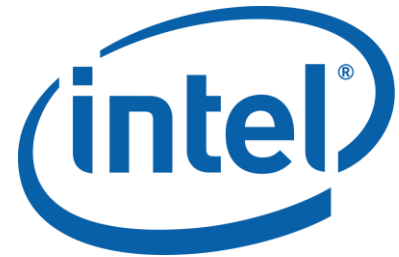


DAQDB: a Key-Value store for Data Acquisition Systems

Danilo Cicalese
on behalf of the **DAQDB** team
OpenLab – CERN
January 24, 2019



Who are we?



Motivation

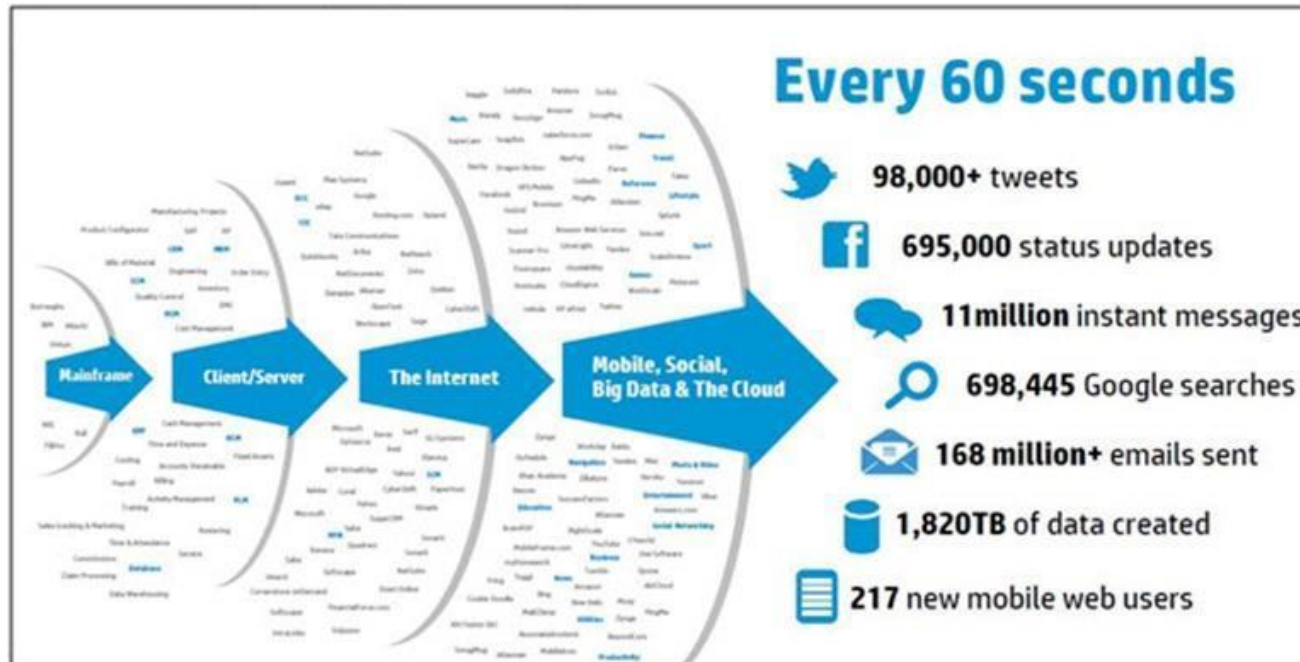


The amount of data created across the world is exploding to new levels.

Motivation



The amount of data created across the world is exploding to new levels.



A. Memon et al. (2017). Big Data Analytics and Its Applications. Annals of Emerging Technologies in Computing.

Motivation



The amount of data created across the world is exploding to new levels.

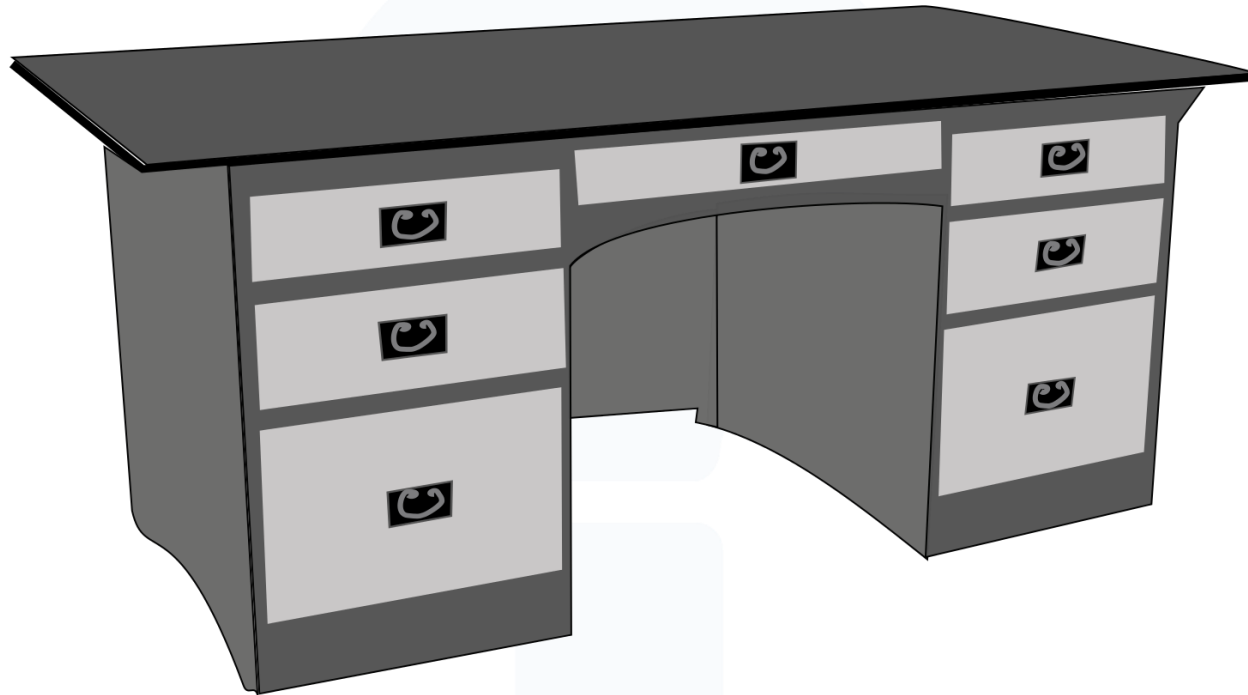
CERN experiments will produce *hundreds of petabytes a day*.

