Portal Development for High Performance Computing (HPC) at Maui High Performance Computing Center (MHPCC)

David Morton, AFRL Director
Maui High Performance Computing Center DSRC
Maui, Hawaii

Abstract

The goal of this program is to transform DoD High Performance Computing (HPC) application delivery, for selected applications, from stand-alone to cloud-based. The benefits of this program to the end application user are manifold: a) eliminates traditional HPC application stove-pipes making application and data sharable world-wide on-demand; b) no installed software required eliminating user configuration and maintenance; c) simplified and enhanced security model d) does not require HPC knowledge or traditional hurdles to fully exploit HPC; e) provides immediate access to HPC resources using fieldable, low power, portable devices such as the iPad. Four stand-alone software product areas were specified for initial deployment:

- *Computational Research & Engineering Acquisition Tools and Environments (CREATE) Kestrel*
- *Distributed MATLAB*
- *Distributed TASAT*
- *Virtual Applications*

Figure 1 shows the current application suite startup panel which includes Kestrel, Distributed MATLAB, TASAT, and Virtual Applications. The application delivery technology is described in a subsequent section. For example, virtual applications provide a quick-start capability for existing or COTS programs and uses a VNC based technology referenced as Level 0 Application Delivery.

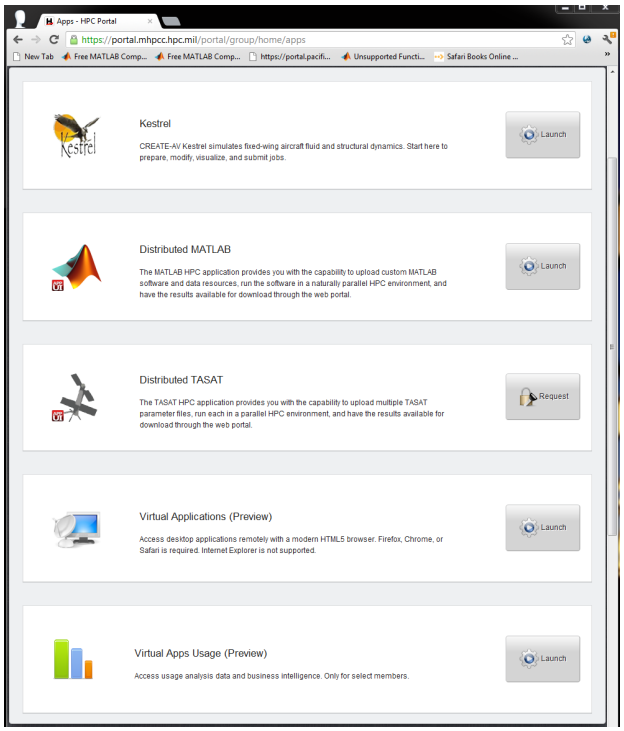

Figure 1. Application Startup Panel

1. Introduction

Last year's product roll out, which included Kestrel and Distributed MATLAB, were successfully demonstrated live at the 2011 NDIA Physics-based Modeling in Design & Development for U.S. Defense Conference in Denver CO, with the web-server and compute resources running at MHPCC in Maui HI. Since then a number of significant enhancements have been implemented for a November 2012 release.

An MHPCC team of consisting of University of Hawaii (UH) and Pacific Defense Solutions (PDS) developers were assembled in Q1 2011. This team has fashioned key alliances with the Army Research Laboratory (ARL), the USACE Engineer Research and Development Center (ERDC), and the Computational Research and Engineering Acquisition Tools and Environments (CREATE). Initially, the team is focused on the development of the necessary infrastructure. In Q1, 2012 MHPCC began offering on-demand Portal-based services for Pioneer users for distributed MATLAB and CREATE Kestrel. On 3-July-2012 an Authority to Operate (ATO) was received. Other applications including CREATE-AV DaVinci, CREATE-AV Helios, JSpOC Mission Systems (JMS) HPC Development Test Bed, and One Semi-Automated Forces (OneSAF) will be available beginning in Q2, 2012.

A Portal Software Development Kit (SDK) will be released in Q4 2012 to speed application transformation by both internal and external development teams. Continual enhancements to the SDK are planned for 2013 including a ReSTful service interface for language independence. MHPCC is planning to implement a support infrastructure leveraging existing HPCMP resources to train customers on the use of the Portal environment, provide customer assistance, and grow the user base for the Portal environment.

2. Long-Term Vision

While working to demonstrate specific application delivery of CREATE and MATLAB, the MHPCC team has adopted a longer term vision to guide the near-term development making sure the portal design is applicable and scalable to other domains in the future. As networks and browsers continue to improve cloud-based application delivery appears well suited to the DoD and may become the standard delivery mechanism in the future. Immediate benefits of application delivery with portal include: a) no client software required on users machine b) eliminates all user configuration and maintenance c) seamless collaboration within groups d) HPC scalability e) delivery is OS and device independent and f) simplified security model. Newer technologies are being incorporated into the portal today such as openID and oAuth which will facilitate secure access to the full power of HPC resources using low power, non-traditional portable devices such as the iPad. These same technologies will allow applications to scale across DSRCs without requiring user interaction.

3. Application Delivery

Application Delivery Levels for the portal classify how existing or new applications will be delivered within the portal environment. A higher numerical level delivery generally implies more integrated web-based client side architecture while a lower level implies a legacy (thick) client application was configured for quick portal deployment. A key objective in portal development is to provide all application access, regardless of delivery level, through a standard web-browser. Applications delivered in this manner are considered to provide a "zero footprint" in that there are no additional user-side applications or browser plug-ins required. Since most modern browsers support JavaScript and significant portions of the HTML-5 spec, only these intrinsic browser facilities are assumed. Examples of browsers with these capabilities include Internet Explorer 9 (IE9) and all recent versions of Chrome, Firefox, and Safari. **Level 0 Application Delivery**. Level 0 application delivery allows legacy applications which are not web based to be quickly implemented within the portal environment. Such legacy applications may currently require considerable client-side software and associated configuration. In Level 0, the server side software is assumed unchanged. The client-side software is also unchanged but is installed at the server site (e.g., at the HPC facility) using various options including:

- Installation on a Windows-based node (terminal server or single user)
- Virtualized machine (Windows or Linux) running on a Linux server

For example if the legacy application interface only operates in the windows environment, previously assumed to be running on the users windows box, the first step in the Level 0 solution is to install the client code on an HPC node using the first option or the second option with the virtual machine instantiation script invoking the required windows OS. The second step to delivering the application in Level 0 is to provide a client viewer to the, now remote, client using zero footprint HTML5 + JavaScript (AJAX) technologies (e.g., Guacamole). The goal of this effort is to provide an Information Assurance (IA) approved mechanism to reliably and efficiently provide this display remotely to the user.

