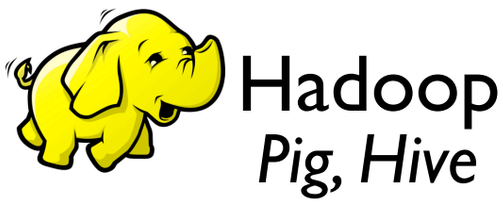
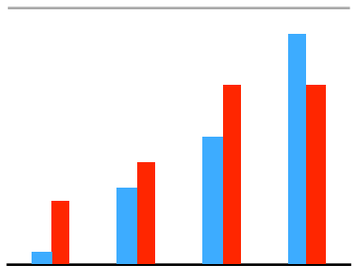
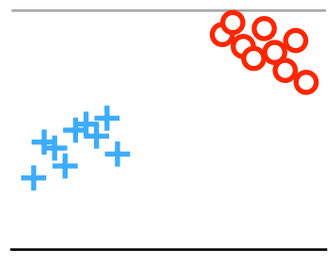
MixApart: Decoupled Analytics for Shared Storage Systems

Madalin Mihailescu, Gokul Soundararajan, Cristiana Amza
University of Toronto and NetApp

NetApp



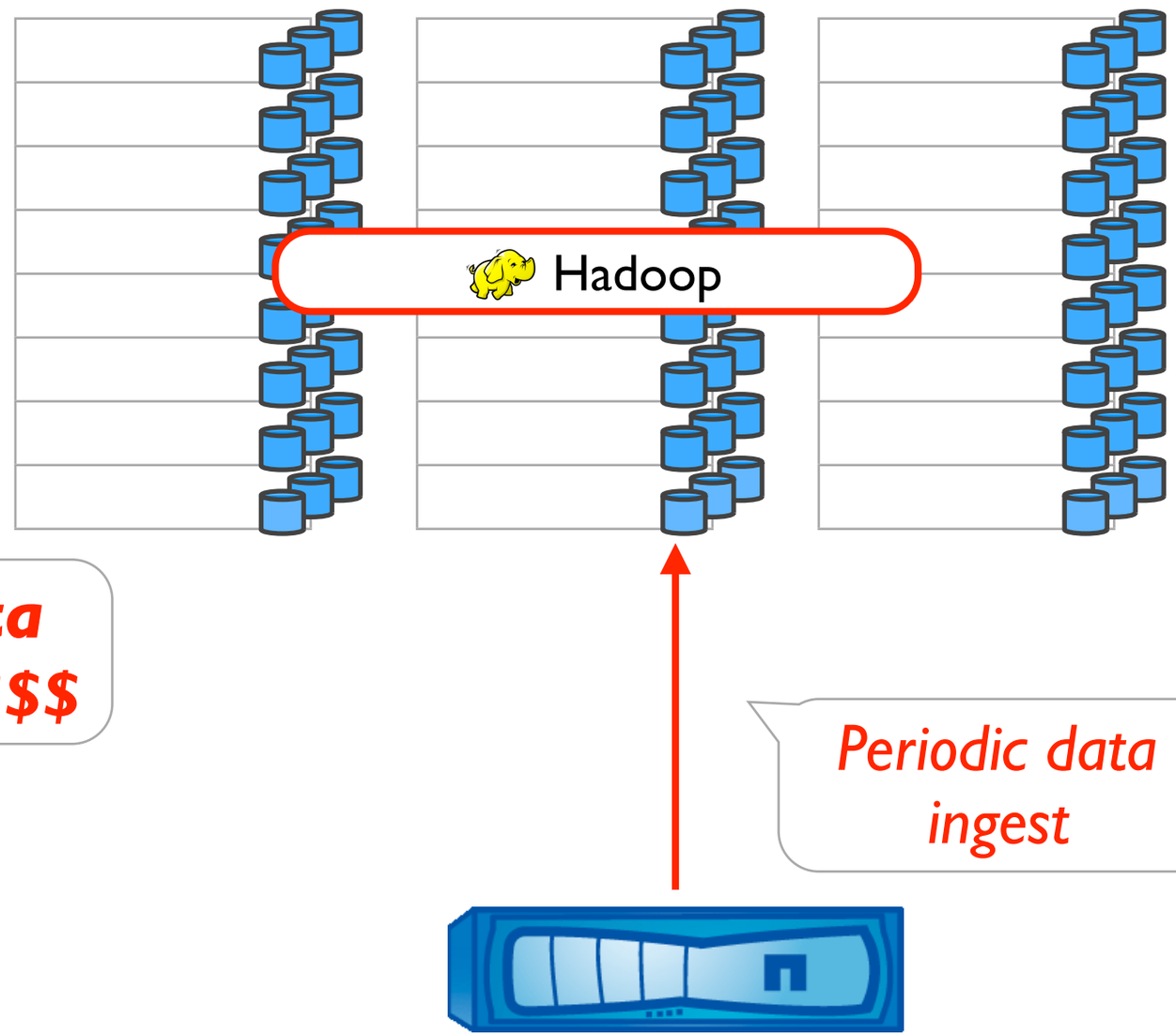




Hadoop + Enterprise storage?!

Shared storage (e.g., NAS)

