SPD OnLine Filter.

Status update

VIII SPD Collaboration Meeting. 07.11.2024
Oleynik Danila.

SPD Online filter

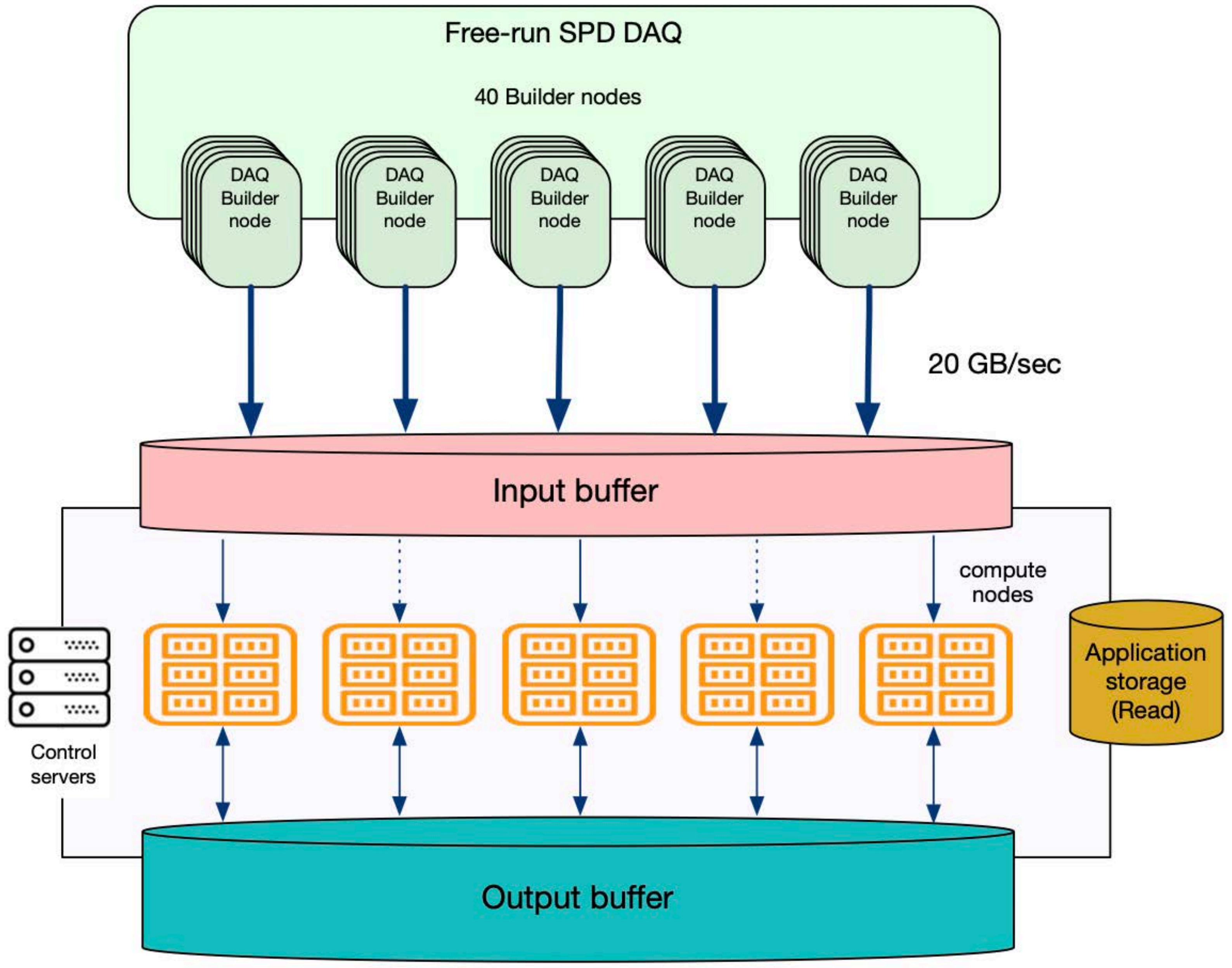
Reminder :-)

Online filter is the first stage in data processing chain for SPD Experiment (right after DAQ)

Main goals:

