SCI Studies for the Level-2 Trigger System of the ATLAS Experiment

A. Bogaerts[2], D. Botterill[5], J. Dawson[1], E. Denes[2], F. Giacomini[2], A. Guglielmi[3], R. Hauser[2], R. Hatley[5], R. Hughes-Jones[4], S. Kolya[4], M. Liebhart[2,6], D. Mercer[4], R. Middleton[5], J. Schlereth[1], P. Werner[2], F. Wickens[5]

[1]Argonne National Lab, Illinois, US
[2]CERN, 1211 Geneva 23, Switzerland
[3]DEC Joint Project Office, CERN, 1211 Geneva 23, Switzerland
[4]University of Manchester, Manchester, M13 9PL, UK
[5]Rutherford Appleton Laboratory, Chilton, Didcot, Oxon OX11 0QX, UK
[6]T. U. Graz, Institute for Technical Informatics, 8010 Graz, Austria

ABSTRACT

The Scalable Coherent Interface (SCI) [1] is being investigated for applications in the Level-2 trigger of the ATLAS experiment at CERN's Large Hadron Collider (LHC). The system under study consists of a farm of commercial processors communicating over a high performance (200 MBytes/s) SCI network.

Tests have been made on configurations of up to ten SCI nodes representing a small slice of what would be required for a full system. The performance of components which represent key elements of the Level-2 trigger system have been measured in different configurations, with and without the inclusion of an SCI switch. Since the small slice should scale to a much larger system the impact of some forms of pipelining and parallelism has been studied. The results are presented.

This work is part of a more general programme within ATLAS to explore different architectures and technologies for the implementation of the Level-2 trigger system.

INTRODUCTION

ATLAS [2] is a general-purpose detector designed to study proton-proton collisions at the LHC. Bunches of protons running in opposite directions around the accelerator ring will cross at a frequency of 40 MHz (every 25 ns) and at each crossing several collisions will occur. Given the large number of electronic channels, the expected rate of data production is of the order of 10^{15} Bytes/s. A three-level trigger system will be used to filter these data to reduce them to a more manageable size for long term storage.

The Level-1 trigger accepts data at the full LHC bunch-crossing rate. Here special-purpose processors act on reduced-granularity data from a subset of the subdetectors and reduce the rate by a factor ~1000. Data which pass this selection are moved from the front-end electronics for each part of the detector to corresponding Read-Out-Buffers (ROB), where they are stored in a standardized format.

The Level-2 trigger is designed to reduce the event rate from ~100 kHz to ~1 kHz; it uses full-granularity and full-precision data from most of the subdetectors, but examines only regions of the detector identified by the Level-1 as containing interesting information (Regions of Interest or RoIs). Owing to this approach the Level-2 system needs to access only a small fraction of the total detector data, with corresponding advantages in terms of the required processing power and of data-movement capacity; nevertheless current estimates foresee that the system will require ~1000 processors and a network capacity of a few GBytes/s [3]. Based on the Level-2 decision, to be taken in ~10 ms, the ROBs discard the data or forward them to the third level of the trigger, the Event Filter.

Finally, the Event Filter, which acts on complete events, reduces the data-storage rate to 10-100 MBytes/s, by reducing the event rate and/or the event size.

Given the characteristics of SCI, in particular low latency and high throughput, this technology is a good candidate for use in the Level-2 trigger system.

A parallel push architecture is assumed: under the control of a Supervisor process [4], data are pushed from ROBs to Feature EXtractor local processors (FEX), that in parallel for each RoI

