



Development of a web interface for submitting jobs to SLURM

Desarrollo de una interfaz web para el envío de trabajos a SLURM

Fabián León^{1a}, Gilberto Díaz^{1b}

¹Departamento de Ciencias de la Computación, Facultad de Ingenierías Fisicomecánicas, Universidad Industrial de Santander, Colombia.

Email: ^a fabian.leon@correo.uis.edu.co, ^b gilberto.diaz@uis.edu.co

Received: 11 October 2018. Accepted: 24 February 2019. Final version: 9 August 2019.

Abstract

Protocols such Secure Shell have been commonly used by Linux clusters to allow users sending jobs to SLURM. However, it implies the use of a console emulator to establish the remote communication which, in some cases, is not available. Therefore, this paper presents the development of the Web Submit SLURM API which offers a quick and safe web interface for submitting jobs to SLURM, querying the job queue, and creating and uploading batch files.

Keywords: SLURM; Cluster Linux; CGI; C++.

Resumen

Protocolos como Shell seguro han sido utilizados comúnmente por los clusters de Linux para permitir a los usuarios enviar trabajos a SLURM. Sin embargo, implica el uso de un emulador de consola para establecer la comunicación remota que, en algunos casos, no está disponible. Por lo tanto, este documento presenta el desarrollo de la API Web Submit SLURM, que ofrece una interfaz web rápida y segura para enviar trabajos a SLURM, consultar la cola de trabajos, crear y cargar archivos batch.

Palabras clave: SLURM; Cluster Linux; CGI; C++.

1. Introduction

Simple Linux Utility for Resource Management and job scheduling (SLURM) [1] is an open source cluster management and job scheduling system which offers: the allocation of exclusive and/or non-exclusive access to resources to users on the set of allocated nodes; a framework for starting, executing, and monitoring jobs, and the management of a pending job queue to moderate the connection to the available resources. Therefore, SLURM has been used by linux cluster which share their resources with different users to execute their jobs, such as the supercomputer tianhe-2 in National Supercom-

puter Center, Guangzhou, China [2]; the SEQUOIA of the Lawrence Livermore National Laboratory [3], and Guane, the cluster of the Universidad Industrial de Santander in Bucaramanga, Colombia [4].

The secure shell protocol (SSH)[5] is commonly used to access the cluster command line in the management node that executes the slurmctld daemon for secure remote communication. However, it implicates that the user requires a console emulator.

To overcome this limitation, this paper presents the Web Slurm Submit (WebSS) API wich allows to submit and



query the job queue through a simple Web interface. For this, WebSS establishes a communication between any authentication server, in this case, the Lightweight Directory Access Protocol (LDAP)[6], and SLURM to submit jobs. In this manner, the user just requires to login to a personal account, have a directory in the management node, query the job queue, and submit, or create a batch file. Unlike Pyslurm[7] and Heron[8] SLURM supporting APIs implemented in python and java languages, WebSS uses PHP, HTML, and the Common Gateway Interface (CGI) [9], together with the library Cgicc[10] that simplifies the creation of CGI applications in C++ for the World Wide Web.

2. Materials and methods

WebSS was developed following the waterfall model [11] shown in Figure 1 since it provides a rigid, structured, and easy to manage development software process. The user needs and the system characteristics to solve them were evaluated in the initial stage. The most important obtained requirements were: the user will be able to create a batch file using a form; the user will be able to upload a batch file; the user will be able to query the job queue; the interaction of the user with the API must be via Web. The design and implementation stages will be exposed in sub sections 2.1 and 2.2, respectively.

The verification was performed on the local PC described in section 2.2 obtaining positive results free of errors and security failures. Further, WebSS was implemented in the real cluster Guane to guarantee the correct operating when interacting with different users; details will be described in results section. Finally, in the maintenance stage, it was planned to update WebSS when changes are made to new versions of SLURM or Cgicc that may create incompatibilities or errors in the normal operation of WebSS.

2.1. Flow of Information

Figure 2 shows the WebSS basic structure and the flow of information indicated by the arrows that connect the serves, users, and systems, named actors, that participate in the WebSS performance. Note that, each actor can receive or respond to a message or request from another author, indicated by going in and out arrows, respectively. In (1) the flow initiates when the user communicates with WebSS and fills the login form; in (2) the form is sent to WebSS such that in (3) the information is validated by the LDAP server; in (4) LDAP responds to WebSS with the authentication response.

