PROCEEDINGS OF SCIENCE

ATLAS Remote Online Farms

Sander Klous[*†]
Nikhef, PO Box 41882, 1009 DB Amsterdam, The Netherlands
E-mail: sander@nikhef.nl

H.P. Beck, Laboratorium fuer Hochenergiephysik, Universitaet Bern, Switzerland
C. Bee, Centre de Physique des Particules de Marseille, Marseille, France
B. Caron, Department of Physics, Faculty of Science, University of Alberta, Canada
H. Garitaonandia and S. Sushkov, IFAE, Universidad Autonoma de Barcelona, Spain
R. Hughes Jones, School of Physics and Astronomy, University of Manchester, UK
K. Korcyl, Henryk Niewodniczanski Inst. Nuclear Physics, Krakow, Poland
B. Martin, CERN, Geneva, Switzerland
J. Vermeulen, Nikhef, Amsterdam, The Netherlands
A. Negri, S.N. Stancu[‡]*and S. Wheeler,*
Department of Physics and Astronomy, University of California at Irvine, USA

The ATLAS collaboration is investigating the possibility to extend its online computing system with Remote Online Farms. These farms, accessed over a wide area network, would assist with the quasi real-time analysis of events for detector monitoring and calibration purposes. The deferral of the online calibration tasks to Remote Online Farms would relieve this task from the on-site online computing and selection infrastructure, thus allowing it to concentrate on event selection. In this paper, status and plans of this project are presented.

XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research
April 23-27 2007
Amsterdam, the Netherlands

[*]Speaker.
[†]Acknowledgements: Nikhef PDP group and ATLAS HLT Routing and Streaming working group
[‡]Also at University Politehnica, Bucharest, Romania

PoS(ACAT)021

1. Introduction

The Large Hadron Collider (LHC), currently under construction at CERN in Geneva, is a 27-kilometer-circumference double synchrotron to accelerate bunches of protons in opposite directions [17]. The bunches cross at four different interaction points, resulting in head-on collisions with a center-of-mass energy of 14 TeV. The bunch-crossing frequency is 40 MHz and on average 23 proton interactions per bunch crossing are expected at the nominal design luminosity of $10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. Around the four interaction points, huge experiments are constructed to detect the remnants of the collisions. Only a small part of the enormous amount of information collected by these experiments can be stored due to technical and economic limitations. Advanced real-time selection systems, known as trigger systems, are installed by all experiments to select the most interesting information for storage and further study.

ATLAS, the largest LHC experiment, is a 4π general purpose detector, *i.e.* it is designed to cover as much phase space around the interaction point as possible. On each bunch crossing, it records the signals induced by particles traversing the detector systems. At such an event a total of 1.5 MB of data are collected to determine *e.g.* the type and energy of the detected particles. This sets the scale for the trigger system, since available storage space is limited to about 6 PB per year [15] and the ATLAS operational period is about $2 \cdot 10^7$ s/year [7]. As a result, the recording frequency is roughly 200 Hz.

The ATLAS collaboration has designed a multilevel trigger system to select the most interesting events, a schematic diagram is presented in Fig. 1. The first level trigger (LVL1) [6] is implemented in custom hardware and reduces the event rate from the bunch crossing rate of 40 MHz to a design value of 75 kHz, upgradable to 100 kHz. The second trigger level (LVL2) [8] is software based and runs on a cluster of computers interconnected with high performance Ethernet. The LVL2 selection algorithms request partial event data stored in read out buffers (ROBs) from physical regions of the detector identified by LVL1. The output rate of LVL2 is about 3 kHz, which makes it feasible to send full event information to the last stage of the trigger system, the "Event Filter" (EF) [8]. Full event information is collected, assembled and made available to the EF by about 100 Sub Farm Inputs (SFIs). The EF baseline configuration consists of a cluster of 1800 quad core dual CPU nodes with a clock frequency above 2 GHz. So, the 3 kHz output rate of LVL2 results in an average processing time limit of $4\,s$ per event per core. The Sub Farm Output (SFO) is the final element of the online system and serves as a proxy and/or gateway to the mass storage facility. Later in this document, the network in the left column of Fig. 1 is called DataFlow network. The LVL2 and the EF are collectively known as Higher Level Trigger (HLT).

EF nodes act as clients, requesting events from the SFIs and acknowledging successful transfer as soon as a full event arrived. This provides an interesting opportunity to acquire additional EF processing capacity on geographically distributed locations. The SFIs can forward events to any available EF node, independent of its location. This node could either be on the local farm, on a dedicated Remote Online Farm (ROF) or on general purpose grid resources. Of course, this kind of heterogeneity adds additional complexity to an online system. The issues arising from this additional complexity are discussed in this paper.

