Discrete Event Simulation of the ATLAS Second Level Trigger

J.C. Vermeulen[1], S.Hunt[2], C. Hortnagl[3], F. Harris[2],
A. Erasov[4], R.J. Dankers[1], A. Bogaerts[5]

[1]NIKHEF, Amsterdam, Netherlands, [2]Oxford University, U.K, [3]University of Innsbruck, Austria
[4]MSU, Moscow, Russia, [5]CERN, Geneva, Switzerland

Abstract

Discrete event simulation is applied for determining the computing and networking resources needed for the ATLAS second level trigger. This paper discusses the techniques used and some of the results obtained so far for well defined laboratory configurations and for the full system.

I. INTRODUCTION

For the ATLAS experiment, one of the two general-purpose experiments at the LHC, a trigger system with three levels will be built. This system should achieve a reduction in event rate from the initial 10^9 Hz at maximum to about 100 Hz. The first level will be a pipelined systolic system that analyzes the data from the calorimeter and muon subdetectors. During the decision time of about 2 μs the raw data is buffered on or close to the detector in the pit. After a level one accept (maximum rate 100 kHz) the data will be transmitted via 1 Gbit/s optical fibres to about 2000 Read Out Buffers (ROBs) located at the surface. The first level trigger identifies also Regions of Interest (RoIs) and sends position and type information to the second level trigger. For each event the ROBs that need to provide the input data for the second level trigger are selected on the basis of RoI information, which also may be generated by the second level trigger itself. After a second level trigger accept (maximum rate about 1 kHz) the data from all the ROBs is sent via the event builder to one of the processors of the event filter. Here the third level trigger produces a final decision on acceptance of the event.

The second level trigger can logically be divided in a number of steps : "feature extraction" for single subdetectors (e.g. finding of track segments and determination of the track parameters), combination of features found for different subdetectors within the same RoI, combination of the information from different RoIs and the generation of decisions. For the processing required it is estimated [1] that up to 1000 500 MIPS processors are needed, assuming that there are adequate communication facilities between the ROBs and these processors. It is possible that part of the processing will be done with special-purpose processors implemented with FPGAs.

Simulation is necessary to acquire a good understanding of the factors controlling the behavior of the system. This is due to the large number of processors in combination with the networks and switches providing the communication facilities required and the use of RoIs for control of the dataflows.

In order for the results of simulation to make sense many input parameters need to be set to realistic values. The operation of the system also depends on the type of events selected by the first level trigger and the number and types of RoIs associated with them. Two approaches are possible here. In the first the relevant information is extracted from simulated events and used as input for the simulation on an event-by-event basis. Alternatively an estimate can be made of the number of each type of events selected by the first level trigger, of the number of RoIs associated with each type and of the distribution of and correlation between the RoI positions. With this information the relevant properties of the events can be generated during running of the simulation program. This approach in principle is less accurate than the first, but provides a good first order estimate without requiring access to large samples of Monte-Carlo events. The estimates can also be used, together with the other input parameters, for computing the minimum requirements with respect to bandwidth and processing power. This type of calculations is also referred to as a "paper modelling", as the calculations are simple and could be done by hand on a piece of paper. In practice using a spreadsheet for paper modelling makes sense. Results of paper modelling of the ATLAS second level trigger system have been presented at the CHEP97 conference [1]. An overview of all the relevant parameters can be found in [2].

"Computer modelling", i.e. system simulation, takes into account contention for resources and resulting queuing. The latency (i.e. the time required to produce an accept or reject decision) distribution for the system modelled can be determined, as well as distributions of the filling degree of queues and of the utilization of resources such as communication link bandwidth and processor capacity. The computer model can be checked against the paper model, as the average resource utilization should be the same as the utilization computed from the paper model, provided that the same parameters and models are used. A second condition is that queues in the simulated system on average have sufficient storage capacity and that the average

utilization of the available resources (link bandwidth, processing power) is less than 100 %.

II. SIMULATION TECHNIQUE

Simulation can be done in two ways : clock driven or event driven. In the first case the state of the system modelled is updated after each tick of a (simulated) clock. The time intervals between consecutive ticks are short with respect to the response times inside the system. In the second case events (not to be confused with events due to particle interactions, observed in an experiment and in this document referred to as "physics events") occur at certain simulation times. For each event the response of the simulated system is determined. The response can consist of the generation of new events at the same simulation time as the original event occurred or at future simulation times. This type of simulation is called discrete event simulation. For large systems that need only a limited amount of system state updating after an event this method has the advantage that the simulation executes faster and also that the results will be more accurate, as the time of occurrence of the events is not determined by the ticks of a clock. Hence discrete event simulation is the method of choice for simulation of the behavior of the ATLAS second level trigger system with respect to queuing, data throughput and latency (i.e. not with respect to its physics event selection abilities).

