Scheduling algorithms in LHCb’s upgrade software framework

N. Nolte, S. Stahl
J. Albrecht, E. Govorkova, C. Hasse, R. Matev, G. Raven
1 TU Dortmund, 2 CERN, 3 Nikhef
On behalf of the LHCb Collaboration

LHCb in Run 3 [CERN-LHCC-2018-007, LHCB-TDR-017]
- Single arm forward spectrometer specialized on beauty and charm hadrons
- Detector upgrade during LS2 (2018-2020)
- From Run 3 onwards:
  - purely software based trigger system
  - full online reconstruction at 30 MHz

General design choices
Running a 30 MHz software trigger requires a multithreading friendly framework with low overhead
- Functional data processing in form of reentrant and thread-safe algorithms, explicitly declaring data dependencies
- Parallelization over events
- Data and control flow is configured once before event processing to achieve minimal scheduling overhead during runtime
- Detect unmet dependencies early
- Automate dependency resolution
  - easier configuration
  - less prone to errors
  - no production of unused data by construction

Event processing - Runtime
- Iterate over the list of basic nodes
- Is the current node requested by any parent node, i.e. is any parent node not yet evaluated? If so:
  1. execute its data producers in the right order, if they have not been executed yet
  2. execute basic node itself
  3. save its decision and notify its parent about the outcome
  4. the direct parent evaluates its decision if the execution policy permits it
     if short-circuiting is allowed: evaluate as soon as possible
     else: evaluate after the last child has been evaluated
  5. If evaluated, the parents’ decision is given higher up the chain, continuing recursively from Step 4.

Barrier - Sharing work
- Resource intensive work may worth be sharing
- Example: Multiple lines select tracks with some intersection
  - invoke the track fit algorithm only once on the union of all track selections
- This requires optional data dependencies, since some lines might have retired before
- Optional dependencies are not pursued in the data dependency resolution, otherwise algorithms might be scheduled unnecessarily
- Instead, the barrier introduces additional order constraints in the configuration step to ensure that all necessary algorithms ran before invoking the expensive algorithm

Outlook
- Scheduler implemented as default in the LHCb upgrade framework.
- One can define control flow and write upgrade trigger lines, which are processed in multiple threads.
- Performance overhead needs to be tested in the realistic environment of HLT1 and HLT2.
- Benchmark the bookkeeping overhead of barriers
  - Gathering inputs into a union
  - Scattering results to each successing algorithm