Concepts and Technologies Used in Contemporary DAQ systems

Martin L. Purschke
Brookhaven National Laboratory, Upton, NY, USA

Abstract

The concepts and technologies applied in data acquisition systems have changed dramatically over the past 15 years. Generic DAQ components and standards such as CAMAC and VME have largely been replaced by dedicated FPGA (field-programmable gate array) and ASIC (application-specific integrated circuit) boards, and dedicated real-time operating systems like OS9 or VxWorks have given way to Linux- based trigger processor and event building farms. We have also seen a shift from proprietary bus systems used in event building to GigaBit networks and commodity components, such as PCs. With the advances in processing power, network throughput, and storage technologies, today's data rates in large experiments routinely reach hundreds of Megabytes/s.

We will present examples of contemporary DAQ systems from different experiments, try to identify or categorize new approaches, and will compare the performance and throughput of existing DAQ systems with the projected needs of the LHC experiments to see how close we have come.

INTRODUCTION

Over the past 30 years the field of physics instrumentation has undergone a dramatic shift. The 70's, 80's, and early 90's were dominated by standard readout equipment, most notably a wide variety of commercial CAMAC and VME boards, with a smaller number of FastBus-based electronics. While there were many vendors of electronics, this era of physics instrumentation was dominated by the *LeCroy* company. Even today, many smaller test setups for new detectors see their first test using LeCroy readout modules, and experimentalists still refer to smaller discrete electronics setups (e.g., for a simple trigger in a test beam) as "Blue Logic", referring to the standard blue front-panel color of LeCroy modules.

In the early 90's, the planning phase for the experiments taking data now (and with the LHC originally planned to turn on in early 2000), it was realized that he classic design of a data acquisition system was no longer viable. The classic design had a dedicated trigger, and used delay cables for each readout channel to hold the analog signals long enough to allow the trigger logic to take a decision (figure 1).

In the early 90's, with experiments and the channel count getting larger and the coverage getting close to 4π, it became clear that the delay cable method would no longer work – if only for cost reasons – a long enough delay cable to get 1000ns delay (200m) would make the cost per channel prohibitively expensive. There is just not enough space in the core of a modern experiment to accommodate that many cables. It was realized that other technologies had to be found, and the focus shifted to arrays of storage cells, so-called *pipelines*, that are able to hold the signal in a given cell long enough to wait for the trigger decision (figure 2).

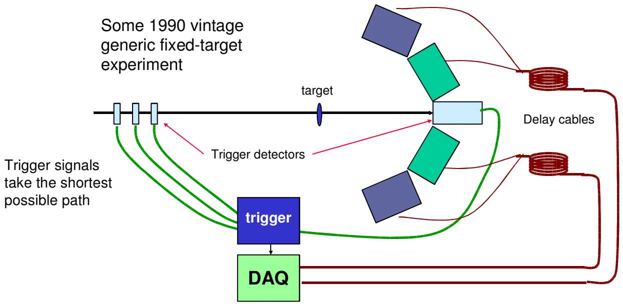
Some 1990 vintage generic fixed-target experiment
target
Delay cables
Trigger detectors
Trigger signals take the shortest possible path
trigger
DAQ

Figure 1: A schematic view of a vintage-1990 fixed-target experiment. The signals needed to form the trigger decision are sent to the trigger electronics on the shortest path, while the actual signals are sent through delay cables, delaying the arrival long enough for the trigger decision to be ready.

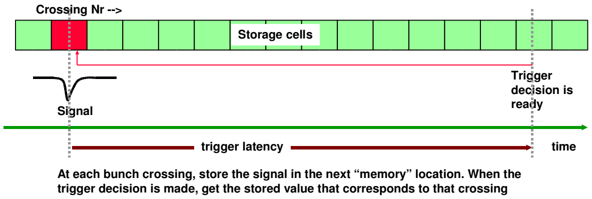
Crossing Nr -->
Storage cells
Trigger decision is ready
Signal
trigger latency
time
At each bunch crossing, store the signal in the next "memory" location. When the trigger decision is made, get the stored value that corresponds to that crossing

Figure 2: Storage pipelines replace the classic delay cables. The signal is stored in one of the cells long enough until the trigger decision is ready.

There are two sorts of pipelines: Analog pipelines, also called *Analog Memory Units* (AMU), and digital ones, also called *Flash ADC's*. The choice for a given experiment and detector is driven by the required resolution, power consumption, storage depth, and cost. Virtually every experiment uses one or the other form of a pipeline, or both.

