The ATLAS experiment software on ARM

Johannes Elmsheuser[1,*], *Fernando* Barreiro Megino[2], *Alessandro* De Salvo[3], *Asoka* De Silva[4], *Reiner* Hauser[5], *Dmitri* Konstantinov[6], *Attila* Krasznahorkay[6], *Mario* Lassnig[6], *Andre* Sailer[6], and *Scott* Snyder[1] on behalf of the ATLAS Computing Activity

[1]Brookhaven National Laboratory, Upton, NY, USA
[2]University of Texas, Arlington, TX, USA
[3]Sapienza Universita e INFN, Roma I, Italy
[4]TRIUMF, Vancouver, Canada
[5]Michigan State University, East Lansing, MI, USA
[6]CERN, Geneva, Switzerland

Abstract. With an increased dataset obtained during the Run 3 of the LHC at CERN and the even larger expected increase of the dataset by more than one order of magnitude for the HL-LHC, 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 architecture is becoming a competitive and energy efficient alternative. Some surveys indicate its increased presence in HPCs and commercial clouds, and some WLCG sites have expressed their interest. Chip makers are also developing their next generation solutions on ARM architectures, sometimes combining ARM and GPU processors in the same chip. Consequently it is important that the ATLAS software embraces the change and is able to successfully exploit this architecture.
We report on the successful porting to ARM of the Athena software framework, which is used by ATLAS for both online and offline computing operations. Furthermore we report on the successful validation of simulation workflows running on ARM resources. For this we have set up an ATLAS Grid site using ARM compatible middleware and containers on Amazon Web Services (AWS) ARM resources. The ARM version of Athena is fully integrated in the regular software build system and distributed in the same way as other software releases. In addition, the workflows have been integrated into the HEPscore benchmark suite which is the planned WLCG wide replacement of the HepSpec06 benchmark used for Grid site pledges. In the overall porting process we have used resources on AWS, Google Cloud Platform (GCP) and CERN. A performance comparison of different architectures and resources will be discussed.

ATL-SOFT-PROC-2023-008
18 August 2023

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

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

more than one order of magnitude for the HL-LHC from 2029 onwards, the ATLAS experiment is reaching the limits of the current data processing and especially simulation model in terms of these CPU resources and an extensive program for software upgrades towards the HL-LHC has been set up. The ARM CPU architecture [2] is becoming a competitive and energy efficient alternative.

In the following sections we report on the successful port of the ATLAS software stack with its dependencies to ARM CPUs on the CentOS7 Linux operating system. This new software stack successfully passed a standard physics validation for reconstruction and full detector simulation workflows. ARM computing resources at the Cloud computing provider Amazon Web Services (AWS) [3] have been integrated into the ATLAS workflow management system PanDA [4] and the ATLAS data management system Rucio [5]. In addition the workflows have been integrated into the new HEPscore benchmark suite [6], which is the planned WLCG [7] wide replacement of the HepSpec06 benchmark used for Grid site pledges.

Future larger resources at High Performance Computing centers might become available, similar to the Fugaku HPC at the Riken Center for Computational Science in Japan, which is the number 2 in TOP500 supercomputer list of June 2023 [8]. With the successful port and validation of particle physics software on ARM CPUs, WLCG Grid sites are also starting to be interested in ARM CPUs as a cost and electrical power efficient system.

2 ATLAS software

The ATLAS software stack consists of several million lines of code in C++ and Python. At present it is mainly built on the CentOS7 Linux operating system. In addition there are a few nightly builds on AlmaLinux9. The software stack is divided into three major blocks, which are maintained by three different teams:

- ATLAS offline software with its overarching project Athena [9],
- ATLAS online trigger and data acquisition TDAQ software stack,
- External packages like e.g. ROOT, Geant4, Python, Monte Carlo Generators.

The first two items are maintained by two different groups in ATLAS while the last item is maintained by the CERN EP/SFT group. The ATLAS online trigger uses the Athena framework and the software is managed coherently with the offline software part. Figure 1 summarises these building blocks of the ATLAS offline software. All these packages have been successfully ported to the 64-bits ARM flavour of CentOS7 and fully integrated into the automatic ATLAS nightly release build system. The ATLAS build makes extensive use of CMake [10] for build automation, testing, packaging and installation of the software. Several small changes had to be applied to make the build procedures more generic across the default x86_64 and new ARM platforms.

Four dedicated builds for Athena, AthSimulation, AnalysisBase and DetCommon projects are built every night from the standard ATLAS Athena gitlab master branch [11] and the binaries are distributed via the read-only caching file system CVMFS [12]. The release builds successfully pass the continuous integration unit tests and a set of larger scale integration tests. The stable production releases Athena 23.0.3 and 23.0.14 have been used in the physics validation as described later in Section 4.

