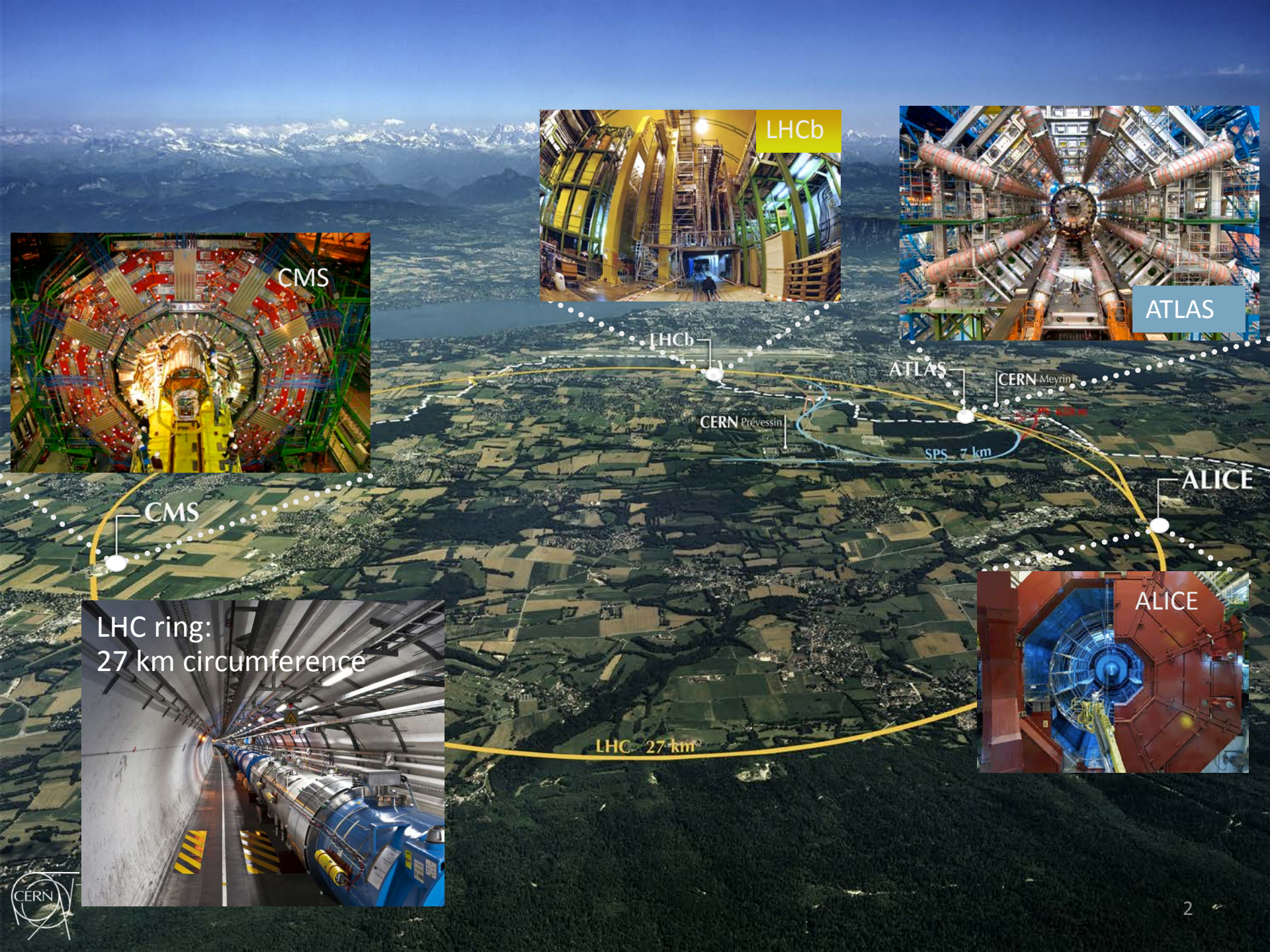
Improving Packet Processing Performance of a Memory-Bounded Application

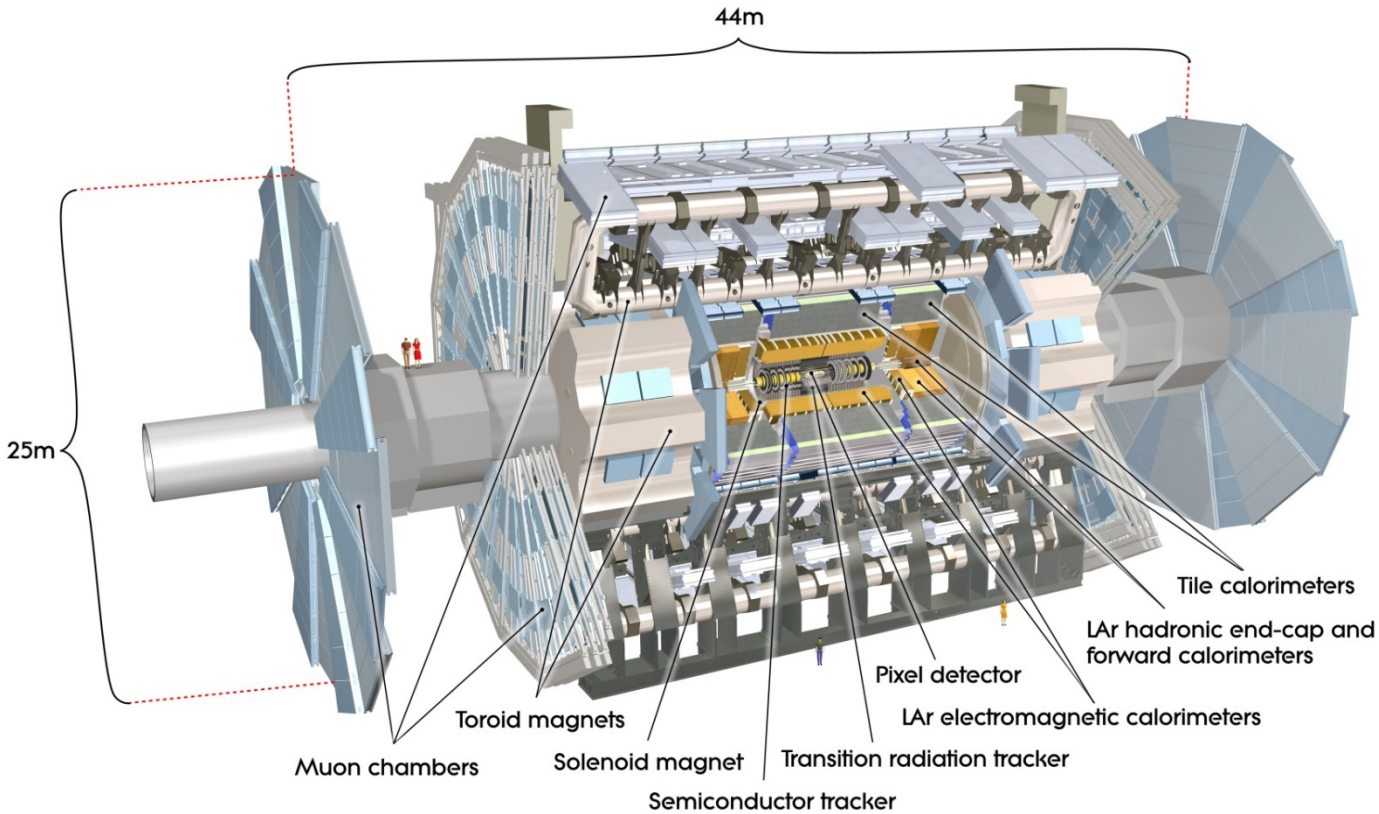
Jörn Schumacher
CERN / University of Paderborn, Germany
jorn.schumacher@cern.ch

On behalf of the ATLAS FELIX Developer Team


LHCb
CMS
ATLAS
LHCb
ATLAS
CERN Meyrin
CERN Prévessin
SPS 7 km
ALICE
CMS
ALICE
LHC ring:
27 km circumference
LHC 27 km

ATLAS EXPERIMENT

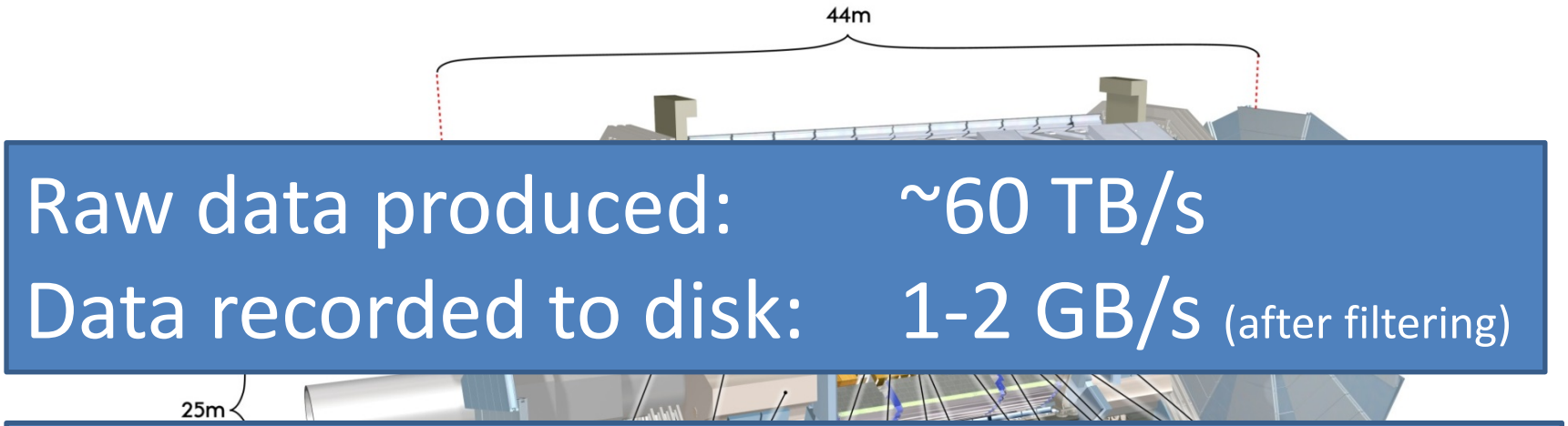
Channels: $100 \cdot 10^6$
Weight: 7000t
Collaborators: >3000



ATLAS: General-purpose particle detector, designed to observe phenomena involving high-energy particle collisions

ATLAS EXPERIMENT

Channels: $100 \cdot 10^6$
Weight: 7000t
Collaborators: >3000



Raw data produced: ~60 TB/s
Data recorded to disk: 1-2 GB/s (after filtering)

Data collection, selection, processing, monitoring, etc. requires tens of thousands of distributed applications, processing in quasi-realtime

ATLAS: General-purpose particle detector, designed to observe phenomena involving high-energy particle collisions

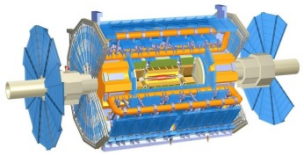
In this talk

1. Integration of a new Distributed Event-Based System in the ATLAS experiment

2. Experience in analysis and optimization of a packet-based software

ATLAS DAQ:

40 MHz



Detector Cavern

FE | FE | FE | FE | FE — Frontend

Service Cavern

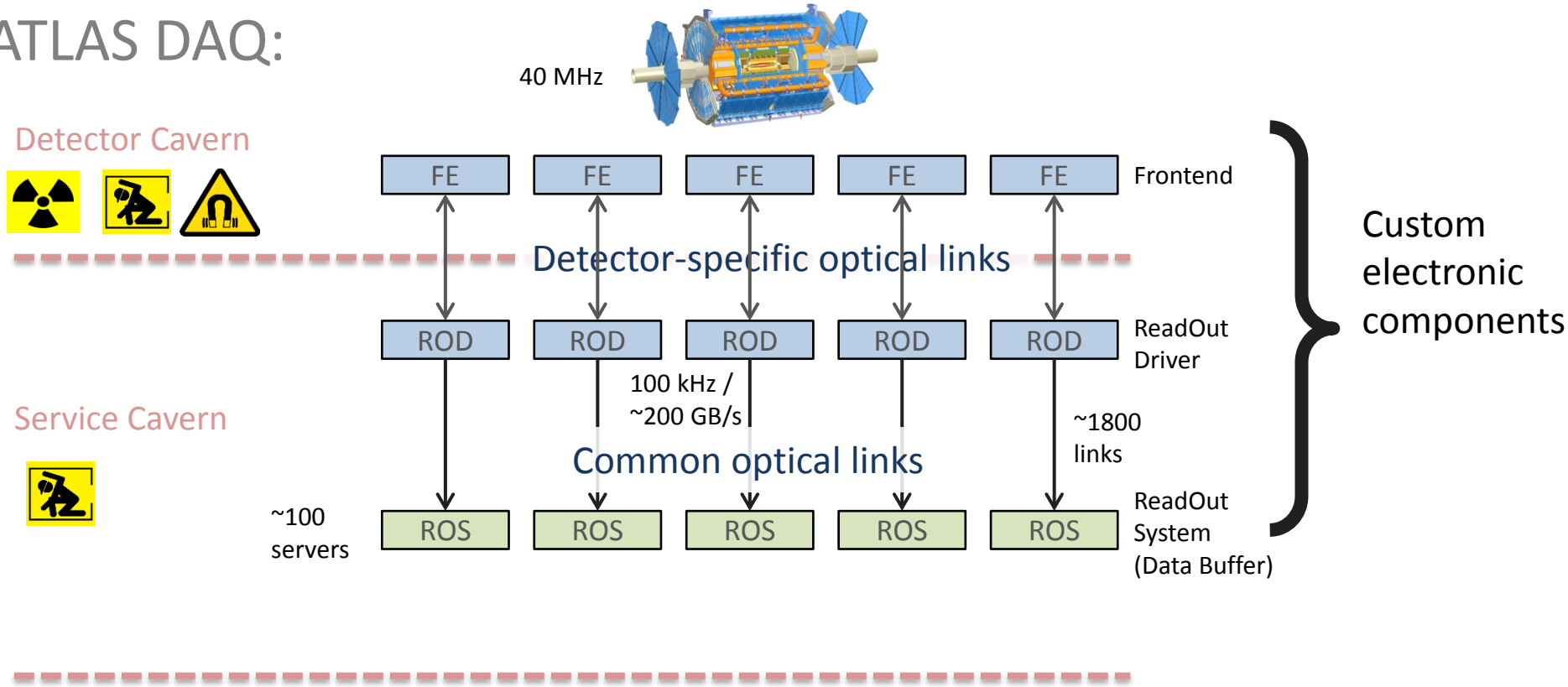
Datacenter

Today

Run 2 — Run 3 — Run 4

2015 — 2017 — 2019 — 2021 — 2023 — 2025

ATLAS DAQ:

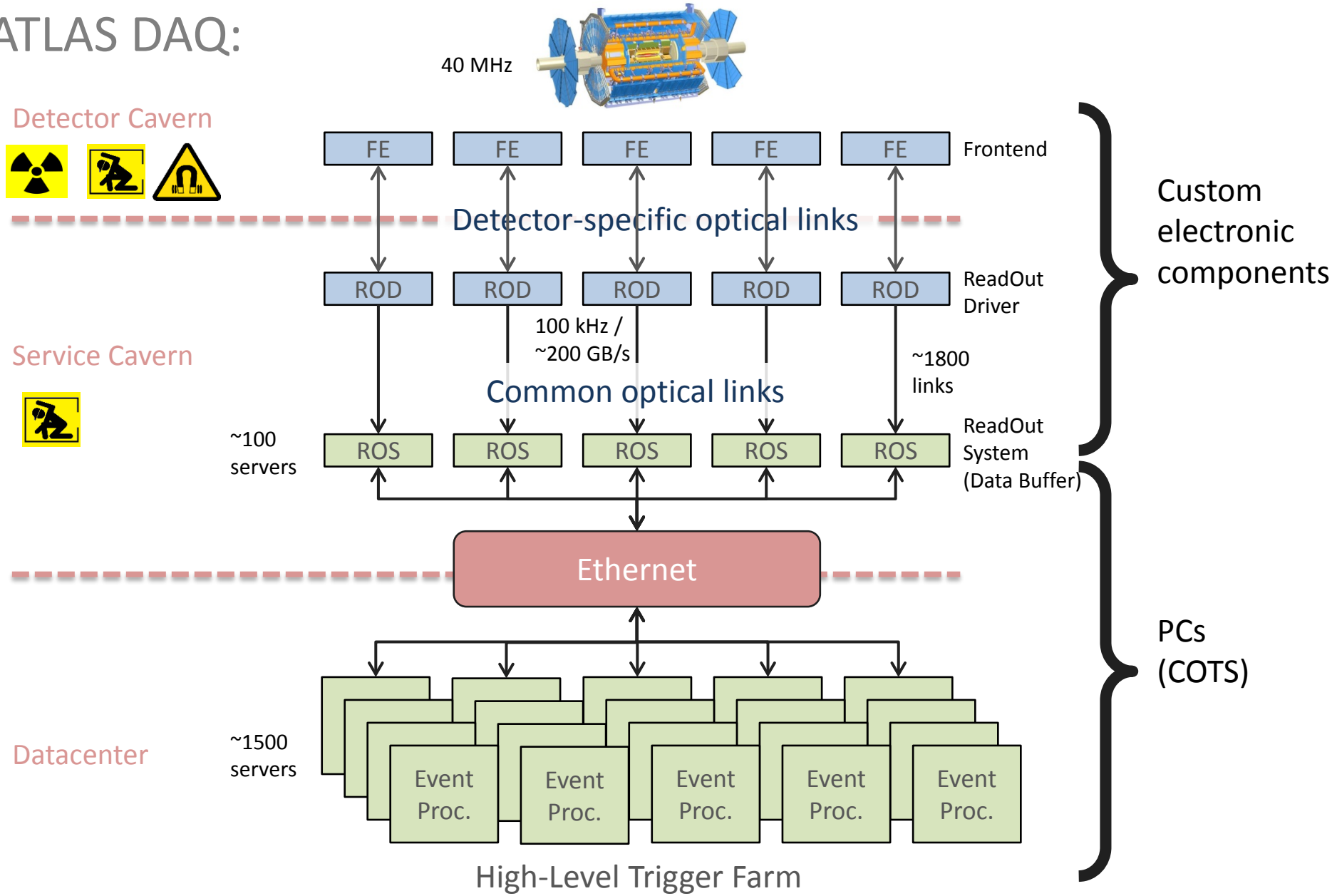


Today

Run 2 Run 3 Run 4

2015 2017 2019 2021 2023 2025

ATLAS DAQ:

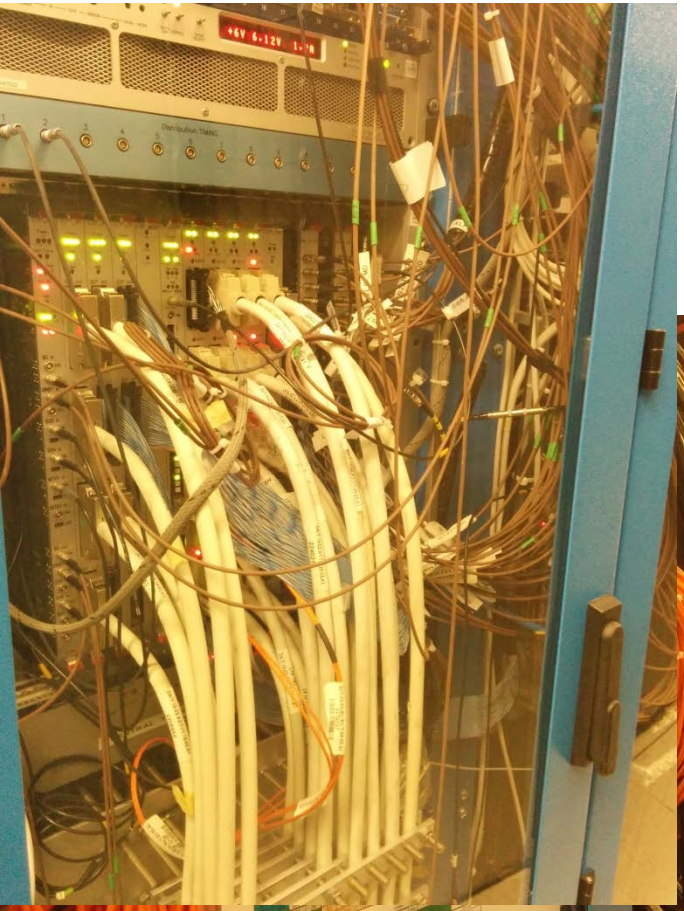


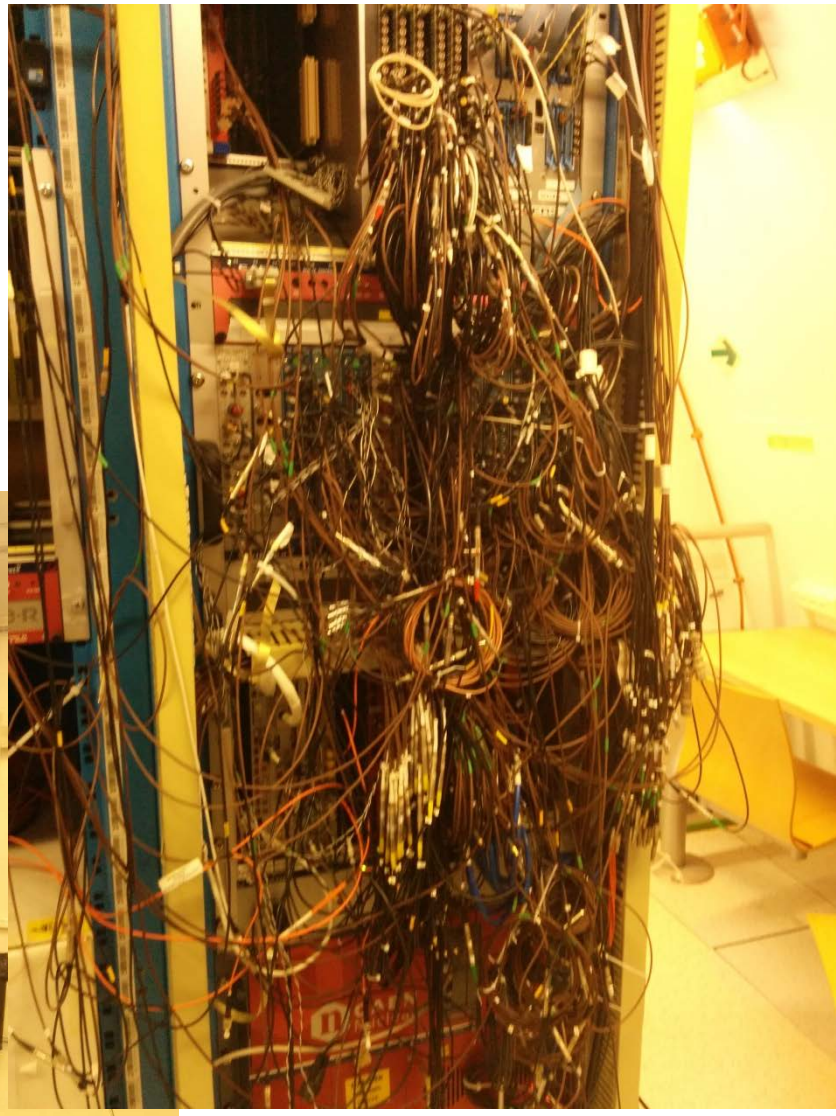
Today

Run 2 — Run 3 — Run 4

2015 — 2017 — 2019 — 2021 — 2023 — 2025

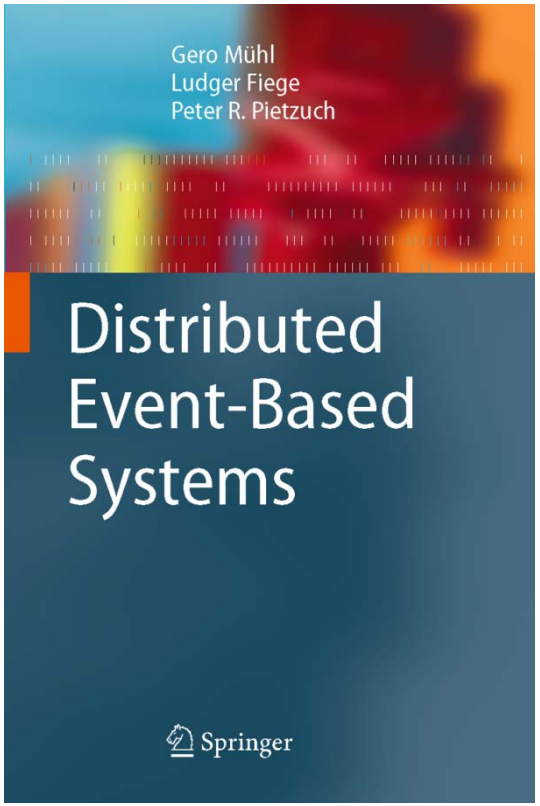
Custom Electronics





Challenging maintenance and operation:
Advantages in using COTS components
(PC Technology)

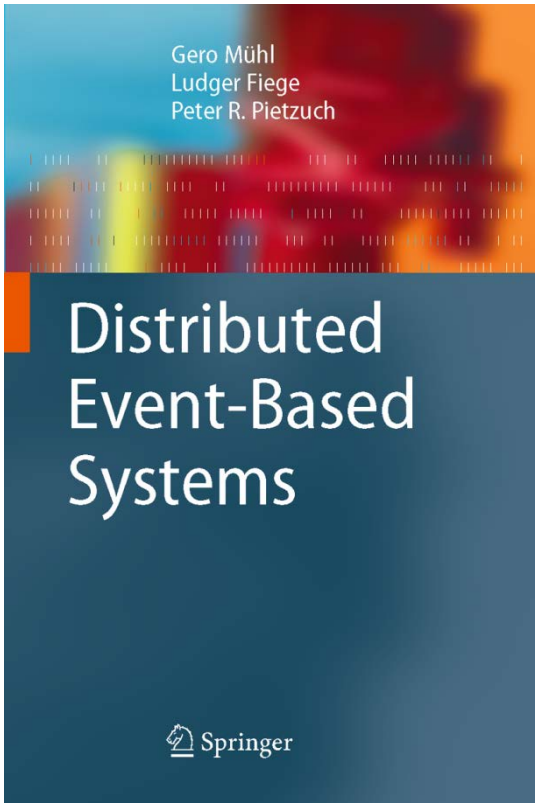
Event Processing in Software



"In an event-based mode of interaction components communicate by generating and receiving **event** notifications […] An **event notification service** […] mediates between the components of an event-based system (EBS) and conveys notifications from **producers** […] to **consumers** […]

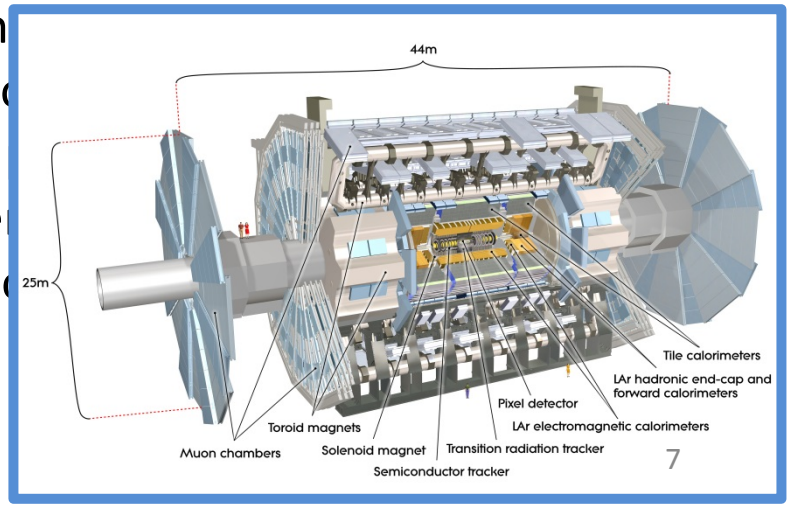
[…] **The notification service decouples the components** so that producers unaware of any consumers and consumers rely only on the information published, but not on where or by whom it is published. […] The event-based style carries the potential for easy integration of autonomous, heterogeneous components into complex systems that are easy to evolve and scale."

Event Processing in Software

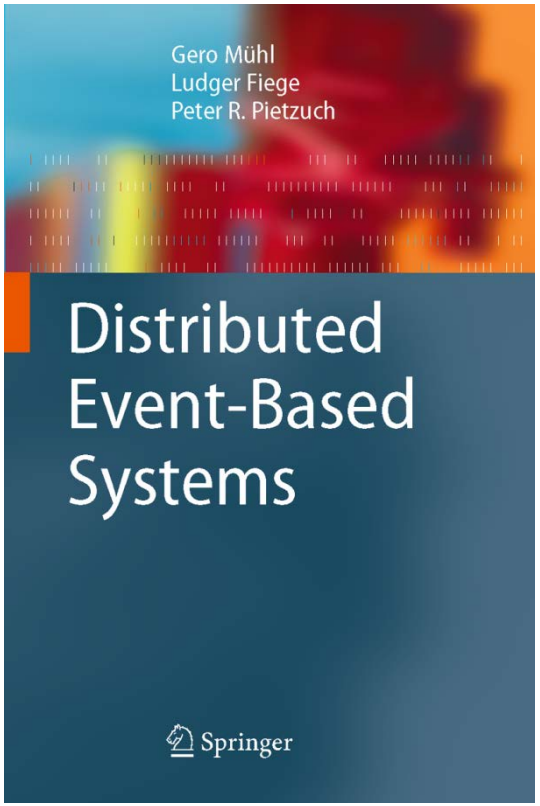


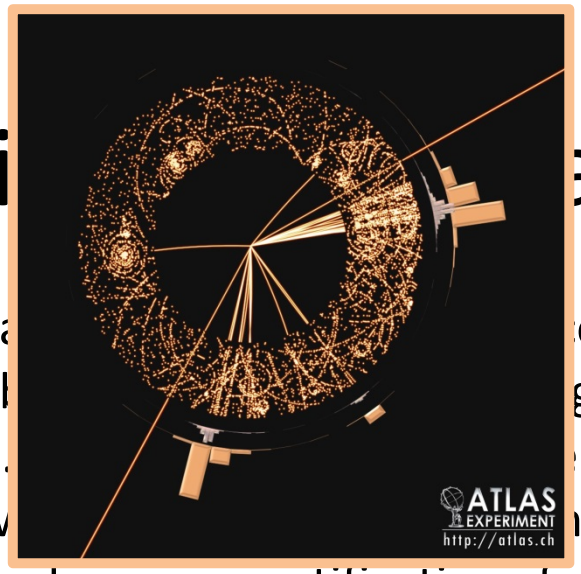
"In an event-based mode of interaction components communicate by generating and receiving **event** notifications [...] An **event notification service** [...] mediates between the components of an event-based system (EBS) and conveys notifications from **producers** [...] to **consumers** [...]

[...] **The notification service decouples the components** so that producers unaware of any consumers and consumers rely only on
not on where or by wh
based style carries the
autonomous, heteroge
systems that are easy t



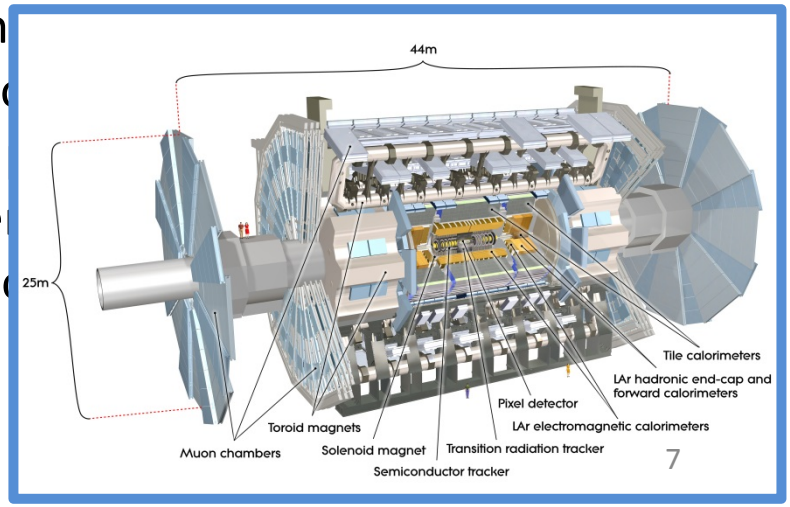
Event Processi are



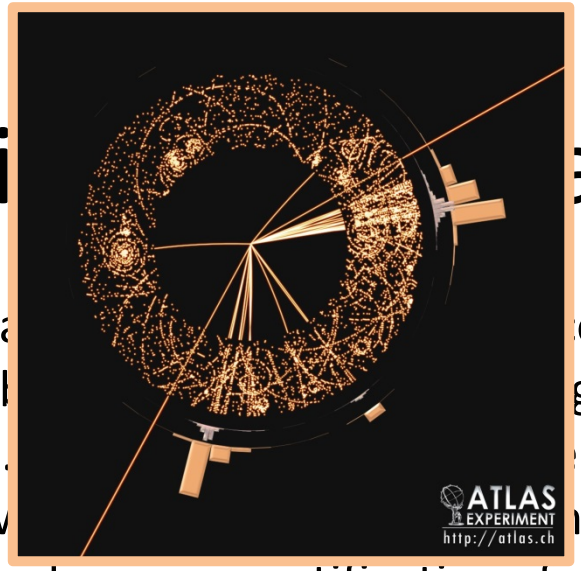


"In an event-ba omponents communicate b g **event** notifications [… rvice […] mediates betw event-based system (EBS) and conveys notifications from **producers** […] to **consumers** […]

[…] **The notification service decouples the components** so that producers unaware of any consumers and consumers rely only on not on where or by wh based style carries the autonomous, heteroge systems that are easy t

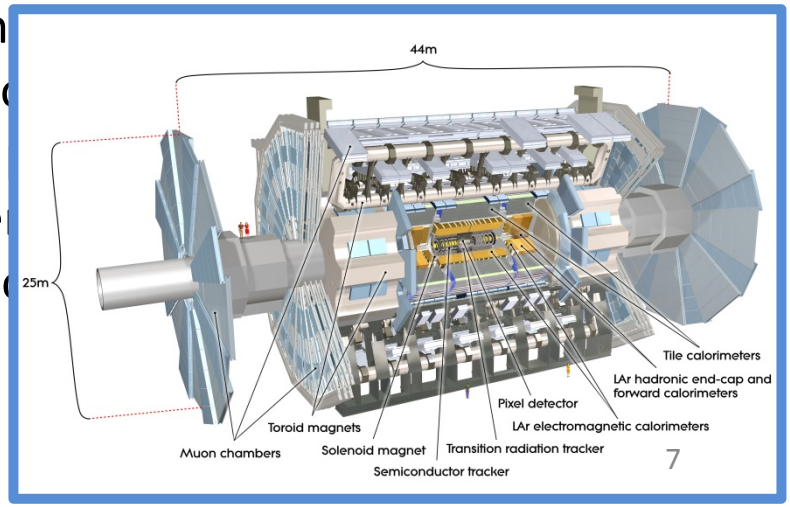


Event Processi are

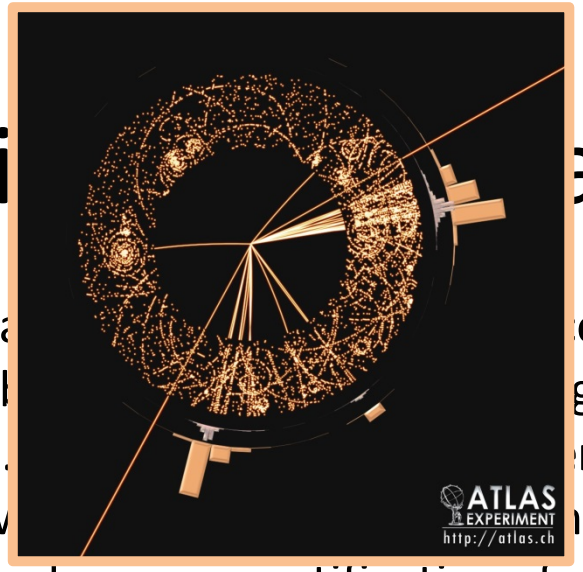


"In an event-ba omponents communicate b g **event** notifications [… **rvice** […] mediates betw n event-based system (EBS) and conveys notifications from **producers** [...] to **consumers** […]

he notification service decouples the components at producers unaware of any consumers and umers rely only on n where or by wh d style carries the nomous, heteroge systems that are easy t



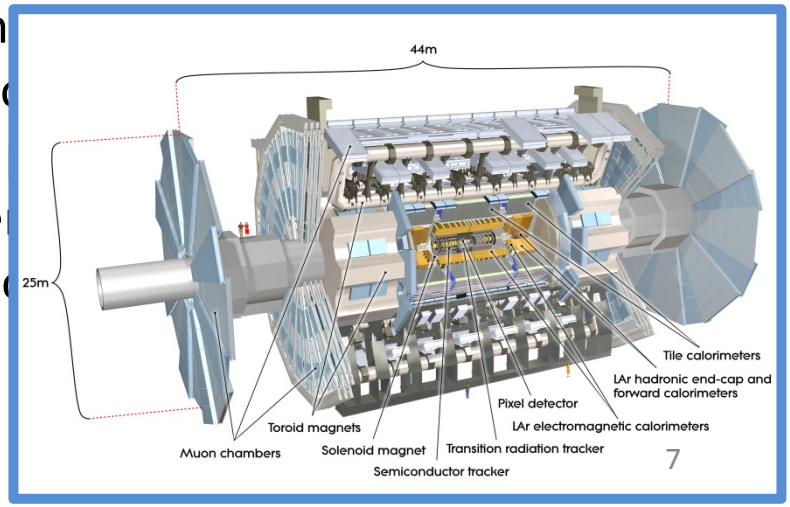
Event Processi are



"In an event-ba omponents communicate b g **event** notifications [… **rvice** […] mediates betw n event-based system (EBS) and conveys notifications from **producers** [...] to **consumers** […]

Still missing today?

