A step towards GPU benchmarking

GridPP 53

Albert (Bruno) Borbely, David Britton, Emanuele Simili, Gordon Stewart, Samuel Skipsey

On Grid GPUs

Chicken or Egg?

- What GPUs do we have on the Grid?
- Do we invest in £££ of hardware without anything to run?
- Do we invest lots of effort re-writing software without any hardware to run it on?
- We have some GPUs available on the Grid, tends to be a GPU node that sees sporadic usage
- I ran some scouting jobs via panda (prun) to see what GPUs are available for ATLAS CMS may have more (I don't have access)

- Found 7 sites: SLAC, Manchester, Forschungszentrum Karlsruhe (FZK), CERN, BNL, QMU and Glasgow
- Found a variety of GPUs all Nvidia, Primarily used for dev work / testing:
  - 8x A100 40 / 80 GB
  - 30x Tesla T4 16 GB
  - 2x Tesla V100 32 GB
  - 5x H100 94 GB

GPUs are coming

Both software and hardware

- ATLAS CPU budget ~ 40% on detector simulation (in 2018)
  - ~ 20 % are “Full-Simulation” Geant4 jobs
- Makes sense to target this large fraction of CPU expenditure
- Geant4 is also widely used by other Nuclear/ Particle Physics collaborations
  - So could be utilised by the wider WLCG/ Scientific community

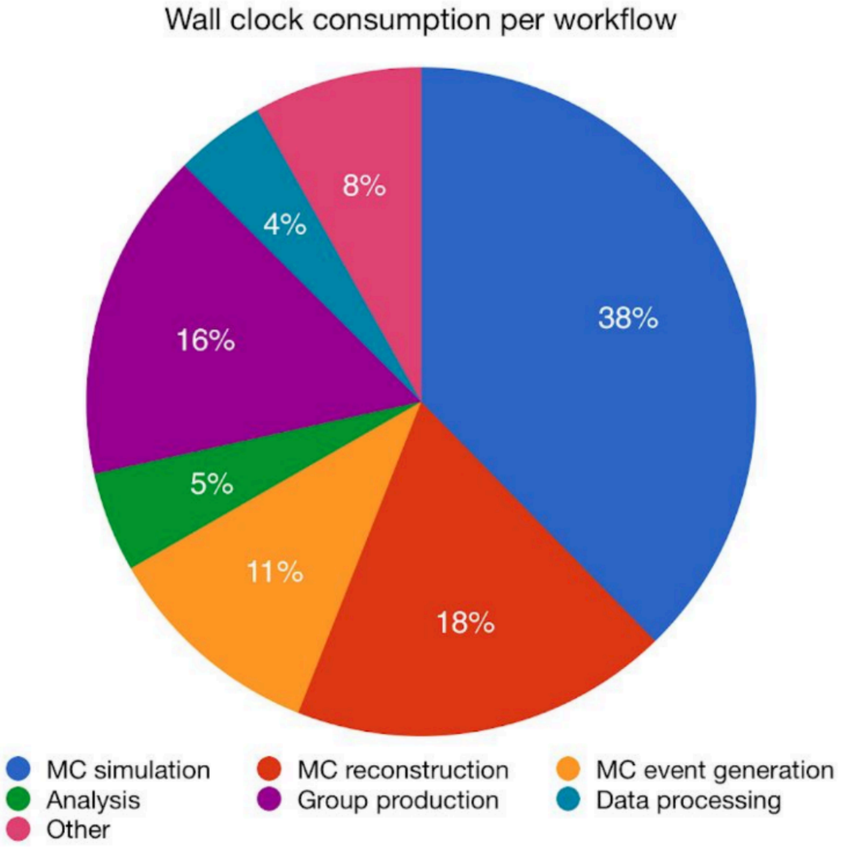
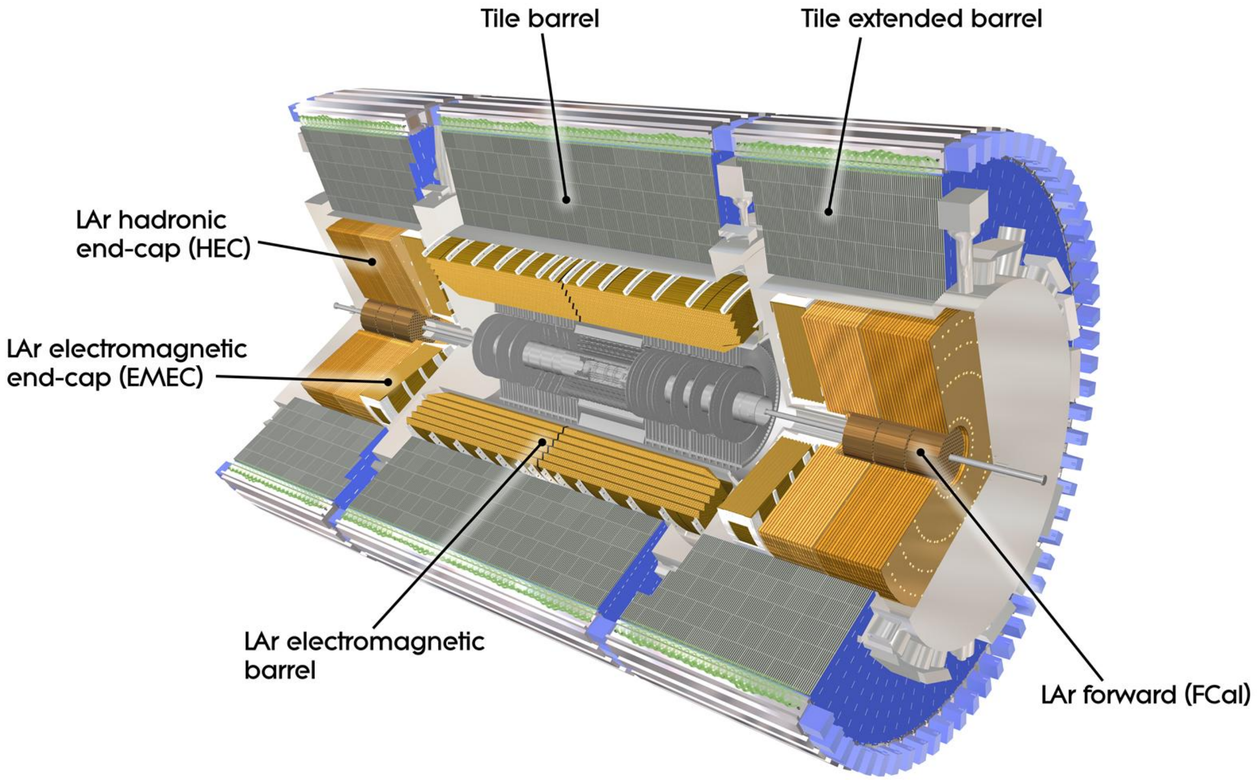


