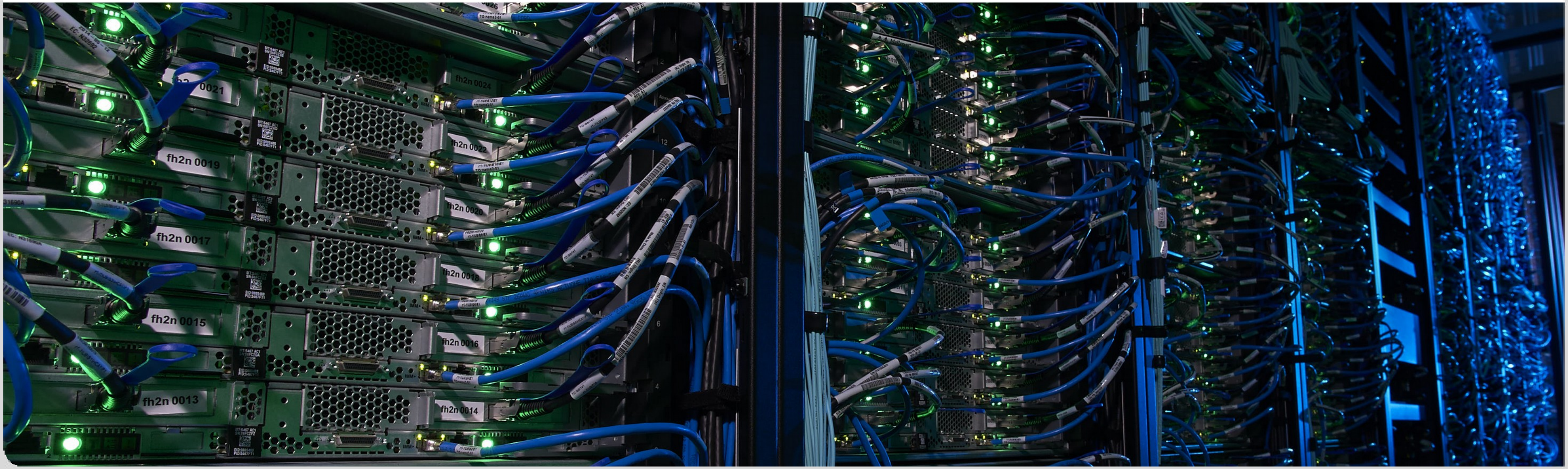
Caching at KIT

René Caspart, Tabea Feßenbecker, Max Fischer, Manuel Giffels,
Christopher Heidecker, Eileen Kühn, Günter Quast

IDT-UM Collaboration Meeting, Karlsruhe – 01.10.2019

STEINBUCH CENTRE FOR COMPUTING (SCC)
INSTITUTE OF EXPERIMENTAL PARTICLE PHYSICS (ETP)



Coordinated Distributed Caching

- Long term experience at KIT on coordinated caching

  Data Locality via Coordinated Caching for Distributed Processing, Max Fischer, Eileen Kuehn at CLOUD COMPUTING 2016 : The Seventh International Conference on Cloud Computing, GRIDs, and Virtualization

- Work motivated by increasing usage of opportunistic resources
  - Often with available local storage
  - Often with limited network bandwidth
  - Jobs reading data from grid SEs saturate bandwidth
  - Focus on enabling efficient processing for recurring jobs

