The Challenges of Global-scale Data Management

Faisal Nawab Divyakant Agrawal Amr El Abbadi

Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106
{nawab,agrawal,amr}@cs.ucsb.edu

ABSTRACT

Global-scale data management (GSDM) empowers systems by providing higher levels of fault-tolerance, read availability, and efficiency in utilizing cloud resources. This has led to the emergence of global-scale data management and event processing. However, the Wide-Area Network (WAN) latency separating data is orders of magnitude larger than conventional network latencies, and this requires a reevaluation of many of the traditional design trade-offs of data management systems. Therefore, data management problems must be revisited to account for the new design space. In this tutorial we survey recent developments in GSDM focusing on identifying fundamental challenges and advancements in addition to open research opportunities.

1. INTRODUCTION

Internet applications strive for high-performance 24/7 service to users dispersed around the world. Achieving this is threatened by complete datacenter outages and the physical limitations of both the datacenter infrastructure and wide-area communication. To overcome these challenges, systems are increasingly being deployed in multiple datacenters spanning large geographic regions. The replication of data across datacenters (geo-replication) allows requests to be served even in the event of complete datacenter-scale outages. Likewise, distributing the processing and storage across datacenters brings the application closer to users and sources of data, enabling higher levels of availability and performance. Additionally, extremely large applications consume huge amounts of resources. These resources vary and include computing infrastructure in addition to power and real-estate. Globally distributing the computing infrastructure of large applications allows them to utilize resources beyond the restrictions of a single datacenter or cloud provider.

Moving to global-scale data management (GSDM), despite its benefits, raises many novel challenges that are not faced by traditional deployments. The large WAN communication latency is orders of magnitude larger than traditional communication latency (See Figure 1). This invalidates the traditional space of design trade-offs and makes the WAN latency a significant bottleneck. Likewise, WAN bandwidth (∼100 Tbps [3]) is larger than traditional networks bandwidth. However, big data applications transfer large volumes of data reported to be in the order of hundreds of TBs per day and projected to be increasing in the future [62]. These increasing demands will lead to a bottleneck in WAN communication. The design of efficient mechanisms that better utilize the WAN links are necessary to avoid limiting the growth of global-scale big data applications. Also, due to privacy legislation concerns in some parts of the world, there is a direction to limit control on where data is placed and at what level of privacy it is stored [1, 2]. These restrictions affect the design of global-scale data placement and task scheduling.

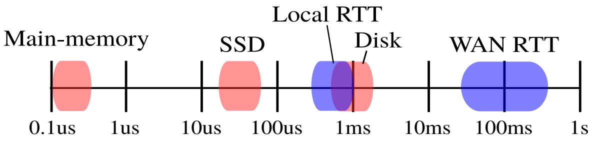

Figure 1: Latency of the Wide-Area Network Round-Trip Time communication (WAN RTT) compared to memory access latency [50] and network latency within the datacenter (local RTT)

Building GSDM systems requires a rethinking of data management problems in light of the new design trade-offs and constraints. In this tutorial we survey recent work in building GSDM systems. We focus on exposing the salient features of GSDM systems that make them different, and also identify the short- and long-term research opportunities in this space.

2. TUTORIAL INFORMATION

The target audience are database researchers and practitioners with basic knowledge of transactions. Some knowledge of replication, stream processing, or analytics is helpful but not necessary. For newcomers, the tutorial introduces GSDM as an emerging framework with unique research challenges and novel system designs. For researchers experienced in GSDM, the tutorial provides a broader view of GSDM research beyond the work within the databases community. Generally, we aim to enable data researchers to extend their work and expertise to a global-scale framework.

We present the challenges and design principles of GSDM. The development of GSDM research is presented in chronological order, emphasizing the trends that led to the current state-of-the-art. We then extend these trends to project the upcoming challenges and opportunities of GSDM. To provide more depth to the presented topics, for each part we highlight and discuss in more details one or two representative systems. Each work that we discuss in detail is underlined in this proposal.

