Comparison of Cloud Database: Amazon's SimpleDB and Google's Bigtable

Ms. Shalini Ramanathan,
Assistant Professor(Senior Grade),
Department of CSE,
P.S.G. College of Technology,
Tamil Nadu, India
sha.cse@gapps.psgtech.ac.in

Dr.Savita Goel
Senior System Programmer,
Computer Services Centre,
IIT Delhi, New Delhi, India.
savita@cc.iitd.ac.in

Mr.Subramanian Alagumalai,
PG Student,
Department of CSE,
P.S.G. College of Technology,
Tamil Nadu, India.
subbu230287@gmail.com

Abstract— **Cloud Computing can be defined as a service or a platform, or an operating system over the Internet to perform tasks. Database has become a part and parcel of life and is being used in almost every computer application. As it is considered the most basic thing, Cloud computing offers this database service too. There are different cloud providers or platforms like Amazon, Google, Microsoft, and many more available in the market. Every cloud platform provides a database for the developers and each one of them has their own merits and demerits. In this paper the characteristics, architectures, advantages of Amazon's SimpleDB and Google's Big Table database are analyzed and discussed in detail. From the comparison of these databases, users can better understand the different cloud database and more reasonably choose what they want.**

Index Terms—**: Cloud Computing, Database.**

I. INTRODUCTION

Practically keeping a conventional relational database management system (RDMS) operating even on a small scale is a very significant job. At least one person needs to take responsibility of monitoring the database, taking backups and installing updates. When working on a large scale, it is obvious to assign a team for taking up the responsibility of clustering and replicating a database to ensure the scalability appropriately.

These require a worth mentioning investment both economically and in terms of workforce. Also, this investment must be made straight. Hiring Operational teams (Ops) is a must and required hardware to meet the demands should be purchased. Due to the inaccurate data in hand, organizations usually opt to over purchase the hardware and over allocate the resources. Yet, every software application should store and query for data, and there are lot of potential difficulties in operating a RDBMS which every organization should tolerate. These are the financial and organizational hurdles to be dealt with innovation and keeping in mind about going the extra mile to add value for the end customer.

Amazon's SimpleDB [2], a hosted cloud-based web service, comes up with an alternative to the conventional relational databases. It follows a streamlined approach by providing only the core functionality for storing and querying data for all complex and ambiguous operations frequently found in a traditional database system. And it's being XML based; we can store data quickly and retrieve or edit them through a simple set of web service API calls via any modern programming language and platform.

II. AMAZON'S SIMPLE DB

Amazon's SimpleDB is a web service that can provide you the core database features like speedy, real time lookup and querying of structured data. Amazon's SimpleDB influences Amazon's cloud infrastructure to provide high availability, scalability and fault-resistant data store a highly available, scalable, fault-tolerant data store to developers.

Amazon's SimpleDB is considered as the most valuable data storage solution for building new web applications, since it routinely carries out many day to day jobs associated with management and scalability of database. It removes the levy on the Developers to think about data modeling, maintaining index and performance tuning (done routinely), or data manipulation (also done automatically).

It is also an ideal solution for exiting applications. Utilizing an absolutely measurable data storage solution like Amazon's SimpleDB can guarantee a new life to existing applications. We can also refactor huge applications using cloud based data store thereby increasing the scalability, minimizing operational expenses and considerably reducing database administration time.

A. Architecture

Coming to SimpleDB's architecture, it possesses the capability for improving your data model on a dynamic basis which renders it as a great match for any agile development. Referencing in Amazon's SimpleDB is the same always, no matter that the web application aspect to simple customer

978-1-4577-0792-6/11/$26.00 ©2011 IEEE

exists in a local datacentre, in Amazon Web Service, or with some third party hosting provider. Just an internet access is enough to establish communication with Amazon's SimpleDB.

As database exists within the Amazon Web Services cloud itself, SimpleDB eliminates the cost and difficulty to maintain an in-house solution. The above said factor paves way for you to concentrate on adding unique value to the application rather than planning on commodities for a database management. With Amazon's cloud, Amazon's SimpleDB scalability grows up and down automatically to meet incoming traffic.

B. Advantage

It uses the whole spectrum of Amazon's highly accessible data centers, thus data stored in SimpleDB ensures dispersion of data geographically and routine data manipulation. These factors ensure the availability and stability of your data. You can always expect flexibility in support for your present and future works.

