

# A Grid Enabled Monte Carlo Hyperspectral Synthetic Image Remote Sensing Model (GRID-MCHSIM) for Coastal Water Quality Algorithm

Gen-Tao Chiang\*<sup>a</sup>, Martin Dove<sup>a</sup>, Stuart Ballard<sup>a</sup>, Charles Bostater<sup>b</sup>, Ian Frame<sup>a</sup>

<sup>a</sup>National Institute for Environmental eScience and Department of Earth Sciences, University of Cambridge, Cambridge, United Kingdom; <sup>b</sup> Marine Environmental Optics Laboratory & Remote Sensing Center, College of Engineering, Florida Tech, Melbourne, FL, USA

## ABSTRACT

Previous studies indicate that parallel computing for hyperspectral remote sensing image generation is feasible. However, due to the limitation of computing ability within single cluster, one can only generate three bands and a 1000\*1000 pixels image in a reasonable time. In this paper, we discuss the capability of using Grid computing where the so-called eScience or cyberinfrastructure is utilized to integrate distributed computing resources to act as a single virtual computer with huge computational abilities and storage spaces. The technique demonstrated in this paper demonstrates the feasibility of a Grid-Enabled Monte Carlo Hyperspectral Synthetic Image Remote Sensing Model (GRID-MCHSIM) for coastal water quality algorithm.

**Keywords:** environmental eScience, eScience, GRID, cyberinfrastructure, hyperspectral remote sensing, coastal water quality, synthetic image generation, monte carlo method,

## 1. INTRODUCTION

The monte carlo hyperspectral synthetic remote sensing model (MCHSIM) is used to produce images using reflectance of the water surface and these synthetic images are compared to an actual aerial photo obtained from an aircraft in order to improve the accuracy of the numerical model<sup>1</sup>. The MCHSIM is used to simulate a photon at any time step, and then examining the results of the photon tracks or the displacement of the photons as they pass through the water column in a three-dimensional coordinate system. This process simulates a three-dimensional light field of photons which originate at a single point just above the air/water surface.

The MCHSIM and associated image processing techniques require a large amount of computing power. Previous research indicates the use of a parallel processing or MPI is feasible. However, due to the limitations of local clusters, the program can only simulate one band and 1000\*1000 pixels image each time on 16 processors. In comparison, real hyperspectral data usually has more than 200 channels. For instance, NASA is continuously gathering hyperspectral images using Jet Propulsion Laboratory's Airborne Visible-Infrared Imaging Spectrometer (AVIRIS), which measures reflected radiation in the wavelength range from 0.4 to 2.5  $\mu\text{m}$  using 224 spectral channels at spectral resolution of 10 nm. Currently, NASA is using parallel computing on hyperspectral data processing, such as geometric correction and feature extraction. However, in order to generate a hyperspectral cube (Figure 1) and doing more advanced algorithm testing, such as, extracting bottom reflectance of submerged aquatic vegetation (Figure 2), more computational power is required. Grid computing could provide the necessary computational power to resolve this issue.

---

\* Gen-Tao Chiang; gen-tao@niees.ac.uk; phone +44 1223 764918; fax +44 1223 333450; <http://www.niees.ac.uk>; National Institute for Environmental eScience, Department of Earth Sciences, University of Cambridge, Downing Site, Cambridge, CB2 3EQ, United Kingdom.

Grid computing is a term referring to the sharing of computer resources over the internet. The Grid goes well beyond simple communication between computers, and aims ultimately to turn the global network of computers into one vast resource of computing power and storage capacity. A group of researchers could set up a virtual organization (VO) that shares the computer processing power, databases, data storage facilities, and scientific instruments from their institutions

Grid computing software usually referred to as “middleware”, manages these shared resources. When a computational job is submitted, the middleware allocates appropriate resources to that job, moves the necessary input data for that job, returns the results and provides the visualization capacity to present the results. When a computational ‘job’ is submitted, the middleware either places the job in a queue, or if appropriate resources are available, it allocates them to that job or places that job in a queue if resources are currently utilized.

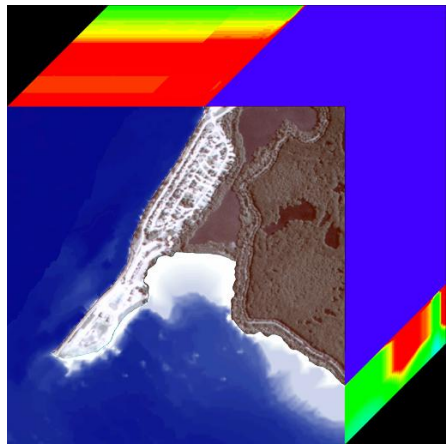


Fig. 1. The synthetic hyperspectral cube generated by analytical model.