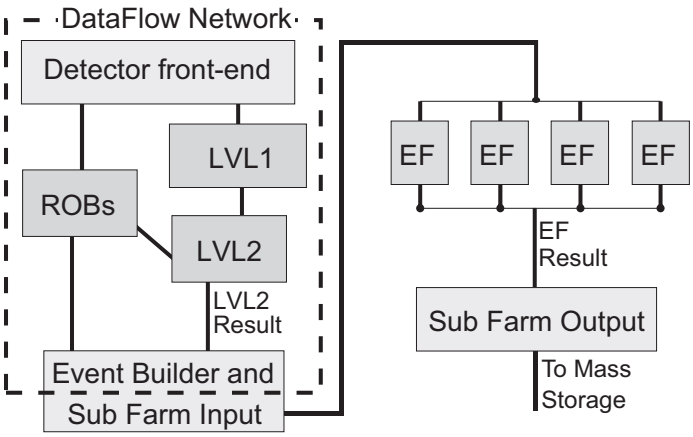

Figure 1: Schematic diagram of the ATLAS trigger system.

2. Event categorization

The EF is designed with the assumption that the EF nodes can handle all events, irrespective of their type [16]. The SFIs handle event requests (GET_EVENT) from the EF nodes solely based on a first in first out principle (FIFO). With the introduction of remote farms, we have to revisit this principle. Obviously, the most valuable event types should be handled by the most reliable resources. Hence, a hierarchy arises where events are preferably handled by the local cluster, followed by dedicated ROFs and eventually grid-based resources. In practice this means that events should be classified according to their type. The trigger system could route events to the best available resources, based on this event type.

Recently the ATLAS event data format [3] has been extended with features that allow writing information of events into different offline streams based on the content, *e.g.* events selected by a jet algorithm will end up in a different stream than events selected by a muon algorithm [24]. These features not only facilitate offline analysis, but allow to prioritize different event categories in the online system as well. Specifically, we are able to distinguish detector calibration events from the rest and route them to ROFs. An overview of this functionality is shown in Fig. 2.

Events can be routed to an ROF for EF processing at 2 different stages. The most efficient mode of operation (in terms of networking and CPU resources) would be to send LVL2 accepted events directly to an ROF (indicated in the figure with Route 1). This could be the default configuration in case of a well tested and stable infrastructure, but we need to implement additional routing functionality in the SFI. Alternatively, a promiscuous mode will be available (Route 2), where all LVL2 accepted events pass through the local EF farm for inspection. Compared to Route 1, this puts a small additional load on the EF nodes and increases the bandwidth requirements of the EF network. This network is able to handle the LVL2 output rate of 3 kHz, but as shown in Fig. 2, traffic from EF to ROF flows mostly over the same lines and switches. Since this traffic runs in opposite direction and sufficient margin exists in the concentrator switches and core network, we expect only a marginal impact of this extra load on the network performance. Preliminary tests done with an increased EF output rate support this conclusion, although some tuning of the switch buffers was required.

