Research Statement

Global-Scale Data Management

Faisal Nawab
http://nawab.me

Processing large quantities of data is becoming more ubiquitous and is the driving force behind the sustained growth and impact of Internet Services and Big Data analytics. The way data-intensive applications are deployed has been radically transformed by the cloud computing paradigm realized through massive-scale datacenters. However, datacenter-scale failures have occurred numerous times in the past and continue occurring many times annually due to various events such as power outages and natural disasters. These failures impact all services in a datacenter for extended periods of time and cause disruption to a large number of users, leading to losses in revenue and utility of Internet and Big Data applications. Also, deployment at a single datacenter gives rise to very high latency for user requests that are geographically distant.

Recently, Big Data and large-scale systems have addressed datacenter-scale failures and high service response times by distributing applications and data globally across multiple datacenters, thus providing datacenter-scale fault-tolerance and data proximity to users. Methods based on asynchronous replication are widely used in practice to keep the replicas up-to-date with each other. Asynchronous replication, however, is vulnerable to data losses and inconsistency when failures occur, which threaten the integrity of the applications. This also complicates application design and recovery for developers and system administrators. This is now broadly recognized as a serious drawback that is limiting the adoption of multi-datacenter deployments. This has led to a growing interest from both industry and academia in *global-scale* systems with stringent *consistency* guarantees.

Research Summary. Through my dissertation research, *I investigate the fundamental challenges and best practices for designing consistent **Global-Scale Data Management (GSDM)** systems.* Consistent GSDM systems allow developers and administrators to deploy applications globally without sacrificing the benefits of easy-to-use and easy-to-manage traditional data management systems while maintaining the integrity of the applications. Thus, it facilitates the adoption of the fault-tolerance and performance benefits of GSDM for a wide-range of Big Data applications and Internet Services. Consistency, however, requires *coordination* between replicas to ensure that requests do not overwrite or contradict each other. Global-scale coordination is expensive due to the large wide-area latency between datacenters. The wide-area latency ranges from hundreds of milliseconds up to several seconds. This is 2–4 orders of magnitude larger than the typical communication latency between replicas within a single datacenter. Such a high latency, studies show, drives a significant percentage of users to abandon the application, *e.g.*, more than half the users leave applications with a multi-second delay. High latency, in addition to being a disadvantage in itself, also makes other performance characteristics, such as scalability and throughput, more challenging.

In my work, I have focused on addressing the grand challenges of adopting multi-datacenter deployments for Big Data and large-scale systems. Application developers and administrators are torn between easy-to-use consistent GSDM systems that perform poorly and high-performance asynchronous replication that is difficult to develop and use correctly. My work aims at solving this dilemma by *designing a new generation of consistent GSDM systems that are scalable and achieve high performance with significantly improved coordination latency compared to traditional consistent GSDM systems.* My approach to research work begins with studying aspects of the global-scale environment and GSDM systems to extract the main challenges and problems of GSDM. I then use newfound insights and understanding from such studies to propose fundamental design principles that improve the performance of consistent GSDM systems.

In the following, I summarize my dissertation research followed by my other work. I then conclude with my vision of a **globally-connected, data-driven world** and the data management challenges it faces.

Global-Scale Data Management

Understanding Global-Scale Coordination. To design efficient consistent GSDM systems, it is essential to understand the fundamental challenges and characteristics of global-scale coordination. To gain such an understanding, I started my Ph.D. work with studying existing consistent GSDM systems to extract the design characteristics that led to poor performance. With this understanding, I propose evolutions of these protocols that perform better in the global-scale environments. My work on **Paxos-CP** [14] is a product of such a study on a consensus protocol called Paxos. Paxos is an essential component in many large-scale consistent systems such as Google Megastore. However, it was designed for distributed systems without considering the intricate challenges of global-scale Big Data applications. Paxos orders requests sequentially, thus limiting its throughput in environments with large communication latency. Paxos-CP identifies the sequential guarantee of Paxos as too strict for data management (transactional) workloads that only need *serializable* consistency. With this observation, techniques to reach consensus that allow more concurrency while maintaining serializability are developed. My other work on **Replicated Commit** [4] studies the limitations of a traditional approach of fault-tolerance for partitioned data that is adopted widely by GSDM systems such as Google Spanner. In this traditional approach, each partition is made fault-tolerant by replicating a log of processing steps to other datacenters. Multiple steps are replicated for each request, causing coordination latency to be amplified. Replicated Commit proposes a method to replicate the whole request as a single piece—rather than replicating each step—even for systems that partition data like Spanner. Thus, only a single wide-area round of communication is needed, reducing the request latency significantly. Paxos-CP and Replicated Commit are two of the first studies that shed light on the effect of the wide-area latency limit on coordination latency and proposed designs specifically to improve coordination performance in a global-scale environment.

