Content-Based Retrieval of Remote Sensing Images On the Grid Platform

I. Guarino and A. Guerriero*
*Politecnico di Bari
Via Re David 200, 70121 Bari
Italy

Abstract: - Huge volumes of data are received every day from remote sensing sensors and the data amounts will increase rapidly following the technologies improvements. Traditional database cannot be used to store and manage these data efficiently. Every National Agency or department manage its own database independently forcing users to search data on repositories geographically distributed, moreover information extraction from these data requires powerful computing resources, but high performance resources are not anywhere. The high potentiality of Grid technology in integration of distributed and heterogeneous computational and storage resources provides new ways to solve these problems. This paper reports the results of a research project FIRB “Grid.it” on a retrieval system for remotely sensed images on GRID platform. A prototype of a retrieval system, developed by a new structured program languages (Assist 1.2), is presented and discussed.

Key-Words: Grid, Globus, ASSIST, Skeleton, Remote sensed Image Database, Sorting, Image Features.

1 Introduction

The increasing availability of remote sensed images with high spatial, spectral, radiometric and temporal resolution, due to the launching of advanced commercial satellites and satellites constellations, has led to the problem of managing huge data volumes. Information is stored in heterogeneous Earth Observation (EO) repositories, geographically distributed, with slow access and often with no or little information about the information content. A number of operational remotely sensed imagery database management systems is available 1-6. Though advanced image compression technology has solved part of the storage problem, the lack of shared effective semantic description metadata makes searching and locating through such a database a difficult task. Therefore, an automatic retrieval method is essential for people who want to locate their desired images exactly.

In the 90's Content-based Image Retrieval (CBIR) 7 has gained increasing popularity among researchers. It is generally agreed that retrieval based on image content is more useful in some applications. People can adopt color, shape and texture as features to retrieve similar images. However, the way of retrieving the content of an image efficiently and effectively still lacks of common recognition. This is because the low level features of an image including color, shape, texture, etc., which could be easily analyzed do not coincide with the high level concepts of an image.

To face these problems, we proposed a compound signature to clusterize images by geomorphologic features [1].

Another major problem in the practical implementation of a Content-based Image Retrieval (CBIR) for remotely sensed images is that the content-based indexing and searching process always requires extremely high computational power. On the other hand, the content-based image retrieval algorithms are very suitable for parallel computation as the algorithms can be broken into several data independent processes for running on a parallel computer.

In this paper, a Grid platform and a new structured programming languages (Assist 1.2) are used to solve
the high computing demanding power of a remote sensed CBIR system which, otherwise, would have requested powerful and very expensive massively parallel machines in order to achieve high performances in terms of throughput and response time for a real time or near real time applications. This paper is organized as follows. Section 2 describes the background and related works, section 3 outlines the main characteristics of the Grid platform and software tools utilized, section 4 gives an overview to image features and retrieval system realized. Testing procedures are presented in section 5. Finally, section 6 concludes the paper discussing the prototype performances.

2 Background and related works

Grid is designed to share resource on internet allowing performing computationally intensive tasks on computational resources that are temporarily made available on the Grid by members of a “virtual organization”. In a “virtual organization” all members contribute their limited resources for the benefits of all partners. This makes Grid useful for collaboration between geographically wide-spread partners who can use all computational resources available on the Grid during their work hours while other partners do not work.

Based on Grid technology new remote sensing databases systems have been proposed in literature, challenging the retrieval on huge amount of distributed multiresolution, multispectrum, multisensor images. Dong and Hu [2] discussing the features of remote sensing data and requirement of a remote sensing database on Grid platform, highlight the necessity of an appropriate model to express the image characteristic so that the image can be easily searched by content. The rationality of content based metadata determines the query efficiency of the database system.

Some works to solve one of the main obstacle to realize efficient retrieval, the lack of effective semantic description methods, propose ontology based approaches. Ontology being a formal specification of a domain, can be usefully used to exchange knowledge between representations and applications. Durbha et al. [3], provided a framework on Landsat images and Anderson classification. This system uses only three interrelationship, limiting the reasoning capacity of the inference engine. Visser te al. [4] describe three geographical information ontologies, but their shared vocabularies do not support the Remote Sensing Image System. Sun et al. [5] propose RSI2Grid a prototype for semantic retrieval by ontology and grid technologies. To manage heterogeneous information representation of remote sensed images, RSI2Grid adopts a predefined vocabulary and the Resource Description Framework (RDF) as ontology description language to obtain semantic annotations from images metadata descriptions.

Zhu et al. [6] propose and implement a system architecture of RS image information services based on three layers: resource layer, service layer and application layer, with data management based on metadata described by GML (Geography Markup Language).

