Performance of the final Event Builder for the ATLAS Experiment

Hans Peter Beck, Maris Abolins, Andreas Battaglia, Robert Blair, Andre Bogaerts, Martine Bosman, Matei Ciobotaru, Robert Cranfield, Gordon Crone, John Dawson, Robert Dobinson†, Marc Dobson, Andre Dos Anjos, Gary Drake, Yuri Ermoline, Roberto Ferrari, Maria Lorenza Ferrer, David Francis, Szymon Gadomski, Sonia Gameiro, Benedetto Gorini, Barry Green, William Haberichter, Christian Häberli, Reiner Hauser, Christian Hinkelbein, Richard Hughes–Jones, Markus Joos, Gerard Kieft, Sander Klous, Krzystof Korcyl, Konstantinos Kordas, Andreas Kugel, Lucian Leahu, Giovanna Lehmann, Brian Martin, Livio Mapelli, Christophe Meessen, Catalin Meirosu, Andrzej Misiejuk, Giuseppe Mornacchi, Matthias Müller, Yasushi Nagasaka, Andrea Negri, Enrico Pasqualucci, Thilo Pauly, Jorgen Petersen, Bernard Pope, James Schlereth, Ralf Spiwoks, Stefan Stancu, John Strong†, Sergey Sushkov, Tadeusz Szymocha, Louis Tremblet, Gokhan Unel, Wainer Vandelli, Joseph Vermeulen, Per Werner, Sarah Wheeler-Ellis, Fred Wickens, Werner Wiedenmann, Maoyuan Yu, Yasushi Yasu, Jinlong Zhang and Haimo Zobernig

***Abstract*—Event data from proton-proton collisions at the LHC will be selected by the ATLAS experiment in a three level trigger system, which reduces the initial bunch crossing rate of 40 MHz at its first two trigger levels (LVL1+LVL2) to ~ 3 kHz. At this rate the Event-Builder collects the data from all Read-Out system PCs (ROSs) and provides fully assembled events to the the Event-Filter (EF), which is the third level trigger, to achieve a further rate reduction to ~ 200 Hz for permanent storage. The Event-Builder is based on a farm of $O(100)$ PCs, interconnected via Gigabit Ethernet to $O(150)$ ROSs. These PCs run Linux and multi-threaded software applications implemented in C++. All the ROSs and one third of the Event-Builder PCs are already installed and commissioned. We report on performance tests on this initial system, which show promising results to reach the final data throughput required for the ATLAS experiment.**

***Index Terms*—High Energy Physics, LHC, ATLAS, TDAQ, Event Building.**

Manuscript received May 11, 2007.

H.P. Beck is corresponding author. Email: Hans.Peter.Beck@cern.ch

H.P. Beck, A. Battaglia, S. Gadomski, C. Häberli and K. Kordas are with the Laboratory of High Energy Physics Department, University of Bern, Switzerland

M. Abolins, Y. Ermoline, R. Hauser and B. Pope are with the Michigan State University, Ann Arbor, MI, USA

R. Blair, J. Dawson, G. Drake, W. Haberichter, J. Schlereth and J. Zhang are with the Argonne National Laboratory, Argonne, IL, USA

A. Bogaerts, R. Dobinson, M. Dobson, D. Francis, S. Gameiro, B. Gorini, M. Joos, G. Lehmann, B. Martin, L. Mapelli, G. Mornacchi, T. Pauly, J. Petersen, R. Spiwoks, L. Tremblet, G. Unel, W. Vandelli and P. Werner are with CERN, Geneva, Switzerland

M. Bosman and S. Sushkov are with the Institut de Fisica de Altas Energias (IFAE), Universidad Autonoma de Barcelona, Spain