No specific settings have been used in the ARM builds, except for a different link flag called “max-page-size”. The current production Athena release version 23.x use the gcc 11.2 compiler [13]. In local single tests, using the “armv8.6-a” target architecture setting on the Google Cloud’s [14] “Neoverse-N1” processor, this compiler version demonstrated a

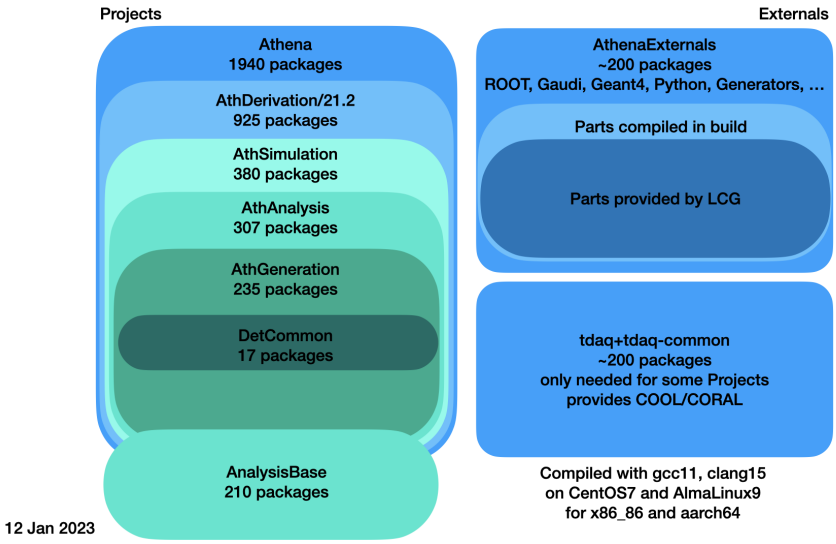


Figure 1: Overview of the building blocks of the ATLAS experiment offline software.

1–3% improvement in execution time for reconstruction and simulation workflows compared to the standard compiler setting. The ATLAS offline software stack can also be compiled successfully with the LLVM Clang compiler version 16 [15] in combination with a gcc 11.2 version of the externals and TDAQ parts.

The numerical precision is slightly different between x86_64 and ARM processors. This can in very rare cases result in floating point exceptions (FPE) of reconstruction or simulation algorithms. The ATLAS offline software uses a dedicated mechanism to trace and better diagnose these rare FPEs with x86_64 assembler code on these types of CPUs. This code has not yet been ported to ARM CPUs but poses no problem in a successful processing on ARM CPUs. The potential numerical differences due to different numerical precision show some fluctuations in physics objects quantities at the level of $10^{-4} - 10^{-6}$.

3 ARM resources

The amount of Linux based ARM resources is still rather limited for standard CERN users. There are a very limited amount of Ampere Altra and Cavium ThunderX2 machines physically on the CERN premises. A small public cluster with Ampere Altra CPU is provisioned through the Oracle Cloud [16]. One of these Oracle Cloud ARM machines has been integrated into the ATLAS offline nightly build system for Athena release builds.

In addition Ampere Altra/Neoverse-N1 nodes in the Google Cloud have been used for repeated ATLAS software build and performance tests in the software porting part of the project.

ARM CPU resources from the AWS Cloud provider with its Graviton2 and Graviton3 processors have been fully integrated into ATLAS PanDA and Rucio based production system. This procedure is described in detail in [17]. This integration grants the possibility of elastic scaling to a few thousand ARM CPUs to be used for a standard physics validation. The full ARM WLCG Grid setup has been made available with operating system containers, Grid middleware and Kubernetes [18] based workflow job submission.

Future larger resources at High Performance Computing centers are planing to provide some ARM CPU partitions and WLCG sites are also starting to be interested in ARM CPUs as a cost and electrical power efficient system.

4 Physics validation

In summer and autumn 2022 the Geant4 MC simulation workflow was chosen as the first candidate for a technical physics validation. One million events of $t\bar{t}$ decays in the standard ATLAS Run 3 pp–collision setup were simulated in release Athena version 23.0.3 on ARM Graviton2 CPUs using the AWS integration in the ATLAS production system described in the previous section. The same events were simulated in parallel on x86_64 CPUs using standard WLCG resources. All other steps in the MC event generation, reconstruction and analysis chains were executed on x86_64 CPU WLCG resources. The execution on AWS produced 1k files with a total size of ≈700 GB and took about 1.5 days on 300 8-core CPU PanDA job slots with a cost of around 1800 USD. The comparison showed a very good agreement of all physics objects distributions between the ARM and x86_64 produced events with some fluctuations reported. These fluctuations were well within the expectations of the numerical precision differences of the CPUs and the statistical uncertainty of the simulated samples.

