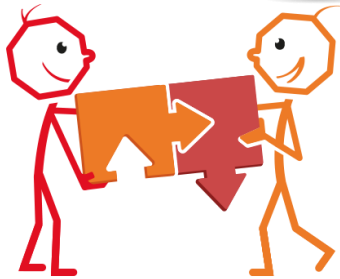
A Large Ion Collider Experiment

O^2 Project :Upgrade of the online and offline computing

ECFA HL-LHC Workshop
PG7: Trigger, Online, Offline and Computing

Pierre VANDE VYVRE for the O^2 project

O^2 Project
Institutes



- Institutes
  - FIAS, Frankfurt, Germany
  - GSI, Darmstadt, Germany
  - IIT, Mumbay, India
  - IPNO, Orsay, France
  - IRI, Frankfurt, Germany
  - Jammu University, Jammu, India
  - Rudjer Bošković Institute, Zagreb, Croatia
  - SUP, Sao Paulo, Brasil
  - University Of Technology, Warsaw, Poland
  - Wiegner Institute, Budapest, Hungary
  - CERN, Geneva, Switzerland
- Active interest from
  - Creighton University, Omaha, US
  - KISTI, Daejeon, Korea
  - KMUTT (King Mongkut's University of Technology Thonburi), Bangkok, Thailand
  - KTO Karatay University, Turkey
  - Lawrence Berkeley National Lab., US
  - LIPI, Bandung, Indonesia
  - Oak Ridge National Laboratory, US
  - Thammasat University, Bangkok, Thailand
  - University of Cape Town, South Africa
  - University of Houston, US
  - University of Talca, Chile
  - University of Tennessee, US
  - University of Texas, US
  - Wayne State University, US

Requirements

Focus of ALICE upgrade on physics probes requiring high statistics: sample 10 nb^{-1}

Online System Requirements
Sample full 50kHz Pb-Pb interaction rate
- current limit at ~500Hz, factor 100 increase
- system to scale up to 100 kHz

TPC drift time >> minimum bias rate
Continuous (trigger-less) read-out

⇨ ~1.1 TByte/s detector readout
However:
- Storage bandwidth limited to a much lower value (design decision/cost)
- Many physics probes have low S/B:
  classical trigger/event filter approach not efficient

O^2 System from the Letter of Intent

Design Guidelines

Handle >1 TByte/s detector input
Produce (timely) physics result

} Online Reconstruction to reduce data volume
Output of System AODs

Minimize “risk” for physics results

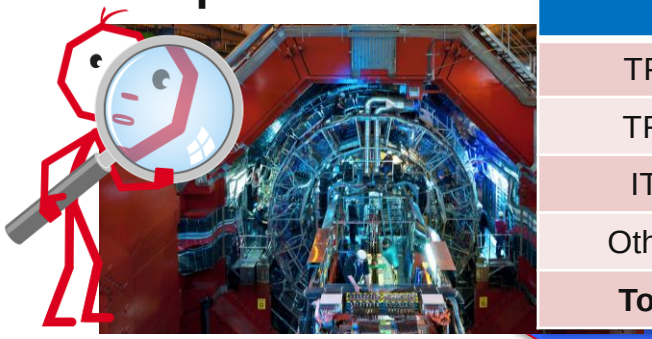
- Allow for reconstruction with improved calibration, e.g. store clusters associated to tracks instead of tracks
- Minimize dependence on initial calibration accuracy

Keep cost “reasonable”

- Limit storage system bandwidth to ~80 GB/s peak and 20 GByte/s average
- Optimal usage of compute nodes

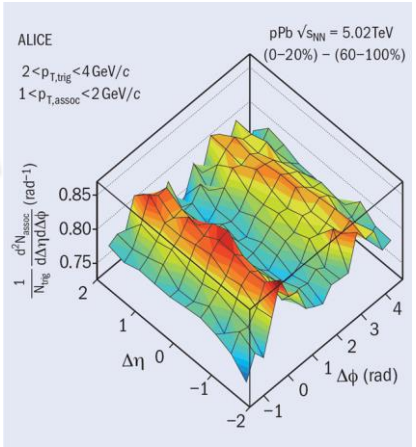
O^2 Project

Requirements



Detector	Input to Online System (GByte/s)	Peak Output to Local Data Storage (GByte/s)	Avg. Output to Computing Center (GByte/s)
TPC	1000	50.0	8.0
TRD	81.5	10.0	1.6
ITS	40	10.0	1.6
Others	25	12.5	2.0
Total	**1146.5**	**82.5**	**13.2**

- Handle >1 TByte/s detector input
- Support for continuous read-out
- Online reconstruction to reduce data volume
- Common hw and sw system developed by the DAQ, HLT, Offline teams

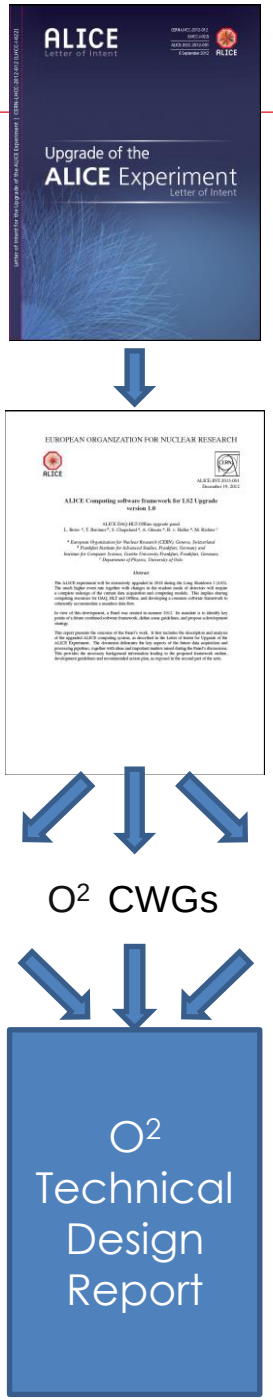


O^2 Project

Project Organization

PLs: P. Buncic, T. Kollegger, P. Vande Vyvre

Computing Working Group(CWG)		Chair
1.	Architecture	S. Chapeland
2.	Tools & Procedures	A. Telesca
3.	Dataflow	T. Breitner
4.	Data Model	A. Gheata
5.	Computing Platforms	M. Kretz
6.	Calibration	C. Zampolli
7.	Reconstruction	R. Shahoyan
8.	Physics Simulation	A. Morsch
9.	QA, DQM, Visualization	B. von Haller
10.	Control, Configuration, Monitoring	V. Chibante
11.	Software Lifecycle	A. Grigoras
12.	Hardware	H. Engel
13.	Software framework	P. Hristov

