Information Engineering Express
International Institute of Applied Informatics
2019, Vol. 5, No. 2, 57 – 72

Automating Account Management on an Academic HPC System with Authentication Federation

Junya Nakamura [*], Masatoshi Tsuchiya [*]

Abstract

There is increasing demand for high-performance computing (HPC) cluster systems because more and more computational resources are necessary to carry out large-scale simulations and machine learning, which are a crucial part of research and business today. There are two choices for HPC cluster systems: academic HPC cluster systems and public cloud services. Academic HPC cluster systems can be used with reasonable usage fees, although they take a long time until the cluster system is ready to use, because of its user registration procedure. On the other hand, public cloud services such as Amazon Web Services can be used immediately, but the usage fees are expensive compared to academic systems. To fill the gaps between them, we propose an academic HPC cluster system with automated user registration that allows on-demand HPC cluster use.

Keywords: authentication federation, high-performance computing, user management, web service

1 Introduction

Large-scale computations, such as computer simulation and machine learning, have become increasingly important and are an essential part of product manufacturing and service development today. These computations require a vast amount of computational resources. There are three ways to conduct such computations in academic research.

The first way is to acquire a large research grant and build your own high-performance computing (HPC) cluster. This way is ideal because researchers can utilize an HPC cluster that is optimized for their research. However, it takes two or more years from proposing a research theme and receiving approval for the theme to get the cluster. Moreover, the scale of the cluster strongly depends on the amount of the grant.

The second way is to use existing supercomputers, such as one installed at the researcher's organization or at a joint usage/research center, e.g., JHPCN[1]. This way is costeffective because the usage fees for such supercomputers are typically low. However, it also takes three or more months before researchers can use the supercomputers, because the application period to use the supercomputers is limited, and it takes some time until the application is approved.

[*] Toyohashi University of Technology, Aichi, Japan

[1] `https://jhpcn-kyoten.itc.u-tokyo.ac.jp/en/`

The third way is to borrow computing resources from public cloud services, such as Amazon Web Services, Microsoft Azure, and Google Computing Platform. It takes only a few minutes to get a user account for these services, so a researcher can run large-scale computations almost immediately. However, the servers and networks of cloud services are not optimized for HPC, especially as the interconnections between the servers are narrow. Additionally, because public cloud services charge a storage fees based on the amount of data, they are not well suited for research themes that require a large data set.

To summarize the discussion above, although three types of HPC systems exist, at present, no system can be used on demand with reasonable fees. In this paper, we propose an HPC cluster system that addresses these shortcomings: The proposed system allows users to register automatically without any human interaction on the HPC service side. To achieve this goal, we must solve two problems: user authentication and account management.

User authentication: For user registration, an HPC cluster system operator must con-firm that the account information of the HPC cluster system is really delivered to the user of the account. To ensure confirmation, the user registration procedure for most existing HPC cluster systems is conducted manually. In contrast, in the proposed HPC system, we utilize an academic authentication federation called GakuNin [1] to identify who registers an account in the system. GakuNin is a Shibboleth-based single sign-on system and allows the user of an organization to be authenticated with the organization's account. Because the proposed system directly authenticates users with the GakuNin federation, we avoid the user authentication problem.

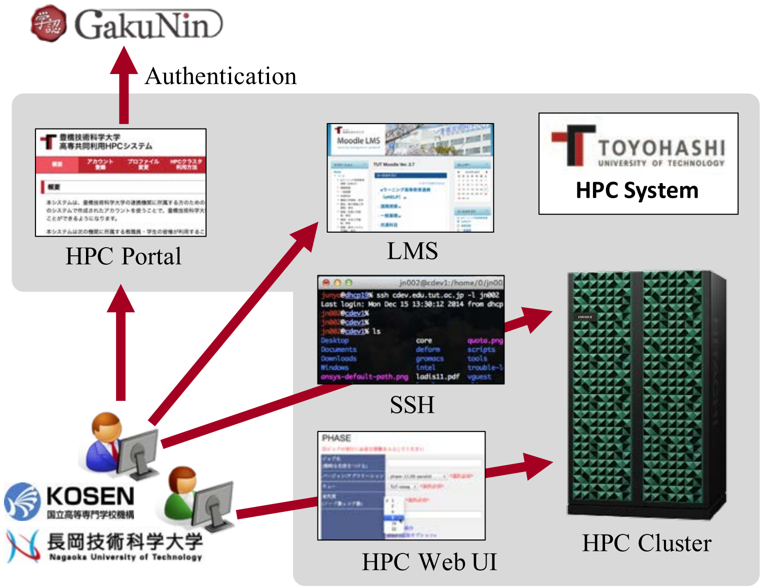
Account Management: An HPC operator must manage user accounts periodically, i.e., creating and disabling accounts. Although this can be done manually if there are only a few accounts, it is unrealistic if there are thousands of users. Therefore, we develop a web-based account management system for the proposed HPC cluster system in which the existing user management system interacts automatically and allows users to manage their accounts themselves.

The remainder of this paper is organized as follows. In Section 2, we briefly describe the design of the proposed system, and in the following section (Section 3), we provide details about the implementation of the proposed HPC cluster system. In Section 4, we explain the proposed HPC cluster system's utilization status and inquiries from users we encountered during the operation of the proposed HPC system. In Section 5, we discuss the difference between the proposed system and typical academic HPC systems and describe why our system can automate the user registration process. In Section 6, we conclude the paper.

