

Visualization Use Cases

August 19, 2019

Version 1.5

These use cases describe the most common scientific data visualization methods. This is an evolving field of work, since the ability of desktop computers to visualize data is improving rapidly. But it is still common for researchers to need to visualize data at scales that exceed their local resources, requiring them to use high-performance and high-throughput computing resources to advance their work. Large-scale visualization resources must often be used remotely over a research network with the results shown on a local display.

Advanced visualization resources are designed, constructed, and operated by a *service provider (SP)* organization, such as the Texas Advanced Computing Center (TACC) or University of Utah. A visualization resource may be used in one or more *public research computing communities*, such as XSEDE or Open Science Grid (OSG).

[VIS-01: Visualize research data using streaming video](#)

[VIS-02: Visualize research data using streaming geometry data](#)

[VIS-03: Generate visualization data for later viewing](#)

[VIS-04: Visualize and steer a simulation running on a remote resource](#)

[VIS-05: Visualize a simulation as it runs on a remote resource](#)

[VIS-06: Visualize research data using a web application](#)

[History](#)

VIS-01: Visualize research data using streaming video

A **researcher** needs to visualize a specific set of research data using an application running on a remote visualization resource. The application provides a streaming video display.

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 visualization resource.
2. The visualization is non-interactive.
3. The application is installed and validated on the visualization resource.
4. The data the researcher needs to visualize can be accessed by the visualization resource.
5. The application uses OpenGL libraries for display.
6. The researcher has installed VNC on a local display system.
7. The network between the researcher's local display system and the application running on the remote visualization resource has sufficient bandwidth to transmit 10 frames/second.

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

1. First, the researcher opens a command shell on the visualization resource.
2. Then, the researcher submits a visualization job to the resource's batch queue.
3. When the researcher's job begins running, the researcher obtains the display port number from a status file written by the job.
4. Then, the researcher launches the local VNC application and enters the visualization resource's network address and port number to connect to the video stream.
5. As long as the researcher's job runs, the researcher watches the video stream.

It will always work like this except when the researcher doesn't have a local VNC application. In that case, in Steps 4-5, the researcher may instead use a Java-enabled web browser and use a Java application provided by the visualization resource to connect to and display the video stream.

We'll take any solution, as long as in Step 1, the experience for opening a command shell is as described in use case CAN-04.

VIS-02: Visualize research data using streaming geometry data

A **researcher** needs to visualize a specific set of research data using an application running on a remote visualization resource. The application provides geometry streaming and the geometry is rendered on a local display.

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 visualization resource.
2. The visualization is non-interactive.
3. The application is installed and validated on the visualization resource.
4. The data the researcher needs to visualize can be accessed by the visualization resource.

5. The resource being used must provide software capable of delivering geometry to the researcher's local resource.
6. The researcher's local display system must have a graphics card and software capable of performing the local rendering.
7. The network between the researcher's local display system and the remote visualization resource has sufficient bandwidth to transmit 10 frames/second.

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

1. First, the researcher installs any display software required by the application onto the researcher's local display system.
2. Then, the researcher opens a command shell on the visualization resource.
3. Then, the researcher submits a visualization job to the resource's batch queue.
4. When the researcher's job runs, the researcher connects to it using the display software.
5. As long as the researcher's job runs, the researcher watches the visualization on the local display system.

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

1. In Step 1, the display software can be installed on a laptop or desktop owned/administered by the researcher.
2. In Step 2, the experience for opening a command shell is as described in use case CAN-04.
3. In Step 4, the solution must be able to handle a standard NAT/firewall on the researcher's local network.

VIS-03: Generate visualization data for later viewing

A **researcher** needs to generate a set of visualizations using an application on a compute resource so the researcher can view the visualizations later.

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 compute resource.
2. The application is non-interactive.
3. The application is installed and validated on the compute resource.
4. The inputs for the application can be accessed by the compute resource.

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

1. First, the research opens a command shell on the compute resource.
2. Then, the researcher submits one or more visualization jobs to the resource's batch queue. (At this point, the researcher may disconnect from the resource.)
3. When the researcher's jobs run, the visualization application generates visualization products on the compute resource.
4. At a later time, the researcher accesses the visualization products.

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

1. In Step 1, the experience for opening a command shell is as described in use case CAN-04.

2. In Step 4, the researcher may transfer the visualization products as described in use case CAN-02, or the researcher may view the visualizations remotely as described in use cases VIS-01 or VIS-02.

