Using the ATLAS experiment software on heterogeneous resources

Johannes Elmsheuser[1,*] on behalf of the ATLAS Computing Activity

[1]Brookhaven National Laboratory, Upton, NY, USA

Abstract. With the large dataset expected from 2030 onwards by the HL-LHC at CERN, the ATLAS experiment is reaching the limits of the current data processing model in terms of traditional CPU resources based on x86_64 architectures and an extensive program for software upgrades towards the HL-LHC has been set up. The ARM CPU architecture is becoming a competitive and energy efficient alternative. Accelerators like GPUs are available in any recent HPC. In the past years ATLAS has successfully ported its full data processing and simulation software framework Athena to ARM and has invested significant effort in porting parts of the reconstruction and simulation algorithms to GPUs. We report on the successful usage of the ATLAS experiment offline and online software framework Athena on ARM and GPUs through the PanDA workflow management system at various WLCG sites. Furthermore we report on performance optimizations of the builds for ARM CPUs and the GPU integration efforts. We will discuss performance comparisons of different ARM and x86_64 architectures on WLCG resources and Cloud compute providers like GCP and AWS using ATLAS productions workflows as used in the HepScore23 benchmark suite.

ATL-SOFT-PROC-2025-004
07 January 2025

1 Introduction

The data processing and simulation software of the ATLAS experiment [1] at the LHC at CERN currently uses traditional CPU resources based on x86_64 architectures. With an increased dataset obtained during Run 3 and the even larger expected increase of the dataset by more than one order of magnitude for the HL-LHC from 2030 onwards, the ATLAS experiment is reaching the limits of the current data processing and especially simulation model in terms of these CPU resources [2]. An extensive program for software upgrades towards the HL-LHC has been set up to close the gap between the extrapolations of flat computing budgets and the projected resource needs. The CPU resource projections so far include only the use of classic x86_64 CPUs and no assumptions are made about the usage of accelerators like GPUs or e.g. ARM CPUs.

These additional compute resources and architectures are briefly discussed in the following sections: the ARM CPU architecture [3] is becoming a competitive and energy efficient alternative. We report on the usage of ARM CPUs in ATLAS production, highlight the software development on Apple silicon ARM CPUs and its usage on Cloud computing resources.

*e-mail: johannes.elmsheuser@cern.ch

© 2025 CERN for the benefit of the ATLAS Collaboration. CC-BY-4.0 license.

Parts of the offline software Athena [4] is currently being ported to compute accelerators like GPUs. Enabling the first usage of GPUs within the ATLAS offline software on WLCG resources [5] will be described subsequently. A comparison of ARM CPUs with x86_64 CPUs using the HEPscore benchmark suite [6] will be discussed in the last section.

2 The ATLAS grid setup: PanDA and Rucio

The ATLAS experiment uses the workflow management system PanDA [7] and the data management system Rucio [8] for world-wide distributed processing on heterogeneous resources. Figure 1 shows a schematic overview how both systems act on the different Grid, HPC and Cloud resources [9]. Dedicated ARM and GPU PanDA job queues are configured in the

