Grid-HSI: Using Grid Computing to Enable a Hyperspectral Imaging Testbed

Carmen L. Carvajal-Jiménez
Advisor: Dr Wilson Rivera

Center for Subsurface Sensing and Imaging Systems
Laboratory of Applied Remote Sensing and Image Processing
Electrical and Computer Engineering Department
University of Puerto Rico, Mayagüez Campus
Email: carmen.carvajal@ece.uprm.edu

ABSTRACT

Hyperspectral imagery analysis demand large input data sets, as well as great time of processing and memory capacity. Parallel computers can accelerate the performance of these applications. The potential for massive parallel CPU capacity is one of the most attractive features of a grid.

This paper provides a short description of previous software tools have been developed to enable Grid Computing, and then performance analysis tools. Last section is dedicated for description of the prototype for a computational Grid testbed future work.

1. INTRODUCTION

Hyperspectral Imaging (HSI) processing is based on the notion of imaging spectrometry where spectral and spatial information is used to identify or detect objects as well as estimate parameters of interest. Hyperspectral imagery analysis demands large input data sets and requires significant cpu time and memory capacity. currently it is needed for HSI researchers at the NSF Engineering Research Center for Subsurface and Imaging Analysis (CENSSIS) to develop tools that allow them to make faster and efficient the work with image processing. Parallel computers can be used to significantly reduce the runtime of these applications. Moreover, Grid-level resources can play a significant role in improving performance and pervasity of the image processing algorithms.

We propose a toolbox immersed into a grid platform that supports unsupervised classification. The toolbox contains routines for dimensionality reduction and unsupervised classification; we focus on the problem parallelization and its mapping onto a grid system. And also build our testbed on existing grid computing software such as globus and MPICH-G2. In this paper, we present the description of the three main components of architecture of the system.

2. BACKGROUND

2.1 Hyperspectral Image Analysis

Hyperspectral Image processing is becoming increasingly popular due to the wealth of data hyperspectral images can provide geologists, oceanographers and the military. These images consist of the same image taken in different spectral bands. Hyperspectral Image exhibit high correlation between the pixels in both the spatial and spectral dimensions.

Rivera [5] implemented a system nonsupervised iterative system of reduction of bands and classification of pixels that integrates these two stages through a closed knot. It considers the number of pixels classes as entrance parameter to carry out the reduction of the dimension of the image. The objective of the system is to choose a good subset of bands in which the distance is maximized among the classes or centroids. The system was implemented using matlab.

A MATLAB Toolbox for Hyperspectral Image Analysis [1] which focuses on the optimization and integration of the unsupervised and supervised HSI classification algorithms has been developed at the UPRM Laboratory for Applied Remote Sensing and Image Processing (LARSIP). This toolbox contains algorithms to load images of several file formats, routines for dimensionality reduction, and both supervised and unsupervised classification algorithms.

2.2 Grid Computing

The Globus grid toolkit is a collection of software components designed to support the development of applications for high-performance distributed computing environments [Foster98]. MPICH-G2 extends the implementation of MPICH to be able to use the services provided by Globus. MPICHG2 allows connecting multiple machines in different architectures to execute MPI implemented applications using Globus services.

Grid computing involves coordinating and storage and network resources across dynamic and geographically dispersed organizations in a transparent way for users. Grid technologies emphasize effective operation in large scale, multiinstitutional, wide area environments, including access to remote computation, information services, high speed data transfers, special protocols (e.g., multicast), and gateways to local authentication schemes.

A number of software tools have been developed to enable Grid Computing . Among then we can point at:

1. Globus is an multi-institutional initiative for the investigation and the development of fundamental technologies for Grids. Globus is the standard de facto for the implementation of Grid applications. The toolkit addresses issues for resource monitoring, resource discovery and management, security and file management. Java CoG Kit (CoG Kit) defines and implements a set of general components that map Grid functionality into a commodity environment/framework [3]. The idea is that the Java CoG kit implements Globus functionality though Java standards and Java Beans. This provides access to Grid services using Tomcat and Apache as well as the latest reference implementation of Java Servlets and Java Server Pages.

2. Grid Portal Development Kit [4] is designed to provide access to Grid services by using Java Server Pages (JSP) and Java Beans using Tomcat. The GPDK Java beans derive most of their functionality from the Globus Java Commodity Grid (CoG) toolkit. Java CoG Kit [Laszewski02] defines and implements a set of general components that map Grid functionality into a commodity environment/framework [Foster01]. The idea is that the Java CoG kit implements Globus functionality in a manner that integrates well with existing Java standards and Java Beans.

