

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

¹University of Cape Town

²European Laboratory for Particle Physics (CERN)

³Brookhaven National Laboratory (BNL)

⁴Budker Institute of Nuclear Physics

⁵The University of Texas at Arlington

⁶University 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 43 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 43 projects that are hosted in the CERN IT Openstack[3] service with 5310 cores allocated to 930 virtual machines. Most are in use by 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 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

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

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 Services 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 Design Philosophy

As time passes, things evolve and new tools are always being released in industry, 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. CSOPS 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 our daily workload as possible via automation, all while using tools that are powerful enough to complete the tasks required by the team members, while still being cutting edge to not fall behind the latest technological advancements.

3 How was it done

In general, more than half of a working day is spent by the CSOPS team 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 are now in use to facilitate operational tasks.

- 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. Besides the time wasted on manual installations, each package has to be checked manually, applications configured and quality checks performed on the system. With Puppet [7], this entire procedure can now be done in less than 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 it has been completed, 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. Currently all of the 930 VM's controlled by CSOPS running on Openstack are puppet managed, with manifests constantly being updated and changed as the services/applications on these VM's evolve. This task has been performed with the help of experts in the team who had acquired puppet expertise working in other experiments. See the ATDS 2012 CHEP paper for reference. [8]

3.2 Continuous Integration in GitLab

Versioning systems are nothing new. Currently all the code maintained by CSOPS 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 guarantee 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)

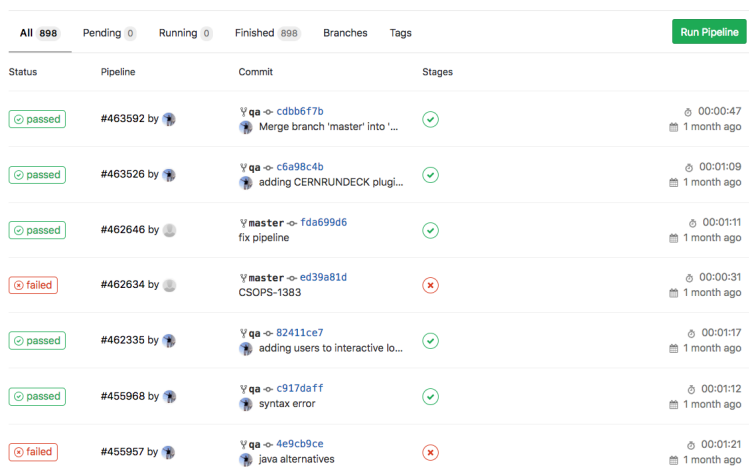


Figure 1. Pipelines of the continuous integration tests

3.3 BASH

BASH as a scripting language is a well established tool that the CSOPS team 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/or 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. It is difficult to schedule installations so that disruption for users and operations would be minimal. To address this, CSOPS have installed a RunDeck[9] 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 time-consuming process and a waste of resources. 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.

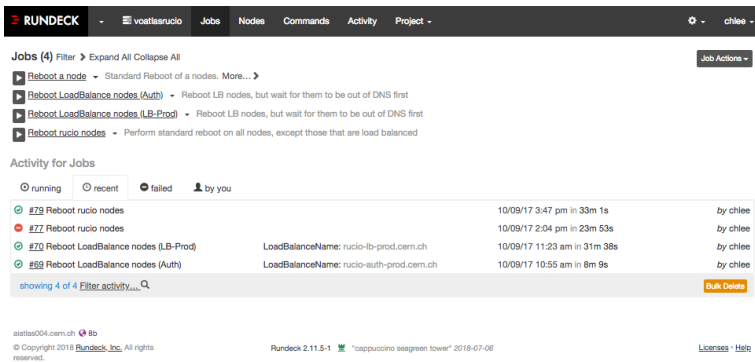


Figure 2. RunDeck server showing successes and failures of rebootes

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[10]. 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. To make things easier, CSOPS are using the "Database on Demand" (DBOD) and OpenShift[11] services provided directly by CERN IT, and are injecting the data from EOS command line operations, using BASH scripts and a curl tool, into an InfluxDB[12]. Grafana is used as a visualisation frontend providing necessary plots to overview space usage (figure 3 on page 5), which provides an instantaneous overview. The next step is implementation of alerts in Grafana in order to notify operators when particular storage areas are getting full. This will allow us to proactively fix and make the required changes to the storage quota nodes and ensure continuous operations by users.



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

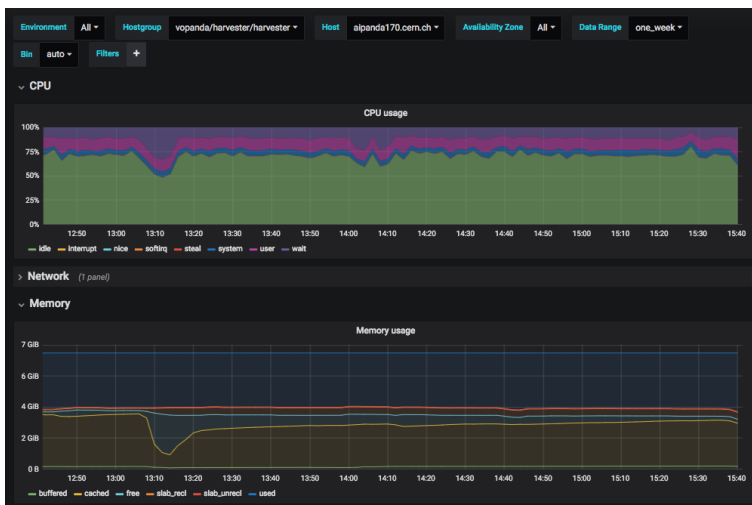


Figure 4. Grafana dashboard showing system metrics

Grafana is now also being used to monitor host metrics (figure 4 on page 5) provided by the "collectd" service running on all nodes, and information from the Openstack services on project usage.

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. Tools and applications will of course change, and many of them mentioned here in this paper may even be obsolete by the time this goes to print. However, we believe that by sticking to the design philosophy of keeping it simple, CSOPS have been able to create a robust and manageable system, which is expected to be able to continue well into RUN 3 of the LHC for our services.

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] T. Maeno 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] M. Barisits 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] S. Ballestrero, et al. , *Upgrade and integration of the configuration and monitoring tools for the ATLAS Online farm* (J.Phys.Conf.Ser. 2012), Vol. 396, p. 042005
- [9] *Rundeck*, <https://www.rundeck.com/what-is-rundeck>, accessed: 2018-09-21
- [10] *Eos*,
<http://information-technology.web.cern.ch/services/eos-service>,
accessed: 2019-02-12
- [11] *Openshift*, <https://www.openshift.com/learn/what-is-openshift/>,
accessed: 2019-02-12
- [12] *Influxdb*, <https://www.influxdata.com/time-series-platform/influxdb/>,
accessed: 2018-09-21