

Optimization of HLT2 selections at the LHCb experiment

Daniel Magdalinski
01 December 2023



SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086



About me: Daniel Magdalinski

- Born and raised in Stockholm, Sweden
- Both Bachelor and Master at Lund University



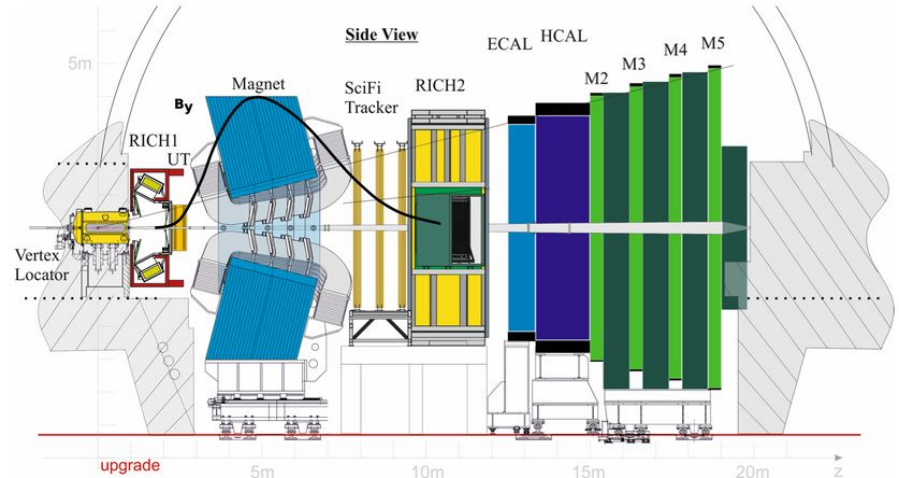
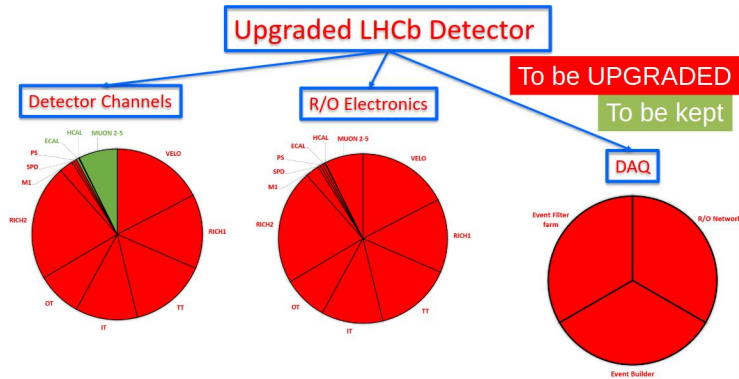
About me: Daniel Magdalinski

- Born and raised in Stockholm, Sweden
- Both Bachelor and Master at Lund University
- CERN Summer Student 2022
- Now: ESR6 in Amsterdam since October 2022
 - Optimization of HLT2 trigger



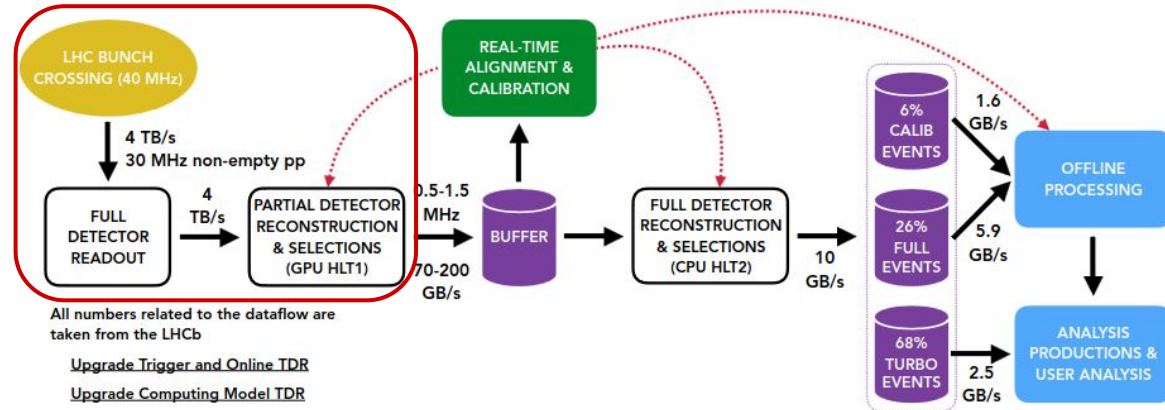
LHCb experiment

- Forward spectrometer designed for flavour physics through beauty and charm decays
- Detector upgraded for Run 3 to handle increased luminosity
 - Changes to nearly all subsystems
- Flexible full-software trigger system enables a more general physics program



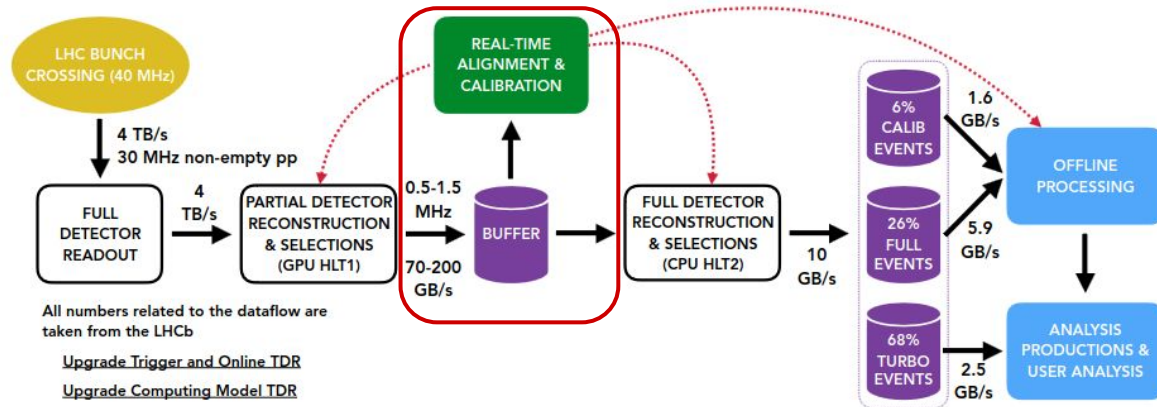
LHCb Run 3 trigger

- 30 MHz and 4TB/s of detector read-out → 10 GB/s data storage
- **HLT1:**
 - GPU-based algorithms focused on tracks, displaced decay vertices and muons



