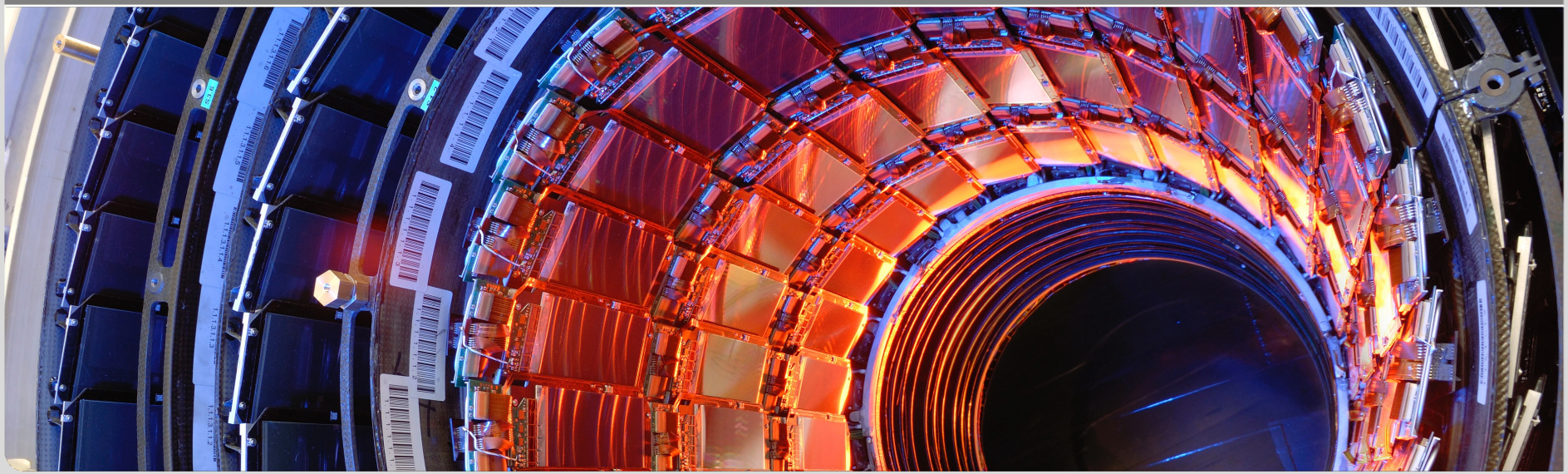
Karlsruher Institut für Technologie

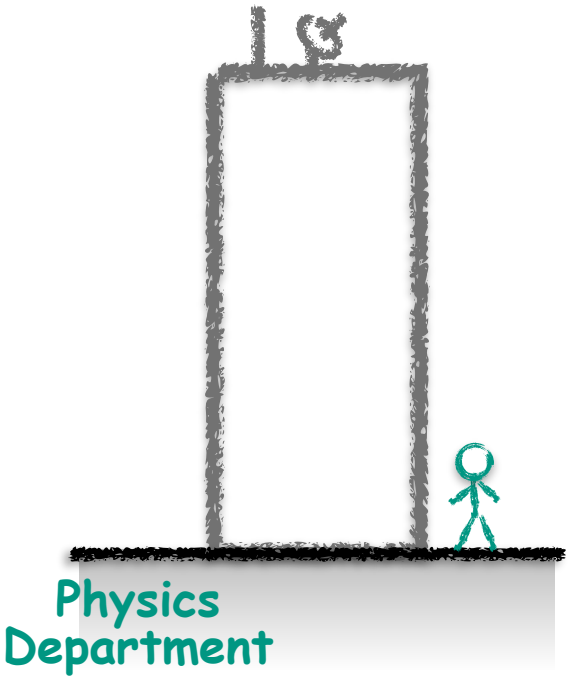
High Performance Data Analysis via Coordinated Caches

