

# NoSQL Database - Google's Firebase: A Review

Pratik Lahudkar <sup>1</sup>, Sarita Sawale <sup>2</sup>, Vijay Deshmane <sup>3</sup>, Krishna Bharambe <sup>4</sup>

Final year IT Student, Department of Information Technology, Anuradha Engineering College, Chikhli, MH, India<sup>1,3,4</sup>

Associate Professor, Department of Information Technology, Anuradha Engineering College, Chikhli, MH, India <sup>2</sup>

**ABSTRACT:** Nowadays, the number of users, speed of internet and processing power of devices are increasing at a tremendous rate. For maintaining the balance between users and company networks with product or service, our system must evolve and modify to handle the future load of data. Currently, we are using file systems, Database servers and some semi-structured file systems. But all these systems are mostly independent, differ from each other in many except and never on the single roof for easy, effective use. So, to minimize the problems for developing apps, website, game development easier, Google came with the solution as their product Firebase. Firebase is implementing a real-time database, crash reporting, authentication, cloud functions, cloud storage, hosting, test-lab, performance monitoring and analytics on a single system platform for speed, security as well as efficiency. Systems like these are also developed by some big companies like Facebook, IBM, LinkedIn, etc for their personal use. So we can say that Firebase will have the power to handle the future requirement.

**KEYWORDS:** Firebase, NoSQL Database, Cloud Database

## I. INTRODUCTION

The trend for building of applications is now continuously changing day-by-day. In decade of 90's, web companies came up with the scaling features in various dimensions of applications due to the following factors: [1]

- The increase in number of simultaneous global user's access the applications through web and mobile devices, these users are widely known as big users.
- The huge volume of data is getting composed and processed today, and it becomes mandatory to collect a range of structured and unstructured data and its use become an integral part and it adds richness to applications which is popularly known as big data.
- Today, with the materialization of cloud, various applications use a three-tier internet architecture that runs on a public or private cloud that supports big users along with big data.

Production using big users and big data using relational database technology becomes more and more difficult with time. The major reason is that relational databases depend on stationary schemas (Static Schemas), and a rigid approach toward implementing the data.

Google, Amazon, Facebook, and LinkedIn are among the first companies to discover the severe limitations of relational database technology for providing big data and big users' requirements. To overcome these limitations, these companies brought up with new data management techniques, their Initiative resulted in producing a large awareness among several developing companies that were facing the correlated problems.

As an output, a new database is designed with novel data management model called as NoSQL (widely called as - Not Only SQL). Today, the NoSQL databases are quickly rising and deploying in many internet organization and other companies. It is slowly but surely considered as a realistic option when compared to relational databases. Especially, more organizations recognized that, the performance and scalability necessities of big users and big data on a cloud environment can be successfully achieved using NoSQL databases. [2]

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 3, March 2018

## II. MIGRATION TOWARDS NOSQL DATABASES

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

The exponential growth of complexity of data generated by social networks, IoTs, real time systems, global users, and the storage of this huge amount of data on big distributed system require evolution of new data management model.[6] Organizations that collect large amount of shapeless and ever changing data are increasingly turning towards non-relational or NoSQL databases.[7]

NoSQL databases focus on analytical processing of large scale datasets in warehouses, offering enlarged measurability over product hardware and servers [8]. Computational and storage necessities of applications such as Big Data Analytics[9], Business Intelligence[10] and social networking over PETA-byte datasets have introduced SQL-like Centralized database from their limit.[11] This pilot to the improvement of non-relational data stores called NoSQL databases which are distributed and horizontally measurable, such as Google’s Big-table[12] and its open source accomplishment i.e. HBase[13] and facebook Cassandra.[14] The appearance of distributed key-value stores, such as Cassandra and Volde-mort[15], proves the efficiency and cost effectiveness of their approach.[16][17]

The stringent relational schema of relational databases can be a burden for web based applications like blogs, which consists of much unusual kind of attributes. Text, audios, pictures, videos, real time data and other rapidly changing information have to be always stored within multiple tables. Since such web applications are very responsive, principal database have to be flexible and active as well in order to support easy schema evaluation method.[18] NoSQL systems reveals the ability to store and index randomly Big Data sets while enabling a large amount of simultaneous user requests.[8]

## III. CLASSIFICATION OF NOSQL DATABASES

Numerous types of NoSQL database have been produced to execute particular needs and use cases. These databases can broadly be categorized into four types as

- Key-Value store
- Document databases (or store)
- Wide-Column (or Column-Family) store
- Graph databases.