3. ALiCE [Teo02], developed at the National University of Singapore, is a Java-based grid computing middleware that supports the development and execution of generic Grid applications over a geographically distributed, heterogeneous collection of resources. ALiCE supports job parallelism to maximize throughput and object (task) parallelism to maximize performance.

4. Espresso [7] implements a high perfomance climate modeling system as an example of the application of Grid based technology to climate modeling. The climate simulation system at Argonne currently includes a scientific modeling interface (Espresso) written in Java which incorporates Globus middleware to facilitate climate simulations on the Grid. The climate modeling system also includes a high performance version of MM5v3.4 modified for long climate simulations on our 512 processor Linux cluster (Chiba City), an interactive web based tool to facilitate analysis and collaboration via the web, and an enhanced version of the Cave5D software capable of visualizing large climate data sets.

3. FEEDBACK ITERATIVE METHOD ALGORITHM

The system nonsupervised iterative system of reduction of bands and classification of pixels [Rivera] that integrates these two stages through a closed knot. It considers the number of pixels classes as entrance parameter to carry out the reduction of the dimension of the image.

3.1. Conformation of combinations of classes and bands.

First an arrangement of all the possible combinations should be built, with a number of elements same to the dimension of reduction of the total of bands of the image. Should also be formed another arrangement of all the possible combinations, of two elements of the total of established classes.

3.2. Condition initials.

The system to start requires noticing the following parameters: dimension of the reduction, corresponds to the number of bands that are wanted to choose. Number of classes or pixels groupings that will be structured. Number of iterations or times that will repeat the stages of dimension reduction and classification. For finish the finalization threshold that is an optional parameter and similar to the wanted minimum value of the sum of the distances to the square (SSD) that when it is reached, the system concludes its operation without caring if the has ended with all the established iterations.

Besides setting up the previous parameters, it is required of an initial classification of the pixels, which could be prepared assigning to each pixel arbitrarily one determines class or choose arbitrarily a subset of bands to provide the values of the pixels in this subset to an algorithm of classification nonsupervised.

3.3. Selection of bands

If it is used as function approach the Distance Euclideana among centroides, the system uses the arrangements structured in the previous step to calculate the centroides of the classified data and to look for the subset of bands that provides the biggest distance average among the centroides or the subset that it provides the between two bigger than the smallest distances centroides.

If it is used as function approach the distance bhattacharyya among classes, the system moreover calculating the centroids of the classified data, calculates the covariance matrix for each class and search the band subset that maximizes the distance among the classes or the subset that it provides the between two bigger than the smallest distances classes.

3.4. Pixels classification in low dimension

With the subset of chosen bands the system carries out a process of píxeles classification using the function discriminate of distance euclideana or the function discriminate of maximum verisimilitude.

4. GRID PERFORMANCE METRICS

Usual metrics such as response time, resource usage, through put, efficiency are not necessarily applicable for grids because the grid performance is not characteristic to an application itself rather to the interaction between the application and the infrastructure. Performance tuning is more difficult due to dynamic environment, changing infrastructure, diversity and heterogeneity of resources. While performance evaluation of parallel and distributed systems there exist practical solutions, in most cases these techniques cannot be transferred directly to grids. In the last years the research communities is being explored introduce novel approaches to performance monitoring and evaluation.

A number of performance analysis tools have been developed to enable Grid Computing. Among then we can point at:

1. Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. It is based on a hierarchical design targeted at federations of clusters, relies on a multicast-based listen/announce protocol to monitor state within clusters and uses a tree of point-to-point connections amongst representative cluster nodes to federate clusters and aggregate their state [8].

2. GridMon is a network performance monitoring toolkit to identify faults and inefficiencies. The toolkit is composed of a set of tools that are able to provide measures concerning different aspects related to network performance [8].

3. MapCenter is a flexible monitoring system and presentation layer of the services and applications available on a Grid. It incorporates automatic resources discovery from MDS, R-GMA, or Web browsing as well as stealth monitoring techniques to avoid logs on end systems and ease the crossing of firewalls [8].

4. VAMPIR is a tool for performance analysis of message passing applications based on program traces. VAMPIR is currently being extended in the DAMIEN project for multisite Grid applications. The applications are coupled via MpCCI implemented on top of PACX-MPI. In addition, enhancements to VAMPIR allow to analyze the communication infrastructure [8].

5. PROTOTYPE FOR A COMPUTATIONAL GRID TESTBED

The architecture of the system is divided in three components.

