H1 CALORIMETER DAQ UPGRADE FOR HERA-II

DIRK HOFFMANN, PIERRE-YVES DUVAL, CLAUDE VALLEE
Centre de Physique des Particules
CNRS — IN2P3 — Univ. Méditerranée, Marseille, France
E-mail: Hoffmann@CPPM.In2p3.Fr

The H1 collaboration has performed an upgrade of its data acquisition system for the calorimeters in view of the HERA-II programme. A heterogeneous system based on 29K/VRTX, 68k/OS9 and Vax/VMS was replaced by an integrated Unix cluster composed of two PPC/LynxOS VME boards and Sparc/SunOS stations, using TCP/IP protocols for inter process communication (IPC) and POSIX standards in general. Software transcription consisted of porting three essential functions: hardware setup, calibration datataking with a high serial data throughput and online datataking which emphasizes low frontend deadtime through a three level buffering by means of POSIX threads and messages. Low performance control tasks were programmed in Perl, the user interface has been written in Java. Although the very frontend electronics remain unchanged, a factor two increase in performance was obtained together with a manifestly improved environment for monitoring and diagnostics.

1. System Context and Environment

1.1. *H1 Calorimetry*

The H1 experiment[1] is a multi-purpose 4π detector, which has measured electron-proton and positron-proton collisions on the HERA-I accelerator since 1992. Since 2000, the accelerator underwent a major upgrade[2] with the aim to increase instantaneous luminosity by a factor five and to deliver longitudinal lepton polarization. The detector was modified according to the new conditions around the interaction region, and the collaboration also used the time for further enhancements of the experiment. After the upgrade the Calorimeter Subdetector group henceforth comprises

- a Liquid Argon (LAr) Calorimeter, which contains 44,352 cells and covers the major part of the polar angle,
- a Spaghetti Calorimeter (SpaCal) with 1,300 photomultiplier channels in the backward direction for scattered electron detection
- a small PLUG Calorimeter in the very forward direction for particle measurements at very low angle, which contains eight tiles for a rough

calorimetric measurement, and
- the analog readout part of the instrumented iron, called Tail Catcher for the measurement of hadronic shower tails leaking out of the LAr calorimeter, with a total of 3,888 channels.

They are readout by a common data acquisition (DAQ) system, the Calorimeter DAQ[3] (CaloDAQ), which delivers in turn the data to the Central H1 DAQ.

1.2. *Energy and Time Measurement*

The analog data of in total 49,548 channels are shaped in different electronic equipment, according to their origin and time behaviour[a]. The final digitization for the energy measurement is performed by identical 12-bit ADCs, which are multiplexed, sequenced and read out by a total of 70 custom made DSP boards[4] equipped with CPUs of the 56k generation[5]. Three additional DSPs of the same type treat special detector signals from photodiodes, trigger electronics and TDCs in the SpaCal partition. Certain channels with large signal dynamics are fed to the ADC battery through two amplifiers with different gains, thus allowing for an effective 14-bit precision. A third order polynomial calibration correction and zero-suppression are performed by the DSPs on the fly, without adding overhead to the generic sequencing time of the ADCs.

In parallel to the precise and stable energy measurement, the LAr cells are used in the LAr Trigger[6] branch to deliver fast signals for time measurement and trigger purposes. To this end, all channels are fed into a total of 512 big tower sums according to their geometrical location and separately for the cells of the electromagnetic and hadronic calorimeter parts. Their signals are sampled after fast shaping by 8-bit Flash-ADCs at the HERA bunch crossing rate of 10.4 MHz. They fill a digital pipeline delivering a trigger decision after 2.3 μs, but internally keep a history of 256 bunch crossings. The readout is performed by eleven 56k DSPs, mounted on a different type of custom made VME board[7]. These are in charge of the calorimeter data transmission to the higher trigger levels and perform the readout of the big tower signal history around the nominal interaction time and the calculation of time weighted energy sums for signal pileup monitoring. The trigger raw data amount to 150 kB, which is comparable to the data volume in the calorimetry ADC branch.

---

[a] The pulse length in the SpaCal is of the order of 1 ns, whereas the drift time in the LAr Calorimeter nears the 0.5 μs.

