ATLAS DataFlow: the Read-Out Subsystem, Results from Trigger and Data-Acquisition System Testbed Studies and from Modelling

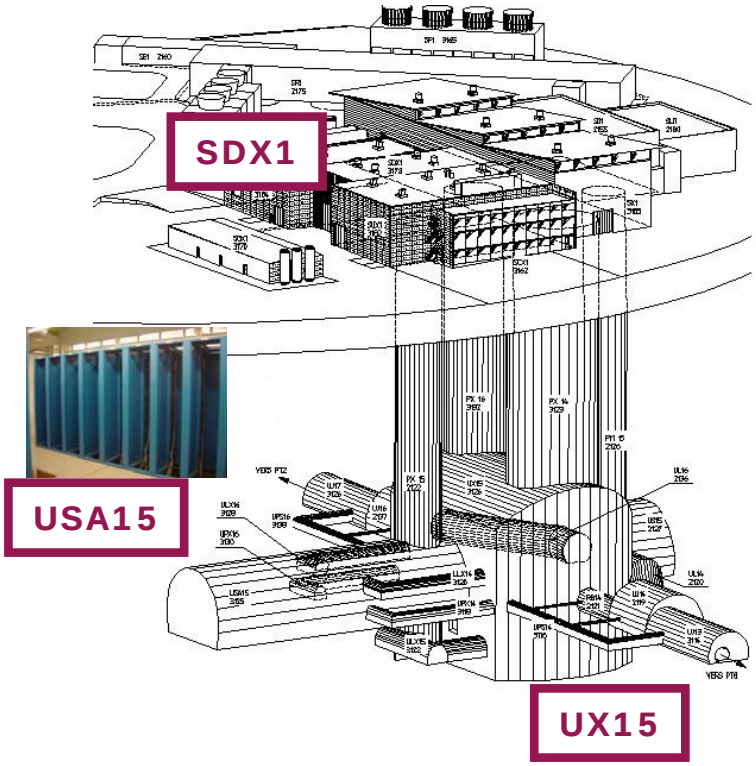
Jos Vermeulen on behalf of the ATLAS TDAQ DataFlow Community

M. Abolins[1], I. Alexandrov[2], A. Amorim[3], A. Dos Anjos[4], E. Badescu[5], N. Barros[3], H.P. Beck[6], R. Blair[7], D. Burckhart-Chromek[8], M. Caprini[8,5], M. Ciobotaru[8,9], A. Corso-Radu[8], R. Cranfield[10], G. Crone[10], J. Dawson[7], R. Dobinson[8†], M. Dobson[8], G. Drake[7], Y. Ermoline[1], R. Ferrari[11], M.L. Ferrer[12], D. Francis[8], S. Gadomski[6,20], S. Gameiro[8], B. Gorini[8], B. Green[13], M. Gruwé[8], S. Haas[8], W. Haberichter[7], C. Haeberli[6], Y. Hasegawa[14], R. Hauser[1], C. Hinkelbein[15], R. Hughes-Jones[16], M. Joos[8], A. Kazarov[8,17], G. Kieft[18], D. Klose[3], S. Kolos[19,17], K. Korcyl[20], K. Kordas[12], V. Kotov[2], A. Kugel[15], A. Lankford[19], G. Lehmann[8], M. J. LeVine[21], L. Mapelli[8], B. Martin[8], R. McLaren[8], C. Meirosu[8,9], M. Mineev[2], A. Misiejuk[13], G. Mornacchi[8], M. Müller[15], R. Murillo[8], Y. Nagasaka[22], J. Petersen[8], B. Pope[1], D. Prigent[8], Y. Ryabov[17], J. Schlereth[7], J. E. Sloper[8], I. Soloviev[8,17], R. Spiwoks[8], S. Stancu[8,19], J. Strong[13], L. Tremblet[8], N. Ünel[8], W. Vandelli[11], J. Vermeulen[18], P. Werner[8], F. Wickens[23], M. Wiesmann[8], M. Wu[15], Y. Yasu[24]

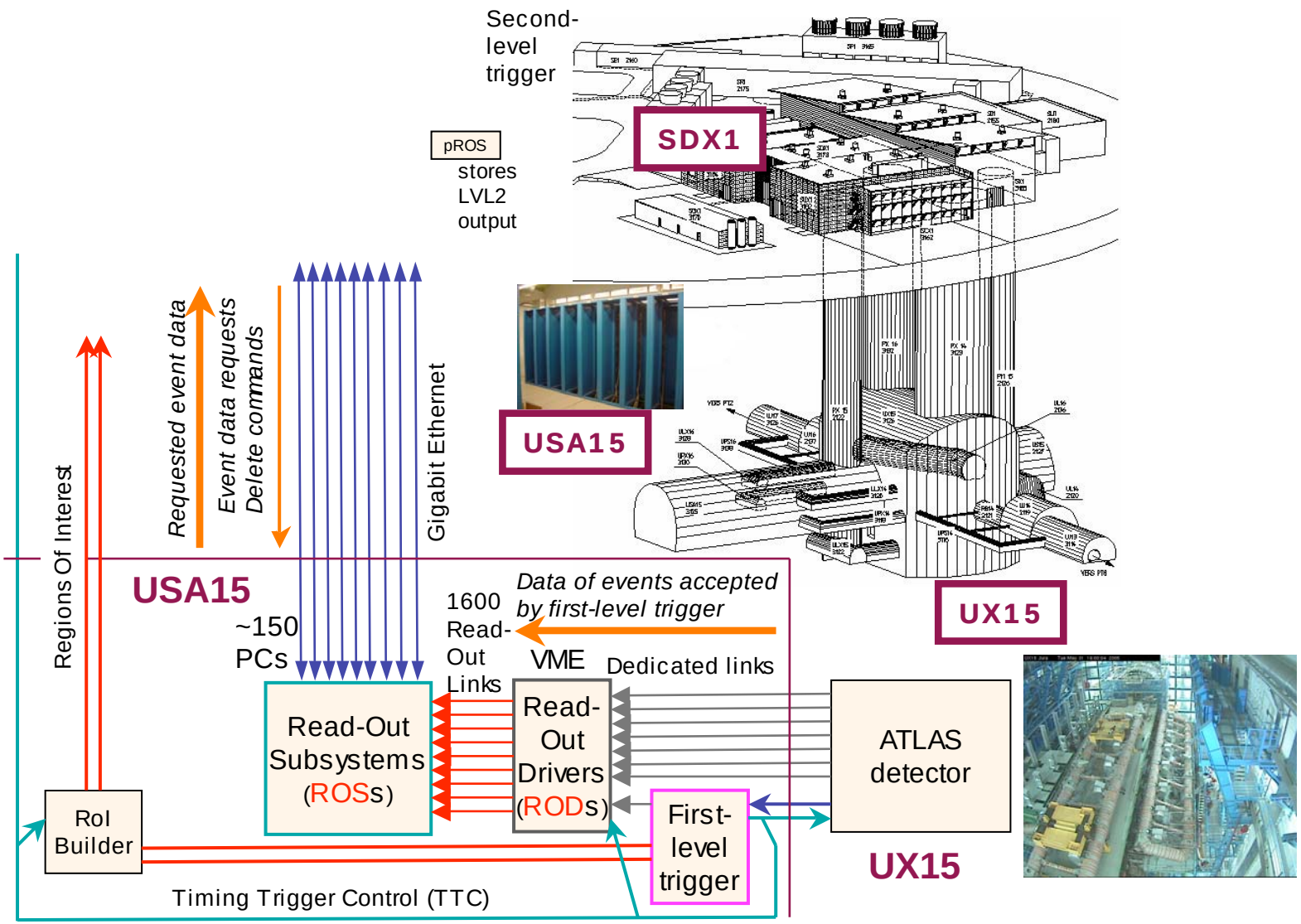
† deceased

[1]*Michigan State University, Department of Physics and Astronomy, East Lansing, Michigan, USA*
[2]*JINR, Dubna, Russia*
[3]*CFNUL/FCUL, Universidade de Lisboa, Portugal*
[4]*University of Wisconsin, Madison, Wisconsin, USA*
[5]*Institute of Atomic Physics, National Institute of Physics and Nuclear Engineering Bucharest, Romania*
[6]*LHEP, University of Bern, Bern, Switzerland*
[7]*Argonne National Laboratory, Argonne, Illinois, USA*
[8]*CERN, Geneva, Switzerland*
[9]*Politehnica, Bucuresti, Romania*
[10]*UCL, London, UK*
[11]*Dept. Fisica Nucleare e Teorica dell'Università di Pavia e INFN, Pavia, Italy*
[12]*Laboratori Nazionali di Frascati dell'INFN, Frascati, Italy*
[13]*Royal Holloway University of London, Egham, UK*
[14]*Department of Physics, Faculty of Science, Shinshu University, Matsumoto, Japan*
[15]*Lehrstuhl für Informatik V, Universität Mannheim, Mannheim, Germany*
[16]*The University of Manchester, Manchester, UK*
[17]*PNPI, St. Petersburg, Russia*
[18]*NIKHEF, Amsterdam, The Netherlands*
[19]*University of California at Irvine, Irvine, California, USA*
[20]*IFJ-PAN, Krakow, Poland*
[21]*Brookhaven National Laboratory (BNL), Upton, New York USA*
[22]*Hiroshima Institute of Technology, Hiroshima-shi, Hiroshima, Japan*
[23]*Rutherford Appleton Laboratory, Oxon, UK*
[24]*KEK, High Energy Accelerator Research Organisation, Tsukuba-shi, Ibaraki, Japan*

ATLAS Trigger / DAQ DataFlow Overview

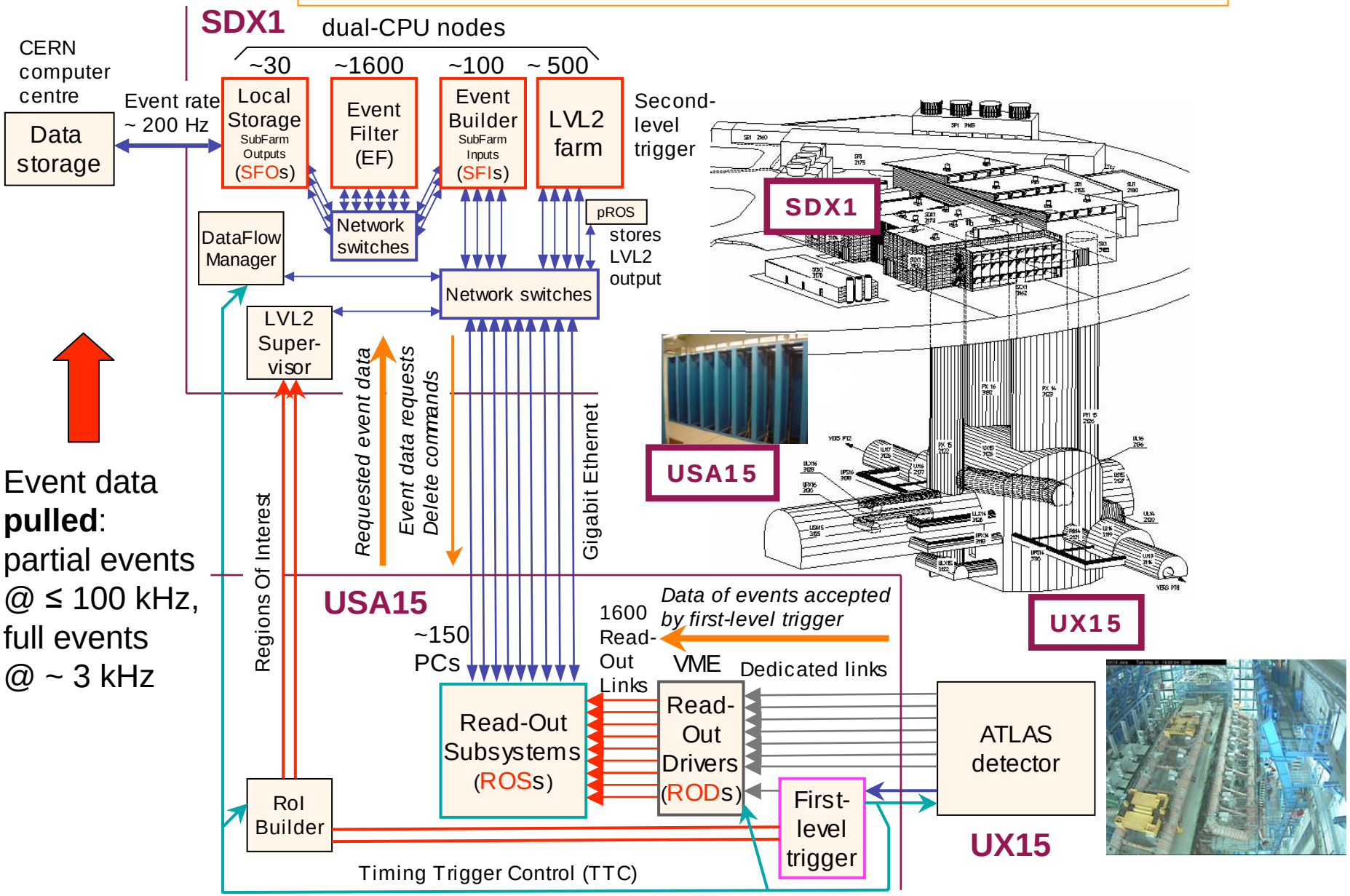


ATLAS Trigger / DAQ DataFlow Overview



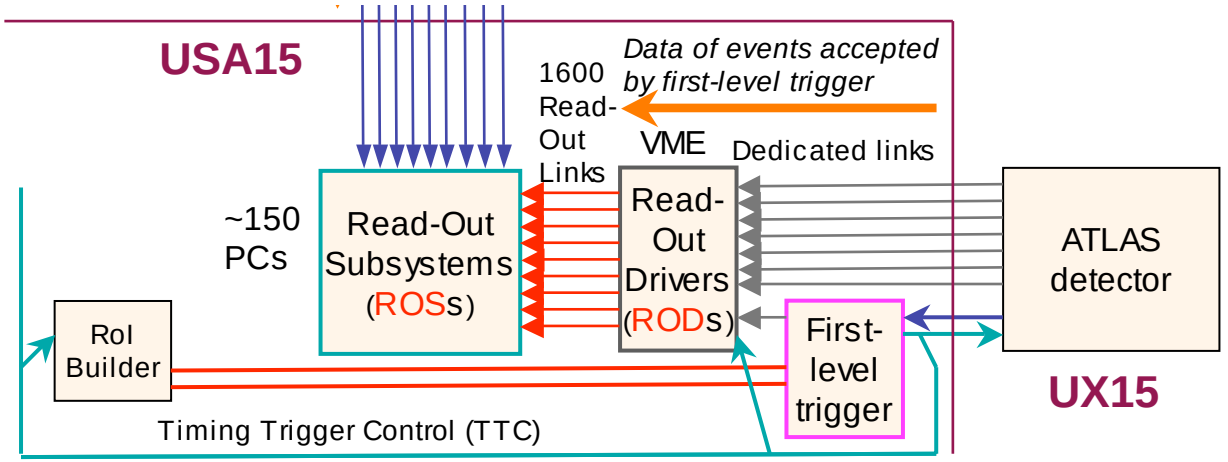
Event data **pushed** @ ≤ 100 kHz,
1600 fragments of ~ 1 kByte each

