Adding Standards Based Job Submission to a Commodity Grid Broker

David Colling[1], A. Stephen McGough[2], Tiejun Ma[3], Jazz Mack Smith[2], Vesso Novov[2], David Wallom[3] and Xin Xiong[3]

Outline

The Condor matchmaker provides a powerful mechanism for matching together both users' job requirements and resource providers' requirements in such a manner that not only is a pairing selected which meets the requirements of both parties but the pairing is optimal for both. This has made the Condor system a good choice for use as a meta-scheduler within the Grid. Integrating Condor within a wider Grid context is possible but only through modification to the Condor source code. We describe how the emerging standards for job submission and resource descriptions can be integrated into the Condor system to allow arbitrary Grid resources which support these standards to be brokered through Condor.

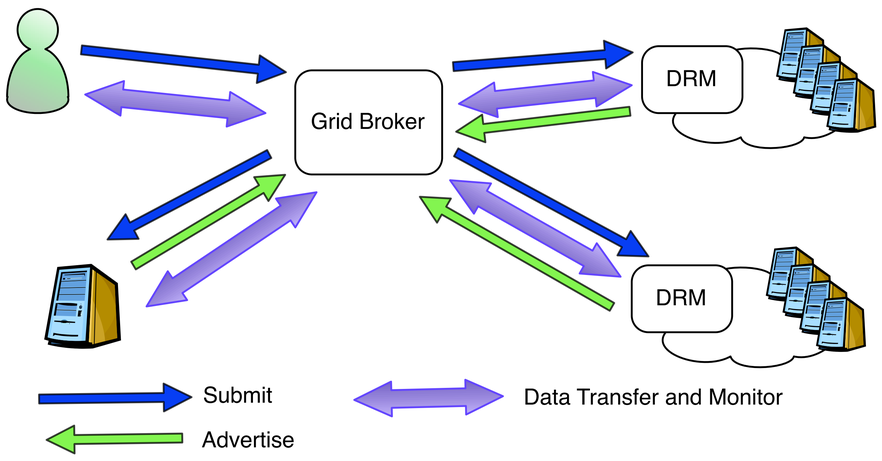

Fig 1: Grid Brokering Architecture

Grid Standards

Over the last few years the Open Grid Forum (OGF) have been standardizing the way jobs are submitted to the Grid and how resources are discovered for these jobs. This has lead to the Job Submission Description Language (JSDL), the Open Grids Services Architecture Basic Execution Service (OGSA-BES) and the Grid Laboratory Uniform Environment (GLUE) standards. Grid resource brokers are now adopting these standards. We look here at adding these standards to Condor.

Grid Standards Based Architecture

We have extended the Condor Matchmaker (Broker) with standards based resource discovery and job submission – GridBS. Figure 2 illustrates the general architecture with the three main components to support the Broker: advertising of resources, deployment of jobs and file staging.

The user submits a job to the broker using the Condor submission tools. However, for a complete Grid solution a BES/JSDL to Condor interface – such as GridSAM – could be placed in front of Condor completing the encapsulation. Matched jobs in Condor are translated from ClassAds into JSDL documents and then submitted to the remote BES instance. Information about these DRM resources is captured and described using the GLUE schema which can be stored in a persistent repository such as Grimoires. The GLUE documents are translated into ClassAds and entered into the Condor Collector. File stage handling provides the mechanism for ensuring files are staged to and from the appropriate resource.

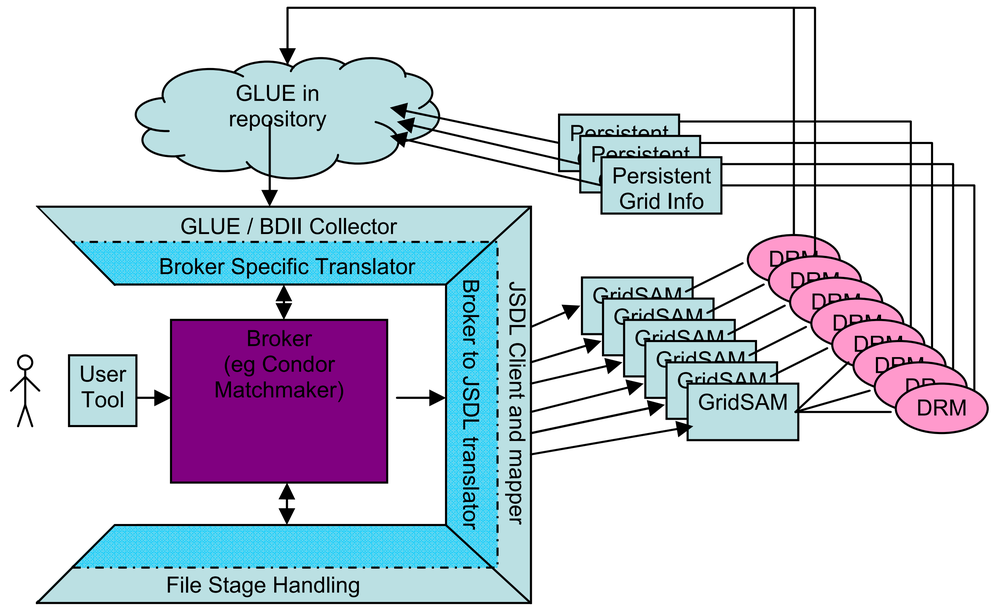

Fig 2: GridBS Architecture

[1] Department of Physics, Imperial College London
[2] Department of Computing, Imperial College London
[3] Oxford e-Research Centre, University of Oxford

