Benchmarks and Implementation of the ALICE High Level Trigger

T. Alt, H. Appelshäuser, S. Bablok, B. Becker, S. Chattopadhyay, C. Cheshkov, C. Cicaló, J. Cleymans, R. W. Fearick, H. Helstrup, V. Lindenstruth, C. Loizides, M. Richter, D. Röhrich, B. Skaali, F. Staley, T. Steinbeck, A. Szostak, H. Tilsner, K. Ullaland, G. de Vaux, A. Vestbø, T. Vik, Z. Z. Vilakazi, A. Wiebalck, and G. Øvrebekk for the ALICE collaboration

***Abstract*—The ALICE High Level Trigger combines and processes the full information from all major detectors in a large computer cluster. Data rate reduction is achieved by reducing the event rate by selecting interesting events (software trigger) and by reducing the event size by selecting sub-events and by advanced data compression. Reconstruction chains for the barrel detectors and the forward muon spectrometer have been benchmarked. The HLT receives a replica of the raw data via the standard ALICE DDL link into a custom PCI receiver card (HLT-RORC). These boards also provide a FPGA co-processor for data-intensive tasks of pattern recognition. Some of the pattern recognition algorithms (cluster finder, Hough transformation) have been re-designed in VHDL to be executed in the Virtex-4 FPGA on the HLT-RORC. HLT prototypes were operated during the beam tests of the TPC and TRD detectors. The input and output interfaces to DAQ and the data flow inside of HLT were successfully tested. A full-scale prototype of the dimuon-HLT achieved the expected data flow performance. This system was finally embedded in a GRID-like system of several distributed clusters demonstrating the scalability and fault-tolerance of the HLT.**

***Index Terms*—Data acquisition, data processing, distributed computing, real time systems, software fault tolerance, triggering.**

I. Introduction

THE ALICE experiment at the LHC will investigate Pb-Pb collisions at a center of mass energy of about 5.5 TeV per nucleon pair and p-p collisions at 14 TeV. The detectors are optimized for charged particle multiplicities of up to $\mathrm{d}N_{ch}/\mathrm{d}\eta$ of 8000 in the central rapidity region [1]. The hardware trigger in ALICE is organized into three different levels, L0, L1 and L2, which have different latencies. Their main tasks are to detect interactions, to select events according to their multiplicity and to provide past-future pile-up protected clean events.

The main central tracking detector, the Time Projection Chamber (TPC), is read out by about 600 000 channels, producing a data size of up to 75 MB per event for central Pb-Pb (most extreme scenario). The overall event rate is limited by the foreseen bandwidth to permanent storage of 1.25 GB/s. With no further reduction, the ALICE TPC can only accumulate central Pb-Pb events up to 20 Hz. Higher event rates are possible (up to 200 Hz) by either online event selection and/or data compression. Both applications require a real-time analysis of the detector information. To accomplish the pattern recognition tasks at an incoming data rate of 10–20 GB/s, a massive parallel computing system, the High Level Trigger (HLT) system, is under construction [2].

II. Data Flow and Architecture

The High Level Trigger combines and processes the full information from all major detectors in a large computer cluster. A farm of clustered SMP-nodes (about 400 nodes), based on off-the-shelf PCs and connected by a high-bandwidth, low overhead network, provides the necessary computing power for event reconstruction. The HLT farm is designed to be completely fault-tolerant avoiding all single points of failure. Based on the publisher subscriber principle, a generic communication framework has been developed, which allows the construction of any hierarchy of communication processing elements.

Fig. 1 shows a sketch of the architecture of the system adapted to the anticipated data flow from the ALICE detectors. The TPC consists of 36 sectors, each sector being divided into six subsectors. Data from each sub-sector is transferred via the DDL (optical fibers equipped with source and destination interfaces) from the detector front-end into the ReadOut Receiver Cards of the DAQ system (D-RORC), from where a copy is sent to the HLT-RORC. These are interfaced to the receiving nodes through their internal PCI-bus. The HLT-RORC provides—in addition to the data transfer functionality—a FPGA co-processor for the data intensive local tasks of the pattern recognition and enough external memory to store several dozen event fractions.

