NoSQL DATABASES: NEW MILLENNIUM DATABASE FOR BIG DATA, BIG USERS, CLOUD COMPUTING AND ITS SECURITY CHALLENGES

Asadulla Khan Zaki

Student, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, India

Abstract

The field of databases has emerged in last decades of years. New architectures try to meet the need to store more and more various kinds of diverse data. The current trend of Big Data(too diverse data, unstructured data, semi-structured data, fast changing data), Big Users(global users 24 hours a day, 365 days a year) and Cloud Computing(new applications use a three-tier internet architecture, run in a public or private cloud) are the driving force for the organizations to migrate towards Non-Relational databases(referring as NoSQL popularly called as "NOT ONLY SQL")from Relational Databases. The main market of Relational Databases is business data processing and these databases are architected to run a single machine and uses a rigid and scheme-based approach to modeling the data and dealing with Big Data and global users on a cloud environment becomes more and more difficult with relational databases. Non-relational databases (NoSQL databases) are considering as new Era database, it provides dynamic schemas, flexible data model, scale-out architecture, efficient big data storage and access requirement. Today the use of NoSQL is mainly due to its Scalability and Performance characteristics. Only a few years ago the Scalability and Performance were not such a big problem but the huge amount of data that is collected today is infinitely much more than ten years ago and also the growth of cloud computing results in large data store even more. This paper includes the introduction, causes of migrating towards NoSQL databases, characteristics, classification of NoSQL databases. Finally the security issues in NoSQL Databases are described and the security enforcement mechanism is proposed.

Keywords: *NoSQL, Big Data, Big Users, Key-value store, RDBMS, Security*

----------***----------

1. INTRODUCTION

Today, there exist many different types of databases, not only the traditional relational databases but several other architectures designed to handle different types of data. Since the 70s the relational model was the dominant, with the implementations like Oracle database, MySQL and Microsoft SQL Servers and almost all databases followed the same basic architecture.

At the beginning of the new millennium, developers started to realize that their data did not fit for the relational model and some of them started to develop other architectures for storing data in databases. When choosing a database today the problem is much more complex to decide the best architecture for data storage and retrieval of data. [1]

Building of applications is now continuously changing.In decade of 90', web companies come up with the scaling features in various dimensions of applications due to the following factors [2]:

- The increase in number of concurrent global users access the applications via web and mobile devices, these users are popularly known as big users.
- The huge volume of data is getting collected and processed today, and it becomes mandatory to collect various kinds of structured and unstructured data and its use became an integral part and it adds richness to applications, popularly known as big data.
- Today, with the emergence of cloud, applications use a three-tier internet architecture that run in a public or private cloud that support big users and big data.

Dealing with big users and big data using relational database technology becomes more and more difficult. The main reason is that relational databases depend on static schemas, and a rigid approach toward modeling the data.

Google, Amazon, Facebook, and Linkedln are among the first companies to discover the serious limitations of relational database technology for supporting big data and big user's requirements. To overcome these limitations, these companies brought up with new data management techniques, their initiatives results in producing a large interest among several developing companies that werefacing the related problems.
As a result, a new database is designed with novel data management model called as NoSQL (populary called as "Not Only SQL"). Today, the NoSQL databases are rapidly growing and deploying in many internet companies and other enterprises. It's gradually considered as a feasiblechoice when compared to relational databases, especially, more organizations identify that, the performance and scalability

requirements of big users and big data on a cloud environment can besuccessfully achieved by using NoSQL databases.

This paper begins with the causes of migrating towards NoSQL databases by introducing big data, big users and cloud. Meanwhile, this paper takes a deeper look on scalability and performance characteristics of NoSQL databases by explaining CAP theorem, and at the end it address the different security issues related to NoSQL databases.

2. MIGRATING TOWARDS NoSQL DATABASES

Out of the many different data-model architectures, the relational data model architecture has been dominating since the 80s, with the implementations like Oracle database[3], MySQL[4] and Microsoft SQL Servers[5]. Laterly, however, the relational databases leads to the problems in many cases because of its data modeling techniques.

The exponential growth of complexity of data generated by social networks, sensors, real time systems, and global users etc, and the storage of this huge amount of data on big distributed system, demands evolution of new data management model [6].

Organizations that collect large amount of unstructured and ever changing data are increasingly turning to non-relational or NoSQL databases [7].

