Scaling up ATLAS Event Service to production levels on opportunistic computing platforms

D Benjamin[1], J Caballero[2], M Ernst[2], W Guan[3], J Hover[2], D Lesny[4], T Maeno[2], P Nilsson[2], V Tsulaia[5], P van Gemmeren[6], A Vaniachine[7], F Wang[3] and T Wenaus[2] on behalf of the ATLAS Collaboration

[1] Duke University, Durham, NC 27708, United States of America
[2] Brookhaven National Laboratory, Upton, NY 11973, United States of America
[3] University of Wisconsin, Madison, WI 53706, United States of America
[4] University of Illinois at Urbana–Champaign, Champaign, IL 61801, United States of America
[5] Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States of America
[6] Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439, United States of America
[7] Tomsk State University, Lenina Avenue 36, 634050, Tomsk, Russian Federation

E-mail: sacha.vaniachine@cern.ch

Abstract. Continued growth in public cloud and HPC resources is on track to exceed the dedicated resources available for ATLAS on the WLCG. Examples of such platforms are Amazon AWS EC2 Spot Instances, Edison Cray XC30 supercomputer, backfill at Tier 2 and Tier 3 sites, opportunistic resources at the Open Science Grid (OSG), and ATLAS High Level Trigger farm between the data taking periods. Because of specific aspects of opportunistic resources such as preemptive job scheduling and data I/O, their efficient usage requires workflow innovations provided by the ATLAS Event Service. Thanks to the finer granularity of the Event Service data processing workflow, the opportunistic resources are used more efficiently. We report on our progress in scaling opportunistic resource usage to double-digit levels in ATLAS production.

1. Introduction

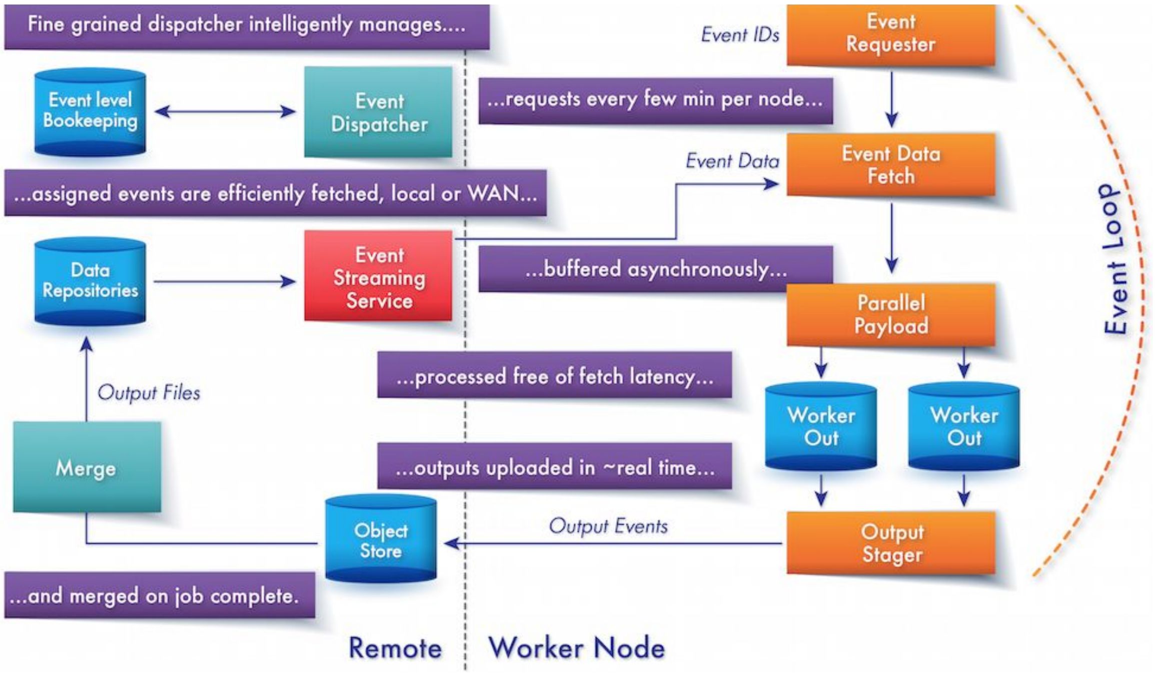
Using opportunistic resources efficiently and fully for ATLAS processing is an important means for maximizing ATLAS computing throughput within budget constraints. Examples of such opportunistic resources are:

- High Level Trigger (HLT) farm, available opportunistically during LHC Run 2;
- High Performance Computing (HPC), also known as supercomputers;
- Opportunistic grid resources, e.g. non-ATLAS OSG sites;
- Cloud resources, particularly, the Amazon EC2 spot market;
- Volunteer computing, also known as ATLAS@Home via BOINC.

