The base-line DataFlow system of the ATLAS Trigger & DAQ

Hans Peter Beck, Maris Abolins, Andre Dos Anjos, Marcello Barisonzi, Matteo Mario Beretta, Robert Blair, Joannes Andreas Bogaerts, Henk Boterenbrood, David Botterill, Matei Dan Ciobotaru, Enrique Palencia Cortezon, Robert Cranfield, Gordon Crone, John Dawson, Beniamino Di Girolamo, Robert W. Dobinson, Yuri Ermoline, Maria Lorenza Ferrer, David Francis, Szymon Gadomski, Sonia Maria Gameiro, Piotr Golonka, Benedetto Gorini, Barry Green, Magali Gruwe, Stefan Haas, Christian Haeberli, Yoji Hasegawa, Reiner Hauser, Christian Hinkelbein, Richard Hughes-Jones, Emil Knezo, Peter Jansweijer, Markus Joos, Anna Kaczmarska, Gerard Kieft, Krzysztof Korcyl, Andreas Kugel, Andrew James Lankford, Giovanna Lehmann, Micheal LeVine, Weiyue Liu, Tadashi Maeno, Marcia Losada Maia, Livio Mapelli, Brian Martin, Robert McLaren, Catalin Meirosu, Andrzej Stanislaw Misiejuk, Remigius Mommsen, Giuseppe Mornacchi, Matthias Müller, Yasushi Nagasaka, Kazuo Nakayoshi, Ioannis Papadopoulos, Jorgen Petersen, Paulo de Matos Lopes Pinto, Daniel Prigent, Valeria Perez Reale, James Schlereth, Makoto Shimojima, Ralf Spiwoks, Stefan Nicolae Stancu, John Strong, Louis Tremblet, Jos Vermeulen, Per Werner, Frederick John Wickens, Yoshiji Yasu, Maoyuan Yu, Georg Zobernig, Marian Zurek

Hans Peter Beck on behalf of the ATLAS TDAQ DataFlow community [1]

ATL-DAQ-2004-006
22 March 2004

Manuscript received May 30, 2003.

H.P. Beck, C. Haeberli and V. Perez Reale are with the Laboratory of High Energy Physics, Bern University, 3012 Bern, Switzerland (e-mail: Hans.Peter.Beck@cern.ch)

S. Gadomski is with with the Laboratory of High Energy Physics, Bern University, 3012 Bern, Switzerland and on leave from Henryk Niewodniczanski Institute of Nuclear Physics, Cracow

M. Abolins, Y. Ermoline and R. Hauser are with Michigan State University, Department of Physics and Astronomy, East Lansing, Michigan

A. Dos Anjos and M. Losada Maia are with Universidade Federal do Rio de Janeiro, COPPE/EE, Rio de Janeiro

H. Boterenbrood, P. Jansweijer, G. Kieft and J. Vermeulen are with NIKHEF, Amsterdam

M. Barisonzi is with NIKHEF, Amsterdam and with Universiteit Twente, Enschede, Netherlands

M. Beretta, M.L. Ferrer and W. Liu, are with the Laboratori Nazionali di Frascati dell' I.N.FN, Fracasti

R. Blair, J. Dawson, J. Schlereth, Argonne National Laboratory, Argonne, Illinois

A.J. Bogaerts, M. Ciobotaru, E. Palencia Cortezon, B. Di Girolamo, R. W. Dobinson, D. Francis, S. Gameiro, P. Golonka, B. Gorini, M. Gruwe, S. Haas, M. Joos, E. Knezo, G. Lehmann, T. Maeno, L. Mapelli, B. Martin, R. McLaren, C. Meirosu, G. Mornacchi, I. Papadopoulos, J. Petersen, P. de Matos Lopes Pinto, D. Prigent, R. Spiwoks, S. Stancu, L. Tremblet and P. Werner are with CERN, Geneva, Switzerland

D. Botterill and F. J. Wickens are with Rutherford Appleton Laboratory, Chilton, Didcot

R. Cranfield and G. Crone are with the Department of Physics and Astronomy, University College London, London

B. Green, A. Misiejuk and J. Strong are with the Department of Physics, Royal Holloway and Bedford New College, University of London, Egham

