

Design, deployment and functional tests of the on-line Event Filter for the ATLAS experiment at LHC

A. Negri¹ on behalf of the ATLAS High Level Triggers Group²

Abstract--The Event Filter 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 off-line data reconstruction and analysis steps. The computing instrument of the EF is organized as a set of independent sub-farms, each connected to one output of the Event Builder switch fabric. Each sub-farm 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 specially designed for EF but are adapted from the off-line 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.

¹ Contact address: INFN Sezione di Pavia, via Bassi 6, 27100 Pavia (Italy); e-mail: Andrea.Negri@pv.infn.it

² S. Armstrong^a, A. dos Anjos^b, J.T.M. Baines^c, C.P. Bee^d, M. Biglietti^e, J.A. Bogaerts^f, V. Boisvert^f, M. Bosman^g, B. Caron^h, P. Casado^g, G. Cataldiⁱ, D. Cavalli^j, M. Cervetto^k, G. Comune^l, P. Conde Muino^f, A. De Santo^m, M. Diaz Gomezⁿ, M. Dosi^g, N. Ellis^f, D. Emelianov^c, B. Epp^o, F. Etienne^d, S. Falciano^p, A. Farilla^q, S. George^m, V. Ghete^o, S. González^r, M. Grothe^f, S. Kabana^l, A. Khomich^s, G. Kilvington^m, N. Konstantinidis^t, A. Kootz^u, A. Lowe^m, L. Luminari^p, T. Maeno^f, J. Masik^v, A. di Mattia^p, C. Meessen^d, A.G. Mello^b, G. Merino^g, R. Moore^h, P. Moretti^k, A. Negri^w, N. Nikitin^x, A. Nisati^p, C. Padilla^f, N. Panikashvili^y, F. Parodi^k, V. Perez Reale^l, J.L. Pinfold^h, P. Pinto^f, Z. Qian^d, S. Resconi^l, S. Rosati^f, C. Sanchez^g, C. Santamarina^f, D.A. Scannicchio^v, C. Schiavi^k, E. Segura^z, J.M. de Seixas^b, S. Sivoklokov^x, R. Soluk^h, E. Stefanidis^l, S. Sushkov^g, M. Sutton^z, S. Tapprogge^z, E. Thomas^l, F. Touchard^d, B. Venda Pinto^{aa}, V. Vercesi^m, P. Werner^f, S. Wheeler^{h,bb}, F.J. Wickens^c, W. Wiedenmann^f, M. Wielers^{cc} and G. Zobernig^f

^aBrookhaven National Laboratory (BNL), Upton, New York, USA; ^bUniversidade Federal do Rio de Janeiro, COPPE/EE, Rio de Janeiro, Brazil; ^cRutherford Appleton Laboratory, Chilton, Didcot, UK; ^dCentre de Physique des Particules de Marseille, IN2P3-CNRS-Université d'Aix-Marseille 2, France; ^eUniversity of Michigan, Ann Arbor, Michigan, USA; ^fCERN, Geneva, Switzerland; ^gInstitut de Física d'Altes Energies (IFAE), Universidad Autónoma de Barcelona, Barcelona, Spain; ^hUniversity of Alberta, Edmonton, Canada; ⁱDipartimento di Fisica dell'Università di Lecce e I.N.F.N., Lecce, Italy; ^jDipartimento di Fisica dell'Università di Milano e I.N.F.N., Milan, Italy; ^kDipartimento di Fisica dell'Università di Genova e I.N.F.N., Genoa, Italy; ^lLaboratory for High Energy Physics, University of Bern, Switzerland; ^mDepartment of Physics, Royal Holloway, University of London, Egham, UK; ⁿSection de Physique, Université de Genève, Switzerland; ^oInstitut für Experimentalphysik der Leopold-Franzens Universität, Innsbruck, Austria; ^pDipartimento di Fisica dell'Università di Roma 'La Sapienza' e I.N.F.N., Rome, Italy; ^qDipartimento di Fisica dell'Università di Roma 'Roma Tre' e I.N.F.N., Rome, Italy; ^rDepartment of Physics, University of Wisconsin, Madison, Wisconsin, USA; ^sLehrstuhl für Informatik V, Universität Mannheim, Mannheim, Germany; ^tDepartment of Physics and Astronomy, University College London, London, UK; ^uFachbereich Physik, Bergische Universität Wuppertal, Germany; ^vInstitute of Physics, Academy of Sciences of the Czech Republic, Prague, Czech Republic; ^wDipartimento di Fisica Nucleare e Teorica dell'Università di Pavia e INFN, Pavia, Italy; ^xInstitute of Nuclear Physics, Moscow State University, Moscow, Russia; ^yDepartment of Physics, Technion, Haifa, Israel; ^zInstitut für Physik, Universität Mainz, Mainz, Germany; ^{aa}CFNUL - Universidade de Lisboa, Faculdade de Ciências, Lisbon, Portugal; ^{bb}University of California at Irvine, Irvine, USA; ^{cc}University of Victoria, Victoria, Canada.

