

ATLAS software – porting for Summit

BigPanDA Technical Interchange Meeting

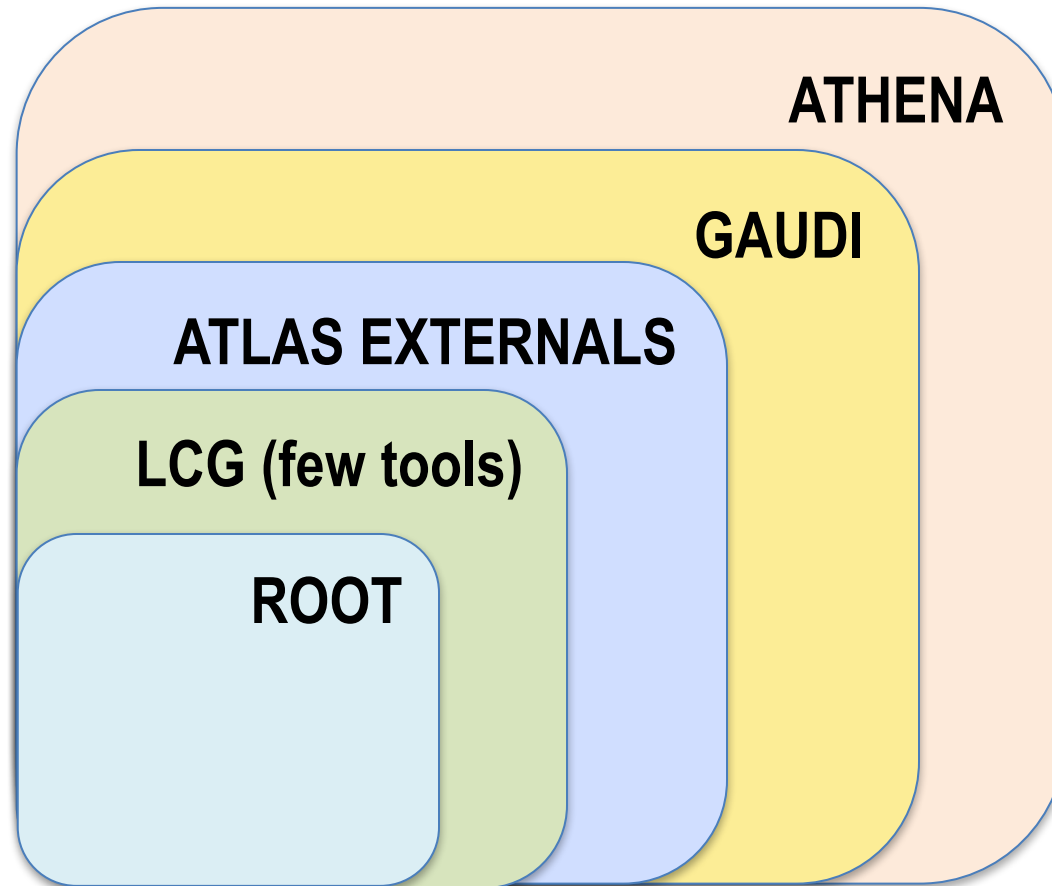
January 17, 2018



Alexander Undrus (BNL)

First presented at ADC TCB on Oct. 30, 2017:

Install ATLAS Software on supercomputers from sources taking the nesting structure of components into account



- Supercomputers architectures do not fit current ATLAS software distribution schemes
 - CVMFS is not available. “Extracting /cvmfs” – difficult, can affect performance and stability
 - PowerPC architecture has a different set of instructions, emulating a different chipset can result in a huge performance loss
- Installation from sources could be a better solution
 - Due to the above reason standard ATLAS containerized distributions do not fit well for PowerPC (but container technology can still be applied to the recompiled source code)

ATLAS Comprehensive Supercomputer Compilation ACSC Project

Another attempt of porting ATLAS software on:



Friendly Linux kernel and AMD CPUs
(ATLAS kits binaries work)

