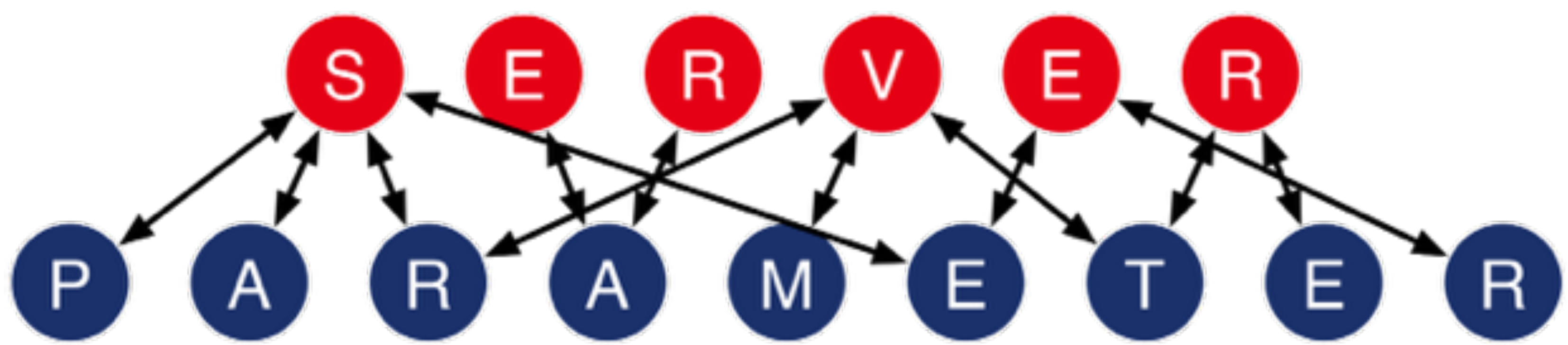
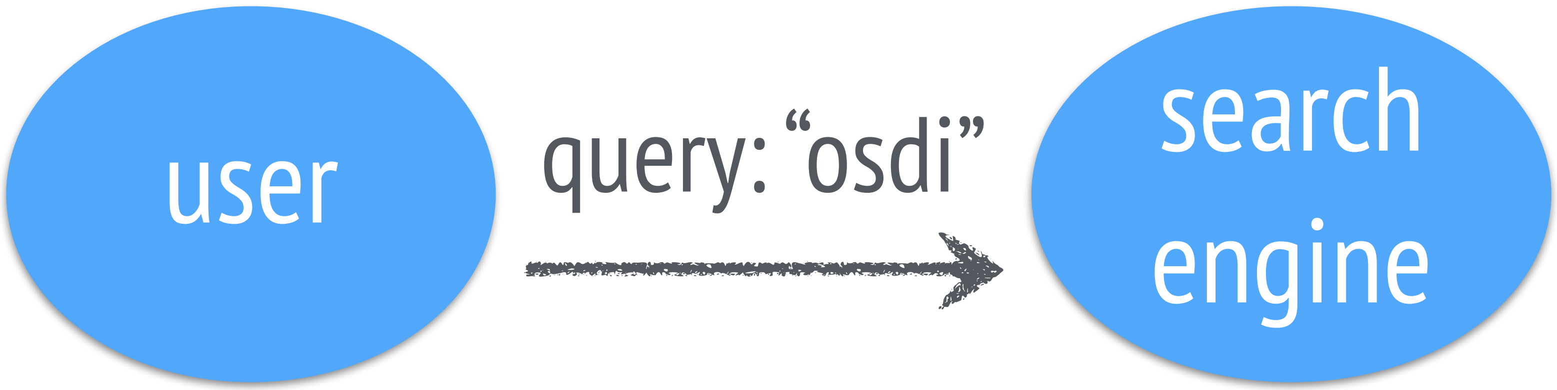
Scaling Distributed Machine Learning with the

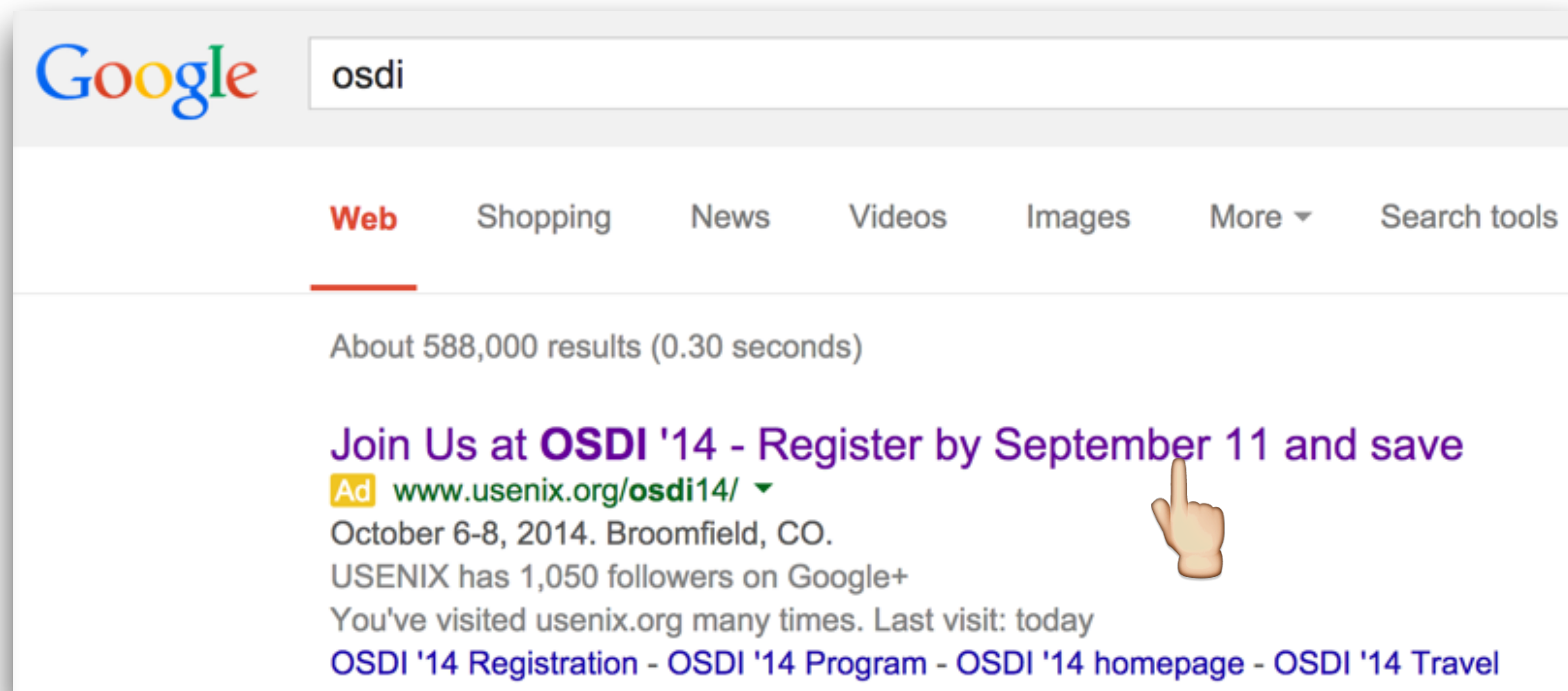


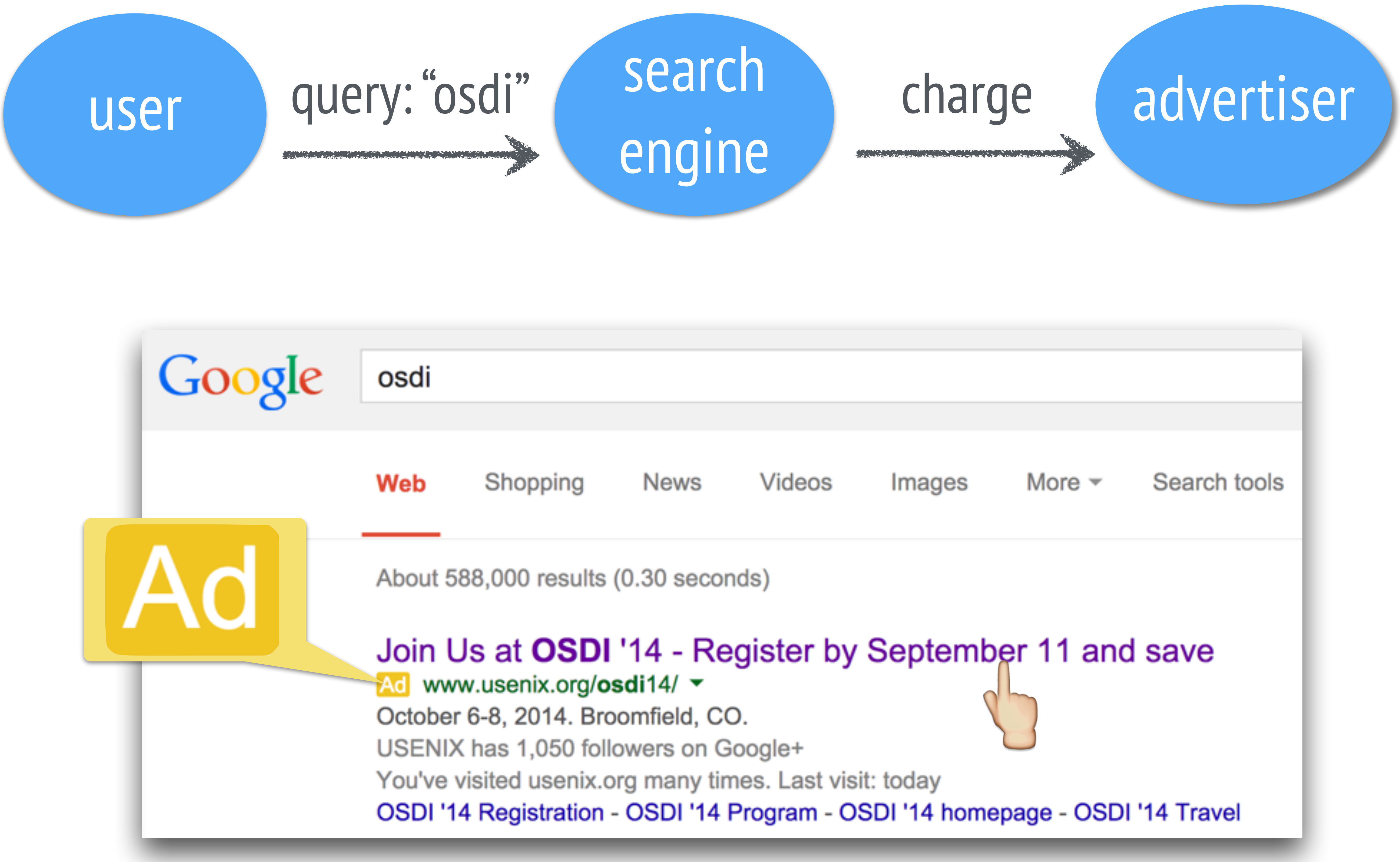
Mu Li

muli@cs.cmu.edu


OSDI '14 | USENIX
https www.usenix.org/conference/osdi14
Reader
back to usenix
Sign in
Conferences
Students
11th USENIX Symposium on Operating Systems Design and Implementation
OSDI 14
OCTOBER 6-8, 2014
BROOMFIELD, CO
usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION
Sponsored by USENIX in cooperation with ACM SIGOPS
Register Now
Home » OSDI '14
+1 13
Tweet 32
Like 24
OSDI '14 Home
Symposium Organizers
At a Glance
Registration Information
Registration Discounts
Venue, Hotel, and Travel
Technical Sessions
Co-Located Workshops
Purchase the Box Set
Activities
Birds-of-a-Feather Sessions
Poster Sessions
Sponsorship
Students and Grants
Questions?
Overview
CONNECT WITH US
Join us in Broomfield, CO, October 6-8, 2014, for the 11th USENIX Symposium on Operating Systems Design and Implementation. OSDI '14 will bring together professionals from academic and industrial backgrounds in what has become the premier forum for discussing the design, implementation, and implications of systems software. The program includes two poster sessions and over 40 paper presentations on data, security, cloud computing, storage, transactions, and much more.
Register Today!
The following workshops are co-located with OSDI '14, and will take place on Sunday, October 5, 2014:
Diversity '14: 2014 Workshop on Supporting Diversity in Systems Research
HotDep '14: 10th Workshop on Hot Topics in System Dependability
HotPower '14: 6th Workshop on Power-Aware Computing and Systems
INFLOW '14: 2nd Workshop on Interactions of NVM/Flash with Operating Systems and Workloads
TRIOS '14: 2014 Conference on Timely Results in Operating Systems

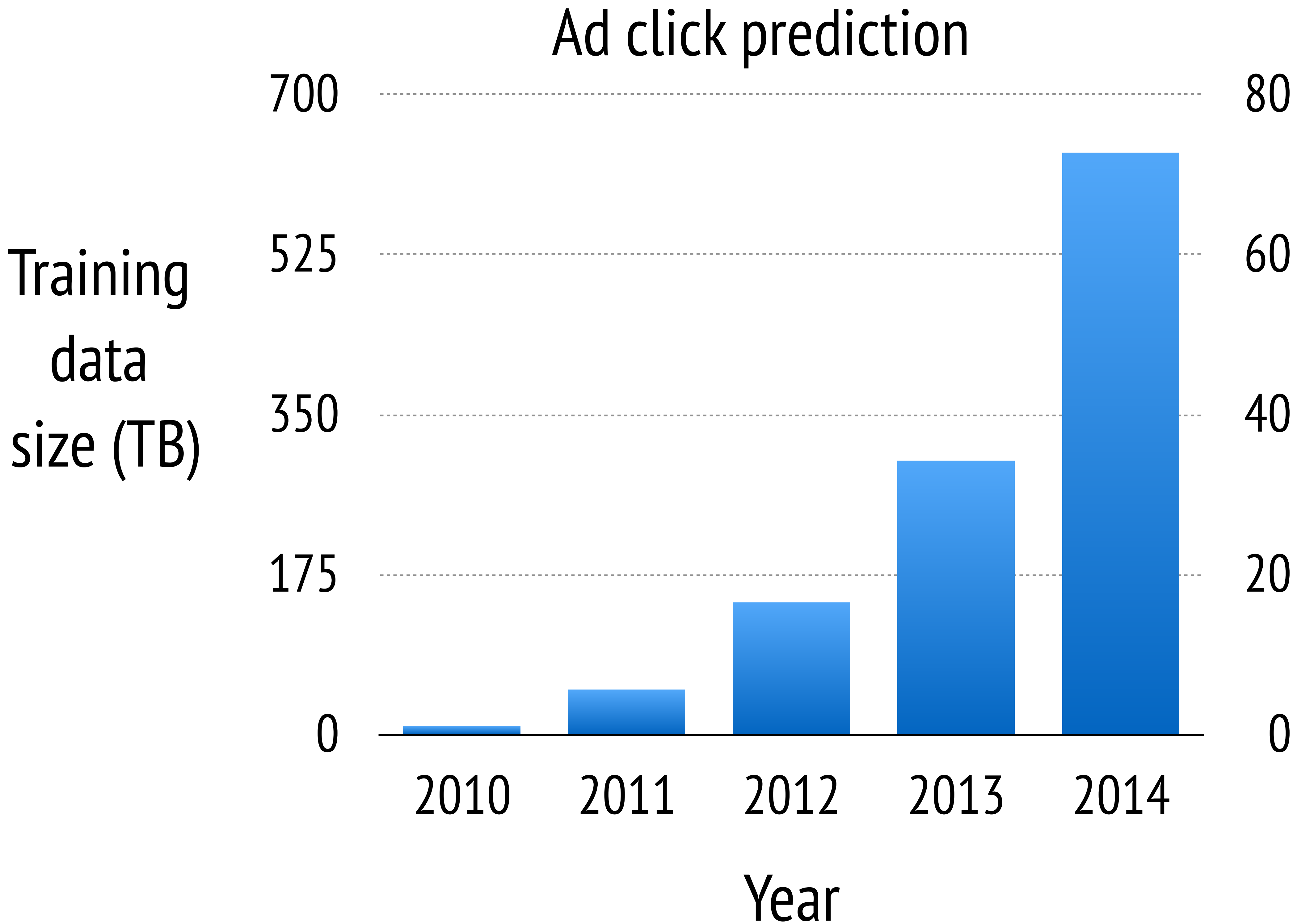

user
query: "osdi"
search engine


Google
osdi
Web
Shopping
News
Videos
Images
More
Search tools
About 588,000 results (0.30 seconds)
Join Us at OSDI '14 - Register by September 11 and save
Ad www.usenix.org/osdi14/
October 6-8, 2014. Broomfield, CO.
USENIX has 1,050 followers on Google+
You've visited usenix.org many times. Last visit: today
OSDI '14 Registration - OSDI '14 Program - OSDI '14 homepage - OSDI '14 Travel

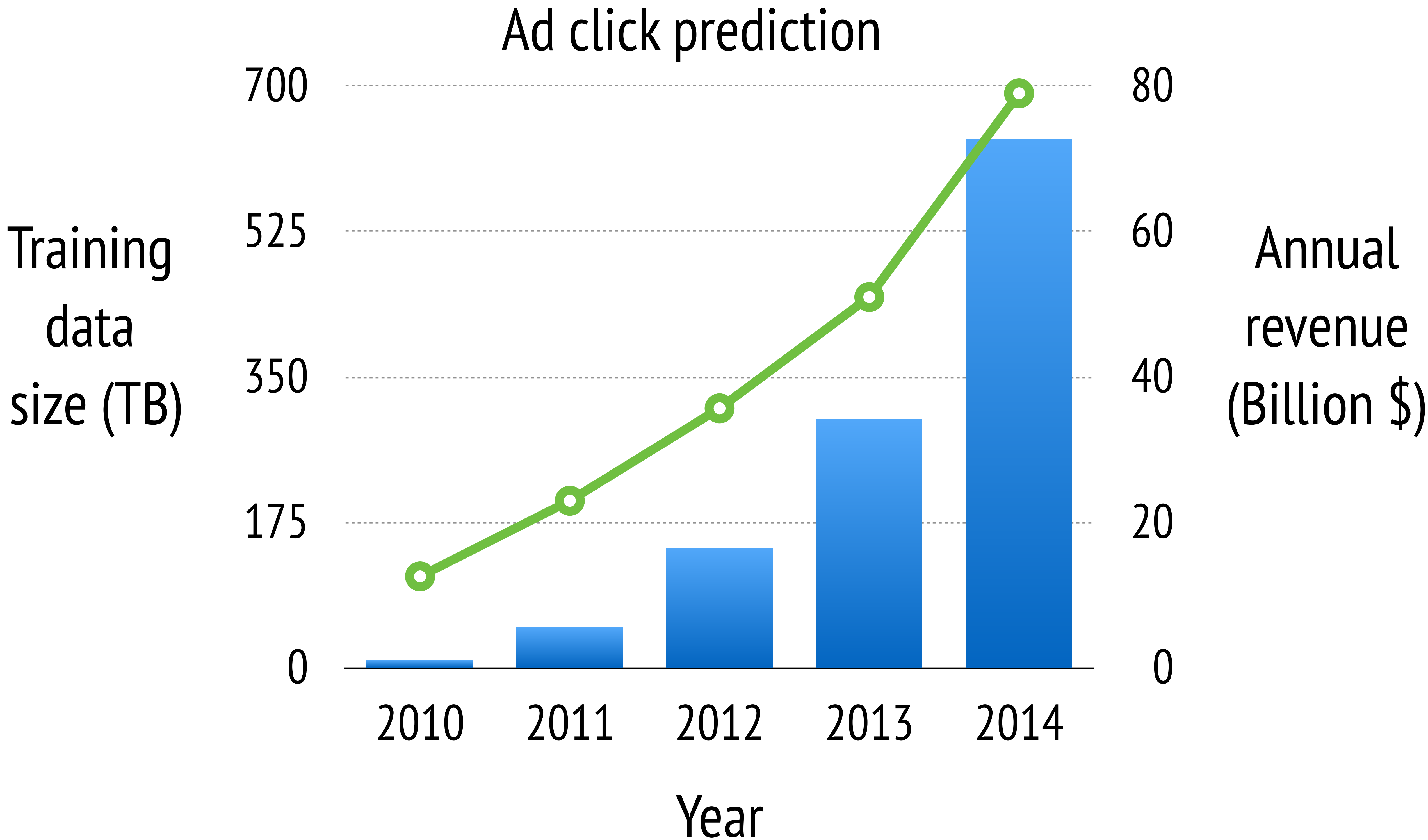

user
query: "osdi"
search engine
charge
advertiser
Google
osdi
Web
Shopping
News
Videos
Images
More
Search tools
Ad
About 588,000 results (0.30 seconds)
Join Us at OSDI '14 - Register by September 11 and save
Ad www.usenix.org/osdi14/
October 6-8, 2014. Broomfield, CO.
USENIX has 1,050 followers on Google+
You've visited usenix.org many times. Last visit: today
OSDI '14 Registration - OSDI '14 Program - OSDI '14 homepage - OSDI '14 Travel

Machine learning is concerned with systems that can learn from data

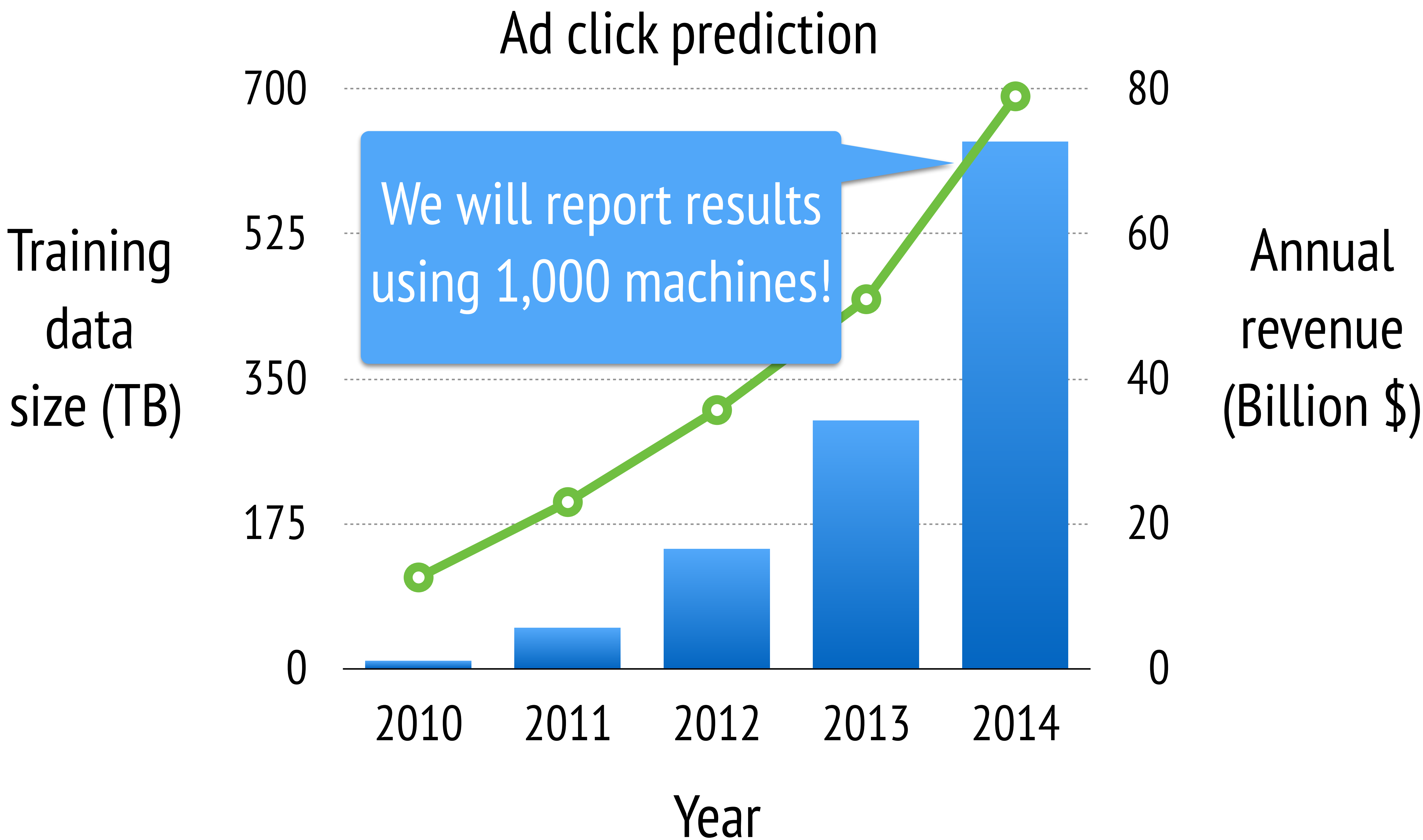
Machine learning is concerned with systems that can learn from data



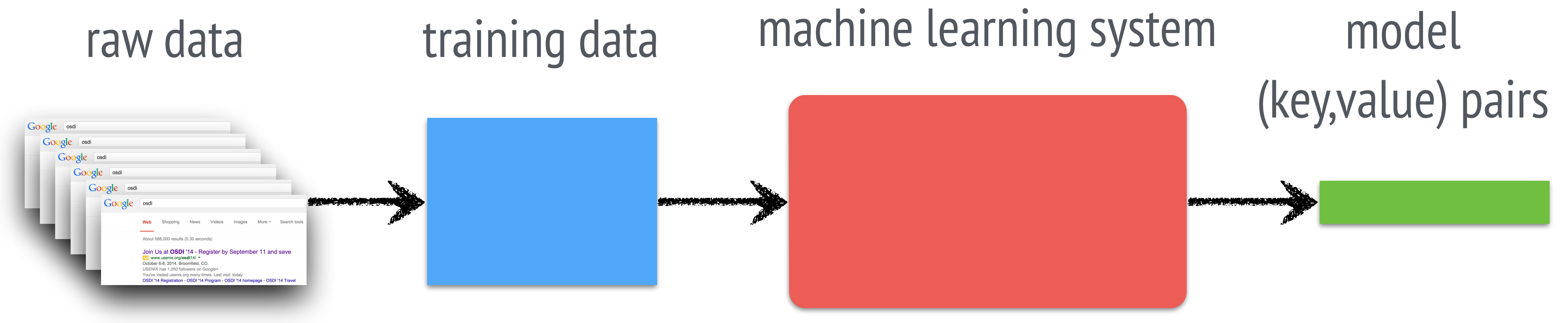
Machine learning is concerned with systems that can learn from data