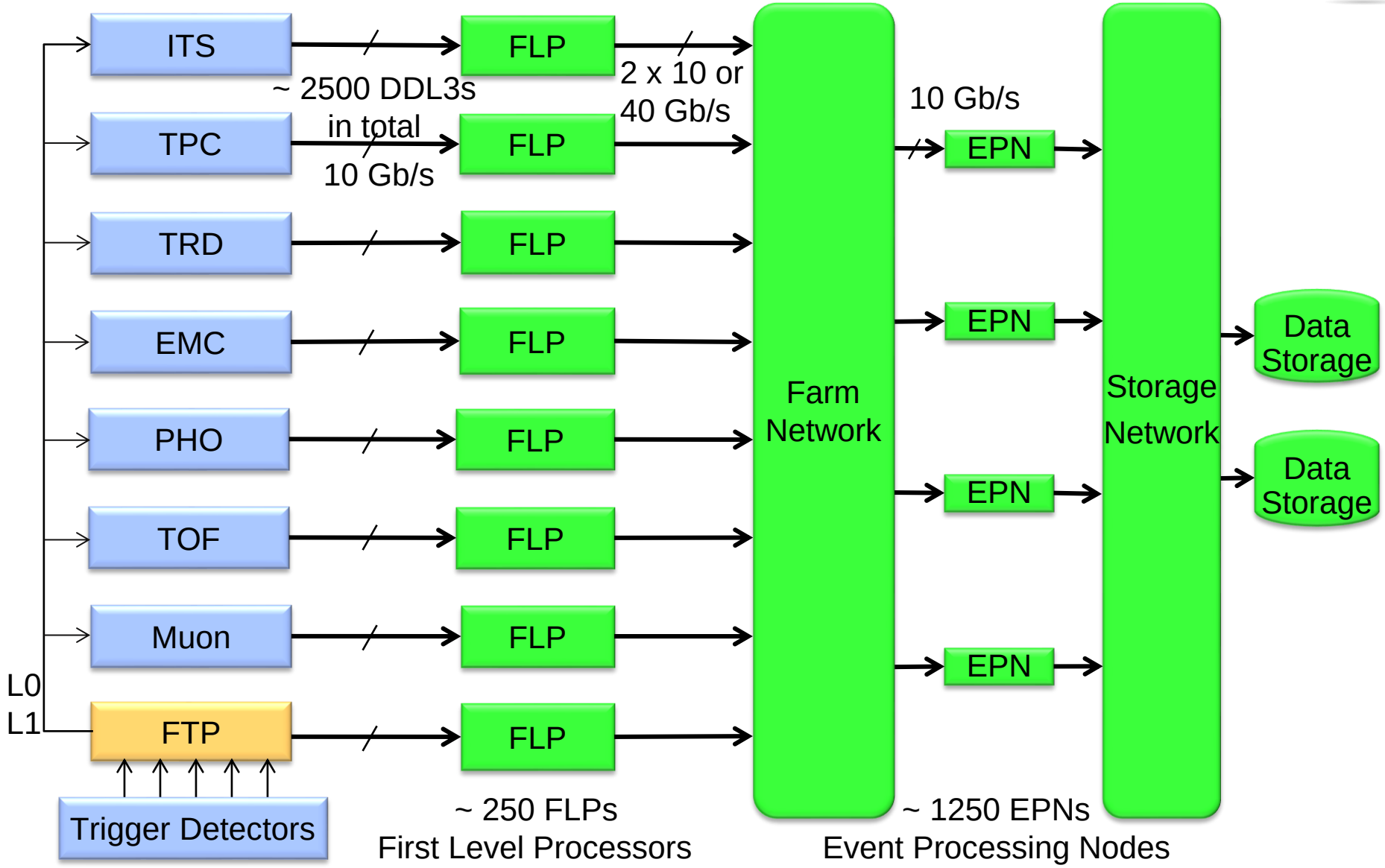


Editorial Committee

L. Betev, P. Buncic, S. Chapeland, F. Cliff, P. Hristov, T. Kollegger, M. Krzewicki, K. Read, J. Thaeder, B. von Haller, P. Vande Vyvre

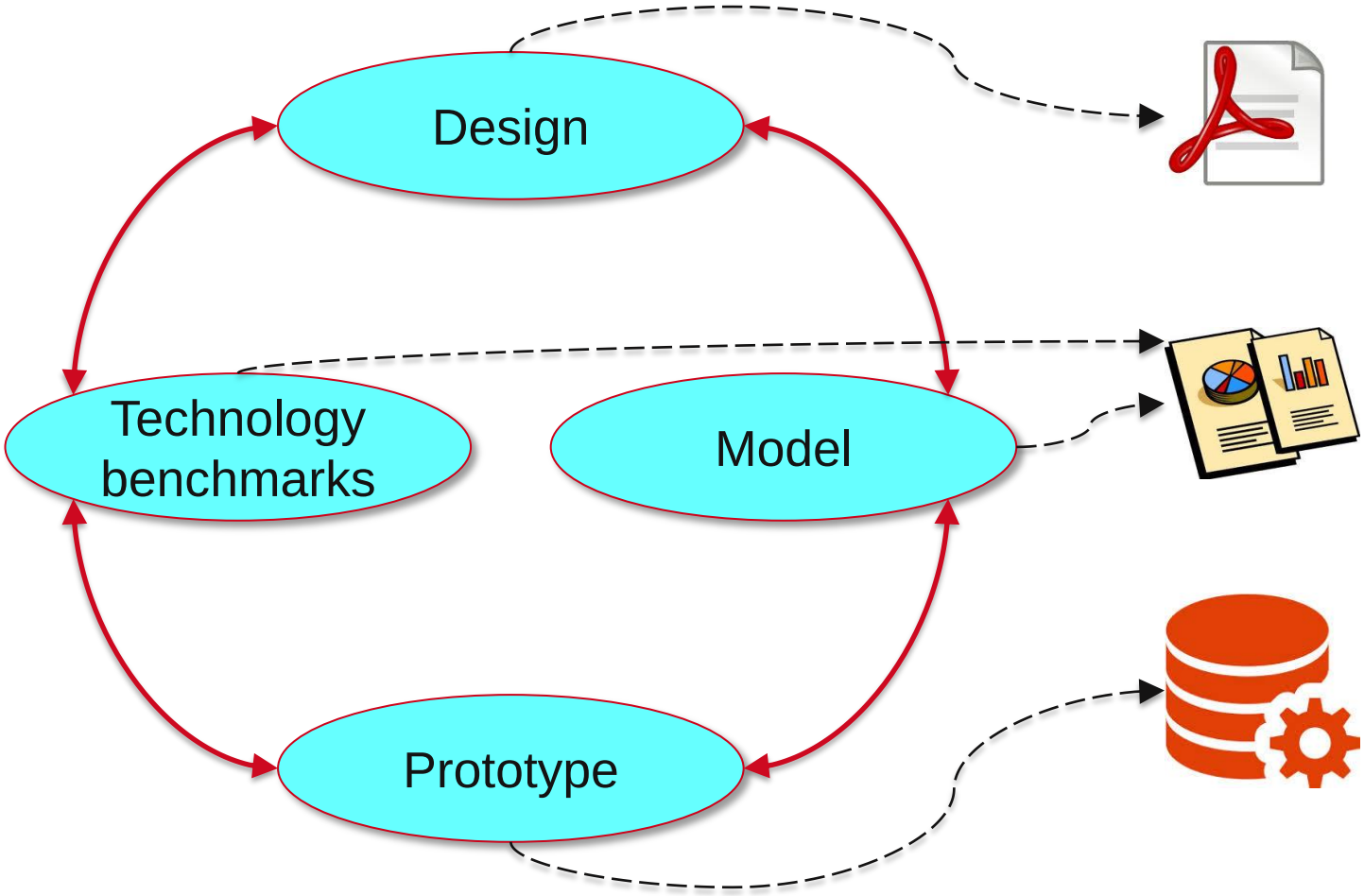
Physics requirement chapter: Andrea Dainese

