Proceedings of the Third IASTED International Conference
Communications, Internet, and Information Technology
November 22-24, 2004, St. Thomas, US Virgin Islands

GRID-HSI: USING GRID COMPUTING TO ENABLE HYPERSPECTRAL IMAGING ANALYSIS

Carmen L. Carvajal-Jiménez[1], Wilfredo Lugo-Beauchamp[2], and Wilson Rivera[1]

[1]Electrical and Computer Engineering Department
University of Puerto Rico, Mayagüez Campus
P.O.Box 9042, Mayaguez, Puerto Rico, USA
Email: {carmen.carvajal, wilson.rivera}@ece.uprm.edu

[2] Software Solutions Group
Hewlett Packard Technology Center
Aguadilla, Puerto Rico, USA
Email: wilfredo.lugo@hp.com

Abstract

This paper outlines the design and implementation of Grid-HSI, a Service Oriented Architecture-based Grid application to enable hyperspectral imaging analysis. Grid-HSI provides users with a transparent interface to access computational resources and perform remotely hyperspectral imaging analysis through a set of Grid services. Grid-HSI is composed by a Portal Grid Interface, a Data Broker and a set of specialized Grid services. Experimental results show the suitability of the prototype system to perform efficiently hyperspectral analysis.

Key Words

Grid computing, portal grid, remote sensing, image processing, hyperspectral images.

Introduction

Hyperspectral imaging allows a spatial scene to be decomposed into multiple two-dimensional images obtained at different spectral bands. These images can then be analyzed to discriminate among different features within the scene. Most applications of hyperspectral imagery require processing techniques that achieve one fundamental goal: detect and classify the constituent materials for each pixel in the scene. With the rapid advances in the resolution, frame rate, and dynamic range of spectrometers, the required bandwidth has soon exceeded throughput limits inherent in store and process systems. Thus, hyperspectral imaging analysis demands large input data sets and requires significant CPU time and memory capacity. Parallel computing has been succesfully used to significantly reduce the runtime of these applications. It is then expected that Grid-level resources can play a significant role in improving performance while increasing pervasity of the image processing algorithms.

A HSI analysis toolbox immersed into a grid platform, which supports remote analysis and visualization, has been developed. Our first Grid-HSI prototype is composed by a Portal Grid Interface, a Data Broker and a number of Grid services to enable HSI analysis. This paper outlines the Grid-HSI architecture and its implementation.

The structure of this paper is as follows. Section 2 provides an overview of hyperspectral imaging and Grid services technologies. Section 3 describes the architecture and components of Grid-HSI. Section 4 discusses related works. Section 5 presents experimental results. Finally, section 6 draws conclusions and potential future work.

2. Background

2.1 Hyperspectral Imaging

Sensors based on imaging spectrometry or so called hyperspectral imagers collect high spectral resolution data over a couple of hundred of wavelengths effectively producing an image where at each pixel we get the spectral response of the object(s) in the field of view of the sensor. Hyperspectral Imaging (HSI) data contains high spectral resolution and spatial information of the object under study. HSI analysis is based on the concept of imaging spectrometry where spectral and spatial information is used to identify or detect objects, or estimate parameters of interest (Figure 1). As the object of interest is embedded in a complex media (i.e. coastal waters or skin), the measured signature is a distorted version of the original object signature (e.g. a coral reef or a blood vessel) mixed with clutter. By large hyperspectral imaging analysis concentrates on dimensionality reduction and classification algorithms. Dimensionality reduction algorithms reduce the data volume (dimensionality), without loss of critical information, so that it can be processed efficiently. Classification of a hyperspectral image sequence, in turn, identifies which pixels contain

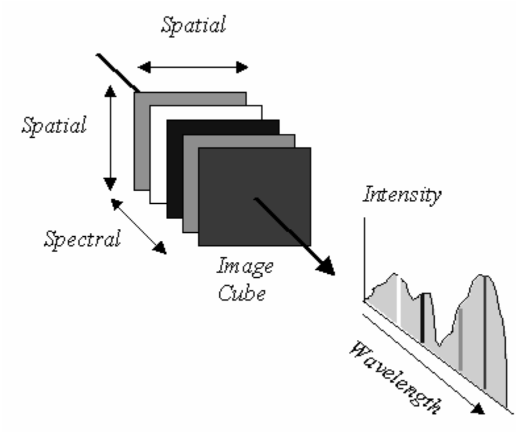

Figure 1. Hyperspectral Imaging

various spectrally distinct materials. Once all pixels are classified into one of several classes or themes, the data may be used to produce thematic maps. Depending on the nature of the application, the thematic maps may be used to produce summary statistics regarding the objects in a scene or for object or target recognition purposes.

