Resource Measurements or Water Detection Algorithm in MedioGrid Architecture

Dorian Gorgan, Victor Bâcu
Computer Science Department
Technical University of Cluj-Napoca
dorian.gorgan@cs.utcluj.ro
victorbacu@yahoo.com

Outline

- Objectives
- MedioGrid Project
- MedioGrid Platform Kernel
- Data Repository
- Landsat Satellite Images
- Gond's Water Detection Algorithm
- Greenland and Waterland Applications
- Resource and Performance Measurements
- Results
- Conclusions

Research Objectives

- Resource measurements and performance evaluation
- Optimize workflow throughout the Grid
- Supervise Grid system (e.g. MonALISA framework)
  - Information for computer nodes and clusters
  - Network information (i.e. traffic, flows, and connectivity)
  - Security
  - Remote administration for services and applications
  - Graphical user interfaces to visualize complex information, and global monitoring repositories for distributed Virtual Organizations
- Optimize computation time
- Experiments on computation performance
  - As a function of data dimension, number of processing jobs and number of available workstations
- Specific application context
  - Waterland application runtime
  - Detects the water areas in Landsat satellite images

MedioGrid Project

MedioGrid

Parallel and distributed graphical processing on Grid structure of geographical and environment data, 19CEEX-I03 (2005-2008)

The MEDIOGRID project aims to accomplish a pilot program to process the images acquired in real time from meteorological and resource satellites, in order to extract the meteorological and environment parameters that characterize the atmospheric and terrestrial state.

Project consortium:

1. Computer Science Department, Technical University of Cluj-Napoca, coordinator
2. Faculty of Geography, Babes Bolyai University, Cluj-Napoca
3. iQuest Company , Cluj-Napoca
4. National Administration of Meteorology, Bucharest
5. Computer Science Department, Politehnica University of Bucharest
6. Informatics Department, West University of Timisoara
7. Computer Science Department, Politehnica University of Timisoara

MedioGrid Project

- Main objectives
  - Develop Grod structure to support the parallel and distributed processing of huge data (geographical and environmental)
  - Develop algorithms for Grid based processing of satellite images
  - Develop and experiment environment supervising applications with data extracted from satellite images
  - Model and visualize the virtual geographical space

- The project schedule:
  - 1st year achieves and experiments the grid infrastructure, and analyzes the raw data and the processing techniques.
  - 2nd year develops the Software Platform Kernel consisting of fundamental algorithms and components for image segmentation, and parallel and distributed data processing. It follows the kernel system experimentation over the grid by test and real input data.
  - 3rd year develops and tests a pilot application specific for the analysis of social and ecological systems.

