ATLAS Sim@P1 upgrades during long shutdown two

Frank Berghaus[1,*], *Franco* Brasolin[2], *Alessandro* Di Girolamo[3], *Marcus* Ebert[1], *Colin Roy* Leavett-Brown[1], *Chris* Lee[4], *Peter* Love[5], *Eukeni* Pozo Astigarraga[3], *Diana Alessandra* Scannicchio[6], *Jaroslava* Schovancova[3], *Rolf* Seuster[1], and *Randall* Sobie[1,**] on behalf of the ATLAS Collaboration

[1]University of Victoria, Victoria, Canada
[2]Università e INFN, Bologna, Italy
[3]CERN, Geneva, Switzerland
[4]University of Cape Town, Cape Town, South Africa
[5]Lancaster University, Lancaster, United Kingdom
[6]University of California Irvine, Irvine, United States of America

Abstract. The Simulation at Point1 (Sim@P1) project was built in 2013 to take advantage of the ATLAS Trigger and Data Acquisition High Level Trigger (HLT) farm. The HLT farm provides around 100,000 cores, which are critical to ATLAS during data taking. When ATLAS is not recording data, this large compute resource is used to generate and process simulation data for the experiment. At the beginning of the current long shutdown (LS2), the HLT farm including the Sim@P1 infrastructure was upgraded. Previous papers emphasised the need for "simple, reliable, and efficient tools" and assessed various options to quickly switch between data acquisition operation and offline processing. In this contribution, we describe the new mechanisms put in place for the opportunistic exploitation of the HLT farm for offline processing and give the results from the first months of operation.

1 Introduction

ATLAS [1] is a general purpose experiment located at point one (P1) of CERN's large hadron collider (LHC). ATLAS employs a large computer farm, summarised in table 1, to facilitate data acquisition and event selection. The High Level Trigger (HLT) [2] is a mission critical part of the ATLAS experiment and is physically connected to the control network of the detector and the "data" network which allows connections to the CERN data centre through a switch at P1. The Sim@P1 project aims to opportunistically use the trigger and data acquisition high level trigger resources for offline computing. When working with Sim@P1 it is important to ensure the secure isolation from the physical resources at P1, seamless integration into the ATLAS distributed computing system, and reliable transition between the functions of the resources. Throughout this text we will refer to standard operation of the HLT as *online mode* and the operation as part of the ATLAS distributed computing system as *offline mode*.

*e-mail: berghaus@cern.ch
**e-mail: sobie@uvic.ca

Table 1. The hardware at P1 currently available for use with Sim@P1. The C6100 nodes are the decommissioned old HLT. They provide 11008 hyper-threaded (HT) cores permanently running in offline mode. Not all cores are used to ensure the virtual machines provide sufficient memory for ATLAS offline workloads. The other hardware is switched to offline mode when data taking is not foreseen in the next 24 hours. These opportunistic resources provide up to 97216 additional cores. Usually the trigger and data acquisition team retains some resources for their needs.

Product name	Intel® Xeon®	HT cores	Memory [GB]	Instance cores	Nodes
C6100	X5650	24	24	16	688
Centerprise	E5-2650 v4	48	64	48	360
Persy	E5-2660 v4	56	64	56	440
MegWare	E5-2680 v3	48	64	48	680
QuantaPlex	E5-2680 v3	48	64	48	472

A system satisfying these criteria was developed during the first long shutdown of the LHC [3]. Isolation is achieved by running virtual machines on the physical HLT hardware. The virtual machines were originally managed using the cloud framework OpenStack [4]. The virtual machines shared the "data" connection of the HLT hardware through a tagged VLAN, which provided network isolation on the level of the Ethernet frame managed by the switches. This VLAN allowed the Virtual Machines to connect to a controlled list of interfaces in the CERN general purpose network. This list specifies the interface of the machines needed to allow offline workloads to be delivered and executed. To minimise impact of Sim@P1 on the Trigger and Data Acquisition (TDAQ) operation, only simulation tasks from the central production system are submitted to run at P1.