M. Ciobotaru, A. Negri, S. Stancu and S. Wheeler-Ellis are with the University of California Irvine, CA, USA

R. Cranfield and G. Crone are with the University College London, UK

A. Dos Anjos, W. Wiedenmann and H. Zoberning are with the University of Wisconsin, Madison, WI, USA

R. Ferrari is with the INFN Sezione di Pavia, Italy

M.L. Ferrer is with INFN Frascati, Italy

B. Green and A. Misiejuk are with the Physics Department, Royal Holloway College, University of London, UK

C. Hinkelbein, A. Kugel, M. Müller and M. Yu are with the Unversität Mannheim, Germany

R. Hughes–Jones is with Manchester University, UK

G. Kieft, S. Klous and J. Vermeulen are with NIKHEF, Amsterdam, The Netherlands

L. Leahu and C. Meirosu are with the National Institute for Physics and Nuclear Engineering "Horia Hulubei", Bucarest, Romania

C. Messens is with the CPPM Marseille, France

Y. Nagasaka is with the Hiroshima Institute of Technology, Japan

E.Pasqualucci is with the Universita di Roma "La Sapienza" and with the INFN Roma, Rome, Italy

K. Korcyl and T. Szymocha are with the Henryk Niewodniczanski Institute for nuclear Physics, PAS, Cracow, Poland

F. Wickens is with the CCLRC Rutherford Appleton Laboratory, Chilton, Didcot, Oxon OX11 0QX, UK

Y. Yasu is with the High Energy Accelerator Research Organization (KEK), Tsukuba, Japan

†deceased

I. Introduction

THE ATLAS trigger and data-acquisition (TDAQ) system is based on three levels of online event selection [1], [2]. Each trigger level refines the decisions made at the previous level and, where necessary, applies additional selection criteria. Starting from an initial bunch-crossing rate of 40 MHz, corresponding to an interaction rate of $\sim 10^9$ Hz at a luminosity of 10^{34} cm^{-2}s^{-1}, the rate of selected events must be reduced to $\mathcal{O}(200)$ Hz for permanent storage. This requires an overall rejection factor on the trigger level of 10^7 against minimum-bias events, while retaining the rare new physics processes, such as Higgs boson decays.

The LVL1 trigger [3] reduces the event rate to 75 kHz (upgrade-able to 100 kHz) based on an initial selection using reduced granularity information from a subset of the detectors. High transverse momentum muons are identified using only the muon-trigger chambers, RPCs [2] in the barrel, and TGCs [2] in the end-caps. The calorimeter selections are based on reduced granularity information from all the calorimeters (electromagnetic and hadronic; barrel, end-cap and forward).

The LVL2 further reduces the event rate down to ~ 3 kHz. The latency of the LVL2 trigger is variable from event to event and is expected to be $\mathcal{O}(10)$ ms.

After LVL2, the event is fully assembled by the Event Builder and then sent to the last stage of the online selection,

the Event Filter. The Event Filter employs offline algorithms and methods, adapted to the online environment. It will use the most up to date calibration and alignment information and an accurate magnetic field map. The Event Filter makes the final selection of physics events which are written to mass storage for subsequent offline analysis. The output rate from LVL2 is reduced by an order of magnitude, giving ~ 200 Hz, corresponding to an output data rate of ~ 300 MB/s. Over the course of a year's data taking a data volume of $\mathcal{O}(2-3)$ PB will accumulate and be analyzed in the various institutes involved in ATLAS.

II. The ATLAS Detector Readout

The readout of the ATLAS detector starts at the level of the on- or near-detector front-end electronics. Here the signals for every channel are stored, depending on the sub-detector either analog or digitized, for every bunch crossing during the LVL1 trigger latency ($2.5\mu s$). When an event is accepted by the LVL1 trigger, the stored signals are moved via front-end links into readout drivers, located off the detector in the underground technical cavern next to the ATLAS main cavern. The readout drivers perform data formatting, and some can do zero suppression. The formatted data fragments are pushed via ~ 1600 optical links at an aggregated bandwidth of up to ~ 120 GB/s to the Read-Out System (ROS) which buffers the data and makes it available via a multi-stage Gigabit Ethernet network to the LVL2 trigger system and to the Event Builder.