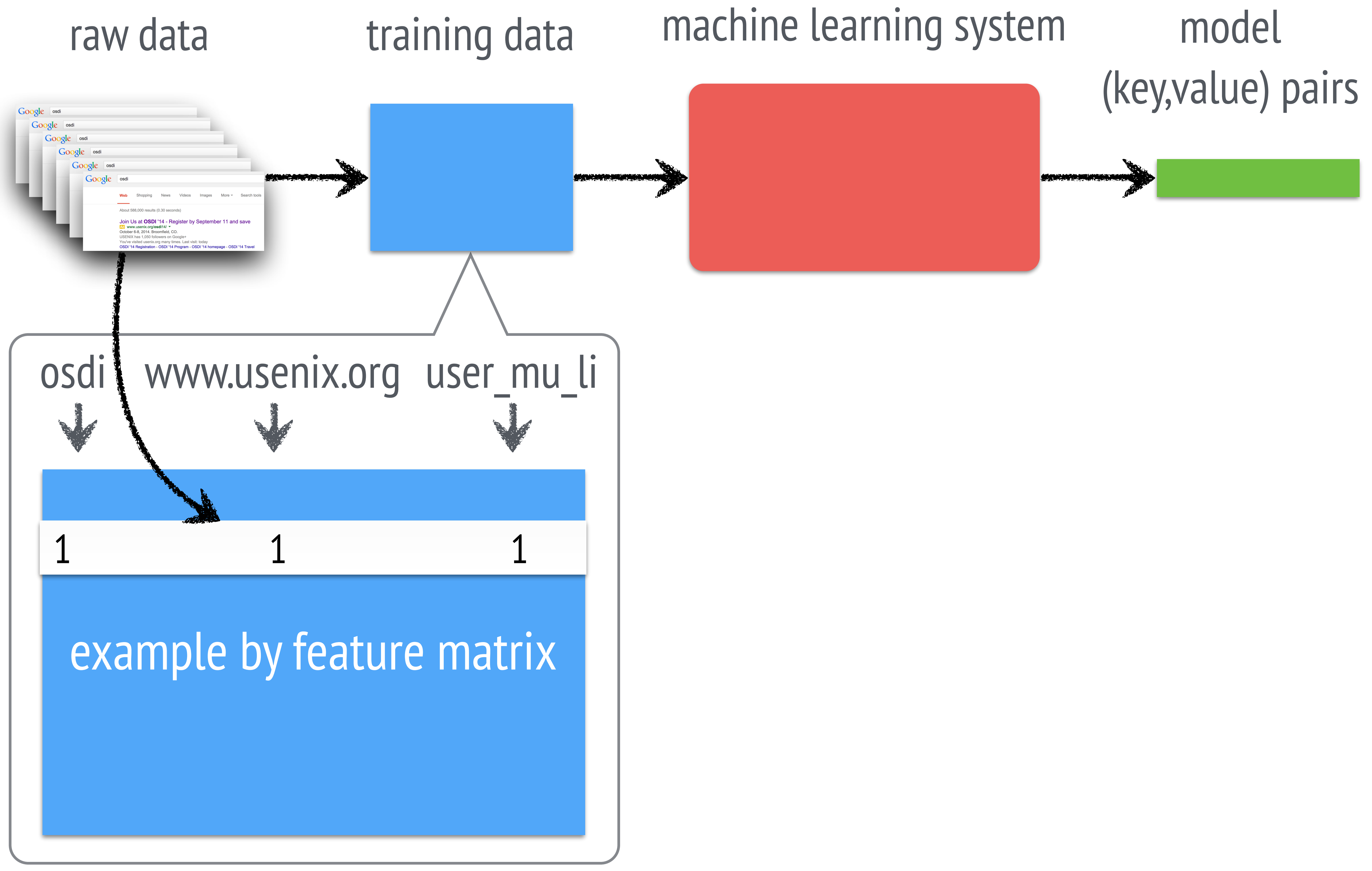
Machine learning is concerned with systems that can learn from data



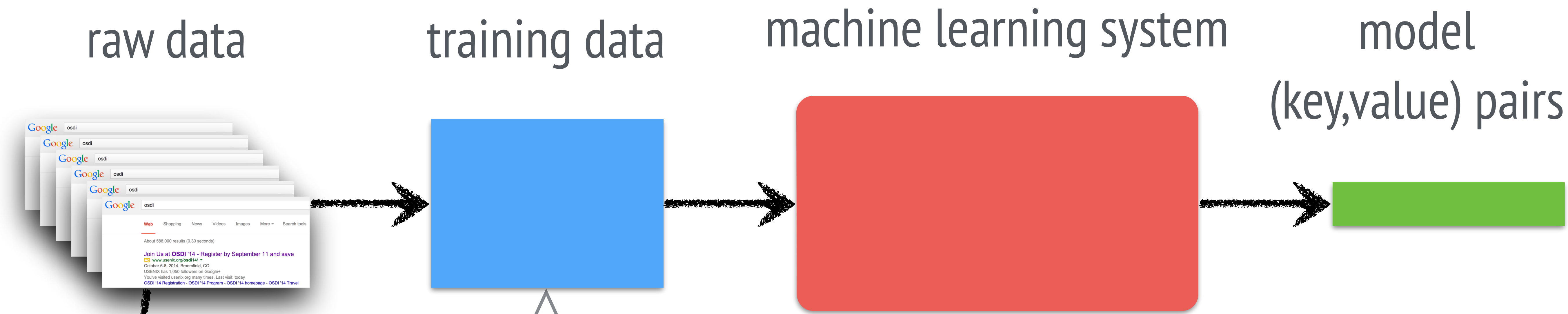
Overview of machine learning



Overview of machine learning



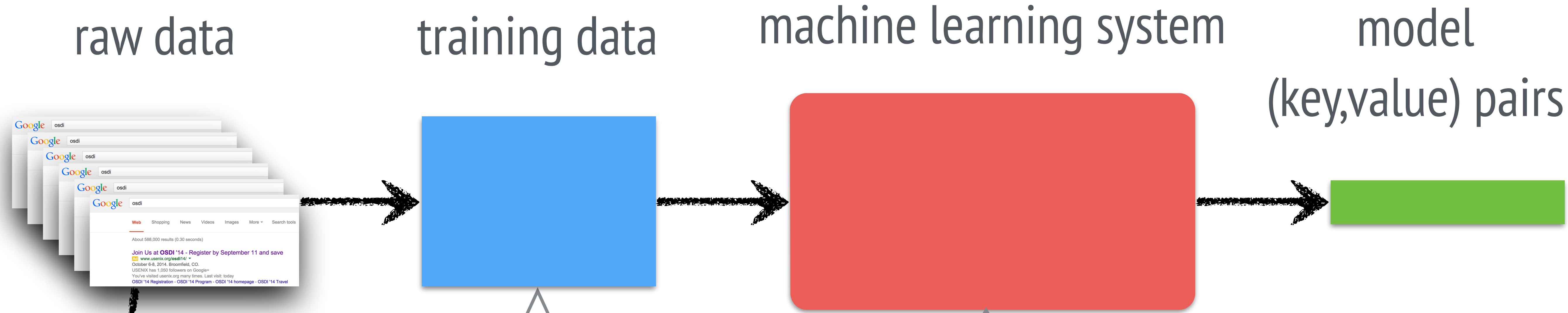
Overview of machine learning



Scale of Industry problems

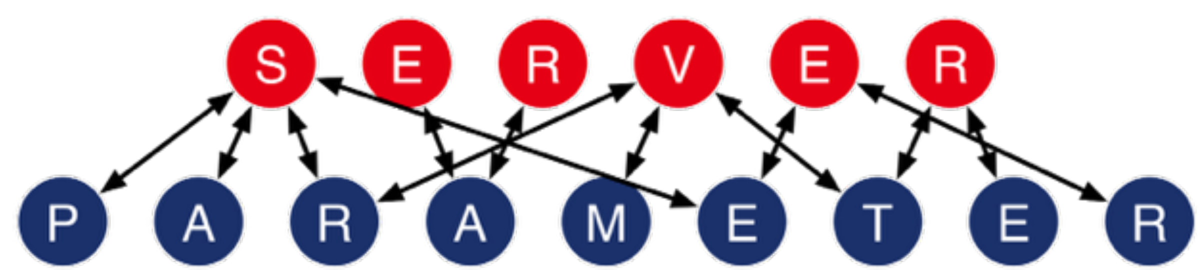
- 100 billion examples
- 10 billion features
- 1T – 1P training data
- 100–1000 machines

Overview of machine learning



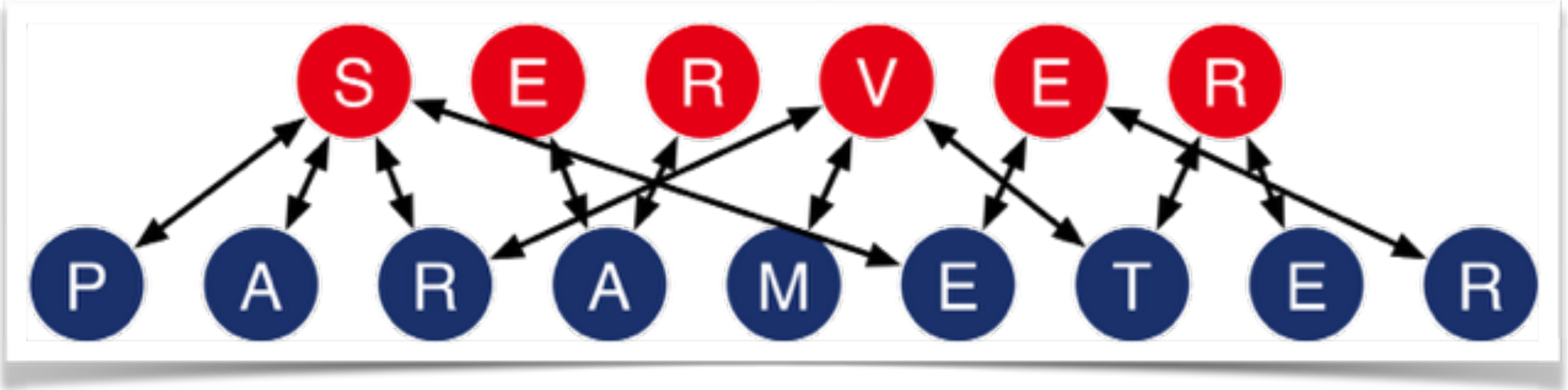
Scale of Industry problems

- 100 billion examples
- 10 billion features
- 1T – 1P training data
- 100 – 1000 machines



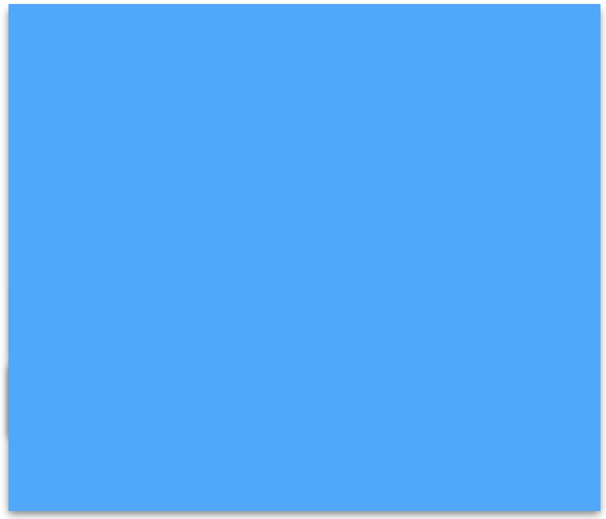
- scale to industry problems
- efficient communication
- fault tolerance
- easy to use

