Power efficiency gains from GPU optimised workloads

Geant4 based detector simulations with Celeritas

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

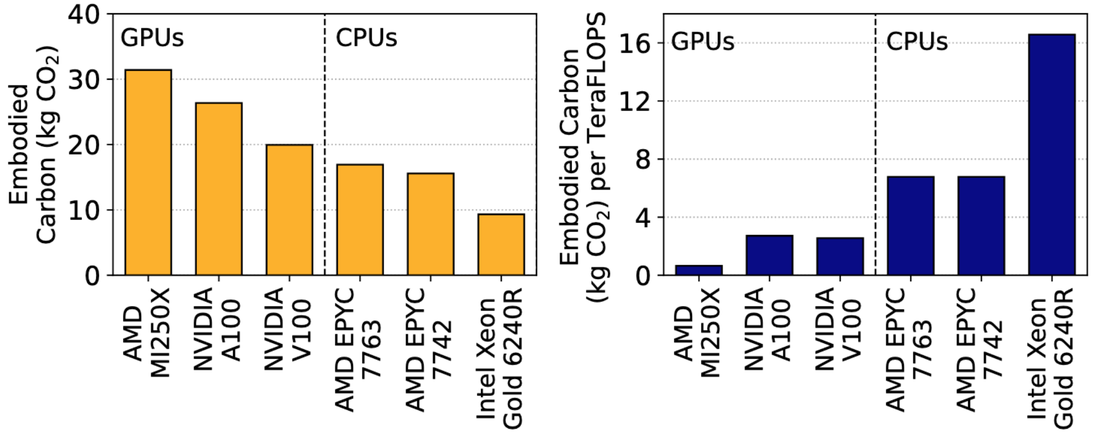
Processors CO2 life cycle

Embodied + Energy Fraction (EF) CO2

- Embodied CO2 due to manufacturing + shipping, fixed value for each component (may have a geographical component due to shipping)
- Energy fraction is the CO2 emitted by the local energy GRID and depends upon the length of its lifetime (KWh)

Embodied Carbon in Processor Components

- Embodied carbon in typical CPU / GPU normalised by double floating point precision (64FP)
- AMD CPUs have similar kg CO2 / TFLOPS between generations
  - AMD EPYC 64 core 7742 (2019), 7763 (2021)
  - Intel CPUs are uncompetitive
- Overall GPUs perform better when performance in normalised
  - Nvidia GPUs perform ~ 2x than AMD CPU
  - AMD GPUs perform ~ 6x than AMD CPU
  - Raw double floating point precision (64FP) isn't the full story, Nvidia focusing on 32, 16, 8 in recent years for ML

