INTERNATIONAL JOURNAL of COMPUTERS
Issue 2, Vol. 1, 2007

Land-Cover Classification on Computational Grids

Dana Petcu[1], Silviu Panica[2], Andrei Eckstein[3]

***Abstract*—Satellite image processing is a very demanding procedure in terms of data manipulation and computing power. Grid computing is a possible solution when the required computing performance or data sharing is not available at the user's site. The paper discusses a possible scenario of using Computational Grids. According to this scenario a prototype code for satellite image classification was designed, implemented and tested in two different virtual organizations. The approach can be applied also to other image processing procedures.**

***Keywords*—Computational Grids, Image processing, Remote sensing**

I. Introduction

There are at least three reasons for using Grid computing for satellite image processing: (a) the required computing performance is not available locally, the solution being the remote computing; (b) the required computing performance is not available in one location, the solution being cooperative computing; (c) the required computing services are only available in specialized centres, the solution being application specific computing. In this paper we address mainly the first issue.

Grids are classified in different types according to their main functionalities. In principle, most people distinguish between pure Computational Grids and the more enhanced Data Grids. Other classifications were recently surveyed in [22].

A Computational Grid focuses mainly on computationally-intensive operations. A Data Grid deals with the controlled sharing and management of large amounts of distributed data. An Instrument Grid is a special type of Grid which has a primary piece of equipment (e.g. a satellite), and where the surrounding Grid is used to control the equipment remotely and to analyze the data produced.

Since remote sensing image processing is both data and computing demanding, and involves special instruments, a Grid infrastructure for satellite image processing can be considered also as a Computational Grid as well as a Data Grid or an Instrument Grid. A new Grid type, named Service Grid, is currently gaining a significant place among the current production Grids. While the Computational Grid was designed for high performance computing at Web scale, the Service[1] Grid reflects the recent evolution towards a Grid system[2] architecture based on Web services concepts and technologies. The Service Grids' potential for remote sensing has already been pointed out at the beginning of this evolution, for example in [6]. More concrete initiatives in this context are mentioned in Section II of this paper.

A special Grid infrastructure, was recently built within a Romanian research project, called Medio-Grid. Its architecture was described in [14]. A software platform aiming to process the data acquired from meteorological satellites by using Grid computing technology is under construction. The main application that is envisioned is dealing with the prediction of water floods and forest fires. The infrastructure is based on the distributed hardware resources of academic service providers (for image acquisition, storage and processing) from three Romanian centers. The service requirements are defined with the help of the specialists from the National Agency of Meteorology. In particular, high performance computing requirements that cannot be covered by only one computing center are of special interest for the project members. MedioGrid has been designed to rely mainly on a Service Grid. Open standards like WSRF or SOAP are used in conjunction with Globus Toolkit 4. Several specific WSRF services are already available, related to the detection of river beds [17], clouds [16] and vegetation [2]; issues related to MedioGrid seen as a Data Grid were treated in [4].

In what follows we discuss the potential of using MedioGrid infrastructure as a Computational Grid. The paper is organized as follows. The next section discusses the state-of-the-art in the satellite image processing on Computational and Service Grids. Section III describes a scenario for using the Computational Grid, while Section IV gives a concrete example.

II. State-of-the-art in Grid Processing Satellite Images

Image processing applications require not only the

Manuscript received 17 February, 2007. This work was partially supported by the Grants 19CEEX-I03 MedioGrid and CEEX-II-5919 GRAI of the Romanian Ministry of Research. The tests were performed on the infrastructure provided by the European project SEE-Grid.

[1] D. Petcu – Western University of Timisoara, 4th V. Parvan, 300223 Timisoara, Romania (email: petcu@info.uvt.ro)

[2] S. Panica – Western University of Timisoara, 4th V. Parvan, 300223 Timisoara, Romania (email: silviu@info.uvt.ro)

[3] A. Eckstein – Western University of Timisoara, 4th V. Parvan, 300223 Timisoara, Romania (email: eckstein@info.uvt.ro) . Revised received June 1,07

processing of large volumes of data, but also various types of resources, and it is not reasonable to assume the availability of all resources on a single system. An early paper [12] describes a metacomputing application that integrates specialized resources, high-speed networks, parallel computers, and virtual reality display technology to process satellite imagery. The inputs of the near-real-time cloud detection code are two-dimensional infrared and visible light images from satellite sensors. Later on, the paper [15] proposes a distributed system that connects users, data bases and method bases; users are helped to find an appropriate sequence of methods for processing, and, using a broker, they schedule execution onto fast remote processing units.