Industry size machine learning problems

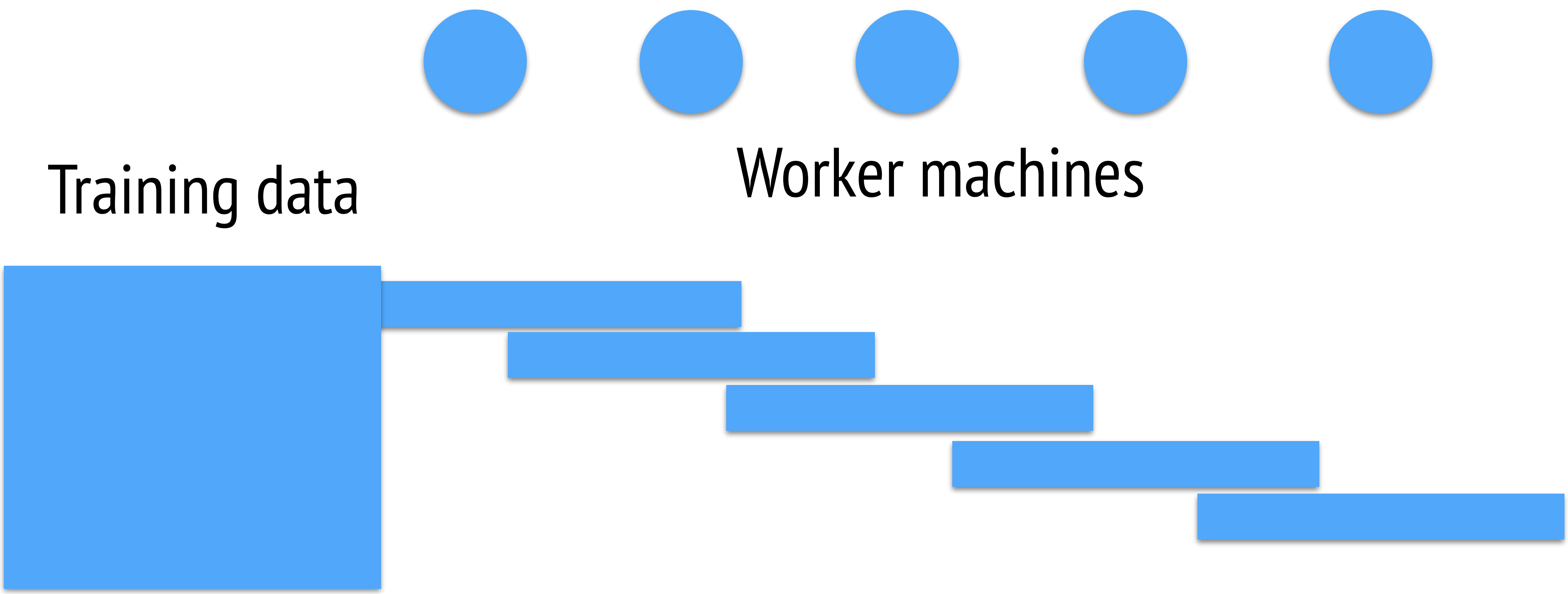


Data and model partition

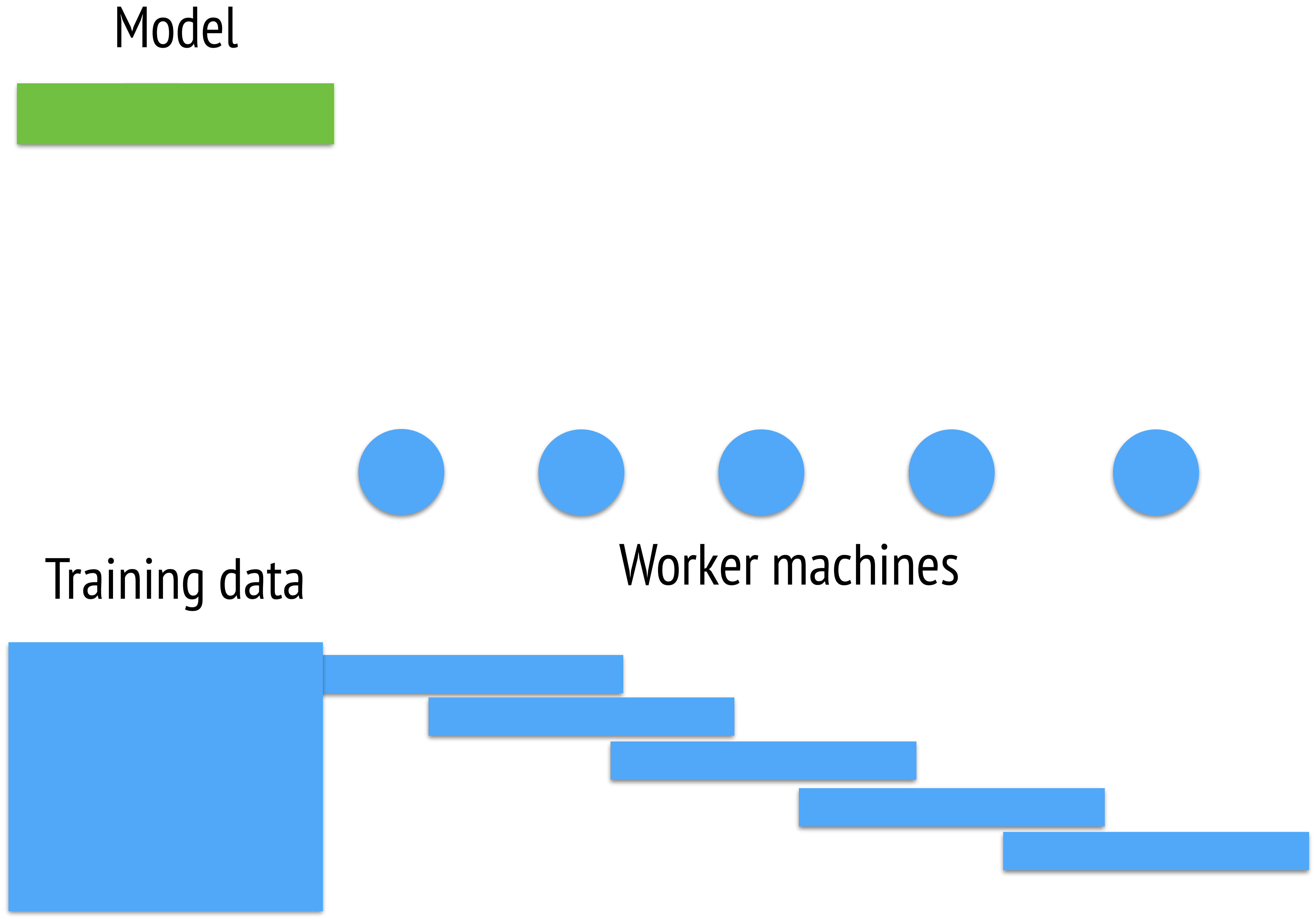
Training data



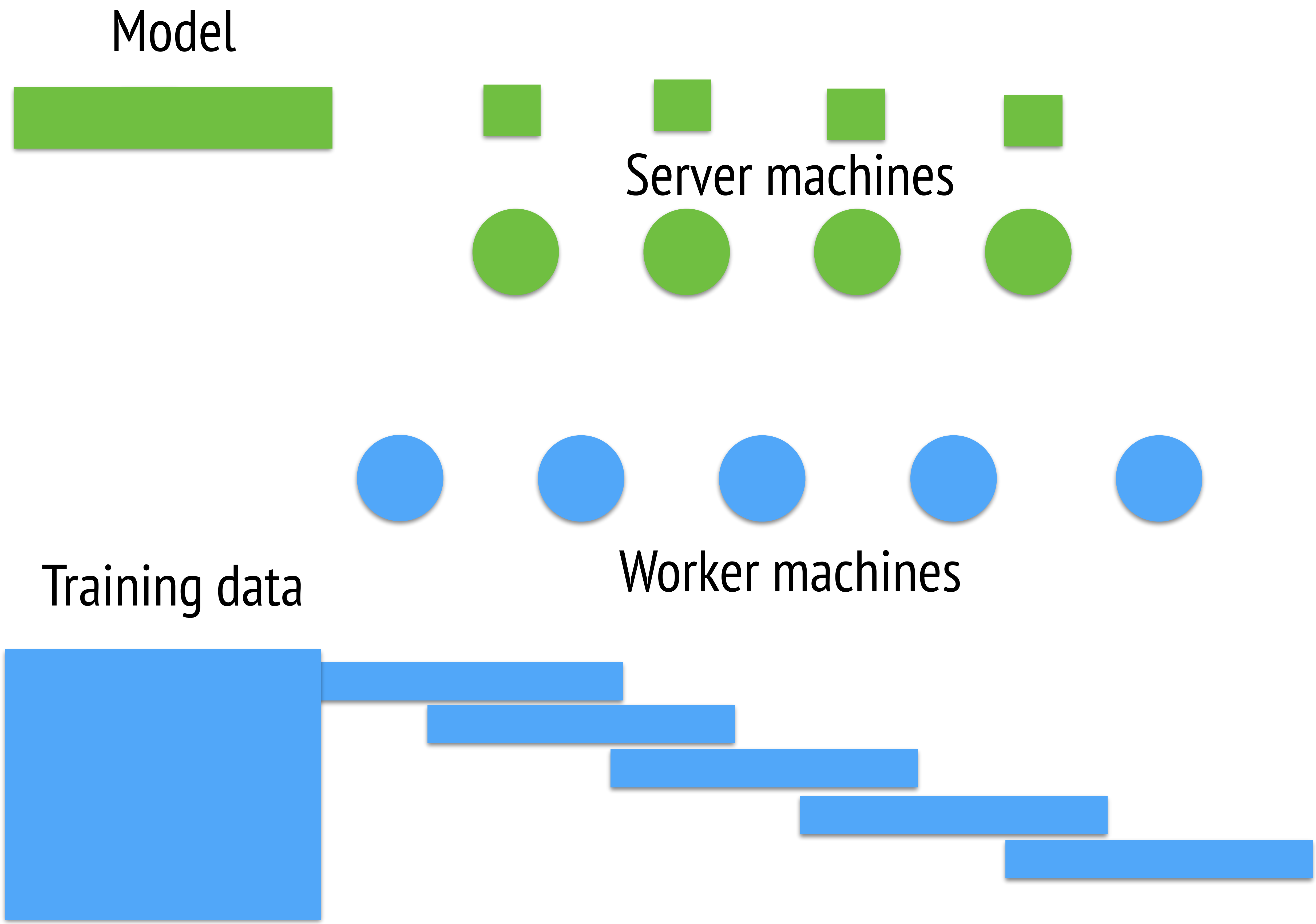
Data and model partition



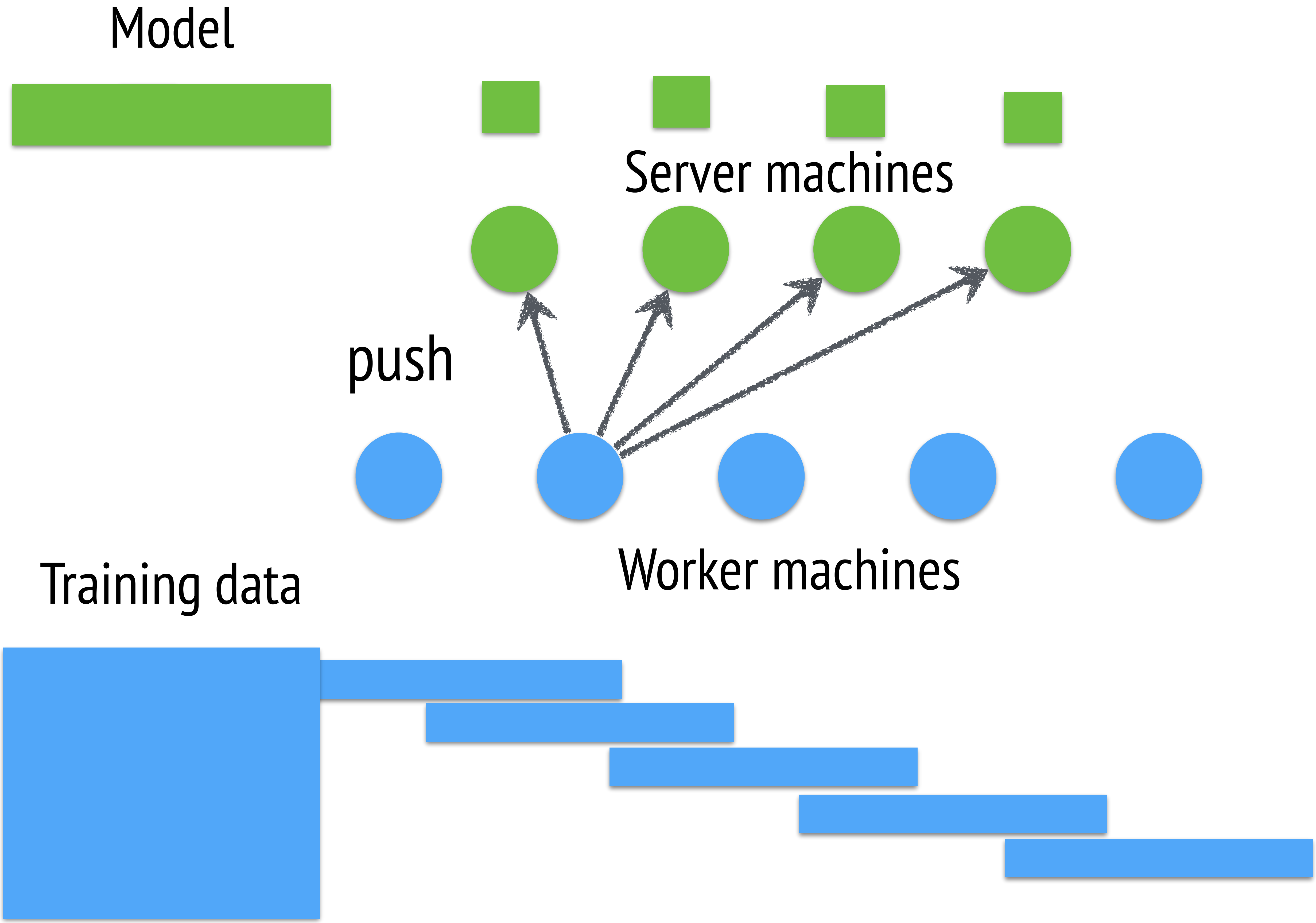
Data and model partition



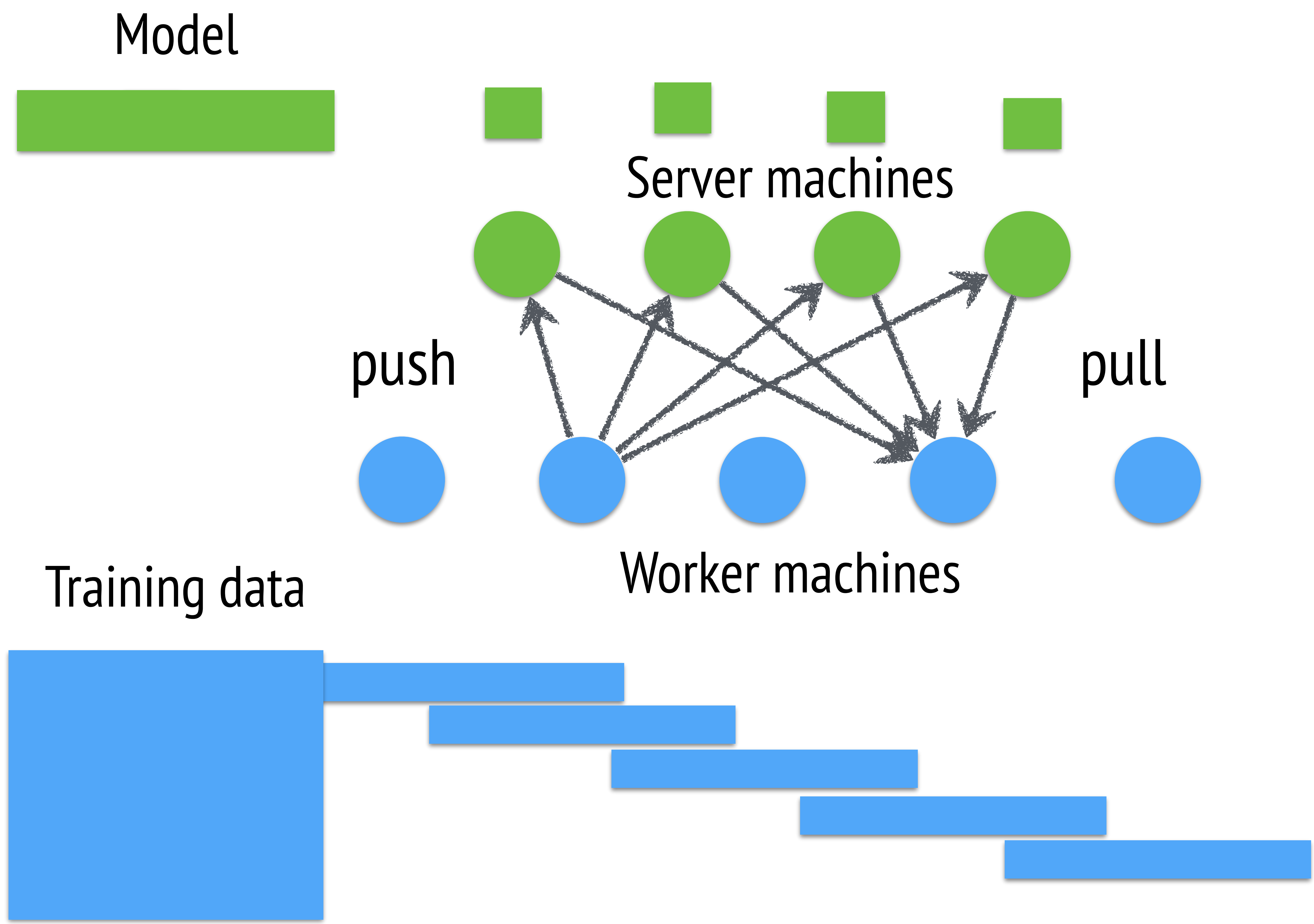
Data and model partition



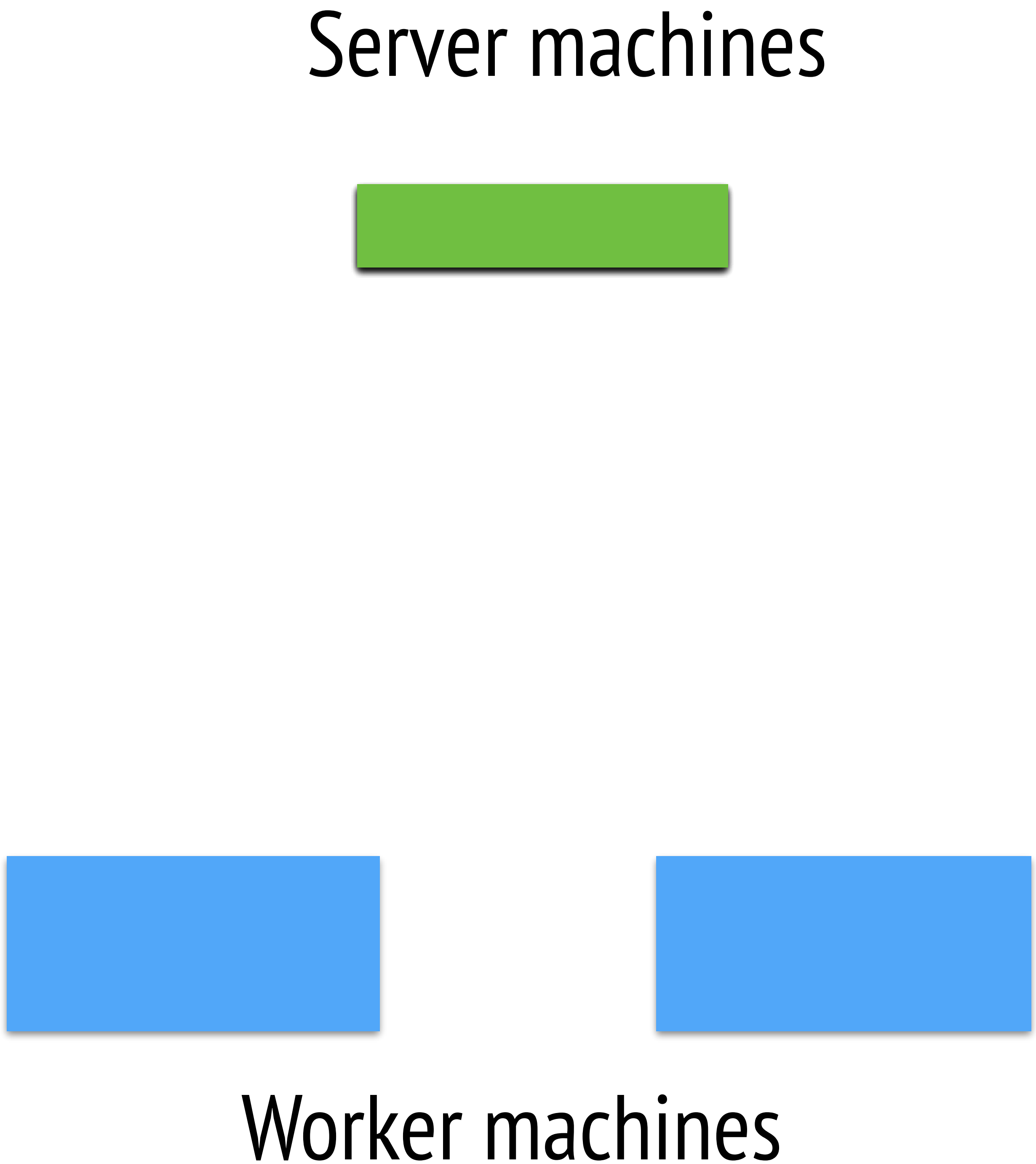
Data and model partition



Data and model partition

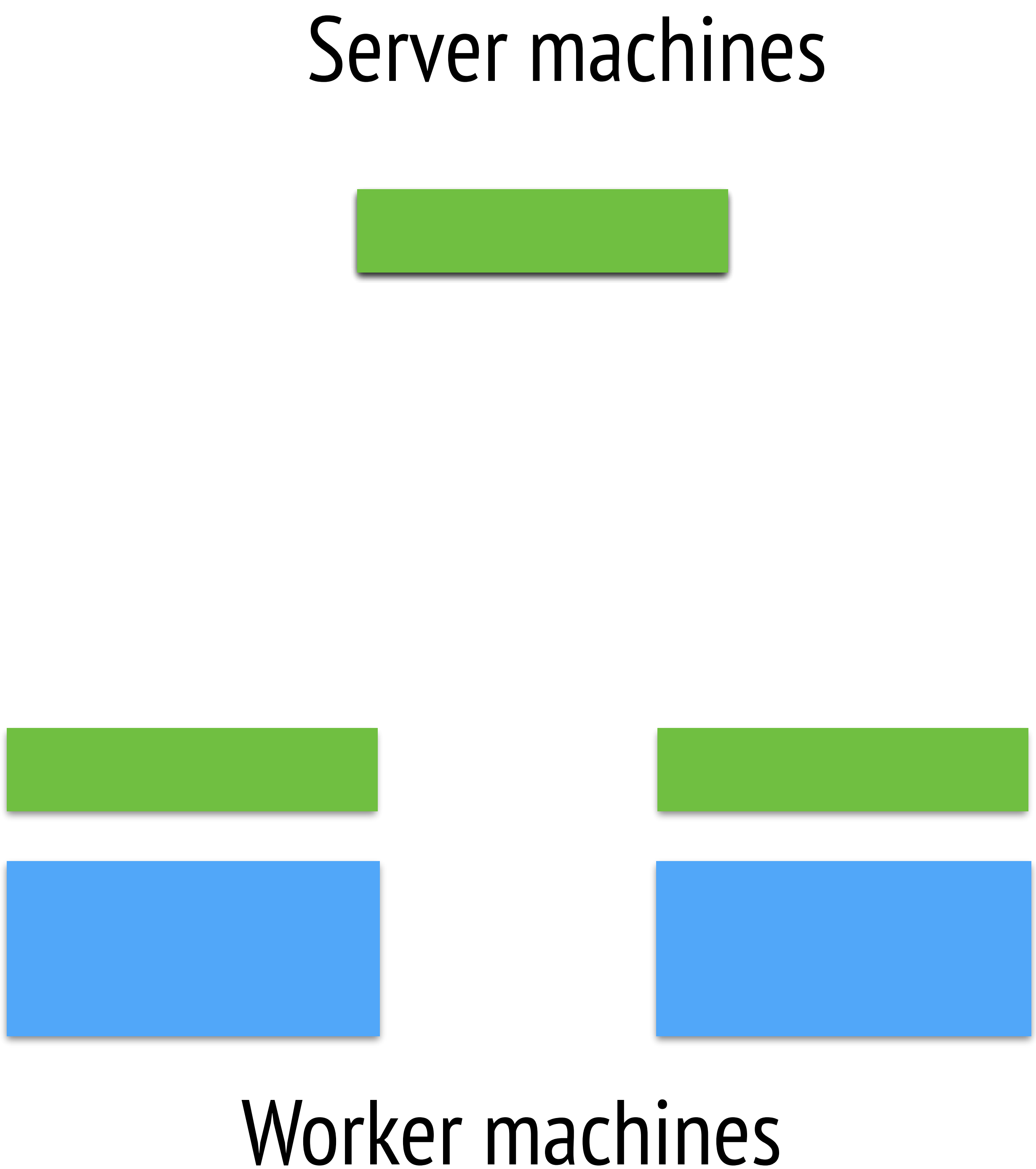


Example: distributed gradient descent



Example: distributed gradient descent

Workers **pull** the working set of model

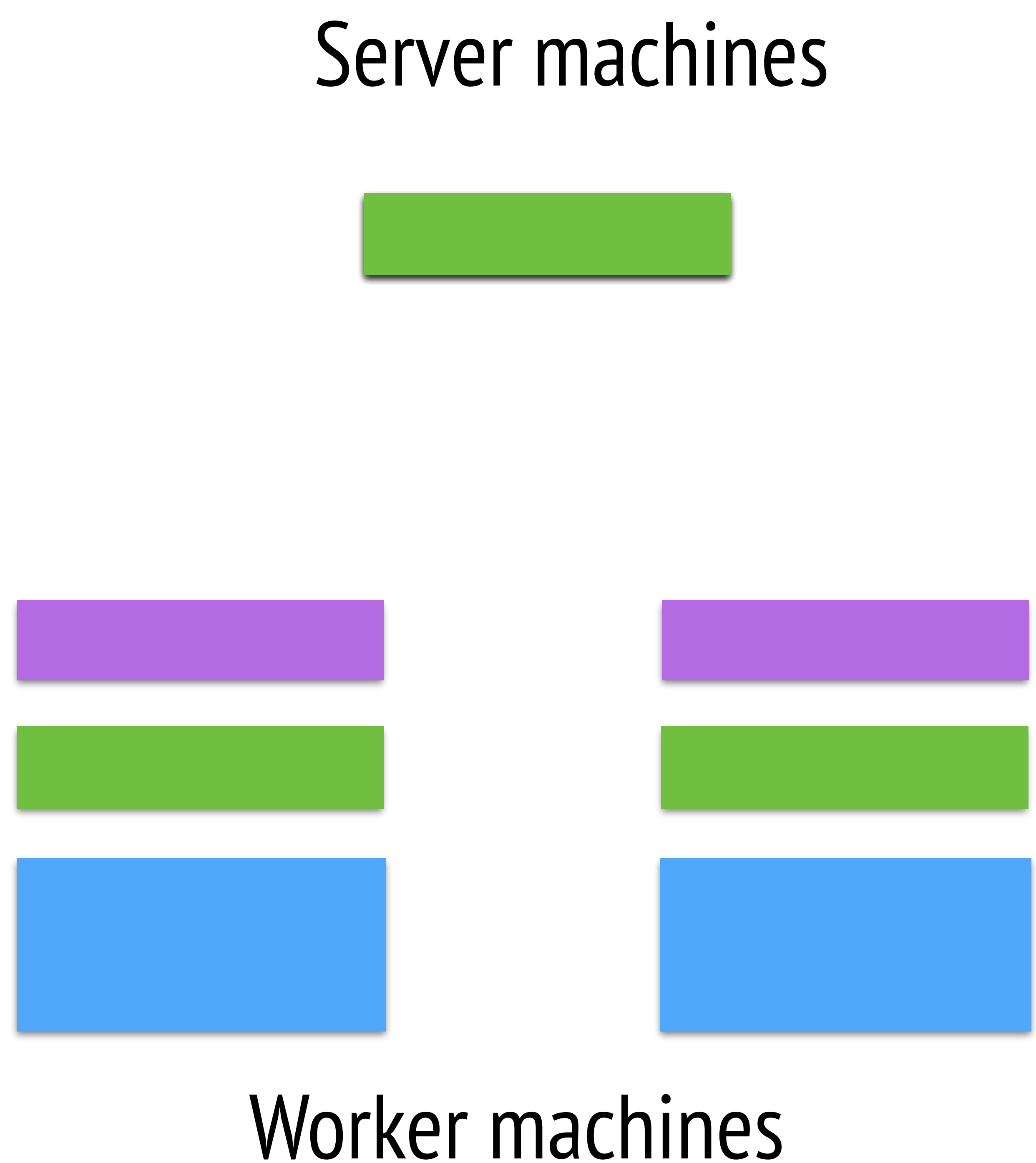


Example: distributed gradient descent

Workers **pull** the working set of model

Iterate until stop

workers compute gradients



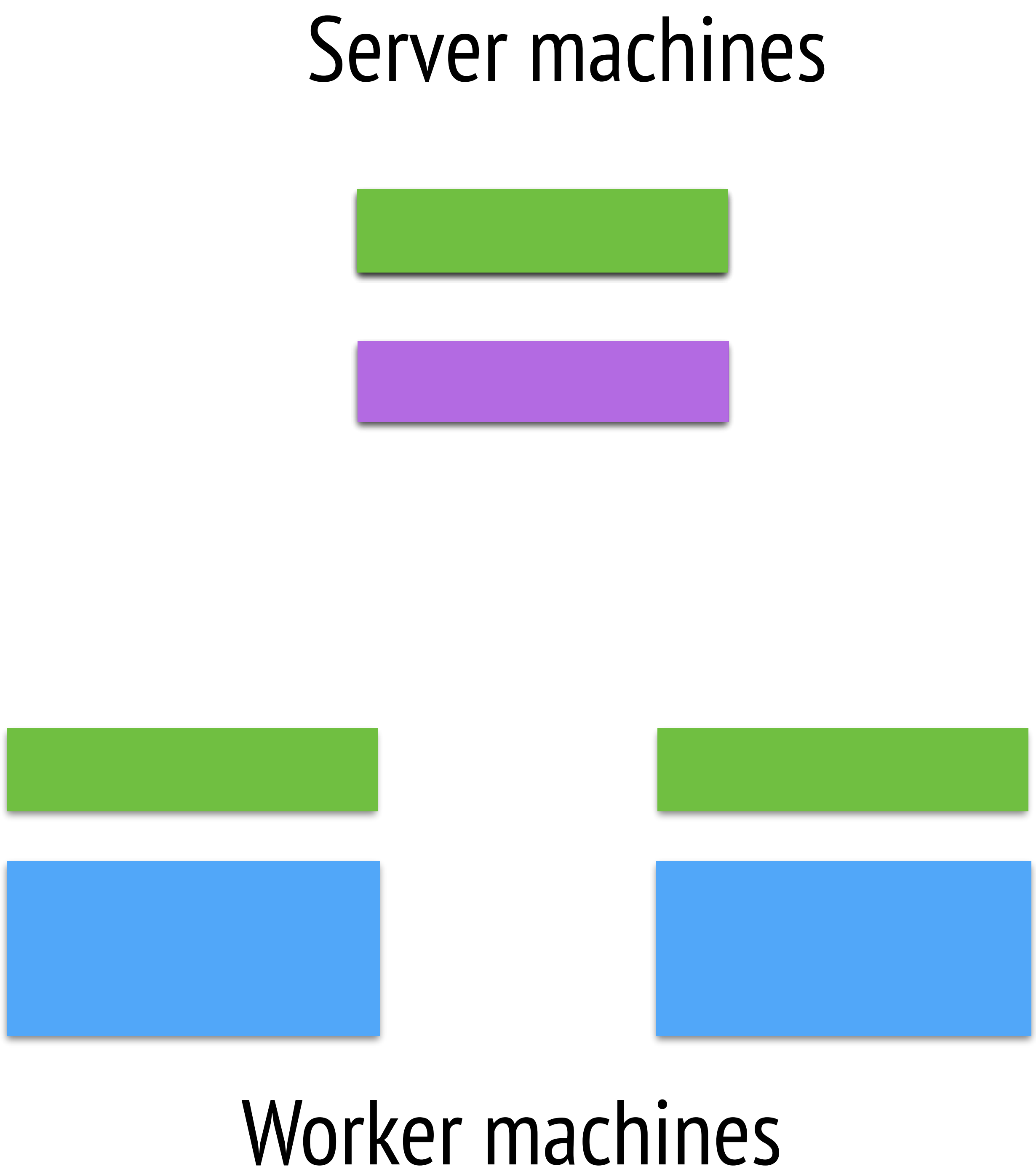
Example: distributed gradient descent

Workers **pull** the working set of model

Iterate until stop

workers compute gradients

workers **push** gradients

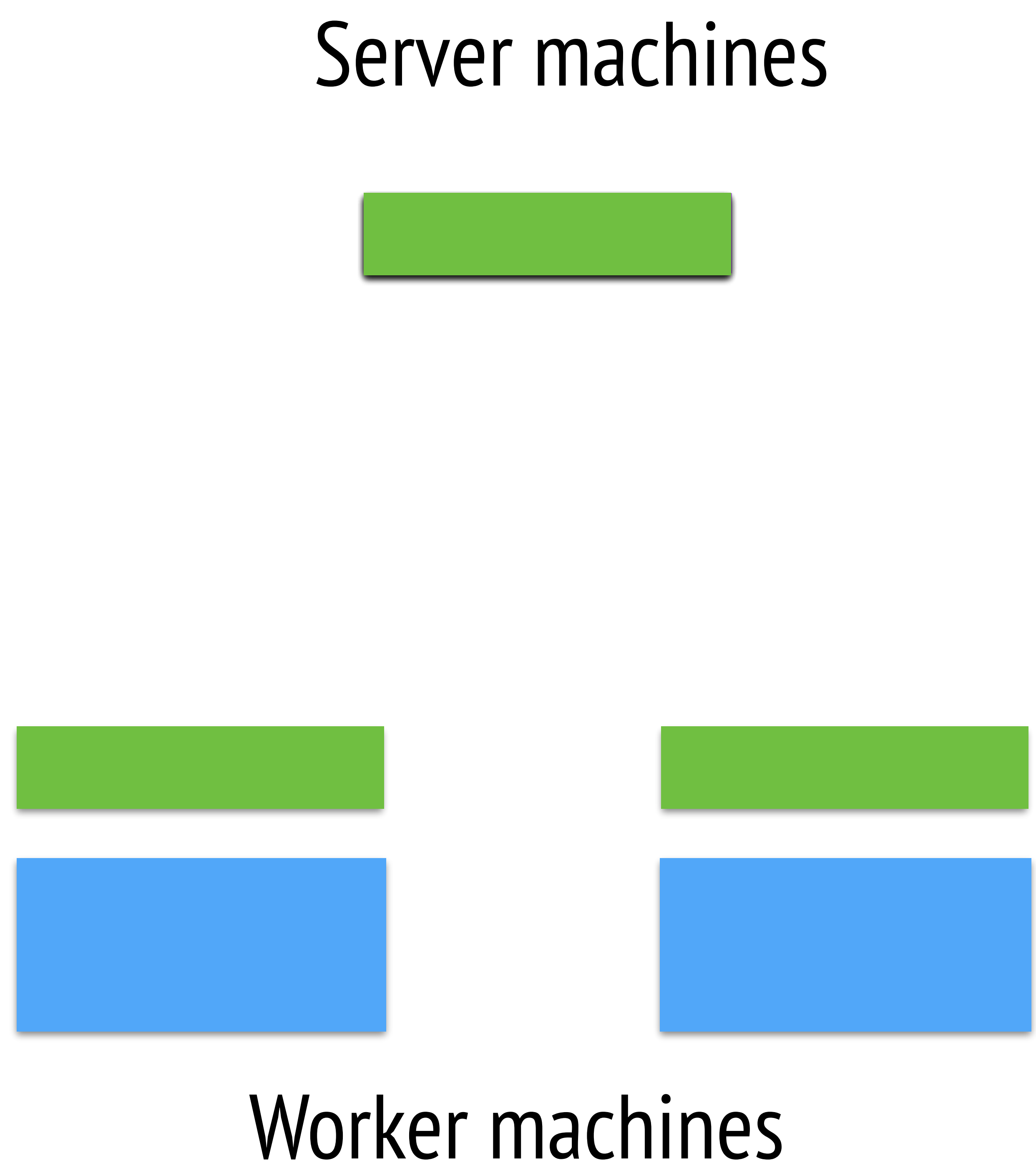


Example: distributed gradient descent

Workers **pull** the working set of model

Iterate until stop

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

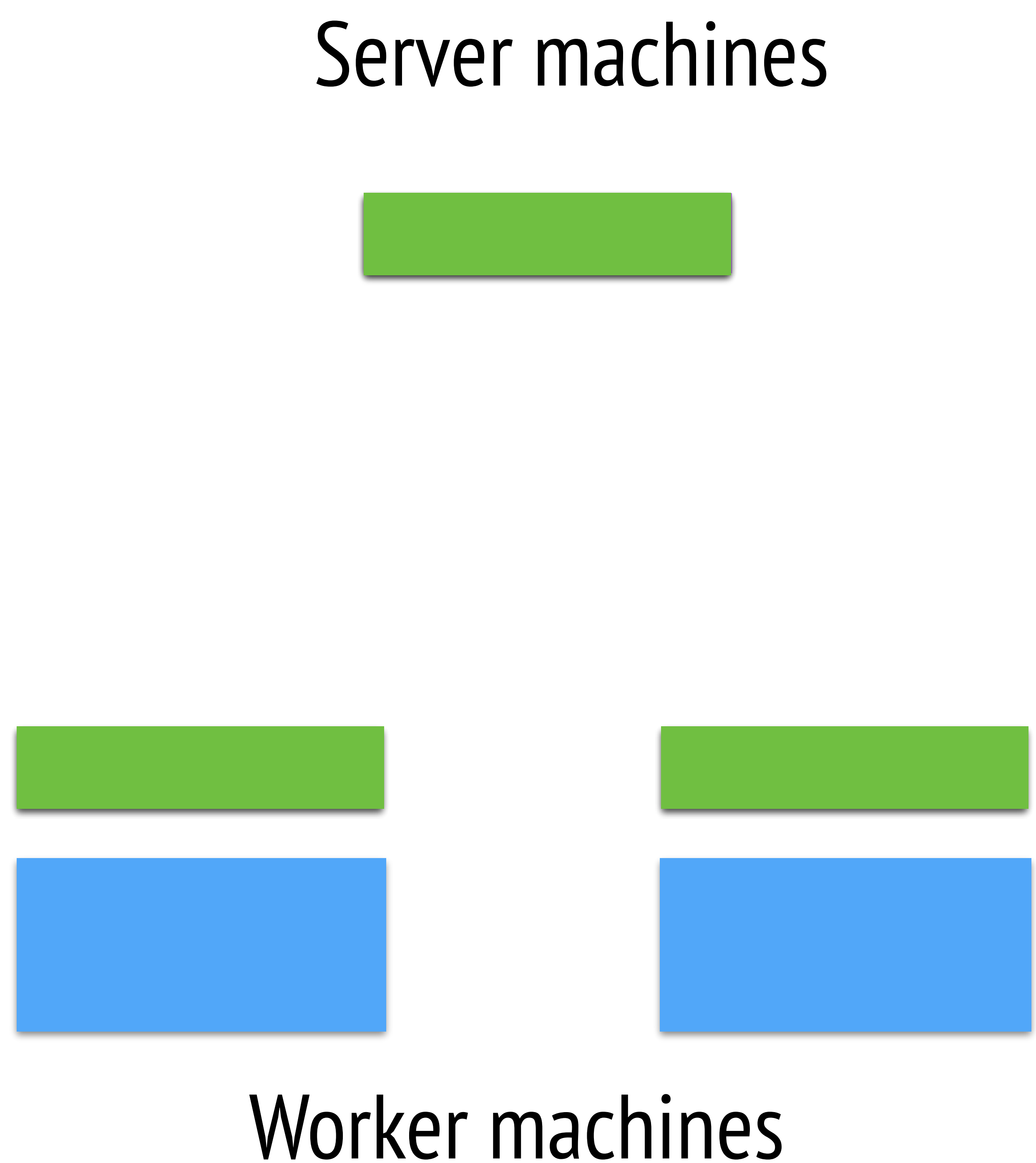