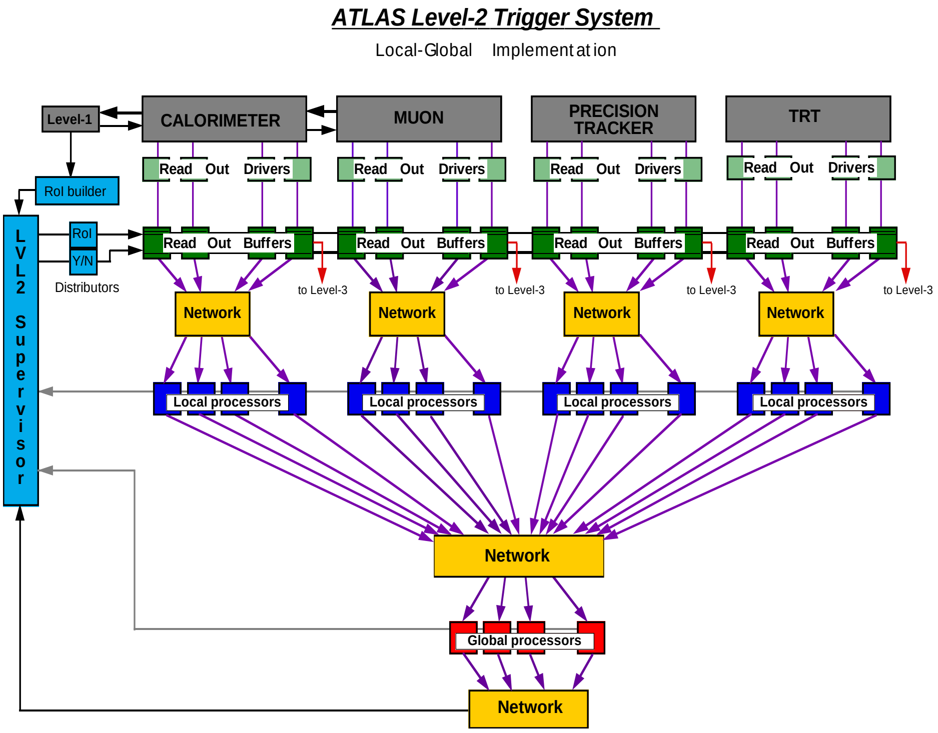

Figure 1. The Local-Global option of the ATLAS Level-2 trigger system.

and for each detector layer in the RoI determine specific characteristics of an event (e.g. particle energies and track parameters). The features from a single event are then passed to a Global processor, that combines them and generates a trigger decision and forwards it to the Supervisor, which then decides whether the event should be kept or discarded. This architecture, shown in Figure 1, is also referred to as the "Local-Global" option [5].

The first part of this paper contains a brief description of the SCI technology and an overview of the communication software used to run the system. The second part describes the studies made on a simplified version of the processor farm that will constitute the Level-2 trigger system.

Overview of the SCI Studies

The Scalable Coherent Interface is an IEEE standard for interconnecting multi-processor systems. An SCI network can be seen as a logical bus but it is actually constructed from unidirectional point-to-point links between processors. SCI uses a split transaction protocol and the communication is based on the exchange of small packets. An SCI packet has a 16-byte header, a 16- or 64-byte payload and a 2-byte CRC (the 256-byte payload allowed by the standard has not yet been implemented). The simplest possible configuration of an SCI network is a ring; more complex topologies are possible by interconnecting rings using switches. Custom interfaces exist, e.g. the PCI-SCI card developed at the Dept. of Physics of the University of Manchester. The results presented in this paper have been obtained using commercial PCI-SCI interfaces from Dolphin Interconnect Solutions based on their link controller LC-1 running at a link speed at 200 MBytes/s. In the near future the newer version with LC-2 supporting link speeds at 500 MBytes/s will be used both in the PCI and PMC (for VME-based processors) formats.

In these tests the FEX and the Global processors were Alpha computers from DEC of different clock speeds (AXPpci33 at 166 or 233 MHz and Multia at 166 MHz) running MicroC/OS, a small stand-alone real-time kernel [6]. The Supervisor processor and the ROBs were VME-based RIO-2s from CES (type 8061 and 8062) running LynxOS. To accept the PCI cards, RIO-2's were fitted with PMC-PCI adaptors (Technobox).

Although the SCI link speed is 200 MBytes/s, the bandwidth into the memory of a processor node is limited by the PCI bus, the highest bandwidth we have measured being 80 MBytes/s, achieved writing from an AlphaServer 4000, running at 300 MHz, to an AlphaStation 500, running at 400 MHz. However, the Alpha boards used in these tests transfer up to 50 MBytes/s; the RIO-2s up to 33 MBytes/s. SCI also has intrinsically a very low packet latency of ~2.5 µs between processes in two nodes.