Even a minute change in schema can cascade many features of your software development effort in a traditional relational database. But Amazon's SimpleDB provides you more flexibility and extensibility with their attribute-based system. As and when there is a change in attributes, the data will be automatically indexed by the system accordingly. This is possible because Amazon's SimpleDB does not need pre-defined schemas. The capacity in storing structured data without defining a schema prior to that reduces the necessity to refactor your database when your applications evolve.

SimpleDB [3] covers about 80% of all database necessities. Lengthy queries which runs for a long time and uses complicated table joins (as used in data warehouse applications) does not fit best for SimpleDB applications. Even if RDBMS provides profound functionality, it introduces more cost complexity than required.
SimpleDB provides availability, stability, and scalability.
Replication of stored data is done many times in different data centers distributed geographically which reduces the hurdle of backing up customer databases. Even if one cluster is not available it switches over to another available cluster based on availability. Requests can be handled through https for encryption.

C. Limitation

- No guaranteed data integrity.
- Inconsistency can offer a terrible user experience.
- Collective operations will require more code.
- Complex reports, and ad hoc queries, will need excessive coding.
- Aggregate operations will be comparatively much slower if RDBMS is not used.
- Importing and exporting data and backup will be slow and complex.
- SimpleDB is not that quick.
- Relational databases are measurable, even with huge datasets.
- Super-scalability is overestimated. Slowing the pace of the Product development is even worse.
- SimpleDB is significant only in certain contexts.

III. GOOGLE'S BIGTABLE

A Bigtable [1] is a light, scattered, constant multidimensional sorted map. Indexing of the map is done by a row key, column key, and a timestamp. In Bigtable, un-interpreted arrays of bytes are used as values. Bigtable stores structured data. Any type of data from text to serialized objects can be stored by applications. It does not impose any size constraint for each value. A table is allowed to have limitless number of columns .Data is indexed using row and column names that can be arbitrary strings.

A. Architecture

Bigtable [4] has been designed to scale into the petabyte range across hundreds or even thousands of computers, and also to ease the addition of more machines without much reconfiguration, thereby making the fullest use of the resources.

Bigtable is built on top of the Google File System, Chubby and stored in an immutable data structure called SSTable which facilitates the storage of log and data files. Chubby is used by Bigtable to store the root tablet, schema details, access control lists, coordinate and identify tablet servers.

The application is capable of defining the number of entries based on the timestamp kept. Also, the application can also decide on the duration of entries to be kept. Unused data will also be cleaned up by the Bigtable by removing the SST tables with unwanted data using Mark-And-Sweep algorithm. Many traditional views on file systems and databases have been eradicated with the design of GFS and Big table, keeping big performance and availability into account.

B. Advantage

Big table was designed to maintain chronological queries and response time is far better for a query when compared to RDBMS. Conventional querying approaches like joins and normalization methodology used in RDBMS are not required here. Data compression is easier because of sparse rows.
The primary technique used by the Bigtable is Compaction, which is of two categories called minor compaction and major compaction. In the former the memtable is transformed into an SSTable whereas the latter saves output into only one SSTable. Minor compaction uses less memory resources comparatively and also reduces the log traffic on restart.

C. Limitation

- It is not an open source database.
- Does not support final consistency.
- Capability of queries is limited.
- Inadequate access control.
- Requires adaptation to the Bigtable approach for application writing.
- Demands manual query programming as Structure query language is not supported by Bigtable.
- No support for ACID transactions as used in RDBMS.

I V. COMPARISON OF SIMPLEDB AND BIGTABLE

Amazon's SimpleDB and Google Bigtable arrange application data into tables. A table is organized as a set of data objects with unique primary keys. The data objects are illustrated by attribute value pairs. Every attribute value is typed as strings. Attributes can vary for data objects within the same table. Data objects are accessed through primary key. Multi table operations like join queries are not supported. Bigtable cannot assure data consistency as it can support transactions only over a single data item.

SimpleDB arranges application data into a many domains, where each domain can only maintain a restricted quantity of update workload. It does not force a predefined schema for its tables. Bigtable arranges attributes into predefined column families. Accessing an attribute is done by including the corresponding column family name as its prefix. Multiple table operations such as join queries are not supported by any of them. Range queries inside a table is supported by SimpleDB with its definite language whereas Bigtable provides similar feature with table scanning using several filtering conditions or predicates. Tomcat v5.5.20 is used as an application by SimpleDB in the Amazon Cloud. SimpleDB supports multiple values per attribute of a data object, while Bigtable allows only single value. Multi-versions with timestamp are not supported by SimpleDB.

