The ATLAS experiment software on ARM

Johannes Elmsheuser (BNL) on behalf of the ATLAS Collaboration

08 - 12 May 2023, CHEP 2023

Acknowledgements

- Thanks to Andre Sailer and Dmitri Konstantinov from the CERN EP/SFT group
- Thanks to the ATLAS collaborators Fernando Barreiro (University of Texas at Arlington), Alessandro De Salvo (Sapienza Universita e INFN, Roma I), Matthew Doidge (Lancaster), Asoka De Silva (TRIUMF), Mario Lassnig (CERN), Reiner Hauser (Michigan State University), Attila Krasznahorkay (CERN), Scott Snyder (Brookhaven National Laboratory) and experts of the ATLAS distributed production and analysis and the physics validation teams

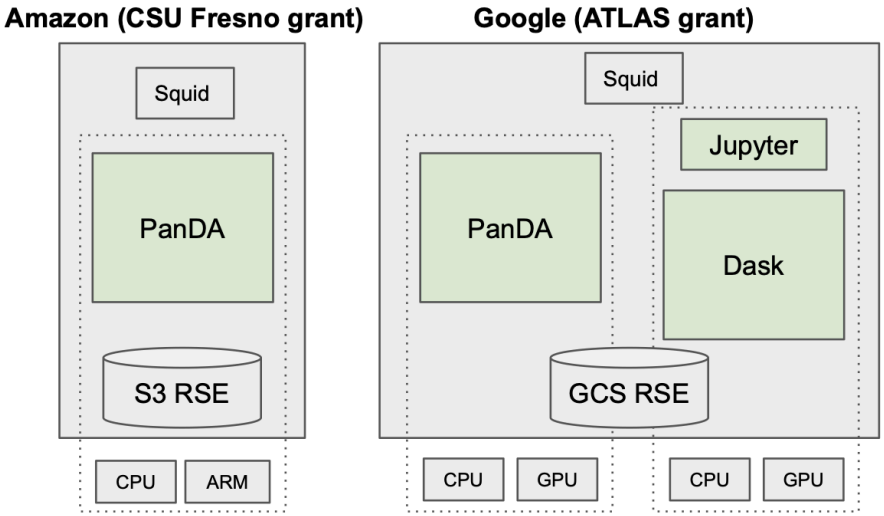
Introduction - Definitions of ARM/AArch64

- **What is ARM ?** (From Wikipedia)
  - Family of reduced instruction set computer (RISC) instruction set architectures for computer processors,
  - Arm Ltd. develops the architectures and licenses them to other companies for their products like system on a chip (SoC) and system on module (SOM) designs,
  - Low costs, minimal power consumption, and lower heat generation than their competitors
- **What is AArch64 ?** (From Wikipedia)
  - 64-bit extension of the ARM architecture family

Introduction (II) - Access to ARM resources

- Access to ARM resources:
  - CERN (usually single instances):
    - lxplus8/9-arm (in Oracle Cloud)
    - ATLAS build machine provisioned by CERN IT and Openlab on prem and in Oracle Cloud (Neoverse and Cavium ThunderX2)
  - AWS (Graviton2 and 3)
    $\rightarrow$ fully integrated in ATLAS PanDA/Rucio based production system (next slide)
  - Google Cloud (Ampere Altra/Neoverse-N1)
- Current and future large scale resources:
  - Fugaku HPC at Riken Center for Computational Science, Japan is #2 in TOP500 supercomputer list of November 2022 (link)
  - A few other HPCs plan to have ARM partitions
  - Grid sites are starting to be interested in ARM when the experiment software is validated

