Software driven implementation of the latency buffer and supernova buffer with COTS solutions

Adam Abed Abud (Univ. Liverpool / CERN)

Upstream DAQ Readout Technology Review
July 30, 2020 (CERN)

Outline

Implementation of the 10s latency buffer

- Description of the latency buffer
- Implementation and integration in ProtoDUNE-SP Phase 1
- COTS technology: market overview (CPUs and interconnects)

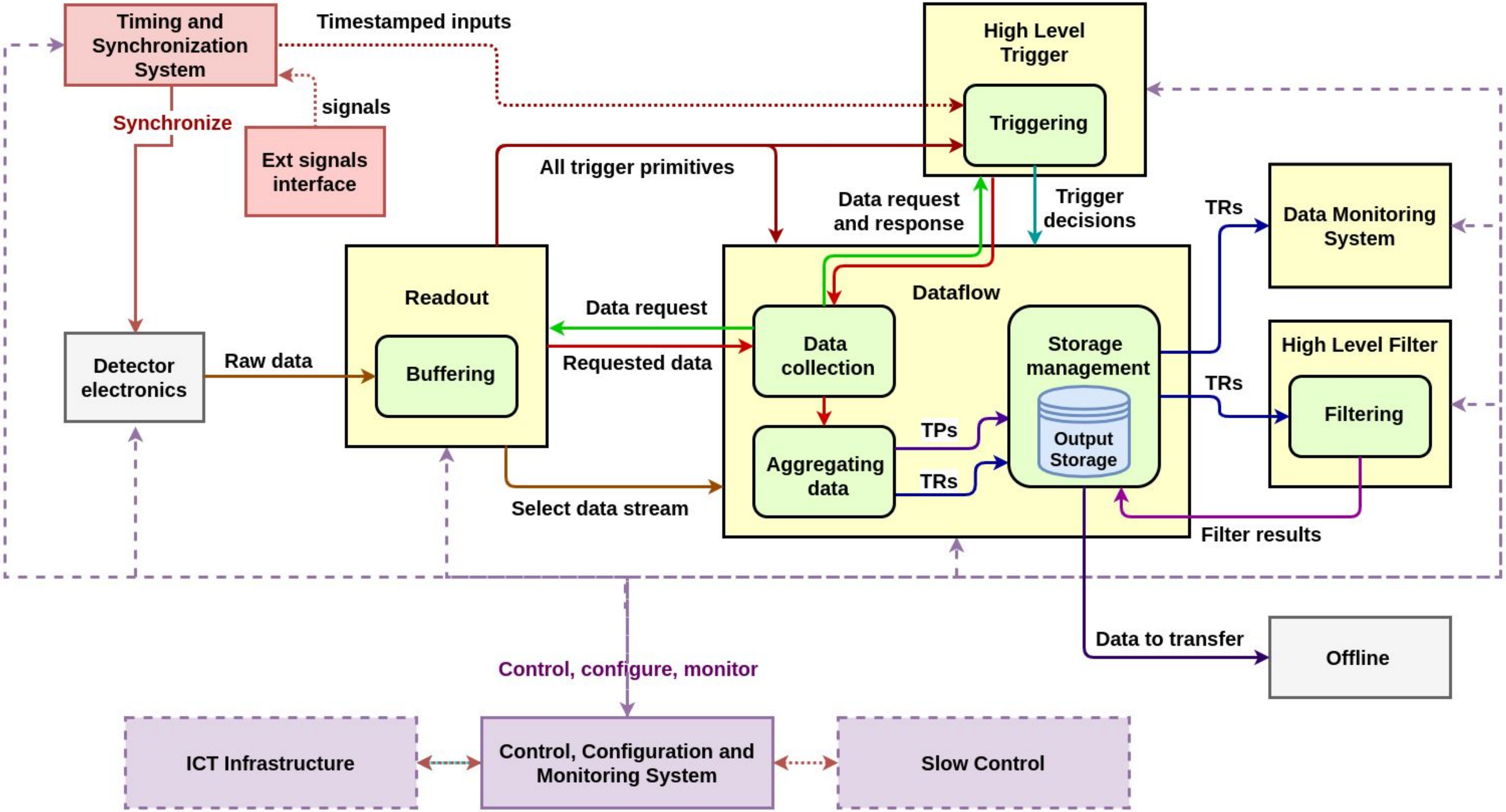
Supernova storage buffer with COTS solutions

- Description of the supernova storage buffer
- Implementation with COTS solutions
- COTS technology: market overview (storage devices)

