ATL-SOFT-PROC-2018-026
24 November 2018

ATLAS Distributed Computing: Its Central Services core

Christopher Jon Lee[1,*] and *Alessandro* Di Girolamo[2], *Johannes* Elmsheuser[3], *Alexey* Buzykaev[4], *Emil* Obreshkov[5], *Ivan* Glushkov[5], *Shaojun* Sun[6], on behalf of the ATLAS Collaboration

[1]University of Cape Town
[2]European Laboratory for Particle Physics (CERN)
[3]Brookhaven National Laboratory (BNL)
[4]Budker Institute of Nuclear Physics
[5]The University of Texas at Arlington
[6]University of Wisconsin

Abstract. The ATLAS Distributed Computing (ADC) Project is responsible for the off-line processing of data produced by the ATLAS experiment at the Large Hadron Collider (LHC) at CERN. It facilitates data and workload management for ATLAS computing on the Worldwide LHC Computing Grid (WLCG). ADC Central Services operations (CSOPS) is a vital part of ADC, responsible for the deployment and configuration of services needed by ATLAS computing and operation of those services on CERN IT infrastructure, providing knowledge of CERN IT services to ATLAS service managers and developers, and supporting them in case of issues. Currently this entails the management of 37 different OpenStack projects, with more than 5000 cores allocated for these virtual machines, as well as overseeing the distribution of 29 petabytes of storage space in EOS for ATLAS. As the LHC begins to get ready for the next long shut-down, which will bring in many new upgrades to allow for more data to be captured by the on-line systems, CSOPS must not only continue to support the existing services, but plan ahead for the expected increase in data, users, and services that will be required. This paper attempts to explain the current state of CSOPS as well as the strategies in place to maintain the service functionality in the long term.

1 Introduction

ATLAS[1] Distributed Computing (ADC[2]) is currently responsible for the management of more than 700 virtual machines spread across forty three projects that are hosted in the CERN IT Openstack[3] service with 5310 allocated to the various projects mostly used for computing on the Worldwide LHC Computing Grid (WLCG[4]) . A third of these are used by the three main Projects in ADC, Build, Panda[5], and Rucio[6]. The build service is responsible for ensuring that the ATLAS Software is correctly "built" and operational via

*e-mail: chris.lee@cern.ch

testing. PanDA, the Production and Distributed Analysis system is the workload management system that controls all the ATLAS jobs distributed on the WLCG. Finally, Rucio ensures that all the data required for these jobs running with ATLAS software is distributed to the correct place allowing scientists within ATLAS to sift through the petabytes of data acquired during beam time by the detector. The last two thirds of used cores are spread across multiple other projects that are needed for services from data preparation to detector safety systems.

The Central Service project (CSOPS) is central to these activities. Its role is to support and maintain the different applications and systems, act as the interface between CERN IT and the service managers of the various projects, whilst ensuring that security and good computing practises are maintained.

2 What was done

As time passes, things evolve and new tools are always being released, many of which bring in new problems of their own. There is a very fine line between just how helpful a tool can be, how much time is required to gain the expertise to use the tool, and how long implementation of these tools will take. Central Service coordinators decided to follow the design principle noted by Kelly Johnson of the U.S. Navy in 1960: KISS. KISS is an acronym for "Keep it simple, stupid". Having a system that is overly sophisticated and takes forever to repair and maintain, wastes precious time that could be spent on other tasks. With this philosophy in mind, CSOPS have tried to simplify as much as possible via automation, using currently available tools that are powerful enough to complete the task without too much effort, while still being cutting edge to not fall behind the latest technological advancements.

3 How was it done

More than half of a working day is spent on going through emails, reading questions from users, checking logs sent from failed systems, and so forth and the remainder of the day is usually spent debugging the logs and replying to emails. Not much can be done about this, though many of the repetitive tasks can be automated. The following tools have been put into place to try help make these tasks much easier.

