INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®
RICE UNIVERSITY

Study and Implementation of a High Level Trigger for the STAR Experiment

by

Martin J. DeMello

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE Master of Science

APPROVED, THESIS COMMITTEE:

Pablo Yepes
Pablo Yepes, Chair
Senior Faculty Fellow,
Department of Physics and Astronomy

Gordon S Mutchler
Professor of Physics and Astronomy

Qimiao Si
Associate Professor of Physics and Astronomy

Houston, Texas

November, 2001
ABSTRACT

Study and Implementation of a High Level Trigger for the STAR Experiment

by

Martin J. DeMello

We have designed and implemented various components of the STAR Level 3 trigger, including a Sector Level 3, a Global Level 3 and a Sector Broker. In addition, we have implemented and studied a small prototype of the STAR tracking farm, using Intel dual Pentiums connected via an ethernet ring.
Acknowledgments

I'd like to thank my advisor, Pablo Yepes, for all his support and patience. Thanks also to Naresh Sen, for many helpful discussions, Angela Bellavance, whose LaTeX templates saved me a lot of time and effort, and all my friends who rallied round with encouragement when my thesis ran behind schedule.
Contents

1 Heavy Ion Physics 1

1.1 Quarks and Gluons .............................. 1

1.2 Quark confinement ............................. 3
  1.2.1 Deconfinement ........................... 4
  1.2.2 Strangelets .............................. 7

1.3 Quark Gluon Plasma Signatures ............... 8
  1.3.1 Kinematic Probes ......................... 8
  1.3.2 Electromagnetic Probes ................. 9
  1.3.3 $J/\psi$ suppression .................... 9
  1.3.4 Strangeness Enhancement ................ 10
  1.3.5 Jet Quenching .......................... 10
  1.3.6 Disordered Chiral Condensates ........... 11

2 The Star Experiment 12

2.1 Particle Detectors ........................... 12

2.2 RHIC ........................................ 13
2.3 STAR .................................................. 14
2.4 The STAR components .............................. 14
   2.4.1 Time Projection Chamber: .................. 14
   2.4.2 Fast Detectors ............................... 16
   2.4.3 The trigger system ............................ 22

3 Prototype Online Analysis Farm ...................... 24
   3.1 Introduction .................................... 24
   3.2 Setup ........................................... 25
      3.2.1 The Dispatcher ............................ 27
      3.2.2 The Analyzers ............................. 28
   3.3 Data Processing ................................. 28
   3.4 Event Generation ............................... 30
      3.4.1 Detector Geometry .......................... 31
      3.4.2 Track Generation ........................... 31
      3.4.3 Hit Generation .............................. 33
   3.5 Analysis Farm Results .......................... 34
      3.5.1 Single Event ............................... 34
      3.5.2 Event Size ................................. 35
      3.5.3 Number of PCs .............................. 37
      3.5.4 One-way data transfer ..................... 39

4 The STAR Level 3 Trigger ............................ 44
   4.1 Hardware Architecture .......................... 45
4.2 Software Architecture ........................................ 47
   4.2.1 Event Display ........................................ 49

4.3 Data Processing .............................................. 49
   4.3.1 Cluster Finder ........................................ 50
   4.3.2 Track Finder .......................................... 50

4.4 Sector Level 3 Implementation ............................... 51
   4.4.1 SL3 State Machine .................................... 51
   4.4.2 Thread-based Implementation ......................... 53
   4.4.3 Sector Broker Implementation ....................... 55
   4.4.4 Communications and Networking ..................... 56

4.5 Global Level 3 Implementation ............................. 56
   4.5.1 The Message Queue .................................. 58
   4.5.2 The Event Pool ...................................... 59
   4.5.3 GL3-SL3 interaction ................................. 60
   4.5.4 Algorithms .......................................... 60

5 Conclusions .................................................... 61
Chapter 1

Heavy Ion Physics

1.1 Quarks and Gluons

The standard model, the currently accepted theory of the fundamental particles and their interactions, divides particles into two main classes, the spin-1/2 fermions and the spin-1 bosons. The fermions further divide into leptons and quarks, and form the constituent particles of matter. The fermions interact via four forces; gravity, electromagnetism, and the weak and strong nuclear forces. (The standard model only deals with the latter three forces). These forces are mediated by gauge bosons; an interaction between two fermions consists of an exchange of a virtual gauge boson. The relative strengths of the forces are summarized in table 1.1.

The quarks are six in number, up ($u$), down ($d$), charm ($c$), strange ($s$),
<table>
<thead>
<tr>
<th>Force</th>
<th>Boson</th>
<th>Relative Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gravity a</td>
<td>graviton</td>
<td>$10^{-40}$</td>
</tr>
<tr>
<td>Electromagnetic</td>
<td>$\gamma$</td>
<td>$10^{-13}$</td>
</tr>
<tr>
<td>Weak</td>
<td>$W^+, W^-, Z^0$</td>
<td>$10^{-2}$</td>
</tr>
<tr>
<td>Strong</td>
<td>gluons</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1.1: The four fundamental forces

<table>
<thead>
<tr>
<th>Quark</th>
<th>Symbol</th>
<th>Charge</th>
<th>Mass (GeV$/c^2$) a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>u</td>
<td>+2/3</td>
<td>4.10^{-3}</td>
</tr>
<tr>
<td>Down</td>
<td>d</td>
<td>-1/3</td>
<td>8.10^{-3}</td>
</tr>
<tr>
<td>Charm</td>
<td>c</td>
<td>+2/3</td>
<td>1.5</td>
</tr>
<tr>
<td>Strange</td>
<td>s</td>
<td>-1/3</td>
<td>0.15</td>
</tr>
<tr>
<td>Top</td>
<td>t</td>
<td>+2/3</td>
<td>176</td>
</tr>
<tr>
<td>Bottom</td>
<td>b</td>
<td>-1/3</td>
<td>4.7</td>
</tr>
</tbody>
</table>

Table 1.2: The quarks and their properties

a from SLAC: http://www2.slac.stanford.edu/vvc/theory/quarks.html

top ($t$) and bottom ($b$), and grouped into three generations, (u,d), (c,s) and (t,b), with the quark masses varying sharply between generations. The quark properties are summarized in Table 1.2. There are also the corresponding six antiquarks, \{\bar{u}, \bar{d}, \bar{s}, \bar{c}, \bar{t}, \bar{b}\}.

Since quarks carry charge, they are subject to the electromagnetic force (mediated by the photon, $\gamma$). Quarks also have another quantum number, \textit{colour}, that is subject to the strong force. The strong force is mediated by gluons, which couple to particles possessing colour - this includes the six
quarks and the gluons themselves. Thus, unlike photons, which are chargeless, gluons are themselves subject to the strong force.

