ISOTDAQ

International School of Trigger and Data Acquisition

Storage systems for DAQ

Adam Abed Abud (CERN)

ISOTDAQ 2023

13 - 22 June 2023 (Istanbul, Turkey)

Storage Examples in Bytes

4K video stream
(4 MB/s)

kilo 10^3 | mega 10^6 | giga 10^9 | tera 10^{12} | peta 10^{15} | exa 10^{18}

Storage Examples in Bytes

Google global storage
(10-15 EB)

YouTube to storage
(8 GB/s)

YouTube to storage
(240 PB/year)

4K video stream
(4 MB/s)

kilo 10^3 mega 10^6 giga 10^9 tera 10^{12} peta 10^{15} exa 10^{18}

Storage Examples in Bytes

Google global storage
(10-15 EB)

YouTube to storage
(8 GB/s)

YouTube to storage
(240 PB/year)

DUNE to storage
(250 MB/s)

DUNE pre-trigger
(1.5 TB/s)

DUNE to storage
(7.5 PB/year)

4K video stream
(4 MB/s)

kilo 10^3 mega 10^6 giga 10^9 tera 10^{12} peta 10^{15} exa 10^{18}

Storage Examples in Bytes

Google global storage
(10-15 EB)

YouTube to storage
(8 GB/s)

YouTube to storage
(240 PB/year)

ATLAS to storage
(1-5 GB/s)

ATLAS pre-trigger
(60 TB/s)

ATLAS to storage
(40 PB/year)

DUNE to storage
(250 MB/s)

DUNE pre-trigger
(1.5 TB/s)

DUNE to storage
(7.5 PB/year)

4K video stream
(4 MB/s)

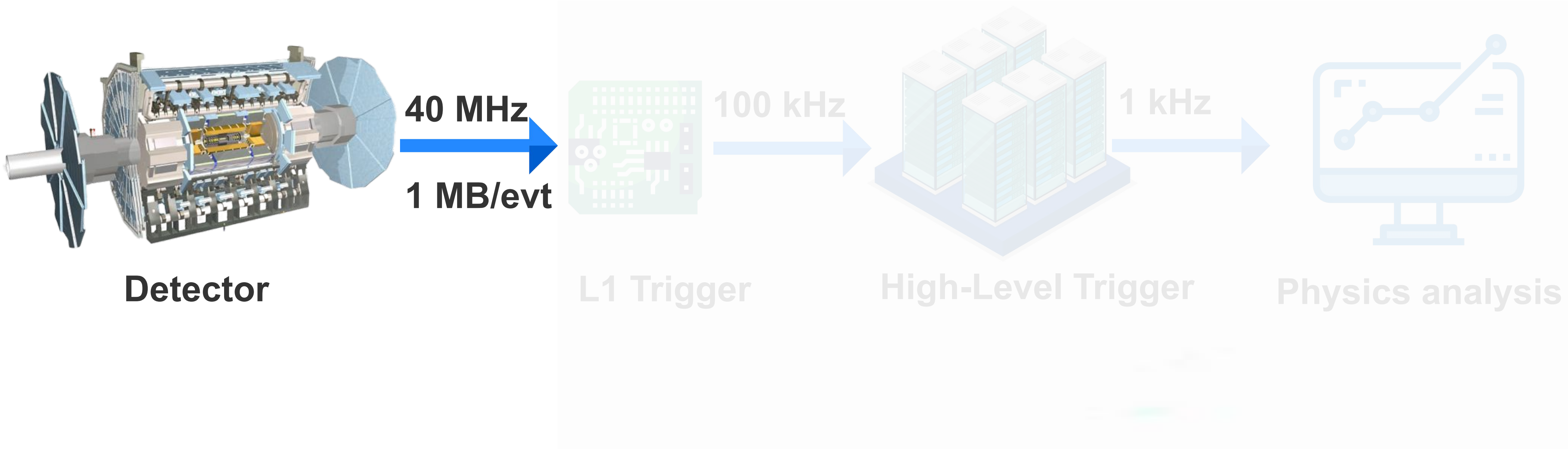
kilo 10^3 — mega 10^6 — giga 10^9 — tera 10^{12} — peta 10^{15} — exa 10^{18}

Outline

- Why are storage systems relevant for DAQ ?
- Storage concepts
- Technology overview
    - HDD, SSD, NVM and DRAM
- Performance benchmarking
    - DD and FIO
- Storage challenges for the future
- Storage system for the DUNE-DAQ
- Conclusion

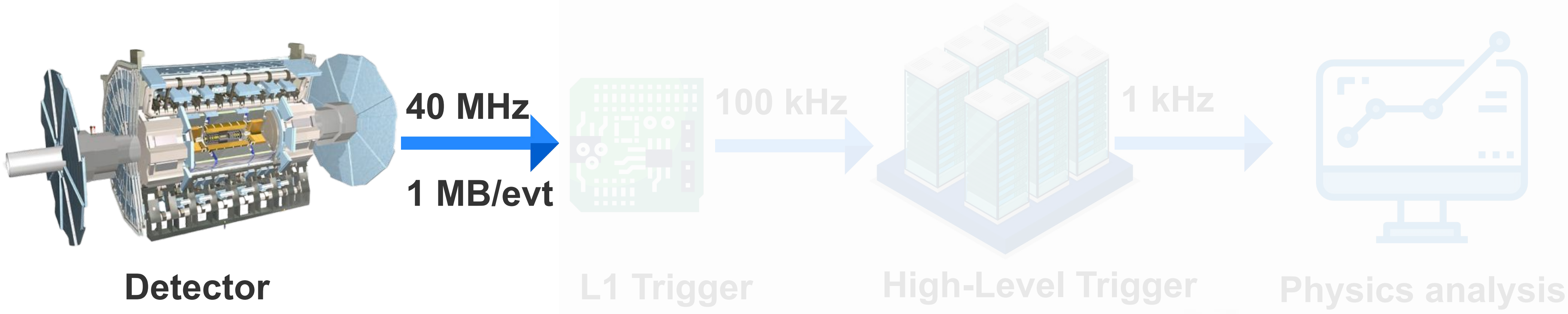
Why are storage systems relevant for DAQ ?

TDAQ pipeline



Why are storage systems relevant for DAQ ?

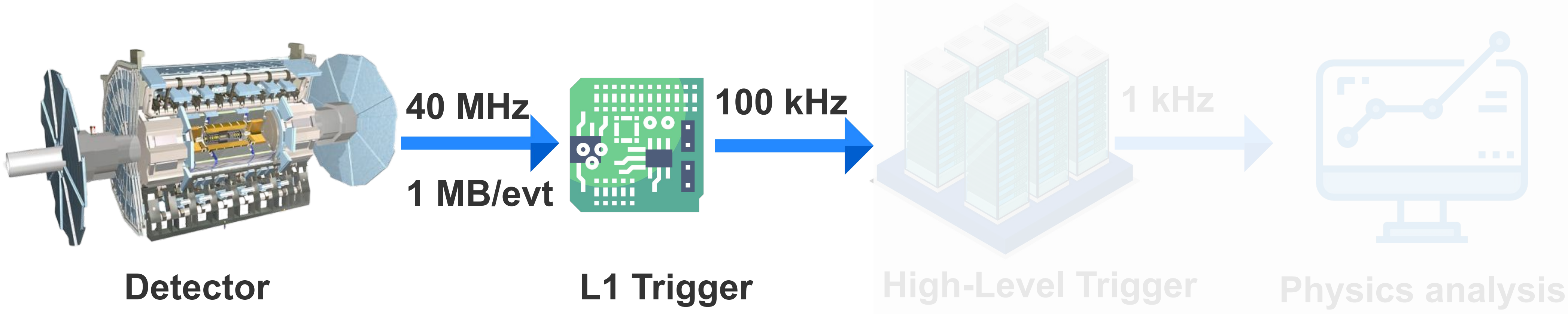
TDAQ pipeline



- Not all the data can be stored:
  - Lack of storage resources
  - Not enough (offline) processing power

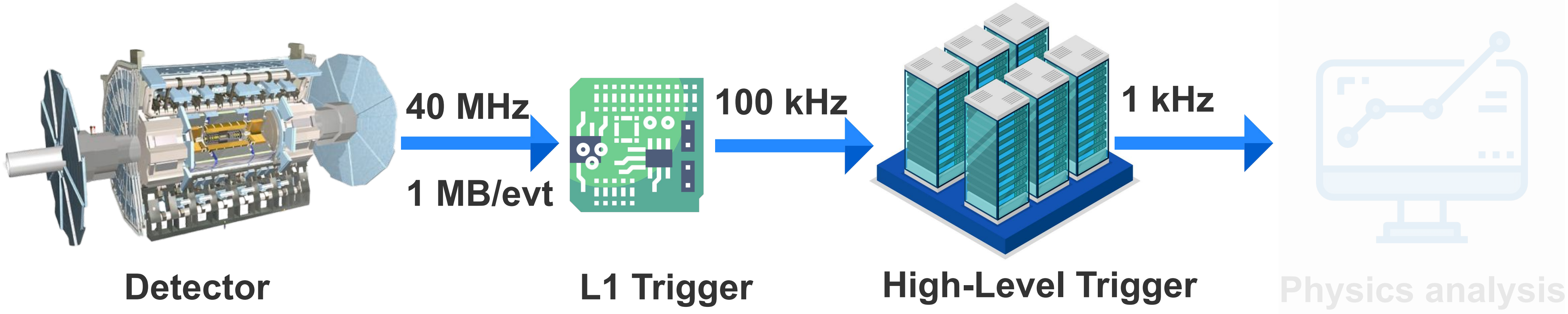
Why are storage systems relevant for DAQ ?

TDAQ pipeline



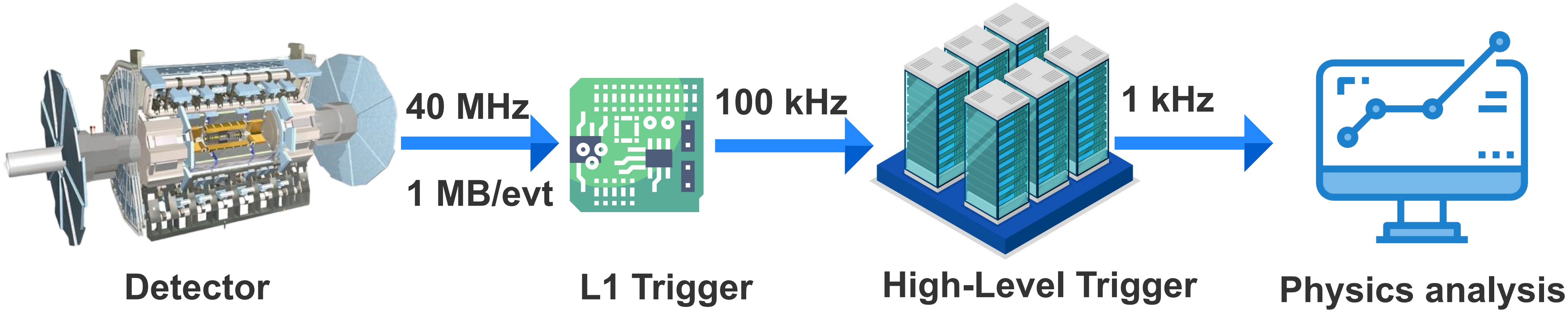
Why are storage systems relevant for DAQ ?

