Journal of Physics: Conference Series **119** (2008) 062023 doi:10.1088/1742-6596/119/6/062023

Enabling a priority-based fair share in the EGEE infrastructure

D Cesini, V Ciaschini, D Dongiovanni, A Ferraro, A Forti, A Ghiselli, A Italiano, D Salomoni
INFN-CNAF, Bologna, Italy

Abstract. While starting to use the Grid in production, applications have begun to request the implementation of complex policies regarding the use of resources. Some Virtual Organizations (VOs) want to divide their users in different priority brackets and classify the resources in different classes, others instead do not need advanced setups and are satisfied in considering all users and resources equal. Resource managers have to work for enabling these requirements on their site, in addition to the work necessary to implement policies regarding the use of their resources, to ensure compliance with Acceptable Use Policies.

These requirements end up prescribing the existence of a security framework not only capable to satisfy them, but that must also be scalable and flexible enough in order to do not need continuous and unnecessary low-level tweaking of the configuration setup every time the requirements change. Any security framework implementing these priorities should not require constant tweaking by site administrators.

Here we will describe in detail the layout used in several Italian sites of the EGEE (Enabling Grid for E-sciencE) infrastructure to deal with these requirements, along with a complete rationale of our choices, with the intent of clarifying what issues an administrator may run into when dealing with priority requirements, and what common pitfalls should be avoided at any cost.

Beyond the feedback on interfaces for policy management, from VO and site administrators, we will especially report on the aspects coming from the mapping of Grid level policies to local computing resource authorization mechanisms at Grid sites and how they interfere from a management and security point of view.

1. Introduction

While Grid usage is becoming more widespread, Virtual Organizations (VO) [1] are constantly increasing in size and internal structure complexity. So the naïve strategy of executing all jobs with the same user priority is not satisfactory anymore, since it would allow any VO user to overload the Grid resources and thus compromising others work. This is a strong limitation considering that VOs have to work in a collaborative multi-domain trust environment with no direct control of the distributed resources. Therefore VOs and resource providers should take into account a more complex scenario with the possibility for VO users to submit jobs with priorities reflecting their roles or group membership.

To achieve such goal the middleware of the Grid should evolve its authorization mechanisms in a more flexible way rather than guaranteeing simple access controls to the resources; indeed the middleware should provide a suitable way to guarantee a concrete fair share allocation for jobs submitted by different groups of users.

Remarkable efforts were carried on by the EGEE Job Priority Working Group and huge improvements towards suitable authorization standards were studied inside OGSA [2] and OASIS [3] so authorization for intra-VO fair share still raises several open issues and several approaches are being evaluated in current Grid environments in order to improve the balance of jobs among different VO users.

Remarkable efforts were carried on by the EGEE Job Priority Working Group and several approaches are being evaluated in current EGEE Grid middleware in order to improve the balance of jobs among different VO users. Moreover it is worth mentioning the huge improvements towards suitable authorization standards inside OGSA [2] and OASIS [3].

In this article we present a strategy founded on G-PBox [4] policy engine. Our approach leverage on G-PBox rich authorization features in order to enforce policies that associate VO groups and roles to different abstract service classes with different priority classifications. Such policies will be read and enforced by the Grid services and resources involved during a job submission process started by every VO user.

We underline that the analysis, design, implementation and tests described in this paper were performed by INFN [5] staff inside the EGEE [6] Project funded by the European Commission.

In Section 2 we will describe our strategy to face the intra-VO fair share challenge. In Section 3 we will describe the G-PBox policy framework used for verify our approach. In Section 4 we will report the results of a series of preliminary tests, while Section 5 will summarize our conclusion.

2. Priority-based fair share

2.1. Description of the problem

Recent years have witnessed the evolution of various approaches in the field of fair share in the Grid, however the current Grid production environments still lack the flexibility required for a large scale and a dynamic resource sharing, where Virtual Organizations and resources cohabit in the same environment based on a set of agreements and collaborations.

In current production Grid infrastructures information of fair share of the sites is not externally visible and there is no simple way for a VO to have direct control on such settings for its own groups of users. This fact represents an obstacle to the intra-VO fair share of resources, especially for those VOs having a huge number of members classified in many groups and roles.

2.2. Our approach

In this section, we describe our approach for enabling a concrete allocation of differentiated resource shares to different groups or roles of users of a VO.