The colour (or 'colour charge') is described by Quantum Chromodynamics, and may be represented by an SU(3) group. There are three possible values of colour, labelled red, blue and green. Quarks can possess any of the three colours. Gluons carry a colour-anticolour charge, e.g. blue-antigreen. Since \{red-antired, blue-antiblue, green-antigreen\} form a linearly dependent set ¹, this gives us eight distinct gluons.

1.2 Quark confinement

Another peculiarity of QCD is that the strong force increases with inter-quark distance. Experimentally, for large \( r \) the inter-quark potential \( V(r) \) increases linearly with \( r \). The colour force between two quarks is confined to a narrow flux tube between them; increasing their separation adds enough energy to the flux tube to cause it to break into a new quark-antiquark pair, each member of which is attached to one of the original particles. The overall system thus remains grouped into a set of colour-neutral particles, and it is not possible to isolate a lone quark. This phenomenon is known as quark confinement.

¹since the combination \( r\bar{r} + b\bar{b} + g\bar{g} \) is noninteracting
<table>
<thead>
<tr>
<th>Atoms</th>
<th>( \sim 10^{-10} ) m</th>
<th>10^3 K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nuclei</td>
<td>( \sim 10^{-14} ) m</td>
<td>10^{10} K</td>
</tr>
<tr>
<td>Nucleons</td>
<td>( \sim 10^{-15} ) m</td>
<td>10^{11} K</td>
</tr>
<tr>
<td>Quarks</td>
<td>( \sim 10^{-18} ) m</td>
<td>10^{13} K</td>
</tr>
</tbody>
</table>

Table 1.3: Dissociation temperatures at which the bound states of various objects are broken

1.2.1 Deconfinement

Quark confinement implies that at normal temperatures and densities, matter consists mainly of colour-neutral hadrons, whose constituent quarks are tightly bound. However, it is thought highly likely that at high enough temperatures, a phase transition takes place in which the quarks and gluons are deconfined and can move freely over the high temperature region.

While such conditions do not presently exist (except, perhaps, in the interior of neutron stars), it is thought that the early universe, up to about 10 \( \mu s \) after the big bang, consisted of such a deconfined state. As the universe cooled down, hadronization took place, and the quarks were confined into 'normal' matter.

Table 1.3 shows the temperatures at which the bound states of objects at various scales are broken.

Thus the energy needed to deconfine the quarks in a nucleon is two orders of magnitude higher than that needed to split the nucleus, for example.

As temperatures rise, a phase transition is expected to occur, resulting in a state called a quark gluon plasma (QGP), which displays neither con-
finement nor chiral symmetry breaking. (‘Phase transition’ since there are certain order parameters that vanish at high temperatures, so that we expect some sort of discontinuity). Figure 1.1 shows the nuclear matter phase diagram, with the deconfinement phase transition marked, and several accelerators labelled with the regions they operate in.

Lattice gauge theory simulations indicate a first order phase transition, with $T_c \approx 260$ MeV. Around that temperature range, the free energy of a deconfined quark drops to a finite value, and the quark condensate, an order parameter that measures chiral symmetry, likewise drops sharply.\(^2\)

One way to probe conditions at such energies is by means of heavy ion colliders. Earlier accelerators, such as AGS\(^3\) and SPS\(^4\) used high energy proton beams; later, these were adapted to use heavy nuclei. SPS has attained a beam energy of 160 GeV/nucleon using lead nuclei, and AGS currently serves as an injector for the Relativistic Heavy Ion Collider (RHIC), which collides gold ions at a centre-of-mass energy of $\sqrt{s} = 130 - 200$ AGeV.

Both SPS and RHIC are operating in the energy range where QGP formation is expected to occur - indeed, SPS has already claimed to have detected signs of a QGP, before it was shut down to make way for the forthcoming Large Hadron Collider.

---

\(^2\)Ginzburg-Landau Type Effective Theory for Deconfinement and Chiral Transitions in QCD, Akio Hosoya, FERMILAB-PUB-86/113-T August, 1986

\(^3\)The Alternating Gradient Synchrotron at Brookhaven National Laboratory (BNL)

\(^4\)The Super Proton Synchrotron at the European Organization for Nuclear Research (CERN)
Figure 1.1: Nuclear Matter Phase Diagram
1.2.2 Strangelets

Another possibility at RHIC energies is the formation of strange quark matter. Strange quark matter (SQM) or strangelets are clusters containing a large number of delocalized quarks, in multiquark droplets. Multiquark states consisting only of u- and d-quarks have masses considerably larger than ordinary nuclei. Droplets of SQM, which would contain approximately the same amount of u-, d- and s-quarks might also be denser than ordinary nuclei. They might exist as long-lived exotic isomers of nuclear matter inside strange neutron stars.

There is also some possibility that strangelets might indeed be stable, based on the fact that:

1. The (weak) decay of a s-quark into a d-quark could be suppressed or forbidden because the lowest single particle states are occupied.

2. The strange quark mass can be lower than the Fermi energy of the u- or d-quark in such a dense quark droplet. Opening a new flavour degree of freedom therefore tends to lower the Fermi energy and hence also the mass per baryon of the strangelet, maybe even below the proton mass.

SQM may then appear as a nearly neutral state (the total charge $Q(u + d + s) \sim 0$).
1.3 Quark Gluon Plasma Signatures

The problem with experimental study of quark deconfinement is that the plasma has an extremely small lifetime (\(5 \times 10^{-23}s\)). The QGP formation cannot, therefore, be detected directly, but must be inferred from some of its side effects. Also, QGP signals have to be distinguished from the background emitted by the hot hadronic gas phase following hadronization of the plasma, and are modified by final state interactions in the hadronic phase. Nonetheless, there are several ways in which the formation of a QGP might be tested, some of which are described below.

1.3.1 Kinematic Probes

This is a class of signatures based on the determination of the energy density \(\epsilon\), pressure \(P\), and entropy density \(s\) of superdense hadronic matter as a function of the temperature \(T\) and the baryochemical potential \(\mu_B\). The phase transition would be signalled by a rapid rise in the effective number of degrees of freedom, as expressed by the ratios \(\epsilon/T^4\) or \(s/T^3\), over a small temperature range.

The usual measurable quantities related to \(T\), \(s\) and \(\epsilon\) are the average transverse momentum, the hadron rapidity distribution, and the transverse energy respectively.
1.3.2 Electromagnetic Probes

Photons and lepton pairs provide probes of the QGP interior during the hottest phase of the fireball evolution, being less affected by final state interactions (since they do not interact via the strong force). However, they have small yields, and must compete with large background signals from electromagnetic hadron decays and other processes.