The system to be simulated consists of interacting objects. Therefore, object-oriented programming provides a natural way to construct software models. This leads to the choice of an object-oriented programming language. For discrete event simulation one can choose between a language with explicit support for this type of simulation and without support for it. Both types of language are used. MODSIM-II [3] represents the first category, C++ the second. MODSIM-II is based on an object-oriented version of the Modula-II language. Objects can be made to wait until a certain point in time or until certain actions of other objects occur. The events that drive the simulation are hidden from the programmer. In contrast, in C++ a mechanism needs to be implemented for generating events, storing the events in the correct order in an "event list", retrieval from the list and sending them to the objects that need to handle them. The objects themselves need to maintain state machines. Events arriving and also invocation of member functions from others object may induce then state changes.

A program written in MODSIM consists of a number of interacting objects, which are concurrently active. Hence, the program in essence is a parallel program (but executing on a single processor). A discrete event simulation program in C++ [4] in contrast is a sequential program seen from the programmers point of view. This makes explicit coding of the state machines necessary, which may be viewed as a disadvantage. However, the source code documents explicitly the possible states and the transitions between the states. The requirement for explicit coding of the discrete event simulation mechanism may also be seen as a disadvantage when using C++, but run-time efficiency is a critical issue for simulating large systems. The discrete event simulation mechanism plays a crucial role in this respect. It has been found that the possibility to adapt and tune the mechanism allows for appreciable improvements in execution speed.

MODSIM (the current version is MODSIM-III) is marketed by CACI Products Company [3], a license needs to be purchased for each platform (Windows95, Windows NT or UNIX machine) to be used for development. When using C++ the standard development tools can be used.

III. SIMDAQ

For simulation of the ATLAS second level trigger the SIMDAQ program is developed. Its first implementation has been in MODSIM-II. This version [5], apart from the code for the simulation of the SCI and ATM technologies, has been translated and extended in C++. Further development of the MODSIM and C++ programs has been going on in parallel, with a transition to the exclusive use of C++ foreseen. In MODSIM recently work has been done on the simulation of DS-link technology, as reported below. In C++ the emphasis until now has been on the organization of the simulation program (this work has also been used for reorganizing the original MODSIM program), on simulating "generic" models and on the correlation with paper models. Implementation of models of the various network technologies of interest has been partially done in C++.

The C++ program makes use of SUIT (Simple User Interface Toolkit) [6] for platform independent graphical user interfacing. UNIX, Windows95 / WINDOWS NT and MacOS versions are available. The graphical user interface can be used for inspection of the histograms and of summary information during running of the program. It can also be disabled to allow for batch processing. The contents of the histograms and other results, together with a copy of the contents of the configuration file (see below), can be written to an ASCII file. A file is produced by clicking a button when the graphical user interface is enabled and / or when execution of the program finishes and / or each time that a certain number (specified in the configuration file) of physics events has passed through the system model. The histograms in this file can be displayed with a program with a user interface similar to that of the simulation program. Due to the use of SUIT program development and simulation is possible on any of the platforms mentioned above.

The model to be simulated is specified, at the level of processors and switches and their connections, for both the MODSIM and the C++ program with a configuration file. The details at lower level can be controlled by parameters, that can be specified in the configuration file. The same file format is used for both programs.

IV. RECENT RESULTS

A. DS link technology

DS links are bi-directional serial data links which operate at 100 Mbit/s. The T9000 transputer from Inmos [7] has four of these links, while the C104 packet switch from SGS Thomson [8] has 32. Systems built from these components are studied in the context of the demonstrator program for the ATLAS second level trigger. They are also used in the GPMIMD project, for which extensive measurement results are available [9,10].

Models of the components and configuration of the GPMIMD system have been produced. These link T9000 and C104 models in a configuration allowing the comparison of laboratory and modelling measurements. The T9000 model is based on the utilization of 3 key resources, a link resource which represents usage of the physical link, virtual channel resources which represent usage of each of the virtual channels available and virtual channel message resources which are used to model the restriction on virtual channels to handling a single event at once. Messages are passed to the send function of the model, split into packets of at most 32 bytes, assigned a virtual link and finally sent out over the physical links. An acknowledge is produced by the receiver and routed back to the sender to free the virtual channel resource. The model of the C104 includes wormhole routing, interval labeling and group adaptive routing. The more complex properties of the C104 chip have been left out as it was decided that a simple model which encompasses the most influential characteristics of the C104 chip was desirable. The model of the GPMIMD machine connects 18 T9000s via a switching network incorporating 8 C104 chips to another set of 18 T9000s. Good agreement with the results of the measurements was found.