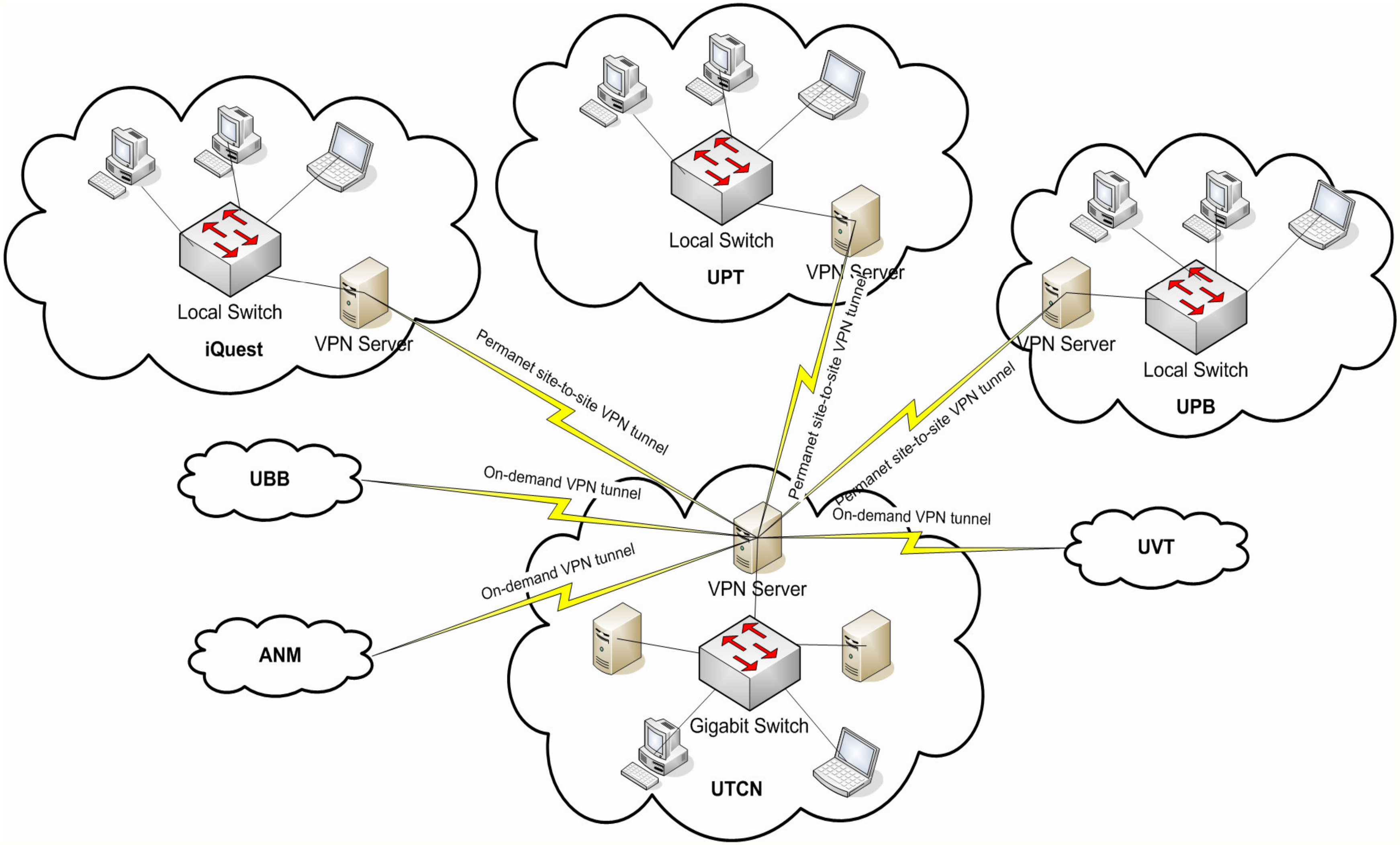
MedioGrid Project

- Outcomes (2005-2007):
  - Functional MedioGrid network (experimental Grid of 6 servers - Cluj, Timisoara, Bucharest and more than 50 workstations)
  - Software applications: MedioGrid Software Platform Kernel, Image processing MODIS (NASA), Cloud detection, Vegetation classification (Greenland), Water detection (Waterland)
  - Modeling and visualization of the virtual geographical space, GIS and LBS Kernel (Location Based Services)
  - Active Objects based modeling and execution
  - User interaction techniques for image based applications
  - Visual Grid process description and scheduling for satellite image domain
  - Grid and Web services based architecture
  - Organize conferences and workshops: GridCAD (Timisoara 2006, 2007), MedioGrid Workshop (Cluj-Napoca 2005)

