Online data filtering for the SPD experiment

Alexey Zhemchugov
(JINR)
on behalf of SPD Collaboration

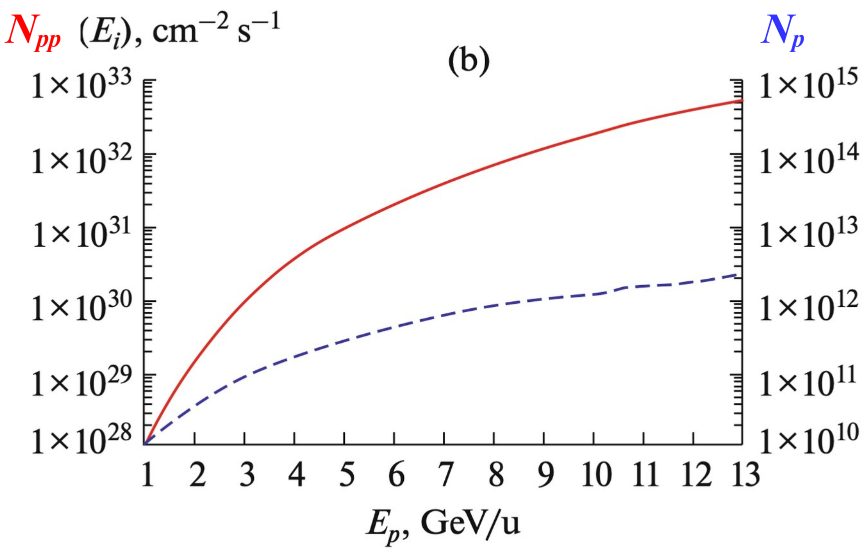
04 July 2023

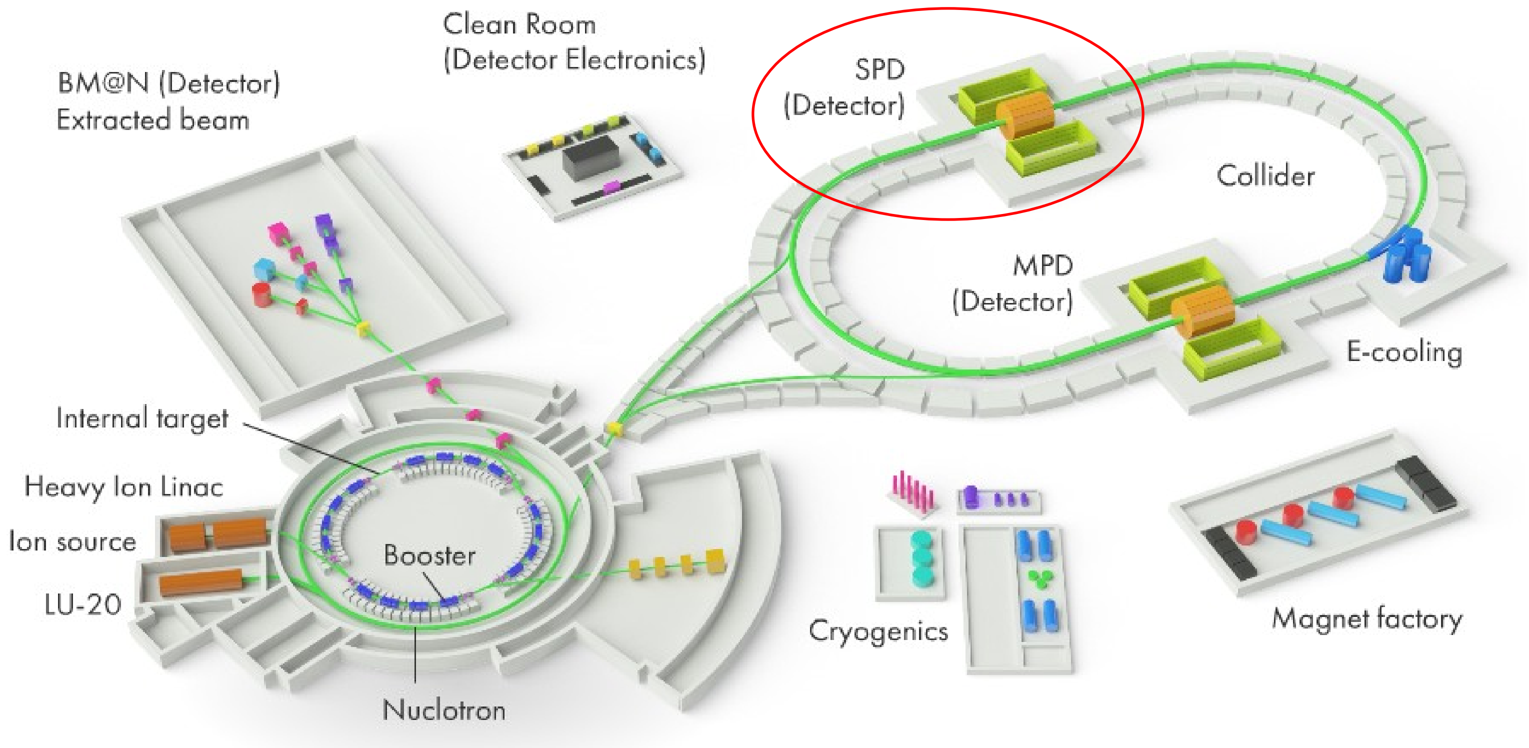
The 10th International Conference "Distributed Computing and Grid Technologies in Science and Education" (GRID'2023)

