Coded Elastic Computing

Yaoqing Yang*, Matteo Interlandi**, Pulkit Grover*, Soummya Kar*, Saeed Amizadeh**, Markus Weimer**

* Carnedie Mellon University ** Microsoft

Introduction

Cloud providers introduced low-priority machines to reduce the computation cost, but these machines can

- **elastically leave** through preemption (up to 90%);
- **unpredictably join** the computation at any time.

Thus, we need to design fault-aware techniques that can

- **transparently continue** the computation in the presence of preemptions;
- **positively utilize** the newly available resources in a fast and adaptive way;
- **seamlessly transit** between different configurations with little or no data movement;

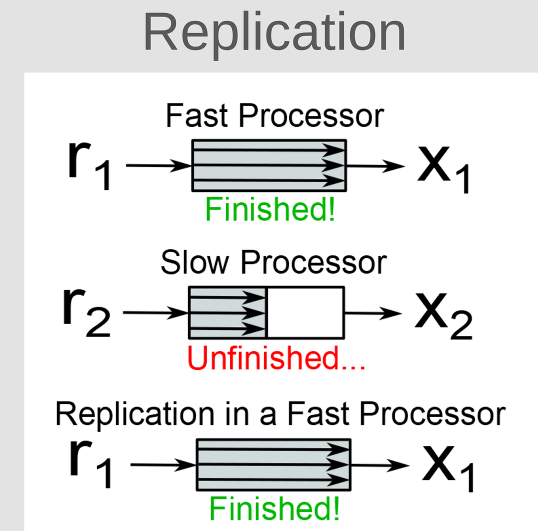
Definition of elasticity: seamless transitions between ***optimal*** configurations with ***zero*** data movement at existing machines.

Existing Techniques

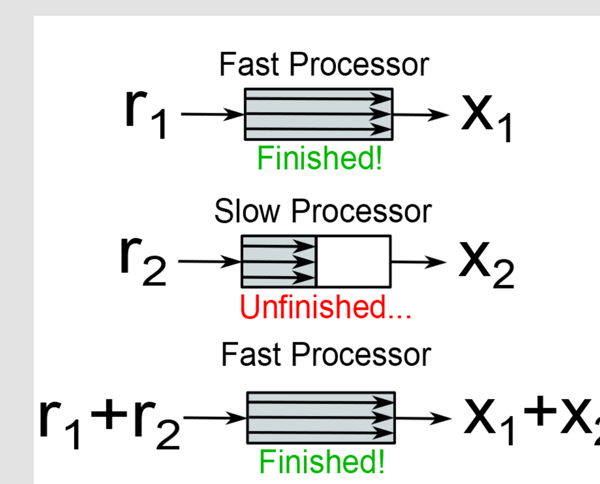
- "**Stop-the-world**": cannot deal with frequent and a large number of machine preemptions;
  may achieve *zero* progress according to our observation.
- **Ignore:** can lead to reduced learning performance;
  not acceptable from a customer's perspective.
- **Dynamic task allocation**: requires frequent data movement.
- **Algorithm-based elastic computing [1]**: fault-dependent
  variance on the final result.

Background: Introduction to Coded Computing [2-7]

