International Environmental Modelling and Software Society (iEMSs)
7th Intl. Congress on Env. Modelling and Software, San Diego, CA, USA,
Daniel P. Ames, Nigel W.T. Quinn and Andrea E. Rizzoli (Eds.)
http://www.iemss.org/society/index.php/iemss-2014-proceedings

Design and Implementation of a Web Service-Oriented Gateway to Facilitate Environmental Modeling using HPC Resources

Ahmet Artu Yıldırım , David Tarboton, Pabitra Dash and Dan Watson

Department of Computer Science, Utah State University, Logan, UT, USA (ahmetartu@aggiemail.usu.edu, dan.watson@usu.edu)

Utah Water Research Laboratory, Utah State University, Logan, UT, USA (dtarb@usu.edu, pabitra.dash@usu.edu)

Abstract: Environmental researchers, modelers, water managers, and users often require access to high-performance computing (HPC) resources for running data and computationally intensive models without being an HPC expert. To address these challenges, we have developed a gateway to HPC storage and computational resources. This gateway software (that we have named HydroGate) is a CGI based REST web service that takes input via HTTP methods then transmits commands to the HPC system using SSH. The gateway abstracts away many details and complexities involved in the use of HPC systems including authentication, authorization, data and job management - transferring the data back and forth as well as creation, monitoring and scheduling of the jobs without installing any third-party software on the HPC systems. We demonstrate the use of these web services to provide a web based interface to the TauDEM hydrologic terrain analysis tools. This prototype implementation allows a user with a HPC account and resource allotment to upload a digital elevation model (DEM), execute TauDEM functions on this DEM to delineate a watershed and derive other TauDEM products. The TauDEM tools are executed on the HPC system taking advantage of parallel methods. The user can then prepare (offline) the inputs for a model, such as the Utah Energy Balance (UEB) Snowmelt model, for a delineated watershed and submit the UEB input files and execute the model on the HPC system. The contribution of this study is the realization of the gateway service exposing an interface to the client applications that require access to the resources and services on the HPC centers in a secure and straightforward manner. The design and implementation are described, and the computational experience gained while developing the gateway is reported.

Keywords: Grid gateway web service, high performance computing (HPC), science web portal, environmental research

1 INTRODUCTION

Accessing and using High Performance Computer (HPC) centers pose inherent challenges for non HPC specialists such as environmental researchers, modelers, water managers and so on. HPC users typically perform authentication, data transfers, program installation and job management using a terminal user interface and difficult to use commands whose communication is established over secure shell (SSH). To remedy this problem, science web portals [Krishnan et al., 2011; Blodgett et al., 2012; Wang et al., 2013] have been introduced that integrate scientific models, data analysis and tools to visualize results via web browsers. We developed HydroGate to provide functionality to a web portal serving as a gateway for using and accessing HPC resources.

A fundamental goal of grid computing is to provide a software abstraction layer that isolates all of the details through a unified interface to access heterogeneous computer systems, often multiple HPC centers. The

Globus Toolkit has emerged as a de Facto Standard for grid computing [Ferreira et al., 2003] by providing a set of tools for application programming (API) and software development kits (SDKs). However, given the steep learning curve for service commands and program interfaces to use Globus directly [Zhu and Shen, 2009], challenges to integrating Globus with grid platforms and, difficulties in solving the sociological and institutional problems, researchers seek easy-to-use and concise APIs in the form of URLs to expose grid computing capabilities over the web [Wang et al., 2005; Pallickara and Pierce, 2008; Cholia et al., 2010; Erwin and Snelling, 2001].