Portal currently provides a Level 0 solution using an enhanced version of Guacamole which should allow nearly any application to be accessed from the portal with a very minimal development effort. In some cases, the actual user experience may suffer depending on the complexity of the display output bandwidth, and/or the technology used in the initial client-side development. Even so, there will be immediate benefit to implementing a Level 0 solution including:

- No client software required on users machine
- Eliminates all user configuration and maintenance
- Legacy CREATE applications up and running quickly

Figure 2 shows the virtual application menu, and the result of selecting FieldView, a non-web based, commonly used commercial application, delivered via Portal to user's browser using the portal zero footprint technology. For DoD users which do not currently have access to an HTML-5 compliant browser, for example IE7 or IE8, a signed Java Plug-in was developed and is available for download which provides access to the Virtual Apps.

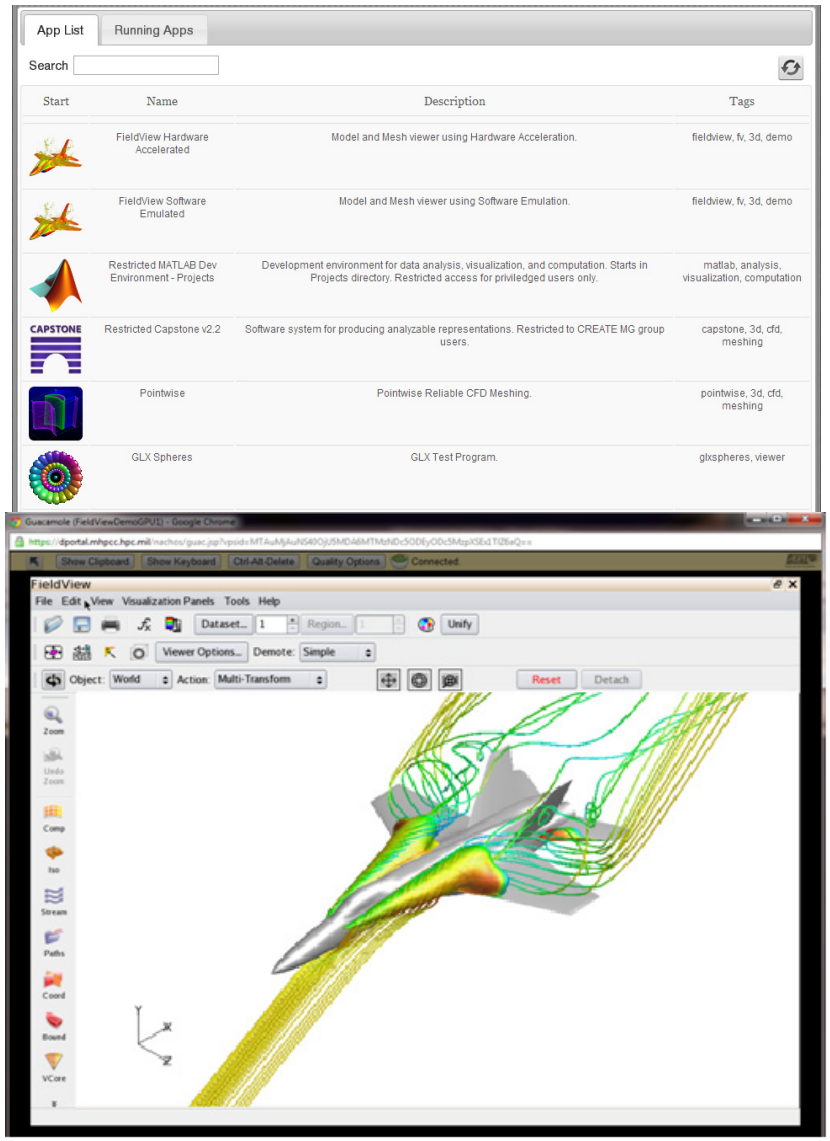

Figure 2. Virtual App Menu and FieldView Delivered Using Zero Footprint Guacamole (Level 0)

Level 1 Application Delivery. Level 1 application delivery is considered a hybrid between a clean slate web-based application design and the Level 0 solution. For example, many legacy CREATE applications are based on sound design concepts, such as the model view controller (MVC) architecture, which separate the view from control logic. Instead of moving the entire client to the server side using, for example, Level 0 virtualization, only the control logic need to migrate to the server side. Serializable message passing such as eXtensible Markup Language (XML) is typically used to communicate between the browser and the server application. Once this separation is complete, the Level 0 virtualization is unnecessary and its viewer is no longer needed since the user interface components are implemented directly using AJAX technologies within the browser.

Level 2 Application Delivery. This represents a native HTML5/JavaScript web-based application. For future CREATE (other) applications intended to be integrated with the portal, the goal is to provide a bandwidth-aware software development kit (SDK) to assist with the development process. Since viewing large datasets is often integral to tasks that demand HPC, efficient technologies to support changing level of detail (e.g., mesh viewing for CREATE Air Vehicles (AV) applications) are required. We are exploring existing open source toolkits such as Visual Toolkit (VTK), and ParaViewWeb as a basis for developing a bandwidth-aware SDK. Web-GL capabilities (example: http://www.chromeexperiments.com/globe) are being developed for graphically intense applications such as CREATE Capstone, which includes a mesh viewing/editing capability. For legacy applications, the SDK will help facilitate a migration path from Level 0 or 1 to the preferred Level 2 delivery.

Web Shell Access

An important new feature of our HPC-Portal is the ability to access a traditional shell capability as shown in Figure 3. For legacy or power users this provides the ability to directly access HPC resources, within a browser, without requiring installation of a Kerberos client kit.

