A Remote Sensing Application Workflow and Its Implementation in Remote Sensing Service Grid Node

Ying Luo^{1,4}, Yong Xue^{1,2,*}, Chaolin Wu^{1,4}, Yincui Hu¹, Jianping Guo^{1,4},
Wei Wan^{1,4}, Lei Zheng^{1,4}, Guoyin Cai¹, Shaobo Zhong¹, and Zhengfang Wang³

¹ State Key Laboratory of Remote Sensing Science, Jointly Sponsored by the Institute of Remote Sensing Applications of Chinese Academy of Sciences and Beijing Normal University, Institute of Remote Sensing Applications, Chinese Academy of Sciences, P.O. Box 9718, Beijing 100101, China

² Department of Computing, London Metropolitan University, 166-220 Holloway Road, London N7 8DB, UK

³ China PUTIAN Institute of Technology, Shangdi Road, Beijing 100085, China

⁴ Graduate School of the Chinese Academy of Sciences, Beijing, China
{jennyjordan@hotmail.com, y.xue@londonmet.ac.uk}

Abstract. In this article we describe a remote sensing application workflow in building a Remote Sensing Information Analysis and Service Grid Node at Institute of Remote Sensing Applications based on the Condor platform. The goal of the Node is to make good use of physically distributed resources in the field of remote sensing science such as data, models and algorithms, and computing resource left unused on Internet. Implementing it we use workflow technology to manage the node, control resources, and make traditional algorithms as a Grid service. We use web service technology to communicate with Spatial Information Grid (SIG) and other Grid systems. We use JSP technology to provide an independent portal. Finally, the current status of this ongoing work is described.

1 Introduction

Grid has been proposed as the next generation computing platform for solving large-scale problems in science, engineering, and commerce [3][4]. There are many famous Grid projects today: DataGrid, Access Grid, SpaceGRID, European Data Grid, Grid Physics Network (GriPhyN), Earth System Grid (ESG), Information Power Grid, TeraGrid, U.K. National Grid, etc. Of particular interest are SpaceGRID and ESG, which focus on the integration of spatial information science and Grid.

The SpaceGRID project aims to assess how GRID technology can serve requirements across a large variety of space disciplines, such as space science, Earth observation, space weather and spacecraft engineering, sketch the design of an ESA-wide Grid infrastructure, foster collaboration and enable shared efforts across space applications. It will analyse the highly complicated technical aspects of managing, accessing, exploiting and distributing large amounts of data, and set up test projects to see how well the Grid performs at carrying out specific tasks in Earth observation,

* Corresponding author.

space weather, space science and spacecraft engineering. The Earth System Grid (ESG) is funded by the US Department of Energy (DOE). ESG integrates supercomputers with large-scale data and analysis servers located at numerous national labs and research centers to create a powerful environment for next generation climate research. This portal is the primary point of entry into the ESG.

The Condor Project has performed research in distributed high-throughput computing for the past 18 years, and maintains the Condor High Throughput Computing resource and job management software originally designed to harness idle CPU cycles on heterogeneous pool of computers [2]. In essence a workload management system for compute intensive jobs, it provides means for users to submit jobs to a local scheduler and manage the remote execution of these jobs on suitably selected resources in a pool. Condor differs from traditional batch scheduling systems in that it does not require the underlying resources to be dedicated: Condor will match jobs (*matchmaking*) to suited machines in a pool according to job requirements and community, resource owner and workload distribution policies and may vacate or migrate jobs when a machine is required. Boasting features such as check-pointing (state of a remote job is regularly saved on the client machine), file transfer and I/O redirection (i.e. remote system calls performed by the job can be captured and performed on the client machine, hence ensuring that there is no need for a shared file system), and fair share priority management (users are guaranteed a fair share of the resources according to pre-assigned priorities), Condor proves to be a very complete and sophisticated package. While providing functionality similar to that of any traditional batch queuing system, Condor's architecture allows it to succeed in areas where traditional scheduling systems fail. As a result, Condor can be used to combine seamlessly all the computational power in a community.