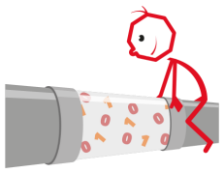
Hardware Architecture



Design strategy

Iterative process: design, benchmark, model, prototype





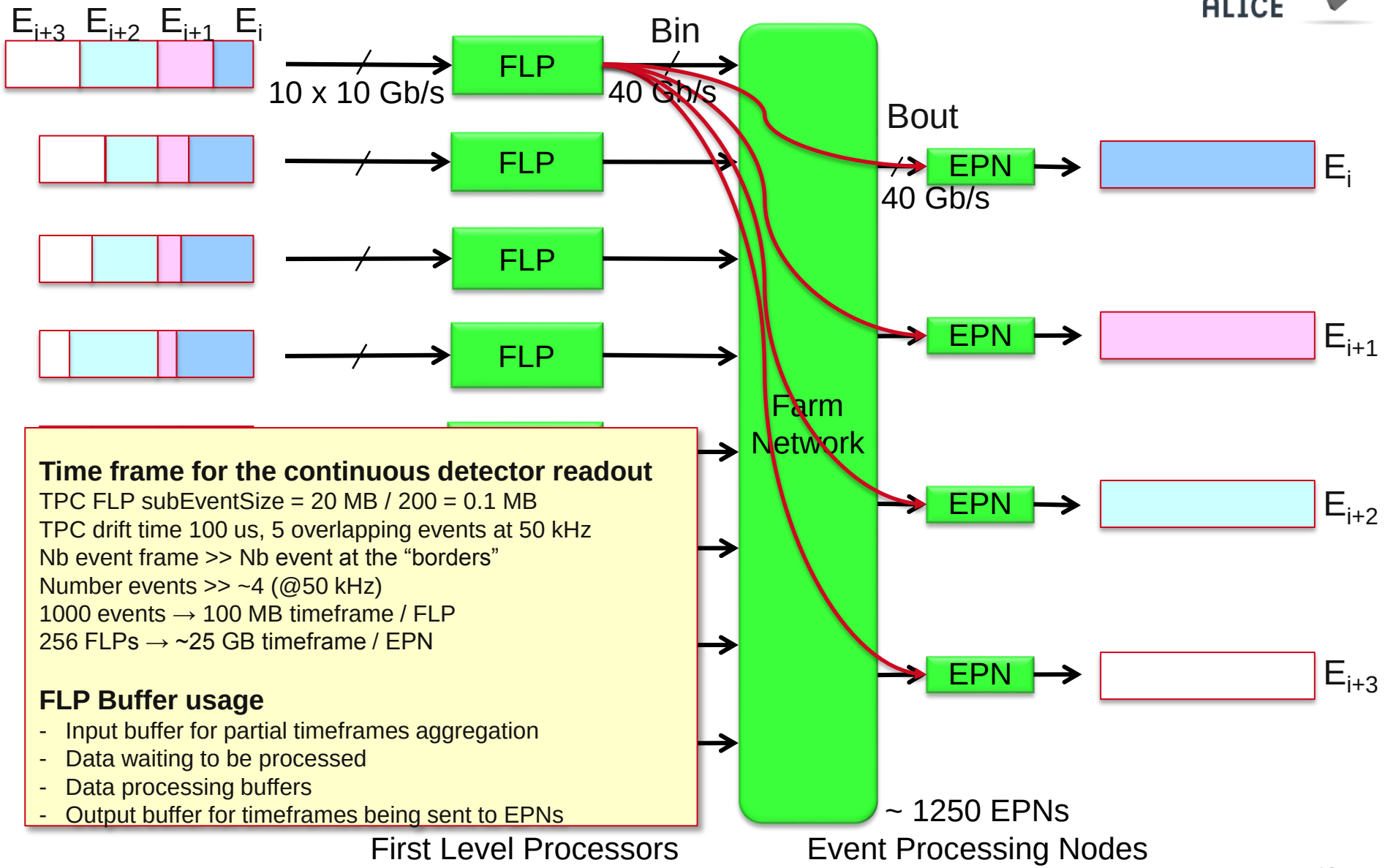
Dataflow

Dataflow modelling

- Dataflow discrete event simulation implemented with OMNET++
  - FLP-EPN data traffic and data buffering
    - Network topologies (central switch; spine-leaf),
    - Data distribution schemes (time frames, parallelism)
    - Buffering needs
    - System dimensions
  - Heavy computing needs

    Downscaling applied for some simulations:
    - Reduce network bandwidth and buffer sizes
    - Simulate a slice of the system
- System global simulation with ad-hoc program

Hardware Architecture: FLP buffer size



Time frame for the continuous detector readout

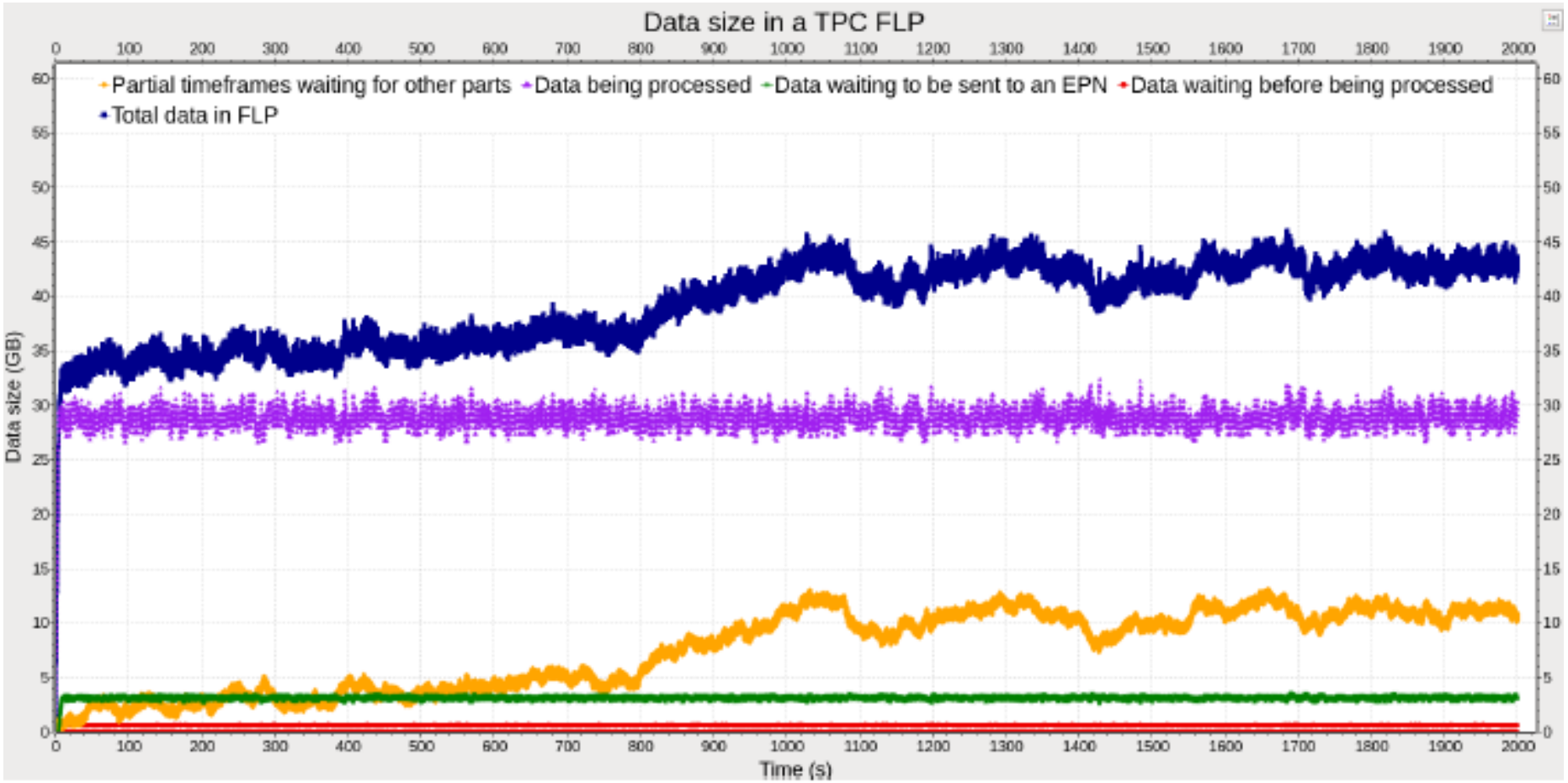
TPC FLP subEventSize = 20 MB / 200 = 0.1 MB
TPC drift time 100 us, 5 overlapping events at 50 kHz
Nb event frame >> Nb event at the “borders”
Number events >> ~4 (@50 kHz)
1000 events → 100 MB timeframe / FLP
256 FLPs → ~25 GB timeframe / EPN

