SIMULATION OF PARALLEL PROCESSING SYSTEMS
The ZEUS Central Tracking Detector. A Case Study.

A. Quadt, S. Topp-Jørgensen, H.A.J.R. Uijterwaal, R.C.E. Devenish

University of Oxford, Department of Physics, Keble Road, Oxford, UK, ZEUS Collaboration

To simulate and predict the performance of a highly parallel system with dynamic loading is a non-trivial task. The online performance of such systems is characterised by two quantities, the average throughput (speed) and the processing time per event (latency). Their detailed understanding in readout and trigger systems is of vital importance for the efficient data taking of present and future HEP experiments. This paper presents a case study on the online performance of the ZEUS Central Tracking Detector (CTD) Second Level Trigger (SLT).

Key words: Parallel processing; simulation; trigger.

1 Introduction

The optimal use of parallel processing systems requires an equal distribution of tasks on the different processors. Their number and the complexity of the tasks also has to be balanced in order to minimize idle time of the CPUs and make optimal use of the resources. These aspects are particularly important for online trigger and data acquisition systems in experiments where the loading with event data is dynamical and varies considerably with time and detector region. The online performance of such systems is mainly characterised by two quantities, the average throughput (speed) and the processing time per event (latency). Only their detailed understanding allows the identification of bottlenecks and guarantees therefore the most efficient data taking. The simulation of the performance of such systems is a crucial tool in such studies and allows their optimisation via the predicition of speed and latency resulting from hardware or software modifications.

2 The CTD-SLT.

The ZEUS experiment [1] uses a 3-level trigger system to cope with the high interaction rate of 10^5 Hz at the HERA collider. The first 2 levels each consist of a pipelined trigger system per detector component and a global decision unit. The third level consists of a computer farm. In 1995 and 1996, typical rates of the 3 trigger levels were $500, 50$ and $5 - 10$ Hz respectively. The CTD-SLT [2] and readout consist of 130 INMOS T800 Transputers (TP) [3], running a parallel processing track and vertex reconstruction algorithm, coded in OCCAM. The TPs, clocked at 25 or 30 MHz, can exchange data with other TPs via bi-directional links at 1.8 Mbit/sec. For trigger and readout purposes the CTD is divided into 16 sectors. In each sector 4 TPs take care of the readout of the electronics (labelled 'ROT' and 'MERGE') while 3 TPs run the SLT algorithm (SLT1-3). The remaining TPs perform data merging, determine global event parameters (SLT4) and communicate with the external control systems. As the speed of the system, which is in general simply the inverse processing time of the slowest module, is ≥ 750 Hz and therefore sufficiently high, the following study concentrates on the latency. The CTD-SLT latency, in 1995 and 1996 measured to be up to 24 msec (average $\approx$ 10 msec), is significantly longer than for most other ZEUS components (≤ 3 msec) due to the complexity of the system. The long CTD latency can therefore affect the others, which, in turn, contribute to the dead time of the ZEUS experiment.

3 The Simulation.

The simulation of the CTD-SLT online performance is based on measurements of the input and output data volume distributions of every SLT processing step, and the determination of mean and maximum processing time on each stage in the entire readout and trigger chain. Random numbers are generated according to the positive half of a gaussian. The processing times of the readout Transputer as well as that of SLT1 and SLT2 per event in one sector are approximated by such a random number, scaled by an appropriate factor. The procedure is motivated by the observation that the processing times on all readout and trigger stages are linear functions of the input data volume. The time on SLT3 is assumed to be proportional to the sum of random numbers from two adjacent sectors while the processing is assumed not to start before SLT1 and SLT2 have finished their task in both sectors, reflecting the TP

communication structure. The data volumes involved are also estimated by scaling the random numbers, except on SLT3 where a non-linear function is used. The time spent merging the output data from all 16 sectors is estimated from the data volumes and the network topology. The processing time on SLT4 is simulated to be a quadratic function of the relevant data volume.
The simulation is coded in FORTRAN77. Apart from the TP network topology further input quantities are the data volumes and processing times on the different stages of the system and their functional relations.