Conclusion and outline

DUNE DAQ

System design

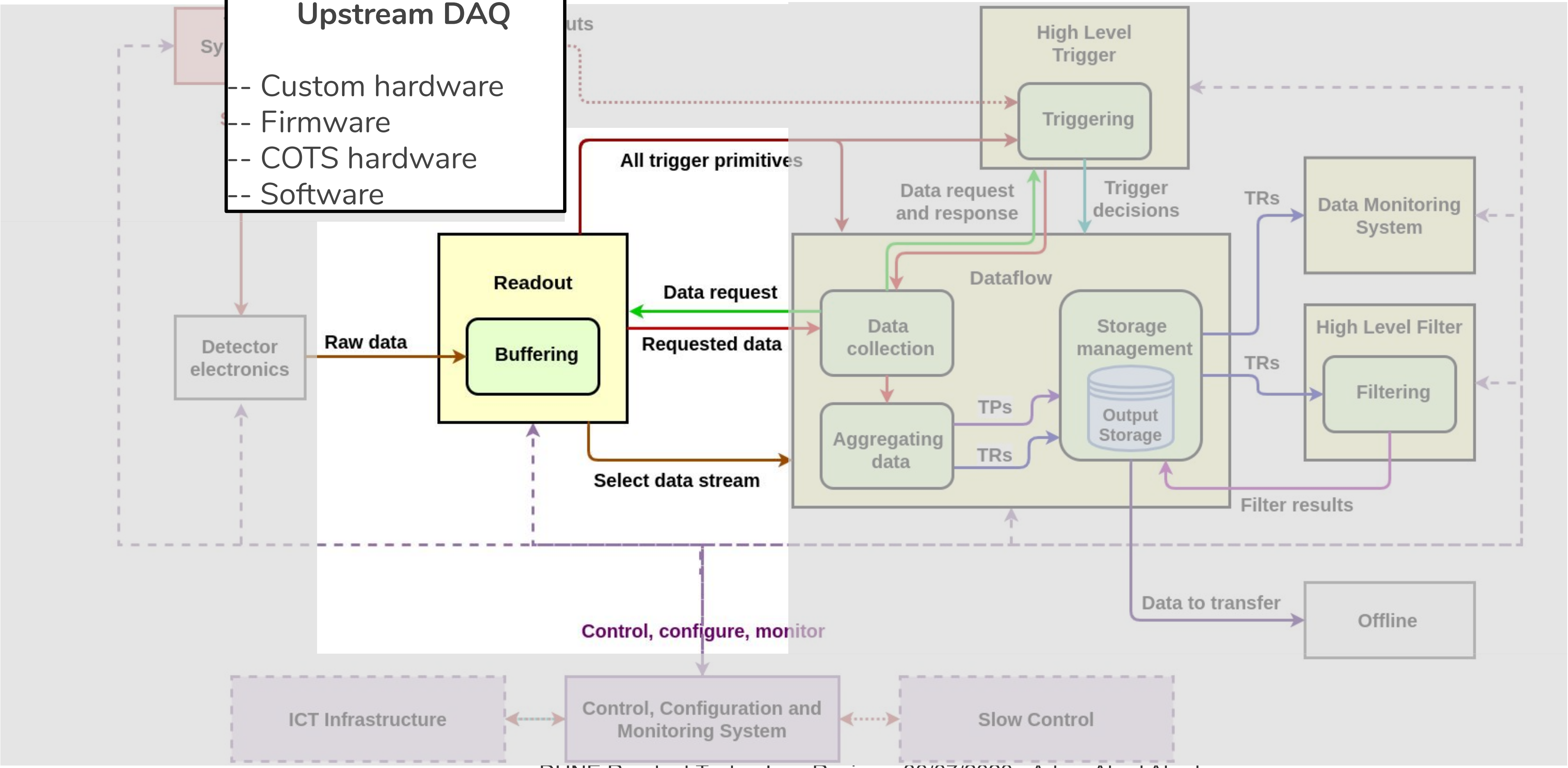


DUNE DAQ

System design

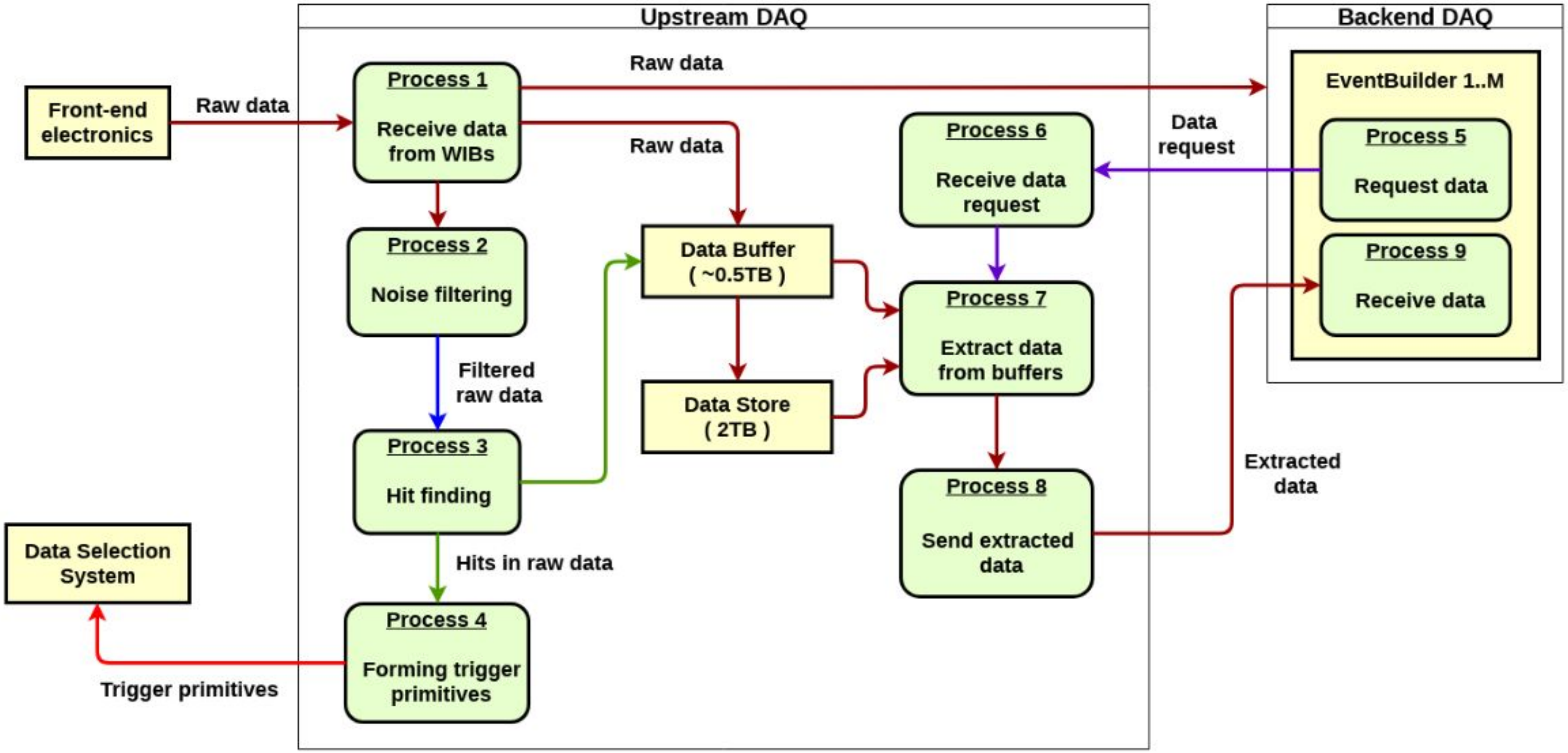
Upstream DAQ

-- Custom hardware
-- Firmware
-- COTS hardware
-- Software



Upstream DAQ

System design

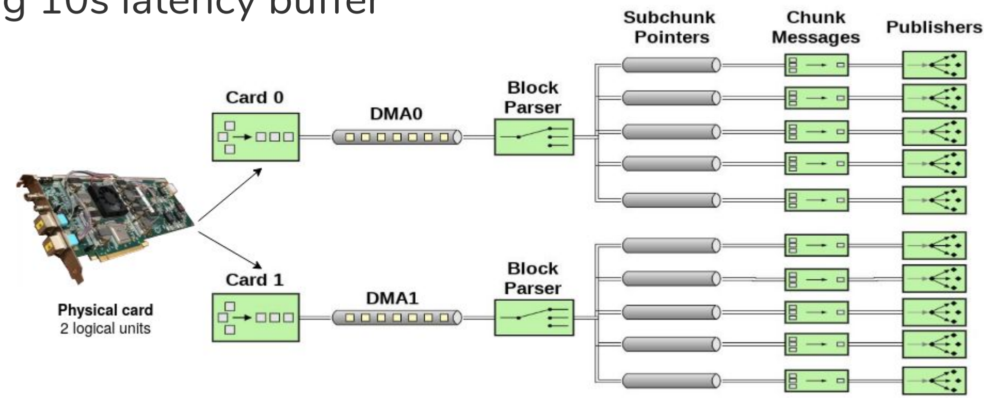


Software driven implementation of the latency buffer

Readout system in ProtoDUNE

FELIX readout

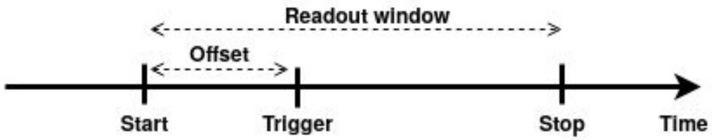
- All data is transferred from the FELIX card to the host's DRAM continuously via DMA
- Block data is parsed from DMA buffers to circular buffers (i.e. 10s latency buffer)
  - Implementation based on lock-free SPSC queue from folly library
- For each of the ten FELIX links there is a corresponding 10s latency buffer
  - User payload: 464 bytes @ 2 MHz per link
  - Firmware aggregation of the data payload
    - 5568 bytes (superchunk) @ 166 kHz
    - Lower memory I/O operations
  - Total rate for 10 links (per APA): 8.8 GB/s *



(* for rough estimates we may refer to 10 GB/s for a single APA)

FELIX OnHost BoardReader Application

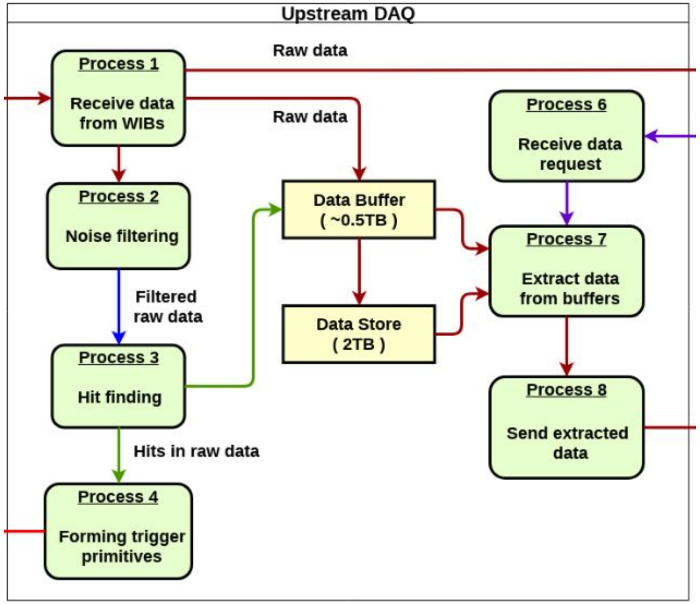
- Serializes and buffers up the superchunks (aggregated WIB frames) from the DMA blocks
- Receive trigger information (timestamp and event identifier)
  - Note: in PDSP trigger requests are time ordered



