Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

IDIUM

Federal Ministry of Education and Research

Status of *PHYSnet* cluster integration

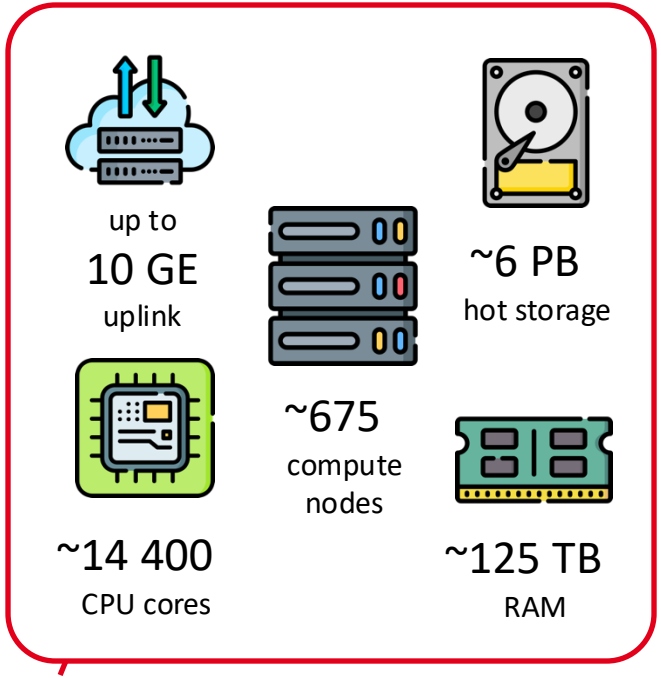
14 November 2024

Johannes Haller, Johannes Lange, Daniel Savoiu, Hartmut Stadie

PHYSnet cluster

compute resources shared by all institutes of physics faculty

- heterogeneous cluster, various 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
  - can integrate dedicated resources for HEP applications
- adaptable to HEP workflows using containerization technologies



[Icons: flaticon.com]

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

**) recently updated from SGE*

FIDIUM project – Federated Digital Infrastructures for Research on Universe and Matter

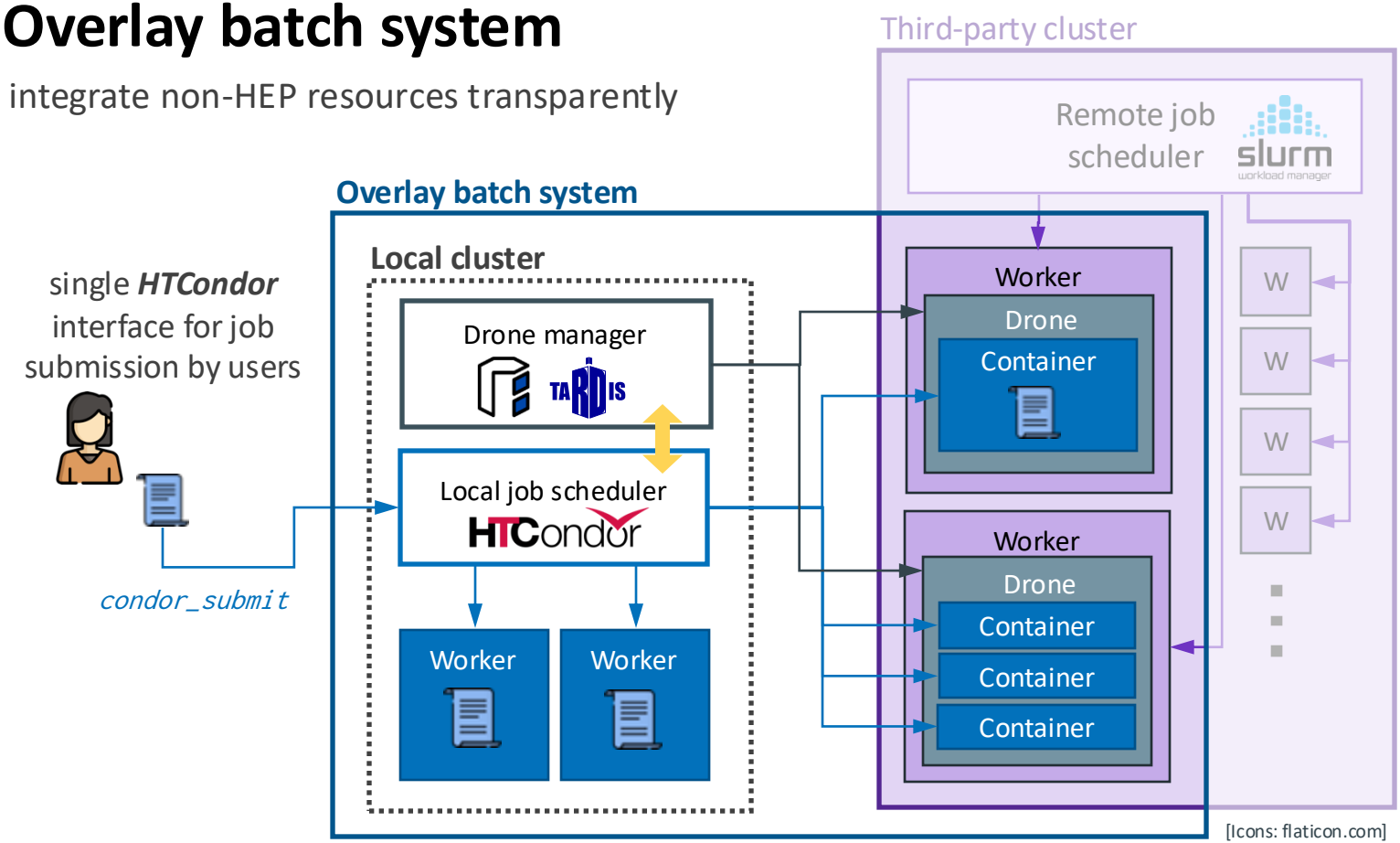
- project funded by the German *Federal Ministry of Education and Research* (BMBF)
- *aim*: develop strategies for handling large amounts of research ***data*** and the associated ***compute*** and ***storage*** needs of its users

