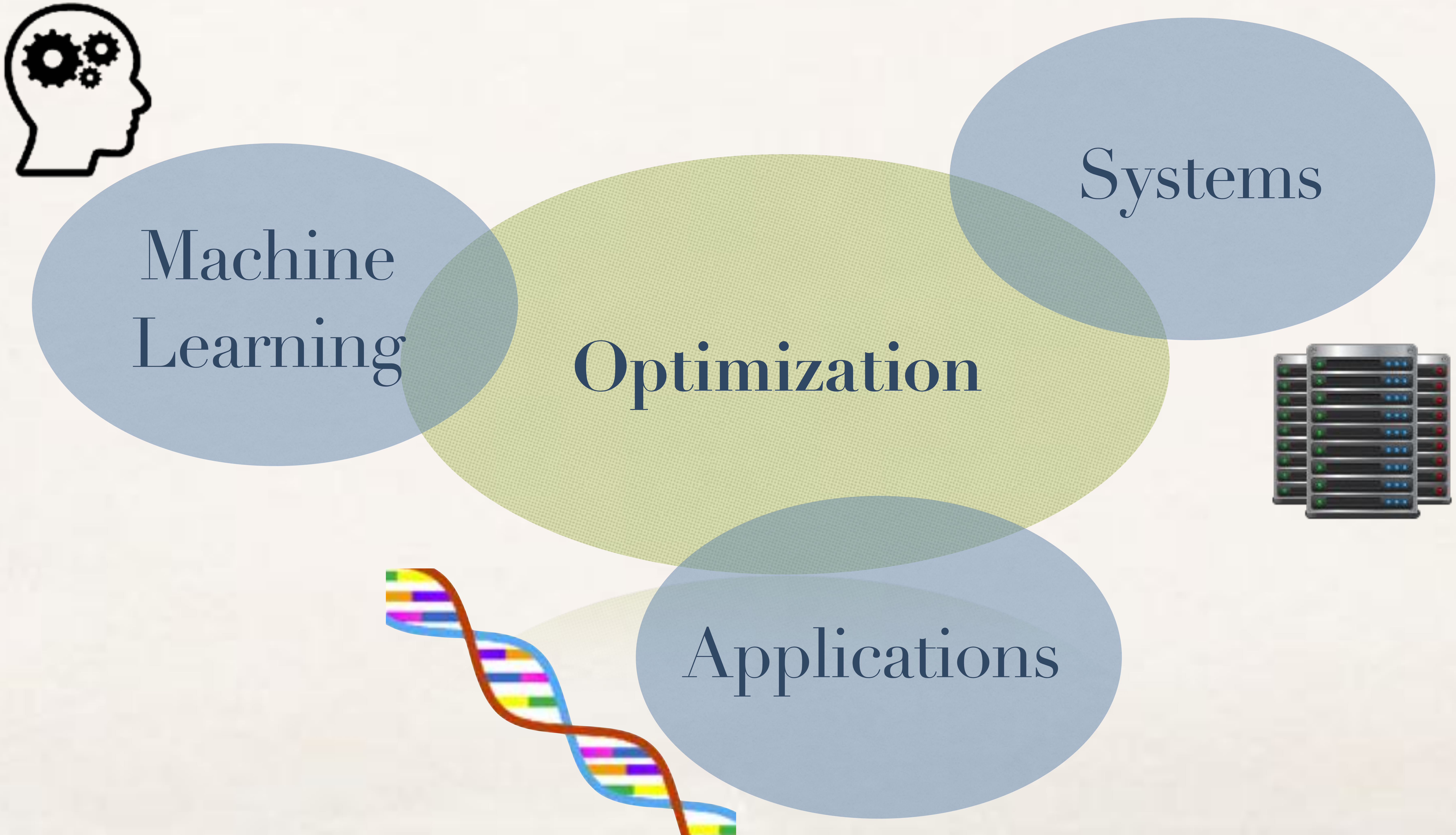
Distributed Machine Learning

Current Bottlenecks in Algorithms and Software Frameworks on HPC and Cloud Architectures

Martin Jaggi
EPFL Machine Learning and Optimization Laboratory
mlo.epfl.ch

Machine Learning Methods to Analyze Large-Scale Data



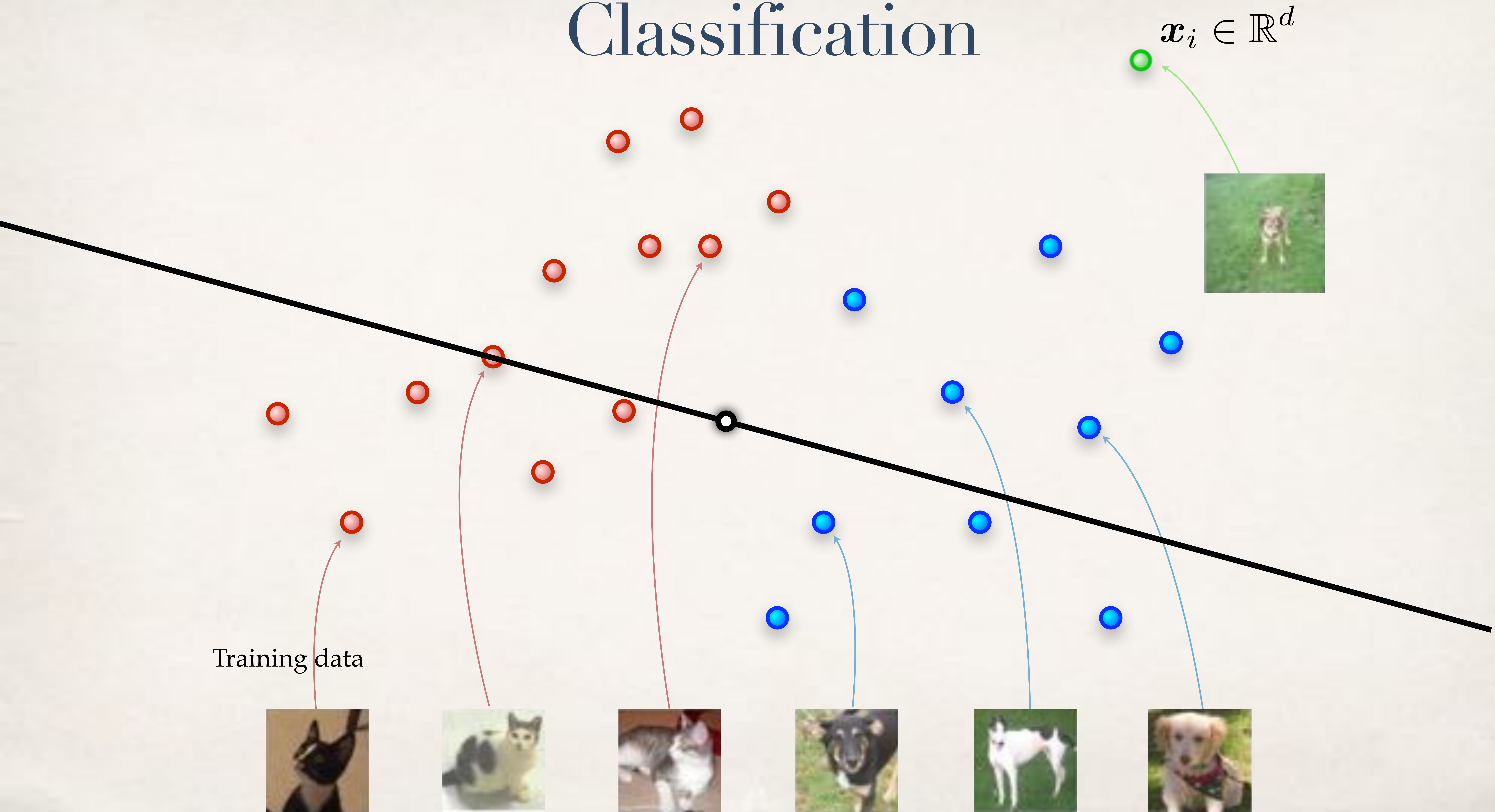
What is Machine Learning?