The authors of [5] developed a framework for the remote access and manipulation of large georeferenced images. A dataflow visual programming language allows users to develop programs out of chains of image operators. TerraShare [20] is a modular, client-server system designed to address the image management and distribution needs of geo-imaging producers and distributors.

In the context of increasingly larger amounts of data due to advances in sensor and storage technology, Grid computing promises to enable image analysis of a large amount of data of computationally expensive operations on disparate data and heterogeneous clusters of computers. The paper [11] presents a case study that illustrates how to migrate the traditional image process applications to Grid infrastructures for different roles of image processing.

An experiment aiming to demonstrate the use of Grid technology for remote sensing applications has been carried out in the frame of the European projects DataGrid [7] and MediGrid [10]. Other famous Grid projects focused on spatial information are SpaceGrid, EarthObservation Grid, GODIS, or GENESIS. Middleware for remote sensing image processing on Grid platform has been developed and reported also in [26] (based on Globus Toolkit, OpenPBS, and Condor-G). The goal of IP4G [9] (Image Processing For The Grid) is to create a Web service interface to a Grid infrastructure in which image processing can be used across multi-institutional resources. Specialized tools are used as distributed workflow system, visualization toolkit, and as segmentation and registration toolkit.

The paper [25] presents a programming model, called PAGIS, that provides an interface to introduce and customize Grid functionality. The prototype implementations focus on the domain of satellite image processing for geographical information systems; exemplified operations include geo-rectification and scaling the data. The paper [1] presents an architecture that allows access to remote supercomputing facilities from a Web gateway; the implementation exploits the Globus toolkit; the utility of the approach is proved in the context of using an active digital library of remote sensing digital data.

There are several computationally demanding algorithms that have been Grid-enabled in the last years. For example, the paper [23] focuses on the parallelization of satellite image geo-rectification on an Alice-based Grid. The paper [3] outlines the design and implementation of Grid-HSI, a service-oriented architecture-based Grid application to enable remote hyperspectral image analysis. The paper [24] discusses a remote sensing image classification algorithm. The aim of the Grid middleware is to divide jobs into several assignments and submit them to the computing pool.

III. Scenarios for the Use of Grids in Satellite Image Processing

In [17, 18] we have considered two scenarios of using the MedioGrid infrastructure, both pointing towards the necessity of Web-based Grid services. We give a short description of these two scenarios before presenting another possible scenario that is pointing towards the necessity of Computational Grid services.

Scenario 1: Remote Processing of Satellite Images [17]. A distributed database of large satellite images is available to a virtual organization (VO) of several research or academic institutions and companies as a shared facility. A user wants to apply a sequence of image processing standard procedures to extract particular information from specific images. He has access to a workstation and owns a Grid certificate issued by the virtual organization. The user's workstation has reduced computing facilities and no license for a commercial image processing tool. The user has an advanced knowledge about the operations to be performed on the images and low or medium programming knowledge. The higher the number of images to be simultaneously analyzed or the image dimensions, the higher the CPU's time to extract different features from the images. The user will quickly realize that the available computing and storing resources are not allowing the local processing of the satellite images.

The infrastructure supporting this scenario should consist in the following components: the client's codes and minimal facilities to access Grid infrastructure, at the user's node; the service codes and the catalog services that index Grid services and satellite data at the repository nodes; the satellite data and image processing software at the storage and computing nodes.

In order to respond to this scenario, a prototype Grid service and a Web client were built using Globus Toolkit 4 as Grid middleware and GIMP as image processing tool. As mentioned above, it was assumed that GIMP was available at the nodes where the satellite data are residing.

Scenario 2: Remote Parallel Processing of Satellite Images [18]. The conditions and requirements are the same as in Scenario 1, but, in addition, neither the local nor the remotely accessible computing nodes can individually do the task in a reasonable time. It is assumed that several resources of the virtual organization can be harnessed through a Grid service to deal with the computationally demanding task.

