



# SLURM plugin: a Jenkins plugin to connect Anvil with SCARF

E Chadwick, G Poulter, A Kyffin, C Jones

June 2019

©2019 Science and Technology Facilities Council



This work is licensed under a [Creative Commons Attribution 4.0 Unported License](https://creativecommons.org/licenses/by/4.0/).

Enquiries concerning this report should be addressed to:

RAL Library  
STFC Rutherford Appleton Laboratory  
Harwell Oxford  
Didcot  
OX11 0QX

Tel: +44(0)1235 445384  
Fax: +44(0)1235 446403  
email: [libraryral@stfc.ac.uk](mailto:libraryral@stfc.ac.uk)

Science and Technology Facilities Council reports are available online at: <http://epubs.stfc.ac.uk>

**ISSN 1358-6254**

Neither the Council nor the Laboratory accept any responsibility for loss or damage arising from the use of information contained in any of their reports or in any communication about their tests or investigations.

# SLURM Plugin: A Jenkins plugin to connect Anvil with SCARF

Eli Chadwick<sup>1</sup>, Gemma Poulter<sup>1</sup>, Alan Kyffin<sup>1</sup>, Catherine Jones<sup>1</sup>

June 5, 2019

## Abstract

This report presents a new Jenkins plugin which allows jobs to be submitted to HPC systems running the SLURM job scheduler. This plugin will be used to create a link from Anvil, a Jenkins-based service for STFC staff, to SCARF, an HPC cluster at STFC. Integrating the plugin into Anvil will enable users to run tests of their code on SCARF via Anvil jobs. A number of current Anvil users have an interest in using such a feature, and early feedback on the plugin has been positive. There is potential to extend the plugin to other batch systems, which leads to a long-term goal of making other HPC systems such as ARCHER available through Anvil.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Existing Work</b>	<b>2</b>
2.1	Python Framework . . . . .	2
2.2	Jenkins Plugins . . . . .	3
<b>3</b>	<b>Design</b>	<b>3</b>
3.1	Requirements . . . . .	3
3.2	Class Structure . . . . .	4
<b>4</b>	<b>Usage</b>	<b>4</b>
4.1	Build step . . . . .	4
4.2	SLURM Agent . . . . .	6
4.3	Current users . . . . .	6
<b>5</b>	<b>Future</b>	<b>6</b>
5.1	Possible Developments . . . . .	6
5.2	Possible Usage . . . . .	6
<b>6</b>	<b>Summary</b>	<b>7</b>
<b>7</b>	<b>Acknowledgements</b>	<b>7</b>

---

<sup>1</sup>Scientific Computing Department, Science and Technology Facilities Council, Rutherford Appleton Laboratory, Harwell Campus, Didcot, OX11 0QX, United Kingdom.