Figure 1: ATLAS CPU hours used by various activities in 2018

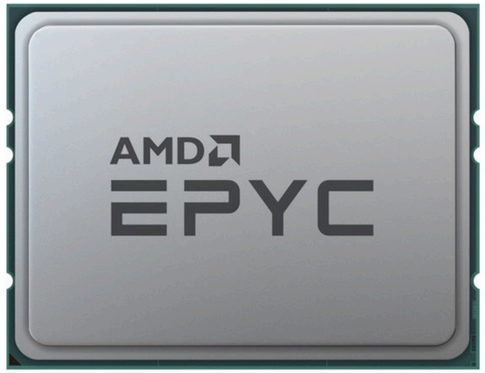
Celeritas Project

Using Celeritas ATLAS Tile calorimeter test run

- The Celeritas project is aimed at developing GPU-based Monte Carlo simulations in HEP
- Currently focused on EM physics e.g. the ATLAS Tile calorimeter
- I got in touch with the Celeritas team with the goal of setting up a benchmark
- Work is still on going and results are preliminary
- Primarily implanted for CUDA but ROCm support is in development
  - Some standalone tests are available, ATLAS geometry not yet supported



Test system



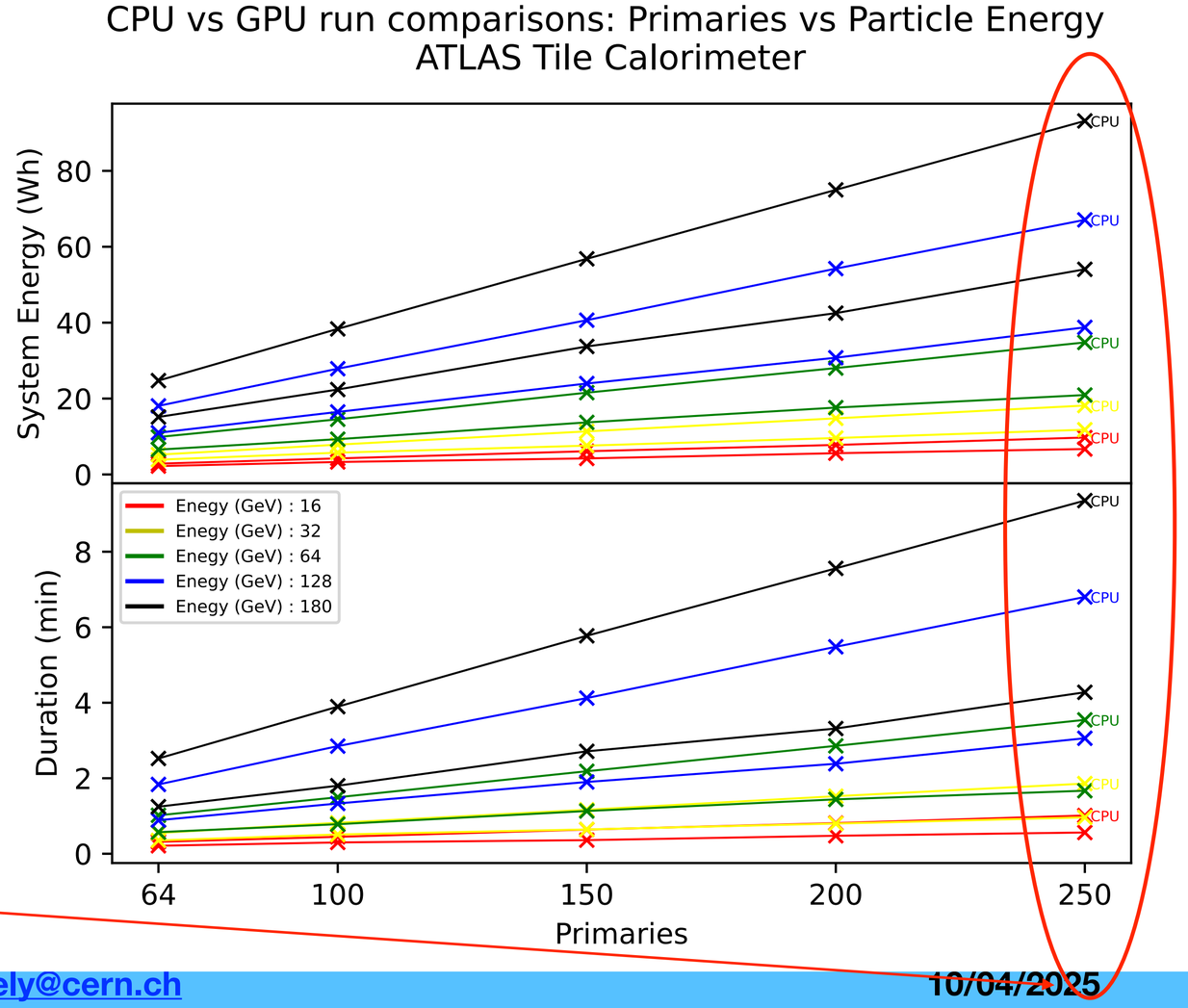
- CPU 2x AMD EPYC 7443 x48 cores (2021, x86)
- GPU 2x a100 (80GB)
- RAM 256 GB

- Most software is being written for CUDA
- There is a push to not be locked into CUDA
  So ROCm support / ALPACA
- It would be interesting to also test on AMD GPUs

GPU benchmarking: previously

CPU vs GPU comparison

- Using the ATLAS Tile Calorimeter as a test geometry
- Using Celeritas in GPU and CPU mode
- 2 run parameters were varied:
  - Number of primaries (Pions)
  - Initial particle energy
- The higher the parameters
  —> more intensive job
  —> more work offloaded to GPU
  —> greater reduction in duration/energy
- @ lowest (N64 & E16 GeV) ~ 22% & 33% decrease in job en
- @ highest (N250 & E180 GeV) ~ 42% & 54% decrease in job
- This was showing lots of potential
  Scales linearly with number of particles and energy