1.3. *Inter Crate Communication and VME Tree*

The CaloDAQ readout electronics consist of several hundred boards in VME standard, accommodated in 47 VME crates, which communicate through

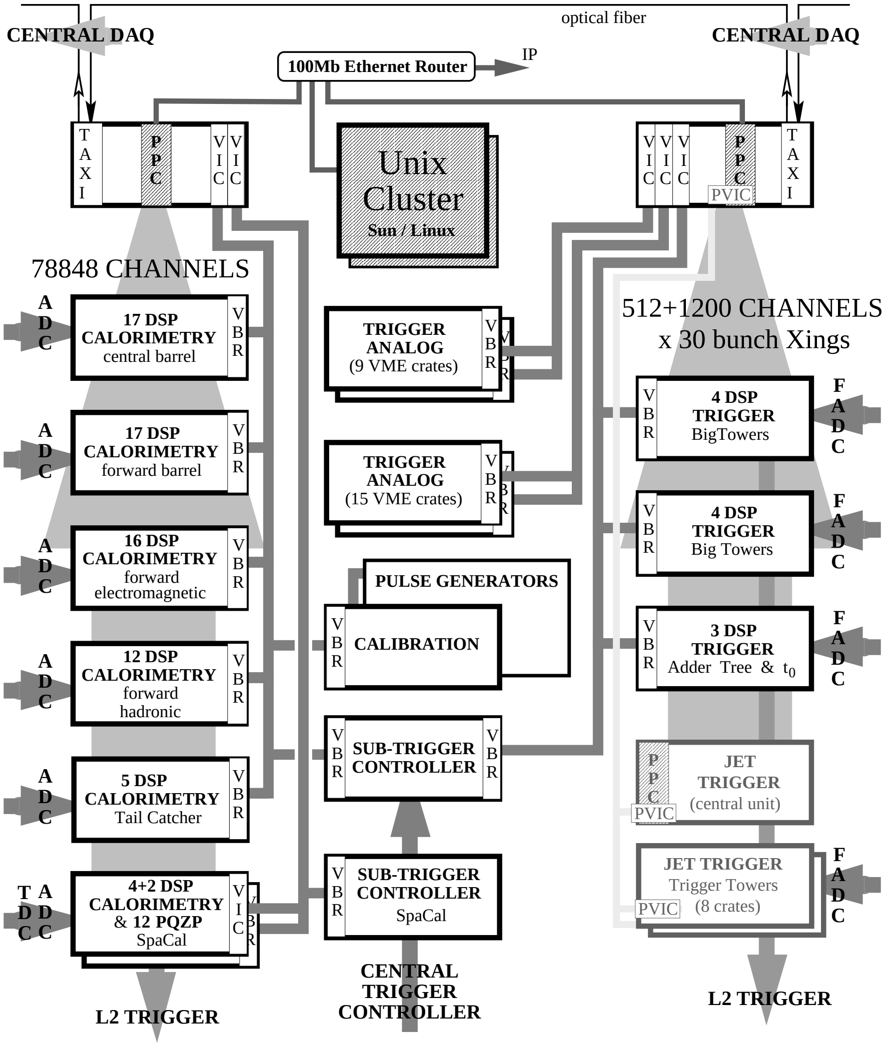

Figure 1. VME tree of the readout electronics in the two CaloDAQ branches: Crates on the left side contain the DSPs of the ADC/TDC branch. DSP crates for the trigger readout are placed on the right. Two big arrows indicate the main directions of the readout data flow. The Event Builder PPCs on top of these arrows feed their data into the Central DAQ Taxi boards, which are linked by an optical fiber ring to the other H1 DAQ branches. All VME crates of the frontend electronics are connected by VICbus branches, starting from either of the Event Builder crates. The PPCs are connected to each other and a local Unix cluster fileserver through Fast Ethernet.

VICbus[9] connections on two and three vertical busses respectively in two separate branches. Fig. 1 illustrates this segmentation into the ADC/TDC branch and the Trigger branch, each equipped with a Taxi[8] connection to the Central DAQ. The readout hardware properly speaking is completed by crates for setup, calibration, interface to the Central H1 Trigger and monitoring.

The address space in this VMEtree is managed by a custom-made VMEtree compiler, which allocates VME pages dynamically and translates the tree into the setup of the master VIC boards. Extensions of the Calorimeters like the Jet Trigger[10] are supposed to use the faster, smaller and more reliable PVIC[11] connections for the same price. The PVIC setup will be integrated in the VIC-VMEtree compiler as well, thus providing the option to replace single VICbus branches transparently in future.

The Event Builder PPCs are connected to each other and to a local Unix cluster by 100baseT connections (Fast Ethernet) on two Gigabit IP switches, which communicate through a glass fiber backbone. This is the gateway to all other Internet hosts and the switches can isolate the CaloDAQ hosts completely or limit the bandwidth for outsiders, if needed.