Grid workflows consist of a number of components, including computational models, distributed files, scientific instruments and special hardware platforms. Abramson *et al.* described an atmospheric science workflow implemented by web service [1]. Their workflow integrated several atmosphere models physically distributed. Hai studied the development of component-based workflow system, and he also studied the management of cognitive flow for distributed team cooperation [5]. We describe the Grid workflow in remote sensing science and show how it can be implemented using Web Services on a Grid platform in this paper. The workflow supports the coupling of a number of pre-existing legacy computational models across distributed computers. An important aspect of the work is that we do not require source modification of the codes. In fact, we do not even require access to the source codes. In order to implement the workflow we overload the normal file Input/Output (IO) operations to allow them to work in the Grid pool. We also leverage existing Grid middleware layers like Condor to provide access to control of the underlying computing resources.

In this paper we present remote sensing application workflow in building a Remote Sensing Information Analysis and Service Grid Node at Institute of Remote Sensing applications (IRSA) based on the Condor platform. The node is a special node of Spatial Information Grid (SIG) in China. The node will be introduced in Section 2. Several middleware developed in the node and the remote sensing workflow, with

some detail of the functions of the various components will be demonstrated in Section 3. Finally, the conclusion and further development will be addressed in Section 4.

2 Remote Sensing Information Analysis and Service Grid Node

2.1 Spatial Information Grid (SIG)

SIG is the infrastructure that manages and processes spatial data according to users' demand. The goal of SIG is to build an application Grid platform for spatial information community, which can simplify and shield many complex technology and settings, facilitate SIG users, and make good use of resources physically distributed. There are many reasons why one might wish to have the SIG. First, the amount of spatial data is increasing amazingly. So that real time or almost real time processing needed by applications confronts much more difficulties in one single computer. Second, data, algorithm, and/or computing resources are physically distributed. Third, the resources may be "owned" by different organizations. Fourth, the use frequency of some resources is rather low.

A SIG at least contains: (1) A remote sensing Remote Sensing Information Analysis and Service Grid Node; (2) A data service node: traditional data base to a web service; (3) A management centre: resource register, find, trade, and management; (4) A portal: an entry to SIG user.

2.2 Remote Sensing Information Analysis and Service Grid Node

Remotely sensed data is one of the most important spatial information sources, so the research on architectures and technical supports of remote sensing information analysis and service grid node is the significant part of the research on SIG.

The aim of the node is to integrate data, traditional algorithm and software, and computing resource distributed, provide one-stop service to everyone on Internet, and make good use of everything pre-existing. The node can be very large, which contains many personal computers, supercomputers or other nodes. It also can be as small as just on personal computer. Figure 1 describes the architecture of Remote sensing information analysis and service Grid node. The node is part of the SIG, but it also can provide services independently. There're two entries to the node:

1. A portal implemented by JSP. User can visit it through Internet browses, such as Internet Explorer and others.
2. A portal implemented by web service and workflow technology. It is special for SIG. Other Grid resources and Grid systems can integrate with our node through this portal.

The node provides the application services such as aerosol optical depth retrieval, land surface temperature retrieval, soil moisture retrieval, surface reflectance retrieval, and vegetation index applications by MODIS data.

Fig. 1. Architecture of remote sensing information analysis and service Grid node

The remote sensing node server (Figure 2) responds for all the corresponding events about the services on it. It contains:

- Node container: provide the necessary grid work environment;
- Service/workflow: some functional entities or workflows will be encapsulated to services for Grid. Others functional entities called within the node will be encapsulated to workflows. By adding new services into the node container, the capability of the node can be augmented and in turn the function of the whole system can be improved incrementally.
- Middleware: traditional applications cannot be used on condor directly. So middleware is needed to provide an access to condor. The middleware stands for the division and mergence of sub-tasks, and monitor the condor computing pool and the process of sub-tasks running in the pool.
- Node management tool: it responds for issuance, register, and update to SIG management centre. It's up to the node management tool to trigger service and monitor the status of the service.

The node issues and registers its services to SIG manage centre periodically, responses calls of SIG, triggers services, and reports status. Receiving require from SIG manager, the node will find data form either local or remote data server according to the user's requirements, organize computing resource dynamically, trigger services, and monitor their running status.