SPD at NICA

$p\uparrow p\uparrow : \sqrt{s} \leq 27$ GeV
$d\uparrow d\uparrow : \sqrt{s} \leq 13.5$ GeV
$d\uparrow p\uparrow : \sqrt{s} \leq 19$ GeV

U, L, T
|P|>70%



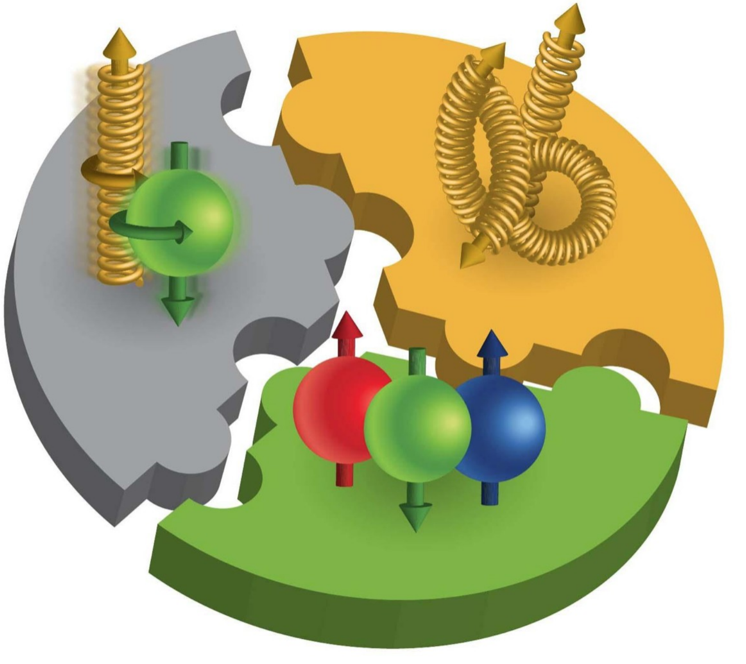


Alexey Zhemchugov on behalf of SPD Collaboration

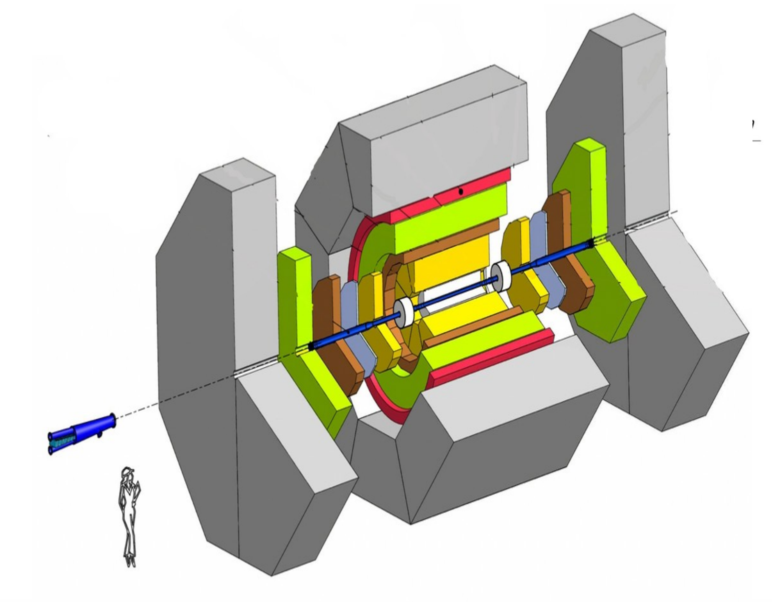
Physics program

- SPD - a universal facility for comprehensive study of gluon content in proton and deuteron at large x
  - Prompt photons
  - Charmonia
  - Open charm
- Other spin-related phenomena
- Other physics

More details:
Prog.Part.Nucl.Phys. 119 (2021) 103858
arXiv:2011.15005



SPD as a data source

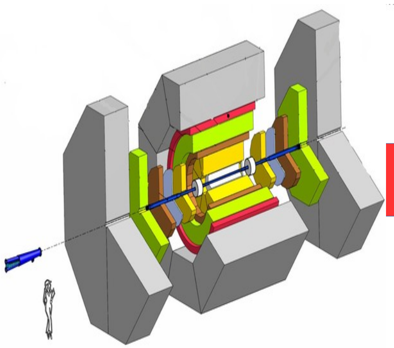


- Bunch crossing every 76.3 ns = crossing rate 13 MHz
- ~ 3 MHz event rate (at 10^{32} $cm^{-2}s^{-1}$ design luminosity)
- 20 GB/s (or 200 PB/year (raw data), $3*10^{13}$ events/year)
- Selection of physics signal requires momentum and vertex reconstruction → no simple trigger is possible

The SPD detector is a medium scale setup in size, but a large scale one in data rate!

