2012 Fourth International Conference on Computational Intelligence and Communication Networks

A Study on Cloud Database

Deka Ganesh Chandra
DGE&T, MoLE, New Delhi-1, India

Ravi Prakash
Dish TV India Ltd
NOIDA, India

Swati Lamdharia
Airports Authority of India
New Delhi-1, India

***Abstract*—From the data management point of view, cloud computing provides full availability where users can read and write data at any time without ever being blocked. The acknowledgment times are almost stable and do not depend on the amount of associated users, size of the database or any added computing/communication constraints. Furthermore, users are freed for the burden of taking backups. If components fail, it is the responsibility of the service provider to replace them and make the data available using replicas in the meantime. This paper is a study on some of the popular Cloud databases.**

Keyword—NoSQL;Big Data; Cloud Database; Open Source Software.

I. INTRODUCTION

Exponential growth of Internet and the explosion of data sources have raised the problems of storage and usability of data as the usual data management tools were unable to handle the exponentially growing data. Multiplication of data sources and types has lead to the problem of storing and manipulation of these unstructured data by fixed and structured data models provided by Relational Database Management System.

These issues have lead companies and open source communities to build new tools known as NoSQL system or "Key-Value-Store" systems. These systems share a common goal of Massive scale "On Demand" and simplified application Development and Deployment. NoSQL database as the alternatives to traditional database will simplify the roll-out; look for vendors providing supported NoSQL solutions [1]. NoSQL data storage systems are useful for applications that deal with very large semi-structured data. Large database will compel lots of companies to use NoSQL databases [3].

This paper is an analytic study on NoSQL database systems. The rest of the paper is organized into **5** Sections. Section-**2** is a study on **17** Cloud database systems. Section-**3** is an analytic discussion on Cloud database while Section-**4** is observations based on the Section-**2** and Section-**3.** Section-**5** is a brief study on Open source projects while Section-**6** concludes the paper.

II. SOME NOSQL SYSTEMS

This section is a brief survey of cloud Database systems. The databases chosen can be divided into two groups depending on their elasticity level. The first group contains all the databases that are truly elastic, meaning that it is possible to add new nodes into a cluster under load without any observable downtime for the clients. The second group contains all the databases for which there is a significant downtime when new nodes are added into the cluster.

Constant availability of the data when nodes are added or removed from the cluster is made possible by routing mechanisms and algorithms that take decisions to move data chunks are working together i.e. when new nodes are added for example, data that must be moved to new nodes is copied from its existing location to the new one but during all the time of the copy, the data are served by the original location. When the new node has an up to date version of the data, the routing processes start to send requests to this node [24].

There are many new serving databases available. Some of them discussed in this paper are mentioned in the following table.

TABLE I. LIST OF NOSQL DATABASE STUDIED

Sl No	**Name of NoSQL**	Sl No	**Name of NoSQL**
1	PNUTS	10	CouchDB
2	BigTable	11	Voldemort
3	HBase	12	MongoDb
4	Hypertable	13	Infinispan
5	Azure	14	Dynomite
6	Cassandra	15	Redis
7	Xeround	16	ClearDB
8	SimpleDB	17	Google AppEngine Data Store
9	Dynamo		

The systems discussed here are representative, rather than comprehensive. A variety of other systems make different decisions, since it is not possible to survey them all a few interesting characteristics are mentioned in this study.

A. Cassendra

Apache Cassandra database is the right choice for scalability and high availability without compromising performance. Demonstrated fault-tolerance on commodity hardware i.e. cloud infrastructure as well as linear scalability makes it the ideal platform for mission-critical data. Cassandra features permitting the replication across multiple datacenters is the best-in-class, permitting lower latency for survival of regional outages. Cassandra's ColumnFamily information model offers the convenience of column indexes with the performance of log-structured updates, strong support for materialize views and powerful built-in caching.

Netflix, Twitter, Urban Airship, Reddit, Cisco, OpenX, Digg, CloudKick, Ooyala are some of the companies that uses Cassandra to deal with huge, active online interactive datasets. Largest known Cassandra cluster has over 300 TB of information in over 400 machines [8].