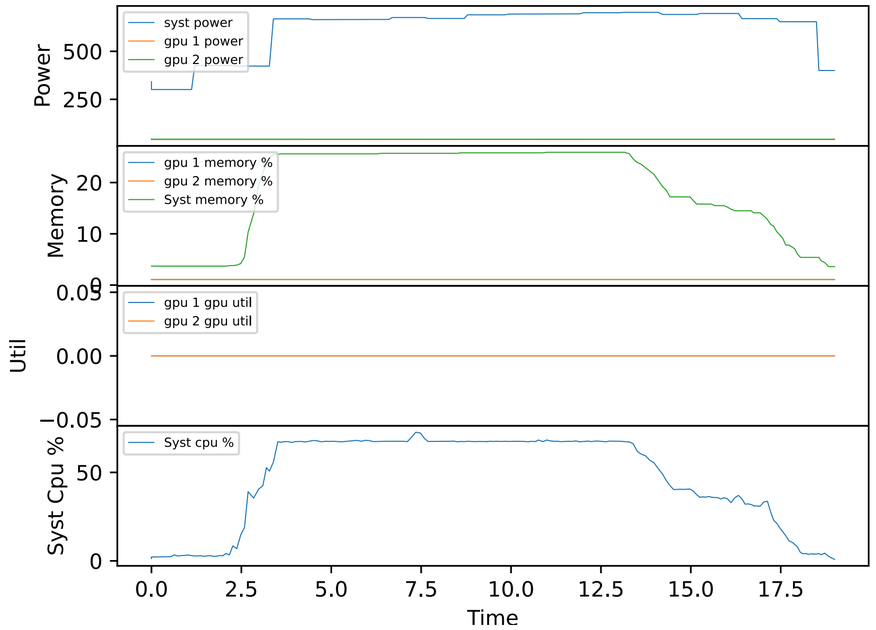


GPU benchmarking: prev.

Details

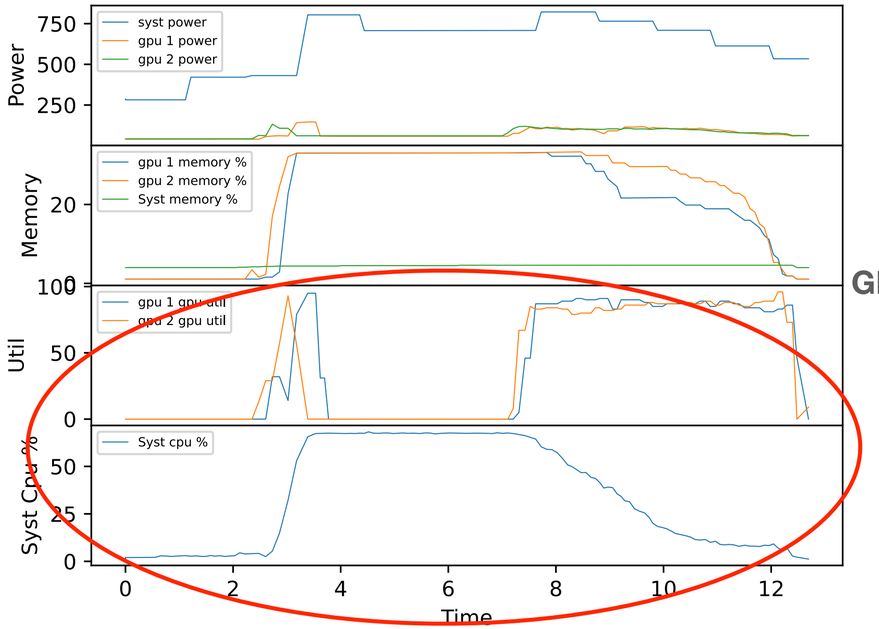
- CPU threads set to 32
- System Power gathered with IPMI
- GPU Power gathered with NVML
  - —> Some differences with sampling…
- CPU Energy ~ Syst. energy - GPU energy (Verified)
- To keep things consistent 2 cpu jobs were launched in parallel as well as 2 gpu jobs (one targeting each card), so values shown are for two jobs in parallel in both cases
- GPU variant doesn't always maximise GPU utilisation
- Large job initialisation, the more events the smaller the fraction

(Wh) syst: energy 3.28; gpu 1: energy 0.21; gpu 2: energy 0.22;
Duration: 19.0 s



CPU

(Wh) syst: energy 2.21; gpu 1: energy 0.26; gpu 2: energy 0.26;
Duration: 12.69 s

