Design, deployment and functional tests of the online event filter for the ATLAS experiment at LHC

S. Armstrong, A. dos Anjos, J. T. M. Baines, C. P. Bee, M. Biglietti, J. A. Bogaerts, V. Boisvert, M. Bosman, B. Caron, P. Casado, et al.

To cite this version:

S. Armstrong, A. dos Anjos, J. T. M. Baines, C. P. Bee, M. Biglietti, et al.. Design, deployment and functional tests of the online event filter for the ATLAS experiment at LHC. IEEE Transactions on Nuclear Science, 2005, 52, pp.2846-2852. 10.1109/TNS.2005.862790 . in2p3-00145437

HAL Id: in2p3-00145437
https://hal.in2p3.fr/in2p3-00145437

Submitted on 10 May 2007

Design, Deployment and Functional Tests of the Online Event Filter for the ATLAS Experiment at LHC

S. Armstrong, A. dos Anjos, J. T. M. Baines, C. P. Bee, M. Biglietti, J. A. Bogaerts, V. Boisvert, M. Bosman, B. Caron, P. Casado, G. Cataldi, D. Cavalli, M. Cervetto, G. Comune, P. Conde Muino, A. De Santo, M. Diaz Gomez, M. Dosil, N. Ellis, D. Emeliyanov, B. Epp, F. Etienne, S. Falciano, A. Farilla, S. George, V. Ghete, S. González, M. Grothe, S. Kabana, A. Khomich, G. Kilvington, N. Konstantinidis, A. Kootz, A. Lowe, L. Luminari, T. Maeno, J. Masik, A. di Mattia, C. Meessen, A. G. Mello, G. Merino, R. Moore, P. Morettini, A. , N. Nikitin, A. Nisati, C. Padilla, N. Panikashvili, F. Parodi, V. Perez Reale, J. L. Pinfold, P. Pinto, Z. Qian, S. Resconi, S. Rosati, C. Sanchez, C. Santamarina, D. A. Scannicchio, C. Schiavi, E. Segura, J. M. de Seixas, S. Sivoklokov, R. Soluk, E. Stefanidis, S. Sushkov, M. Sutton, S. Tapprogge, E. Thomas, F. Touchard, B. Venda Pinto, V. Vercesi, P. Werner, S. Wheeler, F. J. Wickens, W. Wiedenmann, M. Wielers, and G. Zobernig

Abstract—**The Event Filter (EF) selection stage is a fundamental component of the ATLAS Trigger and Data Acquisition architecture. Its primary function is the reduction of data flow and rate to values acceptable by the mass storage operations and by the subsequent offline data reconstruction and analysis steps. The computing instrument of the EF is organized as a set of independent subfarms, each connected to one output of the Event Builder (EB) switch fabric. Each subfarm comprises a number of processors analyzing several complete events in parallel. This paper describes the design of the ATLAS EF system, its deployment in the 2004 ATLAS combined test beam together with some examples of integrating selection and monitoring algorithms. Since the processing algorithms are not explicitly designed for EF but are adapted from the offline ones, special emphasis is reserved to system reliability and data security, in particular for the case of failures in the processing algorithms. Other key design elements have been system modularity and scalability. The EF shall be able to follow technology evolution and should allow for using additional processing resources possibly remotely located.**

Index Terms—**Data acquisition, data processing, triggering.**

I. INTRODUCTION

THE A Toroidal LHC ApparatuS (ATLAS) detector [1] is a High Energy Physics (HEP) experiment designed to exploit the full physics potential provided by the Large Hadron Collider (LHC), under construction at CERN. Its inner elements are tracking detectors enclosed in a solenoidal magnet, which is in turn surrounded by the calorimetry system. The global detector dimensions (diameter 22 m, length 42 m) are defined by a large air-core muon spectrometer, whose toroidal magnetic geometry motivates the detector name. The physics program [2] is widely diversified; it ranges from discovery physics to precision measurements of the Standard Model parameters. LHC will provide pp collisions at a centre-of-mass energy of 14 TeV and a design luminosity of 10^{34} cm^{-2}s^{-1}. The corresponding 40 MHz bunch crossing rate (with an average of $\sim$23 superimposed events) and the huge amount of detector channels ($\sim 10^8$) outline the challenge of the ATLAS Trigger and Data Acquisition (TDAQ) system.

II. ATLAS TDAQ

The ATLAS TDAQ system must be able to select and store each second, out of millions of events (1 GHz interaction rate corresponding to about 60 TB/s), the most interesting ones, with a tolerable storage rate of some hundreds MB/s. Given an average event size of $\sim$1.5 MB, the event rate is then about 200 Hz. The required data reduction, equivalent to a rejection factor of about 6 orders of magnitude, is achieved online via a data acquisition system organized in three different trigger levels (Fig. 1).

