CMS Reco with GPUs

Viktor Khristenko (CERN), Maria Girone (CERN)
Outline

• DEEP-EST Project

• CMS Calorimeter Reconstruction with GPUs

• What can I do here for HEPiX
The DEEP-EST Project: Objectives

• In short -> Build Modular Supercomputing Architecture (MSA)

• Build a fully working, energy efficient prototype of the MSA

• Support HPC and HPDA convergence

• Extend a proven resource management and scheduling system to fully support the MSA

• Enhance and optimize the programming environment based on MPI and OpenMP. Add support for data analytics and machine learning frameworks

• Validate the full hardware / software stack with relevant HPC / HPDA applications
DEEP-EST: Goals and Motivation for HEP

- Explore conventional HEP workflows on HPC infrastructure
  - Experiment with ways to deliver software stack
  - Experiment with new architectures (arch/uarch)

- Explore (R&D) heterogenous options for data processing
  - CUDA / OpenCL / etc. devices
  - Potentially AMD HIP (replace cuda with hip and u have AMD HIP :) )
  - Etc..

- Explore large scale ML/DL training/inference with HPC resources
  - usability of Apache Spark for HEP Data Analytics with HPDA resources
CMS Ecal/Hcal Reco

- Working with patatrack group
  - [https://patatrack.web.cern.ch](https://patatrack.web.cern.ch)
  - Many different parts are being rewritten, Pixel is already in a good shape

- Take existing CPU ecal/hcal reco workload, and port towards usage with CUDA
  - Had to adapt Eigen lib to be applicable on the device
    - *Overall effort from patatrack since tracker also uses it*
  - Simple/naïve port
    - Think of Ecal data as AoS (or SoA, whatever you prefer) and convert your host side for loop into __kernel__ invocation…
    - *Not very robust, but works!!! And reproduces results 1-1 which is very important for HEP!*
  - Complete rewrite on top of patatrack release
    - *Patatrack release provides a bunch of things/utilities + _currently_ a simplified but efficient mechanism for conditions transfer to device mem*
    - Reverse engineer whatever is written for cpu
    - Overall, if computing all variables -> 10 kernels per stream. Different events are done in different cuda streams, currently.
    - *Right now only for ecal, needs further optimization. 1 kernel is a bottleneck currently*

- I had tests with opencl for Arria 10 for a portion of this reco, but this needs further work
CMS Ecal Reco Validation (GPU vs CPU)

Reconstructed Energy – good match is observed
What can I do for HEPiX?

• Was provided a clean vm with nvidia v100
  – Install Nvidia drivers + CUDA Toolkit
  – Verify nvidia samples + my samples are working
  – Pull cms patatrack release + merge my branch
  – Test/verify works/reproduces results 100%
  – Install nvidia-docker2 - Nvidia Container Runtime
  – Test/verify cuda stuff works from the running docker container

• End result
  – Given a node with nvidia drivers installed + docker-ce + nvidia-docker2
    ▪ *In all of this, I will assume the latest drivers/etc…. For the case of incompatibilities…*
  – Provide a docker file to build an image with
    ▪ Nvidia/Cuda Stuff
    ▪ *Cms patatrack release + setup to pull conditions from local cern proxies*
      – Patatrack release is not on cvmfs, I believe…
    ▪ *Encapsulate open data (a must, can not use regular cms data)*
    ▪ *Encapsulate cmsrun config to run ecal reco on a gpu*
  – And instructions to build and run the image
  – Will be a heavy image, but this is as far as I will/can go.
The DEEP projects have received funding from the European Union’s Seventh Framework Programme (FP7) for research, technological development and demonstration and the Horion2020 (H2020) funding framework under grant agreement no. FP7-ICT-287530 (DEEP), FP7-ICT-610476 (DEEP-ER) and H2020-FETHPC-754304 (DEEP-EST).