

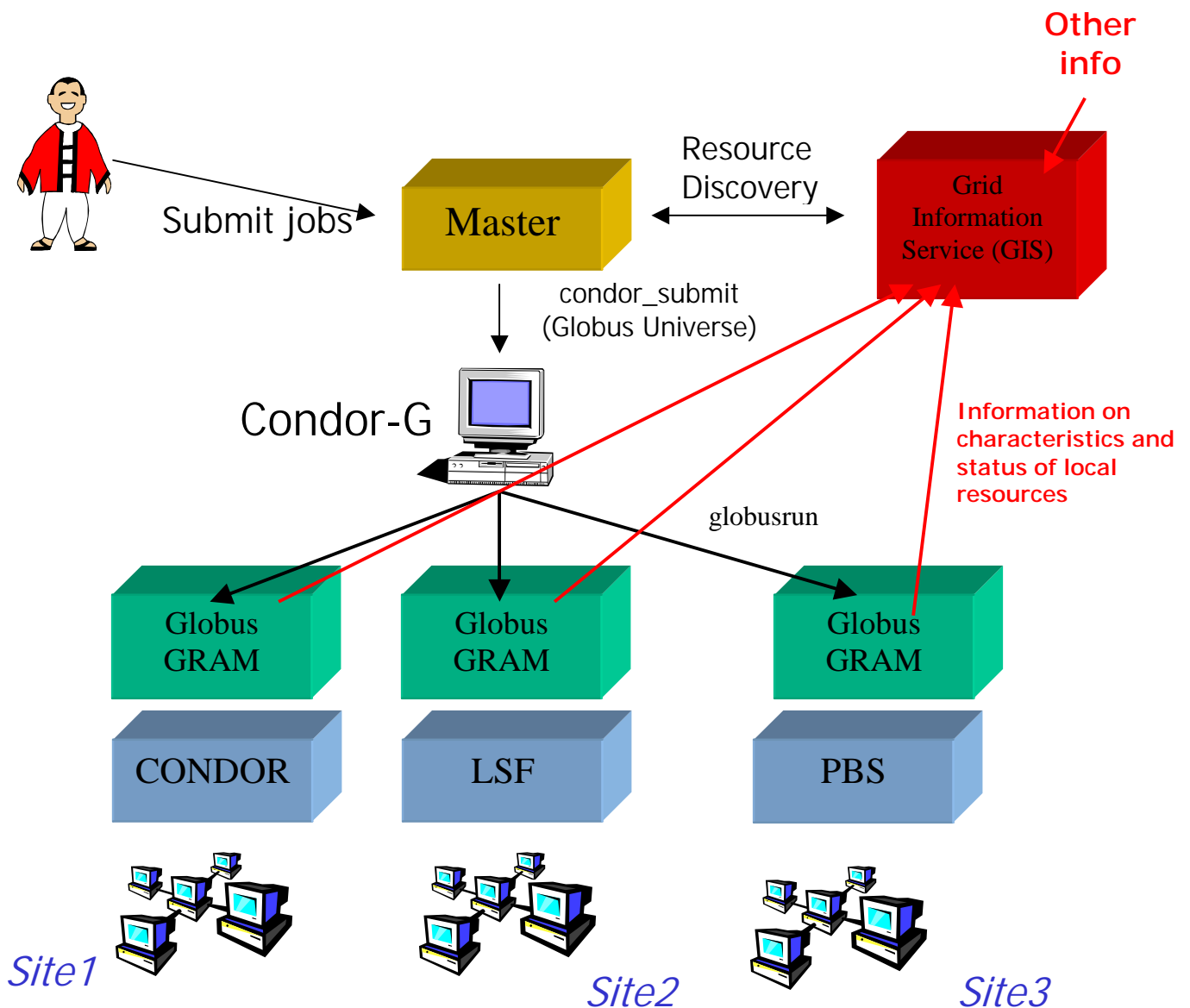
Workload Management WP Status Report

Release 1.0.3
November 28, 2000

Workload system prototype

As already reported in Marseille, the activities of the workload management work package have been “bootstrapped” considering a real use case: the HLT activities for the CMS experiment (Monte Carlo production and reconstruction).

