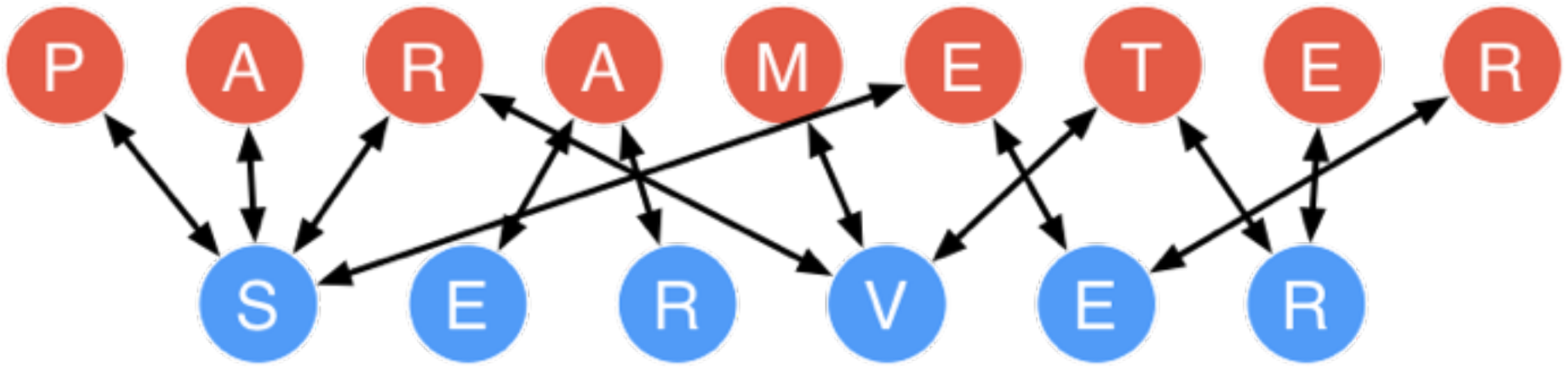
Scaling Distributed Machine Learning with the



Mu Li

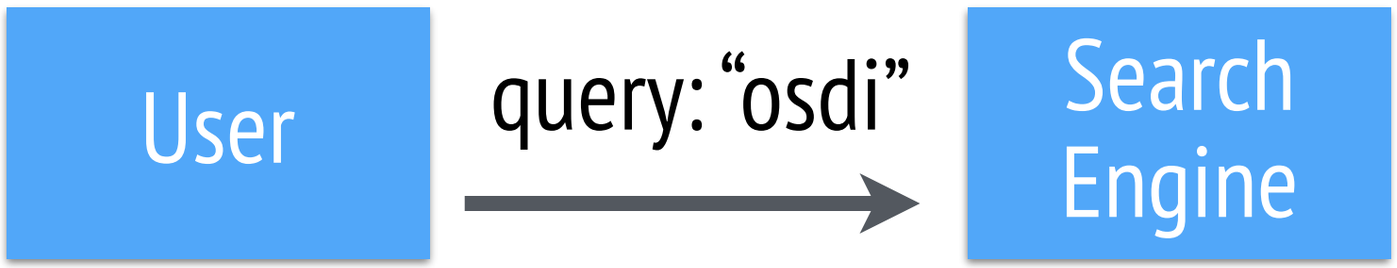
muli@cs.cmu.edu

PARALLEL DATA LABORATORY
CARNEGIE MELLON UNIVERSITY

Baidu 百度

Google

How does a search engine make money?

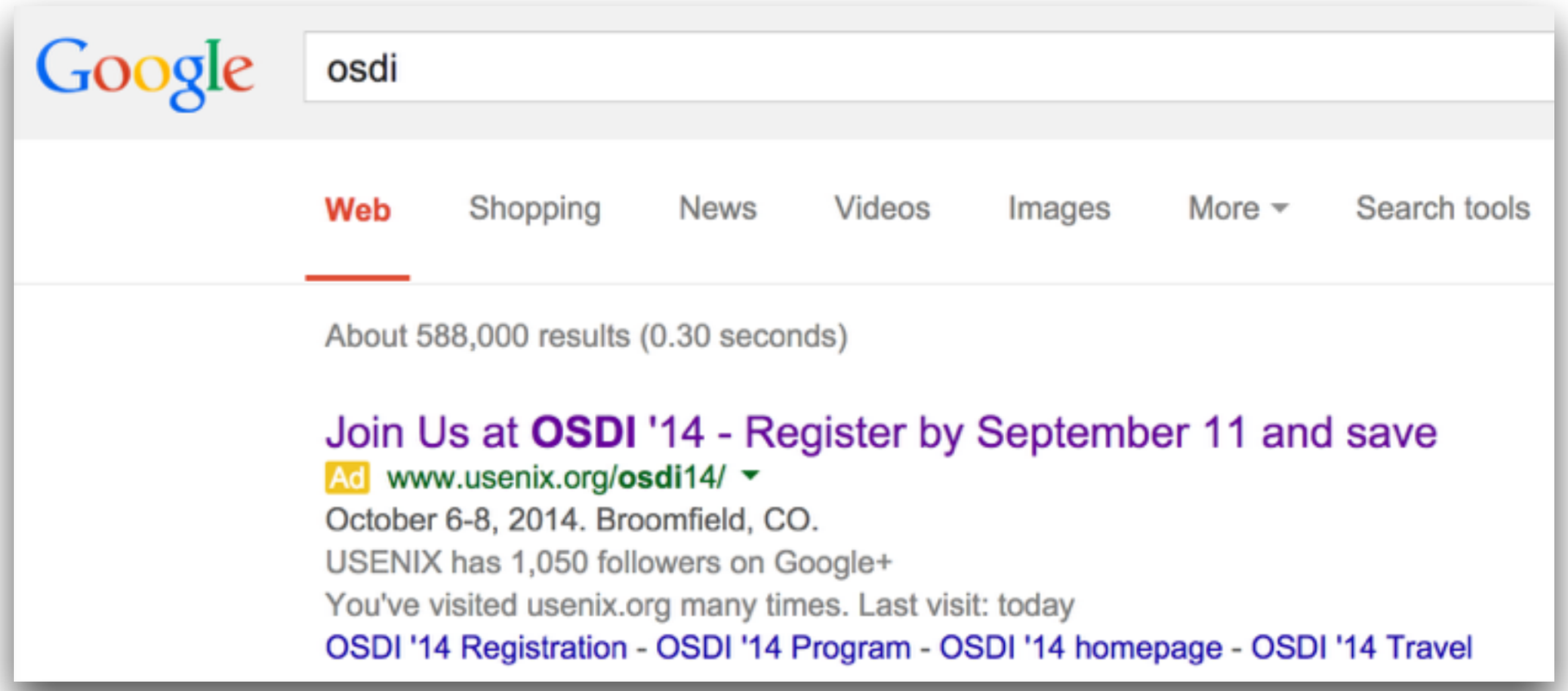


How does a search engine make money?

User

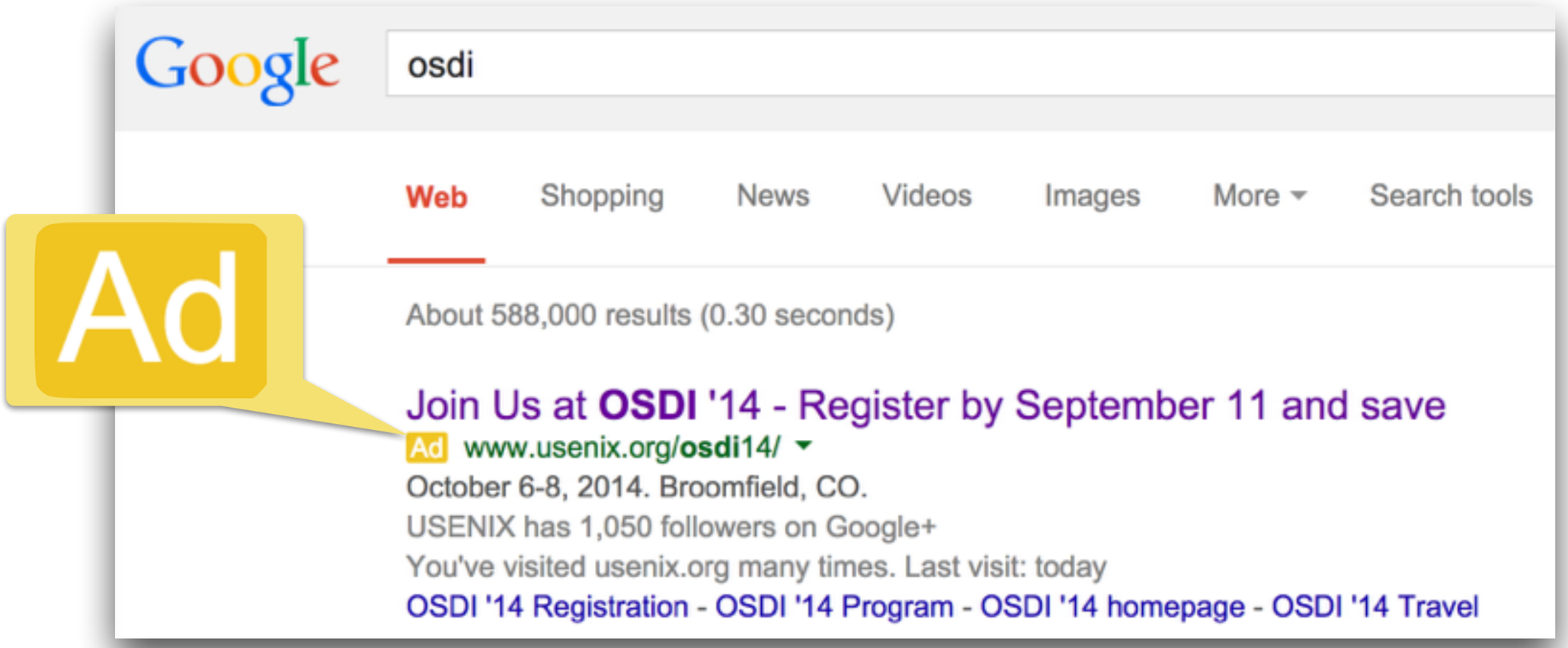
query: “osdi”

Search Engine



How does a search engine make money?

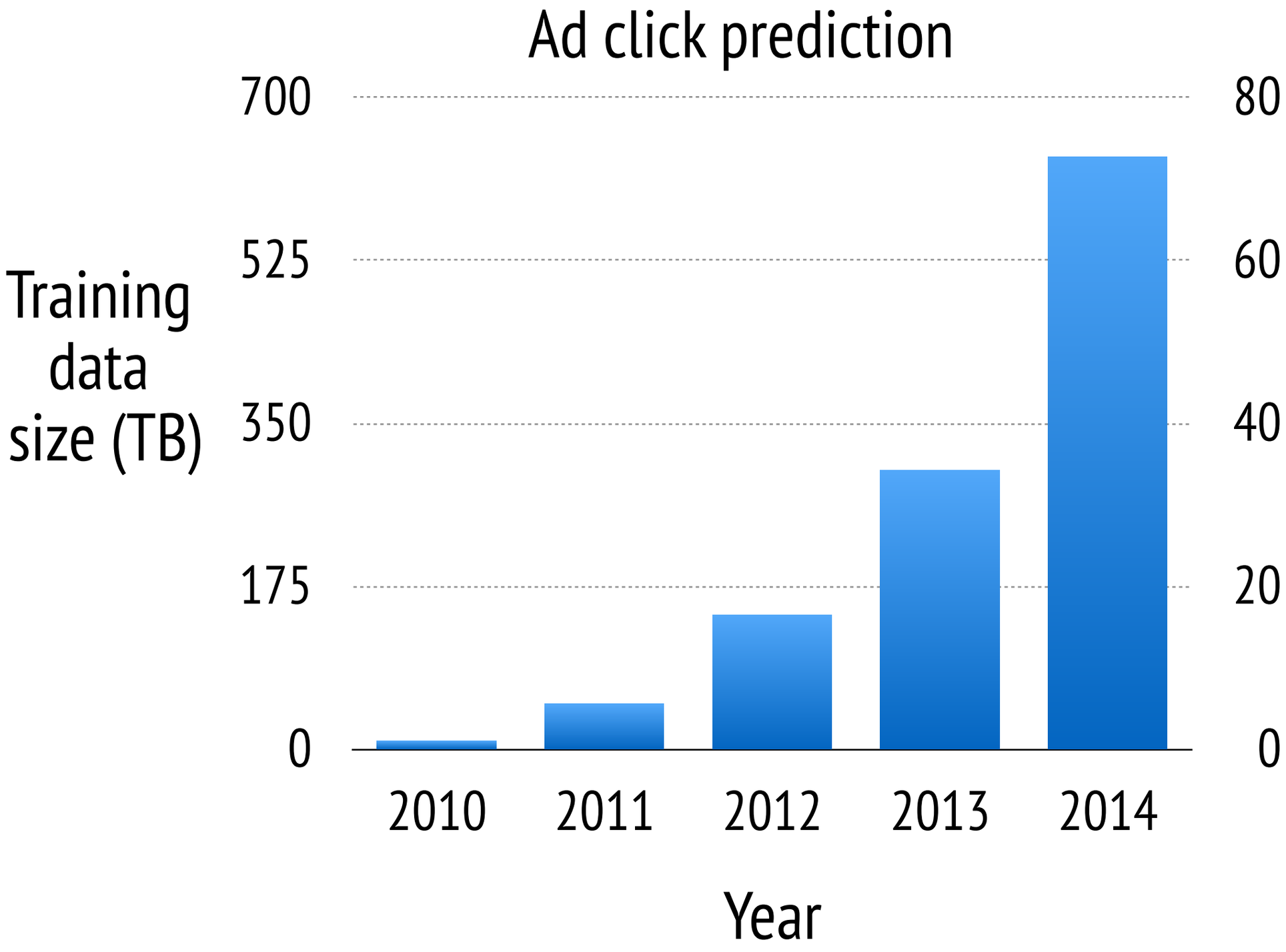
User → query: "osdi" → Search Engine → charge if click → Advertiser



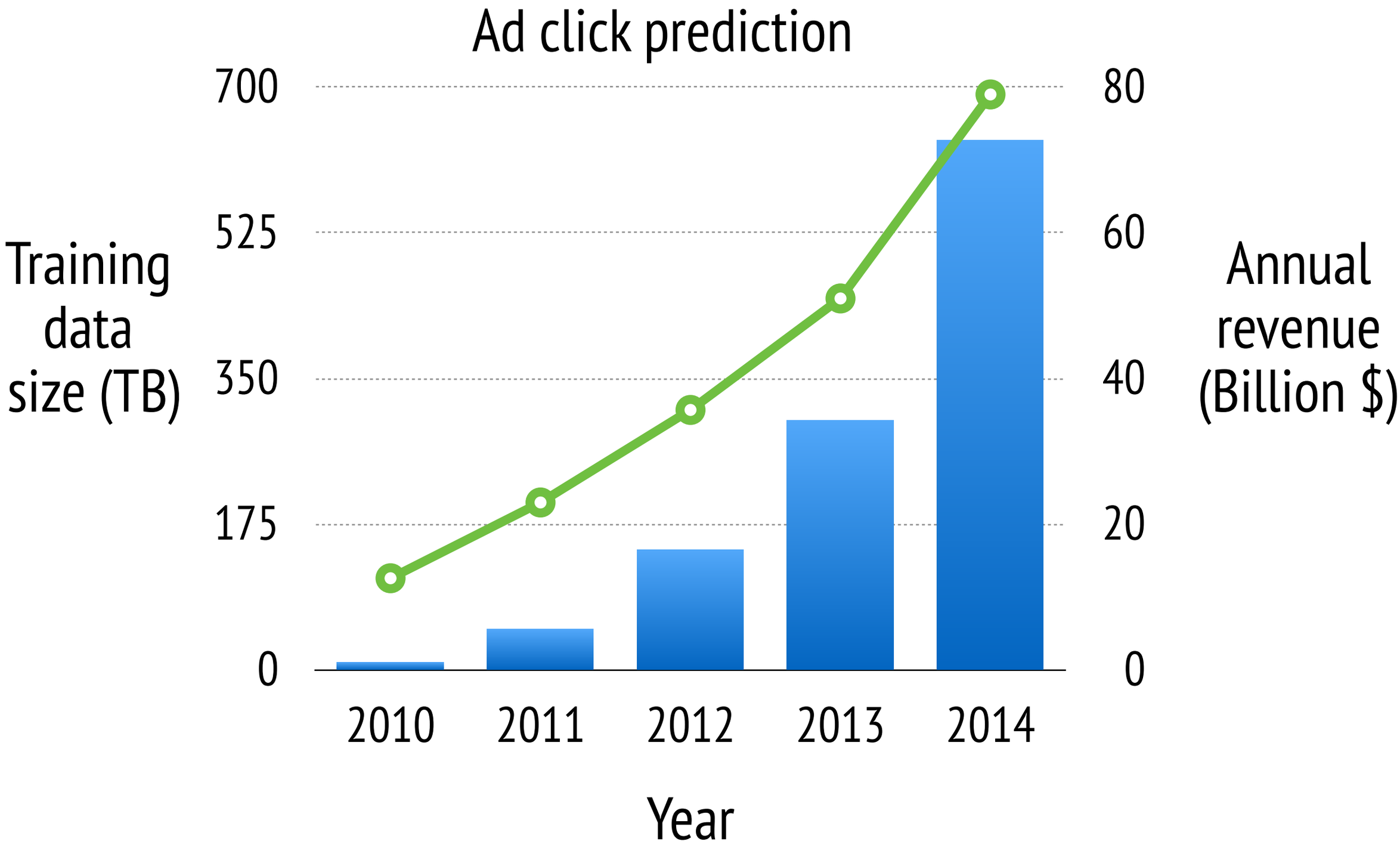
Machine learning problem:
Predict wether or not a user will click an ad

Machine learning is concerned with systems that can learn from data

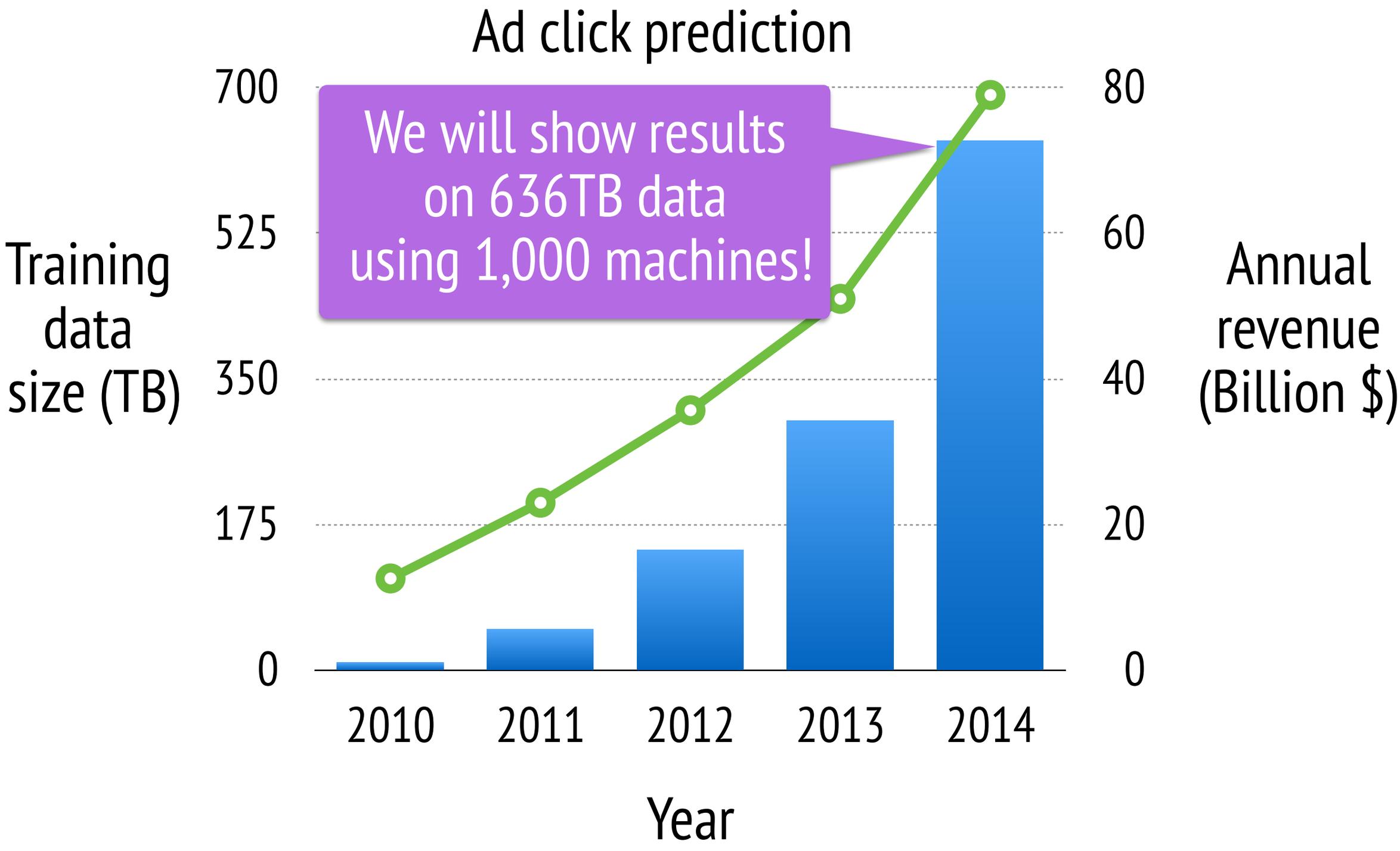
Machine learning is concerned with systems that can learn from data



Machine learning is concerned with systems that can learn from data



Machine learning is concerned with systems that can learn from data



Overview of machine learning

Raw data

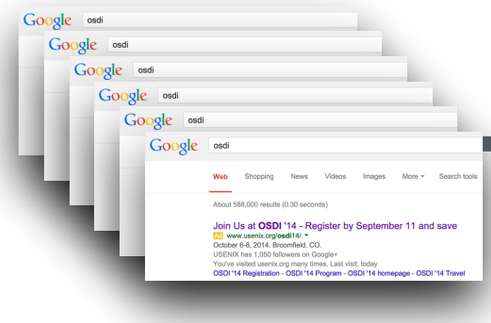
Training data

Machine learning system

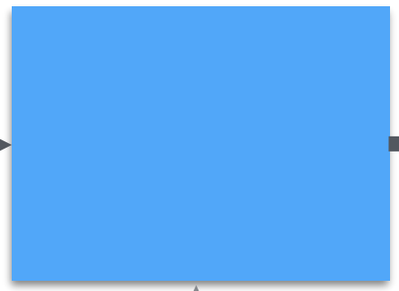
Model:
(key,value) pairs

Overview of machine learning

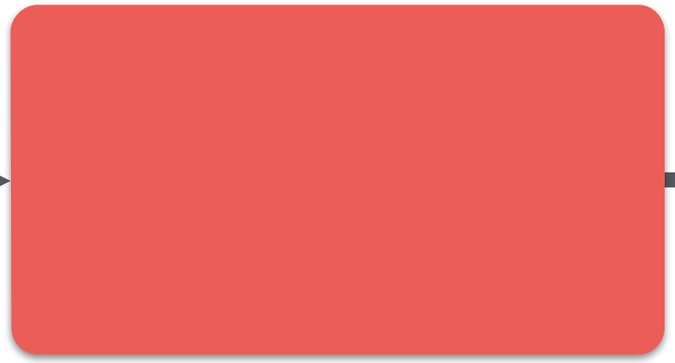
Raw data



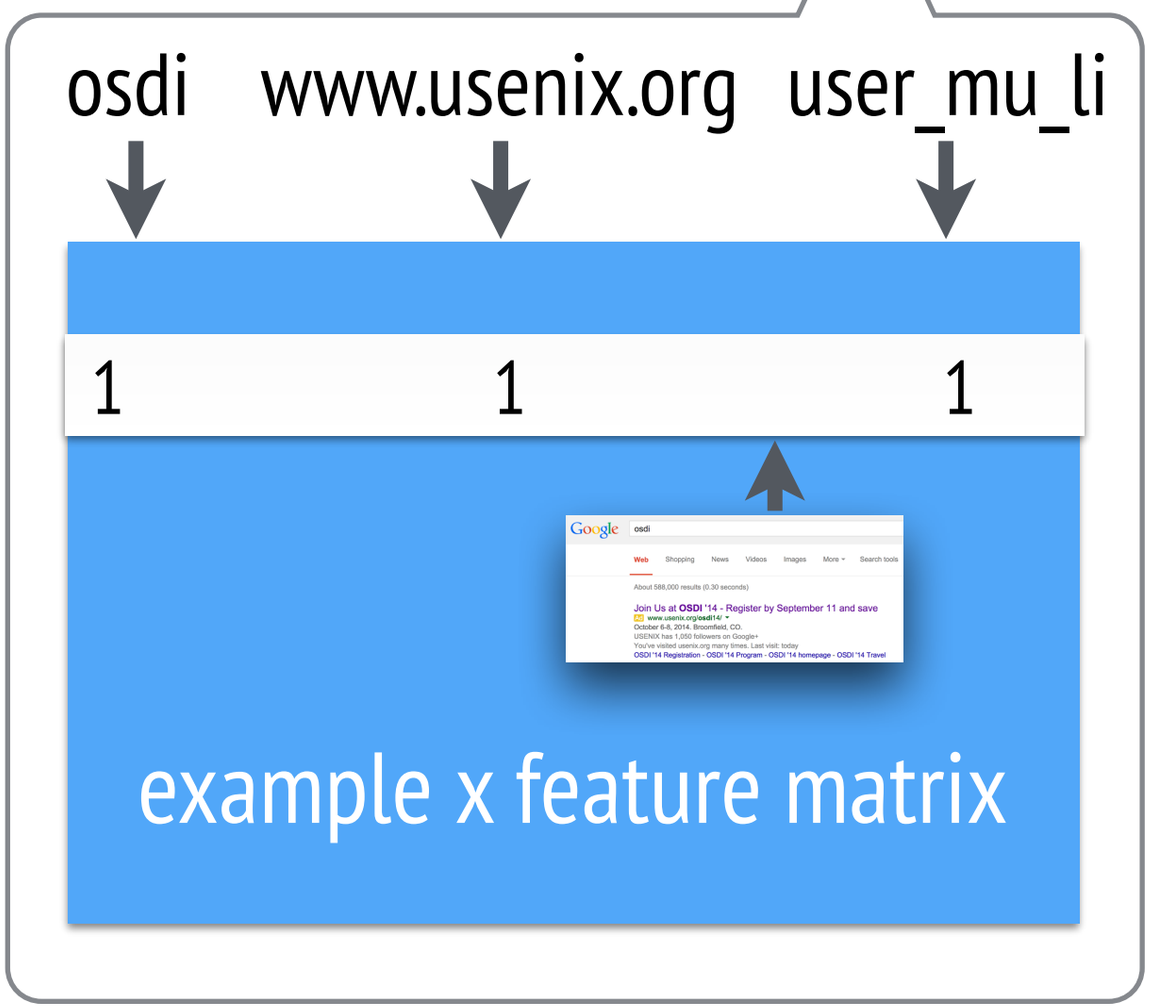
Training data



Machine learning system



Model:
(key,value) pairs



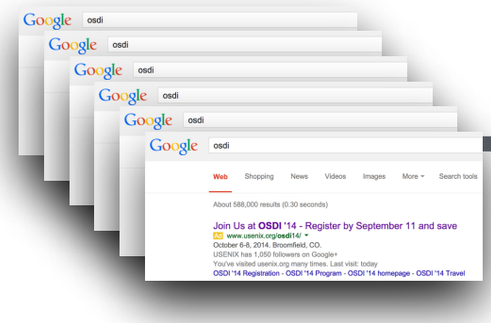
Overview of machine learning

Raw data

Training data

Machine learning system

Model:
(key,value) pairs

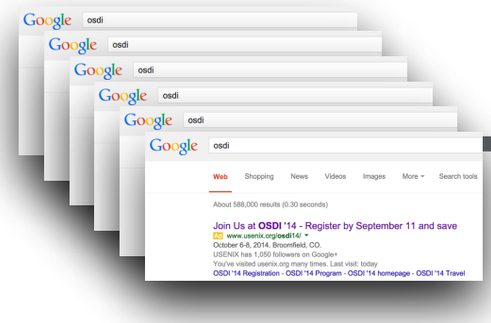


Scale of Industry Problems

- 100 billion examples
- 10 billion features
- 1TB – 1PB training data
- 100 – 1000 machines

Overview of machine learning

Raw data



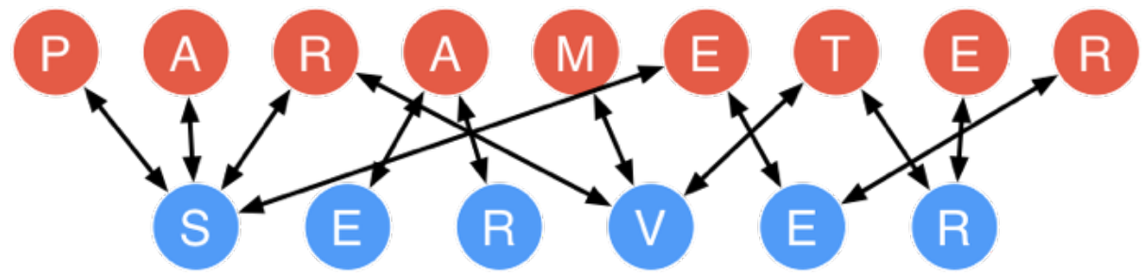
Training data

Machine learning system

Model:
(key,value) pairs

Scale of Industry Problems

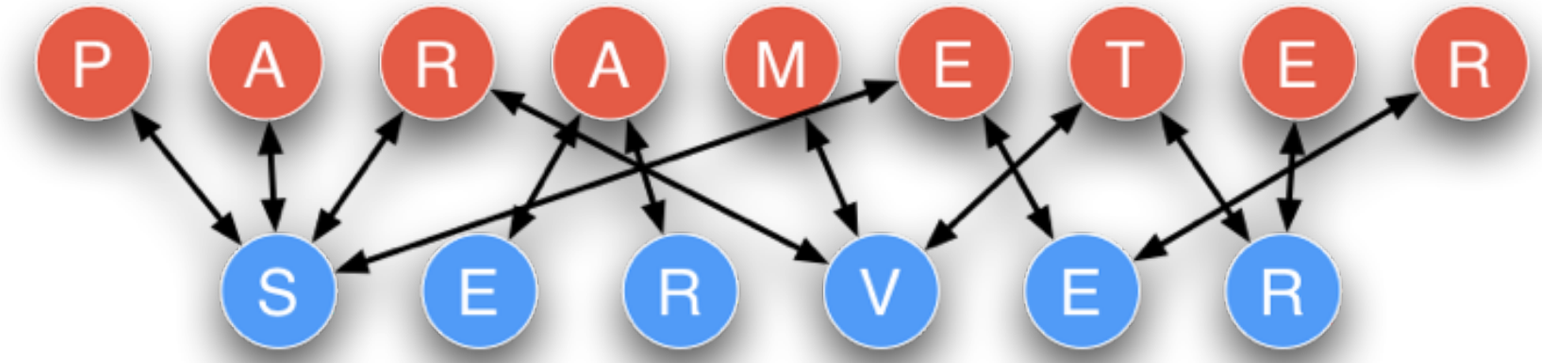
- 100 billion examples
- 10 billion features
- 1TB – 1PB training data
- 100 – 1000 machines



