

ESR6: Optimization of HLT2 selections at the LHCb experiment

Daniel Magdalinski
01 October 2024

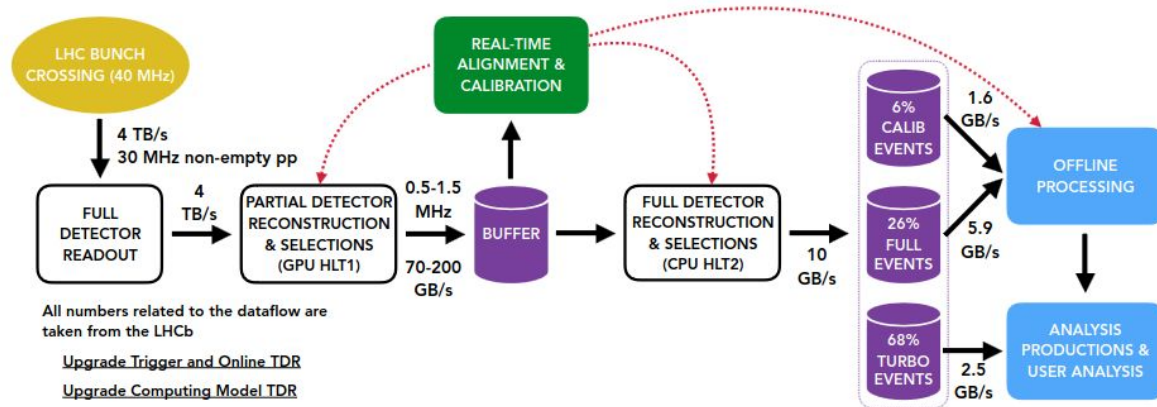


SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086



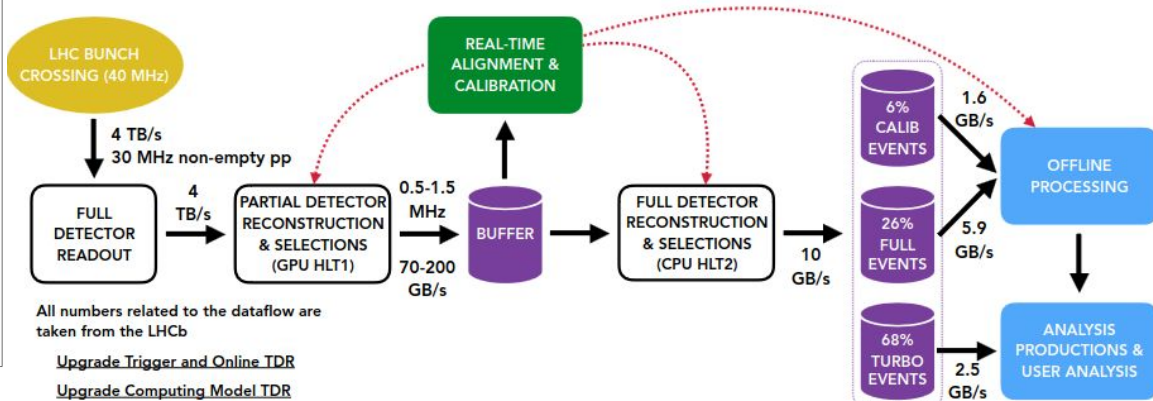
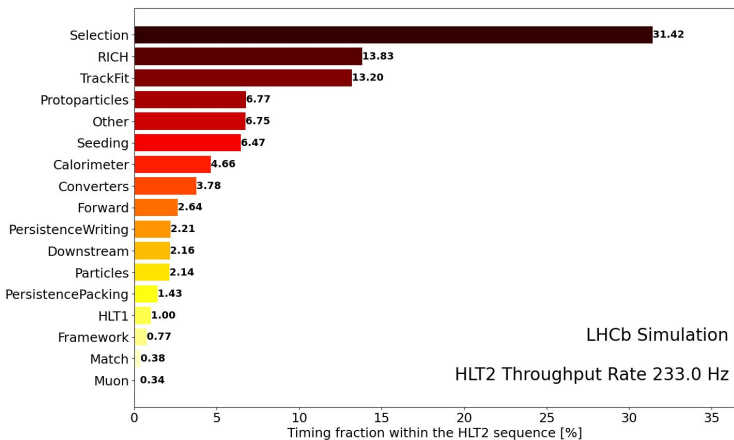
LHCb Run 3 trigger

