Accelerating science: the usage of commercial clouds in ATLAS Distributed Computing

Fernando Barreiro Megino[1,*], *Mikhail* Borodin[2], *Kaushik* De[1], *Johannes* Elmsheuser[3], *Alessandro* Di Girolamo[4], *Nikolai* Hartmann[5], *Lukas* Heinrich[6], *Alexei* Klimentov[3], *Mario* Lassnig[4], *FaHui* Lin[1], *Tadashi* Maeno[3], *Zachary* Marshall[7], *Gonzalo* Merino[8], *Paul* Nilsson[3], *Jay* Sandesara[9], *Cedric* Serfon[3], *David* South[10], *Harinder* Singh[11] on behalf of the ATLAS Computing Activity

[1]University of Texas at Arlington, Arlington, TX, USA
[2]University of Iowa, Iowa City, IA, USA
[3]Brookhaven National Laboratory, Upton, NY, USA
[4]CERN, Geneva, Switzerland
[5]Ludwig-Maximilians-Universitaet Muenchen, Munich, Germany
[6]Max-Planck-Institut für Physik, Munich, Germany
[7]Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[8]Port d'Informació Científica, Barcelona, Spain
[9]University of Massachusetts at Amherst, Amherst, MA, USA
[10]DESY, Hamburg, Germany
[11]California State University at Fresno, Fresno, CA, USA

Abstract. The ATLAS experiment at CERN is one of the largest scientific machines built to date and will have ever growing computing needs as the Large Hadron Collider collects an increasingly larger volume of data over the next 20 years. ATLAS is conducting R&D projects on Amazon Web Services and Google Cloud as complementary resources for distributed computing, focusing on some of the key features of commercial clouds: lightweight operation, elasticity and availability of multiple chip architectures.

The proof of concept phases have concluded with the cloud-native, vendor-agnostic integration with the experiment's data and workload management frameworks. Google Cloud has been used to evaluate elastic batch computing, ramping up ephemeral clusters of up to O(100k) cores to process tasks requiring quick turnaround. Amazon Web Services has been exploited for the successful physics validation of the Athena simulation software on ARM processors.

We have also set up an interactive facility for physics analysis allowing end-users to spin up private, on-demand clusters for parallel computing with up to 4 000 cores, or run GPU enabled notebooks and jobs for machine learning applications.

The success of the proof of concept phases has led to the extension of the Google Cloud project, where ATLAS will study the total cost of ownership of a production cloud site during 15 months with 10k cores on average, fully integrated with distributed grid computing resources and continue the R&D projects.

ATL-SOFT-PROC-2023-020
04 September 2023

*e-mail: fernando.barreiro[at]uta.edu
Copyright 2023 CERN for the benefit of the ATLAS Collaboration. CC-BY-4.0 license

1 ATLAS Cloud Projects: Leveraging Key Partnerships

The ATLAS experiment [1], conceived several decades ago, was accompanied by the development of its computing model to meet its unique needs. At that time, the IT landscape was inadequate for its requirements. To overcome this challenge, ATLAS and the other LHC experiments embarked on their own path and developed the Worldwide LHC Computing Grid (WLCG) [2].

Since then, technology and the IT industry have rapidly advanced, leading to significant changes in the landscape. Cloud computing has reached a level of maturity where it offers substantial advantages. Today, some of the main motivations for investing in projects with commercial cloud computing providers are:

1. Knowledge and technology transfer: Learning from the industry's solutions to large-scale computing problems.
2. Finding additional sources of computing power that can be utilized on a stable or elastic basis.
3. Gaining access to architectures that are not readily available on-premise.
4. Minimizing efforts associated with infrastructure and maintenance.

Driven by these motivations, the ATLAS experiment has actively pursued various research and development initiatives in cloud computing in the last ten years. In recent years, ATLAS has conducted longer-term cloud computing initiatives, significantly improving the integration of compute and storage [3]. This has been achieved through the adoption of a cloud native approach that is not restricted to a particular cloud provider. The ability to run fully fledged ATLAS sites in the cloud has expanded the scope of the cloud activities, which will be the central focus of this article.

