

GA-A27087

INTERACTIVE MONITORING PORTAL FOR FUSION SIMULATIONS

by

**G. ABLA, D.P. SCHISSEL, E.N. KIM,
S.M. FLANAGAN and X. LEE**

AUGUST 2011



DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

INTERACTIVE MONITORING PORTAL FOR FUSION SIMULATIONS

by

G. ABLA, D.P. SCHISSEL, E.N. KIM,
S.M. FLANAGAN and X. LEE

This is a preprint of a paper to be presented at the 8th Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, San Francisco, California, June 20-24, 2011 and to be published in *Special Issue Fusion Engineering and Design*.

Work supported by
the U.S. Department of Energy
under DE-FC02-06ER54861

GENERAL ATOMICS PROJECT 30268
AUGUST 2011

ABSTRACT

The Center for Simulation of RF Wave Interactions with Magnetohydrodynamics (SWIM) Project is a proto-Fusion Simulation Program (FSP) whose goal is to study high-performance fusion plasmas and perform comprehensive simulations that are essential to the development of fusion. SWIM team members are geographically distributed and utilize distributed supercomputers for computational simulations. Due to the highly distributed computational work environment, the SWIM team has the difficulty of monitoring code runs and discovering historical runs. To alleviate this difficulty a web-based monitoring portal has been developed and deployed.

The monitoring portal tracks the progress of simulations and automatically collects metadata in real-time. This capability helps scientists to effectively utilize precious computer resources. Furthermore, the portal provides a web-based interface for post-run analysis, such as visualizing the results, logging the user comments, and rating the simulation quality. The user interface provides rapid discovery capability via multi-field searching and sorting.

The development of the monitoring portal used open source software, such as Python, Django, MySQL, and Apache. It uses MDSplus for data management, Memcached for data caches, and OpenID for single sign-on security.

This paper describes the software architecture, related technologies and deployment experiences of the monitoring portal.

1. INTRODUCTION

The Center for Simulation of Wave Interactions with Magnetohydrodynamics (SWIM) project is dedicated to pursue research on integrated multi-physics simulations [1]. The Integrated Plasma Simulator (IPS) is a framework created by the SWIM team. It provides an integration infrastructure for loosely coupled component-based simulations by facilitating services for code execution coordination, computational resource management, data management, and inter-component communication. Improving resource utilization, implementing application-level fault tolerance, and supporting concurrent “multi-tasking” execution model are the features of IPS framework.

The IPS design utilizes component architecture. The components are the executable physics codes wrapped with necessary scripts to fit in the framework. This is due to the overall project requirement of supporting very high scalability of physics codes while addressing the need of data exchange among the codes that were originally written independently. The IPS defines component boundaries, as well as the data exchange format. A lightweight framework provides component definition by providing a modest set of services to manage configuration, resource allocation, data, and task execution. For data exchange, IPS uses “Plasma State”, a NetCDF file that holds the data that needs to be shared among components in the coupled simulations [2].

Scientists pre-configure an IPS run by specifying a workflow with multiple components that participate in the simulation. The input and output data for each component are also specified in the configuration process. Configuration details are stored in a file, which is used by the IPS framework during execution.

The primary execution environments of IPS are high performance clusters and high-end super computers. IPS execution relies on batch job management systems. A simulation is executed by submitting it as a single batch job specifying the needed computational resources (e.g. number of CPUs and memory size).

Monitoring IPS runs is necessary in order to keep track of overall simulation health. Components running on different nodes can fail due to one of the multiple reasons, such as broken hardware, non-existent input data or non-valid intermediate results. The high degree of scalability and flexibility of the IPS design is not without cost. It is not easy to track the status of ongoing simulations, especially when the simulations run long periods of time. Furthermore, SWIM team members are geographically distributed at multiple locations and execute IPS runs on multiple supercomputers, which makes it difficult to monitor code runs and detect errors. A monitoring feature which tracks IPS steps is needed. With this feature scientists can monitor the status and know when the simulation results are available.