The proposed infrastructure is slightly modified. The computing nodes from the same institution, forming a cluster,

have fast access to the referenced images. One site's computing nodes are represented by a head node that runs the Grid services. The head-nodes are grouped together and visible through the service index of the VO. It is also assumed that replicas of the images to be processed are available on each head node, and through them, they are available to each computing node from the same institution.

In order to respond to this scenario, a new prototype Grid service and a Java client were build. The user selects the remote images that will be processed and specifies the action to be undertaken, in GIMP language. The classical 'farming model' from parallel processing theory is applied. The farmer is the Grid service that splits the work between several workers. The images are split into a number (provided by the user) of smaller images (sub-images). The number of the workers is equal to the number of sub-images. The worker's Java classes are available at the head nodes. The list of head nodes is available and the user selects the cluster(s) where the tasks will be launched. The workers are running on the head nodes. Each worker launches a remote GIMP server on the local cluster and acts as a dispatcher for the incoming commands and as GIMP a client.

Remarks. The two scenarios have in common the idea that processing is moved to the place where the images are available due to the fact that satellite images are huge and their transfer is time expensive. The practical solutions have assumed that GIMP is available at the data sites. This requirement can be imposed on small Grids, especially in the one dedicated to image processing, like MedioGrid, but not in general-purpose world-spread Grids. The following scenario denies the existence of the same image processing tool at each Grid site. The second scenario assumes that the Grid is a Service Grid that has several facilities for high performance computing. This is the case for MedioGrid which includes three departmental clusters. The prototype Grid service built to answer the second scenario is site-dependent since it must be aware of the local resources and their availability to run the GIMP operations. The following scenario assumes platform dependence.

Scenario 3: Grid Processing of Satellite Images. A member of a virtual organization (VO) of several research or academic institutions and companies accesses a local database of large satellite images. The user wants to apply a special procedure for image processing in order to extract a particular information from the selected images. He has access to a workstation that has reduced computing facilities. The user has advanced knowledge about the operations to be performed and implements them in a special programme. The higher the number of the images to be simultaneously analyzed or the dimensions of the images, the higher will be the CPU's time to extract different features from the images. The user will quickly realize that the available computing and storing resources are not allowing the local processing of the satellite images.

The infrastructure supporting this scenario should have the following components: the satellite images, the client's codes and some minimal facilities to access Grid infrastructure, at the user's node; the Grid middleware which allows the execution of client codes on client's data at remote computing nodes.

The infrastructure mentioned above is specific for a Computational Grid.

The user should rewrite the code. The special procedure for satellite image processing should be split into tasks that can be performed in parallel and that are requiring similar computing effort. Ideally these tasks should be independent. If this is not possible, techniques like message passing between these tasks should be taken into account. The full image transfer should be avoided since the satellite images are big.
Therefore the splitting into tasks acting on small parts of the images is preferred. The number of the tasks and image parts should be careful selected by theoretically performance studies or experiments, in order to obtain a response in a reasonable time.

Variants of this scenario have already been considered also in previous works mentioned in Section 2, like in [23, 24]. In our case it is assumed that the user has direct access to the Grid node that stores the satellite images.

In the following section we present a simple example subscribing to this scenario and the prototype that was build as demo facility in the frame of MedioGrid project.

IV. Case Study

There are several repositories of satellite images. For example the Global Land Cover Facility (GLCF [8]) hosts an image archive that has Landsat TM images available for most of the world for free download. With a multispectral sensor, such as those from the Landsat satellites, information from different wavelengths of light (visible, infrared, thermal, etc.) is collected. The information for each specific wavelength range is stored as a separate image (a band). The dimension of one image varies between 64 Mb and 1 Gb. The number of bands can vary between 7 and more than 200 (multi or hyperspectral imagery). A classical operation is to combine the images from the different wavelengths (gray images) to create a color image. More details about the fundamental remote sensing concepts, image interpretation and classification are presented in guides like [19].