- HLT1: 30 MHz \rightarrow 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



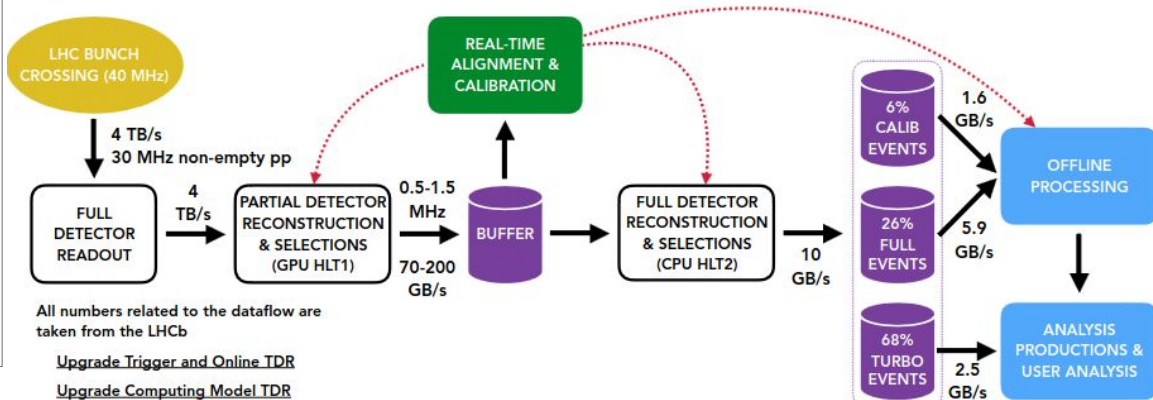
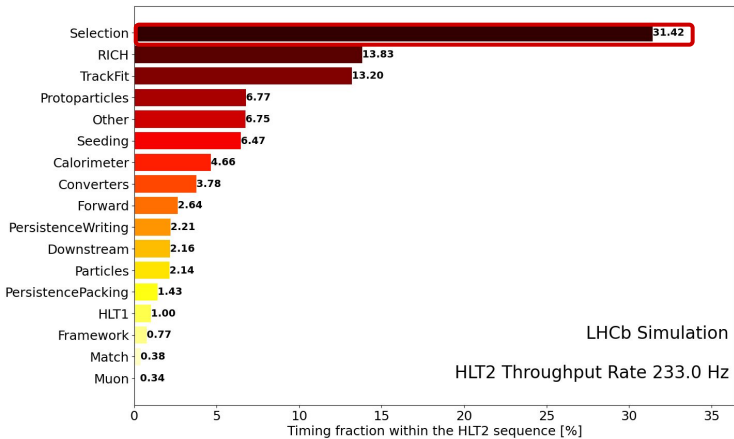
LHCb Run 3 trigger

- 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



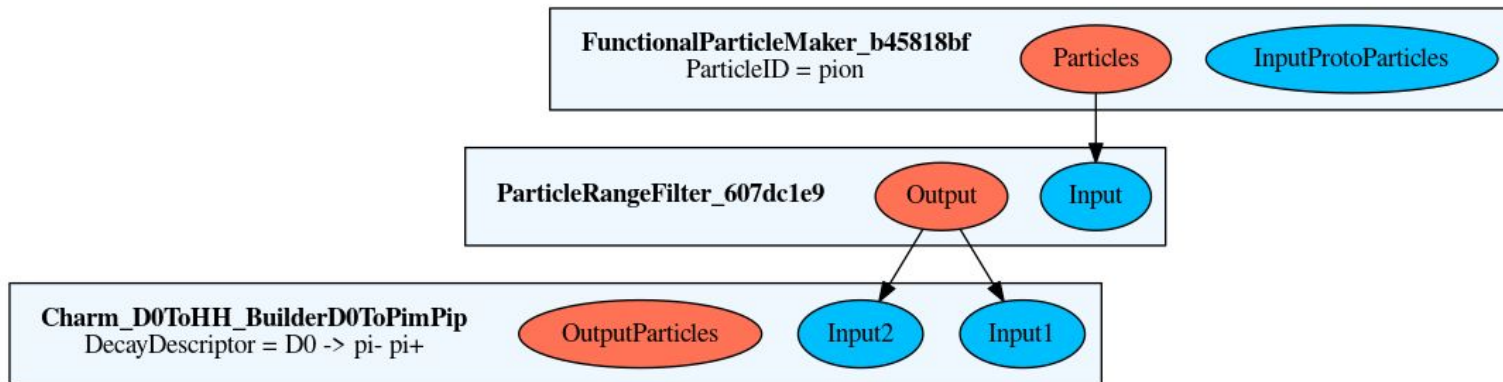
LHCb Run 3 trigger

- 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



HLT2 trigger selections

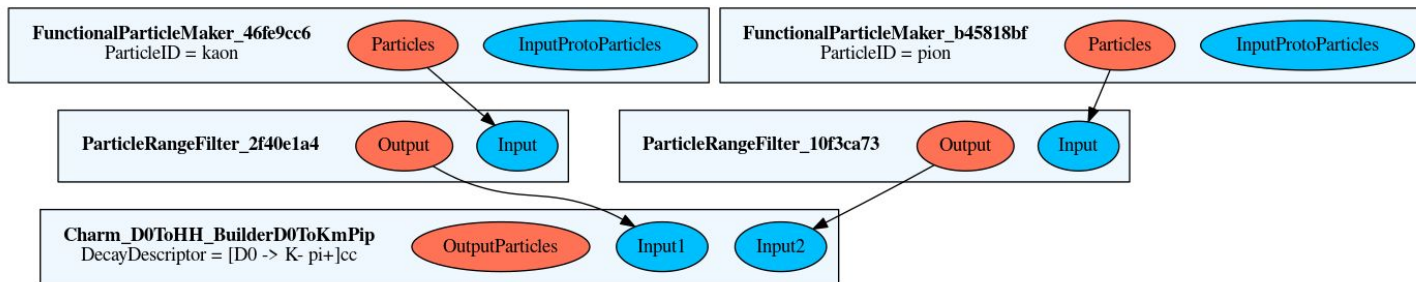
- ~3000 trigger lines currently written
- Line is a collection of selection algorithms
- Lines usually consists of at least
 - Maker: Container of particles coming from reconstruction
 - Filter: Performs cuts on input particles
 - NBodyCombiners: Iterates over combinations of N input particles and performs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