Data workflow

SPD Hall



DAQ

Online Filter
Monitoring
DQA

Tier-0: JINR Computing Center

Reco

Worldwide Grid

Analysis

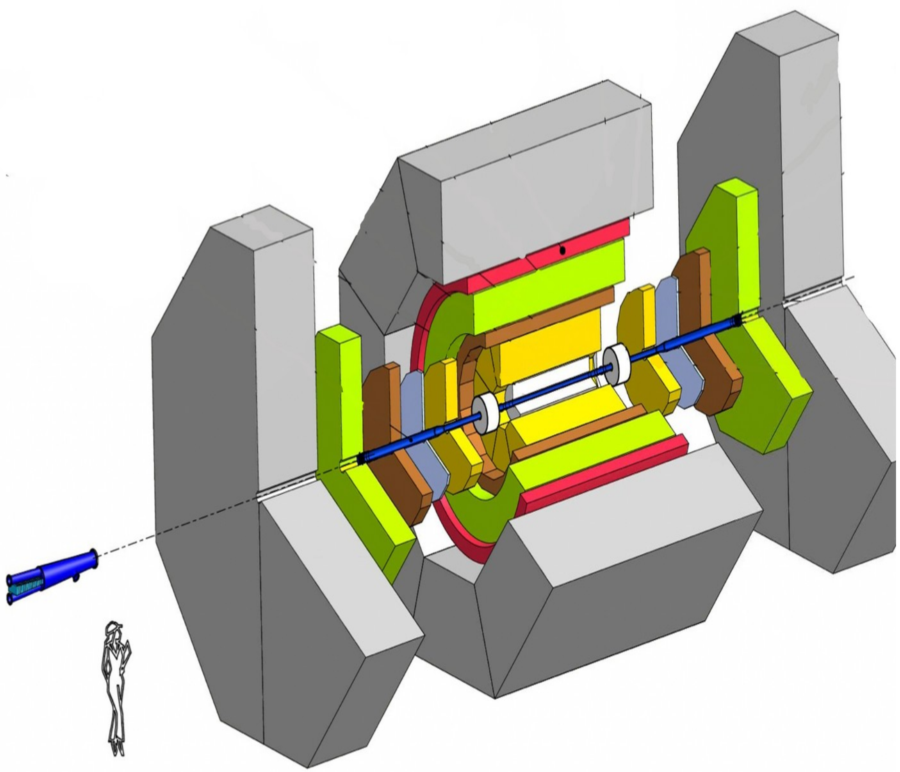
Reco

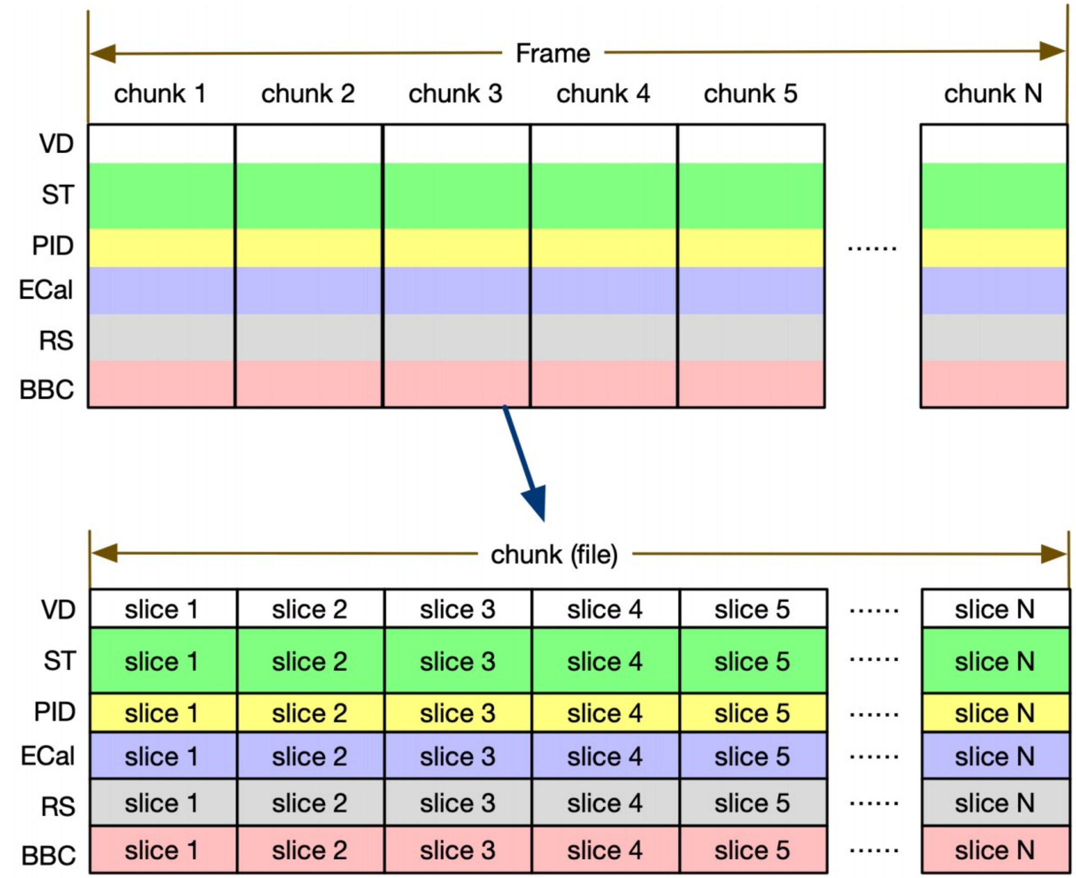
MC

Continuous data reduction:

- DAQ: noise suppression
- Online filter: event building, partial reconstruction, software trigger
- Offline computing: data analysis and long term storage

Free running DAQ





Input data structure

No trigger = No classical events anymore

Primary data unit: **time slice** (1 us — 8.3 ms)