Even though at the lowest level the communication is based on the exchange of packets, the Dolphin hardware offers different facilities to send and receive data.

The options on the sending node are:

- transparent mode: the CPU writes into its virtual memory. The memory management hardware maps this address onto the PCI card which in turn sends SCI packets destined to the remote node. The only software intervention is a possible barrier-like operation to flush outstanding buffers and check for transmission errors.

- DMA mode: the CPU loads the DMA engine on the PCI-SCI card within the physical memory location, the length of the data to be transferred and the SCI destination address. The interface then, independently from the CPU, fetches the data from memory, creates and transmits SCI packets as required. Status registers provide information on the progress of the transmission and on any errors.
- packet mode: raw SCI packets are constructed by the CPU and sent over SCI by the interface.

The options on the receiving node are:

- transparent mode: the interface places the incoming data directly in the memory location whose address is specified in the SCI packet.
- ring buffer mode: the interface places the incoming raw SCI packets into a user specified ring buffer in memory, from where the application software has to extract them.

Two combinations of the above options have been evaluated in these tests:

- transparent mode - transparent mode, or remote shared memory. In this scheme both the sender and the receiver use transparent mode: when the sender writes into its own address space, the operation is automatically converted by the hardware into a write into the receiver memory.

  A message passing library has been implemented over shared memory. It provides synchronisation between source and destination, with a minimal flow control to avoid a source overrunning the destination. Between an AlphaServer 4000, running at 300 MHz, and an AlphaStation 500, running at 400 MHz, the use of the message passing library allows an effective throughput of ~70 MBytes/s and causes a message overhead of ~7.5 µs.
- DMA mode - ring buffer mode. In this scheme the sending CPU sets up the DMA engine, which transfers data to the receiving node. Here the CPU polls a CSR register to check if any packet has been placed by the SCI interface into the ring buffer; if so it extracts and manages the packet.

  The main advantage of this approach is that the sender does not have to know the remote memory address of the receiver since the ring buffer is addressed using a fixed CSR location. On the other hand, setting up the DMA engine requires a time overhead, that, especially for small messages, can be significant.

Although commercial software is becoming available for more and more platforms, all the software used in these tests, including simple device drivers for the different platforms, has been developed internally. Alternative software is being developed under the EU funded SISCI ESPRIT project (Software Infrastructure for SCI) [7] with the aim to provide a standard low-level API [8] in a heterogeneous SCI environment as well as high-level communication packages such as MPI.

Vertical Slice Tests

A vertical slice (Figure 2) represents a small subset of a full Level-2 trigger system used to test key elements of the required functionality. The results obtained from transparent and DMA modes were very similar, within the uncertainties caused by different processor types.

In the vertical slice the basic event sequence is as follows: details of the RoI (i.e. the position within the detector which identifies the ROBs containing data needed to make the Level-2 decision) are generated either by the Level-1 emulator, that sends them to the Supervisor processor via the Input Router, or by the Supervisor itself, which reformats them into RoI Requests to be sent to ROBs. The Supervisor tags the requests with a Global processor identifier indicating which one will be used. If no Global processor is available, the Supervisor waits until one becomes free. The request is sent via the Output Router and S-link [9] to the RoI Distributor which in turn transmits it to the required ROBs via the VME backplane. The ROBs send pre-loaded “event” data of a specified length to the FEX allocated via a lookup table in the RoI Distributor using the RoI position (multiple RoIs in a single event require a FEX each); all FEXs working on the same event send a short “feature” message (64 bytes) to the assigned Global processor. The Global processor combines all of the features of the event and generates an “event decision” which it sends to the Supervisor Network Server. The Network Server manages the network connection and passes the message details to the Supervisor processor. The Supervisor notes that the Global processor is free and uses the identifier for a new event. It also groups event decisions to avoid sending messages to all ROBs at the full rate of potentially 100 kHz. When a sufficient number of event decisions have been grouped together, the Supervisor sends the grouped decisions via the Output Router and S-link to the RoI Distributor for onward transmission to the ROBs. The ROBs then releases the event buffer (in the final system they would transmit accepted events to the Event Filter).

Since the purpose of the test was to verify correctness and robustness of protocols and to measure the performance of the data communication, no physics algorithms were applied to the events and no data manipulation was applied to the message contents.