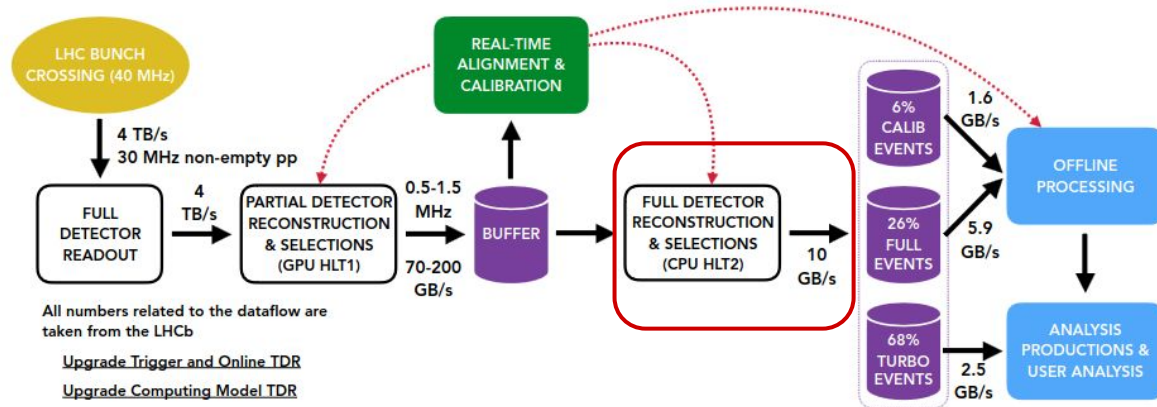
LHCb Run 3 trigger

- 30 MHz and 4TB/s of detector read-out → 10 GB/s data storage
- HLT1:
 - GPU-based algorithms focused on tracks, displaced decay vertices and muons
- **Alignment & Calibration**
 - Event buffer between HLT1 and HLT2
 - Real-time alignment and calibration giving offline-level reconstruction to HLT2



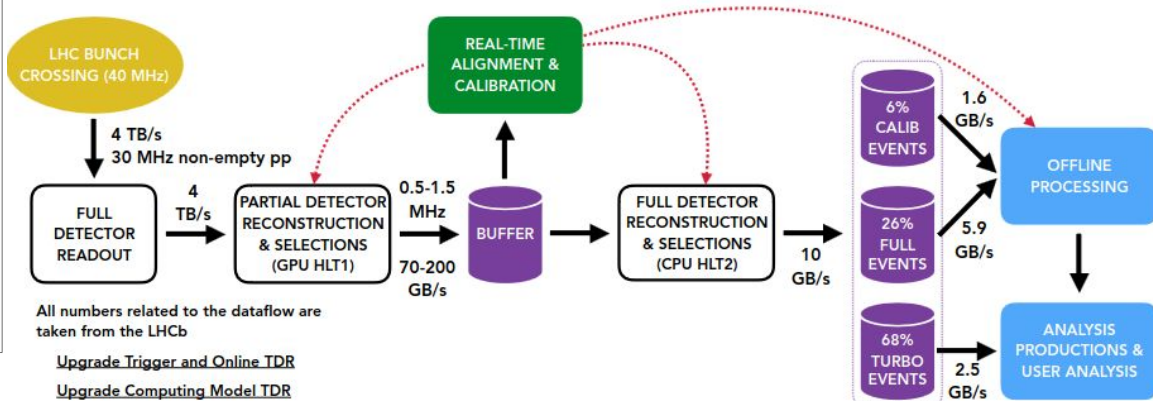
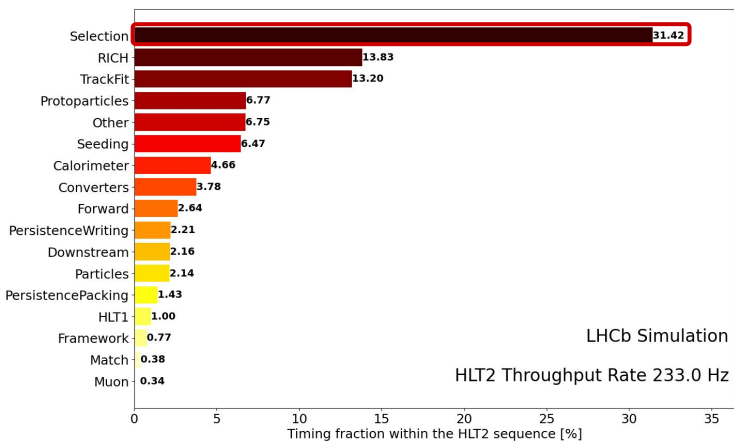
LHCb Run 3 trigger

- **HLT2:**
 - Reconstruction and selection of physics objects for analysis through trigger lines



