Building Highly-Available Geo-Distributed Datastores for Continuous Learning

Nitin Agrawal
Samsung Research

Ashish Vulimiri
Samsung Research

1 Extended Abstract

Data generation is becoming increasingly decentralized. As a consequence, techniques for building and continuously updating machine learning models, typically confined to a data center, need to evolve to encompass geo-distributed data sources if they are to remain relevant and cost-effective. Timely access to distributed data for the purposes of analytics and machine learning is extremely valuable but challenging to provide [25, 17, 29, 20]. Scaling up these data and compute intensive tasks mandates distribution [20]. However, key algorithms for learning, such as stochastic gradient descent, are not trivially parallelizable; distributed implementations often require carefully re-architecting the underlying computational model [11]. For these analyses, if the global view of data can be maintained at a single node, extracting parallelism can become significantly simpler.

Data decentralization is pushed further with edge computing. Applications that span edge devices such as smartphones, home appliances, and sensors, exchange significant amounts of data with cloud-based services. Autonomous vehicles alone are expected to generate about a Gigabyte of data every second [28]; edge services can include real-time driving decisions, monitoring and guidance, and fleet-wide situational awareness [30]. These applications require the edge to continuously consolidate data from a large number of clients and maintain uptodate copies in multiple sites, often geographically distant, for both availability and performance. The cost of data movement over wide-area networks is thus a significant hurdle in learning and a driving motivation behind systems specifically optimized for geo-distributed machine learning [17, 5].

Techniques to accommodate geo-distributed learning largely follow two approaches: 1) a centralized approach wherein data from remote locations is migrated, continuously if needed, to a single data center which relies on *conventional* machine learning approaches. 2) a decentralized approach wherein the machine learning computation itself is spread across the remote sites; the different sites communicate with each other to exchange intermediate state and parameters.

The centralized learning approach has the benefit of leveraging existing algorithms without the need to create custom distributed, or geo-distributed, variants. However, centralizing data can be prohibitively expensive, as has been noted by others [17]. The decentralized approach can provide speedup through parallel execution, but requires sophisticated systems and algorithmic enhancements to account for and mitigate the significant communication overhead incurred during the learning tasks. A recent system, Gaia [17], leverages weaker synchronization semantics to better utilize WAN bandwidth; Gaia proposes a new synchronization model, Approximate Synchronous Parallel (ASP), which maintains an approximately-correct copy of the global ML model within each data center thereby reducing the extent of data transfers. STRADS [19] is designed to extract model parallelism in order to accelerate convergence of ML algorithms and improve the learning performance. A model-parallel algorithm aims to update a subset of parameters on each site — often using the entirety of available data — while ensuring overall correctness; a model-parallel approach does not address the issue of the cost of data movement.

Motivated by the observation that for geo-distributed learning, both centralized and decentralized approaches have associated weaknesses, we pose the following question: *can a system provide the convenience of centralized approaches (i.e., little to no change in the semantics of the ML compu-*

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

tation) and the benefit of decentralization (i.e., ability to scale out for larger models and improve performance) at the same time? Since a crucial factor in relieving this tradeoff is the inherent cost of migrating and replicating data over wide-area networks, we believe it is worthwhile revisiting the assumptions behind the underlying distributed storage systems that provide the data access.

Replicating data in distributed systems, especially over wide-area networks, is typically both slow and expensive [4, 23, 24, 27, 2]; tremendous growth in data [28, 18, 26, 21, 8, 9] without commensurate growth in network capacity has only exacerbated the problem [31, 16, 3]. Timely synchronization of distributed replicas under massive data-ingest rates strains the underlying network, further burdening operational costs [13]. Weaker distributed consistency models relax the timeliness requirement to gain efficiency and availability but do not counter the fundamental drawbacks. Even with weakened semantics, conventional replication provides *exactness*, guaranteeing the eventual availability of *all* data; this is overkill for this growing class of applications – in analytics and machine learning – which requires frequent re-computation on continuously-evolving data.