The SWIM monitoring portal and corresponding monitoring component of IPS presented in this paper are designed for two purposes: 1) Real-time monitoring of ongoing simulations; 2) Discovering the steps of completed simulations [3]. The portal collects information about IPS

runs and tracks their progress. The status information is sent from the IPS to the SWIM monitoring portal via HTTP protocol. The received status information and related data is further processed and stored in a relational database. Corresponding physics data is stored in an MDSplus data repository [4]. The web-based portal interface displays status information, visualizes data, and supports interactive post-run analysis.

The rest of this paper describes the SWIM monitoring portal software architecture (Sec. 2) and supporting technologies for its development (Sec. 3). It also presents deployment experiences (Sec. 4) followed by conclusion and possible future improvement plans (Sec. 5).

2. SOFTWARE ARCHITECTURE

2.1 OVERALL ARCHITECTURE

The SWIM monitoring portal software design utilizes Model View Controller (MVC) architecture [5]. Figure 1 shows the SWIM monitoring portal architecture.

1. **Model.** The Model component is responsible to access the data. It provides the View component with the needed information for web display and visualization. The status data sent by the IPS and data generated by post-run user activities are stored in a MySQL relational database. The physics values generated by simulation are stored in MDSplus. The Model interfaces both with the MySQL database and the MDSplus.
2. **View.** The View component is responsible for creating webpages and generating visualization plots for both desktop and mobile devices. It creates web page based on pre-prepared HTML templates.
3. **Controller.** The Controller component is the processing engine of the system. Messages sent by the IPS and requests created by user activities are intercepted by the Controller component. It parses the messages and passes them to the Model component.

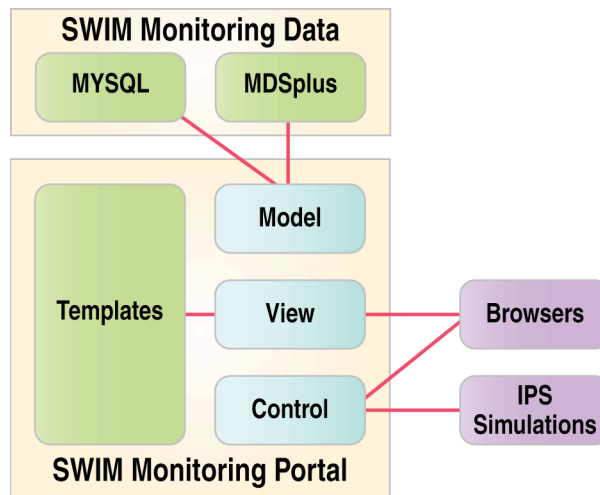


Fig. 1. SWIM Monitoring Portal Architecture.

The Controller component mainly receives three types of messages and requests. First is the “RunID generation request” sent by the IPS monitoring component. When this request is received, the Controller responds with a unique identification number, RunID. RunID is assigned to each new simulation regardless of the location of IPS runs. On the monitoring portal, RunID is used to associate, store and identify all information and data that belong to a single simulation. Second is the simulation status information sent by the IPS framework. The IPS attaches RunID to all messages belonging to the same simulation whenever it sends simulation status message to the monitoring portal. Communication between the IPS and the SWIM web portal utilizes HTTP

POST. The third type of message is user activity generated requests. These are HTTP GET/POST messages created by user activities, such as viewing, searching, commenting, purging, and rating.

2.2 MONITORING CLIENT

The monitoring client is part of the IPS framework. It is provided by a special IPS component, called the portal bridge. From the framework's standpoint, the portal bridge is another component. However, this component is permanently attached to the framework, and therefore always available as part of the IPS framework. With this design, the monitoring client works seamlessly with any newly integrated physics component of the IPS.