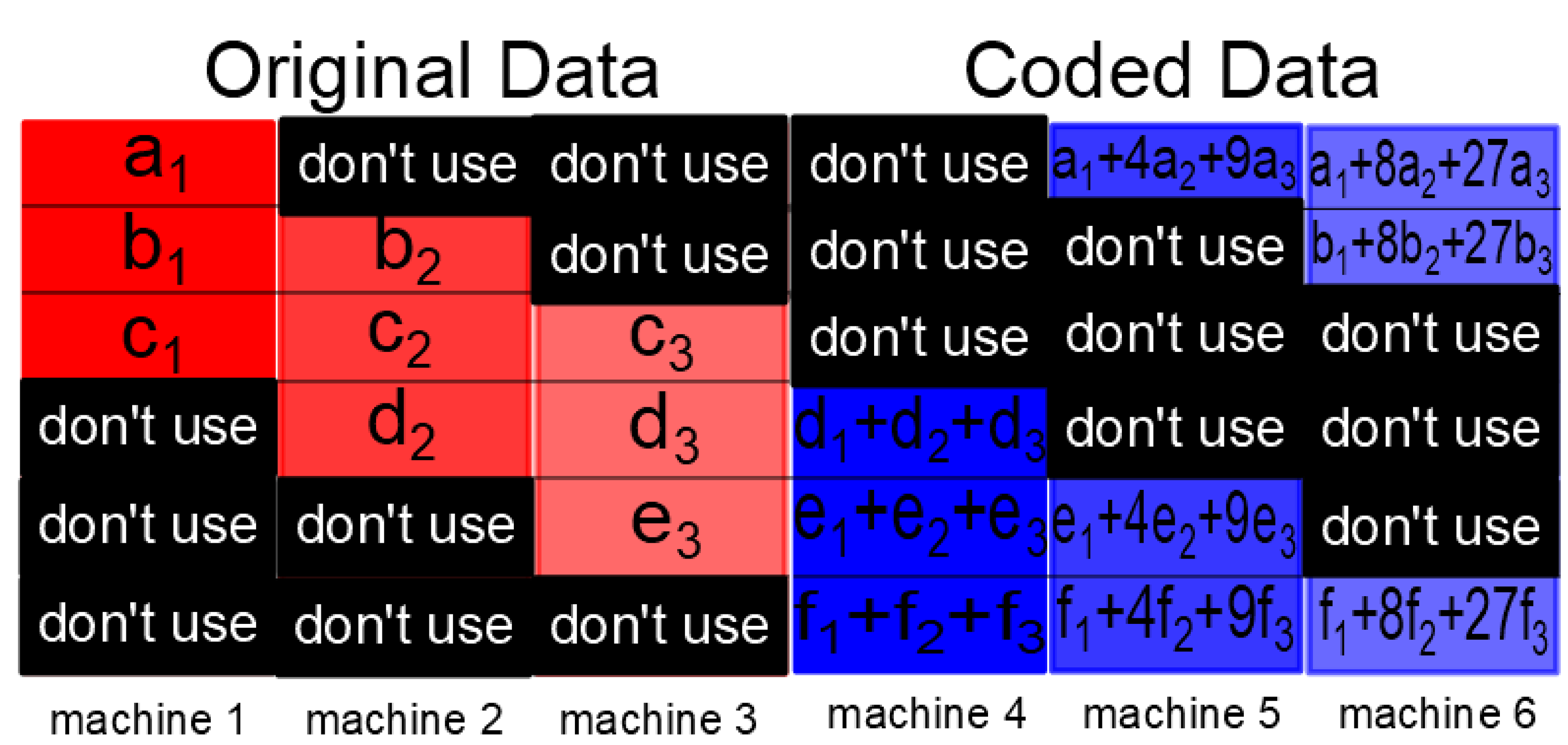
Replication

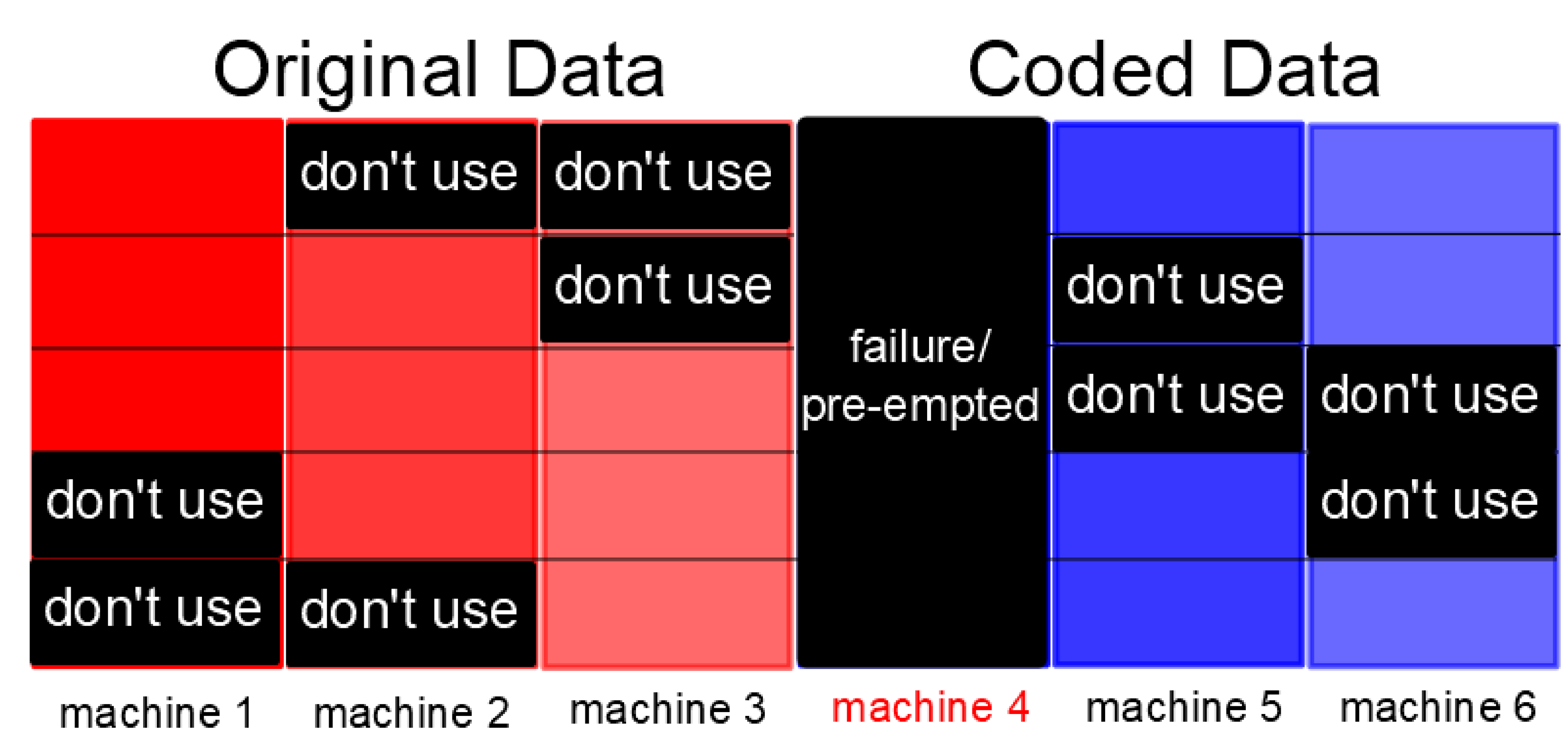


