based on our experience in the Apache Software Foundation, we realize that we need to go beyond simply providing source code to also have an explicitly defined community governance model that will encourage a community of scientific tool developers to freely contribute to SEAGrid's code base (Pierce et al Apache Airavata, 2015). We believe that this will lead to the long term vitality of the project and increase its ability to serve the research community.

7. Acknowledgements: This work is partially supported by NSF award #1339774, "Collaborative Research: SI2-SSI: Open Gateway Computing Environments Science Gateways Platform as a Service (OGCE SciGaP)". NSF XSEDE project is acknowledged for continued allocation grant #CHE070035 and ECSS support.

8. References

Soddemann, Thomas (2007) "Science gateways to DEISA: user requirements, technologies, and the material sciences and plasma physics gateway."Concurrency and Computation: Practice and Experience 19, no. 6, 839-850.

N. Wilkins-Diehr, D. Gannon, G. Klimeck, S. Oster, and S. Pamidighantam (2008) TeraGrid Gateways and Their Impact on Science. Computer 41, 32-41

Lawrence, Katherine A., Michael Zentner, Nancy Wilkins‐Diehr, Julie A. Wernert, Marlon Pierce, Suresh Marru, and Scott Michael (2015) "Science gateways today and tomorrow: positive perspectives of nearly 5000 members of the research community." Concurrency and Computation: Practice and Experience .

Dooley, Rion, Kent Milfeld, Chona Guiang, Sudhakar Pamidighantam, and Gabrielle Allen (2006) "From proposal to production: Lessons learned developing the computational chemistry grid cyberinfrastructure." Journal of Grid Computing 4, no. 2: 195-208.

Shen, Ning., Fan, Ye., Pamidighantam, Sudhakar (2014) E-science infrastructures for molecular modeling and parametrization. J. Comput. Science 5(4): 576-589.

Marru, Suresh, Lahiru Gunathilake, Chathura Herath, Patanachai Tangchaisin, Marlon Pierce, Chris Mattmann, Raminder Singh et al., (2011) "Apache airavata: a framework for distributed applications and computational workflows." In Proceedings of the 2011 ACM workshop on Gateway computing environments, pp. 21-28. ACM.

Marru, Suresh, Marlon Pierce, Sudhakar Pamidighantam, and Chathuri Wimalasena. (2015) "Apache Airavata as a laboratory: architecture and case study for component-based gateway middleware." In Proceedings of the 1st Workshop on The Science of Cyberinfrastructure: Research, Experience, Applications and Models, pp. 19-26. ACM, 2015.

Smith, Warren, Sudhakar Pamidighantam, and John-Paul Navarro (2015) "Publishing and consuming GLUE v2. 0 resource information in XSEDE." InProceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure, p. 25. ACM.

Fan, Y., Pamidighantam, S., Smith, W., (2014) “Incorporating Job Predictions into the SEAGrid Science Gateway”, XSEDE 14 Conference, Atlanta, US, July 2014.

Badieyan, S., Bach, R. D., Sobrado, P. (2015) Mechanism of N-Hydroxylation Catalyzed by Flavin-dependent Monooxygenases J. Org. Chem. J. Org. Chem., 2015, 80 (4), pp 2139–2147 DOI: 10.1021/jo502651v

Coleman, S., Koesterke, L., Van Moer, M., Pamidighantam, S., Spearot, D., Wang, Y., (2014) Performance Improvement and Workflow Development of Virtual Diffraction Calculations, XSEDE 14 Conference, Atlanta, US, July 2014.

Coleman, S.P., and Spearot, D.E. (2015) “Atomistic simulation and virtual diffraction characterization of homophase and heterophase alumina interfaces”, Acta Materialia, 82, 403-412: doi:10.1016/j.actamat.2014.09.019

Lopez-Berganza, Josue A., Diao Yijue, Pamidighantam Sudhakar, and Espinosa-Marzal Rosa M. (2015) “Ab Initio Studies of Calcium Carbonate Hydration” The Journal of Physical Chemistry A, *119* (47), 11591-11600; DOI: 10.1021/acs.jpca.5b09006

Brookes, Emre H., Abhishek Kapoor, Priyanshu Patra, Suresh Marru, Raminder Singh, and Marlon Pierce. (2015), "GSoC 2015 student contributions to GenApp and Airavata." Concurrency and Computation: Practice and Experience *28: 1960–1970. doi: 10.1002/cpe.3689*.