978-0-7695-4850-0/12 $26.00 © 2012 IEEE
DOI 10.1109/CICN.2012.35

B. HBase

HBase is an open-source, distributed, versioned, column-oriented store modeled after Google's Bigtable [5]. This is basically a clone of Bigtable providing a real-time, structured database on top of the Hadoop distributed file system. HBase is suitable for application that needs random, realtime read/write access to **BigData**. HBase project's goal is the hosting of very large tables with billions of rows X (by) millions of columns atop clusters of commodity hardware.

HBase provides:

- Linear and modular scalability.
- Strictly consistent reads and writes.
- Automatic and configurable sharding of tables
- Automatic failover support between RegionServers.
- Convenient base classes for backing Hadoop MapReduce jobs with HBase tables.
- Easy to use Java API for client access.
- Block cache and Bloom Filters for real-time queries.
- Query predicate push down via server side Filters
- Extensible jruby-based (JIRB) shell
- Support for exporting metrics via the Hadoop metrics subsystem to files via JMX

C. MongoDB

MongoDB is a scalable, high-performance, open source NoSQL database written in C++. MongoDB is a schema free, document oriented and scalable database. It is fault tolerant, persistent and provides a complex query language as well as an implementation of MapReduce.

MongoDB features [11]:

- Document-oriented storage: JSON-style documents with dynamic schemas offer simplicity and power.
- Full Index Support: Index on any attribute
- Availability: Mirror across LANs and WANs for scale and peace of mind.
- Auto-Sharding: Scale horizontally without compromising functionality.
- Querying: Rich, document-based queries.
- Updates: Atomic modifiers for contention-free performance.
- Reduce: Flexible aggregation and data processing.
- GridFS: Store files of any size without complicating your stack.

D. PNUTS

PNUTS system is a massive-scale, hosted, centrally-managed database system shared by multiple applications to support Yahoo!'s web applications. The focus is on data serving for web applications, rather than complex queries e.g. offline analysis of web crawls [25].

PNUTS provides data management as a service which significantly reduces application development time, since developers do not have to architect and implement their own scalable, reliable data management solutions [12]. Consolidating multiple applications onto single service allows users to amortize operations costs over multiple applications, and apply the same best practices to the data management of many different applications. Moreover, having a shared service allows keeping resources (servers, disks, etc.) in reserve and quickly assigning them to applications experiencing a sudden upsurge in popularity [8] [14].

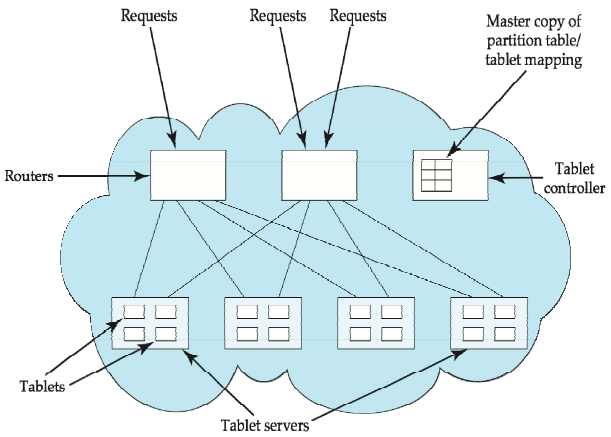

Figure 1. PNUTS Data Storage Architecture

PNUTS always force log updates to disk when committing a transaction, although this log force can be disabled. PNUTS supports Asynchronous replication i.e. wide-area replication without adding significant overhead to the update call itself.

E. BigTable

BigTable maps two arbitrary string values (row key and column key) and timestamp (hence 3 dimensional mapping) into an associated arbitrary byte array. BigTable may be characterizing as a light, distributed multi-dimensional sorted map [16]. BigTable is developed to scale the petabyte range amongst huge number of machines to make it easy to add more machines automatically taking advantage of those resources without any reconfiguration [17]. When sizes threaten to grow beyond a specified limit, the tablets are compressed using the algorithm BMDiff[20] (referenced in [18]) and the Zippy compression algorithm[19] publicly known and open-sourced as Snappy,[13][20] which is a less space-optimal variation of LZ77[21] but more efficient in terms of computing time.

