Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Federal Ministry
of Education
and Research

Preparations for the transparent integration of compute resources

status report for Topic Area III

FIDIUM Collaboration Meeting | DESY, Hamburg | 20–21 October 2022

Johannes Haller, Johannes Lange, Daniel Savoiu, Hartmut Stadie

Commitments in FIDIUM

- Topic II – **Data lakes, distributed data, caching**
  - investigate and deploy data caching technologies
  - integrate dynamic data caches near newly integrated CPU resources

- Topic III – **Adaptation, testing, optimization**
  - deploy tools developed within FIDIUM to selected computing centers
  - integrate into production/analysis environments of HEP experiments
  - optimize to requirements for typical analysis workflows

First goals

- integration of ***PHYSnet*** computing cluster at Uni Hamburg
  - run HEP workflows using container solutions to provision software
  - integrate into overlay ***HTCondor*** batch system via ***COBalD/TARDIS***

- testing of user-level tools for deploying analysis to HPC clusters
  - contributions to ***dask-jobqueue*** project for scalable interactive analysis

PHYSnet **cluster**

compute resources shared by all institutes of physics faculty

- heterogeneous, multiple pools/queues for diverse applications:
  - ***idefix.q*** – mixed single-threaded applications
  - ***infinix.q*** – for multi-node applications using MPI + InfiniBand
  - ***obelix.q***, ***epyx.q*** – for large-memory applications
  - ***graphix.q*** – for GPU applications

- parts reserved for exclusive use by various project groups
  - high flexibility for tailoring to individual/group use-cases

- adaptable to HEP workflows using ***containerization*** technologies

up to
10 GE
uplink

~6 PB
hot storage

~675
compute
nodes

~14 400
CPU cores

~125 TB
RAM

[Icons: flaticon.com]

	PHYSnet	**Typical WLCG sites / *NAF***
OS	***Ubuntu***	***RedHat*-based (SLC/CentOS)**
Batch system	***SGE***	***HTCondor***

*(transition to **SLURM** planned for early 2023)*

Containers at *PHYSnet*

Docker

- popular containerization solution
- centralized: system-wide daemon running with elevated privileges
- ***not*** available at *PHYSnet*

Singularity

- containerization solution developed for HPC environments
- can be run without superuser privileges (albeit with restricted functionality)
- ***v3.5.3*** available at *PHYSnet*
- limited interoperability with *Docker* (can run containers based on *Docker* images)

HEP workflows at *PHYSnet*

provision necessary software using containers and CVMFS

- ***Singularity*** container derived from official CERN ***Docker*** image (`cern/cc7-base`)
  - ***gfal2*** libraries installed for grid access
  - grid authentication handled via ***X.509*** user proxy
- provision ***CernVM-File System*** (CVMFS) using ***cvmfsexec***
  - scalable distributed file system designed for software distribution for HEP experiments
  - normally requires superuser privileges, with ***cvmfsexec*** it can be mounted in userspace
  - made accessible inside job containers using **bind-mounts**
- functional setup able to interact with grid storage elements and transfer files
  - ran first performance benchmarks (see report for Topic Area II)
  - *next steps*: test experiment-specific workflows & automate integration using ***COBalD/TARDIS***

