On The [Ir]relevance of Network Performance for Data Processing

Animesh Trivedi, Patrick Stuedi, Jonas Pfefferle, Radu Stoica, Bernard Metzler, Ioannis Koltsidas, Nikolas Ioannou

IBM Research, Zurich

How [Ir]relevant is the Network?

Making Sense of Performance in Data Analytics Frameworks

Kay Ousterhout[*], Ryan Rasti[*†◇], Sylvia Ratnasamy[*], Scott Shenker[*†], Byung-Gon Chun[‡]
[*]UC Berkeley, [†]ICSI, [◇]VMware, [‡]Seoul National University

Network optimizations can only reduce job completion time by a median of at most 2%. The network is not a bottleneck because much less data is sent over the network than is transferred to and from disk. As a result, network I/O is mostly irrelevant to overall performance, even on 1Gbps networks.

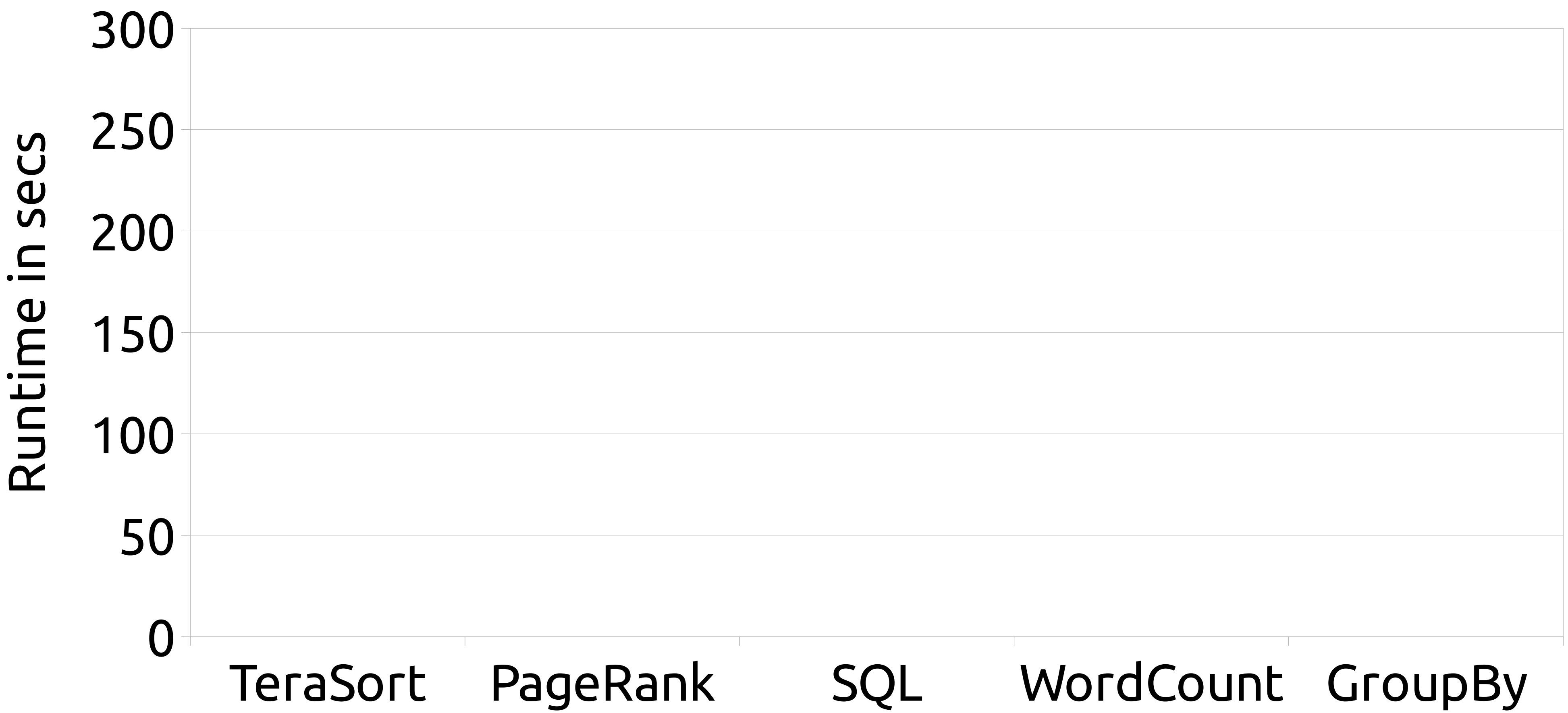
How [Ir]relevant is the Network?

Making Sense of Performance in Data Analytics Frameworks

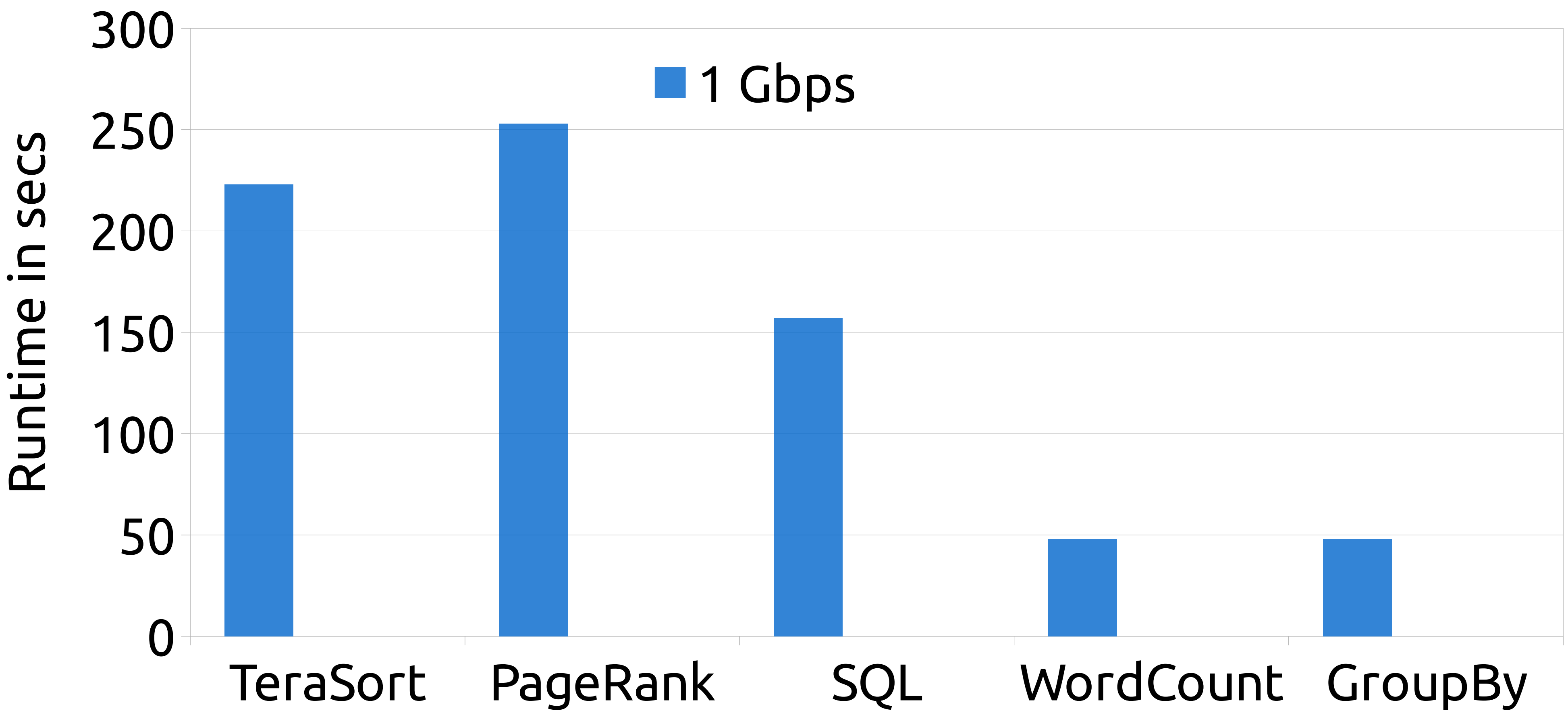
Kay Ousterhout*, Ryan Rasti*†◇, Sylvia Ratnasamy*, Scott Shenker*†, Byung-Gon Chun‡
*UC Berkeley, †ICSI, ◇VMware, ‡Seoul National University

Network optimizations can only reduce job completion time by a median of at most 2%. The network is not a bottleneck because much less data is sent over the network than is transferred to and from disk. As a result, network I/O is mostly irrelevant to overall performance, even on 1Gbps networks.

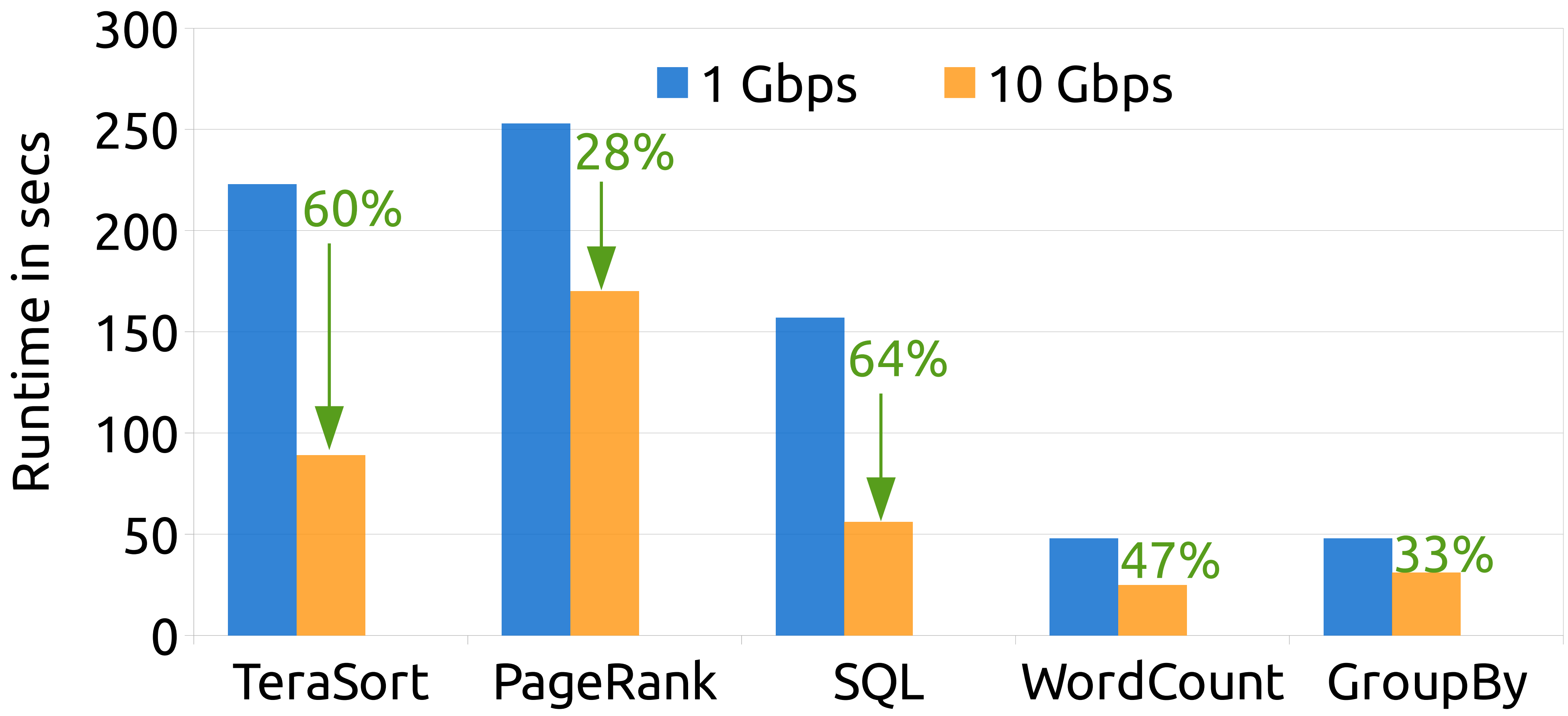
How [Ir]relevant is the Network?



