Karlsruhe Institute of Technology
Institute of Experimental Particle Physics

Provisioning of data locality for HEP analysis workflows

Christoph Heidecker, Matthias Jochen Schnepf, Max Fischer, Manuel Giffels, Günter Quast

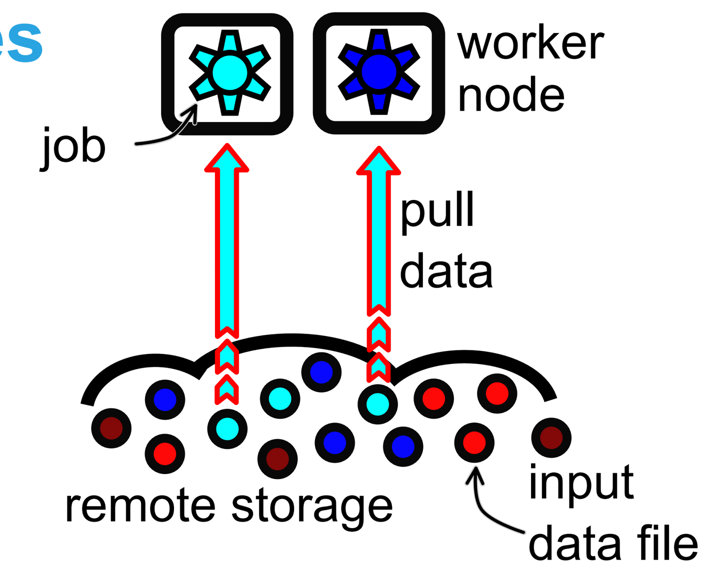
Introduction

Increasing input/output in HEP analyses

- Heavily increasing amount of data
- Fast processing of huge datasets required

Current HEP computing approach

- Dedicated network storage at WLCG Tiers
- Batch farms provide CPUs for processing



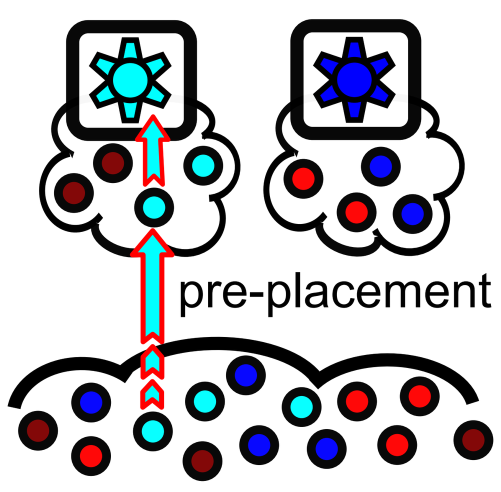
Data transfers limited by network

- Shared transfer capacity
- Low CPU efficiency

Optimization

- Caching of input data
- Transparent integration into batch system

Coordinated Caching



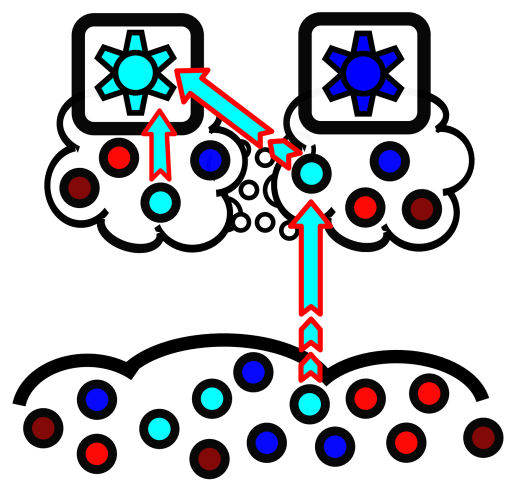
- Shared metadata
- Strong data locality on cache node
- Few, highly performant devices (SSDs)
- Job and data scheduling at node level

Advantages

- Good horizontal and vertical scalability
- Independent of infrastructure
- Customisability of jobs and data locality
- High throughput rates

