Journal of Computational Science 5 (2014) 576–589

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

E-science infrastructures for molecular modeling and parametrization

Ning Shen [1], Ye Fan [1], Sudhakar Pamidighantam *

National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

ARTICLE INFO

Article history:
Received 2 November 2013
Received in revised form 23 January 2014
Accepted 30 January 2014
Available online 7 February 2014

Keywords:
Science gateways
Cyberinfrastructure
E-science
GridChem
Virtual organizations
XSEDE
Scientific workflows
Science and engineering simulation and design
Computational chemistry
Molecular modeling

ABSTRACT

E-science infrastructures are becoming the essential tools for computational scientific research. In this paper, we describe two e-science infrastructures: Science and Engineering Applications Grid (SEAGrid) and molecular modeling and parametrization (ParamChem). The SEAGrid is a virtual organization with a diverse set of hardware and software resources and provides services to access such resources in a routine and transparent manner. These essential services include allocations of computational resources, client-side application interfaces, computational job and data management tools, and consulting activities. ParamChem is another e-science project dedicated for molecular force-field parametrization based on both ab-initio and molecular mechanics calculations on high performance computers (HPCs) driven by scientific workflow middleware services. Both the projects share a similar three-tier computational infrastructure that consists of a front-end client, a middleware web services layer, and a remote HPC computational layer. The client is a Java Swing desktop application with components for pre- and post-data processing, communications with middleware server and local data management. The middleware service is based on Axis2 web service and MySQL relational database, which provides functionalities for user authentication and session control, HPC resource information collections, discovery and matching, job information logging and notification. It can also be integrated with scientific workflow to manage computations on HPC resources. The grid credentials for accessing HPCs are delegated through MyProxy infrastructure. Currently SEAGrid has integrated several popular application software suites such as Gaussian for quantum chemistry, NAMD for molecular dynamics and engineering software such as Abacus for mechanical engineering. ParamChem has integrated CGenFF (CHARMM General Force-Field) for molecular force-field parametrization of drug-like molecules. Long-term storage of user data is handled by tertiary data archival mechanisms. SEAGrid science gateway serves more than 500 users while more than 1000 users use ParamChem services such as atom typing and initial force-field parameter guess at present.

1. Introduction

Science and engineering disciplines have been doing simulations with high performance computers (HPCs) for testing new concepts, corroborating experiments, prototyping new designs and predicting events hitherto unknown. Since most understanding of nature is multidisciplinary and HPC hardware evolves rapidly, the abstraction of the complexities of HPC systems from researchers has been shown to improve the simulation experience and reduce the time to find solutions. This idea [1] resulted in many Science and engineering gateways or e-science portals [2,3]. A science gateway, gateway in short, is an infrastructure that provides an organization of integrated resources and services and manages users and their simulation activities. An evolving gateway is expected to support user communities and their changing needs, particularly those require increasingly interdisciplinary science and engineering simulations in a sustainable environment. In the TeraGrid organization alone, there were 22 science gateways by 2010, which consume 40 million standardized services and support 40% of the total users [4]. Some of the chemically oriented e-science infrastructures include CSE Online [5], Nanohub [6], chem-informatic data mining infrastructure [7] and crystal structure submission infrastructure [8]. The growth in the science gateway program shows that this is a fundamental way for researchers to perform researches. The gateway uses a variety of technologies to provide a diverse set of resources to their respective communities. The resources could be dedicated computer clusters, national and local supercomputing facilities such as XSEDE (https://www.xsede.org/) or Ohio Supercomputer center, Cloud resources such as Amazon EC2 or Microsoft Azure. The technologies that integrate these resources range from simple secure shell protocols to grid and web services. All these critical components of successful gateways have been reviewed in [9,10].

* Corresponding author. Tel.: +1 217 333 5831.
E-mail addresses: spamidig@ncsa.illinois.edu, s.pamidighantam@gmail.com (S. Pamidighantam).

[1] These authors contributed equally.

1877-7503/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jocs.2014.01.005

This paper describes recent developments in two such science gateways: SEAGrid and ParamChem infrastructures to illustrate the e-science infrastructures in production and development.

A basic deployment experience of the initial production version of SEAGrid known as Computational Chemistry Grid (CCG) has been described in [3,11,12]. Here we describe the extensions of the gateway services beyond supporting chemistry applications to serve both science and engineering applications. Two e-infrastructures SEAGrid (Sections 2.1–2.3) and ParamChem (Sections 3.1–3.3) will be described in detail. In order to present how services are integrated from the perspectives of the user and the gateway infrastructures, the manuscript is organized as follows. The description of the Graphical User Interface (GUI) client of SEAGrid (DESSERT) is presented in Section 2.1. The Grid Middleware Services (GMS) that handle the common transactions in the SEAGrid and ParamChem infrastructure are described in Section 2.2. The production usage of science and engineering applications in SEAGrid is given in Section 2.3. Introduction of the ParamChem organization is presented in Section 3.1, followed by the descriptions of the GUI client (Paramberoo) in Section 3.2. Section 3.3 describes the middleware services that mediate the operations in ParamChem infrastructure along with the integration of Apache Airavata workflow management system [13]. Finally, the current work and outlook in these projects are summarized in Section 4.

2. Science engineering applications Grid: SEAGrid

The SEAGrid (Science Engineering Applications Grid) previously known as CCG is a virtually organized community cyber-infrastructure in production. The cyber component of the infrastructure is made up of several XSEDE supported HPC resources, middleware computer servers, and mass storage facilities to archive user data. The organization has provided over 11 million service units (SUs) of allocated CPU time to the community in the last year. The components of the infrastructure are described in the sections below.

2.1. The graphical user interface: DESSERT

User interacts with SEAGrid through a desktop client named DESSERT, which is refactored based on the GridChem client [3]. It is a Java Swing application deployed as a JNLP application. This client, launched using Java Web Start application locally on a laptop or desktop system, provides users the convenience of click-and-run and automatically updates itself whenever the software is updated by the developers with new features or bug fixes. Fig. 1 shows the simulation job setup panel in the client, which provides a way to select application software, HPC resource (Blacklight at Pittsburgh Supercomputer Center, Trestles at San Diego Supercomputer Center, etc.), job requirements (execution queue, wall-clock time, the number of processors, memory), and options to provide required input file(s) for the application job. It also shows the client's capability to accept multiple job submissions through a single job script with a plugged-in job parser, which parses and validates the job script. The job script is an XML file that defines a series of parallel jobs with similar job parameters, as needed in jobs such as parameter sweep calculations. The input file for each job is defined in "<input>" tag of the XML file. In this example, two jobs are submitted here with name "lmp_batch_1", "lmp_batch_2". The computing resource being used is Trestles. User has requested 32 CPUs with 30 min wall-time with allocation id "uic151" for each job using the common requirement configuration read from the input file for each job. User also has the capability to change the job requirements through the requirement section of client interface for single jobs.

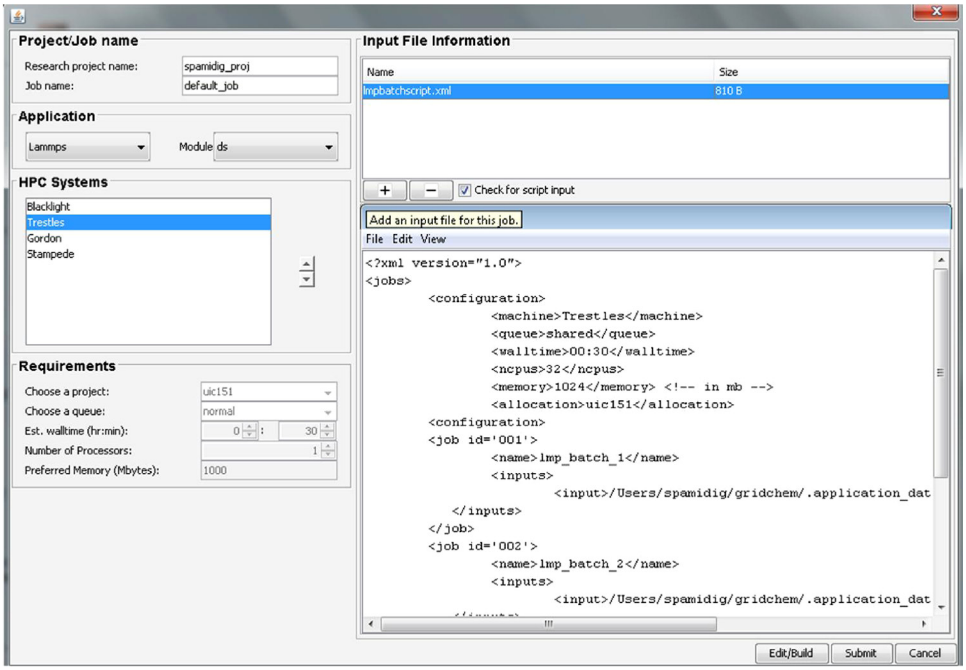
To support the growing community of users, various HPC resources and software suites supported on them are integrated into DESSERT. The set of scientific software currently supported includes quantum chemistry codes LAMMPS [14], Gaussian [15], Gamess [16], NWChem [17], Molpro [18] and DMol3 [19] and molecular dynamics codes AMBER [20], NAMD [21], and Gromacs [22], and molecular scattering code DDScat [23]. Some of the software is restricted to only the bona fide group users. Apart from the above scientific suites, the computational structural mechanics suites Abaqus [24] has been integrated along with a visualization interface CAE [25] that can be launched as part of post processing if one is installed on the user's local system. The newly deployed hardware systems such as trestles at San Diego Supercomputer Center (SDSC) and Blacklight at Pittsburgh Supercomputer Center (PSC) are included that support some of these software. For example, a stress simulation of solid flanges in contact conducted by the Abaqus modules is shown in Fig. 2, which is post-processed with CAE interface launched by the DESSERT client (Fig. 2b).

2.2. Grid Middleware Service: GMS

The Grid Middleware Service (GMS) is designed to hide the worker services from the client with a set of common interfaces to perform essential operational services (such as user authentication, data staging, and job monitor, etc.) common to both SEAGrid and ParamChem infrastructures. The basic organization of the middleware is described in [3] and enhancement in data organization, refactoring the core services using AXIS2 software framework and new extensions to login services are discussed further in this section. For ParamChem infrastructure, GMS is also integrated with Apache Airavata scientific workflow services, which will be described in later sections. Fig. 3 shows the abstraction model of GMS with DESSERT client.

Access to the gateway services is controlled and only verified academic investigators are provided with an allocation. Researcher firstly needs to register for both username and password through a secure web portal and request community HPC allocations under the block allocations to the gateway and/or register their existing NSF/XSEDE allocations. Upon registration and successful allocation, a project ID is provided. Additional users can be added to the project and the group together can use the resources allocated under the project ID. Then the user can run computations optionally organizing them under different research project IDs. Behind the scene, a MySQL database is used in the GMS to store the data and map the relations between the data with five relational tables. *User*, *ComputeResource* and *Project* tables store the raw data while *UserProjects* and *ProjectResources* tables store the mapping relations among the data. For example, the MySql fields, userID and projectID are primary keys in User and Project tables, which also serve as foreign keys in the *UserProjects* table. Similar rule applies to the *ComputeResource*, *Project* and *ProjectResource* tables. Additional features are implemented in the database to address various restrictions and scenarios. It is common to have HPC service stopped time to time due to maintenance or eliminated at retirement of the resource. A Boolean field named "enabled" is added in *ComputeResource* table to identify whether the HPC resources are available at that moment. This design provides a consistent way to address the resource availability. Secondly, a blacklist table is added to database to prevent user groups from accessing particular compute and/or application resources based on a requirement, such as license. Hibernate library [26] is applied to communicate with the database, which features with the object relation mapping (ORM) to map between Java classes and database tables.

