ASYNC: A Cloud Engine with Asynchrony and History for Distributed Machine Learning

Saeed Soori[1], Bugra Can[2],

Mert Gurbuzbalaban[2], Maryam Mehri Dehnavi[1]

sasoori@cs.toronto.edu

University of Toronto[1], Rutgers University[2]

Machine Learning on Cloud

- Machine learning algorithms need to scale on cloud systems.
- Cloud systems are cheap but faulty.
- Nodes can be slow and stall the server.

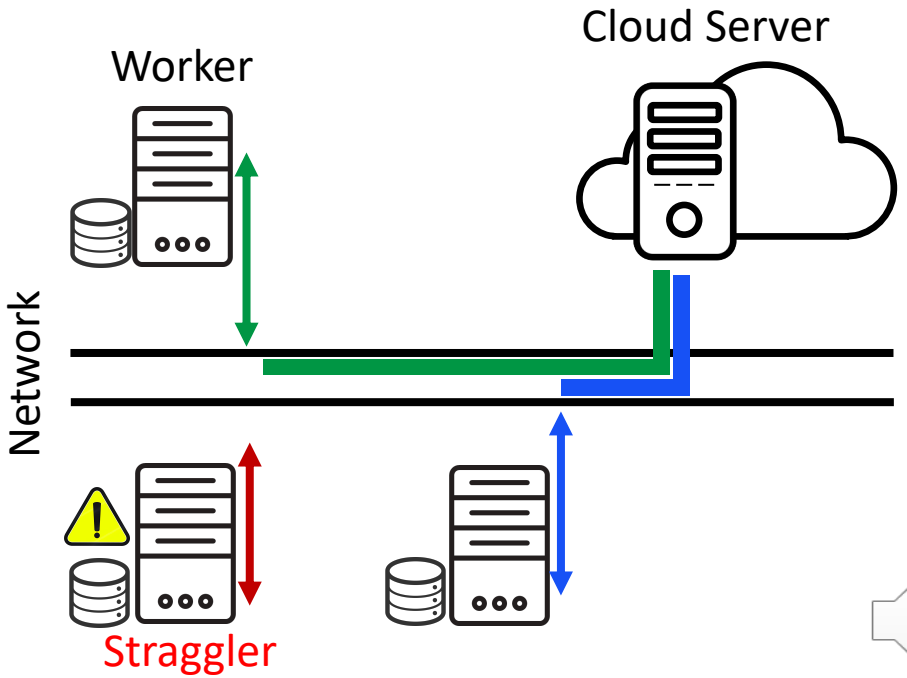
Finance sciences

Traffic control

Medical sciences

facebook

Web related applications



Optimization in Machine Learning

- In many machine learning application, a model is obtained by applying an optimization algorithm on noisy data.
- Recent optimization algorithms must support asynchrony and history to mitigate the effect of slow workers and noise.
- Existing frameworks:
  - Create static dependency graphs to implement asynchrony[1].
  - Store history using checkpointing or lineage-based methods[2].

[1] S. Wang, J. Liagouris, R. Nishihara, P. Moritz, U. Misra, A. Tumanov, and I. Stoica, "Lineage stash: Fault tolerance off the critical path," in Proceedings of Symposium on Operating Systems Principles, SOSP, vol. 19, 2019.
[2] M. Zaharia et al., "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in Proceedings of the9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012, pp. 2–2.

Optimization in Machine Learning

- Problems with current tools:
  - Limited to asynchrony with fixed execution model throughout the algorithm.
  - Overhead of lineage and checkpointing is high.

- Our work:
  - Supports asynchrony by implementing dynamic dependency graphs.
  - Recovers history by partial model updates.