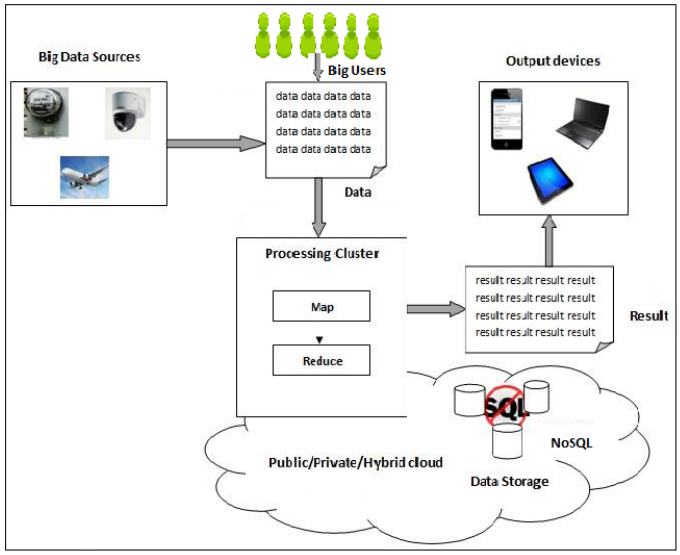

Fig -1: Organizations migrating towards NoSQL database

NoSQL databases focus on analytical processing of large scale datasets in warehouses, offering increased scalability over commodity hardware and servers[8]. Computational and storage requirements of applications such as for Big Data Analytics [9], Business Intelligence [10] and social networking over peta-byte datasets have published SQL-like Centralized database to their limits [11]. This led to the development of non-relational data stores called NoSQL databases which are distributed and horizontally scalable, such as Google's Bigtable[12] and its open source implementation HBase[13] and facebook Cassandra[14]. The emergence of distributed key-value stores, such as Cassandra and Voldemort [15], proves the efficiency and cost effectiveness of their approaches[16]. The limitations with non-relational databases are it is hard to scale with Data warehousing, Grid, Web 2.0 and cloud applications[17].

The strict relational schema of relational databases can be a burden for web applications like blogs, which consists of much different kind of attributes. Text, audios, pictures, videos, real time data and other fast changing information have to be stored within multiple tables. Since such web applications are very agile, underlying database have to be flexible and dynamic as well in order to support easy schema evaluation process [18]. NoSQL systems exhibit the ability to store and index arbitrarily Big Data sets while enabling a large amount of concurrent user requests [8].

Main advantages of NoSQL are the following aspects [20]:

1) Reading and writing data quickly; 2) Support mass storage; 3) Easy to expand; 4) low cost.

2.1Big Data

Capturing and collecting the data becomes easier and can be access via third parties such as D&B, Facebook, and Twitter etc.User related personal information, location dependent data, graph oriented data, user generateddata, system logging data, and real time generated data are just a few examples of the ever-changing and expanding blocks of data being collected. It's not amazing that developers feels the increasing value in leveraging this data to improve existing applications and develops new ones made possible by it. The application of the data is continuously changing the nature of web life that includes web communication, online shopping, web advertisement, entertainment hobbies, and relationship management. The Applications that doesn't meet the current big data market trends will quickly fall behind.

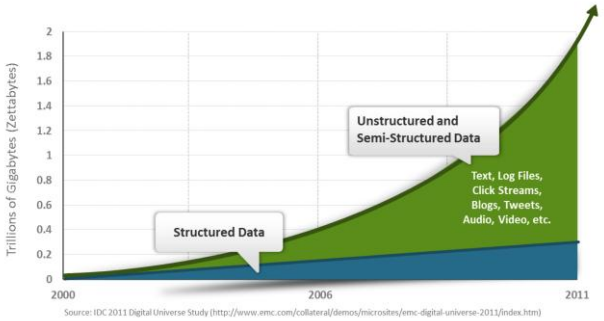

Fig -2: Big Data: The amount of data is growing

rapidly, and the nature of data is changing as well [36]

The various kinds of data is collected and it demands for a very different type of database which should be very flexible and easily incorporate any new type of data. So the database must have a capability of efficiently storing and very fast access to the new types of data that includes semi-structured and unstructured data.

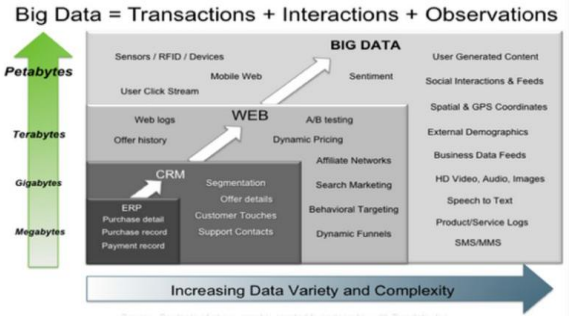

Fig -3: Big Data Transactions with Interactions and Observations [37]

Unfortunately, the relational databases have very poor features to quickly adopt new types of data because of its rigid and static schema based approach, and is not suitable for semi-structured and unstructured data.

