Procedia Computer Science

Volume 66, 2015, Pages 458–467

YSC 2015. 4th International Young Scientists Conference on
Computational Science

An Approach to Problem-Oriented Interfaces for Applications in Distributed Computing Systems

Aleksei A. Sorokin[1]*, Sergey P. Korolev[1], Sergey I. Malkovsky[1]
, Alexei G. Tarasov[1], and Konstantin V. Mikhailov[1]

Computing Center of the Far Eastern Branch, Russian Academy of Sciences, Khabarovsk, Russia
alsor@febras.net

Abstract

The paper describes the implementation of an efficient user interface for applied software in distributed computing systems. As part of this work, a software platform has been developed for the formation of problem-oriented interfaces for application software packages and components of distributed computing systems. The main purpose of the system is to simplify the input of technical settings, launch specific jobs if required and monitor all intermediate processes. Thus, it provides an intuitive user interface for working directly with an application system, and does not require knowledge of low-level operations. Using web services and modern information technology, the developed platform makes it possible to create a set of simple and convenient interfaces for researchers who use high-performance computer systems in their work. As an example, we describe the adaptation of the FHI98md code for a Grid system.

Keywords: distributed computing systems, problem-oriented interface, database, Representational State Transfer, JavaScript Object Notation, application programming interface, web services, Grid, FHI98md.

1 Introduction

A large number of applied software and information technologies capable of having a significant impact on the quality and speed of ongoing researches are nowadays eliminated or used at minimal capacity. One reason is that the complexity of interaction with computer systems increases due to higher levels of performance. The other reason is because the specificity of HPC does not allow professionals in subject fields far from computing technologies to quickly resolve any software and hardware issues.

One solution to these issues is to hide all technical details of the computing tasks from the user and, instead, provide an intuitive user interface that does not require low-level operations.

*Corresponding author. Tel.: +7-421-270-3913 fax: +7-421-222-7267.

Selection and peer-review under responsibility of the Scientific Programme Committee of YSC 2015

doi:10.1016/j.procs.2015.11.052

The aim of this work is to develop a software platform that can generate a user interface for an application package, thus solving a whole spectrum of problems related to computational operations in HPC.

2 Statement of the problem

Currently, considerable attention is focused on the development of HPC software systems and the improvement of their operations. It is worth mentioning here that international consortia are designing tools for Grid implementations [7] (e.g., Globus [6]], parallel computing scheduling package Condor-G [9] and X-COM system [11]). Many other solutions, based on different hardware platforms and effectively integrating distributed computing nodes into a single managed system with unified access to resources, also become available. Nevertheless, only a limited number of studies are aimed at simplifying the end-user interaction with computing systems [12]. For example, the Russian Grid NNN project [5] uses a single information portal to implement the application programming interface (API), which allows the addition of new Grid application suites. In spite of the advantages of this project, its interaction with the end user has the following disadvantages:

1. The API is quite restrictive. Because of the specificity of the portal, one cannot create user interfaces that go beyond the provided API. This is well illustrated by the following examples:

    - The API is located on the portal server, so users require (unwanted) access rights to this server to modify the library and load extension modules (plugins);

    - The API uses an existing implementation of a JavaScript library, which makes it difficult to create automated problem-oriented interfaces (POI) for computing jobs outside this environment with AJAX support.

2. The API combines two separate sets of functions that implement both the creation of configuration files and submission of computing jobs for execution. This complicates the end user work dramatically, because a wide range of technical operations must be performed (data entry, setting job related parameters, downloading the generated files to their workstation, creating and submitting the job alongside with a set of the corresponding configuration files to the server). This can lead to errors and resubmitting of the job.

3. A significant part of the API functions is handed to the user interface, potentially compromising the system operations by substituting client side plugins.

A similar approach is used in existing solutions of manufacturers of computing systems and management software for them. These include Moab HPC Suite Application Portal Edition [1] from Adaptive Computing, Altair Compute Manager [4] from Altair Engineering and Platform Application Center [8] from IBM. The applicability of above solutions to heterogeneous computing resources management is hampered by their binding to specific computer systems or to specific control software. In addition, the opportunity of modifying such information systems for performance of specific tasks is significantly limited by provided APIs and lack of access to the source code.

The most common approach to solving the above listed problems of software adaptation is to provide a user interface for a specific application system. For example, the "ABINIT