Example: distributed gradient descent

Workers **pull** the working set of model

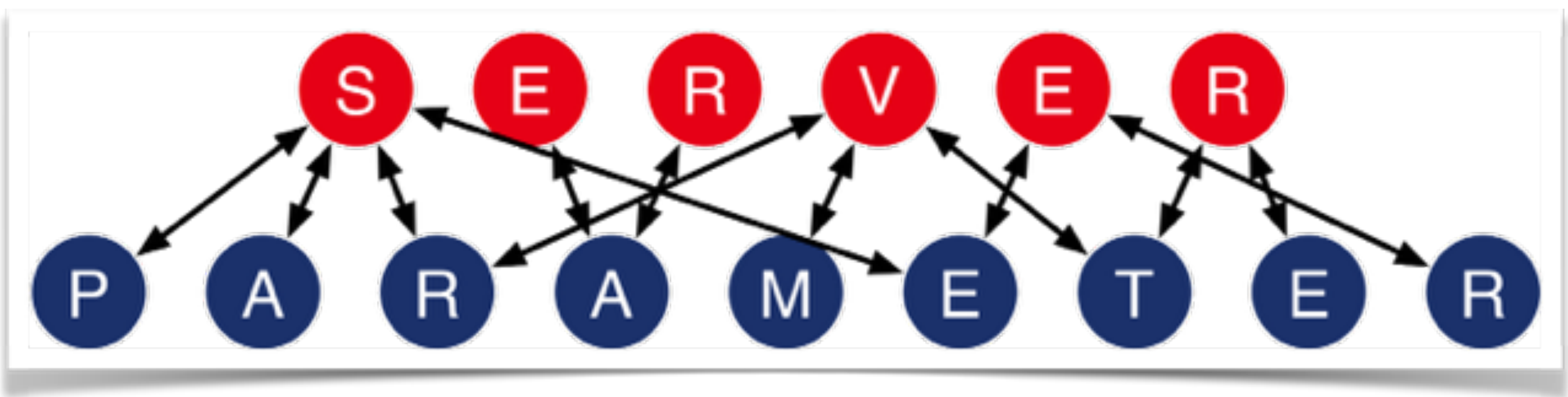
Iterate until stop

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



Industry size machine learning problems

Efficient communication



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

Task

Task

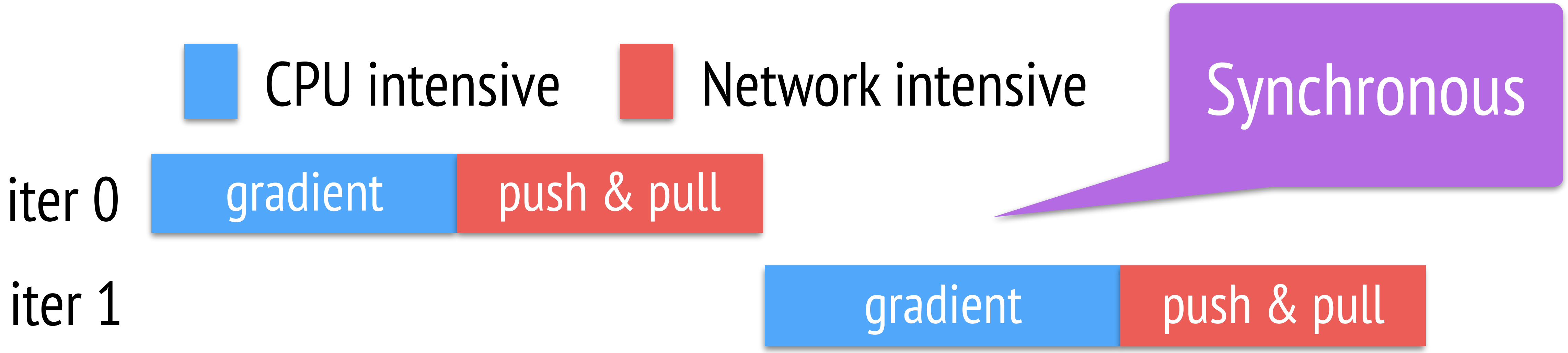
- a push / pull / user defined function (an iteration)

Task

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

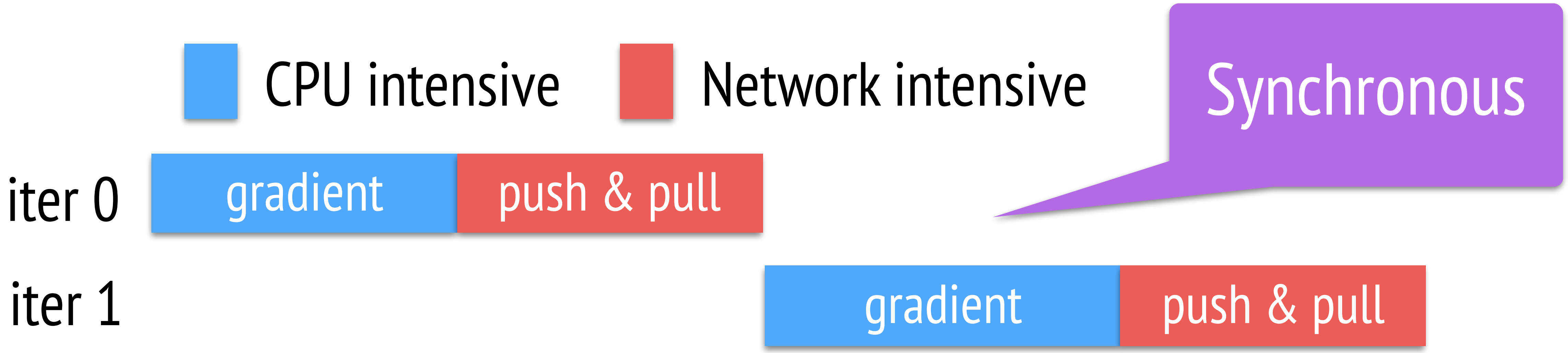
Task

- a push / pull / user defined function (an iteration)
- "execute-after-finished" dependency



Task

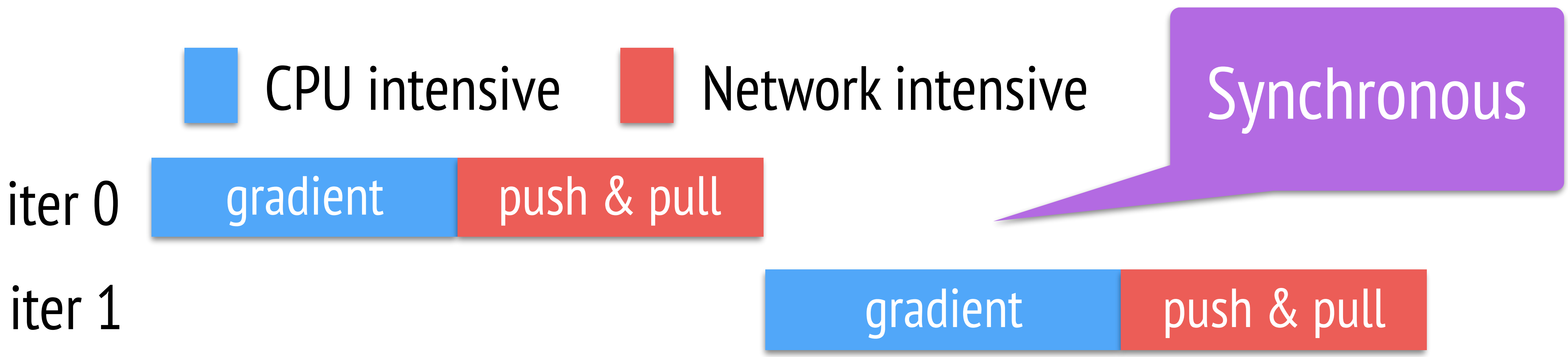
- a push / pull / user defined function (an iteration)
- “execute-after-finished” dependency



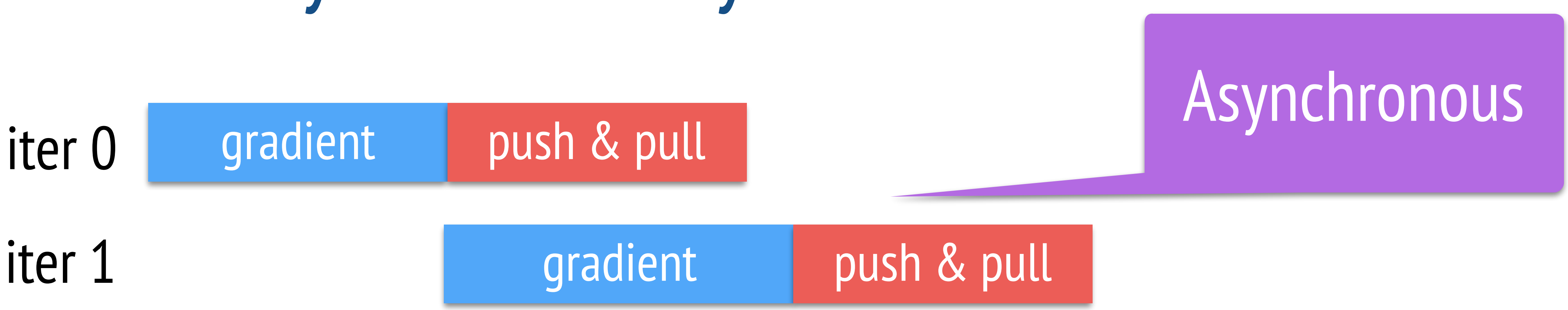
- executed asynchronously

Task

- a push / pull / user defined function (an iteration)
- "execute-after-finished" dependency



- executed asynchronously



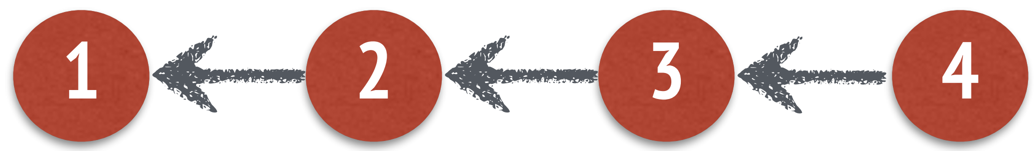
Flexible consistency

- Trade-off between algorithm efficiency and system performance

Flexible consistency

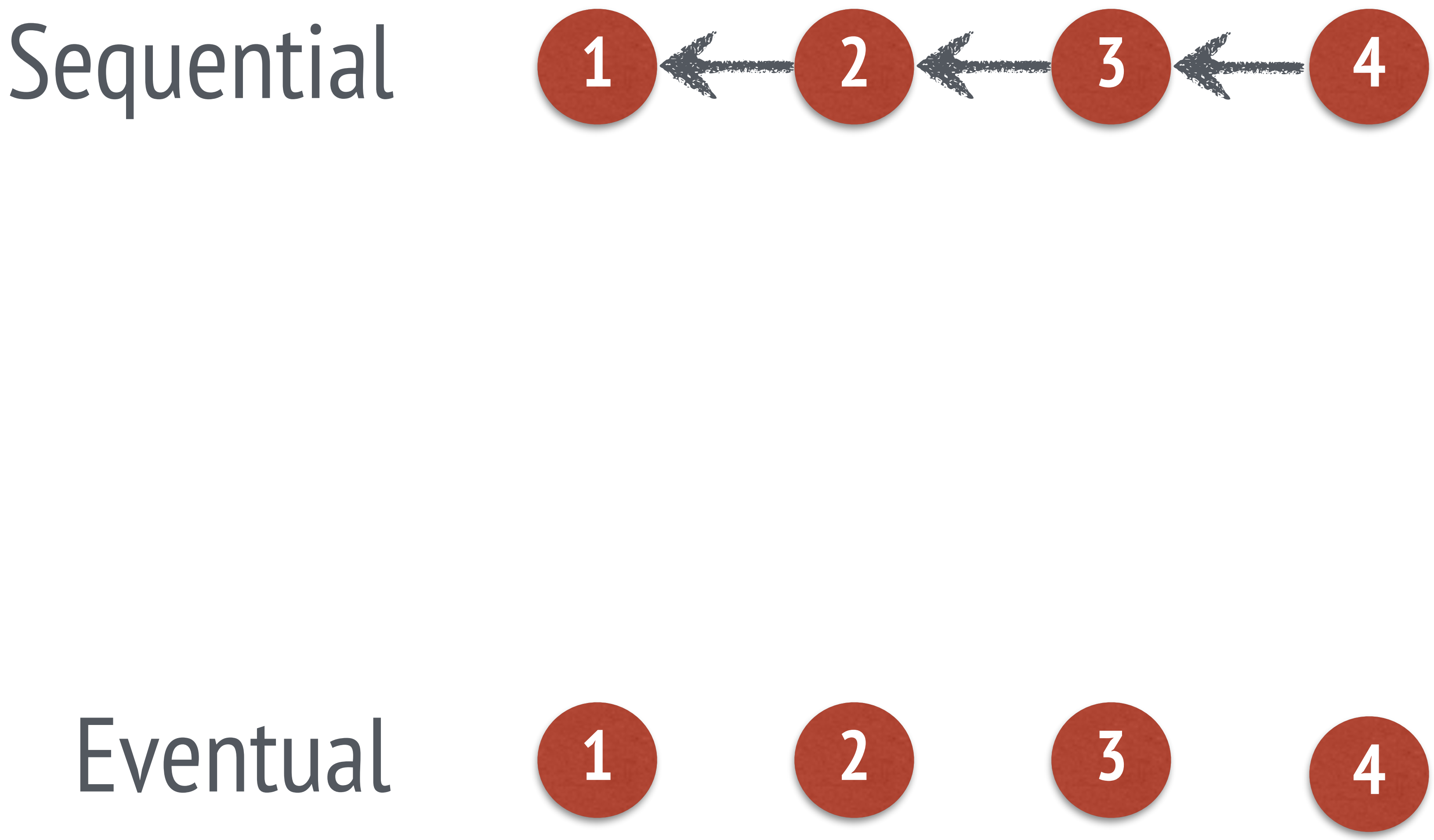
- Trade-off between algorithm efficiency and system performance