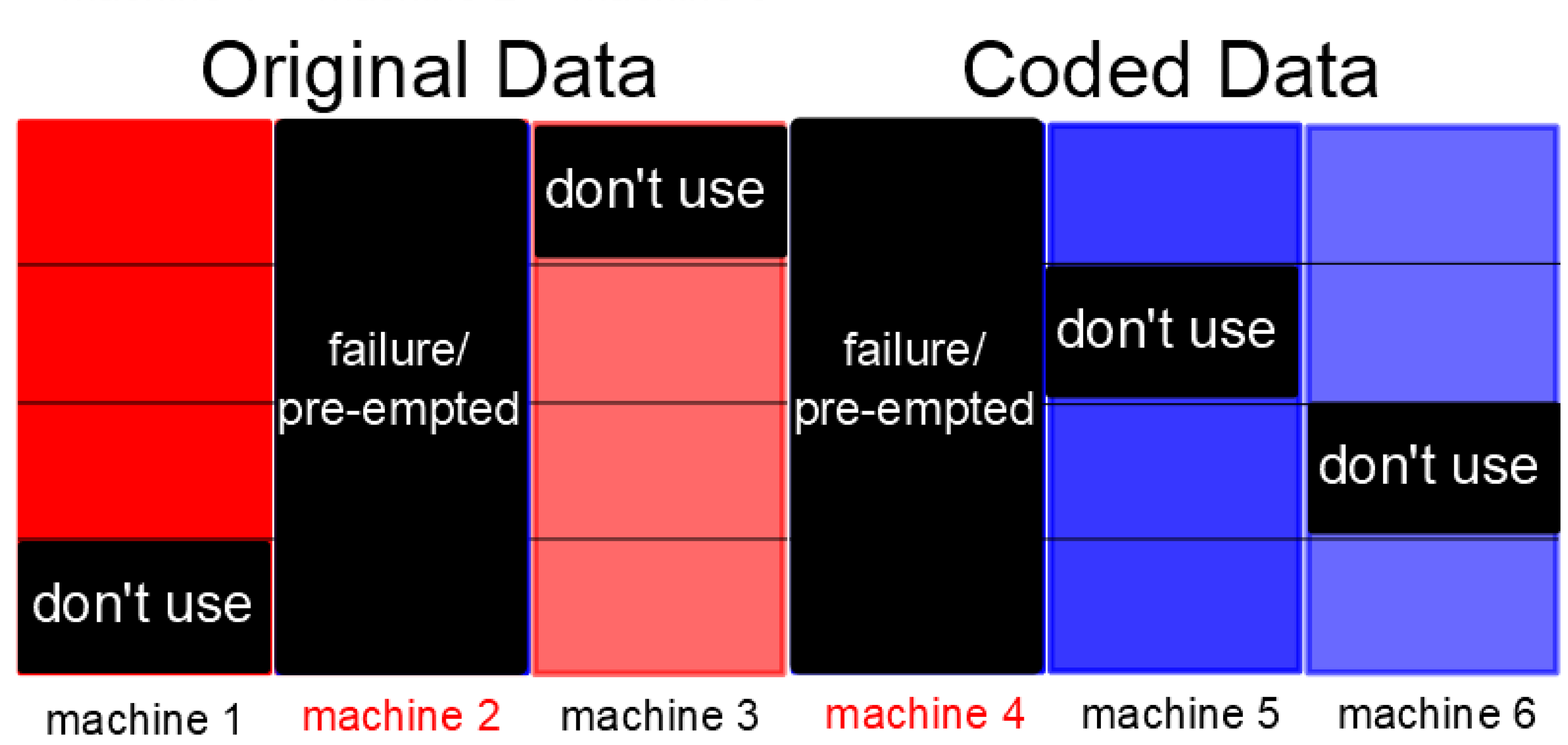
coded computing