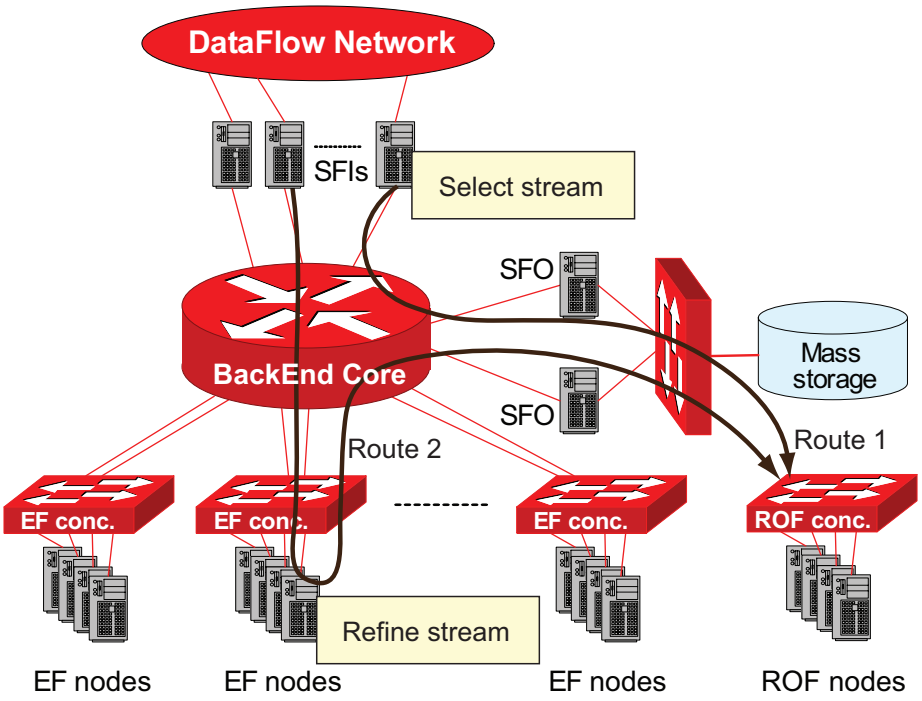

Figure 2: Overview of streaming functionality. The BackEnd Core router provides network connections of the HLT. Traffic to and from the EF nodes is bundled per rack via concentrator switches (EF conc.).

At the moment, ATLAS is working on a proof of concept environment for the exploitation of ROFs. In this phase, the feasibility of ROFs will be demonstrated with selection of events needed for detector calibration. The payload of these so-called calibration events only contains partial information, *e.g.* of a single subdetector or a specific region in phase space. Hence, they are considered less valuable than physics events, making them the ideal test case for this new technology.

3. Communication protocol

On top of the event-format changes implemented for event routing and streaming, additional features are required for the handling of calibration events. These events are a lot smaller than full events (they can be as small as 100 kB), which changes the dynamics of the EF buffer management and makes it difficult to transport them efficiently, especially over long distances (high latency connections). The communication protocol between an SFI and a single EF node is serial in nature (the EFIO protocol) [11], *i.e.* a new event is only requested after the previous one has been received. This is no problem as long as the average transport time of an event is large compared to the latency, as is the aim on a local cluster with full events (and properly configured switch buffers). In our case, we have to deal with small partial events transported over long distance connections, which results in a latency comparable to or larger than the average transport time. This reduces the throughput considerably, even on local area network connections, as shown in the first row of Table 1.

We made two modifications in the communication between LVL2 and EF to accommodate for the transfer of partial events over long distance connections. First of all, the client-server

Table 1: Latency effects in the EFIO protocol for small events on a 1 Gbps local area network

Number of parallel connections	Throughput for 1.5 MB events [MB/s]	Throughput for 20 kB events [MB/s]
1	115.4 ± 0.2	52.80 ± 0.07
2	118.3 ± 0.1	93 ± 2
3	118.6 ± 0.2	117.1 ± 0.1
4	118.6 ± 0.2	117.19 ± 0.02

architecture between EF and LVL2 is exploited to establish multiple (parallel) connections from a single EF node to a single SFI. Each EF node establishes one or more connections with an SFI. Specifically, on the local cluster only 1 connection is made, which results in the classical serial event request behavior. But for ROFs, the number of connections can be much larger than 1 to facilitate parallel requests. The last row in Table 1 demonstrates that we can fully exploit the available bandwidth with parallel requests, even when the event size is reduced to 20 kB.

We also improved the scheduling of event requests, to make sure that they are evenly distributed in time. This reduces the risk to overload the SFIs with a burst of event requests and allows us to stabilize the occupancy of the EF buffer. A simple proportional controller schedules the requests based on the occupancy of the EF buffer, *i.e.* a higher buffer occupancy is linearly translated into a lower event request rate. The results of our modifications are demonstrated in Fig. 3, where the buffer occupancy is plotted as a function of time. The old on/off control mechanism is shown on the left hand side and the proportional controller on the right. Note that the dots, reflecting event requests, come in bursts in the old situation and are evenly distributed with the new mechanism. Still, on the server side, SFIs handle simultaneous event requests from different EF nodes with non-blocking I/O (nothing changed in this respect).