In [7] we proposed a theoretical solution for fair share based on the concept of service classes characterizing attributes that describe different quality of service levels. Examples of parameters characterizing a service class are the target share for the utilization of the site resources or policies related to the maximum walltime of a single job or a priority level. The rigorous definition of each service class can be considered as a contract among a VO and a resource provider. The resource provider will configure its computing resource capabilities (based on its local resource management system) according to the contract with the VO. That approach relies on the requirement to discover service class information. Such requirement obliged all sites (also for testing purposes) to publish to the Information System new service classes attributes not endorsed by the current Glue Schema [8] specification.

The current approach relaxes constraints with respect to [7] in the fact that it uses service classes concepts without requiring the service class attribute in the current Glue Schema. Indeed we underline that the described work relies on the existing Grid layout used in production sites of the EGEE infrastructure in order to prove its possible first adoption without dramatic changes.

The Grid middleware should take into account the following general requirements to deal with the proposed priority-based fair share: (1) using VOMS attributes for Grid users, (2) managing

and enforcing a set of authorization policies based on VOMS attributes and GLUE resources attributes.

The first requirement is the management of privilege attributes associated to users. In particular, we refer to the concepts of groups and roles that are currently provided by the Virtual Organization Membership Service [9] as VOMS users attributes.

The second requirement is to provide an authorization mechanism for Workload Management System (WMS) [10] of the VOs and CEs of the sites allowing them to enforce a set of policies based on users attributes and resources attributes. We used G-PBox facilities to set and enforce suitable authorization statements for WMSes and CEs regarding Grid users (with specific VOMS attributes) and Grid resources (with specific not-published service classes or AccessControlBaseRule GLUE attributes). G-PBox is an authorization architecture grounded on a set of Policy Decision Points (PDP) communicating among each other and managed by VO managers (VO G-PBox servers) and Site managers (Site G-PBox servers). During the administrative phase G-PBox offers facilities to create, manage, distribute, accept and reject XACML [11] policies. During the runtime phase G-PBox acts as a policy decision point accepting authorization requests from Grid services and resources.

It is also needed an agreement between the VOs and the resource providers about values of existent GLUE attributes describing the queues. Such attribute values characterize queues with different quality of service levels. The site will publish the attribute values to the information system and configure local queues according to such values. As an initial testing layout, we propose to use the existing AccessControlBaseRule attribute published by current EGEE Computing Elements(CE). Indeed currently the selection of a suitable queue for a job by the the WMS is done through the matchmaking process taking also into account authorization attributes, taken from the VOMS proxy extensions on the users' side and from the AccessControlBaseRule attribute on the resource side. We underline that the usage of AccessControlBaseRule GLUE attribute during our current work is due to our choice to preserve the EGEE production service. For an easier implementation of our approach the ideal scenario would require a new GLUE attribute (e.g. ServiceClass, defining a target share for the utilization of the site resources), but as we show in the remainder of this article this is not mandatory.

All the requirements described above have been considered in a our first prototype that will be described in the next section.

2.3. Our approach setup

In this section, we describe an INFN prototype for evaluating the feasibility of the proposed approach for production environments. The prototype has been developed and deployed by using the facilities provided by the INFN infrastructure. The key middleware components that have been involved are: a VO VOMS server used for the creation and management of privilege attributes associated to the VO users; a VO G-PBox server for the management and enforcement of the VO policies; a Site G-PBox server (one for each site) receiving policies (to be accepted) from the VO G-PBox server; a gLite WMS asking to the VO G-PBox server policies regarding suitable CEs with the proper AccessControlBaseRule attributes; some LCG CEs configured to ask (through a LCAS/LCMAPS plugin) to the Site G-PBox policies regarding mapping information for the user submitting the job.

The VO manager is responsible for the following actions: using the VO VOMS to set VO groups and roles, using the VO G-PBox to define routing policies (associating VOMS attributes to AccessControlBaseRule attribute values) useful for WMS, using the VO G-PBox to define high-level mapping policies (associating VOMS user groups/roles with not-published service classes) to be send to Site G-PBoxes. The Site manager is responsible for: configuring the Local Resource Management System (LRMS) in accordance with both the published AccessControlBaseRule attribute values and the not-published service classes values, using the

Site G-PBox to define low-level mapping policies (associating not-published service classes with real UNIX pool accounts), accepting (or rejecting) high-level mapping policies from VO G-PBoxes.