Max Fischer, Christian Metzlaff, Manuel Giffels, Günter Quast, et al.
CHEP 2015

Institute for Experimental Nuclear Physics - Steinbuch Centre for Computing

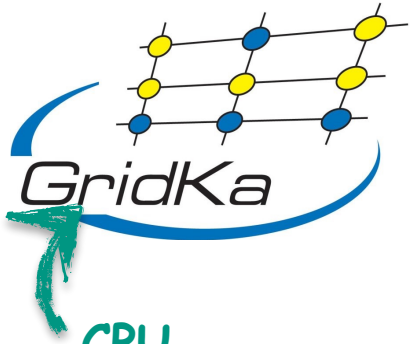


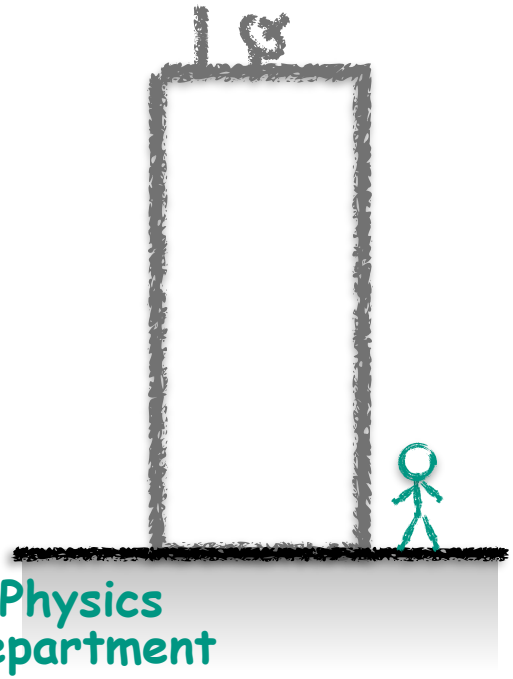
KIT Computing Resources



KIT Computing Resources

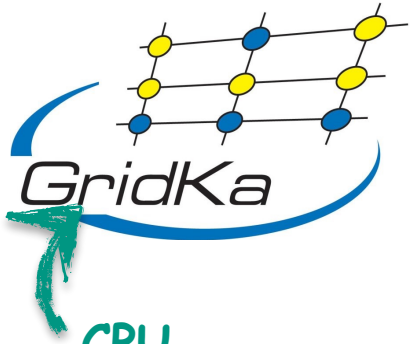
Tier 1
Track 6

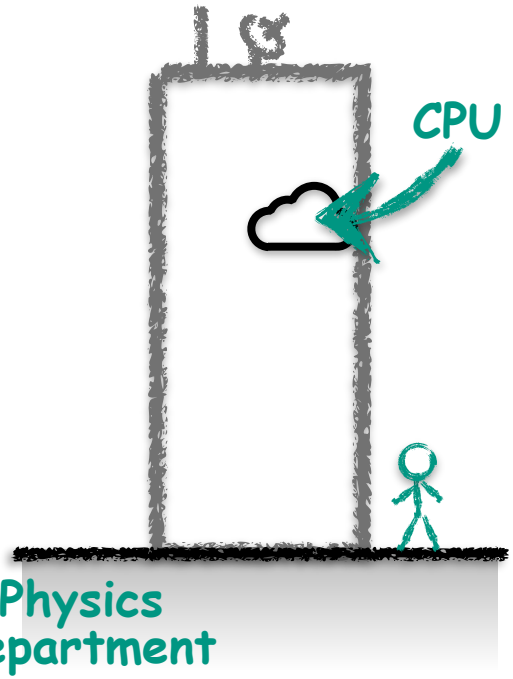




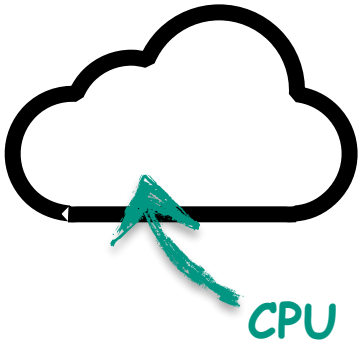
KIT Computing Resources

Tier 1
Track 6



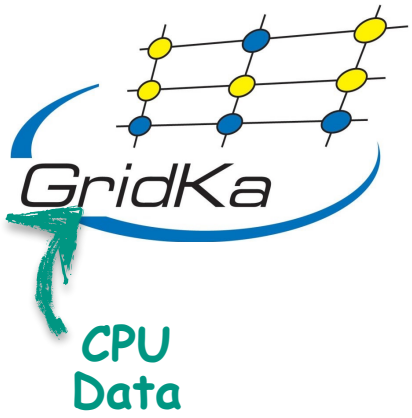


Opportunistic HPC
Track 7

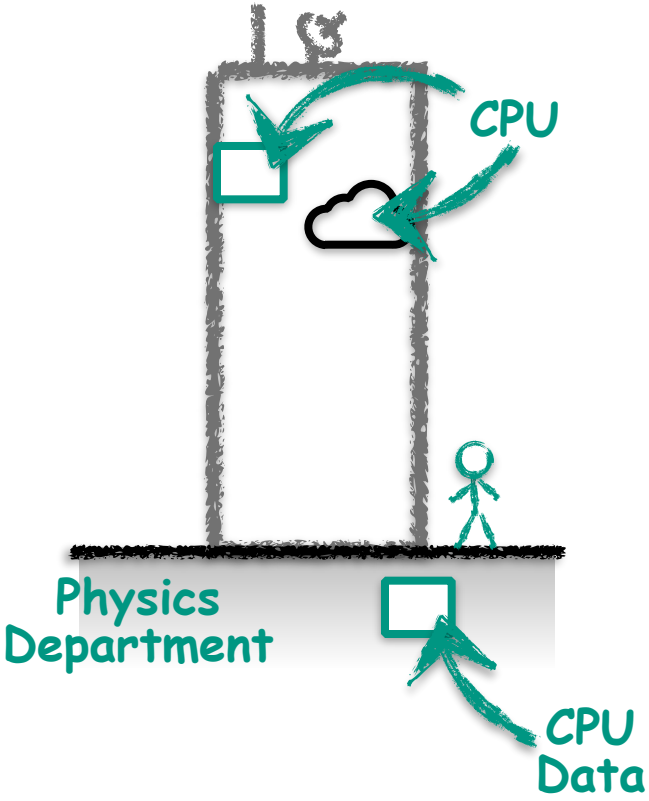


KIT Computing Resources

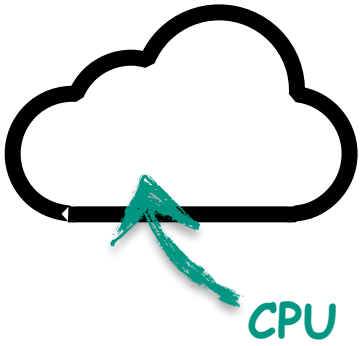
Tier 1
Track 6



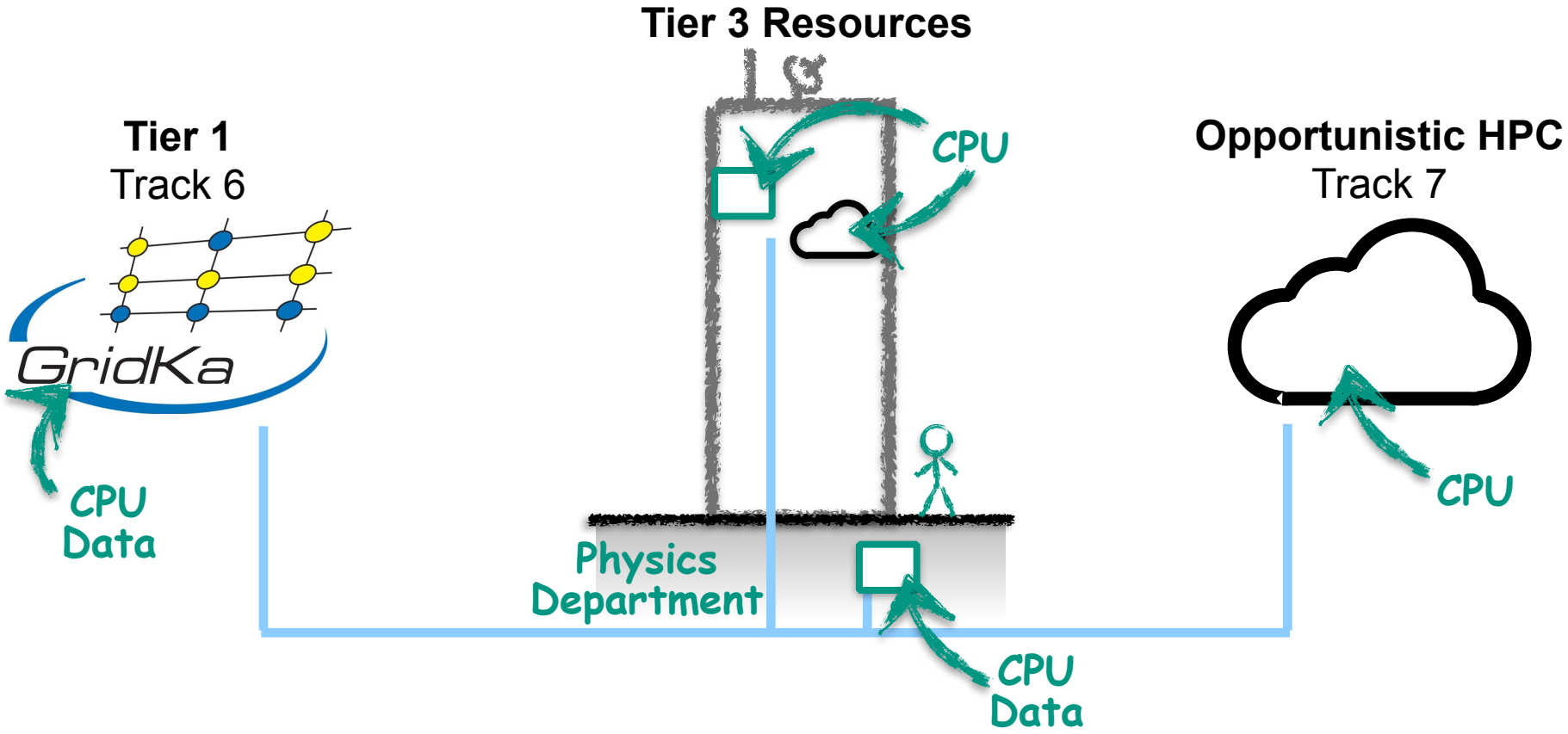
Tier 3 Resources



Opportunistic HPC
Track 7



KIT Computing Resources



End User Analysis Characteristics

- Range of applications targeting HPC to HTC
  - Toy Monte Carlo simulation
  - Data based physics analysis

End User Analysis Characteristics

- Range of applications targeting HPC to HTC
  - Toy Monte Carlo simulation
  - Data based physics analysis

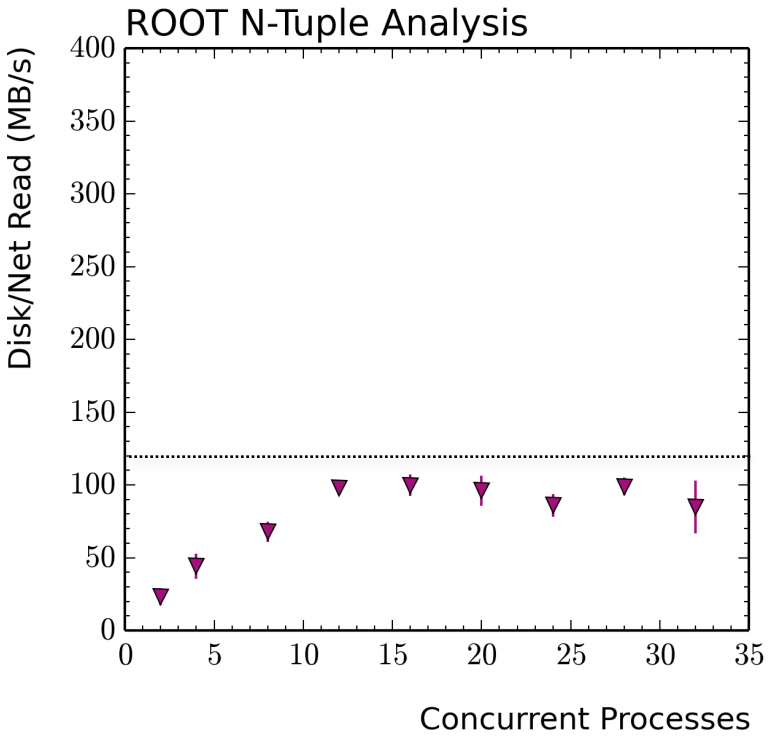
- Data analysis approach
  - Compact skimmed data sets
  - Multitude of variates/parameters
  - Iterative development and tuning