### A. Key-Valued Store

Typically, these DMS store items as alpha-numeric identifiers (keys) and related values in simple, standalone tables (further referred as hash tables). The values may be simple text strings or added intricate lists & sets. Data searches can generally only be implemented against keys not values, and are restricted for exact matches.

Car	
Key	Attributes
1	Make : Nissan Model : Pathfinder Color : Green Year : 2003
2	Make : Nissan Model : Pathfinder Color : Blue Year : 2003 Transmission : Auto

Figure 1 Key/Value Store NoSQL Database [6]

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 3, March 2018

## Primary Use:

The plainness of Key-Value Store makes them ideally matched to lightning-fast, highly-scalable repositioning of the values needed for application tasks like supervision over user profiles or session or to retrieve product names. That's the reason why Amazon makes widespread use of its own K-V system, Dynamo, in its shopping cart. Dynamo is a globally offered key-value storage system that some of Amazon's core services use to supply highly available and measurable distributed data store.

Examples: Key-Value Stores Dynamo (Amazon),  
Voldemort (LinkedIn).

## B. Document Databases -

After getting inspired by Lotus Notes, document databases are introduced, according to their name implies, developed to control and store documents. These documents are encoded in a standard data exchange formats like XML, JSON(JavaScript Option Notation) or BSON (Binary JSON). Distinct from the simple key-value stores described at the top, the value column in document databases include semi structured data-specifically attribute with named or valued pairs. A single column can dwell hundreds of such attributes, and the number and type of attributes recorded can differ from row to row. Also, unlike simple key-value stores, both keys and values are totally traceable in document databases.[6]

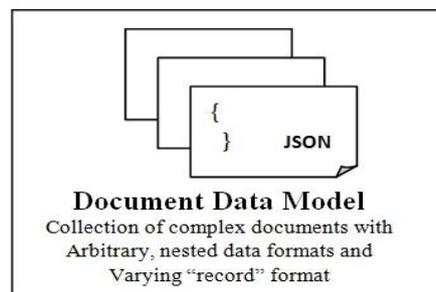


Figure 2 Document Databases [6]

## Primary Use:

Document databases are superior for storing and running Big Data-size collections of factual documents, like text documents, email messages, and XML documents, as well as theoretical documents like collective representations of a database entity such as of a product or customer. They are also good for storing meager data in common, that is to say uneven (semi-structured) data that would involve an extensive use of nulls in an RDBMS (nulls being place holders for omitted or absent values).

Examples: CouchDB (JSON), MongoDB (BSON), MongoDB and CouchDB are open source and they are document dependent and schema free.

## C. Wide-Column (or Column-Family) Stores

Wide-Column stores a circulated, column oriented data structure which accommodates multiple attributes per key. While some WC/CF stores have a Key-Value DNA (e.g., the Dynamo-inspired Cassandra), most are architected after Google's Big-table, the PETA-Byte scale internal distributed data storage mechanism that Google developed for its search index and other products like Google Earth and Google Finance.[6]

## Primary Use :

This type of DMS is great for :

- Distributed data storage, especially versioned data because of WC/CF time-stamping functions.
- Large-scale, batch-oriented data processing: sorting, parsing, conversion (i.e. conversions between hexadecimal, binary and decimal code values), algorithm crunching, etc.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 3, March 2018

## D. Graph Databases -

Graph databases substitute relational tables with structured relational graphs of linked key-value pairings. They are identical to object-oriented databases as the graphs represent as an object-oriented network of nodes (conceptual objects), node relationships (edges) and properties (object attributes shown as key-value pairs). They are the four NoSQL types discussed here that apprehend themselves with affairs, and their focus on image exhibition of information moulds them more user sociable than other NoSQL DBMS.[6]

### Primary uses

In general, graph databases are valuable when you are more concerned in relationships between data than the data itself for example, in representing and working over social networks, generating recommendation (e.g., up-sell or cross-sell suggestions), or conducting forensic investigations (e.g., for pattern detection). Note that these DMS are optimized for relationship working, not for querying. If you want to discover relationships as well as querying and analyzing the values embedded within them, then a search-based DMS is a wiser choice.