This article delves into two main projects undertaken in recent years, highlighting the gained experience. The first project involved a fruitful collaboration with Amazon Web Services (AWS), spanning from July 2020 to May 2023. The credits for this project were acquired through California State University at Fresno.

The second project formed a stable relationship with Google Cloud Platform (GCP), commencing around 2018. Engagements with the Google Cloud team have played a pivotal role in shaping our cloud integration strategy. The most recent funding round with Google Cloud covers the 15-month period from July 2022 to October 2023. During this time, ATLAS entered into a "User Subscription Agreement for the US Public Sector", allowing for a fixed cost and flexible resource consumption model. The primary objective of this collaboration with Google Cloud is to evaluate the feasibility and gain experience in running a fully-fledged ATLAS site on the cloud. It is our aim to execute all ATLAS workloads, ideally replicating a similar mix as found on other sites. Towards the end of the project, a whitepaper detailing the Total Cost of Ownership is expected to be released. By leveraging these partnerships with industry-leading cloud providers, ATLAS has successfully developed a comprehensive, cloud-native integration model and implemented it at scale.

2 Cloud native integration

ATLAS Distributed Computing comprises two primary components responsible for the centralized orchestration of data and jobs. The high-level integration model can be seen in Figure 1 and will be described in the following subsections.

2.1 Storage integration through Rucio

Rucio [4] serves as the distributed data management system within ATLAS. Sites participating in the project are required to provide Rucio Storage Elements, which store the

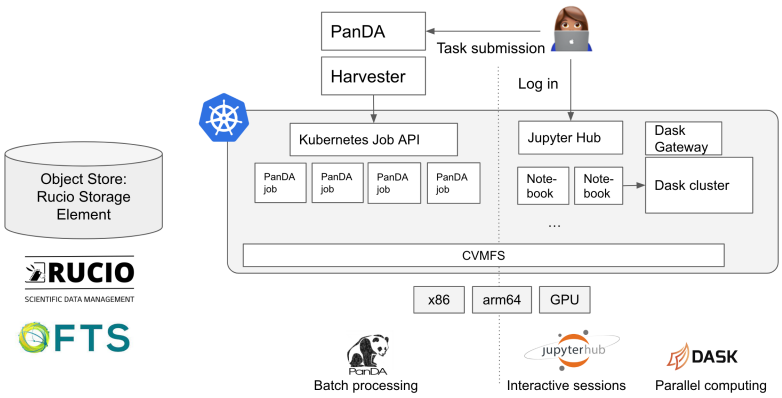

Figure 1: Cloud integration elements

data local to each site and are utilized for job input/output operations. Storage and data management have seen significant advancements in the realm of cloud computing. A pivotal development has been the adoption of standard protocols, specifically HTTP, by Rucio and the WLCG middleware.

In the cloud environment, the Rucio Storage Element is implemented as an Object Store bucket [5]. Cloud storage platforms typically support the concept of Signed URLs, which enable authorized read, write, and delete operations on specific files for a predetermined period. This is achieved by downloading a signing key that can generate signed HTTP URLs. By incorporating the signing key into the Rucio and File Transfer Service (FTS [7]) servers, essential functionalities such as download, upload, deletion, and third-party copy can be implemented. These actions cater to the typical data access requirements within the ATLAS project.

2.2 Compute integration through PanDA

PanDA [6] is the distributed workload management system used in ATLAS. Sites are expected to provide a PanDA queue configured on top of their local batch system.

In the cloud context, compute integration within the ATLAS project relies on Kubernetes [8] as a shared foundation for executing both batch and interactive workloads. Leading cloud providers offer on-demand Kubernetes clusters that can be easily set up using a few clicks or command line interactions.

The resource-facing component of PanDA, known as Harvester [9], directly interfaces with Kubernetes clusters [10] and utilizes its native job controller for submitting batch jobs. To resemble a grid worker node, the Kubernetes cluster is equipped with CVMFS [11], a read-only file system that contains ATLAS physics and middleware software. This integration allows a pod in Kubernetes to function as a virtual worker node within the grid infrastructure.