Motivation



**Where will we store
this information?**

Motivation

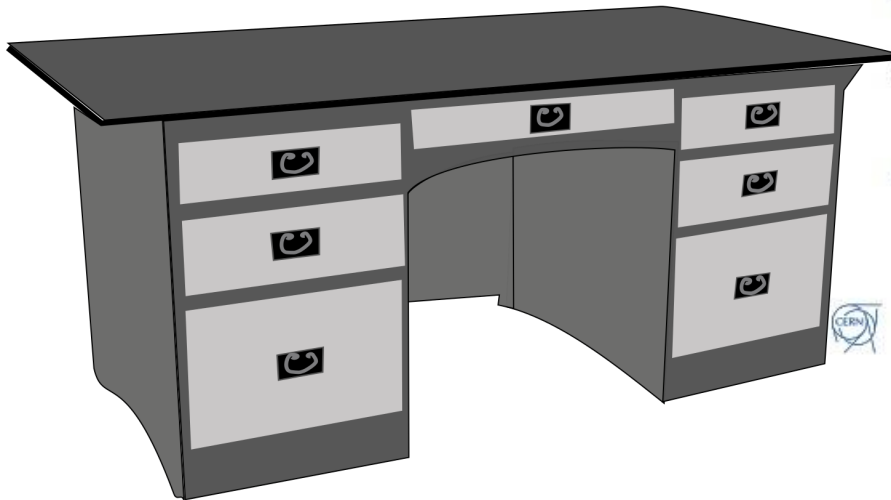
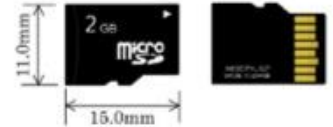


Motivation



Important digression

- a MicroSD card has a volume of $V_{SD} = 15 \times 11 \times 0.8 = 132 \text{ mm}^3$
 - Available with 512 GB or (soon) 1 TB size
- a 3.5" HDD is $V_{HDD} = 101 \times 146 \times 25.4 = 374'548.4 \text{ mm}^3$
- You can pack many microsd cards in the volume of one hard disk. What storage would you have ?
 - $V_{HDD} / V_{SD} = 2837$ cards. Capacity = 1.4 PB or (soon) 2.8 PB.
 - 100 PB would require 35 HDD, which fit in my drawer.
 - 100 PB can already fit my drawer **today** using microsd cards
- Will it be slow ? Unreliable ?
 - With striping and erasure encoding you can expect these new storage devices to be arbitrarily reliable (unbreakable) and arbitrarily fast: Always matching the performance of the external interface (Eg: SATA 6 GB/s)
- Media Cost ?
 - Today 250 - 350 K\$/PB using microsd. 20 - 30 K\$/PB using HDD. 5 - 10 K\$/PB using Tapes.
 - So the only question left is :
 - in 10 years, will flash memory match HDD cost ? Will it match tape cost ?
- Intrinsic advantage
 - No power consumption when idle
 - Significant higher performance and reliability