The tutorial is divided into three sections: (1) The system model and unique characteristics of GSDM (Section 1), (2) Data access technology (Section 3.1) that includes principles and system de-

SIGMOD'16, June 26-July 01, 2016, San Francisco, CA, USA
© 2016 ACM. ISBN 978-1-4503-3531-7/16/06. . . $15.00
DOI: http://dx.doi.org/10.1145/2882903.2912571

signs of data access to geo-distributed storage, and (3) Event processing (section 3.2) that includes global-scale stream processing and analytics. The target length of the tutorial is 3 hours.

3. TUTORIAL OUTLINE

3.1 Data access

Data access is the ability to read and modify data objects typically via put and get operations or groups of operations forming a transaction. This section surveys literature on methods to access globally-distributed data. Globally-distributed data is either geo-replicated where data objects are replicated in multiple datacenters, geo-distributed where different partitions of the data exist in different datacenters, or a combination of both.

The central problem tackled by data access of globally-distributed data is managing the consistency-performance trade-off that is amplified due to large wide-area latency. To guarantee consistent outcomes, coordination between distributed components is necessary. This coordination is expensive due to WAN links, and thus affects performance immensely. The first reaction to the new trade-off was the abandonment of strong notions of consistency in favor of performance [10,16,19]. Systems that adopt this approach provide weak guarantees like eventual consistency and single-key atomicity. These guarantees were sufficient for many applications, however, it turns out that many use-cases require stronger notions of consistency. In addition, developing applications on top of weakly consistent data is error-prone and less natural to developers.

The need for stronger forms of consistency on globally-distributed data sparked a trend to explore different points in the consistency-performance trade-off spectrum. The goal is to explore existing consistency notions, that are weaker than expensive strong consistency, and extend them in ways suitable for globally-distributed data. Causal consistency, inspired by the causal ordering principle [35], preserves causal relations between operations. Causal consistency is attractive for GSDM because it does not require coordination between replicas. Thus, causally consistent systems that are built for globally-distributed data, do not suffer significantly from the cost of wide-area latency [11,20,40,41,44]. For example, COPS and Eiger [40,41] are scalable causally consistent systems that offer single-key operations in addition to read-only or write-only transactional access. COPS extends the notion of causal consistency and proposes Causal+ consistency that, in addition to causality, guarantees data convergence.

Snapshot Isolation (SI) [13] guarantees that transactions always read a consistent snapshot and ensures the absence of write-write conflicts. SI can lead to better performance compared to stronger notions of consistency like serializability specially for read-intensive workloads [13]. Many solutions leverage SI for GSDM [18,21,38,39,57]. Walter [57] extends the notion of SI and proposes Parallel SI (PSI). PSI redefines snapshot reads and write-write conflicts to accommodate the new environment of globally-distributed data. With PSI, Walter is able to replicate data asynchronously while still providing strong guarantees within each site. In the original SI, asynchronous replication is not possible.

Strong consistency guarantees, such as serializability [14] and linearizability [27], require extensive coordination between globally-distributed data. Despite their performance implications, industry and academia have recently shown that strong guarantees are needed for a wide range of applications. This has started a movement towards preserving strong consistency guarantees while trying to reduce the cost of coordination on performance. Paxos [36] is a fault-tolerant consensus protocol that has been used to implement GSDM systems that provide strong guarantees, such as megastore [12] and Paxos-CP [48]. Later though, it was shown to perform better as a synchronous communication layer integrated with a transaction commit protocol [17,23,42]. For example, Spanner [17] and Replicated Commit [42] commit transactions using variants of Two-Phase Commit (2PC) and Strict Two-Phase Locking (S2PL) and leverage Paxos to replicate across datacenters. Paxos, however, requires two rounds of communication to commit, which is a cost amplified by WAN latency. This led to efforts to optimize the cost of Paxos by the use of leases to eliminate the cost of the first round of communication and by adopting Fast Paxos [37] that allows committing with a single round to a super majority [32,47].

The existing 2PC and Paxos protocols require one round of communication at least to commit a transaction. Nawab et. al. explored breaking the round-trip barrier by decoupling consistency from fault-tolerance [43,45]. Message Futures [43] reserves commit points for future transactions to achieve low commit latency. Commit points are represented as logical timestamps in shared replicated logs. Because transactions can use previously assigned commit points, they can commit in less than a round-trip time. Helios [45] detects conflicts using timestamp-based conflict detection. For each transaction, it calculates time ranges at which conflicts might occur. The transaction commits once it verifies that there are no conflicting transactions in these time ranges. The calculation of time ranges leverages a theoretical result on lower-bound transaction latency that leads to close to optimal transaction latency.

