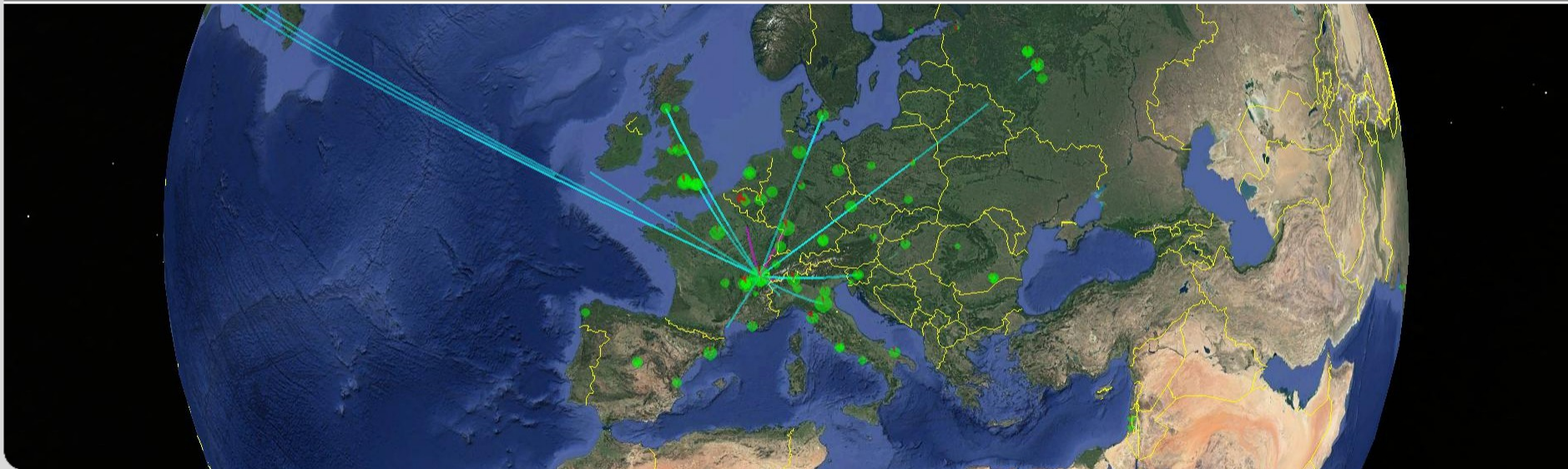


Advancing throughput of HEP analysis work-flows using caching concepts

23rd International Conference on Computing in High Energy and Nuclear Physics | 2018-07-09

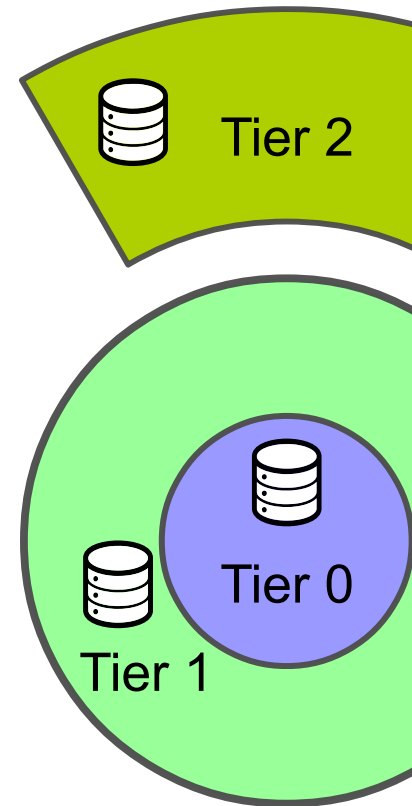
Christoph Heidecker, Max Fischer, Manuel Giffels, Eileen Kühn, Günter Quast, Martin Sauter, Matthias Schnepf

