The Use of Commodity Products in the ATLAS Level-2 Trigger

M. Dobson[2] *on behalf of the ATLAS level-2 trigger groups*

[1] Argonne National Laboratory, USA
[2] CERN, Geneva, Switzerland
[3] DAPNIA, CEA Saclay, France
[4] Henryk Niewodniczanski Institute of Nuclear Physics, Cracow, Poland
[5] Michigan State University, East Lansing, MI, USA
[6] Niels Bohr Institute, Copenhagen, Denmark
[7] NIKHEF, Amsterdam, Netherlands
[8] Royal Holloway, University of London, U.K.
[9] Rutherford Appleton Laboratory, U.K.
[10] Università di Roma 'La Sapienza', Italy
[11] Universität Mannheim, Germany
[12] University College London, U.K.
[13] University of Liverpool, U.K.
[14] University of Manchester, U.K.
[15] University of Wisconsin, USA
[16] Weizmann Institute of Science, Rehovot, Israel

Abstract

The ATLAS level-2 trigger has to offer an event rate reduction of approximately 1 in 100, from an input rate of up to 100 kHz. Studies indicate that using geometrical guidance from the level-1 trigger and a sequential selection strategy, this can be achieved using largely commodity products, both for the processors and the communication networks. This paper will present the results of recent studies, indicating where commodity items are now sufficiently powerful and flexible to be used for this demanding real-time task and where custom items — either software or hardware — may still be required.

Keywords: ATLAS, Level-2, Trigger, Commodity

1 Introduction

The high rate of interactions in future LHC experiments places stringent demands on the trigger and data acquisition systems. The ATLAS experiment uses a three-tier trigger [2]. Level-1 [1] is based on custom hardware to reduce the trigger rate from the 40 MHz bunch-crossing rate to below 100 kHz. After a level-1 ACCEPT decision, all data for that event (1–2 MByte per event) are sent to readout buffers (ROBs) for temporary storage. The higher-level triggers (level-2 and Event Filter, EF) are required to reduce the event rate for permanent storage to ~ 100 Hz.

The level-2 trigger uses Regions of Interest (RoIs) guidance from the level-1 trigger to reduce the amount of data requested from the ROBs and to reduce the processing power and network bandwidth required. Further computing-power and data-volume reduction is possible by using a sequential selection strategy. This strategy involves processing the data in the RoIs initially from only one or two subdetectors, and taking a decision to continue or abandon the event before processing the data in the RoIs from the other subdetectors. After each step, the event rate is reduced leading to a cut in total data volume and processing power required.

This paper presents the results of studies which indicate where commodity items are now or soon will be sufficiently powerful and flexible to be used for the demanding real-time task of the level-2 trigger, and where custom items — either software or hardware — may still be required.

2 The Level-2 Trigger and the Testbeds

2.1 The Level-2 Trigger

The level-2 trigger system has four different components, the ROBs, the processors, the RoI builder and Supervisor, and the network (see figure 1).

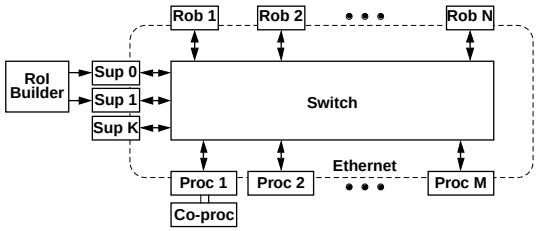

Figure 1: The level-2 architecture.

The RoI builder [3] is custom hardware designed to combine RoI fragments from the level-1 processors into one event record, which it passes on to the supervisor farm. The latter is made up of general-purpose processors with a simple custom input card for receiving the event records. The supervisors assign an event to one of the processors. The processor, possibly helped by a co-processor, performs one or more steps of data collection and analysis from relevant ROBs. The trigger decision can be issued at any step. It is returned to the supervisor which distributes it to the ROBs. Rejected events are discarded; accepted events are passed on to the EF for further analysis

2.2 The Testbeds

Prototypes referred to as Testbeds have been set up to check that individual components (commodity items wherever possible) meet the required performance; provide information on scaling up to moderate size systems; and to provide data for the full system computer models.

The Testbed systems vary in size from 25 to 50 nodes, with a maximum of 96 nodes for the commercial cluster at Paderborn University [1]. These systems correspond to a few percent of the final ATLAS system. Ethernet (Fast and Gigabit), ATM and SCI technologies have been studied for the network. All the Testbeds follow Level-2 trigger architecture shown in figure 1.

