An Overview of the ATLAS High-Level Trigger Dataflow and Supervision

J. T. Baines, C. P. Bee, A. Bogaerts, M. Bosman, D. Botterill, B. Caron, A. dos Anjos, F. Etienne, S. González, K. Karr, W. Li, C. Meessen, G. Merino, A. Negri, J. L. Pinfold, P. Pinto, Z. Qian, F. Touchard, P. Werner, S. Wheeler, F. J. Wickens, W. Wiedenmann, and G. Zobernig

***Abstract*—The ATLAS high-level trigger (HLT) system provides software-based event selection after the initial LVL1 hardware trigger. It is composed of two stages, the LVL2 trigger and the event filter (EF). The LVL2 trigger performs event selection with optimized algorithms using selected data guided by Region of Interest pointers provided by the LVL1 trigger. Those events selected by LVL2 are built into complete events, which are passed to the EF for a further stage of event selection and classification using off-line algorithms. Events surviving the EF selection are passed for off-line storage. The two stages of HLT are implemented on processor farms. The concept of distributing the selection process between LVL2 and EF is a key element in the architecture, which allows it to be flexible to changes (luminosity, detector knowledge, background conditions, etc.) Although there are some differences in the requirements between these subsystems there are many commonalities. An overview of the dataflow (event selection) and supervision (control, configuration, monitoring) activities in the HLT is given, highlighting where commonalities between the two subsystems can be exploited and indicating where requirements dictate that implementations differ. An HLT prototype system has been built at CERN. Functional testing is being carried out in order to validate the HLT architecture.**

***Index Terms*—ATLAS, event selection, high level trigger, trigger control and supervision.**

NOTE: *This paper was presented by Sarah Wheeler on behalf of the ATLAS High-Level Trigger Group* [1].

I. Introduction

ATLAS is a general-purpose high-energy physics experiment for recording proton-proton collisions, currently under construction at the Large Hadron Collider (LHC) at CERN. ATLAS has been designed to study the largest possible range of physics at the LHC including searches for as yet unobserved phenomena such as the Higgs boson and supersymmetry.

Manuscript received June 2, 2003; revised November 26, 2003.

J. T. Baines, D. Botterill, W. Li, and F. J. Wickens are with Rutherford Appleton Laboratory, Chilton, Oxon OX11 0QX, U.K.

C. P. Bee, F. Etienne, C. Meessen, Z. Qian, and F. Touchard are with CPPM, 13288 Marseille, France.

A. Bogaerts, A. dos Anjos, P. Pinto, and P. Werner are with CERN, CH-1211 Geneva 23, Switzerland.

M. Bosman, K. Karr, and G. Merino are with Barcelona IFAE, 08194 Bellaterra, Spain.

B. Caron, J. L. Pinfold, and S. Wheeler are with the University of Alberta, Edmondton, AL T6G 2N5, Canada (e-mail: sarah.wheeler@cern.ch).

S. González, W. Wiedenmann, and G. Zobernig are with the University of Wisconsin, Madison, WI 53706 USA.

A. Negri is with the Università di Pavia e I.N.F.N., 27100 Pavia, Italy.

Digital Object Identifier 10.1109/TNS.2004.828875

The ATLAS Trigger and Data Acquisition (TDAQ) system will have to deal with extremely high data rates, due both to the high bunch crossing frequency at the LHC (40 MHz) and the large amount of data produced by the ATLAS detector itself (1–2 Mbytes per event). The task of the TDAQ system is to select from this unprecedented amount of data, the most interesting events and save them for later analysis at a rate of about 200 per second. ATLAS relies on a three-level trigger system to perform the selection: a very fast, coarse granularity, hardware-based LVL1 trigger which reduces the event rate to 75 kHz followed by two software-based triggers, the LVL2 trigger and the Event Filter (EF) which perform increasingly fine-grained selection of events at lower rates. The LVL2 trigger, working on a subset of full granularity detector data reduces the rate to about 2 kHz and finally the EF using full event data reduces the rate to about 200 Hz after which the selected events are written to mass storage. The LVL2 and EF comprise the ATLAS high-level trigger (HLT) system.