- Scale to industry problems
- Efficient communication
- Fault tolerance
- Easy to use

Industry size machine learning problems

Efficient communication



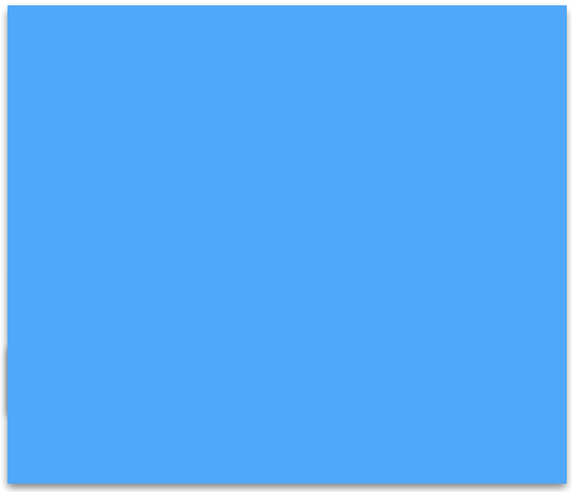
Fault tolerance

Easy to use

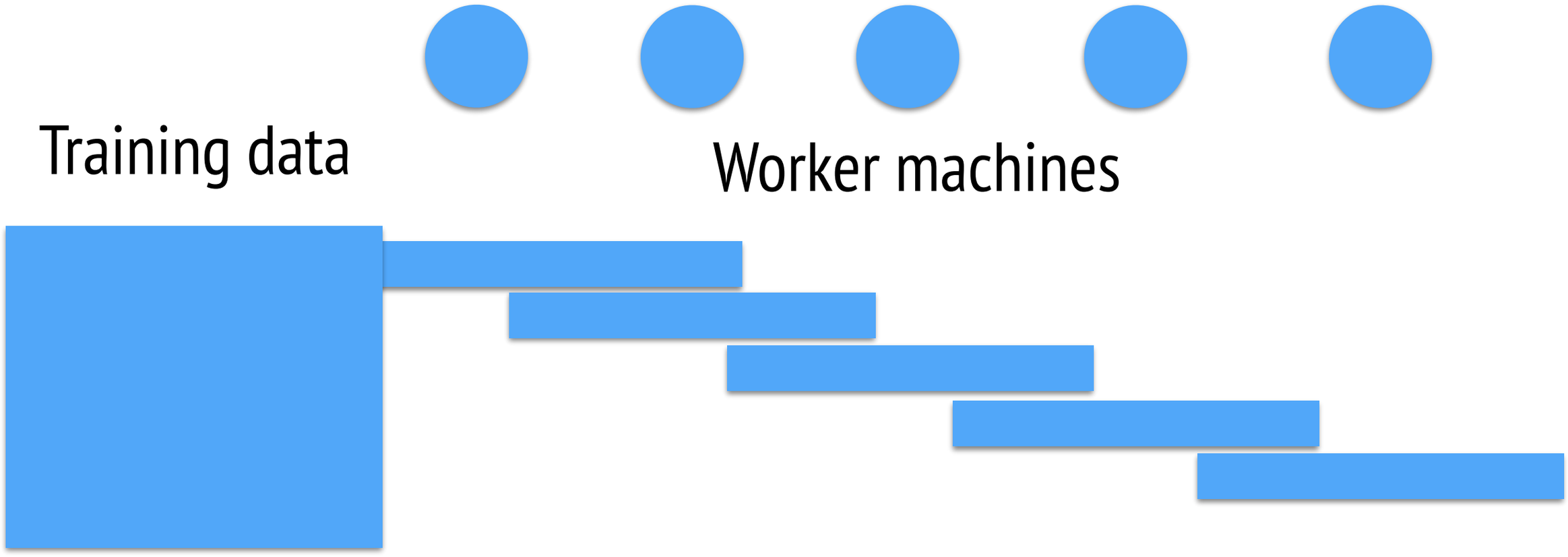
Evaluation

Data and model partition

Training data



Data and model partition

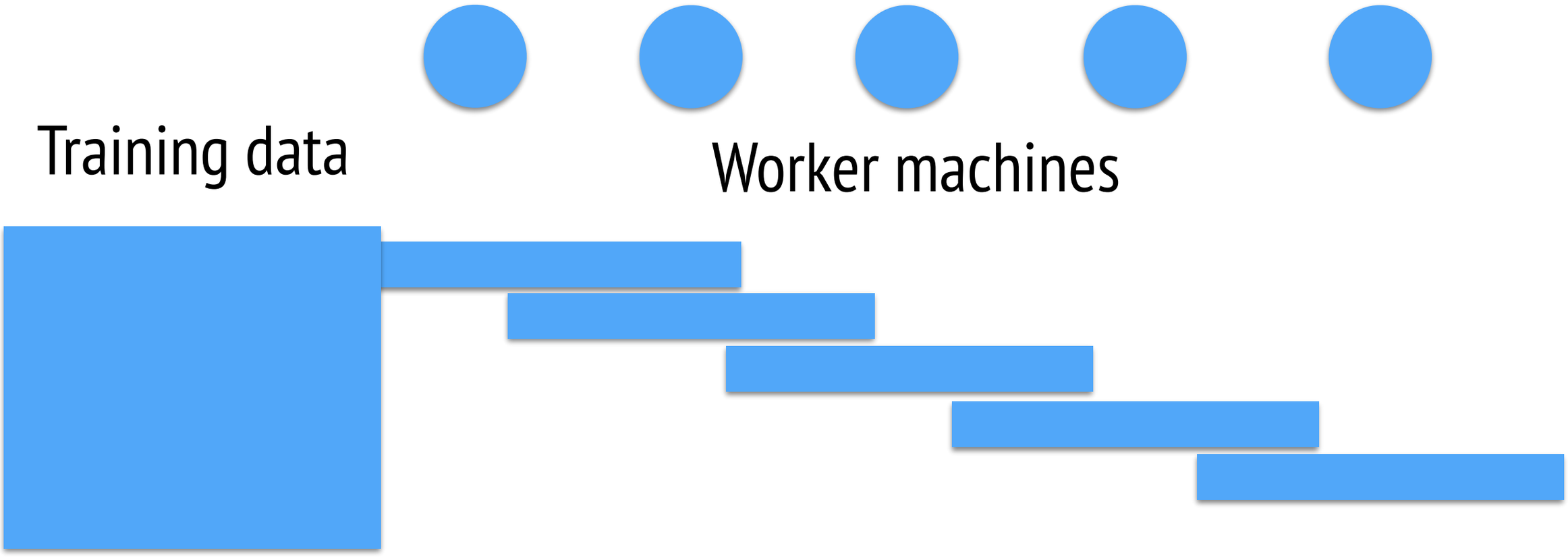


Data and model partition

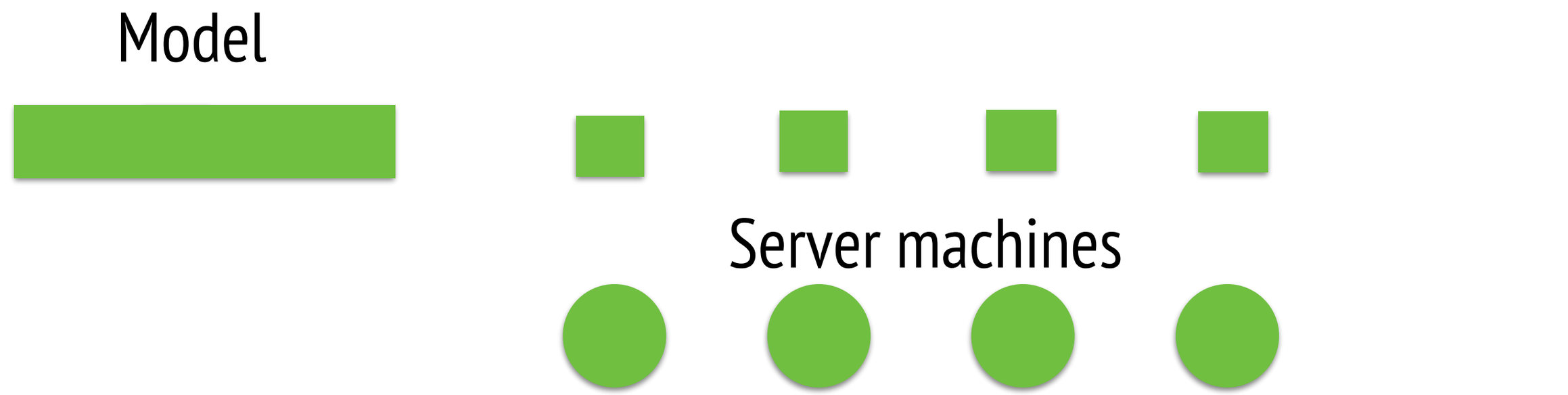
Model

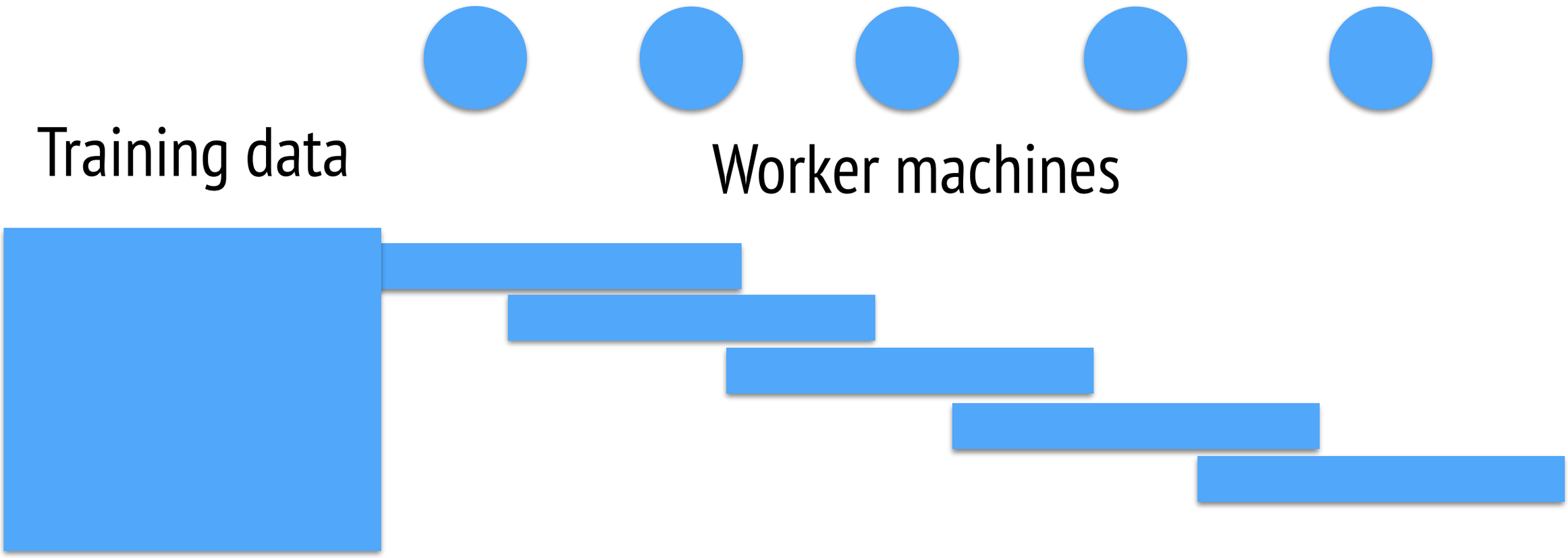
Training data

Worker machines

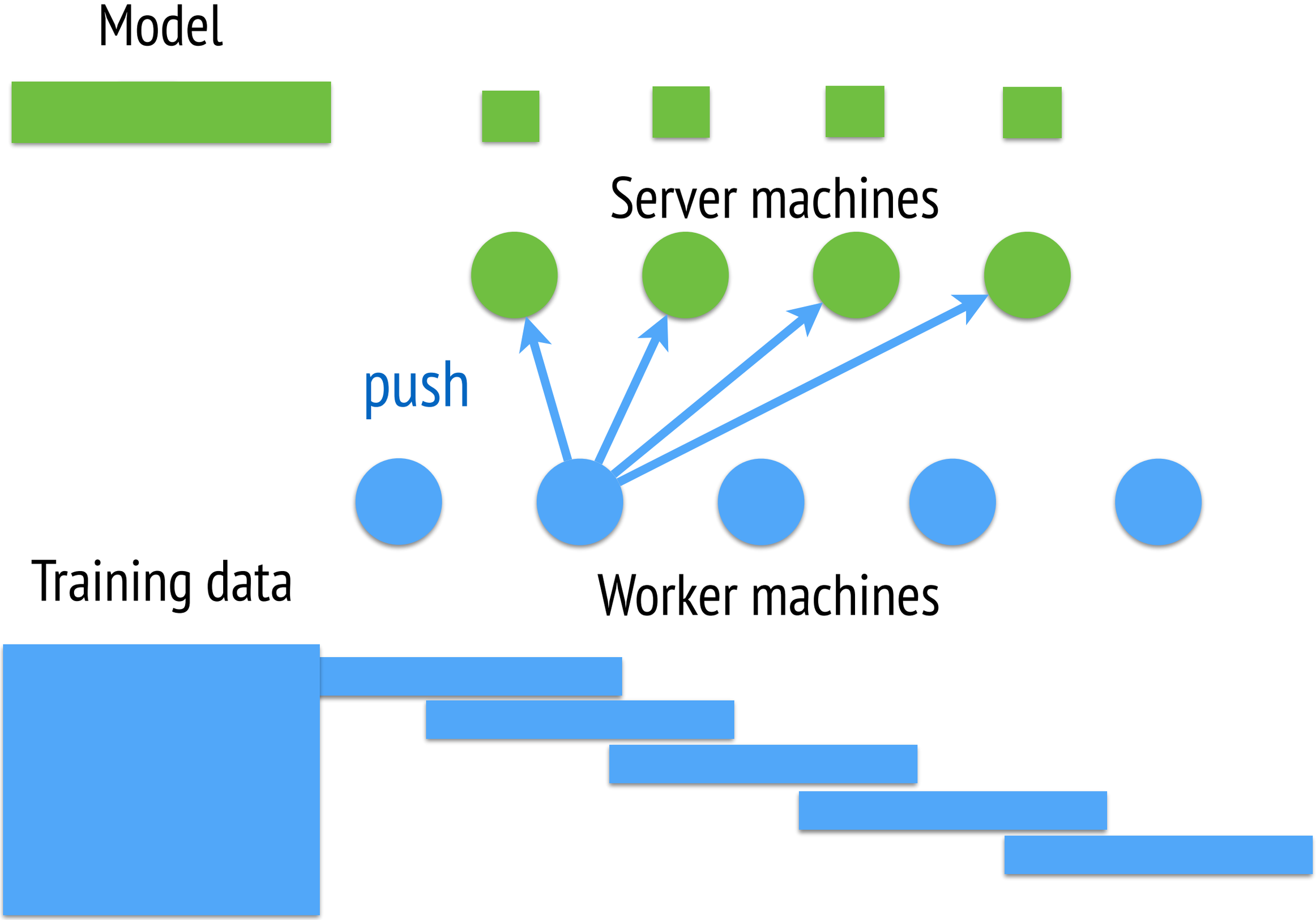


Data and model partition



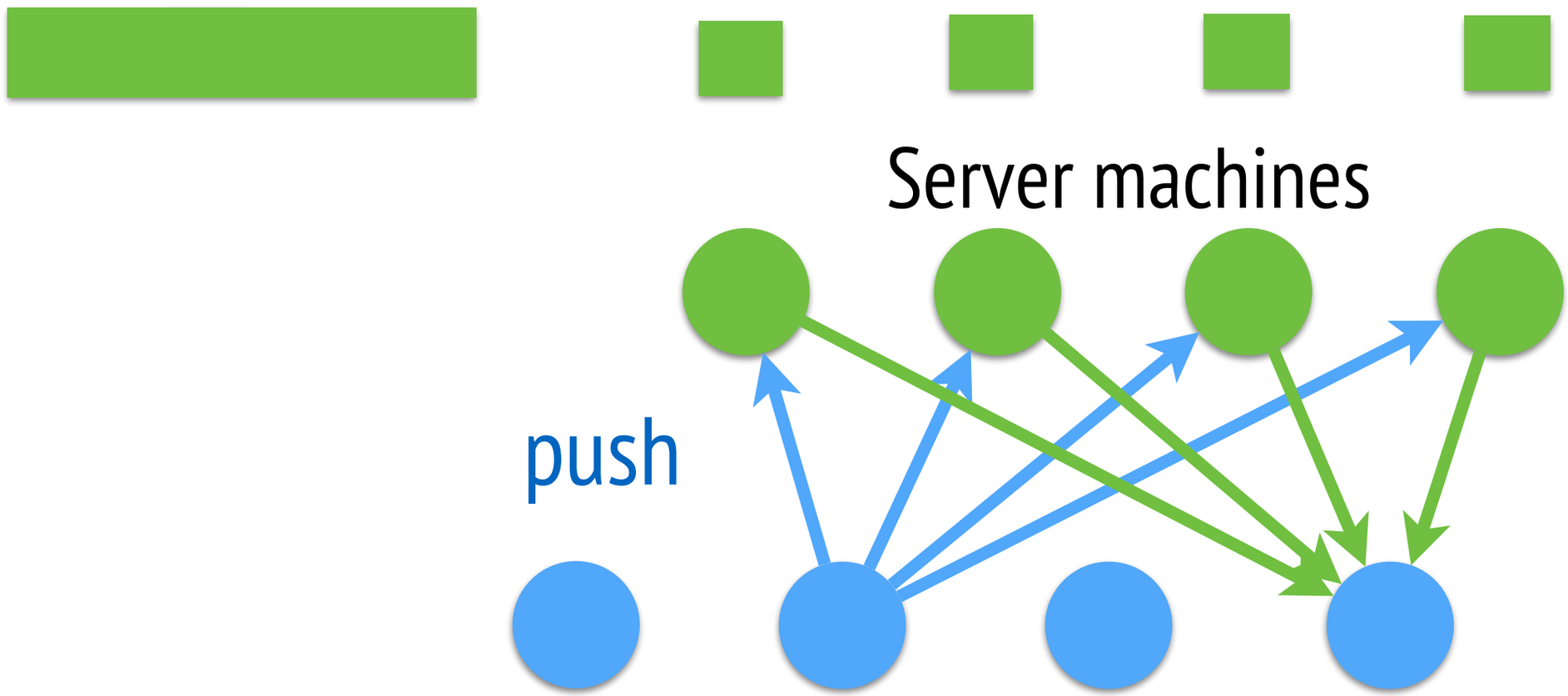


Data and model partition



Data and model partition

Model



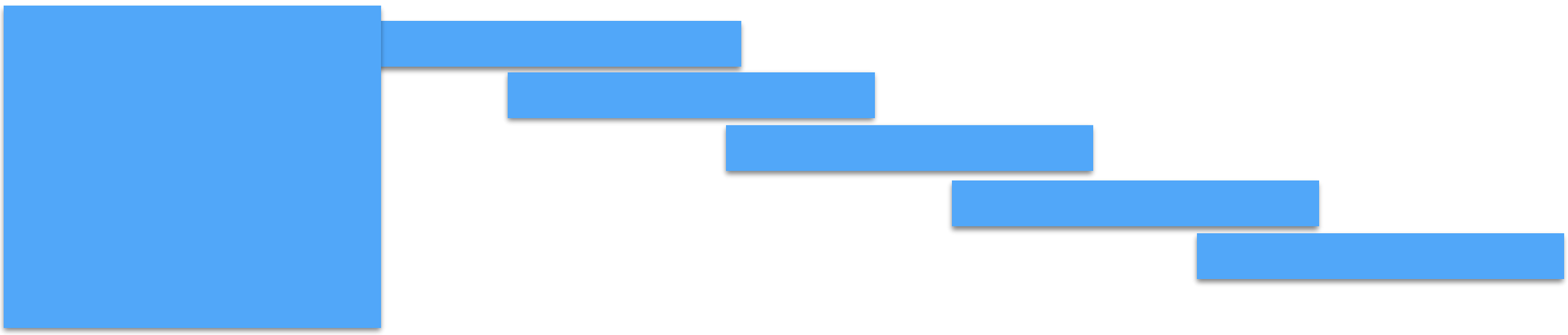
Server machines

push

pull

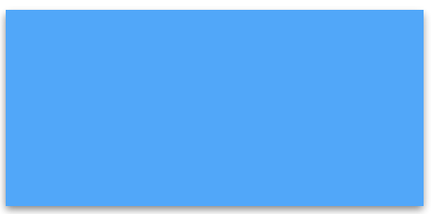
Worker machines

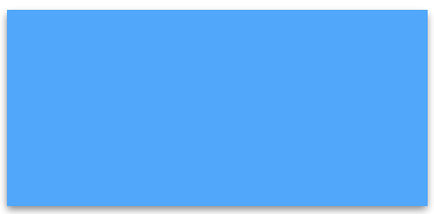
Training data



Example: distributed gradient descent

Server machines



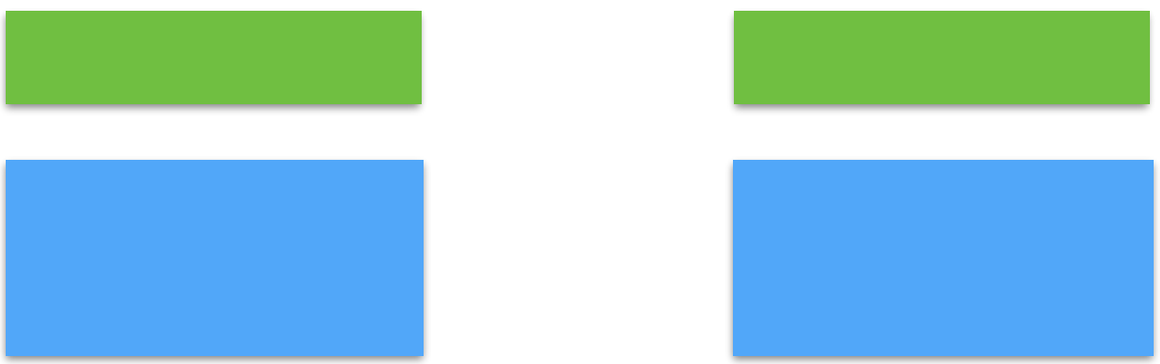


Worker machines

Example: distributed gradient descent

Workers **pull** the working set of model

Server machines



Worker machines

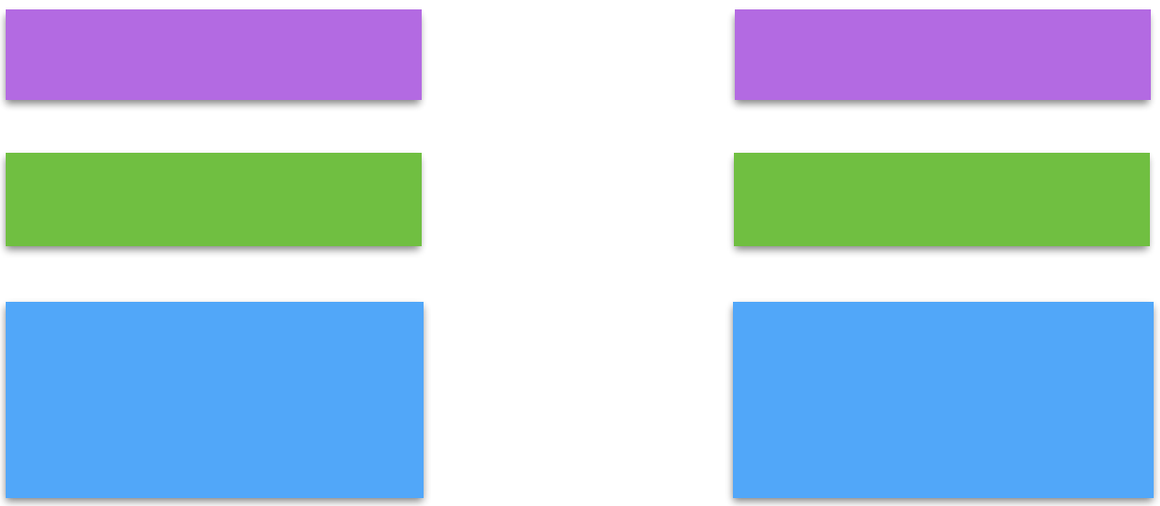
Example: distributed gradient descent

Server machines

Workers **pull** the working set of model

Iterate until stop

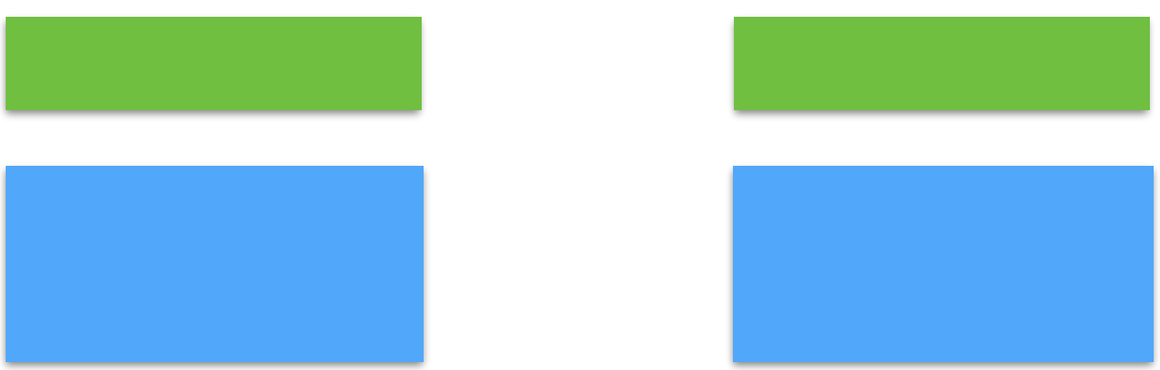
workers compute gradients



Worker machines

Example: distributed gradient descent

Server machines



Worker machines

Workers **pull** the working set of model

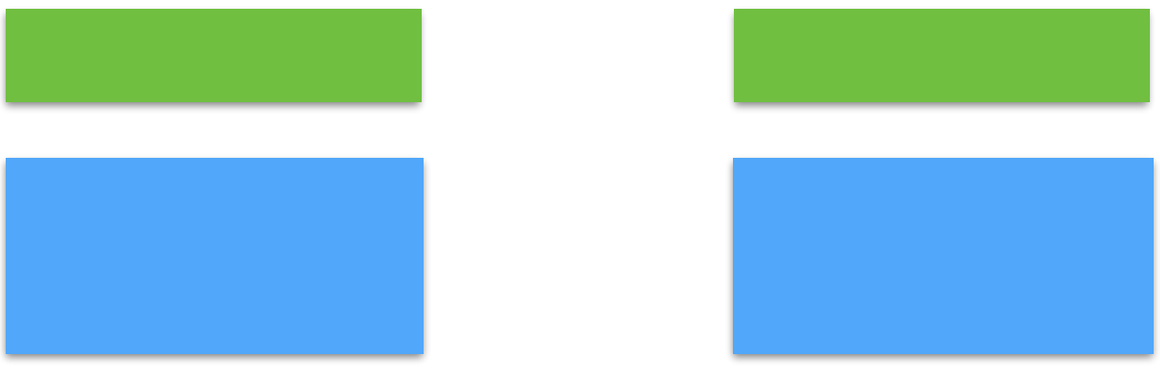
Iterate until stop

workers compute gradients

workers **push** gradients

Example: distributed gradient descent

Server machines



Worker machines

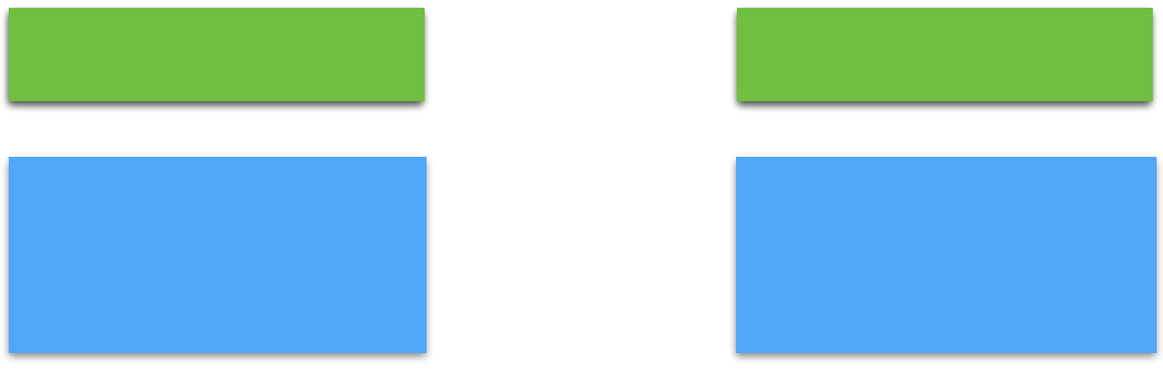
Workers **pull** the working set of model