The contribution of this study is the development of a gateway service providing an interface for client applications to access resources and services on HPC centers in a secure and straightforward manner. We introduce HydroGate, a grid gateway web service, exposing a RESTFul API developed to enable science web portals/applications to transparently access and use state-of-the-art HPC resources. The gateway abstracts away many details and complexities involved in the use of HPC systems without relying on any third-party software installed on HPC centers, but only basic components such as SSH server and HPC job scheduler. The major distinctive feature of HydroGate from other toolkits [Wang et al., 2005; Cholia et al., 2010; Erwin and Snelling, 2001], we adopted zero-installation philosophy that no HydroGate software component needs to be installed particularly on HPC systems to manage HPC jobs. HydroGate provides the following functionality over its RESTful Web API (Table 1):

- **Security** using token-based authentication to the HydroGate service, and then SSH-based authentication to the HPC centers
- **File transfer** back and forth between HPC storage and file server transparent to the service user using secure copy (scp)
- **Submission of jobs** that the user has right to perform to the specified HPC center
- **Monitoring of job status** by means of a URL callback mechanism is carried out by HydroGate to avoid requiring end users to poll job status continuously that notifies service users when the status of job is changed
- **Automatic batch script generation** based on the HPC center preferences and program requirements
- **Discovery** functions to determine the capabilities of HPC centers, HPC programs and program parameters

This paper is organized as follows. Section 2 gives the architecture, software stack and the approach followed. We illustrate the functionality of the service in Section 3. Finally, Section 4 gives concluding remarks.

2 ARCHITECTURE

HydroGate has been developed as part of the CI-WATER project which aims to broaden the application of cyberinfrastructure and HPC techniques into the domain of integrated water resources modeling. Initial implementation of HydroGate is as part of the CI-WATER web portal which is the service user of HydroGate. However we believe that HydroGate can be utilized by not only water resource modeling, but can also serve as general-purpose grid middleware for scientific web portals/applications.

A number of web standards are used in science grid gateways such as Java servlet engine [Pierce et al., 2002], OGSA [Foster et al., 2002], WS-Resource framework [Czajkowski et al., 2004], SOAP [Majithia et al., 2004] and WPS [Baranski, 2008]. While each has its own advantages and disadvantages for grid gateway programmers, we adopt the Representational State Transfer (REST) architectural style that attempts to minimize latency and network communication, while maximizing the independence and scalability of component implementations [Fielding and Taylor, 2002]. RESTful web services also encourage the integration of web service and client applications through standard HTTP methods (GET, POST, PUT, DELETE).

