International Journal of u- and e- Service, Science and Technology
Vol.8, No. 10 (2015), pp.377-384
http://dx.doi.org/10.14257/ijunesst.2015.8.10.36

IceBreaker: An Efficient Simulation Management Framework for Web-based Supercomputing Services

Jung-Lok Yu[1], Hee-Jung Byun[2] and Du-Seok Jin[§3]

[1]*Korea Institute of Science and Technology Information (KISTI), Deajeon, South Korea*
[2]*Department of Information & Communications, Suwon University, South Korea*
[3] *Korea Institute of Science and Technology Information (KISTI), Deajeon, South Korea*
Email : [1]junglok.yu@kisti.re.kr, [2]heejungbyun@suwon.ac.kr, [3]dsjin@kisti.re.kr

Abstract

With the recent advent of microprocessors and networks, supercomputing system has been increasingly adopted as a general-purpose simulation environment for a broad spectrum of applications. In particular, the simulation environment has allowed theoretical and/or experimental research in various fields of computational science and to be free of time and spatial limits, and maximized the use of cyber-infrastructure for next generation problem solving. However, traditional supercomputing services have only been handled through a simple form of command-line based console, which leads to the critical decrement of accessibility and usability of heterogeneous computing assets. To address this problem, this paper describes the design and implementation of heterogeneous computing resources/jobs management framework, named IceBreaker, to enable any web-based supercomputing services. The proposed framework has essential functions including user authentication, data management, physical/virtual computing resource management, job management, etc. to provide efficient supercomputing service environments. The framework follows highly extensible modula design principle to manage different kinds of simulations efficiently and allow heterogeneous computing resources, for example, computing clusters, clouds, grids, to be easily integrated with our framework. We also present the detailed specification of standard web-based RESTful endpoints for 3rd-party developers to configure and build a variety of supercomputing service gateways. We hope the attempt presented in this paper will encourage more researchers to join us in designing highly efficient and cost-effective supercomputing service platform.

Keywords: *Supercomputing, Web, Resource management, Job management, Implementation*

1. Introduction

With the advent of microprocessors and networks, recently, supercomputing system has been increasingly adopted as a general-purpose simulation environment for a broad spectrum of applications such as internet services as well as large-scale computational science/engineering applications [1, 2]. In particular, the simulation environment has allowed theoretical and/or experimental research in various fields of computational science (*e.g.*, physics, chemistry, fluids, structural dynamics, healthcare, *etc.*,) to be free of time and spatial limits, and tomaximize the use of cyber-infrastructure for next generation problem solving. However, traditional supercomputing services have only

§ Du-Seok Jin (dsjin@kisti.re.kr). Corresponding author

ISSN: 2005-4246 IJUNESST

been handled through a simple form of command-line based console (*e.g.*, telnet or ssh), which leads to the critical limit of accessibility and usability of computing resources. This is mainly due to the absence of formal/open application programming interfaces that can be used to access resources and execute different kinds of applications.

This paper describes the design and implementation of heterogeneous computing resources/jobs management framework, named IceBreaker, to enable any web-based supercomputing services. The proposed framework has essential functions including user authentication, data management, physical/virtual computing resource management, job management, *etc.*, to provide efficient supercomputing service environments. The framework follows extensive module-based design principles so as to manage different kinds of simulations and allow heterogeneous computing resources to be easily integrated with our framework. We also present the detailed specification of standard web-based RESTful endpoints [3] for 3rd-party developers to configure and build a variety of supercomputing service gateways.