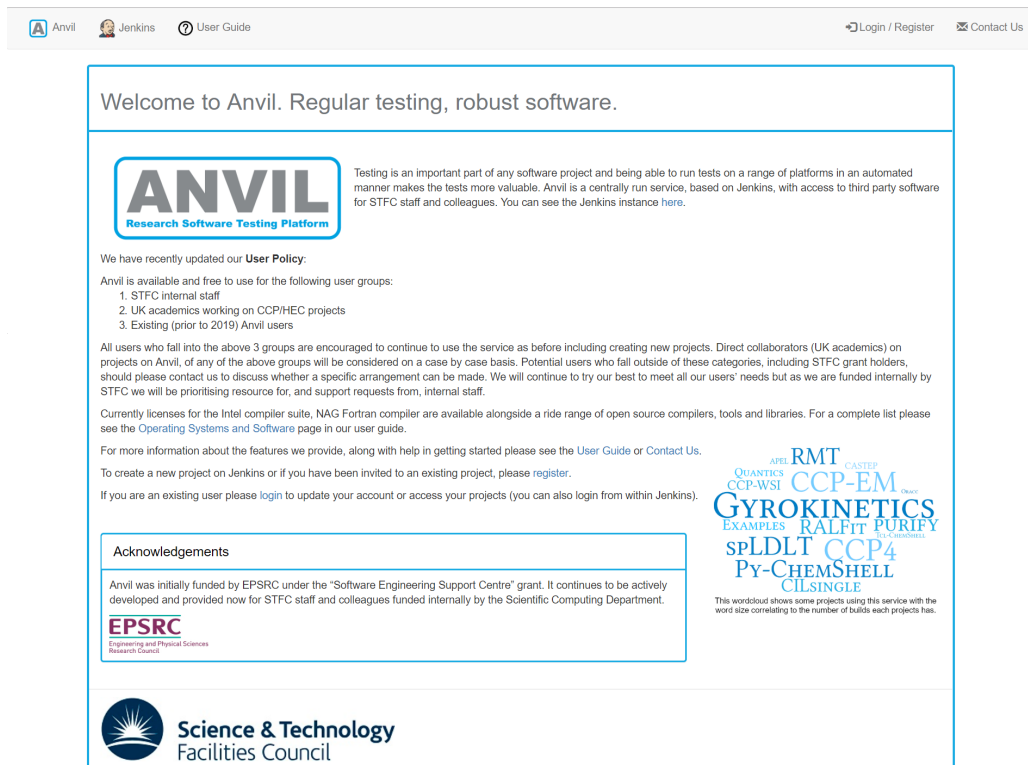


Figure 1: Anvil home page.

## 1 Introduction

The Software Engineering Group (SEG), part of the Scientific Computing Department (SCD) at STFC, develops and runs Anvil, a service which STFC internal staff can use to run tests of their software. Anvil [1] is built on Jenkins [2], with additional access to third-party software, virtual machines in the SCD Cloud, and hardware purchased specifically for Anvil.

Numerous Anvil users develop software that is run on HPC systems, and some of those users have expressed direct interest in running their tests in an HPC environment through Anvil. A link has therefore been developed between Anvil and SCARF, an HPC cluster at STFC, with the aim of allowing Anvil jobs to be run on HPC.

## 2 Existing Work

### 2.1 Python Framework

A Python framework for cloning a repository and running compilation on a PBS system was developed by Jan Hybš [3]. In this framework, the user creates a YAML configuration file with instructions for compiling and testing their code, which is placed into the framework repository.

The whole framework is cloned to HPC, where a script is run to launch the compilation on either the login node or a compute node (via a PBS job).

While this method works, the framework does not currently contain options for requesting resources (number of nodes, walltime, queue etc.).

## 2.2 Jenkins Plugins

Plugins are a highly customisable way to add features to Jenkins. Three plugins providing links to HPC already exist on GitHub, implementing job submission for PBS [4], LSF [6], and SGE [5]. However, all of these are developed for outdated versions of Jenkins, and are incompatible with newer versions. The PBS and LSF plugins are also no longer maintained, though development is continuing on the SGE plugin. As plugins have deep integration with the Jenkins UI, and these references exist for any new development, the plugin approach was selected over the Hybš framework.

## 3 Design

A new Jenkins plugin was developed for the SLURM batch system used by SCARF, taking inspiration from the PBS and LSF plugins.

### 3.1 Requirements

Anvil users with an interest in running HPC jobs through Anvil were invited to discuss their needs from such a feature. From these discussions, the following requirements were drafted.

- **Input:** through Jenkins job configuration interface, e.g. by selecting an option to run the job on HPC, and then entering relevant options for HPC configuration. Ideally compatible with Jenkins Pipeline script format.
- **Output:** as similar as possible to existing job output in Anvil. Different stages of the build should be timed and have output noted. Should be able to view the console output from the HPC job.
- **Multiple nodes:** be able to run jobs in parallel across multiple nodes. Without this capability, there is little gain to be made over the existing Anvil service (the only improvement would be the ability to compile and test code in the HPC environment).
- **Adapt existing Jenkinsfiles:** be able to adapt existing Jenkinsfiles (which contain the job configuration) to run on HPC, rather than totally rewriting into a new format.
- **Multiple HPC configurations:** be able to use multiple different HPC configurations in a single pipeline (e.g. vary the number of nodes for a job).
- **Third party support:** support Git and Mercurial repository hosted on sites such as GitHub and Bitbucket.

The plugin design described below satisfies the majority of these requirements. Pipeline compatibility was not attainable, so multiple HPC configurations cannot be used in a single job, and existing Jenkinsfiles cannot be used. All the rest of the features above have been satisfied (as they are Pipeline-independent).

