Supporting Science Gateways Using Apache Airavata and SciGaP Services

Marlon Pierce
Science Gateways Research Center,
Pervasive Technology Institute,
Indiana University
marpierc@iu.edu

Suresh Marru
Science Gateways Research Center,
Pervasive Technology Institute,
Indiana University
smarru@iu.edu

Eroma Abeysinghe
Science Gateways Research Center,
Pervasive Technology Institute,
Indiana University

Sudhakar Pamidighantam
Science Gateways Research Center,
Pervasive Technology Institute,
Indiana University

Marcus Christie
Science Gateways Research Center,
Pervasive Technology Institute,
Indiana University

Dimuthu Wannipurage
Science Gateways Research Center,
Pervasive Technology Institute,
Indiana University

ABSTRACT

The Science Gateways Platform as a service (SciGaP.org) project provides a rapid development and stable hosting platform for a wide range of science gateways that focus on software as a service. Based on the open source Apache Airavata project, SciGaP services include user management, workflow execution management, computational experiment archiving and access, and sharing services that allow users to share results and other digital artifacts. SciGaP services are multi-tenanted, with clients accessing services through a well-defined, programming language-independent API. SciGaP services can be integrated into web, mobile, and desktop clients. To simplify development for new clients, SciGaP includes the PGA, a generic PHP-based gateway client for SciGaP services that also acts as a reference implementation of the API. Several example gateways using these services are summarized.

CCS CONCEPTS

• **Information systems → Information systems applications**;

KEYWORDS

Science gateways, software as a service, cyberinfrastructure

ACM Reference Format:
Marlon Pierce, Suresh Marru, Eroma Abeysinghe, Sudhakar Pamidighantam, Marcus Christie, and Dimuthu Wannipurage. 2018. Supporting Science Gateways Using Apache Airavata and SciGaP Services. In *PEARC '18: Practice and Experience in Advanced Research Computing, July 22–26, 2018, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3219104.3229240

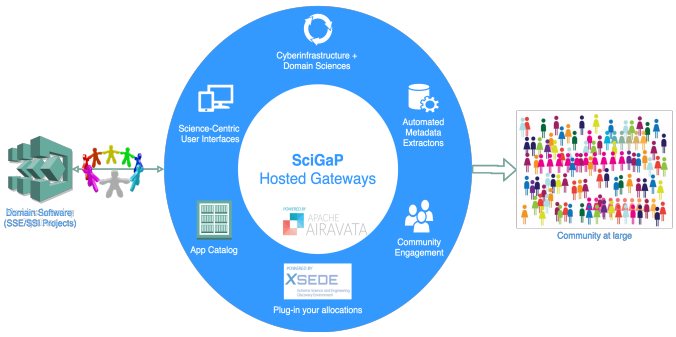

Figure 1: SciGaP services provide gateways that connect software and resource providers with communities.

1 INTRODUCTION

Since the Science Gateways Platform as a service (SciGaP.org) project's start in 2013, it has been used to build and host over thirty science gateways. Hosted gateways include campus gateways that provide cyberinfrastructure for university computing facilities, domain gateways that target a particular field of science, and science gateways that provide "software as a service" for scientific applications, whose developers seek to make their software available to a larger audience without going through the traditional packaging, release, and support cycles for downloadable software. Supporting education and classroom usage as well as research is an important goal for all types of gateways. Figure 1 shows a conceptual representation of SciGaP.

SciGaP services support these gateways through a single, scalable, hosted version of the Apache Airavata software system [4] [8] that supports multiple tenants simultaneously and connects them to multiple backend computing and storage resources. Apache Airavata services include scientific application execution management on HPC and cloud environments, input and output data staging, and provenance tracking for user-created computational experiments. The latter can be searched and shared with colleagues and groups through fine-grained mechanisms [6]. Apache Airavata also exposes a rich set of services for gateway administrators, allowing them to manage metadata about computing resources and scientific applications that power their gateways. All SciGaP services are exposed through a programming language-independent API [7]. Each tenant's users are authenticated through a multi-tenanted identity management system [5] based on Keycloak [1] that can be

PEARC '18, July 22–26, 2018, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-6446-1/18/07.
https://doi.org/10.1145/3219104.3229240

