THE DATA-FLOW SYSTEM OF THE ATLAS DAQ AND EVENT FILTER PROTOTYPE "-1" PROJECT

M. Niculescu[c,e], E. Arik[b], G. Ambrosini[c], H-P. Beck[a], S. Cetin[b], T. Conka[b], A. Fernandes[c], D. Francis[c], Y. Hasegawa[d], M. Joos[c], G. Lehmann[a,c], J. Lopez[c], A. Mailov[b], L. Mapelli[c], G. Mornacchi[c], Y. Nagasaka[f], K. Nurdan[b], J. Petersen[c], D. Prigent[c], J. Rochez[c], R. Spiwoks[c], L. Tremblet[c], G. Unel[c], Y. Yasu[g]

a. Laboratory for High Energy Physics, University of Bern, Switzerland
b. Department of Physics, Bogazici University, Istanbul, Turkey
c. CERN, Geneva, Switzerland
d. ICEPP, University of Tokio, Tokio, Japan
e. Institute of Atomic Physics, Bucharest, Romania
f. Nagasaki Institute for Applied Science, Nagasaki, Japan
g. National Laboratory for High Energy Physics (KEK), Japan

Abstract

A prototyping project has been undertaken by the ATLAS DAQ and Event Filter group to design and implement a fully functional vertical slice of the ATLAS DAQ and Event Filter. It supports the evaluation of hardware and software technologies as well as their system integration aspects.

This paper describes the Data-flow component, its design, implementation and performance

1 INTRODUCTION

The final design of the Data Acquisition (DAQ) and Event Filter (EF) system for the ATLAS experiment at the LHC is not scheduled to start before 2005. The ATLAS/DAQ group has addressed the design of the ATLAS DAQ system by building a fully functional prototype consisting of a complete "vertical slice" of the ATLAS DAQ/EF architecture [1]. It includes all the elements of an on-line system, from detector read-out to data recording. Since it is understood that this prototype will not fulfil the final performance requirements it has been given the name DAQ/EF prototype "-1".

The DAQ/EF prototype –1 architecture includes a component which is responsible for receiving and buffering event fragments, event building and mass storage [2]. This logical component, called the Data-Flow is shown schematically in Figure 1.

2 FACTORISATION OF THE DATA-FLOW SYSTEM

Three main functions are provided by the Data-flow : the collection and buffering of data from the detector (Front-End DAQ), the merging of fragments into full events (the Event Builder) and the flow of full events through the Event Filter farm (Farm DAQ).

The segmentation of the detector read-out suggests to organise the Front-End DAQ into a number of modular, independent elements each supporting the read-out from one or more detector segment and having one or more connections to the Event Builder: the read-out crates (ROCs). Similarly the Farm DAQ is seen as a set of logically independent elements, each corresponding to an Event Builder output. The Event Builder combines the various "modules" into a complete data-flow system.

2.1 The Read Out Crate

The ROC (Figure 1) provides the following functions:

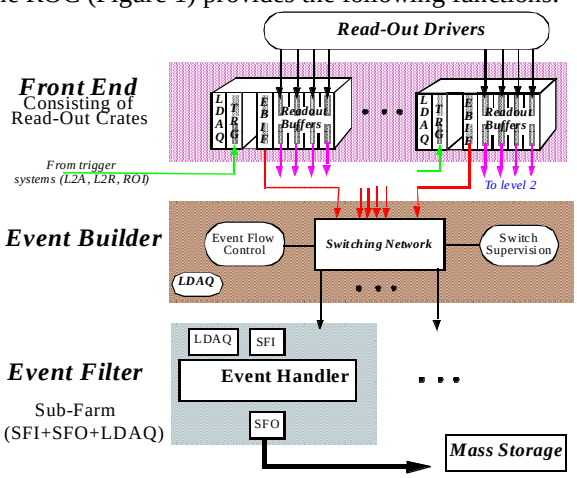

Figure 1:DAQ/EF Data-flow architecture

- Detector read - out, buffering and data distribution to other Data-flow elements in the crate. This function is provided by the read-out buffer (ROB) element.
- The control of the flow of data within the crate (e.g. when event fragments buffered in the ROBs have to be discarded or moved to the EB according to the response provided by a data reduction device). This is provided by the trigger (TRG) module.
- Fragments of accepted events are moved from the ROB memories and merged into a "crate fragment" (consisting of all the elementary fragments from the individual ROBs). Crate fragments are buffered and then sent to the Event Builder via the event builder interface (EBIF) element.
- More ancillary functions must also be provided locally in the crate: control of the crate, errors handling,

support for monitoring the system behaviour and event data. Also an interface point to the overall DAQ control system is needed. The component which is assigned to all these tasks is the Local DAQ (LDAQ).
An important role is played by the intra-crate links (e.g. the one connecting the ROBs and the EBIF to support the data collection function) and the related communication protocols (e.g. the data collection protocol).