1.4. *Hardware Choices for the Upgrade*

The newly installed upgrade parts are hatched in fig. 1; foreseen extensions are shaded in light gray. Two PPCs on a RIO2[12] VME board replace the former[3] AMD 29k processors; and a Unix cluster replaces the anterior OS/9-Vax system in the central control. Control and management tasks (cf. sec. 2) have been partly shifted into the frontend, due to significantly higher computing power of contemporary CPUs. All PPCs run the real-time operating system LynxOS[13].

The extension by the Jet Trigger[10] project will demand additional capacities in the readout of the Trigger branch, which can be provided through a satellite PPC station on a PVIC[11] bus.

The Unix Cluster was extended from an existing Sparc station running SolarisOS. It is used as a fileserver for all LynxOS systems and included in the central DESY backup for maximum data integrity. Tests with a Pentium III under a recent version of Linux showed that there are no drawbacks when using this lower-priced alternative, except for the different byte order in the CPUs, which might imply some careful rewriting of offline analysis programs.

2. Software Design

2.1. *Servers, Clients and Protocols*

The Event Builder CPUs can be accessed through Internet protocol (IP). Firewalls and server internal host selection algorithms protect against unwanted accesses. Both Event Builders have an identical client/server program structure (see fig. 2), which enables them to call each other in case of requests which need synchronization. The “entrance” for all DAQ requests is a Unix socket, which is bound by the daemonizing[b] server `daqd`. Depending on the kind of request, `daqd` selects VME service programs, which do the actual hardware access. In this way the exclusive access to VME can be granted if necessary.

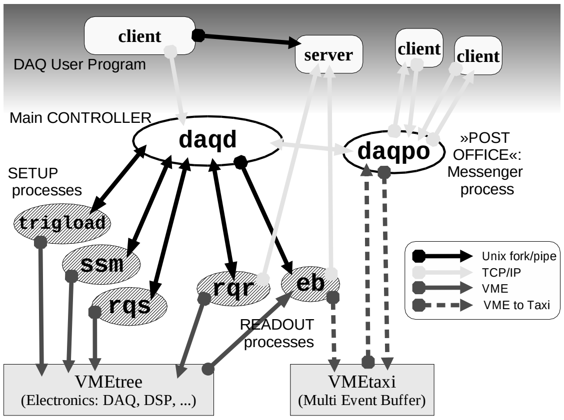

Figure 2. Software architecture of the new CaloDAQ for one Event Builder: On request from a DAQ client, the DAQ server **`daqd`** can launch one or several service processes, which access the VME bus. Service tasks which send back data, can write into a file, connect in turn to a data receiving server, or dump data into the Multi Event Buffers of the Central DAQ Taxi. The post office process **`daqpo`** is explained in the text.

The `daqd` request protocol is not time critical and therefore plain text, which has proven to provide efficient ways for debugging or test of clients and servers independently from the current status of other parts of the project.

Service programs include request for setup (`rqs`) of the VME boards, request for readout in standalone mode (`rqr`), expert setup of the LAr Trigger (`trigload`, `ssm`) and high performance event building for real time data taking (`eb`, sec. 3). They are implemented as standalone programs, taking their configuration data from the standard input channel, which allows once more uncoupled (object oriented) testing of single blocks. In case they produce any data for the requester, these are formatted in H1 standard BOS[15] format and sent in binary form through an extra IP channel.

As the response of `daqd` is strictly sequential and prioritized, the task of asynchronous status and error reports between processes has been delegated to the DAQ post office (`daqpo`) daemon. It allows message exchange in the DAQ through an unlimited number of sockets from various clients with low priority compared to other tasks on the CPU. It broadcasts incoming messages to all connected clients. In addition, `daqpo` translates Central DAQ run requests

---

[b] detached from its calling (parent) process and running in background, typically started at system boot time

coming through the traditional VMEtaxi channel, as long as the IP protocol is not yet implemented there.

2.2. *Choice of Programming Languages*

The different elements of the DAQ software have been encapsulated to the most possible extent during design and are only seen by others through their interface definition. This enabled us to choose different technologies according to the aim and the requirements of each of the programs.

`daqd`: modest requirements on reactivity and data throughput on one hand and high flexibility for the implementation of new requests on the other led to the decision for a compiled scripting language, Perl 5, which is optimal for the recognition of text patterns and reliable for long uptimes.

VME service programs: The high demand for performance and the usage of system level C routines made it necessary to use compiled programs for these tasks. The chosen standard for message queues and threads was POSIX[18].

`daqpo`: was designed in the style of all other VME service programs.