Y. Hasegawa is with the Department of Physics, Faculty of Science, Shinshu University, Matsumoto

R. Hughes-Jones is with the Department of Physics and Astronomy, University of Manchester, Manchester

A. Kaczmarska, K. Korcyl and M. Zurek are with Henryk Niewodniczanski Institute of Nuclear Physics, Cracow

C. Hinkelbein, A. Kugel, M. Müller and M. Yu are with Lehrstuhl für Informatik V, Universität Mannheim, Mannheim

A. Lankford and R. Mommsen are with University of California, Irvine, California

M. LeVine is with Brookhaven National Laboratory (BNL), Upton, New York

Y. Nagasaka is with Hiroshima Institute of Technology, Hiroshima

K. Nakayoshi and Y. Yasu are with KEK, High Energy Accelerator Research Organisation, Tsukuba

M. Shimojima is with the Department of Electrical Engineering, Nagasaki Institute of Applied Science, Nagasaki

G. Zobernig is with the Department of Physics, University of Wisconsin, Madison, Wisconsin

Abstract—**The base-line design and implementation of the ATLAS DAQ DataFlow system is described. The main components realizing the DataFlow system, their interactions, bandwidths and rates are being discussed and performance measurements on a 10% scale prototype for the final Atlas TDAQ DataFlow system are presented.**

This prototype is a combination of custom design components and of multi-threaded software applications implemented in C++ and running in a Linux environment on commercially available PCs interconnected by a fully switched gigabit Ethernet network.

I. Introduction

COLLISIONS of 7 TeV protons will be studied with the Large Hadron Collider (LHC) at CERN, Geneva, Switzerland. The LHC accelerator complex is currently in construction and scheduled to start operation in 2007. ATLAS is one of four detectors being built with the aim to explore the physics potential of LHC in its widest possible range [2].

Bunches of 10^{11} protons will collide at periods of 25 ns at the interaction point in the center of ATLAS. This will result in ~25 interaction events and ~2000 charged and neutral particles to be tracked with every crossing. Although individual proton-proton interact at ~1 GHz, the rate of new signatures such as production of Higgs particles, or other new heavy objects will be as low as a few events per hour and often much fewer. The event selection of ATLAS will therefore need to identify interesting physics signatures online while providing the required event rate reduction of 10^7, which gives a data volume still manageable for further offline analysis.

A three level trigger system reduces the initial bunch-crossing rate of 40 MHz at its first level trigger (LVL1) to 75 kHz with a fixed latency of 2.5 μs. The second level trigger (LVL2) analyses region of interests identified by LVL1 and reduces the event rate further to ~3 kHz with an average latency of 10 ms. The third trigger level is the event filter (EF) that analyses the entirety of the event data to achieve a further rate reduction to ~200 Hz, with a latency of ~1 s.

The amount of data produced for one ATLAS event is $O(1-2)$ MB read from as many as 140 million detector elements. This results into a data rate of ~300 MB/s for mass storage and a total amount of ~3 PB/year for detailed offline analysis.

The ATLAS DataFlow system is designed to cope with this amount of data being produced and serves data accepted by LVL1 to LVL2 and EF, i.e. the high level triggers (HLT) and, for accepted events, to mass storage.

II. DataFlow

On reception of a LVL1 accept signal (L1A), event data is moved from the detectors front-end electronics via point-to-point links into sub-detector specific read-out driver modules (RODs), where the data undergo preparation and formatting into ROD fragments. There are ~1600 RODs foreseen for ATLAS.

ROD fragments are moved at LVL1 rate into read-out buffers (ROBs), which are held in read-out systems (ROSs).

The role of the ROS is to provide an interface to the data kept in the ROB to the LVL2 processing farm and to the event building system.

A. Read-out link

The read-out link (ROL) connects the sub-detector RODs with the TDAQ system and is responsible for transmitting error-free data from the output of the ROD to the input of the ROB. As shown in Fig. 1, the ROD end of the ROL is called the link source card (LSC) and the ROB end is called link destination card (LDC).