Each level refines the decisions made at the previous one and, where necessary, applies additional selection criteria. The processing time available at each level increases, allowing the use of greater amounts of information to either accept or reject the event. The first trigger level (LVL1), realized in hardware by custom electronics, reduces the data rate from the 40 MHz collision rate to about 75 kHz. The High Level Triggers (HLT, composed of LVL2 and Event Filter (EF)) [3], implemented on two different commodity component farms, provide a further reduction factor of about 10^3.

The LVL1 trigger [4] is directly connected to the detector front-end electronics of the calorimeter and muon detectors. Data from accepted events are stored in pipeline memories, connected to the read-out drivers (RODs) and made available to the HLT through about 1600 read-out buffers (ROBs). Several ROBs are logically grouped in Read Out System (ROS) elements. For accepted events the LVL1 identifies the detector regions, defined in rapidity and azimuthal angle, where the signals exceed programmable thresholds. These Region of Interests (RoIs) are used to guide the LVL2 selection process which can access full granularity event data from all detectors. The selection algorithms request data only from the ROBs corresponding to the LVL1 defined RoIs. In this way, only 2% of the

---

Manuscript received November 15, 2004; revised April 5, 2005.

Please see the Acknowledgment section of this paper for the author affiliations.

Digital Object Identifier 10.1109/TNS.2005.862790

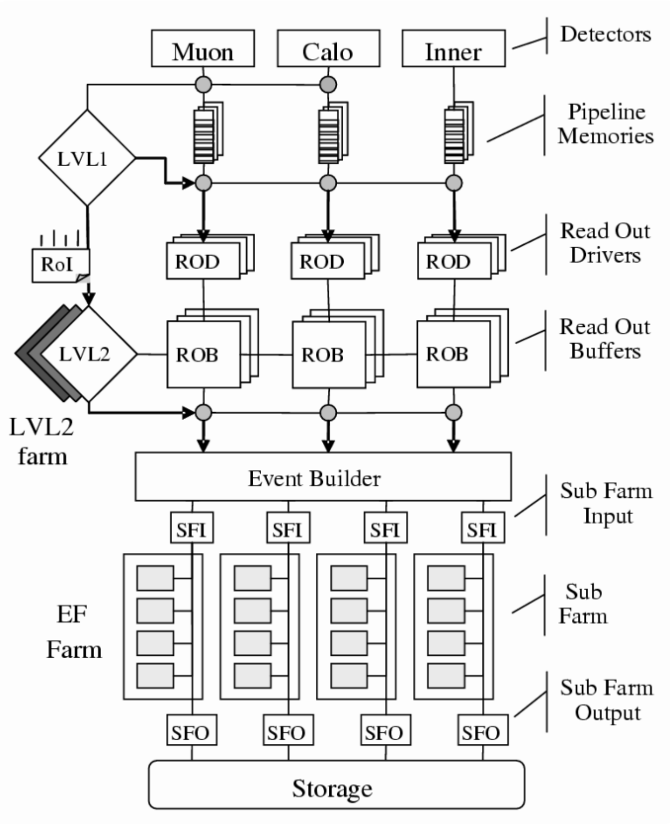


Fig. 1. Block diagram of the ATLAS TDAQ system.

full event data are needed for the LVL2 decision process, thus reducing dramatically the size of the network needed to serve the LVL2. The data are held in the ROBs until the LVL2 accepts or rejects the event. The results of the LVL2 processing are stored in an additional ROS (called pROS) and can be used to guide the EF reconstruction in the same geometrical region of the detector (seeding mechanism).

If an event is accepted by LVL2, the Event Builder (EB) collects all the event data fragments from the ROBs. The complete event is then made available to the EF for the final stage of trigger processing. At the EF, more complex algorithms provide a further rate reduction, down to about 200 Hz with typical decision times of about 1 s. While the LVL2 reconstructs localized regions, the baseline for the EF is a full offline-like event reconstruction guided by the LVL2 result. It also benefits from more complete calibration, alignment, and magnetic field information.

III. THE EF SYSTEM

The EF selection stage has become a fundamental component of high energy physics TDAQ architectures. Its name reflects the fact that it is located downstream of the EB and therefore can operate on full event data. Its primary function is the reduction of data flow and rate to a value acceptable by the mass storage operations and by the subsequent offline data reconstruction and analysis steps. The EF can also provide initial event sorting into streams for offline production and global physics and detector monitoring, essential to ensure the quality of recorded data. It is usually characterized by modest input rate, a low as possible output rate, and very high computational needs. Indeed, whereas the upstream trigger levels are latency and bandwidth limited, the EF is highly processing time dominated.