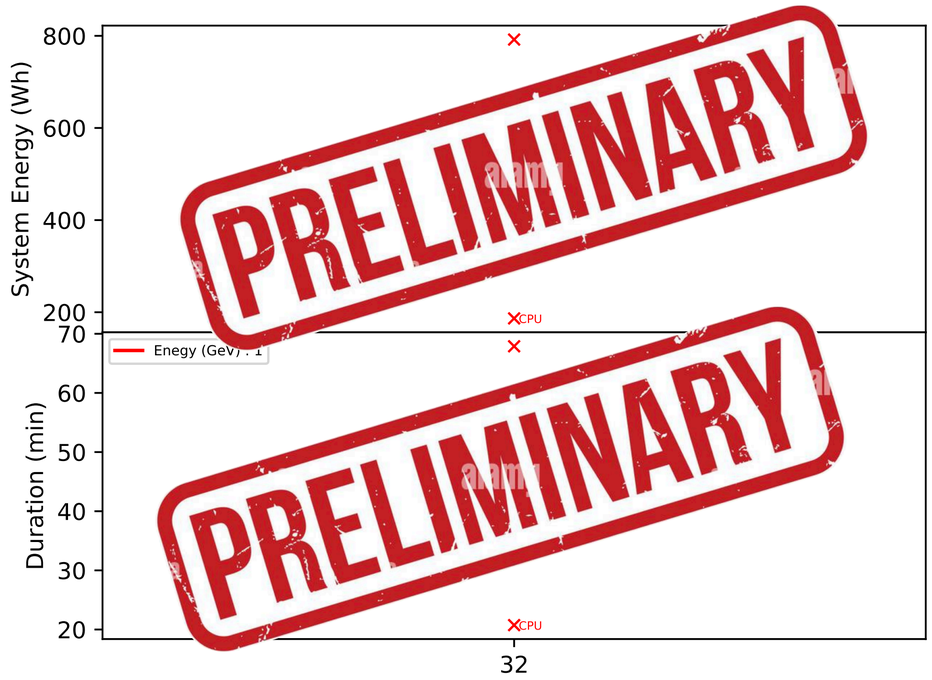


GPU

GPU benchmarking: next step

Full ATLAS detector simulation

- There has been allot of work on-going from both the Celeritas and AdePT projects, including analysis framework integration (Athena) and full detector geometry description
- The full ATLAS geometry can now be simulated, not just the Tile calorimeter
- Full tt bar events are now being simulated (generated with Pythia) at 13 TeV
  - Gives a more realistic test
  - Allows for event validation
    - It wasn't so easy to port to ARM CPU architecture,
      NOT all VOs managed
    - Currently only ATLAS is successfully running production ARM CPU jobs
  - GPU to accelerate the Tile calorimeter simulation part
- GPU utilisation ~100%, but only using (120 W / 300 W)
- GPU utilisation not a perfect metric
  —> all SM in use
  —> but not all of the SM is used (maybe tensor cores)
  —> new GPUs are now optimising for less precision, not FP64
- Only using ~ 30GB / 80GB of memory
- It also allows for job optimisation, GPU Grid jobs have yet to be defined
  - Current ATLAS geometry description is inefficient, historically optimised for CPUs
- The two projects recently had a review: https://indico.cern.ch/event/1507476/
- More details available in the next talk by Ben Morgans: talk



Preliminary CPU vs GPU tt events using full ATLAS geometry
32 threads

GPU benchmarking: Future steps

On-going

- I only just managed to compile / run the full SimLight ATLAS detector simulation, haven't had time to digest things, few merge requests / fixes missing from my setup
  - —> Plan to scan number of CPU threads, aim to see how many CPUs needed to "feed" the GPU
- The Tile calorimeter only version is also available to compare the performance for the most optimal part
- Event generators are also being ported to GPUs (MadGraph)
  - —> plan to test as much of the simulation chain being offloaded to GPUs as possible
- Simulation is expected to be one part of the GPU benchmark, and should not just be defined by ATLAS. Need to get in-contact with CMS, LHCB and ALICE.
  - Rumours of CMS being able to accelerate ~60% of the software stack, yet be verified by me

GPU benchmarking: ML activities

Misc. activities

- ML activities have become widespread in WLCG collaborations
- Recent Physics Nobel prize also awarded for “enabling machine learning with artificial neural networks”
- ML models currently fit on one card, this is expected to change in the near future as models get bigger and inference becomes more expensive
  - Currently inference uses CPUs via ONNX, experimental GPU inference servers could be tested
- Grid is obvious choice for widespread GPU access, currently various groups have widely different GPU access —> leads to certain people becoming disadvantaged
- ML “jobs” don’t fit nicely into the classic Grid jobs
- Plan to work with the ATLAS ML-Forum conveners to create a useable submission framework / tutorial
  - Major community push would be to have training data in root files (not HDF5) to leverage XrootD and Xcache
- ML training jobs also planned to be another part of the GPU benchmark, initially testing the ATLAS b-tagger GN2 training on ~ 250GB of training data