Hadoop+Enterprise: Two Storage Silos



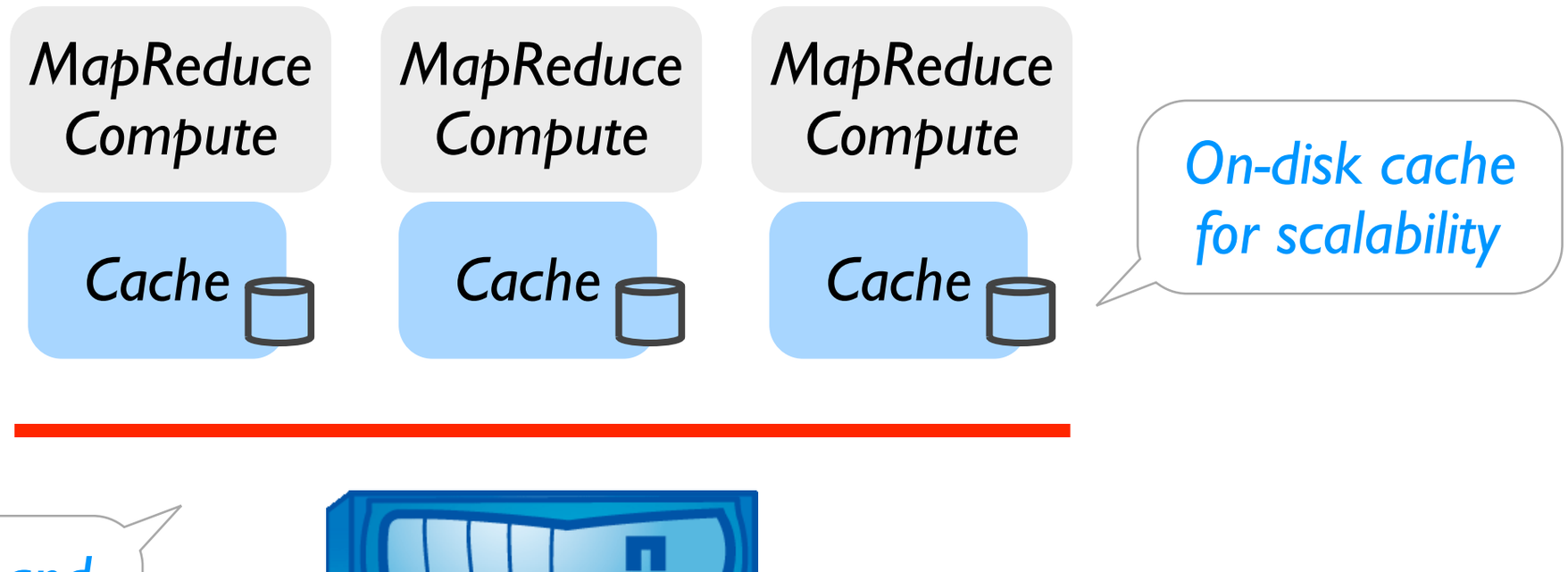
Hardware $$$

Cross-silo data management $$$

Periodic data ingest

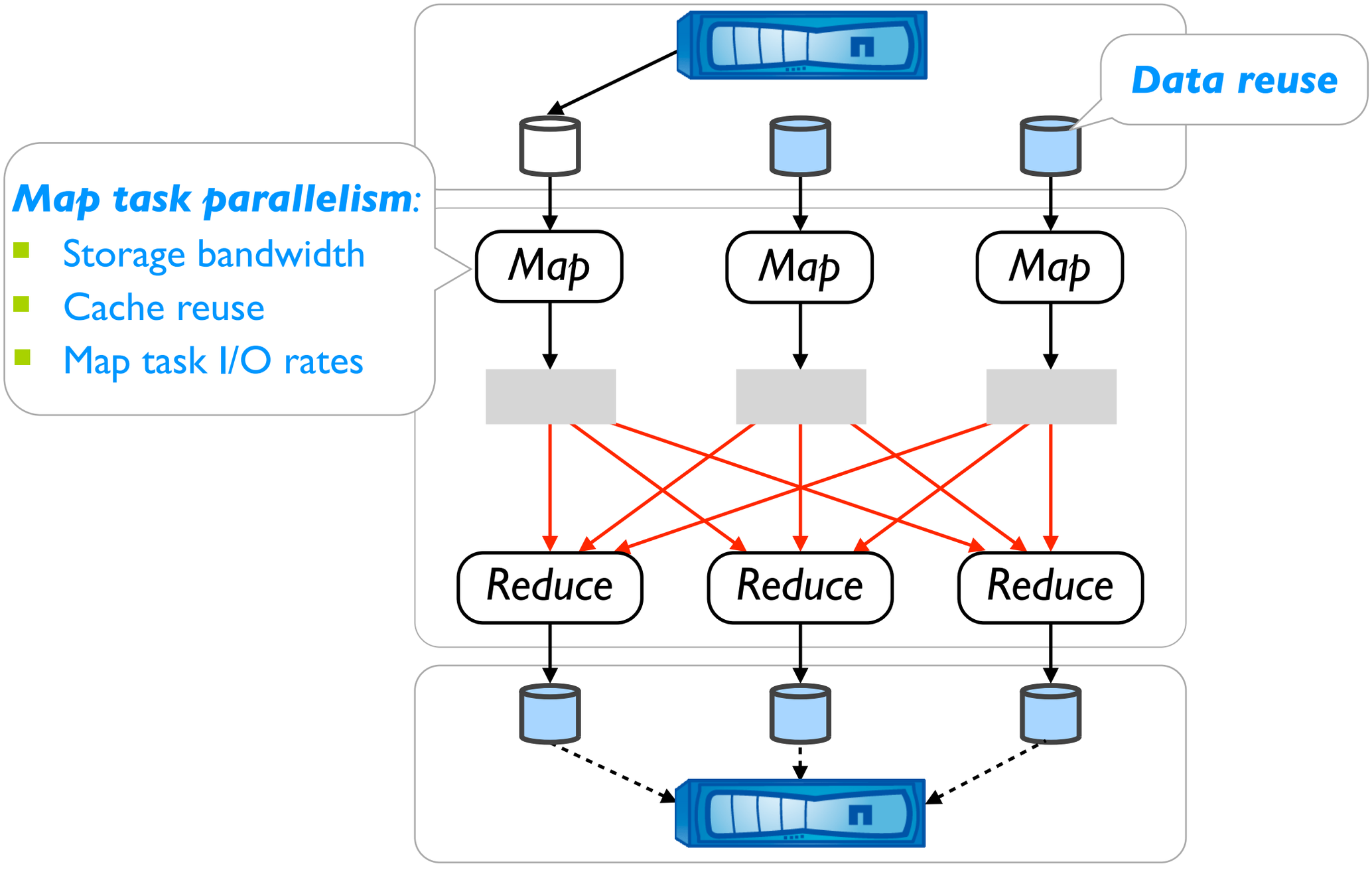
Our Solution: MixApart

- *MapReduce* analytics on *enterprise storage*
  - *Enterprise storage* – **single** reliable data store



Transparent *and* ***on-demand*** *ingest*

Data Flow with MixApart



Workload Analysis

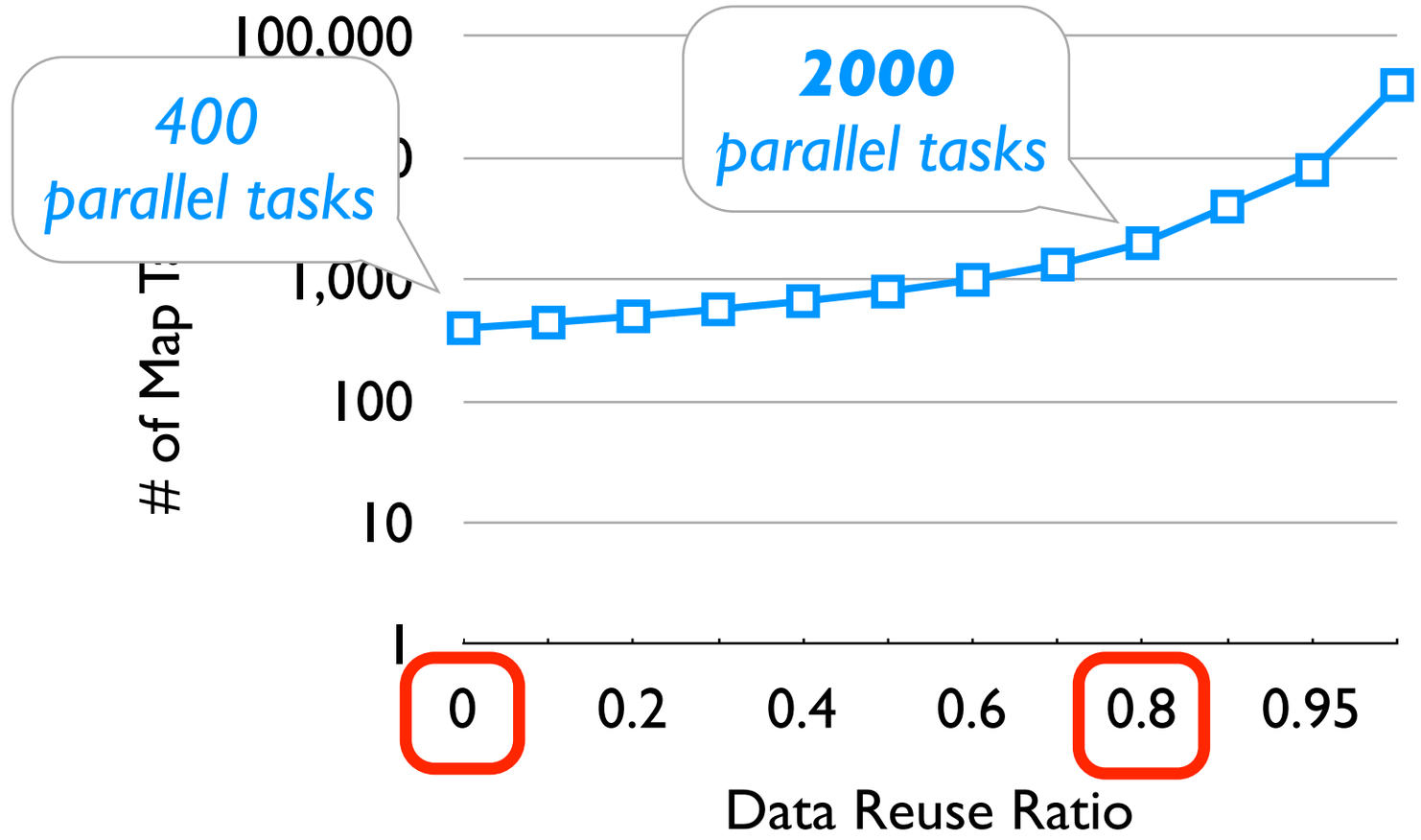
- Extrapolate from recent studies*
  - *Production traces* from Facebook, Bing, Yahoo

- *Insights*
  - High *data reuse* across jobs e.g., **~60%**
  - Low IO to CPU ratio in input phases e.g., **~25Mbps**
  - Predictable IO demands