Sequential



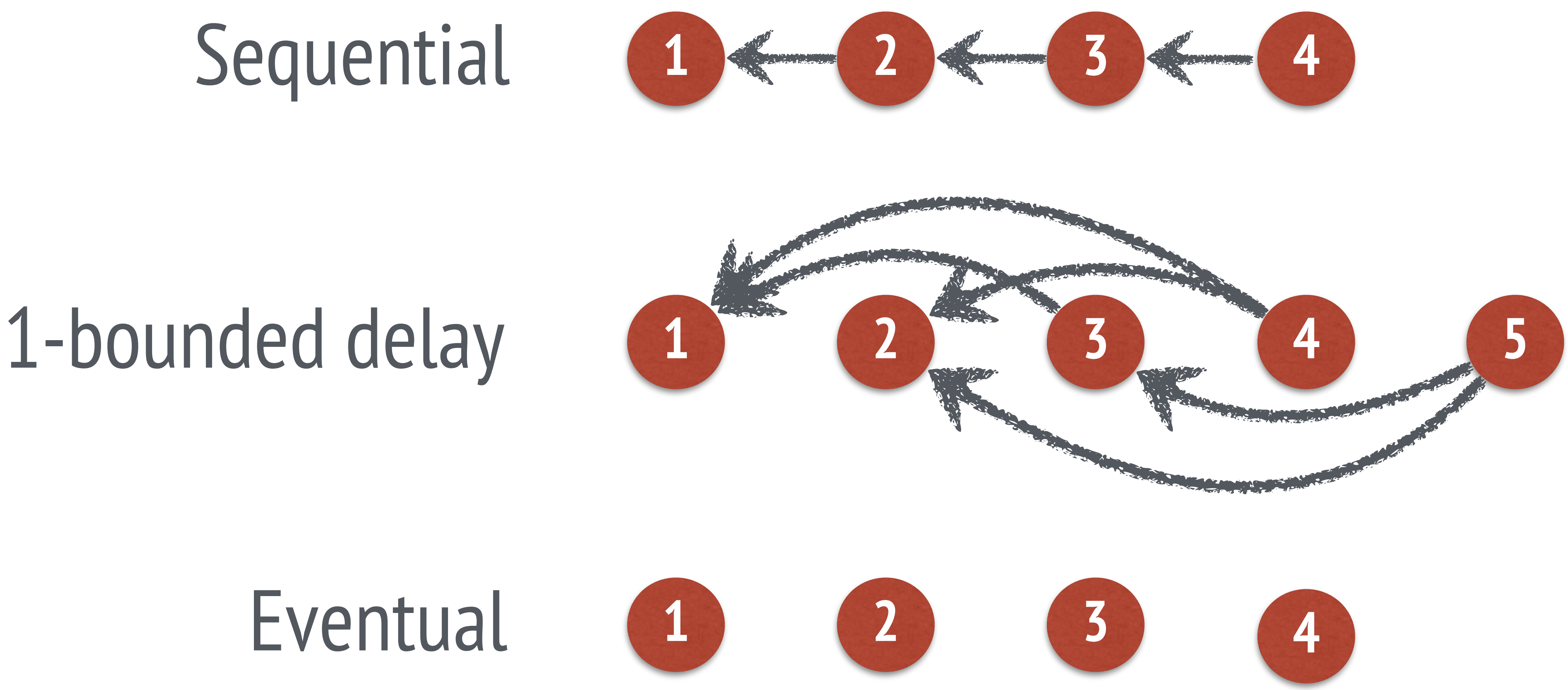
Flexible consistency

- Trade-off between algorithm efficiency and system performance



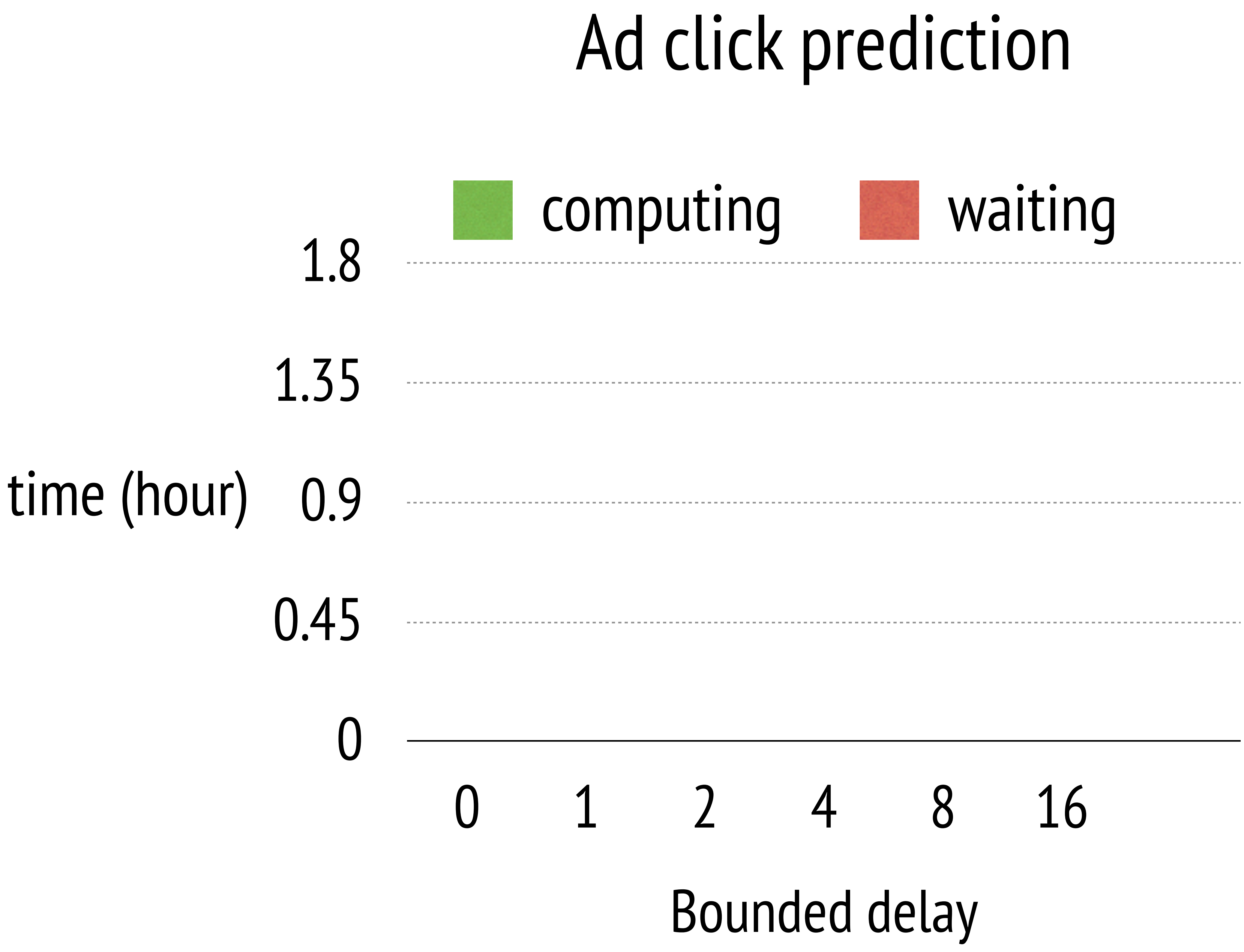
Flexible consistency

- Trade-off between algorithm efficiency and system performance

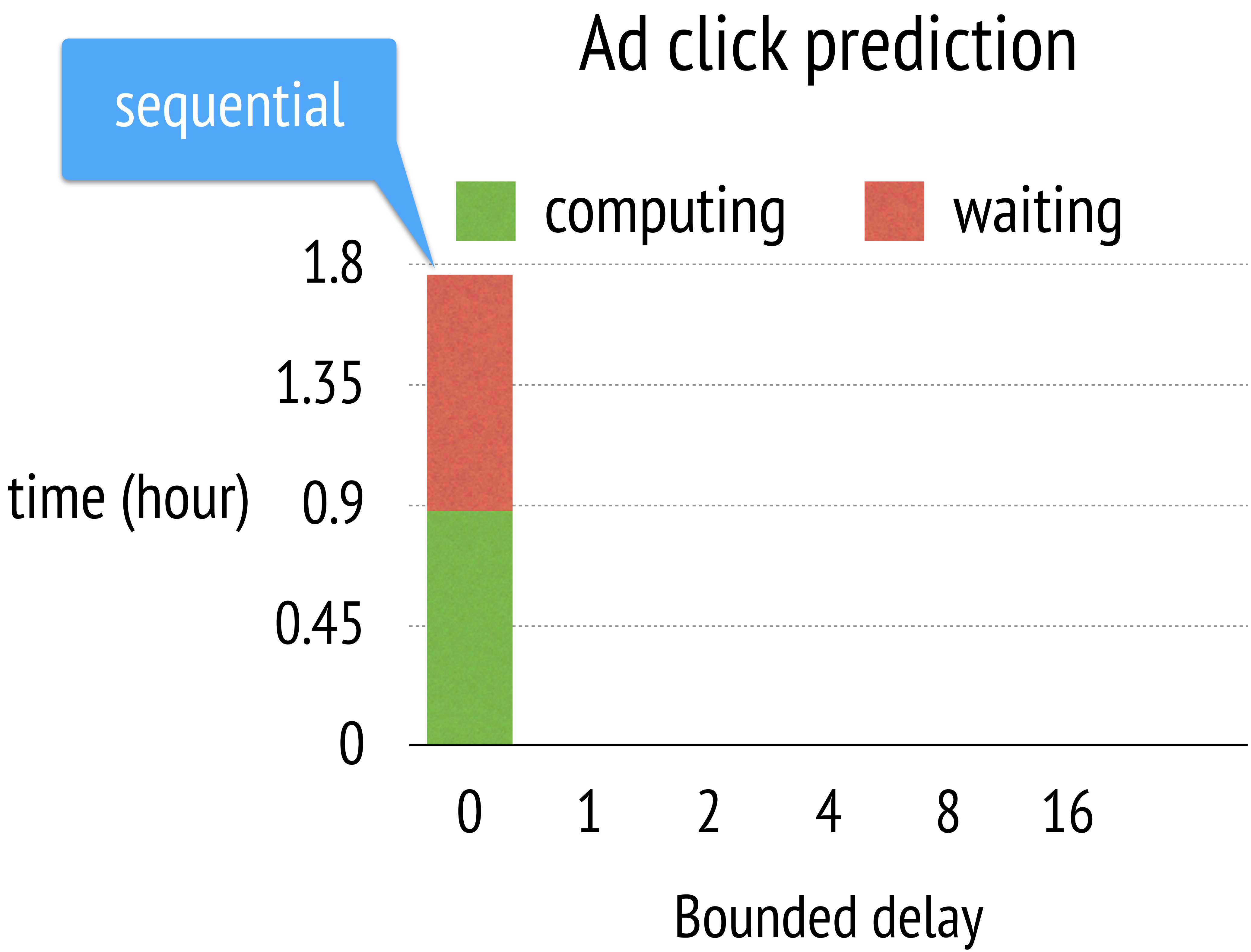


Results for bounded delay

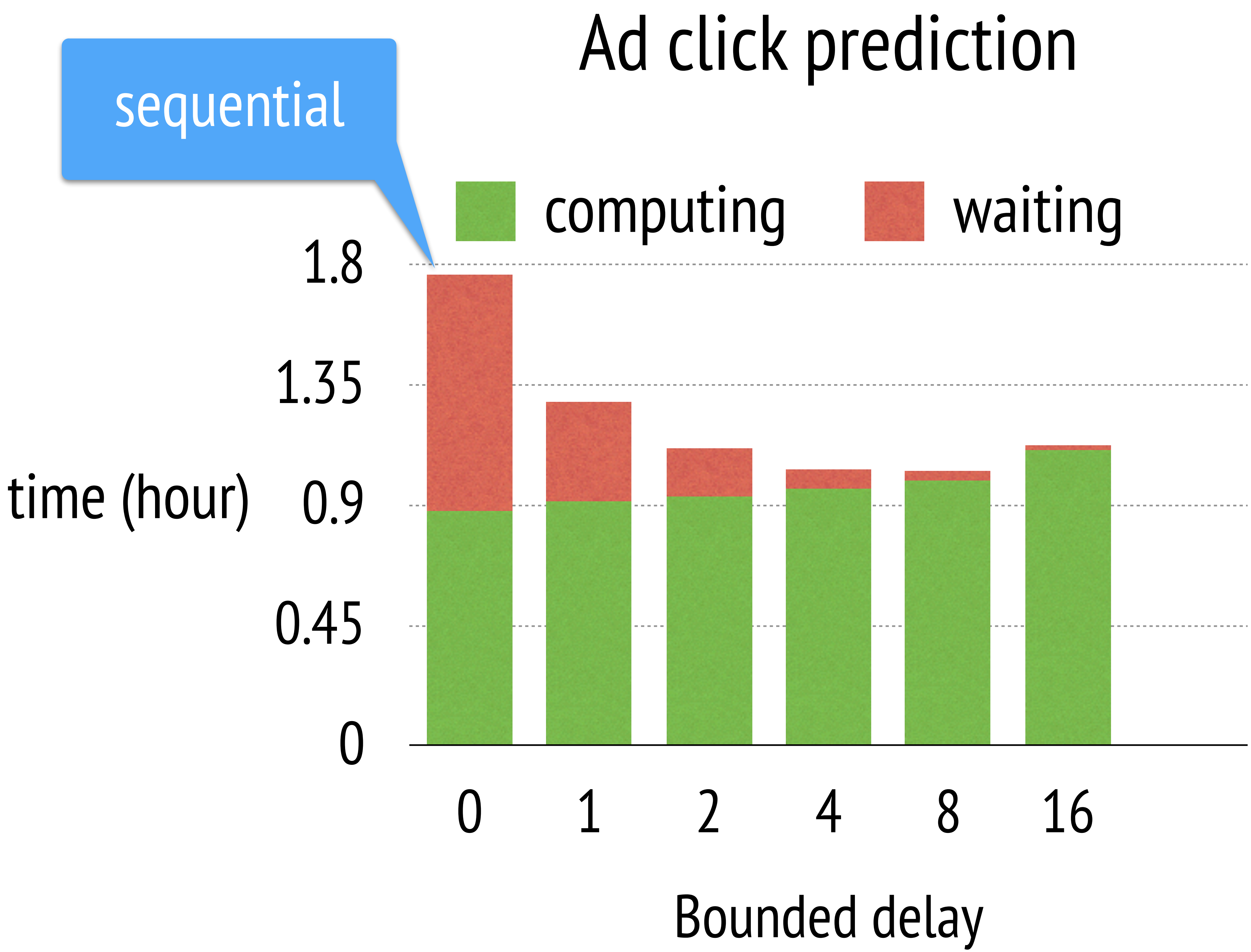
Results for bounded delay



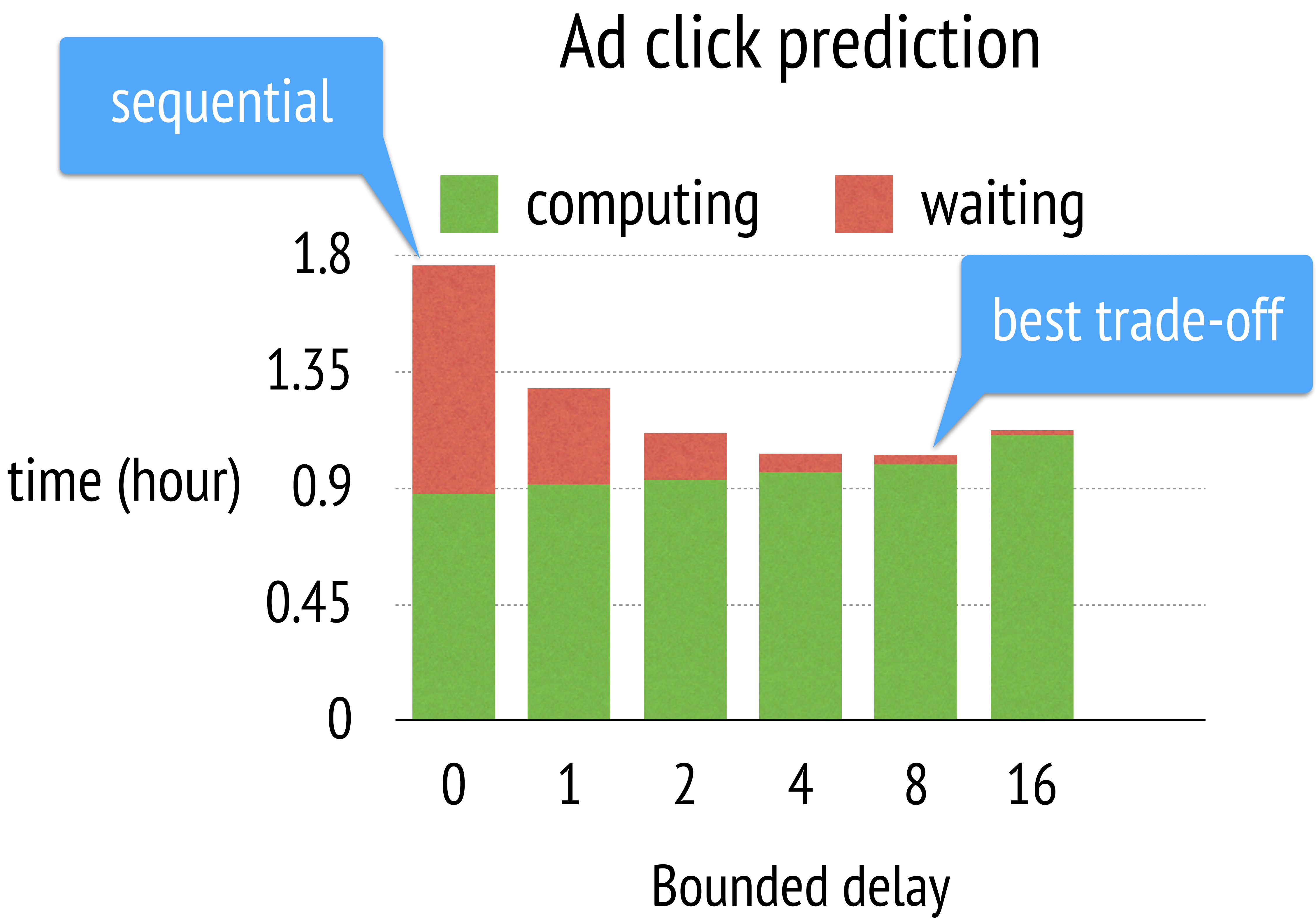
Results for bounded delay



Results for bounded delay



Results for bounded delay



User-defined filters

User-defined filters

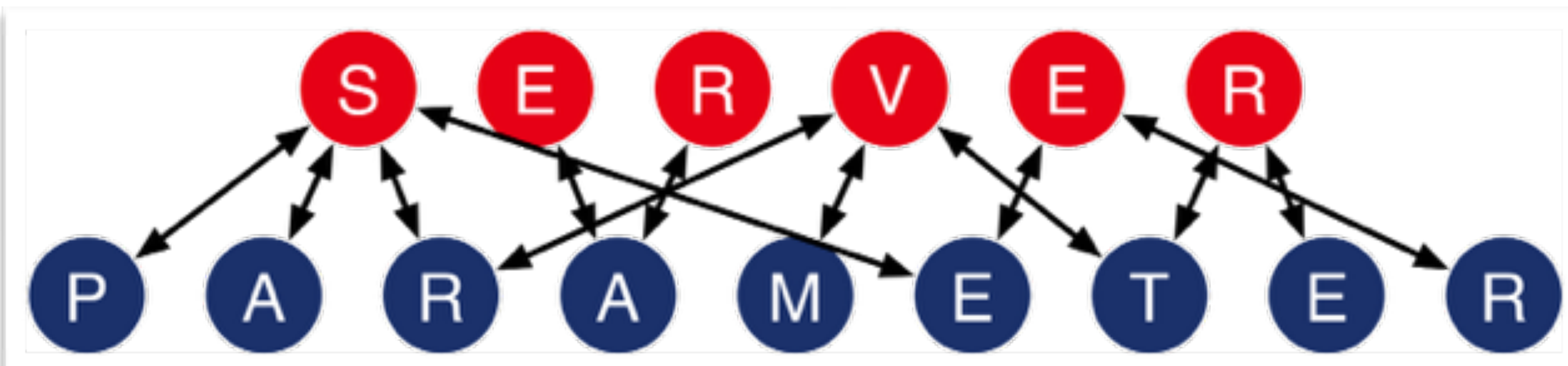
- Selectively communicate (key, value) pairs

User-defined filters

- Selectively communicate (key, value) pairs
- E.g., the KKT filter
  - send pairs if they are likely to affect the model
  - >95% keys are filtered in the ad click prediction task

Industry size machine learning problems

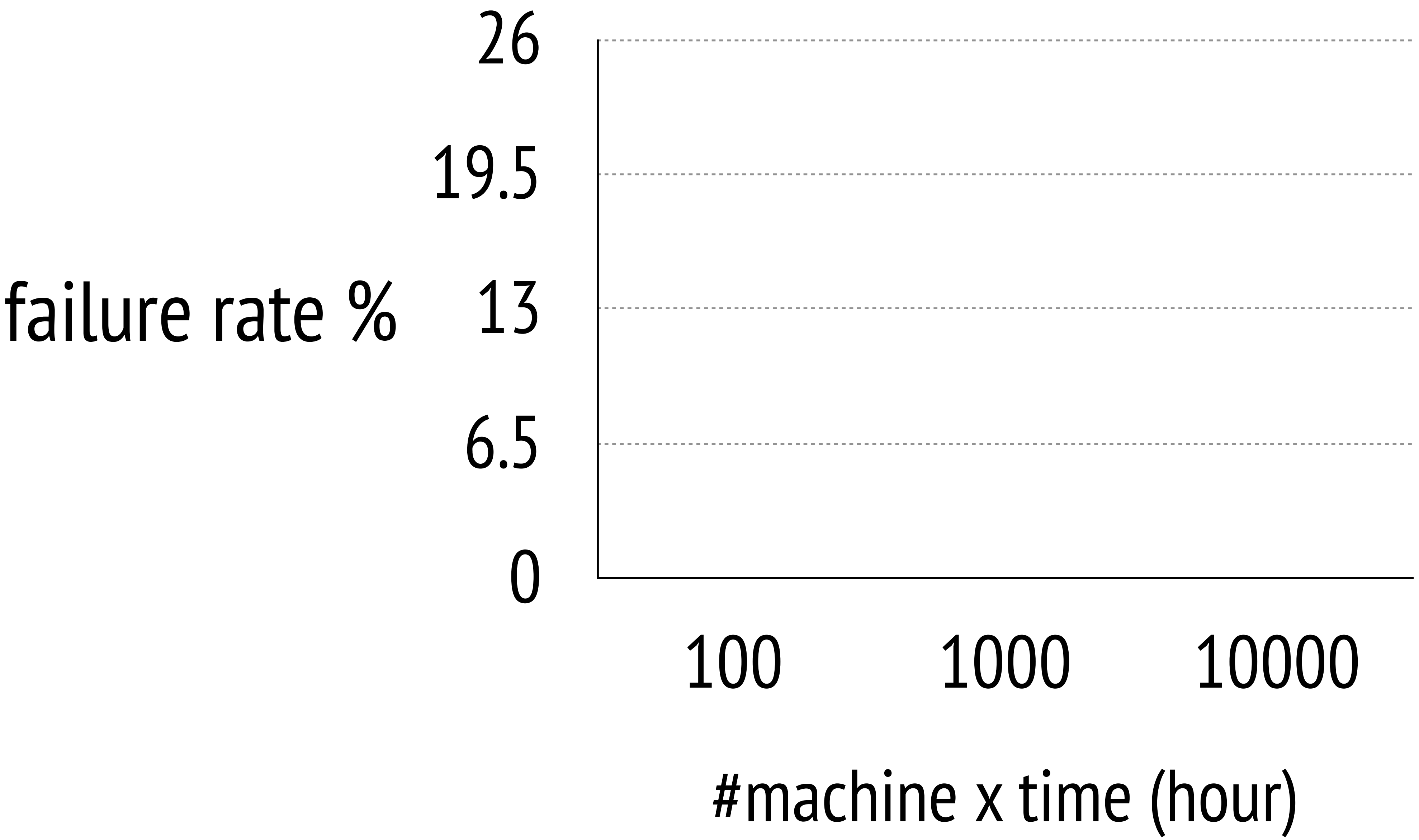
Efficient communication



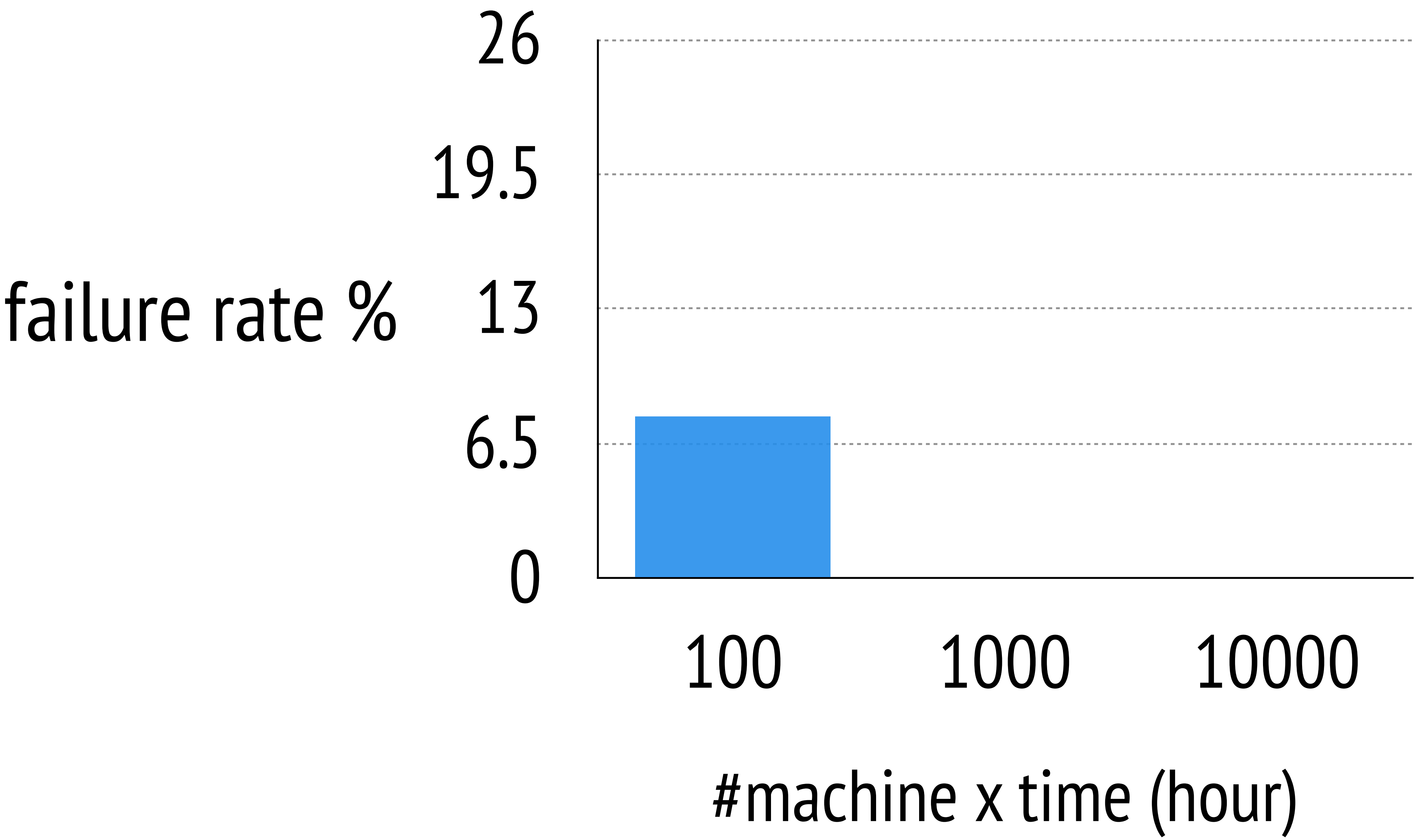
Fault tolerance

Machine learning job logs in a three-month period:

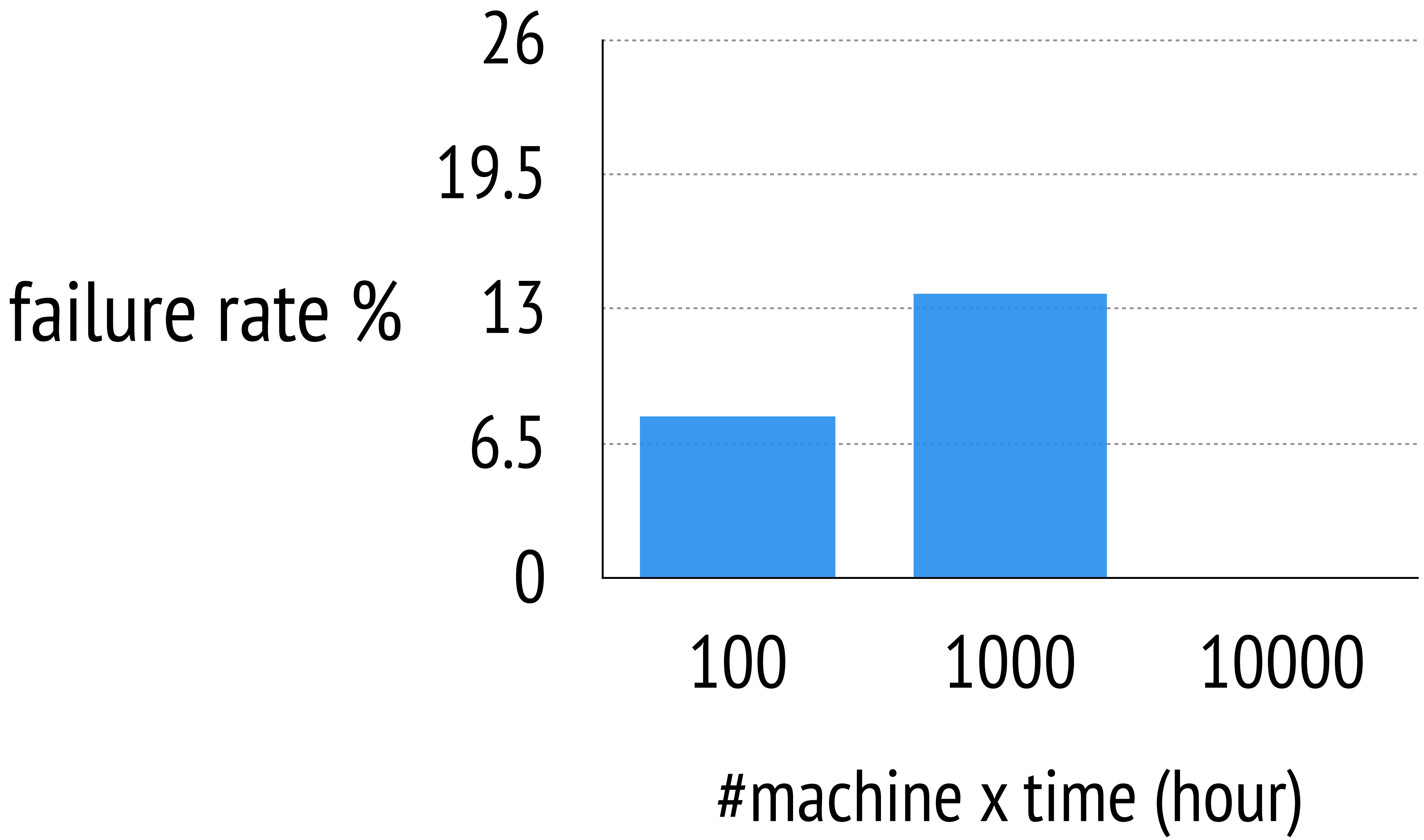
Machine learning job logs in a three-month period:



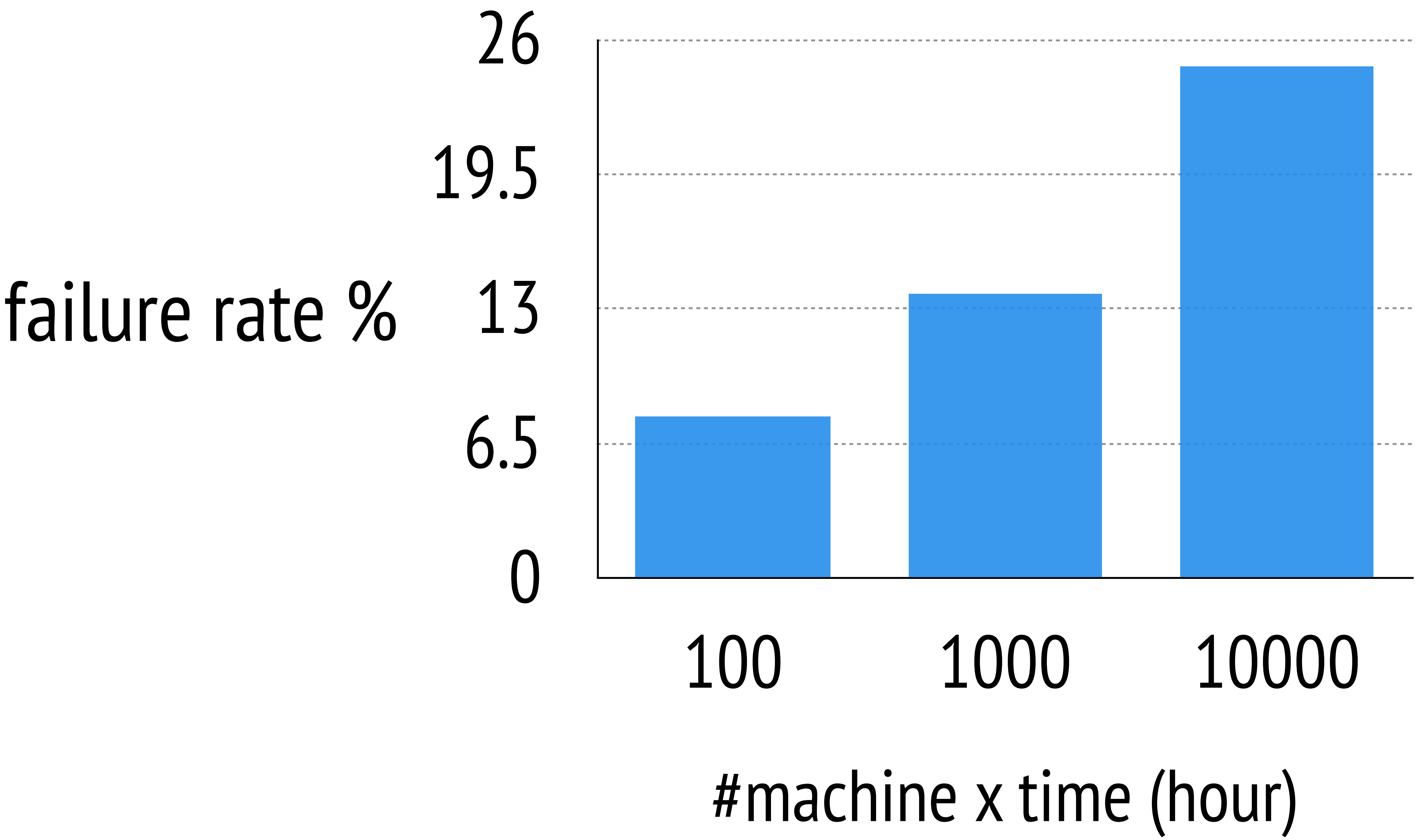
Machine learning job logs in a three-month period:



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)