Pierce, Marlon, Suresh Marru, Borries Demeler, Raminderjeet Singh, and Gary Gorbet. (2014) "The apache airavata application programming interface: overview and evaluation with the UltraScan science gateway." InProceedings of the 9th Gateway Computing Environments Workshop, pp. 25-29. IEEE Press.

Kanewala, Thejaka Amila, Suresh Marru, Jim Basney, and Marlon Pierce. (2014) "A credential store for multi-tenant science gateways." In Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on, pp. 445-454. IEEE, 2014.

Heiland, Randy, Scott Koranda, Suresh Marru, Marlon Pierce, and Von Welch. (2015) "Authentication and Authorization Considerations for a Multi-tenant Service." In Proceedings of the 1st Workshop on The Science of Cyberinfrastructure: Research, Experience, Applications and Models, pp. 29-35. ACM, 2015.

Nakandala, Supun, Sachith Dhanushka Withana, Dinu Kumarasiri, Hirantha Jayawardena, H. M. N. Dilum Bandara, Srinath Perera, Suresh Marru, and Sudhakar Pamidighantam. (2015) "Schema-independent scientific data cataloging framework." In Moratuwa Engineering Research Conference (MERCon), 2015, pp. 289-294. IEEE.

Gesing, Sandra, and Nancy Wilkins-Diehr. (2015) "Science gateway workshops 2014 special issue conference publications." Concurrency and Computation: Practice and Experience 27.16 (2015): 4247-4251.

Wilkins-Diehr, Nancy, Sandra Gesing, and Tamas Kiss. (2013) "Science gateway workshops 2013 special issue conference publications." Concurrency and Computation: Practice and Experience 27.2 (2015): 253-257.

Miller, Mark, Wayne Pfeiffer, and Terri Schwartz. (2010) "Creating the CIPRES Science Gateway for inference of large phylogenetic trees." Gateway Computing Environments Workshop (GCE), 2010. IEEE, 2010.

Klimeck, Gerhard, Michael McLennan, Sean P. Brophy, George B. Adams III, and Mark S. Lundstrom. (2008) "nanohub. org: Advancing education and research in nanotechnology." Computing in Science & Engineering 10, no. 5: 17-23.

Goecks, Jeremy, Anton Nekrutenko, and James Taylor. (2010) "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences." Genome Biol 11.8: R86.

Craig, Alan B. (2015) "Science gateways for humanities, arts, and social science."Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure. ACM, 2015.

Krüger, Jens, Richard Grunzke, Sandra Gesing, Sebastian Breuers, André Brinkmann, Luis de la Garza, Oliver Kohlbacher et al. (2014) "The MoSGrid science gateway–a complete solution for molecular simulations." Journal of Chemical Theory and Computation 10, no. 6 : 2232-2245.

Madduri, Ravi, Kyle Chard, Ryan Chard, Lukasz Lacinski, Alex Rodriguez, Dinanath Sulakhe, David Kelly, Utpal Dave, and Ian Foster. (2015) "The Globus Galaxies platform: delivering science gateways as a service." Concurrency and Computation: Practice and Experience *27: 4344–4360. doi: 10.1002/cpe.3486*..

McLennan, Michael, and Rick Kennell. (2010) "HUBzero: a platform for dissemination and collaboration in computational science and engineering."Computing in Science & Engineering 12, no. 2: 48-53.

Goff, Stephen A., Matthew Vaughn, Sheldon McKay, Eric Lyons, Ann E. Stapleton, Damian Gessler, Naim Matasci et al. (2011) "The iPlant collaborative: cyberinfrastructure for plant biology." Frontiers in plant science 2, 1.

Kacsuk, Peter, Zoltan Farkas, Miklos Kozlovszky, Gabor Hermann, Akos Balasko, Krisztian Karoczkai, and Istvan Marton. (2012) "WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities."Journal of Grid Computing 10, no. 4: 601-630.

Pierce, Marlon E., Suresh Marru, Lahiru Gunathilake, Don Kushan Wijeratne, Raminder Singh, Chathuri Wimalasena, Shameera Ratnayaka, and Sudhakar Pamidighantam. (2015) "Apache Airavata: design and directions of a science gateway framework." Concurrency and Computation: Practice and Experience *27: 4282–4291. doi: 10.1002/cpe.3534* .

Pierce, Marlon E., Suresh Marru, and Chris Mattmann. (2015) "Patching It Up, Pulling It Forward." Journal of Open Research Software 3.1 (2015).