A GENERAL AND FLEXIBLE FRAMEWORK FOR VIRTUAL ORGANIZATION APPLICATION TESTS IN A GRID SYSTEM

T. Coviello[†§], G. Donvito[†], G. Maggi[†], A. Pierro[†]
[†]Dipartimento Interateneo di Fisica & INFN di Bari - ITALY
[§]Dipartimento di Elettrotecnica ed Elettronica - Politecnico di Bari - ITALY

Abstract

A grid system is a set of heterogeneous computational and storage resources, distributed on a large geographic scale, which belong to different administrative domains and serve several different scientific communities named Virtual Organizations (VOs). A virtual organization is a group of people or institutions which collaborate to achieve common objectives. Therefore such system has to guarantee the coexistence of different VO's applications providing them the suitable run-time environment. Hence tools are needed both at local and central level for testing and detecting eventually bad software configuration on a grid site. In this paper we present a web based tool which permits to a Grid Operational Center (GOC) or a Site Manager to test a grid site from the VO viewpoint. The aim is to create a central repository for collecting both existing and emerging VO tests. Each VO test may include one ore more specific application tests, and each test could include one ore more subtests, arranged in a hierarchic structure. A general and flexible framework is presented capable to include VO tests straightforwardly by means of a description file. Submission of a bunch of tests to a particular grid site is made available through a web portal. On the same portal, past and current results and logs can be browsed.

INTRODUCTION

Grid computing is emerging as a new key enabling infrastructure for resource sharing and coordinated problem solving in dynamic multi-institutional virtual organisations [1]. A virtual organisation is a group of people or institutions which collaborate to achieve common objectives. Therefore a grid system has to guarantee the coexistence of different VO's applications providing them the suitable run-time environment. Hence tools are needed both at local and central level for testing and detecting eventually bad software configuration on a grid site. Different grid projects, such as LCG [2], contemplate some activity for obtaining tools able to detect and rectify promptly grid component failures. So the main aim of these activities is to evolve toward a full set of tests to provide system level regression testing on grid middleware. Testing activity is considered a mechanism by which deployed middleware can be made robust, predictable and supportable. A test suite which is the natural result of a testing project, is a software package that includes different kind of tests. Such tests can be grouped into four levels:

- 1st Level - In this level are grouped all tests for checking the installation and configuration of the machines and services of a site. Some of the typical verifications, performed at this level, concern the actual status of running services such as network services (NFS, AFS, NTP), checking whether the necessary network communication ports are opened or filtered by a firewall, etc.
- 2nd Level - The tests that belong to this level have the goal to verify the correct working of the basic functionalities such as job submission, output result retrieving, data transfer, etc.
- 3rd Level - At this level the grid system is seen from a more global point of view. In fact the tests of this level provide information regarding the performance of whole testbed. In particular they focus their attention at the load of network, computational and storage resources. The main consumer of this information is the Grid Operational Centre, which can track statistical distributions and discover the critical components of the system. This is useful looking at the future improvements of the grid performance and reliability.
- 4th Level - Finally, at this level we find application-oriented tests. Application running failures are more difficult to detect and fix, because they are not imputable to a failure of a grid service or component. In fact they could be caused by bad running environment or by a bad application software configuration or installation. A test, specific for the particular application, is the unique way to tackle and solve this kind of problems.

Grid computing can provide a common platform for a wide variety of scientific and industrial applications. Therefore it is not possible to have a general and fixed application test suite suitable for every kind of application.

In this paper we present a web based tool which permits to a Grid Operational Centre (GOC) or a Site Manager to test a grid site from the VO viewpoint. This tool, called GridAT (which is the acronym for Grid Application Test), is the result of a project activity coordinated within the INFN organization. The aim is to provide a framework for collecting both existing and emerging VO tests. Each VO test

may include one ore more specific application tests, and each test may include one ore more subtests, arranged in a hierarchic structure. A general and flexible framework is presented capable to include VO tests straightforwardly by means of a description file. Submission of a bunch of tests to a particular grid site is made available through a web portal. On the same portal, past and current results and logs can be browsed. In the following section GridAT architecture is presented. Details are also described about the two GridAT software module and GridAT working.

GRIDAT ARCHITECTURE

GridAT project results from the idea of providing a framework for including straightforwardly any application grid test. GridAT has the goal to promote grid enabled applications and then to tackle all questions related to problems of execution and configuration which could take place in a grid environment. It is a general and flexible framework which can provide the possibility to test a wide domain of applications. The overall GridAT architecture is depicted in Fig. 1 GridAT is composed by two modules:

1. the grid side module
2. the web side module

The first module is responsible for the whole operation about test management, test validation and result storing. The second module provides a web interface by which it is possible to submit a test, view test results, plot summary graphs etc.

