Optimizing Network Performance in Distributed Machine Learning

Luo Mai

Imperial College London

Chuntao Hong

Microsoft Research

Paolo Costa

Microsoft Research

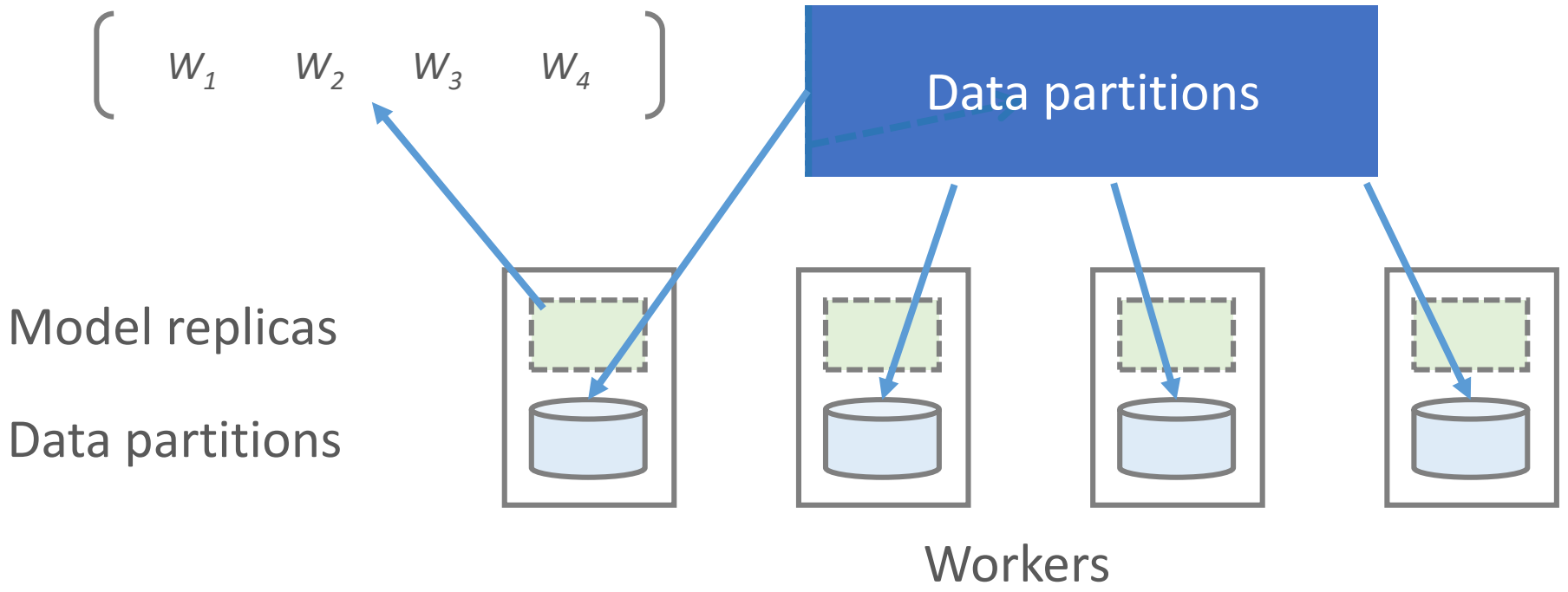
Machine Learning

- Successful in many fields
  - Online advertisement
  - Spam filtering
  - Fraud detection
  - Image recognition
  - …
- One of the most important workloads in data centers

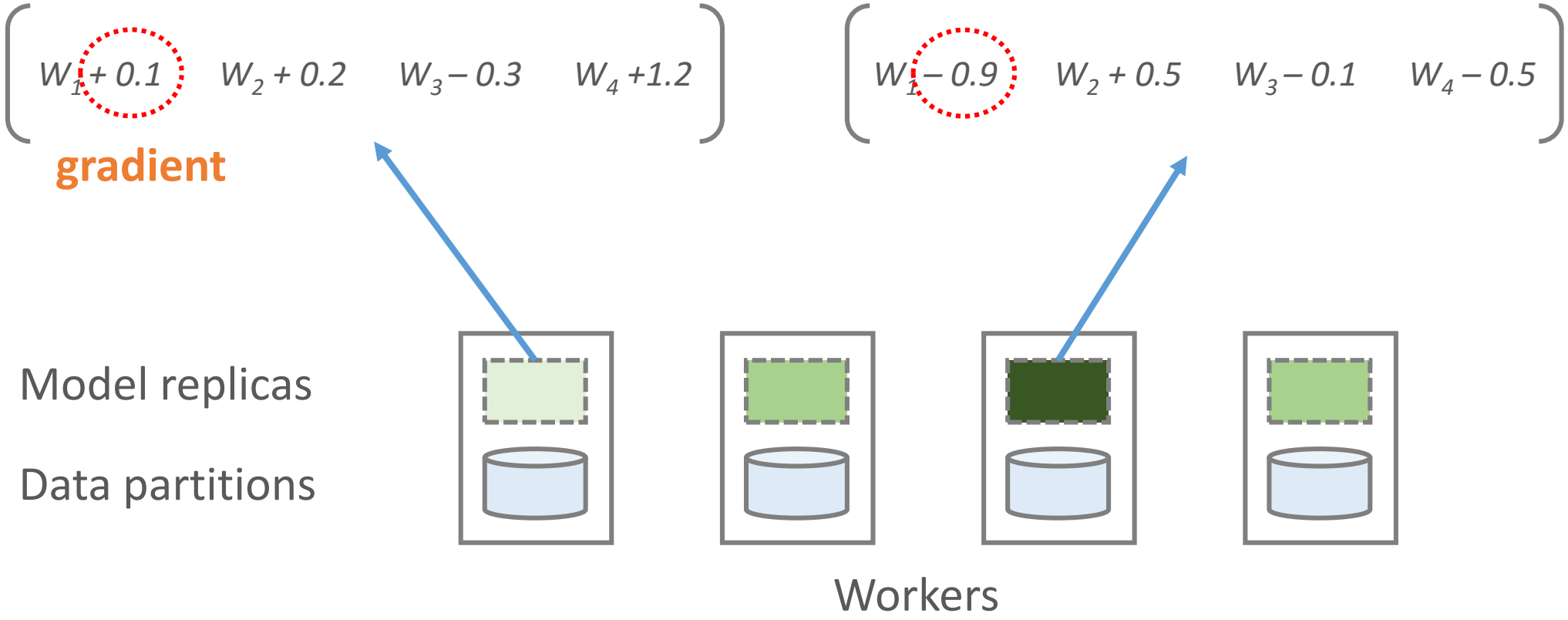
Industry Scale Machine Learning

- More data, higher accuracy
- Scales of industry problems
  - 100 Billions samples, 1TBs – 1PBs data
  - 10 Billions parameters, 1GBs – 1TBs data
- Distributed execution
  - 100s – 1000s machines