Breaking the Latency Barrier of the Request-Response Paradigm. The studies conducted for Paxos-CP and Replicated Commit exposed a fundamental coordination latency limit. This limit is due to the polling nature of traditional protocols that I call the Request-Response paradigm. In the Request-Response paradigm, the coordination for a request starts *after* the request is made, where the replica that received it polls other replicas to inquire about their state and detect conflicts. The request is served only after receiving a *response* from other replicas. This makes a Round-Trip Time (RTT) of communication inevitable—an expensive cost in GSDM. This leads to the question: *Is it possible to avoid the Request-Response paradigm of coordination?* My work on **Message Futures** [5] demonstrates this possibility by an observation that coordination of future requests can start before they arrive. As requests arrive, they are assigned to a predetermined future coordination point. I call this approach *Futures Coordination*. Coordination points are judiciously calculated to ensure conflicts are detected. A coordination point still needs an RTT for coordination. However, because a request is assigned to a coordination point that already started, the request's observable latency is less than RTT. Message Futures is the first protocol that shows the possibility of faster-than-RTT coordination for all the replicas of a distributed system. Also, it introduces Futures Coordination, a new approach to coordination that overcomes the limitations of the Request-Response paradigm.

Theoretical Lower-Bound on Coordination Latency. Breaking the RTT latency barrier via the Futures Coordination paradigm invalidates the previously held convention that coordination cannot be performed faster than the RTT latency. Thus, it opens the question: *What is the lower-bound on coordination latency?* This is a fundamental question in understanding the extent of the effect of the wide-area latency limit on coordination latency. Such a lower-bound, if proven, will provide system designers and researchers with a theoretical foundation on what is achievable by current and future systems.

To tackle the question of lower-bound coordination latency, I begin by formalizing and modeling the concept of coordination, which is the process of detecting a conflict between two requests. This model of coordination is then used to answer the question: *What are the cases that make detecting a conflict between two requests impossible*? To answer this question, I observe that for any potential conflict between two requests, a and b, one of them must know about the other before making the decision (commit or abort). If they both commit without knowing about the other, then they do not detect the conflict, possibly leading to a consistency violation. My work shows that these cases are inevitable if the coordination latency of a *plus* the coordination latency of b is less than the RTT between the datacenters hosting them. Thus, for any consistent global-scale system, the sum of the coordination latency of any two transactions must be greater than or equal to the RTT between their host datacenters [8]. For example, it is possible that the latency of both a and b is equal to half the RTT between their host datacenters. The coordination model inspired a protocol based on Futures Coordination, called **Helios** [8] that theoretically achieves the lower-bound, thus proving that the lower-bound is tight.

The lower-bound result shows that the coordination latency can be faster than what is previously achieved by traditional protocols and even faster than what is achieved by Message Futures. The model of coordination, in addition to being essential for deriving the lower-bound, advances our understanding of the cost of global-scale coordination. It also brings a newfound understanding of the latency characteristics of traditional and Futures Coordination protocols.

Global-Scale Data Communication. Large-scale Big Data applications process massive amounts of data that cannot be supported by traditional communication protocols. I address this problem by investigating communication designs that scale to the needs of large-scale applications and be able to support coordination-oriented communication, such as the communication needed for Message Futures, Helios, and other consistent GSDM protocols. **Chariots** [7] is the product of this investigation. To scale Chariots to large-scale workloads, the task of communication is made a priority. Chariots manages a group of machines dedicated for multi-datacenter communication that provides global-scale communication as a service to applications. The problem of communications scalability is tackled by observing that the traditional total ordering guarantees of communication protocols are too strict for coordination-related communication. Rather, it turns out that *causal-order* guarantees are sufficient for coordination-related communication. The design of Chariots proposes novel methods of managing distributed causally-ordered communication that enable it to scale to the demands of large-scale applications.

Global-Scale Data Placement and Configuration. The coordination latency of GSDM systems is directly influenced by the deployment topology, *i.e.*, the locations of used datacenters and the communication latency between each pair of datacenters. **DB-Risk** [15] is an interactive tool that allows system administrators to experiment with various placement and configuration scenarios. This facilitates a better understanding of the effect of placement and configuration decisions in a global-scale topology. In my own experience with DB-Risk, I was able to deduce design optimizations to coordination protocols to improve performance. The most surprising of these optimizations is *request handoff* where it turns out that sometimes better latency is achieved if *users send requests to a remote datacenter rather than the closest or local datacenter* [15].

Machine Learning with Globally-Generated Data. Machine learning is essential for Big Data analytics. This motivated my work on **COP** [6] that specifically targets efficiently supporting global-scale machine learning workloads. In typical global-scale machine learning, data is collected at different locations around the world and then processed at a centralized location. COP targets improving the learning performance for this *Collect then Learn* pattern. COP's main purpose is to pre-process data as it is collected so that when it is received at the centralized location it can be processed faster. COP's approach ensures a partial order of the execution that will preserve the used machine learning algorithm's theoretical properties. The pre-processing allows COP to enforce the partial order with light-weight operations that outperform traditional methods.