MODIS imagery (MODerate Resolution Imaging Spectroradiometer [13]) is used in MedioGrid. MODIS has 36 observational channels covering a wide frequency spectrum from visible to infra-red radiation, optimized for visualizing specific surface and atmospheric features. Each MODIS data has an associated metadata file, describing characteristics such as spatial resolution, image size, spatial extent, timeframe, satellite characteristics. MedioGrid project aims to create a cache with the most recent data for Romania and surrounding areas which can then be processed by using Grid computing resources.

Tests for this work where performed on images related to the West side of Romanian that are provided by GLCF and include MODIS images.

The case study refers to the implementation of a simple classification algorithm based on the binary decision tree proposed by [19] to classify forest and non-forest classes. The classification process involves translating the pixel values in a satellite image into meaningful categories. Decision trees are a set of binary rules that define how specific land-cover classes should be assigned to individual pixels. The inputs are two MODIS bands (the red and the infrared ones). The detected land-cover classes include: water, cloud, non-forest, forest, and scrub. To each class it is assigned a different color in the result image (Figure 1).

Since the classification algorithm is applied on the pixel level, the splitting of the computational effort into similar tasks acting on parts of the image is straightforward. The bands are split into equal subimages.

Assuming that the user has a sequential code that applies the decision tree classification on satellite images, the code prepared for the Computational Grid consists of three components:

1. the splitter that takes the two bands and splits them into a number of sub-images;
2. the classifier that receiving two images (pieces from the red and infrared bands) applies the binary decision tree and produces the sub-image storing for each pixel the color of the associated land cover class;
3. the composer that gathers the colored subimages.

The classifier is the initial sequential code that is applied to smaller images.

While the splitter and the composer are acting only at the user's site where the satellite images are residing, the classifier and the sub-images are submitted for processing on the Computational Grid.

The prototype codes were written in Java and their efficiency was studied on the computational Grid provided by the SEE-Grid infrastructure [21]. The test codes are available at http://web.info.uvt.ro/~petcu/mediogrid/ClassDecTree.zip.

Parameter studies were performed to detect the best choice of the number of tasks (sub-images) depending on the image's dimension. The results show that in the case of a small satellite image (old MODIS images for which a band is around 64 Mb) a small number of tasks (e.g. four) is sufficient to obtain a reasonable response time using the Grid infrastructure instead only one computer, while for larger images (that can hardly be loaded into one computer memory, e.g. for 450 Mb) a larger number of tasks (e.g. sixteen) should be consider, motivating the effort to write the splitter and the composer. Table 1 illustrates the response time reduction by using the computational Grid for the classification of smaller images: the bands of only 5 Mb are split into 4, 9, respectively 12 pieces and processed independently; due to the small dimension of the bands, the response time is of the order of few seconds and

TABLE I
RESPONSE TIME IN SECONDS FOR THE CLASSIFICATION APPLIED TO SMALL IMAGES

Split type	Nodes No	Response time	Image file size in bytes
1x1	1	5,660	5,324,604
2x2	4	2,230	1,331,018
3x3	9	1,630	591,947
4x3	12	1,498	443,758