The use of timestamps and time synchronization has been explored for GSDM systems [17,20,21,45]. Spanner [17] provides external consistency guarantees by leveraging accurate time synchronization using specialized infrastructure such as atomic clocks. Without accurate time synchronization, other systems resort to loosely-synchronized clocks methods [20,21,45].

To commit arbitrary transactions with strong consistency guarantees, the cost of coordination is inevitable [45]. However, coordination-free execution is possible for some types of transactions [9, 52, 66]. This is possible by inferring application-level invariants of transactions correctness and then exploring whether a coordination-free execution is permissible. Transaction Chains [66] derives an execution plan of distributed transactions that allows for a fast response time. A client needs to wait for the execution of the transaction at only a single site, rather than waiting for the execution at all accessed sites. Often, the first site is local to the client, making the response latency unaffected by the WAN latency.

Placement of data and workers has a significant effect on GSDM systems performance [4, 22, 49, 53, 54, 62, 63, 65]. In this part of the tutorial we discuss how placement plays a role in general data access systems. SPANStore [63] proposes an optimization formulation of placement to minimize monetary cost. SPANStore's formulation considers constraints on fault-tolerance, consistency requirements, and performance SLOs. Sharov et al. [54] propose an optimization formulation for placement with an objective of minimizing latency. What distinguishes this work is that it considers transactional access to storage.

Global-scale placement depends on the workload, which includes the location of users and the type of requests being issued. A successful placement needs to correctly estimate workload and adapt to workload changes. A correct estimation of the workload has to be done using an accurate system model and using efficient log and trace collection and analysis. Adapting to workload changes is more challenging, since the new workload characteristics potentially lead to choosing a significantly different configuration. SPANStore changes configurations in an epoch-based fashion. Violations might occur while changing configurations, but are rare since they only occur at epoch boundaries.

The dynamic and variable nature of workload and communication links in GSDM led to work on dynamic techniques for GSDM systems [7, 58, 64]. Variability of the communication link leads to the reordering and delaying of sent messages. CosTLO [64] proposes adding redundancy in messaging to lower the latency variance. GSDM systems trade-off consistency and performance and could be locked in their initial protocol choice. Pileus and Tuba [7, 58] allow applications to dynamically control the consistency-performance trade-off by declaring their consistency and latency priorities. The application's priorities then influence the decision on which servers to access. Stronger consistency requires accessing more servers and thus increases latency. In turn, more relaxed consistency requires accessing lesser number of servers and thus decreases latency.

3.2 Event processing

Global-scale applications receive and generate large volumes of data across datacenters. It is reported that the amount of data processed by large web applications is in the order of 10s to 100s of TBs per day [62]. This has led to the design of many GSDM systems for stream processing and data pipelining from Google [5, 6, 24, 55, 56], Twitter [33, 60], Facebook [59], and LinkedIn [8]. These systems are designed to support the high volume and velocity of events while conserving availability and high performance. In addition, they guarantee correctness of computations that use the streamed data. Correctness invariants vary depending on the application, but the following correctness conditions are common: (1) *At-most-once semantics*, which means that no event is processed more than once, and (2) *Near-exact semantics*, which means that with no significant delay, all events will be processed. These correctness conditions, albeit simple, are challenging on the scale of global web applications where streams might be significantly reordered and delayed. Photon [6] is a system deployed at Google to join global-scale continuous streams. To tolerate failures, an event can be processed at any of the operating datacenters. To guarantee at-most-once semantics, before a joiner starts processing an event it ensures that the event has not been processed before. It does so by using a logically centralized Paxos process for the event's unique ID.