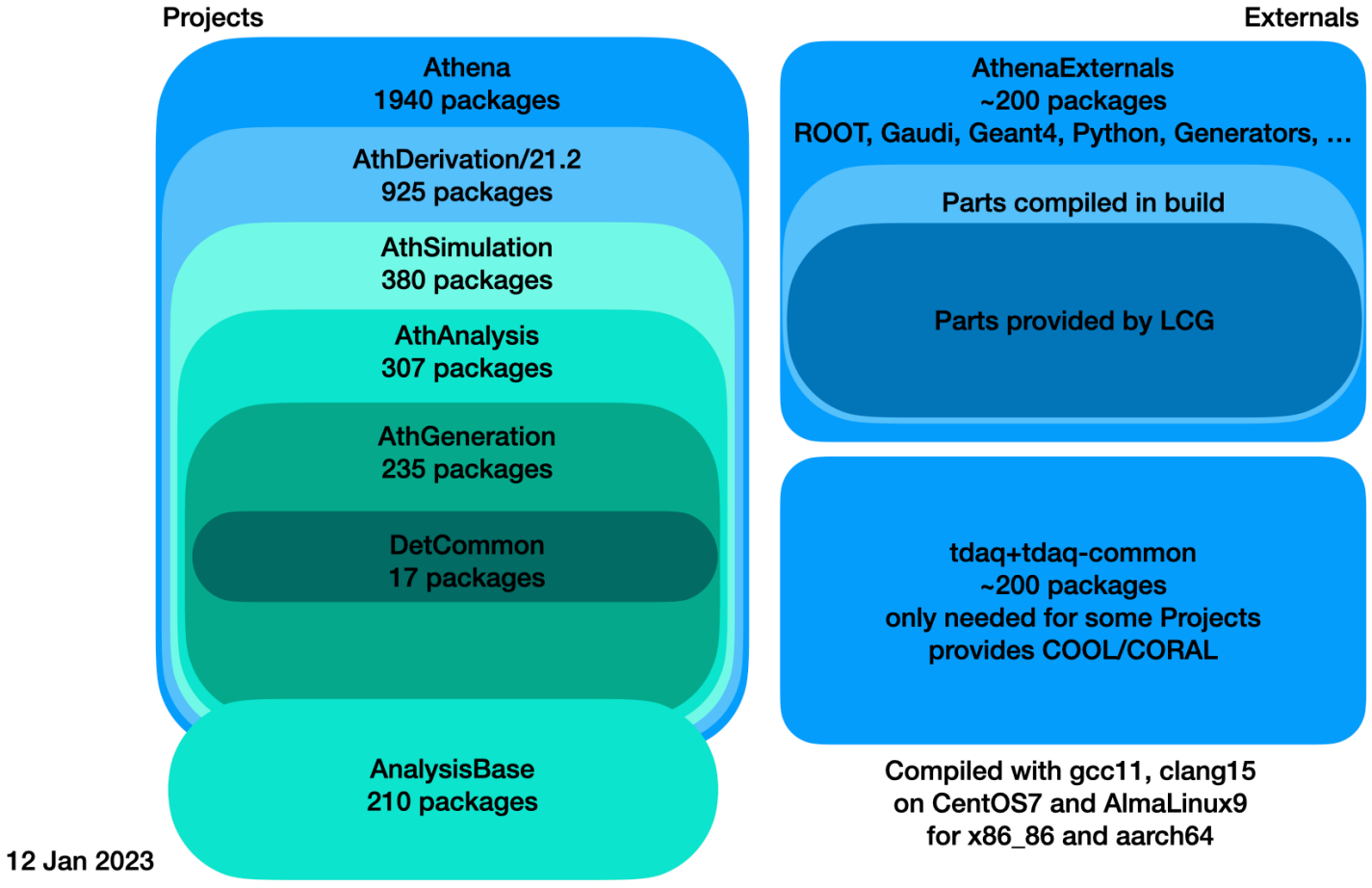
ATLAS PanDA and Rucio setup in AWS



- Self-contained, cloud-native, vendor-agnostic, auto-scaled, auto-healing
- Rucio:
  - Object Store + Signed URL + http protocol
  - Integrated in Rucio and WLCG MW
- PanDA:
  - Harvester submitting to K8S directly
  - CVMFS K8S plugin
  - Frontier Squid either in K8S cluster or as separate load balanced VM group
  - Auto-scaling: no jobs, almost no cost

- Full aarch64 grid setup available with OS container, middleware, Kubernetes etc.
- More details in talks from Fernando Barreiro (link) and Mario Lassnig (link) at this CHEP 2023 conference

ATLAS SOFTWARE SCHEMATIC OVERVIEW



- 3 major blocks build by different teams:
  - Projects: ATLAS offline software
  - Externals: CERN EP/SFT trough LCG layers and ATLAS offline software
  - TDAQ: ATLAS trigger/DAQ

