EECS 598: Paper Summary
Gaia: Geo-Distributed Machine Learning Approaching LAN Speeds

Hasan Al Muraf
hasanal@umich.edu

Fan Lai
fanlai@umich.edu

Henry Oskar Singer
hosinger@umich.edu

1 Problem and Motivation

Large-scale organizations produce a massive amount of data every day across their geo-distributed data centers (DCs). To obtain business insights and extract useful information, different machine learning (ML) techniques are being applied to this rapidly generated huge volume of data. A common operational approach is to accumulate all the data across different geo-distributed DCs into a centralized DC, and afterward, run the ML techniques locally in that DC. This centralized approach demands a huge volume of data transfer over the wide-area networks (WAN). But, as WAN bandwidth is a scarce and heterogeneous resource, this is a slower process compared to the intra-DC communication. Besides, the WAN communication is costly. For example, Amazon EC2 provides free LAN communication while adds extra charges for WAN communication. Moreover, trans-oceanic communication is prone to privacy or ownership concerns for some regions. These issues motivate to develop Gaia - a geo-distributed ML system which minimizes the communications over the WAN while guaranteeing algorithm convergence and accuracy without modifying the underline implementations.

2 Hypothesis

These days, a good number of frameworks have been proposed (i.e. TensorFlow, Petuum, Adam, PowerGraph, MLlib etc.) to perform distributed ML. Although these frameworks show significant performance improvement within a datacenter, their performance degrades when they operate across multiple datacenters. On the other hand, existing geo-distributed systems mainly focus on data analytics. As ML algorithms need back and forth communication to propagate the updates among the parameters, in geo-distributed case, extensive communication over the WAN will cost a lot. Currently available geo-distributed systems are not efficient enough for ML computations. In this work, it is found that the vast majority of the parameter updates are insignificant towards the global convergence of the ML model. So, they hypothesized that postponing the parameter update broadcast till they become significant enough will not hurt the accuracy of the whole model.

3 Solution Overview

Gaia is built on top of the parameter server architecture. Here, each DC has some worker machines and parameter servers. Every worker machine works in parallel on different shards of the input data available in that DC. The parameter servers in each DC collectively maintain a version of the global model copy. Each parameter server handles a shard of this global model copy. A worker machine only READs and UPDATEs the global model copy in its DC.

Instead of using existing communication patterns that can overwhelm the scarce WAN, Gaia decouples the synchronization within a DC (LANs) from the synchronization across different DCs (WANs). The worker machines and parameter servers within a DC use the existing Bulk Synchronization Parallel(BSP) or Stale Synchronization Parallel(SSP) models to have a faster synchronization. To reduce the communication overhead over WANs, they introduced a novel synchronization model, called Approximate Synchronous Parallel (ASP). ASP syncs the parameter servers across multiple DCs trading-off the model accuracy with lower WAN bandwidth consumption. The idea of ASP came from the observation that more than 95% of the updates may produce less than a 1% change to the parameter value. With ASP, these insignificant updates to the same parameter within a DC are aggregated. They are not synced with the parameter servers of other DCs until the aggregated updates are significant enough. ASP has the following three component.

3.1 The Significance Filter

ASP allows the programmer to specify the significance function and the threshold to determine the significance of updates for each ML algorithm. The simplicity of the significance function can vary from the measurement of the update's magnitude relative to the current value to some other non-linear sophisticated function of multiple parameter's updates. ASP maintains two significance thresholds and dynamically tunes them over the time. The first one, the hard significance threshold, guarantees ML algorithm convergence, while the second one, the soft significance threshold, controls the usage of the underutilized WAN bandwidth to speed up the convergence.

3.2 ASP Selective Barrier

When a parameter server receives the updates from it's local worker machines at a rate that is higher than the WAN bandwidth can support, instead of sending updates, it first sends a short control message to other DCs. The receiver of this ASP selective barrier message blocks its local workers from reading the specified parameters until it receives the significant updates from the sender of the barrier.

3.3 Mirror Clock

Mirror Clock provides a final safety net implementing SSP across DCs. When each parameter server receives all the updates from its local worker machines at the end of a clock (e.g., an iteration), it broadcasts its clock to the parameters servers that are in charge of the same parameters in the other DCs. When a server detects its clock is ahead of the slowest one, it blocks its local worker machines from reading its parameters until the slowest mirror server catches up.

To make the system scalable for a large number of DCs, Gaia groups DCs based on their geographical location and assigns a hub DC for each group to accumulate all the significant updates generated in the DCs within that group. The hub DC the broadcasts the updates to other hub DCs. When a hub DC gets a update from another group, it broadcasts it to the other DCs of its group.

4 Limitations and Possible Improvements

- The paper did not discuss much about the fault tolerance. Especially, during the ASP Selective Barrier and Mirror Clock mechanism, if the sender machine or the WAN network fails, then, the whole system may halt due to these barriers. This can be solved by attaching a time-to-live attribute with each barrier and mirror clock message.

- The paper focused on the problem imposed by limited WAN bandwidth rather than WAN latency. The author did not discuss Gaia's performance in the presence of high bandwidth contention or WAN latency. Besides, the size of the update massages sent by the parameter servers across the WAN should have some impact on the system's performance. In the paper, there is no such performance evaluation.