We want to state that a VO G-PBox is the essential component for two VO administrative tasks:

- to create routing policies for VO WMSes
- to create high-level mapping policies and send them to Site G-PBoxes

The Site G-PBox is the essential component for two site administrative tasks:

- to create low-level mapping policies
- to accept (or to reject) high-level mapping received from the VO G-PBoxes

On the WMS side the matchmaking process interacts with the VO G-PBox in order to know which are the CEs that are assigned to the submitting user based on his/her VOMS credentials and AccessControlBaseRule values published of CEs.

On the CE side, the CE authorization layer interacts with the Site G-PBox in order to know how the user should be mapped to the CE LRMS. The Site G-PBox will evaluate which is the abstract service class associated with the user (using accepted high-level mapping policies received by the VO G-PBox) and upon a valid mapping, it will return the local UNIX Group ID (using the low-level mapping policy).

3. G-PBox overview

G-PBox (Grid Policy Box) is an authorization framework developed inside INFN. Its design foresees the deployment of G-PBox servers spread among different virtual and physical administrative domains. A VO G-PBox server contains policies created (and sent to Site G-PBox servers if needed) by the VO manager or received by Site G-PBox servers; such policies will be enforced in behalf of VO services, like VO WMS. A Site G-PBox server contains policies created (and sent to VO G-PBox servers if needed) by a Site manager or received by VO G-PBox servers; such policies will be enforced in behalf of site resources, like CEs.

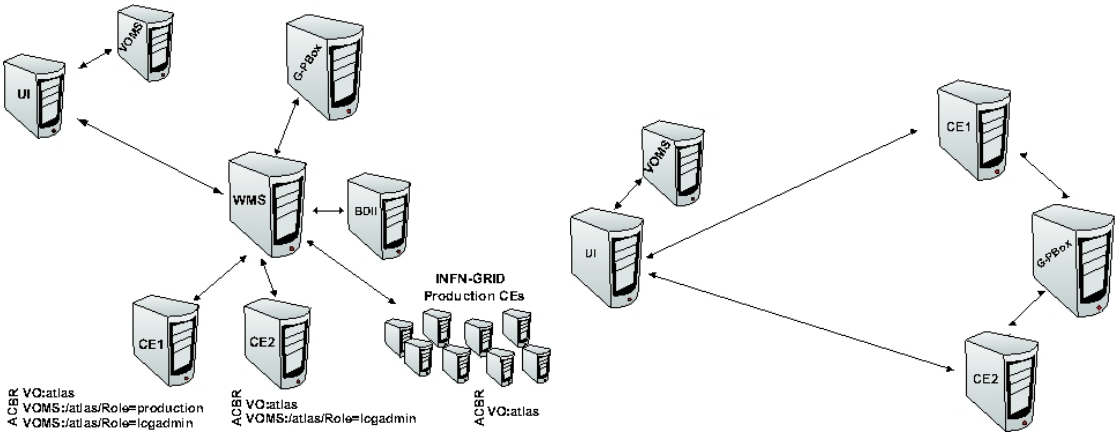
G-PBox is composed by two main components: a server and a graphical client.
The server is composed by the following modules:

PDP The Policy Decision Point (PDP) is the module that receives requests for decision, evaluates the policies regarding them and finally sends back its decisions. The XACML language is used for both policies and requests/responses. XACML is a XML policy language allowing a strict definition of the access control requirements regarding users, resources and actions. The language supports data types, functions, and combining logic which allow to build complex rules. XACML also includes an access decision syntax needed to represent the runtime request/response interaction between a PDP and PEP.

PR The Policy Repository (PR) is a native XML DB storing XACML policies, both locally- and remotely-originated, along with non XACML information (origin, active/inactive, etc.)

PCI The Policy Communication Interface (PCI) is a layer around the G-PBox used for communicating with Policy Enforcement Points (PEP) and with PCIs of other G-PBox servers.

The other main component of G-PBox, the graphical client, acts as a Policy Administration Point (PAP) and is used for policy management and distribution. Indeed policies can be created, removed and moved among different policy sets. XACML is a very powerful and flexible language but, on the other side, writing XACML policies could not be easy. The G-PBox graphical client provides a XACML editor to help the administrator to accomplish this task. An integrated VOMS handler allows to retrieve VO groups and roles. The policy distribution section of the



(a) VO G-PBox testbed.