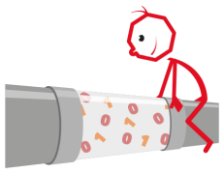
FLP Buffer usage

- Input buffer for partial timeframes aggregation
- Data waiting to be processed
- Data processing buffers
- Output buffer for timeframes being sent to EPNs

FLP-EPN Dataflow simulation

FLP buffer size





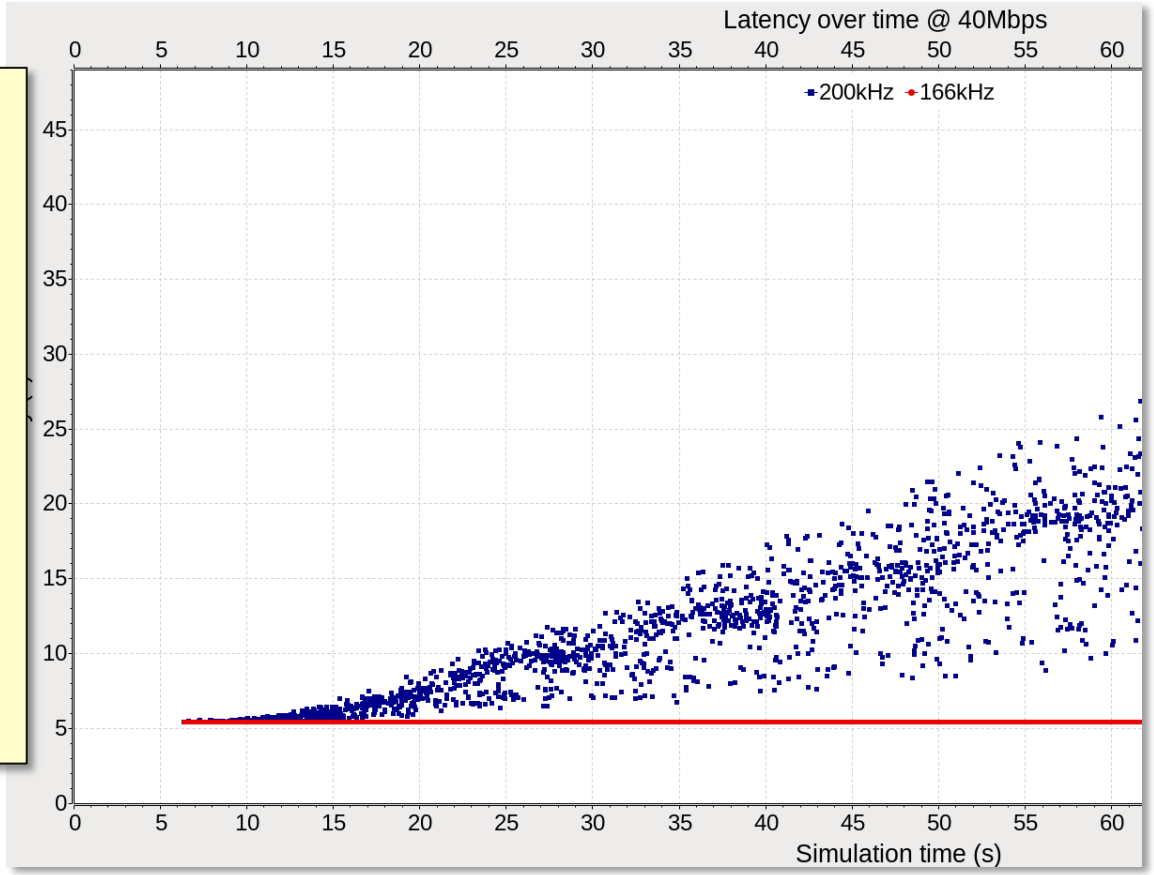
FLP-EPN Dataflow simulation

System scalability study

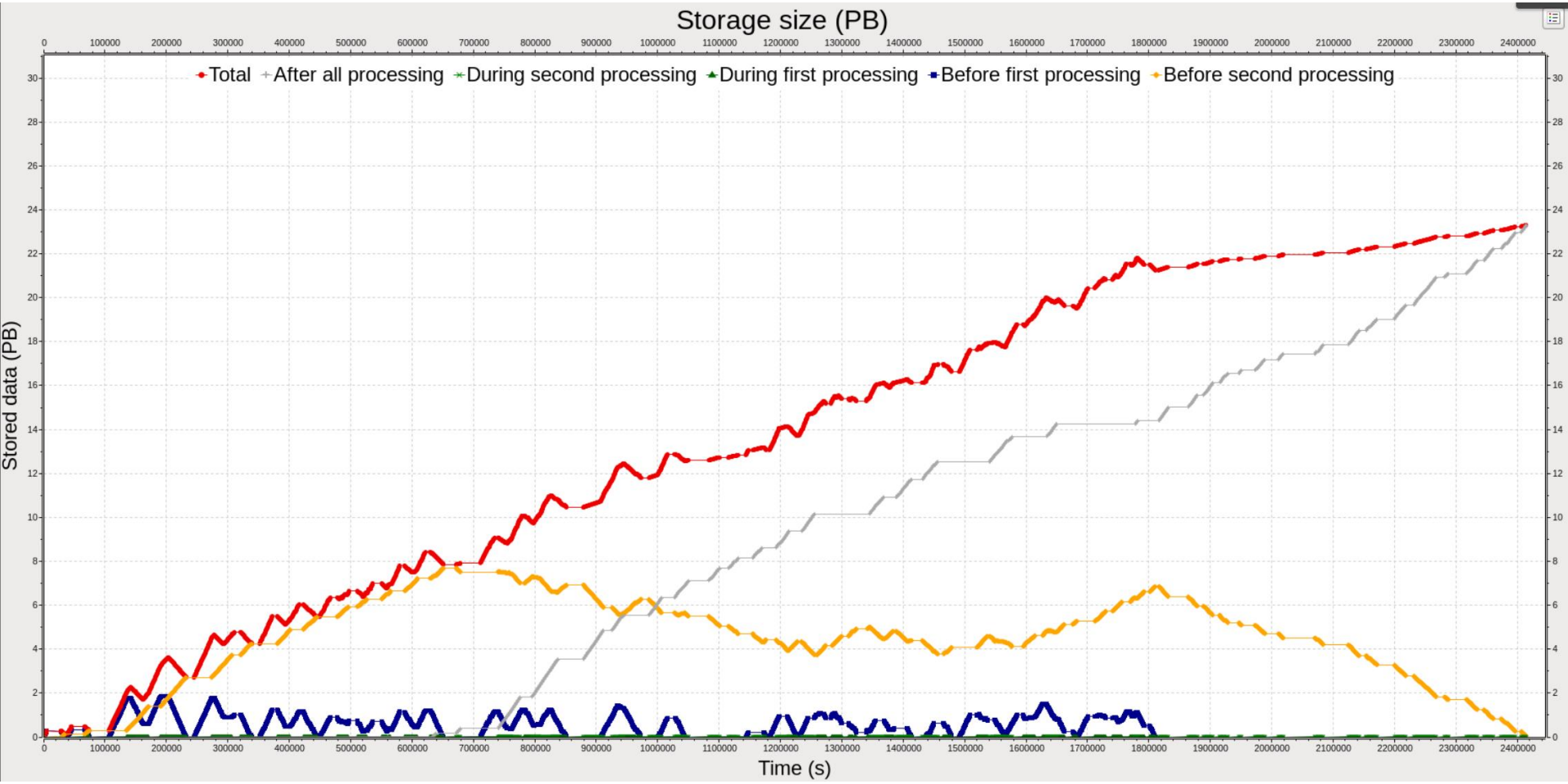
S. Chapeland, C. Delort

- System scalability study
- System studied on a ¼ of the entire system and lower bandwidth to limit the simulation time