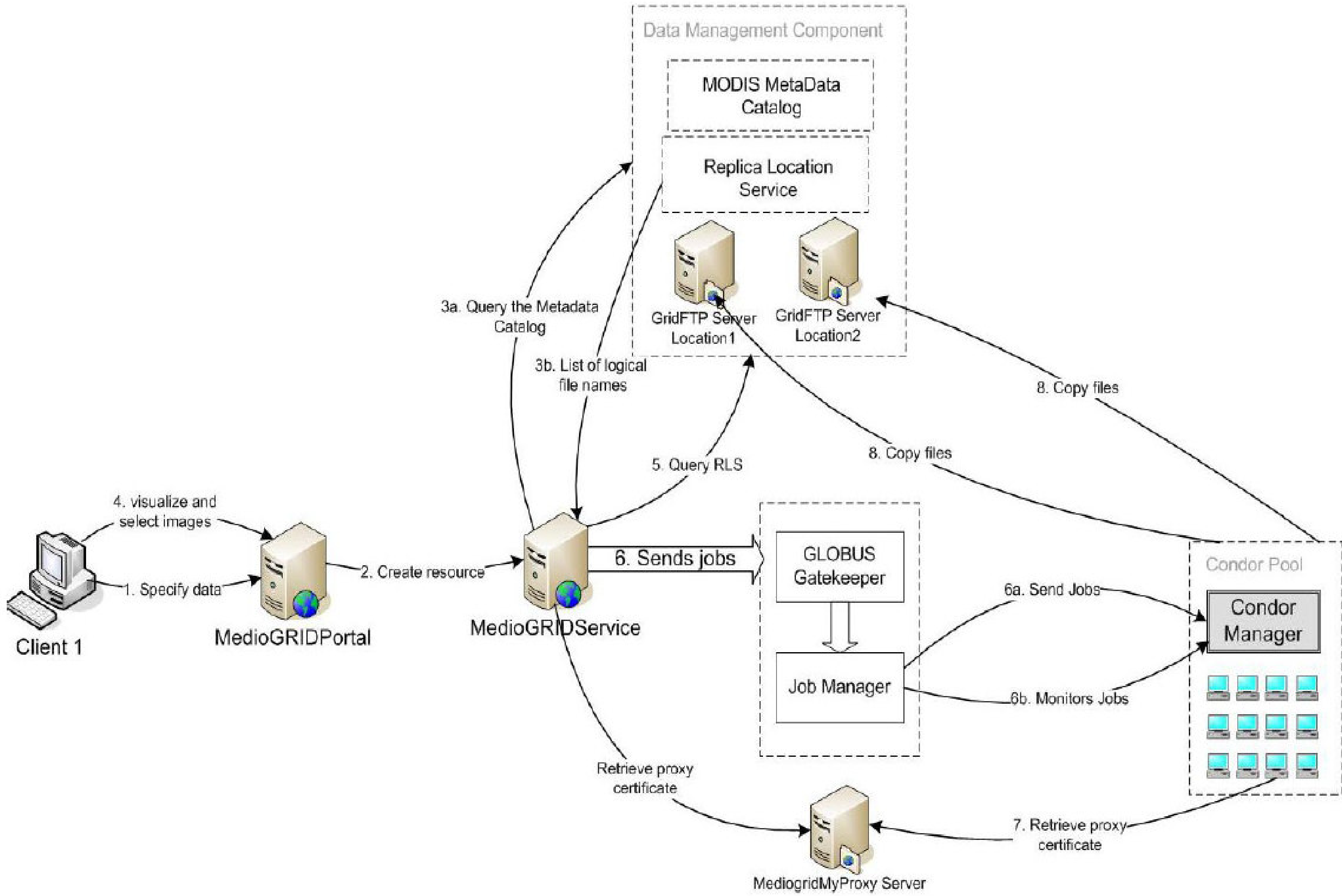
MedioGrid Project

- Future works:
  - Develop the Grid and Web services for geographical and environment applications
  - Web Semantic services toward MedioService Architecture
  - Geographical and Environmental Ontology and Knowledge Database
  - Geographical and environmental Grid pilot applications
  - Grid visualization
  - GIS and LBS applications
  - Active objects based distributed modeling and processing

MedioGrid architecture



MedioGrid Architecture



MedioGrid Platform Kernel

- MediogridService
  - supports as Web and Grid service the creation, the execution, and the scheduling of the jobs
- MediogridFactory
  - creates resource entities which keep information about the jobs
- MediogridResource
  - contains information about the job state, the start and the end time of the job