Different classification metrics have been proposed from minimum distance, such as Euclidean, Fisher Linear Discriminant, and Malahanobis, to maximum likelihood [1] to correlation matched filter-based approaches such as spectral signature matching [2]. There are two major techniques to image classification: supervised and unsupervised. In supervised classification techniques, an analyst develops quantitative descriptions of the spectral characteristics of the various classes of interest for a particular scene. These descriptions are then used as reference spectral signatures against which every pixel in an image is compared. The pixels are classified according to the spectral signature they most closely resemble. In unsupervised classification, the algorithms do not use training data as the basis for classification. Instead, the algorithms examine the unknown pixels in the image and aggregate them into various classes according to the clusters found in the spectral space that contains the image.

A Matlab Toolbox for hyperspectral imaging analysis [3] has been developed by researchers at the NSF Enginnering Center for Subsurface Sensing and Imaging Systems (CenSSIS). This toolbox contains algorithms for dimensionality reduction and classification. In addition Rivera [4] implemented an unsupervised iterative system of reduction of bands and classification of pixels. It considers the number of pixels classes as input entrance parameter to carry out dimensionality reduction. The objective of the system is to choose a good subset of bands in which the distance is maximized among the classes or centroids. We have developed efficient sequential and parallel (MPI/C) implementations of a subset of algorithms in [3] and [4], as demonstrated in [5].

2.2 Grid Services

Grid computing [6] involves coordination, storage and networking of resources across dynamic and geographically dispersed organizations in a transparent way for users. The Open Grid Services Architecture (OGSA) [7] and its associated implementation, the Globus Toolkit 3.0 [8], are becoming a standard platform for Grid services and application development, based upon Web services protocols and open standards.

Web services [9] is a distributed computing technology (like CORBA, RMI, Enterprise Java Beans, etc) that allows us to create client/server applications. Web services are platform-independent and language-independent since they use standard XML languages. Web services are more adequate for loosely coupled systems where a client might have no prior knowledge of the Web service until it actually invokes the service.

The Open Grid Services Architecture (OGSA) defines Grid Services [10] as extensions of Web services. Thus, Grid services are basically Web services with improved characteristics to make them adequate for Grid-based applications. The Open Grid Services Infrastructure (OGSI) [11] provides a formal and technical specification of Grid Services. The Globus Toolkit 3.0 (GT3) [8] implements OGSI plus some other services. Web Services Resource Framework (WSRF) [12] represents a refactoring and evolution of OGSI.

3. Grid-HSI Architecture

This section describes the architecture underlying Grid-HSI. The components of the Grid-HSI architecture are described as follows.

Portal Grid Interface: This interface allows users to enter the required input parameters for executing Grid services associated to each of the HSI algorithms implemented in Grid-HSI. Users follow an authentication process to access the resources. This authentication process is based on Grid Security Infrastructure (GSI) [13], which delivers a secure method of accessing remote resources. It enables a secure, single sign-on capability, while preserving site control over access control policies and local security infrastructure. The portal Grid Interface is implemented with servlets [9], which provides web-based GUI design for user interfaces. The Portal Grid Interface uses Java servlets hosted within a Tomcat servlet container environment. All requests to the Portal Grid Interface go through an Apache server, which forwards requests to the Tomcat using Apache Jserv Protocol (AJP).

Data Broker: This component is a link between the Portal Grid Interface and the HSI Grid services. For each HSI