ATLAS Trigger / DAQ DataFlow Overview



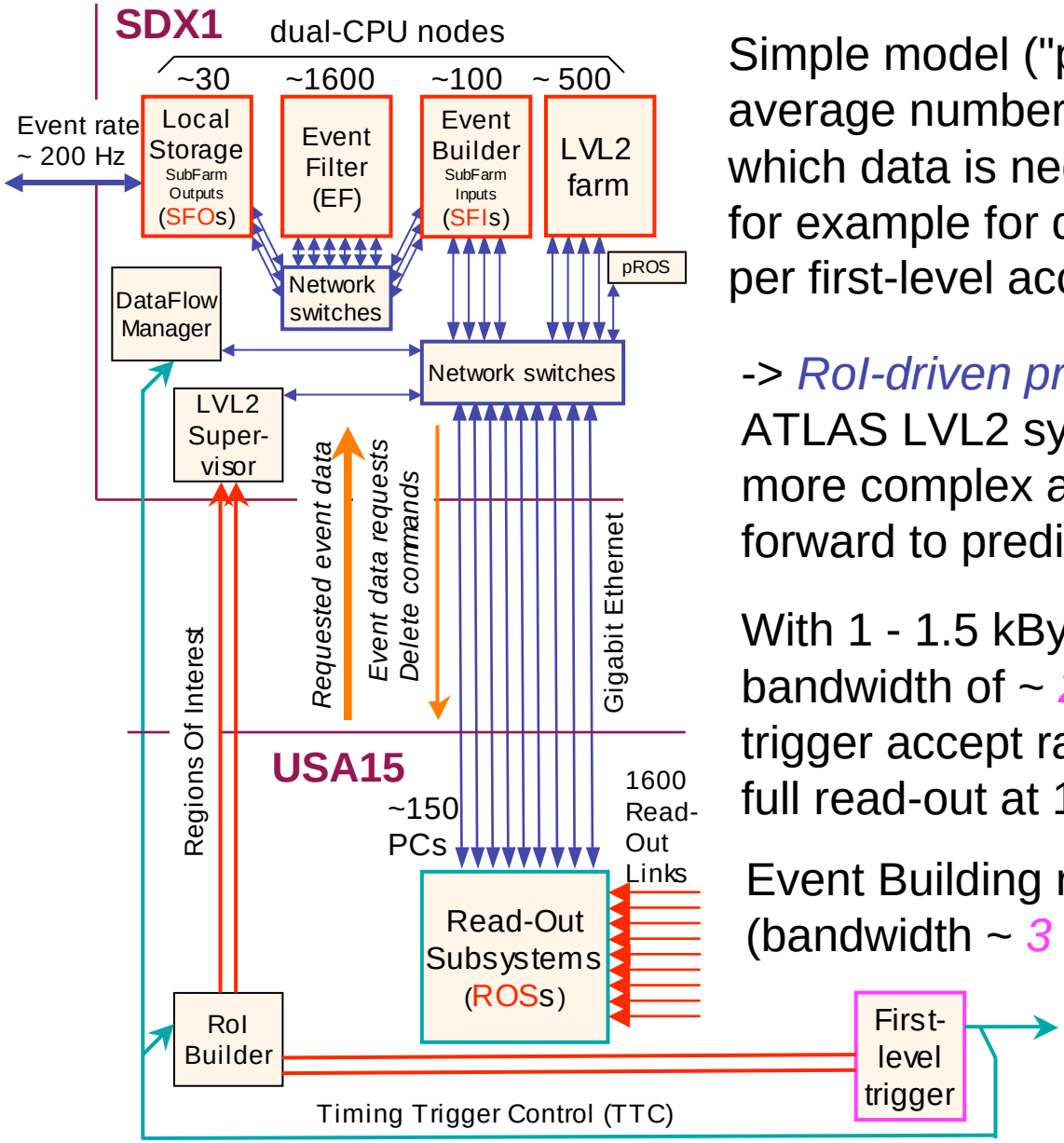
RODs, ROS PCs and ROBINs

- Read-Out Drivers (ROD):
  - subdetector-specific,
  - collects and processes data (no event selection)
  - output via Read-Out Link (ROL, 160 MByte/s optical fiber) to buffer on ROBIN card in Read-Out Subsystem (ROS) PC
- *Same type of ROLs, ROBINs and ROS PCs used for all sub-detectors*
- ROBIN: 64-bit 66 MHz PCI card
  Final version with 3 ROL inputs - results only for a single card
  Earlier prototype with 2 ROL inputs - system measurements also available
- ROS PC: 4U rack-mounted PC with 4 ROBINs => 12 ROLs per ROS PC



Event data **pushed** @ ≤ 100 kHz,
1600 fragments of ~ 1 kByte each

Second-Level (LVL2) Trigger and Event Building



Simple model ("paper model") used to predict average number of ROS PCs and ROLs from which data is needed for LVL2 trigger processing, for example for design luminosity trigger menu, per first-level accept: **16.2 ROLs** or **8.4 ROS PCs**

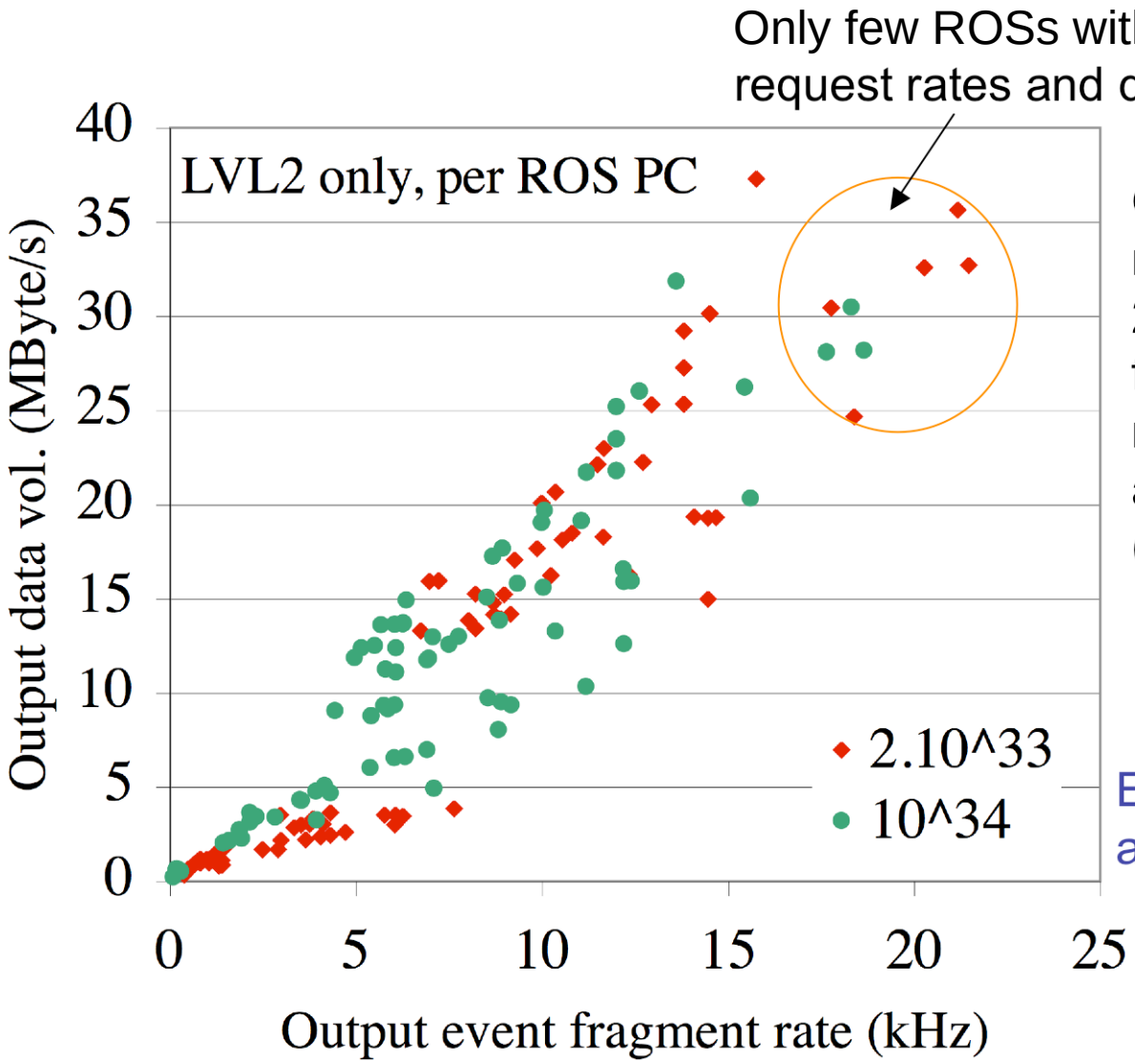
-> *RoI-driven processing* is a key property of the ATLAS LVL2 system, but also makes the system more complex and its performance not so straight-forward to predict.

With 1 - 1.5 kByte per fragment need network bandwidth of ~ *2 GByte/s* at 100 kHz first-level trigger accept rate (instead of ~ 150 GByte/s for full read-out at 100 kHz)

Event Building rate ~ **3 - 3.5 kHz**
(bandwidth ~ *3 - 5 GByte/s*)

ROS requirements, from paper model

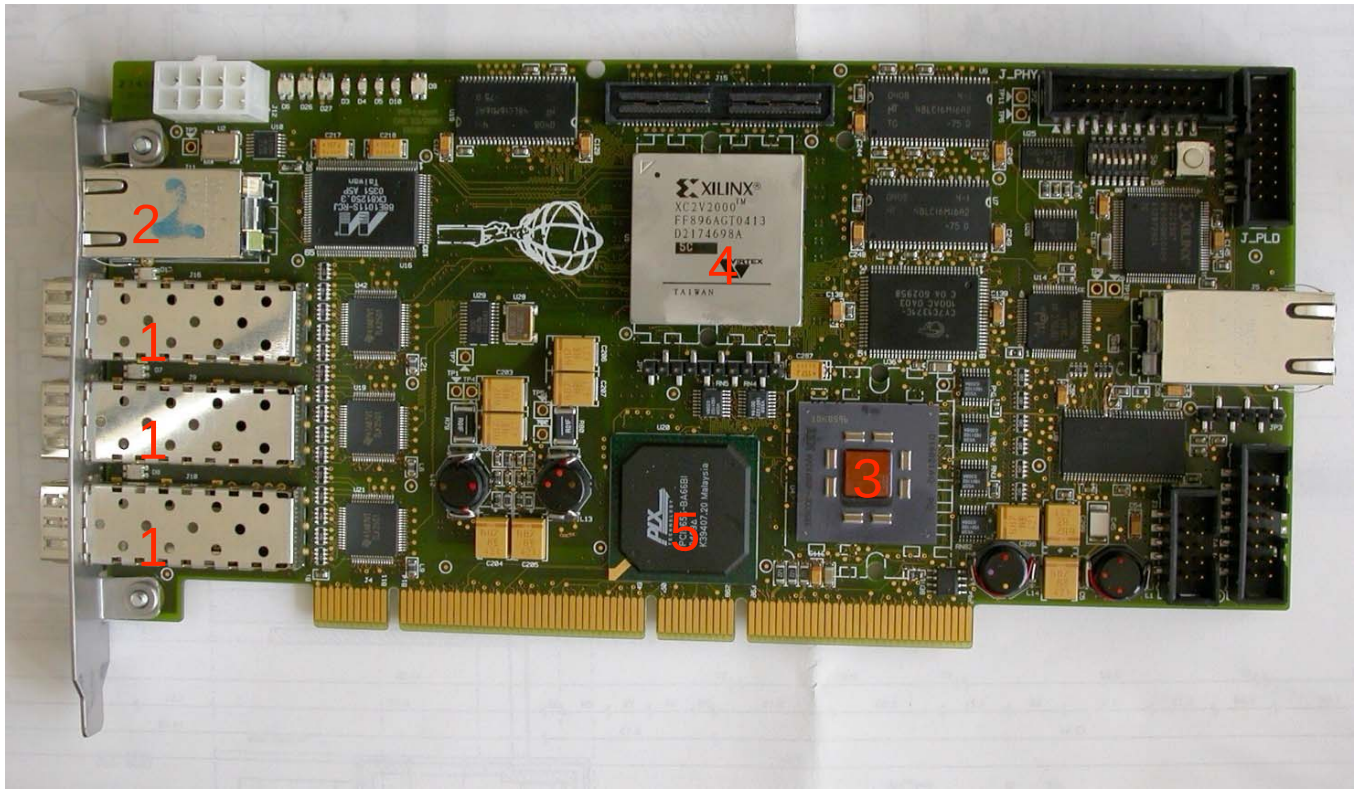
For first level trigger accept rate of 100 kHz



On average data requested from 2 ROLs for each ROS from which data is requested, per ROL at max. ~ 5 - 8 kHz (12 ROLs per ROS!)