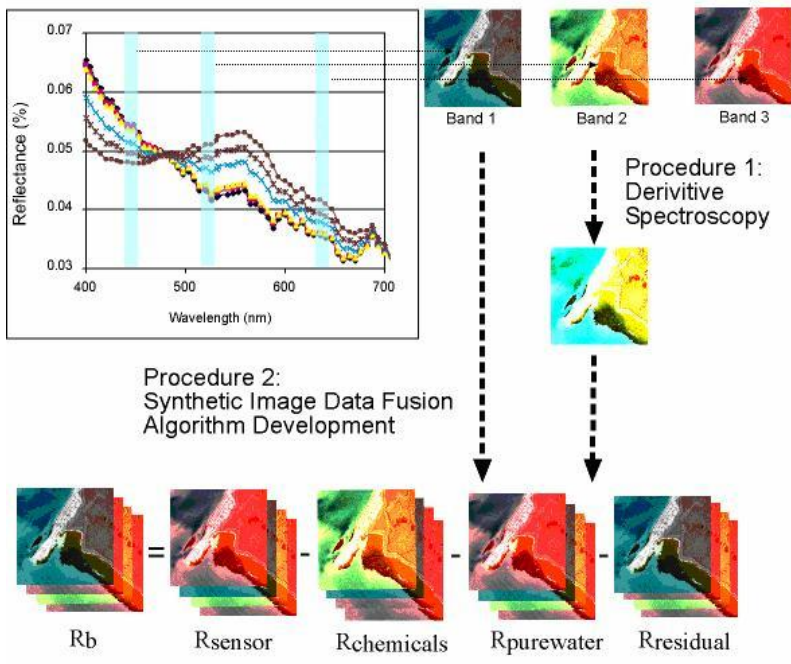


Fig. 2. Conceptual diagram indicating procedures for using synthetic images for remote sensing algorithm testing and data fusion of aerial sensor data using synthetic images. The reflectance which the sensor received could be the results from Bottom reflectance, reflectance of chemicals, water, and residuals.

There are three challenges regarding the usability of escience. 1. Maximising escience technologies to support new forms of global communities; 2. Exploitation of escience infrastructure to support knowledge production and expertise in escience; 3. Design, assessment and management in global escience systems. In the UK eMinerals project, consists three major components (Figure 3), which are Data Grid providing high performance integrated access to distributed massive data resources, Computing Grid supporting large scale and distributed modeling capability, and Collaborating Grid allowing large scale group working together. In this paper, we will discuss in more detail about these three components in the following section to see how we can use those escience tools to obtain more computing resources to expend the capability of MCSHIM and how to handle those generated images or output data in data grid.

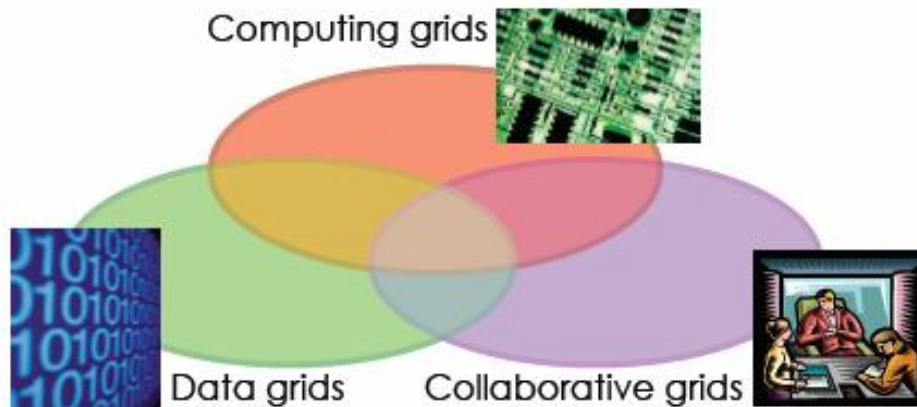


Fig. 3. A typical grid infrastructure usually consists of three components, Computing grids, Data grids, and Collaborative grids<sup>3</sup>.