The ATLAS EF system is organized as a set of independent processor farms (subfarms), each connected to one output port of the EB switch (Subfarm Input (SFI) elements). The SFIs perform the actual building operation, while the Subfarm Output (SFOs) are the interfaces to the storage system (in the baseline design each subfarm connects to a single SFI and a single SFO, but the possibility to use multiple SFI and SFO connections will enhance the redundancy and modularity of the system). The final dimensions of the ATLAS EF system are not yet fully fixed and will also vary during the lifetime of the experiment, but the total number of processors will be of the order of thousand.

The running environment for the trigger algorithms is the HLT event selection software framework (ESS [5]), which is based on the ATLAS offline reconstruction and analysis environment ATHENA [6]. A common framework for developing and running both the online and offline software allows the reuse of existing offline algorithms, facilitates the development procedures, and guarantees the consistency of trigger performance evaluation and trigger selection validation (avoiding selection biases). The *HLT Steering* schedules the *HLT Algorithms* corresponding to the input seed (LVL2 result) so that all necessary data for a trigger decision are produced. The *HLT Algorithms* either reconstruct new event quantities or check the computed event features against a list of trigger hypotheses.

IV. DESIGN

The challenging ATLAS online environment imposes strong requirements on the design of the EF system [7]. System modularity and scalability are important design elements because the EF processing resources will evolve during the lifetime of the experiment and it must be possible to track the technology evolution. Fault tolerance and data security are fundamental requirements for any online architecture. Events wrongly rejected are lost forever and insufficiently robust code is bound to crash often, reducing the live time of the online DAQ. These requirements become even more critical in the context of the ATLAS selection, since EF algorithms are not specifically developed for the online environment, but are inherited from the offline one. Therefore, even if particular attention is paid to ensure the highest possible robustness of the processing code, the EF framework must provide additional reliability in case of an algorithm crash in order to avoid biases to the recorded physics sample (some crashes could be related to specific event topology and the loss of these events could invalidate the physics results).

The design of the ATLAS EF system is object-oriented, it is fully implemented in C++, and it uses multithread programming techniques. The multithreaded approach minimizes overheads from context-switching and avoids stalling the CPU during I/O operations. Asynchronous services are executed in separate threads. This allows a better exploitation of Symmetric Multi-Processor (SMP) architectures.

The key principles of the EF design are the following:

- decoupling of the event reconstruction and selection from the data flow functionalities;

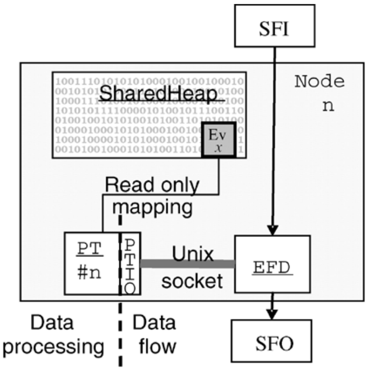


Fig. 2. Subfarm node design: decoupling between data processing operations and data flow functionalities.

- data driven event flow (i.e., no data flow manager to assign the event to a specified target) without data copying inside the processing host;
- local storage based on memory mapped files for event recovery capability;
- exploitation of SMP architectures.

A. Systemwide Architecture

In order to allow the dynamic insertion or removal of processing resources, each processing host manages its own connection with the SFI and SFO elements and implements the client part of the communication protocol. Processing nodes can then be transparently added to a running subfarm and software or hardware failures in a processing node do not affect the operation of other subfarm elements. Similarly full subfarms can be easily hot-plugged into a running system. Furthermore the design supports geographically distributed implementations and depending on the network topology, dynamic rerouting in case of SFI malfunction or crash.

B. Processing Node Architecture

In order to ensure data security and fault tolerance, the data processing operations are completely decoupled from the data flow functionalities in each node. The latter ones are provided by the Event Filter Data Flow (EFD) process that manages the communication with the SFI and SFO elements and makes the events available to the processing tasks (PTs), which are in charge of data processing and event selection (Fig. 2).

The PTs are implemented as separate processes running the EF algorithms in the standard ATLAS offline framework [6]. The events are made available to the PTs via shared memory (called the SharedHeap), which stores the events during their transit through the processing node. Communication and synchronization between EFD and PTs is maintained via messages exchanged on a UNIX domain socket.

The EFD function is divided into different specific tasks which can be dynamically interconnected to form a fully configurable EF dataflow network (Fig. 3 shows an example of an internal data flow implementation). Tasks are daisy chained, with each task knowing the identity of the next task to be executed. All object pointers used in the EFD are implemented as smart pointers providing garbage collection facilities.