Each dot represents a ROS PC

Final ROBIN: passed Production Readiness Review (PRR) on March 1, 2005.
350 to be produced in Germany, 330 in UK + 50 for "pre-series" (availability: summer 2005)
Autumn 2005: production should be finished.



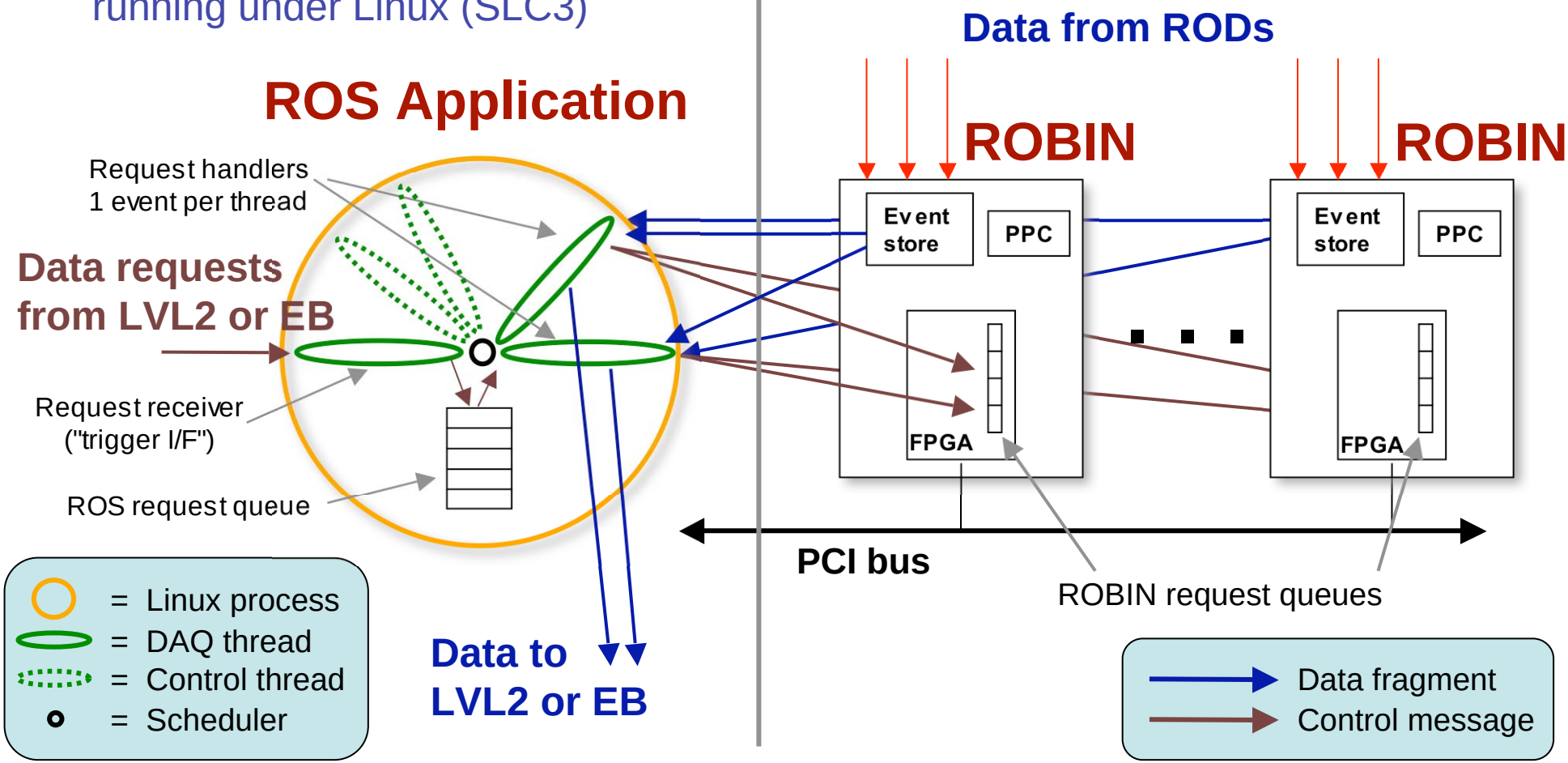
- 3 Read-Out Link channels (**1**) (200 MByte/s per channel),
  64 MByte buffer memory per ROL, electrical Gigabit Ethernet (**2**) ,
  PowerPC processor (466 MHz) (**3**), 128 MByte program and data memory,
  Xilinx XC2V2000 FPGA (**4**) , 66 MHz PCI-64 interface (**5**)
- 12 layer PCB, 220*106mm,
- Surface Mounted Devices on both sides
- Power consumption ~ 15W (operational)

DataFlow in a ROS PC

Prototype ROS - 3 GHz Xeon PC
(SuperMicro X5DL8-GG Motherboard)

Multi-threaded C++ program
running under Linux (SLC3)

PowerPC processor in ROBIN runs
C program booted from FLASH memory

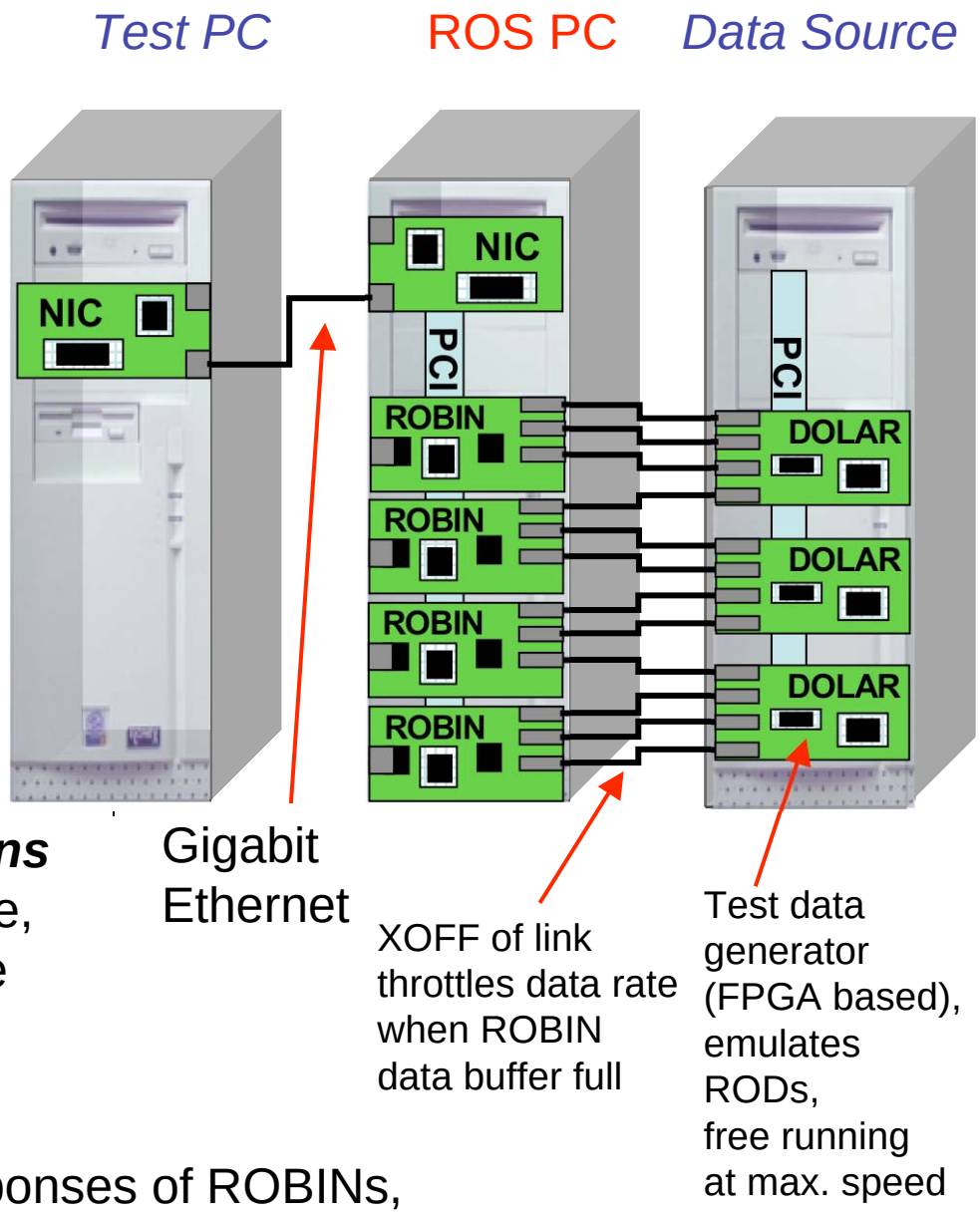


The application retrieves data fragments from the ROBINs,
combines them in a single fragment and sends this to the requester
Fragments have to be requested for each ROL individually

Measurements

Setup

- Test PC generates following messages:
  - *"**data request**"*, 2 types:
    - *"**LVL2**"* -> data requested from only few ROBIN channels
    - *"**EB**"* -> data requested from all ROBIN channel of the ROS
  - *"**event delete**"* (or *"**clear**"*) containing a list of 100 events

