Karlsruhe Institute of Technology

Institute of Experimental Particle Physics

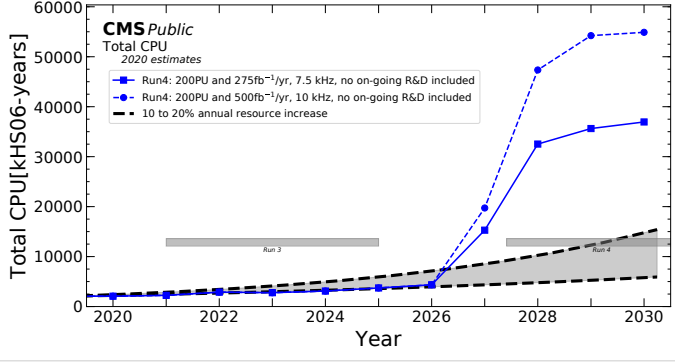
Steinbuch Centre for Computing

Current development on HEP computing at KIT

Matthias J. Schnepf on behalf of the KIT HEP-Computing team | 24. November 2021

Computing is Crucial for HEP

- amount of data from HEP experiments increasing drastically

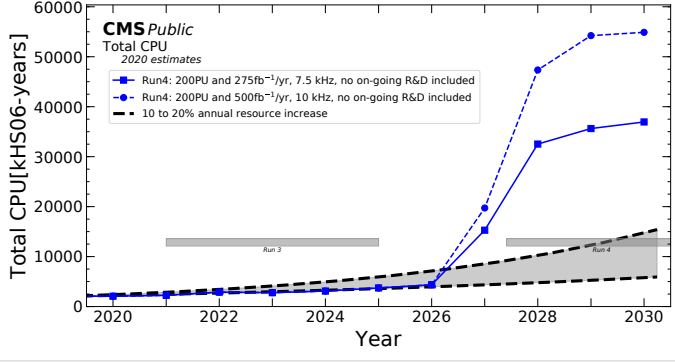


Computing is Crucial for HEP

- amount of data from HEP experiments increasing drastically

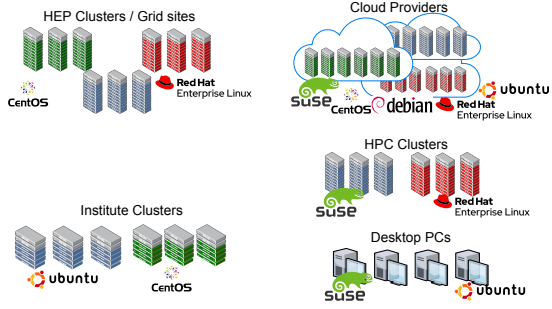
⇒ advancing distributed computing for end-user clusters and Grid needed

  - more computing resources
  - specialised computing resources
  - efficient infrastructure



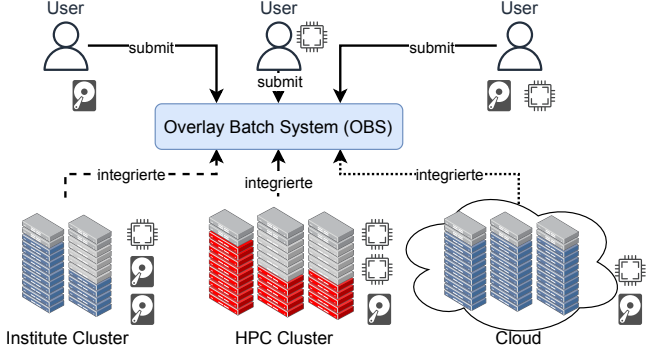
Additional Resources for HEP

- HEP dedicated computing resources
  - institute clusters
  - Grid sites
- resources that are not designed for HEP (opportunistic resources) can be used
  - cloud providers
  - non-HEP Grid sites
  - HPC clusters
  - institute clusters
  - desktop PCs
  - ...
- challenges
  - software environments provisioning
  - dynamic integration
  - transparent usage