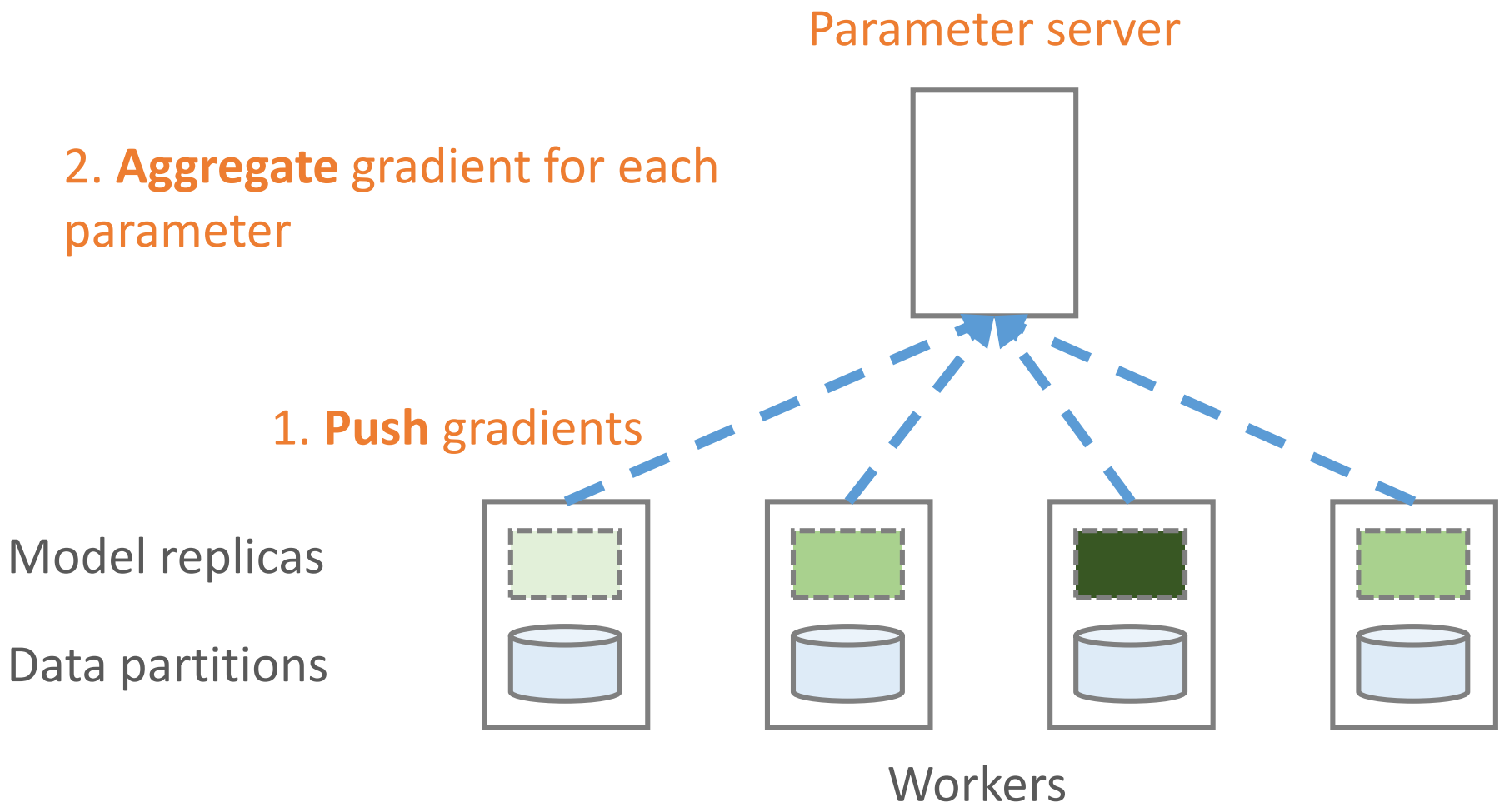
Distributed Machine Learning



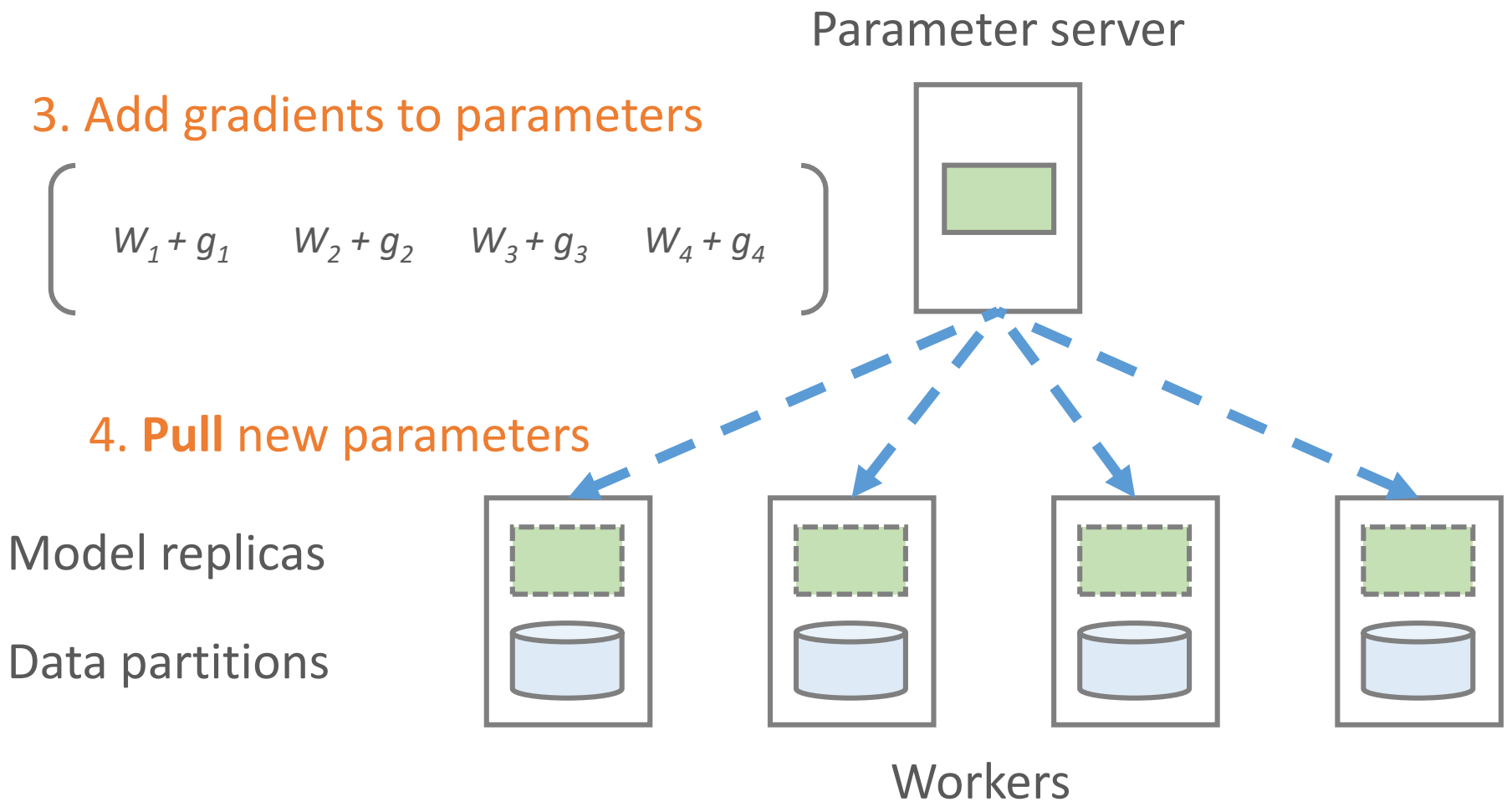
Distributed Machine Learning



Distributed Machine Learning

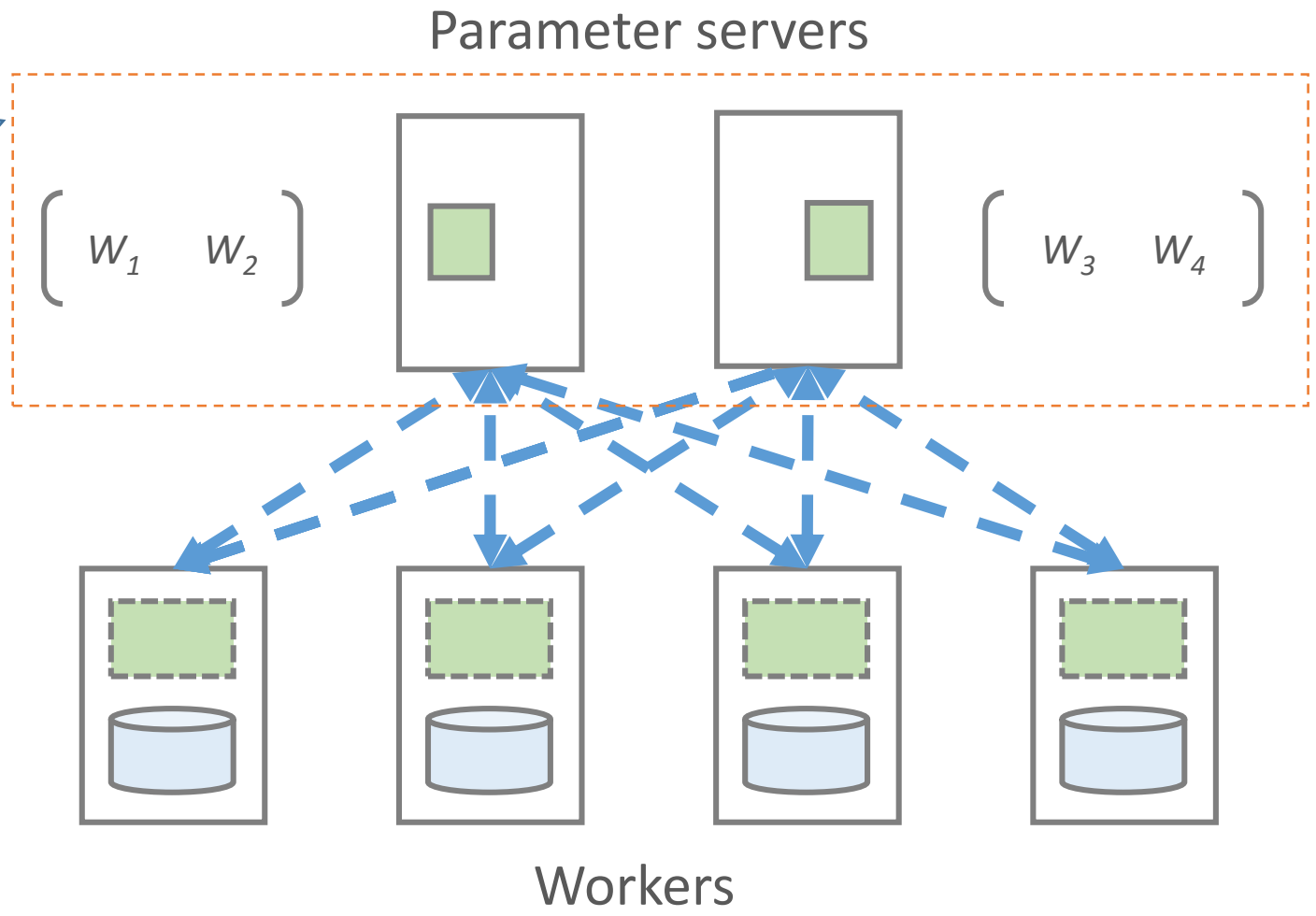


Distributed Machine Learning

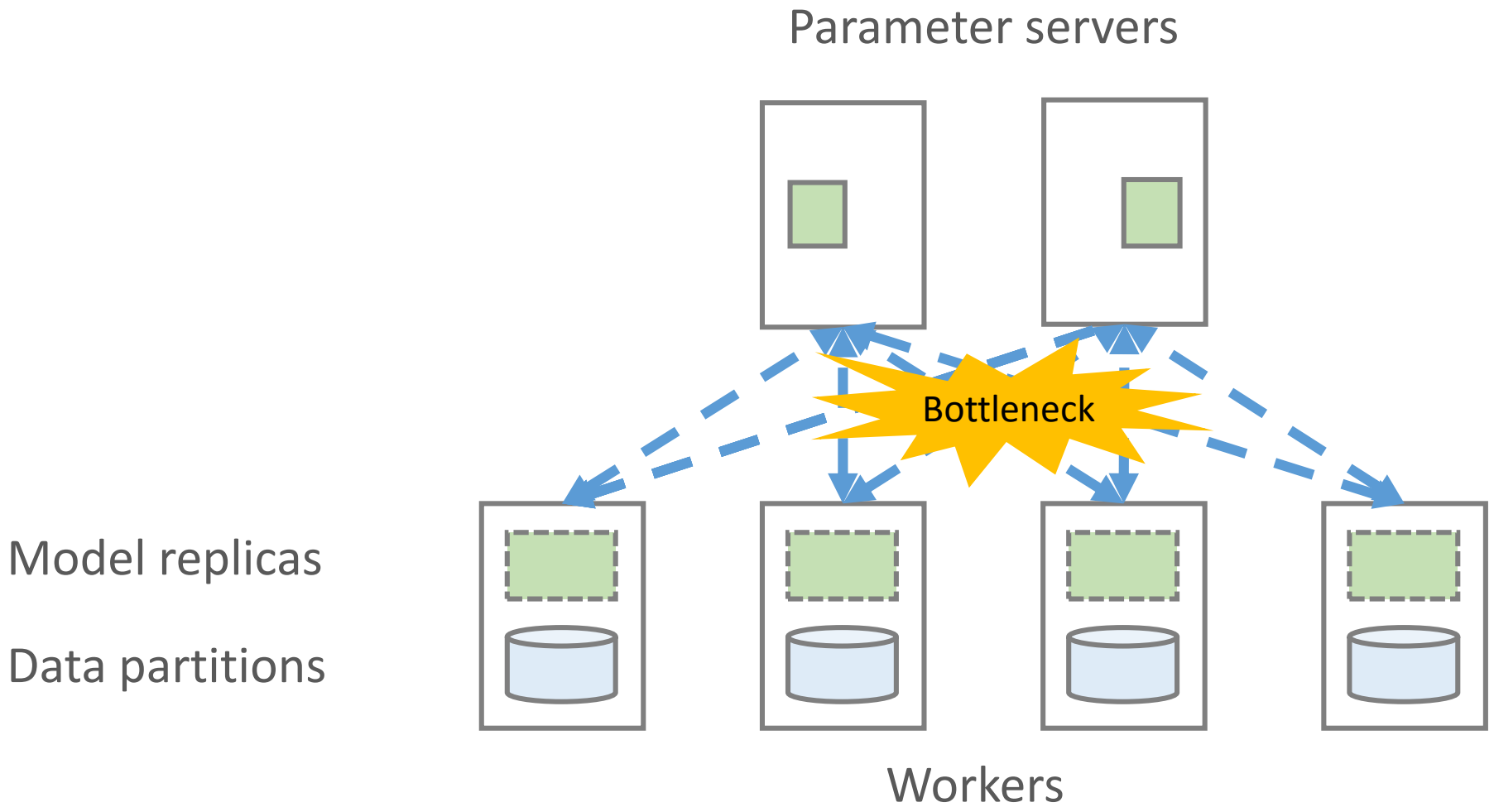


Distributed Machine Learning

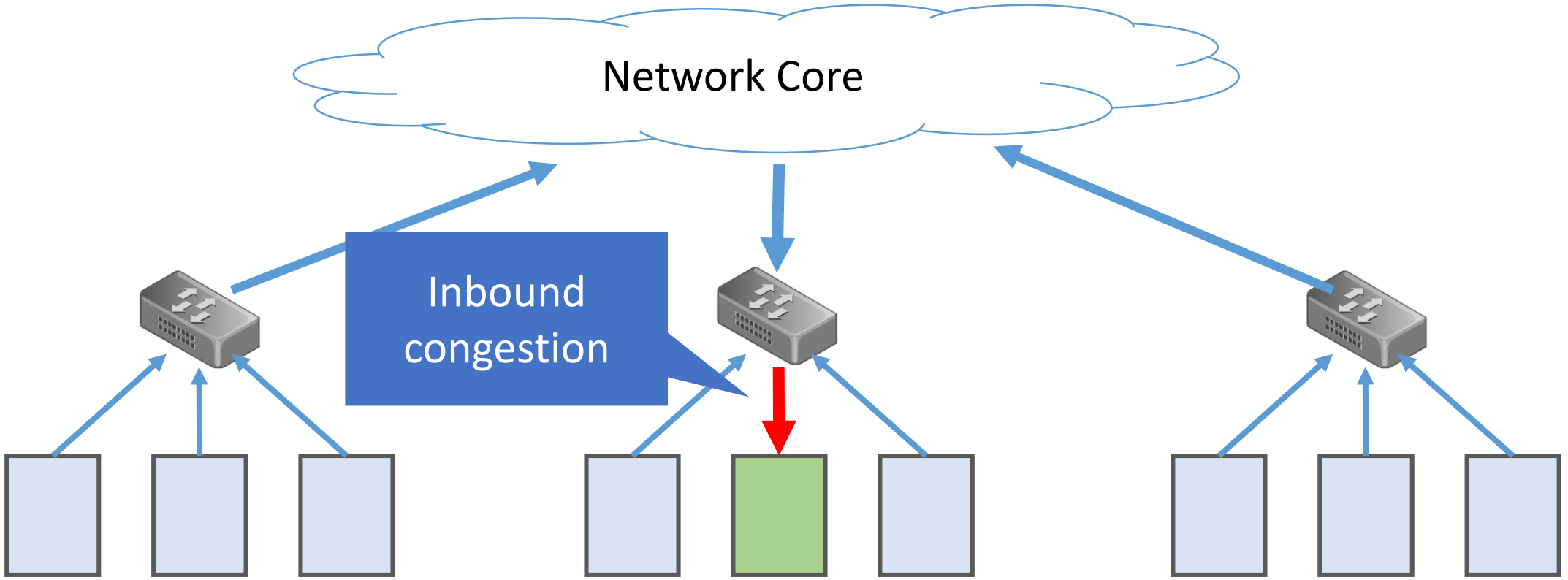
Use multiple PS to avoid bottleneck



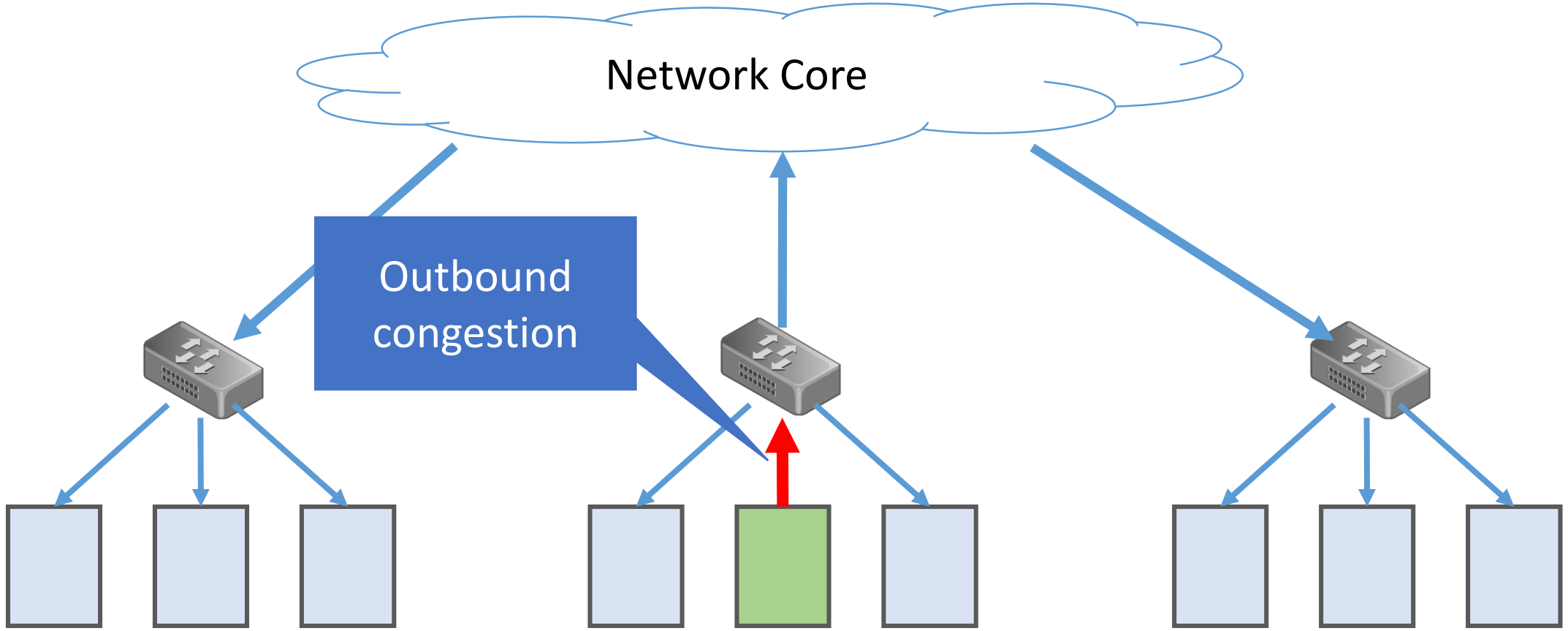
Distributed Machine Learning



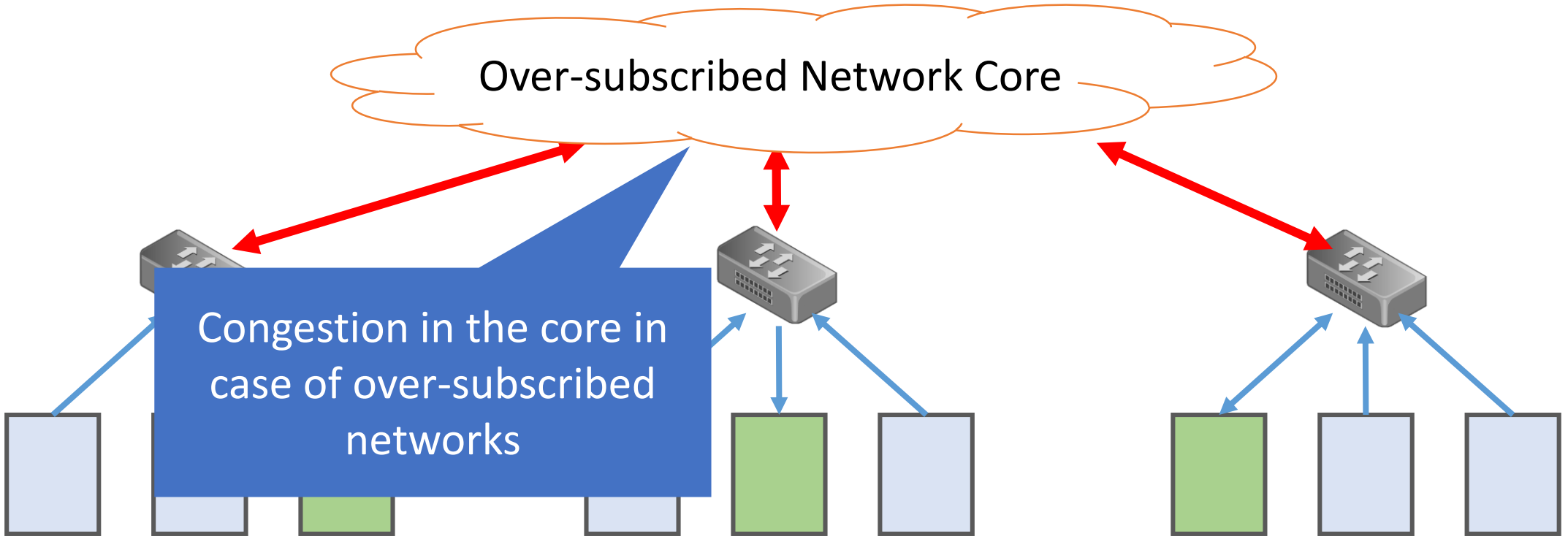
Inbound Congestion



Outbound Congestion

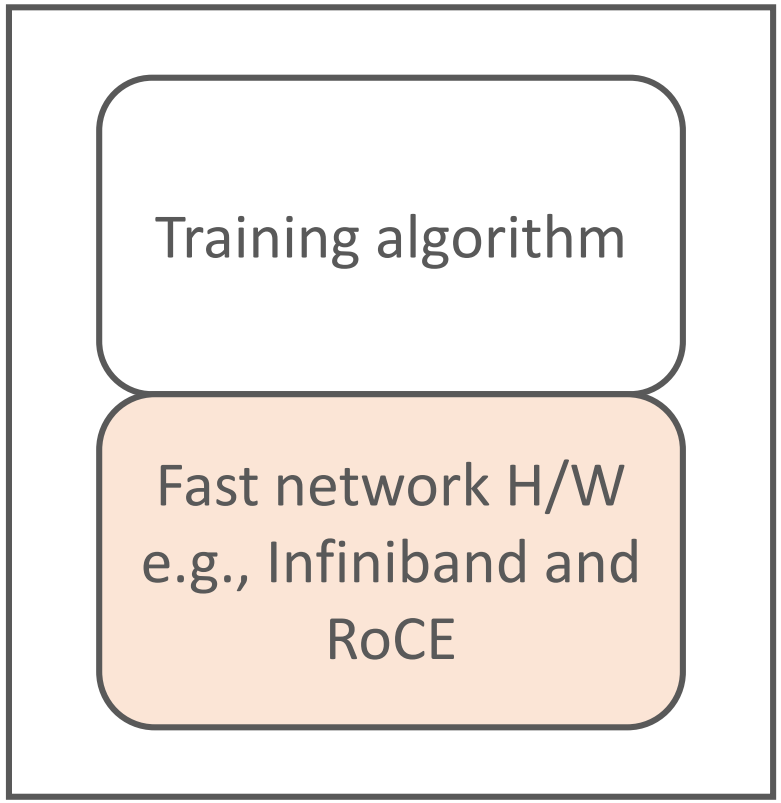


Network Core Congestion



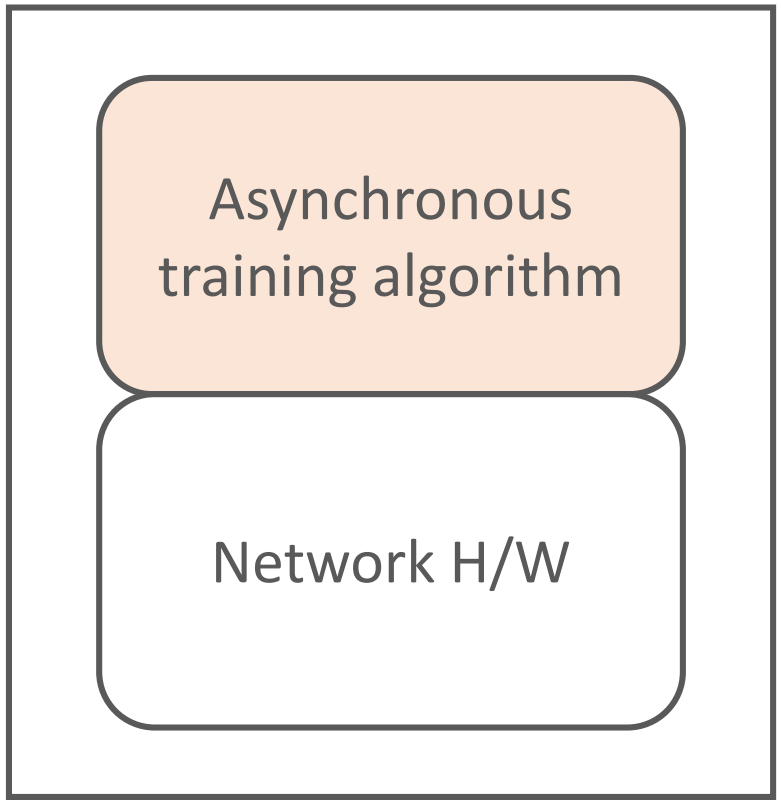
Existing Approaches

- Over-provisioning network
  - ☹ Expensive
  - ☹ Limited deployment scale
  - ☹ Not available in public clouds



Existing Approaches

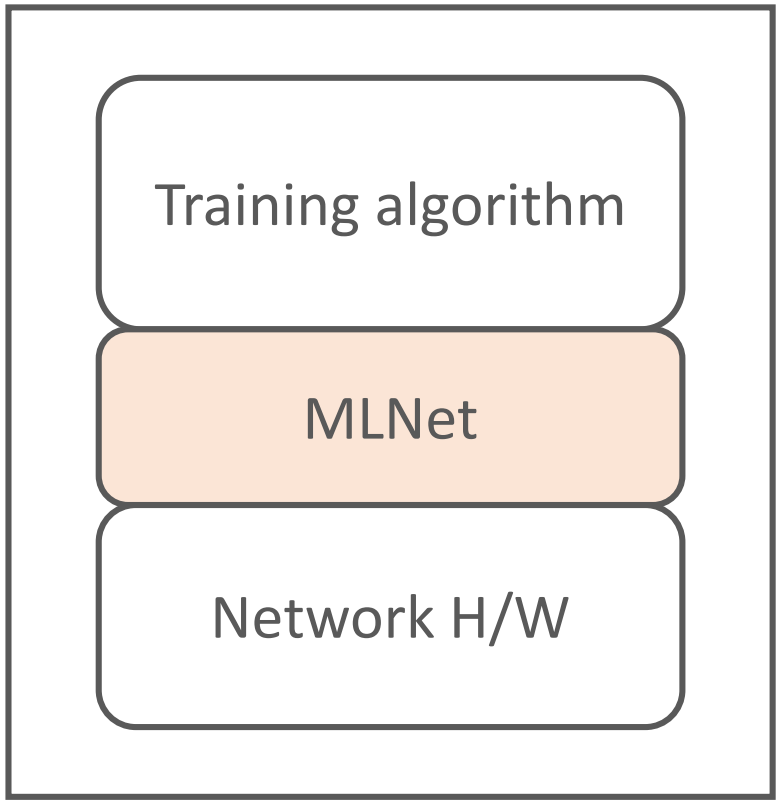
- Over-provisioning network
  - ☹ Expensive