User authentication is essential under the multiple user environment infrastructures. The user authentication process in SEAGrid occurs in two stages. Registered users firstly authenticate



Research project name: spamidig_proj
Job name: default_job

Lammps
Module ds

Blacklight
Trestles
Gordon
Stampede

Choose a project: uic151
Choose a queue: normal
Est. walltime (hr:min): 0 : 30
Number of Processors: 1
Preferred Memory (Mbytes): 1000

Name	Size
lmpbatchscript.xml	810 B

☑ Check for script input

Add an input file for this job.

```
<?xml version="1.0">
<jobs>
        <configuration>
                <machine>Trestles</machine>
                <queue>shared</queue>
                <walltime>00:30</walltime>
                <ncpus>32</ncpus>
                <memory>1024</memory> <!-- in mb -->
                <allocation>uic151</allocation>
        <configuration>
        <job id='001'>
                <name>lmp_batch_1</name>
                <inputs>
                        <input>/Users/spamidig/gridchem/.application_dat
          </inputs>
        </job>
        <job id='002'>
                <name>lmp_batch_2</name>
                <inputs>
                        <input>/Users/spamidig/gridchem/.application_dat
```

Fig. 1. Job set up panel in DESSERT client with multi-job submission script.

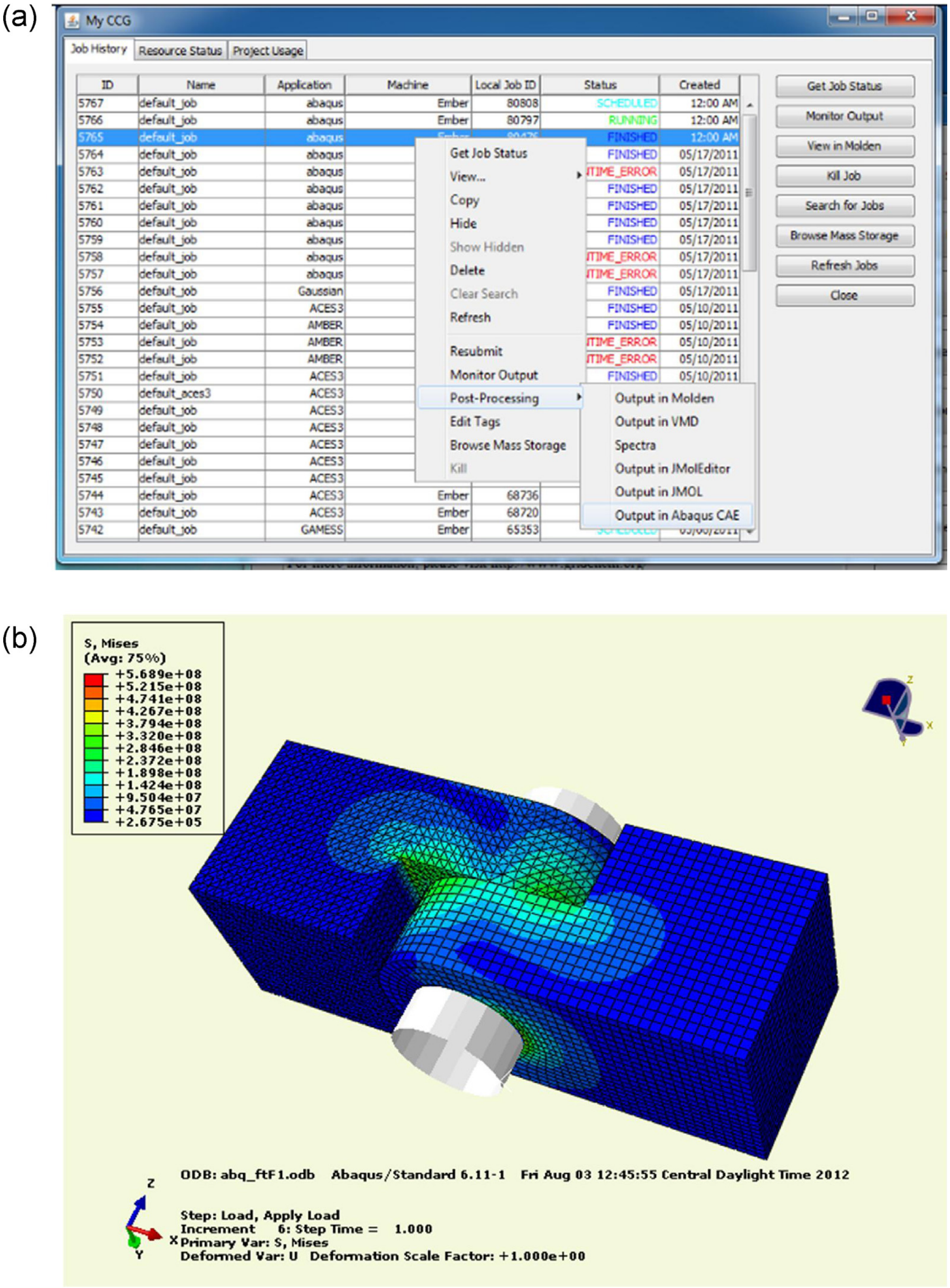
themselves through the middleware database. After that, the GMS applies a Globus web service [27] and delegates an X.509 security credentials [28] using Secure Socket Layer (SSL) [29] for socket based secure communications to provide an authorized access to HPC resources.

Factory design pattern is applied to provide the login function for different types of access such as community or XSEDE user based on their SEAGrid allocation type selected in the client. A Java Factory class *LoginFactory* firstly returns an appropriate *LoginProvider* object based on the access type. Then, the *LoginProvider* performs the actual login task and creates an *AutenticationBean* object. The *SessionManager* function generates a new login session with default 12 h validity as shown in Fig. 4. After that, the last login timestamp is recorded through a Hibernate database transaction and an authentication token ID is created that also serves as the session identity and sent back to the client. While the session is active, the client communicates with the service (GMS) using the token instead of authentication itself every time and subsequent transactions are labeled with the token ID, which make it easy to track them and help troubleshooting in the case of any unexpected behavior. The middleware is integrated with the XSEDE science gateway attribute-based authentication through GridShib and SAML [30] tokens to provide individual user attributes such as SEAGrid ID and an as a community credential to the HPC resources. These attributes may be used to enforce certain policies for the community users at the HPC resource. For the users using XSEDE allocation, additional XSEDE userid and pass phrase are read along with SEAGrid username and password. Both sets are used in the login service and processed accordingly. The *LoginFactory* class returns an appropriate *LoginProvider* object based on the access type 'XSEDE login', and subsequently runs the login task to create an *Autentication-Bean* object. The X.509 credential will use the XSEDE user id and passphrase in this case and the rest of the process is same resulting in an authentication token.

The GMS web services are based on Apache Axis2 framework [31] and use Simple Object Access protocol (SOAP) to transfer XML-formatted messages. Contract-First development strategy is applied in developing the Axis2 web service. With the contract-first approach, a web services description language (WSDL) document is used to describe the contract. Then, an Axis2 tool 'wsdl2java' generates the stub codes from the WSDL contract for accessing web services. A full list of 12 Axis2 web services implemented in GMS can be found at the URL (http://gridchem-mw.ncsa.illinois.edu:8080/axis2/services/listServices).

Here, we illustrate the implementation of web service with the JobService as an example. Fig. 5a shows the service endpoint reference (EPR) of the JobService and 14 associated operations, which we will use in the process of the submitting a job by describing the operations in both client and service sides as depicted in Fig. 5b and c. Fig. 5b shows three steps to call the operations of web service from client side. Firstly, get web services from different Axis2 middleware server URL through the getClient (URL) function. Secondly, prepare the necessary input parameters for specific web service operations. Thirdly, get the JobServiceStub on the server through the getClient().getJobService() method and run the specific operation(such as submit in this example) with necessary input parameters. If the operation has return value (like submit operation), get_return() method can be used to return it to the client. Fig. 5c shows the necessary implementations on the server side. Firstly, getJobService() returns the JobServiceStub which can be used by the client to access specific web services. Secondly, concrete class JobServiceImpl is used to implement all the methods declarations of JobService interface. Implementation of submit operation is shown as an example, where job is submitted after validations of session and job information.

Data management is another important service for multiple user environments in both the infrastructures. We applied tertiary storage mechanism that ensures a long-term backup of user's research data. This includes two independent data transfer processes from HPCs, one, to the client upon request by the user and a second to a dedicated mass storage server, automatically. When client needs to retrieve files from the computer resources, it first notifies the Axis2 FileService about the location of the file. After checking for the existence of file and user's access privileges, the middleware service retrieves the file from the remote HPC to a local temporary directory on middleware server. Then the file is downloaded to client's machine through a socket pooling connection manager to improve performance. Message



ID	Name	Application	Machine	Local Job ID	Status	Created
5767	default_job	abaqus	Ember	80808	SCHEDULED	12:00 AM
5766	default_job	abaqus	Ember	80797	RUNNING	12:00 AM
5765	default_job	abaqus	Ember	[illegible]	FINISHED	12:00 AM
5764	default_job	abaqus			FINISHED	05/17/2011
5763	default_job	abaqus			TIME_ERROR	05/17/2011
5762	default_job	abaqus			FINISHED	05/17/2011
5761	default_job	abaqus			FINISHED	05/17/2011
5760	default_job	abaqus			FINISHED	05/17/2011
5759	default_job	abaqus			FINISHED	05/17/2011
5758	default_job	abaqus			TIME_ERROR	05/17/2011
5757	default_job	abaqus			TIME_ERROR	05/17/2011
5756	default_job	Gaussian			FINISHED	05/17/2011
5755	default_job	ACES3			FINISHED	05/10/2011
5754	default_job	AMBER			FINISHED	05/10/2011
5753	default_job	AMBER			TIME_ERROR	05/10/2011
5752	default_job	AMBER			TIME_ERROR	05/10/2011
5751	default_job	ACES3			FINISHED	05/10/2011
5750	default_aces3	ACES3				
5749	default_job	ACES3				
5748	default_job	ACES3				
5747	default_job	ACES3				
5746	default_job	ACES3				
5745	default_job	ACES3				
5744	default_job	ACES3	Ember	68736		
5743	default_job	ACES3	Ember	68720		
5742	default_job	GAMESS	Ember	65353	SCHEDULED	[illegible]

Fig. 2. (a) Post-processing of Abacus CAE interface, (b) stress contour from a post processed out of "Flanges in Contact" problem solved with Abaqus using DESSERT/SEAGrid.

Transmission Optimization Mechanism (MTOM) [32] is applied to transfer data to and from web server efficiently. Currently, the file transfer service is capable to transfer files up to many Gigabytes to accommodate jobs with large input and output files.

2.3. SEAGrid in production

This infrastructure is one of the most successful XSEDE science gateways with consistent quarterly use at various XSEDE HPC sites. This gateway is featured with "job based" control, where each user's SEAGrid allocation is tracked by the middleware. SEAGrid manages the job distribution and monitoring and collects job level usage data. Currently, the virtual organization serves more than 540 users allocated with 260 projects, and has supplied more than 2.3 million raw CPU Hours as represented in Fig. 6(a)–(d). The SEAGrid community has been growing steadily during the initial deployment phase and in production. The number of individual jobs that has also grown steadily during this period currently totals close to 60,000. Equally important to the number of jobs submitted is the overall usage resulting from those jobs. Fig. 6(d) shows a detailed XSEDE usage distribution that provides a way to estimate future resource requirements.