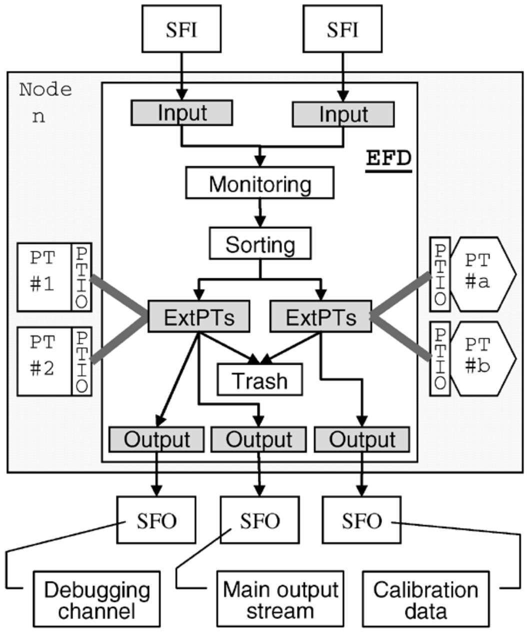


Fig. 3. Implementation example of the EFD process (internal data flow). The figure shows several task types and many possible event paths. The PTs on the left are processes running selection software, while the ones on the right are performing calibration.

The tasks that implement interfaces to external components (SFI, SFO, and PTs) are executed by dedicated threads in order to absorb communication latencies and enhance performance. Input and output tasks manage the communication with the SFI and SFO elements while the ExtPT task implements the interface to the PTs.

When an event is received by the input task, it is stored in the SharedHeap. The event remains in the SharedHeap until it is rejected or sent, by the output task, to the downstream SFO element. A plug-in interface (the PTIO library) allows PTs to access the data flow by opening a connection with the UNIX domain socket server implemented by the ExtPT task.

When a PT (one or many per host) requests an event, the PTIO library transmits the request to the EFD, obtains the offset and size of the SharedHeap portion containing the event to be processed, and maps this portion in memory. The returned memory pointer is used by the PT to access the event and process it. Since the map is read only, the PT cannot corrupt the event or the SharedHeap structure. PT problems are reliably handled by the EFD which can identify PT crashes via socket hang-ups and PT dead locks by means of configurable processing timeouts. In both cases, the EFD, which owns the event, can assign it to another PT or send it directly to the SFO.

Inside the PT, the event processing operation produces a filtering decision and a selection object (used to classify the event and guide the offline analysis step) which are communicated back to the EFD. The selection object is stored in the event header, while the filtering decision is used to steer the internal dataflow. Referring to Fig. 2, the event selection PTs can, for example, sort the processed events to either the main output stream, the trash task (the event is deleted) or to a special debugging channel if processing problems occurred.

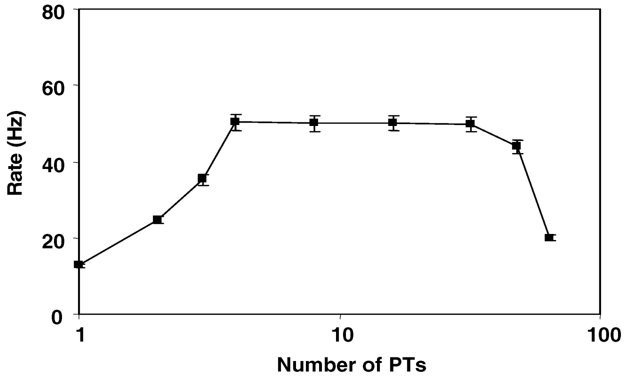


Fig. 4. Architecture scalability. Node processing rate as a function of the number of processing tasks. The PTs run a muon track reconstruction algorithm on data recorded at test beam. The hardware is an SMP machine with 4 Xeon CPUs clocked at 3.2 GHz and 4 GB of main memory.

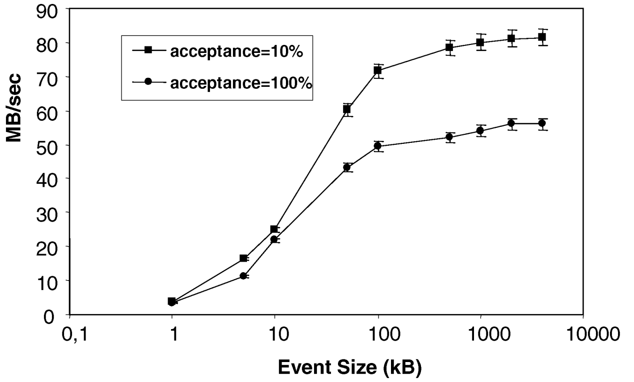


Fig. 5. Node throughput as a function of the event size for two different values of the PT acceptance.

