

Scalability Tests of R-GMA Based Grid Job Monitoring System for CMS Monte Carlo Data Production.

D. Bonacorsi, D. Colling, L. Field, S. Fisher, C. Grandi, P. R. Hobson, P. Kyberd, B. MacEvoy, J. J. Nebrensky, H. Tallini, S. Traylen

Abstract-- High Energy Physics experiments such as CMS (Compact Muon Solenoid) at the Large Hadron Collider have unprecedented, large-scale data processing computing requirements, with data accumulating at around 1 Gbyte/s. The Grid distributed computing paradigm has been chosen as the solution to provide the requisite computing power. The demanding nature of CMS software and computing requirements, such as the production of large quantities of Monte Carlo simulated data, makes them an ideal test case for the Grid and a major driver for the development of Grid technologies. One important challenge when using the Grid for large-scale data analysis is the ability to monitor the large numbers of jobs that are being executed simultaneously at multiple remote sites. R-GMA is a monitoring and information management service for distributed resources based on the Grid Monitoring Architecture of the Global Grid Forum. In this paper we report on the first measurements of R-GMA as part of a monitoring architecture to be used for batch submission of multiple CMS Monte Carlo simulation jobs running on the CMS-LCG0 Grid test-bed. Monitoring information was transferred in real time from remote executing nodes back to the submitting host and stored in a database. Scalability tests were undertaken whereby the job submission rate was ramped up to rates comparable with those expected in a full-scale production.

I. MONITORING ARCHITECTURE

THE management of a large Monte Carlo (MC) production or data analysis, as well as the quality assurance of the results, requires careful monitoring and bookkeeping. BOSS (Batch Object Submission System) [1] has been developed as part of the Compact Muon Solenoid (CMS) suite of software to provide real-time monitoring and bookkeeping of jobs submitted to a compute farm system. Its original design assumed that jobs were submitted to a local

batch farm. Individual jobs to be submitted are wrapped in a BOSS executable which, when it executes, spawns a separate process that extracts information from the running job's input, output and error streams. Pertinent information (such as status or events generated) for the particular job is stored, along with other relevant information from the submission system, in a local database.

In order for the BOSS database to monitor reliably jobs in a Grid environment, jobs executing on a remote compute element need to pass information back to the submitter's site in real-time. Hence BOSS has been modified to use the Relational Grid Monitoring Architecture (R-GMA) as a transport mechanism to deliver information from the remotely running job to the centralized BOSS database at the User Interface (UI) of the Grid system, from whence the job was submitted.

The R-GMA architecture is based on that of the Grid Monitoring Architecture (GMA) [2] of the Global Grid Forum (GGF) [3]. The GMA as shown in Figure 1 consists of three components: consumers, producers and a directory service, which we refer to as a registry as it avoids any implied hierarchical structure.

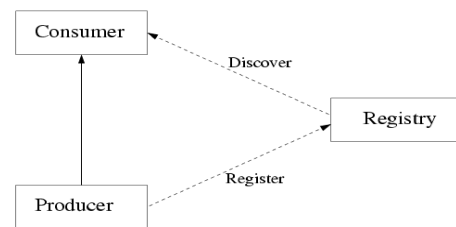


Fig.1 Grid Monitoring Architecture

Producers of information register themselves with the registry when they are instantiated. The registry describes the type and structure of information the producers want to make available to the Grid. Consumers of information query the registry to find out what type of information is available and locate producers that provide the required information. Once a consumer has this information, it contacts the producer directly to obtain the relevant data. R-GMA implements this general

Manuscript received October 29, 2003. This work was supported in part by PPARC and by the European Union.