(b) Site G-PBox testbed.

Figure 1. Testbeds deployed for the tests.

client allows to send policies to other G-PBox servers (e.g. from a VO G-PBox to a Site G-PBox) and to accept or reject incoming policies. The intent is to facilitate the interaction between different domains allowing the concrete enforcement of the agreement between Resource Owners (RO) and Virtual Organizations.

3.1. Authorization enforcement

A service (like a WMS) or a resource (like a CE) wishing to use G-PBox must interact with a PEP handling all access requests. A PEP performs the access control by making decision requests to a remote PDP and enforcing an authorization decision received by the PDP.

Currently two Grid components implement a PEP for G-PBox: the g-Lite WMS and LCAS/LCMAPS for LCG CEs. LCAS/LCMAPS is used by CEs to acquire information on the credentials of a user and to enforce authorization and mapping statements based on such credential (in this case interacting with a G-PBox PEP plug-in).

G-PBox supplies Java, C and C++ libraries to be used by a PEP to communicate with the PDP. Up to now these libraries use a proprietary protocol that guarantees high performance communication speed. The integration of a communication protocol based on agreed standards is foreseen and it will be realized inside the OMII project [12] with the exposition of a Web Service interface allowing to use the Security Assertion Markup Language (SAML) for PEP/PDP communication.

4. Test results

In this section we will describe initially the testbed used to verify our fair share solution, then we will present the obtained results. Two testbeds were set up, one for VO G-PBox interaction tests (fig. 1(a)), the other for the Site G-PBox (fig. 1(b)) tests.

4.1. VO G-PBox tests

4.1.1. Testbed description A dedicated testbed was setup to test the functionality of the VO G-PBox. It involved a gLite3.1 WMS, a LCG BDII, a G-PBox server, two virtualized LCG CEs with virtualized WNs, a dedicate VOMS server and a gLite UI. The WMS was modified in order to install a pluggable library for the communication with the G-PBox server. Only

the two virtualized CEs involved in the test published proper AccessControlBaseRule(ACBR) values, while 40 INFN-GRID production CEs, with unmodified ACBR values, were added to the BDII (fig. 1(a)) in order to have a reasonable number of sites for the G-PBox computation. The selection of the queue performed by the G-PBox server is based on the ACBR value once the proper policies are inserted into the G-PBox.

The tests were performed using a VOMS server dedicated to the Atlas Virtual Organization needed to create relevant groups and roles. Two VOMS roles were created: /atlas/Role=production and /atlas/Role=lcgadmin. The virtualized CEs were publishing four queues each (short, long, infinite and preview) and all the queues were opened also to other VOs. The ACBRs attribute values, published for the preview queue of the two testing CEs, were set as follow:

```
ACBR for preview queue in CE1:     ACBR for preview queue in CE2:
VO:atlas                           VO:atlas
VOMS:/atlas/Role=production        VOMS:/atlas/Role=lcgadmin
VOMS:/atlas/Role=lcgadmin
```

All other queues of the two virtualized CEs as well as all the other production CEs in the BDII were publishing their usual simple ACBR values for every atlas VOView:

```
VO:atlas
```

4.1.2. Performed tests To show G-PBox flexibility, two policy scenarios were considered and tested.

First Scenario: Extended access for production and lcgadmin roles. Policies were defined as follows:

(i) generic VO group /atlas/ users can only access every queue with ACBR VO:ATLAS

(ii) users with role production can access all the queues accessible by normal atlas users plus all the queues containing ACBR "VOMS:/ATLAS/Role=production"

(iii) users with role lcgadamin can access every queue with any ACBR

Given these policies and the ACBR published by the virtualized CEs (CE1, CE2) reported in previous paragraph, the WMS+GPBox system should:

(i) allow normal users (group /atlas/) to use all CE atlas queues but the preview queue of the CE1 and CE2

(ii) allow users with role production to use the preview queue on CE1 other than all queues of the normal atlas users

(iii) allow atlas users with lcgadmin role to use all queues including both the preview queue on CE1 and on CE2

The glite-wms-job-list-match command, with three different user credentials (/atlas, /atlas/Role=production and /atlas/Role=lcgadmin), was used to test the CE-queue selection by the WMS attached to the G-PBox server with the policies described above. Fig. 2 shows that the system behaves as expected. For the sake of readability, only a subset of CEs are shown as result of matchmaking process.

