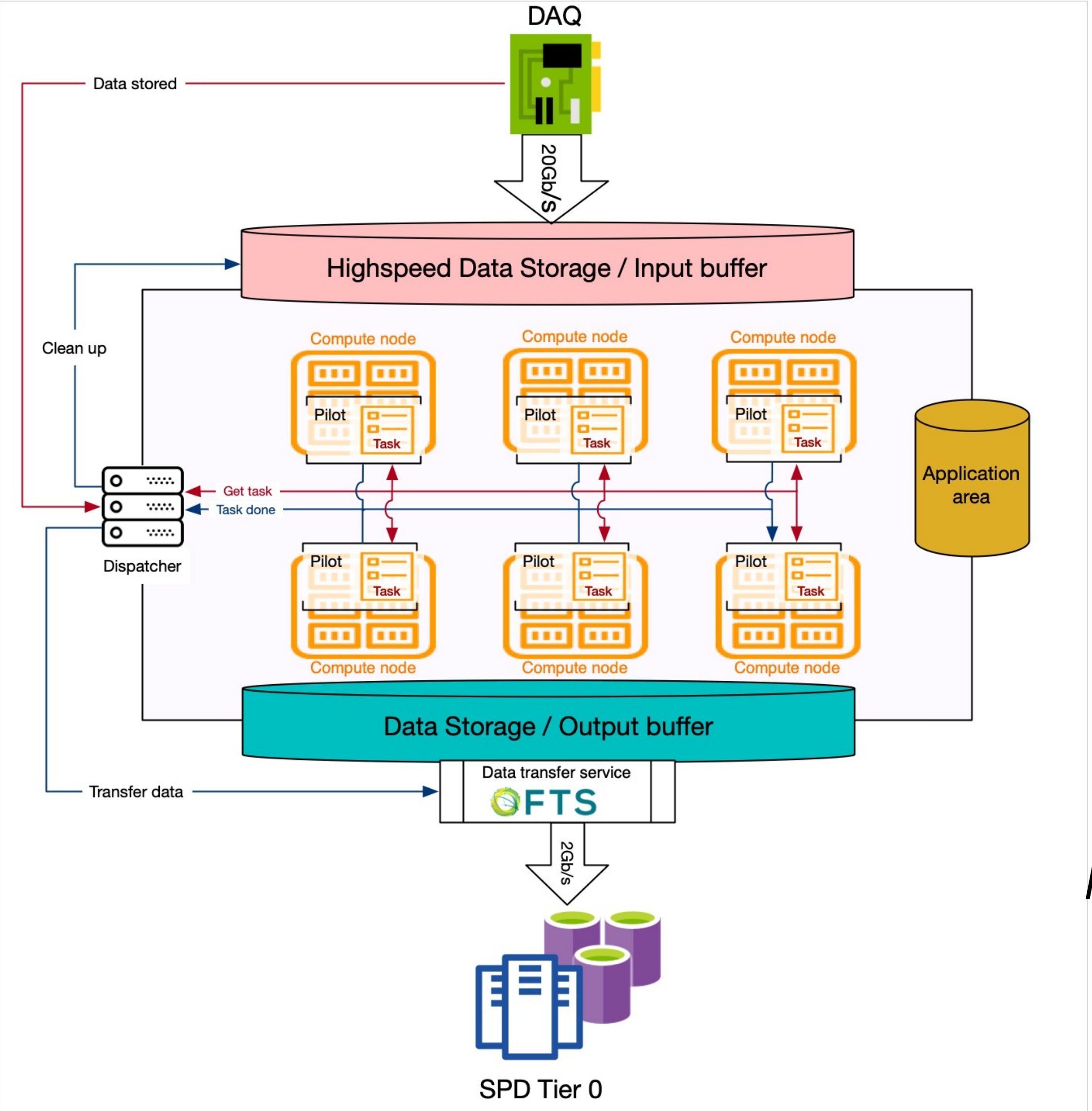
Status of the online filter

A. Zhemchugov, A.Belova, D.Oleynik, A.Petrosyan

13 December 2021

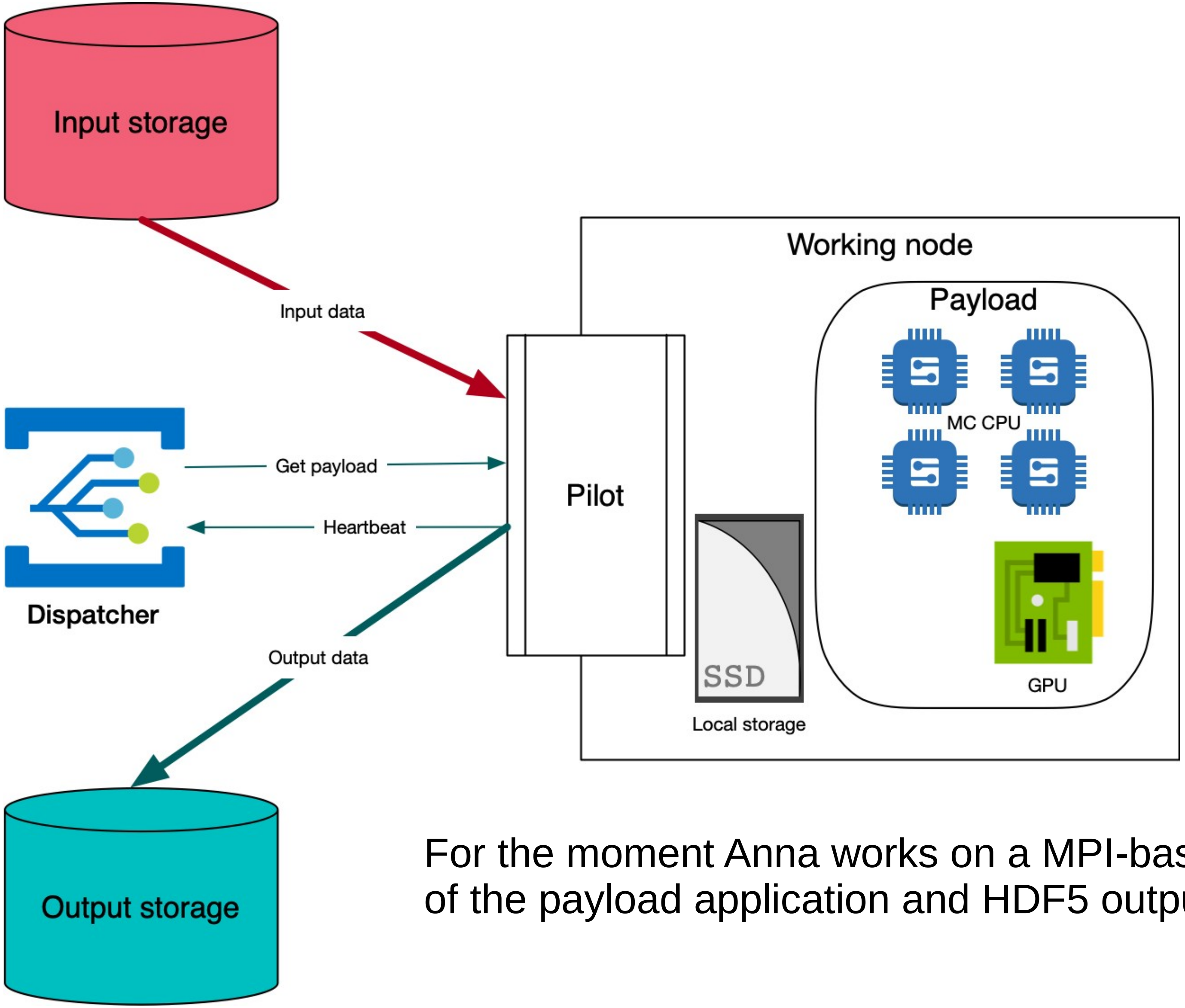
Online filter operation



https://git.jinr.ru/SPD_online_filter

Main ingredients

- **Input buffer:** 20 GB/s write, 20 GB/s read, delete 5 files/s
- **Output buffer:** 2x400 MB/s write, 2x400 MB/s read
- **Dispatcher**
- **Identical workers:** multicore nodes with GPUs or FPGA co-processors. 1000 or 5000 WNs ?— depends on the performance of our algorithms!
- *We should foresee using these computing resources for offline data processing between the data taking campaigns*