2 System Design

In this section, first, we give readers a brief explanation of the proposed HPC system, and then we describe the design.

Figure 1 illustrates an overview of the proposed HPC cluster system. The HPC cluster system was funded byacollaboration project involving three organizations (Nagaoka University of Technology, Toyohashi University of Technology, and National Institute of Technology (KOSEN), which has 51 national technical colleges). One of the purposes of the project is training students who have advanced innovative skills, and the mission of the HPC cluster system is providing faculty staff and students of the



```
jn002@cdev1:/home/0/jn00
junya@dhcp19% ssh cdev.edu.tut.ac.jp -l jn002
Last login: Mon Dec 15 13:30:12 2014 from dhcp
jn002@cdev1%
jn002@cdev1%
jn002@cdev1% ls
Desktop                 core          quota.png
Documents               deform        scripts
Downloads               gromacs       tools
Windows                 intel         trouble-l
ansys-default-path.png  lodis11.pdf   vguest
```

Figure 1: Overview of the proposed HPC system

organizations a useful HPC environment that can be easily accessed throughout Japan to support the purpose. The cluster is installed on and operated by Toyohashi University of Technology.

The fundamental ideas of the proposed HPC system to accomplish the mission are as follows: (1) use the academic authentication federation called GakuNin [1] to authenticate users and (2) build a registration system that allows users to self-register and self-manage their user accounts. Below, we discuss both ideas and explain the services provided by the proposed HPC system.

2.1 User Authentication

The proposed HPC system potentially provides an HPC service to many users and many organizations, i.e., about 7,000 faculty staff and about 53,000 students from two universities and 51 national technical colleges (KOSEN). Of course, we could register and manage the users manually, but this is unrealistic because it requires too many human resources from our IT division. For that reason, we decided to use GakuNin Federation [1] as the user authentication foundation. This decision enables us to avoid managing users' authentication information (username and password).

GakuNin is a Japanese academic authentication federation operated by the National Institute of Informatics (NII). The federation specifies the application standard of the fed-eration and provides a single sign-on (SSO) service among identity providers and service providers that rely on the standard. A user whose organization joins the federation can uti-lize many services with his or her account issued by the organization and without having to manage many service-specific user accounts. Technically, the federation is built on Shibbo-leth [2], which is a Security Assertion Markup Language (SAML)-based SSO middleware developed by Shibboleth Consortium. In early 2018, 209 higher education organizations (identity providers) and 88 service providers joined the federation. Most importantly, the 51 national technical colleges also joined the GakuNin federation in late 2013.

We also provide another registration method that does not require the GakuNin federation for some organizations that do not intend to join GakuNin federation. The other method uses OpenIdP [3], which is an identity provider operated by the GakuNin Federation Office to help users who belong to organizations that have not yet joined GakuNin federation.

2.2 Account Management

Although we are not required to manage users' authentication information thanks to the GakuNin authentication federation, we still have to spend time managing user accounts for the HPC system. Specifically, the account management task is composed of the following subtasks: (1) creating a Linux account for the HPC system, (2) disabling an account when a user has lost the right to use the service (e.g., leaving his or her organization), and (3) registering a public key for Secure SHell (SSH), which is used when connecting to the HPC cluster.

To process the management tasks automatically, we built a web-based account regis-tration system called *the HPC portal system*. The system provides three functions: (1) account registration, (2) profile maintenance, and (3) extension of the account expiration date. Hereafter, we describe these functions.

When a user wants to use the HPC system, he or she visits the HPC portal system and authenticates himself or herself by the identity provider of the user's organization through the GakuNin federation. At this time, the HPC portal system can recognize that the user belongs to the organization without requiring any additional human interaction. Then, the user inputs his or her information, such as name, email address, and student ID. The HPC portal system creates an account with the information in the existing user management system of Toyohashi University of Technology through the external application programmable interface (API) of the user management system. In the account creation process, the system also creates the user's UNIX account and the account's home directory. When the process finishes, the user is ready to use the HPC system.

The HPC portal system also allows a user to change his or her account information. This function is used mostly to register a user's SSH public key to the user management system. Users can also change their login shell and email address with the function.

An account issued by the HPC portal system is valid for at most 12 months. Every May, the HPC portal system asks a user via email whether the user wants to continue to use his or her account or not. A user who wants to continue to use his or her account visits the HPC portal system and authenticates himself or herself with GakuNin authentication. If the authentication is confirmed, the HPC portal system confirms that the user still belongs to the organization and modifies the expiration date of the account through the external API of the user management system. If the user takes no action until the end of May or if the user cannot authenticate because he or she has left the organization, the user management system disables the account automatically.