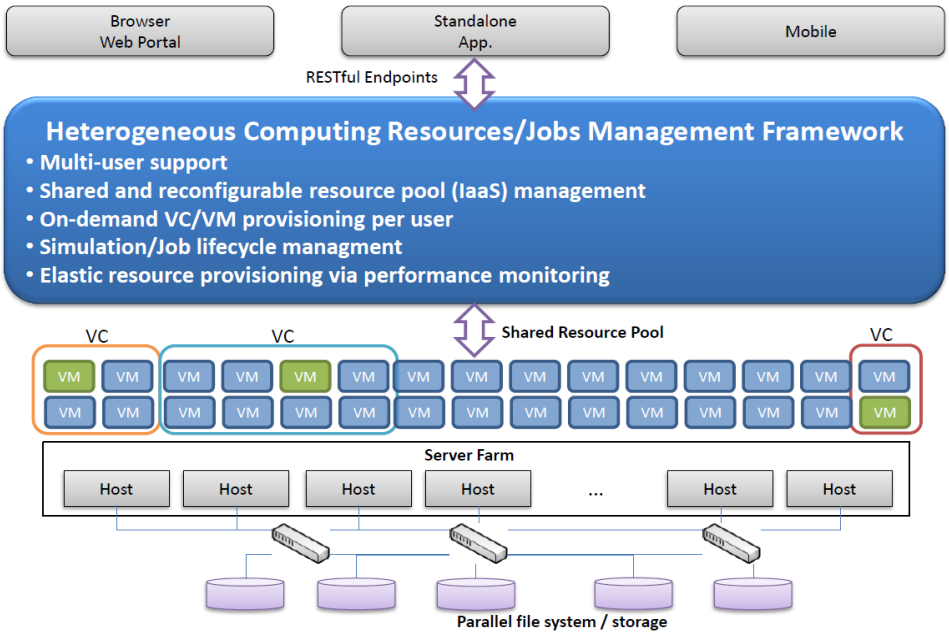

Figure. 1. Heterogeneous Computing Resources/Jobs Management Framework (IceBreaker)

2. Related Works

Recently, simulation environments involving cyber infrastructure are being utilized in various fields for the purpose of education/research. Some representative examples are nanoHUB [4], ICLCS [5], ICEAGE [6], and e-AIRS [7].

Developed in 1995 under the Network for Computational Nanotechnology (NCN) project, nanoHUB offers various simulation software and content for nanotechnology researchers. It is used by more than 200,000 researchers from 172 countries.

nanoHUB has carried out approximately 350,000 simulations as of May 2011. It has been cited 575 times in research papers, indicating its increasing popularity. Their ranges of services include Rappture, a tool for utilization of simulation software based on the HUBZero platform [8], simulation sharing, and content conversion. The open-source HUBZero platform is being used in more than 30 application fields such as manufacturing, disaster, and healthcare.

ICEAGE, a multi-national education-oriented project led by the European Union (EU), provides large-scale and multi-purpose cyber infrastructure based on Enabling Grids for

E-science (EGEE). Educational programs and materials are developed for researchers and educators to teach in a grid environment.

e-AIRS is a system for virtual wind tunnel experiments via the web, allows numerical analysis through the utilization of heterogeneous computing resources, and facilitates comparative analysis of such results. Since 2008, the e-AIRS portal has been available online, enabling computational fluid simulations even without professional knowledge.

3. IceBreaker: A Heterogeneous Resources and Jobs Management Framework

To provide large-scale users with an efficient simulation environment, essential services include user authentication, simulation-related data input/output, efficient physical/virtual computing resource management, and job management. This section explains a heterogeneous computing resources/job management framework, which offers functions such as user authentication, file input/output management, physical/virtual resource management, and job management services.

3.1. Design Principles

The purpose of proposed heterogeneous computing resources/jobs management framework (IceBreaker), as shown in Figure 1 is to manage different kinds of computing resources and jobs, and to establish the foundation APIs for various supercomputing services. Its design principles are as follows:

• Efficiency

The framework must efficiently and fully utilize available resources to execute different characteristics of workloads.

• Expandability

The framework must maintain neutrality to computing resources on which applications are running. It also must be compatible with various virtualized environments (Xen, KVM, *etc.*,) [9] and support various job managers (OpenPBS, LoadLeveler, Sun Grid Engine, *etc.*,) [10]. This means to provide an abstract layer that is interoperable with them.

• Openness

The framework must provide an open, web-standard interface to accommodate various clients such as web portals, stand-alone applications, and mobile environments.

