On the Integration of High Performance ATM-based Event Builders

C. Bizeau, M. Costa, J.-P. Dufey, M. Letheren, A. Pacheco, C. Paillard
CERN, 1211 Geneva 23, Switzerland

D. Calvet, P. Le Dû, I. Mandjavidze
CEA Saclay, 91191 Gif-sur-Yvette CEDEX, France

M. Weymann, A. Wiesel
Creative Electronic Systems, Geneva, Switzerland

Abstract

A first demonstrator has shown encouraging results for the use of Asynchronous Transfer Mode (ATM) switching networks in the implementation of high performance parallel event builders. Our present goal is to show that the integration of event builders, including the implementation of the source and destination functions and the bandwidth adaptation to the switching network, can be realized with commercially available products and that good performance can be achieved. We shall review some critical issues, present results from performance measurements and analyse the overheads.

I. INTRODUCTION

An event builder implementation provides high performance, in our definition, if a) it ensures that a significant fraction of the available network bandwidth is effectively used for data transport and b) that this is achieved for the smallest possible event fragments.

The overheads due to the protocol layers required to access the network link can be large and it is easier to reach high performance when transferring large blocks of data. This fact leads event builder designers to, for instance, group fragments from many events. Another consequence is that usually the control of the event builder is implemented on a separate network. In contrast, the performance of ATM networks is very high for small data packets, as well as for larger ones. On an STM-1 link (155 Mbit/s), single cells carrying 48 bytes of information can be delivered every 2.7 µsec, corresponding to a maximum frequency of 370 KHz. This includes the operations of the ATM layer protocol (header handling, traffic shaping) and of the AAL5 protocol (segmentation and reassembly) which are implemented in hardware.

This capability of ATM allows to envisage different architectures for event builders: the events can be built individually, independently of the event fragment size, it is possible to use the switching network to transport the event builder control messages and more ambitious systems can be conceived where event building is part of a phased event selection process [1].

In order to benefit from the high performance of ATM networks, in particular for small messages, it is necessary to optimize and match the performance of the various components of the event builder: the switching network itself, the network adapters, the accretion of data packets from the front end data buffers in the sources and the transfer of assembled events to the analysis processors. We have investigated extensively the performance of ATM switching networks together with the network adapters. We are currently evaluating solutions, based on commercially available components, for efficient data transfers between front end modules within the sources and for data delivery to the analysis processors.

We restrict this presentation to the discussion on how to achieve high data flow performance. Considerations on efficient supervising functions, such as the destination assignment or the distribution of information from the Level 1 trigger, are not tackled. For ATM we consider only STM-1 links with a bandwidth of 155 Mbit/s at the physical layer (SONET/SDH). The effective bandwidth is 150 Mbit/s at the ATM level and 135 Mbit/s or 16.8 Mbyte/s at the user data level.

II. GENERIC EVENT BUILDER MODEL. PERFORMANCE REQUIREMENTS

The data flow structure of a generic event builder is presented in Figure 1. The implementation of a complete event

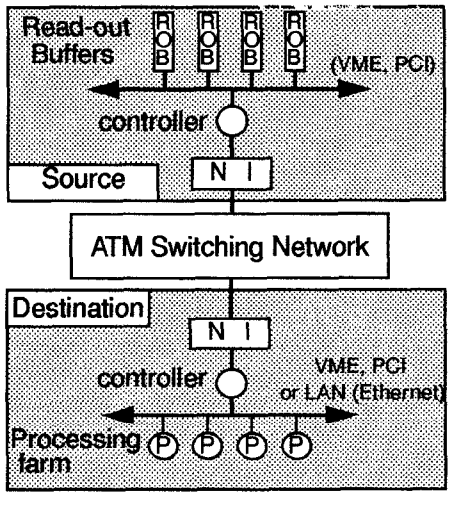

Figure 1: Data flow structure of a generic event builder.