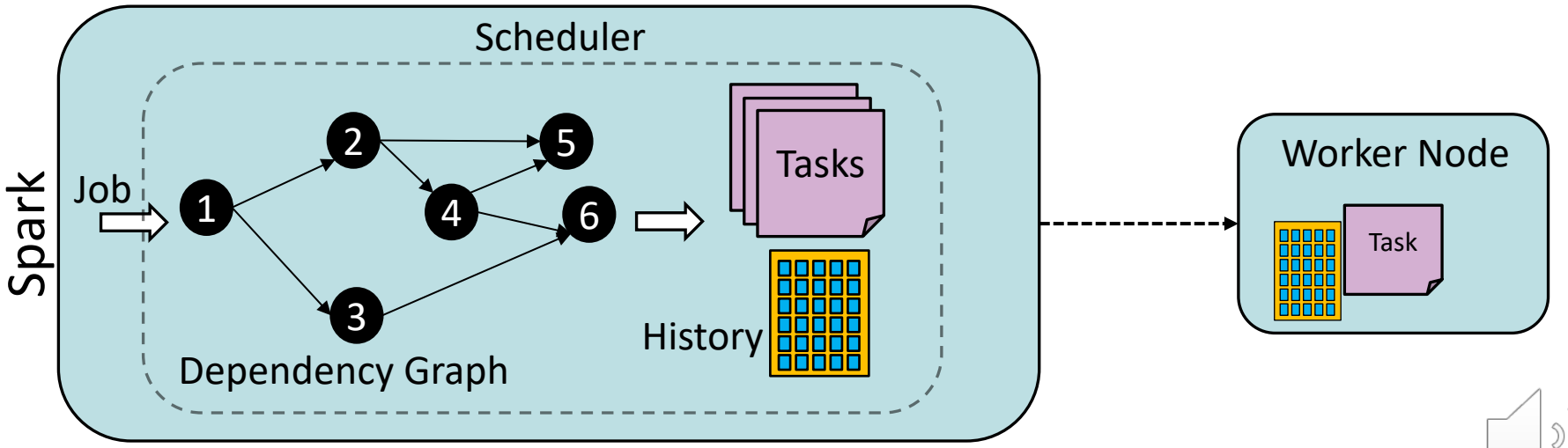
- We build our framework on top of Apache Spark[1].

[1] M. Zaharia et al., "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in Proceedings of the9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012, pp. 2–2.