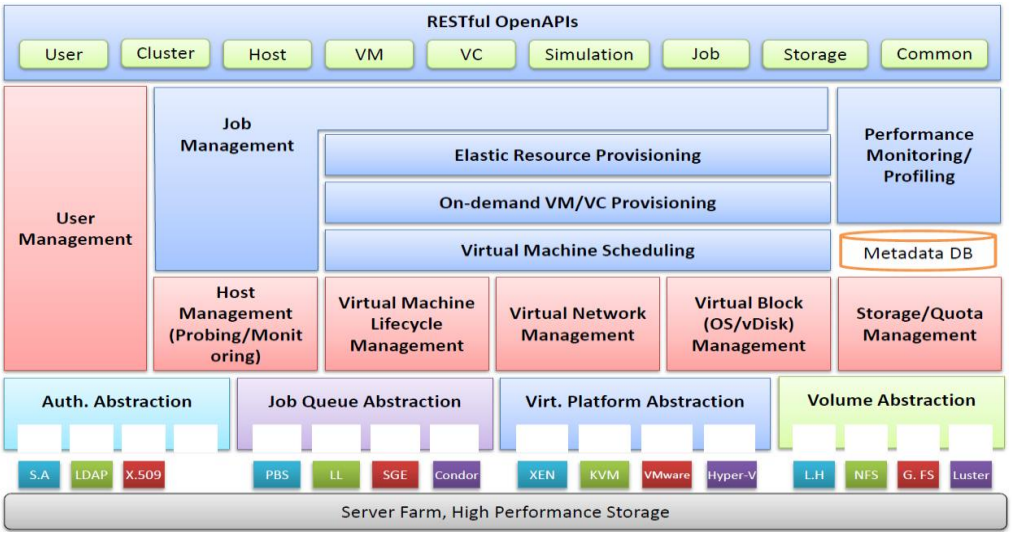

Figure 2. Layered Architecture of IceBreaker

3.2. Architectural Overview of IceBreaker

Figure 2 shows the hierarchy of IceBreaker framework. From bottom to top, the framework consists of the abstraction layer, service core, and web service layer. The abstraction layer supports different types of existing environments (authentication, resources, job, storage subsystems, *etc.*,), while the service core allows user management, provisioning of physical/virtual servers, and job management. The web service layer provides a web-standard HTTP(S) REST interface.

The abstraction interface is designed for user management, virtualized platform management, job management and storage management. It supports basic functionalities with the pre-established infrastructure. Currently, the user management and authentication layer provides the local database and Light-weight Directory Access Protocol (LDAP) plugins. For inter-operation with the virtualized platform, the Xen and KVM plugins are available on a trial basis. Supported plugins for job management and storage management are OpenPBS and NFS, respectively.

The service core layer stores/manages objects including user, cluster, host, virtual machine (VM), virtual cluster (VC), virtual network, virtual image, storage, and simulation job in the form of Plain Old Java Objects (POJOs) using Hibernate/HSQL. It carries out functions such as actual user registration/deletion/authentication, physical server addition/deletion, on-demand virtual machine/cluster provisioning, and job controls. We have adopted multi-threaded workers model in our service core implementation. Please refer to Figure 3 for more details.

The uppermost web service layer provides RESTful endpoints (will be described in next section) based on the Create-Retrieval-Update-Delete (CRUD) model for user, physical server, virtual machine, virtual cluster, simulation/job, and storage. It has been developed using the Spring Web MVC framework (see Figure 3).

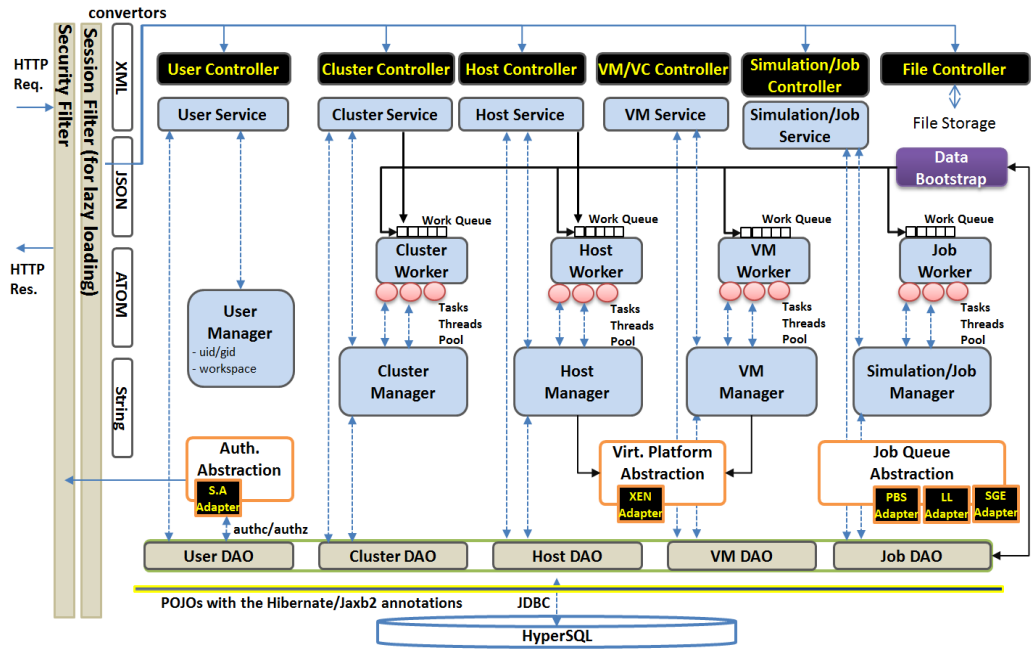

Figure 3. IceBreaker Internals

4. IceBreaker RESTful Endpoints

The REST interface provided by the heterogeneous computing resources/job management framework is as follows.