- System scales at up to 166 kHz of MB interactions

Configuration 40 Mbps 250x288



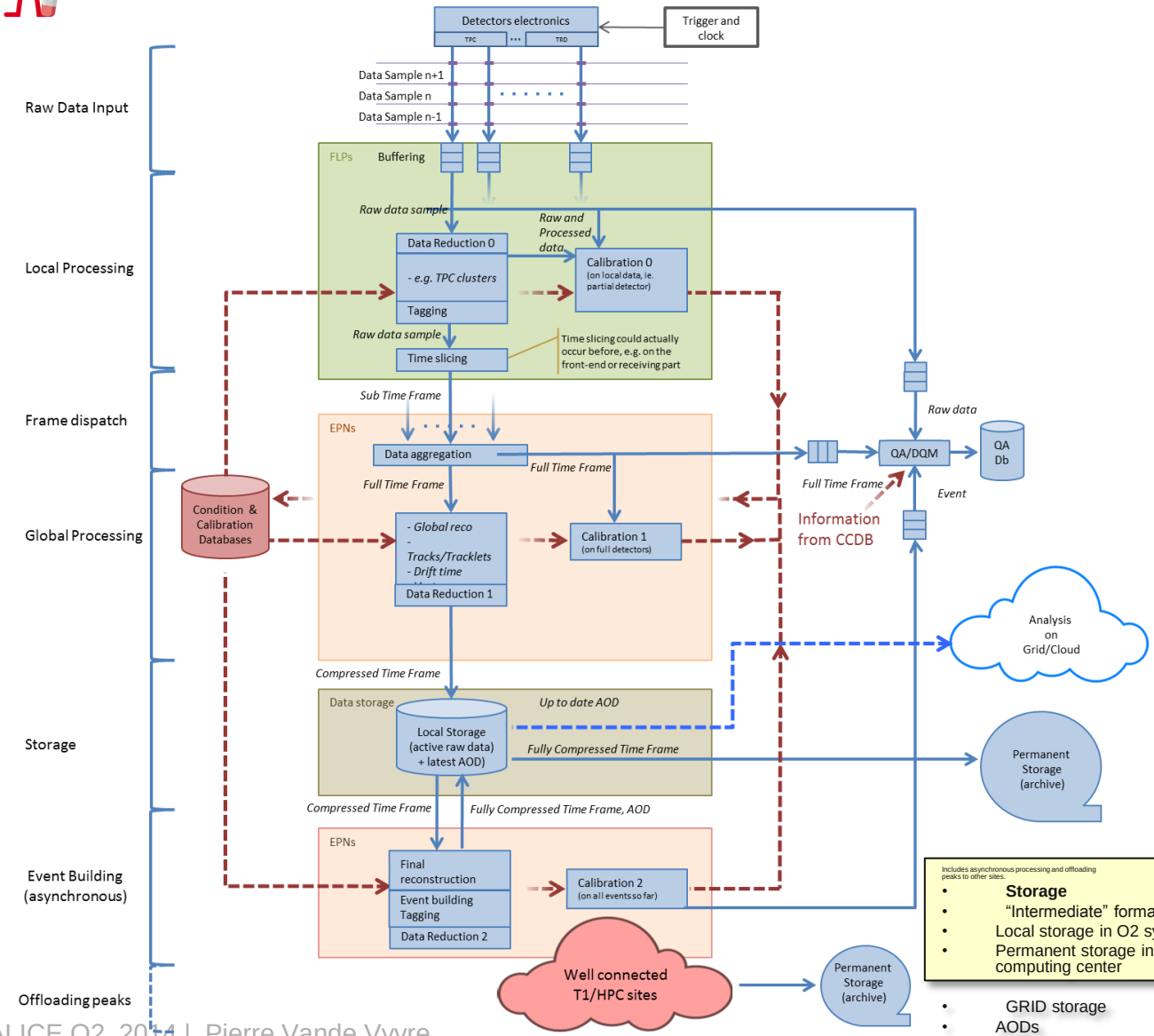
Data storage needs of the O^2 facility



Need of ~25 PB of local data storage for 1 year of data taking

Architecture

O² Architecture and data flow



TPC Data
1 TB/s

250 GB/s

125 GB/s

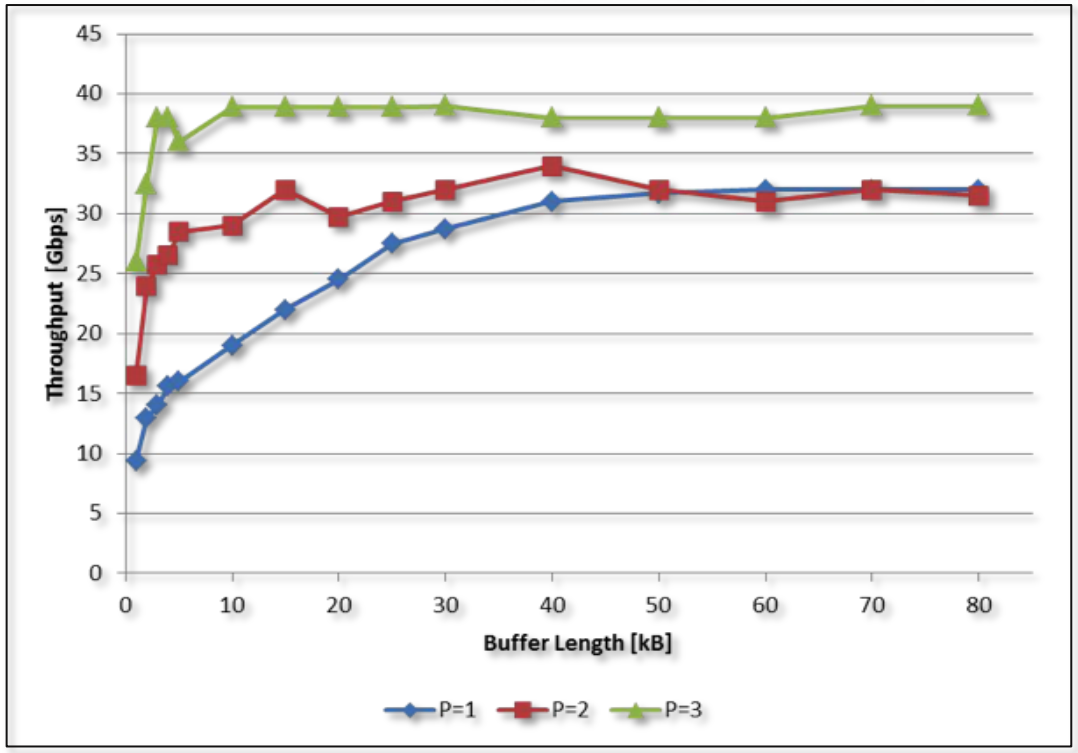
80 GB/s

FLP and Network prototyping

- FLP requirements
  - Input 100 Gbit/s (10 x 10 Gbit/s)
  - Local processing capability
  - Output with ~20 Gbit/s

