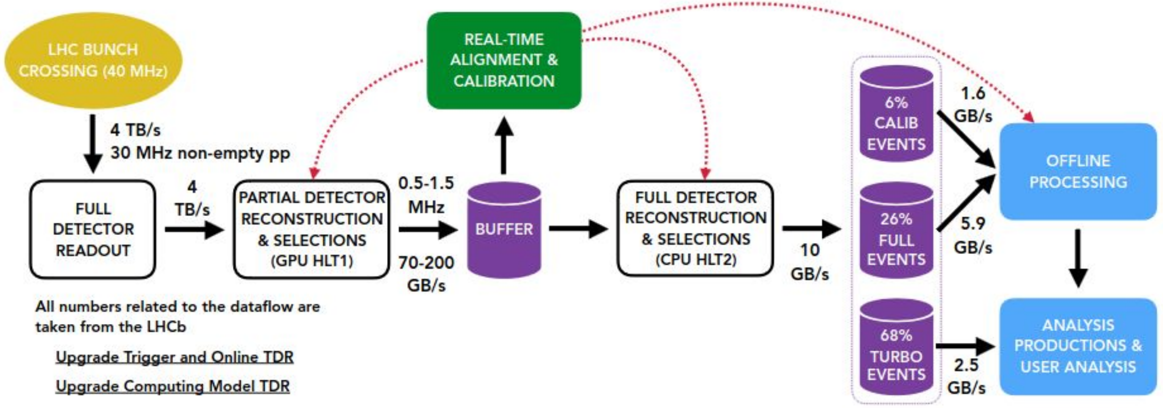
ESR6: Optimization of HLT2 selections at the LHCb experiment

Daniel Magdalinski
01 October 2024

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

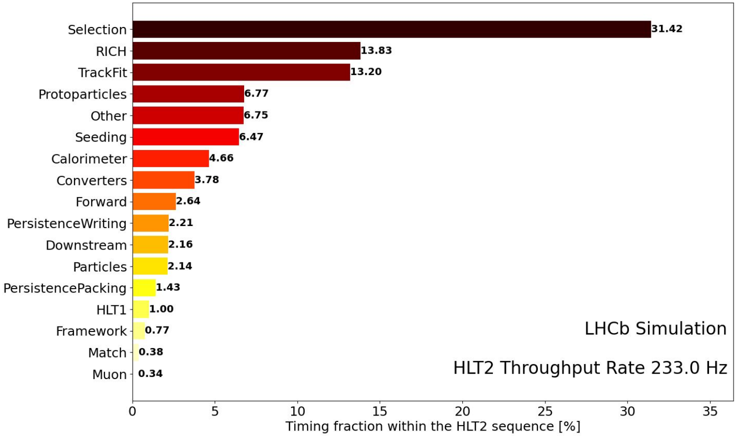
LHCb Run 3 trigger

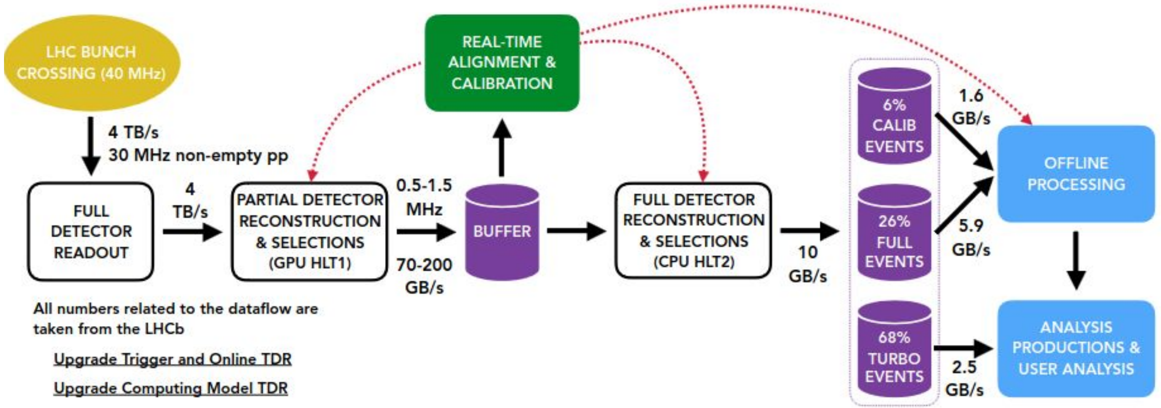
- HLT1: 30 MHz -> 1 MHz
  - GPU-based algorithms focused on tracks, displaced decay vertices and muons
- Alignment & Calibration: 30 PB
  - Event buffer between HLT1 and HLT2 enabling real-time alignment and calibration



