HEP-CCE

PPS Next Steps

Oliver Gutsche (FNAL), Charles Leggett (LBNL), Meifeng Lin (BNL)

HEP-CCE All-Hands Meeting
Nov 4-6, 2020

What we have accomplished so far

- Three use cases covering different experiments and representing different algorithms
  - FastCaloSim (ATLAS) – fast calorimeter MC simulations
  - Wire-Cell Toolkit (DUNE) – detector response simulations
  - Patatrack (CMS) – track reconstruction
- All three codes were profiled and suitable kernels identified.
- CUDA implementations had existed prior to official start of HEP-CCE.
- Kokkos implementations are either complete or in progress.
- FastCaloSim also completed SyCL port
- Many lessons learned from Kokkos experimentation
  - Most needed features are supported; some lacking
  - Once complete, portable across different architectures
  - Is there a performance penalty? May vary case by case.
- Considerable effort to port codes from CUDA to Kokkos

PPS Year 1 Work Plan from Feb 2020

- **Q1 (FY20Q2): code preparation / benchmark definition**
  - Deliverable: Take use cases and make them run stand alone and on Cori II GPUs [done]
  - Deliverable: develop metrics [done]
- **Q2 (FY20Q3): benchmarking on CPU and GPU**
  - Deliverable: benchmark use cases [done]
  - Choose first technology to try and start implementing in all 3 use cases [done]
- **Q3 (FY20Q4): finish first implementation**
  - Deliverable: finish implementation of first technology on all 3 use cases, write up findings
    - FastCaloSim [done]
    - Patatrack [done]
    - Wire-Cell *[in progress]*
  - Finalize matrix of technologies to investigate *[in progress]*
- **Steady progress.**
- **Most deliverables accomplished. Plus extras: FastCaloSim-SyCL, WCT refactoring, etc.**

Next Steps

- There are other portability solutions to investigate:
  - SyCL seems comparable to Kokkos in terms of effort of porting, feature support
  - OpenMP/OpenACC may require less code change, but how well it supports C++ and what the performance may be remains to be seen.
  - RAJA may be similar to Kokkos, but probably a bit more lightweight. Worth investigating?
  - Should we consider Alpaka as well?
  - Parallel STL may eventually be a suitable choice (depending on compiler availability)?

Code/Programming Model	CUDA	Kokkos	SyCL	OpenMP OpenACC	Alpaka	RAJA	Parallel STL
FastCaloSim	★	✭	✭	?	?	?	?
Wire-Cell	★	☆	✧	?	?	?	?
Patatrack	★	✭	✧	?	?	?	?

★ Complete and Optimized ✭ Complete, needs optimization ☆ In progress ✧ Planned

Points for Discussion (1)

- Different use cases have different complexities and different baselines.
  - Reflected in different levels of progress in Kokkos implementation
  - Should we bring all to the same level before moving on to the next programming model?
- Should we expand the use cases? Possible addition: ACTS
  - New C++ tracking software
  - Will be used by ATLAS for LHC Run 3 and Run 4
  - Designed with concurrency in mind; CUDA effort ongoing
- Coordination within PPS and with other CCE groups
  - Is there anything we can do to facilitate cross-pollination within PPS?
  - IOS: what can we do w.r.t data models for portability, reduced I/O overhead?
  - CW: is there anything we should take into consideration from the workflow group?

Points for Discussion (2)

- How do we disseminate our results and finding?
  - Need a common publication/presentation policy
  - Need to identify venues for outreach
  - Possibility to organize a HEP portability workshop?
- How can we better communicate our needs to the framework developers?
  - Kokkos & SyCL so far relatively responsive to requests (Slack&mailing lists).
  - Do we want to have a more organized discussion with them?
  - Is there a (need for) common portability layer over optimized vendor libraries?
- Can we help drive the development of the frameworks and standards?

Year 2 Priorities (Preliminary)

- Finish Wire-Cell Kokkos implementation and write up findings
- Learn from the Kokkos experience and re-examine code structures/porting strategies for all the use cases, and adjust for the next programming model being investigated
- Wire-Cell and Patatrack move on to SyCL
- Will probably still have time to investigate another programming model: OpenMP/OpenACC?
- Introduce ACTS as a fourth use case?
- Coordinate and collaborate more closely with IOS