Project 2 A Simple Java-based Framework for Grid Computing Fall 2004
Based on http://www.javaworld.com/javaworld/jw-04-2003/jw-0425-grid.html

1. Introduction

Grid computing is a natural evolution of the information infrastructure successfully realized using the Internet. It provides an infrastructure for the flow of services by exploiting the vast pool of resources networked by the Internet. Early beginning of the grid computing can be observed in the SETI@home project. Currently many toolkits such as Globus Toolkit 3.0x and Condor 6.X.X are available to implement the grid framework. However these frameworks are production-quality and are quite complex for us to understand, deploy and take apart to study and experiment with the code. (For example, In CSE421 Operating Systems course we can study, understand and extend the code whereas with a complete Unix BSD or Solaris you may not be to do as easily.) So we have decided to let you build a minimal grid framework based on the article "A do-it-yourself framework for grid computing" by Anthony Karre in Java World. Our focus in this project will be on the client-side of the grid computing (though we will run a server). The framework given in paper satisfies the following grid requirements:

1. Machine independence (using Java, Apache Tomcat servlet container and Apache Axis SOAP implementation)
2. Secure and scalable infrastructure achieved through the use of SOAP-based Web services for client-server communication.
3. Task abstraction achieved through the use of jar files, and Java classloader. A classloader is capable of identifying and executing an arbitrary Java class.

In this project we will use *SOAP with Attachments API for Java* (SAAJ) instead of the Apache Axis.

2. Purpose of the Project

Implement a simple framework for the grid clients to retrieve, load and execute a specified task given the *jar* files of the job submitted to the grid. The client will also return at a later time a representative result of the execution to the server.

3. Technology Requirements

Java 2 platform standard edition (J2SE 1.3.1 or later), Apache Tomcat servlet container, SAAJ, and JAX-RPC based webservices.

4. Assignment

4.1 Architectural Model

The block diagram of the overall system is given Figure 1. Computing task is submitted to the grid server as a JAR file with the manifest indicating the task thread. The client side contains a main class and a custom classloader. The client side operations are as follows:

1. The main class instantiates the custom classloader.
2. The custom classloader uses a SOAP service to fetch the JAR file containing the classes for the task thread. The JAR file stored locally upon retrieval.
3. Then the main class in the client uses the classloader to load and instantiate the primary compute thread identified by the manifest in the JAR file.
4. The compute thread is executed by starting the thread.
5. The result of the computation is returned to the server using other methods such as Webservices. (Observe that server and client are loosely coupled.)

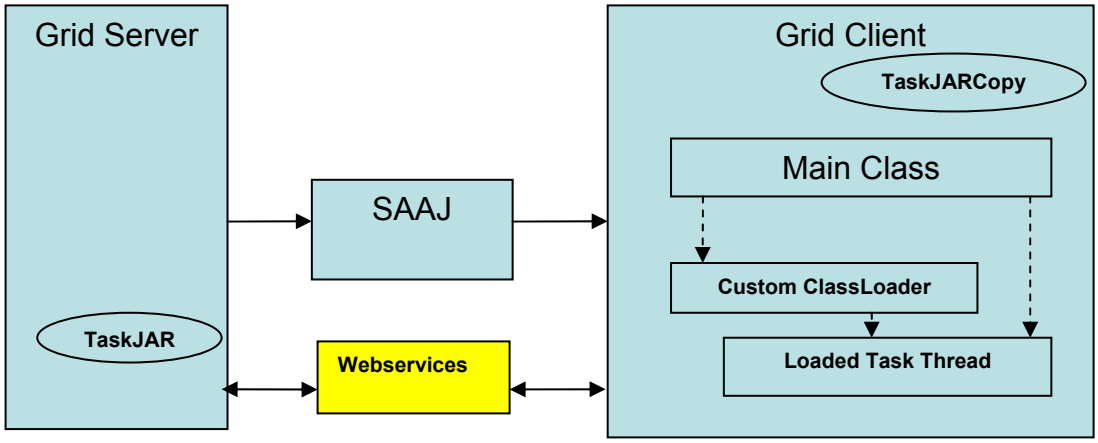

Figure 1: Components of the Grid Server and Grid Client of the Project2

4.2 Implementation Details

1. You will design and implement a very simple server that will allow users to submit computing tasks in the form of JAR files.
2. Initially a trivial compute thread similar to the one discussed in the paper referenced will be coded and packaged into the JAR file and stored on the server. It will have a manifest file that identifies the task thread.
3. A SOAP service will be used to transfer the task JAR files to the client on request. Use the SAAJ implementation of SOAP.
4. On the client side you will build a custom classloader to retrieve the JAR file using the SOAP service. It will also load the class containing the task thread.
5. Design and implement the main client application for instantiating the custom classloader, then using it to load task thread.
6. Design and implement a means for returning the results back to the server.
7. Use the infrastructure build to create and run any non-trivial application.

5. Report and Submission

See Project 1 for the Report that you need to prepare and for the submission details.

Due Date: November 10, 2004 by midnight. No extension will be given.