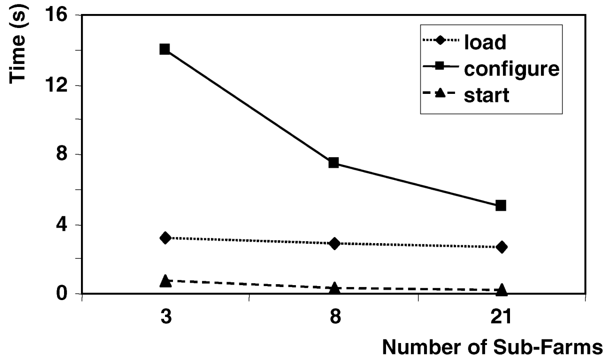


Fig. 6. Times for the different configuration steps as a function of the number of subfarms for a constant full farm size of 230 nodes.

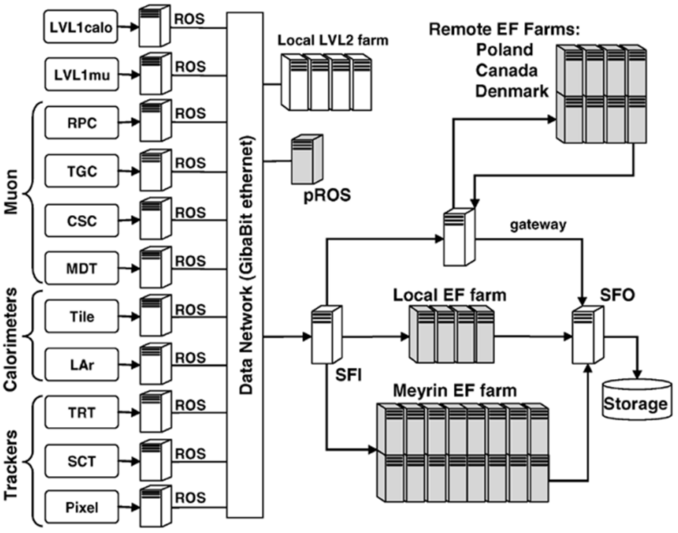


Fig. 7. ATLAS combined test-beam setup. It comprises almost all the detector types, the LVL2 farm, and some EF farms.

The processing operation can provide some additional reconstructed information, which can be added to the raw event. In this case, these data are serialized by the PT in a writable SharedHeap zone provided on demand by the EFD. The PTIO maps in write mode this portion only and therefore event security is still assured. The EFD takes care of appending this data fragment to the raw event.

The SharedHeap is implemented as a memory mapped file hosted by the local file system. The memory resource, allocated at configuration time, is handled by a simple and efficient algorithm that allows the dynamic management of memory blocks of dimension 2^n bytes. The memory mapped file solution provides a safe and elegant event recovery mechanism. Indeed, in case of EFD crash, the events can be recovered from the file system at EFD restart. The OS itself directly manages the write operations avoiding useless disk I/O overhead. The system would only be out of synchronization in case of power cut, OS crash, or disk failure. However, these occurrences are completely decoupled from the event types and topology and therefore do not entail physics biases on the recorded data.

V. FUNCTIONAL TESTS

Extensive validation tests of the EF system have been performed on test-beds of different sizes using special SFIs, which emulate event building behavior. They read recorded events from file and make them available to the EFDs. Details of these tests are given in [8] and [9]. The tests address robustness and fault tolerance of the implementation and the design scalability both at the node and farm level.