TDAQ pipeline



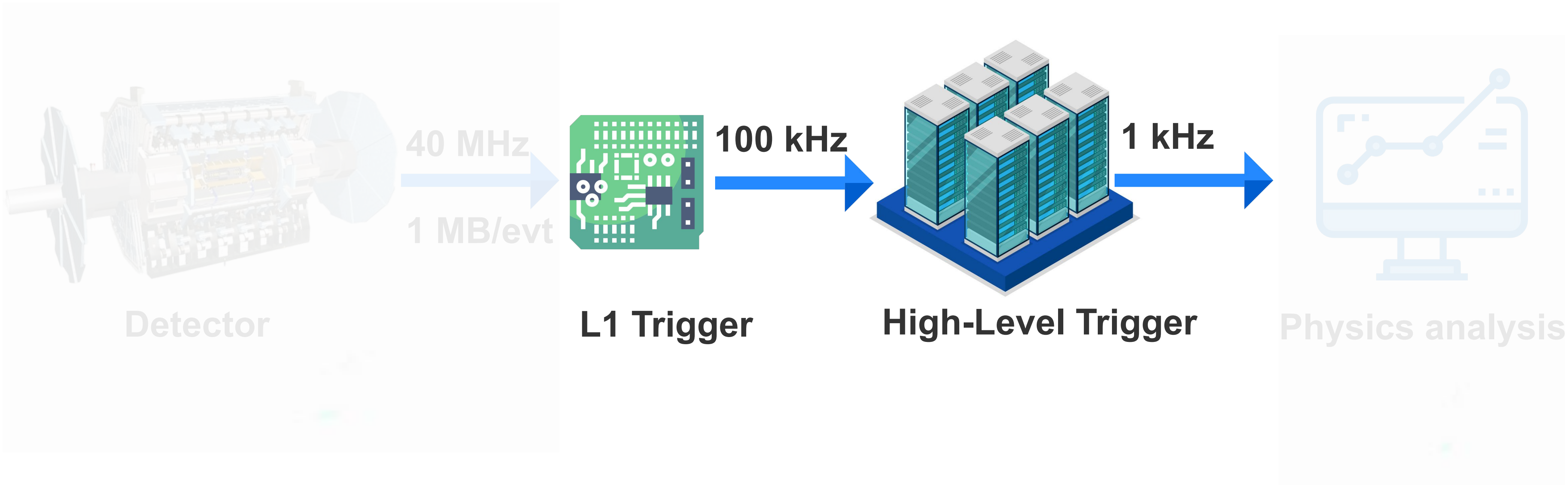
Why are storage systems relevant for DAQ ?

TDAQ pipeline and physics analysis



Why are storage systems relevant for DAQ ?

TDAQ pipeline - Online data taking ("DAQ")



"Safely store data from point A to point B"

DAQ takeaway
Online vs Offline

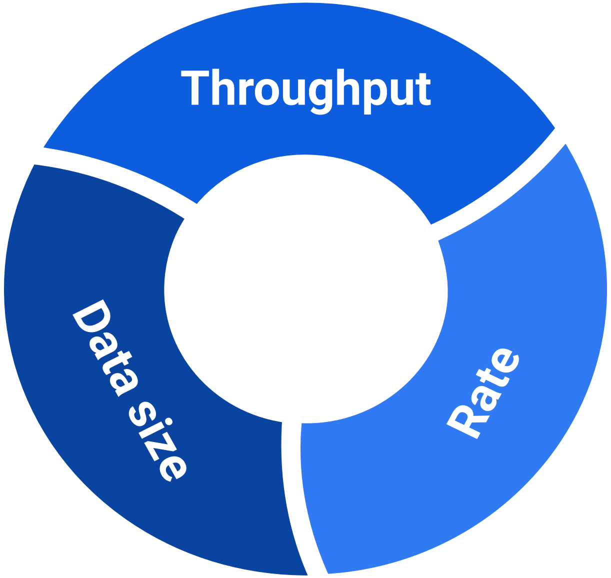
- Storage systems ensure that data is stored and physics results can be produced!
  - Data stored → physics results
- DAQ requirements are different from offline analysis:
  - Storage used to buffer data:
    Absorbs rate fluctuations from the rest of the system
  - Continuous stream of data flow **in and out** the storage system
  - **Throughput** and **latency constraints**
  - Technology choice affected by **total expected data**

DAQ takeaway
Online vs Offline

- Storage systems ensure that data is stored and physics results can be produced!
  - Data stored → physics results
- DAQ requirements are different from offline analysis:
  - Storage used to buffer data:
    Absorbs rate fluctuations from the rest of the system
  - Access pattern: continuous stream of data flow **in and out** the storage system
  - **Throughput** and **latency constraints**
  - Technology choice affected by **total expected data**

DAQ takeaway
Online vs Offline

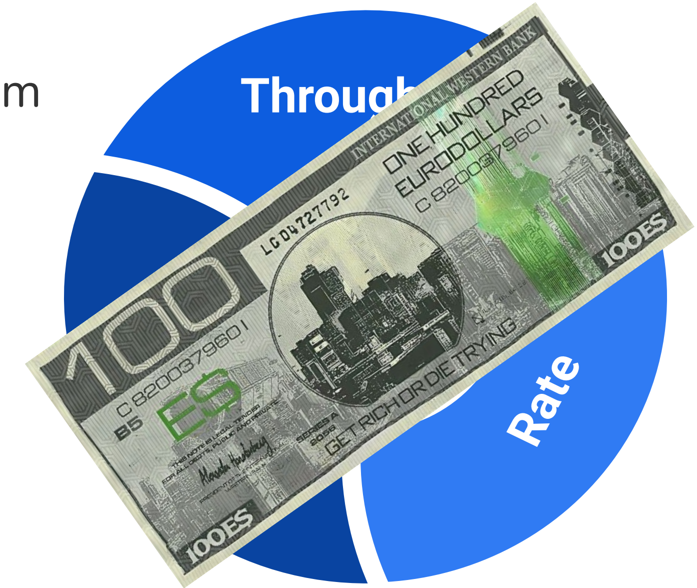
- Storage systems ensure that data is stored and physics results can be produced!
  - Data stored → physics results
- DAQ requirements are different from offline analysis:
  - Storage used to buffer data:
    Absorbs rate fluctuations from the rest of the system
  - Access pattern: continuous stream of data flow
    in and out the storage system
  - **Throughput** and **latency constraints**
  - Technology choice affected by **total expected data**



DAQ takeaway
Online vs Offline

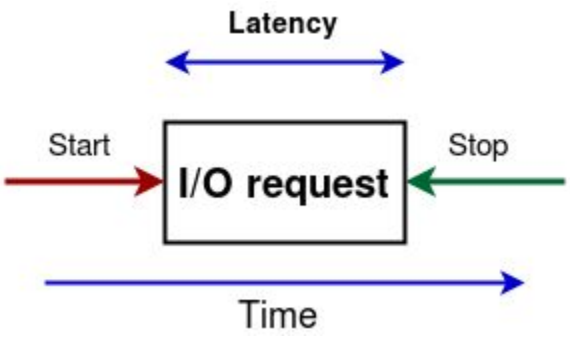
- Storage systems ensure that data is stored and physics results can be produced!
  - Data stored → physics results
- DAQ requirements are different from offline analysis:
  - Storage used to buffer data:
    Absorbs rate fluctuations from the rest of the system
  - Access pattern: continuous stream of data flow
    in and out the storage system
  - **Throughput** and **latency constraints**
  - Technology choice affected by **total expected data**

and cost!



Storage concepts and Technology overview

Storage concepts
Some definitions



- **I/O:** input/output operation
- **Access pattern:** sequential/random read or write
- **Latency**: time taken to respond to an I/O. Usually measured in ms or in µs
- **Rate:** number of I/O per second to a storage location (**IOPS**)
- **Blocksize:** size in bytes of an I/O request
- **Bandwidth**: product of I/O block size and IOPS

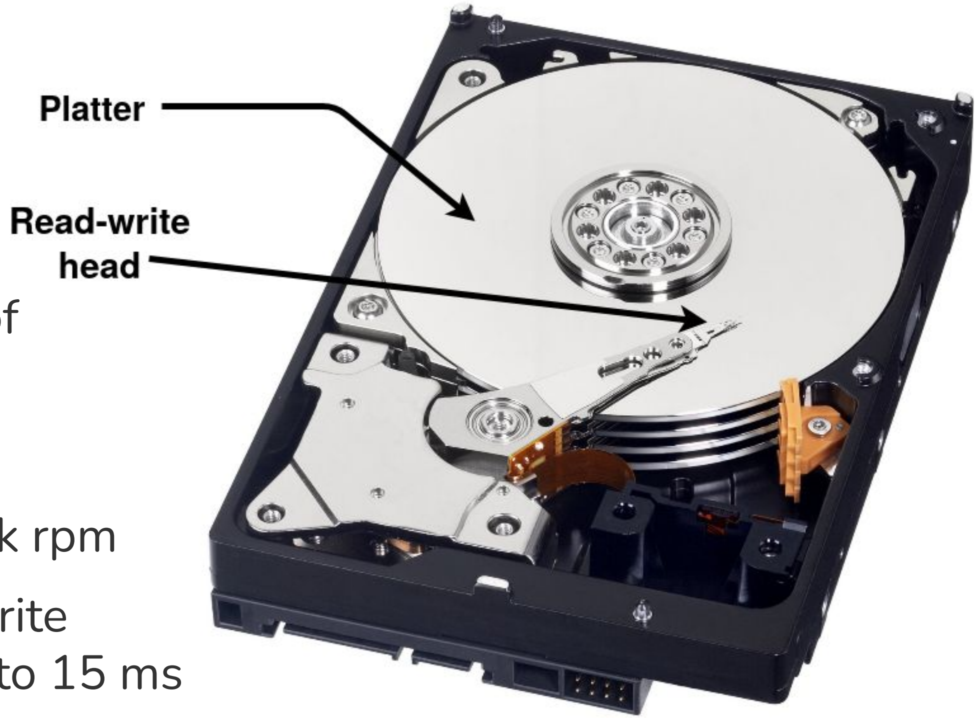
Bandwidth = [I/O block size] x [IOPS]

Hard drives (HDD)

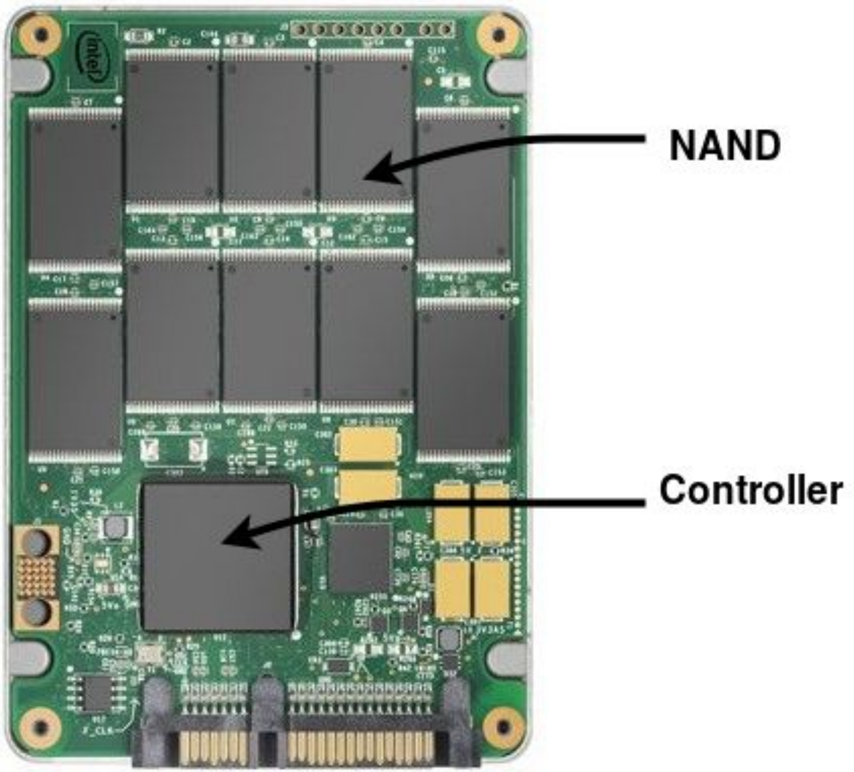
Quick introduction

- Electromechanical device
- Circular rotating platter divided into millions of magnetic components where data is stored
- Typical rotational speed of HDDs:
  - 5400 rpm, **7200 rpm**, 10k rpm and 15k rpm
- **Seek time:** time required to adjust the read-write head on the platter. Typical values: from 3 ms to 15 ms
- **Rotational latency:** time needed by the platter to rotate and position the data under the read-write head

$$IOPS = \frac{1}{\text{Avg. seek} + \text{Avg. latency}}$$



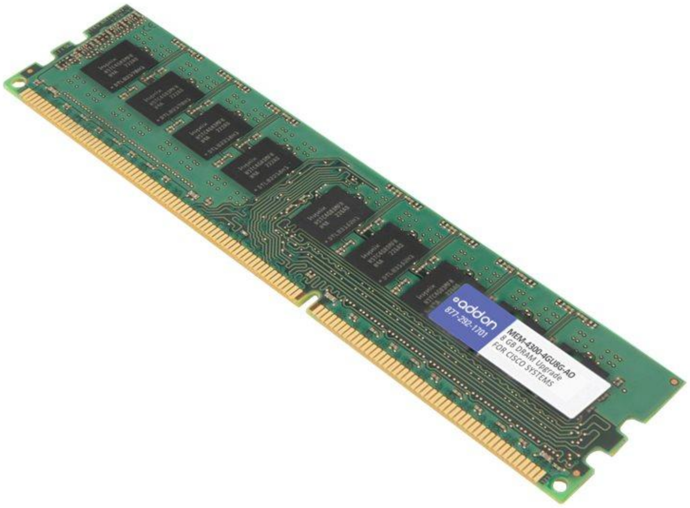
Solid state drives (SSD)
Quick introduction