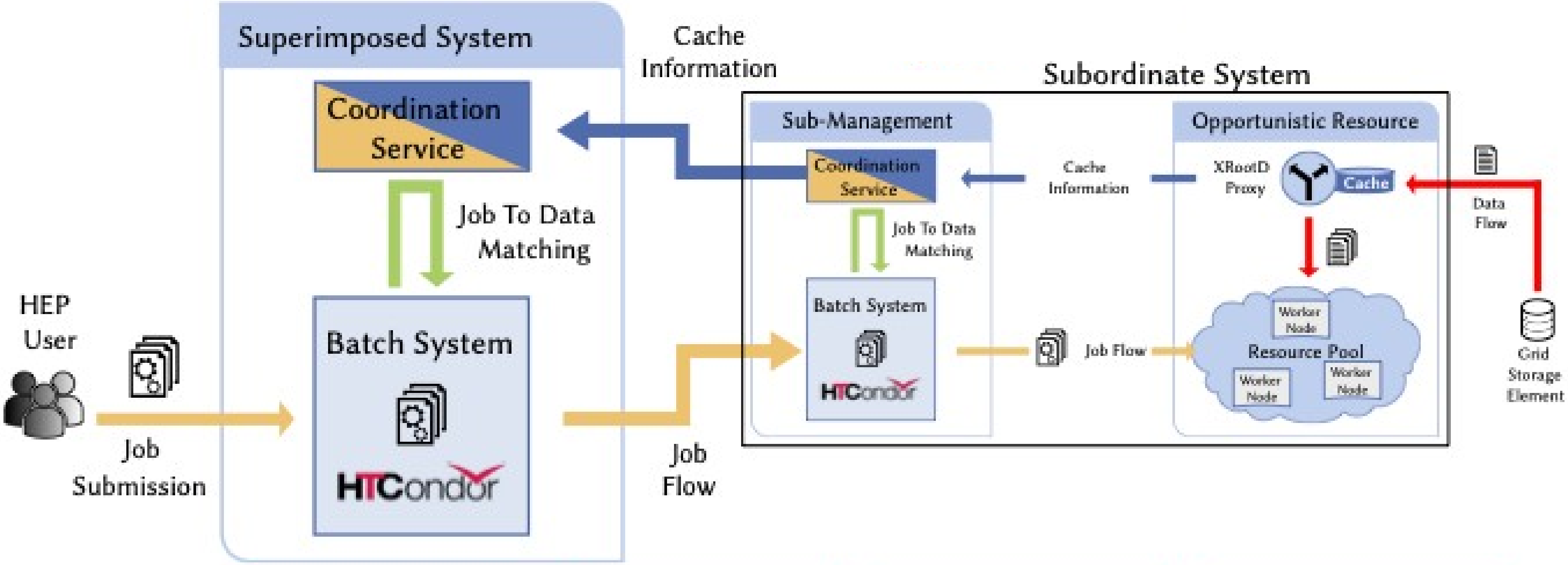
- General idea
  - Reduce data access from jobs running on opportunistic resources
  - Reuse transferred input data via caching
  - Coordinate jobs to resources to use cached data

Technologies for Coordinated Distributed Caching

- Established tools and protocols are suitable for our approach
  - XRootD for data access, caching and cache metadata
  - HTCondor for scheduling and coordinating jobs

- Advantages of these technologies
  - Provide all necessary information
  - Coupling of cache metadata information and batch system scheduling
  - Possibility to setup a hierarchical system to allow for scaling

Hierarchical Structure



... National collaboration | Regional collaboration | Institute resources | Local cluster

In the Scope of Opportunistic Resources

- Caching should be suitable for opportunistic resources
  - Caches can be added on demand
  - No need for permanent caches
  - Keep overhead as low as possible

- Deploying caches on demand
  - Setting up XRootD proxy and manager
  - Register with (central) XRootD manager
  - Querying of current state from XRootD manager

- No need for registering with a dedicated central service/database

Prototype for Coordinated Distributed Caching

- At KIT we developed a first prototype system „NaviX“
- Relying on established tools and protocols
  - XRootD for data access, caching and cache metadata
  - HTCondor for scheduling and coordinating jobs

- Prototype deployed on local and opportunistic resources
  - First experience with the caching system
  - Collect data of data accesses and caching
  - Use data for caching studies