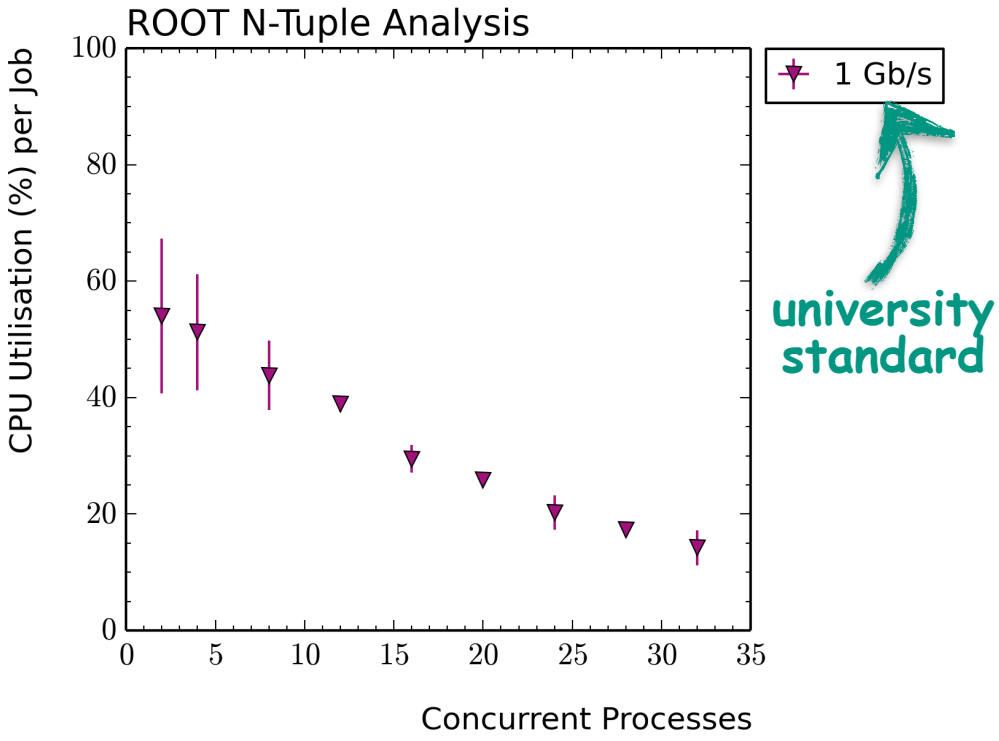
End User Analysis Characteristics

- Range of applications targeting HPC to HTC
  - Toy Monte Carlo simulation
  - Data based physics analysis
- Data analysis approach
  - Compact skimmed data sets
  - Multitude of variates/parameters
  - Iterative development and tuning
- Data analysis workflows
  - Run on Tier 3 batch systems/storage
  - ~1-4 TB input data for LHC run1
  - Split on O(100) jobs

I/O Performance Evaluation

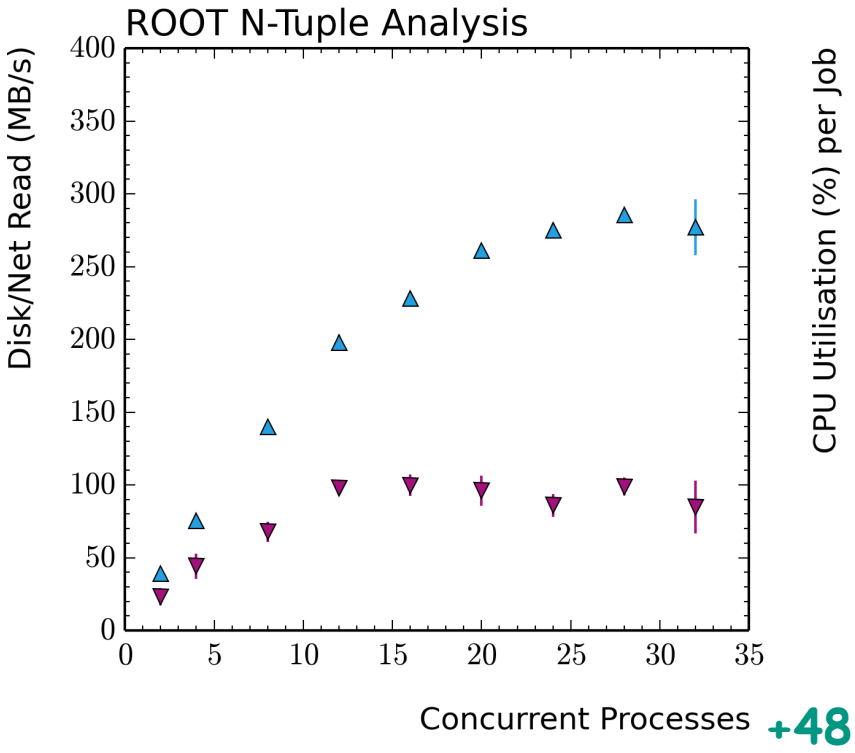
- CMS user analysis (ROOT n-tuple)

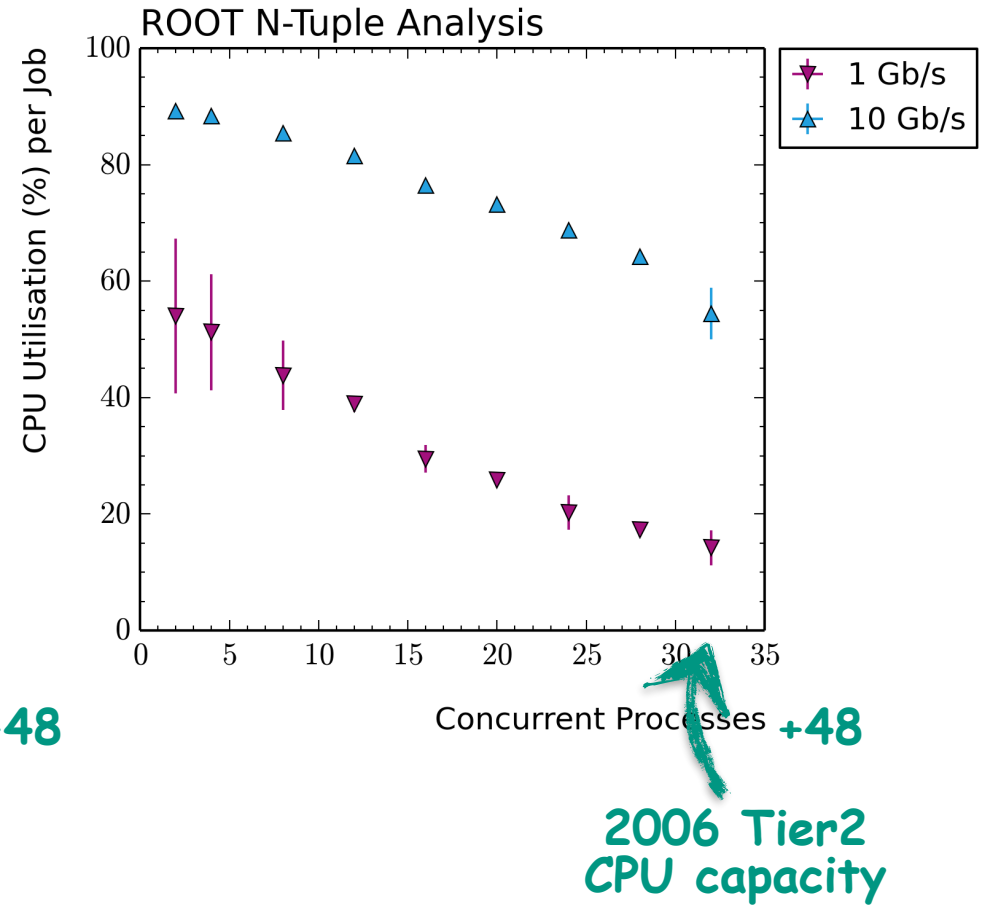




I/O Performance Evaluation

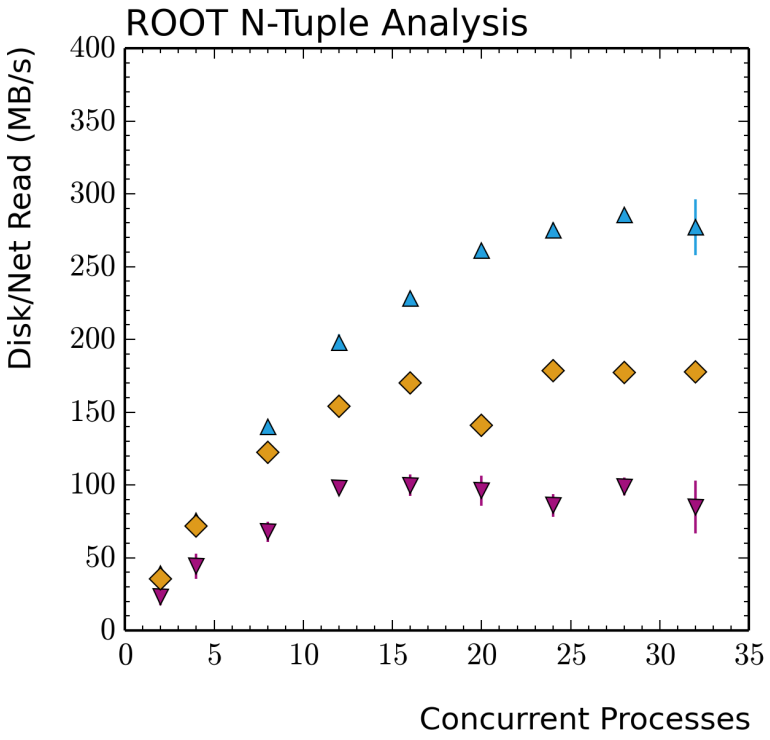
- CMS user analysis (ROOT n-tuple)
- Additional 48 concurrent reads from other workers for 10 Gb/s test