The overall architecture of the system is driven by the inherent readout granularity and the requirement for a complete event reconstruction and trigger decision. The internal topology will have a tree-like structure, where the result from the processing on one layer (e.g., track segments on sector level) will

Manuscript received June 19, 2005; revised December 23, 2005.

T. Alt, V. Lindenstruth, T. Steinbeck, H. Tilsner, and A. Wiebalck are with the Kirchhoff Institute of Physics, University of Heidelberg, D-69120 Heidelberg, Germany.

H. Appelshäuser and C. Loizides are with the Institute for Nuclear Physics, University of Frankfurt, D-60438 Frankfurt, Germany.

S. Bablok, M. Richter, D. Röhrich, K. Ullaland, A. Vestbø, and G. Øvrebekk are with the Department of Physics and Technology, University of Bergen, Norway.

B. Becker, J. Cleymans, R.W. Fearick, A. Szostak, G. de Vaux, and Z.Z. Vilakazi are with the Department of Physics, University of Cape Town, South Africa.

S. Chattopadhyay is with the SAHA Institute of Nuclear Physics, Kolkata, India.

C. Cheshkov is with CERN, Geneva, Switzerland.

C. Cicaló is with INFN-Cagliari, Italy.

H. Helstrup is with the Faculty of Engineering, Bergen University College, Norway.

B. Skaali and T. Vik are with the Department of Physics, University of Oslo, Norway.

F. Staley is with DAPNIA/SPhN, CEA-Saclay, F-91191 Gif-sur-Yvette, France.

Digital Object Identifier 10.1109/TNS.2006.873770

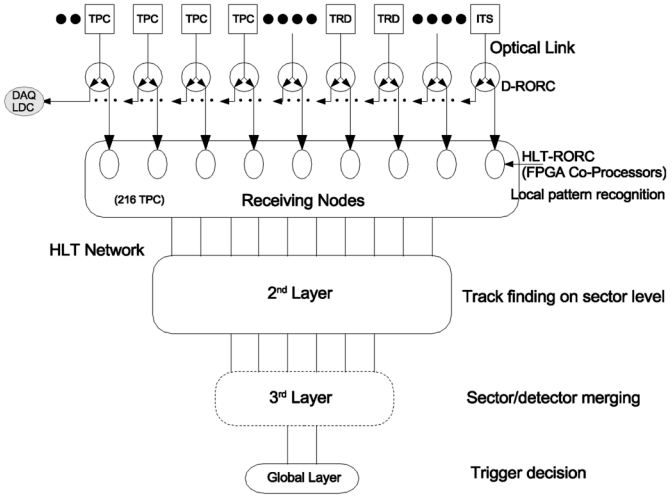

Fig. 1. Data-flow architecture of the HLT system. The detector raw-data is duplicated and received by both the DAQ and HLT system.

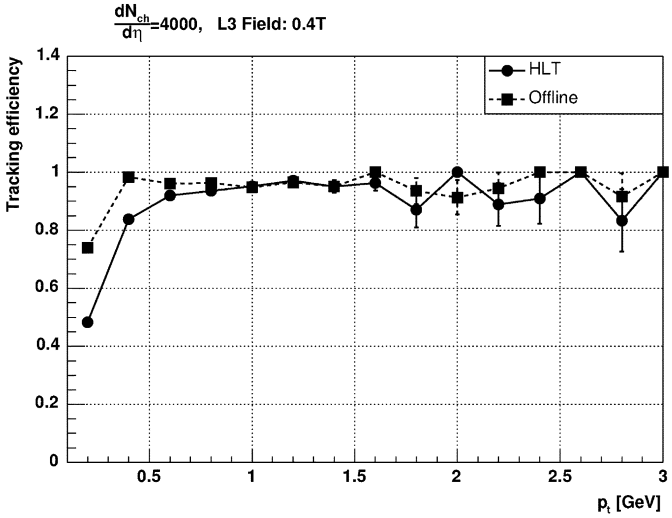

Fig. 2. Performance of the HLT tracking algorithms compared to the offline reconstruction chain: CF and TF efficiency as a function of the transverse momentum for a pseudorapidity density of 4000.

be merged at a higher layer (sector merging and track fitting). Finally all local results will be collected from the sub-detectors and combined on a global level where the complete event can be reconstructed and trigger decisions can be issued. Both the trigger decision and the event summary data as well as the modified/compressed raw data (optionally) will be transfered to the DAQ via DDL-links.