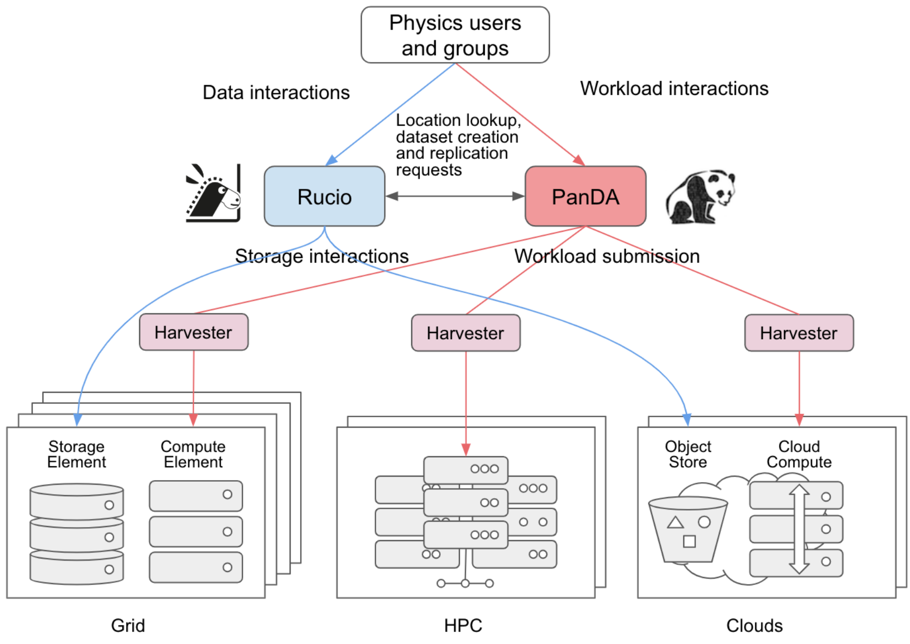

Figure 1. Simplified, high-level overview of ATLAS Distributed Computing. Rucio and PanDA serve as the central entry point for ATLAS users and hide the complexity of the underlying distributed resources.

Computing Resource Information Catalogue (CRIC) [10]. The PanDA jobs are either submitted to dedicated computing elements that host these extra resources or the PanDA pilot job configuration uses *"WantARM=True"* or *"+RequireGPUs = True"* and *"+RequestGPUs = 1"*, respectively. ATLAS analysis users can then target their PanDA jobs with the PanDA job submission tools using the extra command line options *'--architecture "&nvidia-*"'* or *'--architecture "@el9#aarch64"'* or the usage will be automatically detected by these tools if the user analysis code requires e.g. GPUs or is being developed on ARM CPUs.

3 Usage of ARM CPUs in ATLAS

In 2024 the ATLAS experiment has started to use ARM CPUs in Monte Carlo (MC) simulation and reconstruction production and user analysis jobs. Figure 2 shows the number of