How [Ir]relevant is the Network?

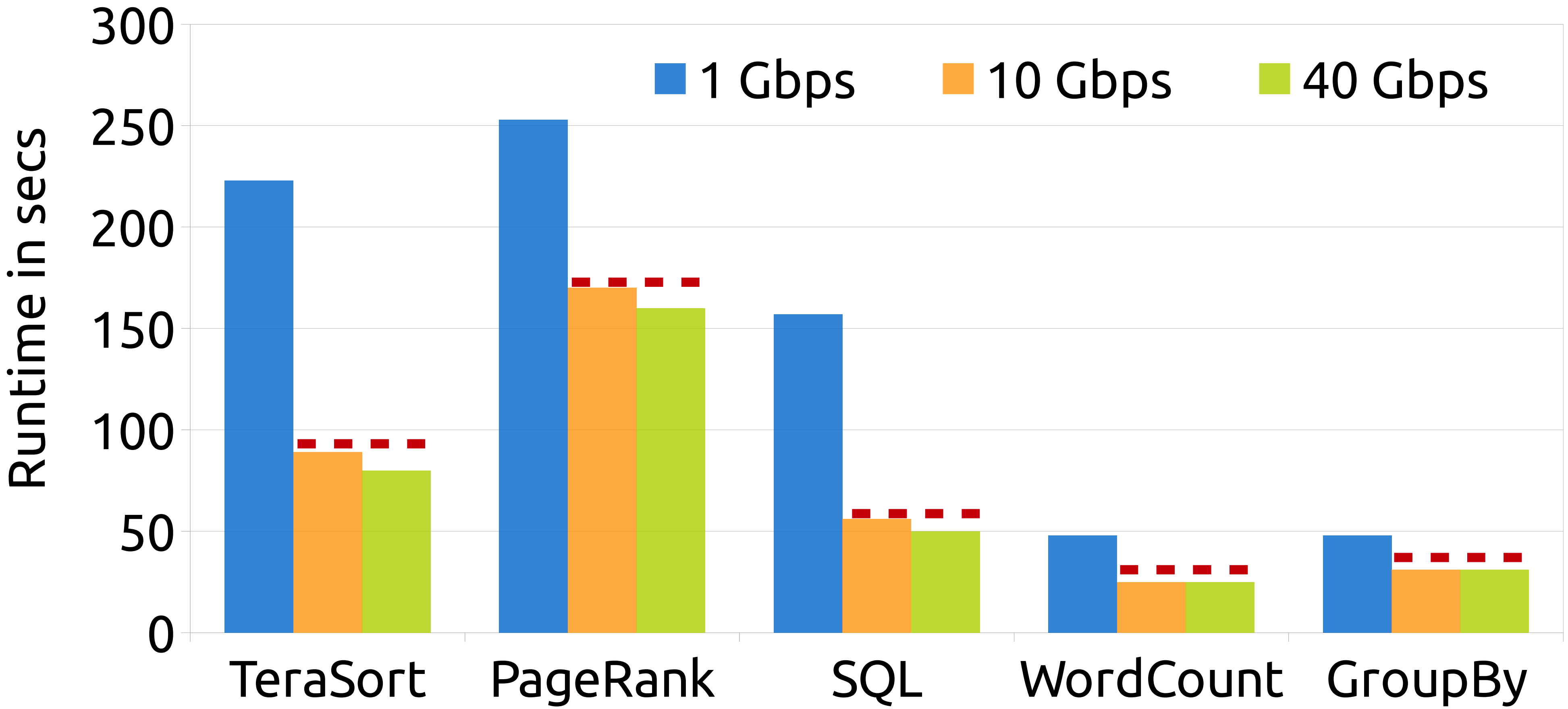


How [Ir]relevant is the Network?



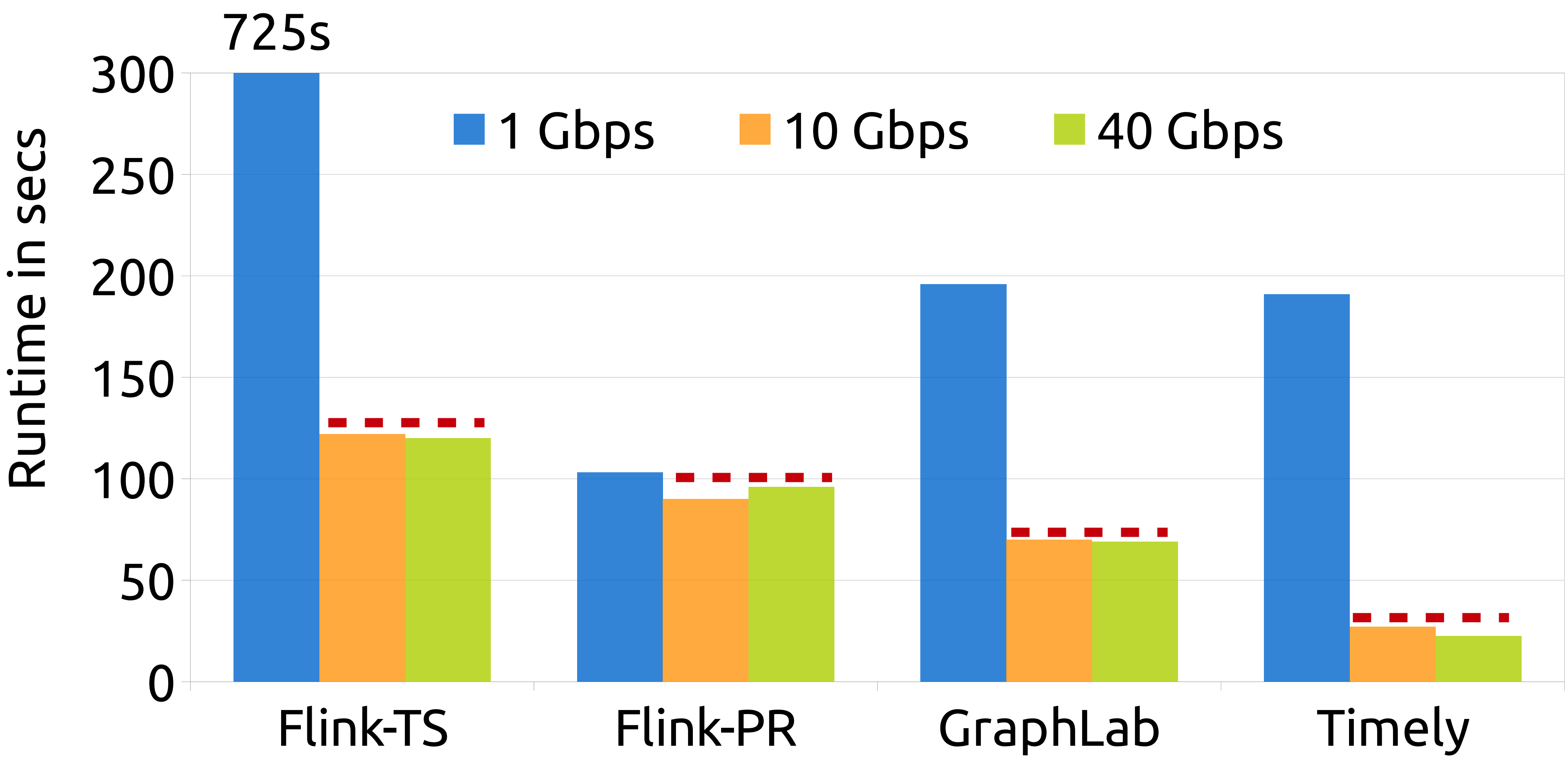
1 Network IO is very relevant - up to 64%

How [Ir]relevant is the Network?

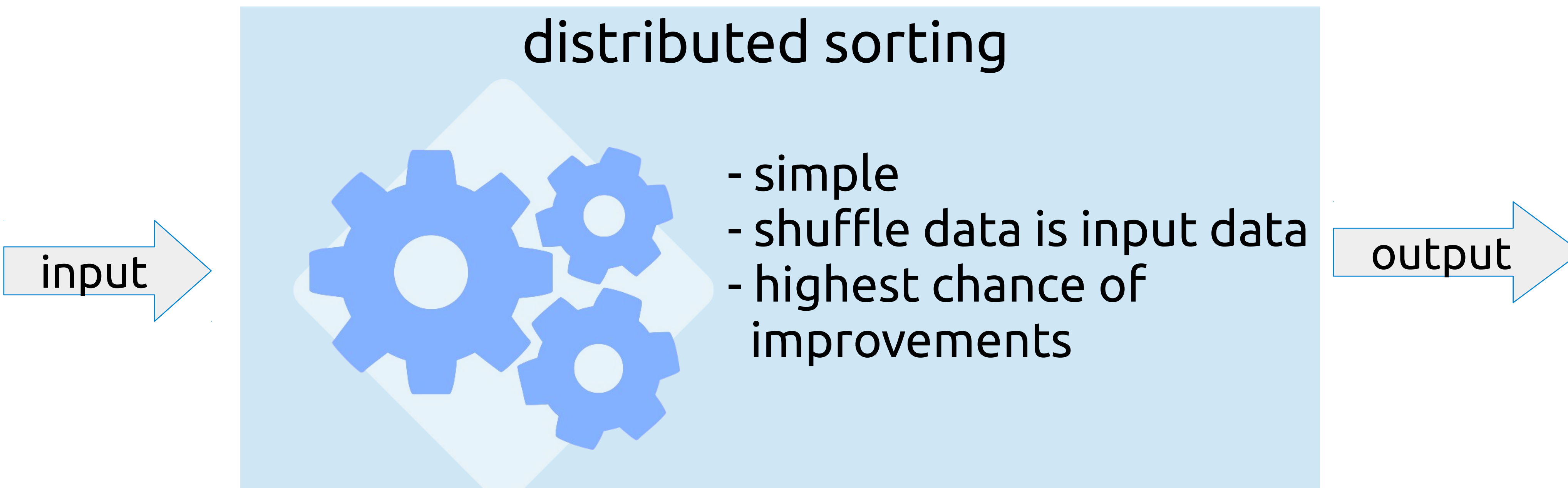


❶ Network IO is very relevant - up to 64% ??

