“Science Studio”: Remote control of ‘big science’ facilities and data processing/data storage requirements

N. Stewart McIntyre1, Michael Bauer1, Elder Matias2, Chris Armstrong3, John Haley4, Dominic Lam5, Yan Yuhong5, Marina Suominen Fuller3

A Joint project between The University of Western Ontario1, Canadian Light Source2, Concordia University3, IBM Canada4 and SHARCNET5

ABSTRACT

“Science Studio” is one of the Canadian Network-Enabled Platform (NEP) projects that will be funded through CANARIE in the NEP area. The goal of the “Science Studio” project is to create a complete experiment management system that will allow researchers to control and observe, from their home base, all aspects of research that must be carried out at specialized laboratories throughout Canada. “Science Studio” will be built around a newly-developed set of web services software that permit remote access to and control of experiments, easy sharing of results with collaborators, the automation of the peer review process, scheduling of user time, access to the most recent data treatment programs at different computational facilities as well as the potential for integration of results from the use of several techniques. The system would be used to integrate and ensure collaboration with diverse organizations, systems, devices, and information systems to effectively meet the needs of researchers and more effectively leverage the joint capabilities of participating organizations.

“Science Studio” will be a distributed system that provides the end user with a common interface to the devices and analysis programs they need to run scientific experiments. The User Interface (UI) will be run from a standard web browser and will communicate with the Beamline Services and applications over HTTP. “Science Studio” will enable users to move data from the devices to a high performance computing (HPC) facility such as SHARCNET, for processing and storage. To an outside user, this distributive application will look like one application, even though the data may be coming from different databases and the devices are located in different facilities.

Part of the “Science Studio” initiative will make use of CANARIE’s lightpaths to move data from the CLS across the country to SHARCNET for analysis and display at the scientists workstation. To make such processing “near real time”, the project will make use of analysis software to be developed for the Cell processors that SHARCNET has recently acquired. The aim is to combine the web services, lightpaths and stream computing using the Cell processors to provide view of the beam line experiment as it runs. This could enable the scientist to make adjustments to the experiment, change parameters or location on the specimen, etc. The overall approach makes the “big facilities” seem as if next door and provides the feedback that researchers need to optimize the use of their time on the facilities.

ARCHITECTURE

As illustrated in the diagram above, all access will be through a common portal at CLS. This portal will handle the presentation services for all services regardless of where the services are located. The services will reside on application servers on multiple nodes, Experiment Management Services that are specific to a facility/device can reside at that another facility or they can be located at the same facility as demonstrated in examples 1 and 2 in Figure 2 below. All communication that is within the facilities’ intranet will be accomplished using, JMS. Communication between servers that go over the Internet will use SOAP over HTTP. Distributed services crossing a network boundary will communicate using SOAP over HTTP (if it is over the internet) or the light path and SOAP over TCP/IP (if it is within a WAN). For example, within CLS, each Beamline runs off a separate network while any service existing outside of CLS will use SOAP as its interface. A different approach will only be used if a performance issue occurs.

Figure 1: High Level Component Diagram (above)

Example 1

Example 2

Figure 2: Two examples of using “Science Studio” (above)

Figure 3: “Science Studio Component and Service Architecture Overview (above)

The purpose of the above diagram is to outline the main components and services provided by “Science Studio” and the technology choices that have been made.

The services in the “Science Studio” have been classified into: Presentation Services, Business Application Services, made up of Experiment Management Services, and User & User Management Services. Common Services, Infrastructure Services and External Services.

BENEFITS TO SCIENCE IN CANADA

- Improve user access to some major Canadian science facilities.
- Enable researchers to move raw data easily from these major instruments to computational facilities and subsequent processing and visualization.
- Enhance collaboration and awareness of other related research in the structure of materials.
- More efficient use of large scientific facilities.

BENEFITS TO SCIENTISTS

- Cutting down on travel time and travel costs.
- Reducing the amount of duplicated data entry.
- Eliminating the time to fix errors incurred when failing to duplicate the data entry correctly.
- Removing the need to learn multiple system interfaces.
- Providing the information they need without having to look for it.
- Providing organization and ease handling of large amounts of data.

FACILITIES

Remote access will be developed for two scientific facilities across Canada:

1. VESPER’s beamline at the Canadian Light Source at the University of Saskatchewan
2. Nanotribulation Laboratory at the University of Western Ontario
3. SHARCNet
4. WestGrid

Facilities connected by high speed CANARIE Lightpath.

Figure 4: Application Layers (above)

The above diagram shows some of the product choices the project team is making and how these conceptually fit together. The Experiment Management Services, Application Services, Common Services (including Infrastructure Services), UI Services (Presentation Services), are all built on top of the Spring Application Framework. This framework provides “Inversion of Control” (IOC) to enable various components to communicate with one another with limited concern about implementation, location, or driving the dependencies across the code.

Figure 5: “Science Studio” installed at UWO