III. Online Pattern Recognition

The main processing task is to reconstruct the tracks in the TPC, and in a final stage combine the tracking information from all detectors. Given the uncertainties of the anticipated particle multiplicities, different approaches are being considered for the TPC track reconstruction.

The conventional approach of TPC track reconstruction consists of a Cluster Finder and a subsequent Track Follower. In a first step the Cluster Finder reconstructs the cluster centroids from the generated two-dimensional charge distributions in the TPC pad-row planes. Together with the position of the pad-row-planes the centroids are interpreted as three-dimensional space points along the particle trajectories, and serve as an input for the Track Follower which connects the space points into track segments. A final helix-fit of the track segments provides the track parameters and thus the kinematic properties of the particles.

Such an approach has been implemented and evaluated on simulated ALICE TPC data [3]. The algorithms were originally developed for the STAR L3 trigger [4] and consist of a straightforward center-of-gravity calculation of cluster centroids, and a Track Follower which applies conformal mapping on the space points. The latter enables the circular tracks to be fitted by a linear parametrization, thereby significantly reducing the computational requirements. The overall measured performance of the reconstruction chain represented by the tracking efficiency as a function of the transverse momentum is shown in Fig. 2.

The tracking efficiency for $\mathrm{dN_{ch}}/\mathrm{d}\eta \leq 4000$ is similar to that achieved by the standard offline reconstruction chain. The algorithm is relatively fast, and is therefore well suited for the lower multiplicity regime. For higher multiplicities the observed tracking performance deteriorates. This is due to the increasing detector occupancy which gives rise to a significant amount of overlapping clusters. In such a scenario the Cluster Finder fails to reconstruct the cluster centroids due to its incapability to deconvolute overlapping charge distributions. Information about the tracks is needed *prior* to reconstructing the cluster centroids in order to be able to fit the individual distributions to a known shape. This can be done because the cluster shape depends mainly on the track parameters. Together with the knowledge of the number of tracks contributing to a given cluster, the deconvolution can be done based on a two-dimensional Gauss-fit. Such an approach has been evaluated by applying an implementation of the Hough Transform on the raw ADC-data, and subsequently fitting the clusters to a two-dimensional Gauss-function based on the found track candidates. However, too many candidates are produced by this gray-scale Hough Transform which result in too many fake tracks. A better approach is a counting Hough Transform [5]. The fact that the TPC is a continuous tracking device is taken into account and therefore all padrows contribute to a good track. Large gaps indicate fake candidates and parameter space bins containing gaps are removed from the filling procedure. In addition, the parameter space is linearized using a conformal mapping. Both methods speed up the transformation and result in a simple peak structure in the parameter space. The obtained tracking efficiency as a function of track transverse momentum is shown in Fig. 3. The efficiency is better than 95% for $p_T \geq 0.7\,\mathrm{GeV/c}$ and does not depend on the event multiplicity. The abundance of fake track candidates is less than 5%.

The overall computing time needed for the TPC tracking for different multiplicities is shown in Table I. The reference platform was an Intel Pentium 4 (2.8 resp. 3 GHz) which corresponds to a performance rating of approx. 1k SPECint. The CF + TF approach produces track parameters as well as space points for refitting and dE/dx analysis, while the fast Hough transform just results in track parameters. Assuming a multiplicity of

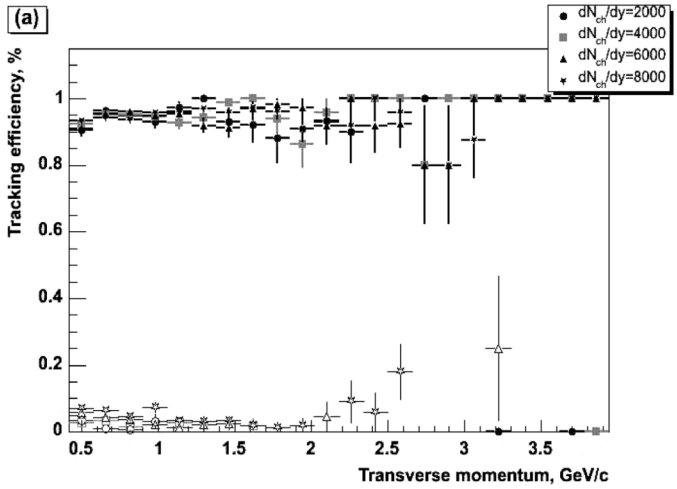

Fig. 3. Performance of the HLT tracking algorithms: fast Hough transform efficiency and fake track abundance as a function of the transverse momentum for different pseudorapidity densities (from 2000 to 8000).