The original implementation of Sim@P1 ran successfully during the first long shutdown of the LHC facilities between 2013 and 2015. Once the experiment resumed data-taking, the system was used in opportunistic mode [5]. The HLT was switched from TDAQ function to Sim@P1 mode for intervals of a few days during technical stops and machine development. To allow this opportunistic usage a set of scripts were developed to manage the transition of resources between TDAQ and Sim@P1 mode.

During the second long shutdown (LS2) of the LHC facilities, starting in 2019, the HLT was be upgraded. The changes to the HLT necessitated an upgrade of the Sim@P1 infrastructure[1]. A previous publication [6] offered multiple options for the upgrade of the Sim@P1 infrastructure during the LS2. In this paper, we describe how the system was modified for operation with the upgraded HLT hardware.

2 The Sim@P1 infrastructure

During the year end technical stop from December 2017 to March 2018, the computing hardware of the HLT was replaced with new nodes. The old HLT hardware was retained at P1 and is permanently operating in offline mode. The new hardware has been used offline during the various technical and machine development stops throughout 2018. The current hardware configuration is summarised in table 1. Groups of 32 or 40 servers are organized into racks in the data centre at P1.

When a rack is not needed for data taking a shifter can set that rack to offline operation. This action triggers a change in the configuration database used by the TDAQ. The next time the configuration management system runs on any server, or trigger processing unit (TPU), in that rack[2] it changes the system configuration to reflect the change in the configuration

[1] The OpenStack Icehouse does not support operating on CentOS7 running on the TDAQ HLT starting in 2019

[2] The configuration management tool, Puppet, runs once an hour on the TPUs.

database. That means an ephemeral disk providing 20 GB per core is created and a virtual machine instance is started[3]. This document will refer to such a running virtual machine as an *instance*.

Instances are contextualised using amiconfig[4] [7]. The contextualisation is delivered using an ISO image added to the instance by libvirt [8]. The ISO image is formatted as an OpenNebula data source. The contextualisation sets up the computing environment for the ATLAS offline workloads and sets the virtual machines to advertise themselves to a HTCondor [9] system running in the CERN general purpose network. Instances are configured to use HTCondor's dynamic partitioning feature to map workloads to the resources. Process isolation is achieved using the control groups feature of the Linux kernel.

The HTCondor system for Sim@P1 was rebuilt with a single central manager and four schedulers. The virtual machines are managed by CERN's configuration management system. Work is submitted to the schedulers using the PanDA Harvester [10]. Sim@P1 presents a single unified production queue to the PanDA system. The Harvester is operating the queue in push mode allowing the workloads to request the resources they need leveraging the dynamic partitioning of workers. The HTCondor system now directly notifies the PanDA workload management system when resources are added or removed from Sim@P1.

2.1 Content delivery

CernVM is a good solution for Sim@P1 because the micro-image can be distributed to all the TPUs and require a trivial disk space during HLT operation. Using CernVM means that we rely on the CernVM file system (CVMFS) to provide both the operating system content as well as the experiment software. In our CHEP2018 contribution we incorrectly stated that the load on the Frontier squids during the switch to offline mode was low [6]. This measurement was flawed in two ways: the instances were incorrectly contextualised to retrieve the CVMFS content from the Frontier squids operated by CERN IT for their batch system. Furthermore, the Frontier squids at P1 were blocked by the Frontier servers, causing Frontier requests to fail over to the CERN IT's squids Figure 1 shows that, after correcting the contextualisation,

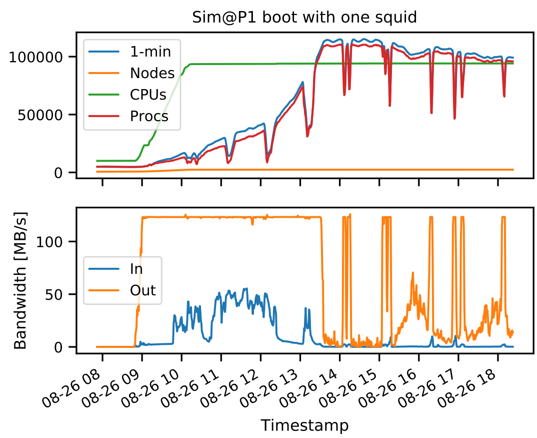

Figure 1. The top figure shows the number of instances, CPUs, and running processes as well as the aggregate one minute load on the TDAQ HLT farm as it transitions to Sim@P1 mode. Below, using the same timestamps, the network load on the single Frontier proxy delivering the CVMFS content to all the nodes is displayed. The correlation of saturated outgoing bandwidth in the lower plot with the steady increase in the load and processes in the upper plot suggests the squid is the rate determining system in the transition of the farm.