Two parameters were used to characterize the performance of the system: the event latency and the average time per event. The former is defined as the time from when the Supervisor assigns a Global processor to an event to the time it receives the decision from the Global farm. The latter is the average time between two trigger decisions taken by the Supervisor and it is measured dividing the duration of the test by the number of events that have flown through the system. The values of the two parameters differ considerably when more than one event

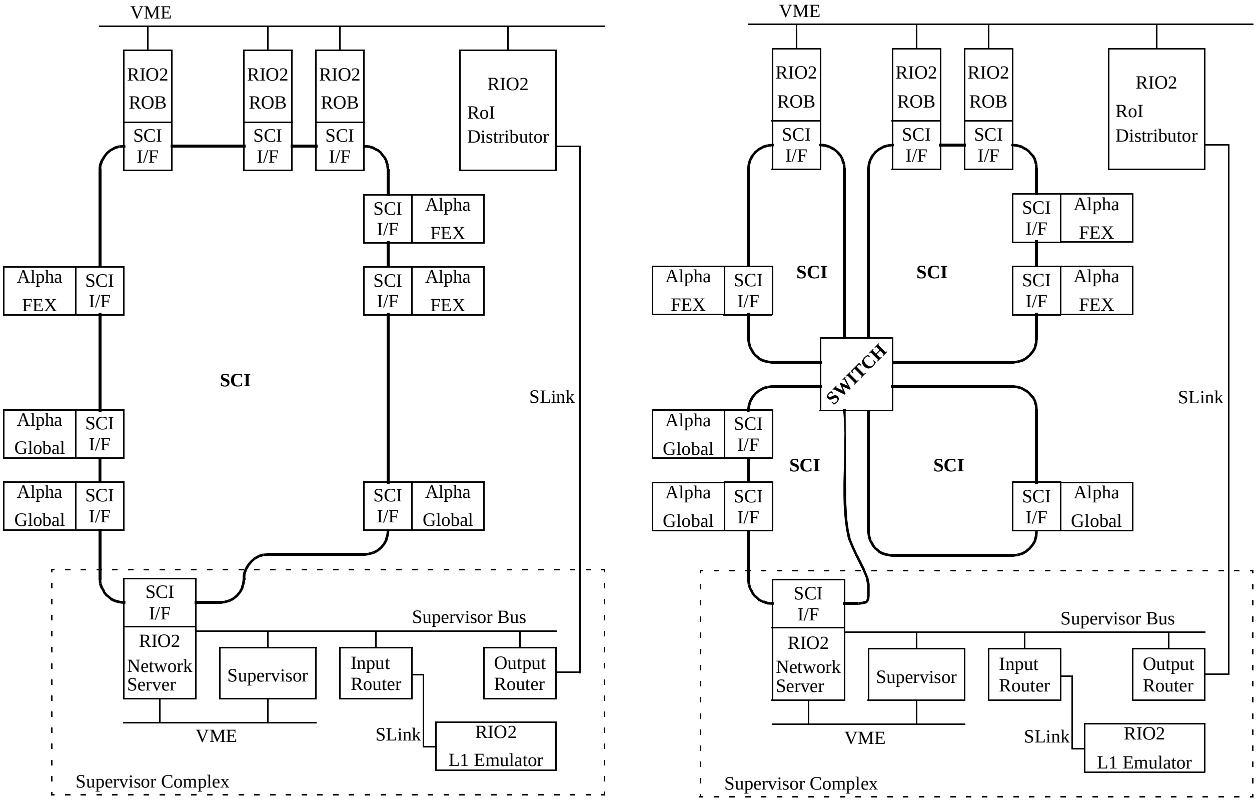

Figure 2. Schematic of a vertical slice of the ATLAS Level-2 trigger system with three ROBs, three FEXs, three Globals, the RoI Distributor and the Supervisor complex. The processor nodes could be arranged in a single ring (left) or in four ringlets connected via a 4-port switch (right).

is allowed in the system at the same time.