4.1. User Authentiction and Management

The administrator designated by the system is able to register/delete users. General users can gain system authentication through the login/logout interface, and access other service endpoints using the authentication token provided when successfully authenticated. Within the framework, a HTTP(S) BASIC authentication mechanism is used for user authentication and authority delegation. Different levels of access are granted to administrators and general users (see Table 1).

4.2. Heterogeneous Resource Management

To perform simulations according to the user requests, heterogeneous computing resources must be registered in the shared pool. For this purpose, the physical server driving the virtual machine/cluster is equipped with the add/delete/list function. The endpoints for resource management have been designed such that access is granted to administrators only (see Table 1).

Table 1. IceBreaker Endpoints for A User and Resource Management

Category	HTTP Method	Endpoints	Meaning	Access Control	
				admin	user
User Mgmt.	POST	-/api/user/login	Login	O	O
	GET	-/api/user/logout	Logout	O	O
	GET	-/api/user/list	Show all the registered users	O	X
	GET	-/api/user/{UserId}/info	Show the specified user	O	O
	POST	-/api/user/register	Add a new user	O	X
	PUT	-/api/user/{UserId}	Update the user info.	O	O
	DELETE	-/api/user/{UserId}	Unregister the specified user	O	X
	POST	-/api/user/deactivate	Deactivate user(s) and backup all the users data	O	X
Resource Mgmt.	POST	-/api/host/register	Register a new host to the system	O	X
	GET	-/api/host/list	Show all the hosts	O	X
	GET	-/api/host/{HostId}/info	Show the specified host	O	X
	DELETE	-/api/host/{HostId}	Delete the specified host	O	X

4.3. Virtual Machine (VM) and Virtual Cluster (VC) Provisioning

These endpoints are for virtual machine/cluster provisioning on physical servers registered by administrators. As shown in Table 2, it is accessible by both administrators and general users. Virtual machine provisioning is requested with specifications for number of processors and memory size, and detailed information of provisioned virtual machines is available for owners. The owner of a virtual machine may submit requests to suspend/resume the machine.

4.4. File Management

In general, simulation jobs require multiple data inputs and produce multiple results upon completion. The endpoints for file input/output per user are as shown in Table 2.

Table 2. IceBreaker Endpoints for VM/VC and File Management

Category	HTTP Method	Endpoints	Meaning	Access Control	
				admin	user
VM/VC Mgmt.	POST	-/api/vm/provision	Provision a new VM	O	O
	PUT	-/api/vm/{VmId}/suspend	Suspend the specified VM	O	O
	PUT	-/api/vm/{VmId}/resume	Resume the specified VM	O	O
	PUT	-/api/vm/{VmId}/shutdown	Shutdown the specified user	O	O
	GET	-/api/vm/list	Show all the owned VMs	O	O
	GET	-/api/vm/{VmId}/info	Show the specified VM's	O	O

			Information		
File Mgmt.	POST	-/api/file/upload?cluster={clusterName}	Upload a file to the user's repository on the clusterName storage	O	O
	POST	-/api/file/write?name={fileName}&cluster={clusterName}	Write the body content to a file on the clusterName storage	O	O
	GET	-/api/file/download?id={fileId}	Download the file	O	O
	GET	-/api/file/read?id={fileId}	Read the file	O	O

4.5. Simulation and Job Management

Simulation, which serves as a virtual parent, is an object that encompasses job sets. The simulation managements' endpoint facilitates management of parameter-sweep job sets and provides various functions such as simulation generation, information viewing, deletion, and modification. After creating a simulation object, users can engage in job submission and control. The job title, job type (sequential or parallel), the application to be executed, and variables are specified using XML (or JSON). The submit endpoint is then called. The status of a submitted job can be monitored or cancelled through the status/cancel API.

