ATLAS DATA CHALLENGE PRODUCTION ON GRID3

M. Mambelli*, R. Gardner# (University of Chicago)
Y. Smirnov, X. Zhao (University of Chicago)
G. Gieraltowski, E. May, A. Vaniachine (Argonne National Laboratory)
R. Baker, W. Deng, P. Nevski (Brookhaven National Laboratory)
H. Severini (Oklahoma University)
K. De, P. McGuigan, N. Ozturk, M. Sosebee (University of Texas at Arlington)

Abstract

We describe the design and operational experience of the ATLAS production system as implemented for execution on Grid3 resources. The execution environment consisted of a number of grid-based tools: Pacman for installation of VDT-based Grid3 services and ATLAS software releases, the Capone execution service built from the Chimera/Pegasus virtual data system for directed acyclic graph (DAG) generation, DAGMan/Condor-G for job submission and management, and the Windmill production supervisor which provides the Jabber messaging system for Capone interactions with the ATLAS production database at CERN. Datasets produced on Grid3 were registered into a distributed replica location service (Globus RLS) that was integrated with the Don Quijote proxy service for interoperability with other Grids used by ATLAS. We discuss performance, scalability, and fault handling during the first phase of ATLAS Data Challenge 2.

INTRODUCTION

ATLAS (A Toroidal LHC ApparatuS) [3] is one of the four experiments that will use the Large Hadron Collider at CERN (Geneva, Switzerland) beginning in 2007. Many institutions worldwide are collaborating to provide the necessary effort for this frontier high energy physics experiment. Besides the accelerator and the detector, the computing infrastructure is complex as well because of the required scale and the extensive development of new software. In 2001 the LHC Computing Review [2] recommended the experiments undertake data challenges (DC) of increasing size and complexity to drive development of the necessary computing infrastructure. In response, data challenges have been organized to exercise both the ATLAS software and the IT infrastructure that will be used to reconstruct and analyze data when the LHC becomes operational. More specifically, the goals of the data challenges are to validate the LHC computing model, develop distributed production and analysis tools, and to provide large datasets for physics working groups.

A total of four DCs have been planned. DC0 was a simple readiness test for the ATLAS software. DC1 [1], from spring 2002 to spring 2003, was the first full chain test of the ATLAS computing model. It produced over 30 TB of data and used about 75000 CPU days. Although mostly batch queues at large facilities were used, initial steps were taken to use Grid resources. DC2, started in summer 2004, was designed to employ a fully automated Grid-based production system, with batch production to be used as backup. The next step, DC3, will test the full computing infrastructure for the commissioning of the ATLAS apparatus in 2006 and the initial data taking in 2007.

ATLAS DC2 OVERVIEW

The ATLAS computing model has a hierarchical structure with a single Tier0 center located at CERN, several Tier1 regional centers that provide a reliable replicas of raw and processed datasets and additional facilities for production and analysis, and associated Tier2 centers that will support and coordinate the activities of physics working groups including user-based Monte Carlo production and analysis of highly summarized datasets. The current organization of computing resources is following the Grid paradigm which facilitates controlled sharing of distributed resources. There are currently three different Grids being exploited by ATLAS: the LHC Computing Grid, LCG [13], which is deployed on about 70 sites worldwide, the mostly Scandinavian-based NorduGrid project [5], the U.S.-based Grid3 [14]. Interoperation between Grids was possible but the main operational mode in DC2 was to coordinate the activity at the higher levels in the application/middleware stack, namely at the ATLAS production database level. Then each Grid system could operate independently, using abstract job definitions from the production database. Most ATLAS resources in the US were organized into Grid3, with the Tier1 facility located at the Brookhaven National Lab (Upton, NY)[15] and prototype Tier2 centers at Boston, Chicago, and Indiana, as well other ATLAS and non-ATLAS sites in Grid3.

DC2 was launched in June 2004, and consisted of three phases: Phase I, large scale production of physics datasets (order 10M events) with a chain of production steps including Pythia event generation, GEANT4-based simulation, digitization and pileup. The produced

