

## NoSQL Products: IT Giants Perspectives

**Shagufta Praveen\***

*Computer Science Department, Glocal University, Saharanpur, Uttar Pradesh, India*

**Dr. Umesh Chandra**

*Computer Science Department, Glocal University, Saharanpur, Uttar Pradesh, India.*

### Abstract

The immense growth of unstructured data provoked many organizations to think about the future of database. No doubt, NoSQL proved to be a game changer in IT world. Companies like Yahoo, Amazon, Google, linkedIn, Cisco opted for NoSQL. This paper highlights the reason behind selection of NoSQL product. This also concludes that every organization has chosen different NoSQL product due to their diverse objectives. This analysis can bring out the most efficient NoSQL product among selected ones.

**Keyword:** NoSQL Products, IT Giants, Different Perspectives

### 1. INTRODUCTION

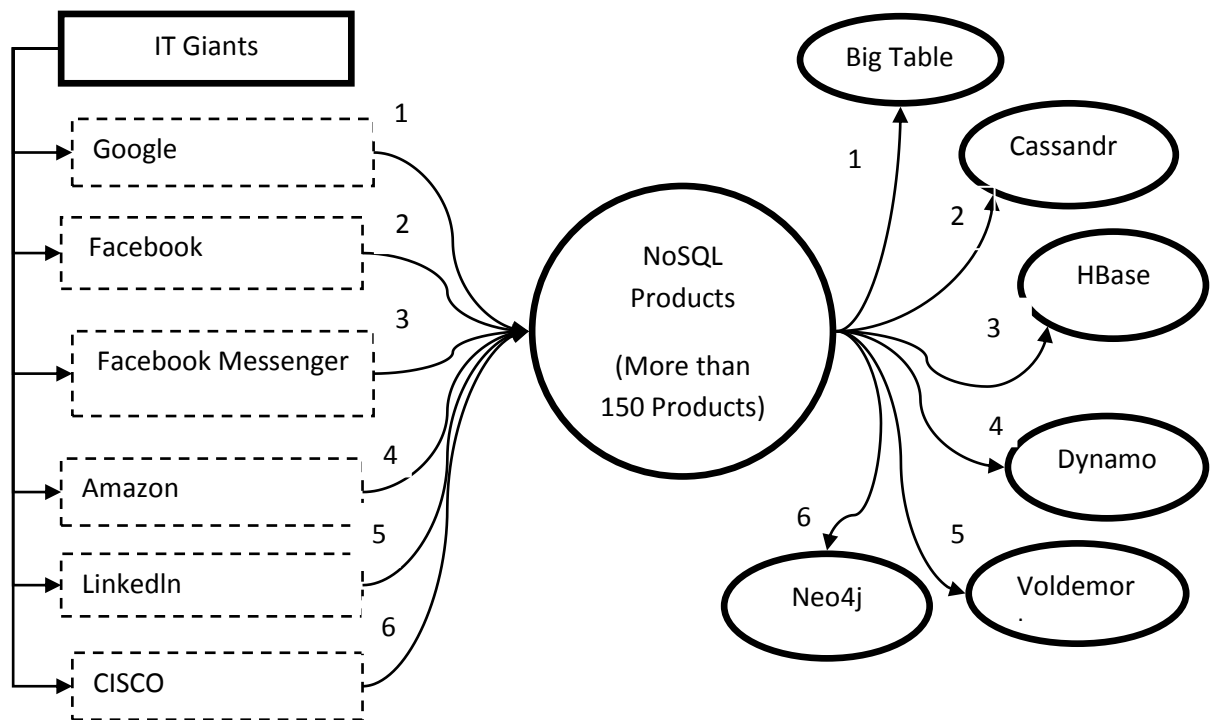
NoSQL is a new emerging topic in the field of database. It does not follow the rules of conventional database whereas it has its own rules to achieve data storage that can solve the scalability issue. It analyzes and deals with large amount of data stored in distributed forms on multiple servers. It is a non-relational, schemaless database based on BASE property [2] and CAP Theorem [3]. As per IDC survey in 2012 [4] Statistics of Data in 2010 has reached to ZettaByte from 130 ExaByte. This proves that every five years data multiplies 10 times. This rise of data is because of Web Based

applications (Facebook, twitter, LinkedIn etc) which have become an essential part of our day to day lives. Social sites and projects (Google map, Indian Railways etc) that deal with massive data and are also aware that their content will rise with time, so to handle the problems related to scalability and storage NoSQL is the only option they have. Due to this many Leading organizations opted for various products of NoSQL.