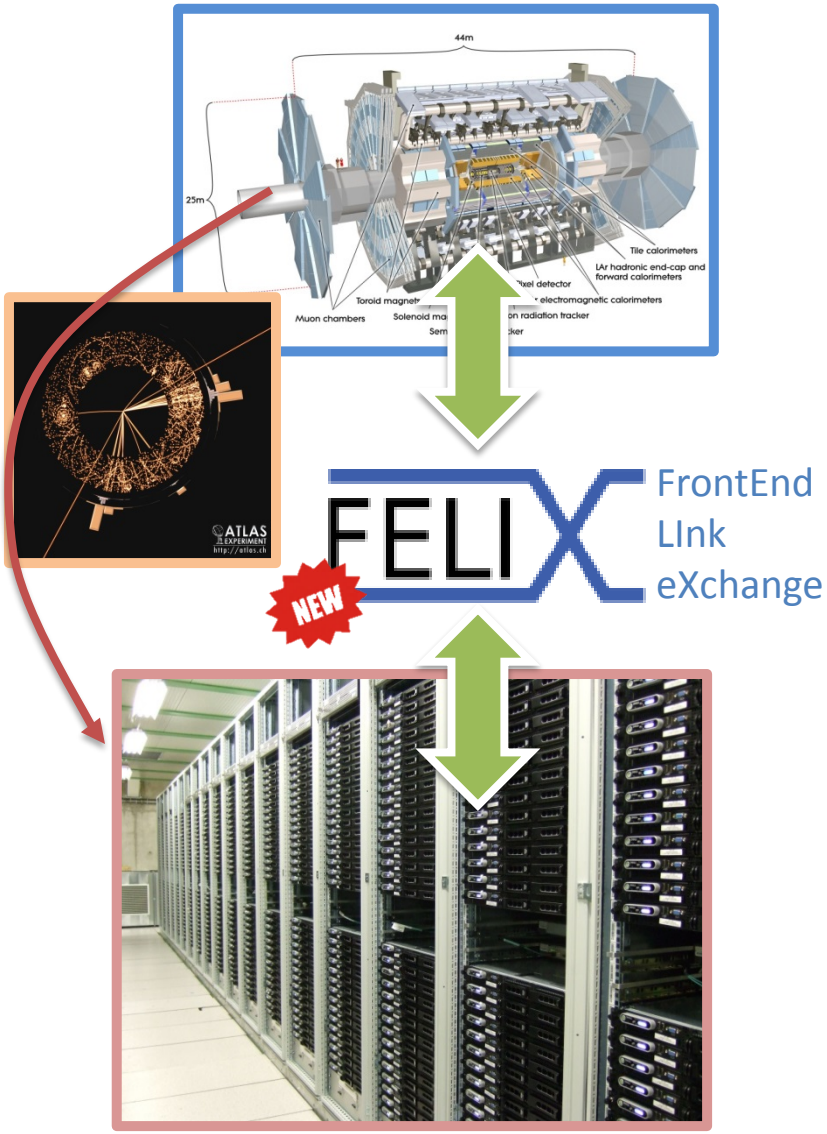
he notification service decouples the components at producers unaware of any consumers and umers rely only on n where or by wh d style carries the homous, heteroge systems that are easy t



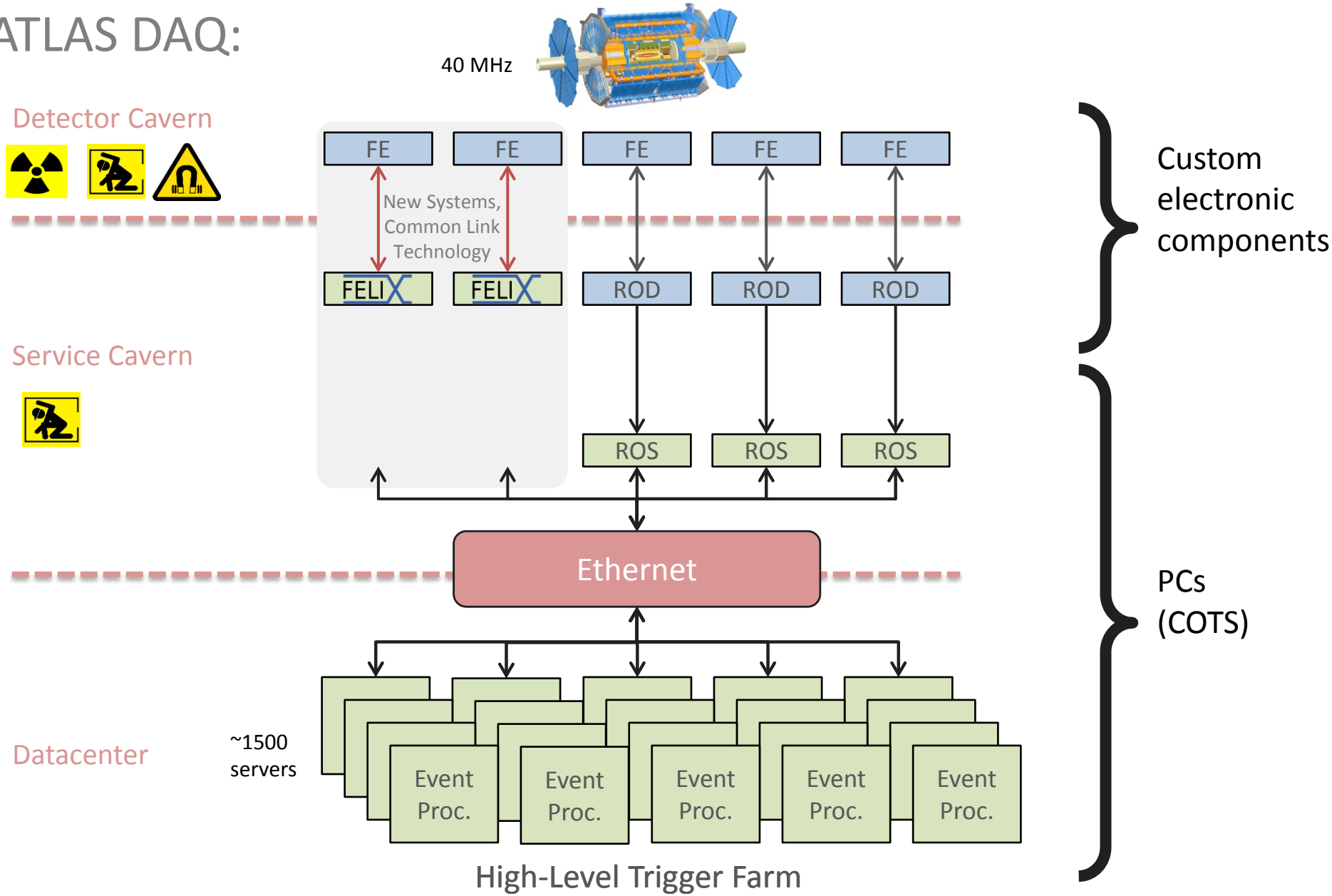
An Event Distribution System for ATLAS

Requirements:

- Simple operation and maintenance (PC technology over custom designed electronics)
- Scalability (Switched networks over point-to-point links)
- Interface to radiation-hard detector links
- Heterogeneous workloads



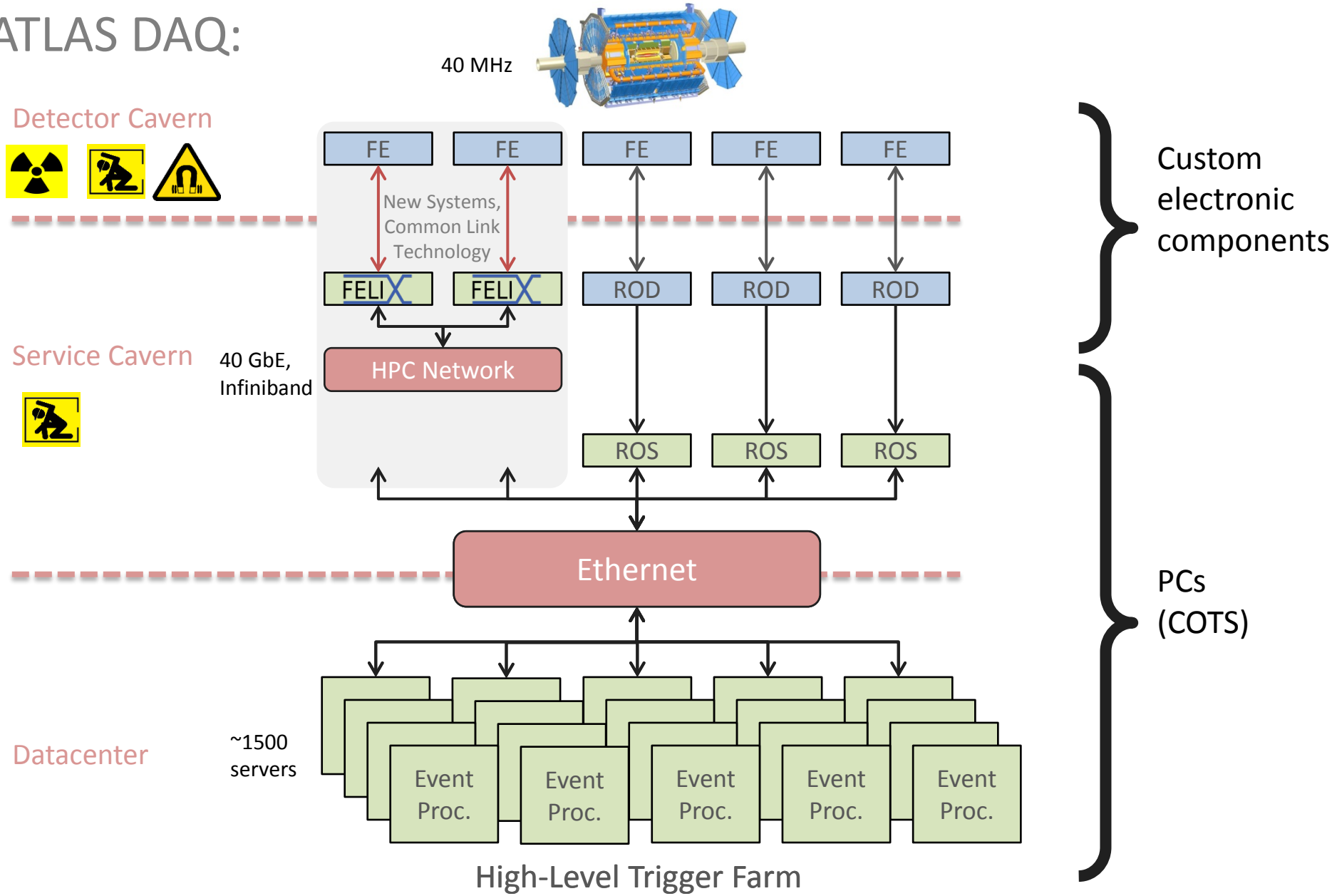
ATLAS DAQ:



2018

Run 2 2015 2017 Run 3 2019 2021 Run 4 2023 2025

ATLAS DAQ:

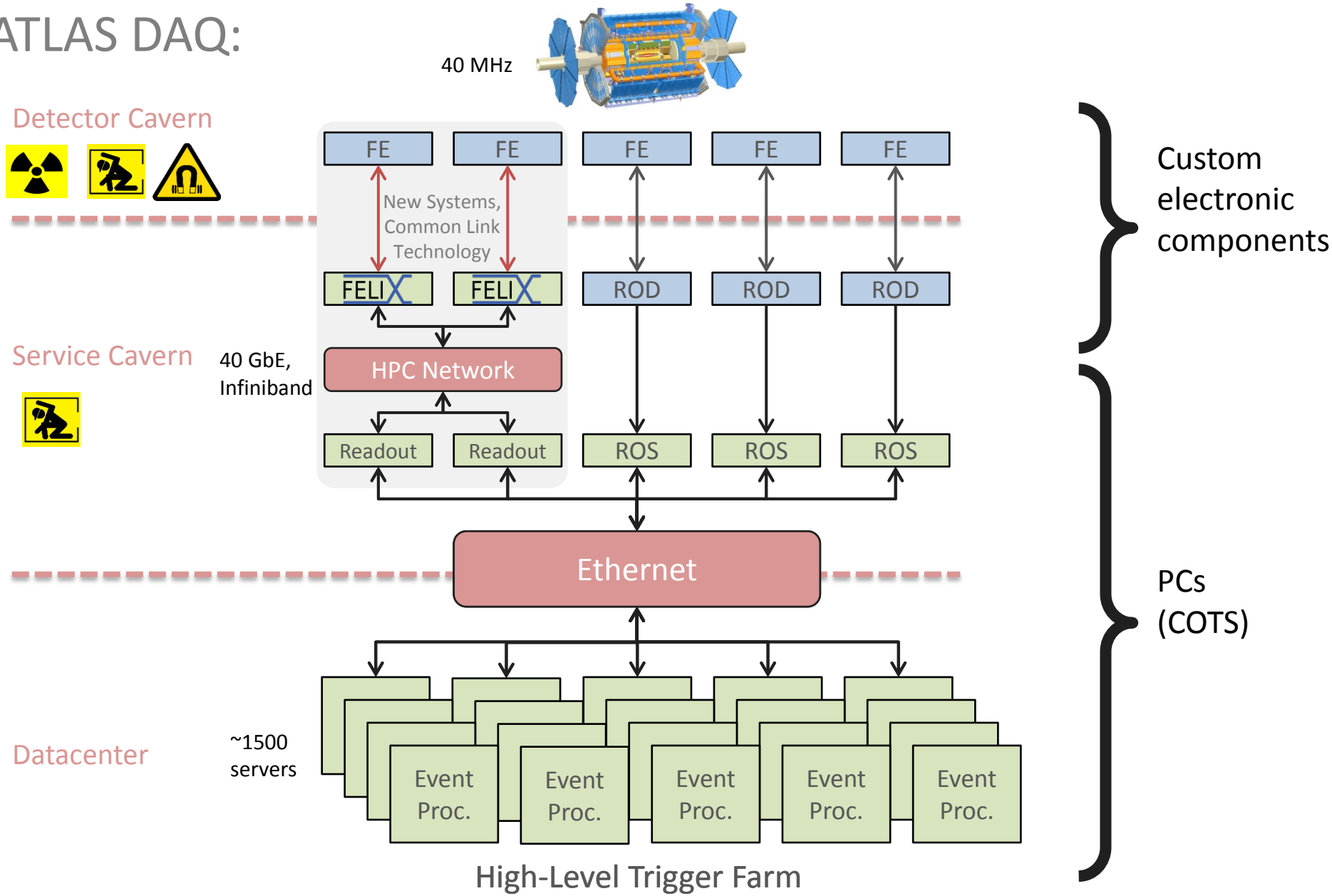


2018

Run 2 — Run 3 — Run 4

2015 2017 2019 2021 2023 2025

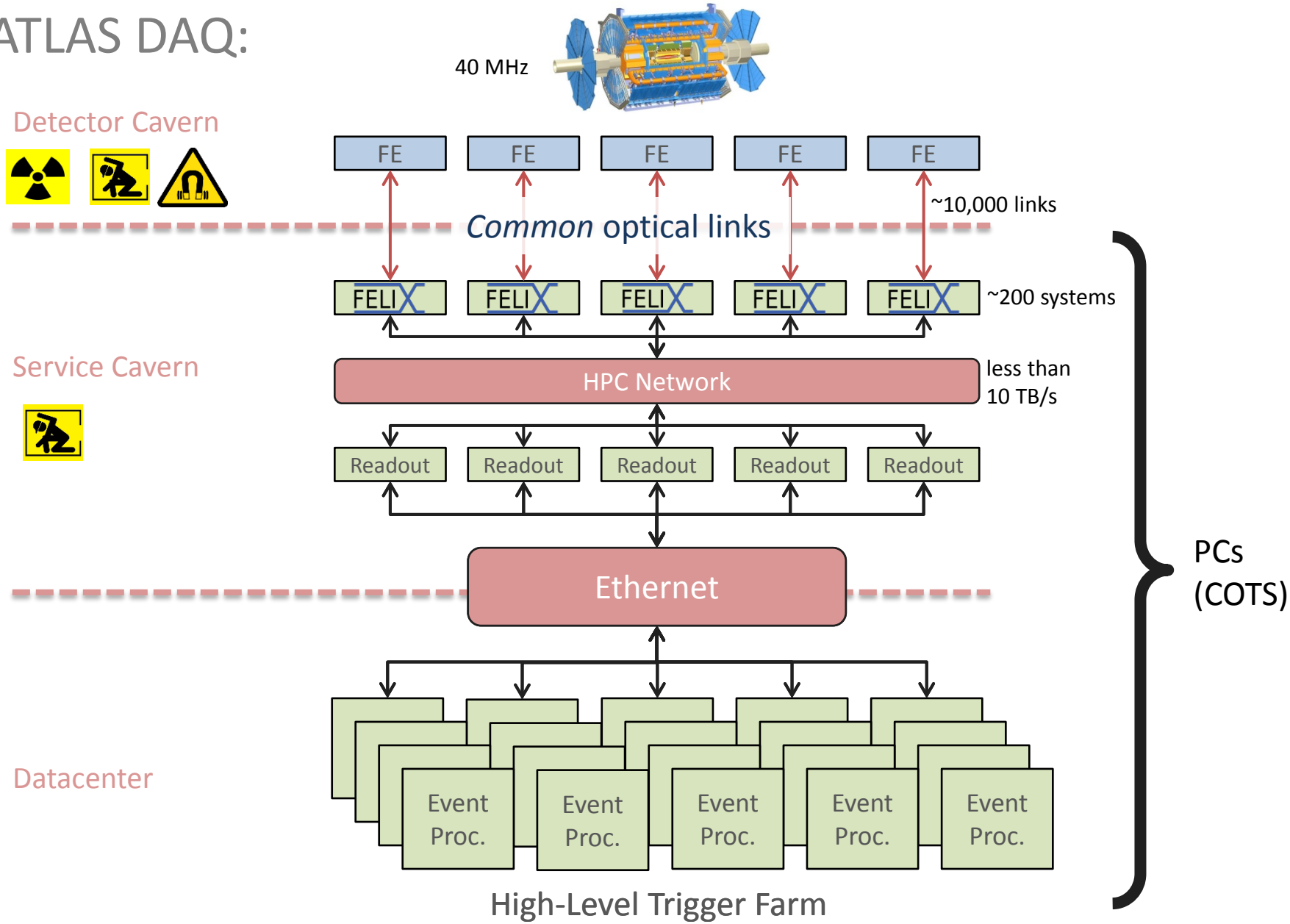
ATLAS DAQ:



2018

Run 2 2015 2017 Run 3 2019 2021 Run 4 2023 2025

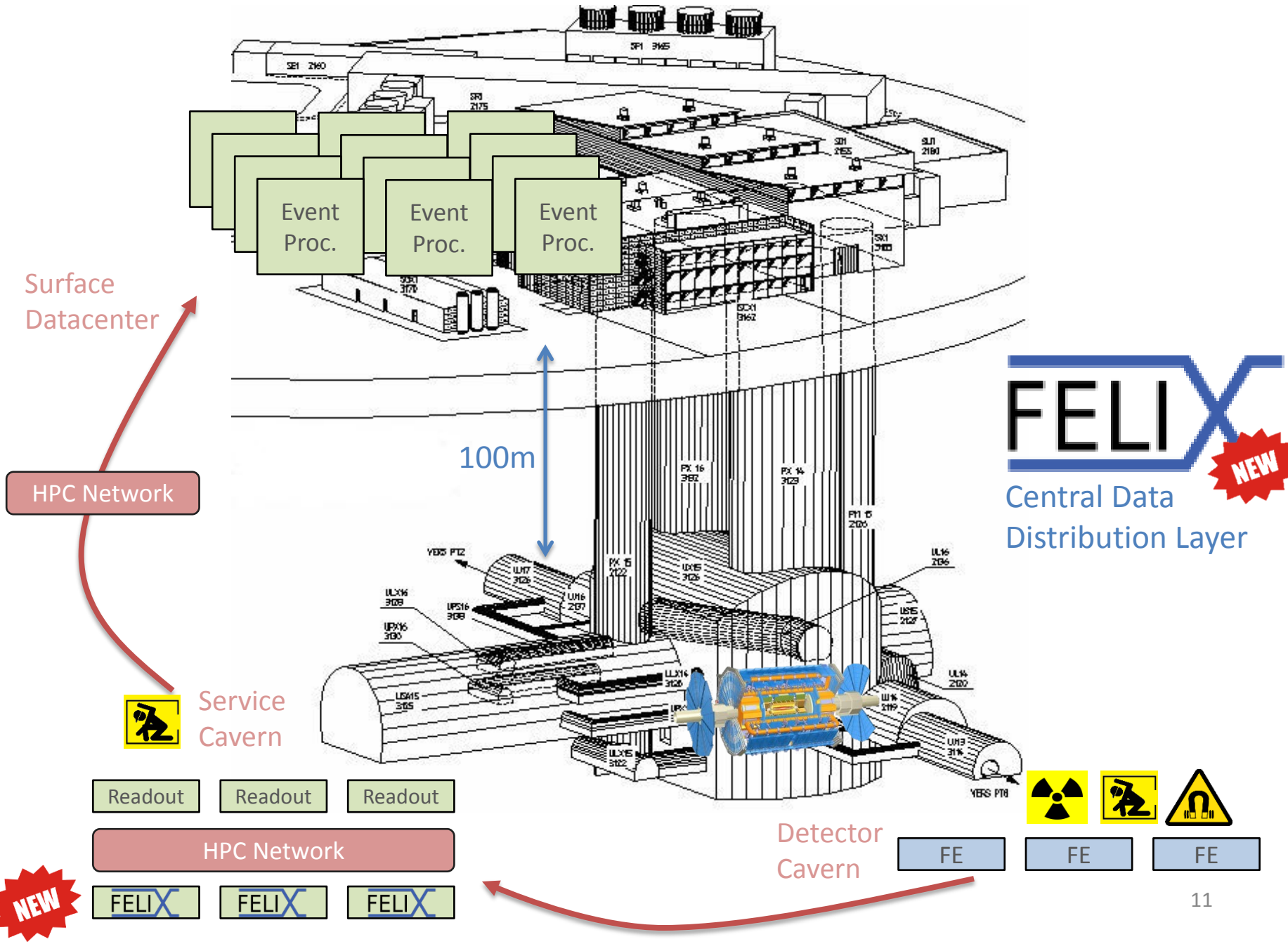
ATLAS DAQ:



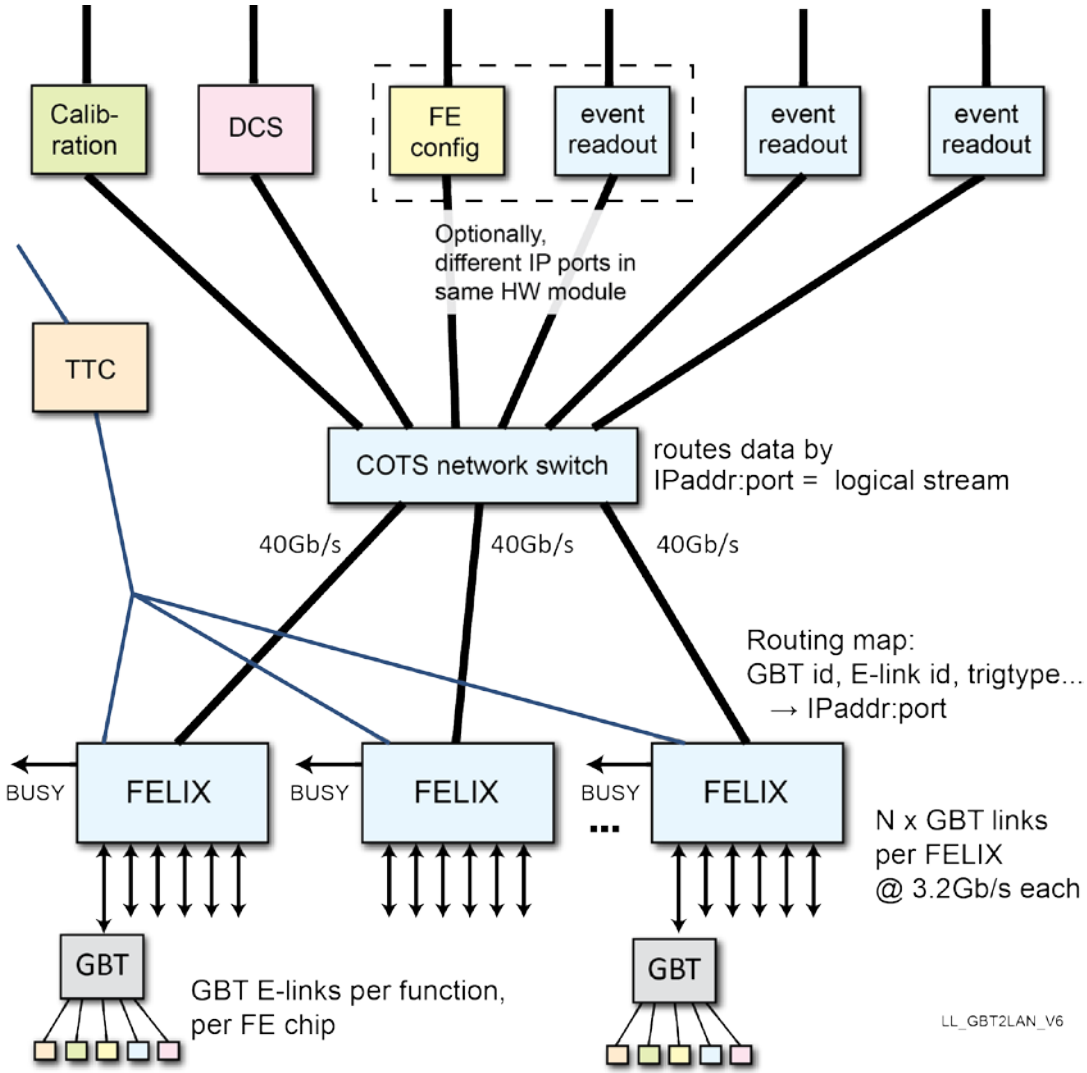
2023

Run 2 Run 3 Run 4

2015 2017 2019 2021 2023 2025


Event Proc.
Event Proc.
Event Proc.
Surface Datacenter
HPC Network
100m
FELIX
NEW
Central Data Distribution Layer
Service Cavern
Readout
Readout
Readout
HPC Network
NEW
FELIX
FELIX
FELIX
Detector Cavern
FE
FE
FE

FELIX: Detector-to-Network Data Routing



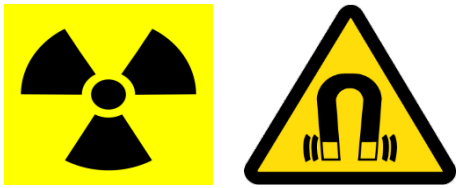
- Scalable architecture
- Routing of multiple traffic types: physics events, detector control, configuration, calibration, monitoring
- Industry standard links: data processors/handlers can be SW in PCs - Less custom electronics, more COTS components
- Reconfigurable data path, multi-cast, cloning, QoS
- Automatic failover and load balancing

CERN GBT Link Technology

(GigaBit Transceiver)

Point-to-point link technology developed at CERN, progressively replaces detector-specific links in ATLAS

Designed for common High-Energy Physics environments (high radiation, magnetic fields, …)



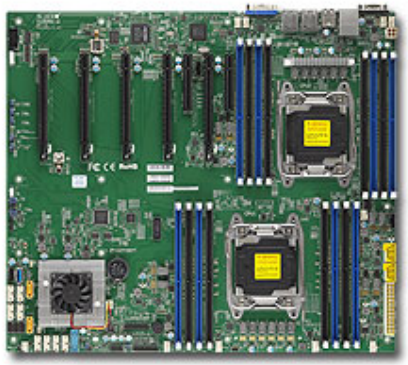
Typical raw bandwidth 4.8 Gbps or 9.6 Gbps

Supports variable-width virtual links (“elinks”) for mixed traffic types

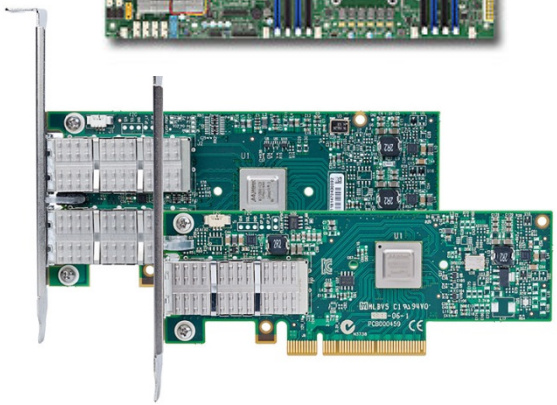
(Optional) Forward Error Correction

Development Platform

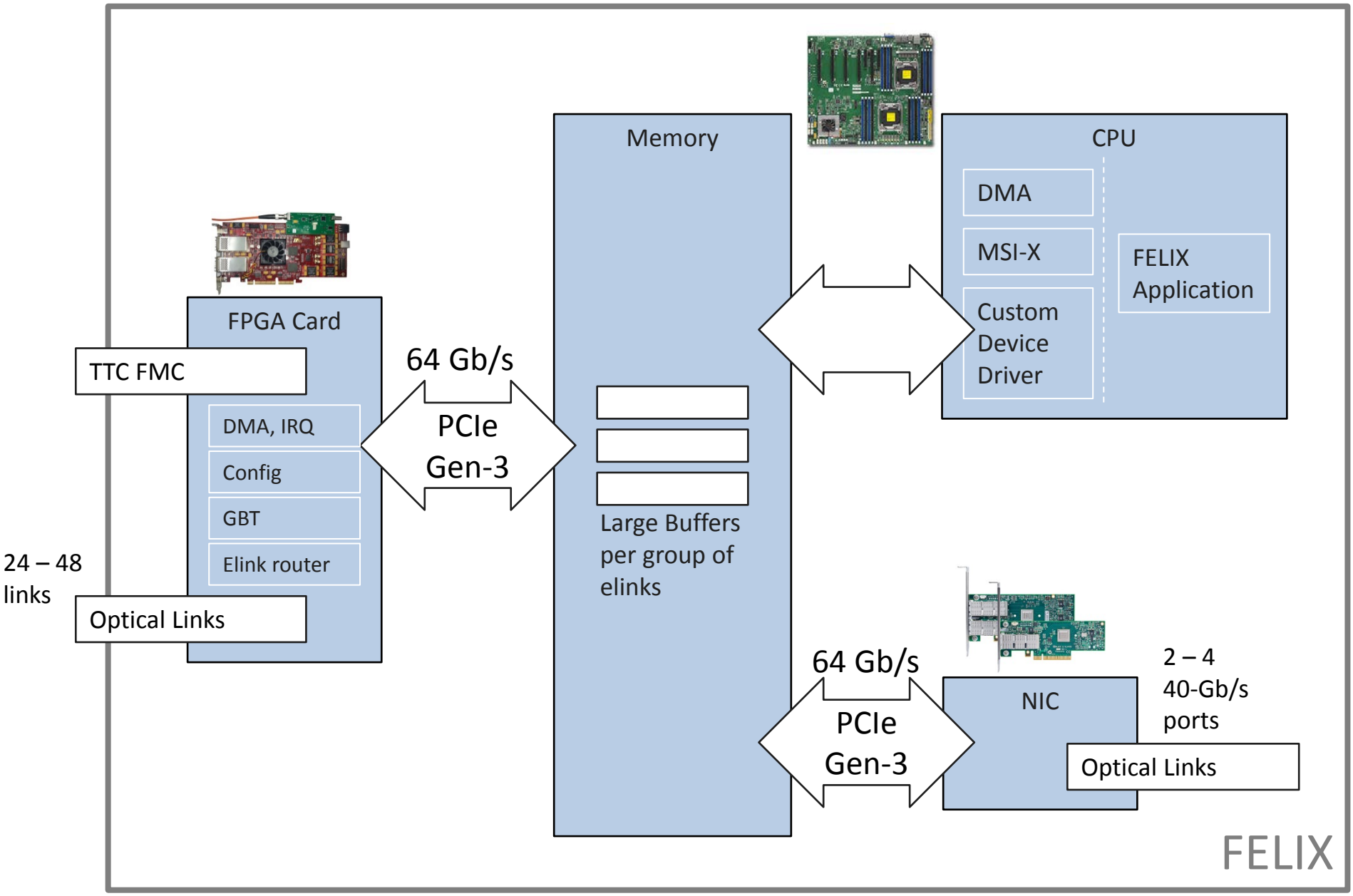
HiTech Global PCIe development
Xilinx Virtex-7
PCIe Gen-2/3 x8
24 bi-directional links
http://hitechglobal.com/Boards/PCIE-CXP.htm
With custom TTCfx FMC



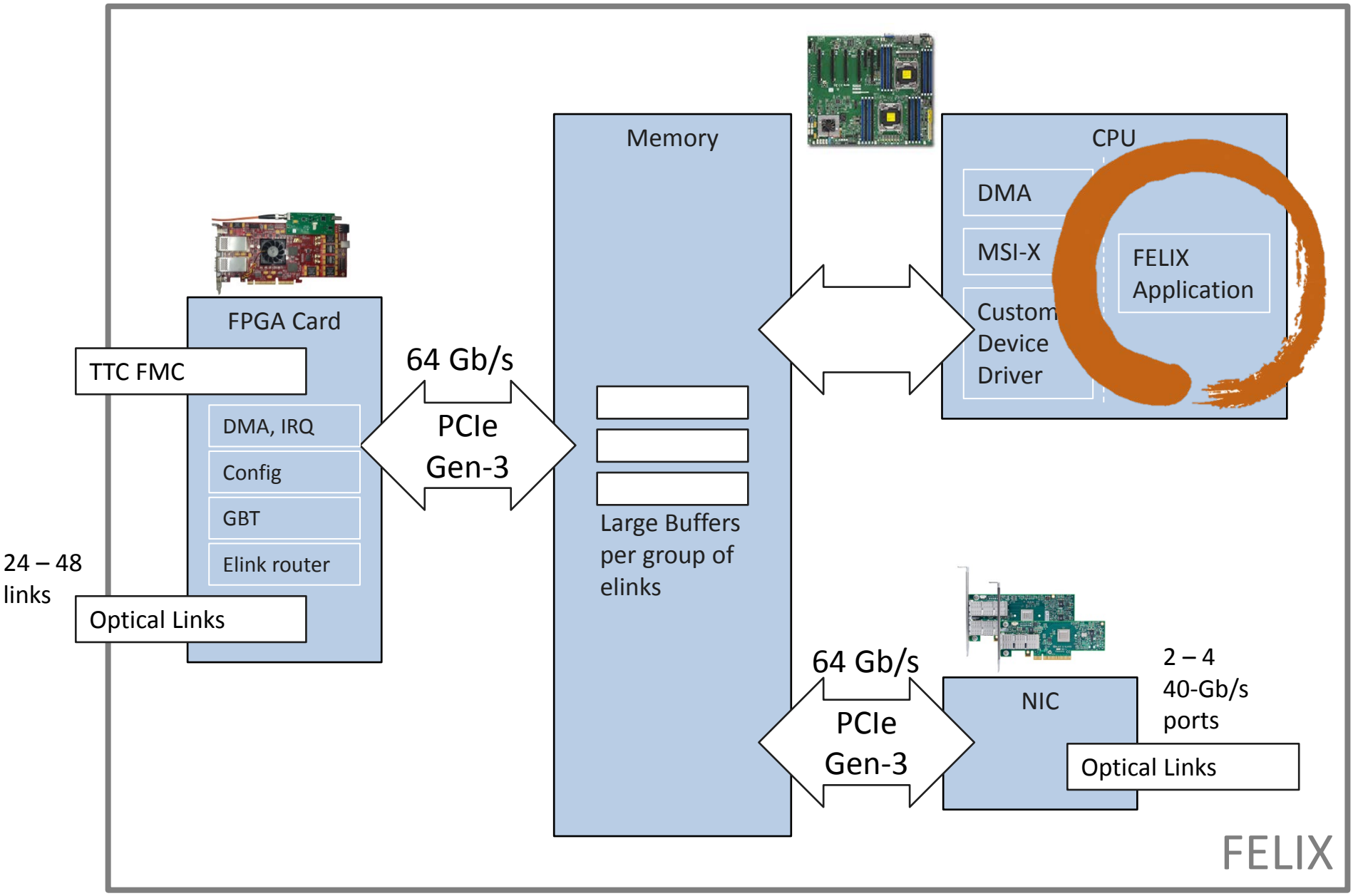
SuperMicro X10DRG-Q
2x Haswell CPU, up to 10 cores
6x PCIe Gen-3 slots
64 GB DDR4 Memory
http://supermicro.com/products/motherboard/Xeon/C600/X10DRG-Q.cfm



Mellanox ConnectX-3 VPI
FDR/QDR Infiniband
2x10/40 GbE
http://www.mellanox.com/page/products_dyn?product_family=119&mtag=connectx_3_vpi

