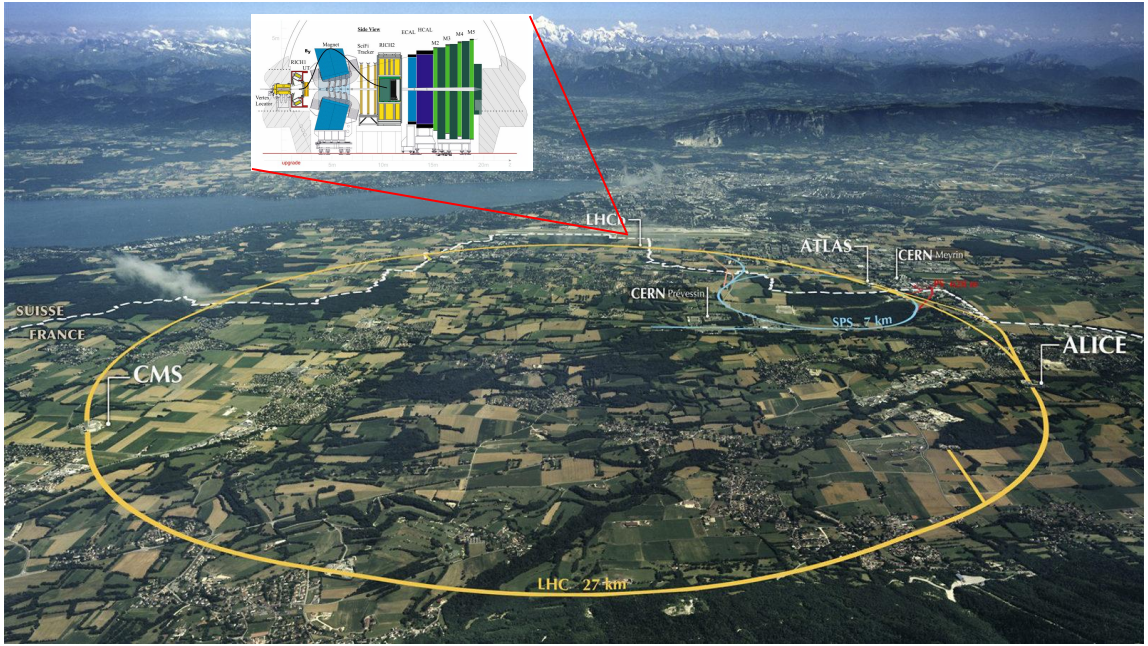
Optimization of HLT2 selection algorithms at the LHCb experiment

