Making Sense of Performance in Data Analytics Frameworks

Authors: Kay Ousterhout, Ryan Rasti, Sylvia Ratnasamy, Scott Shenker, Byung-Gon Chun

Presenter: Zi Wang

Why?

- Commonly Accepted mantras
  - Network
  - IO/disk
  - Straggler

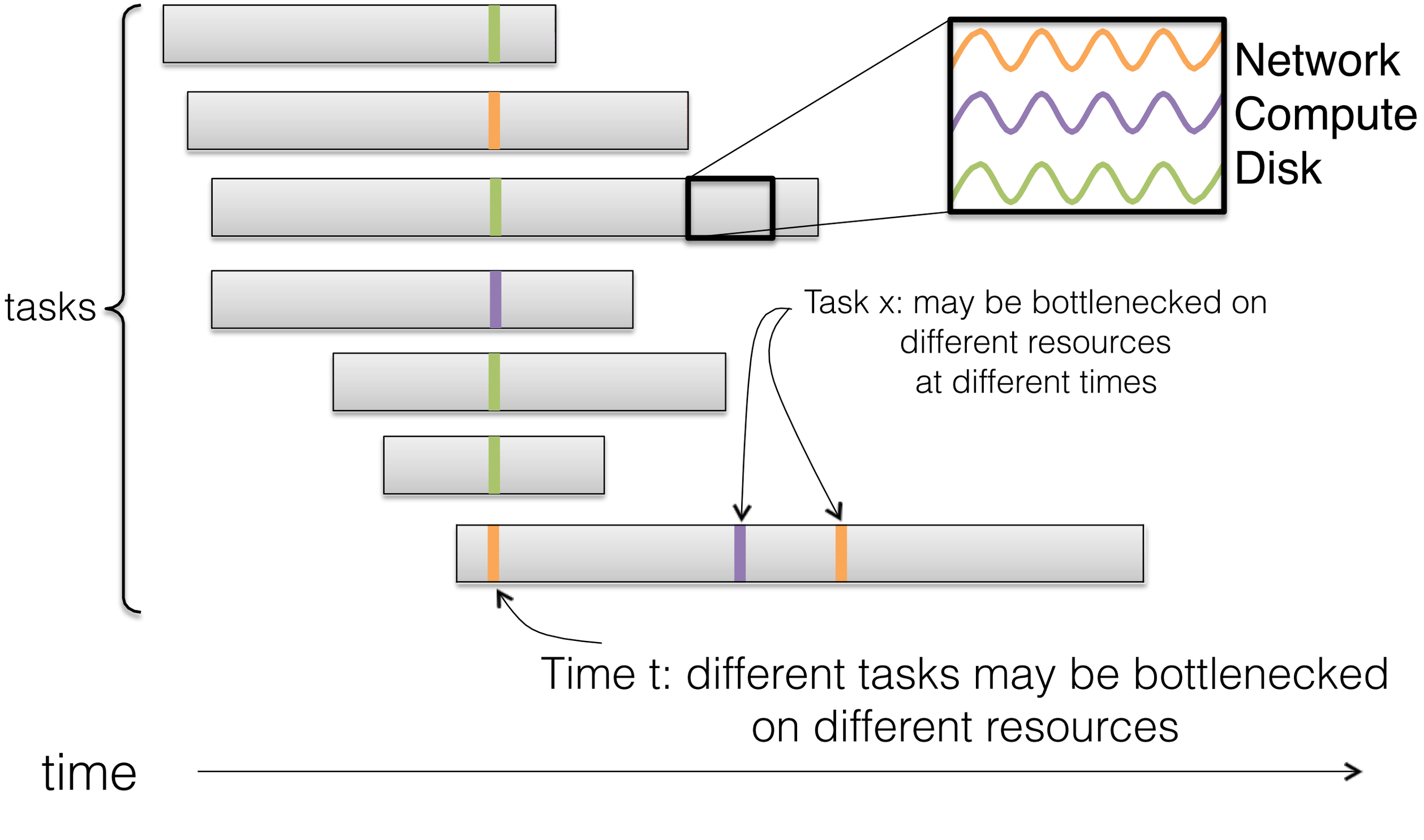
Takeways

- Network can reduce job completion time by at 2%
- I/O optimizations lead to <19% reduction in completion time
- Many straggler causes can be identified and fixed
- CPU is in general the bottleneck