To get a specific row stored in Bigtable, a new client has to connect to all the levels of the tree, but the information obtained on the upper levels are cached, meaning that further requests for data stored in tablets already looked for, will directly be made to the last level of the tree.

F. Hypertable

Hypertable is a high performance distributed open source cloud data storage system designed to support applications requiring maximum performance, scalability and reliability. Hypertable will be acceptable to any organization that needs to administer rapidly evolving data support for online real-time applications.

Modeled after Google's Bigtable project, Hypertable is designed to manage the storage and processing of

information on a large cluster of commodity servers, providing resilience to machine and component failures. Hypertable aims to set the accessible antecedent accepted for awful availability, petabyte scale, database systems [6].

G. Windows Azure

Windows Azure is an open and flexible cloud platform that enables to quickly build, deploy and manage applications across a global network of Microsoft managed datacenters [7]. Windows Azure is an Internet-scale computing service platform hosted in Microsoft data Centre. The Windows Azure platform includes the foundation layer Windows Azure as well as set of developer services which can be used individually or collectively by a group of developers [2].

H. Apache CouchDB

Apache CouchDB is a document-oriented database that can be queried and indexed using JavaScript in a MapReduce fashion. CouchDB also offers incremental replication with bi-directional conflict detection and resolution. CouchDB provides a RESTful JSON API than can be accessed from any environment that allows HTTP requests. There are myriad 3rd party client libraries that make this even easier from programming language of choice. CouchDB's built in Web administration console speaks directly to the database using HTTP requests issued from the browser.

CouchDB is written using Erlang, a robust functional programming language ideal for building concurrent distributed systems. Erlang allows for a flexible design that is easily scalable and readily extensible [9].

I. Voldemort

Voldemort is a fully distributed key-value storage system. Each node is independent with no central point of failure or coordination. Voldemort is designed for use as a simple storage which is fast enough to avoid using a caching layer on top of it. The software architecture is made of several layers, each of them implementing the *put*, *get* and *delete* operations. Each layer is responsible for a specific function like TCP/IP communications, routing or conflict resolution.

Voldemort features are as following:

- Data is automatically replicated over multiple servers.
- Data is automatically partitioned, hence each server contains only a subset of the total data
- Server failure is handled transparently
- Pluggable serialization is supported to allow rich keys and values including lists and tuples with named fields, as well as to integrate with common serialization frameworks like Protocol Buffers, Thrift, Avro and Java Serialization
- Data items are versioned to maximize data integrity in failure scenarios without compromising availability of the system
- Each node is independent of other nodes with no central point of failure or coordination
- Good single node performance i.e. capable of 10-20k operations per second depending on the machines, the network, the disk system and the data replication factor
- Support for pluggable data placement strategies to support things like distribution across data centers that are geographically far apart.

It is used at LinkedIn for certain high-scalability storage problems where simple functional partitioning is not sufficient. It is still a new system which has rough edges, bad error messages and probably plenty of uncaught bugs. The source code is available under the Apache 2.0 license [10].

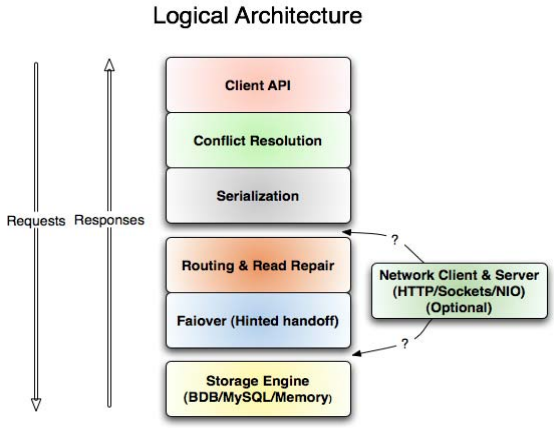

Figure 2. Voldemort Architecture (http://project-voldemort.com /design.php.)