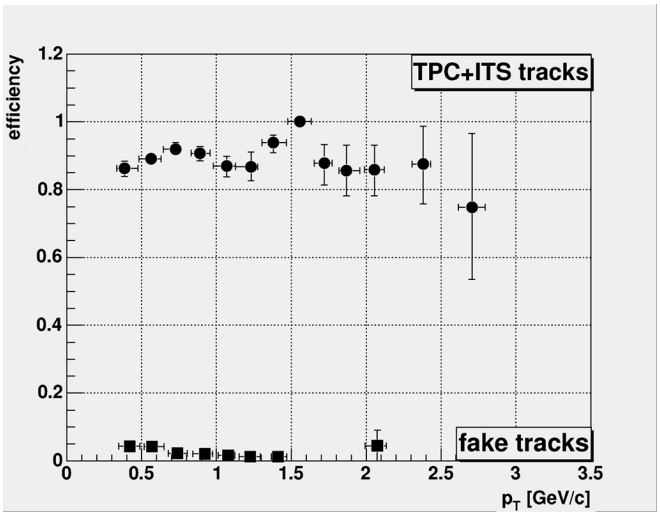

Fig. 4. Performance of the HLT ITS tracking: Combined TPC-ITS tracking efficiency and fake track abundance a function of the transverse momentum for a pseudorapidity density of 4000.

TABLE I
TIMING MEASUREMENTS OF THE TPC TRACKING CODES. THE BENCHMARKS WERE PERFORMED ON A 1 k SPECint MACHINE (INTEL PENTIUM4, 2.8 GHz)

$dN_{ch}/d\eta$	0	2000	4000	6000	8000
CF+TF [sec]		6.3	13.2	21.2	
Fast HT [sec]	0.8	3.5	5.8	8.7	11.8

$dN_{ch}/d\eta = 2000$–4000, as predicted by many models based on RHIC results, a farm of about 1000 CPUs would suffice to solve the pattern recognition task within the time budget of about 5 ms. The computing power scales approximately linearly with the number of CPUs because the pattern recognition in the TPC is local and hierachic, e.g. tracking is done on a sector level (36 sectors per TPC) and cluster finding on a padrow level. There is little communication between the pattern recognition processes for the different sub-detecetors. Latency is not a significant issue due to the availability of 64 bit systems and current memory prices. The PCs receiving the data from the detector, both in the HLT as well as for DAQ can therefore be equipped with enough buffer memory to accomodate large amounts of data. As an example we assume a DAQ receiver PC equipped with one DDL receiving data from the TPC and 8 GB of event buffer memory. Assuming an event fragment size per DDL of 350 kB, corresponding to a central event, about 23 000 events can be stored in the PC's buffers. At an event rate of 200 Hz this corresponds to about 2 min of data taking, which should be enough to compensate for variations in processing.

IV. ITS TRACKING AND D^0 TRIGGER

The tracks found in the TPC are followed into the Inner Tracking System (ITS). The offline code was used for the processing of the ITS data and the tracking [6]. The efficiency of the combined tracking (Fig. 4) is slightly lower than for the TPC only (Fig. 3). The impact parameter resolution is dominated by the resolution of the innermost layer of the silicon pixel detector. A transverse resolution of 60 microns has been achieved—comparable to offline results. Based on this impact parameter resolution, track candidates stemming from a secondary vertex can be selected. The D^0 finder used here is the offline code processing HLT-tracks. The invariant mass resolution (Fig. 5) is $35 \pm 5\,\mathrm{MeV/c^2}$—about 2–3 times larger than the offline result. The rate of background events can be reduced by a factor of 20.