A geo-distributed datastore that provides fast and cost-effective access to data generated across widely-distributed nodes can substantively impact the quality and programmability of continuous machine learning applications. Several open-source and commercial weakly-consistent stores are highly scalable and well-suited to handle large volumes of data. Commercial NoSQL stores,*e.g.*, Dynamo [12], BigTable [7], and their open-source implementations, *e.g.*, Cassandra [6], HBase [15], are just some of the examples. Many of these are publicly-available as cloud services, and accessible to application developers, or widely deployed in-house; for instance, the largest deployments of Cassandra at Apple, and Netflix, span over 75,000 nodes storing over 10 PB of data, and 2500 nodes storing 420 TB, respectively [6]. MongoDB is a causally-consistent document store that can scale to 100s of nodes and ∼Billion documents [22]. Spanner [10] and Mesa [14] are geo-replicated stores primarily aimed at business-critical applications that require strong consistency at scale. CosmosDB [1] is another geo-distributed database with tunable consistency and horizontal scalability.

However, a differentiation for continuous geo-distributed machine learning, as discussed earlier, is the requirement for near-instant availability of remote data; the "exact" stores are designed to scale to large number of nodes and large volumes of data, but are inefficient when it comes to high-velocity ingest. To overcome the challenges in geo-distributed machine learning services, we propose the abstraction of an *inexact replica* for distributed data management. Inexact replicas deliberately, and controllably, reduce the precision of data that is sent over the network to allow replicas be inexact copies of each other. The key insight being that replication can turn from slow and expensive to fast and efficient by embracing data approximation, leading to the cost-effective construction of highly-available services. The primary challenge is to ensure that the replicas are indeed useful to performing the computational tasks that they set out to. The system not only needs to control the nature and extent of the imprecision but, more importantly, ensure that the replicas are actually interchangeable for the purposes of the ML tasks; for example, a model-parallel algorithm can run on any replica without noticeable relative degradation in the output quality. The degree of inexactness, or replica skew, is carefully controlled via a global coordination process to ensure that replicas do not diverge beyond a (small) pre-specified threshold.

As part of this effort, we are building a geo-distributed datastore specifically designed for analytics and machine learning applications based on the inexact replica abstraction, providing data replication at a fraction of the cost of conventional storage systems. The resulting store has a decisively small time lag between data ingest and completion of replication even over wide-area nodes ingesting in the order of Terabytes of data per day, and reduced wide-area traffic by orders of magnitude.

With our store, centralized machine learning algorithms can operate, near instantly, on the entire global view of data without modification; the storage system ensures that the data replication costs are substantially reduced. Distributed algorithms relying on model parallelism can now expect different nodes to operate on all data, leading to more efficient parallel execution. Initial evaluation of our system on a geo-distributed setup shows promise in reducing wide-area traffic while maintaining low degrees of replica divergence and high overall accuracy for a forecasting workload. We believe this opens potential avenues for research exploration in both systems for machine learning and co-designing distributed learning algorithms in light of the improvements in data access. We aim to leverage the participation of both systems and machine learning practitioners to further this discussion and solicit feedback for future work.

References

[1] Tunable data consistency levels in Azure Cosmos DB. https://docs.microsoft.com/en-us/azure/cosmos-db/consistency-levels, 2018.

[2] Amazon. Amazon ec2 pricing: Data transfer. https://aws.amazon.com/ec2/pricing/on-demand/, 2018.

[3] Amazon. AWS Snowball Edge: Petabyte-scale data transport with on-board storage and compute capabilities. https://aws.amazon.com/snowball-edge/, 2018.

[4] E. A. Brewer. Lessons from Giant-Scale Services. *IEEE Internet Computing*, 5(4):46–55, July 2001.

[5] I. Cano, M. Weimer, D. Mahajan, C. Curino, and G. M. Fumarola. Towards geo-distributed machine learning. *arXiv preprint arXiv:1603.09035*, 2016.

[6] Apache Cassandra Database (Landing Page). http://cassandra.apache.org.

[7] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2):1–26, 2008.

[8] Cisco. Cisco White Paper: The Zettabyte Era: Trends and Analysis. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html, 2013.

[9] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021 White Paper. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, 2016.

[10] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, et al. Spanner: Googles globally distributed database. *ACM Transactions on Computer Systems (TOCS)*, 31(3):8, 2013.