  - ☹ Limited deployment scale
  - ☹ Not available in public Clouds
- Asynchronous training algorithm
  - ☹ Training efficiency
  - ☹ Might not converge



Rethinking the Network Design

MLNet is a communication layer designed for distributed machine learning systems

- ✓Improves communication efficiency
- ✓Orthogonal to existing approaches



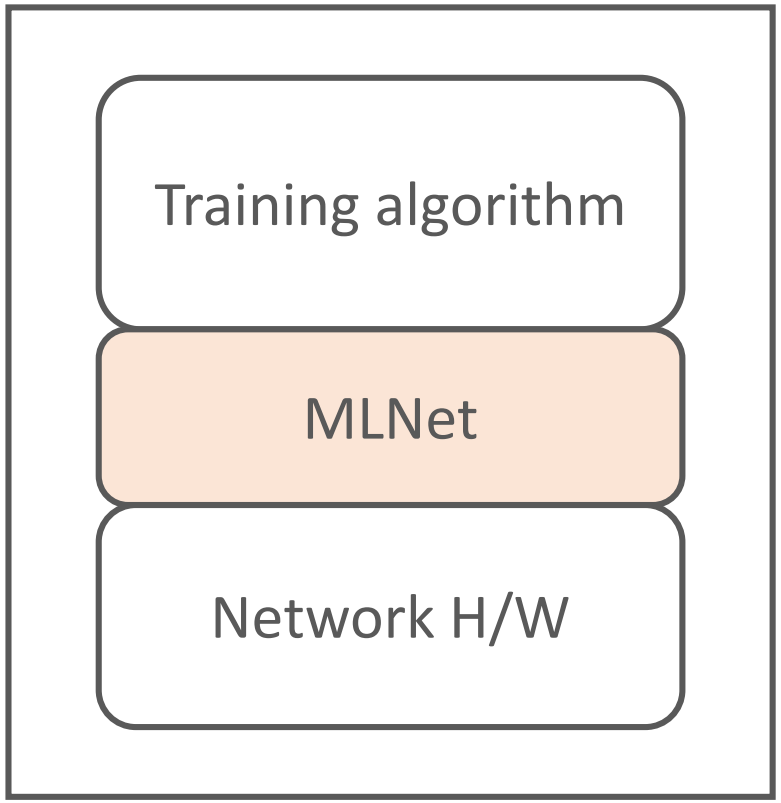
Rethinking the Network Design

MLNet is a communication layer designed for distributed machine learning systems

- ✓Improves communication efficiency
- ✓Orthogonal to existing approaches

Optimizations:

- ✓Traffic reduction
- ✓Flow prioritization



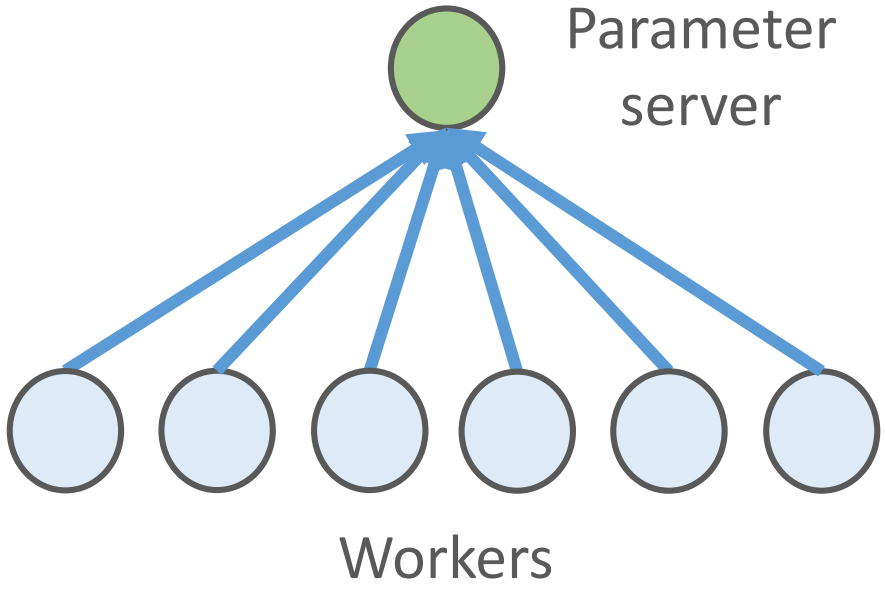
Traffic Reduction

Traffic Reduction: Key Insight

Aggregate the gradients from 6 workers

$$g_1 = g_{11} + g_{12} + g_{13} + g_{14} + g_{15} + g_{16}$$

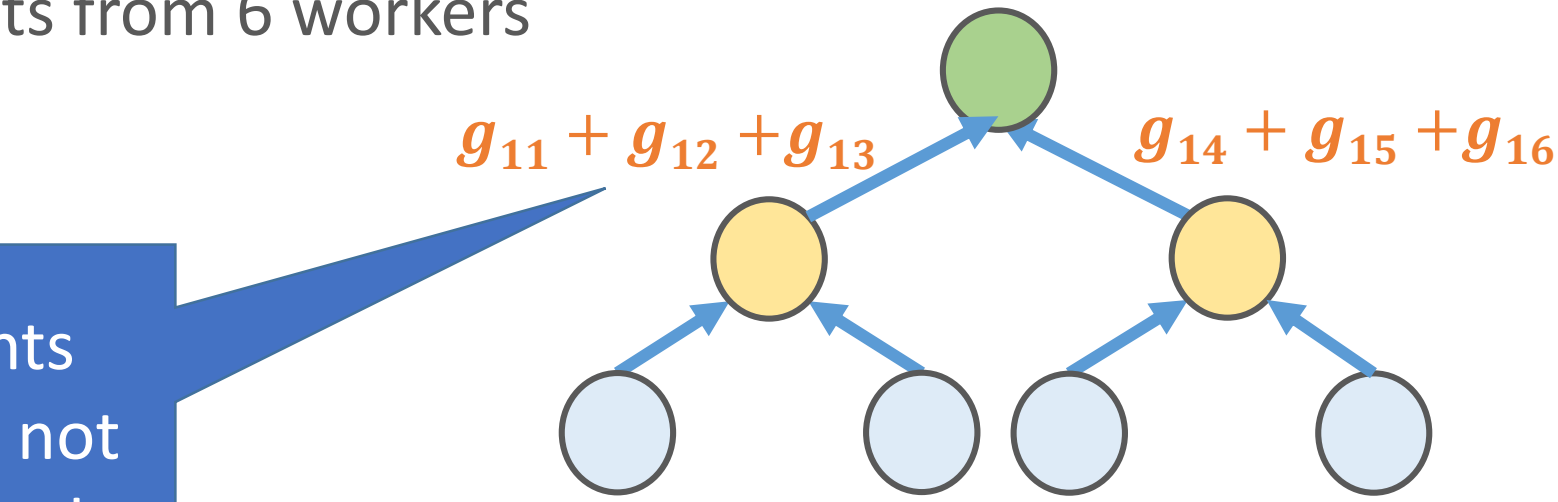
Aggregation is commutative and associative