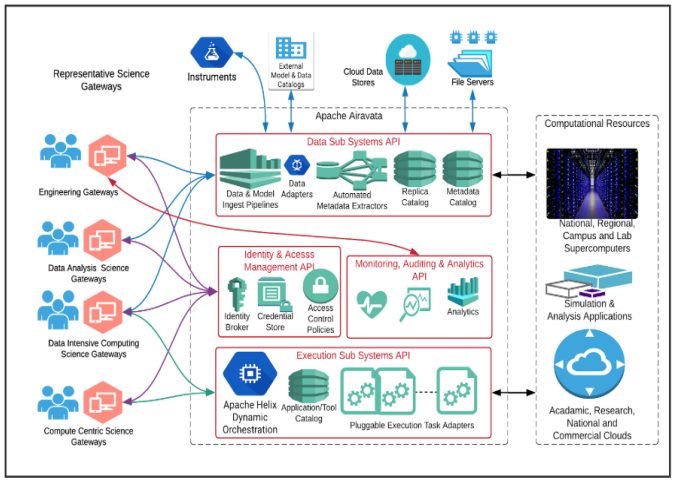

Figure 2: Based on Apache Airavata, SciGaP services provide multi-tenanted hosting and integration with diverse computing and storage resources.

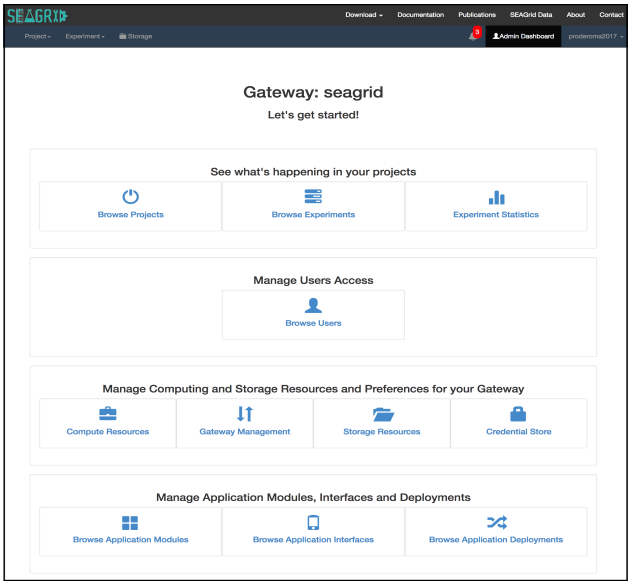

Figure 3: All SciGaP tenant gateways get a PGA gateway that can be used to administer gateway software, resources, and users. The PGA can also serve as a basic gateway for users.

integrated with a wide range of Web and desktop clients. Figure 2 depicts the SciGaP ecosystem.

To quickly onboard new gateways so that they can make use of these services, we provide a hosted, PHP-based reference implementation gateway for the API, which we call the PGA. Developers who need additional functionality can modify the PGA, or they can develop completely new interfaces using the API. Current clients use all of these approaches. We provide examples of SciGaP hosted gateways covering many different use cases, scientific applications, and computing resources in Section 4.

2 SCIGAP SERVICES AND FEATURES

SciGaP services, including Apache Airavata, supporting services such as RabbitMQ and Zookeeper, and PGA-based tenants are hosted on Indiana University's Intelligent Infrastructure hosting environment. SciGaP hosts, manages, and provides storage for its gateway tenants.

All gateway clients get an instance of the PGA, which they can use to configure and manage their gateway tenant. Configuration includes making backend computing and storage resources available through the gateway to the gateway users, and providing prescriptive metadata about software applications that the gateway will provide. These are used to generate form-based user interfaces and to create queue submission scripts that are submitted to remote resources.

SciGaP manages communications with remote resources using a secure set of public-private keys [2]. Gateways may allow users to submit jobs to community accounts (such as used by XSEDE) or to individual accounts. Gateway tenants are also configured to use authentication mechanisms chosen by the client; these are usually campus-based authentication mechanisms accessed through CILogon.