In these systems, the way of retrieving the content of an image efficiently and effectively depends on annotation on metadata or keyword rationality. In [1] we have proposed a remote sensed images retrieval system, based on low level image features, able to clusterize images by geomorphologic features.

A major problem in the practical implementation of a remotely sensed CBIR system, is that the indexing and searching process always requires extremely high computational power, even the fastest computer still requires a significant amount of time to search a medium sized database. On the other hand, the content-based image retrieval algorithms are very suitable for parallel computation since the algorithms can be broken into several data independent processes for running on a parallel computer.

In this paper, we show the results of our activity in the FIRB Strategic Project Grid.it (Enabling Platform for High-performance Computational Grids Oriented to Scalable Virtual Organizations) to port and test a prototype of that remote sensed image retrieval system on a Grid platform using a new programming environment (ASSIST 1.2) oriented to the development of high-performance Grid applications.

3 Grid platform and Parallel programming tools

In 2002 the Italian Spatial Agency (ASI) supported the ASI-PQE2000 Project on High-Performance Systems and Tools for Earth Observation Applications. In that project we ported a remote sensed image indexing application on MPI based cluster architectures. To reduce the load balancing problem, in MPI directly managed by the programmer, in the same project the first version of the high-performance programming environment, ASSIST, for parallel and distributed portable applications was released. In a following 2004 MIUR (Legge 449/97) project, “High-performance
Large-scale Distributed Platform”, important results have been achieved in programming tools and environments (the realization of the first Grid enabled version of ASSIST), knowledge-based and data-mining grid-aware tools, as well as in applications. Finally, in the ongoing FIRB Strategic Project Grid.it (Enabling Platforms for Highperformance Computational Grids Oriented to Scalable Virtual Organizations) in the area of Enabling Technologies for Information Society, we realized a remote sensed CBIR prototype on Grid technology.

The ASI-Grid realized in the FIRB project is composed by Computing and Storage Elements (CE and SE) distributed in the south of Italy. The main SE is located in the ASI facilities of Matera, other smaller SE are present in all the other Node of Bari and Taranto. The Taranto Node is organized in a cluster of heterogeneous workstations and servers, 20 Pentium processor based PC and 2 supermicro© multiprocessors servers. The Bari and Matera nodes are composed by other 4 workstations and 2 multiprocessors servers. Fig. 1 shows the nodes localization and Fig. 2 shows a connection scheme to the Taranto Node.

Fedora1 or Red Hat 9, the Globus Toolkit 3.2.1 and Assist 1.2 are the O.S. and the software environment of the ASI-Grid. The open source Globus Toolkit ® is an enabling technology for the Grid, that allows to share computing power, databases, and other tools securely online across institutional and geographic boundaries. It includes software for security, information infrastructure, resource management, data management, communication, fault detection, and portability. It is packaged as a set of components that can be used either independently or together to develop applications [7].

ASSIST (A Software development System based upon Integrated Skeleton Technology) [8-12] is a programming environment oriented to the development of parallel, distributed, high-performance, Grid-aware applications according to a unified approach. It provides the application programmer with a highly layered programming environment supporting a high level programming model [13]. The ASSIST programming environment is implemented using a structure such as the one depicted in Fig. 2:

- The compiler layer performs static optimisations and takes care of generating the object code of the component assembly representing the application and taking heterogeneity into account.
- The run time layer is responsible of supporting the execution of the object code and it accounts for dynamic optimisations and Grid targeting.
- The middleware /operating system level provides all the services needed to implement the upper layer policies.

Fig. 3 Layered grid application development
ASSIST applications are described by means of a coordination language, which can express arbitrary graphs of modules, interconnected by typed streams of data. Modules can be either sequential or parallel. A sequential module wraps a sequential function. A parallel module (parmod) can be used to describe the parallel execution of a number of sequential functions, that are activated and run as Virtual Processes on items arrival onto input streams. The sequential functions can be programmed by using a standard sequential language (C, C++, Fortran). Virtual Processes may synchronize with one other by means barriers. Overall, a parmod may behave in a data-parallel (e.g. SPMD/for-all/apply-to-all) or task-parallel (e.g. farm) way and may exploit a distributed shared state which survive to Virtual Processes lifespan. A module can not deterministically accept from one or more input streams a number of input items, which may be decomposed in parts and used as function parameters to instantiate Virtual Processes, according to the input and distribution rules specified in the parmod. Virtual Processes may send onto output streams items or parts of them which are gathered according to the output rules. More details on ASSIST environment can be found in [8-13].