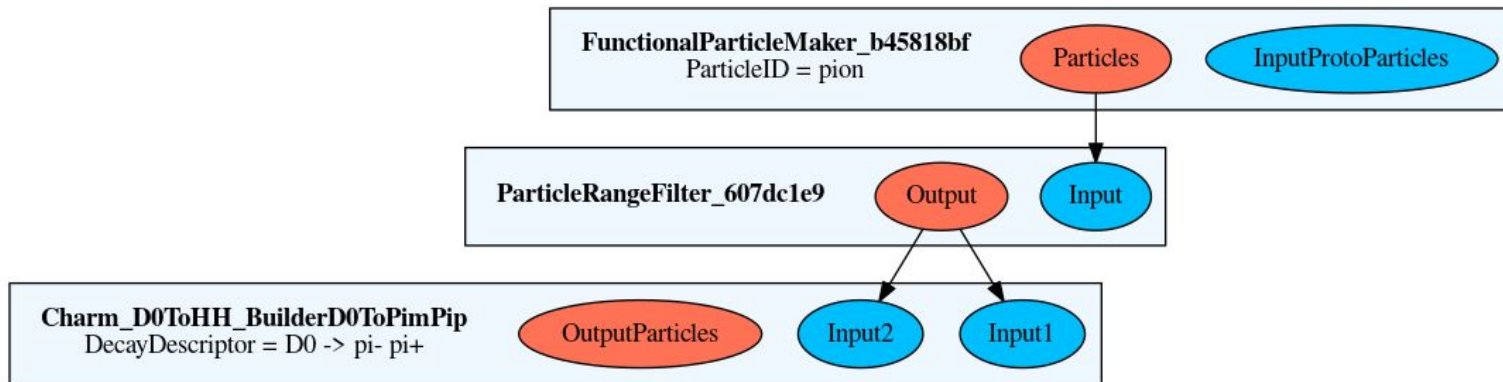
LHCb Run 3 trigger

- HLT2:
 - Reconstruction and selection of physics objects for analysis through trigger lines
- **Selections:** ~30% of HLT2 computing cost
 - Fraction growing as new lines are added



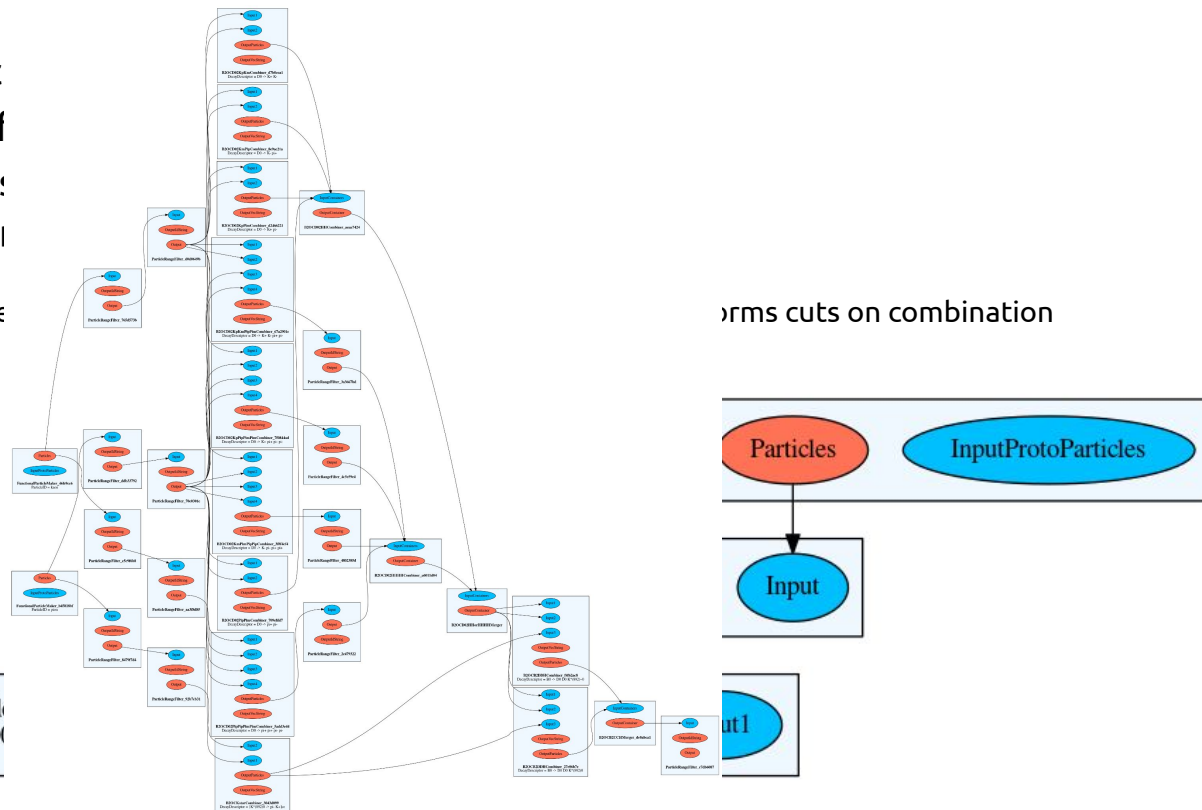
HLT2 trigger selections

- ~2400 trigger lines currently written
- Line is a collection of selection algorithms
- Lines usually consists of at least
 - Maker: Container of particles coming from reconstruction
 - Filter: Performs cuts on input particles
 - NBodyCombiners: Iterates over combinations of N input particles and performs cuts on combination