VIS-04: Visualize and steer a simulation running on a remote resource

A **researcher** needs to steer a simulation running on a community resource by interacting with a visualization of the simulation. To “steer” means to adjust the parameters used for the remainder of the simulation.

We assume the following are true.

1. The researcher is registered with the community and manages a project that has received an allocation on the resource.
2. Both the simulation application and the visualization application are installed and validated on the resource.
3. The inputs for the simulation application can be accessed by the resource.
4. The simulation application can be steered while it runs by adjusting one or more parameter files used by the application.
5. A visualization application can generate a visualization of the current state of the simulation using files generated by the simulation application.
6. The visualization can be viewed remotely as described in use case VIS-01 or use case VIS-02.

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

1. First, the researcher installs any display and/or control software required by the application onto the researcher’s local display system.
2. Then, the researcher opens a command shell on the compute resource.
3. Then, the researcher submits a job to the resource’s batch queue that includes both the simulation and the visualization application so they run at the same time and can communicate with each other.
4. When the researcher’s job runs, the researcher connects to it using the display and control software on the local display system.
5. While the researcher’s job runs, the researcher interacts with the job using the display and control software on the local display system.

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

1. In Step 1, the display software can be installed on a laptop or desktop owned/administered by the researcher.
2. In Step 2, the user experience for opening a command shell is as described in use case CAN-04.
3. In Step 4, the solution must be able to handle a standard NAT/firewall on the researcher’s local network.

VIS-05: Visualize a simulation as it runs on a remote resource

A **researcher** needs to visualize a simulation as it runs on a remote resource. (This is often called *in situ* visualization because the visualization happens as part of the simulation.)

We assume the following are true.

1. The researcher is registered with the community and manages a project that has received an allocation on the resource.
2. The simulation application is installed and validated on the resource.
3. The inputs for the simulation application can be accessed by the resource.
4. The simulation application generates a visualization of its current state while it runs.
5. The visualization can be viewed remotely as described in use case VIS-01 or use case VIS-02.
6. The network between the researcher's local display system and the application running on the remote visualization resource has sufficient bandwidth to transmit 10 frames/second.

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

1. First, the researcher installs any display software required by the application onto the researcher's local display system.
2. Then, the researcher opens a command shell on the resource.
3. Then, the researcher submits a simulation job to the resource's batch queue.
4. When the researcher's job runs, the researcher connects to the simulation's visualization stream using the display software.
5. As long as the researcher's job runs, the researcher watches the visualization on the local display system.

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

1. In Step 1, the display software can be installed on a laptop or desktop owned/administered by the researcher.
2. In Step 2, the experience for opening a command shell is as described in use case CAN-04.
3. In Step 4, the solution must be able to handle a standard NAT/firewall on the researcher's local network.

VIS-06: Visualize research data using a web application

A **researcher**, **educator**, or **student** needs to use a web application that visualizes research data. The researcher connects to the web application using a web browser. (Note: Such web applications are also referred to as "science gateways" or "visualization gateways.")

We assume the following things are true.

1. The researcher is registered with the community.
2. The web application is available online and already integrated with any visualization resources it requires.
3. The inputs required by the visualization can be accessed by the web application.

4. The network between the researcher's web browser and the web application has sufficient bandwidth to transmit 10 frames/second.

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

1. First, the researcher opens a web browser, navigates to the web application, and logs in to the web application.
2. Then, the research interacts with the web application, choosing a visualization tool or application and specifying any required inputs.
3. Then, the web application presents the visualization in its web interface (within the browser).

It will always be like this unless the web application requires that its users have their own allocations to use a community resource OR the researcher wants the web application to use the researcher's allocation. In that case, the researcher must have an allocation to use one of the resources the web application is integrated with, and the researcher must first add the web application's community identity to the researcher's allocation project and notify the web application operator to use the project when the researcher uses the web application. Then the use case works as described above.

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

1. The only software the researcher needs to have on the local machine is a standard web browser. (The browser may need to have one or more plug-ins installed, such as Java, Flash, or a plug-in provided by the web application.)
2. In Step 1, the login experience is as described in use case IDM-06. (IDM-06 is a version of use case CAN-06 specialized to web applications.)

History

| | Version | Date | Changes | Author |
|--------------------------|---------|-----------|---------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
| First use case draft | 1.3 | 3/5/2014 | Document created | Kelly Gaither, Sean Ahern, David Bock |
| Final XSEDE-1 version | 1.4 | 7/24/2014 | ? | ? |
| Entire document | 1.5 | 8/19/2019 | Reformatted to XSEDE-2 format; standardized terminology with other use case areas; and removed unnecessary XSEDE terminology | L. Liming |