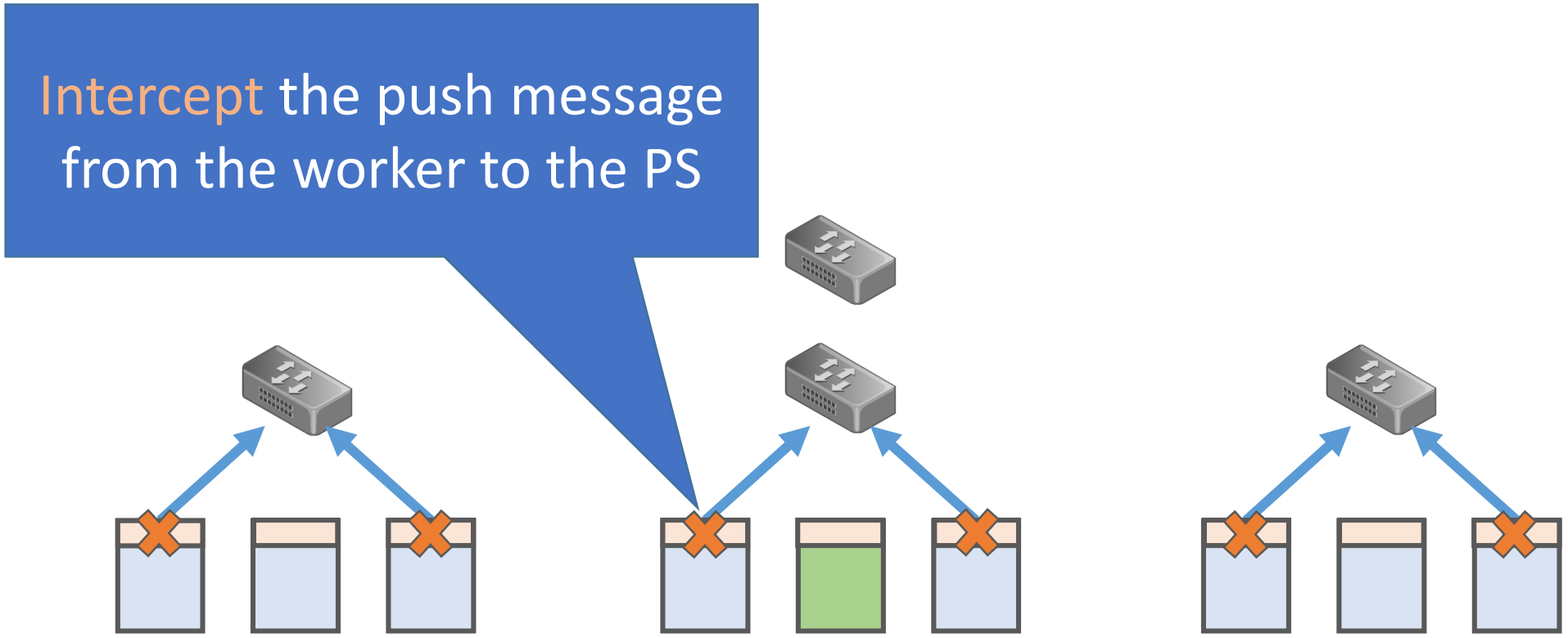
Traffic Reduction: Key Insight

Aggregate the gradients from 6 workers

Aggregate gradients incrementally does not change the final result

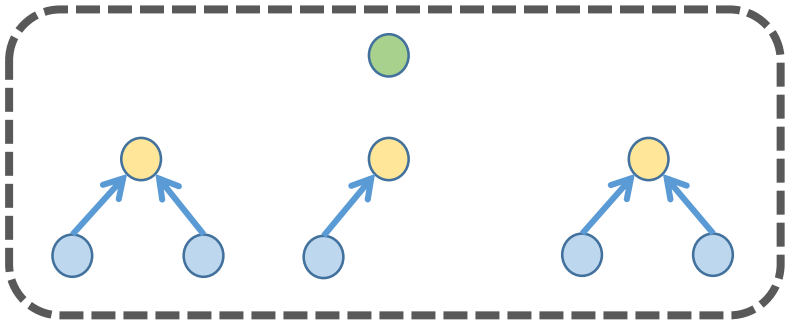


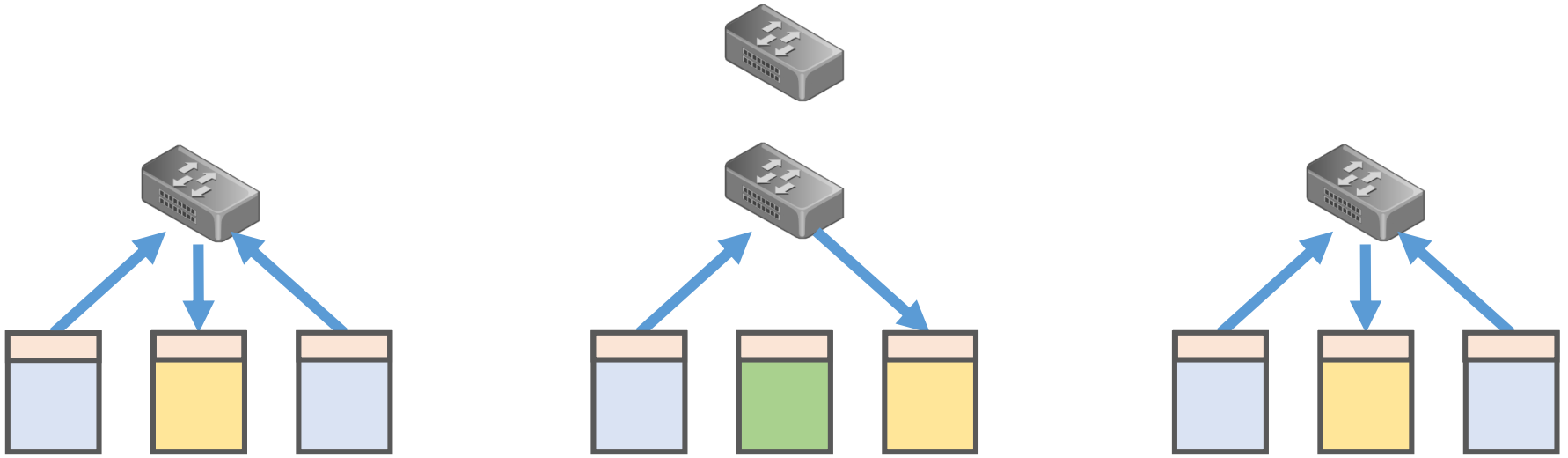
Traffic Reduction: Design



Traffic Reduction: Design

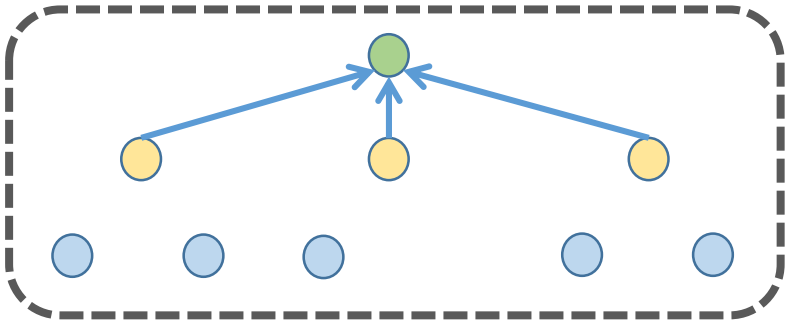
Redirect the messages to a local worker for partial aggregation

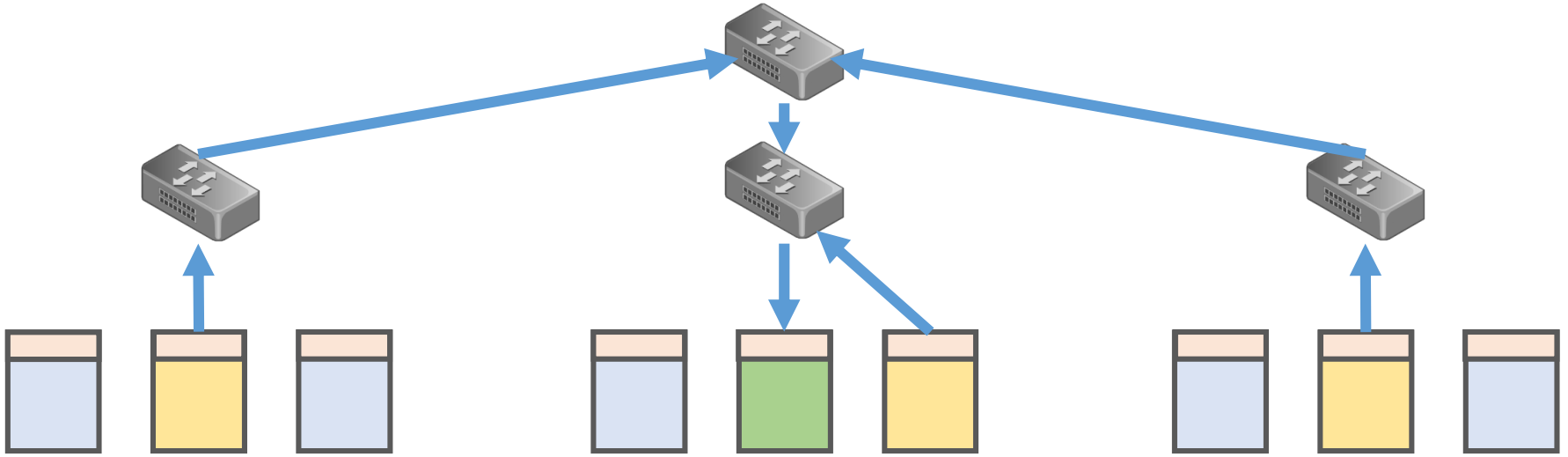




Traffic Reduction: Design

Send the partial results to the PS for final aggregation



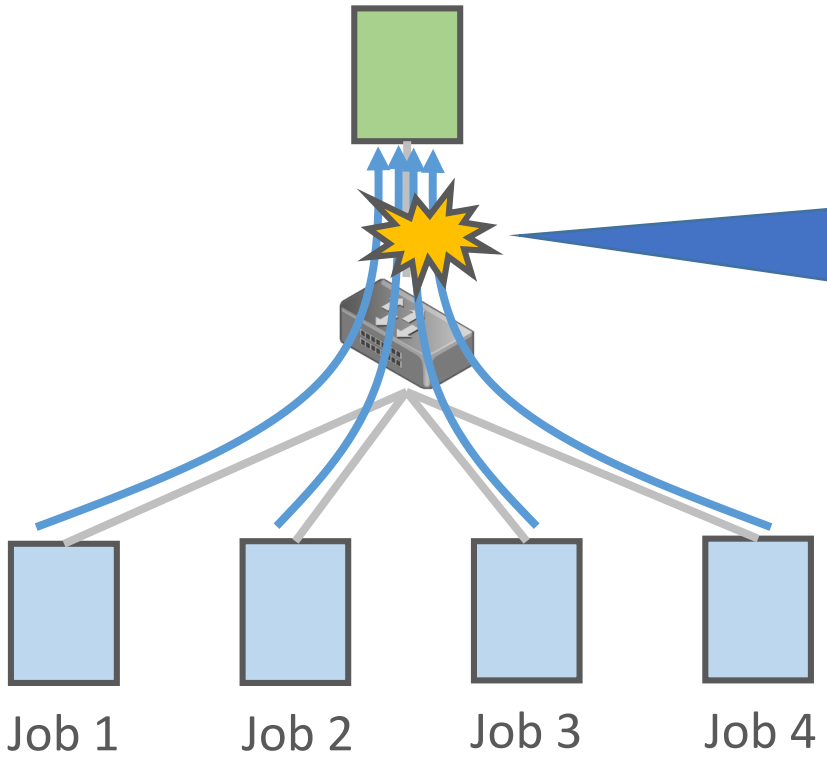


More details on the paper:

1. Traffic reduction in pull request
2. Asynchronous communication

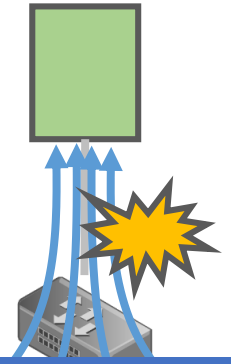
Traffic Prioritization

Traffic Prioritization: Key Insight

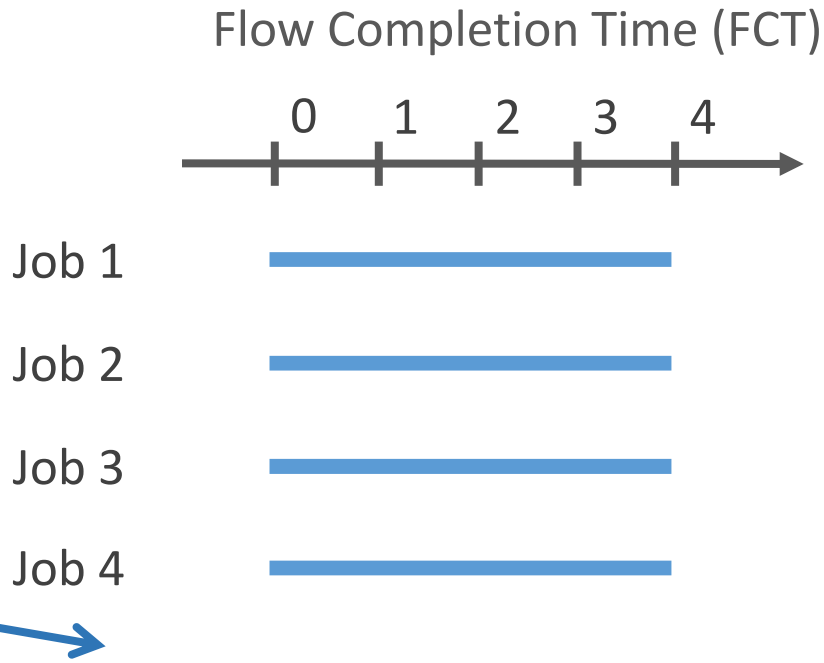


These four TCP flows share a bottleneck link and each of them gets 25% of its bandwidth

Traffic Prioritization: Key Insight

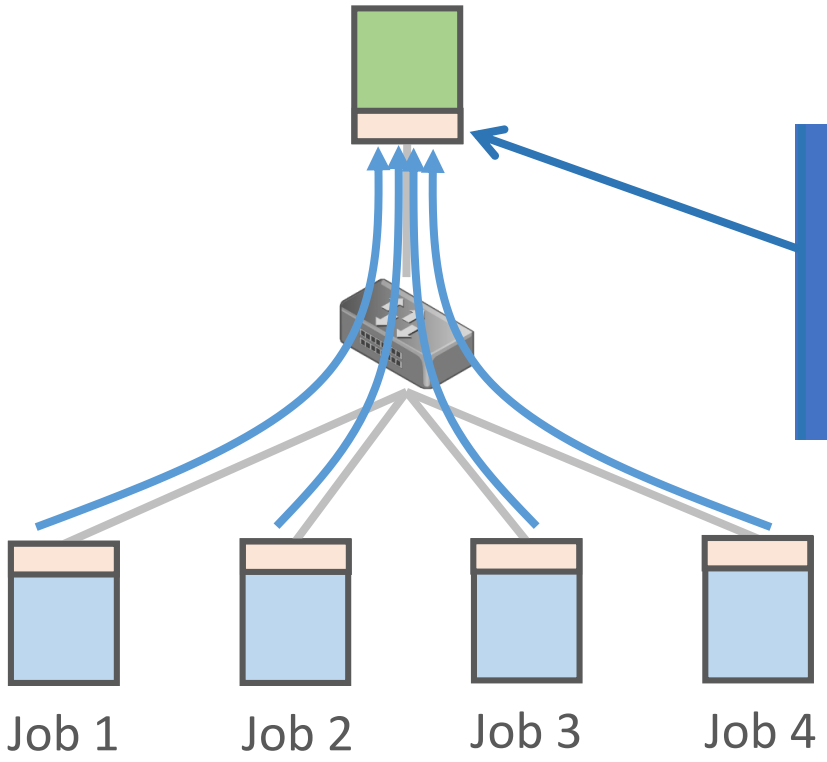


All flows are delayed! TCP per-flow fairness is harmful in distributed machine learning.