HLT2 control flow optimization

HLT2 trigger control flow

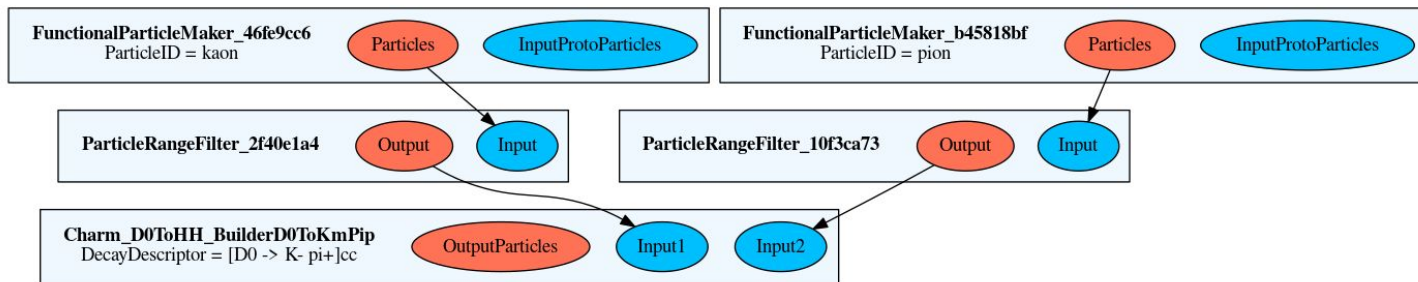
- Control Flow optimization
 - Data flow: The data dependence of trigger line algorithms



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

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



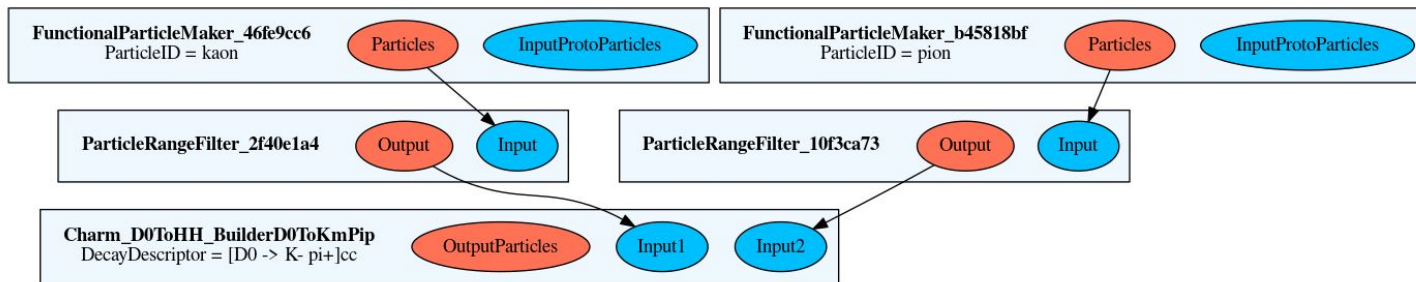
HLT2 trigger control flow

- 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
- 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

```
@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)
```