LHCb Run 3 trigger

- HLT1: 30 MHz -> 1 MHz
  - GPU-based algorithms focused on tracks, displaced decay vertices and muons
- Alignment & Calibration: 30 PB
  - Event buffer between HLT1 and HLT2 enabling real-time alignment and calibration
- HLT2: ~ 0.5-1 MHz
  - Offline-level reconstruction and selection of physics objects for analysis





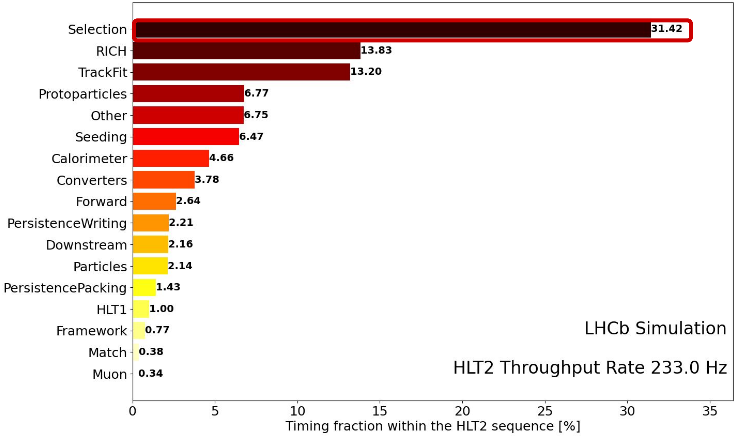
All numbers related to the dataflow are taken from the LHCb

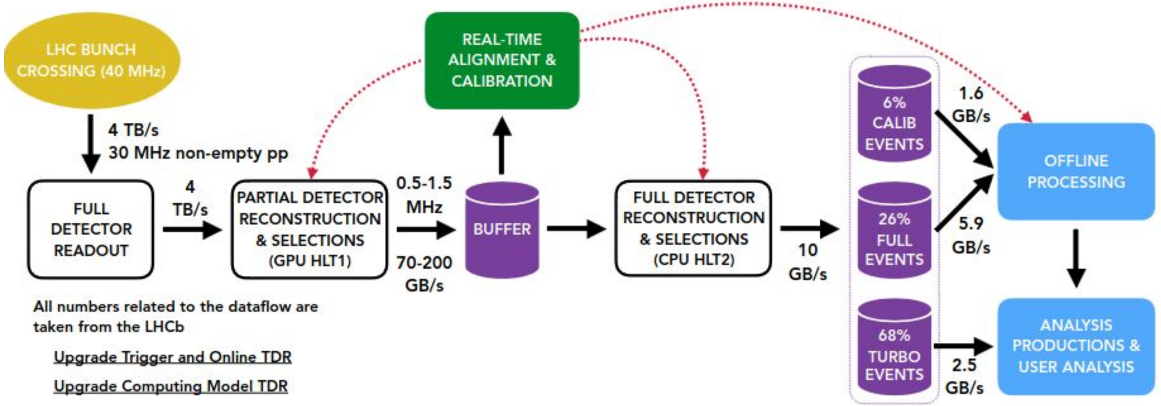
Upgrade Trigger and Online TDR

Upgrade Computing Model TDR

LHCb Run 3 trigger

- HLT1: 30 MHz -> 1 MHz
  - GPU-based algorithms focused on tracks, displaced decay vertices and muons
- Alignment & Calibration: 30 PB
  - Event buffer between HLT1 and HLT2 enabling real-time alignment and calibration
- HLT2: ~ 0.5-1 MHz
  - Offline-level reconstruction and selection of physics objects for analysis





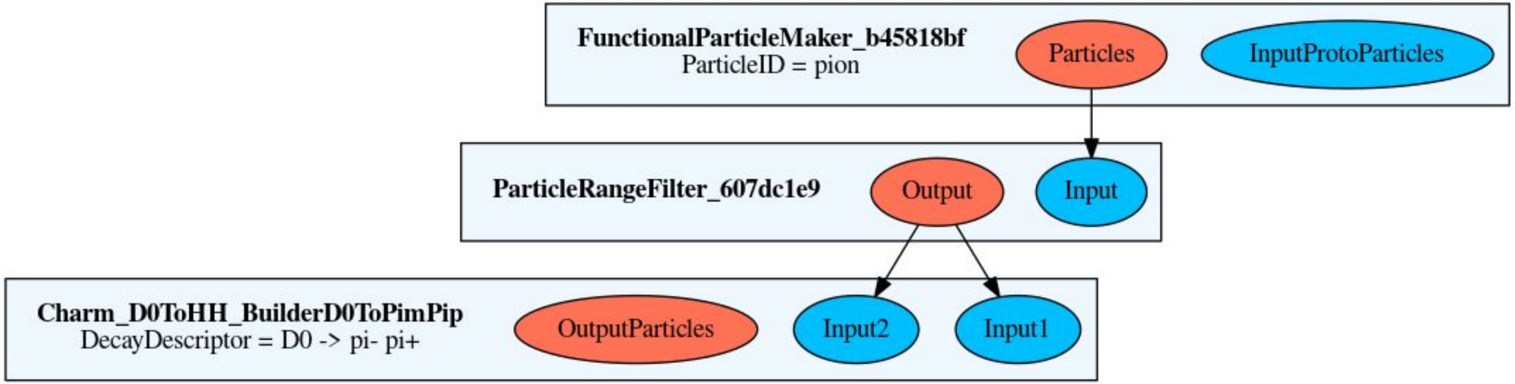
All numbers related to the dataflow are taken from the LHCb