Portal Grid: interface that allows the user to introduce all the necessary parameters for the execution of the program. This component takes charge of the validation of users and the distribution of the tasks sent by the user among all the available resources that provides Infrastructure server.

First the portal grid obtains the authentication using the certification to obtain the credential of Globos. and after that, communicates with the Infrastructure Server requesting the necessary resources. Finally, it invokes the service Master Manager Job Factory Service (MMJFS) of Globus to execute the tasks in the remote resources indicated by the served of nodes.

On the user interface it is possible to choose the image, numbers of bands, number of iterations and classification, number of classes, method of selection bands, method of classification (Not supervised) (See Figure 1).

Infrastructure Server: it contains a database with informed on all the available nodes for the execution of tasks in a given instant of time and on the tasks that are already being executed in the Grid. The server takes charge of the administration of nodes that continually they are added or eliminated the Grid, and the administration from the assignment of nodes to the users.

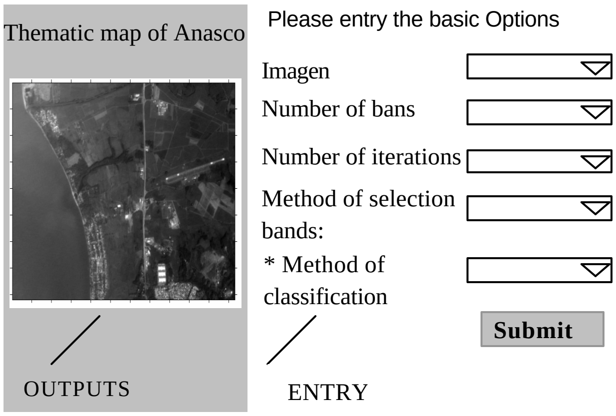

Figure 1. Interface of the application

This communicates with the portal grid to provide the available resources and to inform of possible changes.

Nodes: they are resources available potentially like part of the grid for the execution of works; each one of the nodes contains an activity demon (DA) running that it notifies to the Infrastructure Server the readiness of that node for the execution of remote tasks.

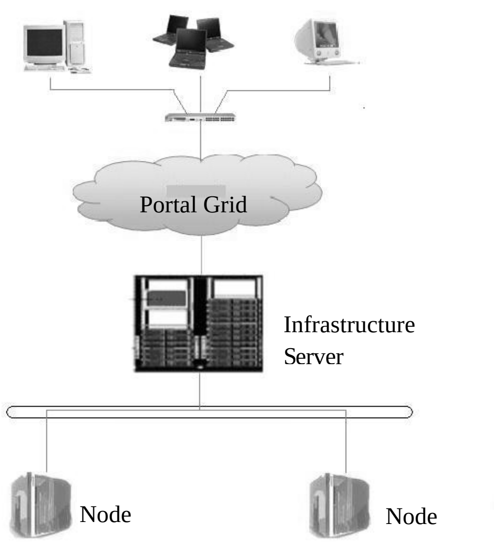

Figure 2. Architecture Computional Grid Testbed

6. REFERENCES

[1] Arzuaga-Cruz E., Jimenez-Rodriguez L. O., Kaeli D., Velez-Reyes M., Rodriguez-Diaz E., Santos- Campis L. E., Santiago C. “A MATLAB Toolbox for Hyperspectral Image Analysis”, 2003.

[2] Foster I. and Kesselman C. “The Grid: Blueprint for a Future computing Infrastructure”. Morgan Kaufmann Publishers. 1998.

[3] Laszewski G., Foster I., Gawor J., Lane P. “A Java Commodity Grid Kit”, Proceedings of the Concurrency and Computation Journal: Practice and Experience, 2001.

[4] J. Novotny. “The Grid Portal Development Kit”, Accessed from Http://Www.Cogkits.Org/. 2001.

[5] D. Rivera-Gallego. “Feedback in Pattern Recognition”, Master in Science Thesis, University of Puerto Rico, Mayaguez, 2002.

[6] Y.M. Teo, S.C. Tay, and J.P. Gozali. “Distributed Geo-rectification of Satellite Images using Grid Computing”, Proceedings of the Internacional Parallel and Distribuited Processing Symposium, IEEE Computer Society Press, Nice, France, 2003.

[7] J. Taylor, M. Dvorak, and S. Mickelson. “Developing Grid based infrastructure for climate modeling”, Proceedings of the International Conference on Computational Science, Amsterdam, 2002.

[8] R. Wismuller, M. Gerndt. “Performance Tools for the Grid: State of the Art and Future”, 2003.