INSTITUTE OF EXPERIMENTAL PARTICLE PHYSICS (ETP) · DEPARTMENT OF PHYSICS



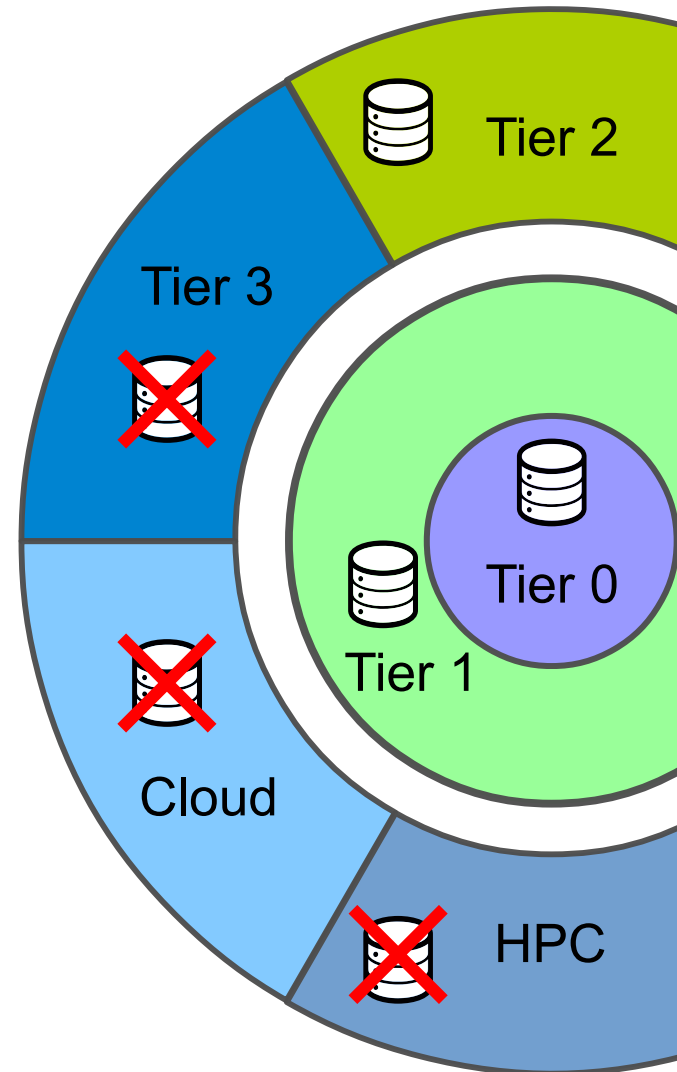
Computing challenge in HEP

- Heterogeneous infrastructure
 - Specialized centers providing Grid storage



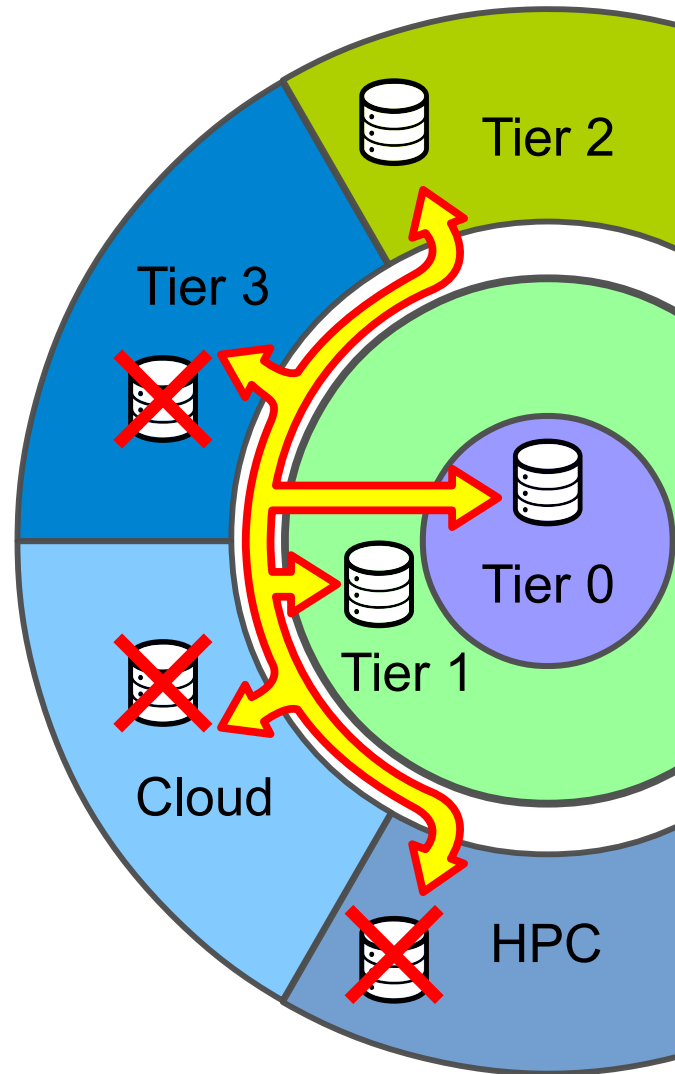
Computing challenge in HEP

- Heterogeneous infrastructure
 - Specialized centers providing Grid storage
 - Diverse computing resources for analysis processing providing no dedicated storage
- Challenges
 - 24/7 operation of Grid storage elements is expensive
 - Concentrate on a few providers
 - Reduce data replication on long term storage