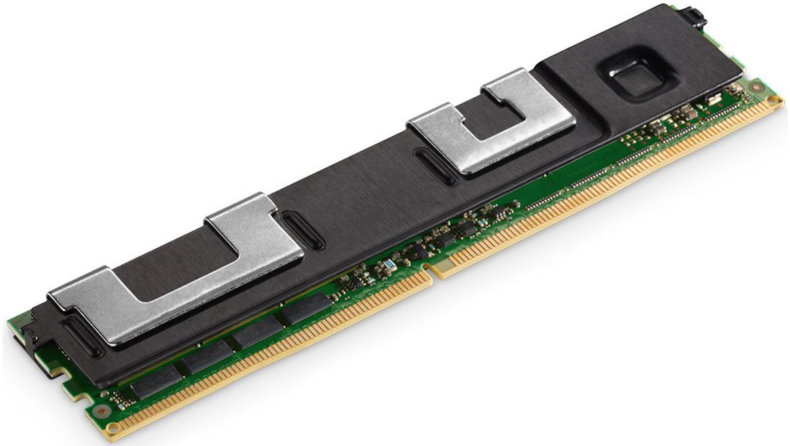


- **Architecture**:
  - NAND flash chipset: store data
  - Controller: caching, load balancing and error handling
- Capacity limited to number of NAND chipsets a manufacturer is able to insert into a device
- (Typically) better performance compared to HDDs
  - There is no mechanical component
  - Reduced latency and seek time
- Optimized controller and communication technology for higher bandwidth devices
  - NVM Express (NVMe) SSD

DRAM and Non-Volatile Memory

Quick introduction

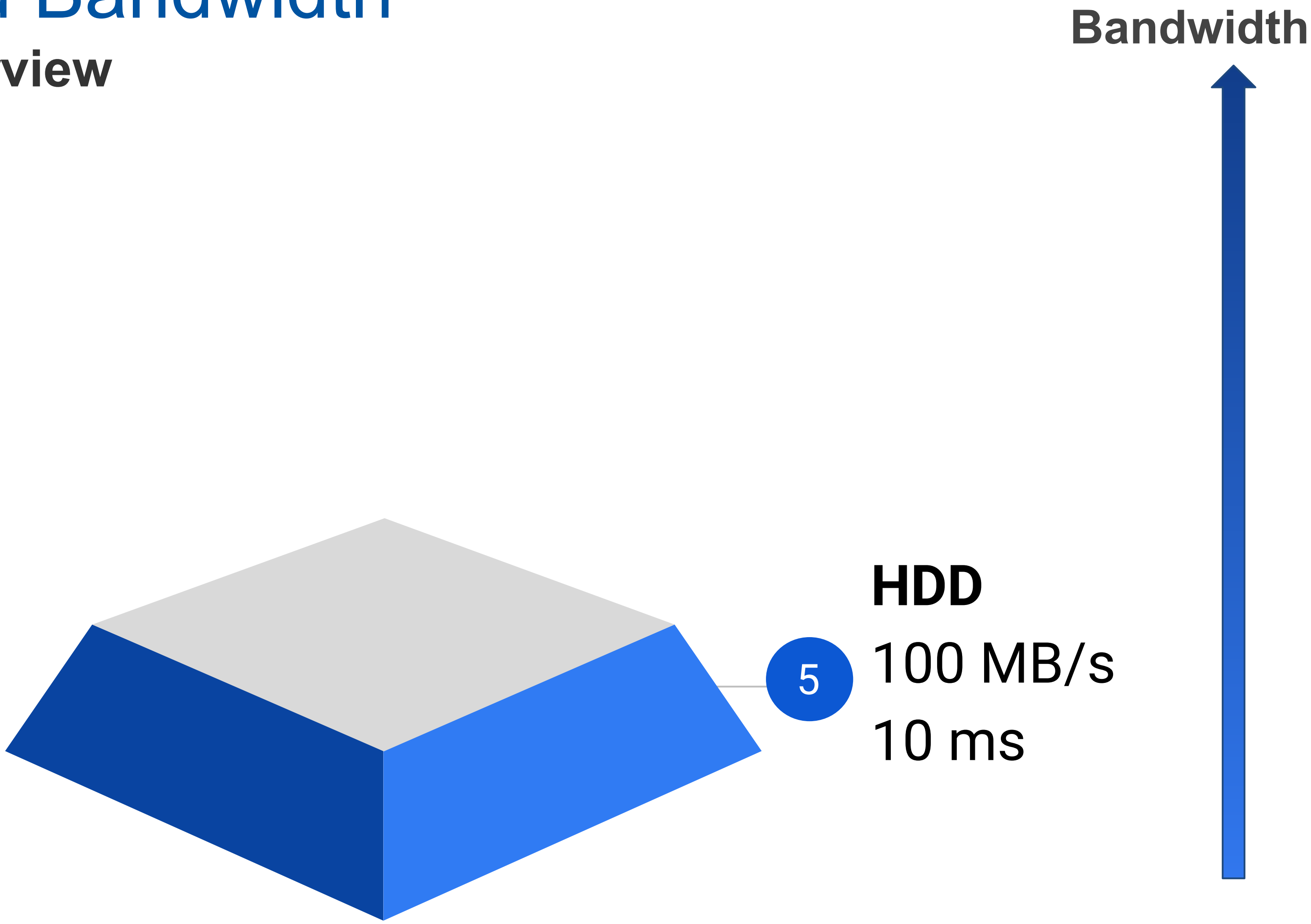




- **DRAM**
    - Semiconductor memory technology
    - Data is not persisted, only temporary storage cells (capacitors and transistors)
    - Low latency (0.1 µs)
- **Non-volatile memory (NVM)**
    - Hold data even if device is turned off
    - Higher storage capacity than DRAM
    - Latency (1 µs)
    - 3D XPoint technology (Intel and Micron, 2015)

