Efficient Cloud Storage Solutions for Data-Intensive Scientific Applications

Student: Xiaoming Gao {gao4@indiana.edu}

Advisor: Prof. Judy Qiu {xqiu@indiana.edu}

School of Informatics and Computing

Indiana University

Scientific research has entered the Big Data era, with terabytes and even petabytes of data generated from sensors, social media and lab experiments across the globe. Researchers nowadays routinely batch load and process large-scale data, which demands the capabilities of streaming data processing and interactive analysis. Numerous cloud storage systems have been proposed and developed in order to satisfy these new requirements and challenges, ranging from low-level block storage to high-level object storage, distributed file systems, and NoSQL database systems.

Cloud platforms such as Amazon EC2 and OpenStack are good candidates to support data-intensive applications through elastic virtual resource scheduling mechanisms. They are able to dynamically provision and allocate more computing resources in the form of virtual machines (VM) or virtual clusters to accommodate increasing requirements for handling larger data sizes. However, compared to the computation capability, the storage space derived from the VM instances is relatively small, which limits the system's capability for storing and processing huge amounts of data. Amazon Web Services solves this problem with an optimized Elastic Block Store (EBS) service utilizing a proprietary implementation. This motivated us to develop the open-source Virtual Block Store (VBS) system and its improved version, VBS-Lustre, which is designed to support persistent large-scale block storage services on cloud platforms. VBS-Lustre can connect VMs to virtual disks that are much larger than the instance storage space, thus solving the space limitation. Moreover, the lifetime of the virtual disks is independent from the VMs, providing a persistent method of storing data and results even when VM instances are terminated. Using Lustre file system as the storage backend, VBS-Lustre is able to support multiple virtual disks to a large number of VMs at the same time, while achieving efficient and scalable read and write performance.

IndexedHBase is a research project that extends the HBase system with inverted indices to support incremental data updating and interactive analysis. As part of my PhD research goal, we design and implement a customizable indexing framework that allows users to flexibly define the most suitable customized index structures to facilitate query evaluation. This framework is further extended with a two-phase parallel query evaluation strategy that can make the best use of user-defined customized index structures and complete complicated queries using MapReduce.

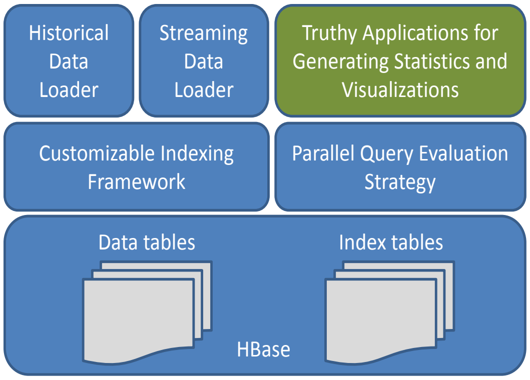
Historical Data Loader
Streaming Data Loader
Truthy Applications for Generating Statistics and Visualizations
Customizable Indexing Framework
Parallel Query Evaluation Strategy
Data tables
Index tables
HBase

Figure 1. System Architecture of IndexedHBase

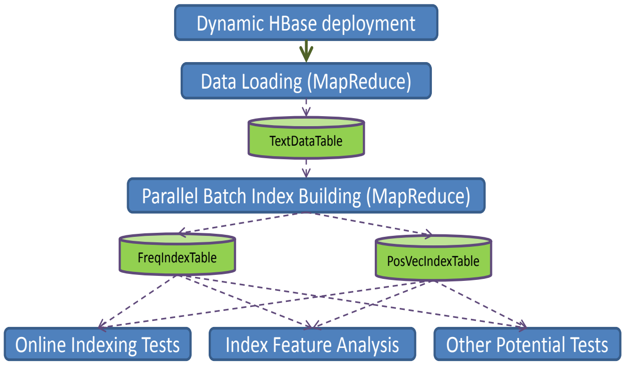

Figure 2. Parallel Data Loading and Index Building

IndexHBase is used to support Truthy project (citation) to build a public social data observatory as an efficient and scalable storage solution to host TB-level large-scale structured social datasets in the format of both historical files and real-time streams. Furthermore, the storage systems must provide efficient evaluation mechanisms for its unique type of query, which could potentially involve analysis of hundreds of millions of social updates. For example, the Truthy social data observatory hosts data from Twitter, and a query may be required to extract all the retweet edges from all tweets containing common hashtags created during a given time window. With the purpose of finding a solution for these challenges, we evaluate NoSQL databases such as Solandra, Riak, and MongoDB,which support text search using distributed inverted indices. However, our investigation shows that traditional inverted indices used in these systems are designed for text retrieval purposes with unnecessary storage and computation overhead for the use case of a social data observatory.

