High-Performance Computing Use Cases

August 8, 2019

Version 1.1

These use cases describe the most common ways in which researchers use high-performance computing (HPC) resources. Each HPC resource is designed, constructed, and operated by a *service provider (SP)* organization, such as the Pittsburgh Supercomputing Center (PSC) or University of Michigan. An HPC resource may contribute services to one or more *public research computing communities*, such as XSEDE or Open Science Grid (OSG).

These use cases focus on the experiences researchers in a given community have with the HPC resources in that community. Using an HPC resource is most often part of a larger research process, so these use cases also mention the community's website and documentation, registration and account management services, allocation services, and tools to help move code and data into and out of individual HPC resources.

HPC-01: Use a single HPC resource for a research project

A **researcher** needs to use a single HPC resource to carry out a research computing program. We assume the following things are true.

1. The researcher is registered with the community and manages a project that has received an allocation on the HPC resource.
2. The project allocation is based on an existing application and reasonable estimates for the required computing resources (CPU hours, RAM, storage, etc.).
3. The project plan is to employ the HPC resource's manual interfaces, as opposed to workflow engines, a science gateway, or other automation. (See HPC-03, HTC-*, SW-*, and SGW-* use cases for other use modes.)
4. The project doesn't require dynamic data sources, such as streaming data from an instrument. (See DA-* use cases for other use modes.)

In most cases, the researcher expects it to work as follows.

1. First, the researcher studies the documentation for the HPC resource to understand the resource's basic interface. As part of this, the researcher determines which application setup tasks (e.g., file transfers, job submissions, output analysis and staging, data management) to perform locally on the HPC resource, and which tasks are easier to handle with other community services (e.g., the community's user portal, a community data transfer service, a community job management system) or with the researcher's local (personal or campus) computing systems.
2. Then, the researcher transfers the application to the HPC resource, if it isn't already installed and configured.
3. Then, the researcher builds, tests, debugs, and measures the performance of the application on the HPC resource and decides whether the application requires any changes for correctness or better job performance ("optimization") on the HPC resource.
4. If necessary, the researcher optimizes the application for the HPC resource until the researcher determines that the project can be completed within the allocated time and SU budget.
5. Then, the researcher formulates the sequence of application runs required to execute the project and begins submitting the jobs for execution using the resource's interface.
6. While the project's jobs are executing, if the researcher experiences deviations from the expected resource behavior, the researcher reports them to the community's user support team.
    a. The support team may follow up with the researcher to diagnose the issue.
    b. The researcher cooperates with these follow-ups.
    c. The support staff resolves the issue and notifies the researcher of the resolution.
7. While the project's jobs are executing, the researcher monitors the community's news service and the resource's interface for any changes that might require changes to the application or the job execution, such as resource down times, malfunctions, resource policy changes, or resource reconfigurations.
8. While the project's jobs are executing, the researcher manages the results of the jobs (including performance measurements).

9. When the project is complete, the researcher uses the results to complete the researcher's research goals. The researcher will acknowledge the community and the 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 be like this, except when the researcher requests community assistance with optimizing the application for the HPC resource. In that case, enlisted community members may assist in Steps 3 and 4, and possible 5.

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

1. Projects that last one year can be supported.
2. In Steps 3 and 4, the HPC resource provides diagnostic tools to help measure application performance. These are described in the resource's documentation. (See Step 1.)
3. In Step 6, the community support mechanisms engage the HPC resource's SP organization as needed to resolve issues.
4. In Step 7, the community's news service communicates all relevant changes in the HPC resource's status.

HPC-02: Use two or more HPC resources for a research project

A **researcher** needs to use multiple HPC resources to carry out a research computing program. We assume the following things are true.