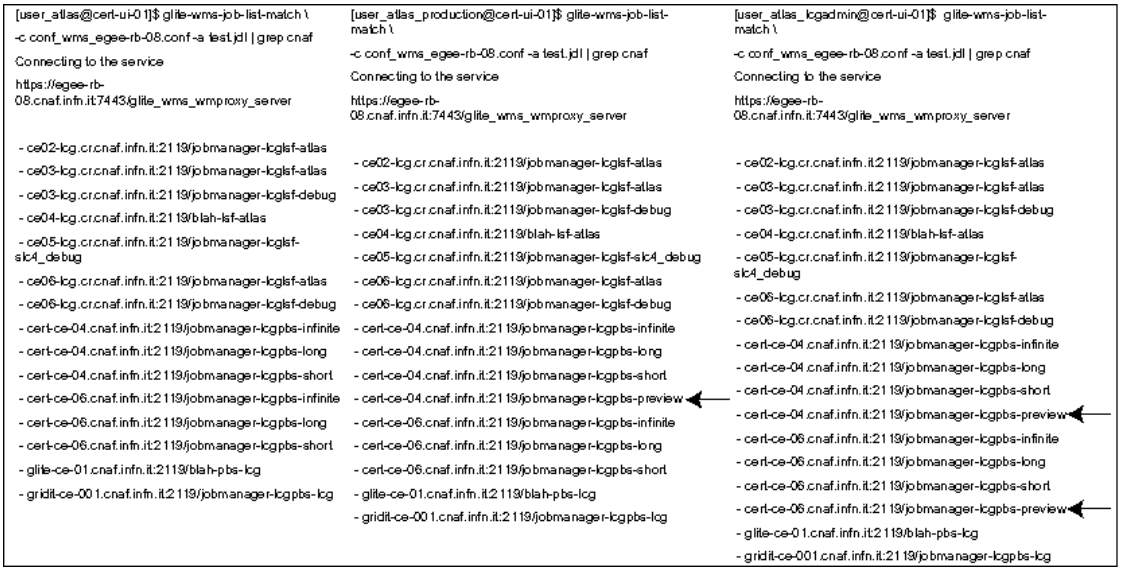


Figure 2. The output of glite-wms-job-list-match. First column shows that the generic Atlas user is not allowed to use the preview queue of the CE1 and CE2; second column shows that Atlas user with role production can submit to the preview queue on CE1 other than all queues of the normal atlas users; third column shows that Atlas users with lcgadmin role can submit to all queues including both the preview queue on CE1 and on CE2.

Second Scenario: Restricted access to production and lcgadmin roles. In this second scenario the policies were defined as follows:

(i) generic VO group /atlas/ users can only access every queue with ACBR "VO:ATLAS"
(ii) atlas users with role production can only access the queues containing ACBR "VOMS:/ATLAS/Role=production"
(iii) atlas users with role lcgadmin can only access the queues containing ACBR "VOMS:/ATLAS/Role=lcgadmin"

Given these new policies and the usual ACBR published by the virtualized CEs (CE1, CE2) reported in previous paragraph, the WMS/G-PBox interaction should:

(i) allow normal users (group /atlas/) to use all CE atlas queues but the preview queue of the CE1 and CE2
(ii) allow users with role production to use only the preview queue on CE1
(iii) allow atlas users with lcgadmin role to use only the preview queue on CE1 and on CE2

Fig. 3 shows the output of glite-wms-job-list-match with the CEs selected in this second scenario for the interesting cases of users with production and lcgadmin roles. The WMS/G-PBox interaction behaves as expected.

We underline that for both scenarios two storms of 1000 list-match requests were sent in parallel to the WMS to test robustness of our testing environment. The resulted selection efficiency was 100% for all streams.

Performance tests and optimization for WMS workload are ongoing.