<You can describe the data set up and query features>

We evaluate the performance of IndexedHBase on the Bravo cluster of FutureGrid and compare it with Riak, a widely adopted commercial NoSQL database system. The results show that IndexedHBase provides significantly (6 times) faster data loading speed while requiring much less storage size (Table 1), and is more efficient (by multiple times) in evaluating queries involving large result sets (Figure 3). These queries involve hundreds of millions of social updates and complete execution within seconds or minutes as shown in Figure 2.

Table 1. Data loading performance comparison between Riak and IndexedHBase

	Loading time (hours)	Loaded total data size (GB)	Loaded original data size (GB)	Loaded index data size (GB)
Riak	294.11	3258	2591	667
IndexedHBase	45.47	1167	955	212
Comparative ratio of Riak / IndexeHBase	6.47	2.79	2.71	3.15

Raw data size before loaded is 352GB. Data replication level is set to 2 in both Riak and IndexedHBase.

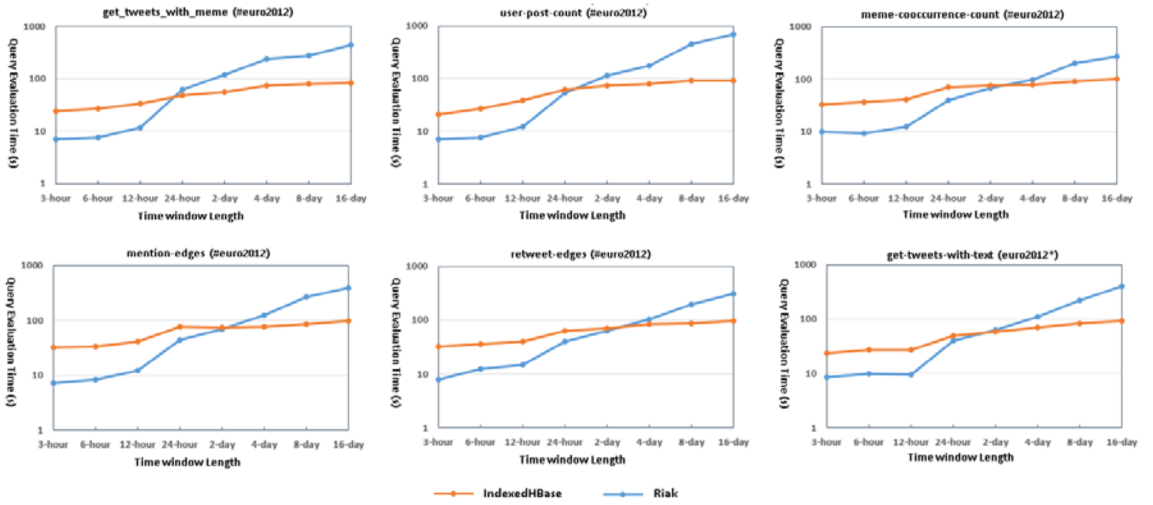

Figure 3. Query evaluation performance comparison between Riak and IndexedHBase

More information at http://mypage.iu.edu/~gao4/ and http://salsaproj.indiana.edu/IndexedHBase/index.html

Selected Publications

[1] **X. Gao**, M. Lowe, Y. Ma, M. Pierce, "Supporting Cloud Computing with the Virtual Block Store System". In Proceedings of e-Science 2009, Oxford, UK, Dec. 2009.

[2] **X. Gao**, Y. Ma, M. Pierce, M. Lowe, G. Fox, Building a Distributed Block Storage System for Cloud Infrastructure. In Proceedings of the 2010 IEEE Second International Conference on Cloud Computing (Indianapolis, IN, USA, November 30 - December 3, 2010).

[3] **X. Gao,** V. Nachankar, J. Qiu. "Experimenting Lucene Index on HBase in an HPC Environment". In Proceedings of the 1st workshop on High-Performance Computing meets Databases at Supercomputing 2011 (Seattle, WA, USA, November 18, 2011).

Selected Publications Under Review

[4] **X. Gao**, E. Roth, J. Qiu, Supporting Social Data Observatory with Customizable Index Structures on HBase - Architecture and Performance, book chapter for “Cloud Computing for Data Intensive Applications”, Springer publishers, 2013.