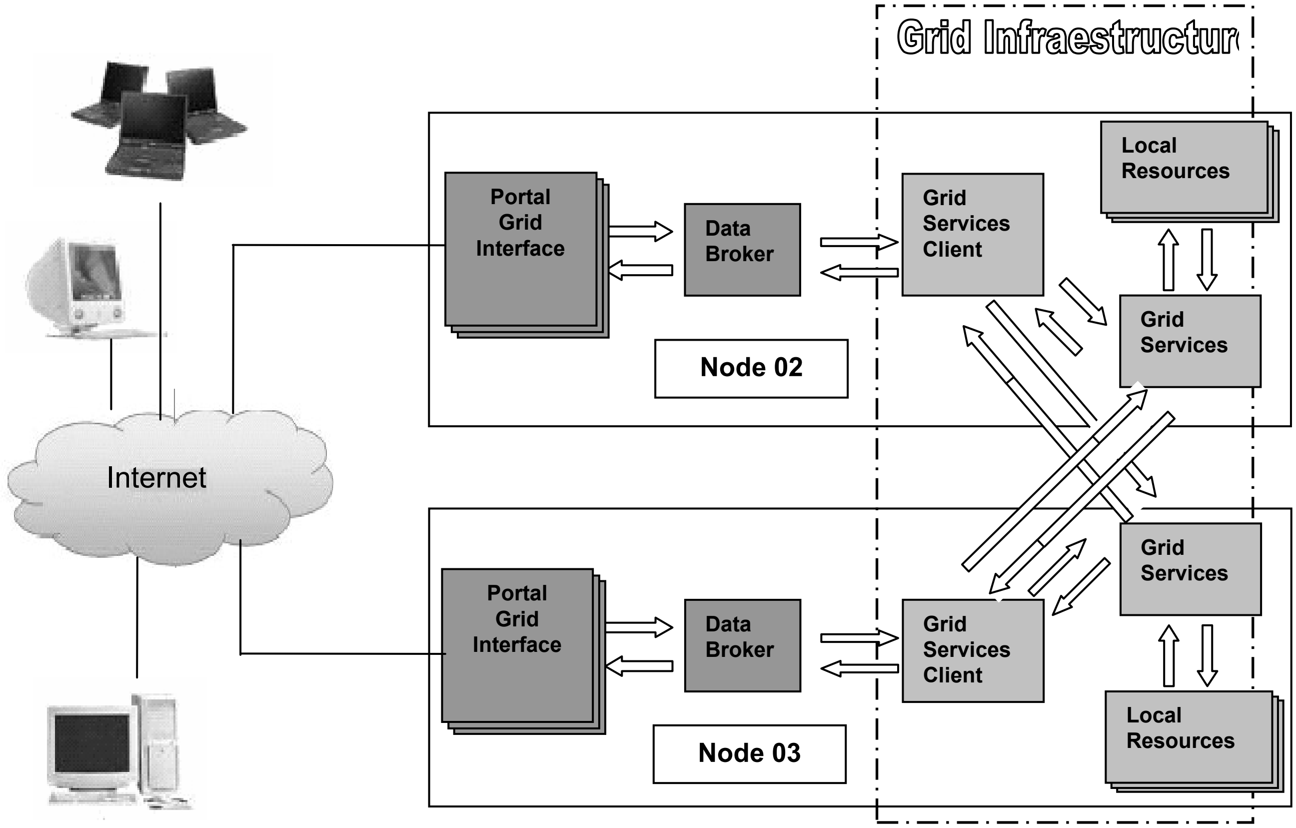

Figure 2. Grid-HSI architecture

Grid service, implemented in Grid-HSI, there is a Data Broker which assures the access from the Portal Grid Interface. The Data Broker manages data related to the Grid services available on each resource so a match between local resources and user requests is met. When the Data Broker receives a user request it seeks for node availability and selects the node with highest performance to respond the user request. The Data Broker then sends the request with information about the selected node to a Grid Service Client.

HSI Grid Services: These services implement the HSI algorithms. For each service a client stub is implemented so continuous access to Grid services will be perform through the associated client stub. Jobs are submitted by users through the Portal Grid Interface. A Master Manager Job Factory Server (MMJFS) [13], implemented in GT3, executes the task in the remote resources indicated by the Data Broker, examines results of submitted jobs, views information of resources, and so on. After the client stub receives the request from the Data Broker it proceeds to send the request to the node specified by the Data Broker. Table 1 shows the main HSI Grid Services.

Figure 2 depicts the complete Grid-HSI architecture as described above. The Grid Infrastructure includes the local resource, the HSI Grid services and the associated clients stubs. Each HSI Grid service has a Data Broker associated to provide access through a servlet implemented in the Portal Grid Interface. Figure 3 illustrates the user interface which provides transparent access to resources.

Table 1. HSI Grid Services.

