GPU production workloads in WLCG?

Andrea Valassi (CERN IT-DI)

Benchmarking Pre-GDB, 8 October 2019 – CERN
https://indico.cern.ch/event/739897

Disclaimers!

- I am not an expert in GPU software development or execution
  - Please excuse any factual mistake or misconception in what I discuss
- I did not have the time to contact many relevant experts… apologies ☹
  - Also because we were a bit late in organizing this session
- I will mainly report from the conclusion of a cross-experiment HPC meeting in May
  - Including ATLAS, CMS and LHCb feedback on software and computing for HPCs
  - Complementing this with my current understanding of other ongoing activities on GPUs
- This talk is mainly meant to stir up thoughts and introduce the following talks

Why discuss GPUs for WLCG: HPCs

"Summit, the new TOP500 champ, […] is powered by IBM servers, each one equipped with two Power9 CPUs and six V100 GPUs. According to NVIDIA, ***95% of Summit's peak performance*** *(187.7 petaflops) is derived from the system's 27,686 GPUs."*

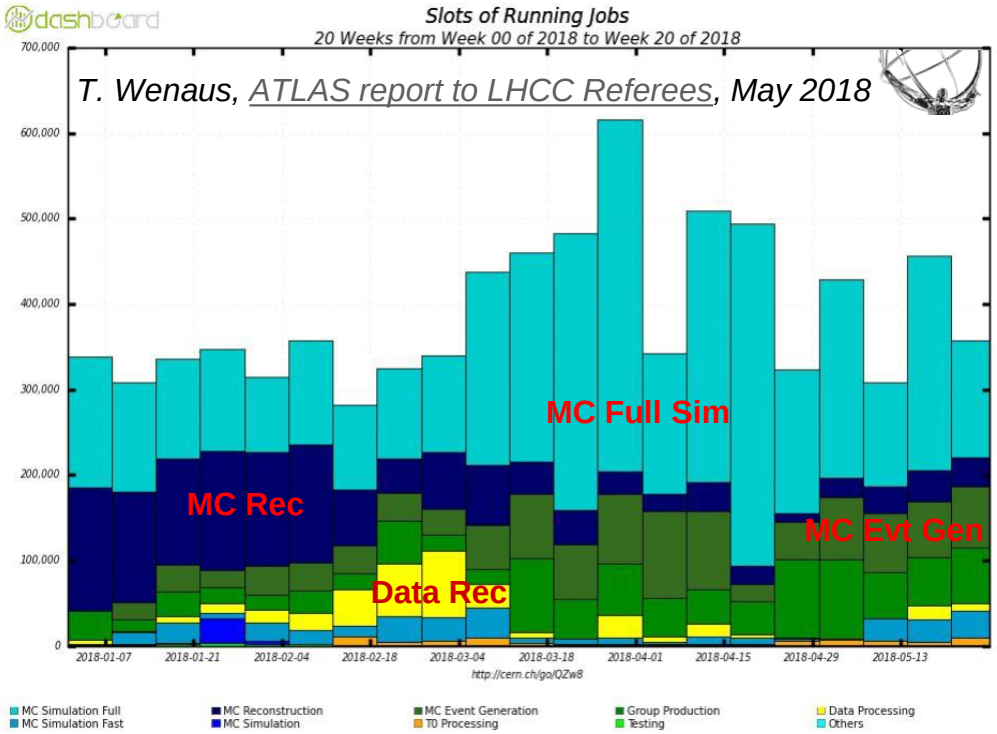
https://www.top500.org/news/new-gpu-accelerated-supercomputers-change-the-balance-of-power-on-the-top500

By Carlos Jones/ORNL,CC BY 2.0

- In the latest generation of HPCs, almost all computing power comes from GPUs
  - Being able to exploit HPCs effectively in WLCG passes through porting software to GPUs

(Aside: and also through porting it to Power9, ARM…)

Software workloads: who are the big players in WLCG?...