This use case has been discussed with INFN representatives of the LHC experiments and also with members of the Globus team (S. Tuecke and L. Liming) and of the Condor team (M. Livny). As result of these discussions, a prototype architecture of a workload management system, represented in the following picture, able to “manage” such high throughput applications, and suitable in particular for “scheduled” applications (such as Monte Carlo production and production analyses) has been proposed.



Using a “bottom-up” approach to describe this picture, various farms, spread across in different sites, are managed by possible different local resource management systems (such as LSF, PBS, Condor, etc...): it is not possible to assume that a single type of resource management system can be considered in a Grid environment.

The Globus GRAM service is used as a uniform interface to these different resource management systems.

To submit jobs to these Globus resources, Condor-G has been considered, in order to exploit some features of the Condor architecture, such as the reliability (Condor save the information related to the submitted jobs in a persistent queue), the logging and the monitoring capabilities, etc...: the jobs submitted to Condor-G via a *condor_submit* command (using the so-called Globus Universe) are “translated” in Globus jobs, and submitted to the various farms (Globus resources).

In this architecture Condor-G is not a “smart” component, since it is not able to choose in which resources the jobs must be submitted. This functionality is up to the so-called Master.

The master chooses the Globus resources (the farms) where to submit the jobs, querying an Information Service, where all the information necessary to perform this scheduling (such as the characteristics and the status of the various farms, the location of the various data sets, the characteristics and the status of the network, etc...) must be published. Some information (the characteristics and the status of the local farms) could/should be provided to the Grid Information Service by the GRAMs, while the other information should be published in the Information Service by specific “information providers”.

Therefore this model of workload management system include “elements” of the Globus (in particular the GRAM service for resource management, the Grid Information Service, the GSI service for authentication) and of the Condor architectures, but “major” developments (in particular the Master), are necessary.

The on-going activities are finalized to evaluate the existing components (deliverable D1.1), in “customizing” and “integrating” together the various building blocks of this architecture (implementing the missing ones).

The Globus and Condor-G services have already put in place (for this purpose visits of some members of the Globus and of the Condor team in Italy, and a visit to the Condor team in Madison and to the Globus team in Argonne [1] has been quite useful to solve some problems and speed up the works). A deep and quite comprehensive evaluation of these two architectures has already been performed, testing the interactions between these two components considering also real applications in real production environments.

The implementation of the master will start soon (decisions will be taken during the next meeting of the WP, scheduled for December 4, 2000)

Evaluation of the Globus services

An evaluation of the Globus services considered in this architecture has been performed, in collaboration with the Work package 1 of the INFN-GRID project (Installation and Evaluation of the Globus toolkit) [2].

For what concerning the GRAM service, tests have been performed in particular to evaluate its functionalities to find if it can be used as a uniform interface to different resource management systems (tests have been performed considering LSF, Condor and PBS as underlying resource management systems).

Although the current implementation of the GRAM service is able to meet some basic requirements, we think that some major problems should be addressed, in order to have a reliable and robust service that can be used in production systems. For example there is a scalability problem: a Globus job manager process runs in the gatekeeper for each job submitted to a Globus resource. This can be a serious problem if the considered resource is a farm with a Glous front-end machine, and many jobs are submitted to this farm (a typical scenario that must be considered in the HEP environment). An other possible problem is related with the reliability, since the Globus job manager (responsible to “take care” of the job submitted via Globus) is not persistent, and therefore is not resilient to faults of the farm nodes.

The Globus Resource Specification Language (RSL) has been evaluated, to understand if it could be a viable and suitable solution to “describe” the resources needed by the jobs.

Although the RSL syntax model seems suitable to define even complicated resource specification expressions, we think that a language with a non-extensible common set of attributes is not a viable solution. In our opinion much more flexibility is required: the administrators of the Grid resources should be allowed to define new attributes that describe these resources, and the users should be allowed to use these new attributes in their resource specification expressions.

The “cooperation” between the GRAM and the GIS services has been considered as well, to find which information related to a farm are published in the Grid Information Service.

In the default configuration many useless attributes (at least for our needs) are published in the GIS, while important information are missing.