The project has enabled more than 80 publications, 50 conference presentations, 9 graduate theses and several presentations from the SEAGrid community (details at https://www.gridchem.org/papers/index.shtml). The users are supported through

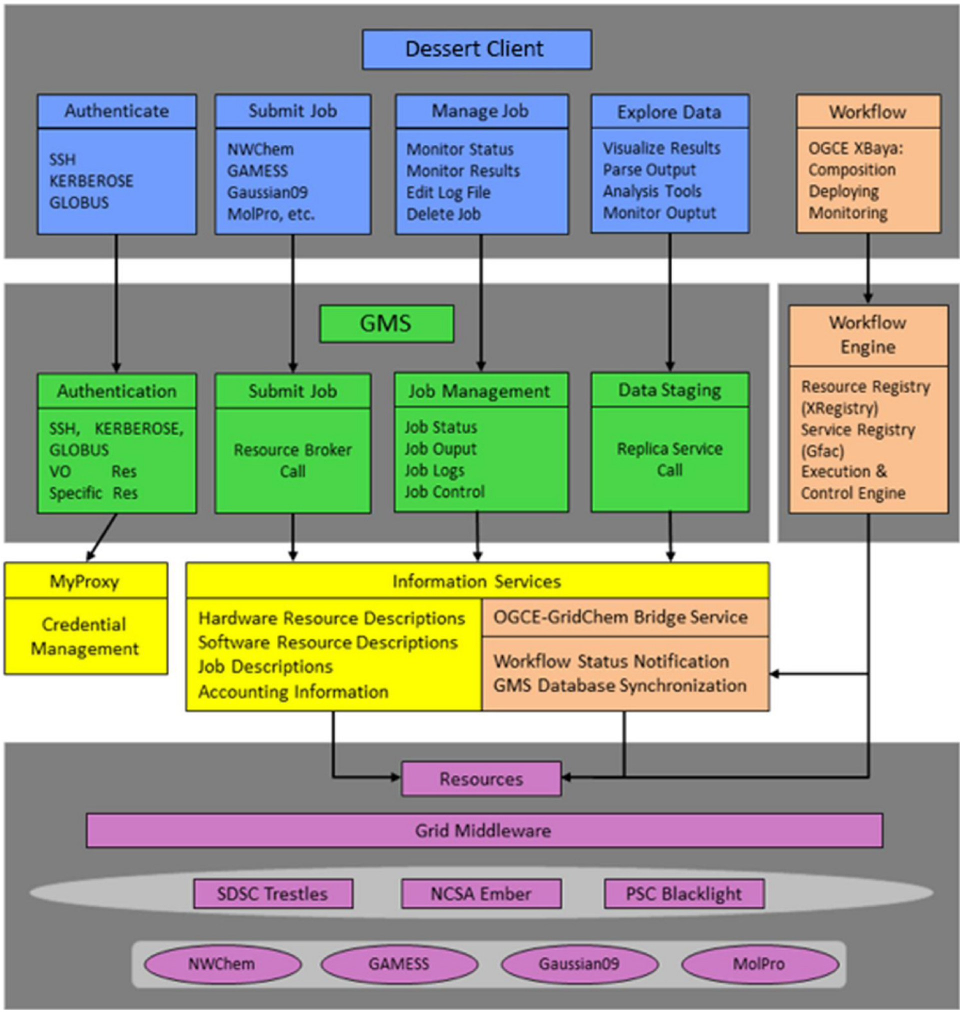
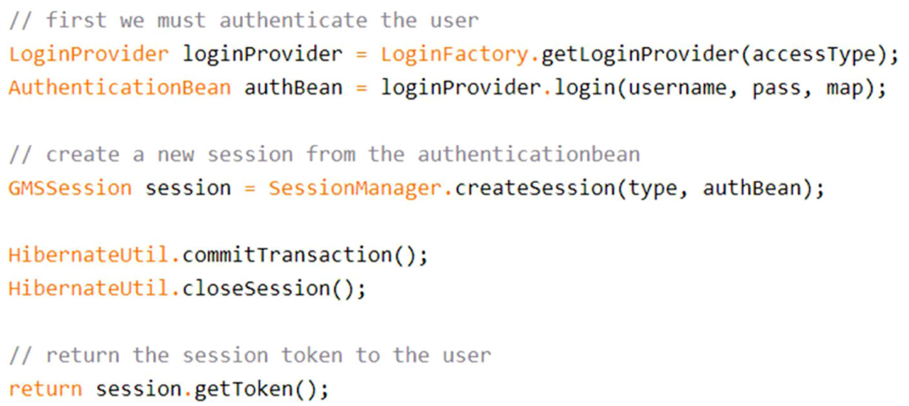


Fig. 3. Schematic architecture of Grid Middleware Services (GMS).

various methods such as a consulting portal to register issues (http://www.gridchem.org/consult), GridChem help e-mail (help@gridchem.org) and personal telephone contact. More than 450 tickets have so far been answered with the consulting portal alone. Significant events and user news such as system availability, service disruptions, new application deployment, and service enhancements are notified through mailing lists, the SEAGrid website, and an embedded RSS feed in the client.

3. Parametrization automation with workflow: ParamChem

ParamChem is another gateway in development based on the extension of GridChem project to provide automatic parametrization of molecular mechanics force-field parameters with integration of Apache Airavata workflow framework [13]. Similar but more limited infrastructures such as ForceBalance [30], FFToolKit [33]

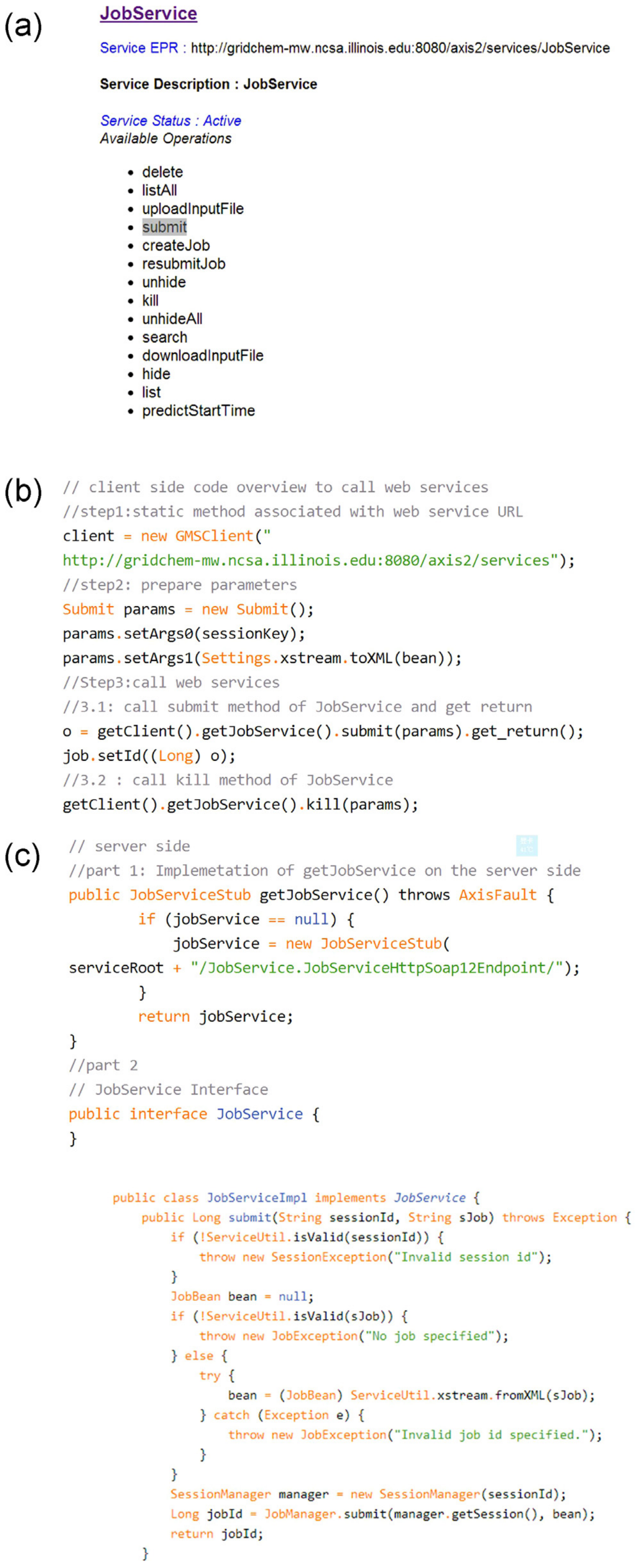

```
// first we must authenticate the user
LoginProvider loginProvider = LoginFactory.getLoginProvider(accessType);
AuthenticationBean authBean = loginProvider.login(username, pass, map);

// create a new session from the authenticationbean
GMSSession session = SessionManager.createSession(type, authBean);

HibernateUtil.commitTransaction();
HibernateUtil.closeSession();

// return the session token to the user
return session.getToken();
```

Fig. 4. Authentication with session management in GMS.



JobService

Service EPR : http://gridchem-mw.ncsa.illinois.edu:8080/axis2/services/JobService

Service Description : JobService

Service Status : Active
Available Operations

- delete
- listAll
- uploadInputFile
- submit
- createJob
- resubmitJob
- unhide
- kill
- unhideAll
- search
- downloadInputFile
- hide
- list
- predictStartTime

```
// client side code overview to call web services
//step1:static method associated with web service URL
client = new GMSClient("
http://gridchem-mw.ncsa.illinois.edu:8080/axis2/services");
//step2: prepare parameters
Submit params = new Submit();
params.setArgs0(sessionKey);
params.setArgs1(Settings.xstream.toXML(bean));
//Step3:call web services
//3.1: call submit method of JobService and get return
o = getClient().getJobService().submit(params).get_return();
job.setId((Long) o);
//3.2 : call kill method of JobService
getClient().getJobService().kill(params);
```

```
// server side
//part 1: Implemetation of getJobService on the server side
public JobServiceStub getJobService() throws AxisFault {
        if (jobService == null) {
            jobService = new JobServiceStub(
serviceRoot + "/JobService.JobServiceHttpSoap12Endpoint/");
        }
        return jobService;
}
//part 2
// JobService Interface
public interface JobService {
}


      public class JobServiceImpl implements JobService {
          public Long submit(String sessionId, String sJob) throws Exception {
              if (!ServiceUtil.isValid(sessionId)) {
                  throw new SessionException("Invalid session id");
              }
              JobBean bean = null;
              if (!ServiceUtil.isValid(sJob)) {
                  throw new JobException("No job specified");
              } else {
                  try {
                      bean = (JobBean) ServiceUtil.xstream.fromXML(sJob);
                  } catch (Exception e) {
                      throw new JobException("Invalid job id specified.");
                  }
              }
              SessionManager manager = new SessionManager(sessionId);
              Long jobId = JobManager.submit(manager.getSession(), bean);
              return jobId;
          }
```

Fig. 5. Example of JobService web service. (a) JobService with its EPR and 14 operations. (b) Steps from Client-end to call JobService. (c) Server-end implementation of submit operation of JobService.

and Gaamp [34] have been reported in literature. The force balance software is a set of python scripts for systematic force-field optimization that runs in command line execution only. FFToolKit provides software tools for setting up the optimization but no computational services while the Gaamp is restricted to XSEDE users only with no graphical set up tools. Paramchem infrastructure provides a combination of the tools and services in a comprehensive service organization. In order to separate the essential functions for better extensibility and stability of the services, ParamChem also adopts the three-tier architecture that includes a front-end desktop client, a middleware tier integrated with scientific workflow framework, and a back-end tier of computational resources as shown in Fig. 7.

3.1. Design of the ParamChem e-infrastructure

Molecular mechanics is an approximate computational model for representing chemical systems in atomic detail. It is presently the only computational method fast enough to routinely perform molecular dynamics simulations of large molecular systems on relevant time scales. Force-field consists of large numbers of so-called “force-field parameters” used in the context of a mathematical “potential energy function” in molecular mechanics to relate the atomic structure to the corresponding energy and forces. The quality of the model is determined by the accuracy of the energies and forces that result from the force-field parameters. While force-field parameters for bio-molecular systems are readily available, this is not always the case for small organic molecules because of the wide chemical space covered by organic chemistry. The ParamChem project is such an e-science infrastructure to provide a way to optimize these force-field parameters and is a result of collaboration with the parametrization domain experts driving the design and deployment.