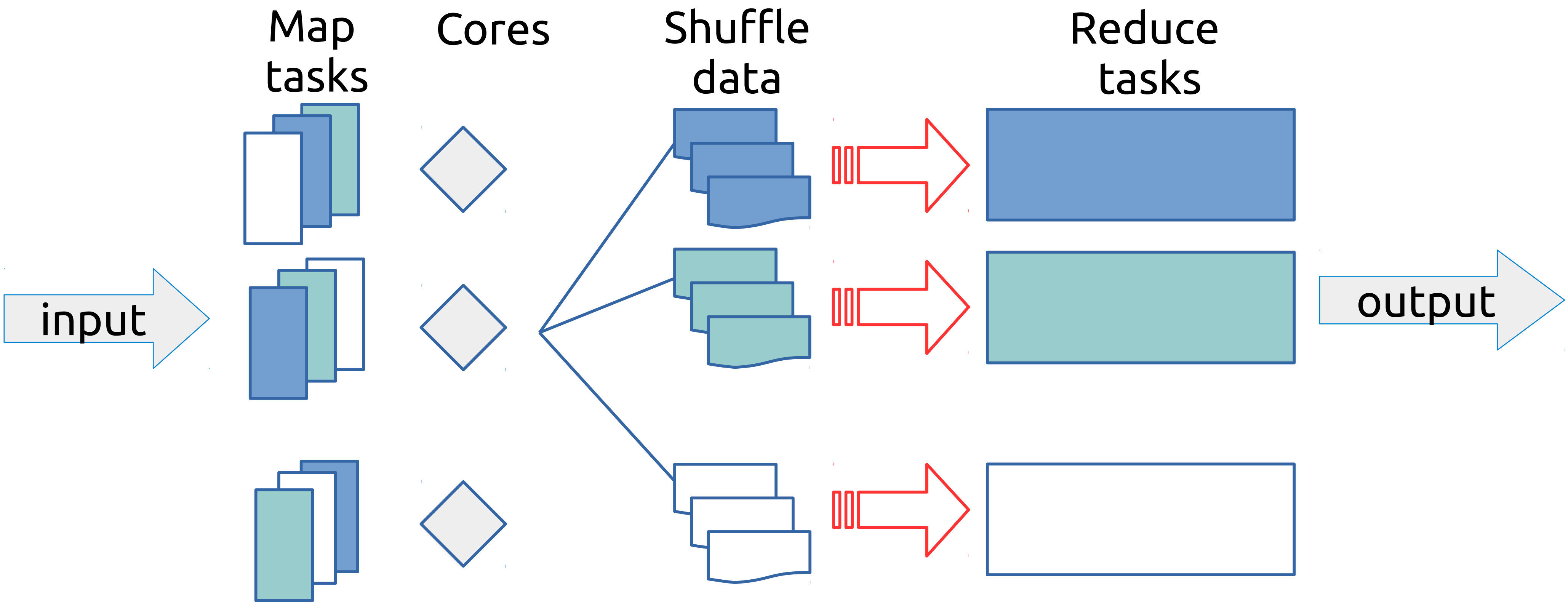
Is It Spark Specific?