DAQ Clients: Up to now, and not including the trivial method by a `telnet` command, three different technologies have been used to design clients for the new DAQ. The main user interface (cf. sec. 4) has been written in Java; a very complex trigger analysis and calibration package had been inherited in FORTRAN and was adapted to the TCP/IP protocol by means of a C wrapper; and some debug clients and receiver servers have been written in Perl.

The combination of all programs for the standard electronic offline calibration resulted in a gain in performance of an approximate factor two. This will allow to control and perform the calorimeter calibration more often and more reliably, as the regular idle time of the accelerator without beam is usually not more than 30 minutes.

3. Real Time Behaviour

The most challenging project in DAQ systems is the readout part, where optimal performance is needed in interplay with many other systems of the detector. The event building programs of the two CaloDAQ branches are split into three main threads, which are shown in fig. 3 for the ADC/TDC branch[c].

---

[c] The Trigger branch is designed identically with the exception that DSP and IT routine have three front end buffers.

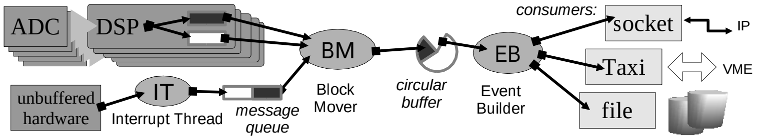

Figure 3. Program threads (in ovals) of the real time event building program, from left to right with decreasing priority: The IT provides the buffering for unbuffered front end registers. The bulk of data is stored in one of two DSP buffers. The BM gathers the primary buffer blocks and writes them to a circular buffer, from where the EB dispatches them among possible consumers after formatting.

As not all frontend hardware contains internal buffers like the DSPs described above, an interrupt routine (IT) is used to readout these registers and hands them over to the block mover (BM) in two buffers. The BM task acts, when the first order dead time has finished, because at least one front end buffer for the next event is available in the DSPs. It stores the data from the front end buffers into a ring buffer, which can hold approximately twenty events, depending on their size. A third task is in charge of the actual event building (EB), which means data formatting and delivery to the Taxi board of the Central DAQ or optionally a monitoring file or IP client.

We recorded fig. 4 as a benchmark for the new CaloDAQ. The slope of the

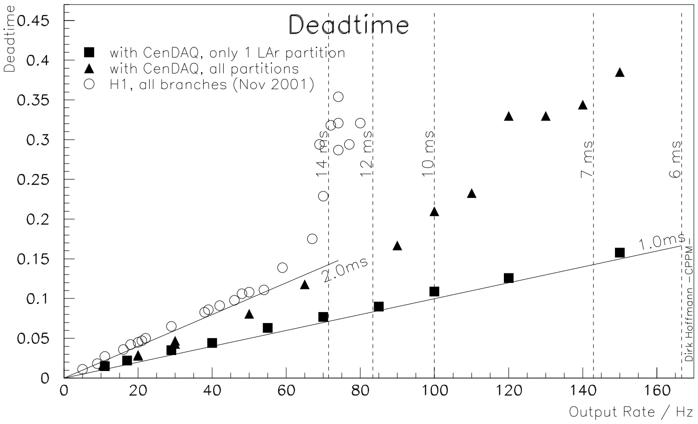

Figure 4. Deadtime behaviour of a single CaloDAQ partition and of the complete CaloDAQ system in comparison to the behaviour of the DAQ for the whole experiment: The first order or synchronous deadtime leads to a linear rise for moderate rates, while events are sufficiently distant in time to allow for the asynchronous tasks to be executed in between. The hypothetic lines for 1.0 and 2.0 ms are drawn for comparison. The available time for the secondary tasks becomes sparser with increasing datataking rate; the deadtime "clambers the wall of saturation". The maximum rate is determined by the average overall time for treating a single event. The theoretical limits for some values are drawn vertically as guidelines.

first order deadtime for moderate rates is in agreement with the independently measured deadtime of less than 1.2 μs. Inclusion of all partitions yields a performance which is better than the overall experiment limitations. Saturation presently occurs at more than 400 Hz for the ADC/TDC branch and 210 Hz for the Trigger branch.

4. User Interface

A graphical user interface for a comprehensive series of standalone tests and hardware check programs has been implemented in Java. The physical and statistical evaluation of the results is done with routines of the Java Analysis Studio[14].

It performs also the regular calibration level ramps including an elaborate algorithm for sequential pulsing of the generators, in order to avoid crosstalk effects. Sums over 100 events at the same level are performed in the DSPs, which minimizes data transmission over the VICbus. The present limitation is however exactly the speed of this transmission.