II. Overview of the HLT

The HLT is a software trigger implemented as software applications running on large processor farms consisting of commodity components connected via high-speed Ethernet networks. For the sake of simplicity and flexibility the LVL2 trigger and event filter farms have been implemented separately. The farms themselves are split into subfarms for reasons of practicality and organization.

The role of the HLT is to reduce the LVL1 trigger rate to a rate compatible with writing the events to mass storage. Note that this is no longer a technical limit. It is rather constrained by the cost of storage devices and on the computing power available to analyze the data off-line. Input to the HLT consists of events that have passed the LVL1 trigger. The LVL1 trigger is a coarse-grained hardware trigger which uses information from the calorimeters and muon detectors only. In addition to providing the trigger result, the LVL1 trigger identifies the geographical locations in the detector of candidate muons, electrons, photons, jets and hadrons. These are known as regions of interest (RoIs). The RoIs are classified into two types. Primary RoIs that caused the LVL1 accept, and secondary RoIs, which were not used in the LVL1 accept. Both types of RoI are used to seed the LVL2 selection. The concept of seeded reconstruction is fundamental to the HLT.

In parallel with forwarding accepted LVL1 events to the HLT, the corresponding event fragments are sent from the detector

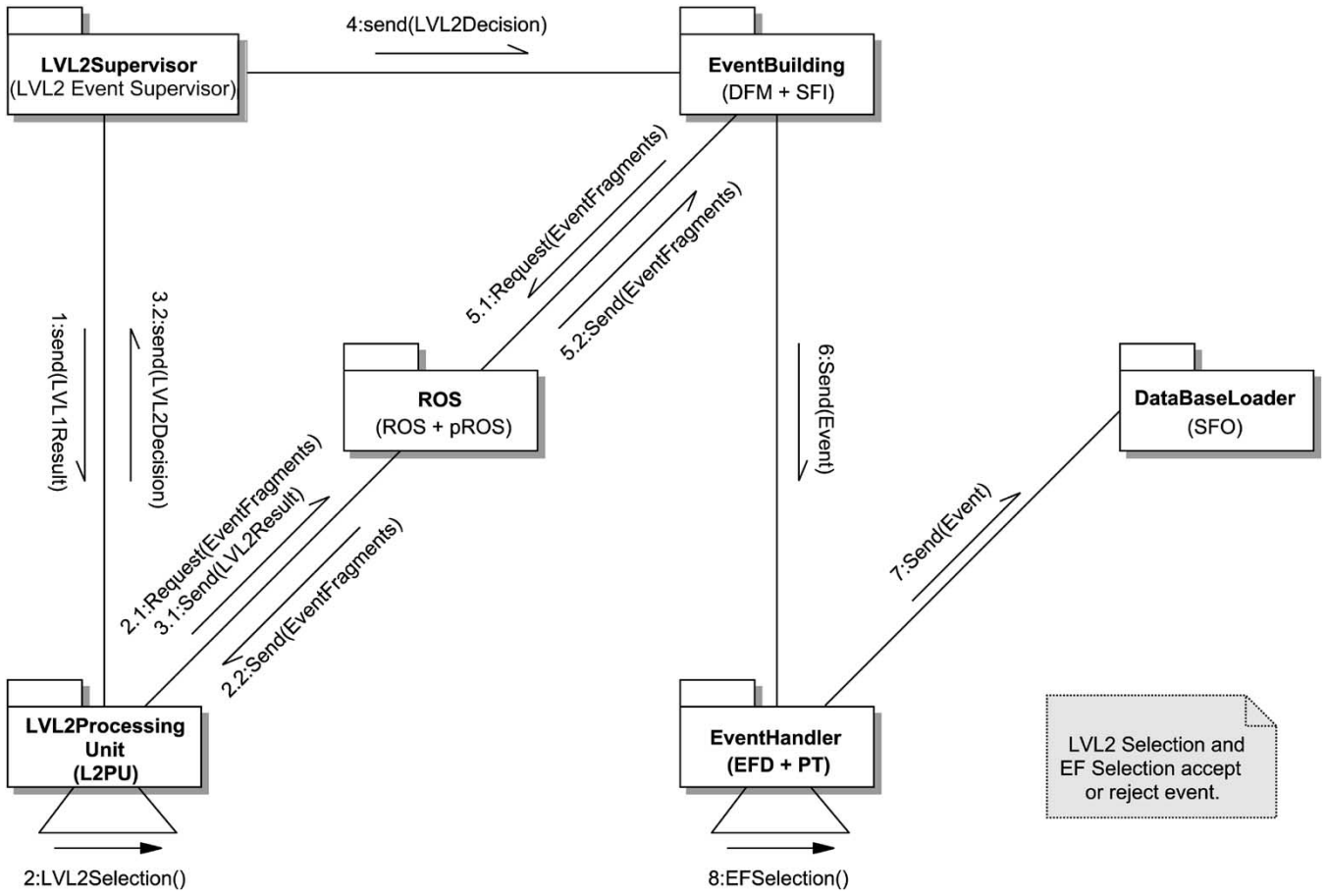

Fig. 1. Functional decomposition of the ATLAS HLT system. The boxes represent the different functions and the name of the software application(s) implementing each function are shown in brackets. The arrows represent the exchange of messages and event data between the various functional components.

front-end electronics to the readout systems (ROS) which contain $\sim$1600 readout buffers.

The various actions which are then executed by the HLT are summarized in the following sections and in Fig. 1. The figure is a logical representation of the HLT, where the boxes represent the various functions and the name(s) in brackets are those of the software application(s) which implement each function. It should be noted that each function is implemented as many instances of the associated software applications.

A. LVL2 Supervisor

The LVL1 trigger sends the RoI information (LVL1 Result) to one of several LVL2 Event Supervisor applications which implement the first function of the LVL2 trigger and therefore of the HLT. The LVL2 Event Supervisor allocates the event to one of many LVL2 Processing Unit applications (L2PUs) and forward the LVL1 result to the selected L2PU.

B. LVL2 Processing Unit