At the beginning of each IPS run, the portal bridge sends a "RunID generation request" to the SWIM web portal and gets a unique RunID. Then, the portal bridge subscribes to status and progress events published by the framework as part of the invocation of various service methods. It acts as a client, and transmits simulation status messages to SWIM web portal server through a simple HTTP POST requests.

2.3 MESSAGE PROCESSING AND AUTOMATIC POST-RUN ANALYSIS

The Control component of the web portal processes each incoming IPS status message. Based on the RunID number, a new message is classified either as the beginning of a new simulation or part of an older simulation and stored into the appropriate database table.

If a message announces the completion of an IPS run, the Control component also triggers a series of automated post-run activities, such as downloading of an IPS monitor file, which contains the calculated values by the completed simulation.

2.4 INTERACTIVE POST-RUN ANALYSIS

The SWIM Monitoring portal provides users with a series of web pages to display the status of simulations. The main page displays a snapshot of the latest status of all simulations. Status information can be filtered or sorted by multiple keys, including RunID, user name, physics code and comment. The details page, which is linked from the main page via RunID, provides detailed historical information of each simulation. It also provides information about physics codes and data participated in the simulation.

Both the main and simulation detail pages provide search capabilities. Simple search can be used to search by keywords in multiple fields. The advanced search gives more options for configuring the search query with combinations of multiple keywords and parameter ranges

The SWIM monitoring portal facilitates interactive post-run analysis. Users can enter comments on a specific simulation and view comments entered by others. Users also can rate, purge and un-purge simulation runs. The metadata created during these interactive activities are permanently stored in the relational database and available to search and view at a later date by the entire team.

2.5 DATA MANAGEMENT

Both status data sent by the IPS and metadata generated by interactive post-run analysis are stored in a relational database.

There are multiple database tables created to store a variety of data in the SWIM web portal. The “simulation_summary” table stores the latest up-to-date information on each simulation and is designed so that each simulation has only one table. The “simulation_details” table stores information about all simulation steps for both ongoing and historical simulations. There are other tables for storing post-run comments, rating information, and purging/unpurging activities of users.

The SWIM monitor file stores values of physics variables computed by simulations, and its content is always updated with the latest values of IPS physics variables during simulations. The monitor file is in the NetCDF file format. Although the NetCDF monitor file structure is efficient in serving the needs of the IPS running on clusters and super computers, it is not convenient for quickly generating visualization plots for separate variables in a web-based environment. Therefore, during and after the simulation the corresponding monitor file is extracted by the monitoring portal and its physics values are stored in the MDSplus repository via a cron job that runs a Python code. The code uses pupynere, a lightweight NetCDF reader/writer, to read in the monitor file [6]. The data from the NetCDF variables is then matched to the corresponding dimension values in order to create 1 or 2D "signals" to be stored in SWIM "tree" in MDSplus using the MDSplus Python API. The simulation RunID uniquely identifies the data stored in MDSplus.

2.6 DATA VISUALIZATION

Along with simulation status information, data plots provide visual clues about ongoing simulations via the SWIM web portal. During simulations, static data plots are generated automatically in thumbnail format. The detailed interactive (pan, zoom, data value readout) visualization plots are linked via thumbnail plot images. Both static and interactive visualizations are supported on the backend by the MDSplus repository. Since the SWIM data is automatically loaded into MDSplus as the simulation progresses, the visualization plots are not only available after the completion of a simulation, but also available for each step of the simulation so that the scientist can follow the simulation progress graphically.

Details of the data visualization are described in a separate paper titled “Web-Based (HTML5) Interactive Graphics for Fusion Research and Collaboration” submitted to this meeting [7].

3. TECHNOLOGIES USED FOR DEVELOPMENT

All of the software and libraries used to develop the SWIM monitoring portal are open source. The software stack is made up of HTML and Java script at the front end and Python at the backend. MySQL as well as MDSplus are used for storing data. The Django web framework provides the glue for all software components described in Sec. 2 [8].