- MediogridOGSADAI
  - provides access to the image database
- MediogridURLCopy
  - supports the file transfer using the GridFTP protocol
- MediogridRLS
  - returns a list of physical images in order to allow the worker to decide where is the most appropriate GridFTP location to download data

- Job Manager
  - Condor
  - Alternatives: PBS, Sun Grid Engine

Data Repository

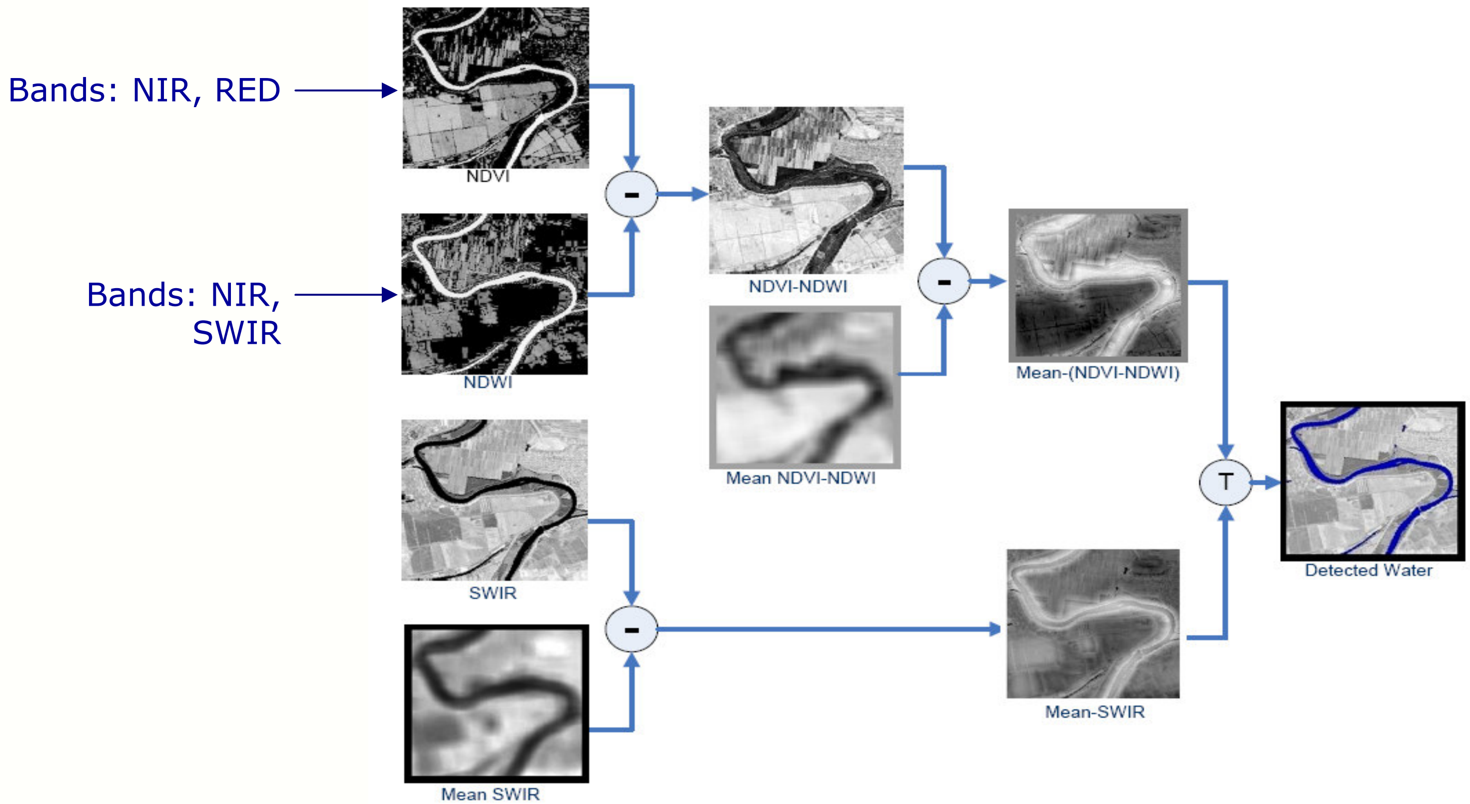
- Satellite image
  - Landsat
    - Massive data. E.g. One image is about 600MB
    - Seven bands, 1-7
  - Modis
    - Produced by sensors onboard the Terra and Aqua satellites
    - Covers the entire surface of the Earth
    - 36 observational channels
    - 250m to 1km spatial resolution
    - Data distributed by the NASA DAAC
- Layered on eXist (XML database)
- OGSA-DAI technology