B. Parallel push architecture

A detailed study has been conducted on a second level trigger system with a "parallel push architecture". In this type of system a supervisor (the second level trigger supervisor) receives RoI information from the first level trigger and distributes this to the ROBs, together with information on the processors to be used for feature extraction and global processing. For each subdetector there are separate farms of feature extraction processors, while there is also a farm with global processors (see figure 1). For each RoI and for each subdetector involved in processing data inside that RoI an individual processor is used for feature extraction.

Input and output on the processors and on the ROBs are assumed to be performed by DMA controllers that interrupt the main process when input or output is finished. However, for the ROBs the input of raw data (not indicated in figure 1) is assumed to proceed without requiring attention of the processor. The processes taking care of feature extraction and global processing, as well as the processes in the ROBs taking care of the extraction of the data to be sent to the feature extraction processors are modelled as low-priority processes. These are polling flags in memory that indicate the availability of data to be processed. In order to avoid overload of the ROBs decisions are sent by the supervisor in blocks of 100, so that only one interrupt per 100 decisions in a ROB is generated. All relevant parameters and models for the processes in the system, as well as the number of ROBs, are documented in [2]. For the switches and network technology no models are provided in this document. For the "generic model" the switches are crossbar switches with unlimited buffering on input and output links and with arbitration for access to the output buffers. This arbitration can be switched off for studying the effect of it. Aggregate switches can be built from these switches. The links transport data with a fixed speed of 10 Mbyte / s.

"Paper modelling" results for the system of figure 1 are available for two different trigger menus for high luminosity running. A trigger menu is a list of possible trigger items. Each trigger item defines the number and type(s) of RoIs. For each item an estimate of the rate is available. On the basis of this information events are generated in the simulation program. For this comparison and also for other results presented in this paper the execution times of the various processes have been set to fixed values. This has also been the case for the sizes of the data fragments sent by the ROBs. The time intervals between successive first level trigger accepts had all the same duration. The type of trigger item and the positions of the RoIs however have been selected at random and for the trigger items with probabilities given by their estimated frequencies. The results of the simulation program for processor and communication link utilization have been found to be in excellent agreement with the "paper model" results. However, the computer model needed to be run with a first level trigger rate that guarantees a stable latency distribution, i.e. the resource utilization is everywhere well below 100 %. The results obtained were scaled with a factor given by the ratio of the nominal first level trigger rate and the actual rate.

For the "extended trigger menu without missing energy trigger items" a surprising result was found for the latency distribution, using the nominal first level trigger accept rate of 40.0 kHz. With this trigger menu the utilization of the processing resources of the hadron calorimeter ROBs would be more than 100 % when the parameter values of [2] are used. Therefore the task switching time was reduced from 50 to 35 μs in order to obtain utilizations below 100 %. All other parameter values were set as specified in [2], resulting in processor utilizations ranging from 28 to 42 %. For a bandwidth of 10 Mbyte / s the link utilizations are all below 30 % except for the hadron calorimeter ROB output link with an average utilization of 59 %. All processing times were fixed, as well as the length of the interval between consecutive first level trigger accepts. The switches were modelled as single crossbar switches of the type described earlier in this section. The internal bandwidth was set to 50 MByte / s per connection.