0-7803-3534-1/97 10.00©1997IEEE

building system, in its simplest form, i.e. the "push" architecture, requires that data are collected from one or several Read-Out Buffers (ROB) in the sources before being sent as an event fragment through the network via the Network Interface (NI). The destination, in turn, submits the full event to a processor in a farm for analysis. The main data flow is from the sources towards the destinations. Only control messages are sent in reverse direction, in particular when point to point flow control and reception acknowledgements are required (transport protocol).

In the "pull" architecture, the destination initiates the transfer of data by the sources, one of the processors in the farm being in charge of the selection and/or analysis of an event. The control messages issued by this processor to request data from the sources are routed by the switching network.

A. *Characteristics and performance requirements in the case of push architecture*

The maximum size of an event fragment F [KByte] depends on the available link throughput of user data T [Mbyte/s] and the trigger rate f [KHz]:

$$F = k * T / f \qquad (1)$$

k is the load factor of the link ($0 \leq k \leq 1$) which takes into account the fact that the performance of the network adapter may be limited for packets of size F, or that a limitation may be imposed to avoid congestion in the switching network. It is not safe to use a value of 1, even with very efficient network adapters and perfect traffic in the switch. On the other hand, it is desirable to use as much as possible of the link bandwidth.

The full size (NXN) of the event builder, assuming equal fragments on all ports, is then determined by the full event size E [Mbyte]:

$$N = E / F \qquad (2)$$

The chart in Figure 2, established for T=16.8 Mbyte/s (maximum user data throughput on ATM links at 155 Mbit/s) provides an estimate of N, knowing the event size E and the trigger frequency f. The maximum event fragment size (for k = 1) is indicated in parentheses with each frequency value. As an example (dotted line in the chart) events of 1 MByte at a rate of 1 KHz and with k = 0.6 require an event builder of 100 X 100. Thus an event fragment size of 10 KByte is needed in order to efficiently load the switch.

In the sources, event fragments have to be built by accretion of sub-fragments, the size of which depends on the distribution of data in the read-out buffers. Although this is completely dependant on the particular DAQ system, one can nevertheless formulate a general requirement for the bus that links n ROBs: its throughput, measured for the transfer of sub-fragments of size of 1/n of the event fragment size, should be at least equal to the network link throughput. This is quite a stringent condition when the sub-fragments are just of a few hundred bytes. It should be noted that this problem of accretion is not specific to ATM and that in all cases the bus linking the front-end buffers should have better performance than the switching network link itself.

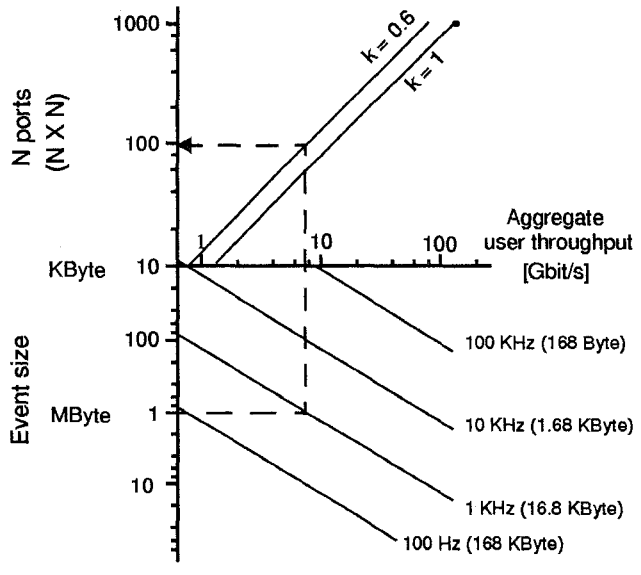

Figure 2: Event builder size as a function of event size and trigger frequency.

Full events are assembled in a destination at a rate f/N. We assume that each destination has a farm of N_P processors. N_P depends on the analysis time per event, t [msec]: $N_P \geq t * f/N$. The rate at which events are submitted to a processor is $\sim f / (N * N_P)$. In the example, assuming 1 sec/event and 100% utilization of the CPU, $N_P = 10$. The sub-network supporting the processor farm must have approximately the same bandwidth as the network link.