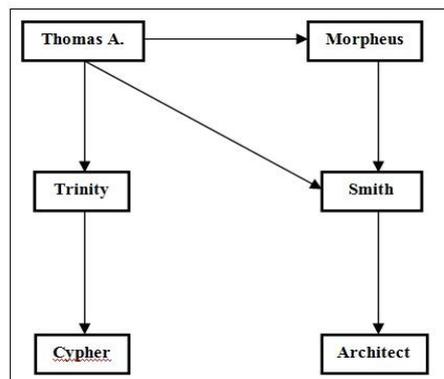


Figure 4. Graph NoSQL Database[6]

## IV. FIREBASE

The Firebase i.e. the Real-time Database is a cloud hosted database. Data is stored as JSON and coordinated in real-time to each linked client.

### A. Structure

In normal database systems, the information is stored in tabular representation. But in this, data or records are called documents and the documents get stored in a binary JSON (BSON) format. Records documents are additionally prearranged in collections. On the further phase, in the identical collection, the schema of one document can be different from the other. Hence, flexible schemas are uplifted. Because developers are familiar with JSON formats, using BSON is much easier. All Firebase Real time Database content is considerably stored as a JSON object. You can assume over the database as a cloud hosted JSON hierarchy tree. Contrasting a SQL database, there are no tables or records. When you add data to the JSON tree, it becomes a node in the offered JSON structure with an allied key.[16]

The Structure of JSON is shown in Fig. 5 below. A JSON database proceeds query results that can be easily parsed, with little or no alteration, directly by JavaScript and most popular programming languages deducing the amount of logic you need to construct into application that you are going to use for your project.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 3, March 2018

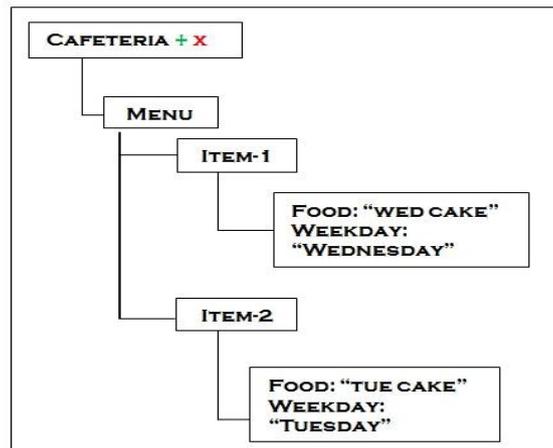


Figure 5 Structure Of NoSQL database

### B. Comparison

There are multiple NoSQL Database systems available. So, for finding the best of them, we have compared them on the basis of using the event type as “value” and then calling the following tests in terms of speed. On an average, Couch DB takes the longest time to execute all of our tests. Uploading and updating large quantity of data requires over twice much time to complete as the next slowest database i.e. Mongo DB. In terms of trying the finest and nastiest case scenario for each database performing each test, Firebase depicts most firm performance out of all databases. The worst case test for firebase is uploading large data which took just over 1000 milliseconds.[21]

Test / DB	MongoDB	DynamoDB	CouchDB	Firebase
Upload Small Data	250	210	470	70
Upload large Data	1200	680	2800	500
Retrieve Small Data	160	150	366	55
Retrieve large Data	740	300	700	540
Update large Data	740	300	540	700
Update Small Data	250	210	520	40

Table : Average time in ms taken to perform each test [21]

### V. FIREBASE PLATFORM

The reason behind efficiency is its platform. Firebase Platform contains the multiple services under one single roof. That means the database don’t have to go over exterior links for storage, validation and Hosting individually. In order to get stability firebase provides the following features that help it to minimize the external work. Firebase provided features are shown below:

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 3, March 2018

## A. Analytics

This feature enables the application developer to recognize the statistics of his application. The SDK capture events and properties on its own and also enables you to acquire tradition data. The console provides details like your most active user or what feature of your application is used the most. It also provides you with summarized data.

## B. Authentication

Authorization quality in firebase lets you allow only the authorized users to use your application. Firebase offers login all the way through Gmail, Github, Twitter, Facebook and also letting the developer to create custom authentication.

i.e. Third-Party Authentication (OAuth)

## C. Storage

Firebase also provides storage capability. It is able to store and retrieve content like images, videos and audio files openly from client SDK. Uploading and downloading is completed in the background part. Data stores are safe and the only official users can access it.

## D. Hosting

Firebase is furthermore used for hosting purposes. Firebase distributes web content rapidly and content is always delivered in a secure way.

## E. Crash reporting

Crash reporting feature on firebase generate reports of errors in the app even after the release. Errors get listed into different groups on the basis of the harshness of the error. You can also produce custom events to trace steps leading towards the crashing of the application.