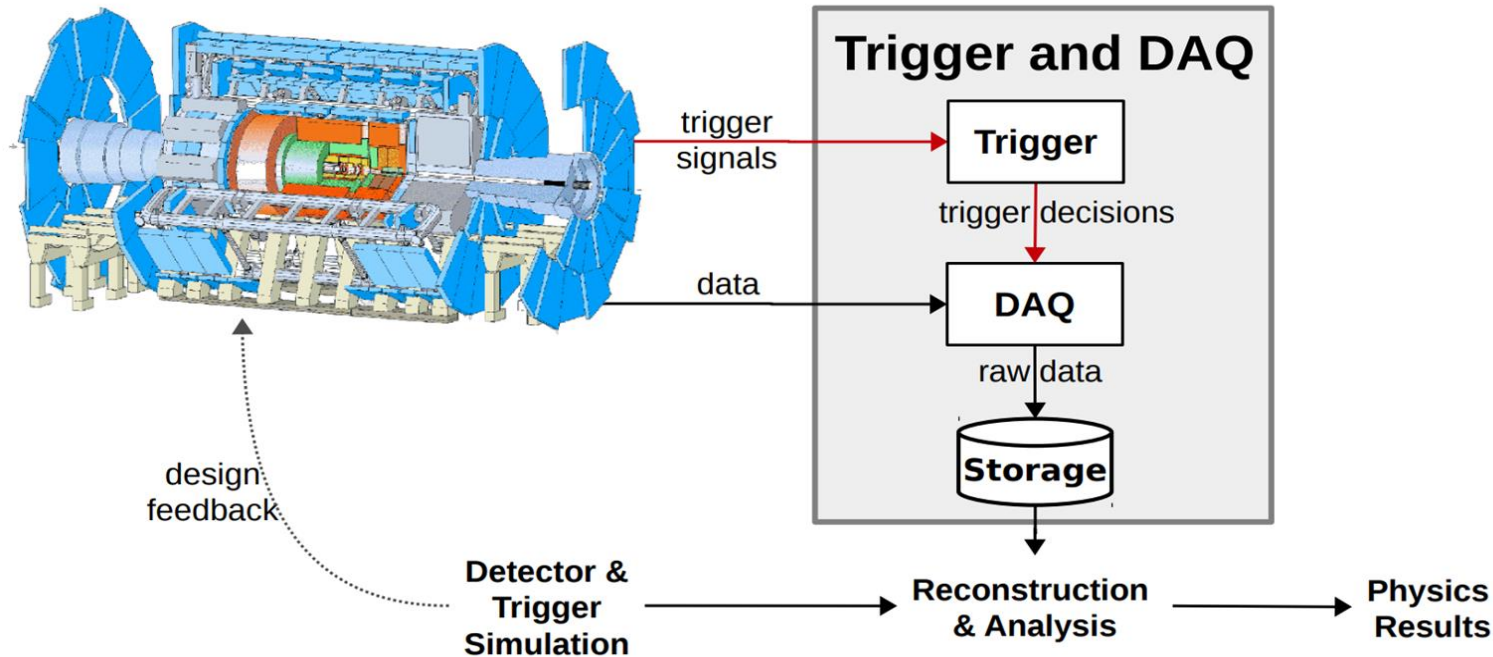
Outline



- *Motivation*
- *Trigger and Data Acquisition system*
- *Available and Emerging technologies*
- *Data AcQuisition DataBase*
- *Integration in the ATLAS TDAQ*
- *Next steps*

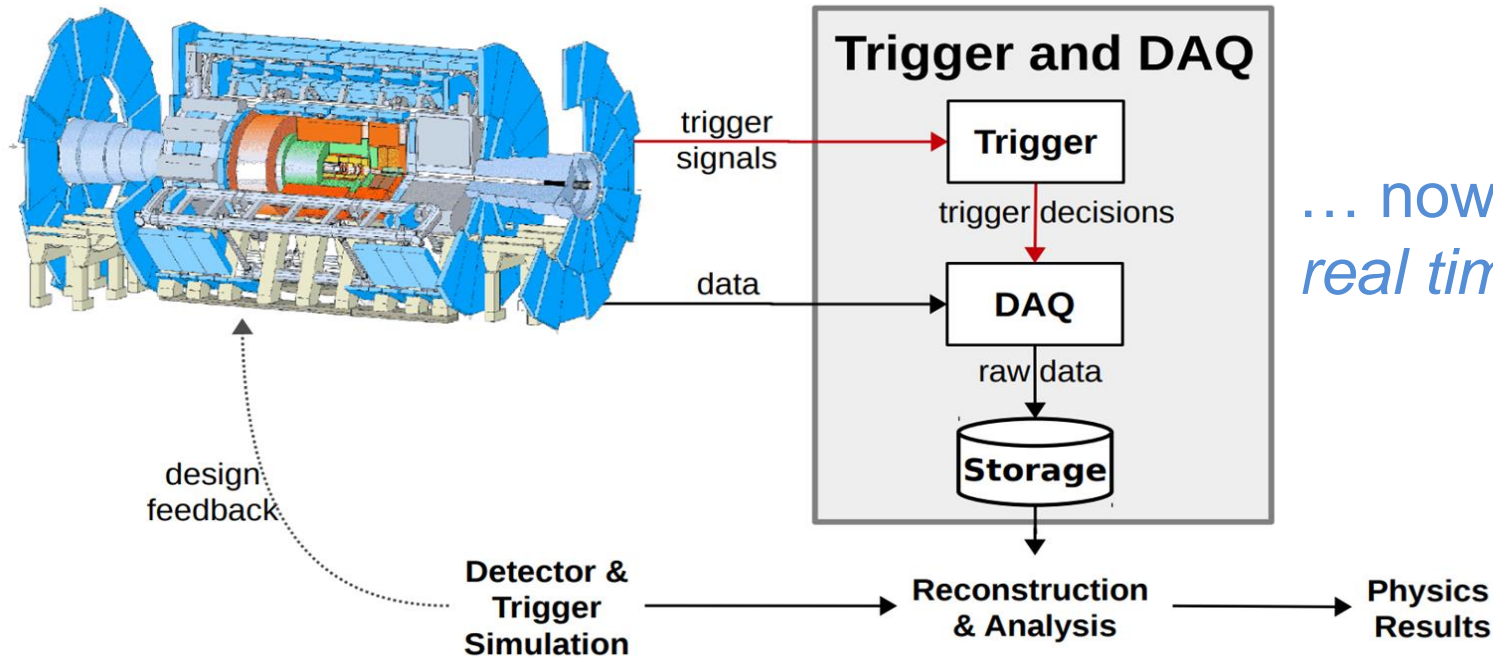
Trigger and Data Acquisition System

TDAQ processes the signals generated in a detector and saves the interesting information on a permanent storage.



Trigger and Data Acquisition System

TDAQ processes the signals generated in a detector and saves the interesting information on a permanent storage.



... nowadays, a
real time system



Our collaboration [1/2]

Develop **a storage system** to decouple *real-time data acquisition* from *asynchronous event selection*.

