Poster: Pathstore, A Data Storage Layer For The Edge

Seyed Hossein Mortazavi
University of Toronto

Bharath Balasubramanian
AT&T Labs-Research

Eyal de Lara
University of Toronto

Shankaranarayanan Puzhavakath Narayanan
AT&T Labs-Research

CCS CONCEPTS

• **Computer systems organization** → *Cloud computing*;

1 INTRODUCTION

With Edge Computing, traditional cloud architectures have transformed into multi-tier data centers that are distributed over the geographical span of the network. These architectures support new mobile and Internet of Things (IoT) applications [4] that require low latency, or produce large volumes of data by providing computation and storage in the vicinity of the end users and devices. Through Edge computing, a wide-area cloud data center which serves a large country can be extended to a hierarchy of datacenters that provide coverage for different geographical areas such as cities, neighborhoods, and buildings.

Offloading the compute to the edge has received significant attention of-late, which brings about the challenges in maintaining the associated state across multiple sites. Examples include a wearable smart assistant that could execute latency sensitive and bandwidth intensive functions, such as audio data analysis or face recognition on a nearby edge datacenter, while executing infrequent and latency tolerant functions, such as user authentication and preference editing, on a traditional wide-area cloud datacenter.

1.1 Consistency Models For the Edge

Web services and applications are typically deployed on highly available storage layers that commonly relax consistency between replicas. A common approach for web services and applications in these scalable systems is to rely on *eventual consistency* where the storage system guarantees that if no new updates are made to an object, eventually all reads will return the last updated value [1]. However, many eventually consistent systems implicitly provide a stronger consistency model called *session consistency*, where a client performs all operations pertaining to a single session at any one replica of the state store ("sticky sessions"). Session consistency guarantees include: (i) *read-your-writes*, where subsequent reads by a client that has updated an item will return the updated value or a newer one; and, (ii) *monotonic reads*, where if a client has seen a particular value for the object, subsequent reads will return the same value or a newer one[5]. This may not hold for edge computing where the mobile nature of clients and the inherent partitioning of applications prevents sticky sessions. Hence, more involved algorithms are needed to provide session consistency at the edge. In addition, some services such as authentication require *strongly consistent* operations (i.e. consistent joins, transactions) all of which depend on some form of distributed consensus. Achieving distributed consensus even across a few data centers separated by the wide-area network due to both performance and availability is difficult. This is exacerbated by edge computing wherein there might be hundreds and thousands of data centers. So, there need to be a hierarchical techniques that enable consensus in a scalable manner.

2 PATHSTORE

PathStore [2, 3] is a hierarchical storage system designed to provide a storage layer for edge computing applications. The PathStore hierarchy consists of a persistent replica at its root, and an unlimited number of layers of partial replicas below it. Data replication is on demand in response to application queries. PathStore supports concurrent object reads and writes on all replicas of the hierarchy; updates are propagated through the hierarchy in the background, providing eventual consistency [3]. PathStore also supports session consistency using replica reconciliation algorithms that executes when a client switches from a source replica to a destination replica. The basic reconciliation algorithm involves tracking all the items either read or written on the source replica by the session and ensuring the destination has as up to date values by obtaining them from the source. The eventually consistent state-store of PathStore can be further extended to provide other consistency guarantees by adding a strongly consistent locking service. While PathStore can provide strong consistency guarantees on a single replica, an on demand hierarchical locking mechanism that passes on locks for specific rows between replicas through atomic operations, can provide stronger consistency requirements for applications.

MobiSys'18, June 10–15, 2018, Munich, Germany

ACM ISBN 978-1-4503-5720-3...$15.00
https://doi.10.1145/3210240.3210813

REFERENCES

[1] Kawell Jr, L., Beckhardt, S., Halvorsen, T., Ozzie, R., and Greif, I. Replicated document management in a group communication system. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work* (1988), ACM, p. 395.
[2] Mortazavi, S. H., Balasubramanian, B., de Lara, E., and Shankaranarayanan, P. N. Toward session consistency for the edge. submitted to the USENIX Workshop on Hot Topics in Edge Computing, 2018.
[3] Mortazavi, S. H., Salehe, M., Gomes, C. S., Phillips, C., and de Lara, E. Cloudpath: A multi-tier cloud computing framework. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing* (New York, NY, USA, 2017), SEC '17, ACM, pp. 20:1–20:13.
[4] Shi, W., and Dustdar, S. The promise of edge computing. *Computer 49*, 5 (2016), 78–81.
[5] Viotti, P., and Vukolić, M. Consistency in non-transactional distributed storage systems. *ACM Computing Surveys (CSUR) 49*, 1 (2016), 19.