HIGH-LEVEL TRIGGERS

The time between bunch crossings rates in a modern accelerator is very short, reaching 106ns bunch spacing at RHIC and 25ns at the LHC. The strategy is to look quickly at each crossing, and forming a Level-1 trigger decision. The short time, and the fact that those Level-1 triggers have only a keyhole view of the detector means that it is only possible to recognize localized interesting patterns, such as a high energy deposit in a calorimeter, or certain patterns in the tracker, which make the event interesting enough to be passed on to the next-level trigger.

The standard design calls for 3 levels of triggers, with each level receiving fewer events and thus having a longer time to look at each remaining event. In the case of the ATLAS experiment [1] at the LHC, the Level-1 trigger typically reduces the 40MHz collision rate to about 100kHz, Level-2 to 1kHz, and the final Level 3 trigger to the eventual archiving rate of about 100Hz (figure 3).

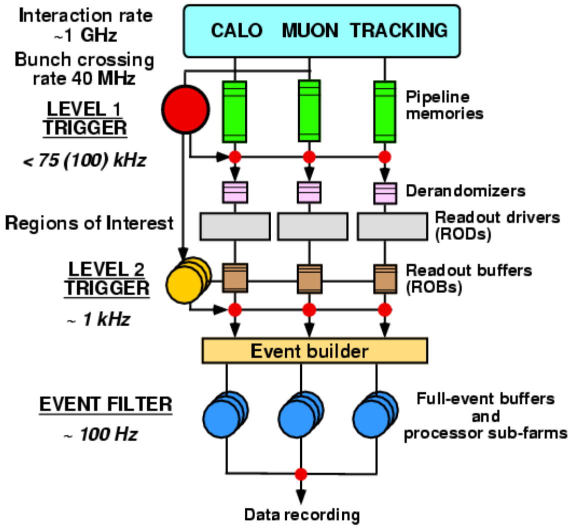

Figure 3: A schematic overview of the ATLAS trigger. The Level-1 trigger reduces the event rate to about 100KHz and forms *Regions of Interest* for the next trigger level to look at. The final Level 3 trigger filters can have an in-depth look at the events and select the 10% most interesting ones for archiving.

There are differences in the design of the high-level triggers, however. The ALICE experiment [2], with higher data volumes due to higher detector occupancies in the Heavy-Ion runs of the LHC, has an additional trigger level to further reduce the event rate (figure 4).

The BELLE experiment [3] at the KEK has an additional "2.5"-level trigger stage to cope with the data rates.

The CMS experiment [4, 5] has adopted a different design. While the Level-2 trigger in ATLAS reduces the event rate to about 1KHz before the final level-3 trigger, the CMS experiment has only *two* trigger levels. The final trigger level therefore receives the full level-1 rate of about 100KHz, which means that all networks and switches have to be able to cope with a 100 times higher throughput than in the ATLAS experiment (figure 5).

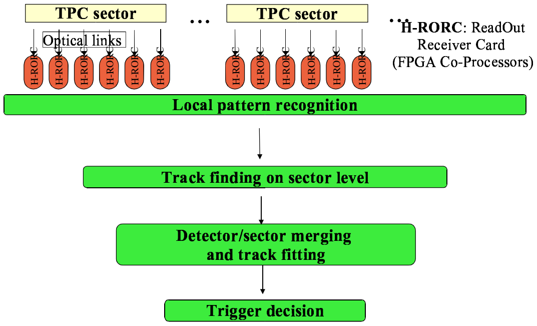

Figure 4: The trigger schematic of the ALICE experiment. While the ATLAS experiment has 3 trigger levels, the higher detector occupancies in the Heavy-Ion runs of the LHC call for an additional trigger level for the ALICE experiment.

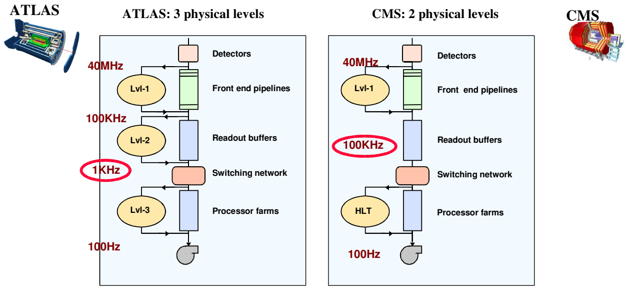

Figure 5: The overview of the trigger design in the ATLAS and CMS experiments. CMS has only two trigger levels, and the finals trigger stage has to be able to cope with the full Level-1 data.