J. Infinispan

Infinispan is an extremely scalable, highly available, open source data grid platform written in Java. The principle of Infinispan is representation of data structure which is highly coexisting, designed ground-up to make the most of up to date multi-processor/multi-core architectures as well as providing distributed cache capabilities. At the core of Infinispan, it exposes a cache interface extending java.util.Map as well as optionally backing a peer-to-peer network architecture to distribute state efficiently around a data grid. Most of the internals are essentially lock-and synchronization-free, favoring state-of-the-art non-blocking algorithms and techniques wherever possible. Even if Non-clustered caching (LOCAL mode) is not the primary goal still Infinispan competitive here [12].

Offering top accessibility by architectonics replicas of accompaniment beyond the arrangement as efficient as optionally constant accompaniment to configurable Cache stores, Infinispan offers intelligent boot algorithms as efficient as JTA compatibility. In accession to the peer-to-peer architectonics of Infinispan on the roadmap is the adeptness to run farms of Infinispan instances as servers and abutting to them application a deluge of audience both accounting in Java as able-bodied as added accepted platforms[12].

K. Dynomite

Dynomite provides integrated storage and distribution, requiring developers to adopt a simple, key/value data model to get the ***Availability*** and ***Scalability*** advantages. By separating these 2 functions, developers can take advantage of the sophisticated distribution and scaling techniques of Dynomite with great flexibility in the choice of data model. [13].

Written in Erlang Dynomite is a consistent distributed key value store NoSQL database system. The design is based on Amazon's Dynamo paper. Dynomite currently implements the following features described in the Dynamo paper, plus some material not covered in this paper.

- Vector clocks
- Merkle trees
- Consistent hashing
- Tunable quorum
- Gossiping of membership
- Gossiped synchronization of partitions
- Pluggable storage engines
- Thrift interface
- Web console with canvas visualizations

L. Redis

Redis is an open source, advanced key-value store. It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets and sorted sets [14].

M. Xeround

Xeround (pronounced zeh-round) offers its own elastic database service based on MySQL. Xeround offers their services through the datacenter of Amazon Web Services located in Europe and North America. Customers can choose whichever location is closest [2]. Xeround's database is host agnostic hence users' will be able to migrate between the Cloud Service Providers' hassles free.

Xeround's database is host agnostic, so customers will be able to migrate freely between providers.

Xeround's two tier architecture is comprised of Access Nodes and Data Nodes. Data Nodes are responsible for storing the data, while Access Nodes receive application requests, communicate with Data Nodes, perform computations and deliver request results. Xeround stores data in virtual partitions that are not bound to the underlying hardware infrastructure. Each partition is replicated to the different Data Nodes located on separate servers, providing high availability and full resiliency. In addition to offering multiple geographic locations Xeround is planning to offer their services to other cloud providers such as GoGrid and Rackspace.

Xeround BASIC:

- Up to 0.5 GB in data size
- Supports up to 40 concurrent connections. Xeround PRO supports up to 4,800 connections by default.
- Max throughput up to 8 MB/s

N. SimpleDB

Amazon Web Services has its own NoSQL cloud database service called SimpleDB which is not only simple but also it could be useful for basic use cases. It is also free for minimal use written using Erlang functional programming language.

O. Google App Engine

Google App Engine (referred to as GAE or App Engine also referred by the acronym GAE/J) is a Platform as a Service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers. It virtualizes applications across multiple servers [22]. App Engine offers automatic scaling for web applications as the number of requests increases for an application, App Engine automatically allocates more resources for the web application to handle the additional demand.[23]. Google App Engine is free up to a certain level of consumed resources. Fees are charged for additional storage, bandwidth or CPU cycles required by the application [3].

P. Dynamo

Amazon Dynamo is used to manage the state of services having very high reliability requirements and need tight control over the tradeoffs between availability, consistency, cost-effectiveness and performance. Dynamo provides a simple primary-key alone interface to accommodated the requirements of these applications.