4 System design and implementation in Assist

The aim of the following experiment is to evaluate a Grid high performance implementation of a retrieval system for remotely sensed images. As all retrieval systems it is based on the following steps:

- Information Extraction and Signature generation;
- Query;
- Signatures comparison & sorting.

In the first phase, in order to extract the image features representing the geomorphology and the textures, we use: angular spectrum, Hough and Gabor transform, and pattern directionality. A complete characterization of these features can be found in [1].

This step, characterized by the highest computational load among all the phases of the retrieval system, have been implemented as a pipeline of five blocks:

- the Generator modulus: reads the image names list from a NFS File System and sends it to the ISM modulus;
- the ISM modulus: distributes the file names to Virtual Processors;
- a set of Virtual processors: performs the parallel image processing task. It is characterized by a none topology;
- the OSM modulus: collects outputs from each VP of the set and it sends them to Print modulus;
- the Print modulus: sends signatures to file system to form the signature database.

Performance was here evaluated in terms of throughput and speed. In the current evaluation all the algorithms are coded in the C language. The computational load of algorithms used in this application, i.e. Hough transform that uses the Canny filter which computing time depends on image, is characterized by high variance. This variance, in classical parallel implementation environment such as in MPI based implementation, where the computational load is manual balanced, can be reduced by distributing a group of images to each processor [14]. In this test ASSIST balances automatically the computational load in each Grid Node by distributing images on demand to processors of local cluster with insignificant overhead. In this way, optimal load distribution can be achieved also in heterogeneous systems. In our evaluation, we compared the Grid-ASSIST performances to a single workstation.

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential on a single</td>
<td>240 min</td>
</tr>
<tr>
<td>workstation</td>
<td></td>
</tr>
<tr>
<td>Grid-ASSIST</td>
<td>20 min</td>
</tr>
</tbody>
</table>

Table 1 Time requested for the image features extraction and signatures generation steps in the retrieval system.

Test results, Table 1, show the excellent performance of the Grid-ASSIST prototype.
5 Prototype testing

5.1 Creation of the binary modules
Signature extraction and sorting modules, written in C++ and Matlab languages, have been compiled for Intel 86 architecture. Subsequently a C procedure has been created that calls such modules and supplies images from the DB_SAR archives. So that the binaries work correctly, it has been necessary to transfer, on every Grid nodes chosen by the uses for the execution, the matlab library: mglinstaller.

5.2 Test 1: execution on the grid of the binary modules by means of Globus primitive
Tests have been executed from the Taranto Node on an archives of 455 SAR images (tot. 662Mbyte) distributed on the grid. The operation is carried out by the following steps:
- Signatures extraction on remote nodes
- Collect of the generated signatures
- Distance Computation and sorting
Extraction is executed by means of globus-job-run command, in every allocated grid resource.
The collect operation has been performed using the command globus-url-copy addressing the previously selected resources.

5.3 Test 2: execution of the Assist binaries files in grid environment
The execution of the Assist binaries files on the grid has demanded the presence, on the nodes used for the execution, of ACE libraries.
The reference to these libraries can easy be managed in remote by means of the option - env of globus.
By means of globus-URL-copy command, it has been provided the ACE library transfer to every used nodes.
After libraries transfer it is possible to execute the globus-job-run command.

5.4 Test 3: execution on the grid of the signature extraction modules as Assist 1,2 source
In this phase of the experimentation the modules for the extraction of the signature have been executed. These modules are developed in 1,2 ASSIST environment and compiled with not clam modality.
> astCC - Cc ordinamento.ast
it has supplied the following modules:
- ND0000__genera
- ND0001__qs_ism
- ND0001__qs_vpm
- ND0001__qs_osm
- ND0002__stampa
So that the Assist binaries work correctly, the ACE libraries must be transferred to the nodes.
It has been provided to copy the images archives on the grid nodes participants to the process.

6 Conclusion
In this paper, we discuss the prototype of remote sensed CBIR system for Grid environment realized for the MIUR FIRB project “Grid.it”.
The GRID offers capabilities for distributed computation, but has severe limitation in EO data processing. Remote sensing applications process large data amount, transmission between nodes of these data could eliminate the distributed computation advantage. For this reason a GRID remote sensed CBIR system have to be realized as a Grid Service Spread model (GSS) where Grid service are transmitted instead of data.
A reasonable configuration for a RS CBIR system in GRID environment can be described as follows:
- The image database should be distributed among the different nodes belonging to the grid.
- The number of nodes for each centre should be sized in order to offer adequate computational power to process locally EO images.
- In order to guarantee the privacy and security of the data and of the systems the GRID could be built on a private networks (VPN) with private IP, protected by tunnels and accessible by only one or very few WEB Portals or input computers, using SSH and Certification authority.

References


