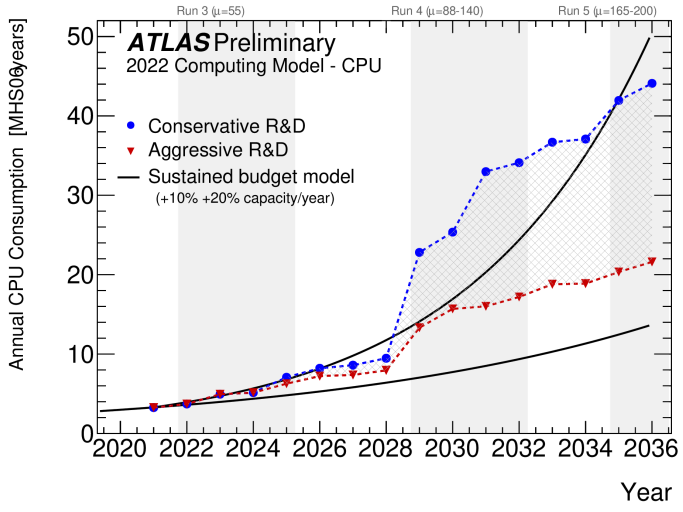
Using the ATLAS experiment software on heterogeneous resources

Johannes Elmsheuser (BNL) on behalf of the ATLAS Collaboration

23 October 2024, CHEP 2024, Kraków, Poland

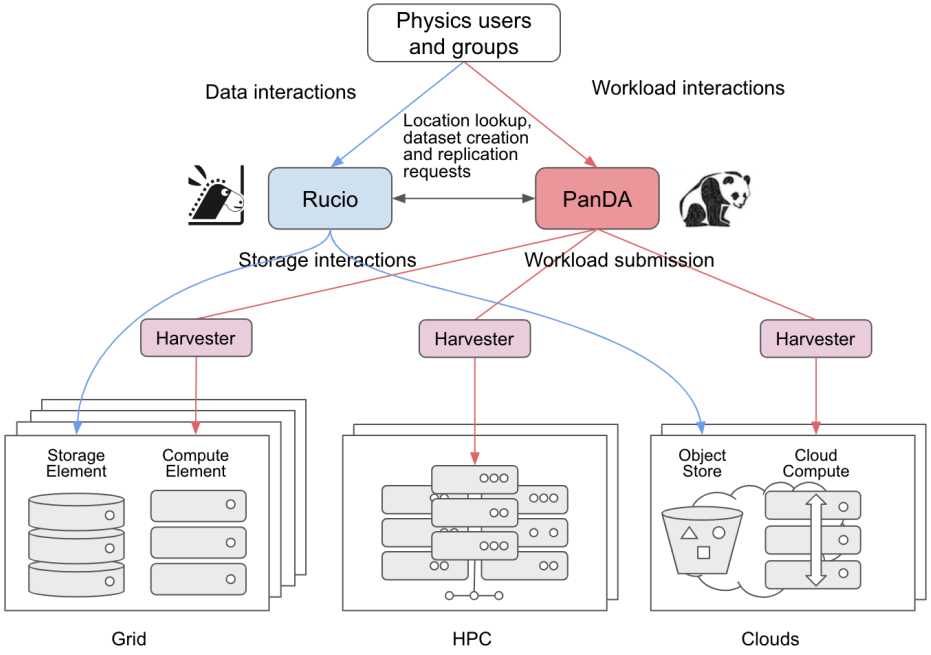
Introduction - ATLAS HL-LHC resource projections

- HL-LHC with increased luminosity, event size and event rates
- Flat computing budgets require R&D efforts to close the projected resource gaps
- CPU projections include no assumptions about GPU usage



- In this presentation, discussion of closing resource gap:
  - ARM CPUs: extra and power efficient resources
  - Cloud computing
  - GPUs
- Note: *All accessible with ATLAS full software stack through the ATLAS workflow management system PanDA*