Outline

- Methodology
- Results
- Threats to validity

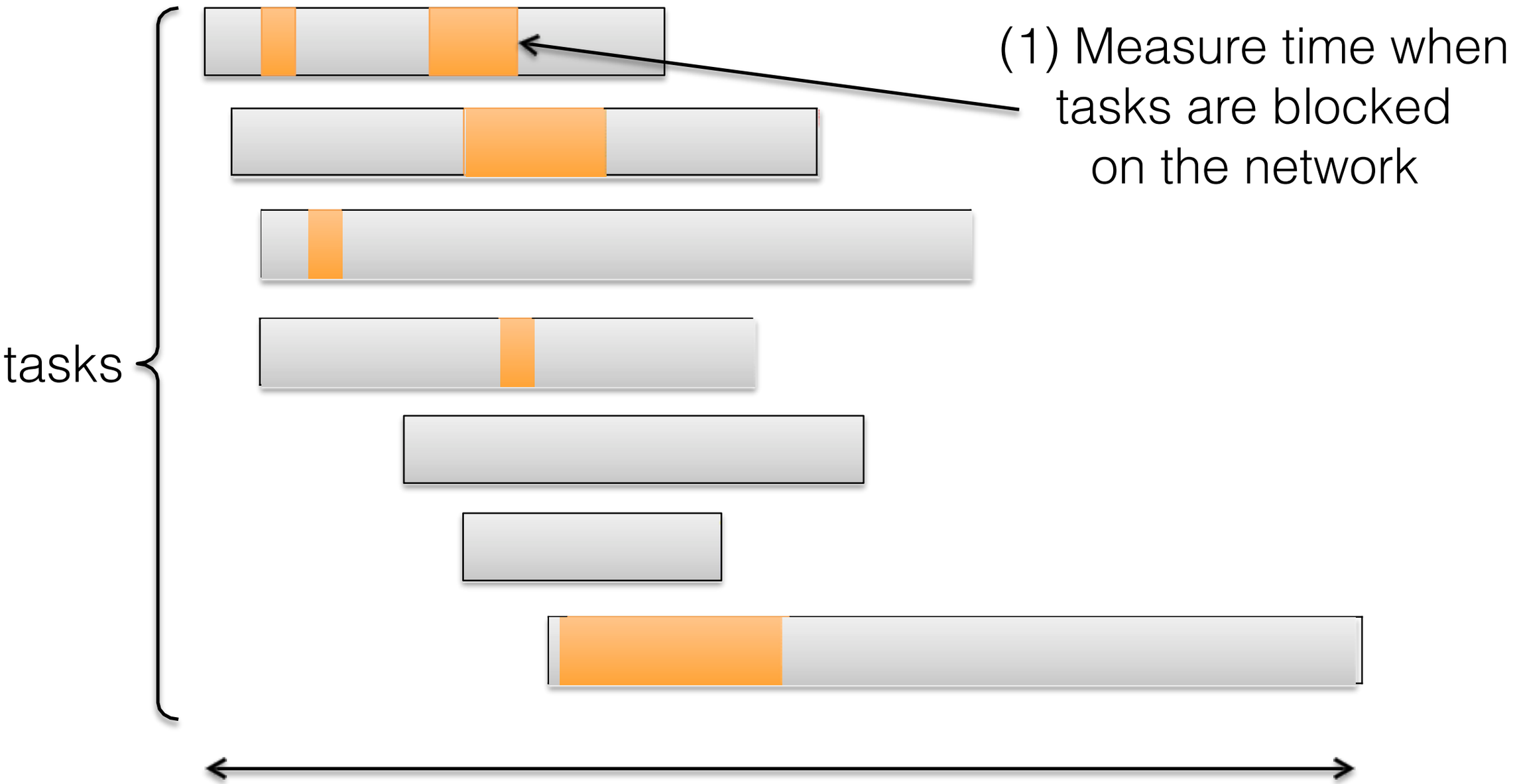
What is the job's bottleneck



Blocked Time Analysis

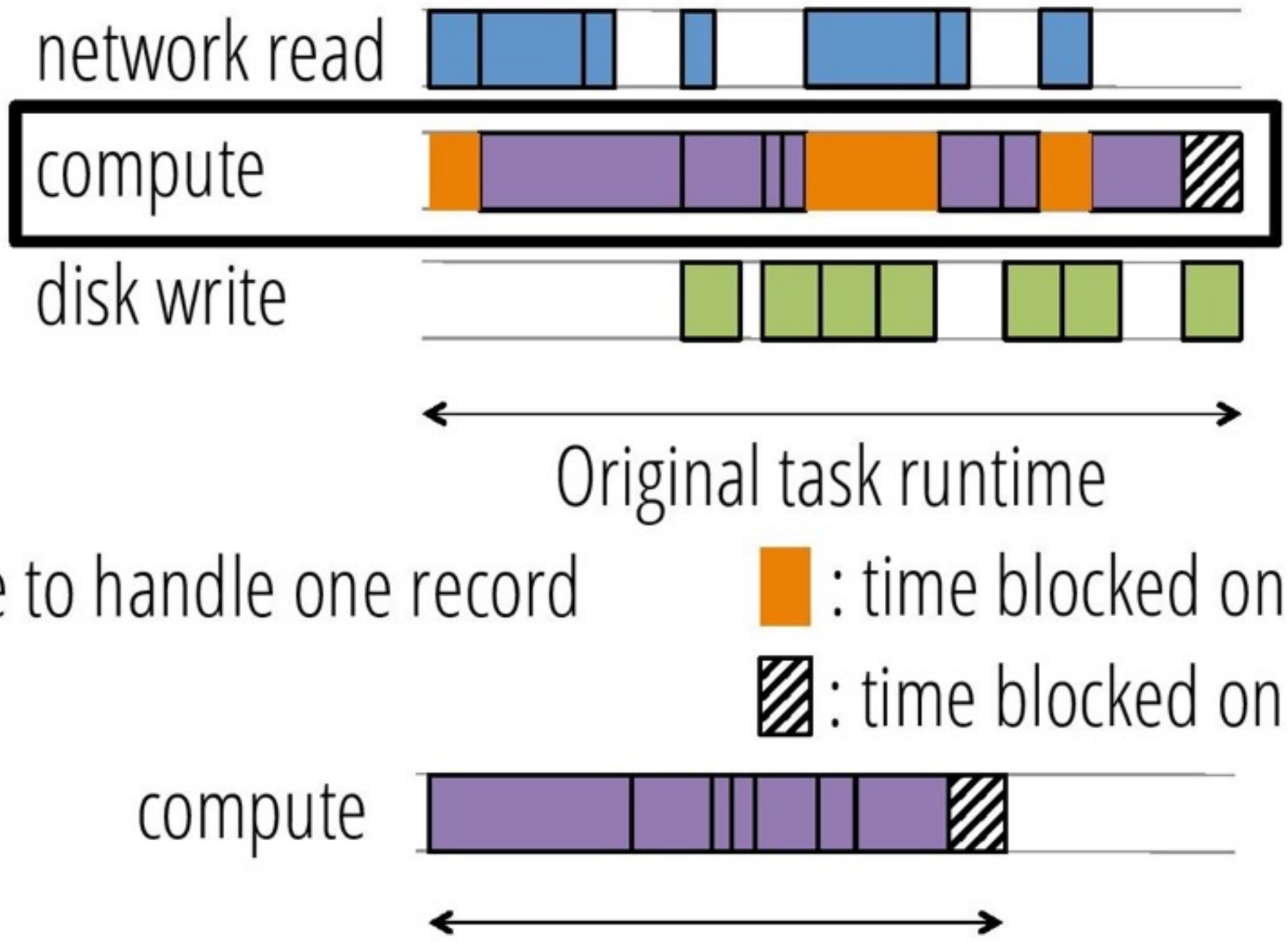
- Time when task is blocked on one resource (e.g network)
- Blocked time analysis: how much faster would the job complete if tasks never blocked on the resource?