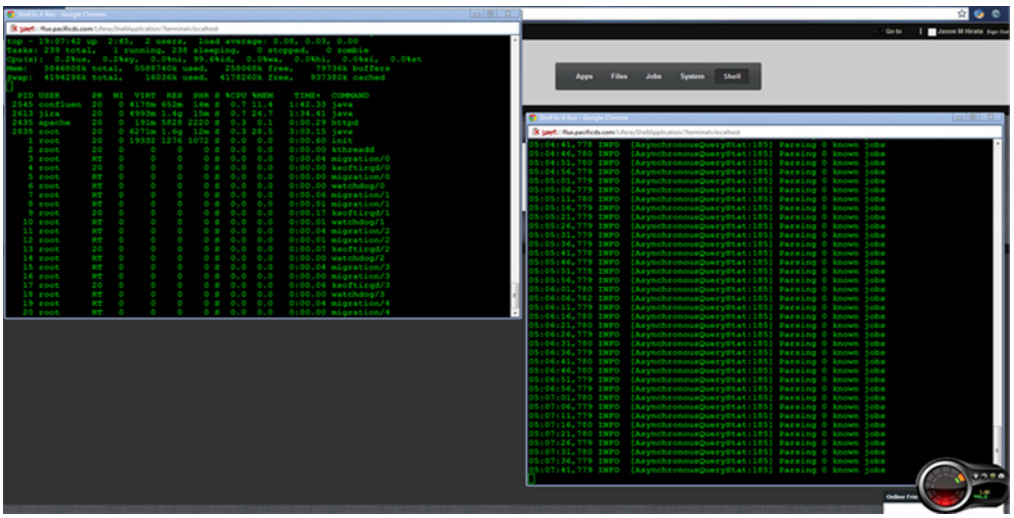

Figure 3. Multiple Web-Shells Available From HPC-Portal

For power users, the shell can provide scripting or development of custom workflows in addition to native HPC-Portal applications. An additional benefit is the ability to access shell resources from non-traditional devices such as an CAC or Yubikey enabled iPad. Delivery to these types of devices was previously not possible since they are not compatible with the Kerberos client kit.

4. Portal Software Development Kit (SDK)

The portal SDK itself is being developed for the DoD. Developers share a common repository and the MHPCC portal team operates as source code change moderators. In this way new versions of the SDK are compatible with prior versions

but may include enhancements and bug fixes supplied by outside teams using the SDK to get their applications incorporated into the portal. The SDK is still in development, but the first version is slated for release in Q4 of 2012. As shown in Figure 4the SDK supplies a Portal FrameWork (backend) for HPC resource access and user interface appUI (frontend) to provide a common look and feel interface to the portal application while AppTop supports legacy and COTS applications. When developers use the SDK, end users benefit from the commonality that the SDK provides in the user interface and resource management including files, job setup/status, quick-look capabilities etc.

Figure 4. Portal Developer Tools

5. Open Source Frameworks

To expedite implementation and facilitate future enhancements a number of open source Portal and content management frameworks were traded including Liferay, WSO2, Drupal and Joomla. Example proof-of-concept test cases were implemented in each as part of the evaluation process to gauge effectiveness of leveraging such frameworks for the HPC portal implementation. Although the proof-of-concept test cases were successfully completed for all frameworks, the selection was narrowed to Liferay because the others had considerably less functionality. Liferay includes built-in content management, enterprise collaboration, and identity management (including OpenID), all of which made it a clear choice relative to the HPCMP Portal goals.

Independent developers using different frameworks successfully implemented a proof-of-concept HPC MATLAB objective to "drag-and-drop" a MATLAB script into a portal window for execution on a remote server and then displaying the output, back into the portal window. After selection of Liferay, the developers were challenged to extend this capability to include the upload and execution of an entire directory of user code and data. This proof-of-concept demonstration represents the use case in which the user-developed package could include an arbitrarily large number of functions (MATLAB .m files) and data files. The developers successfully implemented both the initial and extended proof-of-concept test cases in Liferay. An additional critical objective was demonstration of the implementation in a "zero-footprint," browser-only environment. Achieving this objective provides immediate flexibility to potentially deploy HPC capability to a wide-range of devices, independent of the underlying operating system (see Figure 4).

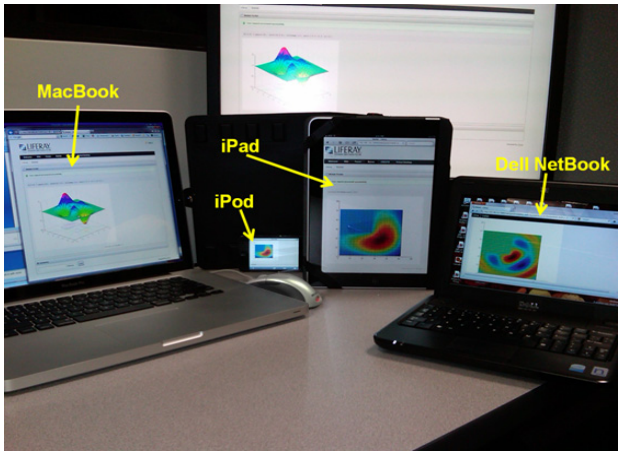

Figure 4. MATLAB Delivered to Traditional and Non-Traditional Devices.

6. Single Sign-on

Single sign-on (SSO) is a combination of identity management and authentication. As documented in the "*Enhancements to High Performance Computing Modernization Program (HPCMP) Authentication and Authorization Services*" memorandum of 5 April 2011, the HPCMP is updating existing access mechanisms, such as PKI (CAC & hToken) to be based on Open Identity technologies (OpenID) and Open Authentication (OAuth). In support of this new initiative, the MHPCC portal development team has successfully integrated Portal with the DoD OpenID concept demonstration server. As an initial risk reduction, the MHPCC portal development team has also implemented a traditional PKI-CAC authentication solution also using Central Authentication Service (CAS). Figure 5 shows the common CAC login page using either OpenID or Legacy PKI-CAC.