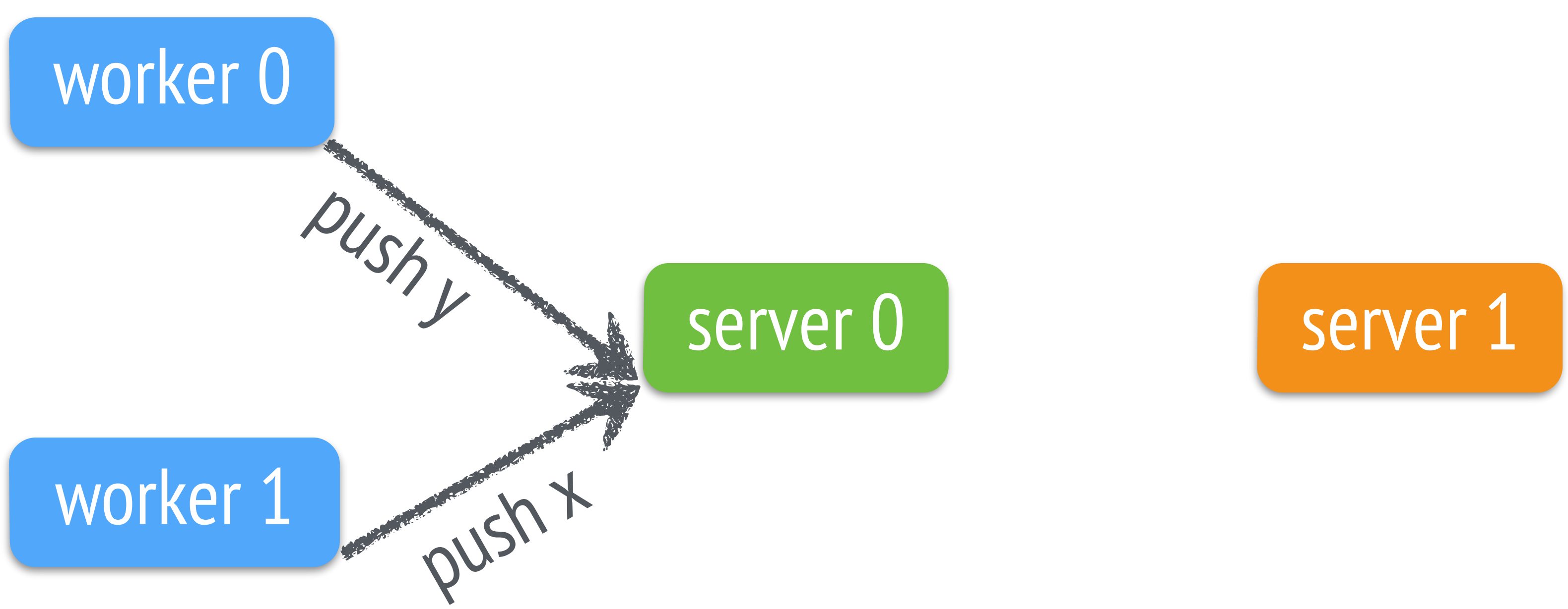
Fault tolerance

- 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)

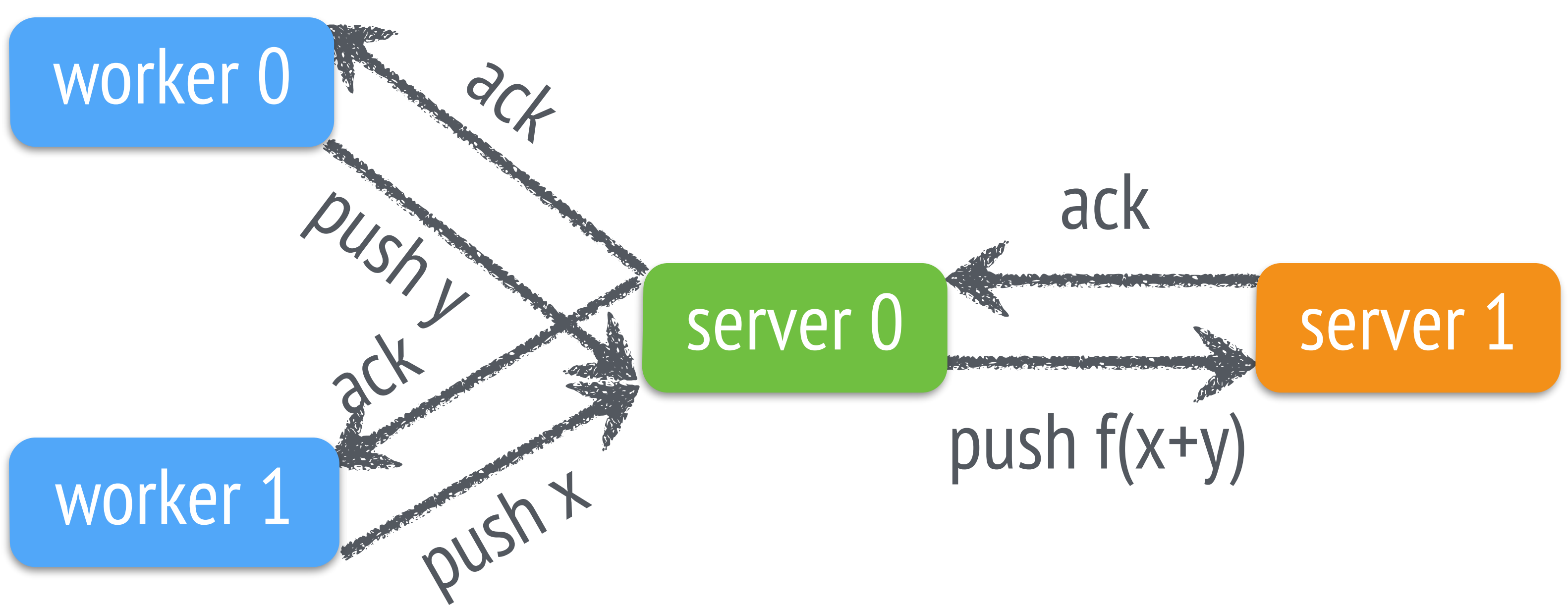
- Option: Aggregation reduces backup traffic (algo specific)



Fault tolerance

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

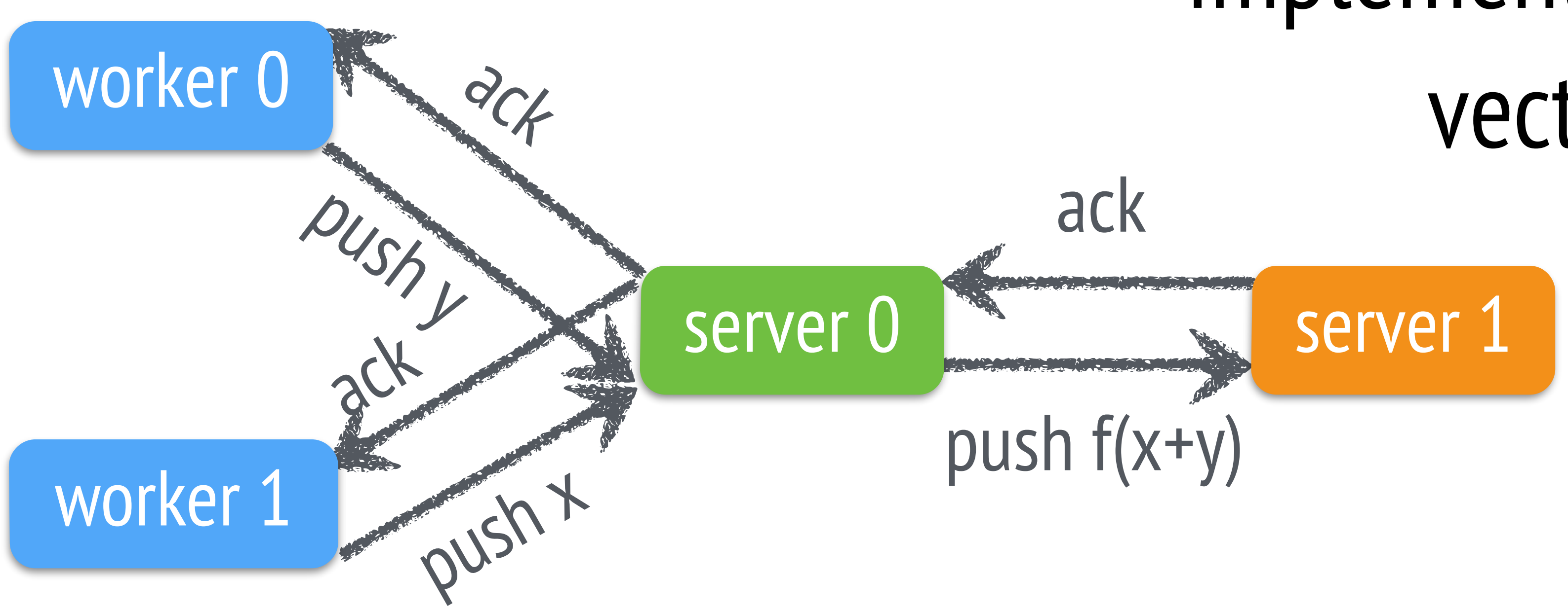
- Option: Aggregation reduces backup traffic (algo specific)



Fault tolerance

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

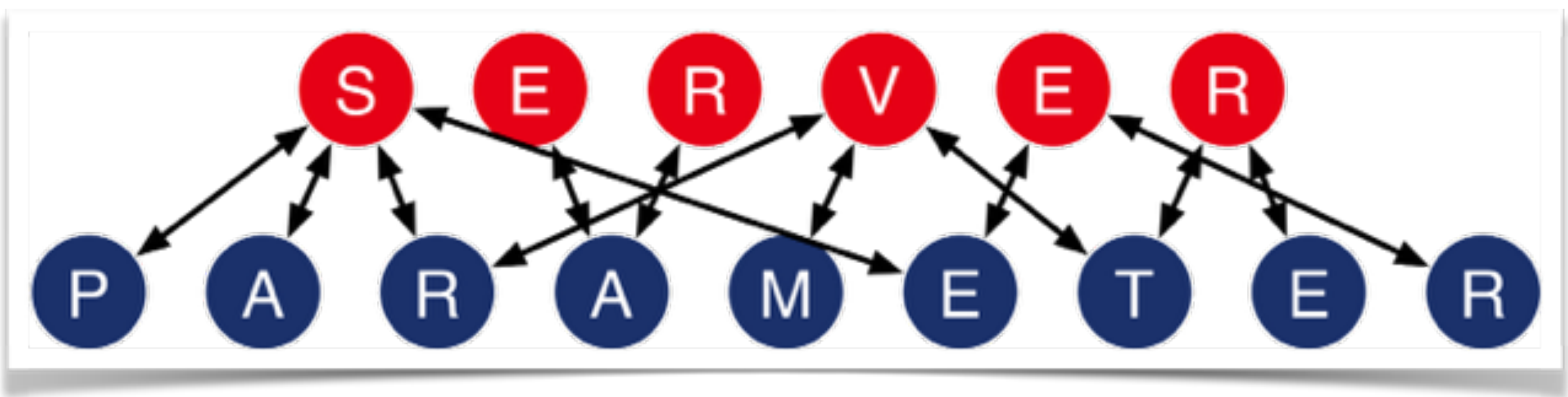
- Option: Aggregation reduces backup traffic (algo specific)



implemented by efficient vector clock

Industry size machine learning problems

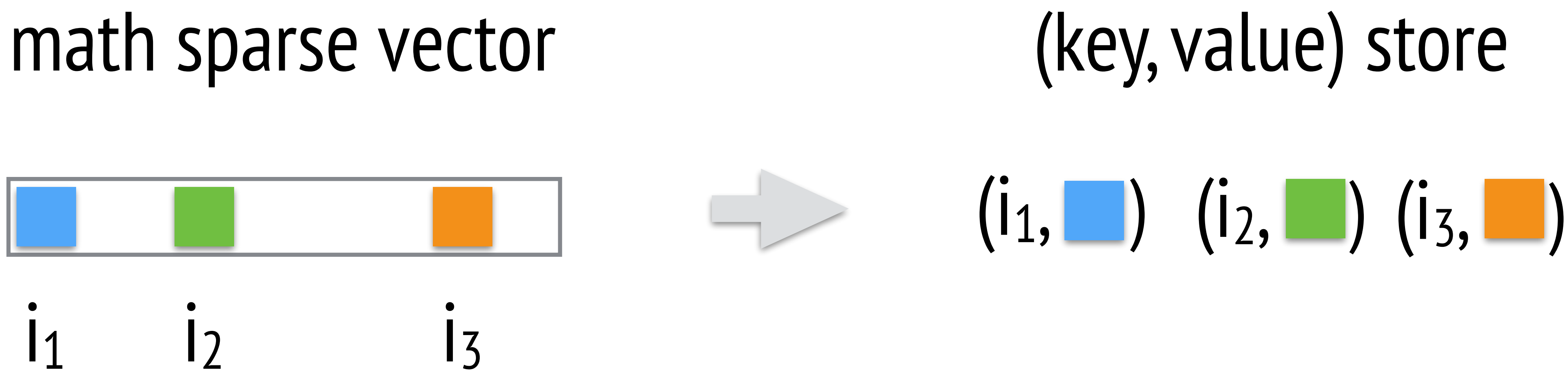
Efficient communication



Fault tolerance

Easy to use

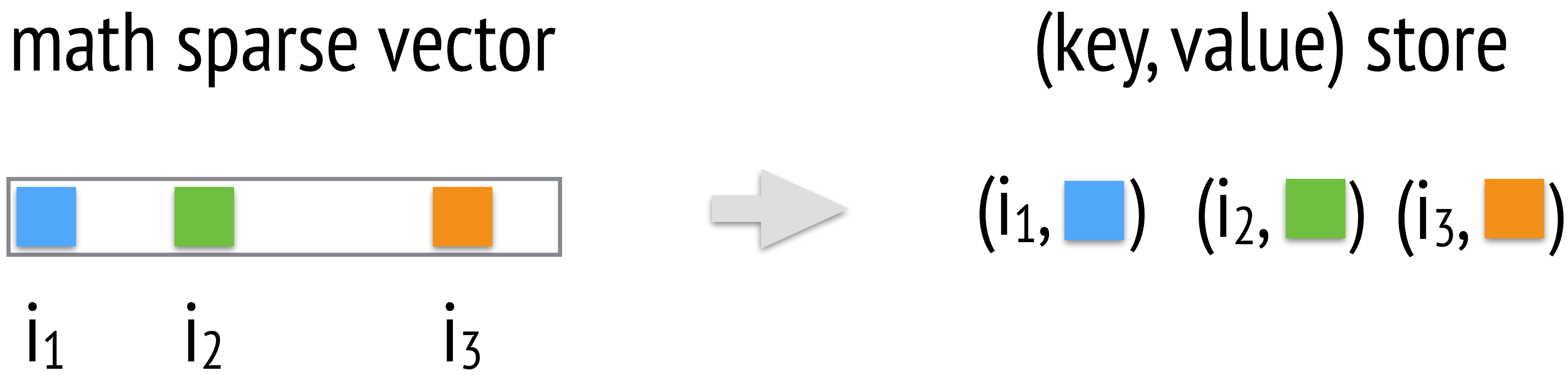
(Key, value) vectors for the shared parameters



(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



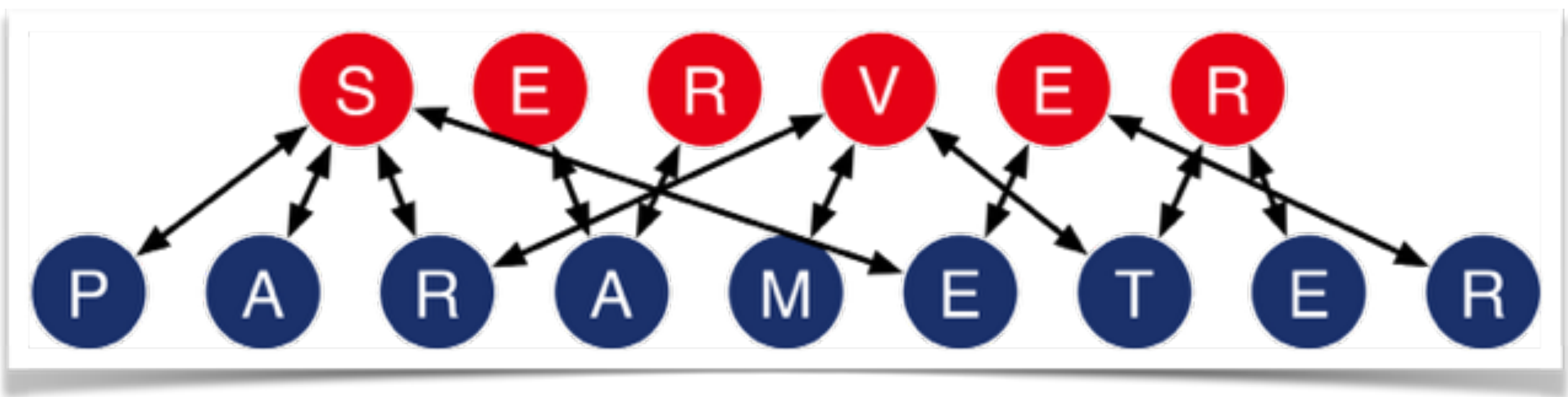
- Good for programmers: Matches mental model
- Good for system: Expose optimizations based upon structure of data

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 ads will be clicked or not

Sparse Logistic Regression

- Predict ads will be clicked or not
- Baseline: two systems in production

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

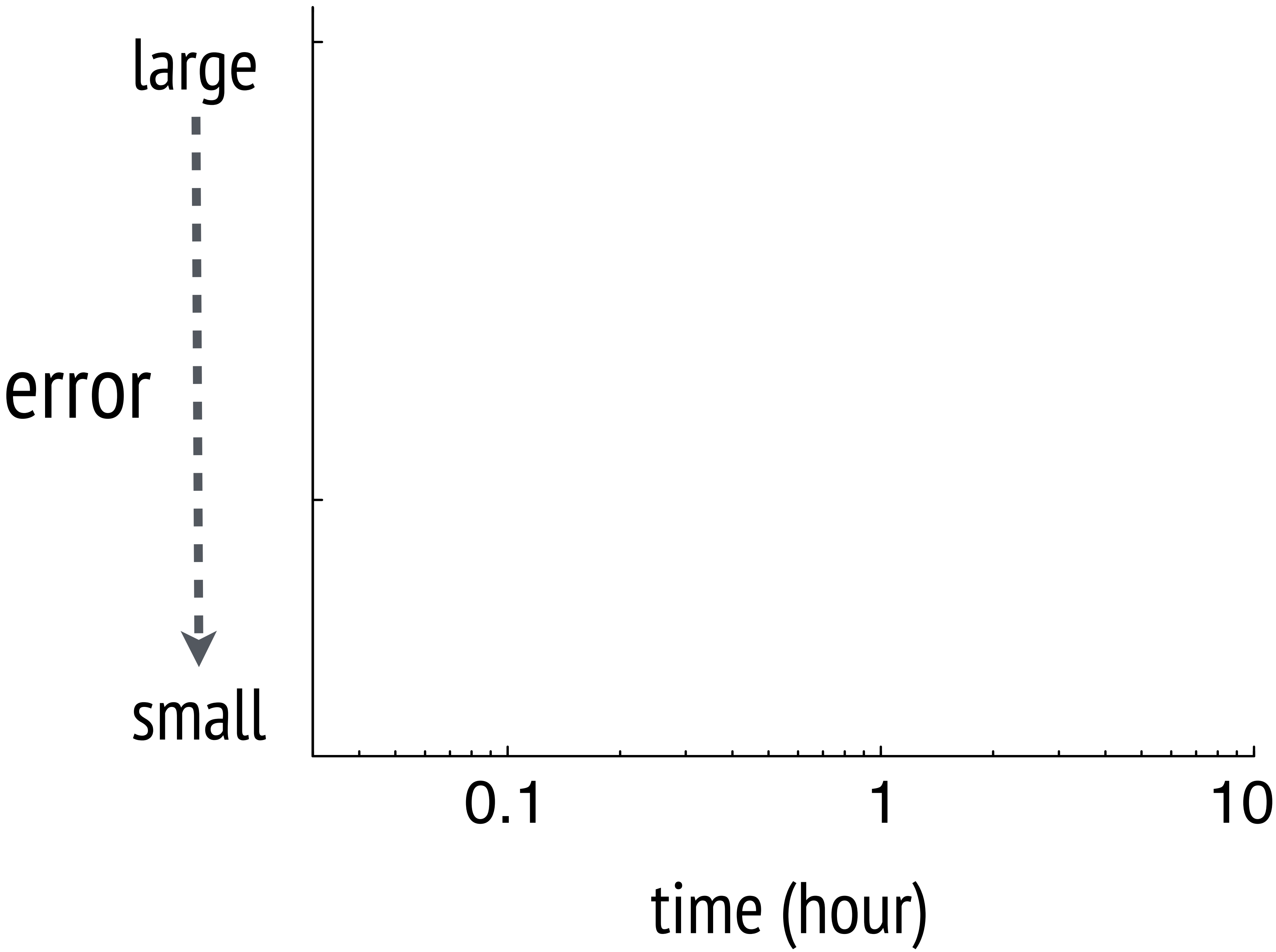
Sparse Logistic Regression

- Predict ads will be clicked or not
- Baseline: two systems in production

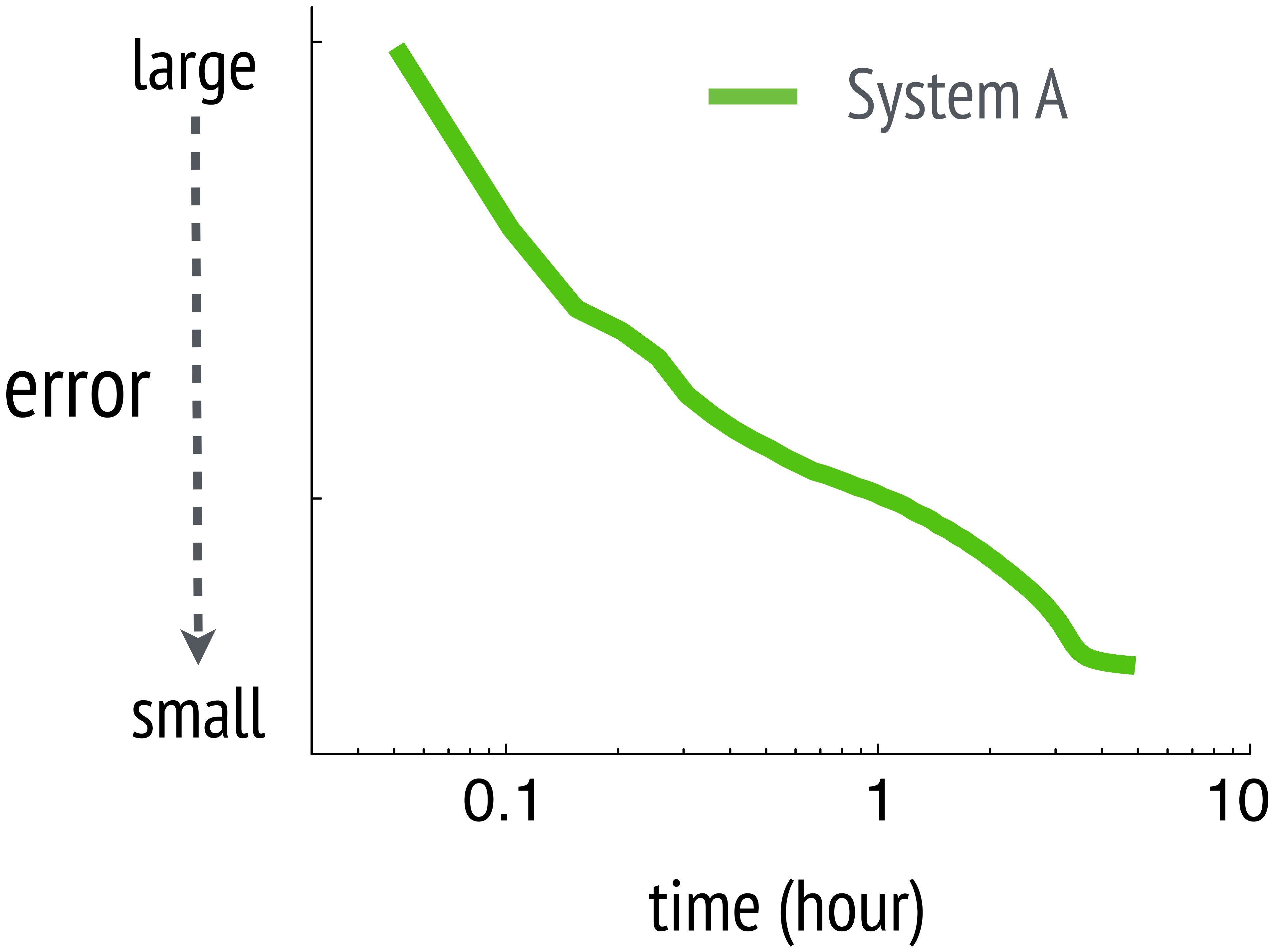
	Method	Consistency	LOC
System-A	L-BFGS	Sequential	10K
System-B	Block PG	Sequential	30K
Parameter Server	Block PG	Bounded Delay + KKT	300

- 636T real ads data
  - 170 billions of examples, 65 billions of features
- 1,000 machines with 16,000 cores