2.3 Provided Services

We provide four HPC-related services on the HPC system: (1) HPC cluster, (2) HPC web UI, (3) HPC cluster user manual, and (4) learning management system. The HPC cluster is the main component of the system and provides an HPC software development environment, a CUI-based job management interface, and HPC applications via SSH. The HPC web UI provides another job management interface that suits users who are

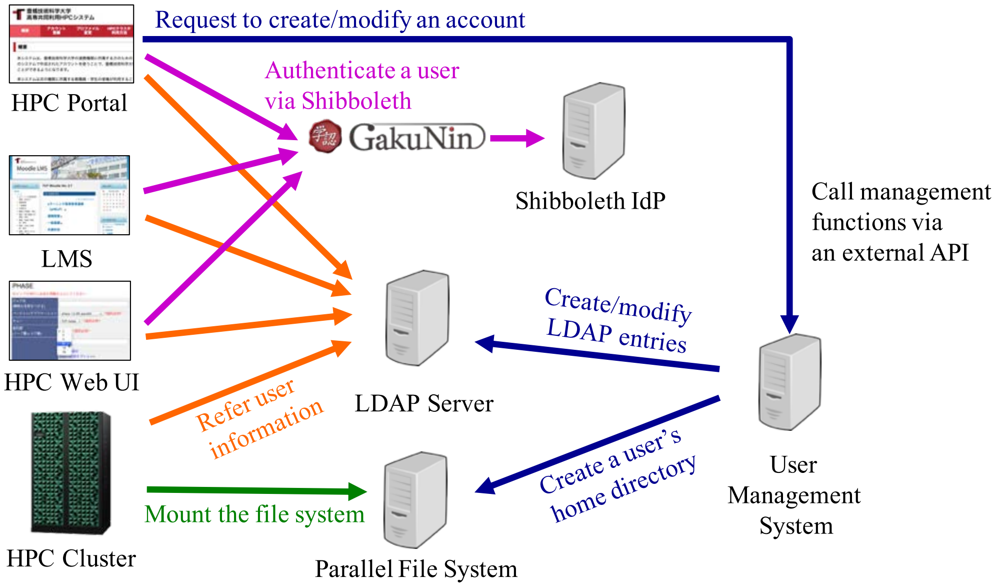

Request to create/modify an account
Authenticate a user via Shibboleth
GakuNin
HPC Portal
LMS
HPC Web UI
HPC Cluster
Shibboleth IdP
Call management functions via an external API
Create/modify LDAP entries
LDAP Server
Refer user information
User Management System
Create a user's home directory
Mount the file system
Parallel File System

Figure 2: Relations between the Subsystems

not familiar with CUI. A user can submit and monitor his or her jobs with the service via a web browser. The HPC cluster user manual is hosted by MoinMoin Wiki[2], running on the same server as the HPC portal system. The learning management system provides HPC-related e-learning courses to users, such as HPC programming, lecture movies about HPC software, and state-of-the-art research presentations. We use Moodle [4] as the learning management system platform.

As discussed in Section 2.1, the three services except the HPC cluster authenticate users with GakuNin authentication [1]. The HPC cluster authenticates a user with an SSH public key, which can be registered by the user through the HPC portal system, instead of GakuNin authentication. We use an SSH public key because Shibboleth is a web-based authentication mechanism, and it is hard to use Shibboleth with SSH client software.

3 HPC Cluster System Implementation

In Section 2, we discussed the design of the proposed HPC system. In this section, we explain how we developed the system based on the design.

3.1 System Organization

The system consists of the seven subsystems illustrated in Figure 2. The HPC portal system plays an important role in the system and is a web-based account management system that provides basic account management functions for users. The HPC portal system creates and modifies a user's account through the external application programmable interface provided by the user management system. In Section 3.2, we discuss in detail how the HPC portal system cooperates with the user management system.

The HPC portal system, learning management system, and HPC web UI run Shibboleth SP services on their systems and join the GakuNin federation as service providers. Thus,

[2] https://moinmo.in/

Table 1: Specifications of the HPC Cluster (1st Generation)

Type	#units	CPU	Memory	Accelerator	OS
Login node	2	Xeon E2680 × 2	64 GB	Tesla K20X	RHEL6.2
Comp. node A	2	Xeon E2680 × 2	64 GB	Tesla K20X	RHEL6.2
Comp. node B	26	Xeon E2680 × 2	64 GB	N/A	RHEL6.2

Table 2: Specifications of the HPC Cluster (2nd Generation)

Type	#units	CPU	Memory	Accelerator	OS
Login node	2	Xeon E2680v2 × 2	128 GB	Xeon Phi 5110P × 2	RHEL6.4
Comp. node A	16	Xeon E2680v2 × 2	128 GB	Xeon Phi 5110P × 2	RHEL6.4
Comp. node B	16	Xeon E2680v2 × 2	128 GB	N/A	RHEL6.4

these systems can cooperate with each organization's identity provider (IdP). All of the subsystems retrieve user account information from the LDAP server. Because the LDAP server stores the information as a subclass of the posixAccount[3], Linux-based computation nodes of the HPC cluster can use the user information with the Linux standard manner, i.e., Pluggable Authentication Module (PAM) and Name Service Switch (NSS). The computa-tion nodes mount a Lustre [5] or GPFS [6] volume on the parallel file system with the native driver of the file system other than NFS.

Tables 1 and 2 show the specification of the HPC clusters[4]. Table 3 shows the specification of the other subsystems. We use existing systems with small configuration changes as the subsystems and develop from scratch only the HPC portal system for the system.