In addition to processing streams and managing data pipelines, GSDM systems often perform analytical and machine learning queries on data to extract insight and business knowledge. Global-scale analytics — as opposed to traditional analytics within a datacenter — face novel challenges. Until recently, global-scale analytics were performed by pulling all data to a central location [61]. This, however, consumes the WAN links, which causes monetary loss and poses a physical constraint on throughput. Additionally, data movement could be constrained due to emerging data sovereignty legislation and privacy concerns. Recent solutions address the challenges of global-scale analytics [15, 25, 26, 28, 30, 31, 46, 49, 51, 61, 62]. Geode [62] and PIXIDA [31] are two systems that propose a query planning and replication framework that targets reducing the bandwidth cost between datacenters. Additionally, Geode uses optimizations such as aggressive caching and measurement collection that allow for the reuse of past queries. Unlike Geode and PIXIDA, Iridium [49] aims to minimize the latency of global analytics by deriving the placement of both data and tasks. Geode, PIXIDA, and Iridium derive their solutions using an optimization formulation. They all face a common problem of the intractability of optimization solvers with 10s of datacenters. To overcome this, Geode and Iridium leverage a greedy heuristic and PIXIDA proposes a flow-based approximation algorithm.

4. OUTLOOK

Modern web applications require higher levels of fault-tolerance and availability. Also, they are increasingly consuming larger volumes of data. Globally deploying web applications is a necessary step towards fulfilling these requirements. However, GSDM has unique characteristics that change the traditional space of design trade-offs. Most notably are the WAN link characteristics. The cost of coordination has been amplified due to WAN latency, and the WAN bandwidth, albeit large, is limited as it serves a large number of data-intensive applications. Additionally, regulatory constraints may limit the control of data management systems.

Leveraging advances in WAN research from the networking community is an important step towards building efficient GSDM systems. Networking techniques, like Software-defined Networking (SDN), are now being applied to the context of WANs (*e.g.*, BwE [34] and B4 [29]). A promising opportunity is to develop GSDM systems that integrate these advances. Also, adopting privacy-preserving techniques for GSDM is an important endeavour to address the geo-political regulatory concerns on data usage. However, current privacy-preserving protocols are communication-intensive and rely on redundancy that cause over-utilization of network bandwidth. As such, this makes these protocols especially inappropriate for GSDM. We expect a proliferation of studies on the trade-off between privacy and performance at the global scale.

The unique challenges of GSDM introduce inefficiencies to existing designs across various data management problems. The opportunity is for researchers to explore the implications of transforming their work from a local traditional computing framework to the global scale.

5. BIOGRAPHICAL SKETCHES

Faisal Nawab is a doctoral student at the University of California at Santa Barbara. His current research work is in the areas of global-scale data management, big data analytics, and data management on emerging non-volatile memory technology.

Divyakant Agrawal is a Professor of Computer Science at the University of California at Santa Barbara. His current interests are in the area of scalable data management and data analysis in Cloud Computing environments, security and privacy of data in the cloud, and scalable analytics over big data. Prof. Agrawal is an ACM Distinguished Scientist (2010), an ACM Fellow (2012), and an IEEE Fellow (2012).

Amr El Abbadi is a Professor of Computer Science at the University of California, Santa Barbara. Prof. El Abbadi is an ACM Fellow, AAAS Fellow, and IEEE Fellow. He was Chair of the Computer Science Department at UCSB from 2007 to 2011. He has served as a journal editor for several database journals and has been Program Chair for multiple database and distributed systems conferences. Most recently Prof. El Abbadi was the co-recipient of the Test of Time Award at EDBT/ICDT 2015. He has published over 300 articles in databases and distributed systems and has supervised over 30 PhD students.

6. ACKNOWLEDGMENT

This work is supported by NSF grants IIS 1018637, 1528178, and 1442966 and is partially funded by a gift grant from NEC Labs America.

7. REFERENCES

[1] Amazon web services. whitepaper on eu data protection. https://d0.awsstatic.com/whitepapers/compliance/AWS_EU_Data_Protection_Whitepaper.pdf. 2015.

[2] European commission press release. commission to pursue role as honest broker in future global negotiations on internet governance. http://europa.eu/rapid/press-release_IP-14-142_en.htm. 2014.

[3] Global internet geography. https://www.telegeography.com/research-services/global-internet-geography/. 2015.

[4] Agarwal, S., Dunagan, J., Jain, N., Saroiu, S., Wolman, A., and Bhogan, H. Volley: Automated data placement for geo-distributed cloud services. In *NSDI* (2010).