I. INTRODUCTION

THE ATLAS (A Toroidal LHC ApparatuS) 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 based on liquid Argon technology. The global detector dimensions (diameter 22 m, length 42 m) are defined by a large air-core muon spectrometer, whose toroidal 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 $10^{34} \text{ cm}^{-2}\text{s}^{-1}$. The corresponding 40 MHz bunch crossing rate (with an average of ~ 23 superimposed events) and the huge amount of detector channel ($\sim 10^8$) outline the challenge of the ATLAS Trigger and Data Acquisition (TDAQ) system.

II. ATLAS TRIGGER DAQ

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, compatibly with a tolerable storage data flux of some hundred MB/s. Given an average event size of ~ 1.5 MB, the storage rate is then about 200 Hz. The required data reduction factor, equivalent to a rejection factor of about 6 orders of magnitude, is achieved on-line via a data acquisition system organized in three different trigger levels (Fig. 1).



design elements because the EF processing resources will evolve during the experiment lifetime and it must be possible to track the technology evolution. Fault tolerance and data security are fundamental requirements for any on-line architecture. Events wrongly rejected are lost forever and code not robust enough is bound to crash often, reducing the livetime of the on-line DAQ. These requirements become even more critical in the ATLAS selection environment, because EF algorithms are not specially developed for on-line environment, but they are inherited from the off-line ones. Therefore, even if particular attention will be paid to ensure the highest possible robustness of the processing code, the EF framework must provide additional reliability in case of algorithm crash in order to avoid biases on the recorded physics sample⁴.

The design of the ATLAS EF system is object oriented, it is fully implemented in C++ and it uses multi-thread programming techniques. The multi-threaded 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 between event processing operations and data flow functionalities;
- data driven event flow⁵ without data copying inside the processing host;
- local storage based on memory mapped files for event recovery capability;
- exploitation of SMP architectures.

A. System wide 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 sub-farm and software or hardware failures in a processing node do not affect the operation of other sub-farm elements. Similarly full sub-farms can be easily hot-plugged in a running system. Furthermore the design supports geographically distributed implementations and, depending of the network topology, dynamic re-routing in case of SFI malfunction or crash.

B. Processing node architecture

In order to ensure data security and fault tolerance, in each node the data processing operations are completely decoupled from the data flow functionalities. The latter ones are provided by the Event Filter Dataflow process (EFD) that manages the communication with 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 by separated processes running the EF algorithms in the standard ATLAS off-line framework [6]. The events are made available to PTs via a shared memory (called SharedHeap), which stores the events during their transit in the processing node. Communication and synchronization between EFD and PTs is maintained via messages exchanged on a UNIX domain socket.

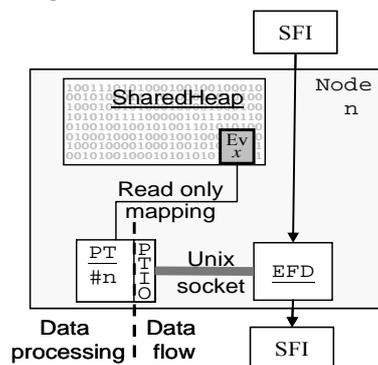


Fig. 2. EFD design: decoupling between data processing operations and data flow functionalities.

The EFD function is divided into different specific tasks that could be dynamically interconnected to form a fully configurable EF dataflow network (Fig. 3 shows an example of internal data flow implementation). Tasks are daisy chained, with each task knowing the identity of the next task to be executed. Therefore the worker thread, starting execution from the first task, drives the event reference through the internal dataflow network. The event and task references (as all other object pointers used in the EFD) are implemented as smart pointers providing garbage collection facilities.

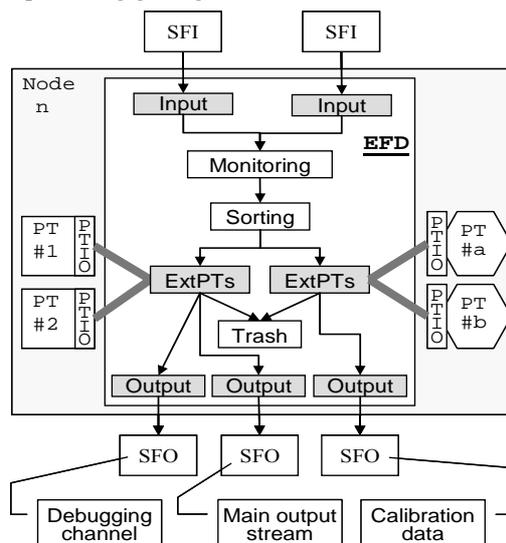


Fig. 3. Implementation example of internal EFD data flow. The figure shows several task types and many possible event paths. The PTs on the left are processes running selection software, while the ones on the right have calibration purpose.