## 2. METHODS

Building an e-Infrastructure is the first step in order to achieve the scope of escience. E-infrastructure consists of the network infrastructure, hardware resources (computers, data archives, and instruments), grid middleware, development tools, domain specific applications, certificate authorities, supporting organizations...etc. The concept of the environmental e-infrastructure can be described as the Figure 4 and explain below.

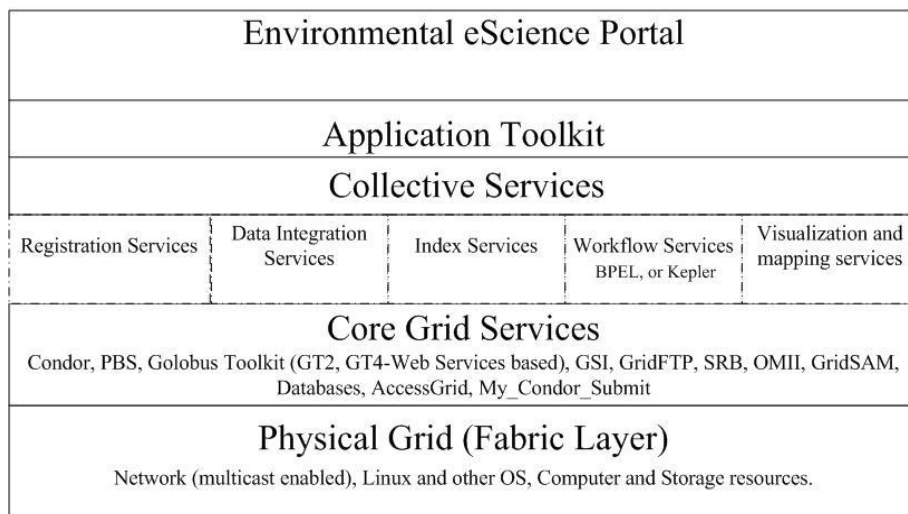


Fig. 4. A concept of the environmental e-infrastructure is going to be built in NIEeS.

The Physical Grid layer provides the network, computing resources, storage resources, and operation systems. Due to a large amount data transfer and the nature of collaborating tools, multicast enabled network is required. In order to provide reliable services in the future, fabric management technologies have to be implemented, for example, cluster management and monitoring systems, storage usage monitoring system, and accounting system.

The Core Grid Services layer provides common grid services, such as: Condor, Globus, GridSAM, SRB and access grid, to access distributed resources. The Collective Services layer includes many services on top of the core services. For instance, registration service may provide data and metadata registration tool, such as Rcommands, which provides a set of scriptable commands to associated metadata to files stored within a distributed files system such as the Storage Resource Broker, which will be discussed later in section 2.2. Rcommands enable the creation of metadata to be automated. Data integration and index services may integrate ontology for improving searching function and XML data parsing abilities. Workflow services will provide the tool for environmental scientists to design their experiments and scientific workflow. Visualization and mapping services will integrate GIS for environmental data visualization. The Application toolkit layer will implement applications from different environmental science fields, for instance, the MCHSIM will be considered in this layer. On top of the environmental e-infrastructure is an environmental eScience portal, which provides a user friendly interface for environmental scientists to access applications and grid services.

## **2.1 Building a Computing Grid Environment**

The National Institute for Environmental eScience (NIEeS) provides a computing grid environment for a number of different use case scenarios.

### **COMPUTING GRID**

#### **Condor**

Condor is a specialized workload management system for compute-intensive jobs developed by University of Wisconsin. Like PBS or other queuing systems, Condor provides a job queuing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Users submit their serial or parallel jobs to Condor, Condor places them into a queue, chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion. Condor's novel architecture allows it to succeed in areas where traditional scheduling systems fail, such as managing heterogeneous computing resources. Condor can also be used to manage a cluster of dedicated compute nodes (such as a "Beowulf" cluster).

Condor can be used to build Grid-style computing environments that cross administrative boundaries. Condor's "flocking" technology allows multiple Condor compute installations to work together. Condor incorporates many of the emerging Grid-based computing methodologies and protocols. It is also fully interoperable with resources managed by Globus. As a result, Condor can be used to seamlessly combine all of an organization's computational power into one resource<sup>4</sup>.

NIEeS has deployed Condor version 6.8.0 and integrated into the CamGrid; CamGrid is a University of Cambridge project which aims to build a university-wide grid based on the Condor middleware. However, since NIEeS is providing services for the UK environmental science community, we need to allow external collaborators use NIEeS grid resource. Globus and GridSAM are used for this purpose.