the Frontier squid at P1 was saturating its network bandwidth to deliver the content required by the booting instances. To address the bottleneck posed by the single squid we added a

[3]The size of the ephemeral disk is reduced to ensure at least 20% of the hard drive is free.

[4]Originally a project by rPath, Inc. now maintained by the CernVM team.

second squid instance, then added CVMFS caches that persist throughout HLT operation, and finally created a hierarchy of squids on the instances in offline mode.

The addition of a second squid doubled to total bandwidth used to deliver the content required for CVMFS. As a result the transition to Sim@P1 mode finished in 3 hours - approximately twice as fast as with the single Frontier squid.

2.2 Persistent CVMFS caches

Much of the CVMFS content remains unchanged between successive switches between offline and online mode: changes to the files in the CernVM system tend to be minimal and an ATLAS software release once downloaded does not change. So keeping the CVMFS cache throughout online operation reduces the amount of content the caches need to provide when transitioning to offline mode.

A 50 GB virtual disk image was created on each TPU to serve as persistent CVMFS cache. The ephemeral disk may be smaller to compensate for the size of the cache. Libvirt is configured to mount the cache as an additional drive. The contextualisation checks for a second drive. If it finds the second disk exists and is formatted with the label "cache" it mounts the disk as the CVMFS cache. If the second disk exists but carries no labelled partition the disk is formatted with the label "cache" and mounted.

Figure 2 shows that with a warm CVMFS cache the farm boots in the hour, with a slight

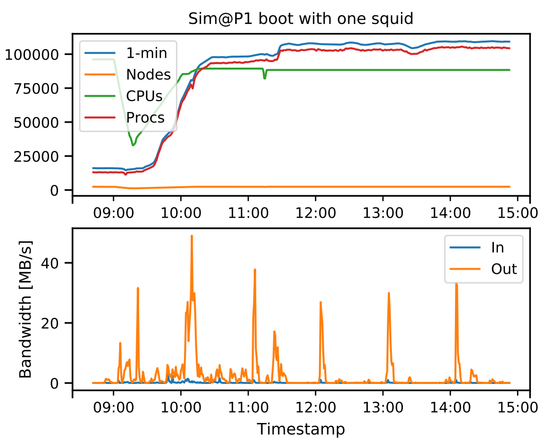

Figure 2. The top figure shows the number of instances, CPUs, and running processes as well as the aggregate one minute load on the TDAQ HLT farm as it transitions to offline mode. Below, using the same timestamps, the network load on the two Frontier squids delivering the CVMFS content to all the nodes is displayed. With the persistent cache the network traffic on the squids is greatly reduced.

time delay between the virtual machine being created and being fully occupied with work.

2.3 Squid hierarchy

The contextualisation of the instances was adjusted to run a squid in two instances in each rack, as a complimentary approach to the persistent CVMFS caches[5]. The squids treat each other as siblings and the Frontier squid caches at P1 as parents. A web proxy-autodiscovery service is set up to connect virtual machines to proxy caches on boot: those serving a squid connect to the central Frontier squids at P1, others are connected to the two squids in the same rack. The CernVM team added functionality to the CernVM microkernel to delay the boot process until at least one proxy cache is up.

[5]The first and fifth TPU to ensure the squids are in separate chassis.

3 Operational experience

The new configuration of Sim@P1 was operational within two days of the upgrade to the TDAQ HLT. This quick transition is a testament to the simplification of the Sim@P1 infrastructure under the new configuration. A feature of libvirt was discovered when returning the resources to HLT operation. The following section described the workaround that was implemented. The squid hierarchy provided a marginal improvement in the transition from online to offline operation while reducing the overall stability of the system: we started loosing racks when the two squids in the rack cease functioning.