- On trigger request:
  - Calculate position of requested data using timestamp of the queue's oldest WIB frame
    - WIB frames are continuous and time ordered
    - Use 25ns ticks to convert timestamp request into position request for the SPSC queue
  - Pop items from queue until requested timestamp is matched (time ordered requests in PDSP)
- Extract data defined by a specific readout window (configuration parameter)
  - Note: extraction would also work if window size is given with trigger request
- Form artDAQ fragments from extracted data

Implementation of the data buffer

Features demonstrated in ProtoDUNE (1/2)



- **Receive data from links for a full APA**
- **Buffer raw data**
- **Data request handling** (trigger matching)
- **Data extraction from buffer**
  - Serve data for software based hit finding
  - Long readout window extraction (dataflow long window tests)
  - Feed data to *accelerators*
    - Hardware accelerated compression
    - ML/DL inference

Implementation of the data buffer

Features demonstrated in ProtoDUNE (2/2)

- **Interface with Dataflow system**
  - Form artDAQ fragments
  - Send fragments to event builders for downstream processing
- **Interface with CCM**
  - BoardReader fully integrated into the CCM software of NP04
  - Control and configuration by using RPC libraries (xmlrpc)
    - Example: Link initialization, configuration, GTH resets
    - Use of low level tools provided by the FELIX software suite
  - Monitoring of the link status
- Prototype of SNB data store **integrated** and **tested** with BoardReader application

Implementation of the latency buffer

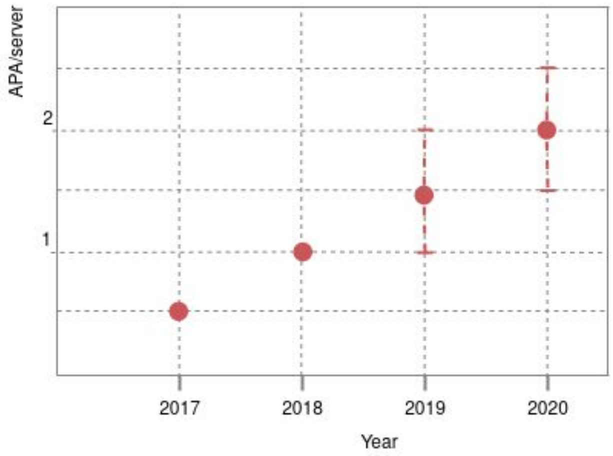
Adaptability of the solution

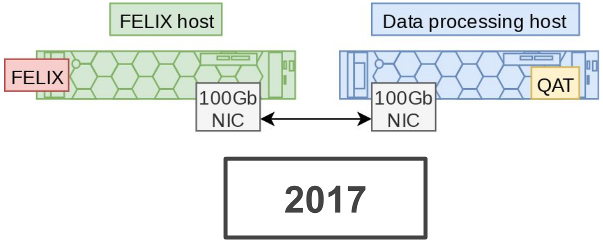
- Support **both SW and FW** based **hit finding** solutions
- Feed the data to hardware accelerators
    - Demonstrated both hardware accelerated compression and ML/DL inference
- Adapt to evolving computer architectures and promptly integrate them in the system
    - Successfully transitioning from 2 servers for a single APA to testing 2 APAs with one server (in 2 years)
- **Flexibility** provided by using a software defined solution
    - Adapt to new requirements (e.g. variable trigger windows with different offsets)
    - Low level software and consolidated integration tools provided by COTS solutions

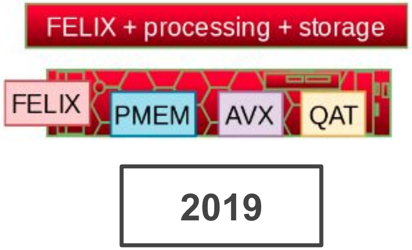
Implementation of the latency buffer

Adaptability of the solution

- FELIX default operation: read from latency buffer over a 100Gb network interface
- Adapt to evolving computer architectures:
  - Higher memory bandwidth systems
- Promptly integrate developments into production system
  - **2017**: reading half of APA with 1 server
  - **2020**: testing 2 APAs in a single server







Readout operation modes of DUNE

DUNE requirement	ProtoDUNE testing
Long readout tests	Tested 1s readout window
10s offset	Tested and verified
Optional data compression on ouput	Tested in PDSP
Handling non-time ordered requests	In progress: modification of queue access mechanism
2 APAs on single host	In progress: testing 2 FELIX card

- Provide data for some or all channels with a variable readout window and offset
  - Data requests are not time ordered
  - Readout and offset variability on event by event basis
  - Partial overlap of trigger windows
- Readout requirements and their implementation with COTS solutions
  - Efficiently adapt to different readout modes
  - Tested several DUNE requirements with ProtoDUNE-SP

Reliability of the latency buffer

Failures and mitigation

- **Detection of failures**: rate for a single link drops to 0 or rate fluctuation are detected
- Reliability in case of individual link failures
  - Recovery mechanisms are in place on the FELIX side
    - E.g: transceiver misalignment failure: control, realign, reset links and flush buffers (if needed)
  - Enabling/Disabling single links
- Data inspection and anomaly detection on the WIB content <u>not implemented</u> yet
  - Need to be evaluated for ProtoDUNE-II

Long term support and maintenance

COTS components and invested effort

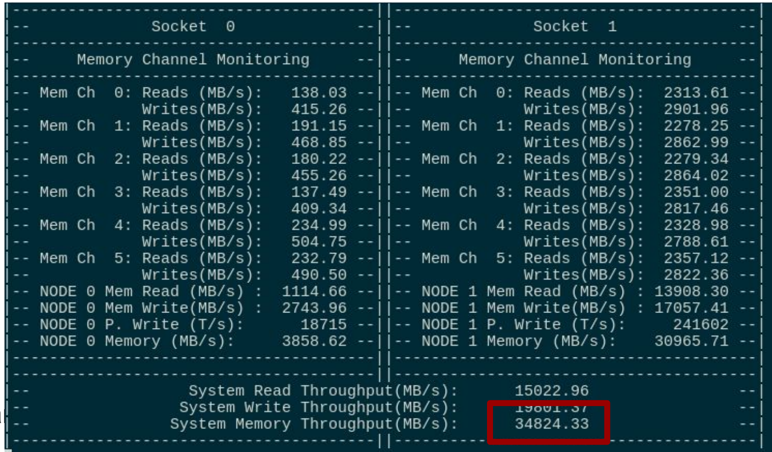
- Balance between in-house development and COTS components
  - Latency buffer implementation with COTS solution relies **only** on server grade memory modules
    - **Requirement** for 10 links: 12 second buffer x 10 GB/s ---> **120 GB**
  - Profit from developments made by industry on memory bound high throughput applications
  - Large user community using the same technologies (e.g. large in-memory databases, high-speed video streaming, RDMA for high speed low latency applications)

- Invested time effort for the integrated system:
  - Core development of basic functionality: ~6 months by 3 FTE developers
  - Interfaces and extensions of this for final datataking: ~3 months by 2-3 students

