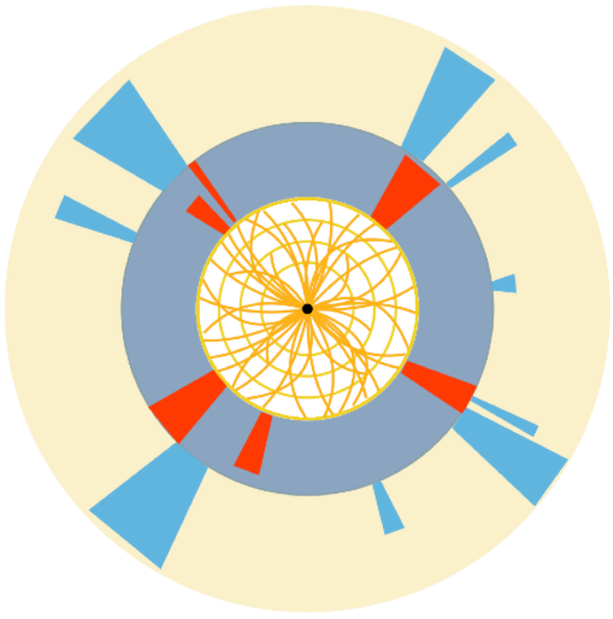
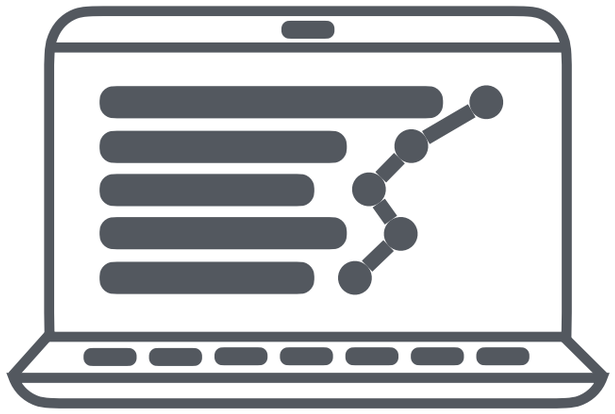
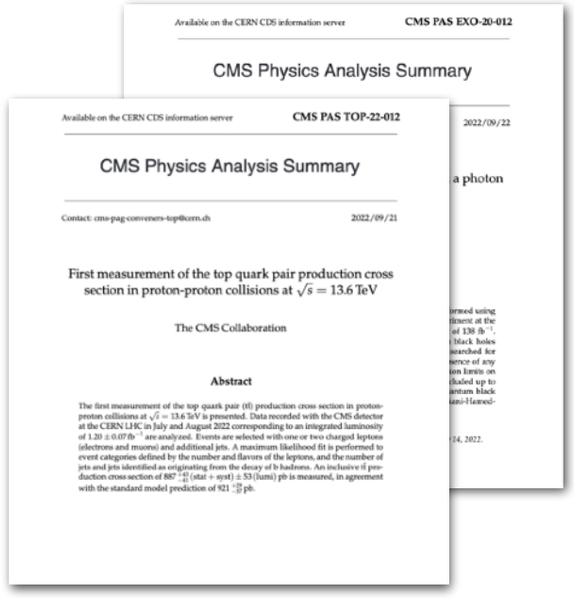




Fast!



Fast Columnar Physics Analyses of Terabyte-Scale LHC Data on a Cache-Aware Dask Cluster (2207.08598)

Niclas Eich, Martin Erdmann, Peter Fackeldey,
Benjamin Fischer, Dennis Noll, Yannik Rath

FIDIUM Meeting

21.10.22

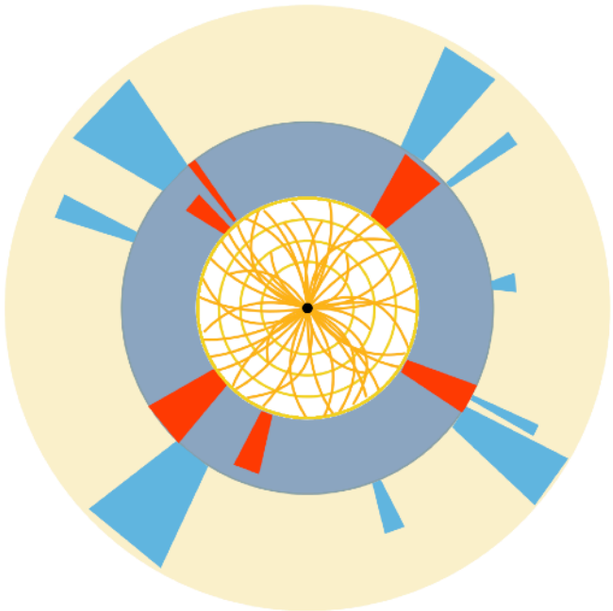
RWTH AACHEN UNIVERSITY

Motivation

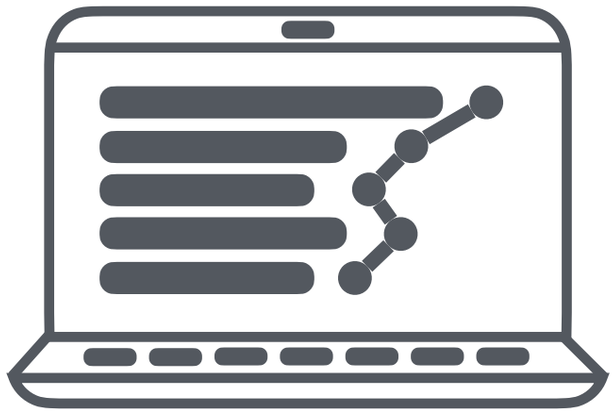
RWTH AACHEN UNIVERSITY
Dennis Noll - 21.10.22

Input

Data ~ O(TB)



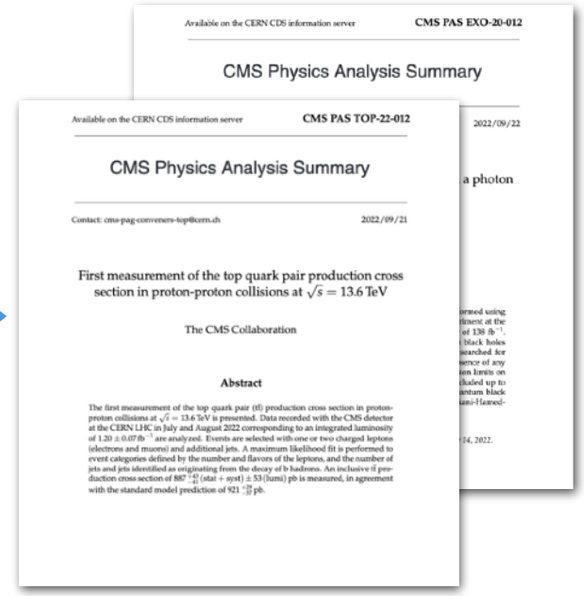
Analysis



Faster = Better

Output

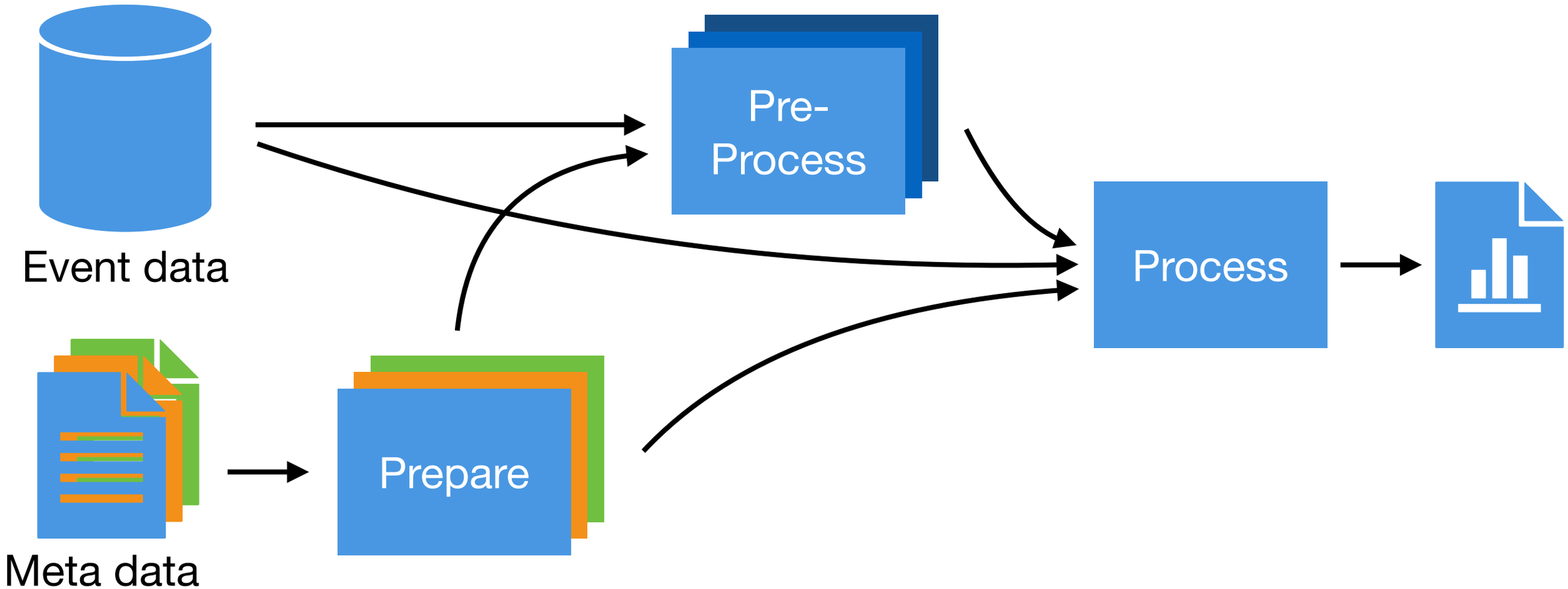
Results ~ O(kB)



- Small time-to-insight drives high physics potential:
  - Tuning of existing methods
  - Investigations of new methods
- Ultimate goal: Analysis in duration of a coffee break

Analysis Structure

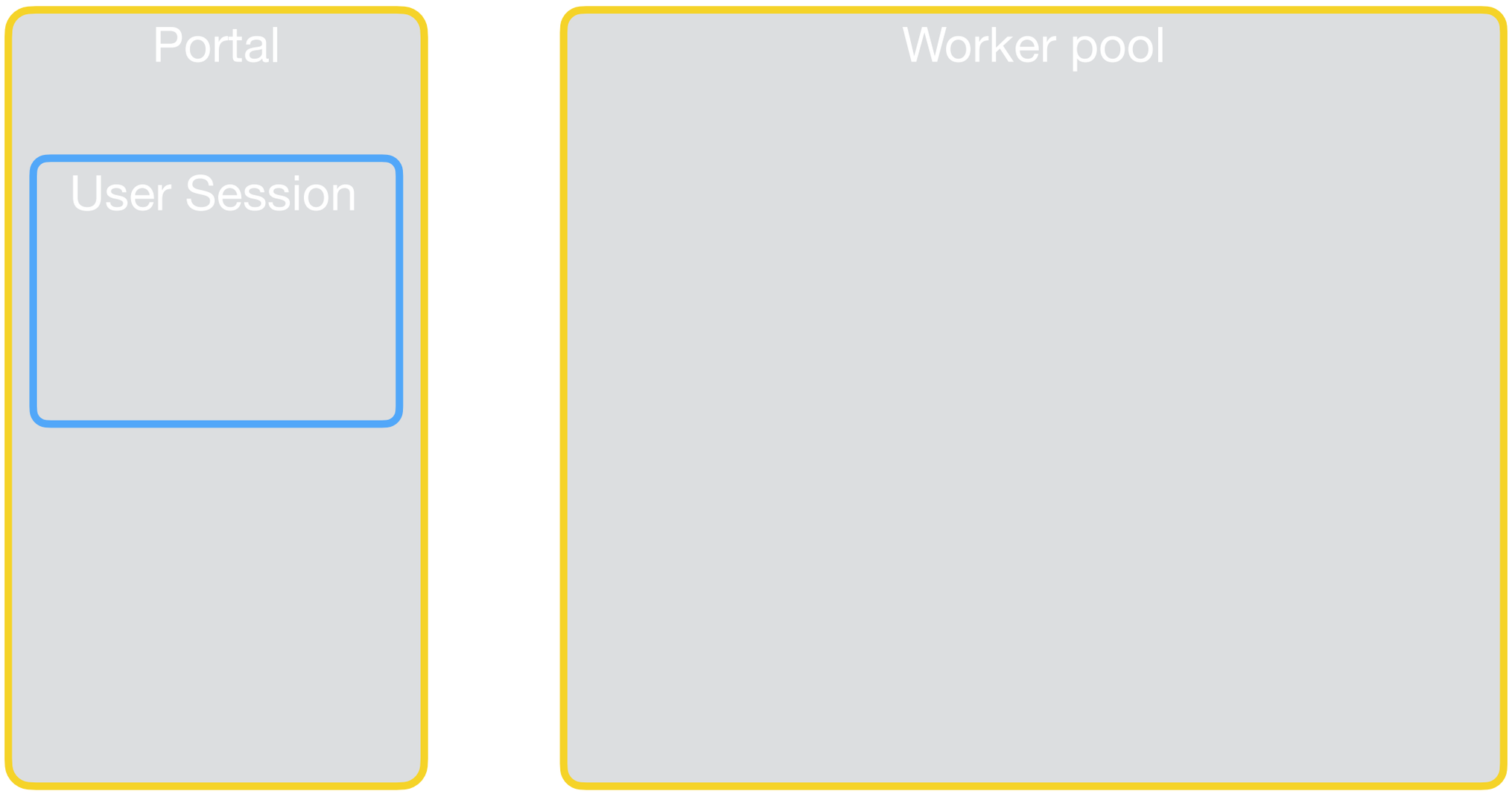
Dennis Noll - 21.10.22



- O(100) different computing tasks
- Workflows: Development, debugging, full analysis, ...
- Repeated processing of event data necessary

