A HIGH PERFORMANCE REMOTE SENSING RETRIEVAL APPLICATION ON AN INSTITUTIONAL DESKTOP GRID

Wan Wei1,3 Yong Xue1,2 *, Senior Member, IEEE, Shuning Lu3, and Jie Guang1,4

1State 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
2Department of Computing, London Metropolitan University, 166-220 Holloway Road, London N7 8DB,
3China Center for Resource Satellite Data and Application, Beijing 100830, China
4Graduate School of the Chinese Academy of Sciences, Beijing, China

Email: eric_104@163.com y.xue@londonmet.ac.uk

ABSTRACT

In this paper, we focus on making desktop Grids adapted to remote sensing application. The initial efforts were made to build a remote sensing retrieval application on the RSIN framework for dealing with climate change, named High Performance Aerosol property Retrieval Software (HiPARS). HiPARS uses integration of applications that access the backend of desktop Grid comprising of office PCs with a client. The HiPARS remote sensing application was built on RSIN framework, which uses the loose-coupled architecture of a desktop client and high throughput computing (HTC) and Grid backend. It allows for fast development by enabling existing code and new algorithms, and provides a familiar graphical environment for remote sensing users. The proposed solution is to transition the current PCs to a desktop Grid that can be cost-effectively sustained. It will be accessible for processing satellite data in quasi real-time, for parameter retrieving or for parallel processing efforts.

Index Terms— Desktop Grid, Grid computing, Quantitative retrieval, Aerosol optical thickness, Remote Sensing

1. INTRODUCTION

Quantitative retrieval of aerosol property from remotely sensed data is a both data and computing intensive scientific application, where the complexities of processing, modeling and analyzing large volumes of remotely sensed data sets have significantly increased computation and data demands. The remote sensing quantitative retrieval within the atmospheric aerosol domain is an example of the type of scientific application that will benefit from an advanced Grid computing [1]. Grid computing takes advantage of many networked computers to model a virtual computer architecture that is able to distribute process execution across a distributed infrastructure [2]. Grids gather the computational resources of separate computers connected by a network to solve large-scale computation problems. Grids provide the ability to perform computations on large data sets by breaking them down into many smaller ones.

Institutional desktop Grid comprised by the desktop machines of an institution (academic or corporate) can act as an important source of computing power for local user, if the high volatility of such resources is properly harnessed. Office PCs sit unused during idle hours, which can easily count for large portion of computing time [3]. An institutional desktop Grid using existing office PCs has the potential to be an enormous computing resource, particularly considering current and planned connections of network [4].

In this paper, we focus on making desktop Grids adapted to remote sensing application. For that purpose, a middleware framework for remote sensing retrieval named the Remote Sensing Information service grid Node (RSIN) is described. We proposed the design of the Remote Sensing Information service Grid Node (RSIN) framework [5] and the High Performance Aerosol property Retrieval Software (HiPARS) application [6]. RSIN framework meets the needs of the remote sensing community by providing high performance, ease of use, and extensibility.

This framework has features:
1) Run on high throughput computing Grid,
2) Enable a workflow system for task management and data placement,
3) Enable accompanying unified data-and-computation-schedule algorithm and a task partition
algorithm, which help load balancing between and within workflow steps,
4) Be usable without advanced knowledge of Grid architecture or parallel algorithms,
5) Provide a mechanism for code reuse and module Grid-enabling, and
6) Provide a graphical environment and tools that facilitate fast model development, data visualisation and performance analysis.

In this paper, the initial efforts were made to build a remote sensing retrieval application on the RSIN framework for dealing with climate change, named High Performance Aerosol property Retrieval Software (HiPARS). HiPARS uses integration of applications that access the backend of desktop Grid comprising of office PCs with a client. The HiPARS remote sensing application was built on RSIN framework, which uses the loose-coupled architecture of a desktop client and high throughput computing (HTC) [7] and Grid backend. It allows for fast development by enabling existing code and new algorithms, and provides a familiar graphical environment for remote sensing users. A graphical environment is provided which allows users to specify a workflow for the application by creating and connecting pluggable modules, launch jobs, select parameters and data, analyze performance and visualize results. Issues dealing with how to achieve the parallelism, schedule tasks, and pass data are hidden from the user. With HiPARS, advanced understanding of Grid system has been strayed away from remote sensing users.