The Ethernet and ATM Testbeds (see figure 2) share the same PCs, which are single or dual-processor machines with processor speeds from 200–450 MHz. The ATM Testbed also uses ten PowerPC single-board computers running LynxOS. The network equipment for ATM is a 48-port, 155 Mbit/s per port, FORE switch. For Ethernet it is three BATM Titan 4 Fast or Gigabit switches with up to 32 Fast ports or 4 Gigabit ports per switch.

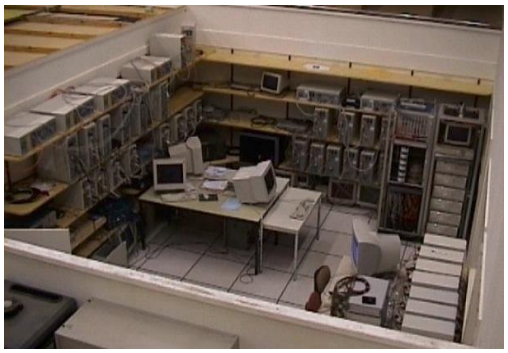

Figure 2: The Ethernet and ATM Testbeds.

The SCI Testbed has 23 single or dual-processor PCs with processor speeds of 300–450 MHz. The SCI switch is a 16 port Dolphin switch. The Siemens cluster at Paderborn University with 96 nodes, interconnected with SCI technology has also been used.

The OO C++ prototype level-2 software, called the reference software [2], has been run on all Testbeds under Linux and Windows NT, and on Solaris at Paderborn. On the ATM Testbed, earlier optimised C based software [4] has been run under Windows NT, Linux and LynxOS.

3 Results

3.1 Supervisor Performance

The supervisor tasks are to get an event from the RoI builder, allocate it to a processor, receive the decision back, update the statistics, pack the decisions and multicast them to the ROBs. To study the supervisor performance, the system

[1] http://www.uni-paderborn.de/pc2/systems/psc/index.htm

[2] http://www.cern.ch/Atlas/project/LVL2testbed/www/

is configured so that it saturates the supervisor. The rate achieved as a function of the number of RoIs in the event record is shown in figure 3. With a single RoI, a rate of ~ 11 kHz per supervisor emulator is reached. The rate is independent of the number of ROBs when a hardware multicast is used. The rate versus the number of level-2 processors increases linearly until the supervisor is saturated. The results also show that the system rate scales with the number of supervisors and a rate of 120 kHz was achieved with 12 supervisor emulators (no RoI builder) on the Paderborn cluster. Up to 100 kHz event rate was also achieved with the RoI builder and 4 supervisors in the ATM Testbed [4].

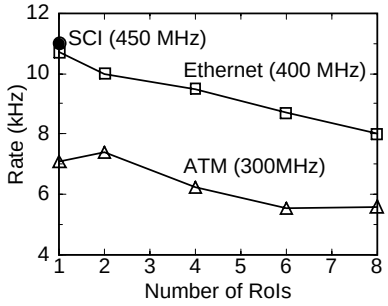

Figure 3: Supervisor rate versus number of RoIs in the event record.

3.2 ROB Access Performance

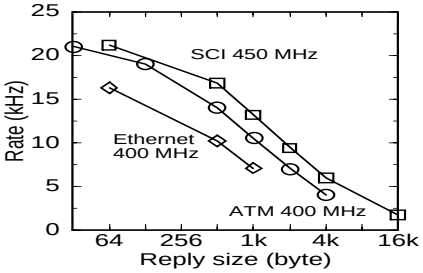

Figure 4: ROB request rate versus data reply size.

ROB performance tests use a ROB emulator running on a PC. The system is configured to saturate the ROB, and the rate of requests for many processors which can be met by a single ROB is shown in figure 4. The performance is consistent with that expected for real ROBs and used in system models [5].

3.3 Processor Performance

The first task of the processor is to collect data from many sources, which can be very demanding, and the second task is to process the data received. The emphasis in the Testbeds has been on the first task. The system was configured to saturate a processor using up to 16 ROBs. When collecting 64 Bytes from each ROB (see figure 5), the processor can sustain a rate of 5–7 kHz for 1 ROB, down to 1 kHz for 16 ROBs. For a typical RoI spanning 4 ROBs, and a total level-1 rate of 100 kHz, the data collection overhead requires CPU cycles equivalent to ~ 30 of todays processors. This is acceptable in view of the total farm size that is envisaged.