software that can

learn from data

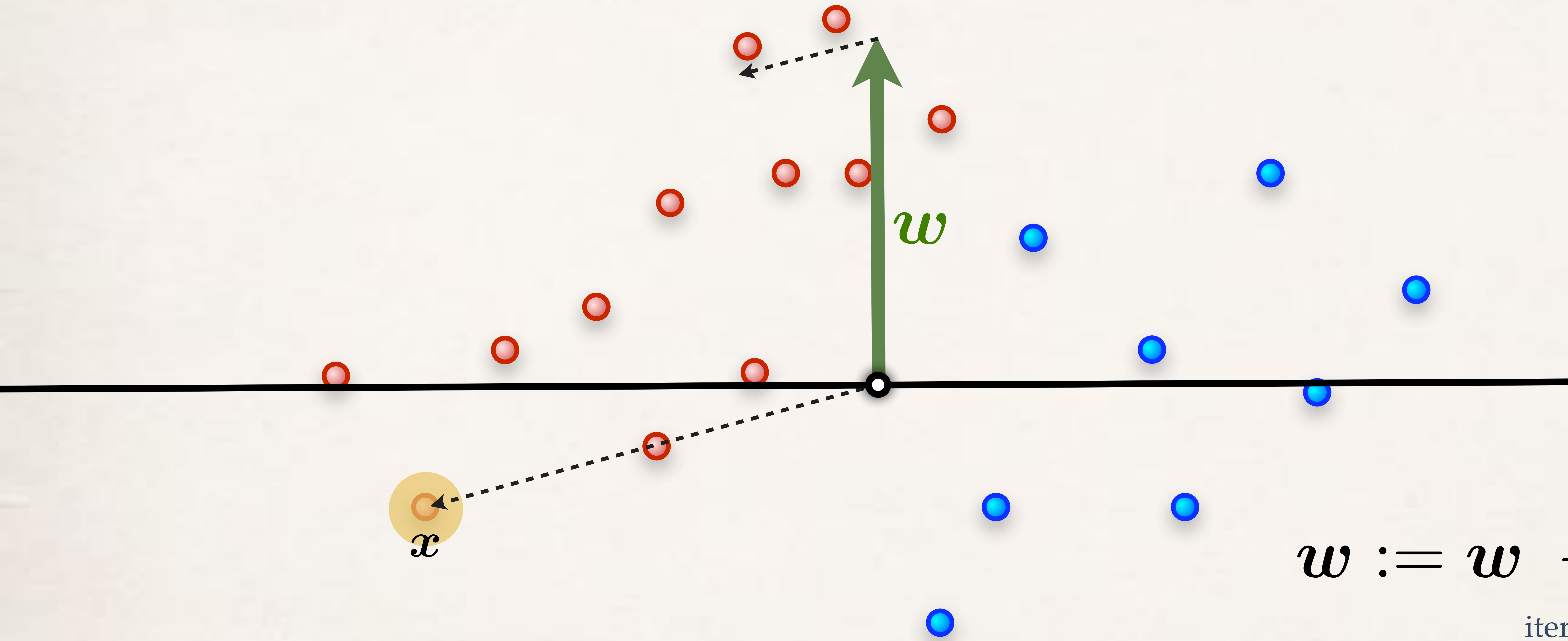
image source

Classification



The Learning Algorithm

$\boldsymbol{x}_i \in \mathbb{R}^d$



(Stochastic Gradient Descent)

$$\boldsymbol{w} := \boldsymbol{w} + \lambda \cdot \boldsymbol{x}$$

iteration cost: O(d)

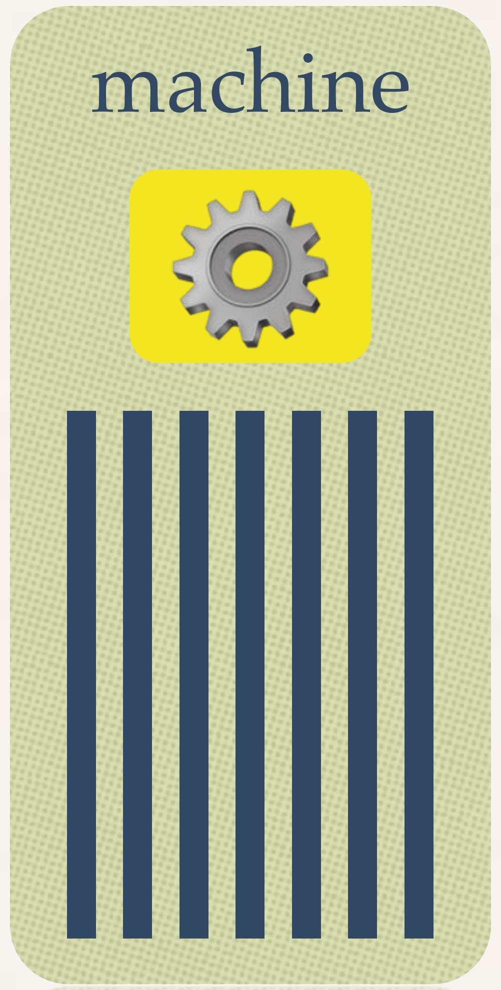
Perceptron

(Rosenblatt 1957)

Support-Vector-Machine

(Cortes & Vapnik 1995)

Machine Learning Systems



Machine Learning Systems

What if the data does not fit onto one computer anymore?