Django is an open source web framework, written in Python. It emphasizes reusability and pluggability, and makes easy to create complex, data driven web sites. Among many features, its core provides an object-relational mapper, which provides Python class interface to relational database (MySQL in our case). The built-in capabilities of the object-relational mapper provide easy and safe access to database content. Another core capability of Django is the powerful web template system, which easily allowed the SWIM web portal to provide multiple HTML templates for different pages. It also facilitates creating web pages for different devices (desktop and mobile) with same data content.

Another software used for the SWIM monitoring portal development is Memcached, a distributed memory object caching system [9]. Memcached is a key-value store for small chunks of arbitrary data (strings, objects) from results of database calls. Due to the dynamic nature of the SWIM web portal, each time a user requests a page, the web server makes a query to the MySQL database, and executes some type of computation (e.g. sorting), to create the requested web page. This is a computationally expensive overhead, especially when a page is created for simulation details, which may include thousands of steps. While the details page changes during a simulation, its content becomes static after the simulation. Thus, the results can be stored in the memory just after the simulation and there is no need to re-query the relational database for each view request afterwards. Although, Django supports multiple caching systems, the SWIM web portal used Memcached since it is robust and developers already have experience with its implementation. Figure 2 shows the Memcached utilization on the monitoring portal development.

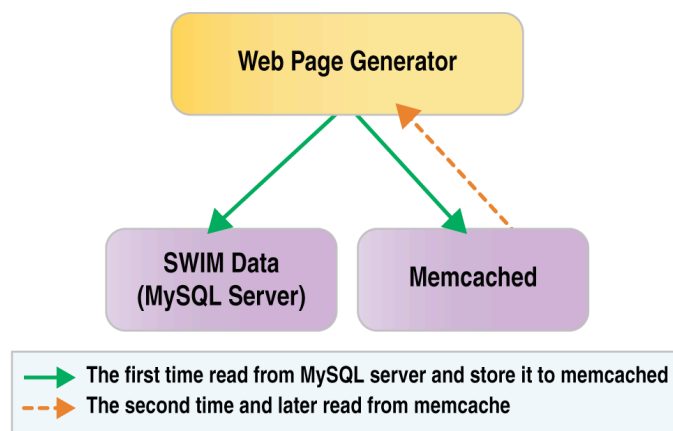


Fig. 2. Using of Memcached on the monitoring portal development.

The SWIM web portal relies on OpenID technology for user authentications [10]. A number of the SWIM web portal. basic activities, such as monitoring of ongoing and historical simulations, are allowed without requesting user identities. However, certain actions (comments, ratings, purging) require users to authenticate. Originally, the SWIM web portal maintained its own user database, in which it keeps user names and passwords. However, the SWIM project also has dedicated website, <http://cswim.org/>, which is used for project coordination and as a source code repository. The project website relies on a user account database at Oakridge National Laboratory (ORNL). Having multiple accounts for the same user in the same project created some confusion. Therefore, the SWIM monitoring portal utilizes OpenID technology to consolidate user accounts.

Unlike many other OpenID consumers, the SWIM web portal recognizes only one OpenID provider — the OpenID server at ORNL. Now, when a user requests to login on the SWIM monitoring portal, they will be briefly forwarded to the ORNL OpenID provider server asking to provide login information. If the OpenID provider confirms the username and password, they will be forwarded back to SWIM web portal with a successful login status. Figure 3 shows the OpenID implementation of the SWIM monitoring portal.

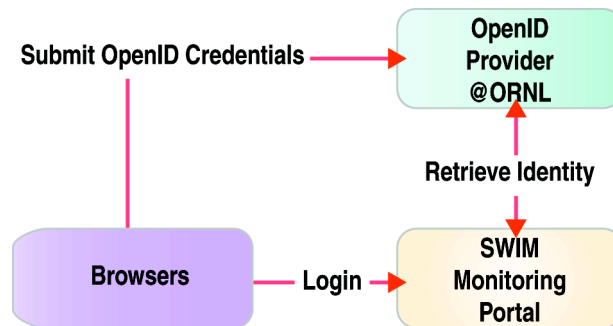


Fig. 3. The OpenID implementation of SWIM Monitoring Portal.