Nightly builds on ARM/aarch64

ARM	master_AnalysisBase_aarch64-centos7-gcc11-opt	2023-04-18T0220	18-APR 10:34	0 (1)	0 (0)	N/A	N/A	18-APR 11:11
ARM	master_Athena_aarch64-centos7-gcc11-opt	2023-04-18T2300	19-APR 06:16	0 (1)	4 (4)	N/A	N/A	19-APR 09:21
ARM	master_AthSimulation_aarch64-centos7-gcc11-opt	2023-04-18T2101	18-APR 21:37	0 (0)	1 (1)	N/A	N/A	18-APR 22:11
ARM	master_DetCommon_aarch64-centos7-gcc11-opt	2023-04-18T2000	18-APR 20:02	0 (0)	0 (0)	N/A	N/A	18-APR 20:21

- 4 nightly builds for Athena, AthSimulation, AnalysisBase and DetCommon (link) projects fully integrated in standard ATLAS build system and available on CVMFS
- Selected stable Athena releases like 23.0.3 and 23.0.14 installed on CVMFS - used in physics validation (see later)
- Since December dedicated build machine provided by CERN IT hosted in Oracle Cloud, 40 vCPUs (Ampere Altra A1), 250 GB RAM, special manual CentOS7 installation
- Used before techlab-arm64-thunderx2-01 provided by CERN OpenLab and shared with CERN SFT
- LCG_102b_ATLAS_* / LCG_103 layers and tdaq/tdaq-common builds available and used in ARM/aarch64 builds
- Can easily build stand-alone docker/podman container for e.g. AthSimulation

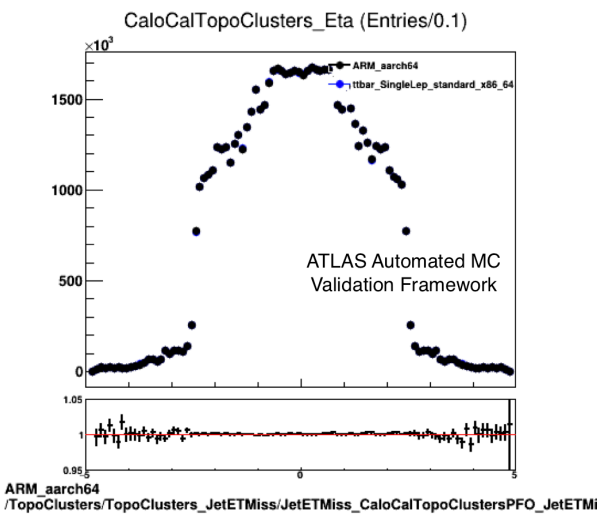
ARM/aarch64 ATLAS builds and caveats

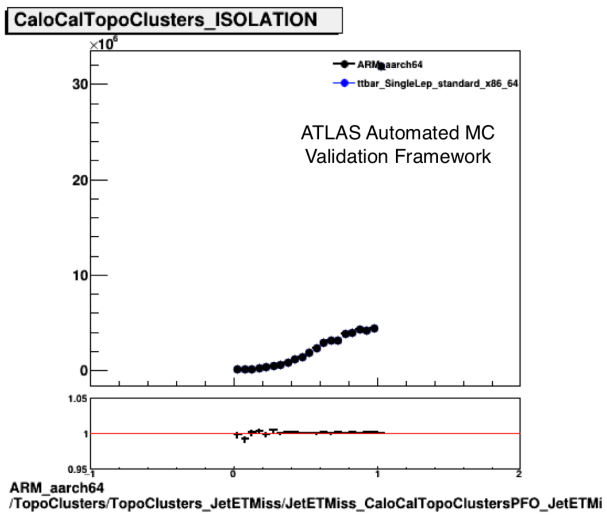
- Build flags
  - Using Armv8 defaults (gcc 11.2 allows up to armv8.6-a, gcc docu link)
  - No special `arch` option and no special compilation flags set apart from different linker flag in max-page-size
  - Able to build Athena/AthenaExternals with clang16 as well
- Floating Point Exception (FPE)
  - Athena FPE auditor code not working on ARM/aarch64 since it uses x86 specifics
  - Discussions about different FPE behaviours of x86_64 vs. ARM/aarch64 compilers: StackOverflow 1, StackOverflow 2, ARM compiler
- Potential numerical differences
  - E.g. due to different floating point libraries used - see StackOverflow link
  - Some fluctuations in physics objects at the level of $(10^{-4} - 10^{-6})$ (see later)
  - N.B. small numerical differences or Intel vs. AMD if IntelMathFunction used