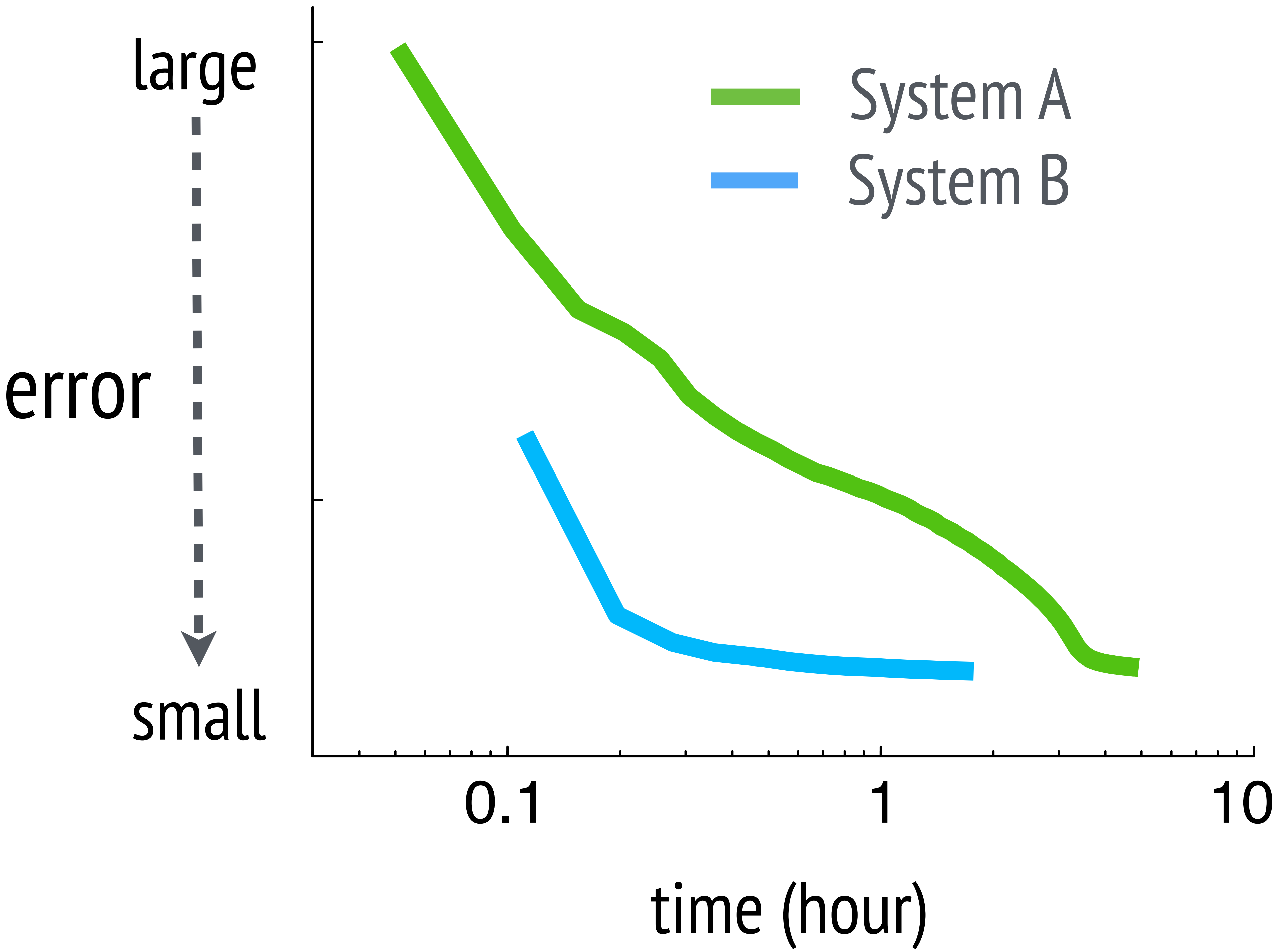
Progress



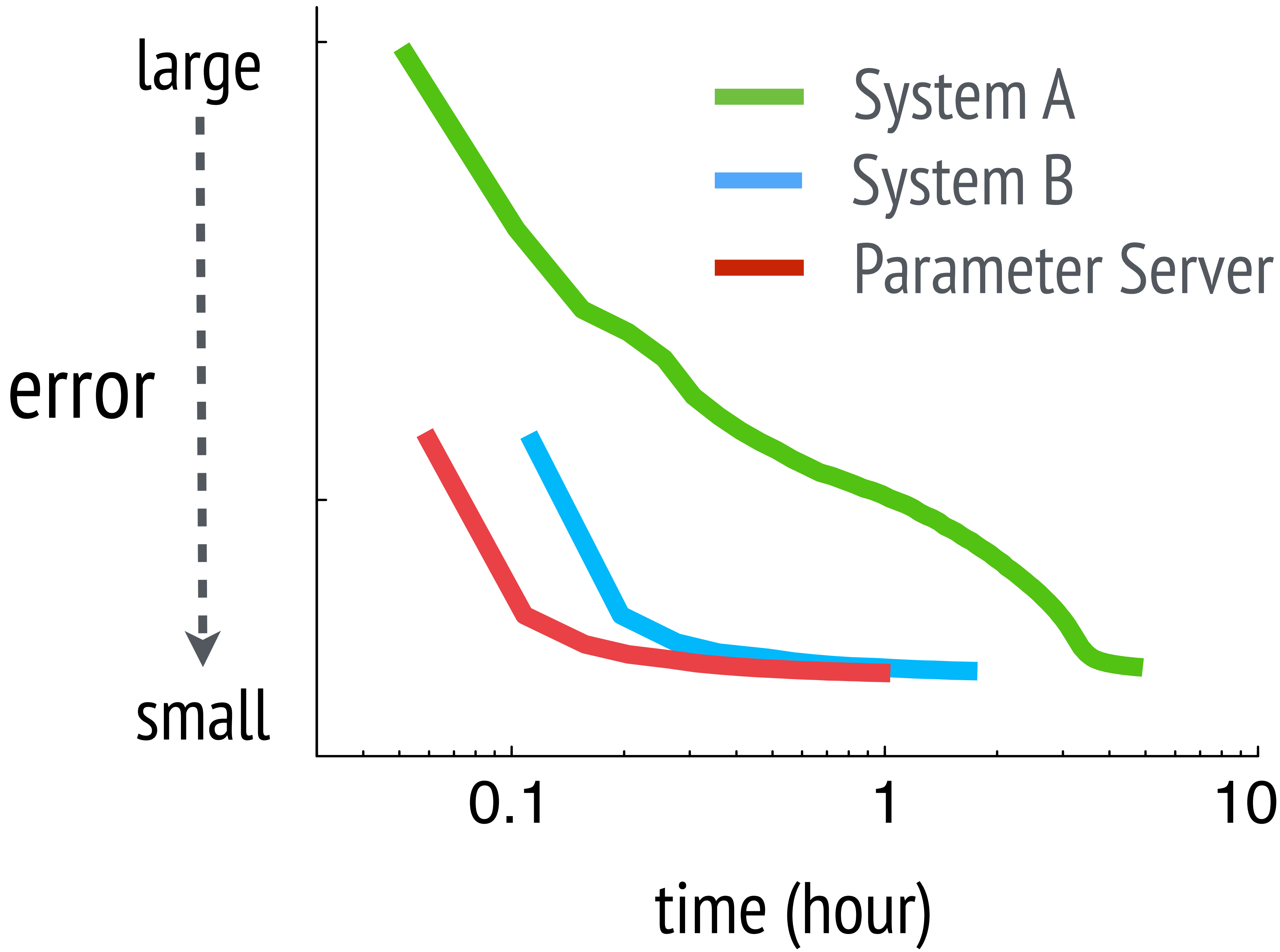
Progress



Progress



Progress

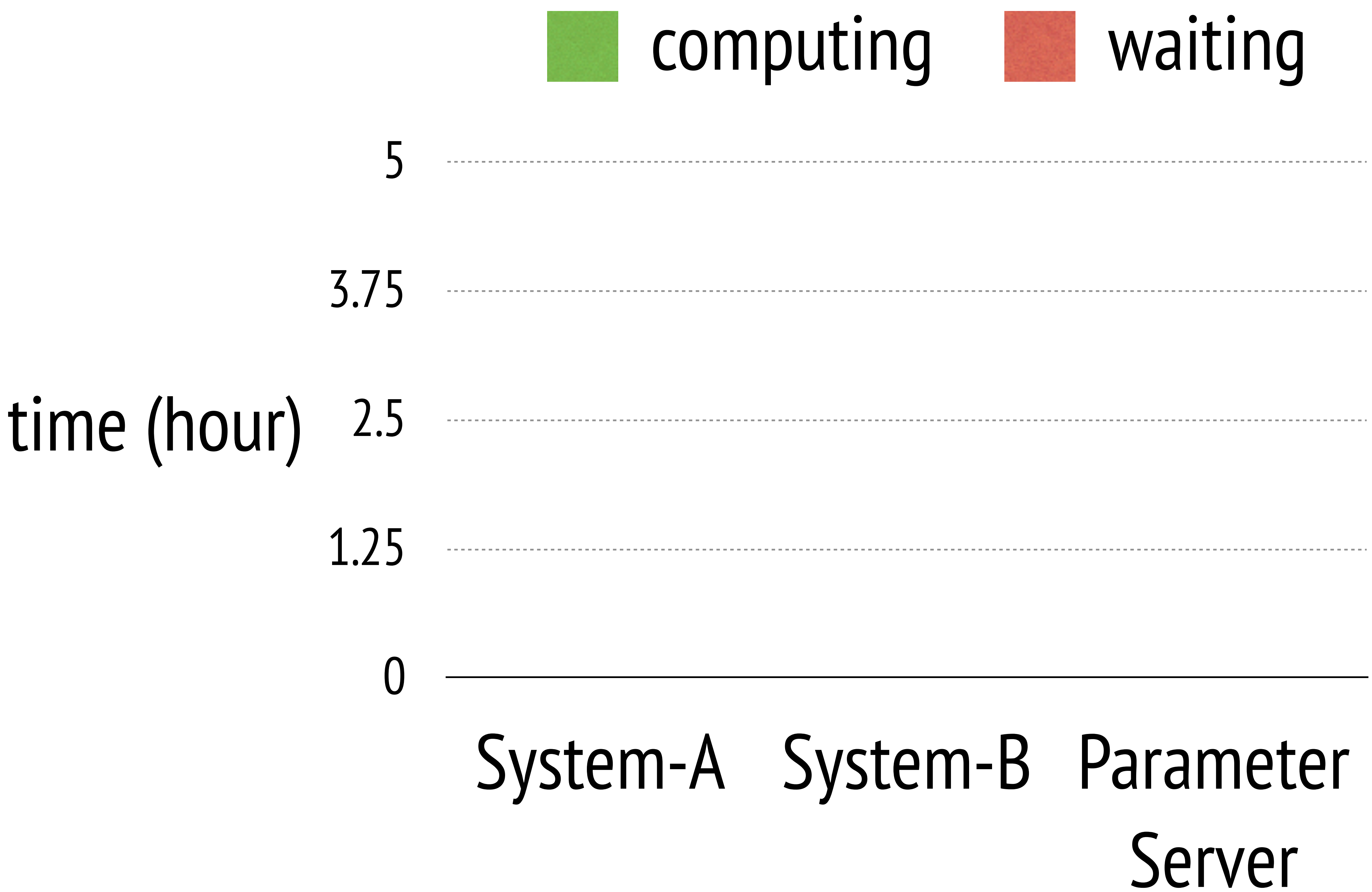


large
error
small
System A
System B
Parameter Server
0.1
1
10
time (hour)

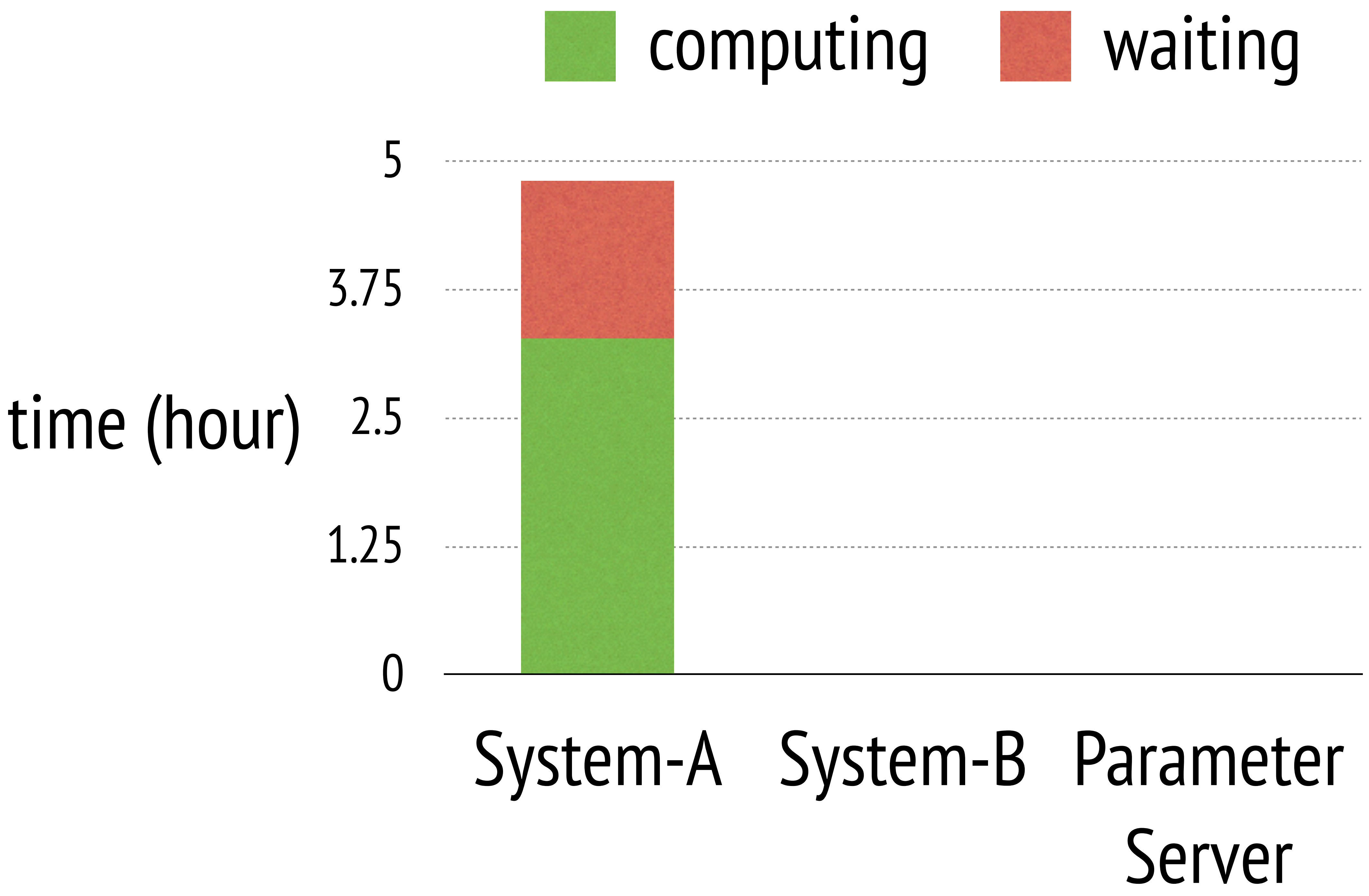
Time decomposition

time (hour)

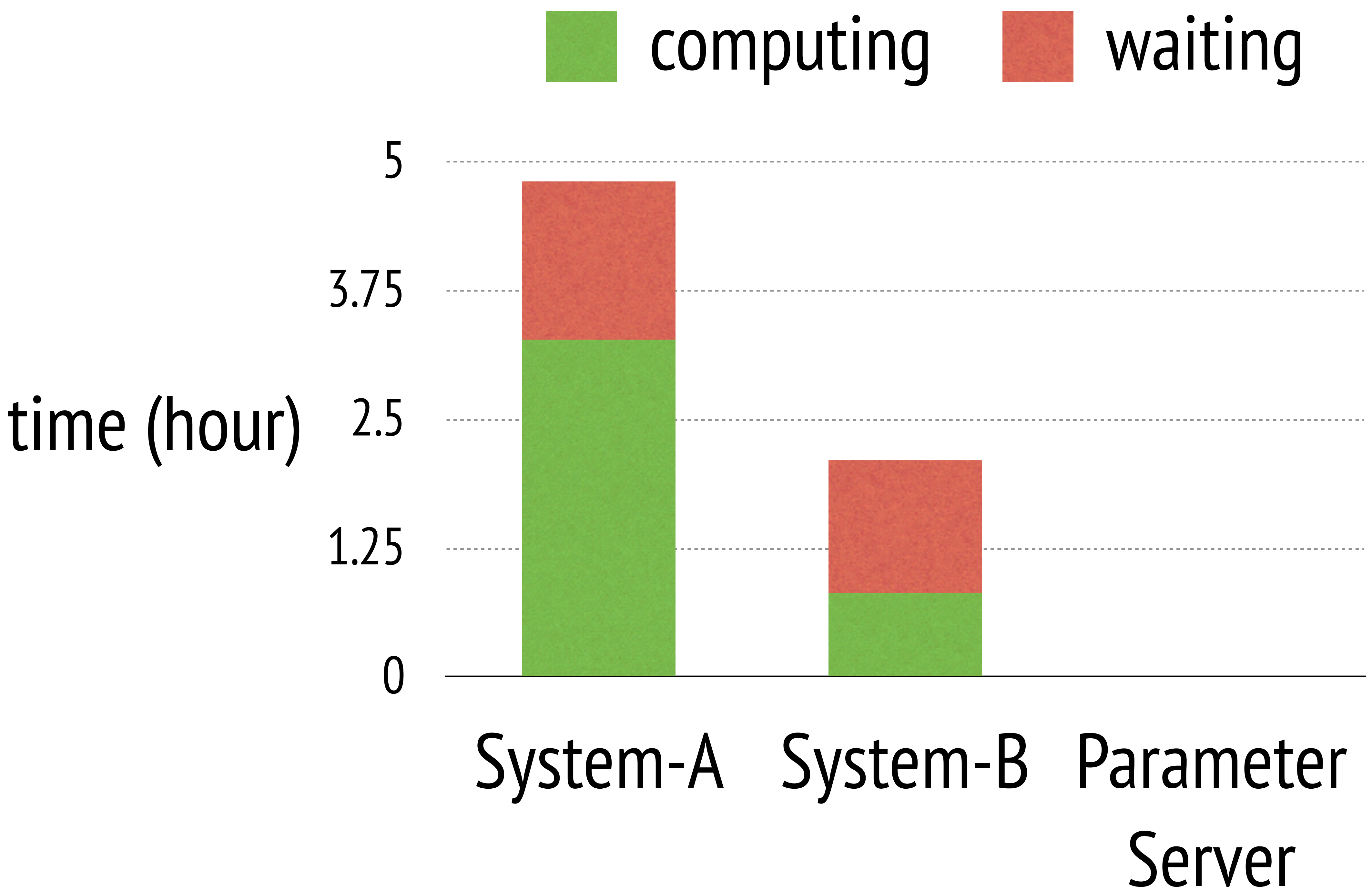
Time decomposition



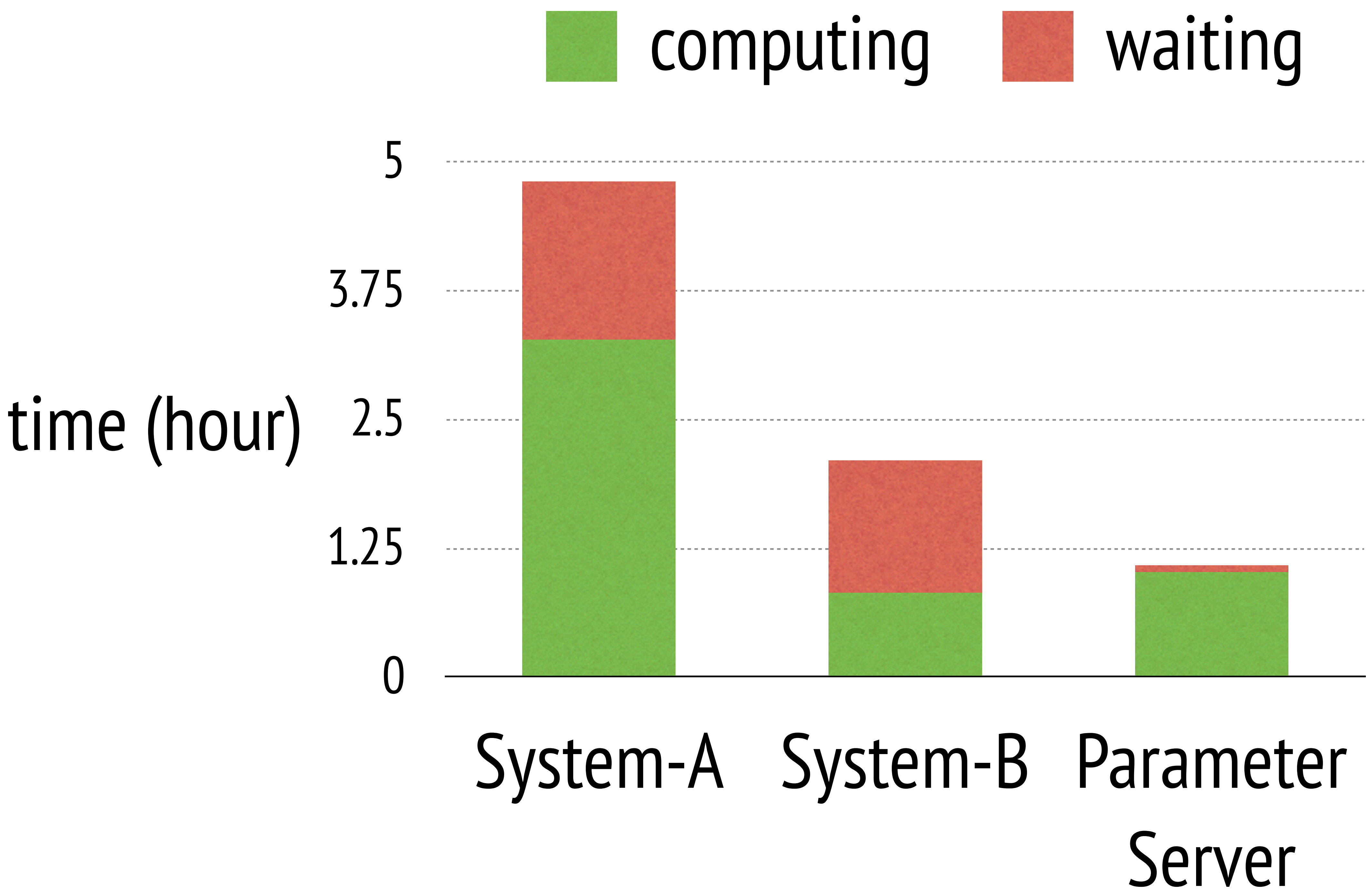
Time decomposition



Time decomposition



Time decomposition

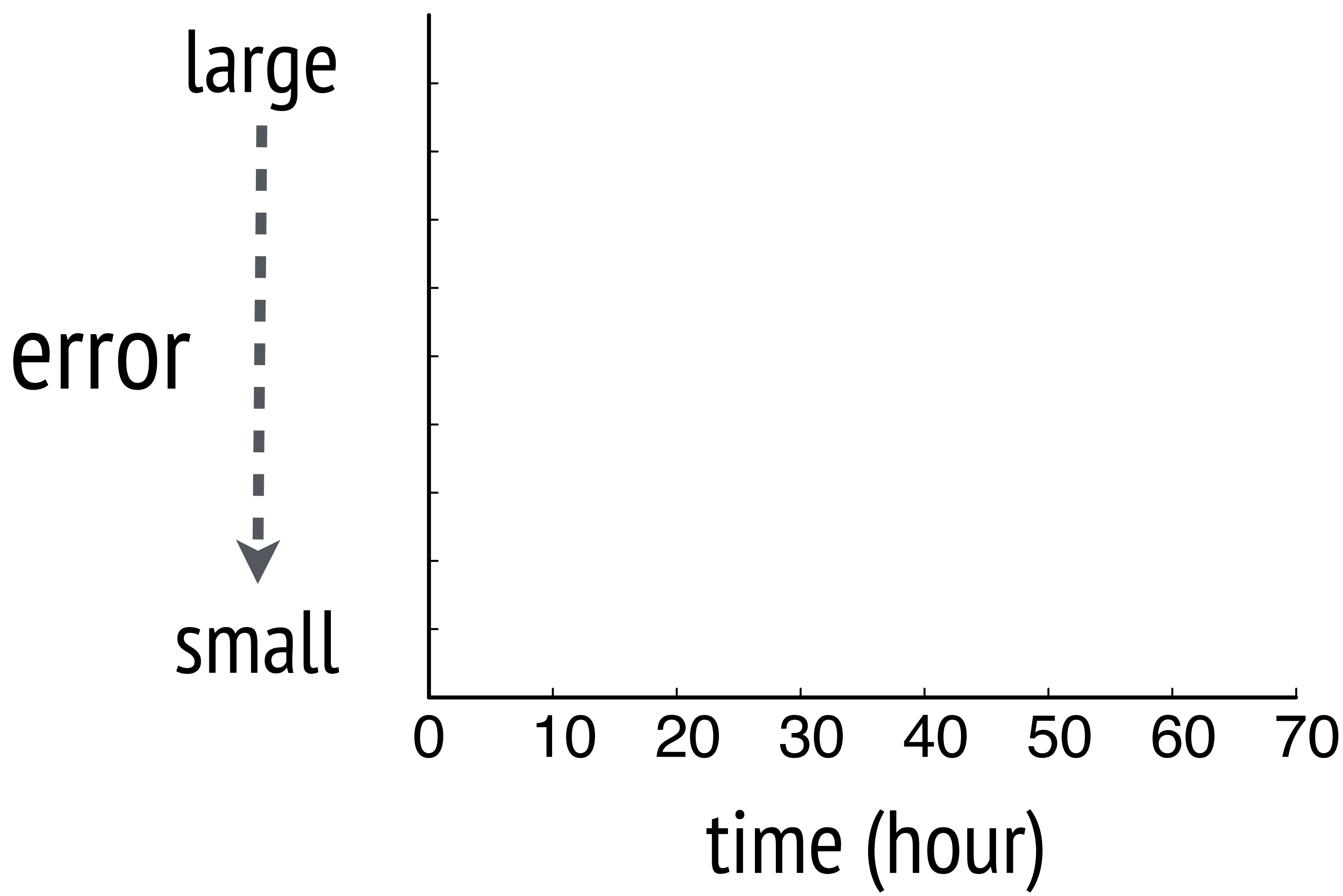


Topic Modeling ("LDA")