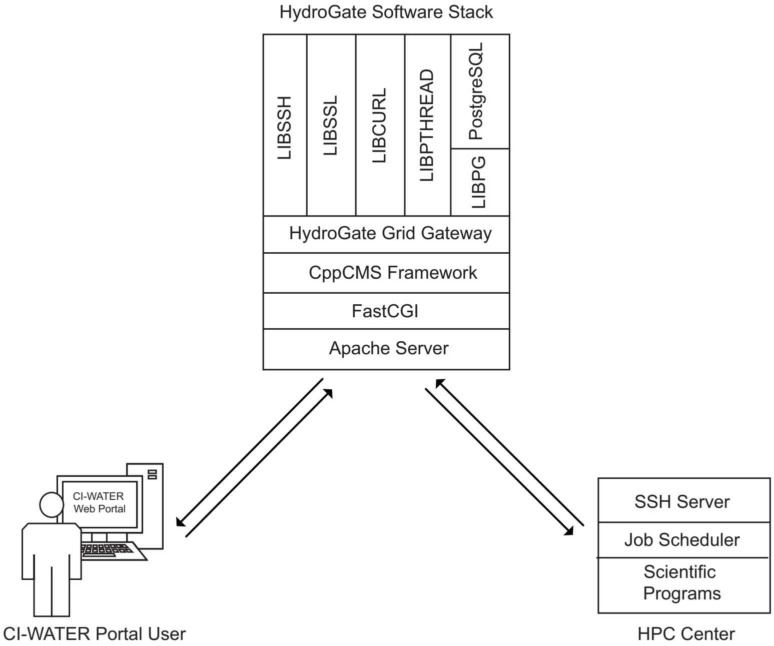


Figure 1. HydroGate and a typical HPC center software stacks

2.1 Software Stack

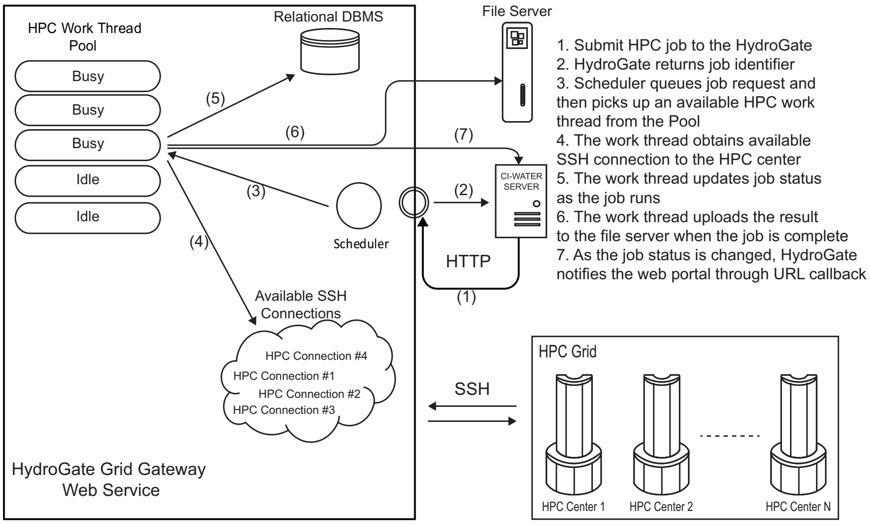
Figure 1 shows the HydroGate software stack. The HydroGate service is hosted on a Linux machine running Apache web server that is passing HTTP commands to CppCMS framework through FastCGI module. The CppCMS framework is a high performance C++ web development framework used to build web services and web pages [Beilis and Tonkikh, 2014]. HydroGate services rely on a PosgreSQL database to keep records of HPC centers, service users, user rights, package information, jobs, programs installed on HPC centers, job batch script templates and so on. The PG library is used to pass queries to the PostgreSQL database. We use the SSH library to be able to login to HPC clusters in a secure fashion, and SSL library to generate a token security key for the service user for authenticating the client application without repeatedly passing username and password information.

The current version of HydroGate is able to utilize multiple HPC clusters that only contain SSH server and PBS job scheduler. As far as most HPC centers is concerned, SSH server and a job scheduler are ubiquitous that do not have Globus or similar grid toolkit installed. As a key design philosophy, HydroGate manages HPC jobs without relying on any HydroGate software installed on HPC clusters.

2.2 SSH Connection Pool and HPC Work Thread Pool Mechanism

HydroGate is a multi-threaded web service maintaining a thread pool containing a number of work thread items that are managed by the scheduler component. The goal of this thread pool is to minimize the turnaround time. As each web service function is invoked, the scheduler passes the request to the work item. If the number of allowed work thread items is exceeded, the request is queued to the FIFO queue to be assigned later when an available work item exists. The work item thread is used for file transfers, job status check, job submission etc. which are considered expensive operations.

A *package* is a compressed zip file that contains input files required for the HPC job. Packages are created by the gateway client, then packages are transferred to a HPC center. HydroGate creates folders with unique names using globally unique identifier (GUID) generator for each package, and then creates sub folders for each HPC job on a HPC center. This allows execution of HPC jobs that might have different input parameters using same input files (package) without duplicating the package. Once the package transfer is completed, the science web portal can submit multiple HPC jobs bound to the package. Thus, the package can be



1. Submit HPC job to the HydroGate
2. HydroGate returns job identifier
3. Scheduler queues job request and then picks up an available HPC work thread from the Pool
4. The work thread obtains available SSH connection to the HPC center
5. The work thread updates job status as the job runs
6. The work thread uploads the result to the file server when the job is complete
7. As the job status is changed, HydroGate notifies the web portal through URL callback

Figure 2. HPC Job Submission Process

used by multiple jobs without the need of retransferring the same package containing the same input files, but possibly with different job parameters. Input and output files in zip format are stored in a file server that provides a shared disk storage for HydroGate and the service user. Use of a dedicated file server provides centralized administration tasks including backup tasks and security updates.

