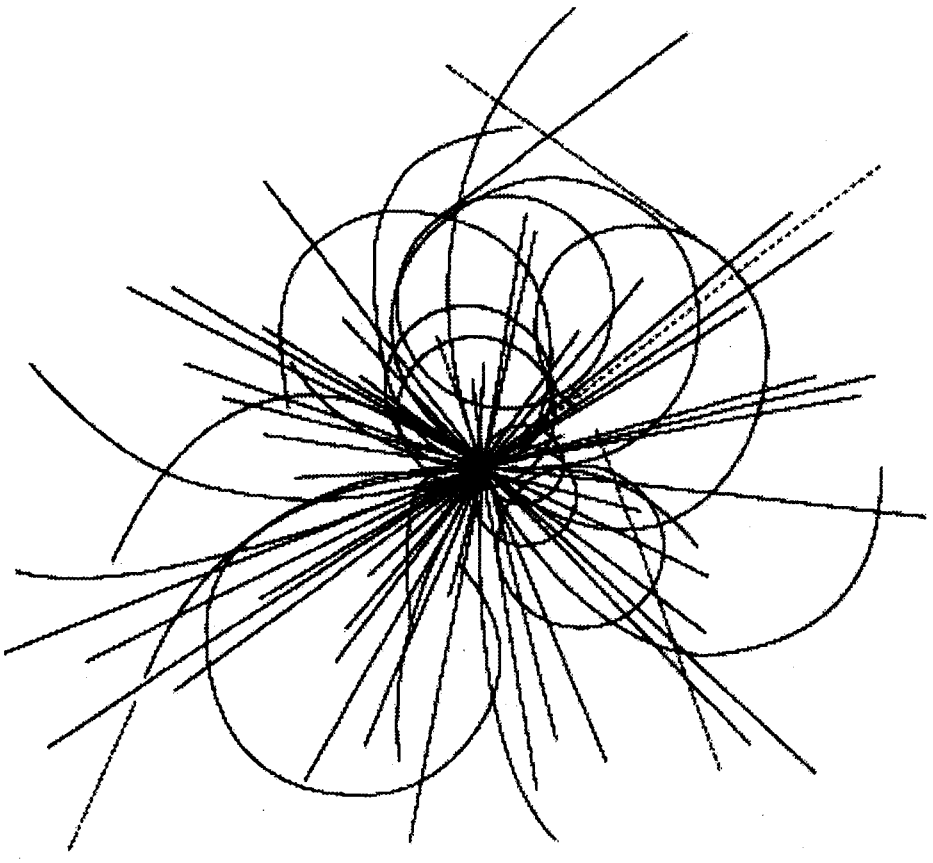
SSCL-Preprint-156

SSCL-Preprint-156
May 1992
Distribution Category: 400

M. Botlo
A. Booth
J. Dorenbosch
R. Idate
E. Milner
V. Kapoor
V. Raj
E. Wang

Data Acquisition for Super Colliders



Superconducting Super Collider Laboratory

SSCL-Preprint-156

Data Acquisition for Super Colliders*

M. Botlo, A. Booth, J. Dorenbosch, R. Idate, E. Milner,
V. Kapoor, V. Raj, and E. Wang

Superconducting Super Collider Laboratory†
2550 Beckleymeade Ave.
Dallas, TX 75237

May 1992

*Presented at the Como Conference held in Como, Italy, May 1992.
†Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

Data Acquisition for Super Colliders

M. Botlo[a], A.W. Booth[a], J. Dorenbosch[a], R. Idate[a], E.C. Milner[a], V.S. Kapoor[a], V. Raj[b] and E.M. Wang[a]

[a] Physics Research Division, Superconducting Super Collider Laboratory,
2550 Beckleymeade Avenue, Dallas, TX 75237 USA*

[b] The University of Texas at Arlington, Arlington, TX 76019 USA

This paper describes simulation studies undertaken at the Superconducting Super Collider, in which the behavior of different Data Collection protocols was investigated as a function of input data variations, bandwidth fluctuations, and input channel correlations . We have sent three kinds of input data through our models of DAQ architectures: fixed size data (every channel carries the same amount of data), exponential distributed data sizes with means coming from physics and detector simulations, and physics data directly. We find differences in the performance, depending on the input data and the DAQ-architecture we study. We conclude that DAQ models should use input data distributions which are as close as possible to reality.