The proposed solution is to transition the current PCs to a desktop Grid that can be cost-effectively sustained. It will be accessible for processing satellite data in quasi real-time, for parameter retrieving or for parallel processing efforts.

The remainder of the paper was organized as follows. Section 2 describes the desktop Grid solution to remote sensing retrieval application. Section 3 describes the framework of the desktop Grid application. Section 4 describes experimental results. We concluded in section 5 with a summary and a brief discussion of future work.

2. DESKTOP GRID SOLUTION TO REMOTE SENSING RETRIEVAL APPLICATION

The proposed solution is seeking low-cost sources of additional computing cycles to provide remote sensing retrieval services. For this purpose, we designed the desktop grid solution for remote sensing retrieval, with an architecture and deployment using networked PCs in the institution. As an added benefit, desktop grid processes data and parallels the processing of them over a network of PCs running on the common platform available, Linux and Windows, and administrators can customize the computing environments and tune the processing flow. The system uses the existing Desktop Grid software. To implement, the solution needs transition from the current cluster to a Desktop Grid with up to 50 PCs, and technology transfer improvements for applications moving from PC-based environments to an operational status.

3. HIPARS SYSTEM

3.1. System description

The High Performance Aerosol property Retrieval Software (HiPARS) is both a component-based software and an application for quantitative retrieval of aerosol property [8]. The purpose of this project is to develop an operational application for atmospheric remote sensing, building on the RSIN grid that is made up of commodity PC workstations. The toolkit provides user-friendly capabilities for performing parameters retrieval and information analysis using computational Grids, and developing modular codes for parallel computers. It helps technical users directly benefit from accessing cyberinfrastructure capabilities.

HiPARS modules are subroutines in IDL, Fortran, Java and C. IDL (Interactive Data Language) is an array-oriented data analysis and visualization application, which is used for efficiently developing customized Remote Sensing algorithms. It is meaningful to make user algorithms collaborative between computers over networks, using Grid as the underlying communication system. The aerosol retrieval models along with other RS algorithms are written in IDL or Fortran, and control modules are usually written in Java.

HiPARS customizes the RSIN middleware to the remote sensing domain by changing the terminology presented in the interface to be more familiar to the remote sensing user. It provides basic functions for selecting datasets through file lists, choosing a region of interest, calibrating data, geo-referencing, stitching mosaic images, enhancing image, screening cloud, converting map projection, and visualization of results. Additionally, a Grid Service Interface assists in development of a grid application by providing a place for describing the dataflow graph, launching jobs on the parallel computer, visualizing results in real-time, and analyzing performance after job completion.

With the concept of loose-coupled integration of applications presented in HiPARS design, we have brought together in the research a Grid-based collaboration paradigm, a desktop front-end, a grid pool back-end, RSIN middle wares, a workflow management, and Grid-enabled remote sensing programs.

3.2. Major Components

HiPARS is built upon three major technologies: the desktop based GUI client, the Job Management Framework (JMF) and the D-W-S (Decompose, Wrap and Submit) method. JMF is a low-level middleware for managing tasks created by user, maintaining task lists, and mapping them to job
queues on parallel computers. D-W-S is a routine used to enable existed applications or legacy codes on the grid environment. D-W-S (Decompose, Wrap and Submit) routine is our method to enable existed application or legacy codes on Grid. It is composed of: Functionality Decompose, Module Wrapping, Modules Interface Description format (MID).

Job Submitting Code - Submitting Code is the module call interface. It actually is glue codes (like batch or shell scripts) that are loaded dynamically by the Module Wrapping so that all operations are called by local scheduler/RPC and run on remote machines.

Building a customized Grid-enable remote sensing application from stretch is not trivial work, therefore, we chose the RSI’s ENVI 4.3 as our software development kit. ENVI is a powerful RS image process environment that allows users to access the core functionality of ENVI modules outside the monolithic IDL’s integrated software development environments. In other words, ENVI module provides a feasible channel for the component-based integrated application using both existed image process modules and Grid. Particularly, ENVI Batch Programming Interfaces (APIs) and IDL-Java Bridge were used to build the Grid-enabled modules.