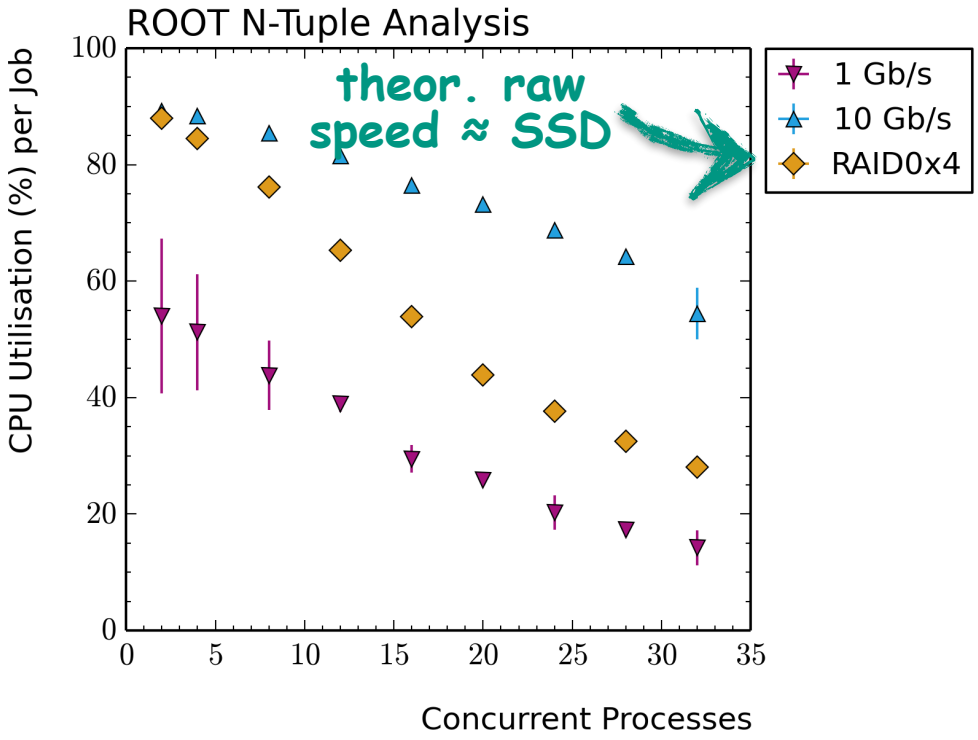




I/O Performance Evaluation

- CMS user analysis (ROOT n-tuple)
- Additional 48 concurrent reads from other workers for 10 Gb/s test

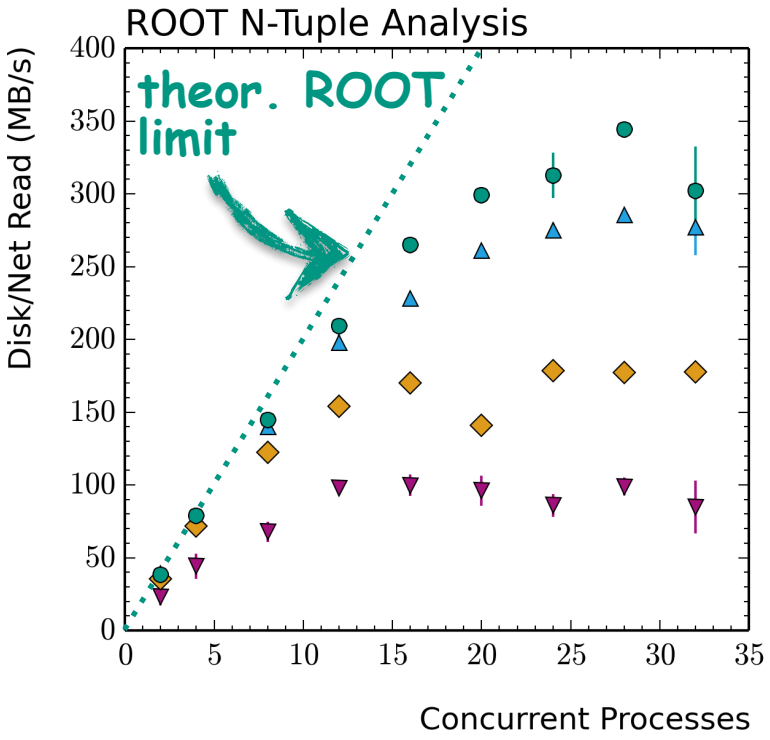


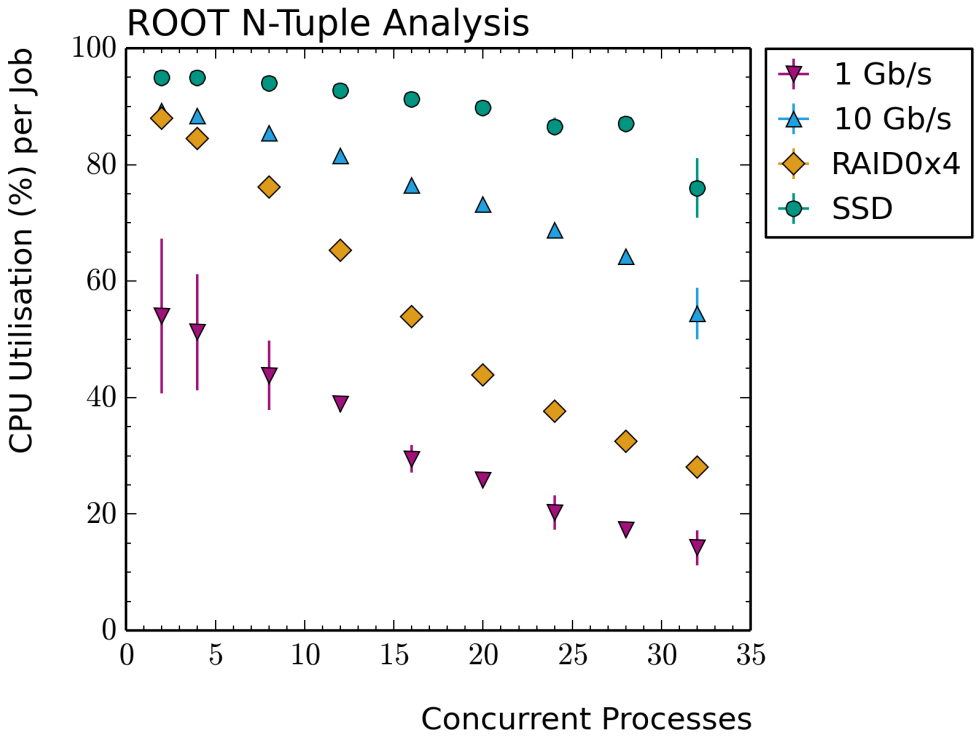


- HDDs limited on concurrent accesses

I/O Performance Evaluation

- CMS user analysis (ROOT n-tuple)
- Additional 48 concurrent reads from other workers for 10 Gb/s test





- HDDs limited on concurrent accesses
- SSDs exploit full system capacities

Feasibility of SSDs for Analysis

- SSDs still expensive, limited capacity
  - Inadequate as primary storage
  - Too valuable for replications, inactive data

Feasibility of SSDs for Analysis

- SSDs still expensive, limited capacity
  - Inadequate as primary storage
  - Too valuable for replications, inactive data
- Volatile, automated caches

Feasibility of SSDs for Analysis

- SSDs still expensive, limited capacity
  - Inadequate as primary storage
  - Too valuable for replications, inactive data
- Volatile, automated caches

- Distributed analyses, wide range of features
  - Caches must act as single entity
  - Cache strategy sensitive to workflows

Feasibility of SSDs for Analysis

- SSDs still expensive, limited capacity
  - Inadequate as primary storage
  - Too valuable for replications, inactive data
- ▶ Volatile, automated caches

- Distributed analyses, wide range of features
  - Caches must act as single entity
  - Cache strategy sensitive to workflows
- ▶ Coordinated pool of caches