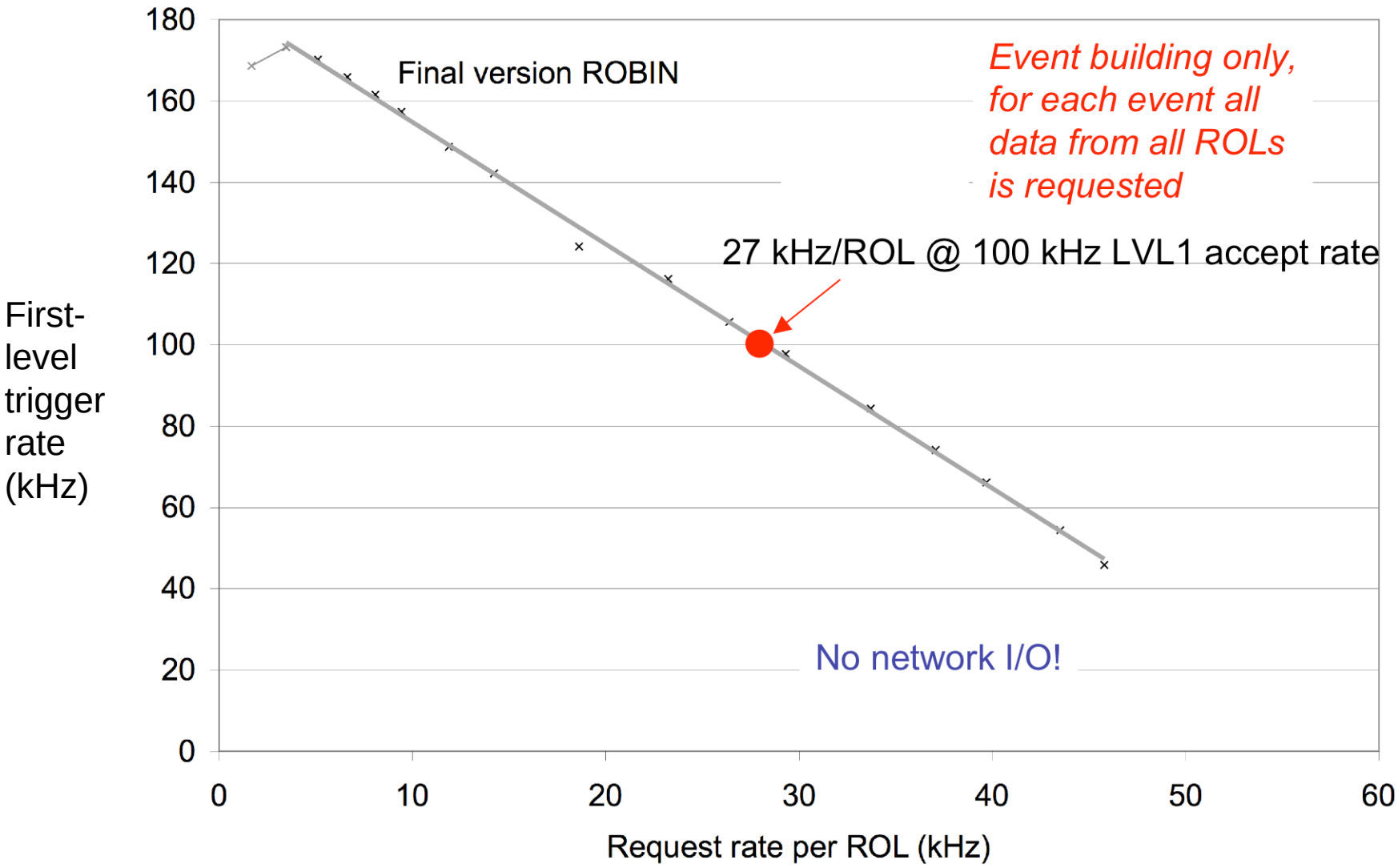


Philosophy

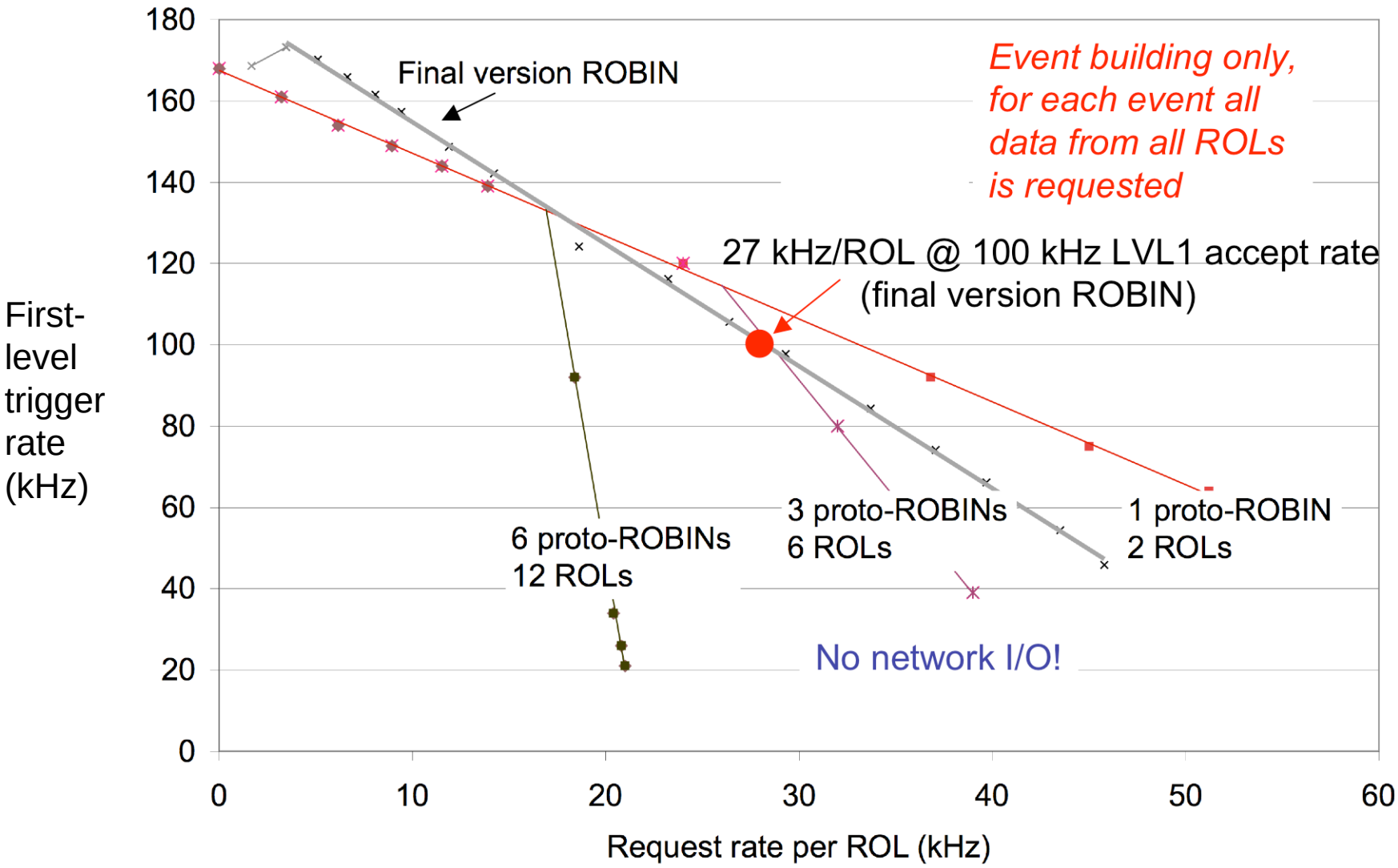
- Let the system run as fast as possible in different ATLAS-like conditions, defined by different ***LVL2 accept fractions***
- Measure the sustained "event delete" rate, this is the maximum first-level accept rate compatible with the LVL2 and EB rates following from the fractions chosen

Emulation: software in ROS emulates responses of ROBINs, calibrated with results from measurements with data generators, used in larger system tests.

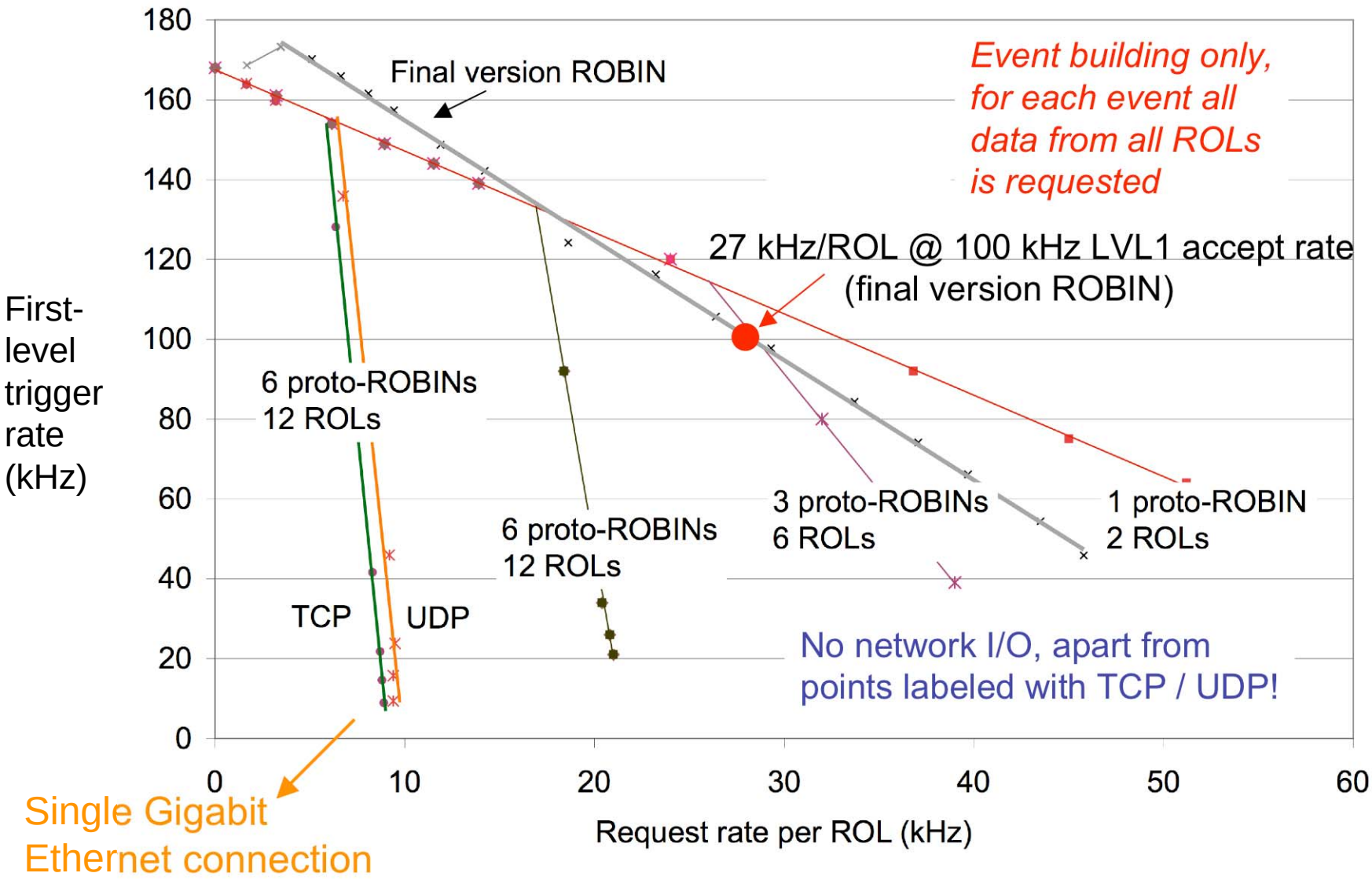
Results, Event Building only, PCI bus



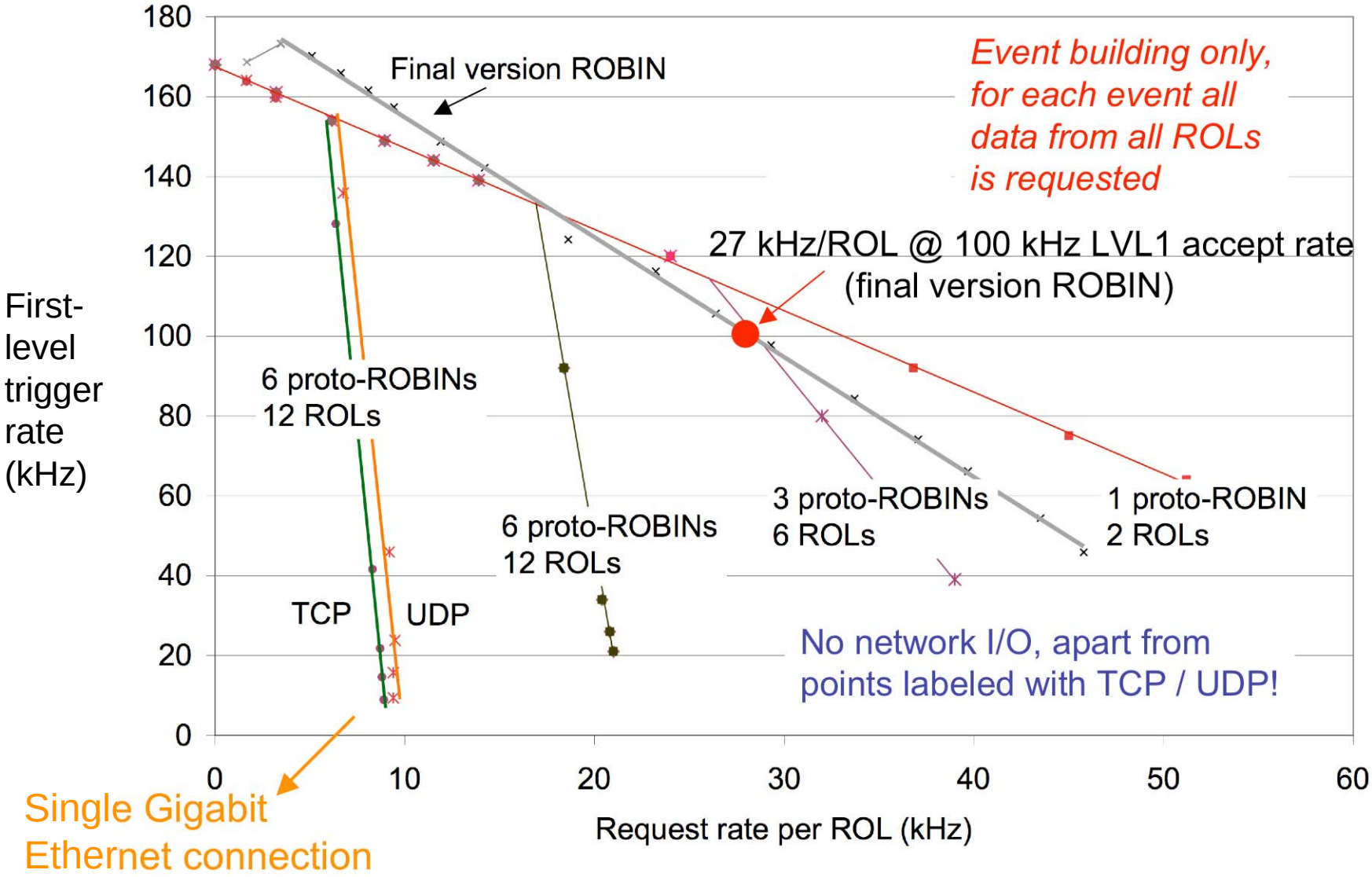
Results, Event Building only, PCI bus



Results, Event Building only, PCI bus



Results, Event Building only, PCI bus



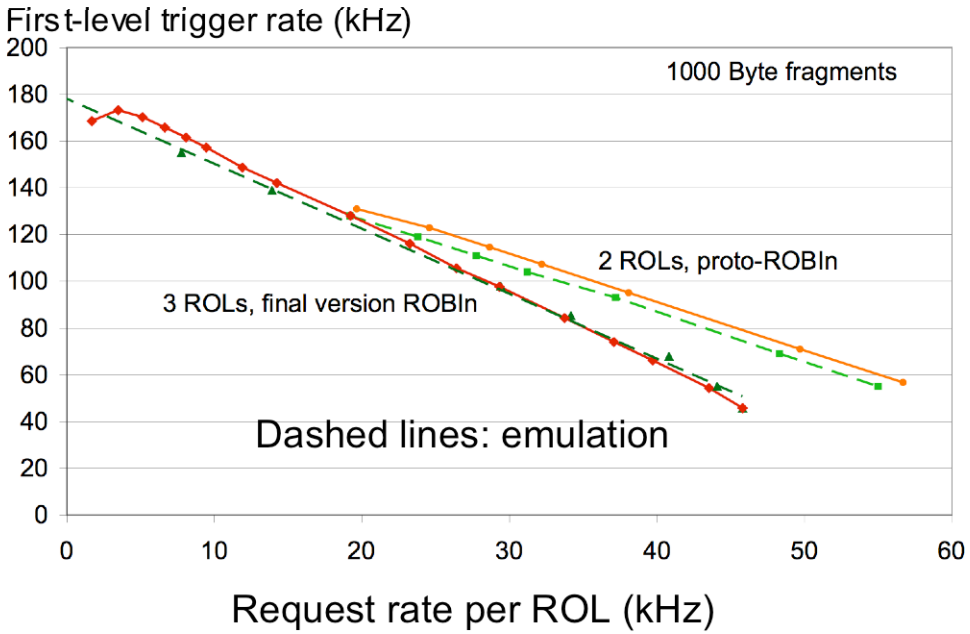
Requirements: from model: RoI requests: for **some** ROLs rate 5 - 8 kHz;
Event Building: **all** ROLs 3 - 3.5 kHz