3.3. Configuration and process flow
In our design and development, we deploy the Condor Grid as a Grid and use it for resource management and meta-scheduler [7]; and we deploy the RSI and HiPARS as application levels on top of infrastructure and make them collaborate under control. The process of generating a specific configured system comprehends the following phases:

Install and Deploy - The system designer selects the necessary RSI middleware and HiPARS components. RSI middleware, JAVA, and RSI ENVI environment are installed on every work node. HiPARS client is required on the submit node. In this phase, new components may be projected or existent ones may need adaptation;

System Configuration - The system administrator registers the physical and logical configuration in the Grid service interface component. For this, two scripts files are executed by the operations: register-gridnode-resource and register-software-resource.

Application Configuration - To build a RS retrieval application, the application designer (HiPARS user) define local, external, and global metadata to describe modules through three scripts files to be processed by the define-module-metadata function. Modules that perform a RS function each are presented in a remote sensing perspective. Then expert user can place them in workflow steps to assemble a RS application.

According to the application requirements, components are included. The range of possible configurations of HiPARS can vary from middleware with the functionality of a simple wrapper to a complex workflow manager. If a component is needed in a session, it is deployed beforehand on the same directories among the grids. In cases where a full HiPARS functionality is required, the middleware system should be deployed on both the hosts of the HiPARS and all the participant nodes.

Parallelism is achieved with HiPARS with the data parallel paradigm. Advanced users can use distributor modules WEA and DPA which break satellite datasets into partitions, pass them to different processes on the parallel machine, and reassemble the results. Queues of data to be processed are maintained internally which facilitate load balancing between the processors. With these queues, once a processor is done with a portion of the dataset it merely requests an additional portion from the master process. Parallelism takes forms as follows: Workflow level, Job Queue Level, Thread Level.

Performance Analysis - After jobs have been run, HiPARS provides performance logs and analysis of many internal events which can be useful for hand-tuning between modules.

HiPARS allows advanced users can extend their own functionality to achieve parallelism. For instance, users can use the Message Passing Interface (MPI) to communicate between parallel modules if a problem does not fit the simple data partitioning model, or create a pipeline parallel as data pass through the system from node to node to improves performance of the queuing system.

4. APPLICATION

The following case study explores the challenge of retrieving data for an aerosol optical thickness application. An example of a module is one that calculates the inversion of aerosol optical thickness parameters from multi-bands of MODIS satellite images. This example shows the process of taking existing (legacy) code and incorporating it within HiPARS. The experiments are presented in a realistic application, using real data collected by MODIS over land area of China.

The beginning of the function uses four calls to HiPARS MODIS L1B data per-process functions. These functions retrieve the requested data from file storage lists transparently.

In HiPARS, functions calls invoke the JMF to create a list of tasks and the JMF maps tasks to grid jobs before submitting them to Grid back-end.

After data processes are completed, call in the AOT RETRIEVAL module is made to retrieve AOT values from band data. This module will be run by many processors many times and each time will operate on only a portion of the whole dataset. The WEA determines each time the module runs exactly which scan lines will be processed.
JMF is responsible for monitoring the jobs by querying the Grid job manager.

Finally, there are loops which iterate over the total number of scan lines given to a particular processor. These loops are where image post-process occurs.

As these modules are executed by sequence on each slave node, the workflow will contain data relevant only to a portion of a dataset. We achieve a great deal of data parallelism with this HiPARS method.

4. CONCLUSIONS

In this work, we propose the HiPARS, which is designed as an integrated framework with a desktop graphical client and a Grid back-end. We aim to provide high performance computing, extensibility, and usability for remote sensing aerosol retrieval problem. A framework for modular code development and easy pluggability of modules is provided. This framework makes it easy to both incorporate existing, legacy software as well as develop new algorithms in the Grid environment. The GUI addresses usability by providing tools for workflow graph creation, data visualization, grid service interfaces and additional functionality specific to remote sensing problems such as calibration, image processing, choosing map projections, and retrieving aerosol properties.

We are now implementing the first version of the HiPARS application. It is being used to build applications for the realm of remote sensing science.

5. ACKNOWLEDGMENT

This work was supported in part by Chinese Academy of Sciences under Grant No. KZCX2-YW-313, by National Science Foundation of China under Grant No. 40671142 and Grant No. 40471091, by the Ministry of Science and Technology, China under grant nos. 2008AA12Z109 and 2007CB714407.

6. REFERENCES