Resource Information Integration

We have designed a Grid information retrieval agent, which can parse the different versions of the GLUE information model and translate instances into Condor ClassAds. The main components: **GLUE information retriever**, which can interrogate an LDAP-based directory service or UDDI-based registry to collect resource information (e.g. CPU count). **Information translators**, which translate LDAP objects or XML into ClassAds according to the mapping rules. **Writer/verifier**, which writes these ClassAds into files. Before the information is written, some critical attributes such as the name of the job manager, are compared with static values pre-stored on the VOMS server. If the files fail verification they are not written, this helps ensure the correctness of the generated ClassAds. These ClassAds can then be published into Condor.

Since the Microsoft HPC Server 2008 does not have a method to publish information in a standards compliant manner, which we could use to generate ClassAds, we have developed our own publishing system. HPCS08 GLUE information publishing system is implemented as three packages: GLUE-Generation, GLUE-Translation and GLUE-Publish. With these components, HPC Server 2008 can publish its resource information to an XML-based UDDI registry, such as Grimoires 1.7.

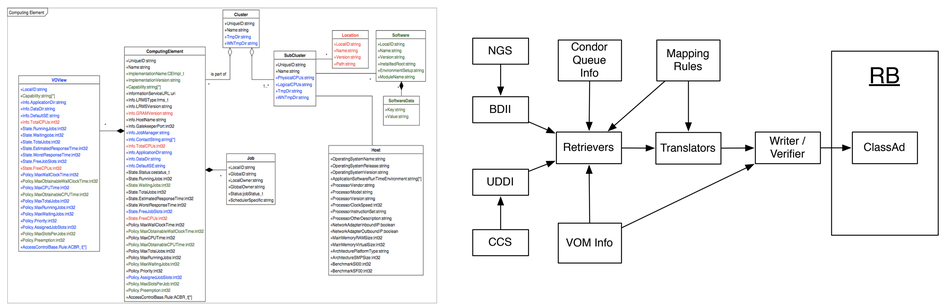

Fig. 3: GLUE Information Retrieval and Translation

Job Submission and Monitoring

Condor has a mechanism for adding new job execution hosts called the Grid ASCII Helper Protocol (GAHP). Unfortunately this is not standards based and it requires modification of the Condor source code for each new Grid resource type you wish to add. We have implemented our own GAHP client and server with the server side using the OGF JSDL and BES standards to communicate with standards based Grid resource brokers. This allows new Grid brokers to be used without the need to modify the Condor source code.

The JSDL/BES GAHP server translates the ClassAd document into a JSDL document. If the files are not located on a file space that the BES service can read then these files are copied to a file space that the BES service can see. In either case the file locations are added into the JSDL document. The job is submitted to the BES service which returns a unique ID for the job. Condor maps this ID to a standard Condor job ID.

The state of a job can be determined through queries to the JSDL/BES GAHP server. The state model for BES jobs is a subset of those available in Condor.

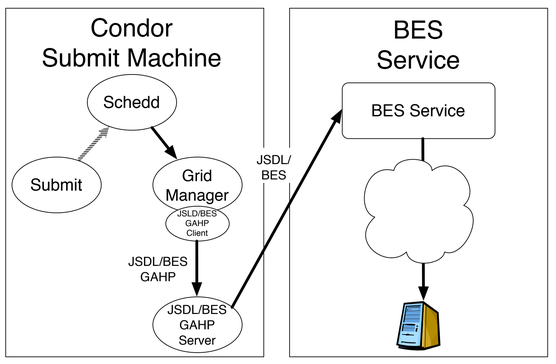

Fig4: Condor Jobs through JSDL/BES

Conclusion

We have shown how the Condor system can be integrated with standards for job submission (JSDL / BES) and resource description (GLUE) developed through the OGF. This allows the use of a powerful brokering service to be used with resources exposed through standards based interfaces. A more scalable solution than using GAHP alone as different resource developers need only implement the BES interface rather than developing both a Client and Server GAHP to use within Condor. This will also allow them to integrate with other BES compliant clients. We are currently evaluating this work with the Condor resource broker at Oxford – brokering jobs to around 35 cluster resources, which include the NGS UK Grid resources, Oxford University departmental Condor resources, Oxford Super Computing Centre resources and Windows HPC resources, having so far processed in excess of 17000 individual jobs. This is allowing us to develop a whole Grid ecosystem which can make the basis of a campus/other Grid toolkit allowing quick and easy deployment of Grid systems through institutions.

Acknowledgements

We would like to thank OMII-UK for funding this work.

Imperial College London
High Energy Physics Imperial College London
internetcentre
London e-Science Centre
OpenGridForum OPEN FORUM | OPEN STANDARDS
omii-uk www.omii.ac.uk