Can become quite complicated

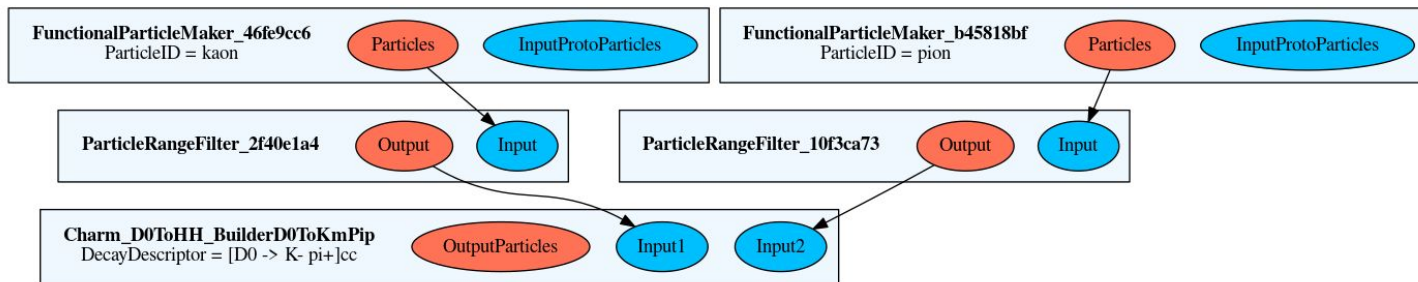
- ~2400 trigger lines c
- Line is a collection of
- Lines usually consist:
 - Maker: Container of p
 - Filter: Performs cuts
 - NBodyCombiners: It



HLT2 control flow optimization

HLT2 trigger control flow

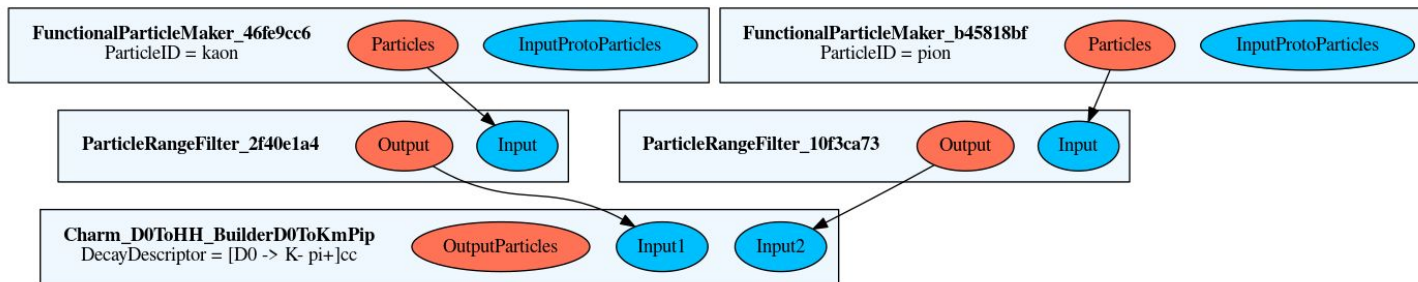
- Control Flow optimization
 - Data flow: The data dependence of trigger line algorithms



HLT2 trigger control flow

- Control Flow optimization
 - Data flow: The data dependence of trigger line algorithms
 - Control flow: Which **algorithm(s)** that decide if the line triggers
 - Can be a list of algorithms to perform early stopping

```
@register_line_builder([all_lines])
def dzero2kpi_line(name='Hlt2Charm_D0ToKmPip', prescale=1):
    kaons = make_kaons()
    pions = make_kaons_pions()
    dzeros = make_dzeros(kaons, pions, '[D0 -> K- pi+][cc]')
    return Hlt2Line(
        name=name, algs=charm_filters() + [dzeros], prescale=prescale)
```



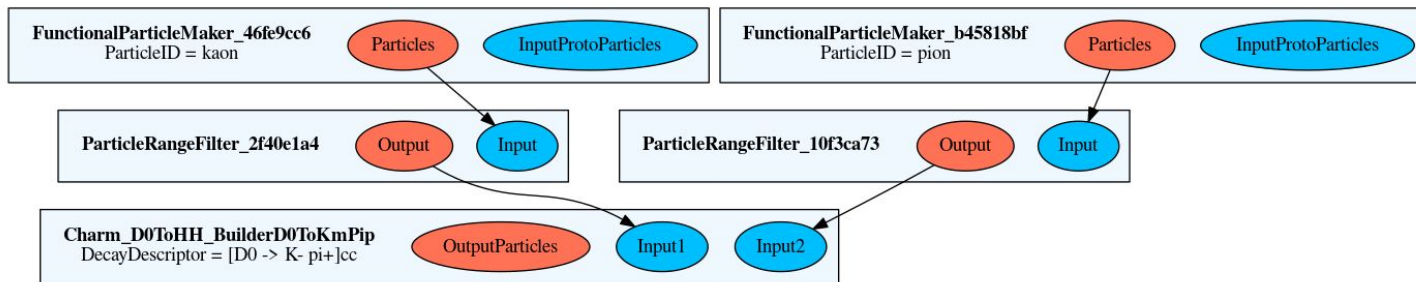
HLT2 trigger control flow

- Control Flow optimization
 - Data flow: The data dependence of trigger line algorithms
 - Control flow: Which **algorithm(s)** that decide if the line triggers
 - Can be a list of algorithms to perform early stopping
- Optimization approach
 - Test run for statistics on how often an algorithm outputs
 - Iterate through data flow
 - Add rarest algorithms to control flow

```
@register_line_builder([all_lines])
def dzero2kpi_line(name='Hlt2Charm_D0ToKmPip', prescale=1):
    kaons = make_charm_kaons()
    pions = make_charm_pions()
    dzeros = make_dzeros(kaons, pions, '[D0 -> K- pi+][cc]')
    return Hlt2Line(
        name=name, algs=charm_prefilters() + [dzeros], prescale=prescale)
```