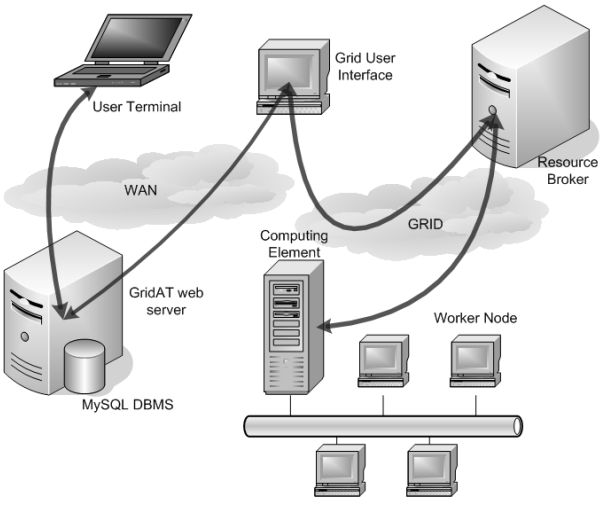


Figure 1: GridAT architecture

In the same figure a sketch of the grid reference architecture is shown. Grid computational resources are available as Computing Elements (CE). Each computing element delivers jobs to the Worker Nodes (WN), which will perform the real work. The Computing Element provides an interface to the local batch queuing system and can manage one or more Worker Nodes [4]. The Resource Broker (RB) plays the role of a super-scheduler. Its main activities are resource selection, mapping and scheduling [5]. The component, which allows users to interact with the grid system, is the User Interface (UI). User interface permits the submission, the status monitoring and the output result retrieval of jobs. The following subsections will give a more detailed description about these two modules.

The grid side module

Current GridAT implementation works on LCG middleware. The system core is the grid side module which run on a user interface and is responsible for the two main tasks:

1. Test execution;
2. Test validation.

When a test, that may be composed by multiple jobs, is submitted, a new process is created. At runtime, for each job a new object is instantiated which encapsulates job status. Job execution includes the whole operations, normally performed by a grid user:

1. Job submission;
2. Checking status;
3. Output retrivial.

Test validation task is performed by comparing job output results against the expected output result. Different kind of comparisons may be implemented according to the application under test. It would be impossible to perform a full comparison between actual and expected output for any kind of application. In fact the application could provide a different output which may be a function of a random variable unknown before job execution. Nevertheless output may present some tokens or text message which could be used to probe job success or failure. Test is passed if every job result is accordant to the expected output. Both in case of success or failure, test result and single job logs are stored in a relational database. The insertion of new tests is a straightforwardly operation. Building a new test starts from the collection of all jobs which will be included. For each job, grouped in a test, it is necessary to provide:

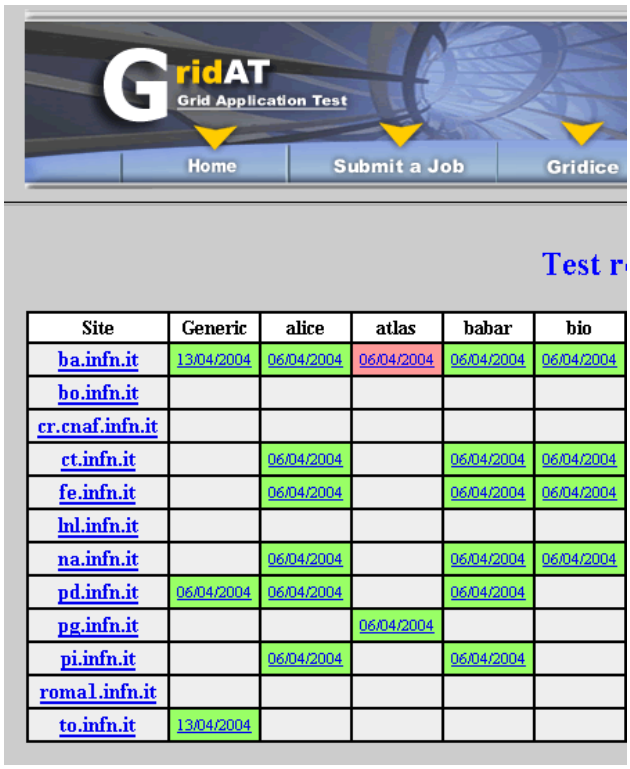
1. the Job Description Language file that contains a formal description of job executable, input data, output date, requirements and rank [3].
2. all the data, script and executable files neccessary for job execution, in the case they are localized on the user interface. If executable files are localized and installed on the target resource and data files are localized on a whatever grid resource, they must be specified in the jdl file.
3. a test description file, which arranges jobs and formalizes the validation rules for each job.