4. DEPLOYMENT EXPERIENCES

The SWIM monitoring web portal has been deployed since early 2010. New capabilities continuously have been added since that time. In the early release, the monitoring portal only provided text-based status information. The HTML5-based visualization capability was added recently. Due to the increased usage of mobile devices, the mobile version of SWIM Monitoring Portal website also was released. This version is optimized for mobile devices and it makes monitoring and navigating SWIM simulation results with mobile devices much easier. Up to this point, there have been more than 8000 IPS simulations recorded on the monitoring portal. These simulations were carried out by 15 scientists at high-end super computers and clusters located at multiple international fusion research institutions, including ITER. Figures 4 and 5 show both the main portal page and detail pages of the SWIM portal.

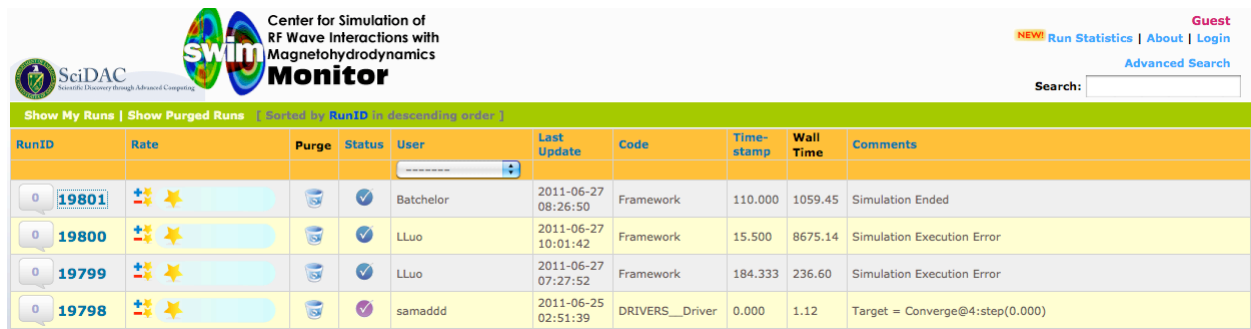


Fig. 4 A screen capture of the main page of SWIM Monitoring Portal.

