

KIT - Analysis Infrastructure Status & Plans

Manuel Giffels

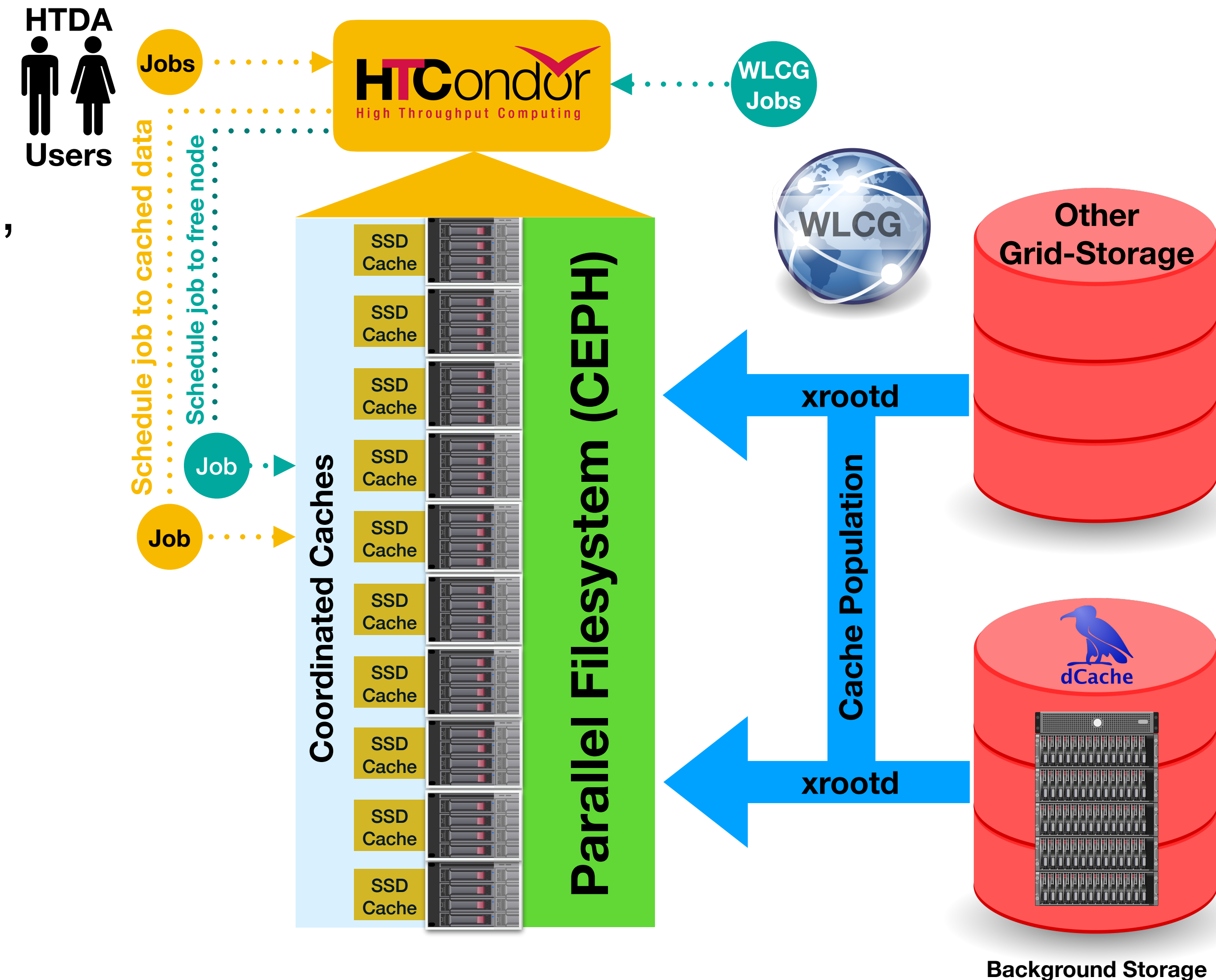
ErUM-Data Analysis Facilities Workshop, Garching, 19.06.2024