Scale Out: Map

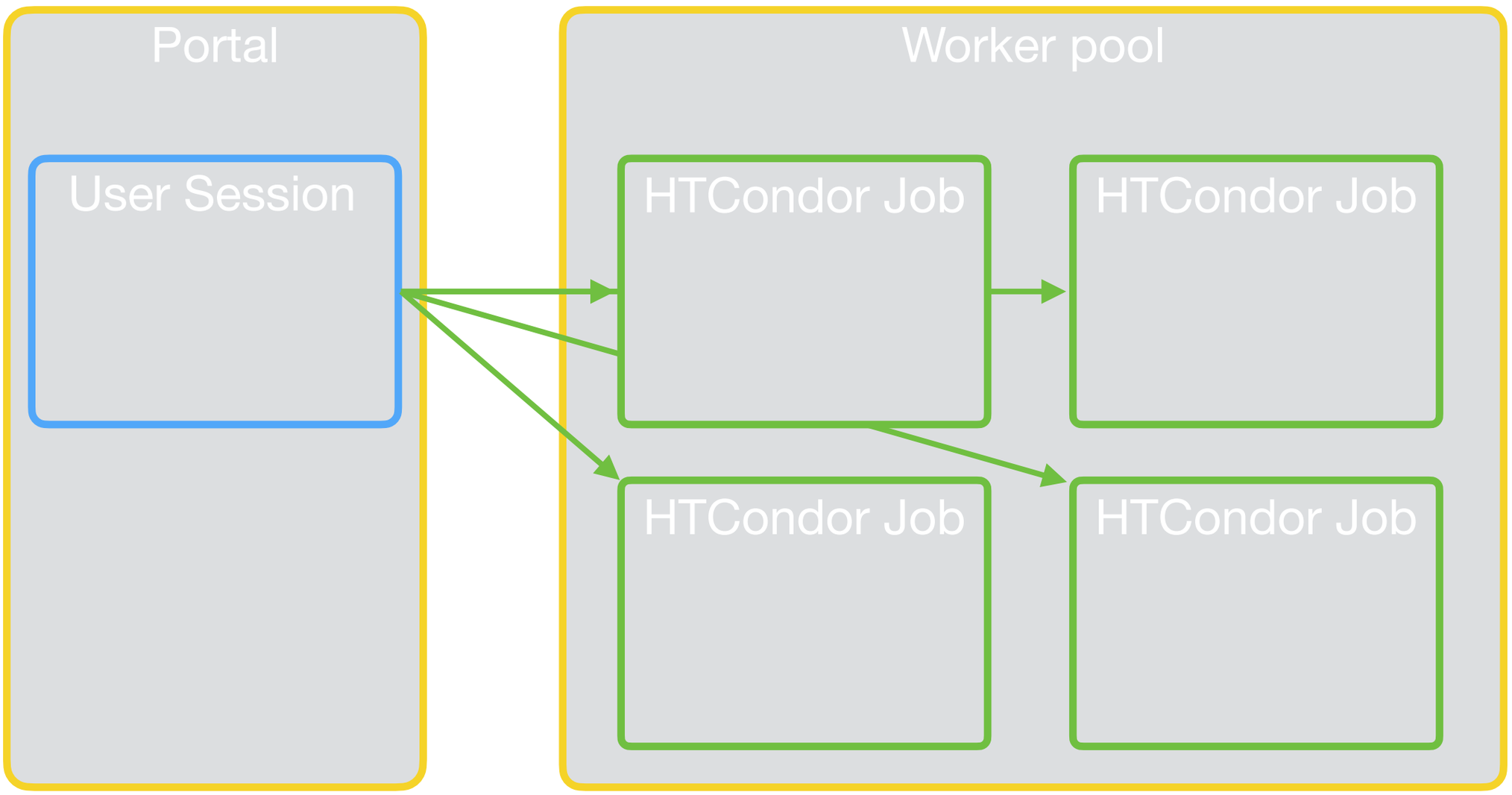
- Real computing payload by fast vectorised processing (map)
- Computations parallelised over chunks of data (100k events)
- Two step process:



Scale Out: Map

RWTH AACHEN UNIVERSITY
Dennis Noll - 21.10.22

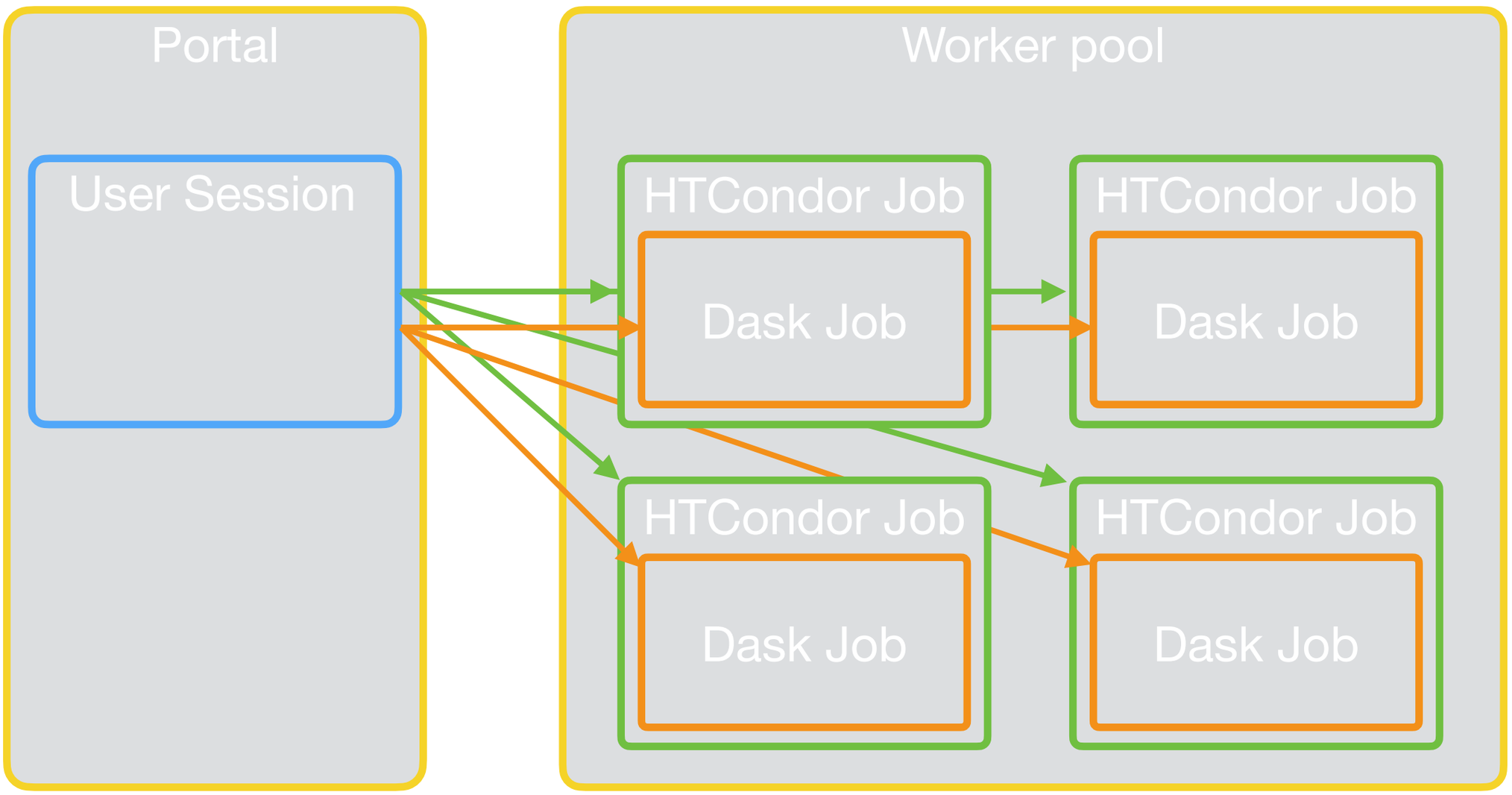
- Real computing payload by fast vectorised processing (map)
- Computations parallelised over chunks of data (100k events)
- Two step process:
  1. HTCondor opens jobs in worker pool (1 CPU Thread, 1.5 GB RAM)



Scale Out: Map

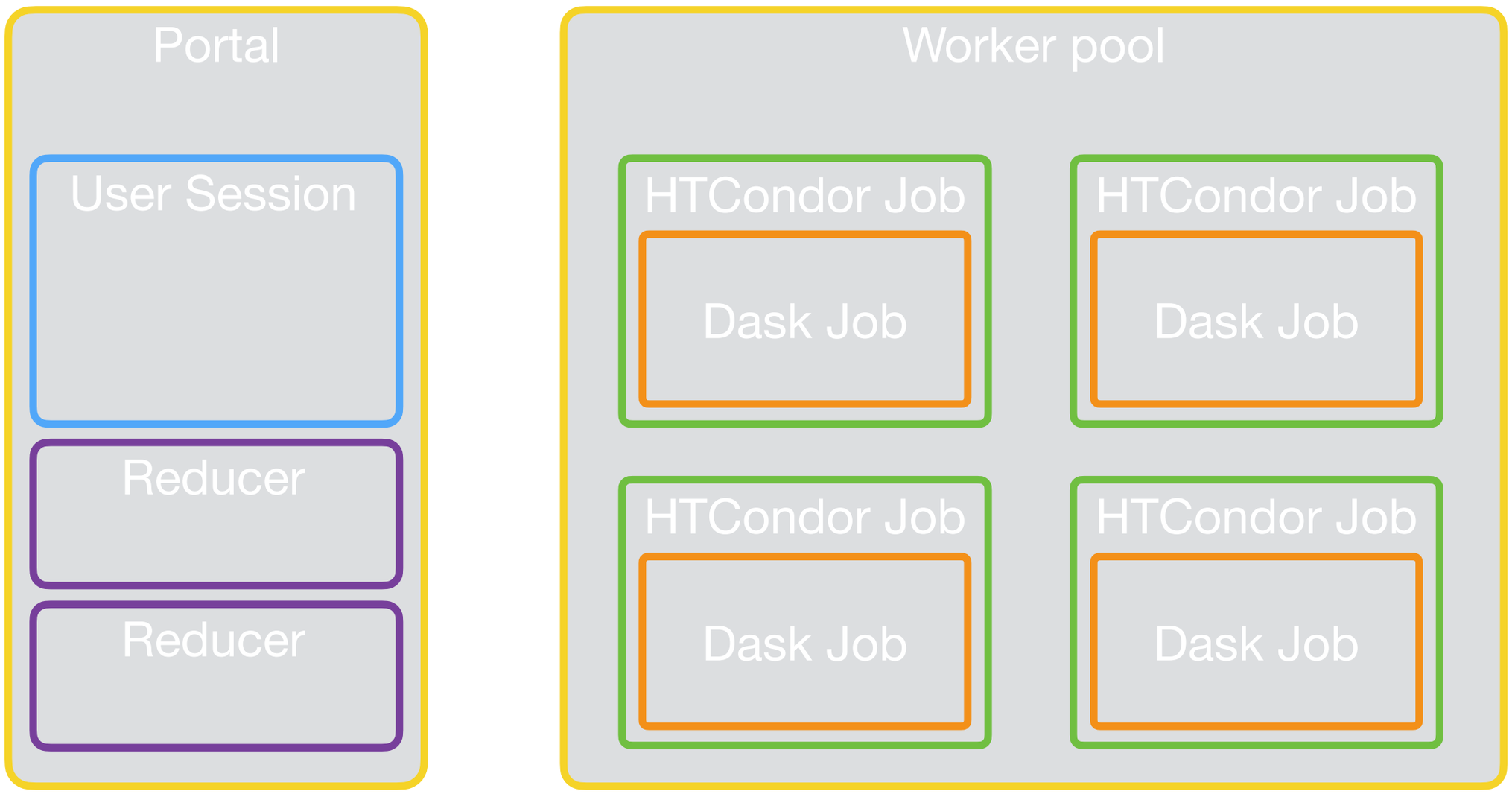
RWTH AACHEN UNIVERSITY
Dennis Noll - 21.10.22