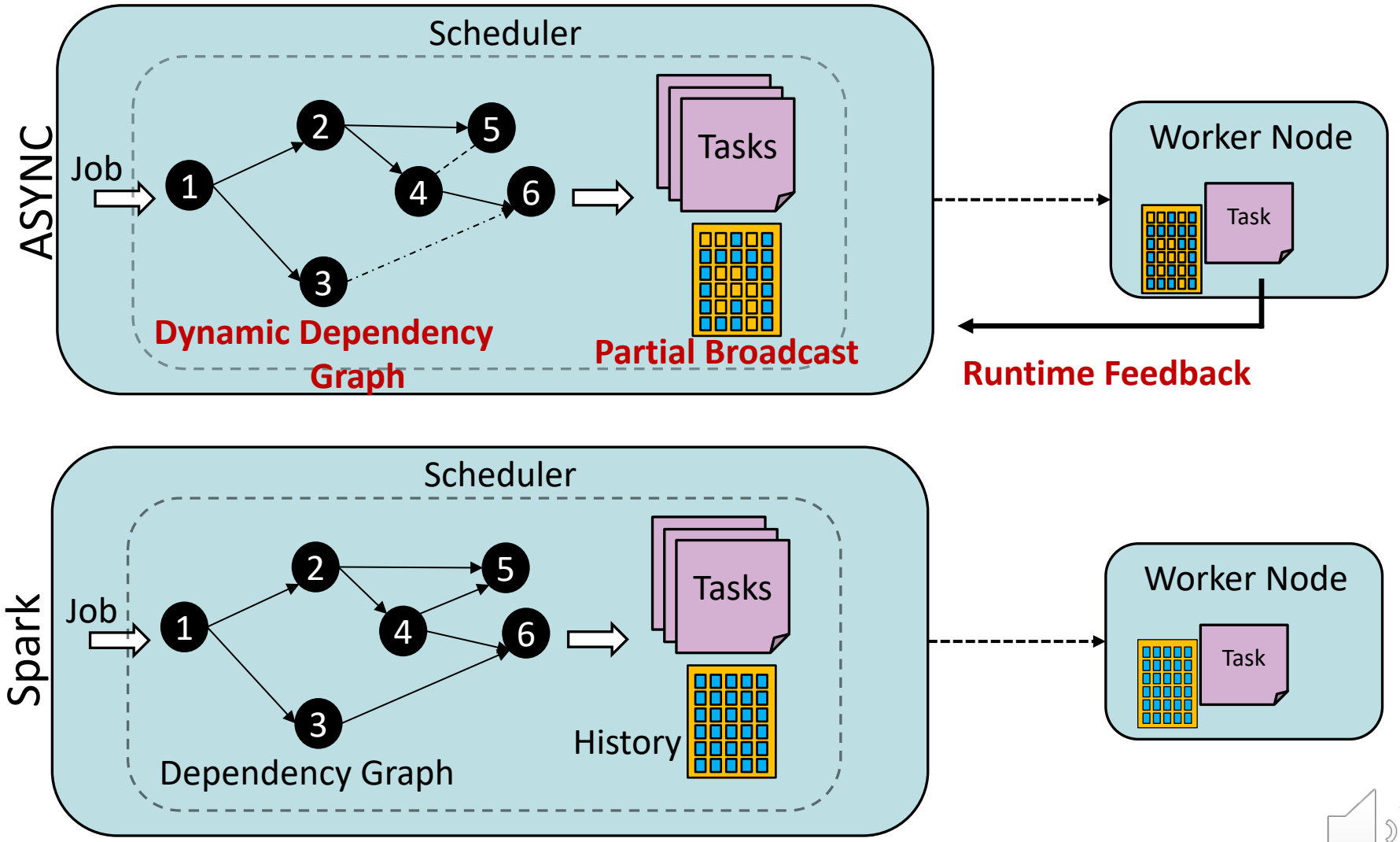
ASYNC vs SPARK: INTERNALS

- Spark uses a fixed dependency graph with periodic synchronization => relax the dependencies.

- It sends the bulky history with tasks => send partial history and recover the rest.



ASYNC vs Spark: Internals



ASYNC vs Spark: Code

- ASYNC uses a programming model written on top of Spark.
- In ASYNC, the user only adds the blue lines to convert a synchronous optimization method to an asynchronous variant.

```
Input: points, learning rate α
Output: model ω

for i = 1:N
    //broadcast model workers
    broadcast(ω)
    //compute gradients
    ∇f = points.map(...).reduce()
    //update model
    ω = ω − α∇f
end
```

Spark

```
Input: points, learning rate α
Output: model ω

for i = 1:N
    //partial broadcast
    ASYNCbroadcast(ω)
    //apply custom asynchrony
    points.ASYNCbarrier(...)
    ∇f = points.map(...).ASYNCreduce()
    ω = ω − α∇f
end
```