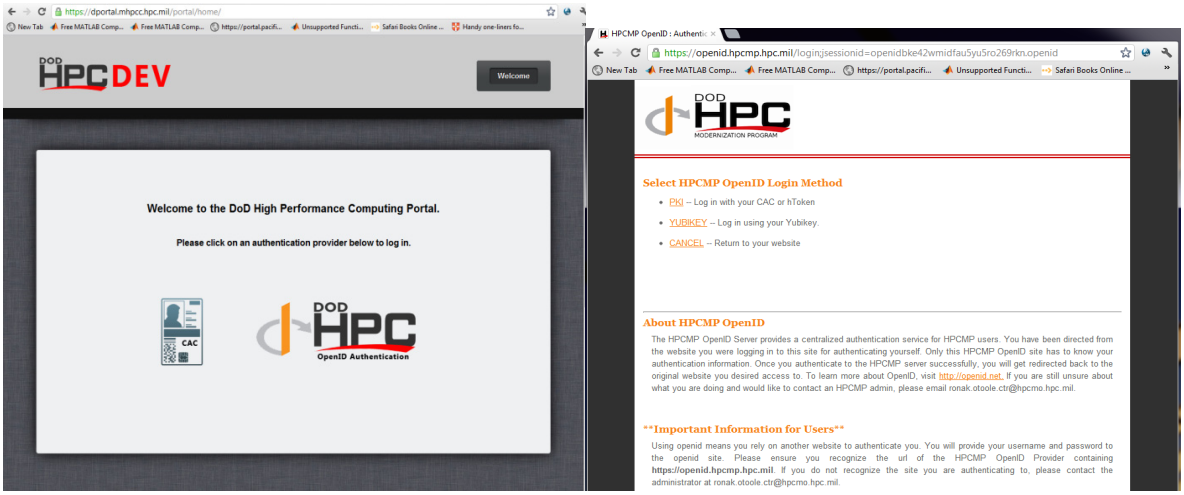

Figure 5. Single Sign On (SSO) From Portal Using OpenID or Legacy PKI-CAC

The portal implementation of SSO also requires a security token in support of DoD high security standards. Currently only the DoD Common Access Card (CAC) token is supported but the Yubikey token (shown in Figure 6) planned for the Q4 2012 release will allow access to a wider DoD contractor community.

Figure 6. Yubikey

The Yubikey was selected by the HPCMO as a lower cost, higher reliability alternative to existing solutions such as hToken and secureID.

7. User Interface Toolkit

The MHPCC team is leveraging key functionality from ERDC's User Interface Toolkit (UIT) for Portal development including providing authentication for remote HPC services as shown in Figure 7. In direct support of portal objectives and for future applications, a new version of UIT is being developed by ERDC that incorporates OpenID and Oauth in place of Kerberos. Incorporation of UIT into the portal framework should allow easy access to resources across DSRCs. As planned the portal will demonstrate seamless resource management between MHPCC and ERDC as a proof of concept for future DSRC wide resource sharing.

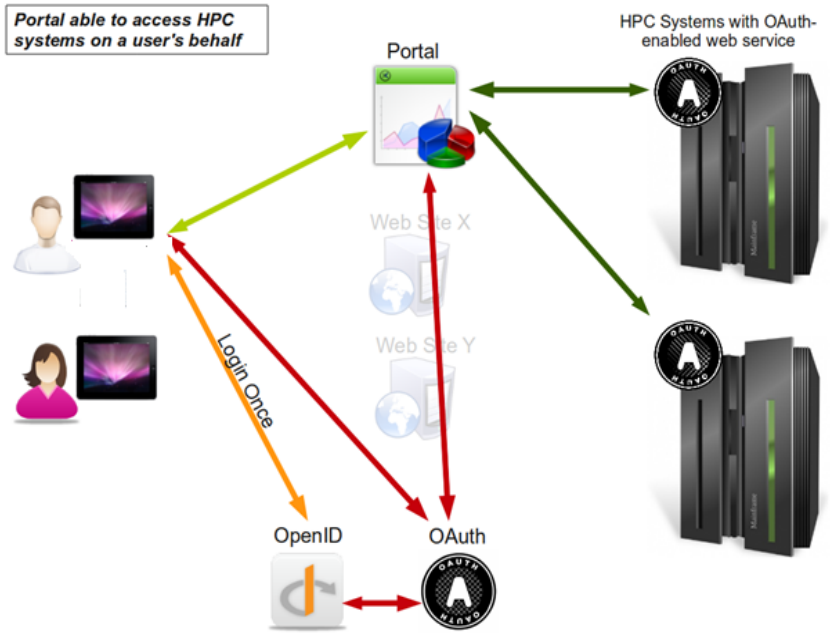

Figure 7. Portal Can Access (Remote) HPC Resources on User's Behalf

8. CREATE-Kestrel Integration

Overview

Kestrel, part of the CREATE Air Vehicles (AV) product line, was selected as the first application to be integrated into the portal. Version 2.1.2 of Kestrel has been installed on the portal development platform. An experienced aerodynamicist has been identified to help support testing and compare, contrast, and evaluate the user experience both within and outside the portal environment. Early feedback from the user is considered a critical part of the CREATE portal integration process. In Q2 2013 CREATE-AV DaVinci and Helios applications will be available through the Portal.

Implementation

Kestrel was well suited for level 1 application delivery since it is based on the MVC architecture and uses xml messages to communicate. The level 1 integration effort of Kestrel was shared by the CREATE and MHPCC portal teams as shown in Figure 8. The CREATE team will be primarily responsible for separating the control logic from the existing Kestrel client (i.e. Kestrel User Interface or KUI) so that it can reside on the server-side. They provide a JavaScript / HTML5 interface that the MHPCC portal development team will integrate and test within the framework.