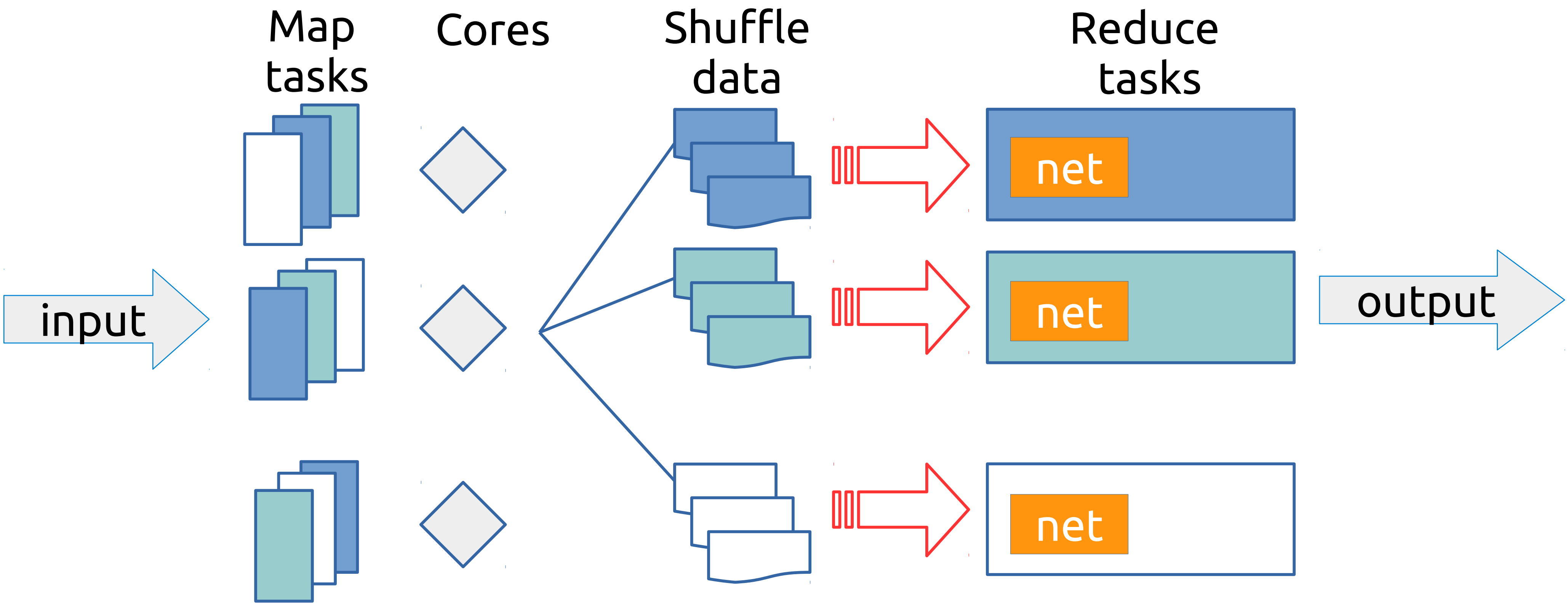
Spark TeraSort: The Shuffle Story



Spark TeraSort: The Shuffle Story

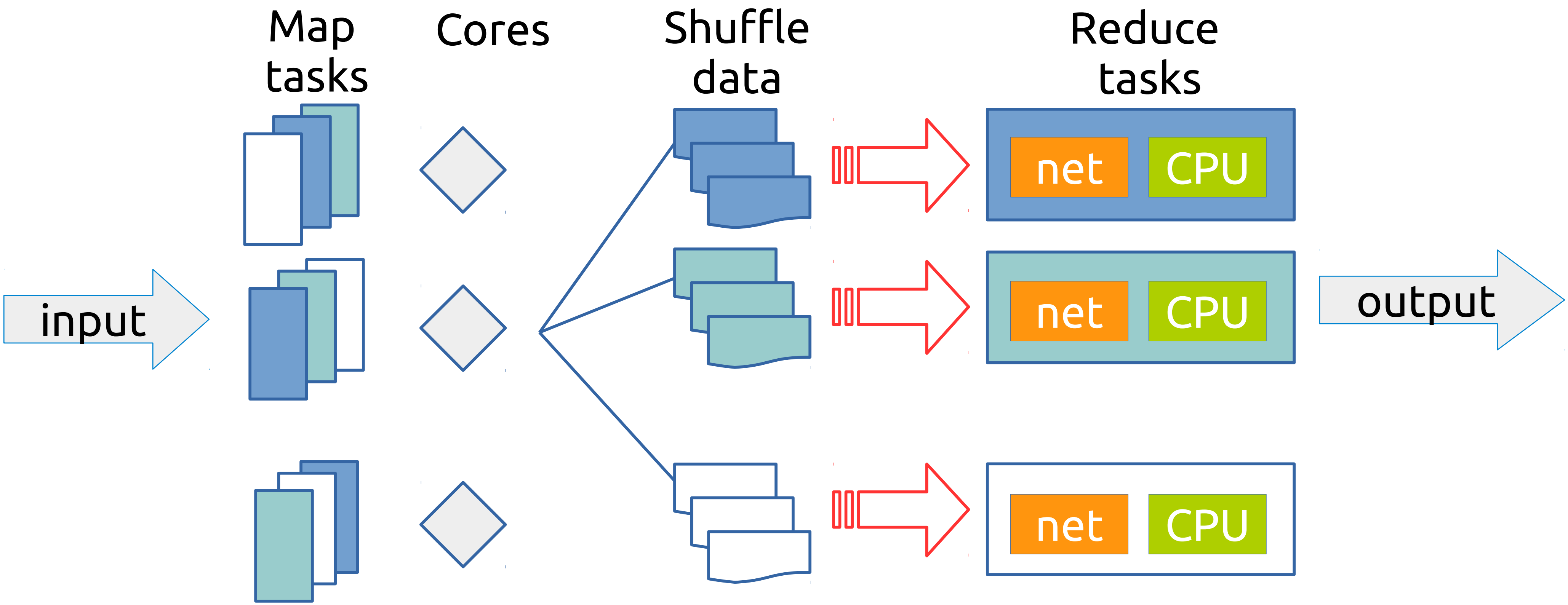


Spark TeraSort: The Shuffle Story



reading in shuffle data

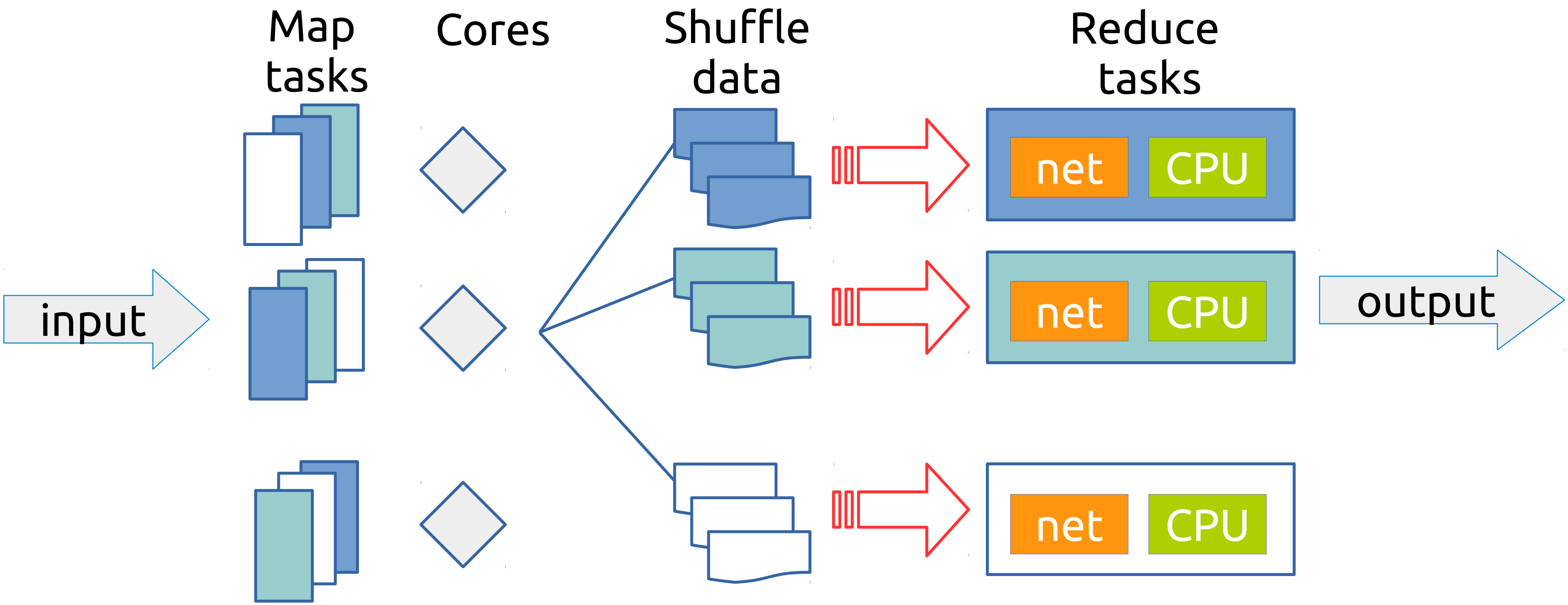
Spark TeraSort: The Shuffle Story



reading in
shuffle data

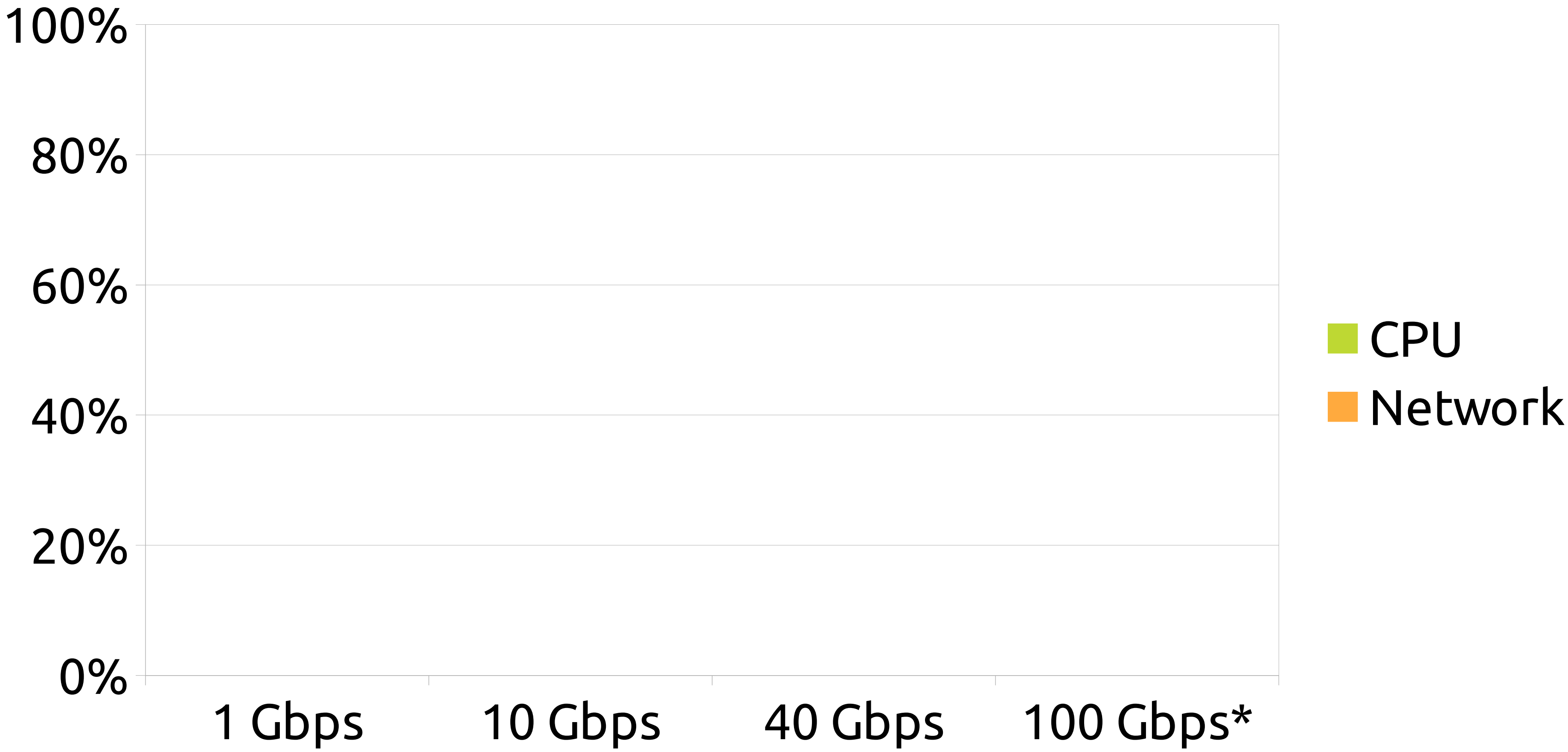
sorting
shuffle data