During the evaluation of workload management systems for the Vera Rubin experiment, PanDA's cloud readiness played a significant role in providing a competitive advantage. The ability of PanDA to seamlessly integrate with cloud infrastructure, specifically the Google Cloud interim data facility used by Vera Rubin, allowed for immediate utilization of the resources.

Additionally, as a side project, novel interactive physics analysis techniques were explored. By leveraging commonly available Helm charts [12], JupyterHub [13], Dask [14]

and Dask Gateway [15] can be installed on Kubernetes. This infrastructure enables interactive, distributed, pythonic physics analysis and facilitates the utilization of machine learning applications. It is important to note that this setup was primarily conducted for research and development purposes and is not a standard component of the conventional ATLAS site infrastructure.

3 Use cases and experience

3.1 ATLAS-Google Cloud site

The primary objective of the ongoing ATLAS-Google Cloud project is to operate an ATLAS site, consisting of a Rucio Storage Element and a conventional PanDA queue utilizing *x86* CPUs. ATLAS grid sites can vary significantly in size, with pledged grid sites ranging from several hundred vCPUs to as large as 40 000 vCPUs. Notably, there are exceptional cases like the ATLAS trigger farm [16], which boasts 100 000 vCPUs, and the Vega SuperComputer [17] in Slovenia, which often contributes a few hundred thousand vCPUs.

The ATLAS-Google Cloud site operates with a capacity of either 5 000 or 10 000 vCPUs (see Figure 2), depending on the allocated budget. The PanDA queue configuration does not impose any restrictions and is open to executing any ATLAS grid payloads. However, as part of the Total Cost of Ownership (TCO) studies, specific job types have occasionally been limited within the queue to investigate potential cost differences associated with varying I/O profiles.

Since the start of operations, the ATLAS-Google Cloud site has experienced a 5% failure rate in terms of wallclock time which amounts to around 240 000 failed jobs. This aligns with the expected performance of ATLAS pledged resources. Several aspects should be taken into consideration:

1. Increased failure rate during the initial weeks of the project: Configuring the infrastructure optimally posed challenges during the early stages.

2. Spot instance-related failures: The queue has been utilizing Spot instances. Spot instances [18] refer to virtual machine instances that are available at significantly reduced prices, but can be preempted by the cloud provider at any point in time. Typically, spot preemptions account for a fraction of job failures, but the frequency can increase during periods of high cloud usage. Approximately 40 000 out of 240 000 failed jobs were attributed to spot preemptions (non-exhaustive study, some cases could be missed).

3. Failure rate during Google Kubernetes Engine (GKE) cluster auto-upgrades: Failures have been observed during the periodic auto-upgrades of the GKE cluster, where updates roll through all nodes and result in the termination of running jobs. We relate approximately 10 000 out of 240 000 failed jobs to these upgrade events (again non-exhaustive study). Optimization of the auto-upgrade configuration has not been attempted.

During the operation of our cloud site, an important observation has been the critical reliance on high network usage and data replication for the experiment. Given the scale of our site, substantial egress (data transfers out of the cloud) reaching several PB per month is necessary. However, it's worth noting that cloud egress is metered and can significantly impact the monthly bill. As we approach the project's completion, one of our key priorities is to explore potential solutions to address this challenge. This may involve minimizing the volume of egress or establishing an interconnect between Google Cloud and a research provider to benefit from lower network fees.

