Fusion Engineering and Design 87 (2012) 2052–2056

Contents lists available at SciVerse ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

Interactive monitoring portal for fusion simulations

G. Abla*, D.P. Schissel, E.N. Kim, S.M. Flanagan, X. Lee

General Atomics, P.O. Box 85608, San Diego, CA 92186-5608, USA

ARTICLE INFO

Article history:
Available online 17 May 2012

Keywords:
Portal
Fusion simulation
Monitoring
Web
OpenID
Interactive

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

* Corresponding author. Tel.: +1 858 455 3103; fax: +1 858 455 3586.
E-mail address: abla@fusion.gat.com (G. Abla).

0920-3796/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.fusengdes.2012.04.007

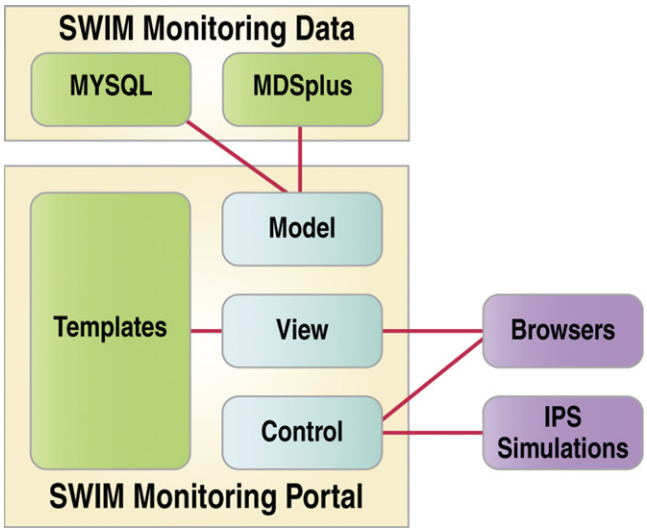

Fig. 1. SWIM monitoring portal architecture.

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 (Section 2) and supporting technologies for its development (Section 3). It also presents deployment experiences (Section 4) followed by conclusion and possible future improvement plans (Section 5).

2. Software architecture

2.1. Overall architecture

The SWIM monitoring portal software design utilizes Model View Controller (MVC) architecture [5]. Fig. 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 web pages 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.

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, which has a variety of functional components [6]. 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. All authenticated users are also able to rate, purge and un-purge simulation runs. The metadata created during interactive activities are permanently stored in the database and available to search and view at a later date by the