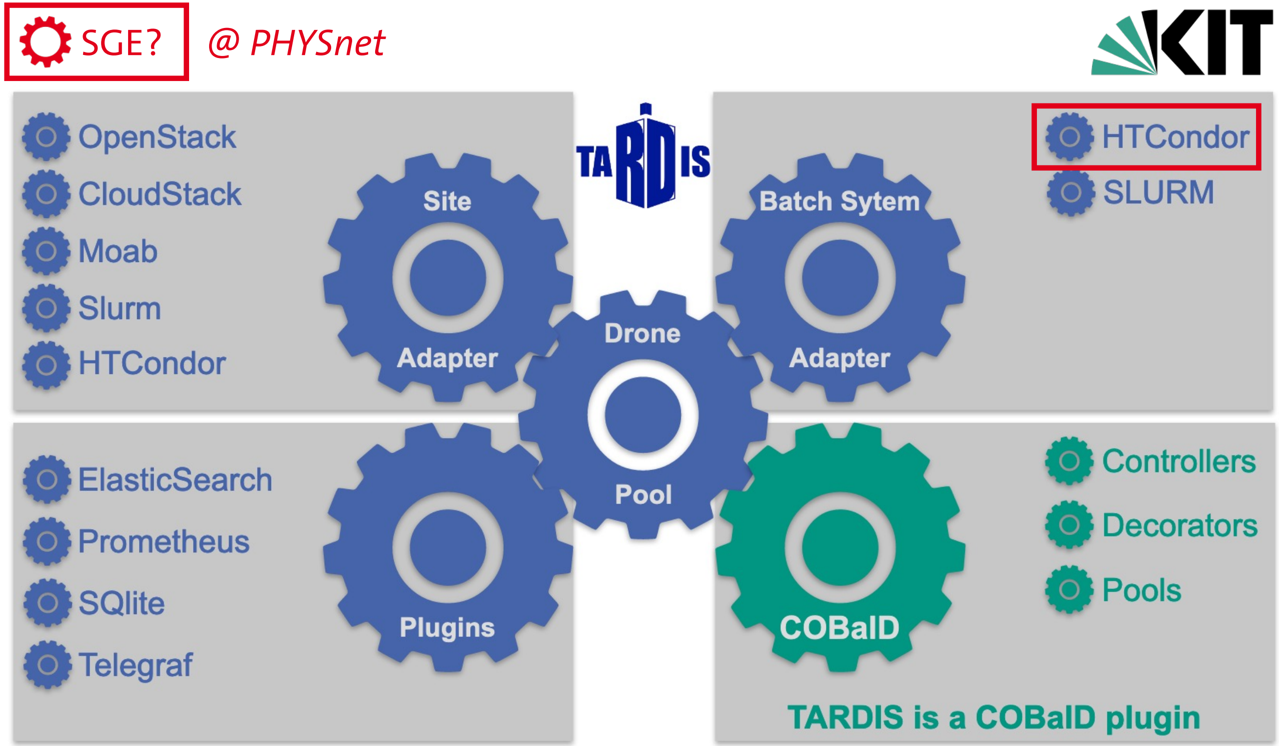
Towards automation with COBalD/TARDIS

on-demand provisioning of resources based on cluster use metrics

+ SGE? @ *PHYSnet*

integrate resources provided by these site types

plugins provide access to external services for e.g. ***monitoring*** and ***persistency***



@ NAF

provision resources to users of these batch systems

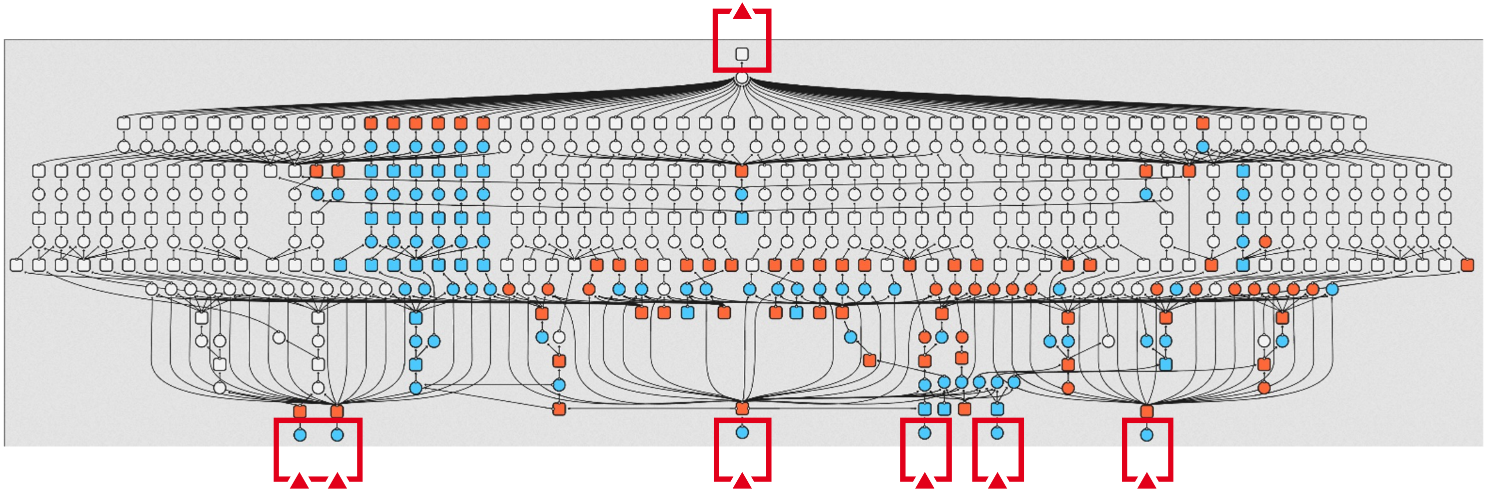
control dynamic provisioning of resources

