Unifying Updates in Distributed Polystores at Memory Speed
(Corporate Partner Research Proposal)

PI: Mohammad Sadoghi
Assistant Professor
Department of Computer Science
Purdue University

Phone: 914-319-7937
Email: msadoghi@cs.purdue.edu
Website: https://msadoghi.github.io/

Project Description

Overview:

In this proposal, we focus on a new data management era centered around data heterogeneity by **developing a real-time, distributed, updatable polystore**. Arguably, today's **enterprise is now faced with an unparalleled data variety** that is growing at an unprecedented volume and velocity. This data variety is rooted in a technological revolution that is now unfolding and has broad impacts ranging from everyday life (e.g., personalized medicine and education) to every industry (e.g., data-driven healthcare, commerce, agriculture, and mining). This transformation is facilitated by sensing, gathering, and connecting all physical entities, collectively known as Internet of Things (IoT), to construct a rich and dynamic computational model of reality in real-time. Every procedure and every decision needed in the physical world will soon be optimized in real-time by ingesting and analyzing massive volume of heterogeneous data at high velocity. To cope with such extreme scale, I propose to build **a generic lineage-based storage architecture to unify the ingestion of real-time updates across the federation of heterogeneous data models** that will be present in a polystore. This proposal will be implemented within ExpoDB initiative, *an exploratory data science platform*, that I am leading in my research group at Purdue [5].

Intellectual Merit:

In this proposal, building upon my past work on building hybrid transactional/analytical processing (HTAP) platforms [1, 2, 4, 3, 5, 6], I plan to study the key problem of how to develop a lineage-based storage architecture (LSA) in order **to lazily stage mutable data (i.e., optimized for high-velocity transactional workloads) into fault-tolerant, immutable form (i.e., optimized for massive-volume analytical workloads) that facilitates storing and querying heterogeneous data (i.e., data-model agnostic)** in a transactionally consistent approach. My broader goal is to employ modern hardware such as RDMA and GPUs in polystore. Given that my envisioned LSA will reside in a distributed shared memory, then I plan **to exploit RDMA to efficiently support remote fine-grained access that guarantees strong consistency**. Furthermore, I plan **to employ GPUs to transform the LSA into a fast, distributed content-addressable storage**, i.e., given an object's key quickly find its physical location. The direct outcome of my updatable polystore vision is to substantially enhance and foster the open-source research infrastructure by implementing my proposed plan inside ExpoDB that is built within the Apache Hadoop and Spark ecosystems.

Technical Approach:

The first step towards addressing data variety obstacle is the emergence of new class of data management systems knowns as polystores [7], i.e., a federation of heterogeneous data models including structured (e.g., relational model), semi-structured (e.g., graph, RDF, XML, JSON), and unstructured data (text, images, speech, and video). At the core of polystore lies a unifying query language and a basic framework to bridge the runtime engines of these heterogeneous and disparate platforms (as demonstrated in Figure 1). To enable this unification, the first ever realized polystore referred to as BigDAWG [7], initiated by Intel Science and Technology Center, offers two key runtime interoperability constructs as first-class citizens, namely, shim and cast. A *shim* translates the unified query language into a subquery dialect that is native to each data model, while a *cast* transforms data from one data model to another. However, BigDAWG does not address the essential question of how to update highly heterogeneous data at scale. Therefore, I primarily focus on the unification of updates in polystore by introducing a lineage-based storage architecture (LSA) that is data model agnostic and enables staging data lazily from a mutable into an immutable form.

A major design point in LSA is the realization that the underlying distributed storage subsystem in data centers must be assumed as strictly immutable in order to ensure simplified fault-tolerance [8] and transac-

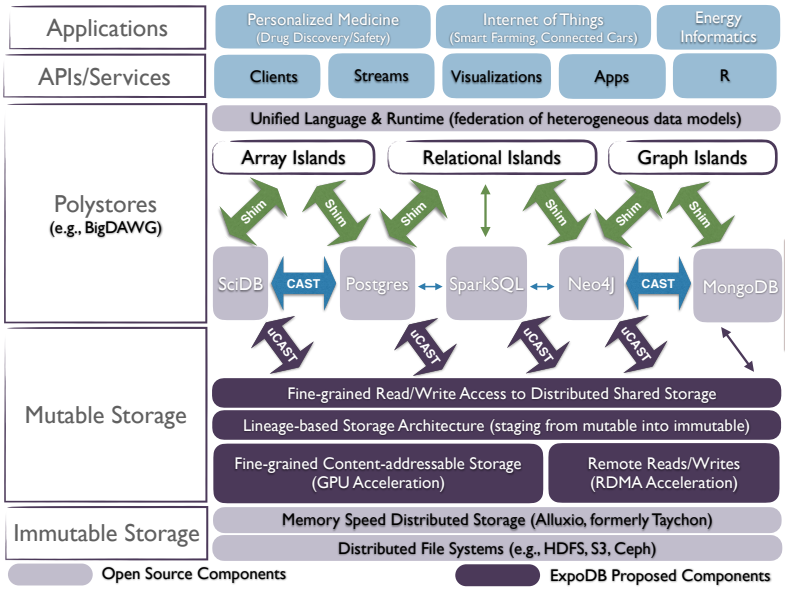

Figure 1: Update Unification in Polystores.