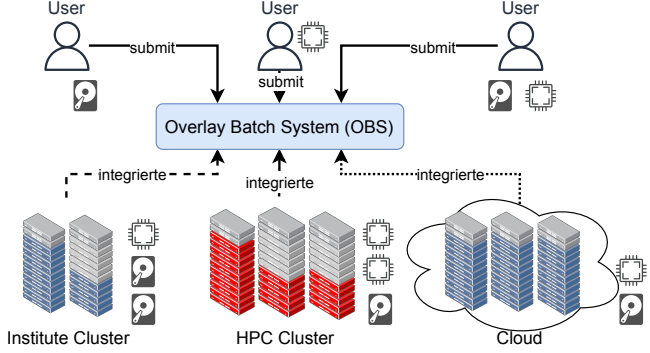
Integration of Resources

- dynamic integration via drones (virtual machine, container, batch job) into OBS
- HEP software environment provided by virtualization and container technology



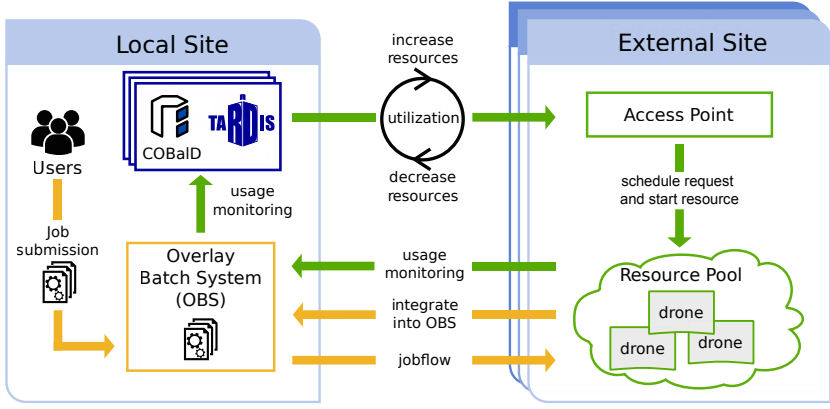
Integration of Resources

- dynamic integration via drones (virtual machine, container, batch job) into OBS
- HEP software environment provided by virtualization and container technology



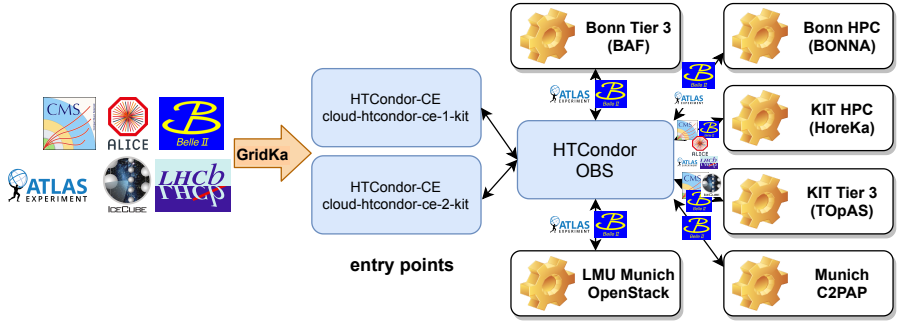
- How many resources of which type are needed at which provider?

Resource Management: COBalD & TARDIS



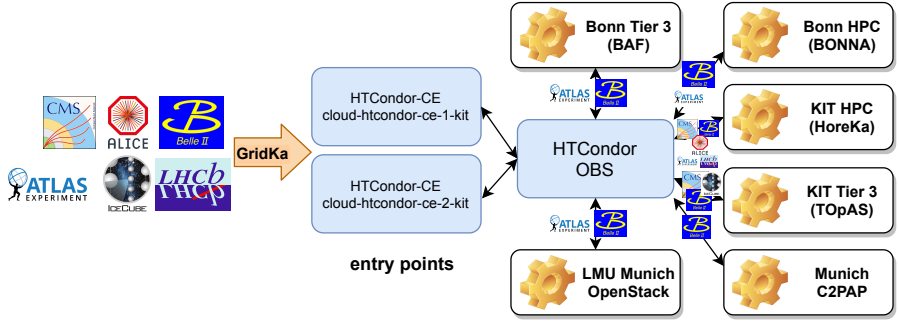
- load balancing daemon COBalD (COBalD - the Opportunistic Balancing Daemon)
- life cycle management TARDIS (Transparent Adaptive Resource Dynamic Integration System)

Computing Resources provided by the "German HEP Cloud"