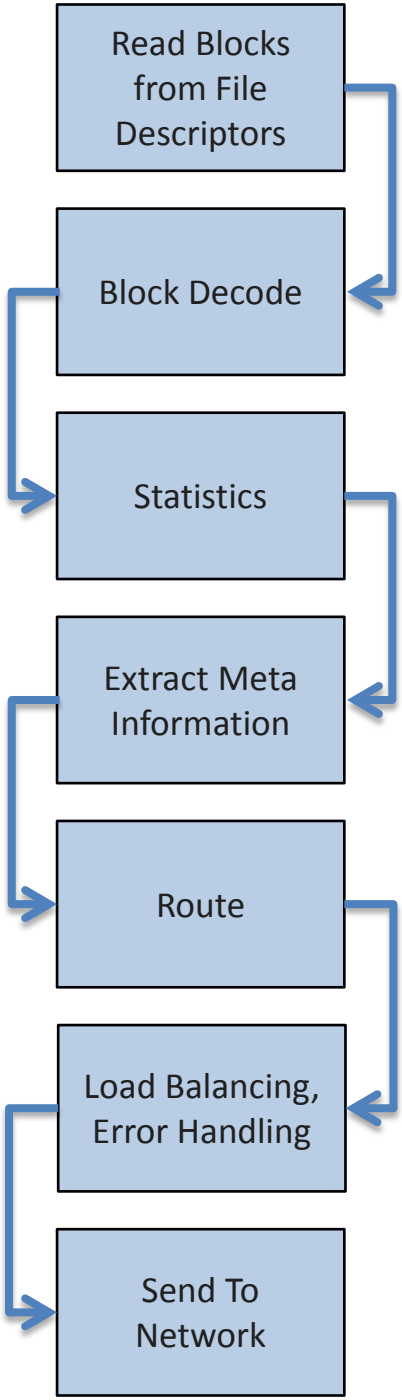


FELIX Architectural Overview

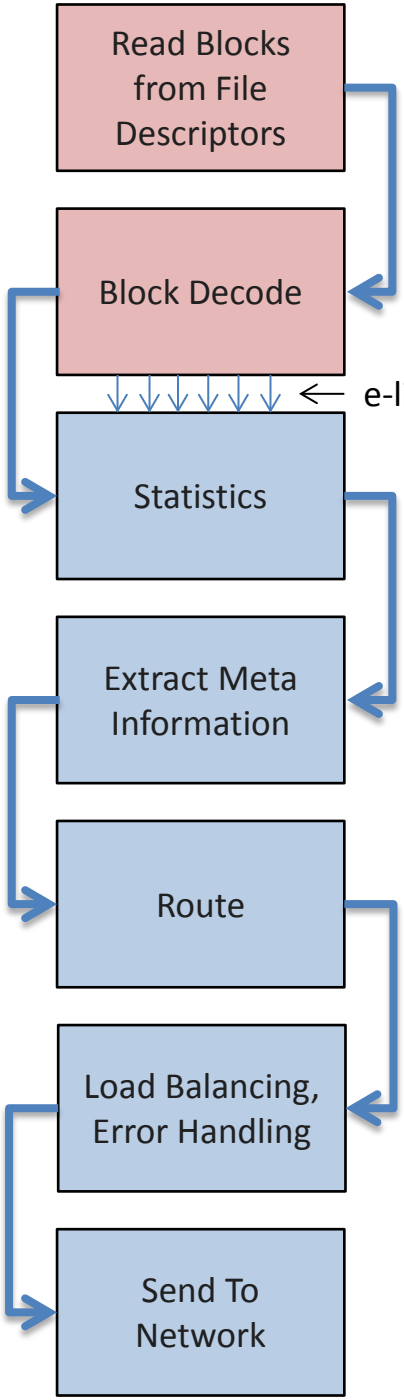


FELIX Architectural Overview

CPU Data Processing Pipeline



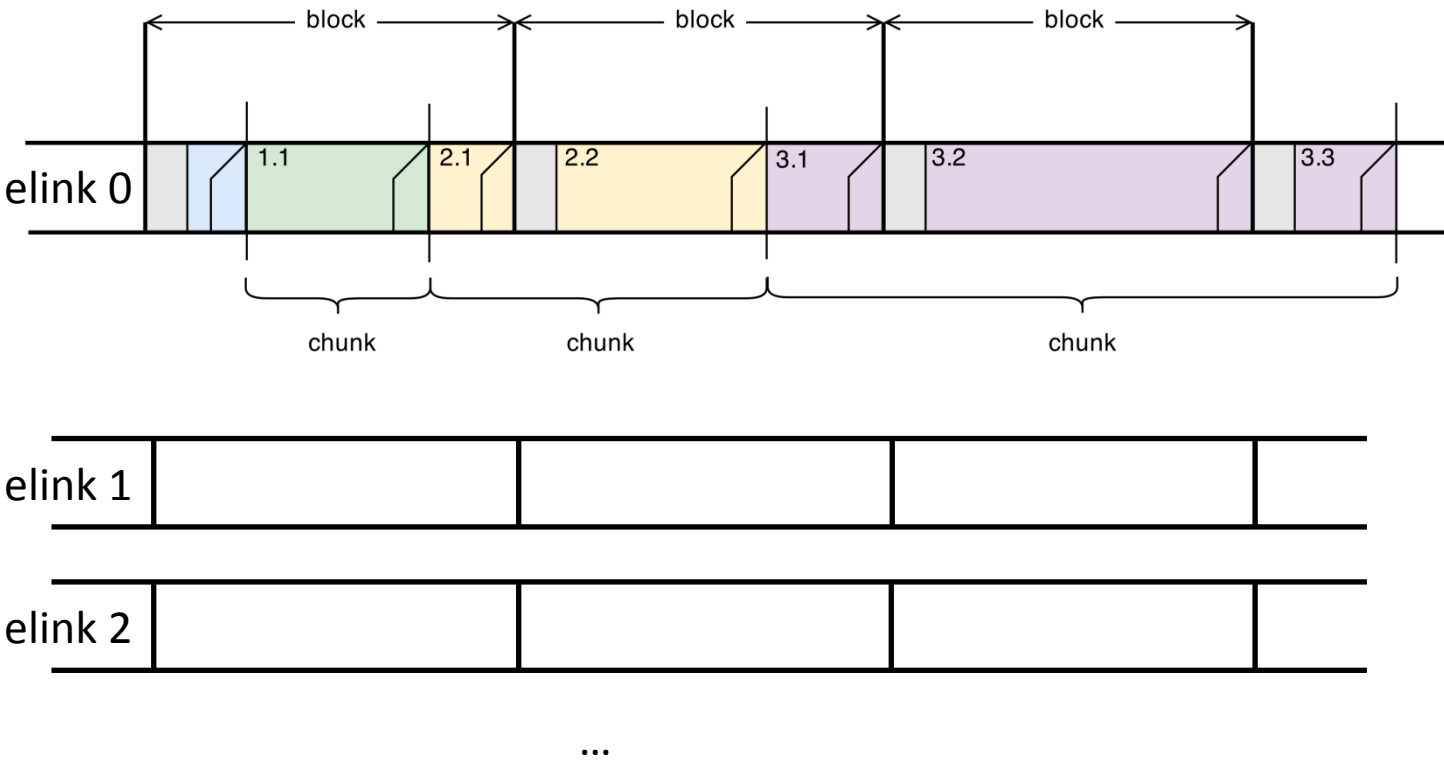
CPU Data Processing Pipeline



Program DMA transfers and read fixed blocks of data that have been encoded for the transfer over PCIe

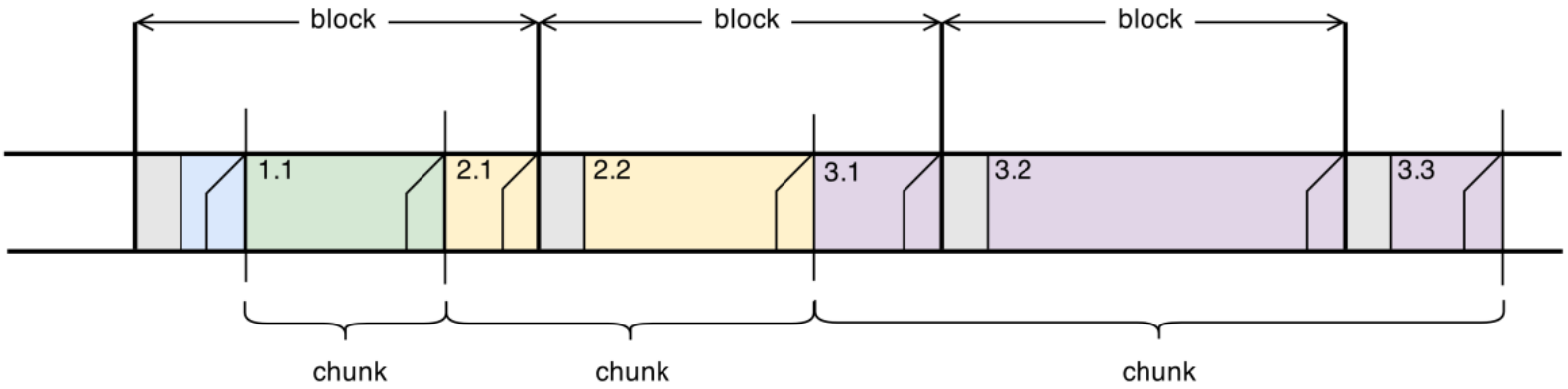
Decode into variable sized chunks for transmission over network

e-link streams share the same pipeline



Fixed Block Decoding

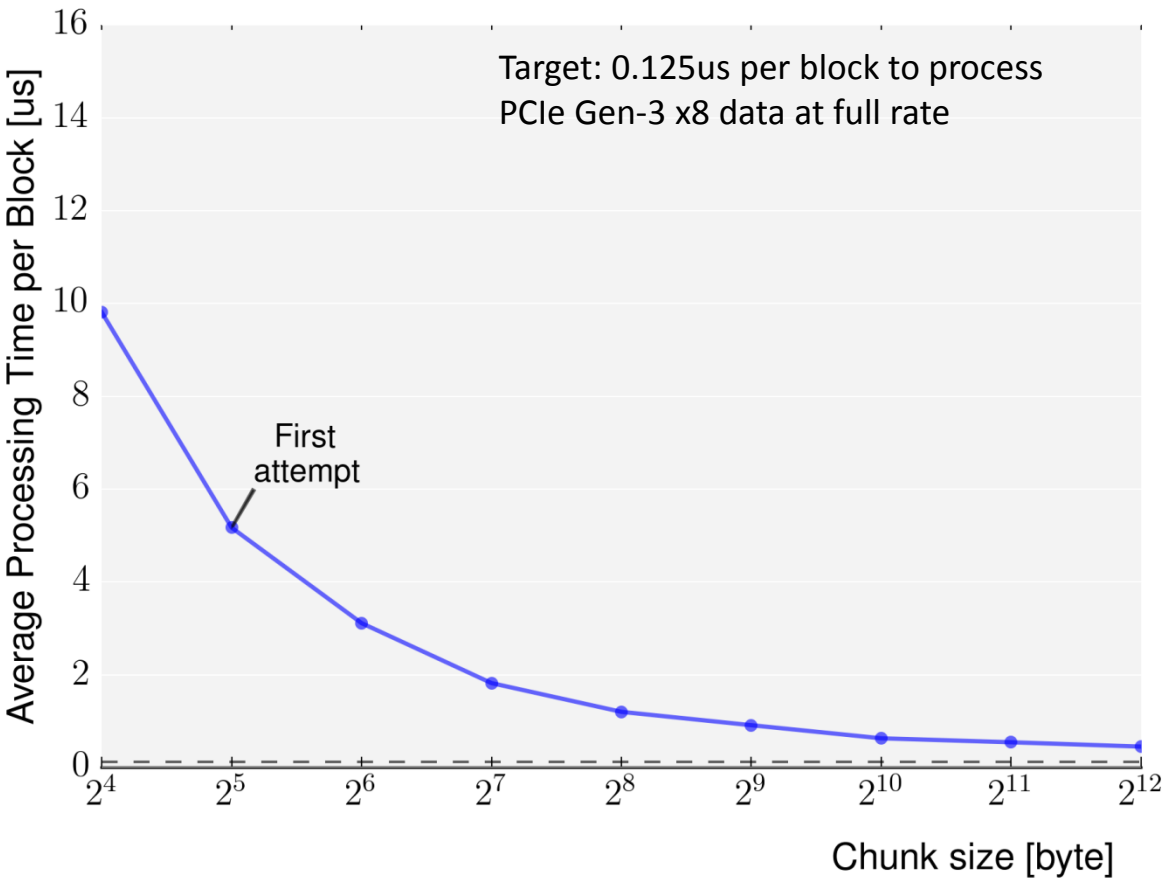
Stream of Blocks transmitted over PCIe:



Stream of reconstructed chunks with meta information for further processing:

Packets are analyzed, routed and transmitted to network destinations

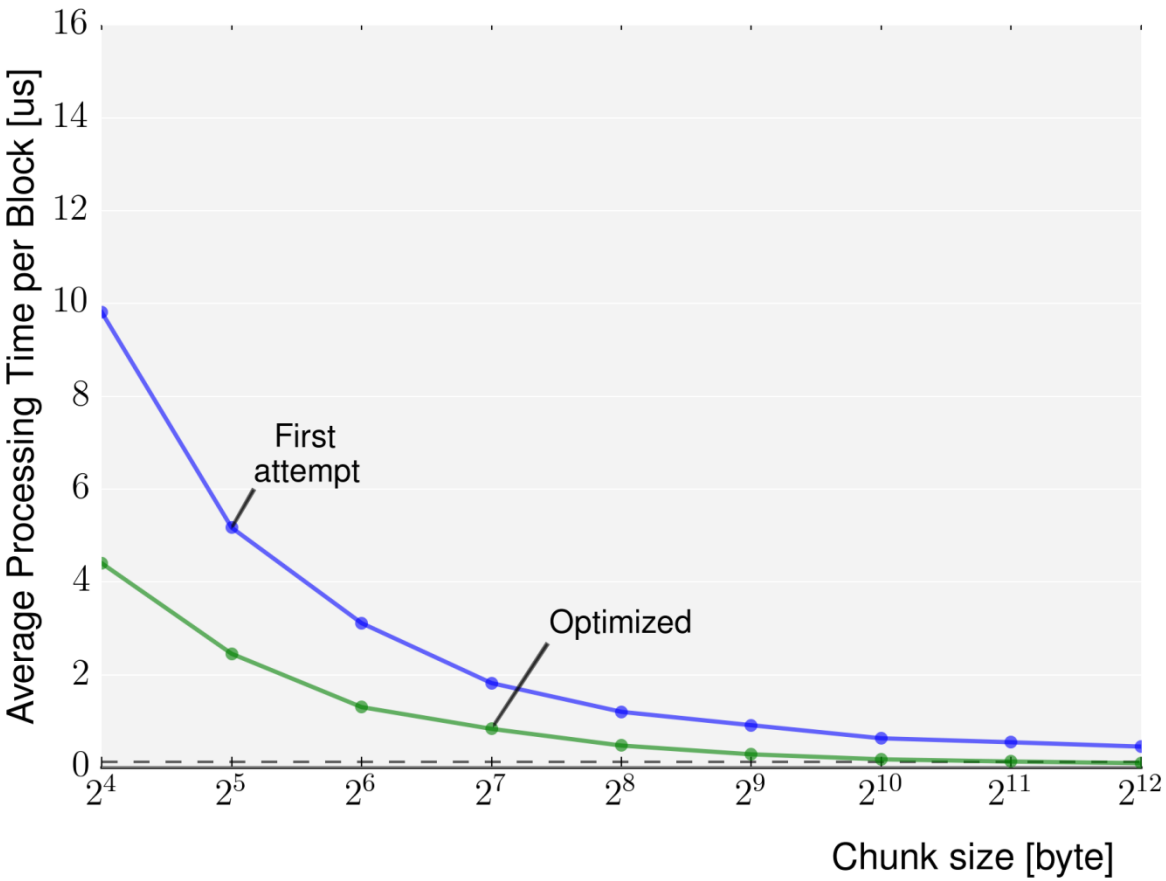
Optimizations (single-threaded)



Isolated benchmarks of the packet processing benchmarks with **in-memory** input data

Measurements done on
Intel Core i7-3770 CPU
4 cores @ 3.40 GHz

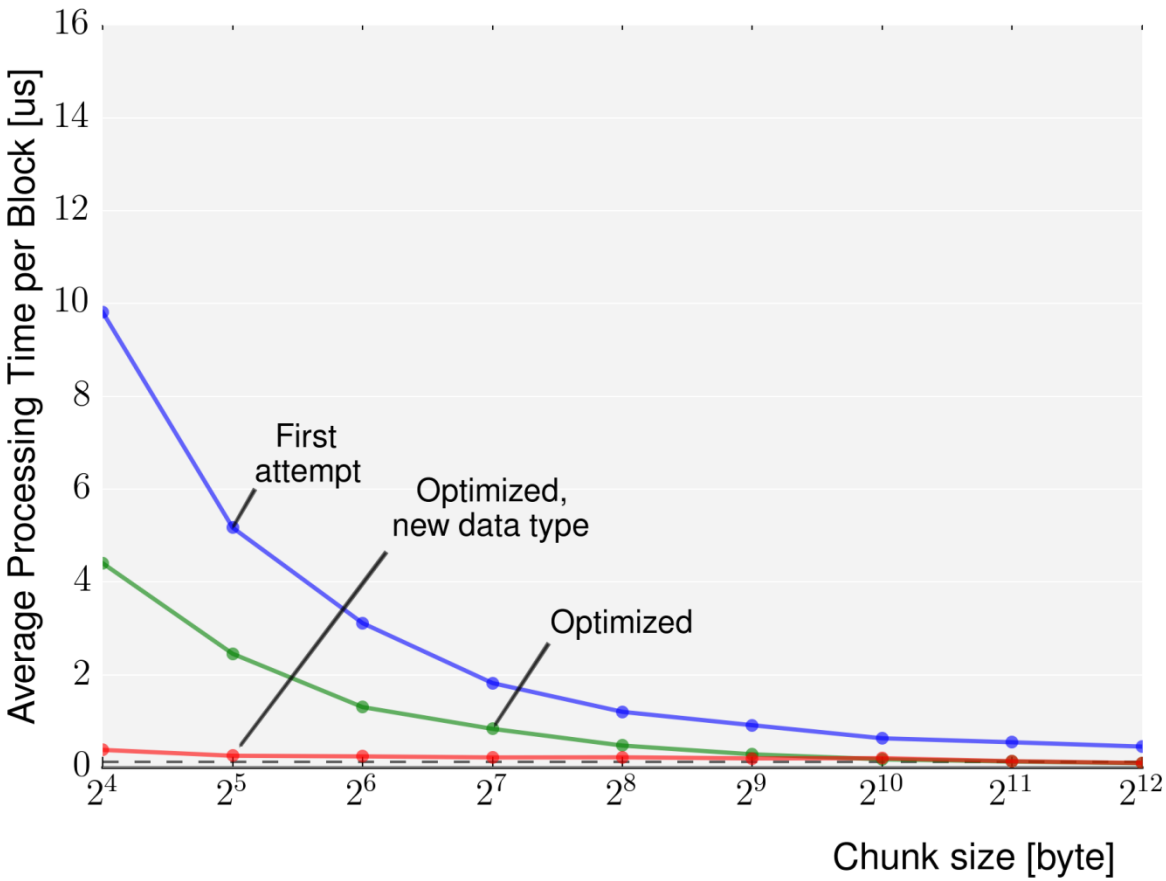
Optimizations (single-threaded)



Measurements done on
Intel Core i7-3770 CPU
4 cores @ 3.40 GHz

- `emplace` instead of `push_back` (Construction of objects in-place in containers)
- Moving support data structures to class-scope to avoid initializations on each algortithm call
- Pre-reserve memory for `std::vector` on initialization
- NUMA-aware memory allocations (using libnuma)
- Data prefetching using GCC's `__builtin_prefetch`
- Compiler option tuning