Daniel Magdalinski
3 November 2023

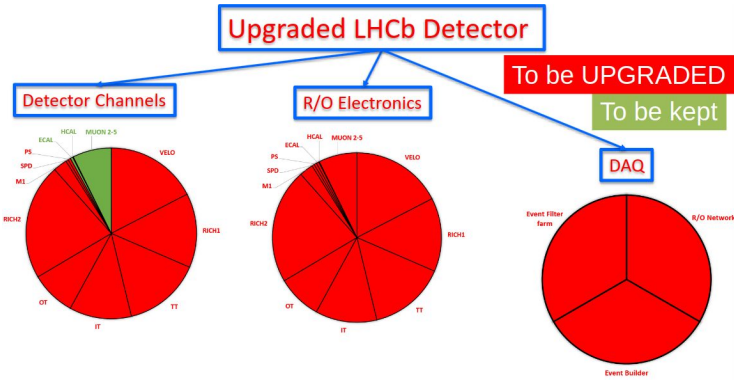
LHCb experiment at CERN

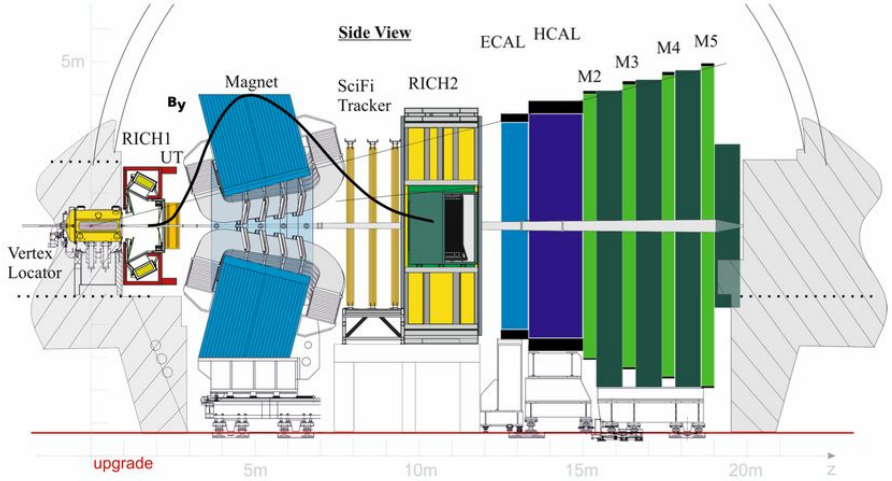
- LHC: Circular particle accelerator at CERN
- 4 main experiments:
  - CMS
  - ALICE
  - ATLAS
  - LHCb



LHCb experiment

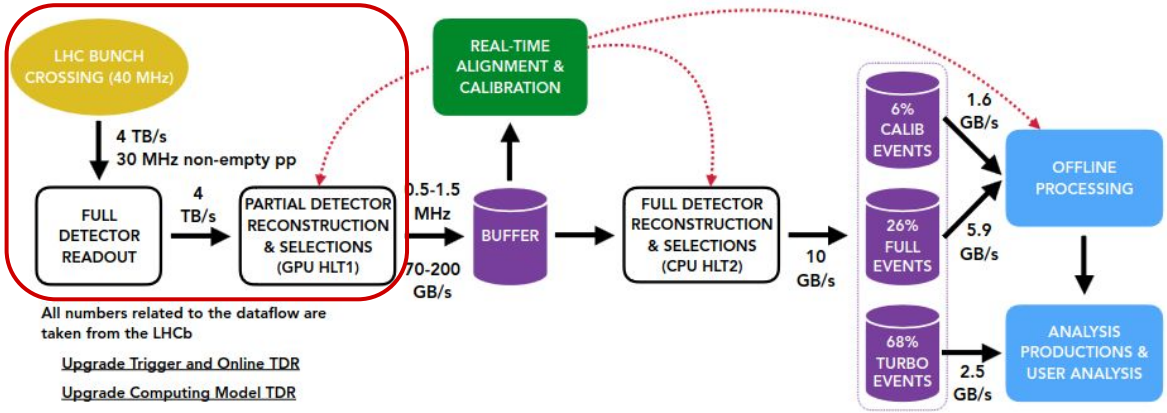
- Forward spectrometer designed for flavour physics through beauty and charm decays
- Detector upgraded for Run 3 to handle increased luminosity
  - Changes to nearly all subsystems
- Flexible full-software trigger system enables a more general physics program