Current setup

Testing platform used in ProtoDUNE

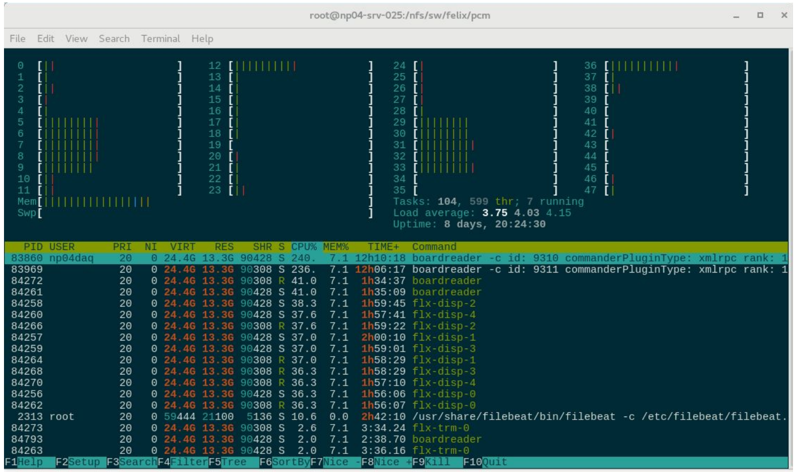
- Example of platform used for testing
  - Wide range of servers tested in ProtoDUNE as FELIX host (mid range to very high end)
  - **Example:** Intel® Xeon® Gold (high mid-range server), dual socket, 48 threads in total
  - 192 GB DDR4 2666 MHz DRAM (used approximately 65% for the BoardReader application)
  - Memory bandwidth per socket: ~120 GB/s
- During operation the OnHost BR application has an impact of ~30-40 GB/s on active memory throughput of the system (30% of single socket's maximum throughput)
- Advantages of sw driven implementation of 10s latency buffer
  - Scale up the system with marginal cost increase (mainly due to cost of DRAM)
  - With current setup extra memory is still available for other tasks



Resource requirements for OnHost BoardReader

Resources used and future prospects to reduce them

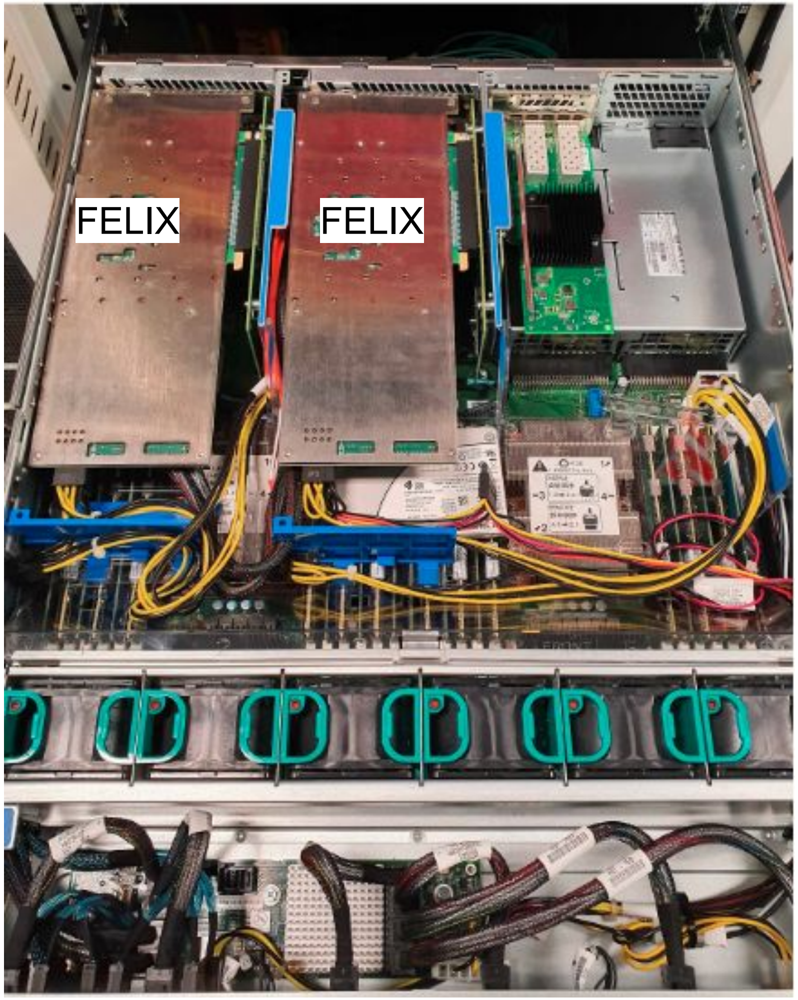
- CPU utilization is ~30-45 % for each of the ten core threads
    - Many factors affecting the system: L3 cache size, BIOS configuration, NUMA awareness, interrupt balancing
- Configuration for maximum throughput
    - Pinning threads to physical cores
    - Tuned NUMA locality
- **Future prospects** to best exploit performance:
    - Systematic study of interconnect technologies and cache-friendly optimizations on different computing architectures



Setup evolution for DUNE

Testing 2 APAs in a single host

- **Target goal for DUNE**: run 2 APAs (i.e. 2 FELIX cards) on a single host
- **Consequences** on the latency buffer
  - Double the amount of DRAM available:
    120 GB x 2 ---> 240 GB
  - Maximum throughput achieved by fully populating memory DIMMs
- **NOTE**:
  - 2 card setup is being evaluated
  - **In progress**: optimization of the FELIX driver for a 2 card setup
  - Hit finding studies made possible thanks to the COTS solution (see Phil's talk)

