Software Project Discussion on Multi-Threading

Short Introduction

2020-05-06

Motivation

- Transition to multi-threading has been ongoing in HEP for most of the last decade
  - Some projects are very mature now
  - Some are maturing and approaching completion
  - There are even some areas not started yet
- Many technical options were examined at a community level through the *Forum on Concurrent Programming Models and Frameworks*
  - http://concurrency.web.cern.ch/
  - This was extremely useful for a technical interchange of ideas and results
  - And taking a fairly consensual decision in the community

Intel Threaded Building Blocks

- Decision to use TBB was adopted by multiple projects
- Motivated by this being one of the best technical solutions at the time
  - Plus long term support commitment from Intel
  - Evidently excellent performance on x86_64 CPUs, which were (and are) the processing workhorses for the community
- Additionally, converging on a single underlying solution was important for a coherent solution in HEP applications
  - Total number of active work threads can be optimised
  - Overcommitting too many work intensive threads hurts throughput (context switches) and risks memory blow-ups
    - This happens if, e.g., TBB and OpenMP are both used

tbb::task

- For HEP use of tbb::task became ubiquitous
  - Low level task support allowed flexibility
  - Not many things we wanted to do in parallel fitted with the higher level parallel constructs (like parallel_for)
    - Although I think these were used successfully in a few places
    - Again, showing the success of a unified low-level interface
- But… the world changes
  - https://threadingbuildingblocks.org/ ➜ https://github.com/oneapi-src/oneTBB
  - We now know multi-threading isn't enough
  - https://github.com/oneapi-src/oneTBB/issues/243
    - Intel are deprecating the current task interface
    - So we will have to evolve our use

As usual, we have Live Notes!

Meeting Today

- There are alternatives to the use of tbb::task that we have today
  - It's also highly unlikely that this is removed from the version of TBB we'd use for Run 3
  - So no real risk that things will break hard short term
- However, the coherence from TBB for complete HEP applications we don't want to lose
  - Experiment Framework + Geant4 + ROOT
- Discussion today to know where each major piece of the framework is
  - (Thanks to everyone for preparing something for this meeting at rather short notice!)
- Start to understand options for evolution
  - Certainly not going to take decisions at this stage (e.g. for things like HPX)
- Doing R&D here, in the spirit of the Concurrency Forum, we would like to track in the HSF
  - Frameworks WG would be a very natural place as we go on