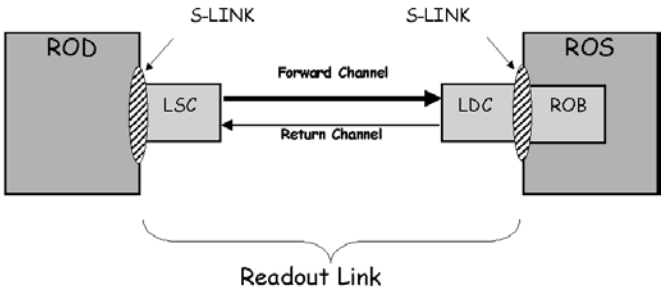

Fig. 1. The ROL implements the point-to-point connections between RODs and ROBs using the S-Link protocol.

The ROL is based upon the S-Link protocol [3] and provides

- 32 bit data words at 40.08 MHz, i.e. ~160 MB/s
- Xon/Xoff flow control
- Error detection with a bit error rate $\prec 10^{-12}$

The High-speed Optical Link for ATLAS (HOLA) [4] implements the ROL using a small FPGA, for handling the S-LINK protocol, and using the SERDES chip from Texas Instruments running at 2.5 Gbit/s, for handling both the forward and the return channels (one per card). For the optical transceiver, the Small Form Factor Pluggable Multimode 850 nm 2.5 Gbit/s with LC Connectors is foreseen, e.g. the Infineon V23818-N305-B57. The use of pluggable components allows the optical components to be changed in case of failure.

B. Read-out buffer

The number of ROB buffers is the same as the number of RODs (indeed, see below, the LVL2 trigger needs to access

data at the level of the individual ROD fragments). Event fragments are kept in the ROB until they are either moved downstream (accepted by LVL2) or they are removed from the system (rejected by LVL2). The depth of the ROB buffers is determined by the time needed by LVL2 to select events (10 ms), plus the additional overhead to clear (in case of a LVL2 reject) or transfer the fragment to the Event Builder and then to clear it. Taken the link speed of a ROL, 10 ms of buffering at the ROB require a minimum of 1.6 MB of memory per ROB. The current prototype RoBIn implements 64 MB of memory per ROB buffer allowing to absorb temporary congestions in the data flow.

Fig. 2 shows a RoBIn, a module implementing the ROB functionality, capable of receiving and buffering ROD fragments via S-Link and making these available on request.

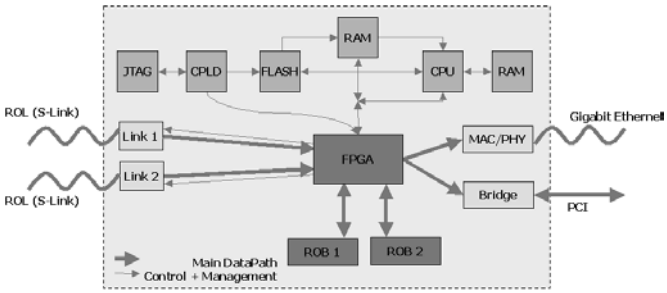

Fig. 2. Schematic diagram of the prototype RoBIn. The ROD fragments are received with LVL1 rate via two independent ROLs and buffered in two respective memory banks, the ROBs.

More than one ROL and thus ROB can be implemented on a RoBIn module, while the current prototype shows two input ROLs the final RoBIn may hold as many as four [5].

Two output interfaces have been implemented, based on gigabit Ethernet and PCI bus technology, and further study will be needed to decide which technology will be used in the final system.

C. Read-out system

The ROS houses a number of RoBIns, each multiplexing up to four ROLs into a single output interface. It provides individual event fragments, out of the ROBs, to the LVL2 trigger and to the event builder: in this latter case a further level of buffering, multiplexing several individual ROBs into a single event builder input, may be provided by the ROS.

Two deployment schemes for the ROS are under study:

1) Bus-based ROS

Three RoBIns, each with four ROLs and one PCI output, are mounted into the PCI slots of a PC equipped with four independent PCI bus segments.

Requests for fragments coming from LVL2 and requests for super-fragments (sequential merging of up to 12 fragments) from the event builder are handled by the ROS, i.e. by the PC, with the data moved across the PCI busses of the PC. Two Gigabit Ethernet interfaces connect the ROS to, respectively, the LVL2 and event builder networks.