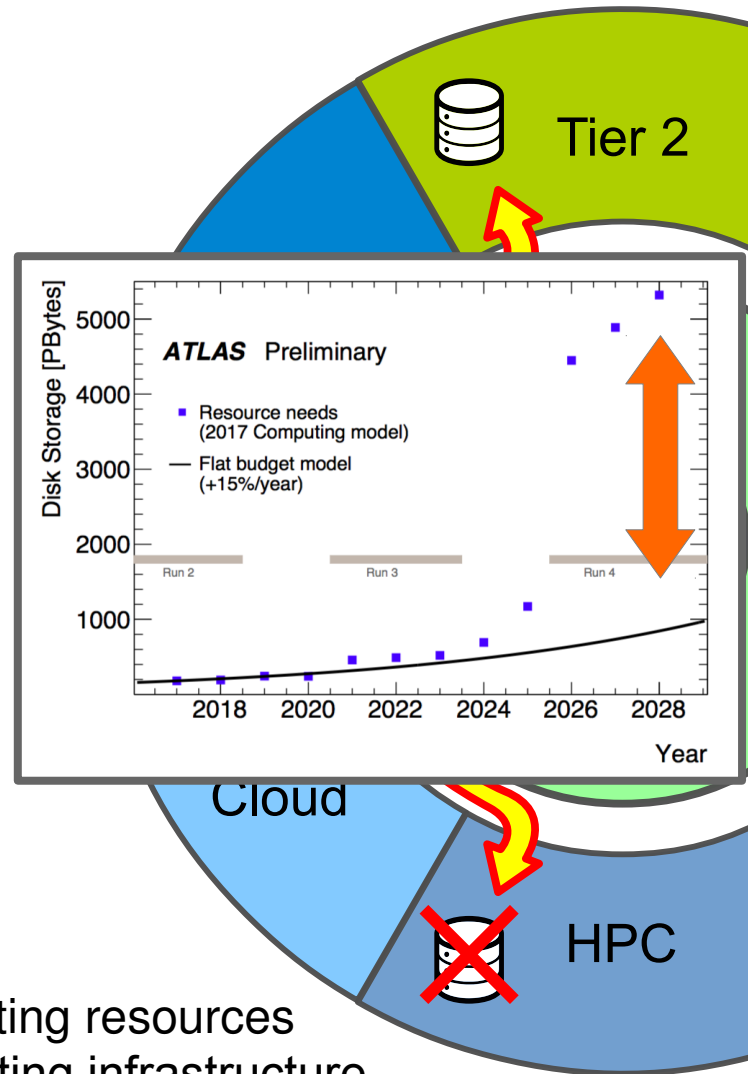
Computing challenge in HEP

- Heterogeneous infrastructure
 - Specialized centers providing Grid storage
 - Diverse computing resources for analysis processing providing no dedicated storage
- Challenges
 - 24/7 operation of Grid storage elements is expensive
 - Concentrate on a few providers
 - Reduce data replication on long term storage
 - Network interconnection is limited
 - inefficient processing of remote data



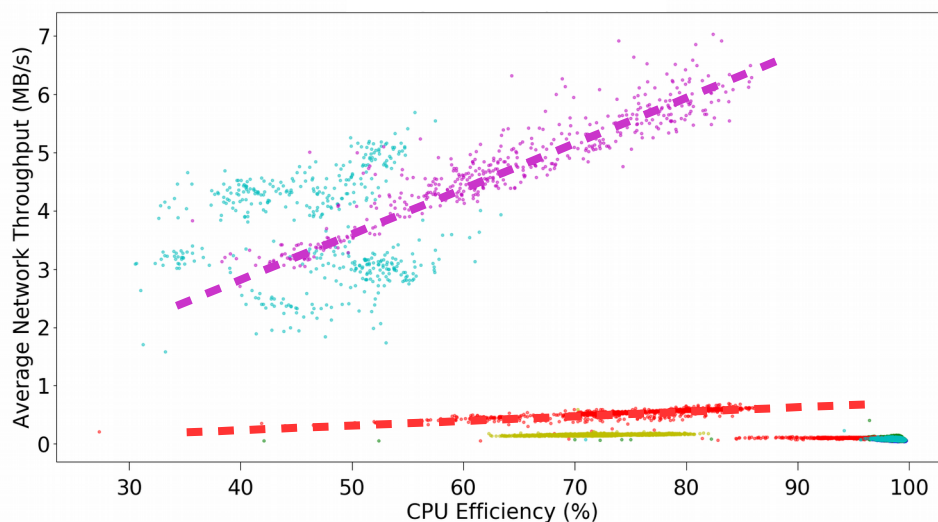
Computing challenge in HEP

- Heterogeneous infrastructure
 - Specialized centers providing Grid storage
 - Diverse computing resources for analysis processing providing no dedicated storage
- Challenges
 - 24/7 operation of Grid storage elements is expensive
 - Concentrate on a few providers
 - Reduce data replication on long term storage
 - Network interconnection is limited
 - inefficient processing of remote data
 - Future HEP experiments cause heavily increasing demand for storage and computing resources
 - Physics results will be limited by computing infrastructure



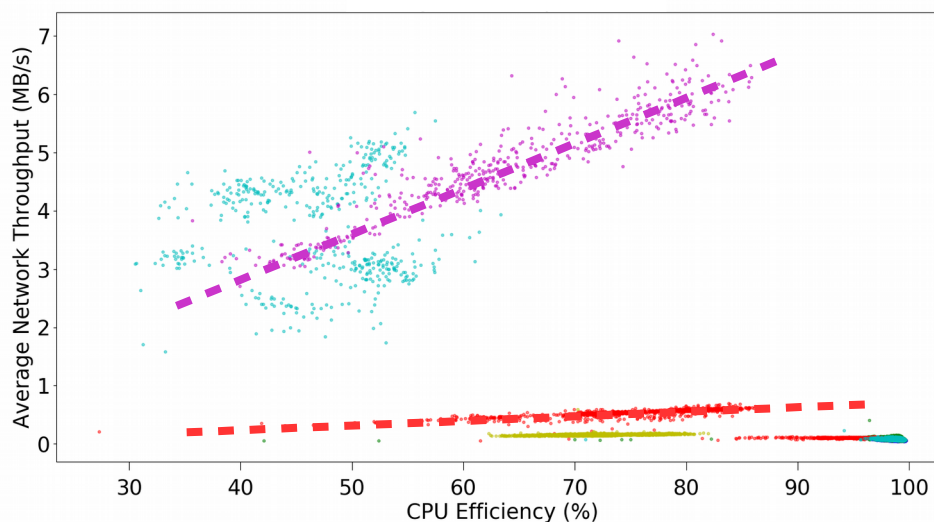
The data access challenge

- User analysis work-flows on computing resources of institute
 - Access data stored on remote Grid storage systems
 - Distribute work-flows to Tier 3, Cloud, and HPC resources for processing.



The data access challenge

- User analysis work-flows on computing resources of institute
 - Access data stored on remote Grid storage systems
 - Distribute work-flows to Tier 3, Cloud, and HPC resources for processing.



- Observed dependency between CPU efficiency and data throughput
 - Processing is limited by available bandwidth
 - Effect is independent of resource type

➔ Data Throughput needs to be optimized!