- Real computing payload by fast vectorised processing (map)
- Computations parallelised over chunks of data (100k events)
- Two step process:
  1. HTCondor opens jobs in worker pool (1 CPU Thread, 1.5 GB RAM)
  2. Dask uses HTCondor slots



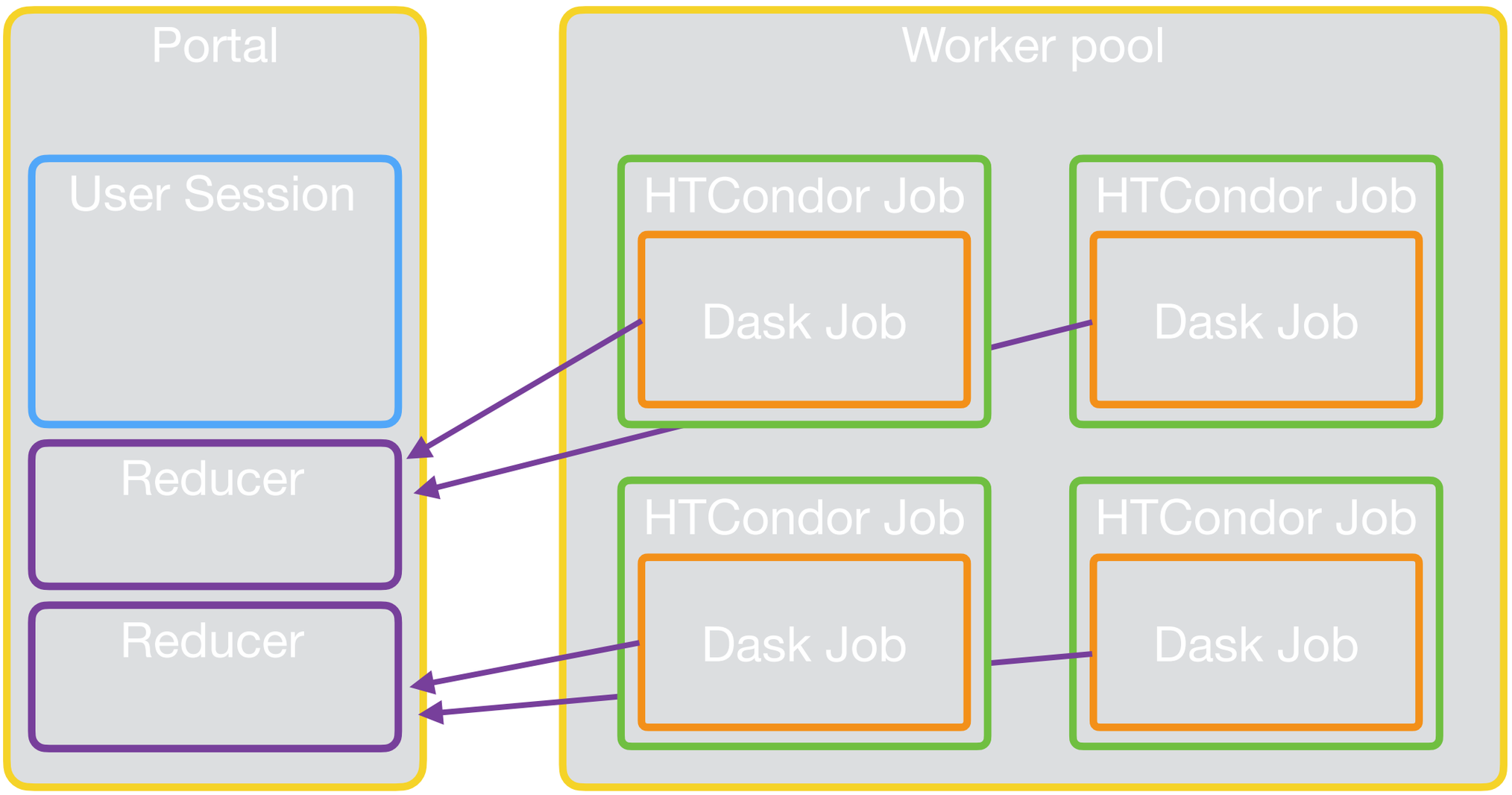
Scale Out: Reduce

- Outputs of parallel map jobs (e.g. histograms) must be *reduced*
- Ten parallel reducers on portal node
- Each reducer has two reducing instances:
  - Normal reduce: map jobs finished regularly, reduce job pulls
  - Early reduce: map jobs need space, map job pushes



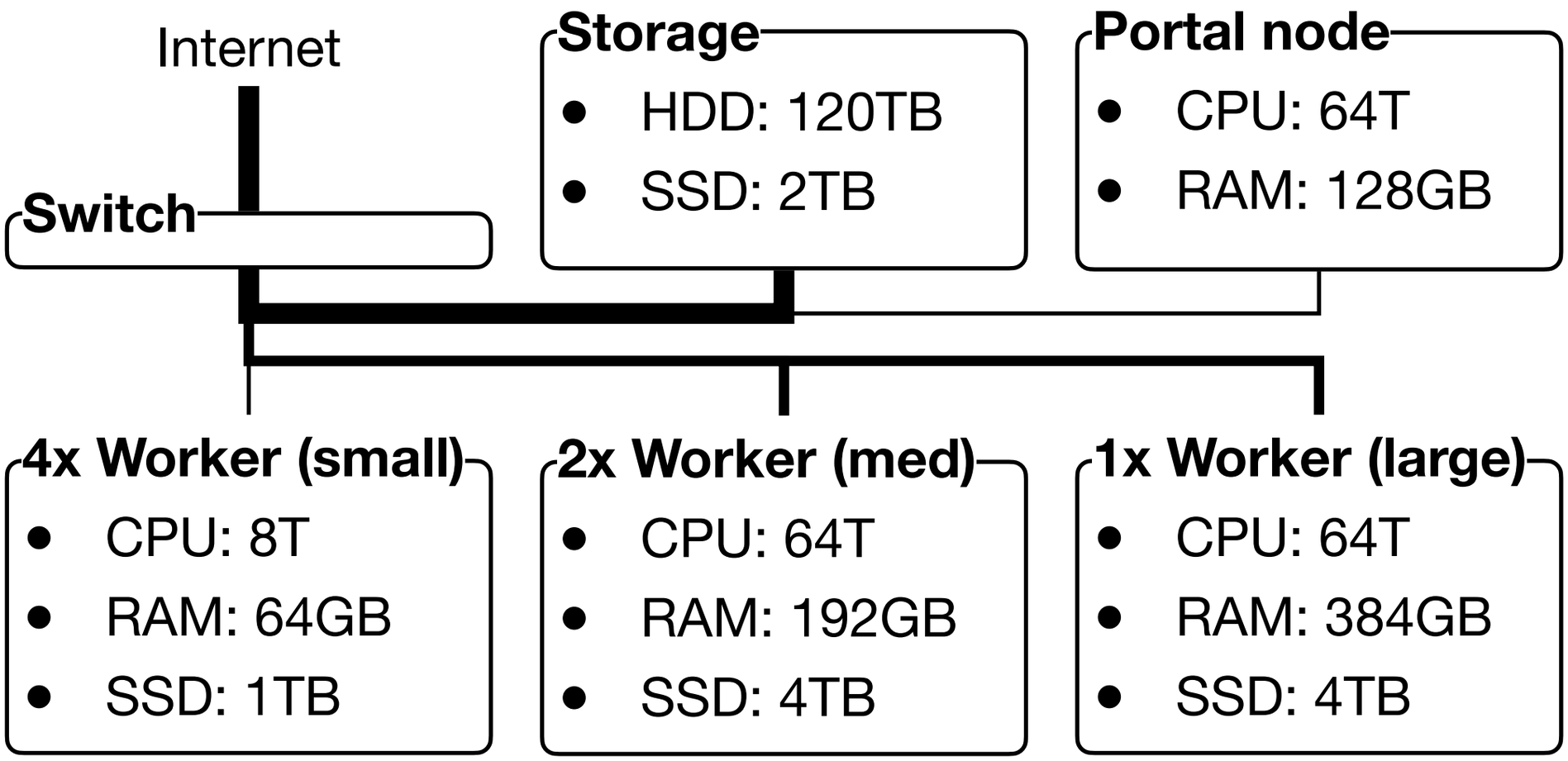
Scale Out: Reduce

- Outputs of parallel map jobs (e.g. histograms) must be *reduced*
- Ten parallel reducers on portal node
- Each reducer has two reducing instances:
  - Normal reduce: map jobs finished regularly, reduce job pulls
  - Early reduce: map jobs need space, map job pushes



Resource: VISPA Cluster

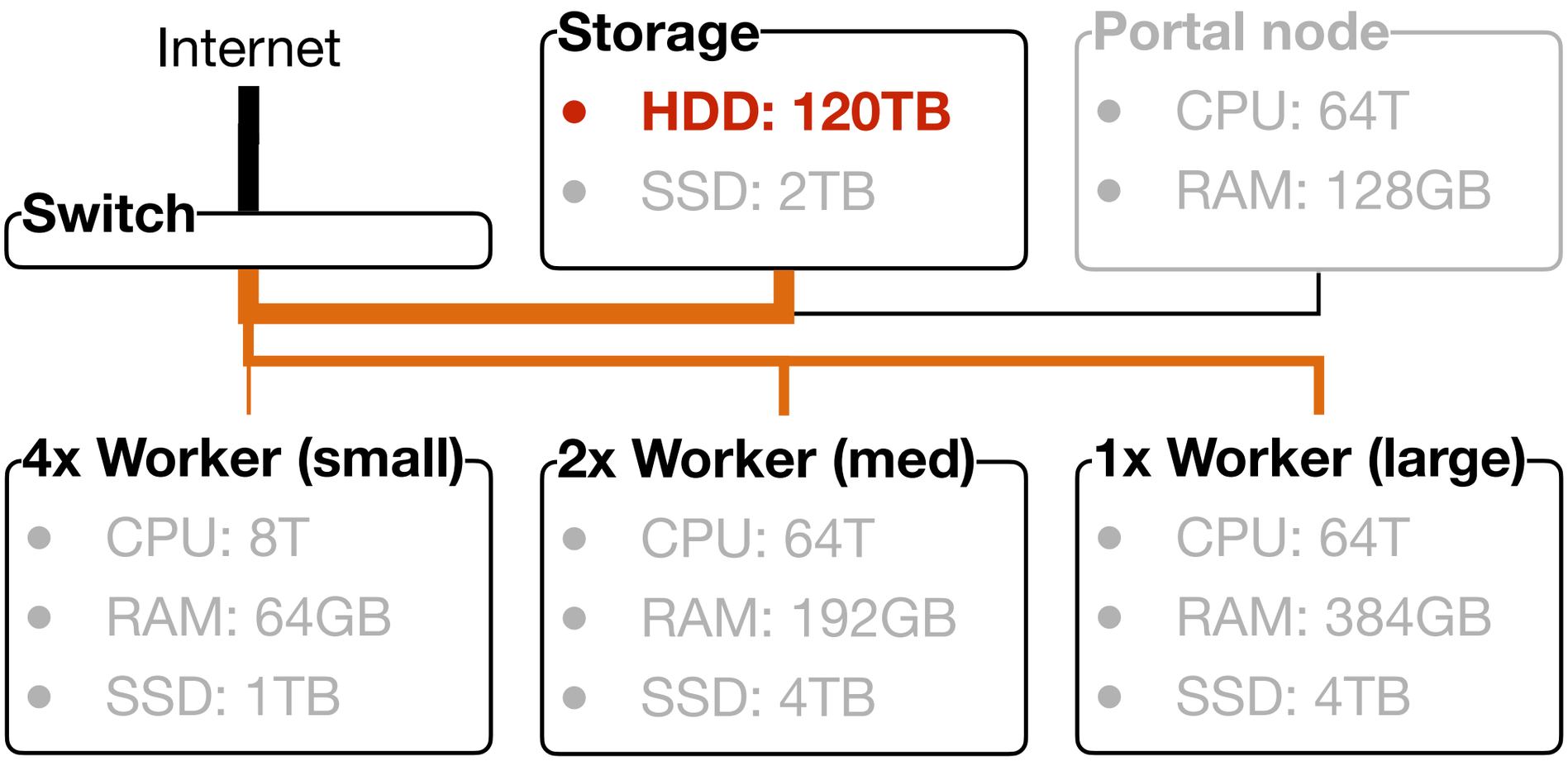
- All analyses run on local institute cluster (VISPA)
- Optimised for scientific data analysis and machine learning