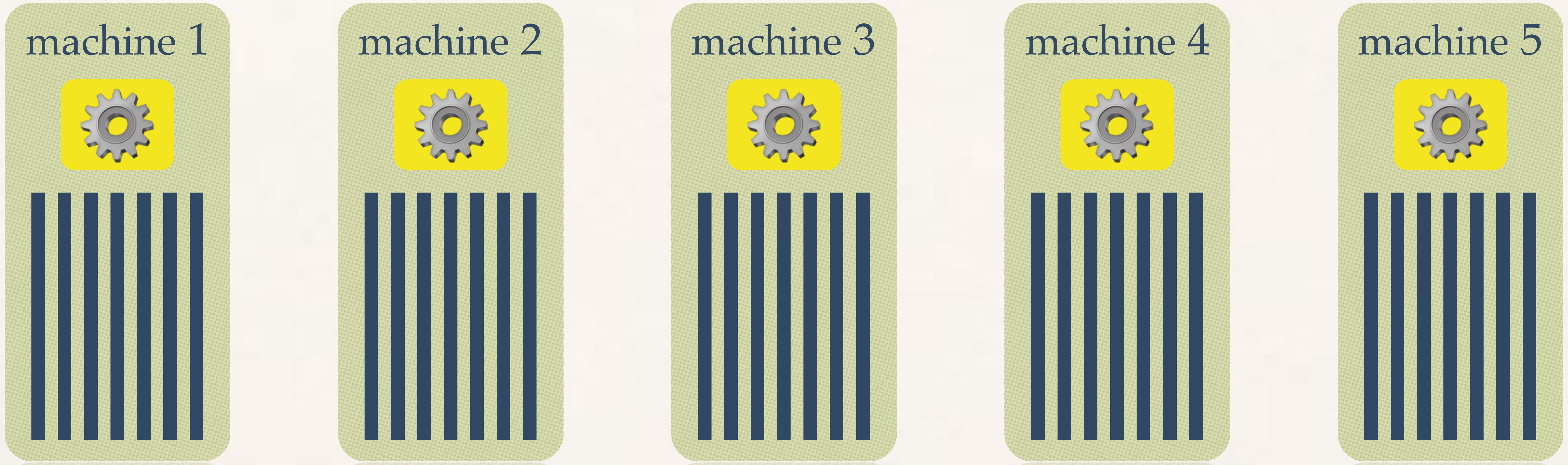
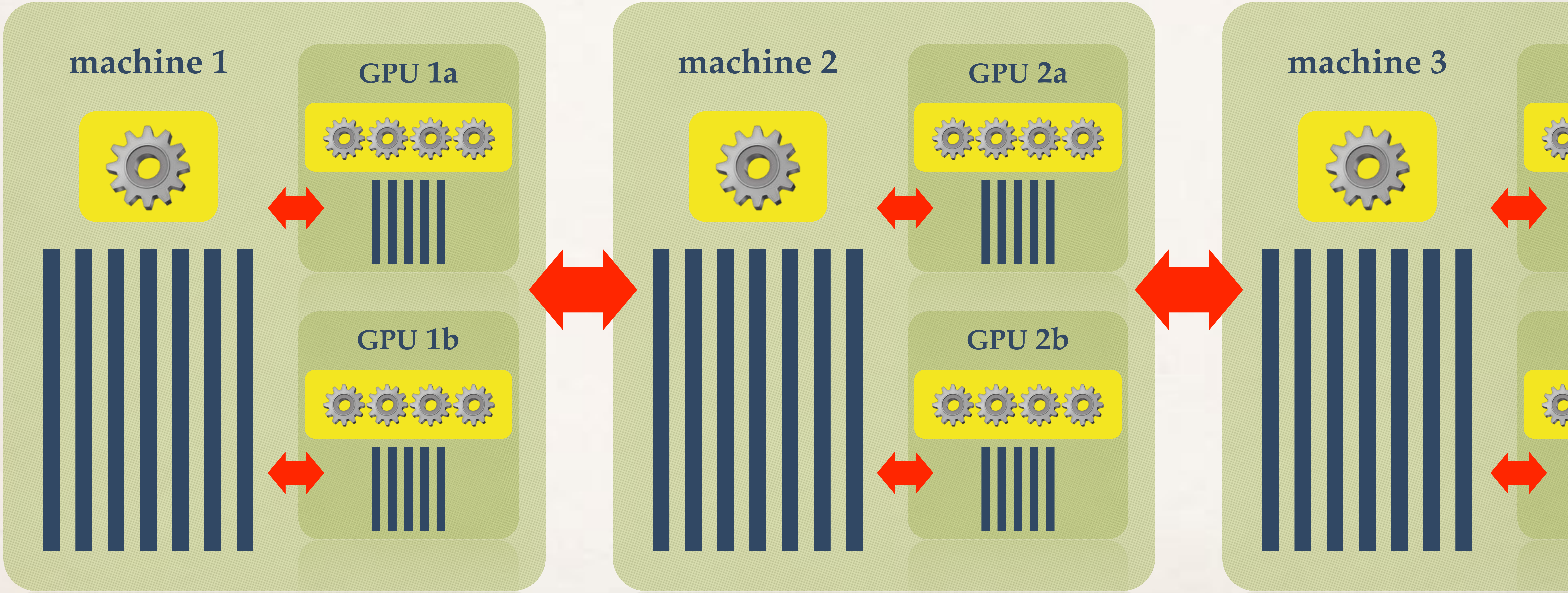



Foto: Florian Hirzinger

Machine Learning Systems



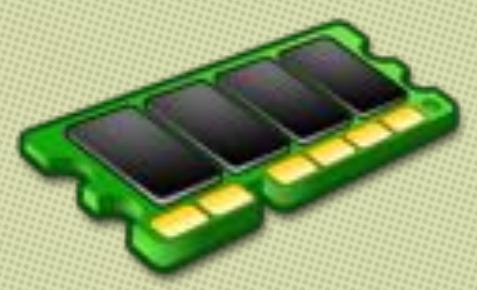
Challenge

The Cost of Communication

$\boldsymbol{v} \in \mathbb{R}^{100}$

- Reading $\boldsymbol{v}$ from memory (RAM)