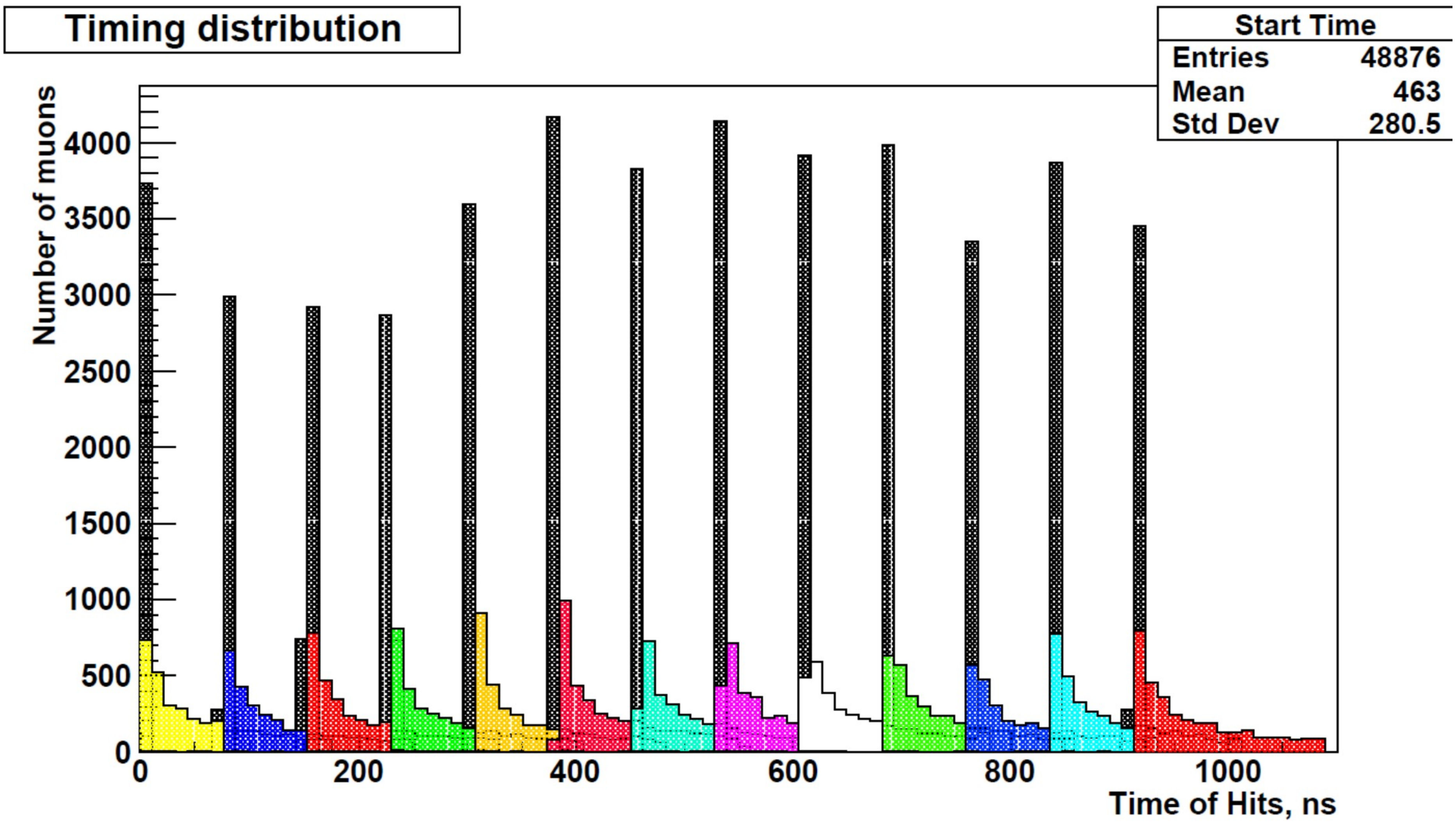
Time slices combined in **time frames** (up to 549 s, 16 GB max, < 160 MB to fullfil 20 GB/s limit)

Intermediate units — **time chunks** of 0.1-0.2 s (2-4 GB or ~10^5-10^6 events) are being discussed now

Every time slices will contain signals from a few to many collisions (events)

Event building have to unscramble events from a series of time slices.

Example: simulated hits in the SPD straw tracker



Online Data Filter

High-performance heterogeneous computing cluster

- Partial reconstruction
  - Fast tracking and vertex reconstruction
  - Fast ECAL clustering
  - Fast RS clustering
- Event unscrambling
- Software trigger
  - several data streams

Machine learning is a key technology

Control of systematics?

- Monitoring and Data quality assessment
- Local polarimetry

Reconstruction workflow

- **Tracking in the vertex detector (at the second stage)**
  - Vertices
  - Track seeds

- **Tracking in the straw tracker (+ MCT at the first stage)**
  - T0s (crude, ~10 ns) → bunch crossing time
  - Tracks
  - Unassociated straw hits