ATLAS Grid setup: PanDA and Rucio



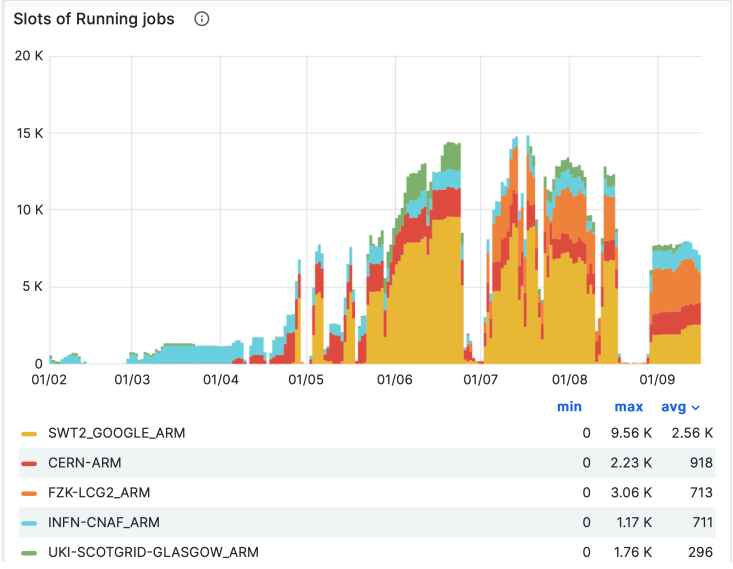
- More details in arxiv:2403.15873
- Dedicated ARM and GPU PanDA queues configured in Computing Resource Information Catalogue (CRIC)
- Full ARM/aarch64 grid setup available with OS container, middleware, Kubernetes etc.
- For NVIDIA GPUs need matching CUDA linux kernel module version and redistributable CUDA libraries

- Pilot Job configuration:
  - Dedicated ARM queue on the CE or using "WantARM=True" in PanDA pilot job jdl
  - Dedicated GPU queue on the CE or using "+RequireGPUs = True" and "+RequestGPUs = 1" in PanDA pilot job jdl
- ATLAS user job submission:
  - Use following options for PanDA job submission tools (prun/pathena): *–architecture "&nvidia-*"* or *–architecture "@el9#aarch64"*

Usage of ARM in ATLAS

Last CHEP23 presented "The ATLAS experiment software on ARM" (link) - much has happened since then

- 7 nightly builds with gcc13 for development and production branch and different projects of ATLAS software stack Athena, which are built on 4 build machines provided by CERN IT (Ampere Altra/Neoverse-N1) - nightly and stable releases automatically installed on CVMFS
- Running MC simulation and reconstruction on 5 PanDA queues with up to 15k concurrent job slots
- Configured PanDA queue as extension of US Tier2 in UT Arlington in *Google Cloud* with up to 9.5k job slots



ATLAS is the first WLCG experiment which will accept ARM resources as pledge in 2025/26

ARM/aarch64 ATLAS builds and caveats

- Build flags
  - Using Armv8 defaults (gcc 13.1 allows up to armv9.3-a, gcc docu link)
  - Test builds of Athena/AthenaExternals with clang17 work as well
  - Speed up of Geant4 simulation by 2-3% when using:
    - *CXXFLAGS="-march=armv9.2-a -mtune=neoverse-v2"* (NVIDIA Grace)
    - *CXXFLAGS="-march=armv8.3-a -mtune=neoverse-n1"* (Ampere Altra)
    - But code not necessarily portable anymore to all ARM processor versions
- Potential numerical differences
  - Due to different math and/or run time libraries used - see StackOverflow link
  - Some fluctuations in physics objects at the level of $(10^{-4} - 10^{-6})$

ATLAS software on Apple silicon

Lima

- ATLAS nightlies and stable releases can easily be used for development/execution on Apple Silicon (documentation) using Lima (link)
- What is ”Lima” ?
  - Linux Machines, ”Lima launches Linux virtual machines with automatic file sharing and port forwarding (similar to WSL2).”
- ATLAS provides automated instructions for container/VM with AlmaLinux 9.4 + HepOSlibs + CMVFS inside the container
  → All ATLAS code accessible via CVMFS and useable with VSCode on user laptop
  → See HepScore23 benchmark on later slide
  → Documentation how to reproduce for other experiments at link