  100 ns



- Sending $\boldsymbol{v}$ to another machine

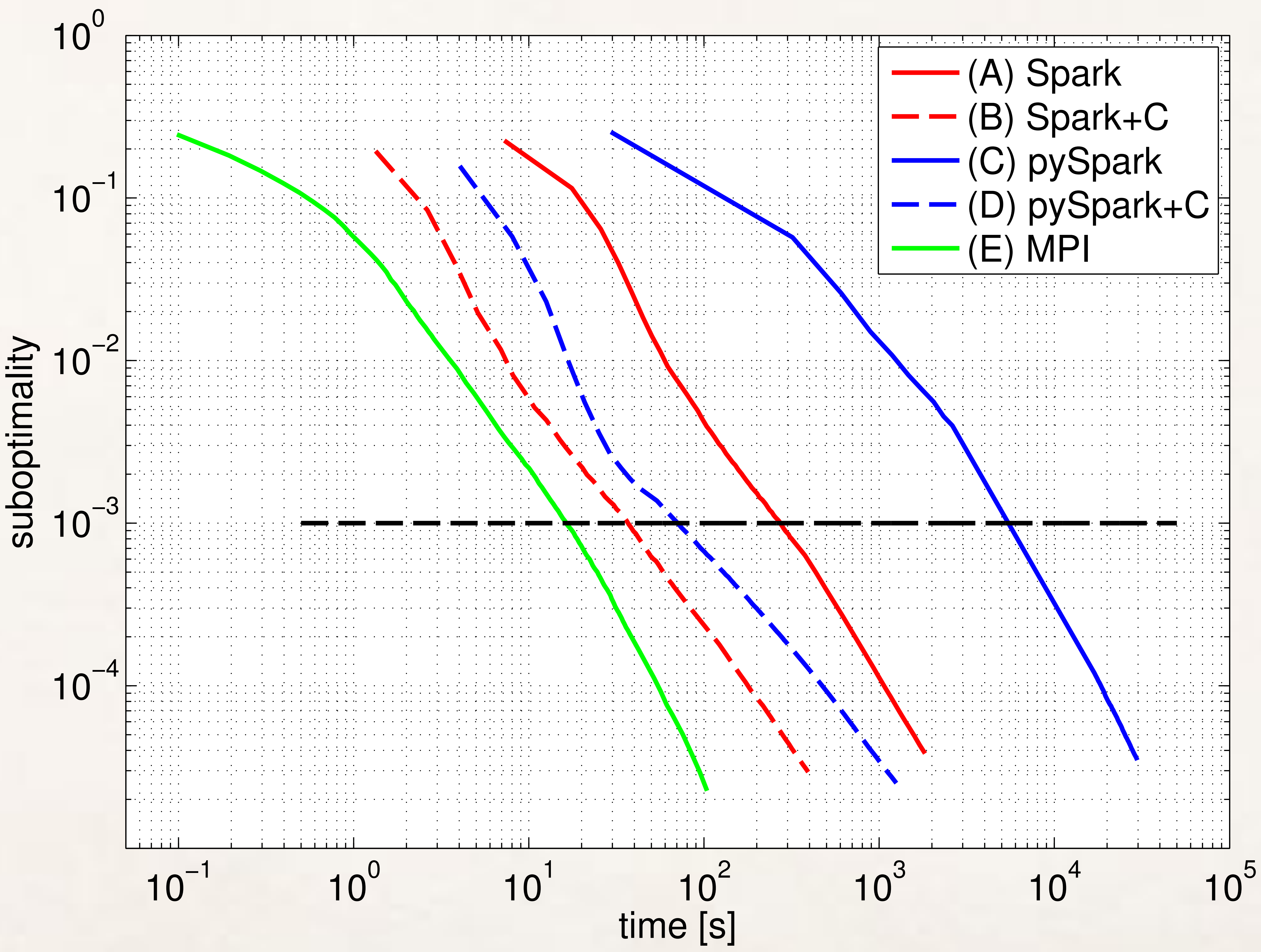
  500’000 ns

- Typical Map-Reduce iteration

  10’000’000’000 ns

Challenge

The Cost of Communication

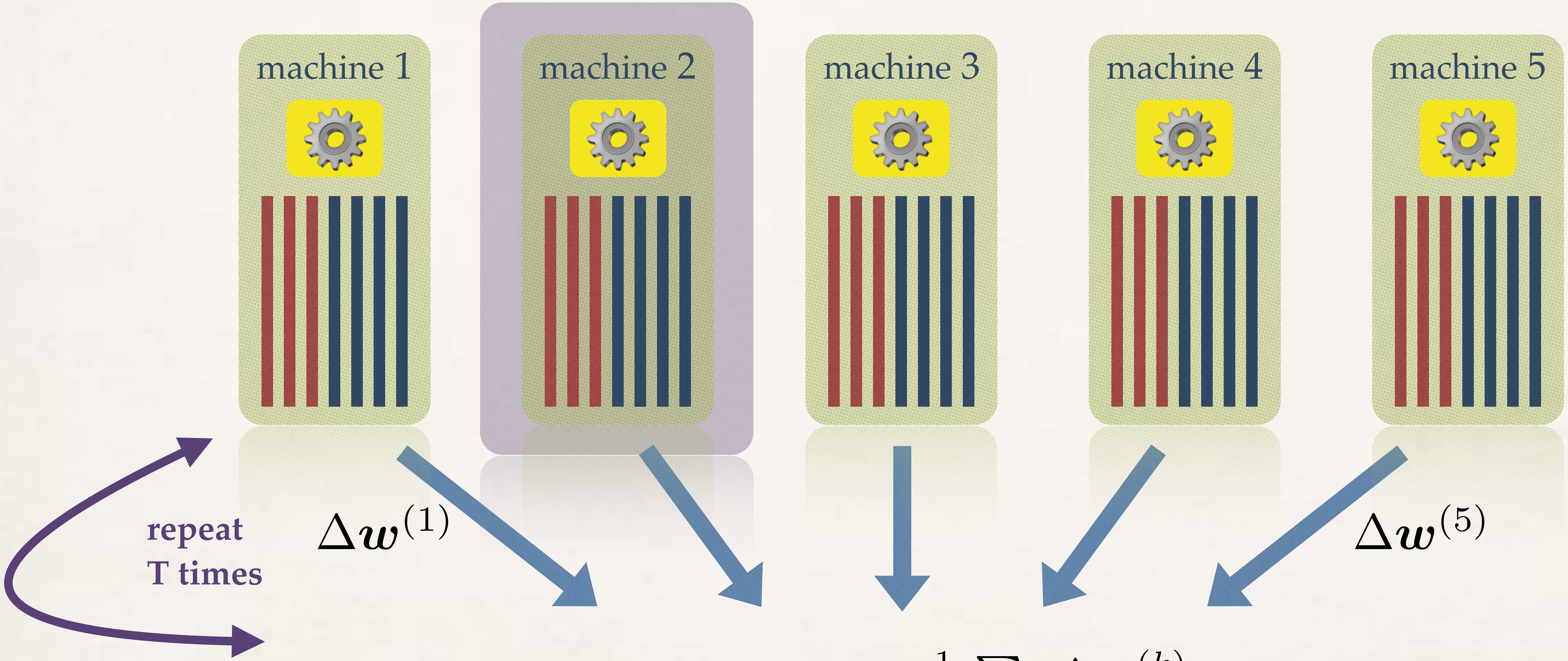


High-Performance Distributed Machine Learning using Apache Spark
Dünner et al. 2016, arxiv.org/abs/1612.01437