Finally, the NoSQL meets the growing trends of storage, processing and retrieval of data by providing a flexible, schema-less data model that maps the organization's requirement and simplifies the communication between the application and database, that results in less writing code, debugging and maintenance becomes more easier.

2.2 Big Users

Not then long ago, one thousand users of an application treated as a lot of users and ten thousand users treated as an extreme case. But today with the emerging field of cloud, many applications are hosted on it and it is made available over the internet 24 hours a day and 365 days a year so that it supports many users globally [2]. A survey shows that more than two billion peoples are connected to worldwide and amount of time they spent online per day is gradually increasing and results in increase number of concurrent users. And now many applications have millions of different daily users.

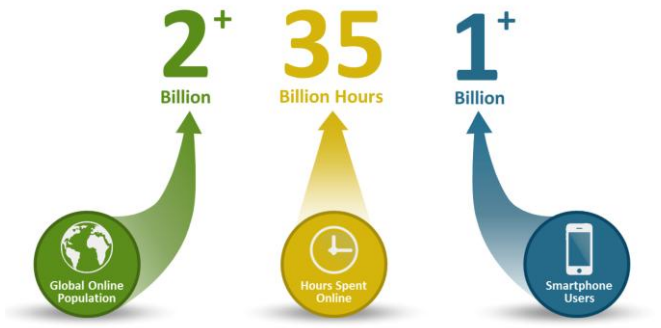

Fig -4: BigUsers:With the growth in global Internet use,the number of hours spent online,and increase in Smartphoneusers,its not uncommon for apps to have millions of users per day [36]

Because of huge number of concurrent users, it is very difficult to predict at application usage requirement. It is very much important that an application dynamically support rapidly growing huge number of concurrent users.

To achieve this goal, an application must possess following features:

- An application can have features that supports zero to millions of users.
- Application must support frequent active global users while considering those users which access application for some time.
- New applications can be dramatically scalable and provide higher fast access process.

The huge number of global users along with dynamic, flexible usage pattern is the driving force for easily scalable new database technology.Many application developers find very much complication to get scalability and faster access rate with relational database technologies. To overcome this limitation of relational database technology many application developers are turning toward NoSQL for help.

2.3 Cloud Computing

Cloud Computing [21] was initially proposed by Google, Amazon and IBM. There are many definitions, and each described cloud computing from a different point of view.
A comprehensive definition [22] is "Cloud computing is a platform (system) or a type of application. In a cloud computing environment, the server can be physical server or virtual server. Cloud computing describes a scalable application which can access through the internet."

Not long ago, most applications are used by single user that runs on a single system and these applications uses a two-tier client server architecture supported by a limited number of users [2].

Today, with the emergence of cloud, applications uses a three-tier internet architecture that run in a public or private cloud that support a huge number of global concurrent users. With this shift in software architecture, cloud provides many data intensive business services like platform-as-a-service, software-as-a-service and infrastructure-as-a-service and these service models have become more prevalent.

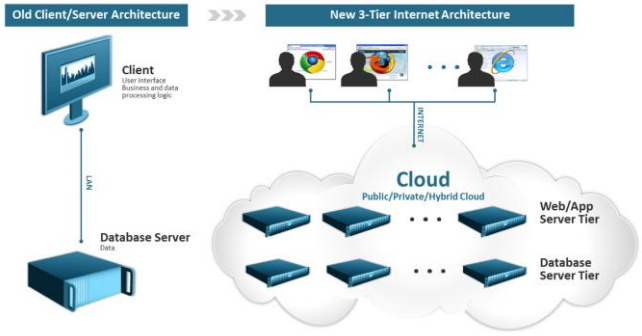

Fig -5: Applications today are increasingly developed using three-tier internet architecture, requiring a horizontally scalable database tier that easily scales with the number of users and amount of data that an application has [36]

In the three-tier architecture, users interact with the applications through a web browser or by using mobile apps that is connected to the internet. In the cloud, scale-out approach is used. If the number of global concurrent user increasesthen another commodity server is added to the web/application tier to manage the incoming traffic and this work will be done by a load balancer very beautifully.