4.2. Site G-PBox tests

4.2.1. Testbed description This testbed involved two LCG CEs, one G-PBox server and one LCG UI. One CE (CE1) was installed on a Intel Xeon 3.06 GHz CPU with 4 GB RAM, while

the other CE (CE2) was a Fully Virtualized server based on Xen and installed on a Intel Xeon 2.66 GHz CPU with 6GB RAM. Once received a job from UI both CEs queried the Site G-PBox (gpbox1) in order to know how to map the user submitting the job into a local UNIX account. This action is performed by a dedicated LCAS/LCMAPS plugin contacting the Site G-PBox with a XACML request/response interaction. The specific policies shown in tables of fig. 4 were used in the test, but other ∼3500 fake policies were defined and examined by G-PBox during CE request in order to face a realistic amount of policies as one can find in a common lcmaps file.

4.2.2. Performed tests The test performed consisted of 10^3 runs of the following command line for each LCG CE separately:

```
#> globus-job-run CE_HOSTNAME /usr/bin/whoami
```

Both the mapping result and the command execution time have been recorded. In tab. 1 the mean execution time is reported with error calculated under the hypothesis of Gaussian distribution of execution times (fig. 5 c,d). Concerning the fully virtualized LCG CE it is interesting to note that the distribution of execution times is bimodal (fig. 5 a,b), with a subset of execution time occurrences being far over the average. These "long" execution time occurrences are independent from G-PBox and peculiar of virtualization.

In tab. 1 we report mean execution time for VO Atlas in both real and virtual LCG CEs. For the virtualized CE the average execution time, given the bimodal distribution of execution times, the Gaussian assumption on the error estimation is not met. Therefore we report execution times with no error associated just as an indication of mean execution times in the two cases

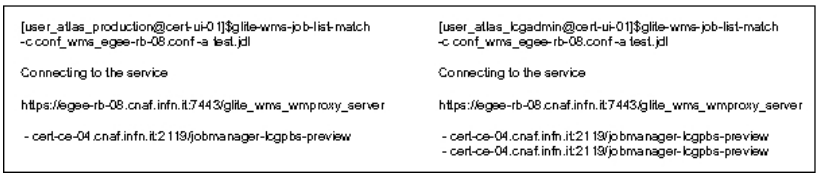


Figure 3. The output of glite-wms-job-list-match. First column shows that Atlas user with role production can only submit to the preview queue on CE1; second column shows that Atlas user with role lcgadmin can only submit to the preview queue on CE1 and CE2.

FQAN	Abstract Service Class
/atlas/Role=production	ATLAS_HIGH
/atlas/Role=lcgadmin	ATLAS_MID
/atlas	ATLAS_LOW

(a) High-level mapping policies.

Abstract Service Class	Local user
ATLAS_HIGH	atlasprd
ATLAS_MID	atlassgm
ATLAS_LOW	.atlas

(b) Low-level mapping policies.

Figure 4. Tables showing policies in the Site G-PBox.

Hostname	With G-PBox	Without G-PBox
virtual lcg-CE	7.7(*)	8.2(*)
lcg-CE	6.438 +/- 0.006	6.525 +/- 0.008

Table 1. Mean execution time for the test command. (*) For the virtual lcg-CE, given the bimodal distribution, the Gaussian assumption on the error estimation is not met.