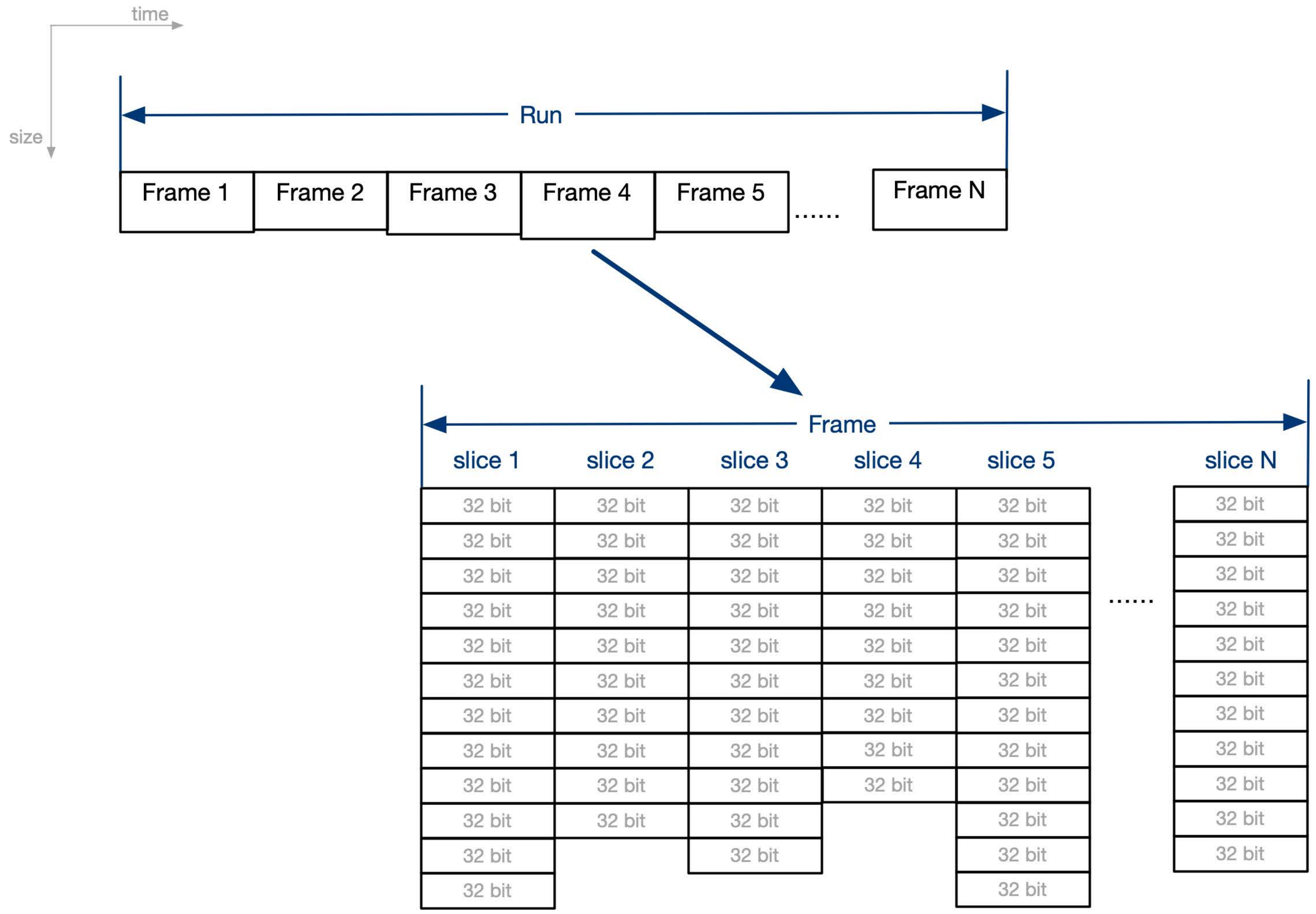
  - Events unscrambling through partial reconstruction
  - Software trigger, which essentially is event filter

- SPD Online Filter is a high performance computing system for high throughput processing
  - **Hardware component:** compute cluster with two storage systems and set of working nodes: multi-CPU and hybrid multi CPU + Neural network accelerators (GPU, FPGA etc.)
  - **Middleware component:** software complex for management of multistep data processing and efficient loading (usage) of computing facility.
  - **Applied software**: performs informational processing of data



Initial data

- Free run DAQ, means that the output of the system will not be a dataset of raw events, but a set of signals from detectors organized in time slices
- Primary data unit: time slice (~10 μs)
  Time slices combined in time frames (1-10 sec.)
- Every slice will contain signals from a few to many collisions (events)
- Event building have to unscramble events from a series of time slices