5. Conclusion

The speed improvement by a factor two is a convincing and desired achievement. The performance is now limited by the unchanged ADC/DSP architecture and VME interconnection on the front end. The manifest benefit of the upgrade is the innovative and nevertheless 100% backward compatible redesign of the core readout and control software of the CaloDAQ in an integrated LynxOS-Solaris Unix cluster, using exclusively Unix standards and conforming wherever possible to POSIX conventions. It puts a comfortable environment for the development of future survey and analysis programs at the disposal of the expert physicists.

Acknowledgements

We thank all our colleagues between the Mediterranean and the Baltic Sea, who participated with smaller and bigger contributions in the preparation and completion of our data acquisition system. We would like to mention in particular Jean-Baptiste Chabane for his work on the Java user interface, Daniele Laugier for the fundamental studies on the Unix data transfer, Sepp Huber, Tibor Kurča and Burkhard Reisert for the implementation of the trigger loading software in the new framework, František Kriván for the continuous support for the frontend hardware and the Sun cluster service, and last but not least Etienne Barrelet for the original plan of "our new DAQ", his unequaled way of thinking and his continuous supply with new ideas.

References

1. H1 Collaboration, The H1 Detector, Preprint DESY-H1-96-01 and Nucl. Inst. and Meth. **A 386**, 310–347 & 348–396 (1997)
2. L. Suszycki et al., HERA Luminosity Upgrade, DESY-Report (July 1997); updated version available at
   ▷ `http://www.desy.de/~parker/IR/Report970702/Report970702.html`
   The HERA Luminosity Upgrade, ed. U. Schneekloth, DESY HERA 98-05 (1998)
3. F. Descamps, C. Vallée, Data Acquisition for the H1 Calorimeters, Internal note H1-10/92-256 and Proc. of the III. International Conference on Calorimetry in HEP, Corpus Christi (1992)
4. Digital Signal Processor Board DSP 8510, User's Manual, Revision 0.2, Creative Electronic Systems, Geneva (1990)
5. Motorola DSP56000 Digital Signal Processor, User's Manual, Motorola Inc. (1986)
6. T. Carli et al., Performance of the H1 LAr Trigger in 1994, Internal note H1-07/95-445 and Contr. to the Int. Europhys. Conf. on HEP, EPS0644, Brussels (1995)
7. F. Blouzon et al., Manuel utilisateur du DSPT et de la carte t_0, LPNHE Paris, unpublished (1992)
8. W. J. Haynes, Experiences at HERA with the H1 Data Acquisition System, Proc. Int. Conf. Computing in HEP, Annecy, France, 151–161 (1992)
9. VIC 8250 VMV to VME One Slot Interface, User's Manual ver. 5.0, DOC 8250/UM, Creative Electronic Systems, Geneva
   VICbus, VME Inter-Crate Bus, ISO/IEC JTC 1/SC26, ISO/IEC 11458
10. H1 Coll., Proposal to Upgrade the LAr Calorimeter Trigger: The Jet Trigger, DESY-PRC-99-02 (1999)
11. PVIC Systems, PCI to PCI connections, User's Manual, v. 4.0, DOC 3925A/UM, Creative Electronic Systems, Geneva (2000)
12. RIO2, PowerPC based RISC I/O Board, User's Manual, v. 7.0, DOC 8062/UM, Creative Electronic Systems, Geneva (2000)
13. LynxOS releases 3.0.1 and 3.1.0 by Lynux Works$^{\mathrm{TM}}$, presented on
    ▷ `http://www.lynxos.com`
14. Java Analysis Studio, Java library for physics analyses; available from
    ▷ `http://www-sldnt.slac.stanford.edu/jas/`
15. V. Blobel, The BOS System. Dynamic memory management (1977)
16. G. Cozzika, The H1 Detector, Proc. of the III. International Conference on Calorimetry in HEP, Corpus Christi (1992)
17. M. Fleischer, The Performance of the H1 Liquid Argon Calorimeter, Proc. of the VII. International Conference on Calorimetry in HEP, Tucson (1997); Talk and Proc. available from
    ▷ `http://www-h1.desy.de/calo/conf`
18. Institute of Electrical and Electronics Engineers (IEEE), Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API), IEEE 1-55937-061-0; ISO 9945-1 (1990) "POSIX 1"
    IEEE, POSIX — Part 2: Shell and Utilities, IEEE 1-55937-255-9 "POSIX 2"
    IEEE, POSIX — Part 1: API — Amendment: Realtime Extensions, IEEE 1-55937-375-X "POSIX 4"