One such signature is the 'dilepton anomaly'. The basic dilepton production mechanism is \( q\bar{q} \rightarrow \gamma^* \rightarrow l^+l^- \); thus, the production rate of the \( l^+l^- \) pairs should be correlated to that of the \( q\bar{q} \) pairs. Various sources of background dilepton pairs include Drell-Yan processes, hadron-antihadron annihilation and hadron resonance decay, making this a difficult signature to observe.

1.3.3 \( J/\psi \) suppression

In 1986, Matsui and Satz\(^5\) predicted that the observed number of \( J/\psi \) mesons (c\(\bar{c} \)) would be reduced by the formation of a QGP. The plasma would tend to screen the gluonic interactions between the quarks (Debye screening), thereby not permitting a stable c\(\bar{c} \) pair to form within the QGP. Furthermore, the production rate of c\(\bar{c} \) pairs by quark-quark, quark-gluon and gluon-gluon collisions is negligible at QGP temperatures, due to the large mass of the

charmed quarks. Therefore, production of $J/\psi$ mesons during rehadronization is likewise negligible.

However, a suppression is also found in proton-nucleus reactions, and hence, a hadronic origin of the effect - absorption of the $J/\psi$ in the nuclear medium - appears plausible.

1.3.4 Strangeness Enhancement

While strange hadrons are normally suppressed in hadronic reactions (relative to hadrons containing only up and down quarks), when a QGP is formed the production of hadrons with strange quarks is expected to saturate, due to equilibration of the strange quark content of the plasma (via formation of $s\bar{s}$ pairs through gluon-gluon collisions).

Therefore $\phi$ ($s\bar{s}$) meson production would be enhanced in QGP rehadronization, and $\phi$ to $J/\psi$ production ratio could provide a sensitive test for QGP formation.

1.3.5 Jet Quenching

A jet is a highly collimated beam of secondary particles emerging from a collision. The chief source of jets is the rehadronization of high-momentum partons; the hadronization products inherit the original quark's momentum.

When a QGP is formed, jets passing through the plasma interact with the dense medium, losing energy via gluon radiation, at about 1 GeV/fm
6. This jet quenching effect thus serves as another signature of the plasma formation.

1.3.6 Disordered Chiral Condensates

The temporary restoration of chiral symmetry in nuclear collisions may result in the formation of domains of disoriented chiral condensate (DCC). A DCC is a coherent excitation of the pion field corresponding to a local misalignment of the chiral order parameter. Such domains would decay into neutral and charged pions, favoring pion charge ratios \( N_\pi^+/N_\pi^- \) substantially different from one-third.

Domains of disoriented chiral condensate may also contribute to antibaryon production through the formation of topological defects in the chiral order parameter. Such defects can arise at the intersection of chiral domain walls, which carry baryon number and eventually evolve into baryons and antibaryons, possible leaving a signature of the chiral phase transition in regions of phase space that are normally baryon-poor.

Chapter 2

The Star Experiment

2.1 Particle Detectors

At the heart of high-energy nuclear and particle physics experiments is the detector. The basic concept behind a detector is simple - an interaction region is defined, in which either two particle beams moving in opposite directions, or a single beam and a fixed target, collide. Such a collision generates a shower of particles, which various detection systems attempt to identify and analyze.

Detectors divide into three main systems - tracking, particle identification and calorimetry. The tracking detectors capture the charged particles' trajectories as they leave the interaction point. Typical data inferred include particle momenta and charges.

Usually surrounding the tracking devices is the calorimetry system, which
measures the energy of charged and neutral particles. Calorimeters divide into two types: electromagnetic and hadronic. Surrounding these there may be a muon detector, since muons typically have a far lower probability of interacting with and being absorbed by the calorimeters.

Individual experiments differ in the energy and nature of the colliding particles, whether the particles follow linear or circular trajectories, and whether they involve two beams or a beam on a fixed target. Some of the main beam types include $e^+e^-$ (e.g. the decommissioned LEP at CERN), $pp$ (e.g. the upcoming LHC, again at CERN), $p\bar{p}$ (e.g. the Tevatron at Fermilab) and heavy ion (e.g. RHIC at BNL, which uses Au).

### 2.2 RHIC

The Relativistic Heavy Ion Collider (RHIC), located at Brookhaven National Lab (NY, USA) is a superconducting ring 1.2 km in diameter in which gold ions collide at center of mass energies of 100-200 GeV/nucleon.

The collider consists of two rings with ion beams circulating in opposite directions. The ion trajectories along each ring intersect in 6(4) regions where the collisions are produced and detected by four detectors: BRAHMS, PHENIX, PHOBOS and STAR. In each ring particles are distributed in 57 bunches travelling at .99995c. The collider has been designed to produce $10^3 - 10^4$ AuAu collisions per second.
2.3 STAR

STAR (the Solenoidal Tracker at RHIC) (Figure 2.1) is one of the two 'large' experiments at RHIC.

The primary goals of STAR are to search for signatures of quark-gluon plasma (QGP) formation and to probe the behavior of strongly interacting matter at high energy density. The emphasis is on the correlation of many observables on an event-by-event basis, implemented via a flexible detection system that can simultaneously measure many experimental observables.

2.4 The STAR components

The base STAR system consists of three main components - the Time Projection Chamber (TPC), the Central Trigger Barrel (CTB) and the Trigger system.

2.4.1 Time Projection Chamber:

At the heart of STAR is the Time Projection Chamber (TPC) (Figure 2.2. The TPC is a large gas ionization chamber which tracks the paths of particles emerging from the collision. It essentially consists of a cylindrical drift chamber and two endcaps. The endcaps, comprising an anode wire plane and a pad plane, also serve as a multiwire proportional chamber. The TPC is a cylinder of radius 2.08m and length 4.200m, covering a region from
Figure 2.1: The STAR detector
\[ \eta = (-1.5,1.5) \]

There are 24 anode and pad planes on each end, grouped into 12 supersectors (each consisting of two inner and two outer sectors). Each supersector contains 45 rows of pads ("padrows"), with the number of pads per padrow increasing with radial distance (Figure 2.3).

Particles passing through the TPC ionise the gas inside it, releasing free electrons, which drift under the influence of a longitudinal electric field to one of the pads in the anode. As a particle passes through the TPC volume it undergoes a series of such ionising impacts, from which its trajectory can be reconstructed. Since the field is oriented along the \( z \) direction, the \( x \) and \( y \) coordinates of the electrons can be read off directly from the pad they're detected on. From the time taken to drift to the pad plane, the \( z \) coordinate can also be inferred, thus allowing the particle's position to be recorded in three dimensions. The spatial resolution achievable is 500 \( \mu \)m in \( r \phi \), and 2 mm in \( z \). The average drift velocity of an electron is 5 cm/\( \mu \)s, leading to a maximum drift time of 40 \( \mu \)s.