- The authors mentioned that Gaia permits users to provide a advanced significance function. But, it is not clear how the system allows the scope of adaptive or stateful significance function. The impact of different complex significance function on the convergence of ML models can also be an interesting thing to observe.

5 Related Literature

For the matrix factorization and image classification tasks, [Hsieh et al.,] employ stochastic gradient descent (SGD) to estimate model parameters. They employ Gibbs sampling to estimate topic model parameters. There are empirically successful asynchronous variants of both of these algorithms. Additionally, [Hoffman et al., 2013] introduce a method for estimating topic model parameters using something similar to SGD, and there is much successful follow-up work on this method. Thus, it seems reasonable for geo-distributed machine learning systems to restrict consideration to SGD and similar estimation algorithms.

[De Sa et al., 2015] provide favorable empirical results and theoretical convergence rates for asynchronous SGD for matrix completion problems. With respect to the machine learning model, theoretical results require standard and realistic continuity assumptions on the objective function and the stochastic gradients. With respect to the hardware, the assumptions are much stronger and cannot be expected in the GDA setting. For example, the authors assume a single, central copy of the parameters that is read and updated asynchronously by each of multiple threads. [De Sa et al., 2016] make similarly realistic model assumptions and similarly ill-fitting (for GDA) hardware assumptions for an asynchronous Gibbs sampling algorithm inspired by the algorithms analyzed in [De Sa et al., 2015].

[McMahan and Streeter, 2014] makes much stronger assumptions about the objective function and gradients, but much weaker assumptions on the hardware. The assumptions made on the gradients and objective function are standard in this type of theoretical analysis but tend not to be necessary for empirical success. The algorithm discussed by the authors is explicitly intended for the geo-distributed setting.

[Duchi et al., 2015] makes similar hardware assumptions to [De Sa et al., 2015, De Sa et al., 2016] with looser restrictions on thread behavior and similar model assumptions to [McMahan and Streeter, 2014], then favorable reaches theoretical convergence results.

It seems like basic attempts to extend the algorithms discussed in these works to GDA should have been made before completely dismissing the asynchronous option. If that was not feasible, there should have at least been more in-depth discussion in the paper.

6 Summary of Class Discussion

- The presenters expressed skepticism about the estimated monetary cost of WAN usage. [Hsieh et al.,] cite the AWS pricing page but do not detail how they calculated their estimates from the information on this page.

- The presenters were concerned about the paper's focus on size of change to the parameter estimates rather than size of effect on model performance. A student suggested that, although the authors should have considered the difference between the filtering criterion and the final objective of the optimization routine (maximizing model performance), given the continuity of model performance with respect to parameters, this is probably not a huge issue.

- A student asked whether the significance filter introduces an additional knob for users to tune, and the presenters confirmed.

- A student asked whether it is possible to introduce a stateful significance filter. Another student asked if the first student was referring to an adaptive update scheme and commented that this might result in biased parameter estimates. The presenters explained that the authors used a time-varying significance filter with an initial threshold γ and threshold $\gamma^{(t)} \leftarrow \frac{\gamma}{\sqrt{t}}$ at time t. The summary author notes that, as $\gamma^{(0)}, \gamma^{(1)}, \gamma^{(2)}, \ldots$ is a common choice of step size sequence in first order optimization settings, this threshold schedule makes intuitive sense.

- [Hsieh et al.,] did not present experiments for Gaia within a LAN. The presenters suspect this is due to poor within-LAN performance of Gaia compared to alternative systems.

- The presenters suggested that attempting to compare Gaia with existing machine learning frameworks is meaningless to some extent due to the fact that many of these systems can be run as Gaia's local, within-data-center subroutines. The instructor pointed out that Gaia differs from other GDA frameworks in that it takes advantage of the inherently lossy nature of machine learning, whereas other GDA frameworks are focused on tasks that necessitate lossless results.

- The instructor noted that Gaia does not discuss the variation in size of the messages and pointed out that smaller messages would make minimizing rount-trip-time more important.

References

[De Sa et al., 2016] De Sa, C., Olukotun, K., and Ré, C. (2016). Ensuring rapid mixing and low bias for asynchronous gibbs sampling. In *JMLR workshop and conference proceedings*, volume 48, page 1567. NIH Public Access.

[De Sa et al., 2015] De Sa, C. M., Zhang, C., Olukotun, K., and Ré, C. (2015). Taming the wild: A unified analysis of hogwild-style algorithms. In *Advances in neural information processing systems*, pages 2674–2682.

[Duchi et al., 2015] Duchi, J. C., Chaturapruek, S., and Ré, C. (2015). Asynchronous stochastic convex optimization. *arXiv preprint arXiv:1508.00882*.

[Hoffman et al., 2013] Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

[Hsieh et al.,] Hsieh, K., Harlap, A., Vijaykumar, N., Konomis, D., Ganger, G., Gibbons, P., and Mutlu, O. Gaia:geo-distributed machine learning approaching lan speeds. in NSDI '17.

[McMahan and Streeter, 2014] McMahan, B. and Streeter, M. (2014). Delay-tolerant algorithms for asynchronous distributed online learning. In *Advances in Neural Information Processing Systems*, pages 2915–2923.