Upgrade Trigger and Online TDR

Upgrade Computing Model TDR

HLT2 trigger selections

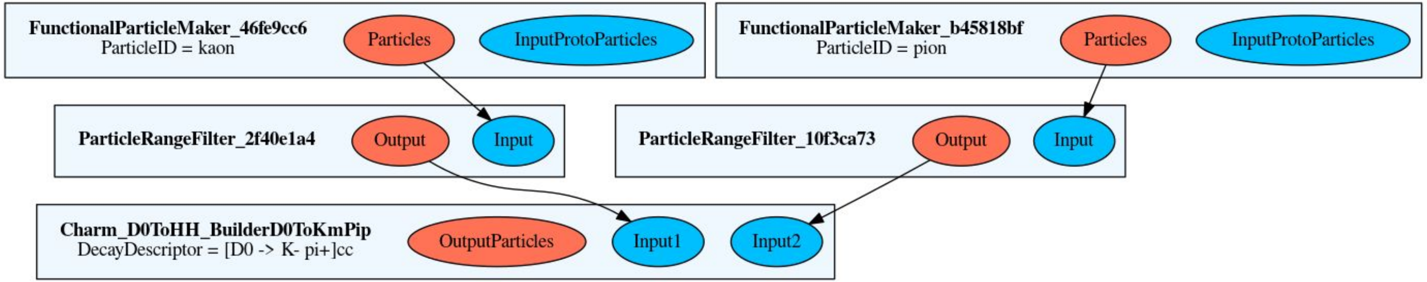
- ~3000 trigger lines currently written
- Line is a collection of selection algorithms
- Lines usually consists of at least
  - Maker: Container of particles coming from reconstruction
  - Filter: Performs cuts on input particles
  - NBodyCombiners: Iterates over combinations of N input particles and performs cuts on the combination



HLT2 control flow optimization

HLT2 trigger control flow

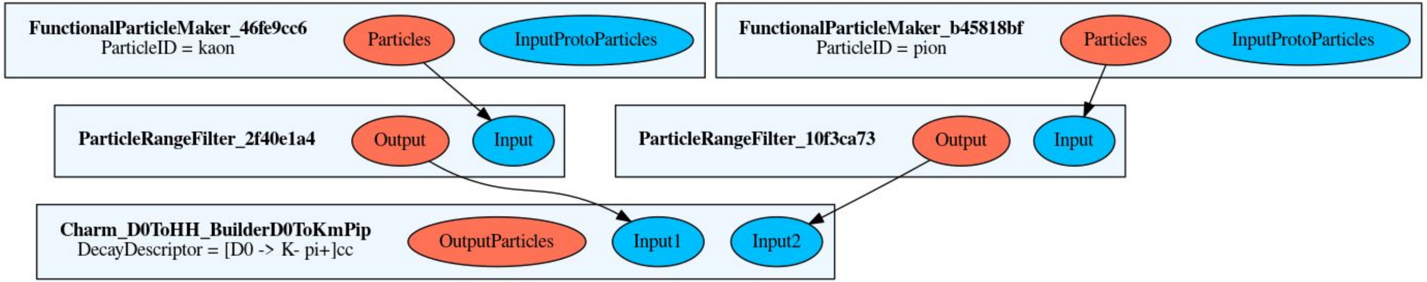
- Control Flow optimization
  - Data flow: The data dependence of trigger line algorithms



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

```python
@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)
```