U Hamburg commitments

1. **Integration of Opportunistic Resources**
   - enable running HEP workflows on non-HEP-specific resources
   - on-demand provisioning of these resources and integration into the analysis environments of HEP experiments
   - optimize to requirements for typical analysis workflows
2. **Caching**
   - investigate and deploy data caching technologies
   - set up dynamic data caches near newly integrated CPU resources

Overlay batch system

integrate non-HEP resources transparently

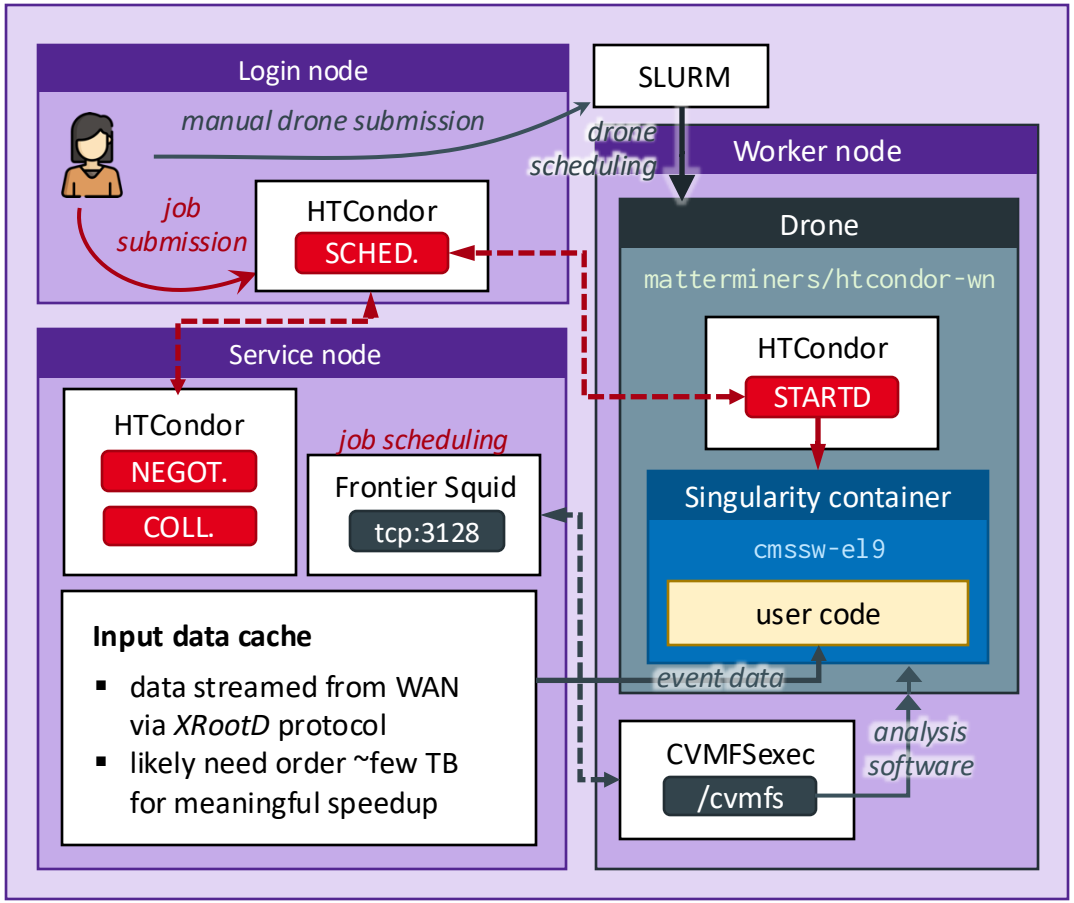


drones for acquiring and holding on to resources

remote jobs run in ***containers*** that emulate familiar environment

Setup at PHYSnet

- working setup for scheduling HEP analysis jobs to ***PHYSnet*** cluster
  - central ***HTCondor*** instance
  - jobs scheduled to drone containers provisioned via native ***SLURM*** batch system
- CVMFS deployed in userspace via ***cvmfsexec***
- unpacked container images taken from `/cvmfs/unpacked.cern.ch`
- ***XRootD*** proxy server for caching input data



HTCondor setup

Service host

- updated to condor **v23.0** *(to match worker node image)*
- system-wide installation, configured with ***central manager***, ***submit*** roles
- authentication via pool password
- **collector** & **negotiator** daemons run here
- **schedd** runs on login node (same subnet as worker nodes)

Drone / Worker node