Highly optimized algorithms run in the L2PU applications and perform the selection, which is broken down into a series of sequential processing steps. Typically, the first step involves confirming the LVL1 trigger RoIs. Full-granularity event data (Event Fragments) from the calorimeters and muon detectors are requested from the relevant ROS applications and analyzed. In subsequent steps, data are requested from other detectors, e.g., tracking detectors, corresponding to the RoI and analyzed. At each step, if the result is not consistent with any of the possible physics candidates, the event is rejected and no further processing occurs. Only the events surviving all steps are accepted by the LVL2 trigger. For these, the detailed LVL2 Result is sent to the pROS. For all events the LVL2 Decision is sent to the LVL2 Event Supervisor.

The LVL2 selection uses full-granularity event data, including information from the inner tracking chambers not available to the LVL1 trigger. The RoI mechanism and sequential processing allows this to be done transferring only a few percent of the entire event data to the LVL2 trigger, thereby considerably reducing the network bandwidth required (to $\sim$2.5 Gbyte/s).

C. Event Building

The LVL2 Event Supervisor passes the LVL2 decisions to the event builder and the events accepted by the LVL2 trigger are built. Event fragments in the ROSs are collected under the guidance of Data Flow Manager (DFM) applications and assembled into complete event records by SubFarm Input (SFI) applications.

For the event building, the pROS is treated as just another ROS so the LVL2 Result is included in the event to be built. The bandwidth required for the event building is $\sim$5 Gbyte/s.

D. Event Handler

The complete events are sent from the event builder to Event Filter Dataflow applications (EFDs) running in the event handler (see Fig. 1). The role of the EFD is to distribute the events to so-called Processing Task applications (PTs) which run off-line filtering algorithms, adapted for the EF, to perform the selection process using the entire event data. It is foreseen that the LVL2 result, passed via the pROS should be used by the algorithms to seed the selection in order to reduce the time taken to obtain a result.

E. DataBaseLoader

Events selected by the event handler are sent to permanent storage, via the DataBaseLoader implemented by SubFarm Output (SFO) applications.

III. Functional Components of the HLT

There are many similarities between the two stages of the HLT. The boundary between LVL2 and EF is intentionally flexible in order that the HLT can be configured to take into account different running environments. The aim of this section is to discuss each of these aspects in turn, describing where similarities exist between LVL2 and the EF and how these may be exploited and why in other cases, due to differing requirements, implementations differ.