**2. ORGANIZATION AND NOSQL PRODUCTS**

**Table .1.** NoSQL Products and Family

S.No.	NoSQL Products	Family it belongs
1.	BigTable	Column-Oriented Database
2.	Cassandra	Column-Oriented Database
3.	HBase	Column-Oriented Database
4.	Dynamo	Key-value Store
5.	Voldemort	Key-Value Store
6.	Neo4j	Graph Data Store
7.	MongoDB	Document oriented Database

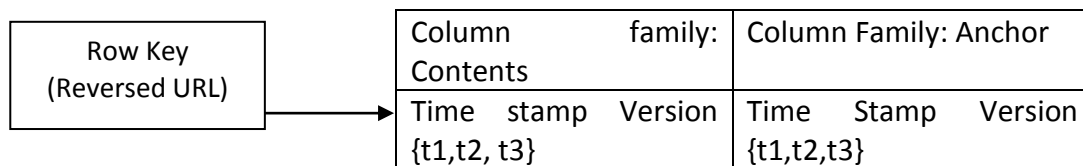


**Fig. 1.** Organization and NoSQL Products

## 2.1. BigTable: Used By Google

Big Table is a NoSQL Product used by Google. Google is one of the renowned and leading organization of the world. More than Sixty products of Google use Bigtable today. Products like Google Analytics, Orkut, Search, Google earth, Gmail, Google Book Search, Google Code and You Tube and many more are the example of products or projects that deal with BigTable.

BigTable is distributed, multidimensional database indexed by row key, column key and time stamp(64 bit , int). Among all three row key is the primary key, collection of column keys are called column family and name of the column family is called column qualifier.



**Fig.2.** Row key and Columns

Row keys in BigTable are arranged in alphabetical order. These are connected to number of column families where every row key has reversed URL. Every column family has number of columns where every cell has different content with different time stamps .In Fig 1. “Column family: Contents” contain the all content of the web pages of the Google as per the row key URL (which is reversed) and Column family: Anchor contains the content of all the link and text of anchors that refer the webpage. This huge data of WebPages called as Web Table.

Single row transaction is only possible in BigTable. It doesn't not provide transactions across multiple row keys that is why consistency across rows cannot be achieved in Big Table. Code of C++ is used to make row mutation. API of BigTable provides functions for making, removing, manipulating tables, clusters, column families. It also support map reduce to process large amount of data in various nodes of cluster together. Clusters help BigTable in Job scheduling and resource management. A single node in a cluster processes 10,000 queries per second.

GFS (Google File system) is the storage part of the Google. All logs and data files are stored in GFS of Big Table.

**Table 2.** BigTable Features

<b>Features</b>	<b>Response</b>
Persistence	yes
Replication	Yes (3 copies are created )
Avalability	Yes
Transaction	Atomic Row update
Implementation	Top of C++ libraries(open source Code)
Reliability	Yes
GFS/Bigtable	GFS-> File System, BigTable-> Database
Structured/ unstructured/semi-structured	GFS stores unstructured data, BigTable stores structured and semi-structured data
MapReduce	Yes
Compactions	Yes(shrink memory of tablet server, reduce amount of data to be read)
Compression	Encode(100-200 MB/s), Decode(400-1000 MB/s)
Cache	Scan Cache, Block Cache [5]
Garbage Collection	Yes(Last version of time stamps are considered)
Schema	Column family schema

## 2.2. Dynamo: Used by Amazon

Dynamo is a NoSQL product that belongs to key-value family. Amazon, one of the famous e-commerce organization uses dynamo as data store. Dynamo is preferred due to its better availability and high scalability. Amazon can't compromise with availability as customer accessing Amazon's website can add items and view different products only when products and products information will be available to them. In order to achieve this scenario dynamo provide primary-key only interface. Replication of data is done to maintain the backup but to achieve consistency between all the replication is another big issue. The update conflict generally arises during write and read operation when customer made an update and changes are not reached by replicas. This may result in poor inconsistency and bad customer experience. Dynamo handles application where updates are not rejected even in network partition or server failure. The core distributed system techniques used in dynamos: partitioning, replication, versioning, membership, failure handling and scaling [6].

