Grid production with the ATLAS Event Service

Esteban Fullana Torregrosa[1,*], *Doug* Benjamin[4], *Paolo* Calafiura[5], *Taylor* Childers[4], *Kaushik* De[3], *Alessandro* Di Girolamo[2], *Jose-Enrique* Garcia Navarro[1], *Wen* Guan[8], *Mario* Lassnig[2], *Tadashi* Maeno[6], *Paul* Nilsson[6], *Vakhtang* Tsulaia[5], *Peter* Van Gemmeren[4], *Torre* Wenaus[6], and *Wei* Yang[7]; on behalf of the ATLAS collaboration.

[1]IFIC-(Univ. of Valencia and CSIC) (ES)
[2]CERN-Switzerland
[3]University of Texas at Arlington (US)
[4]Argonne National Laboratory (US)
[5]Lawrence Berkeley National Lab. (US)
[6]Brookhaven National Laboratory (US)
[7]SLAC National Accelerator Laboratory (US)
[8]University of Wisconsin (US)

Abstract. ATLAS has developed and previously presented a new computing architecture, the Event Service, that allows real time delivery of fine grained workloads which process dispatched events (or event ranges) and immediately streams outputs. The principal aim was to profit from opportunistic resources such as commercial cloud, supercomputing, and volunteer computing, and otherwise unused cycles on clusters and grids. During the development and deployment phase, its utility also on the grid and conventional clusters for the exploitation of otherwise unused cycles became apparent. Here we describe our experience commissioning the Event Service on the grid in the ATLAS production system. We study the performance compared with standard simulation production. We describe the integration with the ATLAS data management system to ensure scalability and compatibility with object stores. Finally, we outline the remaining steps towards a fully commissioned system.

1 Introduction

ATLAS Event Service (AES) [1] is a new approach to event processing capable of the finest granularity in every step of the computing chain: input, processing and output.

Today's excellent networks, distributed federated data access (using xrootd [2]) and high scalable object store [3] allows for a dynamic, flexible and distributed workflows that adapt in real time to resource availability. AES is the implementation of such a workflows in the ATLAS [4] computing infrastructure.

AES has been successfully implemented in opportunistic resources such as High Performance Computers (HPCs) (through the Yoda [5] interface) and Amazon Spot Cloud [6]. This paper describes the natural evolution of commissioning AES to opportunistic grid sites such as Tiers3 [7] and High Level Trigger farms (off data taking periods) as well as use in pledge resources.

*e-mail: Esteban.Fullana@ific.uv.es

2 Motivation and benefits

AES allows for event-granularity splits of the jobs making it beneficial for both opportunistic and pledge resources.

In opportunistic resources is important to be robust against their disappearance with minimal losses. The event granularity processing and delivery of the output immediately after completion allows for a minimal loss once the resource disappears.

The flexibility of event service also improves the use of pledged resources. First the event-granularity in each task can be configured to make it equivalent to the standard approach and second it allows for a very efficient use of each CPU cycle; plus the flexibility to allocate more resources for high priority tasks.

The current and future computing needs of the ATLAS detector will require both the optimal use of current pledge resources and the efficient use of opportunistic resources. AES is therefore the logical approach for the future of ATLAS computing.

3 Architecture and workflow

The architecture of AES is based upon three pillars: the PanDa distributed manager [8], JEDI for flexible dynamic workflow management [9] and Athena MP [10], a process-parallel version of the ATLAS data processing framework Athena; plus a fourth pillar (Yoda) when it is used in HPCs.

The bricks that carry the workload are the pilot jobs. They are autonomous jobs that request to Panda the dispatch of the work. The JEDI extension enables Panda to dynamically partition and manage workflows down to the event level granularity. JEDI also manages the bookkeeping in the PANDA Oracle database. The payloads requested by the pilots to Panda are instances of AthenaMP that manages the distribution and processing of events concurrently among parallel workers. Figure 1 shows an schematic of AES workflow. First,

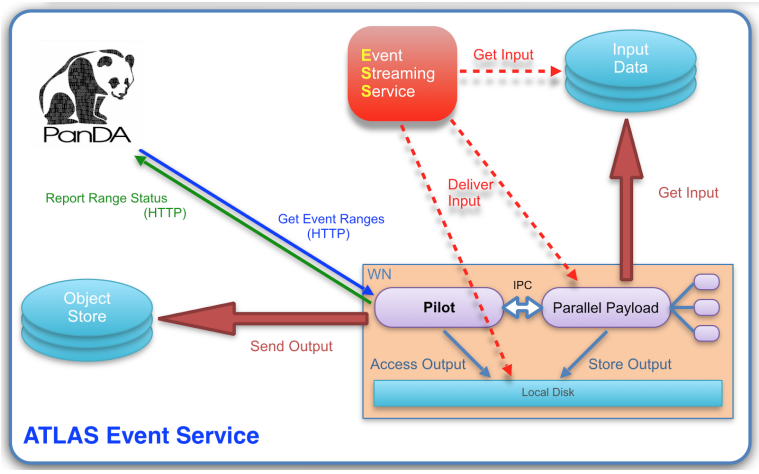

Figure 1. Schematic view of Event Service workflow

on the compute node, the PanDA Pilot connects with the PanDA server over HTTP and starts a parallel event processing application that allows for the use of all available CPU cores. The payload application in AES is represented by AthenaMP. AthenaMP starts as a serial process, which first goes through the application initialization phase, then forks several event