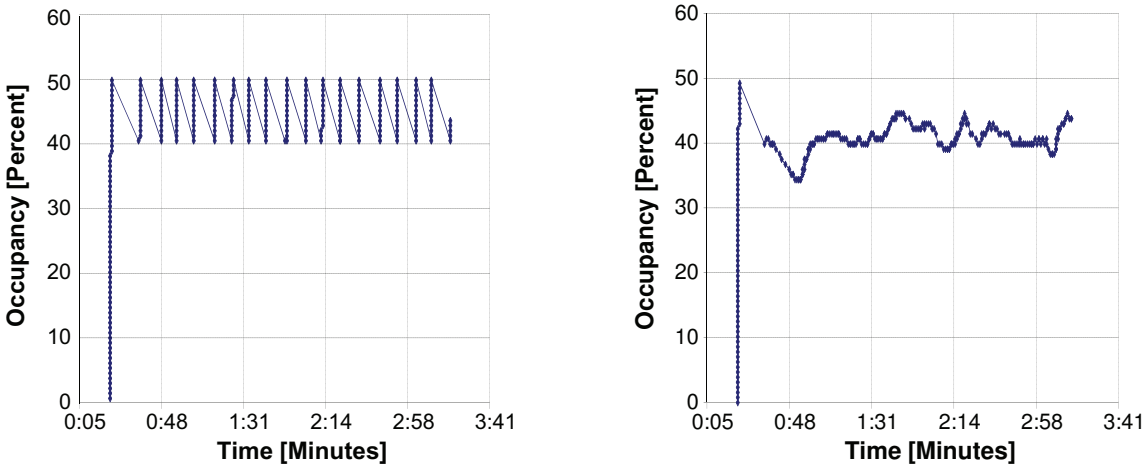

Figure 3: Event buffer occupancy as a function of time. On the left hand side, the old on/off mechanism results in bursts of event requests (reflected by the distribution of dots). On the right hand side, the new proportional controller provides evenly distributed requests and stabilizes the buffer occupancy.

The new control mechanism monitors both the EF buffer occupancy as well as the throughput. This not only facilitates ROF operations, it also allows us to maintain a relatively low buffer occupancy. As a result, the EF nodes are able to clear their buffers within the specified timeouts when the system is shut down.

4. Infrastructure and Security

We plan to do a number of tests of gradually increasing complexity to demonstrate the ability to securely and reliably include an ROF in the trigger system. Figure 4 shows the relevant parts of the current CERN network infrastructure. The ATLAS trigger system is part of the box on the left upper side (Experiments / CPUs, disks, tapes). The output of the trigger system is written to the CERN storage element, located in the box on the left hand side, called "LCG Backbone". Next, data is distributed to large offline computing centers, known as Tier-1 centers (on the left bottom side), through direct lightpath connections. The sustained data rate from raw-data recording activities is in the order of 100 MB/s per Tier-1 center [15], which is 50% of the peak capacity. No firewalls are present on these links, since the Tier-1 centers are trusted sites with signed agreements on service and security levels.