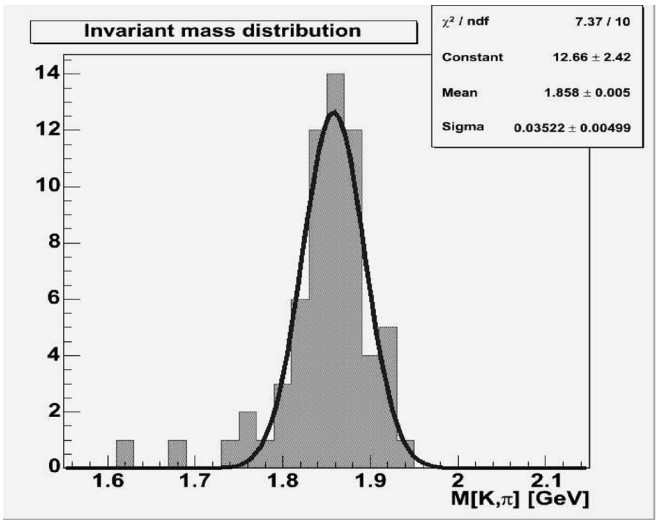

Fig. 5. Invariant mass distribution of open charm candidates. The mass resolution is $35 \pm 5\,\mathrm{MeV/c^2}$ (about 2–3 times larger than the offline result).

The computing time needed by the ITS processing and tracking and the D^0 finder for different multiplicities is shown in Table II. The reference platform was a 1.3k SPECint machine. Only the silicon pixel and silicon strip detectors were included in the HLT processing. The processing is fast, both for the ITS part and the open charm trigger.

V. I/O INTERFACE TO DAQ

The HLT system interfaces to the DAQ via the DDL. The detector data is split on the D-RORC and a copy is sent to the DIU on the HLT-RORC. The HLT system ships the trigger decision,

TABLE II
TIMING MEASUREMENTS OF THE ITS TRACKING CODE AND THE D^0 TRIGGER. THE BENCHMARKS WERE PERFORMED ON A 1.3 kSPECint MACHINE

$dN_{ch}/d\eta$	2000	4000	6000	8000
ITS tracker				
Clusterer [sec]	0.53	0.61	0.70	0.79
Vertexer [sec]	0.04	0.08	0.13	0.18
Tracking [sec]	0.26	0.54	0.90	1.38
D^0 finder [msec]	10	30	90	160

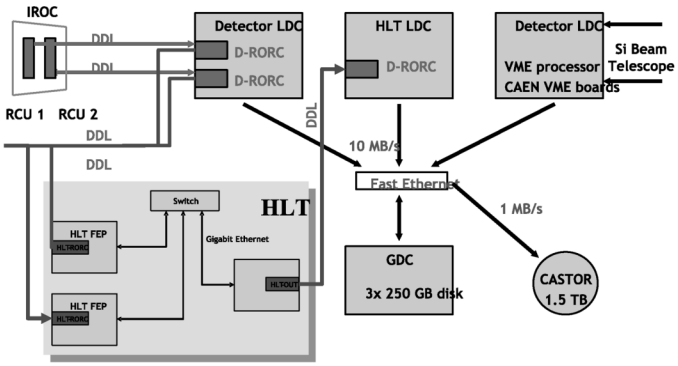

Fig. 6. Common DAQ and HLT setup for the TPC beam test at the PS.

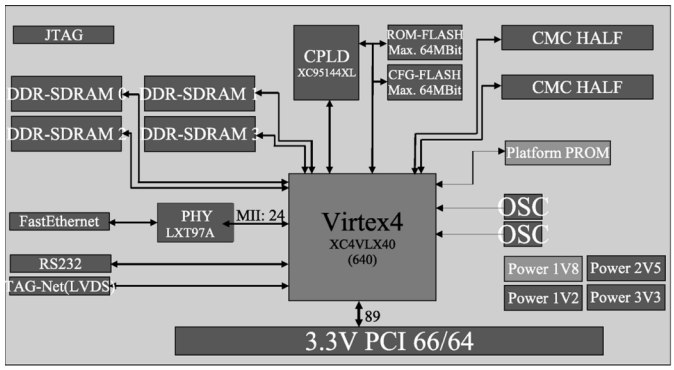

Fig. 7. Final design of the HLT-RORC.

modified and compressed as well as additional data (e.g., event summary) back to DAQ via DDLs. The data flow into and out of the HLT has been successfully tested in the TPC test beam setup at the PS (see Fig. 6).