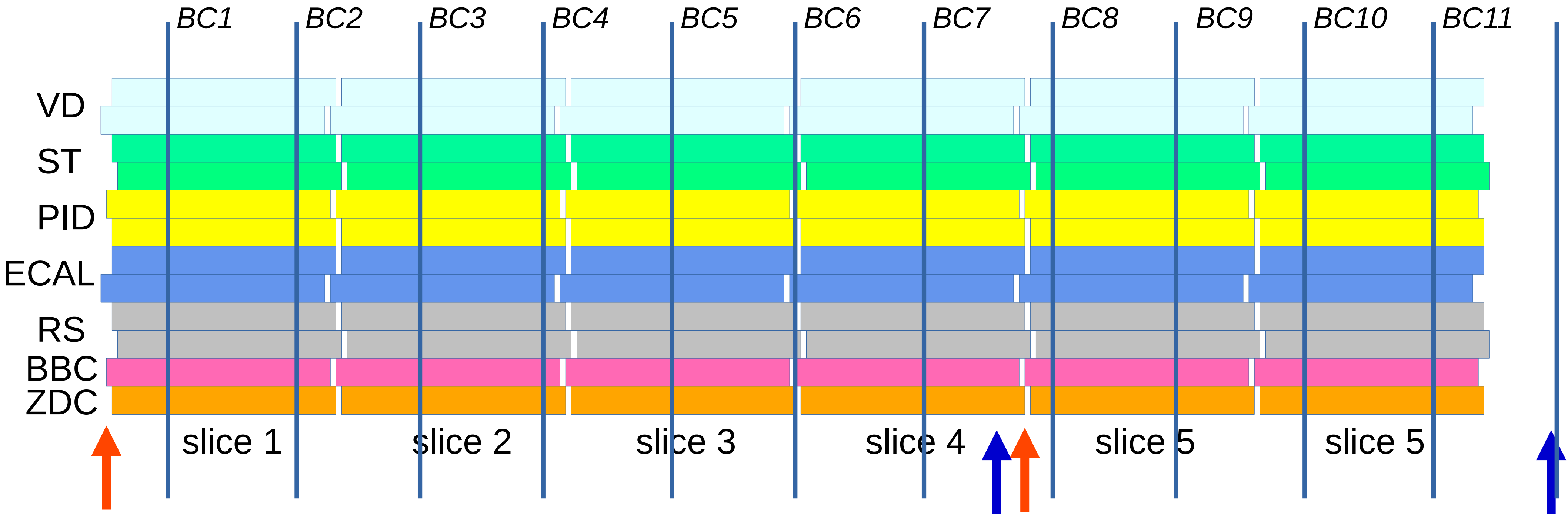
For the moment Anna works on a MPI-based prototype of the payload application and HDF5 output

Input data

- Vertex detector: pixel_id = x,y,z (or fired strips at the first stage)
- Straw tracker: straw_id, TDC counts (STOP of drift time), ADC counts
- ECAL: cell_id, TDC counts, ADC counts
- RS: tube_id, TDC counts (STOP of drift time)
- Bunch crossing time or BCID (precision of about 1 ns — we hope!)
- Other detectors (PID, ZDC, BBC?…) are not used
- Precise calibration data are not available

Data structure

in each chunk (file)



Data blocks of 5-15 us (depending on track multiplicity and reconstruction efficiency) are suggested as an information unit for data processing

Data block reconstruction workflow

- **Tracking in the vertex detector**
  - Vertices
  - Track seeds

- **Tracking in the straw tracker**
  - T0s (crude, ~10 ns) → BCID
  - Tracks
  - Unassociated straw hits

- **ECAL**
  - Clusters
  - Pi0 candidates

- **RS reconstruction**
  - Clusters
  - Muon candidates (?)

- **Association of tracks, RS and ECAL clusters to vertices (event building)**
- **Copy raw data from PID, BBC, ZDC to events according to BCID**

Vertex detector

- Pixel_id → pixel xyz (lookup table?)
- Need to extract:
  - Vertices (x,y,z)
  - Track seeds associated with vertices
- LOOT network? Need a prototype
- At the first stage (w/o MAPS):
  - Space point reconstruction (with fake hits)
  - TrackNet_v3 (prototype exists for BESIII-CGEM, needs to be adapted for SPD)
  - Vertices??

Straw tracker

- Straw_id → R, ϕ, stereoangle (lookup table? Or neural network can be trained to work directly with straw_ids?)
- Find straw hits (fired wires) compatible with tracks, starting from track seeds in VD. Only wire position is used.
- Modified TrackNet_v3? Need a prototype
- For each track candidate find most appropriate BCID, using measured drift time (STOP) and generic RT dependence
- Track fit (something faster than Kalman filter??)
- Collect unassociated hits in a time window of ~200 ns and attach it to the events from the given data block, for offline processing

ECAL

- Hitmap should be translated to a set of clusters
- Each cluster should be identified as a pi0 candidate or not
- CNN looks a good start. Dimitrije and Andrey are working on a prototype (*more details in the next talk*)
- The performance measurement would be extremely interesting to see

RS

- Hitmap should be translated to a set of clusters
- Each cluster should be identified as a muon candidate or not
- Prototype is being developed by Georgy and Igor
- The performance measurement would be extremely interesting to see

Event building

- For each data block
  - Take a list of vertices
  - Associate tracks with each vertex
  - Determine BCID for each vertex
  - Associate ECAL and RS hits with each vertex (by BCID)
  - Attach unassociated straw hits in a selected time window according to BCID
  - Attach raw data from other subdetectors accoring to BCID
  - Call the block of information associated with each vertex an event
  - Store reconstructed events

Urgent issues for the TDR (online filter)

- Simulation of a continuous data stream
- Fast reconstruction algorithms and their performance
- Event unscrambling procedure
- Event selection procedure and criteria

Summary

- The online filter is a very important player for the SPD data processing
- Design of the online computing system exists. A prototyping is under way.
- We urgently need at least an alpha version of machine learning algorithms
- We urgently need the performance estimate for the TDR