Iterate until stop

- workers compute gradients
- workers **push** gradients
- update model

Example: distributed gradient descent

Server machines



Worker machines

Workers **pull** the working set of model

Iterate until stop

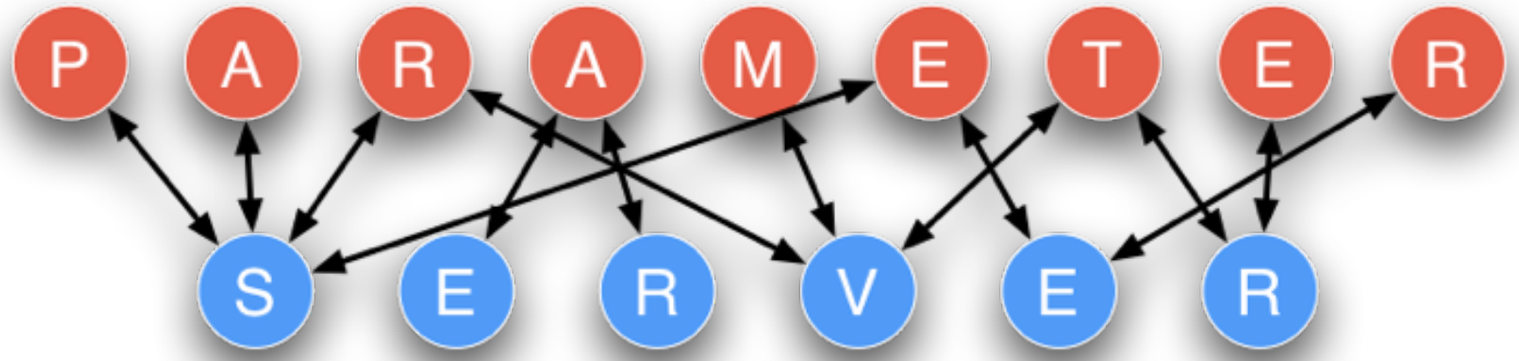
workers compute gradients

workers **push** gradients

update model

workers **pull** updated model

Industry size machine learning problems



Efficient communication

Fault tolerance

Easy to use

Evaluation

Challenges for data synchronization

Challenges for data synchronization

- Massive communication traffic
  - Frequent access to the shared model

Challenges for data synchronization

- Massive communication traffic
  - Frequent access to the shared model
- Expensive global barriers between iterations

Challenges for data synchronization

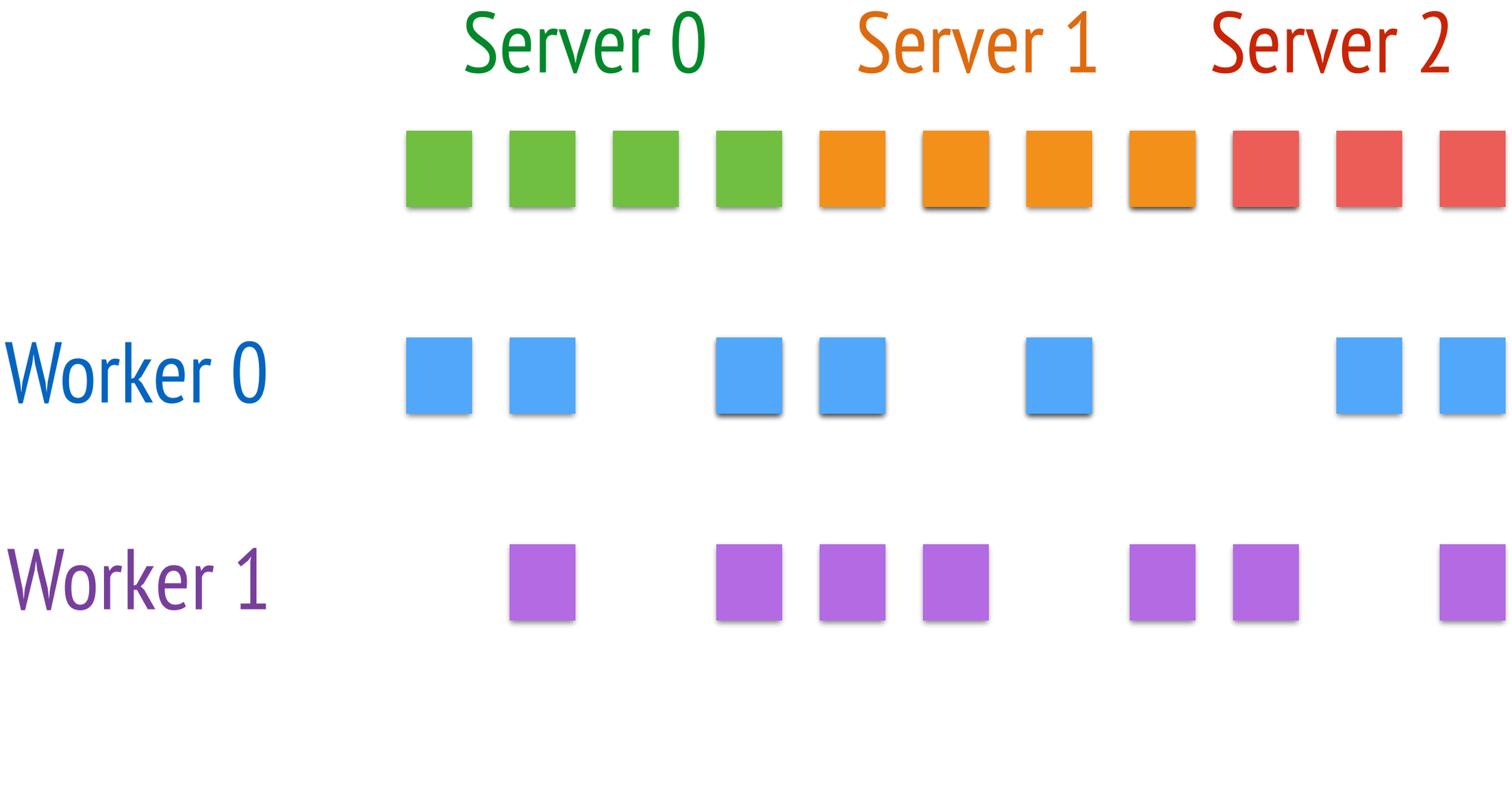
- Massive communication traffic
  - Frequent access to the shared model
- Expensive global barriers between iterations
- Our solution is to relax the data consistency in two levels
  - Task dependency graph
  - User-defined filters

Range-based Push & Pull

- Programmer: fit most machine learning algorithms
- System: communication are batched

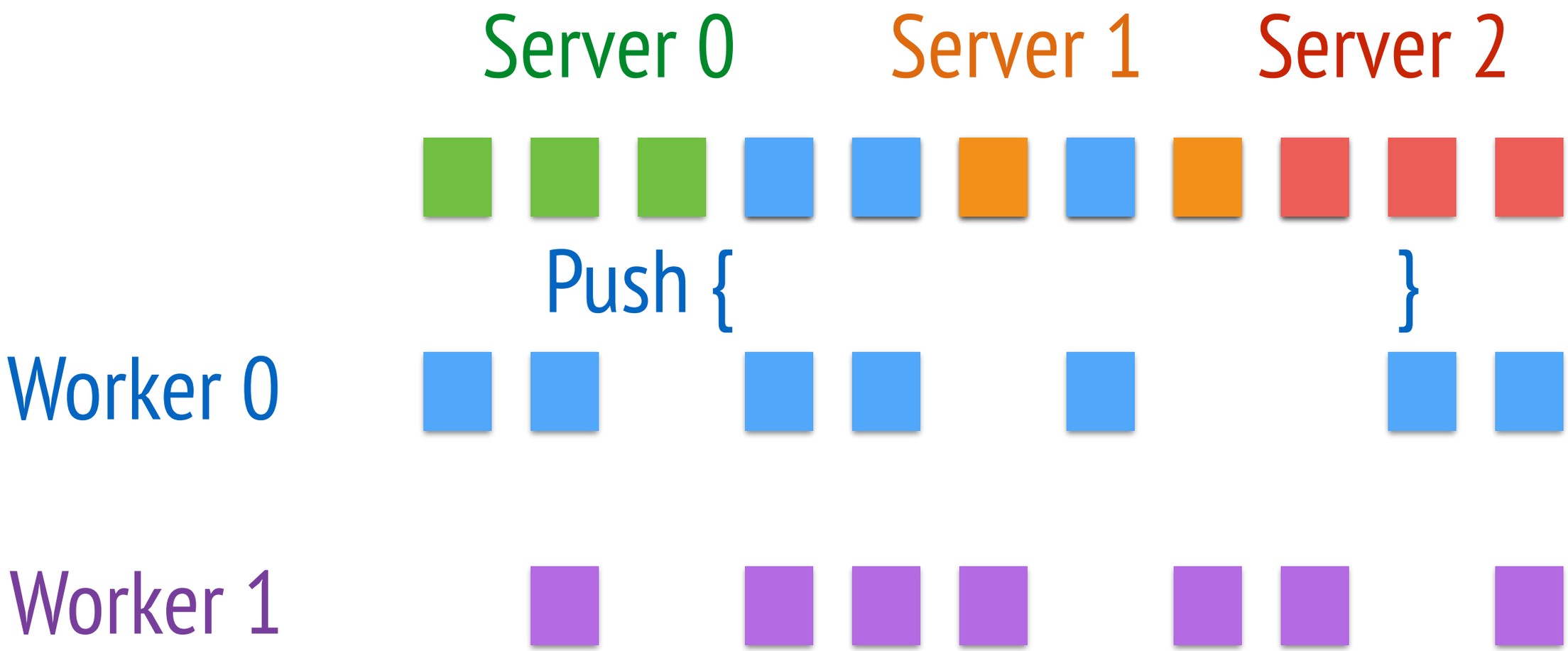
Range-based Push & Pull

- Programmer: fit most machine learning algorithms
- System: communication are batched



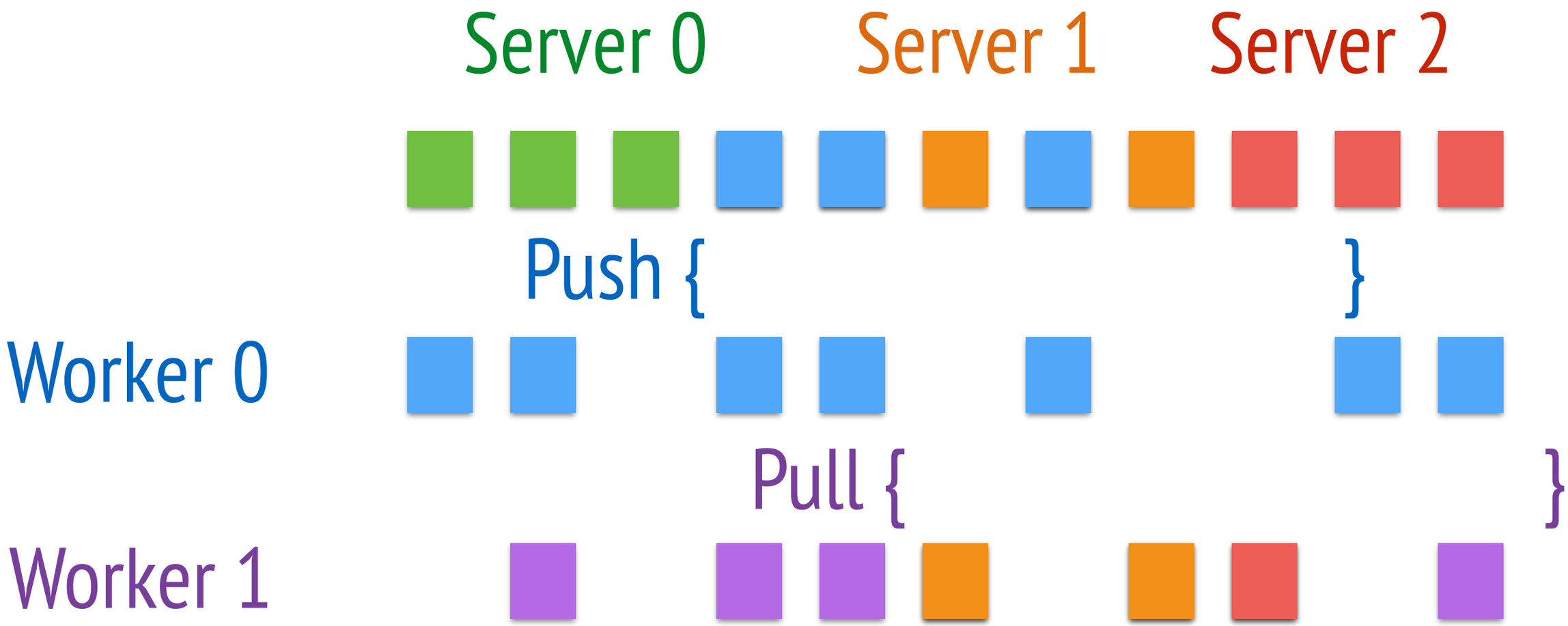
Range-based Push & Pull

- Programmer: fit most machine learning algorithms
- System: communication are batched



Range-based Push & Pull

- Programmer: fit most machine learning algorithms
- System: communication are batched



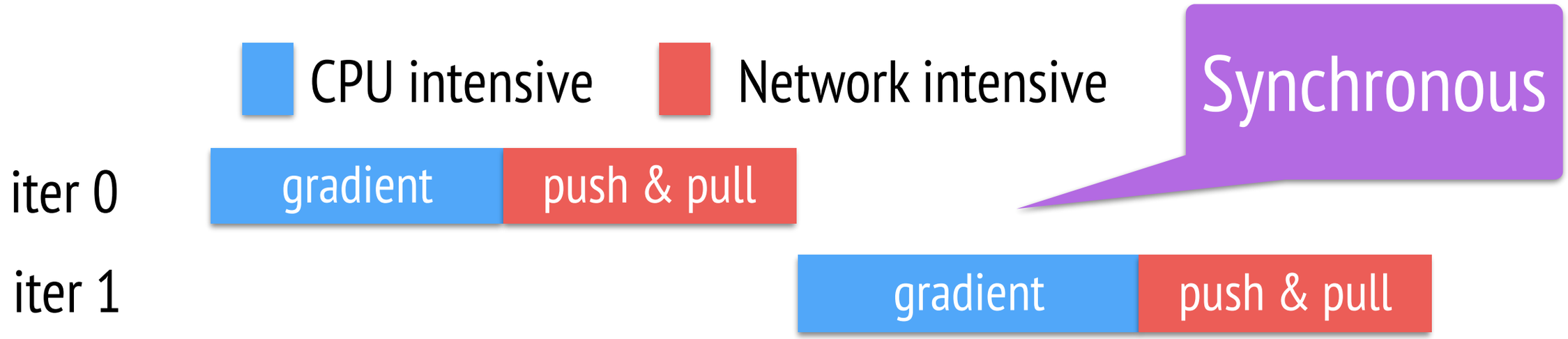
Task

Task

- A push / pull / user defined function (an iteration)

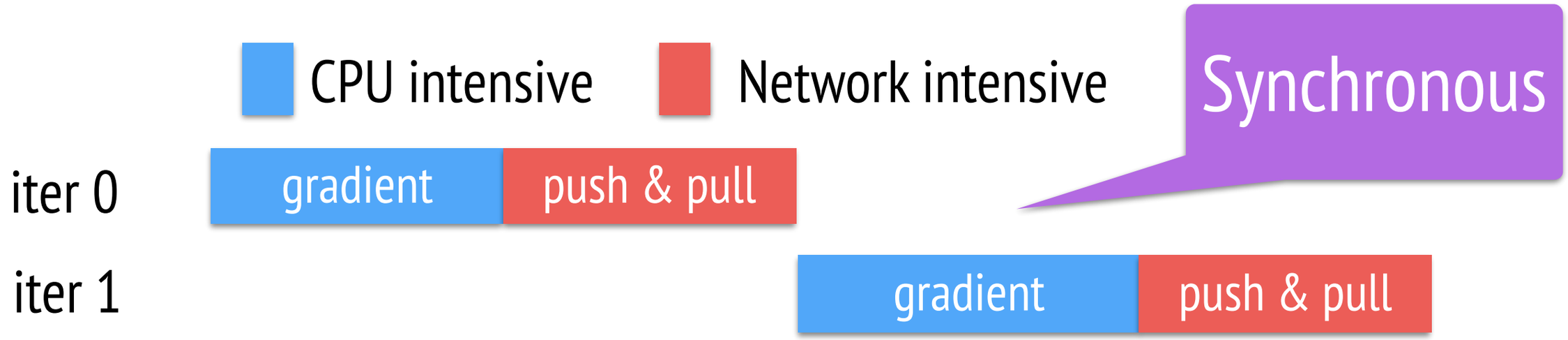
Task

- A push / pull / user defined function (an iteration)
- “execute-after-finished” dependency

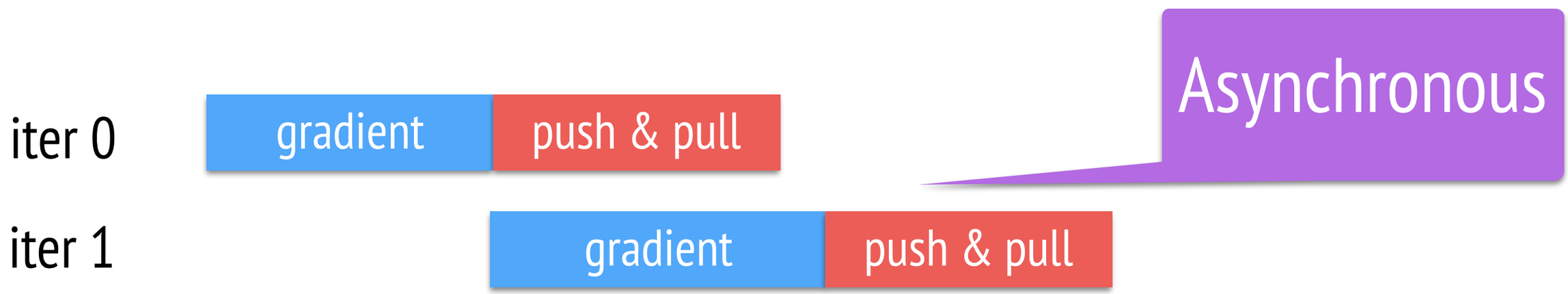


Task

- A push / pull / user defined function (an iteration)
- “execute-after-finished” dependency

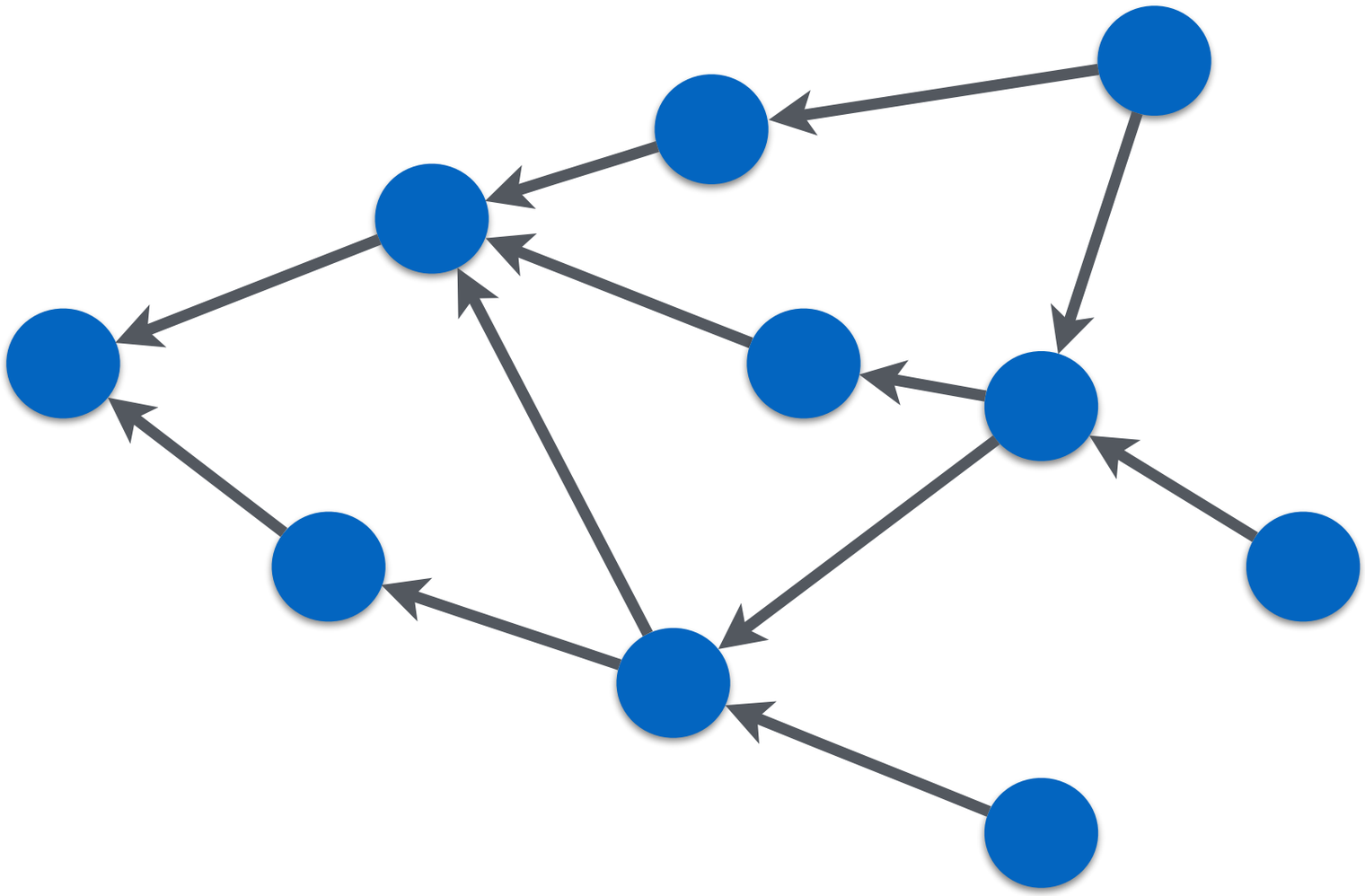


- Executed asynchronously



Data consistency

- Algorithm efficiency vs. system performance
- Flexible models via task dependency graph



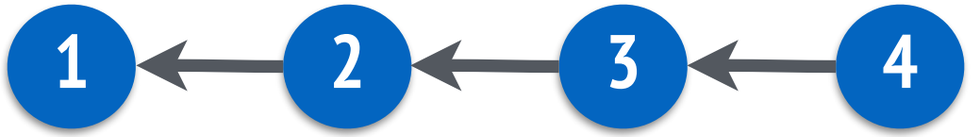
Data consistency

- Algorithm efficiency vs. system performance
- Flexible models via task dependency graph
- Some examples [Bertsekas 89]

Data consistency

- Algorithm efficiency vs. system performance
- Flexible models via task dependency graph
- Some examples [Bertsekas 89]

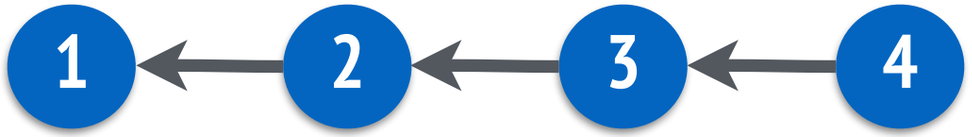
Sequential / BSP



Data consistency

- Algorithm efficiency vs. system performance
- Flexible models via task dependency graph
- Some examples [Bertsekas 89]

Sequential / BSP

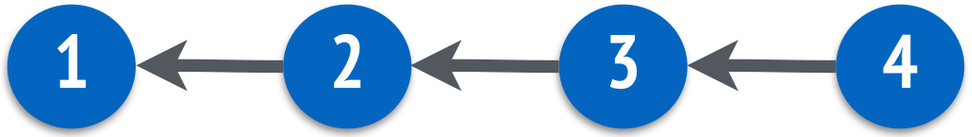


Eventual /
Total asynchronous
[Smola 10]

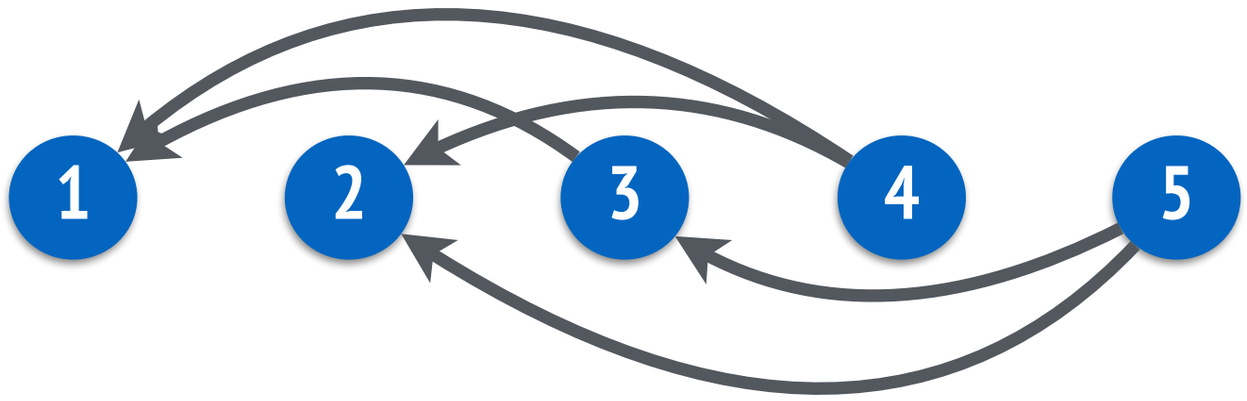
Data consistency

- Algorithm efficiency vs. system performance
- Flexible models via task dependency graph
- Some examples [Bertsekas 89]

Sequential / BSP



Bounded delay / SSP
[Langford 09, Cipar 13]



Eventual /
Total asynchronous
[Smola 10]

Time to the Same Convergence Criteria

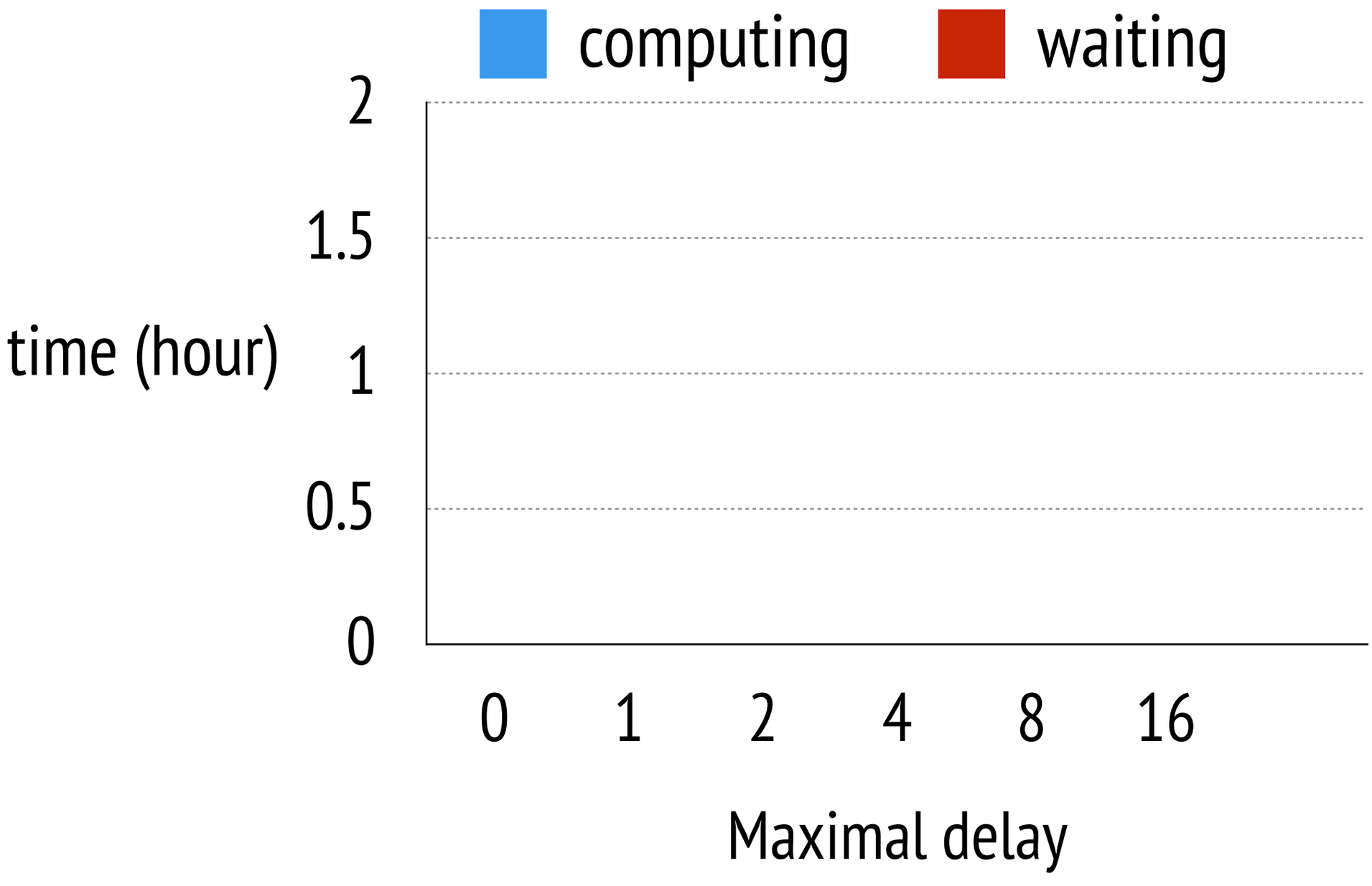
Ads click prediction

636TB data, 1TB model, and 1000 machines

Time to the Same Convergence Criteria

Ads click prediction

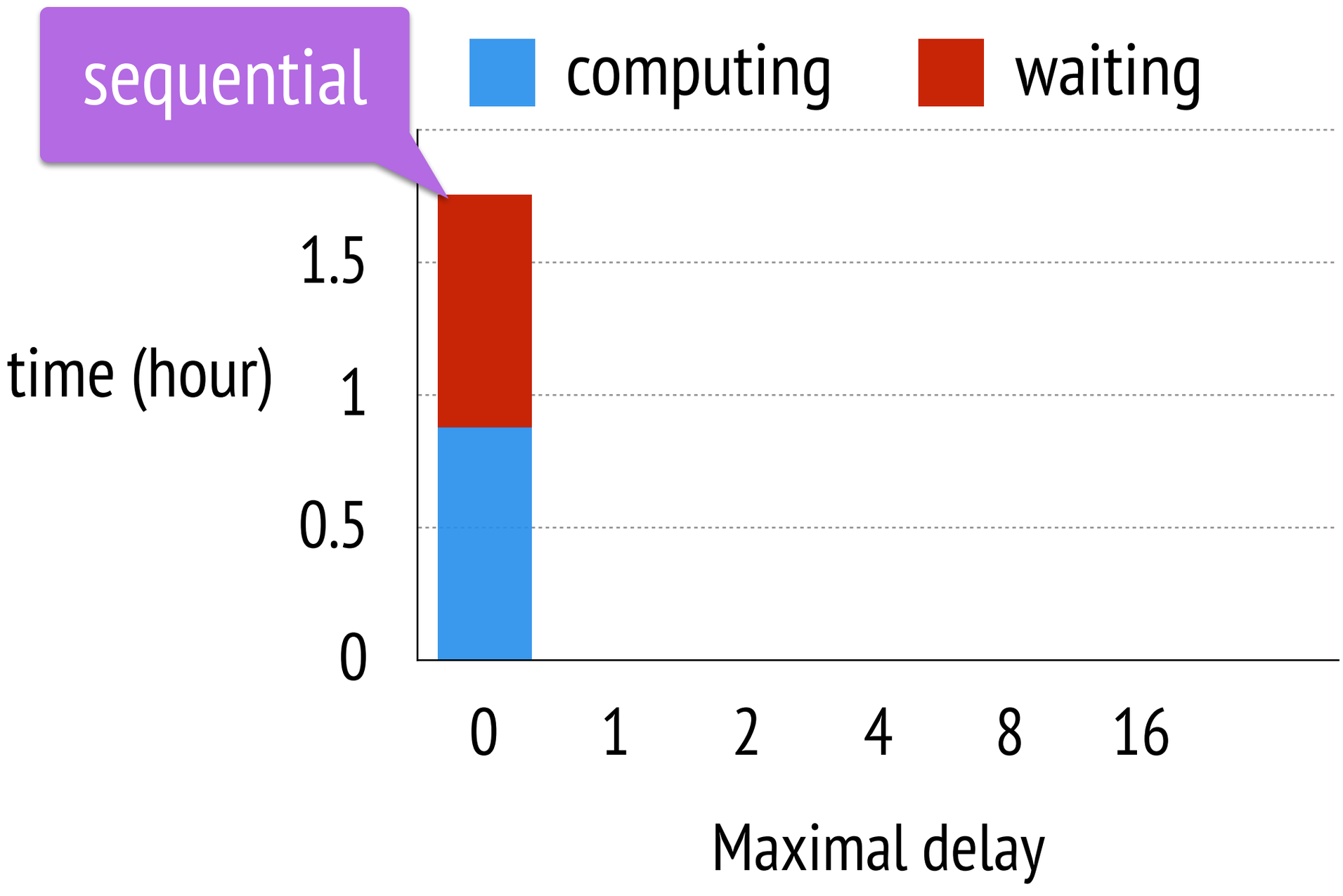
636TB data, 1TB model, and 1000 machines