**Table.3.** Dynamo Features

Features	Response
Availability	High
Reliability	Yes
Consistency	Sacrifices consistency at some situations
Replication/partitioning	Using Consistent Hashing[14]
Consistency among replica	Achieved by Quorum technique
Built on	Java, Node.js, C# .NET, Perl, PHP, Python, Ruby, Haskell
Network Failure	Read and write operations are possible due to different conflict mechanism
Flexible	No Fix Schema
Cost-effective	\$1 for storing 1GB per month

### 2.3. Cassandra : Used by Facebook

Cassandra is a NoSQL product from column oriented family. It is a distributed storage system that deals with a large amount of data located across many servers. It works on inexpensive hardware, deals with high write throughput without compromising with read efficiency. A highly scalable and reliable data store designed for Inbox Search problem of Facebook. Inbox search data store is support by Cassandra. In June 2008 around 100 million user were using Inbox search whereas today 250 Million users are using it [7].API of Cassandra has three simple methods : Insert(), get(). delete(). A search box presents in the messages tab is to search message content or message sender. It also uses cache mechanism to search data fast. When user searches content on search bar then the actual query executed on the cluster's buffer cache where search results are probably in the memory. This way search in the box becomes faster.

**Table.4.** Cassandra Features

Features	Response
Reliability	Yes
Scalability	Yes
Partition	Consistent Hashing[14]
Schema	Schema-free
Implementation Language	Java
Transactions	Do not support transaction
Store	Structured./unstructured/semi-structured
Developed by FB	For Inbox search
Content	Open Source

#### 2.4. Voldemort: Used by LinkedIn

Voldemort is a distributed system, Big, fault-tolerant hash table. It provides horizontal scalability and high availability which manages multiple data centers using storage system. To achieve good performance and high availability key-value data access is used. Voldemort uses in-cache memory, replication of data occurs automatically over multiple servers and it also handle server failures [8 ]. Voldemort is inspired by Amazon's Dynamo in order to support bulk loading terabytes of read only data [ 9].

**Table.5.** Voldemort Features

Features	Response
Consistency	Tunable consistency (Follows strict quorum & eventual consistency)[15]
Replication, Partitioned	Automatically replicated and partitioned
No Failures	Node Independency
In memory caching	Yes
Read Performance	Good
Scalability	High
Concurrency	High

#### 2.5. HBase: Used by Facebook messenger

The Facebook came up with new social box having features of E-mail, SMS, IM, text messages, on-site Facebook message. It able to store 135 Billion messages a month [11].The current message infrastructure of Facebook handles 350 million users sending over 15 billion person to person messages per month [10]. Message has two kinds of data: temporal, volatile data and ever- growing set of data that rarely get accessed. To fulfill these objectives, people of Facebook thought about Cassandra but they found it difficult pattern to reconcile for their new message infrastructure [10]. Due to this Facebook decided to work with HBase, a scalable and simple consistent model that work over HDFS. It gives very high row level updates over large amount of data.

**Table.6.** Hbase Features

Features	Response
Schema	Schema-less
Scalability	Yes, Horizontal Scalable
Transaction	No Transaction
Data	Semi-Structured, Structured

Failure Support	Automatic failure support
Reliability	Yes
Language Used	Java
Consistency	Immediate Consistency

**2.7. Neo4j:** Neo4j is an open source, NoSQL graph database which is well known in the area of networks and web application. High scalability, availability and better reliability is a cause for use of Neo4j. Delete and update operations is required in order to claim that neo4j is a realistic and strong candidate for replacing relational database [12]. Cisco uses the commercial edition of Neo4j and it is surprised to know that out of 2000, there are more than 20 companies using Neo4j as database [13]. In Neo4j, 4j is describing about java that's make this more robust and secure.