B. *Characteristics of an event builder with pull architecture and phased selection*

In the case of pull architecture and phased event selection, each processor receives an event to manage at a frequency $f / (N * N_P)$ and may, for instance, request of the order of 1 or 2 data fragments per event (e.g. ~ 1-2 kByte or less) for the first phase of selection. The network interface of the destination collects and distributes the event data needed for the first phase at a frequency ~f / N. As an example, in the ATLAS architecture C, the design frequency is 100 KHz, N = 256. The frequency at a destination is 400 Hz and, with a 10 processors farm, events are assigned to a single processor at a rate of 40 Hz. The processor will request full event building for the events that have passed successfully all the earlier phases, but this occurs at a frequency much lower than $f / (N * N_P)$.

C. *Critical points regarding performance*

The critical points where data flow performance requirements may be difficult to achieve are:

- the *switching network* that routes the main data streams and possibly interleaves control messages. Depending on the traffic, it may be necessary to reduce the load to avoid congestion.

- the *sources and destinations* where the implementation of concurrent data flows can be challenging.
- the *NIs (Network Interfaces)* where software overheads can reduce the efficiency of utilization of the link bandwidth.

III. PERFORMANCE OF LINK ADAPTORS

The ATM standard includes an adaptation layer with several protocol options designed for the traffic characteristic of different applications. For transfer of data, the protocol, called AAL5 is defined for blocks with variable size, up to 64 KByte. On the transmission side the AAL5 protocol specifies that a trailer with a CRC is added and that the data block is segmented in fixed size ATM cells. On the receiving side reassembly of the original data block and CRC check are performed. The ATM layer is in charge of the cells and routes them according to their virtual connection identifier. Many virtual connections can be active simultaneously in a single NI. At reception, cells are sorted out according to the virtual connections so that reassembly at AAL5 level can occur. ATM does not provide a transport protocol. Corrupted packets are detected, but retransmission is not performed as it is not required in every application. If needed, it has to be provided in the upper layer.

Commercial chip-sets provide hardware implementations of the ATM and AAL5 protocol layers. The complex operations of routing, segmentation and reassembly are performed in a very efficient way with negligible overheads. In addition the chip sets supports standard rate policing services, one of which, Constant Bit Rate (CBR), allows the implementation of an efficient traffic shaping scheme by means of rate division.

Larger overheads originate in the higher layers, on top of the ATM and AAL5 layers: the optional *transport protocol* layer and the *event building protocol* which provides event fragment identification, their assembly into events (event fragments arrive in unpredictable order) and determines when an event is completed. A short description of our implementation of the event protocol layer and the data structures can be found in [2].

In a first implementation of a Network Adaptor we have developed an ATM interface and studied the best performance that could be achieved [2]. The event building software runs without operating system on a MIPS R3000 at 25 MHz. When sending or receiving AAL5 packets, in the absence of the event building protocol layers, the software overhead per packet is 16 µsec and is independent of the packet size. When the event building layer is added, the overhead, measured on the receiving side, where it is highest, is 25 µsec. Small packets can be received at a frequency of 40 KHz. For larger packets, the frequency is determined by the transfer time.

Figure 3 shows the performance measured on a commercial ATM interface from CES [3]. Based on the chip NICStar from IDT [4], it is a PCI mezzanine card (PMC) on a RIO2 board which implements a PowerPC 604 at 100 MHz [5]. We have

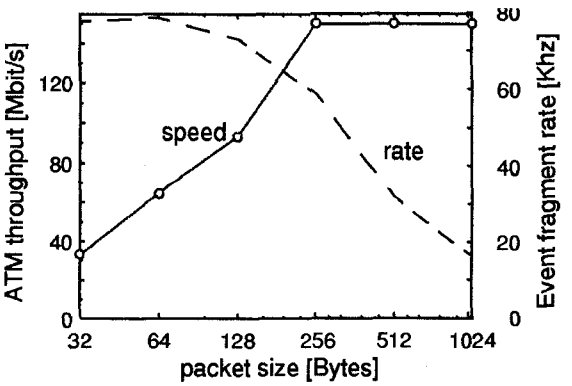

Figure 3: Transmission throughput and rate for the CES-ATM8468 adaptor with "zero-copy" driver under Lynx-OS.

developed a "zero copy" driver under LynxOS with the aim of reaching the best possible performance at AAL5 level. In order to minimize the overheads due to the operating system, the driver checks asynchronously for the completion of a packet transfer or the arrival of a new packet instead of using interrupts. The overhead per packet is 10 µsec (not including the event building protocol).