* Ananthanarayanan et al. NSDI ’12, Chen et al. VLDB ’12

Compute Scale Estimates

Map Task I/O Rate **25 Mbps**



Shared storage bandwidth **10 Gbps**

MixApart Design

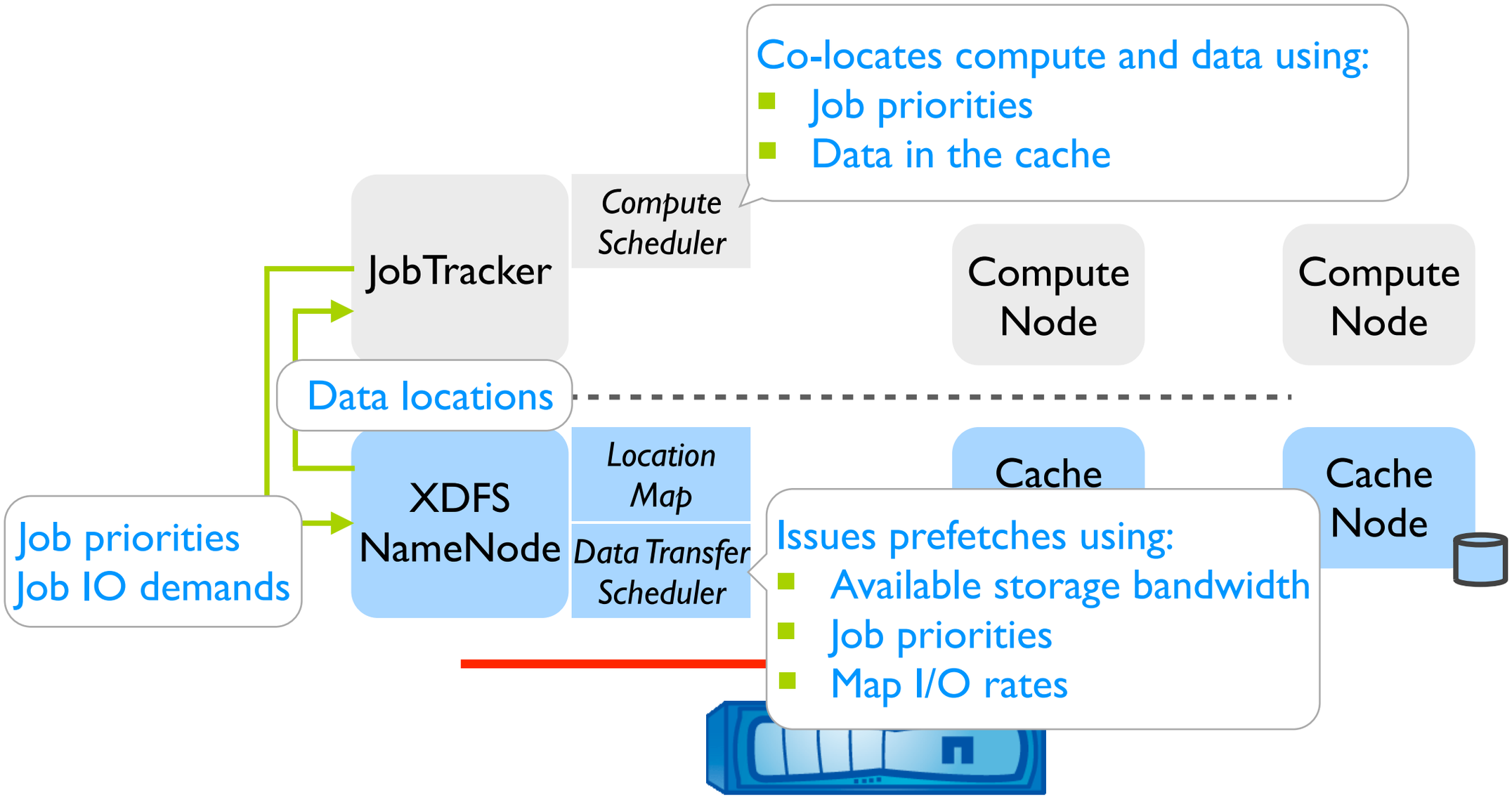
- *Storage back-end bandwidth* management
  - Saturate bandwidth with Map I/O streams without impacting job performance
- *Cache* management
  - Ensure high cached data reuse
- *Compute* management
  - Assign Map tasks to nodes with cached data

MapReduce Optimization

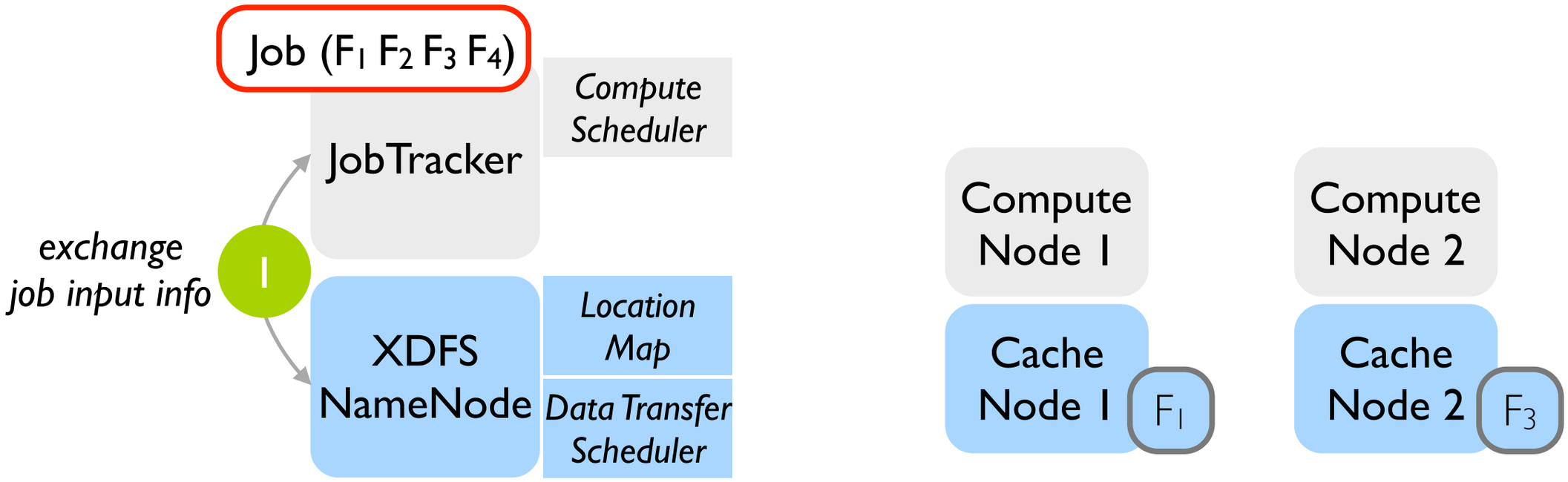
- *Predictable* job I/O demands at submission
  - User-specified job *input* data path
  - Derived Map task *I/O rates*