MC Full Simulation (Geant)

Approx. ~25% to ~90% in the four experiments

MC and Data Reconstruction

Approx. ~30% for ATLAS and CMS

MC Event Generation

Approx. ~10% for ATLAS, less for others

In addition: Derivation, Fast Simulation, Analysis…

… and what is their status on GPUs?

Outline of the following slides

- MC detector simulation (full and fast)
- MC event generation
- User analysis
- MC and data event reconstruction

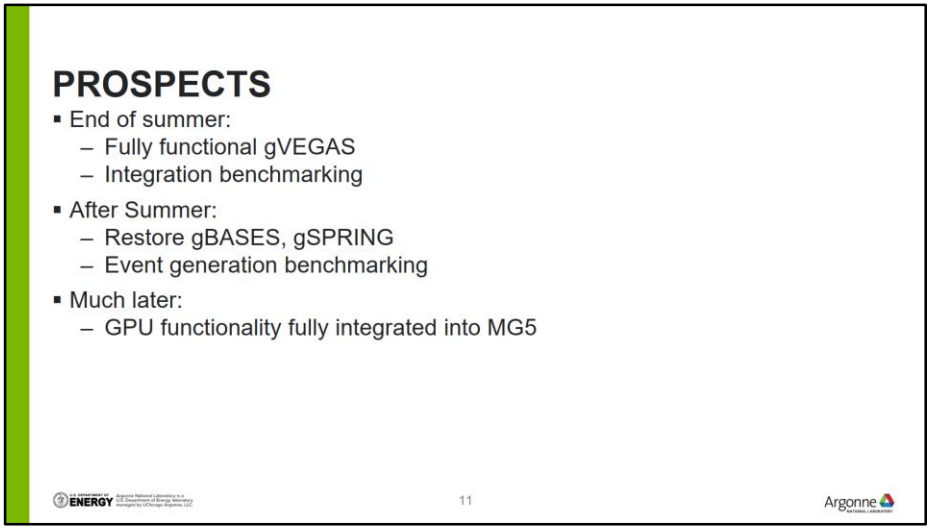
MC detector simulation

- **Full detector simulation** is overall the largest consumer of CPU in WLCG
  - *Being able to run Geant4 on GPUs (IMO, even inefficiently) would help us enormously*
  - Past and ongoing R&D on vectorization and GPU port of Geant4
    - See talk by P. Canal at HSF/OSG/WLCG Workshop in March 2019
  - *But no clear indication that Geant4 will be production-quality on GPUs in the short term*

- **Fast detector simulation** is expected to complement full simulation more and more
  - Ongoing R&D on fast simulation with ML techniques (GAN) that can be ported to GPUs
  - *But no clear indication that this will be production-quality on GPUs in the short term*

MC event generation

- **Event generation** seems like an ideal match for porting to GPUs?
  - Computationally intensive with no input data
  - It is a non negligible fraction of overall CPU (~10% in ATLAS) and expected to rise
  - Past and ongoing R&D on GPU port of generators such as MadGraph
    - Past work (2008-2013): see talk by J. Kanzaki at HSF/OSG/WLCG Workshop in March 2019
    - New work (2019): see talk by K. Fielman at HSF Generator WG in July 2019
  - *But no clear indication that MadGraph will be production-quality on GPUs in the short term*



K. Fielman, HSF generator WG, 4 October 2019

User analysis

- **User analysis** is certainly an area where GPUs can be very useful
  - Training of Machine Learning and Deep Learning models
  - Maximum likelihood fits
  - *But fraction of needed compute resources is negligible with respect to other workloads*

MC and data event reconstruction

- **Event reconstruction** is where all experiments seem most advanced on GPUs
  - ALICE has a production-quality (IIUC) reconstruction for its online data processing in HLT
  - LHCb has an advanced prototype of HLT1 reconstruction, to be further reviewed internally
  - ATLAS is progressing on an heterogeneous framework to offload some algorithms
  - CMS has a functional heterogeneous framework, with several algorithms on GPUs

- Goals and GPU offload strategies differ in subtle and less subtle ways
  - LHCb has a standalone GPU application, all others have an heterogeneous framework
  - ALICE/LHCb focus primarily/exclusively on online reconstruction, unlike ATLAS/CMS

- *CMS event reconstruction may be IMO the first workload to reach GPUs on the Grid*
  - *We identified this workload as our first GPU candidate for the benchmarking suite*
  - *See the next two talks by F. Pantaleo and A. Sciaba*

Reconstruction on GPUs: LHCb (trigger)

The Allen framework

The *Allen* framework is a compact, scalable and modular framework, built for running the LHCb HLT1 on GPUs.