Dynamo uses an amalgam of able-bodied accepted techniques to accomplish scalability and availability: Data is abstracted and replicated using consistent [27] and consistency is facilitated by object versioning [28]. Consistencies among replicas are maintained during updates by synchronization protocol by a quorum-like technique and decentralized replica. Dynamo employs an account based broadcast abortion apprehension and associates protocol. Dynamo is an absolutely decentralized arrangement with basal charge for minimal administration. Storage nodes can be added and removed from Dynamo automatically without without requiring manual administration or redistribution.

Dynamo has been the fundamental storage technology for a number of the core services in Amazon's e-Commerce platform. It was competent to scale to extreme peak loads efficiently without any downtime during the eventful holiday shopping season. For example, the service that maintains shopping cart (Shopping Cart Service) served millions requests that resulted over 3 million checkouts in a single day and the service that manages session state handled thousands of concurrently active sessions [26].

Q. ClearDB

ClearDB also offers a hosted relational database.

III. DISCUSSION

BigTable sort systems such as Cassandra and HBase attempt to perform sequential I/O for updates as records on disk are never overwritten; instead, updates are written to a buffer in memory and the entire buffer is written sequentially to disk. Multiple updates to the same record may be flushed

at different times to different parts of the disk. The result is that to perform a read of a record, multiple I/Os are needed to retrieve and combine the various updates. Since all writing is sequential hence it is very fast; but reads are correspondingly de-optimized.

Dynomite provides integrated storage and distribution, requiring developers to adopt a simple, key/value data model to get the availability and scalability advantages.

In contrast the traditional buffer-pool architecture PNUTS overwrites records when they are updated. Since updates require random I/O, they are comparatively slower than the BigTable like systems but reads are fast because a single I/O can retrieve the entire latest record. PNUTS always force log updates to disk when committing a transaction, although it can be disabled. HBase does not synchronize log updates to disk, which provides low latency updates and high throughput. This is appropriate for HBase's target use cases, which are primarily to run batch analytics over serving data, rather than to present guaranteed robustness for such data. For such a system, high throughput sequential reads and writes are favored over durability for random updates. Synchronous reproduction ensures freshness of replicas, and is used in HBase and Cassandra. Cassandra also supports asynchronous duplication as do PNUTS. Asynchronous replication supports wide-area replication exclusive of adding significant overhead to the update call itself. In this model, writes are allowed anywhere and conflicting writes to the same object are resolved afterward.

Infinispan is an extremely scalable, highly available, open source data grid platform written in Java. The objective Infinispan is to design highly concurrent data structure to make the best utilization of modern multi-processor as well as multi-core architectures at the same time providing distributed cache capabilities.

Redis is an open source, advanced key-value store. It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets and sorted sets.

MongoDB is a schema free, document oriented and scalable database. It is fault tolerant, persistent and provides a complex query language as well as an implementation of MapReduce. Similarly, Apache CouchDB is a document-oriented database that can be queried and indexed using JavaScript in a MapReduce fashion.

Column storage is beneficial for an application that needs access to a known subset of columns for each request. BigTable, HBase, and Cassandra all provide the ability to declare column groups or families, and add columns to any of them. Each group/family is physically stored separately. On the other hand, if requests typically want the entire row, or arbitrary subsets of it, partitioning that keeps the entire row physically together is best. This can be done with row storage in PNUTS or by using a single column group/family in a column store like Cassandra.

Amazon SimpleDB and Microsoft Azure are hosted cloud serving stores providing transactional functions not found in other serving stores. The caveat is that the user must partition their data into different containers, both in terms of size and request rate. SimpleDB calls these containers domains, whereas Azure defines them databases [4]. Dynamo, Voldemort and Cassandra use eventual consistency to balance replication and availability.

Voldemort is Apache product. BigTable, Hypertable and Google App Engine are GFS compatible database product while PNUTS is Yahoo! Product. Xeround are available from Amazon Web Service. Xeround's database is host agnostic, so customers will be able to migrate freely between providers.