4 Results.

Table 1 shows the comparison of the maximum processing times for the different readout and trigger stages, as measured in 1995/96, to the simulation of the present system (SIM1). It also shows the simulation results of a possible future scenario (SIM2) with 40 MHz TPs[1] on SLT1 and SLT2 (30 and 25 MHz at present), 30 MHz TPs on SLT3 (25 MHz at present) and the processing time on SLT1 and 2 reduced by 30% and 20% respectively, resulting from software developments. SIM2 also assumes a 60% reduction in processing time on SLT4, which is considered to be realistic but still has to be realised.

processing step	max. time in 95/96 [$msec$]	max. time in SIM1 [$msec$]	max. time in SIM2 [$msec$]
ROT	3.00	3.10	3.10
transfer to SLT1	0.44	0.44	0.44
SLT1	3.58	3.50	2.13
transfer to SLT2	0.22	0.22	0.22
SLT2	3.71	3.50	1.67
transfer to SLT3	0.11	0.11	0.11
SLT3	4.42	4.02	3.30
transfer & merge	2.80	2.70	2.70
SLT4	6.02	6.60	2.50
latency	24.26	24.19	~ 16.17

Table 1
Measured processing times compared to the simulation of the present and a potential future hardware and software setup.

[1] A 40 MHz TP prototype has been used in these timing studies; regular production of these faster TPs is expected in the near future.

Figure 1 shows the latency distribution, as measured in 1996, compared to the simulations SIM1 and SIM2. SIM1 provides an adequate overall description of the shape of the latency distribution in the range latency $\geq$ 6 msec, in particular of the tail. The simulation does at present not take empty events or detector noise into account resulting in an undershooting of the distribution at very low latencies. This effect, however, is of no particular importance as it only reduces the average latency. SIM2 shows a drastic latency improvement with a slight change in the shape of the distribution and a maximum value of only 16 msec.

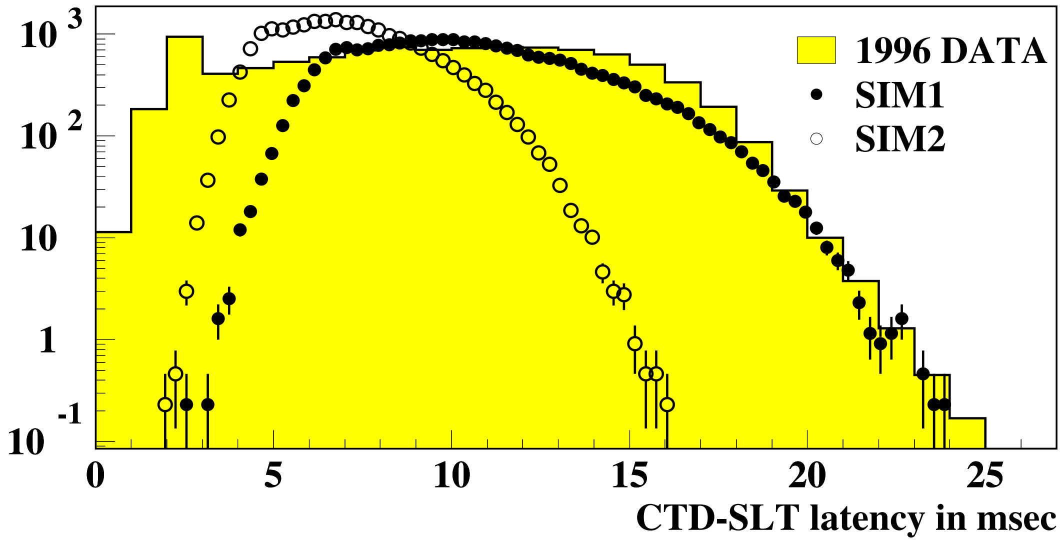

Fig. 1. *CTD-SLT latency distribution as measured in 1996 (shaded histogram) compared to the simulation of the corresponding trigger and readout system (SIM1, closed circles) and a possible future scenario (SIM2, open circles).*

Despite the simple approach of the model the performance of the current hardware and software setup is well described by the simulation. The data transfer time is negligible, except for the merging between SLT3 and SLT4. For the time reduction obtained in scenario SIM2 the hardware upgrade to faster TPs is found to be as important as the software development on SLT1, 2 and 4. The precise identification and modification of bottlenecks in the system results in a vastly improved performance using the full potential of the CTD-SLT. The simulation of the CTD-SLT timing performance serves as a powerful tool to find an optimal balance between software developments and hardware upgrades of the architecture. Provided the relevant information about data volumes and local processing times is available the method presented here can be easily applied to a vast range of parallel processor networks.

References

[1] The ZEUS Collaboration, M. Derrick et al., *Phys. Lett.* **B293** (1992) 465.

[2] H.A.J.R. Uijterwaal et al., The ZEUS Central Tracking Detector Second Level Trigger, *Proceedings CHEP95.*

[3] INMOS Limited, *The Transputer Databook*, (2^{nd} edition; 1989).