In the following sections four different aspects of the system are considered and their impact on the time-per-event parameter is shown:

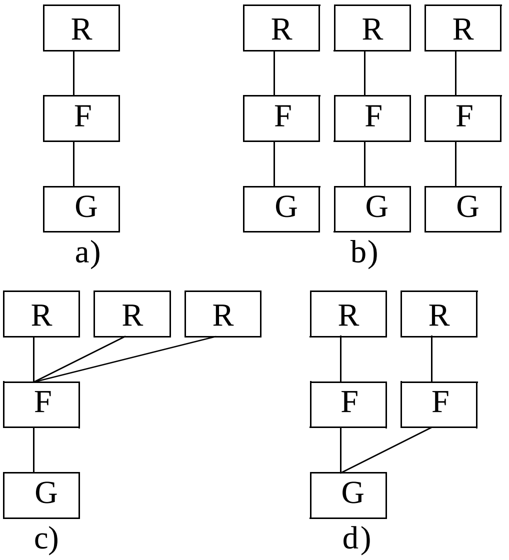

Figure 3. a) pipeline b) event parallelism c) RoI fragment building d) RoI parallelism

- inherent pipeline structure of the Supervisor - RoI Distributor - ROB - FEX - Global - Supervisor chain, when multiple events are allowed to enter the system quasi simultaneously (Figure 3a);
- event parallelism, with multiple ROB - FEX - Global streams running under the control of a common Supervisor (Figure 3b);
- RoI fragment building, allowing several fragments from different ROBs to be sent in parallel to the same FEX (Figure 3c);
- RoI parallelism, allowing several FEXs (each possibly receiving data from multiple ROBs) to analyse multiple RoIs of several detectors of the same event in parallel (Figure 3d)

The latency and the time per event can also be affected by other factors that have been investigated: the size of the messages transferred between a ROB and a FEX and the introduction of an SCI switch in the system.

Pipeline

A single stream is constituted by a ROB, a FEX and a Global; for this test each event has only one RoI and this RoI is contained in a single ROB. The stream is initiated by the Supervisor when an RoI record of an event is received from Level-1

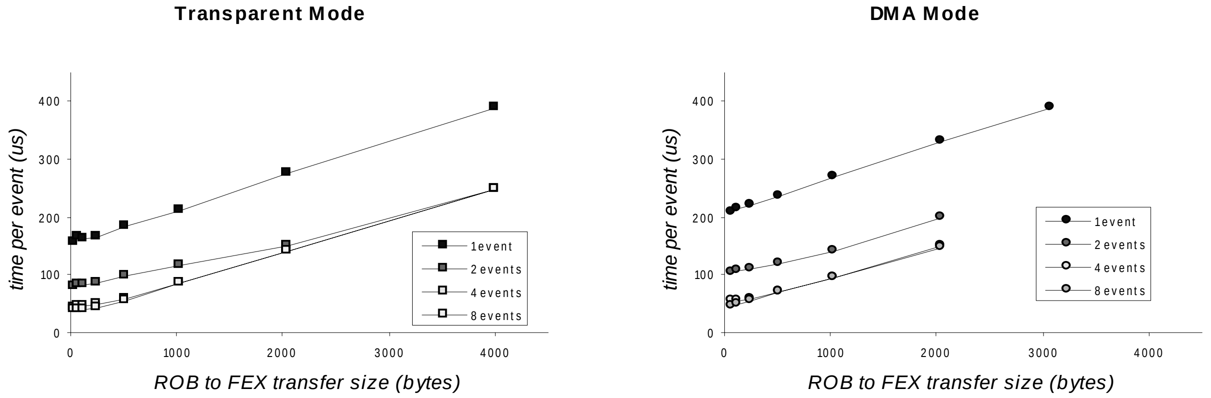

Figure 4. Pipeline. Multiple events are allowed to enter the system at the same time.

and is terminated when the Supervisor receives the trigger decision from the Global processor.

Since the stream has an intrinsic sequential structure and the processing steps for an event correspond to the stream stages, several events can be pipelined in the system. In Figure 4 the dependence of the time per event on the size of the data transferred from ROB to FEX is shown for different numbers of events allowed in the stream.

For one event in the system the time per event is determined by the total loop latency. If two events are allowed in the stream at a time, they are distributed around the system with no queues forming (i.e. no increase in latency) until the data size is slightly over 2 kBytes and the time per event just scales.