Feasibility of SSDs for Analysis

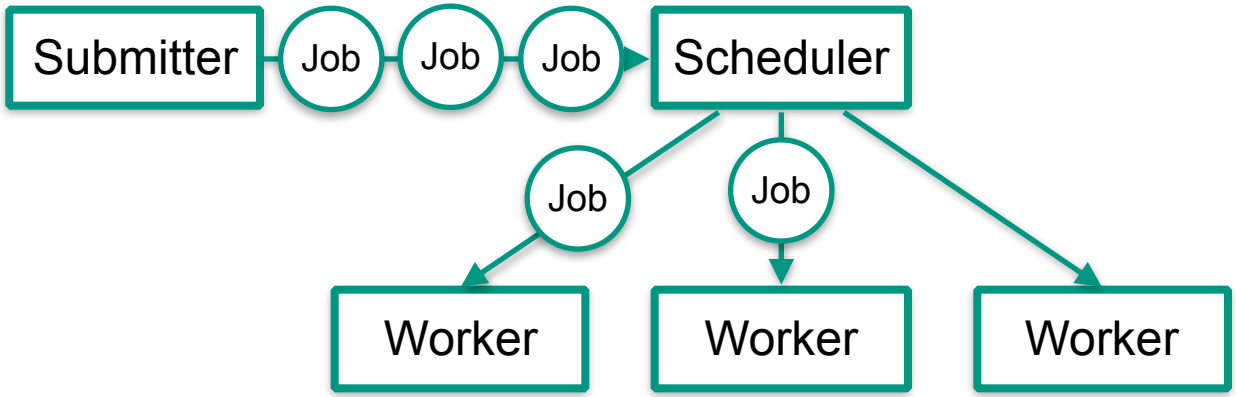
- SSDs still expensive, limited capacity
  - Inadequate as primary storage
  - Too valuable for replications, inactive data
- ▶ Volatile, automated caches

- Distributed analyses, wide range of features
  - Caches must act as single entity
  - Cache strategy sensitive to workflows
- ▶ Coordinated pool of caches

- Constraints from existing workflows
  - Classic batch systems with POSIX storage
  - Dataset splitting performed by job management tools

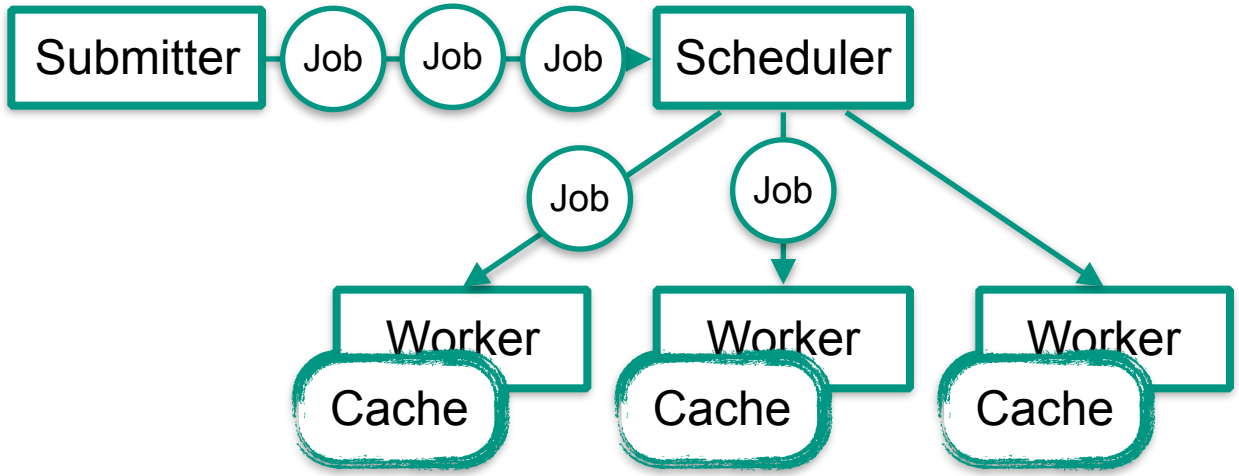
HTDA Batch System Extension

High Throughput Data Analysis



HTDA Batch System Extension

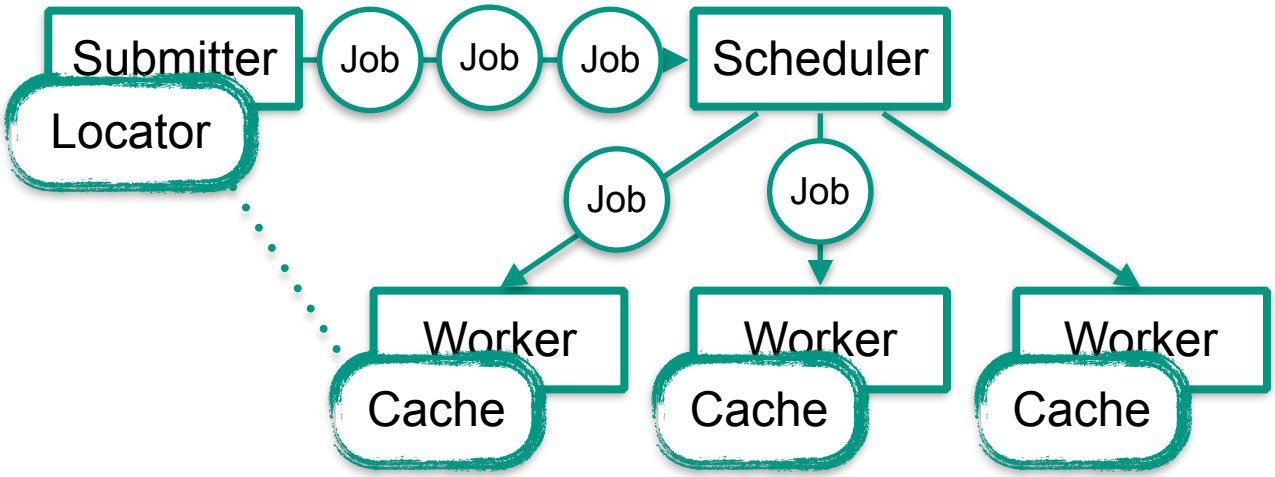
High Throughput Data Analysis



- Caches maintain data copies on worker nodes

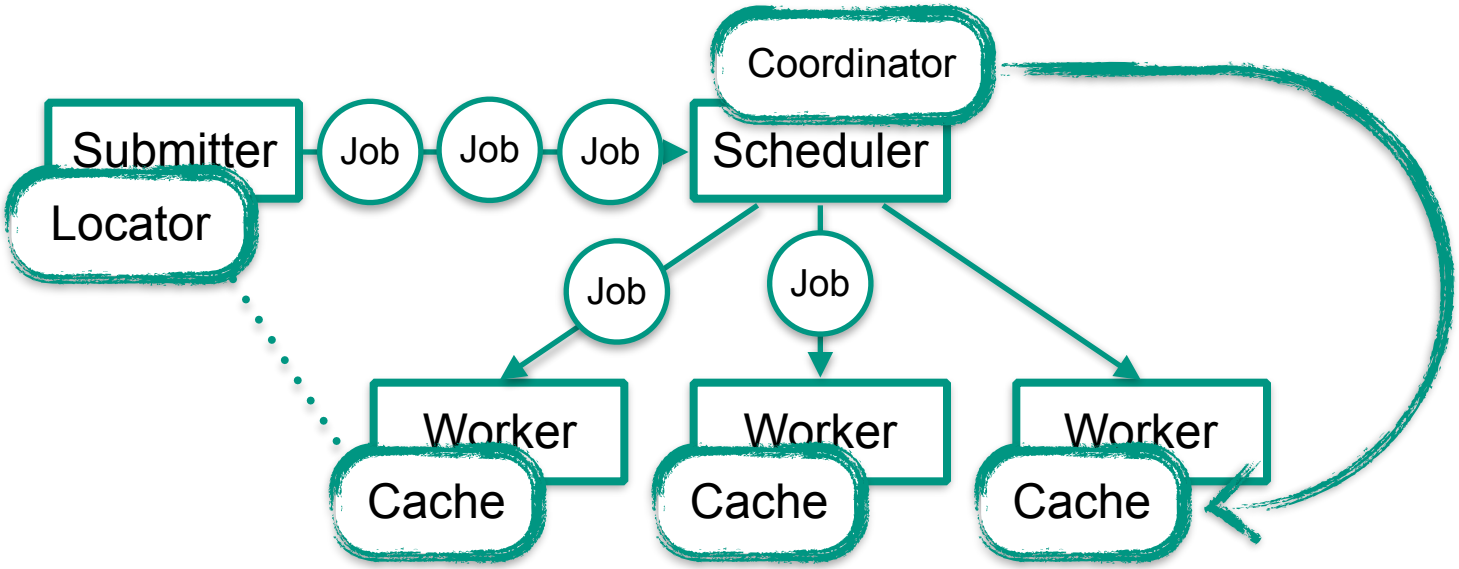
HTDA Batch System Extension

High Throughput Data Analysis



- Caches maintain data copies on worker nodes
- Locator provides locality information for jobs