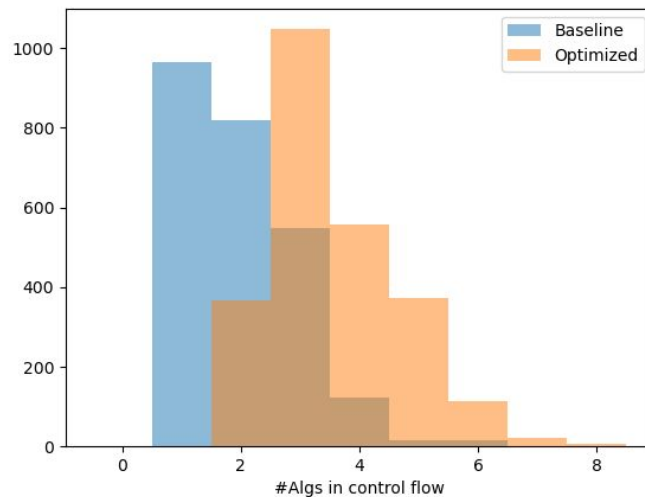


```
@register_line_builder(all_lines)
def dzero2kpi_line(name='Hlt2Charm_D0ToKmPip', prescale=1):
    kaons = make_charm_kaons()
    pions = make_charm_pions()
    dzeros = make_dzeros(kaons, pions, '[D0 -> K- pi+]'cc')
    return Hlt2Line(
        name=name, algs=charm_prefilters() + [kaons,dzeros],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



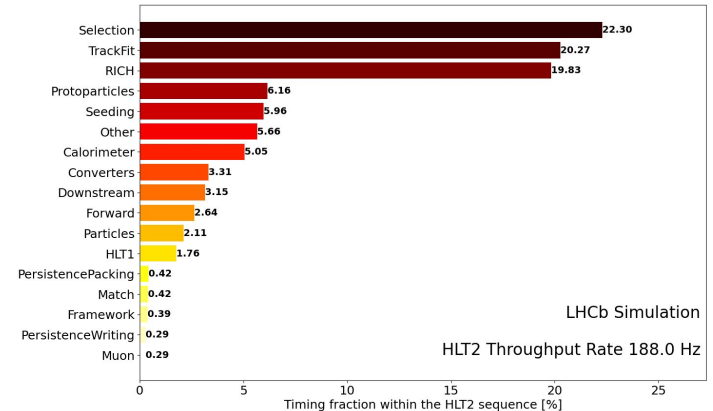
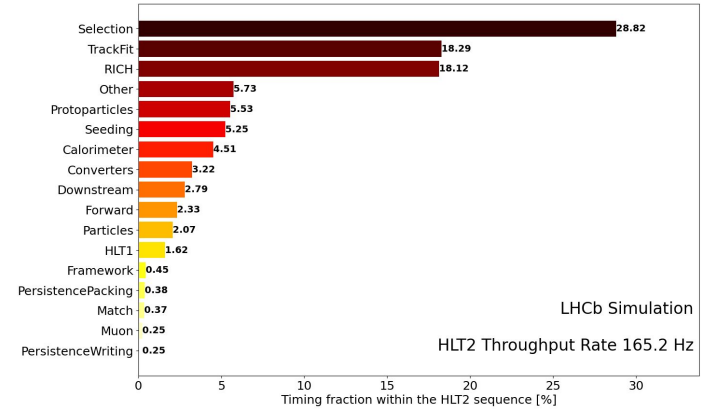
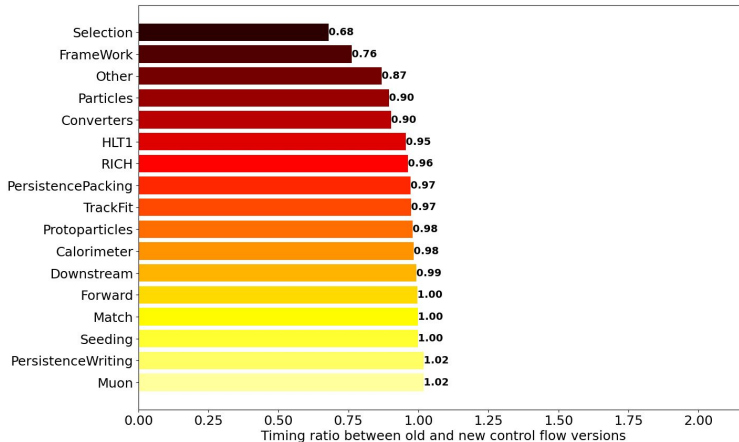
Results

- 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

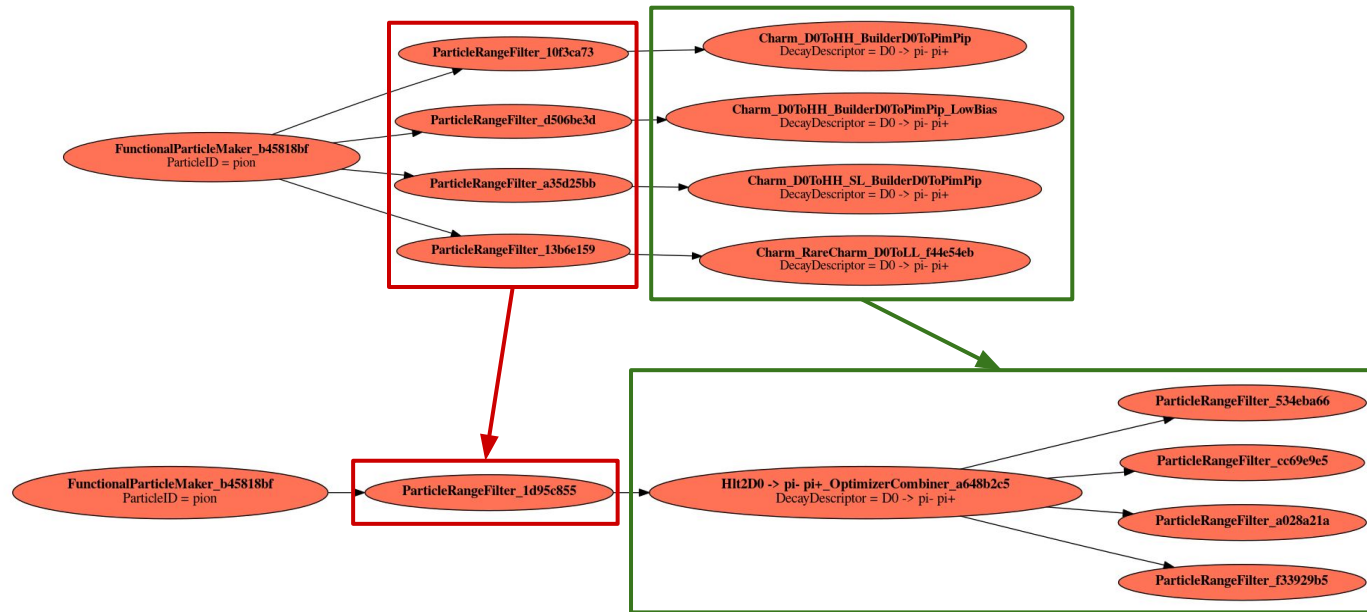