Landsat Satellite Images

- Massive data. E.g. One image is about 600MB
- seven bands 1-7
  - band 1 - water body penetration
  - band 2 - green reflectance of vegetation
  - band 3 - sensitive to chlorophyll absorption, determine the vegetation types
  - band 5 – information on vegetation and soil moisture
  - band 6 - vegetation stress
  - band 7 - discriminates the mineral and rock types
- Different information is highlighted by various band weighted combinations
  - bands 4, 3, and 2
    - Classify land water boundaries and different types of vegetation
  - bands 4 (NIR), 5 (SWIR), and 3 (RED)
    - Land/water boundaries and vegetation areas
    - Water detection

Gond’s Water Detection Algorithm

- Gond’s water detection algorithm
  - Gond V., Bartholom E., Ouattara F., Nonguierma A. and Bado I. Surveillance et cartographie des plans d’eau et des zones humides et inondables en rgions arides avec l’instrument VEGETATION embarque sur Spot 4, International Journal of Remote Sensing, 2004, 25,5. pp. 987- 1004.
- SWIR (5), Red (3) and NIR (4) spectral bands



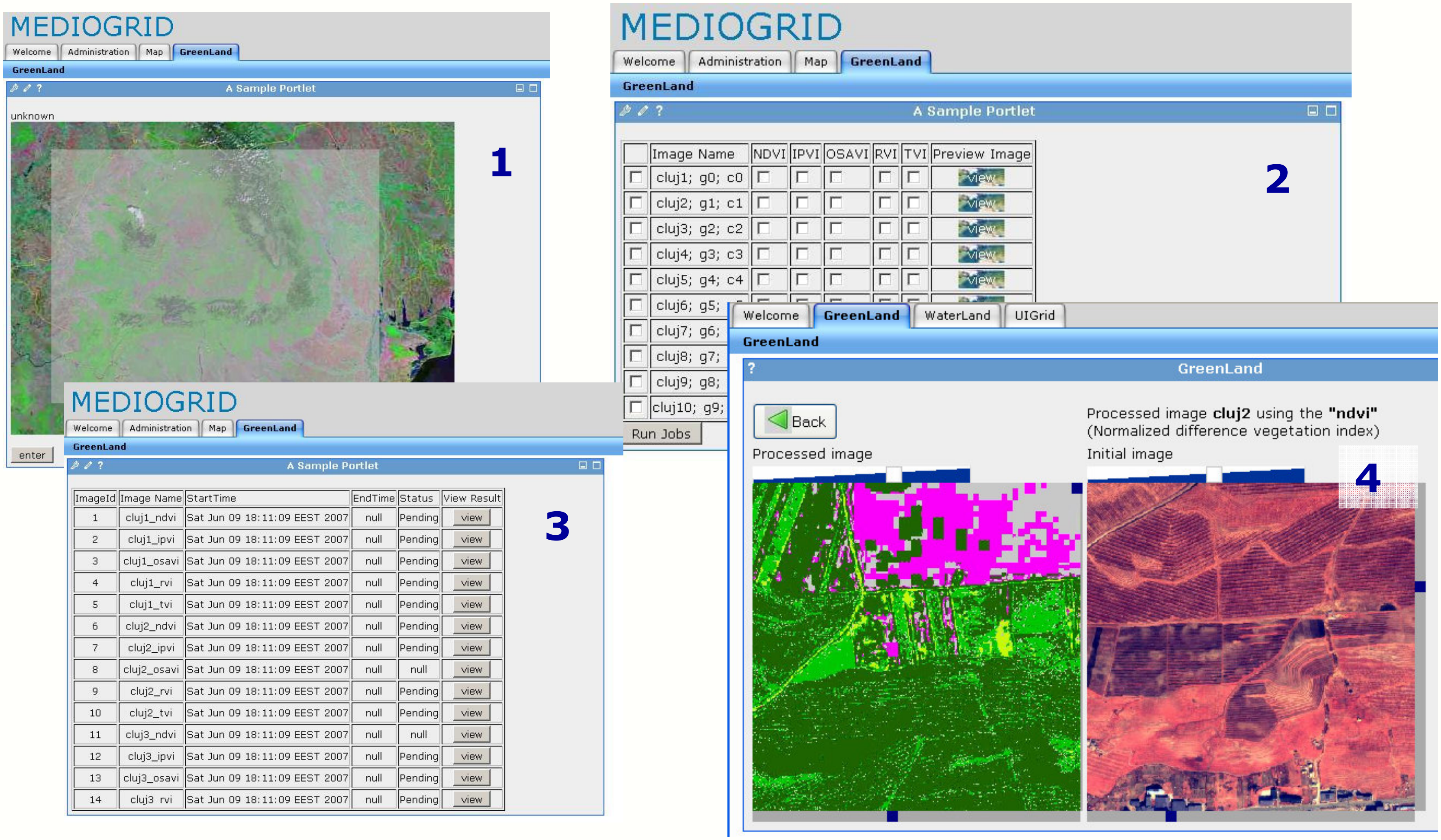
Water Detection Algorithm - Results

Pseudo colored initial Landsat image.

Water detected areas.