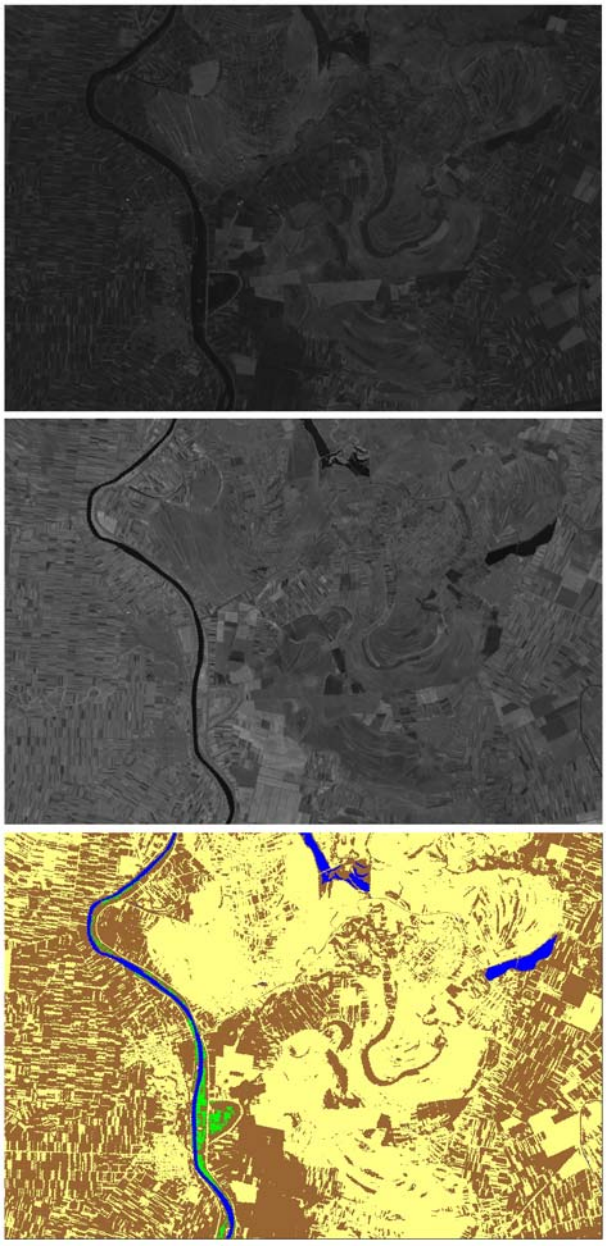

Figure 1: An implementation of a decision tree classification applied to the Bands 3 (top) and 4 (middle) of a MODIS (sub)image of the West part of Romania downloaded from the GLCF archive produces a color image (bottom) in which each color has a specific meaning

the improvement of the response time is not spectacular.

Unfortunately, due to the SEE-Grid security solutions of using sandbox that are restricting the file dimensions to 10 Mb we cannot perform more tests in such Grid environment.

In order to integrate developed codes into the MedioGrid infrastructure that uses Globus Toolkit 4 and no restrictions on file sizes, a Web interface for a service that launch the discussed codes has been build. It is located at *http://ui01.info.uvt.ro:8080/uGOC/* and can be used by anyone for testing purposes upon request addressed to the paper authors to obtain a user account. The interface is build using JSP, Tomcat/5.5 and MySQL.

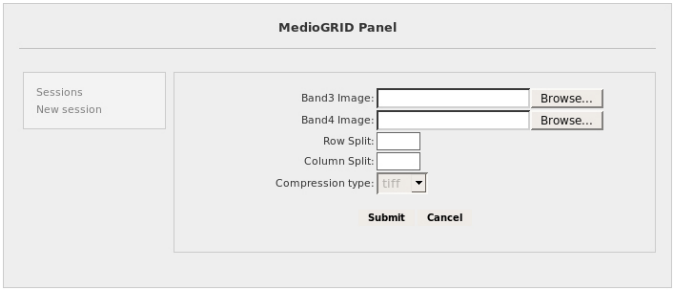

Figure 2: User interface for uploading the image files and selecting the splitting parameters

The user can upload the image files and selects the splitting parameters (Figure 2). After the job registration and launch request the control panel allows the visualization of the status and the user actions like canceling the job. After the successful finish of the jobs (Figure 3), the user can download the resulting images.

The service works as follows:

1. Image upload and registration of a new session;
2. Transfer of the files to the code site (denoted here SE);
3. Upon the user request of lunching the job, the splitting code is called and and the smaller pieces of images are generated as well as the files needed by PBS to launch the classifier on each peer of small images;
4. The JobManager of Globus Toolkit 4 take over the files and interpret them and finally PBS sends the jobs on the cluster of workstations;
5. After the job executions the output file are stored on SE;
6. The user can access the output files through the user interface.

The above steps are marked by numbers in Figure 4.

The second testing architecture is known to be a PC cluster with 28 processors on 7 chips: 7 HP Pro-Liant DL-385 with 2 x CPU AMD Opteron 2.4 GHz, dual core, 1 MB L2 cache per core, 4 GB DDRAM, 2 network cards 1 Gb/s. The results obtained using this cluster and medium size images of 64 Mb are more encouraging: a speedup of 53 % when 9 processors are used instead 39% in the case of smaller images.

