191202 - CCE PPS Weekly Meeting

Agenda: https://indico.fnal.gov/event/22584/
Attendance: MarkD, Charles, Meifeng, MattiK, SabaS, Salman, PeterN, Paolo

Email Samantha to ask for email lists.

- Sub lists
- And one general list

Slack ➜ OLI

Questions from Mark:

1. What are the dimensions of future growth? Presumably processing more events is the main one, but will there be growth in the amount of data or computation per event?
    - CMS: yes, yes, yes, especially if we go from today to HL-LHC
    - DUNE: yes, yes, yes
    - ATLAS: yes, yes, yes
2. Somewhat related, is the set of kernels fixed, or does adding new events/analysis require adding new kernels?
    - Yes, physicists will permanently come up with new code
3. For Patatrack, it was mentioned that multiple events are needed to fill a V100 card. What is the variation in the amount of data/computation for events?
    - Factor 10, some events take 10 times more time than other lighter events

Funding situation update

- Possibility that funds could come sooner than January ➜ Salman will contact institutional leads

Portability solution - What should we look at first?

- Kokkos
    - Matti: From training, Kokkos seemed more suitable than Raja, OpenMP, SYCL
        - Nested loops looked inconvenient in Raja
        - OpenMP: cannot get it to work as efficiently as TBB on CPU, doesn't mix well with TBB threads
        - SYCL implementation seem not to be very mature yet
        - Kokkos is higher level than SYCL

- Salman:
  - One advantage: Kokkos implementation for CUDA
  - For SYCL, don't have to wait for the next generation of hardware
- Charles:
  - ATLAS FastCaloSim is the most advanced GPU code in Atlas right now ➜ should we put that into the plan?
  - This is using CUDA right now, but also working in SYCL
  - Kokkos: difficult to run with concurrent kernels
    - If we cannot fill up a GPU with one kernel, you want to fill it with several concurrent kernels
  - Agrees that backends for SYCL are not yet very mature
  - End goal: solution that we select should become part of the C++ standard
- Meifeng:
  - ATLAS FastCaloSim: also looking at HIP (easy translation from CUDA to HIP, can already run on NVidia GPUs, will be able to run on AMD GPUs when they become available to test
  - Kokkos, HIP and SYCL on different levels
    - Kokkos is higher level, allows data structure controls, higher level than the others
    - Salman: dangerous to start writing wrappers for lower level tools ➜ should use standard supported tools
  - Wire cell:
    - C++17: Support for parallel STL
      - Moving forward, C++ will support more parallel structures
      - Charles: both Intel and AMD will port their solutions into C++
      - Matti: Kokkos is trying to make itself obsolete by being absorbed into the standard
    - Would be ok to study Kokkos first.
- Email discussion with Tim
  - Kokkos is a good starting point

Kokkos it is!

Use cases - Are we ready to share code?

- FastCaloSim for ATLAS
  - Git repository that has the code base, all self-contained, comes with data files ➜ in Meifeng's private github repository, send email to Meifeng
  - Easy to distribute, instructions to run
  - Has been shown to be easily run at different places, Cori II, A21 test nodes, etc.
- Wire cell

    - Open source
    - Instructions how to run, might need test data
    - Was not run on Cori II
- Patatrack
    - Code is in CMSSW, open source in github
    - Scattered around within other CMSSW code
    - Prepared documentation that walks you through the code
        - https://gitlab.com/NESAP/cms-gpu-tracking/benchmark/blob/master/CMSSW.md#application-structure
    - Docker container which holds the release, needs a couple of steps to get it running, setup to run on Cori II GPU nodes
        - Instructions for running on Cori GPU nodes: https://gitlab.com/NESAP/cms-gpu-tracking/benchmark/blob/master/NESAP.md#recipe
    - Data input: CMS open data test samples

General question about allocations

- Meifeng: get allocations for the PPS project on all major SuperComputers
    - Cori II: ERCAP
    - ANL: test nodes and Theta
    - Summit: talk with Jack Wells,
    - Action item for Salman

Question about data sizes

- Can we run it on a laptop?
    - FastCaloSim: yes
    - Wire cell: yes
    - Patatrack: in principle yes, container image is 10s of GB, input data can be 10s of MB, need to try
- Can we run profilers on it and will it point me to the right kernels?
    - FastCaloSim: yes
    - Wire cell: yes
    - Patatrack: nvprof is included in the container image, others depends on how you manage to run your prop files in the container

Overall goal: define work plan for the next weeks

- 1st step:

    - Everybody run all the code
    - Start with Cori II, because accounts is what is difficult to get
- 2nd step:
    - Get to know the code, use profilers to figure out what part of the code to look at
- 2nd step
    - Start looking at Kokkos

- major players: Charles, Meifeng, Matti, Mark
    - Charles is adding Mark to his allocation
    - Paolo is sending email to Meifeng
    - Matti has GPU Cori access

Status for Mark

- Wirecell
    - Compiled on laptop
    - Ran cfg/uboone/simpsp (“wirecell -c main-simple-quiet.jsonnet”). Realized this only does generation, not signal processing. Switch to main-simple-quiet-sp.jsonnet input
    - Adding “-l stdout -L debug” to the command line prints helpful output
    - Ran VTune on the input (“hotpots” collection) to get a feel for what the call stacks look like and where some kernels might be located.
        - OmnibusSigProc::operator() seems to sequence the bulk of operations
        - OmnibusSigProc::decon_2D_init - main deconvolution (according to comments) - does this form a kernel? Are the other decon_2D_* routines other kernels?
- Cori - Can’t login, MFA tokens were removed from account, submitted help ticket
- Patatrack - download script failing
- FastCaloSim - have access to Github repo