2.2 The Event Builder

The Event Builder (EB) merges the event fragments, with the same event ID[1], from all the ROCs into a complete event at a sub-farm DAQ.The EB is built around a switching network which allows the concurrent merging of events. One of the objectives of the project is to use different commercial technologies for the switching network. To this end, the event builder is partitioned into two layers:

- a *technology-independent layer*, which implements the event building protocol (e.g. it determines which destination Sub-Farm will receive a given event). It consists of the Data-flow Manager (DFM), which implements the high-level event building protocol, and source and destination processes which are responsible for sending/receiving the event.

- a *technology-dependent layer* which interfaces, in a common way, the technology-independent components to the features of the switching hardware.

Ancillary functions, such as local control and monitoring of the behaviour of the event builder as well as interfacing to the overall DAQ control system, are the responsibility of an LDAQ module.

2.3 The Sub-Farm DAQ

The Sub-Farm DAQ receives full events from the EB, buffers and sends the events to the EF. It then stores on permanent mass storage the events produced by the EF. The element between the event builder and the event filter is called the Switch to Farm Interface (SFI). The element sitting between the event filter and the mass storage is defined as the Sub Farm Output (SFO).

3 DATA-FLOW IMPLEMENTATION

Predesign analysis and prototypes lead to the selection of basic hardware and software components. In the area of the ROC the VMEbus is used as the crate integration bus as well as the initial implementation of the intracrate links. PCI has been selected as the I/O integration bus within a module (ROB, EBIF, SFI, etc.). The PMC format is the preferred one for I/O interfaces. The modules are currently implemented by VMEbus, PowerPC based processors with two (or more) PMC sites. LynxOS is the preferred operating system, while Linux is progressively taking more place. Intracrate communication protocols have been defined and prototyped to support both the local DAQ (e.g. control transactions between LDAQ and a ROB) and the flow of data within a ROC (e.g. the data control messages exchanged between the TRG module and the EBIF). The PVIC bus [6] may be used as an alternate intra-crate link.
A suitable event builder protocol has been designed [3] and implemented [4]. We had initially chosen ATM, Fibre Channel and Switched Ethernet as the switching technologies onto which to implement the event builder. Fibre Channel development has then been frozen in favour of Gigabit Ethernet.
Based on the above hardware choices and software basis, the dataflow components have been implemented. First at the level of elements, such as a ROB, and then at the level of a ROC, event builder and subfarm DAQ. The integration of the full dataflow system has then taken place. A laboratory implementation on a 2x2 configuration (two ROCs , 2 sub-farms and an event builder) is available since several months for functionality and performance studies.

4 PERFORMANCE MEASUREMENTS

4.1 The Read-Out Crate

The global performance of the ROC is assessed by measuring the rate of events flowing through the crate. The performance depends strongly on the efficiency of the message passing system and, consequently, provides a measurement of the latter. Measurements [8] have been made with a TRG, EBIF and up to five ROBs connected via VMEbus and PVIC. The event rate in the ROC has been measured [7] as a function of the number of ROBs in the crate in two configurations. In the first configuration, all communications are via VMEbus, while in the second the data control messages are exchanged via the PVIC.

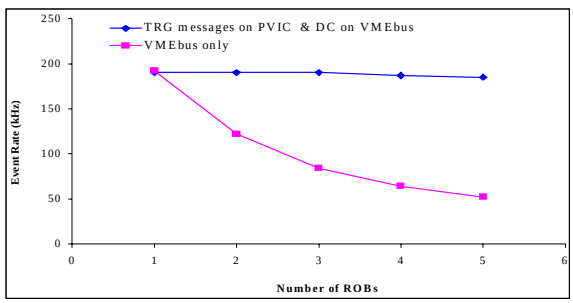

Figure 2: ROC performance measurements

The performance of the ROC is either CPU or I/O bound. In the first configuration, VMEbus only, and for more than one ROB, the TRG is I/O bound. The event

[1] By event ID we mean the value (over 24 bits) uniquely identifying an event provided by the Atlas level-1 trigger

rate decreases inversely to the number ROBs and is due to the TRG sending data control messages sequentially to the ROBs. However, in the case of only one ROB, it is the latter which is CPU bound. The ROB can only process events up to a rate of about 190 kHz. When data control messages are sent over PVIC, the event rate is almost independent of the number of ROBs (up to five) which clearly demonstrates the importance of sending data control messages over a bus with broadcast capability. The ROC performance is, in this configuration, determined by the performance of the ROB. The slight decrease in event rate for five ROBs suggests that at this point, the TRG is becoming I/O bound.