Geant4 physics validation with Athena,23.0.3

- Summer/Autumn 2022 successfully passed Geant4 10.6 physics/technical validation
- 1 million $t\bar{t}$ events (1k files, ≈700 GB output 1.5 days, 300*8 core job slots, 1800 USD) with Geant4 produced with stable Athena,23.0.3 release at AWS Graviton2 PanDA queue
- Compared with same events produced on x86_64 grid sites - all subsequent steps (reconstruction, merging histogramming) on x86_64
- Some fluctuations in different areas reported, but at the level as in other physics/technical validations

ttbar_SingleLep

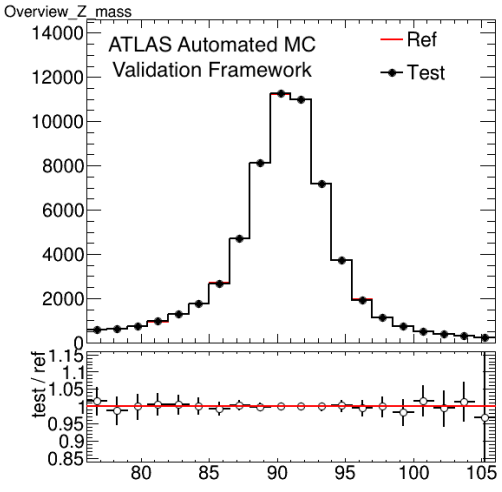




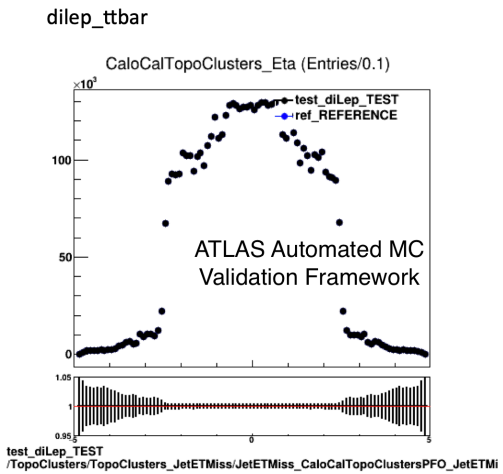
Example plots from calorimeter cluster validation of ARM/aarch64 (test) vs. x86_64 (ref)

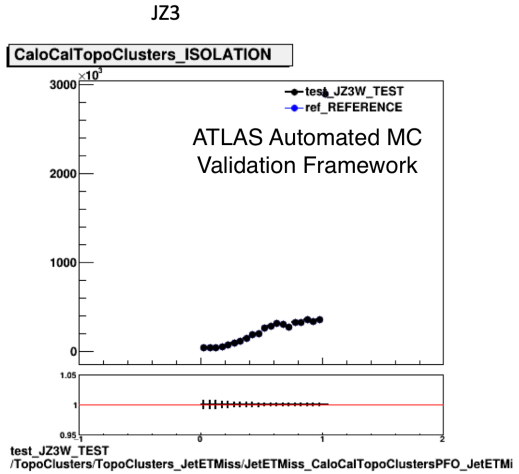
Reconstruction physics validation with Athena,23.0.14

- March 2023 successfully passed reconstruction physics validation
- Standard procedure of 13 MC physics processes (100k events each), no pile-up
- Digitization+Reconstruction step on AWS Graviton2 PanDA queue (130 jobs, 215 GB output, 1.5 days, 450 USD in total)
- Compared with same events produced on x86_64 grid sites - all subsequent steps (merging, histogramming) on x86_64 - task speed comparison in the backup