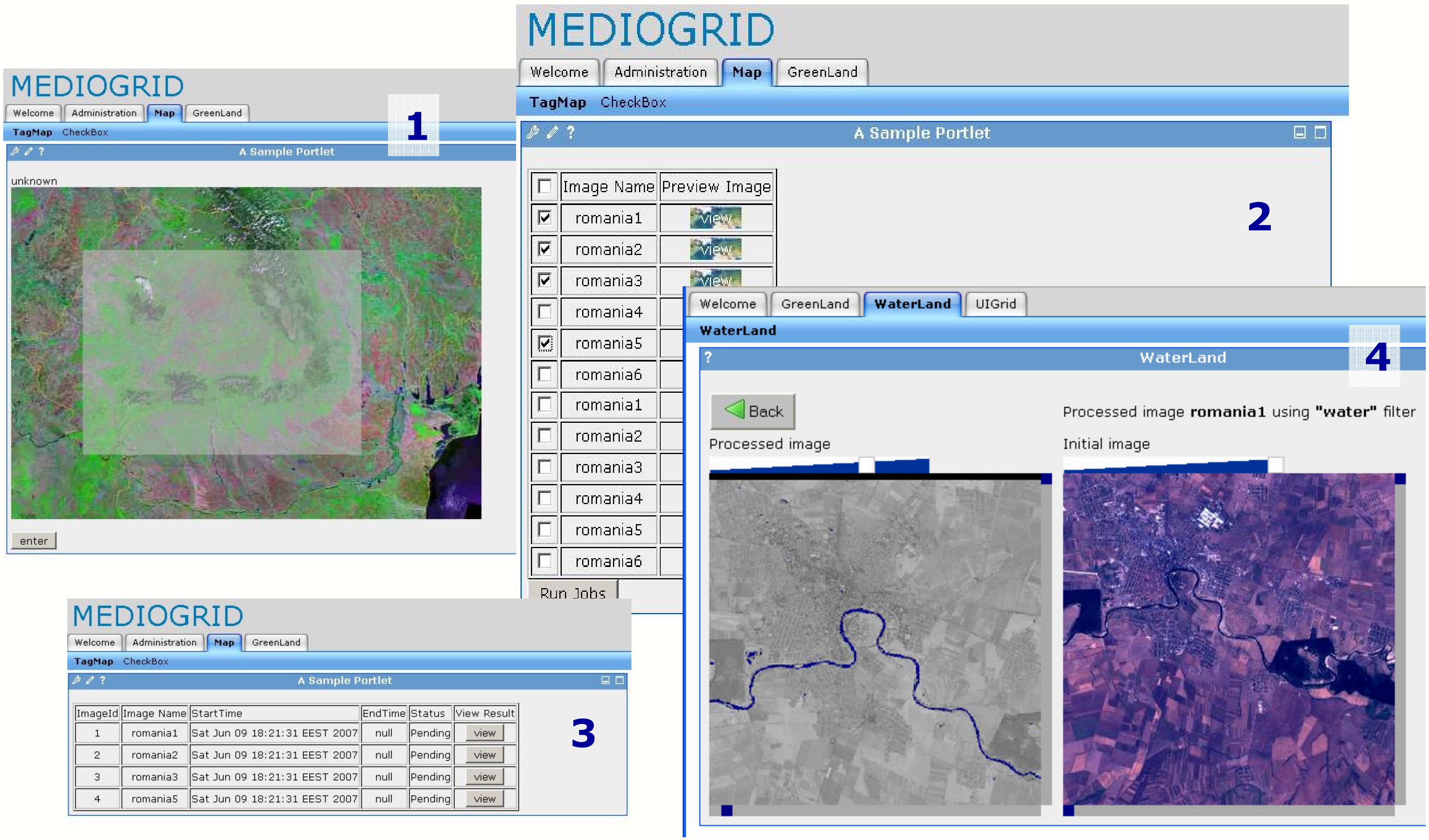
Greenland Application

On-line available Web application: greenland.mediogrid.utcluj.ro



Waterland Application

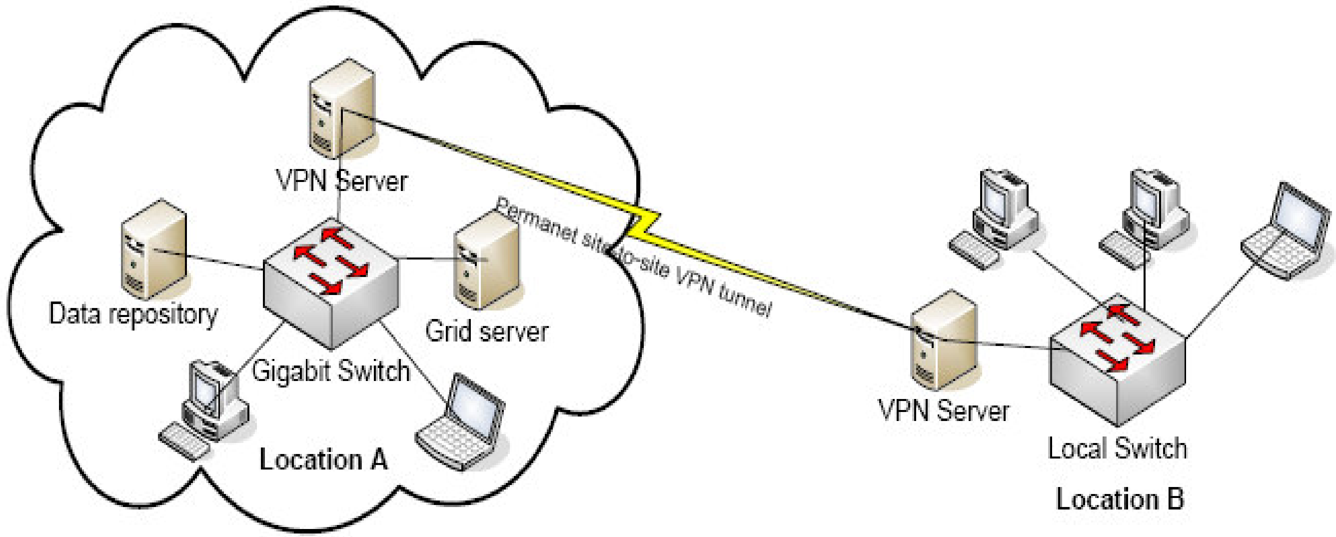
On-line available Web application: greenland.mediogrid.utcluj.ro



Resource Measurement Objectives

- Optimize computation time
- Experiments on computation performance
  - As a function of data dimension, number of processing jobs and number of available workstations
- Specific application context
  - Waterland application runtime
  - Detects the water areas in Landsat satellite images

Performance Measurement Environment



No.	Specifications	OperatingSystem	Location
4	Intel Xeon 3GHz	Linux	Location A
4	Intel Xeon 3GHz	Windows	Location A
7	INTEL PIV 630 3,0GHz	Windows	Location A
5	INTEL PIV 630 2GHz	Windows	Location B

Experimental Cases

1. Image dimensions

    1500x1500 pixels

    8000x8000 pixels

2. Number of active workstations

    10 workstations

    20 workstations

3. Number of jobs

    1 . . 150 jobs

Concepts:

