CWP WG - Software Triggering and Event Reconstruction

The software triggering and event reconstruction working group is charged with considering new approaches to the software trigger and event reconstruction. These areas have challenges including how event processing time depends sensitively on instantaneous luminosity, the need for vectorization, execution concurrency and frameworks that exploit many-core architectures. Software in support of triggering includes the process from the output of the hardware trigger through the decisions that lead to the real-time compression and archival of data (aka the high-level trigger at HL-LHC). Online algorithms may leverage commodity CPUs, CPU+coprocessor hybrids, GPUs and/or FPGAs. Import considerations include trigger steering, event building, data ``parking'' (for offline trigger decision), and data flow control systems, novel algorithmic technical such as machine learning algorithms adapted to real-time use, and real-time analysis systems.

For notes from the discussion session at the Jan 2017 UCSD workshop, see here.
Following this discussion, the triggering WG has been merged into the reconstruction WG
For notes from the discussion session on March 9th at CTD/WIT, see here

Participants:

ALICE/FAIR: Mohammad Al-Turany, Thorsten Kollegger
ATLAS: Andy Salzburger, Walter Lampl, Markus Elsing, Ed Moyse,Sami Kama, Frank Winklmeier
CMS: David Lange, Andrea Rizzi
LHCb: Johannes Albrecht, Michel De Cian,Vava Gligorov
FCC: Benedikt Hegner
ILC: Frank Gaede

Scope:

Reconstruction of raw detector or simulated data in both online and distributed offline computing facilities represents a major component of today's computing requirements. The software trigger and event reconstruction working group is to evaluate the most important components of next generation algorithms, data structures, and code development and management paradigms needed to cope with highly complex environments expected in high-energy physics detector operations in the next decade. It also deals with the transfer of traditional data analysis tasks such as the selection of specific exclusive signal, sidebands, and control samples into real-time environments. These advancements are critical to preserve both physics performance and to reduce computational requirements.

Cross-cutting issues which are not necessarily our focus, but will need to be addressed together with other WGs, include : developing experiment agnostic reconstruction frameworks, especially for fast prototyping; data structures and I/O considerations, especially when scheduling reconstruction algorithms; cross-platform development and validation of software, particularly in the context of heterogeneous processing farms.

Challenges over the next 5-10 years:

New approaches to software triggering and event reconstruction are required. Primary drivers for upgrades in algorithms and data structures include:

- Event complexity and event rate: the processing time depends sensitively on instantaneous luminosity. Data volumes are forecast to increase dramatically.
- The overabundance of signal : the LHC production cross-section is so large, and the detector hardware is working so well, that there is a unique opportunity to use these detectors to make precision measurements not only in the core high-PT and beauty-physics programmes but across a whole range of softer and more complex signatures, IF those can be reconstructed, selected, and the output data compressed in real-time. This is particularly true because the hardware technology allows the detectors to be read out, and the events built and sent into processing farms, at ever higher rates, far higher than can be affordably stored for offline analysis.
- Detector upgrades: Detector upgrades needed to cope with high luminosity facilities are typically more complex for reconstruction algorithms (e.g., fine segmentation).
- Technology evolution, hardware: The ability to efficiently use computing capabilities include vector units,many-core architectures and lower memory per core systems. At some point, 'efficiency' will also include operating costs (HVAC, electric power, even human operators).
- Data structures and I/O performance: Data structures are an important component for communication between algorithms and the throughput of applications reading the output of the software trigger or event reconstruction.
- Online calibration and alignment: as more reconstruction and selection is moved into the trigger, there is a growing need to reliably calibrate and align our detectors in real-time.
- Code maintenance challenges: Issues around testing, validation and interoperability and interdependencies especially in light of technology evolution.

Charge/Questions for the WG:

The software triggering and event reconstruction working group should evaluate the relative importance new approaches to the software trigger and event reconstruction foreseen on the 5-10 year timescale. Important questions related to the above challenges include:

1) What are the most critical algorithms to target (both online and offline)? How do we continuously integrate (pun intended) the ongoing developments in ML software and techniques within our reconstruction and triggering?
2) What is the impact of the increase in trigger complexity as more and more algorithms currently part of the offline event reconstruction and analysis applications are moved into the software trigger?
3) How to make efficient use of x86 and evolving computer architectures? Can current software implementation be evolved? Are more revolutionary solutions needed?
4) How can we make time-critical algorithms as agnostic to computing hardware as possible? Are some algorithms, e.g. some kinds of ML algorithms, particularly sensitive to the hardware?
5) How does the evolution of triggering approaches and output rates impact event reconstruction?
6) How can the data volume per event be reduced without impacting the physics quality of the reconstructed data?
7) Do we need more sophisticated frameworks for monitoring and DQ of real-time alignment and calibration? How does anomaly detection fit into this (link to ML WG). What if a calibration algorithm needs large statistics? What if a second processing is needed for specific analyses? How do we propagate the new calibration constants in real-time, and archive them for later use (link to conditions WG)?
8) How can data structures evolve to allow more effective algorithm interoperability and data readback?
9) Can future detector designs be tweaked to optimize reconstruction timing and performance?

The desired outcome is a list of identified areas of concentration, including specific areas of work that would allow current algorithms to be transformed or replaced in favor of algorithms capable of addressing future facility needs. Demonstrator projects should be considered together with their potential impact on the physics or technical performance of computing for future facilities. Projects should be identified with estimated timescales, estimates of expert development personal needs, and computing facilities needed for completion.

Areas to be addressed by this working group include:

1) Identify on-going R&D efforts in the area of software trigger and event reconstruction in the HEP community. Understand the goals, timelines and (if any) relevant R&D that is needed to consolidate each these efforts to a corresponding sustainable software toolkit.

2) Identify and prioritize algorithm designs and demonstrators needed to produce analysis quality analysis data formats in near real time. This should include a systematic analysis of how online and offline algorithms are expected to evolve in different areas of HEP. Example considerations in this area include how to create reliable detector-level and physics-object level calibration procedures and how to reduce the distinction between the reconstruction software deployed in online facilities and distributed offline facilities.

3) Identify and prioritize challenges to operating robust software trigger applications as their algorithm set and data structures increases in complexity. Example considerations include memory management, interaction between algorithms or trigger components and application validation.

3) Identify and prioritize algorithm designs and demonstrators needed to improve the most costly algorithms, such as charged particle tracking and high-granularity calorimetric clustering in complex (e.g., high pileup) conditions with high accuracy in complex environments while achieving low computational requirements. Examples could include how to utilize heterogeneous grid, HPC or cloud computing facilities and specialized computing architectures. An important consideration are the pros and cons of using these include either generic algorithm implementations that do not bring a significant performance penalty, or re-engineering of experiment specific implementations.

4) Identify and prioritize needed research and demonstrators needed to show the feasibility of deep learning or other advanced techniques to replace specific reconstruction algorithms with computationally efficient alternatives based on new techniques or new applications of techniques including those developed outside of the HEP community. Considerations include how algorithms using these techniques are best incorporated into trigger or event reconstruction applications.

5) Identify and prioritize needed research and demonstrators to reduce the I/O needed of event reconstruction event stores. Evolution in experimental event rates and event complexity may outstrip disk technology evolution by a large factor. Examples could include techniques to allow for event reduction, selection of subevents (eg, particles from only the most interesting vertices) and libraries for flexible precision/range definition.

Comments:

Brett Viren: For DUNE, we can not take much advantage of triggering to reduce what data goes into full reconstruction. At least at the level of one detector unit (APA) the full 2560 channels at 2MHz over 5 ms block of data hits the reconstruction algorithms. This means DUNE is on both branches I/O and CPU (which I attribute to an illustration by Frank Wuerthwein which I can try to dig up) that OSG defines the problem in terms of.

Johannes Albrecht: We should think about the overlap of event reconstruction and triggering, as a larger and larger part of triggering is moved to software triggers with object reconstruction close to offline quality. We could think of merging the two groups.
Another point is the use of accelerators in the event reconstruction. A systematic analysis of their use across the different experiments could be very useful.

Tommaso Boccali: I think we could explore also other dimension; just a brain dump here.

- Full reco / fast reco in the same workflows (once identified an event as interesting / trigger level??) decide whether to reconstruct it fully or partially. For example, just detector regions, or even just save a few identified quantities (for MinBias, one could be interested in just # of jets, # of tracks, etc). OR: for “not so interesting events”, change precision in tracking.
- Conditional reconstruction, based on presence/absence of hardware feature (GPU onboard, FPGA….) - and how to have all the approaches stay in sync.
- Compute vs save: if I/O becomes the problem for the reconstruction output, it could be needed not to save persistently large quantities, but produce on the fly at analysis time. Define methods + handles to turn.