The screenshot shows a GitHub pull request titled "Optimized control flow" by user lncb/Moore!2757, created 10 months ago. It is part of the 2024-patches branch. The pull request is merged and approved. It includes several labels: PR/Moore_hit2_and_spruce_bandwidth, PR/full_throughput_test, RTA, ci-test-triggered, and hit2. The pull request has 133 comments and 8 likes, and was updated 4 months ago.

- 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

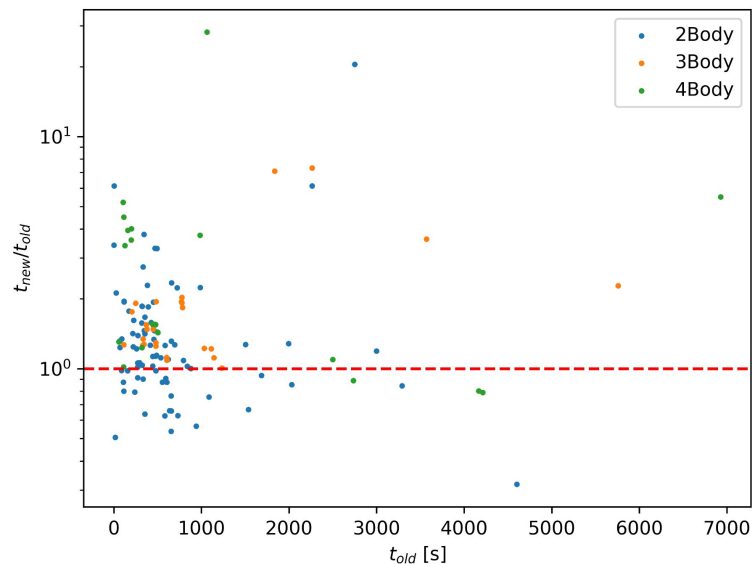
Algorithm overlap

- 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

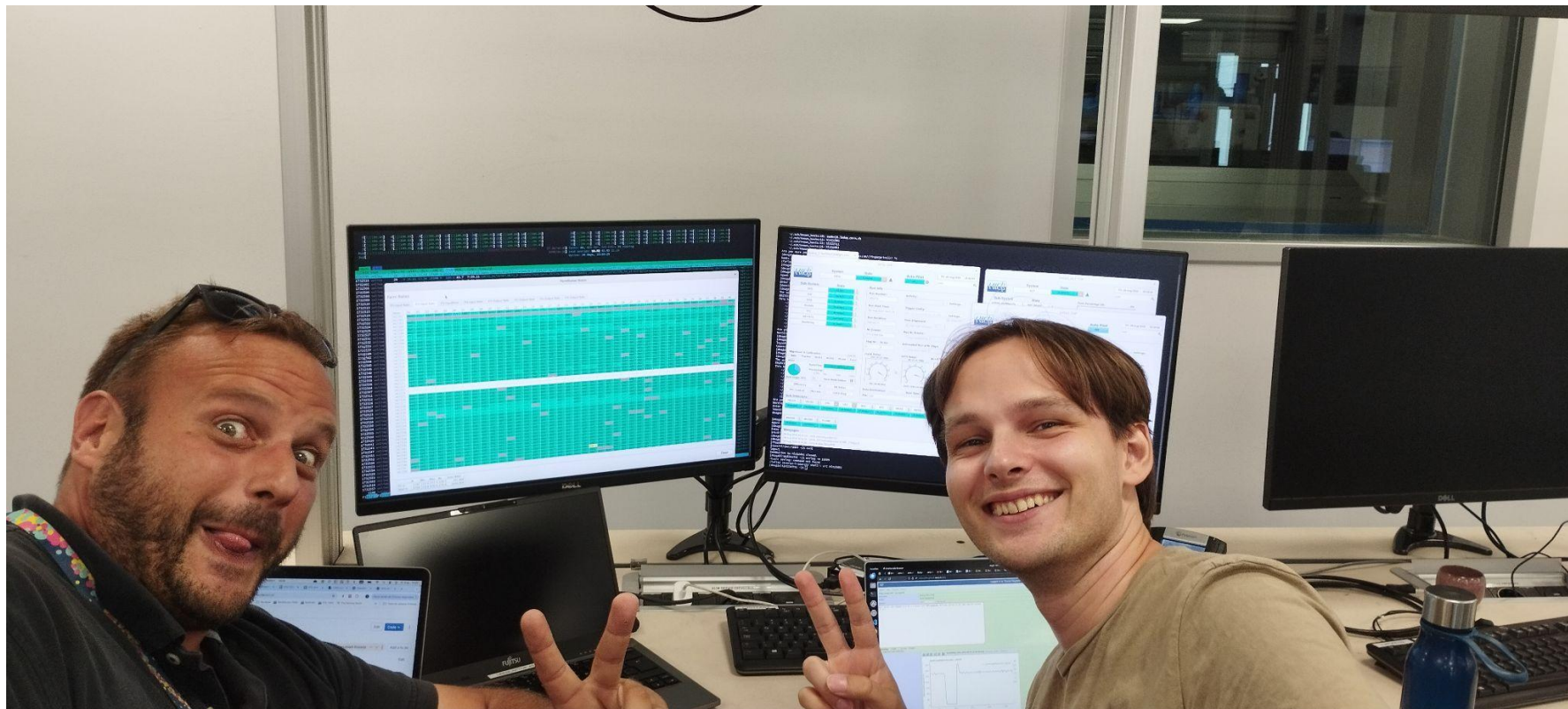
- HLT piquet shifts in August(available 24/7) for two weeks
- Online installed new 256 thread CPUs for HLT2, is helping to commission them
- CPUs were spending 75% of their time in kernel space...