## VI. ADVANTAGES

NoSQL isn't all build-up. There is significant crackle following this trend. Here are the reasons that make an ideal database:

### A. Made for big data

By design, NoSQL is able of storing, processing, and running tremendous amount of data. This not only indulges the structured data that you build up from your web form or at the point-of-sale but also text messages, word processing documents, videos and other type of unstructured data as well. While RDBMS implementations are raising in terms of what they could hold in capacity, they are largely diminished and outmatched by NoSQL.

### B. Elastic scaling

For many years, DBA's have been dependent on buying bigger servers as database complexity boosted due to distributing the database across multiple hosts when load increases. However, as business costing and availability necessities hiked, and as databases jumped into the cloud or onto a virtual environment, the financial advantages of measuring out on service become pleasing.

RDBMS may not scale out effortlessly on product clusters. But the new diversity of NoSQL database is evolved to enlarge clearly to take benefits of new nodes, and they are usually designed with low-cost utility hardware in consideration.

### C. Economics

NoSQL databases regularly use clusters of low priced product servers to supervise the exploding data and transaction volumes, while RDBMS tends to depend upon costly proprietary servers and storage systems. The result is that the price per gigabyte or dealings per second for NoSQL can be lots of times less than the price for RDBMS, allowing you to store and process more data at a much lower price point.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 3, March 2018

## D. Flexible data models

Change management is a huge headache for large production RDBMS. Even negligible changes to the data model of an RDBMS have to be cautiously managed and may impose downtime or condensed service levels. NoSQL databases have far more relaxed or even non-existent data model limitations. NoSQL Key Value stores and file databases allow the application to store up virtually any structure that it wants in a data element.

It results into changing of application and database representation changes commonly with the application which does not have to be managed as a complicated change unit. In theory, this will allow applications to iterate faster, though, clearly, there can be undesirable side effects if the application fails. [22][23]

## VII. DISADVANTAGES

The assurance from the NoSQL database has created a lot of curiosity, but there are many deficiencies to beat prior to demand the majority enterprises. Here are a few of the toughest challenges.

### A. Maturity

RDBMS architectures have been executed for a long period of time. NoSQL supporters will disagree that their advancing age is a sign of their obsolescence, but for most CIOs, the progress of the RDBMS is cheering. For the most part, RDBMS systems are stable and richly useful.

### B. Support

Enterprises need the guarantee that if an important feature fails, they could be capable enough to get well timed and skilled support. All RDBMS vendors lean towards great lengths to provide a high level of enterprise support.

### C. Analytics and business intelligence

NoSQL databases have inherited to meet the computable requirements of modern Web 2.0 applications. Consequently, most of their feature set is leaning towards the requirement of these applications. However, data in an application has an importance towards the business that goes beyond the “insert-read-update-delete” cycle of a usual Web application. Companies extract information into corporate databases to develop their efficiency and competitiveness, and Business Intelligence is a key that issues for all medium to large companies.

### D. Expertise

There are accurately millions of developers all over the world, and in each business segment, who are familiar with RDBMS concepts and programming. In contrast, almost every NoSQL developer is in a learning mode. This situation will attempt naturally over time. But for now, it is lot easier to find skilled RDBMS programmers or database administrators than a NoSQL expert.[22][23][24]

## VIII. CONCLUSION

NoSQL databases are becoming an increasingly important part of the database landscape. And when used properly, can offer real benefits. However, enterprises must proceed with caution with full alertness of the genuine limitations and issues that are linked with these databases and must implement some smarter ways of database which could be easy to handle like the Google’s Firebase.

## REFERENCES

- [1] Find the Couchbase exclusively from <http://www.couchbase.com>
- [2] Asadulla KhanZak. [NoSQL DATABASES: NEW MILLENNIUM DATABASE FOR BIG DATA, BIG USERS, CLOUD COMPUTING AND ITS SECURITY CHALLENGES].
- [3] Oracle Databases from <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 <http://www.microsoft.com/en-us/sqlserver/default.aspx>



ISSN(Online): 2319-8753  
ISSN (Print): 2347-6710

# International Journal of Innovative Research in Science, Engineering and Technology

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

Visit: [www.ijirset.com](http://www.ijirset.com)

**Vol. 7, Special Issue 3, March 2018**

[6]. A B M Moniruzzaman and syed Akhtar Hossain, [NoSQL database: New Era of databases for Big data Analytics Classification, Characteristics and comparison.]

[7] Levih,N(2010). [Will NoSQL databases live up to their promise? com-puter43(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. ]