processors (workers) and informs the pilot that it is ready for data processing. The pilot downloads event range identifiers (strings) from the PanDA server and delivers them in real time to the running AthenaMP application, which assigns them to its workers on a first-come, first-served basis. The worker uses the event range string to locate the corresponding input file and find event range data within the file. Once the event range is processed, the worker writes the output into a separate file on the local disk and declares its readiness to process another event range. AthenaMP reports back to the Pilot the locations of output files produced by its workers and the Pilot takes care of streaming the outputs in real time to a remote storage system (Object Store), and informing the PanDA server of the event range completion status. Once all the event ranges have been processed a separate job, a merging job, collects all the inputs and merges them in a single file that is uploaded to the storage element. Thus the final output file is equivalent to the output of the simulation using the standard framework and transparent for the user.

4 ATLAS Event Service commissioning on the grid

The AES is being gradually commissioned on the grid following three stages. In the first stage, a physics validation effort was put in place to validate the output of AES jobs against standard simulation processing. In a second stage several simulation tasks, low priority, were manually assigned to be run under the AES framework. Finally in the last stage the simulation tasks, under certain conditions, were automatically assigned to run under AES.

4.1 Physics validation

Two different strategies were put in place for the physics validation of the output samples simulated under the AES framework. First a small set of events were simulated using both AES and the current standard framework with the same seed in the random number generator. After the proper ordering the two outputs perfectly match digit by digit.

Second a larger sample was simulated again through both frameworks, AES and standard. The output was compared using official ATLAS physics validation tools and found to be compatible within the statistical uncertainty.

As a summary no sign of any physics bias was found in the simulation samples processed under the AES framework.

4.2 Manual assignment of tasks

After the physics validation a set of low priority simulation tasks were manually assigned to be run under the AES framework. During this stage the tasks were both assigned to all sites (properly configured to run AES) and manually assigned to specific site. In parallel the same task was simulated in no AES mode for performance comparisons.

The success of this exercise, with minor problems, either site-related, either easily solved by tuning AES configuration parameters build up the confidence to move to the next commissioning step.

4.3 Automatic assignment of tasks

In August 2017 ATLAS started to automatically assign tasks to be simulated through the AES framework. Two conditions were needed for the automatic assignment, first the queue of events that were waiting to be simulated should be lower than a certain threshold, and second the task should have low priority.

The automatic assignment of tasks helped to understand issues related with the scaling of events simulated under the framework, specially the performance of object stores -as a new storage technology- when the demand to store event-granularity files increase.

Figure 2 shows the number of jobs (in green) and number of slots (jobs times the number of cores in each job) that were running under AES for 100 days starting March 2018. The number of jobs and slots depends on the ATLAS needs to run simulation jobs and the availability of opportunistic or pledge resources configured to run AES. Also notice that most of the resources are single core and most likely not used without the AES framework.

Figure 3 shows the number of events simulated in one week for a consecutive period of 90 days. Again depending on the availability of resources the number of events produced varies but on average up to tens of millions of events are simulated each week under the AES framework.

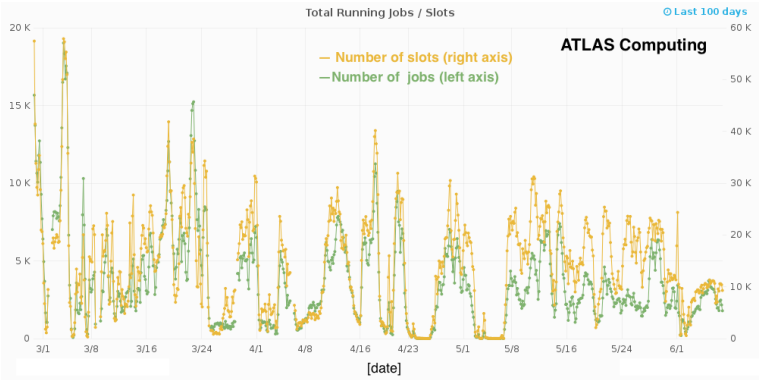

Figure 2. Number of event service jobs (green) and slots (number of cores, in yellow) during 100 days starting March 1st 2018.