An Example of Blocked Time Analysis for Network



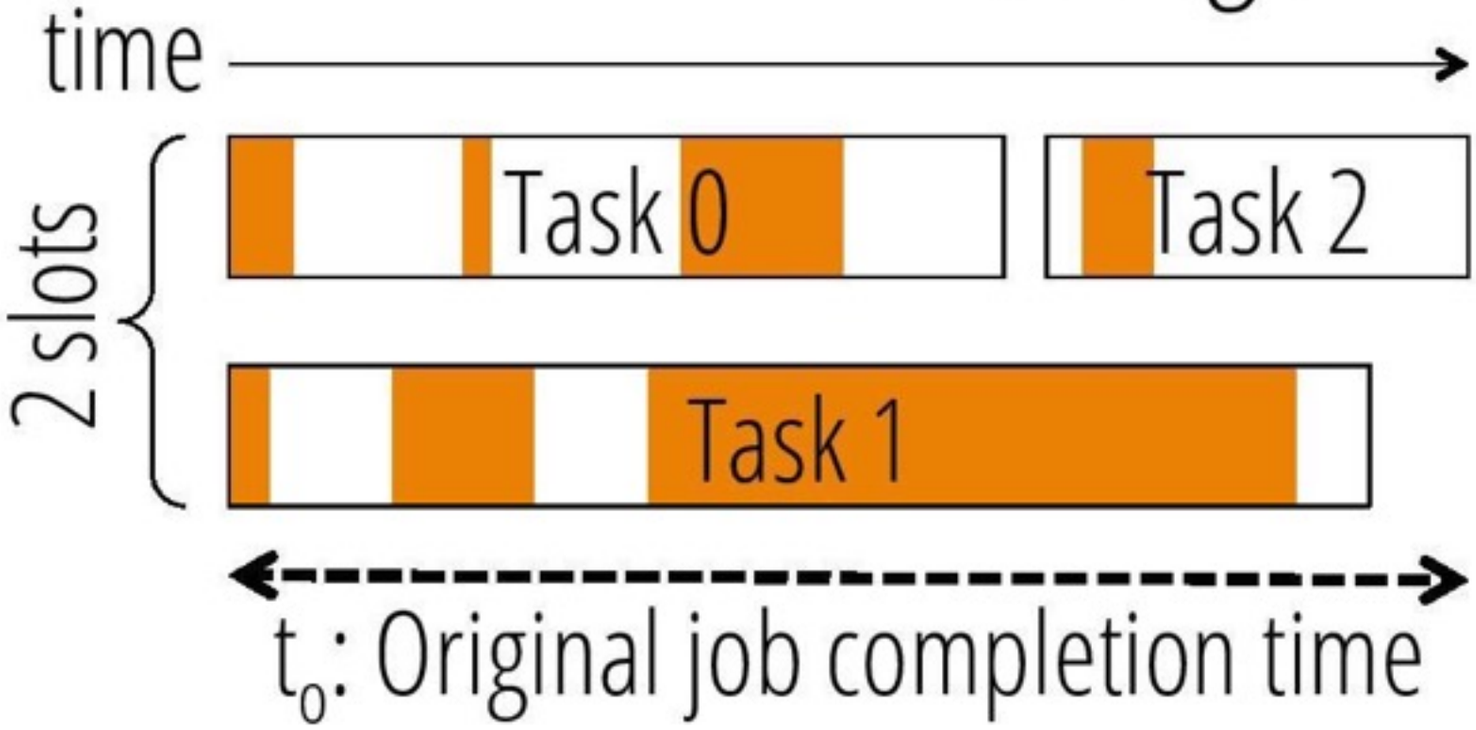
(2) Simulate how job completion time would change