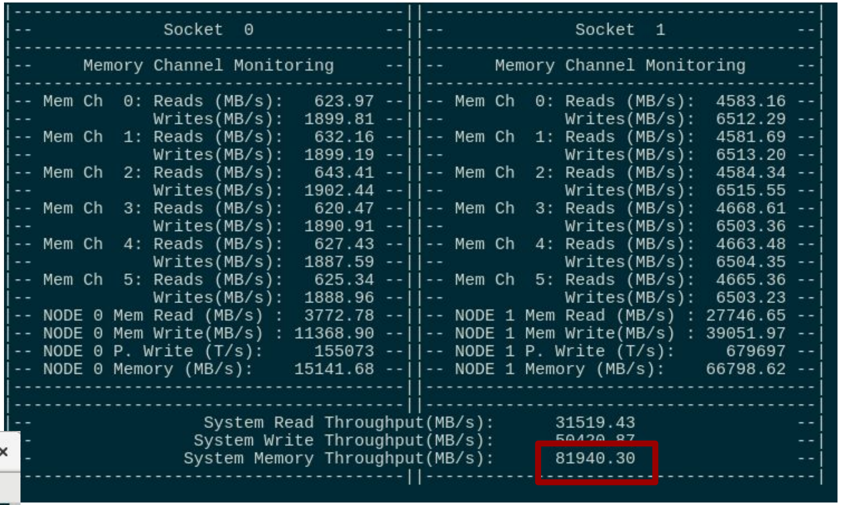


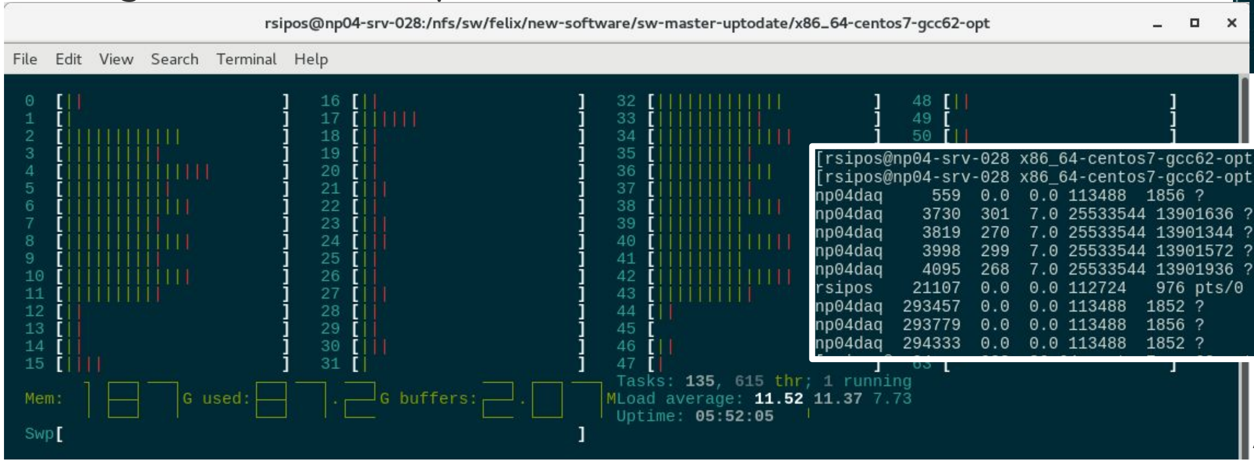
Test with two card setup on a single host

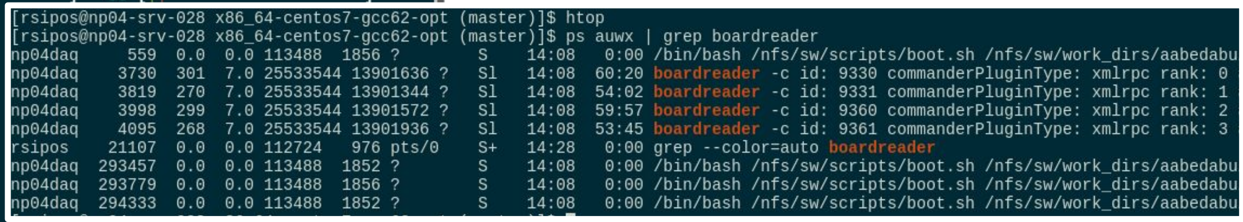
Setup evolution for DUNE

Testing 2 APAs in a single host - “Hot off the press”

- BoardReader applications buffering data from 2 APAs (APA2 and APA6)
- Integration with OnHost BR in PDSP (~5s buffering)
    - Not enough memory available on host for 10s setup
- Memory throughput scales accordingly
- Core pinning and tested also without
    - Same cores are parsing both of the cards
- Logs available on np04-srv-028

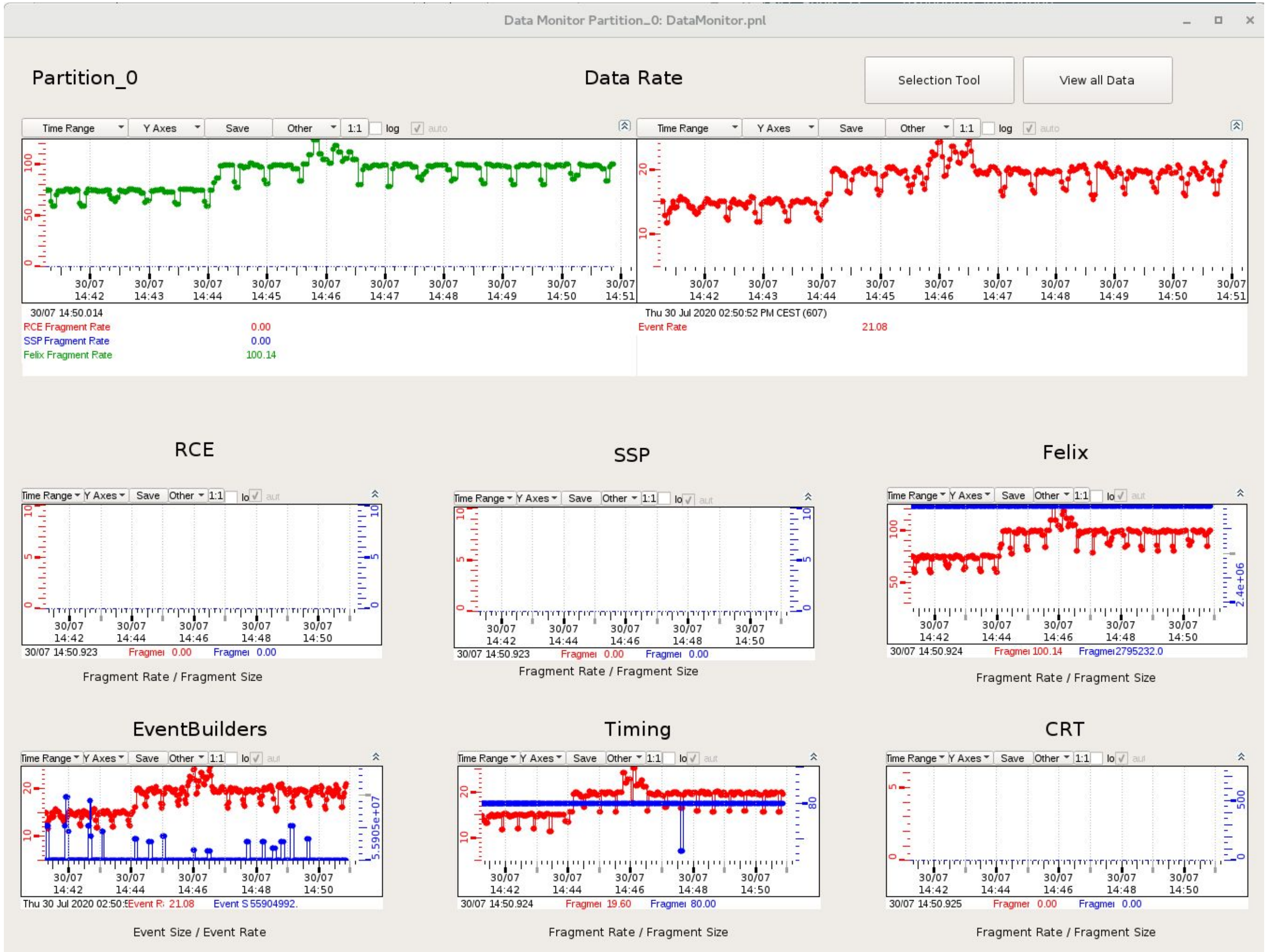






Setup evolution for DUNE

Testing 2 APAs in a single host - "Hot off the press"

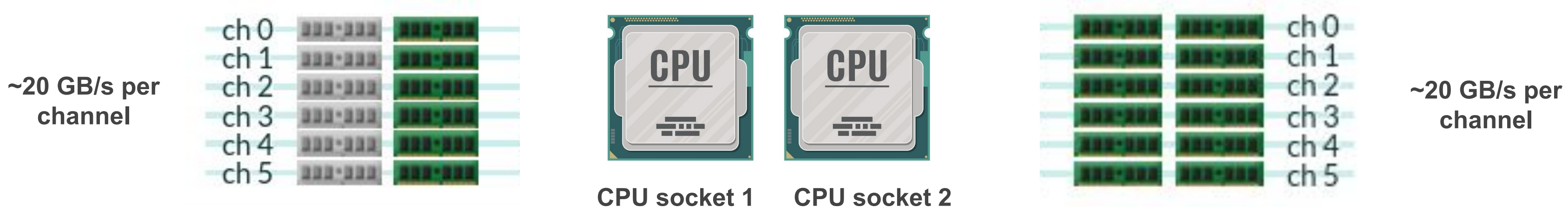


Long term support and maintenance

Can the COTS solution profit from R&D outside DUNE?