time

size

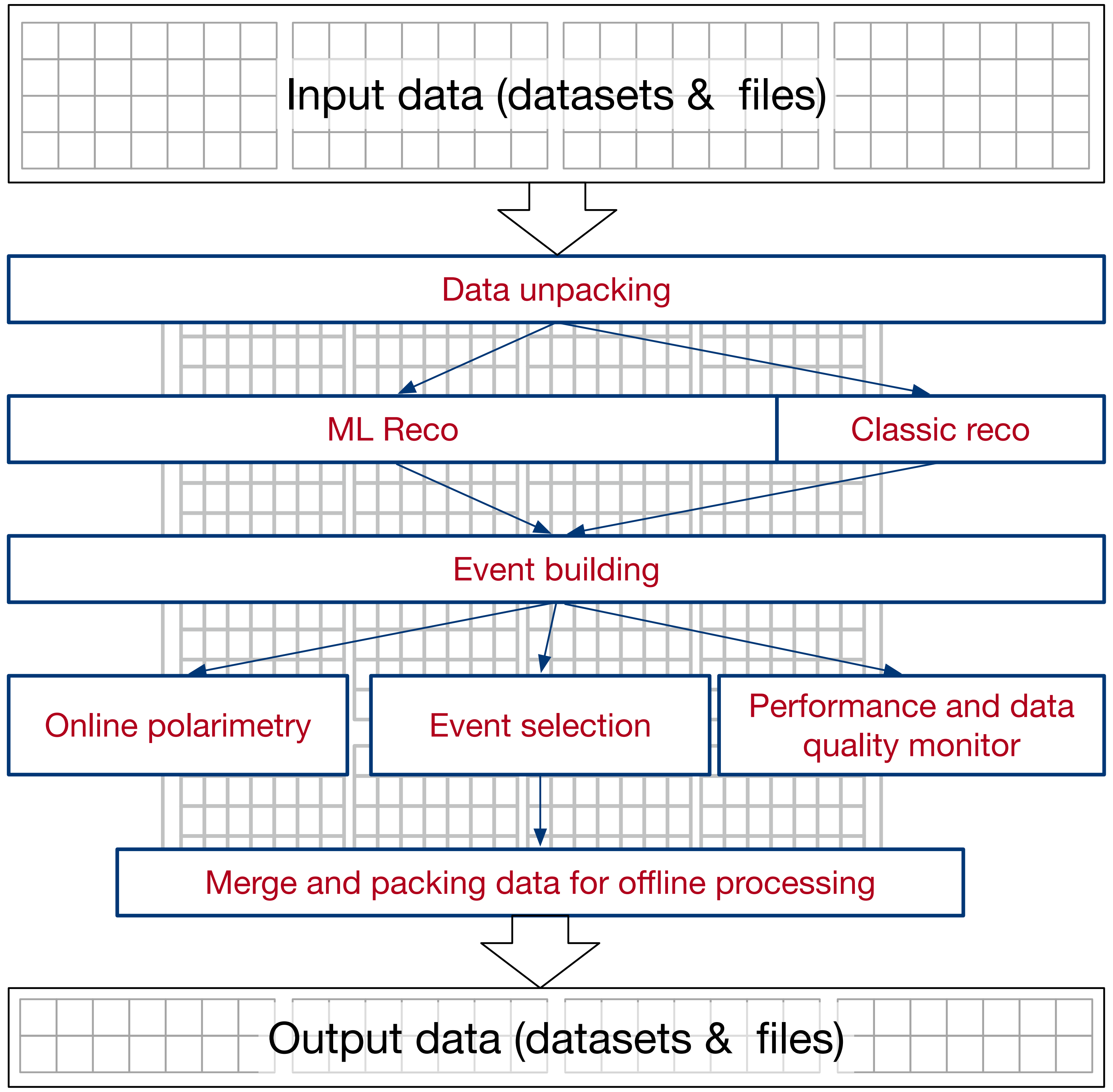
Run

Frame 1 | Frame 2 | Frame 3 | Frame 4 | Frame 5 | ...... | Frame N

Frame

slice 1	slice 2	slice 3	slice 4	slice 5		slice N
32 bit	32 bit	32 bit	32 bit	32 bit		32 bit
32 bit	32 bit	32 bit	32 bit	32 bit		32 bit
32 bit	32 bit	32 bit	32 bit	32 bit		32 bit
32 bit	32 bit	32 bit	32 bit	32 bit	......	32 bit
32 bit	32 bit	32 bit	32 bit	32 bit		32 bit
32 bit	32 bit	32 bit	32 bit	32 bit		32 bit
32 bit	32 bit	32 bit	32 bit	32 bit		32 bit
32 bit	32 bit	32 bit	32 bit	32 bit		32 bit
32 bit	32 bit	32 bit	32 bit	32 bit		32 bit
32 bit	32 bit	32 bit		32 bit		32 bit
32 bit		32 bit		32 bit		32 bit
32 bit				32 bit		

Data processing


Input data (datasets & files)
Data unpacking
ML Reco
Classic reco
Event building
Online polarimetry
Event selection
Performance and data quality monitor
Merge and packing data for offline processing
Output data (datasets & files)

- **DATA PROCESSING** – the execution of a systematic sequence of operations, performed with data, e.g. handling, merging, sorting, computing.
  - Note: Where data processing is performed in order to increase the value or significance (from a certain point of view) of the information conveyed by the data, it may be called **INFORMATION PROCESSING**.

THE TERMINOLOGY WORK OF IFIP (International Federation for Information Processing) AND ICC (International Computing Centre)
I. H. GOULD and G. C. TOOTILL