- Gradient descent with eventual consistency
- 5B users' click logs, Group users into 1,000 groups based on URLs they clicked

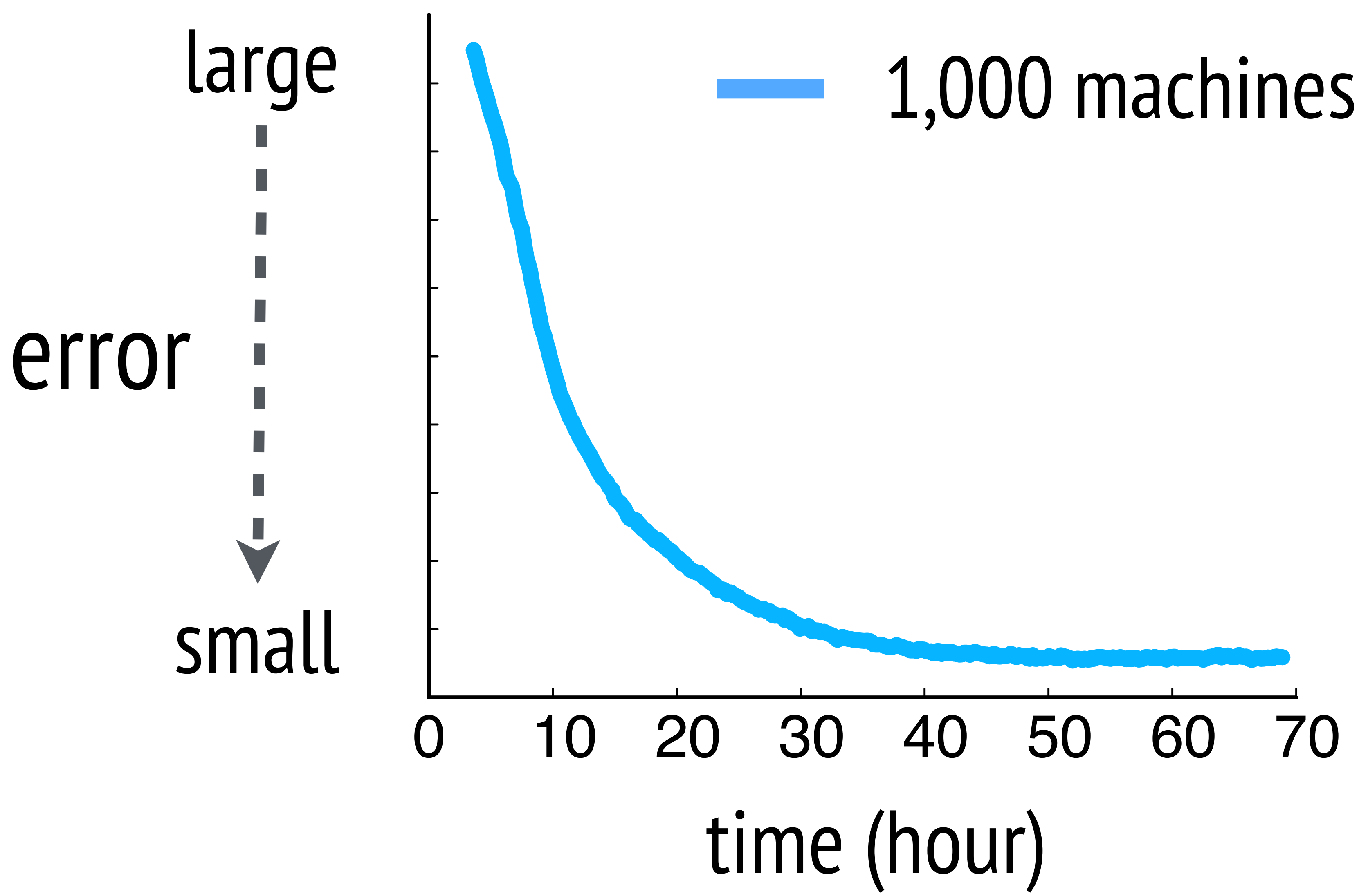
Topic Modeling (“LDA”)

- Gradient descent with eventual consistency
- 5B users’ click logs, Group users into 1,000 groups based on URLs they clicked



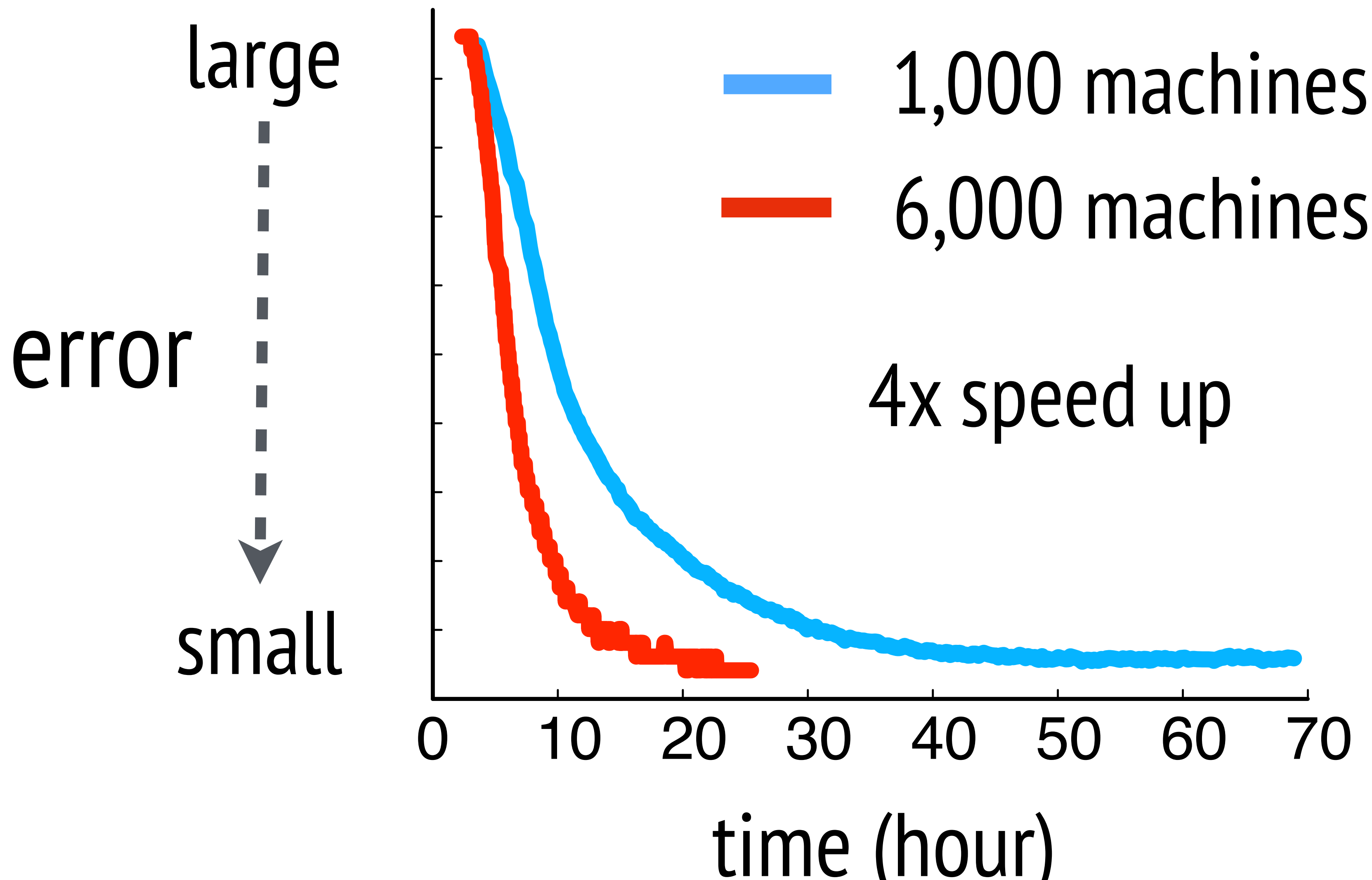
Topic Modeling (“LDA”)

- Gradient descent with eventual consistency
- 5B users’ click logs, Group users into 1,000 groups based on URLs they clicked



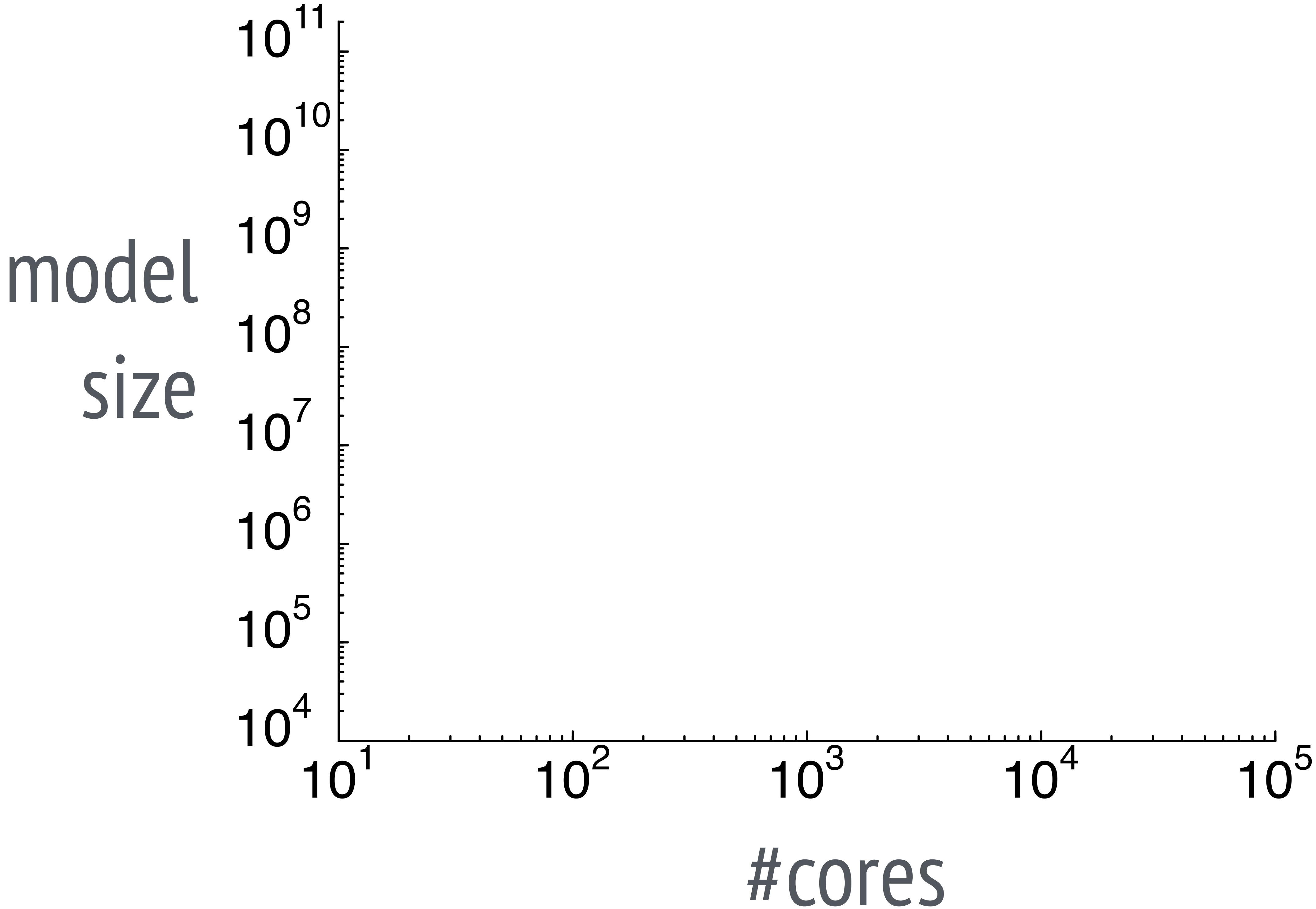
Topic Modeling ("LDA")

- Gradient descent with eventual consistency
- 5B users' click logs, Group users into 1,000 groups based on URLs they clicked



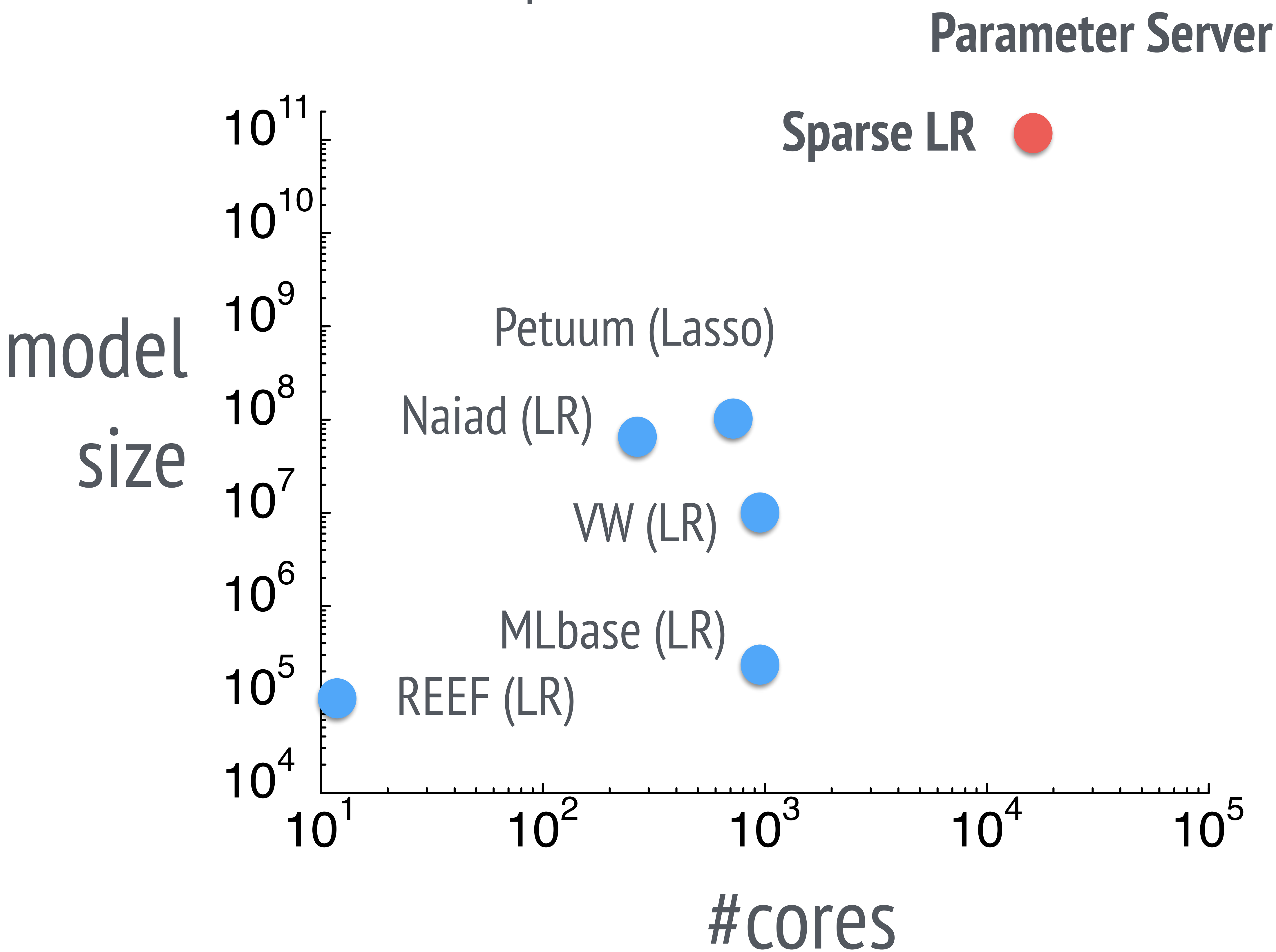
Largest experiments of related systems

Data were collected on April'14



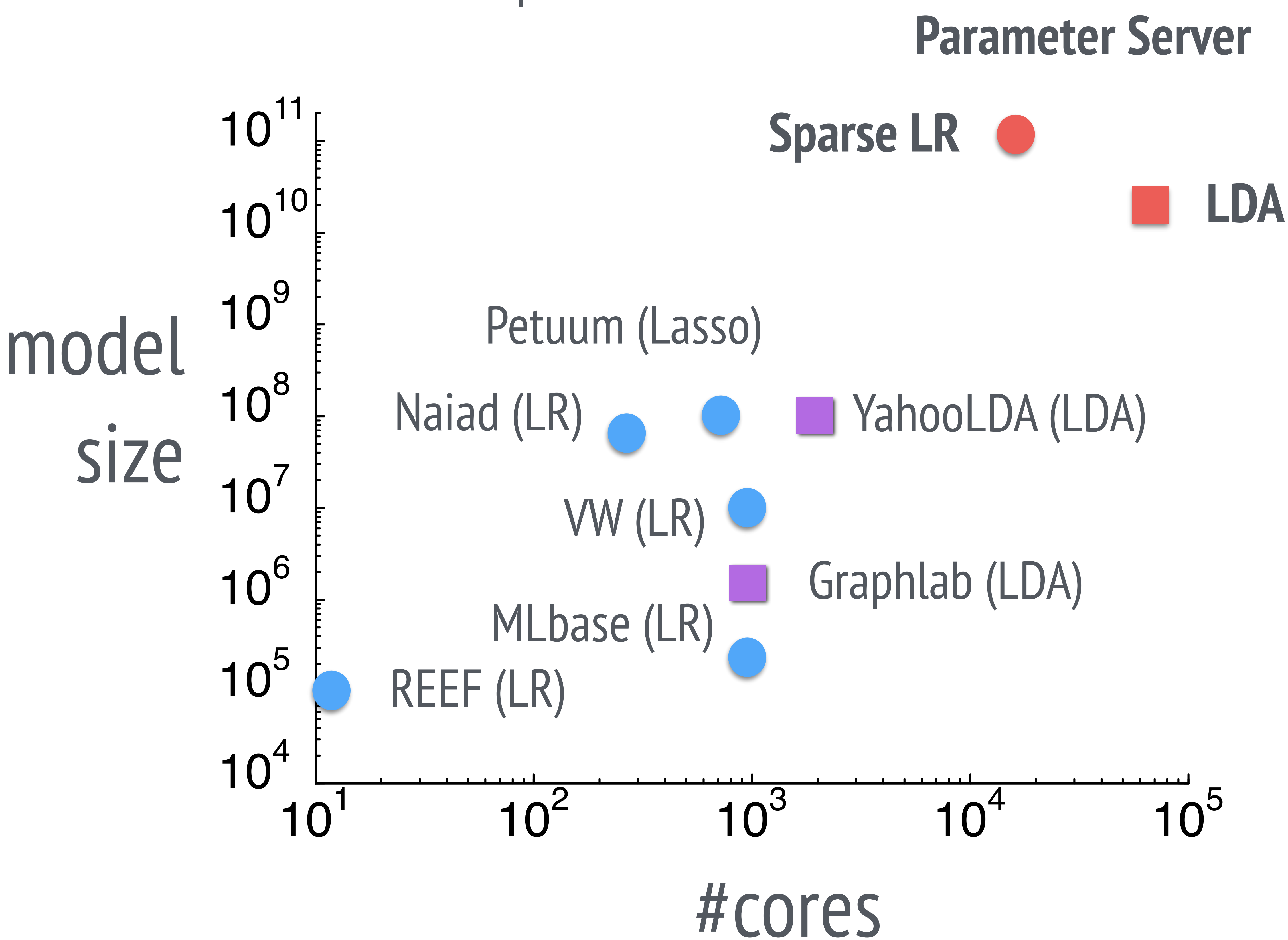
Largest experiments of related systems

Data were collected on April'14



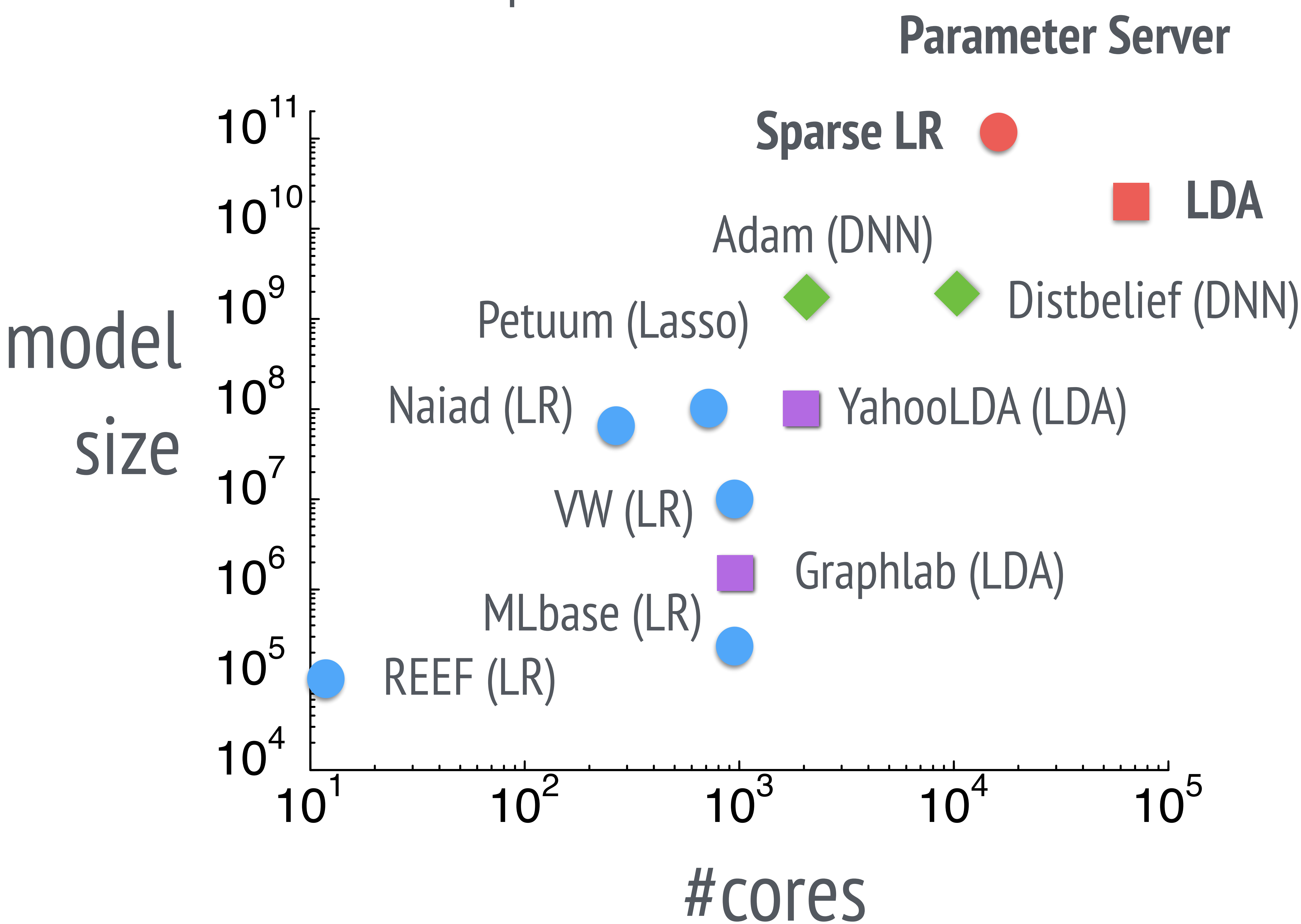
Largest experiments of related systems

Data were collected on April'14



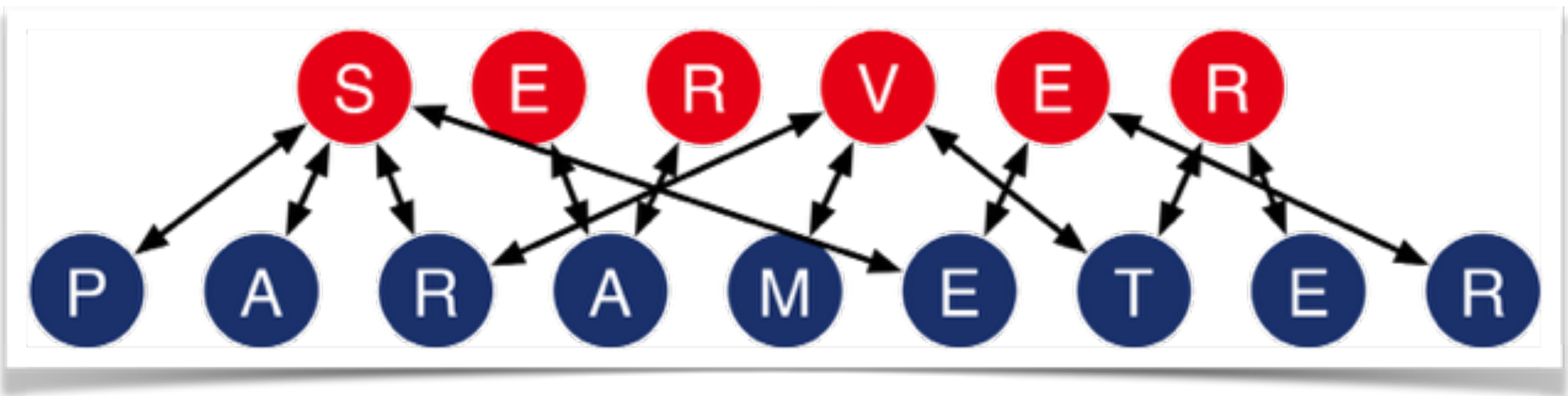
Largest experiments of related systems

Data were collected on April'14



Industry size machine learning problems

Efficient communication



Fault tolerance

Easy to use

Evaluation

Industry size machine learning problems

Efficient communication

Code available at
parameterserver.org

Fault tolerance

Easy to use

Evaluation

Q&A