Latency and Bandwidth

Technology overview



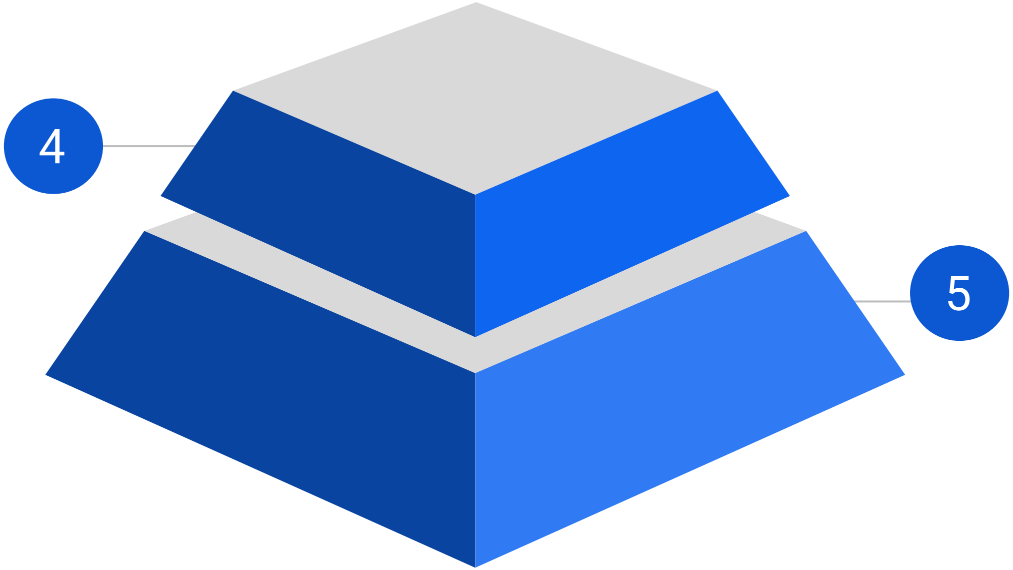
Latency and Bandwidth

Technology overview

SSD

500 MB/s

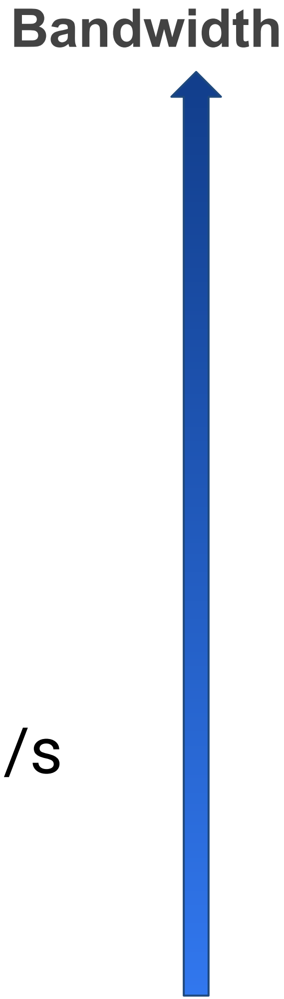
100 µs



HDD

100 MB/s

10 ms



Latency and Bandwidth

Technology overview

Bandwidth

SSD
500 MB/s
100 µs

NVMe SSD
2 GB/s
30 µs

HDD
100 MB/s
10 ms

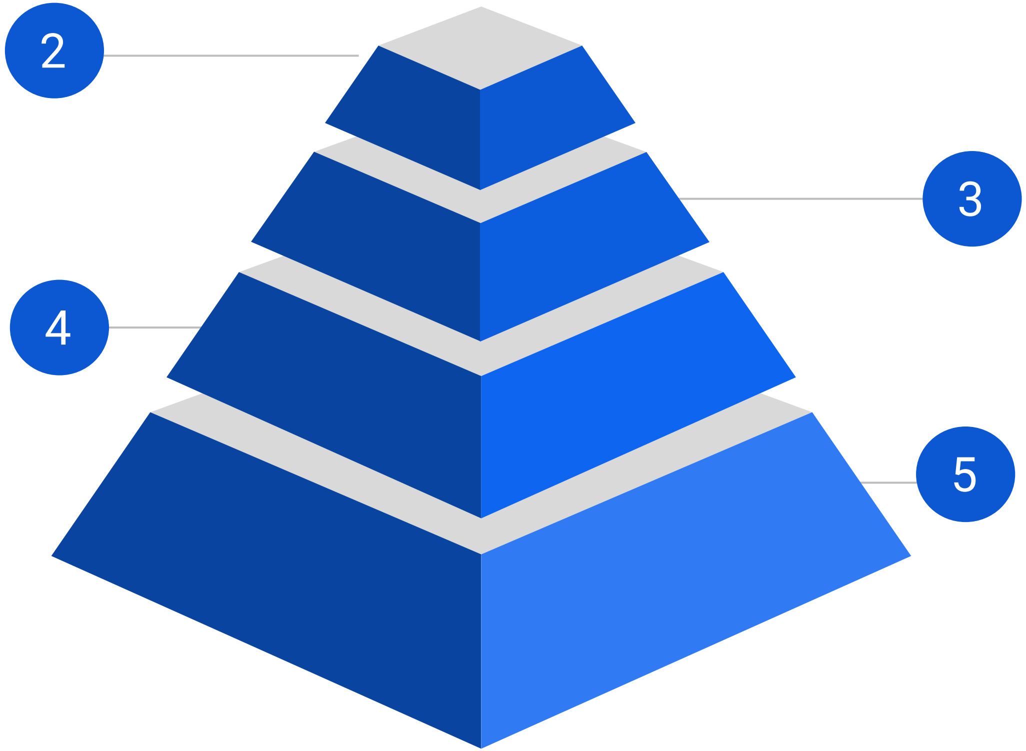
Latency and Bandwidth

Technology overview

Bandwidth

NVM
10 GB/s
1 µs

NVMe SSD
2 GB/s
30 µs

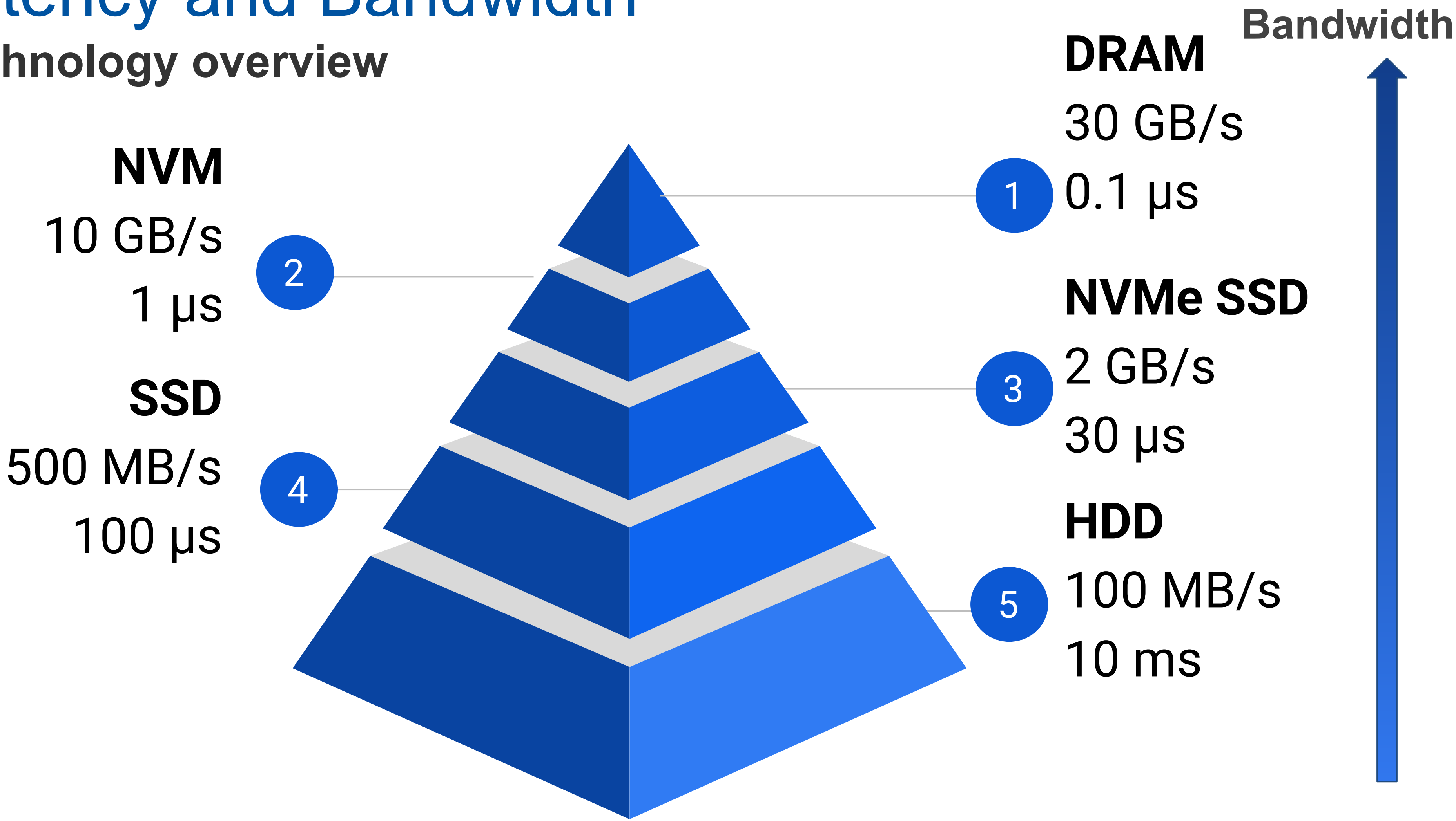


SSD
500 MB/s
100 µs

HDD
100 MB/s
10 ms

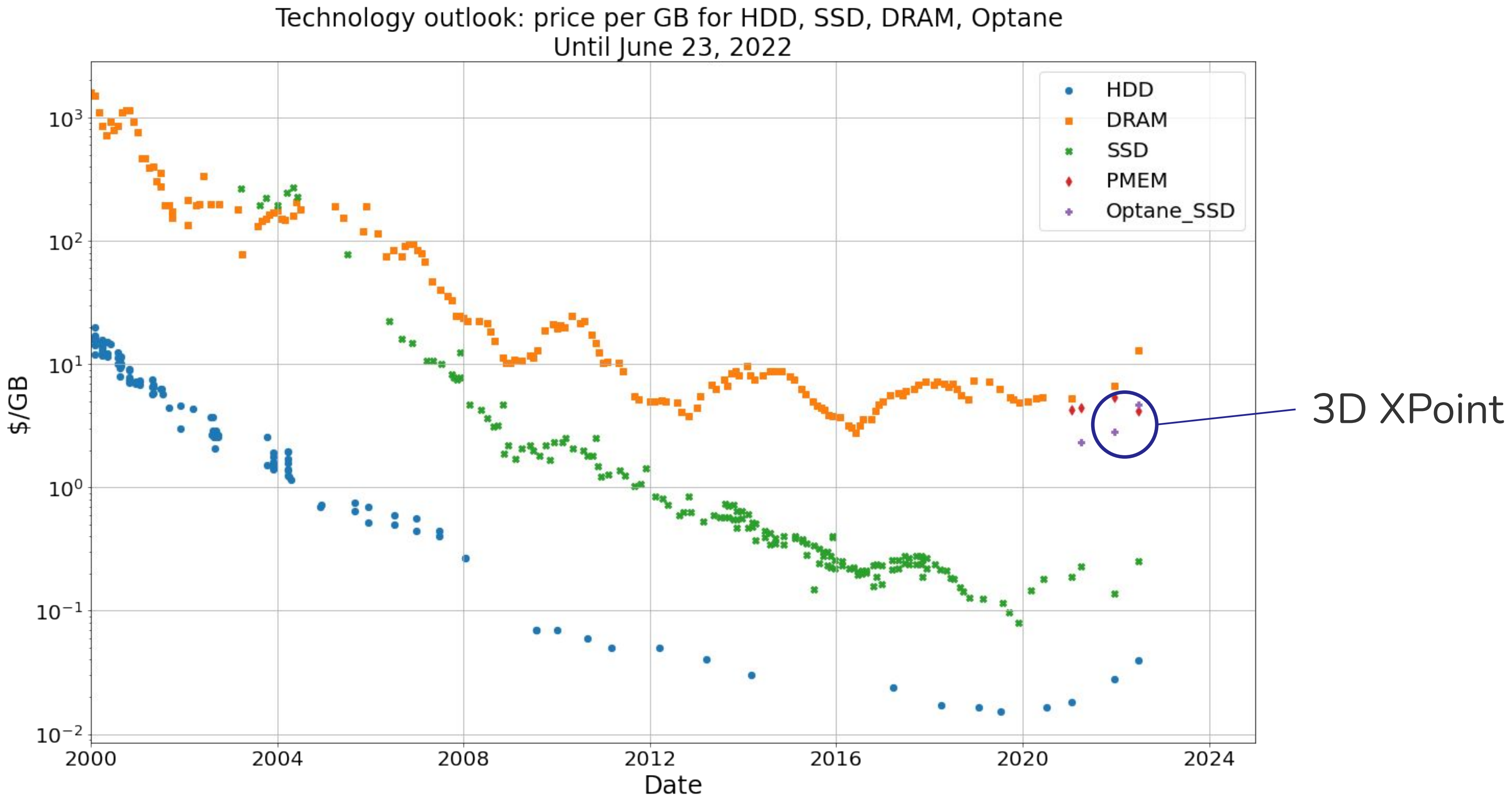
Latency and Bandwidth

Technology overview



Market trend for storage technologies

Price per GB for HDD, SSD, Flash and RAM



Data collected by John C. McCallum.
Data collected by Adam Abed Abud since 2018

Storage benchmarking

DD

- Linux tool to copy data at the block level
- Usage:
    - `dd if=/path/to/input/file of=/path/to/output/file bs=block_size count=amount_blocks`
- Avoid operating system cache by adding **oflag=direct** option

```
[student@storage_lecture]$ dd if=/dev/zero of=deleteme bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 3.67626 s,  285 MB/s
```

Storage benchmarking

Flexible I/O (FIO)

LAB 11: Storage systems

- Advanced tool for characterizing I/O devices
- Usage:
  - **fio --rw**=<opt1> **--bs**==<opt2> **--size**=<opt3> **--filename**=<opt4> **--direct**=<opt5> **--ioengine**=libaio **--name**=isotdaq

```
[student@storage_lecture]$ fio --rw=write --bs=1M --size=1G --filename=deleteme
--direct=0 --ioengine=libaio --name=isotdaq

fio-3.12
Starting 1 process
isotdaq : Laying out IO file (1 file / 1024MiB)
… … …
Run status group 0 (all jobs):

  WRITE: bw=276MiB/s (282MB/s), 276MiB/s-276MiB/s (282MB/s-282MB/s), io=1024MiB
(1074MB), run=4424-4424msec
```

Redundant Array of Inexpensive Disks (**RAID**)

Redundancy and fault tolerance

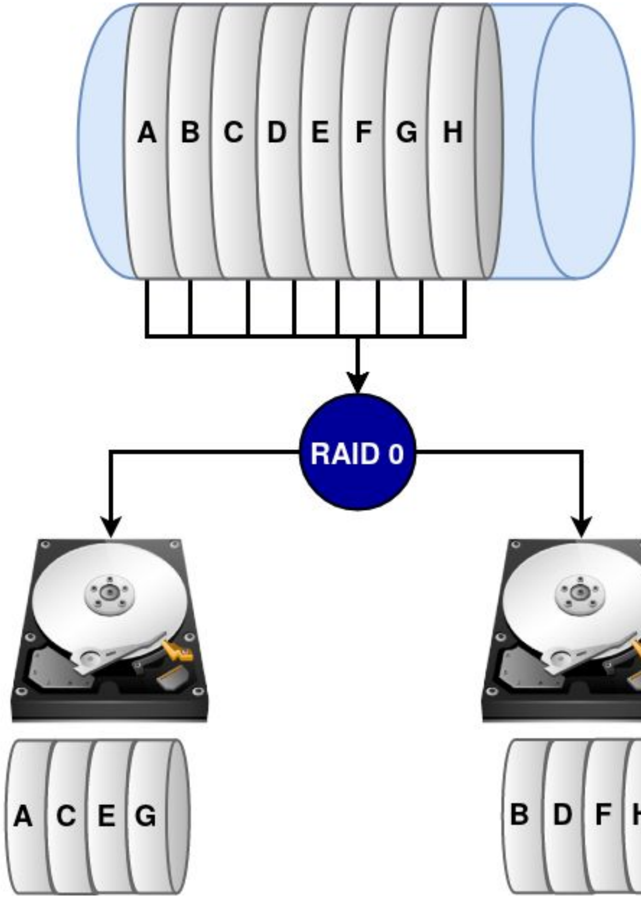
- Multiple physical disk drives are logically grouped into one or more units to increase data performance and/or data redundancy
- Invented in 1987 by researchers from the University of California
- Most common RAID types: RAID 0, RAID 1, RAID 5, RAID 10
- **Fault tolerance** guaranteed by using **parity** as an error protection scheme
    - Based on the XOR logic operation
    - For series of XOR operations, count the number of occurrences of 1:
        - If result is even then bit parity is 0
        - If result is odd then bit parity is 1