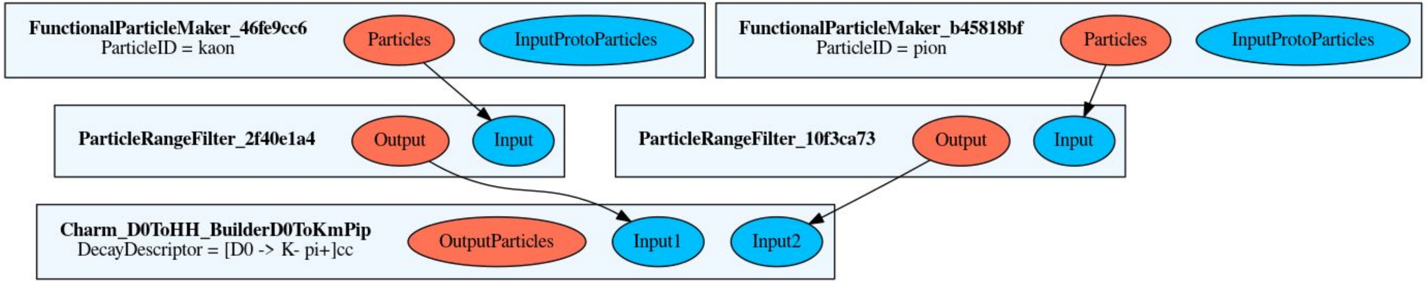


HLT2 trigger control flow

- 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
- Automatic generation
  - Iterate through data flow
    - Add next algorithm to control flow
    - if > 1 option: add the first algorithm
  - Simple but works well because combiner inputs are sorted

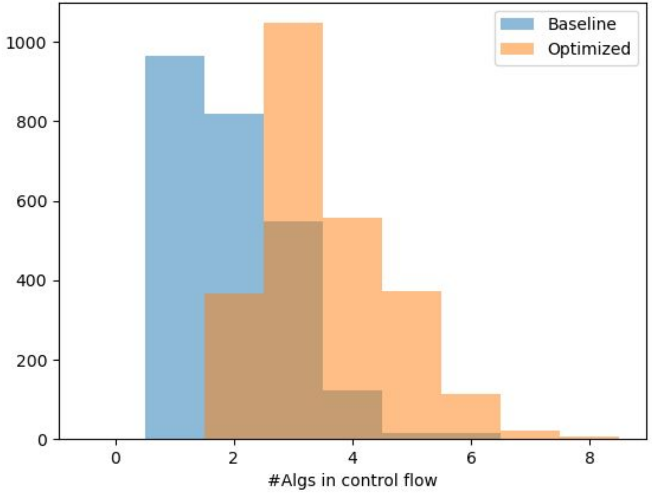
```python
@register_line_builder(all_lines)
def dzero2kpi_line(name='Hlt2Charm_D0ToKmPip', prescale=1):
    kaons = make_charm_kaons()
    pions = make_charm_pions()
    dzeros = make_dzeros(kaons, pions, '[D0 -> K- pi+]cc')
    return Hlt2Line(
        name=name, algs=charm_prefilters() + [dzeros],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