### 2.4.2 Fast Detectors

**Central Trigger Barrel**

The Central Trigger Barrel (CTB), shown in Figure 2.4, is an annular ring of scintillator detectors just outside the TPC. There are 240 detectors, each

\[ 1 \text{The pseudorapidity, } \eta = -\log(\theta/2), \text{ is a commonly used coordinate in high energy physics} \]
Figure 2.2: Time Projection Chamber
Figure 2.3: TPC Pad Plane
consisting of a scintillator slat and a photomultiplier, arranged in pairs in 120 aluminium trays. Each slat, when installed in the CTB, covers 6 degrees in azimuthal angle and 0.5 units in pseudorapidity.

The principle behind a scintillator detector is as follows: the slats are transparent, and constructed of a material that, when a particle passes through it, emits a flash of light. A photomultiplier is located at one end, to convert the light pulse into an analog electrical signal, which can be further processed electronically and stored.

**Multiwire Proportional Chamber**

The endcaps of each of the TPC sectors consist of a set of wires that can serve as a multiwire proportional chamber (MWC).

A multiwire proportional chamber is a charged particle detector consisting of a number of thin, parallel and equispaced anode wires, sandwiched between a pair of cathodes. The anode wires are grounded, so when the cathodes are charged the space between them is filled with a homogeneous electric field, with field lines leading from the cathodes to the anode wires.

Particles passing through the MWC ionise the gas in the chamber, and the strong field near the wires acts as a multiplication region, leading to an avalanche of electrons for each particle. The resulting pulses in the anode wires can then be read off, and particle multiplicities measured.
Central Trigger Barrel

Side View - Opposite Orientation

Figure 2.4: Central Trigger Barrel
**EMC**

A calorimeter is a detector which uses total absorption of particles to deduce their energy. Electromagnetic calorimeters are composed of material that undergoes showering when photons and electrons are incident upon it. Hadrons usually pass through and are absorbed by a surrounding hadronic calorimeter.

The STAR Electromagnetic Calorimeter (EMC) is a lead-scintillator sampling electromagnetic calorimeter in the form of a barrel and an end cap calorimeter. The barrel covers $-1 < \eta < 1$ and $0 < \phi < 2\pi$ with a segmentation in pseudorapidity and azimuthal angle of $(\Delta\eta, \Delta\phi) = (0.05, 0.05)$. The STAR EMC barrel, when fully implemented, will consist of 120 modules each covering $0 < |\eta| < 1$ and $\Delta\phi = 0.1$ (6 degrees).

**Zero Degre Calorimeter**

The Zero Degree Calorimeter consists of two hadronic calorimeters. Each of them is placed along the axis of the two incoming beams. Its purpose is to detect the neutral fragments of the incoming ions which are not deflected by the collision. Neutral undeflected fragments will be produced in nearly all the collisions. This characteristic makes this device a good detector to identify when a collision actually occurs.
2.4.3 The trigger system

The RHIC beams cross the interaction point at STAR with a frequency of $10^7$ Hz. When the collider is operating with design parameters with Au beams, a collision takes place only in one in every $10^4$ of those crossings. The first purpose of the trigger system is to identify those crossings when a collision take place. Once that is done, it needs to inform the rest of the detector, so that they provide the data produced in them in that particular crossing.

The different systems in the detector generate data in digital electronic format. The data is written on electronic media and stored for later off-line analysis. The amount of data for a violent (central) AuAu collision is around 10 Mb. Since the amount of data which can be stored for off-line analysis is limited to 40 Mb/s, further collision selection is needed. That further selection is also the purpose of the trigger system. Therefore the 'trigger' system has to discriminate, in real time, between interesting and uninteresting events and only store the latter.

The STAR trigger system is implemented in a multi-level, modular, pipelined system. Level 0, 1 and 2 use the fast trigger detectors: CTB, MWC and ZDC. Level 0 receives data from the detectors and accepts events. The other levels can only abort events.

The timescales of the triggers are set to match the TPC time structure. The Level 0 has to trigger in time to open the TPC amplification grid, which gives it a timescale of approximately 1.5 $\mu$s.
The TPC has a drift time of about 50 $\mu$s, during which time signals from the TPC pads are strobbed onto a switched capacitor array (SCA). Level 1 is an additional analysis trigger that runs during the TPC drift time, and is (arbitrarily) required to return a decision within 100 $\mu$s.

Once the SCA is filled, data begin moving through the TPC digitizer sequence, a process that requires 5 ms regardless of occupancy in the TPC. During this time, all data from the trigger detectors can be moved to a central location for further analysis. Simultaneously, additional summary data from other fast detectors (EMC, TOF) can be added to this trigger data to determine whether the event should be kept or rejected. Time allocated for Level 2 analysis depends on event type - typically 4ms for a TPC type event, or 1ms for an EMC type event, since we want a faster response time for the latter.

Level 3 is a high-level trigger that integrates the data from other detectors. In the first year of STAR operation, it only used data from the TPC. For future runs there are plans to integrate other detectors into the system. Level 3 controls the final decision to write data out to tape, and thus has to operate at the rate of a few events per second.
Chapter 3

Prototype Online Analysis Farm

3.1 Introduction

Much of the data processing in High Energy Physics experiments, like STAR, involves tasks that can be readily broken up into a number of reasonably independent sub-tasks. This means that they are inherently parallelizable, and leads almost immediately to the idea of a cluster of independent machines, to which sub-tasks are farmed out, solved and collated.

A typical raw dataset consists of an 'event' - the complete set of data collected by the detector hardware in the course of one beam collision. The information about which event a piece of data belongs to is maintained throughout the system, since data from different events is analyzed independently.
An important area in which such clusters are used is to run trigger algorithms. A trigger is a piece of hardware or software that examines events online, as they are generated, and based on its analysis, decides whether the event is worth keeping or not. (Some systems use multiple triggers, working in series).

One example of a task involved in modern triggering is track finding (see section 4.3.2). In STAR, this is accomplished using a farm of DEC Alpha machines \(^1\), connected using a proprietary Myrinet network\(^2\). Here we implement a prototype tracking farm using off-the-shelf components; in particular, using ethernet as the choice of network.

Rather than use actual experimental data, we have simulated data corresponding to different experimental setups. This was accomplished by means of an event generator, which produced Monte Carlo events with parameters abstracting those of various real detectors. This data is used to generate a set of tracks, which are then analyzed using the level 3 tracking software from STAR.