Fine-grained event-based processing enables agile, efficient utilization of opportunistic resources that appear and disappear with unpredictable timing and quantity. Event consumers can be injected into such a resource when it becomes available, and participate in ongoing task processing, concurrently delivering outputs to an aggregation point until the resource disappears, with negligible losses if the resource disappears too suddenly.

2. Event Service

The new ATLAS production system prodsys2 [1] includes a PanDA extension JEDI [2] that adds to the PanDA server the capability to intelligently and dynamically break down tasks (as defined by the prodsys2 task definition component DEFT [3]) based on optimal use of the processing resources currently available. JEDI can break down tasks not only at the job level but also at the event level. The initial application of this capability is for event-level job splitting: jobs are defined in terms of the event ranges within files that they should process, and their dispatch and execution otherwise proceeds as normal. However this capability also presents the opportunity to manage processing and dispatch work at the event or event cluster level.



Fine grained dispatcher intelligently manages....
Event level Bookeeping
Event Dispatcher
...assigned events are efficiently fetched, local or WAN...
Data Repositories
Event Streaming Service
Output Files
Merge
...and merged on job complete.
Object Store
Remote
Event IDs
Event Requester
...requests every few min per node...
Event Data
Event Data Fetch
...buffered asynchronously...
Parallel Payload
...processed free of fetch latency...
Worker Out
Worker Out
...outputs uploaded in ~real time...
Output Events
Output Stager
Event Loop
Worker Node

Figure 1. Event Service architecture including the Event Streaming Service (work-in-progress).

2.1. Requirements

We have to be agile in how we use these opportunistic resources:

- Rapid setup of software, data and workloads (quick start) when they become available;
- Quick exit when resources are about to disappear;
- Robust against resources disappearing with no notice: minimal losses;
- Use resources until they disappear – harvest unused cycles, filling resources with fine-grained workloads.

2.2. Implementation

The Event Service (ES) [4] leverages excellent networks for efficient remote data access, distributed federated data access (using xrootd), and highly scalable object store technologies for data storage that architecturally match the fine-grained data flows, to support dynamic, flexible, distributed workflows that adapt in real time to resource availability and leverage remote data repositories with no data locality or pre-staging requirements (Figure 1).

2.3. Commissioning

During Event Service commissioning more that 150 test and/or validation tasks has been submitted and/or processed. Commissioning optimized the set of configuration parameters for production tasks for Event Service workflow execution. Table 1 shows the representative set of the validated configuration parameters for HPC and AWS sites.

Table 1. Event Service task configuration parameters.

Parameter	HPC	AWS
cmtconfig	x86_64-slc6-gcc48-opt	x86_64-slc6-gcc48-opt
spacetoken	ATLASDATADISK	ATLASDATADISK
cloud	US	US
site	NERSC_Edison	BNL_EC2E1_MCORE
skipScout	yes	yes
coreCount	8	8
nEventsPerWorker	1	1
nEsConsumers	1	1
esProcessingType	validation	validation
disableReassign	yes	yes
maxAttemptES	10	

Further optimization was achieved through comparison of various techniques necessary for event-by-event navigation. After detailed comparison of remote access to EventIndex [5], use of in-situ tag files [6], or use of events position in the input files, we chose the last technique as most robust for large scale production.

3. Lessons Learned

3.1. Results

Event Service production commissioning and operation scales up steadily. On the Amazon AWS platform the production level achieved up to 50k concurrent cores (Figure 2). Thanks to the attractive pricing of the AWS EC2 spot instances, the total costs were lower than those associated with certain ATLAS dedicated resources. To scale up the AWS productions level further, preparations are ongoing for the 100k core production run.

At the HPC platform at NERSC we successfully commissioned scaling tests of up to 50k concurrent cores, with production level achieving up to 25k concurrent cores. Following improvements in I/O throughput, the slope of simulation production CPU consumption sharply increased with two equal contributions: Yoda [7] and Generator [8] (Figure 3).

3.2. Work in progress

On the opportunistic HLT farm we are working with simulation production and scaling up in tests now ongoing towards 35k core scale. (Off) Grid platforms at various stages of commissioning and operation: backfill on conventional grid sites, US Tier 3, BNL shared Tier 3, campus resources and non-ATLAS Open Science Grid sites. The ARC/Event Service/Yoda integration is working, with ARC Control Tower [9] integration is underway. Also the ATLAS@Home [10] implementation is underway.