HTDA Batch System Extension



- Caches maintain data copies on worker nodes
- Locator provides locality information for jobs
- Coordinator schedules files for caching on nodes

Batch System Integration

- Hooks modify jobs on HTCondor submission nodes

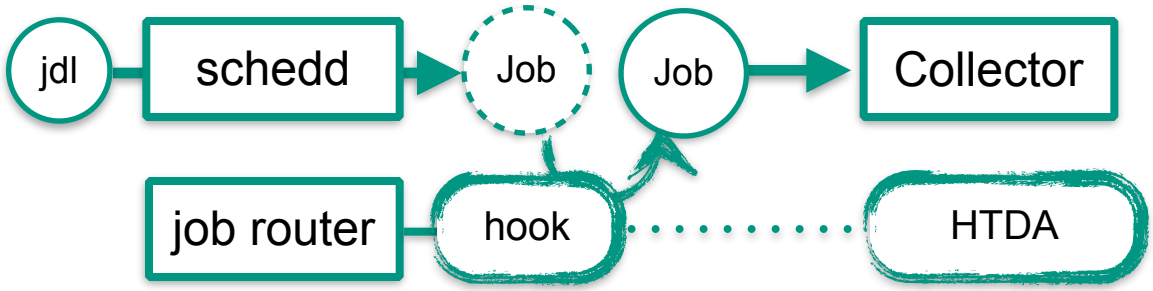
```
# HTC+HPDA analysis.jdl

Executable  = artus_job_wrapper
Output      = ...
...
Arguments   = 1
Input_Files = job1_files.txt
Queue

Arguments   = 2
Input_Files = job2_files.txt
Queue

...
```

Batch System Integration

- Hooks modify jobs on HTCondor submission nodes
- Plugin for HTC Job Router collects job features, adds scheduling hints

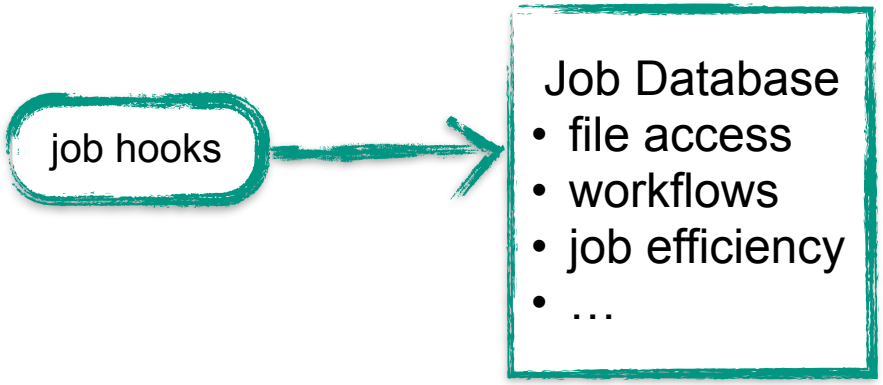
```
# HTC+HPDA analysis.jdl

Executable  = artus_job_wrapper
Output      = ...
...
Arguments   = 1
Input_Files = job1_files.txt
Queue

Arguments   = 2
Input_Files = job2_files.txt
Queue

...
```

```
$ condor_q -long
...
Rank = ( 0.0 ) + HTDA_RANK
HTDA_RANK = 0 + (
( machine == "ekpsg01" ) * 12
) + (
( machine == "ekpsg03" ) * 13 )
...
```



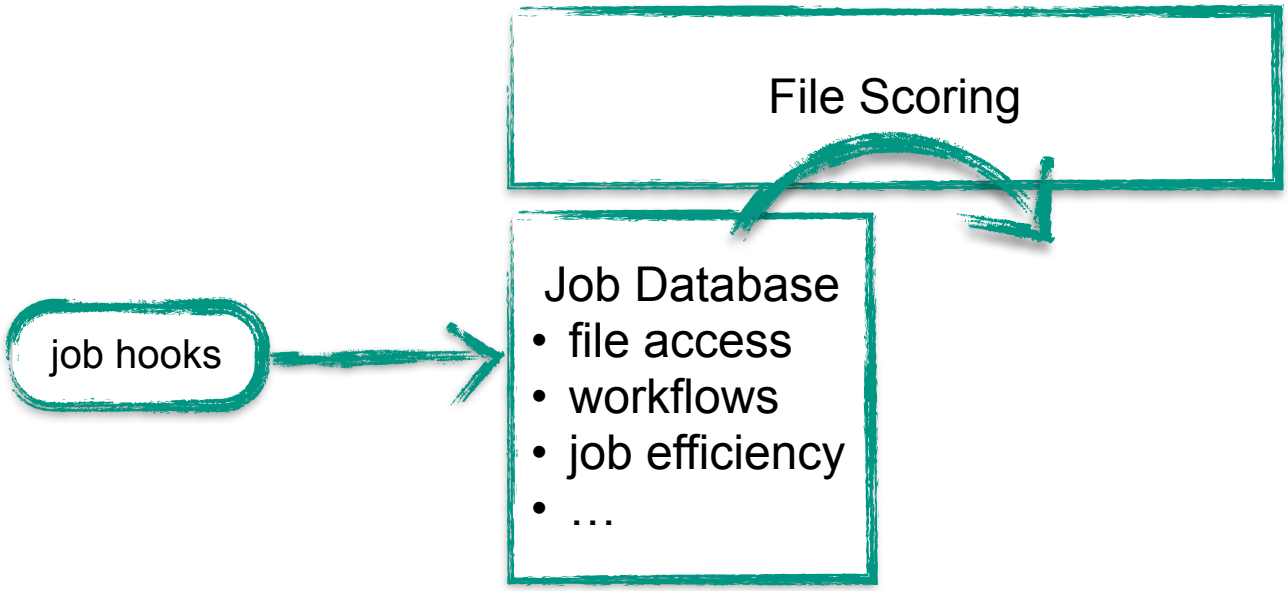
Coordinator

- Database tracks history of job features



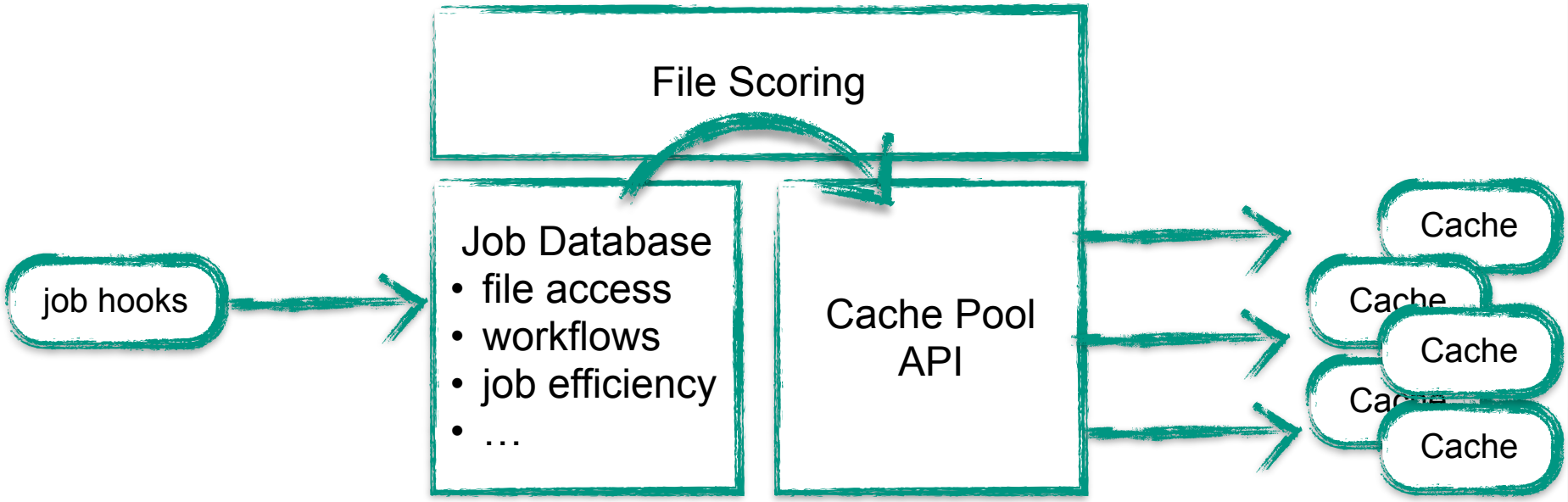
Coordinator

- Database tracks history of job features
- File importance calculated for all files
  - Currently expanded LRU with historical information
  - Planning for predictive caching using dataset/workflow information



