7th PataTrack Hackathon

Team: Multiple Architectures

Objectives:

Running CMS-Patatrack on multiple architectures. (CUDA, HIP(AMD), X86, ARM, IBM)

- Evaluating different ways (ALPAKA, KOKKOS, QT) on a standalone code (…)
  - ➢ Ease of management
  - ➢ Integration with CMSSW
  - ➢ Performance differences and checks.
- PataTrack Code conversion and performance checks.
- Pull request New Branch for moving the existing code.
- Adding guidelines in README / PataTrack Wiki

Hackathon: Day:1

- Access to the Machines
- Setting up CMSSW and Patatrack development branch with validation and tests
- Setting up Alpaka/Cupla/Kokkos/QT
- Trying out on standalone code - https://github.com/makortel/pixel-standalone
- Tests within CMSSW+Patatrack

Day:2

- Tried different examples with Alpaka/Cupla/Kokkos/QT and trying to understand further.
  - Ran the test code on different machines
    - Tests on AMD Readon7 card
    - Tests on V100 card.
    - Tests on x86

- Understanding CMSSW+Patatrack updates needed to add compilation flags and features of Alpaka/kokos using SCRAM

Day:3

- Tried converting some of the code from CMSSW-Patatrack using cupla and testing.

Day:4

Objectives:

- Running CMSSW-Patatrack on multiple architectures
- Evaluating different ways on a standalone code → Portability libraries as Alpaka, Cupla, Kokkos
  → Ease of management
  → Performance differences, if there are any

Achievements:

- Configuration of the GPUs machine
- Understanding the portability libraries cupla and Alpaka
- Starting with the porting of the standalone code → Clustering algorithm CLUEAlgo

For the future:

- Complete the CLUEAlgo porting
- Move to CMSSW-Patatrack
- Understand how to implement compilation flags in SCRAM