Temporary storage

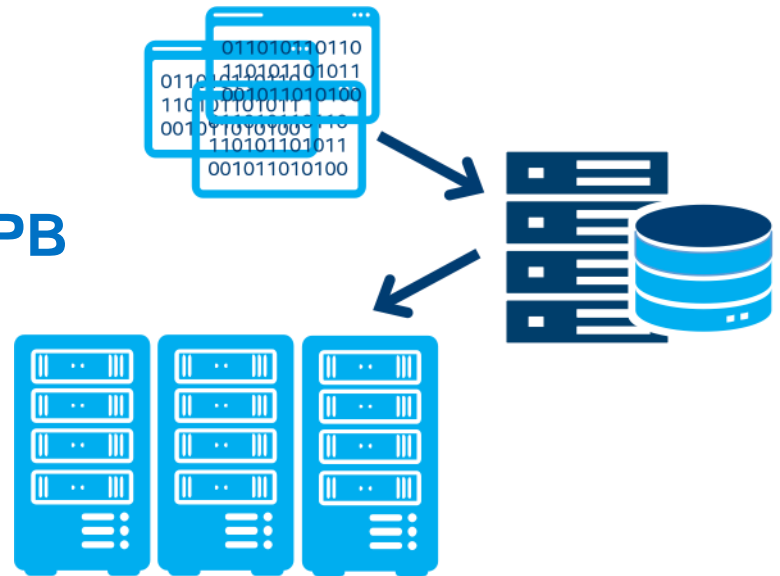
- ✓ Make use of the **inter-fill/no-beam time** for data selection
- ✓ Store **maximum number** of events over short time for offline-like selection
- ✓ **Multiple replica** of the data

Our collaboration [2/2]

Develop **a storage system** to decouple *real-time data acquisition* from *asynchronous event selection*.

Requirements:

- ✓ Distributed over **O(100) nodes**
- ✓ Large, temporary storage of **O(100) PB**
- ✓ Total throughput of **O(10) TB/s**
- ✓ with **O(100000) clients**.





Available solutions

NoSQL data stores, scale by limiting the operations.

Multiple categories: key-value, wide column, document, graph stores.



ORACLE
NOSQL
DATABASE



AEROSPIKE





Available solutions

NoSQL data stores, scale by limiting the operations.

Multiple categories: KEY-VALUE, wide column, document, graph stores.

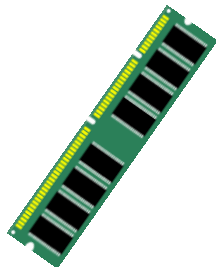
Key	Value
Detector_1	1, 2, 30, 2, 3
Detector_2	0, 0, 1, 1, 3, 87, 6
Detector_3	976.4973, 9785
Detector_4	1.2, 5.6, 78.9

Available solutions

NoSQL data stores, scale by limiting the operations.

Multiple categories: KEY-VALUE, wide column, document, graph stores.

Redis: widespread in memory
key-value store.



DRAM: fast, but volatile, expensive and limited storage size

... like the current readout buffers in DAQ.

Available solutions

NoSQL data stores, scale by limiting the operations.

Multiple categories: **KEY-VALUE**, wide column, document, graph stores.

Redis: widespread in memory
key-value store.



SSD: non volatile, not-limited storage size...

... *slow*

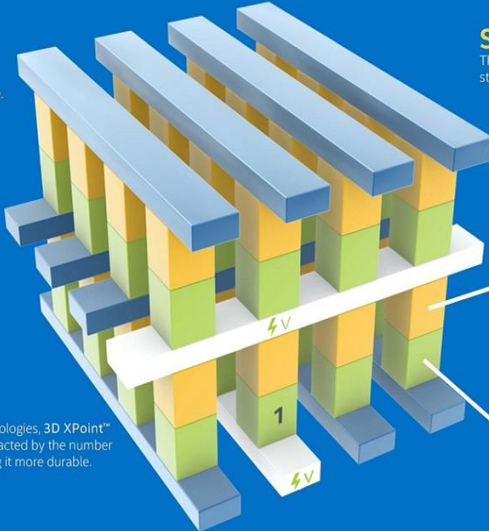


Storage technologies



New technologies
are emerging...

3D XPoint™ Technology: An Innovative, High-Density Design



The diagram illustrates the 3D XPoint technology structure. It shows a grid of vertical columns (submicroscopic) connected by horizontal wires (perpendicular). The columns are stacked vertically, with each layer representing a memory cell. The structure is shown in a perspective view, with columns of varying heights and colors (blue, yellow, green, grey) representing different layers and components. A white horizontal bar with a lightning bolt symbol and the letter 'V' is shown across the columns, representing a selector wire. A small '1' is visible on one of the columns, and another lightning bolt symbol with 'V' is at the base of a column.

Cross Point Structure
Perpendicular wires connect submicroscopic columns. An individual memory cell can be addressed by selecting its top and bottom wire.

Non-Volatile
3D XPoint™ Technology is non-volatile—which means your data doesn't go away when your power goes away—making it a great choice for storage.

High Endurance
Unlike other storage memory technologies, 3D XPoint™ Technology is not significantly impacted by the number of write cycles it can endure, making it more durable.

Stackable
These thin layers of memory can be stacked to further boost density.

Selector
Whereas DRAM requires a transistor at each memory cell—making it big and expensive—the amount of voltage sent to each 3D XPoint™ Technology selector enables its memory cell to be written to or read without requiring a transistor.

Memory Cell
Each memory cell can store a single bit of data.

Storage technologies



New technologies
are emerging...

...from houses



Storage technologies

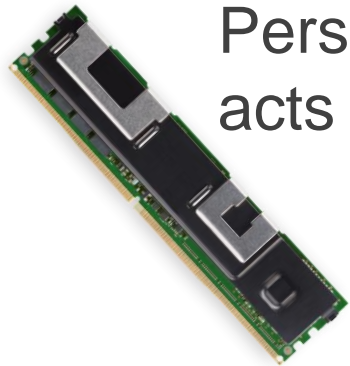


New technologies
are emerging...

...from houses to
skyscrapers!