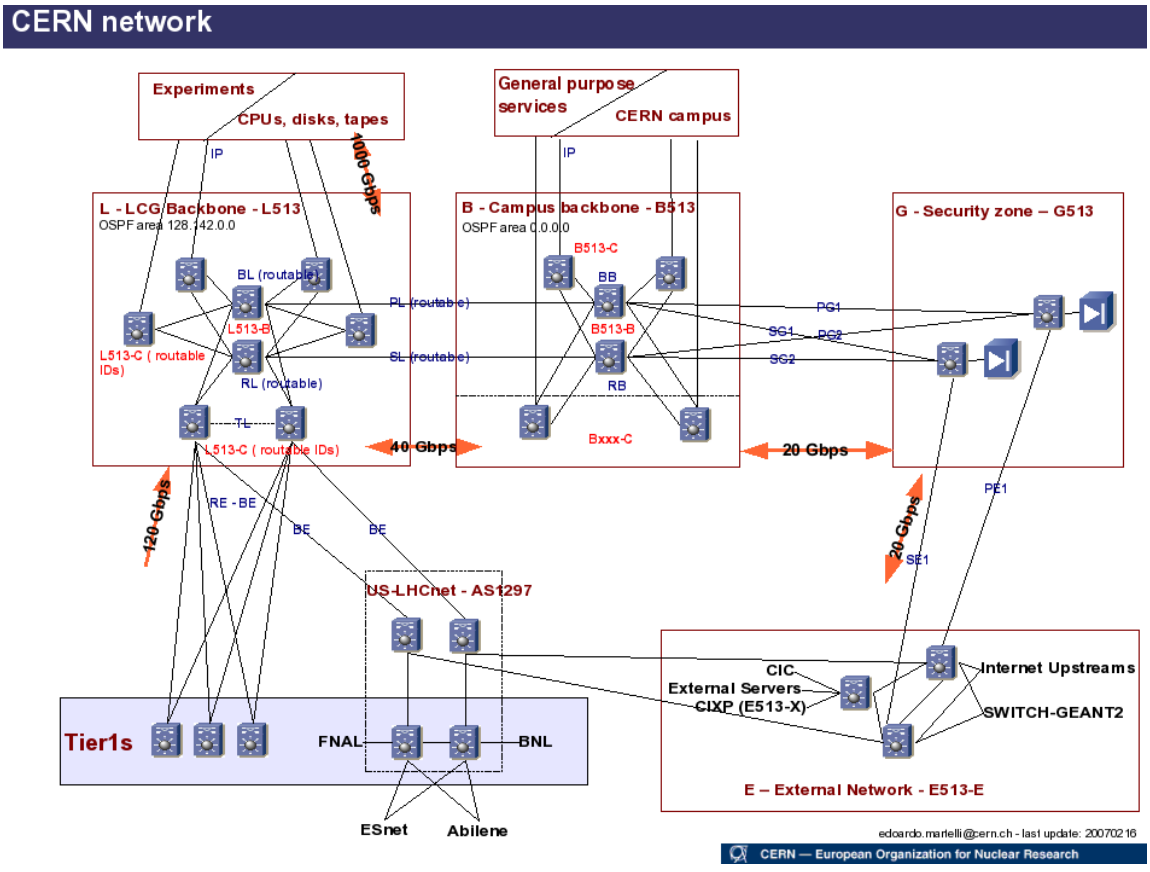

Figure 4: CERN network infrastructure

The ROFs are connected in the lower right corner either as "Internet Upstreams" or via the Geant2 research network. In contrast to the Tier-1 sites they are not considered as trusted network infrastructure. Hence, their connection to the ATLAS trigger system has to run through the CERN firewall, shown on the right hand side of the picture with the name "Security Zone". Next, the links pass through the CERN local area network (the Campus Backbone, in the middle of the picture) and terminate at an application gateway with access to the ATLAS trigger system. The available bandwidth for ROFs during the proof of concept phase is limited to about 1 Gb/s per connection, due to firewall scalability issues.

In the first tests, we will deploy the EF system on dedicated ROFs connected to CERN with lightpaths. Basic Layer 2 and 3 Access Control Lists (ACLs) are implemented at CERN and on the remotes site to avoid interference from non-ROF traffic. A trust relation between the nodes will be established through Public Key Infrastructure (PKI) authentication with X509 certificates [22].

All required software components for the online system and their locations are described in the DAQ configuration database (ConfDB) [12]. The consistency of the descriptions in the ConfDB should not be affected by the intrinsically dynamic Wide Area Network structure between CERN and the ROFs, *i.e.* we do not want to update the ConfDB description when the network topology between CERN and an ROF (node) changes. Instead, the application gateway will serve as a proxy to allow dynamical subscription of registered ROF nodes.

ROF nodes included in the online system behave just like other EF nodes, *e.g.* they will retrieve information about the trigger selection logic and the trigger setup from the trigger configuration database (TriggerDB) as described in [9], produce near real-time status reports in the Information System [8] and respect the state transitions of the run control application (start, stop, shutdown, etc.). Note that the overhead on the wide area network can be reduced with the installation of a database proxy (ProxyDB) on each of the remote sites. The same solution is already implemented on the EF cluster at CERN to reduce the load on the TriggerDB server.

It is still unclear where ROFs will store their accepted events. From a technical point of view, it makes sense to write these events to the closest available mass storage device. Most likely, for ROFs this will not be at CERN, but rather at one of the Tier-1 centers. As a result we lose the single entry point for raw data at CERN, which complicates data management issues in the offline analysis. Alternatively the accepted events could be fed back to CERN, either into the online system or into CERN mass storage via an SFO serving as proxy on the ROF. This would avoid data management problems at the cost of some additional network traffic (in opposite direction) and an additional load on the CERN mass storage system. Probably, this is an acceptable drawback as long as the accept rate is much lower than the input rate and the number of ROF nodes is small compared to the number of EF nodes. In any case, different approaches will be investigated to compare the performance.

5. Data Provenance and Fault Tolerance

The complexity of experimental techniques and data analysis methods continues to increase in High Energy Physics. Thus, the importance of data provenance (*i.e.* traceability and validity of the processing chain) grows accordingly [4]. In an offline environment, the emphasis is naturally put on reproducibility, *i.e.* a physicist has to make sure that others are able to repeat the analysis

and verify the results. In an online environment this approach is only partially feasible because rejected events will not be stored. Of course, this argument holds for any online system and is not specific for ROFs. Still, tracking the data flow is even more difficult in a distributed environment, as pointed out by [18].

In a production environment, the integrity of each operational ROF node has to be guaranteed. On top of this, real-time applications put stringent requirements on version management. At any given moment in time, only one specific version of the software may process the data on a limited number of different architectures. With these two requirements in mind, we turn to virtualization techniques (*e.g.* Xen, KVM or VMWare) to simplify our software deployment policies. An automated process should produce an up to date virtual machine snapshot of an Event Filter node to keep the system synchronized. This snapshot is validated at CERN together with the rest of the EF system in technical and commissioning runs. After validation, the virtual machine is approved (*e.g.* by providing a valid host certificate), which guarantees its consistency with the other EF nodes. Only a single virtual machine has to be validated, which is a clear advantage when scalability becomes an issue.