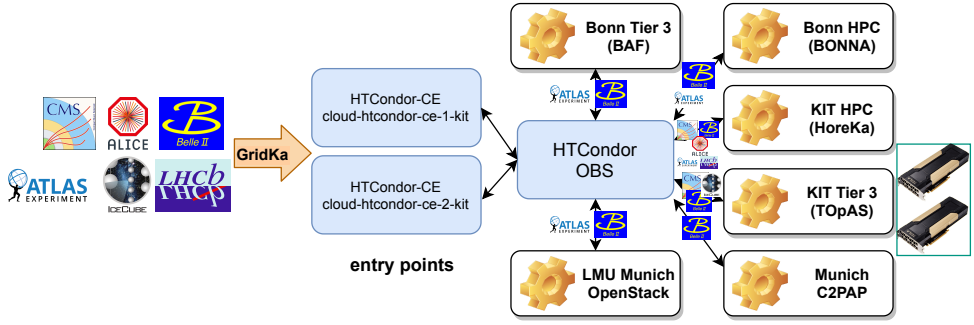
- transparent provisioning of computing resources to specific collaborations, see monitoring
- integration of further resources in the future - fully transparent and experiment independent

Computing Resources provided by the "German HEP Cloud"



- transparent provisioning of computing resources to specific collaborations, see monitoring
- integration of further resources in the future - fully transparent and experiment independent
- further development in optimization and accounting
- Do you want to be part of the growing ecosystem? Contact us

Computing Resources provided by the "German HEP Cloud"



- transparent provisioning of computing resources to specific collaborations, see monitoring (with GPUs)
- integration of further resources in the future - fully transparent and experiment independent
- further development in optimization and accounting
- Do you want to be part of the growing ecosystem? Contact us

GPUs for HEP

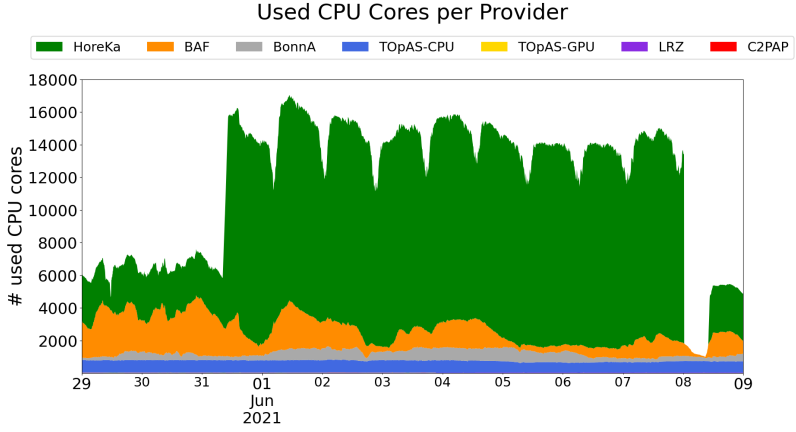
- more and more end-user analyses use GPUs
- GPUs at KIT
  - 8x NVIDIA V100
  - 24x NVIDIA V100s
  - 24x NVIDIA A100
- already in use by ETP via our batch system
- accessible via GridKa cloud CEs

GPUs for HEP

- more and more end-user analyses use GPUs
- GPUs at KIT
  - 8x NVIDIA V100
  - 24x NVIDIA V100s
  - 24x NVIDIA A100
- already in use by ETP via our batch system
- accessible via GridKa cloud CEs
- available via GridKa to test
  - software provision
  - resource demand (memory, CPUs)
  - performance

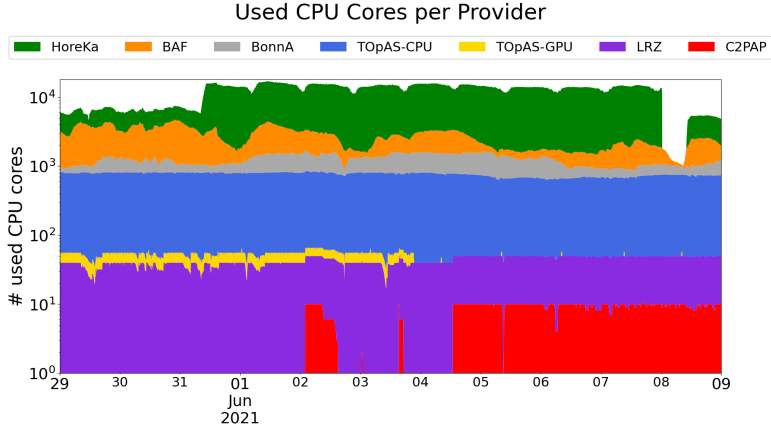
Do you have experience with computing and are interested in using GPUs? Contact us