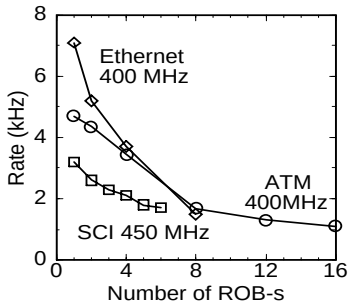

Figure 5: Processor event rate versus number of ROBs in the RoI.

The data size collected from the ROBs can also be varied. For the collection of 1 kByte and 4 kByte data sizes from one ROB, a rate of 4 kHz and 2.5 kHz is achieved respectively.

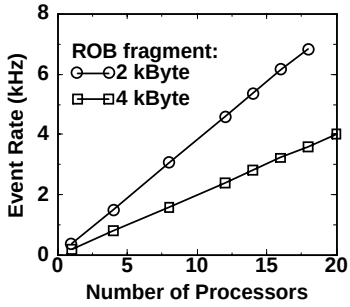

Figure 6: System event rate versus number of processors, for collection from 20 ROBs.

In addition to the RoI data collection, some algorithms need to scan a complete subdetector for tracks. This data collection was simulated

on the ATM Testbed by collecting data from 20 ROBs (see figure 6). The event rate scales linearly with the number of processors and a bandwidth of 260 MByte/s and 328 MByte/s was achieved respectively for 2 kByte and 4 kByte data fragments. Some algorithms will require data collection from up to several hundred ROBs. FPGA co-processors [6] are being investigated to handle the associated computing intensive algorithms.

3.4 System Performance

A large system operated on the Paderborn cluster has shown: linear scaling of the system rate with the number of ROB/processor pairs added; stable performance of the supervisor versus the number of processors; and correct operation of the reference software on a large system.

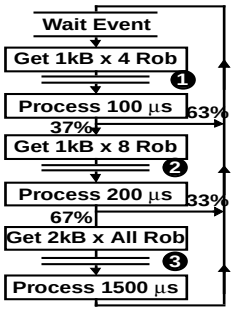

Figure 7: Selection strategy used for measuring the latency of figure 8.

The use of sequential selection reduces the network and processor requirements and allows more complex algorithms to be run at lower rates. A test selection strategy is shown in figure 7. The corresponding latency plot obtained with [4] on the Testbed is shown in figure 8. The structure in the plot reveals the different algorithm steps in the sequential selection strategy.

4 Conclusions

The level-2 strategy and architecture have been implemented on moderately large Testbed systems with success. The performance of the different components has been measured and it is now clear that commodity products can be used for the majority of the level-2 components (OS, supervisor, processors and network technology). It may, however, be necessary to use custom items for the drivers and software, and for the co-processors. The RoI builder is implemented in custom hardware because of the need to combine seven high-rate data streams. It is, however, too early to conclude if commodity items can be used within the ROBs.

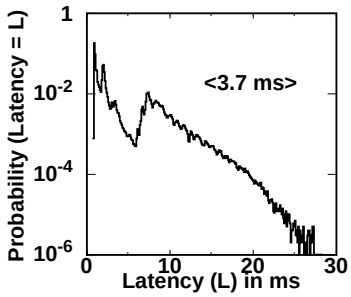

Figure 8: Event latency probability when using sequential selection.

References

1 ATLAS Collaboration, “Level-1 Trigger Technical Design Report”, CERN/LHCC 98–14, CERN, June 1998.

2 ATLAS Collaboration, “ATLAS Trigger Performance Status Report”, CERN/LHCC 98–15, CERN, June 1998.

3 R. Blair et al., “A Prototype RoI Builder for the Second Level Trigger of ATLAS Implemented in FPGA’s”, LEB’99, Snowmass, September 20–24 1999.

4 D. Calvet et al., “Operation and Performance of an ATM based Demonstrator for the Sequential Option of the ATLAS Trigger”, IEEE TNS vol. 45 pp 1793–1798, August 1998.

5 M. Dobson et al., “Paper Models of the ATLAS Second Level Trigger”, ATLAS Internal Note, ATL-DAQ-98-113, June 1998.

6 J. Hesser et al., “ATLANTIS – A Hybrid Approach Combining the Power of FPGA and RISC Processors based on Compact PCI”, proceedings of 7th Reconfigurable Architectures Workshop (RAW 2000), Cancun, Mexico.