For longer events or more than two allowed in the system a queue forms at the slowest element and the rate is limited to the speed of this element. For most event lengths, the slowest element is the ROB to FEX transfer (with an effective bandwidth of 18-20 MBytes/s), but for very short events it is the RoI Distributor. In addition, for these very short messages, there is a small but significant contention of the PCI bus on the RIO-2 of the ROB, because the RoI Distributor is accessing the ROB memory via the VME-PCI bridge while the ROB is transferring data to SCI via the PCI-SCI bridge; the contention slows down the RoI Distributor as the number of SCI packets increases.

Event Parallelism

Scalability is one of the most important characteristics that the Level-2 trigger system should possess. With the available equipment it was possible to arrange up to three ROB - FEX - Global streams, controlled by a unique Supervisor. As in the previous case, each event has only one RoI and this RoI is contained in a single ROB.

In Figure 5 the dependency of the time per event on the message size is shown for one single stream, two and three parallel streams. Only the case with one event allowed in each stream is considered.

In going from one stream to two and three streams one would expect a proportional increase in the aggregate bandwidth and a proportional decrease in the time spent for each event. Although scaling has been observed, it is not perfect, due to the following:

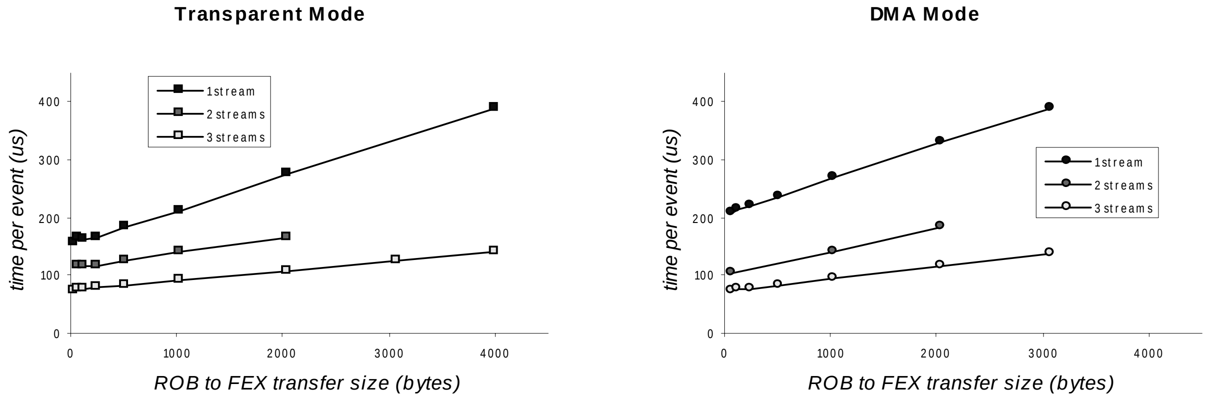

Figure 5. Event Parallelism. The system is constituted of one, two or three independent pipelines that process events in parallel.

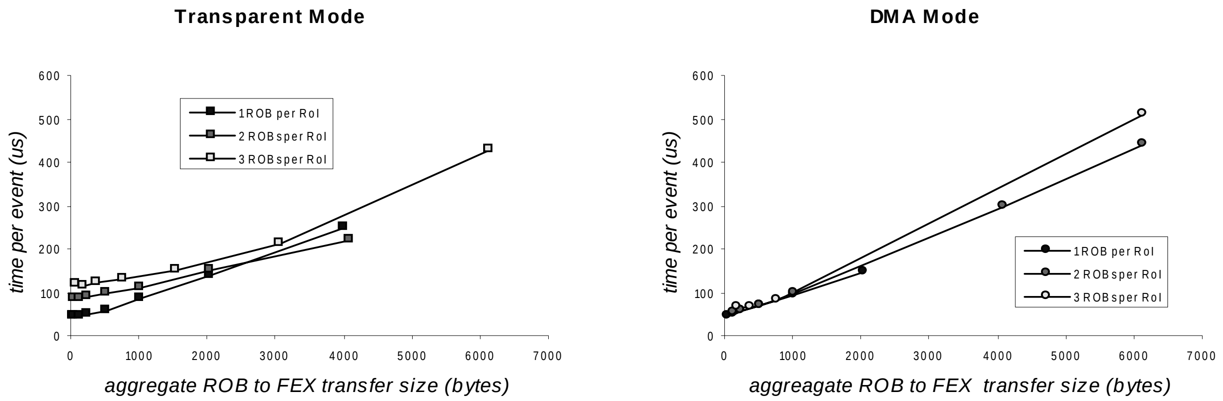

Figure 6. RoI Fragment Building. One, two or three ROBs send data to the same FEX, which has to collect the fragments and build the RoI.