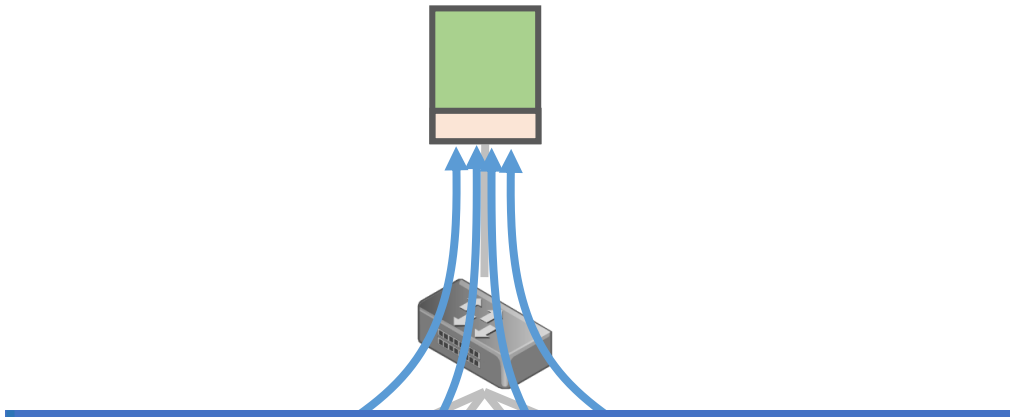
Average completion time is 4

Traffic Prioritization: Key Insight

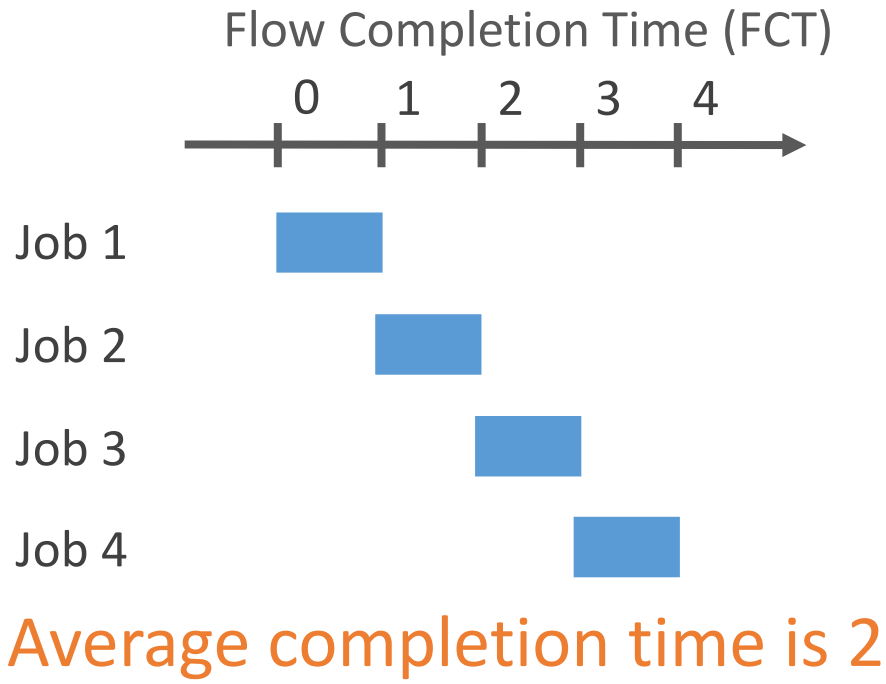


MLNet prioritizes the competing flows to minimize the average training time

Traffic Prioritization: Key Insight



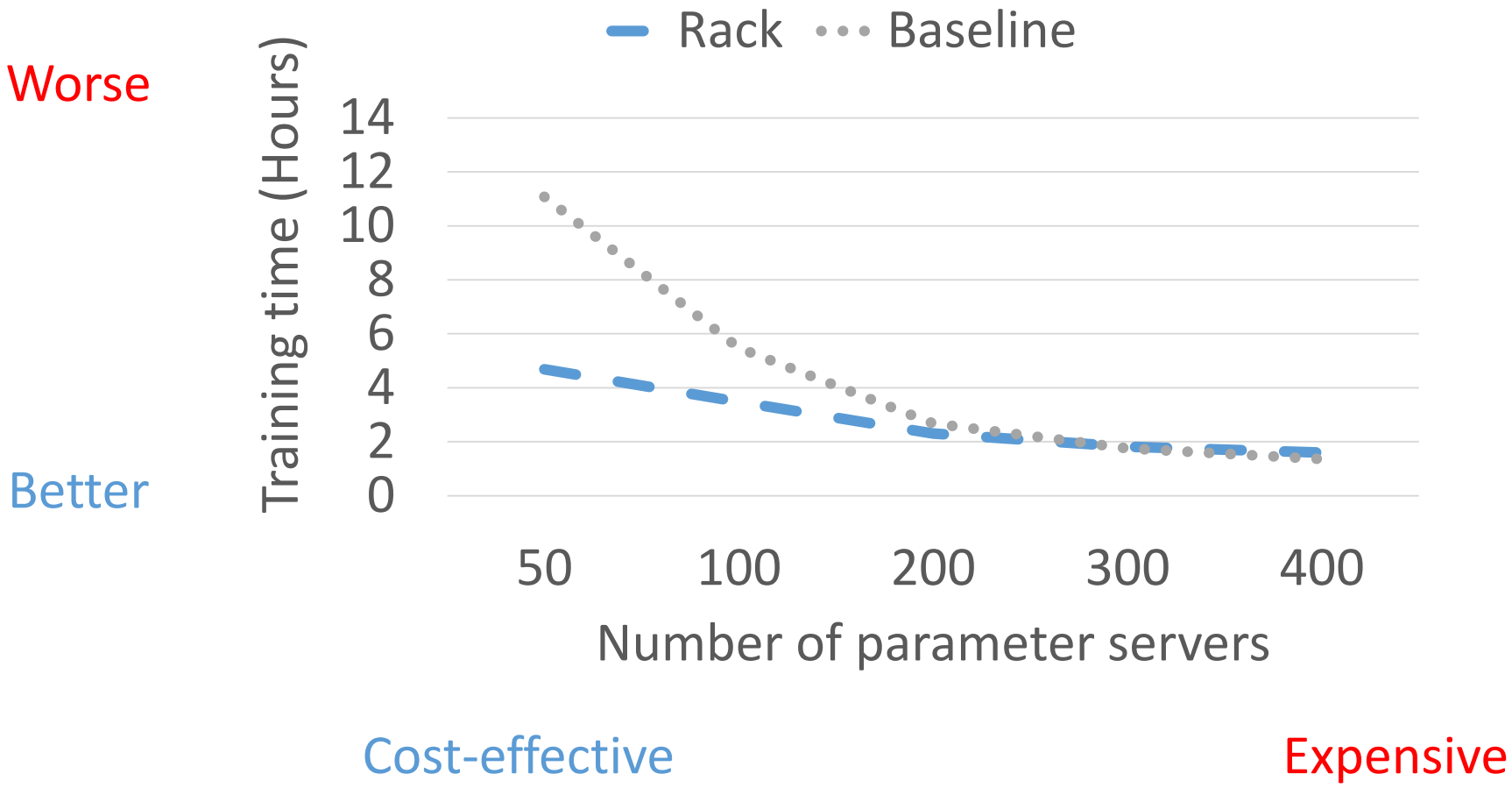
Shorten average FCT can largely improve average training time



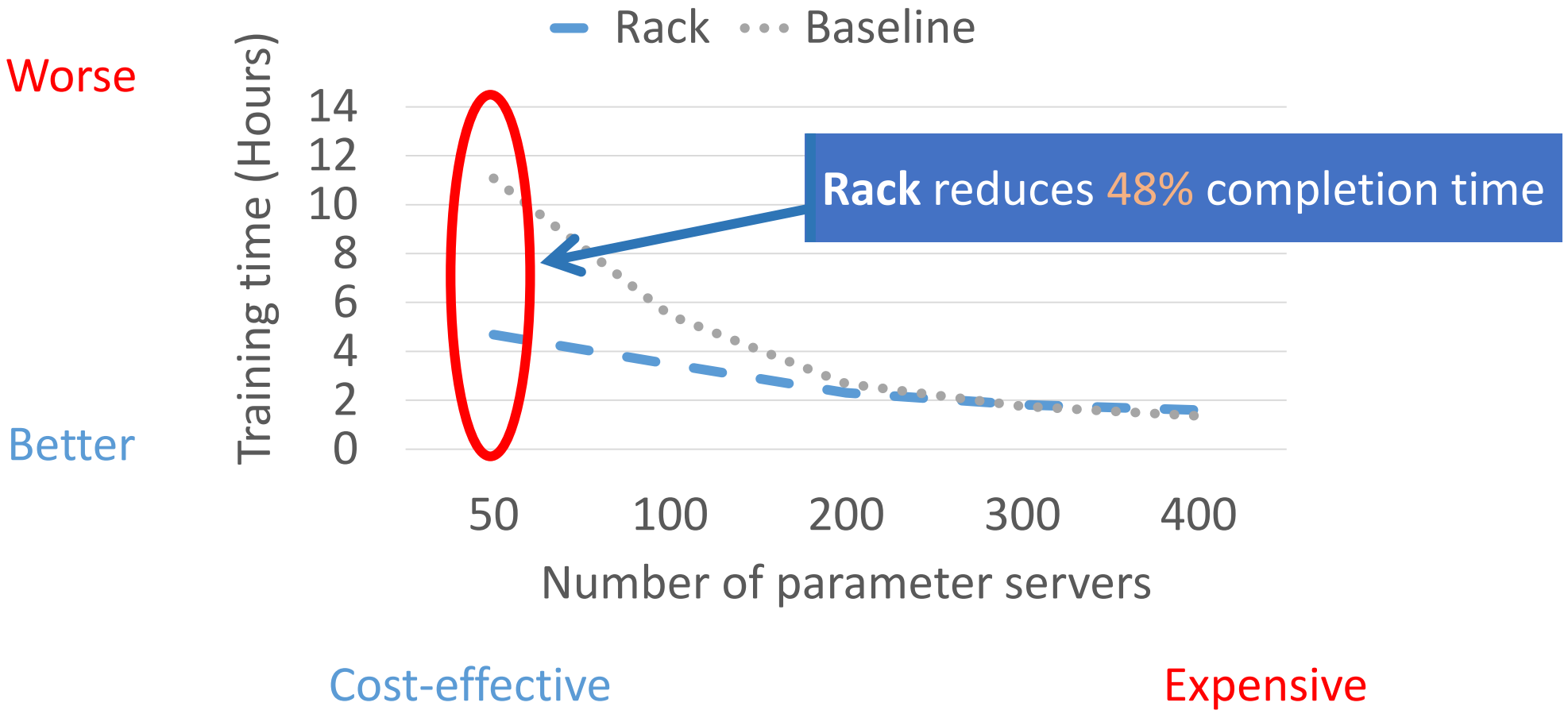
Evaluation

- Simulate common network topology in data centers
  - Classic 10Gbps 1024-node data center topology [Fat-Tree, SIGCOMM'08]
- Training large scale logistic regression
  - 65B parameters, 141TB dataset [Parameter Server, OSDI'14]
  - 800 workers [Parameter Server, OSDI'14]
- With production trace
  - Data processing rate: uniform(100, 200) MBps
  - Synchronize every 30 seconds

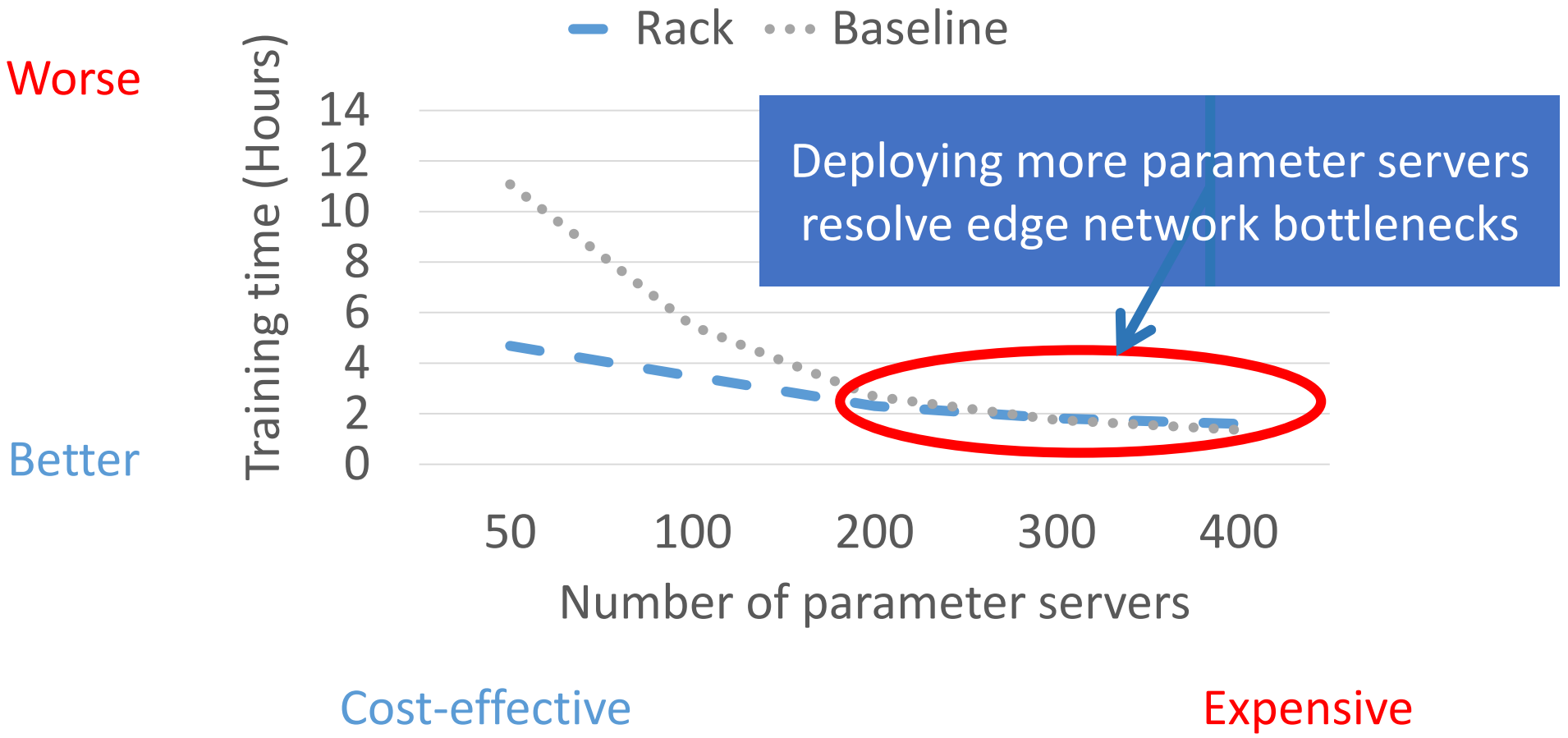
Traffic Reduction (Non-oversubscribed Net.)



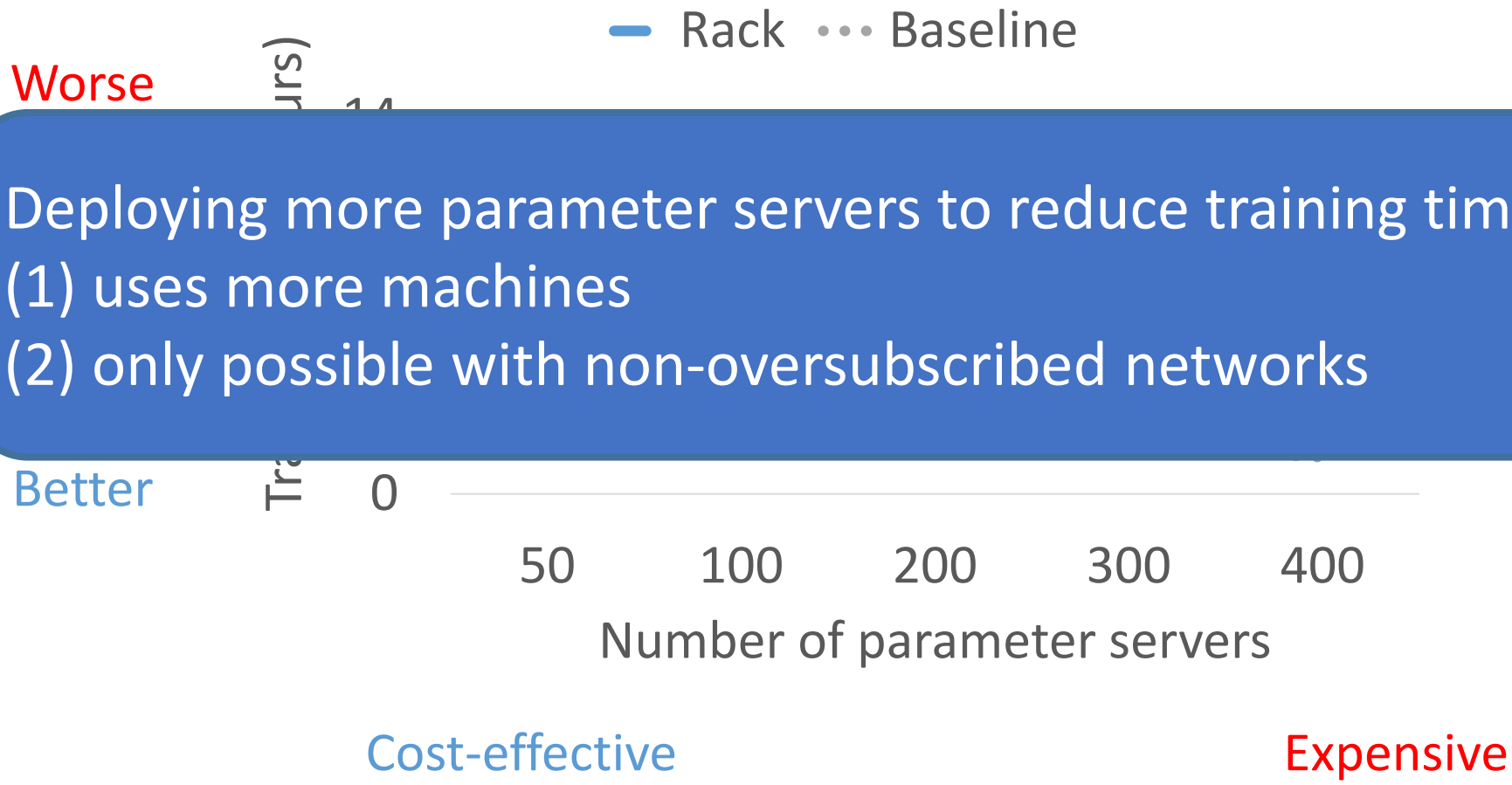
Traffic Reduction (Non-oversubscribed Net.)



Traffic Reduction (Non-oversubscribed Net.)