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Redundant Array of Inexpensive Disks (**RAID**)

RAID 0 - Striping

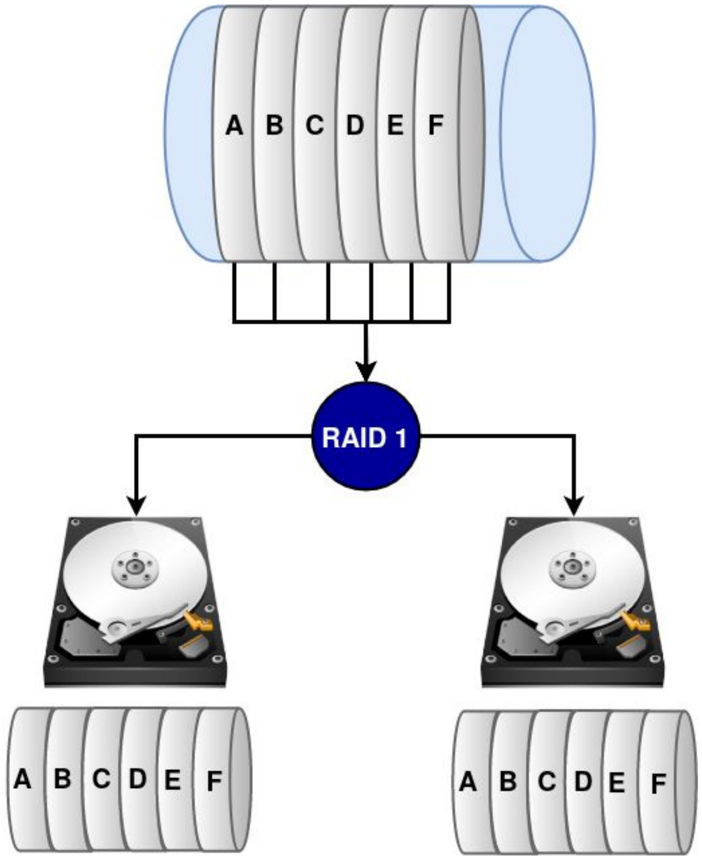
- Data divided in blocks and <u>striped</u> across multiple disks
- **Not fault tolerant** because data is not duplicated
- Speed advantage
  - Two disk controllers allow to access data much faster



Redundant Array of Inexpensive Disks (**RAID**)

RAID 1 - Mirroring and Duplexing

- Data divided in blocks and copied across multiple disks
- **Fault tolerant** because of data mirroring
  - Each disk has the same data
- **Disadvantage**: usable capacity is half of the total



Redundant Array of Inexpensive Disks (**RAID**)

Redundancy and fault tolerance

- Multiple physical disk drives are logically grouped into one or more units to increase data performance and/or data redundancy
- Invented in 1987 by researchers from the University of California
- Most common RAID types: RAID 0, RAID 1, RAID 5, RAID 10
- **Fault tolerance** guaranteed by using **parity** as an error protection scheme
    - Based on the XOR logic operation
    - For series of XOR operations, count the number of occurrences of 1:
        - If result is <u>even</u> then bit parity is 0
        - If result is <u>odd</u> then bit parity is 1

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

A crash course on bit parity

Example for a "3-bit" hard drive

Disk 1	Disk 2	Disk 3	Count	Parity
0	1	1		
1	0	0		
1	1	0		

A crash course on bit parity

Example for a “3-bit” hard drive

Disk 1	Disk 2	Disk 3	Count	Parity
0	1	1	2	0
1	0	0	1	1
1	1	0	2	0

A crash course on bit parity

Disk failure

Disk 1	Disk 2	Disk 3	Count	Parity
0	1	1	2	0
1	0	0	1	1
1	1	0	2	0

A crash course on bit parity

Example for a "3-bit" hard drive

Disk 1	Disk 2	Parity	Count	Disk 3
0	1	0		
1	0	1		
1	1	0		

A crash course on bit parity

Example for a “3-bit” hard drive

Disk 1	Disk 2	Parity	Count	Disk 3
0	1	0	1	
1	0	1	2	
1	1	0	2	

A crash course on bit parity

Example for a “3-bit” hard drive

Disk 1	Disk 2	Parity	Count	Disk 3
0	1	0	1	1
1	0	1	2	0
1	1	0	2	0

A crash course on bit parity

Example for a “3-bit” hard drive

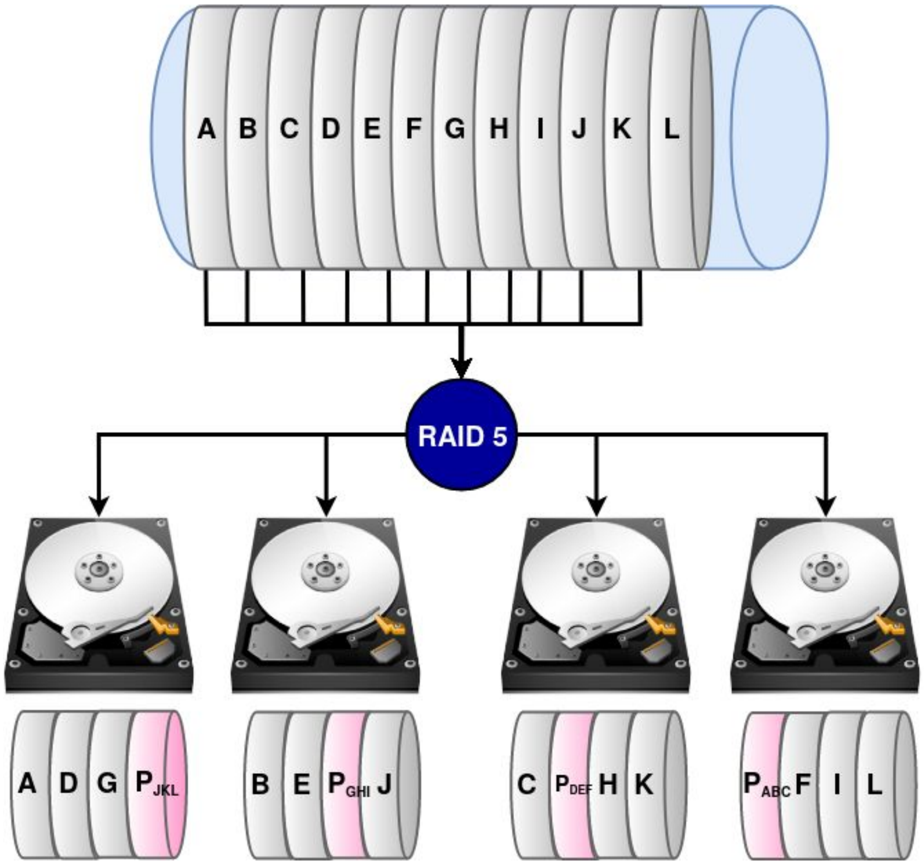
Disk 3
1
0
0

Disk 1	Disk 2	Parity	Count	Disk 3
0	1	0	1	1
1	0	1	2	0
1	1	0	2	0

Redundant Array of Inexpensive Disks (**RAID**)

RAID 5 - Striping with parity

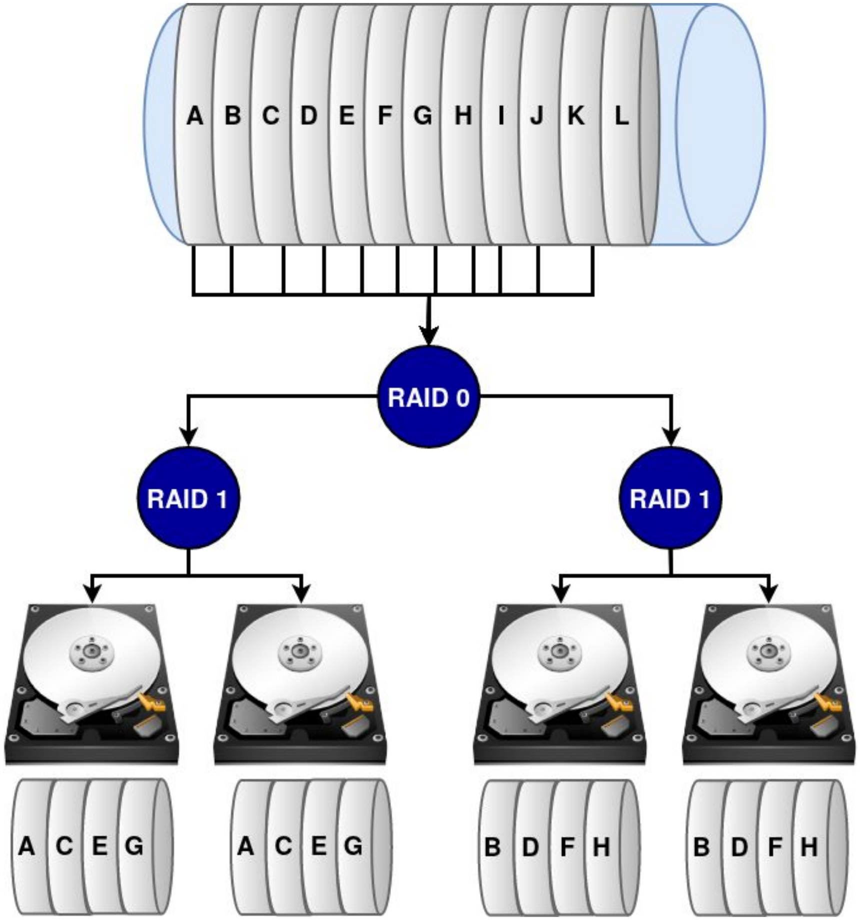
- Requires 3 or more disks
- Data is not duplicated but **striped** across multiple disks
- Fault tolerant because **parity** is also striped with the data blocks
- Larger capacity provided compared to RAID 1
- Disadvantage: an entire disk is used to store parity



Redundant Array of Inexpensive Disks (**RAID**)

RAID 10 = RAID 1 + RAID 0

- Requires a minimum of 4 disks
- Data is **striped** (RAID 0)
- Data is duplicated across multiple disks (RAID 1)
- **Advantage**: fault tolerance and higher speed
- **Disadvantage**: only half of the available capacity is usable



Redundant Array of Inexpensive Disks (**RAID**)

HW, SW

- **Hardware** implementation:
    - Use of RAID controllers
    - Manage system independently of OS
    - Offload I/O operation and parity computation
    - Cost usually high
- **Software** implementation:
    - OS used to manage RAID configuration
    - Impact on CPU usage can be high
- **Disadvantage**: scaling to multiple servers is not possible

Redundant Array of Inexpensive Disks (**RAID**)

HW, SW

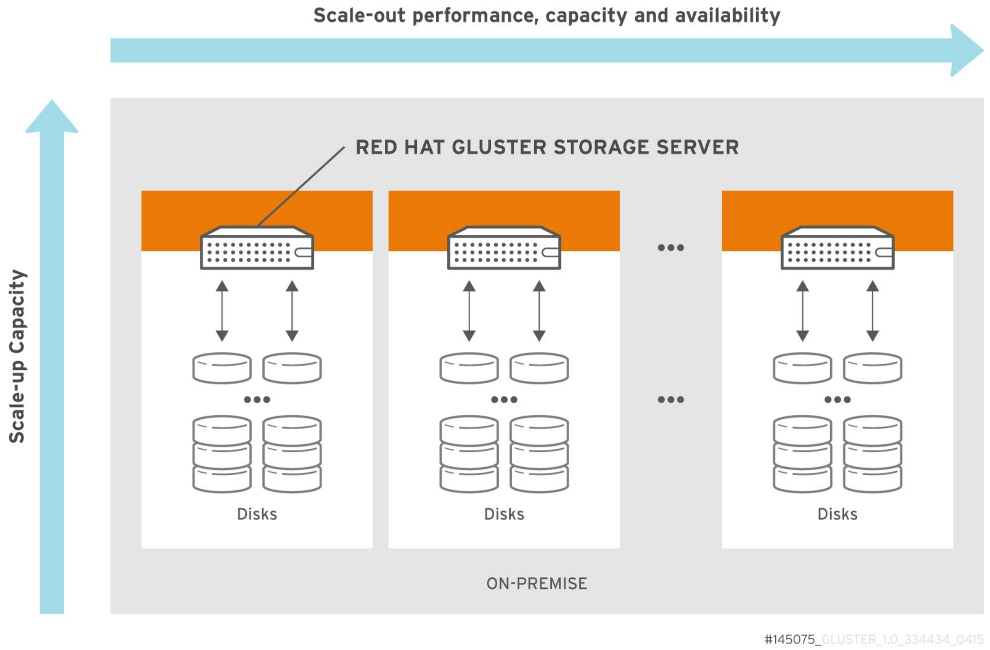
- **Hardware** implementation:
  - Use of RAID controllers
  - Manage system independently of OS
  - Offload I/O operation and parity computation
  - Cost usually high
- **Software** implementation:
  - OS used to manage RAID configuration
  - Impact on CPU usage can be high
- **Disadvantage**: scaling to multiple servers is not possible