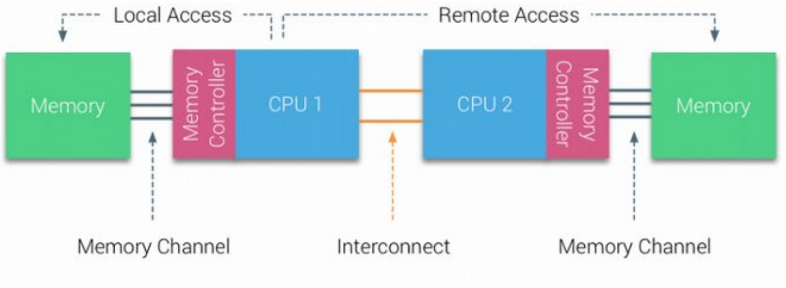
- Developments and R&D driven by datacenter industry
  - Solution relies on software and low level tools written by industry
  - Higher memory bandwidths (today’s maximum ~120 GB/s, recent architectures over 200 GB/s)
  - Increased memory channel density for higher throughput applications
    - Today 6 channels per CPU socket; 8 channels per socket already existing
  - DUNE bandwidth requirements are <u>feasible with current generation</u> of COTS technologies

Example of current dual-socket computer architecture with memory throughputs for six channels:



COTS technology

Today's trends



Source: http://frankdenneman.nl/2016/07/07/numa-deep-dive-part-1-uma-numa/

- **CPU:**
  - Increasing number of physical cores per socket
  - Fixed processor clock speed
- **Memory:**
  - Higher memory-CPU bandwidths
- **Interconnect**
  - PCIe 4th Gen is on the market
  - PCIe 5th Gen specification is ready and CXL (Compute Express Link) for fast interconnect has been agreed by major companies
- **Advantages for DUNE**:
  - **Server configuration** and **memory bandwidth** will not be a critical element
    - Current generation architectures already provide required bandwidths

Conclusion and outlook

- Software implementation of the 10s latency buffer is **feasible** with today's technologies
    - **Tested** and **verified** several DUNE requirements with ProtoDUNE readout system
    - **Flexibility** of the BoardReader application is provided by the COTS implementation
        - Example: change trigger readout windows and adaptability to multiple interfaces
    - Satisfy DUNE requirements by taking advantage of developments in industry
        - COTS technologies evolving in favor of DUNE use-cases
- Current readout software is at still a prototype, it may be redesigned for DUNE

Outlook

- Reliability: error handling for data loss and data corruption to be tested in ProtoDUNE-II
- Demonstrate 2 FELIX card setup on a single host
- Evaluate server configurations relevant for DUNE readout system

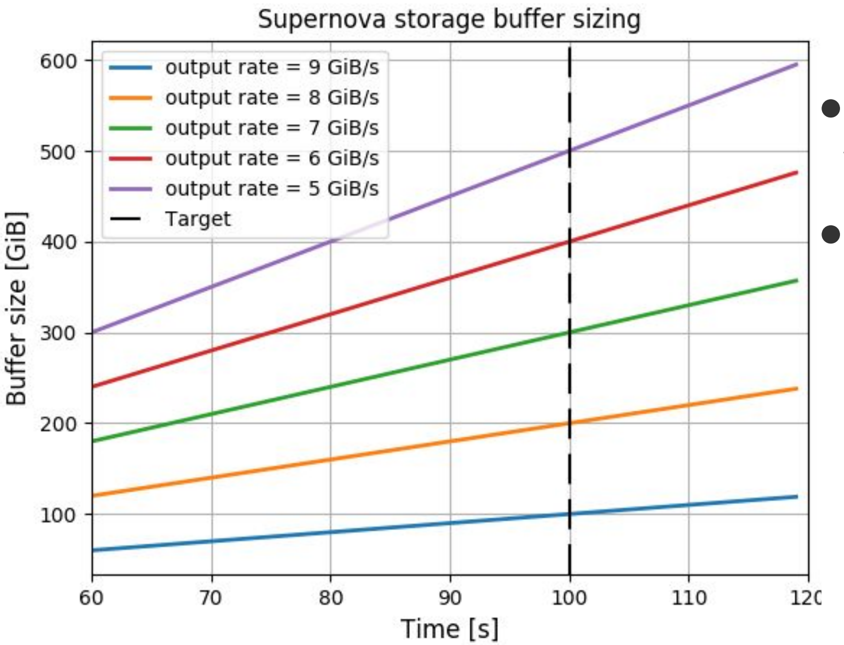
Software driven supernova storage buffer with COTS solution

DUNE Upstream DAQ

Supernova storage buffer (SNB)

- **Goal:**
  - Store more than 100s of continuous data stream from the detector
  - Design storage solution capable of writing out data files with required throughput for 2 APAs
- **DUNE requirements:**
  - Throughput of 10 GB/s per APA (**DUNE requirement: two APAs in single host 20 GB/s**)
  - On trigger continuously persist 1 TB of data per APA (2 TB for two APAs in single host)
  - Store data volume for 2 supernova events (4 TB in total for 2 APAs)
- **Software defined storage solutions**
  - Develop functional aspect of the solution <u>independently</u> of the underlying hardware
  - Suitable COTS technologies relevant today for the supernova buffer:
    - PCIe 4 SSD and 3DXPoint storage devices
    - RAID system
    - Persistent memory devices

Software defined storage implementation



- Simulation of the latency buffer size assuming input bandwidth of 10 GB/s
- Output bandwidth (sequential writing) of different storage technologies affects the total size of the latency buffer
- **Examples for a target data taking time of 100s**
  - Fast storage device with output bandwidth of 8 GB/s
    - Oversize latency buffer by 200 GB of DRAM
    - **Commonly available** in today's servers
  - Slow storage device with output bandwidth of 5 GB/s
    - Oversize latency buffer by 500 GB of DRAM
    - Supported on current generation of servers

Balance performance of storage technologies, features provided and overall sizing of the system

Storage technologies suitable for SNB

PCIe 4th gen SSDs and modern storage media

- **PCIe 4 gen SSDs**
  - Bandwidths up to 8 GB/s for a 4 lane device
  - Reasonable to expect an average bandwidth per device of ~ **5 GB/s**
  - High bandwidth achieved by using performance tools developed by industry (e.g. SPDK)
- Some devices already available today on the market [EXAMPLE]
  - NVMe PCIe 4th Gen x4 lanes with sequential write bandwidths 5 GB/s
  - Combining two devices already satisfies throughput requirement for 1 APA
- **3DXPoint storage devices** (e.g. Intel/Micron)
  - Claimed 9 GB/s in sequential write throughput (single device for one APA)
  - Establishing partnership and preparation of a testing platform in the next months
  - Planning full evaluation of the technology to test suitability for DUNE DAQ applications

Storage technologies suitable for SNB

RAID system

- RAID = Grouping multiple disks to form a single logical volume
- **Applications** for the **DUNE** supernova storage buffer:
    - Load balancing
    - Increase throughput
    - Data redundancy
    - Device hot swapping in case of failures
- RAID systems based on PCIe 4.0 x16 lanes SSD adapters already existing [LINK]
    - Connect up to 4 SSD devices with bandwidths up to 20 GB/s (realistically 15 GB/s)
        - Comfortably satisfies supernova buffer throughput requirement for single APA
    - Procurement expected in the next months to test the technology