Other Projects

Data Management over Emerging Memory Technology. The innovations in hardware technology enable and drive software system designs to support the ever increasing demands of large-scale Big Data applications. Therefore, it is important for system researchers to follow and anticipate the advances in hardware technology. This motivated me to pursue internships and collaborations with industry where I am exposed to real world problems and cutting-edge research on both hardware and software.

During my graduate studies, I spent two years working with a team of systems and programming languages researchers in a collaborative project with HP Labs. In this collaboration, I studied the challenges and opportunities of adopting emerging Non-Volatile Memory (NVM) technology for large-scale data management. In my first project with HP Labs, I saw in a recent hardware technology an opportunity to reduce NVM durability overheads. This technology is *flush-on-fail*, that allows persisting the volatile state of the cache when a failure is signaled. I studied the implications of having flush-on-fail support on the NVM durability overheads. **Procrastination Beats Prevention (PBP)** [9] is the product of this study, where I show that it is possible to avoid cache-line flushing (*i.e.*, pushing data from volatile CPU caches to persistent NVM) overhead completely, even for transactional workloads, via the use of flush-on-fail. Also, I prove that combining flush-on-fail with non-blocking programs obviates the need for logging. The outcome of the work is a realization that flush-on-fail and non-blocking algorithms lead to *Zero-Overhead NVM Crash Resilience* [9, 10]. PBP introduces a new way of thinking about durability, where program-specific properties can be leveraged for cheaper durability techniques.

The work in PBP was instrumental to highlight fundamental design principles for large-scale applications on NVM. This inspired my work on **Dali** [11], a transactional data management system. Prior approaches for durability either relied on group commit and suffered from logging overheads, or relied on in-place persistence and suffered from flushing overheads. Dali's main innovation is that it utilizes CPU whole-cache-flush instructions in a specialized data structure that do not require external logging or cache-line flushing, thus not suffering from their overheads. Dali introduces a new family of durability techniques that offer better performance than traditional group commit and in-place persistence for NVM. A patent application of this work is filed and a research paper about it is under submission.

As a research intern in Microsoft Research, I have worked on building data access technology that adds the functionality of temporal multi-versioned B-Trees to high-performance lock-free indexing. The product is the Time-Split Bw-tree **(TSBw-tree)** [2] that integrates the algorithms of a multi-versioned Time-split B-tree within the implementation of a lock-free B-tree called the Bw-tree.

Other than the research work I listed, I have also worked on fairness for Wireless Mesh Networks' communication [12, 13], distributed transaction processing [3], and graph summarization techniques for visualization of social trends [1] (in collaboration with UC Santa Barbara's department of Film & Media Studies). Working on these diverse research problems in addition to my focus on GSDM and modern hardware technology, broadens the scope of my research and enables me to tackle problems with the tools of many disciplines.

Future Research

Vision (A Globally-Connected, Data-Driven World). The technology and computing trends are driving us to an influx of *machine-centric applications* (*e.g.*, Internet of Things (IoT), mobile and autonomous systems) and *data-centric applications* (*e.g*, data science and Big Data). Devices will be integrated to many aspects of our lives. These devices will be *globally connected* and will *generate data* that can be used to extract knowledge for various applications. To enable this vision, it is necessary to transform the global-scale system infrastructure to support the explosion of the number of devices and amount of generated data. Also, it is necessary to transform data processing platforms to support the increasingly diverse and complex data-intensive applications. My future research directions tackle these challenges at the frontier of global connectivity, data science, and their intersection, and will enable the advances that are made. Next, I outline these directions:

Data Management at the Edge. One of the vision's grand challenges is building global-scale systems that are capable of supporting the influx of connected devices and generated data. However, the demands of such an environment are so large that the current datacenter-based infrastructure does not have the capacity of supporting it. I foresee that a promising path to increasing the performance and capacity of the global-scale infrastructure is to augment Edge computing technology—beyond its traditional applications—with data management capabilities. However, to utilize Edge resources efficiently, it is necessary to build data management systems specifically for this new environment. This includes revisiting many of the traditional data management problems such as partitioning, placement, load balancing, and fault-tolerance.

Managing Global-Scale Diverse and Heterogeneous Data. Emerging machine- and data-centric applications generate and process diverse workloads with various platforms. GSDM will face new workloads such as event streams of IoT, event-driven workloads of autonomous systems, and data science and machine learning workloads. These workloads are often interconnected and used to serve common high-level functionalities. For example, a single IoT application may be simultaneously processing workloads that are transactional (for the control and web interface), analytical (for business intelligence), and streaming (for data generated by sensors and commands sent to actuators). This, however, makes the task of developers extremely difficult as their applications need to interact with various data processing platforms. This invites efforts to provide unifying high-level managers that abstract the underlying platforms to application developers. This raises many research problems on how to design the high-level manager, how to integrate the various platforms, and how to abstract the data processing platforms to enable supporting future applications.