Spark TeraSort: The Shuffle Story

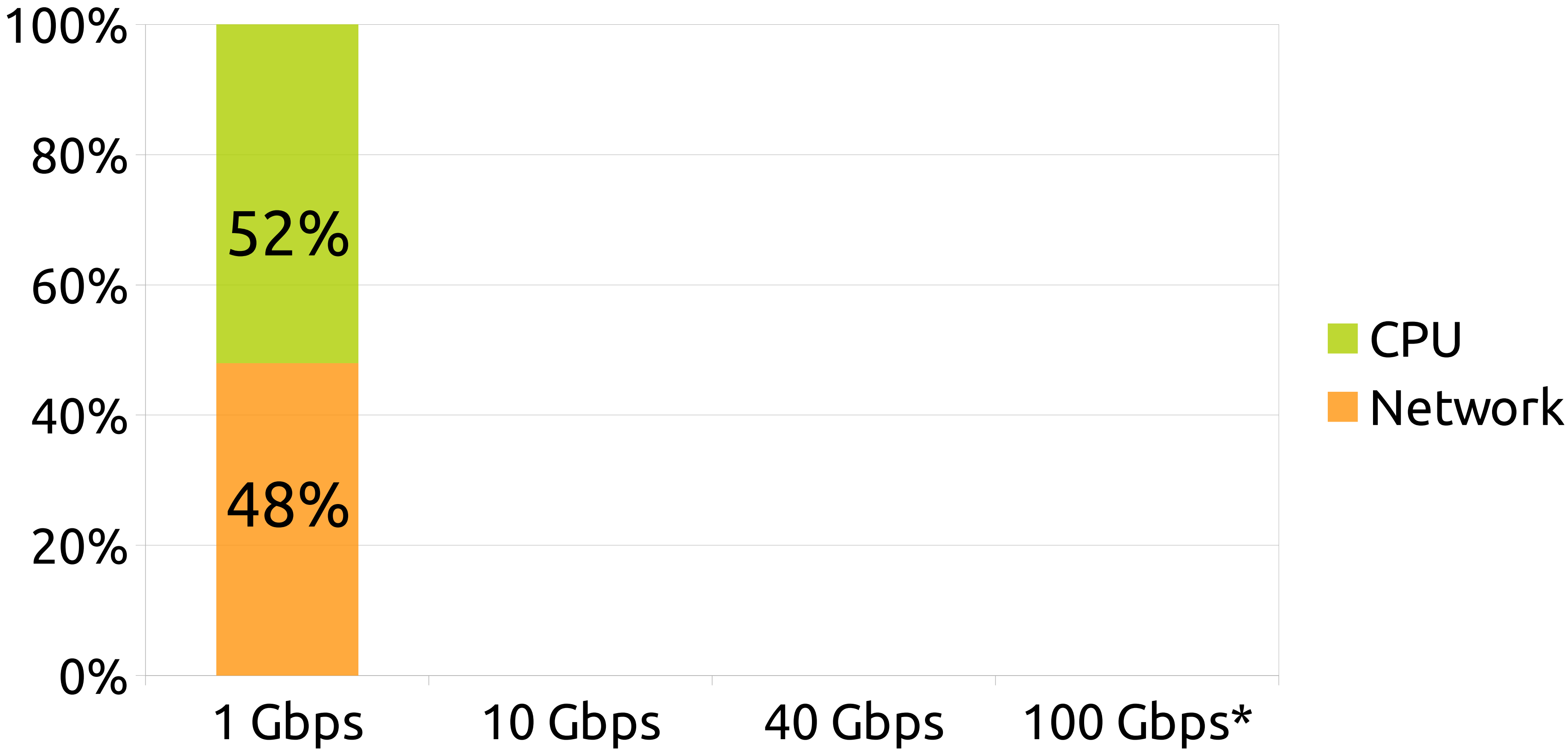


reading in shuffle data + sorting shuffle data = performance

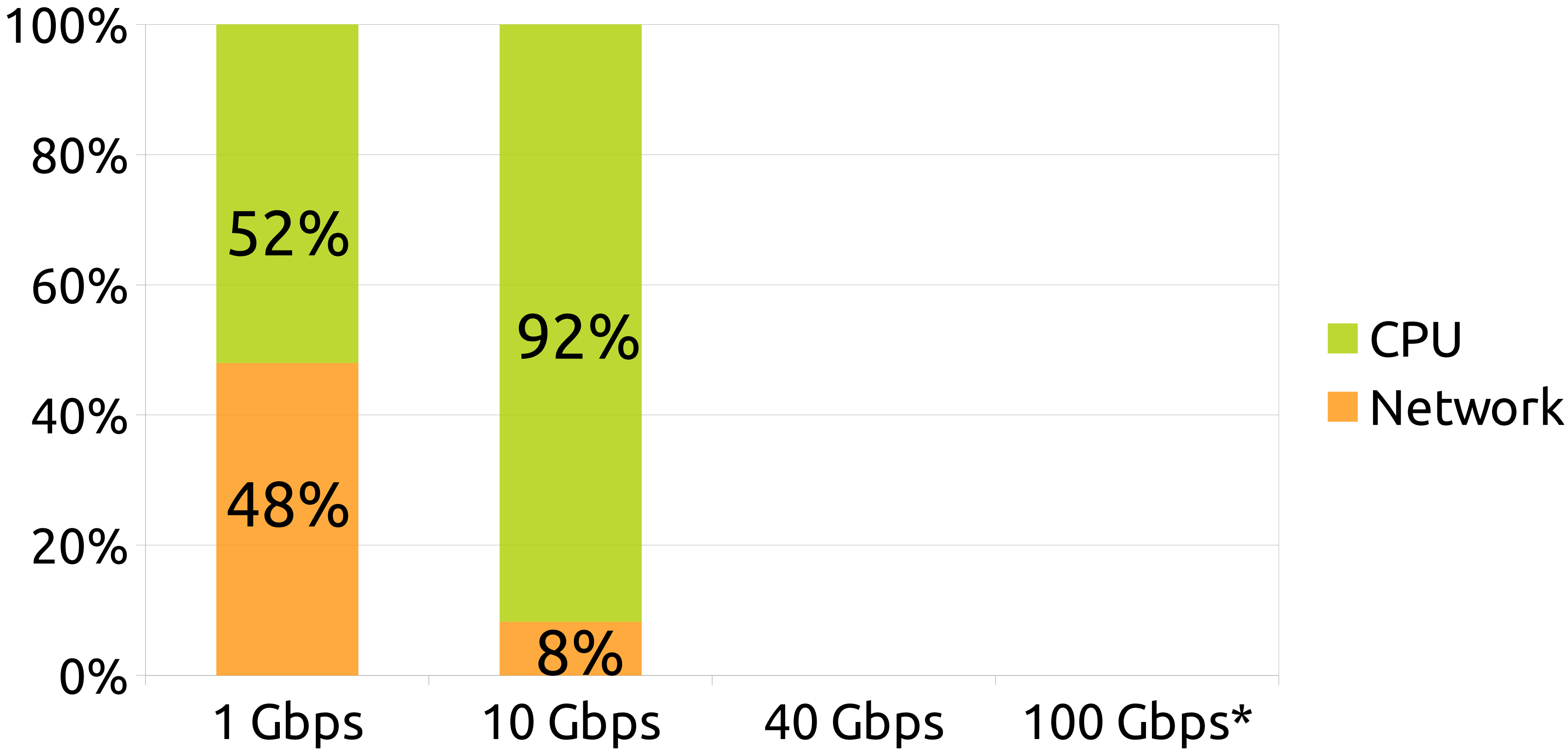
How Important is the Network?



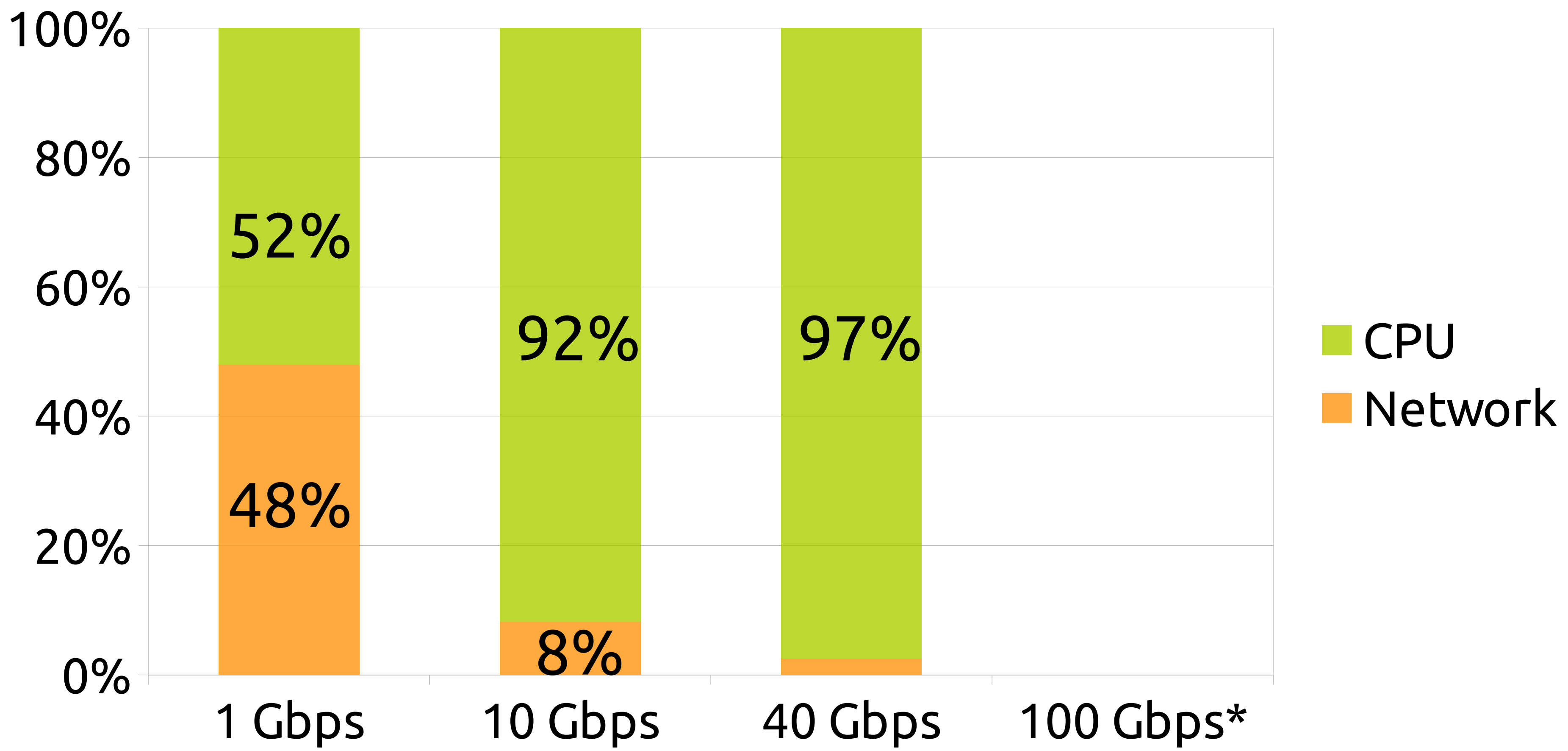
How Important is the Network?



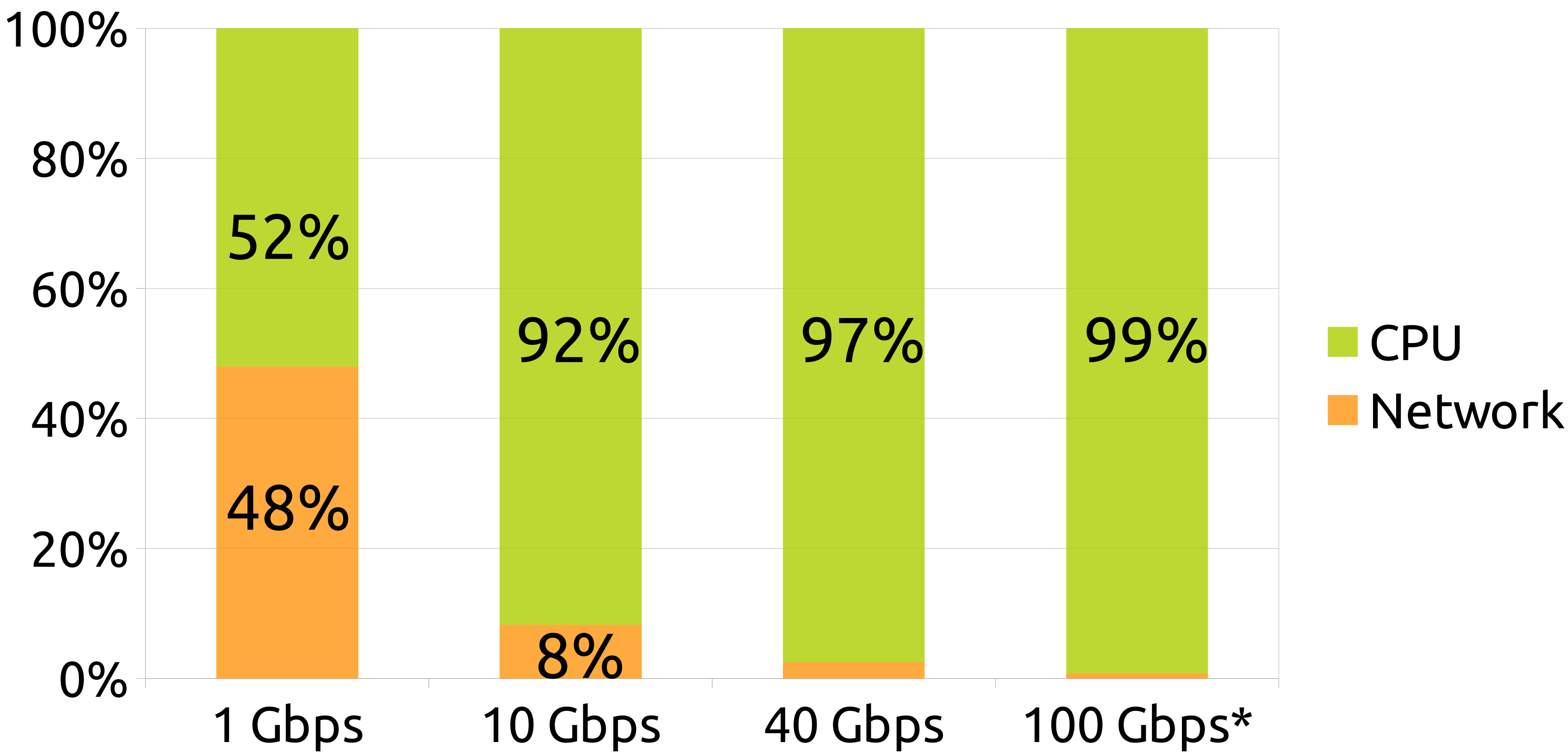
How Important is the Network?



How Important is the Network?