1. The researcher is registered with the community and manages a project that has received allocations on each of the HPC resources.
2. The project allocation is based on a set of existing applications and reasonable estimates for the required computing resources (CPU hours, RAM, storage, etc.).
3. The project plan is to employ the HPC resources' manual interfaces, as opposed to workflow engines, a science gateway, or other automation. (See HPC-03, HTC-*, SW-*, and SGW-* use cases for other use modes.)
4. The project doesn't require dynamic data sources, such as streaming data from an instrument. (See DA-* use cases for other use modes.)

In most cases, the researcher expects it to work as follows.

1. First, the researcher studies the documentation for each of the HPC resources to understand their basic interfaces. As part of this, the researcher determines which application setup tasks (e.g., file transfers, job submissions, output analysis and staging, data management) to perform locally on each HPC resource, and which tasks are easier to handle with other community services (e.g., the community's user portal, a community data transfer service, a community job management system) or with the researcher's local (personal or campus) computing systems.
2. Then, the researcher transfers the applications to the appropriate HPC resources if the applications aren't already installed and configured.
3. Then, the researcher builds, tests, debugs, and measures the performance of the applications on the HPC resources and decides whether any of the applications require changes for correctness or better job performance ("optimization") on the HPC resources.

4. If necessary, the researcher optimizes the applications for the HPC resources until the researcher determines that the project can be completed within the allocated time and SU budget.
5. Then, the researcher formulates the sequence of application runs required to execute the project and begins submitting the jobs for execution using the resources' interfaces.
6. While the project's jobs are executing, if the researcher experiences deviations from the expected resource behavior, the researcher reports them to the community's user support team.
    a. The support team may follow up with the researcher to diagnose the issue.
    b. The researcher cooperates with these follow-ups.
    c. The support staff resolves the issue and notifies the researcher of the resolution.
7. While the project's jobs are executing, the researcher monitors the community's news service and the resources' interfaces for any changes that might require changes to the application or the job execution, such as resource down times, malfunctions, resource policy changes, or resource reconfigurations.
8. While the project's jobs are executing, the researcher manages the results of the jobs (including performance measurements).
9. When the project is complete, the researcher uses the results to complete the researcher's research goals. The researcher will acknowledge the community and the resources' service providers 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 be like this, except when the researcher requests community assistance with optimizing one or more applications for the relevant HPC resources. In that case, enlisted community members may assist in Steps 3 and 4, and possible 5.

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

1. In Steps 1-3 and in Step 8, the interfaces for the HPC resources are as consistent as possible given each resource's unique architecture. For example, it makes no sense to expect the same compilers or MPI libraries on resources with different architectures. However, it is reasonable to expect that each resource will offer "modules" to customize the software environment, that the resources will use similar module names for a given software package, that the resources will use similar paths for libraries and 3rd party applications, and that the resources will use similar endpoints and mechanisms for data transfers.
2. Projects that last one year can be supported.
3. In Steps 3 and 4, the HPC resources provide diagnostic tools to help measure application performance. These are described in each resource's documentation. (See Step 1.)
4. In Step 6, the community support mechanisms engage the HPC resources' SP organizations as needed to resolve issues.
5. In Step 7, the community's news service communicates all relevant changes in the HPC resources' status.

HPC-03: Use a workflow system to conduct a project on HPC resources

A researcher needs to use a workflow system to carry out a research computing program on one or more HPC resources. The workflow system orchestrates the submission of individual jobs on the HPC resources using a plan provided by the researcher. The workflow system may also use dynamic resource status information to optimize job submissions. We assume the following things are true.

1. The researcher is registered with the community and manages a project that has received allocations on each of the HPC resources.
2. The project allocation is based on a set of existing applications and reasonable estimates for the required computing resources (CPU hours, RAM, storage, etc.).
3. The project plan is to employ a workflow service that interfaces with the HPC resources' programmatic interfaces, as opposed to the manual interface provided by each resource. (See HPC-01 and HPC-02 for other use modes.)
4. The workflow system has already been integrated with the interfaces offered by each of the HPC resources and the integration has been verified.
5. The project doesn't require dynamic data sources, such as streaming data from an instrument. (See DA-* use cases for other use modes.)