#### **Globus**

Globus tool kit is developed by Argonne National Lab. It provides a set of software tools to implement the basic services and capabilities required to construct a computational Grid, such as security, resource location, resource management, and communications. Globus includes programs such as: GRAM (Globus Resource Allocation Manager), which figures out how to convert a request for resources into commands that local computers can understand; GSI (Grid Security Infrastructure), which provides authentication of the user and works out that person's access rights, and GridFTP which provides secure data transfer mechanism.<sup>5</sup> NIEeS is running Globus 4.0.2 as interface to the NIEeS Condor Pool, which is part of the CamGrid and PBS cluster.

NIEEs deploys GRAM and Web Services based GRAM for UK environmental scientists to use. It also supports GridFTP but not MDS at this moment. MDS (Monitoring and Discovery Service) is used to collect information about resource (processing capacity, bandwidth capacity, type of storage, etc). In order to use grid services, one can apply for a NIEEs certificate. A certificate uses a digital signature to bind together a public key with an identity information such as the name of a person, organization, email, and so forth. The certificate can be used to verify that a public key belongs to an individual. At this moment, NIEEs accepts both UK eScience and NIEEs CA. Certification Authority (CA) is an entity which issues digital certificates for use by other parties

### GridSAM

GridSAM is an open-source job submission and monitoring web service. The aim of GridSAM is to provide a Web Service for submitting and monitoring jobs managed by a variety of Distributed Resource Managers (DRM). The modular design allows third-party providing submission and file-transfer plug-ins to GridSAM. Moreover the job management API used by the GridSAM web service can be embedded into grid application that requires job submission and monitoring capabilities. NIEEs has deployed GridSAM as interface to NIEEs condor pool. The topography of the NIEEs Computing Grid can be described in Figure 5. Basically, one can submit jobs from anywhere to NIEEs Computing Grid with valid certificates. Then condor will find the matching resource and flock jobs to other departments.

### My\_Condor\_Submit (MCS)

My\_condor\_submit (MCS) is a tool developed by the eMinerals project to allow simplified job submission to remote grid resources with in built meta-scheduling and load balancing, data management and metadata management functionality. The meta-scheduling is implemented within MCS itself while the job submission is handled by Condor-G and the metadata capture and storage are handled by RCommands respectively. Data management is handled using the SRB Scommands<sup>6</sup>. SRB is the data grid tool used in NIEEs and will discuss later in section 2.2. The benefit of MCS is solving the data IO problem of Globus which benefit from GSI security function. One can submit MCS jobs from client machine with Condor-G, Globus and SRB client installed.