Reconstructed MC $Z \to \mu\mu$ invariant mass





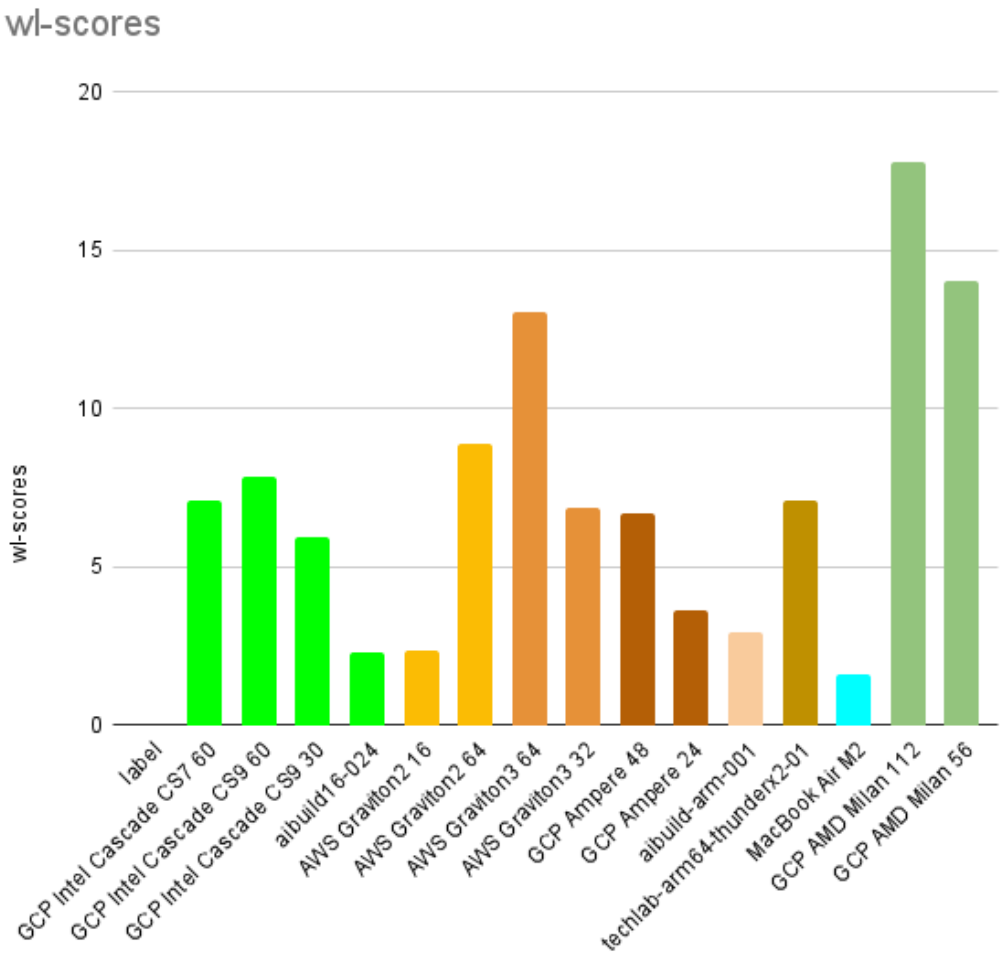
Perfect agreement

Example plots from calorimeter cluster validation

HepScore integration (I)

- WLCG is planning to replace HepSpec06 benchmarks used for resource pledging with experiments workflows based HepScore in 2023/24 - more details in HepScore workshop
- ATLAS has integrated 3 workflows in Athena,23.0.3 for x86_64 and ARM/aarch64 (gitlab link)
  - single core (times the number of cores) Sherpa (link)
  - AthenaMT 4-threads (times the number of cores/4) data17 reco with some small tweaks (link)
  - Geant4MT with 4 threads (times the number of cores/4) (link)
- Benchmarks/workflows can easily be executed using singularity/apptainer
- Ran benchmarks on a few nodes at CERN, Google (up to 112 vCPUs), AWS (up 64 vCPUs), Intel Cascade Lake, AMD EPYC Milan, Graviton2/3, Ampere, Apple M2 laptop