2) Switch-based ROS

Ten RoBIns, each with four ROLs and one gigabit Ethernet output are mounted into an industrial PC providing enough PCI slots. The role of the PCI bus is to provide configuration, bookkeeping and power for the RoBIns. A ten by four gigabit Ethernet ports switch, which concentrates the ten ROB outputs into four gigabit Ethernet outputs reduces the number network ports needed for the LVL2 network and for the event builder network. No merging of fragments into super-fragments for the event builder is foreseen [6].

D. Region of interest builder

The region-of-interest builder (RoIB) collects the information relevant for LVL2 from the LVL1 calorimeter and muon triggers and from the LVL1 central trigger processor (CTP), and combines all data into a single block that serves as input to the LVL2 trigger. The data are transmitted in S-LINK format. The RoIB has to operate at the highest foreseen LVL1 output rates without introducing additional deadtime.

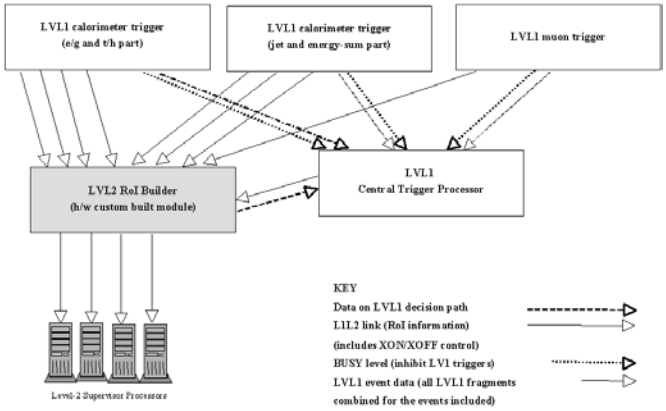

Fig. 3. The RoIB collects information relevant for the LVL2 from the LVL1 trigger system, and combines all data into a single block, which serves as input to the LVL2 trigger.

This enables a LVL2 processor to precisely select the region of the detector in which the interesting features reside and therefore from which ROBs to request the data for analysis.

The RoIB is a VME based system, which uses FPGAs to combine the LVL1 fragments into a single record. It is composed of two parts [7].

E. DataCollection

DataCollection is responsible for the movement of event data from the ROS to the LVL2 trigger and EF and from the EF to mass storage. This includes the movement of the LVL1 RoIs to the LVL2 processing units (L2PUs) and the LVL2 result (i.e. the LVL2 decision and a detailed LVL2 record in

case of accept) to the EF, which implies collection of RoIs, event building (EB) and I/O to and from the EF (EF I/O).

DataCollection components are software processes deployed on Linux PCs that are interconnected via a fully switched gigabit Ethernet network [8].

1) Level-2 supervisor

The Level-2 supervisor (L2SV) receives the RoI information produced by the RoIB and assigns a level-2 processing unit (L2PU) to process the event. The final system will contain less than ten L2SVs.

2) Level-2 processing unit

The L2PU is the component which, using the information provided by the L2SV, requests event fragments from the ROS, processes the RoI (i.e. runs trigger algorithms in the event data belonging to the RoI) and produces a decision (accept/reject) for the event. The decision is passed back to the L2SV. Strictly spoken, the algorithms performing the LVL2 selection are not DataCollection components, but these are embedded into the framework provided by DataCollection [9]. The final system will contain a few hundreds of L2PUs.

3) Pseudo-ROS

The Pseudo-ROS (pROS) receives the detailed result records of the L2PUs for accepted events and participates to the event building process, such that the LVL2 detailed result appears within the full event record. From the point of view of the event building process there is no difference between the pROS and the ROS. One pROS will be sufficient for the final system.

4) DataFlow Manager

The DataFlow Manager (DFM) receives the information about which events have been accepted or rejected by LVL2, assigns an event builder node (the SFI described below), and sends clear messages to the ROSs for their subsequent freeing of buffer space. One DFM will be sufficient to even building in the final system.