- Two network technologies under evaluation
  - 10/40 Gbit/s Ethernet
  - Infiniband FDR (56 Gbit/s)
  - Both used already (DAQ/HLT)

- Benchmark example
- Chelsio T580-LP-CR with TCP/UDP Offload engine
  1, 2 and 3 TCP streams,
  iperf measurements



CWG5: Computing Platforms

The Conversion factors

- Shift from 1 to many platforms
- Speedup of CPU Multithreading:
  - Task takes **n1** seconds on **1** core, **n2** seconds on **x** cores
  - → Speedup is **n1**/**n2** for **x** cores, Factors are **n1**/**n2** and **x**/**1**
  - With Hyperthreading: **n2‘** seconds on **x‘** threads on **x** cores. (**x‘** >= 2**x**)
  - → Will not scale linearly, needed to compare to full CPU performance.
    - Factors are **n1** / **n2‘** and **x** / **1** (Be carefull: Not **x‘** / **1**, we still use only **x** cores.)
- Speedup of GPU v.s. CPU:
  - Should take into account full CPU power (i.e. all cores, hyperthreading).
  - Task on the GPU might also need CPU resources.
    - Assume this occupies **y** CPU cores.
  - Task takes **n3** seconds on GPU.
  - Speedup is **n2‘**/**n3**, Factors are **n2‘**/**n3** and **y**/**x**. (Again **x** not **x‘**.)
- How many CPU cores does the GPU save:
  - Compare to **y** CPU cores, since the GPU needs that much resources.
  - Speedup is **n1** / **n3**, GPU Saves **n1** / **n3** – **y** CPU cores.
  - → Factors are **n1** / **n3**, **y** / **1**, and **n1** / **n3** - **y**.
- Benchmarks: Track Finder, Track Fit, DGEMM (Matrix Multiplication – Synthetic)

CWG5: Computing Platforms

Track finder

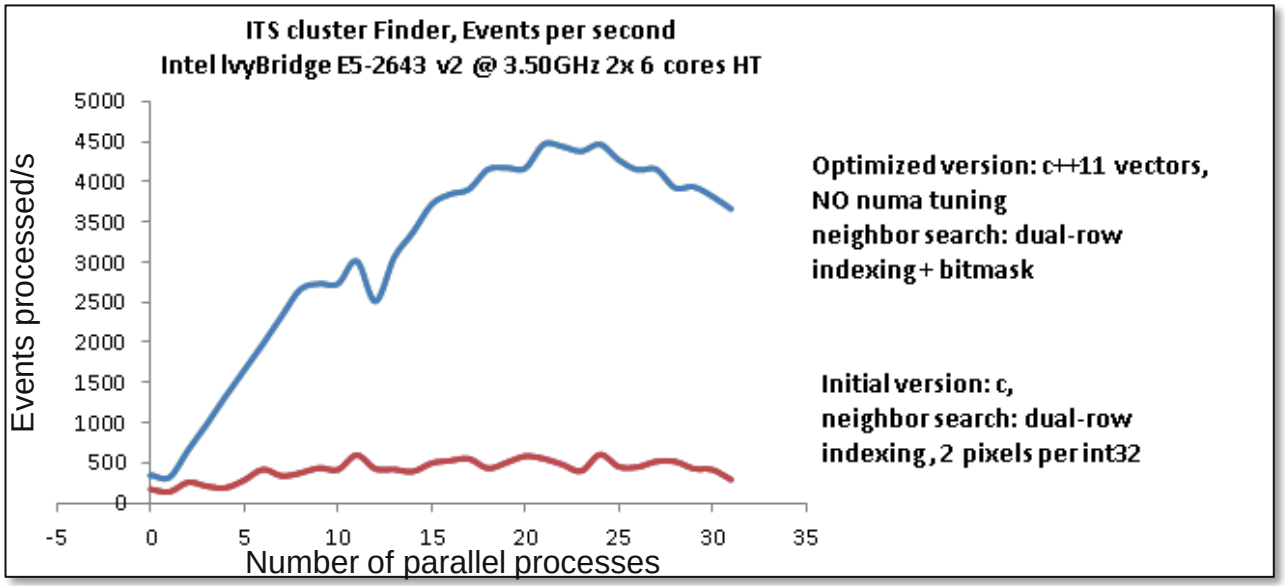
Nehalem 4-Core 3,6 GHz (Smaller Event than others)		
1 Thread	3921 ms	**Factors:**
4 Threads	1039 ms	3,77 / 4
12 Threads (x = 4, x‘ = 12)	816 ms	4,80 / 4
Westmere 6-Core 3.6 GHz		
1 Thread	4735 ms	**Factors:**
6 Threads	853 ms	5.55 / 6
12 Threads (x = 4, x‘ = 12)	506 ms	9,36 / 6
Dual Sandy-Bridge 2 * 8-Core 2 GHz		
1 Thread	4526 ms	**Factors:**
16 Threads	403 ms	11,1 / 16
36 Threads (x = 16, x‘ = 36)	320 ms	14,1 / 16
Dual AMD Magny-Cours 2 * 12-Core 2,1 GHz		
36 Threads (x = 24, x‘ = 36)	495 ms	

3 CPU Cores + GPU – All Compared to Sandy Bridge System			
		Factor vs x‘ (Full CPU)	**Factor vs 1 (1 CPU Core)**
GTX580	174 ms	1,8 / 0,19	26 / 3 / 23
GTX780	151 ms	2,11 / 0,19	30 / 3 / 27
Titan	143 ms	2,38 / 0,19	32 / 3 / 29
S9000	160 ms	2 / 0,19	28 / 3 / 25
S10000 (Dual GPU with 6 CPU cores	85 ms	3,79 / 0,38	54 / 6 / 48

