2019-05-22 - HSF Reconstruction and software trigger meeting

C. Doglioni, D. Lange, A. Dziurda - Intro

First of a series, summarizing the recent ATLAS and CMS cross-talks
Note: LHCb and ALICE's trigger systems will be summarized in another meeting (time constraints, these meetings should last ~1h or so)

Silvio Donato - Trigger system, DAQ and limitations

Data flow in ATLAS and CMS:

L1 readout: 40 MHz (no tracker) 100 kHz (full readout, with 1.5 MB/event)

- Hardware object reconstruction
- Evaluation of trigger bit (did hardware trigger pass)
- Application of trigger prescale (some triggers only record a fraction of passing events -> prescaled), but can also use low luminosity from LHC to increase the trigger rates and lower the prescales
- Application of detector veto during deadtime
- Triggering the full event readout
    - All events accepted by L1 are fully read

DAQ readout throughput bandwidth: 200 GB/s

Limits of the HLT farm (40k CPUs):
- Computing power
    - Every event can only use 3-400 ms
- Storage, merging, link to Tier-0
    - Max bandwidth 6-8 GB/s

HLT chain: sequence of reconstruction & filter modules, sortable to reduce the latency

Different implementation in CMS and ATLAS

CMS:
- Trigger chain is a process integrated in the offline framework
- Menu: long configuration file defining 700 HLT chains
    - Events saved into files, depending on the result of the HLT chain
    - Stored in Oracle database

ATLAS:

- Scheduler is separate from offline software framework, will be integrated in Run-3
- Each chain has steps (sequences of algorithms), event can be rejected at the end of each step
- Execution of algorithm either on a Region of Interest (RoI) or for the full event

ATLAS/CMS can use **multithreading** to parallelize (process two events in parallel, or two systems in parallel). CMS since 2015, ATLAS migrating.

Heterogeneous architectures for HLT can be used: GPU and co-processors (FPGAs)
- promising results for CMS tracking -> will install GPUs in HLT farms in Run-3
- ATLAS: studies for updating to GPUs for Run-4, as speed-ups for individual parts of algorithms are offset by data conversion and inter-process communication

After the HLT, data is written to the **Tier-0**
- Regular stream
- Saving only HLT objects (“data scouting” or “trigger-level analysis”)
- Partial event readout where only partial detector or HLT data is saved

Prompt reconstruction takes place right after recording, or much later (parked data, recorded now and reconstructed later)

CMS-only definition of **datasets and streams**:
- Dataset = list of HLT paths
- Data stream = set of datasets with the same event content

Limitation of the data streams come from technical limits (file limits)

Unconventional data taking: data scouting / TLA for ATLAS and CMS (Turbo for LHCb). Done for jets and B-physics (CMS)

Q&A

Conclusions between ATLAS and CMS are different on GPU. Reasons?
- Efficiency studies different
- Different hardware may have been used
- Costs of data transfer hard to amortize with framework used (migration will help)

Are we expecting changes on DAQ for Run-3?
- Not necessarily, we will lumi-level for Run-3

Heather Russell - Physics menu definition and monitoring

Trigger menu

What is a trigger menu? A list of types of events selected in each bunch crossing

Why do we need a menu? Because each trigger path has a cost -> build carefully, keeping everyone's physics priority into account.

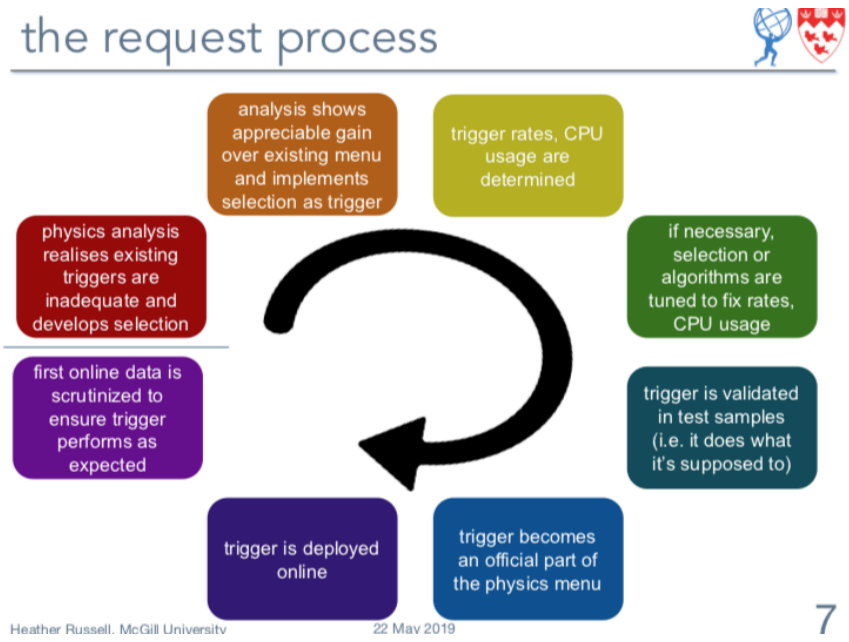
Definitions:

- Thresholds = cuts applied in the trigger.
- Prescales = factor p so that only every p-th event is recorded
- Disabled/enabled = if p=0 or p>=1
- Online/offline = events occurring during a run or after run has completed

Kinds of triggers in a menu:

- Generic (a single object)
- Analysis-specific (tailored to a specific event selection)
- Supporting (background estimation, performance measurements)

How does an analysis come up with a request, and how does this request get handled?



Determination of rate is an important ingredient: collect special unbiased dataset, rerun the L1/HLT and give an estimate before the trigger is run.

Trigger strategy driven by physics priorities of the experiment, and has to fit within the peak luminosity / be stable on average. Data that doesn't fit can be put in special streams (delayed reconstruction from parked data / smaller data size by reading out HLT object data or only

recording parts of the dataset). One can also increase the amount of data taken when the LHC gives less rate, e.g. end-of-fill.

Monitoring

Many different people/systems involved.

Online monitoring:
rate of triggers is the first alarm bell: monitor against expected/unexpected rates (expected rates: using fits to previous good runs)
Offline monitoring:
Select “good data” for physics analysis -> check turn-on curves, rate comparisons

Q&A

Q: How does ATLAS “survive” with 20 Hz in the express stream?
A: We have to, otherwise we’ll have trouble processing it
Follow-up: maybe one should ask what one needs 100 Hz for...

Q: What happens in case of a noisy channel? Automatic?
A: Generally manual for physics triggers (either via prescale or by masking the cell), for other kinds of triggers they are masked automatically (LHCb: automatic -> covered in other meeting). In general this is rare that it’s a major problem, one suffers the rate a bit more for a while. One tests carefully new menus and new triggers.

Q: would one want to prescale things dynamically, as a function of time?
A: ATLAS: we do not put physics into autoprescaling, only special triggers (e.g. empty bunches). Reason: want to mask “noisy” triggers without affecting the rest of the physics.
CMS: same, we would like to be able to change prescales without stopping a run but still do it manually.

Closing

Next meeting: June 5th, about algorithms & data structures to exploit many-core architectures

There will be another meeting after that, covering LHCb/ALICE as well to be announced on the mailing list. Cross-talk will be useful also when including both GPD and non-GPD, so we will invite ATLAS/CMS people to join the meeting and the discussion.