5) Subfarm Input

The Subfarm Input (SFI) receives information about which events to build and subsequently requests event data from all participating ROSs (which includes the pROS). It also implements traffic shaping in order to minimize congestion occurrences in the switching network. In case of temporary congestion and thus loss of event fragments, the SFI will re-ask these from the specific ROSs. Fully built events are buffered and made available to the EF for the final online trigger selection. The final system will contain ~100 SFIs.

6) Subfarm Output

The Subfarm Output (SFO) receives events accepted by the EF and stores them in files on a local hard disk. These files contain meta-information about the ongoing data-taking and are accessed by the ATLAS mass-storage system for permanent storage. The final system will contain ~10 SFOs.

III. Message Passing

The flow of event data between components of the DataFlow system is achieved by the exchange of control messages and subsequent event data messages via gigabit Ethernet network connections [6].

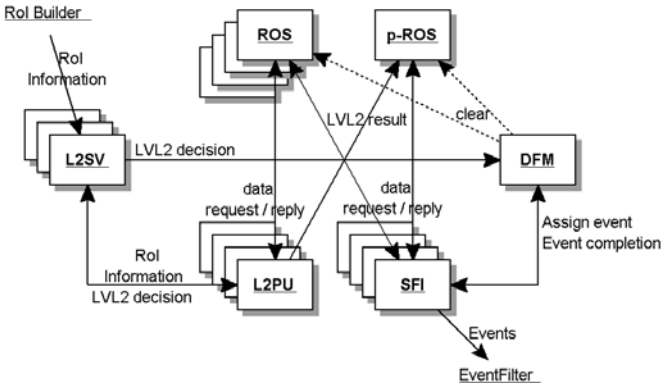

Fig. 4. Interaction between components of the DataFlow system.

Fig. 4 shows the basic interactions between components of the DataFlow system as realized by the DataCollection subsystem [10]. The sequence commences with the reception by a supervisor process of the RoI information, which represents the LVL1 result, from the RoIB. Using a load-balancing algorithm the supervisor assigns the event to a L2PU. The L2PU receives the RoI information from the L2SV which it uses to seed its processing. This results into a series of RoI data requests to a set of ROSs identified based on a geometry look-up table held by the L2PU. At a granularity of individual ROB data blocks, the selected ROSs service the request for data by responding to the requesting L2PU with a ROS event fragment message. The data volume per RoI is in the order of 2% of the total event size that needs to be moved this way from the ROBs into the requesting L2PU. Upon reaching a decision as to whether to accept or reject an event, the L2PU sends a LVL2 decision message back to its assigned supervisor process. In the case that the event is accepted for further processing by the EF the L2PU also sends the detailed result of its analysis to the pROS. The supervisor process receives the LVL2 decision and forwards a group of them to the DFM. On reception of a group of LVL2 decisions the DFM, based on a load-balancing algorithm, assigns an SFI to perform the building of the event for every accepted event. For rejected events and for events finished the event building process, the DFM multicasts a clear message to all ROSs. The SFI builds the event by sequentially requesting event date from all ROSs

(incl. pROS). The built event is subsequently sent to the EF subfarm for further processing.

The aggregated bandwidth sent through a switching matrix for the LVL2 and event building traffic is expected to be ~1.2 GB/s and 5 GB/s respectively.

Table I summarizes the control and data message rates exchanged between the DataFlow components. The impact of switch-based vs. bus-based ROS architecture is shown. The values presented depend on the final number of components for ROSs, L2SVs and L2PUs as well as on event size and its distribution and thus have to be taken as indicative only.