Time to the Same Convergence Criteria

Ads click prediction

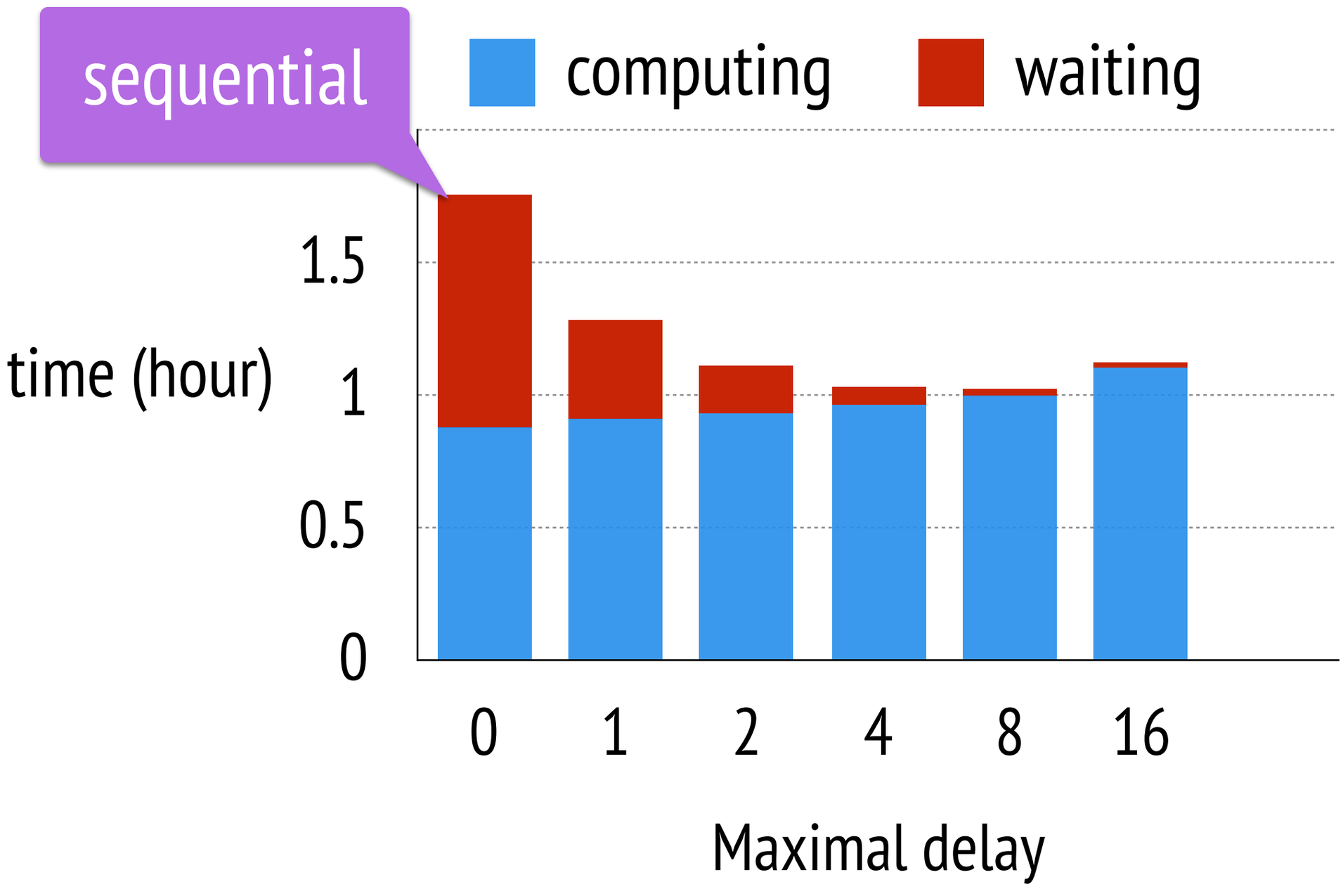
636TB data, 1TB model, and 1000 machines



Time to the Same Convergence Criteria

Ads click prediction

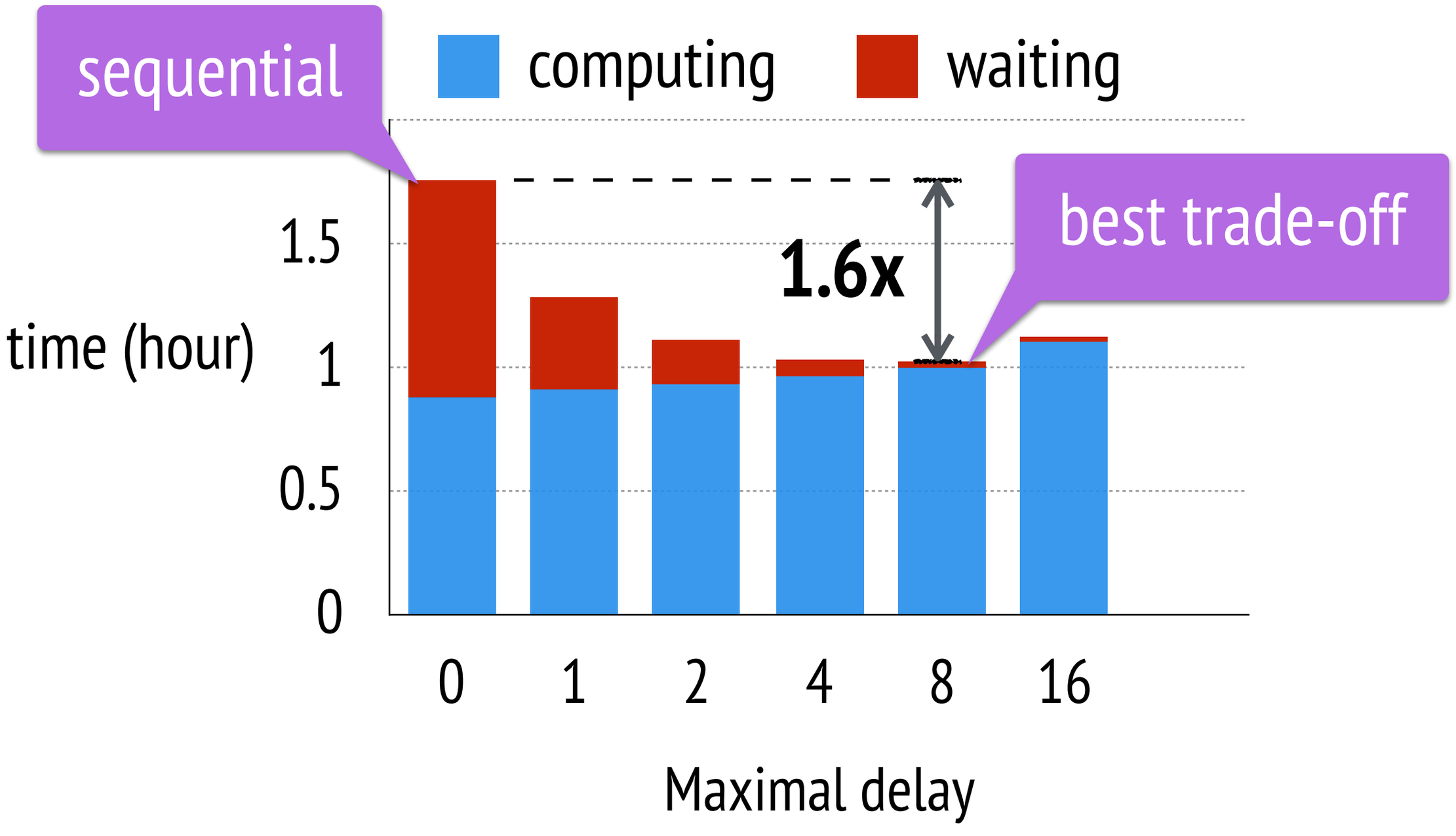
636TB data, 1TB model, and 1000 machines



Time to the Same Convergence Criteria

Ads click prediction

636TB data, 1TB model, and 1000 machines



Filters: control data consistency in a task

Filters: control data consistency in a task

Filters: control data consistency in a task

Filters: control data consistency in a task

- A filter could
  - Selectively filter (key, value) pairs
  - Transform data

Filters: control data consistency in a task

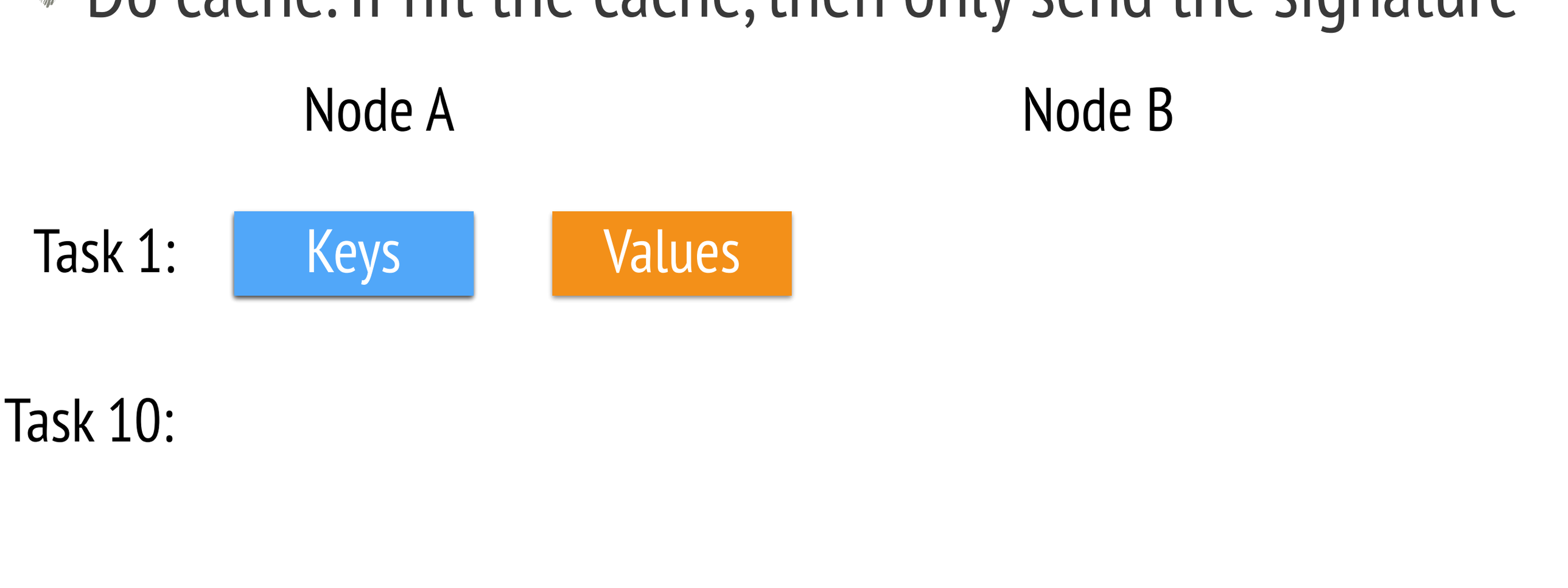
- A filter could
  - Selectively filter (key, value) pairs
  - Transform data
- Examples
  - Significantly modified filter: send entry e only if $|\Delta e| > \epsilon$
  - Randomly skip filter: randomly skip entries

Key caching

- The keys sent in a task could be resent by another task
  - Push the gradients, then pull the updated weights
  - Keys are fixed between iterations
- Do cache. If hit the cache, then only send the signature

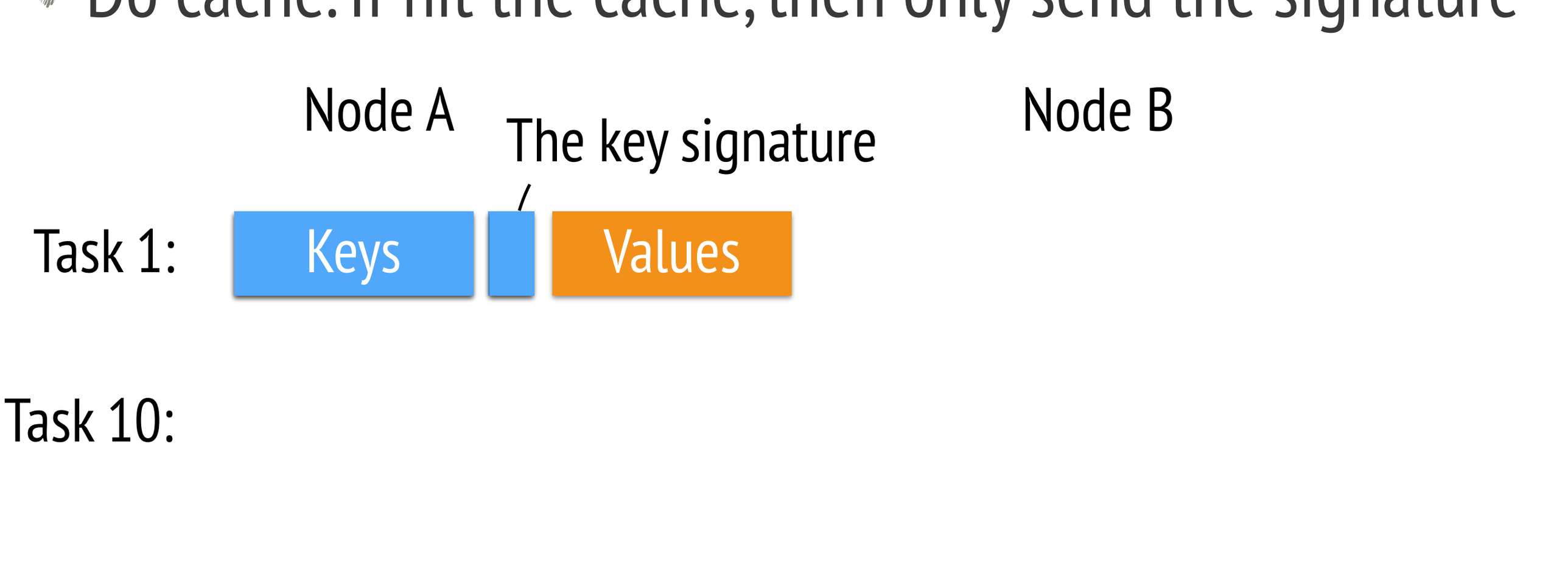
Key caching

- The keys sent in a task could be resent by another task
  - Push the gradients, then pull the updated weights
  - Keys are fixed between iterations
- Do cache. If hit the cache, then only send the signature



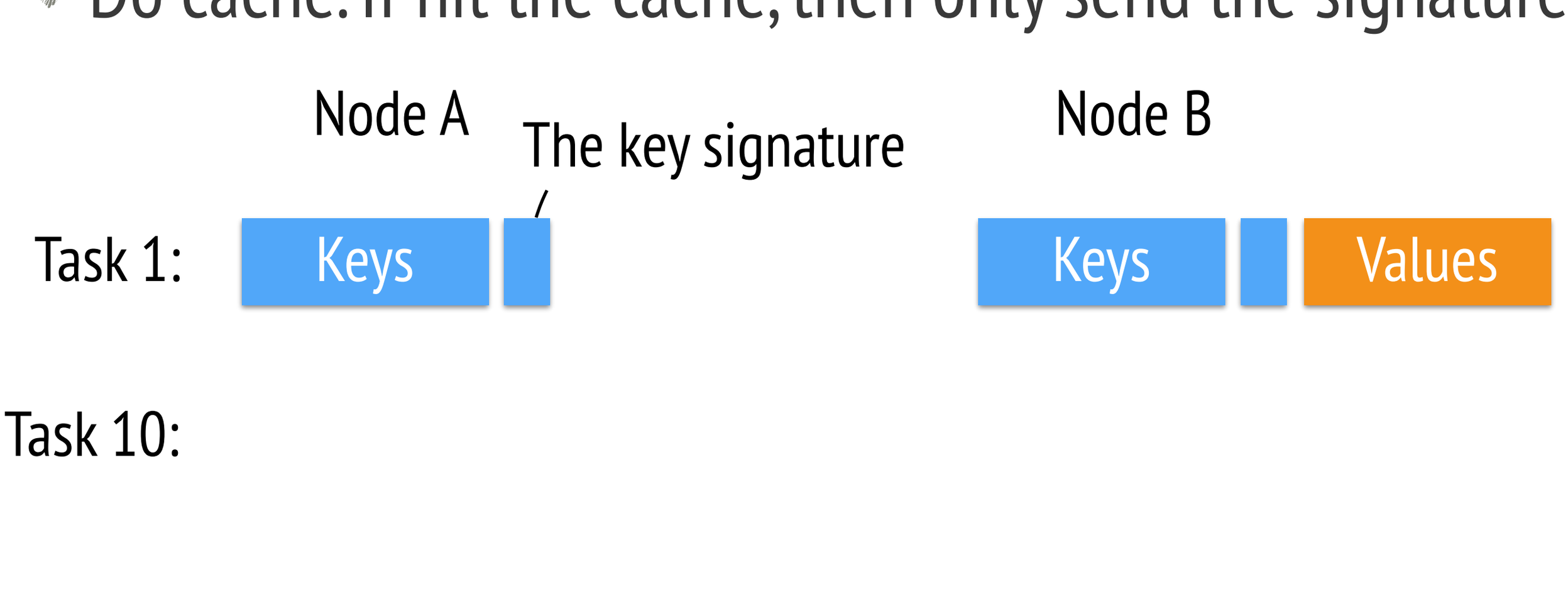
Key caching

- The keys sent in a task could be resent by another task
  - Push the gradients, then pull the updated weights
  - Keys are fixed between iterations
- Do cache. If hit the cache, then only send the signature



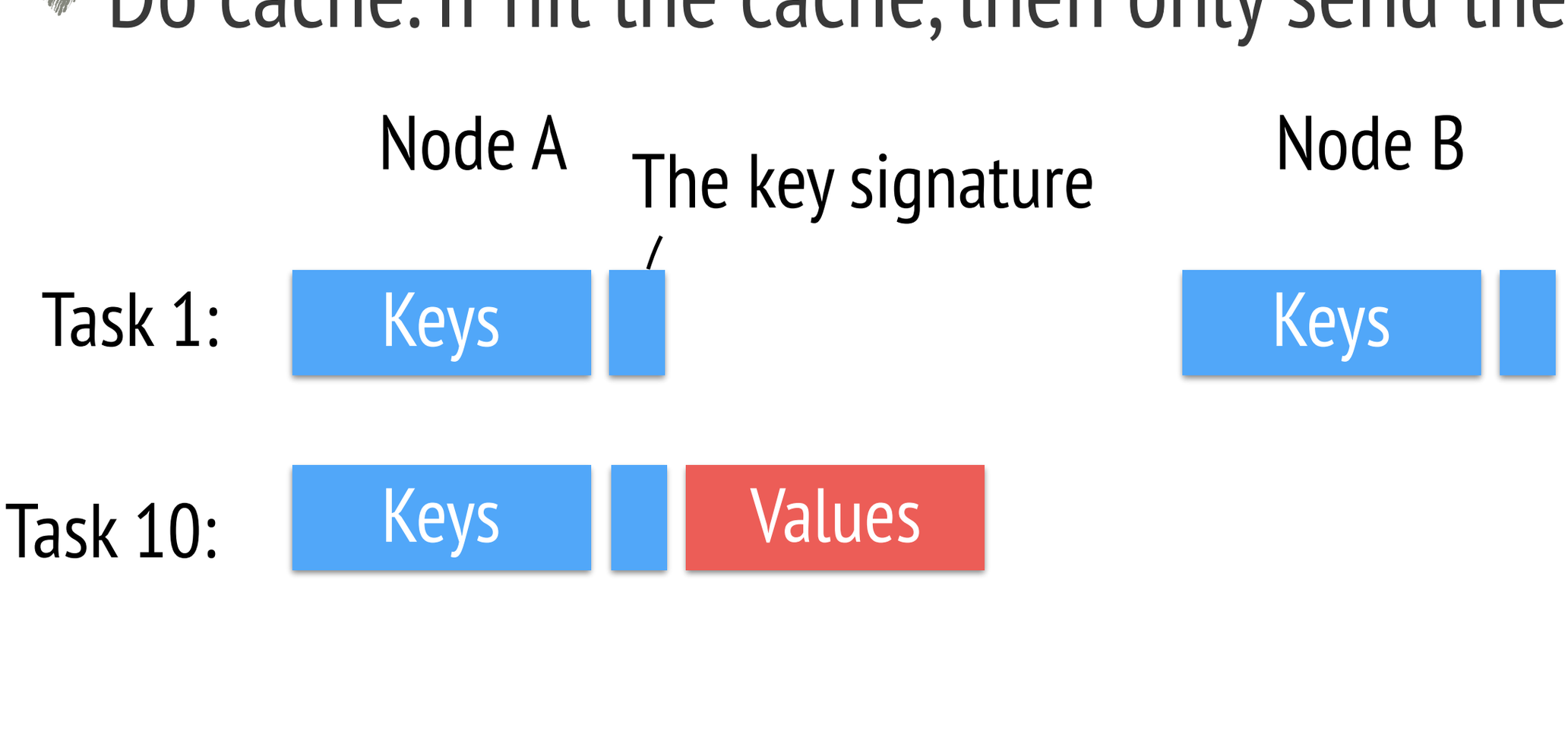
Key caching

- The keys sent in a task could be resent by another task
  - Push the gradients, then pull the updated weights
  - Keys are fixed between iterations
- Do cache. If hit the cache, then only send the signature



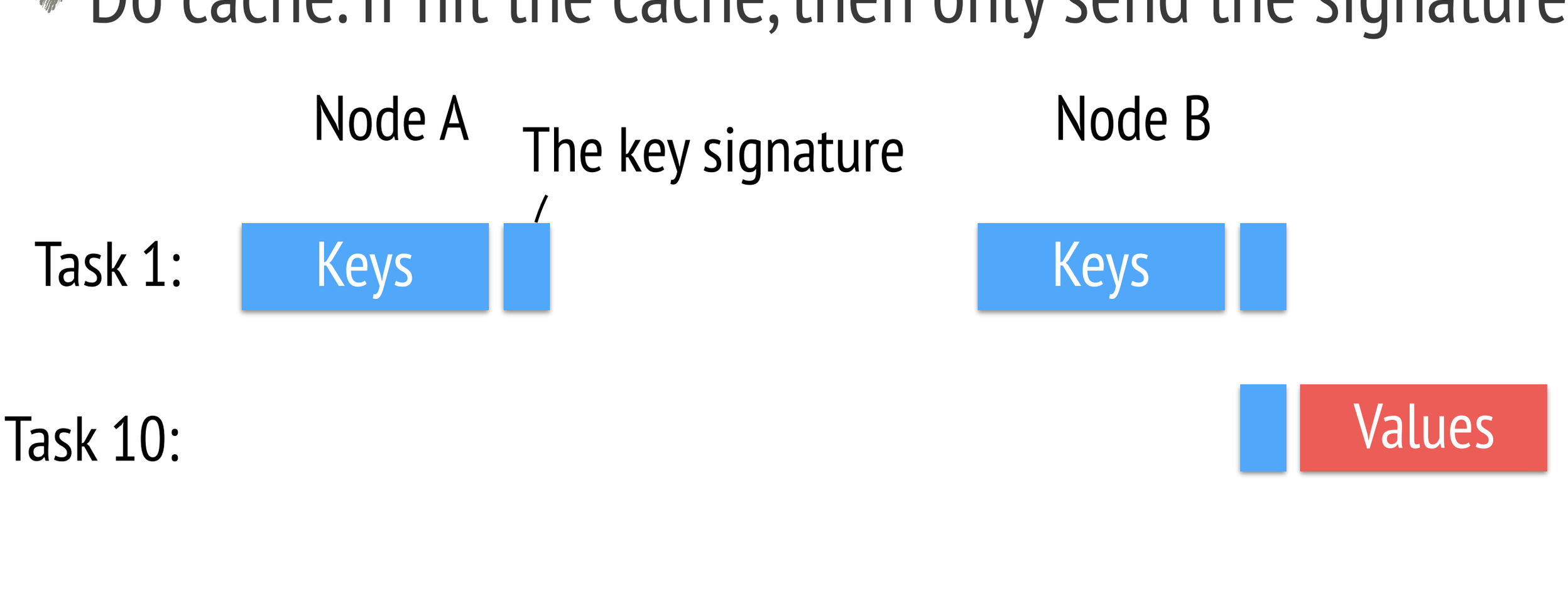
Key caching

- The keys sent in a task could be resent by another task
  - Push the gradients, then pull the updated weights
  - Keys are fixed between iterations
- Do cache. If hit the cache, then only send the signature



Key caching

- The keys sent in a task could be resent by another task
  - Push the gradients, then pull the updated weights
  - Keys are fixed between iterations
- Do cache. If hit the cache, then only send the signature



Data Compressing

- Values are often compressible numbers
  - 0s, small integers, numbers with more than enough precision
- Lossness compression algorithms: LZ, LZR, …
- Variable-length integers
- Lossy compression such as fixed-point encoding