When we compared relational databases and NoSQL databases, relational databases are problematic because they are centralized, share-everything technology and scale-up rather than scale-out.

NoSQL databases are emerged with scale-out approach and better fit for the three-tier internet architecture and cloud services.

4. CHARACTERISTICS OF NOSQL

Eric Brewer [23] introduces the CAP theorem for the shared-data systems. It states that there are three properties of shared-data systems namely data consistency, system availability and tolerance to network partitions. The NoSQL databases primarily classified based on CAP theorem [24] as follows [25]:

- *Availability and Partition tolerance (AP)*
  Such systems ensure availability and partition tolerance primarily by achieving consistency.

  Systems concern AP are Voldemart(Key-value), CouchDB(Document), Riak(Document) and so on…

Fig -6: Characteristics of NoSQL database [38]

- *Consistency and Availability(CA)*
  Here the database mainly uses replication approach to ensure data consistency and availability. And the Part of the database is not concerned about the partition tolerance.

  System concerns the CA are Vertica(column-oriented), GreenPlum(Relational) and so on..

- *Consistency and Partition tolerance (CP)*
  The database ensures the consistency and the data is stored in distributed nodes but database support for availability is not good.

  System concerns the CP are BigTable(Column Oriented), MongoDB(Document), Berkeley DB(Key-value) and so on.

4.1 NoSQL's Performance andScalability

Applications and their underlying databases need to choose either scale-up approach or scale-out approach to deal with the concurrent global users, commonly referred as Big Users.[2]. Scaling-up approachrefers to a centralized architecture in which functionalities are added to existing serversbased on the increase number of global concurrent users and these servers becomes bigger and bigger.

Scaling-out refers to a distributed architecture, instead of adding functionalities to the existing servers the commodity servers are added to meet the requirement of global users.

NoSQL uses scale-out approach on the three-tier internet architecture and worked very well. If more global users use the application, more commodity servers are added to the application/web tier, and performance is achieved by distributing the load on increased number of commodity

servers, and cost depends on number of users. As users increases, cost increases linearly.

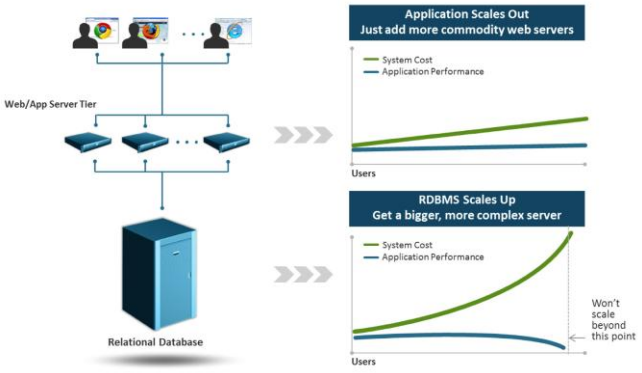

Fig-7(a) **Fig-7(b)**

Fig -7: a. With relational databases,to support more users or store more data,need bigger servers with more CPU's, more memory and more disk storage, **b**. NoSQL databases provide a more linear,scalable approach than relational database [36]

The scale-out approach of NoSQL databases is very much easier. If huge number of users start using application then another commodity server is added very simply. There is no need to modify the application since the application always sees a single (distributed) database.

Along with performance, cost and scalability of NoSQL databases, the flexibility is also equally attractive. As users come and go, commodity servers/virtual machines can be quickly added or removed from the server pool by keeping track of the user population and thus operating cost is also reduced. And, the NoSQL databases are highy fault tolerent databases because the load is distributed across many commodity servers and thus support incontinuous operations.

The advantage of scale-out approach is cheaper than the scale-up approach. in scale-out approach, it is very much expensive to build, design and support the large big server and such server is less fault tolerent when compared to commodity servers. The relational databases are commercially available and these are expensive,need to purchase liscence, whereas NoSQL databases are generally open source, priced based on addition of servers and relatively inexpensive.

4.2 Classification of NoSQL Databases

Based on the data model, NoSQL databases can be classified, some important are listed as follows [26][20]:

4.3 Key Value StoreDatabases

These are the simplest NoSQL databases. It helps developers to build applications with schema-less, unformatted data storage approach, resulting in elimination of fixed data model. Here the data is stored as a key-value pair. The key is associated with every single item in the database and it represents an attribute name together with its value.This type of database support high concurrency, faster execution of queries compared to non relational databases.
Ex:Redis [28], TC and TT.