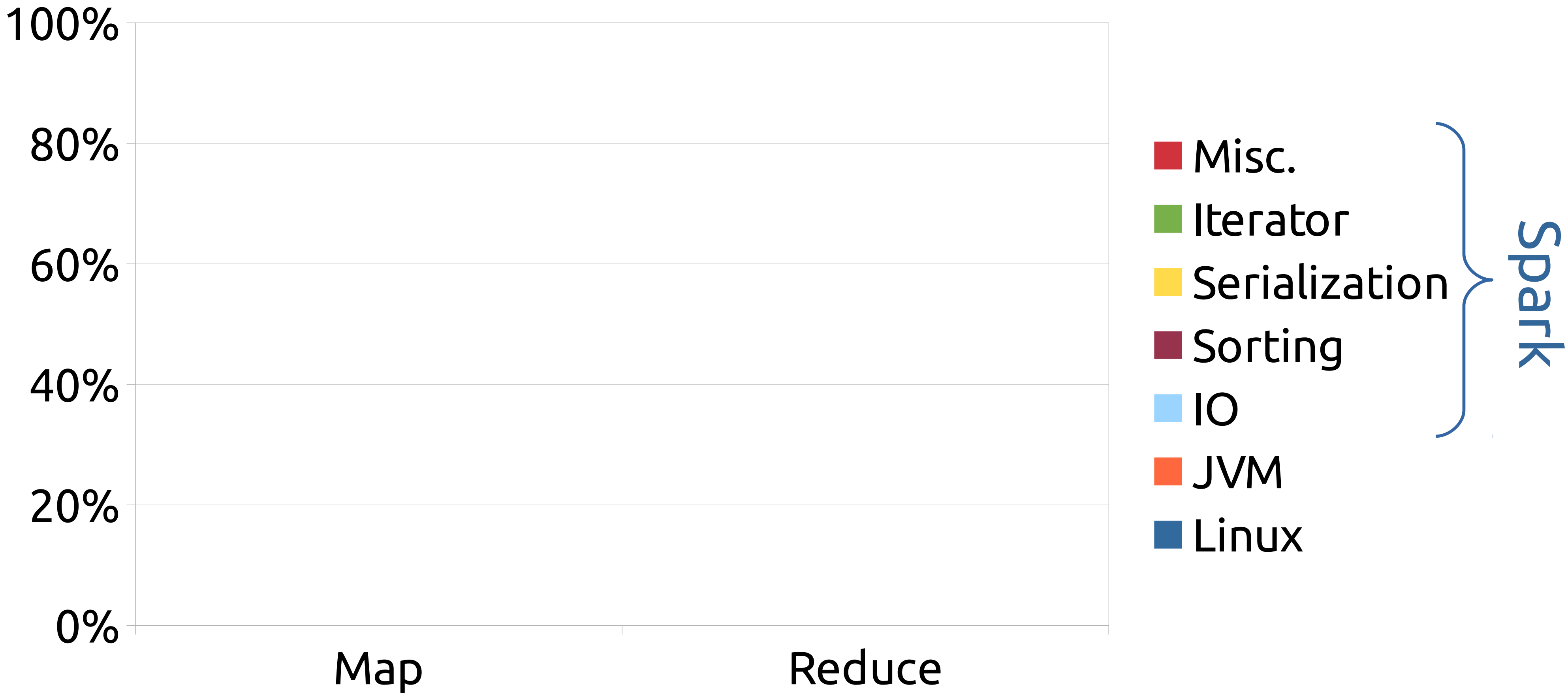


How Important is the Network?

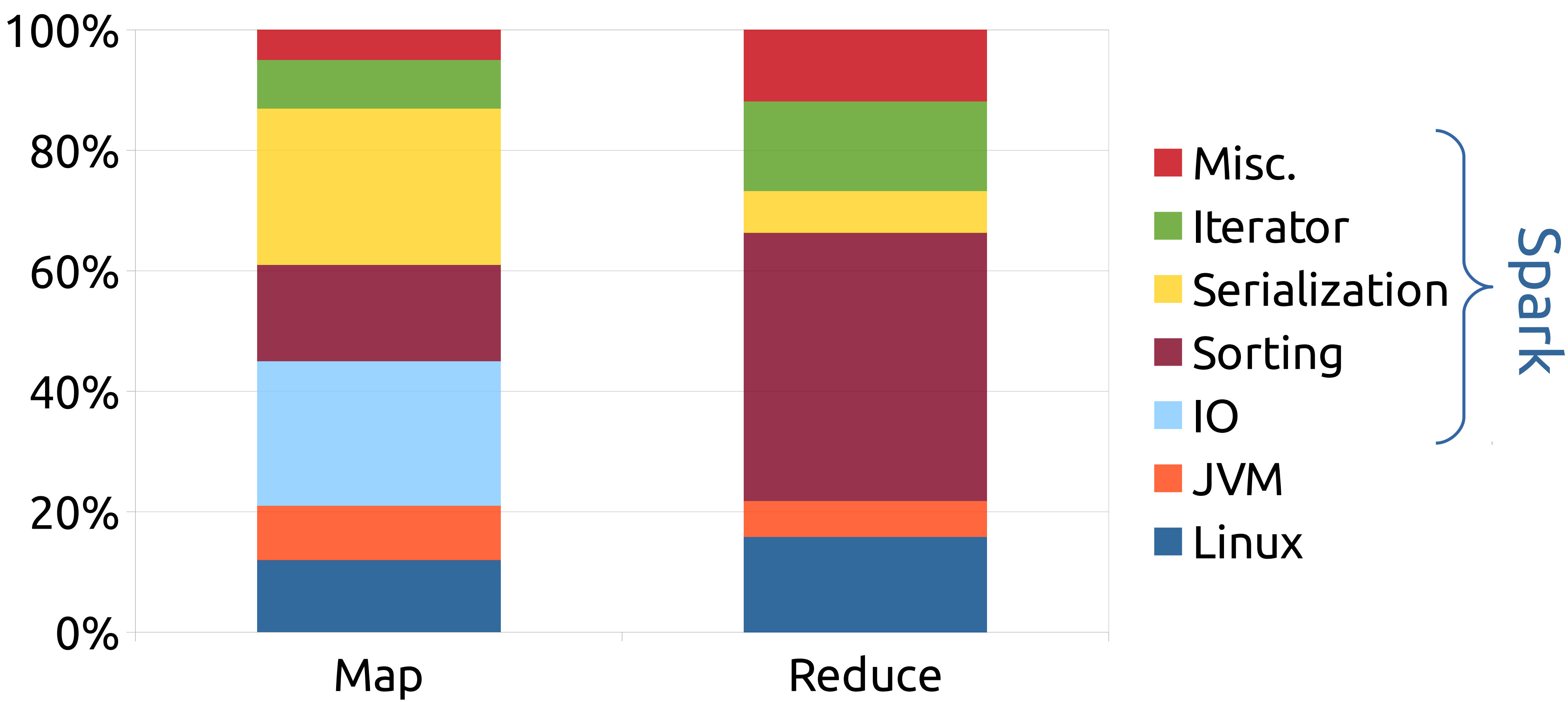


Network gains are shadowed by the CPU

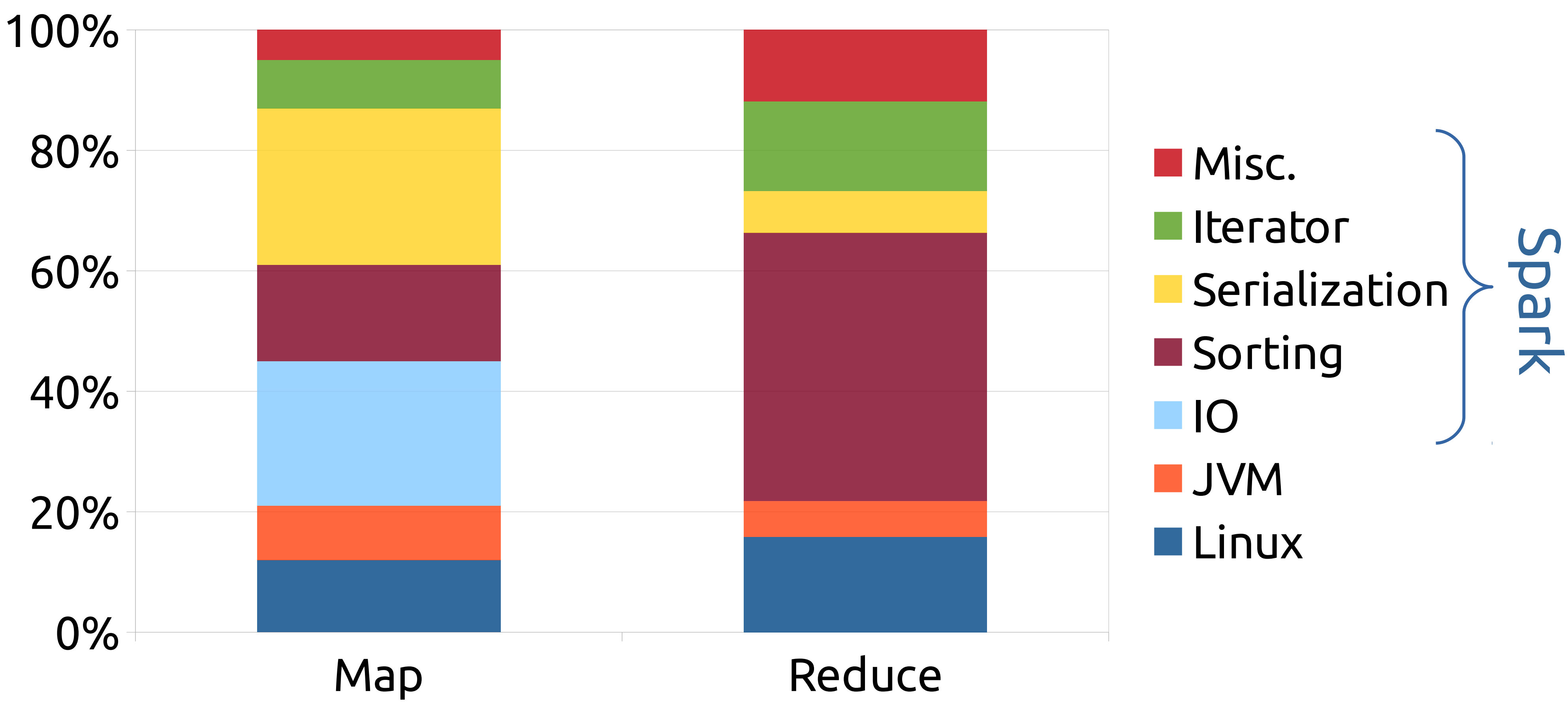
What Exactly is the CPU Doing?



What Exactly is the CPU Doing?



What Exactly is the CPU Doing?



❷ Overheads are spread across the entire stack
- serialization, abstration, execution model etc.

The Balancing Act: CPU vs Network

The Balancing Act: CPU vs Network

I.
Balance out the CPU
with the network time

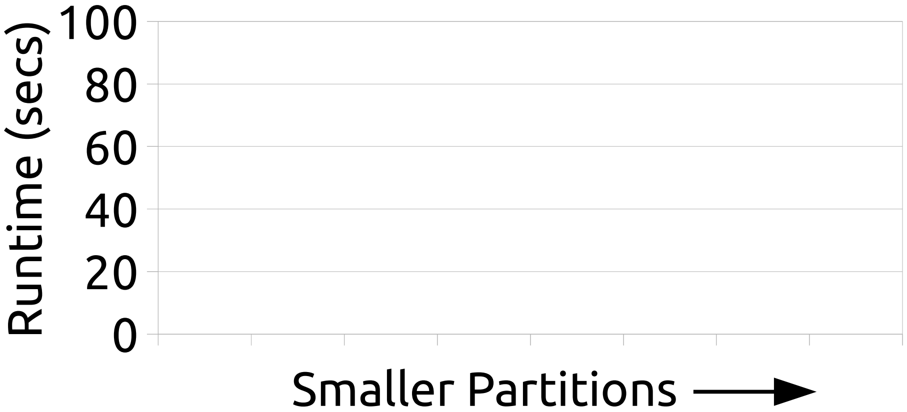
Sorting : O(nlog(n))
Network: O(n)

use smaller 'n'

The Balancing Act: CPU vs Network

I.