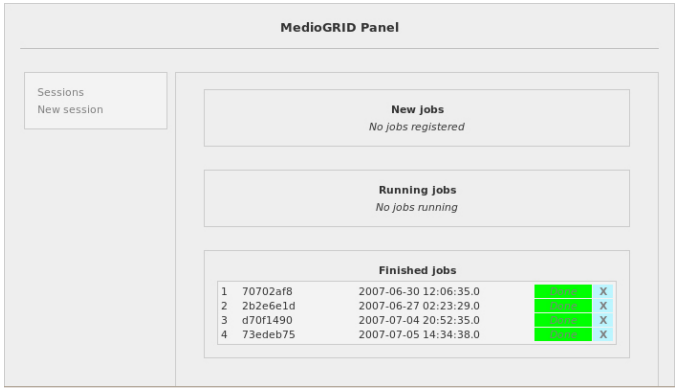

Figure 3: User interface controlling the job status

Similar codes can be built for more complex satellite image processing, such as supervised or unsupervised classifications.

TABLE II
RESPONSE TIME IN SECONDS FOR THE CLASSIFICATION APPLIED TO MEDIUM SIZE IMAGES

Split type	Nodes No	Response time	Image file size in bytes
1x1	1	23,367	62,396,000
2x2	4	9,608	15,601,000
3x3	9	4,919	7,099,426
4x3	12	4,537	5,324,604

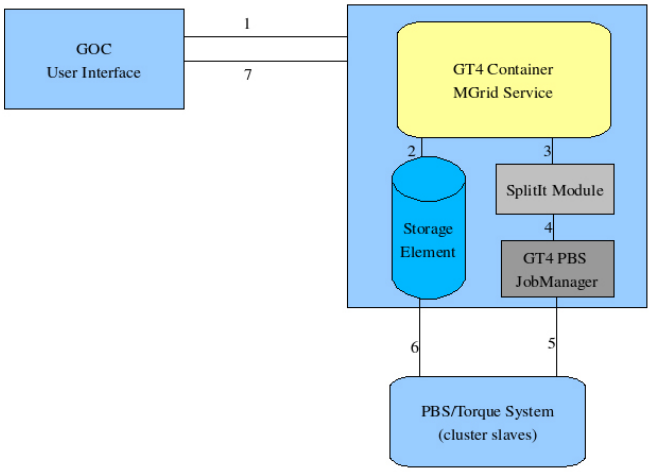

Figure 4: The components of the service and their interaction with the system components and the user interface

V. CONCLUSIONS

Current Grid technologies provide powerful tools for remote sensing data sharing and processing. After an overview of the recent initiatives of 'gridifying' satellite image processing, a specific usage scenario in which the Grid is conceived as a powerful computing resource was analyzed. A code which is user property for satellite images is split into smaller tasks that are submitted to the Grid middleware for

processing. The software prototype was built as concept demonstrator for the usage scenario. It is only a small piece of the framework of the MedioGrid on-line system that will operate on MODIS satellite images in at most one year.

Acknowledgements: The research was partially supported by the Grants 19CEEX-I03 MedioGrid and CEEX-II-5919 GRAI of the Romanian Ministry of Research. The tests were performed on the infrastructure provided by the European project SEE-Grid.