German HEP Cloud Provided Resources



- up to 17000 CPU cores from 7 providers

German HEP Cloud Provided Resources



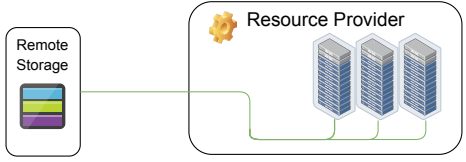
- up to 17000 CPU cores from 7 providers

Data Access in Distributed Systems

- not all providers have permanent HEP storage

⇒ read and write from remote storage

- limited network bandwidth between computing resources and remote storage



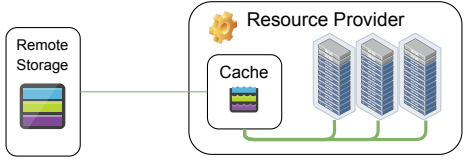
Data Access in Distributed Systems

- not all providers have permanent HEP storage

⇒ read and write from remote storage

- limited network bandwidth between computing resources and remote storage
- most resource providers have local storage

⇒ caches at providers can reduce external network traffic



Data Access in Distributed Systems

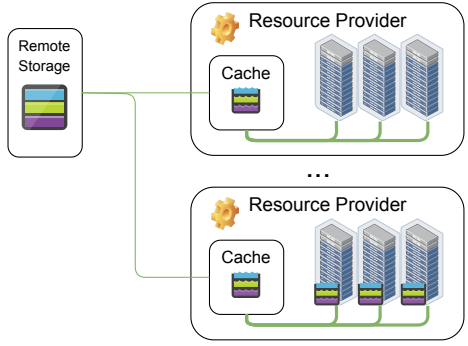
- not all providers have permanent HEP storage

⇒ read and write from remote storage

- limited network bandwidth between computing resources and remote storage
- most resource providers have local storage

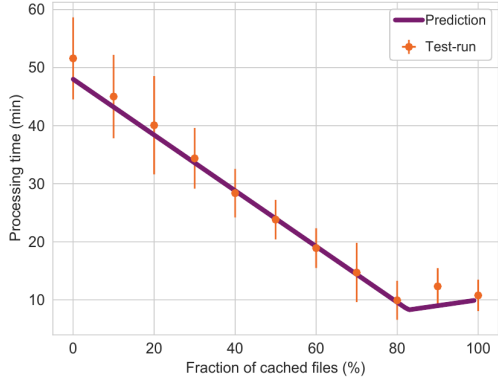
⇒ caches at providers can reduce external network traffic

- more complex in a distributed environment



Caching in a Distributed Environment

- data locality vs. dynamic resources
- maximal throughput for jobs by a combination of caching and read from remote storage

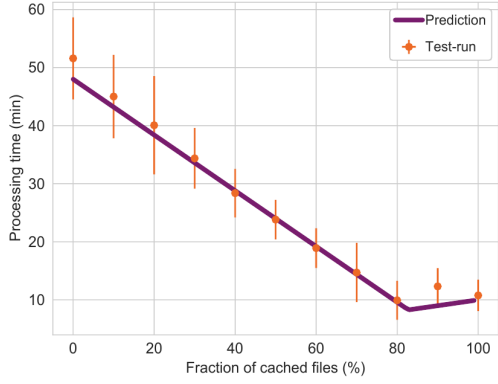


Jet Momentum Resolution for the CMS Experiment and Distributed Data Caching Strategies,

Christoph Heidecker, 2020

Caching in a Distributed Environment

- data locality vs. dynamic resources
- maximal throughput for jobs by a combination of caching and read from remote storage
- caching design studies
  - cache only files from which jobs benefit
  - coordinate caching
    - data location aware job scheduling
    - coordinate data placement
  - simulation to study different scenarios and settings