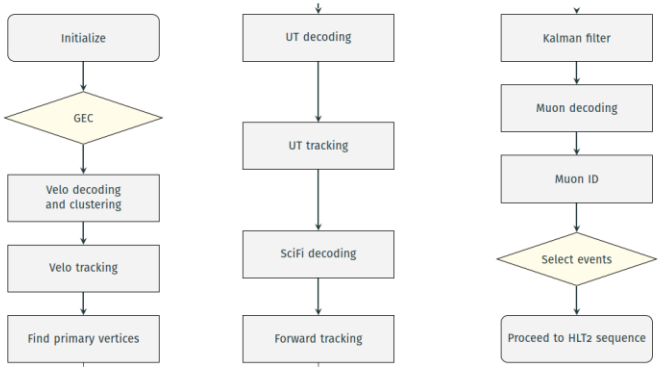
Requirements

- A C++17 compliant compiler
- CUDA v10.0

Features

- Multi-threaded framework with asynchronous GPU streams.
- Configurable static sequences.
- Pipelined stream sequence.
- Custom memory manager, no dynamic allocations.
- Support of MDF and custom binary input format to read in raw events.
- Built-in validation with Monte Carlo.
- Optional compilation with ROOT for generation of graphs.

Implemented sequence



R. Aaij, Openlab/E4 meeting, 4 October 2019

Fully on GPUs
Targets HLT only (no Grid)

Reconstruction on GPUs : ALICE

Summary

- ALICE will heavily employ GPUs during Run 3 online processing.
  - Mandatory for TPC tracking during synchronous phase.
  - Use for other detectors and during asynchronous phase where available.
- Baseline solution with TPC + part of ITS tracking nearly fully implemented.
- One order of magnitude speedup by new algorithm, one order of magnitude via GPU usage.
- Tuning should be switchable and in a general way – must run fast on CPU and GPU.
- Code written in generic C++ language, supports OpenMP, CUDA, OpenCL, HIP with common source code.
- Identical binaries used on GPU-servers and normal servers.
- No strict need for professional features like double precision, ECC, … / but no active cooling!
- Difficult to achieve binarily identical results on CPU / GPU:
  - Requires disabling of math optimization and usage of 64 bit weight functions → performance degradation.

Run 3 processing on GPUs

- **Which of the ALICE Run 3 computing workload can benefit from GPUs?**

Processing type	Location	GPU usage
Synchronous real-time reconstruction	O2 Farm	yes
Asynchronous raw-data reconstruction on the O2 farm.	O2 Farm	yes
Asynchronous raw-data reconstruction elsewhere	GRID	If possible
Monte-Carlo Reconstruction	GRID	If possible
Monte-Carlo Simulation	GRID	Not foreseen yet
Analysis	GRID	Not foreseen yet

Heterogeneous (IIUC)
Primary goal is trigger
May be used on the Grid

D. Rohr, HSF/OSG/WLCG Workshop, March 2019

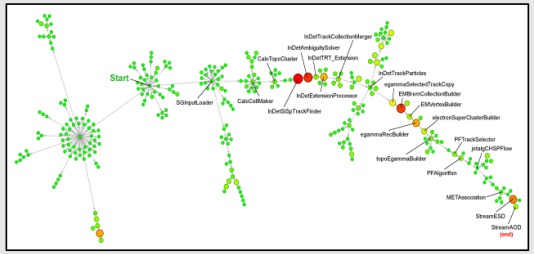
Details of GPU implementation

- **Software can run in 3 modes:**
  - Enforce GPU.
  - CPU-only.
  - Opportunistic GPU usage with automatic CPU fallback.
- **In any case, the same virtual interface class is used to drive the backend.**
- **O2 farm will enforce GPU usage.**
- **CPU only might be for some GRID sites, or when we encounter problems.**
- **Opportunistic usage is for your laptop, use whatever hardware is available.**