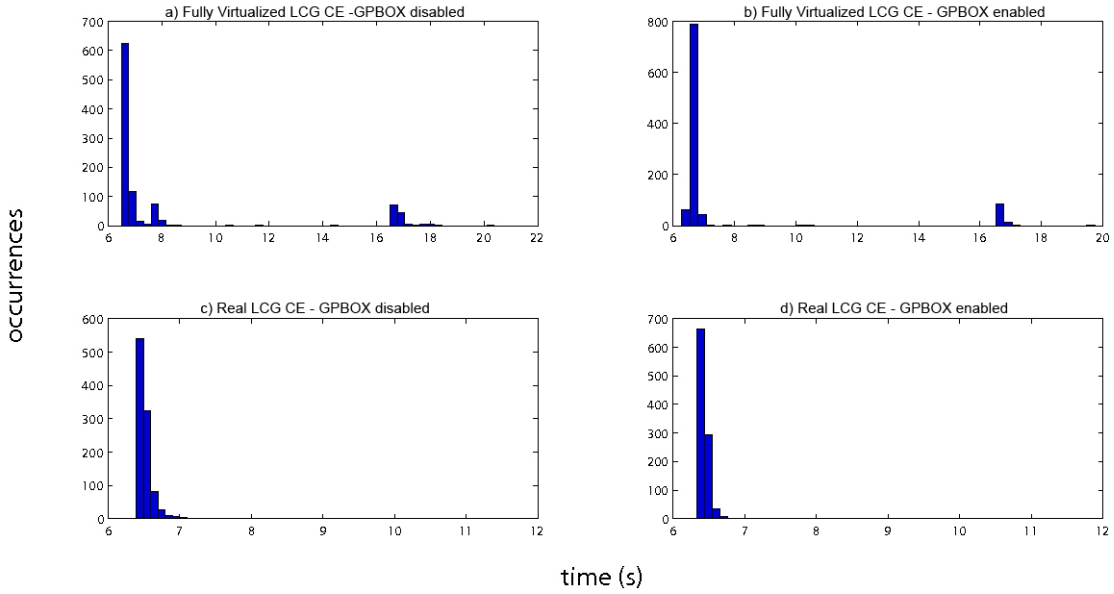


Figure 5. Distribution of test command execution times on LCG CEs with and without G-PBox.

(with/without G-PBox). Concerning the virtual CE, given the complex distribution of the execution times, we can only observe that the averages of execution times are compatible in the two cases.

When focusing on the real LCG CE we can observe that the execution time difference between G-PBox and LCMAPS user mapping is negligible for typical production environments. To check the magnitude order of amount of time spent in Site G-PBox communication, we run 1250 of such authentication requests tracking the correspondent execution time. The test was conduced on the real LCG CE on which the mean request time measured was (8.7 +/- 0.3)ms. This last test suggests that G-PBox execution time is negligible with respect to magnitude order of a globus-job-run execution.

5. Conclusion

In this paper, we have proposed a suitable mechanism that grounds on a rigorous definition of service classes and on a dynamic binding of class instances to privilege attributes associated to Grid identities.

The proposed approach has been prototyped without publishing the defined service classes in the context of the gLite preview testbed and with the collaboration of a huge Grid site, the LCG Tier1 at INFN CNAF (National Centre for Research in Informatics and Telematics), in order to verify its feasibility with current WMS and CE components.

The result of first tests showed the feasibility of this approach. Future activities are targeted at extending the testing phase to more resources.

The final goal is to contribute with our experience to a concrete mechanism for the Grid production middleware.

6. Acknowledgments

We wish to thank Marco Cecchi of INFN for his valuable support in handling the large amount of work related to the interaction between WMS and G-PBox APIs.

References

[1] I Foster C Kesselman S T 2001 *International J. Supercomputer Applications*
[2] Nagaratman *et al.* 2003 Security architecture for open grid services memo GWD-I GGF OGSA Security Workgroup
[3] Bacon J, Moody K and Yao W 2002 *ACM Transactions on Information and System Security (TISSEC)* **5** 492–540
[4] Caltroni A, Ciaschini V, Ferraro A, Ghiselli A, Rubini G and Zappi R 2004 *Proceedings of the International CHEP 2004* (Interlaken, Switzerland)
[5] INFN Grid. http://grid.infn.it
[6] 2006 Enabling Grid for E-sciencE http://www.eu-egee.org/
[7] Andreozzi S, Cecchi M, Ciaschini V, Ferraro A, Ghiselli A, Giacomini F, Italiano A, Rubini G and Salomoni D 2006 *Proceedings of the Cracow Grid Workshop 2006 (CGW2006), Cracow, Poland, October* URL `http://www.cyfronet.pl/cgw06/`
[8] The GLUE schema homepage. http://glueschema.forge.cnaf.infn.it/
[9] Alfieri R, Cecchini R, Ciaschini V, dell'Agnello L, Frohner A, Gianoli A, Lörentey K and Spataro F 2004 *Proceedings of the 1st European Across Grids Conference, Santiago de Compostela, Spain, February 2003, LNCS* **2970** 33–40
[10] Andreetto P, Andreozzi S, Avellino G *et al.* 2004 *Proceedings of the Conference on Computing in High Energy and Nuclear Physics (CHEP 2004), Interlaken, Switzerland*
[11] OASIS eXtensible Access Control Markup Language. URL `http://www.oasis-open.org/committees/xacml/`
[12] 2006 Open middleware infrastructure institute europe http://omii-europe.org