Name	Description
CmeansClassifier	Generate the classification vector (.txt) using the C Means algorithm according selected parameters.
FimClassifier	Generate the classification vector (.txt) using the feedback iterative algorithm according selected parameters.
PcaReduction	Generate a new matrix of reduced dimensionality (.txt) using the Principal Component Analysis (PCA) according selected parameters.
TxtJpg	Convert a ClassificationVector.txt file, that contains the result membership for each pixel on the image, to a jpeg format file.

4. Related Work

The Parallel Computational Environment for Imaging Science (PiCEIS) [14], developed at the Pacific Northwest National Laboratory, is an image processing software designed for efficient execution on massively parallel computers. During processing and visualization an image is fully distributed in contrast to many of the current master-slave model for image processing. The user has the choices of either displaying the output to their monitors or to the IBM Scalable Graphics Engine. The heart of the communication strategy is based on the Global Array/ ARMCI Toolkit, an existing portable NUMA one-way globally addressable "shared memory" model for distributed SMP parallel computers.

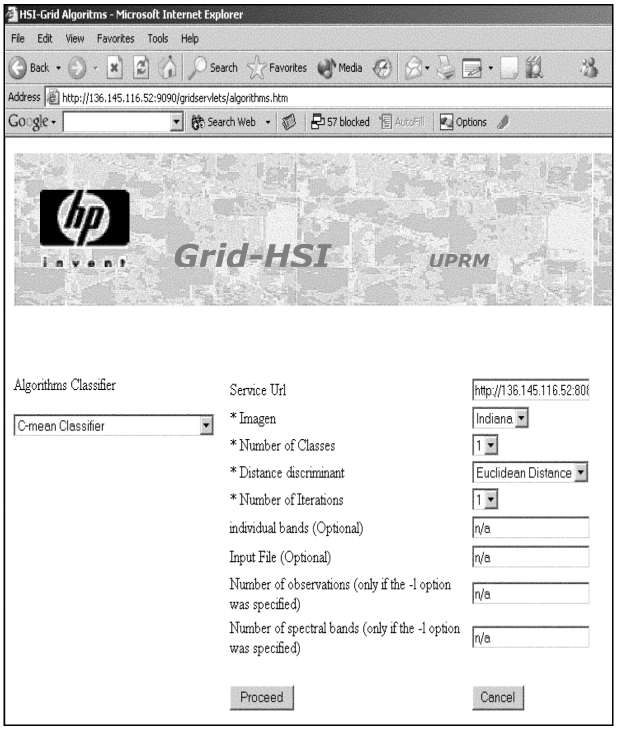

Figure 3. Interface of the application

The Global Array model extends the current models of parallel computing using shared memory, pthread and message passing style of one sided get/put model in MPI-2. While PiCEIS is focused on distributed SMP parallel computers, Grid-HSI is platform independent.

The WebDedip[15] explores object oriented modeling technique in the web domain. The WebDedip has a three-tier architecture composed by GUI, DedipServer and Agents. Java distributed object architecture is used along with the object serialization for network communication among the components. The GUI is the web enabled graphical user interface to make the entire user interaction truly system independent. It has a back-end DedipServer running on the web site. When the GUI submits the request to the DedipServer, it reads the application configuration information from the configuration file. The DedipServer initiates the execution of the first process in the interdependency chart. It informs the agent(s) on the target node to start the execution of the process. The agent sends the status information back to the DedipServer when the process is completed. While WebDesip is based on Java distributed object architecture, Grid HSI is fully based on OGSA architecture providing a functional Grid based application system.

JSIM[16], the Java-based SAR image analysis tool environment, is a tool environment for image analysis of synthetic aperture radar data. The tool environment was designed and developed using Java programming language. JSIM use the client-server approach in which multiple clients from a local or a remote machine can access the same application at any time. The tool runs on various platforms and can be used through the Internet. Similarly to WebDedip, JSIM does not provide Grid computing facilities.

5. Experiments and Analysis

For experimental purposes our local resources consists of a low cost commodity PC cluster consisting of eight nodes connected using a 100 Mbps Ethernet switch. Each node is an Intel P3-651.485Mhz with 256 MB of memory running RedHat Linux 3.2.2-5.

Successfully every local resource can accomplish the requests sent by users with not regard to the sources of the request, and users can submit jobs to several nodes at the same time. That is useful especially for FIM algorithm because the FMI Algorithm nature the execution time is large and to get the algorithm convergence can take a big time depending of selected bands. In our model, a user can run the FIM algorithm with different parameters in several nodes and wait for specific responses. The results obtained from the test cases for Grid HSI services match with those of the earlier C++ algorithms. Table 2 shows the results of C-Means method with Euclidean distance implemented using C++ language and Table 3 show results Principal Component Analysis implemented using C++ language.