Once a certified virtual machine is installed at an ROF, proper operation is verified with an automated online test suite [2]. Contact with the application gateway is only made after all appropriate tests are passed successfully. The application gateway will only accept certified virtual machines run by a certified account, providing a single point of control for the shift crew and dynamic integration with the rest of the online infrastructure. Note that peer to peer distribution of virtual machines to many worker nodes in parallel can become problematic on a grid infrastructure. Effective network strategies have to be investigated in a later stage of the project, to avoid consumption of significant bandwidth. In this context, reliable multicast [20] and the globus workspace service [13] might be interesting developments.

Other complications of ROF deployment in a grid environment include (but are not limited to) resource acquisition and accounting. The LHC computing grid infrastructure has a large number of users. All of them have different kinds of shares, priorities and policies. Nevertheless, a real-time application requires a fairly stable resource pool, including the possibility for advanced planning and scheduling. This functionality is non-existing at the moment and needs to be developed. For an extensive overview of real-time aspects in grid computing, see [21]. An initial effort to setup and run the ATLAS trigger and data acquisition system on grid resources is provided by GRIDtools [10], which is an excellent starting point for further research.

Real-time applications are fundamentally different from other applications in their fault tolerance requirements. Unscheduled down time of ROF nodes directly reduces the event handling rate. As a result, the processing capacity could become insufficient, which leads to buffer overflows and loss of events. Still, performance should largely be independent of failures in any individual component. This can be achieved as long as ROFs are sufficiently overprovisioned, preferably at geographically and logically separated locations. Note that overprovisioning with ROFs provides an interesting opportunity: the ROFs could temporarily replace each other or even part of the EF farm at CERN, *e.g.* during maintenance operations. The client-server architecture provides a self-managing system with natural load balancing when a sufficient number of ROF nodes is available.

The required overprovisioning ratio depends on the reliability of the EF and the ROFs, determined by the Mean Time Between Failures (MTBF) and Mean Time To Repair (MTTR). Let's

take the EF as a reference: the average availability $\left(\frac{MTBF-MTTR}{MTBF}\right)$ of an EF node is expected to be above 99.9% [5]. We start with an oversimplified model where correlated failures are not taken into account. Fig. 5 shows the relative failure probability distribution for a farm of 2000 nodes. Reliability is not really an issue in this case, overprovisioning in the order of 1% is more than enough. Average algorithm execution times are not even known with such an accuracy.

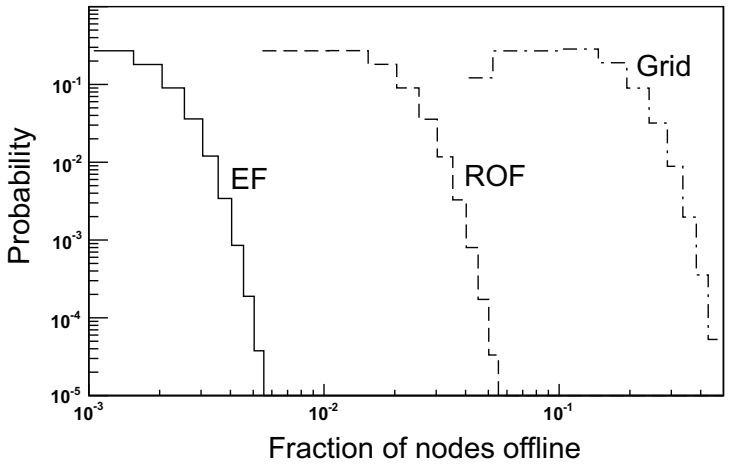

Figure 5: Reliability of the EF and the ROFs. The histogram shows the distributions for a farm of 2000 nodes (EF), 200 nodes (ROF) and 20 nodes (Grid) with the corresponding availability numbers as presented in the text

Now suppose we want to add a dedicated ROF of 200 nodes to provide 10% of the EF capacity. The availability of a dedicated ROF is most likely above 99%, mainly limited by network failures [23] (assuming sufficient redundancy in the firewall configuration). As a result, the relative failure probability distribution is an order of magnitude higher (the dashed line in Fig. 5). Overprovisioning in the order of 10% (20 nodes, or better: 1 extra site for each 10 ROFs) is required to obtain the same reliability as the EF. In case grid resources are used, we estimate the availability at about 90% [19]. This number does not include job submission efficiencies and assumes the node has successfully been registered with the application gateway. Hence, we need to increase the overprovisioning to about 100% (see dash-dotted line). So, as a rule of thumb, 40 grid worker nodes (preferably spread over different sites) can handle an event input rate of 30 Hz with similar reliability as the EF running on 3 kHz.