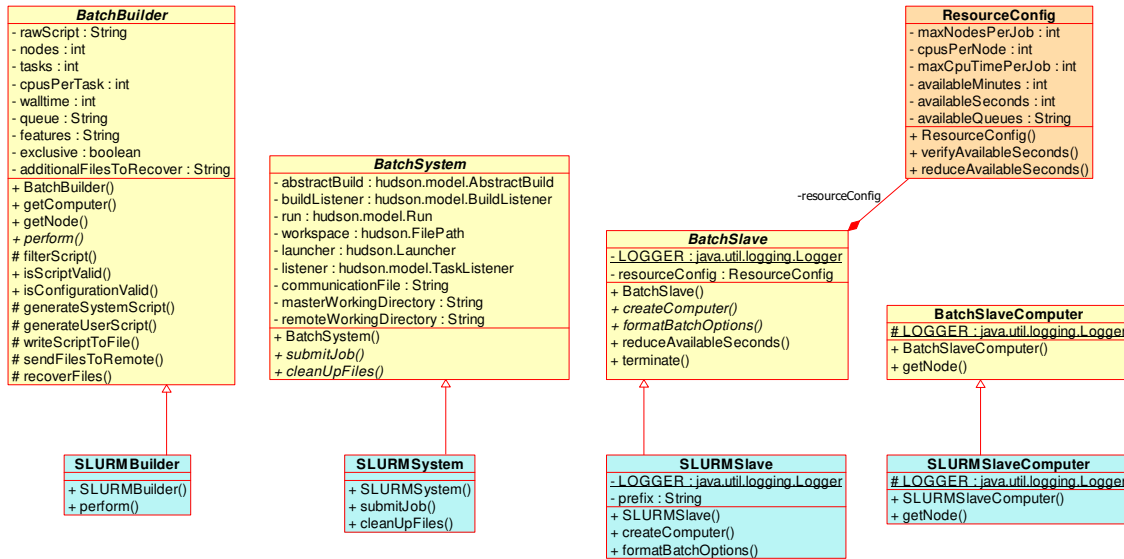


Figure 2: UML diagram for SLURM plugin. Extensible base classes are in yellow, with extensions for SLURM in blue. Method arguments have been omitted for simplicity.

### 3.2 Class Structure

The core plugin design includes 4 ‘base’ classes, plus some extra classes for storage of second-level options (as they appear in Jenkins). These classes are mostly abstract, and all are batch-system-agnostic; a set of 4 ‘SLURM’ classes implements them for the SLURM batch system specifically. The plugin is designed such that the base classes can be used as a starting point for extension to another batch system, with the SLURM implementation as a reference.

The UML diagram in figure 2 shows the structure of the plugin classes.

The SLURMBuilder class creates a new build step in Jenkins, which can be used in freestyle jobs. SLURMSlave creates a new agent type in Jenkins, which is used to connect to HPC systems such as SCARF and can store limits on resource usage. This agent launches SLURMSlaveComputer instances to run jobs. Finally, the SLURMSystem class provides the interaction with SLURM, to submit the job and process the job ID, exit code and CPU time used.

## 4 Usage

This plugin provides multiple new pieces of UI, including a new build step and a new agent type.

### 4.1 Build step

The new ‘Run SLURM script’ build step can be used in a freestyle job (the plugin is not Pipeline-compatible in its current form). The user enters a script with instructions for compiling and testing

**Build**

Run SLURM script X

Script

Nodes

Number of tasks

CPUs per task

Walltime required (minutes)

Queue

Node features

Exclusive mode

Files to recover

Standard output and error logs of the form *slurm-jobID* will be recovered automatically. You only need to enter the names of any additional output files you wish to recover.

Figure 3: Interface for SLURM build step.

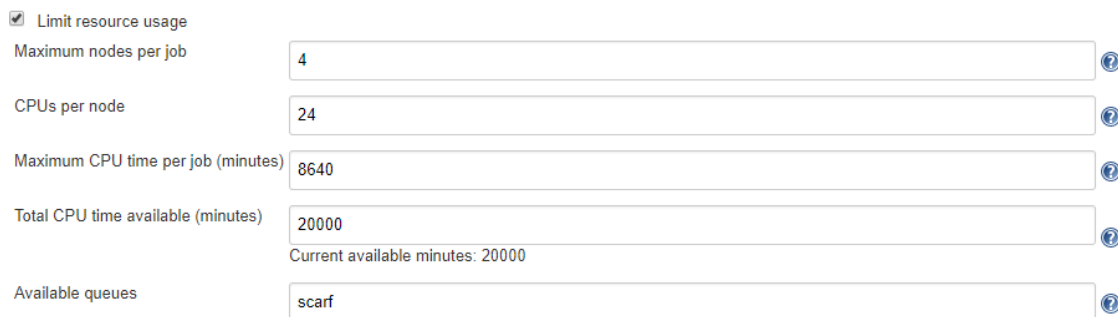
their code, then enters resource requirements for their job (see figure 3). The options that can be entered are:

- number of nodes
- number of tasks
- number of CPUs per task
- walltime
- queue/partition
- node features
- exclusive mode
- files to recover from the job, other than the standard SLURM logs

When the job is run, Jenkins checks the user configuration against any resource limits on the agent. It then packages all the options and the user-entered script and sends them to the HPC system. The job is submitted through SLURM, and Jenkins recovers the output files once this is complete. Users are able to view all the recovered output files on the build page for their job. Files recovered for previous builds are also accessible from their respective pages.