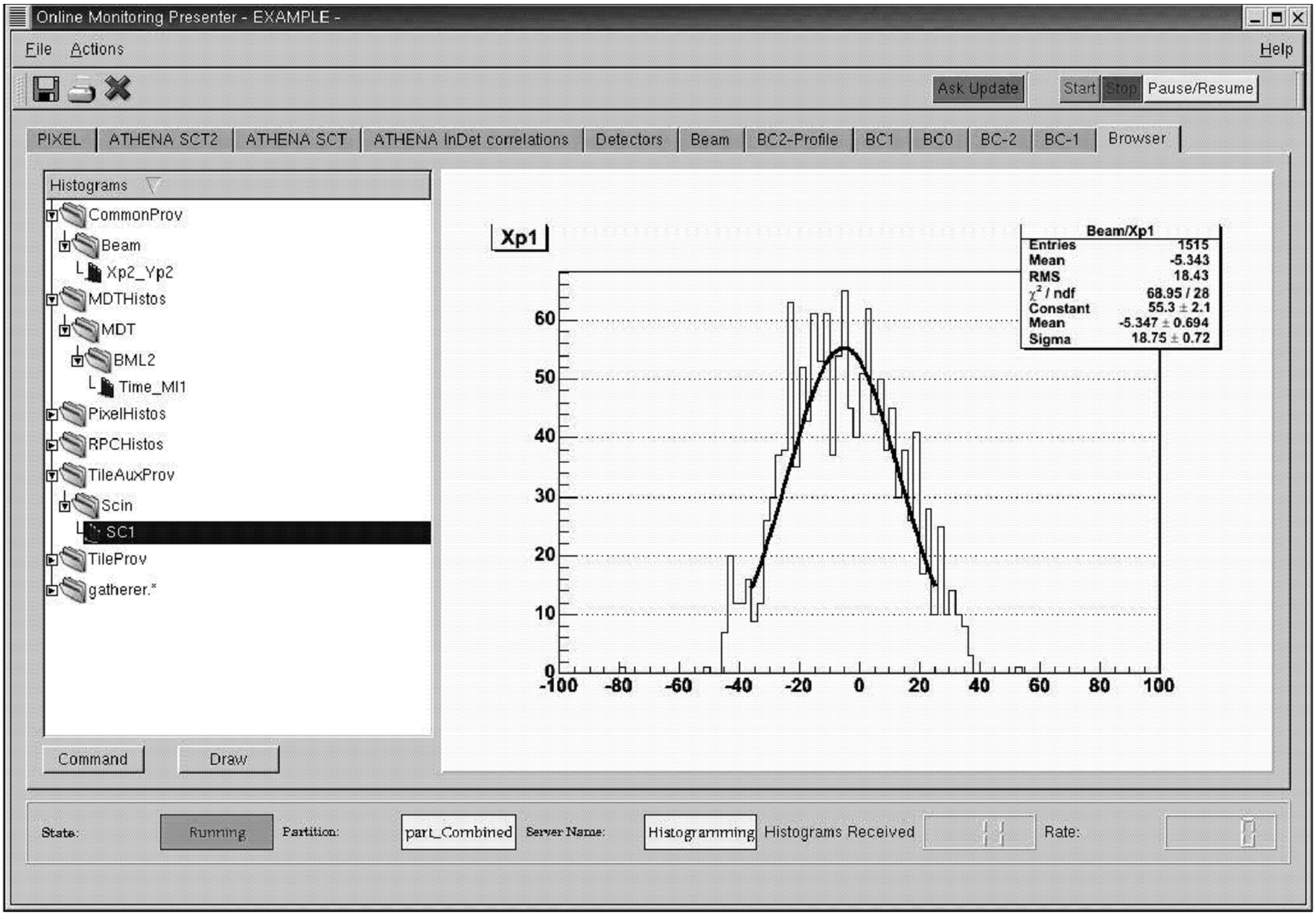
A. *Robustness and Fault Tolerance Tests*

Robustness and fault tolerance of the EF design has been tested in a test-bed composed of 4 EFD nodes, each including 4 PTs. The system ran for 10 days (more than 3×10^9 events handled) without problems or event losses. The test included the repeated random killing of PTs and the subsequent restart of the processes. The EFD correctly handled the situation, recovering the events owned by the killed PT and assigning them to a different PT. No event loss was observed.

It has also been verified that events can be recovered from the SharedHeap when the EFD itself crashes (when it is killed).

B. *EFD Scalability*

Each PT executes in a loop the following sequence of actions: requests an event; maps the event in memory; processes the event; sends the selection decision to the EFD; unmaps the event. Ignoring the event processing step, the measured time to perform this sequence is about 80 μs on a dual processor Xeon at



State: Running
Partition: part_Combined
Server Name: Histogramming

Fig. 8. Example of a data monitoring histogram produced online and displayed by the Presenter.

2.2 GHz and does not depend on the size of the event. This overhead is negligible compared to the nominal average processing time (of the order of 1 s) and defines the upper throughput limit per PT: more than 10 kHz.

The architecture proved to be scalable in terms of the number of PTs and showed correct exploitation of the SMP architecture. Fig. 4 shows the total processing rate in a node as a function of the number of running PTs. The rate increases with the number of PTs, until this quantity equals the number of available CPUs. Then the rate remains independent from the number of PTs until the available memory resources are exhausted.

C. EF Data Flow Performance

Fig. 5 shows the EFD throughput as a function of the event size. The test, performed without real processing, is used to evaluate the communication protocol between the EFD and the adjacent data flow elements (SFI and SFO).