A proposal for a first possible modification of the default schema for what concerning the information related to a farm, integrating the default attributes with new information provided by the underlying resource management system, is under discussion [3].

The fixes for the bugs that have been found during these evaluations of the Globus services have been included in the INFNGRID distribution [4], the toolkit used to make the installation of the Grid software (in particular Globus, at the moment) easier and more automatic, and that allows to implement some specific INFN customizations (user and host certificates signed by the INFN CA, and a hierarchical architecture of the Grid Information Service [5]).

Evaluation of Condor-G

Condor-G, used to submit Condor jobs to Globus resource, has been evaluated.

Note that as Condor jobs we don’t mean standard Condor jobs, that is jobs relinked with the Condor library (in order to profit by the remote I/O and checkpoint features): on the contrary these are “normal” jobs, that don’t require special compilation and/or relinking.

The current implementation of Condor-G is a prototype: therefore there are still various problems that must be solved.

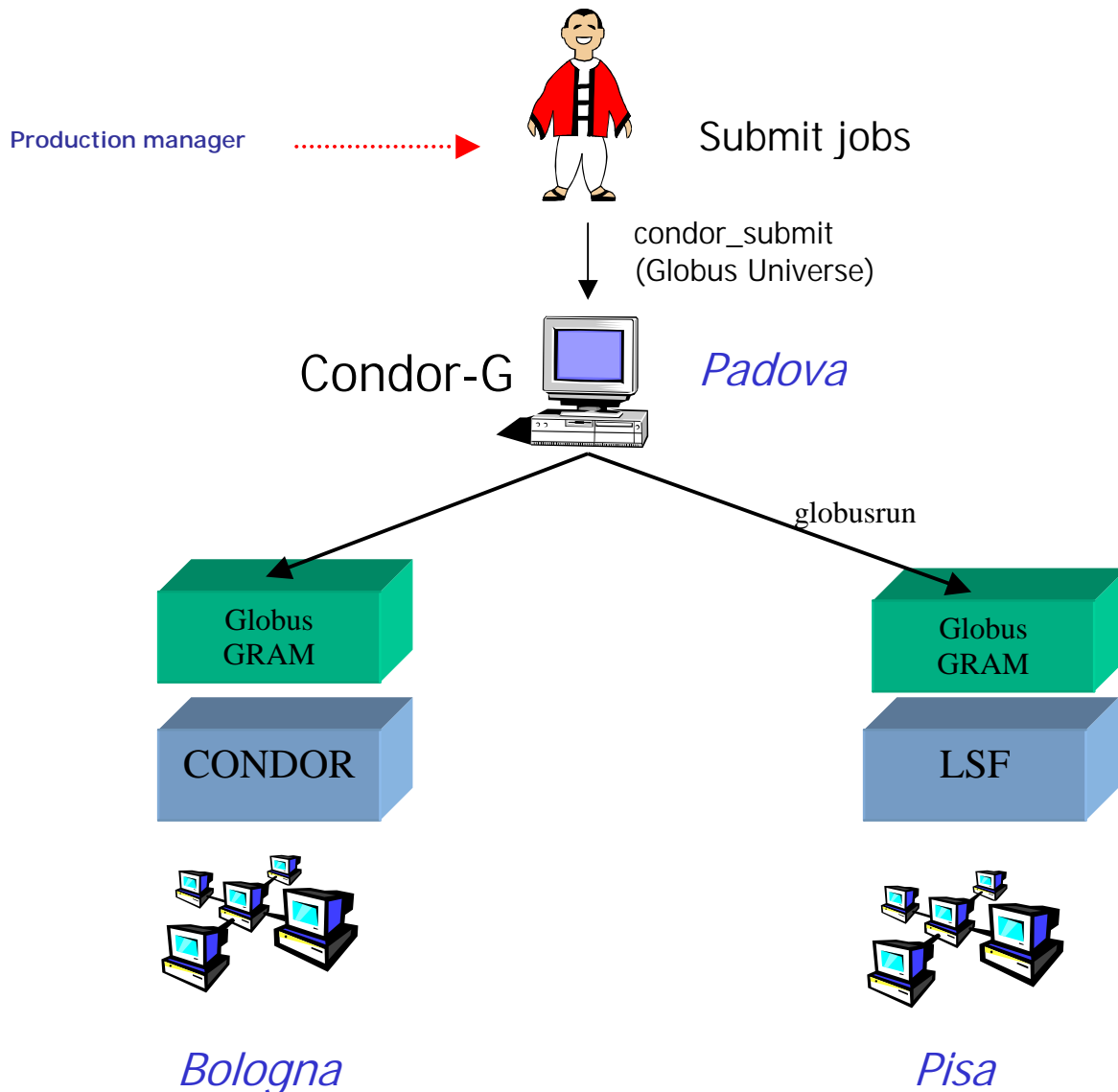
For example there are scalability problems (in the current implementation there is a running shadow process in the submitting machine for each job submitted), problems with logging (the log file reports as executing machine the name of the submitting machine), etc...