D. Bonacorsi and C. Grandi are with the Istituto Nazionale di Fisica Nucleare, Bologna, Italy.

D. Colling, B. MacEvoy and H. Tallini are with the Department of Physics, Imperial College London, London, UK

L. Field, S. Fisher and S. Traylen are with the Particle Physics Department, Rutherford Appleton Laboratory, Chilton, UK

P. R. Hobson, P. Kyberd and J. J. Nebrensky are with the Dept. of Electronic and Computer Engineering, Brunel University, Cleveland Road, Uxbridge, UB8 3PH, UK. Contact Paul.Kyberd@brunel.ac.uk

GMA architecture using Java servlet technology (Tomcat [4]), so that as well as the producer and consumer described above, producer and consumer servlets exist and communicate using HTTP or HTTPS.

In the architecture used for our tests, R-GMA and BOSS were used in the following way. A job is submitted to the Grid from a UI and executes on a remote compute element (CE). When the job runs, BOSS creates a producer that sends its details, via a local producer servlet, to the registry, which records details about the producer including a description of the data, though not the data itself. As the job runs and monitoring data on the job are generated, the producer streams this to the producer servlet. At the UI, a consumer is created. This consumer sends its registration details, via a locally running consumer servlet, to the registry. The registry stores details of this consumer and then sends back a list of available matching producers. The consumer then contacts each producer directly and initiates data transfer, storing the information in the centralized BOSS database.

Using BOSS and R-GMA in this way is a flexible and robust method for retrieving real-time job monitoring information. The use of standard Web protocols for data transfer allows straightforward operation through site firewalls and networks. Moreover, with only a single local connection required from the consumer to the BOSS database (rather than from a potentially large number of remote Grid compute sites) this is a more secure mechanism for storing data.

II. TEST SETUP AND INITIAL RESULTS

The monitoring architecture was initially tested on the European Data Grid (EDG) test-bed [5]. An R-GMA producer servlet (version 2.2.4) was installed on a single Compute Element at Imperial College, a consumer servlet was installed on the Imperial College UI and a registry was installed at Brunel University. The BOSS database was installed on the Imperial College UI. Small numbers of CMS MC jobs were submitted from the UI to the EDG Grid with the requirement for the resource broker to send the job to the Imperial College farm, so that the appropriate R-GMA producer software would be found. It should be emphasized that standard CMS MC production software was used in submitting these jobs, and as such this represented a “real world” application test of R-GMA.

In this initial test, the MC jobs were configured to generate only a small number of events, such that they took about 10 minutes to complete execution. For each job, approximately 30 separate messages were required to be sent back to the BOSS database, via R-GMA. In addition to sending information in real-time to the R-GMA producer, BOSS also writes to a log file that can be recovered later. A comparison can then be made between the information in the database and the log file to verify that no information has been lost or corrupted.

In initial tests, submitting small bunches of jobs such that no more than five jobs were running simultaneously, the

monitoring system was shown to work successfully with all information being successfully relayed and stored in the BOSS database.

As the submission rate was ramped up, however, information was lost. This was thought to be due to the use of a (now deprecated) producer that has a limited buffer size such that at high transfer rates, information gets overwritten before the consumer has had a chance to retrieve it.

III. TEST DESCRIPTION

R-GMA (version 3-3-28) was installed on machines on a CMS specific LHC Computing Grid [6] test-bed. With the initial problem resolved, a more demanding test was required to provide a realistic stress on the system. It was not possible to dedicate all of the test bed machines to a test of the monitoring architecture. Nor is it a suitable use of resources to involve software whose ability to perform is completely unknown in a data challenge. Finally a test of the real system is limited to the performance provided by the current system; it is not possible to test the performance of a future, upgraded system. It was therefore decided to create a simulation of the production system, specifically of the output from *CMSIM*.

Firstly around fifty jobs were individually run. The times of their messages were recorded. From this information a message profile of a typical CMSIM job was created.

It can be seen from Figure 2 that we may split the jobs into three phases with different characteristics. During the first second of the job, messages are sent equally spaced at a rate of around one every 50 milliseconds and a single message occurs some 890 seconds later with an uncertainty of 170 seconds. This ends the first phase of the job. During the main phase of the job messages are generated about every 8800 seconds, gaussianly distributed with a width of 480 seconds. In the final phase, as the job performs various housekeeping operations, 40 messages are generated in bursts over the last 100 seconds. The messages have a mean length of 35 characters.