The HLT consists of three main functional parts:

1) the dataflow code, i.e., the code responsible for transferring the event data and trigger decisions to and from the selection algorithms;
2) the event selection software (algorithms) and the interfaces required for obtaining event data;
3) the supervision system, the software responsible for preparing the HLT for datataking activities.

A. Dataflow

The dataflow system is responsible for moving event data and trigger decisions to and from the selection algorithms. Implementations differ between LVL2 and the EF.

1) LVL2 Dataflow: In the LVL2 trigger, dataflow activities are performed by the LVL2 Event Supervisor and the L2PU applications. These are two of the applications implemented using the dataflow framework [2]. Dataflow is a system of the TDAQ responsible for the movement of event data from the ROS to mass storage via the LVL2 trigger and the EF. The LVL2 Event Supervisor applications send LVL1 results to the L2PU applications. In order to achieve load balancing on the L2PUs, it is foreseen that LVL1 results are assigned to L2PUs on a round-robin or a least-queued basis. The LVL1 result contains only the LVL1 event ID, trigger type and a list of primary and secondary RoIs. It does not contain any event data. It is the responsibility of the L2PU applications to access over the network the event data required from the ROSs. Full event building at the LVL2 input rate would require more than 100 Gbyte/s bandwidth, however by limiting the data requests to the RoIs and only fetching data as algorithms need them the bandwidth reduces to a much more manageable level (~ 2.5 Gbyte/s). The data for each request are assembled and passed to the requesting algorithm via the interface to the LVL2 algorithms inside the L2PU.

2) EF Dataflow: The EF is located after event building, where complete events are already assembled and available, therefore allowing dataflow and selection tasks to be cleanly separated. The dataflow is implemented by the EFD application [3], which receives complete events from the SFI applications and delivers them to the PTs, where event selection takes place. It subsequently collects the analyzed events and delivers them to mass storage via the SFO applications. The EF is the first part of the TDAQ system in which complete events are available. It is therefore a convenient place to perform physics monitoring, calibration and alignment procedures. Therefore, in addition to making events available to the PTs running the selection algorithms, the EFD must sort events, using information in the event headers to different types of Processing Tasks according to the event type. For example, it should send calibration events to the appropriate detector calibration task.

The EFD functionality is divided into specific internal tasks that can be dynamically interconnected to form an EF dataflow network within the EFD application. The tasks can be purely internal (sorting to different data streams, internal monitoring of dataflow) or provide the interface to external components (SFI, SFO, PTs). Within the Event Handler the dataflow is based on reference passing, using a pointer to the event (stored in a memory-mapped file) between the different tasks, thus avoiding copying the events.

B. Event Selection Software (ESS)

The tasks of the ESS are "event selection" and "event classification." The HLT algorithms construct objects representing physics candidates such as electrons, jets, and muons. An event is selected if the reconstructed object is consistent with at least one physics signature. In both the LVL2 and the EF, events are rejected if they do not pass any of the selection criteria.

A common ESS architecture has been developed for use across HLT. This greatly simplifies migration of algorithms between the various environments. For instance, although LVL2 algorithms will not be used in the off-line context to analyze data they can be developed and tested in the more user-friendly off-line environment. Furthermore, it becomes trivial to move LVL2 algorithms to the EF. For instance, LVL2 algorithms might be moved to the EF to crosscheck the efficiency of the algorithms, or to take advantage of more accurate calibration information. Due to the short decision time of the LVL2 trigger, L2PUs will only be able to access calibration databases at the start of each datataking run. In the EF it will be possible to access calibration databases on an event-by-event basis. It should also be remembered that flexibility in the configuration of the HLT will be vital for tuning the trigger to select events corresponding to novel, unpredicted physics.

Since constraints differ for the LVL2 trigger and EF, different implementations exist for the algorithms and dataflow interfaces. These are described in more detail in the following sections.