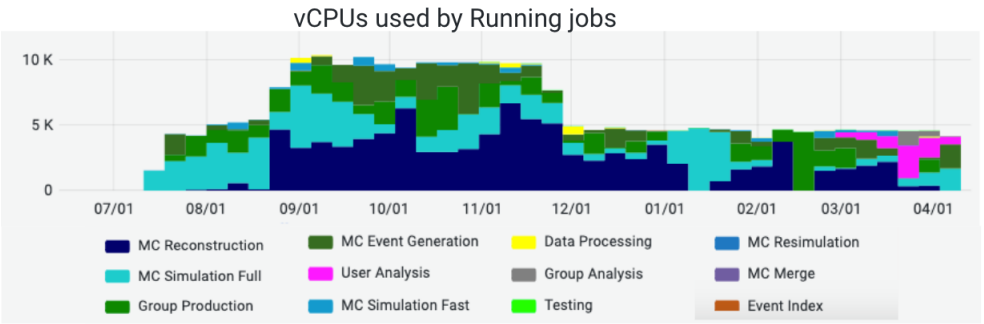

Figure 2: Size of the PanDA queue at Google Cloud and types of jobs executed

3.2 Elastic usage of cloud resources

One significant advantage of cloud computing is its elastic nature, allowing for the flexible allocation of resources based on specific needs and timeframes. This stands in contrast to the traditional ATLAS grid, where work is distributed throughout the year to avoid leaving resources unused.

In the context of the Active Learning [19] use case, which involves the iterative generation of Monte Carlo samples, it is essential to optimize the speed of each iteration, ideally within a day. By executing Monte Carlo generation chains of varying sizes, we were able to scale up a cluster to accommodate approximately 100 000 slots (see Figure 3). This achievement is noteworthy, considering that all resources operate within a single Kubernetes cluster, a single Google Cloud availability zone, and can be managed by a single engineer. During the ramp-up period, our resource capacity made us the second-largest contributor to ATLAS, surpassed only by the contribution from the Vega Supercomputer.

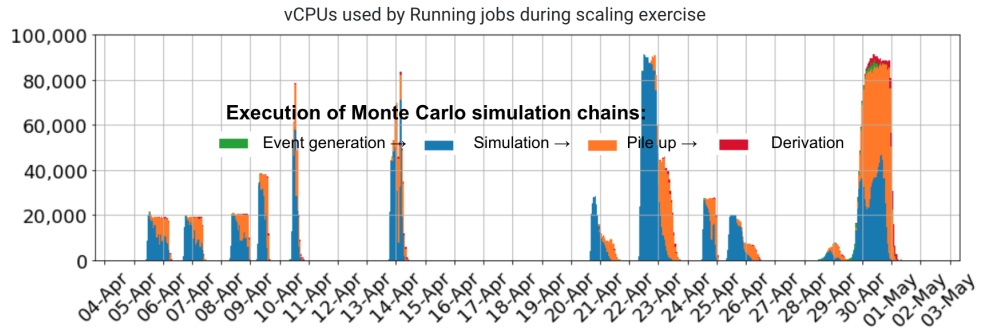

Figure 3: Size of the PanDA queue at Google Cloud during scaling exercise

3.3 Heterogeneous architectures

Traditionally, ATLAS computing has revolved around workloads that are well-suited for parallel execution on *x86* cores. Our computing infrastructure comprises hundreds of thousands of *x86* cores distributed worldwide. However, the outside world has been rapidly

advancing in machine learning applications, and certain countries have been heavily investing in SuperComputers and High-Performance Computing (HPC) systems. These systems are increasingly leveraging GPU architectures to maximize computational power within specified power constraints. Furthermore, the CPU component of SuperComputers is becoming more heterogeneous, featuring processors such as ARM or PowerPC.

3.3.1 GPU

For the majority of our current computing requirements, which consist of traditional applications, there is no inherent need for GPU accelerators. Consequently, ATLAS sites typically do not procure GPU cards specifically for ATLAS work. However, some sites possess GPUs for research and development purposes or to cater to other experiments that actively utilize GPUs. Presently, there are approximately 10 sites out of around 150 in the ATLAS grid that offer GPUs, although the specific number of GPUs associated with each queue remains uncertain to the authors.

ATLAS users who wish to utilize GPUs often face challenges in finding resources at the scale they require. Similarly, it can be difficult to identify users with GPU-based applications who are interested in participating in a project.