↓

Distributed storage systems

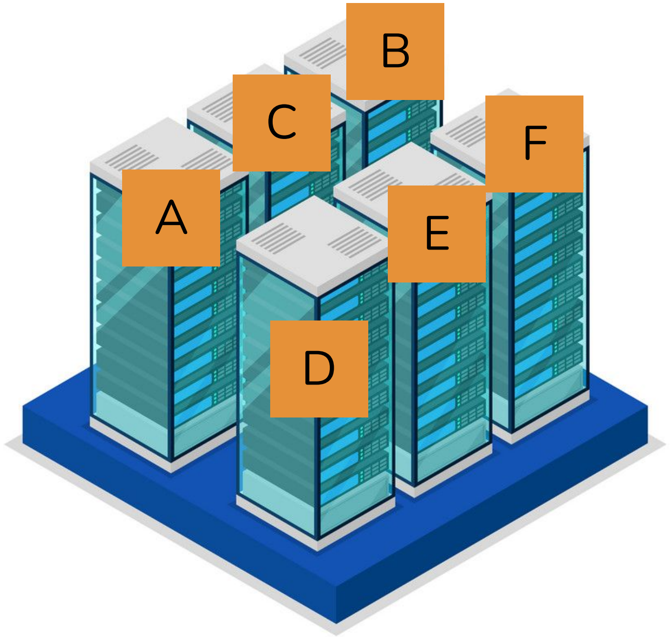
Distributed storage systems

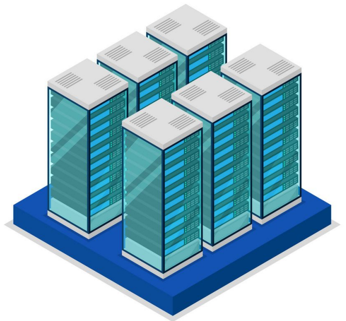
- **Distributed storage system**: files are shared and distributed between multiple nodes
    - Active communities (Red Hat, IBM, Apache, Intel)
    - Example: Ceph, Gluster, Hadoop, Lustre
    - Used by some experiments (CMS)
    - Interesting features:
        - load balancing
        - data replication
        - smart placement policies
        - scaling up to O(1000) nodes



Distributed storage systems in DAQ

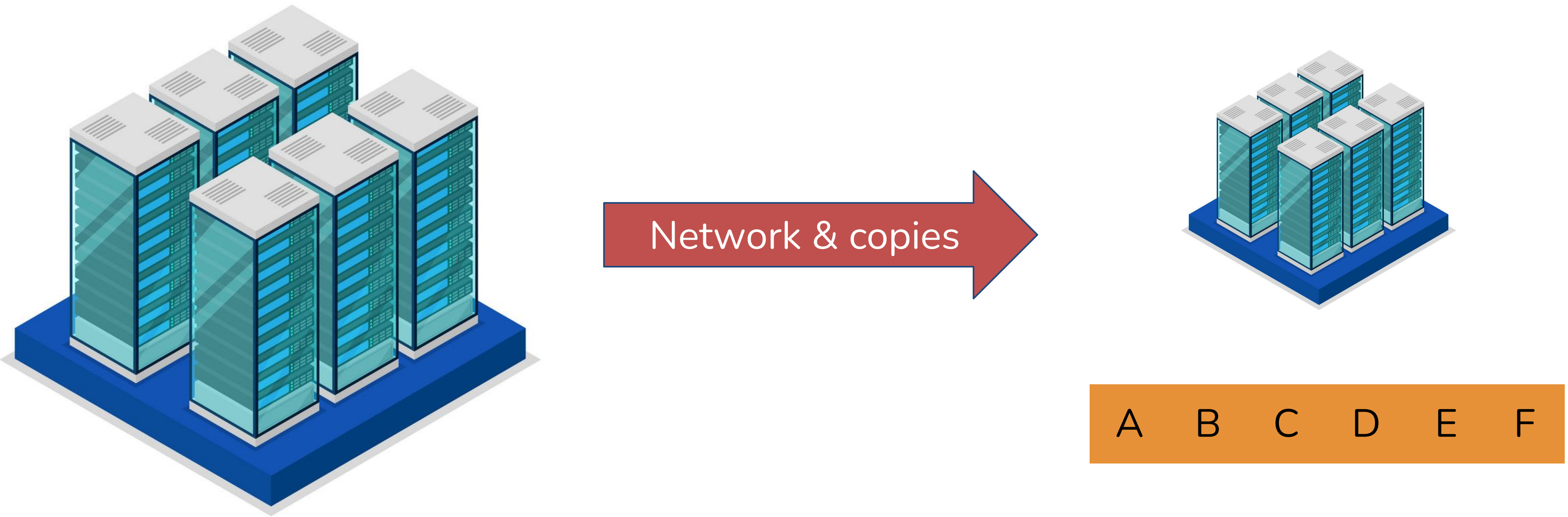
- **Application in DAQ**: implementation of the **event builder**:
  - **Physical event building** (**traditional approach**): data fragments are fetched explicitly over a network from temporary buffers at the readout nodes to a single physical location





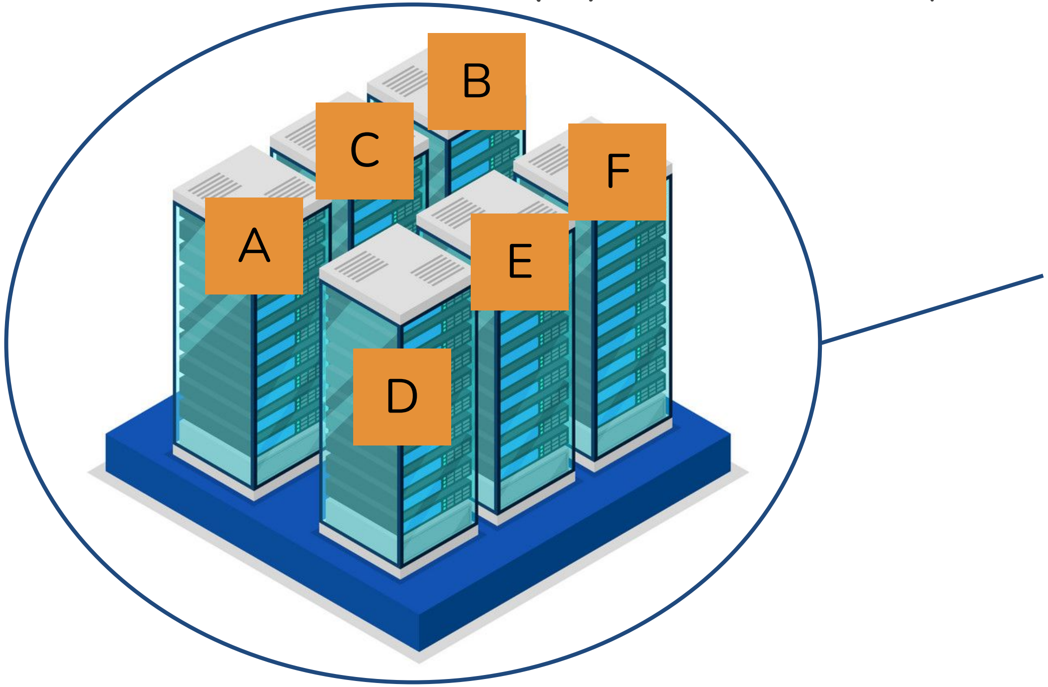
Distributed storage system in DAQ

- **Application in DAQ**: implementation of the **event builder**:
  - **Physical event building** (**traditional approach**): data fragments are fetched explicitly over a network from temporary buffers at the readout nodes to a single physical location



Distributed storage system in DAQ

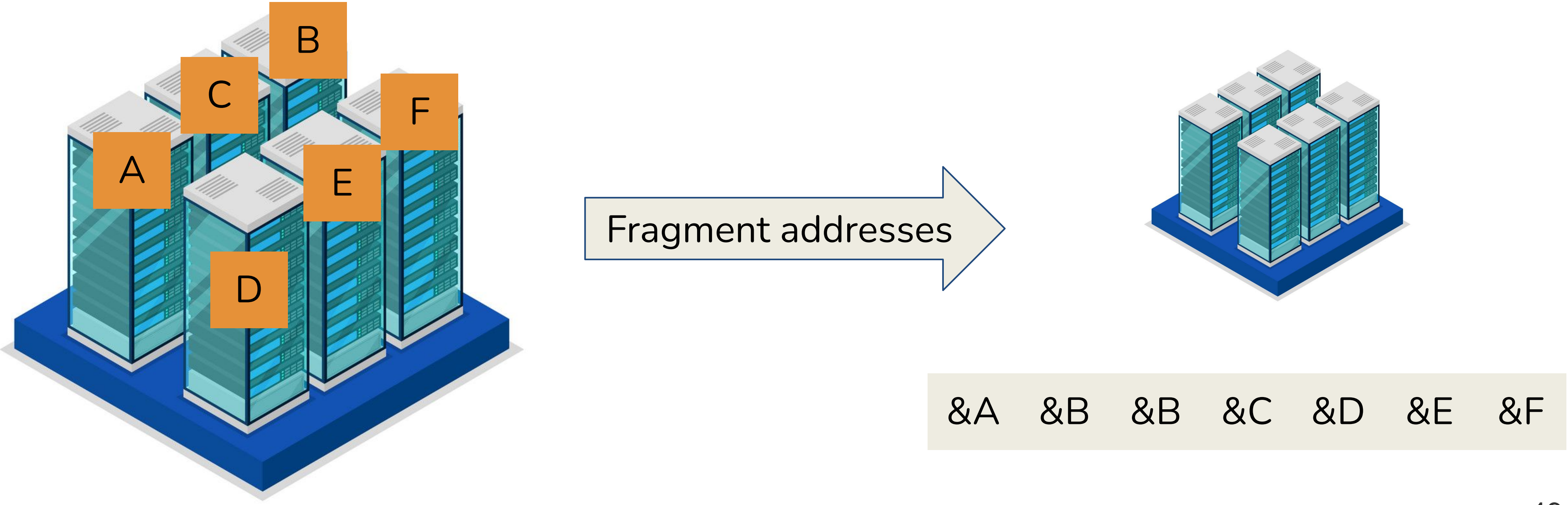
- **Application in DAQ**: implementation of the **event builder**:
  - **Logical event building**: fragments are stored in a large distributed system and events are built by computing the location of the fragments (metadata operation)
- **R&D** for future DAQ systems: ATLAS (Phase-II), DUNE, etc.



Intel DAOS
(Distributed Asynchronous Object Store)

Distributed storage system in DAQ

- **Application in DAQ**: implementation of the **event builder**:
  - **Logical event building**: fragments are stored in a large distributed system and events are built by computing the location of the fragments (metadata operation)
- **R&D** for future DAQ systems: ATLAS (Phase-II), DUNE, etc.