1) LVL2 Trigger Selection Algorithms: The ESS in the LVL2 trigger is implemented by specially written algorithms running in the L2PUs, further details may be found in [4]. In the

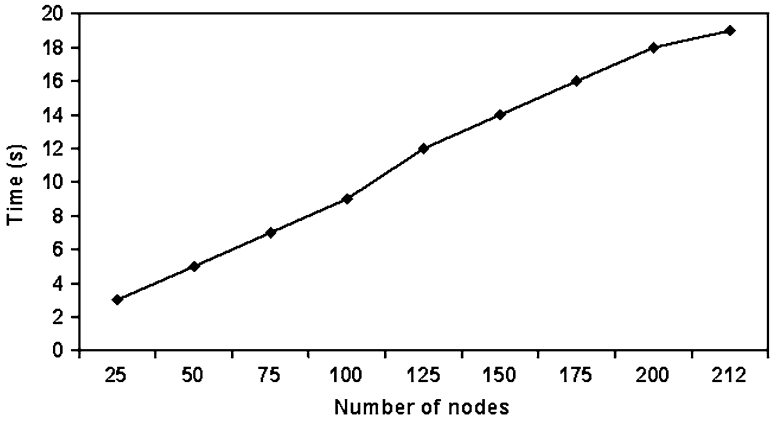

Fig. 2. The plot shows the time required to launch varying numbers of EF PTs on a 212 node testbed using the Axis implementation of Web services. The number of PTs launched per processing node was varied from 2 to 80. Good linearity is observed and the performance is satisfactory considering that no optimization in the use of the services was sought and no asynchronous operations were used due to limitations in the current implementation.

LVL2 trigger, the average trigger processing time must be short ($\sim$10 ms) because of the high rate to be handled. Therefore, algorithms must be highly optimized. Furthermore, no disk access is allowed while datataking is in progress requiring that all necessary database information is available in memory. Algorithms running in the LVL2 environment must also satisfy stringent thread-safety rules. In the L2PU, the selection algorithms run inside one of several "workerthreads," where each workerthread is responsible for processing one event. This multithreaded approach has been chosen to avoid stalling the CPU when waiting for requested RoI data to arrive from the ROSs. It also allows efficient use of multi-CPU processors.

2) EF Selection Algorithms: In the EF the requirement on the trigger decision time is less severe ($\sim$1 s). This allows access to disk-based databases during datataking activities, permitting the use of up-to-date calibration constants in the event reconstruction and selection procedure. Second, the algorithms run in single-threaded processing tasks. Therefore, algorithms need not be thread-safe. Studies are underway [4] in the EF to assess the suitability of all the off-line algorithms for use in the on-line environment and changes implemented where necessary. For example, although the trigger decision time is less critical in the EF, it may still not be entirely practical to run the algorithm with all the precision that it would have in an off-line environment because it takes too long. It may be "detuned" in order to achieve a decision more quickly.

3) Dataflow Interfaces: Due to the different environments in which the ESS runs, the interface used by the ESS to collect event data is different in LVL2 and EF as well as in the off-line situation. In the case of LVL2, a subset of detector data, in on-line format, corresponding to LVL1 RoIs is fetched across the network from the ROSs. In the EF, full-event data, in on-line format, is accessed from the EFD via a memory-mapped file. In the off-line environment, the event data will be received from databases, in off-line format. However, a key design decision has been that the data should be delivered to the algorithms in exactly the same way in all environments. The algorithms themselves are not aware of where they are running. This has been achieved by using the off-line Gaudi [5] and ATHENA [6] frameworks to implement the LVL2 and EF interfaces. The interfaces also provide data conversion services to transform the incoming event data from the on-line format into the off-line format expected by the algorithms.

C. Supervision

The supervision system [7] is responsible for all aspects of software task management and control in the HLT. Mandates of the supervision system include: configuration of the HLT software processes, synchronizing the HLT processes with datataking activities in the rest of the experiment, monitoring the status of HLT processes, e.g., checking that they are running and restarting crashed processes. Both the LVL2 trigger and the EF are implemented as hundreds of software processes running on large processor farms, split for reasons of practicality into a number of subfarms. In view of this, the supervision requirements for the two systems are very similar and an integrated HLT supervision system has been developed. It has been implemented using services provided by the Online Software (OnlineSW) [8]. The Online Software is a system of the TDAQ project. It encompasses the software to configure, control, and monitor the TDAQ. It does not contain any elements that are detector specific and has been successfully adapted for use in the HLT trigger farms.