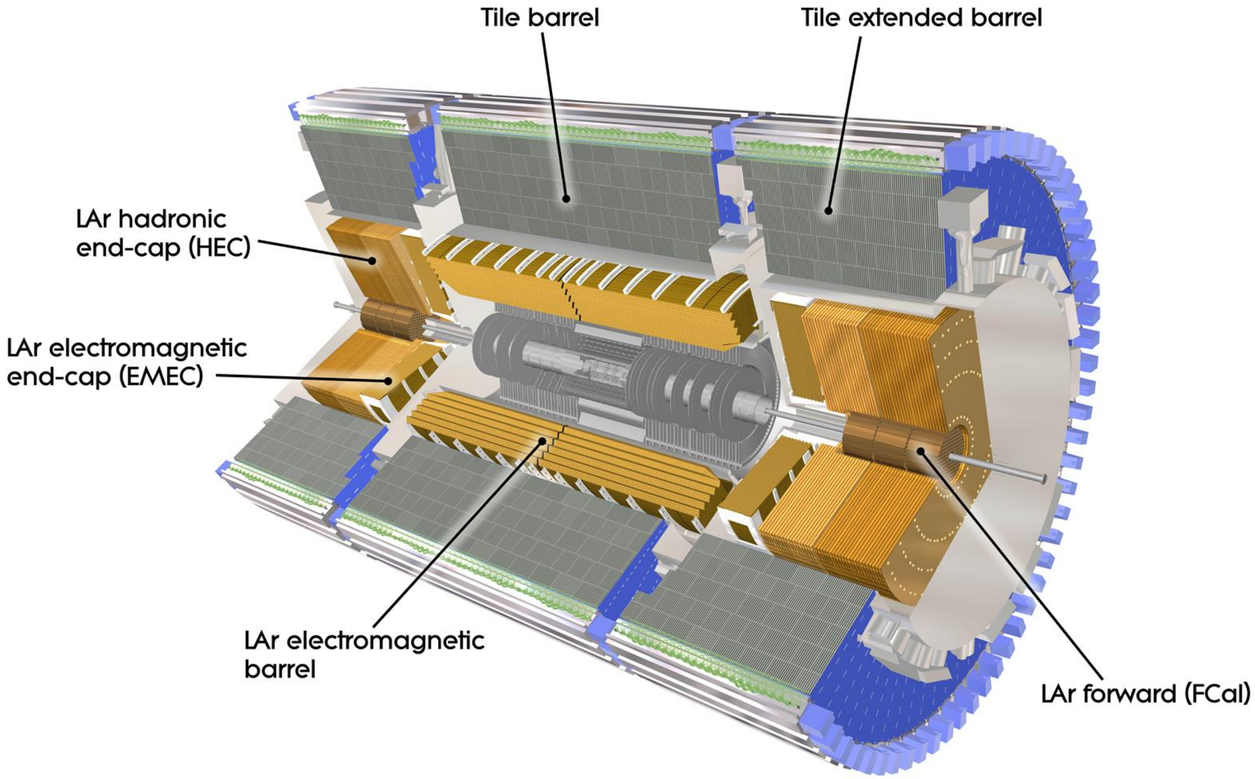


https://arxiv.org/pdf/2306.13177

A step towards GPU benchmarking

Using Celeritas ATLAS Tile calorimeter test run

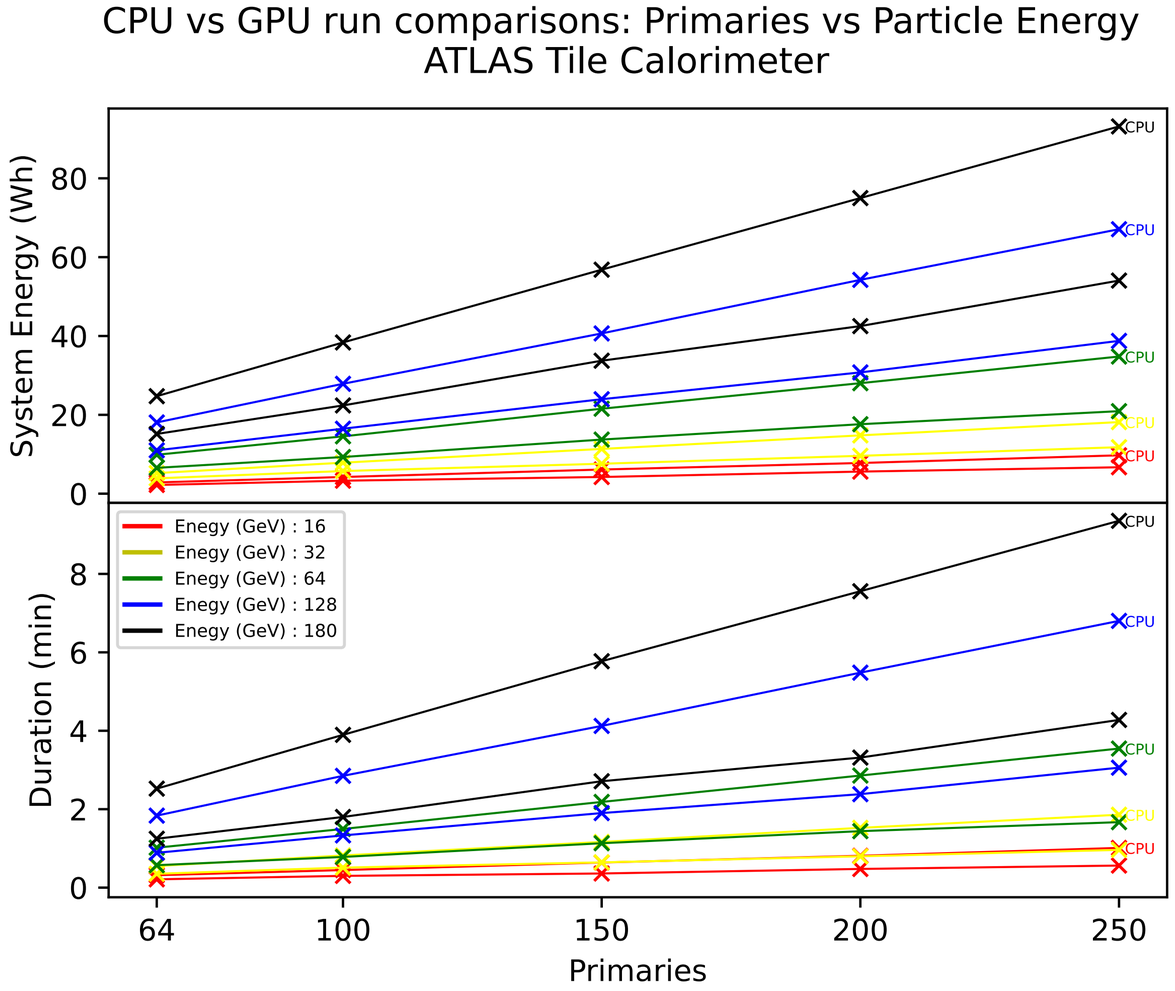
- The Celeritas project is aimed at developing GPU-based Monte Carlo simulations in HEP
- Currently focused on EM physics e.g. the ATLAS Tile calorimeter
- Detector simulation is particularly costly in terms of CPU
  - ATLAS spends ~ 40% of its CPU
  - 20% spent on Geant-4 based full detector simulation