- Puppet
- Continuous Integration in GitLab
- BASH SHELL
- Run Deck
- InfluxDB / Grafana

3.1 Puppet

Installing a new machine from scratch, and then manually configuring it, can take many hours. Not only is time wasted waiting for the initial installation, but then one has to manually check each package, configure the applications and then perform quality checks on the system. With Puppet [7], this entire procedure can now be done in under an hour. Integrated with the tools provided by CERN IT, this is done via a "one liner" on command line. It registers the machine in the CERN Network DB, spawns the VM in Openstack and then ensures that the machine matches the configuration from the Puppet manifest. Once it is up, Puppet then runs every hour ensuring that the same sane state is maintained, attempting to revert any deviation in the configuration. While the initial learning curve is steep, and coding the Puppet manifest can be difficult for new users, once done the machine can be replicated immediately by anyone who has access to the Puppet manifest. Summaries of each Puppet run are received daily, as well as immediate status reports of any failed manifest compilations or changes that have been applied. These reports are used not only as a debugging tool to let us know when something is broken, but also as a security feature, by being able to quickly notice if changes have been made to critical files which are not meant to be modified and investigate if there was any malicious intent.

3.2 Continuous Integration in GitLab

Versioning systems are nothing new. Currently all the code maintained by Central Services is held in gitlab.cern.ch. While this does help ensure that code can always be rolled back, somebody still has to go through the code to ensure that things are working. Also, as numerous people are often working on the same code and everyone has different "programming styles", things can get messy. With the Continuous Integration (CI) in GitLab, CSOPS are now able to check code as soon as it is committed to gitlab. This helps spot everything from syntax errors, bad coding practises, incorrect spacing in code and so forth. (Figure 1 on page 3)

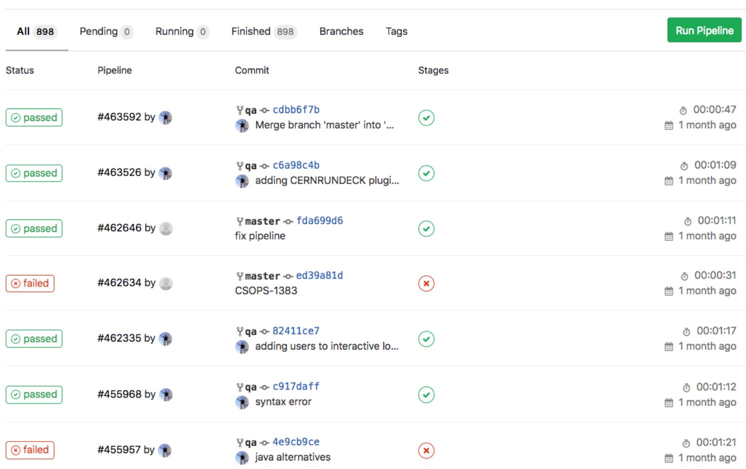

Figure 1. Pipelines of the continuous integration tests

3.3 BASH

BASH as a scripting language is a well established tool that Central Services is currently using for many of its scripts. Although there are other, more sophisticated programming languages available, adopting any particular one requires a large investment of time and money, especially if it is only used sporadically. Linux Systems Administrators are well used to BASH. Further more, since BASH is part of all of the Operating Systems (OS) currently supported at CERN, it is always available. There is never a need to install the latest JAVA, to change all your code because Python 2.7 is not available on the OS you are working on. It just works, every time, everywhere.

It should be noted that this is only for CSOPS' own scripts used to perform administrative tasks on systems. As soon as a script becomes overly complicated by performing various functions, then higher level programming languages are used.

3.4 Rundeck

Once machines are installed, they still need to be updated, patched and maintained as software changes and as security issues are discovered. Many Linux Administrators are very proud of the fact that their systems have been up, and online for years without any disruption. This does however leave these operating systems open to security issues. Unavoidably, these systems need to be restarted to properly update the running kernels. Finding a time to install these updates that minimises disruption for users and operations is not straightforward. To address this, CSOPS have installed a RunDeck[8] server (Figure 2 on page 4) that is mostly used to help in rebooting nodes. For example to reboot all of the nodes in the Rucio project, one would have to stop services, wait for them to be removed from load balancers, update, restart, wait for them to come online, restart the services and then move to the next machine. This is a waste of valuable time. Using RunDeck, this can now be done with the click of a single button; the operation takes a few minutes to perform without any noticeable downtime for the users.