Problem class

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad f(A\boldsymbol{\alpha}) \; + \; g(\boldsymbol{\alpha})$$

CoCoA - Communication Efficient Distributed Optimization

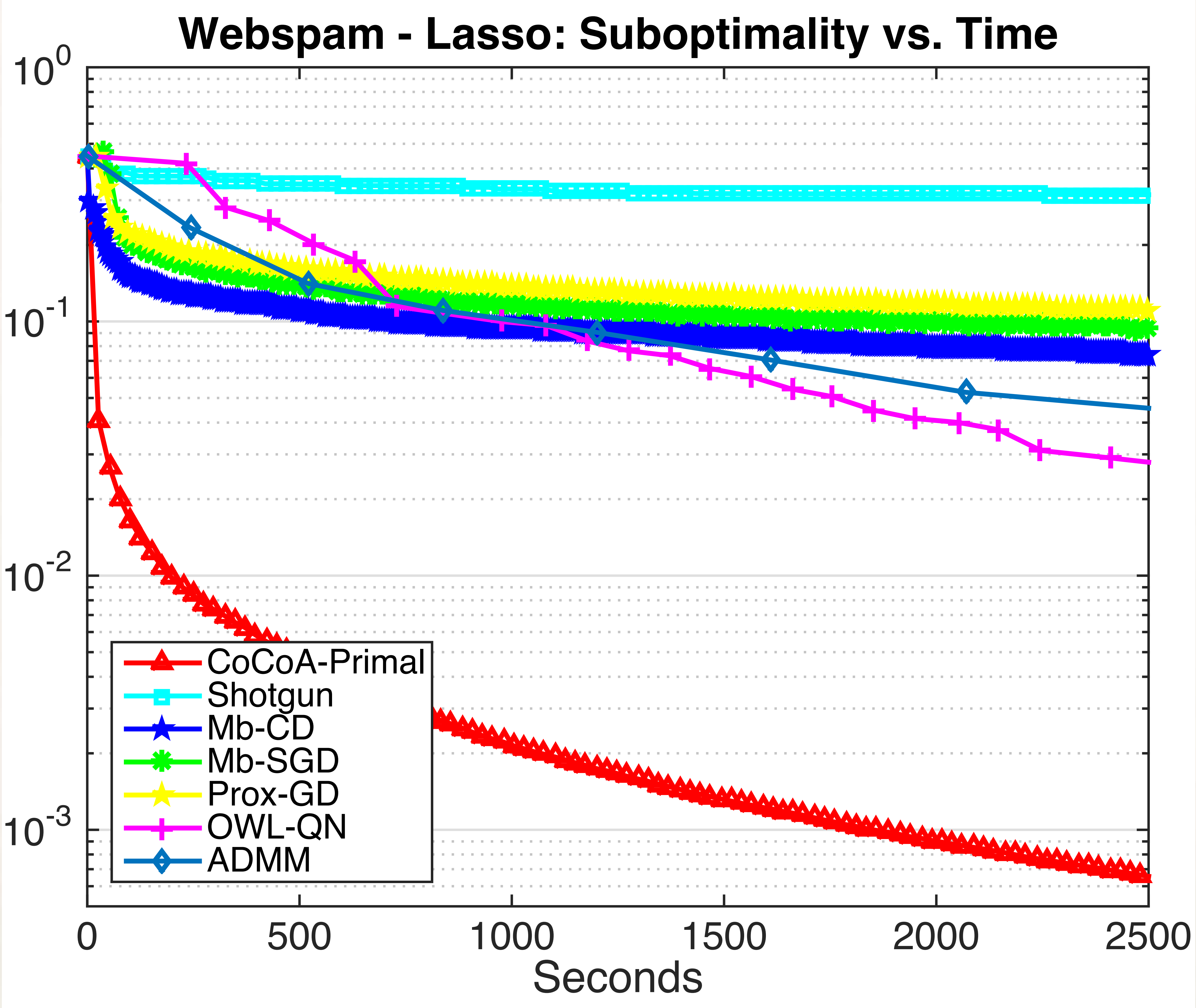


$$\boldsymbol{w} := \boldsymbol{w} + \frac{1}{K} \sum_k \Delta \boldsymbol{w}^{(k)}$$

Distributed Experiments

Dataset	Training	Features	Sparsity
url	2,396,130	3,231,961	3.5e-3%
epsilon	400,000	2,000	100%
kddb	19,264,097	29,890,095	9.8e-5%
webspam	350,000	16,609,143	0.02%