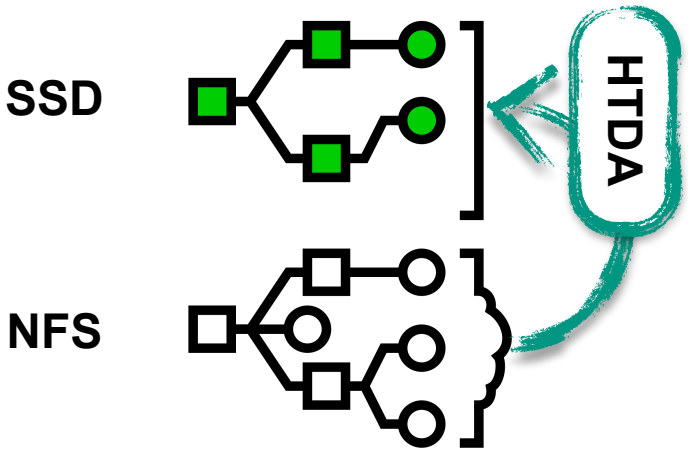
Coordinator

- Database tracks history of job features
- File importance calculated for all files
  - Currently expanded LRU with historical information
  - Planning for predictive caching using dataset/workflow information



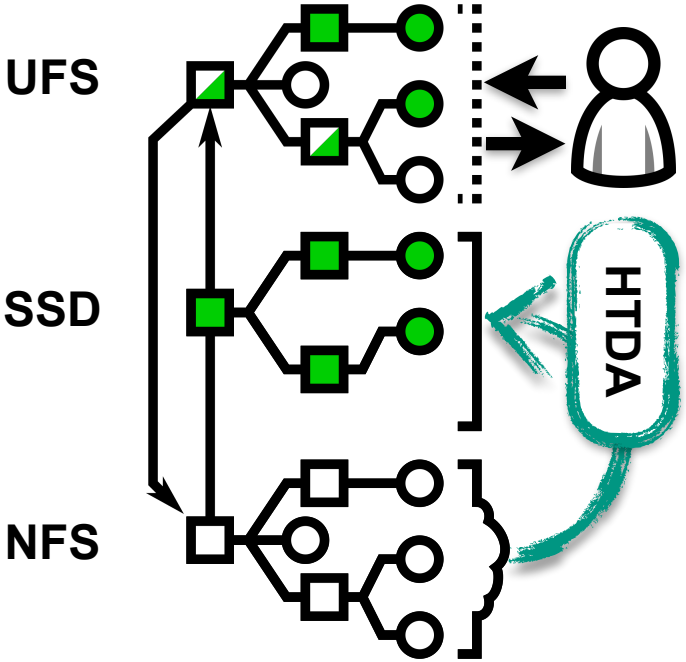
Cache

- Cache node stages/unstages files according to coordinator score



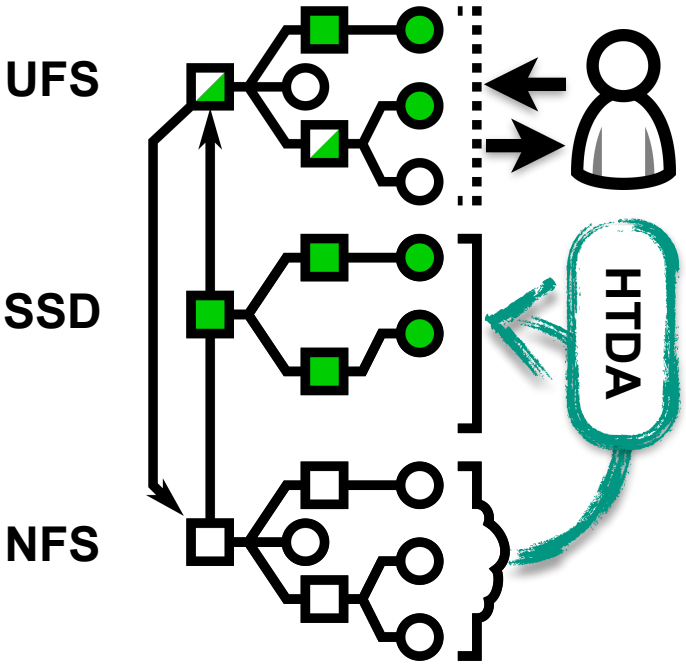
Cache

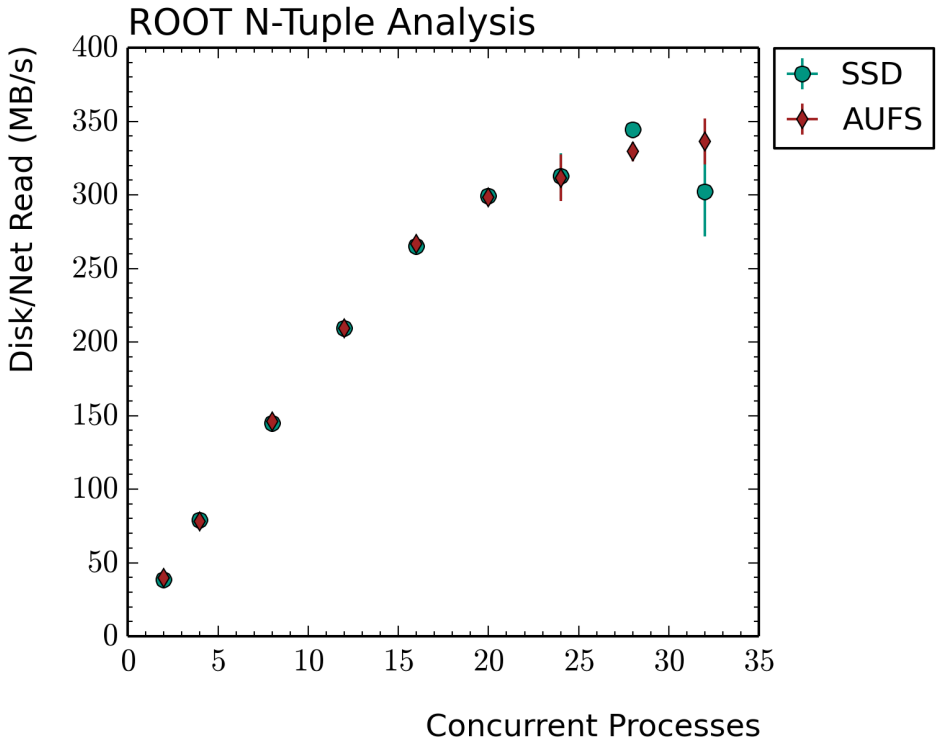
- Cache node stages/unstages files according to coordinator score
- Union File System provides transparent cache access for users



Cache

- Cache node stages/unstages files according to coordinator score
- Union File System provides transparent cache access for users