CHARMM [35] is a set of force-fields used widely for molecular dynamics simulations. ParamChem has integrated CGenFF (CHARMM General Force-Field) [36], an extension of the CHARMM force-field to drug-like molecules into its current infrastructure. CGenFF provides a set of rules to assign an “initial guess” of the force-field parameters for specific molecules by analogy with a previously parameterized chemical moiety. The goodness of the initial guesses is measured by the penalty scores assigned by the CGenFF. The parameter optimization process requires a complex workflow involving setting up an iterative scheme starting with the initial guess, generating the reference data and extracting the right information to fit the parameters. There are multiple applications and complex logistics involved in maintaining data coherency in this process and requires careful orchestration by the researcher.

In order to alleviate these complex and labor-intensive process community centric software technologies such as the ParamChem project was conceived. This project supports this core aim by providing an automated set of tools for generating and optimizing force-field parameters. The present work focuses on optimization of a particular type of force-field parameters called “dihedral angle parameters” (or “dihedral parameters” in short). These dihedral parameters are most often in need of optimization because of their relative lack of transferability between different molecules. The automation process can be divided into two stages: firstly, generating all necessary user inputs on various client operating system platforms automatically. Secondly, feeding these inputs into middleware services that perform automatic execution of reference data generation and optimization programs on remote computing resources.

These automation tasks are solved with the ParamChem Infrastructure schematically shown in Fig. 7, which is briefly summarized below. The client provides a molecular editor and other functions to specify a molecule in the right format required for the atom typing and an initial guess generation. The initial guess function is provided in the middleware as a service and requires user authentication for security. Upon successful authentication, the middleware generates all the credentials for further processing. Once the initial guessed force-field parameters are generated, the

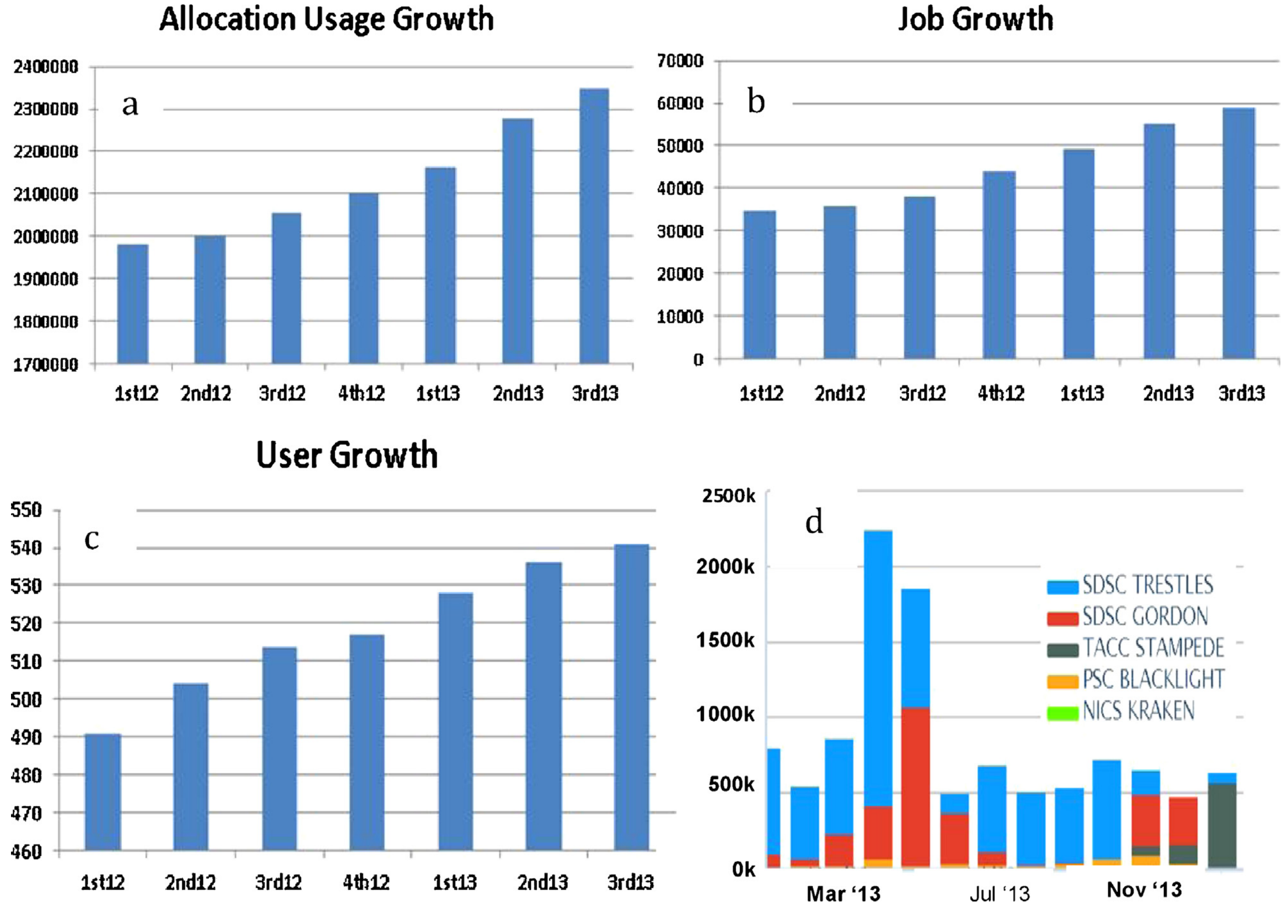


Fig. 6. Production usage of SEAGrid. (a) Allocation usage in SUs. (b) Jobs processed. (c) Number of users, and (d) XSEDE resources usage in SUs per month. The horizontal axis for (a–c) is the quarters in the year.

client provides a wizard interface to guide the user through the whole process of preparing all input files and configuring all the tasks in a pre-selected parametrization workflow registered in the Airavata workflow repositories. After the workflow is launched from the client, the Airavata workflow execution service will deploy it on the computational resources for running. The workflow executed in the Airavata workflow management system is identified with a workflow instance ID that is used as handle to monitor the workflow from the client. The client provides both text-based and graphical monitoring dashboard to monitor workflow execution. The workflows are executed in the XSEDE or ParamChem computational resources based on whether the reference ab-initio calculation results are injected into workflow or not. After the workflow is finished, all the data are compressed and returned to the client along with an automatic graphical display of goodness of the parametrization. In the meantime, the data are also archived on a mass storage device for later retrieval if needed.

3.2. ParamChem client: Paramberoo

The ParamChem client, Paramberoo, plays a central role in the infrastructure since it glues all essential parts together from ParamChem Middleware, Apache Airavata workflow management and user management. Paramberoo can be divided into four major modules, which will be discussed in details below:

1. Molecule visualization and its manipulation
2. Initial force filed parameter generation
3. Configuration and launching a parametrization workflow
4. Monitoring the workflow execution and post processing of the results

The first module, the molecular editor is based on the open-source cross-platform program Jamberoo and is used to visually construct or read the molecule from a file. The original program has been extensively modified in the area of molecule manipulation functions, such as multiple atoms selection/de-selection rules, for better user experience when performing the dihedral angle selections.

The initial parameter generation module interacts with the middleware server to generate initial guess of the force-field parameters. The client submits the molecule file in mol2 format to the ParamChem web server (https://www.paramchem.org), receives the initial guess parameter file and visually presents the parameters to the user automatically. Fig. 8 illustrates the Paramberoo interface showing the initially assigned parameters. The "All Parameters" table shows all the initial guesses of all the dihedral parameters and "Parameters around selected bond" shows the dihedral angle force-field parameters of the highlighted dihedral bonds in the molecule viewer based on a user selection.

The first column, the dihedral parameters, consists of four atom types that are assigned by the CGenFF web service program. The atoms highlighted in the viewer correspond to all possible physical atoms corresponding to the selected dihedral parameter row in the table. Note that there can be multiple physical atoms sharing the same atom types since they are assigned based on the chemical environment of the atom.

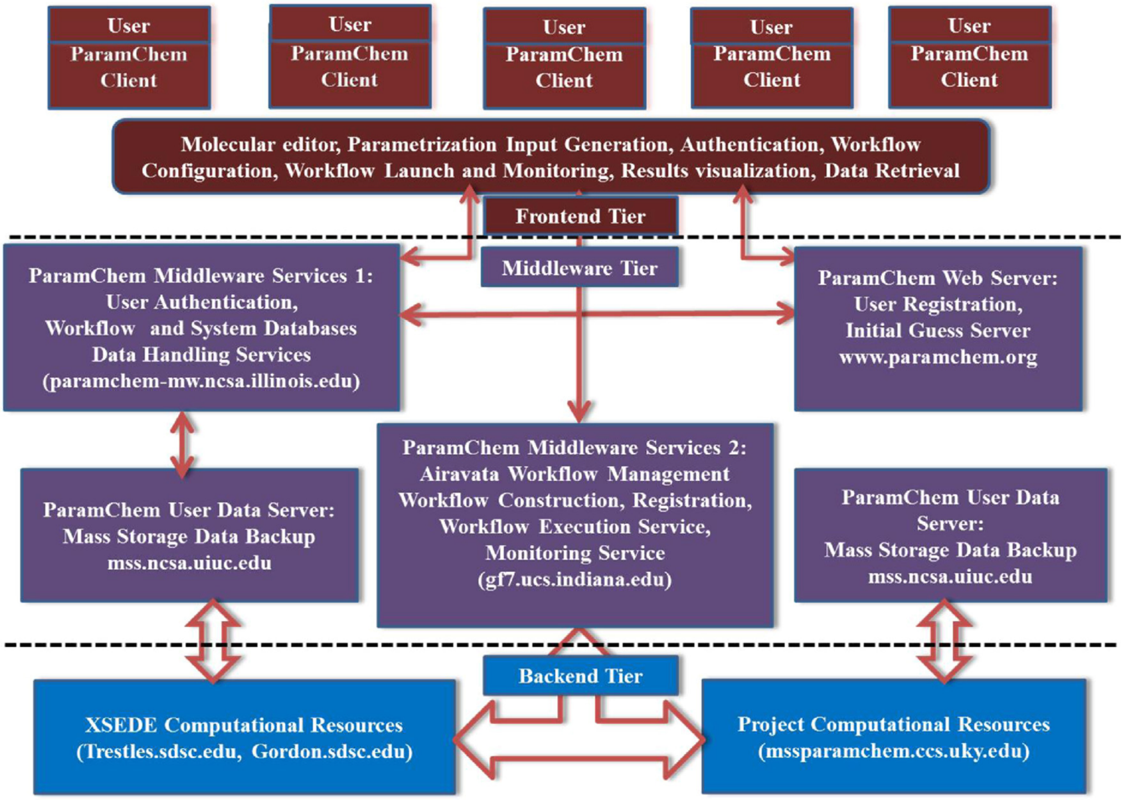
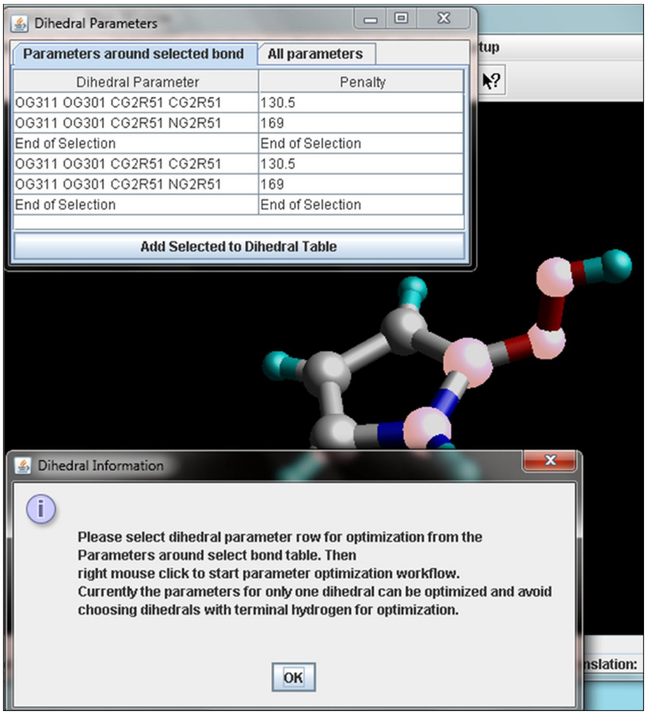

User ParamChem Client
User ParamChem Client
User ParamChem Client
User ParamChem Client
User ParamChem Client
Molecular editor, Parametrization Input Generation, Authentication, Workflow Configuration, Workflow Launch and Monitoring, Results visualization, Data Retrieval
Frontend Tier
Middleware Tier
ParamChem Middleware Services 1: User Authentication, Workflow and System Databases Data Handling Services (paramchem-mw.ncsa.illinois.edu)
ParamChem Web Server: User Registration, Initial Guess Server www.paramchem.org
ParamChem Middleware Services 2: Airavata Workflow Management Workflow Construction, Registration, Workflow Execution Service, Monitoring Service (gf7.ucs.indiana.edu)
ParamChem User Data Server: Mass Storage Data Backup mss.ncsa.uiuc.edu
ParamChem User Data Server: Mass Storage Data Backup mss.ncsa.uiuc.edu
Backend Tier
XSEDE Computational Resources (Trestles.sdsc.edu, Gordon.sdsc.edu)
Project Computational Resources (mssparamchem.ccs.uky.edu)

Fig. 7. Schematic of ParamChem multi-tiered architectures.