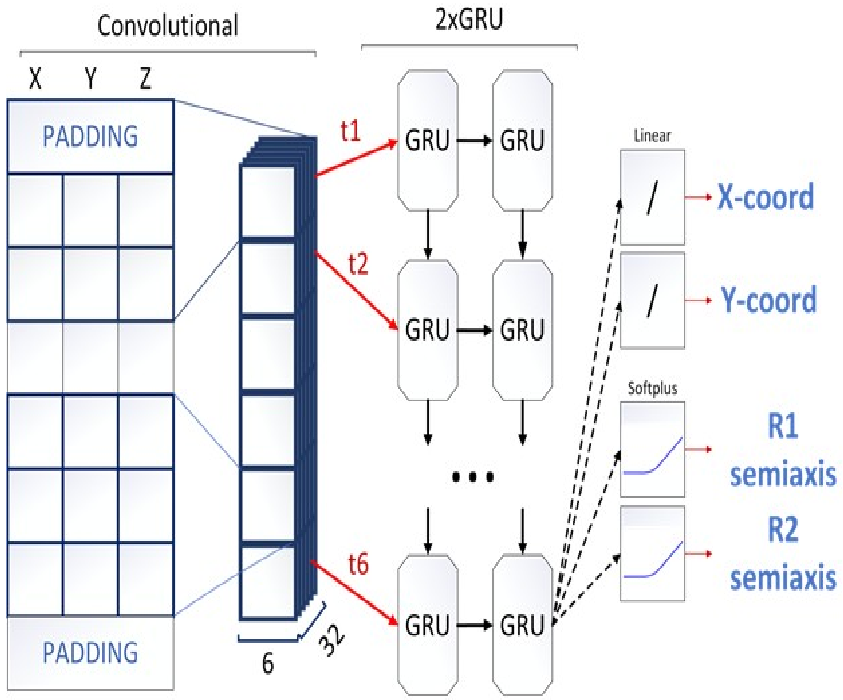
- **ECAL reconsruction**
  - Clusters
  - π^0 candidates

- **RS reconstruction**
  - Clusters
  - Muon candidates

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

Example: TrackNETv3 for track recognition

JINST 17 (2022) 12, P12023
D. Rusov et al, talk at PCT'2023



- Network predicts an area at the next detector layer where to search for the track continuation
- If continuation is found the hit is added to the track candidate and the procedure repeats again
- Essentially reproduces the idea of the Kalman filter: track parameters are predicted by synaptic weights determined by network training
- Generalization? Stability?

	Time slices of 40 events
Track efficiency (recall) (%)	96,54
Track purity (precision) (%)	94.75
Time slices / sec	63.74 (*40 = 2549.6)

PRELIMINARY

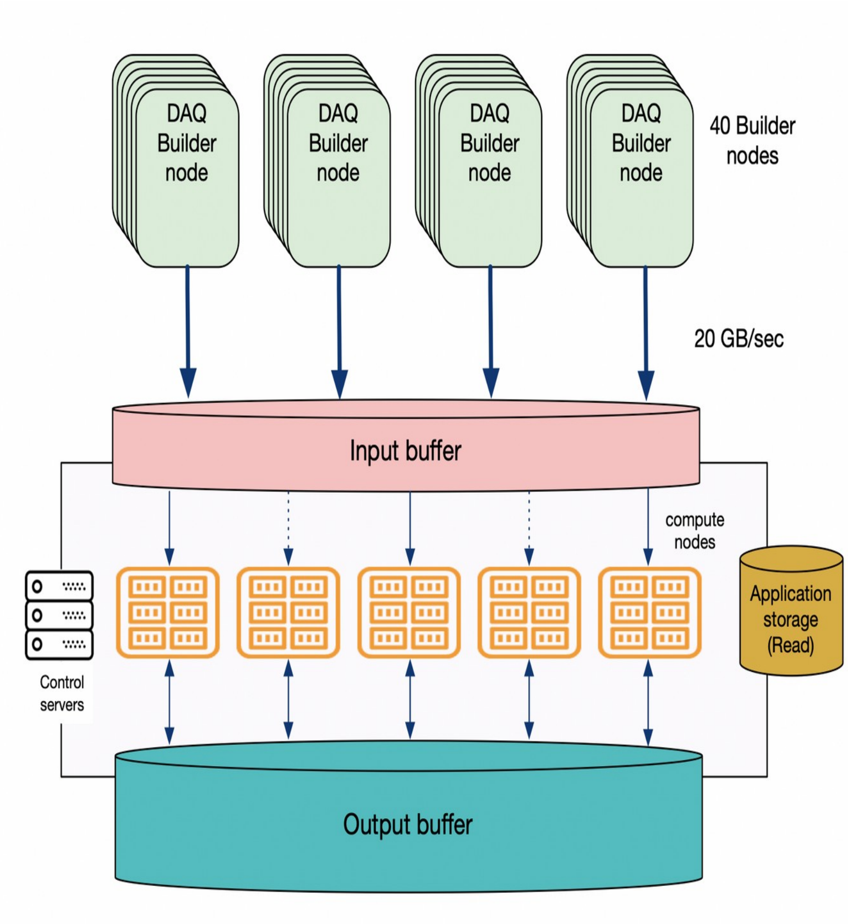
Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz + GPU Nvidia V100 32Gb

Event unscrambling

For each time slice

- Reconstruct tracks and associate them with vertices
- Determine bunch crossing time for each vertex
- Associate ECAL and RS hits with each vertex (by timestamp)
- Attach unassociated tracker hits in a selected time window according to bunch crossing time
- Attach raw data from other subdetectors accoring to bunch crossing time
- Call the block of information associated with each vertex an event
- Store reconstructed events