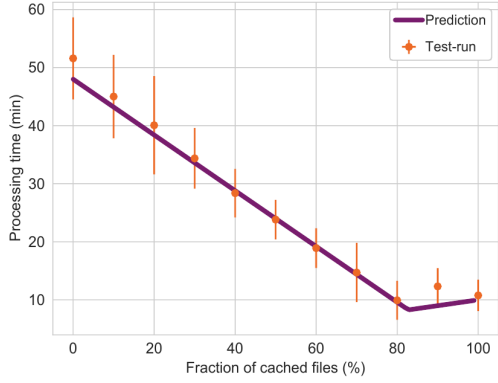


Jet Momentum Resolution for the CMS Experiment and Distributed Data Caching Strategies,

Christoph Heidecker, 2020

Caching in a Distributed Environment

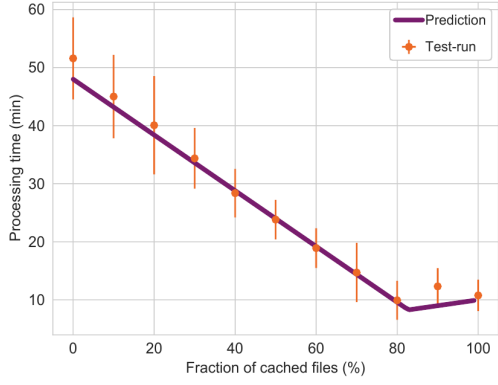
- data locality vs. dynamic resources
- maximal throughput for jobs by a combination of caching and read from remote storage
- caching design studies
  - cache only files from which jobs benefit
  - coordinate caching
    - data location aware job scheduling
    - coordinate data placement
  - simulation to study different scenarios and settings
- integration into experiment Grid infrastructure
  - alternative to full storage for small Grid sites?
  - Grid job scheduler aware of cached data
  - interesting for smaller collaborations such as Belle II



Jet Momentum Resolution for the CMS Experiment and Distributed Data Caching Strategies, Christoph Heidecker, 2020

Caching in a Distributed Environment

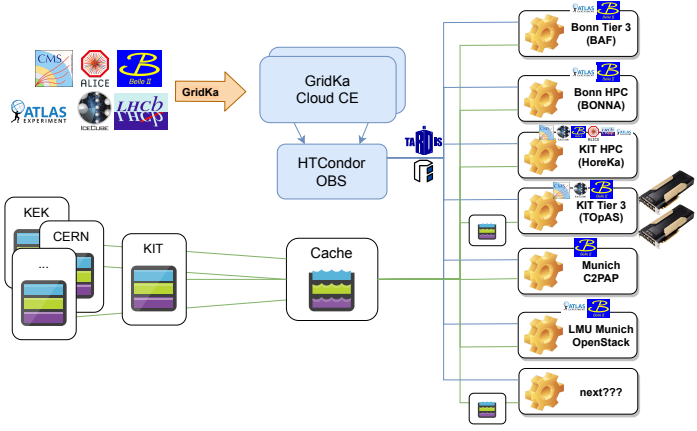
- data locality vs. dynamic resources
- maximal throughput for jobs by a combination of caching and read from remote storage
- caching design studies
  - cache only files from which jobs benefit
  - coordinate caching
    - data location aware job scheduling
    - coordinate data placement
  - simulation to study different scenarios and settings
- integration into experiment Grid infrastructure
  - alternative to full storage for small Grid sites?
  - Grid job scheduler aware of cached data
  - interesting for smaller collaborations such as Belle II
- Are you interested in caching? Contact us



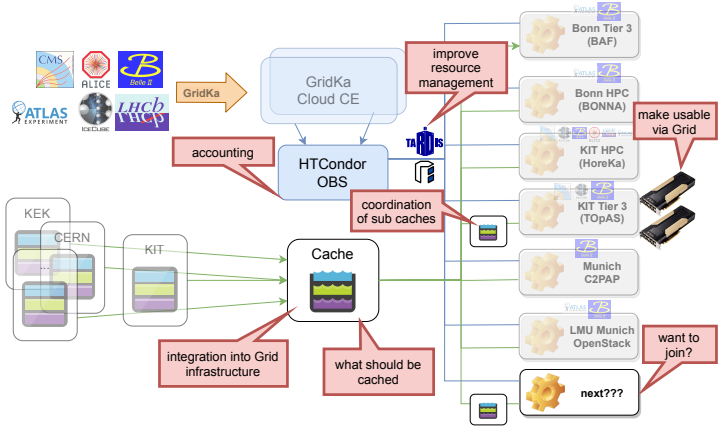
Jet Momentum Resolution for the CMS Experiment and Distributed Data Caching Strategies,