Data flow optimization

- Our approach:

Coordinated Distributed Caching

- “Naive” caching won’t work on distributed computing resources
→ We need to prevent unnecessary replication of data
- Caches need to communicate building a distributed data system
- Cache content needs to influence the job scheduling
→ Reach data locality by bringing job to most suitable cache

Data flow optimization

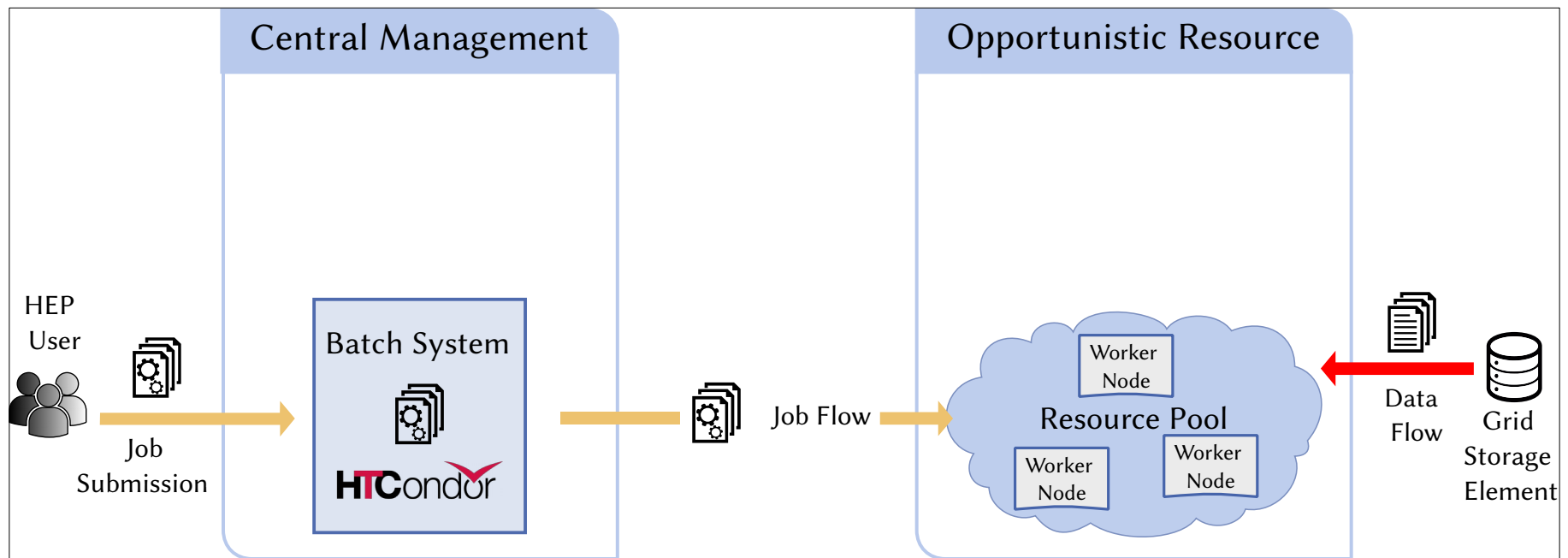
- Our approach:

Coordinated Distributed Caching


- “Naive” caching won’t work on distributed computing resources
→ We need to prevent unnecessary replication of data
- Caches need to communicate building a distributed data system
- Cache content needs to influence the job scheduling
→ Reach data locality by bringing job to most suitable cache
- Concept is suitable
 - For HEP workflows that process same datasets repeatedly
 - For optimization of distributed resources with no or permanent storage
- Challenge: **Transparent integration** into current infrastructure
 - Support HEP data transfer protocols
 - Automatically coordinate without user interaction

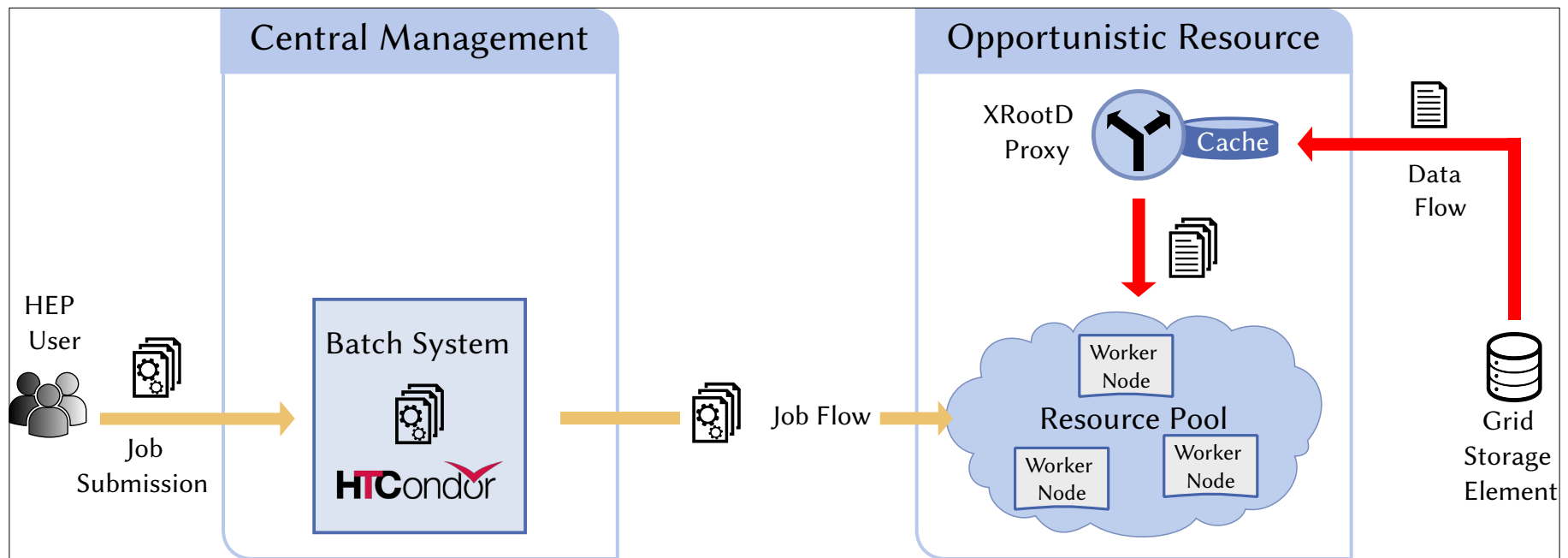
Integration of caching into HEP infrastructure