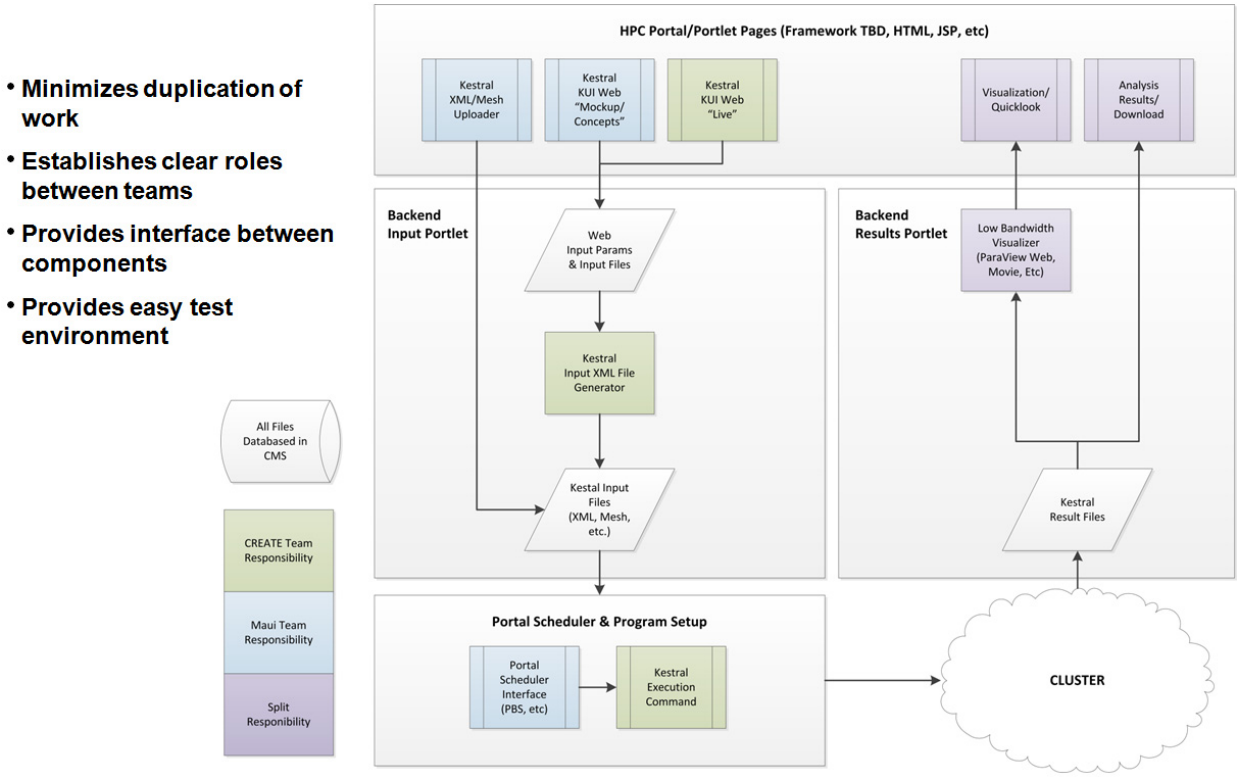

Figure 8. Kestrel Integration into Portal

As shown, the portal provides a scheduler interface via the UIT framework and a visualization capability to support a quick-look user assessment of the Kestrel run. Since the Kestrel architecture uses xml for client / server communication, the development teams can easily exploit this existing design feature for interface testing. The teams compare results using a common set of xml files to ensure consistent behavior between the existing code and the incremental portal development builds. The xml file provides a clean, documented interface between teams facilitating development.

A Liferay portlet that provides a user with a web browser based (zero footprint) user interface to upload required mesh files and resources to the HPC is provided. The Portal Framework is used to upload files to begin the workflow process. The key advantage of Portal Framework for Kestrel is that it provides quick, controlled access to prior runs and data without requiring continual uploads large mesh files. Under user control the full state of the system, including code and data, can be saved for future recall.

As described, UIT is a service oriented framework which supports remote job submission and monitoring across all DSRCs. Integration with the UIT framework facilitates support for DSRC job activities in later phases of the project. The Portal leverages UIT to submit jobs to the HPC cluster via the cluster's scheduler (e.g., PBS).

A key feature of the portal implementation is dynamic status displays and quick look views that help the user determine run quality at a glance. Show in Figure 9 is a dynamic convergence plot. As the run progresses the user is provided a dynamically changing display to provide insight into the run quality before run completion.

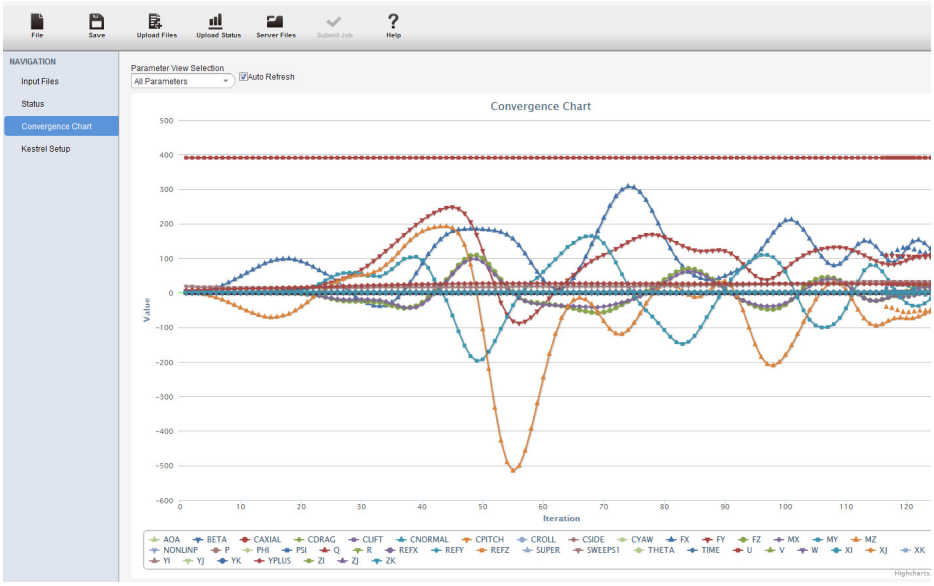

Figure 9. Dynamic Convergence Display

9. MHPCC MATLAB Development

The goal of the MHPCC MATLAB development effort is to provide the user a nearly effortless way to effectively utilize HPC resources using their existing codes. Mechanisms for both naturally parallel and deep parallel MATLAB will be provided that do not require any HPC or HPC programming knowledge.

Naturally Parallel Concept

Figure 10 shows, at a high level, how this concept works for a naturally parallel problem. A specific use case of satellite image processing is given to help explain the utility of a real world example but is easily generalized to any naturally parallel problem set. The user supplies a simple "run function" and the number of iterations N. The run function maps the input parameter n in the range [1, N] to the desired function behavior. Typically n is used to index a particular set of parameters or set of input files and the run function is a trivial MATLAB function typically less than 10 lines long.

- **High-speed short integration frames are collected for multiple satellite / missile pass to freeze atmospheric effects**
- **Research algorithm to remove atmospheric effects written in Matlab requires processing multiple passes for comparison to other techniques (10 to 100 frames are used for each final image in box car fashion)**
- **Portal will provide researcher capability to immediately use HPC for this naturally parallel problem without knowledge parallel programming**

Figure 10. – Naturally Parallel Example and High Level Architecture

Since many naturally parallel problems occur in engineering and science, e.g., monte carlo parameter studies etc., and because such natural problems provide optimal efficiency this portal service should provide immediate benefit to the engineering community. When this option is selected and any iteration is complete, the results can be inspected immediately from within the portal environment. As another example, Figure 12 shows iteration 19 of a blur deconvolution parameter study for indicating sensitivity of deconvolution to blur rate and angle in which n is mapped to the blur rate (pixels/sec) and smear angle relative to the sensor. As shown a quicklook display is automatically provided based on output file type (in this case .png extension). Optionally all MATLAB text and graphics are captured to an html file for each iteration. When this option is selected and any iteration is complete, the results can be inspected immediately from within the portal environment.