In collaboration with one user working on a new implementation of simulation-based inference, we have partnered to address this issue. The user's method relies on large ensembles of deep neural networks to approximate the exact likelihood, with additional neural networks employed to model systematic uncertainties in the measurements. Practically, this analysis necessitates O(100) GPUs to achieve an acceptable turnaround time for the user.

To accommodate this, we have established a PanDA queue with up to 200 NVIDIA T4 GPUs. The GPU nodes are provisioned on-demand, ensuring that no infrastructure costs are incurred when no payloads are available. This PanDA queue currently represents the most active GPU usage for ATLAS and has enabled analyses with a level of precision that would have otherwise been unattainable without engaging in an exceedingly costly infrastructure setup.

3.3.2 ARM

ARM processors have gained attention due to their increased power efficiency, offering significant energy savings compared to traditional processors. This has led to the adoption of ARM architecture in some SuperComputers. Additionally, some ATLAS grid sites have expressed interest in utilizing ARM processors to reduce their electricity bills and environmental impact. However, these sites are cautious and prefer to wait until the ATLAS software has been built and validated for ARM before making any significant purchases.

In collaboration with the ATLAS software and ATLAS middleware installation teams [20], we have successfully deployed the software baseline on CVMFS. This allowed us to set up an *arm64* PanDA queue, which we scaled up to accommodate 2,500 vCPUs. With this queue, we were able to execute the first-ever ATLAS *arm64* tasks. Furthermore, the physics validation for Athena [21] Simulation and Reconstruction, two main components of the ATLAS software, was successfully completed in September 2022 and March 2023, respectively. Although this validation represents only a portion of the ATLAS Software, it serves as an important milestone towards making ATLAS payloads compatible with ARM processors. Moreover, this achievement encourages sites to consider adopting ARM processors, as they can be confident in the readiness and viability of the ATLAS Software on this architecture.

3.4 Interactive analysis with Jupyter and Dask

We have successfully deployed Jupyter and Dask on Google Cloud, providing users with an interactive analysis environment. Notebooks are pre-configured with Dask plugins, and users can easily create private clusters through Dask Gateway using a few lines of Python code.

We integrated the JupyterHub instance with ATLAS IAM, allowing any ATLAS user to connect to the notebook server without the need for manual user management. We also created two images that users can request when starting a notebook. The PHYSLITE and columnar analysis environment image is commonly used with Dask for parallel processing in Python, while the Machine Learning environment image contains popular ML software like TensorFlow and its dependencies and is often used with GPU nodes.

Additionally, we optimized the setup to be cost-effective on Google Compute Engine. Critical components are assigned to on-demand nodes that are not subject to preemption, ensuring their stability. Dask Workers, on the other hand, are assigned to an autoscaled pool of cheaper Spot nodes to reduce costs since the work can be repeated without impacting the overall task. We also introduced the option to run notebooks on special nodes with dedicated GPUs or 1.4 TB of memory.

The cloud environment is well-suited for this infrastructure due to its dynamic nature, allowing users to evaluate Dask at various scales. Tests conducted by our users involved clusters with up to 4 000 workers, which demonstrated that the duration of tasks decreased proportionally to the number of workers. From a cost perspective, the overall cost remained roughly the same, with some overhead when running a larger cluster.

The analysis detailed in section 3.3.1, was developed and tested on an ML notebook with a GPU. It was later submitted in batch to hundreds of GPUs in parallel. Additionally, for this specific use case, we provided the option to boot notebooks with 1.4 TB of memory for a fitting step that required a significant amount of memory. This completed the self-contained environment for users within our Google Cloud infrastructure.

4 Conclusions

ATLAS Distributed Computing has made significant strides in the cloud space, transitioning from small-scale tests to large-scale, production environments. Along the way, we have encountered challenges but have overcome them with innovative solutions. These achievements have not only benefited the ATLAS cloud environment but also the entire ATLAS community, providing additional computing resources for advanced research and facilitating knowledge and technology transfer. Furthermore, many of the cloud technologies and practices we have adopted are being implemented on-premises as well.