The process of HPC job submission is depicted in Figure 2. Each work item updates job status on the database as the HPC job progress. As the service user invokes the service function to check the job status, HydroGate returns the job status by retrieving it from the database. For convenience and effectiveness, we also adopted a URL callback mechanism that notifies the service user as the HPC job status is changed. In the URL callback mechanism, job/package state associated with identifier number is passed in *query string* format in the part of a uniform resource locator (URL). The service user is expected to update the corresponding job/package record on its database once the state information is received through the URL callback mechanism.

A performance consideration is creating SSH connections to be used by the work thread items. We maintain a SSH connection pool for each HPC center (or account in use) to mitigate this. When an HPC connection is requested from the work thread pool by specifying the HPC identifier, the SSH connection pool checks and then returns if the available connection exists. Otherwise, the SSH connection pool constructs the connection and adds it to the pool.

2.3 HPC Authentication

We have considered two ways for the gateway to authenticate science client applications; through an individual account that every science web portal user must have, or a general account. Since HydroGate modifies the structure of the home directory by uploading package files, creating job folders and installing scientific applications locally, the implications of files being changed in individual user accounts or a general account need to be evaluated. Considerations are:

1. In the case of an individual account, possible changes to files being used by HydroGate by the user working separately from HydroGate.
2. The burden associated with establishing an HPC user account.
3. Some portal users may be reluctant to share their HPC account with the science web portal.
4. Maintaining separate SSH connections for each user in the connection pool may be less efficient.

Most of these points suggest that from a users perspective a common account accessible only to the HydroGate service is preferable. Currently, HydroGate supports this global account approach, but we do have plans to implement support for individual user accounts for use on HPC systems where they are mandated.

3 Illustration of Functionality

TauDEM [Tarboton et al., 2009] is software for hydrologic analysis of digital elevation models (DEMs) that uses MPI to speed up the computations in a parallel computing environment. We have installed TauDEM on the University of Wyoming supercomputer, Mount Moran, that is being used by the CI-WATER project. HydroGate is a database-driven gateway. We created HPC program records containing related data attributes, such as records that represent program type, program authors, execution rights for the gateway users, expected program parameters, install paths and other required records on the HydroGate database.

The UEBGrid snowmelt model [Sen Gupta and Tarboton, 2013] is software for the evaluation of snowmelt over a watershed grid delineated using TauDEM. It supports the simulation of snowmelt surface water inputs used to address water availability in snowmelt driven environments. UEBGrid evaluates in parallel the snowmelt at each grid cell over a watershed. We installed UEBGrid on Mount Moran for use with HydroGate.

Services	GET	POST	PUT	DELETE
Authentication	Get remaining expiration time of the token	Request token	-	-
Applications	Get information of the application (e.g., authors, version, and parameters)	Return names of the programs installed on the HPC center	-	-
Jobs	Get job status	Submit job	-	Delete job
Packages	Get package status	Upload package	-	Delete package
HPC	Return names of all HPC centers that the user has right to access	-	-	-

Table 1. HydroGate REST API functions

Table 2. Parameters of the $submit\_job$ method to invoke pitremove and ueb, respectively, on Mount Moran HPC through HTTP POST method