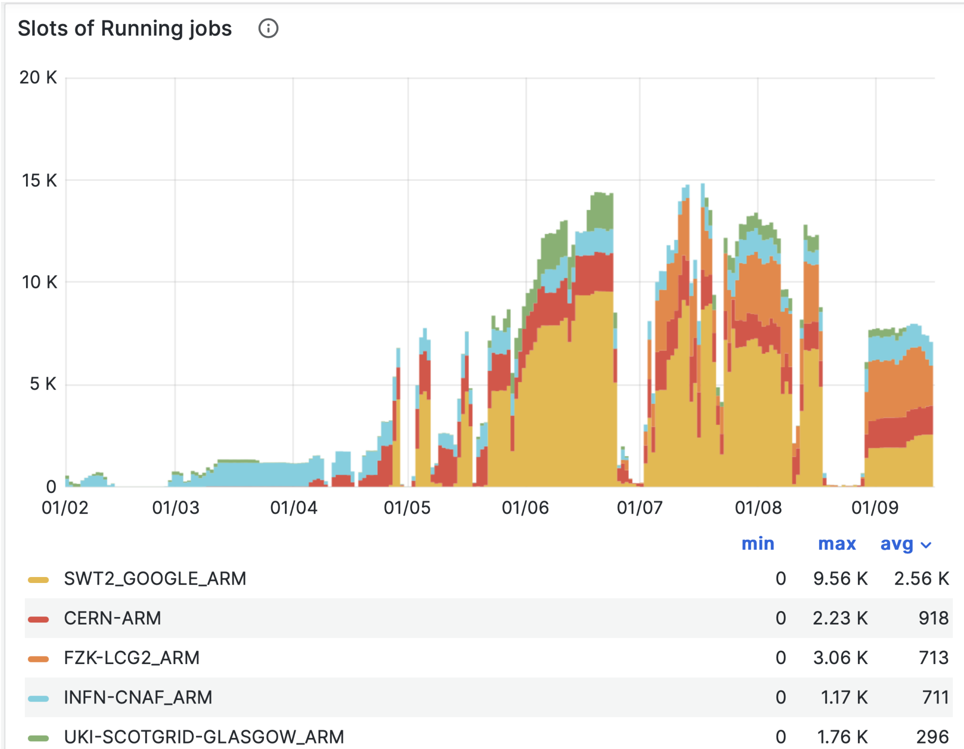

Figure 2. The number of concurrently running PanDA jobs using ARM CPUs on WLCG grid resources from February, 1st 2024 until September, 15th 2024 with a maximum of up to fifteen thousand concurrently running jobs.

concurrently running PanDA jobs using ARM CPUs on WLCG grid resources from February, 1st 2024 until September, 15th 2024 with a maximum of fifteen thousand concurrently running jobs. ATLAS will be the first WLCG experiment that will accept ARM resources as pledge in 2025/26. SWT2_GOOGLE_ARM is a special PanDA queue: it is an extension of the existing WLCG Tier2 site at UT Arlington by using extra CPU resources in the Google Cloud Platform (GCP) [11]. The ARM CPUs used are either Ampere Altra/Neoverse-N1 or Google Axion CPUs. The compute nodes had been initially configured as spot instances with preemption and later as standard resources.

At present there are seven ARM based nightly builds setup for the ATLAS offline software using the gcc compiler version 13.1 [12] for development and production usage. These are built on four build machines provided by CERN IT using Ampere Altra/Neoverse-N1 ARM CPUs and automatically installed on CVMFS [13] for world wide distribution. No specific settings have been used in the ARM builds, except for a different link flag called "max-page-size". In local single tests, speed-ups of Geant4 simulation by 2-3% can be achieved when using the target architecture setting *CXXFLAGS="-march=armv8.3-a -mtune=neoverse-n1"* on Ampere Altra CPUs or *CXXFLAGS="-march=armv9.2-a -mtune=neoverse-v2"* on Nvidia Grace CPUs. The ATLAS offline software stack can also be compiled successfully with the LLVM Clang compiler version 17 [14] in combination with a gcc 13.1 version. Small numerical precision difference for floating point numbers between x86_64 and ARM processors can be observed at the level of $10^{-4} - 10^{-6}$ when different math function libraries and optimizations are used.

On Apple silicon ARM CPUs the ATLAS offline software nightly and stable releases can easily be used for development and execution using the Lima container system [15]. Lima launches Linux virtual machines with automatic file sharing and port forwarding. Full instructions how to setup and use these containers are provided [16] for all ATLAS software developers.

4 Usage of GPUs on the Grid in ATLAS

During the on-going LHC Run3 ATLAS is not using GPUs in the offline software in production. There are major on-going R&D efforts to port parts of the MC simulation, reconstruction and high level online trigger code to use GPUs for the HL-LHC. Since March 2024 the ATLAS offline software supports the usage of GPUs with CUDA 12.4.1 [17]. This version of CUDA supports gcc version 13.1 which is the current production compiler version in ATLAS. Parts of the CUDA software development kit are allowed to be freely redistributed and the ATLAS offline software works fine with these parts of the software distributed via CVMFS on WLCG sites via PanDA. An important ingredient to the successful PanDA GPU queue configuration is a matching CUDA Linux kernel driver version at the PanDA site with the one from CUDA version 12.4.1 or newer. Since there are no automated procedures in place yet, every PanDA site that offers GPUs has to be individually contacted to keep the CUDA Linux kernel driver version up to date. In the future when GPUs will be more commonly used in production an automated WLCG wide procedure for CUDA Linux kernel driver updates should be put in place.

