A comprehensive real-time analysis model at the LHCb experiment

Alex Pearce

Wednesday 20th March 2019
2019 Joint HSF/OSG/WLCG Workshop
A comprehensive real-time analysis model at the LHCb experiment

LHCb-DP-2019-002 released this month

- Describes evolution of our RTA model during Run 2
- LHCb has profited enormously from RTA
  - But how?
  - Was it necessary?
  - What did we learn?
  - Can it be improved?
  - What are the implications for Run 3?
What is real-time analysis, exactly?
What is real-time analysis, exactly?

Source: Manki Kim
What is real-time analysis, exactly?

Source: Manki Kim

Source: Petr Brož
What is real-time analysis, exactly?
What is real-time analysis, exactly?

- **Hard**: Missing a deadline is a total system failure.
- **Firm**: Infrequent deadline misses are tolerable, but may degrade the system’s quality of service. The usefulness of a result is zero after its deadline.
- **Soft**: The usefulness of a result degrades after its deadline, thereby degrading the system’s quality of service.

— Wikipedia: Real-time computing
Here, ‘real time’ is defined as the interval between a collision occurring and the point at which the corresponding event must be either discarded forever or sent offline for permanent storage.
Let’s agree to disagree

1. Real time analysis is the **set of actions we apply to events that may later be discarded forever**.

2. Can also be the **analysis of that information later**, i.e. “offline”.

Real-time analysis makes everything amazing

Source: xkcd: Imposter (edited)
Real-time analysis makes everything amazing

Source: xkcd: Imposter (edited)
Real-time analysis makes everything amazing

Source: xkcd: Imposter (edited)
Motivation

- Triggering is expensive; must fit within computing constraints

\[
\text{Bandwidth [GB/s]} \propto \text{Accept rate [kHz]} \times \text{Event size [kB]}
\]

- Want highest accept rate high to maximise $\varepsilon_{\text{Sig.}}$ and reduce bias
  - Balanced against maximising $1 - \varepsilon_{\text{Bkg.}}$

- Can’t do much to reduce the raw event size; it’s all or nothing!
Motivation

- Triggering is expensive; must fit within computing constraints

\[
\text{Bandwidth [GB/s]} \propto \text{Accept rate [kHz]} \times \text{Event size [kB]}
\]

- Want highest accept rate high to maximise $\varepsilon_{\text{Sig.}}$ and reduce bias
  - Balanced against maximising $1 - \varepsilon_{\text{Bkg.}}$

- Can’t do much to reduce the raw event size; it’s all or nothing!

  **If event size is reduced, there’s room for more physics!**
The idea

- We design our trigger systems to compute useful information
- We do analysis in real-time to compute the trigger decision
- What if that information is good enough to also use directly ‘offline’?
The idea

- We design our trigger systems to compute useful information
- We do analysis in real-time to compute the trigger decision
- What if that information is *good enough* to also use directly ‘offline’?

**Save only what you need, as computed in the trigger!**
The Run 2 trigger

Formidable challenge

- Production rate of around 45 kHz for $b\bar{b}$ events, 1 MHz for $c\bar{c}$
  - Must operate quickly but was high fidelity and selectivity
- Filters proton-proton collision rate down to around 12.5 kHz
  - Bandwidth of around 0.7 GB/s
- Events buffered between split software stages
- Buffered data aligned and calibrated online
- Offline-quality reconstruction in final stage
What this buys us

- Offline-equivalent, fully aligned and calibrated physics objects in HLT2
- Can include offline selections in the trigger with no associated systematic effects
- Offline reprocessing of the raw data is not *necessary* to recover information

**Real-time analysis with offline-quality physics objects**
Enter: Turbo

- Persist **objects from HLT2** directly, analyse only these offline
- Individual trigger selections have **complete control** over what objects are saved
  - Evolved over time to meet increasing needs
Persistence granularity

Raw banks: VELO RICH \cdots ECAL \cdots
Persistence granularity

Raw banks: VELO, RICH, ⋯, ECAL, ⋯
Persistence granularity

Raw banks: VELO RICH ⋯ ECAL ⋯

HLT2 candidate

$D^0 \rightarrow K^- \pi^+$
Persistence granularity
Internals

**HLT2**
- Reconstruct physics candidates
- Serialise and compress selected objects
- Write serialisation to raw banks

**TESSA**
- Persist offline-compatible file format
- Extract event-level counters
- Decode and deserialise raw data

Offline storage
Rewards

- Much smaller average event size → more physics within our resources

<table>
<thead>
<tr>
<th>Persistence method</th>
<th>Average event size (kB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turbo</td>
<td>7</td>
</tr>
<tr>
<td>Selective persistence</td>
<td>16</td>
</tr>
<tr>
<td>Complete persistence</td>
<td>48</td>
</tr>
<tr>
<td>Raw event</td>
<td>69</td>
</tr>
</tbody>
</table>

- Accounted for around one quarter of the trigger rate in Run 2
  - For one tenth of the bandwidth!
What we’re able achieve

What we’re able achieve

Looking back

- Must overcome fear of losing information
- There’s always room for improvement
  - Selective persistence allowed us to reduce Turbo bandwidth, then added new inclusive charm baryon lines
- Must support users in transitioning to any new features
- Turbo isn’t a great name...
Looking forward: Run 3

- Instantaneous luminosity increases 5x
- Triggerless readout, full software trigger
  - Removal of hardware trigger increases efficiency of hadronic signals > 2x
- Huge increase in signal rate!
Challenges

- Run 3 physics programme is **bandwidth-constrained** like charm was in Run 2

- **Turbo fraction must increase** 3 if the programme is to prosper; baseline is 70%
- Must migrate some **inclusive triggers** to the RTA model
- What if we cannot achieve **online/offline parity** in HLT2?
Real-time analysis is what we can do with information computed in the trigger. LHCb has profited enormously from it in Run 2. Must work hard to convince ourselves we're discarding things we will never need. Now have tools we need for Run 3.
End
References

- LHCb Collaboration, *The LHCb Detector at the LHC*, 2008 JINST 3 S08005
- R. Aaij et al., *Performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC*, arXiv:1812.10790
- R. Aaij et al., *A comprehensive real-time analysis model at the LHCb experiment*, arXiv:1903.01360
- LHCb Collaboration, *Computing Model of the Upgrade LHCb experiment*, LHCb-TDR-018
- C. Fitzpatrick, *A 30 MHz software trigger and reconstruction for the LHCb upgrade*, ACAT 2019
Online alignment and calibration

VELO alignment (~7min)
Tracker alignment (~12min)
OT global calibration
RICH calibration (every 15 min)

Calorimeter Calibration

MUON alignment (~3h)
RICH 1&2 mirror alignment (~2h)

((~7min), (~12min), (~3h), (~2h)) - time needed for both data accumulation and running the task
Online alignment and calibration stability

LHCb VELO
Preliminary

Variation [µm]

Alignment number [a.u.]

17/04/2018 - 21/11/2018

x-translation

y-translation