2018 - 2024: High Throughput Data Analysis

Throughput Optimized Analysis System (TOPAS) hosted at GridKa

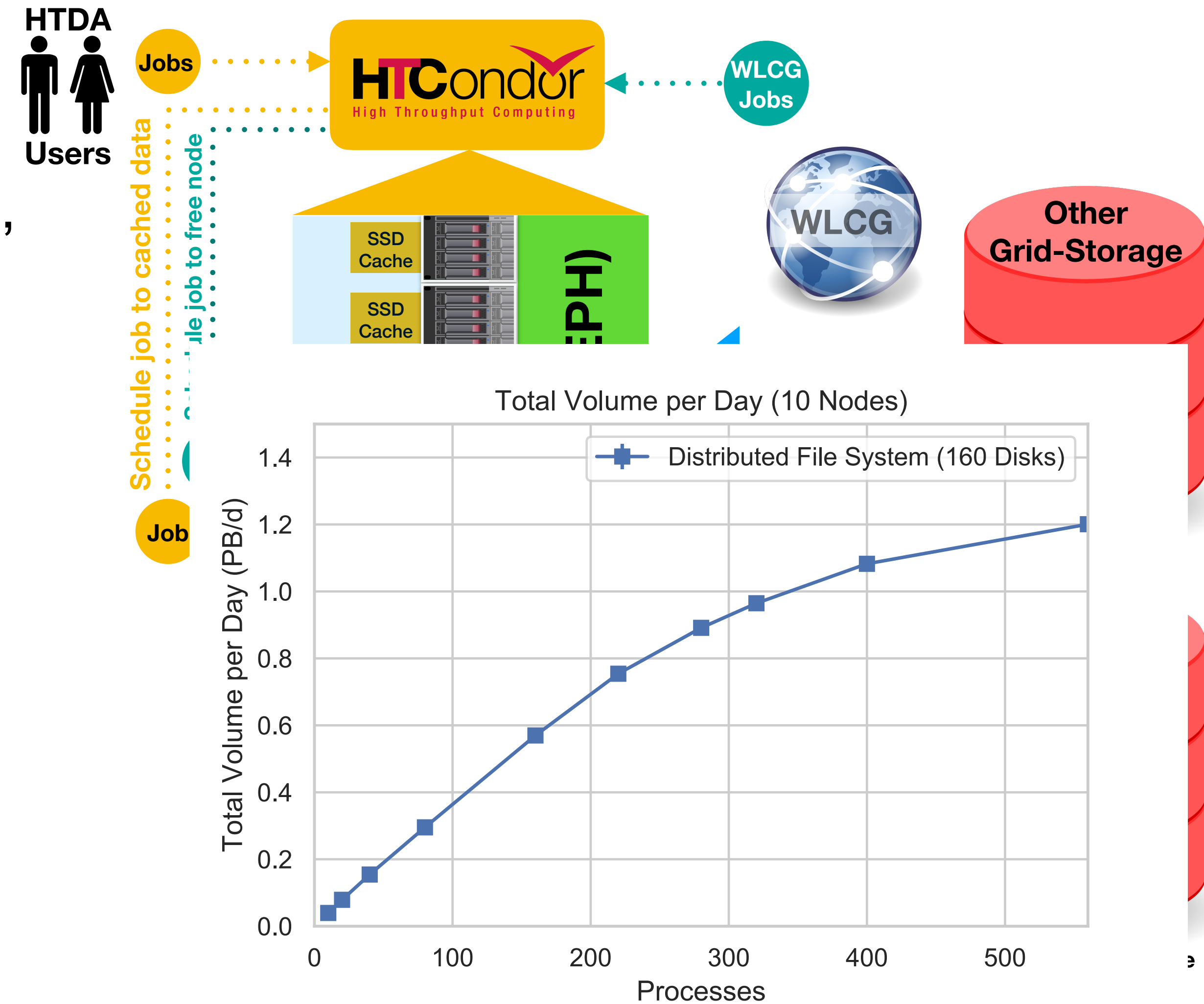
- ~2200 CPU cores, 56 Nvidia GPUs (V100, V100S, A100)
- Each node has a 100Gb/s network connection
- 1PB CEPH disk cache
- 200 Gb/s uplink to background storage and LHCONE/OPN
- HTCondor Batch System
 - Backfilling with WLCG jobs in place
 - Throughput of 1.2 PB per day