4.4 Column Oriented Databases

These database stores their data in the form of columns, making it faster read a particular column to memory and making calculations on all values in a column.These are optimized for queries over large datasets, and stores column of data together.

Example: Cassandara [29], Hypertable [30] etc…

4.5 Document Oriented Databases

These databases make use of JSON or XML format to store the values which is then called as document.These databases support complex data structures and helps in easy debugging, conceptualizing data.

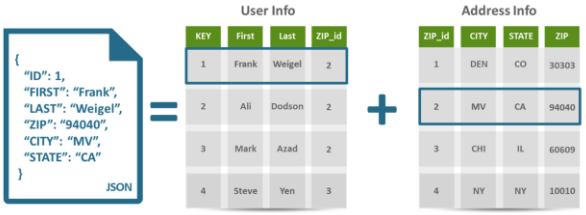

Fig -8: Unlike relational databases which stores and retrieve data from interrelated tables,Document database can store an entire object in a single JSON document, making it faster to retrieve [36]

In comparison with relational databases, document databases support in freely addition of fields to JSON documents, no need to define changes initially. And also these databases support dynamic data that can be changed at any time.

5. SECURITY CHALLENGES IN NoSQL DATABASES

The NoSQL databases emerge with different security issues [33]. The main focus of NoSQL databases is handling the new data sets, with less priority on security [35]. The NoSQL databases are built to meet the requirements of analytical world of big data, and less emphasis on security is given during design stage. NoSQL databases donot provide any feature of embedding security in the database itself. Developers need to impose security in the middleware.

In comparison with the relational databases, NoSQL databases provide a very thin layer of security.

Generally, an external security enforcement mechanism is essential for NoSQL databases.The major security threats of NoSQL databases are listed below [31[32[34]:

5.1 Transactional Integrity

NoSQL databases are failed to ensure transactional integitybecause of its soft nature. Complex integrity constraints cannot be added in NoSQL database architecture because it results in failure to meet the NoSQL's main objective of attaining better performance and scalability.

5.2 Authentication Mechanisms

NoSQL databases are exposing to replay attacks, password brute force attacks, cross-site request forgery, injection attack and man-in-the middle attack results in information leakage. The main reason is NoSQL databases incorporate the weak authentication mechanism and weak password storage techniques. Some NoSQL databases enforce authentication mechanism at local node level, but fail to enforce authentication across all commodity servers.

5.3 Susceptibility to Injection Attacks:

Injection attacks add its own choice of data to the noSQL database results in unavailability and corrupted data. Since NoSQL employs very light weight protocols and loosely coupled mechanism in its architecture that allows an attacker to backdoor access of a file system for malicious activities.

5.4 Lack of Consistency

NoSQL databases does not satisfies simultaneously all the three properties (consistency, availability, and partition fault tolerance) stated by CAP theorem. NoSQL databases make use of many distributed commodity servers, it doesnot assure consistent results at all time, as all participating commodity servers may not entirely synchronized with other servers holding latest information. If a single commodity server gets fail, results in load imbalance among other commodity servers.

5.5 Insider Attacks:

NoSQL databases has poor logging and log analysis methods, due to this an insider attack can gain access to critical data of other users. As NoSQL databases has very thin security layer, it becomes very much difficult for users to maintain control over their data.

6. CONCLUSIONS

Big Users, Big Data, and cloud computing are changing the way that many applications are being developed. The relational databases have dominated industries for many years, but NoSQL databases are now getting attention of application developers due to the following reasons:

- NoSQL databases provides schema-less dyanamic flexible data model, that is most suitable for the big users and big data.
- NoSQL databases have an ability to scale dramatically to support global users and big data.
- NoSQL databases provide an improved performance to satisfy big users expectation without compromising scalability.

To overcome the security issues of NoSQL databases, developers must embed the security mechanism at the middleware along with strengthening the database itself in comparison with the relational databases without compromising the scalability and performance features.