Table 4 shows the software available on the HPC cluster. Applications that provide a graphical user interface, e.g., MATLAB or ANSYS, can be used by connecting the login nodes via X11 forwarding. Some of these applications may require additional preconditions for use, based on their end-user license agreements.

3.2 Details of User Registration and Account Creation

As explained in Section 2, a user who wants to use this system must register his or her information in the user management system and create an account via the HPC portal system, before using each service. In this section, we describe the details of the user registration and account creation process.

The HPC portal system is a web-based system we built from scratch with PHP, Smarty, and MySQL. The user registration and account creation process with the HPC portal system is illustrated in Figure 3 and as follows:

1. A user visits the HPC portal system via a web browser. If this is the user's first visit, the HPC portal system redirects the user to his or her organization's authentication system (Shibboleth IdP) to prove that he or she belongs to the organization.

2. The user enters his or her username and password into the authentication system. After the authentication, the identity provider redirects the user back to the HPC portal system.

[3] `https://tools.ietf.org/html/rfc2307.html`

[4] As the HPC cluster was replaced during the operation of the system, we show the specifications for both HPC clu sters.

Table 3: Specifications of the other subsystems except the HPC cluster

Subsystem	Using Software
Parallel file system	Lustre (1st-generation cluster)
	GPFS (2nd-generation cluster)
HPC portal system	PHP 5.4
	MySQL 5.5
	LDAP Manager ELM Interface for Java
	Shibboleth SP 2.4
Learning management system	Moodle 2.7
	Shibboleth SP 2.4
HPC web UI	ShareTask 5.0
	Shibboleth SP 2.4
User management system	LDAP Manager 6.0
LDAP server	OpenLDAP 2.4
RADIUS server	FreeRADIUS 2.1

Table 4: Software provided on the HPC cluster. Note that the software versions here are the ones installed when the system began. After the system was released, we repeatedly updated the software on demand from users.

Category	Software	Version
Compiler	Intel Compiler	13.0.1.117
	PGI Compiler	13.1–1
	GNU Compiler	4.4.6
	CUDA Compiler	5.0
MPI Library	Intel MPI	13.0.1.117
	Open MPI	1.6.2
	MPICH2	1.5.rc3
Structural Analysis	ANSYS Multiphysics	14.5
	ANSYS CFX	14.5
	ANSYS LS-DYNA	14.5
	ANSYS HFSS	14.5
	ABAQUS	6.12
	COMSOL	4.4
	Patran	2012.2
	DEFORM-3D	10.2
Computational Chemistry	PHASE	11.00
	UVSOR	3.4.2
	OpenMX	3.6
	Gaussian	0.9 Rev.C.01
	NWChem	6.11
	GAMESS	2012.R2
	MPQC	3.0-alpha
	Amber	12
	CONFLEX	7
	CHEMKIN-PRO	15112
Numerical Computing	MATLAB	R2012a

3. The HPC portal system shows an input form to the user. The user must provide his or her name, email address, ID number of the organization, and role of the organization (student, faculty, staff, and so on) in the form.

4. The HPC portal system sends a message to the email address entered in the previous step to confirm the email address. The user opens the URL link in the message via a web browser.

5. The service's terms of service are displayed to the user. If the user agrees to the terms of service, the user presses the "agree" button. When the button is pressed, the HPC portal system kicks a shell script to create an account on the user management system with the information that the user entered as arguments of the script, and the account creation process begins.

6. First, the script validates the arguments, i.e., the user's information. If a format error is found, the script and the account creation process fail. Otherwise, the script runs a Java program that can communicate with the user management system via a proprietary library for that system. The program calls the account-creating function of the user management system. The function stores the account information into the user management system's internal database, registers the account information into the LDAP server, and creates the user's home directory on the parallel file system. If all of the coordination succeeds, the whole account creation process finishes. The HPC portal system then displays the result to the web browser and sends a notification message to the email address to inform his or her account information.

We can skip step 3 because the information entered at step 3 can be retrieved from the organization's Shibboleth IdP as SAML attributes. GakuNin federation's technical application standard [7] specifies these attributes as exchangeable information. However, this cannot be achieved with the policy of the technical college identity providers, and therefore, we decided to require users to enter their information themselves.

We also store a user's eduPersonPrincipalName[5] attribute (ePPN), whose value is guaranteed to be persistent and unique among all GakuNin IdPs and SPs, in the user management system, other than the entered information. The reason why we do this so is that we provide the proposed HPC system as three GakuNin service providers (HPC portal system, learning management system, and HPC web UI) and must distinguish a user throughout them, not just in one of them. Although a typical GakuNin service provider uses eduPerson-TargetedID[6] (ePTID) to distinguish their users, those values are unique only in the service provider, and the values possibly collide with other services' ePTID. Thus, we could not use this attribute.

3.3 How Each Subsystem Authenticates a User

In this section, we explain how each subsystem authenticates a user and how we considered each subsystem's unique circumstances.