Of course our initial assumption of uncorrelated failures is wrong, especially network errors are highly correlated in case of ROF operation. These correlations can be compensated with proper buffers and fail-over capacity. Hence, an ROF should first be deployed without reliability requirements, *e.g.* to test new versions of trigger algorithms in an online environment (new release testing). During this phase, the size and location of buffers as well as the required fail-over facilities can be determined. The ROF should only be included in a production configuration after reliable operation has been demonstrated.

Failure recovery is another important aspect in the fault tolerance of online event selection. Unaccounted events, accepted or rejected, are affecting the statistics in any subsequent analysis.

Hence, we should avoid the loss of events due to hardware and/or software failures. The SharedHeap, a memory mapped file, plays a central role in the EF failure recovery mechanism [1]. Each EF node contains a SharedHeap where events are stored from the moment of arrival until the moment of removal. An event will only be removed from the SharedHeap when it is rejected or when it is accepted and successfully written to mass storage. No events are lost, as long as the SharedHeap can be recovered, this holds both for EF nodes and for ROF nodes. An additional complication arises when grid resources are acquired as ROF nodes. It is not trivial to access the SharedHeap on a grid worker node after a job failure. On many sites, such files are automatically removed. Maybe the SharedHeap can be written on the network disk of a dedicated experiment machine (known as a VOBox), but we need to investigate both the performance aspects as well as the VOBox management implications of such a solution.

The recovery procedure discussed above does not help in case of unrecoverable disk failures. Hence, each system component contains a number of event counters to detect such failures [14]. EF nodes as well as ROF nodes register *e.g.* the number of accepted and rejected events. These numbers are compared with the SFI and mass storage counters. An inconsistency would indicate that events are lost (or duplicated). Affected data samples are flagged automatically and will be excluded from any cross-section analysis unless it is proven that losses can either be neglected or corrected.

6. Conclusions

The design of the EF is well suited for acquisition of additional resources over wide area networks in geographically distributed locations. Furthermore, we are able to prioritize different event categories in the online system with the help of the newest version of the ATLAS event data format. Hence, events can be routed to the EF, a dedicated ROF, or grid based resources based on their content (*e.g.* calibration or physics). Two modes of ROF operation will be provided, an efficient mode where events are directly routed to the correct destination, and a promiscuous mode where the EF inspects all events before they are routed.

The communication protocol between SFIs and EF nodes has been updated to deal with small events (*e.g.* for calibration). Now, the full network capacity can be exploited for events as small as 20 kB. We demonstrated that the new protocol is able to maintain a stable buffer level in the EF with evenly distributed event requests to the SFIs. As an additional advantage we now monitor both the occupancy and the throughput, so we can operate the EF with a lower buffer occupancy. This makes it more likely that the EF can clear its buffers within the specified timeouts when the system is shut down.

ROFs will be connected to an application gateway with access to the ATLAS trigger system. This gateway serves as a proxy to allow dynamical subscription of ROF nodes. Dedicated ROFs can be connected with lightpaths, where basic Layer 2 and 3 ACLs avoid interference from non-ROF traffic. A trust relation between the nodes should be established through PKI authentication with X509 certificates. ROF nodes included in the online system behave just like normal EF nodes (with respect to configuration and logging). It is still unclear where ROFs will store their accepted events, either at CERN or at the closest Tier-1 center. The first option is easier from the data management perspective, the second is more efficient. Both options will be investigated.

The integrity and consistency of ROF nodes (hardware and software) could be guaranteed with certified virtual machines. In case of grid resources, complications are expected with the distribution of these virtual machines. New developments in grid middleware and network communication might provide a solution, but this has to be investigated in a later stage of the project.

ROFs have to be overprovisioned to provide similar reliability as the EF. As a rule of thumb, the overprovisioning of dedicated ROFs should be about 10%. Preferably for every 10 ROFs an extra site should be available, to compensate for correlated network failures as much as possible. Similarly, 40 nodes spread over different grid sites can handle about 30 Hz with the same reliability (in other words: 100% overprovisioning is required). Proper buffers and fail-over capacity have to be installed to compensate for remaining error correlations. Hence, an intermediate phase is proposed before a new ROF is included in the production environment. During this phase, the new ROF should run non-critical trigger algorithms (*e.g.* to test new versions) without reliability requirements, to determine appropriate buffers and fail-over facilities.