Total: CPU: 224T, RAM: 832GB, GPU: 17 (various)

Resource: VISPA Cluster

- All analyses run on local institute cluster (VISPA)
- Optimised for scientific data analysis and machine learning
- Map jobs very fast, new analysis bottleneck:
  - **Read speed of HDDs**
  - **Limited network bandwidth**

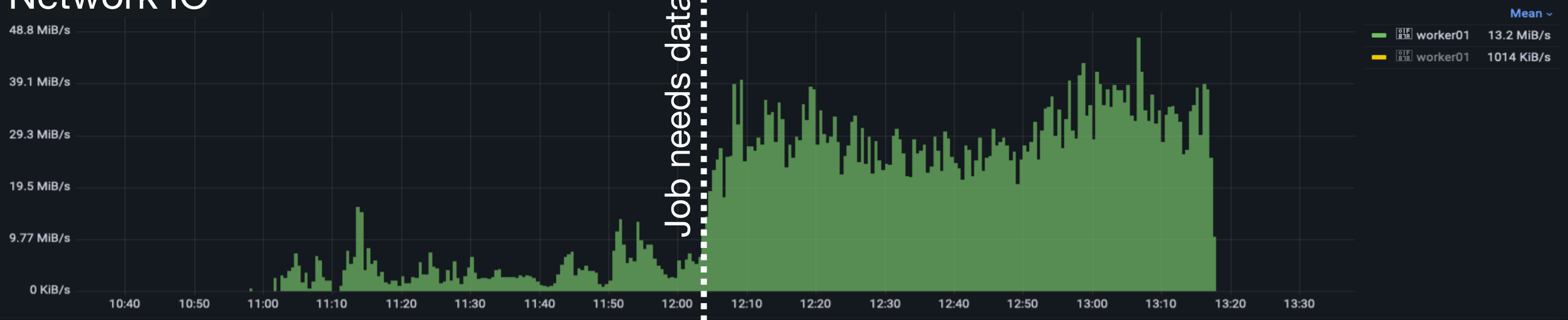


Total: CPU: 224T, RAM: 832GB, GPU: 17 (various)

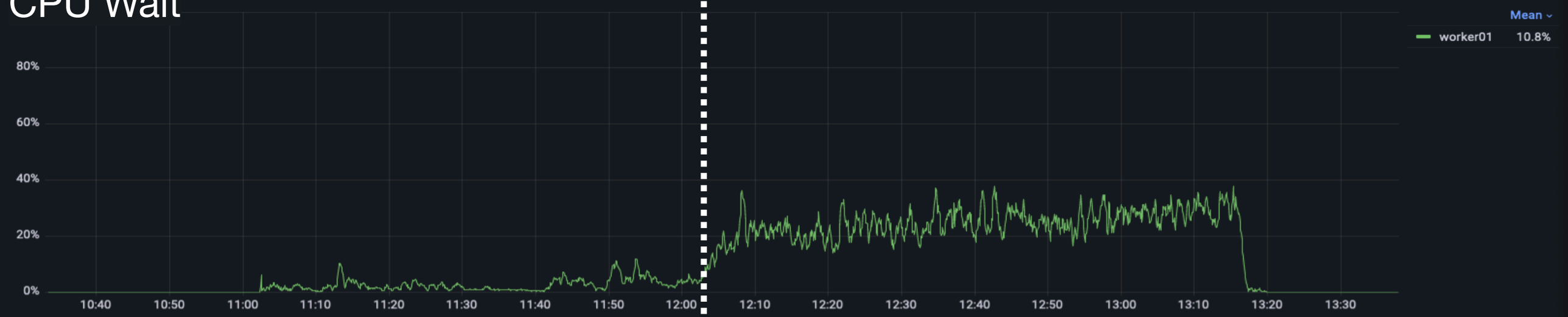
A New Bottleneck (2/2)

RWTH AACHEN UNIVERSITY
Dennis Noll - 21.10.22

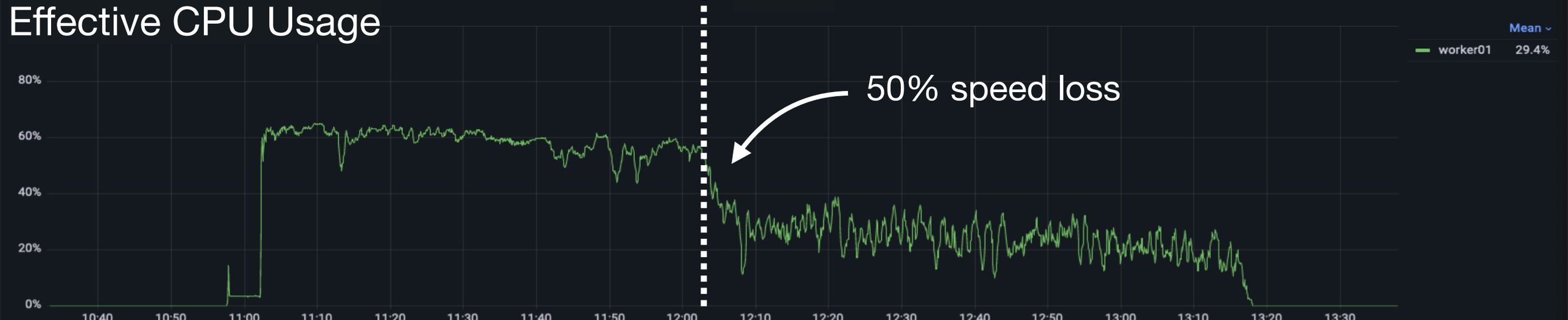
Network IO



CPU Wait

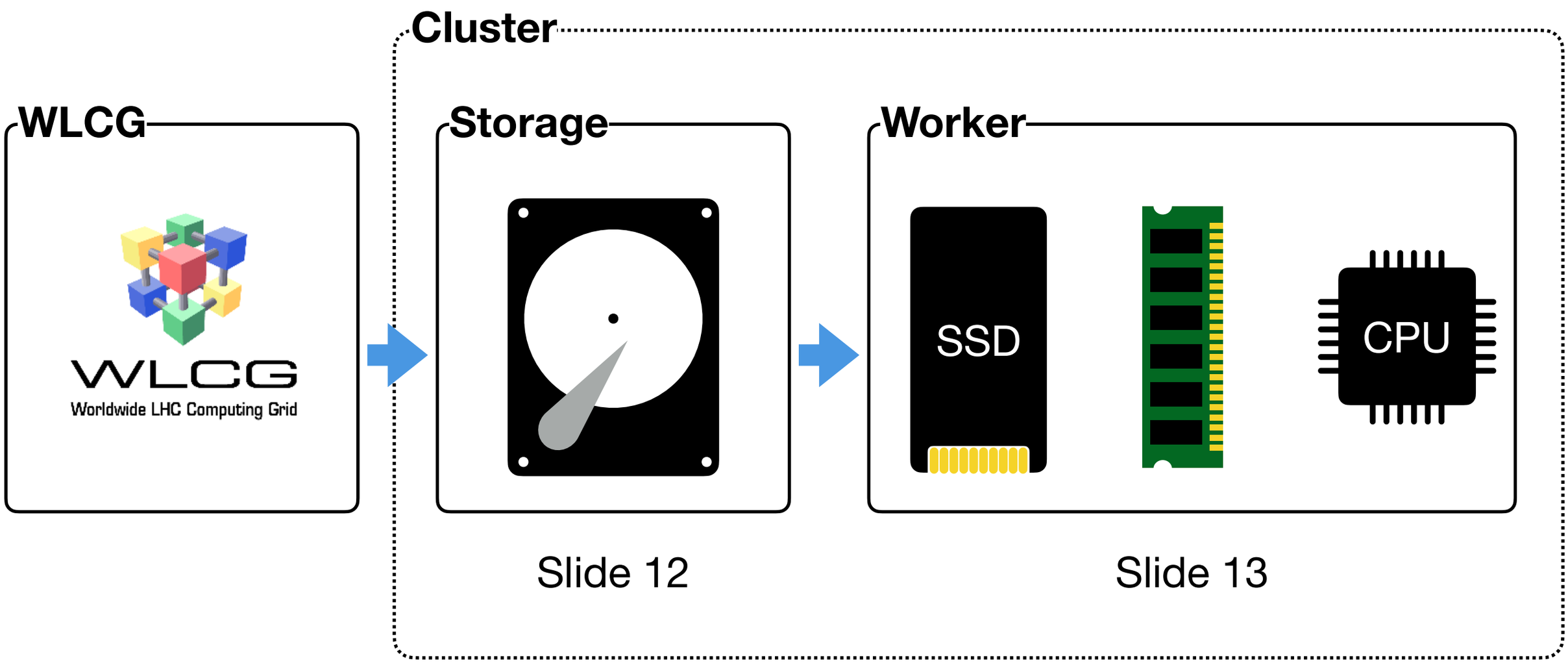


Effective CPU Usage



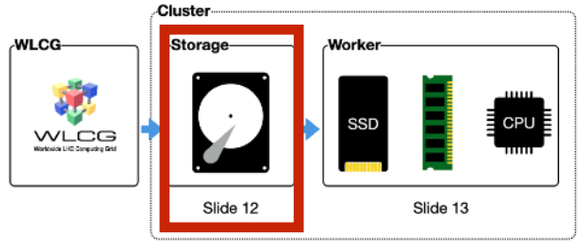
Storage Layout

- Critical bottleneck is streaming of data to the processing elements
- Various storage types and locations available to solve problem
- Two stage solution using central network storage and on-worker SSD