2018 - 2024: High Throughput Data Analysis

Throughput Optimized Analysis System (TOPAS) hosted at GridKa

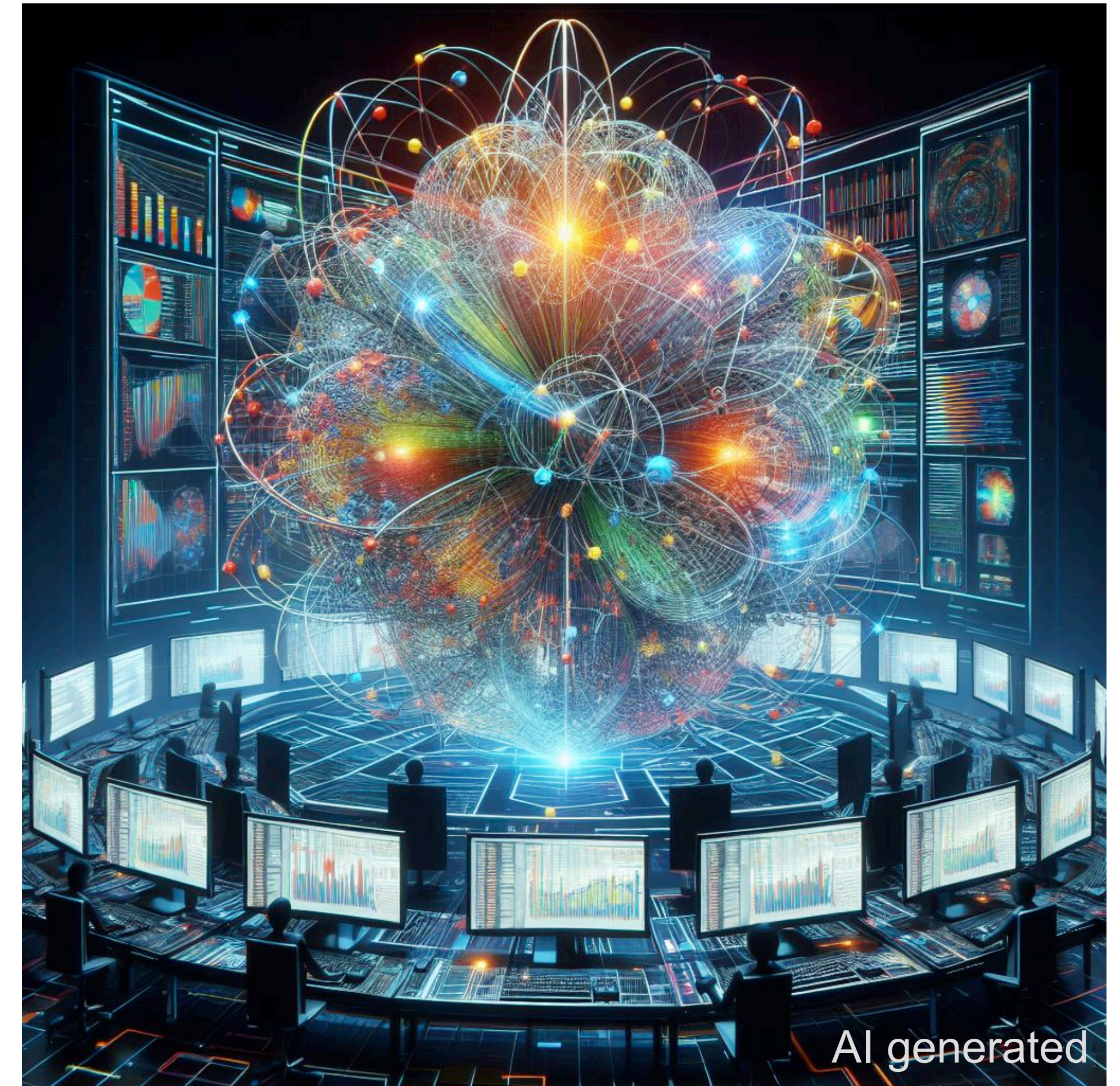
- ~2200 CPU cores, 56 Nvidia GPUs (V100, V100S, A100)
- Each node has a 100Gb/s network connection
- 1PB CEPH disk cache
- 200 Gb/s uplink to background storage and LHCONE/OPN
- HTCondor Batch System
 - Backfilling with WLCG jobs in place
 - Throughput of 1.2 PB per day