A. The Read-Out System

The Read-Out System is implemented with $\mathcal{O}(150)$ PCs, each holding three or four custom PCI modules that can receive and buffer the event data from three optical links [5].

The ROS PC is rack mountable, runs Linux and occupies four rack units. A single 3.4 GHz Intel Xeon CPU [7] can access the buffered event data fragments from the custom modules via four independent PCI busses and serve the data on request to the LVL2 system and the Event Builder via up to four Gigabit Ethernet network ports. It is also the same CPU that must initiate commands to clear events from the memory buffers, once the corresponding data is no longer in needed by LVL2 and/or Event Building.

B. The ATLAS Event Builder

Components involved in the Event Building process are the Read-Out System (ROS), the Dataflow Manager (DFM) and the Event Builder nodes, often referred to as SubFarm Inputs (SFIs).

Events accepted by LVL2 are fully assembled and formatted in the Event Builder nodes. This process needs to be orchestrated by a supervising component responsible for receiving trigger decisions from LVL2 and load-balancing the farm of event building nodes. This task is performed by the Dataflow Manager (DFM). All Event Builder components are rack-mounted PCs and do not require any special hardware apart from high performance network interface cards that provide full connectivity to a central gigabit Ethernet network.

1) The Dataflow Manager: The Dataflow Manager (DFM) component starts the event-building process upon reception of a message by the LVL2 trigger system on all by LVL2 accepted and rejected events. This message is usually grouped, bundling several hundred LVL2 decisions into one message that is transported via Gigabit Ethernet.

For commissioning purposes only, the DFM can also be informed by the LVL1 trigger system directly, thus providing an effective bypass of the LVL2 system. Furthermore, the DFM provides an internal self-triggering mode, which is useful for various system tests of the Event Builder, including the performance tests presented in this paper.

For each event to be built, the DFM then allocates an Event Builder node according to a load-balancing algorithm to which it sends a build command via Gigabit Ethernet to initiate the build.

For events rejected by LVL2 and for events that have been built successfully, the DFM sends a message containing ~ 100 event identifiers to every ROS PC to initiate clearing of the corresponding buffer memories. Again, Gigabit Ethernet is used, taking advantage of the UDP multicast protocol.

The DFM keeps track on the oldest LVL1 identifier that possibly can still be asked for by LVL2 or Event Building [4]. The oldest LVL1 identifier is shipped together with every clear message sent to the ROS PCs and can be used to initiate a garbage collection process on the buffer memories, in cases when the memory consumption becomes critical. This is necessary, as the UDP multicast protocol does not guarantee for the delivery of a message to every peer.

The DFM is a rack-mountable PC, occupying one unit of rack space and running Linux. Two 2.6 GHz AMD Opteron 252 CPUs [8] execute the multi-threaded DFM application code, implemented in C++.

One DFM PC is sufficient to orchestrate the event-building process for ATLAS. However, during the commissioning phase of ATLAS, up to 12 DFMs are made available to allow for up to 12 independent event building slices to be set up, and to be run individually. Commissioning of the various sub-detector systems of ATLAS can thus progress in parallel rather than in sequential steps.

2) The Event Builder Nodes: The SubFarm Input is the event-building node. The SFIs are allocated by the DFM and request and receive event data fragments from the ROSs via a Gigabit Ethernet network. The data fragments received are built and formatted to complete events. In cases where a ROS data fragment does not arrive within a pre-defined time budget, the outstanding data fragment can be re-asked for. Only if several consecutive requests fail, the SFI would give up and build an incomplete event with one or few ROS data fragments missing.

After the build process of an event is finished (successfully or otherwise), the SFI sends a message to the DFM for its internal bookkeeping and load-balancing algorithm.