1. INTRODUCTION

Designing Data Acquisition for future Supercolliders is a challenging task: While most of the system parameters such as channel count, bandwidth requirements, timing constraints change by large factors, the system deadtime should stay small. In work already published [1-2] we have discussed several DAQ architectures which include Frontends (FE), Data Collection Chips (DCC) forming Data Collection Networks (DCN), Level 1(L1) trigger, Segment Output Buffers (SOB) to general purpose processors for higher level triggering (L2/L3), all controlled by a Gating Logic. Computer models describing the behavior of those building blocks of DAQ systems were constructed and the influence of input parameters to the models such as transmission protocols, bandwidth variations, transmission times, error conditions on the throughput and deadtime of the system were measured.

In the present paper we investigate the effect of feeding the DAQ models with realistic data. Among the questions we try to answer are: What is the effect of zero suppression on the throughput of the DAQ? Do data volume fluctuations and data correlations between detector channels degrade the DAQ efficiency? Are simple data distributions as input to DAQ simulations as accurate as more sophisticated data inputs?

The outline of this paper is as follows: After describing the DAQ architectures under study we briefly sketch the techniques and tools we have developed. We then proceed to a discussion of input data fed into the DAQ-models and finally derive results.

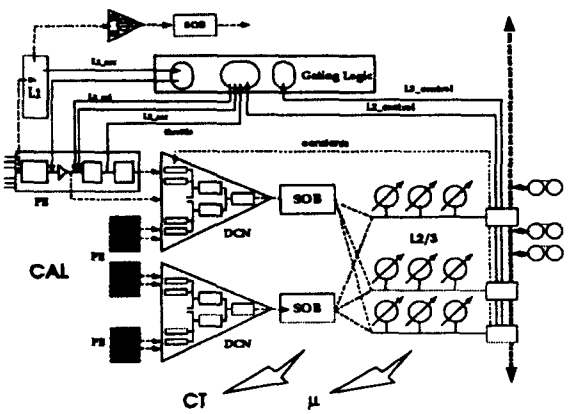


Figure 1: The basic DAQ architecture under study.

* Operated by the Universities Research Association, Inc., for the US Department of Energy under Contract No. DE-AC02-89ER40486

2. ARCHITECTURES

The basic Data Acquisition architecture under study is given in Fig.1. Frontends (FE) receive data every bunch crossing from several detector channels and hold a fraction of them in analog and digital FIFO buffers until a positive Level2 Trigger decision has arrived (signal L2_acc in Fig.1). At this point our study begins. Data are moved from all FE's to the DCN, where DCC's examine the many event fragments for integrity and combine them to larger data packets. After the complete event fragment for one DCN is assembled the data are held in the SOB - a large buffer - ready to be routed to a L3 processor. The DCN performs a simple parallel to serial dataflow conversion.

DCC's behave differently, depending on the transport protocol they use. We have described several 'push' and 'pull' mechanisms [1].

In the push-approach data packets are arriving asynchronously at the input ports of a DCC. The packets consist of several bytes of header information: inclusive wordcount, Level2 identifier, error status and other items we may want to carry, followed by the actual data. The DCC combines packets from each of its input ports, strips the individual packet header and generates a new header with a larger wordcount. As long as the 'buffer almost full' signal from the next downstream DCC has not been asserted the data is transmitted.

In pull protocols, the receiving DCC has no input buffers but only an output buffer. When the DCC finds data in upstream buffers it executes a transfer into its own output buffer.

Different implementations of push and pull architectures are imaginable and were evaluated. Here we use only one generic push and pull protocol and tune the systems to have the same throughput: The transmission speeds of the links were set, somewhat arbitrarily, such that they saturated at 6 kHz event rate (L2 accept rate). The buffering in each layer of the DCN was set to a capacity of five event fragments containing all cells.

Buffer occupancy, the coordination of trigger control signals and deadtime accounting is monitored and controlled by a Gating Logic. In addition, the Gating Logic handles the task of throttling [3], i.e. it regulates and generates deadtime instead of allowing the DCN to overflow. The difference between trottled and non-throttled systems is that in throttling systems, when a buffer in a FE becomes full, a signal is sent to the Gating Logic which then stops the flow of events accepted by Level2 into the DCN (signal throttle in Fig. 1). In the non-throttling situation the buffers in the FE are allowed to overflow and the deadtime is determined by the amount of time these buffers are full. No attempt has been made to throttle from within the DCN. Since the Gating Logic is more accessible for monitoring, throttling might be preferable to creating and measuring deadtime within the DCN itself. Also, as we shall see, this mechanism can help reduce system deadtime by a substantial amount.

A set of Frontends (here 512) and one DCN form a readout segment which has an output buffer where all the data from participating frontends are brought together. The DCN has three layers of DCC's with 64 chips in the first, 8 DCC's in the second, and one DCC in the last layer. A large derandomizing buffer (SOB) holds the data until a processor or event builder is ready. In the present study we have not included the event builder and processing farm. They have been reported elsewhere [4].