To decrease the total processing time of a task, the node will divide a task into several sub-tasks. The exact number and size of the sub-tasks is according to the current PC number and configure in the Condor computing pool and super computer. Only when the task is large enough, or on the user's request, the node triggers off the super computer to do a large sub-task. The method we trigger super computer is different with that of Condor computing pool. The Grid Service Spread (GSS) described by Wang *et al.* was used to implement it [6].

Implementing the node, we refer to use web service technology rather than Globus technology. The reasons are:

Fig. 2. Remote sensing node server

1. Globus is a new technology which progressing fast. From the issuance of Globus toolkit version 1.0 at 1999 to version 4.0 at 2005, the protocol and technology in it changed evidently. The reliability and practicality need textual research. Therefore, it's venturesome to follow it.
2. Web service is a mature technology, which has been tested out in the field of industry for decades of years. Globus is adjusting its steps to web service. So long as our node supports web service, our node will be compatible with Globus. Web service ensures the compatibility and expansibility of our node.
3. It's more complex and difficult to develop using Globus than using web service. There are some convenient tools available for web service. But few is for Globus.

3 Workflow Implementation with Web Service and Grid

Traditional method for link functional components is to program them into a static course, or involve human control. People have to realize all the permutation and combination of the components in order to deal with all possible cases. It need large amount of repeated work, and is discommodious to modify and extend. The workflow technology overcomes these shortages. Using the concept STEP, ACTION, STATE, and TRANSITION in finite state machine (FSM) for reference, workflow can skip among the components flexibly. So it's facility to organize components dynamically according the user request and the environment of the system.

As the temperature retrial example for demonstrating the implement of our node, which will be described in details in sections 3.1 and 3.2, we can cut-out any pre-processing components by configuring the workflow conditions and attributes. For example, if the primary data has been rectified outside our node, the workflow will skip the rectification component and turn into the next one.

The components called by workflows can be reused. So that people were set free from repeated work. For the loose coupling of the components and workflows, it's convenient for developers to modify the components themselves without harm to the workflows, as long as the interface of the components is unchanged. Furthermore, it is convenient to modify the workflows by adding or decreasing components in them.

We also benefit from the termination function of workflow technology. It is particular useful to some remote sensing applications which involve many steps and need long processing time. A workflow instance can be terminate by sending terminal signal artificially, if we don't want the instance execute anymore. An instance can stop itself when it finds out the components in it meet trouble or the required environment is not satisfied. But the workflow cannot stop a running component. It only can stop itself before or after the running component is finished.

Using workflow's status function, we can monitor the running status of a workflow and status of components in it. STATE has four meaning in the remote sensing node:

- State of the node: it describes the attributes of the node, such as busy, idle, current workload, current processing ability, service instance number in queue, etc.
- State of application/workflow instance: it presents whether the instance is finished, or which step is finished.
- State of components in an application: it has only two status: finished and underway.

- State of condor pool: it contains the detail configuration (operation system, IP address, architecture, CPU, and RAM) and running status (IDLE or BUSY) of the PCs in the condor computing pool, the number of the tasks submitted to condor.

3.1 An Implementation with Web Services on the Condor Platform

This implementation is for the connection to the SIG or other Grid application systems. Each application is implemented using web service technology such as SOAP, XML, and WSDL.

In our node, the workflow engine plays the role of a global coordinator to control different components to fulfil the functions requested by users according to a pre-defined process model. For example, Figure 3 presents the pre-defined process model of land surface temperature retrieval. It involved the functional component of data format transfer, rectification, region selection, division, land surface temperature retrieval, merge, and return result to caller. The workflow will skip the functional components in front of the component whose input requirement has been satisfied by the primary data. Unfortunately, the system cannot recognize whether the data has been transferred or rectified without further information. It's necessary for the data node or someone else to describe these characters in a data metafile. The workflow will decide which components should be skipped by the metafile.

