

ATLAS Software Installation on Supercomputers

Alexander Undrus¹ on behalf of the ATLAS Collaboration

¹Brookhaven National Laboratory, Upton, NY 11973, USA

Abstract. PowerPC and high-performance computers (HPC) are important resources for computing in the ATLAS experiment. The future LHC data processing will require more resources than Grid computing, currently using approximately 100,000 cores at well over 100 sites, can provide. Supercomputers are extremely powerful as they utilize hundreds of thousands of CPUs joined together. However, their architectures have different instruction sets. ATLAS binary software distributions for x86 chipsets do not fit these architectures, as emulation of these chipsets results in huge performance loss. This paper describes the methodology of ATLAS software installation from source code on supercomputers. The installation procedure includes downloading the ATLAS simulation release code with 0.7 million C++ and Python lines as well as the source code of more than 50 external packages, such as ROOT and Geant4, followed by compilation, and rigorous unit and integration testing. The presentation reports the application of this procedure at Titan HPC and Summit PowerPC at Oak Ridge Computing Facility (OLCF).

© 2018 CERN for the benefit of the ATLAS Collaboration.

Reproduction of this article or parts of it is allowed as specified in the CC-BY-4.0 license.

1 Introduction

ATLAS (A Toroidal LHC Apparatus) [1] is one of the largest collaborative efforts ever attempted in the physical sciences. Many important findings, such as Higgs discovery, were made and the physics research program is planned for the next 20 years. The new phase of LHC, named as High Luminosity LHC, aims to deliver an integrated luminosity of about 3000 fb⁻¹ over 10 to 12 years of operation [2]. Offline computing will need to use supercomputers to cope with about 10 times larger data volumes. ATLAS software needs adapting for execution on modern PowerPC and HPC architectures which are incompatible with the currently dominant x86-64 architecture.

2 Usage of Supercomputers in ATLAS Offline Computing

ATLAS already uses resources of HPC centers across the world. HPC delivered 20% of ATLAS wall clock usage in June 2017 – May 2018 [3]. Mira at Argonne Leadership Computing Facility was the first large scale non-x86 HPC where an important workflow of event generation was ported and then its largest jobs routinely run, with millions of parallel threads [4]. Titan at Oak Ridge Leadership Computing Facility (OLCF) was the first large scale HPC integrated with the ATLAS Production System. It allowed to run ATLAS

Simulation production jobs on used Titan resources (“backfill”) collecting millions of core hours per month [5].

3 Software Porting at Summit – Next Generation Supercomputer

The new Summit supercomputer at OLCF [6] will deliver more than 5 times the computational performance of Titan. Unlike Titan’s x86 CPUs, the Summit’s IBM POWER9 CPUs does not allow the direct use of ATLAS binary software distributions which are currently available for x86 platforms only. Installation from sources is a viable alternative to binary distributions as it allows avoiding ineffective x86 chipset emulation and considering hardware information at build time (e.g. by tweaking compiler options). Summit runs Red Hat Enterprise Linux (RHEL) version 7 operating system that is similar to the Linux systems ATLAS uses for software development and distribution. It allows using the same build commands, tools, and shell scripts. However, installing from sources requires taking care of the complex external dependencies of ATLAS software.

4 Installation from Sources with ATLAS Comprehensive Software Compilation (ACSC) Procedure

The ACSC procedure provides automatic installation of ATLAS software, together with all external packages, from source code on machines running Linux operating system on various platforms, including x86 and PowerPC. It features:

- Code download from git or web servers via HTTP or HTTPS.
- Use of native compilers and CMake [7] build tool installations.
- Four phases of external packages installations:
 1. ROOT scientific software toolkit [8] installation procedure is optimized for batch jobs needs (e.g. graphical functionalities disabled). It includes downloading and compiling external software, such as Python programming language, used also by ATLAS software.
 2. LCG common High Energy Physics software stack for the LHC experiments is installed with LCGMake tool [9]. Only 48 packages used by ATLAS software are installed out of about 250 packages available, such as sqlite, mysql, zlib, XercesC, expat, HepMC, HepPDT, CppUnit, AIDA.
 3. ATLAS versions of toolkits for simulation of particles and their passage through matter, such as Geant4 [10] are deployed and compiled with tools supplied in the ATLAS software release.
 4. GAUDI software architecture and framework [11] is installed last as it depends on many packages from previous phases.
- Download and compilation of the ATLAS software (Athena) release designed for specific tasks. Currently Athena Simulation releases are considered. Simulation is often the most CPU-consuming task in High Energy Physics experiments with complex detector designs, such as ATLAS. On the other hand, the simulation part only accounts for about one fifth of the full ATLAS software release size.
- Creation of TAR archive files with release source and binaries for installation on other machines or worker nodes together with the Linux shell script for a run-time environment setup.
- Validation of installation with quick tests verifying critical functionalities and small sample production jobs.

4 Results

The ACSC procedure was successfully applied for the installation from source of the Athena Simulation 21.0.34 software release on Titan HPC (compiler gcc 6.2) and Summitdev PowerPC (gcc 6.3) – a Summit early access system at OLCF, and on a x86 machine at CERN (gcc 6.2) for cross-check. Compilation times are shown in Table 1.

Table 1. Compilation (single stream make) times for different ACSC phases.

Phase	X86_64 compilation time	Titan compilation time	Summitdev compilation time
ROOT	20m	34m	27m
LCG stack	73m	100m	90m
Generators	28m	46m	40m
GAUDI	25m	44m	38m
Athena Simulation	4h	6h	6h

Few modifications to software and compiler options were needed:

- Replacement of `__linux` compiler macro with `__linux__`.
- Replacement of linker option `-Wl,-z,max-page-size=0x1000` as virtual addresses and file offsets for the 64-bit PowerPC processor family segments are congruent modulo 64 Kbytes (0x10000).
- Removal of checkpointing packages using architecture-specific assembly commands.
- Patch to ROOT Cling code generation mechanism for different functions addressing modes on PowerPC (reported to ROOT team).

Installations were successfully validated with various tests: ROOT functional tests, GAUDI example algorithms, Athena release unit tests, small sample simulation jobs.

5 Conclusions

The ACSC procedure for ATLAS software installation from source was developed. With this procedure, an Athena Simulation release was installed and tested on Summitdev PowerPC and Titan HPC without chipset emulation.

The author wishes to thank S. Panitkin and S. Snyder (Brookhaven National Laboratory), C. Leggett and V. Tsulaia (Lawrence Berkeley National Laboratory) for valuable advices. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

References

1. The ATLAS Collaboration, JINST **3**, S08003 (2008)
2. B. Schmidt, J. Phys. Conf. Ser. **706**, 022002 (2016)
3. K. De, *Running ATLAS Simulations on HPCs, PASC 2018 Proceedings* (to be published)
4. J. T. Childers et al., J. Phys. Conf. Ser. **664**, 092006 (2015)
5. F. Barreiro Megino et al., J. Phys. Conf. Ser. **898**, 092002 (2017)
6. *Summit – Oak Ridge National Laboratory Next High Performance Supercomputer*, <https://www.olcf.ornl.gov/olcf-resources/compute-systems/summit>

7. <https://cmake.org>
8. <https://root.cern.ch>
9. <https://gitlab.cern.ch/sft/lcgcmake>
10. <https://geant4.web.cern.ch>
11. <http://gaudi.web.cern.ch/gaudi>