Optimizations (single-threaded)

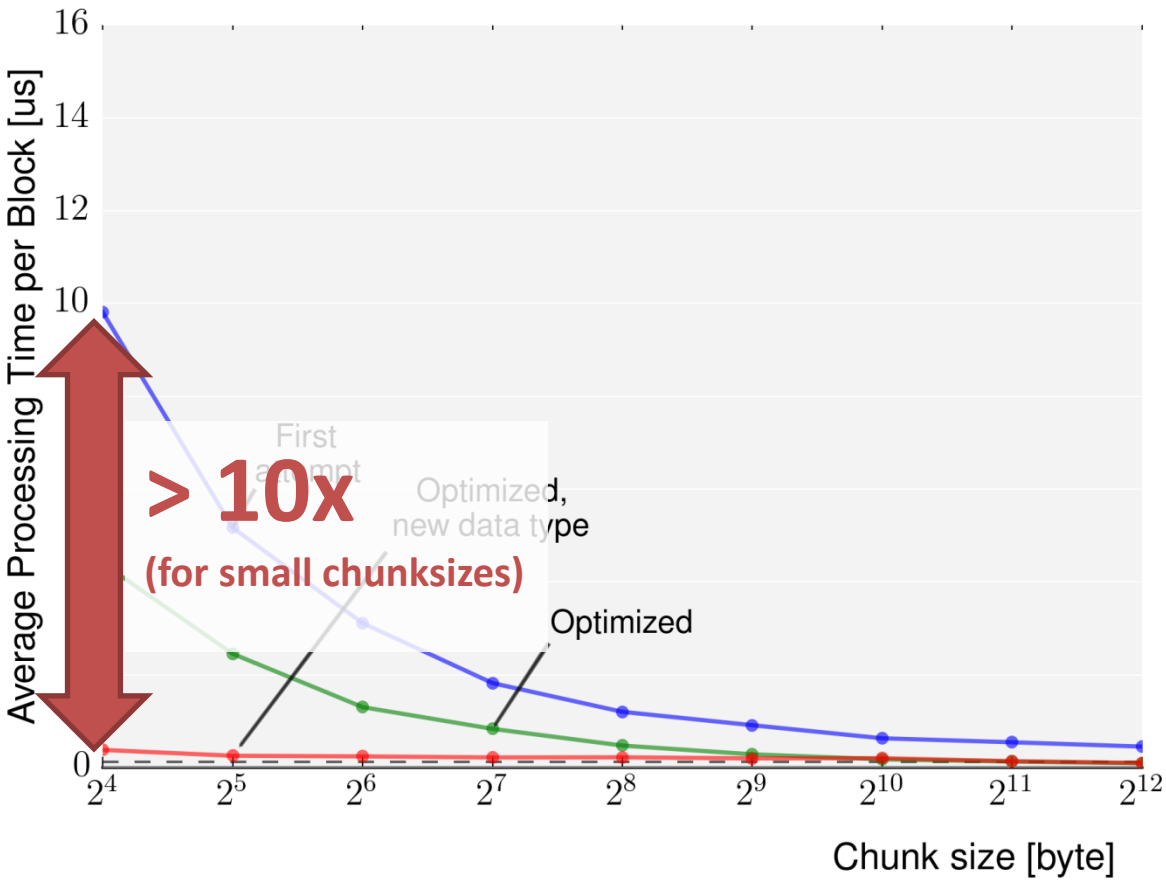


- New data structure is used only for chunks that *fit into a single block*
- Avoids keeping track of past blocks altogether
- Especially useful for small chunk sizes (much more likely that a chunk fits in a block)

- Downside: 2x more code

Measurements done on
Intel Core i7-3770 CPU
4 cores @ 3.40 GHz

Optimizations (single-threaded)

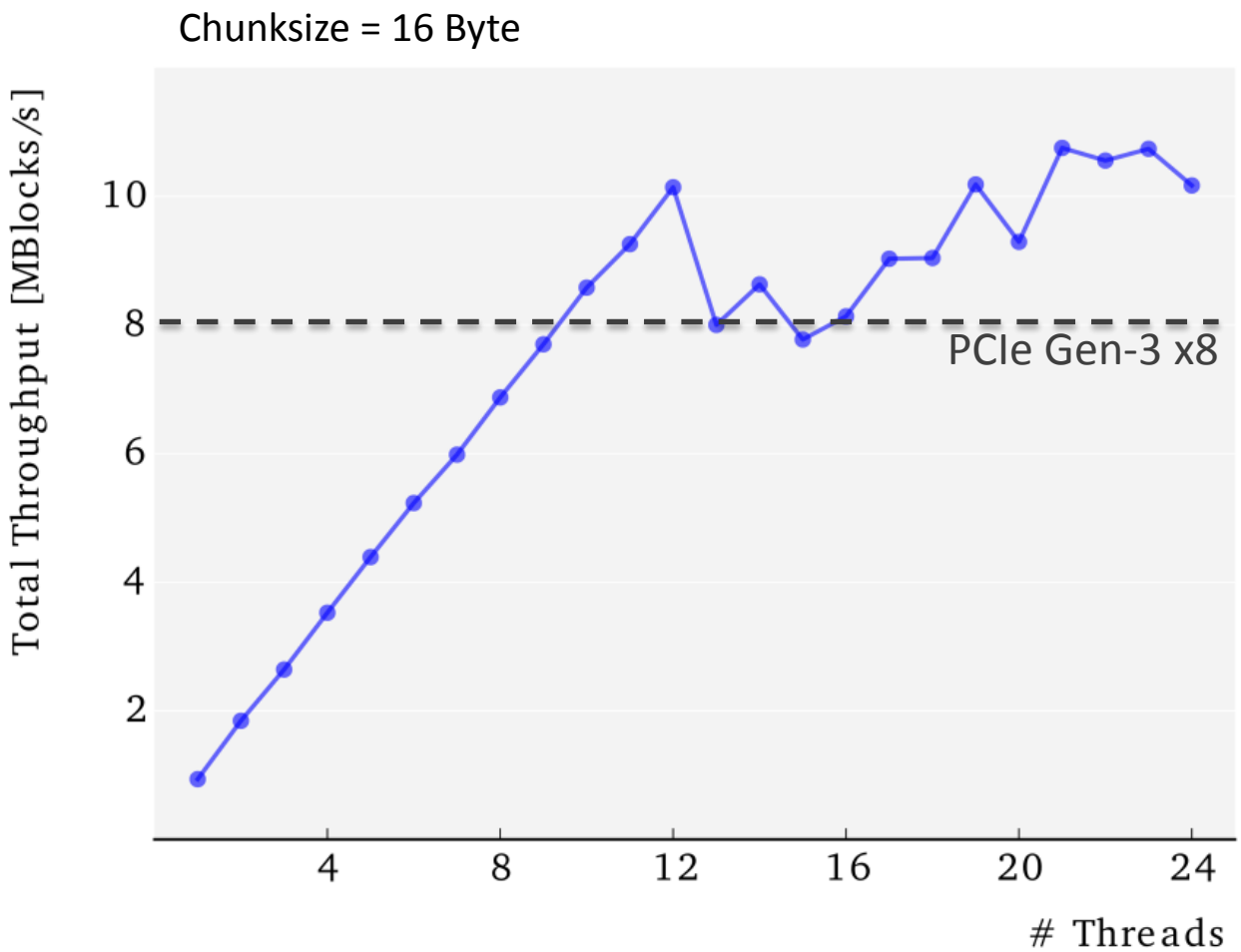


- New data structure is used only for chunks that *fit into a single block*
- Avoids keeping track of past blocks altogether
- Especially useful for small chunk sizes (much more likely that a chunk fits in a block)

- Downside: 2x more code

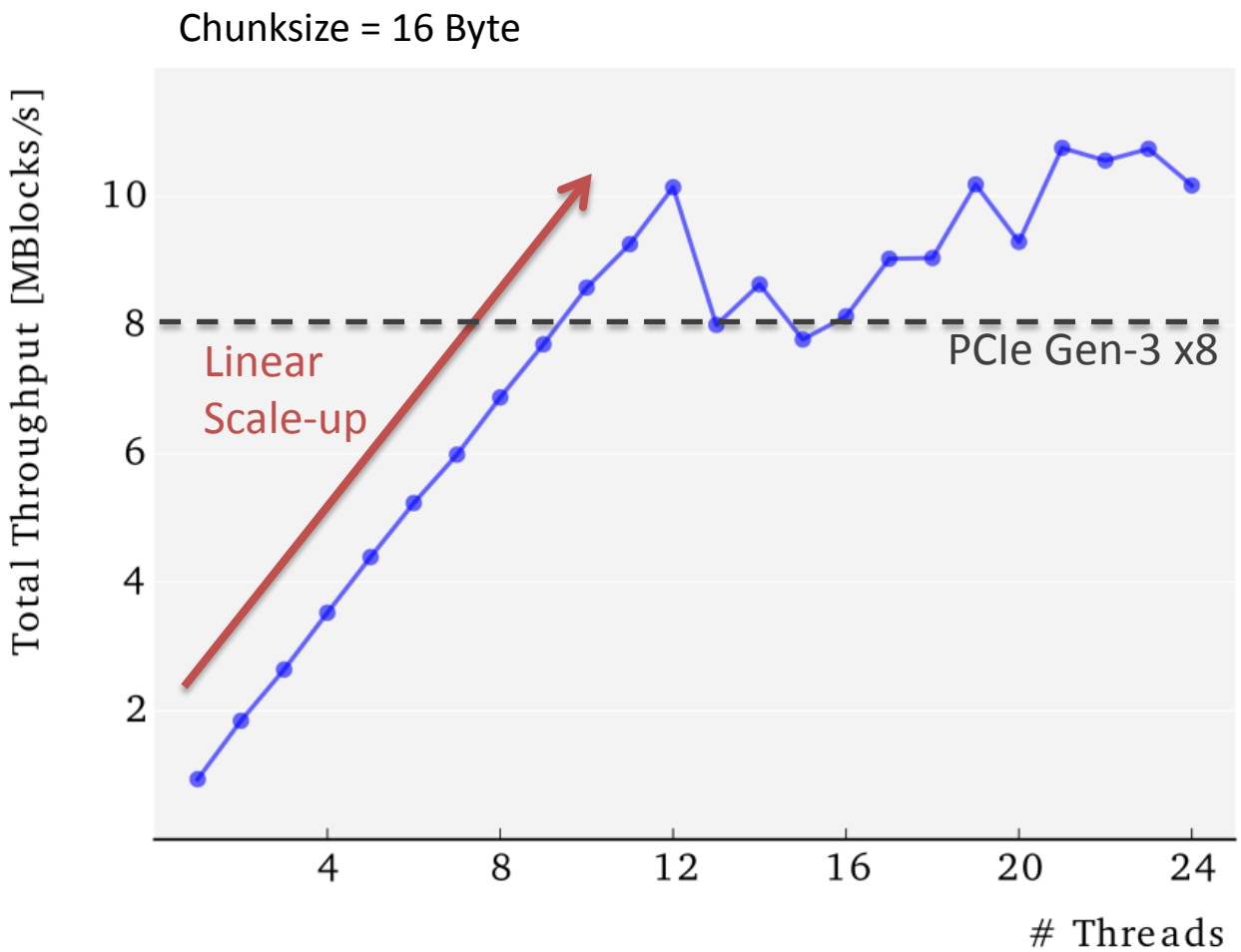
Measurements done on
Intel Core i7-3770 CPU
4 cores @ 3.40 GHz

Multi Threading



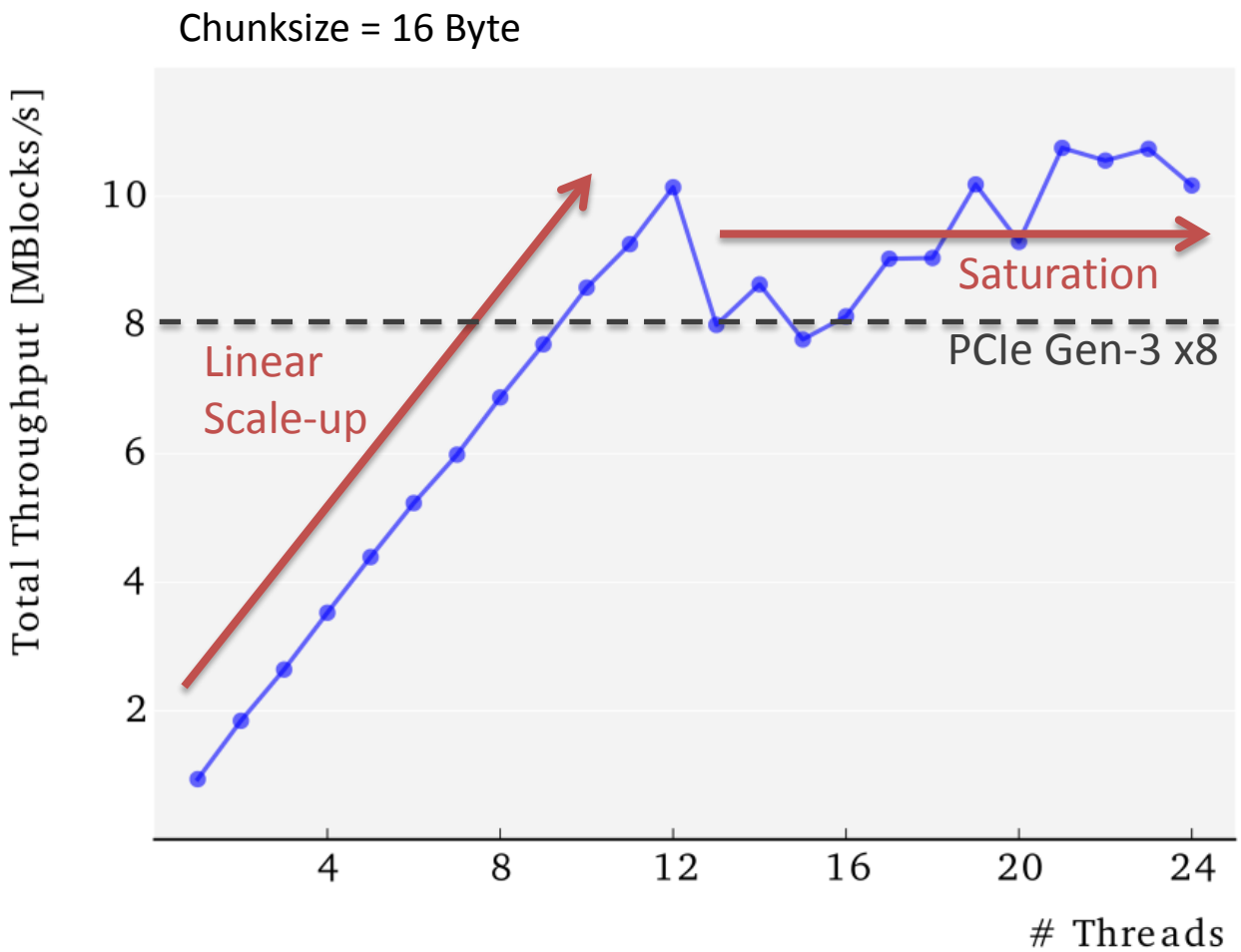
Measurements done on
Intel Xeon E5645
2x6 cores @ 2.40 GHz

Multi Threading



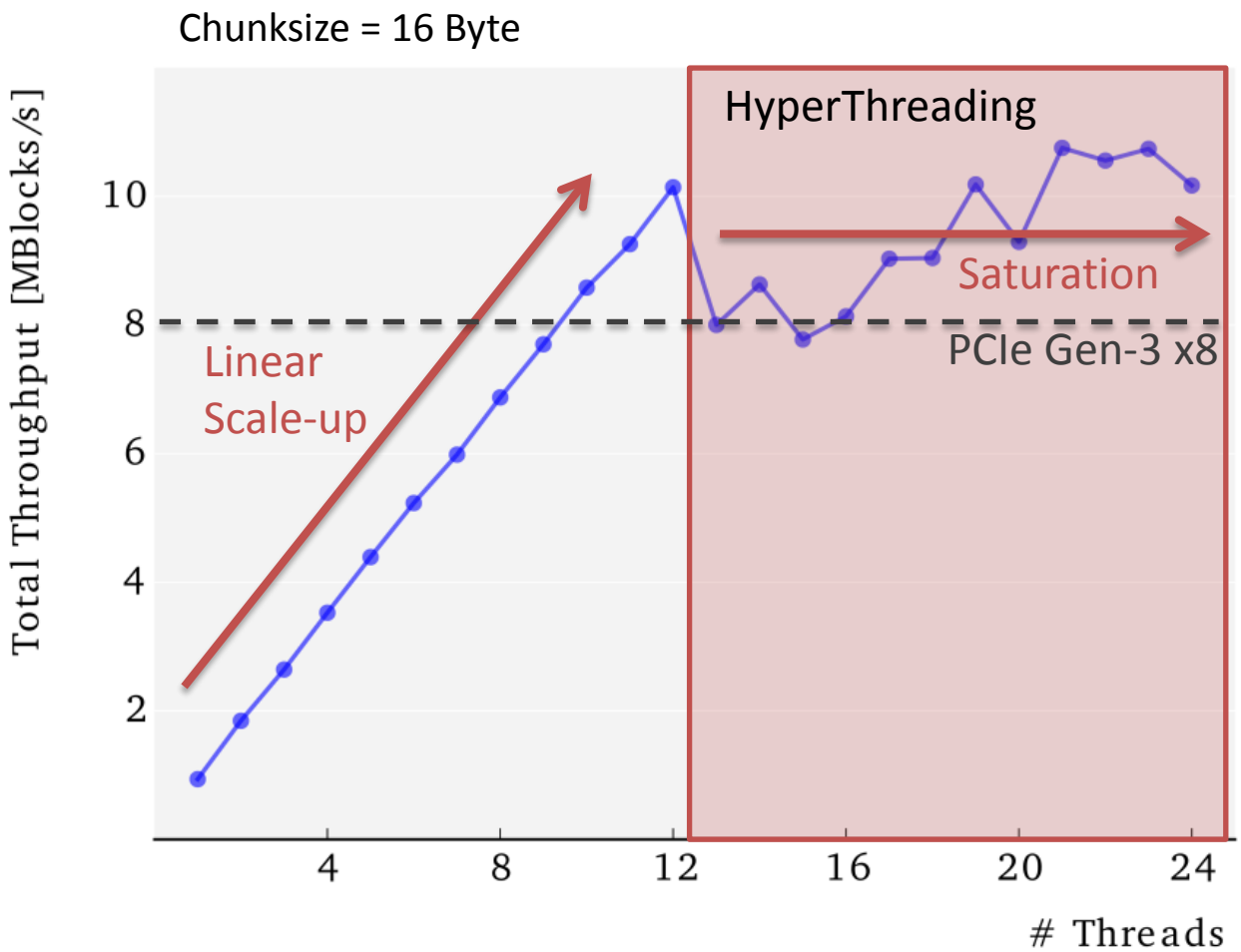
Measurements done on
Intel Xeon E5645
2x6 cores @ 2.40 GHz