REFERENCES

[1] G. Aloisio and M. Cafaro, A Dynamic Earth Observation System, Parallel Computing 29(10), 2003, pp. 1357–1362.
[2] V. Bacu, O. Muresan and D. Gorgan, MODIS Image Based Computation of Vegetation Indices in MedioGRID Architecture, Procs. ISPDC 2006, IEEE Computer Society Press, Los Alamitos, 2006, pp. 267–273.
[3] C.L. Carvajal-Jimenez, W. Lugo-Beauchamp and W. Rivera, Grid-HSI: Using Grid computing to Enable Hyperspectral Imaging Analysis, Procs. CIIT04, 2004, pp. 583–588.
[4] A. Colesa, I. Ignat and R. Opris, Providing High Data Availability in MedioGRID, Procs. ISPDC 2006, IEEE Computer Society Press, Los Alamitos, 2006, pp. 296-302.
[5] S.J. Del Fabbro, Developing a Distributed Image Processing and Management Framework, Honours Degree Thesis, University of Adelaide, 2000.
[6] G. Fox, S. Pallickara, G. Aydin and M. Pierce, Messaging in Web Service Grid with Applications to Geographical Information Systems, In Grid Computing: A New Frontier of High Performance Computing, ed. L. Grandinetti, Elsevier, Amsterdam, 2005, pp. 305–331.
[7] N.A. Giovanni, F.B. Luigi and J. Linford, Grid Technology for the Storage and Processing of Remote Sensing Data: Description of an Application, SPIE 4881, 2003, pp. 677–685.
[8] Global Land Cover Facility, http://landcover.org/.
[9] S. Hastings, T. Kurc, S. Langella, U. Catalyurek, T. Pan and J. Saltz, Image Processing for the Grid: A toolkit for Building Grid-enabled Image Processing Applications, Procs. CCGrid03, IEEE Computer Society Press, 2003, pp. 36–43.
[10] L. Hluchy, O. Habala, M. Laclavik, Z. Balogh and E. Gatial, Knowledge-based Platform for based Platform for Environmental Risk Management Environmental Risk Management, Procs. ISPDC 2007, IEEE Computer Society Press, 2007, pp. 1–9.
[11] H. Jin, X. Shi and L. Qi, Use Case Study of Grid computing with CGSP, LNCS 3597, 2005, pp. 94-103.
[12] C.A. Lee, C. Kesselman, S. Schwab, Near-realtime Satellite Image Processing: Metacomputing in CC++, IEEE Computer Graphics & Applications, 16(4), 1996, pp. 79–84.
[13] Moderate Resolution Imaging Spectroradiometer, http://modis.gsfc.nasa.gov/
[14] O. Muresan, F. Pop, D. Gorgan and V. Cristea, Satellite Image Processing Applications in MedioGRID, Procs. ISPDC 2006, IEEE Computer Society Press, Los Alamitos, 2006, pp. 253–262.
[15] F. Niederl and A. Goller, Method Execution on a Distributed Image Processing Back-end, Procs. 6th PDP, 1998, pp. 243–249.
[16] F. Parauan, M. Ordean, A. Diamandi, Clouds Mask Algorithm, Procs. SYNASC 2006, IEEE Computer Society Press, Los Alamitos, 2006, pp. 259-266
[17] D. Petcu and V. Iordan, Grid Service based on GIMP for Processing Remote Sensing Images, Procs. SYNASC 2006, IEEE Computer Society Press, Los Alamitos, 2006, pp. 251–258
[18] D. Petcu, Grid Services for Satellite Image Processing, WSEAS Transactions on Computers, Issue 2, Vol. 6, 2007, pp. 347-354.
[19] Remote Sensing Guide, http://cbc.rsgis.amnh.org/remote sensing/guides/intro.html
[20] H. Rosengarten, TerraShare: Distributed Image Data Management. Protogrammetric Week, D. Fritsch, R. Spiller (eds.),Wichmann Verlag, Heidelberg, 2001.
[21] South Eastern European Grid – enabled eInfrastructure Development, http://www.see-grid.eu.
[22] H. Stockinger, Defining the Grid: A Snapshot on the Current View, Journal of Supercomputing, 2007, in print.
[23] Y.M. Teo, S.C. Tay and J.P. Gozali, Distributed Geo-rectification of Satellite Images using Grid Computing, Procs. 17th IPDPS, IEEE Computer Society Press, 2003.
[24] J. Wang, X. Sun, Y. Xue, Y. Hu, Y. Luo, Y. Wang, S. Zhong, A. Zhang, J. Tang and G. Cai, Preliminary Study on Unsupervised Classification of Remotely Sensed Images on the Grid, LNCS 3039, 2004, pp. 981-988.
[25] D.Webb and A.L.Wendelborn, The PAGIS Grid Application Environment, LNCS 2659, 2003, pp. 1113–1122.
[26] C. Yang, D. Guo, Y. Ren, X. Luo, J. Men, The Architecture of SIG Computing Environment and Its Application to Image Processing, LNCS 3795, 2005, pp. 566-572.