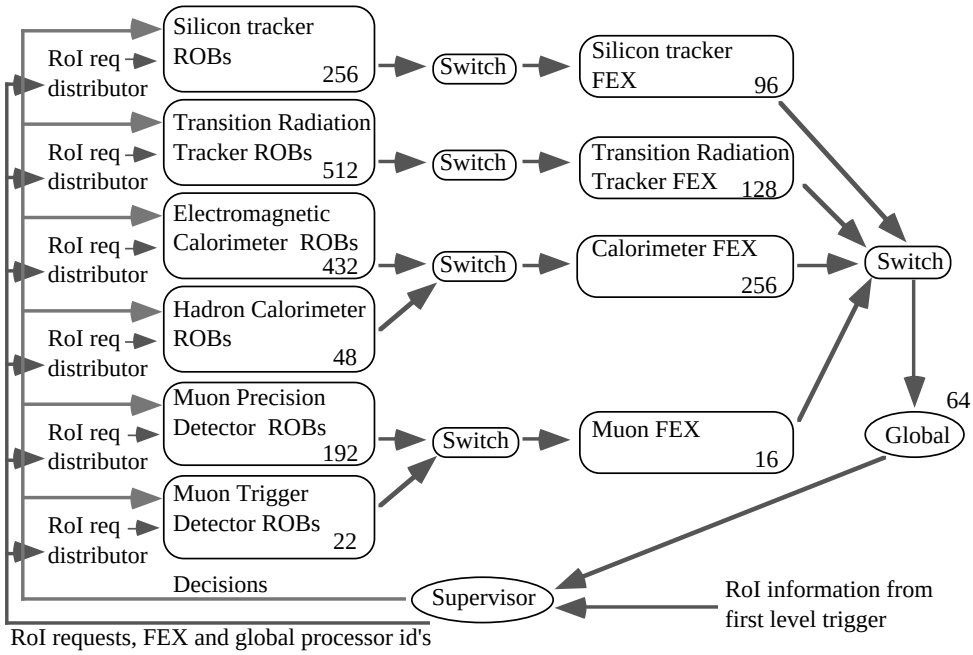

Figure 1: The parallel push model. The numbers indicate the number of Robs or number of processors, "FEX" stands for "feature extraction". The switches are either single switches or aggregate switches built from two layers of switches . The 100 Mbyte / s data links (one per ROB) transporting the raw data to the ROBs are not shown.

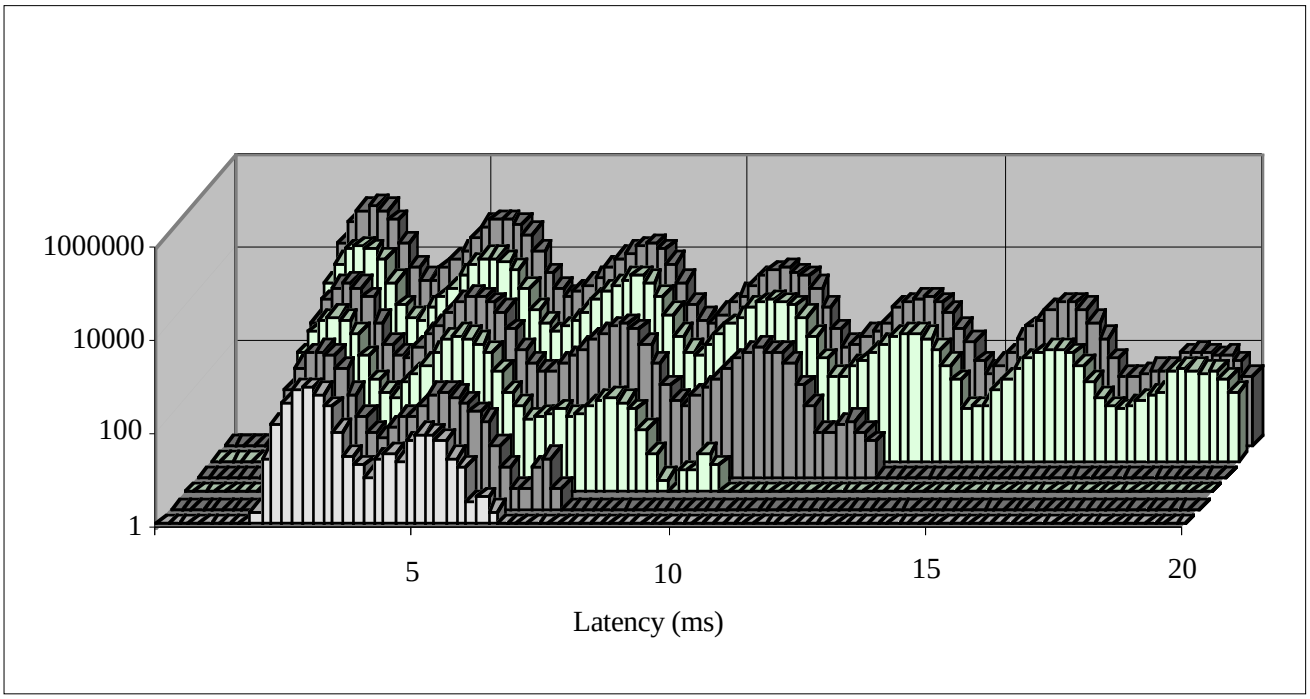

Figure 2: Latency distributions, as obtained after about 0.1, 0.3, 1.0, 3.0, 10.0 and 30.0 s of simulated time for the system of figure 1. The numbers along the vertical axis indicate the number of "physics events". The relative large increase of the tail at longer simulated times shows that the system is nearly or just instable (i.e. longer and longer latencies may occur, until buffer overflow reduces the number of events to be processed).