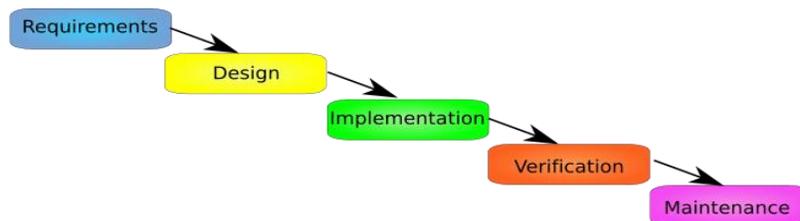


Figure 1. waterfall model used in WebSS

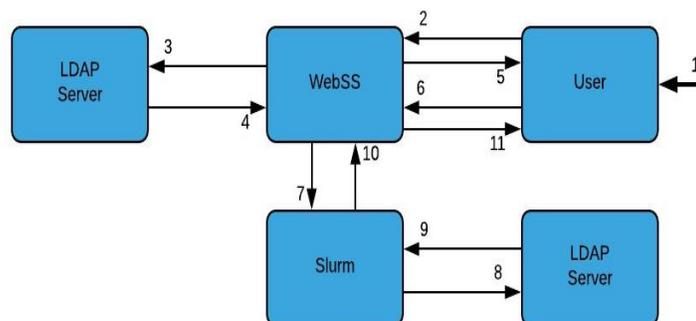


Figure 2. Basic structure, actors and flow of information in WebSS

If authentication fails, the process ends, but if authentication success, in (5) WebSS shows to the user three sections: query the job queue, upload, or create a batch file filling a form; in (6) the user returns the file/form to WebSS, or the process ends if the user just query the job queue; in (7) WebSS processes the file/form and sends the resource request to SLURM; in(8) SLURM queries if the job can be executed or put on hold; in (9) the server communicates to SLURM the result; in (10) SLURM notifies to WebSS, and the process ends in (11) when WebSS notifies the job status.

2.2. WebSS System Implementation

WebSS implementation and operation was carried out on a local PC with Linux operating system Mint 18.3 Sylvia, SLURM 15.08.7, OpenSSL 1.0.2g, and OpenLDAP 2.4.42. It was based on four main files. The first is an HTML login form who asks for an username and a password. The second is a PHP file which validates the user information in the authentication server and presents the options to query the queue job, upload, or create a batch file using an HTML form. Third and fourth files use CGI C ++ via the Cgicc library to process the sent information by the second file and differs depending on the user selection. If the user selects uploading a batch file, WebSS creates a batch file in the user directory; if the user selects filling the form, WebSS creates a new batch file to fill it with the requested resources. These last two files are also responsible of sending the jobs to SLURM, showing the results to the user, and ending the process.

2.3. Security

The Secure Sockets Layer (SSL) and the Transport Layer Security protocol (TLS) were considered to maintain the integrity of the data and the code safety during the communication with WebSS. These protocols encrypt the data to guarantee a secure communication between the user and the servers [12]. Further security measures include a random token in PHP generated at each session starting and validated in all PHP files with WebSS interacts. Also, the inclusion of code lines in the WebSS implementation is not allowed since the interaction forms are assured to avoid sending or including malicious code.

3. Results

WebSS was implemented in Guane which is the cluster of the Universidad Industrial de Santander, located in the Parque Tecnológico de Guatiguara, Santander, Colombia and is composed of 16 nodes ProLiant SL390s G7, Ubuntu 14.10 operating system, SLURM 17.02.11, and OpenLDAP 2.4.40. The access to the API is restricted

due to functional tests are being carried out, and the integration of WebSS with the Web platform of SC3, the research group in charge of Guane, is in process. It is expected that once the correct installation in Guane is made, the performance tests can be done, and the performance between sending a job through WebSS and sending a job using SSH can be compared.

4. Conclusions

This paper presented the development of the simple, fast, and safe web interface WebSS using C++ CGI that allows to submit jobs to SLURM, query the job queue, and creating or uploading batch files. It is limited to these functions due to it is still developing. Options as executing commands to cancel jobs, view their own directory, or view the job results are considered for future work.

Referencias

- [1] A. B. Yoo, M. A. Jette, and M. Grondona, "SLURM: Simple Linux Utility for Resource Management BT - Job Scheduling Strategies for Parallel Processing," in *Job Scheduling Strategies for Parallel Processing*, 2003, pp. 44–60.
- [2] Sun Yat-Sen University, "System Configuration," *National Supercomputer Center In Guangzhou*. 2018. [Online]. Available: <http://en.nsc-gz.cn/newsdetail.html?8527>
- [3] Lawrence Livermore National Laboratory, "Machine Catalog," *Computation*, 2018. [Online]. Available: <https://computation.llnl.gov/computing/machine-catalog>
- [4] Supercomputación y Cálculo Científico UIS, "Cluster Guane," *wiki_sc3*, 2017. [Online]. Available: http://wiki.sc3.uis.edu.co/index.php/Wiki_SC3
- [5] J. Schonwalder, G. Chulkov, E. Asgarov, and M. Cretu, "Session resumption for the secure shell protocol," in *2009 IFIP/IEEE International Symposium on Integrated Network Management*, 2009, pp. 157–163. doi: 10.1109/INM.2009.5188805
- [6] J. Andjarwirawan, H. N. Palit, and J. C. Salim, "Linux PAM to LDAP Authentication Migration," in *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIT)*, 2017, pp. 155–159. doi: 10.1109/ICSIT.2017.66

[7] M. Roberts and G. Torres, “pyslurm,” *Python Software Foundation*. 2018. [Online]. Available: <https://pyslurm.github.io/>

[8] Twitter, “Heron.” github. [Online]. Available: <https://github.com/apache/incubator-heron>

[9] D. Robinson and K. Coar, “The Common Gateway Interface (CGI) Version 1.1.” Oct-2004. [Online]. Available: <https://www.rfc-editor.org/info/rfc3875>

[10] Free Software Foundation, “GNU Operating System,” *gnu*. [Online]. Available: <https://www.gnu.org/software/cgicc/index.html>

[11] W. W. Royce, “Managing the development of large software systems: concepts and techniques,” in *Proceedings of the 9th international conference on Software Engineering*, 1987, pp. 328–338.

[12] E. Rescorla, *SSL and TLS: designing and building secure systems*, vol. 1. Addison-Wesley Reading, 2001.