REFERENCES

[1]. Anna Bjorklund, "NoSQL databases for Software Project data". January 18, 2011.
[2]. Find Couchbase from http://www.couchbase.com
[3].Oracle Databases from web: http://www.oracle.com/us/products/database/overview/index.html
[4]. MySQL Databases from web: http://www.mysql.com/
[5]. Microsoft SQL Server Databases from web: http://www.microsoft.com/en-us/sqlserver/default.aspx
[6]. A B M Moniruzzaman and syed Akhtar Hossain, "NoSQL database: New Era of databases for Big data Analytics-Classification, Characteristics and camparision."
[7]. Levih,N(2010). "Will NoSQL databases live up to their promise?" computer43(2), 12-14.
[8].Konstantinou,I.,Angelou, E. Boumpouka,C., Tsoumakos,D., andKoziris,N(2011) October. "On the elasticity of NoSQL databases over cloud management platforms."
[9]. Russom, P. (2011). big data analytics. TDWI Best Practices Report, 4 th Quarter 2011.
[10]. Luhn, H. P. (1958). A business intelligence system. IBM Journal of Research and Development, 2(4), 314-319.
[11]. Abadi, D. J. (2009). Data management in the cloud: Limitations and opportunities. IEEE Data Eng. Bull, 32(1), 3-12.
[12]. Chang, Fay, et al. "Bigtable: A distributed storage system for structured data."ACM Transactions on Computer Systems (TOCS) 26.2 (2008): 4.
[13]. HBase Databases from web: http://hbase.apache.org/
[14]. Lakshman, A., & Malik, P. (2010). Cassandra—A decentralized structured storage.
[15]. http://www.slideshare.net/adorepump/voldemort-nosql

[16]. Use relational DBMS, N. (2009). Saying good-bye to DBMSs, designing effective interfaces.Communications of the ACM, 52(9).
[17]. Padhy, R. P., Patra, M. R., &Satapathy, S. C. (2011). RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Database_s‖. International Journal of Advanced Engineering Science and Technologies, 11(1).
[18]. Hecht, R., &Jablonski, S. (2011, December). NoSQL evaluation: A use case oriented survey. In Cloud and Service Computing (CSC), 2011 International Conference on (pp. 336-341). IEEE.
[20].Jing Han, Haihong E, Guan Le, and jian DU, "Survey on NoSQL Databases". IEEE C 2011,978-1-4577-0208-2/11
[21]. P. Mell and T. Grance, "Draft nist working definition of cloud computing - v15," 21. Aug 2009, 2009.
[22]. Chen Kang, Zheng Weimin Cloud Computing: System Instances and Current Research[J] Journal of Software, Vol.20, No.5, May 2009,pp.1337−1348.
[23]. E. Brewer. (2000, Jun.)Towards robust distributed systems.[Online]. Available: http://www.cs.berkeley.edu/ brewer/cs262b-2004/PODCkeynote.Pdf.
[24]. S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," SIGACT News, vol. 33, pp. 51–59, June 2002. [Online]. Available: http://doi.acm.org/10.1145/564585.564601
[25]. Nathan Hurst, "Visual Guide to NoSQL Systems.", http://blog.nahurst.comlvisual-guide-to-N oSQL-systems/
[26].NishthaJatana, SahilPuri,MehakAhuja,Ishitakathria, DishantGosain, " A survey and comparision of relational and non-Relational Databases". IJERT,Vol 1,Issue 6,2012
[28].Redishttp://redis.io/
[29]. AvinashLakshman, Prashant Malik, "Cassandra-A Structured Storage System on a P2P Network", http://cassandra.apache.org/
[30]. Hypertable, http://hypertable.org/
[31]. Okman, L.; Gal-Oz, N.; Gonen, Y.; Gudes, E.; Abramov, J.; , "Security Issues in NoSQL Databases," Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on , vol., no., pp.541-547, 16-18 Nov. 2011 doi: 10.1109/TrustCom.2011.70
[32]. Neal Leavitt " Will NoSQL Databases Live Up to Their Promise?" IEEE Computer Society 0018-9162/10/$26.00 © 2010 IEEE.
[33]. SriniPenchikala, "Virtual Panel: Security Considerations in Accessing NoSQL Databases", Nov. 2011. http://www.infoq.com/articles/nosql-data-security-virtual-panel.
[34].Find cloud security alliance https://cloudsecurityalliance.org/research/big-data/
[35]. B. Sullivan, "NoSQL, But Even Less Security", 2011. http://blogs.adobe.com/asset/files/2011/04/NoSQL-But-Even-Less-Security.pdf.
[36].Find Source: www.couchbase.com/why-nosql/nosql-database.
[37]. Find Source: http://hortonworks.com/blog/7-key-drivers-for-the-big-data-market/ [41]
[38]. Find Source: nosqltips.blogspot.com

BIOGRAPHIES

Asadulla khan Zaki is pursuing his Master's degree in computer Science and Engineering form BMSCE, Bangalore and received his Bachelor degree in CS&E from BKEC,Basavakalyan. Currently he is working on Security challenges in NoSQL databases.