A Computer based Pattern Recognition Trigger running at 1 MHz

D. Atanasov, I. Kisel*, V. Lindenstruth and G. Torralba

Kirchhoff-Institut für Physik, Ruprecht-Karls-Universität Heidelberg, Germany

(*) ikisel@kip.uni-heidelberg.de

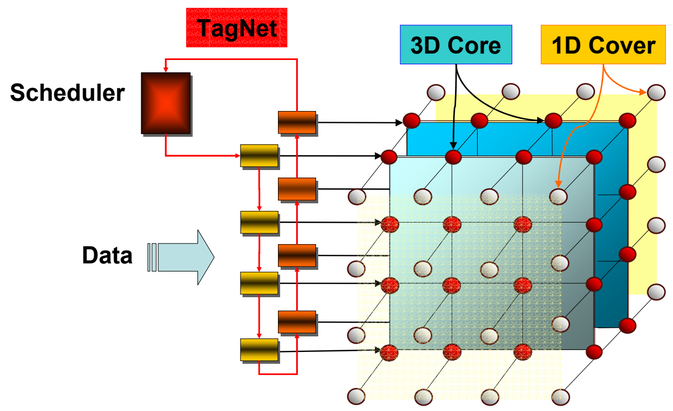
LHCb experiment will study CP violation and other rare phenomena in B-decays with a forward spectrometer at the LHC (CERN). The LHCb trigger has to efficiently select a few Hz of interesting B-decays from a non-elastic pp interaction cross-section of 80 mb, which corresponds to 16 MHz of interactions at the preferred LHCb luminosity. The first trigger level reduces the rate to 1 MHz using large E*t* triggers. At 1 MHz all data is digitised, and a subset of tracking information is used to reduce the rate to 40 kHz, at which rate the full event building is performed. Having access to all detector information the rate is subsequently reduced to around 200 Hz and written to storage.

The Level-1 trigger of the LHCb experiment is a hardware/software based system built around standard components whenever possible. The trigger is the second stage in the LHCb trigger pipeline having an average input of 1 MHz and a bandwidth requirement of more than 4 GByte/s. The input data is initially split amongst several input feeds with sub-events being as small as 128 Byte. Data have to be sent to a compute node which runs a track finding algorithm and produces a result message at a MHz rate. The system uses a high-speed network available off-the-shelf which connects commodity PCs. The interface to the NIC is PCI.

Based on the Scalable Coherent Interface (SCI), tests concerning speed, throughput, latency, and scalability are presented. Based on the approximate baseline system size a basic timing analysis of the system is given. The system is characterized by multiple nodes sending to one single receiver. Therefore, the Level-1 trigger is prone to network congestion since the receiving node can not handle the aggregate input data rate. However, a hardware based data scheduling mechanism, the TagNet, is introduced which avoids congestion in the system.

A 30 node prototype is presented which has been built around Linux PCs connected by an SCI network. The system is able to process data with a MHz rate. Sub-events have been chosen to be as small as 128 Byte. Data transfer has been scheduled by a basic implementation of the TagNet. The system has been used to prove basic functionality and to measure important input parameters concerning the system.

3D Torus Topology



- TagNet – schedule and send small data packets
- Core network – distribute data to the target compute nodes
- Cover network – increase number of compute nodes

- RU
- CN

The Level-1 trigger system is based on PCs connected by Scalable Coherent Interface (SCI) network cards in a 3D torus topology. The input data stream is initially scattered between several Readout Units (RU), which have to send data to a specific node in the network. On reception a software algorithm performs data analysis and sends a result message to the Level-1 decision unit. A maximum latency is set on the overall data path through the Level-1.

To avoid network congestion at the receiver, the data are orchestrated by the TagNet. The TagNet is a scheduling network which provides a flexible congestion control mechanism. The TagNet is managed by the TagNet Scheduler that keeps track of events currently processed and CPUs which are available to accept new data. The Scheduler continuously monitors the network and communicates with CNs at low latency that allows to send data from some detectors on CNs demand thus avoiding not necessary increase of data throughput.

The network consists of two parts: a network core, which is able to accept and distribute data, and a network cover, which allows to increase CPU power to the amount determined by the processing time per event. The core network is build on 3D SCI cards, while for the cover network only 1D SCI cards are needed.

The Prototype in Heidelberg

A prototype of the Level-1 vertex trigger system has been implemented. The system is capable to send small messages with a rate of more than 1 MHz as required by the LHCb experiment. However, the TagNet version implemented is very basic and only allows static assignment of compute nodes. The system comprises three RUs, 26 CNs, and the Level-1 decision unit interface which amounts to 30 nodes total. Mockup data has been analyzed in the receiving CN. A packet loss has never been detected. However, since the system is based on a standard Linux distribution the analyzing process might be suspended such that events could be missed. A frequent occurring reason are interrupts. However, an overall system analysis shows that almost 100% of the events are analyzed.

There are two important parameters that go directly into the system architecture - the maximum link bandwidth and the maximum bandwidth that the B-Link can handle. B-Link has been determined to be 432 MByte/s (75% of maximum SCI net bandwidth) for a system that does not use displaced RUs. For a system using displaced RUs B-Link increases at least to 478 MByte/s which is 83% of the maximum SCI net bandwidth. The maximum B-Link bandwidth has been determined to be 450 MByte/s (88% of maximum B-Link net bandwidth).