HepScore integration (II) - reconstruction scores, full node



Larger values are better

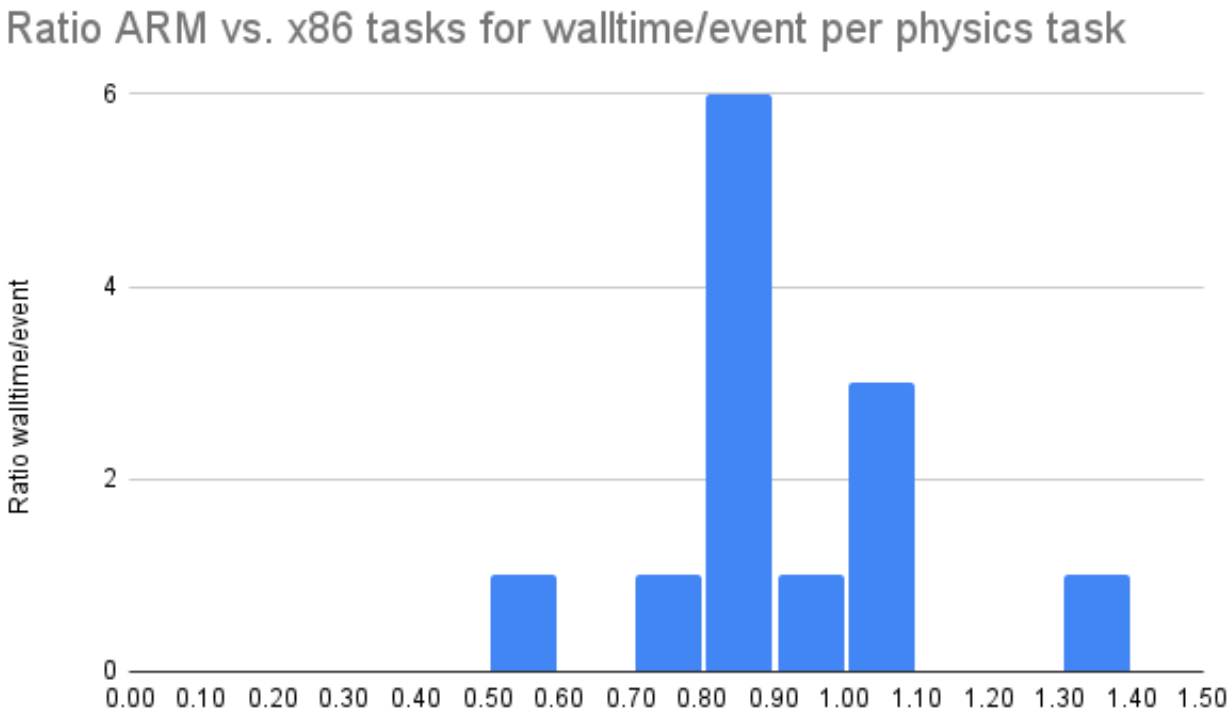
- HepScore numbers for ATLAS reconstruction
- Node fully packed with n×4 threads
  → Different number of available cores explains different scales !
- Orange/Brown/Blue: ARM flavours
- Green: Intel/AMD flavours
- Some hyperthreading or IO related differences
- Similar trends for Event generation and simulation
- See back-up for single workflow measurement

Further ideas, summary and Conclusions

- Further plans:
  - Migrate to AlmaLinux9 (all pieces already available)
  - Investigate further production workflows (e.g. MC generators)
  - Power consumption tricky to assess on virtualized hardware
- Summary:
  - Full ARM/aarch64 ATLAS nightly builds on CVMFS through regular nightly ATLAS build system
  - Geant4 and reconstruction physics/technical validation successfully passed using AWS through PanDA/Rucio production system
  - Athena,23.0.3 sherpa, simulation and reco workflows integrated into new HepScore benchmark
  - Project is part of ATLAS HL-LHC R&D projects and will be part of ATLAS computing technical design report for HL-LHC