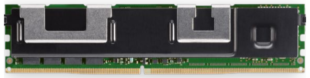
Storage technologies suitable for SNB

Example of RAID configurations with today’s SSDs

- Example of **RAID0 (stripe)** system with PCIe 4th Gen (x4 lanes) SSDs
    - Single drive write bandwidth: 4.5 GB/s
    - 2 non-redundant RAID groups for two APAs
    - Each RAID group with 3 drives of 1 TB:
        - Total bandwidth per APA: **13500 MB/s**
        - Comfortably satisfies supernova DUNE requirements for 2 APAs
- Example of possible **RAID10** system based on PCIe 4th Gen (x4 lanes)
    - Single drive write bandwidth: 4.5 GB/s
    - 2 redundant RAID groups for the two APAs
    - Each RAID group with 4 drives of 1 TB
    - Total bandwidth per APA: 9000 MB/s
        - Almost fully satisfies DUNE requirements for 2 APAs with data redundancy

Storage technologies suitable for SNB

Persistent memory devices



- Memory devices with high bandwidth: O(10) GB/s
  - Suitable candidates for the DUNE supernova buffer
- **Affordable large capacity**
  - Available device sizes: 128 GB, 256 GB, 512 GB
  - Today’s maximum capacity per CPU socket 3 TB
    - Suitable for two APAs
- Persistent Memory Development Kit (**PMDK**)
  - Well established low level software stack to develop applications for PMEM devices
  - Multiple use cases: object storage, data logging, memory mode
  - Performance optimization:
    - Direct access operation
    - High writing throughput less dependent on blocksize (contrary to storage devices)

Supernova storage buffer with persistent memory

Features

- Validated feasibility of persistent memory devices for the supernova storage buffer
  - Sustained 80% of required data throughput for a single APA
  - Stored superchunks of 5568 bytes at 166 kHz per link
  - Produced and persisted more than 100 seconds of data for a single APA
- **Integration with Upstream DAQ**
  - Fully integrated the SNB prototype with the OnHost BoardReader application
  - Implemented trigger command to send data to persistent memory devices instead of forming artDAQ fragments for downstream processing

Supernova storage buffer with **persistent memory**

Example of data stored

```
0000430: 0a14 79b9 9090 69b0 2973 9538 5fa6 f913  ..y...i.)s.8_...
0000440: 903b 5e00 09d4 a33d aa17 d3e9 3991 9632  .;^....=....9..2
0000450: 4369 398e a3f6 b318 388d f9f2 c868 8f96  Ci9.....8....h..
0000460: e241 5819 8e95 2ba5 5963 8e39 11f1 59b3  .AX...+.Yc.9..Y.
0000470: 963e 0ccc a983 983d d36a 1389 3a8f dd9e  .>.....=.j..:...
0000480: 7358 3994 c2d4 23a8 3c98 ab13 d869 938d  sX9...#.<....i..
0000490: 0000 ec87 6b80 578c 0000 0000 aa2a aaaa  ....k.W......*..
00004a0: a95b 39c9 8990 9cc9 f803 893c a2d4 8843  .[9........<...C
00004b0: 903c 38f2 3963 963f b355 4329 3891 5b4a  .<8.9c.?.UC)8.[J
00004c0: 5389 388c 7183 9349 358e 6cc1 6818 8393  S.8.q..I5.l.h...
00004d0: 1a52 c909 9393 5ab3 09a3 953c 3893 4913  .R....Z....<8.I.
00004e0: 9c41 dbdc b9a3 9c3d de96 5369 3990 b91c  .A.....=..Si9...
00004f0: 0309 3a8e a162 d3b9 3789 fed9 c8e8 8f98  ..:..b..7.......
0000500: 0000 d942 02ae 578c 0000 0000 aaaa aaaa  ...B..W.........
0000510: 76e9 59c8 958e 7aa8 a923 9639 1c92 0ae3  v.Y...z..#.9....
0000520: 9c3e d2cd 8983 a23f ed52 5309 3a8e bdfb  .>.....?.RS.:...
0000530: d358 3993 d9db 3348 3c8d 230f caf9 968d  .X9...3H<.#.....
0000540: 2516 e9d9 9796 b5a3 09e3 983b 8ac1 0953  %..........;...S
0000550: 973d e6fc 59f3 9f40 de9e 83b8 3b9a a3f9  .=..Y..@....;...
0000560: 7358 3a98 d017 e3d9 398d 0e5b 29f9 8f94  sX:.....9..[)...
0000570: 0021 8400 0000 0000 aca4 9fa9 2450 1b01  .!..........$P..
0000580: 0000 c7a6 180c 588c 0000 0000 aa2a aaaa  ......X......*..
0000590: e4d5 28f8 8e96 5f99 e943 8e39 729f c973  ..(..._..C.9r..s
00005a0: 963d afe6 0963 a33f c159 1309 3a94 9bb5  .=...c.?.Y..:...
00005b0: 7368 398e 94d7 c388 3894 d3e8 18f8 9c90  sh9.....8.......
00005c0: 9171 2829 8d99 20da 1973 953d 49d4 b953  .q().. ..s.=I..S
00005d0: 973d a806 0924 a13f d8dd f389 3ca1 ce98  .=...$.?....<...
00005e0: 53d9 3a95 915f 1319 3991 11dc a979 9098  S.:..
```

Example of binary output in the persistent memory device

Created 10 files in total, one for each link

- 100 GB for each PMEM file

```
-rw-r--r-- 1 np04daq np-comp 105G Jul 16 16:45 pmem_pool_file_channel_0
-rw-r--r-- 1 np04daq np-comp 105G Jul 16 16:45 pmem_pool_file_channel_1
-rw-r--r-- 1 np04daq np-comp 105G Jul 16 16:45 pmem_pool_file_channel_2
-rw-r--r-- 1 np04daq np-comp 105G Jul 16 16:45 pmem_pool_file_channel_3
-rw-r--r-- 1 np04daq np-comp 105G Jul 16 16:45 pmem_pool_file_channel_4
```

Supernova storage buffer with COTS solutions

Adaptability

- Software defined storage makes the system independent of the underlying storage hardware
  - Treat the SNB buffer as a directory where to store binary files
- Possibility to include new features once the SNB event is stored
  - Post-processing on the collected data
  - Compression (if needed)
- Expose the data to different file formats:
  - Example: store the data output as HDF5 files if needed by Dataflow
- Possibility to use of the data store for other applications
  - Example: store debugging stream

Supernova storage buffer with COTS solutions

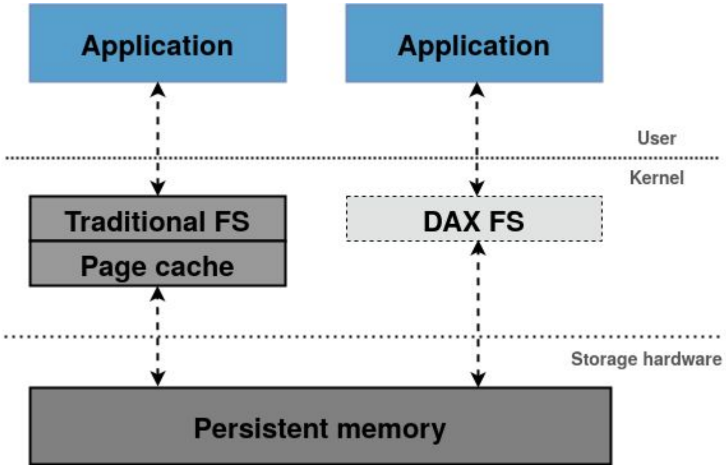
Reliability

- Data integrity can be achieved at the application level
  - Design software to handle data integrity issues caused by memory hardware errors
  - PMEM **tools** are available to handle **error correction** and **check data integrity**
    - Not tested yet in the current prototype
- Data redundancy
  - RAID: copy the collected data
  - Hot swapping of failing disks
- Different reliability features provided by different storage technologies