3. TECHNIQUES

The hardware blocks and protocols of the architectures described above have been modeled using the object-oriented discrete event simulation language MODSIM II [5]. To connect DAQ models to detector data, an interface for binary data, generated previously by a separate detector simulation program, was built. Whenever a new L2 trigger signal is generated, each FE connects to the data file and fetches its data values instead of firing a random number generator. Using a graphical user interface (GUI) the user can configure a simulation run by requesting specific physics data or selecting one out of several numeric data generators. Furthermore, the connection topology between FE's and detector channels, the amount of bytes per hit cell, and zero suppression thresholds in EM and HAD can be specified.

After all parameters for a simulation run are selected, a connection between the GUI and a target

SPARC compatible workstation is established and the job is executed on the target. The main output of the simulation is written to a file specific for one set of runs (normally six different L2 frequencies run on six different workstations) in a form directly readable by a spreadsheet able to display the outcome of the runs in graphical format.

4. INPUT DATA

From a Data Acquisition point of view it is important to study the influence of data volumes, their fluctuations, and event rates on the overall throughput of the system. In general, we can distinguish two cases: Either event sizes are fixed and no noise suppression of data below threshold is performed, or event sizes are random. Data are filtered either in the FE or somewhere in the DCN and only those data which are worth keeping are shipped towards the event builder. Data volumes are now - depending on the zero suppression threshold - considerably reduced, but potentially important information is irrecoverably lost.

For this investigation we have choosen to read approximately 15000 calorimeter channels (30 channels/FE) into one DCN, although technically any subdetector could have been used. The reason for the choice is that a calorimeter readout exhibits large data volume fluctuations and large correlations between neighboring channels. Below we report simulation results for three data size distributions: fixed, exponential, and physics data.

4.1 FIXED DATA

Under this input condition, 60 bytes of information are generated by each FE for each Level 2 trigger accepted event. This corresponds to two bytes information for each of the 30 calorimeter channels in each FE. No data reduction or zero suppression occurs, resulting in more than 180 MByte/sec transmission load at a 6 kHz Level2 accept rate. Because the data sizes are fixed, parts of the DCN behave the same. All buffers in a given layer of data collection contain the same size data packets. When data are transmitted from one layer to the next, all links send data at the same time.

4.2 EXPONENTIAL DATA

FE data sizes, generated according to an exponential distribution can be made to mimic many of the features of physics data. For us the main defect of this distribution is that it does not describe channel correlations and uneven data loads in different parts of the detector. Furthermore, detector specifics such as noise and event pileup are not described by such random distributions.

4.3 PHYSICS DATA

We have used PYTHIA5.6/JETSET7.3 [6] and a shower parametrization [7] to simulate proton-proton interactions and a calorimeter . The following assumptions were made:

Each FE in the central calorimeter region is connected to 16 electromagnetic (EM) cells overlapped by 4 hadronic (HAD) cells of a calorimeter with no longitudinal segmentation in stripes along azimuth ϕ. The cell size is 0.05x0.05 in pseudorapidity η and ϕ. for EM cells and 0.1x0.1 for HAD. EM exists only in $|\eta| < 3$ while HAD goes to $|\eta|=6$. Above $|\eta|=3$ we have choosen to connect 8 HAD cells along ϕ to one FE.

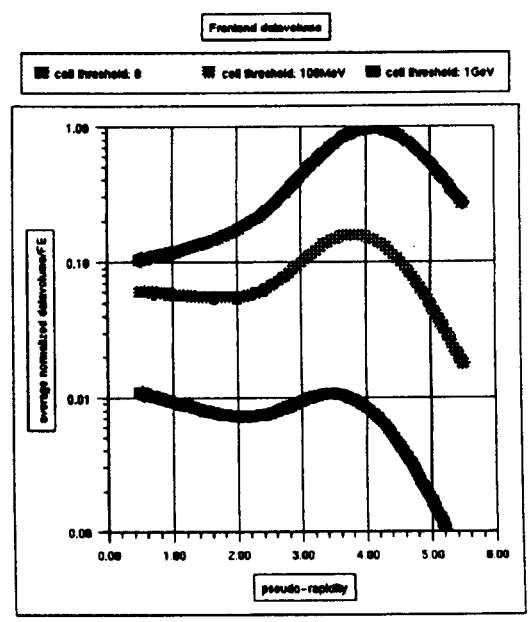


Figure 2. Data volume reduction per FE for different cell threshold vs pseudo-rapidity.