Infrastructure

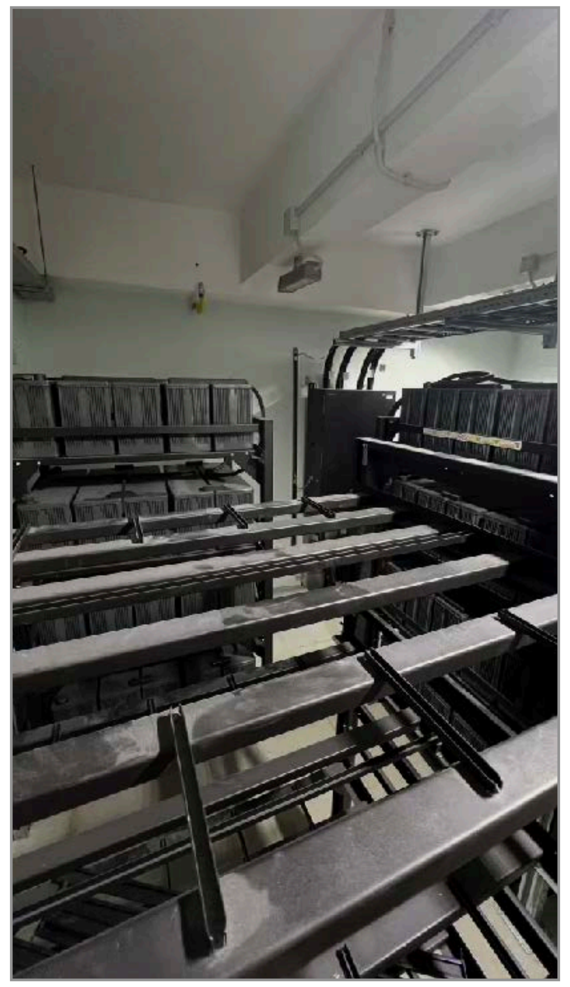
Current situation

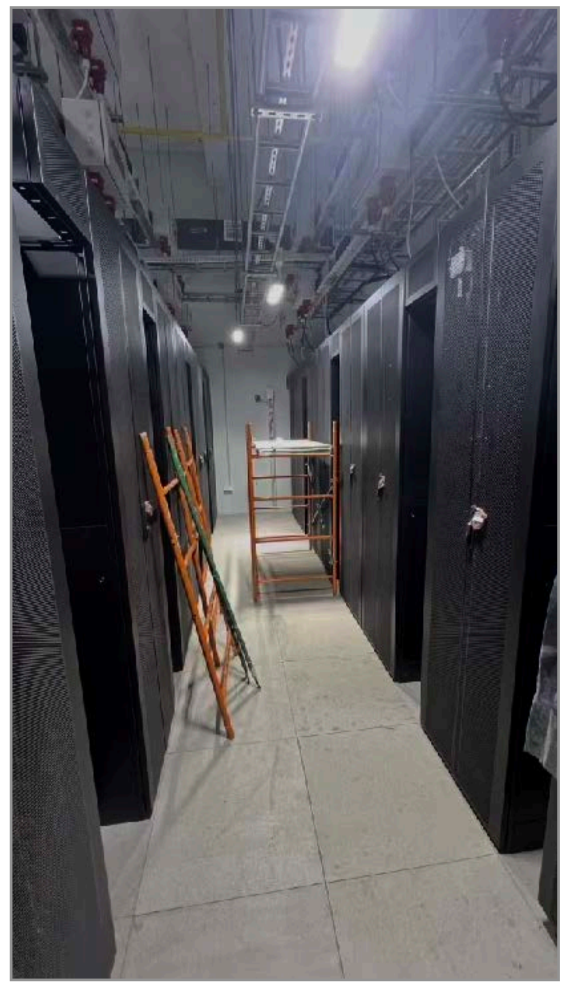
- Integration infrastructure deployed on JINR Cloud resources on 6 VM’s (currently)
    - 4 VM for debugging of general services components
    - 1 - imitation of storages
    - 1 - Dedicated DBMS Server with PostgreSQL 16
    - Rabbit MQ dedicated facility are going to deploy soon
- AlmaLinux 9 - base OS. It will work for us next few years, security support till the middle of 2032
- GIT based CI/CD functionality provided by JINR LIT (Cloud infrastructure)

Infrastructure

Hardware for prototyping (NICA Computing)

- Building 14 almost ready to host a NICA Computing data center
- In progress purchasing of some compute and storage facility





Compute unit	Storage unit	GPU unit	Control units
2 CPU * 44 Core (FMA3, AVX512) 1024 GB RAM 6 * 4 TB NVMe - 24 TB Ethernet: 100 Gb/s InfiniBand: 200 Gb/s	2 CPU * 44 Core (FMA3, AVX512) 512 GB RAM 32 * 30,72 TB NVMe - 983 TB InfiniBand: 2 * 200 Gb/s	2 CPU * 48 Core (FMA3, AVX512) 1024 GB RAM 8 GPU (NVIDIA H100) 4 * 4 TB NVMe - 16 TB per node InfiniBand: 2 * 200 Gb/s	2 CPU * 44 Core (FMA3, AVX512) 512 GB RAM 2 * 4 TB NVMe Ethernet: 100 Gb/s InfiniBand: 200 Gb/s

- 32 Compute units -> **2816 Cores with ~10GB RAM per core**
- 2 Storage units -> **1,8 PB Storage raw volume**
- 1 GPU unit -> **8 NVIDIA H100 GPU**

Middleware

Data management;

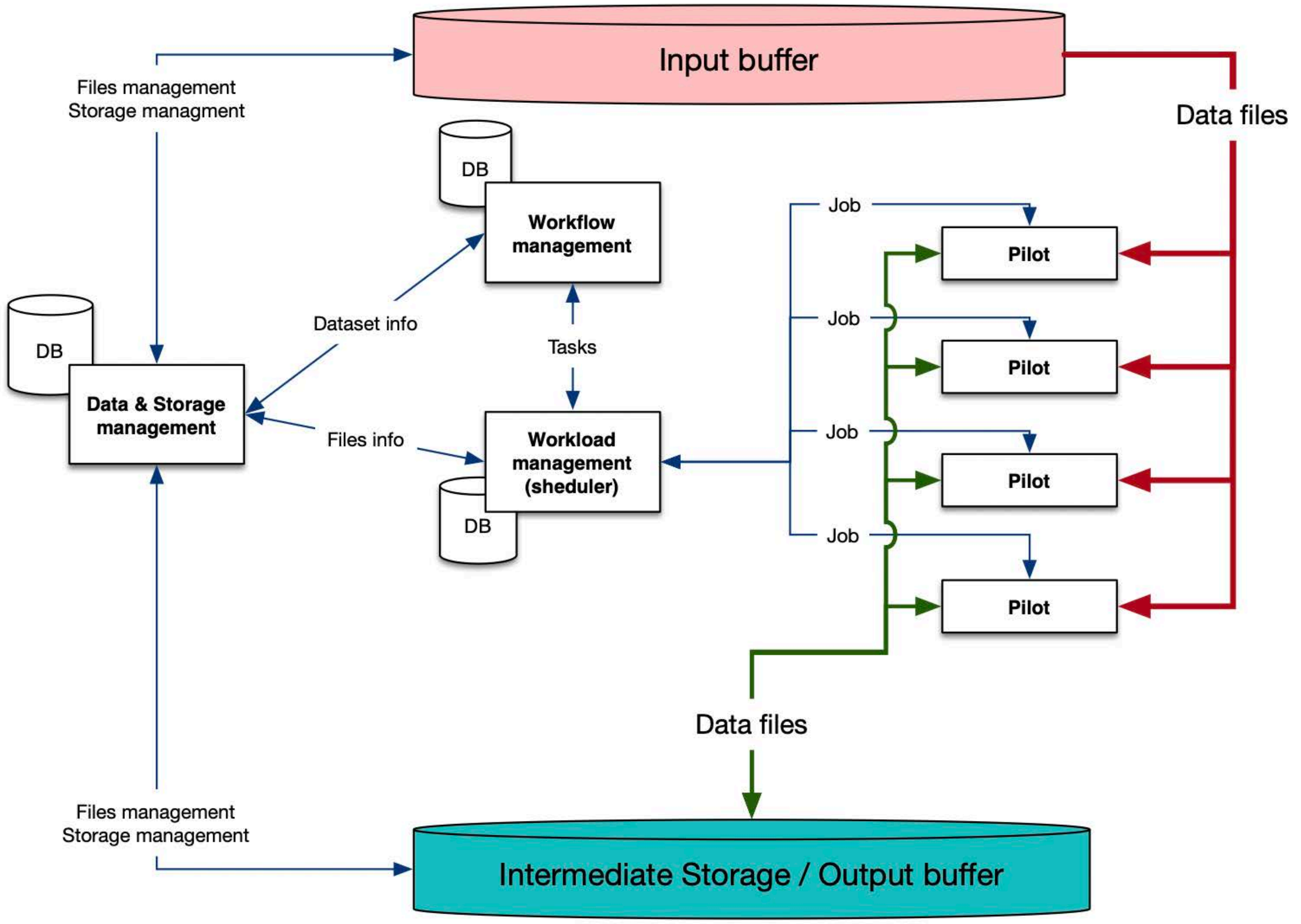
- *Support of data life-cycle and storage usage;*