For realistic event sizes the gigabit link is reasonably exploited by the protocol. The maximum throughput, about 30 events/s, is adequate for the host requirements: ~1 event/s per CPU. A rate limitation is visible for small event sizes. This is due to the handshaking design of the communication protocol: in order to enhance data security, the EFD asks for a new event only after the previous one has been received. Therefore the rate is limited by the inverse of the TCP packet round trip time. This leads to difficulties for remote farm implementations where the communication latency becomes sizeable. Improvements of the communication procedures are currently under evaluation.

Large scale scalability tests [11] have been carried out using the CERN IT LXSHARE test-bed (about 300 dual processors, representing about 20% of the final EF farm). Although the tests mainly focused on the HLT control architecture, they also confirmed the reliability of the design. Configurations with up to 21 subfarms, ~300 EFDs and ~16 000 PTs have been run without problems related to the EF infrastructure (Fig. 6).

VI. DEPLOYMENT

The EF system implementation has also been deployed in test-beam environments: from the summer of 2002 it has been an integral part of the ATLAS test-beam data acquisition chain. The EF is currently used in the 2004 ATLAS combined test-beam, whose aim is the functional integration of all the DAQ elements. Indeed the layout (see Fig. 7) includes almost all the detector types (together with their front-end electronics), all the data collection components, and both the HLT levels. The EF farm comprises a local subfarm (4 dual Xeon PC's 3.2 GHz, 1 GB of memory) and a subfarm located a few kilometers away from the test-beam site (20 dual Xeon 600 MHz).

During the test-beam, geographically distributed subfarm implementations were also tested. Remote farms located in Canada and Poland were integrated into the running data acquisition system. Even though the rate was limited to a few Hertz due to the above described communication protocol implementation, events were correctly transferred to remote processing resources (nodes geographically distributed running an EFD instance and some PTs) and received back at the SFO level.

Owing to the fact that the offline and online environments share the same development and running framework, the integration of reconstruction and monitoring algorithms in the EF system is transparent: any offline algorithm properly configured for the test-beam setup (suitable geometry description and data decoding components) can run in the EF framework. Consequently monitoring algorithms for many detectors have been integrated into the running system. The histograms produced by the different PTs are collected and merged by a dedicated online element [12] and are used to monitor the quality of the recorded events. The shifters can use an interactive presenter program to analyze in real time the produced histograms (see Fig. 8).

The simultaneous presence of all ATLAS trigger levels and, in particular, the integration of the HLT muon reconstruction and selection algorithms into LVL2 and EF, allowed the online validation of the full muon selection slice. In particular the transfer of the LVL2 result (contained in the pROS) to the EF and its subsequent decoding has been verified. The LVL2 result is used by the HLT Steering element to call the correct EF algorithm for the given event.

The use of simple event selection rules allowed the testing of two important EFD capabilities:

- event tagging accordingly to the processing result (classification label written in the event header);
- event sorting to different output streams (SFOs).

VII. CONCLUSION

In this paper we have described the design of the ATLAS EF system, the test-bed validation of its implementation and the deployment in the ATLAS combined test-beam. The object-oriented design relies on multithread programming techniques and fulfils the scalability and modularity requirements.

Data security is the key element of the EF system design. Data processing is separated from data flow operations, so that crashes of the selection algorithms will not lead to event loss or corruption. The data communication protocol based on a memory mapped file provides an elegant event recovery mechanism.

Several test-bed functional tests have validated the design and its implementation. The system has been deployed at the ATLAS test-beam and is currently used for data acquisition monitoring, online reconstruction and validation of the HLT selection architecture.

ACKNOWLEDGMENT

S. Armstrong is with Brookhaven National Laboratory, Upton, New York, USA.

A. dos Anjos, A. G. Mello, and J. M. de Seixas are with COPPE/EE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil.

J. T. M. Baines, D. Emeliyanov, and F. J. Wickens are with the Rutherford Appleton Laboratory, Chilton, Didcot, U.K.

C. P. Bee, F. Etienne, C. Meessen, Z. Qian, and F. Touchard are with Centre de Physique des Particules de Marseille, IN2P3, CNRS, Université d'Aix, Marseille 2, France.

M. Biglietti is with the University of Michigan, Ann Arbor, Michigan, USA.

J. A. Bogaerts, V. Boisvert, P. Conde Muino, N. Ellis, M. Grothe, T. Maeno, C. Padilla, P. Pinto, S. Rosati, C. Santamarina, and P. Werner are with CERN, Geneva, Switzerland.