Event pileup is taken into account. Previous bucket information is added and weighted by a bipolar pulse shape: a rectangular pulse with amplitude +1 for the first bucket and amplitudes -1/2 for the following two buckets is used.

Electronic noise is not included.

A representative sample of data expected at Supercolliders (QCD-jets) was generated and filtered, demanding Pt cuts on the hard scatter outgoing partons. This yields a trigger rate of up to 10 kHz, depending on the Pt threshold.

Fig. 2 shows the average datavolume arriving at one FE as a function of η for different Et cell thresholds, normalized to the non-zero suppressed case.

Note that we now have to attach addresses pointing to each readout channel that fired. We decided on an address length of 2 bytes, effectively doubling the data volume for each detector channel above threshold. The plot shows a strong dependence of datavolume on the level of zero-suppression. Thresholding at zero presents certainly the highest datavolume we can expect. In real life that would correspond to a cut just above electronic noise. A threshold of

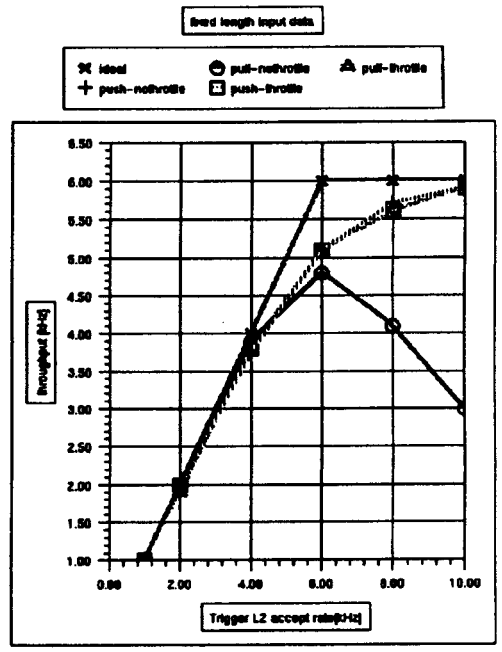


Figure 4: Throughput of push and pull protocols as function of trigger level2 accept rate for fixed length input data.

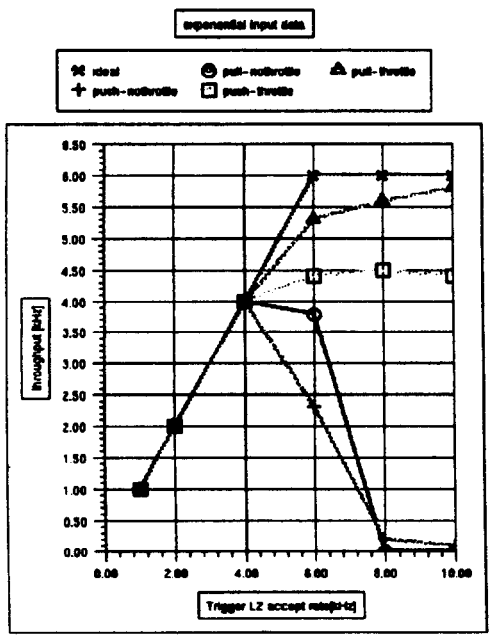


Figure 3: Throughput of push and pull protocols as function of trigger level2 accept rate for exonentially distributed input data.

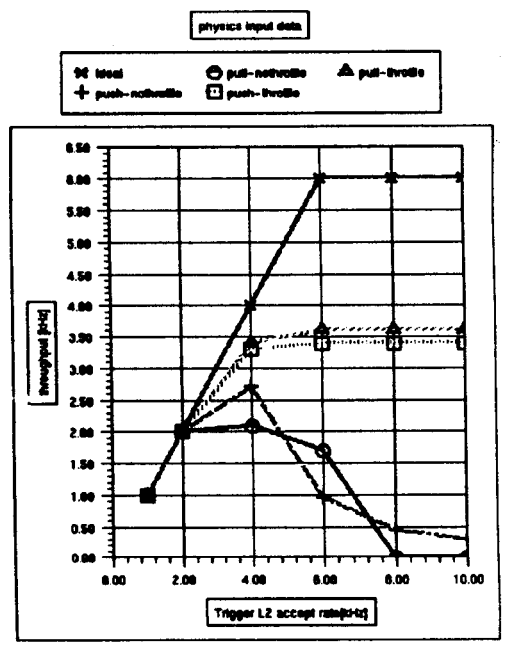


Figure 5: Throughput of push and pull protocols as function of trigger level2 accept rate for physics input data in the central rapidity region.