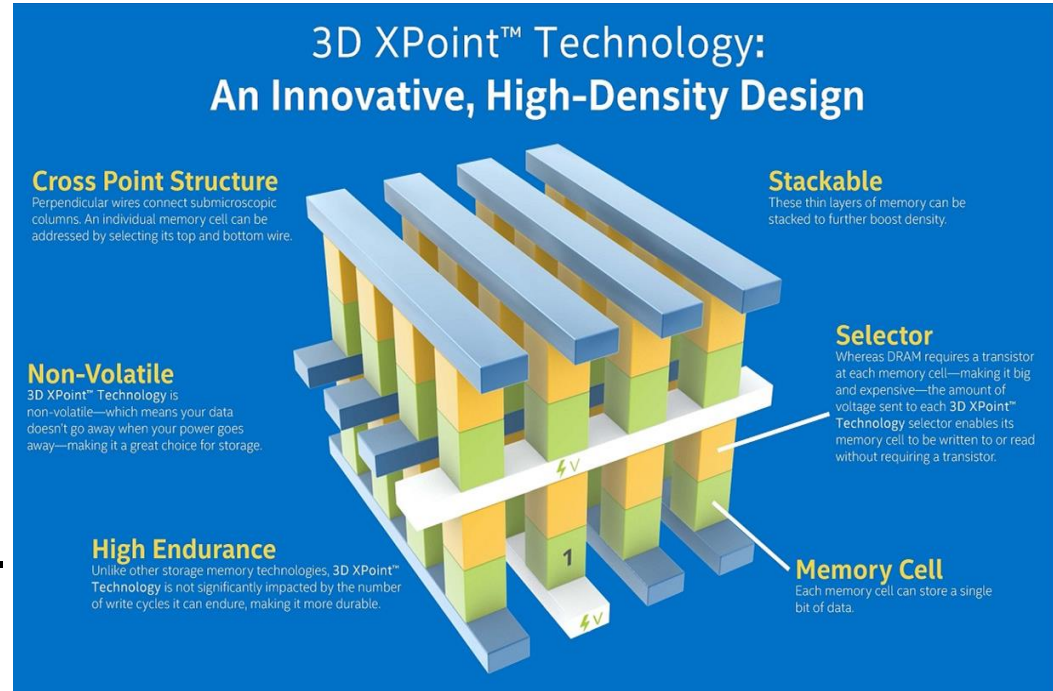


Storage technologies



Persistent memory acts as DRAM.

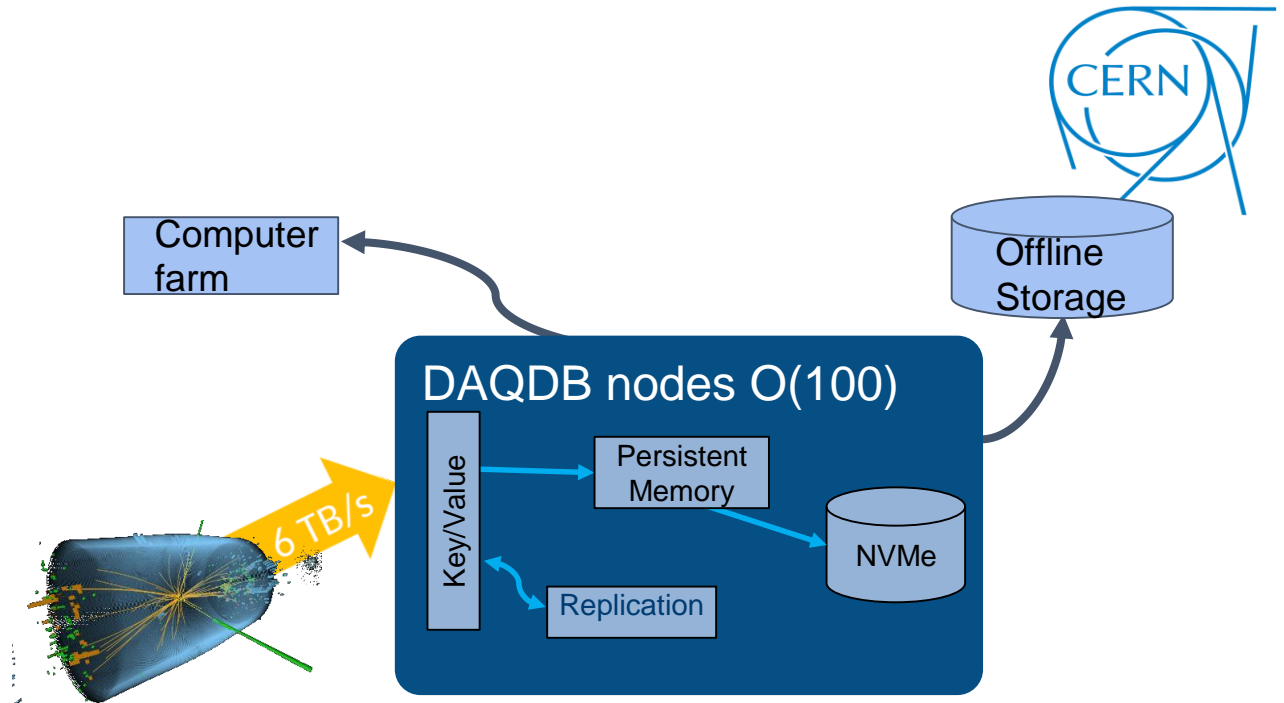
Software:
Persistent Memory Development Kit (PMDK).
Optimal performance of persistent memory.



DAQDB

Key-value store:

- *Low Latency.*
- *Scalable distributed.*
- *Support range queries.*



Technologies:

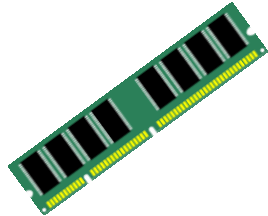
- Storage: Persistent memory and NVMe (ssd) devices.
- Software: Persistent Memory Development Kit, *PMDK*, and Storage Performance Development Kit, *SPDK*.
- Connectivity: *eRPC* a general purpose remote procedure call.