In March 2023 the reconstruction workflow successfully passed the physics validation using the release Athena version 23.0.14. Similar to the simulation validation, reconstruction workflow samples were produced on ARM Graviton2 CPUs on AWS and compared to regular x86_64 CPU produced samples using WLCG sites. All other workflows for MC event generation, simulation and subsequent analysis were executed on x86_64 WLCG sites. 13 different MC physics processes with 100k events each and no extra pile-up were processed on 130 8-core CPU PanDA job slots and produced 215 GB of output in total. The jobs took in total about 1.5 days and had a total cost of about 450 USD.

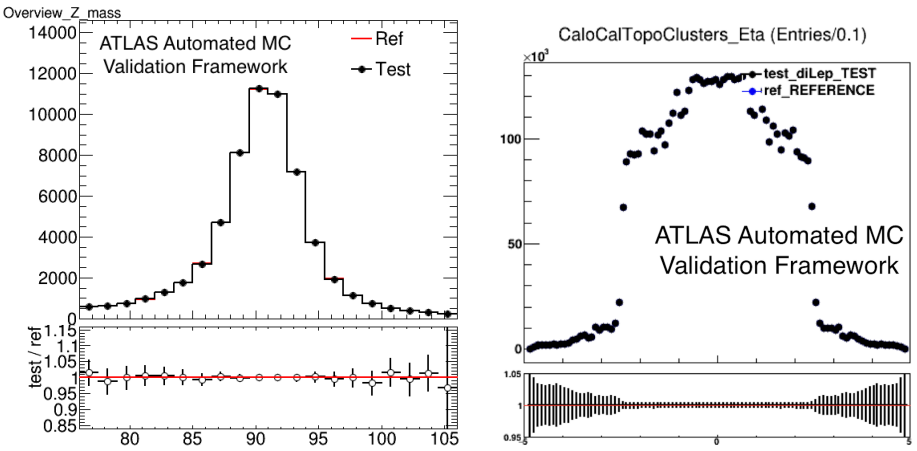
Figure 2 shows examples from the ATLAS automated MC validation framework with distributions comparing the reference x86_64 vs. the test samples production on ARM. Very similar to the simulation validation the comparison shows a very good agreement of all physics objects distributions between the ARM and x86_64 produced events with some fluctuations in regions with low MC statistics. A comparison of the wall clock time average of the ARM vs. the x86_64 PanDA jobs shows that the ARM reconstruction tasks on AWS Graviton2 CPUs was about 8% faster that the average x86_64 WLCG CPUs.

5 HEPscore benchmark

WLCG is planning to replace the HepSpec06 (HS06) benchmarks that are currently used for Grid resource pledging with a new benchmark called HEPscore based on experiment workflows later in 2023 [6]. The ATLAS experiment has integrated the following three workflows using Run 3 pp–collisions workflows using Athena release 23.0.3 for x86_64 and ARM/aarch64:

- Sherpa MC event generation of $t\bar{t}$ decays using one CPU thread,
- Data reconstruction using four CPU threads,
- Geant4 MC simulation of $t\bar{t}$ decays using four CPU threads.

When benchmarking a full CPU node all the workflows are scaled to the number of available CPU cores to fully load the machine. The performance of the ATLAS benchmarks has been tested on several x86_64 and ARM machines at CERN, Google (Intel Cascade Lake, AMD



(a) (b)