Including non-standard triggers: max. req. rate ROBIN (RoI and EB requests): 21 kHz per ROL

Emulation

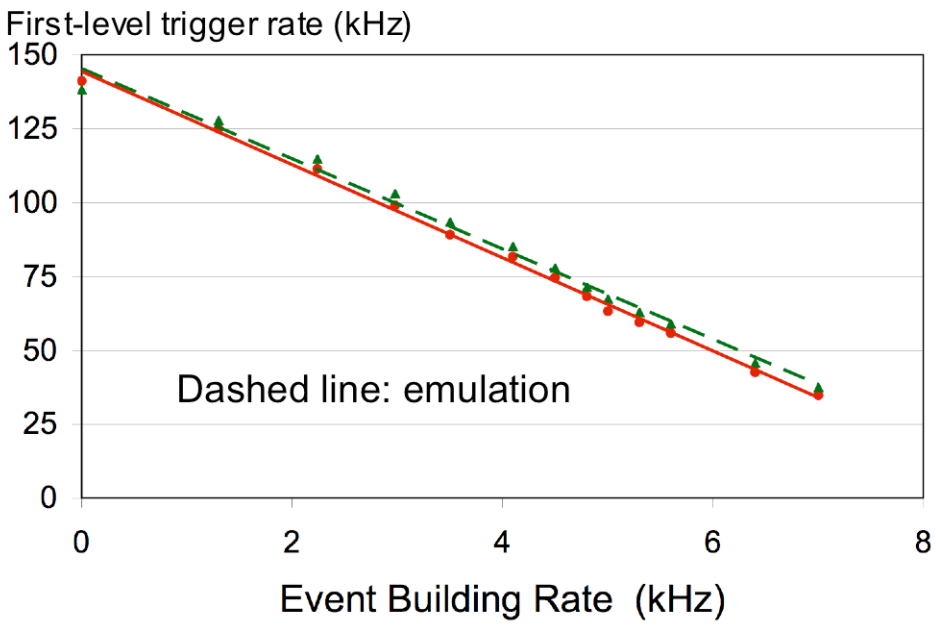
Emulation of prototype ROBIN and of final version

Without network



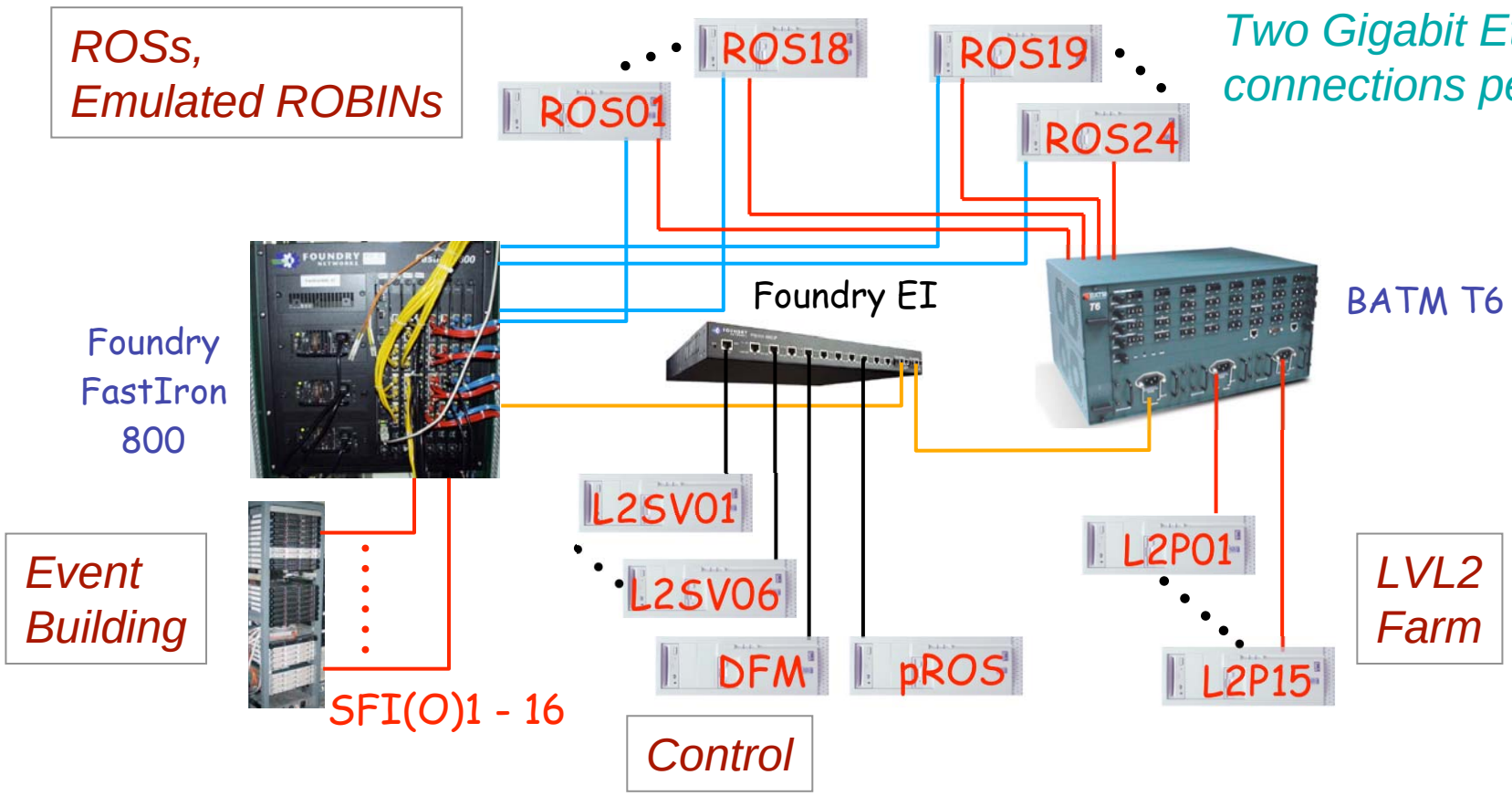
Emulation of ROS PC with 6 prototype ROBINs

TCP, 2.2% per ROL, 1.4 ROL/RoI



Measurement results for a ROS PC with 4 final version ROBINs (12 ROLs) are not yet available, but it has been emulated on the basis of the results shown above. The agreement between emulation and measurement show that this emulation can be assumed to be realistic.

Testbed

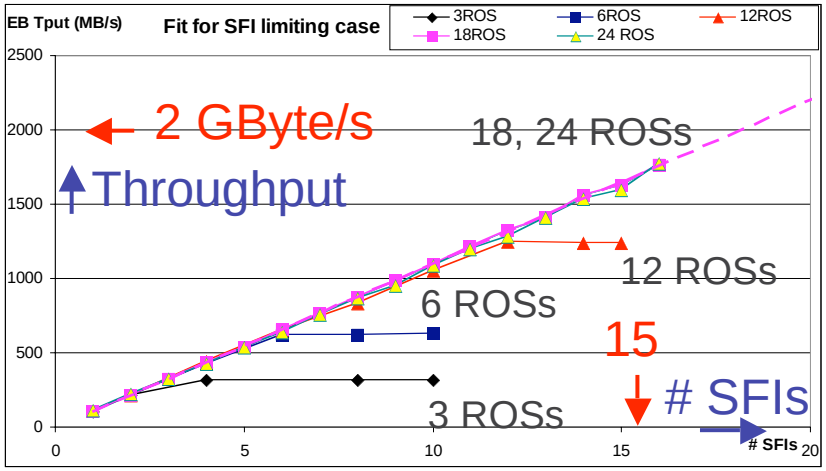


24 ROSs, 16 SFIs, 15 LVL2 processors (L2Ps)

L2Ps do not run algorithms,
15 correspond ~ to 150 in the full system, if ~ 10% of CPU load is for DataFlow
-> with 127 ROSs receiving requests from LVL2 and 60 - 100 SFIs in full system:
testbed size ~20% of full system

Results autumn 2004

- Event Builder throughput scales linearly with number of SFIs
- No show-stoppers as system size increases
- The number of SFIs needed in the full ATLAS system can be estimated from the results

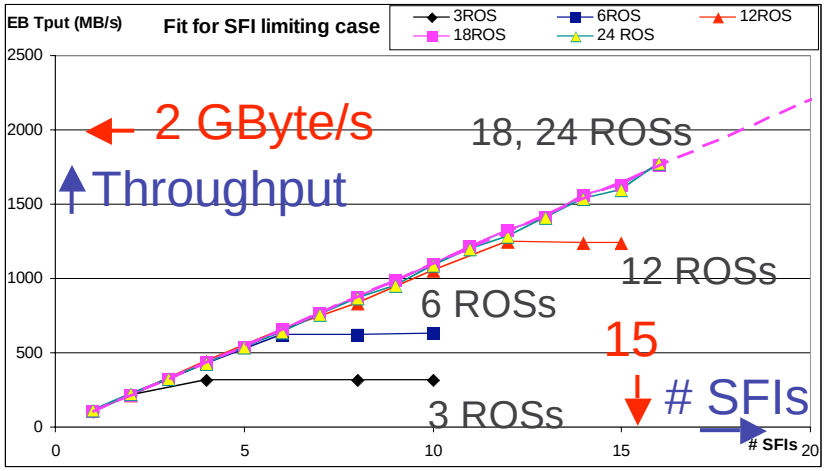


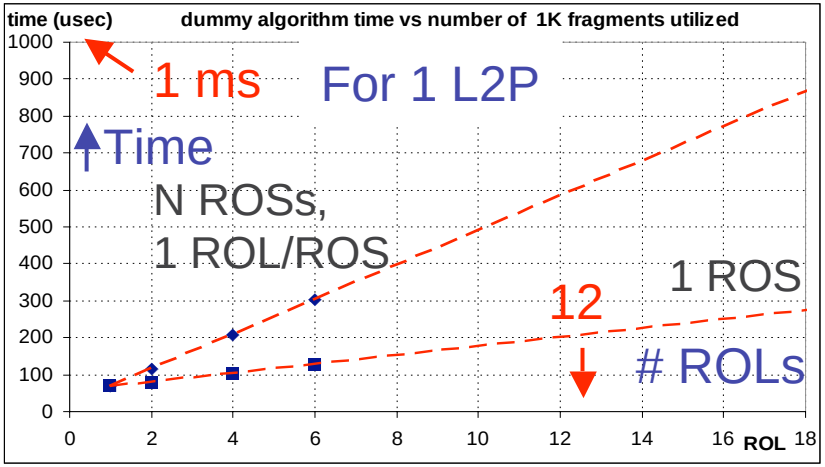
Results autumn 2004

- Event Builder throughput scales linearly with number of SFIs
- No show-stoppers as system size increases
- The number of SFIs needed in the full ATLAS system can be estimated from the results

- Linear dependence between time needed for ROI collection and number of ROSs
- Worst case ROI collection: ~16 ROLs from 16 ROSs) takes less than 10% of the time budget available per event (10 ms)

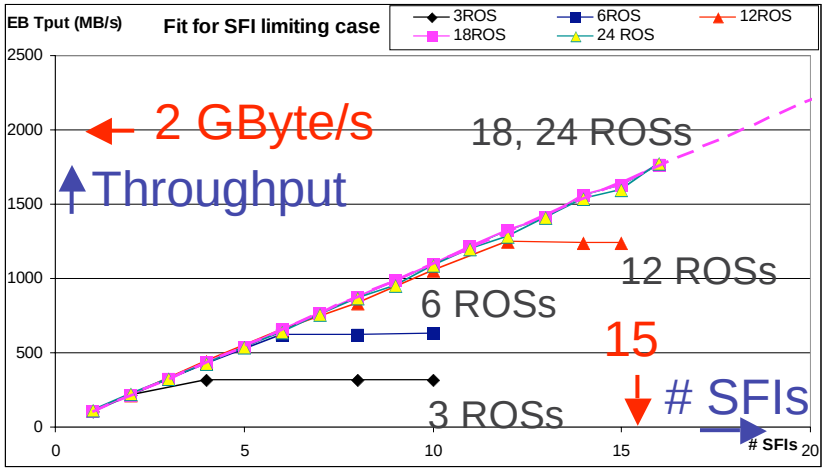
NB: 1 ROS -> 1 request
N ROSs -> N requests