BACKUP

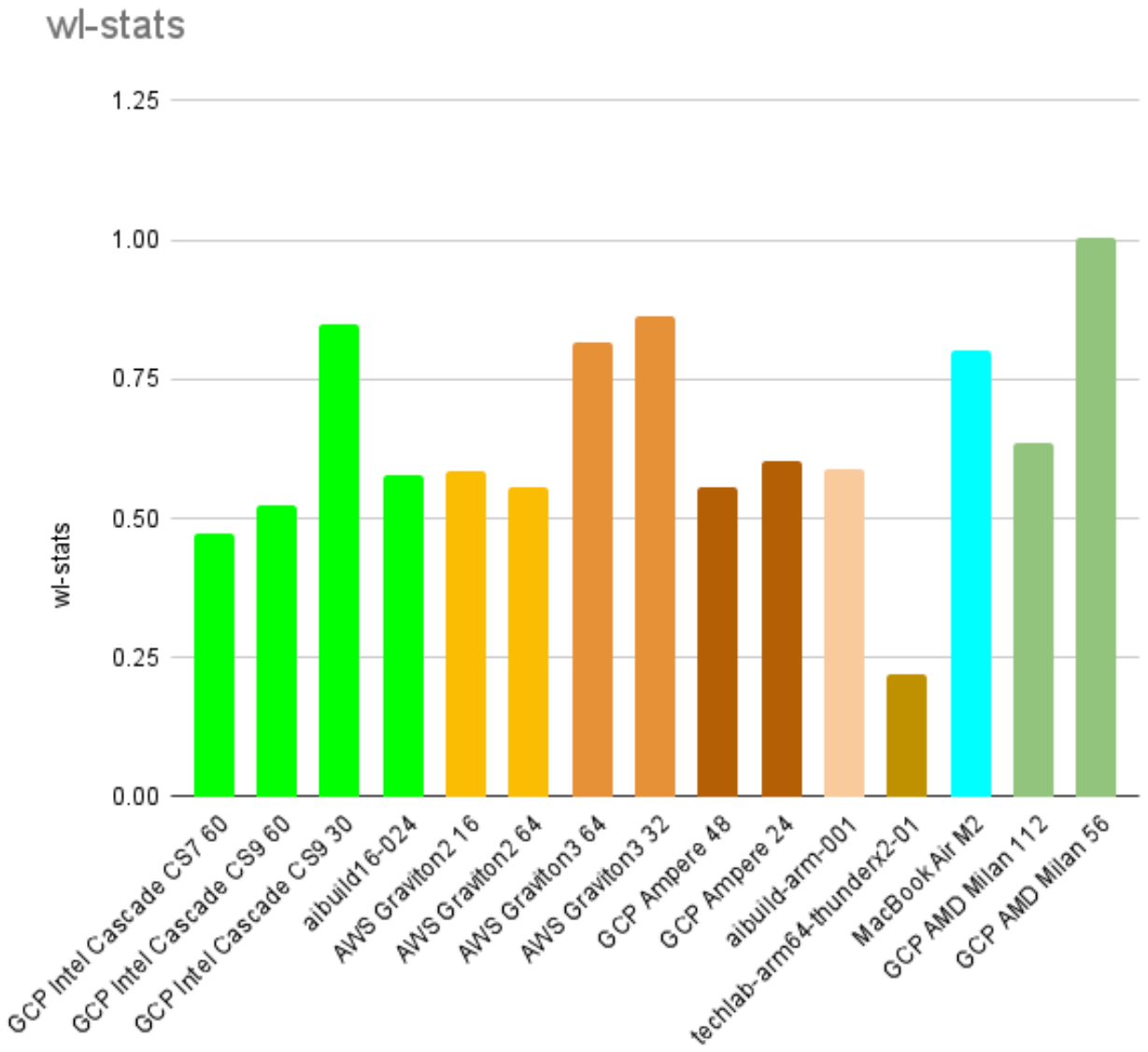
Reconstruction physics validation - PanDA tasks speed