```python
@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)
```



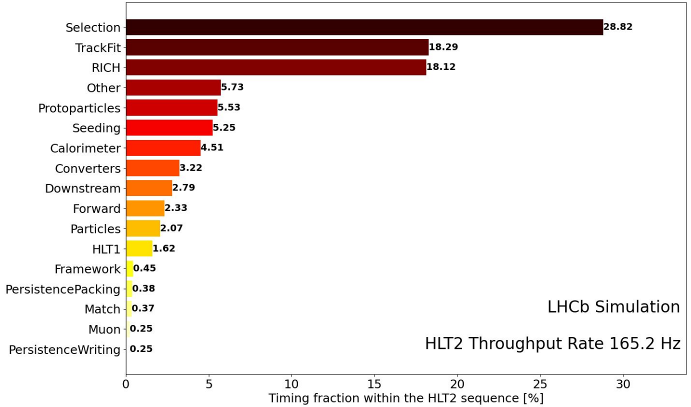
Results

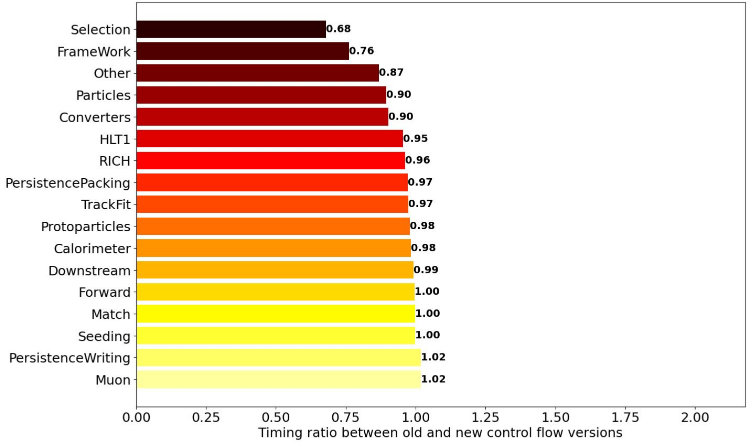
- 92% of control flows altered
  - 2 -> 3.5 average algs per control flow

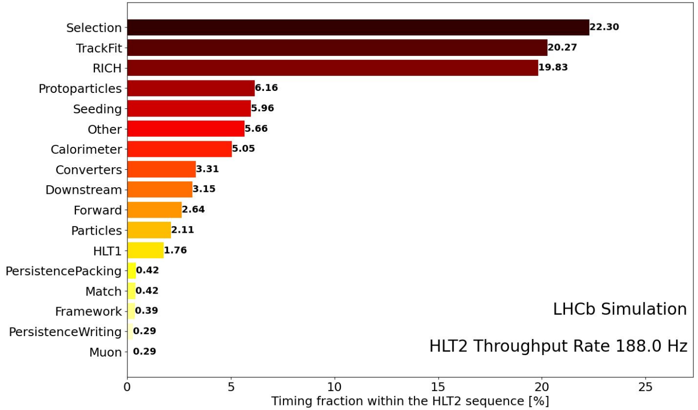


Results

- 92% of control flows altered
  - 2 -> 3.5 average algs per control flow
- LHCb throughput testing of HLT2
- Throughput rate: 165.2 Hz → 188.0 Hz
  - 14% improvement
  - 32% improvement in Selections







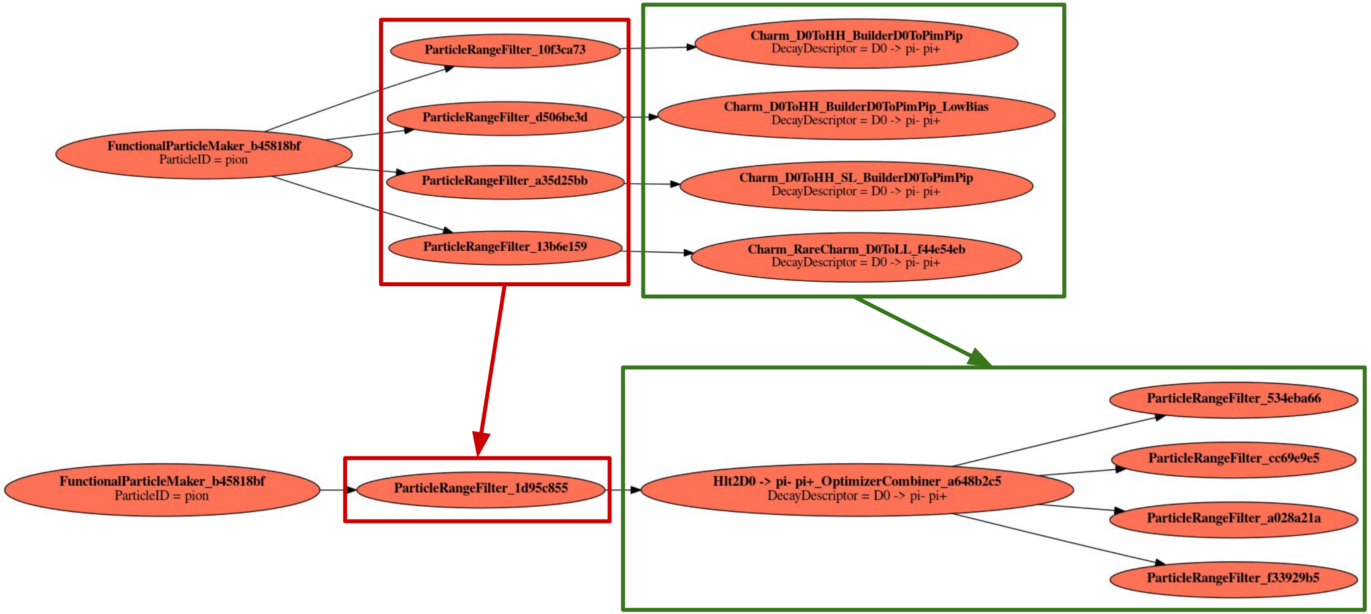
Finally implemented

- Method was merged end of April and activated in data taking shortly after
- Merge request open since December 2023

- This was quite useful as throughput was lower than expected
  - Occupancy in the detector was higher in data than MC
- This optimization together with others helped to increase the throughput to stable data taking levels

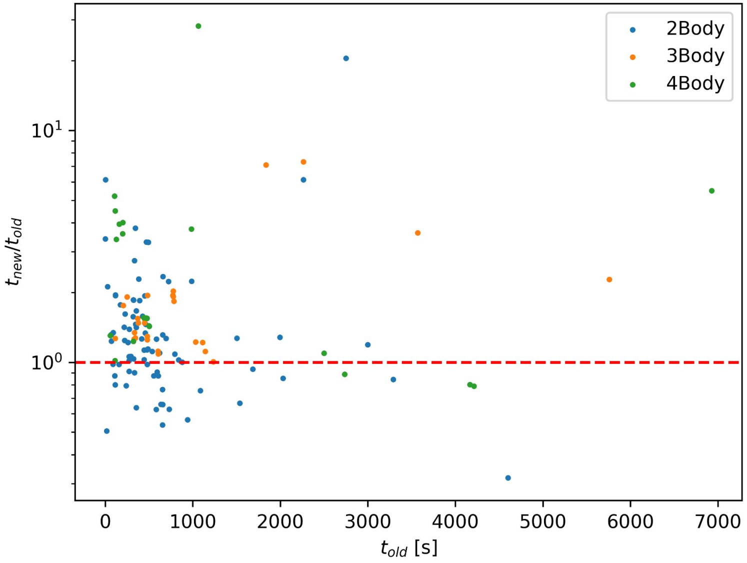
Algorithm overlap

- Last year I presented work on combiner optimization



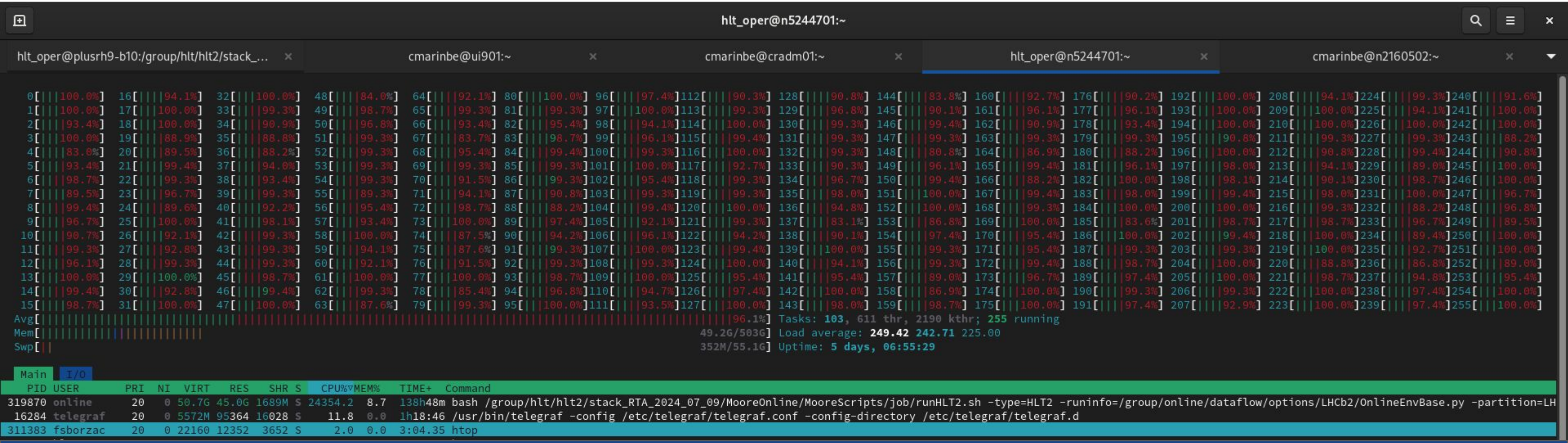
Algorithm overlap

- Last year I presented work on combiner optimization
- Results were not amazing
- Implementation was tricky
  - Creating new line configurations in-situ
  - Framework not made to support it
- Still some interest in approaching this “manually” in LHCb
- I’ve developed overlap testing of algorithms to aid with this
  - Saves input Ids of particles passing an algorithm