Cassandra allows client to specify on a per-call basis whether the write is durably persisted. App Engine is free up to a certain level of consumed resources. Fees are charged for additional storage, bandwidth, or CPU cycles required by the application. Hypertable is a high performance distributed open source cloud data storage system designed to support applications requiring maximum performance, scalability and reliability. With EC2 user can get root access, but also pay for idle time when no computation is being done. With App Engine users have to pay only for the resources used, so sometimes it's possible to believe user is paying more for a data migration when really it was only "free" on a VPS or EC2 because you already paid for the idle capacity. Azure's model is closer to App Engine's, though there is a baseline cost you must pay even if your application serves 0 requests per billing cycle.

Database.com is based on the aforementioned technology that powers Salesforce.com's flagship CRM service. Salesforce.com is an able-bodied and reliable database provider. As of now Database.com is not accessible, has already in pilot run for decade. It's worthy of consideration for robustness and reliability reason.

IV. OBSERVATIONS

Out of the NoSQL Databases studied in the paper the most expensive is Microsoft Azure and the least expensive Google App Engine. For those familiar with SQL the easiest NoSQL Database is Microsoft Azure while Google App Engine is the hardest (non-relational data store).

- An appliance developer has to fit their workload requirements to the best suited cloud database system considering the read-optimized against write-optimized substitution.
- Latency versus durability is another important axis. If developers know that they can lose small fraction of writes (for example, web poll votes), they can acknowledgment success writes without waiting for them to be synched to disk.

Cassandra allows the applicant to specify on a per-call base whether the address is durably persisted. For such a system, top throughput consecutive reads and writes are advantaged over backbone for accidental updates.

The advancement in NoSQL architecture motivated Yahoo! to advance a criterion for quantitatively evaluating those NoSQL systems The Yahoo! Cloud Servicing Benchmark [25] is the well-known benchmarking framework for NoSQL databases. It currently supports many different databases and it can be extended to use various kinds of workloads.

The salient features of the NoSQL discussed in the paper are as following:

TABLE II. SALIENT FEATURES OF NOSQL STUDIED

Name of NoSQL	Storage type/ Platform	License type	Programming language used
PNUTS	Column	Proprietary	Java/JVM
BigTable	Column	Proprietary	
HBase	Column	Open source	Java
Hypertable	Key-Value	GPL/Open source	C++
Azure	*Key-Value	GPL/Open source	
Cassandra	Column	Open source	Java
Xeround	MySQL based	GPL/Open source	
SimpleDB	Document	Proprietary	Erlang
CouchDB	Document	Open source	Erlang
Voldemort	Key-value	Open source	
MongoDb	Document	Open source/ GPL	C++
Infinispan	Data grid Cloud	Open source	Java
Dynomite	Key-value	Open source	Erlang
Redis	Key-value/tuple	Open source	C
ClearDB		Open source	
Dynamo	Key-value	Open source	Erlang

Source: http://nosql.findthebest.com/, http://nosql-database.org/
*** Row, column & time stamp**

As shown in the table, majority of the NoSQL database are open source software. Out of **17** cloud databases studied in this paper Cassandra, HBase & MongoDB are popular and they can be termed as the representative of NoSQL world.

V. ROLE OF OPEN SOURCE SOFTWARE

The Open Compute Project, which applies the open source software model to the creation of public data centers, is set to announce the formation of a not-for-profit foundation dedicated to furthering the organization's goals, according to New York Times [18].

At the original announcement of the project Facebook engineers detailed the idea's main objective to make data centers more scalable and efficient, examining every part of a facility's operations in order to minimize energy use and lessen environmental impact. Having achieved their goal- the team said that its pilot data center was able to use 38% less electricity to accomplish the same tasks as a standard setup, resulting in a 24% savings on operational costs-the engineers decided to share these innovations with the world in an attempt to broaden its benefits and gain insight from collaborators. The basic idea of open source software creation is central to the project, bringing expertise from a variety of areas together to make the best possible system [15].