3.1 Returning the resources

Returning the resource to online mode quickly and reliably is absolutely essential, moreso when ATLAS is taking data. Should an instance running on a TPU be busy with IO intensive work, it may take time O(10 s) for libvirt to destroy the instance[6]. Libvirt has a timeout of 15 s waiting on the destruction of a virtual machine, taking longer produces an error in returning that TPU to online mode. A helper script to manage the instance state was written. It attempts to destroy the instance three times, with a 15 s delay between attempts. With this modification, all resources were successfully returned to online mode without issues.

3.2 Other workflows

Data taken by ATLAS is cached at point 1 and transferred to the CERN data centre. Tests performed at the end of 2018 have shown that we must assume the network between point 1 and CERN to be busy transferring data from the cache to storage in the CERN data centre during technical stops. Offline operation must not interfere with the process. That means workflows requiring little data transfer across the network, such as simulation, are a natural fit for Sim@P1.

Event generation is a frequent task that requires no input and produces very little output. Previous experience with event generation found that depending on the software used and the physics signal generated these tasks have a long tail in the required memory. We created dedicated high memory and single core queues to explore the use of Sim@P1 for event generation. At the end of 2019, the first event generation tasks were successfully executed on Sim@P1. Since event generation in ATLAS are workloads using a single core, the four HTCondor schedulers were fully occupied managing event generation reducing the overall usage of Sim@P1.

3.3 Future improvements

The infrastructure supporting Sim@P1 could be further improved to reduce the work required to maintain and operate the resource.

The success in running event generation in late 2019 shows that this could be done in an automated setup. However workloads that greatly exceeding their memory allocation must be killed by HTCondor to ensure system stability. In addition, additional HTCondor schedulers must be commissioned to accommodate the increased number of jobs and limit the number of single core jobs allowed in the queue.

The hardware serving as Frontier squids operating at P1 needs to be replaced. Supporting greater bandwidth for the CVMFS and Frontier content distribution will further improve the

[6]Usually an active application using swap space.

rate at which the farm can be switched from HLT to Sim@P1 mode. New hardware would mean the squid hierarchy described in section 2.3 could be removed improving the stability of Sim@P1.

As longer term project, OpenStack Heat or Kubernetes could be used to build and auto-scaling pool of HTCondor schedulers. Adding volatile storage to serve as a cache inside P1 may allow more data hungry workflows to be executed without interfering with online operations[7].

4 Conclusions

The upgrade of the Sim@P1 hardware was swiftly and successfully accomplished at the beginning of 2019. The new configuration is simpler and more robust since it relies on low level Linux tools and libraries. The system has become much more responsive by adding persistent CVMFS caches. More improvements promise to make Sim@P1 more robust, versatile and easy to manage.

References

[1] The ATLAS Collaboration, JINST **3**, S08003 (2008)

[2] The ATLAS Collaboration, *Technical Design Report for the Phase-I Upgrade of the ATLAS TDAQ System* (CERN, Geneva, 2013) 120-122

[3] S. Ballestrero *et al*, J. Phys. Conf. Ser. **664**, 2 022008 (2015)

[4] Openstack project, "OpenStack" [software], version Icehouse, available from https://www.openstack.org/software/icehouse/ [accessed 2018-09-24]

[5] S. Ballestrero *et al*, J. Phys. Conf. Ser. **898**, 8 082012 (2017)

[6] F. Berghaus *et al.* [ATLAS Collaboration], EPJ Web Conf. **214**, 07021 (2019)

[7] CernVM project, "amiconfig" [software], available from https://github.com/cernvm/amiconfig [accessed 2018-10-10]

[8] libvirt project, "libvirt virtualization API" [software], version 0.10.2, available from https://libvirt.org [accessed 2018-10-10]

[9] D. Thain, T. Tannenbaum and M. Livny, Concurrency Computat.: Pract. Exper. 2005;17:323 (2005)

[10] T Maeno *et. al.* [ATLAS Collaboration], EPJ Web of Conf. **214**, 03030 (2019)

[11] J. Blomer *et. al.*, J. Phys. Conf. Ser. **513**, 032009 (2014)

[7] Such as Xcache of a volatile Disk Pool Manager