In most cases, the researcher expects it to work as follows.

1. First, the researcher studies the documentation for the workflow system to understand its interfaces and usage modes. As part of this, the researcher determines the best way to use the workflow system to submit and manage jobs on the HPC resources.
2. Then, the researcher studies the documentation for each of the HPC resources to understand their basic interfaces. As part of this, the researcher determines which application setup tasks (e.g., file transfers, job submissions, output analysis and staging, data management) to perform locally on each HPC resource, and which tasks are easier to handle with other community services (e.g., the community's user portal, a community data transfer service, a community job management system) or with the researcher's local (personal or campus) computing systems.
3. Then, the researcher transfers the applications (including any required components of the workflow system) to the appropriate HPC resources if they aren't already installed and configured.
4. Then, the researcher builds, tests, debugs, and measures the performance of the applications on the HPC resources and decides whether any of the applications or the workflow system require changes for correctness or better job performance ("optimization") on the HPC resources.
5. If necessary, the researcher optimizes the applications (including the configuration of the workflow system) for the HPC resources until the researcher determines that the project can be completed within the allocated time and SU budget.
6. Then, the researcher formulates the sequence of application runs required to execute the project, configures the workflow system with the required runs, and initiates the workflow system.

7. While the project's jobs are executing, if the researcher experiences deviations from the expected resource behavior, or if the workflow system deviates from its expected behavior, the researcher reports them to the community's user support team.
    a. The support team may follow up with the researcher to diagnose the issue.
    b. The researcher cooperates with these follow-ups.
    c. The support staff resolves the issue and notifies the researcher of the resolution.
8. While the project's jobs are executing, the researcher monitors the community's news service and the resources' interfaces for any changes that might require changes to the application or the job execution, such as resource down times, malfunctions, resource policy changes, or resource reconfigurations.
9. While the project's jobs are executing, the researcher manages the results of the jobs (including performance measurements).
10. When the project is complete, the researcher uses the results to complete the researcher's research goals. The researcher will acknowledge the community, the resources' service providers, and the workflow service providers 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 be like this, except when the researcher requests community assistance with optimizing one or more applications (or the workflow service) for the relevant HPC resources. In that case, enlisted community members may assist in Steps 4 and 5, and possibly 6.

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

1. Projects that last one year can be supported.
2. In Steps 4 and 5, both the workflow system and the HPC resources provide diagnostic tools to help measure application performance. These are described in the documentation for the workflow system and for each HPC resource. (See Steps 1 and 2.)
3. In Step 7, the community support mechanisms engage the HPC resources' SP organizations and/or the workflow system's support personnel as needed to resolve issues.
4. In Step 8, the community's news service communicates all relevant changes in the HPC resources' status.

History

	Version	Date	Changes	Author
First use case draft	0.1	09/09/2012	Document created	Sanielevici, Fahey
Major revision based on feedback	0.2	10/22/2012	List of 4 use cases; glossary; draft development of first use case; flow chart	Sanielevici, Fahey, Hossain
Minor revision	0.3	10/28/2012	Trying to express what is distinctive about HPC	Sanielevici
Minor revision	0.4	2/4/2013	Based on comments heard on the Architecture call of 1/31	Sanielevici
Major revision, addition of second use case	0.5	4/22/2013	Details added on use case 2	Fahey
Minor revision	0.6	5/1/13	Edits to use case 2	Sanielevici
Minor revision	0.7	5/19/13	Suggested numerical targets for quality attributes.	Sanielevici
Major revision	0.81 / 1.0	6/17/13	Added use case 3	Sanielevici
Entire document	1.1	8/8/2019	Reformatted to XSEDE-2 format; standardized terminology with other use case areas; and removed unnecessary XSEDE terminology	L. Liming