M. Bosman, P. Casado, M. Dosil, G. Merino, E. Segura, C. Sanchez, and S. Sushkov are with Institut de Física d'Altes Energies (IFAE), Universidad Autónoma de Barcelona, Barcelona, Spain.

B. Caron, R. Moore, J. L. Pinfold, R. Soluk, and S. Wheeler are with the University of Alberta, Edmonton, AB, Canada.

G. Cataldi is with Dipartimento di Fisica, Università di Lecce e INFN, Lecce, Italy.

D. Cavalli and S. Resconi are with Dipartimento di Fisica, Università di Milano e INFN, Milan, Italy.

M. Cervetto, P. Morettini, F. Parodi, and C. Schiavi are with Dipartimento di Fisica, Università di Genova e INFN, Genoa, Italy.

G. Comune, S. Kabana, V. Perez Reale, and E. Thomas are with the Laboratory for High Energy Physics, University of Bern, Switzerland.

A. De Santo, S. George, G. Kilvington, and A. Lowe are with the Department of Physics, Royal Holloway, University of London, Egham, U.K.

M. Diaz Gomez is with Section de Physique, Université de Genève, Switzerland.

B. Epp and V. Ghete are with the Institut für Experimentalphysik der Leopald, Franzens Universität, Innsbruck, Austria.

S. Falciano, L. Luminari, A. di Mattia, and A. Nisati are with Dipartimento di Fisica, Università di Roma "La Sapienza" e INFN, Rome, Italy.

A. Farilla is with Dipartimento di Fisica, Università di Roma "Roma Tre" e INFN, Rome, Italy.

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

A. Khomich is with Lehrstuhl für Informatik V, Universität Mannheim, Germany.

N. Konstantinidis, E. Stefanidis, and M. Sutton are with the Department of Physics and Astronomy, University College London, London, U.K.

A. Kootz is with Fachbereich Physik, Bergische Universität, Wuppertal, Germany.

J. Masik is with the Institute of Physics, Academy of Sciences of the Czech Republic, Prague, Czech Republic.

D. A. Scannicchio and V. Vercesi are with Dipartimento di Fisica Nucleare e Teorica, Università di Pavia e INFN, Pavia, Italy.

N. Nikitin and S. Sivoklokov are with the Institute of Nuclear Physics, Moscow State University, Moscow, Russia.

N. Panikashvili is with the Department of Physics, The Technion—Israel Institute of Technology, Haifa, Israel.

S. Tapprogge is with Institut für Physik, Universität Mainz, Mainz, Germany.

B. Venda Pinto is with CFNUL, Universidade de Lisboa, Faculdade de Ciências, Lisbon, Portugal.

S. Wheeler is with the University of California at Irvine, Irvine, USA.

M. Wielers is with the University of Victoria, Victoria, Canada.

REFERENCES

[1] ATLAS Collaboration, "ATLAS: Technical proposal for a general-purpose pp experiment at the LHC,", CERN/LHCC/94-43, 1994.

[2] ATLAS Collaboration, "ATLAS detector and Physics Performance Technical Design Report,", CERN/LHCC/99-014 and 99-015, 1999.

[3] ATLAS Collaboration, "ATLAS high-level triggers, DAQ and DCS technical proposal,", CERN/LHCC/2000-017, 2000.

[4] ATLAS Collaboration, "ATLAS Level-1 trigger technical design report,", CERN/LHCC/98-014, 1998.
[5] M. Elsing *et al.*, "Analysis and conceptual design of the HLT selection software,", ATLAS Internal Note, ATL-DAQ-2002-013, 2002.
[6] ATHENA: User Guide and Tutorial [Online]. Available: http://ATLAS.web.cern.ch/ATLAS/GROUPS/SOFTWARE/OO/architecture/General/Tech.Doc/Manual/2.0.0-DRAFT/AthenaUserGuide.pdf
[7] C. Bee *et al.*, "Event Handler Requirements,", EDMS Note ATL-DH-ES-0003, 2002.
[8] F. Touchard *et al.*, "Event filter infrastructure validation tests,", EDMS Note ATL-DH-TR-0004, 2003.
[9] S. Wheeler *et al.*, "Test results for the EF supervision,", EDMS Note ATL-DH-TR-0001, 2003.
[10] C. Meessen *et al.*. Protocol specification for event transfer between data collection and event handler. [Online]. Available: https://edms.cern.ch/document/391 570/1.0
[11] S. Wheeler *et al.*, "Large scale tests March 2004,", EDMS Note ATL-D-TR-0001, 2004.
[12] P. C. Muino *et al.*, "Portable gathering system for monitoring and online calibration at ATLAS,", ATLAS Internal Note ATL-DAQ-2004-014.