➡ Just-in-time parallel data prefetch within & across jobs

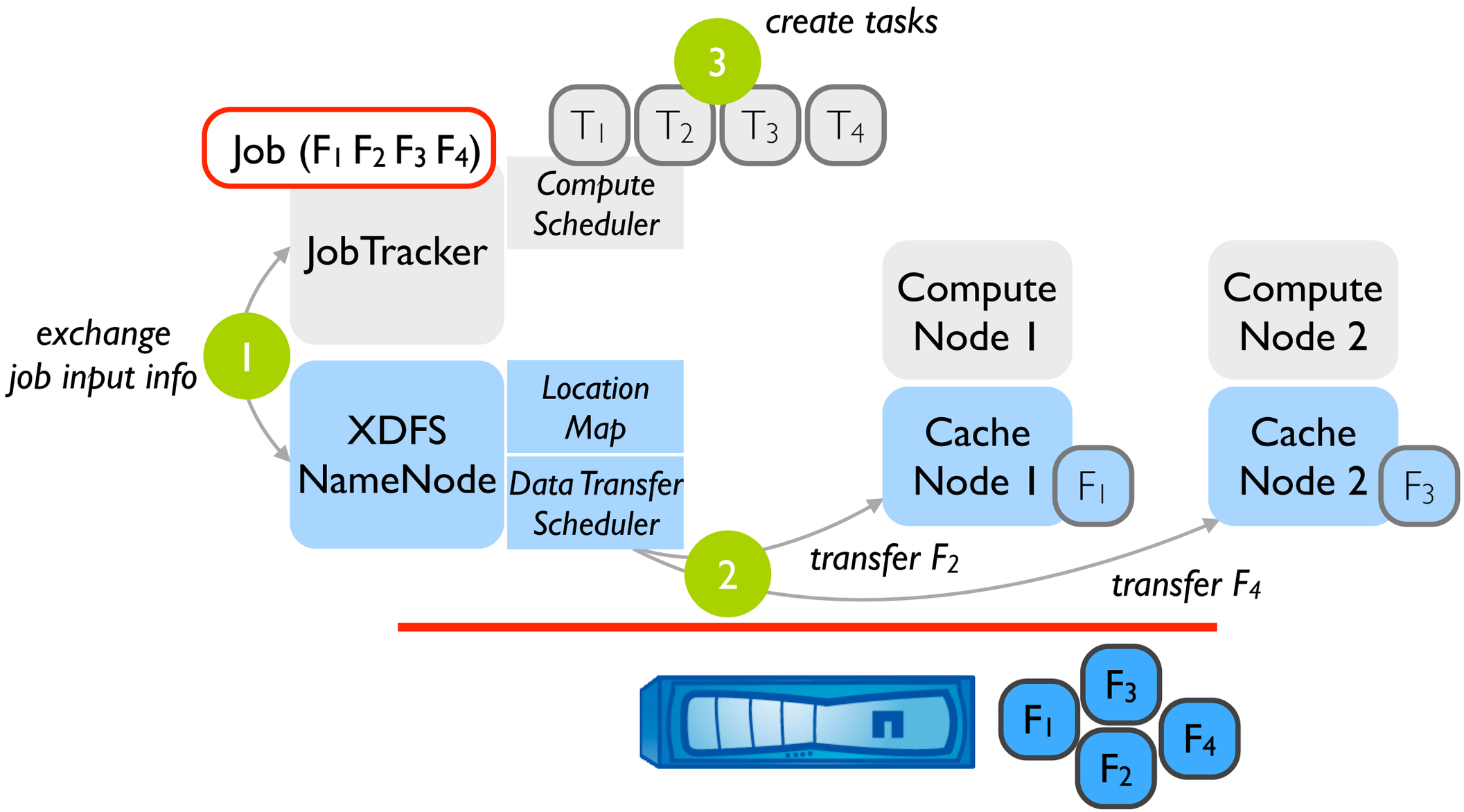
MixApart Architecture



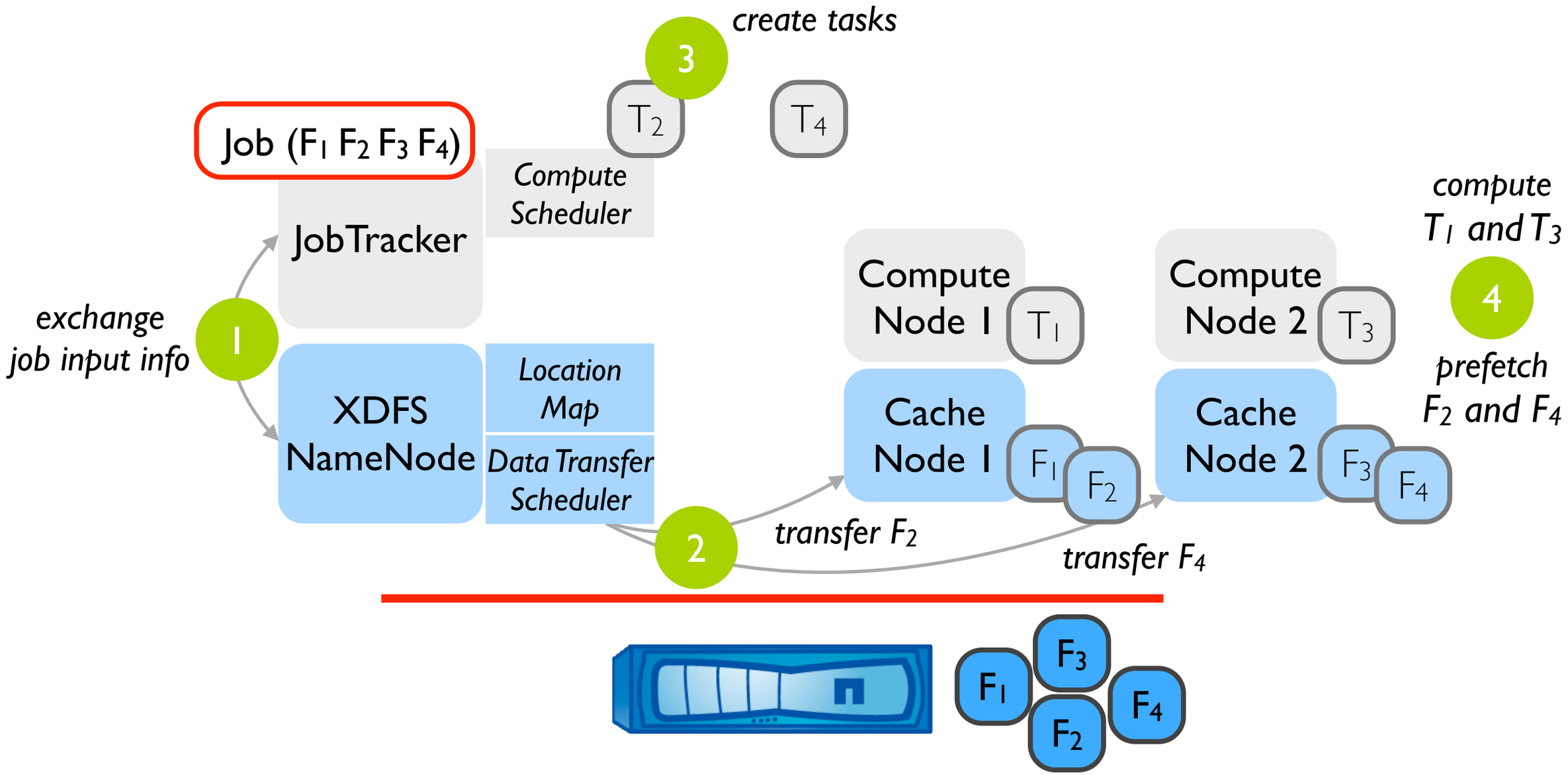
MixApart in Action



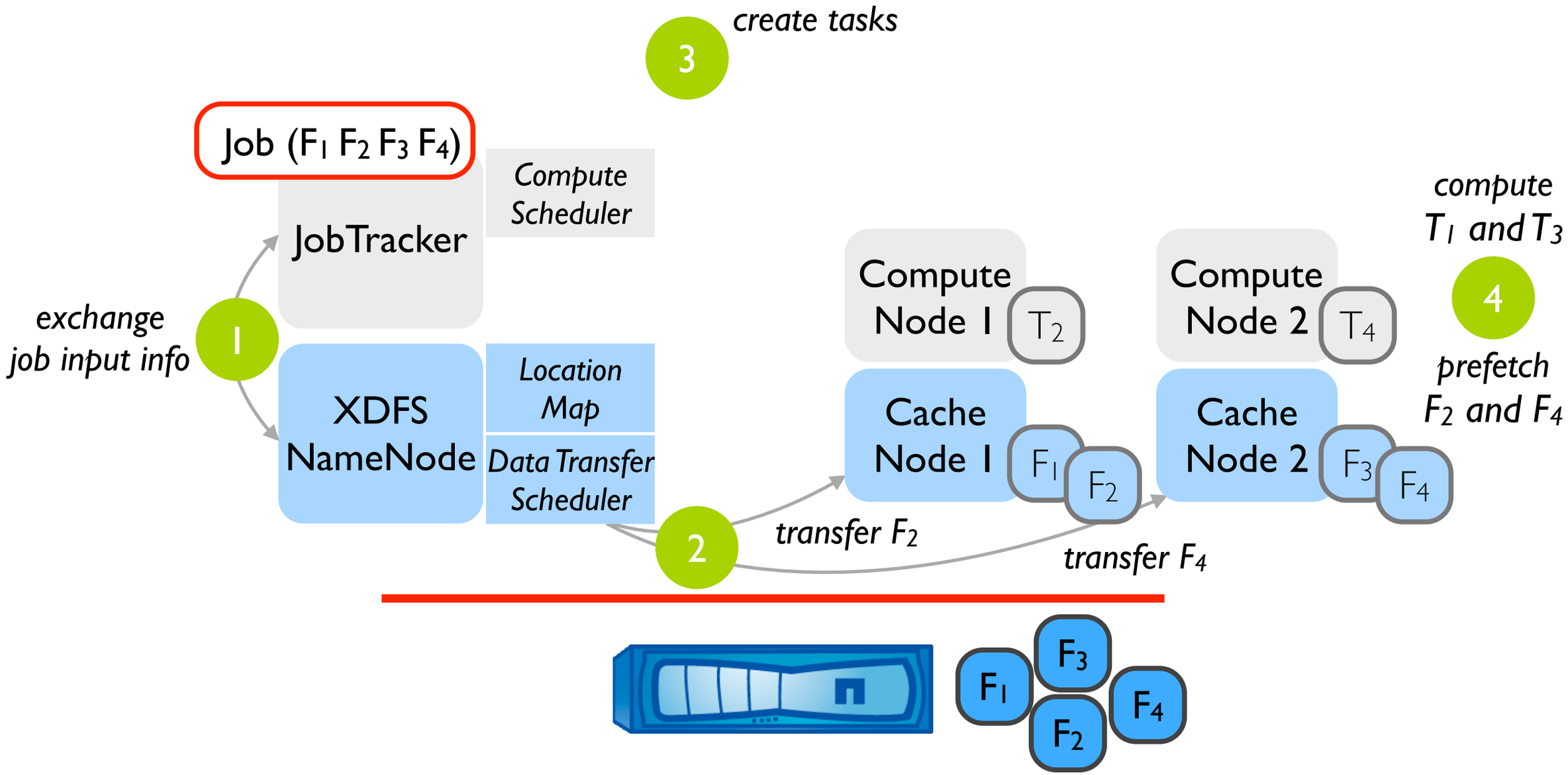
MixApart in Action



MixApart in Action



MixApart in Action



MixApart Prototype

- Re-engineered Hadoop MapReduce and HDFS
  - XDFS cache
    - *Stateless HDFS + NFS support*
  - Compute scheduler
    - *FIFO task scheduler + cache aware*
  - Data transfer scheduler
    - *Module in NameNode*