[5] https://www.internet2.edu/media/medialibrary/2013/09/04/internet2-mace-dir-eduperson-201203.html#eduPersonPrincipalName

[6] https://www.internet2.edu/media/medialibrary/2013/09/04/internet2-mace-dir-eduperson-201203.html#eduPersonTargetedID

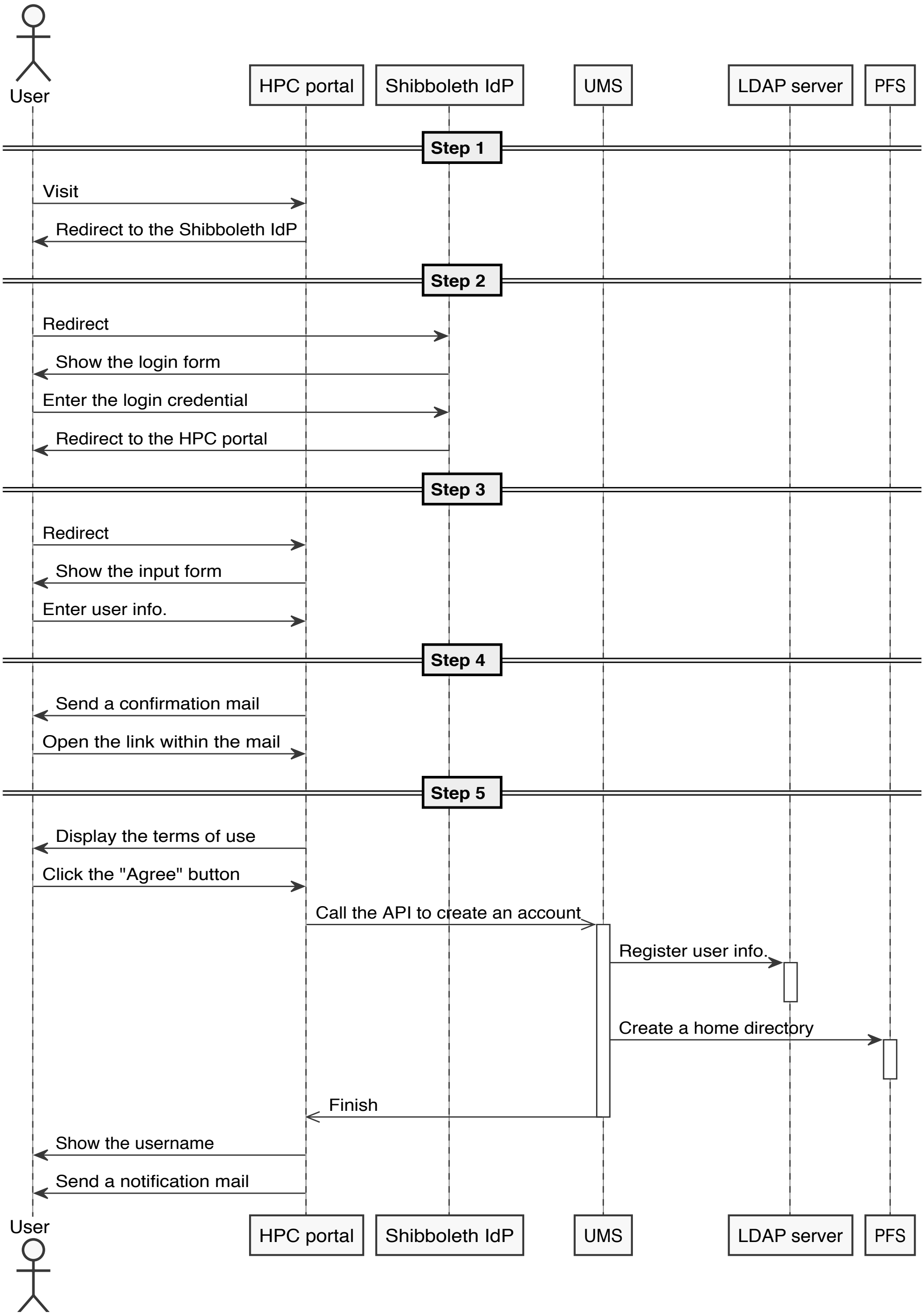


Figure 3: Sequence diagram of the registration flow. Here, we denote by UMS and PFS the user management system and the parallel file system, respectively.

3.3.1 HPC Cluster

When a user wants to use the HPC cluster, he or she runs an SSH client to connect the login node with SSH public key authentication. Although all user information, including SSH public key, is stored in the LDAP server, the OpenSSH service running on the login cannot refer to the SSH public key from the LDAP server when authenticating a user. To solve this problem, we provided two customizations in the OpenSSH service: First, we prepared a wrapper script that retrieves the user's SSH public key from the LDAP server and passes the public key to the OpenSSH service. Second, we customized the OpenSSH service to run the wrapper script when a user attempts to connect.

A user can register his or her SSH public key into the LDAP server as follows:

1. A user visits the HPC portal system and uploads the SSH public key that he or she wants to register from his or her PC.
2. When the upload finishes, the HPC portal system executes the Java program described in Section 3.2 with the public key. The program registers the public key through the external application programmable interface of the user management system.
3. The user management system automatically propagates the registered public key to the LDAP server, to which the login node refers.

3.3.2 Learning Management System and HPC Web UI

It was not hard to cooperate with other organizations' identity providers because both subsystems originally support Shibboleth, which is the standard middleware of GakuNin federation, as their user authentication method. Therefore, we just installed the Shibboleth SP software in the subsystems and configured them.