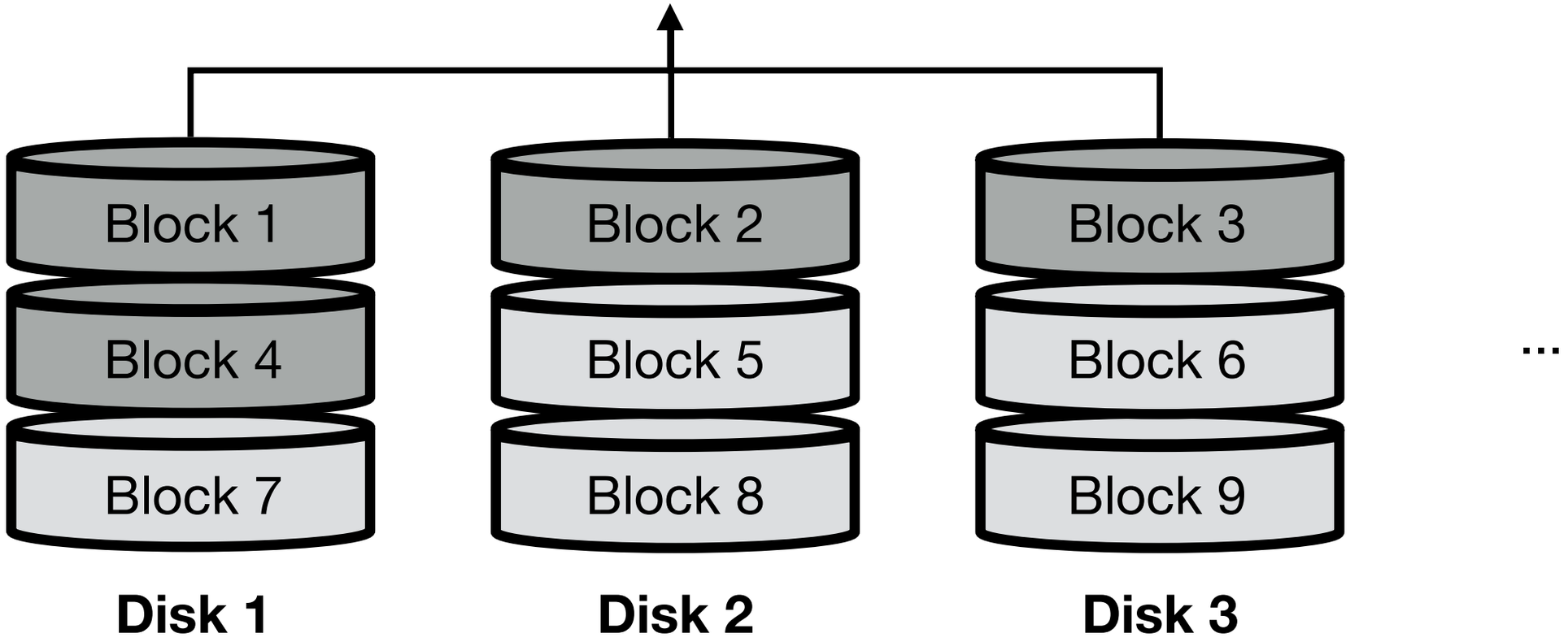


Network Attached Storage

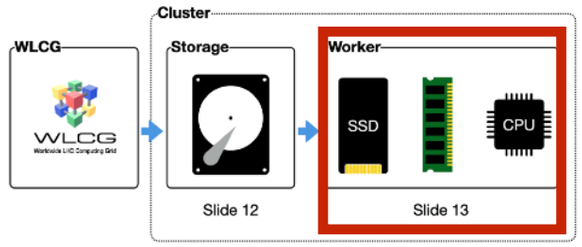
Dennis Noll - 21.10.22



- Save experiment data on-site:
  - Easy and reliable access (no timeouts, credentials, ...)
  - Direct connection to worker nodes (low latency, 10GBit)
- Data saved in RAID0
  - striped across 6 x 16TB HDDs enables fast reading

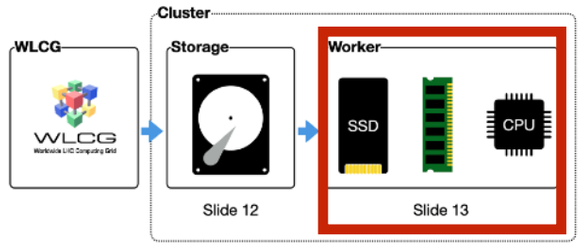


On-Worker Storage (FS-Cache)



- Using on-worker SSDs to minimise network traffic
- Used software implementation: FS-Cache & cachefilesd
  - Transparent caching system, available in Linux kernel
  - Block level (4kB) granularity
- **Challenge: Cache eviction!**

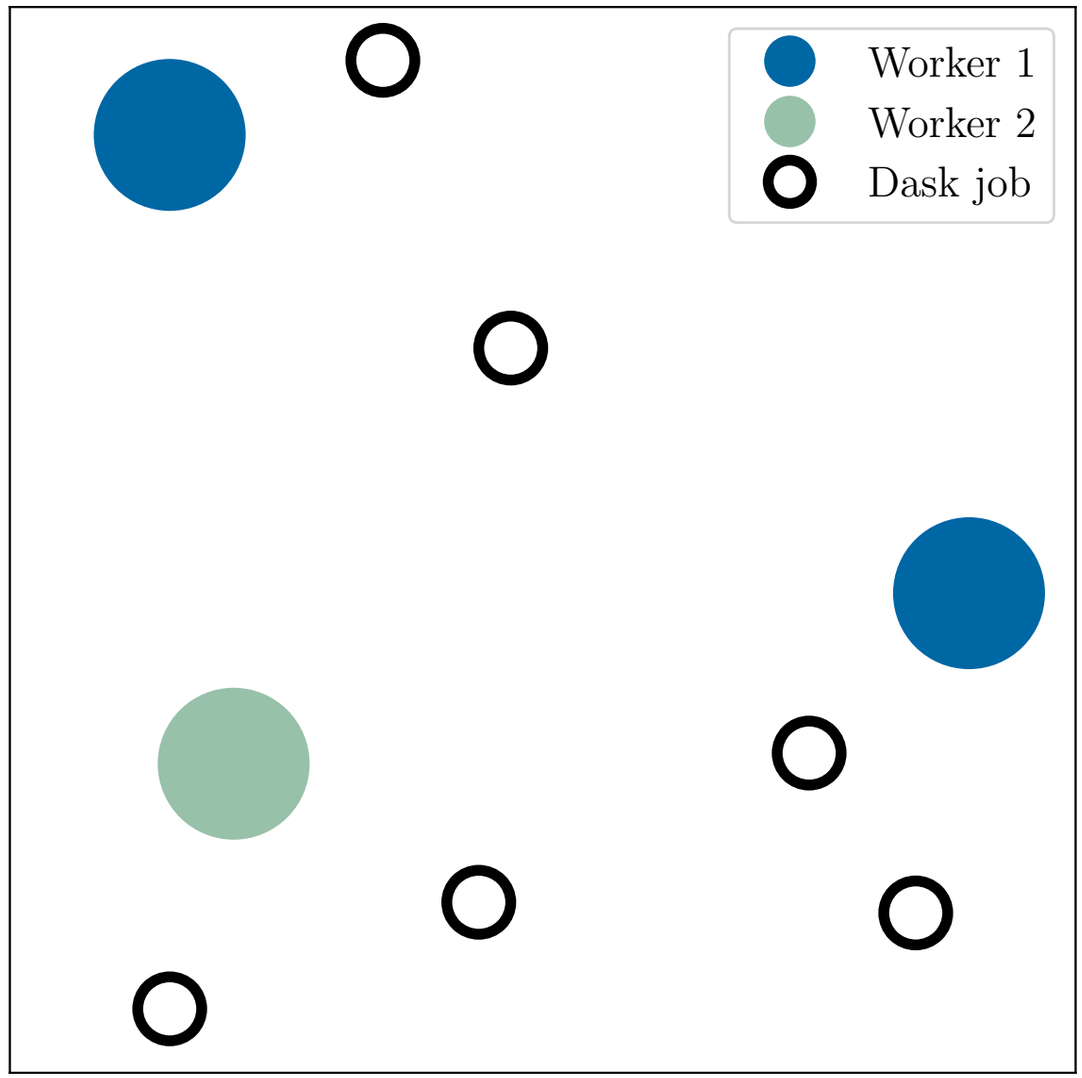
On-Worker Storage (FS-Cache)



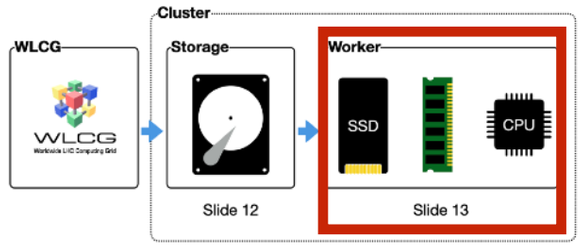
- Using on-worker SSDs to minimise network traffic
- Used software implementation: FS-Cache & cachefilesd
  - Transparent caching system, available in Linux kernel
  - Block level (4kB) granularity
- **Challenge: Cache eviction!**

- Use stable assignment Job to Worker
- Via embedding in in 64-dim space:

name: "worker01"
→ hash512: 10110100 11001001 ...
→ int embedding: (180, 201, ...)



On-Worker Storage (FS-Cache)

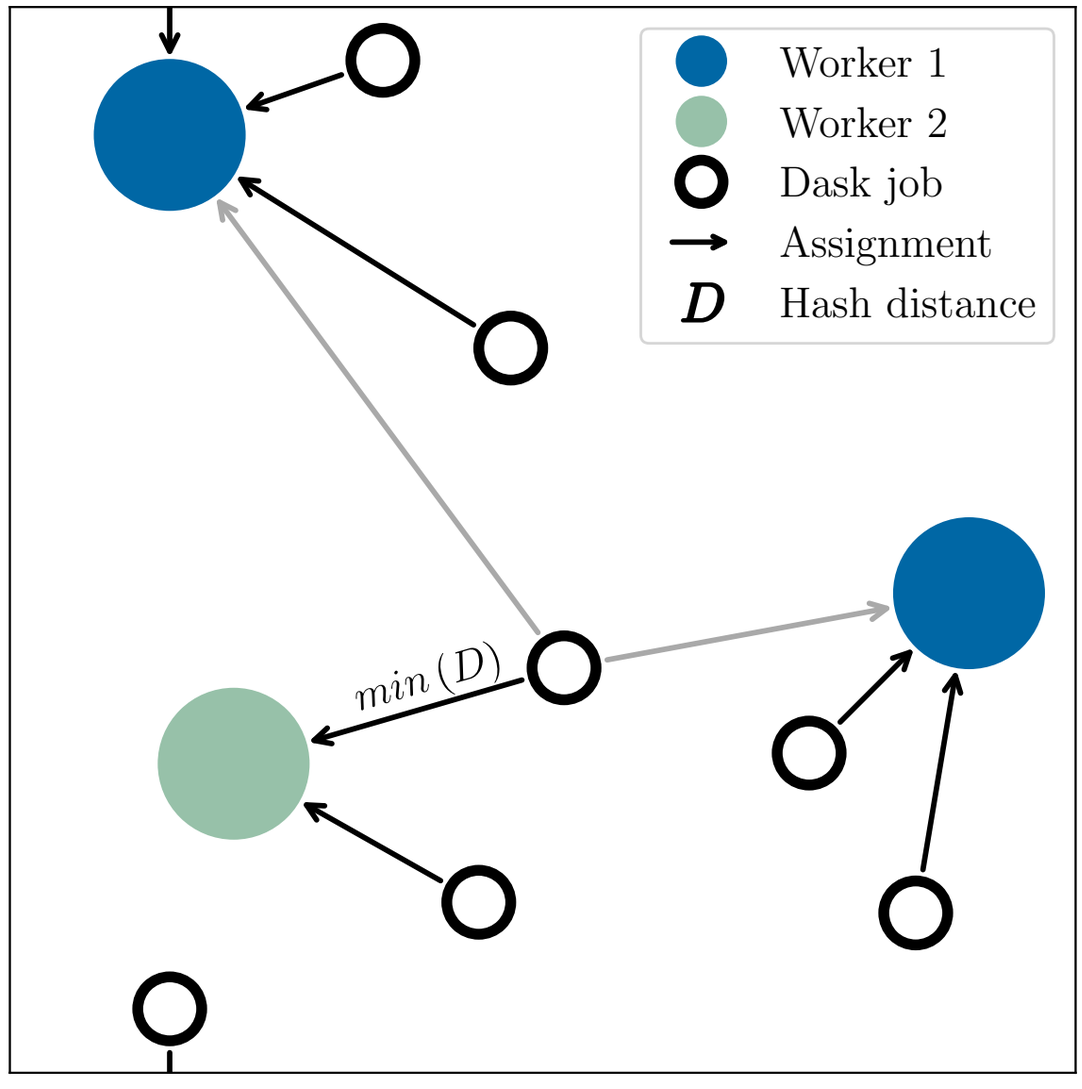


- Using on-worker SSDs to minimise network traffic
- Used software implementation: FS-Cache & cachefilesd
  - Transparent caching system, available in Linux kernel
  - Block level (4kB) granularity
- **Challenge: Cache eviction!**

- Use stable assignment Job to Worker
- Via embedding in in 64-dim space:

name: "worker01"
→ hash512: 10110100 11001001 ...
→ int embedding: (180, 201, ...)