The second column contains the penalty scores, which are used to reflect the confidence quality of the initial guess parameters. A zero penalty score indicates that the parameters of the corresponding dihedral have already been optimized and available in the existing CGenFF parameter database at the ParamChem web server. The higher positive penalty score indicates less confident initial guess and the corresponding dihedral parameter should have higher priority for further optimization. The detailed physical explanations of the parameters are given in CGenFF paper. The client provide informative message dialogs with any new data generation step to guide the user with suggestions or instructions according to the CGenFF program or the parameter optimization process. At the end of this step, two necessary input files for the parametrization workflow are obtained: molecule structure in mol2 format and initial guess in CHARMM stream format. Other input files needed to fully configure and launch the workflow will be generated by the consequent steps.



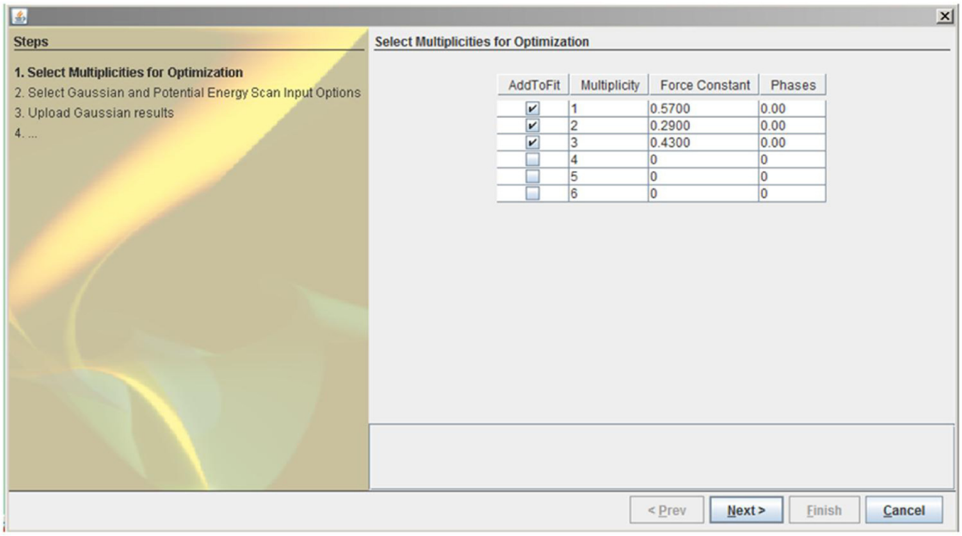
Dihedral Parameter	Penalty
OG311 OG301 CG2R51 CG2R51	130.5
OG311 OG301 CG2R51 NG2R51	169
End of Selection	End of Selection
OG311 OG301 CG2R51 CG2R51	130.5
OG311 OG301 CG2R51 NG2R51	169
End of Selection	End of Selection

Please select dihedral parameter row for optimization from the Parameters around select bond table. Then right mouse click to start parameter optimization workflow. Currently the parameters for only one dihedral can be optimized and avoid choosing dihedrals with terminal hydrogen for optimization.

Fig. 8. Paramberoo molecule editor with a sample molecule and a table of initial guessed parameters from CGenFF web service program.

In order to create the workflow, the user needs to generate five additional input files. Manually editing all these files is a tedious and error-prone process. For example, the order of file generation is important since the file contents are mutually dependent. In addition, the user needs to make delicate choices during this process in order to achieve consistent input data. In order to assist the user in this process, a ‘wizard’ guide is provided in Paramberoo. This will ensure the correct order of execution to generate the required input files. The wizard shown in Fig. 9 provides suggested initial (guess) values for advanced users to manipulate further. There are three dihedral force-field parameters to be optimized: multiplicity, amplitude (i.e. force constants) and phase. The default initial guesses are selected (checked) by default, which can be changed or modified by advanced users if needed.

The next step is to generate the target input file for ab-inito quantum chemistry program (Gaussian in this case) to compute the target data to be fitted against, to optimize the force-field parameters. As shown in Fig. 10, the users can tune the input parameters for dihedral potential energy scan calculation with the Gaussian program [15] such as dihedral angle size & stepsab initio theory level



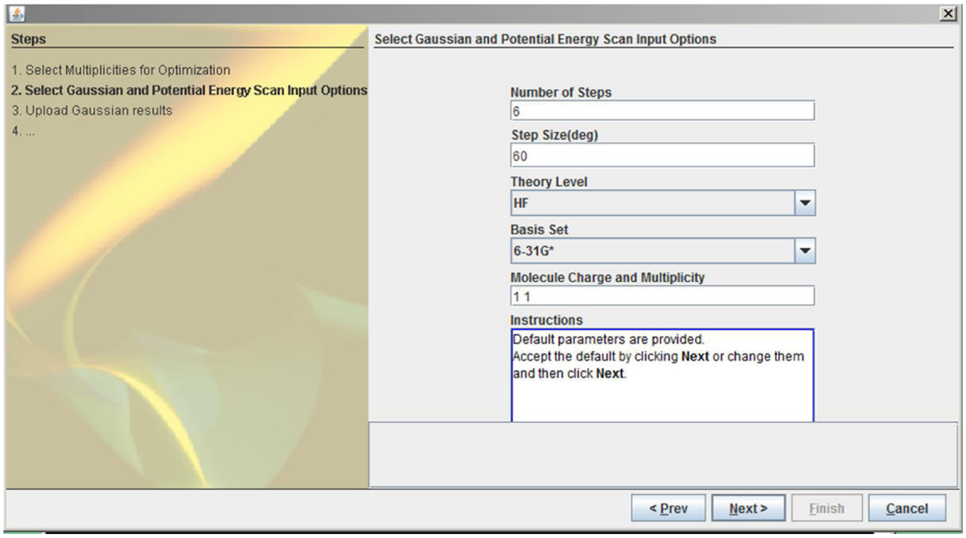
Steps

1. Select Multiplicities for Optimization
2. Select Gaussian and Potential Energy Scan Input Options
3. Upload Gaussian results
4. ...

Select Multiplicities for Optimization

AddToFit	Multiplicity	Force Constant	Phases
☑	1	0.5700	0.00
☑	2	0.2900	0.00
☑	3	0.4300	0.00
□	4	0	0
□	5	0	0
□	6	0	0

Fig. 9. The wizard step to tune three force-field parameters for dihedral angle parametrization.



Steps

1. Select Multiplicities for Optimization
2. Select Gaussian and Potential Energy Scan Input Options
3. Upload Gaussian results
4. ...

Select Gaussian and Potential Energy Scan Input Options

Number of Steps: 6

Step Size(deg): 60

Theory Level: HF

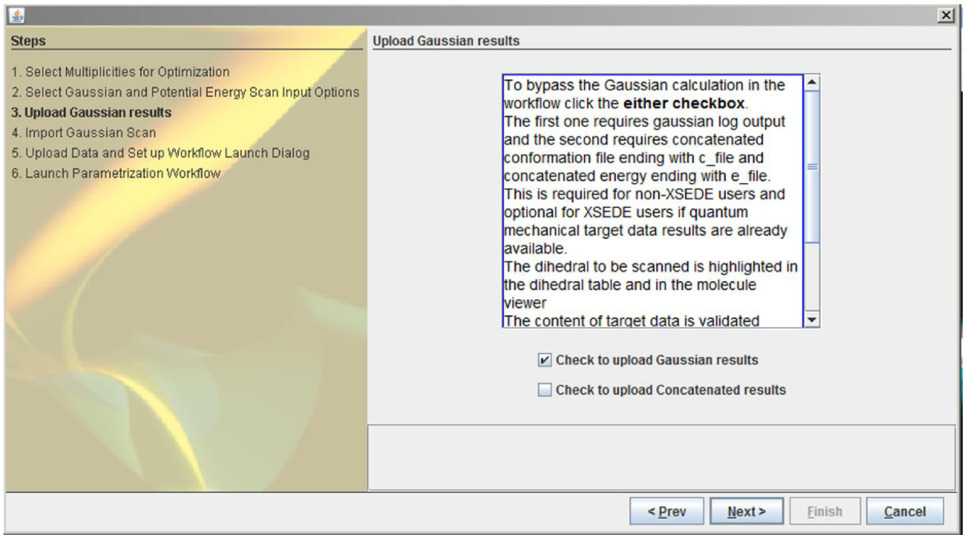
Basis Set: 6-31G*

Molecule Charge and Multiplicity: 1 1

Instructions

Default parameters are provided.
Accept the default by clicking **Next** or change them and then click **Next**.

Fig. 10. The wizard step to tune input parameters of Gaussian potential energy scan calculations.



Steps

1. Select Multiplicities for Optimization
2. Select Gaussian and Potential Energy Scan Input Options
3. Upload Gaussian results
4. Import Gaussian Scan
5. Upload Data and Set up Workflow Launch Dialog
6. Launch Parametrization Workflow

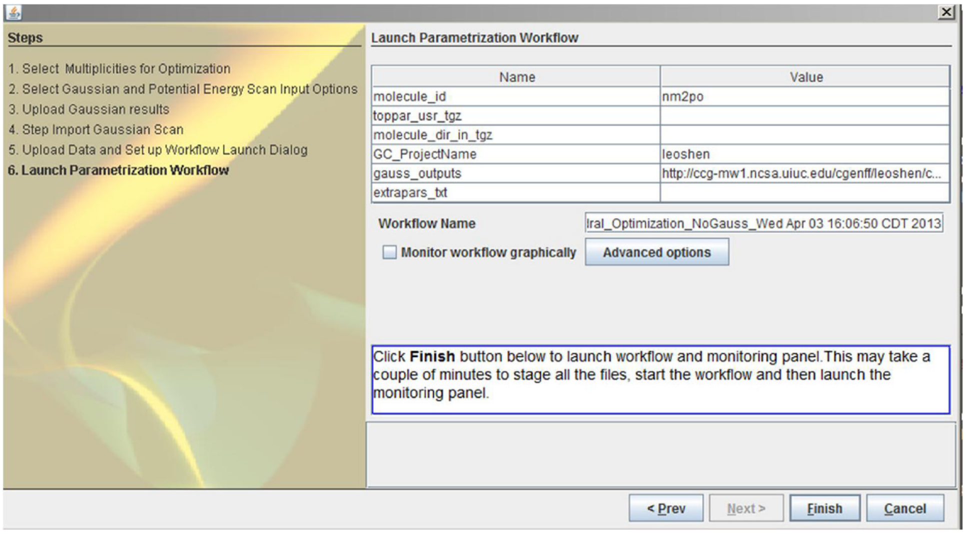
Upload Gaussian results

To bypass the Gaussian calculation in the workflow click the **either checkbox**.
The first one requires gaussian log output and the second requires concatenated conformation file ending with c_file and concatenated energy ending with e_file.
This is required for non-XSEDE users and optional for XSEDE users if quantum mechanical target data results are already available.
The dihedral to be scanned is highlighted in the dihedral table and in the molecule viewer
The content of target data is validated

☑ Check to upload Gaussian results

□ Check to upload Concatenated results

Fig. 11. The branch wizard step to inject Gaussian calculation results in original Gaussian log or concatenated format.