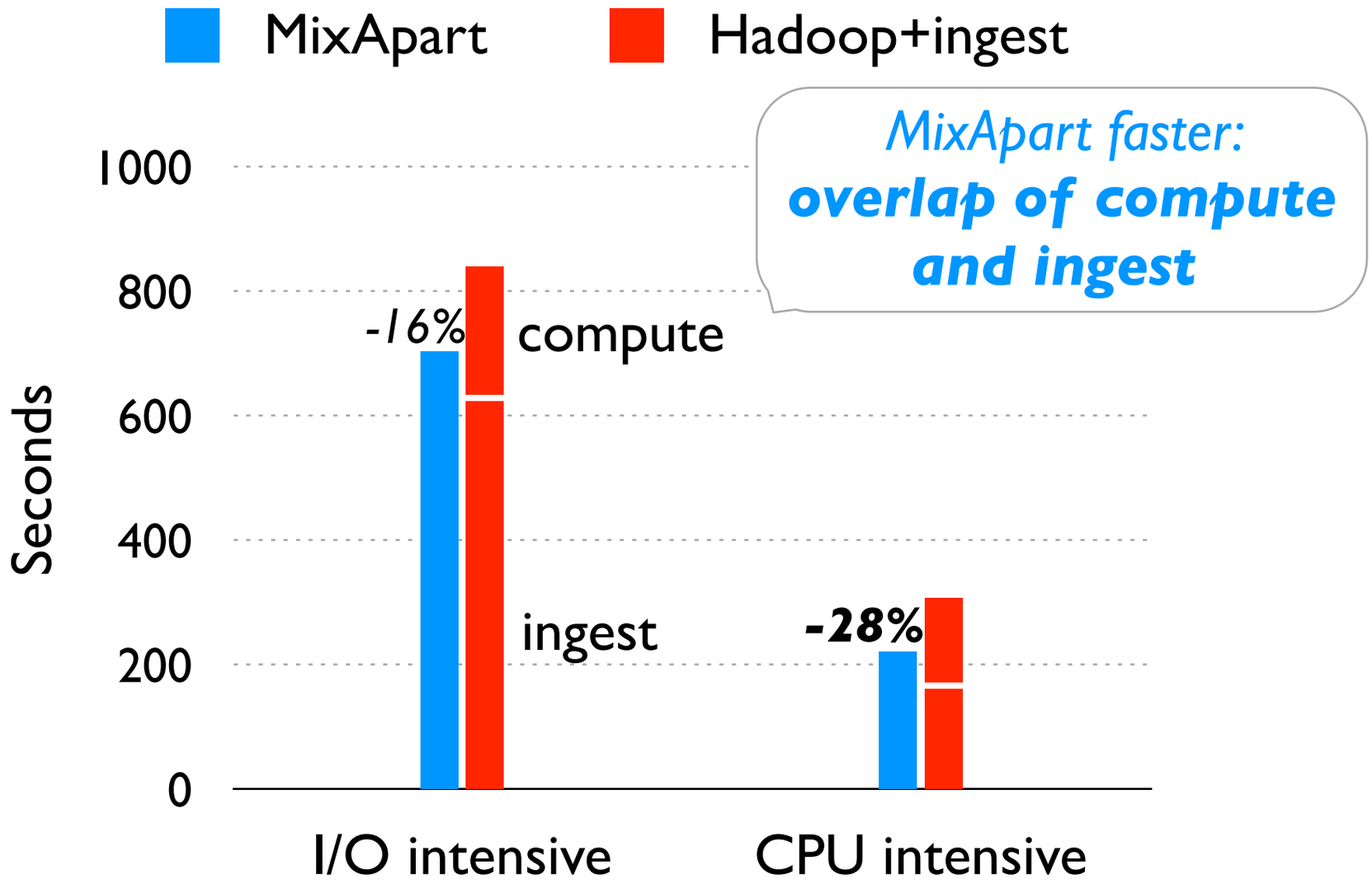
Evaluation on Amazon EC2

- MixApart vs. Hadoop
- *100-core* compute cluster
  - 50 EC2 VM instances
    - 7.5 GB RAM, 850GB local storage
  - *Local VM instance storage* for XDFS cache & HDFS
- *NFS server*
  - EC2 instance
    - 4 EBS volumes in RAID-0 setting
    - *1Gbps* bandwidth for analytics

Microbenchmarks

- *Dataset*
  - *12 days of Wikipedia statistics*
- *Workload*
  - MR Job to *aggregate page views* for regex
  - Job on *uncompressed* data – *I/O intensive*
  - Job on *compressed* data – *CPU intensive*