(1) **Measure** time when tasks are blocked on network

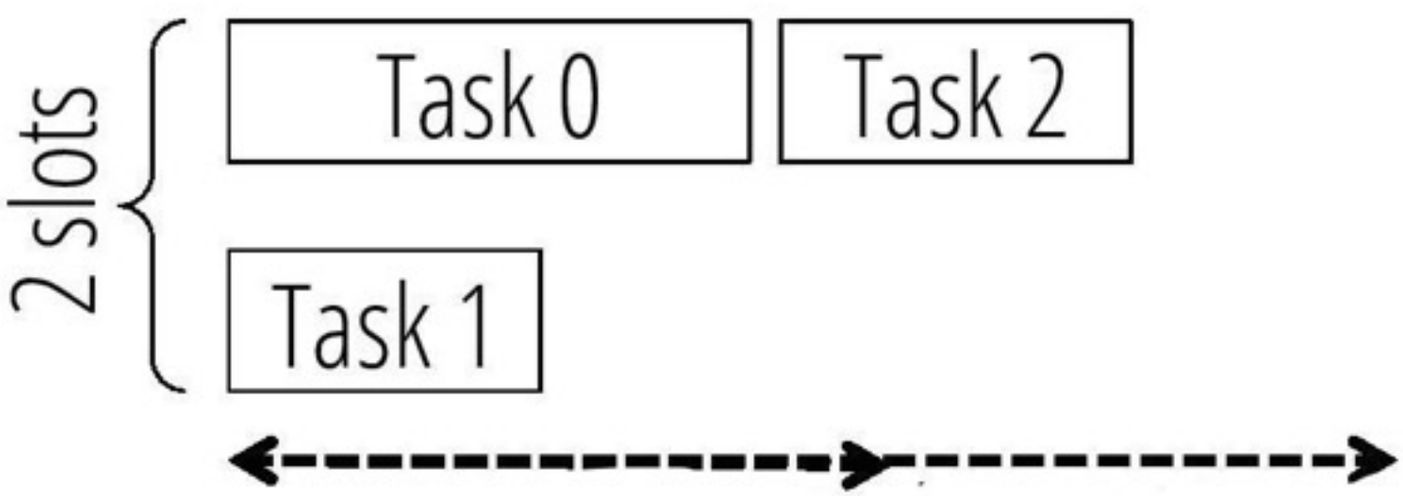


Best case task runtime if network were infinitely fast

(2) **Simulate** how job completion time would change



: time blocked on network



Blocked time analysis: how quickly could a job have completed if a resource were infinitely fast?