The failure recovery mechanisms of ROFs are the same as those for the EF. No events are lost as long as the buffer, a memory mapped file called SharedHeap, can be recovered. On grid resources, it might not be straight forward to recover this file after a job failure. Maybe the SharedHeap can be written on the network disk of a dedicated experiment machine (known as the VOBox). Performance and VOBox management aspects of such a solution need to be investigated. Unrecoverable events are detected with redundant event counters. Affected data samples are flagged automatically and if necessary excluded from subsequent analysis.

References

[1] Christophe Meessen Andrea Negri. Sharedheap. Technical Report v2, CERN, 2003. https://twiki.cern.ch/twiki/bin/viewfile/Atlas/EventFilterDataflow/EventFilterDataFlow.

[2] S. Backlund. Hltonlinetests. https://twiki.cern.ch/twiki/bin/view/Atlas/HLTOnlineTests.

[3] L. Mapelli R. McLaren G. Mornacchi J. Petersen F.Wickens C. Bee, D. Francis. The raw event format in the atlas trigger and daq. Technical Report ATL-D-ES-0019, ATL-DAQ-98-129, CERN, 2005.

[4] R. Cavanaugh. Satisfying the tax collector: Using data provenance as a way to audit data analyses in high energy physics. *Workshop on Data Derivation and Provenance*, Position Papers:1–4, 2002. http://people.cs.uchicago.edu/ yongzh/papers/TAXMan.ps.

[5] CERN Computing Center. Batch system statistics. https://lemonweb.cern.ch.

[6] The Atlas Collaboration. Atlas level-1 trigger: Technical design report. Technical Report CERN/LHCC/98-14, CERN, 1998.

[7] The ATLAS Collaboration. Atlas detector and physics performance. Technical Report CERN/LHCC/99-14, CERN/LHCC/99-15, CERN, 1999.

[8] The ATLAS Collaboration. Atlas high-level trigger, data-acquisition and controls: Technical design report. Technical Report CERN/LHCC/2003-022, CERN, 2003.

[9] A. dos Anjos et al. A configuration system for the atlas trigger. *JINST*, 1:05004, 2006.

[10] H. Garitaonandia et al. Using the grid to test the atlas trigger and data acquisition system at large scale. *IEEE TNS*, 54(5):0, 2007.

[11] HP. Beck et al. Efio: Protocol specification. Technical Report ATL-DQ-ES-0040, CERN, 2006. https://edms.cern.ch/document/391570.

[12] Igor Soloviev et al. Configuration database user's guide. Technical report, CERN, 2002. http://atlas-onlsw.web.cern.ch/Atlas-ug/2.3/pdf/ConfDB.pdf.

[13] K. Keahey et al. Virtual workspaces for scientific applications. In *SciDAC Conference Proceedings*, 2007. http://workspace.globus.org/papers.

[14] M. Shapiro et al. Report from the luminosity task force. Technical Report gen-pub-2006-002.pdf, CERN, 2006. http://cdsweb.cern.ch/record/970678.

[15] R. Jones et al. The atlas computing model. Technical Report CERN-LHCC-2004-037/G-085, CERN, 2004.

[16] S. Armstrong et al. Design, deployment and functional tests of the online event filter for the atlas experiment at lhc. *IEEE Transactions on Nuclear Science*, 52:2846–2852, 2005.

[17] Y. Baconnier et al. The large hadron collider accelerator project. Technical Report CERN/AC/93-03, CERN, 1993.

[18] Y. Simmhan et al. A survey of data provenance in e-science. *SIGMOD Record*, 34:31–36, 2005.

[19] DJ. Groen. Reliability analysis of grid resources: A user perspective. Master's thesis, University of Amsterdam, 2006. http://www.science.uva.nl/research/scs/papers/archive/Groen2006a.pdf.

[20] IETF. Reliable multicast transport. http://rmt.motlabs.com.

[21] int.eu.grid. Interactive european grid project. http://www.interactive-grid.eu.

[22] ITU-T. Recommendation x.509. International Standard 9594-8, ISO/IEC, 1997/2002.

[23] Alcatel Lucent. Network availability in meshed transport networks. Technical report, Technology White Paper, 2007. http://www1.alcatel-lucent.com/com/en/appcontent/opgss/Net_Avail_Meshed_twp_tcm228-1283621635.pdf.

[24] Hans von der Schmitt et al. Atlas streaming study group. Technical report, CERN, 2006. http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/COMMISSIONING/streaming.html.