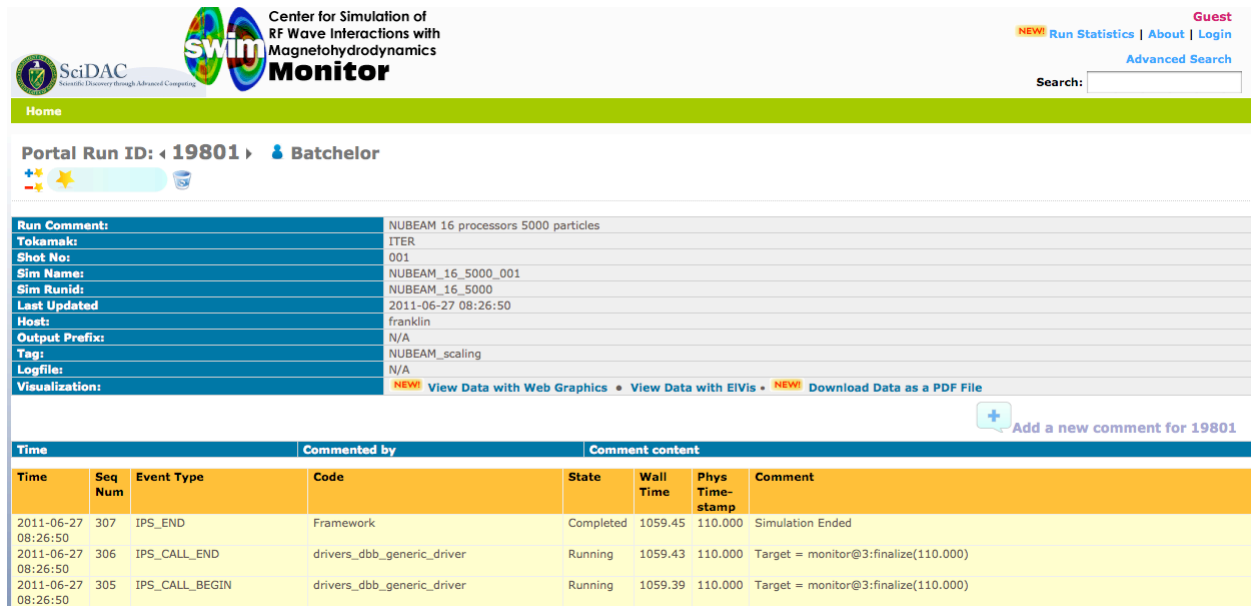


Fig. 5 A screen capture of a details page of SWIM Monitoring Portal.

Currently, there are two Linux (Cent OS) servers being used by the monitoring portal. Both of them are located at General Atomics, San Diego utilizing the same ESnet connection as the DIII-D National Fusion Facility. The first computer is the web server and almost all the processing components of the monitoring portal are running on this server. Apache webserver software is being served on multiple network ports, due to the varying network security policies at institutions where the IPS simulation servers are located. This server also serves as the memory cache server and Memcached software utilizes 0.5 GB RAM for the cache storage. The second computer is responsible for storing data. It serves as both MySQL server and MDSplus server.

5. CONCLUSIONS AND FUTURE WORK

The SWIM monitoring web portal has been well received by the SWIM team. A year and half of production usage has proven that the portal's capabilities, such as real-time monitoring, web-based visualization, as well as support for post-simulation analysis increased work efficiency, and are important for conducting the project's science.

However, it should be noted that the fusion simulation research is evolving. Therefore, it is natural that the SWIM monitoring portal continues to undergo development as new needs or opportunities arise. The possible future work can include examination of increased robustness for message passing and further improvements to conducting science via the web portal.

Currently, the messaging mechanism between the IPS and the SWIM monitoring portal relies on HTTP protocol. This works well when they can communicate without a long delay, which was not always the case due to network problems such as network congestion. The network can be slow when an IPS simulation and the SWIM monitoring portal are located on different continents. If there is a network outage on one side, even temporarily, the status messages can get lost. Utilizing a reliable, highly available messaging method can guarantee the eventual safe delivery of monitoring messages. We are considering an open source enterprise messaging system RabbitMQ, for our messaging mechanism between IPS and the portal [11].

Another improvement will be a web-based tool for online assembling, storing, editing, and sharing of SWIM simulation configurations in the monitoring portal environment. This will not only allow the portal to capture the full provenance of SWIM simulations, but also increase collaborations among scientists.

REFERENCES

- [1] SWIM Project website: <https://cswim.org>
- [2] NetCDF website: <http://www.unidata.ucar.edu/software/netcdf/>
- [3] SWIM Web Portal: <http://swim.gat.com:8080/>
- [4] T. Fredian, J. Stillerman, “MDSplus Remote Collaboration Support-Internet and World Wide Web,” *Fusion Eng. Design* 43 (1999) 327.
- [5] Trygve M.H. Reenskaug MVC XEROX PARC 1978-79: <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>
- [6] Pupynere – NetCDF reader/writer: <http://pypi.python.org/pypi/pupynere/>
- [7] E. N. Kim *et al.*, “Web-Based (HTML5) Interactive Graphics for Fusion Research & Collaboration”
- [8] Django Project website: <https://www.djangoproject.com/>
- [9] Memcached Project website: <http://memcached.org/>
- [10] OpenID web site: <http://openid.net/>
- [11] RabbitMQ web site: <http://www.rabbitmq.com/>

ACKNOWLEDGMENT

Work supported by the U.S. Department of Energy under DE-FC02-06ER54861.