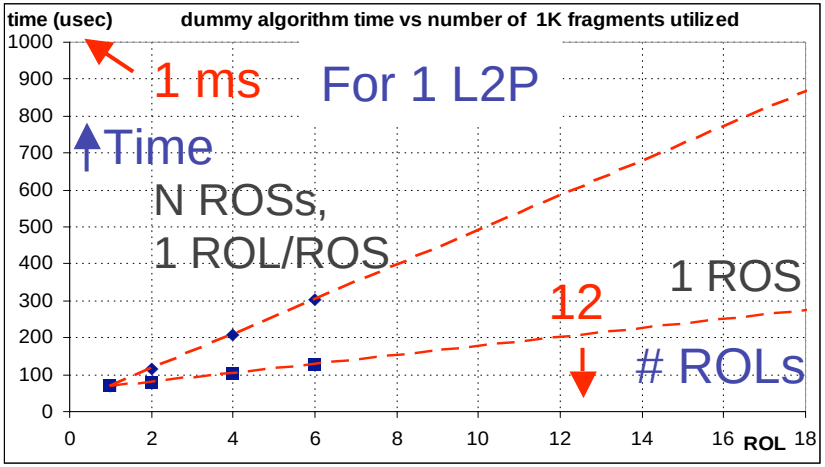
Results autumn 2004

- Event Builder throughput scales linearly with number of SFIs
- No show-stoppers as system size increases
- The number of SFIs needed in the full ATLAS system can be estimated from the results

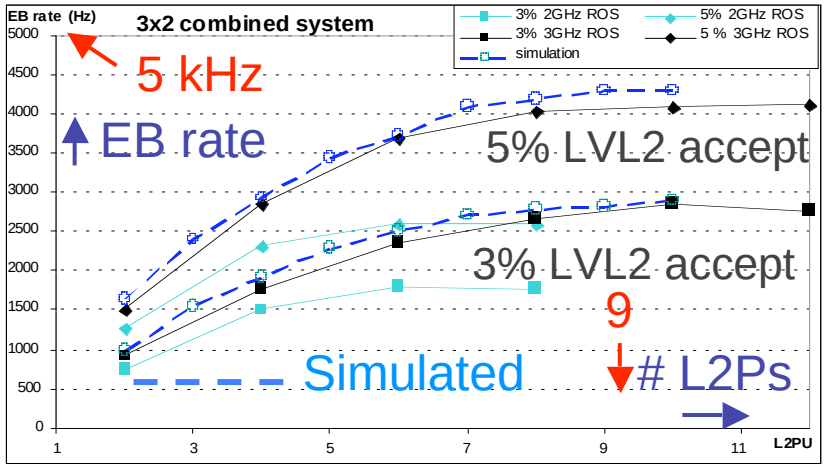


- Linear dependence between time needed for ROI collection and number of ROSs
- Worst case ROI collection: ~16 ROLs from 16 ROSs) takes less than 10% of the time budget available per event (10 ms)

NB: 1 ROS -> 1 request
N ROSs -> N requests



- Combined system results for different LVL2 accept rates were used to tune the discrete event model with which later the full ATLAS DataFlow system was simulated

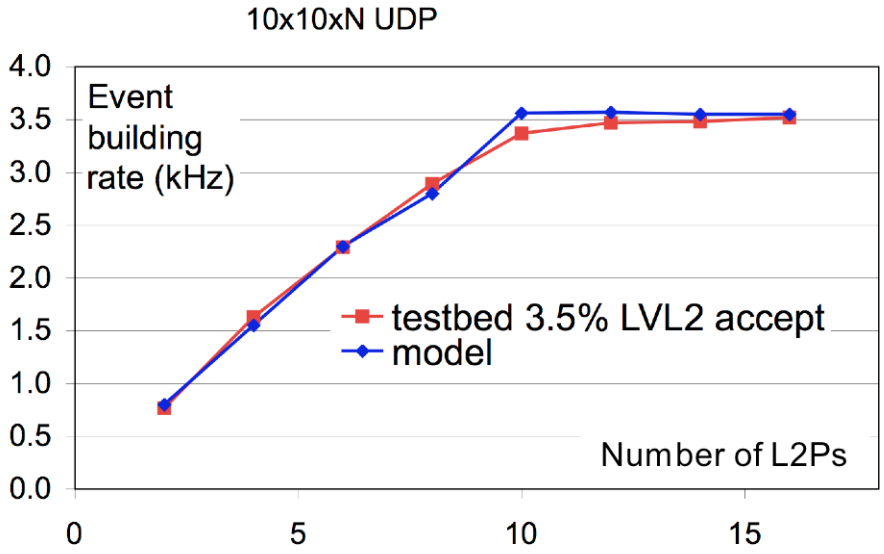


3 ROSs, 2SFIs, 1 ROL/ROS

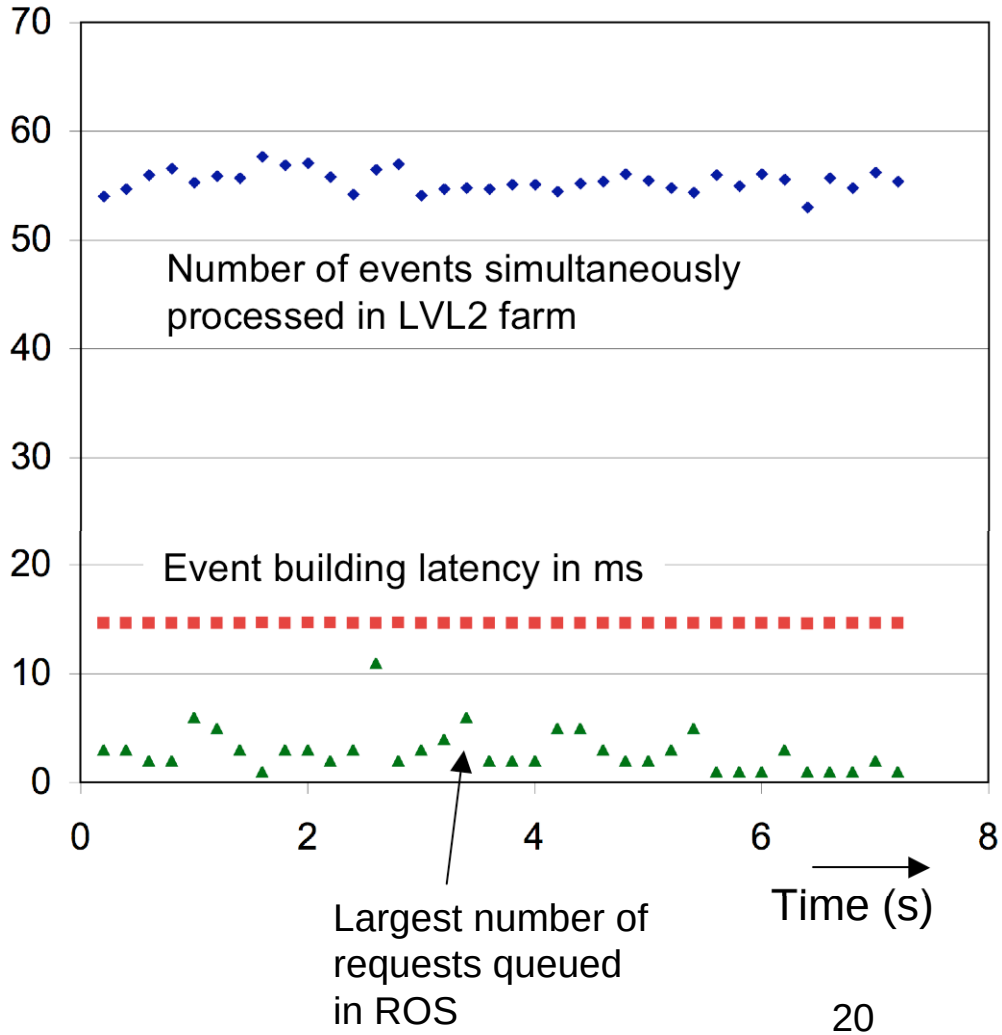
Latest measurement and modeling results, ROBIN emulation for final version ROBIN

Model: discrete event simulation ("at2sim", makes use of "Ptolemy")

Testbed



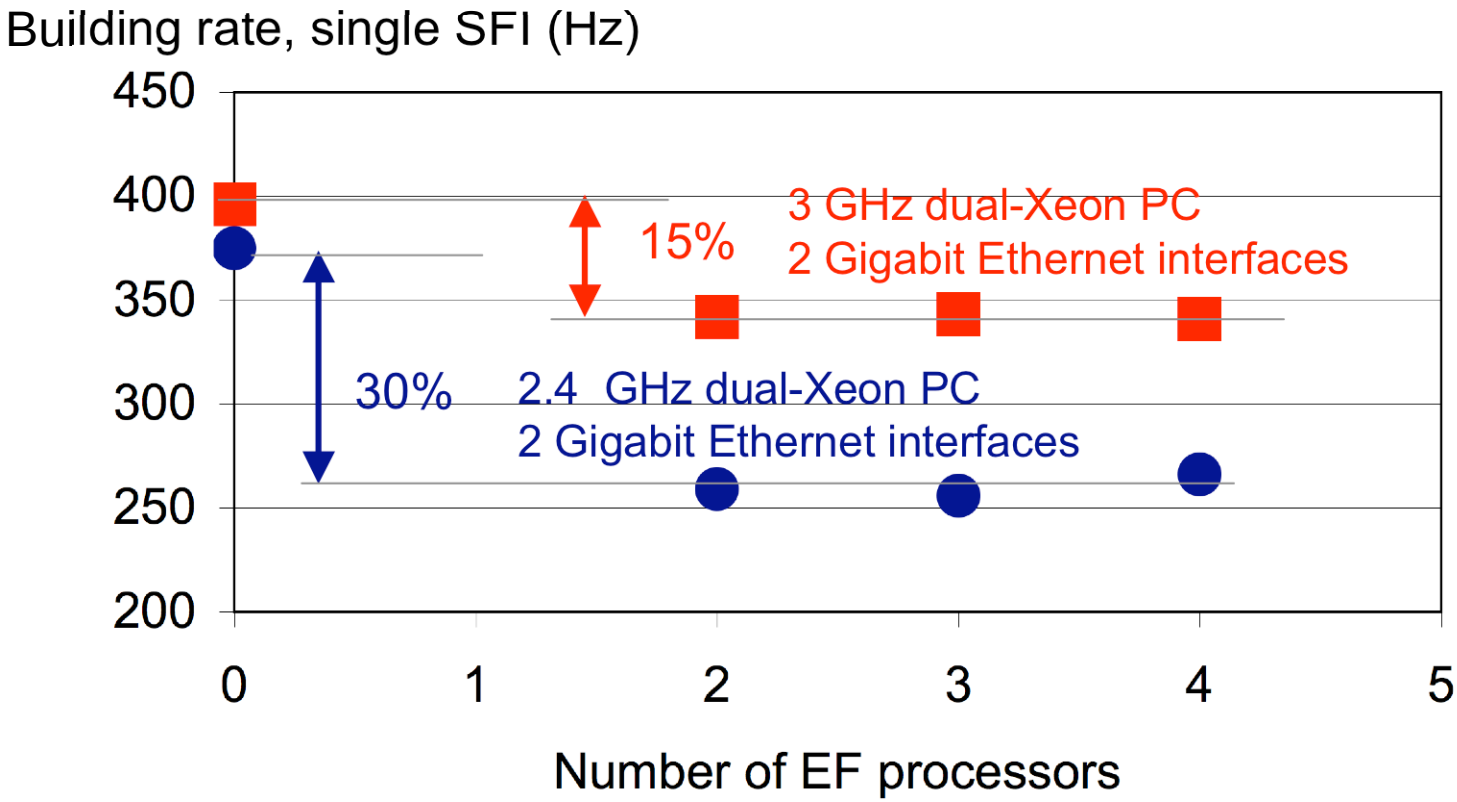
Full system



"Full system": 127 ROSs with final version ROBIN, 110 SFIs, 504 dual-CPU LVL2 processor units (L2Ps), 10 worker threads per L2P (as in testbed), 6 L2Ps per concentrating switch, mixed traffic, *detector mapping of detector onto ROLs and trigger menu and processing sequence as in paper model*, but algorithm execution times negligible(as in testbed).