Christoph Heidecker, 2020

German HEP Cloud and Future



German HEP Cloud and Future



Are you interested in one of these topics? Do you want to join the community? Mail to:
matterminers@lists.kit.edu

Backup

KIT as a location for Computing Development

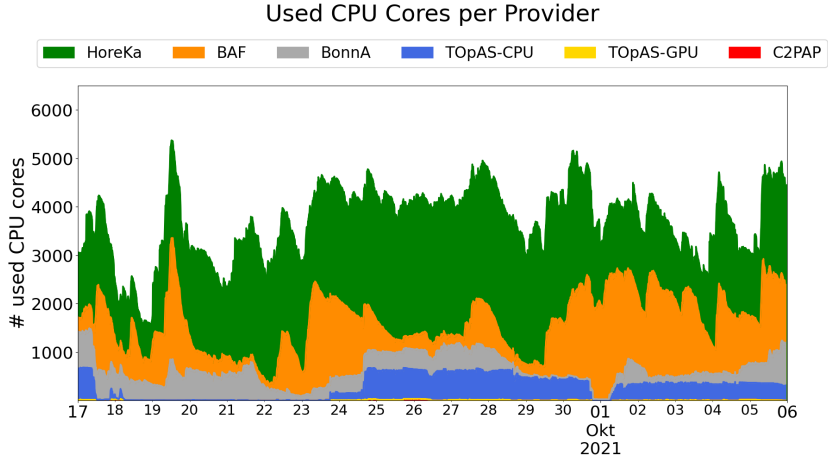
- Institute for Experimental Particle Physics (ETP)
  - big Belle II and CMS group
  - local computing resources and access to HPC clusters
- GridKa
  - biggest WLCG Tier-1 that supports the four big LHC experiments
  - Belle II Raw data centers
  - more than
    - 48.000 CPU cores
    - 45 PB disk storage
    - 63 PB on tape

KIT as a location for Computing Development

- Institute for Experimental Particle Physics (ETP)
  - big Belle II and CMS group
  - local computing resources and access to HPC clusters
- GridKa
  - biggest WLCG Tier-1 that supports the four big LHC experiments
  - Belle II Raw data centers
  - more than
    - 48.000 CPU cores
    - 45 PB disk storage
    - 63 PB on tape

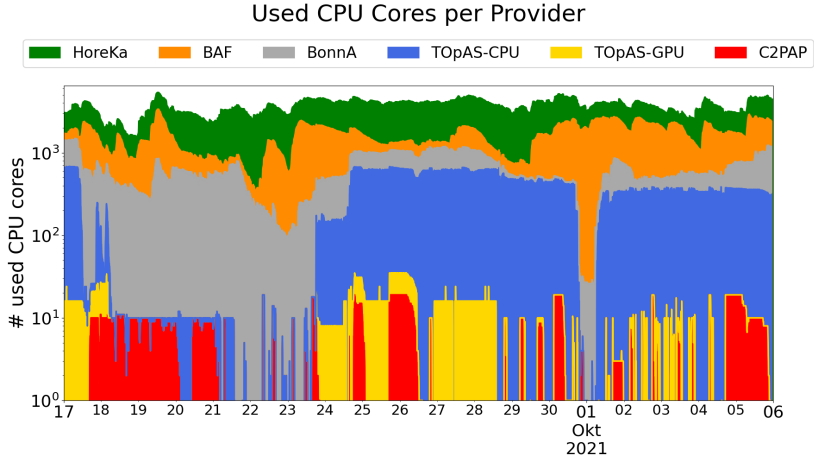
ETP for development and GridKa for large scale and production

Provided Resources



- up to 17000 CPU cores provided by

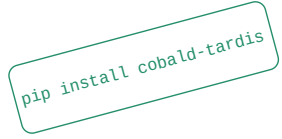
Provided Resources



- up to 17000 CPU cores provided by

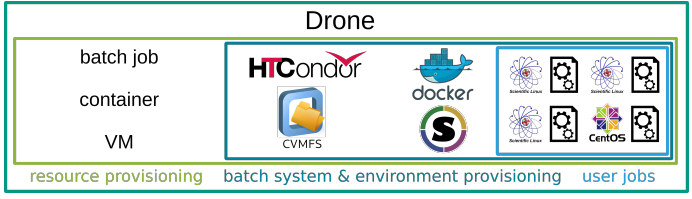
What We Provide

- COBalD & TARDIS
  - https://github.com/MatterMiners/cobald
  - https://github.com/MatterMiners/tardis
- help to setup OBS or integrate site
  - hands on sessions (integration of C2PAP cluster Munich within 4h)
- puppet module
  - https://github.com/unibonn/puppet-cobald
- wlcg-wn container
  - https://hub.docker.com/r/matterminers/wlcg-wn
  - https://github.com/MatterMiners/container-stacks/blob/main/wlcg-wn