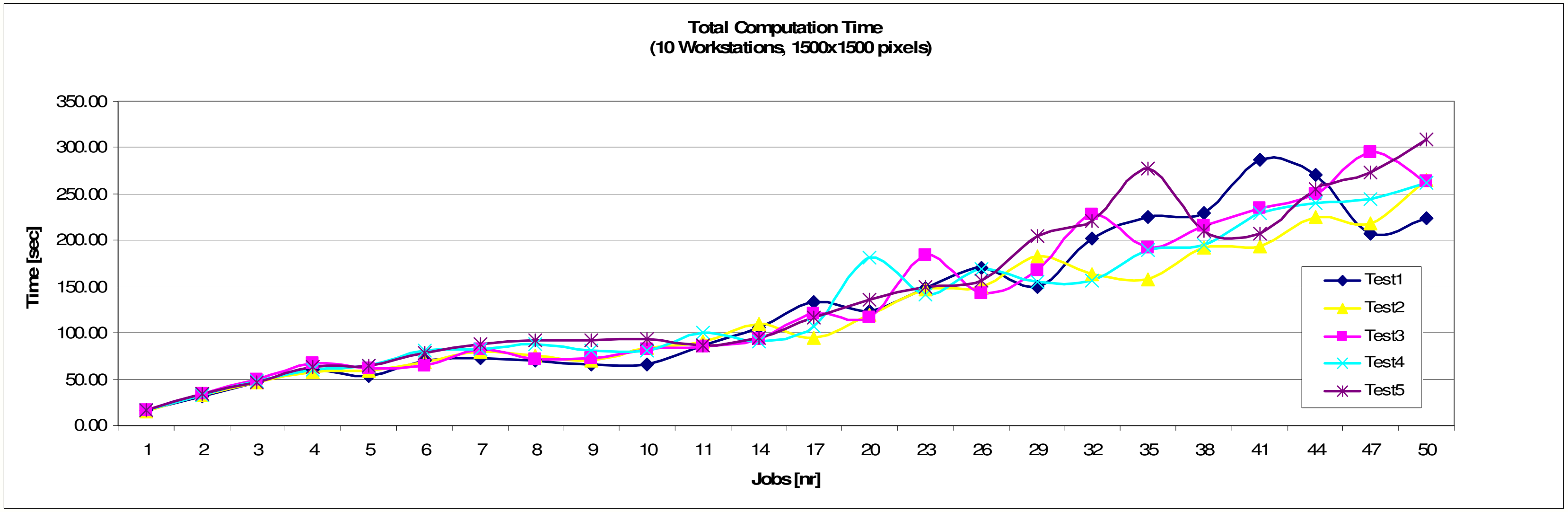
Worker loading: k=j/w,

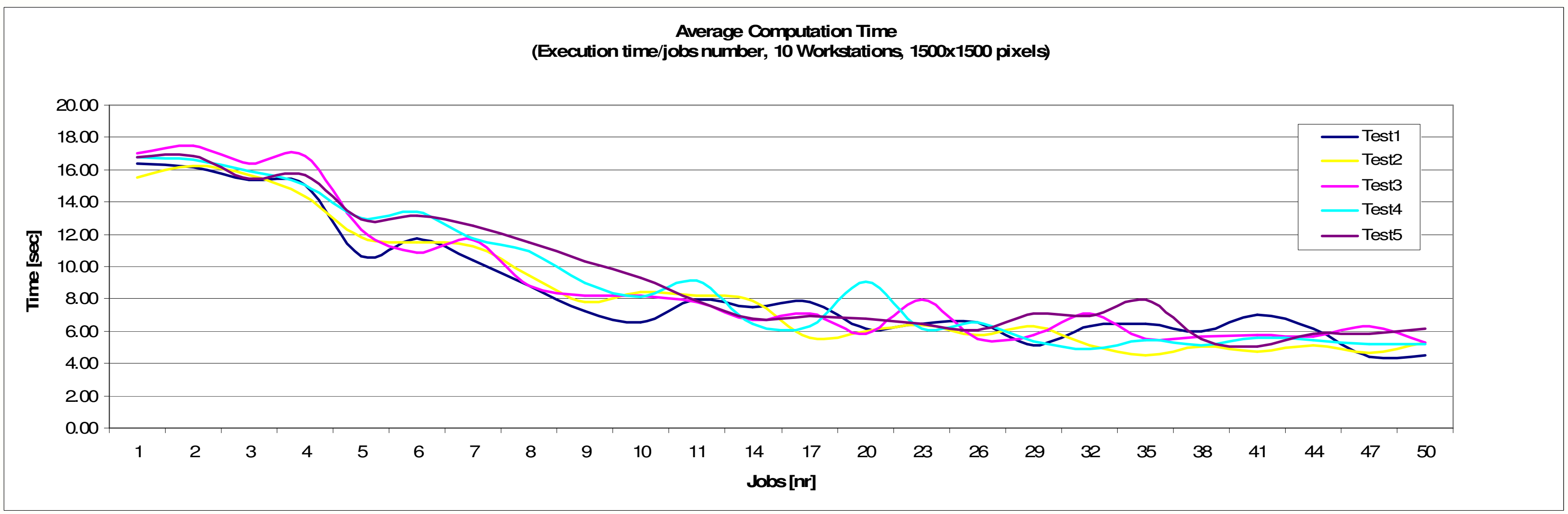
where j = number of running jobs, w = number of active workstations

Average Computation Time: Total computation time/ number of jobs

VET (Virtual Execution Time): average computation time for the virtual Grid computer as an entity