Impact of output from SFI to Event Filter farm on event building rate

Event building rate for 12 ROSs with 12 ROLs,
fragment size 2048 Byte/ROL-> total 295 kByte per event,
-> 400 Hz: throughput ≈ Gigabit Ethernet bandwidth



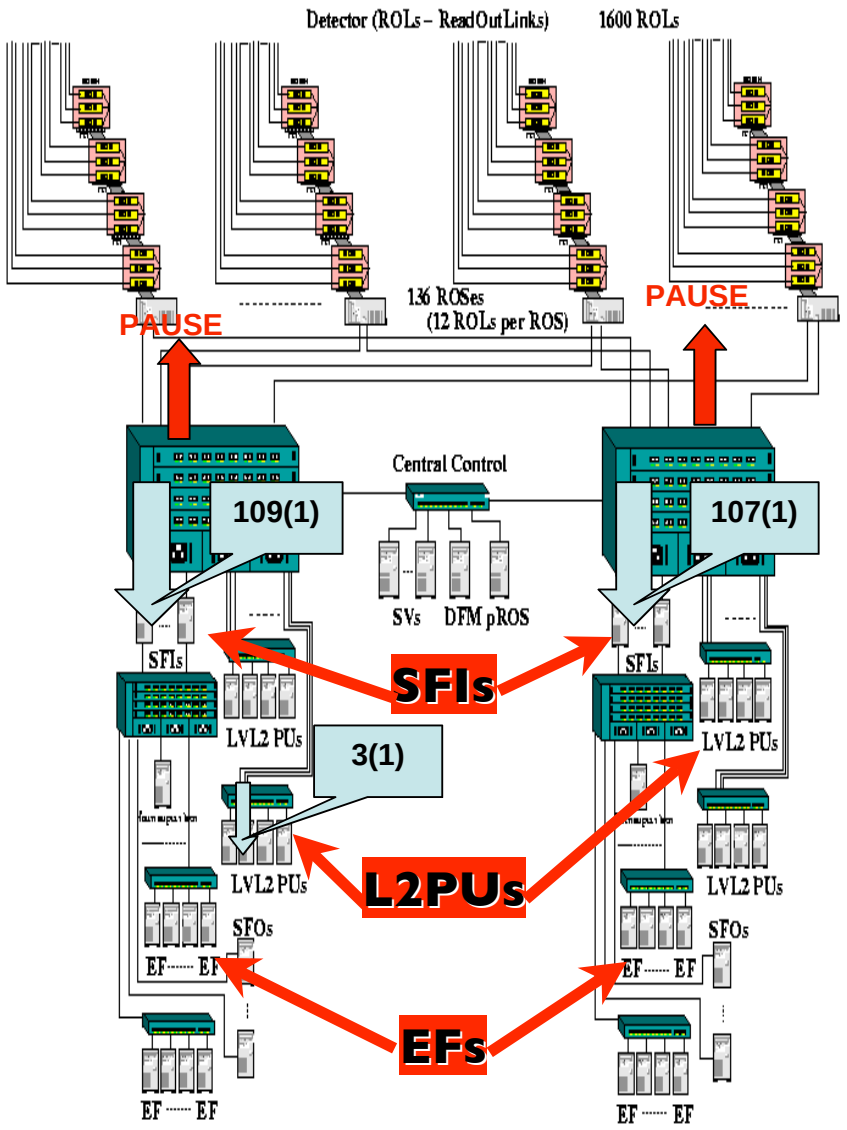
Note: in full system ~ 160 input messages per event instead of 12, still 1 output message per event

-> *expect smaller performance drop for full system*

Optimization: mixing of Event Builder & LVL2 traffic on the same switch

- *Better performance*
  - Traffic spread evenly over both Switches
  - Queues smaller in mixed network
- *More reliable system*
  - Failure of one switch: half of LVL2 and of Event Building system still available
- *More flexibility*
  - (Larger systems possible in test beds)

"Not mixed": LVL2 traffic via one central switch and one ROS network interface, EB traffic via the other central switch and other ROS network interface



Numbers: max. (average) queue size

Further optimization studies

- *Choice of compiler / compiler options*
  SFI rate ~ 6 (5.5) % higher for gcc3.2.3 (icc) with P4 optimization compared to gcc3.2.3 with PentiumPro optimization

- *Using an SMP (dual-processor) PC as ROS PC* instead of an single processor machine: only studied in emulation mode so far, same rate capability using "auto affinity", 5 .. 10% increase if "affinity" is used.

- *Choice of network protocol*:
  - UDP only => best for performance
  - UDP for data, TCP for control => same performance as above
  - TCP only => affects performance, due to load on CPU from OS TCP stack

- *System parameter tuning:*
  Modeling shows that improper setting of the configuration parameters may lead to significant performance degradation or even to system instability

- *LVL2 concentrating switch studies*: use of 10 Gbit Ethernet, poster S. Stancu

- *Network studies with network testers*: presentation M. Ciobotaru

Conclusions and outlook - I

- The performance of ROSs equipped with final version ROBINs need to be studied, confident that there are no problems

- Single-Xeon 3 GHz ROS, dual-Xeon 3.0 GHz SFI, dual-Xeon 3 GHz LVL2 processors and Gigabit Ethernet networking capable of delivering the DataFlow performance required

- The largest test system was 24 ROS x 16 SFI x 15 L2PU
  -> no scalability/functionality problems observed

- at2sim of the final setup:160 ROS x100 SFI x ..500 L2PU
  -> no surprises, no queues, no anomalies

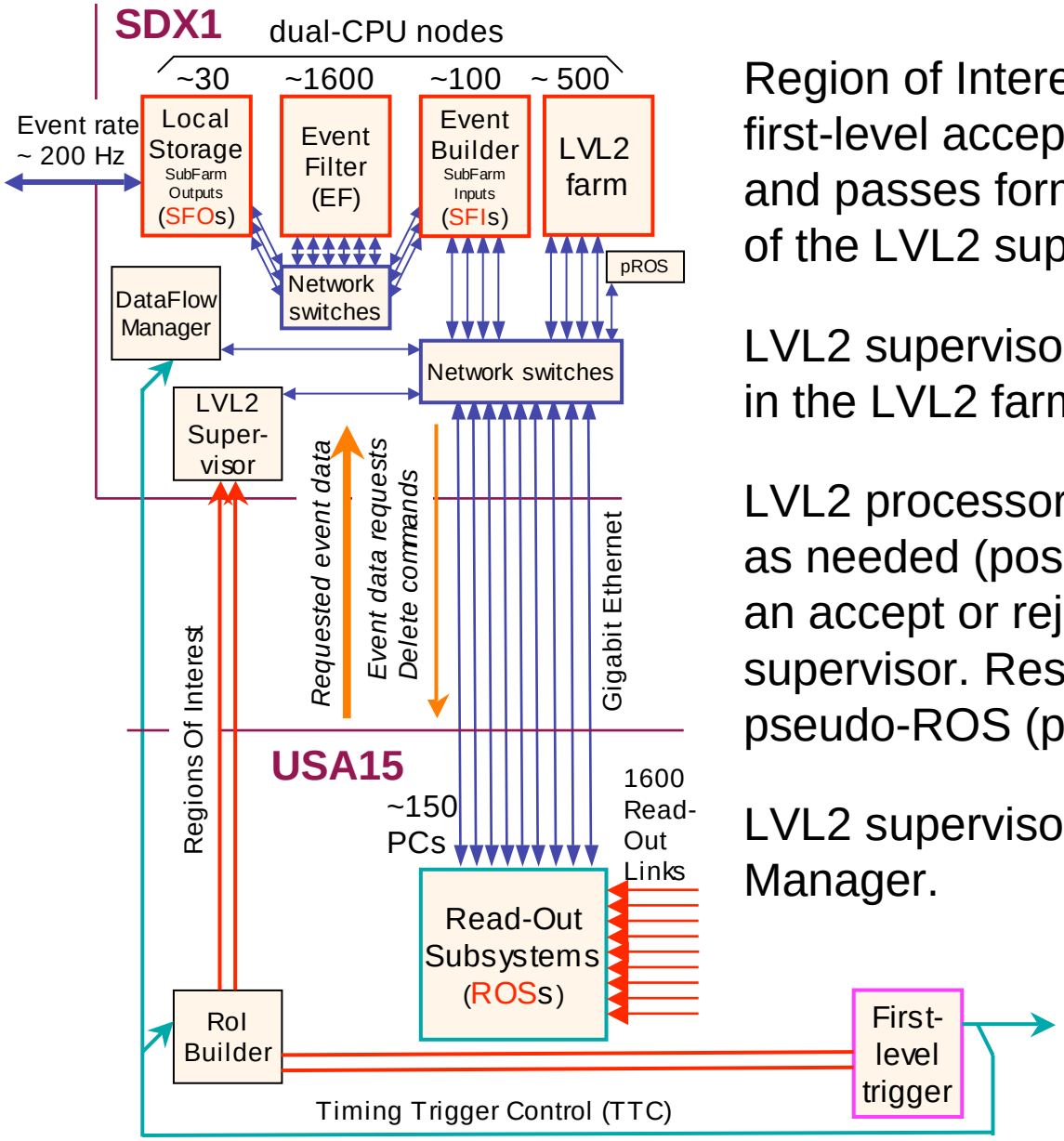
No show-stoppers

Conclusions and outlook - II

- SMP and thread affinity seems promising
- Not much improvement seen using alternative compilers to gcc v3.2.3, Pentium 4 versus Pentium Pro optimizations bring ~ 5% performance
- "Mixed" traffic results in lower LVL2 latency, smaller queues in the switches, more flexibility
- The event rates measured in the testbed do NOT change when using TCP for the dataflow control messages instead of UDP
- Study of application of the AMD Opteron instead of the Intel Xeon has started
- Installation and commissioning of the "preseries" started at "Point 1", to be followed by stepwise building up the full system

Backup slides

Trigger/DAQ DataFlow associated with second-level (LVL2) trigger



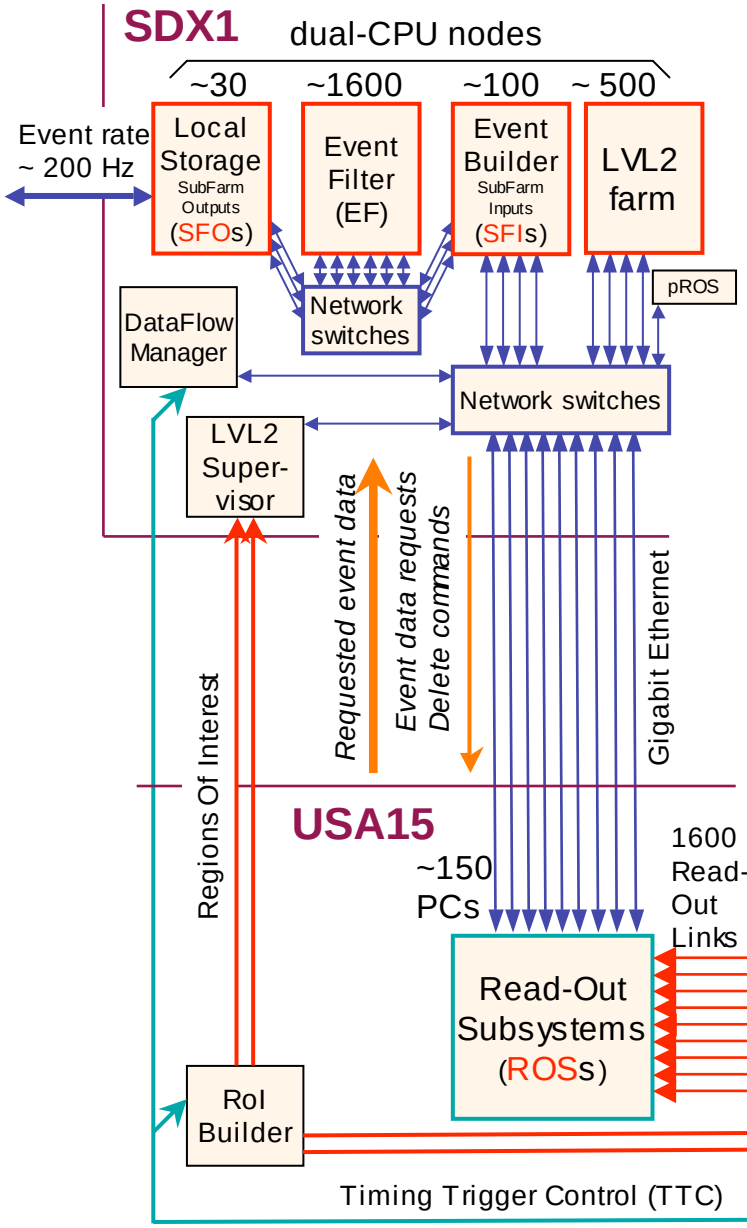
Region of Interest (RoI) Builder receives for each first-level accept information from first-level trigger and passes formatted information to one of the LVL2 supervisors.

LVL2 supervisor decides for one of the processors in the LVL2 farm and sends it the RoI information.

LVL2 processor requests data from the ROSs as needed (possibly in several steps), produces an accept or reject and informs the LVL2 supervisor. Result of processing is stored in pseudo-ROS (pROS) for an accept.

LVL2 supervisor passes decision to the DataFlow Manager.

Trigger/DAQ DataFlow associated with Event Building



For each accepted event the DataFlow Manager decides for a Sub-Farm Input (SFI) and sends it a request to take care of the building of a complete Event.

The SFI sends requests to all ROSs for data of the event to be built. Completion of building is reported to the DataFlow Manager.

For rejected events and for events for which event Building has completed the DataFlow Manager sends "clears" to the ROSs for 100 - 300 events together.

On request the event data is passed from SFI to an Event Filter processor.