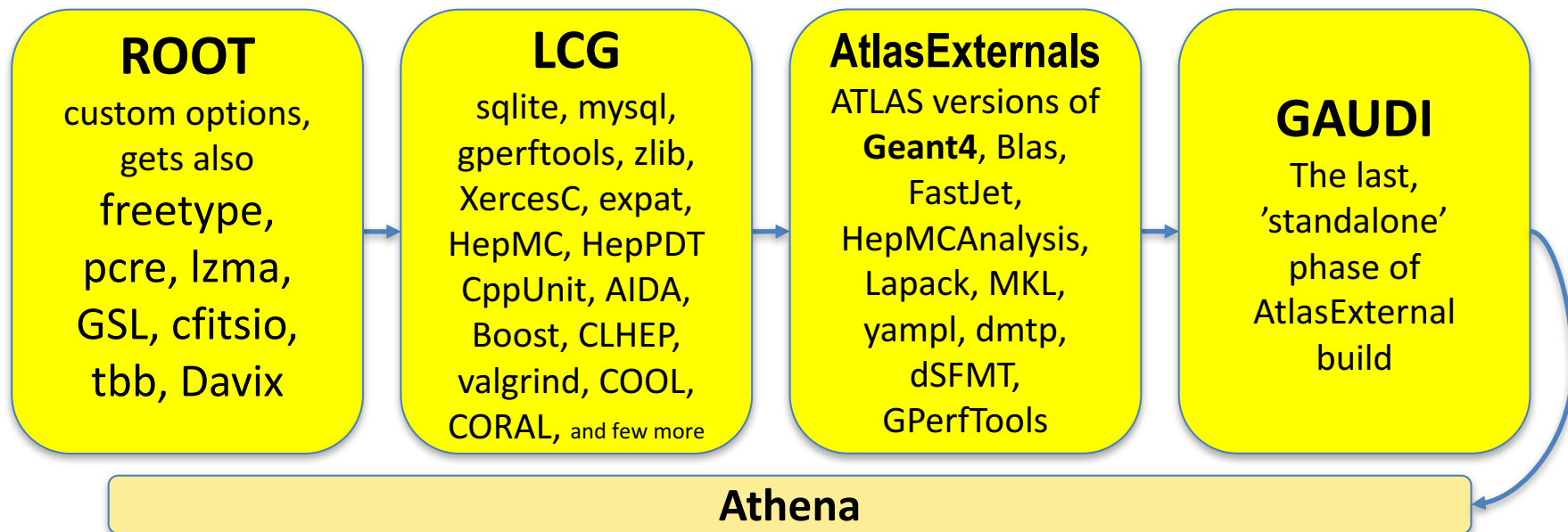


PowerPC, 10X of Titan
IBM CPUs, GNU Linux
(ATLAS kits binaries do **not** work)

- ACSC method
- Status of ACSC
 - Phase I is close to completion, on schedule
- Plans

ACSC Method

- Use **documented procedures** for external components
- Code is getting via **HTTP** (git, wget), **it works on supercomputers**
- **Preinstalled:** standard gcc version 6.2, cmake & python from LCG
- Component builds automatically get necessary externals installed
- **ROOT build is customized** to drop unnecessary functionalities
(some images processing, bonjour support, alien support)
- **Only 40 LCG packages** needed (out of 242 packages of LCG_88)
- **Validation** must follow installation



Project Timeline and Effort

- I. ACSC on ATLAS Farm – completed in December 2017
 - Goal: develop methodology
 - Code: AthSimulation 21.0.34
 - Only simple tests performed
- II. ACSC on Titan – by February 2017
 - No big effort/problems envisaged
- III. ACSC on Summit – by March software workshop
 - Also: investigate I/O behavior with Darshan tool on Titan (requires MPI compilation)

Note on Validation

- ATLAS software releases include ~ 300 unit tests
 - Good for 'smoke' testing purpose
- Newly compiled releases must undergo the full software and physics validation, including chain tests that emulate the sequence of jobs used in MC simulation and data processing

Phase 1 Result: SUCCESS

- Compilation chain *ROOT-LCG-Externals-Gaudi-Athena* is tested for *AthSimulation 21.0.34*
 - No AFS and CVMFS dependence
 - Hardware: ATLAS build VM (x86_64, 16-core, 30 GB RAM)

COMPONENT	COMPILATION TIME	RESULT
ROOT	15 minutes	OK
Selected ~20 LCG tools	63 minutes	OK (1 temp. hack)
AtlasExternals (Geant 4, etc.)	27 minutes	OK
Gaudi	36 minutes	OK
AthSimulation	4 hours	OK (2 temp.hacks)

- Validation: **99% tests passed**
 - 4 tests failed out of 317 (3 - always fail, one – complained about fancy missing python module)

Phase 1: Issues

- Hardwired AFS use in *XML/AtlasAuthentication*:
</afs/cern.ch/atlas/offline/external/AtlasAuth/v18/authentication.xml>
 - Solution: Package copied to a local disk
 - Few sub-packages in *AtlasPyFwdBwdPorts* failed to compile:
 - Solution: remove – not really used by athena
 - Installation procedure of COOL (Conditions DN interface from LCG) attempted to reinstall packages already install
 - Solution provided by CERN SFT group
- **Few issues were not showstoppers**

Conclusion

- The first phase of ACSC is **successful**
- **Methodology** of simulation releases installation was developed:
 - Compilation chain *ROOT-LCG-Externals-Gaudi-Athena*
 - Validation with 317 tests
- **Move to the next phase in the beginning of 2018:**
 - ACSC on Titan