## 4.2 SLURM Agent

For Jenkins administrators, there is a new agent type, called ‘SLURM Agent’. All the configuration options available when setting up a standard agent are also available for a SLURM Agent, but a SLURM Agent can also store limits on resource usage (see figure 4). This includes the maximum number of nodes or amount of CPU time that a single job can use. The amount of time available on the system can also be stored, for cases where HPC administrators have only permitted a certain amount of use.



The image shows a configuration interface for a SLURM Agent. It features a checked checkbox labeled 'Limit resource usage'. Below this, there are five input fields, each with a help icon (a question mark in a circle) to its right. The fields are: 'Maximum nodes per job' with the value '4'; 'CPUs per node' with the value '24'; 'Maximum CPU time per job (minutes)' with the value '8640'; 'Total CPU time available (minutes)' with the value '20000', and a sub-label 'Current available minutes: 20000' below it; and 'Available queues' with the value 'scarf'.

Figure 4: Interface for resource configuration in SLURM Agent.

## 4.3 Current users

During the plugin’s development, three members of the Scientific Computing Department expressed interest in using the plugin to run SCARF jobs through Anvil. Two of these were successfully set up with test cases, using the plugin to compile and test parts of their code, and both individuals provided positive feedback for the plugin interface. Expectations are that when awareness is raised within the department of Anvil and its capabilities, demand for the plugin will increase significantly.

## 5 Future

### 5.1 Possible Developments

Further developments are possible which would improve the plugin’s features. For example, the script input could be improved by having an option to upload a script file from a Git repository, in a similar way to how Pipeline scripts can be loaded. Pipeline compatibility would also be a useful feature. With more users, it is likely that more features will be requested.

### 5.2 Possible Usage

The plugin structure offers opportunity for extension to other batch systems, such as PBS. Such extensions would provide the potential to run jobs on systems like ARCHER, Thomas or Isambard. A broader user base for Anvil, combined with access to time on these systems, would allow HPC to be brought to people who might not otherwise have access to it. Instead of having to apply



individually for time on an HPC system just to run tests, members of smaller projects could join Anvil to run those tests on systems of their choice. If a special queue or project was negotiated for Anvil, then users might not have to wait as long to run those tests either. Essentially, there is real scope for improving access to HPC for smaller projects, which is an exciting prospect for research software engineering in the UK.

## 6 Summary

The SLURM Plugin is a new plugin for Jenkins that allows users to submit their jobs to an HPC system which runs SLURM. In the Scientific Computing Department at STFC, this plugin will be incorporated into Anvil, a Jenkins-based service maintained by SEG, allowing jobs to be run on SCARF. Some interest in the new plugin already exists in SCD, and hopefully this interest will expand when the plugin is rolled out.

The plugin is designed such that it can be forked and developed for other batch systems as required. In the long term, the Anvil developers aim to reach out to services such as ARCHER and discuss making their systems available to Anvil users in addition to SCARF via adaptations of this plugin.

The plugin code is available at <https://gitlab.com/sesc-build-service/hpc/tree/master/slurm-plugin>.

## 7 Acknowledgements

This work was carried out as a graduate scheme project. It was funded by EPSRC as a part of the wider Software Engineering Support Centre (SESC) project. Future developments will be funded by SCD. Many thanks to the SESC team for their valuable feedback during the project.

## References

- [1] Anvil. <https://anvil.softeng-support.ac.uk/>. Last accessed May 2019.
- [2] Jenkins. <https://jenkins.io/>. Last accessed May 2019.
- [3] Jan Hybš. ci-hpc. <https://github.com/janhybs/ci-hpc>. Last accessed May 2019.
- [4] Bruno P. Kinoshita. pbs-plugin. <https://github.com/biouno/pbs-plugin>. Last accessed May 2019.
- [5] John McGehee. sge-cloud-plugin. <https://github.com/jenkinsci/sge-cloud-plugin>. Last accessed May 2019.
- [6] Laisvydas Skurevičius. lsf-cloud. <https://github.com/LaisvydasLT/lsf-cloud>. Last accessed May 2019.