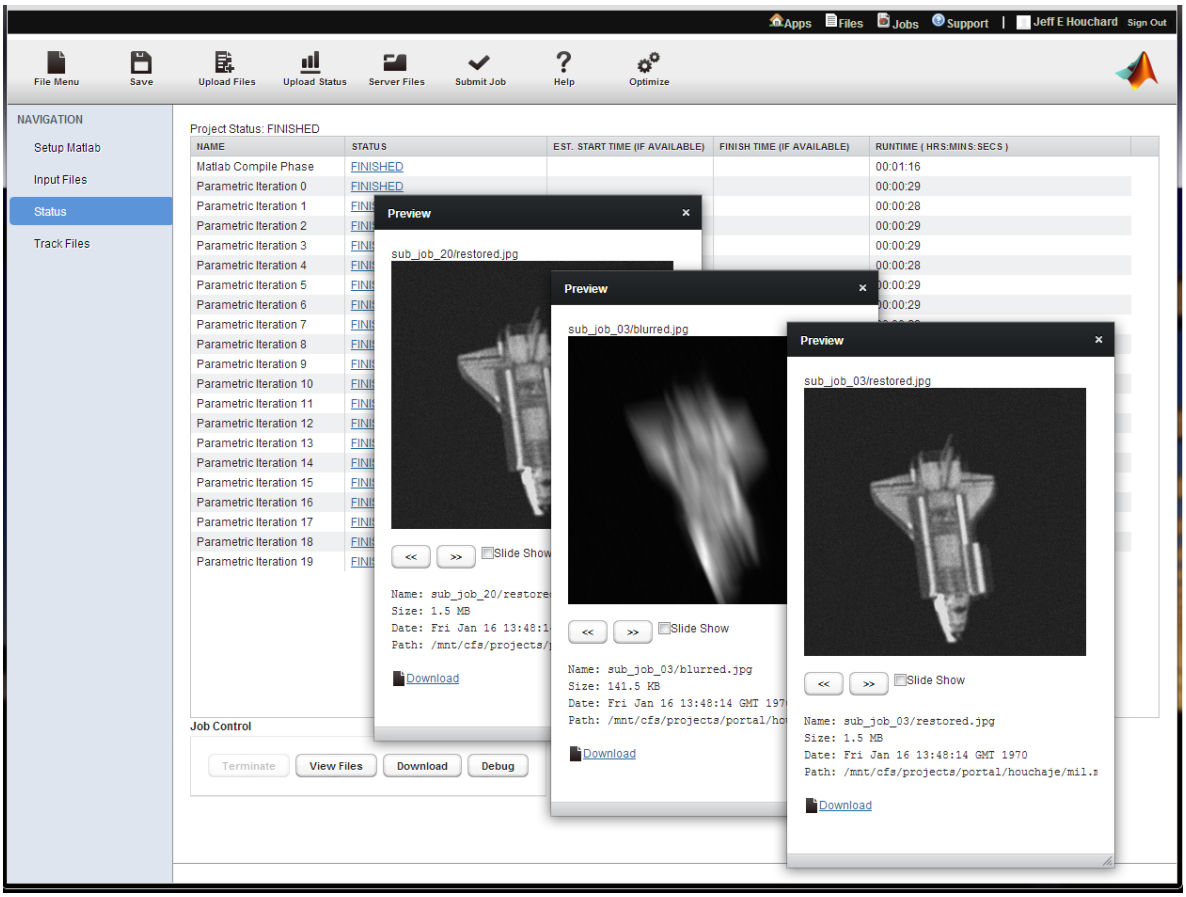

Figure 11. Large Image (2k x 2k) Blur Deconvolution Parametric Study

Figure 13 shows a real world example of running a relatively large MATLAB code (> 500 source files) integrated with the DoD Time-domain Analysis Simulation for Advanced Tracking (TASAT). TASAT itself includes compiled Fortran modules which are used to accurately simulate reflectance and scattering of satellite materials.

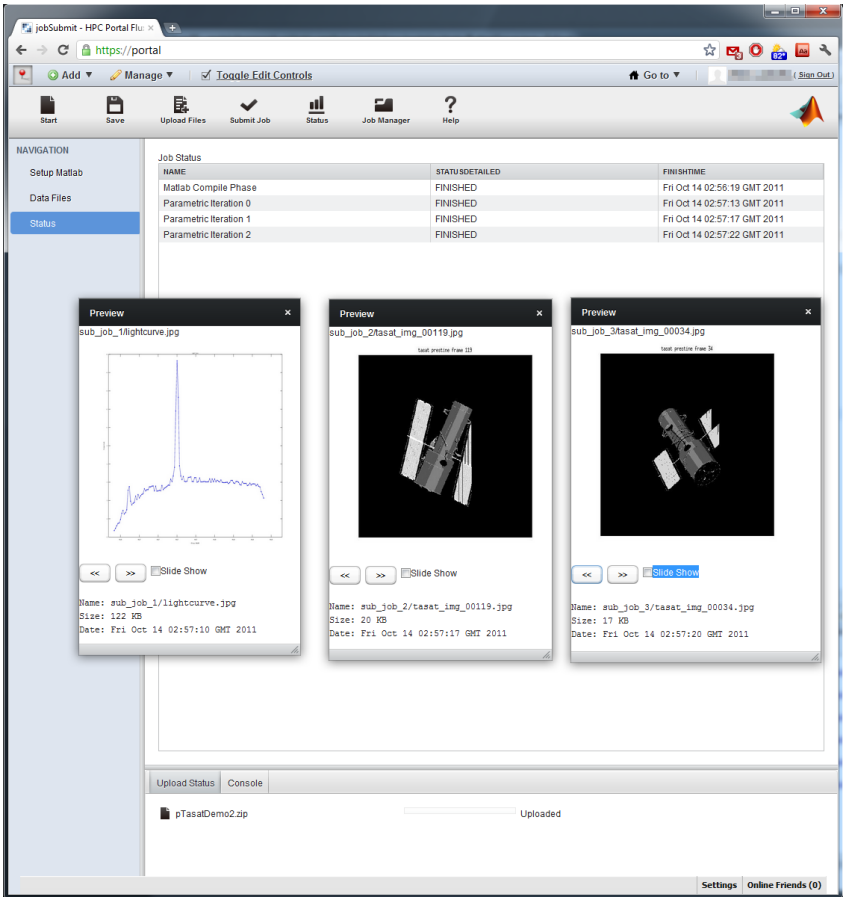

Figure 13. PowerTASAT Implemented Using Naturally Parallel MATLAB

10. Deep Parallel Concept

For problems that are not easily solved with simple natural parallel techniques the MHPCC MATLAB solution will also provide a deep parallel capability. No modification to the user code will be required. The technique exploited is to overload key mathematical operations using the MATLAB Mex interface to run on a cluster of nodes. A simple two dimensional fast fourier transform (fft2 in MATLAB) use case is shown in Figure 14. Analogous to the use case for the naturally parallel example, a specific use case of multi-frame blind deconvoloution (MFBD) image processing is given to help explain the utility of a real world example but any sufficiently large data set requiring 2D FFT processing would benefit from this processing.