It must be stressed that in the current implementation Condor-G can provide a certain level of reliability only in the submitting side, but not in the executing site, since Globus is not able to provide such capability.

CMS HLT production

To fully evaluate the functionalities and the “interactions” between Condor-G and Globus, they have been tested considering a real production of the CMS experiment, therefore with a real application for MC production (*pythia*) and on a real production environment.

The following layout has been considered for this test:



During these tests many memory leaks have been found in the Globus job manager, in various “modules” of the Globus software, and therefore it has not been possible to perform this production using the Condor-G and Globus tools.

A very considerable effort has been put in order to find and fix these bugs (the “style” of the Globus software doesn’t help to understand the protocol and the “mechanisms” used).

We (F. Prelz) have been able to provide fixes for these memory leaks, which have been reported to the Globus team [6].

The feedback from the Globus team has come from Douglas Engert, only for the bugs in the GAA and GSS modules. He has provided some other fixes, which have been “merged” with the original ones. So far we haven’t had any feedbacks for what concerning the other reported bugs and fixes.

These fixes for the Globus job manager memory leaks are being included in the next release (1.3) of the INFN-GRID distribution.

Future activities

Since the problems with the memory leaks of the Globus job manager have apparently been solved, the tests with Condor-G and Globus can be repeated with the next CMS production that will start in a few weeks.

With this production we are planning to evaluate also the Condor bypass software [7], which allows to redirect just the standard input/output/error files in the submitting machines.

Therefore considering CMS applications such as *pythia* and *cmsim*, the production manager (the person responsible to “run” the production) will be able for example to save in the file system of the submitting machine just the standard output, a file of small size that is very useful to check the status of the production (to find if some errors occurred, to find how many events have been produced so far, etc...) while the resulting n-ples will be saved in the file system of the executing machine.

We will have to start the development of the Master.

As first approach, we are going to consider the Condor Class-Ads as language to specify the resources required by jobs, since this mechanism seems flexible enough to describe the various requirements. Therefore in the first implementation of the master we are planning to use the Condor matchmaking library [8], to find the matches between the resources required by the jobs (expressed using the Class-Ads) and the resources available in the Grid (published in the GIS): a “converter” from LDAP attributes (published in the GIS) to Class-Ads must be implemented as well.

We are also going to evaluate the new Condor-G implementation, when ready, which includes also a new “customized” Globus job manager, under development by the Condor team.

This new Condor-G implementation should fix some problems of the current implementation (for example the scalability in the submitting machine and should introduce some new features, such as a new protocol for job submission, and a new Globus job manager e able to “reattach” to a running process, therefore addressing the reliability problem.

A meeting of the workload management WP will be held at CNAF (Bologna) on December 4th, 2000

References

- [1] <http://www.pd.infn.it/~sgaravat/report-from-usa.pdf>
- [2]: <http://www.infn.it/globus>
- [3] <http://www.infn.it/globus/documents.htm#gis-farm>
- [4] <http://www.pi.infn.it/GRID/dist/>
- [5] <http://www.mi.infn.it/~lobiondo/GIS/>
- [6] http://www-unix.globus.org/mail_archive/discuss/msg00646.html
- [7] <http://www.cs.wisc.edu/condor/bypass>
- [8] <http://www.cs.wisc.edu/condor/classad/>