We found that a transport protocol was not needed in an event builder, at least on small systems. However one cannot exclude that it may be needed in specific applications. We have developed and tested a simple transport protocol and measured its efficiency. It implements window based flow control and sends acknowledgments for all packets received. On the first system tested (based on MIPS R3000), we measured an increase of the overhead of some 50 µsec, part of which is due to the transmission of the acknowledgement message.

The development of optimized drivers is feasible in a few man-months. Presently it is unavoidable if high performance is required. Considering that the inefficiencies of the commercial drivers are well recognized and that efforts to improve their performance have been undertaken (see for instance [6]), it is reasonable to assume that faster commercial drivers will be available in the future.

IV. PERFORMANCE OF ATM SWITCHING NETWORKS FOR EVENT BUILDING TRAFFIC

Latest results from our event builder demonstrator setup have been presented in [7]. We summarize the main points.

We have shown that event building traffic, in push architecture, can use a large fraction of the available aggregate throughput without any data loss on ATM switches with 8 ports. As traffic shaping we use the rate division scheme provided by the CBR implementation in the SAR chip. Figure 4 shows the performance measured on a demonstrator using an 8 ports switch from Bell Labs. Eight traffic generators send event fragments of variable size (gaussian distribution) to 8 destinations. The aggregate throughput is 120 MByte/s, i.e. 90% of the available bandwidth (135 MByte/s). Saturation occurs for

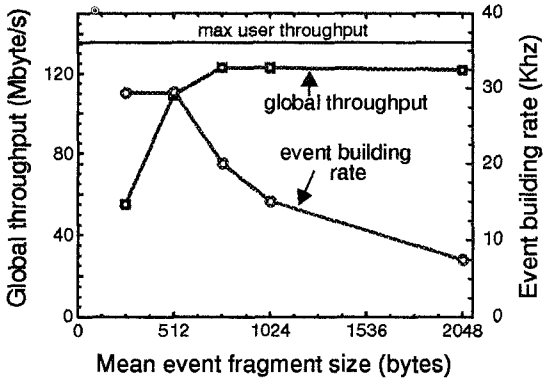

Figure 4: 8 X 8 event building; push data flow; gaussian size distribution (σ^2=30%).

packets larger than 700 bytes. An event building frequency of 30 KHz can be achieved for event fragments up to 512 bytes.

Simulation studies show that the rate division traffic shaping might be sufficient to reach high loads on larger switches based on various technologies. As an illustration, Figure 5 shows the

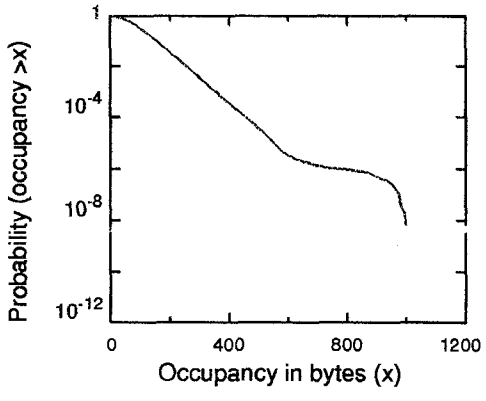

Figure 5: Probability for the occupancy of the shared buffer memory in a switching element of a 256 X 256 Alcatel type switch, 70% load.

probability curve (tail distribution) for the buffer occupancy in an Alcatel type switch [8] with 256 ports. The switching elements are 16X16 and have a shared buffer memory of 256 bytes. Consequently the probability to overflow a buffer is very small for a load of 70%. In the latest implementations even larger buffer sizes are provided. These good results are obtained under the assumption that the NIs in the sources are not synchronized, which is expected, each NI being an independent module with its own internal clock.