For the deep-parallel capability, the near-term plan is to provide the generic interface and implement several mathematical operations as a concept demonstration. Since a large code base of open source software already exists for optimized parallel processing (e.g., LAPAC, BLAS, etc.) a significant portion of the key MATLAB numerical functions could be (easily) included after the concept is demonstrated using the same interface. Additionally, using the Mex interface, such deep parallel libraries, can be used to extend MATLAB beyond its basic functionality to include a wider range of computational capabilities, the domain of HPC, integrating HPC with the most common engineering programming language world-wide.

- **Matlab on current gen i7 processor**
  - **>> tic; fft2(rand(10000)); toc;**
    Elapsed time is 7.320237 seconds.
- **Typical MFBD image processing will require hundreds of forward and inverse 2-D fft's to estimate a single Object and PSF**
- **2D fft (one of many examples) is a naturally parallel**
- **Useful, for example, for near real-time image processing**

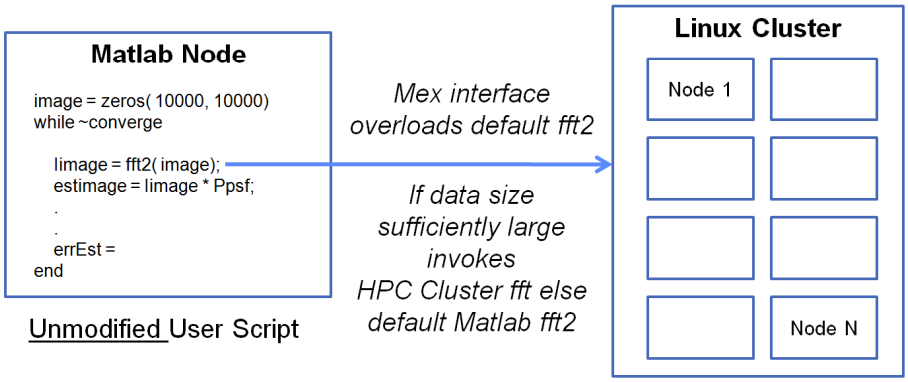

Figure 14. Deep Parallel Example and High Level Architecture

11. MHPCC Time-domain Analysis Simulation for Advanced Tracking (TASAT) Application

Because of local security restrictions imposed on many DoD computers, installing new software or gaining access to DoD provided high performance computing resources (HPC) is not a trivial task. The DoD HPC Portal attempts to solve these issues by proving web applications and services that are accessible from any internet connected computer via a web browser with no plug-ins.

TASAT, a desktop application that has ITAR access restrictions and the need for high performance computing, will benefit from the ease of access and resources that the HPC Portal provides. This article describes these new capabilities that the HPC Portal provides for TASAT users now, and potential new features that enable TASAT as a cloud based web application.

Description of Current TASAT HPC Portal Features

Level 0: MATLAB Interface

The HPC Portal currently provides a MATLAB Natural Parallelization web application. The application allows the user a way to upload custom MATLAB software and data resources, schedule the software to run on an HPC environment, and have the results available for download through the web based portal. The custom MATLAB software should be of the "naturally parallel" (aka embarrassingly or pleasantly parallel) type in which little effort is required to separate the problem into a number of parallel tasks.

A TASAT parallel computing MATLAB library has been developed and is available for download via the HPC Portal. This library consists of MATLAB routines that provide functions to setup TASAT purely in MATLAB. The TASAT runs are distributed to the HPC environment when the user uploads and submits their MATLAB files to the HPC Portal. This is accomplished by using the existing Portal MATLAB Parallelization architecture.

Level 1: TASAT HPC Portal Application

A custom TASAT web application has been developed leveraging the Portal Job Submission Framework (Portal Framework) and Portal Application User Interface Framework (AppUI). In less than 100 lines of code, a fully web based

application was developed that provides TASAT users a way to submit their TASAT run files through the HPC Portal as shown in Figure 15. The web application scans for TASAT run descriptors and distributes each (along with resource files; i.e. nsm files, default parameter files, etc.) to a core on the HPC cluster. The status of each run is monitored by the HPC Portal framework, and when it is finished, made available for post processing to a HPC Portal Virtual Application (VApp) or for download to the user's local workstation.