Test organization is possible through a description file. For this purpose XML formal language has been used [7]. Parsing that file, the core module has the possibility to extract important information about test as for example, the number of jobs which compose the test and the rules that must be applied to validate each job.

The web side module

The web software module is written in php [6] and produces dynamic web pages in which current and past test results can be browsed.

The initial page gives an instantaneous overview of grid testbed (Fig. 2). The main table displays the results and the date of the last submitted test ordered by grid site and VO. In the grid site column, a link is available for each site which gives a summary pie graph. That graph shows the total percentage of the successful and unsuccessful tests for all supported VOs by that site.



Site	Generic	alice	atlas	babar	bio
ba.infn.it	13/04/2004	06/04/2004	06/04/2004	06/04/2004	06/04/2004
bo.infn.it					
cr.cnaf.infn.it					
ct.infn.it		06/04/2004		06/04/2004	06/04/2004
fe.infn.it		06/04/2004		06/04/2004	06/04/2004
lnl.infn.it					
na.infn.it		06/04/2004		06/04/2004	06/04/2004
pd.infn.it	06/04/2004	06/04/2004		06/04/2004	
pg.infn.it			06/04/2004		
pi.infn.it		06/04/2004		06/04/2004	
roma1.infn.it					
to.infn.it	13/04/2004				

Figure 2: GridAT home page

For viewing previous test results about a VO test submitted to a particular site, a link is also available clicking on the date of the last test in the main table. More details about each submitted test is available. This is particularly useful for debug purpose and recovery activity. Another important functionality of GridAT web portal is the on-line test submission. As shown in Fig. 3, user must select the grid resource to test and the VO.

After VO selection, user can choose among the available test for that particular VO. The number of test submission

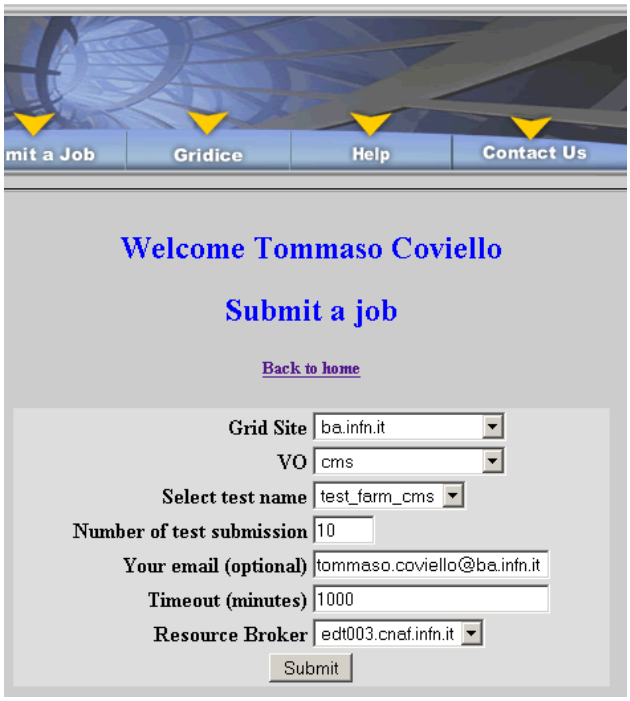


Figure 3: GridAT job submission web page.

indicates how many time to repeat the single job submission in order to saturate worker node queue. In fact LCG middleware does not contemplate the possibility to select a particular worker node but only the computing element. User can optionally provide his e-mail to receive a notification when test is finished. Timeout can also be set in order to cancel pending or looping job after that time. Finally resource broker can be selected. After test submission a java applet shows in real time the log of submitted test running on the user interface. Test submission web page access is restricted only to those users who have a valid certificate issued regularly by the INFN Certification Authority.

CONCLUSION

In this paper GridAT framework and its architecture has been presented. GridAT software modules and its working has been also described. Current activity is evolving towards the specification of a more general XML description file for a more flexible and adaptable inclusion of different application tests.

ACKNOWLEDGEMENTS

We would like to thank all the CMS and INFN collaboration.

REFERENCES

[1] I. Foster, C. Kesselman, S. Tuecke: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Interna-

tional Journal of High Performance Computing Applications.

[2] LHC Grid Computing Project http://lcg.web.cern.ch/LCG/

[3] F. Pacini: JDL Attributes http://server11.infn.it/workload-grid/documents.html, 2003

[4] A.Ghiselli:DataGrid Prototype 1. http://web.datagrid.cnr.it/pls/portal30/docs/3180.pdf

[5] DataGrid-D1.2: http://server11.infn.it/workload-grid/docs/DataGrid-01-D1.2-0112-0-3.pdf: WP1: Workload Management (September 2001).

[6] http://www.php.net/

[7] http://www.w3.org/XML/