We also needed to modify both subsystems' Shibboleth authentication modules. The modules were designed to retrieve user information from SAML attributes sent by Shibboleth IdP, but our service cannot do that, as described in Section 3.2. Therefore, we fixed the modules to get the user information from the LDAP server.

3.4 OpenIdP Integration

In Section 2.1, we mentioned that the HPC system allows a user who cannot use GakuNin authentication to register his or her account. Here, we describe how this was achieved.

A fatal problem in cooperating with OpenIdP instead of GakuNin is we cannot verify whether a registering user belongs to an organization. If we authenticate the user by GakuNin, this can be done automatically by the user's identity provider. Therefore, as shown in Figure 4, when the HPC portal system accepts a registration request from an OpenIdP user, the system requires the user to enter information that will be needed to authenticate him or her, such as the student ID and organization, and delegates this verification process to the user's organization. For instance, we asked the department of educational affairs at Nagaoka University of Technology to verify their students as the university did not join in the GakuNin federation at that time. This verification procedure is also conducted when a user wants to extend his or her account expiration date.

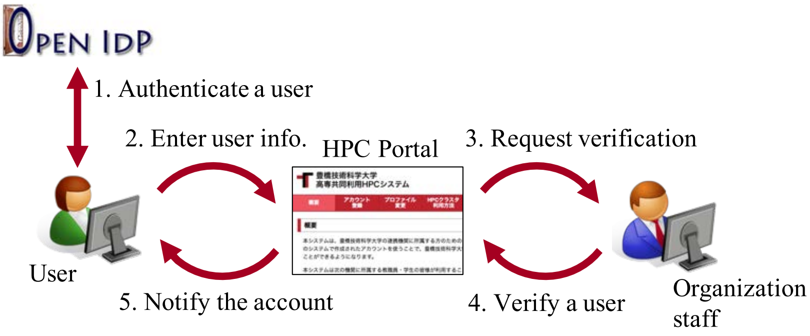


Figure 4: Open IdP Integration

3.5 Remote Access VPN Service and Guest Wi-Fi Service

After we released the HPC service, we added two network services based on user requests: a remote access VPN service and a guest Wi-Fi service. The remote access VPN service was requested by a joint research project that has members of the three organizations because the members The guest Wi-Fi service was maintained to provide Internet access to students and teachers who visit Toyohashi University of Technology (for example, we have a summer internship program for the national technical college students, and they typically stay on campus for a week). A user can easily acquire his or her password for these services from the HPC portal system by logging into the system and clicking the “Issue VPN/Wi-Fi password” button. The issued password is valid for seven days, and the user can get a new one there if the password expires, and he or she wants to extend his or her access. This network account information is stored in a MySQL database in the HPC portal system, and the RADIUS server refers to the database for the guest Wi-Fi and the VPN endpoint.

We decided to build our own password-issuing function on the HPC portal system, although a de facto academic federated network authentication system, eduroam [8], exists. We did not use eduroam for a simple reason: The availability of eduroam was limited in most organizations. When we designed the function, Nagaoka University of Technology and National Institute of Technology had a plan to join the eduroam federation, but when they will is unknown. Therefore, we gave up on using eduroam in the system. [7]

3.6 Development Schedule

We built the proposed HPC system on the schedule shown in Table 5. The development began in September 2014, and it took about six months until the HPC service was released. The development was divided into two parts: system integration and negotiation to related organizations. One person was assigned to each task. After we released the service, we developed the two network services described in Section 3.5. We also switched an authenti-cation method for Nagaoka University of Technology members from OpenIdP to GakuNin, because they joined the GakuNin federation.

To build the HPC system, we had to cooperate with other organizations, e.g., the national technical colleges and the GakuNin federation office. It was important to start

[7] By November 2018, Nagaoka University of Technology and seven national technical colleges had joined the eduroam federation, but this is only 13% of the system targets; thus, the decision seems to have been reasonable.

Table 5: Development schedule. Here, we denote by KOSEN and NUT the information planning department of National Institute of Technology and Nagaoka University of Technology, respectively.

Date	Action
Nov. 2013	Asked KOSEN staff about the possibility of realizing authentication cooperation with GakuNin federation.
Sep. 2014	Designed whole system architecture.
Oct. 2014	Designed HPC portal system.
Nov. 2014	Began to implement HPC portal system.
Nov. 2014	Asked OpenIdP administrators to use their IdP to authenticate NUT members.
Dec. 2014	Fixed Moodle Shibboleth Authentication Plugin.
Dec. 2014	Fixed HPC web UI Authentication Module.
Dec. 2014	Finished implementing HPC portal system.
Dec. 2014	HPC portal system, LMS, and HPC web UI joined GakuNin Test Federation and started testing user authentication.
Jan. 2015	Discussed with KOSEN staff about the configuration change procedure of the national technical colleges' IdPs.
Jan. 2015	HPC portal system, LMS, and HPC web UI joined GakuNin Production Federation.
Feb. 2015	Requested IdP configuration changes from KOSEN for the national technical colleges' IdP administrators.
Mar. 2015	Held an explanatory meeting for KOSEN users.
Mar. 2015	Released the HPC service.
Sep. 2015	Released the remote access VPN service.
Jun. 2017	Released the guest Wi-Fi service on Toyohashi University of Technology
Jan. 2018	Use NUT's IdP to authenticate their members, instead of OpenIdP.

negotiating with them as soon as possible because such cooperation usually takes a longer time than negotiating with another division of our organization. This is what we learned from the development.