Using GPUs in ATLAS - introduction and caveats

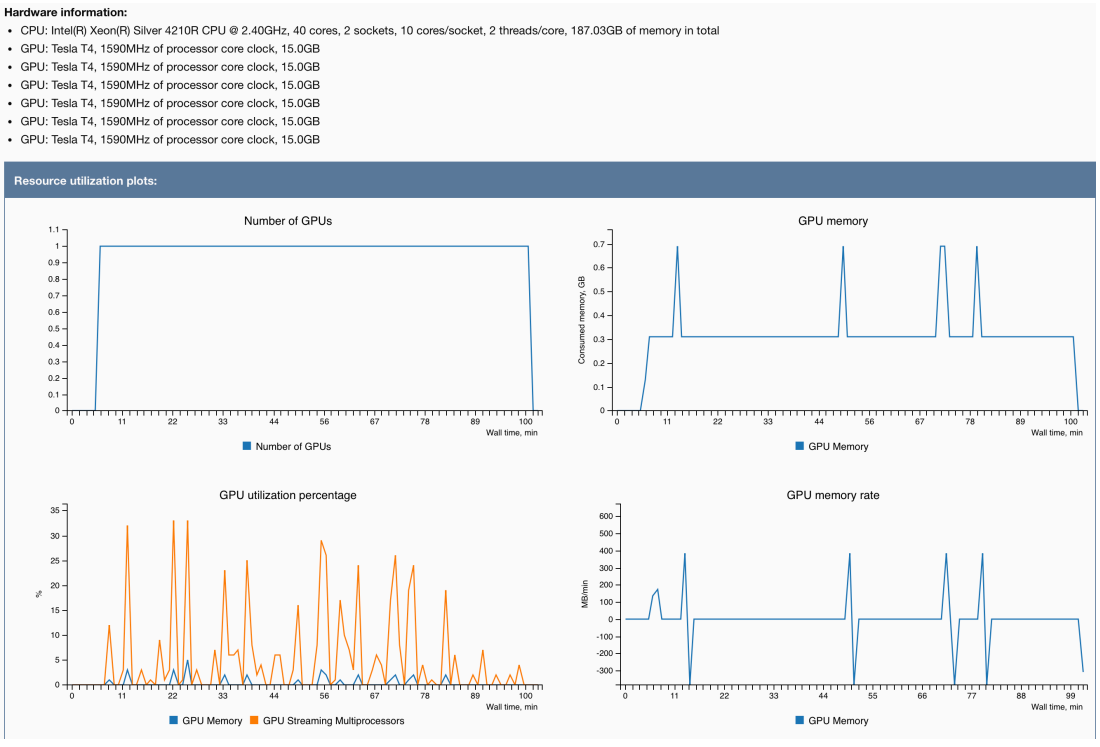
- ATLAS is not using GPUs in production for Run3
- Major R&D effort on-going to port parts of simulation, reconstruction and high level trigger code to use GPUs for HL-LHC (see several talks in other sessions at this conference)
- Since March 2024, Athena main branch uses CUDA 12.4 (and later 12.4.1) - this version of CUDA supports gcc13 (current production compiler version in ATLAS)
- Parts of CUDA SDK are redistributable (see file list at link) and usage works fine also outside of CERN via CVMFS and PanDA
- N.B. ATLAS HLT group successfully ran fully automated offline reprocessing of Calorimeter topo clustering algorithm on GPUs via PanDA
- **But:** every PanDA GPU queue/site has to have at least the kernel driver version 550.54.15 (or newer) from CUDA 12.4.1 installed
  → Tedious process in reaching out to sites and asking for CUDA kernel driver version update
  → Same process repeats potentially when moving to a new major CUDA version
  → A (automated) procedure should be discussed within WLCG when GPUs are more commonly used on the Grid

More detailed overview of PanDA GPU queues (Status September 2024)

PanDA queue name	GPU type	GPU on node	vCPUs on node	Driver	CUDA	Works with Athena ?
ANALY_BNL_GPU_ARC	A100				12.2	✘
ANALY_INFN-T1_GPU	Tesla K40m	1	1	460.106.00	11.2	old GPU
ANALY_MANC_GPU	Tesla T4	4	1	560.35.03	12.6	✔
ANALY_OU_OSCER_GPU_TEST	K20	1	1		11.x	old GPU
ANALY_QMUL_GPU	A100 40GB	1	1	550.54.15	12.4	(✔)
ANALY_SLAC_GPU	A100 40GB	1	1	535.161.07	12.2	✘
FZK-LCG2_GPU	V100S 32GB	8	8	555.42.02	12.5	✔
NERSC_Perlmutter_GPU	A100		8		12.2	✘
UKI-LT2-QMUL_GPU	A100 40GB	1	8	550.54.15	12.4	(✔)
UKI-NORTHGRID-MAN-HEP_GPU	Tesla T4	4	8	560.35.03	12.6	✔
lxplus-gpu.cern	Tesla T4	1	28	550.90.07	12.4	✔