- the Supervisor and the RoI Distributor are shared resources that limit the performance of the rest of the system especially for small message sizes, where the frequency is higher.
- the different speeds of the processors.

RoI Fragment Building

The system was set up with one, two or three ROBs, one FEX and one Global, to evaluate the case where there is only one RoI per event, but the RoI is split amongst many ROBs. Thus a FEX receives data from several ROBs and has to collect the fragments in order to build an RoI. This configuration is to test the efficiency of the fragment builder inside a FEX and the cost or benefit of spreading event data over several sources. Since the FEX has to wait for an RoI fragment coming from each ROB, the performance is affected by the degree of parallelism between the ROB to FEX transfers and between these transfers and the fragment building.

We know that the ROB to FEX transfers are partly serialized: due to the lack of a broadcast option in the VME bus, the RoI Distributor starts successive ROBs with a delay of ~15 µs between them. Also in this case the different speeds of the processors involved have an important effect, because the FEX has to wait for the slowest ROB before completing an RoI.

Figure 6 shows the measured time per event for different aggregate ROB to FEX transfer sizes. For small events the extra overhead leads to a net loss of performance, especially for the transparent mode. For larger data sizes the losses are generally smaller and in very limited circumstances there is a gain in performance.

RoI Parallelism

The system was composed of one, two or three ROB - FEX combinations feeding into one Global. An event contains respectively one, two or three RoIs, each in a single ROB. Since the global has to wait for a feature coming from each FEX before taking its decision on the current event, the performance would be similar to that of a single stream only if there were complete overlap of all the ROB - FEX threads. But, as men-

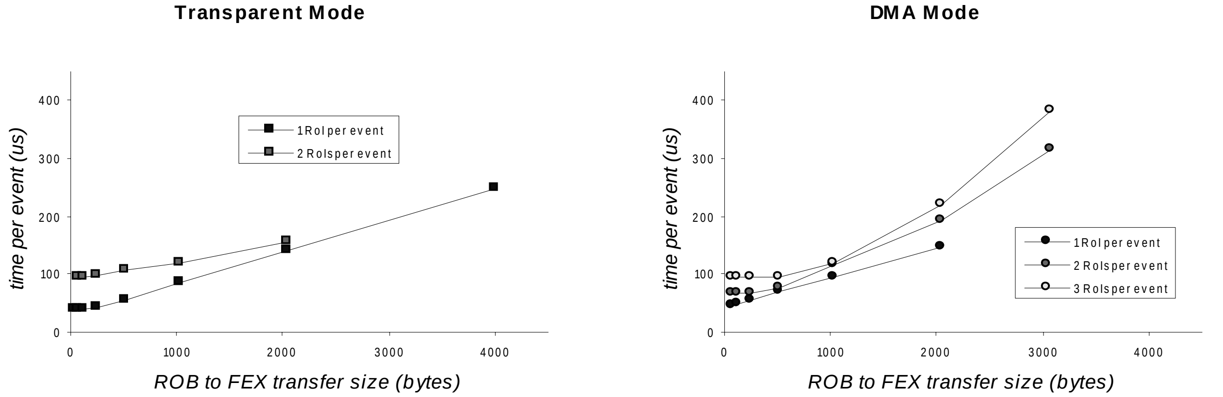

Figure 7. RoI Parallelism. One, two or three RoIs are processed in parallel and the features are collected by the same Global processor.

tioned earlier, the RoI Distributor introduces a delay of ~15 µs between the start of each ROB to FEX transfer. Also, as before, the Global has to wait for the slowest of the ROB-FEX threads, since they proceed at different speeds. Figure 7 shows that there is considerable parallelism in the processing of RoIs, although some of the details of the plots require further study.