*marco@hep.uchicago.edu
#rwg@hep.uchicago.edu

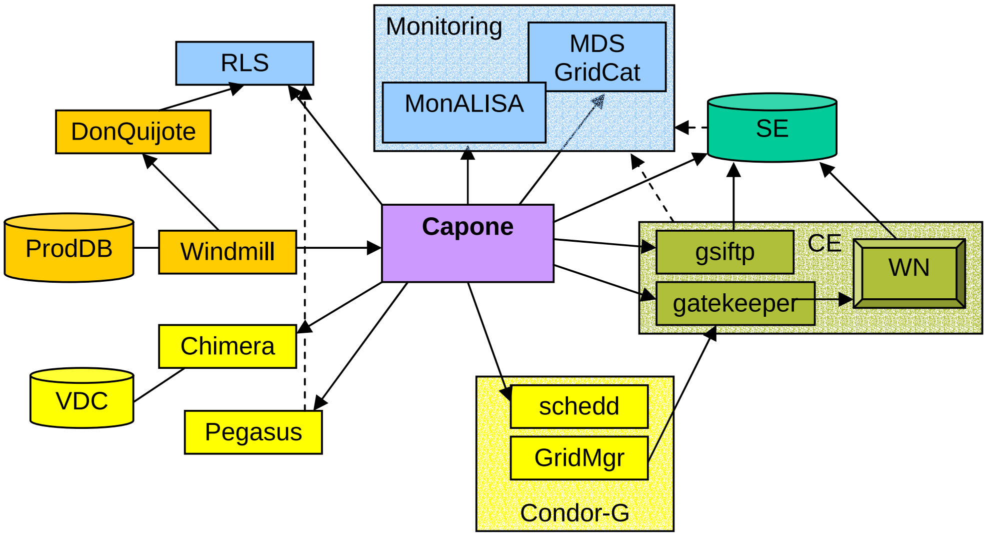

Figure 1: Elements of the system used to run ATLAS Data Challenge 2 on Grid3

datasets were stored on Tier1 centers (Brookhaven lab for Grid3) from where they were to be streamed back to the Tier0 at CERN. Phase II focused on Tier0 functionality testing with the aim of reconstructing the events at 1/10 of the full scale operation, with datasets streamed down to the Tier1 centers. Phase III, to be exercised until December 2004, focused on distributed analysis of the events reconstructed and access to event and non-event data from anywhere in the world both in organized and chaotic ways. During Phase I, almost 45 million CPU days (in SI2K-CPU) were used to execute over 100K jobs, which produced nearly 8 million events and about 30 TB of data. The three Grids contributed with almost an equal amount of work each.

Other papers document the overall DC2 effort [6] or the subsystem for production in other Grids [7, 8]; here we describe the production system for DC2 in Grid3.

GRID3 PRODUCTION SYSTEM

The production system in Grid3 follows the ATLAS approach that involves a supervisor, Windmill [6], interacting with the production database and one or more executors that manage all Grid interactions. The Grid3 executor system, Capone, communicates with the supervisor and handles all the interactions with Grid3 resources and services. Figure 1 shows the main elements of the production system. Job requests from the supervisor are taken by Capone that interfaces to a number of middleware clients on the Virtual Data Toolkit (VDT). These are used with information in grid servers to submit jobs to the grid. The rest of this section provides a more detailed description of this system.

Abstract ATLAS job parameters are delivered to Capone in an XML formatted message from the supervisor. These messages include job-specific information such as specific parameters expected by the transformation (the executable), identification of input and output files, and resource-specific requirements for the job. Capone receives these messages as either Jabber or Web service requests and translates them into its own internal format before beginning to process the job.

The transformation to be executed is determined from the input data contained in the supervisor message. The Globus RLS (Replica Location Sever, a distributed file catalog [12]) is consulted to verify the existence of any required input files and serves metadata information associated with these files.

The next step is to generate an abstract directed-acyclic graph (DAG) which is used to define the workflow of the job. Chimera [10] uses abstract transformation definitions defined in a VDC (Virtual Data Catalog) to produce a concrete DAG defining the workflow that is necessary to produce the desired output.

The job can now be scheduled using a concrete DAG generated from the previous abstract DAG with the addition of defined compute and storage elements. The CE (Compute Element, CPU where the job will be performed) and SE (Storage Element, the disks where the results will be stored) are chosen among those in a pre-defined pool of Grid3 resources. To choose the resource to select, it is possible to use different static algorithms like RR (Round Robin), WRR (Weighted Round Robin) or random, and some dynamic ones that account for site loads.

The submission and subsequent monitoring of the jobs on the grid is performed using Condor-G [11]. On the CE, if necessary, all input files are first staged-in before the execution proceeds. The ATLAS software itself, an Athena executable, is invoked from a sandbox using a VDT supplied executable called *Kickstart* [10]. A wrapper script, called by *K*ickstart and specific to the transformation to be executed, is called first to ensure that the environment is set up correctly before starting any ATLAS-specific executables. The wrapper script also checks for errors during execution and reports results back to the submitter through *K*ickstart. In addition, the wrapper script performs additional functions such as evaluating an MD5 checksum on all output files. The Condor status of each job is checked periodically by Capone and whenever the remote job completes, Capone resumes the control of the job.