  - Enables comparison of algorithms even when the line doesn’t trigger



HLT piquet

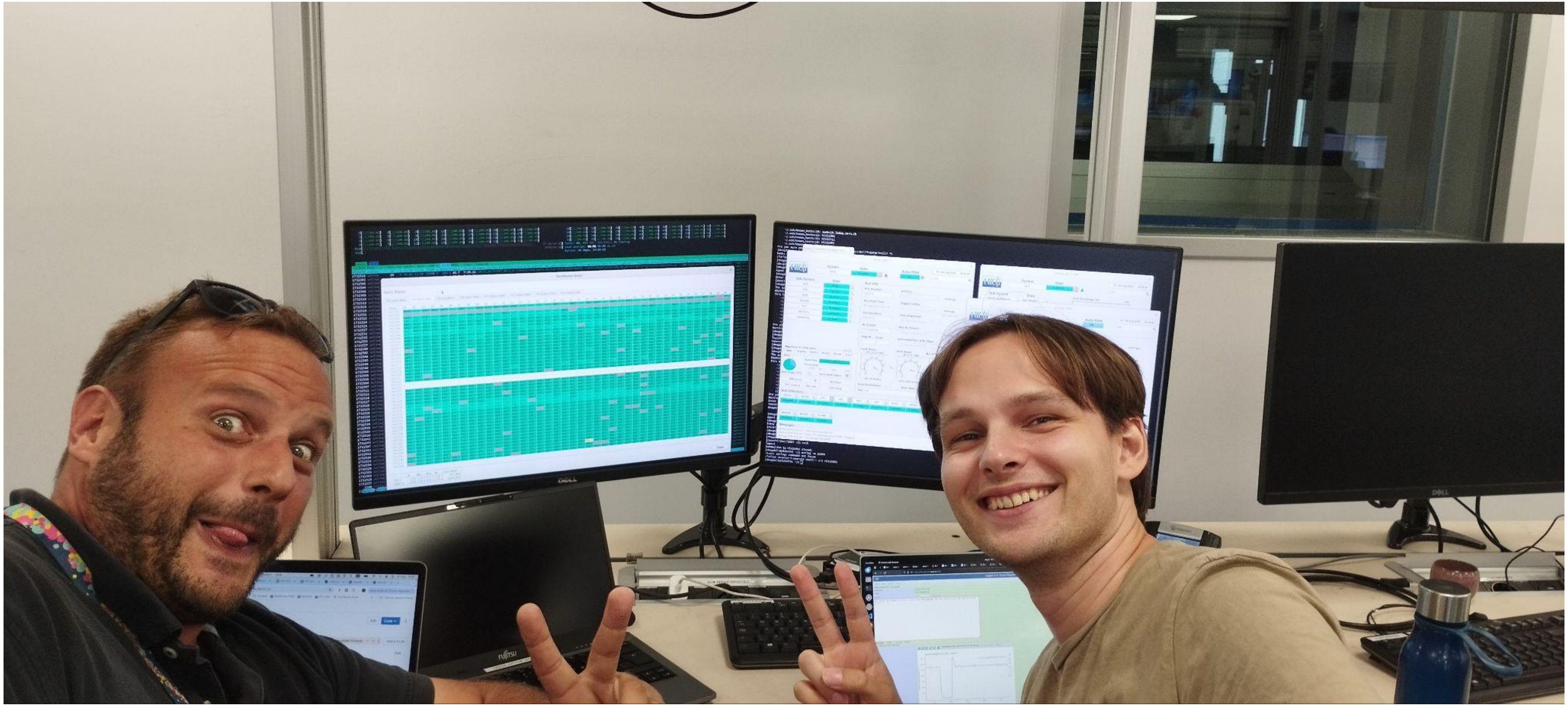
- HLT piquet shifts in August(available 24/7) for two weeks
- Online installed new 256 thread CPUs for HLT2, I was helping to commission them
- CPUs were spending 75% of their time in kernel space…



HLT piquet

- HLT piquet shifts in August(available 24/7) for two weeks
- Online installed new 256 thread CPUs for HLT2, I was helping to commission them
- CPUs were spending 75% of their time in kernel space…
- Together with Online we did some profiling
  - An algorithm had a mutex counter which all threads were queuing to access
  - Mutex counter was moved to only be used in Debug
- With this change the throughput went from 400 Hz -> 1400 Hz
- After some further optimizing HLT2 reached 1 MHz of input rate!

HLT2 reaching 1 MHz!!



Analysis: DeltaGammaD

- Difference in decay width between B0 mass states
  - A “Null test” of the Standard Model
- Experiment: $\frac{\triangle\Gamma_d}{\Gamma_d} = 0.001 \pm 0.01$, Theory: $\frac{\triangle\Gamma_d}{\Gamma_d} = 0.00397 \pm 0.00090$
- Experimental method: Compare decay time distributions of
  - Bd -> J/ψ + K*(flavour-specific)
  - Bd -> J/ψ + KS(CP specific)
- Re-activating Run 2 measurement with Master student
- Preparations for Run 3 measurement

Conclusions & Outlook

- Control flow optimization finally finished and in production