The latency distribution shows a number of peaks with equal distances between the peaks. This behavior was first seen with the C++ program but has been confirmed by results from the MODSIM program. Figure 2 shows latency distributions taken at different times. For this simulation 1 hr of running time corresponded to about 7.5 s of simulated time on a 200 Mhz Pentium Pro machine with Windows NT as operating system. The MODSIM program needs about 1 hr per 0.12 s on a DEC ALPHA 2100 type 5/250 workstation with a 250 Mhz 21164 processor.

It was found that the peaks in figure 2 are due to the round robin allocation of the feature extraction processors. The distance between the peaks is equal to the time of one round robin cycle for the processors handling data from the calorimeter (2.7 ms for 256 processors, as the total RoI rate for the calorimeter is 94 kHz) and changes as expected when their number is changed. A smooth distribution is obtained when the processors with the smallest number of events queued, as determined by the supervisor from previous assignments and from the event id's contained in the decisions received from the global processors, are allocated. However, this does not lead to a decrease of the width of the distribution : the system modelled even becomes instable, i.e. the maximum latency grows without bounds.

The width of the latency distribution can be explained by queuing in the switch connecting the calorimeter ROBs to the feature extraction processors. This is evident from the distribution of the time needed for transport of all data fragments of a single RoI across the switch : the distribution has approximately the same shape and width as the latency distribution. In the switch data fragments from earlier events may become queued after data of later events. The queuing in the switch occurs predominantly for fragments sent by the hadron calorimeter ROBs : the distribution of the time needed for transfer of a single fragment across the switch is relatively narrow for the electromagnetic calorimeter ROBs, while for the hadron calorimeter ROBs it has again approximately the same shape and width as the latency distribution. This is due to the high RoI request rate for these ROBs (on average 5.9 kHz, for the electromagnetic calorimeter ROBs this rate is 2.5 kHz, for all other ROBs it is not higher than 1 kHz) in combination with the arbitration for access to an output port inside the switch. These two factors lead to queuing of the event fragments in the input ports of the switch. The time interval between the arrival of a RoI request at a hadron calorimeter ROB and the availability of the event fragment at the output of the FIFO queue in the switch input port receiving data from that ROB can be longer than the average time interval between two successive assignments of the same feature extraction processor (i.e. for round robin assignment the length of one round robin cycle). This leads to additional contention in the switch. In the case of round robin assignment the amount of contention changes periodically, which most likely causes the peaks in the latency distribution.

V. OUTLOOK

Much further work needs to be done. Different trigger strategies, architectures and network technologies have to be studied. Further validation of modelling results by comparison with measurement results of test setups need to be undertaken. The current programs, although further development is required, provide a solid basis for these studies. It has been proven that by exploiting current software technology (and also with commodity hardware) it is feasible to simulate the full second level trigger system of ATLAS with respect to queuing and utilization of available resources. Functional simulation with respect to the aspects of error handling can be foreseen for the future. This will allow to study the impact of different error handling strategies.

VI. REFERENCES

[1] 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.

[2] S. George et al., "Input Parameters for Modelling the ATLAS Second Level Trigger", ATLAS Internal Note, DAQ-No-070, June 1997.

[3] CACI International Inc, 1100 North Glebe Road, Arlington, Virginia 22201.

[4] J. C. Vermeulen, "Simulation of Data-Acquisition and Trigger Systems in C++", EAST note 93-22, October 1993 and "New Computing Techniques in Physics Research III", World Scientific, 1994, pp. 107-112.

[5] S.Hunt et al, "SIMDAQ - A System for Modelling DAQ/Trigger Systems", IEEE Trans. on Nucl. Sci. 43, no.1, pp. 69 - 73.

[6] SUIT, the "Simple User Interace Toolkit" is available via anonymous ftp fromuvacs.cs.virginia.edu.

[7] "The T9000 Transputer Hardware Reference Manual", Inmos Ltd, Inmos document number 72 TRN238 01.

[8] "The STC104 Asynchronous Packet Switch", Data sheet, April 1995, SGS Thomson MicroElectronics.

[9] R. Heeley at al, "The Application of the T9000 Transputer to the CPLEAR Experiment at CERN", Nucl.Instrum.Meth. A368,1996, pp. 666-674.

[10] R.W. Dobinson et al., "Triggering and Event Building Results Using the C104 Packet Routing Chip", Nucl.Instrum.Meth. A376,1996, pp. 59-66.