3 REQUESTING A GATEWAY TENANT

Requests for a new gateway tenant are made through the SciGaP.org website. Clients can requests basic support, which includes a PGA-based tenant (Figure 3). The SciGaP team works with these clients to provide basic styling of the gateway, configuration with the desired authentication system, configuration with the client's desired computing and storage resources (typically XSEDE or campus clusters), and configuration of desired software. SciGaP provides up to 1 month of this detailed technical support for gateways to get up and running, followed by optional bi-weekly support calls.

Gateways requiring more extensive consultations can request Science Gateway Community Institute support [3] or XSEDE Extended Collaborative Support Services [9] if using XSEDE. Examples include custom Web interfaces and desktop user interface integration, support for novel resources, and novel integration with data services. XSEDE services require an approved allocation, and SGCI services depend upon available consultants.

Gateway administrators are provided with the Admin Dashboard to support their users, from activating their accounts to troubleshooting issues with executions in HPCs (Figure 4).

4 EXAMPLE SCIGAP GATEWAYS

SciGaP supports client gateways in a number of different fields. We highlight the following recent collaborations.

dREG, developed in collaboration with Charles Danko's laboratory at Cornell University, provides a software-as-a-service gateway that efficiently delivers the developers' bioinformatics application on XSEDE resources. dREG uses a customized version of the PGA hosted on Jetstream that integrated additional visualization interfaces. dREG integration with XSEDE was supported by XSEDE's Extended Collaborative Support Services.

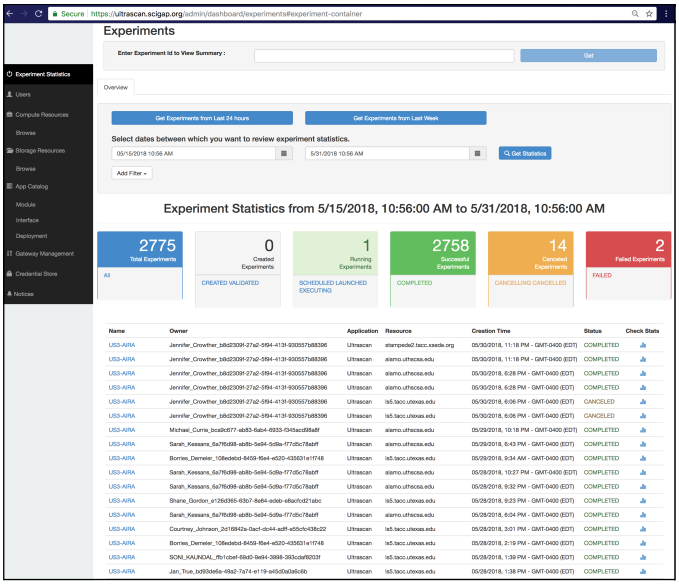

Figure 4: The PGA's administrator dashboard gives a quick look at experiment execution statuses, enabling administrators to debug problems.

The University of South Dakota Campus Gateway gives its users simplified access to a range of popular scientific applications installed on campus clusters at the University of South Dakota. Users authenticate with campus identities and can access resources either through personal or group accounts. This gateway was supported by the Extended Developer Support service from the Science Gateways Community Institute (SGCI).

SimCCS is a gateway to help carbon capture and sequestration planners and decision makers evaluate different sequestration options associated with specific power generation facilities. SimCCS provides both Web and Desktop clients for users. SimCCS is developed with the Indiana Geological and Water Survey through the Science Gateways Community Institute.

The SimVascular Gateway is a software-as-a-service gateway used to deliver SimVascular's flow solver software for modeling blood flow simulation to classroom users at multiple universities. This gateway is developed in collaboration with the SimVascular.org team (Prof. Alison Marsden, Principal Investigator). The SimVascular Gateway uses XSEDE's Comet and is supported by XSEDE ECSS.

The Southern Illinois University Campus Gateway provides a gateway that focuses on providing MaSuRCA and other scientific software to university users of SIU's BigDog cluster. This collaboration used basic support.

The Computational Systems Biology Group at Louisiana State University uses SciGaP services to provide a gateway to several different in-house developed systems biology tools that run on LSU clusters. This gateway is supported by SGCI developer services.

5 CONCLUSIONS AND FUTURE WORK

SciGaP services have been used to run a wide range of gateway tenants, ranging from individual faculty members who wish to deliver their scientific software to a broader audience of users in a scalable way that keeps user support efforts manageable, to computing center resource providers who want to make their systems more accessible and simpler to use.