Sparse Linear Regression



NIPS 2014, ICML 2015,
arxiv.org/abs/1611.02189

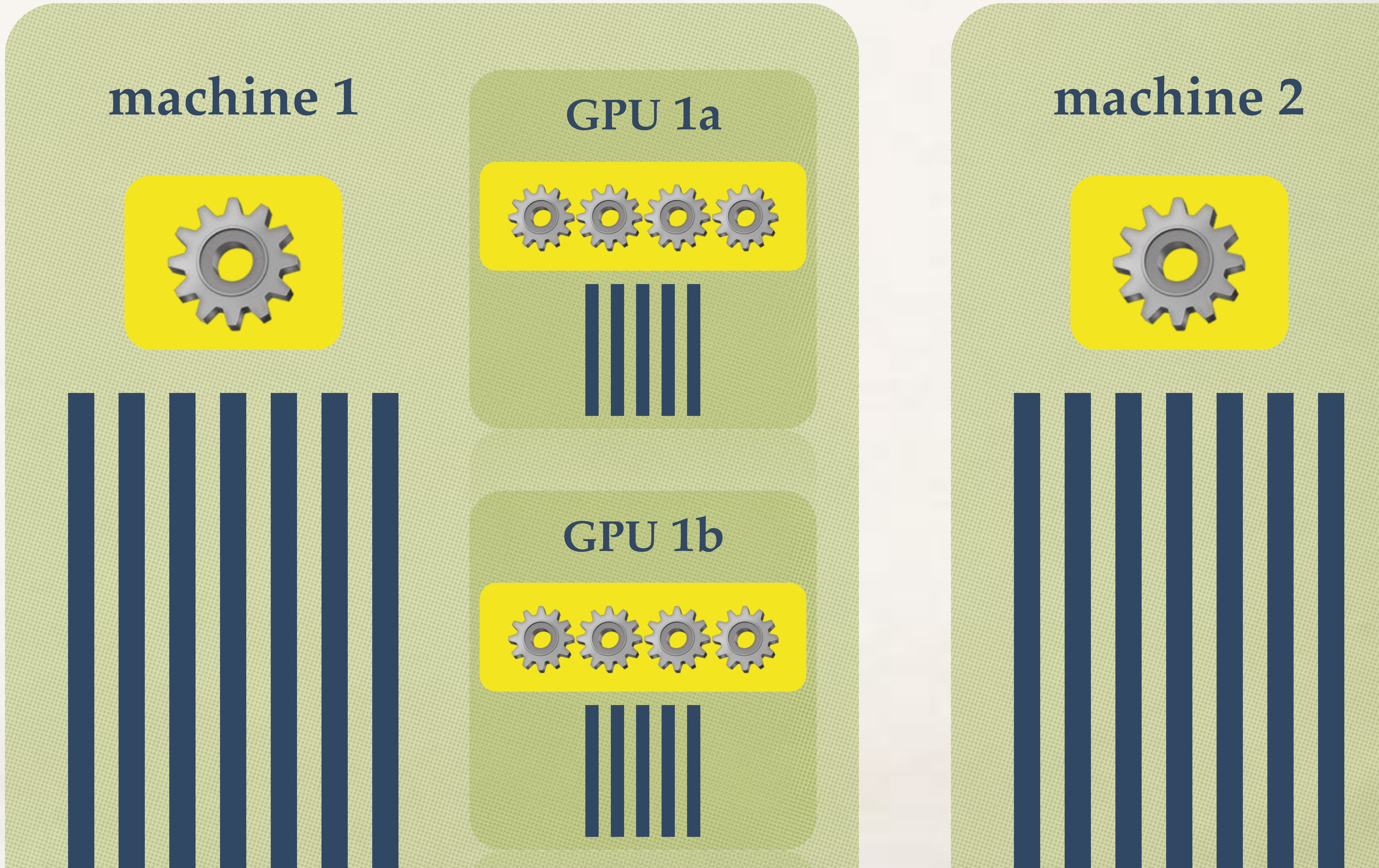
Spark Code:
github.com/gingsmith/proxcocoa

\+ TensorFlow
\+ Apache Flink

Challenge

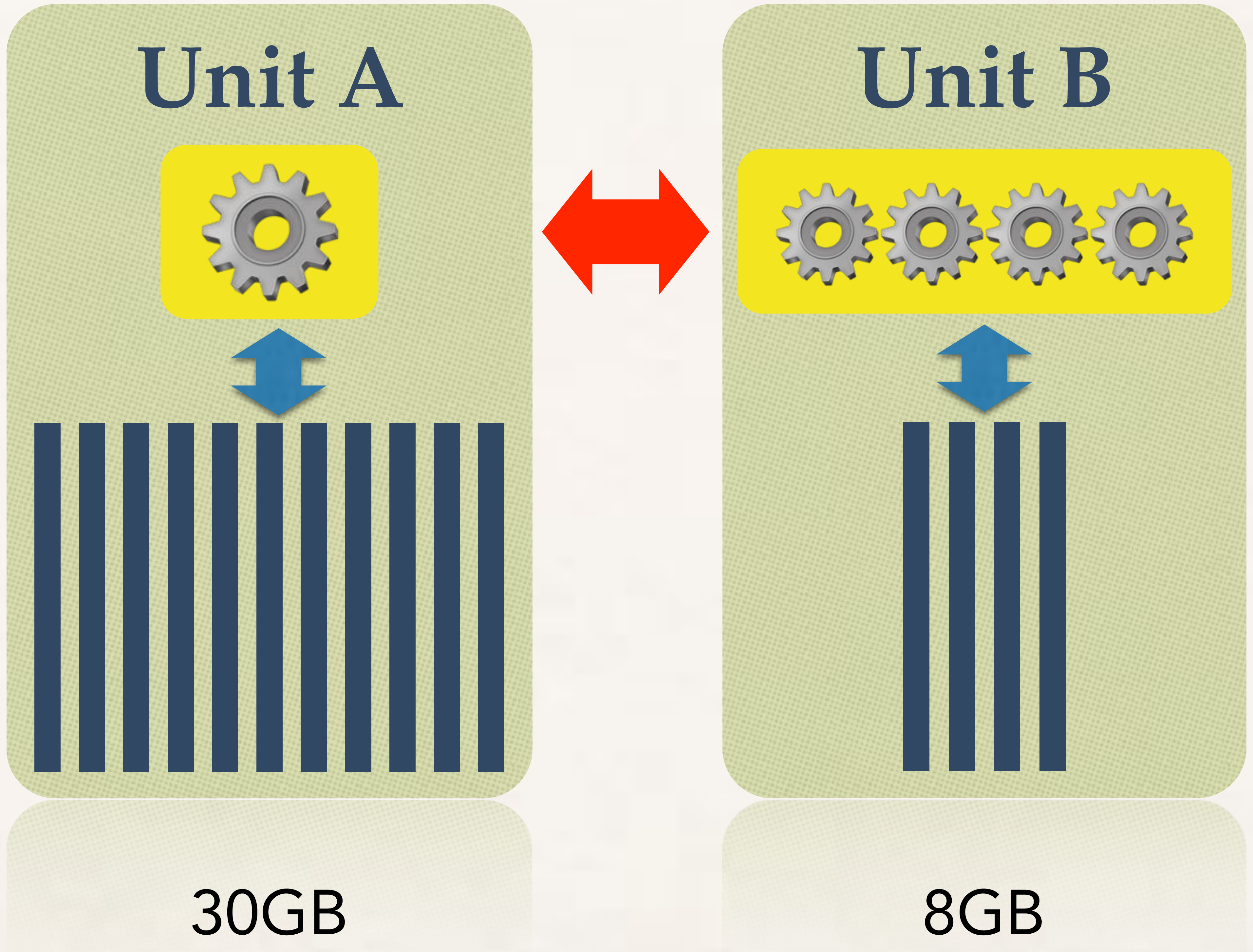
Leveraging Memory Hierarchy

Which data to put in which memory?