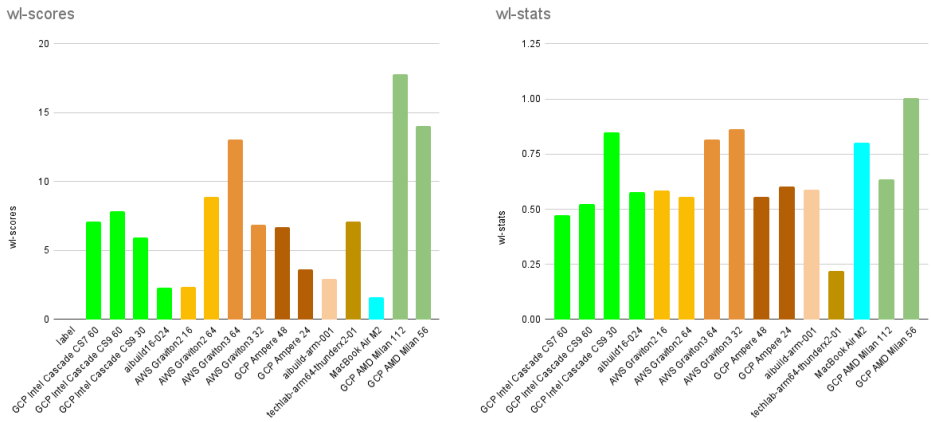
Figure 2: Example distributions from the ATLAS automated MC validation framework: (a) invariant mass $m_{\mu\mu}$ of simulated $Z \rightarrow \mu\mu$ decays. The x-axis shows $m_{\mu\mu}$ in units of GeV and the y-axis the number of events. (b) ϕ coordinate of the reconstructed calorimeter clusters for simulated multi-jet events. The x-axis shows the ϕ coordinate and the y-axis the number of events. In both plots (a) and (b) the events produced on ARM CPUs are labelled with "Test" while the events produced on x86_86 CPUs are labelled with "Ref" or "Reference".

EPYC Milan, Ampere Altra up to 112 CPU threads), AWS (Graviton2 and 3 with up to 64 threads) and an Apple M2 ARM CPU.

Figure 3 (a) shows the HEPscore results for the ATLAS reconstruction workflow and different x86_64 and ARM processors using the full node with all available threads while Figure 3 (b) shows the HEPscore results for only a single four-threads reconstruction workflow. The large differences between the different CPU types can be easily explained mainly with the number of available CPU threads of the different CPU models. The single workflow benchmark provides a more easy to understand comparison of the CPUs per single workflow. The different ARM CPUs show very competitive results in comparison to the tested Intel and AMD x86_64 CPUs.

6 Summary and Conclusions

The software stack of the ATLAS experiment at the LHC at CERN has successfully been ported to ARM on CentOS7 Linux systems. It has been fully integrated into the ATLAS production system based on PanDA and Rucio and successfully passed the physics validations for MC simulation and reconstruction workflows using AWS Gravition2 CPU resources. Three standard ATLAS production workflows have been integrated into the new WLCG HEPscore benchmark which allows a direct comparison of different x86_64 and ARM CPU types with good performances for the latter. ATLAS is planning to migrate to AlmaLinux9 in the coming months and there are no indications that this transition will be problematic for the ARM based releases.



(a) (b)

Figure 3: (a) HEPscore results for the ATLAS reconstruction workflow and on different x86_64 and ARM processors using (a) the full node with all available threads (wl-scores) and (b) a single workflow with four CPU threads only (wl-stats). Larger values of wl-scores and wl-stats are better. ARM type CPUs coloured brown, orange and blue and are grouped in the middle of both plots as bars 5–13. The x86_64 type CPUs are at the very left and right of the plots and coloured green.

References

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

[2] ARM Architecture Reference Manual for A-profile architecture, URL https://developer.arm.com/documentation/ddi0487/latest/ [accessed 2023-08-14]

[3] Amazon Web Services Services, URL https://aws.amazon.com [accessed 2023-06-09]

[4] F. H. Barreiro Megino *et al.*, *PanDA for ATLAS distributed computing in the next decade*, J. Phys. Conf. Ser. **898** (2017) no.5, 052002

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

[6] Randall Sobie *et al.*, *HEPscore: a new benchmark for WLCG compute resources*, these proceedings

[7] Worldwide LHC Computing Grid, URL http://cern.ch/lcg [accessed 2023-06-09]

[8] TOP500 list of High Performance Computers, URL https://www.top500.org/lists/top500/2023/06/ [accessed 2023-06-12]

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

[10] CMake for build automation, testing, packaging and installation of software, URL https://cmake.org/overview/ [accessed 2023-06-12]

[11] Athena gitlab master branch, URL https://gitlab.cern.ch/atlas/athena/ [accessed 2023-06-13]

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

[13] The GNU Compiler Collection, URL https://gcc.gnu.org/releases.html [accessed 2023-06-12]
[14] Google Cloud Services, URL https://cloud.google.com [accessed 2023-06-09]
[15] Clang compiler front end, URL https://clang.llvm.org/index.html [accessed 2023-06-12]
[16] Oracle Cloud, URL https://www.oracle.com/cloud/ [accessed 2023-06-12]
[17] Fernando Barreiro Megino *et al.*, *Accelerating science: the usage of commercial clouds in ATLAS Distributed Computing*, these proceedings
[18] Kubernetes, URL https://kubernetes.io/docs/home/ [accessed 2023-06-09]