Table 2. Results C-Means method with Euclidean distance.

Number of Classes	Iterations	Bands Used	Execution Time (sec)
5	4	220	43.910000
5	5	220	53.630000
5	6	220	67.920000

Table 3. Results Principal Component Analysis.

Number Components	Percent amount Energy	Bands Used	Execution Time (sec)
3	90	220	9.520000
5	90	220	9.850000
7	90	220	10.020000

Our experiments consist in to evaluate grid system accomplish running C-means and feedback iterative method classification algorithm. In order to demonstrate the effectiveness of the proposed approach many test cases were used. We present two of these simulation cases.

The first scenario the node that receives user request via internet (node02) is the same node that provided the grid service and executes the request because this node matches the available resources to the user request. How result in this scenario we get that the grid system can complete the user request. Our graphical output is depicted in figure 4.

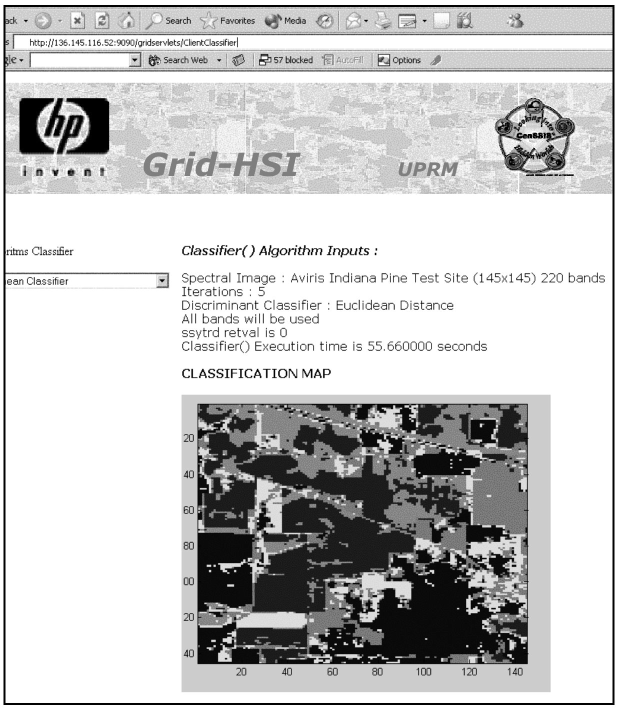

Figure 4. Results for one node scenario.

The second scenario the node that receives user request via internet (node04) is a different node (node02) that provided the grid service and executes the request because this node matches the available resources to the user request. How result in this scenario we get that the grid system can complete the user request. Our graphical output is depicted in figure 5.

6. Conclusions and Future Work

This paper focuses on the description of the architecture underlying Grid-HSI, a hyperspectral imaging analysis immersed into a grid platform which supports remote analysis and visualization. The system is based on Open Grid Service Architecture (OGSA) and the Globus Toolkit 3.0 (GT3). Most existing tolls for supporting

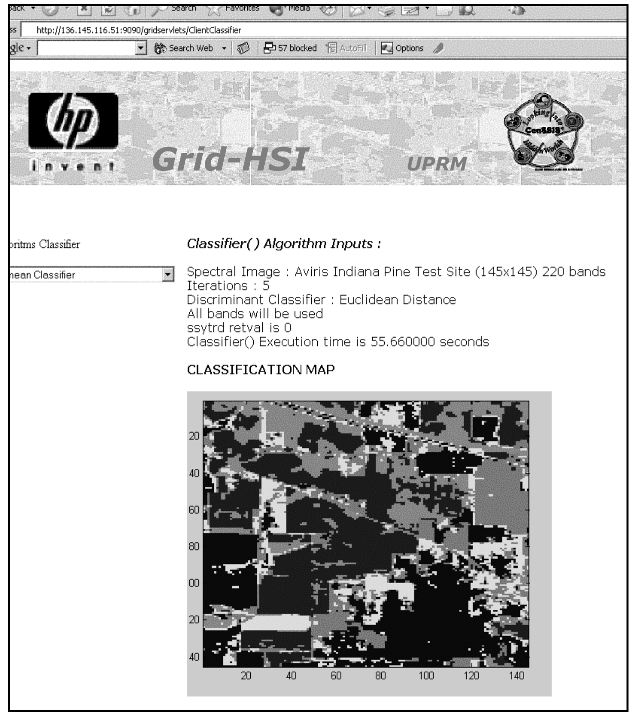

Figure 5. Results for two nodes scenario

processing of remotely sensed data use the client/server paradigm. Grid based applications, contrary to client\server approaches, provides the capabilities of persistence and potential transient process on the web. Thus, we can perform a chain of operations and we would have to get the result of one operation and send it as a parameter to the next operation.