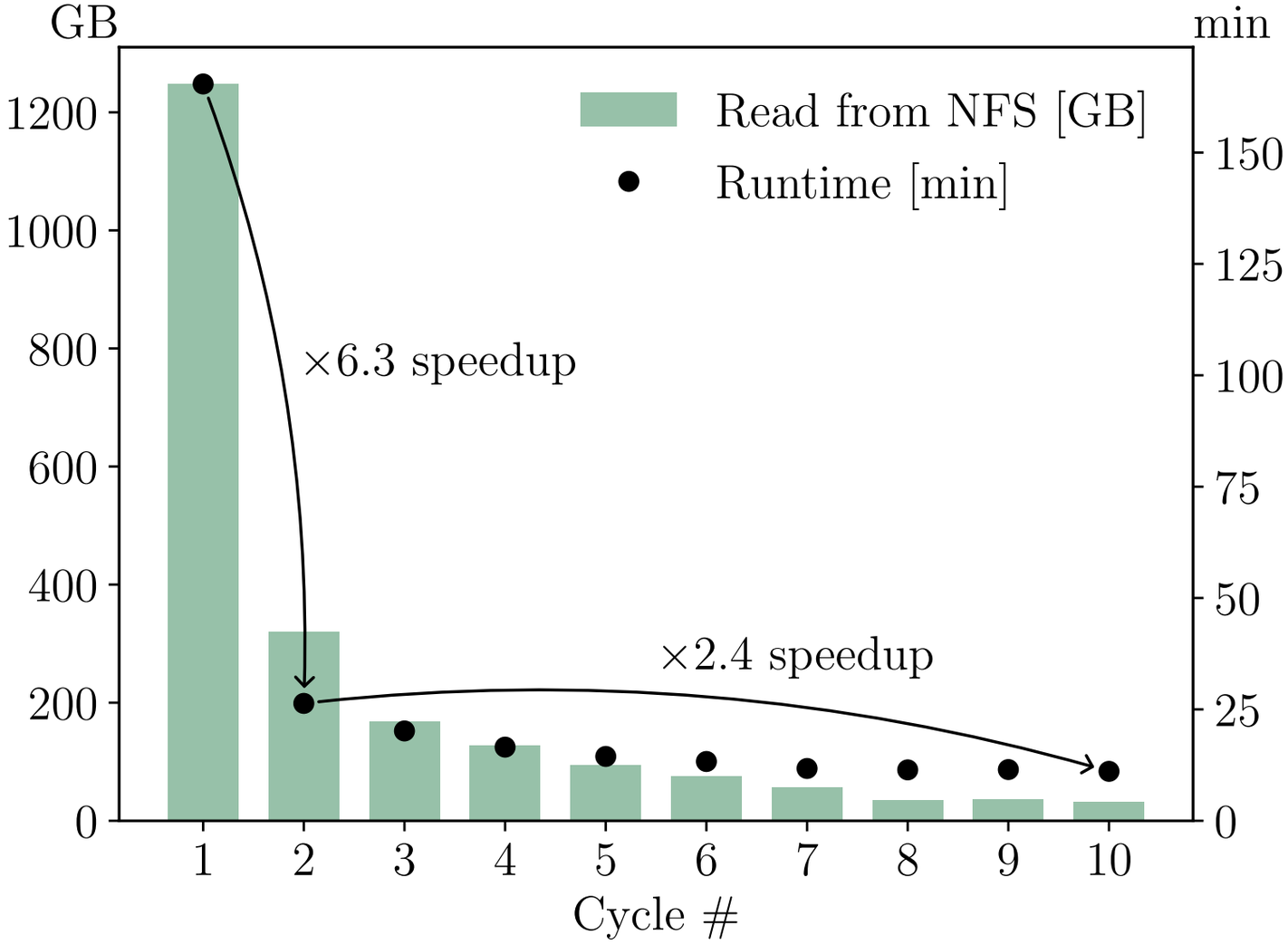
- Assignment via minimum distance D
- Results in:
  - Affine caching
  - Graceful on failures



Logic similar to ceph (link)

Benchmark: Definition and Results

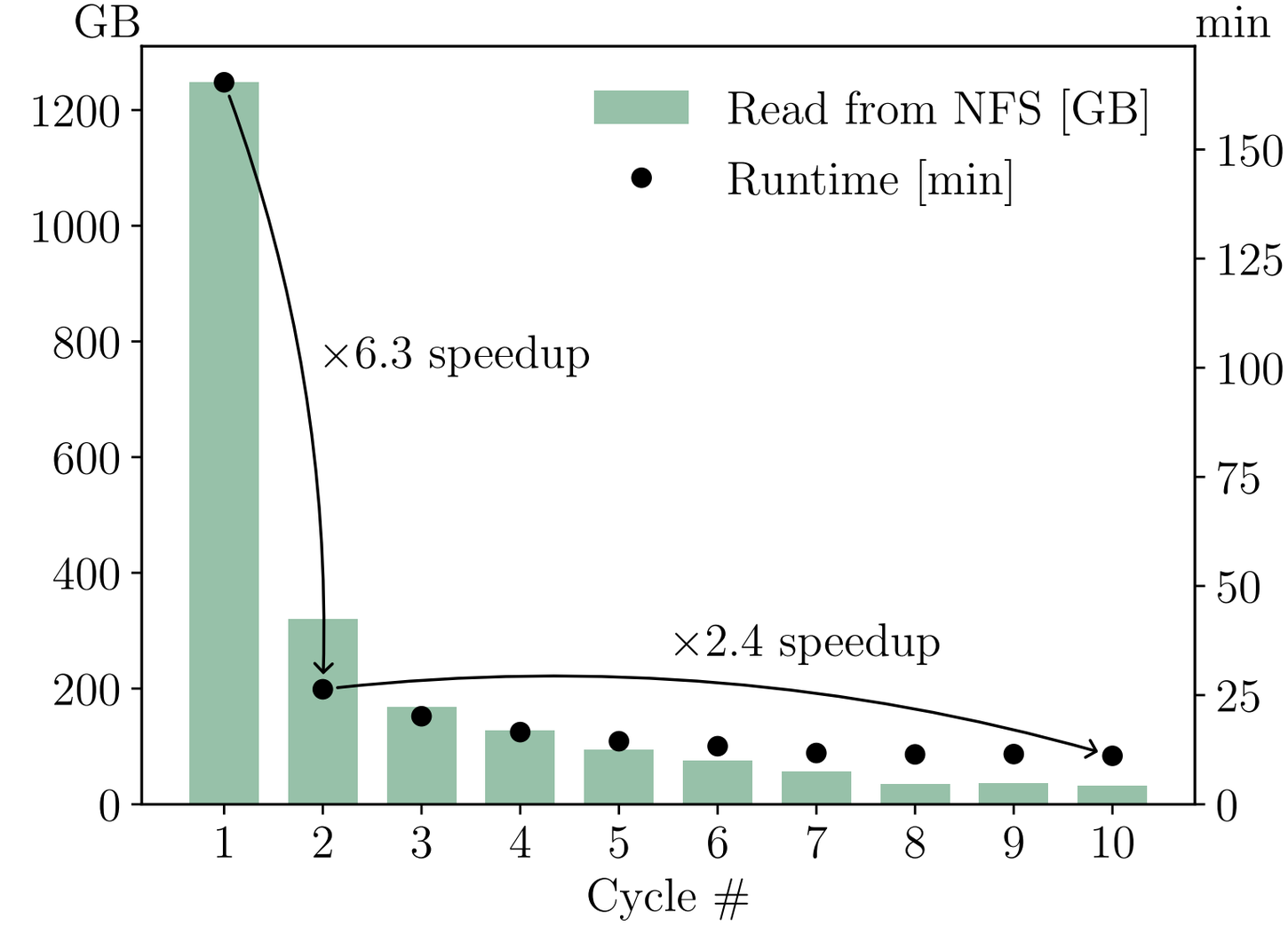
- Data: Higgs pair production analysis (1440GB, 10^9 events, 120 columns)
- Read-only task run 10 times (cycles) with 220 workers
- Results:
  - **Gradual performance: work stealing**
  - **Runtime lower bound: CPU → IO Bottleneck solved**



Benchmark: Definition and Results

Dennis Noll - 21.10.22

- Data: Higgs pair production analysis (1440GB, 10^9 events, 120 columns)
- Read-only task run 10 times (cycles) with 220 workers
- Results:
  - **Gradual performance: work stealing**
  - **Runtime lower bound: CPU → IO Bottleneck solved**

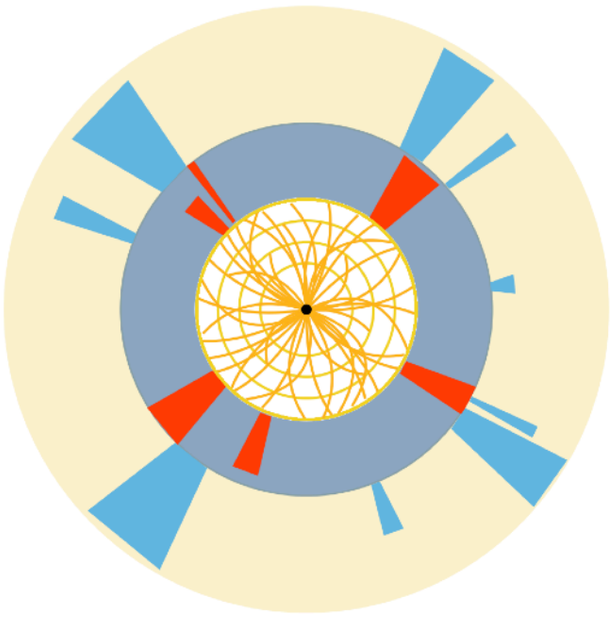


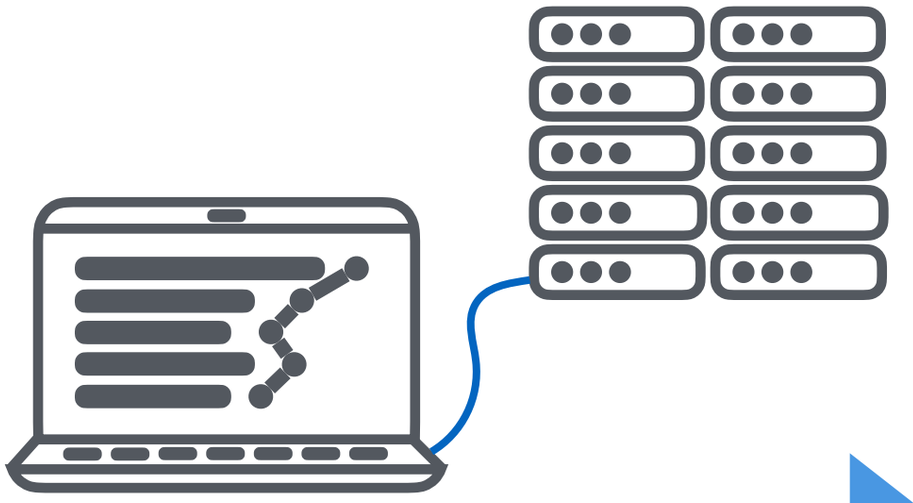
Final analysis runtime

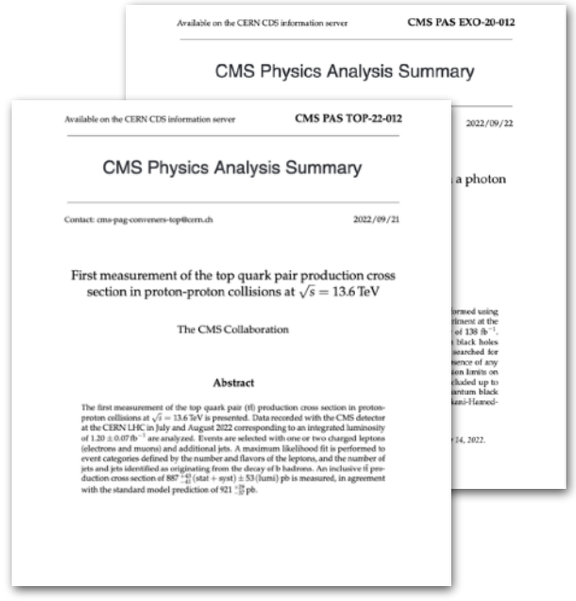
- **Explorative = 20 min**
  - (sim. only, no syst.)
- **Quick = 6h**
  - (no heavy syst.)
- **Full = 20h**
  - (publication ready)

Summary

- Fast O(TB) Physics Analysis on Small Institute Cluster
- Columnar processing via NumPy, Awkward
- Job distribution via map & reduce (Dask)
- Solved IO bottleneck:
  - Optimized storage distribution
  - Affine caching concept
- Analysis runtimes:
  - Explorative: 20 min
  - Full: 20h







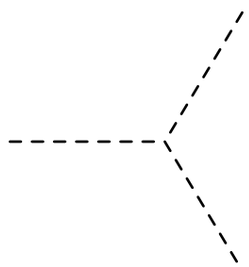
Backup

Physics Analysis & Team

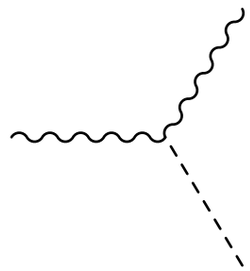
Team (Standard CMS institute)

Prof.