Built events are buffered in the SFIs in order to be served to the Event Filter. Only after an event has been shipped successfully to an Event Filter node, the SFI frees up its buffers. In cases where too many events occupy buffer space, the SFI sends a flow-control message to the DFM to no-longer

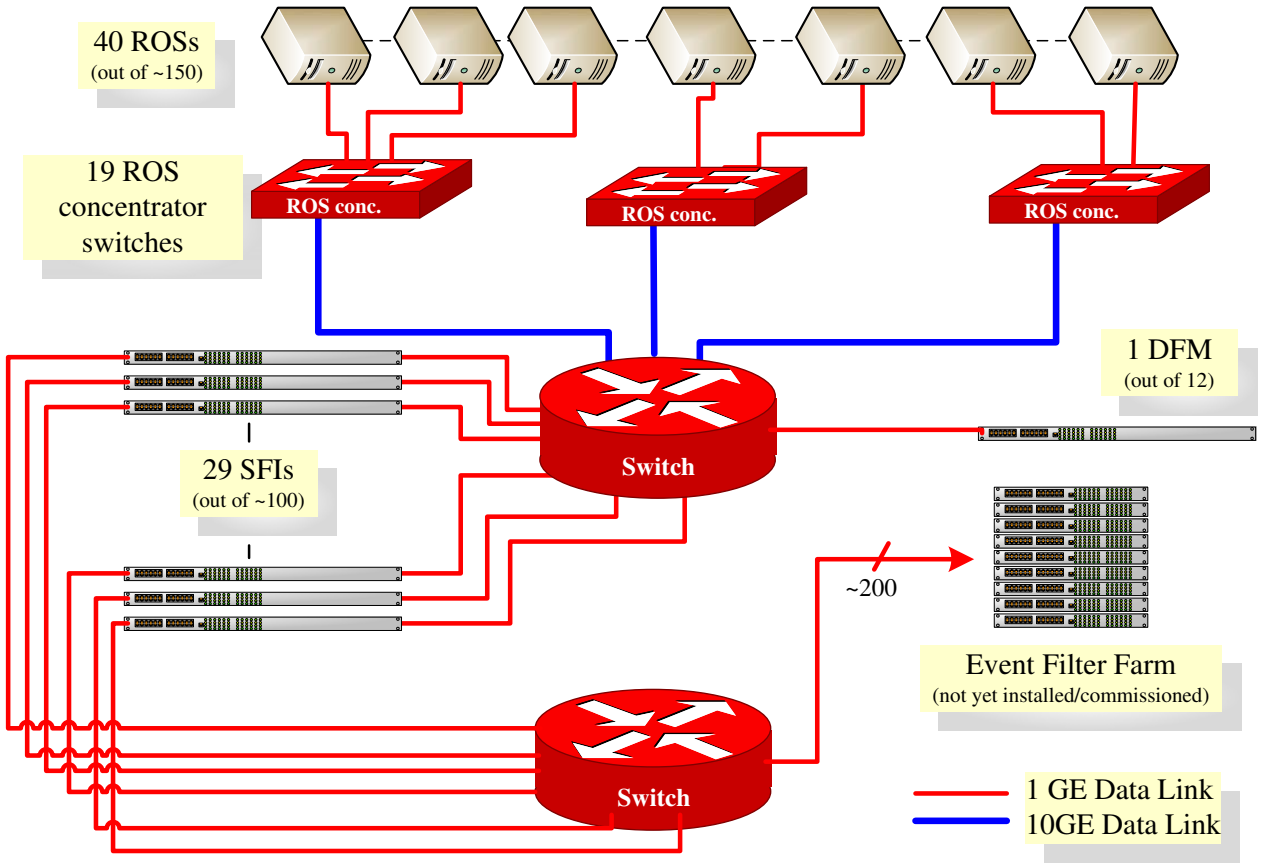

Fig. 1. Topology of the ATLAS Event Builder network. Gigabit Ethernet connections between end-ports and 10 Gigabit Ethernet up-links between switches route the event data and control messages between the data-flow components. 40 out of $\sim$ 150 ROSs and up to 29 out of $\sim$ 100 SFIs in final ATLAS have been used for the measurements described in section IV.

assign new events to it. A corresponding flow-control message can be sent by the same SFI to inform the DFM once it has again capacity to build events.