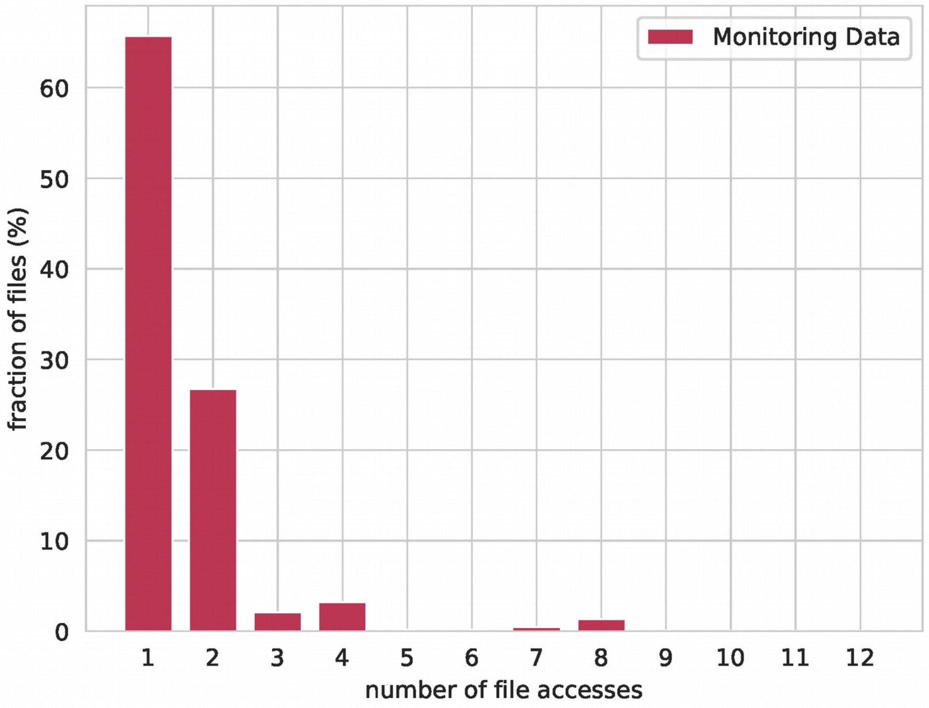
- Learned valuable lessons for building next system
  - Build with scaling in mind
  - Suitable for different data-access and scheduling systems

Current Work and Research

- Currently working on two parts
  - Coordination service acting between XRootD and HTCondor
    - Successor of NaviX
  - Logic for caching and releasing files
    - Currently using a standard XRootD library
      - All accessed data is cached on a per block basis
    - Dynamically decide which data should be cached
      - Studies based on data collected with prototype setup

Studying Caching Logic

- Use data collected with our prototype setup
  - Identify how often files are accessed by jobs
  - Many of the files are only ever read once
  - Naive caching approach is not a sensible choice



Summary

- Caching allows for an efficient usage of resources
  - Coordinated caching well suited for
    - Opportunistic resources
    - Local institute resources
- Setup relies on established technologies
  - XrootD and HTCondor
- Approach is not limited to the WLCG scope