Michel De Cian: Some specific questions that could be interesting:

- What are the needs (and the feasibility) of a fully software based trigger and the consequences for the hardware (CPU, GPU, x86, KNL, …) when the luminosity is

increased by O(10x) compared to now at the LHC. Can we just expand on current ideas or are completely new concepts necessary?
- What MVA techniques exist or need to be developed that can be applied at very early stages of data processing / event reconstruction, that are fast enough to run online.
- How can timing information (in addition to spatial information) from future detectors be included and what are the consequences for computing / reconstruction.

Andrew Norman: This should be expanded to be more generic and apply to more advanced streaming triggers. How should the cross over between L3 triggers and evt reco be treated? (meaning as they merge more and more should there be an intermediate world of pseduo software triggering that is broken out from the hardware).
Also machine learning. This applies at both trigger and at reco, post reco levels.

David Lange: "Parking" has taken a different meaning (eg, essentially just increasing the raw data sent to tape) - I suggest "event building, and data flow control systems. Software for approaches to increasing the latency window for making the final trigger decision and using derived event information to preserve trigger decisions, possibly for a large fraction of all events."

Brett Viren: DUNE Far Detectors have a challenging software triggering situation which will potentially require EB/year level of throughput. This has to mix-in self-triggered supernova burst triggering and externally soft-triggering from packets sent from the beam monitoring at FNAL.

Simon George: for ATLAS, HLT algorithms and frameworks have a large overlap with offline reconstruction, which is covered elsewhere, while online-specific code for FPGA/GPU tends to be in the trigger hardware domain rather than software. So I see this WG as an application domain rather than a problem domain.
I think there is more potential for common DAQ software but that is perhaps outside the scope of the HSF CWP?

Vava : We need to decide whether we really want to try to cover both “hardware” triggers, i.e. those which have to operate within a fixed latency, and “software” triggers (commonly HLTs) which can run asynchronously depending on how much disk buffering is available (and where) within the same WG. For each individual experiment these will be tightly linked, but the technical solutions might be quite different in the two cases. If we do want to cover both, the topic of emulating hardware triggers will probably be relevant and connect to the simulation/reconstruction WGs.
LHCb’s HLT is moving towards full event building at the collision rate and performing analysis at the trigger level. At that point the trigger takes on the role of the full reconstruction, but also has to deal with a lot of problems traditionally taken care of by data management/access. E.g. the trigger is no longer some (say dozens) of inclusive trigger lines, but thousands of exclusive selections for specific analyses, with all the maintenance and validation problems which that

entails. How do we set up a framework for helping the offline analysts design their trigger line, design the relevant lines for control samples, calibration samples, backgrounds, etc.
Real-time analysis implies real-time detector alignment and calibration. This also means we need more sophisticated monitoring systems, e.g. automated anomaly detection? Links to the machine learning WG.
Once you event build at the full rate, the choice of technology for the HLT comes down to a cost/benefit analysis within the constraints of each experiment (e.g. cooling capacity). Do we have the tools to perform such comparisons in a systematic way?
We should discuss the topic of machine learning algorithms which are safe for real-time use together with the ML WG. E.g. LHCb already uses BDTs to cover most of its HLT bandwidth, we already have NNs in the tracking (which is same online and offline), and have put some thought into how to make these safe for running with changing detector conditions etc.

Johannes:
Concerning higher level triggering, we should reflect if it makes sense to separate the event reconstruction from the triggering parts of the CWP.
The LHC experiments are moving more and more to a scheme of trigger aware analyses, where the event reconstruction is done to an almost final quality in the High Level Trigger. Here we should discuss the strategies to align the detectors and calibrate the subsystems in real time, as currently implemented in LHCb. This topic concerns more than only the LHC experiments but rather all data intense HEP experiments.
Another important point is the use of hardware accelerators in the event reconstruction. ALICE seems to be able to profit from GPU’s while this seems difficult to prove at ATLAS, CMS and LHCb. Here it would be good to investigate characteristics of algorithms that make them efficiently accelerated on GPU or other architectures.
Concerning the last stage of software triggering, the trigger decisions are based on more and more complex algorithms and the combinatorics of the input objects will be a major consumer of CPU time. We should investigate how an combinatorics engine that makes efficient use of the resources can be developed.