VI. IMPLEMENTATION

The components of the HLT system are a farm of clustered SMP-nodes, custom PCI receiver cards which receive a replica of the raw data via the standard ALICE DDL link and which also provide a FPGA co-processor for data-intensive tasks of the pattern recognition and a generic communication framework based on the publisher subscriber principle, which allows the construction of any hierarchy of communication processing elements and guarantees fault-tolerance.

A. FPGA Co-Processor

The final design of the HLT-RORC is shown in Fig. 7. Some of the pattern recognition algorithms (cluster finder) have been re-designed in VHDL to be executed in the Virtex-4 FPGA, simulated, synthesized and then benchmarked in hardware. Currently the fast Hough transform is being implemented in VHDL.

B. Data Transport Framework

The design of the framework used to construct the data flow inside the HLT cluster is based on the publisher-subscriber paradigm in which subscribers inform a publisher of their interest in the data offered [7]. From this point on the publisher will broadcast new events that become available to its registered subscribers. In the design of this interface, particular emphasis is placed on efficiency, flexibility and fault tolerance. Efficiency is required for the framework as the need for CPU power for the analysis of the event data will be very significant. CPU resources should therefore only be used as little as necessary for the transport of data, to keep as much CPU time as possible for processing. This is achieved in the framework by not transporting actual data between the framework's components. Instead, data is placed into a shared memory segment by its publishing object and descriptors of that data are transmitted to the subscribers via named pipes. These communication paradigms are necessary because the segmentation of processing components into separate processes has the advantage of isolating faults for each respective component. When all subscribers have informed the publisher that they have finished processing an event, it is released and the shared memory can be re-used. The primary mechanism for providing flexibility is the separation of the framework into components (dataflow, data processing and data sink components) which can be connected in different configurations and any processing hierarchy can then be constructed. As the publisher-subscriber supports dynamic connections and disconnections at runtime, the system configuration can be adapted while it is active. This dynamic reconfiguration is also one of the major features supporting fault-tolerance of a system built with this framework. It allows for the replacement of failed components during runtime and also for the addition and/or removal of components as required for the reaction to events occuring in a system. A second major building block for this important point is related to the bridge components connecting different nodes. These components also have the ability to establish connections dynamically at runtime, not only for re-establishing existing connections but also for new connections between nodes. Through this mechanism it becomes possible to isolate faulty nodes in the system and replace them with other, previously unused nodes.

One of the important challenges in the ALICE HLT will be the management of the large number of framework component processes distributed in the cluster. It has to be ensured that all processes are started and connected in the correct order. For this purpose a system, the TaskManager [8], has been developed to control and supervise the framework components.

C. GRID-Like HLT Configuration

In the ALICE High Level Trigger a GRID approach would in principle be feasible as it does not have any fixed latency requirements for its trigger decision. An upper bound for the latency is of course given by the combination of buffer sizes and

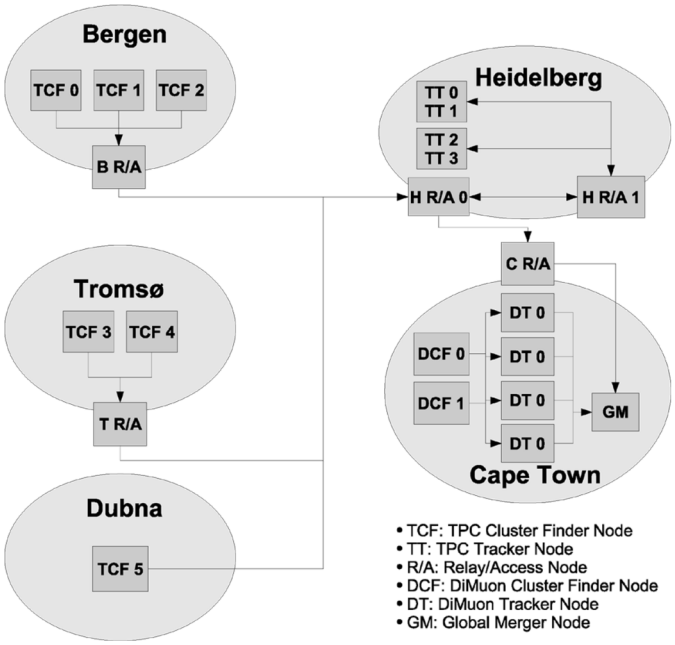

Fig. 8. The global GRID-like setup with all involved sites and nodes.