Workflow management;

- *Definition of processing chains;*
- *Realisation of processing chains as set of computations tasks;*
- *Management of tasks execution;*

Workload management:

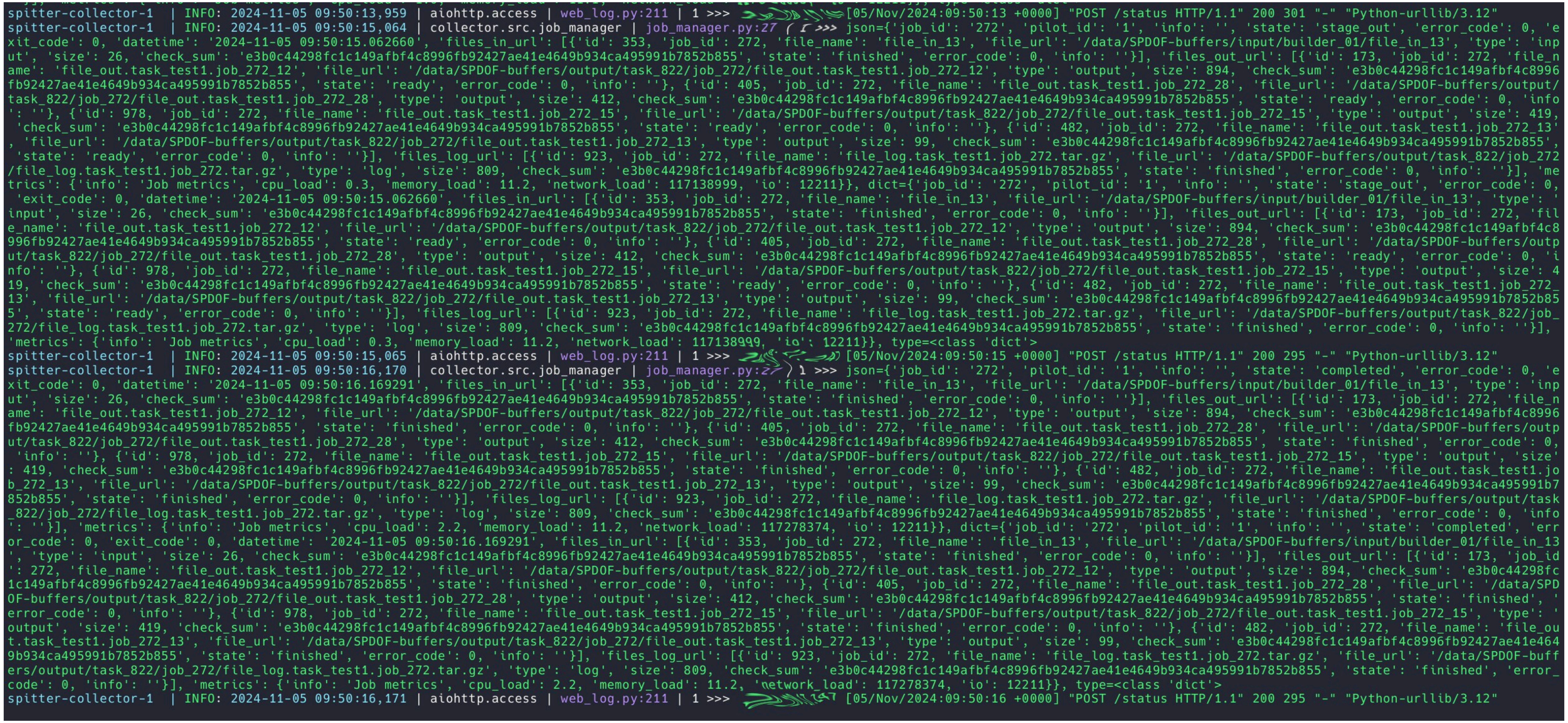
- *Generation of required number of processing jobs for performing of task;*
- *Control of jobs executions through pilots, which works on compute nodes;*



- Manpower: one PHD students (SPbSU) now hired by LIT, one full time researcher (JINR LIT), two master students (MEPHI) - hired by LIT