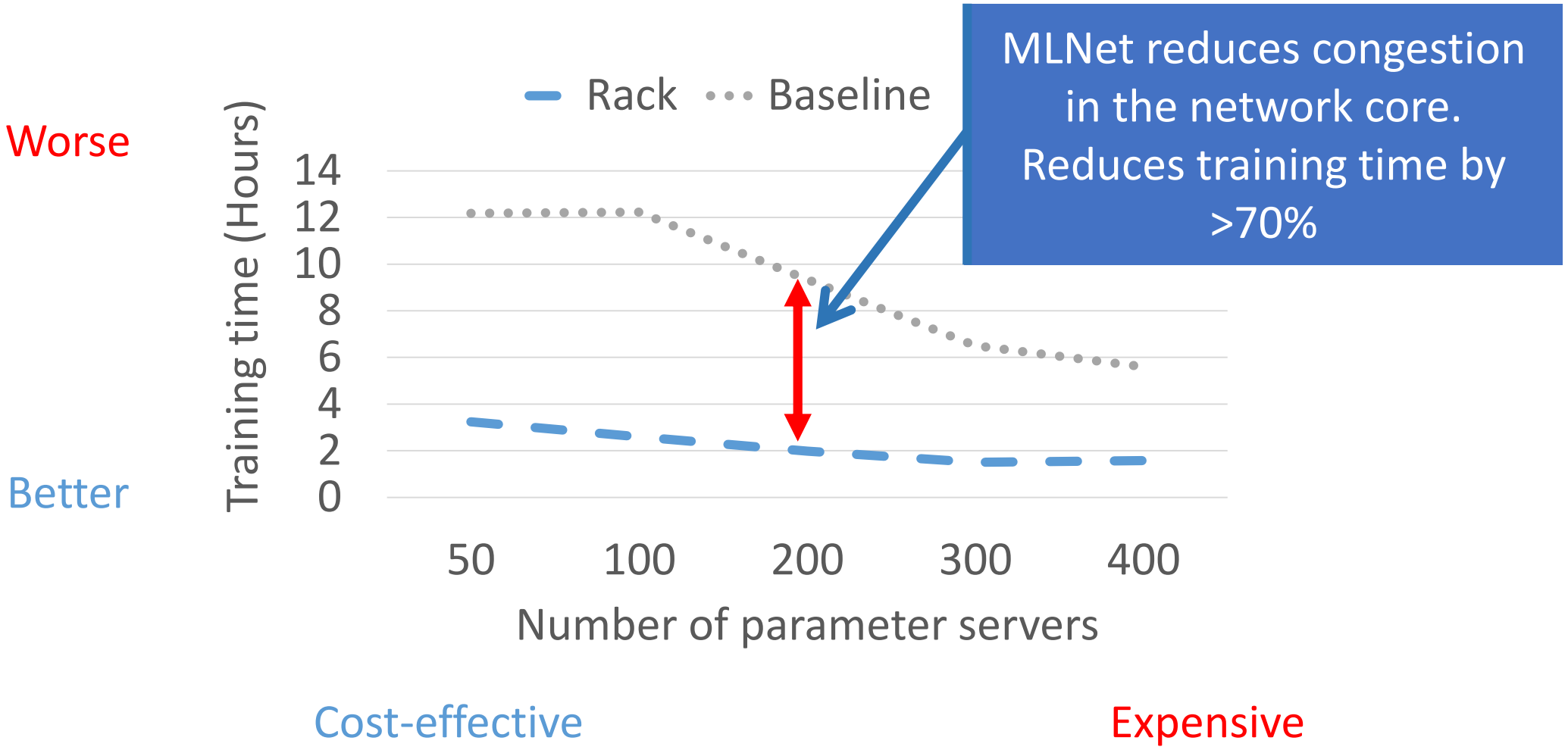


Traffic Reduction (Non-oversubscribed Net.)



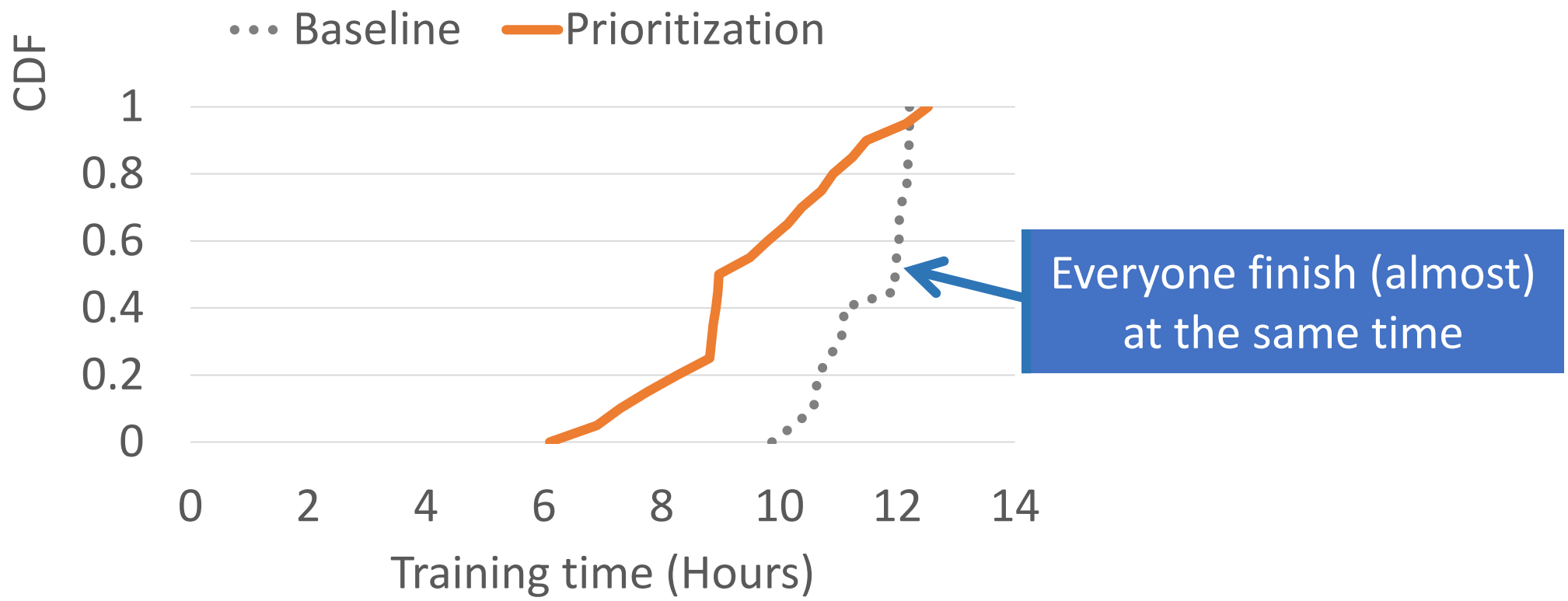
Deploying more parameter servers to reduce training time
(1) uses more machines
(2) only possible with non-oversubscribed networks

Traffic Reduction (1:4 Oversubscribed Net.)

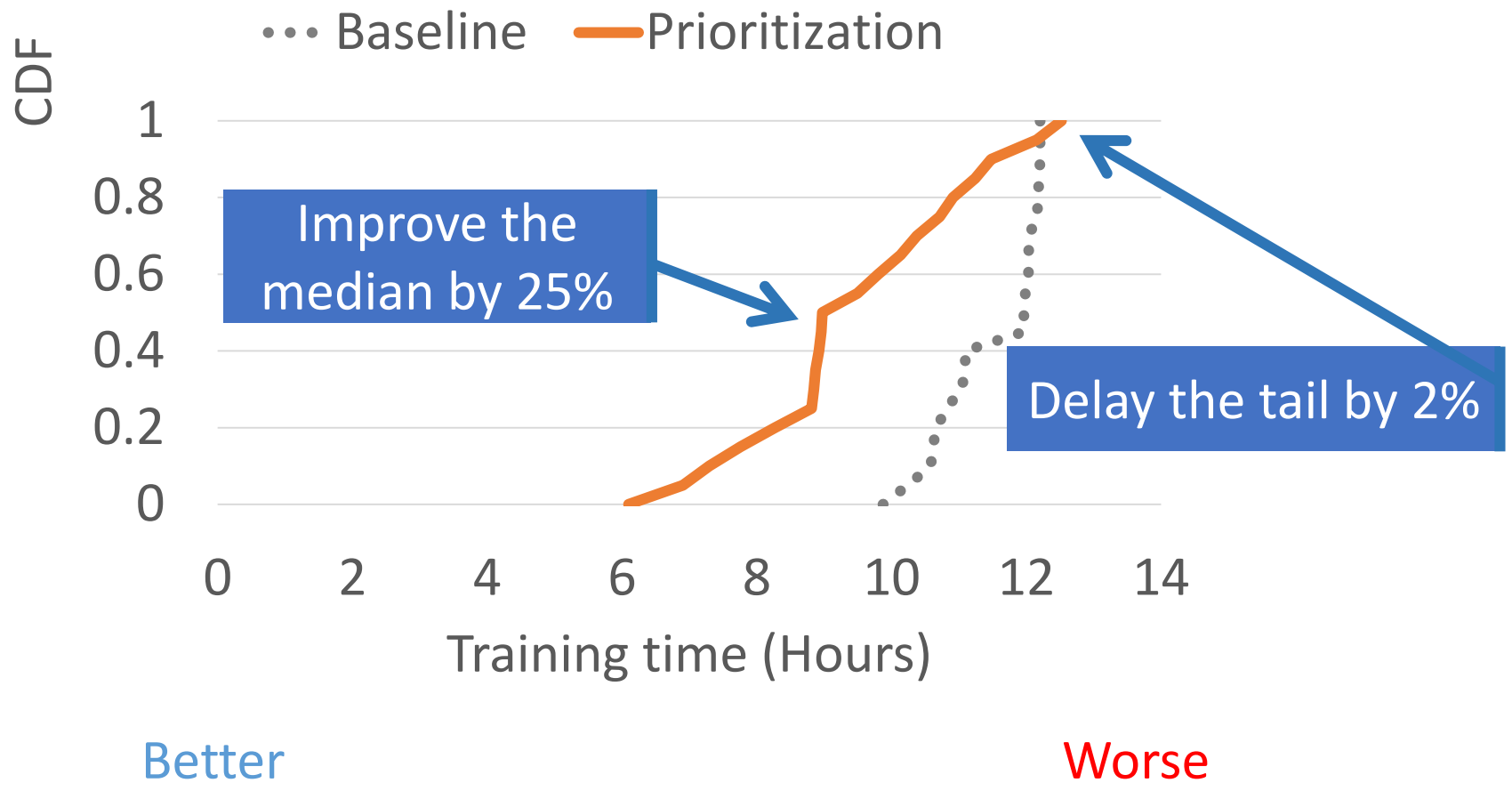


Traffic Prioritization

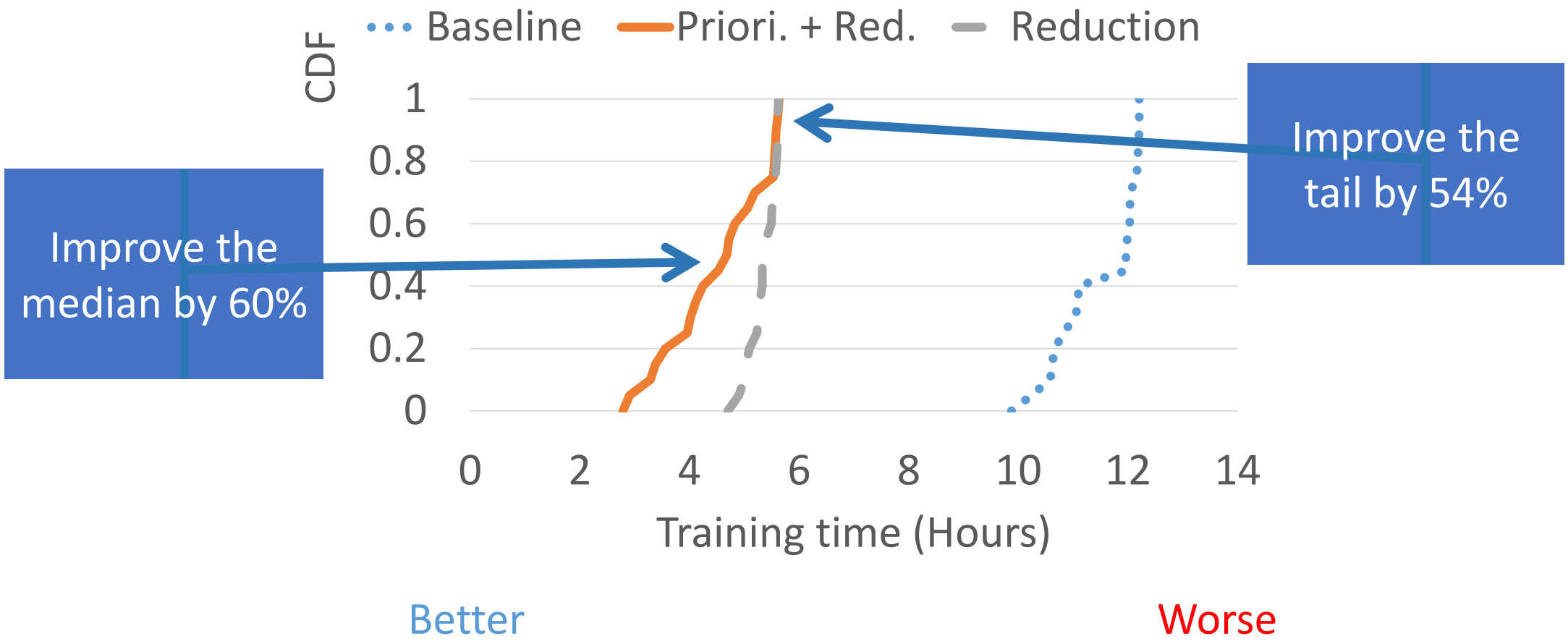
- 20 jobs running in the same cluster



Traffic Prioritization



Traffic Prioritization + Traffic Reduction



More details on the paper:

1. Binary tree aggregation
2. More analysis

Summary

- MLNet can significantly improve the network performance of distributed machine learning
  - Traffic reduction
  - Flow prioritization
  - Drop-in solution

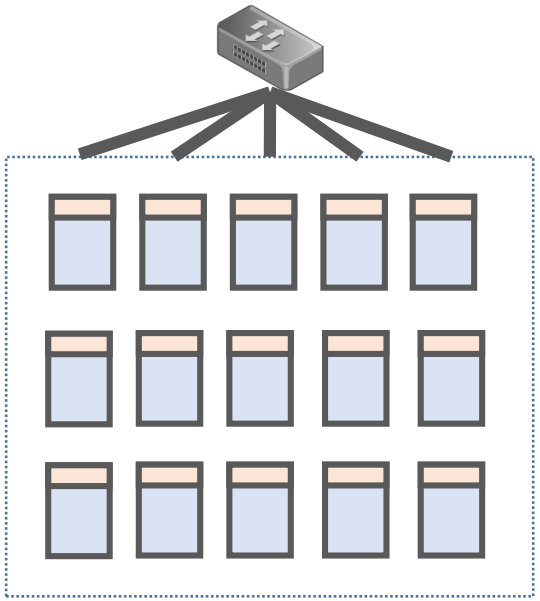
Thanks!

Discussion

- Relaxed fault-tolerance?
  - When worker fails, drop that portion of data
- Adaptive communication
  - Reduce synchronization when network is busy?
- Hybrid network infrastructure?
  - Some with 10GE, some with 40GE ROCE, etc.
- Degree of tree?

Traffic Reduction: Design

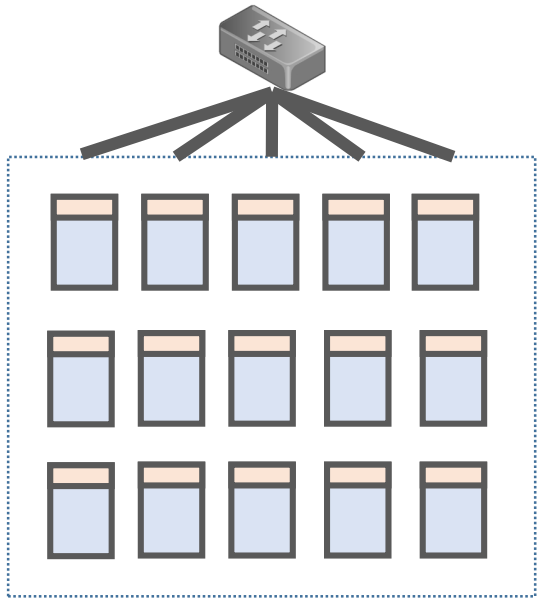
Is the local aggregator a new bottleneck?



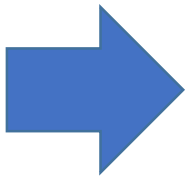
Example: 15 workers in a rack

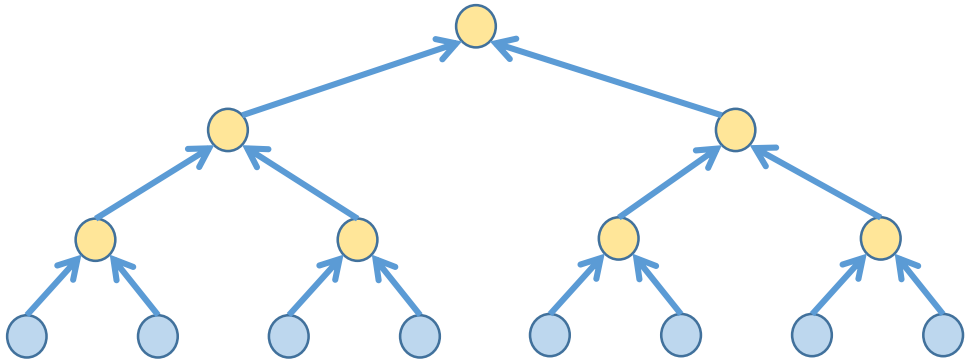
Traffic Reduction: Design

Build a balanced aggregation structure such as a binary tree.