Current setup for persistent memory solution

Testing platform used in ProtoDUNE

- **Hardware:**
  - Cascade Lake Xeon® Platinum L SKU processor (**high end server**)
    - PMEM performance validated also on lower end servers
    - Currently testing SNB buffer integration on lower end servers
  - **192 GB DRAM** and **6 TB persistent memory**
- **Software**
  - Mounted PMEMs as **direct-access** (DAX) filesystem
  - **Low level tools** used to **write directly** to PMEM (pure device operation)
    - Prepare an empty PMEM file
    - Memory map it (function provided by PMDK)
    - Persist in PMEM devices



Current setup for persistent memory solution

Resource requirements per APA

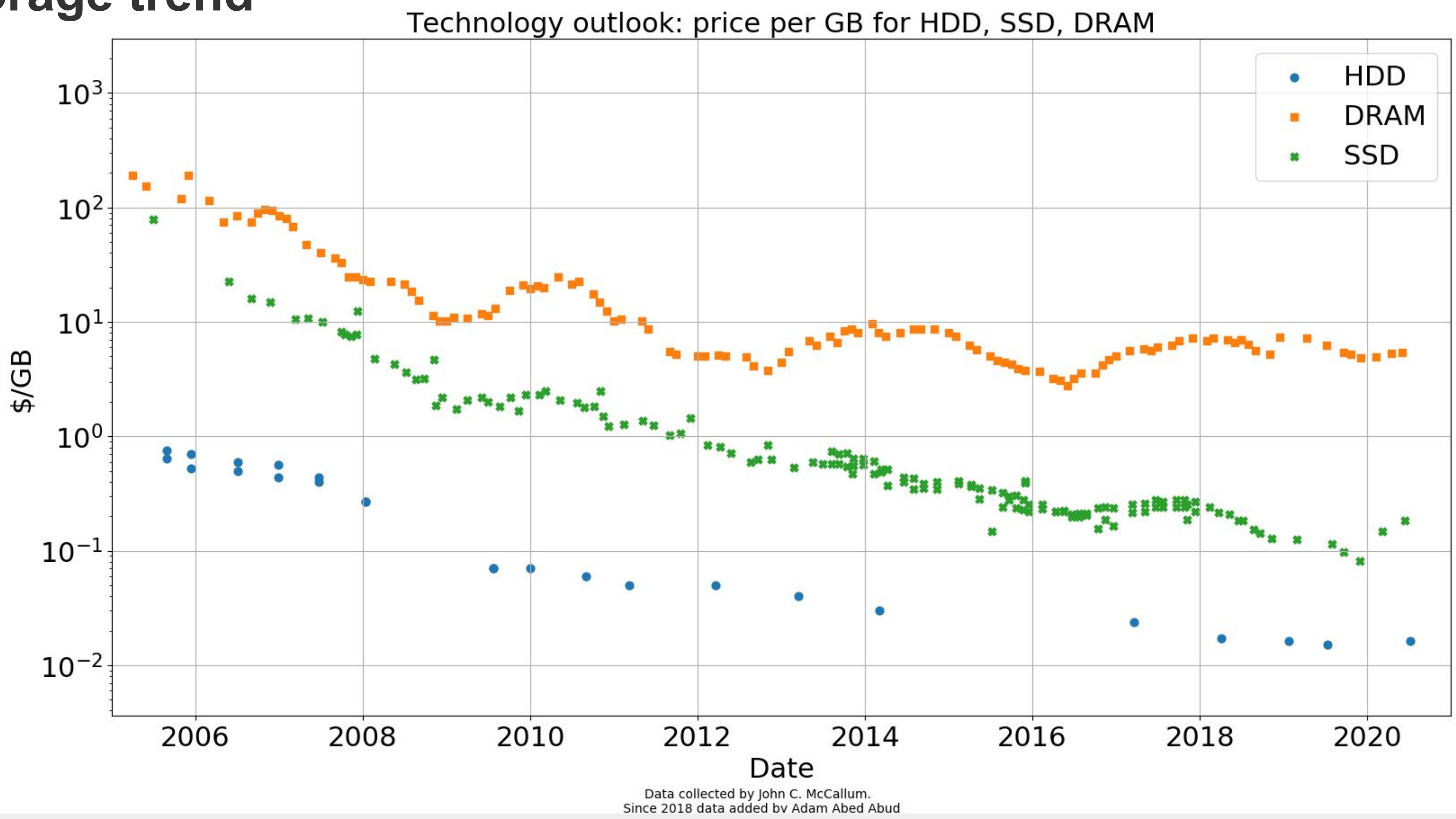
- Used separate physical cores for each FELIX link
  - Core pinning to each CPU socket is necessary to avoid remote access between NUMA nodes
- SNB storage has limited impact on memory utilization (less than 10%):
  - Advantages of using COTS tools to copy data directly into storage system (specific pmem copy function)
  - **Direct access operation**: bypass operating system cache for higher throughput
- **Future development for DUNE**:
  - Optimize the software layer to achieve maximum performance from PMEMs

Supernova storage buffer with COTS solutions
Long term support and maintenance

- High performance storage developments are driven by data center and cloud provider needs:
    - Active user community [LINK] : Cisco, Dell EMC, Huawei Cloud, Oracle, etc.
    - Many use-cases relevant for DUNE applications:
        - Improve performance of by using modern storage media
        - Advanced caching layer for low latency applications [EXAMPLE]
- Advantages provided by solution fully based on COTS components
    - Profit from R&D made by industry and use consolidated development tools
    - Use of extensively tested integration tools (configuration and debugging of the solution)
- Learning and development time for the SNB implementation with persistent memory
    - 4 months by 1 person (PhD student)
    - Further testing and integration with other sub-systems in the next months
    - Test data integrity tools provided by the PMEM technology

COTS technology

Storage trend



Conclusion and outlook

- Software based implementation of the SNB buffer is possible **today** with **COTS technologies**
    - Different solutions available: PCIe 4 SSDs, 3DXPoint devices, RAID system, PMEM devices
    - Implementation with PMEM devices developed only in a few months
    - Satisfies 80% of throughput requirement for a single APA
        - Further improvements/optimizations can still be made
- **Storage trend** is promising and in the right direction for DUNE applications
    - Large memory capacity and fast SSDs are becoming widely spread in industry
    - Industry is moving towards high throughput storage applications

Outlook for the next months

- Optimize workloads for SNB implementation with PMEMs
- Test **storage** solutions
    - Benchmark performance of PCIe 4th gen SSD devices and/or modern storage media
    - Deploy RAID system

Thank you
Questions?

Adam Abed Abud (Univ. Liverpool / CERN)

adam.abed.abud@cern.ch

Cost estimate for RAID system with PCIe 4 SSDs

RAID10 with “Sabrent Rocket” (PCIe 4th Gen x4 lanes SSDs) [LINK]

- Single drive write bandwidth: **4.5 GB/s**
- **2 <u>redundant</u> RAID groups** for two APAs
- Each RAID10 group with 4 drives of 1 TB:
    - Total bandwidth per APA: **9000 MB/s**
    - Cost: **1600 CHF** for the storage and **200 CHF** for the RAID

RAID 10 (Striped mirrors) Performance Calculation:

Total Performance = 18000 MB/s
Total usable capacity = 0.00 TB

Reads 0%, Writes 100%
Number of RAID groups = 2
Number of drives per RAID group = 4
Total number of drives = 8
Single RAID group performance = 9000 MB/s
Single drive cost = 200
Cost per TB usable = 400000.00
Total cost = 1600.00

Compare two RAID configurations...
Calculate RAID10 capacity...

Notes:
Minimum number of drives per RAID10 group = 4
Must have even number of drives.
IO penalty (read) = 1/1 (one RAID IO per each host IO)
IO penalty (write) = 2/1 (2 RAID IOs per each host IO)
Fault tolerance = 1 (min) to 2 (max) disk drives per RAID group

[LINK]

COTS technology

PCIe lane bandwidths