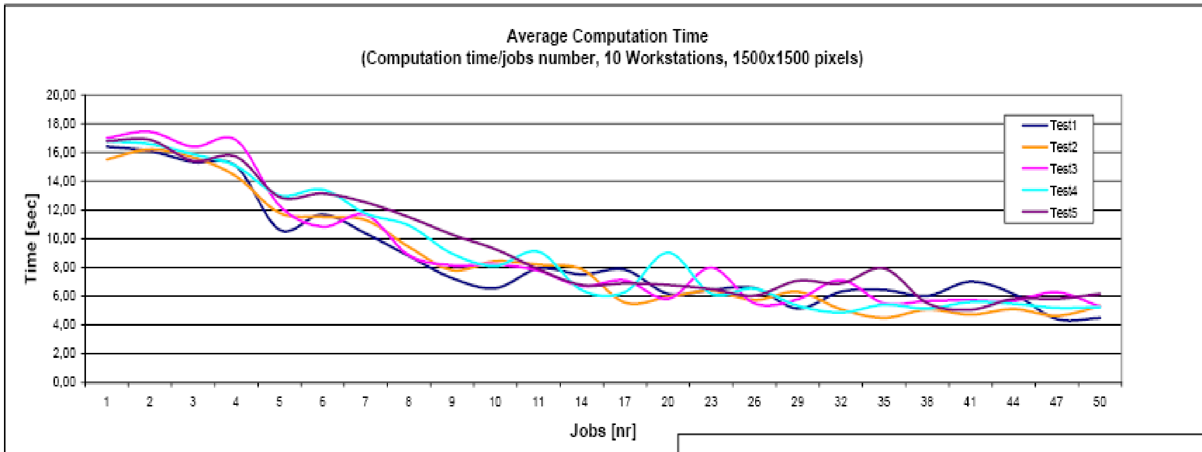
Results

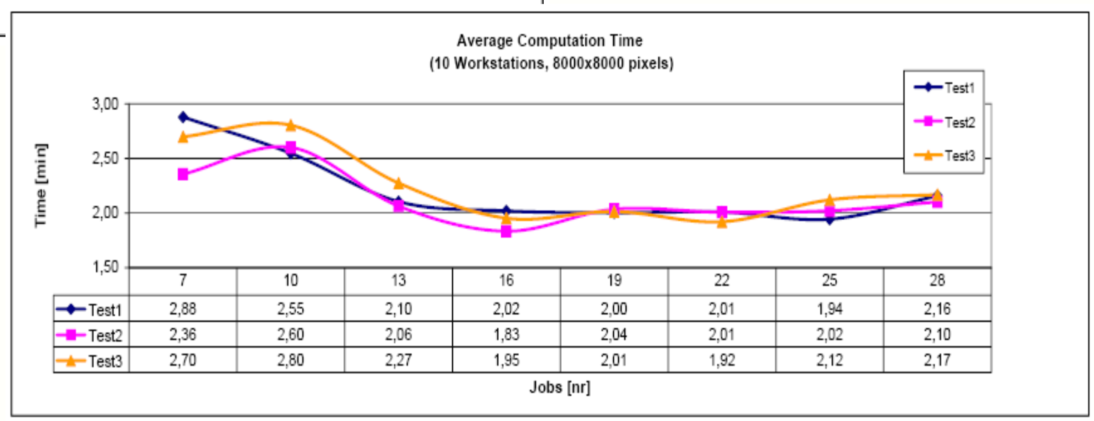

Total Computation Time
(10 Workstations, 1500x1500 pixels)
Time [sec]
350.00
300.00
250.00
200.00
150.00
100.00
50.00
0.00
1 2 3 4 5 6 7 8 9 10 11 14 17 20 23 26 29 32 35 38 41 44 47 50
Jobs [nr]
Test1
Test2
Test3
Test4
Test5


Average Computation Time
(Execution time/jobs number, 10 Workstations, 1500x1500 pixels)
Time [sec]
20.00
18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
1 2 3 4 5 6 7 8 9 10 11 14 17 20 23 26 29 32 35 38 41 44 47 50
Jobs [nr]
Test1
Test2
Test3
Test4
Test5

Results

- The execution time for one job is about 17 sec for the 1500x1500 pixels image, and 13 min for 8000x8000 pixels image
- The execution time across the Grid drops down significantly, to 6 sec for the 1500x1500 pixels image, and about 2.15 min for the 8000x8000 pixels image (10 workstations)





	7	10	13	16	19	22	25	28
Test1	2,88	2,55	2,10	2,02	2,00	2,01	1,94	2,16
Test2	2,36	2,60	2,06	1,83	2,04	2,01	2,02	2,10
Test3	2,70	2,80	2,27	1,95	2,01	1,92	2,12	2,17

Conclusions

- A solution for processing huge satellite images could be the computation of small tiles (i.e. 1000x1000 pixels) reducing in this way the required workstation's resources
- The job execution time is the total time necessary for a workstation to copy and process locally the input data and finally to store the result image into repository
- VET theoretically decreases to zero if the number of workstations is greater or equal with the number of submitted jobs, which goes to infinity
- VET is lower then the real processing time for the overlapping of some stages in the execution of a job (i.e. stage in, active, stage up, clean up)
- VET proves really the concept of Grid as a virtual execution environment

  (i.e. virtual computer)
- To reduce the VET time it should increase the number of workstations and significantly much more the number of jobs as well

  (Obvious conclusion: Does not make sense at all to increase the number of workstations more than the number of jobs)

Many thanks. Questions

Victor Bâcu, Dorian Gorgan
Computer Science Department
Technical University of Cluj-Napoca
dorian.gorgan@cs.utcluj.ro
victorbacu@yahoo.com