The OnlineSW configuration database is used to describe the HLT in terms of the software processes and hardware (processing nodes) of which it is comprised. The HLT supervision and control system uses information stored in the OnlineSW configuration database to determine which processes need to be started on which hardware and subsequently monitored and controlled. The smallest set of HLT elements, which can be configured and controlled independently from the rest of the TDAQ system is the subfarm. This allows subfarms to be dynamically included/excluded from partitions during datataking. Synchronization with the rest of the TDAQ system is achieved using the OnlineSW run control system. Each subfarm has a local run controller, which will interface to the Online Software run control via a farm controller. The controller collaborates with a subfarm supervisor, which provides process management and monitoring facilities within the subfarm. The controller and supervisor cooperate to maintain the subfarm in the best achievable

state by keeping each other informed about changes in supervision or run-control state and by taking appropriate actions, e.g., restarting crashed processes. Where possible, errors should be handled internally within the HLT processes. Only when they cannot be handled internally should errors be sent to the supervision and control system for further consideration.

Scalability tests carried out at CERN [9] using a prototype supervision system implemented with the OnlineSW showed that execution times taken for starting and stopping HLT trigger processes and for performing run control transitions to prepare for datataking do not depend strongly on the number of controlled nodes. The times are shorter than 3 s for configurations of a size varying between 50 and 230 nodes and running up to approximately 1000 HLT software processes.

The Axis implementation of Web services [10] is currently being investigated for possible use in future implementations of the supervision systems. Scalability tests (see Fig. 2) have been encouraging [9]. The HiWesD Histogram Web Service Demonstrator (based on Axis) has already been used successfully in both the LVL2 and EF context.

IV. HLT Integrated Testbed

An integrated HLT prototype system has been built at CERN. The system is a functional integration of previously existing LVL2 and EF prototypes. A schematic diagram of the testbed is shown in Fig. 3. The integrated prototype consists of the following components.

1) One ROS preloaded with event data from a file. The data in the file correspond to $\sim$1000 LVL1 trigger accepted events containing electron and jet RoIs.
2) One pROS to pass the LVL2 trigger result to the EF.
3) A LVL2 system consisting of a LVL2 Event Supervisor (L2SV on the diagram) and two L2PUs. The L2PUs run the TCALO algorithm for e/gamma identification.
4) An Event Builder consisting of a DFM and SFI.
5) An EF system consisting of three EFDs and six PTs running the TCAL algorithm in the ATHENA environment.
6) One SFO application to which the EFD sends the event once the PTs have completed the selection.
7) OnlineSW supervision infrastructure to provide synchronization and process control.

The computing infrastructure from previous LVL2 and EF standalone tests has been reused. The ROS, LVL2, and Event Builder (including the SFI and SFO applications) run on a system in one building at CERN, using PCs and a local switch. The EF components, consisting of the EFD and the ATHENA PTs run on a system located in another building, using PCs and a local switch.

The main emphasis of the prototype is on functionality. Detailed performance measurements have been made in separate LVL2 and EF slice tests [11] and the execution time of algorithms can be measured in off-line mode. However, some of the measurements made for the standalone systems will be repeated in the integrated testbed to verify its correct functioning.

The LVL2 and EF control and supervision infrastructures have been successfully integrated to create a common control structure. This supervision system has been used to start all the HLT processes and bring them to the "running" state. Events have been observed to flow through the full system. Tests are currently underway to repeat key measurements made on the standalone testbeds, in particular, the throughput in the EF will be measured as this depends critically on the correct functioning of the link between the LVL2 and EF parts of the system.