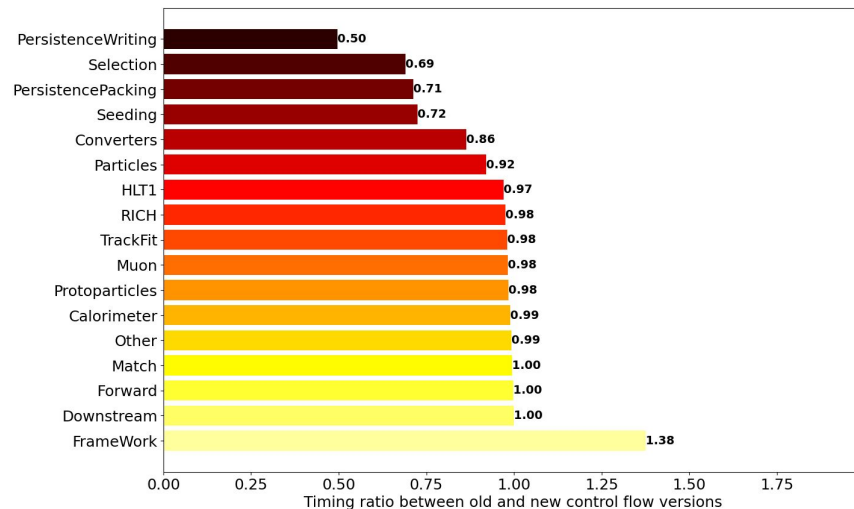


```
@register_line_builder(all_lines)
def dzero2kpi_line(name='Hlt2Charm_D0ToKmPip', prescale=1):
    kaons = make_charm_kaons()
    pions = make_charm_pions()
    dzeros = make_dzeros(kaons, pions, '[D0 -> K- pi+][cc]')
    return Hlt2Line(
        name=name, algs=charm_prefilters() + [kaons,dzeros], prescale=prescale)
```



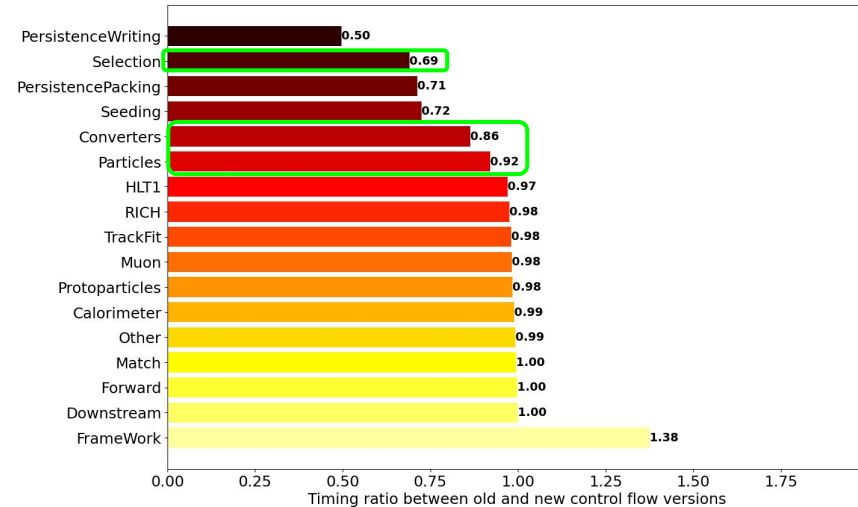
Control flow results

- Performance CI-test
 - 20000 minimum bias events
 - Isolated environment at CERN
 - Similar to trigger operations
- Throughput rate: 233.6 Hz → 272.7 Hz
 - 16.7% improvement



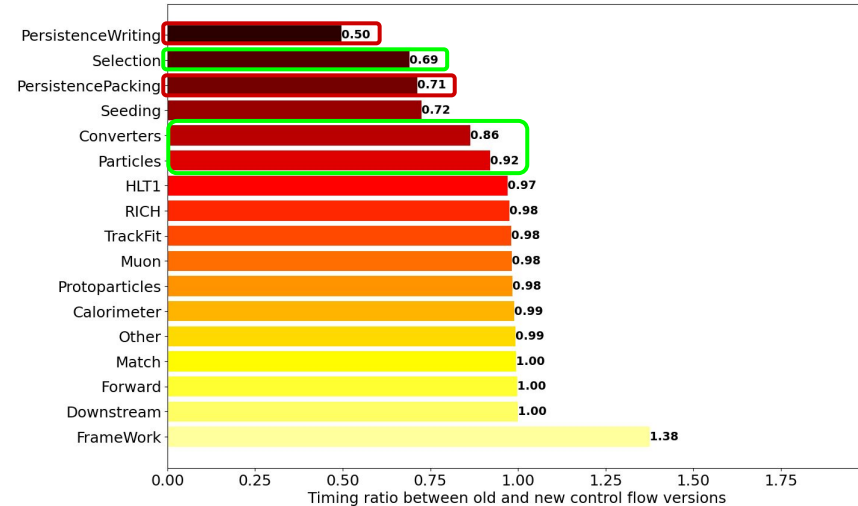
Control flow results

- Performance CI-test
 - 20000 minimum bias events
 - Isolated environment at CERN
 - Similar to trigger operations
- Throughput rate: 233.6 Hz → 272.7 Hz
 - 16.7% improvement
 - Improvements in various selection algorithms