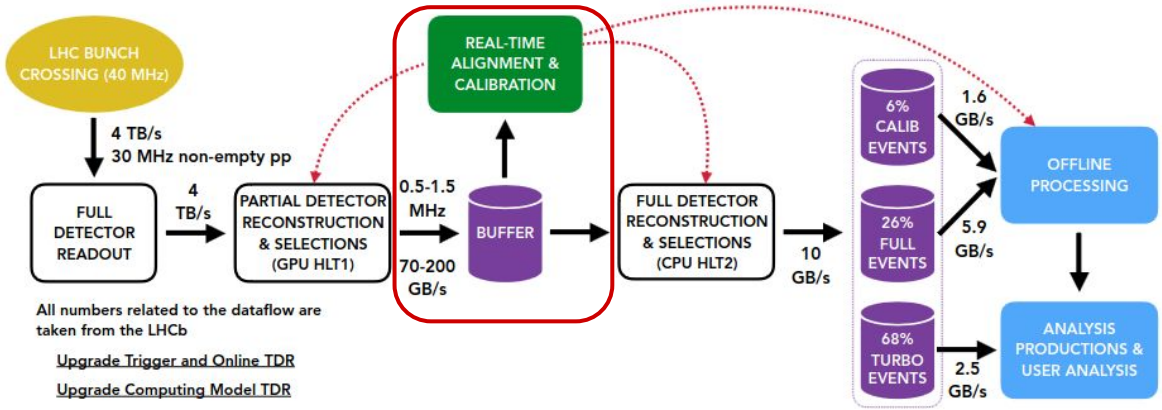
LHCb upgrade trigger

- 30 MHz/4TB/s 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



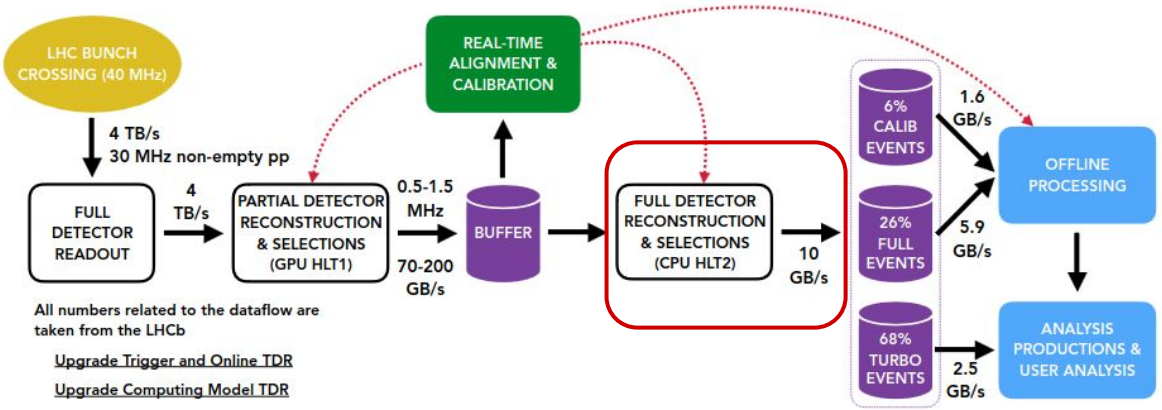
LHCb upgrade trigger

- 30 MHz/4TB/s 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



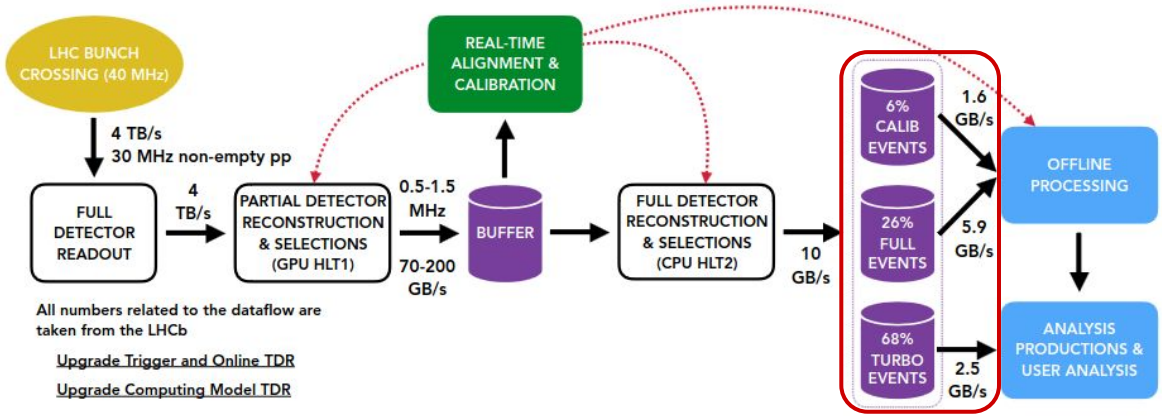
LHCb upgrade trigger

- HLT2:
  - Selections of physics object for analysis through trigger lines
- Selective persistence:
  - Option to save partial event information only relevant for analysis
  - Enables us to do much more physics with the same disk space