For efficiency reasons, an SFI can build more than one event in parallel.

The SFI is a rack-mountable PC, occupying one unit of rack space and is running Linux. Two 2.6 GHz AMD Opteron 252 CPUs execute the multi-threaded SFI application code, implemented in C++.

$\mathcal{O}(100)$ SFI PCs are envisaged for the final system.

3) Traffic Shaping: Receiving data fragments from all ROSs into a single SFI can easily cause congestion of Ethernet packets, as they potentially can arrive all simultaneously at the same SFI. This ultimately results in the dropping of Ethernet packets by the network components. In order to prevent this, the SFI limits at any moment the number of outstanding requests it sends to the ROSs. Furthermore, every SFI maintains a set of randomized lists of ROS PC network ports. These lists are used to define the (randomized) order at which the data requests are sent out to the ROSs, aiming for a complete randomization of the traffic pattern in the underlying network.

4) Message Passing Protocols: Messages exchanged between the ROS, DFM and SFI PCs travel via a Gigabit Ethernet network. The underlying protocol can be either UDP/IP or TCP/IP configurable for every message type individually. The baseline solution foresees to utilize UDP/IP for the data request and data reply messages exchanged between the SFIs and the ROSs. UDP/IP multicast is used by the DFM to send clear messages to all involved ROSs to initiate freeing of the corresponding buffer memories in the ROSs. All other messages that are required to orchestrate the proper operation of the event building process, such as, the load-balancing of the SFIs, and flow-control are foreseen to utilize TCP/IP as underlying protocol.

This choice is motivated by the following:

TCP/IP maintains individual connections between all its peers and guarantees delivery of messages. TCP/IP therefore inherently causes for non-trivial utilization of resources at the ROS and SFI side, as each application has to handle many connections concurrently. Every connection has its own set of timers (handled inside the TCP/IP stack) and does handle congestion avoidance and packet loss recovery independently. The sending and receiving of extra packets to acknowledge the reception of data between two peers cause for another overhead in the network switches and the applications. If packet loss were to occur, TCP/IP will reduce its sending rate which could cause unwanted delays in the Data Flow. The timers in the TCP/IP stack cannot be set by the application which may also be a factor in reducing performance.

Therefore, scaling problems with TCP/IP are thought to occur when many connections need to be handled in parallel.

UDP/IP, on the other hand, is a connection-less protocol with no guarantee for delivery of messages. Therefore, no protocol related scaling issues will occur. Also there is no reduction in the sending data rate. An unreliable protocol is justified for the request-reply traffic type between SFIs and ROSs. A non-delivery of a message will result in a time-out at the SFI level, when waiting for the data reply, and a new request can be made. The reask mechanism together with the traffic shaping can reduce the building when not tuned correctly.

The reliability of the TCP/IP protocol in turn is of great simplification for all the other messages that are needed to orchestrate the event building. No big message rates and no big data volumes are involved for these messages. And thus, no limiting scaling problems are expected.

For comparison reasons, the Event Builder can be configured such that TCP/IP is used everywhere.

III. Network

The network that interconnects the ROS, DFM and SFI PCs must be capable of interconnecting $\mathcal{O}(150)$ ROSs with $\mathcal{O}(100)$ SFIs, and must also allow the LVL2 system to access the ROSs. A layout of the multi-level Gigabit Ethernet network with 10Gigabit Ethernet interconnects between some of the switches involved is depicted in Fig. 1.

The central switch is a rack-mountable chassis based E1200 switch from Force10 [6], capable of interconnecting up to 14 blades of various types providing copper-based or optical-based Gigabit as well as 10Gigabit Ethernet ports. At the time of this write-up, six blades providing a total 24 optical 10Gigabit Ethernet ports and two blades providing a total of 96 Gigabit Ethernet ports were installed.