Higgs Pair Production



Higgs Strahlung



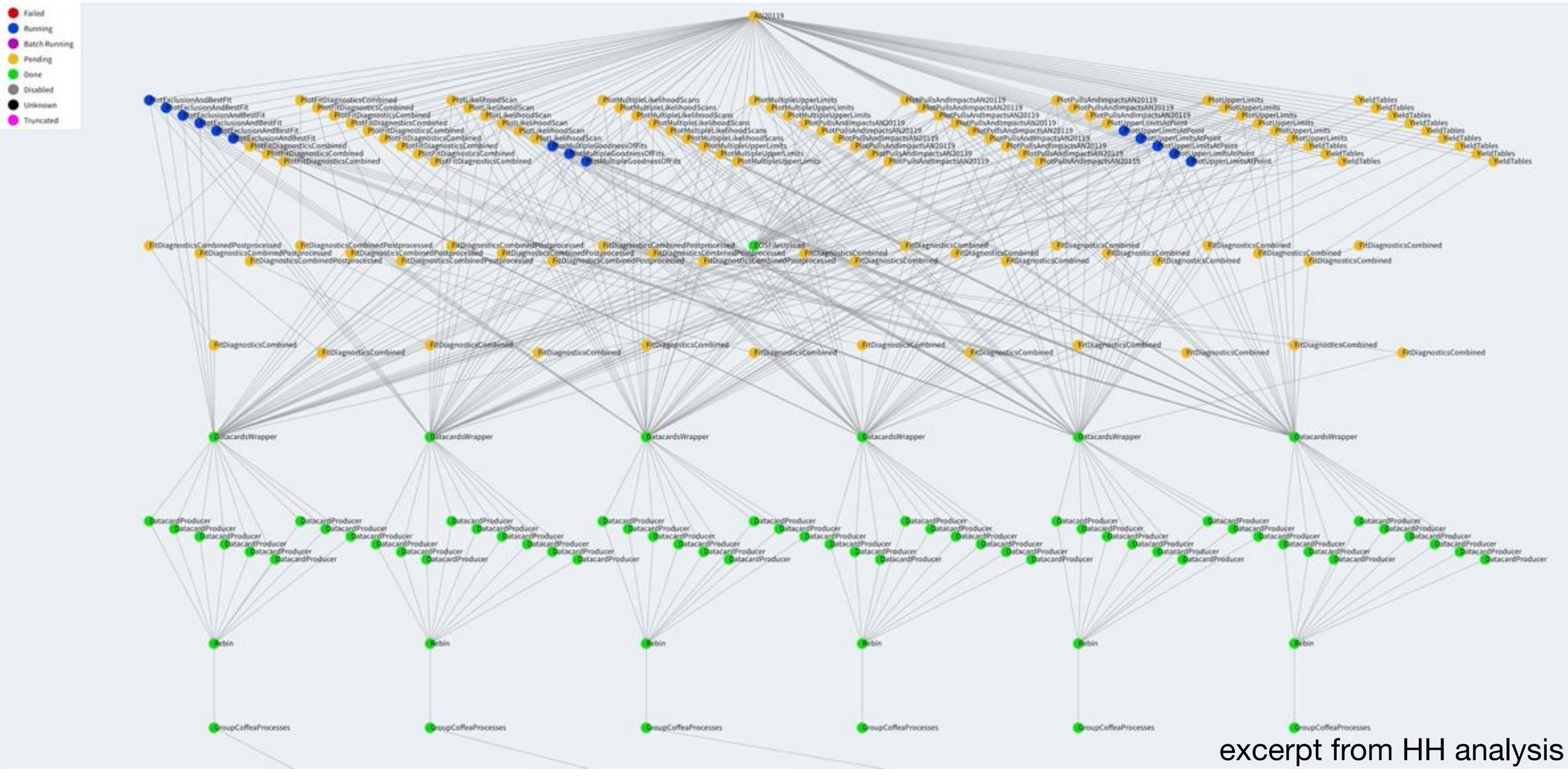
Various

New MVA methods,
Jet-Flavour tagging,
...

- Analyses have common challenges:
  - Many inputs:
    - Simulations and recorded data
    - Meta data (Efficiency factors, Corrections, ...)
  - Heavy computations (Event reconstruction, MVA algorithms, ...)
  - Bookkeeping
- Shared software repository

Analysis Structure (2/2)

RWTH AACHEN UNIVERSITY
Dennis Noll - 21.10.22

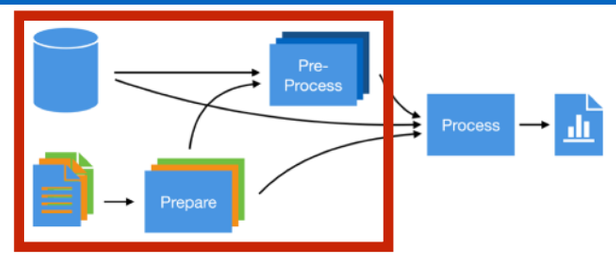


excerpt from HH analysis

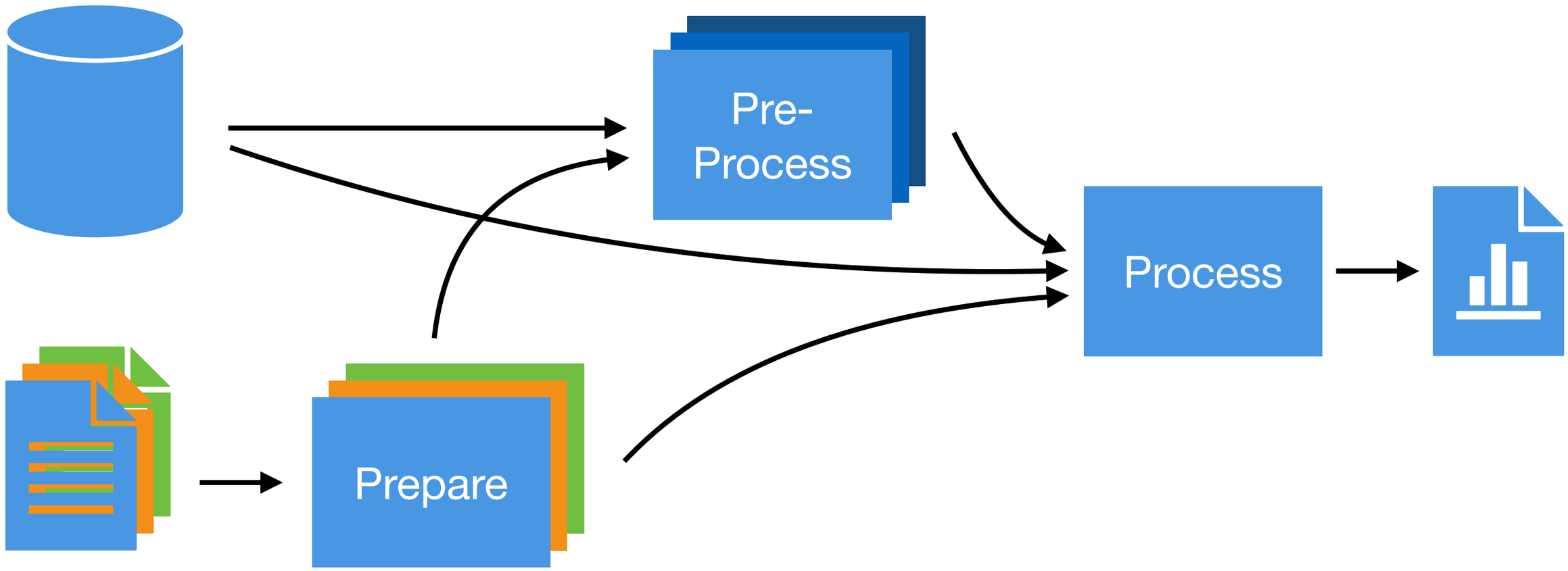
- Make-like execution of whole analysis (`law run FullAnalysis`)
- Visual task graph representation using Luigi Scheduler
  - Used for overview of run status, structural improvements, debugging

Analysis Structure - Inputs

Dennis Noll - 21.10.22

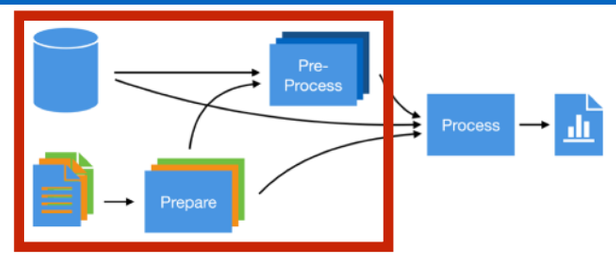


- Different kinds of inputs:
  - Event data O(10TB):
    - Recorded data & Simulation
    - NanoAOD format ~1kB per event
  - Meta data O(MB):
    - Efficiency measurements, scaling factors, ...
    - Twiki pages, JSON files, Custom ROOT files, ...

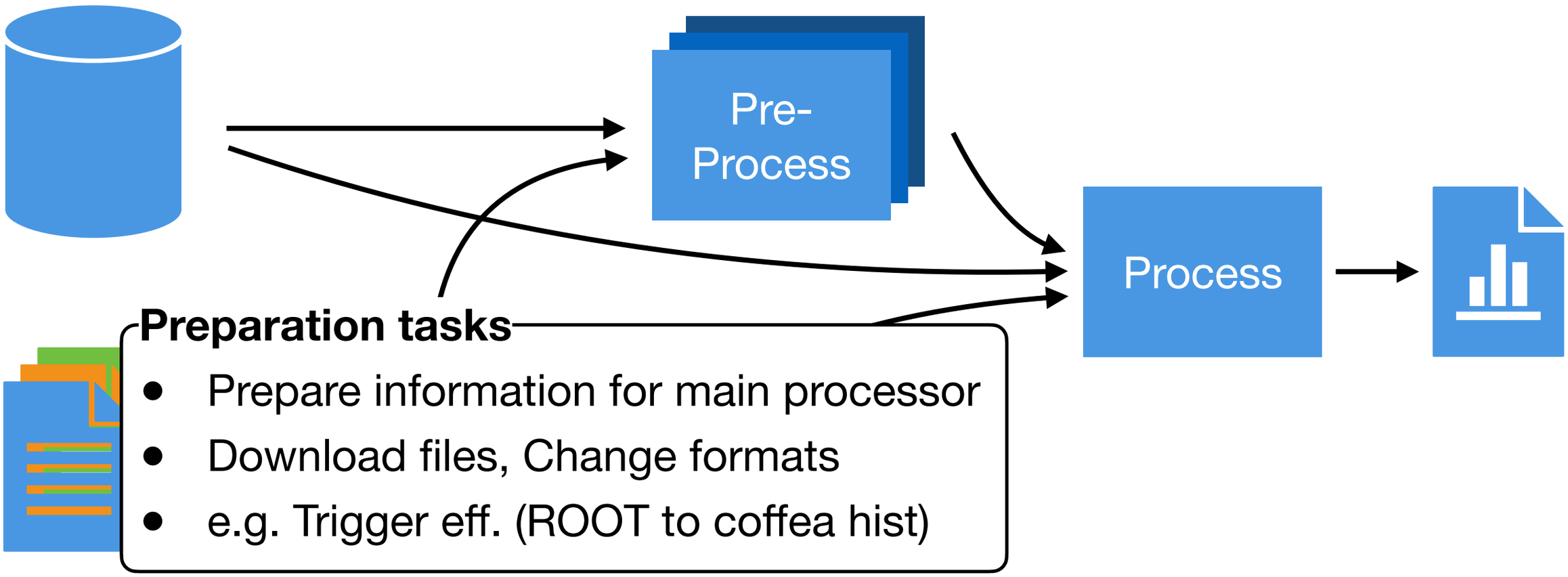


Analysis Structure - Inputs

RWTH AACHEN UNIVERSITY
Dennis Noll - 21.10.22

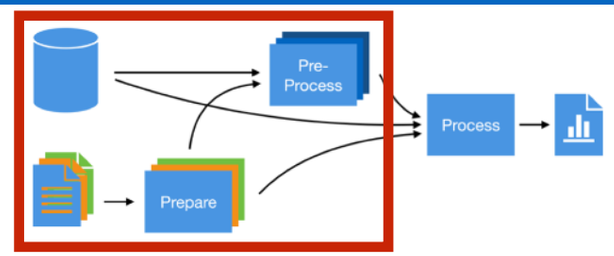


- Different kinds of inputs:
  - Event data O(10TB):
    - Recorded data & Simulation
    - NanoAOD format ~1kB per event
  - Meta data O(MB):
    - Efficiency measurements, scaling factors, ...
    - Twiki pages, JSON files, Custom ROOT files, ...