Control flow results

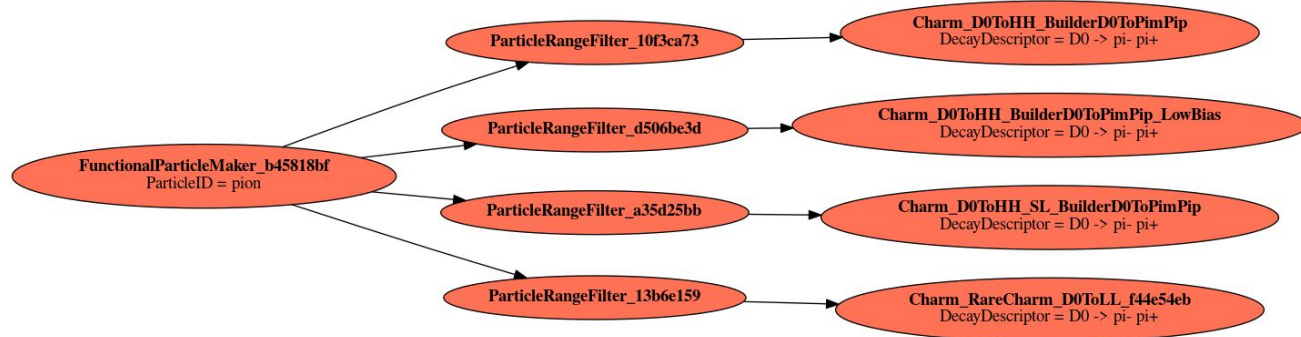
- Performance CI-test
 - 20000 minimum bias events
 - Isolated environment at CERN
 - Similar to trigger operations
- Throughput rate: 233.6 Hz → 272.7 Hz
 - 16.7% improvement
 - Improvements in various selection algorithms
 - Timing gain in persistence(data saving) worrying
 - Thorough checks into throughput agreement under way



HLT2 combiner optimization

HLT2 combiner optimization

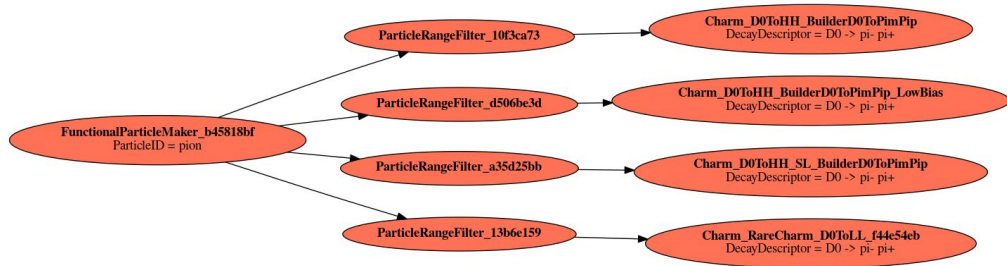
- Problem: Trigger lines might perform very similar operations
 - Tiny differences creates duplicate algorithms
- Goal: Identify and combine overlapping algorithms
 - Timing gains
 - Potential storage gains: Reducing duplicate particles
- Different setup, ~1500 lines



Combiner optimization

- Optimization
 - Identify combiners with common grandparent

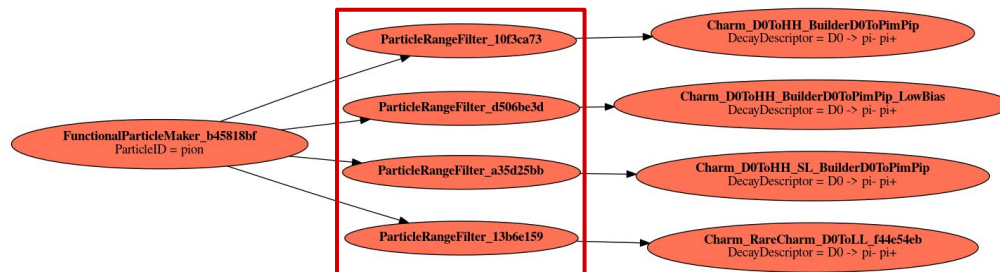
● Old:



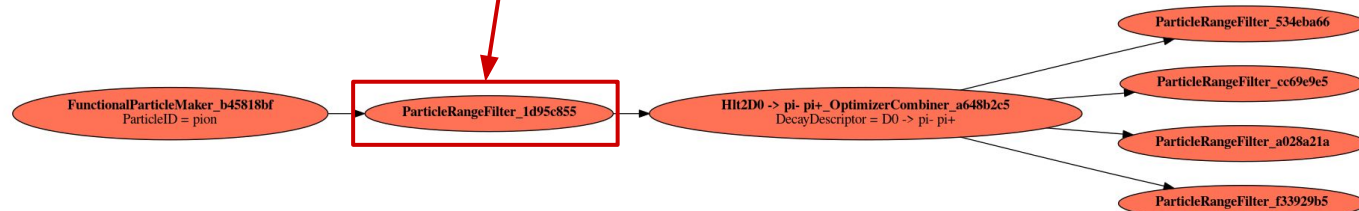
Optimization framework

- Optimization
 - Identify combiners with common grandparent
 - OR operator applied to cuts
 - Separation into original containers

● Old:



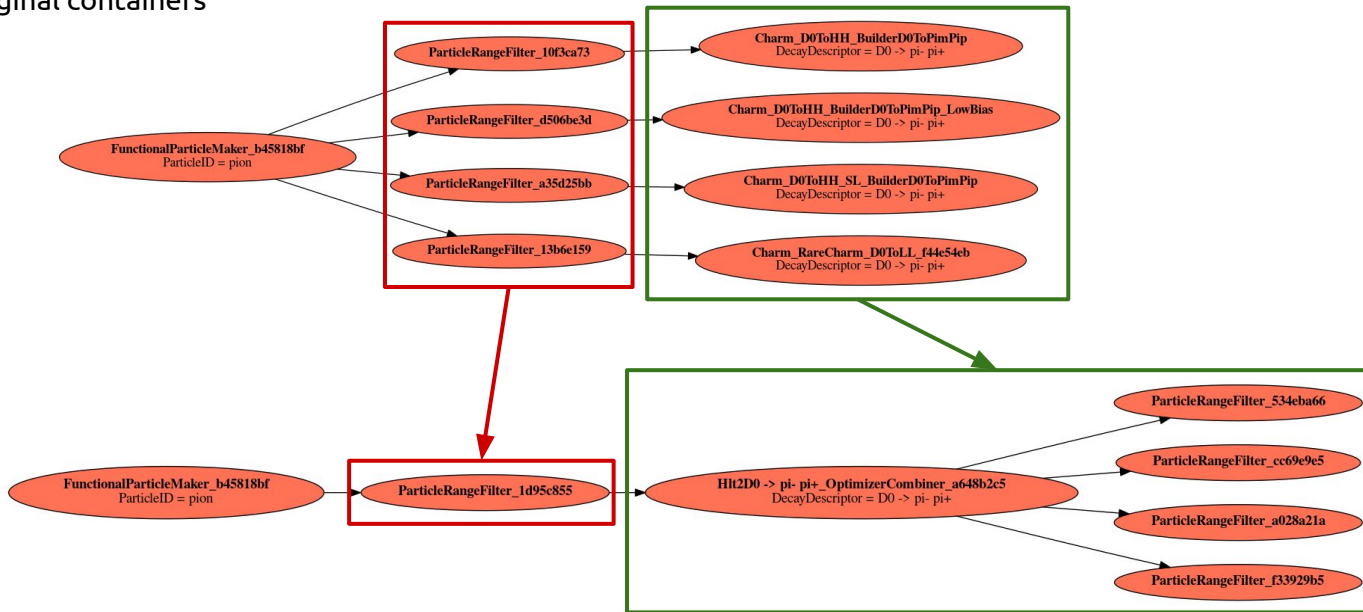
● New:



Optimization framework

- Optimization
 - Identify combiners with common grandparent
 - OR operator applied to cuts
 - Separation into original containers

● Old:



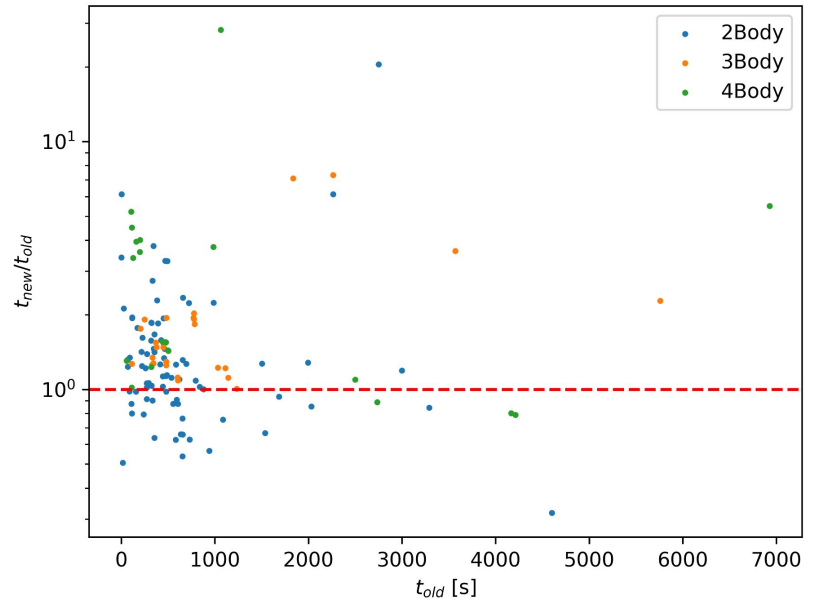
● New:

Initial performance

- Initial performance
 - 134 sets
 - ~500 combiners
 - ~1100 lines affected
- Naive combination
 - Full set always combined

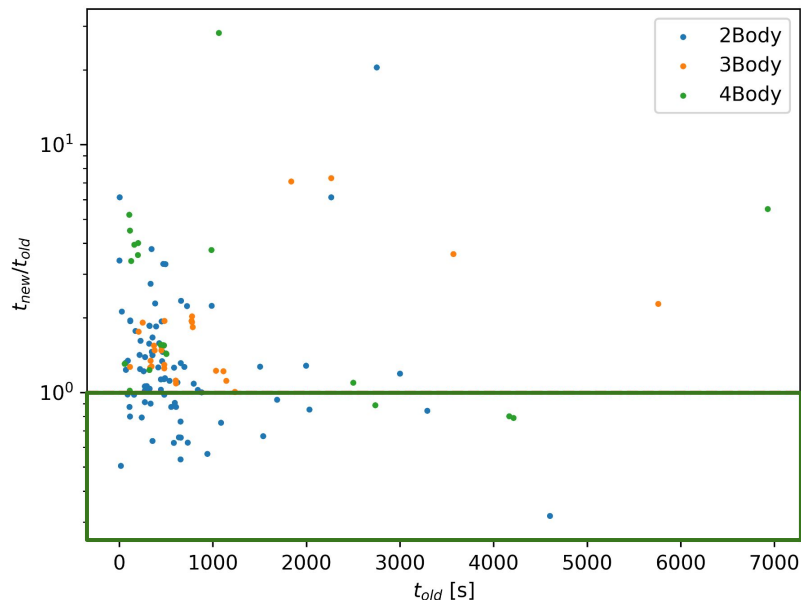
Initial performance

- Initial performance
 - 134 sets
 - ~500 combinators
 - ~1100 lines affected
- Naive combination
 - Full set always combined
- Overall most sets perform worse
 - Increase in combinations
 - Separation filters



Initial performance

- Initial performance
 - 134 sets
 - ~500 combinators
 - ~1100 lines affected
- Naive combination
 - Full set always combined
- Overall most sets perform worse
 - Increase in combinations
 - Separation filters
- Modular framework
 - ~6% improvement on selections
 - ~2% improvement on total trigger timing
- Potential improvements:
 - Identifying optimal combination sets
 - Simplifying cuts