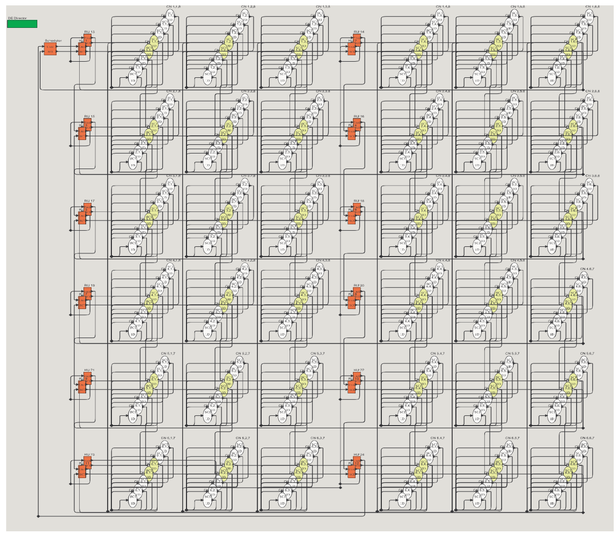
3D Torus (6x6x8) Simulation

The Ptolemy II simulation package works with two types of input files: xml files determining structure of the system and its elements with their interconnections, and java files implementing functionality of the system elements.

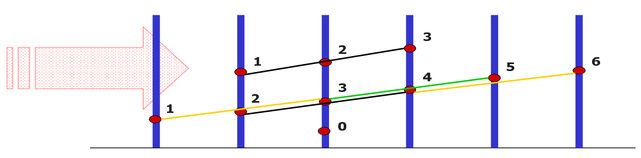
A file *LHCbTorus3D.c* has been written to generate Ptolemy xml files with description of the architecture model. It has as input parameters number of rows, columns and compute nodes thus producing a system of any size and form - from a single ring with a few compute nodes to 3D torus with up to thousand compute nodes. In this way one can easily check scaling of the system at any stage of the system architecture development.

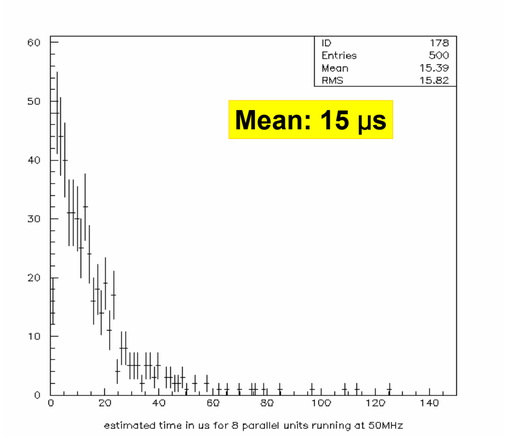
The functionality of the main system elements is described in *LHCbScheduler.java*, *LHCbRU.java*, *LHCbSCI3D.java* and *LHCbSCI1D.java* files.

Parameters of the system measured on the prototype have been used in the simulations.



Reconstruction Algorithm running on FPGA





The algorithms of event reconstruction in the VELO detector - searching for tracks in (R,Z) projection, reconstruction of the primary vertex using 2D tracks, reconstruction of 3D tracks and fit of secondary vertices - aim to find events with displaced secondary vertices that can indicate the b-hadron decays with significant transverse momentum due to the high b-quark mass and long lifetime.

The algorithms have been implemented on CPU and have reconstruction efficiency of ~97% for daughter tracks from B decays and primary vertex (x,y,z) resolution of (17, 19, 46) microns.

About 75% of total reconstruction time is taken by the 2D tracking as the most combinatorial part of the algorithm. Therefore hardware implementation of 2D tracking was investigated using FPGA. There were 8 processing units programmed in FPGA (one unit per VELO sector) running in parallel. The FPGA uses the same algorithm as CPU runs, but simplified with respect to FPGA features. Based on the cellular automaton approach it makes full triplet search and introduces relations between triplets to make ease their gathering into full tracks by CPU. At the same time it filters data suppressing detector noise and hits from tracks out of geometrical acceptance.

As the input data we use data from L0DU, VELO, TT and T1-3 detectors. The data, based on the winter production, contains about 2000 Level-0-yes minimum bias events. Mean amount of data per event is about 6.6 kB with tail up to 20 kB. The time it takes to execute algorithms is 500 microseconds and has being adapted to the size of the farm we can afford during start-up. For monitoring an overhead of the framework we allow 50 microseconds. The system is able to send data in two steps: L0DU, VELO and TT data (about 3.5 Kbyte) for all events and T1-3 data for up to 40% of events on CNs demand. With these assumptions the total data rate is ~4 GByte/s. The number of RUs needed is 24 with 2 RUs per row. For each of the 550 microseconds average computing time we need a CPU. Assuming dual nodes the system has to have at least 275 CNs. A core of the system is organized in six rows and six columns with two layers thus giving 72 CNs. The rest of CNs will be placed in the cover part of the network.

The event building latency for L0DU, VELO and TT data is about 25 microseconds and has narrow structure reflecting compact feature of the 3D torus. Full event building latency has main peak as for VELO and TT data, while the tale of the distribution shows T1-3 data for about 5% of events sent on CNs demand. As amount of VELO+TT data is comparable to T1-3 data one can see twice larger latency for long events - up to 75 microseconds. As a result of 3D torus topology of the system the event building latency is stable with respect to the system size up to maximum size of 1200 CPUs.

Data throughput of the network has been measured on the most overloaded CN which is in the row of the Scheduler. Here there is additional packages stream from CNs to the Scheduler. The highest load is in X direction but well within the SCI capability. The lowest data throughput is in Z direction that offers possibility either to add more compute nodes in the cover network or to share data between compute nodes in the same Z ring.

During all tests the system has shown stable operation without data congestion or buffers overflow. This is a result of data scheduling via the TagNet and ability of the SCI network to buffer data within the network while transferring.