IV. Performance Measurements of the ATLAS Event Builder

At the time of this write-up, the network switches, the complete set of 153 ROSs and about one third of the expected event builder nodes have been installed, commissioned and tested in various configurations involving stand-alone testing as well as fully integrated system tests with some of the ATLAS sub-detectors sending test-pulses as well as cosmic ray data.

The used test set-up involved 40 ROSs, 29 SFIs and one DFM. In order to run the Event-Builder at its utmost speed, the DFM was configured to generate its own trigger signals as fast as possible in a self-triggering mode. The ROSs were configured to deliver an event fragment of fixed, but configurable size to the requesting SFI.

A. Traffic Shaping

The event-building total rate and aggregated bandwidth has been measured with 29 SFIs building events from all 40 ROSs. Different ROS fragment sizes can be requested from the ROSs, to result in event sizes of 210, 418, 834 and 1505 kB. For 1505 kB event size, both UDP/IP and TCP/IP have been exercised for the shipping of event fragment data, whereas for other event sizes, only UDP/IP was used.

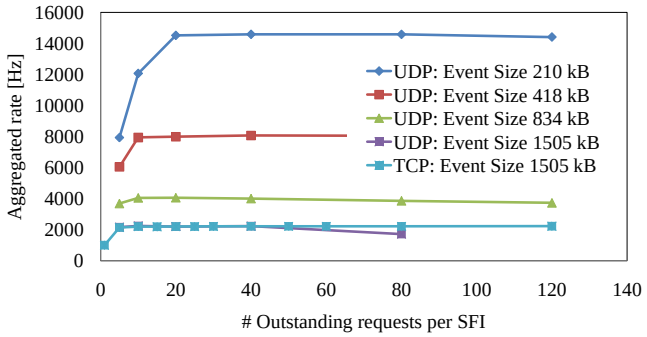

Fig. 2. Total event-building rate when building events concurrently from 40 ROSs with 29 SFIs. The number of outstanding data requests per SFI has been varied.

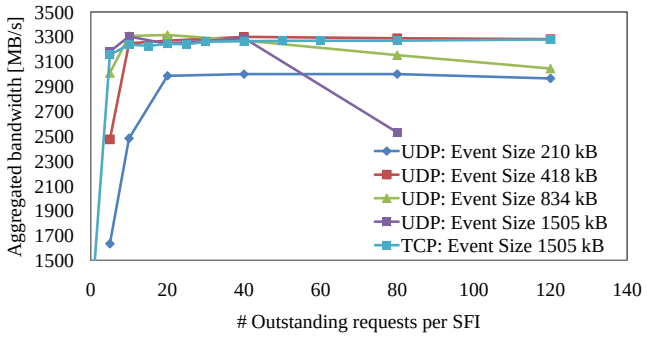

Fig. 3. Total event-building bandwidth when building events concurrently from 40 ROSs with 29 SFIs. The number of outstanding data requests per SFI has been varied.

Following the pull paradigm of the event-building protocol, traffic-shaping is realized in limiting the number of outstanding request that an individual SFI is allowed to execute at any moment in time. This number has been varied and the resulting event builder performance was measured.

As can be seen in Figs. 2 and 3, a ramp-up behavior is apparent when the number of outstanding requests is varied between 1 and 120. When there are too few outstanding requests configured, the latency involved between data requests and the corresponding data delivery cause for an under-utilization of the available network resources. On the other hand, when too many outstanding requests are allowed, too many Ethernet packets need to be handled by the network. Congestion occurs causing delays or packet loss and therefore re-asks for late- and for not-arriving event data occurs. In our measurements, only when the event size was set to the ATLAS default event size of 1.5 MB and when using UDP/IP a degradation became apparent.

When TCP/IP is used as the underlying protocol, no degradation of performance is seen. This means that on a system scale of up to 29 SFIs, TCP/IP handles congestion avoidance in an excellent way.