DAQDB –storage

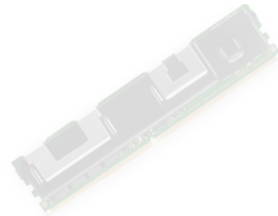


DAQDB

Memory



Persistent
Memory



SSD (sata)



SSD (NVMe)



Keys, Metadata

Values

Capacity & Cost

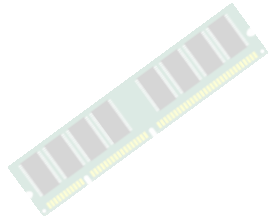
Capable to store only seconds of DAQ system traffic

DAQDB –storage

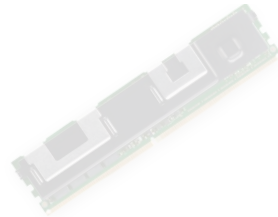


DAQDB

Memory



Persistent
Memory



SSD (sata)



Keys, Metadata

Values

SSD (NVMe)



Keys, Metadata

Values

Performance

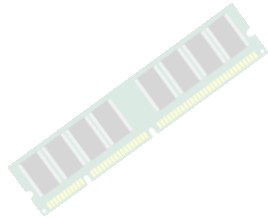
DAQ system requires higher bandwidth

DAQDB –storage

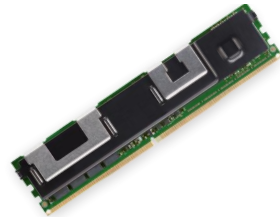


DAQDB

Memory



Persistent
Memory



Keys, Metadata

Values

SSD (sata)



SSD (NVMe)



Capacity

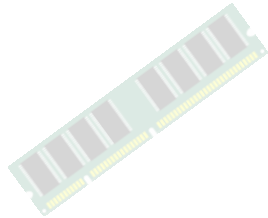
Capable to store minutes of DAQ system traffic

DAQDB –storage

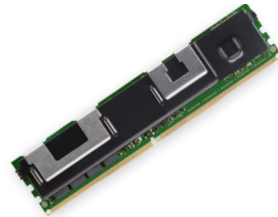


DAQDB

Memory



Persistent
Memory



Keys, Metadata

Values

SSD (sata)



SSD (NVMe)



Values

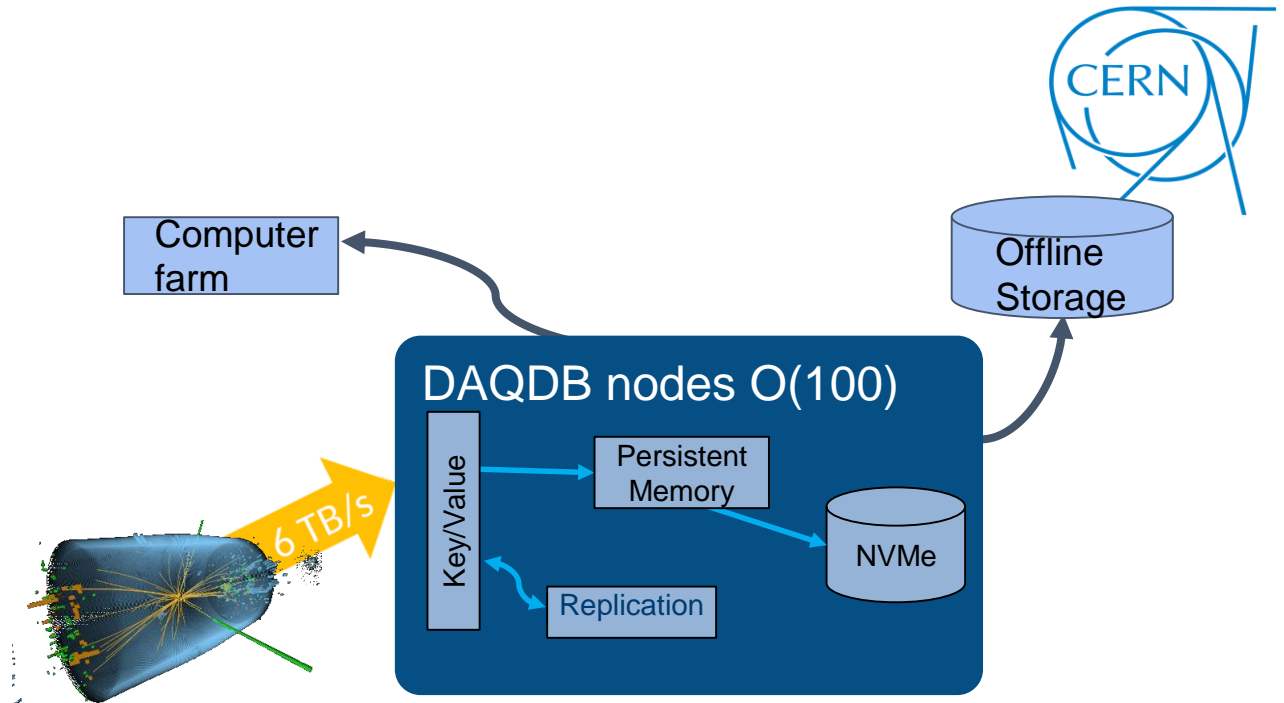
Offload

Pre-filtered values stored on NVMe

DAQDB

Key-value store:

- *Low Latency.*
- *Scalable distributed.*
- *Support range queries.*



Technologies:

- Storage: Persistent memory and NVMe (ssd) devices.
- Software: Persistent Memory Development Kit, *PMDK*, and Storage Performance Development Kit, *SPDK*.
- Connectivity: *eRPC* a general purpose remote procedure call.