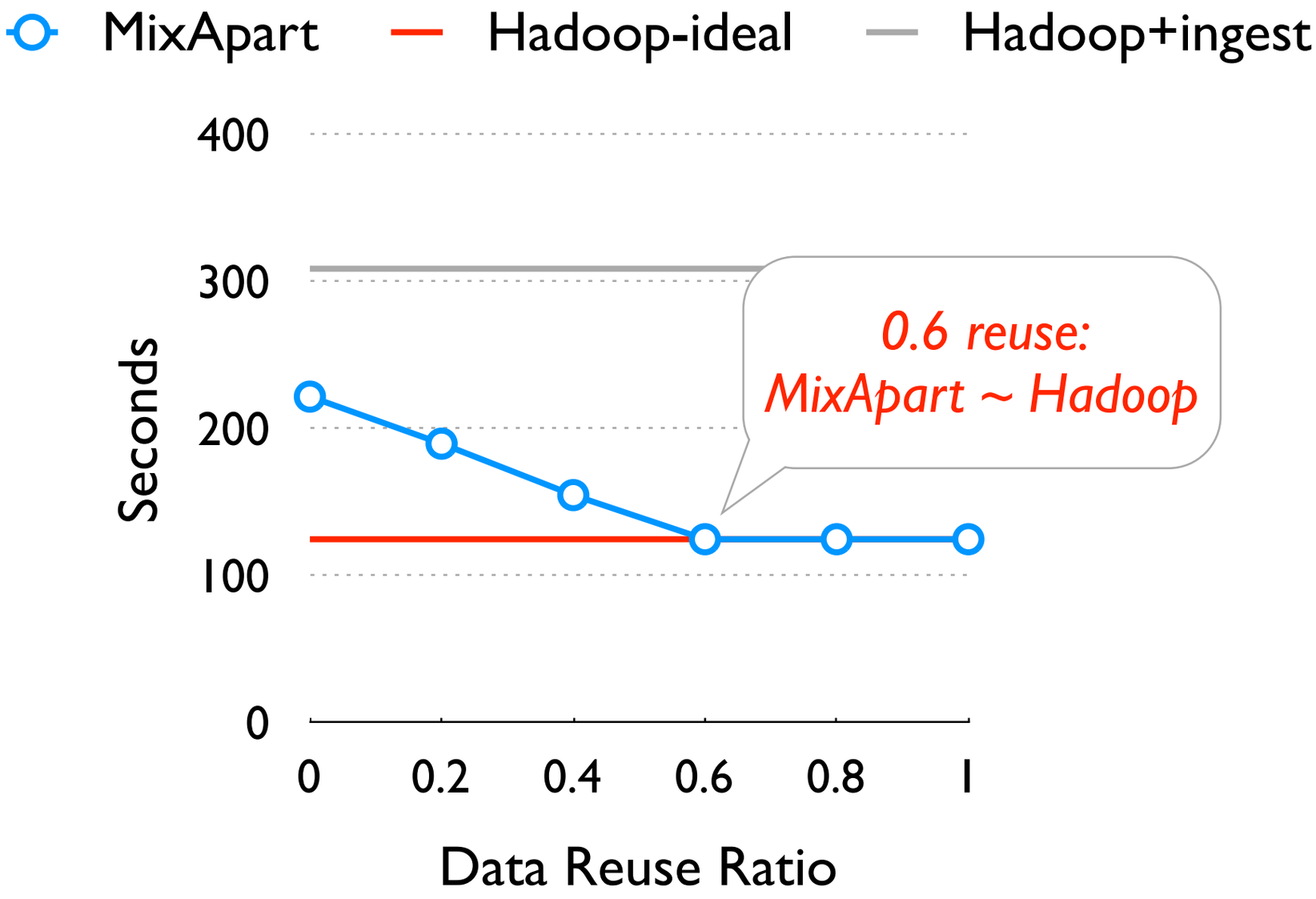
Impact of Ingest



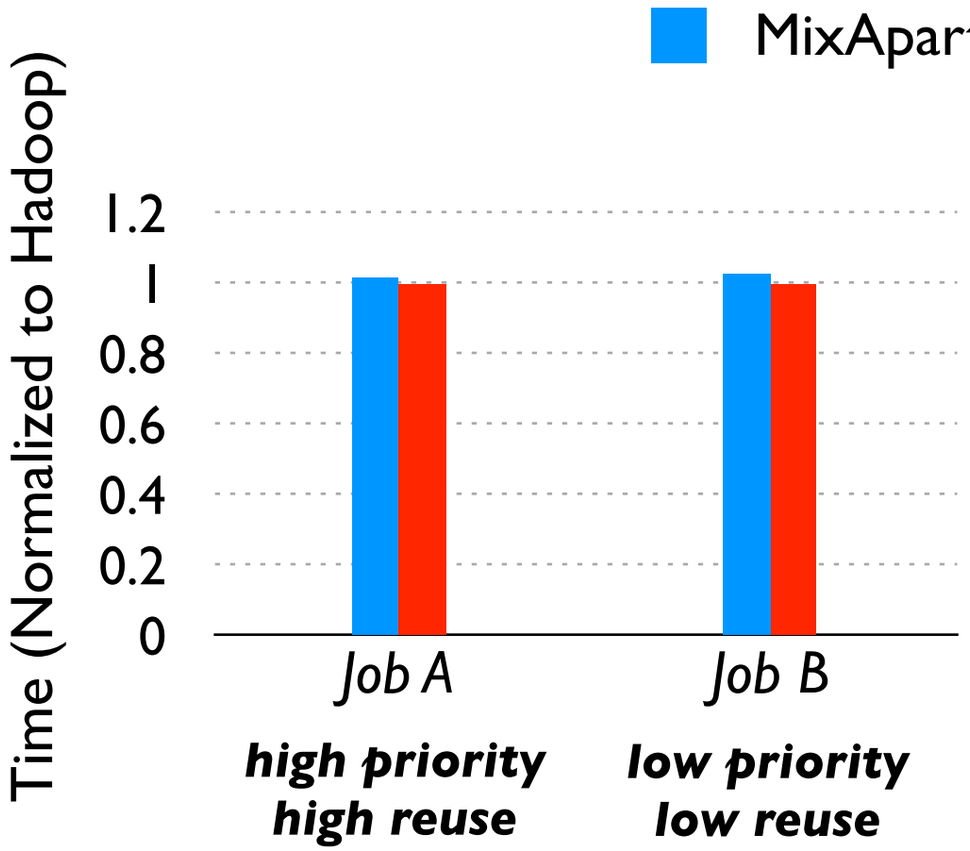
MixApart faster:
overlap of compute and ingest

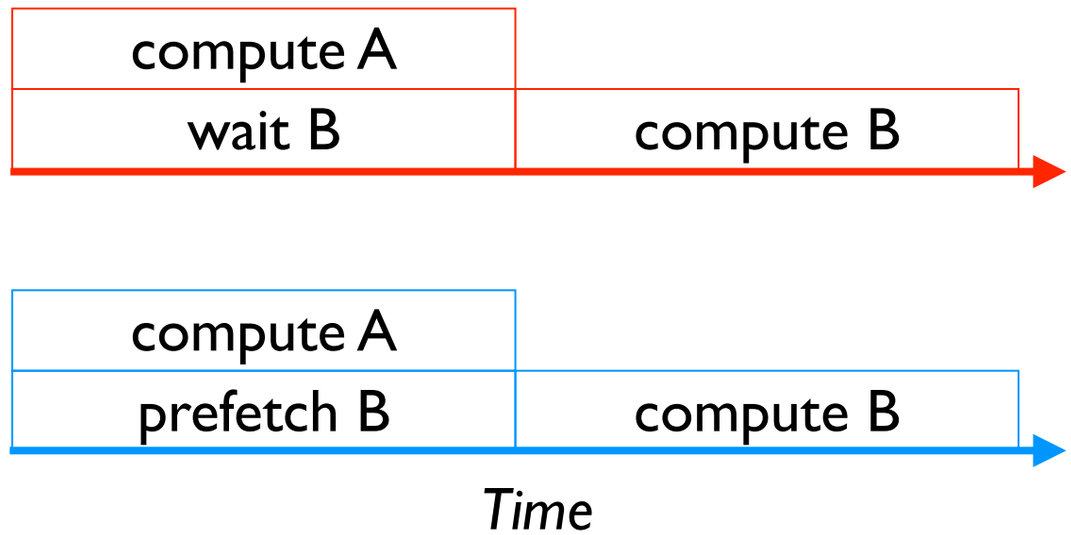
➡ Next: MixApart vs. ***ideal*** *Hadoop with* ***no static ingest***