Steps

1. Select Multiplicities for Optimization
2. Select Gaussian and Potential Energy Scan Input Options
3. Upload Gaussian results
4. Step Import Gaussian Scan
5. Upload Data and Set up Workflow Launch Dialog
6. **Launch Parametrization Workflow**

Launch Parametrization Workflow

Name	Value
molecule_id	nm2po
toppar_usr_tgz	
molecule_dir_in_tgz	
GC_ProjectName	leoshen
gauss_outputs	http://ccg-mw1.ncsa.uiuc.edu/cgenff/leoshen/c...
extrapars_txt	

Workflow Name: Iral_Optimization_NoGauss_Wed Apr 03 16:06:50 CDT 2013

☐ Monitor workflow graphically

Click **Finish** button below to launch workflow and monitoring panel.This may take a couple of minutes to stage all the files, start the workflow and then launch the monitoring panel.

Fig. 12. The wizard step to configure Apache Airavata workflow and Launch it.

and basis set, required by the program. Since ParamChem infrastructure provides support for worldwide users, many users may not have access to Gaussian or XSEDE resources. This is because the XSEDE resources and software provided through ParamChem project are nominally restricted to US academic investigators, while the initial guess generation and fitting and validation services are available to worldwide users. For these non-XSEDE users and those who have pre computed quantum chemical reference data, a way to ingest the data is provided in the next step. As depicted in Fig. 11, the wizard provides two choices to ingest locally generated reference data. For XSEDE users, the interface provides, in addition, a way to generate the ab-initio reference data on XSEDE resources based on the input files generated in previous wizard steps. All users are provided with a way to select and upload their existing reference data either in Gaussian output file format or as concatenated text files containing the energy and the geometry at each of the dihedral scan angles selected in the previous step. Non-XSEDE users have to generate the reference data elsewhere and can only choose to upload that data to proceed to the next step.

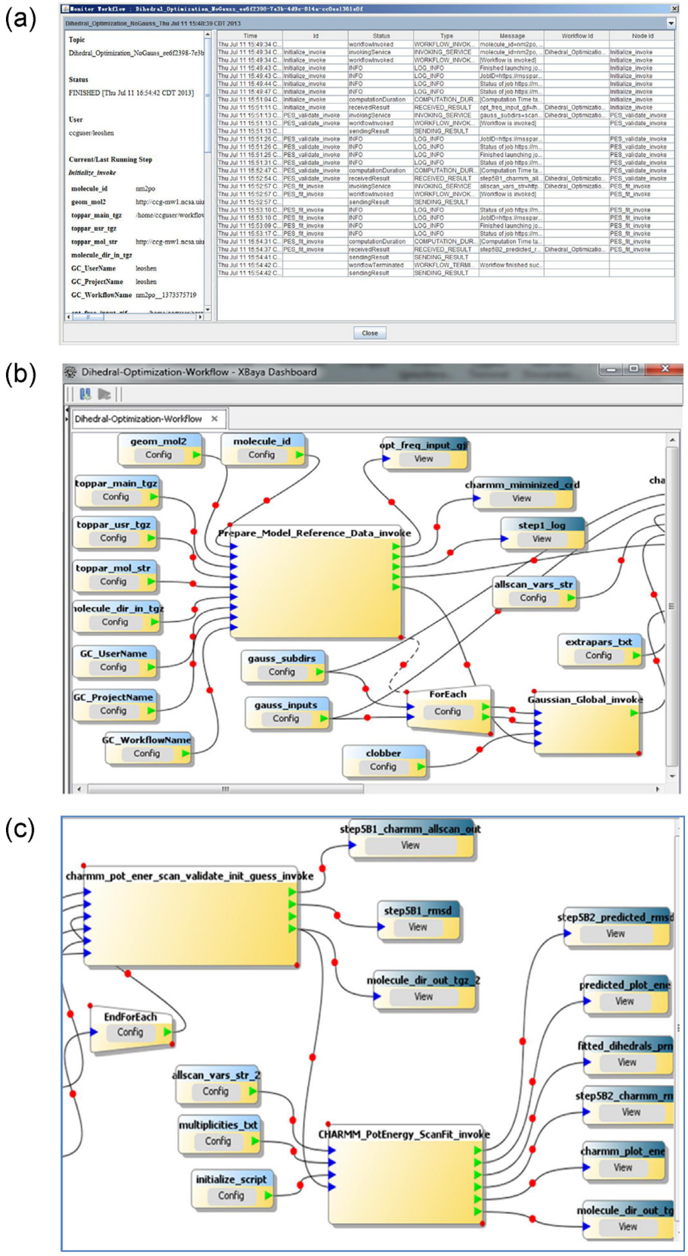
'Under the hood', Paramberoo client will configure separate workflows based on the user's data choices, which will either run on XSEDE or Non-XSEDE computing resources. Finally, the users are guided to the final step to perform common configurations of the Apache Airavata workflow and to launch it by clicking the "Finish" button as shown in Fig. 12.

The last module in the Paramberoo consists of the workflow monitoring and plotting the result when the workflow finishes successfully. The Apache Airavata workflow can be monitored either in a text-based format or graphically as shown in Fig. 13. The text-based workflow monitor is intended to monitor long running workflows (such as those involving reference data generation step) as the user is able to close the monitor window and restart it later to retrieve it from the Airavata workflow management service using the drop-down list on the top. The text-based panel shown in Fig. 13a contains the topic ID and the input/output URLs related to the task block currently running on the left panel and the dynamic log information of specific components on the right panel. A double click on a specific log message line will provide further information regarding this step in a new message window. The graphic panel shown in Fig. 13b and c displays detailed task blocks and associated inputs and outputs in a graphical way. The complete workflow is split into Fig. 13b and c for clarity and Fig. 13c should be viewed as continuing on the right side of Fig. 13b. The colors of the task blocks change as the workflow makes progress in execution. Initially all blocks are colored yellow. Once a task block starts to run, it turns green. If the task block ends successfully, it turns gray. Otherwise, it turns red, which indicates failure of the task block. The construction and execution order of the workflow blocks will be discussed in detail in the next section.

The Apache Airavata Workflow middleware is able to publish both 'push' and 'pull' workflow messages. If the Paramberoo client is kept open during the workflow execution process, the workflow messages are pushed to the client. Otherwise, the client can pull historic workflow messages according to the topic ID when user selects previously launched workflow ID from the drop-down list on the top of the monitor panel. Either way, Paramberoo can automatically download the output data from the ParamChem middleware to the local machine as a compressed file, which is transferred to the middleware by the Apache Airavata workflow. For a successful run, the dihedral parameter optimization data are plotted automatically as shown in Fig. 14. Users can check the status of other workflows by manually selecting them from the drop-down list on the top. The path to the local directory of the result data is provided in the bottom of interface for the user to navigate and analyze the results further.

3.3. ParamChem middleware integration with workflow

The ParamChem middleware incorporates two sets of services: the first set provides essential services related to user and data management and the second focuses on the scientific workflow management. The implementations of the first set of middleware services have been discussed in Section 2.2. In addition to services such as user authentication and data management, a new service for the CGenFF initial guess parameter generation is added in this set. The ParamChem middleware and CGenFF are deployed on separate dedicated servers to increase the stability. The implemented Apache Axis2 web services (JobService and FileService) are used to handle the job submission and the file transfer from the client to the CGenFF server. The client submits the molecule file in the mol2 format to the CGenFF server through the middleware server. After generating the initial guess of the parameters, the initial guess 'stream file' is retrieved by the client and visually presented to user as shown in Fig. 8.


Monitor Workflow : Dihedral_Optimization_NoGauss_ee6f2398-7e3b-4d9c-814a-cc0ee1361e0f

Dihedral_Optimization_NoGauss_Thu Jul 11 15:48:39 CDT 2013

Topic

Dihedral_Optimization_NoGauss_ee6f2398-7e3b

Status

FINISHED [Thu Jul 11 16:54:42 CDT 2013]

User

ccguser/leoshen

Current/Last Running Step

Initialize_invoke

molecule_id	nm2po
geom_mol2	http://ccg-mw1.ncsa.uiu
toppar_main_tgz	/home/ccguser/workflov
toppar_usr_tgz	
toppar_mol_str	http://ccg-mw1.ncsa.uiu
molecule_dir_in_tgz	
GC_UserName	leoshen
GC_ProjectName	leoshen
GC_WorkflowName	nm2po__1373575719

Time	Id	Status	Type	Message	Workflow Id	Node Id
Thu Jul 11 15:49:34 C...		workflowInvoked	WORKFLOW_INVOK...	molecule_id=nm2po, ...		
Thu Jul 11 15:49:34 C...	Initialize_invoke	invokingService	INVOKING_SERVICE	molecule_id=nm2po, ...	Dihedral_Optimizatio...	Initialize_invoke
Thu Jul 11 15:49:34 C...	Initialize_invoke	workflowInvoked	WORKFLOW_INVOK...	[Workflow is invoked]		Initialize_invoke
Thu Jul 11 15:49:43 C...	Initialize_invoke	INFO	LOG_INFO	Finished launching jo...		Initialize_invoke
Thu Jul 11 15:49:43 C...	Initialize_invoke	INFO	LOG_INFO	JobID=https://msspar...		Initialize_invoke
Thu Jul 11 15:49:44 C...	Initialize_invoke	INFO	LOG_INFO	Status of job https://m...		Initialize_invoke
Thu Jul 11 15:49:47 C...	Initialize_invoke	INFO	LOG_INFO	Status of job https://m...		Initialize_invoke
Thu Jul 11 15:51:04 C...	Initialize_invoke	computationDuration	COMPUTATION_DUR...	[Computation Time ta...		Initialize_invoke
Thu Jul 11 15:51:11 C...	Initialize_invoke	receivedResult	RECEIVED_RESULT	opt_freq_input_gjf=/h...	Dihedral_Optimizatio...	Initialize_invoke
Thu Jul 11 15:51:13 C...	PES_validate_invoke	invokingService	INVOKING_SERVICE	gauss_subdirs=scan...	Dihedral_Optimizatio...	PES_validate_invoke
Thu Jul 11 15:51:13 C...	PES_validate_invoke	workflowInvoked	WORKFLOW_INVOK...	[Workflow is invoked]		PES_validate_invoke
Thu Jul 11 15:51:13 C...		sendingResult	SENDING_RESULT			
Thu Jul 11 15:51:26 C...	PES_validate_invoke	INFO	LOG_INFO	JobID=https://msspar...		PES_validate_invoke
Thu Jul 11 15:51:26 C...	PES_validate_invoke	INFO	LOG_INFO	Status of job https://m...		PES_validate_invoke
Thu Jul 11 15:51:25 C...	PES_validate_invoke	INFO	LOG_INFO	Finished launching jo...		PES_validate_invoke
Thu Jul 11 15:51:31 C...	PES_validate_invoke	INFO	LOG_INFO	Status of job https://m...		PES_validate_invoke
Thu Jul 11 15:52:47 C...	PES_validate_invoke	computationDuration	COMPUTATION_DUR...	[Computation Time ta...		PES_validate_invoke
Thu Jul 11 15:52:54 C...	PES_validate_invoke	receivedResult	RECEIVED_RESULT	step5B1_charmm_all...	Dihedral_Optimizatio...	PES_validate_invoke
Thu Jul 11 15:52:57 C...	PES_fit_invoke	invokingService	INVOKING_SERVICE	allscan_vars_str=http...	Dihedral_Optimizatio...	PES_fit_invoke
Thu Jul 11 15:52:57 C...	PES_fit_invoke	workflowInvoked	WORKFLOW_INVOK...	[Workflow is invoked]		PES_fit_invoke
Thu Jul 11 15:52:57 C...		sendingResult	SENDING_RESULT			
Thu Jul 11 15:53:10 C...	PES_fit_invoke	INFO	LOG_INFO	Status of job https://m...		PES_fit_invoke
Thu Jul 11 15:53:10 C...	PES_fit_invoke	INFO	LOG_INFO	JobID=https://msspar...		PES_fit_invoke
Thu Jul 11 15:53:09 C...	PES_fit_invoke	INFO	LOG_INFO	Finished launching jo...		PES_fit_invoke
Thu Jul 11 15:53:17 C...	PES_fit_invoke	INFO	LOG_INFO	Status of job https://m...		PES_fit_invoke
Thu Jul 11 15:54:31 C...	PES_fit_invoke	computationDuration	COMPUTATION_DUR...	[Computation Time ta...		PES_fit_invoke
Thu Jul 11 15:54:37 C...	PES_fit_invoke	receivedResult	RECEIVED_RESULT	step5B2_predicted_r...	Dihedral_Optimizatio...	PES_fit_invoke
Thu Jul 11 15:54:41 C...		sendingResult	SENDING_RESULT			
Thu Jul 11 15:54:42 C...		workflowTerminated	WORKFLOW_TERMI...	Workflow finished suc...		
Thu Jul 11 15:54:42 C...		sendingResult	SENDING_RESULT			