Training & Outreach

- **Teachers assistant:**
 - CP violation course
- **BND school:**
 - Physics school on various astro and particle physics subject
- **Efficient Scientific Computing(ESC23) school:**
 - Programming school covering various ways to design efficient computing
 - C++, GPUs, OpenMP, MPI, TBB, Floating point, Memory usage
 - Gave a lightning presentation
- **NNV yearly conference**
 - 15 min presentation at national dutch physics society conference
- **Poster on LHCb Run 3 trigger at NWO Physics in January**

Conclusions & Outlook

- This work has focused on
 - Control flow optimization
 - Combiner optimization
- Work is ongoing
 - Control flow give a speed up of 16.7%
 - Thorough checks to confirm that output is the same
 - Further optimization
 - Initial results for combiner optimization shows slight improvement of only 2%
 - Optimal combination sets
 - Simplifying cuts
 - Storage gains?

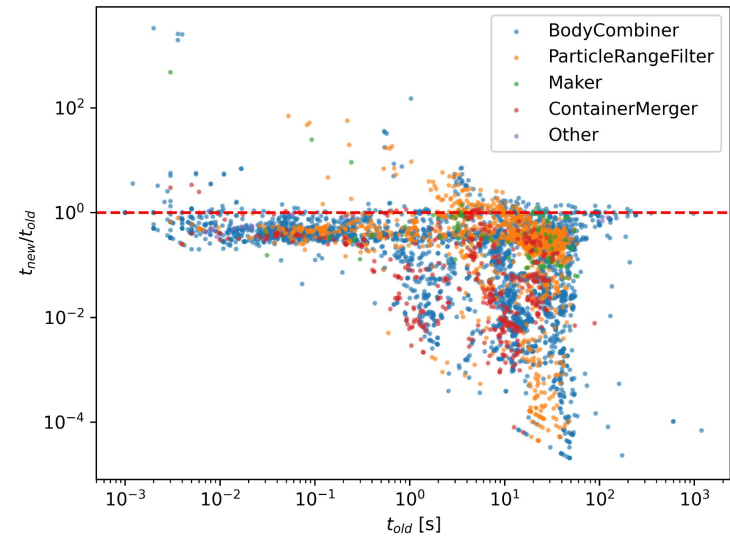
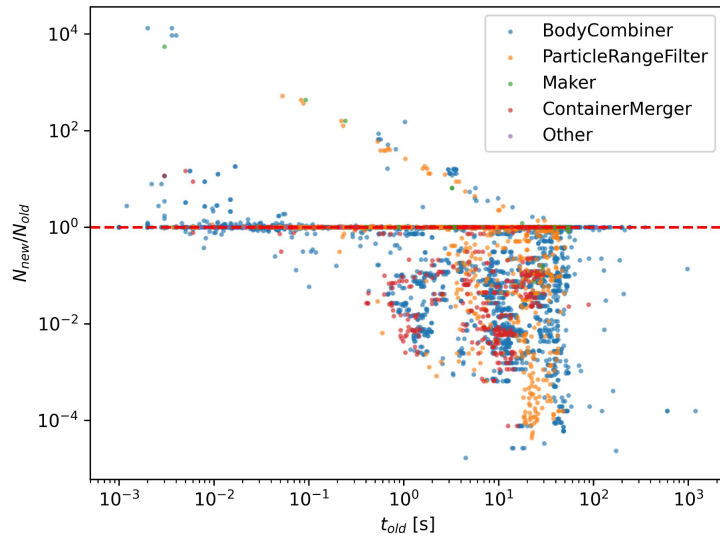
Conclusions & Outlook

- This work has focused on
 - Control flow optimization
 - Combiner optimization
- Work is ongoing
 - Control flow give a speed up of 16.7%
 - Thorough checks to confirm that output is the same
 - Further optimization
 - Initial results for combiner optimization shows slight improvement of only 2%
 - Optimal combination sets
 - Simplifying cuts
 - Storage gains?

Thank you for your attention!

Control flow results

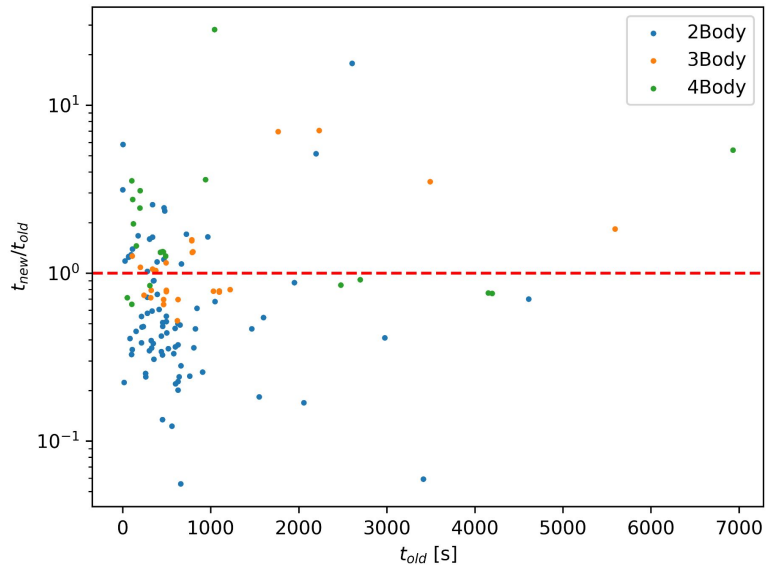
- Performance per alg
 - Event ratio
 - Timing ratio



Initial performance

- **Combiner + Input Filters**

- Separation filters are very expensive
- Ongoing work in simplifying the cuts



- **Full version**