  - ~15% gain for HLT2
- Continued work on trigger optimization
  - Overlap work continues manually
- Analysis work ramping up on DeltaGammaD measurement
- Future:
  - Point 8 secondment: October - December
  - Focus shifting to DeltaGammaD analysis

Conclusions & Outlook

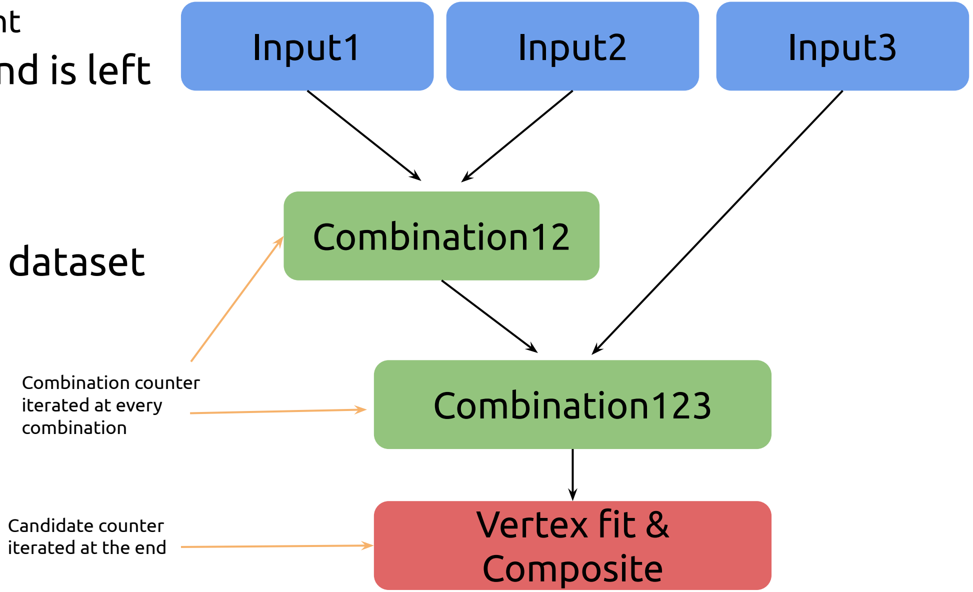
- Control flow optimization finally finished and in production
  - ~15% gain for HLT2
- Continued work on trigger optimization
  - Overlap work continues manually
- Analysis work ramping up on DeltaGammaD measurement
- Future:
  - Point 8 secondment: October - December
  - Focus shifting to DeltaGammaD analysis

Thank you for your attention!

Backup

Combiner limits

- Combinations scale by N^M, MBodyCombiner, N size of Input
- Combination is capped by limits:
  - **2K** candidates and **50M** combinations per event
- If limit is hit the Combiner stops running and is left with whatever it generated so far
  - Creates a difficult inefficiency to handle
  - A sign of a inefficient combiner
- I created a test that runs over a extra busy dataset
  - Track worst offenders
  - Benchmark combiners on this issue

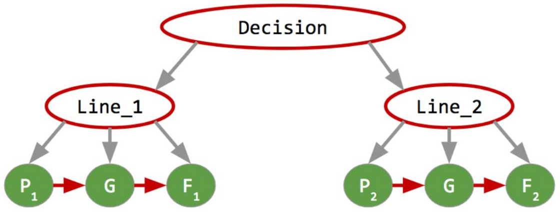


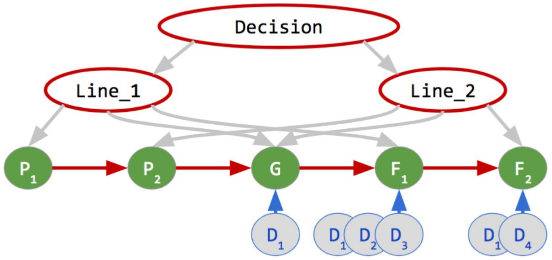
Miscellaneous

- Teachers assistant:
  - CP violation course
- LHCb hackathon
- Poster at NWO Veldhoven: “Real-time analysis at the LHCb experiment”
- EuCAIF
  - Part of conference organizing team
- Future collider lectures and workshop