### 3.2 Setup

The setup used was a farm of 10 Dual 400 MHz Pentium II machines, running Linux 2.2.12-20smp, and connected via 100Mbit ethernet. One machine

---

\(^2\)http://www.myri.com
Figure 3.1: System setup

served as a dispatcher, reading in raw data and farming it out to the other nine analysis machines to process. Each analysis machine ran two 'analyzers' - virtual machines that the dispatcher treated as independent. The analyzers processed the data, and optionally returned the results to the dispatcher.

The machines were connected in a star configuration, with each analyzer communicating only with the dispatcher (see Figure 3.2).

Since, being a small local network, the error rate was negligible, and since performance was a far more critical issue, it was decided to use UDP rather than TCP as the network protocol, thereby optimising for speed rather than
for reliability. The main difference is that TCP guarantees reliable data transmission; every packet sent requires an acknowledgement (ACK), and if the appropriate ACK is not received before a timeout limit, the packet is retransmitted. UDP, on the other hand, simply sends out packets and makes no attempt to verify that they reach intact, or indeed, at all. This can be a problem over error-prone networks, or if data accuracy is far more critical than speed. However, the lack of verification means it has a higher data transfer rate, which is what we are chiefly interested in.

There was also a size limit of 64k per packet; however that was not a problem for reasonable event sizes (around 400 tracks/event).

### 3.2.1 The Dispatcher

The dispatcher is the central machine in the cluster. It reads in events from a file, and sends them out to the first free analyzer it finds. The dispatcher also monitors the analyzers so that it can send a 'time out' message to a machine that has run for too long.

The dispatcher flow diagram is as follows:

- Read in a set of events
- Read in a list of processors and their addresses
- While there are unprocessed events
  - Find a free analyzer (wait if necessary) and mark it 'busy'
- Send the data over to be analyzed
- Wait for the analyzer to acknowledge completion or time out, and mark it 'free'

In practice, the set of events could also be replaced by a loop that iterates over the same event a number of times. Since we are interested in performance rather than in the actual reconstruction results, and since both the track finder and network performance are reasonably independent of the data, this shouldn't make much difference.

3.2.2 The Analyzers

Each analyzer runs in a permanent loop that waits for an event from the dispatcher, processes it, and either returns or discards the results (see Figure 3.2). The latter is to account for the case where the dispatcher and the machine that collates the final data are entirely separate, so that the analyzer need not send anything back to the dispatcher.

3.3 Data Processing

As we have already mentioned, the purpose of the farm is to prototype the STAR tracking cluster. Tracking is one of the fundamental problems in analysis of particle accelerator data. The accelerator detection hardware can be thought of as a three dimensional grid of 'point' detectors that trigger
Figure 3.2: Dialogue between the dispatcher and analyzer

when a particle passes within a certain distance of them. This is a rather simplified view, but it is appropriate to explain the basic issues.

The particles themselves follow the path of charges moving in a homogeneous magnetic field; that is, helical trajectories. As the particles pass through the detector grid, various grid cells along the trajectory are triggered, leading to a set of 'hits' - three dimensional coordinates that indicate (to within an error margin) where the particle passed. (Some detectors have two-dimensional geometries; we only consider three-dimensional ones).

The track finder takes in a set of hits, and attempts to reconstruct the
tracks that generated them. The *tracker*, or *track finder*\(^3\) is the analysis program at the heart of the farm. It takes in a set of clusters and fits helical tracks through them (see figure 3.3). This is done on each of the analyzers, and the resulting tracks returned to the dispatcher.

![Figure 3.3: Two tracks fitted through a set of hits](image)

3.4 Event Generation

An *event*, in a real detector, is a single beam collision, which produces a set of tracks. For this study, such tracks are generated with a simulation program. Only tracks produced at beam interaction point are considered for simplicity. The data of the tracking detector is simulated by calculating

\(^3\)P.Yepes, A fast track pattern recognition, Nucl. Instr. and Meth. 380 (1996) 582.
the intersection of the tracks with its sensitive elements. Energy loss, decays and secondary interactions were not taken into account.

### 3.4.1 Detector Geometry

Our simulated detector consisted of a number of coaxial cylindrical shells. The shells were centred on the $z$ axis, and evenly spaced in $r$. The values for the various parameters were chosen to reflect some actual detectors like STAR, as shown in Table 3.1. The magnetic field was taken to be 0.5T in all cases.

We specify a $r\phi$ rather than a $\phi$ resolution to simulate a detector with larger radii shells having a greater number of detectors spaced around them, preserving the spatial rather than the angular resolution. STAR is one such detector, for example.

### 3.4.2 Track Generation

A track is parametrized by six coordinates: a point, a direction and a momentum. Here we use the vertex position, $(r_0, \phi_0, z_0)$, the pseudorapidity, $\eta$, 

<table>
<thead>
<tr>
<th>Detector</th>
<th>$R_{\text{min}}$ (cm)</th>
<th>$R_{\text{max}}$ (cm)</th>
<th>$N_{\text{layer}}$</th>
<th>$\sigma_{r\phi}$ (cm)</th>
<th>$\sigma_z$ (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (STAR)</td>
<td>50</td>
<td>200</td>
<td>45</td>
<td>0.05</td>
<td>0.2</td>
</tr>
<tr>
<td>2 (ALICE)</td>
<td>50</td>
<td>250</td>
<td>75</td>
<td>0.05</td>
<td>0.2</td>
</tr>
<tr>
<td>3 (CMS)</td>
<td>30</td>
<td>100</td>
<td>8</td>
<td>0.02</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 3.1: Some sample detector parameters
an azimuthal angle $\psi$ ($\eta$ is a function of the spherical angle $\theta$, so ($\eta$, $\phi$) define a direction) and the transverse momentum $p_T$.

$\eta$ is a good choice of coordinate here, since it approximates the rapidity $y$. In an actual event, the number of tracks in a slice $dy$ is roughly constant (whereas the number of tracks in a slice $d\theta$ varies with $\theta$). Also, differences in rapidity are invariant under a Lorentz transform (which merely adds a constant term).

(The rapidity is defined by $y = 0.5 \ln\{(p + p_z)/(p - p_z)\}$. The pseudorapidity $\eta$ is a purely geometrical term that approximates the rapidity, and is
given by $\eta = -\log(\theta/2)$).

Tracks are generated using a Monte Carlo generator. The vertex, or collision point, is assumed to lie on the detector axis, so $r_0$ and $\phi_0$ are set to 0. $z_0$ and a range of values for $\eta$ and $p_T$ are input to the generator.