Scheduler would have moved Task 2 to slot 2

Experiments Setting

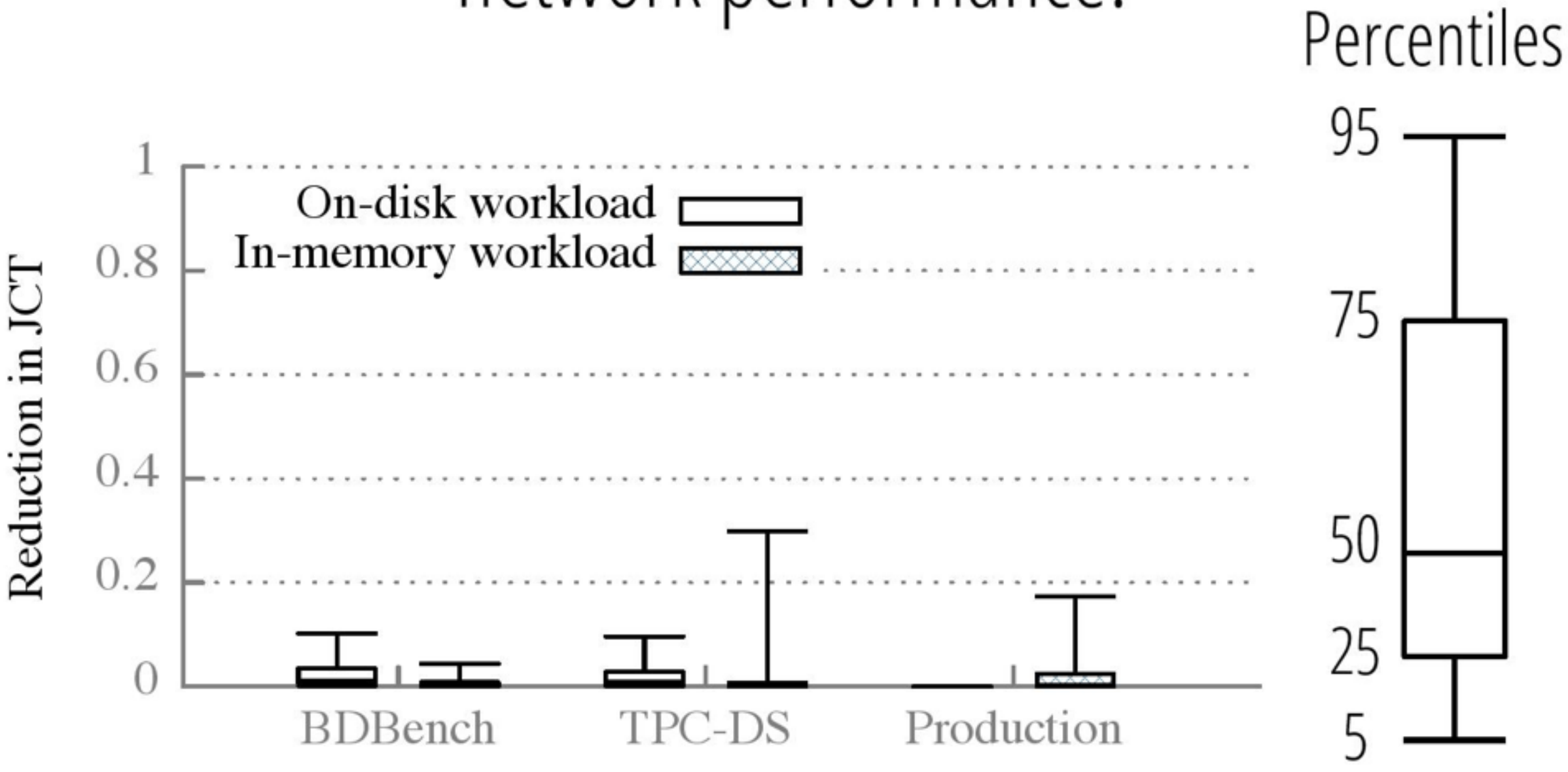
- Big Data Benchmark, 50 queries, 50GB Data, 5 machines
- TPC-DS (Scale 5000), 260 queries, 850GB Data, 20 machines
- Production, 30 queries, tens of GB Data, 9 machines

Experiments Setting

- All three workloads are Spark-SQL workloads

- Coarse-grained analysis of traces from Facebook, Google, Microsoft are used for sanity check