Table 1 shows the status summary of available PanDA sites with GPU support as of September 2024. All available GPUs are from Nvidia and therefore the number, type, Linux kernel and CUDA versions have been determined using the *"nvidia-smi"* command line tool. At present ATLAS offline software workflows are foreseen to use one GPU per PanDA job but potential future usage options could include GPU device sharing or whole node job scheduling.

Table 1. Status of CUDA support with ATLAS offline software Athena at various PanDA sites with Nvidia GPUs as of September 2024.

PanDA site	Nvidia GPU type	GPUs	vCPUs	Kernel driver	CUDA version	Works with Athena ?
BNL	A100				12.2	no
INFN-T1	Tesla K40m	1	1	460.106.00	11.2	old GPU
Manchester	Tesla T4	4	1/8	560.35.03	12.6	yes
OU_OSCER	K20	1	1		11.x	old GPU
QMUL	A100 40GB	1	1/8	550.54.15	12.4	(yes)
SLAC	A100 40GB	1	1	535.161.07	12.2	no
FZK-LCG2	V100S 32GB	8	8	555.42.02	12.5	yes
NERSC	A100		8		12.2	no

For CPU, GPU and memory resource monitoring of PanDA jobs ATLAS makes extensive use of the HSF/ATLAS tool prmon [18]. For Nvidia GPU monitoring prmon currently parses the text output of the nvidia-smi command line tool to collect information about the GPU resources usage and provides reasonable results for 1 or 4 GPUs but not yet for 8 GPUs in parallel. In the future a more robust implementation via the C-API is desirable together with

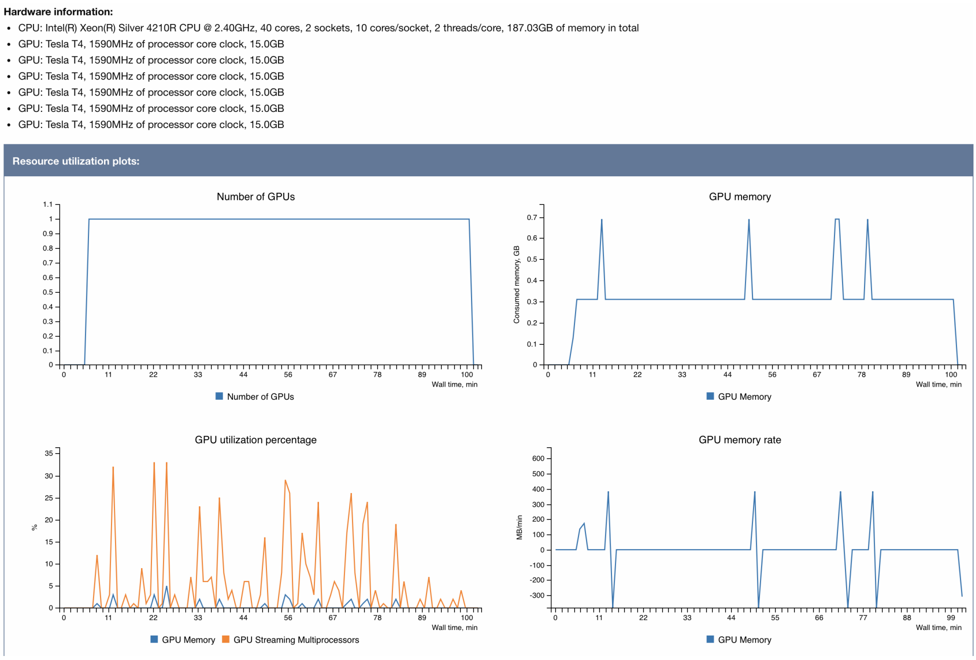

Figure 3. GPU resource monitoring over time of an Athena trigger reconstruction reprocessing PanDA job at Manchester. The top row shows the number of GPUs (left) and GPU memory usage (right). The lower row shows the GPU utilization fraction (left) and GPU memory throughput rate (right).

the support for more GPU vendors. Figure 3 shows the GPU resource monitoring over time of an Athena trigger reconstruction reprocessing PanDA job at Manchester.