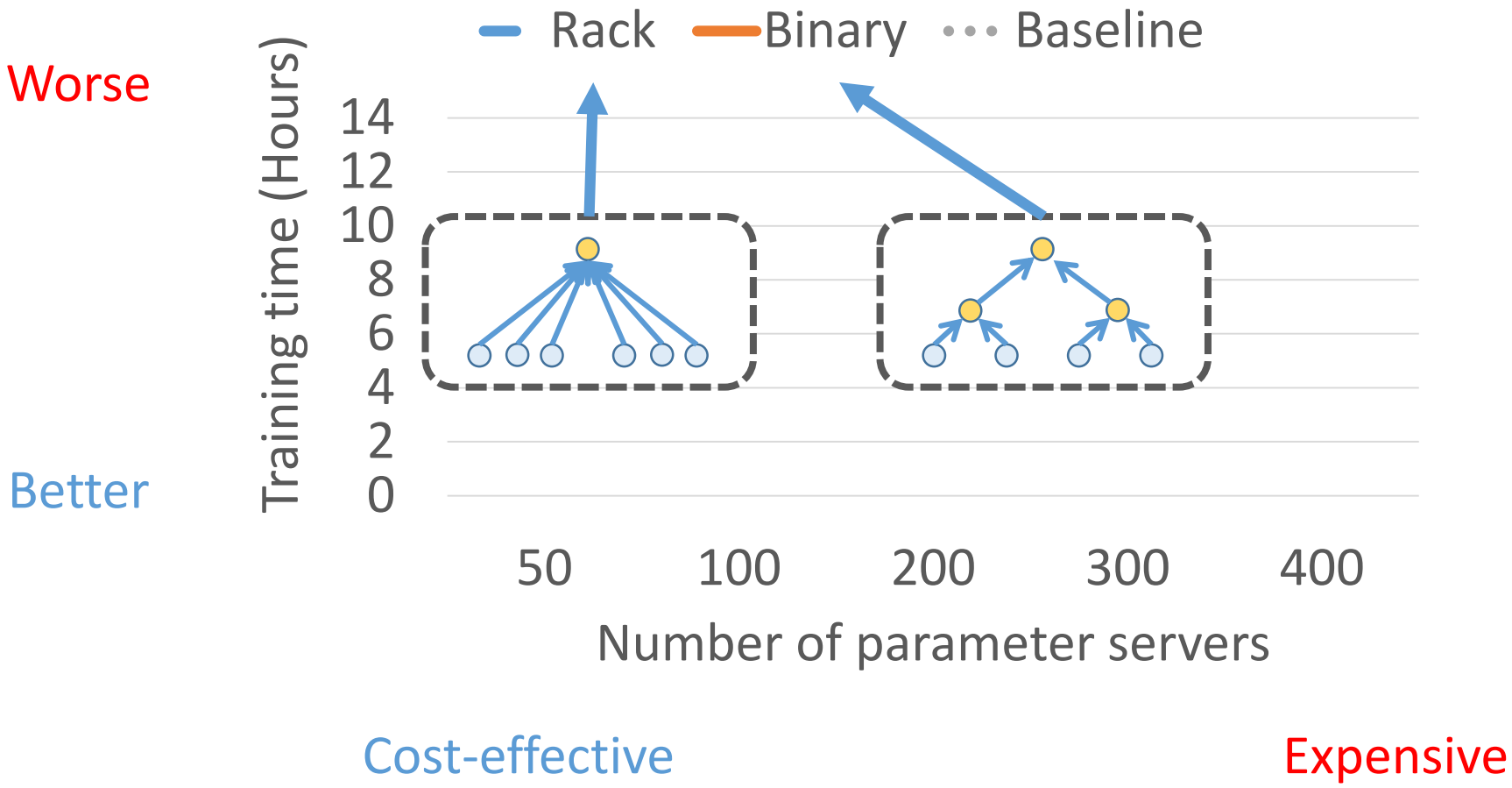
Example: 15 workers in a rack



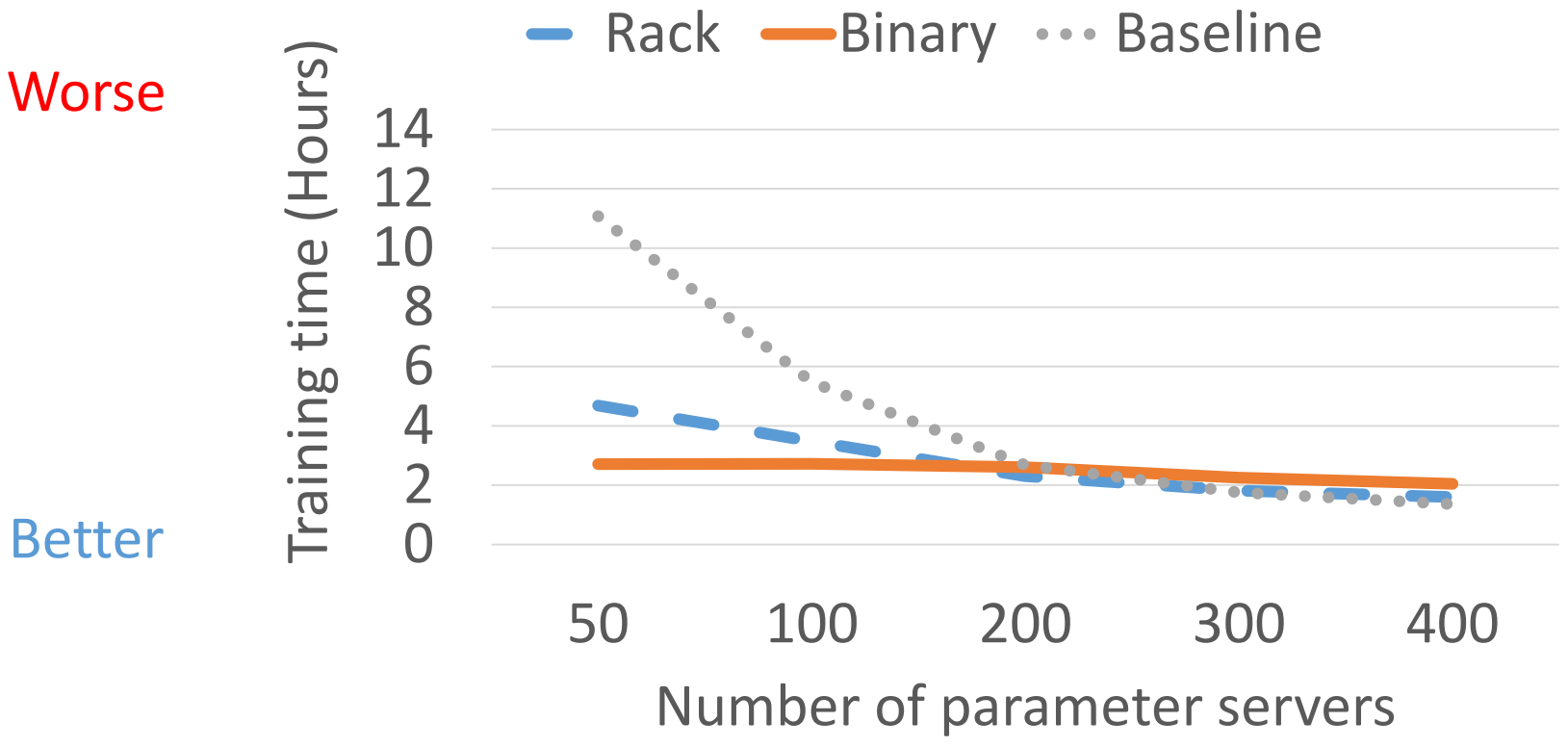


Binary tree aggregation

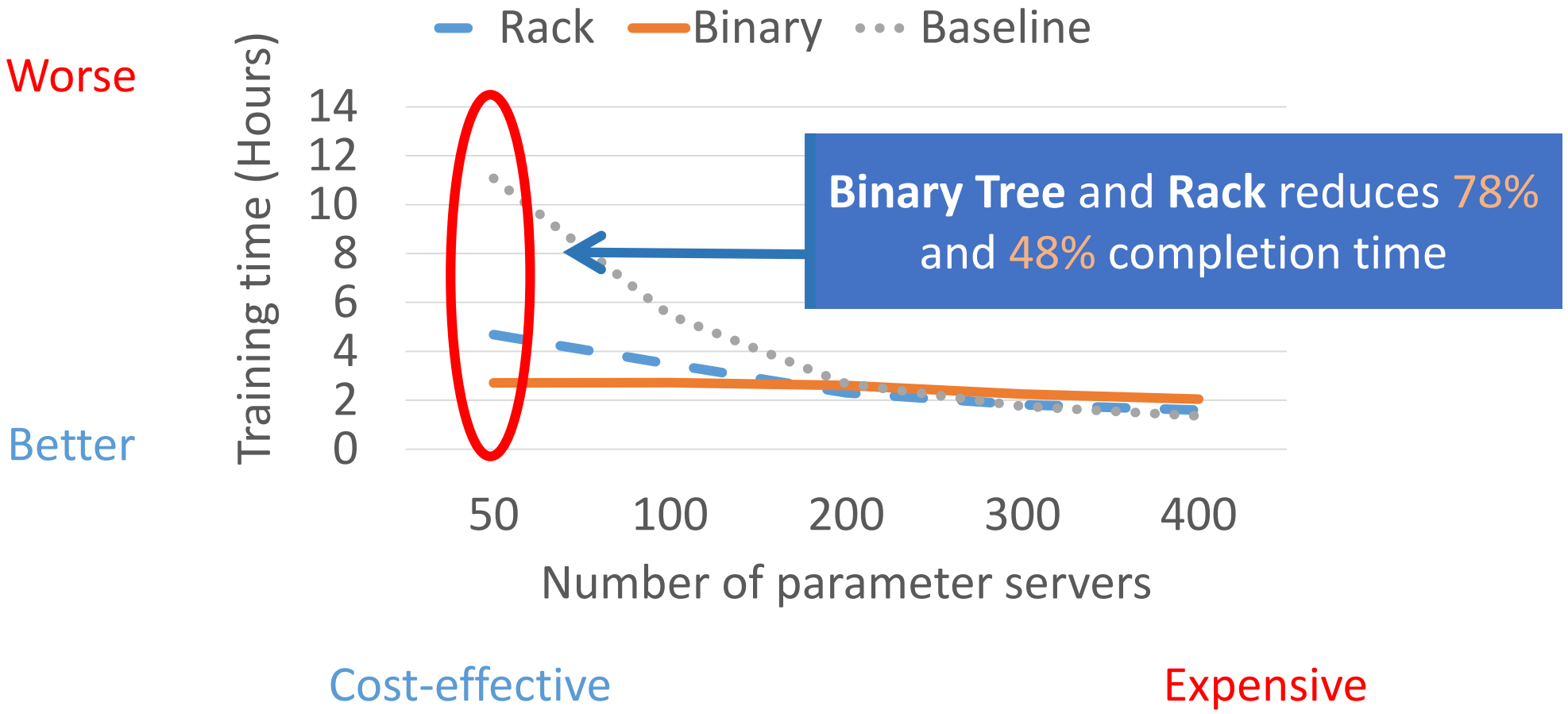
Traffic Reduction



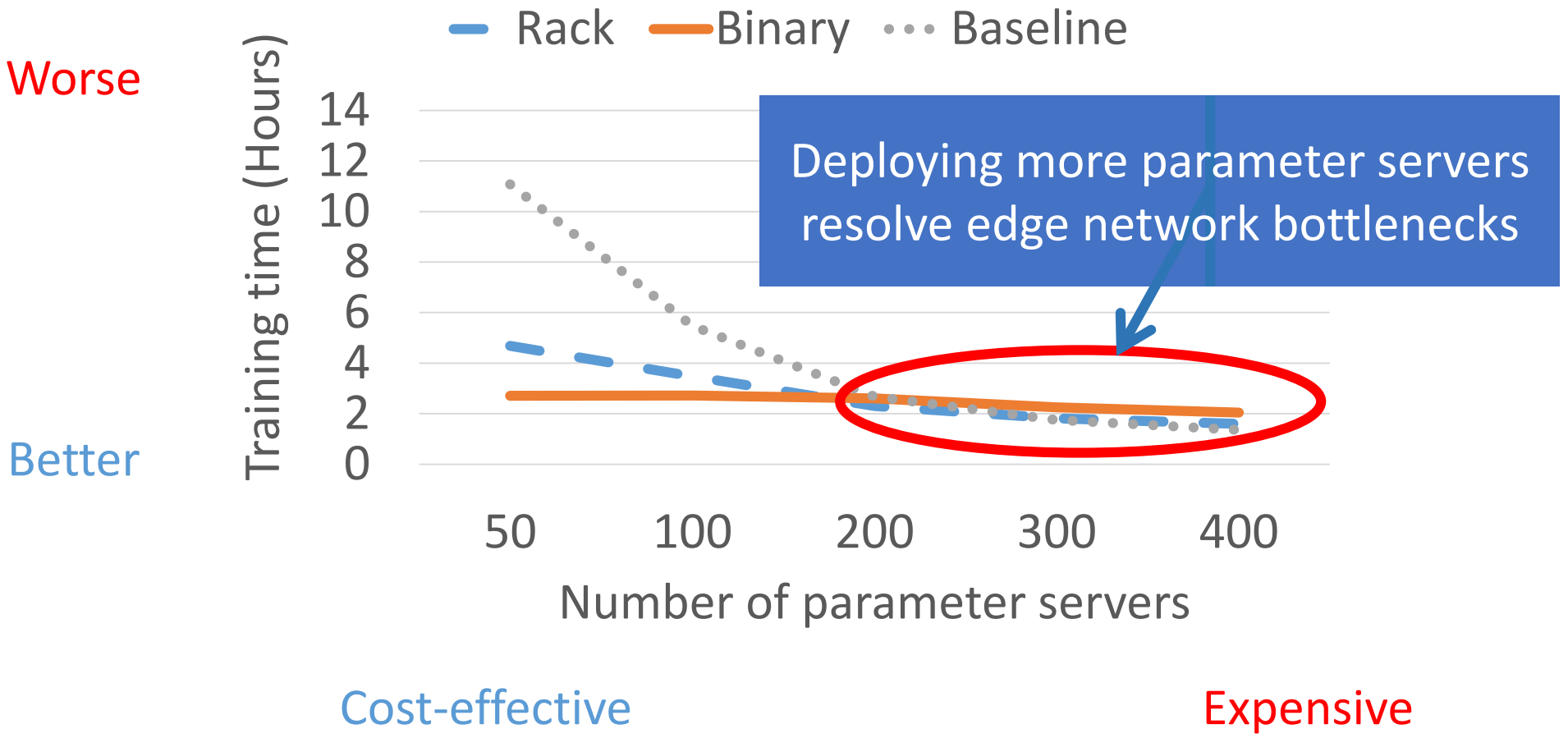
Traffic Reduction (Non-oversubscribed Net.)