KKT Filter

KKT Filter

- Only send the subset of gradient that is likely to affect the model

KKT Filter

- Only send the subset of gradient that is likely to affect the model
- Assume we use sparse penalty $\lambda|w|_1$, then according to the KKT condition:

iteration i: w[key] = 0 and |gradient[key]| < λ

iteration i+1: w[key] = 0

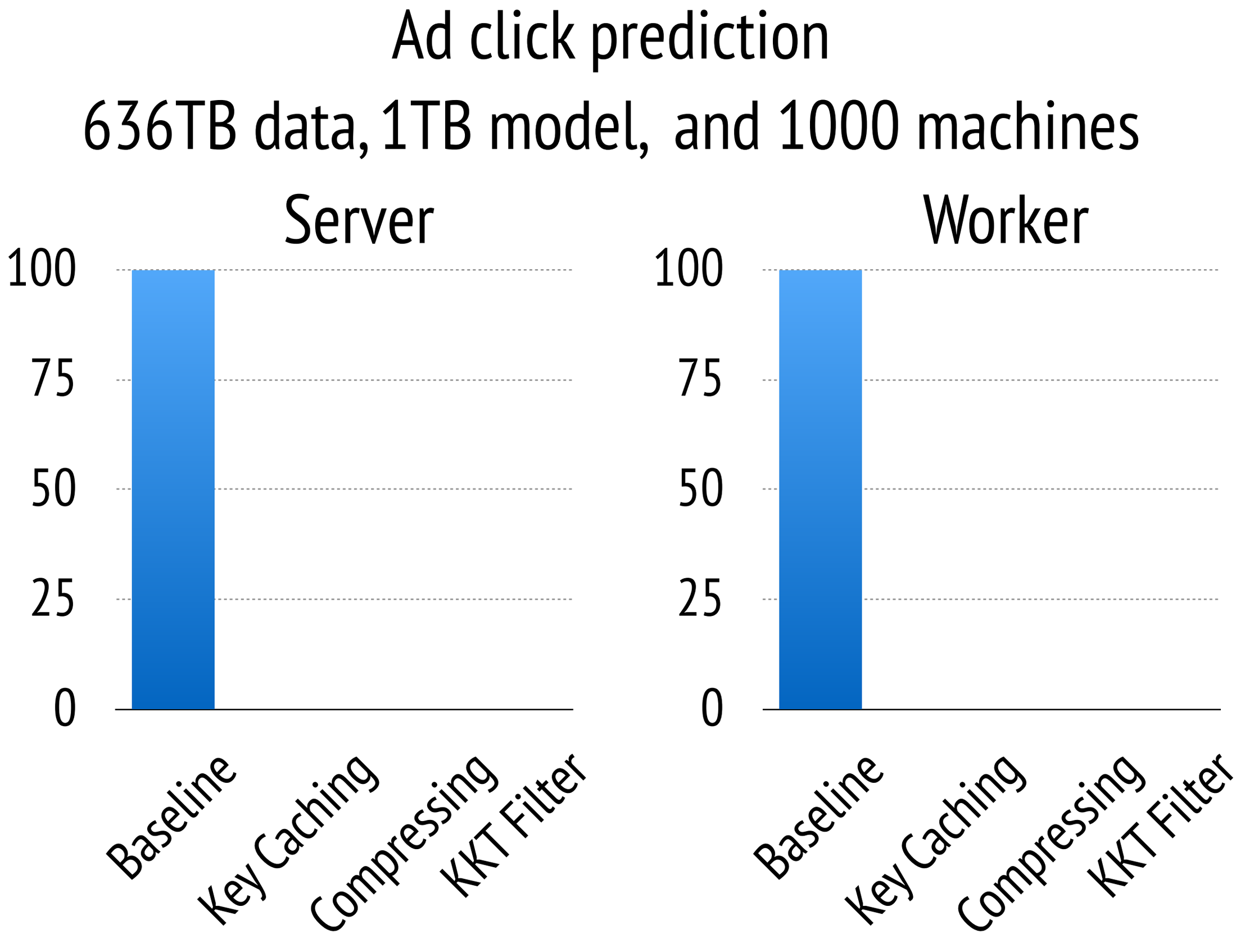
Traffic Reduction by Filters

Ad click prediction

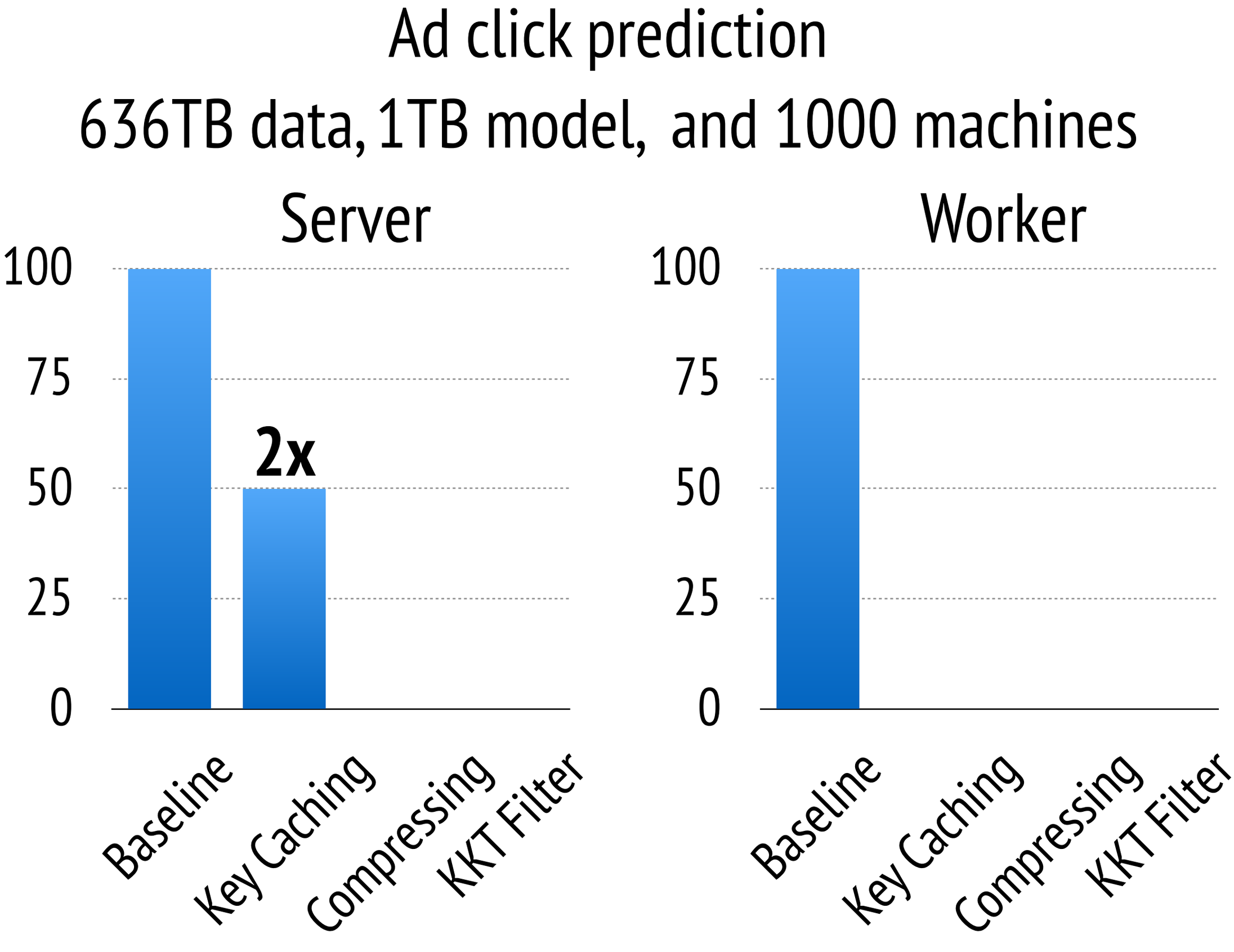
636TB data, 1TB model, and 1000 machines

Server Worker

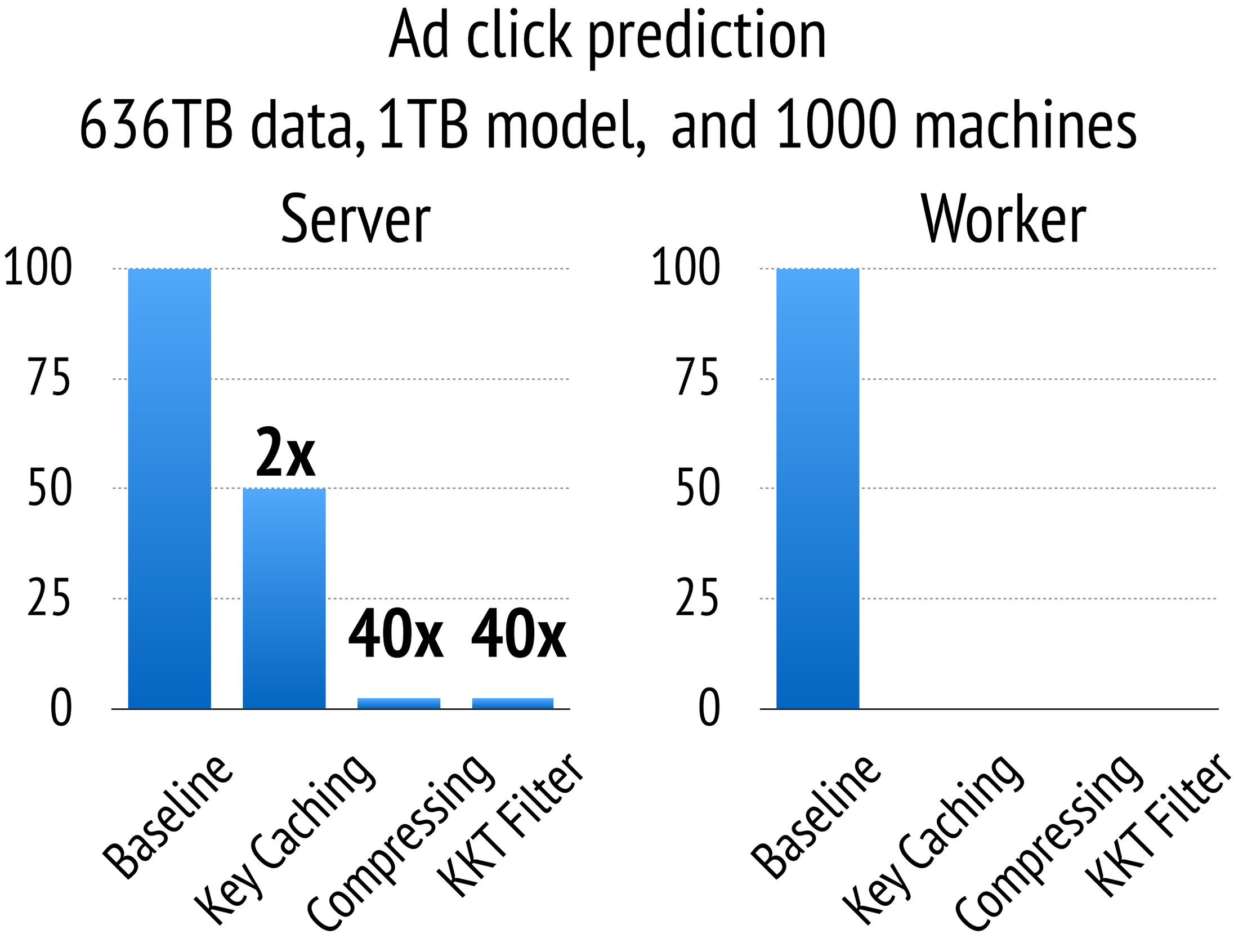
Traffic Reduction by Filters



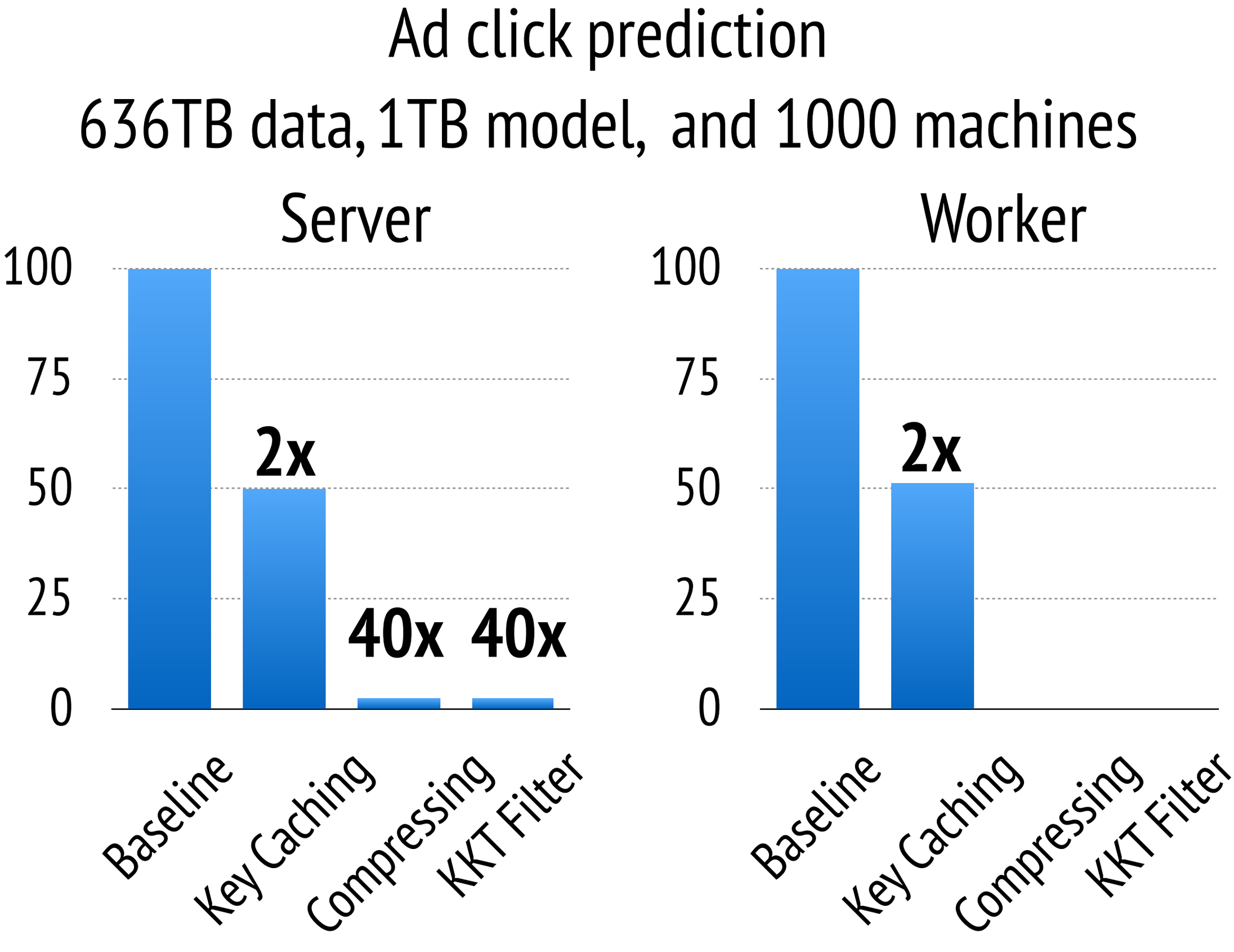
Traffic Reduction by Filters



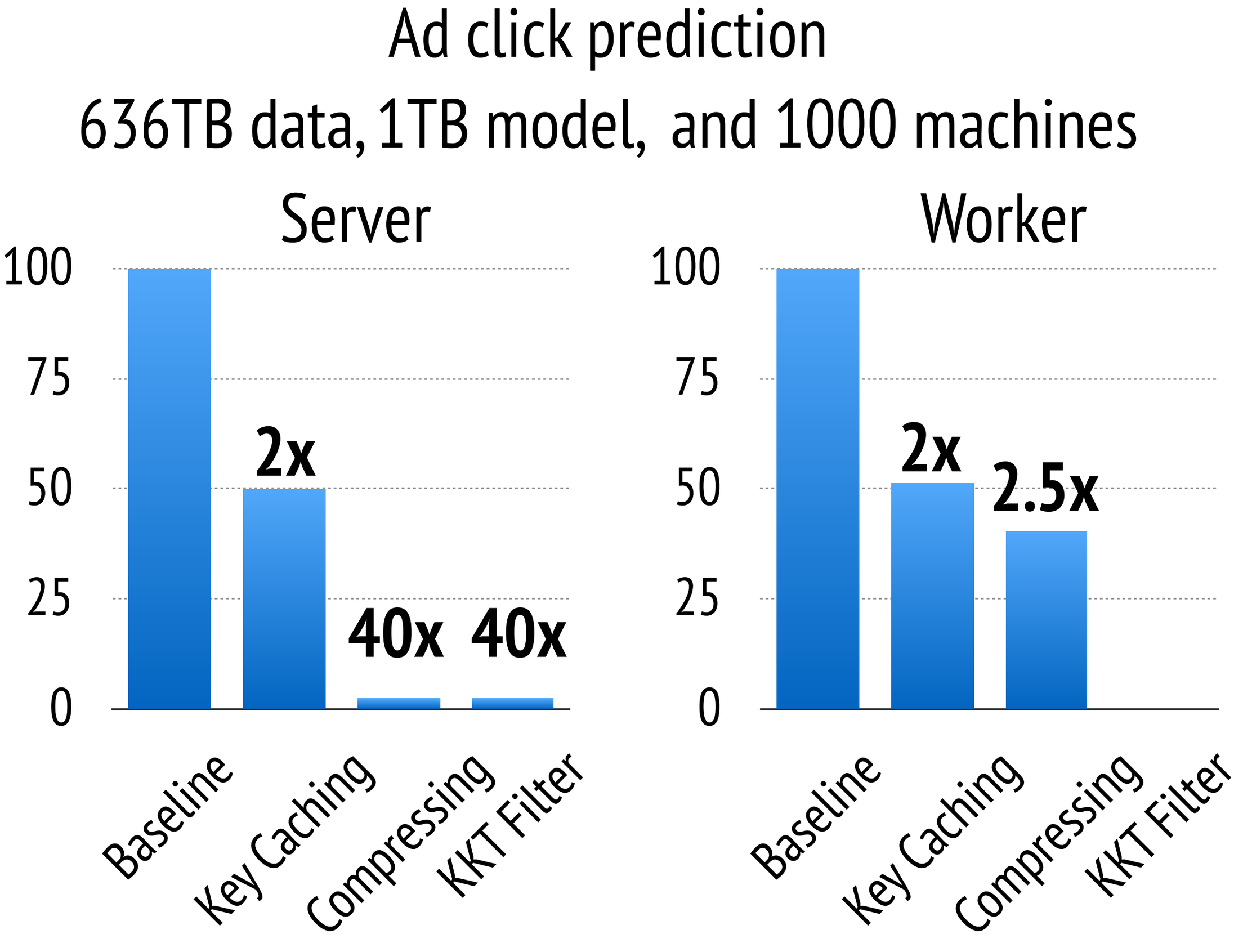
Traffic Reduction by Filters



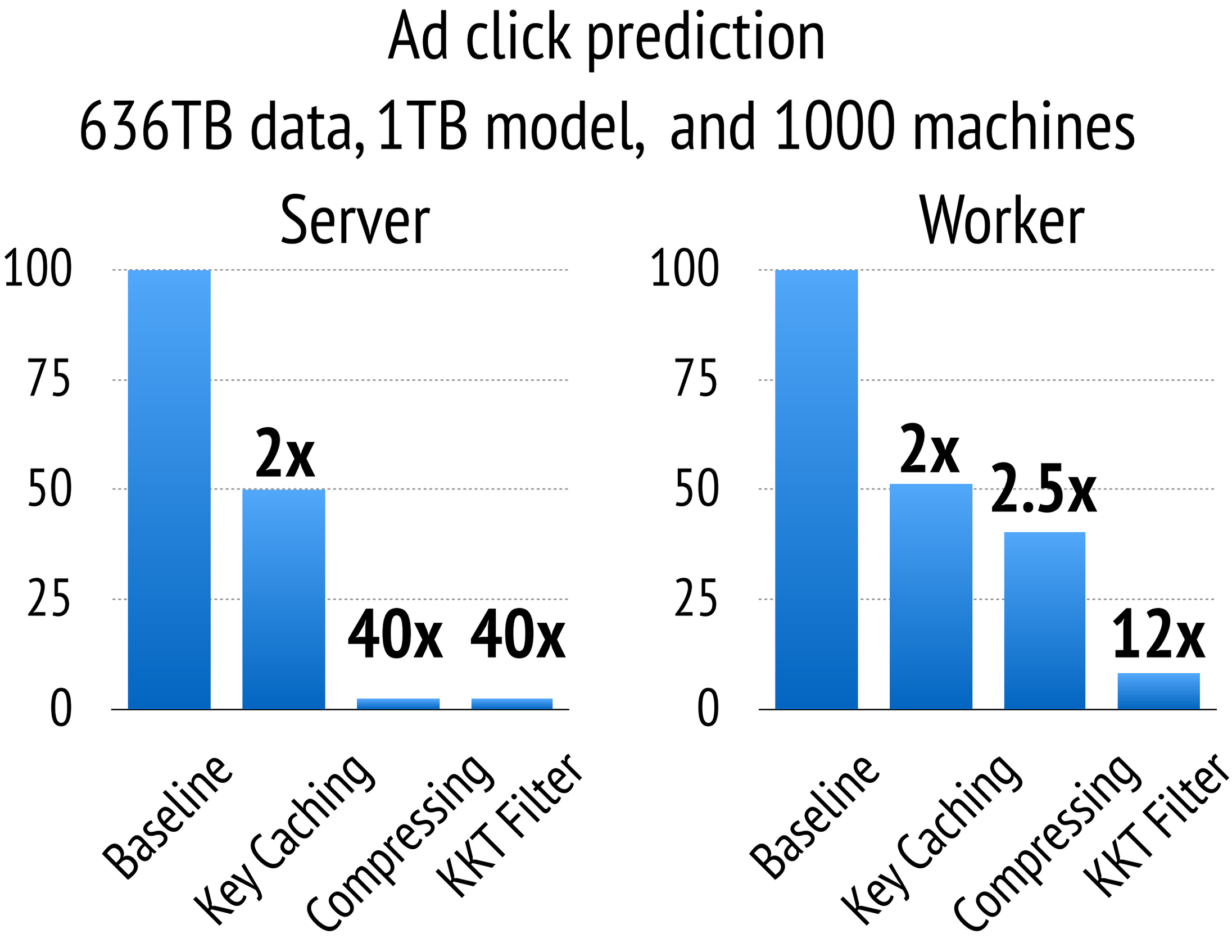
Traffic Reduction by Filters



Traffic Reduction by Filters



Traffic Reduction by Filters



The relax consistency guarantees convergence?

The relax consistency guarantees convergence?

- Short answer: Yes

The relax consistency guarantees convergence?

- Short answer: Yes
- An algorithm often has a tunable step size, a small value guarantees convergence

The relax consistency guarantees convergence?

- Short answer: Yes
- An algorithm often has a tunable step size, a small value guarantees convergence
- **Theorem.** Assume
  - maximal τ delays
  - properly stack previous filters

The relax consistency guarantees convergence?

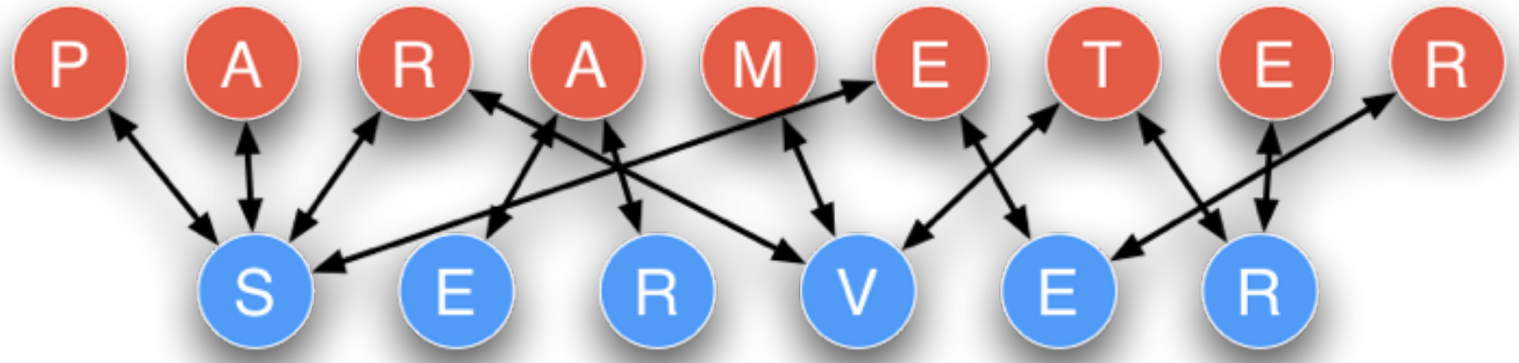
- Short answer: Yes
- An algorithm often has a tunable step size, a small value guarantees convergence
- Theorem. Assume
  - maximal τ delays
  - properly stack previous filters
- Then a large group of algorithms convergence if

$$\text{step size} <= \mathcal{O}\left(\frac{1}{\Delta_1 + \tau\Delta_2}\right)$$

data correlation within a task Δ_1 / between tasks Δ_2

Industry size machine learning problems ✓

Efficient communication ✓



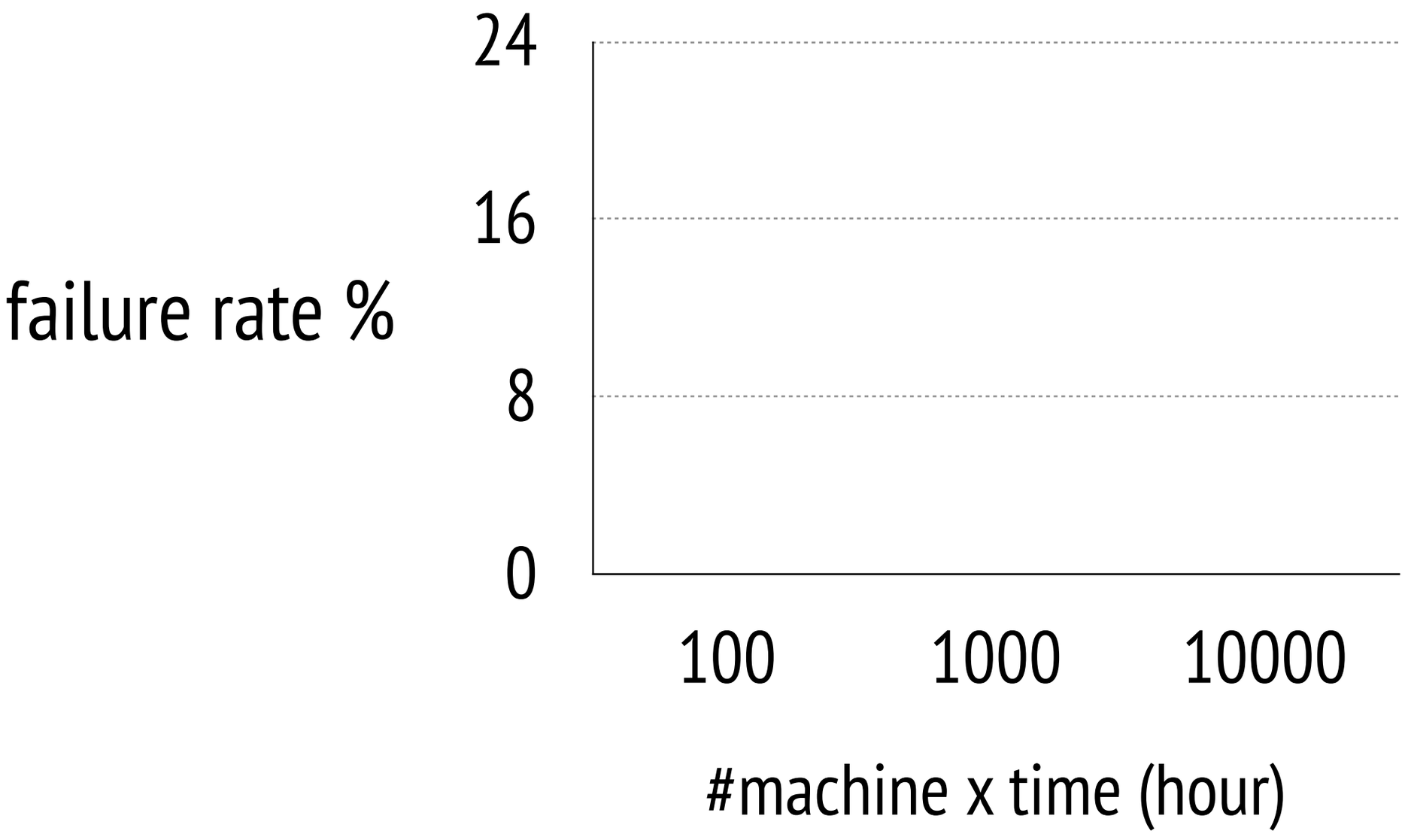
Fault tolerance

Easy to use

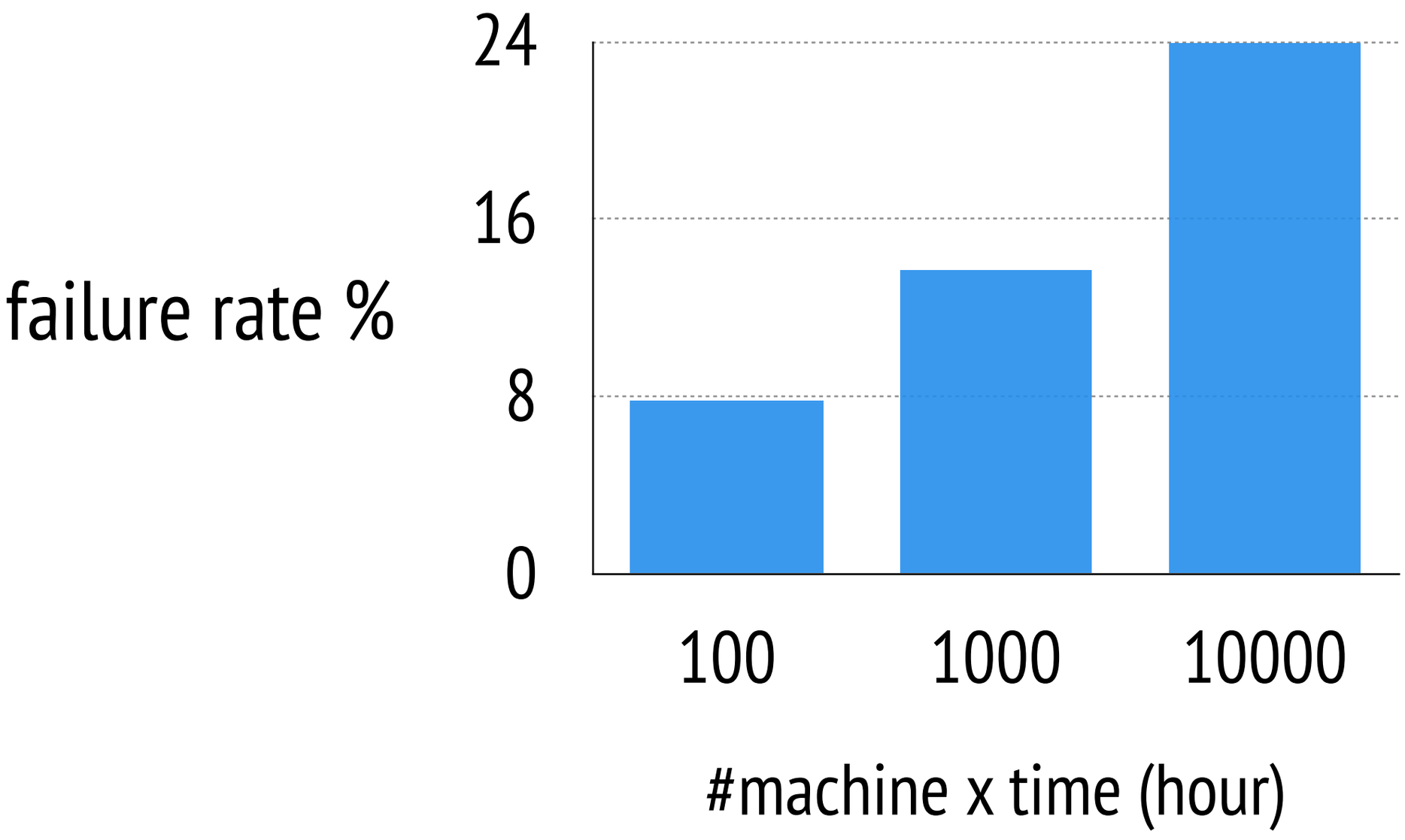
Evaluation

Machine learning job logs
in a three-month period:

Machine learning job logs in a three-month period:



Machine learning job logs in a three-month period:



Fault tolerance

Fault tolerance

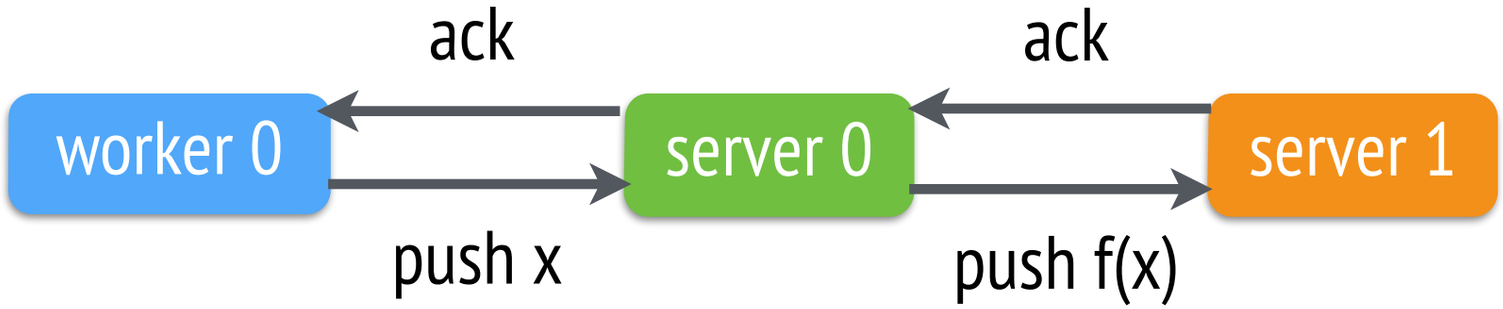
- Model is partitioned by consistent hashing

Fault tolerance

- Model is partitioned by consistent hashing
- Default replication: Chain replication (consistent, safe)

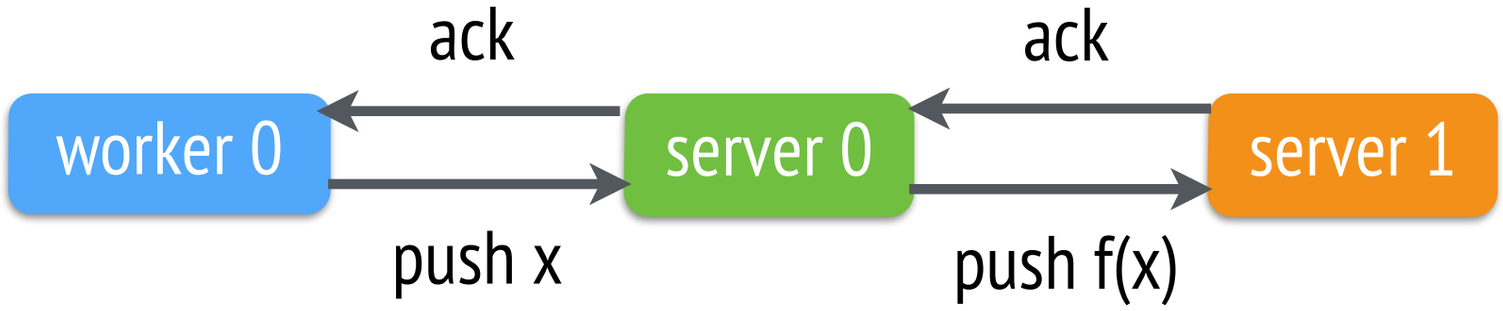
Fault tolerance

- Model is partitioned by consistent hashing
- Default replication: Chain replication (consistent, safe)

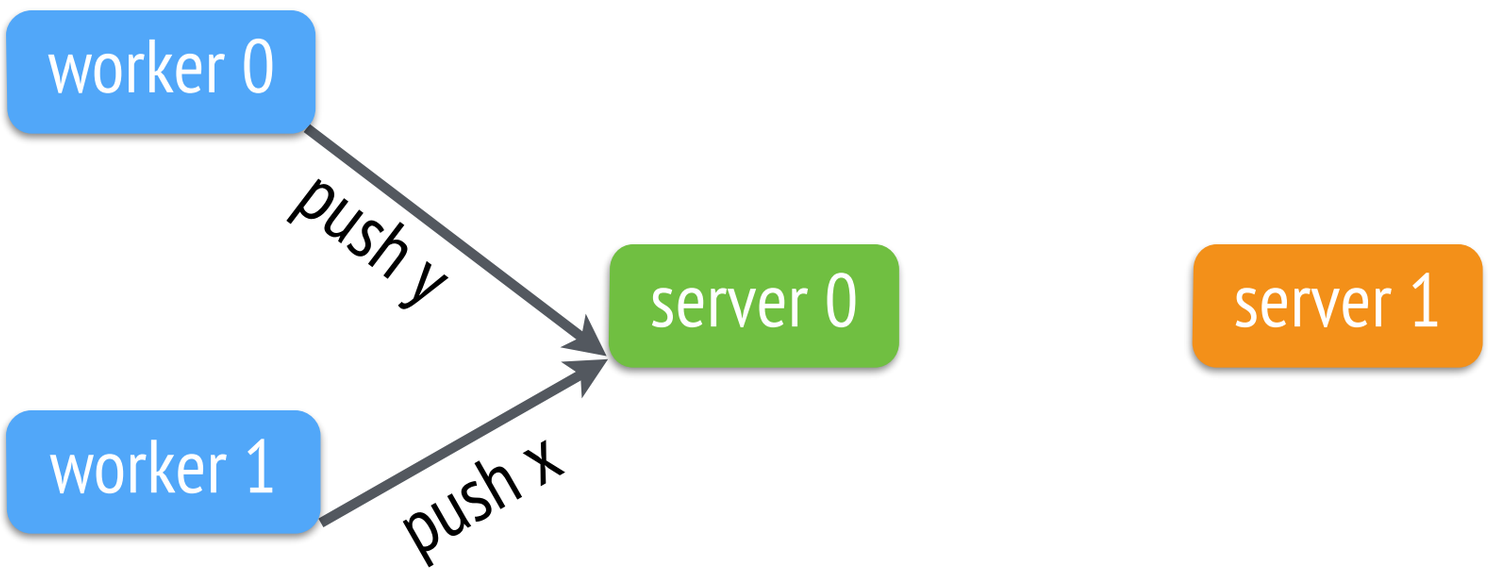


Fault tolerance

- Model is partitioned by consistent hashing
- Default replication: Chain replication (consistent, safe)

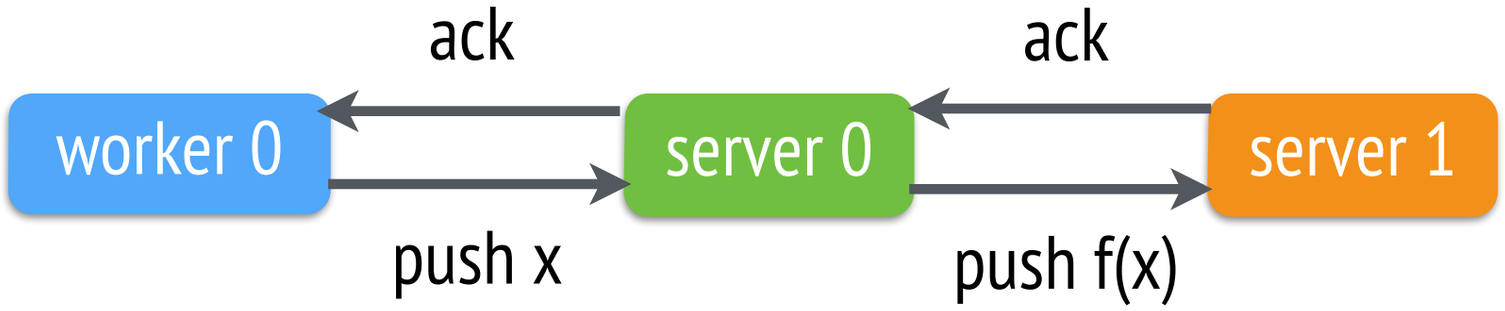


- Option: Aggregation reduces backup traffic (algo specific)

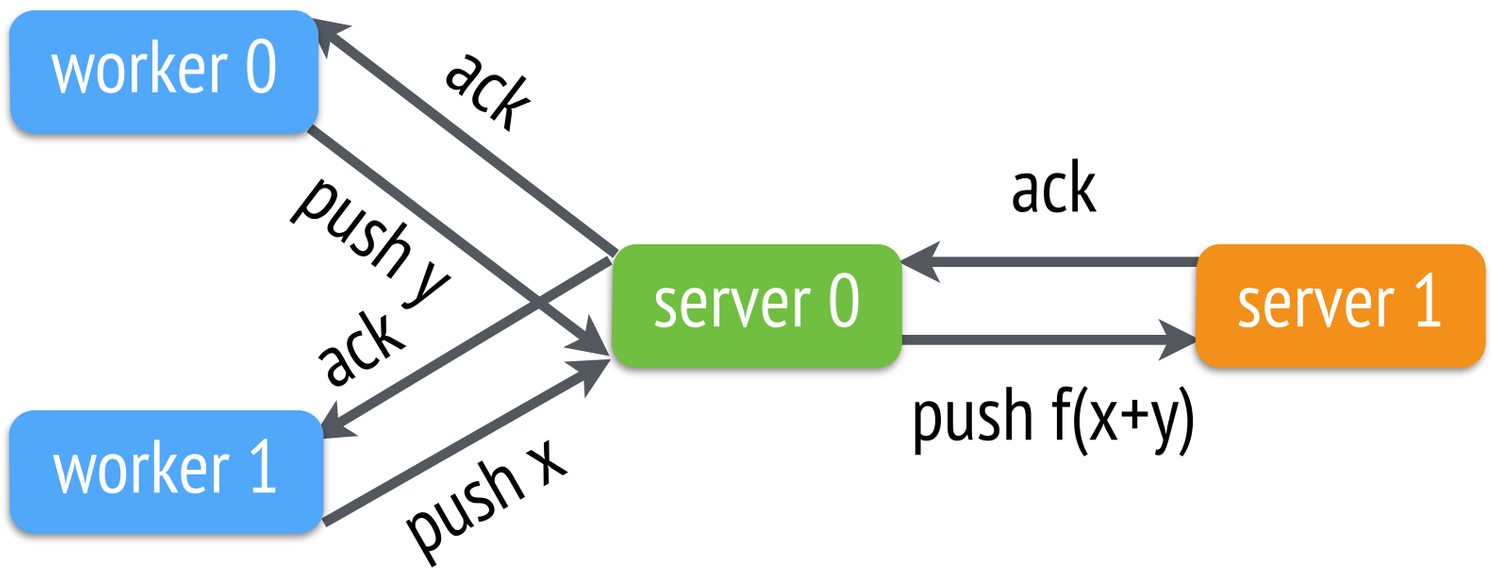


Fault tolerance

- Model is partitioned by consistent hashing
- Default replication: Chain replication (consistent, safe)

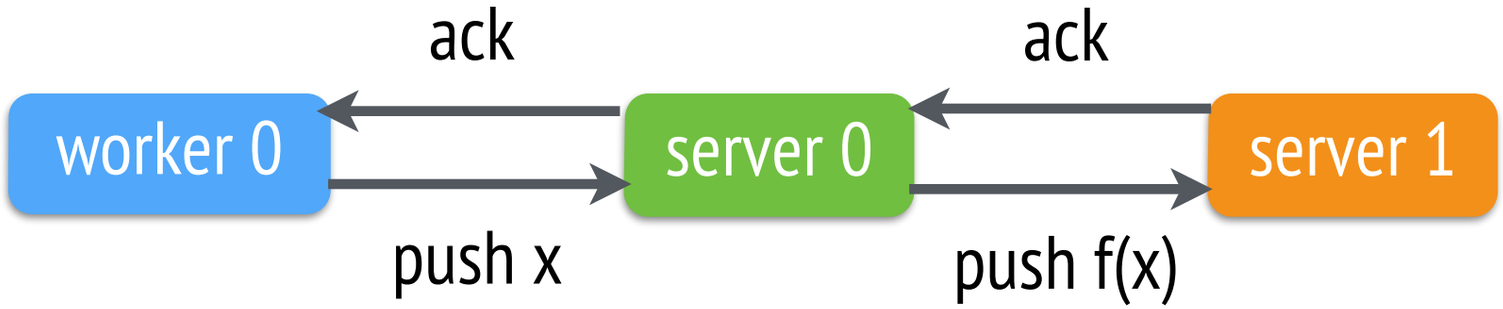


- Option: Aggregation reduces backup traffic (algo specific)



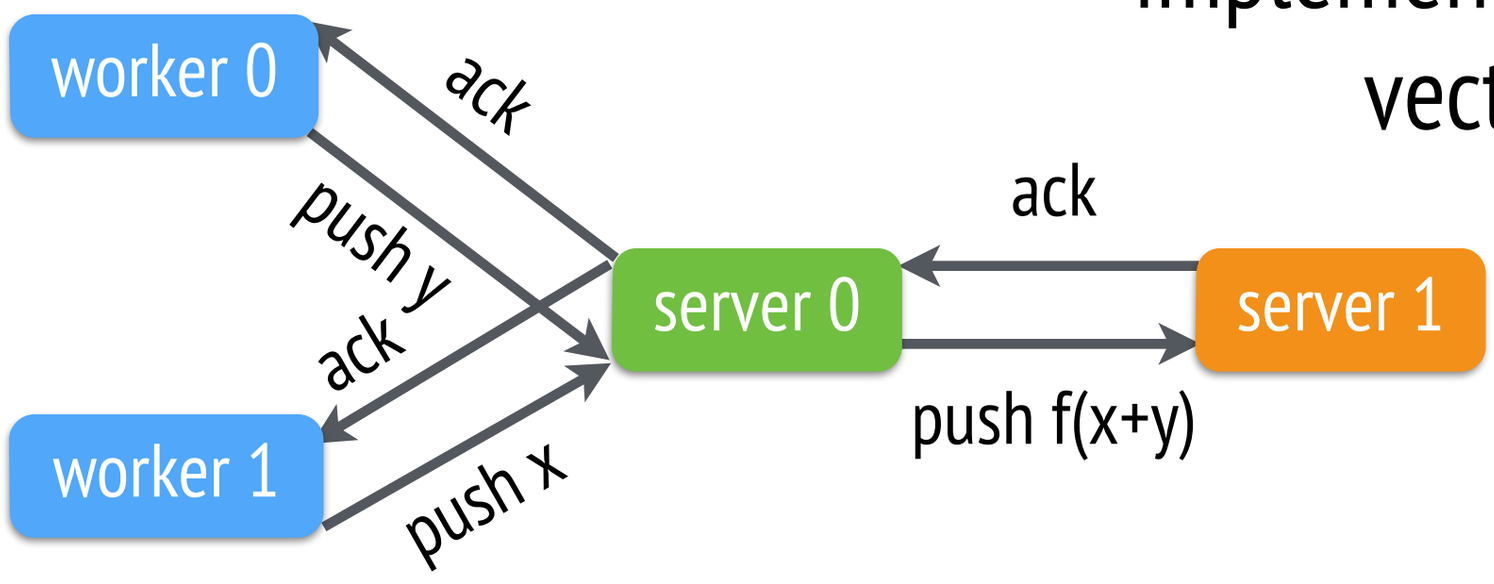
Fault tolerance

- Model is partitioned by consistent hashing
- Default replication: Chain replication (consistent, safe)



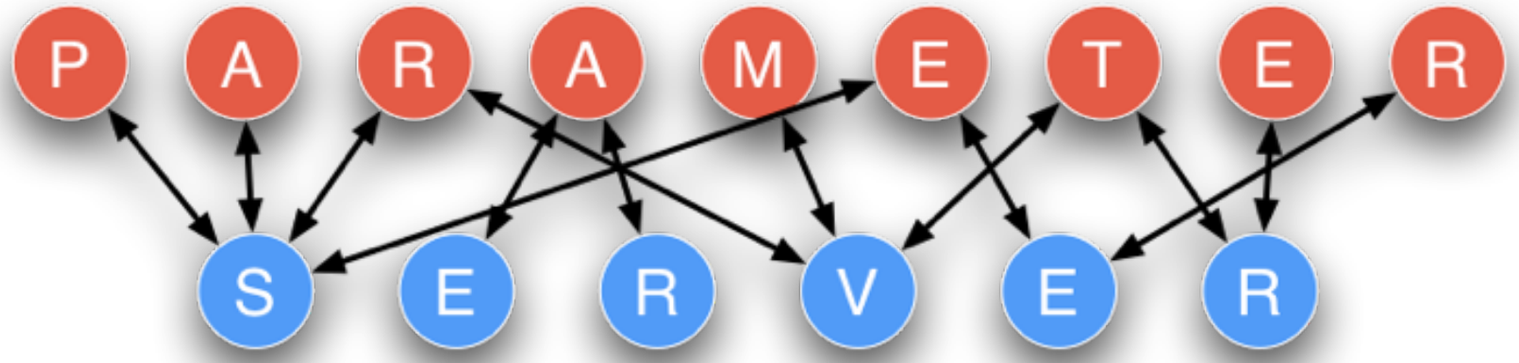
- Option: Aggregation reduces backup traffic (algo specific)

implemented by efficient vector clock



Industry size machine learning problems ✓

Efficient communication ✓



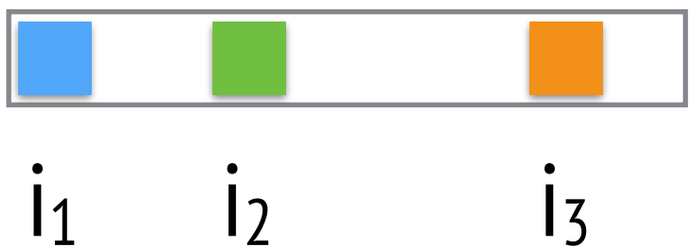
Fault tolerance ✓

Easy to use

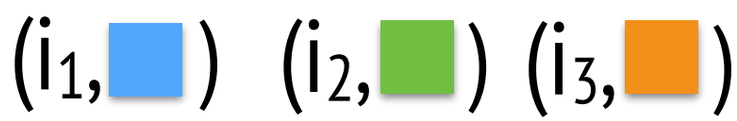
Evaluation

(Key, value) vectors for the shared parameters

math sparse vector



(key, value) store



(Key, value) vectors for the shared parameters

- Good for programmers: Matches mental model
- Good for system: Expose optimizations based upon structure of data

(Key, value) vectors for the shared parameters

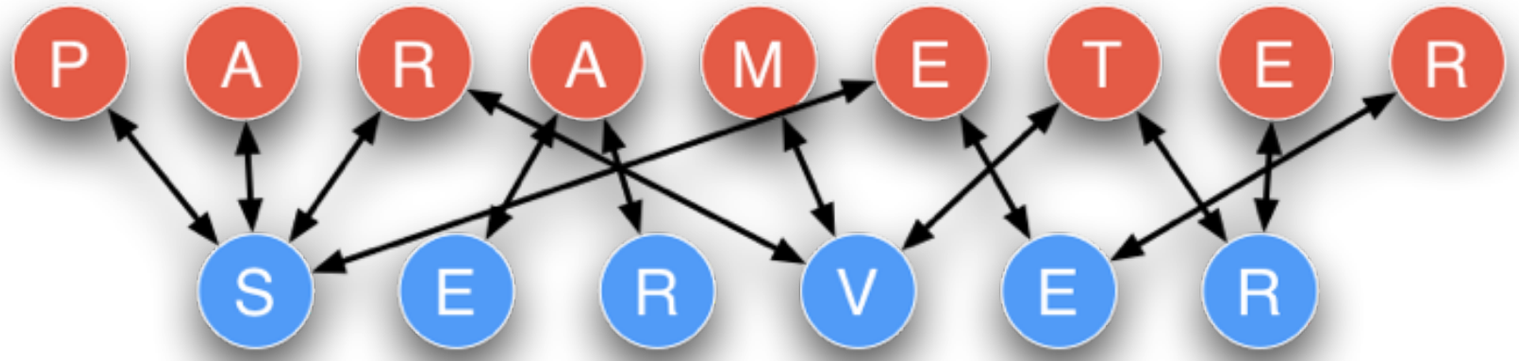
- Good for programmers: Matches mental model
- Good for system: Expose optimizations based upon structure of data

Example: computing gradient

$$\text{gradient} = \text{data}^T \times (- \text{label} \times 1 / (1 + \exp (\text{label} \times \text{data} \times \text{model}))$$

Industry size machine learning problems ✓

Efficient communication ✓



Fault tolerance ✓

Easy to use ✓

Evaluation

Sparse Logistic Regression

- Predict whether ads will be clicked or not
- Baseline: two production systems of a leading Internet company

Sparse Logistic Regression

- Predict whether ads will be clicked or not
- Baseline: two production systems of a leading Internet company

	Method	Consistency	LOC
System-A	L-BFGS	Sequential	10K
System-B	Block PG	Sequential	30K
Parameter Server	Block PG	Bounded Delay + KKT	300

Sparse Logistic Regression

- Predict whether ads will be clicked or not
- Baseline: two production systems of a leading Internet company

Only for sparse logistic regression

	Method	Consistency	LOC
System-A	L-BFGS	Sequential	10K
System-B	Block PG	Sequential	30K
Parameter Server	Block PG	Bounded Delay + KKT	300

Sparse Logistic Regression

- Predict whether ads will be clicked or not
- Baseline: two production systems of a leading Internet company

Only for sparse logistic regression

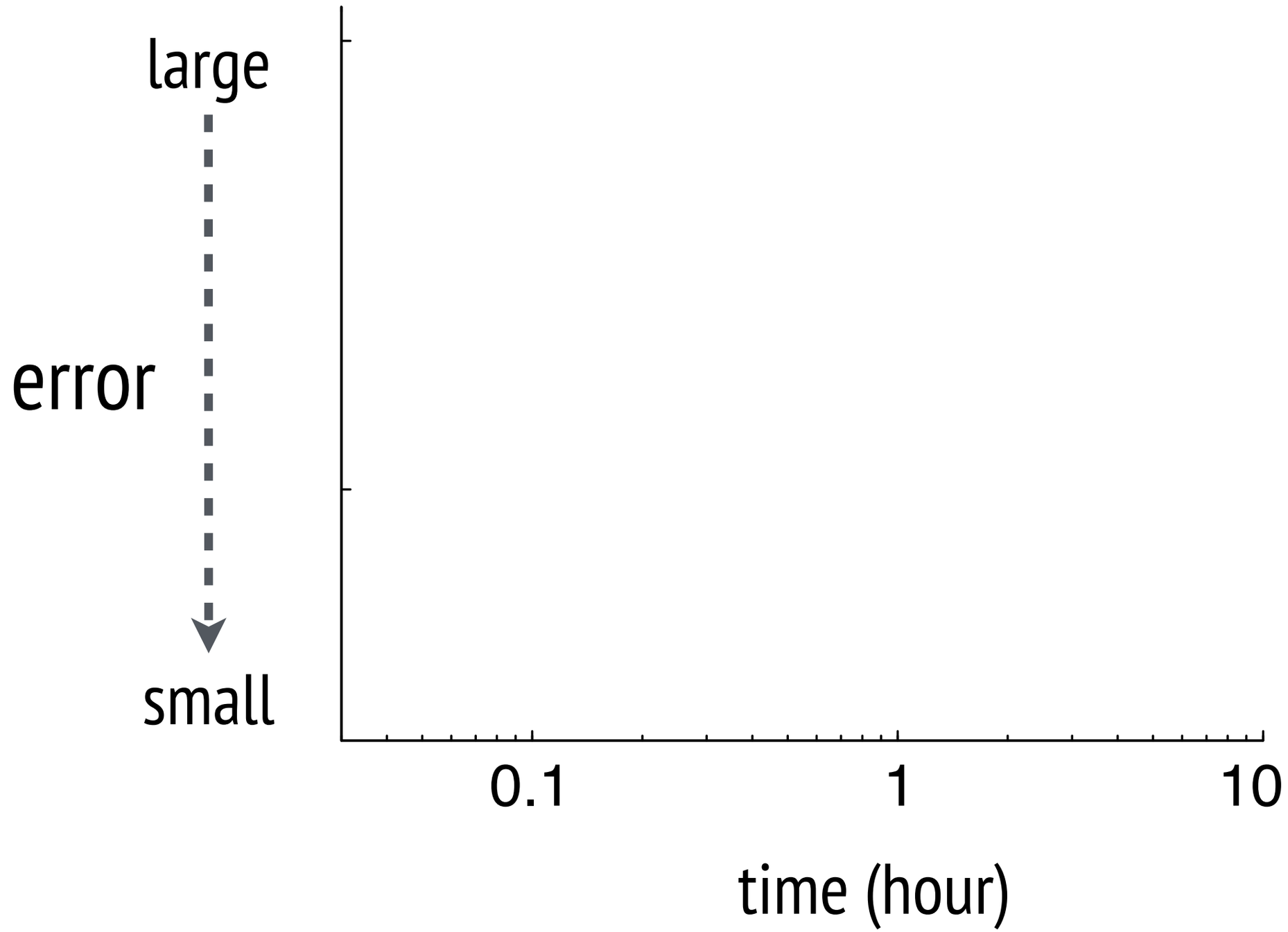
	Method	Consistency	LOC
System-A	L-BFGS	Sequential	10K
System-B	Block PG	Sequential	30K
Parameter Server	Block PG	Bounded Delay + KKT	300

Implement the method used by System-B with another 10K system codes

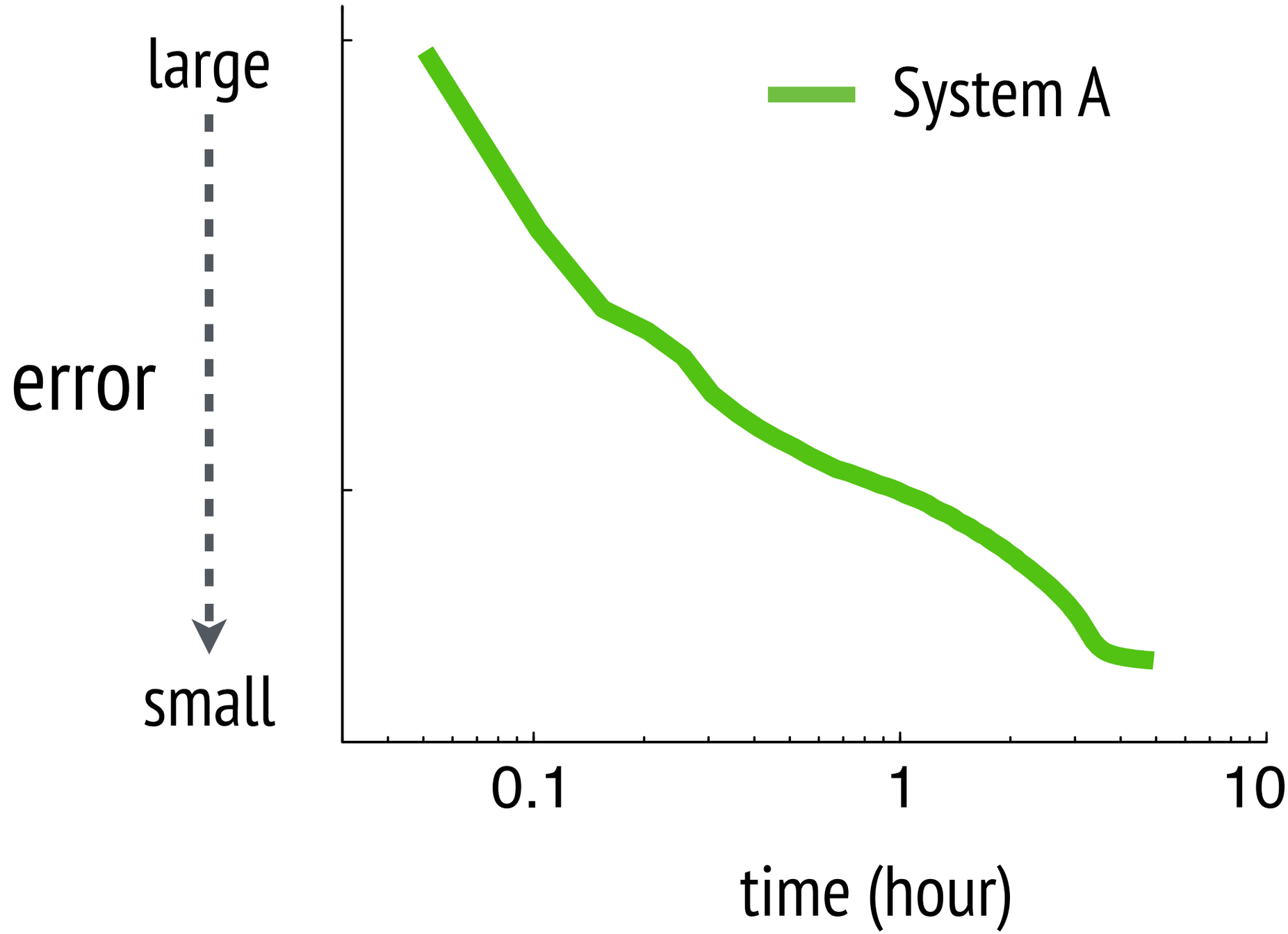
Sparse Logistic Regression

- Predict whether ads will be clicked or not
- Baseline: two production systems of a learning Internet company
- Collected 636TB real ads data
  - 170 billion examples, 65 billion features
- Train 1TB model
- 1,000 machines with 16,000 CPU cores