Figure 6 shows the performance of a demonstrator operating in pull mode. In this configuration full duplex links are used. A destination requests event fragments by sending short messages to the sources. The results are shown for 6 sources and 1 destination. The values are a lower limit of the possible performance because the traffic generators used as sources cannot handle more than 1 request at a time. Nevertheless the results show a good performance for this architecture.

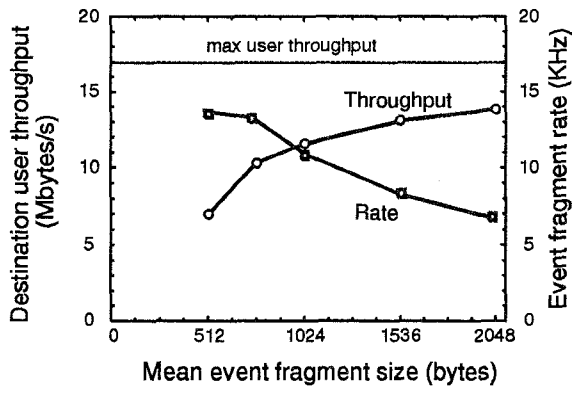

Figure 6: Performance of a "pull" event builder (6 sources and 1 destination).

The results from the demonstrators and from simulation studies are encouraging. The rate division traffic shaping minimizes the congestion probability, provided that the sources have random time correlation.

V. Combined Traffic in Sources and Destinations

So far we have discussed the performance of the event builder assuming that the event fragments were already available in the source memory and discarding events in the destination as soon as they were assembled. We next consider solutions for the complete data transfer from the read-out buffers to the source network link on one hand and from destination link to the analysis processors on the other hand.

A. Source modules

We have seen that, in general, event fragments have to be built by accretion of smaller blocks of data located in several read-out buffers connected to a single source module by a local link. In order to achieve the best performance, the bandwidth of this link has to be at least of the same order as the network link bandwidth, and must provide good performance for data blocks of the size delivered by the ROBs. At present we can envisage VME and PCI as standard local links. PCI seems to offer the best performance characteristics, however its limited physical range restricts its use to a small number of ROBs (typically up to 4). VME offers a bandwidth large enough to match with ATM STM-1 links. Its limitation is due to large overheads for the initialisation of block transfers.

We have measured the performance of a source module using VME to link the ROBs. As source module we have used a CES RTPC board [9] with the ATM 8468 interface from CES [3]. The ROBs were emulated by a single slave board (in our case a FIC 8234 from CES) from which a variable number of data blocks were transferred into the source module memory. The RTPC uses a 100 MHz PowerPC 604 with a second level cache of 512 KByte. The VME interface is connected to the PCI bus of the RTPC. Block transfer between the slave mem-

ory and the source memory is performed by means of the Block Mover Accelerator hardware controller (BMA) which also provides for chained block transfer driven by a list of descriptors stored in the RTPC memory.

We used the VME and ATM drivers provided by CES under LynxOS. They offer asynchronous access to VME and ATM (at the AAL5 level) thus allowing concurrent transfers on both links. The test program consisted of 2 threads, one for VME read-out and one for the transfer of event fragments on the ATM link, each one passing control to the other once it has initiated a transfer.

The results are shown in Figure 7. The chained block transfer in VME is very efficient and no significant throughput variation is observed if the event fragment is composed of 1 single block or several blocks (up to 8). The measured overhead when starting a chained block transfer is about 120 µsec plus 5 µsec for each subsequent block. It is possible to implement a simpler version of the VME driver if higher performance for small fragments is required.