TABLE I
MESSAGE RATES AND BANDWIDTHS OF CONTROL AND DATA MESSAGES BETWEEN DATAFLOW COMPONENTS

	Message Type		Sender Rate	Sender Bandwidth	Receiver Rate	Receiver Bandwidth
L2SV → L2PU	RoI information		7.5 kHz	5 MB/s	0.5 kHz	0.3 MB/s
L2PU → ROS	Data requests	b)	6 kHz	0.6 MB/s	16 kHz	1.6 MB/s
		s)	11 kHz	1 MB/s	6 kHz	0.6 MB/s
ROS → L2PU	Event data	b)	16 kHz	32 MB/s	6 kHz	11 MB/s
		s)	6 kHz	6 MB/s	11 kHz	11 MB/s
L2PU → L2SV	LVL2 decisions		0.5 kHz	50 kB/s	7.5 kHz	750 kB/s
L2SV → DFM	LVL2 decision[g)]		75 Hz	40 kB/s	750 Hz	400 kB/s
DFM → SFI	Assign event		3 kHz	0.3 MB/s	30 Hz	3 kB/s
SFI → ROS	Data request	b)	4 kHz	0.4 MB/s	3 kHz	0.3 MB/s
		s)	48 kHz	4.8 MB/s	3 kHz	0.3 MB/s
ROS → SFI	Event data	b)	3 kHz	36 MB/s	4 kHz	48 MB/s
		s)	3 kHz	3 MB/s	48 kHz	48 MB/s
SFI → DFM	Finished Event		30 Hz	3 kB/s	3 kHz	0.3 MB/s
DFM → ROS	Clear buffers[g)]		250 Hz	0.4 MB/s	250 Hz	0.4 MB/s

b) bus-based ROS
s) switch-based ROS
g) message contents grouped for multiple event identifiers

A wide range of link technologies can handle the message rates and bandwidth. The choice is dictated by price, long term availability, support, inter-operability and suitability for DataFlow. Ethernet in its varieties of 100 Mbit/s and 1000 Mbit/s is the prime candidate and is chosen as base-line technology for the Atlas DataFlow system [11].

IV. PERFORMANCE

The final ATLAS DataFlow system requires simultaneous operation of RoI collection and event building. This section describes results obtained from a testbed capable of delivering ca. 10% of throughput as needed for the final Atlas TDAQ DataFlow system. Performance measurements of individual DataFlow components have been made and show satisfactory results. These are described in detail in [12].

The testbed consists out of 37 dual Intel Xeon 2.0-2.4 GHz CPU [13] rack-mountable PCs, interconnected via a fully switched Gigabit Ethernet network. The operating system used was the CERN certified Linux Redhat 7.2 [14]. The software used compiler version gcc-2.95.2.

Three kinds of traffic generators have been used to emulate large number of ROSs. These were based on custom-built FPGA boards, providing up to 128 ports; re-programmed network interface cards, providing up to 16 ports; and ROS emulation and ROS prototype software applications running on PCs, to be shared with the PCs available in the testbed [11]. Fig. 5 shows a picture of the testbed as currently deployed in CERN. The FPGA based network testers are identifiable on the right-most side of the photograph through the large amount of the 128 Ethernet cables connected to them. Other visible components are 1U and 4U high rack-mounted PCs.

Fig. 5. Atlas DataFlow performance testbed

A. RoI Collection

The maximum rate at which an L2PU can collect RoI data depends on the size of the RoI, the number of ROSs that contribute data and the number of threads that collect RoI data in parallel on the same L2PU. Fig. 6 shows 1/Rate for an RoI of 16 kB collected as 1, 2, 4, 8, 16 or 22 slices of 16, 8, 4, 2, 1 or 0.8 kB respectively. For this test, the L2PUs were completely dedicated to data collection and no CPU time was allocated for algorithm processing. The plot shows that the time for acquiring RoI data is small compared to the execution time of selection software (currently aimed at 10 ms per event on average).

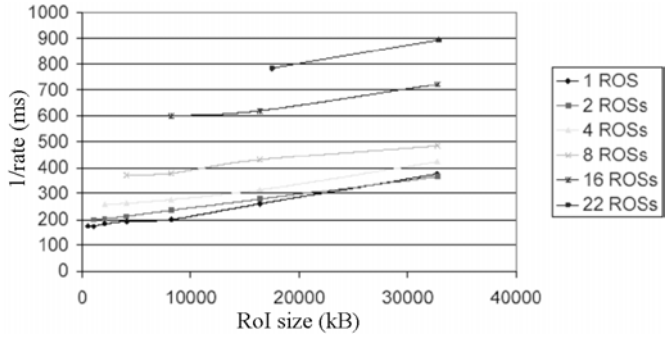

Fig. 6. Performance of RoI data collection for various combinations of RoI sizes.