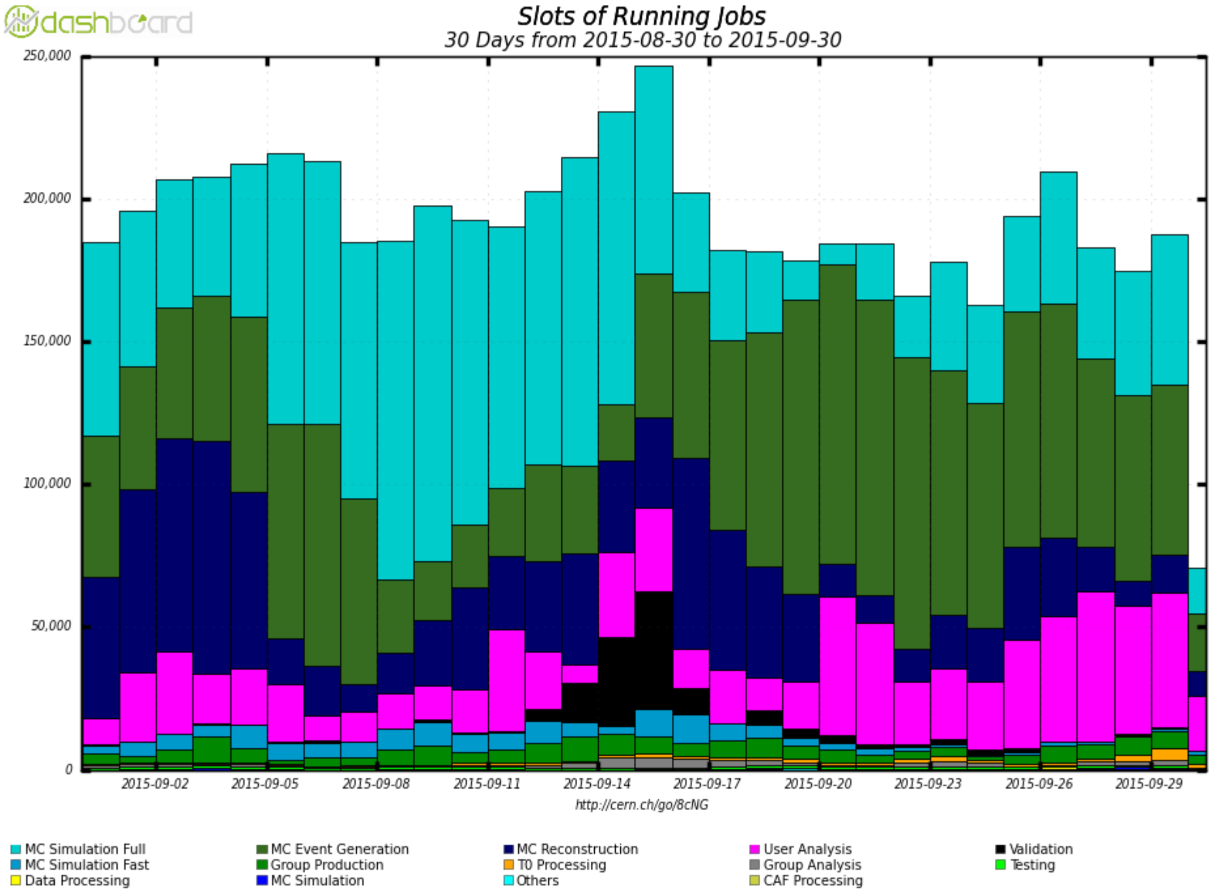


Figure 2. Event Service production level on AWS of up to 50k CPU-cores (black) contributed to the traditional of ATLAS resources of about 200k CPU-cores (colors).

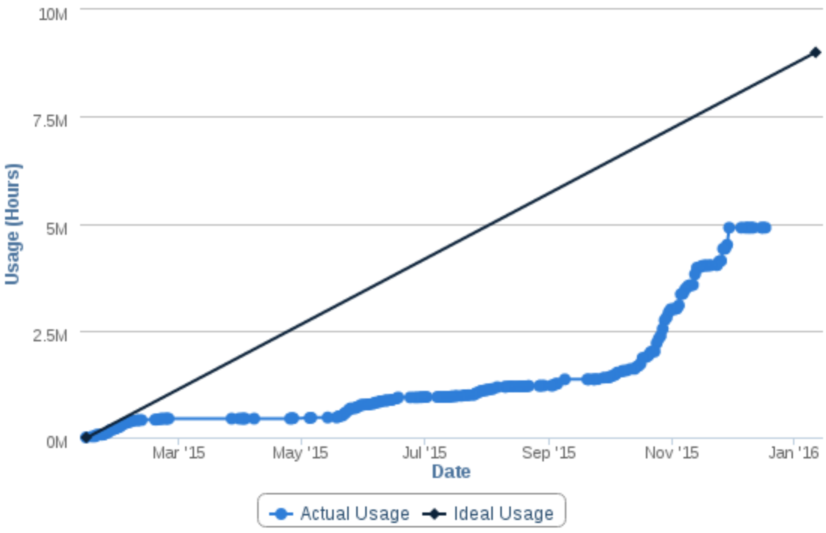


Figure 3. CPU usage of Advanced Scientific Computing Research Leadership Computing Challenge allocation on Edison.