⁴ Some crashes could be related to specific event topology and the loss of these events could invalidate the physics results.

⁵ i.e. no data flow manager to assign the event to a specified target.

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 part 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. Because the map is read only, the PT cannot corrupt the event or the SharedHeap structure. PT problems are reliably handled by the EFD that 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 SFO.

The processing operation produces a filtering decision and a selection object (used to classify the event and guide the off-line analysis step) that are communicated back to the EFD. The selection object is stored in event header, while the filtering decision is used to steer the internal dataflow. Referring to Fig. 2, the event selection PTs can sort e.g. 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.

The processing operation can provide some additional reconstructed information, which could 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 will take care of appending this data fragment to the raw event

The SharedHeap is implemented as a memory mapped file hosted in 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 dimensions 2^n bytes⁶. 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 manages directly the actual write operations avoiding useless disk I/O over-heading. The system could be out of sync only in case of power cut, OS crash or disk failure. However these

⁶ The limited spectrum of possible block sizes is compensated by the algorithm simplicity and performance, which allow an efficient exploitation of the allocated storage space.

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-bed of different sizes using special SFIs, which emulate the event building behavior. They read recorded events from file and make them available to the EFDs. Details of these tests are given in [8]-[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 was run for 10 days (more than 3×10^9 events handled) without problems or event losses. The test included the repeated random kill of PTs and the subsequent re-start of the processes. The EFD handled correctly the situation recovering the events owned by the killed PT and assigning them to a different PT. No event loss has been observed.

It has also been tested that the 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; un-map the event. Ignoring the event processing step, the measured time to perform this sequence is about 80 μ s on a dual processor Xeon at 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 that 10 KHz.

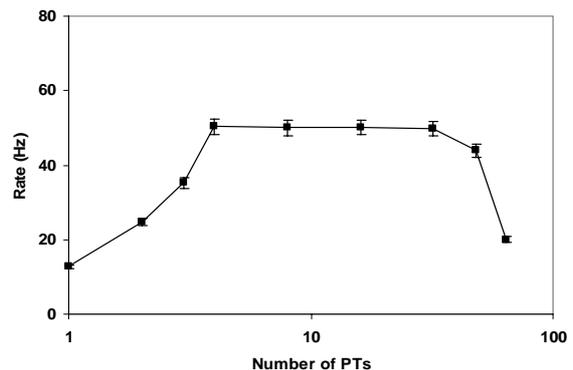


Fig. 4. Architecture scalability. Node processing rate as a function of the number of processing tasks. The PTs run a muon track reconstruction algorithm on data recorded at test beam. The hardware is a SMP machine with 4 Xeon CPUs clocked at 3.2 GHz and 4 GB of main memory.

The architecture proved to be scalable in term of PTs and showed the correct exploitation of 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 EFD and the adjacent data flow elements (SFI and SFO) [10]. For realistic event sizes the protocol reasonably exploits the gigabit link.

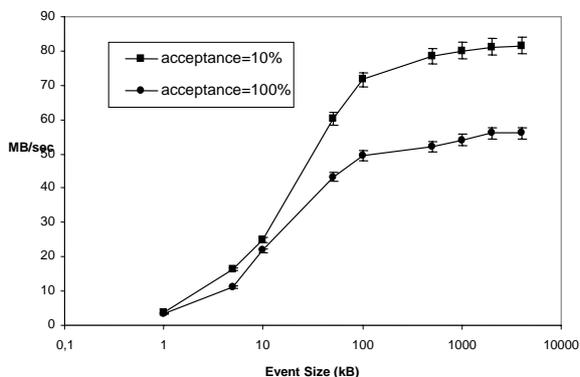


Fig. 5. Node throughput as a function of the event size for two different values of the PT acceptance.

The maximum throughput, about 30 events/s, is adequate for the host requirements: ~ 1 events/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.

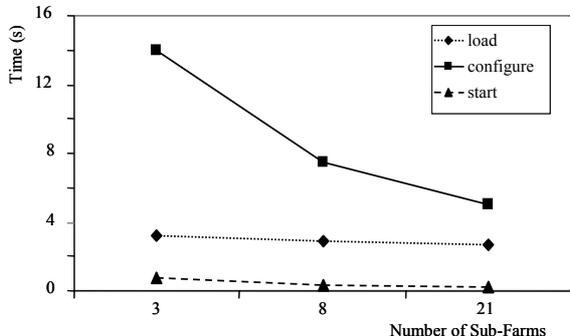


Fig. 6. State transition times as a function of the number of subfarm for a constant full farm size of 230 nodes.