SimpleDB provides eventual consistency so that applications may read stale data, so it is impossible to guarantee the visibility of certain writes in the next read. In contrast, Bigtable supports single row transactions, so it can guarantee returning the latest updates.

TABLE I
COMPARISON TABLE OF SIMPLEDB AND BIGTABLE

CHARACTERISTICS	BIGTABLE	SIMPLEDB
Programming language	C++, Python.	Erlang.
Data Item	Multi-version with time stamp	Multi-value attribute
Schema	Column-families	No schema
Operation	Single-table scan with various filtering conditions.	Range queries on arbitrary attributes of a table.
Consistency	Single-row transaction	Eventual consistency
Scalability	Highly Scalable.	Comparatively less Scalable.
Purpose	Designed to scale massive amount of data.	Designed to scale massive amount of data .
Database model	Column-oriented.	Domain based.
Data storage	Distributed storage of structured data.	Centralized storage of structured data.
Dimension	Single Dimension.	Multi Dimension.
Integrity	Problem of Referential Integrity.	Data integrity is not guaranteed.
Type	Reconfiguration is automatic.	No need of reconfiguration.
Usage	User doesn't need to learn any syntax and it is user friendly.	User need to learn syntax and provides inconsistency.
Cost	Structured data storage on Bigtable less than in SimpleDB.	Structured data storage on SimpleDB costs more than in Bigtable.
Features	Data Import, Export and back up are fast.	Data Import, Export and back up are comparatively slow

V. CONCLUSION

Perfect execution of many Web applications requires rigid data consistency. Although the properties of the Cloud like high scalability and availability make it an excellent platform to host Web content, measurable cloud database services offer only weak consistency properties comparatively. The application needs consistency suitable for SimpleDB. Bigtable is the best fit for scalable and suitable storage of huge data. This paper deals with Google's Bigtable and Amazon's SimpleDB services and offers detailed guidance to choose a database based on users' needs.

REFERENCES

[1] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Bigtable: A distributed storage system for structured data, on Seventh Symposium on Operating System Design and Implementation, 2006.

[2] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Amazon's highly available key value store, on Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles, 2007 .

[3] Cloud Architectures and Best Practices of Amazon S3, EC2, SimpleDB, SQS, on white paper, 2008.

[4] Amazon.com, http://aws.amazon.com/simpledb.

[5] Google.com, http://highscalaibility.com.

Ms R.Shalini received the Bachelor's degree in Computer Science and Engineering from Park College of Engineering and Technology, Coimbatore and Master's degree in Computer Science and Engineering from Government College of Technology, Coimbatore, Tamil Nadu, India, in 2005.She is currently working as an Assistant Professor in Department of Computer Science and Engineering at PSG College of Technology, Coimbatore, India. Her research interests include bio-informatics, data mining, scientific knowledge discovery, parallel processing, distributed computing and cloud computing. She has published two international Journals -one in Journal of Data Mining and knowledge Engineering and the other in journal of Bio-informatics, a paper in International Conference and three papers in National conference. She received best project award from Park College of Engineering and Technology in 2003. She is a member of Indian Society for Technical Education.

Dr Savita Goel received the Bachelor's in 1974 and Master's degree in 1976 in Mathematics from Delhi University, India. In 1977, did her DIIT from IIT Delhi. She is a working in Computer Services Centre, IIT Delhi since 1979. Currently designated as Sr. System Programmer in CSC IITD. In 2007, completed Ph. D. in Parallel Processing of NWP model from IIT Delhi. She has published two papers in International Journals, two papers in International conferences and one in national conference. Currently teaching CAD to the M. Des. Students of IIT and programming languages in the Mathematics department. She has received two National Awards for CAD projects. She has a life membership of CSI. Her research interests include Programming, 3D modeling in product designing, parallel processing, distributed computing and Grid computing.

Mr. A. Subramanian received the Bachelor's degree in Computer Science and Engineering from Government college of Engineering, Bargur and Master's degree in Computer Science and Engineering from P.S.G. College of Technology, Coimbatore, and Tamil Nadu.