The MC generator assumes a flat distribution in $\eta$ and $\psi$ (tracks are equiprobably distributed in $\eta - \psi$ space), and an exponential distribution in $p_T$, with an exponent of 0.2 (consistent with actual experimental data).

3.4.3 Hit Generation

Hits were calculated by considering the intersection of a track with each of the cylindrical shells of the detector, and blurring the resulting points with a random Gaussian in the $z$ and $\phi$ directions to simulate detector resolution (see table for detector resolutions). The radial coordinate was assumed to be exact, since it was by definition equal to the shell radius.

The hits were further binned along the $z$ and $\phi$ directions by discretizing the coordinates according to some (arbitrary) resolution in $z$ and $r\phi$ (see table). The discretized hits were then packed into a data structure, and converted back into space hits by the analyzer.

The total size of a hit was 5 bytes - 1 byte for the shell, and 2 bytes each for $z$ and $\phi$. 
3.5 Analysis Farm Results

3.5.1 Single Event

The time \(^4\) taken for a single event to cycle through the system is composed of three parts - the times taken to send, process and return the event, respectively. The time an event remains active in the dispatcher will be defined as \textit{local time}. In addition the time the analysis machine spends processing an event is referred to as \textit{remote time}. The time spent in transferring the event, as well as the time spent exchanging messages, referred to as \textit{network time}, is then given by the difference between the two times. (See Fig. 3.5).

Single event runs were carried out using one analyzer, with the number of tracks varying in steps of 10, and averaged over 1000 events. The experiment was repeated for all three model detectors.

Figures 3.6, 3.7 and 3.8 show averaged times as a function of number of tracks for events of varying sizes and for each detector. Both the remote and the network time increased fairly linearly with the number of tracks, though the network time tended to flatten out slightly as the number of tracks increased.

The timing results were then binned, and a histogram plotted for each combination of detector and number of tracks. As typical examples, figures

\(^4\)The time is read off the CPU clock, using the unix gettimeofday() system call. The gettimeofday() return value is monotonic, measuring the total elapsed time since midnight, Jan 1, 1970, and has a nominal microsecond resolution (the actual resolution depending on the hardware).
Figure 3.5: The various times taken to process a single event

3.9 and 3.10 show local time distributions with 100 and 500 tracks, for 45 and 8 shells respectively.

As the results (Fig. 3.9) show, the variation in times for a single setup (detector, ntracks) was small, with the results clustered sharply around the mode. However, in some cases the results were bimodal (Fig. 3.10)

3.5.2 Event Size

Here we investigate the effect of the event size on the throughput of the system. Each data point represents a file of fixed size, run through the system 1000 times. The system consisted of 16 analysis machines on 8 PCs.

The event size was varied from around 300 to 64k bytes/event - the maximum number of tracks in an event therefore depended on the number of
Figure 3.6: Averaged times per event as a function of number of tracks for the detector configuration with 8 shells

shells in the detector, as shown in Table 3.5.2

Two different measures of the throughput were used - events/second and Mb/second. Figure 3.11 shows the results of varying the event size from 0 to 64k, using all 8 physical machines (16 virtual analyzers).

As the event size approaches 64k (the maximum size of a UDP/IP packet)

<table>
<thead>
<tr>
<th>Number of Shells</th>
<th>Max #tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1000</td>
</tr>
<tr>
<td>45</td>
<td>570</td>
</tr>
<tr>
<td>75</td>
<td>420</td>
</tr>
</tbody>
</table>

*A maximum of 1000 tracks was imposed since to reach 64k would have required an unreasonable number of tracks.

Table 3.2: Maximum number of tracks for each detector
Figure 3.7: Averaged times per event as a function of number of tracks for the detector configuration with 45 shells

the throughput in Mb/s increases up to a certain point, though it does not reach the theoretical network limit of 10 Mb/s.

3.5.3 Number of PCs

The number of PCs running analyzer units was stepped from 1 to 8 (since these are dual processor machines, and are running two analyzers apiece we are actually going from 2 to 16 analyzers).

Plots were taken for every combination of detector and event size below:

<table>
<thead>
<tr>
<th>Number of Shells</th>
<th>8, 45, 75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event size (tracks)</td>
<td>10, 100, 400</td>
</tr>
</tbody>
</table>

The plots were grouped by event size. Figure 3.12 shows, for all three
Figure 3.8: Averaged times per event as a function of number of tracks for the detector configuration with 75 shells

detector setups, and for a low event size (10 tracks), the effects of increasing the number of PCs from 2 to 8. Figure 3.13 repeats the experiment for an event of intermediate size (100 tracks), and figure 3.14 uses a large event (400 tracks).

Since we were varying two parameters here, the event size and the number of machines, we notice two separate trends manifesting themselves. For the largest event, the performance increased linearly as more machines were added to the system. However, for smaller events, the performance increased at first, but then flattened out sharply. This suggests that there is a certain crossover point below which the events are small enough that the overhead involved in shipping them to the analyzers over the network becomes sig-
significant. This makes sense, since the network transfer time doesn’t decrease linearly with the packet size, whereas the time taken to analyze a given event is a linear function of its size.

3.5.4 One-way data transfer

A performance increase could conceivably be gained if we do not find it necessary to have the analyzer units send the processed event back to the dispatcher. This is not unrealistic, since in practice the analyzers could be sending the finished data on to another set of machines altogether.

Shown in figure 3.15 are the plots for a varying number of machines and a varying event size, with superimposed graphs of one-way and two-way data transfer.
Figure 3.10: **Local time (ms) - 8 Shells, 500 tracks**

Note that there is no difference between the two datasets. This is due to the fact that we are using full duplex ethernet network, and therefore data transfer in one direction is independent of data transfer in the other direction.
Figure 3.11: Throughput, in Mb/s and events/s, through 8 machines, while varying the event size from 0-64k

Figure 3.12: Throughput versus number of processors (varying from 2 to 8): Event size = 10 tracks
Figure 3.13: Throughput versus number of processors (varying from 2 to 8): Event size = 100 tracks

Figure 3.14: Throughput versus number of processors (varying from 2 to 8): Event size = 400 tracks
Figure 3.15: Throughput versus number of analyzers, for the same series of events, run with both one- and two-way data transfer
Chapter 4

The STAR Level 3 Trigger

While the level 0-2 triggers in STAR filter events based solely on the data from the fast detectors, the final decision may include input from the TPC.

Electrons in the TPC achieve a drift velocity of 5 cm/µs, giving them a drift time of 40 µs. The registered pulse is, of course, an analog signal, and the endcap electronics take on the order of 100 µs to convert it to digital form. Therefore the overall TPC response time is orders of magnitude slower than that of the so called 'fast' detectors, so that the TPC is unable to participate in level 0-2 trigger decisions.