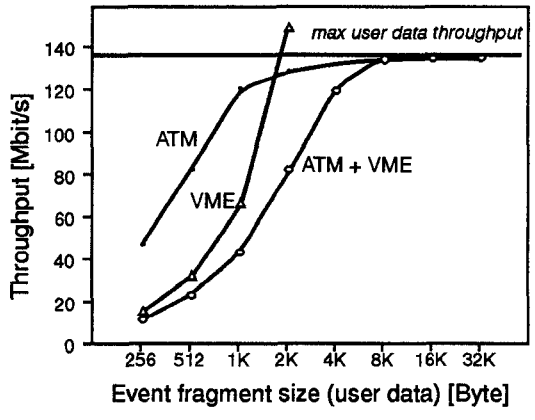

Figure 7: Throughputs for ATM, VME and simultaneous transfer in the CES RTPC board.

The PCI option is being evaluated in a separate project that implements and tests a PMC-based solution for the ATLAS Read-out Buffers [10]

B. *Destination modules*

The problem of data transfer to the analysis processors is in principle not difficult: it is relatively easy to achieve high performance for the large blocks formed by complete events. However, an additional overhead is imposed by the fact that the event fragments, of variable length, arrive in unpredictable order and a copy operation is necessary in order to store the event in a contiguous buffer.

One or two Fast Ethernet ports could be sufficient to carry the traffic of a destination to the processors. In the example of a 1 MByte event with 1 sec analysis time, the transmission time, is of the order of 0.1 sec. This is compatible with the use of TCP/IP which is difficult to avoid for Ethernet and commercial UNIX workstations. We have measured a bandwidth occupation as high as 80% on a 10-Base isolated Ethernet link with TCP/IP packets of 4 KBytes, under Lynx-OS.

VI. Conclusion

An important result is that the performance challenges are not so much in the switching network as in the network interface and in the connection with the rest of the system. It is also clear to us that commercial products deliver good hardware performance but that efforts are required to improve the software which is not designed for small messages on high throughput links. We believe that event builders with good performance can be implemented, based on currently available commercial components.

VII. References

[1] D. Calvet et al., "A Study of Performance Issues of the ATLAS Event Selection System based on an ATM Switching Network", in *IEEE Transactions on Nuclear Science*, vol. 43, No 1, February 1996, pp. 90-98.

[2] M. Costa et al., "Results from an ATM-based Event Builder Demonstrator", in *IEEE Transactions on Nuclear Science*, vol. 43, Num. 4, August 1996.

[3] Creative Electronic Systems SA Geneva, ATM 8468, PCI-ATM Mezzanine Card, DOC 8468/PG, Version 1.0, May 1996.

[4] IDT Inc., Santa Clara, CA, USA, IDT77201 NICStAR chip, User Manual Vers. 2.0, November 30, 1995.

[5] Creative Electronic Systems SA Geneva, RIO2 8060, PowerPC based RISC I/O Board, Technical Manual vers. 1.0, DOC 8060/UM, October 1995.

[6] T. v. Eicken et al., "U-Net: A User-Level Network Interface for Parallel and Distributed Computing", in *Proc. of the 15th ACM Symposium on Operating Principles*, Copper Mountains, Colorado, December 3-6, 1995.

[7] M. Costa et al., "Lessons from ATM-based event builder demonstrators and challenges for LHC-scale systems", in *Proceedings of the 2nd Workshop on Electronics for LHC Experiments*, Balatonfured, Hungary, 23-27 September, 1996.

[8] M. Henrion et al., "Technology, Distributed Control and Performance of a Multipath Self-Routing Switch", in *Proceedings of the XIVth International Switching Symposium*, Yokohama, Japan, October 1992, vol. 2, pp. 2-6.

[9] Creative Electronic Systems SA Geneva, RIO2 8067LK, PowerPC Single Board Computer, Technical Manual vers. 2.0, DOC 8067LK/UM, May 1996.

[10] O. Gachelin et al., "ROBIN: A Functional Demonstrator of the ATLAS Trigger/DAQ Read-Out Buffer", in *Proceedings of the 2nd Workshop on Electronics for LHC Experiments*, Balatonfured, Hungary, 23-27 September, 1996.