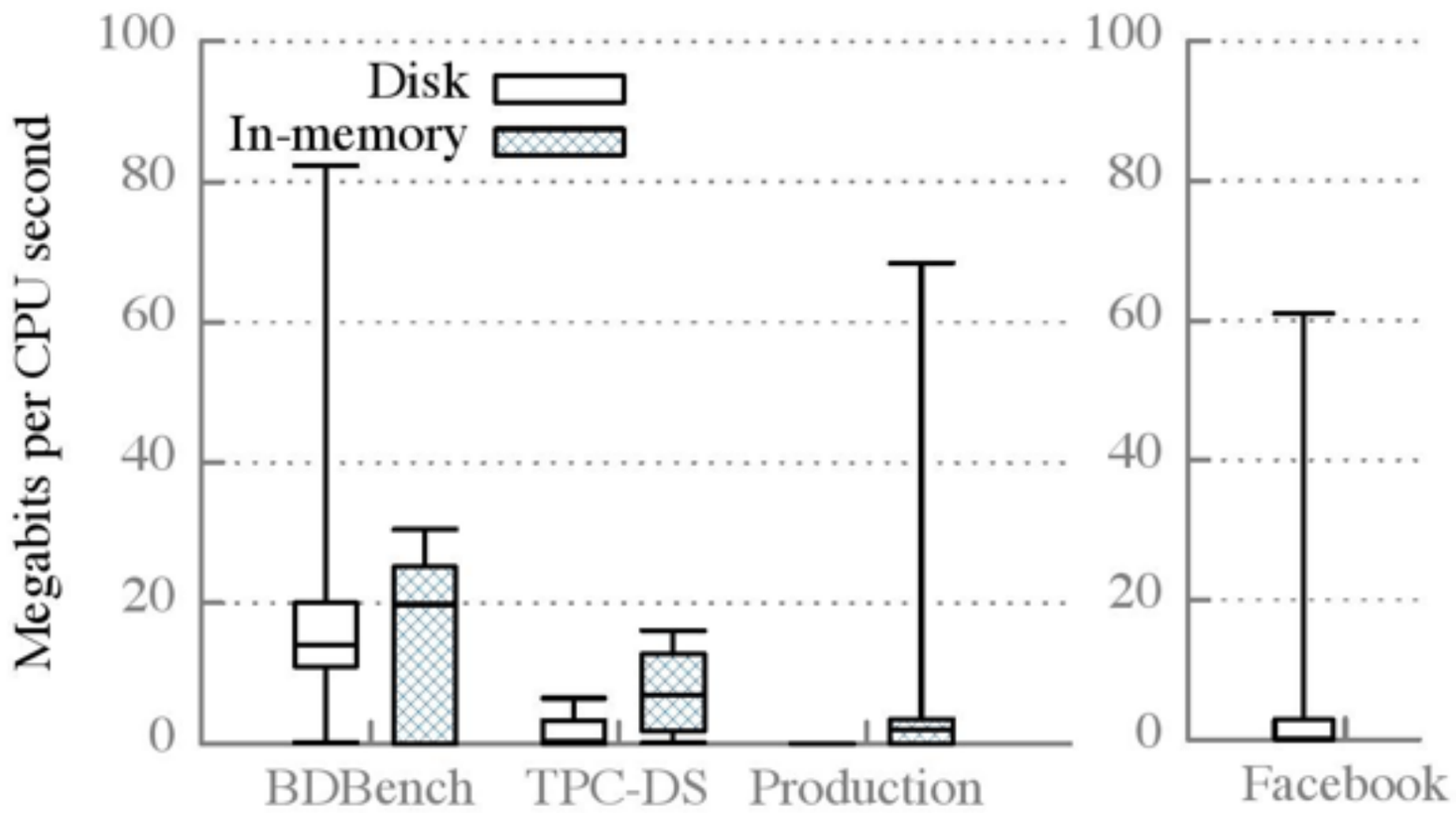
How much faster could jobs get from optimizing network performance?



Median improvement at most 2%

Are jobs network-light?

How much data is transferred per CPU second?



Microsoft '09-'10: **1.9–6.35 Mb / task second**

Google '04-'07: **1.34–1.61 Mb / machine second**

Analysis

- Queries often shuffle and output much less data than they read

- However, the result seems inconsistent from previous work…

Two Reasons

- Incomplete Metric
    - Only look at shuffle time
- Conflation of CPU and network time
    - Sending data over the network has an associated CPU cost

Analysis for I/O

- Compressed data is used, CPU is traded for I/O

- Spark is written in Scala. Data read must be deserialized to Java Objects.

Role of Straggler

- The median reduction from eliminating straggler < 10%
- Common causes: garbage collection, I/O
- Many Stragglers are caused by inherent factors like output size

Threats to Validity

- Only One Framework (Spark)
- Small cluster sizes
- Only three workloads

Related work

- Instead of using Spark, using Naiad can achieve up to 3x speedups going from 1G network to 10G network

- Spark is also memory-efficient, leveraging “in-memory” computation

- Modern hardware (I/O, network links) are also more improved compared to CPU

Comparison to Pivot Tracing

- Static v.s. Dynamic

- Resource Directed Analysis v.s. Crossing Boundaries Analysis

References

- Making Sense of Performance in Data Analytics Frameworks
- Pivot Tracing: Dynamic Causal Monitoring for Distributed Systems
- The impact of fast networks on graph analytics
- Project Tungsten: Bringing Apache Spark Closer to Bare Metal

“The only way to get ahead is to find errors in conventional wisdom.”

–Larry Ellison