- Basic features are provided by
 - **HTCondor** that handles jobs to resource scheduling



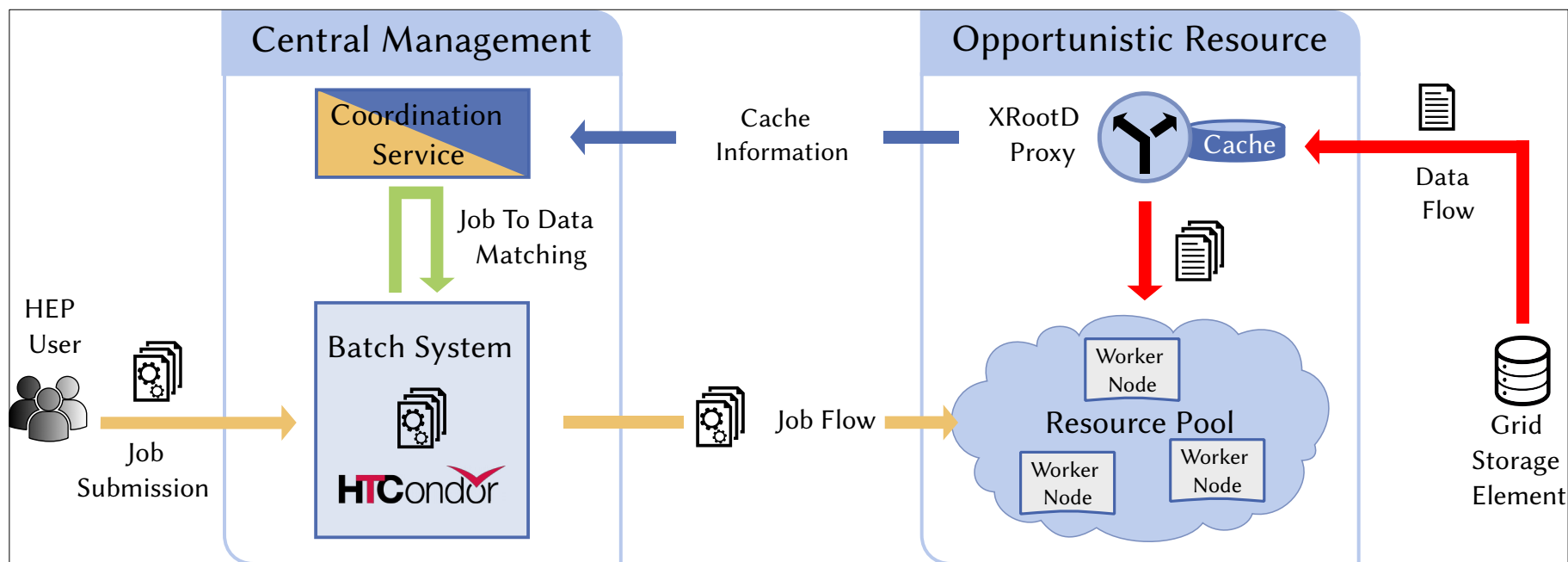
Integration of caching into HEP infrastructure

- Basic features are provided by
 - **HTCondor** that handles jobs to resource scheduling
 -  XRootD that already provides basic caching functionality



Integration of caching into HEP infrastructure

- Basic features are provided by
 - **HTC**ondor that handles jobs to resource scheduling
 - XRootD that already provides basic caching functionality