[5] Akidau, T., Balikov, A., Bekiroğlu, K., Chernyak, S., Haberman, J., Lax, R., McVeety, S., Mills, D., Nordstrom, P., and Whittle, S. Millwheel: fault-tolerant stream processing at internet scale. In *VLDB* (2013).

[6] Ananthanarayanan, R., Basker, V., Das, S., Gupta, A., Jiang, H., Qiu, T., Reznichenko, A., Ryabkov, D., Singh, M., and Venkataraman, S. Photon: Fault-tolerant and scalable joining of continuous data streams. In *SIGMOD* (2013).

[7] Ardekani, M. S., and Terry, D. B. A self-configurable geo-replicated cloud storage system. In *OSDI* (2014).

[8] Auradkar, A., Botev, C., Das, S., De Maagd, D., Feinberg, A., Ganti, P., Gao, L., Ghosh, B., Gopalakrishna, K., Harris, B., et al. Data infrastructure at linkedin. In *ICDE* (2012).

[9] Bailis, P., Fekete, A., Franklin, M. J., Ghodsi, A., Hellerstein, J. M., and Stoica, I. Coordination avoidance in database systems. In *VLDB* (2014).

[10] Bailis, P., and Ghodsi, A. Eventual consistency today: limitations, extensions, and beyond. *Communications of the ACM 56*, 5 (2013), 55–63.

[11] Bailis, P., Ghodsi, A., Hellerstein, J. M., and Stoica, I. Bolt-on causal consistency. In *SIGMOD* (2013).

[12] Baker, J., et al. Megastore: Providing scalable, highly available storage for interactive services. In *CIDR* (2011).

[13] Berenson, H., Bernstein, P., Gray, J., Melton, J., O'Neil, E., and O'Neil, P. A critique of ansi sql isolation levels. In *SIGMOD* (1995).

[14] Bernstein, P. A., Hadzilacos, V., and Goodman, N. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.

[15] Cano, I., Weimer, M., Mahajan, D., Curino, C., and Fumarola, G. Towards geo-distributed machine learning. In *Workshop on Machine Learning Systems at NIPS* (2015).

[16] Cooper, B. F., Ramakrishnan, R., Srivastava, U., Silberstein, A., Bohannon, P., Jacobsen, H.-A., Puz, N., Weaver, D., and Yerneni, R. Pnuts: Yahoo!'s hosted data serving platform. *VLDB* (2008).

[17] Corbett, J., Dean, J., Epstein, M., Fikes, A., Frost, C., Furman, J., Ghemawat, S., Gubarev, A., Heiser, C., Hochschild, P., et al. Spanner: Google's globally-distributed database. In *OSDI* (2012).

[18] Daudjee, K., and Salem, K. Lazy database replication with snapshot isolation. In *VLDB* (2006).

[19] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., and Vogels, W. Dynamo: amazon's highly available key-value store. In *ACM SIGOPS* (2007).

[20] Du, J., Elnikety, S., Roy, A., and Zwaenepoel, W. Orbe: Scalable causal consistency using dependency matrices and physical clocks. In *SoCC* (2013).

[21] Du, J., Elnikety, S., and Zwaenepoel, W. Clock-si: Snapshot isolation for partitioned data stores using loosely synchronized clocks. In *SRDS* (2013).

[22] Endo, P. T., de Almeida Palhares, A. V., Pereira, N. N., Goncalves, G. E., Sadok, D., Kelner, J., Melander, B., and Mångs, J.-E. Resource allocation for distributed cloud: concepts and research challenges. *Network, IEEE* (2011).

[23] Glendenning, L., Beschastnikh, I., Krishnamurthy, A., and Anderson, T. Scalable consistency in scatter. In *SOSP* (2011).

[24] Gupta, A., Yang, F., Govig, J., Kirsch, A., Chan, K., Lai, K., Wu, S., Dhoot, S. G., Kumar, A. R., Agiwal, A., Bhansali, S., Hong, M., Cameron, J., Siddiqi, M., Jones, D., Shute, J., Gubarev, A., Venkataraman, S., and Agrawal, D. Mesa: Geo-replicated, near real-time, scalable data warehousing. *Proc. VLDB Endow. 7*, 12 (Aug. 2014), 1259–1270.