Reconstruction on GPUs : ATLAS

Looking Forward

- Major issues looking forward:
- Heterogeneous platforms
  - Identifying which tasks to offload to accelerator
  - Integration of offloading with framework
  - Portability solutions
- Vertical integration of C&C
  - Raythena



- Don't need to offload many Algorithms to see significant gains
  - offloading only 4 slowest Algorithms decreases total throughput by just 7% (*vs* all on critical path)

Heterogeneous Trigger and Grid (IIUC)

C. Legget, HSF Frameworks WG, 2 October 2019

Framework Integration of Offloading

- Our current model uses synchronous kernel execution
- Assumes
  - Offloading is a blocking task
  - CPU hardware thread is free to do other work while kernel executes
- So far, this has not proved to be correct
  - LOTS of communication between GPU and CPU when kernel executes
    - on NVidia (CUDA), Intel (SYCL)
    - AMD (hip) looks much better! (more testing needed)
  - NVidia claims they now understand the issue and says this will be fixed in the next compiler drop
  - Intel said "hunh. that's interesting"
- If this isn't fixable, we will have to change model to asynchronous execution
  - not insignificant refactoring of framework required

Reconstruction on GPUs : CMS

Why are we caring?

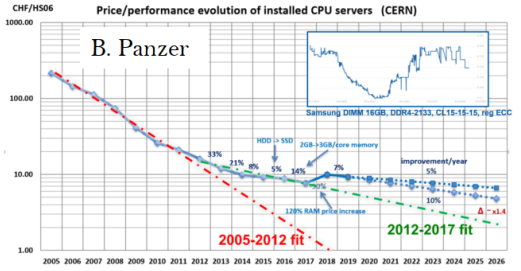
Heterogeneous Trigger and Grid (IIUC)

F. Pantaleo, EP-SFT Group Meeting, 7 October 2019

- The Patatrack team has demonstrated a complete CMS Pixel reconstruction running on GPU:
  - on a NVIDIA T4 can achieve 50% higher performance than a full Skylake Gold node
  - NVIDIA T4 costs approx. 1/5 of a node
  - It is fully integrated in CMSSW and supports standard validation
  - It is written in CUDA for the GPU part, C++ for the CPU part
- Maintaining and testing two codebases might not be the most sustainable solution in the medium/long term
  - Not a showstopper at the moment, but will become one when we will transfer ownership of the code to the collaboration
- In the long term other accelerators might appear

Why should our community care?

- Accelerators are becoming ubiquitous
  - Driven by more complex and deeper neural networks
  - Details hidden to the user by the FW
- Better Time-to-Solution, Energy-to-Solution, Cost-to-Solution



- Experiments are encouraged to run their software on Supercomputers
  - We are not using their GPUs
  - Summit: 190PFLOPS out of 200PFLOPS come from GPUs
- Training neural networks for production workflows is a negligible part
- Redesigning our algorithms and data structures to be well digested by a GPU can speed it up also when running on CPUs

Portability does not imply Performance Portability

Performance Portability frameworks

In the context of Patatrack R&D we have been recently looking into:

- Alpaka/Cupla: https://github.com/ComputationalRadiationPhysics/alpaka
  - Developed by Helmholtz-Zentrum Dresden – Rossendorf
  - Applications in Material science, XFEL, HPC
- Kokkos: https://github.com/kokkos/kokkos
  - Developed by Sandia National Lab, U.S. National Nuclear Security Administration

Conclusions

- Presently, GPUs are essentially not used in WLCG
- This needs to change, especially if we want to use GPU-based HPCs
- CMS reconstruction seems the most advanced GPU candidate workload for WLCG
  - See the next two talks
- Investment in software development/developers (not only on GPUs) is essential!