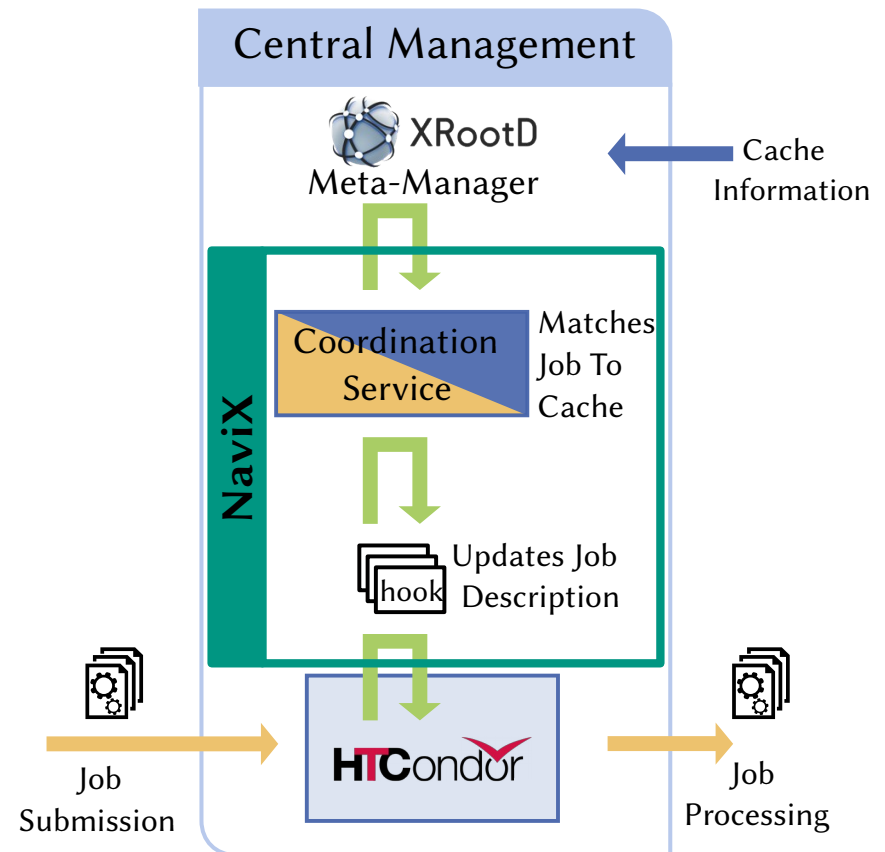
- We developed a **Coordination Service** that
 - matches jobs to the most suitable resource/cache
 - influences data placement via job scheduling

Coordination service: NaviX

- New development based on long-time expertise

Data Locality via Coordinated Caching for Distributed Processing, M Fischer et al. 2016 J. Phys.: Conf. Ser.762 012011

- Extension of existing HTCondor and XRootD components

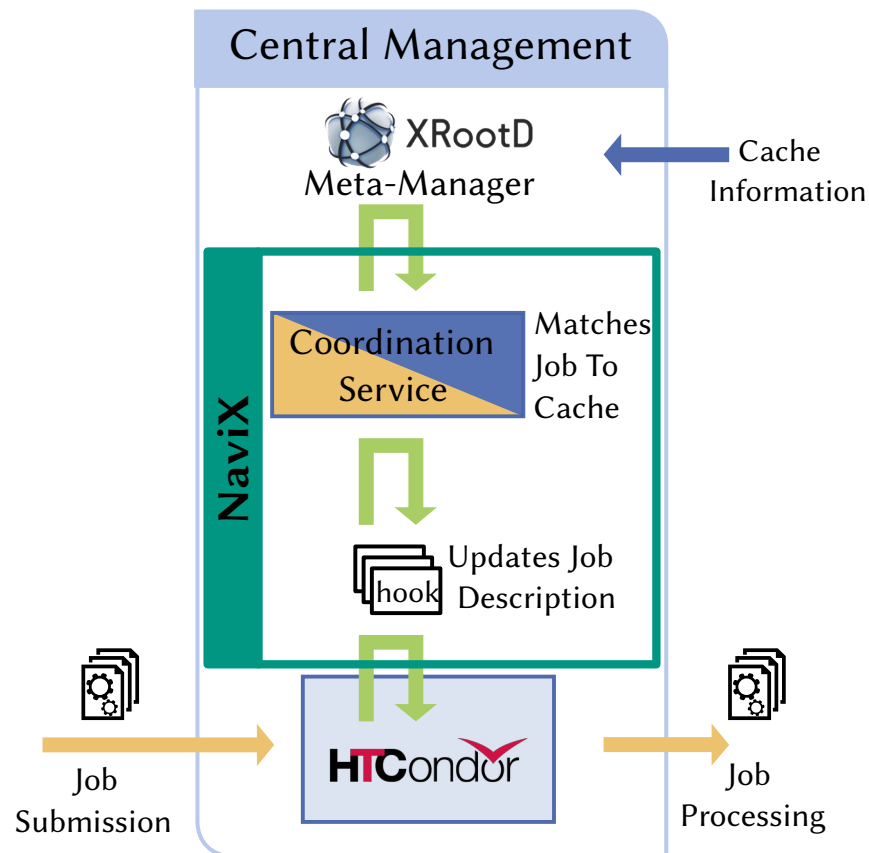


Coordination service: NaviX

- New development based on long-time expertise

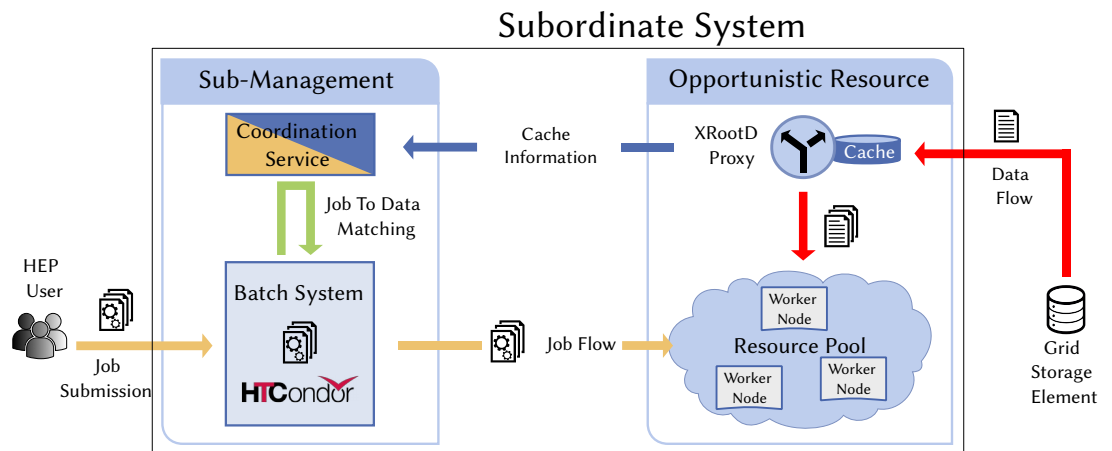
Data Locality via Coordinated Caching for Distributed Processing, M Fischer et al. 2016 J. Phys.: Conf. Ser.762 012011