Large-Scale Systems for Data Science. Developing data-centric applications, such as Big Data and Data Science applications, faces two main challenges. First, the processing requirements are increasing in both the complexity of the processing tasks and the volume and velocity of data. To support these increasing demands, it is necessary to incorporate the advances in both large-scale system software technology in addition to the advances in hardware and memory architecture. My experience with large-scale Big Data systems over emerging memory technology prepares me to introduce such advancements in data science applications. The second challenge is that these applications are often at the intersection of computer science and other disciplines, requiring interdisciplinary collaborations. My past experience in collaborative work (*e.g.*, with the Film & Media Studies department [1] and machine learning researchers [6]) taught me the best practices for initiaiting and pursuing such interdisciplinary research.

In the long run, my goal is to design a data science framework that facilitates the development of a wide-range of data science applications. The current data processing frameworks adopt a top-down approach, where frameworks—built by computer science researchers in isolation—are designed with a pre-defined processing model that constrains data science applications. This has limited the adoption of these frameworks by many applications. I contend that a bottom-up approach to this problem is needed, where problems of different data science applications are solved separately. Then, common processing patterns are extracted to form the basis of the data processing framework. Being inspired by real applications makes the framework able to support future applications more efficiently. This is an arduous process that requires multiple long-term collaborations with researchers of the targeted applications.

References

[1] C. Biafore and F. Nawab. Graph summarization for geo-correlated trends detection in social networks. In *SIGMOD*, p. 2247–8, 2016.

[2] D. B. Lomet and F. Nawab. High performance temporal indexing on modern hardware. In *ICDE*, p. 1203–4, 2015.

[3] H. Mahmoud, V. Arora, F. Nawab, D. Agrawal, and A. El Abbadi. MaaT: Effective and scalable coordination of distributed transactions in the cloud. *Proc. VLDB Endow.*, 7(5):329–40, Jan. 2014.

[4] H. Mahmoud, F. Nawab, A. Pucher, D. Agrawal, and A. El Abbadi. Low-latency multi-datacenter databases using Replicated Commit. *Proc. VLDB Endow.*, 6(9):661–72, July 2013.

[5] F. Nawab, D. Agrawal, and A. El Abbadi. Message Futures: Fast commitment of transactions in multi-datacenter environments. In *CIDR*, 2013.

[6] F. Nawab, D. Agrawal, A. El Abbadi, and S. Chawla. COP: Planning conflicts for faster parallel transactional machine learning. In *EDBT*, 2017.

[7] F. Nawab, V. Arora, D. Agrawal, and A. El Abbadi. Chariots: A scalable shared log for data management in multi-datacenter cloud environments. In *EDBT*, p. 13–24, 2015.

[8] F. Nawab, V. Arora, D. Agrawal, and A. El Abbadi. Minimizing commit latency of transactions in geo-replicated data stores. In *SIGMOD*, p. 1279–94, 2015.

[9] F. Nawab, D. R. Chakrabarti, T. Kelly, and C. B. Morrey III. Procrastination beats prevention: Timely sufficient persistence for efficient crash resilience. In *EDBT*, p. 689–94, 2015.

[10] F. Nawab, D. R. Chakrabarti, T. Kelly, and C. B. Morrey III. Zero-overhead NVM crash resilience. In *Non-Volatile Memory Workshop (NVMW)*, 2015.

[11] F. Nawab, J. Izraelevitz, T. Kelly, C. B. Morrey III, and D. R. Chakrabarti. High-performance transactions on storage-class memory with communal commit. *Patent application filed and paper is under submission*, 2016.

[12] F. Nawab, K. Jamshaid, B. Shihada, and P. H. Ho. TMAC: Timestamp-ordered MAC for CSMA/CA Wireless Mesh Networks. In *ICCCN*, p. 1–6, 2011.

[13] F. Nawab, K. Jamshaid, B. Shihada, and P. H. Ho. Fair packet scheduling in wireless mesh networks. *Ad Hoc Networks*, 414–27, 2014.

[14] S. Patterson, A. J. Elmore, F. Nawab, D. Agrawal, and A. El Abbadi. Serializability, not serial: Concurrency control and availability in multi-datacenter datastores. *Proc. VLDB Endow.*, 5(11):1459–70, 2012.

[15] V. Zakhary, F. Nawab, D. Agrawal, and A. El Abbadi. Db-risk: The game of global database placement. In *SIGMOD*, p. 2185–8, 2016.