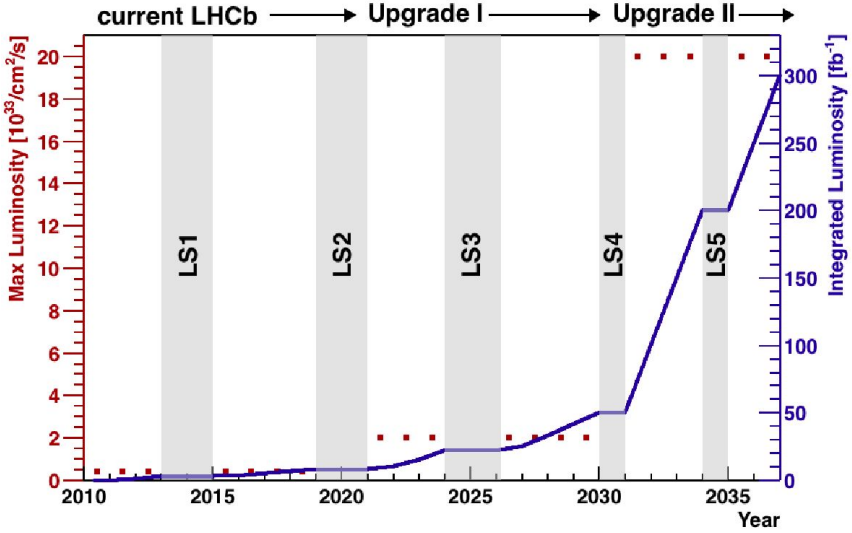
LHCb upgrade trigger

- HLT2:
  - Selections of physics object for analysis through trigger lines
- Selective persistence:
  - Option to save partial event information only relevant for analysis
  - Enables us to do much more physics with the same disk space



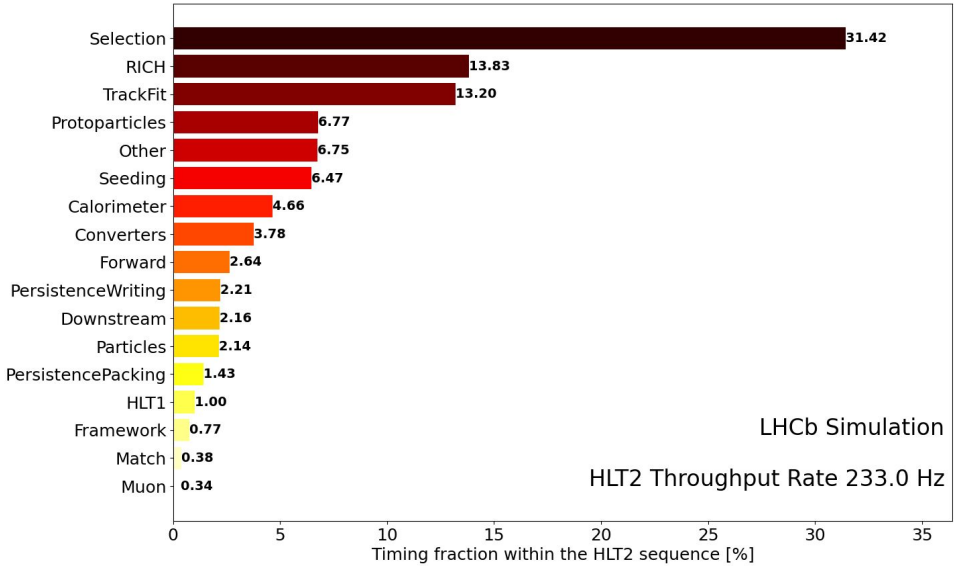
Motivation for optimization

- Luminosity increasing at LHC → more complicated events
  - Computation costs will increase
- LHC experiments are resource limited
  - There is more physics to analyze in the events



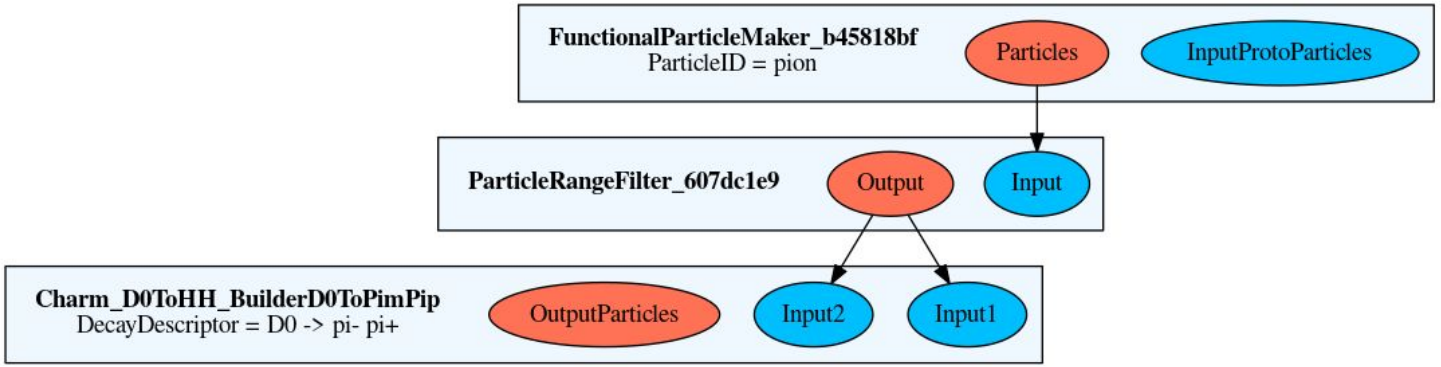
Why optimize selections

- Luminosity increasing at LHC → more complicated events
  - Computation costs will increase
- LHC experiments are resource limited
  - There is more physics to analyze in the events
- Selections: ~30% of HLT2 computing cost