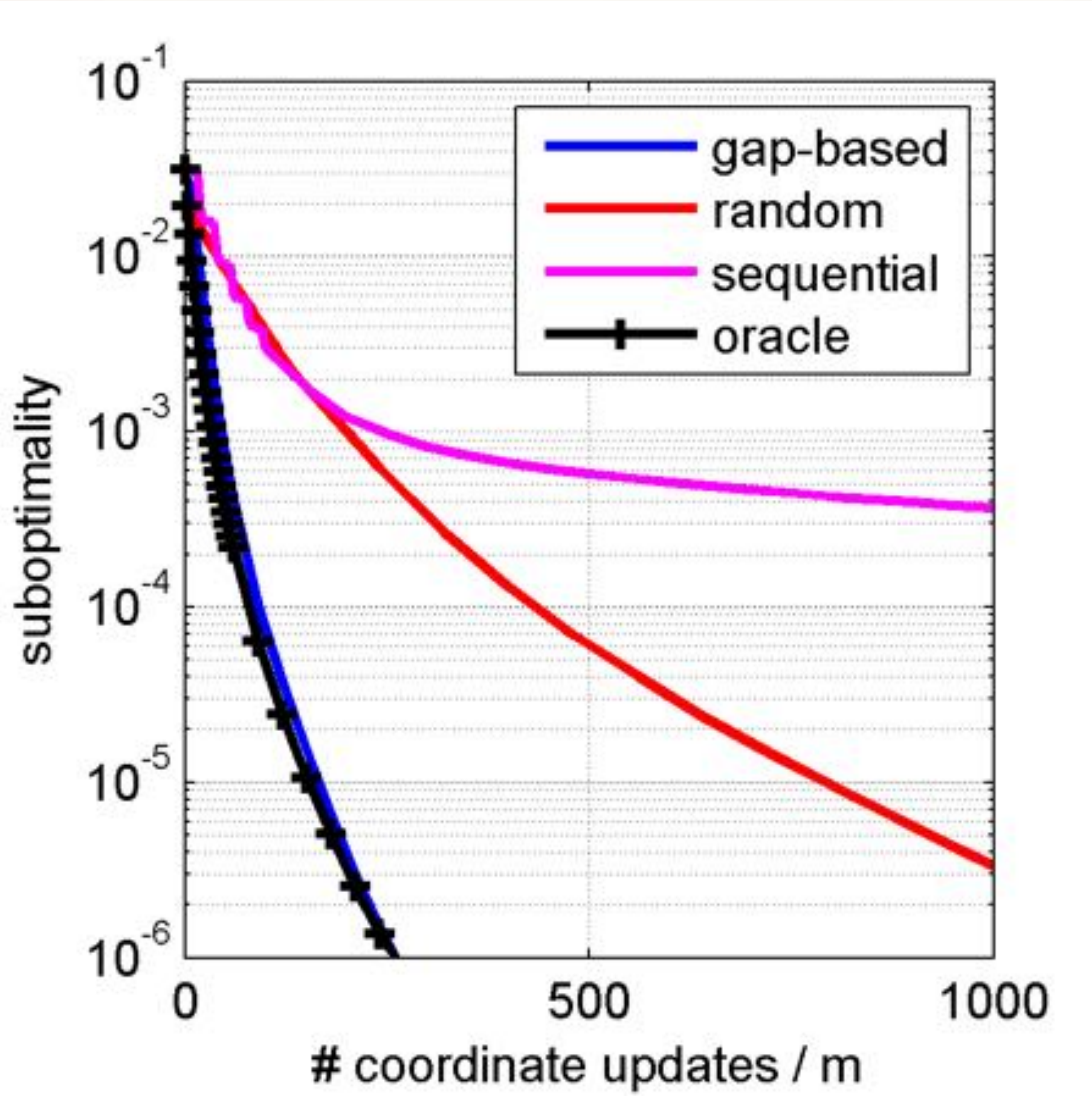
Leveraging Memory Hierarchy

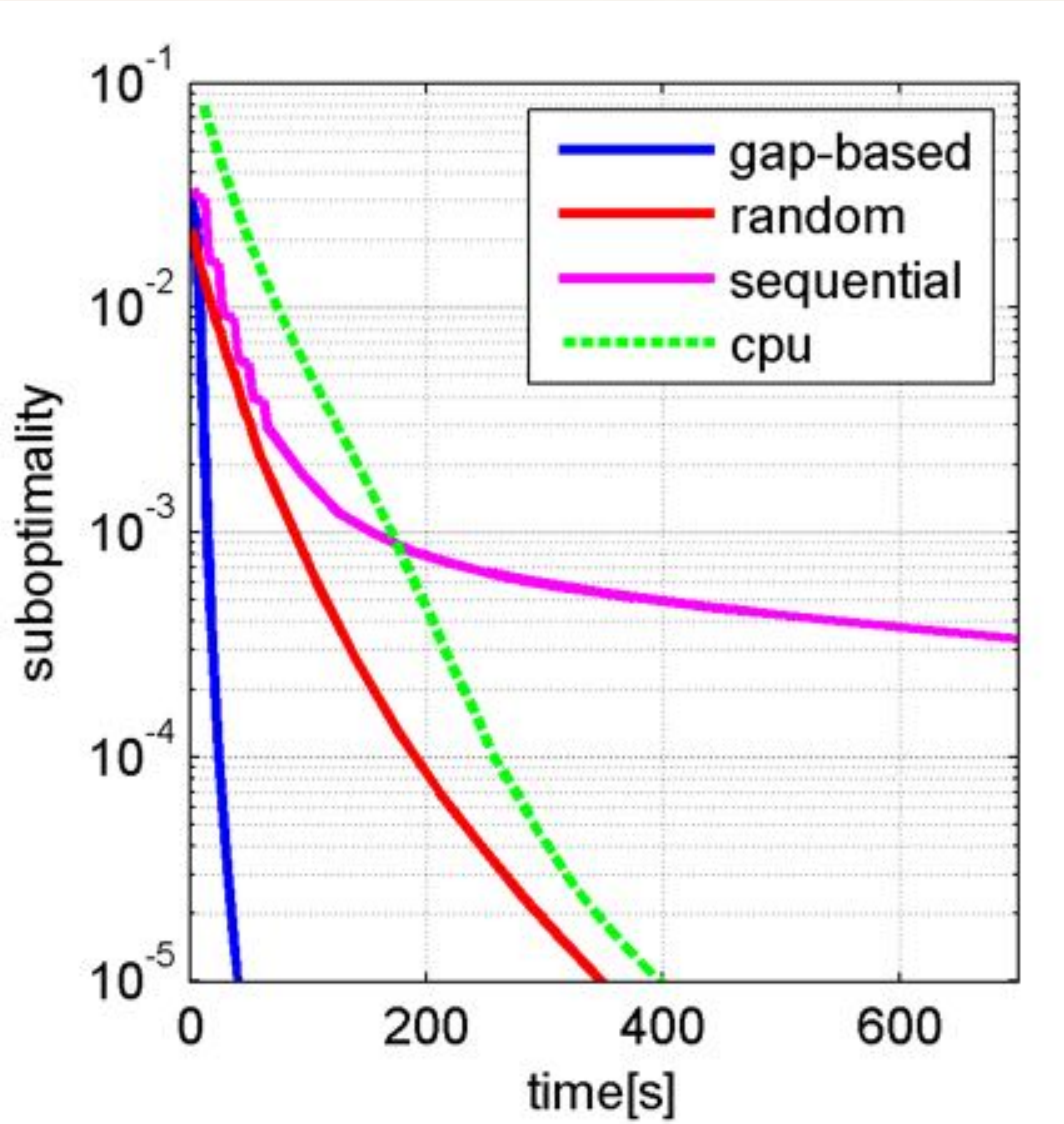
duality gap as selection criterion



Experiments

Sparse Linear Regression, RAM ⟺ GPU



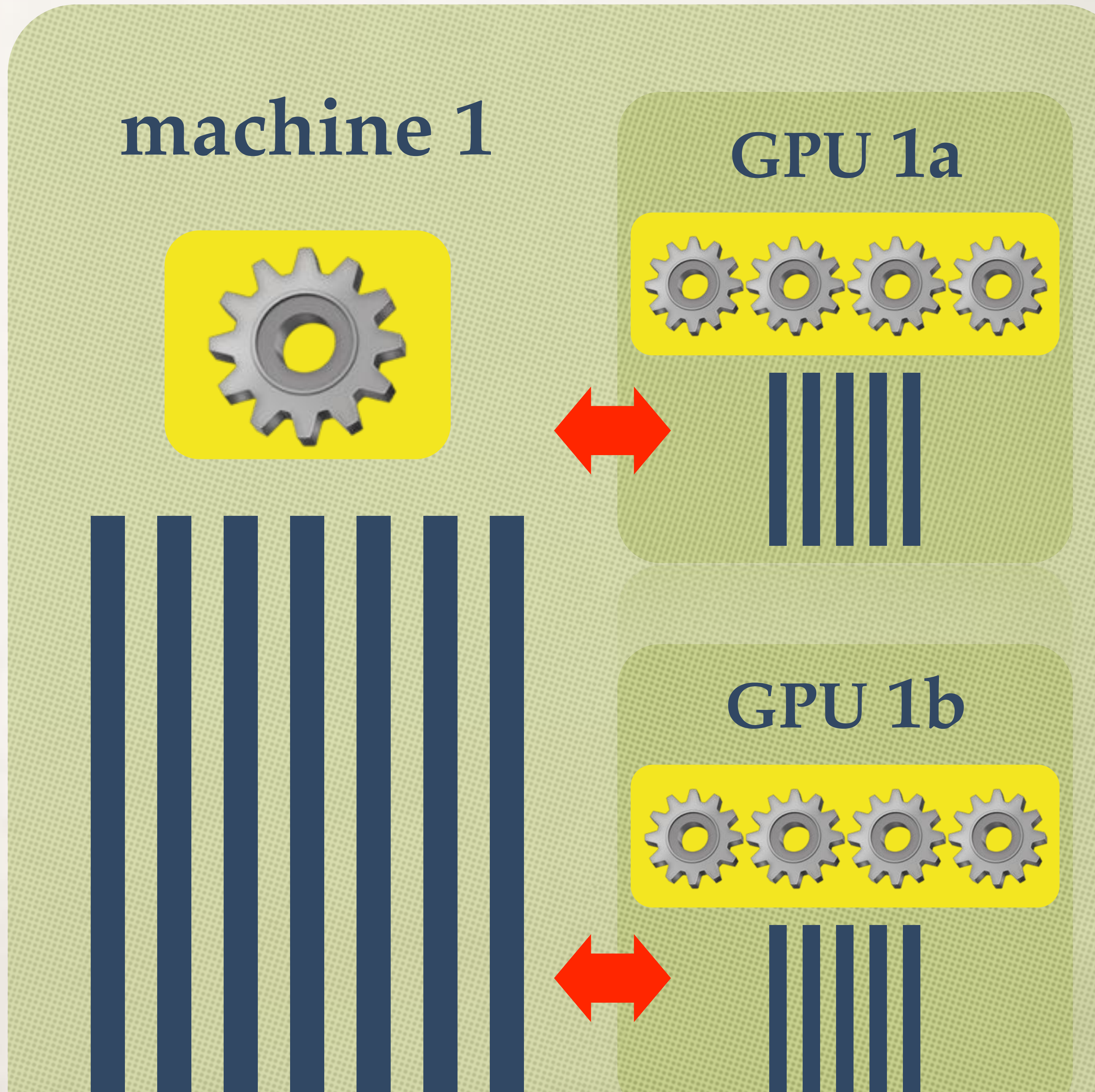


Conclusion

- try to improve **usability** of large-scale ML
- full **adaptivity** to the communication cost, memory hierarchy and bandwidth
- **re-usability** of good single machine solvers
- **accuracy** certificates

Open Research

- **limited precision operations** for efficiency of communication and computation
- **asynchronous** and **fault tolerant** algorithms
- **multi-level approach** on heterogenous systems
- more **re-usable** algorithmic building blocks
  - for more systems and problems



Project:

Distributed Machine Learning Benchmark

Goal:

Public and Reproducible
Comparison of Distributed Solvers