```
htl_oper@n5244701:~  
htl_oper@plush9-b10:/group/hlt/hlt2/stack_... x cmarinbe@ui901:~ x cmarinbe@cradm01:~ x htl_oper@n5244701:~ x cmarinbe@n2160502:~  
0 [ | 100.0% ] 16 [ | 94.1% ] 32 [ | 100.0% ] 48 [ | 84.0% ] 64 [ | 92.1% ] 80 [ | 100.0% ] 96 [ | 97.4% ] 112 [ | 90.3% ] 128 [ | 90.8% ] 144 [ | 83.8% ] 160 [ | 92.7% ] 176 [ | 90.2% ] 192 [ | 100.0% ] 208 [ | 94.1% ] 224 [ | 99.3% ] 240 [ | 91.6% ]  
1 [ | 100.0% ] 17 [ | 100.0% ] 33 [ | 99.3% ] 49 [ | 98.7% ] 65 [ | 99.3% ] 81 [ | 99.3% ] 97 [ | 100.0% ] 113 [ | 99.3% ] 129 [ | 96.8% ] 145 [ | 99.1% ] 161 [ | 96.1% ] 177 [ | 96.1% ] 193 [ | 100.0% ] 209 [ | 100.0% ] 225 [ | 94.1% ] 241 [ | 100.0% ]  
2 [ | 93.4% ] 18 [ | 100.0% ] 34 [ | 90.9% ] 50 [ | 96.8% ] 66 [ | 93.4% ] 82 [ | 95.4% ] 98 [ | 94.1% ] 114 [ | 100.0% ] 130 [ | 99.3% ] 146 [ | 99.4% ] 162 [ | 90.9% ] 178 [ | 93.4% ] 194 [ | 100.0% ] 210 [ | 100.0% ] 226 [ | 100.0% ] 242 [ | 100.0% ]  
3 [ | 100.0% ] 19 [ | 88.9% ] 35 [ | 88.8% ] 51 [ | 99.3% ] 67 [ | 83.7% ] 83 [ | 98.7% ] 99 [ | 96.1% ] 115 [ | 99.4% ] 131 [ | 99.3% ] 147 [ | 99.3% ] 163 [ | 99.3% ] 179 [ | 99.3% ] 195 [ | 90.8% ] 211 [ | 99.3% ] 227 [ | 99.3% ] 243 [ | 88.2% ]  
4 [ | 83.6% ] 20 [ | 89.5% ] 36 [ | 88.2% ] 52 [ | 99.3% ] 68 [ | 95.4% ] 84 [ | 99.4% ] 100 [ | 99.3% ] 116 [ | 100.0% ] 132 [ | 99.3% ] 148 [ | 89.8% ] 164 [ | 86.9% ] 180 [ | 89.2% ] 196 [ | 100.0% ] 212 [ | 98.8% ] 228 [ | 99.4% ] 244 [ | 99.8% ]  
5 [ | 93.4% ] 21 [ | 99.4% ] 37 [ | 94.0% ] 53 [ | 99.3% ] 69 [ | 99.3% ] 85 [ | 99.3% ] 101 [ | 100.0% ] 117 [ | 92.7% ] 133 [ | 90.3% ] 149 [ | 96.1% ] 165 [ | 99.4% ] 181 [ | 96.1% ] 197 [ | 98.0% ] 213 [ | 94.1% ] 229 [ | 89.0% ] 245 [ | 100.0% ]  
6 [ | 98.7% ] 22 [ | 99.3% ] 38 [ | 93.4% ] 54 [ | 99.3% ] 70 [ | 91.5% ] 86 [ | 99.3% ] 102 [ | 95.4% ] 118 [ | 99.3% ] 134 [ | 96.7% ] 150 [ | 99.4% ] 166 [ | 88.2% ] 182 [ | 100.0% ] 198 [ | 98.1% ] 214 [ | 90.1% ] 230 [ | 98.7% ] 246 [ | 100.0% ]  