Clients to SciGaP services may use the PGA, lightly branded but otherwise "as is" and hosted by the SciGaP team, for their gateway, they may make significant modifications to the PGA code, or they may use a different client entirely. SciGaP's API and security model enables these different modes of usage; it is also possible to integrate existing gateways with SciGaP services.

Likewise, SciGaP services can interact with numerous backend resources, queuing systems, and storage, including XSEDE and campus resources. To date, SciGaP has integrated over thirty backend systems into its central platform at the request of client gateways.

We are working actively now to replace the PGA with a more flexible default client developed using the Python-based Django framework. The new portal framework through Django's Wagtail Content Management System integration allow look and feel customization. The Django framework will also simplify how clients can develop richer and science-customized user interfaces. Another future work is to comprehensively integrate with the Airavata sharing service [6] enabling fine grain sharing of experiments, data, applications and compute resources. These are being driven by gateways used in education such as SimVascular. Apache Airavata services themselves will continue to evolve as well, providing better file and data management services, support for supplemental services for gateways to create and manage internal allocations, and a next-generation task execution framework based on the Apache Helix system that will provide more flexible submission mechanisms.

ACKNOWLEDGMENTS

Development of the SciGaP platform and Apache Airavata middleware is supported by NSF Award 1339774.

REFERENCES

[1] Marcus A Christie, Anuj Bhandar, Supun Nakandala, Suresh Marru, Eroma Abeysinghe, Sudhakar Pamidighantam, and Marlon E Pierce. 2017. Using Keycloak for Gateway Authentication and Authorization. (2017).

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

[3] Katherine A Lawrence, Michael Zentner, Nancy Wilkins-Diehr, Julie A Wernert, Marlon Pierce, Suresh Marru, and Scott Michael. 2015. Science gateways today and tomorrow: positive perspectives of nearly 5000 members of the research community. *Concurrency and Computation: Practice and Experience* 27, 16 (2015), 4252–4268.

[4] Suresh Marru, Lahiru Gunathilake, Chathura Herath, Patanachai Tangchaisin, Marlon Pierce, Chris Mattmann, Raminder Singh, Thilina Gunarathne, Eran Chinthaka, Ross Gardler, et al. 2011. Apache airavata: a framework for distributed applications and computational workflows. In *Proceedings of the 2011 ACM workshop on Gateway computing environments*. ACM, 21–28.

[5] Supun Nakandala, Hasini Gunasinghe, Suresh Marru, and Marlon Pierce. 2016. Apache Airavata security manager: Authentication and authorization implementations for a multi-tenant escience framework. In *e-Science (e-Science), 2016 IEEE 12th International Conference on*. IEEE, 287–292.

[6] Supun Nakandala, Suresh Marru, Marlon Piece, Sudhakar Pamidighantam, Kenneth Yoshimoto, Terri Schwartz, Subhashini Sivagnanam, Amit Majumdar, and

Mark A Miller. 2017. Apache Airavata Sharing Service: A Tool for Enabling User Collaboration in Science Gateways. In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*. ACM, 20.

[7] Marlon Pierce, Suresh Marru, Borries Demeler, Raminderjeet Singh, and Gary Gorbet. 2014. The apache airavata application programming interface: overview and evaluation with the UltraScan science gateway. In *Gateway Computing Environments Workshop (GCE), 2014 9th*. IEEE, 25–29.

[8] Marlon E Pierce, Suresh Marru, Lahiru Gunathilake, Don Kushan Wijeratne, Raminder Singh, Chathuri Wimalasena, Shameera Ratnayaka, and Sudhakar Pamidighantam. 2015. Apache Airavata: design and directions of a science gateway framework. *Concurrency and Computation: Practice and Experience* 27, 16 (2015), 4282–4291.

[9] Nancy Wilkins-Diehr, Sergiu Sanielevici, Jay Alameda, John Cazes, Lonnie Crosby, Marlon Pierce, and Ralph Roskies. 2015. An Overview of the XSEDE Extended Collaborative Support Program. In *International Conference on Supercomputing*. Springer, 3–13.