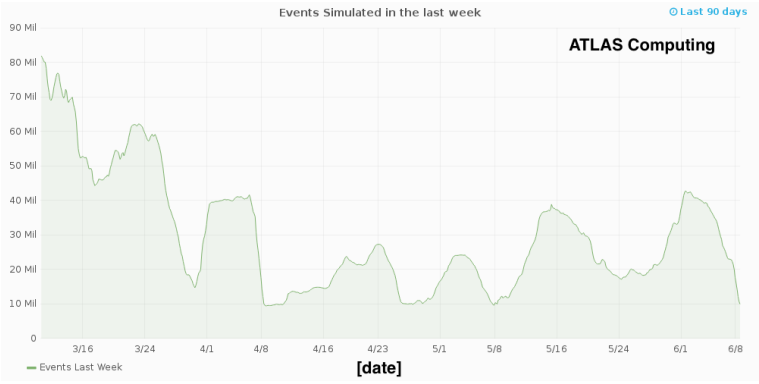

Figure 3. Number of events simulated per week (running weeks) during 90 days starting March 12th 2018.

5 Monitoring and testing tools

The deployment of AES on the grid is completed with the development of specific tools both to monitor AES jobs and events and validate sites running AES jobs. On monitoring, the ATLAS computing monitoring group developed ad hoc web pages for the AES jobs using ATLAS official tools. In parallel an exploratory monitoring page was set in place to test and identify useful monitoring indicators of the good performance of AES and spot sources of problems. In addition a policy was set in place to check the validity of sites to run AES jobs using HammerCloud [11] tools. The HammerCloud jobs systematically check for potential problem in AES sites. In general the deployment of AES help us to rethink good monitoring indicators and put in place the tools needed to isolate problems specifically related to AES and properly solve them.

6 Conclusions

AES has been commissioned to run ATLAS simulation jobs in both opportunistic and pledge resources. The commissioning has been put in place in three stages: in a small set of samples, with a close monitoring by the AES developing team, and then scaling up to tens of millions of events per week, plus a study to assess possible physics bias in samples simulated under AES.

The commissioning exercise includes the development of ad-hoc monitoring tools; both under the current ATLAS computing monitoring and also a workbench monitoring to find indicators to spot of sources of problems. This also includes specific HammerCloud tests to assess the availability of sites for AES jobs.

In summary the framework is ready to simulate all ATLAS events.

References

[1] P. Calafiura, K. De, W. Guan, T. Maeno, P. Nilsson, D. Oleynik, S. Panitkin, V. Tsulaia, P.V. Gemmeren, T. Wenaus, *The ATLAS Event Service: A new approach to event processing*, Journal of Physics: Conference Series **664**, 062065 (2015)

[2] R. Gardner, S. Campana, G. Duckeck, J. Elmsheuser, A. Hanushevsky, F.G. Honig, J. Iven, F. Legger, I. Vukotic, W. Yang et al., *Data federation strategies for atlas using xrootd*, J. Phys.: Conf. Series **513**, 042049 (2014)

[3] M. Mesnier, G.R. Ganger, E. Riedel, *Object-based storage*, IEEE Communications Magazine **41**, 84 (2003)

[4] ATLAS Collaboration, 2008 JINST 3 S08003, *The ATLAS Experiment at the CERN Large Hadron Collider*

[5] P. Calafiura, K. De, W. Guan, T. Maeno, P. Nilsson, D. Oleynik, S. Panitkin, V. Tsulaia, P.V. Gemmeren, T. Wenaus, *Fine grained event processing on hpcs with the atlas yoda system*, Journal of Physics: Conference Series **664**, 092025 (2015)

[6] D. Benjamin, P. Calafiura, T. Childers, K. De, W. Guan, T. Maeno, P. Nilsson, V. Tsulaia, P.V. Gemmeren, T. Wenaus et al., *Production experience with the atlas event service*, Journal of Physics: Conference Series **898**, 062002 (2017)

[7] M. Villaplana, S.G. de la Hoz, A. Fernandez, J. Salt, A. Lamas, F. Fassi, M. Kaci, E. Oliver, J. Sanchez, V.S.M. for the Atlas Collaboration, *ATLAS Tier-3 within IFIC-Valencia analysis facility*, Journal of Physics: Conference Series **396**, 042062 (2012)

[8] T. Maeno, *Panda: distributed production and distributed analysis system for atlas*, Journal of Physics: Conference Series **119**, 062036 (2008)

[9] K. De, D. Golubkov, A. Klimentov, M. Potekhin, A. Vaniachine, the Atlas Collaboration, *Task management in the new atlas production system*, Journal of Physics: Conference Series **513**, 032078 (2014)

[10] P. Calafiura, C. Leggett, R. Seuster, V. Tsulaia, P.V. Gemmeren, *Running atlas workloads within massively parallel distributed applications using athena multi-process framework (athenamp)*, Journal of Physics: Conference Series **664**, 072050 (2015)

[11] D.C. van der Ster, J. Elmsheuser, M.U. Garcia, M. Paladin, *Hammercloud: A stress testing system for distributed analysis*, Journal of Physics: Conference Series **331**, 072036 (2011)