Computing Platforms

ITS Cluster Finder

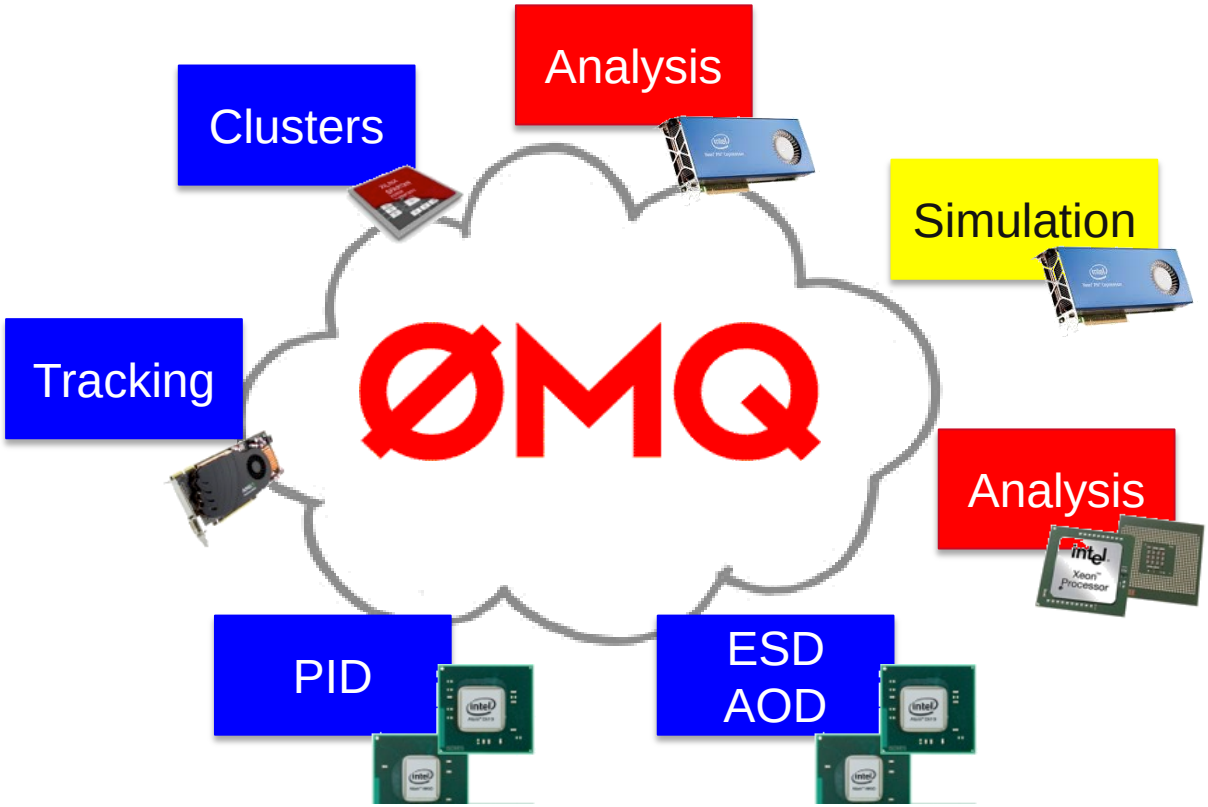
- Use the ITS cluster finder as optimization use case and as benchmark
- Initial version memory-bound
- Several data structure and algorithms optimizations applied



S. Chapeland

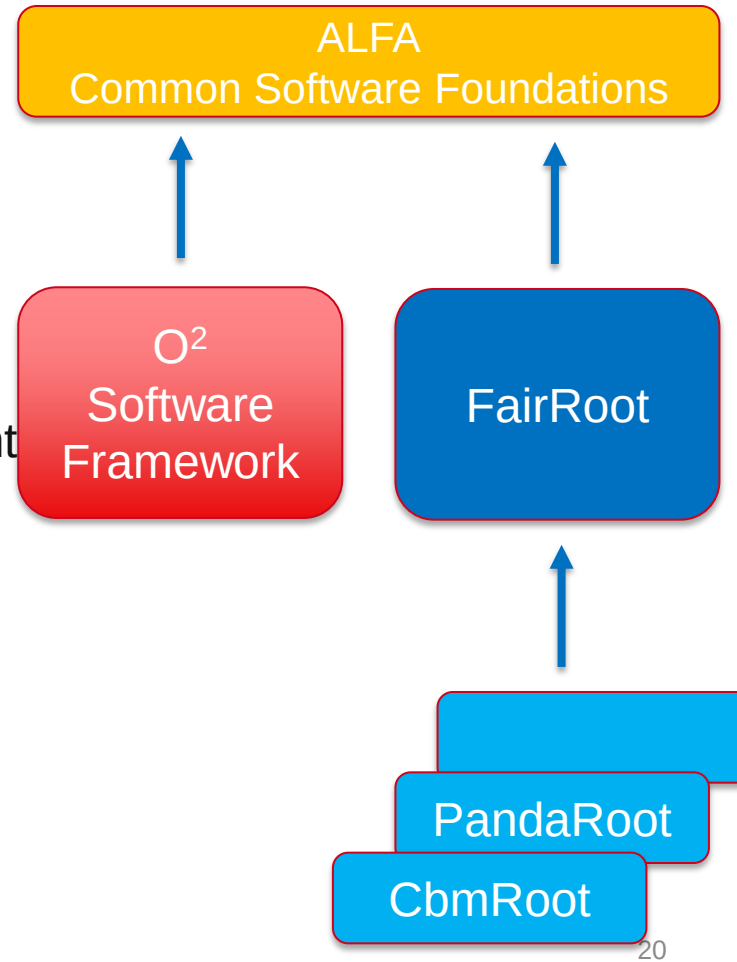
Software Framework

- Multi-platforms
- Multi-applications
- Public-domain software



Software Framework Development

- Design and development of a new modern framework targeting Run3
- Should work in Offline and Online environment
  - Has to comply with O^2 requirements and architecture
- Based on new technologies
  - Root 6.x, C++11
- Optimized for I/O
  - New data model
- Capable of utilizing hardware accelerators
  - FPGA, GPU, MIC…
- Support for concurrency and distributed environment
- Based on ALFA - common software foundation developed jointly between ALICE & GSI/FAIR