Distributed Caching

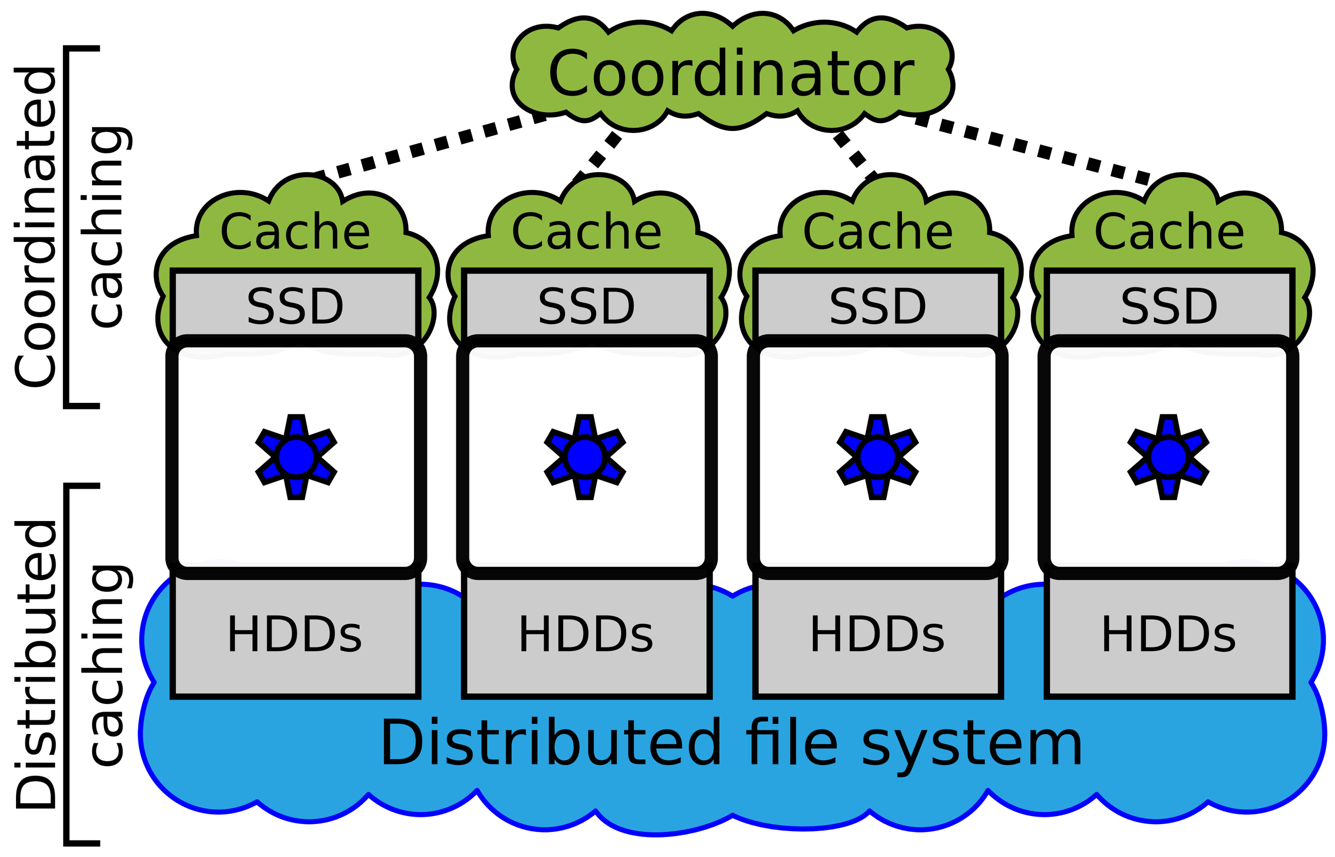
- Shared data
- Weak data locality for all nodes
- Many, low performant devices (HDDs)
- Job and data scheduling at cluster level