2025 - ...: High Throughput Analysis Facility

- (Capital) procurement for next generation analysis cluster started
 - In-kind funding secured thanks to KIT Strategy Funds and ETP
 - DFG proposal to double the in-kind funding in preparation
- Design phase of the cluster has started
 - Based upon HTCondor or K8S
 - CPUs + GPUs
 - NVMe Cache (300 TB)
 - 100 Gb/s per node
 - 4x100 Gb/s uplink
 - Extension of background storage (dCache)

Resources can be shared within ErUM-Data AF consortium



Batch vs. Interactive Utilization



HTCondor

- Most common resource and application scheduler in HEP computing ecosystem
- Users submit their jobs in long queues, preventing interactivensess
- Targets efficient job scheduling maximizing the resource utilization
→ **Cost efficient**



Spark

- Targets **interactive** workload
- Small task granularity (~ 1MB) allows to **scale out** smaller workloads with **low latency**
- Bad resource utilization if no user performs analysis, e.g. due to daytime working hours
→ **Expensive**

Can we have the best of two worlds?

[Slide from Vincenzo Eduardo Pardulano.](#)

Batch vs. Interactive Utilization - Idea

1. Run **HTCondor** ^{or K8S} as the resource **manager** on the **cluster**
2. Get from HTCondor resources quickly if a **user** submits a Spark **application**
3. **Register** these **resources** on the Spark backend side (**YARN**)
4. Share the newly allocated resources of the Spark cluster between **multiple users** to minimize further latency
5. **Give resources back** to HTCondor if the utilization of the Spark cluster drops again

This needs a coordination layer!