Event Building rate ~ **3 - 3.5 kHz**

RoI request rates are estimated with the "paper model"

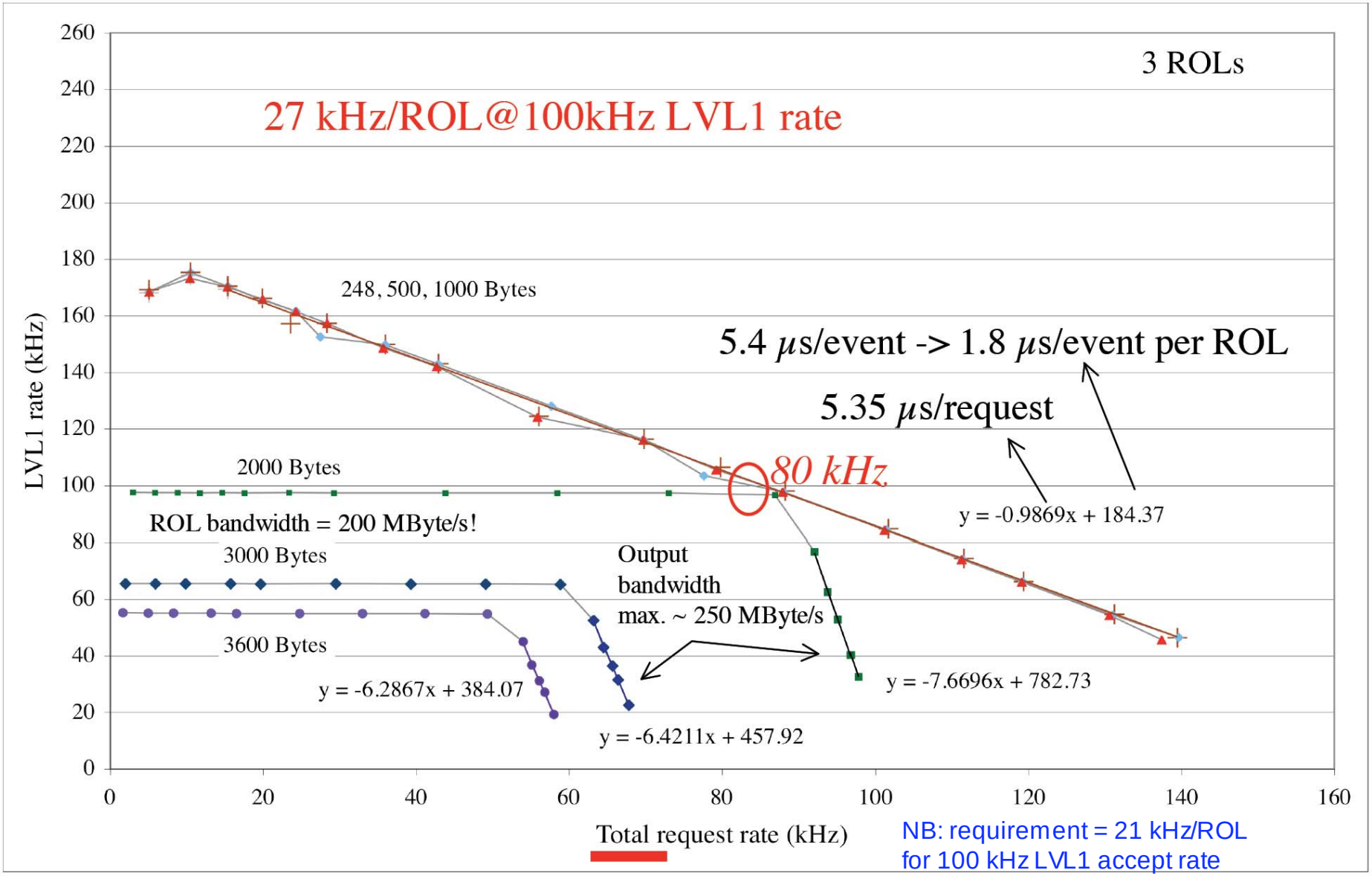
"Paper" -> "back-of-the-envelope" calculations
In practice: C++ program (earlier: spreadsheet).

Basic assumption: RoI rate does not depend on the η and ϕ of the centre of the RoI, only on the area in η-ϕ space associated with the RoI.

The RoI rates are obtained with a straightforward calculation using:
- the LVL1 accept rate,
- exclusive fractional rates for the various LVL1 trigger menu items,
- the number of RoIs associated with each trigger item,
- the η-ϕ area associated with each possible RoI location.

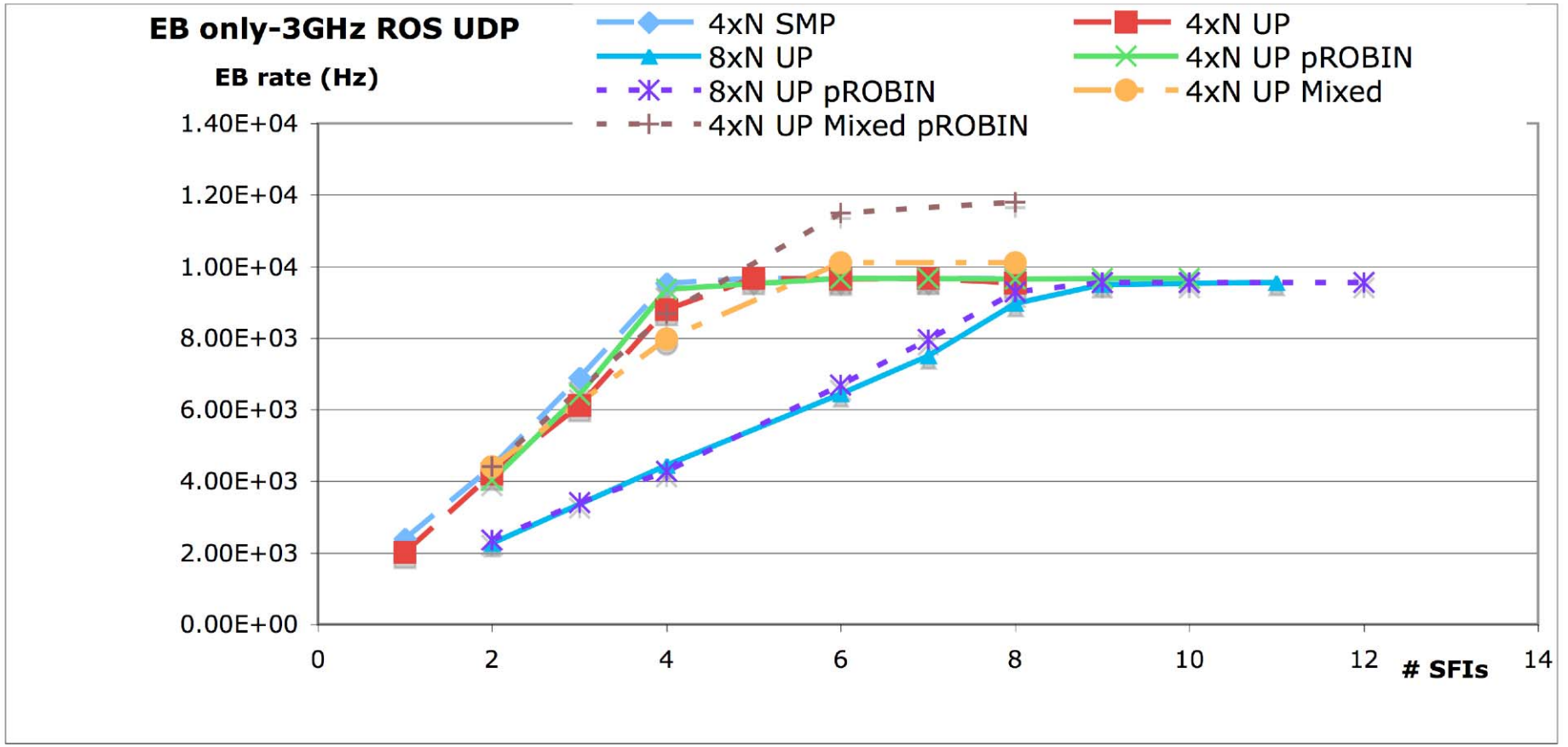
The request rates are then obtained by summing the contributions of all possible RoI locations using:
- information of the mapping of the ROLs onto the detector,
- the acceptance factors of the various LVL2 trigger steps,
- the η-ϕ areas from which data is requested (RoI and detector dependent).

Stand-alone measurements, 3 ROLs active, PCI bus

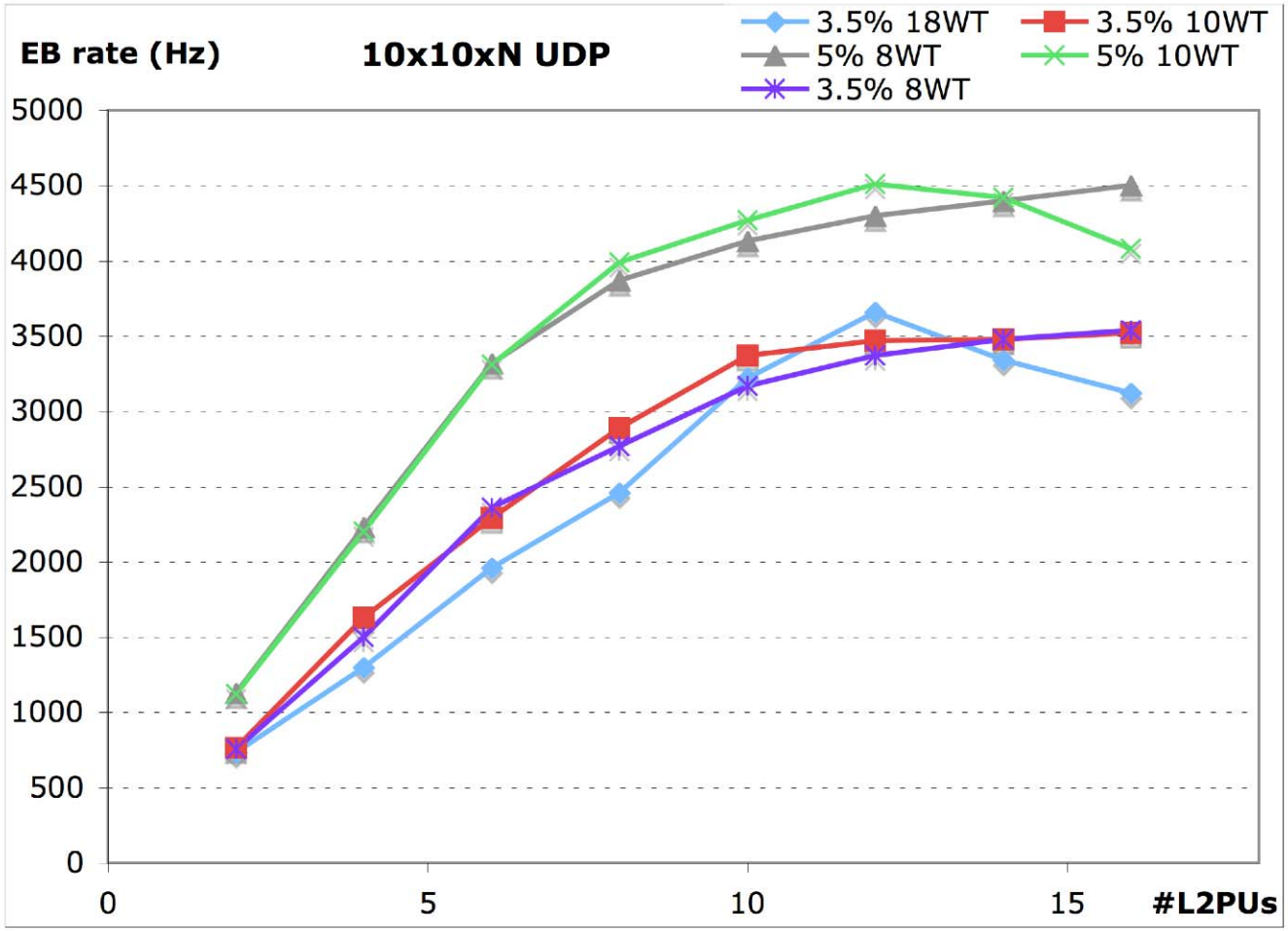


NB: requirement = 21 kHz/ROL
for 100 kHz LVL1 accept rate

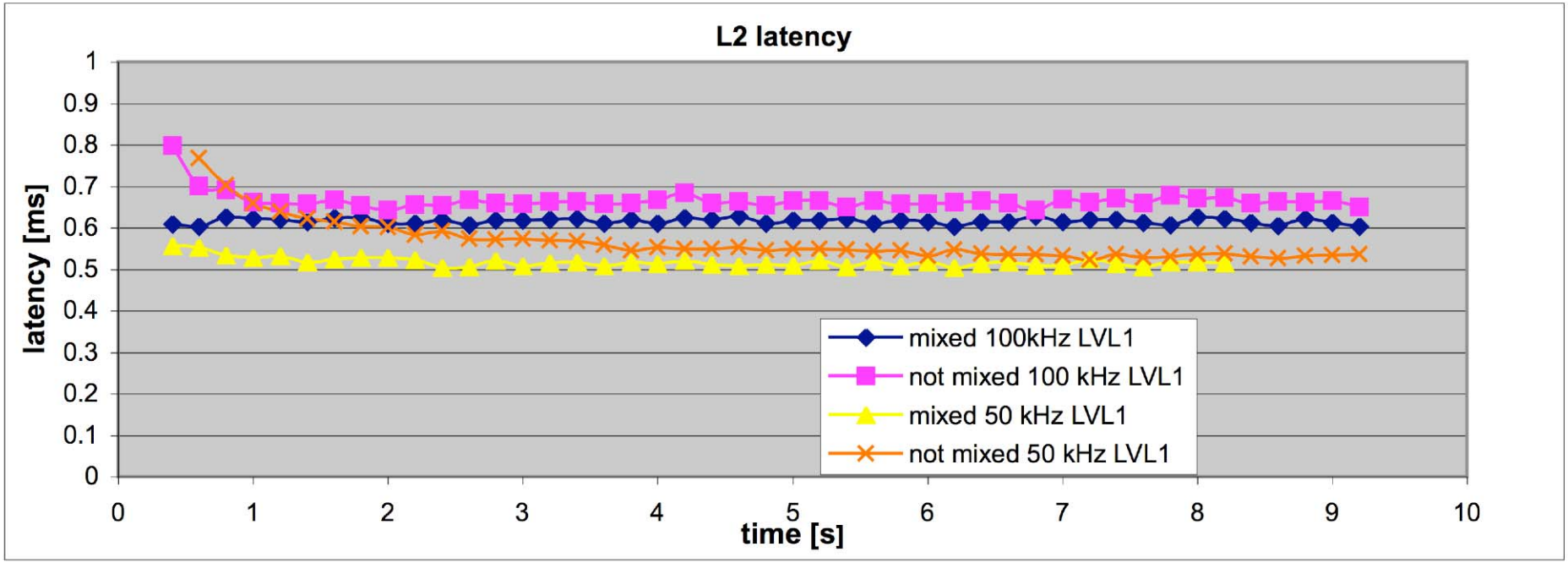
Event Building only - mixed network



Event Building and LVL2 traffic with emulation of final version ROBIN



Full system optimization results with emulation of prototype ROBIN



"Not mixed": LVL2 traffic and EB traffic via different switches and via different ROS network interfaces