HLT2 trigger selections

- ~1600 trigger lines making selections on various physics signatures
- Lines usually consists of at least
  - Maker: Creates a 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



HLT2 trigger line strategy

- Important: Trigger lines should be independent
- This important rule also means that lines might perform very similar operations
  - Tiny differences creates duplicate algorithms

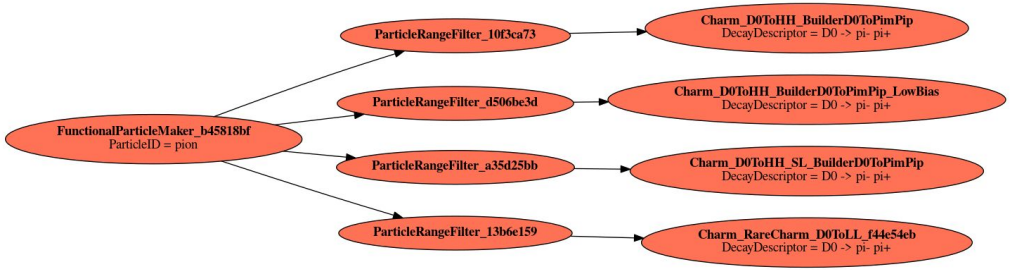
Optimization goal

- Important: Trigger lines should be independent
- This important rule also means that lines might perform very similar operations
  - Tiny differences creates duplicate algorithms
- Goal: Identify and reduce overlapping computations

Optimization framework

- Optimization framework(work in progress)
  - Identify similar combiners with common grandparent
  - Merges Filters and Combiners together with an OR applied to their cuts
  - Separating D0s into their respective lines using filters

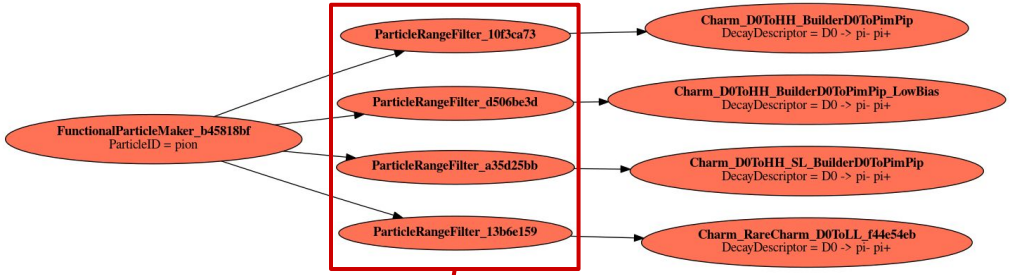
- Old:



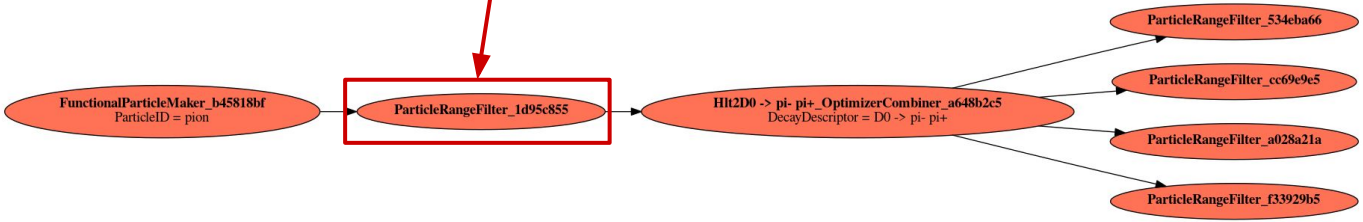
Optimization framework

- Optimization framework(work in progress)
  - Identify similar combiners with common grandparent
  - Merges Filters and Combiners together with an OR applied to their cuts
  - Separating D0s into their respective lines using filters

- Old:



- New:

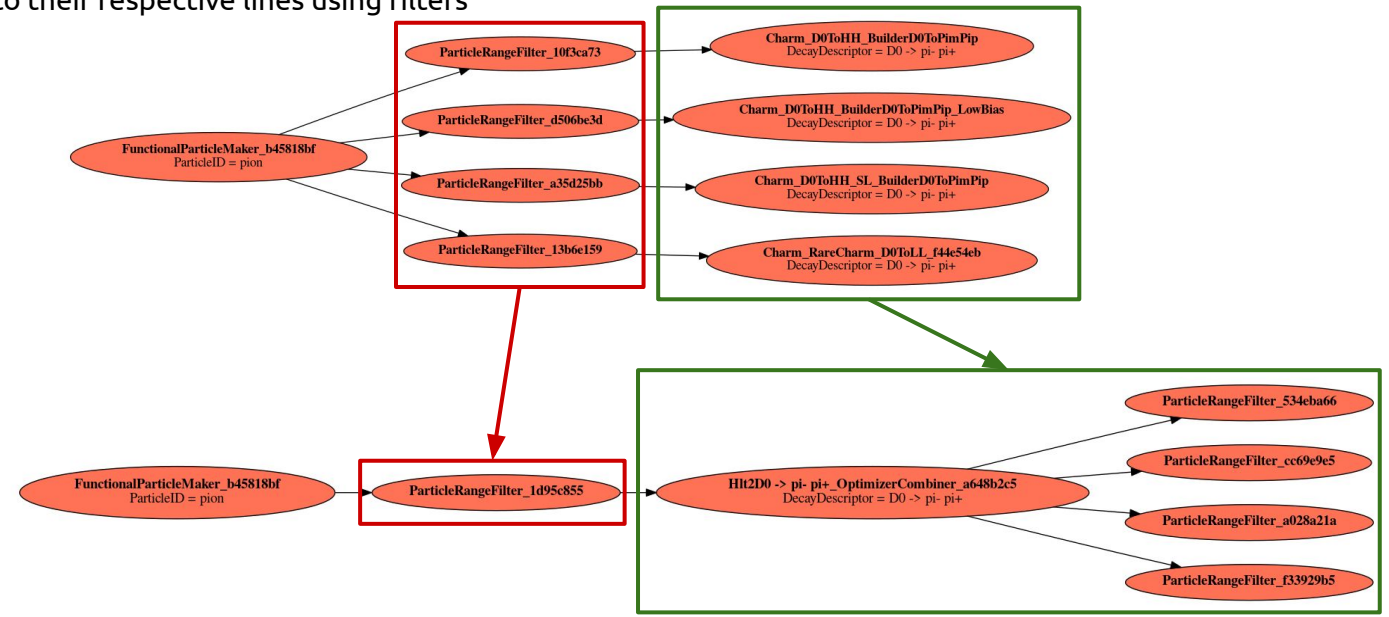


Optimization framework

- Optimization framework(work in progress)
  - Identify similar combiners with common grandparent
  - Merges Filters and Combiners together with an OR applied to their cuts
  - Separating D0s into their respective lines using filters

- Old:

- New:

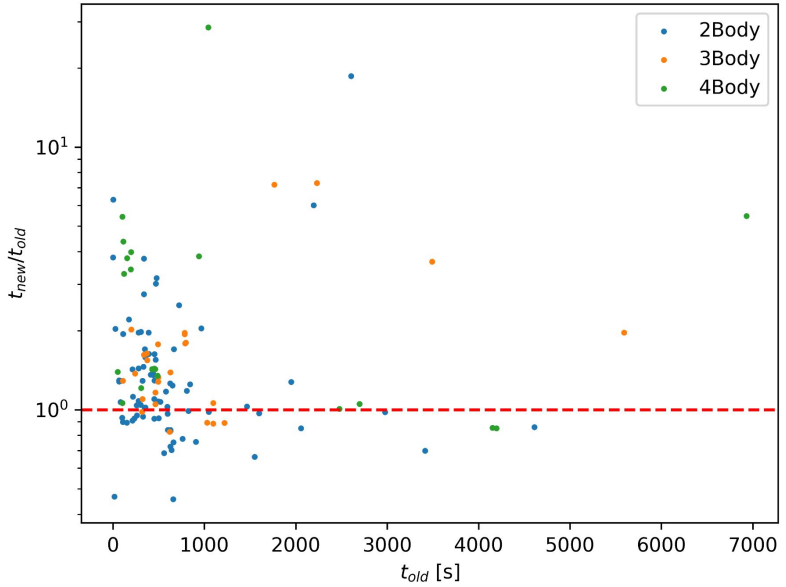


Initial performance

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

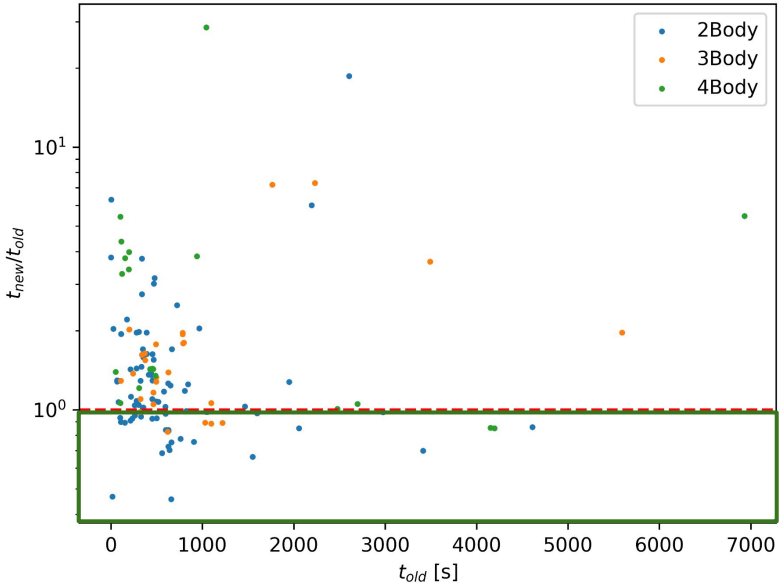
Initial performance

- Initial performance
  - 134 sets of combiners
  - ~500 combiners in total
  - ~1100 lines affected
- Naive combination
  - Full set always combined
- Overall most sets perform worse



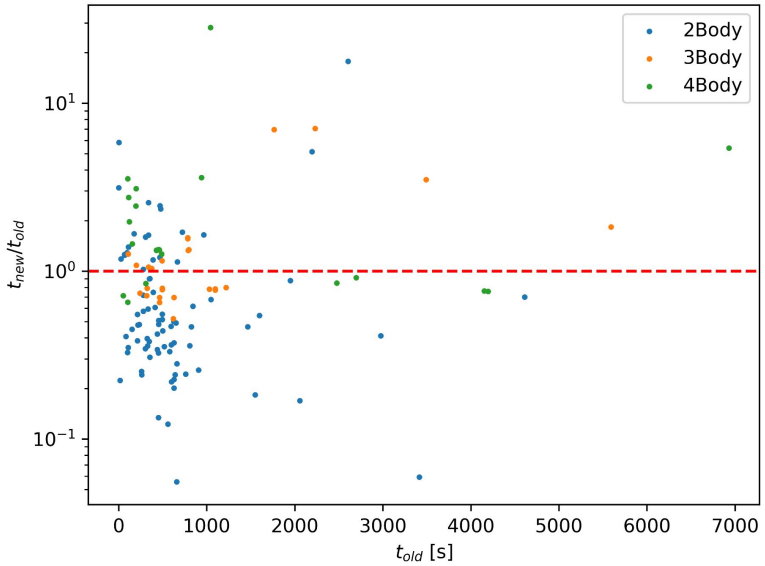
Initial performance

- Initial performance
  - 134 sets of combiners
  - ~500 combiners in total
  - ~1100 lines affected
- Naive combination
  - Full set always combined
- Overall most sets perform worse
- But:
  - Modular framework
  - ~1.8% improvement on total trigger timing