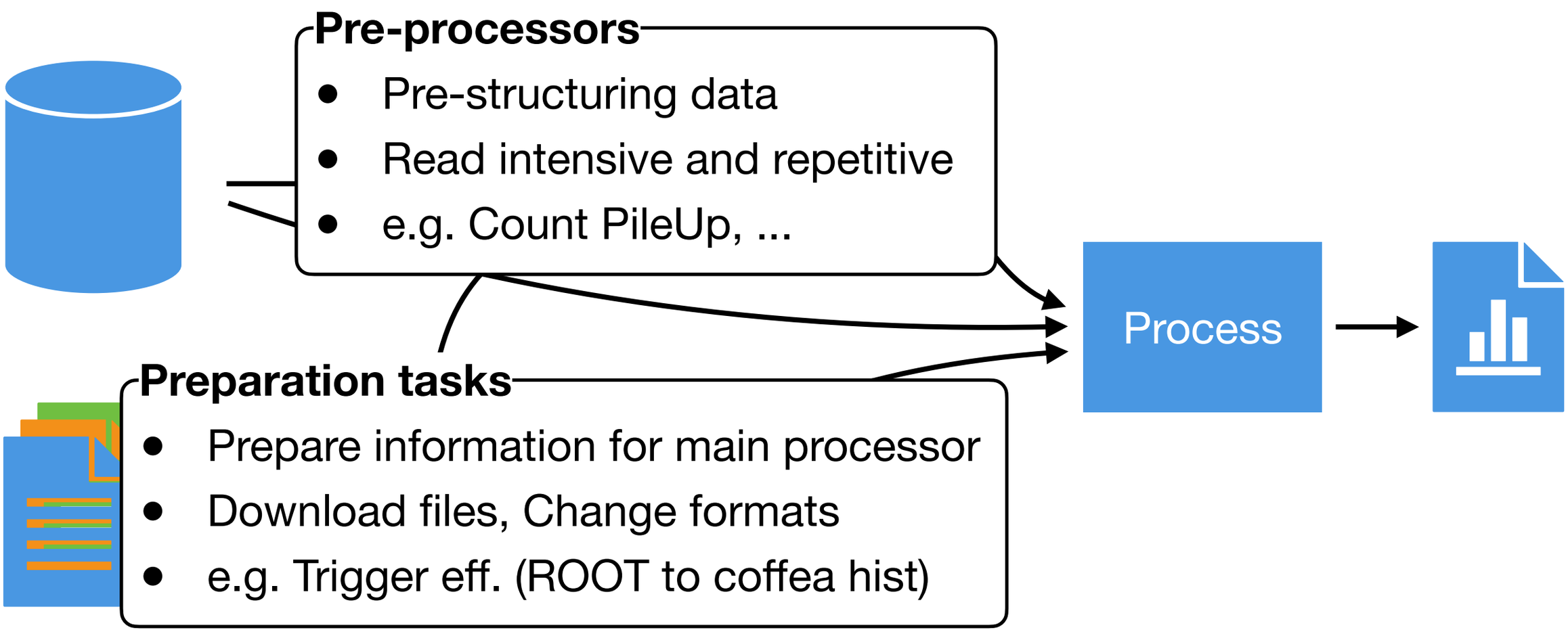


Analysis Structure - Inputs

Dennis Noll - 21.10.22

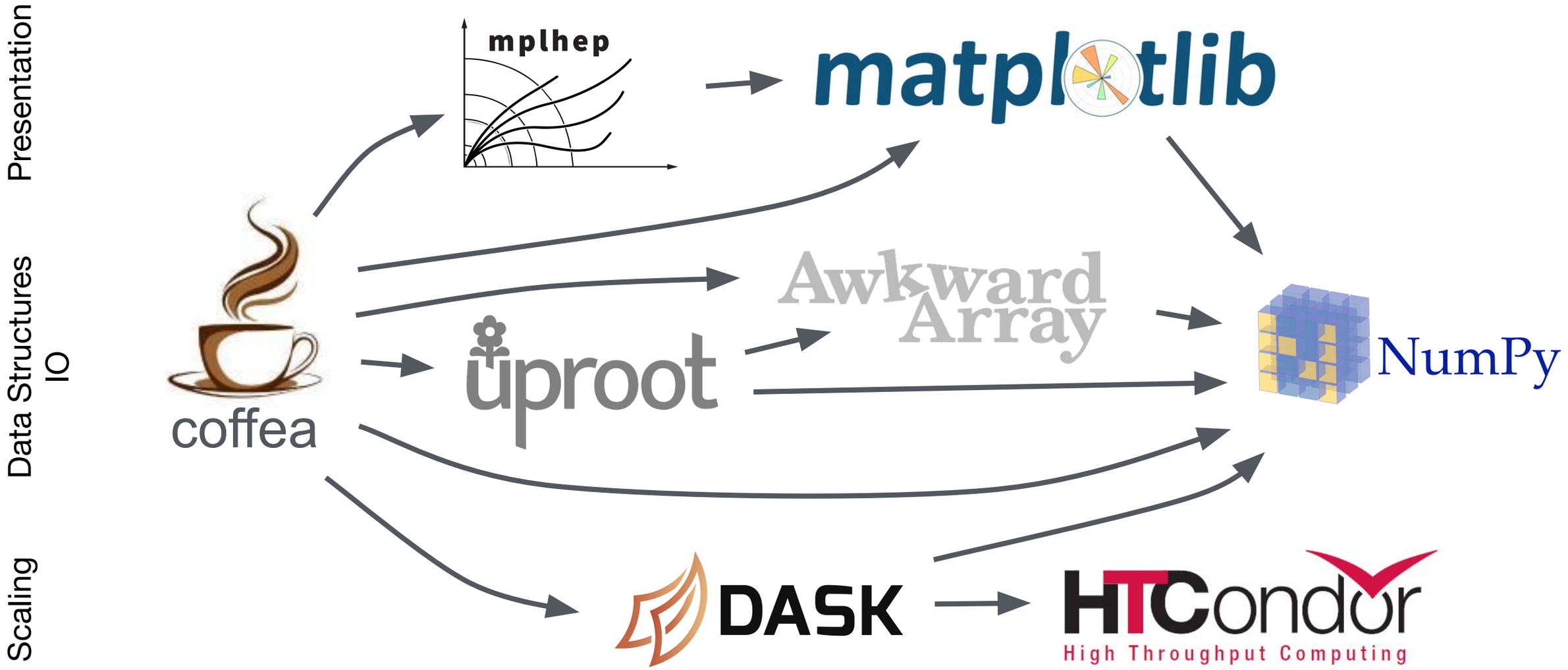


- Different kinds of inputs:
  - Event data O(10TB):
    - Recorded data & Simulation
    - NanoAOD format ~1kB per event
  - Meta data O(MB):
    - Efficiency measurements, scaling factors, ...
    - Twiki pages, JSON files, Custom ROOT files, ...



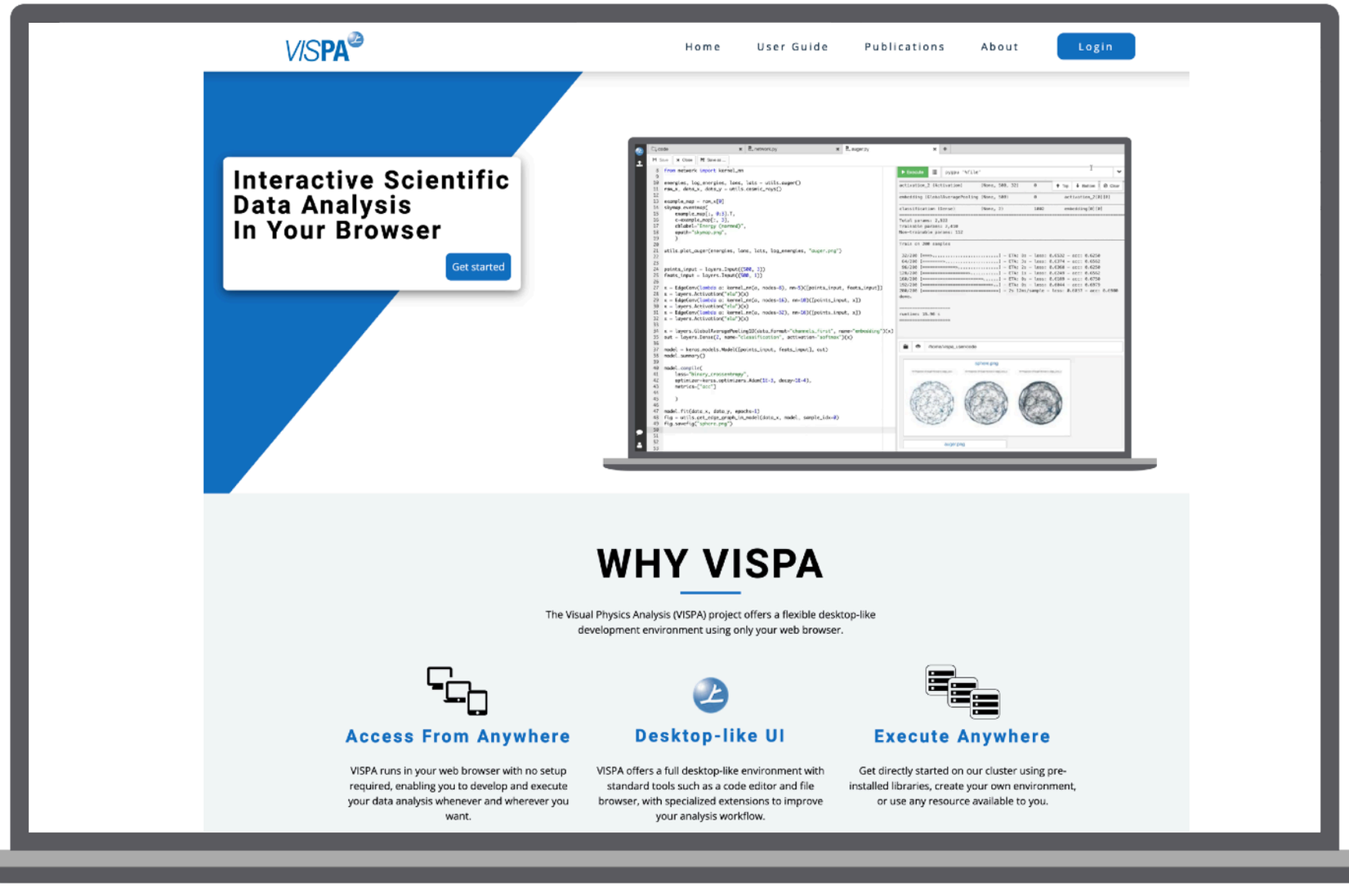
Software Stack

- Everything implemented in **python** (analysis repository)
- Modular software framework:
  - Standard tools: NumPy, matplotlib, Dask, HTCondor, ...
  - HEP specific tools: AwkwardArray, uproot, mplhep, coffea, ...



Resource: VISPA Cluster (2/2)

- User base:
  - 10 Researchers on daily basis (e.g. CMS experiment, Auger observatory)
  - Courses with up to 200 participants (e.g. Nuclear physics, ML in Physics)
  - Schools and workshop with up to 50 participants
- Front-end for data analysis in your web browser (link)



Many Alternatives! (& why we don't use them)

XCache

XRootD

- No POSIX access
- Not easy with many users

Distributed File Systems

ceph

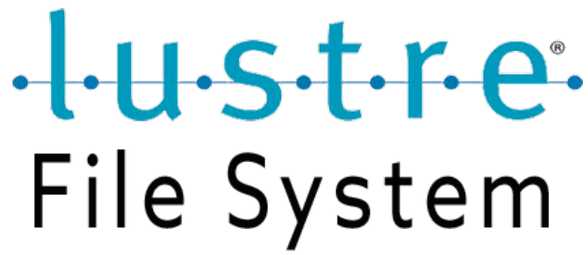
- Does not handle changing files so well

XTREEMFS

- No cache tiering

GlusterFS

- Cache tiering removed



lustre® File System

- No graceful failure of parts of storage

MooseFS

- Caching only based on modification time

- Seems promising but still very new
- Documentation not complete yet