

Federation & Interoperability Use Cases

August 20, 2019

Version 1.1

These use cases describe how scientific projects make use of resources from more than one public research computing community, and how a given community can support such projects. A public research computing community is an organized set of resources and processes that work together coherently to serve the needs of a community of researchers. Many such communities exist worldwide, such as XSEDE (www.xsede.org) and PRACE (www.prace-ri.eu). Each community has computing resources (e.g., an HPC cluster or an archival storage service) and a set of services that add value to the community's resources (e.g., a community login service or a user support service).

Although each use case describes an activity that uses resources from multiple communities, the use cases themselves are written from the point of view of a single community. The goal is to express what a single community must offer to support the activity.

[FI-01: Conduct a high-throughput computing project using resources from multiple communities](#)

[History](#)

FI-01: Conduct a high-throughput computing project using resources from multiple communities

A **researcher** needs to conduct a high-throughput computing project using resources from multiple communities so the project can be completed as quickly as possible. The project consists of a set of loosely coupled application runs. We'll refer to these application runs as "jobs" and the entire set of jobs as the "project." (A project is also often known as an "ensemble," or a "workflow").

We assume the following things are true.

1. The researcher is registered with the community and belongs to a project that has received at least one allocation of community resources.
2. Any project allocations are based on an existing software application and reasonable estimates for the required resources. (CPU hours, RAM, storage, etc.)
3. The application is already installed and validated on the allocated resources. (*Use case HPC-01 describes the process of installing and validating an application on a community resource.*)
4. The inputs to every job are available before any jobs are submitted. Jobs can be run in any order or simultaneously. (Note variation below.)

5. The researcher is using a workflow manager application to manage the project's jobs.

In most cases, the researcher wants to experience it as follows.

1. First, the researcher defines the jobs that need to be run.
2. Then, the researcher logs into a workflow manager application using the researcher's community identity AND the researcher's identities in any other communities.
3. Then, the researcher enters the job descriptions into the workflow manager.
4. Then, the researcher instructs the workflow manager to begin submitting jobs. At this point, the workflow manager begins monitoring the status of each resource and submitting jobs to appropriate resources, monitoring the status of each job, and collecting job results when they complete. This continues until all the project's jobs have completed.
5. While the jobs are queued and/or running, the researcher monitors the project status using the workflow manager's status interface.
6. While the jobs are queued and/or running, the researcher may pause execution, cancelling any queued or running jobs on resources and returning them to the workflow manager's set of pending jobs.
7. While the jobs are queued and/or running, the researcher may cancel execution, cancelling any queued or running jobs on resources and emptying the workflow manager's set of pending jobs.
8. When all of the project's jobs have completed, the researcher uses the results to complete the researcher's research goals. The researcher will acknowledge the community, the workflow manager application, and each resource's service provider in publications resulting from the project, and will cooperate with community requests for help with reports, science stories, etc. based on the results of the project.

It will always work like this except when in Step 4, there are dependencies between the jobs such that some jobs need to be run on the same resource and/or simultaneously, or the inputs to some jobs include the outputs produced by other jobs. In that case, everything works the same, but the workflow manager must take these dependencies into account when submitting jobs.

We'll take any solution, as long as the following are true.

1. The researcher can use the solution to process several million jobs per project.
2. In Steps 1 and 3, the workflow manager accepts job descriptions in one or more common formats, such as DRMAA (<https://www.drmaa.org/>) and equivalent.
3. In Steps 1 and 3, the researcher doesn't need to specify the resource on which each job will run. The workflow manager makes these decisions dynamically.
4. In Step 2, the user experience for logging into the workflow manager using a community identity is as described in use case IDM-06 or IDM-07. (IDM-06 for web applications, IDM-07 for locally installed applications.)
5. In Step 4, the workflow manager can obtain credentials that allow job staging and submission on each resource as described in use case CAN-06; can dynamically obtain configuration and status information from each resource as described in use case CAN-07; and can submit and manage jobs on each resource as described in use case CAN-01.
6. In Step 4, the community's information service (integrated with each community resource) provides sufficient resource status information to enable the workflow manager to make

appropriate decisions about where and when to submit each job to minimize the overall project runtime given the available resources.

7. In Step 4, the community's job submission interface provides sufficient job status information for the workflow manager to know when jobs have completed successfully vs. completed with errors. Error conditions should be sufficiently clear to allow the application to know whether the job may be resubmitted to the same resource (transient resource issue), resubmitted to another issue (inappropriate resource match), or flagged for the researcher to fix (bad job description).
8. In Step 5, the researcher is able to determine the overall project status using the workflow manager in 5 minutes or less.

History

	Version	Date	Changes	Author
First use case draft	0.1	12/01/2012	Document created	Jha, Weidner
Draft update on feedback	0.2	12/04/2012	List of science use cases; glossary; draft development of first 2 use cases	Jha, Weidner, Hossain
Draft update on feedback	0.3/1.0	04/22/2013	Formatting, refinement of use cases	Weidner
Entire Document	1.1	08/20/2019	Reformatted to XSEDE-2 use case format; removed unnecessary XSEDE terminology.	L. Liming