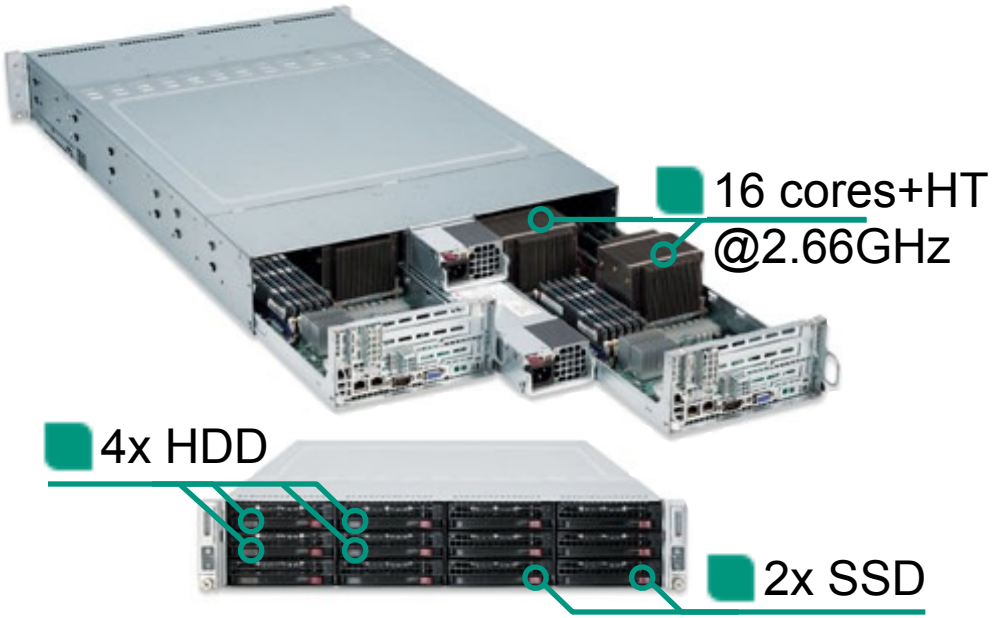


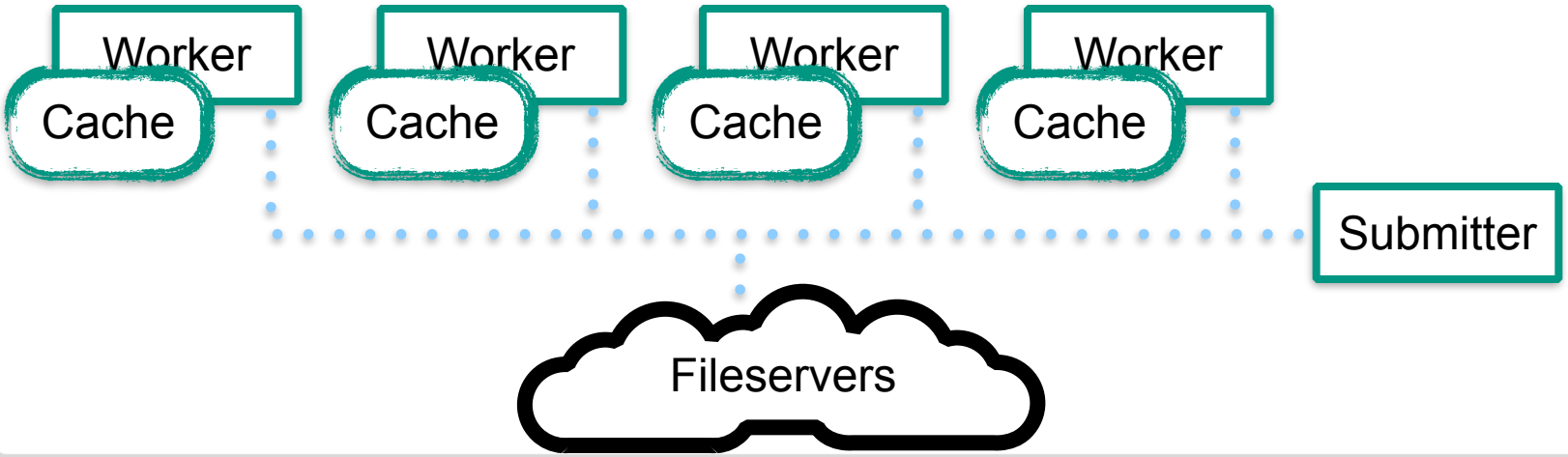


- Lightweight cache access ensures optimal performance

Prototype Setup

- HTCondor test cluster
  - 4 worker/cache nodes
  - 4 TB SSD cache
  - 1 submit/service node
  - 6 fileservers





Prototype Experience

- Prototype operation
  - Test operation for 8 weeks
  - O(300k) successful jobs

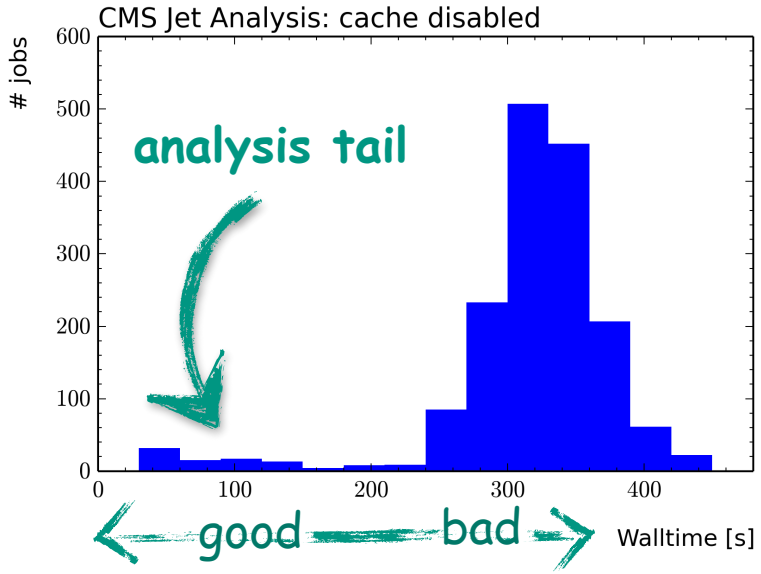
Prototype Experience

- Prototype operation
  - Test operation for 8 weeks
  - O(300k) successful jobs

- Reference end user workflow
  - CMS calibration analysis
  - ROOT n-tuple framework
  - 400 GB LHC run1 input data

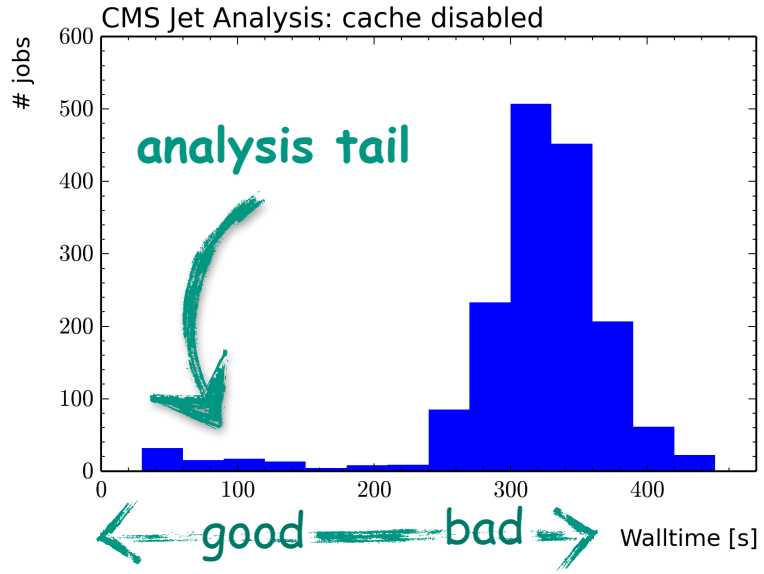
Prototype Experience

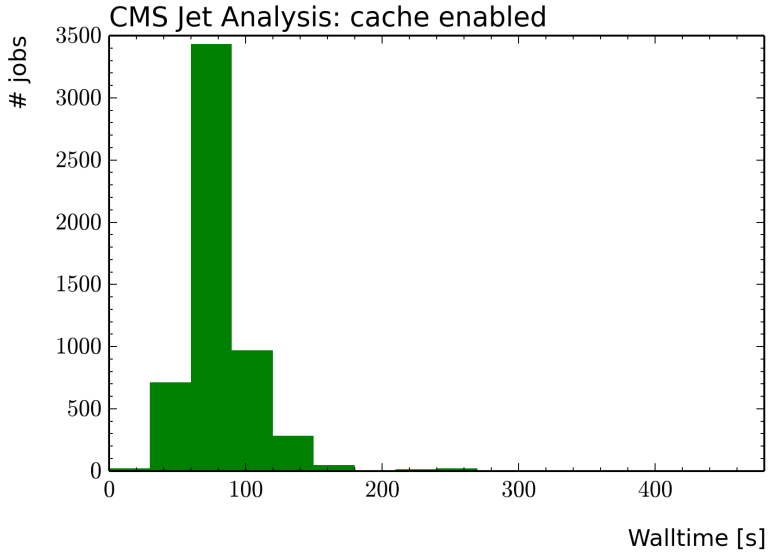
- Prototype operation
  - Test operation for 8 weeks
  - O(300k) successful jobs
- Reference end user workflow
  - CMS calibration analysis
  - ROOT n-tuple framework
  - 400 GB LHC run1 input data



Prototype Experience

- Prototype operation
  - Test operation for 8 weeks
  - O(300k) successful jobs
- Reference end user workflow
  - CMS calibration analysis
  - ROOT n-tuple framework
  - 400 GB LHC run1 input data





Summary

- User analysis requires high data throughput
  - Iterative runs on same data
  - Data size ~1-4 TB
  - Limited scaling with shared I/O
- HTDA introduces distributed cache pool
  - Coupled to batch system/workflows
  - Prototype of SSDs in HTCondor pool
  - Lightweight, transparent integration
- Basis for further developments
  - Predictive caching, dataset readahead
  - Access to grid resources and caching via xrootd