B. *Event Building*

The building of events is performed by the DFM and SFIs requesting data from the ~140 up to 1600, depending whether bus-based ROSs aggregating up to 12 ROLs per ROS fragment or switch-based ROSs with no aggregation of ROLs are deployed.

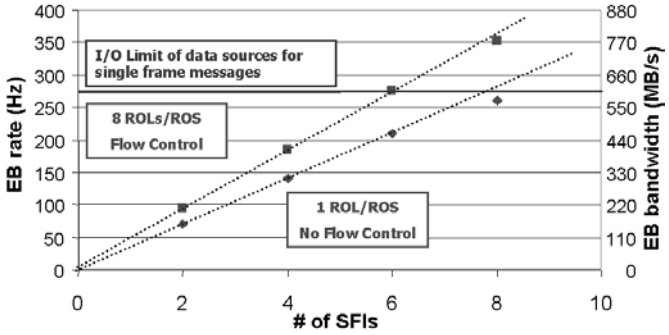

Fig. 7. Scalability of event building for bus-based and switch-based ROS scenarios.

The scalability of the event building of 2.2 MB size events is shown in Fig. 7. In this test the number of SFIs in the set up was increased from one to eight and the corresponding event building rate was measured.

It can be seen that the sustained event building rate increases linearly with respect to the number of SFIs in the system and that every additional SFI contributes to the overall system performance by ~ 40 Hz. It should be noted that the results for eight ROLs/ROS were achieved with Ethernet flow control active. The measurements with Ethernet flow control disabled for eight ROLs/ROS have yet to be understood.

V. Conclusions

Although the testbed necessarily is a scaled down version of the final system, individual components have been operated at rates similar to those expected in the final system. The primary aims of the 10% testbed are to demonstrate full functionality of the data collection in both the LVL2 and the EB subsystems simultaneously and to check for possible interference between the subsystems. The latter is especially important with respect to the choice to be made between a switch or bus based ROS. The testbed results have also been used to calibrate and validate computer models of components and systems [15].

This base-line DataFlow system and the performance of the prototype testbed will be documented in the Technical Design Report, to be published in June 2003.

VI. Acknowledgments

We wish to thank the ATLAS Online Software community for providing a system and useful tools to control, configure and operate large-scale distributed testbed setups.

VII. References

[1] ATLAS TDAQ DataFlow community, http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/DataFlow/DFAuthors.pdf

[2] ATLAS Collaboration, *ATLAS Detector and Physics Performance Technical Design Report*, CERN/LHCC/99-14

[3] The S-LINK *interface Specification*, http://edmsorweb.cern.ch:8001/ceder/doc.info?documnet_id=110828

[4] Design specification for HOLA, https://edms.cern.ch/document/330901/1

[5] A. Kugel et al., *ATLAS Trigger/DAQ Read-Out-Buffer (RoBIn) Prototype*, these proceedings

[6] H.P. Beck et al., *ATLAS TDAQ, a network-based architecture*, TDAQ Data Collection note 59, http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/DataFlow/DataCollection/docs/DC-059/DC-059.pdf

[7] R. Blair et al, *A Prototype RoI Builder for the Second Level Trigger of ATLAS Implemented in FPGA's*, ATL-DAQ-99-016

[8] C. Haeberli et al., *ATLAS-TDAQ DataCollection software*, these proceedings

[9] S. Armstrong, *Algorithms for the ATLAS High Level Trigger*, these proceedings

[10] H.P. Beck and C. Haeberli, *Mesage Flow: High-Level Description*, http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/DataFlow/DataCollection/docs/DC-012/DC-012.pdf

[11] M. LeVine et al, Network Performance Investigation for the ATLAS Trigger/DAQ, these proceedings

[12] ATLAS High Level Trigger, Data Acquisition and Controls, CERN / LHCC / 2003-022

[13] Intel corporation, http://www.intel.com/xeon

[14] http://linux.web.cern.ch/linux/

[15] R. Cranfield, P. Golonka, A. Kaczmarska, K. Korcyl, J. Vermeulen and S. Wheeler, *Computer modeling the ATLAS Trigger/DAQ system performance*, these proceedings