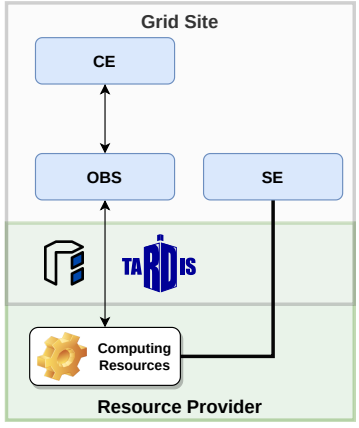
Generalized Pilot Concept

- pilot concept
  - placeholder job allocates resources
  - worker node instance of an Overlay Batch System (OBS) starts payload jobs inside the pilot job
  - requires software environment
- generalized pilot concept ⇒ drone concept
  - resource allocation as
    - batch job
    - virtual machine
    - container
  - provides full Grid software environment
  - drone/pilot/job can run inside a drone

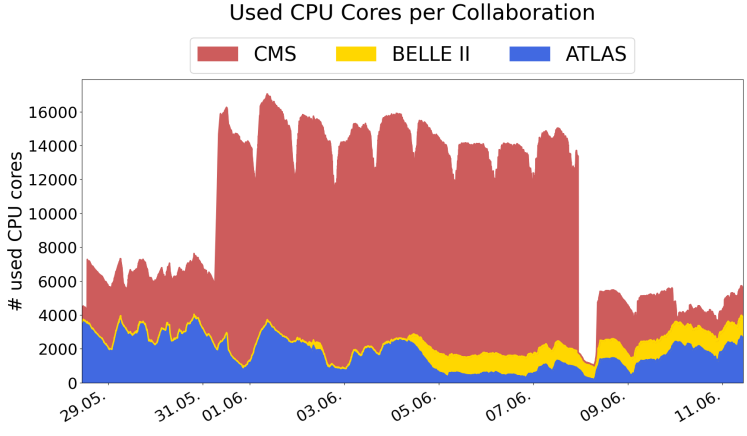


Minimal Setup

- Grid Site
  - standard Grid site services
    - CE
    - OBS for resources
  - provide performant SE and outgoing network
- computing resource provider
  - accessible via HTCondor, Slurm, OpenStack, ...
  - virtualization or container with enables userspace
- COBaID/TARDIS instance
  - lightweight - multiple instances fit on one VM
  - needs just python and resource access
  - instances can be run by Grid site, resource provider, and third party



Provided Resources



- used by several collaborations
- up to 17.400 CPU cores integrated

Supported Providers

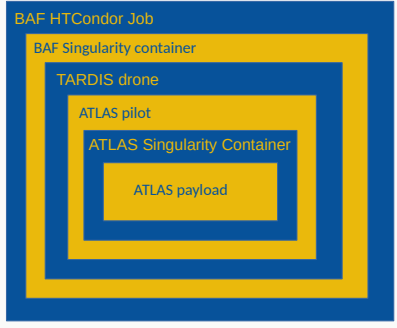
- adapter to interact with provider
- providers
  - HTCondor
  - Moab
  - Slurm
  - CloudStack
  - OpenStack
  - Kubernetes
- further developments are welcome