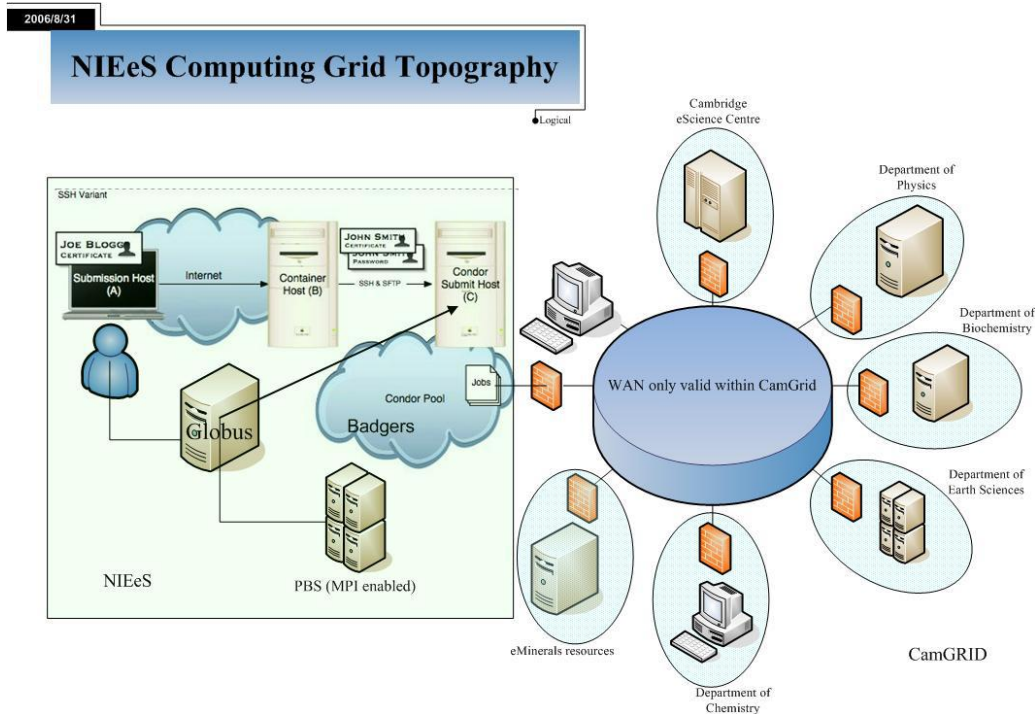


Fig. 5. A concept diagram of Computing Grid environment of NIEEs. NIEEs Grid environment (left), which includes Condor pool, PBS, Globus, GridSAM, MCS, and SRB, links to the CamGrid resources from different departments.

## 2.2 Building a Data Grid Environment

### Storage Resources Broker (SRB)

The **Storage Resource Broker** (SRB), developed at the San Diego Supercomputing Center (SDSC), provides access to distributed data from any single point of access. From the viewpoint of the user, the SRB gives a virtual file system, with access to data being based on data attributes and logical names rather than on physical location or real names. The Physical location is seen as a file characteristic only. One of the features of the SRB is that it allows users to easily replicate data across different physical file systems in order to provide an additional level of file protection<sup>7</sup>.

NIEES deploys a MCAT enabled SRB server. Meta data Catalog (MCAT) is a meta data repository system to provide a mechanism for storing and querying system-level and domain-dependent meta data using a uniform interface. MCAT provides a resource and data object discovery mechanism that can be used to identify and discover resources and data objects of interest using a combination of their characteristic attributes instead of their physical names. On the other words, MCAT is used for mapping the location of logical files and physical files. The results generated from the computing grid can be put in the SRB.

## 2.3 Building a Collaborative Grid Environment

### AccessGrid

The collaborative Grid component utilizes Access Grid software. The Access Grid is developed by the Futures Laboratory at Argonne National Laboratory. The Access Grid is a set of resources that are used to support distributed collaborative interactions across the internet. The main framework is scalable videoconferencing, augmented by a number of presentation and application sharing tools. The Access Grid can be used for large-scale distributed meetings, smaller collaborative work sessions, seminars, lectures, tutorials, and training. The Access Grid thus differs from other tools that focus more on individual-to-individual communication, although it can be used in this mode<sup>8</sup>.

The Collaborative Grid environment will not explored in detail, since the major component for MCHSIM model will be focus on Computing. However, collaborative tools will have been used for future usage in order to share and visualize the model outputs.

## 2.4 Modification of the Hyperspectral Monte Carlo model for the eScience Environment

The hyperspectral MC model was originally developed for serial calculations and produces a value representative of one pixel. The parallel program uses FORTRAN-90, MPI, and has been developed in order to produce an image matrix, which containing the water surface reflectance value. Value of each pixel in this matrix is calculated by the Monte Carlo model. Each band using different coefficient values from each wavelengths. One can then choose any three bands to display an RGB image using visualization tools such as GIS software or Google Earth.

A Grid enable MCHSIM can be produced by two different 'modifications of the code. The first modification approach is to take out the MPI code and rewrite the code or use a script to submit each pixel and input data to a Grid environment. However, in this case, one has to know which condor process is running which pixel, location, or band. This process would therefore be quite challenging. This procedure can be described in Figure 6. The second modification approach is simply keep the original MPI version but using Grid tools for accessing more resources.

### FIRST APPROACH

The workflow of the first way can be described as following. First, images such as an aerial photo for water simulations, containing both land and water area are imported. The land and water areas are then separated. Several methods can be applied to perform this separation, such as classification functions from image processing tool. Then, a matrix will be created contains only 0 (water) and 1 (land). A script will be written to read the above matrix and related spectrum profile and input data, such as:

- specific absorption coefficient for pure water,
- the specific absorption coefficient for chlorophyll,
- the specific absorption coefficient for suspend sediment,

- the specific absorption coefficient for dissolved organic matter (DOM),
- the specific backscattering coefficient for pure water,
- the specific backscattering coefficient for chlorophyll,
- the specific backscattering coefficient for suspend sediment,
- the concentration of chlorophyll,
- the concentration of suspend sediment,
- the concentration of dissolved organic matter (DOM),
- the depth of the water column at that pixel location.