Multi Threading



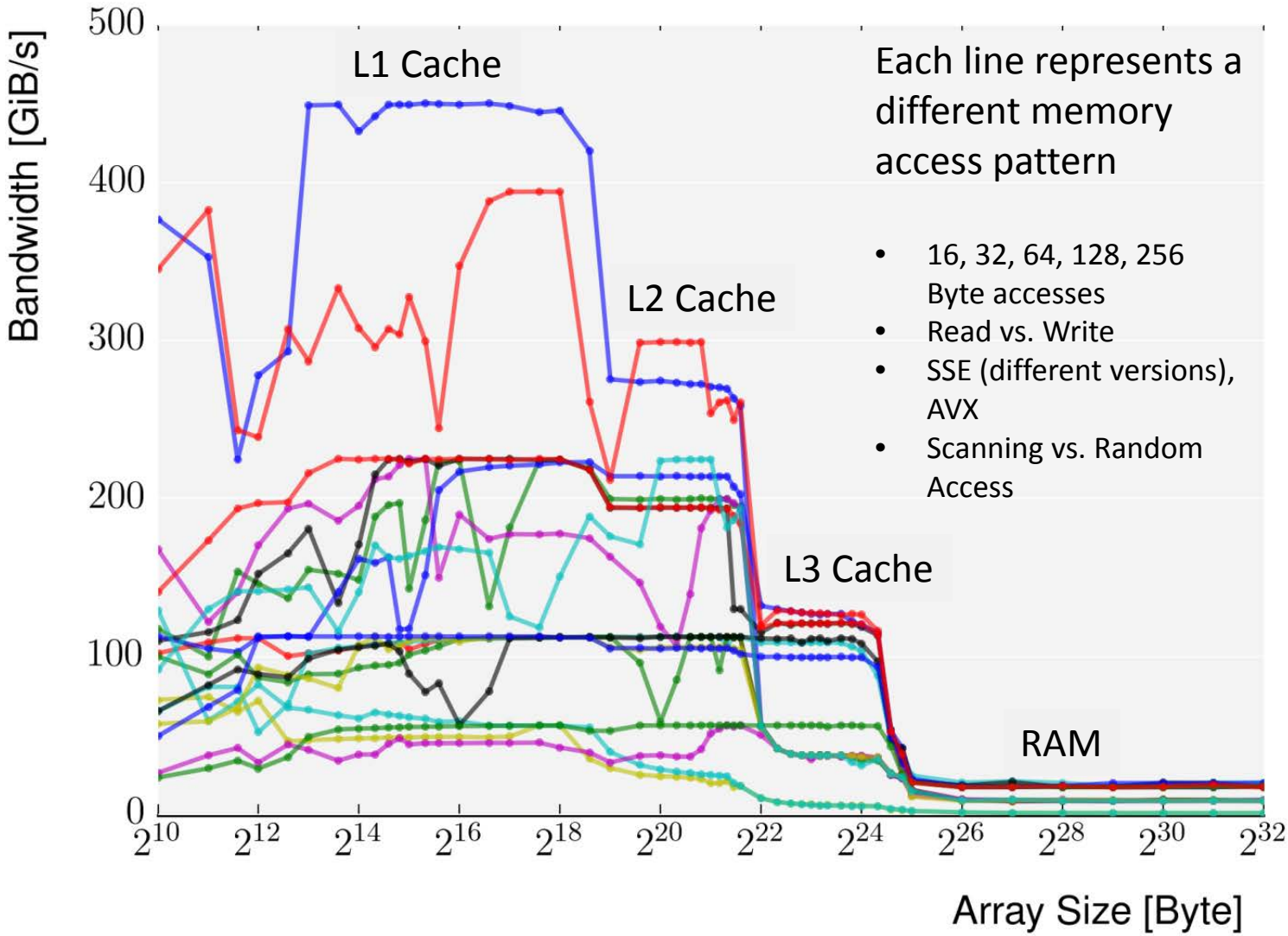
Measurements done on
Intel Xeon E5645
2x6 cores @ 2.40 GHz