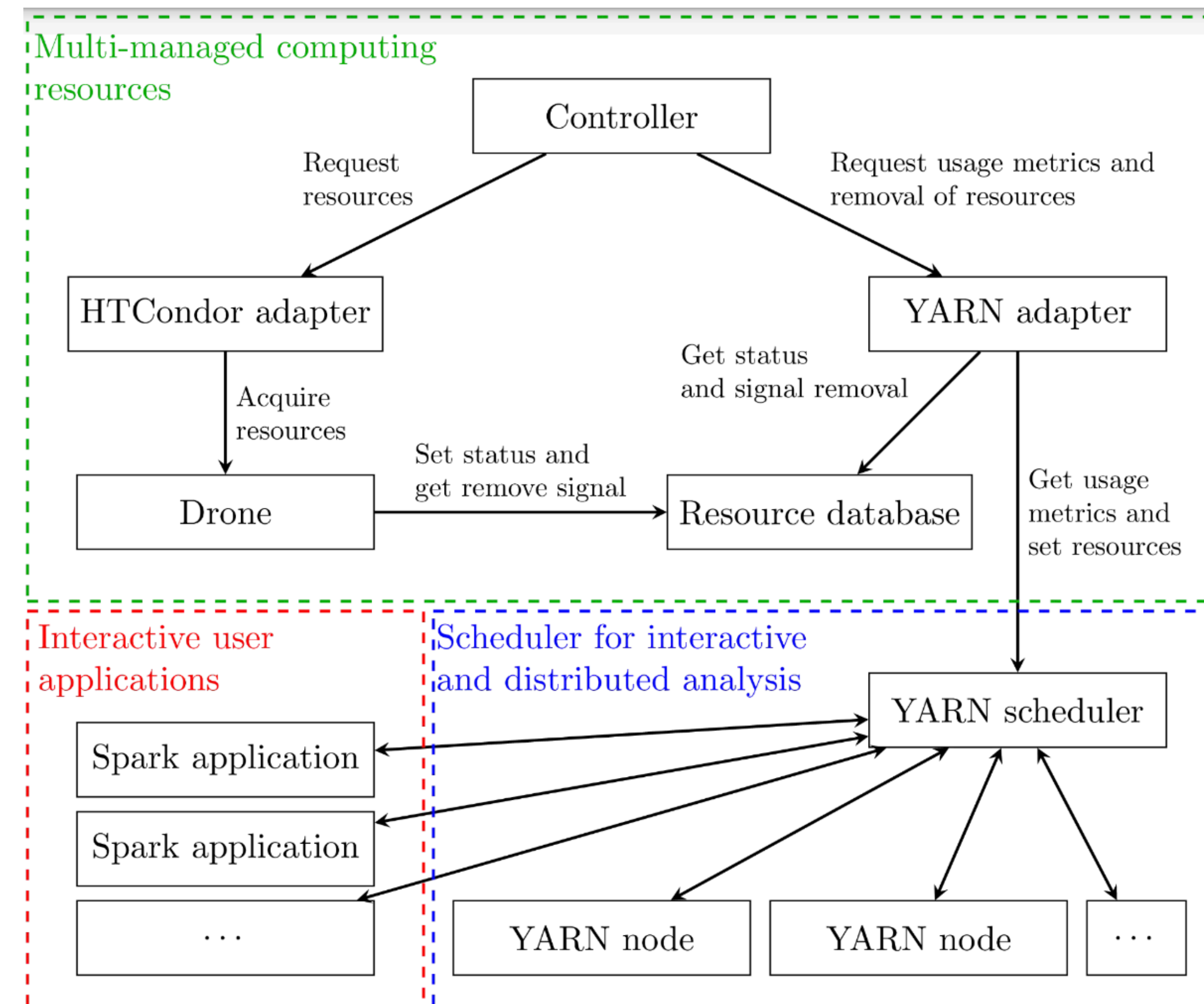
COBaID/TARDIS as Coordination Layer

- A dynamic resource integration system, developed at KIT
- In our use case:
 - It is a **service** running alongside of HTCondor and Spark/YARN
 - It is able to **monitor** and **control** utilization and allocation of physical resources
- Our first (basic) approach at coordinating the cluster resources:
 - **Start good things**
If **user** applications are running, **allocate** resources to Spark
 - **Stop bad things**
Remove idle resources from Spark and **give** them back to HTCondor