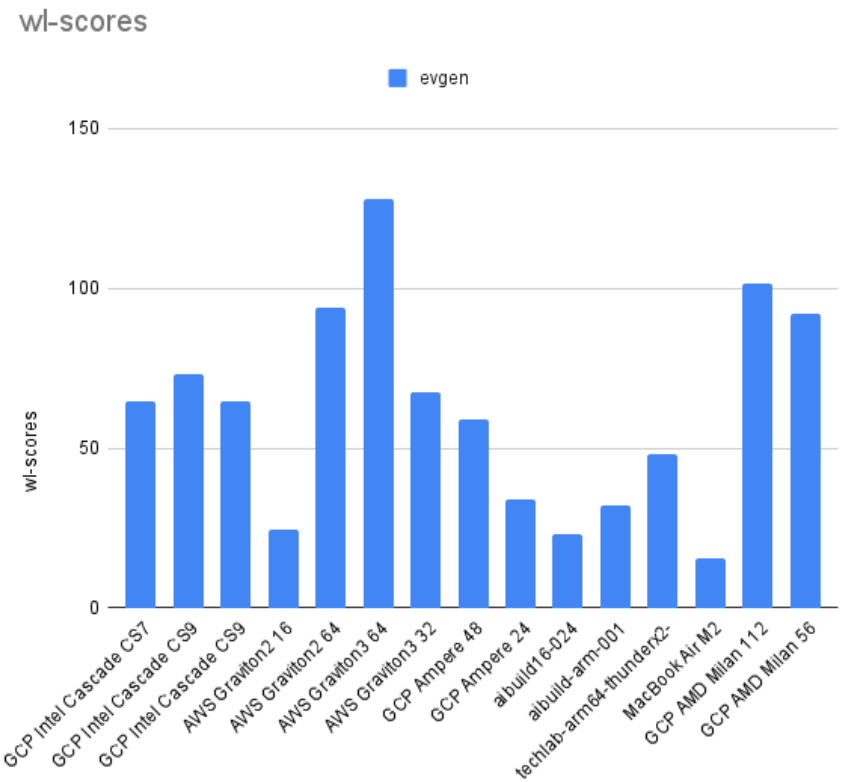
Ratio ARM vs. x86 Grid job averages:

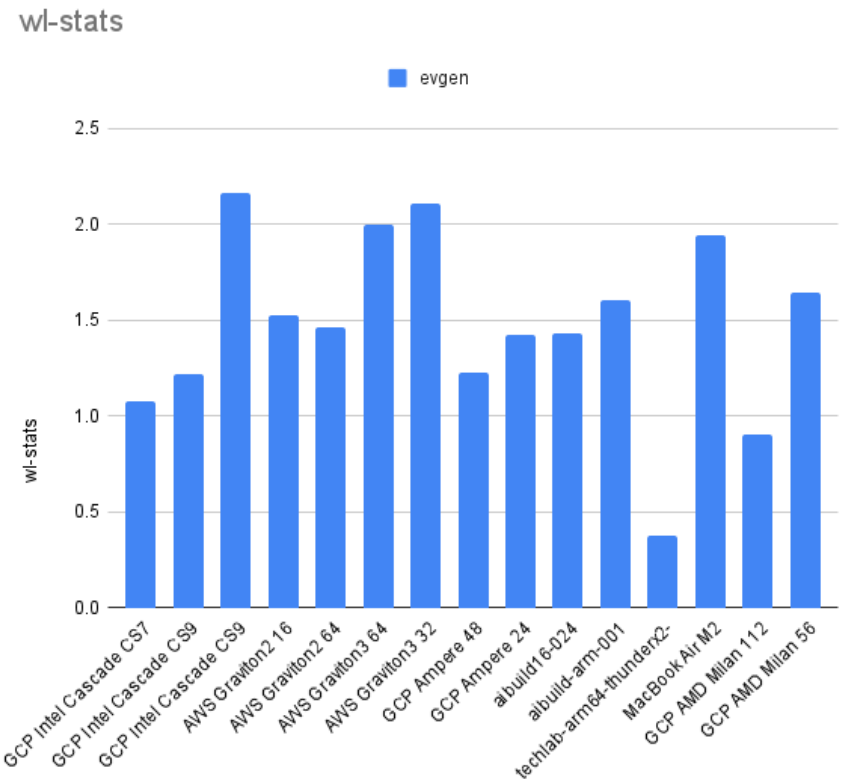
- Walltime/event: 0.92
- CPU time/event: 0.81
- HS06sec/event: 0.79 (with ARM HS06=10)

HepScore integration - reconstruction scores, single workflow



HepScore integration - Event generation results





HepScore integration - Simulation results