[25] Heintz, B., Chandra, A., and Sitaraman, R. K. Optimizing grouped aggregation in geo-distributed streaming analytics. In *HPDC* (2015).

[26] Heintz, B., Chandra, A., and Sitaraman, R. K. Towards optimizing wide-area streaming analytics. In *IEEE Workshop on Cloud Analytics* (2015).

[27] Herlihy, M. P., and Wing, J. M. Linearizability: A correctness condition for concurrent objects. *ACM TOPLAS 12*, 3 (1990), 463–492.

[28] Hung, C.-C., Golubchik, L., and Yu, M. Scheduling jobs across geo-distributed datacenters. In *SoCC* (2015).

[29] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., et al. B4: Experience with a globally-deployed software defined wan. In *SIGCOMM* (2013).

[30] Jonathan, A., Chandra, A., and Weissman, J. Awan: Locality-aware resource manager for geo-distributed data-intensive applications.

[31] Kloudas, K., Mamede, M., Preguiça, N., and Rodrigues, R. Pixida: optimizing data parallel jobs in wide-area data analytics. In *VLDB* (2015).

[32] Kraska, T., Pang, G., Franklin, M. J., Madden, S., and Fekete, A. Mdcc: Multi-data center consistency. In *EuroSys* (2013).

[33] Kulkarni, S., Bhagat, N., Fu, M., Kedigehalli, V., Kellogg, C., Mittal, S., Patel, J. M., Ramasamy, K., and Taneja, S. Twitter heron: Stream processing at scale. In *SIGMOD* (2015).

[34] Kumar, A., Jain, S., Naik, U., Raghuraman, A., Kasinadhuni, N., Zermeno, E. C., Gunn, C. S., Ai, J., Carlin, B., Amarandei-Stavila, M., et al. Bwe: Flexible, hierarchical bandwidth allocation for wan distributed computing. In *SIGCOMM* (2015).

[35] Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM 21*, 7 (July 1978), 558–565.

[36] LAMPORT, L. The part-time parliament. *ACM Trans. Comput. Syst.* (1998).

[37] LAMPORT, L. Fast paxos. *Distributed Computing* (2006).

[38] LIN, Y., ET AL. Middleware based data replication providing snapshot isolation. In *SIGMOD* (2005).

[39] LIN, Y., ET AL. Enhancing edge computing with database replication. In *SRDS* (2007).

[40] LLOYD, W., FREEDMAN, M. J., KAMINSKY, M., AND ANDERSEN, D. G. Don't settle for eventual: scalable causal consistency for wide-area storage with cops. In *SOSP* (2011).

[41] LLOYD, W., FREEDMAN, M. J., KAMINSKY, M., AND ANDERSEN, D. G. Stronger semantics for low-latency geo-replicated storage. In *NSDI* (2013).

[42] MAHMOUD, H. A., NAWAB, F., PUCHER, A., AGRAWAL, D., AND EL ABBADI, A. Low-latency multi-datacenter databases using replicated commits. In *VLDB* (2013).

[43] NAWAB, F., AGRAWAL, D., AND EL ABBADI, A. Message futures: Fast commitment of transactions in multi-datacenter environments. In *CIDR* (2013).

[44] NAWAB, F., ARORA, V., AGRAWAL, D., AND ABBADI, A. Chariots: A scalable shared log for data management in multi-datacenter cloud environments. EDBT.

[45] NAWAB, F., ARORA, V., AGRAWAL, D., AND EL ABBADI, A. Minimizing commit latency of transactions in geo-replicated data stores. In *SIGMOD* (2015).

[46] OKTAY, K. Y., MEHROTRA, S., KHADILKAR, V., AND KANTARCIOGLU, M. Semrod: Secure and efficient mapreduce over hybrid clouds. In *SIGMOD* (2015).

[47] PANG, G., KRASKA, T., FRANKLIN, M. J., AND FEKETE, A. Planet: making progress with commit processing in unpredictable environments. In *SIGMOD* (2014).

[48] PATTERSON, S., ELMORE, A. J., NAWAB, F., AGRAWAL, D., AND ABBADI, A. E. Serializability, not serial: Concurrency control and availability in multi-datacenter datastores. *PVLDB* (2012).