Middleware

active development, more details in Nikita’s report



```
spitter-collector-1  | INFO: 2024-11-05 09:50:13,959 | aiohttp.access | web_log.py:211 | 1 >>> [redacted][05/Nov/2024:09:50:13 +0000] "POST /status HTTP/1.1" 200 301 "-" "Python-urllib/3.12"
spitter-collector-1  | INFO: 2024-11-05 09:50:15,064 | collector.src.job_manager | job_manager.py:27 [redacted] >>> json={'job_id': '272', 'pilot_id': '1', 'info': '', 'state': 'stage_out', 'error_code': 0, 'e
xit_code': 0, 'datetime': '2024-11-05 09:50:15.062660', 'files_in_url': [{'id': 353, 'job_id': 272, 'file_name': 'file_in_13', 'file_url': '/data/SPDOF-buffers/input/builder_01/file_in_13', 'type': 'inp
ut', 'size': 26, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}], 'files_out_url': [{'id': 173, 'job_id': 272, 'file_n
ame': 'file_out.task_test1.job_272_12', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_12', 'type': 'output', 'size': 894, 'check_sum': 'e3b0c44298fc1c149afbf4c8996
fb92427ae41e4649b934ca495991b7852b855', 'state': 'ready', 'error_code': 0, 'info': ''}, {'id': 405, 'job_id': 272, 'file_name': 'file_out.task_test1.job_272_28', 'file_url': '/data/SPDOF-buffers/output/
task_822/job_272/file_out.task_test1.job_272_28', 'type': 'output', 'size': 412, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'ready', 'error_code': 0, 'info
': ''}, {'id': 978, 'job_id': 272, 'file_name': 'file_out.task_test1.job_272_15', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_15', 'type': 'output', 'size': 419,
 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'ready', 'error_code': 0, 'info': ''}, {'id': 482, 'job_id': 272, 'file_name': 'file_out.task_test1.job_272_13'
, 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_13', 'type': 'output', 'size': 99, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855',
 'state': 'ready', 'error_code': 0, 'info': ''}], 'files_log_url': [{'id': 923, 'job_id': 272, 'file_name': 'file_log.task_test1.job_272.tar.gz', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272
/file_log.task_test1.job_272.tar.gz', 'type': 'log', 'size': 809, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}], 'me
trics': {'info': 'Job metrics', 'cpu_load': 0.3, 'memory_load': 11.2, 'network_load': 117138999, 'io': 12211}}, dict={'job_id': '272', 'pilot_id': '1', 'info': '', 'state': 'stage_out', 'error_code': 0,
 'exit_code': 0, 'datetime': '2024-11-05 09:50:15.062660', 'files_in_url': [{'id': 353, 'job_id': 272, 'file_name': 'file_in_13', 'file_url': '/data/SPDOF-buffers/input/builder_01/file_in_13', 'type': '
input', 'size': 26, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}], 'files_out_url': [{'id': 173, 'job_id': 272, 'fil
e_name': 'file_out.task_test1.job_272_12', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_12', 'type': 'output', 'size': 894, 'check_sum': 'e3b0c44298fc1c149afbf4c8
996fb92427ae41e4649b934ca495991b7852b855', 'state': 'ready', 'error_code': 0, 'info': ''}, {'id': 405, 'job_id': 272, 'file_name': 'file_out.task_test1.job_272_28', 'file_url': '/data/SPDOF-buffers/outp
ut/task_822/job_272/file_out.task_test1.job_272_28', 'type': 'output', 'size': 412, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'ready', 'error_code': 0, 'i
nfo': ''}, {'id': 978, 'job_id': 272, 'file_name': 'file_out.task_test1.job_272_15', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_15', 'type': 'output', 'size': 4
19, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'ready', 'error_code': 0, 'info': ''}, {'id': 482, 'job_id': 272, 'file_name': 'file_out.task_test1.job_272_
13', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_13', 'type': 'output', 'size': 99, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b85
5', 'state': 'ready', 'error_code': 0, 'info': ''}], 'files_log_url': [{'id': 923, 'job_id': 272, 'file_name': 'file_log.task_test1.job_272.tar.gz', 'file_url': '/data/SPDOF-buffers/output/task_822/job_
272/file_log.task_test1.job_272.tar.gz', 'type': 'log', 'size': 809, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}],
 'metrics': {'info': 'Job metrics', 'cpu_load': 0.3, 'memory_load': 11.2, 'network_load': 117138999, 'io': 12211}}, type=<class 'dict'>
spitter-collector-1  | INFO: 2024-11-05 09:50:15,065 | aiohttp.access | web_log.py:211 | 1 >>> [redacted] [05/Nov/2024:09:50:15 +0000] "POST /status HTTP/1.1" 200 295 "-" "Python-urllib/3.12"
spitter-collector-1  | INFO: 2024-11-05 09:50:16,170 | collector.src.job_manager | job_manager.py:2[redacted] 1 >>> json={'job_id': '272', 'pilot_id': '1', 'info': '', 'state': 'completed', 'error_code': 0, 'e
xit_code': 0, 'datetime': '2024-11-05 09:50:16.169291', 'files_in_url': [{'id': 353, 'job_id': 272, 'file_name': 'file_in_13', 'file_url': '/data/SPDOF-buffers/input/builder_01/file_in_13', 'type': 'inp
ut', 'size': 26, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}], 'files_out_url': [{'id': 173, 'job_id': 272, 'file_n
ame': 'file_out.task_test1.job_272_12', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_12', 'type': 'output', 'size': 894, 'check_sum': 'e3b0c44298fc1c149afbf4c8996
fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}, {'id': 405, 'job_id': 272, 'file_name': 'file_out.task_test1.job_272_28', 'file_url': '/data/SPDOF-buffers/outp
ut/task_822/job_272/file_out.task_test1.job_272_28', 'type': 'output', 'size': 412, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0,
 'info': ''}, {'id': 978, 'job_id': 272, 'file_name': 'file_out.task_test1.job_272_15', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_15', 'type': 'output', 'size'
: 419, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}, {'id': 482, 'job_id': 272, 'file_name': 'file_out.task_test1.jo
b_272_13', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_13', 'type': 'output', 'size': 99, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7
852b855', 'state': 'finished', 'error_code': 0, 'info': ''}], 'files_log_url': [{'id': 923, 'job_id': 272, 'file_name': 'file_log.task_test1.job_272.tar.gz', 'file_url': '/data/SPDOF-buffers/output/task
_822/job_272/file_log.task_test1.job_272.tar.gz', 'type': 'log', 'size': 809, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info
': ''}], 'metrics': {'info': 'Job metrics', 'cpu_load': 2.2, 'memory_load': 11.2, 'network_load': 117278374, 'io': 12211}}, dict={'job_id': '272', 'pilot_id': '1', 'info': '', 'state': 'completed', 'err
or_code': 0, 'exit_code': 0, 'datetime': '2024-11-05 09:50:16.169291', 'files_in_url': [{'id': 353, 'job_id': 272, 'file_name': 'file_in_13', 'file_url': '/data/SPDOF-buffers/input/builder_01/file_in_13
', 'type': 'input', 'size': 26, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}], 'files_out_url': [{'id': 173, 'job_id
': 272, 'file_name': 'file_out.task_test1.job_272_12', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_12', 'type': 'output', 'size': 894, 'check_sum': 'e3b0c44298fc
1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}, {'id': 405, 'job_id': 272, 'file_name': 'file_out.task_test1.job_272_28', 'file_url': '/data/SPD
OF-buffers/output/task_822/job_272/file_out.task_test1.job_272_28', 'type': 'output', 'size': 412, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', '
error_code': 0, 'info': ''}, {'id': 978, 'job_id': 272, 'file_name': 'file_out.task_test1.job_272_15', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_15', 'type': '
output', 'size': 419, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}, {'id': 482, 'job_id': 272, 'file_name': 'file_ou
t.task_test1.job_272_13', 'file_url': '/data/SPDOF-buffers/output/task_822/job_272/file_out.task_test1.job_272_13', 'type': 'output', 'size': 99, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e464
9b934ca495991b7852b855', 'state': 'finished', 'error_code': 0, 'info': ''}], 'files_log_url': [{'id': 923, 'job_id': 272, 'file_name': 'file_log.task_test1.job_272.tar.gz', 'file_url': '/data/SPDOF-buff
ers/output/task_822/job_272/file_log.task_test1.job_272.tar.gz', 'type': 'log', 'size': 809, 'check_sum': 'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'state': 'finished', 'error_
code': 0, 'info': ''}], 'metrics': {'info': 'Job metrics', 'cpu_load': 2.2, 'memory_load': 11.2, 'network_load': 117278374, 'io': 12211}}, type=<class 'dict'>
spitter-collector-1  | INFO: 2024-11-05 09:50:16,171 | aiohttp.access | web_log.py:211 | 1 >>> [redacted] [05/Nov/2024:09:50:16 +0000] "POST /status HTTP/1.1" 200 295 "-" "Python-urllib/3.12"
```