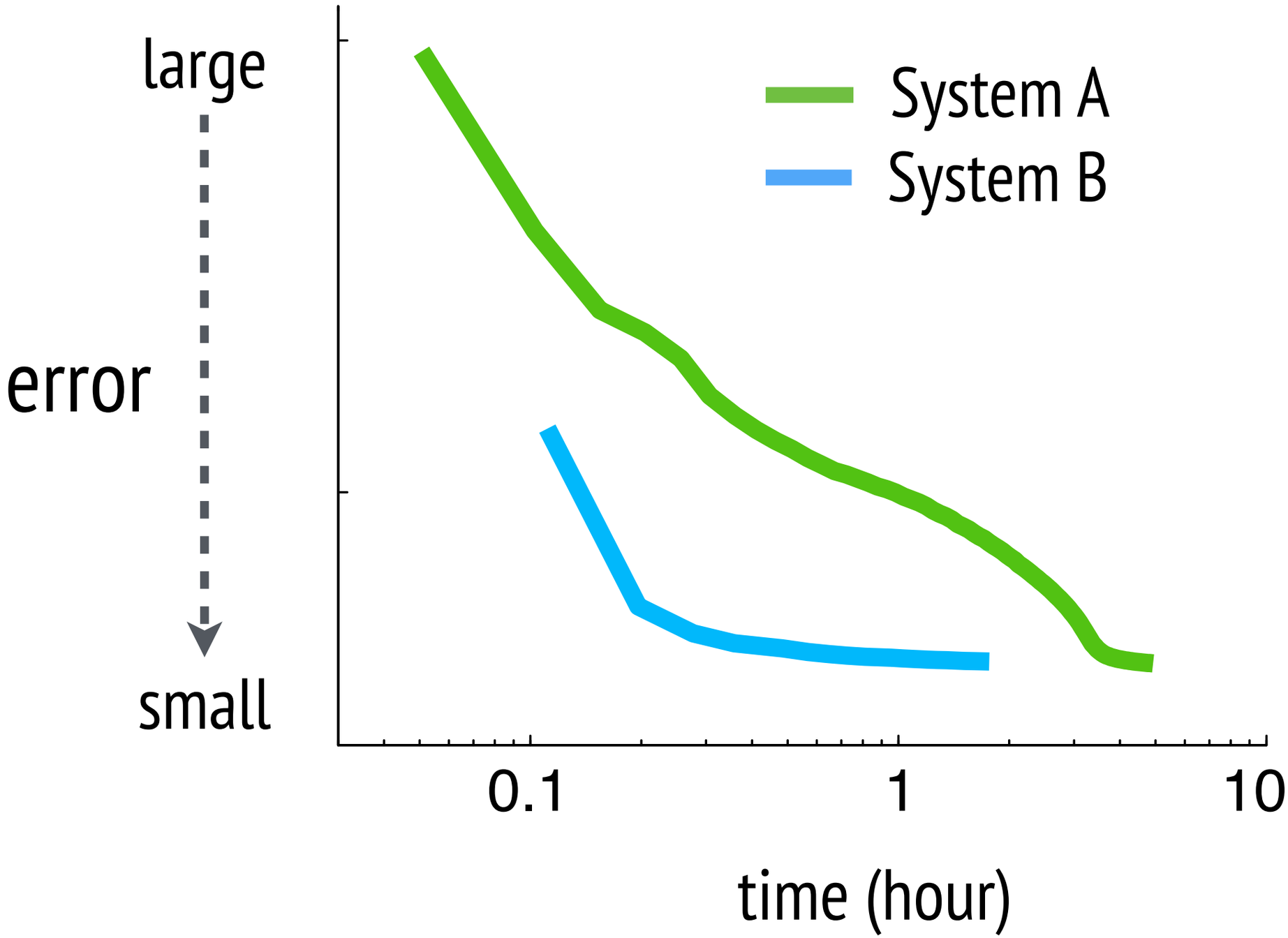
Progress



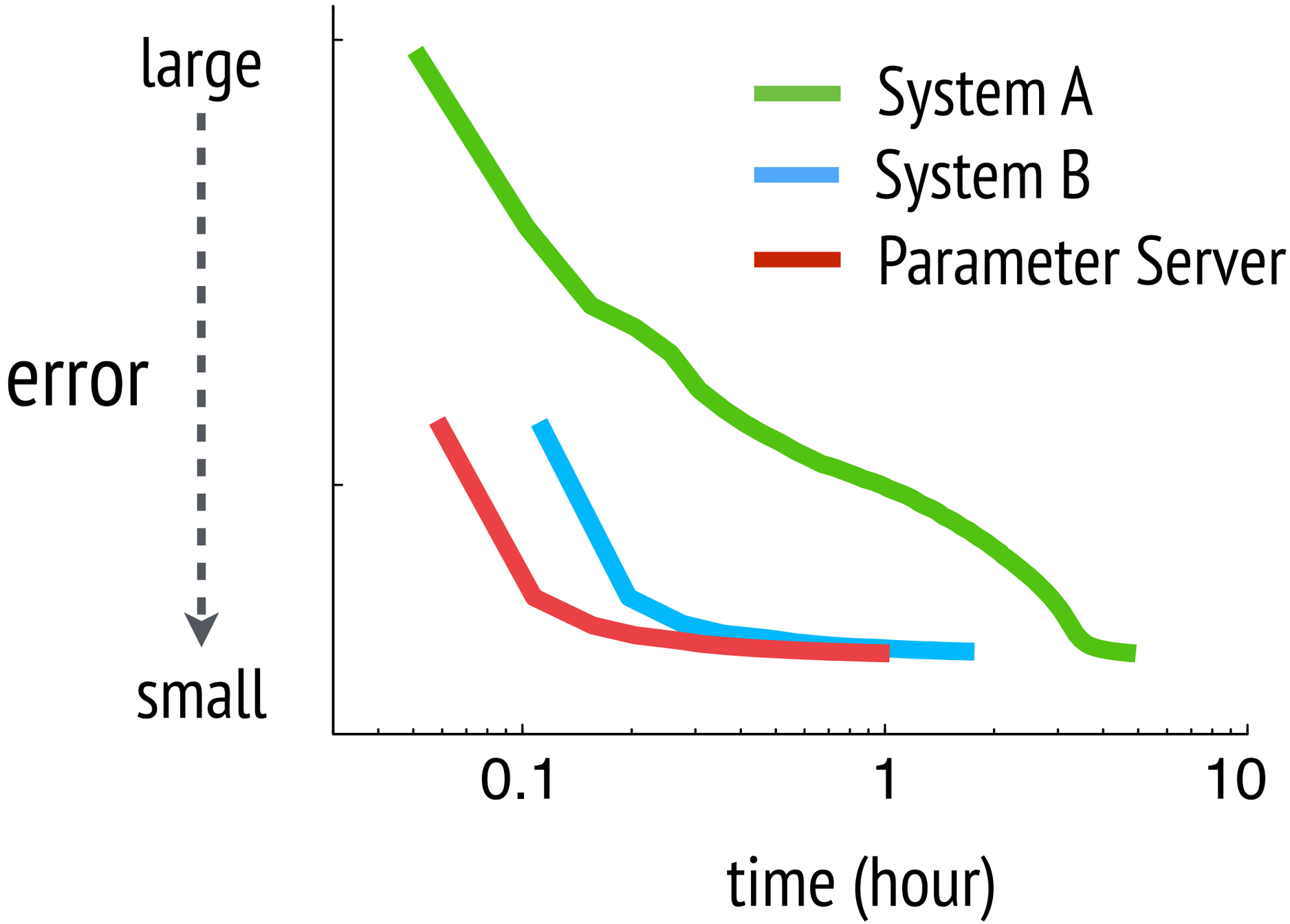
Progress



Progress



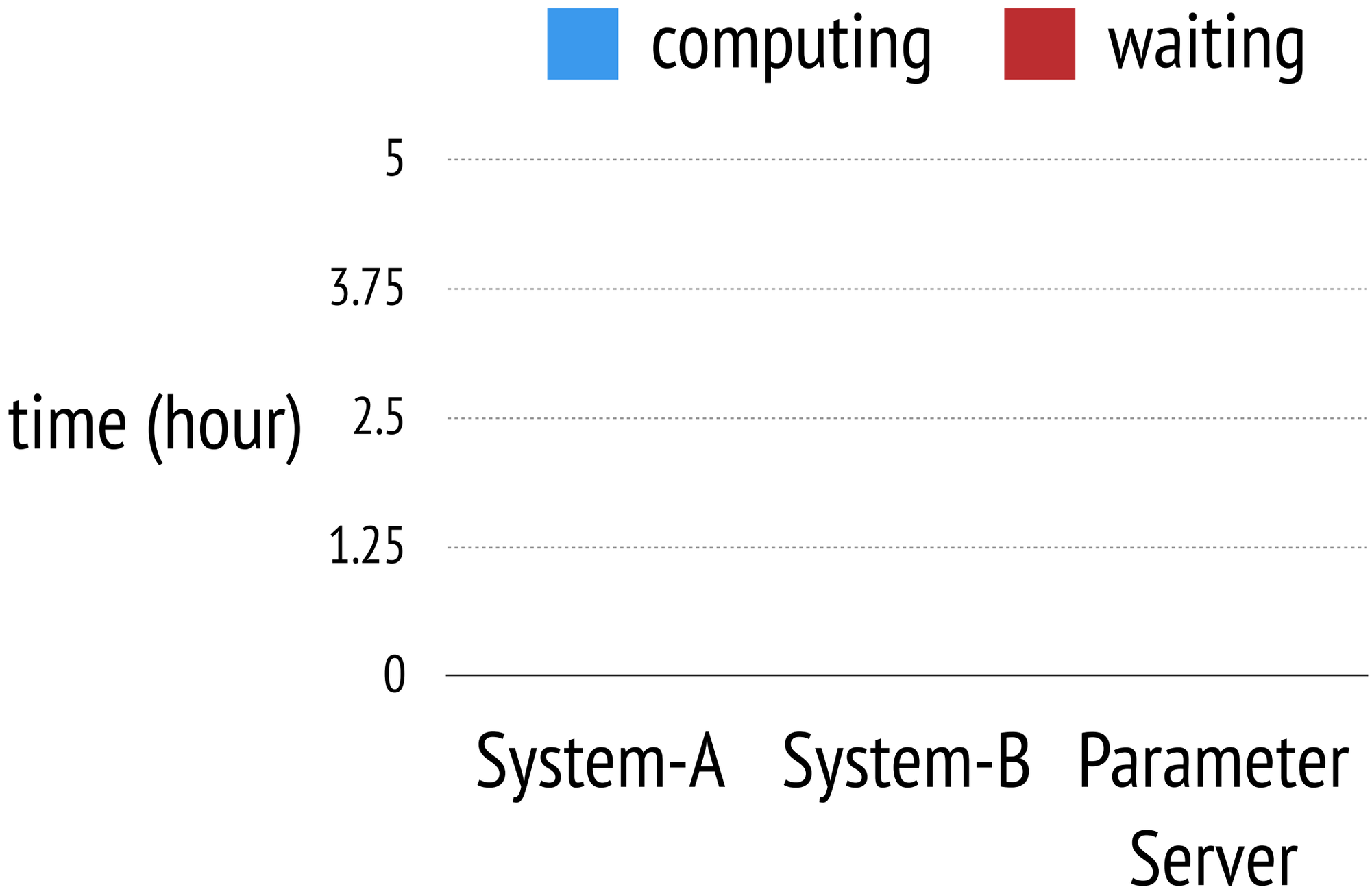
Progress


large
error
small
System A
System B
Parameter Server
0.1
1
10
time (hour)

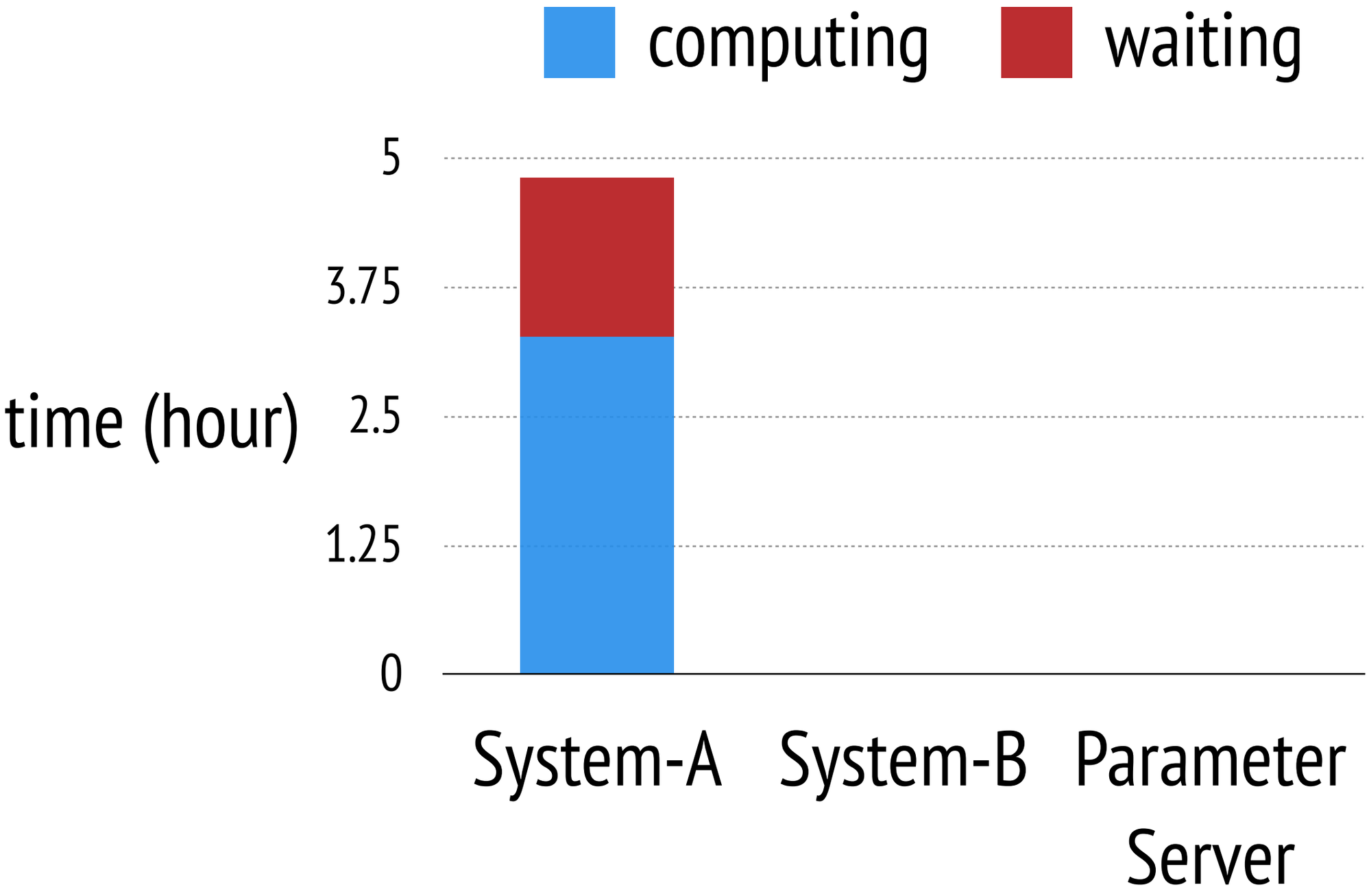
Time to the Same Convergence Criteria

time (hour)

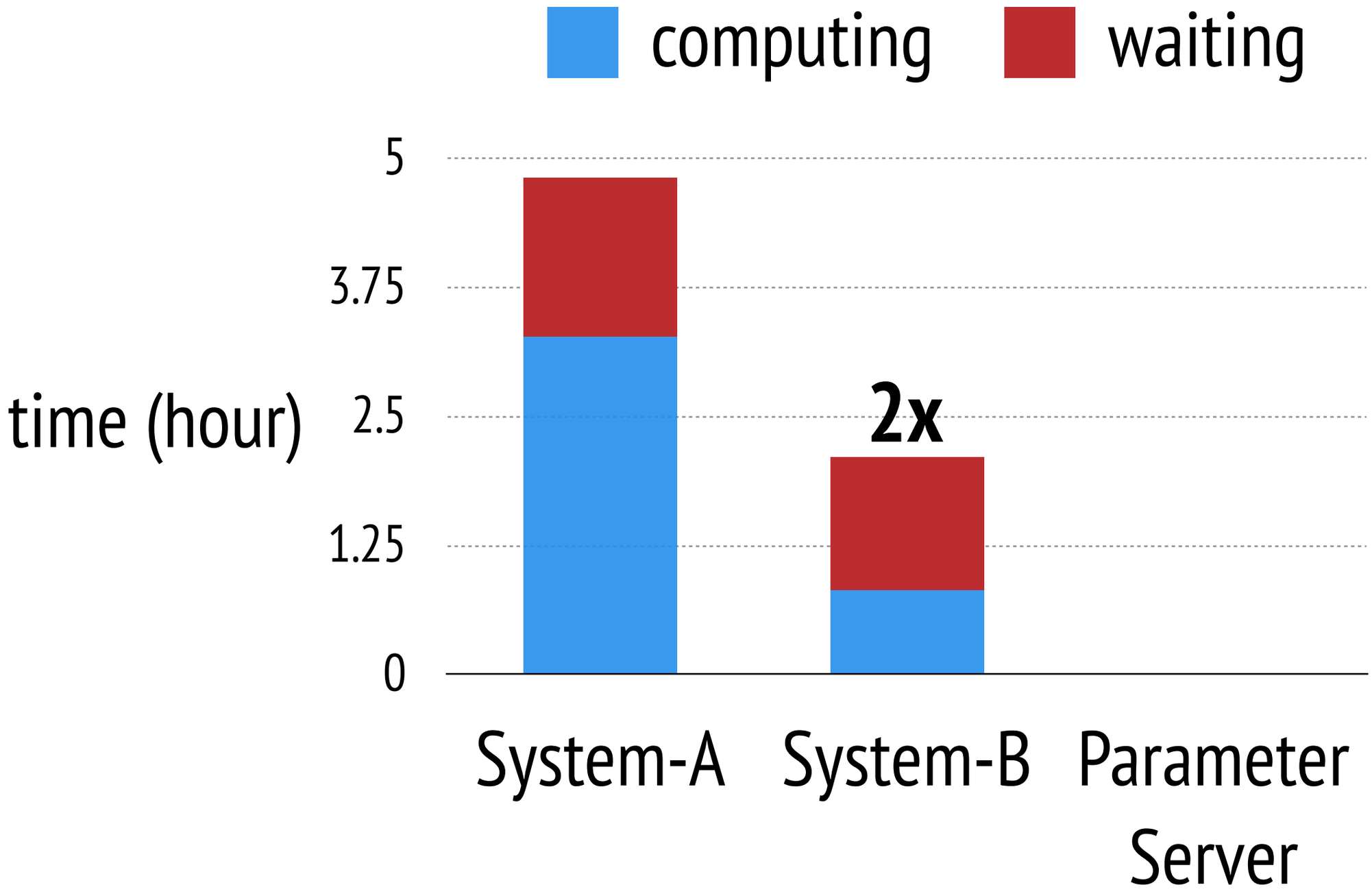
Time to the Same Convergence Criteria



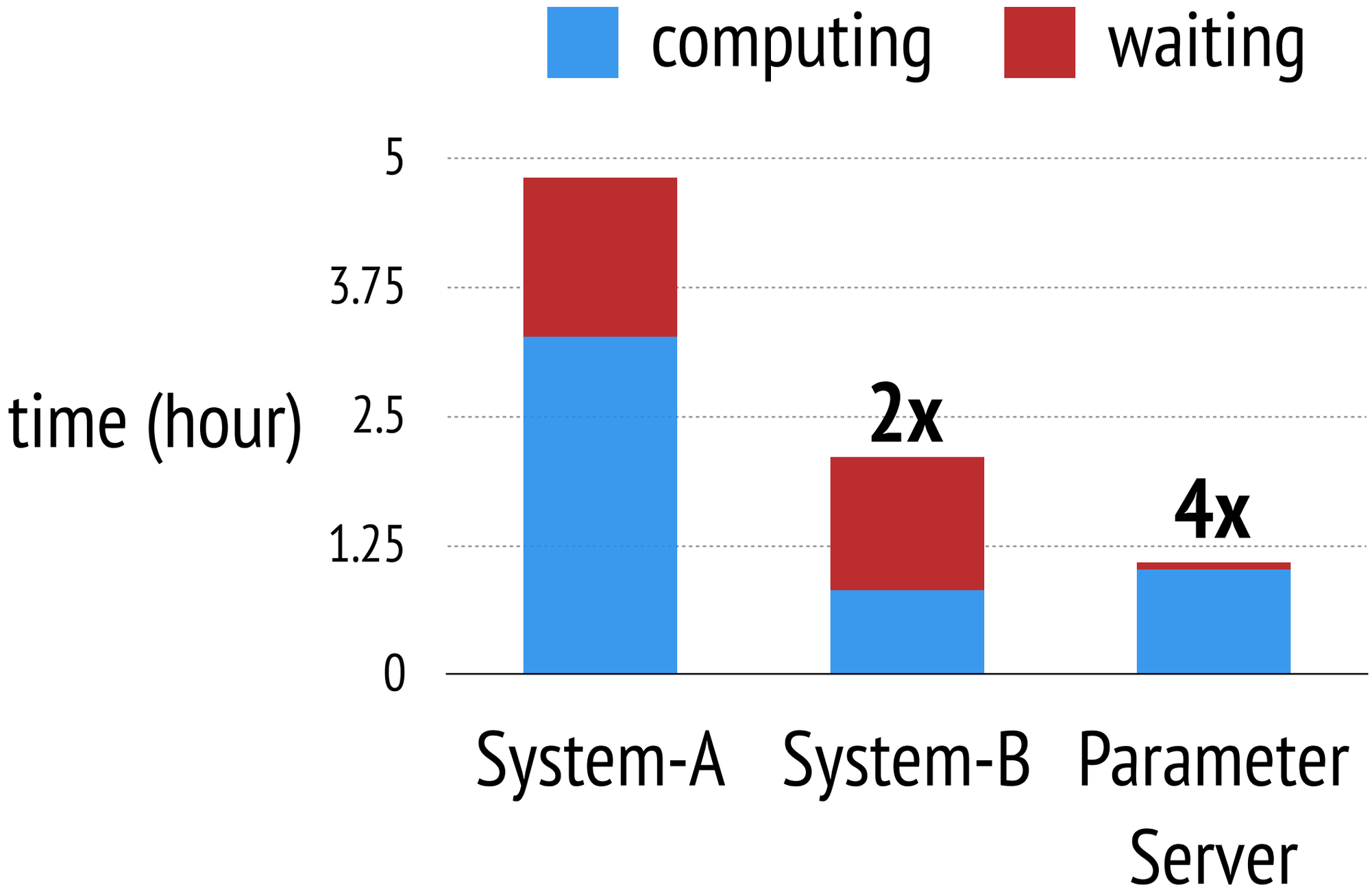
Time to the Same Convergence Criteria



Time to the Same Convergence Criteria



Time to the Same Convergence Criteria

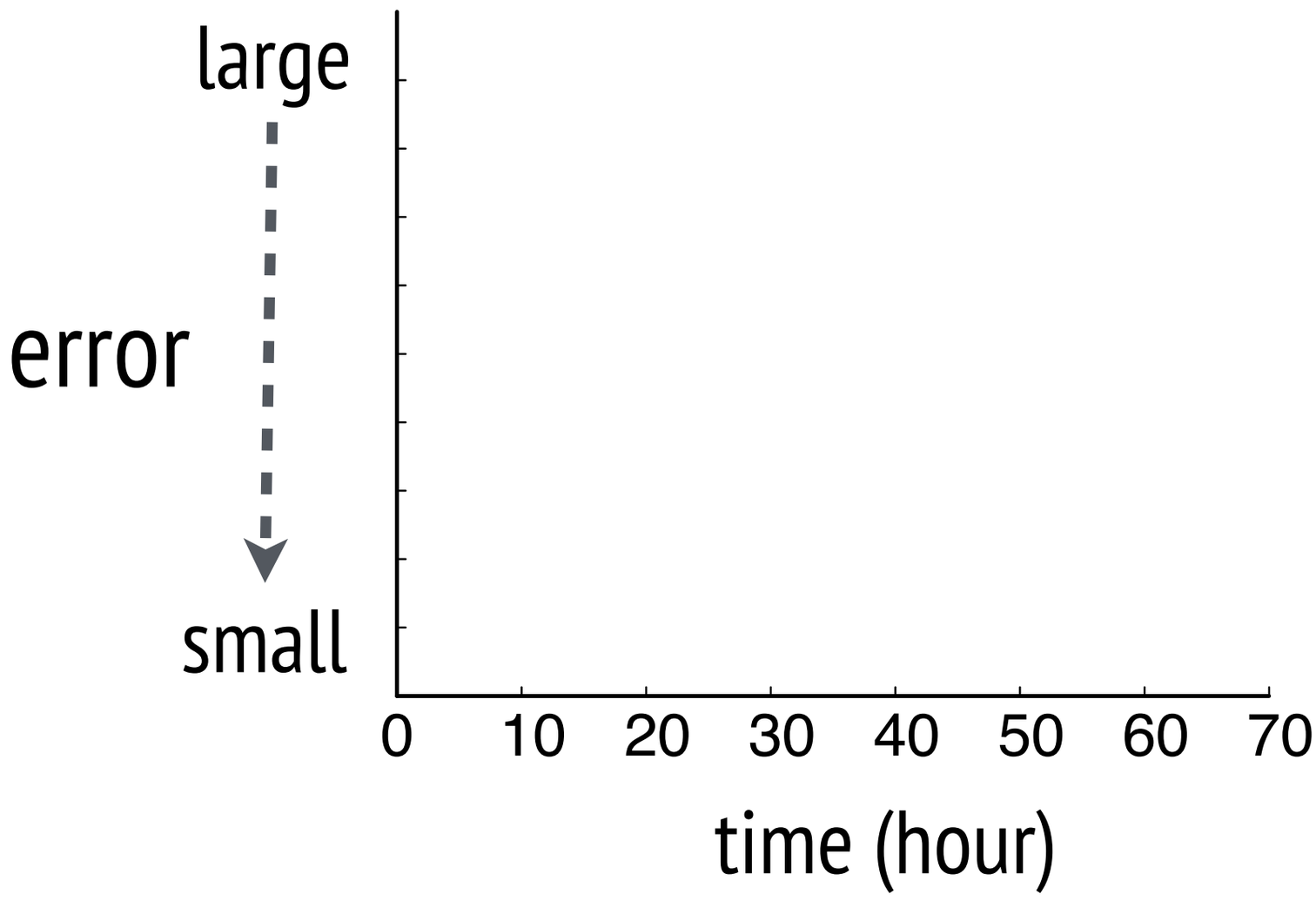


Topic Modeling ("LDA")

- Gradient descent with eventual consistency
- Click logs for 5B users, then group users into 1,000 categories based on clicked URLs

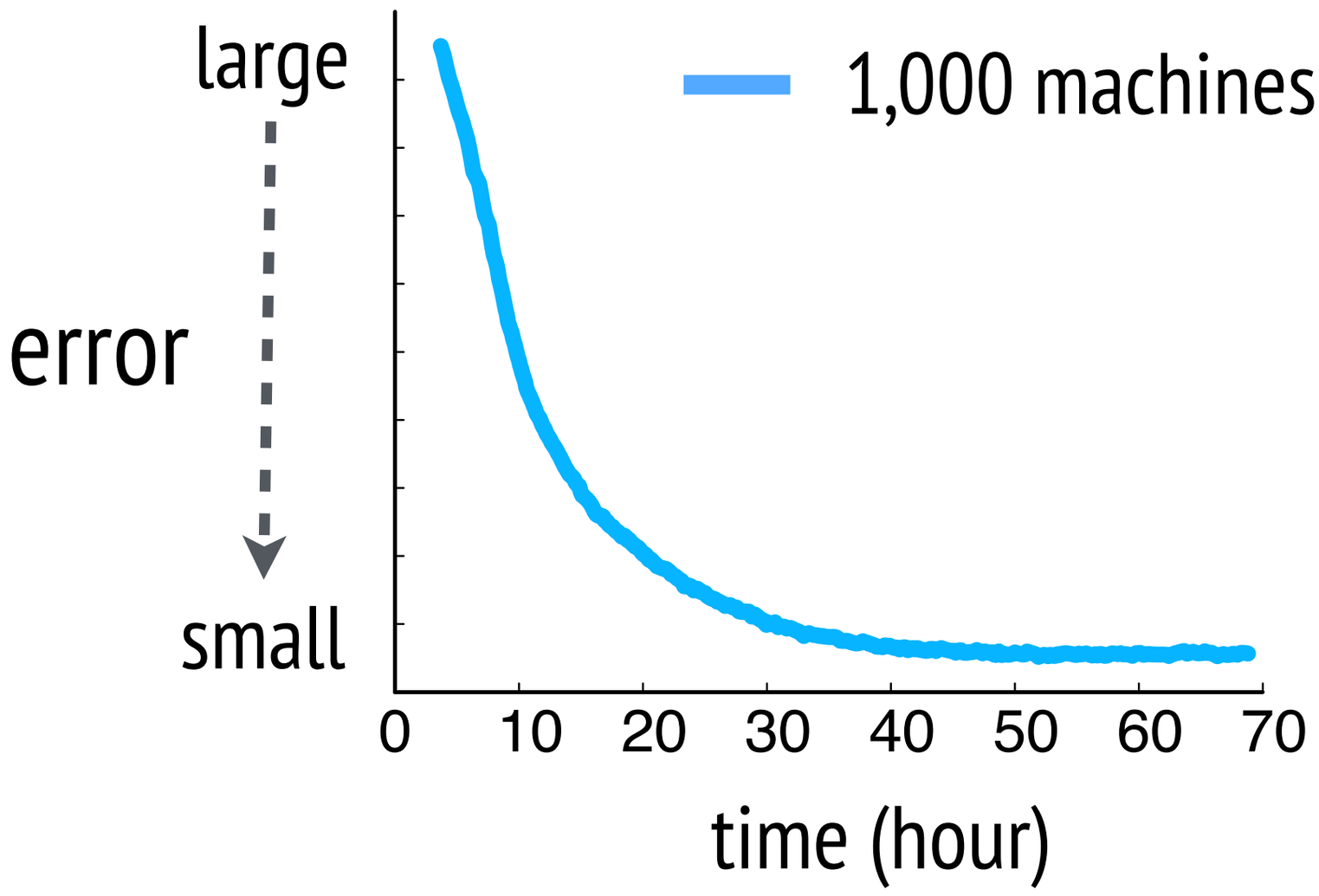
Topic Modeling ("LDA")

- Gradient descent with eventual consistency
- Click logs for 5B users, then group users into 1,000 categories based on clicked URLs