Table 3. IceBreaker Endpoints for Simulation and Job Management

Category	HTTP Method	Endpoints	Meaning	Access Control	
				admin	user
Simulation/ Job Mgmt.	POST	-/api/simulation/create	Create a simulation	O	O
	DELETE	-/api/simulation/{SimID}	Delete the simulation	O	O
	GET	-/api/simulation/list	Show all the owned simulations	O	O
	POST	-/api/simulation/{SimID}/job/submit	Submit a job	O	O
	GET	-/api/simulation/{SimID}/job/{JobID}/status	Show the job's status	O	O
	GET	-/api/simulation/{SimID}/job/list	Show all the jobs in the simulation	O	O
	PUT	-/api/simulation/{SimID}/job/{JobID}/cancel	Cancel the job	O	O
	GET	-/api/job/{JobID}/output?dir={dirName}	Get the metadata of output files	O	O
	GET	-/api/job/{JobID}/download/zip	Download the zipped output file	O	O

5. Conclusion

This paper described the design and implementation details of a novel heterogeneous computing resources/jobs management framework, IceBreaker, for web-based supercomputing services. Along with the architectural overview of IceBreaker, we provided an in-depth explanation of its RESTful endpoints. Deploying proposed framework, we are currently doing an on-going project of e-Science services [11] for various application areas from computational fluid dynamics, nano-physics, and chemistry to structural dynamics and computational design [12]. We expect the framework to be gradually expanded into other application fields. Finally, we hope the attempt presented in this paper will encourage more researchers to join us in designing highly efficient and cost-effective supercomputing service platform.

Acknowledgement

This research was supported by the Korea Astronomy and Space Science Institute under the R&D program supervised by the Ministry of Science, ICT and Future Planning.

References

[1] B. Hindman, A. Konwinski, M. Zaharia and I. Stoica, "A common substrate for cluster computing" in Workshop on Hot Topics in Cloud Computing (HotCloud), **(2009)**.
[2] R. Seungwoo, K. Jik-Soo, K. Sangwan, K. Seoyoung and H. Soonwook, "Effective Distributed Supercomputing Resource Management for Large Scale Scientific Applications", Journal of KIISE, vol. 42, issue 5, **(2015)**.
[3] "REST (Representational State Transfer)ful Interfaces", Available: http://en.wikipedia.org/wiki/Represen tational_state_transfer.
[4] "nanoHub", http://nanohub.org.
[5] "ICLCS (Institute for Chemistry Literary and Computational Science)", http://iclcs.illinois.edu.
[6] "ICEAGE (International Collaboration to Extend and Advance Grid Education)", http://www.iceage-eu.org/.
[7] "eAIRS", http://eairs.kisti.re.kr.
[8] "HubZero Platform", http://hubzero.org/.
[9] A. Chierici and R. Veraldi, "A quantitative comparison between xen and kvm," in Proceeding of the 17th International Conference on Computing in High Energy and Nuclear Physics(CHEP), **(2009)**.
[10] J. Cacheiro, "Analysis of Batch Systems," Technical Report CESGA-2014-002, **(2014)**.
[11] L. Kotthoff, T. Kelsey and M. McCaffery, "Web-scale distributed eScience AI search across disconnected and heterogeneous infrastructures", in Proceeding of the The 10th IEEE International Conference on e-Science, **(2014)**.
[12] K. Hyung-Il, K. Saji, L. Hakjin, R. Minseok, K. Taehee and C. Seongim, "Development of an Engineering Education Framework for Aerodynamic Shape Optimization", International Journal of Aeronautical and Space Sciences, vol. 14, issue 4, **(2013)**.

Authors

Jung-Lok Yu, received his PhD from KAIST in Daejeon, Korea in 2007. He was a senior engineer in Samsung Electronics from 2007 to 2010. He is currently a senior researcher with the Supercomputing Center at the Korea Institute of Science and Technology Information. His research interests include parallel processing, cluster computing, and cloud computing.

Hee-Jung Byun, received her PhD from Korea Advanced Institute of Science and Technology **(**KAIST), Daejeon, Korea, in 2005. She was a senior engineer in Samsung Electronics, Ltd. from 2007 to 2010. She is currently an assistant professor with the Department of Information & Communications, Suwon University, Kyunggi-do, Korea. Her research interests include network modeling and controller design.

Du-Seok Jin, is a senior researcher at the National Institute of Supercomputing and Networking at the Korea Institute of Science and Technology Information. He received his M.S. degree in Computer Science from Chonbuk National Univ., Republic of Korea in 2001 and his PhD degree in Computer Science from Paichai Univ., Republic of Korea in 2011. His research interests include information retrieval systems, cloud storages, and distributed file systems.