4. Conclusions

The ATLAS Event Service enables cost-effective exploitation of opportunistic computing on heterogeneous, high capacity and high value platforms such as HPCs ('hole filling'), commercial clouds (dynamic market based spot pricing) and volunteer computing (no resource availability guarantees). In the next months we will continue efficient utilization of preemptible resources, particularly opportunistic such as HPC and AWS spot instances as well as backfilling dedicated ATLAS resources to keep them truly full. In general, the Event Service will serve as the basis for a uniform resource pool consuming in an automated and flexible way a predefined task pool.

Acknowledgments

We thank all our colleagues who contributed to the ATLAS Event Service development and operations. This work was funded in part by the U. S. Department of Energy, Office of Science, High Energy Physics under Contract No. DE-AC02-06CH11357.

References

[1] Borodin M, De K, Garcia Navarro J, Golubkov D, Klimentov A, Maeno T, South D and Vaniachine A on behalf of the ATLAS Collaboration (2015) Unified System for Processing Real and Simulated Data in the ATLAS Experiment. Proc. of the DAMDID/RCDL’2015 Conf. *CEUR-WS.org* **1536**:157;

Borodin M, De K, Garcia Navarro J, Golubkov D, Klimentov A, Maeno T and Vaniachine A on behalf of the ATLAS Collaboration (2015) Multilevel Workflow System in the ATLAS Experiment. Proc. of the ACAT2014 Workshop. *J. Phys.: Conf. Ser.* **608**:012015

[2] Maeno T, De K, Klimentov A, Nilsson P, Oleynik D, Panitkin S, Petrosyan A, Schovancova J, Vaniachine A, Wenaus T and Yu D on behalf of the ATLAS Collaboration (2014) Evolution of the ATLAS PanDA workload management system for exascale computational science. Proc. of the CHEP2014 Conf. *J. Phys.: Conf. Ser.* **513**:032062

[3] De K, Golubkov D, Klimentov A, Potekhin M and Vaniachine A on behalf of the ATLAS Collaboration (2014) Task Management in the New ATLAS Production System. Proc. of the CHEP2014 Conf. *J. Phys.: Conf. Ser.* **513**:032078

[4] Calafiura P, De K, Guan W, Maeno T, Nilsson P, Oleynik D, Panitkin S, Tsulaia V, van Gemmeren P and Wenaus T on behalf of the ATLAS Collaboration (2015) The ATLAS Event Service: A new approach to event processing. Proc. of the CHEP2015 Conf. *J. Phys.: Conf. Ser.* **664**:062065

[5] Barberis D *et al.* (2015) The ATLAS EventIndex: architecture, design choices, deployment and first operation experience. Proc. of the CHEP2015 Conf. *J. Phys.: Conf. Ser.* **664**:042003

[6] Cranshaw J, Doyle A T, Kenyon M J, Malon D, McGlone H and Nicholson C (2008) Integration of the ATLAS tag database with data management and analysis components. Proc. of the CHEP2008 Conf. *J. Phys.: Conf. Ser.* **119**:042008

[7] Calafiura P, De K, Guan W, Maeno T, Nilsson P, Oleynik D, Panitkin S, Tsulaia V, van Gemmeren P and Wenaus T on behalf of the ATLAS Collaboration (2015) Fine-grained event processing on HPCs with the ATLAS Yoda system. Proc. of the CHEP2015 Conf. *J. Phys.: Conf. Ser.* **664**:092025

[8] Childers J T, Uram T D, LeCompte T J, Papka M E and Benjamin D P (2015) Simulation of LHC events on a millions threads. Proc. of the CHEP2015 Conf. *J. Phys.: Conf. Ser.* **664**:092006

[9] Nilsen J K, Cameron D and Filipčič A (2015) ARC Control Tower: A flexible generic distributed job management framework. Proc. of the CHEP2015 Conf. *J. Phys.: Conf. Ser.* **664**:062042

[10] Adam-Bourdarios C, Cameron D, Filipčič A, Lancon E and Wu W on behalf of the ATLAS Collaboration (2015) ATLAS@Home: Harnessing Volunteer Computing for HEP. Proc. of the CHEP2015 Conf. *J. Phys.: Conf. Ser.* **664**:022009