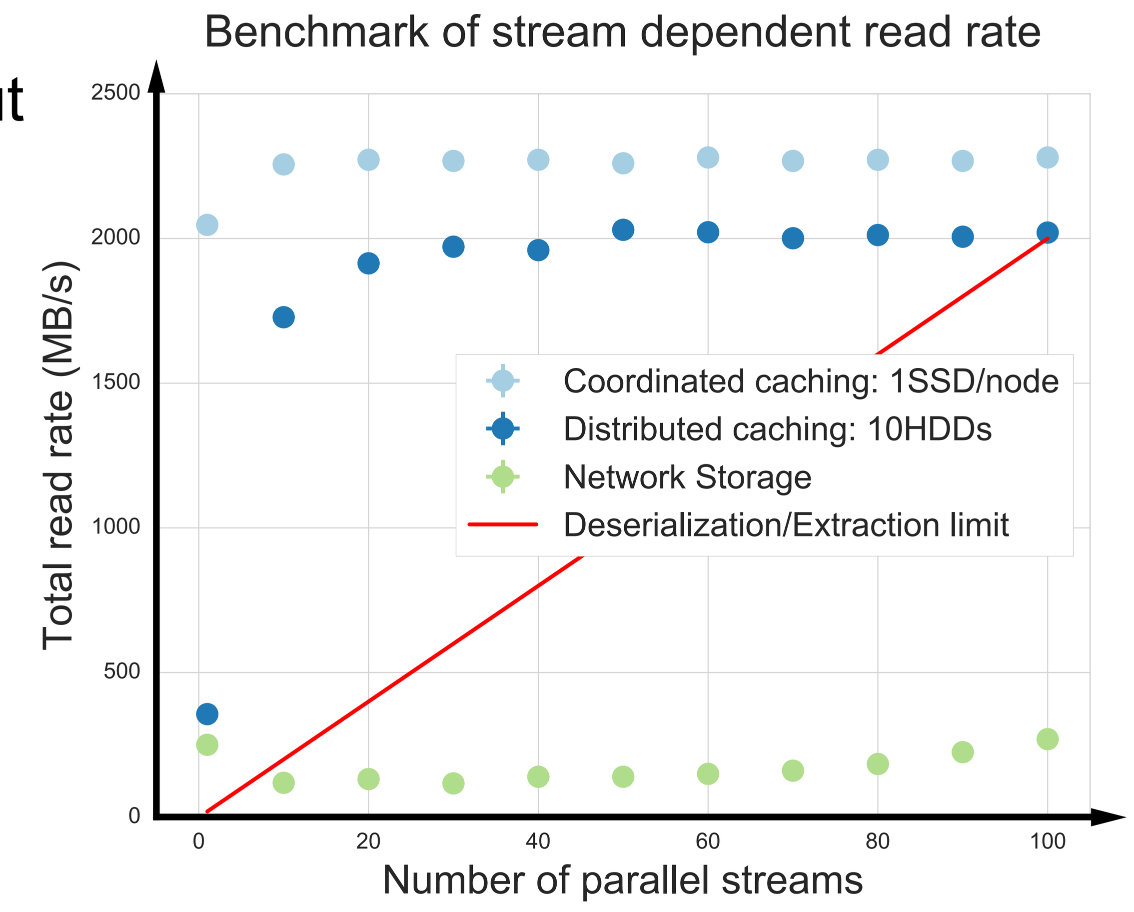
Advantages

- Good vertical scalability
- Exploit static infrastructure, especially interconnection
- Straightforward scheduling of jobs and data locality
- Large cache volume

Prototype and Benchmarks



- Highly improved throughput
- Improved CPU efficiency for I/O dependent jobs
- Coordinated SSD caches easily scalable
- Large distributed HDD cache feasible
- Utilization of network versus local resources efficient adjustable



Conclusion

- Data locality is essential to process large HEP datasets within short cycles
- Limited processing rate at ~20 MB/s/core due to extraction and deserialization of input data files
- Both caching methods achieve this limit and enable fast analyses
- Improvement of caching approach to optimize future analyses workflows