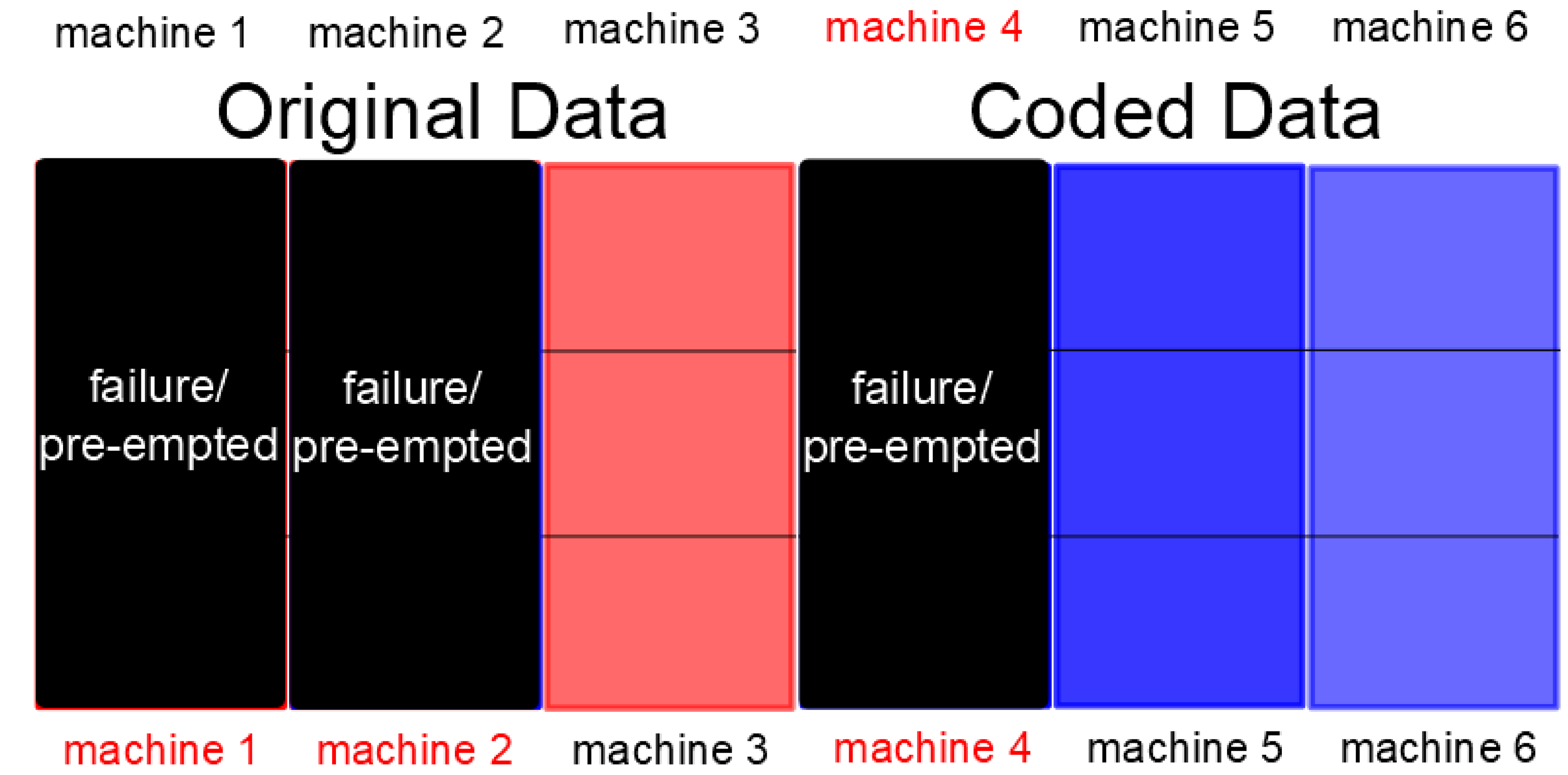


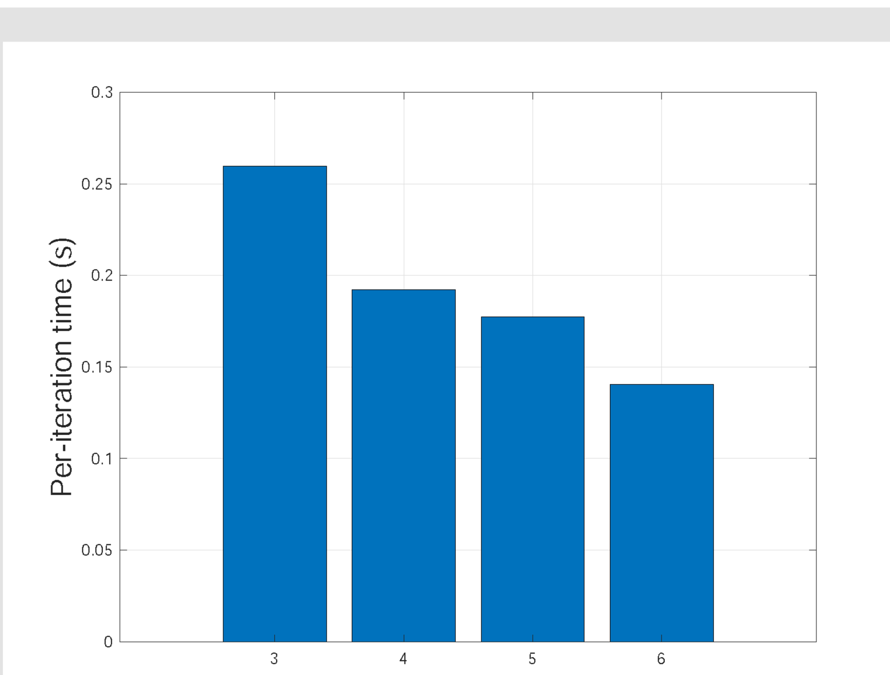
Coded Elastic Data Partitioning Example: number of machines from 6 to 3