GPU benchmarking: step 1

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
  - 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 energy & duration respectively with GPU
- @ highest (N250 & E180 GeV) ~ 42% & 54% decrease in job energy & duration respectively with GPU
- Plenty of gains to be had

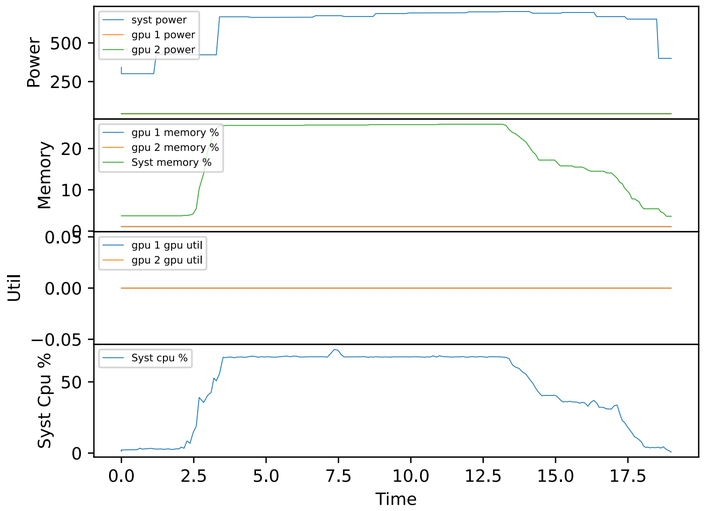


GPU benchmarking: step 2

Details

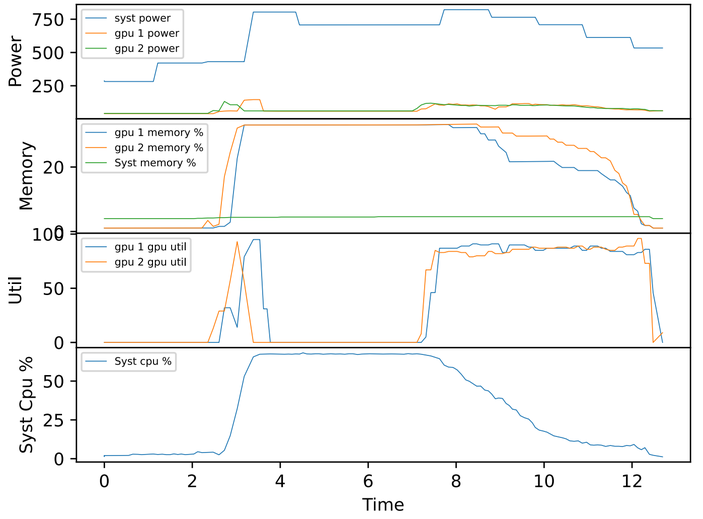
- CPU threads set to 32
- System Power gathered with IPMI
- GPU Power gathered with NVML
- CPU only Energy ~ Syst. energy - idle GPU energy
  - Verified with unplugged GPUs
- System: GPU 2xa100 (80GB),
  CPU 2x AMD EPYC 7443 x48 cores (2021),
  RAM 251 GB
- To keep things consistent 2 cpu jobs were launched 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 —> need to think about CPU/GPU —> to maximise GPU utilisation
- All jobs being launched in docker containers, to deal with dependancies, environment and installation. Also makes GPU management easier.

(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: step 3
Node considerations

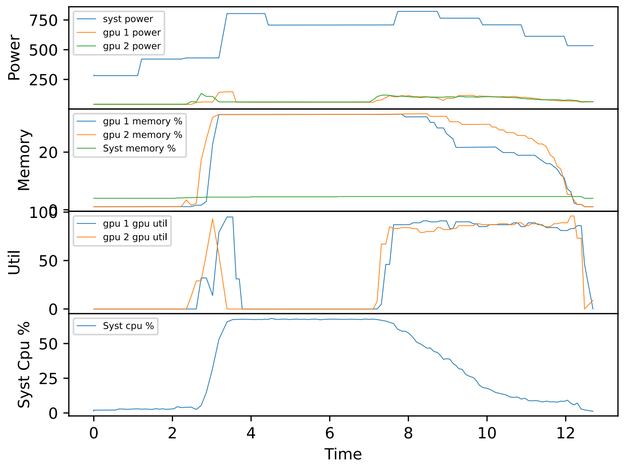
- Compute node now consists of GPU + CPU
- Nvidia a100 GPU 1.55x AMD EPYC 7763 CPU embodied carbon (EC)
- A compute node with: x2 AMD CPU, x2 Nvidia GPU ~ 5x EC
- Its normalised EC (kg CO2/ 64FP)
  is 70% (5) for the GPU version vs (7) for the CPU version
  - More compute in 1 node will leads to less EC / TFLOPS
  - Life cycle CO2 will also heavily depend on GPU utilisation
  - CPU is idle when offloading to the GPU
  - The above number assumes full CPU and GPU utilisation throughout lifecycle not necessarily realistic, still learning on how to utilise GPUs fully

GPU benchmarking: step4

Job considerations

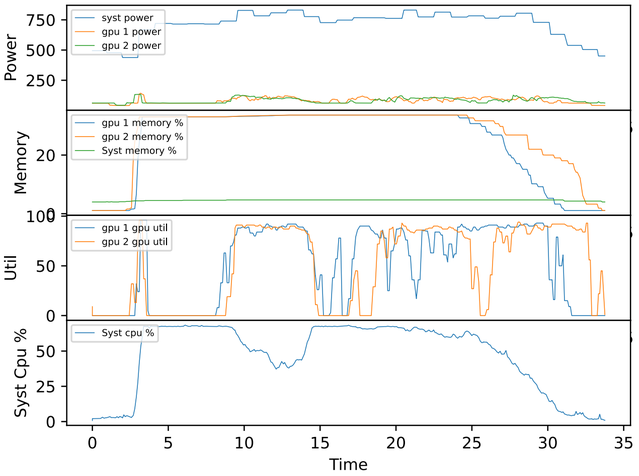
- Job “work” heavily depends on parameters used
- CPU variants effectively use a constant fraction of CPU resources (32 threads per job)
- GPU variant is constantly offloading parts of the job to the GPU
  - —> This causes fluctuations in GPU utilisation
    (see left/top-right plots)
- Need to think about how to maximise GPU utilisation, often most expensive part should not be sitting idle
- Current setup allows a job to hog 1 GPU
  - —> Potential solution to allow multiple job slots to share a GPU resource