Microbenchmark Job Durations

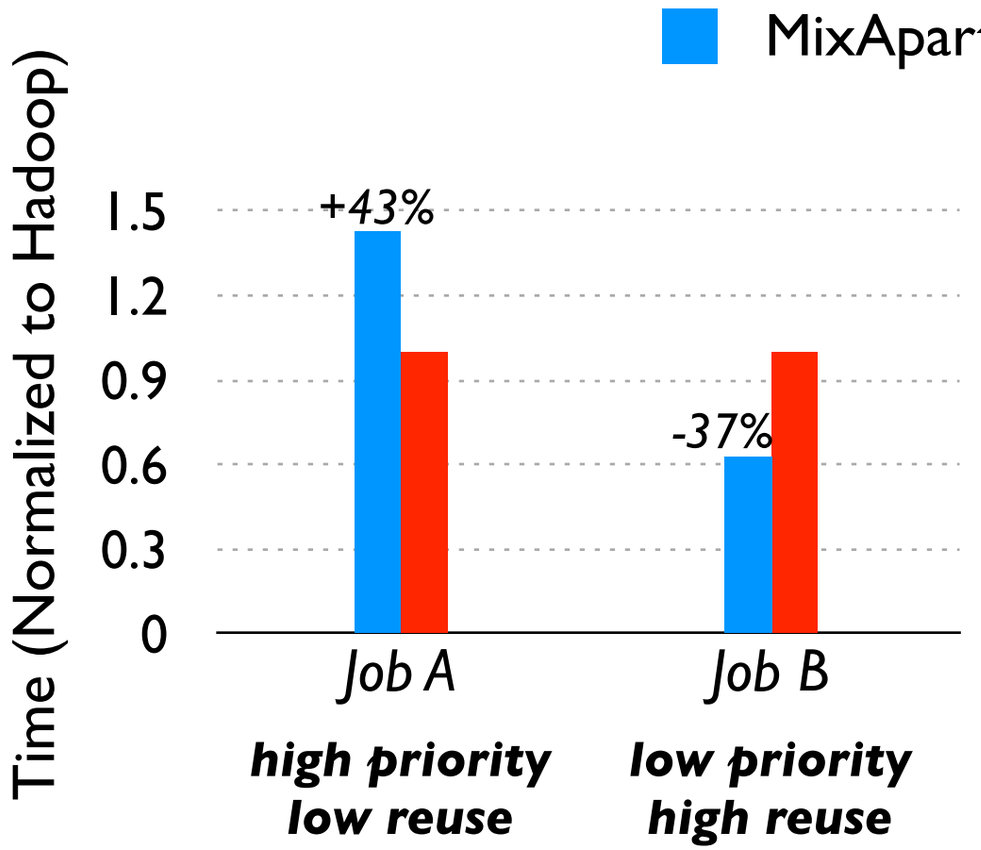


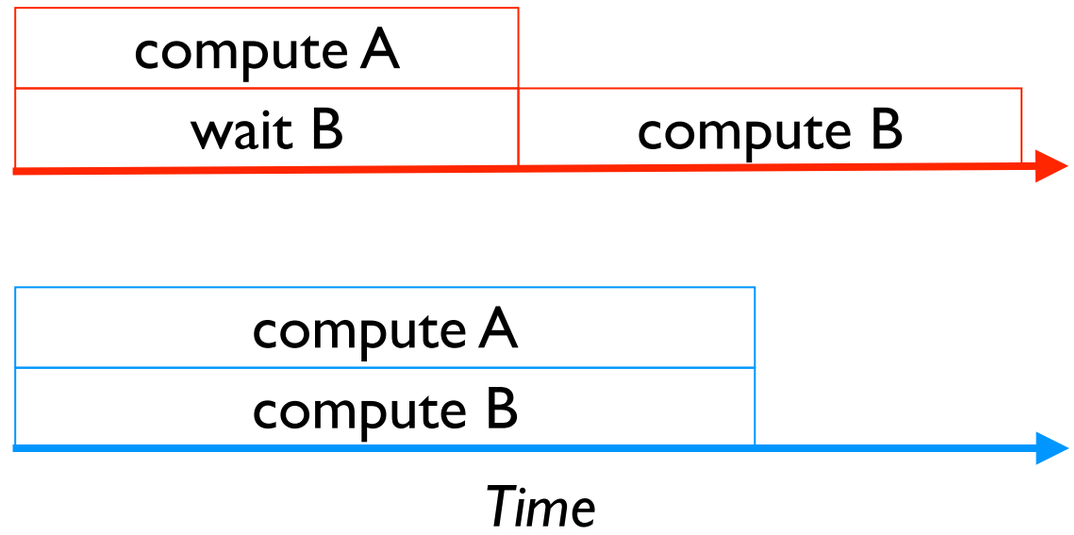
2 Jobs Co-scheduled





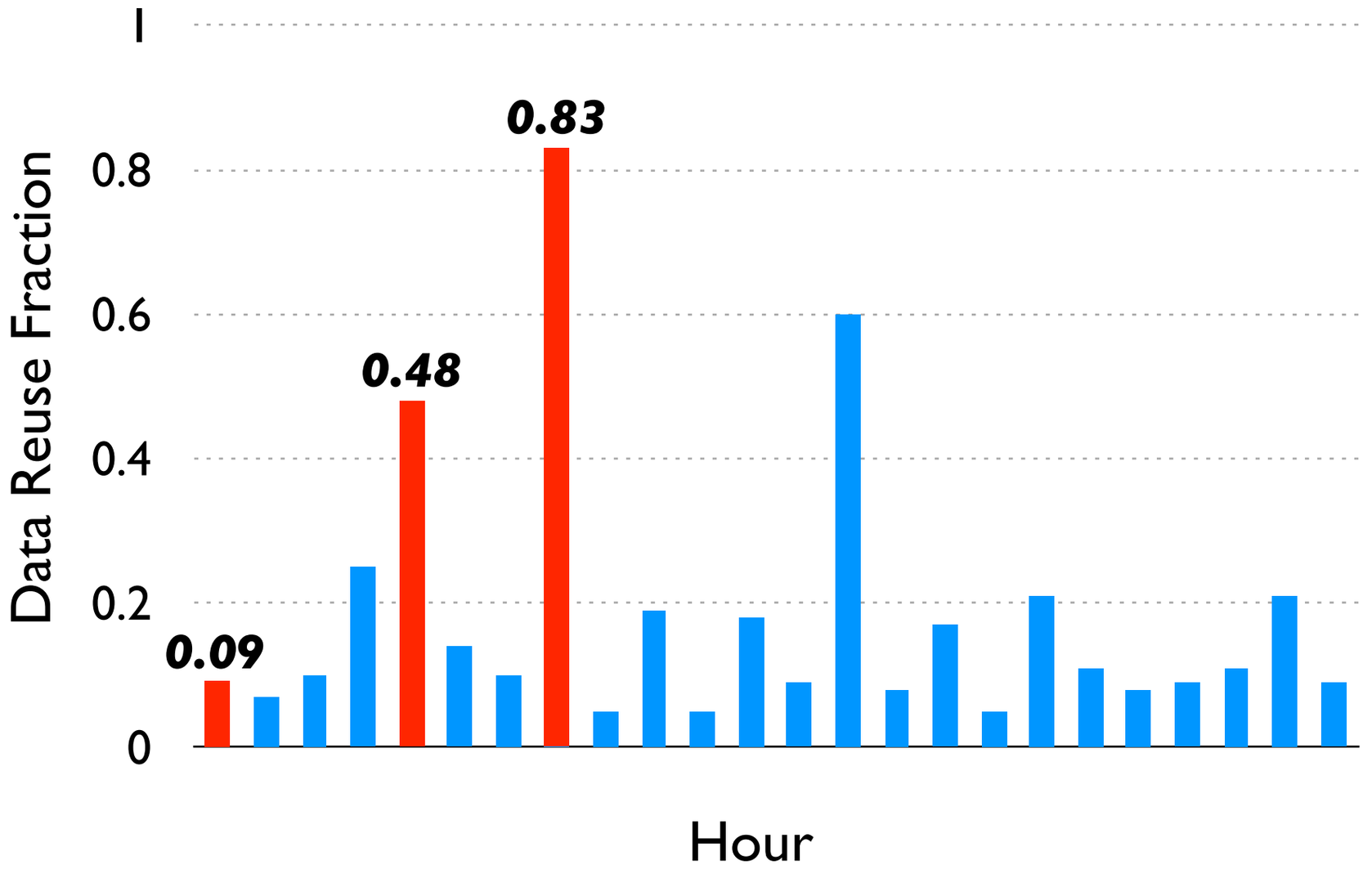
2 Jobs Co-scheduled



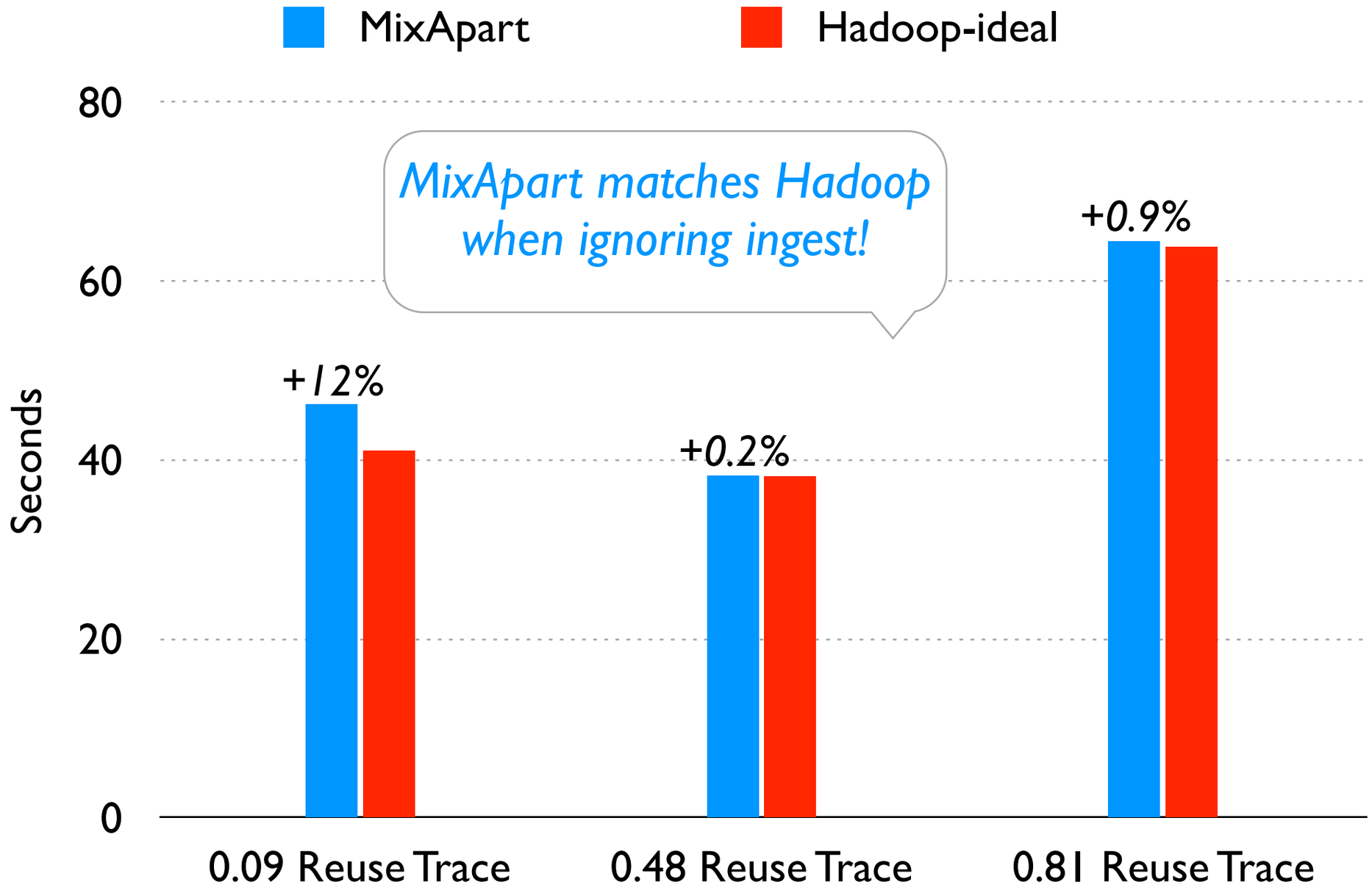


MixApart: ***work conserving*** *compute scheduling*

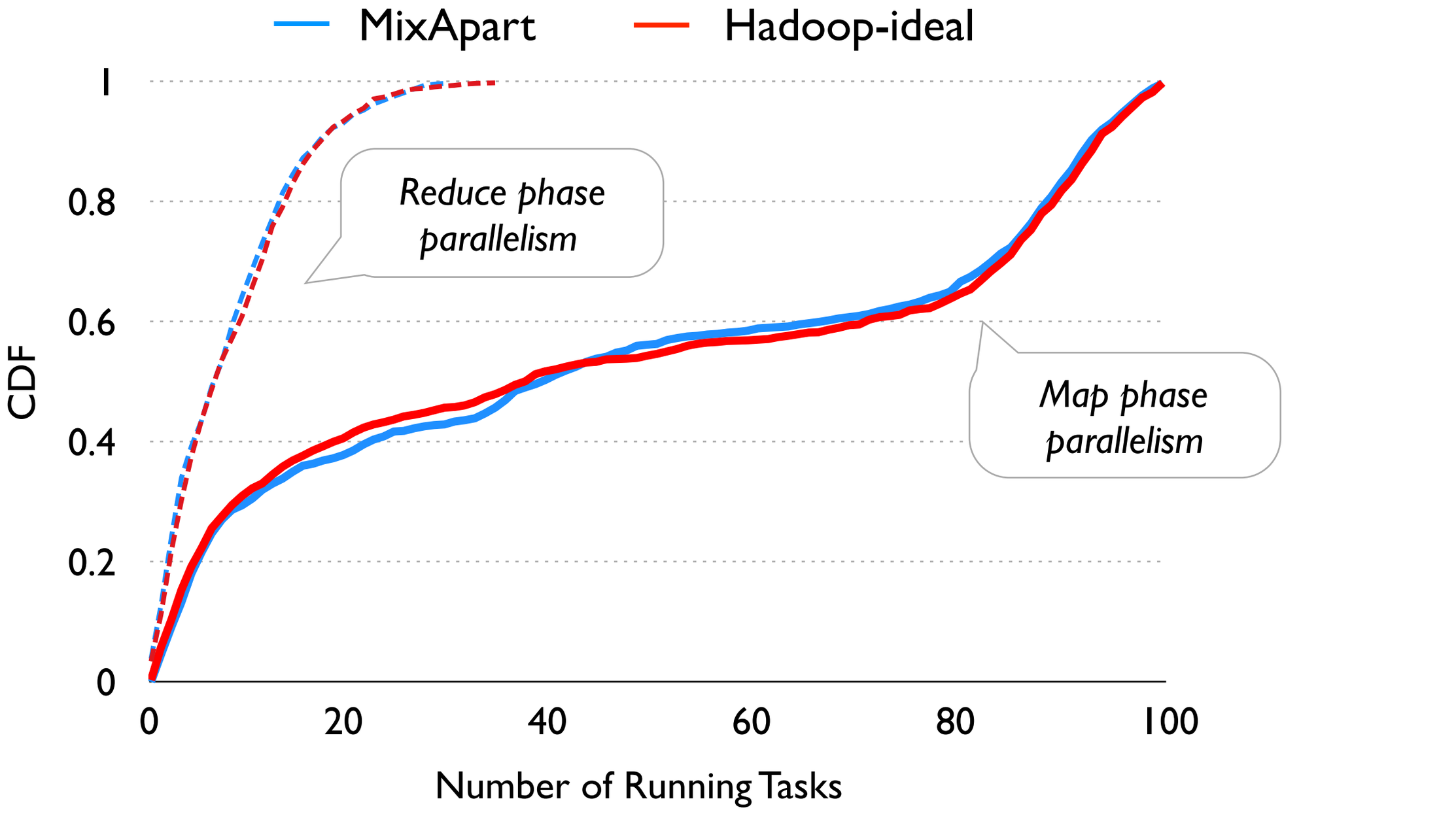
Facebook Hadoop Trace



Facebook Job Durations