4 Utilization Status

In this section, we report the proposed HPC system's utilization status and the difficulties we encountered during the operation of the system.

4.1 Changes in the Number of Users

Figure 5 depicts the changes in the number of registered users of the HPC system, from March 2015 to November 2018. We held an explanatory meeting for faculty staff and students of the national technical colleges in March 2015. After the meeting, 51 users created accounts. The number of users increased by about 80 every year, on average, and in late 2018, there were 388 users from 53 organizations. The recent two-years' growth was mostly brought about by the national technical college students who take an e-learning course provided by the learning management system.

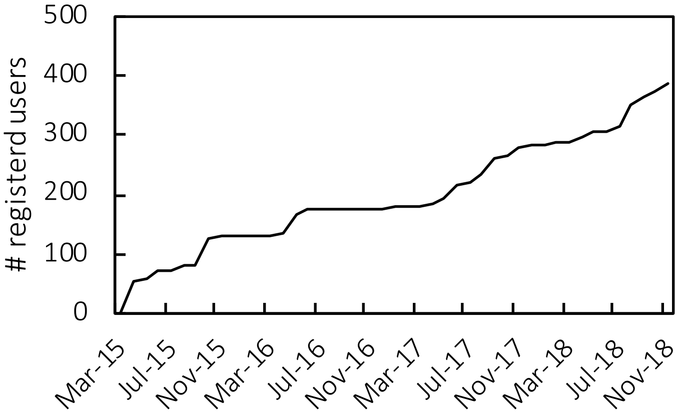


Figure 5: Transitions of the number of registered users

4.2 Typical Inquiries from Users

In this section, we introduce typical inquiries from users during the operation of the HPC system and how we responded.

4.2.1 Using GUI Applications via X11 Forwarding

In the HPC cluster, a user can use a GUI application, e.g., ANSYS or COMSOL, on the login node by using X11 forwarding. However, Windows, which most of the users use, does not support the X11 protocol natively, and it is difficult to configure X server software for Windows. Although there is free X11 server software for Windows, such as Xming [9] and Cygwin X server [10], many users could not configure the software correctly. We received many inquiries about X11 forwarding, such as the following: "I cannot run MATLAB with GUI mode. The application crashes immediately after I run that." "I am running ANSYS with X11 forwarding, but the rendering is very slow."

Using the X11 forwarding function on Linux is more stable than that on Windows. Thus, we prepared a step-by-step guide for installing VirtualBox[8] on a Windows PC and connecting to the login node on a Linux virtual machine running on the VirtualBox.

4.2.2 SSH Public Key Format Differences

There are three major SSH-related software for Windows: Tera Term [11], Putty [12], and WinSCP [13]. Tera Term and Putty are SSH clients, and WinSCP is a file transfer software that supports the Secure Copy Protocol (SCP). Although the software can connect to an SSH server via public key authentication, applicable SSH public key formats differ: Tera Term supports only the OpenSSH format, and the others support the Putty format. These two formats are fundamentally incompatible, although a user can convert from the Putty format to the OpenSSH format with puttygen.exe, with which Putty bundles.

The problem is that a user cannot use the key pair he or she usually uses, because the OpenSSH service running on the login node accepts only OpenSSH format public keys. The problem occurs (1) when a Putty user registers his or her SSH public key in

[8] `https://www.virtualbox.org`

the user management system, or (2) when a Tera Term user wants to copy a remote file to a local PC with WinSCP. We often had inquiries about these problems and suggested the user convert his or her public key with puttygen.exe.

4.2.3 Usage Policy Differences between the National Technical Colleges

As described in Section 2.1, all of the national technical colleges joined the GakuNin federation. However, this does not mean that every user of these technical colleges can use the HPC system. Some colleges allow their students to use the system (i.e., a student knows his or her account and password), but some do not (i.e., a student does not know his or her account; then, the student cannot use our system). In the first year after the service release, such issues often occurred, and this difference made troubleshooting difficult. Fortunately, as time passed, these troubles became rare because using services through GakuNin feder-ation has become popular in the colleges.

4.2.4 Misunderstanding of eduPersonPrincipalName

The two universities and the 51 national technical colleges, which use the HPC system, have a credit transfer system. For this system, our university provides e-learning courses with the management system, which is a subsystem of the HPC system. Although the HPC system is designed for self-registering, accounts for the credit transfer system were registered by our university staff manually based on information submitted by the students' organization to avoid problems.

Because the eduPersonPrincipalName (ePPN) attribute is necessary for our system to identify a user as discussed in Section 3.2, we required the organization's school affairs department to inform these users' ePPN. However, almost all ePPN attributes were wrong. The most typical mistake was entering a student's email address as the attribute because the ePPN format is very similar to that for an email address. Even though we explained that the submitted attribute was wrong and asked the organization staff to provide the correct one, they did not understand what ePPN is.

Although we spent two years explaining what ePPN is to resolve the misunderstanding, finally, we gave up the workflow. We fixed the learning management system to output error logs verbosely when a login error that originates from a wrong ePPN occurred and used the error logs to guess the correct ePPN. We also changed the work-flow to require these students to register their accounts themselves.