Switch

The configuration used to evaluate the event parallelism with three independent streams controlled by the same Supervisor has been used to study the impact of a 4-port switch on the performance of the system. The nodes were arranged in four ringlets each connected to a switch port, as shown in Figure 2.

As shown in Figure 8, the switch leads to a small improvement in the performance of the system, despite the extra delay of ~1.5 µs that it introduces in the packet latency. The improvement could be attributed to the following:

- there were less nodes on each ringlet connected to the switch and this reduced the time to send a packet around the ring from a ROB to a FEX.
- there was less traffic on each ringlet reducing the chance of any delay.

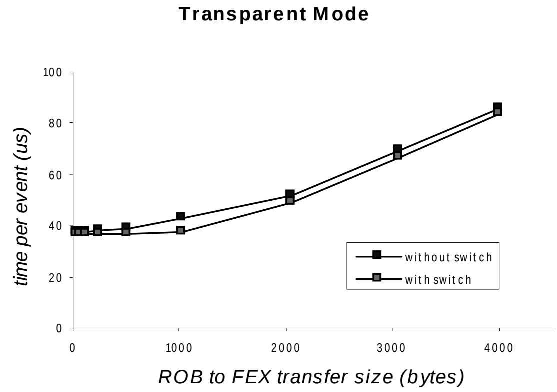

Figure 8. Effect of the introduction of a switch in the SCI network

CONCLUSIONS

The tests have demonstrated high rate operation of the components used in this system. In all cases, except the RoI Distributor, the rates are comparable to those required, albeit with simplified functionality.

Some scalability in terms of pipelining, event parallelism, fragment building and RoI parallelism has been demonstrated for typical ATLAS data sizes.Two bottlenecks have been identified in the system that limited scalability: the PCI bus of the RIO-2s for the bandwidth and the RoI Distributor for the rate. Because of the relatively small size of the system and the discrepancy between SCI and PCI bandwidth it has not been possible to load the network sufficiently to study limitations of the SCI links and switches, although some measurements could indicate that congestion has happened. Consequently, also the introduction of an SCI switch has not shown any significant effect in trigger rate or latency. Investigation in this area will require monitoring of SCI traffic on the rings.

Considerable work remains to be done. Software forms a major part of the system. In order to investigate the effect of combining the above configurations together, larger systems are planned (partially as part of the SISCI project). Special hardware data generators will be needed to study loading the network without using an excessive number of expensive nodes. Comparative studies between different network technologies need to be made. For this an ATLAS programme is starting to design and write technology independent software for the Level-2 trigger architecture with the possibility to link in technology specific software using a standard API. To evaluate the number of CPUs and the required I/O rate into processors, selection algorithms have to be included in the tests. More realistic components such as the interface to the Level-1 trigger and ROBs need further investigation.

REFERENCES

[1] IEEE Computer Society, IEEE Standard for Scalable Coherent Interface (SCI), IEEE Std 1596-1992, August, 1993.

[2] ATLAS Technical Proposal. CERN/LHCC/94-43, 1994.

[3] J. C. Vermeulen et al., “Performance requirements of proposed ATLAS second level trigger architectures from simple models”, presented at CHEP97 by S. George, Berlin, Germany, April 1997.

[4] R. Blair et al., “The ATLAS Level-2 Trigger Supervisor”, presented at the 2nd Workshop on LHC Electronics, Balatonfüred, Hungary, September 1996.

[5] J. R. Hansen, “Local-Global Demonstrator Programme for the ATLAS Second Level Trigger”, presented at the X IEEE Real-Time Conference, Beaune, France, September 1997.

[6] J. J. Labrosse, “MicroC/OS -the Real-Time Kernel”, R&D Publications Inc, Distributed by Prentice-Hall, ISBN 0-13-031352-1, 1992.

[7] SISCI Project - EU Contract ESPRIT 23174.

[8] F.Giacomini et al., “Application Programming Interface for the Scalable Coherent Interface”, October, 1997.

[9] H.C. van der Bij et al., “S-LINK, a Data Link Interface Specification for the LHC Era”, presented at the X IEEE Real-Time Conference, Beaune, France, September 1997.