Evolution and experience with the ATLAS Simulation at Point1 Project

To cite this article: S Ballestrero et al 2017 J. Phys.: Conf. Ser. 898 082012

View the article online for updates and enhancements.
Evolution and experience with the ATLAS Simulation at Point1 Project

S Ballestrero¹, F Brasolin², D Fazio³, A Di Girolamo³, T Kouba⁴, C J Lee⁵,⁶,⁷, D A Scannicchio⁶, J Schovancová³, M S Twomey⁷, F Wang⁸ and A Zaytsev⁹

¹ University of Johannesburg, South Africa
² Istituto Nazionale di Fisica Nucleare Sezione di Bologna, Italy
³ CERN, Switzerland
⁴ Institute of Physics, ASCR, Czech Republic
⁵ University of Cape Town, South Africa
⁶ University of California, Irvine, USA
⁷ University of Washington Department of Physics, USA
⁸ University of Wisconsin-Madison, USA
⁹ Brookhaven National Laboratory (BNL), USA

Email: Tomas.Kouba@cern.ch, atlas-adc-sim-p1@cern.ch

Abstract. The Simulation at Point1 project is successfully running standard ATLAS simulation jobs on the TDAQ HLT resources. The pool of available resources changes dynamically, therefore we need to be very effective in exploiting the available computing cycles. We present our experience with using the Event Service that provides the event-level granularity of computations. We show the design decisions and overhead time related to the usage of the Event Service. The improved utilization of the resources is also presented with the recent development in monitoring, automatic alerting, deployment and GUI.

1. Introduction

During the LHC [1] Long shutdown 1 (LS1) period (started in March 2013 and finished in March 2015), the Simulation at Point1 (Sim@P1) project utilized the TDAQ HLT [2] (Trigger and Data Acquisition system, High Level Trigger) resources very well in a continuous manner. The TDAQ HLT farm provides more than 1300 compute nodes, which are particularly suited for running event generation and Monte Carlo production jobs that are mostly CPU bound, while not I/O bound. With the Sim@P1 project we leverage unused compute cycles that would otherwise be wasted, and thus improve utilization of available compute resources of the ATLAS experiment. The basic description, design decisions and the performance observed of the project itself are described in [3].

LHC Run 2 period (started in April 2015) presented slightly different challenge as the Simulation at Point1 project successfully continued in using the resources reserved for Sim@P1 and we focused on better utilization of the resources given to Sim@P1 during the periods when the resources are not needed for TDAQ HLT computing. Such periods are usually Machine Development (MD) or Technical Stop (TS) periods which usually last several days. We also deployed a Web User Interface (so called “red button”) to ease the switch the resources from/to TDAQ and Sim@P1 role.
There was no major change in the cloud setup and the major version of the operating system setup for Sim@P1. The software stability is a key aspect for TDAQ HLT smooth running, therefore we stayed on SLC6 as the operating system and OpenStack in version “Icehouse”. The newer versions of OpenStack require higher major version of the operating system (at least 7) and since this operating system is directly used to run the TDAQ workload (without any virtualization layer) we could not update neither the operating system nor the OpenStack instance.

So far the overall performance of the Sim@P1 was very good during the Run 2. In terms of generated and simulated events the performance of Sim@P1 is comparable to the performance of the largest tier one (T1) centers in the WLCG [4].

The Figure 1 shows the number of completed jobs since March 2015. Each column represents one week. The various colors represent different job types. Each spike is caused by sudden increase of available computing resources. These resources were available for the Sim@P1 project because there was a stop in LHC data taking and the TDAQ HLT system did not need the resources. The figure 1 also shows that we tried to run reconstruction jobs during some periods. These tests produced high disk load on the servers which was not acceptable for the TDAQ monitoring system so we never managed to run reconstruction jobs for longer (more than one month) periods.

The Figure 2 depicts the biggest portions of all simulation jobs processed by each ATLAS computing site (including only T0, T1 sites and Sim@P1). The chart shows that Sim@P1 (the site is called CERN-P1) contributed more than 12% of all the events simulated for ATLAS since March 2015. Please note that the pie chart in the Figure 2 is limited only to simulation jobs and other sites are not dedicated to simulation only jobs (Sim@P1 is exclusively used for simulation jobs except short periods of tests of other job types – as seen on Figure 1). However the simulation jobs represent the most resource hungry activity and ATLAS has spent 65% of CPU time on simulation since March 2015 till the August 2016.

Figure 1. Sim@P1: Jobs completed since March 2015
2. Event Service pilot test

As described in [5] the ATLAS Event Service (ES) implements a new fine grained approach to HEP event processing, designed to be agile and efficient in exploiting transient, short-lived resources.

Special ES jobs are defined with a range of events to simulate and they send the results regularly to a defined storage (to protect from a sudden shutdown of local system). The jobs also confirm (to a central scheduling system) the range of events that were successfully simulated so no other job needs to retry these events. If the job does not finish (e.g. it is interrupted due to the system shutdown) the central scheduling system is able to reschedule the missing event range to another ES job. More details can be found in [5].

This approach perfectly fits for the short (i.e. hours) bursts in the number of CPUs that are assigned from their TDAQ role to the Sim@P1 role. These short periods are not long enough to finish a regular grid job (which usually takes around 8 hours) but still long enough to process significant number of events. The regular simulation jobs will fail and lose all progress when the resources vanish, but the Event Service jobs with the event-by-event processing and staging-out can minimize the loss of the resource changes.

The Sim@P1 project has piloted in the Event Service (ES) test as the first opportunistic site (a site where the number of available CPUs changes every few minutes).