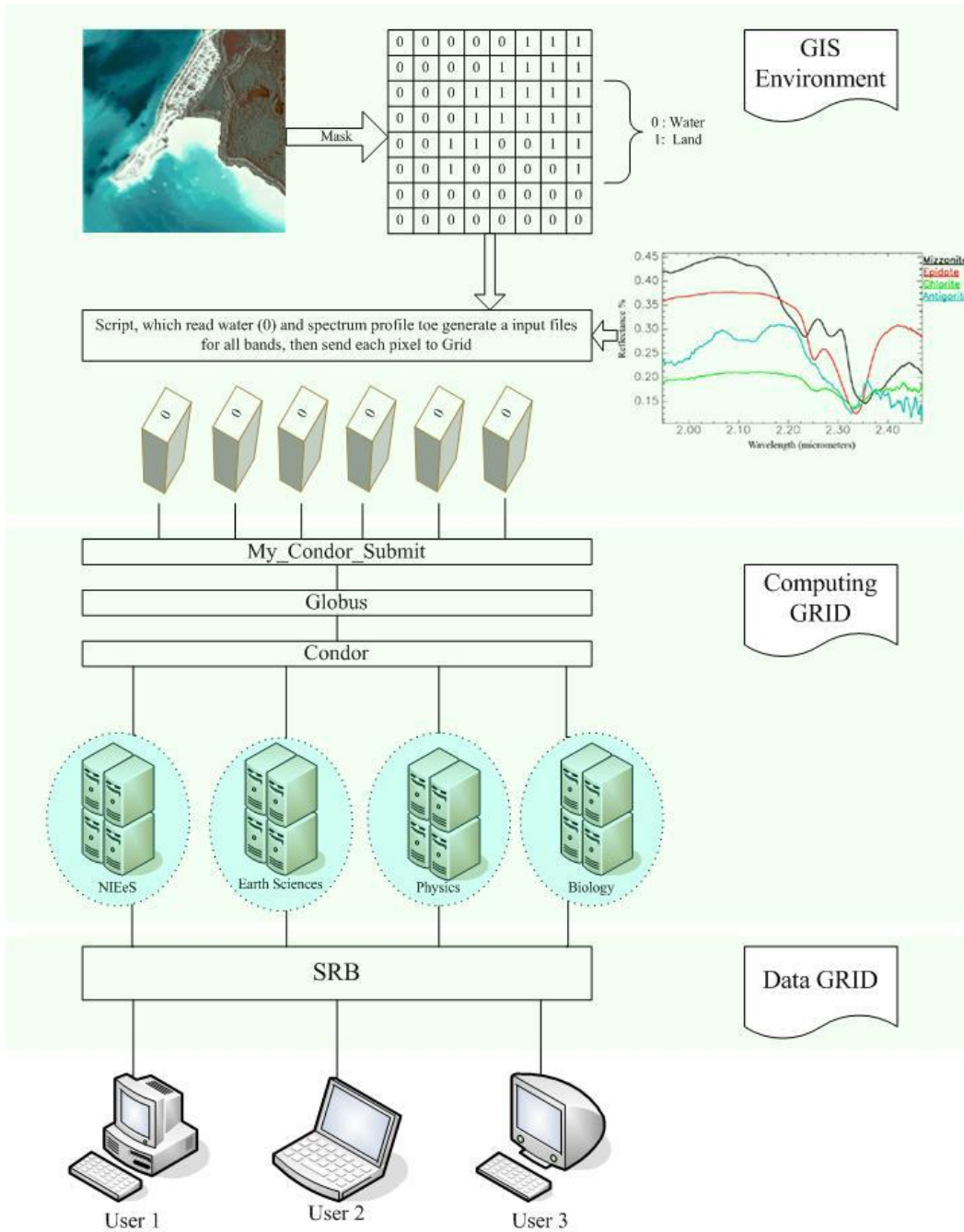


Fig. 6. Schematic figure showing the procedure for synthetic image generation in eScience environment.

After reading all the input data, a number of hyperspectral cubes of each pixel will be created. Each cube will then be submitted to MCS, and mapping the process ID to the pixel ID. MCS then will passing the job to Globus and Globus will verify whether the user has permission to run the jobs on those resources or not. Each tile would then be executed across the CamGrid VO, and run on different resources across a number of different departments.