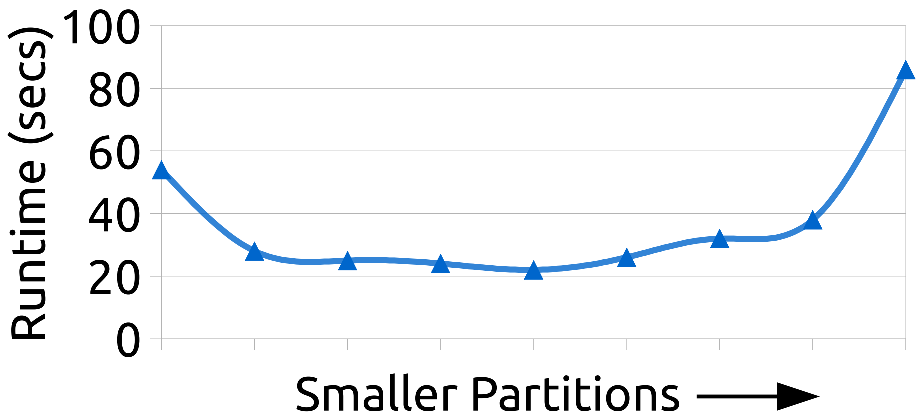
Balance out the CPU
with the network time



The Balancing Act: CPU vs Network

I.

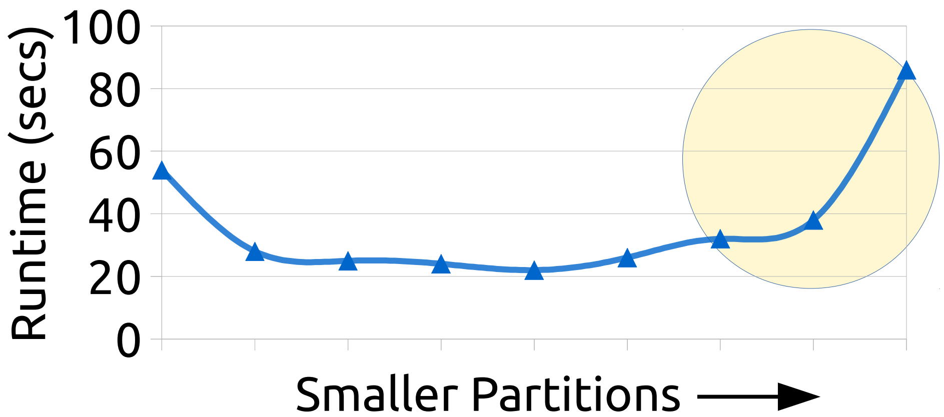
Balance out the CPU with the network time



The Balancing Act: CPU vs Network

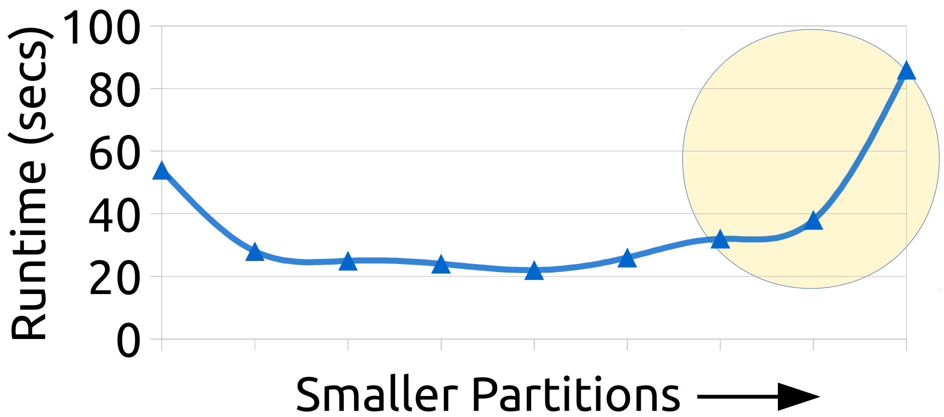
I.

Balance out the CPU
with the network time



The Balancing Act: CPU vs Network

I.
Balance out the CPU with the network time



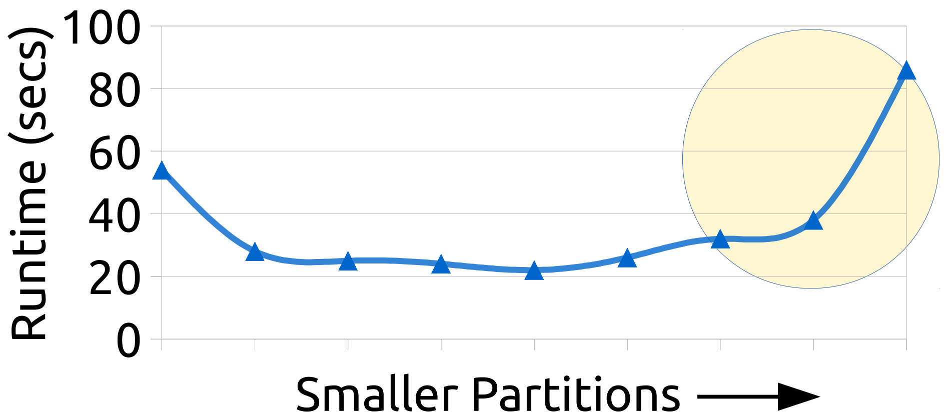
II.
Use more cores to scale up

if a single core cannot do 40 Gbps

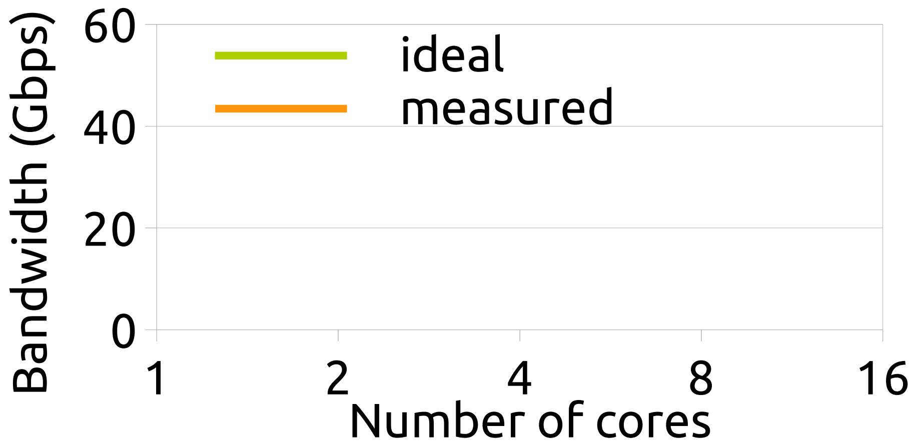
then use more

The Balancing Act: CPU vs Network

I.
Balance out the CPU with the network time



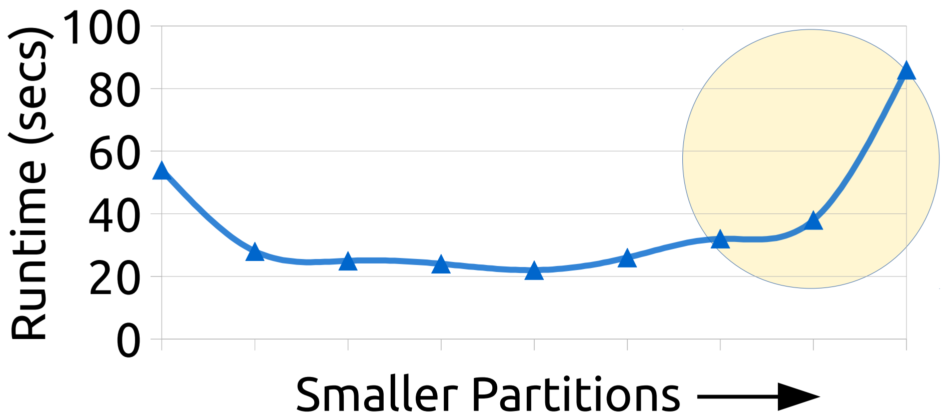
II.
Use more cores to scale up



The Balancing Act: CPU vs Network

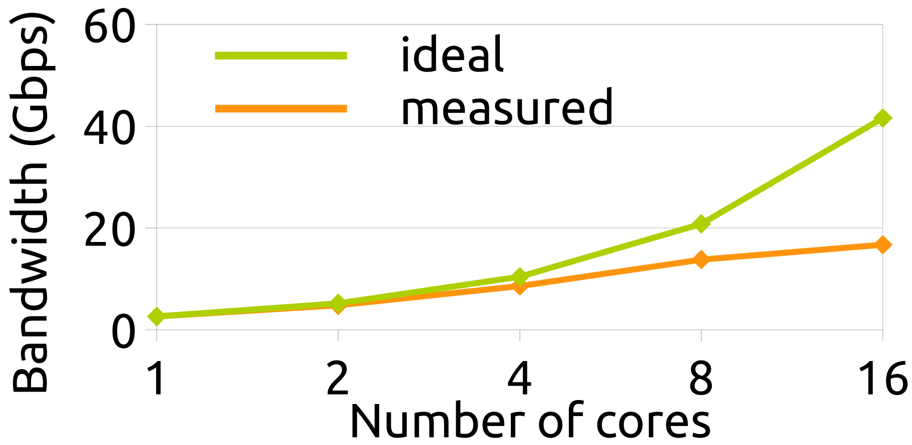
I.

Balance out the CPU with the network time



II.