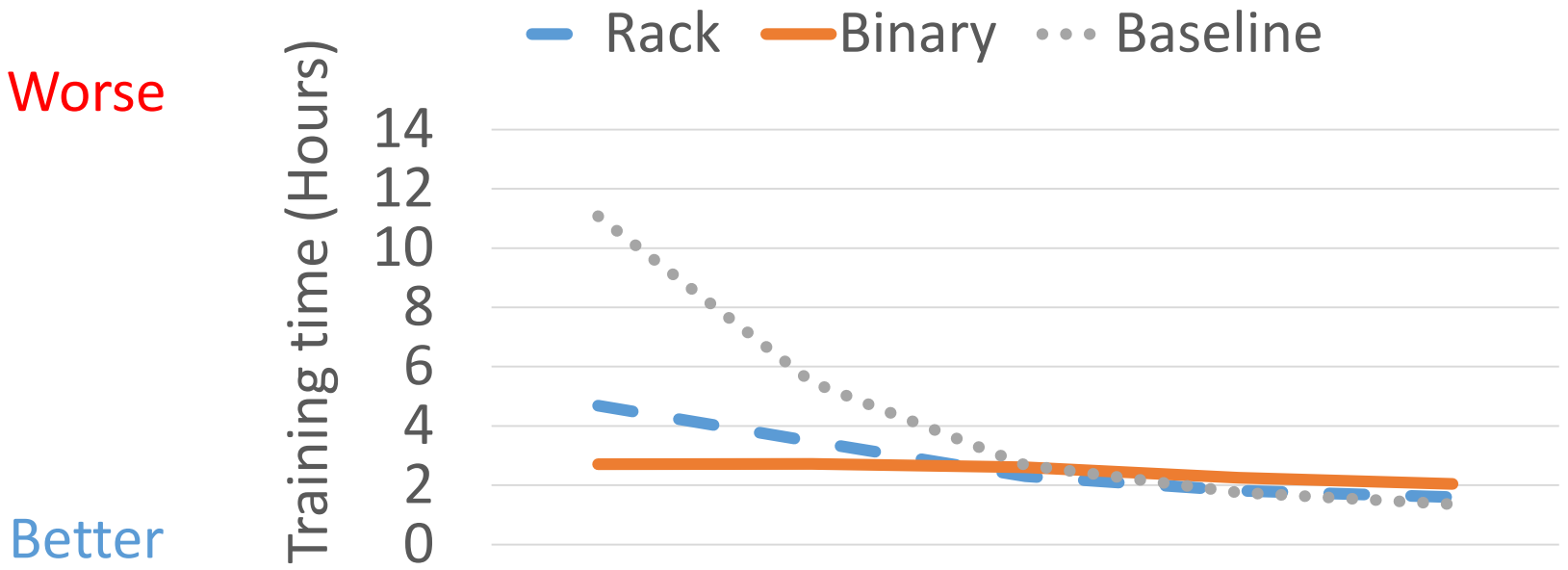
Traffic Reduction (Non-oversubscribed Net.)



Traffic Reduction (Non-oversubscribed Net.)

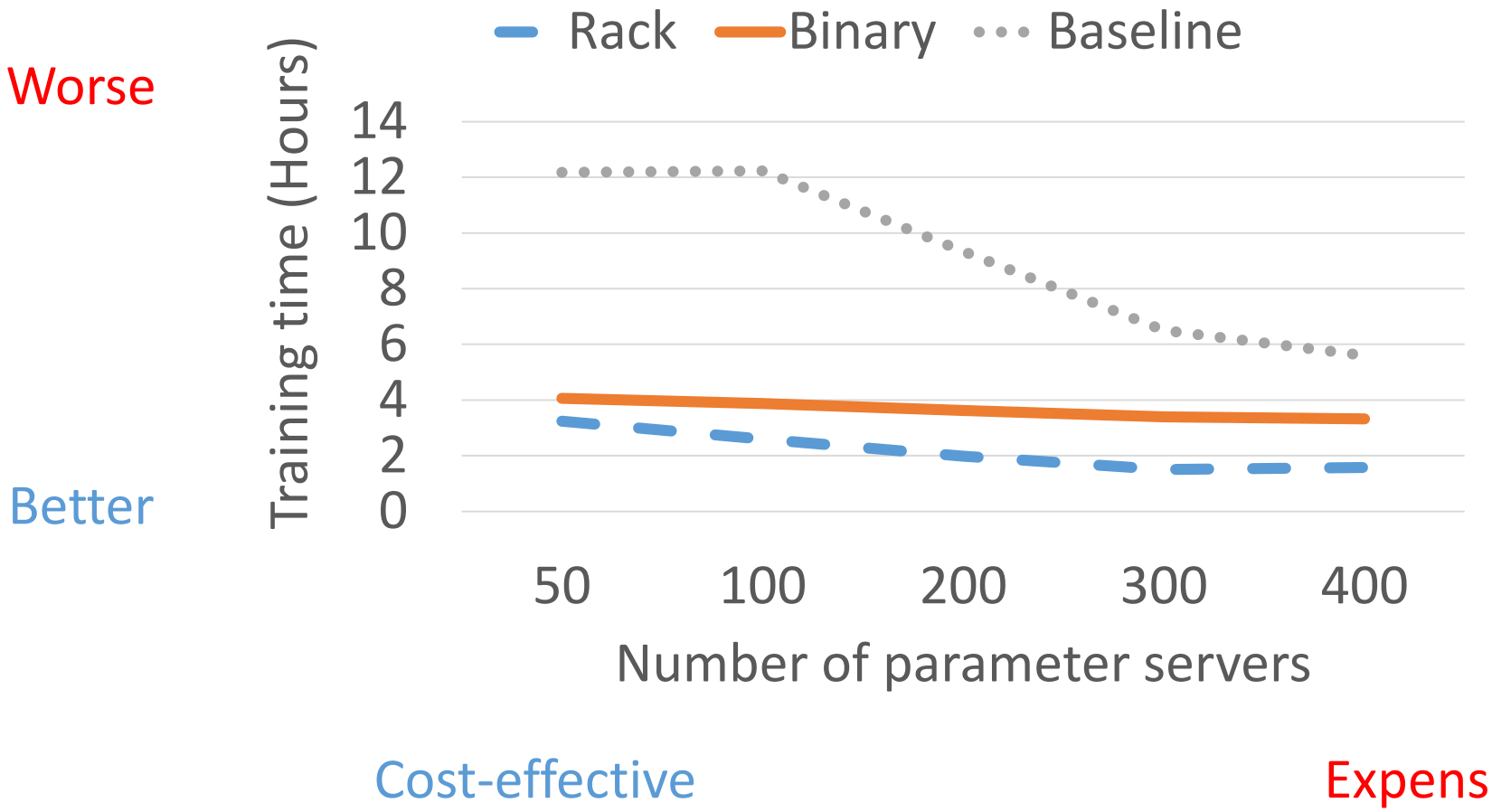


Traffic Reduction (Non-oversubscribed Net.)

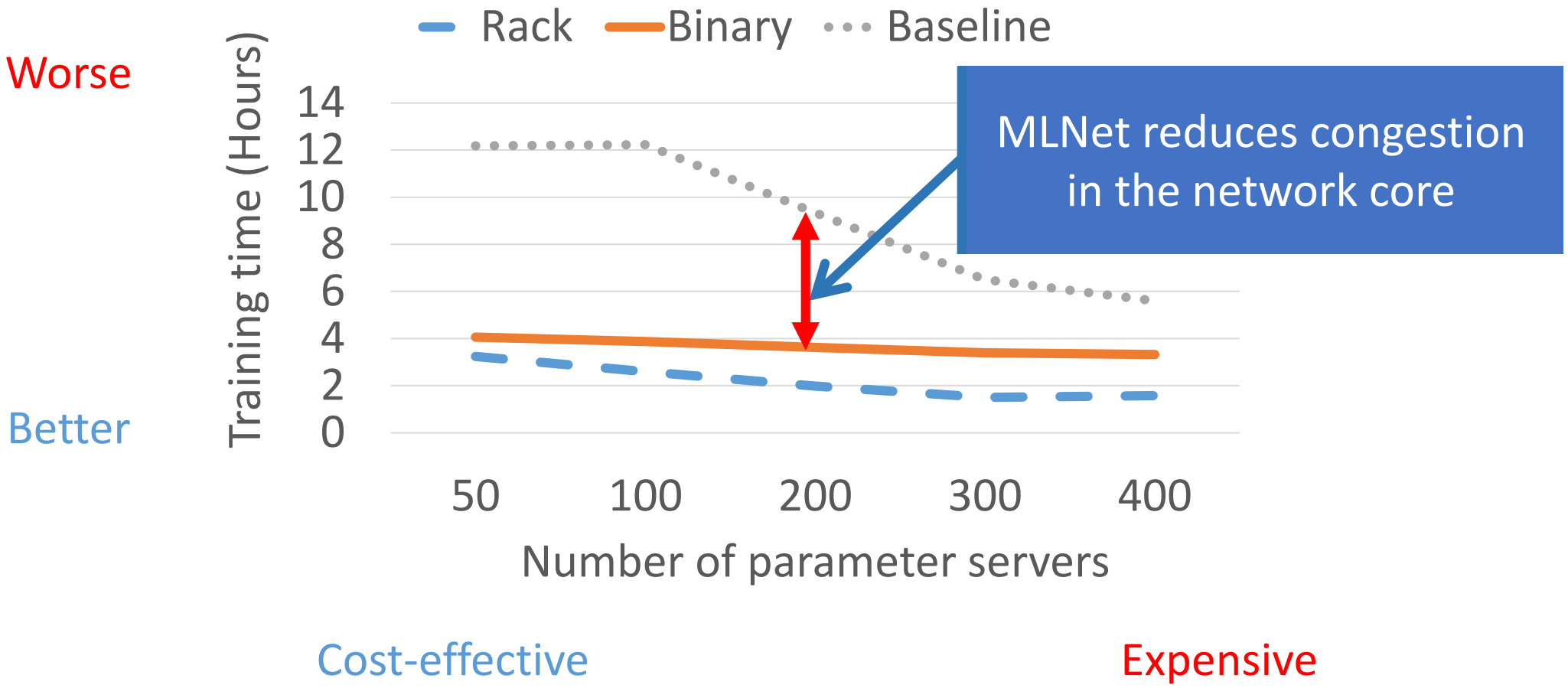


Deploying more parameter servers to reduce training time
(1) needs more machines
(2) only possible with non-oversubscribed networks

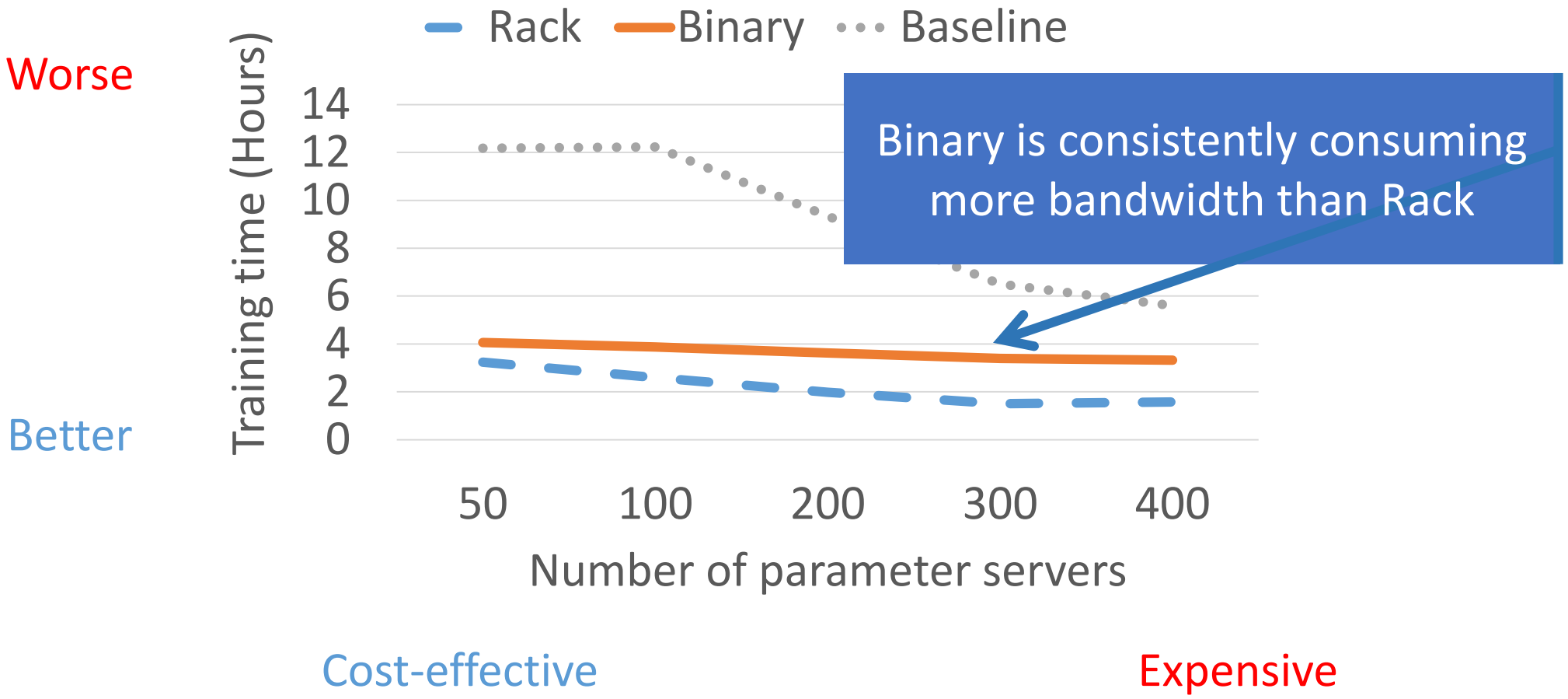
Traffic Reduction (1:4 Oversubscribed Net.)



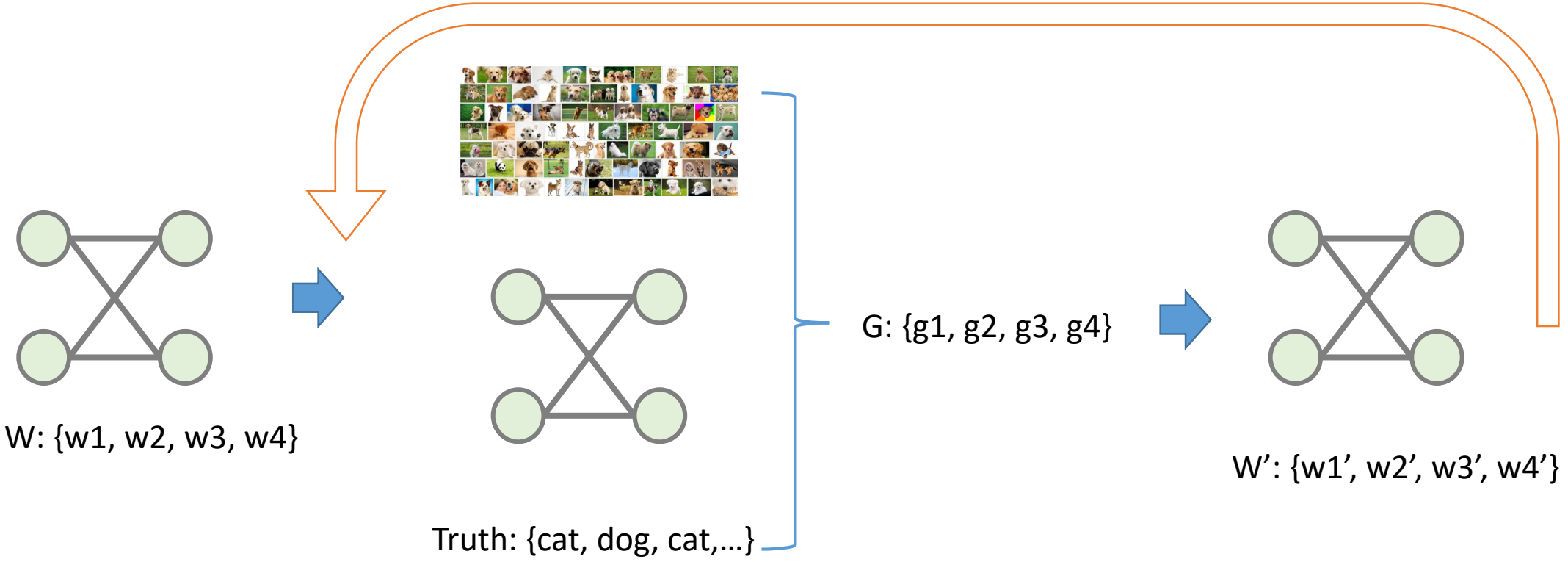
Traffic Reduction (1:4 Oversubscribed Net.)



Traffic Reduction (1:4 Oversubscribed Net.)

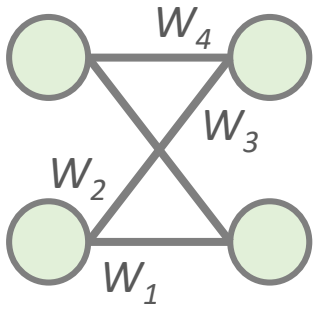


Example: Training a Neural Network



Example: Neural Network

Model



Train

Apply

Dog : 99%

Cat : 1%

Model Training