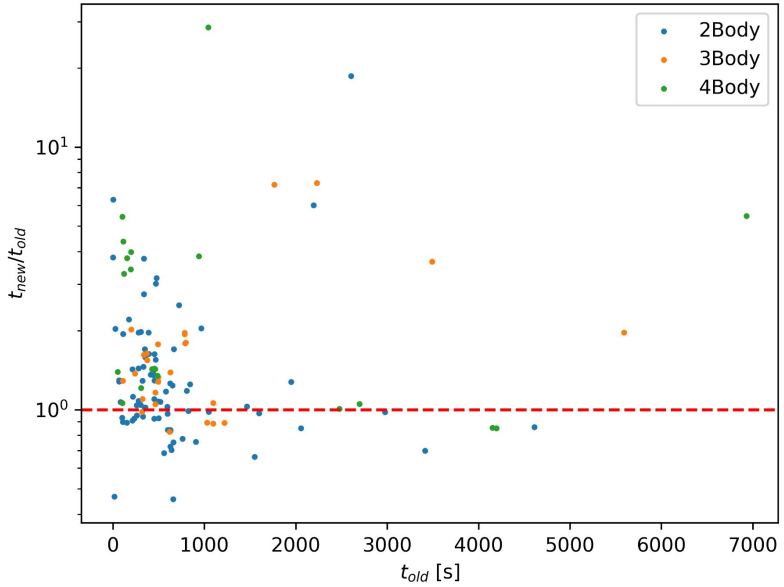


Initial performance

- Combiner + Input Filters
  - Separation filters are very expensive
  - Ongoing work in simplifying the cuts



- Full version



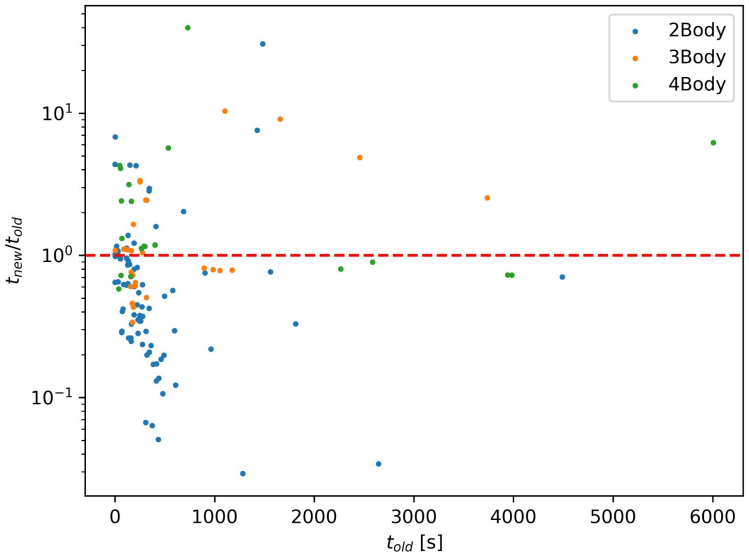
Conclusions & Outlook

- Optimization of HLT2 is needed for future operations and very useful for current
- This work has focused on combiners sharing a common grandparent
- Work is ongoing
  - Initial results show slight improvement of 1.8% but there is potential for more
    - Combination sets
    - Separation filter cuts

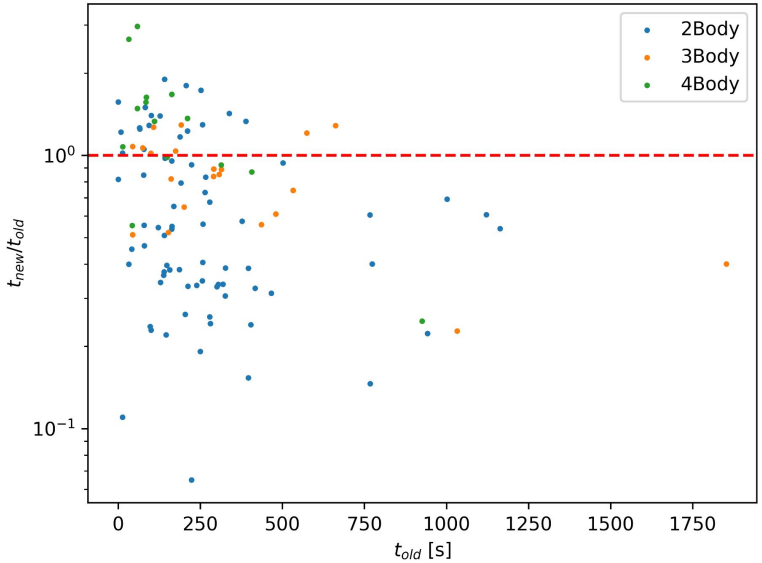
Thank you for your attention! Questions?

Initial performance

- Only Combiners



- Only Input Filters



Prescaling

- Prescale complicates the problem
- Old version:
  - Part of the algorithms run
- New version:
  - The full combiner runs every time it is needed
  - Unnecessary combinations are made