Dynamic Interactive Analysis Cluster

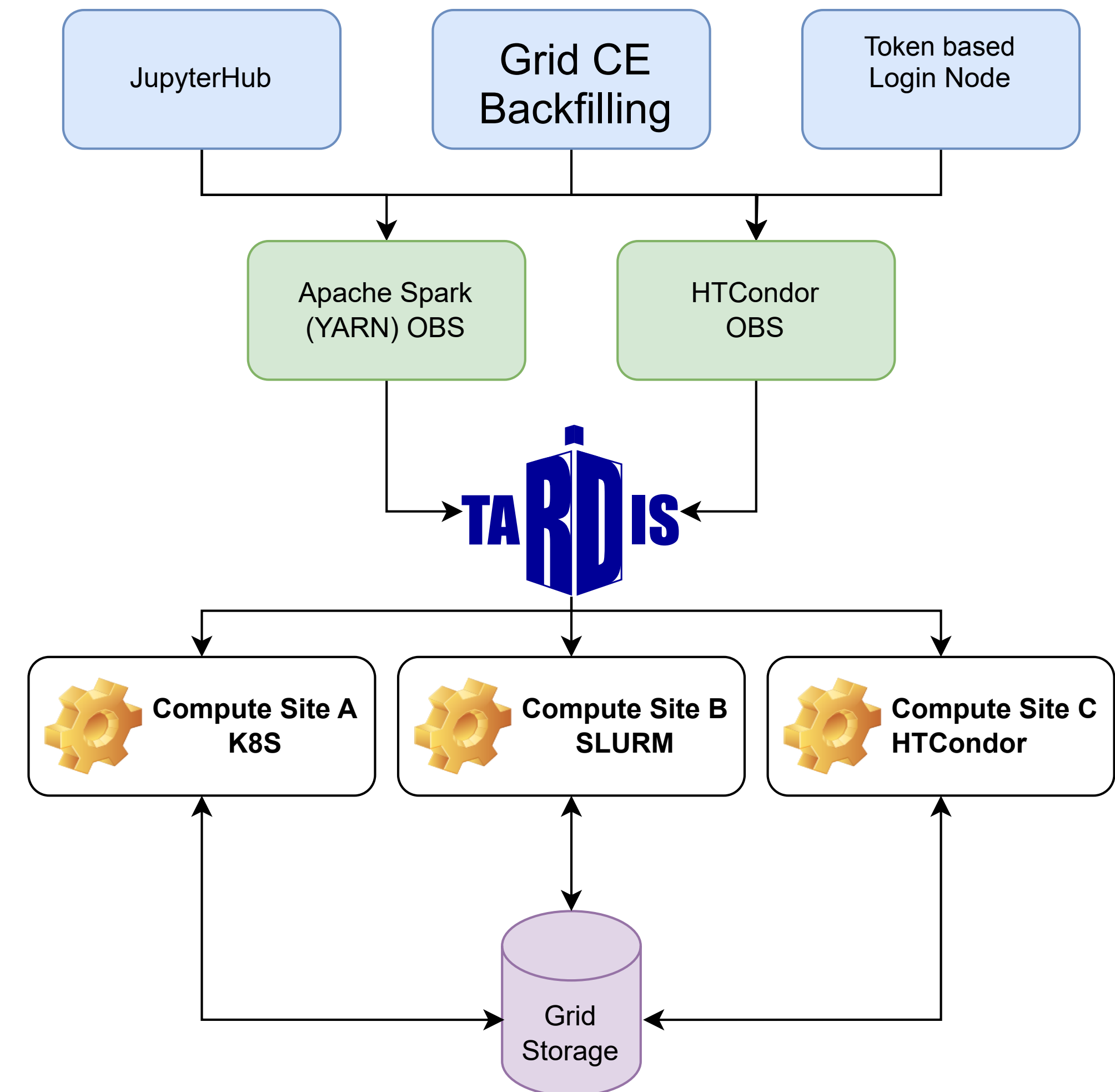
- Find the balance between interactive and batch utilization on analysis clusters
- Idea: Deploy on-demand Apache Spark analysis cluster on HTCondor or K8S
- COBaID/TARDIS perfectly suited to do the balancing depending on utilization and demand
- Why Apache Spark? Cluster can be shared among different users.
- Why not Dask? Each user need its own cluster, no multi-user scheduling possible.