One major player conspicuous in its absence from the Open Compute Project is Google, which is highly secretive about its data center designs. This practice is a holdover from the early days of the search engine industry, the Times reported, since computing resources are so vital to the rapid provision of results. The company's specialized, in-house designs for some semiconductors have not even been patented for fear of revealing too much about them, though the firm is also among the biggest buyers of commercial semiconductors in the world. Nevertheless, the Open Compute Projects are gaining momentum. Based on the fruits of the collaborative process it could soon become a real competitor to some of the established powerhouses of the data center industry like HP and IBM.

Open source software, as well, continues to become more prevalent in the enterprise, as database formats like PostgreSQL and Hadoop make inroads and office suites like LibreOffice challenge the dominance of Microsoft. In near future RedHat is likely to transform the world of data storage software the way it revolutionized the market for Unix-based Operating System software. According to a report from **International Data Group** (IDG) News everything from web server software and cloud computing to database formats and desktop platforms will see activity in the open source sector by 2012. The use of multiple cloud infrastructures coupled with the automation goodness that PaaS endows will make service like cloud databases the natural choice that will be preferred over DIY as well as managed ones [15]

VI. CONCLUSION

Data management is becoming critical, given the wide range of storage locations and the plethora of mobile devices. Data has escaped from IT department control into the wider reaches of cloud-based services, mobile devices and social networking. The proliferation and maturity of cloud computing will continue to strengthen and drive the need for reliable cloud database services. The year 2012 will see development and growth of more and more NoSQL Database.

REFERENCES

[1] NICOS VEKIARIDES: Ten Hot Trends in Cloud Data for 2012

[2] http://www.readwriteweb.com/cloud/2011/01/7-cloud-based-database-service.php

[3] Sherif Sakr et al,"A Survey of Large Scale Data Management Approaches in Cloud Environments",IEEE Communications Survey & Tutorials, Vol-13, NO. 3, 3rd Quarter 2011 pp 311-336

[4] Benchmarking Cloud Serving Systems with YCSB by Brian F. Cooper et al, Yahoo! Research, Santa Clara, CA, USA

[5] http://hbase.apache.org/

[6] http://hypertable.org/about.html

[7] http://www.windowsazure.com/en-us/home/tour/overview/

[8] http://cassandra.apache.org/

[9] http://couchdb.apache.org/

[10] http://project-voldemort.com/

[11] http://www.mongodb.org/

[12] http://www.jboss.org/infinispan/

[13] https://github.com/cliffmoon/dynomite/wiki/dynomite-framework

[14] http://xeround.com/blog/2011/12/2012-predictions

[15] http://enterprisedb.com/news-events/news/major-technology-players-look-open-source-model-new-data-center-project,Submitted on Tue, 2011-12-27 14:04,Open Source Software

[16] Fay Chang et al,Bigtable:A Distributed Storage System for Structured Data

[17] Database War Stories#7:Google File System and BigTable,"Bigtable: A Distributed Storage System for Structured Data"

[18] Data compression using long common strings Bentley McIlroy DCC '99 Data Compression Using Long Common Strings

[19] Google's Bigtable

[20] http://blogoscoped.com/archive/2005-10-23-n61.html

[21] http://en.wikipedia.org/wiki/LZ77

[22] http://code.google.com/appengine/docs/python/runtime.html

[23] Sanderson, Dan (2009). Programming Google App Engine: Build and Run Scalable Web Apps on Google's Infrastructure. O'Reilly Media.ISBN 978-0596522728

[24] Study and Comparison of Elastic Cloud Databases: Myth or Reality? Master's thesis submitted for the graduation of Master in Computer Science and Engineering option Artificial Intelligence by Thibault Dory

[25] Benchmarking Cloud Serving Systems with YCSB, by Brian F. Cooper et al, Yahoo! Research, Santa Clara, CA, USA

[26] Dynamo: Amazon's Highly Available Key-value Store by Giuseppe DeCandia et al

[27] Karger D. et al D. 1997. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web. In Proceedings of the 29th Annual ACM Symposium on theory of Computing (El Paso,Texas, United States, May 04-06, 1997). STOC '97. ACM Press, New York, NY, 654-663.

[28] Lamport L, Time, clocks, and the ordering of events in a distributed system. ACM Communications, 21(7), pp. 558-565, 1978.