Multi Threading

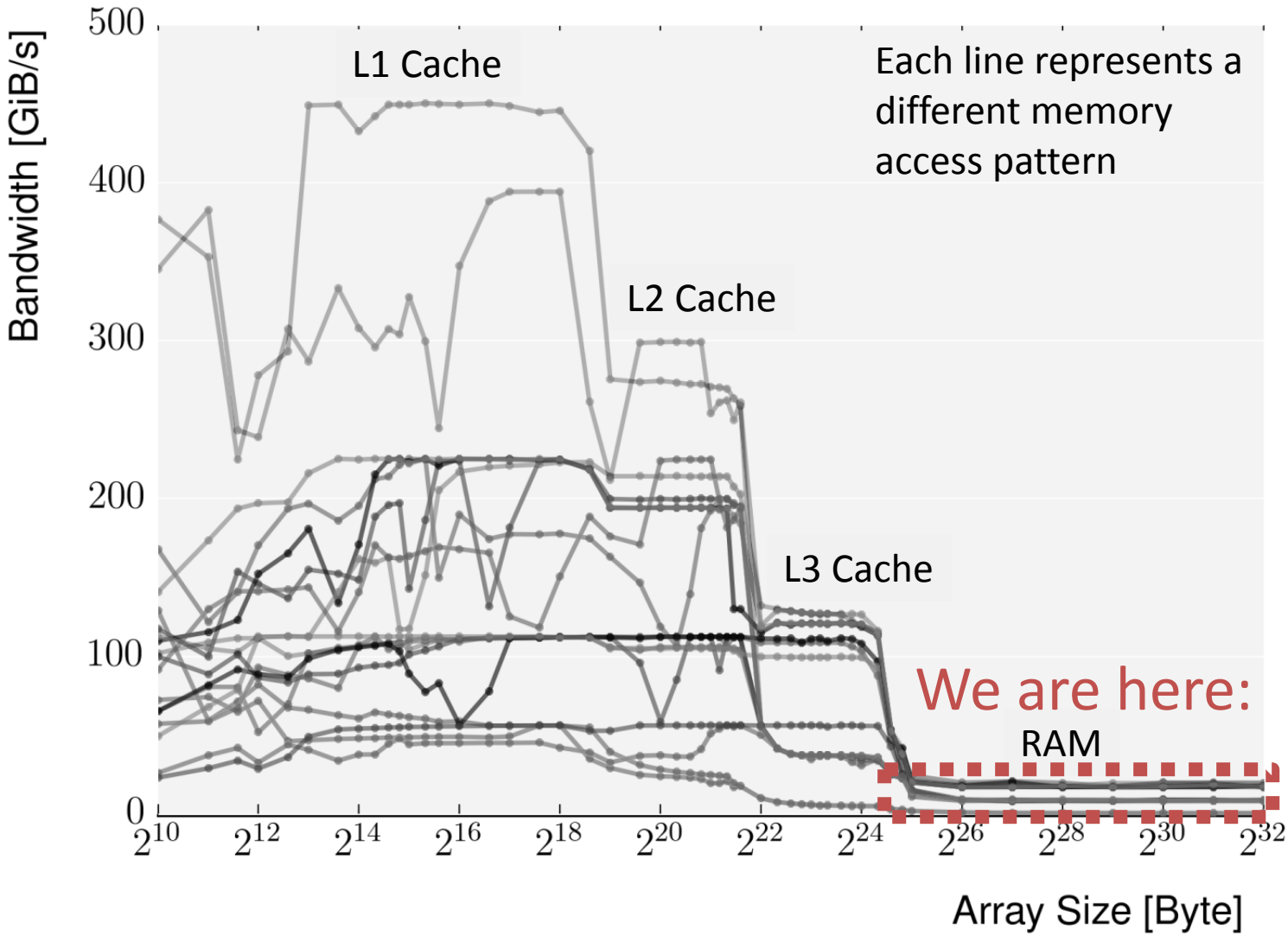


Measurements done on
Intel Xeon E5645
2x6 cores @ 2.40 GHz

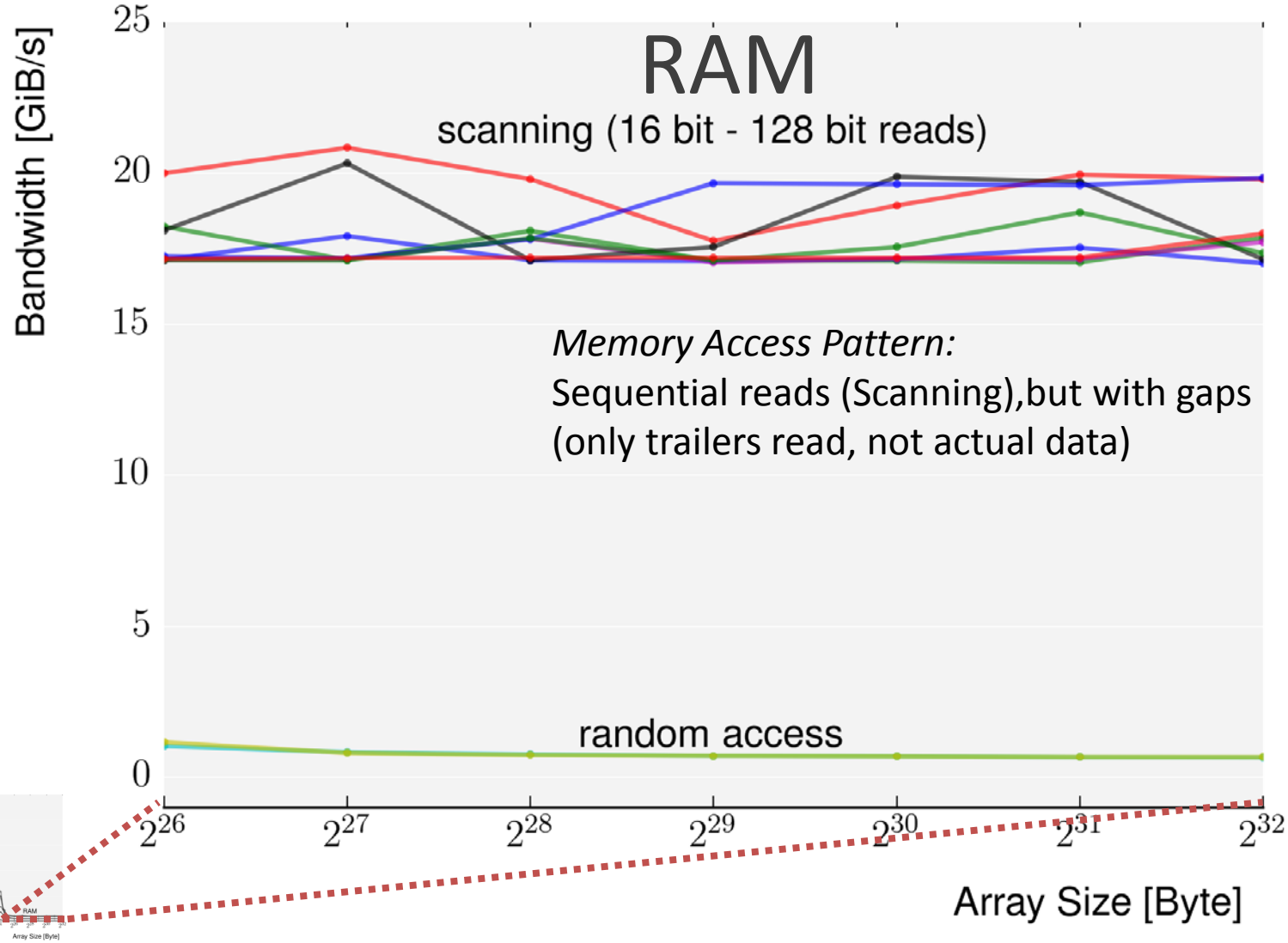
Memory Performance



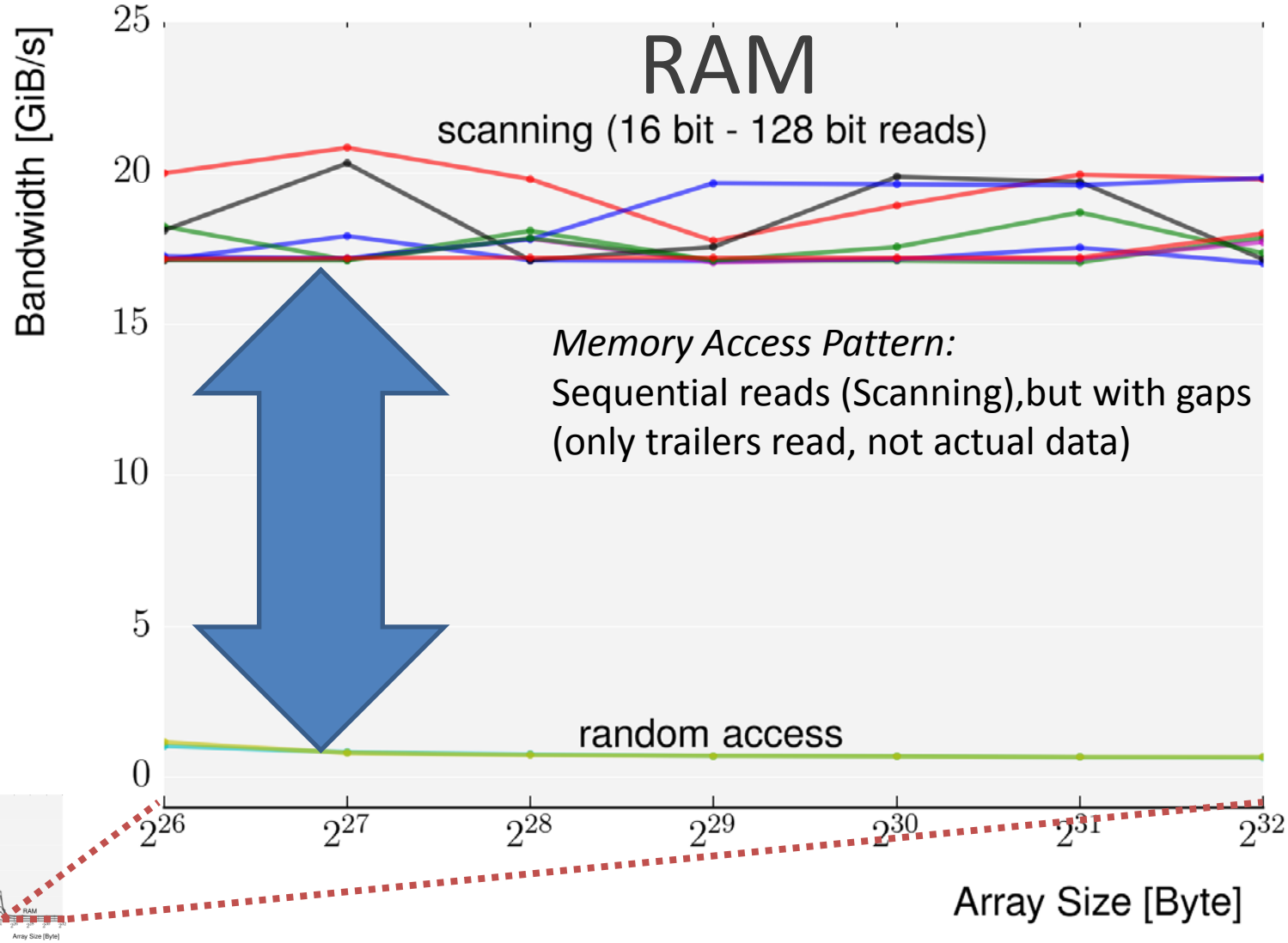
Memory Performance



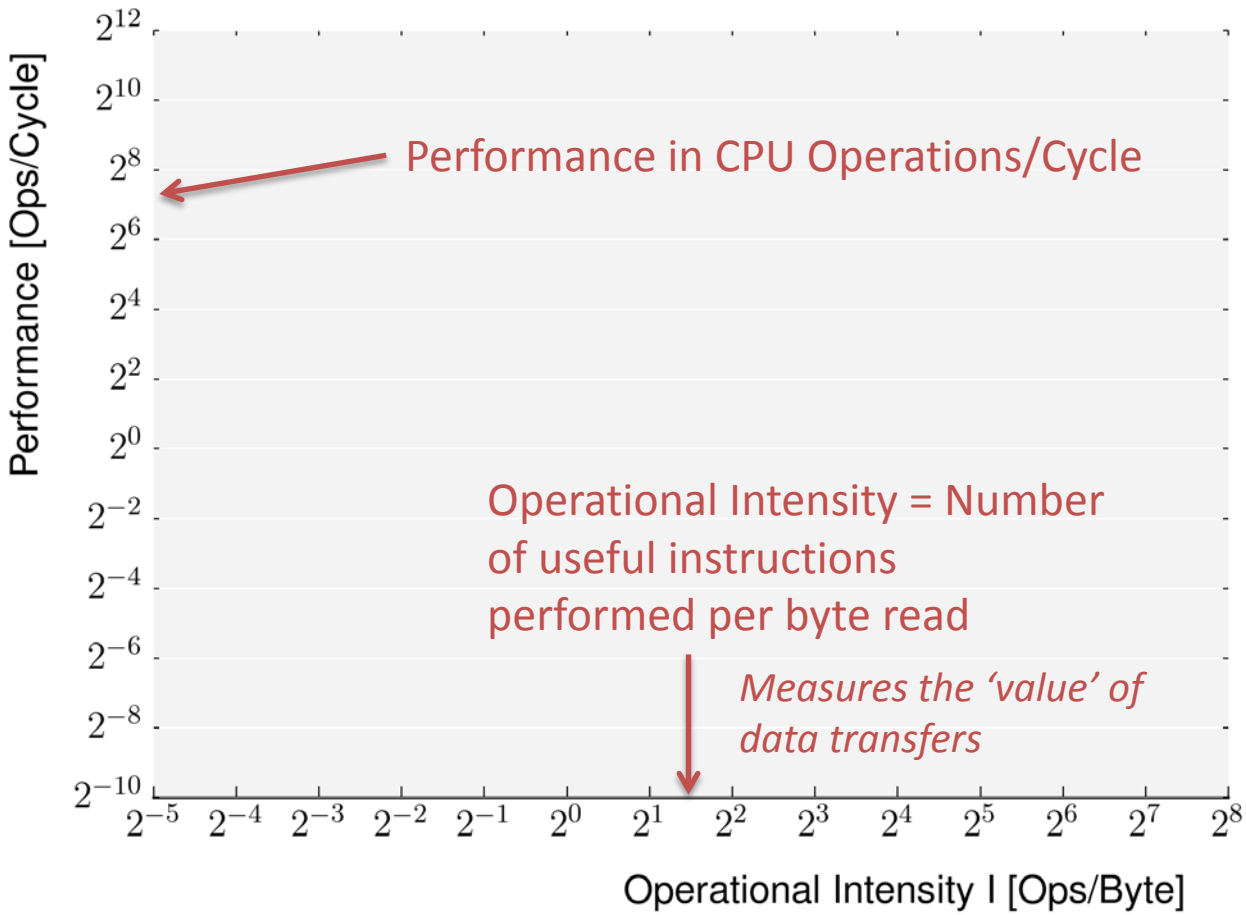
Memory Performance



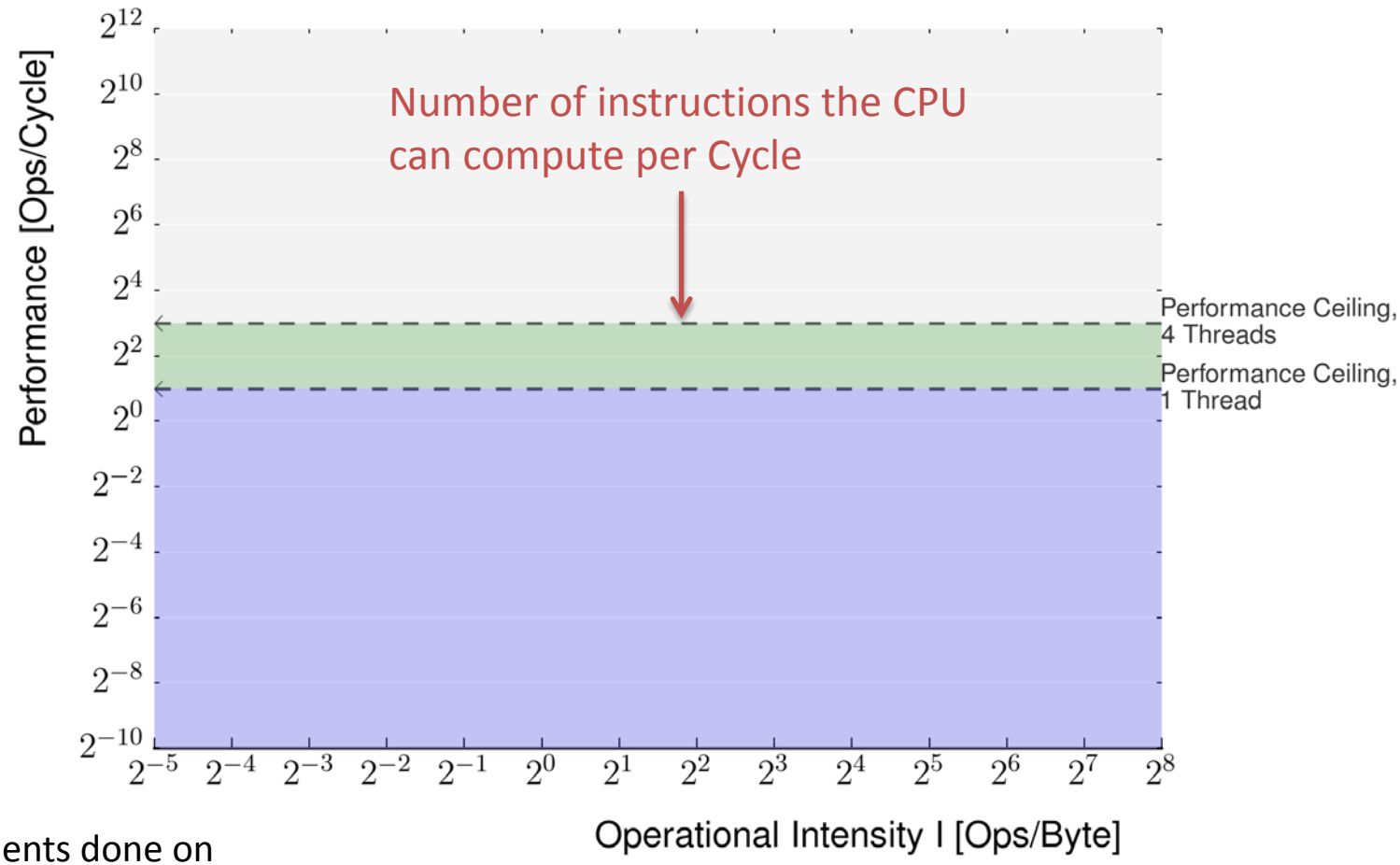
Memory Performance



Roofline Analysis

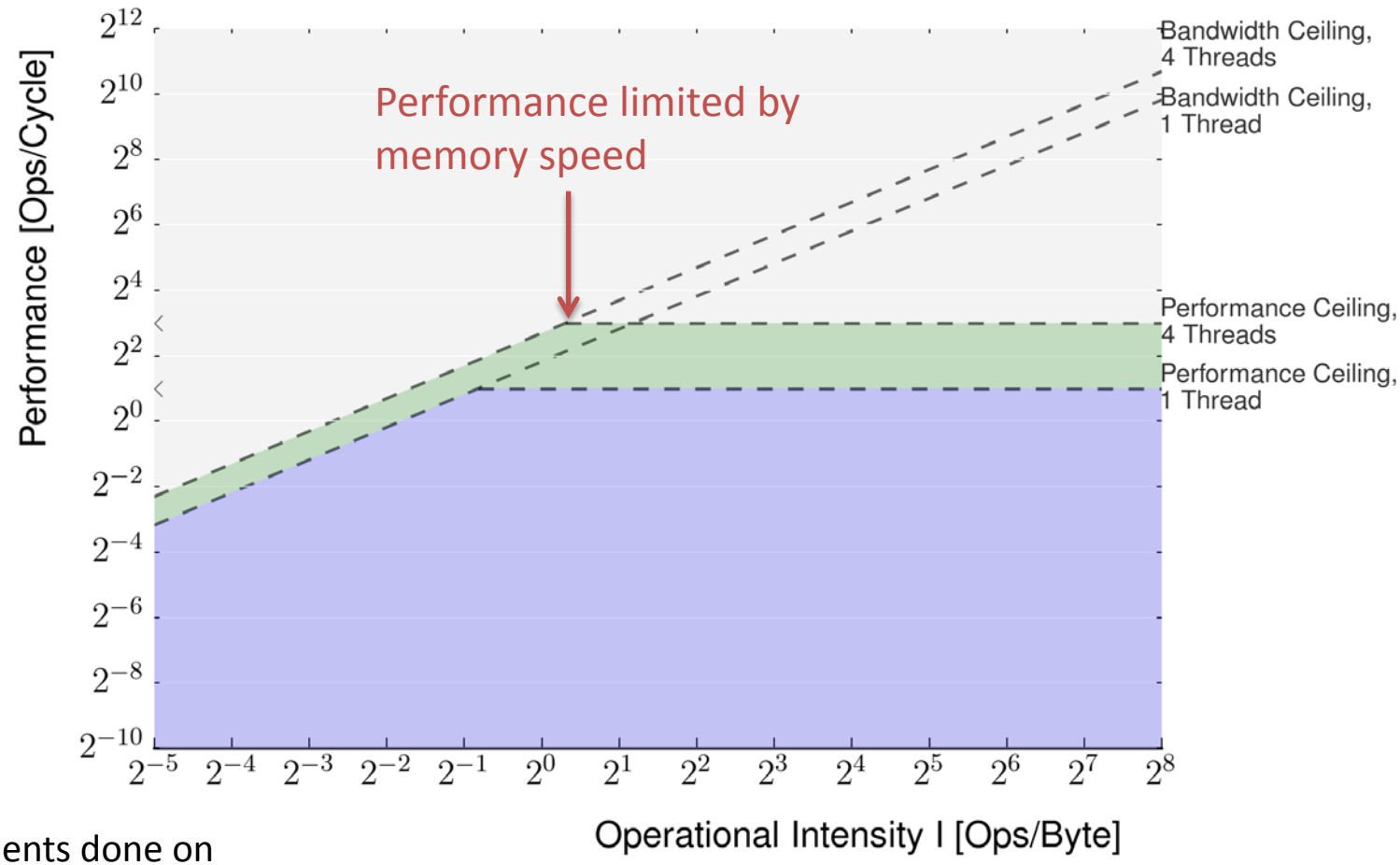


Roofline Analysis



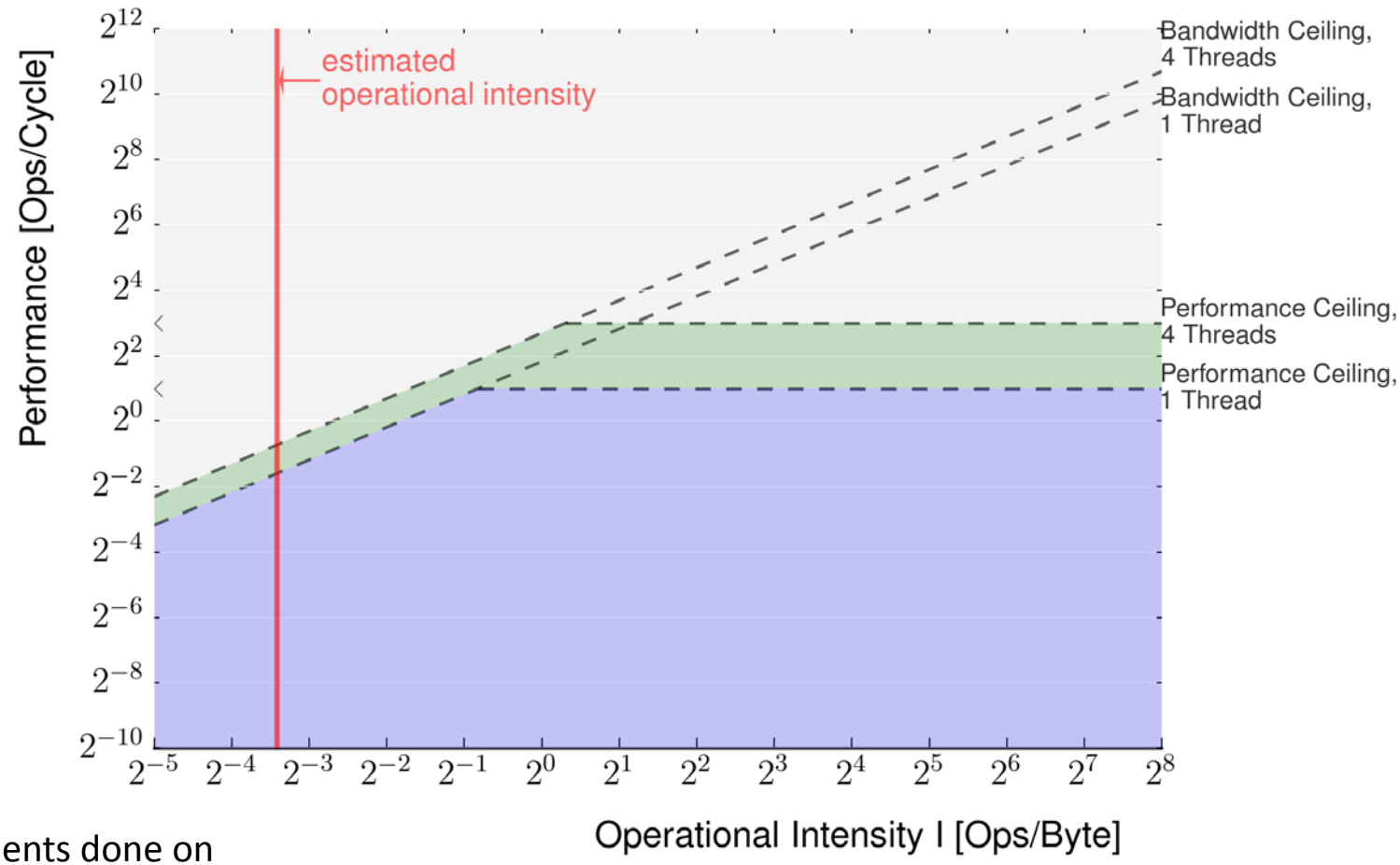
Measurements done on
Intel Core i7-3770 CPU
4 cores @ 3.40 GHz

Roofline Analysis



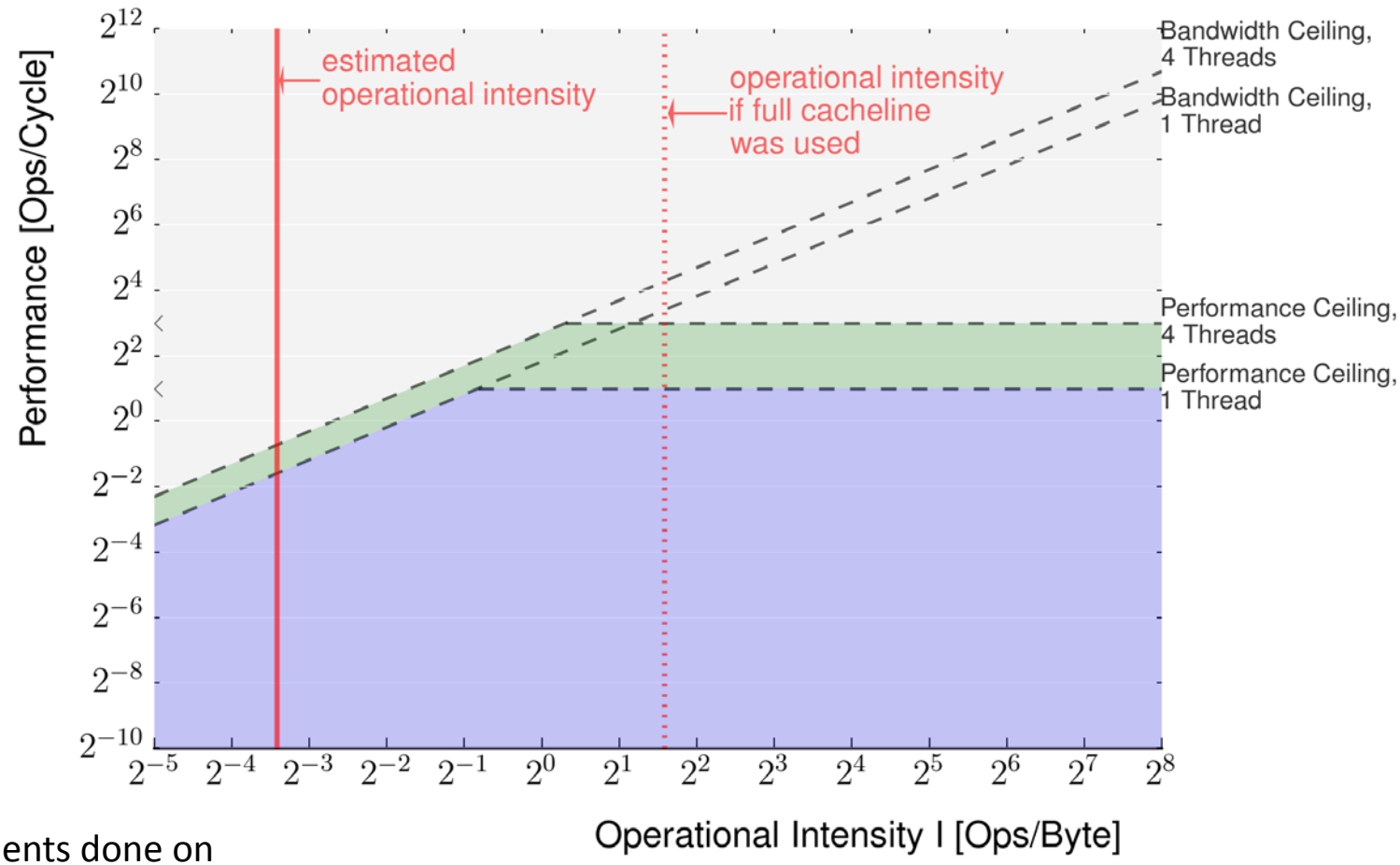
Measurements done on
Intel Core i7-3770 CPU
4 cores @ 3.40 GHz

Roofline Analysis



Measurements done on
Intel Core i7-3770 CPU
4 cores @ 3.40 GHz

Roofline Analysis

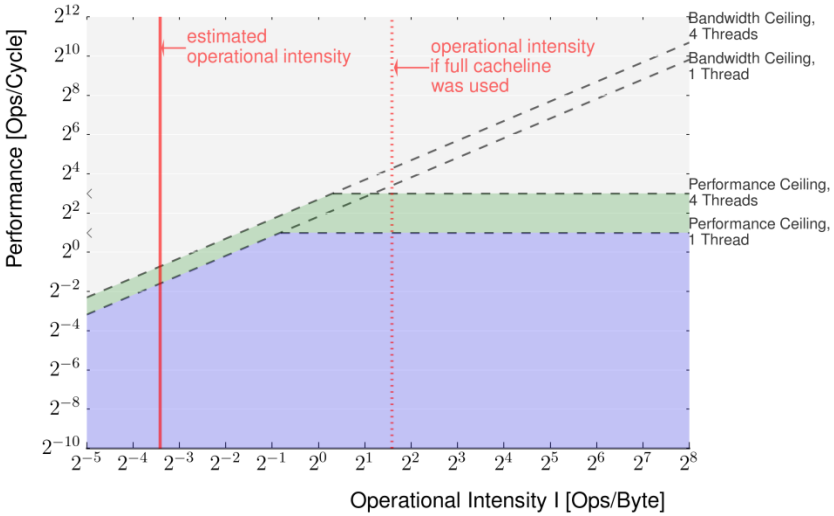


Measurements done on
Intel Core i7-3770 CPU
4 cores @ 3.40 GHz

Roofline Model: Critic

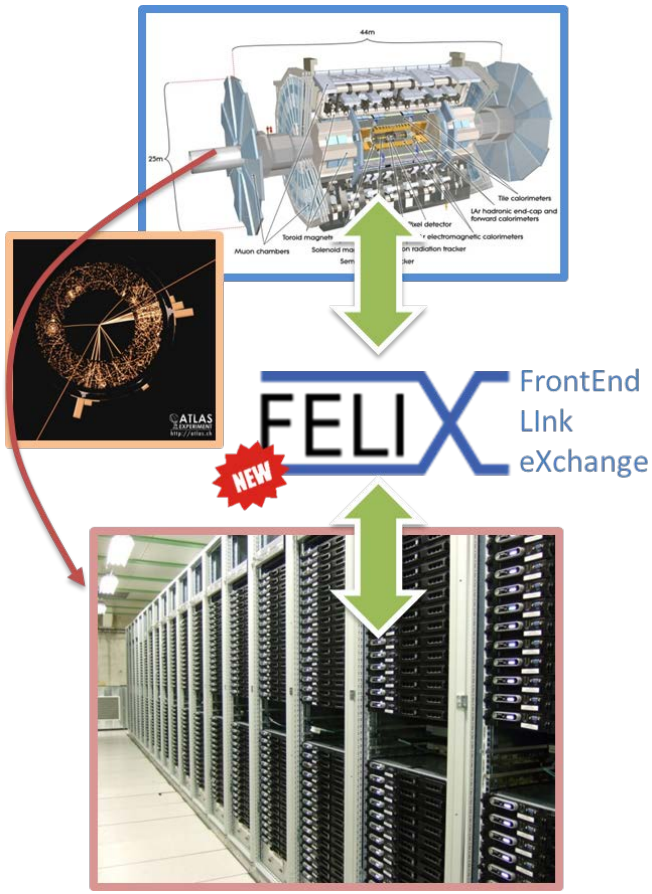
- Hard to obtain accurate data, lots of guessing
- Result can only be seen as a first-order approximation

But: Good way to visualize the results that can be confirmed by a performance analysis tool like Intel VTune



Summary

1. FELIX: A new central event distribution layer for the ATLAS experiment



2. Optimizations and analysis of the application bottleneck (packet processing)