5 HepScore23 benchmarks

In 2023 WLCG has replaced the HepSpec06 (HS06) benchmark for Grid CPU resource pledging with a new benchmark called HepScore23 combining seven WLCG experiment CPU based workflows. GPU workflows are not yet included in this version of the benchmark. Table 2 shows a comparison of the HepScore23 benchmark results on various x86_64 and ARM CPUs. When benchmarking a full CPU node all workflows are scaled to the number of available CPU cores to fully load the machine. The different ARM CPUs show very competitive results in comparison to the tested x86_64 CPUs.

Table 2. Comparison of HepScore23 benchmark results on various x86_64 and ARM CPUs.

Name	nCPU	HepScore23	HepScore23 / nCPU
Intel Gold 6326 (HepScore23 reference)	64	1018	15.9
Nvidia Grace Hopper	72	2319	32.2
Google Axion	72	2373	32.9
Apple M2 Air	8	141.4	17.7
Ampere Altra Neoverse-N1	20	349.4	17.5
Intel Xeon E5-2683 v4	16	258.5	16.2

6 Summary and Conclusions

The access and support of heterogeneous resources will help to address the HL-LHC computing resource challenge. Since the beginning of 2024 the software stack of the ATLAS experiment at the LHC at CERN is used in production on ARM CPUs on Grid and Cloud resources through the ATLAS production system based on PanDA and Rucio. Software developers can use the full software stack through Lima containers on Apple silicon ARM CPUs. The usage of Nvidia GPUs has been enabled in the ATLAS offline software stack and tested at various PanDA sites. The prmon resource monitoring tool provides detailed information about CPU, GPU and memory usage. The HepScore23 benchmark runs seemingly on ARM CPU and awaits an update to also support GPUs.

References

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

[2] ATLAS Software and Computing HL-LHC Roadmap, URL https://cds.cern.ch/record/2802918 [accessed 2024-12-09]

[3] ARM Architecture Reference Manual for A-profile architecture, URL https://developer.arm.com/documentation/ddi0487/latest/ [accessed 2024-12-09]

[4] Athena software framework of the ATLAS experiment at CERN, URL https://doi.org/10.5281/zenodo.4772550

[5] Worldwide LHC Computing Grid, URL http://cern.ch/lcg [accessed 2024-12-09]

[6] D. Giordano *et al.*, HEPScore: A new CPU benchmark for the WLCG, EPJ Web Conf. **295** (2024), 07024.

[7] T. Maeno *et al.*, PanDA: Production and Distributed Analysis System, Comput. Softw. Big Sci. **8** (2024) no.1, 4.

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

[9] F. B. Megino *et al.*, *Operational experience and R&D results using the Google Cloud for High-Energy Physics in the ATLAS experiment*, Int. J. Mod. Phys. A **39** (2024) no.13-14, 2450054.

[10] A. Anisenkov *et al.*, *CRIC: Computing Resource Information Catalogue as a unified topology system for a large scale, heterogeneous and dynamic computing infrastructure,* EPJ Web Conf. **245** (2020), 03032.

[11] Google Cloud Services, URL https://cloud.google.com [accessed 2024-12-09]

[12] The GNU Compiler Collection, URL https://gcc.gnu.org/releases.html [accessed 2024-12-08]

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

[14] Clang compiler front end, URL https://clang.llvm.org/index.html [accessed 2024-12-09]

[15] Lima: Linux Machines, URL https://github.com/lima-vm/lima [accessed 2024-12-09]

[16] ATLAS Lima instructions and configuration, URL https://atlassoftwaredocs.web.cern.ch/athena/containers/lima/ [accessed 2024-12-09]

[17] CUDA, parallel computing platform and programming model invented by Nvidia, URL https://docs.nvidia.com/cuda/cuda-installation-guide-linux/ [accessed 2024-12-09]

[18] Process monitoring prmon, URL https://github.com/HSF/ [accessed 2024-12-09]