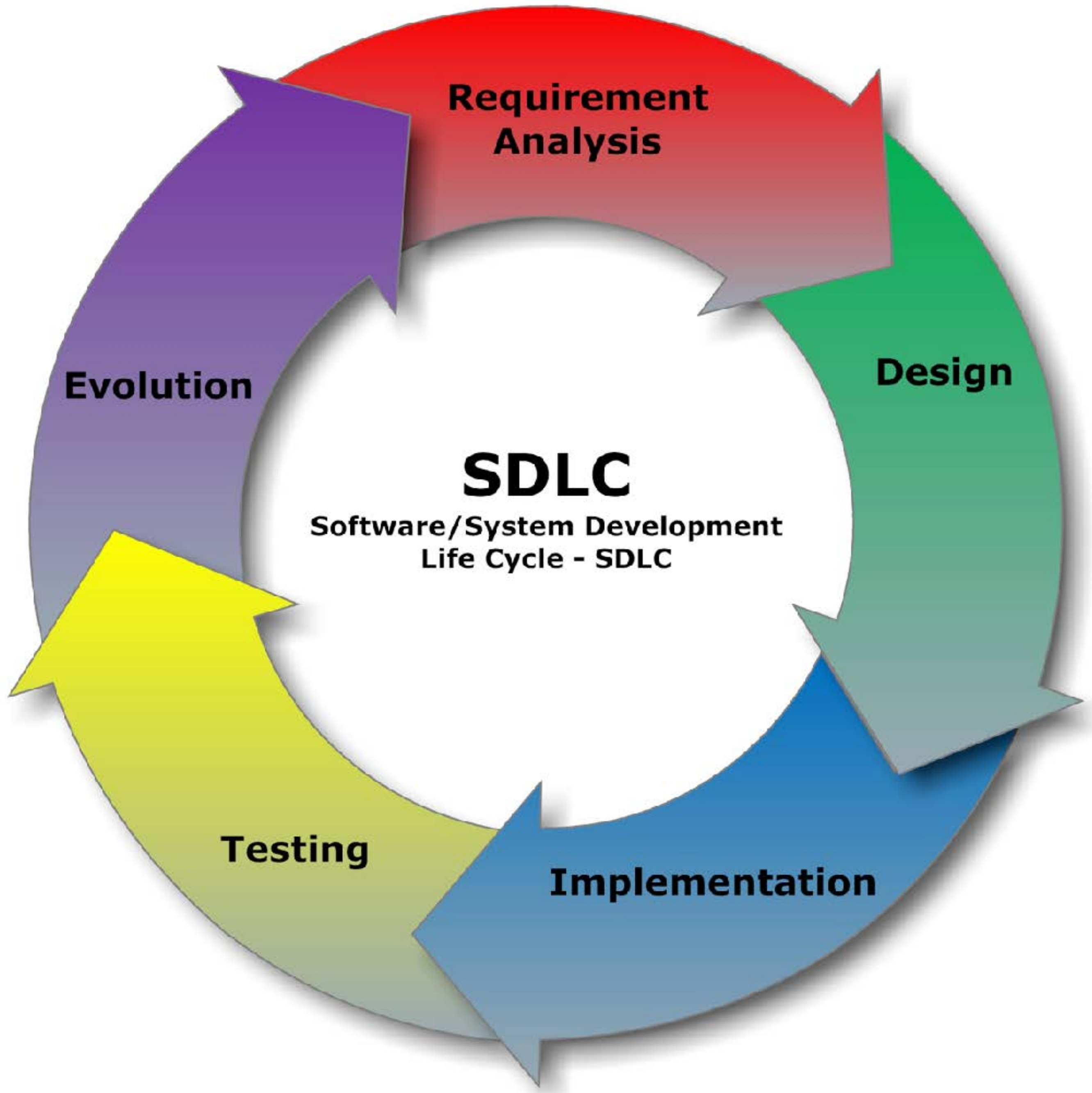
Applied software for Online Filter

- Sampo framework will be used for realisation of applied software for SPD Online filter
  - Initial functional and non-functional requirements should be collected in next couple of months
- ML algorithms was presented at CHEP (Krakow) and MMCP (Yerevan) conferences
  - Integration of ML tools with Sampo will became one of the topics