For the following measurements, the number of outstanding requests per SFI has been set to 20.

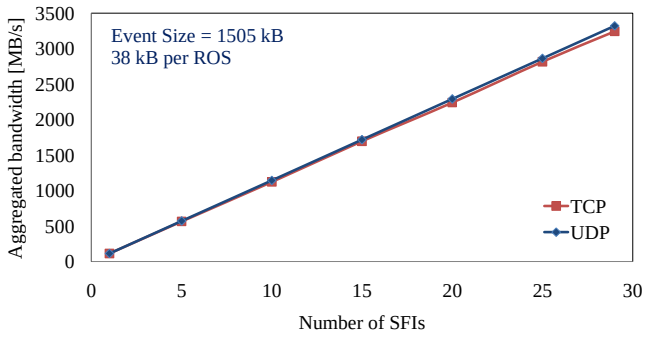

Fig. 4. Scaling of the ATLAS Event Builder using UDP/IP and TCP/IP for the transport of event data from the ROSs to the SFIs.

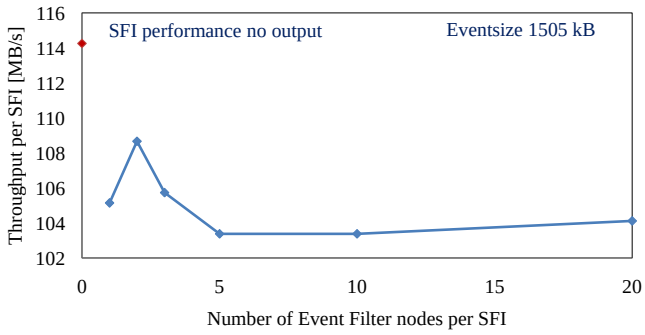

Fig. 5. Throughput of one SFI when more and more event filter nodes read out the built events

B. Event-Builder scaling properties

The event-building aggregated bandwidth has been measured with 1–29 SFIs building events from all 40 ROSs. The assembled event size was fixed at 1.5 MB, which is representing the final ATLAS event size. Both, UDP/IP and TCP/IP have been exercised for the shipping of event fragment data.

As can be seen from Fig. 4, the scaling behavior of the Event Builder is excellent for both the UDP/IP and the TCP/IP protocol. Every SFI adds $\sim$ 78 Hz and 114.2 MB/s to the total event rate and aggregated bandwidth. For the TCP/IP protocol a small deviation from the UDP/IP performance is seen when more than 15 SFIs are deployed. However, this effect is not dramatic, and further measurements with a bigger Event Builder will be required to clarify whether TCP/IP will really show scaling problems, as discussed in section II-B4.

A total aggregated bandwidth of 3.3 GB/s has been obtained for a one third scale of the ATLAS Event Builder. This is very encouraging as it already represents two thirds of the required bandwidth of 4.5 GB/s.

C. Event Builder Throughput

The task of the SFIs is not only to build events from ROS event fragments, but also to serve the built events to the $\mathcal{O}(1900)$ ATLAS Event Filter nodes. A degradation in the overall performance of the event-building capacity has thus to be expected.

As at the time of this write-up no Event Filter nodes are operational, the 29 existing SFIs have to be used instead. One SFI node is configured to build events from 40 ROSs of 1.5 MB event size. The remaining SFI PCs were configured to run the Event Filter I/O protocol [9] and were pulling event data as fast as possible out of the single event-building SFI.

The number of Event Filter nodes was thus varied from one to 20 and the performance of the one SFI was measured. Fig. 5 shows that the bandwidth per SFI drops from 114 MB/s to 103–109 MB/s, depending on the number of Event Filter nodes deployed. A degradation of $\sim 10\%$ in the overall performance of the Event Builder has therefore to be expected when data is also served to the Event Filter.