Our User Subscription Agreement with Google Cloud has allowed us to explore all aspects of the cloud and gain valuable experience with cloud costs. Unlike in the grid environment, where certain costs are hidden, the cloud environment provides detailed metering of storage, compute, and network usage, including notable costs such as egress charges. To mitigate these egress costs, one of our priorities for the remainder of the ATLAS-Google Cloud project is to evaluate the possibility of establishing an interconnect between our cloud infrastructure and ESnet [22], which will link us to all WLCG sites through LHCONE. This interconnect would help reduce, though not eliminate, the egress cost.

Additionally, we are in the process of completing a Total Cost of Ownership study led by independent collaborators. The conclusions of this study will guide the future direction of our cloud projects and may provide insights into which activities are more cost-effective. Overall, the ATLAS project's journey in the cloud has been marked by challenges, achievements and the pursuit of cost optimization. By leveraging the cloud's capabilities and addressing its

associated costs, we are driving innovation and advancing the field of high-energy physics computing.

5 Acknowledgements

References

[1] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider,* JINST **3** S08003 (2008).

[2] Worldwide LHC Computing Grid, URL, http://cern.ch/lcg [accessed 2023-06-07]

[3] J. Elmsheuser *et al.*, *Seamless integration of commercial clouds with ATLAS Distributed Computing*, https://doi.org/10.1051/epjconf/202125102005

[4] M. Barisits *et al.*, *Rucio - Scientific data management*, Comput. Softw. Big Sci. **3** (2019) no.1, 11

[5] M. Lassnig *et al.*, *Extending Rucio with modern cloud storage support: Experiences from ATLAS, SKA and ESCAPE,* Proc. CHEP Conf. (2023) - in these proceedings

[6] T. Maeno *et al.*, *PanDA for ATLAS distributed computing in the next decade*, J. Phys. Conf. Ser. **898** (2017) no.5, 052002

[7] E. Karavakis *et al.*, *FTS improvements for LHC Run-3 and beyond*, EPJ Web Conf. **245** 04016 (2020)

[8] Kubernetes, URL https://kubernetes.io/docs/home/ [accessed 2023-06-07]

[9] T. Maeno *et al.*, *Harvester : an edge service harvesting heterogeneous resources for ATLAS*, EPJ Web Conf. **214** (2019), 03030

[10] F. H. Barreiro Megino *et al.*, *Using Kubernetes as an ATLAS computing site*, EPJ Web Conf. **245** (2020), 07025

[11] J. Blomer *et al.*, *The CernVM File System*, https://doi.org/10.5281/zenodo.4114078

[12] Helm, URL https://helm.sh/ [accessed 2023-06-07]

[13] T. Kluyver *et al.*, *Jupyter Notebooks – a publishing format for reproducible computational workflows*, http://dx.doi.org/10.3233/978-1-61499-649-1-87

[14] M. Rocklin *et al.*, *Dask: Parallel Computation with Blocked algorithms and Task Scheduling*, http://dx.doi.org/10.25080/Majora-7b98e3ed-013

[15] Dask Gateway, URL https://gateway.dask.org/ [accessed 2023-06-07]

[16] F. Berghaus *et al.*, *ATLAS SimP1 upgrades during long shutdown two,* EPJ Web Conf. **245** (2020), 07025

[17] Harnessing a supercomputer for ATLAS, URL https://atlas.cern/update/briefing/vega-supercomputer [accessed 2023-06-07]

[18] Google Cloud Platform Spot VMs, URL, https://cloud.google.com/spot-vms [accessed 2023-06-07]

[19] M. Dunford *et al.*, *Active Learning reinterpretation of an ATLAS Dark Matter search constraining a model of a dark Higgs boson decaying to two b-quarks,* ATL-PHYS-PUB-2022-045

[20] J. Elmsheuser *et al.*, *The ATLAS experiment software on ARM,* Proc. CHEP Conf. (2023) - in these proceedings

[21] P. Calafiura *et al.*, *The athena control framework in production, new developments and lessons learned,* Proc. CHEP Conf. (2005)

[22] Energy Sciences Network, URL, https://www.es.net/ [accessed 2023-06-07]