## SECOND APPROACH

The alternative approach is to simply use the original MPI version but using Grid tools. This approach can access large amount resource and do better data management. According to the Grid resources, a MCHSIM hyperspectral cube can be generated which is not possible can be done before. This approach presented the challenge of configuring the grid environment to support MPI. Unfortunately, most departments within the CamGrid VO are currently not supporting MPI, nor the file sharing system. Coordinating different departments to support MPI and overcome firewall issues between each site is the hardest step. To overcome these issues, a MPI testbed was built within NIEeS grid environment, to test the whole workflow. The results are shown in the next section.

### 3. USING GRID

We have setup a small testbed to support MPI and testing the workflow using MCS, Condor, Globus and SRB. Since MCS needs all input and executable files staged in SRB, Scommands such Sinit, Sput etc have to be used to put all input files in SRB first. Then, one can initiate the globus GSI proxy in order to use MCS. The MCS job description file is shown:

```
*****
Executable      = syntheticp2
Notification     = NEVER

GlobusRSL = (stdout=sig-test.out)(job_type=mpi)(count=2)
Globusscheduler = iguana.niees.group.cam.ac.uk/jobmanager-pbs

#Transfer_input_files = one, two_three, four
# Force overwriting when uploading / downloading files
SForce          = true
SRBHome         = /usr/local/srb/SRB3_4_0/utilities/bin

Sdir            = /NIEeS/home/gtniees.NIEeS-1/SIGtest/
Sget            = *
Sput            = *
queue
*****
```

The executable is the executable file of MCHSIM. It is called syntheticp2 in this case. The second option GlobusRSL is used to specify standard input, standard output, job\_type, and number of processors. We have to specify mpi as job\_type and count=2 means two processors are used. Globusscheduler is the globus gatekeeper, which is the interface allowing local cluster communicate to other clusters. Iguana.niees.group.cam.ac.uk is the hostname of where the globus gatekeeper is and also the head node of the PBS or central manager of condor. Jobmanager is using pbs now. If SForce is true, it means that files will be over written in SRB. SRBHome is the location where Scommands had been installed. Sdir is the logical file address in SRB where we put all the input data. Sget = \* will download all the input files from this place and Sput will upload all output files. Since the only output file of syntheticp2 is matrixo.txt, one can also change Sput = output file name. For instance, the output name could be matrixo-480.txt, matrio-520.txt etc. the number refers to different wavelength.

MCS will create three perl scripts. Pre.pl is used to copy all the input files from SRB to executable node. Post.pl is used to upload the output files back to SRB. The first condor job script is inp.job and it looks like the following;

```

*****
# inp.job written by my_condor_submit 1.0.2 - 27.03.06
Universe      = globus
Globusscheduler = iguana.niees.group.cam.ac.uk/jobmanager-fork
Executable    = pre.pl
Notification   = NEVER
GlobusRSL     = (arguments=/home/gtniees/1155901637)
Stream_Output = false
Stream_Error  = false
Output       = inp.out
Log          = inp.log
Error       = inp.err
Queue
*****

```

Basically, it creates temporary directory under user home directory. The hash number 1155901637 is the name of this temporary directory. Then, pre.pl will be executed to get all the input data from SRB. It uses fork as the globus jobmanager. The synthetic image program will then be submitted using second condor job description file (main.job) as show bellow.

```

*****
# main.job written by my_condor_submit 1.0.2 - 27.03.06
Universe      = globus
Globusscheduler = iguana.niees.group.cam.ac.uk/jobmanager-pbs
Executable    = /home/gtniees/1155901637/syntheticp2
Notification   = NEVER
GlobusRSL     = (stdout=/home/gtniees/1155901637/sig-
test.out)(job_type=mpi)(count=2)(directory=/home/gtniees/1155901637)
Transfer_Executable = false
Transfer_Output = false
Stream_Output  = false
Stream_Error   = false
Output       = /home/gtniees/1155901637/sig-test.out
Error       = job.err
Log        = job.log
Queue
*****

```

This is the main job running synthetcp2 on MPI enabled PBS cluster.

Finally, the out.job is used to copy output files back to SRB.

```

*****
# out.job written by my_condor_submit 1.0.2 - 27.03.06
Universe      = globus
Globusscheduler = iguana.niees.group.cam.ac.uk/jobmanager-fork
Executable    = post.pl
Notification   = NEVER
GlobusRSL     = (arguments=/home/gtniees/1155901637)
Stream_Output = false
Stream_Error  = false
Output       = out.out
Log          = out.log
Error       = out.err
Queue
*****

```

Since Computing Grid generated many synthetic image files for different bands or geographic locations, how to manage those images becomes a big issue. The benefit of using SRB is that one can access data which generated from distributed location all over the world as soon as you are the member of this VO.

When the jobs finished, the simulated reflectance values are automatically returned and put in the DataGrid environment, which is the SRB in our case. Users, can then access the data from anywhere around the world and running visualization tools such as GoogleEarth (Figure 7) to see the results.

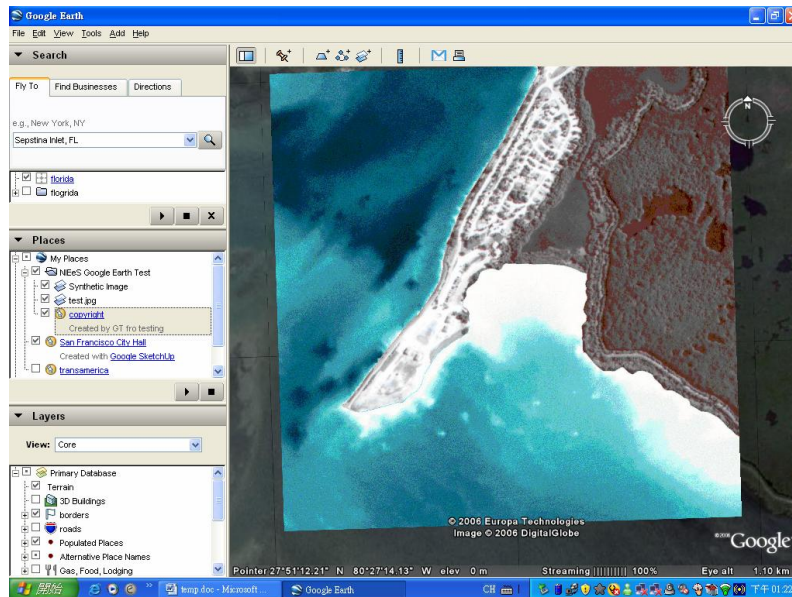


Fig. 7. The results of the synthetic images can be visualized in the client side using GoogleEarth or other GIS software.

#### 4. DISCUSSION AND SUMMARY

Many tools have been developed to help researchers work in collaborations with others across institution and country boundaries. Although both Computing and Data Grid tools are available, how to integrate them within different institutes is still a challenge. There is still a gap between middleware and environmental science applications. Our goal is to build an infrastructure for environmental science fields that is not limited to a specific application. We are filling the gap between Grid and Environment sciences especially remote sensing at the first stage. According to the previous section of this paper, currently, in order to run this synthetic image model, some code development was still required. Ideally, we hope most program can benefit from escience without doing much modification. The ideal situation is that any code, even MPI jobs can simply be executed in our infrastructure.

For the next step, we are evaluating the latest Condor version for new parallel universe and Condor-C. Also, submitting the GRID-MCHSIM to UK National Grid Services (NGS) to make sure it can be executed on a production GRID.

Moreover, we are evaluating OpenGIS software such as GRASS for image enhancement and GoogleEarth for visualization. This new approach would use FOX library. FOX is a set of Fortran libraries developed by eMinerals allowing one to deal with XML within Fortran program. Thus, KML files (GoogleEarth format) can be generated by MCHSIM. Then, GoogleEarth can access those synthetic images directly. In order to process hyperspectral remote sensing data in escience environment, we are also evaluating some open source tools and implementing them in this infrastructure. Scientific workflow tools, such as BPEL or Kepler, will also be implemented to improve the escience usage.

## REFERENCES

1. Bostater, C., Chiang, G., "Synthetic Image Generation of Shallow Waters Using a Parallelized Hyperspectral Monte Carlo & Analytical Radiative Transfer Model," concentrations," In: *Proceedings of the European Optical Society and SPIE — The International Society for Optical Engineering (EUROPTO), Remote Sensing 2002, Crete, Greece*, Vol 4880, 102-116 (2002).
2. Clery, D., "Can Grid Computing Help Us Work Together," *Science*. Vol 303, 433-434 (2006).
3. Dove, M., ..etl., "eScience usability: the eMinerals experience" In: *Proceedings of the UK e-Science All Hands Meeting 2005*, 30-37 (2005)
4. <http://www.cs.wisc.edu/condor/description.html>
5. <http://gridcafe.web.cern.ch/gridcafe/gridatwork/globus.html>
6. Bruin, R., "my condor submit v1.0.1 usage instructions", (2006)
7. <http://www.npaci.edu/dice/srb/whatisrb.html>
8. <http://www.accessgrid.org/>