The STAR Level 3 trigger takes in data from the TPC and reduces the data flow to a level that can be handled by the offline storage. (Incoming data is about 7 Mb/event at 100Hz, or about 700Mb/s, which needs to be reduced to about 16Mb/s). The data is then stored for offline processing.

The main mechanism by which this is accomplished is a series of pattern-
recognition stages applied incrementally to the TPC data, followed by algorithms which use the processed data to output a decision.

4.1 Hardware Architecture

The layout of the level 3 hardware is shown in figure 4.1. Data enters the system from the TPC by way of the VME\(^1\) crates. Each crate contains 36 Intel i960\(^2\) processors, which run cluster finding algorithms on the incoming data.

The cluster data is then shipped on to the Sector Level 3 (SL3) system. This consists of a farm of 12 466MHz Alphas, connected via a Myrinet network. (Myrinet is a proprietary, high-performance network with a throughput of around 140 Mb/s). The SL3s run the track finder on the cluster data, sending the track data on to the Global Level 3 (GL3).

The GL3 is a 600 MHz Intel Pentium III, connected to the SL3 Myrinet network. The GL3 is what makes the final event decision. It collects track data from the SL3s, reconstructs the event, and runs a series of algorithms on it, before deciding whether to keep the event or not. This decision is passed on to the Event Builder, accompanied, if the decision was 'yes', by the event data to be stored on tape.

\(^1\)http://www.vita.com/vmefaq/index.html

\(^2\)http://developer.intel.com/design/i960/
Figure 4.1: STAR Level 3 Trigger Hardware Setup
4.2 Software Architecture

The main logical components of the Data Acquisition system (DAQ) and Level 3 are shown in Figure 4.2.

The entry point of data into the level 3 trigger is the Detector Broker (DET), which forms the interface between the i960 cluster finders and the sector level 3 farm. The data is grouped into supersectors.

The SL3s take in cluster data for a supersector, run the track finding algorithm, and send the list of tracks onward to GL3. It optionally sends cluster data as well, which GL3 doesn't use, but can in turn ship downstream.

The GL3 takes in sector track data from the SL3s, and recombines them into a global list of tracks, merging tracks across sector boundaries, and makes a decision based on the tracked event, determining whether the event will be written to tape.

The Event Builder (EVB) is the piece of DAQ that collects the different pieces of data corresponding to the same event and places them on contiguous memory.

The Global Broker (GB) forms the interface between the GL3 and the Event Builder (which eventually stores the event). Once GL3 has formed an event decision, it passes it on to GB, which in turn sends a 'Build Event' directive to EVB and passes on the event data.

The Token Manager (TM) is another DAQ component, that synchronizes the flow of tokens through DAQ and L3. Given the chain of DAQ subsystems
TAP: Tape
EVB: Event Builder
TM: Token Manager
DET: Detector Broker
SL3: Sector Level 3
GL3: Global Level 3
GB: Global Broker

Figure 4.2: Level 3/DAQ Software Setup
through which an event passes, some method of synchronization is needed (so that, for instance, GL3 does not request an event that SL3 has not yet received). This synchronization is provided by means of tokens.

A token is essentially an event ID; an integer between 0 and 4095 that identifies the event and is attached to all messages involving that event. Of course, the token number is nonunique, since it repeats every 4096 events; we require, then, that no two events in the system at any given time have the same token number.

4.2.1 Event Display

The event display allows us to view events from L3 in real time. It is connected to the SL3/GL3 Myrinet network, and can request track data from the GL3.

4.3 Data Processing

Apart from data organization and transfer, the level 3 trigger transforms the incoming raw data into information about the particles produced in the collision and their trajectories. The two main tasks involved are cluster finding and track finding.
4.3.1 Cluster Finder

The TPC collects data in the form of 'hits', or electrical signals in the anode pad plane. For each padrow, we define a plane whose axes are along the padrow and along z. Within this plane, hits are represented as a combination of a pad number and a time bin (z), leading to a series of two dimensional regions.

Preliminary processing is provided by the application specific integrated circuits (ASICs), which perform zero-suppression and pedestal-subtraction on the hit data and provide a set of pointers to all above-threshold hits for each pad.

A cluster is defined as a contiguous group of hits. The cluster finding algorithms examine sequences of hits along a pad and in neighbouring pads, and attempt to group them into clusters.

The cluster finder runs on a set of Intel i960 CPUs, which take in TPC raw data from the ASICs and sends clusters on to the Detector Broker (DET).

4.3.2 Track Finder

The track finder takes in data from the cluster finder as a set of space points, and attempts to fit helical tracks through the points. It makes use of a conformal mapping, whereby the helices are converted (with a possible assumption about the vertex) to lines. The advantage here is that lines are easier to fit than helices, allowing for a faster algorithm.
Tracking is done at the SL3 stage, and track data passed in to GL3. The level 3 tracker is based on a fast track finding algorithm, using a conformal mapping and a follow-your-nose procedure.

4.4 Sector Level 3 Implementation

One of the L3 components I worked on was the Sector Level 3. To help with the SL3 implementation and testing, I also implemented a Sector Broker (SB), a simplified version of the STAR Detector Broker. The SL3 was based on existing code written for Windows NT; however, porting it proved to be unfeasible, and it was decided to rewrite it from scratch under Linux.

The SB and SL3 are fairly similar in their design. They are both implemented as state machines, with the states corresponding to various stages of processing and shipping of sector data. Transitions between states are event driven, and triggered by various messages between the SL3 and the SB. All actual processing and data transfer is done in a set of independent routines that are called by various state transitions.

The main SL3 and SB framework was written using an object oriented design in C++, with the various helper functions like the tracker and the communicator calling other, external pieces of code.

4.4.1 SL3 State Machine

The SL3 State Machine is shown in figure 4.3.
The states are

- **Idle**: This is the default state. Any state can return to Idle upon a 'cancel processing' directive from the SB

- **Receiving Clusters**: SL3 switches into this state when it is being sent cluster data by SB

- **Processing**: This is the state in which SL3 is actually tracking the cluster data for the current sector

- **Summary Ready**: An intermediate state in which SL3 has finished generating a 'summary' (the track data returned by the track finder), but has not yet been asked by SB to send it on

- **Sending Summary**: Here, SL3 has received a 'send summary' request from SB and is actually sending the track data

The state transitions, as mentioned, are driven by messages from the SB:

- **SB_SL3_SEND_STATUS**: SB wants the status of this CPU. Does not trigger a state transition, just sends a status message to SB

- **SB_SL3_ANNCE_CLUST**: SB announces new clusters ready to be sent. SL3 shifts into 'Receiving Clusters' and calls the 'get data' routine.
- **SB_SL3_START_TRACK**: Clusters have been sent and acknowledged; start tracking them. SL3 shifts into 'Processing' and calls the track finder.