Name	Value (pitremove)	Value (ueb)
token:	eO5AsKZ4HeawtOzudVTreEmb4...	taETnV-kalTP4sS1-Gzs3ekw...
packageid:	174	213
jobdefinition:	{ ”program”: ”pitremove”, ”hpc”: ”mountmoran”, ”nodes”: 4, ”ppn”: 2, ”walltime”: ”00:00:50”, ”outputlist” : [”fel*.tif”] }	{ ”program”: ”ueb”, ”hpc”: ”mountmoran”, ”nodes”: 32, ”ppn”: 4, ”walltime”: ”00:10:00”, ”parameters”: {”control”: ”control.dat”}, ”outputlist” : [”SWE.nc”, ”ZTest0_0.txt”] }

The list of HydroGate API functions is given in Table 1. Table 2 shows the HTTP POST parameters passed to the service to call $submit\_job$ method that is resulted in the execution of the TauDEM pitremove function and UEB model on Mount Moran HPC center. HydroGate uses a $token$ parameter to authenticate the service user. Each job is associated with the input package that needs to be transferred before the $submit\_job$ is invoked. Each package is identified with the unique number, $packageid$ that is specified in the parameter

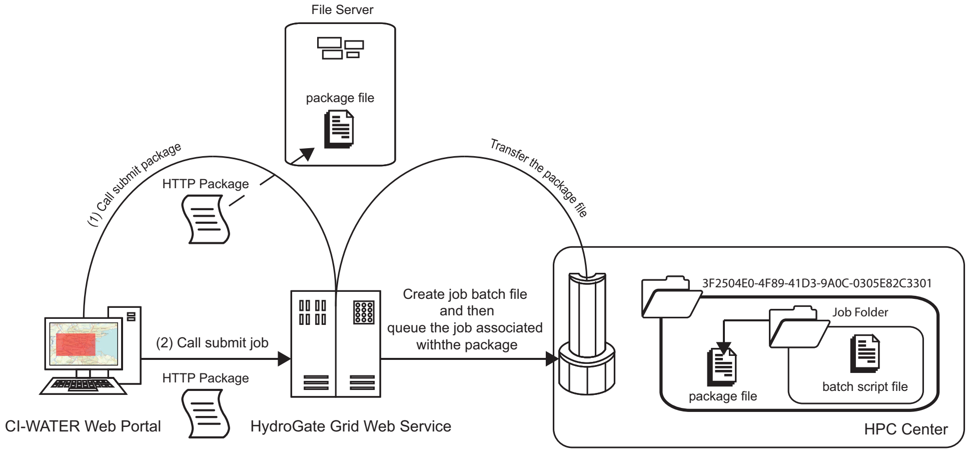

File Server
package file
(1) Call submit package
HTTP Package
Transfer the package file
3F2504E0-4F89-41D3-9A0C-0305E82C3301
Job Folder
Create job batch file and then queue the job associated withthe package
(2) Call submit job
HTTP Package
package file
batch script file
CI-WATER Web Portal
HydroGate Grid Web Service
HPC Center

Figure 3. HPC Job Illustration with package transfer phase, following the job submission phase

list. Each job execution is defined in the $jobdefinition$ parameter encoded in JSON format. This JSON data defines the program name to execute, HPC name, parameters, names of the output files to be transferred back to the file server and other optional parameters, for example, the parameters $nodes$ (number of nodes) and ppn (processor per node) used with MPI programs. The service creates an uniquely-named folder for each package. Once the service user calls $submit\_job$, the service executes its functions inside the package folder. This process is illustrated in Figure 3.

4 Conclusion

In this paper, we describe the HydroGate RESTful gateway web service that we have developed to enhance access to and use of HPC centers in an easy and secure way. The gateway abstracts away many details and complexities involved in the use of HPC systems including authentication, authorization, data and job management, transferring the data back and forth as well as creation, monitoring and scheduling of the jobs without installing any third-party/Hydrogate server software on the HPC systems. The gateway takes care of the managing of HPC job requests in a way that is transparent to the user and client application.