[49] PU, Q., ANANTHANARAYANAN, G., BODIK, P., KANDULA, S., AKELLA, A., BAHL, P., AND STOICA, I. Low latency geo-distributed data analytics. In *SIGCOMM* (2015).

[50] QURESHI, M. K., GURUMURTHI, S., AND RAJENDRAN, B. Phase change memory: From devices to systems. *Synthesis Lectures on Computer Architecture 6*, 4 (2011), 1–134.

[51] RABKIN, A., ARYE, M., SEN, S., PAI, V. S., AND FREEDMAN, M. J. Aggregation and degradation in jetstream: Streaming analytics in the wide area. In *NSDI* (2014).

[52] ROY, S., KOT, L., BENDER, G., DING, B., HOJJAT, H., KOCH, C., FOSTER, N., AND GEHRKE, J. The homeostasis protocol: Avoiding transaction coordination through program analysis. In *SIGMOD* (2015).

[53] SHANKARANARAYANAN, P., SIVAKUMAR, A., RAO, S., AND TAWARMALANI, M. Performance sensitive replication in geo-distributed cloud datastores. In *DSN* (2014).

[54] SHAROV, A., SHRAER, A., MERCHANT, A., AND STOKELY, M. Take me to your leader!: online optimization of distributed storage configurations. *Proceedings of the VLDB Endowment 8*, 12 (2015), 1490–1501.

[55] SHUTE, J., OANCEA, M., ELLNER, S., HANDY, B., ROLLINS, E., SAMWEL, B., VINGRALEK, R., WHIPKEY, C., CHEN, X., JEGERLEHNER, B., ET AL. F1: the fault-tolerant distributed rdbms supporting google's ad business. In *SIGMOD* (2012).

[56] SHUTE, J., VINGRALEK, R., SAMWEL, B., HANDY, B., WHIPKEY, C., ROLLINS, E., OANCEA, M., LITTLEFIELD, K., MENESTRINA, D., ELLNER, S., ET AL. F1: A distributed sql database that scales. In *VLDB* (2013).

[57] SOVRAN, Y., POWER, R., AGUILERA, M. K., AND LI, J. Transactional storage for geo-replicated systems. In *SOSP* (2011).

[58] TERRY, D. B., PRABHAKARAN, V., KOTLA, R., BALAKRISHNAN, M., AGUILERA, M. K., AND ABU-LIBDEH, H. Consistency-based service level agreements for cloud storage. In *SOSP* (2013).

[59] THUSOO, A., SHAO, Z., ANTHONY, S., BORTHAKUR, D., JAIN, N., SEN SARMA, J., MURTHY, R., AND LIU, H. Data warehousing and analytics infrastructure at facebook. In *SIGMOD* (2010).

[60] TOSHNIWAL, A., TANEJA, S., SHUKLA, A., RAMASAMY, K., PATEL, J. M., KULKARNI, S., JACKSON, J., GADE, K., FU, M., DONHAM, J., ET AL. Storm@ twitter. In *SIGMOD* (2014).

[61] VULIMIRI, A., CURINO, C., GODFREY, B., KARANASOS, K., AND VARGHESE, G. Wanalytics: Analytics for a geo-distributed data-intensive world. In *CIDR* (2015).

[62] VULIMIRI, A., CURINO, C., GODFREY, B., PADHYE, J., AND VARGHESE, G. Global analytics in the face of bandwidth and regulatory constraints. In *NSDI* (2015).

[63] WU, Z., BUTKIEWICZ, M., PERKINS, D., KATZ-BASSETT, E., AND MADHYASTHA, H. V. Spanstore: Cost-effective geo-replicated storage spanning multiple cloud services. In *SOSP* (2013).

[64] WU, Z., YU, C., AND MADHYASTHA, H. V. Costlo: Cost-effective redundancy for lower latency variance on cloud storage services. In *NSDI* (2015).

[65] ZAKHARY, V., NAWAB, F., AGRAWAL, D., AND EL ABBADI, A. Db-risk: The game of global database placement. In *SIGMOD* (2016).

[66] ZHANG, Y., POWER, R., ZHOU, S., SOVRAN, Y., AGUILERA, M. K., AND LI, J. Transaction chains: achieving serializability with low latency in geo-distributed storage systems. In *SOSP* (2013).