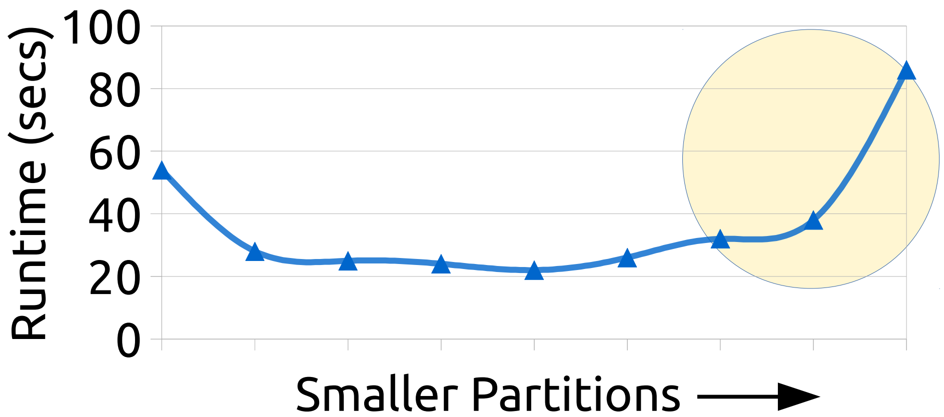
Use more cores to scale up



The Balancing Act: CPU vs Network

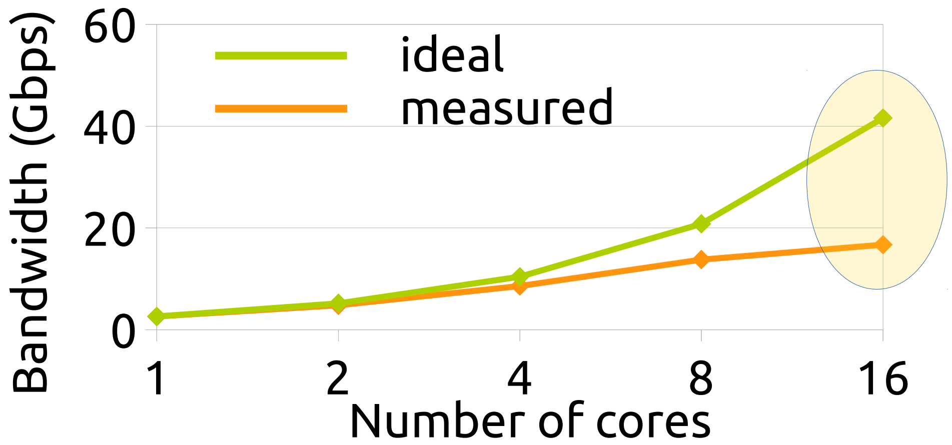
I.

Balance out the CPU with the network time



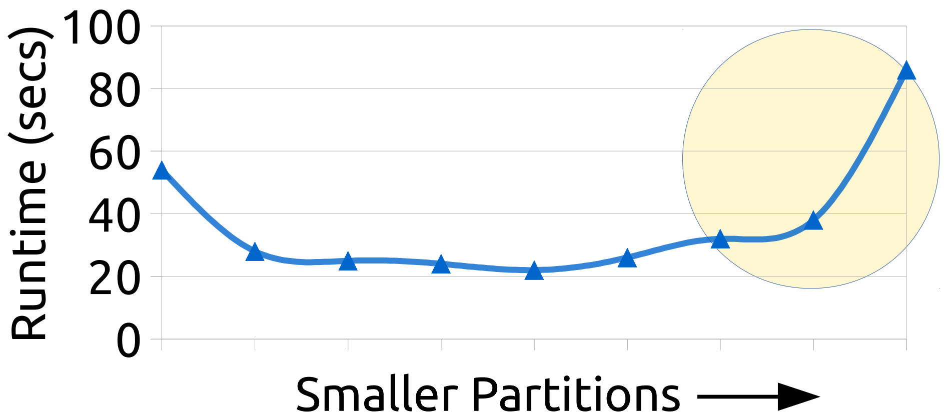
II.

Use more cores to scale up

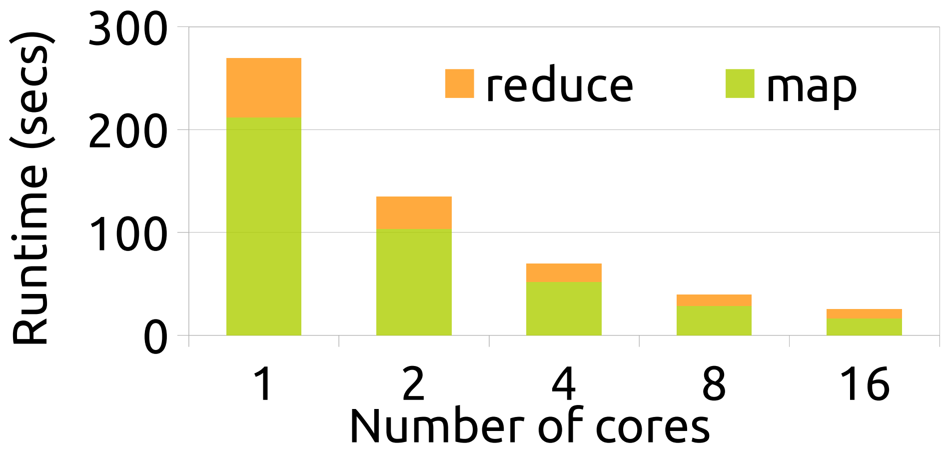


The Balancing Act: CPU vs Network

I.
Balance out the CPU with the network time

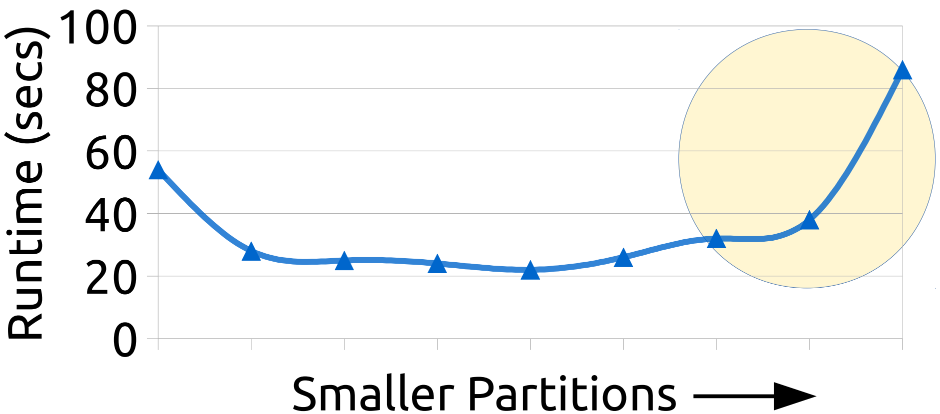


II.
Use more cores to scale up

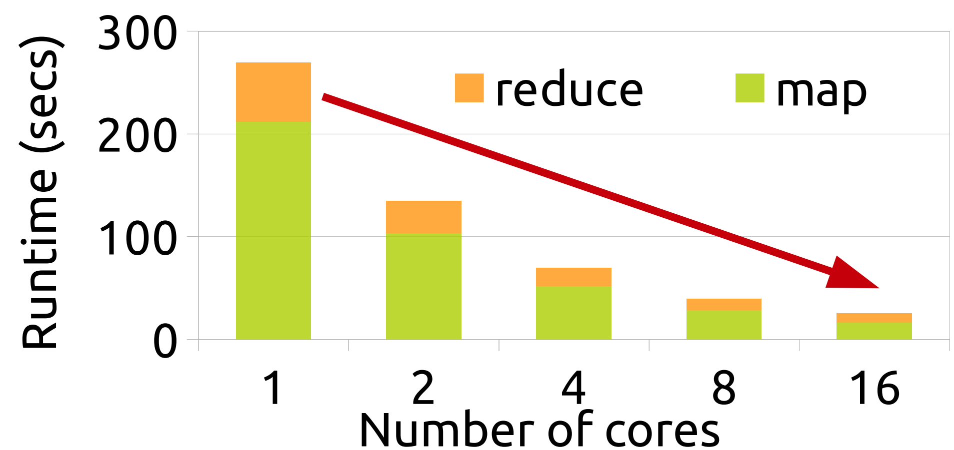


The Balancing Act: CPU vs Network

I.
Balance out the CPU with the network time



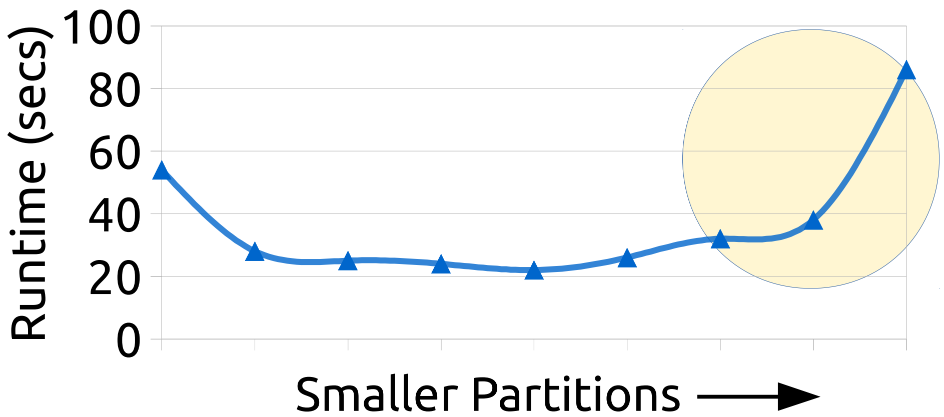
II.
Use more cores to scale up



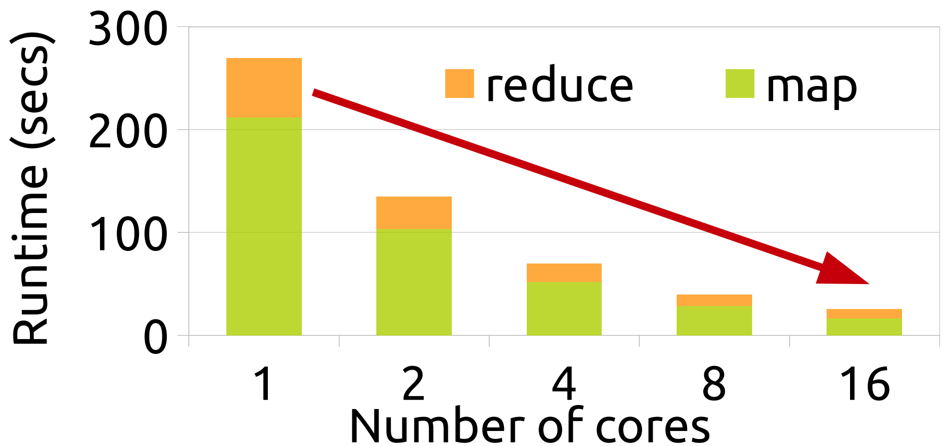
$$\text{runtime} = 9 + \frac{260}{\text{cores}}$$

The Balancing Act: CPU vs Network

I.
Balance out the CPU with the network time



II.
Use more cores to scale up



3 Classical techniques are ineffective

Conclusion

1 Faster networks (IO) are very relevant

- as long as you have CPU cycles
- differentiate between user vs framework CPU usage

Conclusion

1. Faster networks (IO) are very relevant
   - as long as you have CPU cycles
   - differentiate between user vs framework CPU usage

2. Framework's CPU usage is bad
   - CPU-network imbalance : sorting, serialization, volcano execution model, etc.
   - scalability (serial vs parallel components)
   - ineffective classical balancing techniques

Conclusion

1. Faster networks (IO) are very relevant
   - as long as you have CPU cycles
   - differentiate between user vs framework CPU usage
2. Framework's CPU usage is bad
   - CPU-network imbalance : sorting, serialization, volcano execution model, etc.
   - scalability (serial vs parallel components)
   - ineffective classical balancing techniques
3. Knowing today's *usec-era* IO and CPU hardware, how would you re-design modern data processing framework?