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). The tests, even if mainly focused on the HLT control architecture, confirmed the reliability of the design. Configurations with up to 21 sub-farms, ~ 300 EFDs and ~ 16000 PT have been run without problems related to the EF infrastructure (Fig. 6).

VI. DEPLOYMENT

The EF system implementation has been also deployed in test-beam environments: from summer 2002 it is an integral part of the ATLAS test-beam data acquisition chain. The EF is currently used in 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 sub-farm (4 dual Xeon PCs 3.2 GHz, 1 GB of memory) and a sub-farm located few kilometers away from the test-beam site (20 dual Xeon 600 MHz).

During the test beam, geographically distributed sub-farm implementations have been tested too. Remote farms located in Canada and Poland has been integrated in the running data acquisition system. Even if the rate was limited to few hertz due to the above described communication protocol implementation, events have been correctly transferred to remote processing resources (nodes geographically distributed running an EFD instance and some PTs) and received back at the SFO level.

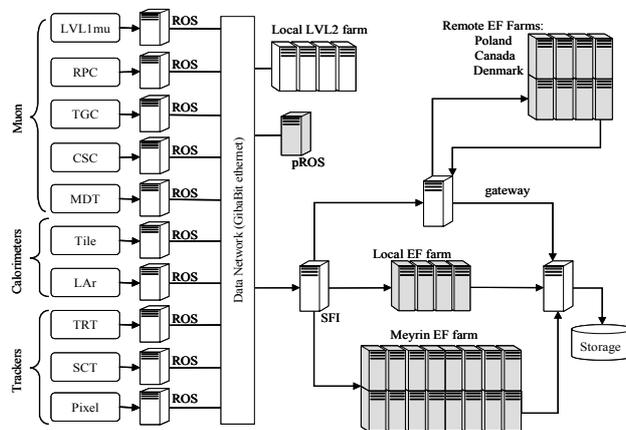


Fig. 7. ATLAS combined test beam setup. It comprises almost all the detector types, the LVL2 farm and some EF farms.

Owing that off-line and on-line environments share the same development and running framework, the integration of reconstruction and monitoring algorithms in the EF system is transparent: any off-line algorithm properly configured for the test-beam setup (suitable geometry description and data decoding components) can be run in the EF framework. Therefore many detectors monitoring algorithms have been integrated in the running system. The histograms produced by

the different PTs are collected and merged by a dedicated on-line element [12] and are used to online 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).

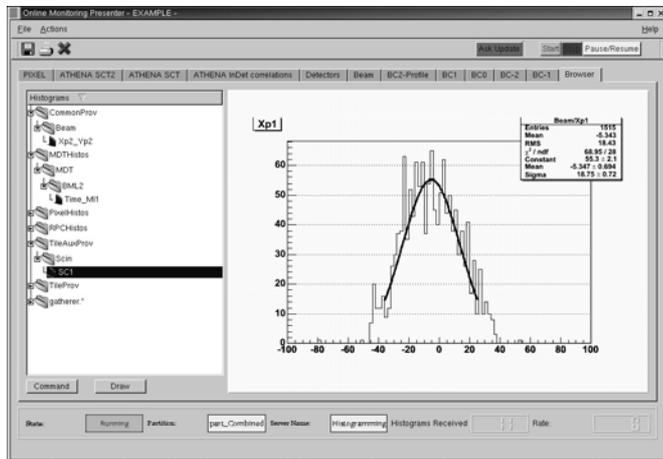


Fig. 8. Example of data monitoring histogram produced online and showed by the Presenter.

The simultaneous presence of all the ATLAS trigger levels and, in particular, the integration in LVL2 and EF of the HLT muon reconstruction and selection algorithms allowed the on-line validation of the full muon selection slice. In particular it was verified the transfer of the LVL2 result (contained in the PROS) to the EF and its decoding. The LVL2 result is then used by the HLT Steering element to call the correct EF algorithm for the given event.

The use of simple event selection rules allowed testing 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. CONCLUSIONS

In this paper we have described the design of the ATLAS EF system, the test-bed validation of the implementation and

its deployment in the ATLAS combined test beam. The object oriented design relies on multi-thread programming techniques and fulfills the scalability and modularity requirements.

The 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 based on memory mapped file provides an elegant event recovery mechanism.

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

VIII. 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." 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," EDMS Note, <https://edms.cern.ch/document/391570/1.0>
- [11] S. Wheeler et al., "Large scale tests March 2004", EDMS note, ATL-D-TR-0001, 2004
- [12] P. Conde Muino et al., "Portable Gathering System for Monitoring and Online Calibration at ATLAS," ATLAS Internal Note, ATL-DAQ-2004-014