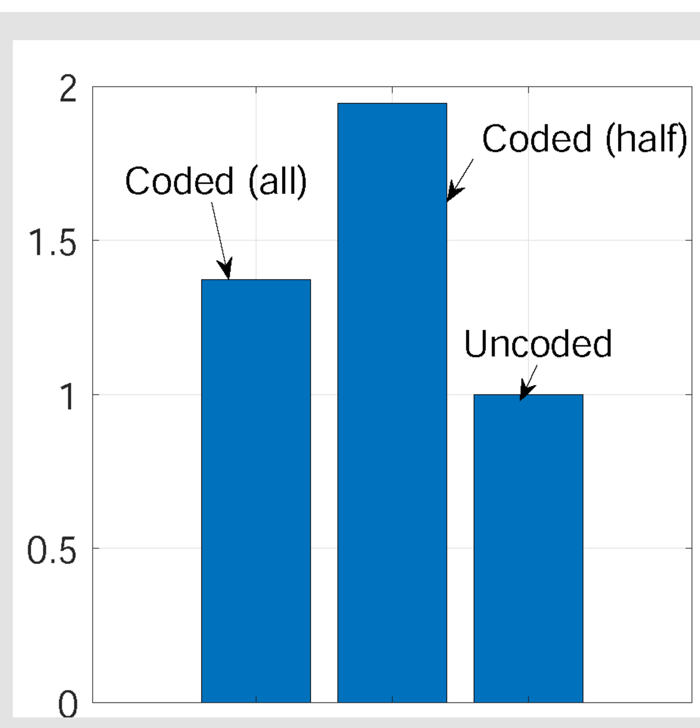




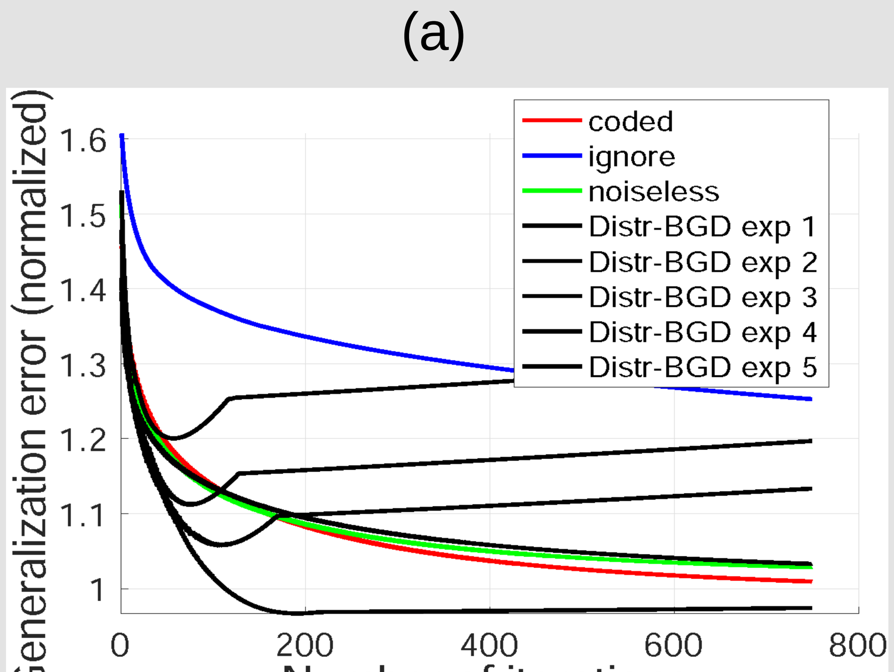




(a)



(b)



(c)

(a) matrix-vector mini-benchmark experiment on Amaon EC2: reduced computation time

(b) Linear model mini-benchmark experiment on Apache REEF [11]: computation time overhead

(c) Generalization error of the linear model: Distr-BGD can overfit and cross the optimal point (we use line search)

Result Analysis

(a) Matrix multiplications: can achieve elastic transitions between optimal configuration points (optimal in storage cost and computation complexity) with zero data movement.

(b) Linear model: can maintain all the data even when there are pre-emption type of failures; have near optimal convergence.

(c) The decoding cost cannot be neglected, although in scaling-sense vanishing when the dimension of data is large.

References

[1] S. Narayanamurthy, M. Weimer, D. Mahajan, T. Condie, S. Sellamanickam, and S. S. Keerthi, ***Towards resource-elastic machine learning***, NIPS 2013 BigLearn Workshop

[2] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr. ***A fundamental tradeoff between computation and communication in distributed computing***, IEEE Transactions on Information Theory

[3] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran. ***Speeding up distributed machine learning using codes***, ISIT 2016.

[4] S. Dutta, V. Cadambe, and P. Grover. ***Short-dot: Computing large linear transforms distributedly using coded short dot products***, NIPS 2016.

[5] R. Tandon, Q. Lei, A. Dimakis, N. Karampatziakis, ***Gradient Coding: Avoiding Stragglers in Distributed Learning***, ICML 2017

[6] N. S. Ferdinand and S. C. Draper. ***Anytime coding for distributed computation***. Allerton 2016.

[7] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr. ***Polynomial codes: an optimal design for high-dimensional coded matrix multiplication***. NIPS 2017.

[8] C. Karakus, Y. Sun, S. Diggavi, W. Yin, ***Straggler Mitigation in Distributed Optimization Through Data Encoding***, NIPS 2017

[9] RK Maity, AS Rawat, A Mazumdar , Robust Gradient Descent via Moment Encoding with LDPC Codes, SysML 2018

[10] Z. Charles, D. Papailiopoulos, J. Ellenberg, **Approximate gradient coding via sparse random graphs**

[11] M. Weimer, Y. Chen, B. Chun, T. Condie, C. Curino, C. Douglas, Y. Lee, T. Majestro, D. Malkhi, S. Matusevych, ***Reef: Retainable evaluator execution framework***, Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 1343—1355