github.com/mlbench/mlbench

Apache Spark

Google TensorFlow™

Apache REEF

HPC MPI

Thanks!

mlo.epfl.ch

Celestine Dünner, Virginia Smith, Simone Forte, Chenxin Ma, Martin Takac,
Dmytro Perekrestenko, Volkan Cevher, Michael I. Jordan, Thomas Hofmann

Matrix Factorizations

$$\min_{\boldsymbol{U},\boldsymbol{V}} f(\boldsymbol{U}\boldsymbol{V}^\top)$$

Optimization: Primal-Dual Context

$$A_{\text{loc}} \Delta \boldsymbol{\alpha}_{[k]} + \boldsymbol{w}$$

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad \left[\; \mathcal{O}_A(\boldsymbol{\alpha}) := f(A\boldsymbol{\alpha}) + g(\boldsymbol{\alpha}) \; \right]$$

primal Lasso
dual L2-reg SVM/Log-Regr
primal L1-reg SVM/Log-Reg

correspondence

$$\boldsymbol{w} := \nabla f(A\boldsymbol{\alpha})$$

$$\min_{\mathbf{w} \in \mathbb{R}^d} \quad \left[\; \mathcal{O}_B(\mathbf{w}) := g^*(-A^\top \mathbf{w}) + f^*(\mathbf{w}) \; \right]$$