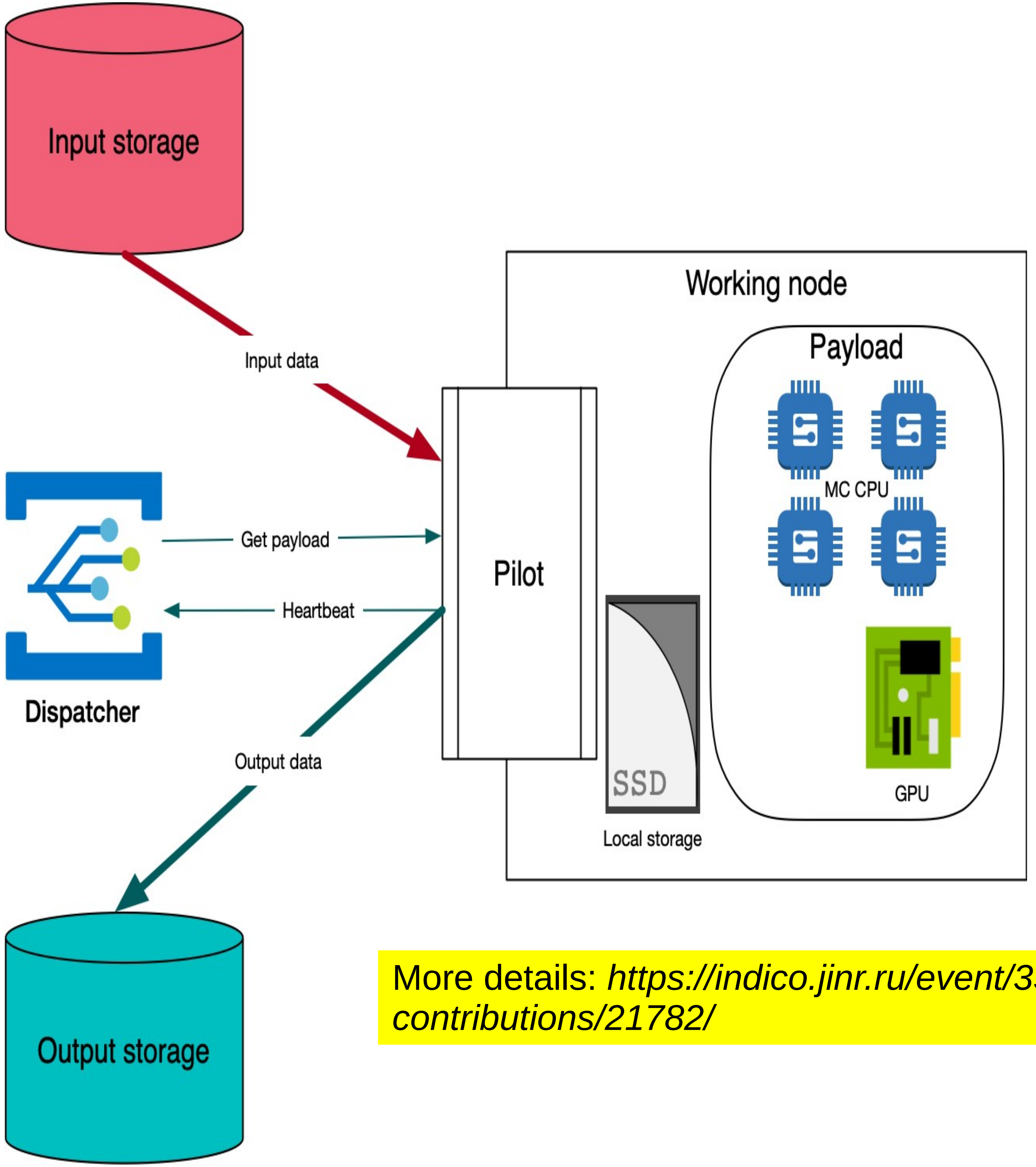
The online filter design



- A high-throughput computing system
- Around 40 DAQ builder nodes running simultaneously produce aggregated data flow to the input buffer up to 20 GB/s.
- Working nodes will read raw data from the input buffer. Intermediate and fully processed data will be placed on output buffer.
- The number of working nodes, which will simultaneously read data from input buffer, may be much higher than the number of builders.
- The number of WNs depends on the performance of reconstruction algorithms.