ASYNC

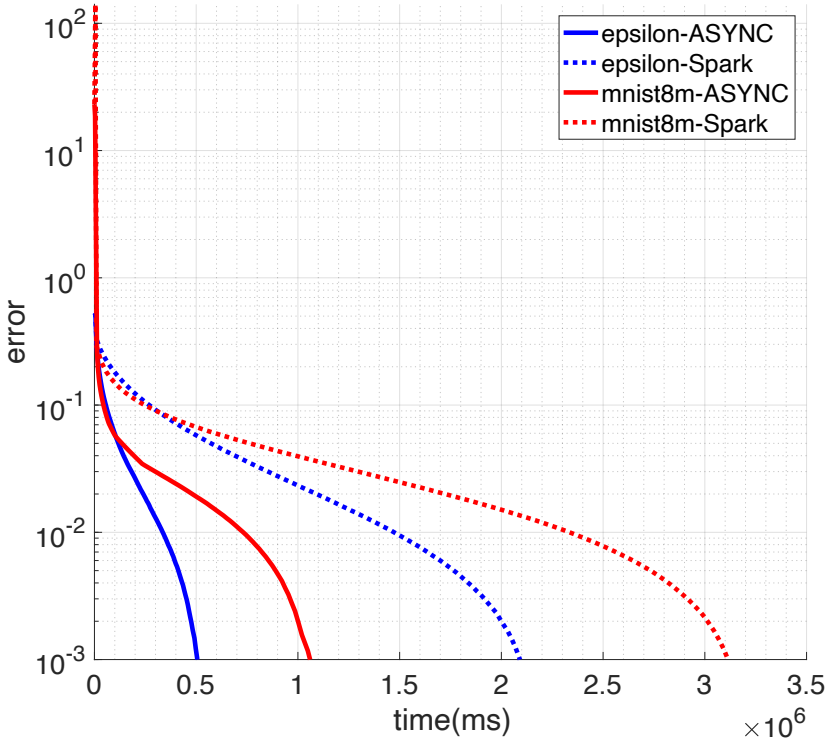
EXPERIMENTS

- We implement two optimization algorithms in ASYNC and compare to their implementations in Spark:
  - Asynchronous stochastic gradient descent (ASGD)[1]: asynchronous without history.
  - ASAGA[2]: asynchronous with history.
- Datasets are epsilon and mnist8m with 400k and 8M samples each[3].
- A cluster[4] of 32 workers with 8 straggler nodes is used.

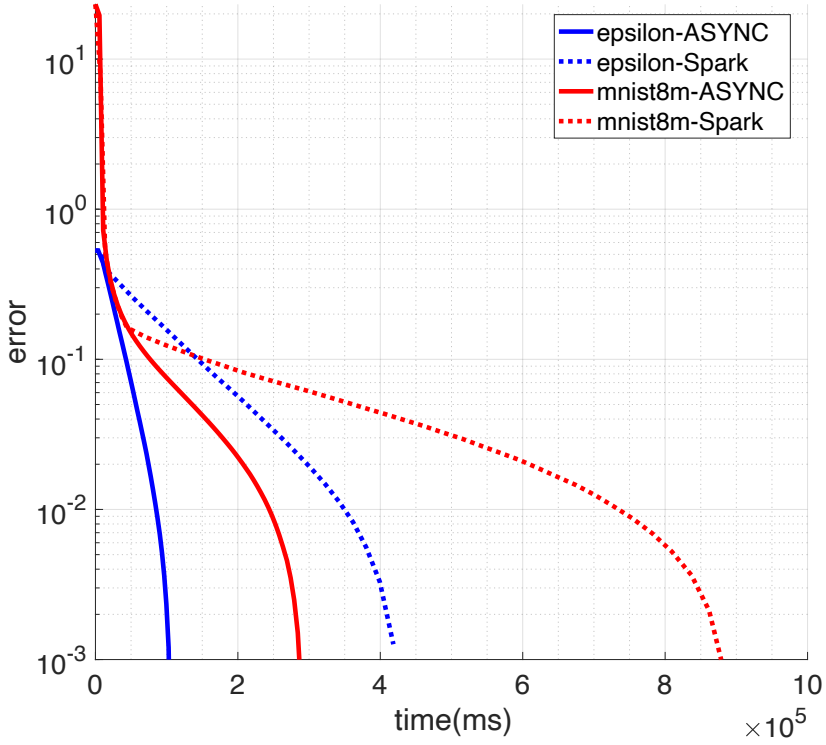
[1] L. Bottou, "Stochastic gradient descent tricks," in Neural networks: Tricks of the trade. Springer, 2012, pp. 421–436.
[2] R. Leblond, F. Pedregosa, and S. Lacoste-Julien, "Asaga: asynchronous parallel saga," arXiv preprint arXiv:1606.04809, 2016.
[3] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," ACM transactions on intelligent systems and technology (TIST), vol. 2, no. 3, p. 27, 2011.
[4] J. Towns et al., "Xsede: accelerating scientific discovery," Computing in Science & Engineering, vol. 16, no. 5, pp. 62–74, 2014.

RESULTS

ASGD algorithm



ASAGA algorithm



ASYNC is 4 times faster than Spark.

Conclusion

- We developed ASYNC, a framework to support asynchrony and history for optimization algorithms in machine learning.
- ASYNC outperforms current implementations and achieves **4x** speedup compared to Apache Spark.
- The link to ASYNC:
  - https://github.com/ASYNCframework/ASYNCframework
- The link to audio of the presentation:
  - https://bit.ly/3bh88F6
  - Email: sasoori@cs.toronto.edu