DAQ takeaway

Storage technologies

- Different storage media available on the market for different use cases
  - Long term storage, mostly sequential access → HDD
  - Low latency and large capacity → SSD
  - High rate and persistent → Non-Volatile memory
  - Fast and temporary → DRAM
- Keep in mind that **price/GB** changes a lot for different storage media
- When designing a DAQ system always keep an eye on the target throughput and required rate for your application
- **Data safety** and **reliability** is an important factor!
  - RAID systems

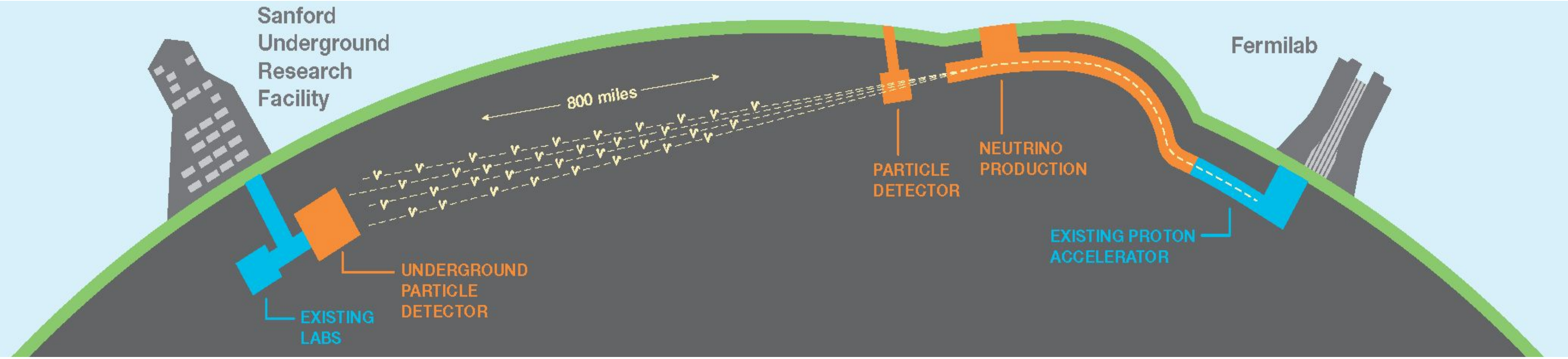
Storage challenges for the next generation DAQ systems

- Physics signals are rare!
  - Higher intensity beams are needed
  - More granular detectors
  - Consequence: store more data
- HL-LHC: Data rates and data bandwidths will increase by ~ 1 order of magnitude
  - Consequence: scale DAQ system
  - Use commercial off-the-shelf technology as much as possible
- Current storage landscape
  - HDD: large and cheap streaming storage
  - SSD: low latency and high throughput


DEEP UNDERGROUND
NEUTRINO EXPERIMENT
PR-538
C.M.U. 40/10t
DE SERIE 10886
ANNEE 2015
Storage systems in HEP
Source: CDS

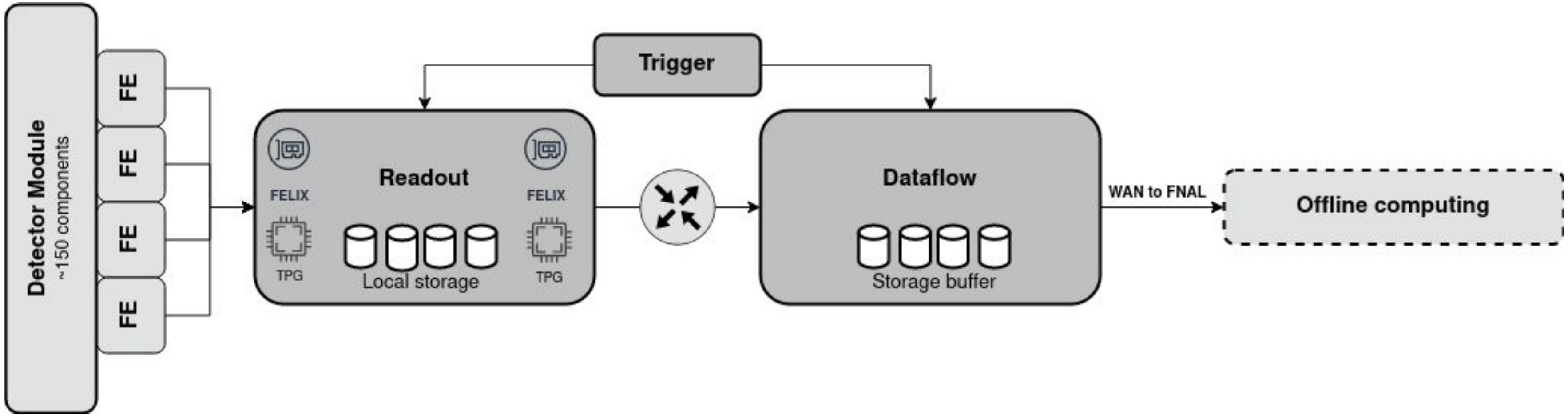
DUNE experiment
Quick overview

- Neutrino experiment located at Sanford Underground Research Facility in South Dakota
- Far detector located 1300 km away from source and approximately 1.5 km underground
- 4 modules of 17 kton LAr time projection chamber
    - Each module can be split in ~150 identical components
- Prototypes available at CERN in the North Area (ProtoDUNE)



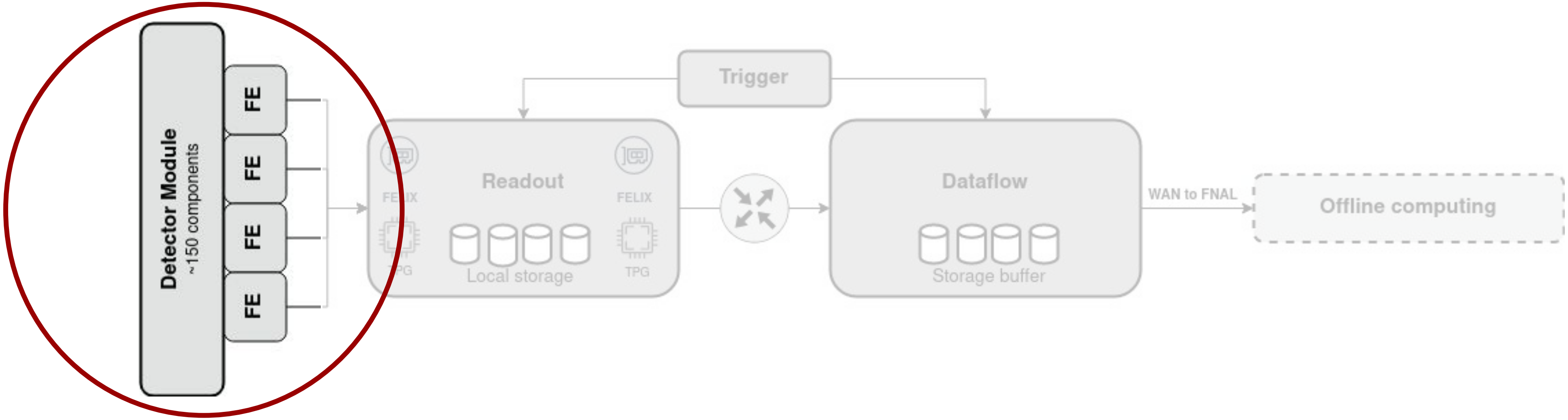
DUNE Data AcQuisition system (DAQ)

- Modular nature of the apparatus allows splitting a cryostat in ~150 identical components



DUNE Data AcQuisition system (DAQ)

- Modular nature of the apparatus allows splitting a cryostat in ~150 identical components

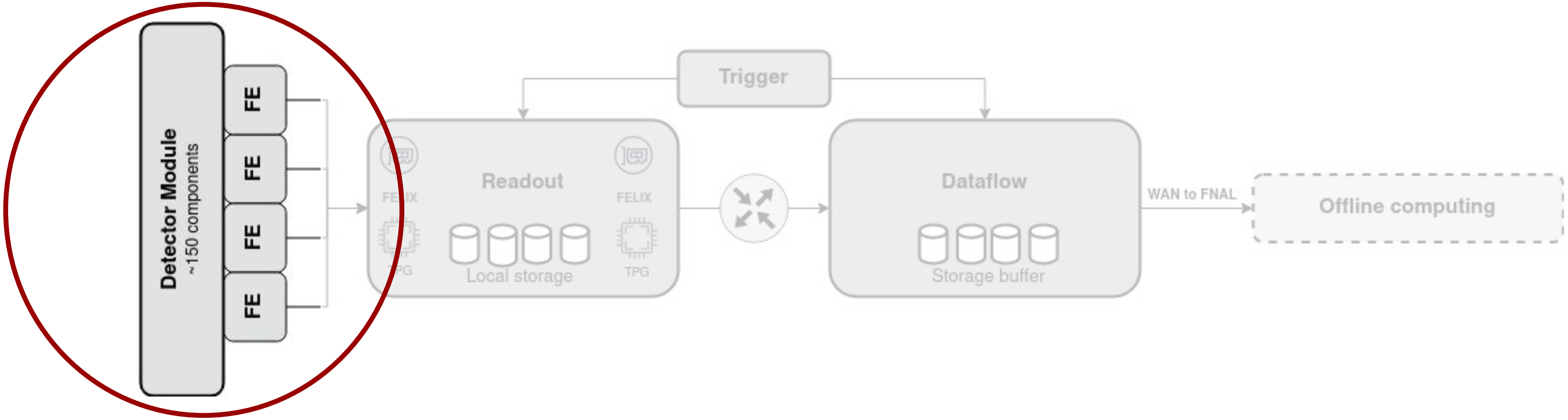


DUNE uses a continuous readout for the LArTPC

- 2 MHz sampling rate, 384k channels, 14 bit ADC
  - Throughput: 1.5 TB/s
- Adding up all the TDAQ from the four cryostats leads to ~**6 TB/s**
  - Similar rate expected for HL-LHC experiments !

DUNE Data AcQuisition system (DAQ)

- Modular nature of the apparatus allows splitting a cryostat in ~150 identical components

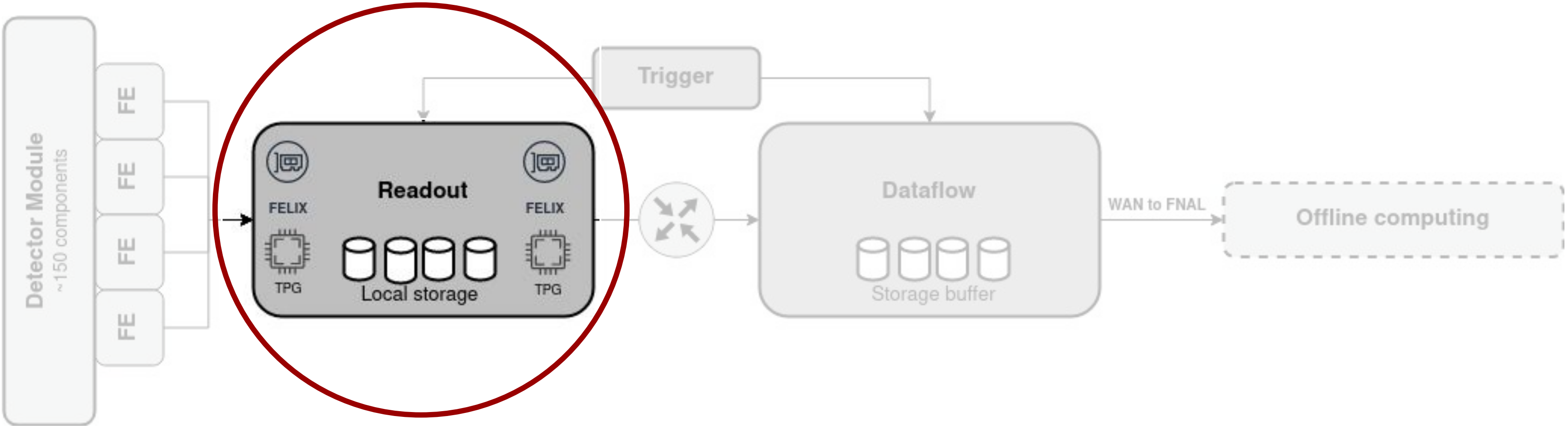


DUNE uses a continuous readout for the LArTPC

- 2 MHz sampling rate, 384k channels, 14 bit ADC
    - Throughput: 1.5 TB/s
- Adding up all the TDAQ from the four cryostats leads to ~**6 TB/s** = 1000 movies in 4k per second NETFLIX
    - Similar rate expected for HL-LHC experiments !