Many feedbacks and suggestions have been provided to the Event Service team during the test, which helped to improve its performance and fix bugs.

In order to demonstrate usefulness of the ES mechanism the Figure 3 below shows the amount of lost jobs when the resources are taken back from Sim@P1 to the regular TDAQ HLT operation. In such situation the regular jobs are simply killed and all simulated events are lost.

3. Automated switcher: “red button”

The process of switching the resources from TDAQ HLT to Sim@P1 role is quite complex and needed a supervision of TDAQ experts. To exploit short periods of time when TDAQ servers are available for simulation purposes we needed a mechanism that would allow shifters (people following carefully TDAQ monitoring and reacting to problems) to easily switch the role of the resources (rack by rack) from TDAQ to Sim@P1 and back. Such mechanism must be easy to use for shifters, granular enough to allow switch the role rack by rack and very robust so the resources do not stay in undefined
state (somewhere between Sim@P1 and TDAQ). Another critical feature of the whole switch is that the resources must be safely and quickly released by Sim@P1 (even on short notice) when they are needed by TDAQ.

Figure 3. Failed jobs in regular simulation tasks for short periods

The web interface implementing all these required features is shown in Figure 4.

Any shifter on duty can “push the red button” which means he/she can select the TDAQ servers (by whole racks, not one server by another) and change their role from TDAQ HLT to Sim@P1 role and vice versa. This operation only changes the role of the servers in the TDAQ database. A daemon running on all servers then starts (or stops) virtual machine for Sim@P1 workload and these virtual machines become automatically (during the boot) part of the Sim@P1 computing cluster. The details of the switching procedure are described in [3].

4. Monitoring developments

The monitoring of all the Sim@P1 related activities had been consolidated on one live page [6] and the Sim@P1 responsibles are notified if the utilization (ratio between existing job slots and running ATLAS jobs) drops below 80%. These two values are depicted in figure 5. The high peak at the end of July 2016 shows a technical stop of LHC when TDAQ shifters assigned many resources to Sim@P1 but the ATLAS grid scheduling and job submission was not able to fill the resources with jobs – hence the big difference between available slots and running jobs. Good utilization of Sim@P1 resources is a responsibility of ATLAS grid operations not TDAQ shifters nor TDAQ operators.
During the year 2016 we tested using the Sim@P1 resources not only for simulation jobs but we tried reconstruction and reprocessing jobs too. These jobs brought higher I/O load to the nodes. This triggers notification to TDAQ administrators and also to the Sim@P1 administrators. We have not yet fully identified the exact root cause of the high I/O so we went back to simulation jobs only at the end of 2016.
5. Dynamic partitioning and Cgroups

The jobs running at Sim@P1 computing cluster are managed with a batch system HTCondor. Since the beginning of the Sim@P1 we have been trying to configure it in a way that would allow the batch system to cope with sudden changes in the number of available computing servers and with various requirements of ATLAS simulation jobs (e.g. different RAM requirements).

During 2016 we switched from static HTCondor slots (8 CPU, no RAM limit) to dynamically allocated slots of two types (8 CPU, 8 GB RAM + 8 CPU, 15 GB RAM) and one additional type (1 CPU, 2 GB RAM) for single core jobs (particularly “merge” jobs – jobs reading input files with simulated events and writing one big file and so ensuring that each final output has enough events for future work).

The limits are enforced with Linux Cgroups [8] with the goal not to overcommit the memory usage on the virtual machines and so not to trigger swap usage on the hypervisors outside of VMs. This helped to test the reconstruction jobs that often had caused memory exhaustion. However, we still see heavy I/O load caused by the reconstruction jobs. This needs to be fully understood before we allow more reconstruction and reprocessing jobs to be run at Sim@P1.

6. Summary

During the LHC Run2 the Sim@P1 project continued to deliver a significant portion of simulated events for ATLAS.

There was no significant upgrade in the infrastructure as it needs to be kept stable for correct TDAQ HLT operations. Any discussed upgrades were postponed to Long Shutdown 2 (planned for years 2019 and 2020). This includes OpenStack version upgrade which requires operating system upgrades (from SLC6 to CC7). Such major upgrade needs careful planning as it involves also validation of ATLAS TDAQ software.

The future plans are now focused mainly on resolving spurious problems with high I/O load caused by reconstruction jobs on the TDAQ nodes and on commissioning Event Service jobs from test phase to production.

References

[1] Brüning O S, Collier P, Lebrun P, Myers S, Ostojic R, Poole J, Proudlock P, LHC Design Report, ISBN 92-9083-224-0
[2] ATLAS Collaboration, ATLAS High Level Trigger, Data Acquisition and Controls: Technical Design Report, CERN/LHCC/2003-022 (2003), ISBN 92-9083-205-3
[3] Zaytsev A et al., Design, Results, Evolution and Status of the ATLAS Simulation at Point1 Project, J.Phys.Conf.Ser. 664 (2015) no.2, 022008
[4] Worldwide LHC Computing GRID: http://wlcg.web.cern.ch
[5] Calafiura P, De K, Guan W, Maeno T, Nilsson P, Oleynik D, Panitkin S, Tsulaia V, Van Gemmeren P, Wenaus T, The ATLAS Event Service: A new approach to event processing, J.Phys.Conf.Ser. 664 (2015) no.6, 062065
[6] ATLAS Sim@P1 processing overview, http://adc-mon.cern.ch/SimAtP1/ATLAS-SimAtP1-live.html
[7] Computing with HTCondor, http://research.cs.wisc.edu/htcondor/
[8] Introduction to Cgroups, https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Resource_Management_Guide/ch01.html