DAQDB



Key-value store:

- *Insert data from each fragment with a composite key: (run_id, event_id, subdetector_id)*
- *Potentially stored for several hours/days with replication*
- *Distributed storage might be local or remote.*

Data selection:

- *Query data when needed*
- *Internal event building with support of range queries*



DAQDB - API

User-defined key structure

- `struct example_key{ uint64_t eventID; uint16_t subdetectorID, uint16_t runID; }`

Range Queries

- `Kvs->getRange(keyMin, KeyMax)`

Asynchronous mode

- `Kvs->getRangeAsync(keyMin, KeyMax)`

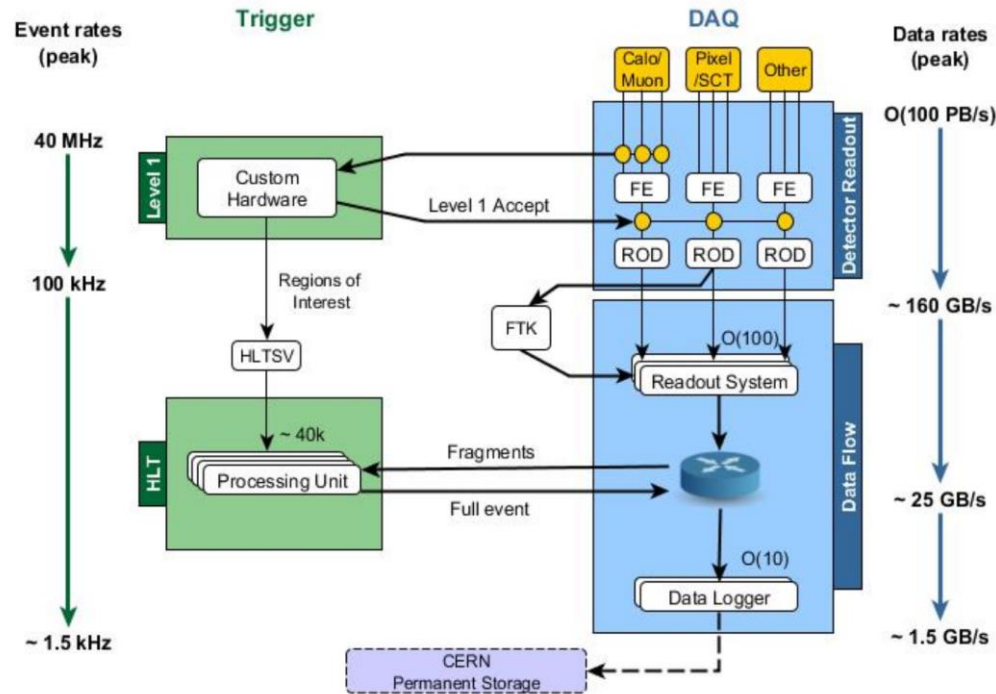
Distributed locking for next event retrieval:

- `Kvs->getAny(options)`

Integration in the system



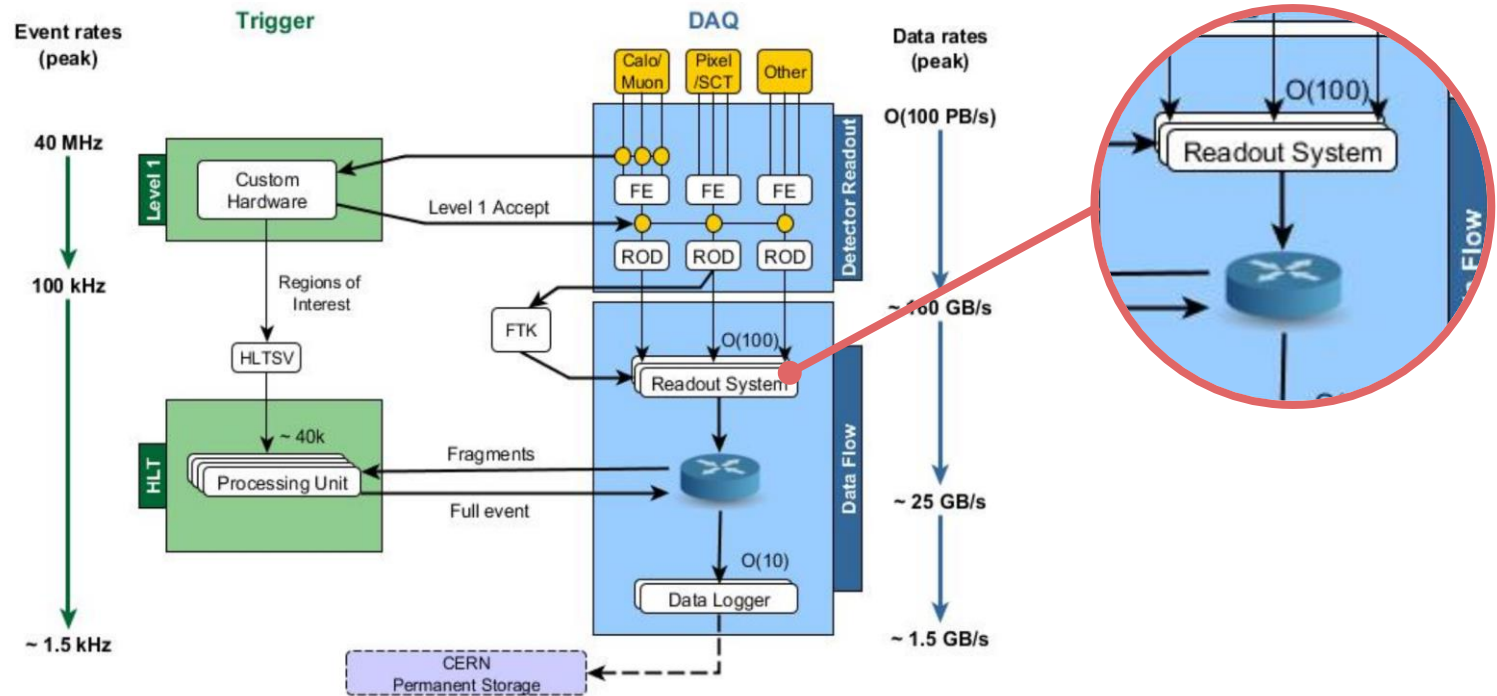
Atlas TDAQ



Integration in the system

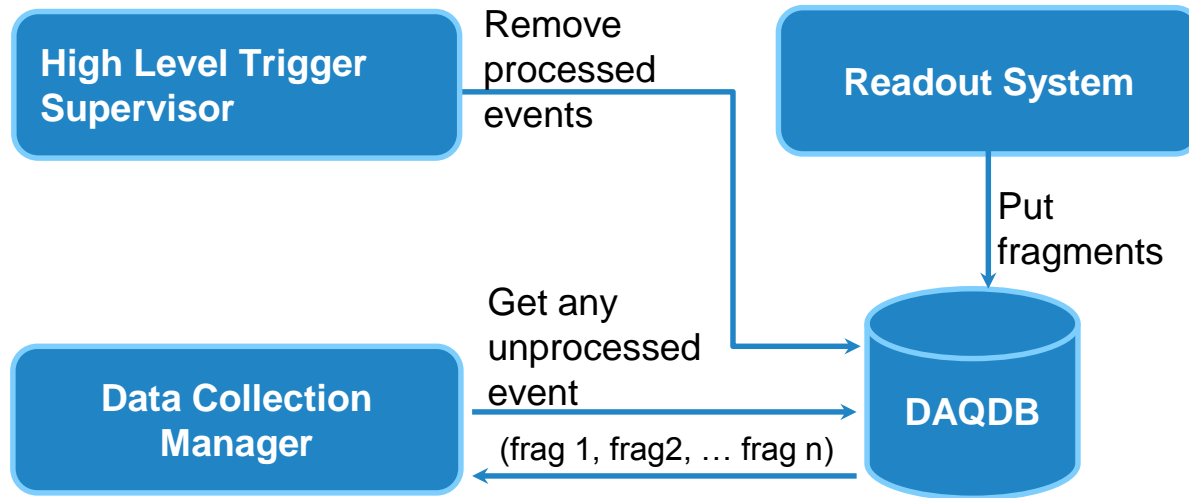


Atlas TDAQ



Integration in the system

Atlas TDAQ

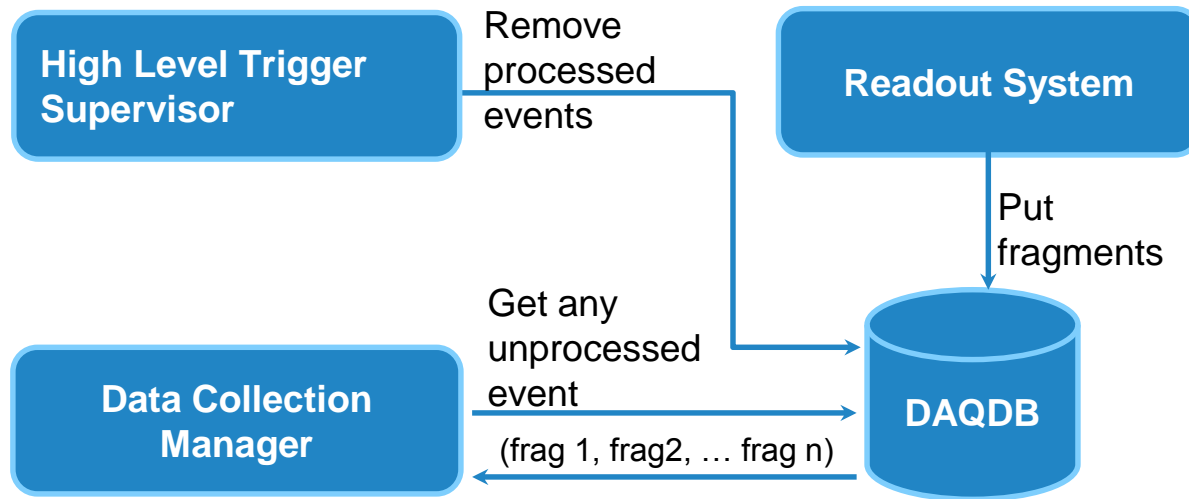


LOCAL DAQDB:

- + fewer servers
- + reduced network load
- both workloads on the same machine

Integration in the system

Atlas TDAQ



Remote DAQDB:

- + optimally distribute the request across the available servers
- more servers
- increase in the network load

Year 1



Requirements and dataflow
understanding

Standalone
benchmarking

Prototype DAQDB
library

D. Cicalese, G. Jereczek, F. Le Goff , G. Lehmann Miotto, J. Love, M. Maciejewski, R. K Mommsen , J. Radtke , J. Schmiegel and M. Szychowska in *The design of a distributed key-value store for petascale hot storage in data acquisition systems*, CHEP 2018.

Future plans



Finish integration
in Atlas

Benchmarking of
DAQDB
in the systems

Integration and
installation in CMS



DAQDB: a Key-Value store for Data Acquisition Systems

Danilo Cicalese
on behalf of the **DAQDB** team
OpenLab – CERN
January 24, 2019