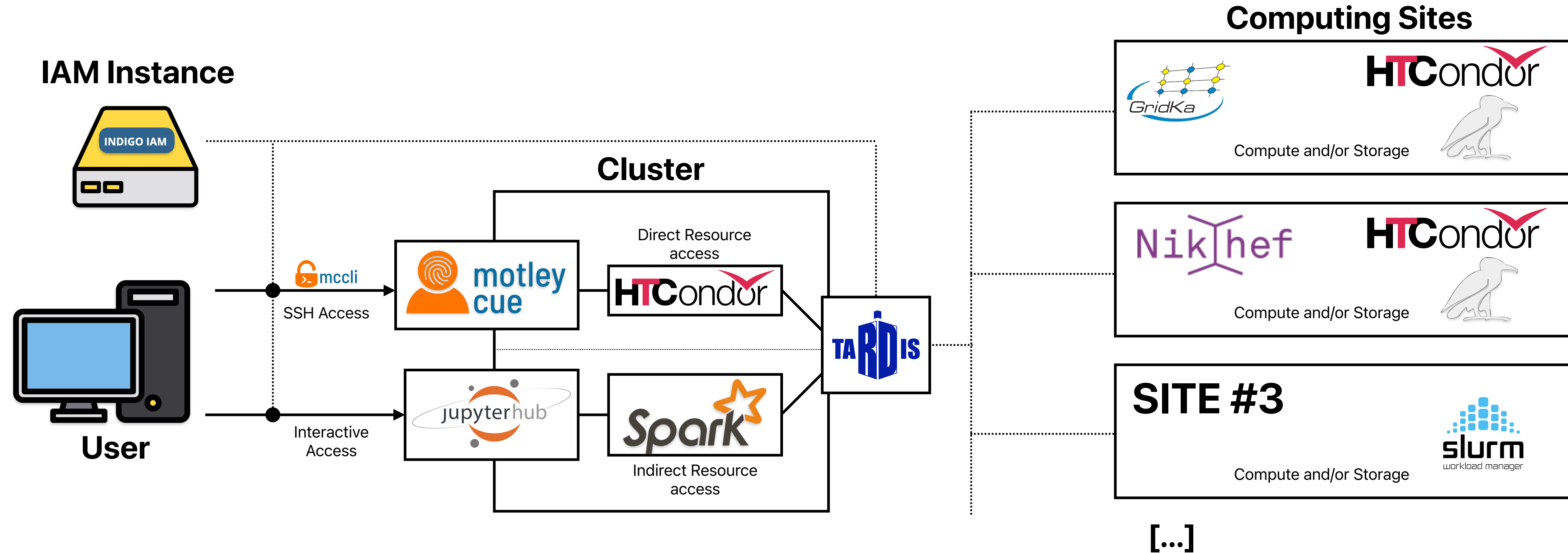
Towards Federated Shared Analysis Facilities

A future interactive analysis infrastructure should ...:

- ... be based on a federated compute infrastructure supplemented by multiple entry points (geographically and method)
- ... allow for dynamic deployment of different kind of clusters, batch (e.g. HTCondor) vs. interactive (e.g. Apache Spark)
- ... allow for fair-sharing resources between different users
- ... provide high network bandwidth to Grid storage systems holding the data



PoC: Computing Infrastructure for DARWIN



See Sebastian's talk this morning!

Summary

- KIT provides and will provide dedicated resources for analysis users
- Focus on dynamic interactive analysis cluster approach using COBaID/TARDIS to solve batch vs. interactive utilization challenge
- KIT resources will be available to project partners in ErUM-Data AF consortium
- Some experiences are available due to a common project with ROOT developers in ErUM-Data IDT-UM
- Recently got a part time PhD student to work on reviving the dynamic interactive analysis cluster approach from this common project



Backup

Why not Dask?

A future interactive analysis infrastructure should ...:

- ... be based on a federated compute infrastructure supplemented by multiple entry points (geographically and method)
- ... allow for dynamic deployment of different kind of clusters, batch (e.g. HTCondor) vs. interactive (e.g. Apache Spark)
- ... allow for fair-sharing resources between different users
- ... provide high network bandwidth to Grid storage systems holding the data

Dask:

- One cluster can only be used by one user
- High potential of idle resources during code development
- Proper culling could reduce inefficient compute utilization, but might annoy users (cluster recreation)
- No averaging effect for large user base