The division component calls the middleware shown in Figure 2. The middleware stands for checking the current status of condor computing pool, dividing data into pieces according to the number of the PCs in condor, generating all the files needed by Condor, packaging them, and transferring them to the location where condor can find them. The workflow submits the temperature component and data pieces to Condor. Then Condor stands for manage these sub-tasks. When Condor returns the results of sub-tasks, the merge component will integrate them into the final result. And return component transfer this final result to caller. In this test, there were 5 PCs running operation system of Windows 2000 professional.

Fig. 3. Process model of land surface temperature retrieval

3.2 An Implementation with JSP on a Condor Platform

This implementation is for users to visit our node directly. The difference between the JSP implementation and web service one is:

1. It is not a web service application. There are no WSDL files and WSDD files for the JSP implementation. It only can be triggered and initialized artificially. It cannot be interact by other machines.

2. The implement with JSP can be man-machine interactive. Of course it can auto run, too. An application can be intervened when it is running.
3. It provides a data up load function before data format transfer component. So that users can process their data using the functions on our node.

Figure 4 shows the execution course of the land surface temperature retrieval application.

Fig. 4. The execution course of a temperature application workflow instance

4 Conclusions and Future Work

In this paper, we introduced our ongoing research on remote sensing information analysis and service Grid node. The node is service-oriented. The whole node is composed of many workflows. Each workflow stands for certain service. Some of the workflows were implemented with web service technology, so that they can be called by the SIG or other systems, which follow web service protocol. Web service technology endows our node with compatibility and machine-machine interaction. Other workflows are for JSP web site, so that user can visit our node directly. They only can be triggered artificially. The JSP technology endows our node with man-machine interaction. The execution course of the JSP workflows can be intervened by users. Workflow technology endows our node with easy modification and extensibility. SIG and Condor endows our node with the power of remote cooperation, resource share, and management physically distributed.

Our node is a demonstration to how to integrate data, traditional algorithms and models, and computing resource in order to provide one-stop service to users. We have implemented it mainly by workflow technology. We try to make a large virtual super computer by integrating cheap personal computers on Internet, and utilize them when they are left unused. Theoretically, the node can provide enough processing ability to anyone at anytime and anywhere if there are enough PCs left unused in it.

The services on the node seem to work well, but actually it is far away from intactness. There are still many problems to deal with. One problem is the long file transfer time between the PCs physically distributed. It weakens the decrease of the total processing time improved by using more PCs doing one task. To some multi-band image data, a feasible approach to deal with it is to transmit only the bands needed by the applications. The amount of the data may decrease a scalar grade.

Acknowledgement

This publication is an output from the research projects "Grid platform based aerosol fast monitoring modelling using MODIS data and middlewares development" (40471091) funded by the NSFC, China, "Dynamic Monitoring of Beijing Olympic Environment Using Remote Sensing" (2002BA904B07-2) and "Remote Sensing Information Processing and Service Node" (2003AA11135110) funded by the MOST, China and "Digital Earth" (KZCX2-312) funded by CAS, China.

References

- [1] Abramson, D., Kommineni, J., McGregor J. L., and Katzfey, J., 2005, An Atmospheric Sciences Workflow and its implementation with Web Services. *Future Generation Computer Systems*, **Vol.21**, Issue 1, pp.69-78
- [2] Basney, J., Livny, M., and Tannenbaum, T., 1997, High Throughput Computing with Condor, *HPCU news*, **Volume 1(2)**
- [3] Foster, I., and Kesselman, C., 1997, Globus: a metacomputing infrastructure toolkit, *Int. J. Supercomputer Application*, **11 (2)**, 115–128.
- [4] Foster, I., and Kesselman, C. (Eds.), 1999, *The Grid: Blueprint for a New Computing Infrastructure*, (Morgan Kaufmann Publishers, USA).
- [5] Hai, Z. G., 2003, Component-based workflow systems development, *Decision Support Systems*, **Volume 35**, Issue 4, Pages 517-536
- [6] Wang, Y. G., Xue, Y., Jianqian Wang, Chaolin Wu, Yincui Hu, Ying Luo, Shaobo Zhong, Jiakui Tang, and Guoyin Cai, 2005, Java-based Grid Service Spread and Implementation in Remote Sensing Applications. *Lecture Notes in Computer Science*, **Vol.3516**, pp.496-503.