The advantage of the CMS approach is that the highest trigger stage has access to all detector data and can perform a more thorough analysis of the data to determine the interesting features of the event.

TECHNOLOGIES IN HIGH-LEVEL TRIGGERS

This area of high-level triggers has seen some very interesting developments over the past few years. Historically, the whole trigger was implemented with specially developed or generic electronics or hard-wired boards that would form the trigger decision. This has largely been replaced by *firmware* in the first trigger stage, where one typically finds massively pipelined logic implemented with FPGAs and ASICs, which have largely replaced the discrete logic designs of the old days. The firmware approach, while being much more cost-effective than dedicated logic boards, also allows one to change the design by replacing the firmware with a new version.

Virtually every experiment implements the higher-level triggers in software, occasionally on VME processors such as the D0 experiment [6], or on standard Linux PC farms, as all LHC experiments do. This has led to a degree of merging of the code bases of the experiments, since the trigger software needs the same cluster finding and tracking algorithms that are readily available in the offline software of the experiment. This puts new requirements on the design of the offline software, because the standards for reliability and robustness of the software are much higher in real-time trigger applications. A software bug will cause events being thrown away forever, and the continuous operation of the trigger leaves no room for memory leaks or similar bugs.

One experiment, BTeV at the Tevatron [7], stands out because of the design of the level-1 trigger in a generic PC farm. The trigger to select the displaced vertexes for the B-decays is implemented on standard hardware. It is an interesting concept that is worth looking at.

DATA ARCHIVING RATES

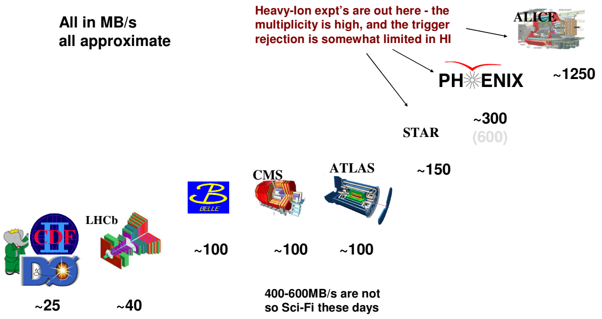

Figure 6: An overview of the archiving rates in different experiments. The rates range from below 20MB/s for the experiments at the Tevatron to the highest rate of 1250MB/s for the ALICE experiment. Of the currently active experiments, PHENIX has achieved the highest sustained rates of about 330MB/s, and an upgrade to a capacity of 600MB/s is planned.

Advances in computer technologies have greatly increased the capacities of disks and other storage media, but not at the same pace as the growth in processor and network speeds. To this day, large-scale data storage remains expensive. The highest data archiving rates are seen in the ALICE experiment, which anticipates rates of 1250MB/s. In general, the heavy-ion experiments have the highest archiving rates, which reflects the fact that the detector occupancies are higher than in p-p collisions, and trigger algorithms are less selective due to the high particle multiplicities. The PHENIX experiment [8] at RHIC has recently increased the archiving capacity to about 400MB/s and taken data at sustained rates of about 330MB/s in Run 4. For the upcoming Run 5 of RHIC, the capacity has been further expanded to 600MB/s, which is about half of what ALICE will archive.

DATA COMPRESSION

After all savings that can be accomplished with the data *reduction* techniques (zero-suppression, thresholds, triggers, etc) have been exhausted, one often finds that the size of the data files on disk can still be reduced by *compression* techniques, e.g. with utilities such as gzip. Compressing files is not practical, however; the compression needs to be applied in the DAQ before the files are written in order to save space. On the other hand, the compression can usually not be applied before any of the high-level triggers – the trigger algorithms would need to uncompress the data first, reversing the savings by the compression.

We have in the past experimented with designs that would compress certain event types only. Candidates for front-end compression are so-called calibration events, where laser or LED light is injected in the calorimeters, or charge is injected in ADC front-ends, in order to measure and monitor the gain. Those events do not need to pass the physics trigger, but have, by definition, an occupancy of 100%, and are thus candidates for front-end compression. However, front-end based generic compression algorithms are too challenging at this time to be practical.