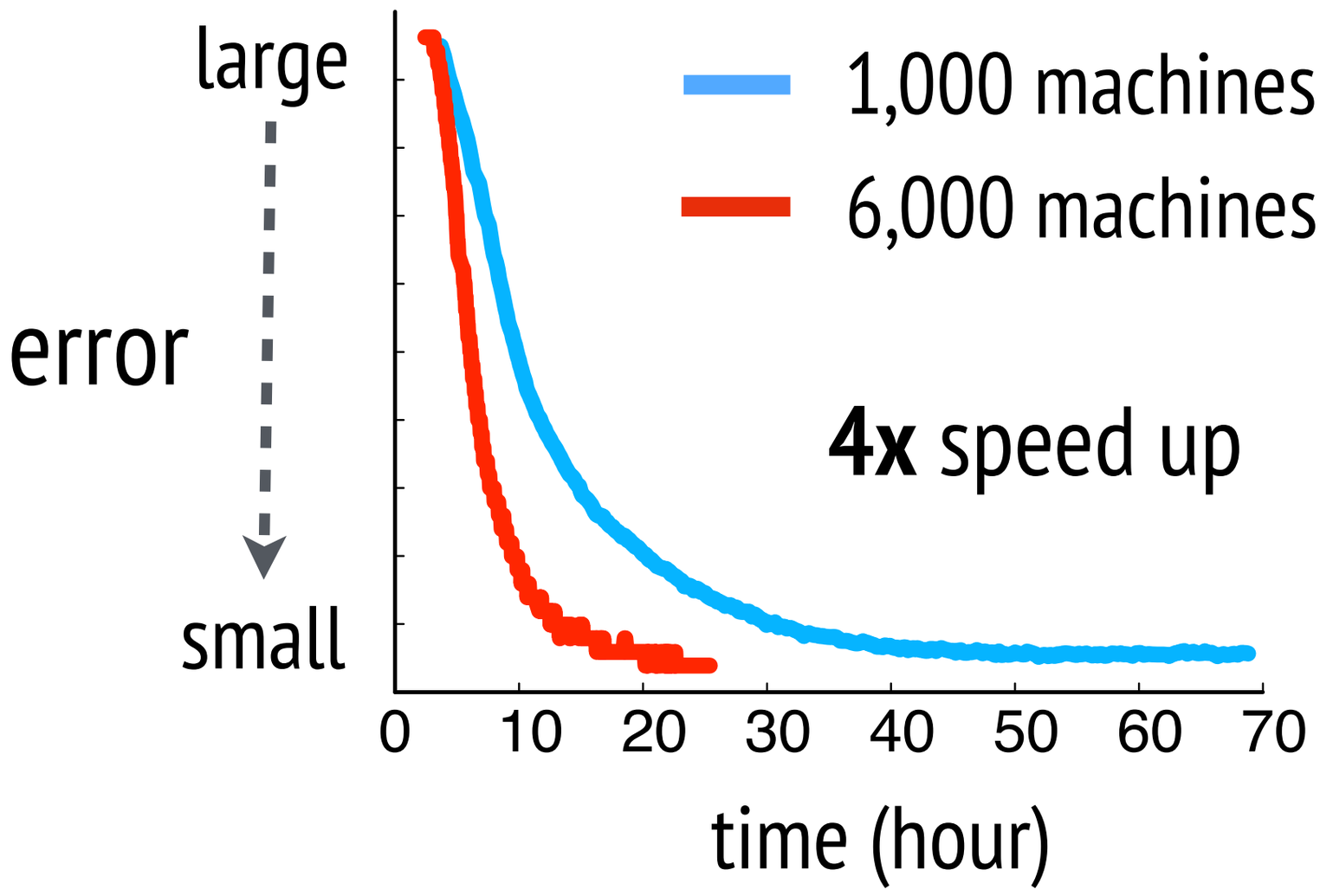
Topic Modeling ("LDA")

- Gradient descent with eventual consistency
- Click logs for 5B users, then group users into 1,000 categories based on clicked URLs



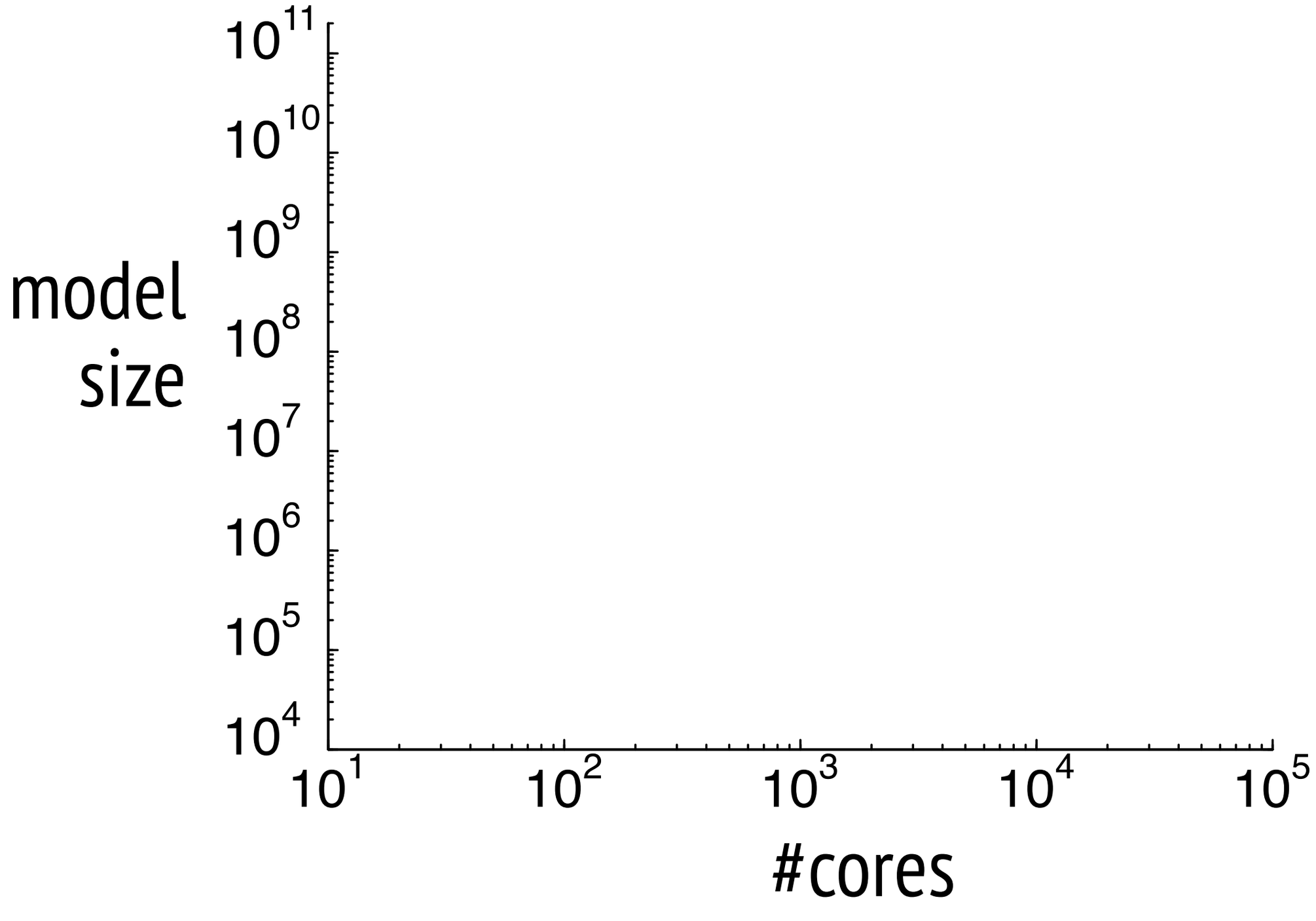
Topic Modeling ("LDA")

- Gradient descent with eventual consistency
- Click logs for 5B users, then group users into 1,000 categories based on clicked URLs



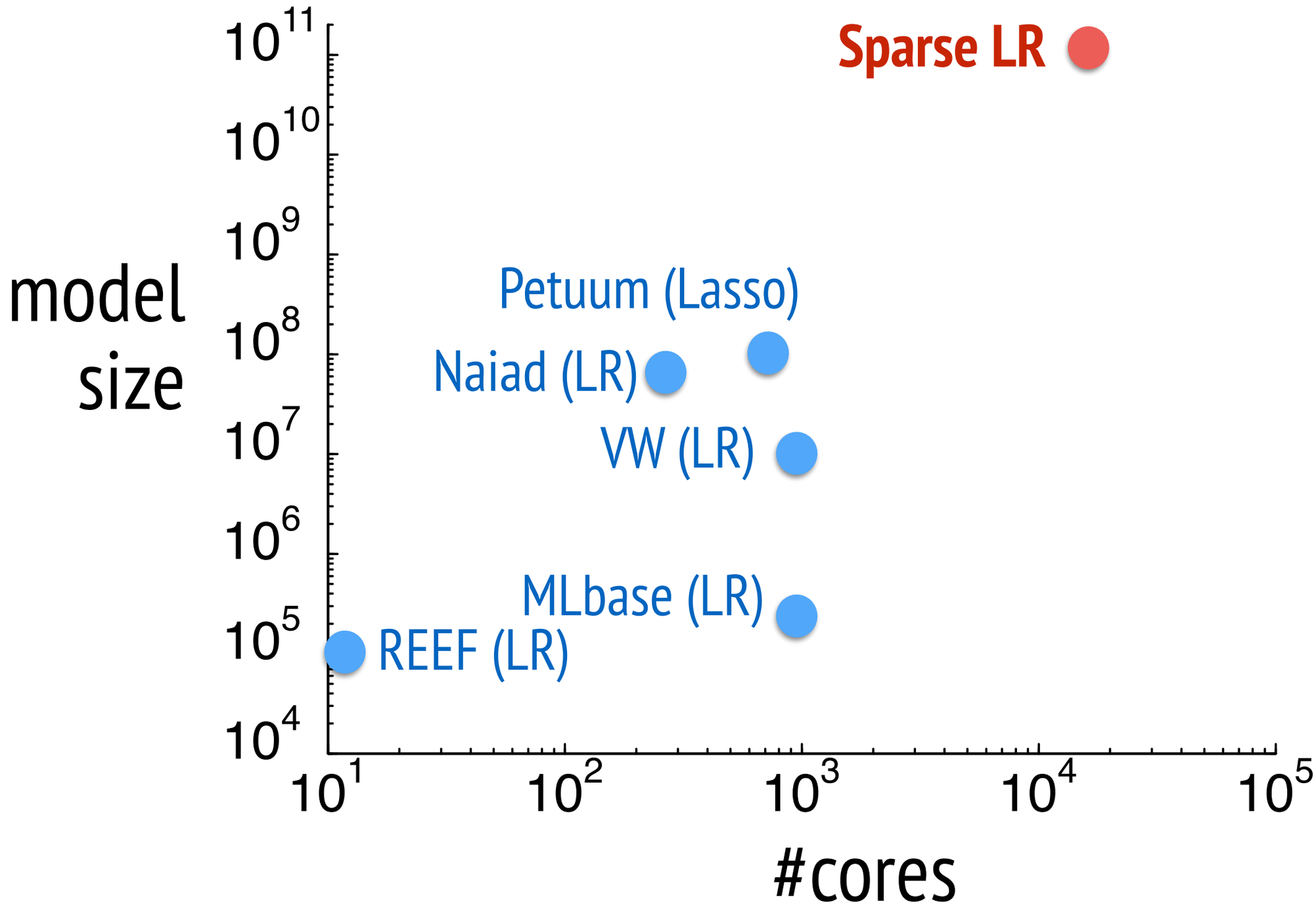
Largest experiments of related systems

Data were collected on April'14



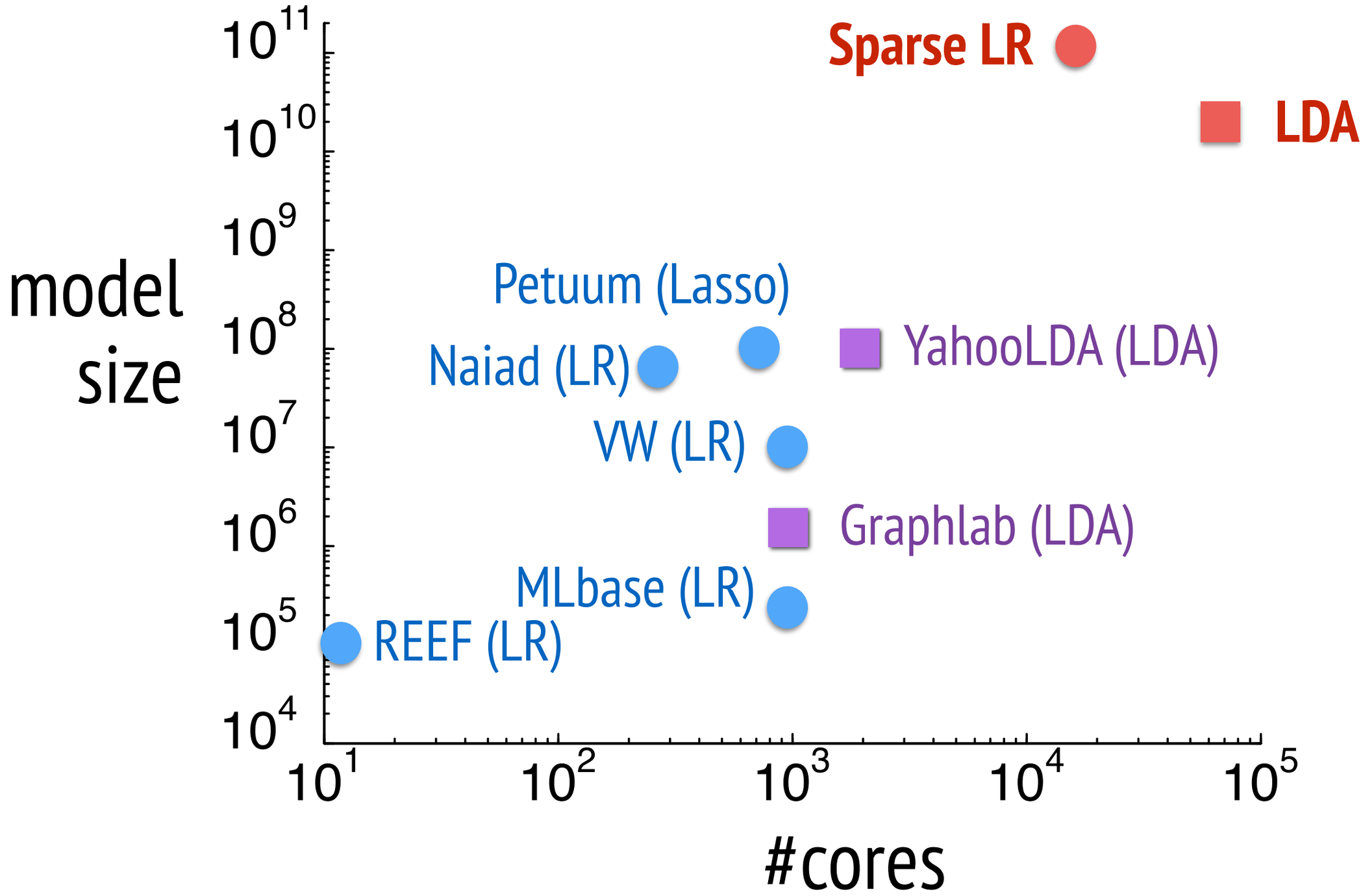
Largest experiments of related systems

Data were collected on April'14



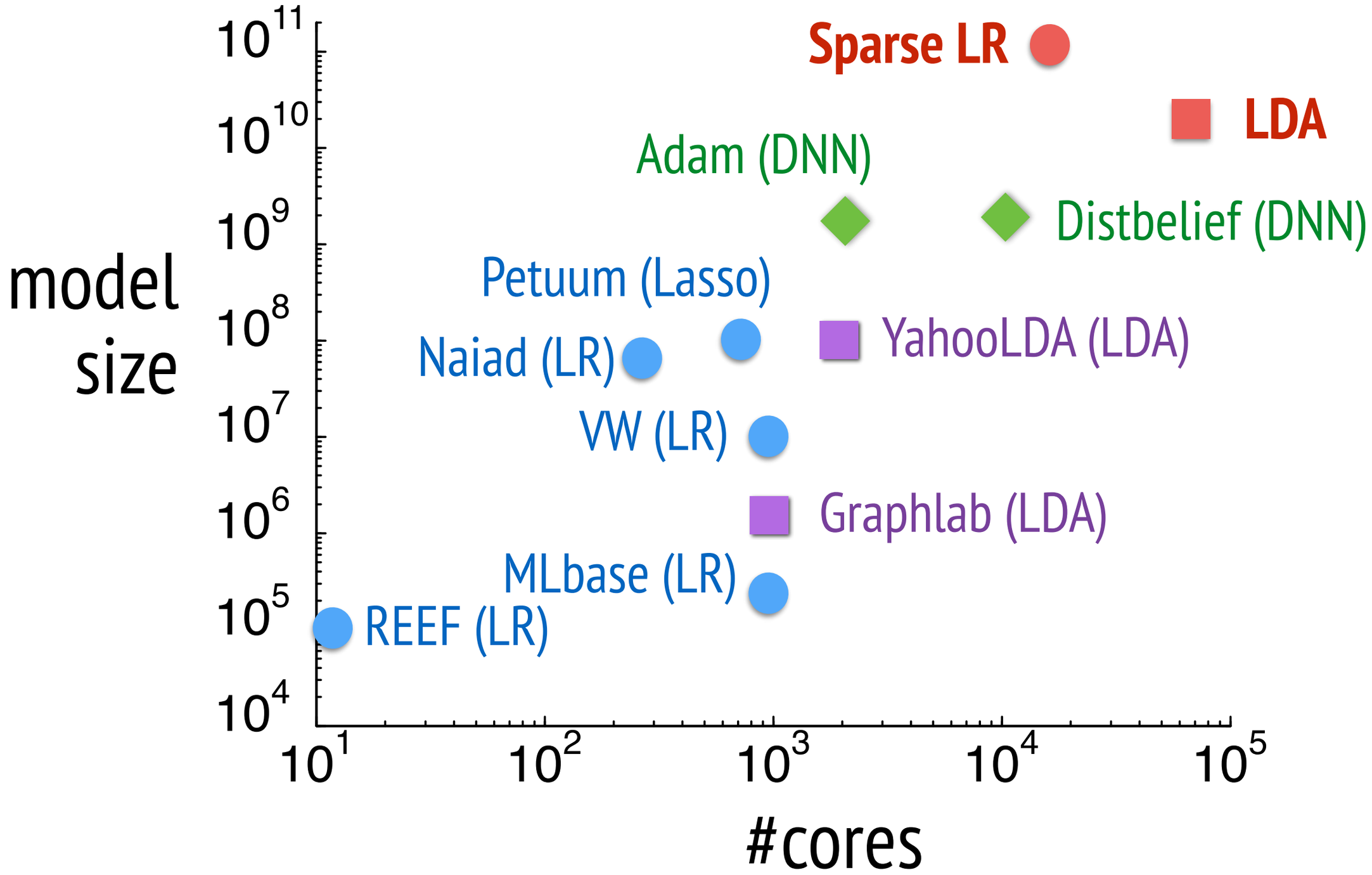
Largest experiments of related systems

Data were collected on April'14



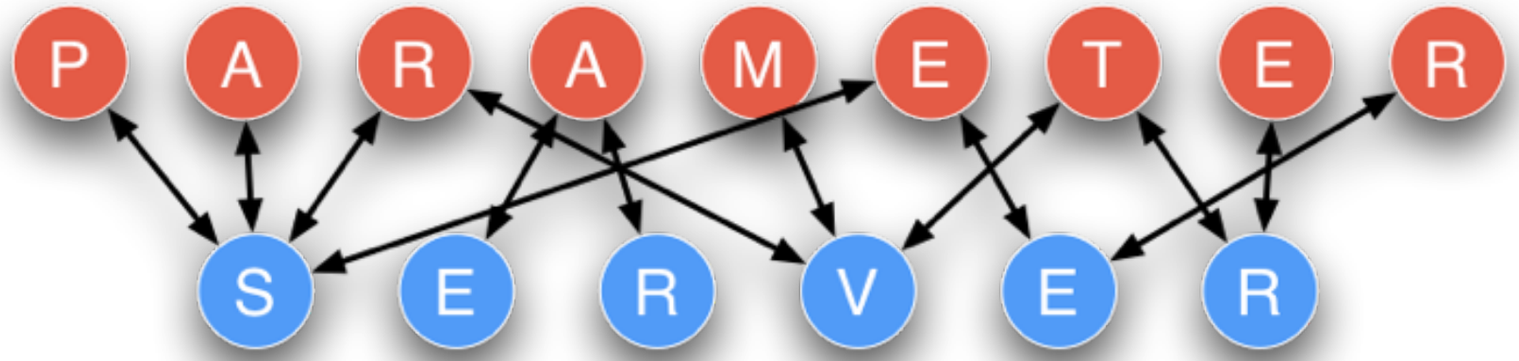
Largest experiments of related systems

Data were collected on April'14



Industry size machine learning problems ✓

Efficient communication ✓



Fault tolerance ✓

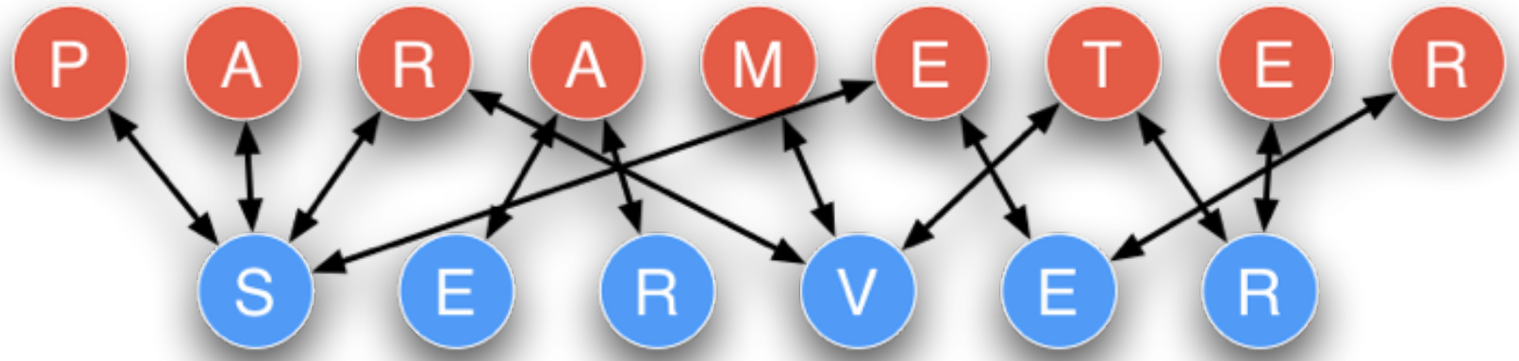
Easy to use ✓

Evaluation ✓

Inc … le … ne … s

Data and model partition

Efficient communication



Fault tolerance

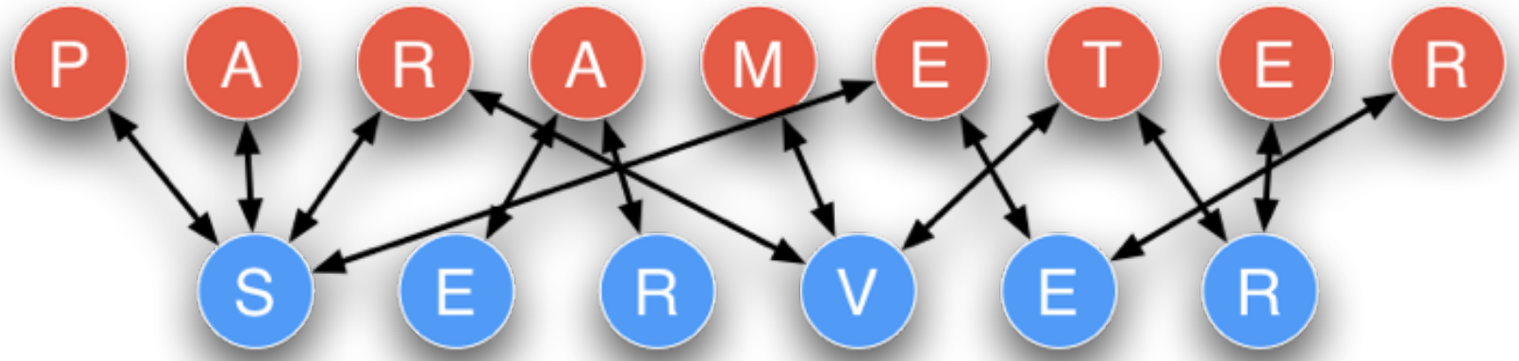
Easy to use

Evaluation

Inc le ne s

Data and model partition

Relax data consistency:
1. Task dependency graph
2. User defined filters



Fault tolerance

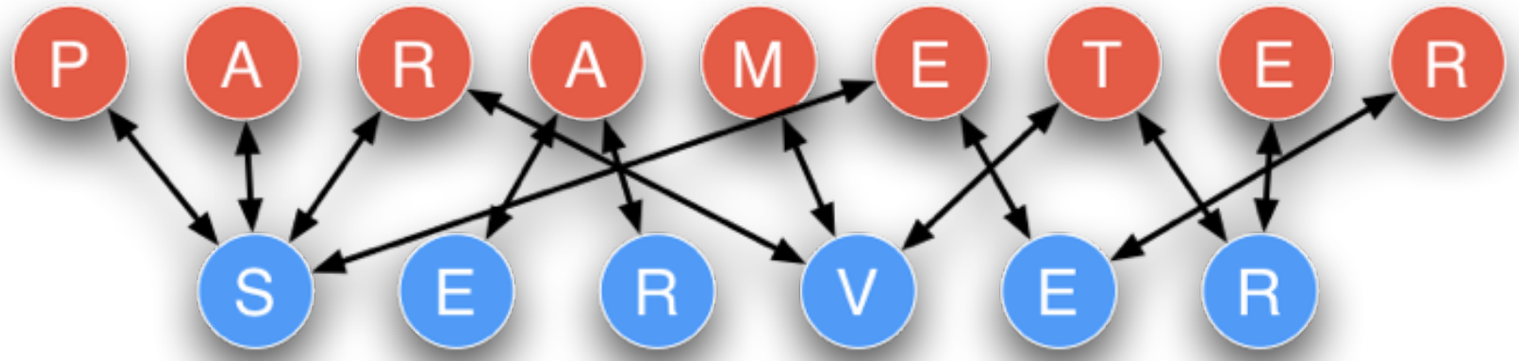
Easy to use

Evaluation

Inc … le … ne … s

Data and model partition

Relax data consistency:
1. Task dependency graph
2. User defined filters



1. Consistency hashing
2. (Aggregated) chain replication

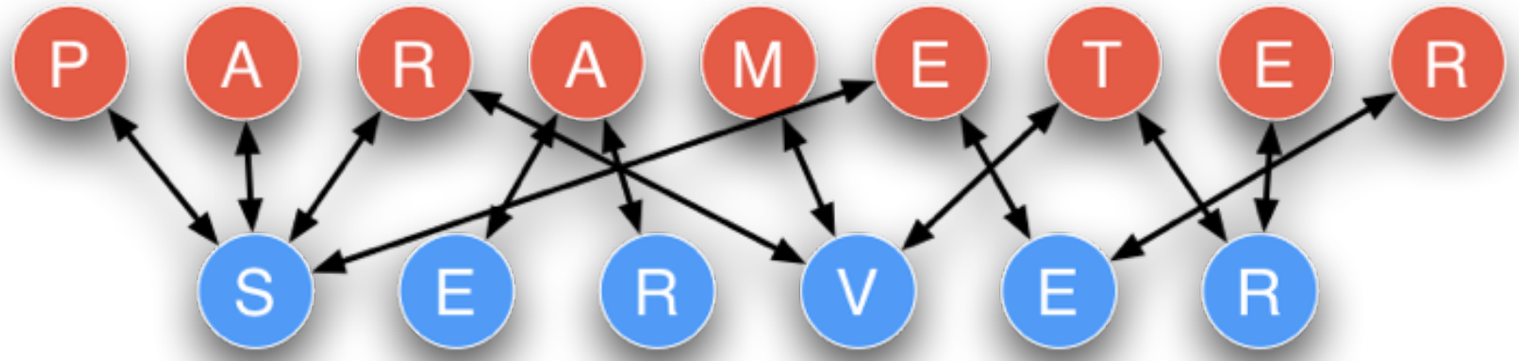
Easy to use

Evaluation

Inc … ne
l… … s

Data and model partition

Relax data consistency:
1. Task dependency graph
2. User defined filters



1. Consistency hashing
2. (Aggregated) chain replication

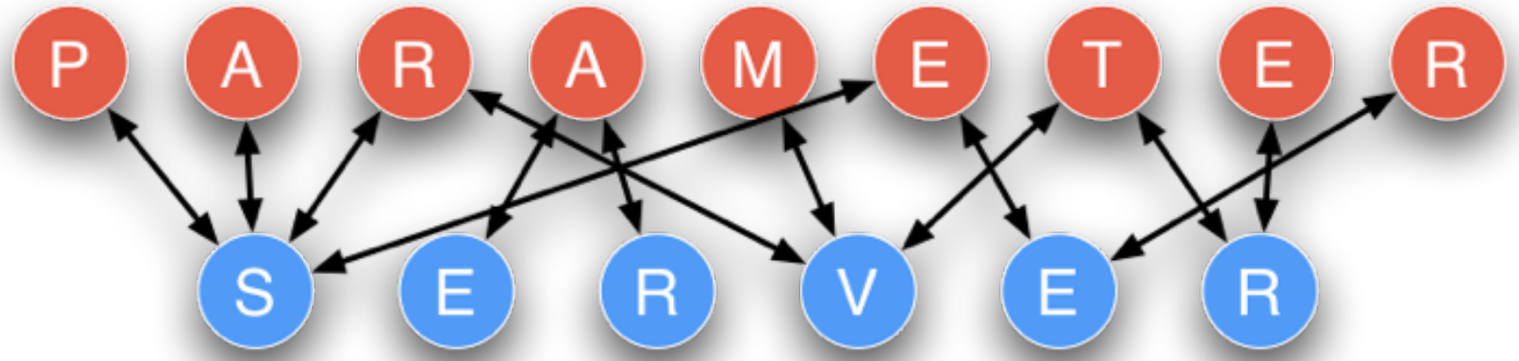
Use as math objects

Evaluation

Inc … ne
l… … s

Data and model partition

Relax data consistency:
1. Task dependency graph
2. User defined filters



1. Consistency hashing
2. (Aggregated) chain replication

Use as math objects

Two large scale applications

Inc … ne
l… … s

Data and model partition

Relax data consistency:
1. Task dependency graph
2. User defined filters

P

Codes are available at parameterserver.org

R

1. Consistency hashing
2. (Aggregated) chain replication

Use as math objects

Two large scale applications