Capone checks the results of the remote execution: the program's exit codes and the presence of all the expected output files. This is a delicate step since a large variety of errors may be discovered here, ranging from IO problems encountered during the stage-in process, to errors in the execution of Athena, to site characteristics that prevent Condor or *Kickstart* from exiting correctly.

The next step towards the job completion is the stage-out of the output files that are transferred from the data area in the CE to the output SE. The transfer also involves evaluation of the MD5 checksum and file size of the destination file(s), to check the integrity of the transfers. Furthermore some important metadata, like the GUID (a globally unique file identifier), is recovered from service files in the remote execution directory.

Finally there is the registration of the output files to RLS inserting LFN, PFN and metadata attributes required by Don Quijote [9], the ATLAS data movement utility that allows data transfers between grids.

The job is now successfully completed and status information will be returned as such to Windmill in the next response to a job status request. If the job fails in any of the previous steps, the attempt is declared failed and a new attempt will start the same job again. To avoid infinite loops the number of possible automatic attempts for the same job is limited. The status of the job was returned to Windmill also during the previous steps.

Before marking the job as successfully completed in the production database, Windmill, together with Don Quijote, perform a verification of the registered data.

RESULTS AND LESSONS LEARNED

DC2 is, for ATLAS, the first comprehensive test of the Grid infrastructure. The production activity on Grid3 is still in progress at the time of submission and has proven to be a very useful exercise. The exercise has allowed ATLAS to evaluate the validity of sharing resources on a grid, stressing the capacities of the whole infrastructure, and providing numerous benchmarks to critically review the errors and process structure.

The Grid3 production grid, used for DC2 production, consists of a number of distributed resources. The number of available CPUs is variable since they can be dynamically added or removed, but the maximum number seen was 3000. DC2 rarely used more than 1000 CPUs simultaneously on Grid3 since some available machines could not meet the speed, memory or network requirements for the jobs. Others were not available because of technical problems or software incompatibilities. Others were shared with different user groups and thus were not always available to ATLAS. In the last two months, more than 60 CPU years were consumed producing DC2 events. This far exceeded the number of resources available solely from CPUs dedicated to ATLAS users. As shown in Figure 2, less than half of the production has been done on dedicated resources, while a significant fraction has been done on clusters leveraged by ATLAS at universities and national laboratories. Note that more than a quarter has been done with opportunistic usage of resources contributed to Grid3 by other VOs (Virtual Organizations).

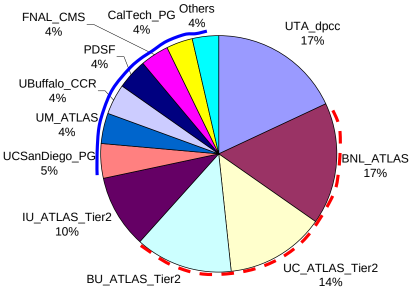

Figure 2: Resources used in US ATLAS DC2: the dashed curve represents dedicated contributions while the solid curve is leveraged on opportunistic contributions from Grid3.

If any error at all occurred during the processing of a job, that attempt was marked as a failure and a new attempt was submitted. The success rate of the end-to-end process is around 70%. Although this number may seem low, the practical success rate is higher since many failures resulted from trivial user problems at the start of the job. In this case no CPU resources were wasted. In terms of CPU resources usage the operation efficiency achieved 93%. Table 1 shows a break down of the failures depending on the processing step where they occurred. Failures in the early steps are less expensive, since almost no resources have been used on the Grid and the attempt is quickly recovered. Failures in the latter steps, like stage-out or RLS registration, are more expensive since the job may have been executing successfully on a CPU for more than 15 hours and may have produced valid output files. It should be noted that of the failures characterized as "Stage out" in Table 1, about 50% were actually remote execution failures due to wrongly failure codes during the initial phase of our project. Problems occurring at the CE included: failures related to gatekeeper overload, hardware failures and NFS problems. Problems detected at submit hosts included overloaded resources, hardware and/or work failures, grid certificate expiration, and operational errors. Due to the absence of central workflow management in Grid3, the submit host is very critical component since a hardware failure, a program crash, or a loss in network connectivity for a long period of time could cause the failure of the managed jobs at that time ('Capone host interruptions' in

Table 1). Therefore, ongoing work is being done to add additional submit host recoverability at all steps in the process.