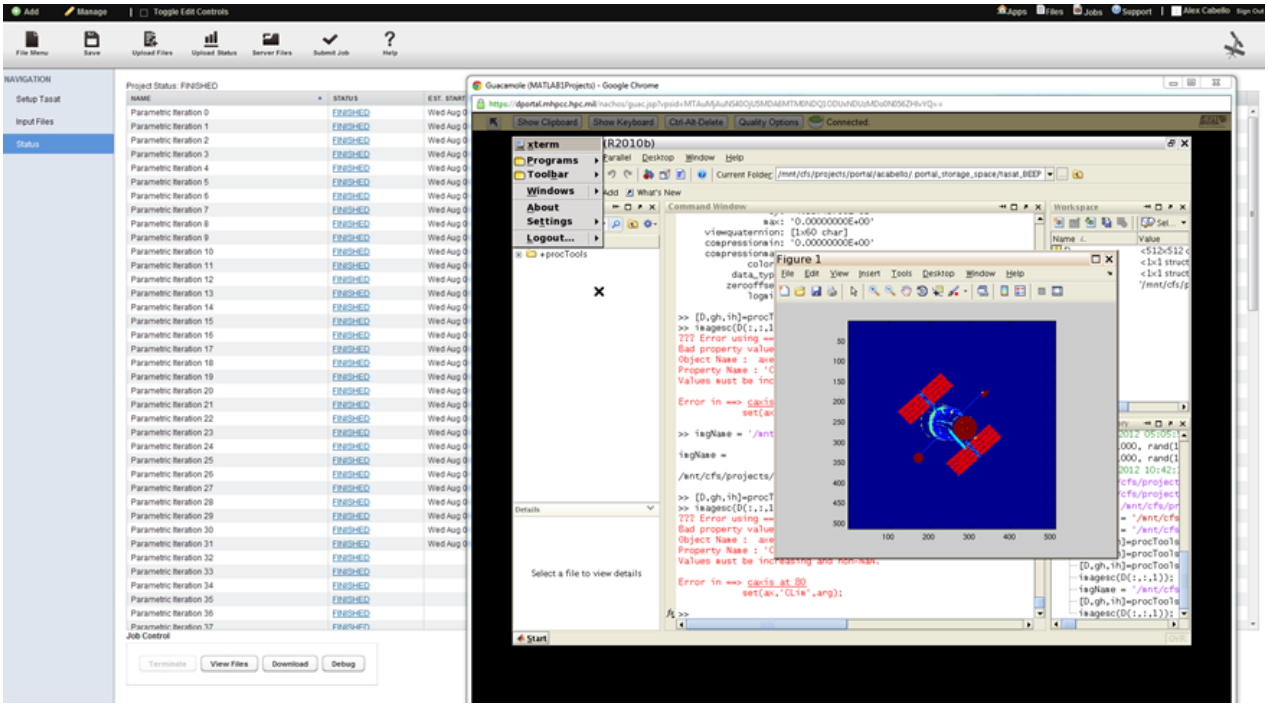

Figure 12. Screenshot of TASAT web application job monitor (back window) and a TASAT output file opened in MATLAB from Portal VApps.

Level 2: Advanced TASAT HPC Portal Application with Cloud Integration

The level 2 concept, which may be available in a future release, would provide an interface to easily submit case studies with existing TASAT parameter files. Certain fields from the TASAT parameter file would be modifiable and allow for multiple input values. Each value would be saved as a new parameter file and distributed to the HPC cluster.

An AJAX based web application to setup a TASAT parameter file could be ported from the TASAT thick client that is shown in Figure 16. This would allow users to build TASAT run descriptors via the cloud similar to Google Documents. Anyone with Internet access could potentially run TASAT. Application access would be handled by the HPC Portal.

Portal would also provide a TASAT resource database and interface. The interface would allow users to choose files from the database. Resource files could include NSM models, BRDF models, predefined sensors, etc.

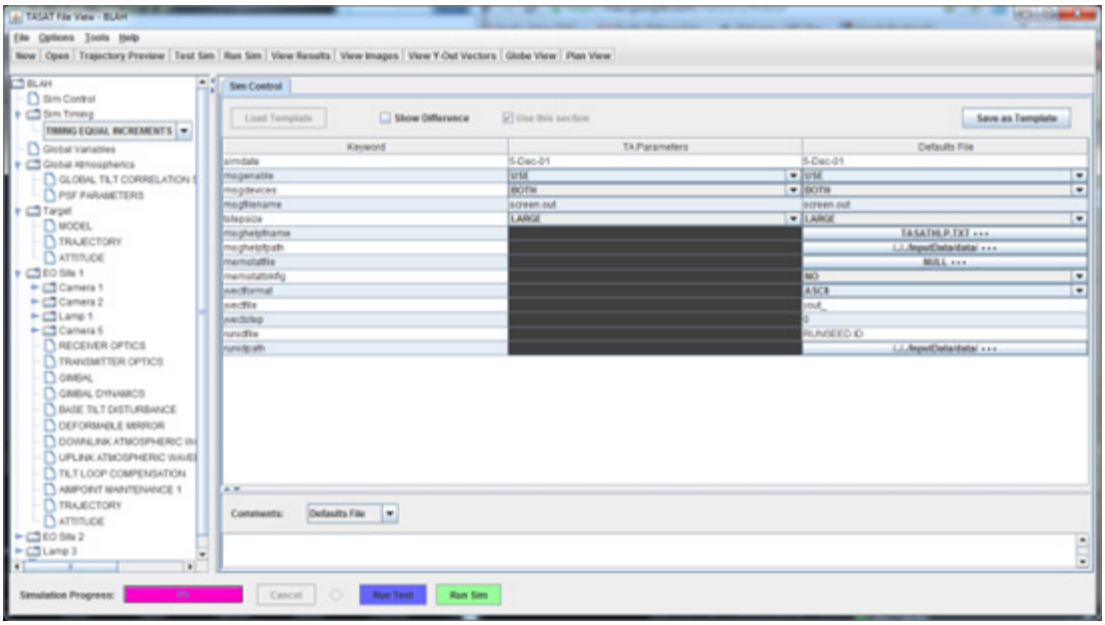

Figure 13. Current Java Based TASAT GUI Provided with TASAT Installation Program. Application could be Ported to a Web Based AJAX Application.

12. Future Capabilities

A number of future capabilities are planned to be implemented in the next release (6 months):

- Additional CREATE applications live including DaVinci, Helios, and Kestrel V3/V4.
- JSpOC Mission System (JMS) HPC development and test platform
- One Semi-Automated Forces (OneSAF) application delivery
- Operational Pilot(s) base on the Kestrel capability available applications
- TASAT Level 2 (optional depending on demand)
- Concept demonstration of how portal could automatically scale using resources across DSRCs (ERDC and MHPCC initially) without requiring user interaction.
- Enhanced Data Management will investigate more sophisticated data sharing with the storage initiatives such as the HEUE Center Wide File System (CWFS) and SRB implementations within the HPCMP.
- Release of enhanced Portal SDK which will include ReSTful services for language independence and will provide an automated stand-alone build capability. A VM test harness will be provided so individual product teams can build the necessary server-side components without standing up a full Portal infrastructure. This will also ensure the UI and workflow framework will integrate easily with the Portal for future builds.

13. Summary

CREATE, distributed MATLAB, and TASAT HPCMP applications are important to the DoD acquisition and engineering communities. The HPC Portal is on track to conclusively demonstrate the high utility a unified distribution mechanism provides. A number of Collaboration services along with the ability to seamlessly access HPC resources within a browser should provide substantial ease-of-use and workflow improvements for these application areas. Pilot users, including CREATE, will be provided access to both the stand-alone and HPC Portal applications which should provide detailed insight into the true utility of Portal based applications to help shape and improve HPC Portal application delivery. For TASAT, discussions with the AF Space Vehicles Directorate were initiated to explore the potential of permanently providing TASAT/PowerTASAT capabilities within the portal environment. The goal of this effort is to provide seamless access to HPC capability to the larger TASAT community. The goal of our longer term effort is to work closely with other Government sponsors and targeted HPC user communities to identify additional applications that should be migrated to the portal environment.