- **SB_SL3_SEND_SUMM**: Send the finished summary to SB

- **SB_SUMMARY_REC**: SB acknowledged the summary; discard the event and return to 'Idle'

- **SB_SL3_RELEASE_TOKEN**: Cancel the current event and return to 'Idle'

### 4.4.2 Thread-based Implementation

Since the SL3 has to respond to 'Send Status' and 'Release Token' requests at any time, we cannot afford to have the various functions called by the state transitions block the SL3 until they complete. The SL3 has therefore been implemented using a multithreaded model, in which the various functions run in separate 'threads' (processes that Linux can execute in parallel with each other). The message queue and state machine run in the main thread, and functions like tracking and data transfer are launched in separate threads. One additional detail here is that messages and data are handled independently by two different pieces of code.

The additional threads are the tracker, and two communication threads to get data from the SB and send a summary back. The threads are launched
Figure 4.3: SL3 State Machine
from the state machine, and report back to it when they finish. This ensures that state transitions are not completed prematurely, but that the SL3 is still able to respond to the SB.

Figure 4.4 shows diagrammatically how the various threads are created when required, run in parallel with the main thread, and are then destroyed when they have completed their tasks. Terminating threads send messages to the SL3 message queue, driving the 'internal' transitions shown in the state machine diagram (figure 4.3).

4.4.3 Sector Broker Implementation

The Sector Broker is a very simplified version of the STAR Detector Broker, which reads cluster data from a file, and passes it to SL3. Like the SL3, it is implemented as a state machine (Figure 4.5), with the various transitions being driven by messages from SL3. The SB is capable of driving a number of SL3s; it assigns a separate state machine to each SL3. The two state machines are closely coupled, so that they advance linearly through their main cycle via an alternating sequence of messages.

At a global level, the SB maintains a queue of incoming events and a pool of available SL3 machines. Events are first broken into sectors (or, more accurately, supersectors, each consisting of two sectors). The SB then looks for free SL3 machines to track the sectors. If it sees an SL3 machine marked as Idle, it sends it a sector from the current event; otherwise it waits for a machine to become free.
The SB also monitors each SL3 and checks that it is not taking too long to process a sector; if it runs over a predefined 'timeout' limit, a 'Cancel' message is sent. When all the sectors of the current event have been tracked and returned, the track data is written out and the next event dequeued.

Both the SL3 and the SB run in an infinite loop, waiting idly when there is no event to process, rather than shutting down.

4.4.4 Communications and Networking

Communications were handled by two main classes, one for general data transfer and a more specialised derived class that handled messages. All details of the underlying network were hidden, so that the code could easily be moved to different hardware like SCI\(^3\), Myrinet and ethernet with very little change.

This was useful when it was decided to move away from SCI, and again when a modified version of the SL3/SB setup was used to implement the prototype event farm.

4.5 Global Level 3 Implementation

While at Brookhaven National Lab, I worked upon designing and implementing the Global Level 3. In this section I shall go into a little more detail upon it.

\(^3\)Scalable Coherent Interface, http://208.179.47.35/sci.html
Figure 4.4: SL3 Threads

Figure 4.5: SB State Machine
GL3 divides into two main parts, communications and processing. The main program is event-driven; it communicates with the rest of the system (DAQ and SL3) via a single message queue. Data transfer is synchronised by way of the message queue, but actual data is sent directly between two machines.

Like SL3, GL3 was implemented using an object oriented approach, in C++ under Linux.

4.5.1 The Message Queue

The GL3 communicator is built around a threaded message queue class from the DAQ library. This class runs in the background, monitoring the myrinet card, and queueing all received messages.

The main GL3 communication routine is a loop that continuously polls the message queue for an incoming message. When a message is received, the appropriate handler is called.

The main classes of messages are:

- Token handling: These are messages from DAQ, announcing a new token, or removing one after it has finished being processed. GL3 in turn rebroadcasts the token to the SL3s, where they are synchronized with the token stream from the other side of the system.

- SL3 Communication: These are the messages by which the SL3s transfer the sector data on to GL3. An SL3 machine announces an incoming
sector, and the GL3 returns a pointer to the memory address the sector has to be placed at.

- Global data: When the GL3 has finished building the completed event and run the proposed algorithms on it, it sends various pieces of data back to DAQ, including an event decision, the list of tracks, and optionally a list of clusters from SL3.

### 4.5.2 The Event Pool

When a GL3 machine starts up, it allocates memory for a pool of events - this event pool stores sectors as they come in from the various SL3 machines, and assembles them into complete events.

Incoming 'accept sector' requests from an SL3 machine include the sector's token number; this allows us to implement the event pool straightforwardly as a list of event buffers. Each buffer is either empty or indexed with a token number, and an incoming sector is accepted either if there is a free buffer or if it is part of an event currently in the pool. When all twelve supersectors of a given event have been received, the finished event is passed on to the algorithms. Once a decision has been made, the event is either discarded or shipped out, and the buffer is freed.
4.5.3 GL3-SL3 interaction

The GL3 communicates with the various SL3 machines in two ways - via the global message queue, and directly with each machine for data transfers. Since the data transfers happen asynchronously, they needn't be taken into account separately, but can be included as part of the message passing.

Therefore, from the SL3s point of view, the GL3 represents a bottleneck, through which processed sectors must pass. The GL3 event pool can, as described, accept a sector if there is a free slot, or if it belongs to a partially assembled event; therefore, if the DET distributes incoming sectors equally across SL3s, no one machine should get disproportionately backed up.

4.5.4 Algorithms

After the GL3 has collected the tracked sectors and reassembled them into an event, various algorithms are run on the event in order to reach a final event decision. GL3 provides the algorithms with a track list and a framework for keeping histograms. When the algorithms have all run, a decision vector, consisting of the decisions from the various algorithms, is output, as well as a final decision on the event.
Chapter 5

Conclusions

Despite advances in hardware, the rates of data acquisition and storage are still orders of magnitude slower than the rate at which raw data is generated by particle detectors.

The trigger is, therefore, one of the key components in the detector setup, allowing the filtering, in real time, of interesting from uninteresting events, and reducing the input data rate to a level that the electronics can handle.

In this study, we have looked at various high-level trigger components, focusing on the problem of track finding and its attendant infrastructure. Also taken into account was the fact that triggers need to be both fast and cost-effective, and we have looked at a setup using off-the-shelf components in place of some of the higher-end hardware currently employed.
References