The advantages of this Service Oriented Architecture-based Grid application also include security. The security mechanism of Grid computing is based on X.509 identity certificates, using digital signatures. It is adequately reliable so as to be more suitable for securing our Grid-HSI portal.

While the Grid-HSI system is still under development, our results suggest that the use of Grid-HSI is a new source of computational power that is accessible and applicable to the remote sensing problem set. We are currently working toward parallelization at classifications algorithms for maximization of performance on Grid platforms.

Future work includes the implementations and deployment on Grid platforms of computational methods for ensembles of classifiers and a systematic study about performance analysis and fault tolerance. This includes testing of the prototype on different scenarios, analysis of algorithms, and analysis of fault tolerance issues.

Acknowledgements

This work has been supported by the NSF Engineering Research Center for Subsurface Sensing and Imaging Systems (CenSSIS) and the Hewlett-Packard Technology Center at Puerto Rico.

References

[1] P. Swain and S. Davis, *Remote sensing: the quantitative Approach* (McGraw-Hill, 1993).
[2] S. Mazer and M. Martin, Image processing software for imaging spectrometry data analysis, *Remote Sensing of the Environment*. Vol 24 no. 1. 201-210. 1988.
[3] E. Arzuaga-Cruz., L. Jimenez-Rodriguez., D. Kaeli,, M. Velez-Reyes, E. Rodriguez-Diaz, L. Santos- Campis, C. Santiago, H. Velazques and Alexey Castrodad, A MATLAB Toolbox for Hyperspectral Image Analysis, *Proc IEEE International Geoscience and Remote Sensing Symposium*, Alaska, 2004.
[4] D. Rivera-Gallego, Feedback in Pattern Recognition, Master in Science Thesis, University of Puerto Rico, Mayaguez, 2002.
[5] W. Lugo-Beauchamp, K. Cruz, C. Carvajal-Jimenez, and W. Rivera, Hyperspectral Imaging Analysis using Itanium architecture. To be submitted.
[6] I. Foster and C. Kesselman, *The grid: blueprint for a future computing infrastructure* (Morgan Kaufmann Publishers, 1998).
[7] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, The Phisiology of the Grid: An Open Grid Serivices Architecture for Distributed Systems Integration, *Technical report, Open Grid Service Infrastructure WG, Global Grid Forum*, june 2002.
[8] I. Foster and C. Kesselman, Globus: A Metacomputing Infrastructure Toolkit, *International Journal of Supercomputer Applications, 11*(2): 115-128, 1997.
[9] Word wide web Consortium: http://www.w3.org
[10] S. Tuecke, K.Czajkowski, I. Foster, J. Frey, S. Graham, and C. Kesselman, Grid Services Specification, 2002, Available at http://www.globus.org/research/papers/gsspec.pdf.
[11] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maquire, T. Sandholm, D. Snelling, and P. Vanderbilt, Open Grid Services Infrastructure (OGSI) Version 1.0, 2003. avaialable at http://www.ggf.org/ogsi-wg
[12] WSRF: http://www.globus.org/wsrf/
[13] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman and S. Tuecke, Security for Grid Services, *Proc Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*, IEEE Press, to appear june 2003.
[14] G. Fann, D. Jones. E. Jurrus, B. Moon, and K. Perrine, “A Parallel Computational Environment for Imaging Science,” In Press, *Proc of the 47th International Symposium on Optical Science and Technology*, july 2002.
[15] H. Bhatt , V. Patel and A. Aggarwal, Web enabled client-server model for development environment of distributed image processing, *Proc of the 7th International Conference on High Performance Computing*, december 2000.
[16] C. Huallparimachi, Design and Development of a Java-based Distributed System Tool for Synthetic Aperture Radar Image Analysis, *Proc IASTED International Conference on Computer Science and Technology, CST 2003*, Cancun, may 2003.