A bit boring about (any) software

> *“Program testing can be used to show the presence of bugs, but never to show their absence!”*
> *— Edsger W. Dijkstra*

Software errors are flaws that result unexpected outcome from a computer program or system or cause it to act in ways that were not intended. ***They are any mismatches between the software and its requirement specification***.



How software errors are illuminated? Some part of them by testing, others by usage experience.

The only way to fix error(s) are launching next iterate of development cycle

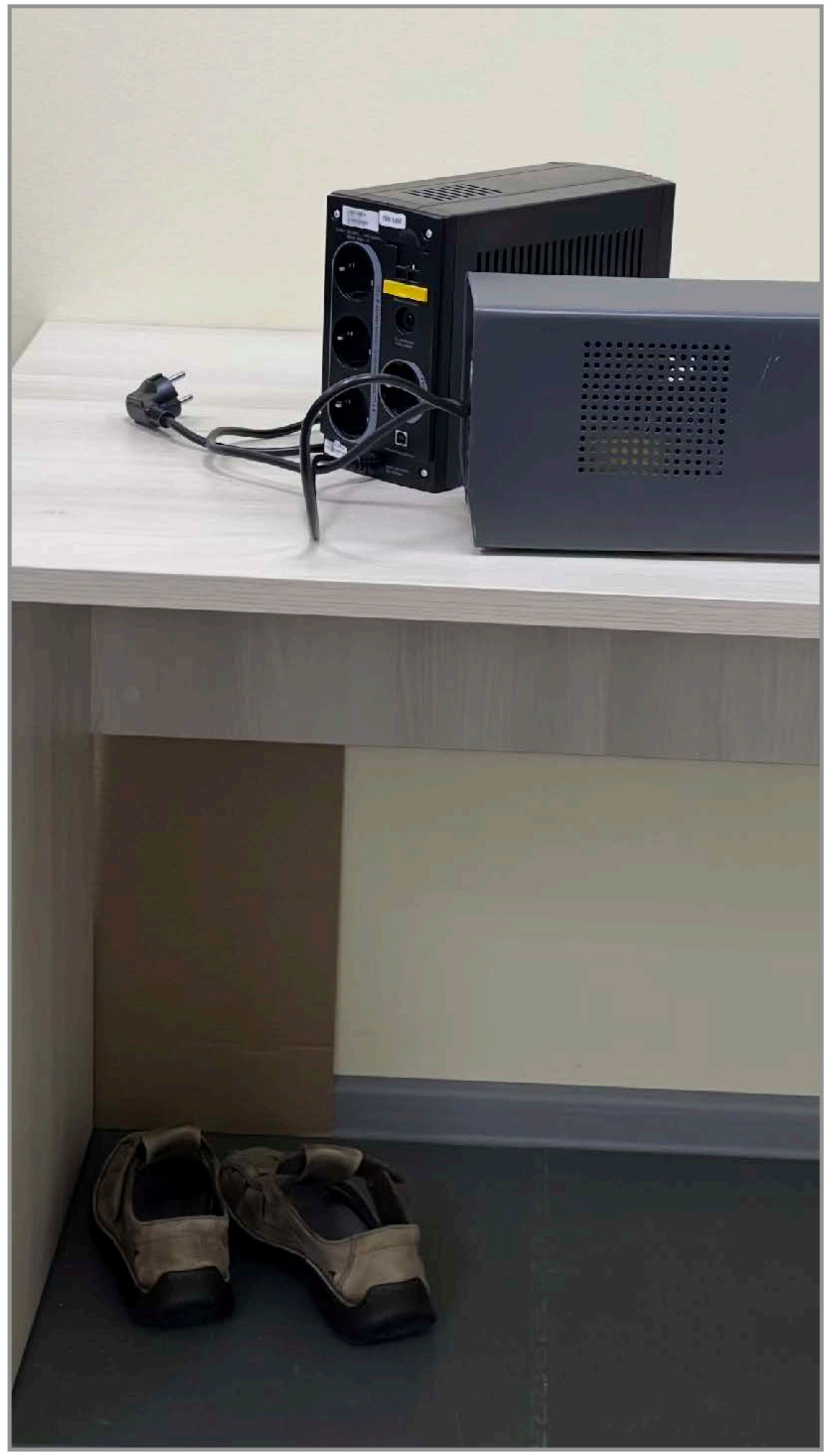
A bit boring about ML usage?

A lot of similarity with software life cycle, is not it?

- In most cases should work, but **mismatches with expected result is unpredictable**
    - so detection of mismatching of results with expected should trigger process of retraining of NN
    - except the situation with poor quality of input data, but in this case we should face similar issues with ‘traditional' methods.
    - *Deep anomaly detection refers to any methods that exploit deep learning techniques to learn feature representations or anomaly scores for anomaly detection.*
- In any case requirements for any ML tools should be **collected and formalised** as much as possible (as for any software)
    - As only we will reach enough formalisation they will be applicable for developing of proper algorithm

DAQ & Online filter testbed in MLIT

- Testbed zone is ready
- Some of habitants (a couple of summer students Andrey and Dmitrii) appeared this summer



Thank you!