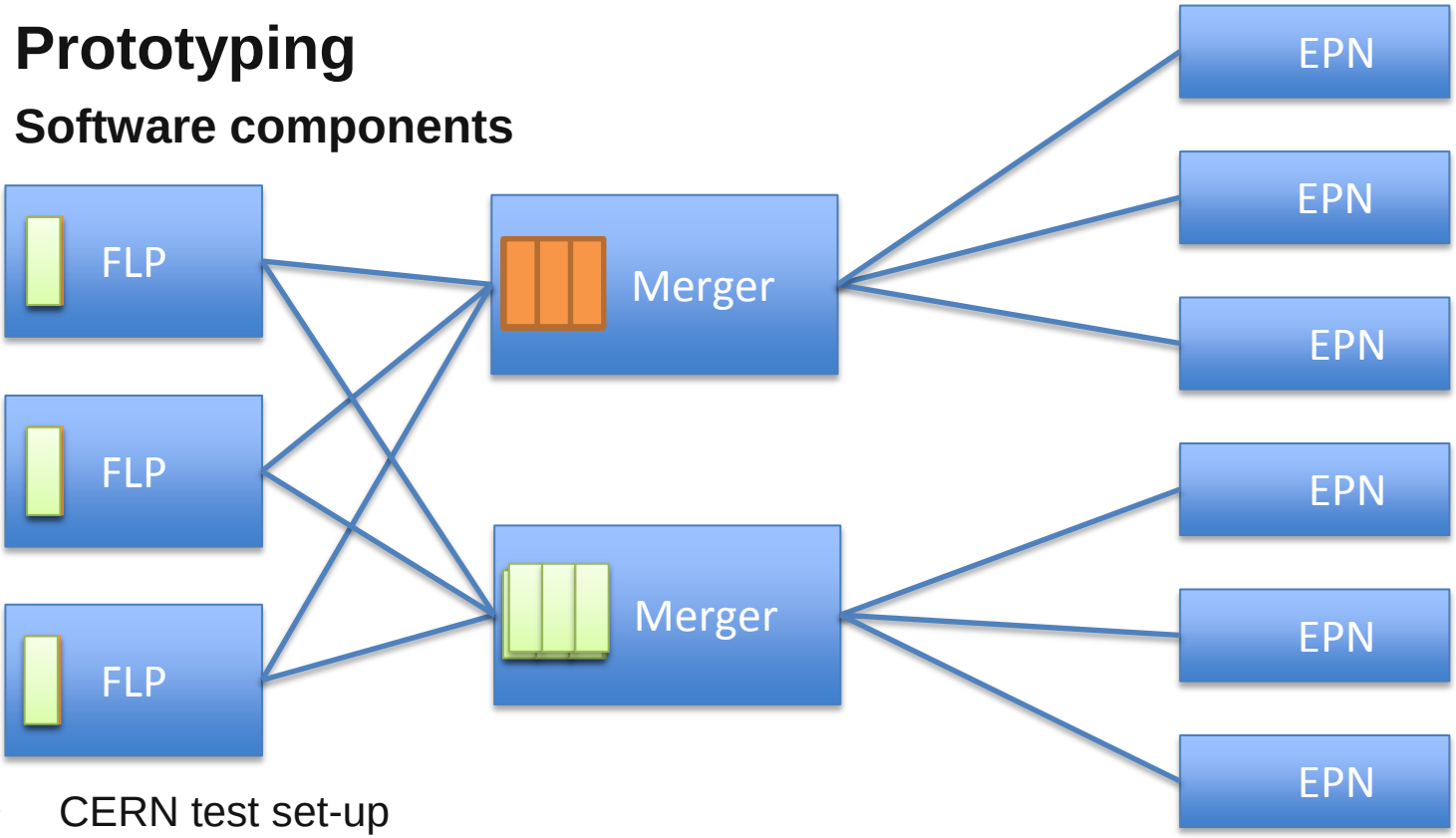
Software Framework Development

ALICE + FAIR = ALFA

- Expected benefits
  - Development cost optimization
  - Better coverage and testing of the code
  - Documentation, training and examples.
  - **ALICE** : work already performed by the FairRoot team concerning features (e.g. the continuous read-out), which are part of the ongoing FairRoot development.
  - **FAIR** experiments : ALFA could be tested with real data and existing detectors before the start of the FAIR facility.
- The proposed architecture will rely:
  - A dataflow based model
  - A process-based paradigm for the parallelism
    - Finer grain than a simple match 1 batch on 1 core
    - Coarser grain than a massively thread-based solution

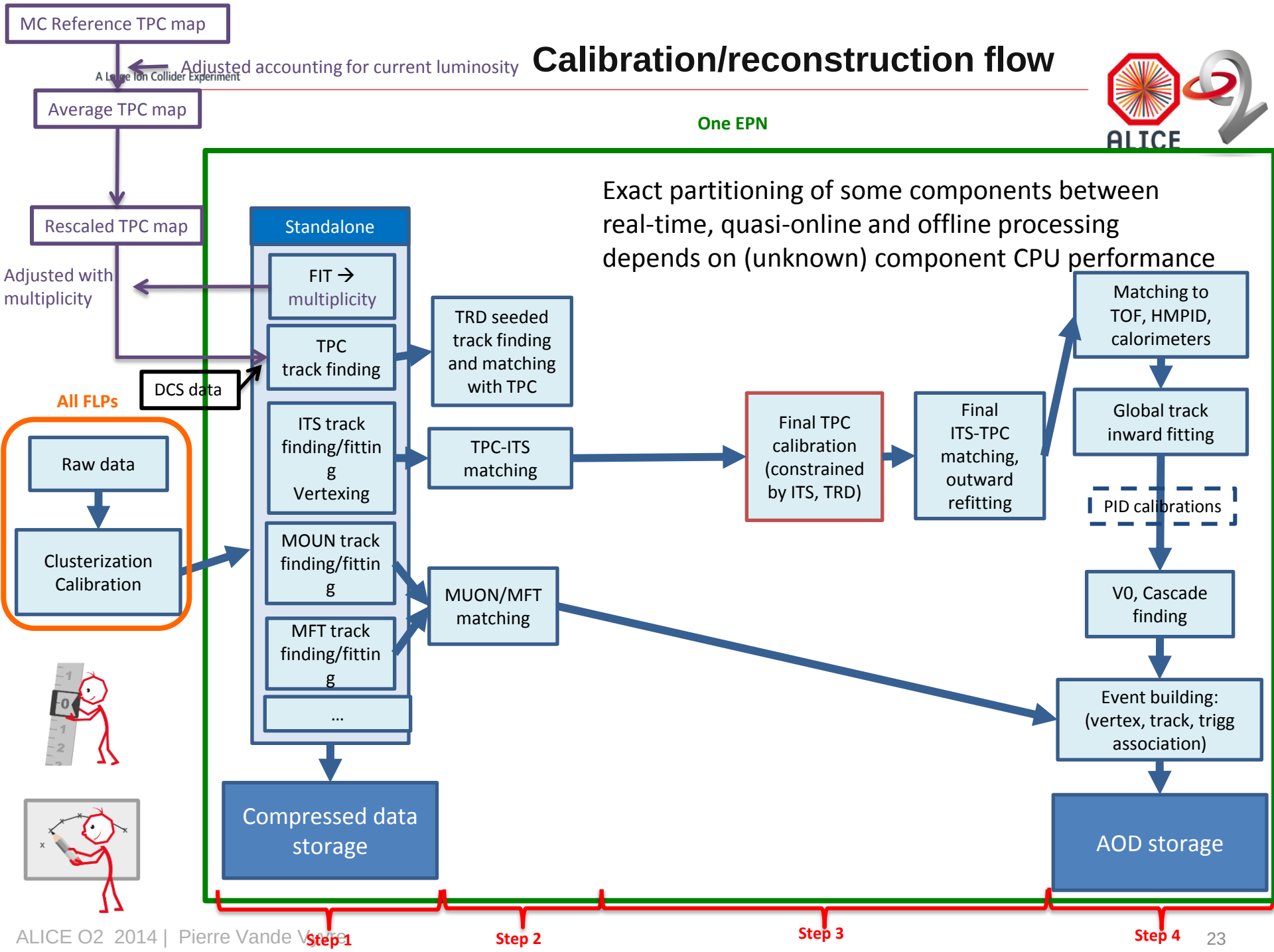
Prototyping

Software components



- CERN test set-up
  - 8 machines : Sandy Bridge-EP, dual E5-2690 @ 2.90GHz, 2x8 hw cores - 32 threads, 64GB RAM
  - Network : 4 nodes with 40 G Ethernet, 4 nodes with 10 G Ethernet
- GSI test set-up
- Software framework prototype by members of DAQ, HLT, Offline, FairRoot teams
  - Data exchange messaging system
  - Interfaces to existing algorithmic code from offline and HLT

Calibration/reconstruction flow



Future steps

- A new computing system (O^2) should be ready for the ALICE upgrade during the LHC LS2 (currently scheduled in 2018-19).
- The ALICE O^2 R&D effort has started in 2013 and is progressing well
  - Design, modelling, benchmarks, prototype
  - Development of the Alfa framework in collaboration with the sottware team of GSI
- Schedule
  - June ‘15 : submission of TDR, finalize the project funding
  - ‘16 – ’17: technology choices and software development
  - June ’18 – June ’20: installation and commissioning