**Table.7.** Neo4jFeatures

Features	Response
Data Model	Flexible/Node-edges
Availability	High
Data	Connected/Semi-structured
Language	Cypher Query language [16]
Content	OpenSource
Database Type	Transactional Database

**Table.8.** NoSQL Products and Summarized Features

NoSQL Products	BigTable	Cassandra	Voldemort	Neo4j	Hbase	Dynamo
Organization Used	Google	Facebook	LinkedIn	CISCO	Facebook Messenger	Amazon
PURPOSE	Google Earth, YouTube, Gmail, Google Maps	Inbox Search	People you May Know	Master Data Management	Messages + email + SMS + Chat	Reliable shopping Cart Services
Data Model	Column-oriented	Column-oriented	Key-value	Graph-Based	Column-oriented	Key-value
Schema	Column-family Schema	Schemaless	Schemaless	Schemaless	Schemas	Schemaless
Availability	High	High	High	High	High	High
Scalability	High	High	High	High	High(Horiz	High

<b>ty</b>					ontal Scalable)	
<b>Language Used</b>	Top of C++ libraries(open source Code	Java	Java	Cyber Query Language	Java	Java, Node.js, C# .NET, Perl, PHP
<b>Consistency</b>	Strong Consistence	Consistent Hashing	Tunable Consistency	Strongly Consistent	Immediate Consistence	Sacrifice consistency at some situation
<b>Data</b>	Semi-structured/structured/unstructured	Structured./unstructured/semi-structured	Semi-Structured, Structured	Conneced/ Semi-structured	Semi-Structured, Structured	Structured./unstructured/semi-structured
<b>Replication</b>	Yes (3 copies are created	Consistent Hashing	Automatically replicated and partitioned	Master – Slave replication, Full graph Replication	Automatically replicated	Using Consistent Hashing
<b>Transaction</b>	Atomic Row update	No Transaction	No Transaction	Transactional Database	No Transaction	Provide TRansaction

## CONCLUSION

A study of the use of NoSQL products by the IT giants namely Google, Amazon, Linked In, Facebook and CISCO shows the use of mainly six products. Using data models, working and internal mechanisms, this paper shall find out the possible reasons for the use. Finally, this paper shall study the prospects of six NoSQL products and suggest which of these is the most efficient for variety purposes.

## REFERENCES

- [1] Lemieux, F., Current and Emerging Trends in Cyber Operations: Policy, Strategy and Practice edited
- [2] Praveen, S., and Chandra, U., 2017, A comparative study on NoSQL, NewSQL and Polygot persistence,
- [3] Gilbert, S., and Lynch, N., " Brewer conjecture and the feasibility of consistent, available and partition-tolerance web services", ACM SIGACT News 33, 2 pp 51-59 March 2002
- [4] Gantz, J., and Reinse, D., The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East[R]. IDC iView, Sponsored by EMC, December 2012
- [5] Bigtable: A Distributed Storage System for Structured Data by Google, Inc., To appear in OSDI 2006
- [6] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Voshall, P., and Vogels, W., Dynamo: Amazon's Highly Available Key-value Store Amazon.com, SOSP'07,



October 14–17, 2007, Stevenson, Washington, USA. Copyright 2007 ACM 978-1-59593-591-5/07/0010

- [7] Lakshman, A., Malik, P., Cassandra – A Decentralized structured storage system
- [8] Chaganti, S. P., Voldemort NoSQL Database, Department of computer science, University of Bridgeport, CT
- [9] sumbaly, R., Kreps, J., Gao, L., Feinberg, A., Soman, C., Shah, S., large-scale batch computed data with project voldemort by, LinkedIn
- [10] BigTable/The Underlying Technology of Messages.html by Kannan is a software engineer at Facebook., Facebook © 2017
- [11] <http://highscalability.com/blog/2010/11/16/facebooks-new-real-time-messaging-system-hbase-to-store-135.html> BY TUESDAY, NOVEMBER 16, 2010 AT 7:52AM
- [12] Melchor, F., Lopez, S., Guillermo, E., de la Cruz, S., 2015, Literature review about Neo4j graph database as a feasible alternative for replacing RDBMS,
- [13] BigTable Neo Technology execs How Neo4j beat Oracle Database\_%20Network%20World.html
- [14] Karger, D., Lehman, E., Leighton, T., Levine, M., Lewin, D., and Panigrahy, R., Consistency Hashing and random Trees. Distributed caching protocols for relieving hot spots on the world wide web. In ACM Symposium on theory of computing pages 654-663, 1997
- [15] Available Online : [ <http://www.project-voldemort.com/voldemort/> ]
- [16] Available Online: [ [https://en.wikipedia.org/wiki/Cypher\\_Query\\_Language](https://en.wikipedia.org/wiki/Cypher_Query_Language) ]