- SMARTHEP outreach hackathon
- SMARTHEP Edge ML school

Circular control flow

- Control flow is scheduled globally
  - Meaning that algorithms have to be ordered globally
  - Trigger lines have to agree on order of algorithms
- This is difficult to take into account
  - Therefore we just want to avoid it
  - If one would only pick one algorithm per data flow iteration we would ensure no circular control flow





Nolte N. A Selection Framework for LHCb's Upgrade Trigger. TU Dortmund U., 2020.

```
@register_line_builder(all_lines)
def dzero2kpi_line(name='Hlt2Charm_D0ToKmPip', prescale=0.2):
    kaons = make_charm_kaons()
    pions = make_charm_pions()
    dzeros = make_dzeros(kaons, pions, '[D0 -> K- pi+]cc')
    return Hlt2Line(
        name=name,
        algs=charm_prefilters() + [pions,kaons,dzeros],
        prescale=prescale,
```

```
def dstarp2dzeropip_dzero2kppim_line(name='Hlt2Charm_DstpToD0Pip_D0ToKpPim',
                                     prescale=1):
    kaons = make_charm_kaons()
    pions = make_charm_pions()
    dzeros = make_dzeros(kaons, pions, '[D0 -> K+ pi-]cc')
    dstars = make_dstars(
        dzeros, self_conjugate_d0_decay=False, d0_name="D0ToHH_D0ToKpPim")
    return Hlt2Line(
        name=name,
        algs=charm_prefilters() + [kaons,pions,dzeros,dstars],
        prescale=prescale,
        #extra_outputs=isolation.make_iso_particles(dstars, coneangle=0.5)
    )
```