Table 1: Capone Failures during Phase I of DC2.

Category	Number
Job completed (validated)	37713
Job attempts failed	19303
• Submission	472
• Execution	392
• Remote execution check	1147
• Stage out	8037
• RLS registration	989
• Capone host interruptions	2725
• Windmill failure	57
• Other	5139

The submit host was not the only critical element in the system. There were several points of failure, both in the system and in the process itself. Several core servers such as the production database at CERN, the Jabber messaging server, the RLS, VDC, and Don Quijote servers were necessary in order to be able to run the production system. Except for the production database server, all of the other servers are all located at BNL[15]. Even though these servers receive 24/7 surveillance, there were several failures and system halts when they were unavailable. Planned and un-planned network maintenance activities introduced bottlenecks in the production activity. This is a highly undesirable situation for a grid and is being revised with failover systems.

Operational difficulties included the usual problems associated with large, integrated systems. Troubleshooting failures in one component we often found that were the symptom of a failure in another. With many systems in play, the expertise required to troubleshoot problems was also distributed among many team members, creating an artificial ‘single point of failure’. The situation could have been helped with better documentation and with more co-development to spread the knowledge base. Better tools for diagnosing end-to-end grid applications are needed.

We have found that tools for Grid3 and DC2 production status monitoring are very helpful for efficient utilization of grid capacities. Because of a high degree of correlations between the failed jobs the monitoring tools providing immediate feedback were instrumental in efficient problem resolutions.

CONCLUSIONS

The Windmill-Capone system has proven to be a successful tool for US ATLAS DC2 production, and Grid3 a stable and reliable platform. Over the course of three months during July-September 2004, about one third of ATLAS DC2 production has been executed on Grid3, keeping pace with peer projects using NorduGrid and the LCG. Over the course of the project, efficiencies steadily improved as lessons learned at production scales were incorporated back into the system.

We found that two areas of grid infrastructure are critical to distributed frameworks: job state management, control and persistency, and troubleshooting failures in end-to-end integrated applications. The next steps in the project will make progress in those areas.

The evolution of Capone is towards increasing its reliability, for example implementing robustness to Grid failures, and making it more flexible to support user-based production during the distributed analysis phase of DC2 phase.

ACKNOWLEDGMENTS

This work was supported in part by the US Department of Energy and by National Science Foundation Grants ITR-0122557 and PHY/ITR-0113343. Argonne National Laboratory's work was supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, under contract W-31-109-Eng-38.

REFERENCES

[1] G. Poulard, “ATLAS Data Challenge 1”, *CHEP*, 2003, La Jolla, San Diego, California.

[2] http://lhc-computing-review-public.web.cern.ch/lhc-computing-review-public/Public/Report_final.PDF

[3] A Toroidal LHC ApparatuS, http://atlas.web.cern.ch/

[4] LHC Computing Grid http://lcg.web.cern.ch/lcg/

[5] NorduGrid http://www.nordugrid.org

[6] L. Goossens “ATLAS Production System in ATLAS Data Challenge 2”, *CHEP*, 2004

[7] O. Smirnova “Performance of the NorduGrid ARC and the Dulcinea Executor in ATLAS Data Challenge 2”, *CHEP*, 2004, Interlaken, Switzerland

[8] D. Rebatto “The LCG-2 Executor for the ATLAS DC2 Production System”, *CHEP*, 2004

[9] M .Branco, “Don Quijote - Data Management for the ATLAS Automatic Production System”, *CHEP*, 2004

[10] I. Foster, J. Voeckler, M. Wilde, Y. Zhao “The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration”, *CIDR*, 2003, Pacific Grove, http://www.griphyn.org/chimera/

[11] J. Frey, T. Tannenbaum, I. Foster, M. Livny, and S. Tuecke, “Condor-G: A Computation Management Agent for Multi-Institutional Grids”, *HPDC10*, 2001, San Francisco, CA, http://www.cs.wisc.edu/condorg/

[12] A. Chervenak, E. Deelman, I. Foster, et al. “Giggle: A Framework for Constructing Scalable Replica Location Services”, SuperComputing, 2002

[13] LHC, Large Hadron Collider http://lhc.web.cern.ch/

[14] The Grid2003 Project “The Grid2003 Production Grid: Principles and Practice”, *HPDC13*, 2004, Honol., HI, Grid2003, http://www.ivdgl.org/grid2003

[15] X.Zhao, et al. “Experience with Deployment and Operation of the ATLAS Production System and the Grid3+ Infrastructure at BNL”, *CHEP*, 2004