Acknowledgment

Funding for this work, by the U.S. National Science Foundation under grant EPS 1135482 is greatly appreciated. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Baranski, B. (2008). Grid computing enabled web processing service. *Proceedings of the 6th Geographic Information Days, IfGI prints*, 32:243–256.

Beilis, A. and Tonkikh, M. (2014). Cppcms high performance c++ web framework. `http://cppcms.com/wikipp/en/page/main`. Accessed: 2014-03-02.

Blodgett, D., Booth, N., Kunicki, T., Walker, J., and Lucido, J. (2012). Description of the u.s. geological survey geo data portal data integration framework. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(6):1687–1691.

Cholia, S., Skinner, D., and Boverhof, J. (2010). Newt: A restful service for building high performance computing web applications. In *Gateway Computing Environments Workshop (GCE), 2010*, pages 1–11.

Czajkowski, K., Ferguson, D., Foster, I., Frey, J., Graham, S., Maguire, T., Snelling, D., and Tuecke, S. (2004). From open grid services infrastructure to ws-resource framework: Refactoring & evolution.

Erwin, D. W. and Snelling, D. F. (2001). Unicore: A grid computing environment. In *Euro-Par 2001 Parallel Processing*, pages 825–834. Springer.

Ferreira, L., Berstis, V., Armstrong, J., Kendzierski, M., Neukoetter, A., Takagi, M., Bing-Wo, R., Amir, A., Murakawa, R., Hernandez, O., et al. (2003). *Introduction to grid computing with globus*. IBM Corporation, International Technical Support Organization.

Fielding, R. T. and Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Trans. Internet Technol.*, 2(2):115–150.

Foster, I., Kesselman, C., Nick, J., and Tuecke, S. (2002). Grid services for distributed system integration. *Computer*, 35(6):37–46.

Krishnan, S., Crosby, C. J., Nandigam, V., Phan, M., Cowart, C., Baru, C. K., and Arrowsmith, J. R. (2011). Opentopography: a services oriented architecture for community access to lidar topography. In Liao, L., editor, *COM.Geo*, ACM International Conference Proceeding Series, page 7. ACM.

Majithia, S., Shields, M., Taylor, I., and Wang, I. (2004). Triana: a graphical web service composition and execution toolkit. In *Web Services, 2004. Proceedings. IEEE International Conference on*, pages 514–521.

Pallickara, S. and Pierce, M. (2008). Swarm: Scheduling large-scale jobs over the loosely-coupled hpc clusters. In *eScience, 2008. eScience '08. IEEE Fourth International Conference on*, pages 285–292.

Pierce, M. E., Youn, C., and Fox, G. C. (2002). The gateway computational web portal. *Concurrency and Computation: Practice and Experience*, 14(13-15):1411–1426.

Sen Gupta, A. and Tarboton, D. G. (2013). Using the utah energy balance snow melt model to quantify snow and glacier melt in the himalayan region. *81st Annual Meeting Western Snow Conference. Adaptive Water Management in a Changing Climate*.

Tarboton, D., Schreuders, K., Watson, D., and Baker, M. (2009). Generalized terrain-based flow analysis of digital elevation models. In *Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Cairns, Australia*, pages 2000–2006.

Wang, S., Anselin, L., Bhaduri, B., Crosby, C., Goodchild, M. F., Liu, Y., and Nyerges, T. L. (2013). Cybergis software: a synthetic review and integration roadmap. *International Journal of Geographical Information Science*, 27(11):2122–2145.

Wang, S., Armstrong, M., Ni, J., and Liu, Y. (2005). Gisolve: a grid-based problem solving environment for computationally intensive geographic information analysis. In *Challenges of Large Applications in Distributed Environments, 2005. CLADE 2005. Proceedings*, pages 3–12.

Zhu, Y. and Shen, P. (2009). Bring ajax to web application based on grid service. In *Intelligent Human-Machine Systems and Cybernetics, 2009. IHMSC '09. International Conference on*, volume 2, pages 162–165.