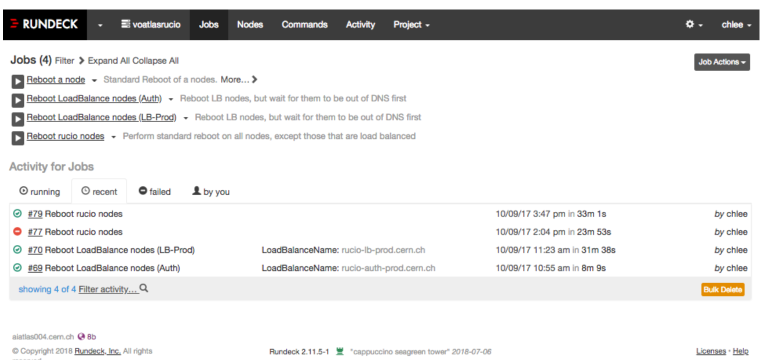

Figure 2. RunDeck server showing successes and failures of rebootes

3.5 InfluxDB / Grafana

We have petabytes of data that must be managed. CSOPS oversees the administration of thirty petabytes of data, shared by more than 1600 users, the "tier zero" scratch space for jobs and the storage of 89 different group areas in EOS. While the allocation of this space is the responsibility of the Computing Resource Management (CREM) group, CSOPS needs to be sure that things are monitored and configured as expected. Using the Database on Demand (DBOD) Service and OpenShift provided by CERN, CSOPS are now injecting the data from EOS command line operations, using BASH scripts and a curl tool, into an InfluxDB[9], and then using Grafana[10] to create plots of the usage (figure 3 on page 5), which provides an instantaneous overview. The next step is the implementation of alerting on these areas in Grafana to warn of issues before these areas have been filled with data. This will allow us to proactively fix and make the required changes to the storage quota nodes and ensure continuous operations by users. Grafana is now also being used to monitor host metrics

(Figure 4 on page 5)with data provided by the "collectd" service running on all nodes, and information from the Openstack services on project usage.

Figure 3. Grafana dashboard showing used files, disk space and rate of change

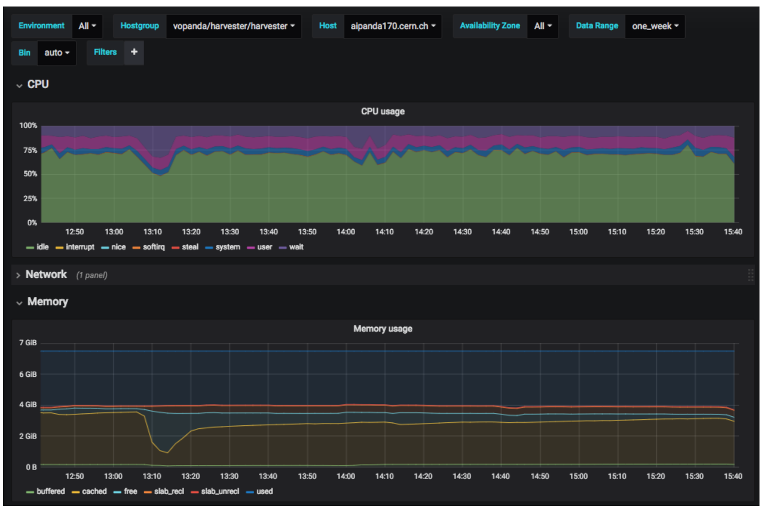

Figure 4. Grafana dashboard showing system metrics

4 Conclusion

Although CSOPS will benefit from further development work, insisting on the minimum level of complexity to deliver the required service has led to a robust and easily maintainable environment that is not only capable of handling the current working systems, but should also be functional for run 3 of the LHC.

References

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

[2] *Atlas distributed computing*, https://twiki.cern.ch/twiki/bin/viewauth/AtlasComputing/AtlasDistributedComputing, accessed: 2018-09-21

[3] *Openstack*, http://opensource.com/resources/what-is-openstack, accessed: 2018-09-21

[4] *Worldwide lhc computing grid*, http://wlcg-public.web.cern.ch/about, accessed: 2018-09-21
[5] Maeno T. et al. on behalf of the ATLAS Collaboration, Tech. Rep. ATL-SOFT-PROC-2017-016. 5, CERN, Geneva (2017), http://cds.cern.ch/record/2242034
[6] Barisits M. et al. on behalf of the ATLAS Collaboration, Tech. Rep. ATL-SOFT-PROC-2017-064, CERN, Geneva (2017), http://cds.cern.ch/record/2291104
[7] *Puppetlabs*, https://puppet.com/products/how-puppet-works/, accessed: 2018-09-21
[8] *Rundeck*, https://www.rundeck.com/what-is-rundeck, accessed: 2018-09-21
[9] *Influxdb*, https://www.influxdata.com/time-series-platform/influxdb/, accessed: 2018-09-21
[10] *Grafana*, https://grafana.com/, accessed: 2018-09-21