DUNE Data AcQuisition system (DAQ)

- Modular nature of the apparatus allows splitting a cryostat in ~150 identical components

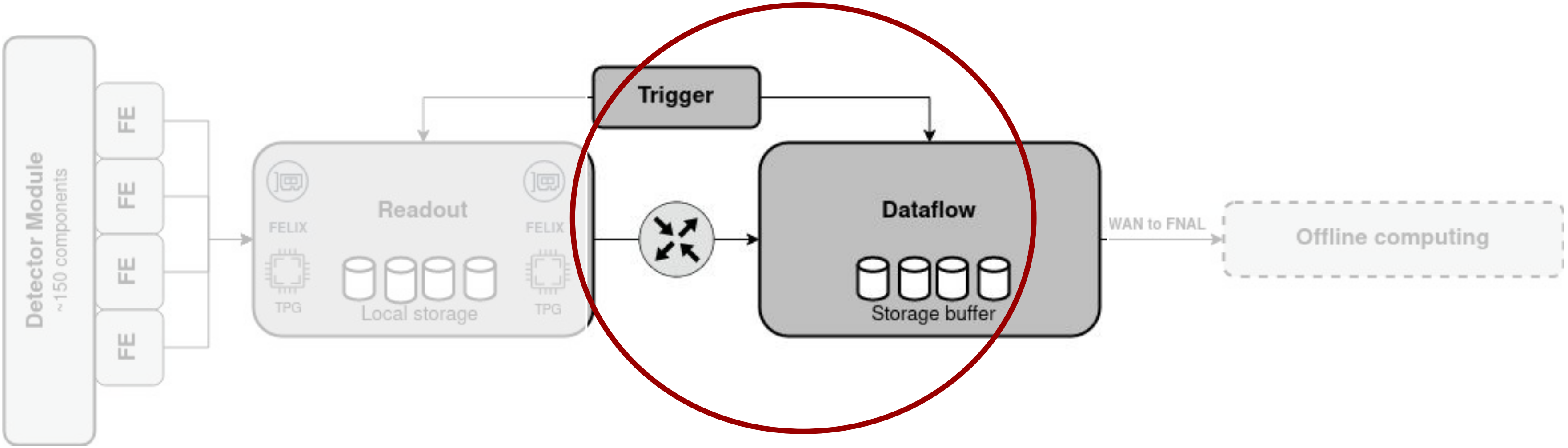


Readout system interfaces the detector front-end with the DAQ processing units

- Commercial-off-the-shelf server with multiple uses:
  - Detector interface: handle the data input from the front-end electronics of the detector
  - Low-level data selection system (*Trigger Primitive Generation*): identify time periods in which the waveforms are noise-free
  - **Local storage buffer**: temporary store the data while waiting for a trigger decision
- **Data throughput** for each readout unit: approximately **10 GB/s**
  - 150 identical readout units —> total of ~1.5 TB/s for each cryostat

DUNE Data AcQuisition system (DAQ)

- Modular nature of the apparatus allows splitting a cryostat in ~150 identical components



Trigger combines a subset of readout (TPs) data into time windows of interesting signals:

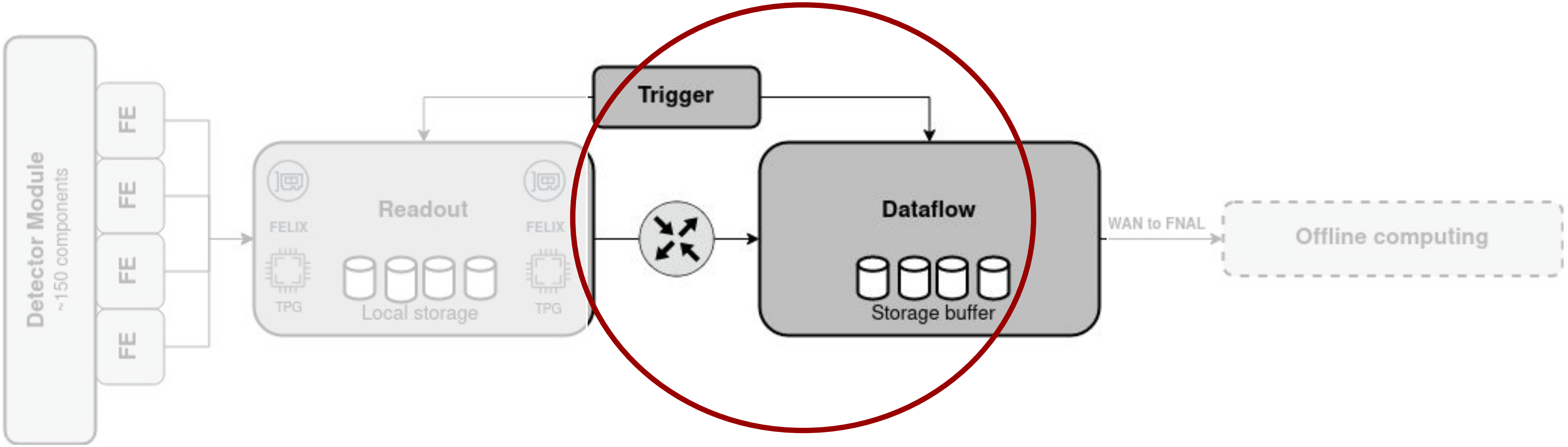
- Time “window” can vary from < 1 ms to ~100s;
- Data size ranging from few MB to ~150 TB

Dataflow moves the data fragments (identified by the trigger) from the Readout nodes to a large storage buffer

- Total storage size is 1 PB (approximately one week of data taking)

DUNE Data AcQuisition system (DAQ)

- Modular nature of the apparatus allows splitting a cryostat in ~150 identical components



Trigger combines a subset of readout (TPs) data into time windows of interesting signals:

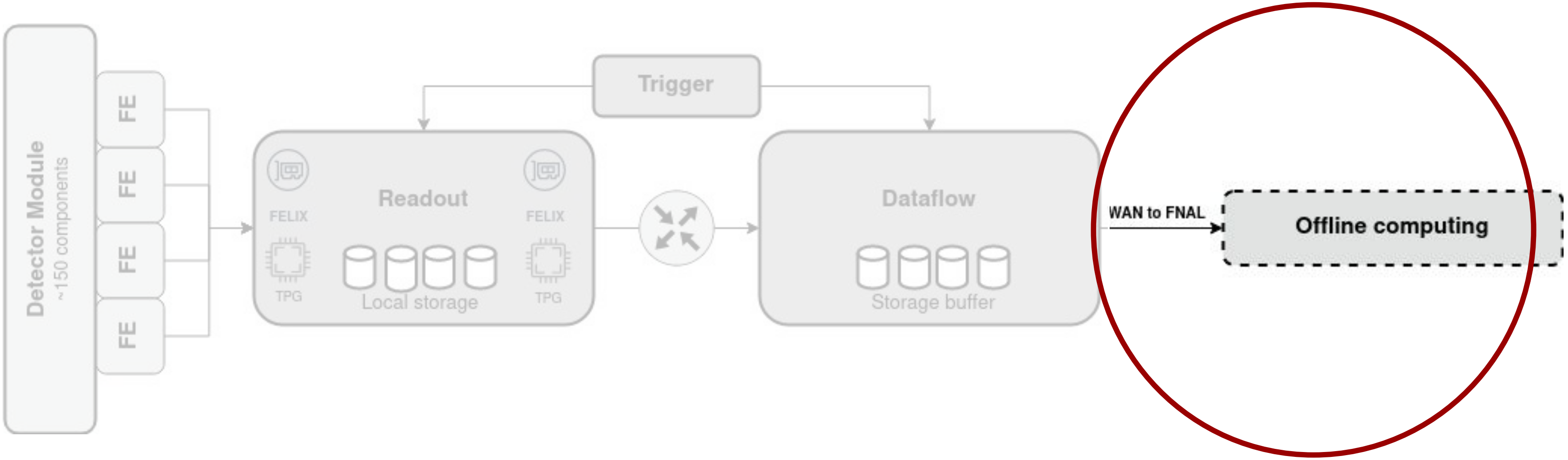
- Time “window” can vary from < 1 ms to ~100s;
- Data size ranging from few MB to ~150 TB

Dataflow moves the data fragments (identified by the trigger) from the Readout nodes to a large storage buffer

- Total storage size is 1 PB (approximately one week of data taking) = 150k movies in 4k NETFLIX

DUNE Data AcQuisition system (DAQ)

- Modular nature of the apparatus allows splitting a cryostat in ~150 identical components



Transfer recorded data to Fermilab computing infrastructure

- Total transfer of 30 PB/year (across all detector modules)

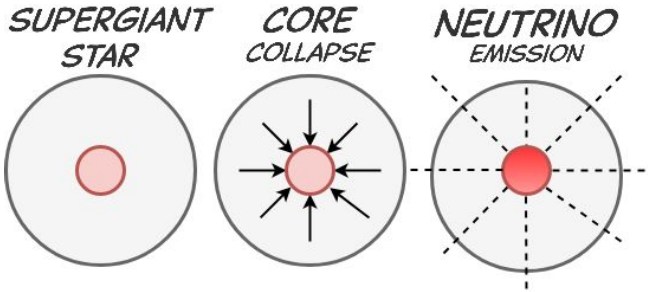
Physics constraints on the DUNE DAQ

The physics goals of the DUNE experiment heavily drive the DAQ design

- Wide physics program results in the study of many different types of events
  - Support data taking over a wide energy spectrum
    - Trigger system will need both a self triggering mechanism for the many low-energy deposits as well as a triggering system for the high energy (>100 MeV) interactions
    - DAQ must support a very wide range of readout windows
      - Data size can vary several orders of magnitude (from MB to TB)

Storage system and buffering becomes crucial to support all data taking operations

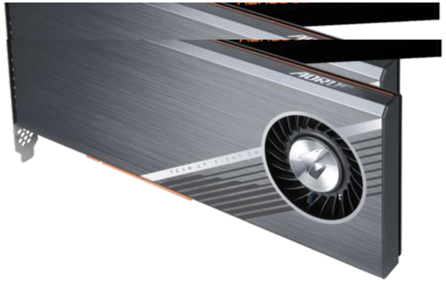
Supernova Neutrino Burst



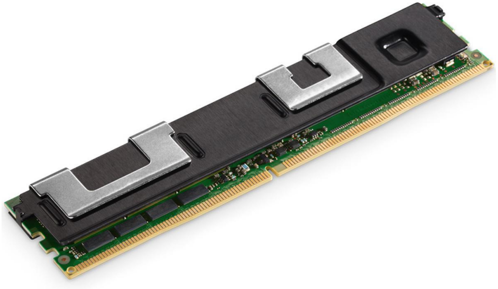
- **Supernova Neutrino Burst** (SNB) detection
  - One of the physics goals of DUNE
  - Detection of **rare** and **low energy** event
- Data taking of SNB events is **complex**:
  - Long trigger latency
  - Physics event distributed over time
  - Critical data: avoid any potential loss
- **Requirements**:
  - A single detector module generates O(10) GB/s
  - On supernova trigger: persist O(100) seconds (i.e. 150 TB per cryostat)

Supernova Neutrino buffer

Persistent memory



- Critical data and high bandwidth:
  - Take advantage of storage adapters
    - Connect multiple SSD drives together: up to 4 x PCIe 4.0 devices
  - Use of Non-Volatile Memory technology (3D XPoint)
- **Successful prototypes** capable of buffering data from the readout system
  - Store for over 100 seconds
  - Sustained target throughput of 10 GB/s
- Successfully tested in DAQ software
  - Next step: full integration of the devices within the DUNE DAQ



Conclusions

- Storage system is crucial for physics results
- Online data taking has different requirements from offline analysis
- Design of a storage system:
    - Focus on bandwidth to support the system
    - Latency constraints
    - Access pattern
    - Several storage media for different use-cases (HDD, SSD, NVM, DRAM)
    - Take into account redundancy and fault tolerance
- Benchmark performance of devices. Tools: DD and FIO (and many others)

ISOTDAQ

International School of Trigger and Data Acquisition

Thank you ! Questions ?

adam.abed.abud@cern.ch