Fig. 13. (a) Text-based workflow monitoring. (b and c) Graph-based workflow monitoring with four building blocks.

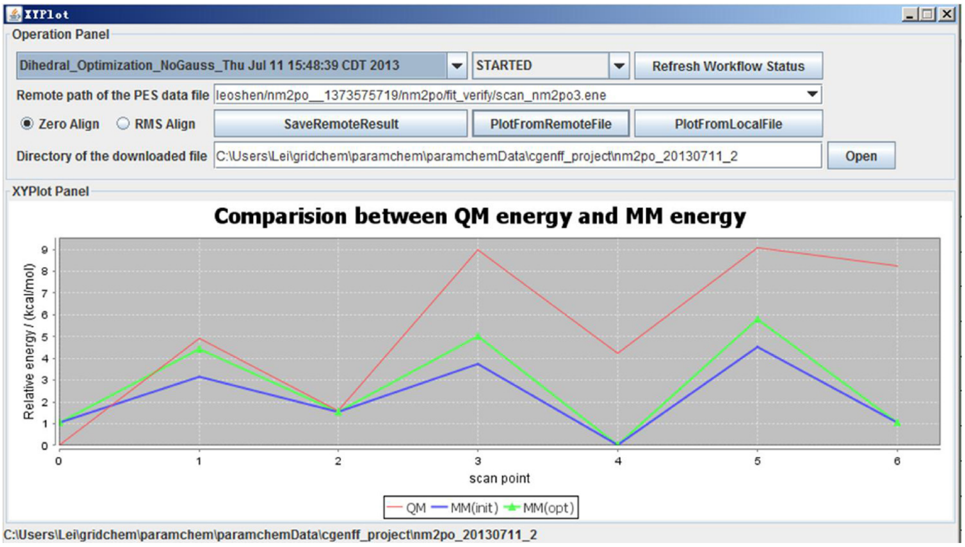
Another group of important middleware services is concerned with workflow execution and management. The Apache Airavata workflow services provide modularized software for workflow management. The detailed descriptions of the package can be found in [13] and here we focus on the descriptions of modules that have been applied in building ParamChem infrastructure and how they are integrated into ParamChem middleware services. Fig. 15 illustrates a schematic architecture and major components of Apache Airavata system. The features of the Airavata components incorporated in ParamChem middleware services are summarized as follows.

- Apache Airavata Application Programming Interface (API) is written in Java, which provides interaction points for the gateway developers to interact with the toolkit. The Airavata graphical interface (XBaya) is used to construct workflows and monitor their execution graphically by either gateway developer or end users of workflows.
- Generic Application Factory (GFac) works as an application abstraction layer to wrap the command line to run an application on remote computational resources as SOAP web service and corresponding WSDL web service descriptions are made available.
- Workflow interpreter, an extremely dynamic and interactive workflow enactment engine designed to enable application steering and interactive needs is used for execution.
- Registry API is implemented on top of the Apache Derby database, which interacts with other components to put and get data, such as workflow information and execution status.
- In addition to the basic web service specifications like SOAP and WSDL, Apache Airavata also support WS-Messenger (an asynchronous publish/subscribe based messaging system), which acts as a nervous system of Airavata to manage communications of the user and the system, and between the GFAC, the Airavata graphic interface, and the workflow interpreter.

The detailed steps to compose the dihedral parameter optimization workflow that exemplify the workflow system integration are as follows:

1. Command line scripts to drive the executions of computational software (such as CHARMM and Gaussian) and file managements are deployed on remote computational resources.
2. The descriptions of these command line scripts such as input/output parameters and remote deployment information are registered with the workflow registry database using the Airavata XBaya graphical interface. Once these scripts are registered, the Airavata GFac can manage the workflow process including input file staging, job submission, execution, and monitoring.
3. The registered scripts saved in workflow registry database (so called applications) are applied as reusable building blocks for a complete scientific workflow. Applying the drag and drop features of Airavata XBaya interface a workflow graph is created visually as shown in Fig. 13b and c.
4. Inside Apache Airavata system, a workflow graph is represented as a Directed Acyclic Graphs (DAG). During its execution, the workflow interpreter loads the composed workflow into memory and creates a new DAG object. This DAG object can be dynamically configured by the client through the Airavata API and launched by calling the workflow interpreter.
5. At the time of execution, the client code also interacts with the WS-messenger bus to monitor the execution of workflow in real time through the Airavata API.

In order to construct the workflow graphically with Apache Airavata, it is important to understand the execution order of the building blocks. The general rule of thumb is: if all input parameters of the block are available, the block can go to the execution stage, unless it is connected to another block by a *dashed line*. This dashed line indicates that it will not execute until the second connected block is finished executing. Taking the workflow in Fig. 13b as an example, the Prepared_Model_Ref_Data_Invoke block is executed at first since all inputs of this block are provided by the ParamChem client automatically when the user is finished with the wizard guide interface. The next block, the Gaussian_Global_Invoke, will not start its execution until the Prepared_Model_Ref_Data_Invoke block completes execution itself since it is connected with Prepared_Model_Ref_Data_Invoke block with a dashed line. Since there can be many Gaussian dihedral scan calculations and these QM calculations can be expensive, the Gaussian_Global_Invoke block is put inside the built-in “ForEach loop” control block in order to be executed in parallel on multi-core or multiple-node computing


XYPlot

Operation Panel

Dihedral_Optimization_NoGauss_Thu Jul 11 15:48:39 CDT 2013 | STARTED | Refresh Workflow Status

Remote path of the PES data file: leoshen/nm2po__1373575719/nm2po/fit_verify/scan_nm2po3.ene

● Zero Align ○ RMS Align | SaveRemoteResult | PlotFromRemoteFile | PlotFromLocalFile

Directory of the downloaded file: C:\Users\Lei\gridchem\paramchem\paramchemData\cgenff_project\nm2po_20130711_2 | Open

XYPlot Panel

Comparision between QM energy and MM energy
Relative energy / (kcal/mol)
0
1
2
3
4
5
6
7
8
9
scan point
0
1
2
3
4
5
6
QM
MM(init)
MM(opt)

C:\Users\Lei\gridchem\paramchem\paramchemData\cgenff_project\nm2po_20130711_2

Fig. 14. Panel to monitor the workflow status and display parameter optimization results.

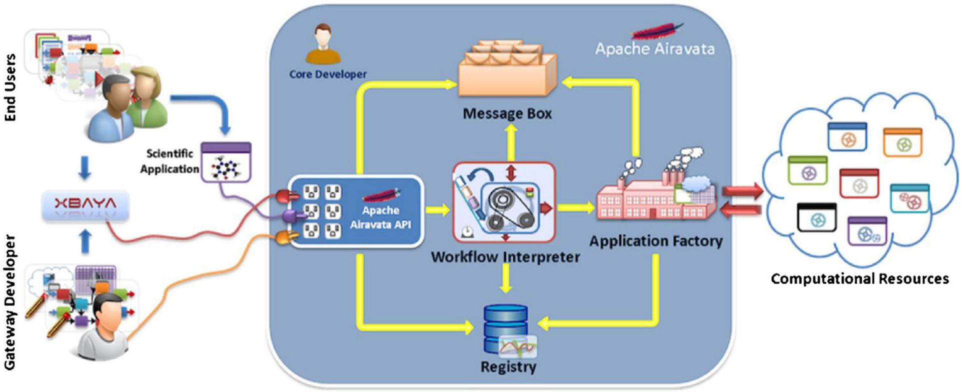


Fig. 15. Architecture and major components of Apache Airavata toolkit [29].

resources. The output of the Gaussian_Global_Invoke block is forwarded to the next block, since the ab-initio calculation results are to be used as target data to be fitted against CHARMM derived data to obtain the optimized force-field parameters. The following two CHARMM calculation blocks will run calculations to optimize the force-field parameters against the ab-initio reference data. The first CHARMM block will run the CHARMM calculation using the initial guess parameter set, while the second CHARMM block will perform a series of CHARMM calculations with a simulated annealing Monte Carlo (MCSA) algorithm to find the best set of parameters to minimize the error between CHARMM results and ab-initio calculation results. All input and output files for each running block of the whole workflow are automatically archived into both middleware server for immediate download and dedicated mass storage server for long-term storage and retrieval.

4. Summary, current status and future work

We have demonstrated that e-science infrastructures have large potential to help communities of users to access resource for science and engineering disciplines. An infrastructure can provide a basis for additional advanced services. SEAGrid is a production e-science infrastructure that is widely used. The open source software base for these projects will be available via Apache Airavata project in collaboration with the OGCE group in Indiana. Toward the goal of deploying multi-disciplinary applications, some engineering applications aligned with chemical parameter development and usage will be deployed. Reactive flow problem such as combustion modeling that requires coupling chemical energetic and kinetic information with fluid dynamics is an area of current interest. OpenFOAM [37] and Fluent [34] are two packages that can potentially be supported in the SEAGrid virtual organization in the near future. The expectation of increased usage of national computational resources in SEAGrid is realized with the new deployments. Due to this increase, an increased productivity in terms of publications is now anticipated.

In the middleware area, work is underway to provide job wait time estimate and eventually runtime estimates based on Karnak Prediction Engine [38]. This will aid the smart scheduling to be implemented. A full featured integration of Apache Airavata workflow suite is planned that will enhance the current script based mechanism for the high throughput computational services.

ParamChem initial guess server is deployed and more than thousand users utilize its services worldwide. Fully functional Paramberoo client for automated parametrization of molecular force-fields is in alpha testing and will be released shortly. Simultaneous optimization of multiple dihedral parameters is in the implementation stage. Additional bond, angle and non-bonded parameter optimization will be included in a future release with integration general AMBER force-field (GAFF) [39].