Pilot inside a Drone

UNIVERSITÄT BONN

JOB STRUCTURE @ U BONN

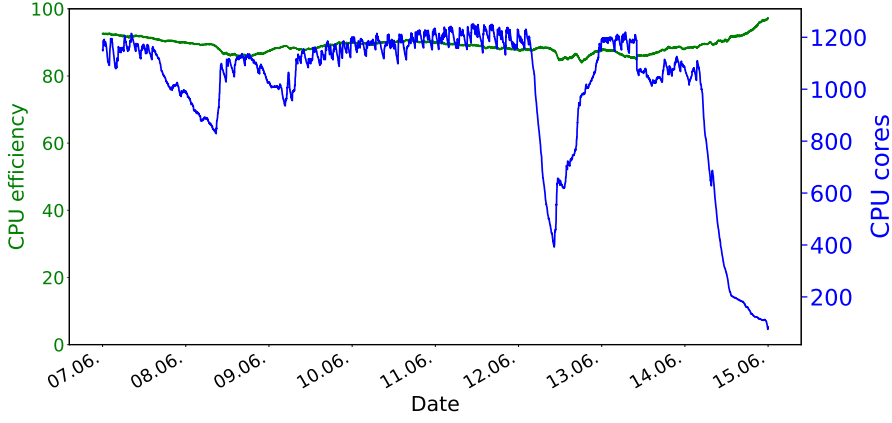
- Nested structure
- BAF containers to decouple cluster operation from user requirements (convenient for operators)
- ATLAS containers to reduce site requirements (convenient for ATLAS)
- ATLAS pilots to improve throughput of ATLAS production system



Peter Wienemann: COBaID/TARDIS @ U Bonn 8

Talk: Opportunistic Resource Mangement with COBaID/TARDIS at U Bonn from Peter Wienemann at the IDT-UM Meeting 30. Sep. 2019: https://indico.physik.uni-muenchen.de/event/22/

Used CPU cores and efficiency for Belle II

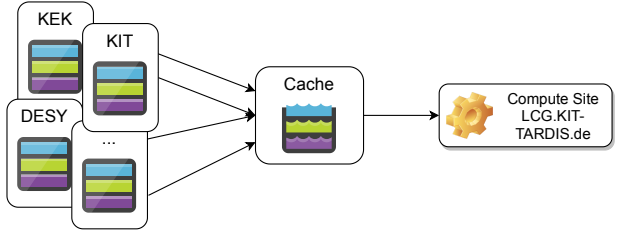


HTCondor Submit file for GPUs at GridKa

```
executable      = test.sh

universe        = grid
grid_resource   = condor cloud-htcondor-ce-2-kit.gridka.de cloud-htcondor-ce-2-kit.gridka.de:9619


request_cpus    = 8
arguments       = foo
request_gpus    = 1
request_memory  = 14000

should_transfer_files   = YES
when_to_transfer_output = ON_EXIT
x509userproxy           = /tmp/x509up_USERID

queue 1
```

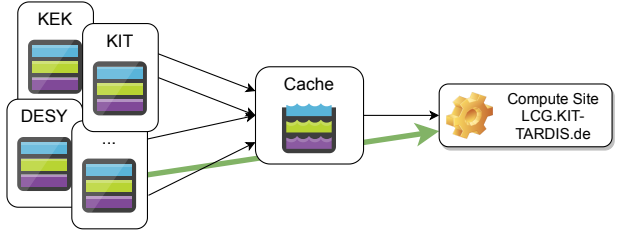
Automated Dataset Copies via Caching

- jobs run only at sites that provide the requested datasets
  - ⇒ long waiting time for jobs
  - ⇒ sites with special hardware need a copy of datasets
- additional copy of datasets via caching (automated copy and cleanup)
- integration into DIRAC currently in development



Automated Dataset Copies via Caching

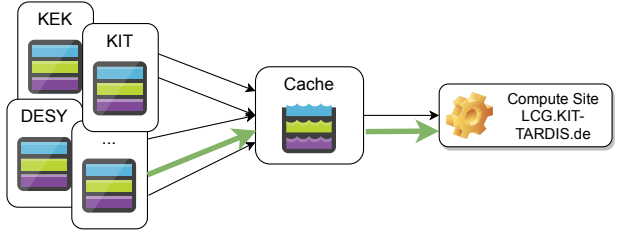
- jobs run only at sites that provide the requested datasets
  - ⇒ long waiting time for jobs
  - ⇒ sites with special hardware need a copy of datasets
- additional copy of datasets via caching (automated copy and cleanup)
- integration into DIRAC currently in development



- read files on demand from specified sites

Automated Dataset Copies via Caching

- jobs run only at sites that provide the requested datasets
  - ⇒ long waiting time for jobs
  - ⇒ sites with special hardware need a copy of datasets
- additional copy of datasets via caching (automated copy and cleanup)
- integration into DIRAC currently in development



- read files on demand from specified sites
- prefetching datasets to the cache before jobs access