input data rate, as already calculated at the end of Section III. The buffering times determined there can be enhanced further simply by adding more memory for this purpose. It should thus be possible to compensate temporary increases in communication times. A globally distributed test of the HLT system was intended as a proof-of-principle demonstration and feasibility study of online grid-like systems. A global setup like this could be useful in enhancing the available online processing capability by distributing the processing, in analogy to what is foreseen for offline data processing. This additional processing capability could then be used to run more complex selection or compression algorithms or even some of the later reconstruction steps, where the input data is not so large anymore. Such a distribution would therefore complement the present scheme of employing fast local networks and FPGA co-processors in order to maximize CPU time available for online processing.

In order to create a full global north-south axis as well as some east-west expansion two further sites in Tromsø, Norway, and Dubna, Russia, have been included in the setup, in addition to the listed HLT collaboration institutes. For the test a configuration was chosen that mimics a part of the HLT processing, incorporating input from TPC and Dimuon detectors [9]. At three of the sites, Bergen, Tromsø, and Dubna, the components were set up to correspond to cluster finding on data from the TPC detector. Output data produced at these three sites was then sent to Heidelberg. Here it was merged together for TPC tracking. The output produced by these four components was then sent to Cape Town. In Cape Town, the mock-up TPC data was merged with mock-up Dimuon data generated by another processing chain. This chain simulated the processing of Dimuon detector data from cluster finding up to tracking. As the last step in the processing chain the tracked mock-up data was then merged with the received TPC data. In a real setup this component is the location where the trigger decision would be made and/or where the completely reconstructed event data is written to permanent storage. The full setup is shown in Fig. 8. This test ran unattended for more than 15 hours. During this time more than 500 000 events were passed through the mock-up processing chain. The event rate was of course limited by the network to about 10 Hz.

VII. Conclusion

The current TPC tracking performance shows that a sufficient event reconstruction within the central Pb-Pb event rate of 200 Hz will be achievable for multiplicity densities of $\mathrm{d}N_{ch}/\mathrm{d}\eta \leq 4000$. For higher densities cluster deconvolution based on track parameters becomes necessary. In this scenario the fast linearized Hough Transform has proven to be efficient and fast up to $\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta \leq 8000$ for transverse momenta larger than about 0.5 GeV/c. The ITS can be included in the HLT processing scheme with sufficient efficiency and moderate CPU requirements. A D^0 trigger is feasible. The final design of the custom HLT-RORC is under way as well as a VHDL implementation of the fast Hough transform. The I/O interface to DAQ has been successfully tested in the TPC test beam. It has been demonstrated that distributed grid-like online systems are feasible in principle, provided that the necessary network conditions regarding bandwidth are met.

References

[1] ALICE Collaboration. CERN/LHCC/1995-71.
[2] ALICE Collaboration. CERN/LHCC/2003-062.
[3] A. Vestbø *et al.* (ALICE Collaboration), "The ALICE high level trigger," *J. Phys.*, vol. G30, pp. s1097–s1100, 2004.
[4] C. Adler *et al.*, "The STAR Level-3 trigger system," *Nucl. Instrum. Meth.*, vol. A499, p. 778, 2003.
[5] C. Cheshkov, "Fast Hough transform track reconstruction for the ALICE TPC," *Nucl. Instrum. Meth.*, submitted for publication.
[6] The ITS code was cleaned up and speeded up by Jouri Belikov.
[7] T. Steinbeck *et al.*, "An object-oriented network-transparent data transportation framework," *IEEE Trans. Nucl. Sci.*, vol. 49, p. 455, 2002.
[8] T. Steinbeck *et al.*, "A control software for the ALICE high level trigger," in *Proc. Computing in High Energy Physics Conf. 2004 (CHEP04)*, 2004 [Online]. Available: http://chep2004.web.cern.ch/chep2004/
[9] T. Steinbeck *et al.*, "Real time global tests of the ALICE high level trigger data transport framework," in preparation.