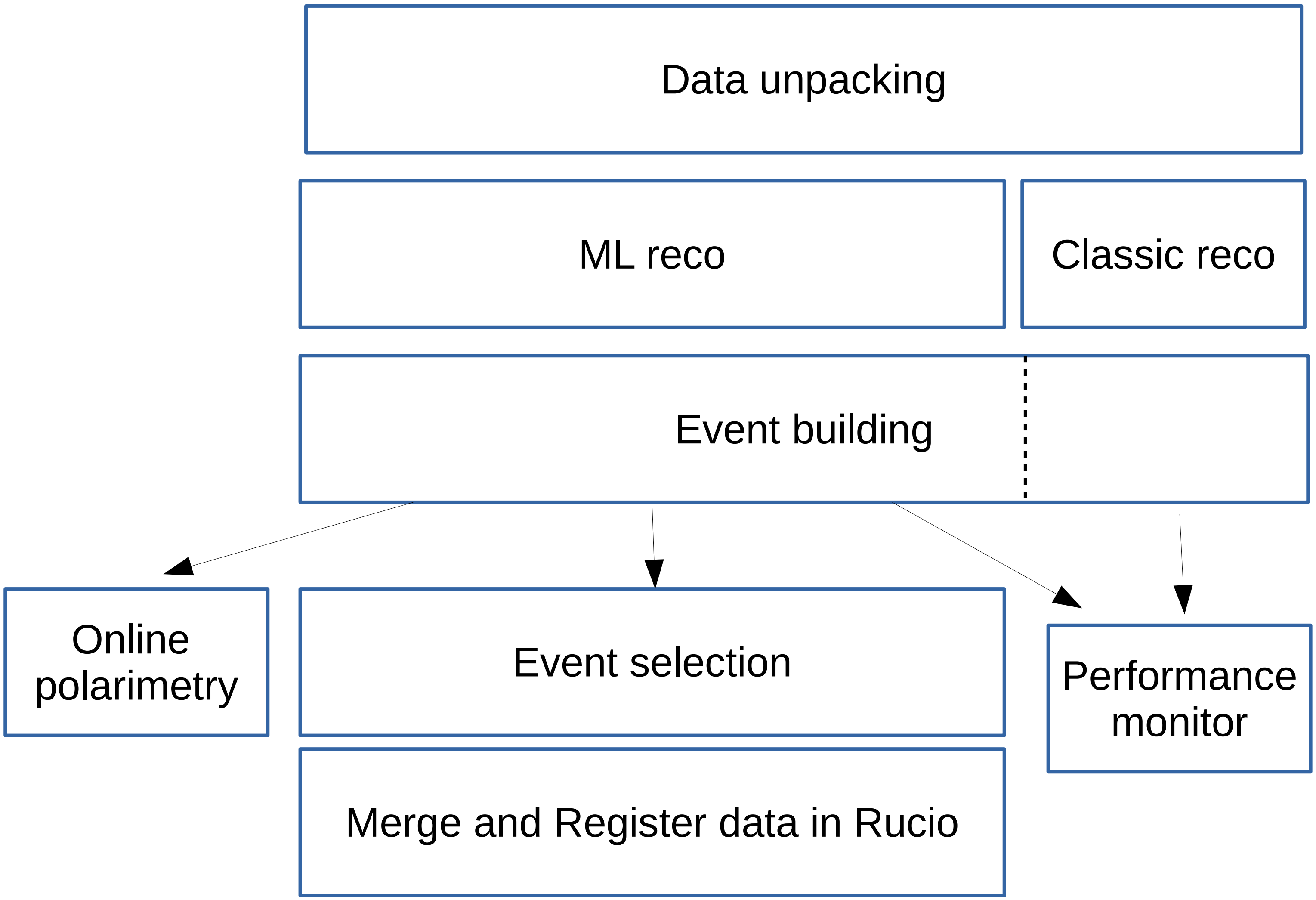
We should foresee using these computing resources for offline data processing between the data taking campaigns

More details: *https://indico.jinr.ru/event/3505/contributions/21778/*



More details: *https://indico.jinr.ru/event/3505/contributions/21782/*

The payload

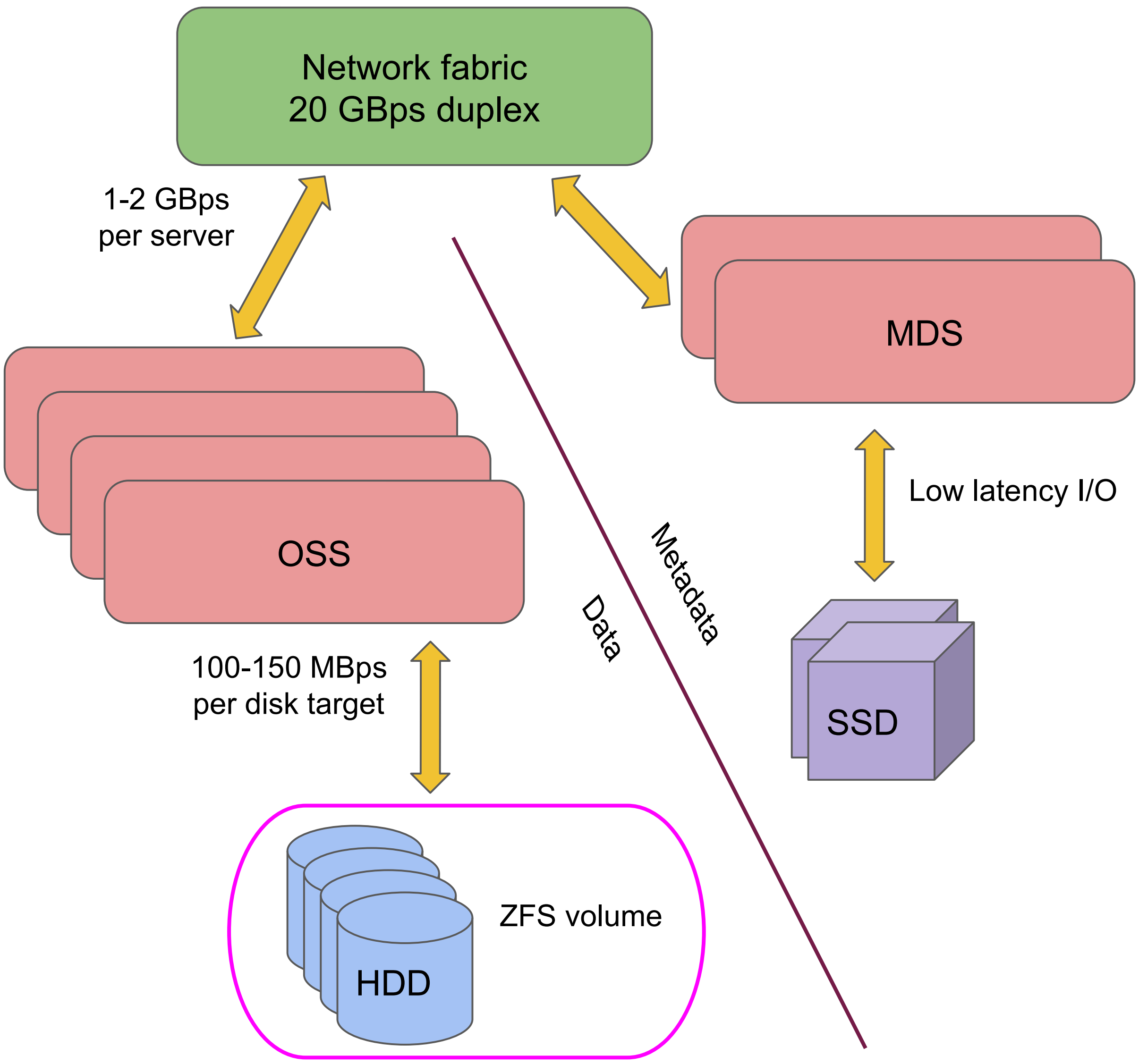

Data unpacking
ML reco
Classic reco
Event building
Online polarimetry
Event selection
Performance monitor
Merge and Register data in Rucio