4.2 The Event Builder

The performance of the EB is assessed by measuring the rate of DFM assigned events as a function of the sub-event fragment size. The performance measurements have been done by using CES RIO2 8062 (200 MHz) processors and an ATM based switching network with the AAL5. A second configuration is based on 450 MHz Pentium III based PCs running Linux and a Gigabit Ethernet based switching network, accessed through the TCP/IP protocol stack.
AAL5 protocol on ATM allows to reach the nominal link speed (155 Mbit/s). The EB implemented with ATM shows three clearly distinct regions of behaviour: the first one, for message sizes below 1KB, dictated by the software overhead, the second one, between 1KB and 6KB, limited by the memory copy speed and the third, above 6KB, limited by the ATM link speed. Figure 3 summarises results for various configurations of a 4 node setup.
TCP/IP over Gigabit Ethernet (Figure 4) identifies two different regions of behaviour. The first one, up to 8KB message size, is constant, and is dominated by the software overhead, in particular by the performance of the TCP/IP stack. The second one, instead, indicates that the limit of the data transfer speed is reached. Indeed with the available PCs, using a 32bit PCI bus, we experience a maximum throughput of 350 Mbit/sec. on a point to point connection using the standard Ethernet frame size of 1500 bytes.

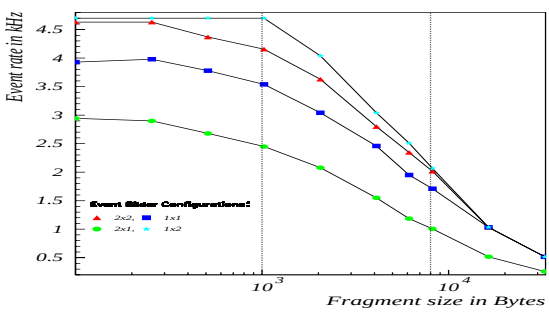

Figure3: EB prototype with ATM

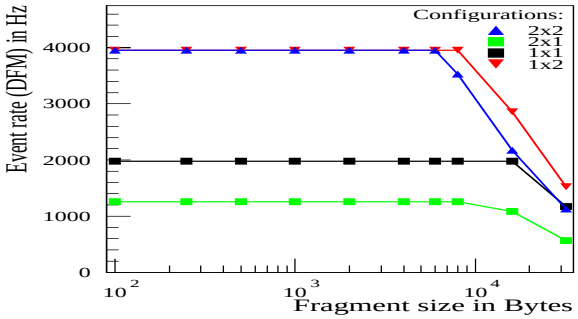

Figure 4: EB prototype with Gigabit Ethernet

5 SUMMARY AND CONCLUSIONS

After a phase of designing and prototyping a full Data-flow system has been implemented. It is configured with two ROCs one event builder implemented with one of the two possible technologies (Fast Ethernet or ATM) and two Sub-farms (with a dummy event handler). An event builder based on PCs connected by a Gigabit Ethernet switching network has also been setup.
The purpose of the Data-flow setup is twofold: to provide a means for studying Data-flow functional and performance issues and to be the basis for the integration into a fully functional DAQ/EF "-1" system.
The performance studies have shown that, for small scale configurations, currently available technology is close to providing a performance adequate for the final ATLAS system [9]. These studies have also suggested a number of places in the system where optimisation via appropriate hardware (such as broadcast support in the ROC, more powerful I/O busses on the Event Builder nodes) and software (such as protocols performance wise enhancement in performance.

REFERENCES

[1] G. Ambrosini et. al., The ATLAS DAQ and Event Filter Prototype "-1" Project, presented at Computing in High Energy Physics 1997, Berlin, Germany. http://atddoc.cern.ch/Atlas/Conferences/CHEP/ID388/ID388.ps

[2]http://atddoc.cern.ch/Atlas/Notes/069/Note069-1.html
[3]http://atddoc. cern.ch/Atlas/Notes/042/Note042-1.html
[4]http://atddoc.cern.ch/Atlas/Notes/069/Note069-1.html
[5]http://atddoc.cern.ch/Atlas/Notes/104/Note104-1.html
[6]http://www.ces.ch/Products/Connexions/PVICFamily/PVIC.html

[7]http://atddoc.cern.ch/Atlas/Notes/120/Note120-1.html
[8]http://atddoc.cern.ch/Atlas/Notes/126/Note126-1.html
[9]The ATLAS Collaboration, Technical Proposal for a General Purpose pp Experiment at the Large Hadron Collider at CERN. CERN/LHCC/94-43