- Extension of existing HTCondor and XRootD components
- Coordination service matches XRootD cache information to HTCondor job description
- Hooks reconfigure job description and thus influence HTCondor scheduling
- NaviX enables monitoring of data accesses, caches and jobs



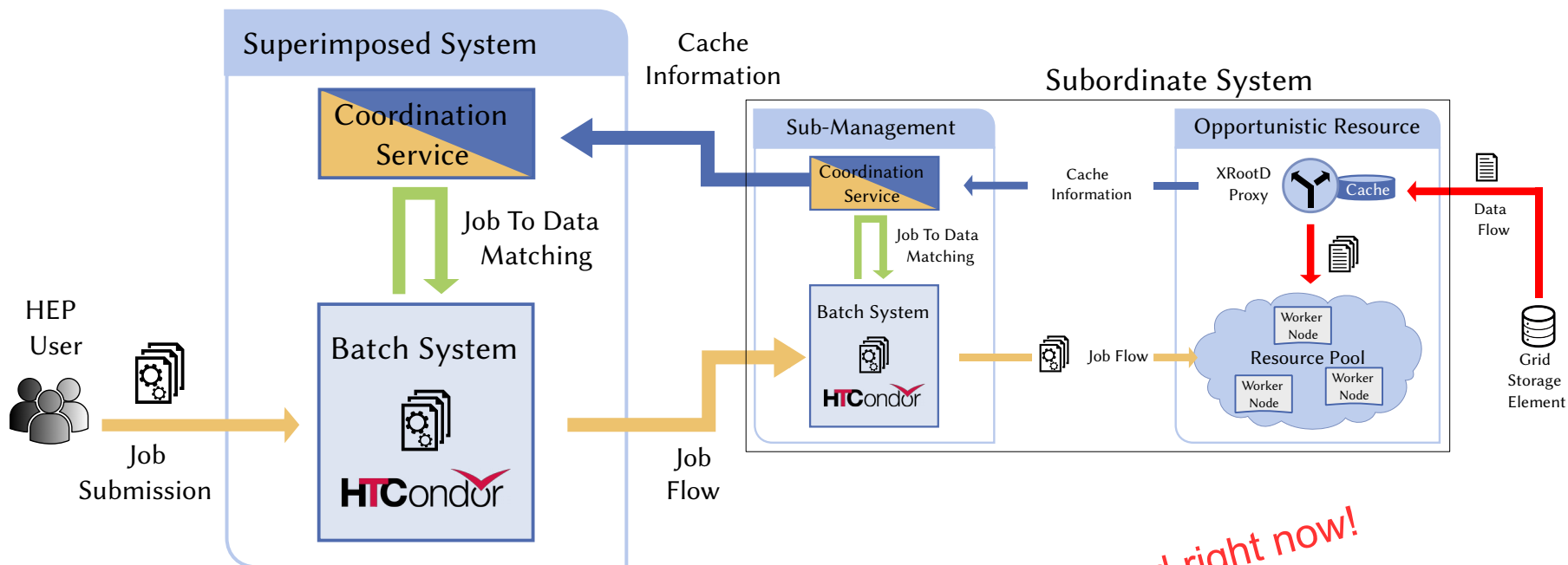
Scalability of XRootD-HTCondor caching

- XRootD and HTCondor take care of hierarchical upscaling



Scalability of XRootD-HTCondor caching

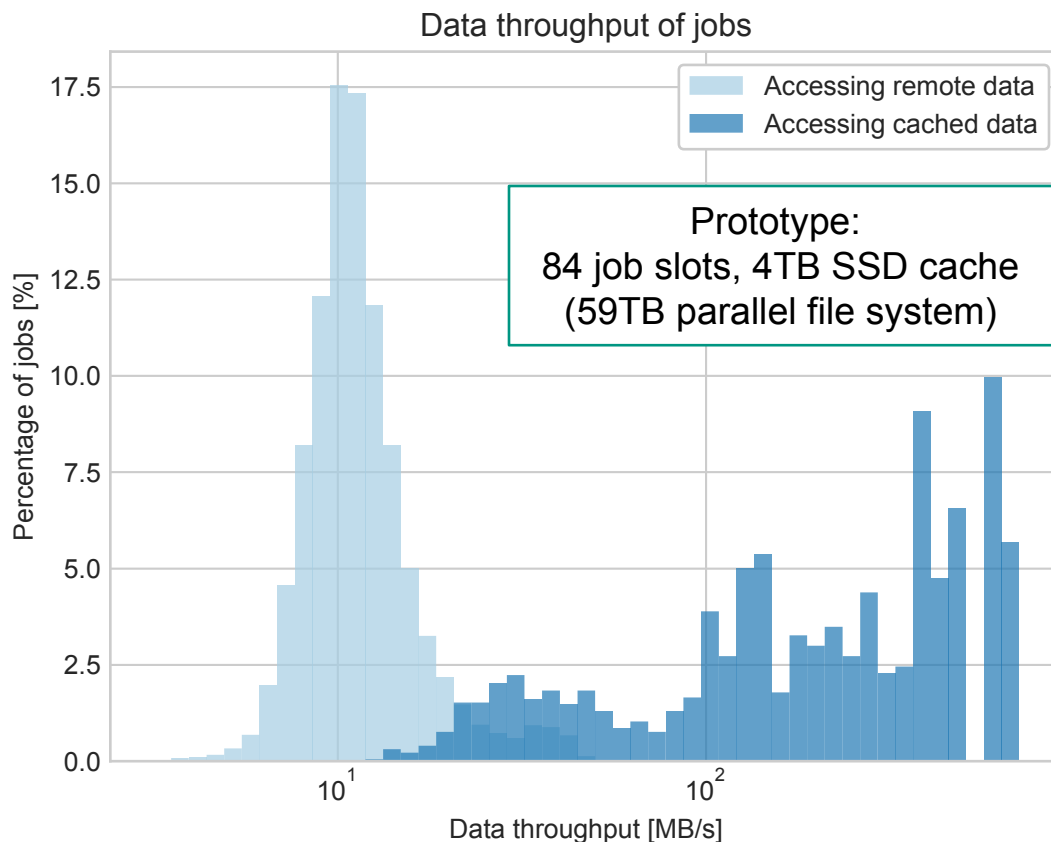
- XRootD and HTCondor take care of hierarchical upscaling
- Job-to-Cache coordination can be performed at all levels with regard to the data location information of the subsystems.