A test class was then produced which created and sent messages with appropriate intervals and distributions. An instance of such a class is a realistic model of the performance of a single CMS Monte Carlo job as far as the R-GMA system is concerned. Since no processing is required between messages, a single CPU is capable of providing a load equivalent to many CPU's running the full job.

A single machine can generate messages at a rate of greater than fifty per second; thus it can provide a load greater than a typical Grid cluster of 200 machines, up to a load of 4 Hz per node.

The termination messages only last a second but with the distribution of termination times *even* for jobs which start simultaneously stretching over 5000 seconds, the chances of two jobs finishing together is less than 0.1%.

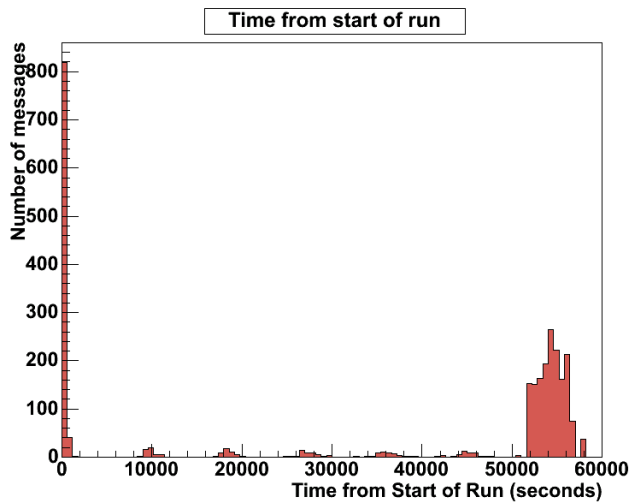


Fig.2 Message time distribution for 44 CMSIM jobs.

Job initialization generates a similar number of messages and here the limiting factor is how fast a submission script is capable of starting jobs on the grid. Preliminary measurements indicate that the time for jobs to propagate through the workload management system and start executing varies between 3 and 10 seconds. Thus the initial message bursts of submitted jobs will not overlap. In our simulation the elapsed time for delivering 20 messages in 1 second from each of a set of threads started as fast as possible exceeds the time for delivering the same twenty messages from a single thread by a time which is much less than 1% of that for fifty threads.

We thus have a system that allows a single machine to generate an R-GMA load that is equivalent to several hundred machines at a single site.

IV. TEST RESULTS

Adding a local machine to run the R-GMA servlets and with two machines at a site we can simulate the full load on R-GMA. To complete the test system we placed a registry at RAL and the consumer at Imperial College. Tests so far completed have involved running such a system at up to four geographically dispersed sites.

With the current implementation R-GMA can handle 400 jobs with no message loss. This represents approximately 20% of the predicted CMS production load. At slightly greater loads the system fails completely. The authors of R-GMA are currently investigating this failure, which seems to be associated with system resource limits on the monitoring (consumer servlet) machines.

V. SUMMARY

We have created a system to simulate the loads that R-GMA will experience during its use as a monitoring tool for the CMS data challenges on the European Data Grid test-bed. The model

indicates that at present R-GMA is unable to sustain the expected loads.

VI. REFERENCES

- [1] C. Grandi and A. Renzi, "Object Based System for Batch Job Submission and Monitoring (BOSS)", CMS Note 2003/005; <http://www.infn.it/cms/computing/BOSS>.
- [2] B. Tierney, R. Aydt, D. Gunter, W. Smith, V. Taylor, R. Wolski, M. Swamy, "A Grid Monitoring Architecture", Technical Report GWD-Perf-16-1, GGF, 2001.
- [3] Global Grid Forum <http://www.gridforum.org/>
- [4] <http://jakarta.apache.org/tomcat/>
- [5] <http://eu-datagrid.web.cern.ch/eu-datagrid/>
- [6] <http://lcg.web.cern.ch/LCG/>