[M. Giffels, https://indico.scc.kit.edu/event/2291/contributions/8129/attachments/3982/5901/COBalD_TARDIS.pdf]

dask-jobqueue for user analysis

- ***dask***: interactive, scalable parallel computing from Python/Jupyter notebooks using ***numpy***, ***pandas***, etc.
  - hot topic in the context of ongoing analysis facilities efforts & popularity of columnar analysis workflows
  - ***dask-jobqueue***: make it run on batch systems

computation result



data sources

○ pending

🟠 running

🔵 done

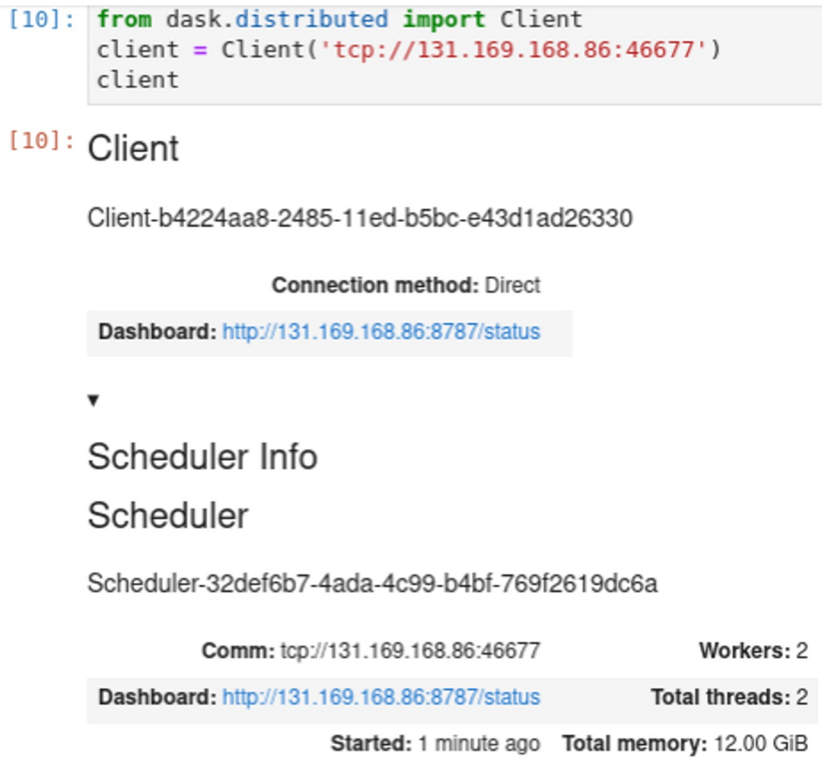
intermediary operations

running on different machines

dask-jobqueue @ NAF

- attempt to make ***dask-jobqueue*** conveniently usable @ ***NAF***
  - 8 merged PRs in upstream with fixes and more clear syntax:
    https://github.com/dask/dask-jobqueue/
  - works on WGS with ***venv***, ***conda***, ***mamba***, etc.
  - can connect to started client from *JupyterHub@NAF* →
  - recommend at least version *0.8.1*

- planned:
  - make it usable directly from *JupyterHub@NAF*
    (not yet configured for job-submission)
  - monitor batch system usage to see if a special treatment (priority) for these jobs is needed
  - ongoing effort to make it run @ ***PHYSNet***
    (individual components work, some work still needed)



```
[10]: from dask.distributed import Client
      client = Client('tcp://131.169.168.86:46677')
      client
```

[10]: Client

Client-b4224aa8-2485-11ed-b5bc-e43d1ad26330

Connection method: Direct

Dashboard: http://131.169.168.86:8787/status

▾

Scheduler Info

Scheduler

Scheduler-32def6b7-4ada-4c99-b4bf-769f2619dc6a

Comm: tcp://131.169.168.86:46677 **Workers:** 2

Dashboard: http://131.169.168.86:8787/status **Total threads:** 2

Started: 1 minute ago **Total memory:** 12.00 GiB

Summary

- developed containerized setup for running HEP-specific software on ***PHYSnet*** cluster
  - successfully provisioned grid access utilities and CVMFS using ***Singularity*** and ***cvmfsexec***
- testing of ***dask-jobqueue*** package for running parallel workflows interactively @ *NAF*
  - deployment at ***PHYSnet*** planned

Next steps

- integration of ***PHYSnet*** resources into overlay batch system using a test ***HTCondor*** cluster
- development of ***TARDIS*** site adapter for ***SGE*** to automate integration process