Vision - Not tested right now!

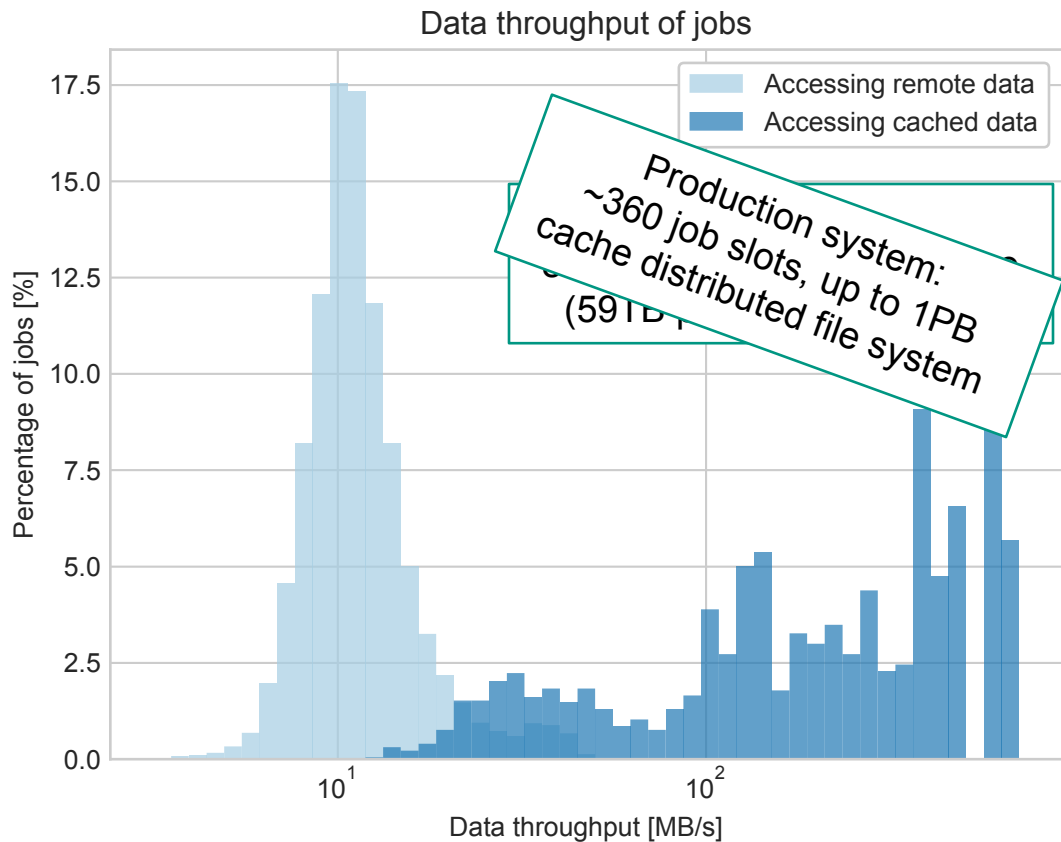
Current status

- Prototype setup is in testing phase
- Deployment of caches on different types of resources
 - At institute resources with high-performant devices
 - At HPC centers and cloud resources



Current status

- Prototype setup is in testing phase
- Deployment of caches on different types of resources
 - At institute resources with high-performant devices
 - At HPC centers and cloud resources
- Production system with advanced coordination logic is scheduled



- ➔ Caches reach maximum read speed
- ➔ Simple coordination logic already improves data throughput

Conclusion

- The amount of data that HEP experiments can collect and process are limited by data throughput
- Efficiency is reduced by bandwidth of data transfers via network
- Solution: **Coordinated Distributed Caching**
 - Reduces load on network using localized caches
 - Reaches data locality by scheduling the job to most suitable cache
 - Data placement via job scheduling
- We developed **NaviX Coordination Service**
 - Extends commonly used HTCondor and XRootD setup
 - Integrates cache location information into job scheduling
 - Enables monitoring and fine-tuning of data accesses