Input buffer requirements

- 2 PB - one full day of data taking
- 20 GBps I/O
- Mountable as a file system from multiple hosts
- Reliable and redundant
- Avoid expensive hardware
- …
- Lustre on ZFS!

Proposed hardware:

- Commodity server with redundant power
- Several 10/25 Gbps network ports
- SAS HBA (no RAID)
- 256 GB RAM
- >=16 SAS disk drive bays



Simulation of the online filter

Input data

Hardware architecture and parameters.

Characteristics of data flows and tasks.

Functionality

Data processing system design.

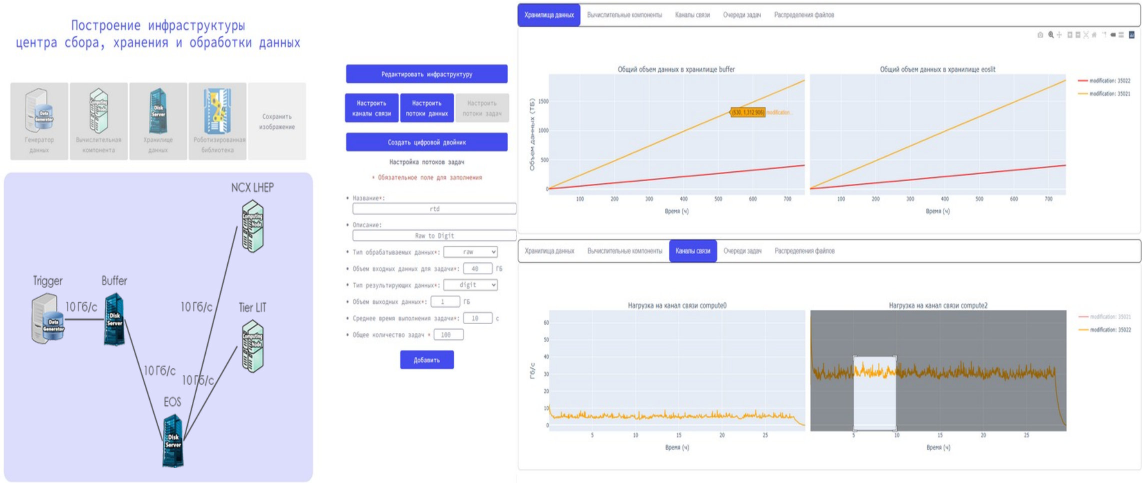
Performance and reliability analysis.

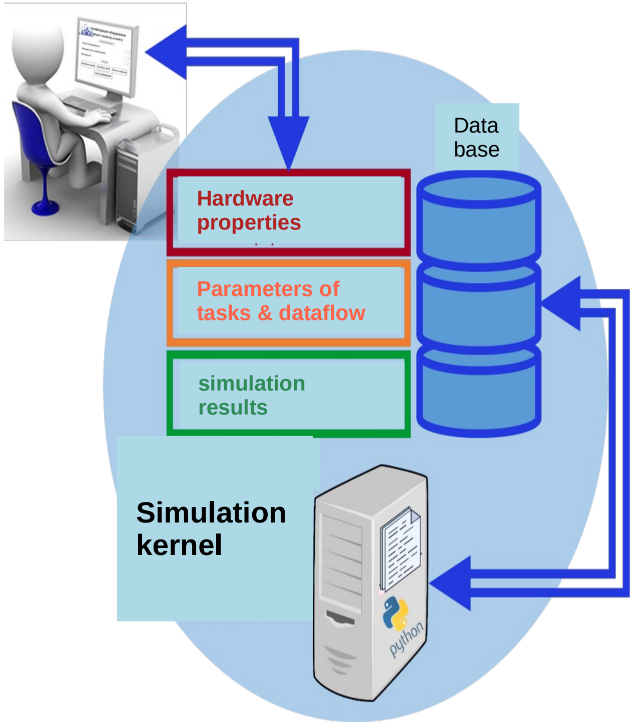
Testing of scaling scenarios, taking into account the requirements for data flows and tasks.

Estimating the amount of resources needed for specific tasks.

Validation of task flow management strategies.

Software package for creating digital twins of distributed data collection, storage and processing centers





For the SPD experiment:

- Online Filter design
- Storage optimization
- CPU resource estimate and optimization
- Data transfer simulation and network optimization

HTTPS://INDICO.JINR.RU/EVENT/3505/CONTRIBUTIONS/21792/

Summary

- Efficient online filter is a key to the success of SPD data processing
- The design of the SPD online filter is presented
- Machine learning is the only way to reconstruct physics events fast enough and to keep the size of the online filter reasonably small
- Control of systematics is essential: reliable methods to control ML performance are needed
- Software prototyping, hardware testbed and cluster simulation are under way