7 [ | 89.5% ] 23 [ | 96.7% ] 39 [ | 99.3% ] 55 [ | 89.3% ] 71 [ | 94.1% ] 87 [ | 90.8% ] 103 [ | 99.3% ] 119 [ | 99.3% ] 135 [ | 98.0% ] 151 [ | 100.0% ] 167 [ | 99.4% ] 183 [ | 98.0% ] 199 [ | 99.4% ] 215 [ | 98.0% ] 231 [ | 100.0% ] 247 [ | 96.7% ]  
8 [ | 99.4% ] 24 [ | 89.6% ] 40 [ | 92.2% ] 56 [ | 95.4% ] 72 [ | 98.7% ] 88 [ | 88.2% ] 104 [ | 99.4% ] 120 [ | 100.0% ] 136 [ | 94.8% ] 152 [ | 100.0% ] 168 [ | 99.3% ] 184 [ | 100.0% ] 200 [ | 100.0% ] 216 [ | 99.3% ] 232 [ | 88.2% ] 248 [ | 96.8% ]  
9 [ | 96.7% ] 25 [ | 100.0% ] 41 [ | 98.1% ] 57 [ | 93.4% ] 73 [ | 100.0% ] 89 [ | 97.4% ] 105 [ | 92.1% ] 121 [ | 99.3% ] 137 [ | 83.1% ] 153 [ | 86.8% ] 169 [ | 100.0% ] 185 [ | 83.6% ] 201 [ | 98.7% ] 217 [ | 98.7% ] 233 [ | 96.7% ] 249 [ | 89.5% ]  
10 [ | 90.7% ] 26 [ | 92.1% ] 42 [ | 99.3% ] 58 [ | 100.0% ] 74 [ | 87.5% ] 90 [ | 94.2% ] 106 [ | 96.1% ] 122 [ | 94.2% ] 138 [ | 90.1% ] 154 [ | 97.4% ] 170 [ | 95.4% ] 186 [ | 100.0% ] 202 [ | 99.4% ] 218 [ | 100.0% ] 234 [ | 89.4% ] 250 [ | 100.0% ]  
11 [ | 99.3% ] 27 [ | 92.8% ] 43 [ | 99.3% ] 59 [ | 94.1% ] 75 [ | 87.6% ] 91 [ | 99.3% ] 107 [ | 100.0% ] 123 [ | 99.4% ] 139 [ | 100.0% ] 155 [ | 99.3% ] 171 [ | 95.4% ] 187 [ | 99.3% ] 203 [ | 99.3% ] 219 [ | 100.0% ] 235 [ | 92.7% ] 251 [ | 100.0% ]  
12 [ | 96.1% ] 28 [ | 99.3% ] 44 [ | 99.3% ] 60 [ | 92.1% ] 76 [ | 91.5% ] 92 [ | 99.3% ] 108 [ | 99.3% ] 124 [ | 100.0% ] 140 [ | 94.1% ] 156 [ | 99.3% ] 172 [ | 99.4% ] 188 [ | 98.7% ] 204 [ | 100.0% ] 220 [ | 100.0% ] 236 [ | 86.8% ] 252 [ | 89.0% ]  
13 [ | 100.0% ] 29 [ | 100.0% ] 45 [ | 98.7% ] 61 [ | 100.0% ] 77 [ | 100.0% ] 93 [ | 98.7% ] 109 [ | 100.0% ] 125 [ | 95.4% ] 141 [ | 95.4% ] 157 [ | 89.0% ] 173 [ | 96.7% ] 189 [ | 97.4% ] 205 [ | 100.0% ] 221 [ | 98.7% ] 237 [ | 94.8% ] 253 [ | 95.4% ]  
14 [ | 99.4% ] 30 [ | 92.8% ] 46 [ | 99.4% ] 62 [ | 99.3% ] 78 [ | 85.4% ] 94 [ | 96.8% ] 110 [ | 94.7% ] 126 [ | 97.4% ] 142 [ | 100.0% ] 158 [ | 86.9% ] 174 [ | 100.0% ] 190 [ | 99.3% ] 206 [ | 99.3% ] 222 [ | 100.0% ] 238 [ | 97.4% ] 254 [ | 100.0% ]  