Outlook

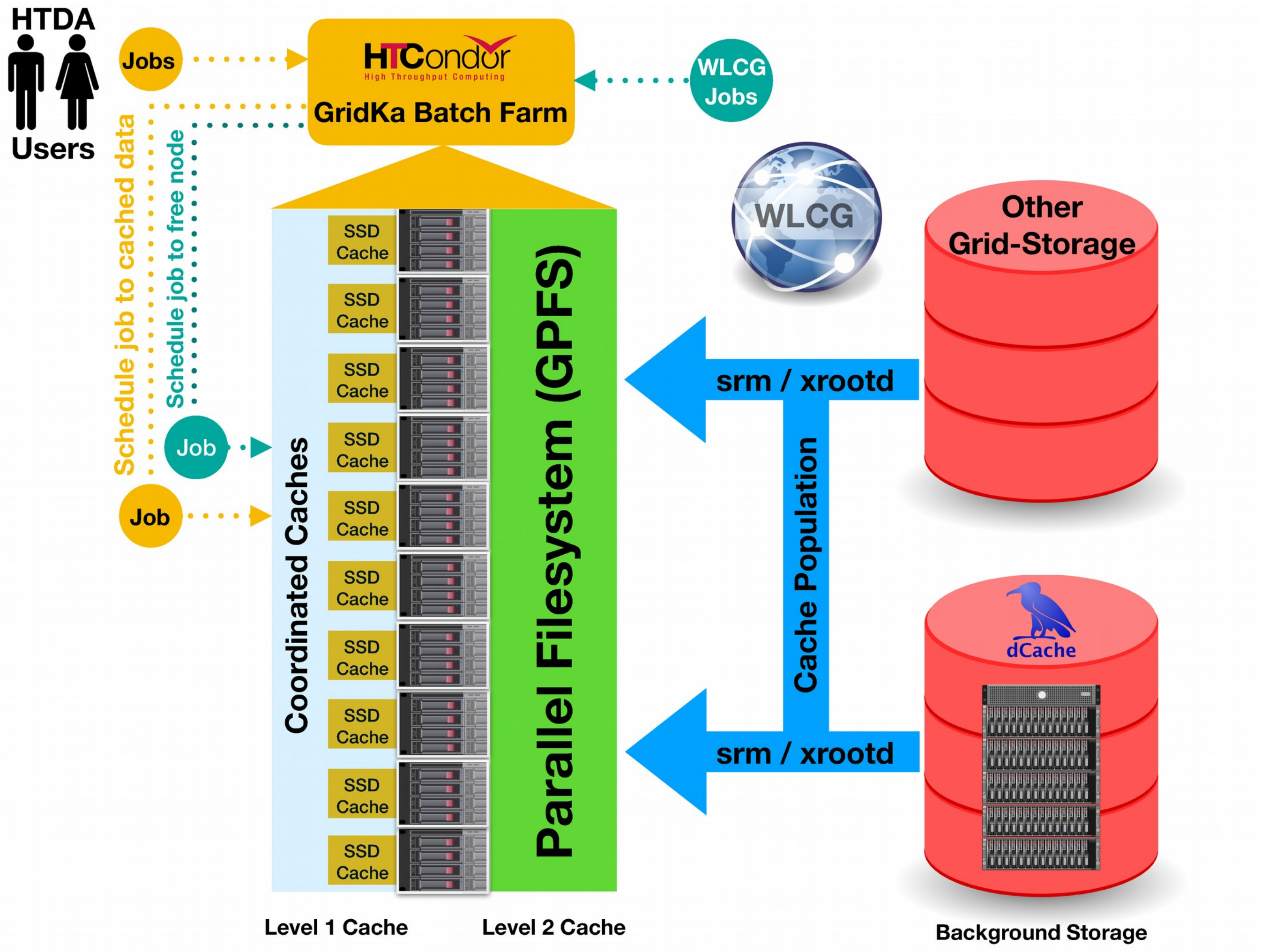
- Development of a coordination service
- Studies to optimize caching decisions

Backup

Test Cluster for Caching

- We operate a dedicated cluster for testing caching solutions in a controlled environment
  - Setup end of 2018
  - Consisting of 11 hyperconvergent workernodes and 1 management/service-node
  - 100 Gbit/s connection between servers, 100 Gbit/s uplink and up to 200 Gbit/s connection to storage at T1_DE_KIT
  - Two levels of caches available
    - ~1 PB distributed file-system span over the workernodes
    - 1 TB NVMe per workernode
    - Allows testing of cascaded caches

Test Cluster for Caching


HTDA
Users
Jobs
HTCondor
High Throughput Computing
GridKa Batch Farm
WLCG Jobs
Schedule job to cached data
Schedule job to free node
Job
Job
SSD Cache
SSD Cache
SSD Cache
SSD Cache
SSD Cache
SSD Cache
SSD Cache
SSD Cache
SSD Cache
SSD Cache
Coordinated Caches
Parallel Filesystem (GPFS)
WLCG
Other Grid-Storage
srm / xrootd
Cache Population
srm / xrootd
dCache
Level 1 Cache
Level 2 Cache
Background Storage