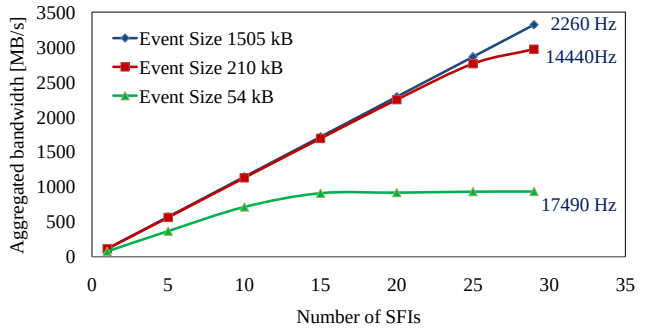

Fig. 6. Limiting the ROS

D. Limits of the Read-Out system

With 40 ROSs and up to 29 SFIs deployed in the test set-ups, the SFI should in principle be the limiting factor of the overall Event Builder. Only when more than 40 SFIs could be used, the ROS becomes a bottleneck. However, when the event size is set to very small values, down to 54 kB, the rate at which the SFIs can request data from the ROSs goes up to very high values.

Fig. 6 shows that the available aggregated event building bandwidth can be saturated with 15 SFIs deployed. A total event building rate of 17.5 kHz has been observed at an event size of 54 kB. With more SFIs deployed, the total rate stays stable, indicating that the SFIs are not the bottleneck and also indicating that deploying more SFIs does not degrade the total rate.

V. Conclusion

One third of the ATLAS Event Builder has been installed and exercised at the time of this write-up. UDP/IP is the baseline protocol to be used for the transport of event data from the ROSs to the SFIs using a pull paradigm. Furthermore, TCP/IP can be used, for requesting and sending event data and is being used for all other messages needed to comply with the event building message flow.

The Event Builder scaling potential when deploying one to 29 Event Builder nodes is excellent for both UDP/IP and for TCP/IP. Per SFI an incremental rate of 78 Hz and 114 MB/s has been measured, and a total rate of 2.2 kHz and

3.3 GB/s has been reached. This corresponds to two thirds of the required rate and bandwidth with a one third system.

However, a degradation of $\sim 10\%$ needs to be expected when the data is also served to the Event Filter. Further degradation may occur once the LVL2 data traffic is also run via the same network. The total size of ~ 100 SFIs for the final system is therefore still a safe estimate of the final ATLAS Event Builder scale.

ACKNOWLEDGMENT

The authors would like to thank ATLAS Online Software group for providing a system and useful tools to control, configure and operate a large-scale distributed TDAQ system as it is in need for the ATLAS experiment.

REFERENCES

[1] I. Riu *et al.*, *Integration of the Trigger and Data Acquisition Systems in ATLAS*, These proceedings.

[2] ATLAS Collaboration, *High-Level Trigger, Data Acquisition and Controls TDR*, CERN/LHCC/2003–022.

[3] D. Berge *et al.*, *Status of the ATLAS Level-1 Central Trigger and Muon Barrel Trigger and First Results from Cosmic-Ray Data*, These proceedings.

[4] H.P. Beck, *An algorithm to determine the "oldest still valid event identifier" in ATLAS TDAQ* , [Online]. Available: https://edms.cern.ch/document/478356

[5] B. Green, B., G. Kieft, A. Kugel, M. Müller and M. Yu, *ATLAS trigger/DAQ RobIn prototype*, IEEE Trans. Nucl. Sci. vol. 51 (2004) 465–469.

[6] Force10 Corporation, *E–Series*, [Online]. Available: http://www.force10networks.com/products/e-series_overview.asp

[7] Intel Corporation, *Xeon processor*, [Online]. Available: http://www.intel.com/xeon

[8] AMD Corporation, *Opteron processor*, [Online]. Available: http://www.amd.com/opteron

[9] H.P. Beck *et al.*, *EFIO: Protocol Specification*, [Online]. Available: https://edms.cern.ch/document/391570