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



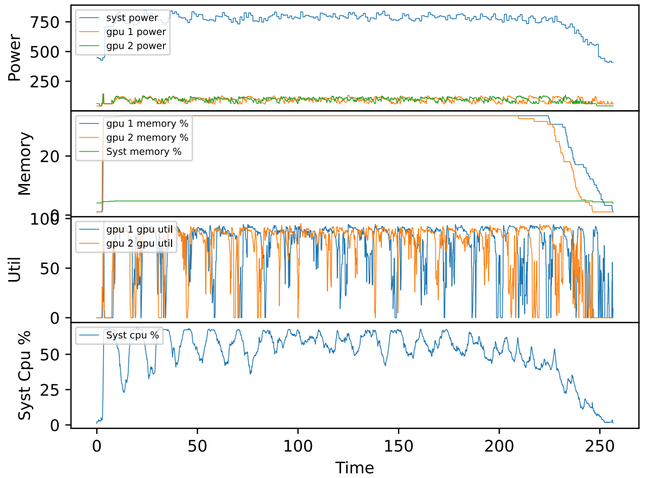
GPU
E16
N64

(Wh) syst: energy 6.7; gpu 1: energy 0.76; gpu 2: energy 0.77;
Duration: 33.76 s



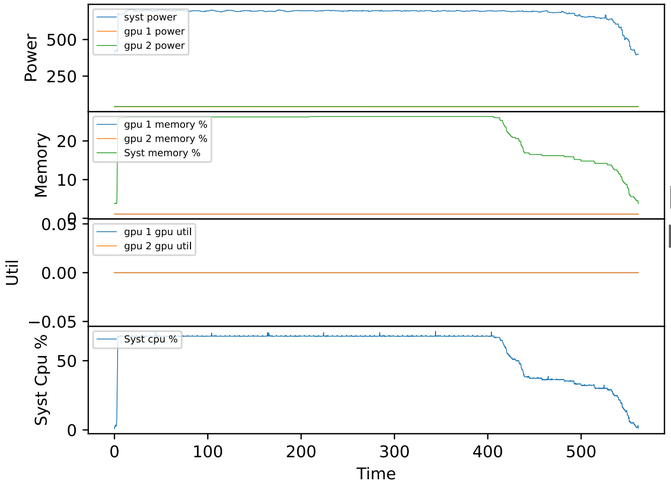
GPU
E16
N250

(Wh) syst: energy 54.06; gpu 1: energy 6.61; gpu 2: energy 6.71;
Duration: 256.59 s



GPU
E180
N250

(Wh) syst: energy 106.11; gpu 1: energy 6.34; gpu 2: energy 6.61;
Duration: 561.23 s



CPU
E180
N250

GPU benchmarking: step 5

Putting things together

- Node + Job consideration
- The CPU + GPU version of a node has ~ 2.5x the EC of the corresponding CPU node
- With a normalised EC of 0.7x corresponding CPU node in terms of double floating point precision
  —> assumes 100% utilisation of both CPU and GPU resources
- With the best possible job run conditions the Celeritas benchmark achieved a 42% decrease in real world energy use
  —> does not fully utilise CPU or GPU —> utilisation fraction dependant on run parameters
  —> the more work offloaded to the GPU the greater the energy savings
- Break even point will depend on where node is geographically located
  - —> lots of dirty power —> EC is a negligible fraction,
    CO_2 due to energy mix will dominate —> frequent component upgrade cycle
  - —> lots of clean power —> EC is a dominant fraction —> longer running time required to break even —> infrequent component upgrade cycle
- Component utilisation rates also play a factor:
  - If a resource is under utilised then a negative factor appears in-terms of idle energy
  - This is not really an issue with CPU only as the job can maximise the CPUs capabilities
  - But with CPU + GPU architectures one of the resources is by definition waiting for the other
- Overall good progress is shown in achieving energy savings in calorimeter simulation using FullSim Geant-4 when high calorimeter fidelity is required