- ***matterminers/htcondor-wn*** container developed by KIT, provides ***HTCondor*** instance configured with ***execute*** role
  - **startd** runs inside drones & connects to other HTCondor daemons
  - dynamically updated configuration from external git repo using ***condor-git-config***
- by default, jobs run in ***Singularity*** container `cmssw/el9`, from `/cvmfs/unpacked.cern.ch` with bind-mounted `/cvmfs`
  - users can supply their own container

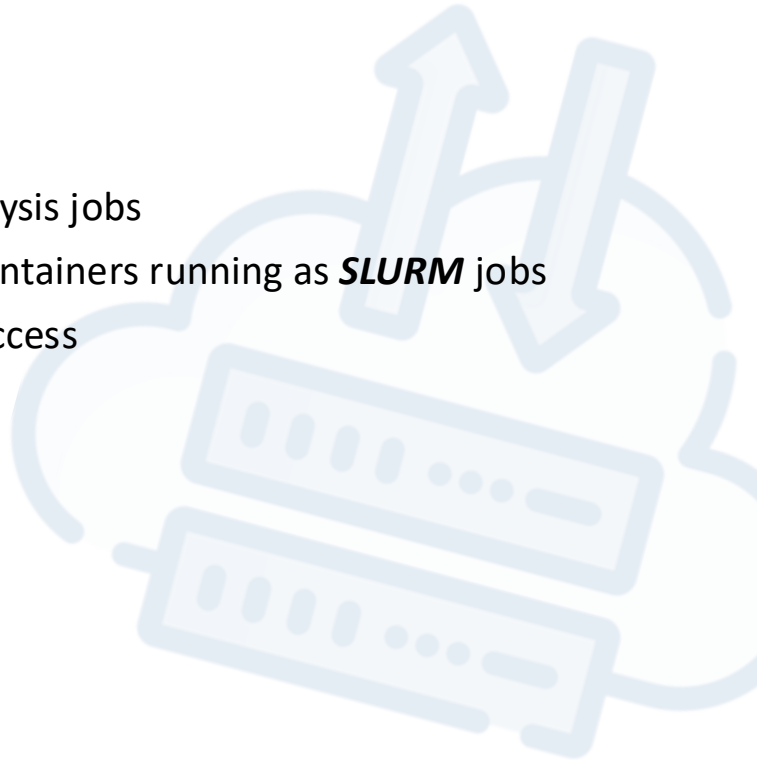
Caching setup

- *aim*: have **disk-based proxy cache** intercept WAN reads from jobs & cache inputs to disk
  - set up local ***XRootD*** server at ***PHYSnet*** running as a proxy cache

    - deployed on service VM as ***Docker*** container, image built on top of ***cmssw/el9***
    - authentication: IGTF host certificate & CERN robot certificate registered w/CMS VO
    - testing ongoing with <1TB disk
    - plan to move to physical machine & expand storage to several TB

Summary

- services were set up at ***PHYSnet*** cluster for running HEP analysis jobs
  - main components: ***HTCondor*** scheduler + worker node containers running as ***SLURM*** jobs
- ***XRootD*** proxy service set up for caching input data on first access



Next steps

- large-scale tests of setup with typical HEP workflows
- evaluate performance of caching with the ***XRootD*** proxy approach
- COBalD/TARDIS, integration into overlay batch system at e.g. NAF

Thank you for your attention!