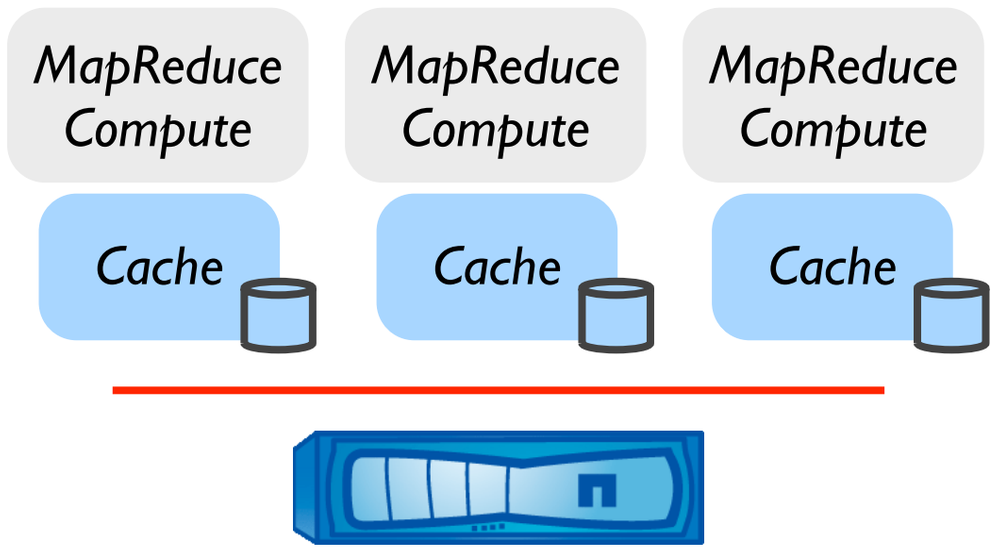


Facebook Compute Concurrency



MixApart Summary

- *MapReduce* analytics on enterprise storage
  - *Enterprise storage* – ***single*** reliable data store
    - Optimized *storage efficiency*
    - Simplified *data management*
  - MixApart *faster* than *ingest-then-compute Hadoop*
  - MixApart *comparable* to *Hadoop with no ingest*



Thank you!

Questions?