Acknowledgements

The work presented is funded in part by National Science Foundation grants Chem-0823198 (Extensible Cyberenvironments for Empirical and Semi-empirical Hamiltonian Parameter Optimization and Dissemination), SCI-0503697 (ETF Grid Infrastructure Group: Providing System Management and Integration for the TeraGrid and OCI-1032742 (Open Gateway Computing Environments – Tools for Cyberinfrastructure-Enabled Science and Education). Previously it was a product of NSF supported NMI Program under Award #04-38312. We acknowledge our colleagues in the respective grants, particularly Rion Dooley for coding the original Axis2 services under TeraGrid ASTA project for the GridChem gateways grant and Suresh Marru of OGCE collaboration with XBaya integration. ParamChem project is a multi-institution collaboration and significant contributions from the team members Alex MacKerell, Kenno Vanommeslaeghe, Narendra Polani, Michael Sheetz, Vikram Gazula, Remya Puthanthodiyil, and Adrian Roitberg are acknowledged. Work of previous post-doctoral students, Jayeeta Ghosh and Nikhil Singh contributed to the initial part of workflow infrastructure setup and the OGCE Airavata members Suresh Marru, Chaturi Wimalasena, Saminda Wijeratne, Lahiru Gunathilake, and Marlon Pierce are gratefully acknowledged for help with Airavata Workflow deployment in Paramchem.

References

[1] T. Hey, A.E. Trefethen, Cyberinfrastructure for e-science, Science (New York, N.Y.) 308 (2005) 817–821.

[2] T. Soddemann, Science gateways to DEISA: user requirements, technologies, and the material sciences and plasma physics gateway, Concurrency and Computation: Practice and Experience 19 (2007) 839–850.

[3] R. Dooley, K. Milfeld, C. Guiang, S. Pamidighantam, G. Allen, From proposal to production: lessons learned developing the computational chemistry grid cyberinfrastructure, Journal of Grid Computing 4 (2006) 195–208.

[4] N. Wilkins-Diehr, A history of the TeraGrid science gateway program: a personal view, in: Proceedings of the 2011 ACM workshop on Gateway computing environments GCE 11, 2011, pp. 1–12.

[5] T.N. Truong, M. Nayak, H.H. Huynh, T. Cook, P. Mahajan, L.T. Tran, J. Bharath, S. Jain, H.B. Pham, C. Boonyasiriwat, N. Nguyen, E. Andersen, Y. Kim, S. Choe, J. Choi, T.E. Cheatham III, J.C. Facelli, Computational science and engineering online1; (CSE-Online): a cyber-infrastructure for scientific computing, Journal of Chemical Information and Modeling 46 (2006) 971–984.

[6] N. Wilkins-Diehr, D. Gannon, G. Klimeck, S. Oster, S. Pamidighantam, TeraGrid gateways and their impact on science, Computer 41 (2008) 32–41.

[7] R. Guha, K. Gilbert, G. Fox, M. Pierce, D. Wild, H. Yuan, Advances in cheminformatics methodologies and infrastructure to support the data mining of large, heterogeneous chemical datasets, Current Computer – Aided Drug Design 6 (2010) 50–67.

[8] S.J. Coles, J.G. Frey, M.B. Hursthouse, M.E. Light, A.J. Milsted, L.A. Carr, D. DeRoure, C.J. Gutteridge, H.R. Mills, K.E. Meacham, M. Surridge, E. Lyon, R. Heery, M. Duke, M. Day, An e-science environment for service crystallography from submission to dissemination, Journal of Chemical Information and Modeling 46 (2006) 1006–1016.

[9] J.P. Greenberg, S. Mock, K. Bhatia, M. Katz, G. Bruno, F. Sacerdoti, P. Papadopoulos, K.K. Baldridge, Incorporation of middleware and grid technologies to enhance usability in computational chemistry applications, Future Generation Computer Systems 21 (2005) 3–10.

[10] R. Dooley, Recipes for success in new science gateway development, in: R. Barbera, G. Andronico, G.L. Rocca (Eds.), Proceedings of the International Workshop on Science Gateways, Consorzio COMETA, Catania, Italy, 2010, pp. 49–53.

[11] G. Allen, R. Dooley, S. Pamidighantam, Computational chemistry grid: production cyberinfrastructure for computational chemistry, in: Proceedings of the 13th Annual Mardi Gras Conference, Baton Rouge, LA, 2005.

[12] K. Milfeld, C. Guiang, S. Pamidighantam, J. Giuliani, Cluster Computing Through an Application-oriented Computational Chemistry Grid, 6th LCI Conference Linux Clusters: The HPC Revolution, Chapel Hill, North Carolina, 2005.

[13] S. Marru, C. Herath, P. Tangchaisin, M. Pierce, C. Mattmann, R. Singh, T. Gunarathne, E. Chinthaka, R. Gardler, A. Slominski, A. Douma, S. Perera, L. Gunathilake, S. Weerawarana, Apache airavata: a framework for distributed applications and computational workflows, in: Proceedings of the 2011 ACM workshop on Gateway computing environments, 2011, pp. 21–28.

[14] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, Journal of Computational Physics 117 (1995) 1–19.

[15] J.A. Montgomery Jr., J.E. Daniels, J. Cioslowski, M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G.A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H.P. Hratchian, A.F. Izmaylov, J. Bloino, G. Zheng, J.L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J.A. Peralta, F. Ogliaro, M. Bearpark, J.J. Heyd, E. Brothers, K.N. Kudin, V.N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J.C. Burant, S.S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J.M. Millam, M. Klene, J.E. Knox, J.B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R.E. Stratmann, O. Yazyev, A.J. Austin, R. Cammi, C. Pomelli, J.W. Ochterski, R.L. Martin, K. Morokuma, V.G. Zakrzewski, G.A. Voth, P. Salvador, J.J. Dannenberg, S. Dapprich, A.D. Daniels, Ö. Farkas, J.B. Foresman, J.V. Ortiz, C.T. Wallingford, J. Fox, Gaussian 09, Revision D, Gaussian Inc., 2009.

[16] M.W. Schmidt, K.K. Baldridge, J.A. Boatz, S.T. Elbert, M.S. Gordon, J.H. Jensen, S. Koseki, N. Matsunaga, S. Su, K.A. Nguyen, T.L. Windus, J.A. Dupuis, Montgomery, general atomic and molecular electronic structure system, Journal of Computational Chemistry 14 (1993) 1347–1363.

[17] M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T.P. Straatsma, H.J.J. van Dam, D. Wang, J. Nieplocha, E. Apra, T.L. Windus, W.A. de Jong, NWChem: a comprehensive and scalable open-source solution for large scale molecular simulations, Computer Physics Communications 181 (2010) 1477–1489.

[18] H.J. Werner, P.J. Knowles, R. Lindh, F.R. Manby, M. Schutz, MOLPRO, Version. A Package of ab initio Programs, 2009, http://www.molpro.net

[19] Materials Studio version 5.0 An integrated Multi-Scale Modeling Environment, 2013, http://accelrys.com/products/materials-studio

[20] D.A. Case, T.A. Darden, I.T. Cheatham, C.L. Simmerling, J. Wang, R.E. Duke, R. Luo, R.C. Walker, W. Zhang, K.M. Merz, B. Roberts, S. Hayik, A. Roitberg, G. Seabra, J. Swails, A.W. Goetz, I. Wong, F. Paesani, J. Vanicek, R.M. Wolf, J. Liu, X. Wu, S.R. Brozell, T. Steinbrecher, H. Gohlke, Q. Cai, X. Ye, M. Hsieh, G. Cui, D.R. Roe, D.H. Mathews, M.G. Seetin, R. Salomon-Ferrer, V. Sagui, V. Babin, T. Luchko, S. Gusarov, I. Kolossváry, A. Kovalenko, P.A. Kollman, AMBER 12, Assisted Model Building with Energy Refinement, A Set of Force-Fields and a Package of Molecular Simulation Programs, 2014.

[21] J.C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R.D. Skeel, L. Kalé, K. Schulten, Scalable molecular dynamics with NAMD, Journal of Computational Chemistry 26 (2005) 1781–1802.

[22] H.J.C. Berendsen, D. van der Spoel, R. van Drunen, GROMACS: a message-passing parallel molecular dynamics implementation, Computer Physics Communications 91 (1995) 43–56.

[23] P.J. Flatau, B.T. Draine, Fast near field calculations in the discrete dipole approximation for regular rectilinear grids, Optics Express 20 (2012) 1247–1252.

[24] The Abaqus Unified FEA Product Suite with Solutions for Engineering Problems, Simulia Corp., Abaqus Dassault, Providence, RI, 2014, http://www.3ds.com/products-services/simulia/portfolio/abaqus/2014

[25] Abaqus/CAE Complete solution for Abaqus Finite Element Modeling, Visualization, and Process Automation, 2014, http://www.3ds.com/products-services/simulia/portfolio/abaqus/abaqus-portfolio/abaquscae/, 2014

[26] Hibernate an Object Relational Mapping Framework, Version 3.0, 2011, http://www.hibernate.org

[27] I. Foster, C. Kesselman, Globus: a metacomputing infrastructure toolkit, International Journal of High Performance Computing Applications 11 (1997) 115–128.

[28] R. Housley, W. Ford, T. Polk, D. Solo, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, 2002, http://tools.ietf.org/html/rfc3280

[29] E.A. Young, T.J. Hudson, Open Source Secure Socket Layer and Transport Layer Security, 2014, http://www.openssl.org

[30] L.-P. Wang, J. Chen, T.V. Voorhis, Systematic parametrization of polarizable force-fields from quantum chemistry data, Journal of Chemical Theory and Computation 9 (2013) 452–460.

[31] Apache Axis2 Web Services (SOAP/WSDL) Engine Version 1.5.1, 2010, http://axis.apache.org/axis2/java/core

[32] SOAP Message Transmission Optimization Mechanism (MTOM), 2005, http://www.w3.org/TR/soap12-mtom/

[33] C.G. Mayne, J. Saam, K. Schulten, E. Tajkhorshid, J.C. Gumbart, Rapid parameterization of small molecules using the force-field toolkit, Journal of Computational Chemistry 34 (2013) 2757–2770.

[34] L. Huang, L. Gaamp, General Automated Atomic Model Parameterization Gateway, 2013, http://gaamp.lcrc.anl.gov/index.html

[35] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, M. Karplus, CHARMM: a program for macromolecular energy, minimization, and dynamics calculations, Journal of Computational Chemistry 4 (1983) 187–217.

[36] K. Vanommeslaeghe, E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, A.D. MacKerell, CHARMM general force-field: a force-field for drug-like molecules compatible with the CHARMM

all-atom additive biological force-fields, Journal of Computational Chemistry 31 (2010) 671–690.
[37] OpenFOAM, a Free, Open Source CFD Software Package, 2013, http://www.openfoam.com
[38] W. Smith, Karnak Prediction Service to Predict Queue Wait Times, 2013, https://portal.futuregrid.org/projects/152
[39] A general AMBER force field (GAFF) for rational drug design, http://ambermd.org/antechamber/gaff.html

Dr. Ning Shen joined NCSA as a post-doctoral fellow after graduating with a Ph.D. in computational physics from Pennsylvania State University in 2011 focusing on solid state materials research. His experiences include multi-scale simulation studies of nanomaterials on HPC systems, parallel and distributed programming to develop scientific software for statistical data analysis and process automation. He is currently a post- doctoral scholar in Prof. Ward Thompson group at Kansas State University.

Ye Fan joined NCSA as a research programmer in 2011 in the persistent infrastructure division after receiving a M.S. in Computer science from Indiana University. His activities include distributed computing and software development.

Dr. Sudhakar Pamidighantam has been a consulting and research scientist for high performance computing and applications at NCSA for the last 16 years. He serves on XSEDE extended collaborative support teams and is a member of Science Gateways group. He received his Ph.D. from University of Alabama at Birmingham after spending couple of preparatory year at IISc Bangalore, after Graduating with an M.Sc. from University of Hyderabad, India. He has been developing and deploying production cyberinfrastructure for chemistry and computational biology communities supported by NSF. He deployed Chemviz (chemviz.ncsa.uiuc.edu) the chemistry educational portal with integration of NCSA Condor resources. He has developed GridChem cyber infrastructure (https://www.gridchem.org) and continues to serve high performance computational chemistry and molecular modeling research and education communities. His interests are in providing production quality e-infrastructure for multi-disciplinary research using scientific workflows and support chemistry domain scientists.