- Scanning CUDA information in simple PanDA jobs with ”nvidia-smi”
- GPUs at 2 sites are too old for CUDA 12.4 update
- Require newer CUDA kernel driver version and/or access via grid CE: BNL, SLAC and NERSC
- #GPUs and #CPUs:
  - Right now Athena workflows are foreseen to use 1 GPU
  - Selecting GPU device possible via `CUDA_VISIBLE_DEVICES` (see e.g. test_trf_athexcuda.sh)
  - Potential future options to explore: GPU device sharing, whole node scheduling

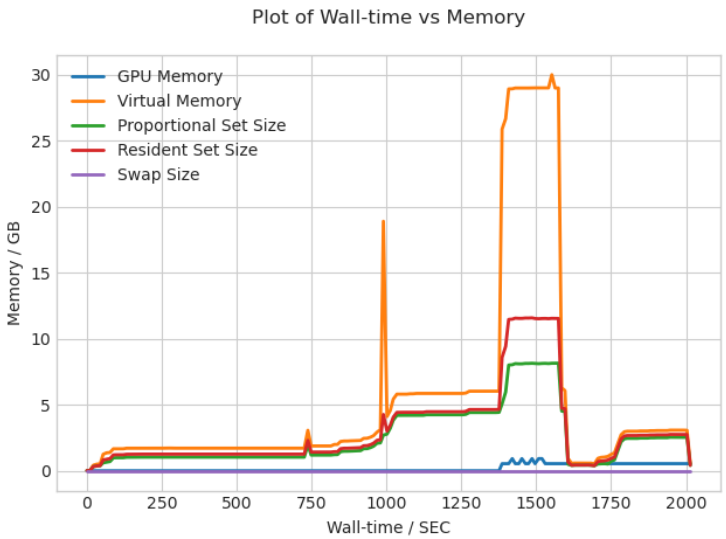
PRMON NVIDIAMON UPDATES



- ATLAS makes extensive use of the HSF/ATLAS tool **prmon** (link) to monitor payload resource usage in PanDA
- Example of an Athena HLT reprocessing test job on PanDA at Manchester

- prmon parses text output of nvidia-smi to collect GPUs resources usage and required some recent update
- Reasonable information collection for 1 or 4 GPUs but not for 8 GPUs
- Resource collection only available for NVIDIA - more robust implementation via C-API possible and add support for GPU vendors
  → ideal student project - contact the prmon authors !

Using Athena on NVIDIA Grace Hopper



prmon plot of memory usage over time for ATLAS HLT reconstruction workflow on Grace Hopper (72 core Arm Neoverse v2 + GH200 GPU)

- Athena HLT reconstruction code ported to GPUs benchmarked on NIVIDA Grace Hopper testbed provided through LBNL
- GPU workflows run out-of-the box on this ARM CPU+GPU testbed (but slower due to missing frontier/squid in this testbed)
- Reliable GPU benchmark needed in future HepScore version - so far only CPUs:

Name	nCPU	HepScore23	HepScore23 per nCPU
HepScore23 reference	64	1018	15.9
Grace Hopper	72	2319	32.2
Apple M2 Air	8	141.4	17.7
Ampere Neoverse-N1	20	349.4	17.5
Intel Xeon E5-2683 v4	16	258.5	16.2

Summary and Conclusions

- Summary:
  - Access and support of heterogenous resources will help to address the HL-LHC computing resource challenge
  - ARM is used in ATLAS for MC simulation and reconstruction production and will accept ARM resources as WLCG pledge in 2025/26
  - ATLAS ARM software nightlies and releases on Apple Silicon through Lima containers
  - Extensive R&D program on-going to port workflows to GPUs for HL-LHC
  - Athena workflows with CUDA supported through PanDA on WLCG
- Wishlist:
  - Ease CUDA kernel driver update on WLCG sites
  - GPU benchmark in HepScore