Late-stage data compression has been standard for some time in the offline world. The ROOT-I/O mechanism has a built-in gzip compression algorithm. The challenge in DAQ is to perform the compression fast enough to keep up with data rates of several 100's of MB/s. The Event Builder of the PHENIX experiment switched to a different compression algorithm of the LZO family [9], which is about a factor 4 faster than the algorithm used in gzip, but yields comparable compression factors. The Assembly and Trigger Processors (ATP's) compress the buffers they send to the data logger. In this way, the CPU load for the compression is distributed over many CPU's, and the compression can keep up with the data rates of the experiment. Since the event rate of the PHENIX experiment is limited by the archiving rate, this gave a factor of two in the number of archived events.

COMMON DAQ FRAMEWORKS

In the past, each experiment was so special and unique that the data acquisition was usually custom-designed and built. Even in the *offline* world, truly generic frameworks are rare when one looks above the lowest-level denominator of ROOT as the underlying framework. In a way, this resembles the situation in the business world in the era of expensive mainframe computers, when each company would design and write the accounting and other business-related software in-house. This approach is unimaginable today. It is conceivable that over the next few years standard DAQ frameworks could emerge that are applicable in a variety of collider experiments. There are some candidates today. The CMS collaboration has designed “XDAQ” [10], an generic “DAQ builder”, which is meant to take care of all DAQ needs of the CMS collaboration. XDAQ is not designed to be “CMS-free” and it remains to be seen if it is applicable to other, non-CMS related experiments.

The best example of a generic data acquisition framework is MIDAS [11], which is a medium-size DAQ system in widespread use in a variety of experiments. Experiment-neutral by design, it might evolve into a standard setup for smaller experiments.

Finally, several experiments at CERN, such as NA60 and COMPASS [12], are using the ALICE DATE [13] data acquisition framework, getting an actively maintained DAQ and at the same time serving as a test bed for ALICE.

SUMMARY

Over the past several years, the field of data acquisition design for large experiments has seen many changes. Among the most important ones are

- FPGA and ASIC designs have largely replaced traditional hardware logic. Most functionality is now implemented in firmware.
- Gigabit networks have replaced the majority of buses (VME, etc)
- Commodity PC farms running the Linux operating system have replaced dedicated VME processors and real-time operating systems.
- With components of the offline software running as a part of the high-level triggers, there is some merging of these previously distinct code bases.
- 500MB/s archiving rates and 500GB/s Event building rate are a reality.
- Data compression techniques can usefully augment standard data reduction techniques.
- There are examples of emerging standard DAQ frameworks (Midas, XDAQ, DATE).
- Early-stage test setups still use CAMAC and VME, and expertise in these technologies is still needed.

REFERENCES

[1] ATLAS Collaboration, *ATLAS Detector and Physics Performance*, Technical Design Report, CERN/LHCC/99-14

[2] ALICE Collaboration, *ALICE Technical Design Report*, CERN/LHCC 2001-021

[3] BELLE Collaboration, *The Belle detector*, Nucl. Instrum. Meth. A479, 2002, 117-232

[4] CMS Collaboration, *The Trigger and Data Acquisition project (TriDAS)*, Technical Design Report, CERN/LHCC 2000-038, 15 December 2000

[5] CMS Collaboration, *The TriDAS project, technical desiggn report: Data Acquisition and high-level Trigger*, CERN/LHCC 02-26, 2002

[6] D0 Collaboration, http://wwww-d0.fnal.gov

[7] BTeVCollaboration, *Description of the BTeV detector*, Int. J. Mod. Phys. A16S1C, 1062-1064, 2001

[8] K. Adcox et al, *PHENIX detector overview*, Nucl. Instr. Meth A 499, 2003, pp 469-479.

[9] M. Oberhumer, *The Lempel-Ziv-Oberhumer data compression library*, http://www.oberhumer.com/opensource/lzo

[10] V. Brigljevic, *Using XDAQ in Application Scenarios of the CMS Experiment*, CHEP2003, La Jolla, California, 2003, http://www.slac.stanford.edu/econf/C0303241/proc/papers/MOGT008.PDF

[11] S. Ritt, http://midas.psi.ch

[12] COMPASS Collaboration, *COMPASS: A Proposal for a Common Muon and Proton Apparatus for Structure and Sprectroscopy*, CERN-SPSLC-96-14

[13] CERN ALICE DAQ Group, *ALICE DATE User Guide*, ALICE Internal Note, ALICE-INT-31-2000