[11] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al. Large scale distributed deep networks. In *Advances in neural information processing systems*, pages 1223–1231, 2012.

[12] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's Highly Available Key-value Store. In *SOSP '07: Proceedings of Twenty-First ACM SIGOPS Symposium on Operating Systems Principles*, pages 205–220, New York, NY, USA, 2007. ACM.

[13] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: research problems in data center networks. *ACM SIGCOMM computer communication review*, 39(1):68–73, 2008.

[14] A. Gupta, F. Yang, J. Govig, A. Kirsch, K. Chan, K. Lai, S. Wu, S. Dhoot, A. Kumar, A. Agiwal, S. Bhansali, M. Hong, J. Cameron, M. Siddiqi, D. Jones, J. Shute, A. Gubarev, S. Venkataraman, and D. Agrawal. Mesa: Geo-replicated, near real-time, scalable data warehousing. In *VLDB*, 2014.

[15] Apache HBase. hbase.apache.org.

[16] K. Hill. NetApp Blog: How to Speed Up Massive Data Migration to Amazon S3. https://blog.netapp.com/how-to-speed-up-massive-data-migration-to-amazon-s3/, 2018.

[17] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G. R. Ganger, P. B. Gibbons, and O. Mutlu. Gaia: Geo-distributed machine learning approaching lan speeds. In *NSDI*, pages 629–647, 2017.

[18] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez. Inter-datacenter bulk transfers with netstitcher. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pages 74–85, New York, NY, USA, 2011. ACM.

[19] S. Lee, J. K. Kim, X. Zheng, Q. Ho, G. A. Gibson, and E. P. Xing. On model parallelization and scheduling strategies for distributed machine learning. In *Advances in neural information processing systems*, pages 2834–2842, 2014.

[20] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su. Scaling distributed machine learning with the parameter server. In *OSDI*, volume 14, pages 583–598, 2014.

[21] Mike Keane. 1.5 million Log Lines per Second. http://www.bigdataeverywhere.com/files/chicago/BDE-15millionLogLinesPerSecond-KEANE.pdf, 2014.

[22] MongoDB at scale. https://www.mongodb.com/mongodb-scale.

[23] S. Muralidhar, W. Lloyd, S. Roy, C. Hill, E. Lin, W. Liu, S. Pan, S. Shankar, V. Sivakumar, L. Tang, et al. f4: Facebook's warm blob storage system. In *Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation*, pages 383–398. USENIX Association, 2014.

[24] A. Muthitacharoen, B. Chen, and D. Mazières. A Low-Bandwidth Network File System. In *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, pages 174–187, Lake Louise, Alberta, Oct. 2001.

[25] Q. Pu, G. Ananthanarayanan, P. Bodik, S. Kandula, A. Akella, P. Bahl, and I. Stoica. Low latency geo-distributed data analytics. *SIGCOMM Comput. Commun. Rev.*, 45(4):421–434, Aug. 2015.

[26] Ted Friedman. Gartner Report: "Internet of Things: Biggest Impact Ever on Information and Master Data" . http://www.gartner.com/webinar/3291728?srcId=1-7389946120, 2016.

[27] D. B. Terry. *Replicated Data Management for Mobile Computing*, volume 5. Morgan & Claypool Publishers, 2008.

[28] M. van Rijmenam. Self-driving cars will create 2 petabytes of data, what are the big data opportunities for the car industry? https://datafloq.com/read/self-driving-cars-create-2-petabytes-data-annually/172, 2017.

[29] A. Vulimiri, C. Curino, B. Godfrey, K. Karanasos, and G. Varghese. Wanalytics: Analytics for a geo-distributed data-intensive world. In *CIDR*, 2015.

[30] Zenlayer. How Autonomous Cars Helped Pave The Way For Edge Computing. https://www.zenlayer.com/autonomous-cars-helped-pave-way-edge-computing/.

[31] B. Zhang, X. Jin, S. Ratnasamy, J. Wawrzynek, and E. A. Lee. Awstream: adaptive wide-area streaming analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 236–252. ACM, 2018.