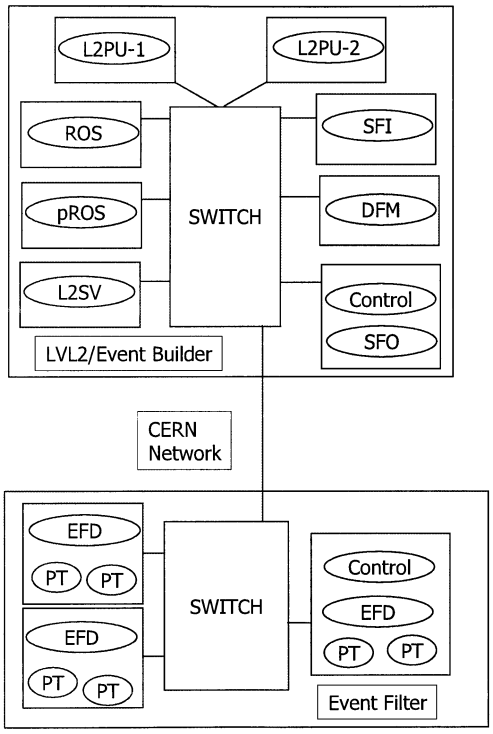

Fig. 3. Schematic diagram showing the configuration of the HLT integrated testbed. The top box represents the previously existing LVL2/Event Builder testbed and the bottom box the previously existing Event Filter testbed. The two testbeds are located in different buildings at CERN. For the integrated HLT prototype they were linked via the CERN network. The small boxes represent processing nodes and the ellipses, the applications running on those processing nodes. For the sake of simplicity the supervision is represented on the diagram by a single "control" application for LVL2 and the EF. In reality it is a number of interacting applications.

V. Conclusion

A final design now exists for the HLT system and is presented in the ATLAS TDAQ Technical Design Report (TDR) [11]. LVL2 and EF prototypes based on the design have already been implemented and the required performance demonstrated in the critical area of data access and the principal aspects of system scalability. Integration of the LVL2 and EF prototypes into a coherent HLT prototype is currently underway and tests will be performed to test functionality and validate the design. Results are included in the ATLAS TDAQ TDR.

Acknowledgment

The authors would like to thank the many people in the ATLAS collaboration who made this work possible, particularly in the other parts of ATLAS Trigger/DAQ.

REFERENCES

[1] ATLAS high-level trigger group author list for real time 2003 (2003, May). [Online]. Available: http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/HLT/AUTHORLISTS/rt2003.pdf

[2] H. P. Beck *et al.*, "The base-line dataflow system of the ATLAS trigger and DAQ" (in Available Online: http://doc.cern.ch//archive/electronic/cern/others/atlnot/Communication/daq/com-daq-2003-037.pdf), *IEEE Trans. Nucl. Sci.*, pp. 470–475, June 2003.

[3] ATLAS TDAQ/DCS Event Filter Event Handler Design, C. Bee *et al.*. (2002, July). [Online]. Available: https://edms.cern.ch/document/367 089/1

[4] S. Armstrong, "An overview of algorithms for the ATLAS high-level trigger" (in Available Online: http://doc.cern.ch//archive/electronic/cern/others/atlnot/Note/conf/conf-2003-003.pdf), *IEEE Trans. Nucl. Sci.*, pp. 367–374, June 2003.

[5] G. Barrand *et al.*. GAUDI—A software architecture and framework for building HEP data processing applications. presented at Proc. CHEP 2000 Computing in High Energy and Nuclear Physics. [Online]. Available: http://lhcb-comp.web.cern.ch/lhcb-comp/General/Publications/longpap-a152.pdf

[6] Athena Web Site (2001, Jan.). [Online]. Available: http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/General/index.html

[7] ATLAS TDAQ HLT-Online Software Interface, D. Burckhart-Chromek *et al.*. (2002, Nov.). [Online]. Available: https://edms.cern.ch/document/363 702/1

[8] I. Alexandrov *et al.*, "Online software for the ATLAS test beam data acquisition system" (in Available Online: http://doc.cern.ch//archive/electronic/cern/others/atlnot/Note/daq/daq-2003-044.pdf), *IEEE Trans. Nucl. Sci.*, pp. 578–584, June 2003.

[9] ATLAS TDAQ Event Filter Test Results for the EF Supervision, S. Wheeler, F. Touchard, Z. Qian, C. Meessen, and A. Negri. (2003, Mar.). [Online]. Available: https://edms.cern.ch/document/374 118/1

[10] Web Services—Axis (2000–2003). [Online]. Available: http://ws.apache.org/axis/

[11] "The ATLAS High-Level Trigger Data Acquisition and Controls Tech. Design Rep.," ATLAS HLT/DAQ/DCS group, Geneva, Switzerland, ATLAS TDR-016, June 2003. CERN.