4.3 Perspective of Improvements of the System

Here, we summarize our improvements of the system. We designed and implemented the user registration system for users who do not belong to our university. This system automatically creates an account for users by collaborating with their institution's user authentication system through GakuNin. We also modified authentication modules of our learning management system and the HPC Web UI system to allow these users to use the systems. The registration system also can issue their passwords for our campus Wi-Fi and VPN system for their convenience.

Typically, we issue about 75 new accounts every year for users who do not belong to our university but want to use our services. Before using the proposed system, we created these accounts manually, and it took three or more days to complete the creation task because we had to check the adequacy for issuing the accounts for each application.

After the release of the proposed system, we can reduce the creation time drastically thanks to system's automation mechanism.

5 Discussion

Although the automated user registration mechanism, which is achieved by the proposed HPC system, is common among Internet platforms and services, such as Twitter and Amazon, it is standard for academic HPC systems. In this section, we describe why this mechanism is hard and why we can realize it.

First, most academic HPC systems are maintained for research. A potential HPC service user must submit an application to an HPC system's organization, and the application must be approved by the organization before the user can access the HPC system. The application is examined for approval from many aspects, e.g., academic impact and fitness with the installation purpose of the HPC system. This examination requires wide and deep knowledge to judge, and it is hard for computers.

The second reason is usage fees. On an academic HPC system, a user is charged for his or her CPU time and storage usage. Typically, the user cannot pay the usage fee from his or her own money. The fee is a public expense paid by the organization. An accountant of the HPC system has to contact the user's organization to confirm that he or she can actually pay the usage fee, and it is hard to automate this confirmation task.

The third reason is the identity verification method. Most HPC systems verify a user's identity strictly to avoid invalid use of the systems. For example, High Performance Com-puting Infrastructure (HPCI)[9] requires a user visit its nearby office and show the user's certificate of employment and ID card issued by a public agency when the user starts using the HPC cluster[10]. Although the verification procedure could be automated, it cannot be achieved in 2018.

Our HPC system was installed from the beginning to provide the HPC service for three related organizations without any usage fee. Therefore, we did not care about these three problems above. Anyone who belongs to the organizations can use the proposed HPC system on demand.

However, another problem arises when an on-demand HPC system such as this system is available. The problem is preventing the HPC system from executing malicious jobs, such as password cracking or cryptocurrency mining like BitCoin [14]. A typical academic HPC system excludes jobs that deviate from research purposes with strict identity verification and terms of use. However, it is difficult for on-demand HPC systems to expect a user to use HPC services under their terms of use because it is much easier to use HPC services than existing academic systems. Fortunately, this was not a problem for our system. If we have more users in the future, we will need a mechanism that detects and stops such malicious jobs.

6 Conclusion

We developed an academic HPC cluster system that provides an automated user registration function to users from multiple organizations. The automated user

[9] http://www.hpci-office.jp/folders/english

[10] "4.5 Identity Vetting" of HPCI Quick Start Guide; http://www.hpci-office.jp/materials/hpci-startguide_e.pdf?19

registration was achieved by fully utilizing the Japanese academic authentication federation called GakuNin and building an automated account management system. We were able to provide the HPC service to 53 organizations with fewer human resources, and it was not necessary to customize the HPC system for each organization, because all the identity providers relied on the GakuNin standard. In the three years since the service was released, the HPC service has been used by 388 users across 53 organizations.

References

[1] Cyber Science Infrastructure Development Department, National Institute of Informatics, “GakuNin,” https://www.gakunin.jp.

[2] Shibboleth Consortium, “Shibboleth Consortium — Privacy Preserving Identity Management,” https://www.shibboleth.net.

[3] National Institute of Informatics, “OpenIdP,” https://openidp.nii.ac.jp.

[4] The Moodle Project, “Moodle – Open-source learning platform,” https://moodle.org.

[5] OpenSFS and EOFS, “Lustre File System,” http://lustre.org/.

[6] F. Schmuck and R. Haskin, “GPFS: A Shared-Disk File System for Large Computing Clusters,” in *Proceedings of the 1st USENIX Conference on File and Storage Technologies (FAST).* USENIX Association, 2002, pp. 231–244.

[7] Committee for Academic Authentication, GakuNin Federation, “GakuNin Federation Technical Application Standard Version 2.3,” http://id.nii.ac.jp/1149/00000212/, 2017.

[8] Global eduroam Governance Committee, “eduroam – World Wide Education Roaming for Research & Education,” https://www.eduroam.org.

[9] C. Harrison, “Xming X Server for Windows - Official Website,” http://www.straightrunning.com/XmingNotes/.

[10] Cygwin authors, “Cygwin/X,” http://x.cygwin.com.

[11] TeraTerm Project, “Tera Term Open Source Project,” https://ttssh2.osdn.jp.

[12] S. Tatham, O. Dunn, B. Harris, and J. Nevins, “PuTTY: a free SSH and Telnet client,” https://www.chiark.greenend.org.uk/~sgtatham/putty/.

[13] WinSCP.net, “WinSCP :: Official Site :: Free SFTP and FTP client for Windows,” https://winscp.net/.

[14] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” Tech. Rep., 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf