An Integration of Several Technologies in the Architecture Definition and Deployment of a Geospatial Grid Web Portal

D. Mera[1], J. M. Cotos[1], J. Varela[1], C. Cotelo[2] and J. I. López[2]
[1]Systems Laboratory, Electronics and Computer Science Department. Technological Research Institute, University of Santiago de Compostela, Santiago de Compostela, Spain.
[2]Centro de Supercomputación de Galicia, Santiago de Compostela, Spain.

Abstract— *This paper shows the development of a virtual laboratory for the Ocean research community based on Grid technology, called Retelab. The main objective is to build a useful tool for the researchers where the necessary computer skills are minimum. The development has four phases. The user access and registration is the first phase where we have built a well balanced solution between simplicity and security based on a Role Access Management Model. The second phase is the Data Grid deployment focused on metadata and thought to store geospatial information. Tools to visualize and analyze the data are also available. Next part is the job submission. A metascheduler was integrated into Retelab and useful procedures were developed to use the stored data as job inputs and to add the job results into the Data Grid. The last phase is the implementation and deployment, of different geospatial applications for the Ocean research community.*

Keywords: Grid Computing, Grid Security, Data Grid, Metadata, Metascheduler, GridWay.

1. Introduction

The increasing number of research projects for Earth Observation due to the launching of new missions every year, with higher spatial and radiometric resolution, produces a large amount of geographical datasets. The analysis, storage and management of these data are often a very hard and tedious job, even unfeasible. For this reason it was necessary to find a uniform way to work, using Internet as a framework. An example of this is the Earth System Grid project [1] for climate study, whose main goal was to make a collaborative framework between centers, users, models and distributed data in the USA.

Specifically, the Ocean Study is an interdisciplinary work where biologists, physicists, meteorologists, oceanographers and computer engineers need to work together to process data in an efficient way.

Grid technology was born to allow the research community to undertake collaborative computation and share heterogeneous resources such as hardware (clusters, PCs, sensors...), software (Operating Systems, scientific software), and datasets, all of them connected through a network, i.e. Internet.

This paper shows the development of the Retelab project. Retelab can be understood as a distributed collaborative working environment which will constitute a virtual laboratory for multidisciplinary research projects related to oceanographic remote sensing [2]. Grid technology is the starting point of the project. We are trying to approach this technology to the ocean researchers. Nowadays, there are many Grid systems focused on different areas but most of them have the focal point in the technology and not in the end user. This project tries to provide a way to interact with a Grid system with needlessness of computer skills. The user does not need to know what kind of technology is under the system or his/her knowledge would be minimum. The main objective of Retelab is that researchers only need to be concerned with their works and not with the tools used in them. Retelab was developed following an iterative and incremental model where each phase was terminated with the implementation of a prototype to add new system functionalities.

The remainder of this paper is outlined as follows: In Section 2 the project is described explaining its main objectives and the different phases in which it was split to be developed. The prototype architectures, used technologies and characteristics about the software integration and the implementation process are shown in Section 3. Section 4 shows the future work, and conclusions are drawn in the last section.

2. Retelab Description

This project was structured in several phases and all of them were developed following various common objectives such as simplicity, friendly interface and the needlessness of computer skills to use it, and specific ones.

2.1 User Access and Registration

The Grid Security Infrastructure (GSI) is one of the main modules in a Grid system. GSI is focused on the Public Key Infrastructure (PKI). The only way to access a system based on PKI is with a digital certificate. The normal process to get a digital certificate takes too much time and it is not an easy task, specially when the user does not understand what is the meaning of PKI and digital certificate. On the other hand, it is also important to find a good solution to manage user access and permissions. The project needs a management model that allows to answer critical questions such as 'Who can have access?', 'What resources can they work with?' or 'What tasks can be done over a resource?'. A Role Based Access Control (RBAC) [3] model seems to be the most suitable management model to answer them.

The main goal of this phase is to look for a trade-off solution between simplicity and security that will allow users to get certificates and access the Grid system in an easier and more comfortable manner. Moreover, it is necessary to integrate the new system with a RBAC model, which will allow an easy management of users.

2.2 Data Grid

Most of the Grid systems are developed focusing on sharing computational resources. Nowadays, one of the main problems is the data storage and the analysis of the information that the projects produce. Some international research projects such as Worldwide LHC Computing Grid [4], will produce more than 15 Petabytes of data every year. Such a large amount of data is impossible to store or analyze using traditional storage data systems. Thus, it is clear that we need new ways to store the information using distributed systems. It would be desirable to have an easy way to analyze the results, to work with them, to search through large datasets and to share information with the research community.

The aim of this project phase is to develop a Data Grid and to deploy it in the Retelab system. This Data Grid system should allow the users to search in large datasets using metadata. Thus, we need a common metadata schema to describe geographic information. It should allow to visualize and analyze the data that the users have retrieved. The Retelab users should be able to edit the metadata adding or removing elements and sharing the results. The system must allow the users to download the data stored into Retelab. The users would be able to upload new data and label them with data attributes. If a user searchs in the Data Grid, the retrieved results should be used as parameters in the Grid jobs and the outputs generated by the jobs should be used as new entries into the virtual Database.

2.3 Job Submission and Monitoring

There are many systems that allow users to submit jobs to Grid systems, but they need some kind of interaction with the end user. This interaction decreases simplicity and transparency to the process and this portal, as we said before, is designed for researchers without experience in Grid systems. Thus, it is necessary to incorporate a decision making tool to increase transparency. A tool that selects the best option in accordance with the existing policies.

The objective is to develop a system which allows the researchers to configure, execute and monitor their jobs, and retrieve results.

2.4 Testbed Applications

The last phase in the Retelab project is to deploy several test applications such as the Regional Ocean Model System (ROMS) [5], [6] and "Sentinazos" [7]. These testbeds will enable both to test the project, and to get useful software that the oceanographers would be able to use.

3. Retelab Design and Implementation

Retelab was developed as a web portal based on portlets. Portlets [8] are pluggable user interface components that are managed and displayed into a web portal. The portlets are very useful in Grid systems because they allow a high interoperability and reuse and they can be developed and added to the web portal on demand. GridSphere has been chosen as a portlet container for Retelab. GridSphere [9] is a portal framework that provides an open source, portlet based web portal and enables developers to quickly develop and package third party portlet web applications. We have integrated some portlets developed by Open Grid Computing Environment (OGCE) [10] to test our system, such as the job submit portlet or the proxy-manager portlet. The Globus Toolkit [11], specifically its version 4 (GT4), was used to develop the Grid system.

As the previous section has shown, Retelab has been developed in four main phases and each phase was finished with the implementation of a functional prototype. The following subsections describe the prototype architectures, used technologies and details about the implementation.

3.1 User Access and Registration

Fig. 1 shows the different modules and technologies integrated to develop this prototype. The main goal was to find a well balanced solution between simplicity and security. This solution is based on a RBAC model to manage the portal user permissions.

There are three possible scenarios to access the system. In the first scenario, the user is not registered. Thus, he/she must use the Registration Portlet to do it. The Registration Portlet is based on the PURSE system, specifically, a PURSE portlet version [12]. PURSE [1] is an integrated solution that provides an easy-to-use web interface for potential users of an application to register themselves and request sign-in credentials. The PURSE system can get user digital certificates. We have improved PURSE to support the role paradigm and also generate X.509 Attribute Certificates (AC) for each user with his/her system roles. The ACs are stored into a LDAP, whereas the digital certificates are stored into MyProxy [13], which is an online credential repository.

In the second scenario the user is already registered. Thus, he/she can access the system via the User Access Portlet, using the login and password selected when he/she was registered. Once the user is authenticated the system retrieves

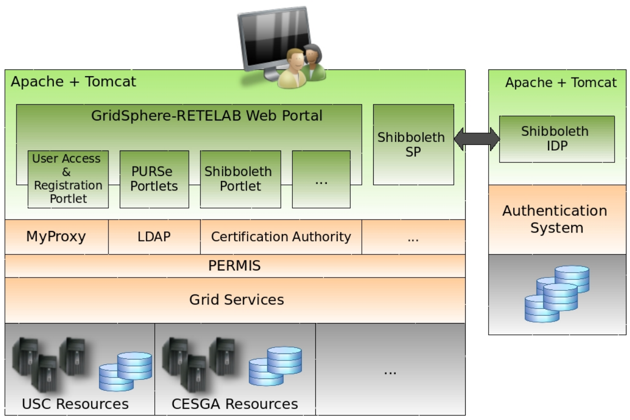

Fig. 1: User Access and Register Architecture.

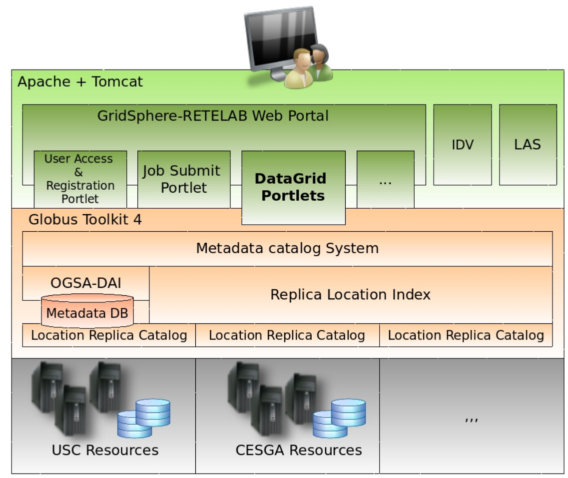

Fig. 2: Data Grid Architecture.

the user credentials, the digital and the attribute certificates, which were generated in the registration phase and stored respectively into MyProxy and into the LDAP. The user credentials allow him/her to use the Grid system according to his/her role. PERMIS [14] was the software used to support the role paradigm in the Grid System, using the ACs.

The last scenario is the access of Shibboleth users. Shibboleth [15] is a software that provides Web Single SignOn (SSO) across a reliable network of partners. The integration of Shibboleth with the Retelab system has as starting point an integration of Shibboleth with Gridsphere developed by Meta Access Management System (MAMS) [16]. We have improved this version to automatically add new users into Retelab generating their ACs and digital certificates and adding their personal information to the system. Using this improved version, when one of the researchers at a partner center wants to access the system, Retelab will just need the confirmation of the user center. The user would use the Shibboleth portlet to access the system. The Shibboleth system may redirect the user to his/her center where he/she would be authenticated and his/her attributes (name, email, role, etc.) would be sent to the Retelab system. Retelab would use the user attributes to generate the user credentials according to the received information. These credentials would be used in the Grid and web Retelab systems.

3.2 Data Grid

Fig. 2 shows the architecture of the Data Grid prototype. We have developed a portlet as a user interface to access the Data Grid. Data are stored in the system resources and the Replica Location Service (RLS), which is part of GT4, is used to store the specific data locations. The RLS has two subsystems, the Location Replica Catalog (LRC) and the Replica Location Index (RLI). The LRC maintains a mapping between the Logical File Names (LFN) and the real data paths or Physical File Names (PFN). The RLI maps each LFN onto the LRC that has mapped the PFN. Our system has a LRC in each resource that is part of the Data Grid, and only one RLI. The LRCs would send messages to the RLI to inform about the resources they have. As we said before, our system must be focused on the metadata. The Metadata Catalog System (MCS) [17] allows us to map each LFN on a set of metadata to identify each resource. The MCS uses the OGSA-DAI [18] Grid service to manage its mapping information, which is stored in a MySQL Database. We need a metadata schema to describe geographic information, which will be common to all Retelab users. We decided to use one of the standards for Geospatial metadata, the ISO 19115 [19].

Data visualization can be done in two ways. In the first way, we have developed a procedure to automatically make a link for each retrieved result which allows to visualize it. This link downloads the Integrated Data Viewer application [20] and it is installed in the user computer using java web start. This installation is automatic and transparent to the user. Once the IDV is installed, it is directly loaded with the previously selected data. This procedure is showed in Fig. 3. In the second way, Retelab has a Live Access Server (LAS) [21] that allows to visualize datasets. This LAS is deployed in the web server and it is accessible via a portlet.

Once the user has done a search, it is also possible to use the data results as parameters to the Grid jobs. To achieve this, the user should make a simple click on each datum that he/she wants to use as parameter and it will be added to the job submission template. After the job execution is completed, the user should be able to use the job outputs as new entries in the Data Grid adding appropriate metadata for them. This new data would be shared with the rest of the research community. Next section has a more detailed explanation about the job submissions.

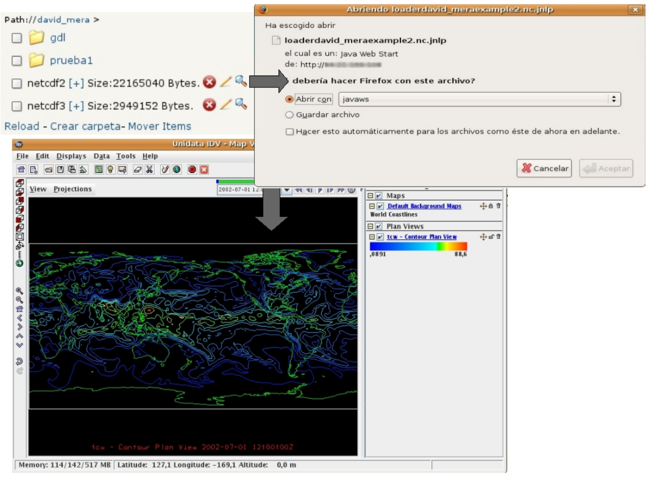

Fig. 3: Example of the IDV Visualization.

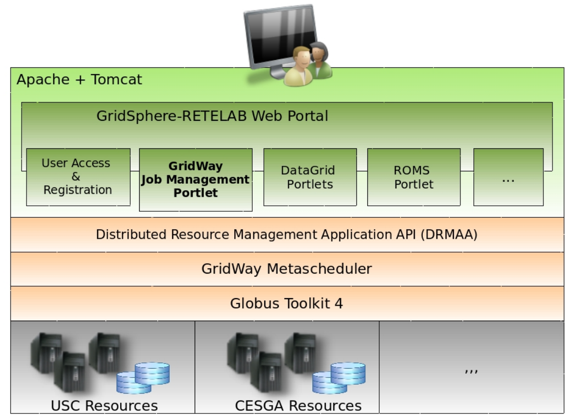

Fig. 4: Job Management Architecture.

3.3 Job Submission and Monitoring

The OGCE package provides a job submission portlet compatible with GT4. It allows users to send jobs to the Grid, but it forces them to select resources from a list. Because the portal is intended for researchers without experience in Grid systems, Retelab has incorporated a Grid metascheduler to make this kind of decisions on behalf of user. Grid metaschedulers facilitate optimal utilization of the Grid resources and they undertake the tasks for resource discovery, job scheduling, execution, monitoring and job output retrieval. Linking the portal with a metascheduler has the advantage of keeping the users free to focus on their jobs, leaving the complexity of resource allocation and scheduling to the broker.

In this phase a job submission and monitoring portlet has been developed. This portlet interacts with the GridWay [22] metascheduler using the Distributed Resource Management Application API (DRMAA) [23] OGF standard. GridWay metascheduler was selected because it provides interoperability with different Grid middlewares, supports DRMAA OGF standard and allows jobs submission using the Job Submission Description Language (JSDL) [24].

Fig. 4 shows the design and GridWay integration with the Retelab system.

We found an important barrier related to the management of proxy credentials in the GridWay environment. As mentioned, Retelab access is based on roles that each user has in the AC, so the laboratory has been designed in order that only one user account is needed for Grid activities. Thus, different users are distinguished by their proxies. Because GridWay was designed for single users, it uses the same Unix process and the same proxy for all the jobs submitted from the same local account. This represents a major obstacle for our approach because we need to submit different jobs linked to different proxies using the same local account.

A first approach was a version modified by the ARCS-GridWay project for the Australian Research Collaboration Service. This solution is close to our needs but it has some limitations. One of the most important limitations is that this version is not compatible with DRMAA standard.

The suitable solution would be a new version of GridWay including modifications around the proxy management while maintaining the compatibility with the DRMAA standard. Finally, we have worked with the GridWay team in order to develop this new version.

As a result, the job submission and monitoring portlet has been deployed in the Retelab portal integrating the new GridWay version . Using the MyProxy portlet, each user can load different proxies from MyProxy servers. If several proxies have been loaded the user decides which one to use by default. The first step when a job is submitted is to retrieve the job parameters defined by the user and to verify which is the default proxy. In the second step, the job submission portlet launches a new system process, which executes a DRMAA java class. This java class is responsible for creating the suitable job template and for submitting it to GridWay. The key issue is that all the processes run under the same local account but each one has its own X509_USER_PROXY variable containing the suitable proxy. After a job is successfully submitted, the job information is stored in a MySQL database. Each submission stores: user DN, name of the portal user, GridWay job id, task name and results directory. GridWay job id is needed for retrieving the job state using DRMAA methods. The job state merged with the job information in the database is used to build the monitoring output. Only information about jobs owned by the registered user is shown in the monitor.

4. Future Work

Once the system is finished, it will be necessary to test it and improve it. To achieve this, we want to develop and deploy some testbed applications in Retelab. These applications were born as a necessity for some of the research groups that are working with us.

There are two main applications to develop. The first specific application is ROMS (Regional Ocean Model System), a new generation model for oceanic circulation specially designed to simulate regional oceanic systems with great accuracy (for example the Galician coast). ROMS is an application already developed, which we want to improve and deploy into Retelab. ROMS implements both MPI and OpenMP parallelization methods [25]. Retelab architecture and the domain decomposition of the MPI version will allow to execute it in a hybrid model (openMP+MPI) [6]. To define ROMS jobs and visualize the results a specific portlet is currently being developed. Jobs will be submitted to the Grid using the previous GridWay portlet development. Following the standards, this submission will be done using JSDL. This testbed is being developed in two phases. The current and first phase is the ROMS portlet development where the user will be responsible to select the application version (serial, MPI or OpenMP) during the job requirements configuration. In a second phase, the metascheduler would be the responsible to decide the best method to run ROMS according to the available resources in each moment and the requirements of the jobs measured by the portlet.

"Sentinazos" project will be the second application to be developed and deployed in Retelab. "Sentinazos" is a project in an early phase of development that will allow the automatic detection of fuel disposal based on satellite and airborne images. Once the images are obtained, different algorithms based on image processing, neuronal networks and fuzzy logic will be applied to detect, monitor and prevent ecologic and economic disasters. We also propose the integration of the information into an interoperable Geographic Information System (GIS) that will help users to easily manage and organize the data. Currently, we are working in collaboration with another research group in the development and testing of the algorithms and techniques, which need a previous training period. Grid migration will be faced in a near future.

5. Conclusions

In this paper we have presented the development of a virtual laboratory thought for the Ocean Research Community. The integration and tuning of different kind of technologies has allowed us to make an easy-to-use Grid based system, where the computer skills are not necessary or they are minimum, without neglecting the security. Standards and open source software were used whenever possible. This milestone was possible due to the collaboration of an heterogeneous set of partners such as the University of Santiago de Compostela, Supercomputing Center of Galicia and different Ocean research centers.

Acknowledgement

The authors wish to thank the Ministerio de Educación y Ciencia of Spain (ESP2006-13778-C04) for finantial support.

References

[1] D. Bernholdt, S. Bharathi, D. Brown, K. Chanchio, M. Chen, A. Chervenak, L. Cinquini, B. Drach, Foster, P. Fox, J. Garcia, C. Kesselman, R. Markel, D. Middleton, V. Nefedova, L. Pouchard, A. Shoshani, A. Sim, G. Strand, D. Williams, "The Earth System GRID: Supporting the next generation of climate modeling research", in *Proc. IEEE'05*, 2005, vol. 93(3), pp. 485–495.

[2] D. Mera, J. M. Cotos, J. R. R. Viqueira and J. Varela, "A User Management Web System Based on Portlets for a Grid Environment Integrating Shibboleth, PURSe, PERMIS and Gridsphere", in *Proc. DCAI'08*, 2008, pp. 38–46.

[3] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman, "Role-based access control models", *Computer*, vol. 29(2), pp. 38–47, Feb. 1996.

[4] J. Shiers, "The Worldwide LHC Computing Grid (worldwide LCG)", in *Proc. CCP'06*, 2006, pp.219–223.

[5] A. F. Shchepetkin and J. C. McWilliams, "The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model", *Ocean Modelling*, vol. 9(4), pp.347–404, Aug. 2004.

[6] C. Cotelo, A. Gómez, I. López and J. M. Cotos. "Adapting ROMS to Execute on GRID Using a Hybrid Parallelization Model", in *Proc. ADVCOMP'08*, 2008, pp. 57-62.

[7] The Sentinazos website. [Online]. Available: http://www.sentinazos.org

[8] A. Abdelnur, S. Hepper. (2003). "Java TM Portlet Specification. Version 1.0". [Online]. Available: http://www.jcp.org/en/jsr/detail?id=168.

[9] J. Novotny, M. Russell, O. Wehrens, "GridSphere: a portal framework for building collaborations", *Concurrency and Computation: Practice & Experience*, vol. 16(5), pp. 503–513, Apr. 2004.

[10] J. Alameda, M. Christie, G. Fox, J. Futrelle, D. Gannon, M. Hategan, G. Kandaswamy, G. von Laszewski, M. A. Nacar, M. Pierce, E. Roberts, C. Severance and M. Thomas. "The Open Grid Computing Environments collaboration: portlets and services for science gateway". *Concurrency and Computation: Practice and Experience*, vol. 19(6), pp. 921–942, Apr. 2007.

[11] I. Foster and C. Kesselman, "GLOBUS: a metacomputing infrastructure toolkit", *International Journal of Supercomputing Applications and High Performance Computing*, vol. 11(2), pp.115–128, 1997.

[12] The PURSE Portlets website. [Online]. Available: *http://www.extreme.indiana.edu/portals/purse-portlets*

[13] J. Novotny, S. Tuecke, V. Welch, "An Online Credential Repository for the Grid: MyProxy", in *Proc. HDPC-10*, 2001, pp. 104–111.

[14] D. W. Chadwick and A. Otenko "The PERMIS X.509 role based privilege management infrastructure", *Future Generation Computer Systems*, vol. 19(2), pp. 277–289, Feb. 2003.

[15] M. Erdos, S. Cantor. (2002). "Shibboleth-Architecture DRAFT v05". [Online]. Available: http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-arch-v05.pdf

[16] The Meta Access Management System website. [Online]. Available: https://mams.melcoe.mq.edu.au/zope/mams

[17] G. Singh , S. Bharathi , A. Chervenak , E. Deelman , C. Kesselman , M. Manohar , S. Patil , L. Pearlman," A Metadata Catalog Service for Data Intensive Applications", in *Proc. SC'03*, 2003, p. 33.

[18] M. Antonioletti, M. Atkinson, R. Baxter, A. Borley, N. P. Chue Hong, B. Collins, N. Hardman, A. Hume, A. Knox, M. Jackson, A. Krause, S. Laws, J. Magowan, N. W. Paton, D. Pearson, T. Sugden, P. Watson and M. Westhead, "The Design and Implementation of Grid Database Services in OGSA-DAI", *Concurrency and Computation: Practice and Experience*, vol. 17, pp.357-376, 2005.

[19] Geographic Information - Metadata.International Organization for Standardization ISO 19115 (E) 1st edition, Geneva, Switzerland, 2003.

[20] The Integrated Data Viewer website. [Online]. Available: http://www.unidata.ucar.edu/software/idv

[21] S. Hankin, J. Davison, J. Callahan, D. E. Harrison, K. O'Brien. "A configurable web server for gridded data: a framework for collaboration", in *Proc. IIPS-14*, 1998, pp.417-418.

[22] E. Huedo, R. S. Montero, and I. M. LLorente, "A framework for Adaptative Execution on Grids", *Software - Practice and Experience*, vol. 34, pp. 631–651, 2004.

[23] H. Rajic, R. Brobst, W. Chan, F. Ferstl, J. Gardiner, A. Haas, B. Nitzberg, H. Rajic, D. Templeton and P. Tröger, (2009), "Distributed Resource Management Application API Specification 1.0". [Online]. Available: http://www.ogf.org/documents/GFD.115.pdf

[24] A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S-McGough, D. Pulsipher, and A. Savva (ed.), "Job Submission Description Language (JSDL) Specification,Version 1.0.", (2005). [Online]. Available: http://www.ogf.org/documents/GFD.56.pdf

[25] P. Wang, Y. Tony Song, Y. Chao and H. Zhang , "Parallel Computation of the Regional Ocean Modeling System". *International Journal of High Performance Computing Applications*, vol. 19(4), pp. 375–385, 2005.