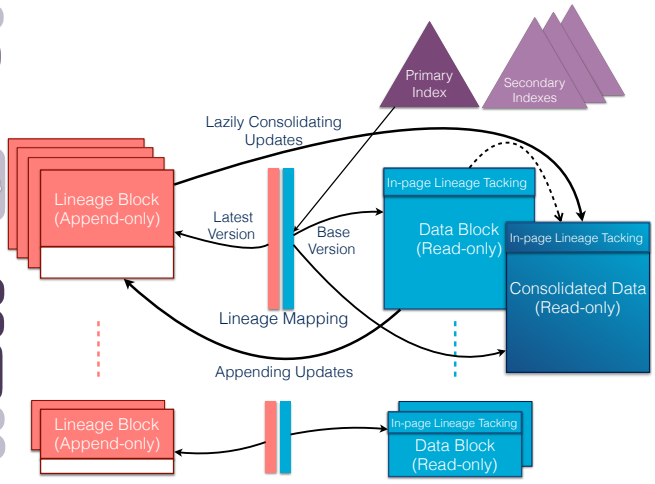

Figure 2: Lineage-based Storage Architecture.

tional consistency guarantees [2, 4]. Therefore, I plan to build the mutable LSA layer over Alluxio (formerly Taychon) [8], which is an immutable distributed caching layer that rests upon immutable distributed file systems such as HDFS, as shown in Figure 1. Through LSA, I will provide a fine-grained read/write access to any objects irrespective of the underlying data model. Therefore, LSA serves as a generic mutable layer to any federation of heterogeneous data. Each data model in polystore simply pushes its changes to LSA through a generic *uCast* construct. Likewise, any engine in polystore can fetch any data from LSA through *uCast*, which in essence serve as a loosely coupled integration of ploystore with LSA and subsequently the entire Hadoop ecosystem.

The basic design of LSA that grants data model agnostic property consists of two key ideas as captured in Figure 2: (i) The base data is kept in read-only blocks, where block is simply an ordered set of objects of any type, while the modification to the *data blocks* are accumulated in the corresponding *lineage blocks*. (ii) In between, sits a lineage mapping that links an object in data blocks to its recent updates in a lineage block, essentially, decoupling updates from the physical location of objects. Therefore, from the lineage mapping, both the base and updated data are retrievable. Through a lazy background process, recent updates are merged with their corresponding read-only base data in order to construct a new set of consolidated data blocks (necessary to ensure the optimal analytical queries performance). Furthermore, each data block tracks its lineage in-page, i.e., maintaining the lineage of the update history consolidated thus far. By exploiting the decoupling and in-page lineage tracking, the merge process, which only creates a new set of read-only consolidated data blocks, is carried out completely independently from update queries, which only append changes to lineage blocks and update the lineage mapping. Hence, there is no contention in the write paths of update queries and the merge process, which is the fundamental property necessary to achieve a highly-scalable distributed storage layer that is updatable.

Budget:

I request a sufficient budget to support one PhD student for a year. I further request to purchase two compute nodes in Purdue's Community Clusters (Halstead), which will be the gateway to access all 408 nodes in the cluster. Each node has Two 10-Core Intel Xeon-E5 and 128 GB of memory. All nodes in the cluster are connected through 100Gbps RDMA links. The cost of each node is $3,600.

References Cited

[1] **Mohammad Sadoghi**, Kenneth A. Ross, Mustafa Canim, and Bishwaranjan Bhattacharjee. Making updates disk-I/O friendly using SSDs. *PVLDB*, 6(11):997–1008, 2013.

[2] **Mohammad Sadoghi**, Souvik Bhattacherjee, Bishwaranjan Bhattacharjee, and Mustafa Canim. L-Store: A real-time OLTP and OLAP system. *CoRR*, abs/1601.04084, 2016.

[3] Masoud Hemmatpour, Bartolomeo Montrucchio, Maurizio Rebaudengo, and **Mohammad Sadoghi**. Kanzi: A distributed, in-memory key-value store. In *Proceedings of the 17th International Middleware Conference, Middleware 2016, Trento, Italy, December 12-16, 2016*, pages 3–4, 2016.

[4] Kaiwen Zhang, **Mohammad Sadoghi**, and Hans-Arno Jacobsen. DL-store: A distributed hybrid OLTP and OLAP data processing engine. In *36th IEEE International Conference on Distributed Computing Systems, ICDCS 2016, Nara, Japan, June 27-30, 2016*, pages 769–770, 2016.

[5] **Mohammad Sadoghi**. ExpoDB: An exploratory data science platform. In *CIDR 2017, 8th Biennial Conference on Innovative Data Systems Research, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings*, 2017.

[6] Mohammadreza Najafi, Kaiwen Zhang, Hans-Arno Jacobsen, and **Mohammad Sadoghi**. Hardware Acceleration Landscape for Distributed Real-time Analytics: Virtues and Limitations. In *37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017, Atlanta, Georgia, USA, June 5-8, 2017*.

[7] Jennie Duggan, Aaron J. Elmore, Michael Stonebraker, Magda Balazinska, Bill Howe, Jeremy Kepner, Sam Madden, David Maier, Tim Mattson, and Stan Zdonik. The BigDAWG polystore system. *SIGMOD Rec.*, 44(2):11–16, August 2015.

[8] Haoyuan Li, Ali Ghodsi, Matei Zaharia, Scott Shenker, and Ion Stoica. Tachyon: Reliable, memory speed storage for cluster computing frameworks. In *Proceedings of the ACM Symposium on Cloud Computing*, SOCC '14, pages 6:1–6:15, New York, NY, USA, 2014. ACM.