15 [ | 98.7% ] 31 [ | 100.0% ] 47 [ | 100.0% ] 63 [ | 87.6% ] 79 [ | 99.3% ] 95 [ | 100.0% ] 111 [ | 93.5% ] 127 [ | 100.0% ] 143 [ | 98.0% ] 159 [ | 98.7% ] 175 [ | 100.0% ] 191 [ | 97.4% ] 207 [ | 92.9% ] 223 [ | 100.0% ] 239 [ | 97.4% ] 255 [ | 100.0% ]  
Avg [ | 96.1% ] Tasks: 103, 611 thr, 2190 kthr; 255 running  
Mem [ | 49.26/5036 ] Load average: 249.42 242.71 225.00  
Swp [ | 352M/55.16 ] Uptime: 5 days, 06:55:29  
Main | T/O  
PID USER PRI NI VIRT RES SHR S CPU%MEM% TIME+ Command  
319870 online 20 0 50.7G 45.0G 1689M S 24354.2 8.7 138h48m bash /group/hlt/hlt2/stack_RTA_2024_07_09/MooreOnline/MooreScripts/job/runHLT2.sh -type=HLT2 -runinfo=/group/online/dataflow/options/LHCb2/OnlineEnvBase.py -partition-LH  
16284 telegraf 20 0 5572M 95364 10028 S 11.8 0.0 1h18:46 /usr/bin/telegraf -config /etc/telegraf/telegraf.conf -config-directory /etc/telegraf/telegraf.d  
311383 fsborzac 20 0 22160 12352 3652 S 2.0 0.0 3:04.35 htop
```

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: $\frac{\Delta\Gamma_d}{\Gamma_d} = 0.001 \pm 0.01$, Theory: $\frac{\Delta\Gamma_d}{\Gamma_d} = 0.00397 \pm 0.00090$
- Experimental method: Compare decay time distributions of
 - $B_d \rightarrow J/\psi + K^*$ (flavour-specific)
 - $B_d \rightarrow J/\psi + K_S$ (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

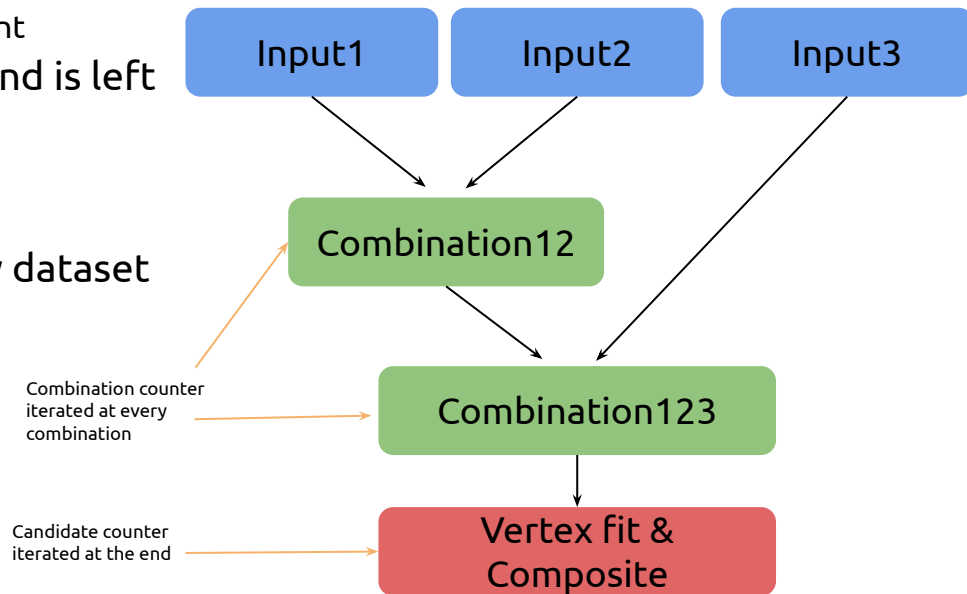
- Control flow optimization finally finished and in production
 - ~15% gain for HLT2
- Continued work on trigger optimization
 - Overlap work continues manually
- Analysis work ramping up on DeltaGammaD measurement
- Future:
 - Point 8 secondment: October - December
 - Focus shifting to DeltaGammaD analysis

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

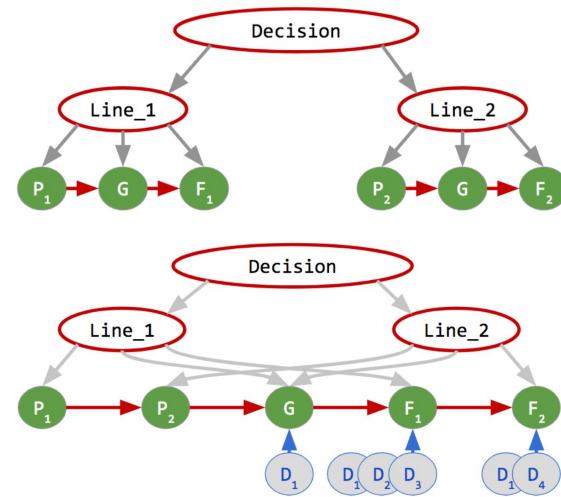


Miscellaneous

- 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)
    )
```