1GeV/cell, on the other hand, is unacceptably high for physics analysis but might be used temporarily for L2-algorithms. We choose a threshold of 100 MeV Et/cell which reduces the amount of data to be read out by a factor of 20 in the central region. Clearly, the higher the cell threshold the lower the DAQ bandwidth requirement. In a forthcoming study we examine the influence of thresholding on physics signal efficiencies and background rejection[8]. As we will see in the next section, cell to cell correlations and data volume fluctuations reduce this apparent gain in throughput for a given DCN.

5. SIMULATION RESULTS

Figures 3, 4, and 5 show the throughput of the two architectures (push and pull, each with and without throttling enabled) as a function of trigger level 2 accept rate, ranging from 1kHz to 10kHz. We define 'throughput' as the number of events transported from the FE's through the DCN to the SOB without any loss of information.

The following observations can be made:

Throttling (in any transport protocol) delivers more throughput in systems with fluctuating data volumes. Network traffic can be controlled better with throttling and links are not loaded with data which would have to be discarded anyway at some later stage .

For fixed data sizes (Fig. 4), push without throttling performs much worse than the other protocols as soon as the links become saturated. The reason is that push exhibits a great deal of synchronism; data enter and leave each FE buffer at the same instant, and data arrive at each DCC in all layers of the network at the same time. This means that if one FE fails to accept an event fragment then all other FE's are also unable to accept their data for the same event (natural throttling). The deadtime generated by the FE's will be very near to the percentage of incomplete events, and no bandwidth is wasted on transmitting incomplete events. On the other hand, pull exhibits less synchronism: The data from the FE's connected to a DCC are taken out serially and moved to the DCC output buffer. If new data arrive when the content of, say, FE1 to FE4 was already moved, then FE5 to FE8 might still overflow. Since there is no synchronization (compare with pull-throttle in Fig. 4), i.e. there is no throttling mechanism in place, bandwidth is wasted and more events have missing fragments.

For architectures which use a throttle mechanism and handle zero-suppressed data, pull shows the best performance (Fig.5). The difference between pull and push is larger the more randomly the data sizes are distributed (Fig. 3). Here again, having no throttle strategy results in a sizeable loss of data at higher L2 frequencies.

Balancing of the data load over the DCN is essential. Data collection will degrade as soon as one channel overflows. All other channels will remain under-utilized. Thus, data volume fluctuations have to be smoothened out by installing sufficiently large buffers. With the parameters used in this simulation (5 full event fragments/buffer) fluctuations from jets pose no problem. However, tuning the network for data volumes expected in the central region of the calorimeter (Fig.2) would, in the example used for this study, result in a serious bandwidth deficiency for the readout of the forward area.

By comparing the throughputs of Fig. 3 (exponential input data) and Fig,.5 (physics input data), we can conclude that effects coming from physics processes and detector response should be included in the evaluation of DAQ-systems. Even detailed simulations might not give the correct performance numbers for some architectures if only the DAQ in itself is studied and data inputs are derived from a simple random generator.

6. SUMMARY

We have studied the influence of 'physics' on the performance of DAQ-architectures. We find that throttling L2 trigger accepts, i.e. converting accepts into rejects, whenever a buffer in the DCN is full delivers best performance for acquisition systems which have to read random data volumes. For fixed size data, any transmission protocol, as long as the DAQ uses throttling, is satisfactory. Finally, DAQ-systems should be studied with as realistic input data as possible. The most accurate description and simulation of acquisition architectures might deliver wrong results if data volumes and correlations be-

tween readout channels are not taken into account.

REFERENCES

1. A. Booth et. al., Simulation Studies of Data Acquisition Systems at the Superconducting Super Collider, SSCL-SR-1148 (1992)
2. M .Botlo et. al., Data collection studies for the Superconducting Super Collider NIM A315 (1992) 472-477
3. E.C.Milner et. al., Data Acquisition Studies for the Superconducting Super Collider, Conference Record of the 1991 IEEE Nuclear Science Symposium, Santa Fe, NM, 1991
4. V.S. Kapoor, Modeling and Simulation of an Event Builder for High-Energy Physics Data Acquisition Systems, Workshop on Data Acquisition and Trigger System Simulations for High Energy Physics, SSCL-SR-1211 (1992)
5. proprietary product of CACI Products Company, 3344 N. Torrey Pines Court, La Jolla, CA-92037, USA
6. H.U.Bengtsson and T.Sjöstrand, Proceedings of the Superconducting Super Collider, Snowmass (1986), p. 311.
7. A. Bay et. al., Proceedings of High Energy Physics in the 1990s, Snowmass (1988), p. 882.
8. E.W.Wang et. al., Proceedings of Calorimetry in High Energy Physics, Corpus Christi (1992), to be published.