Manage Big Data through NewSQL

Conference Paper · April 2014

DOI: 10.13140/2.1.3965.3768

4 authors:

Rakesh Kumar
JECRC Foundation
20 PUBLICATIONS 228 CITATIONS
SEE PROFILE

Neha Gupta
JECRC Foundation
8 PUBLICATIONS 119 CITATIONS
SEE PROFILE

Shilpi Charu
Rajasthan Technical University
10 PUBLICATIONS 132 CITATIONS
SEE PROFILE

Sunil Kumar Jangir
Anand International College of EngineeringJaipur, India -303012
32 PUBLICATIONS 138 CITATIONS
SEE PROFILE

Manage Big Data through NewSQL

Rakesh Kumar[1], Neha Gupta[2], Shilpi Charu[3], Sunil Kumar Jangir[4]

[1,2,3,4]Department of Information Technology & JECRC, Jaipur, India

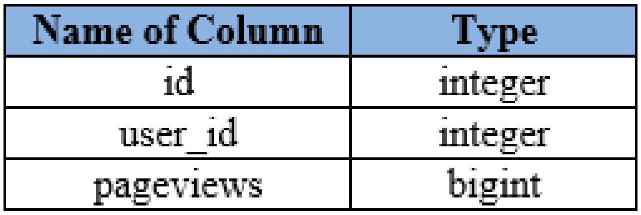
[1]rakeshkumar.it14@jecrc.ac.in; [2]it.nehagupta@gmail.com; [3]shilpicharu@gmail.com;[4]sunil.jangir07@gmail.com

***Abstract* — Bigdata is a horizontally-scaled storage, open-source architecture for indexed data and computing fabric supporting optional transactions, very high concurrency and operates in both a single machine mode and a cluster mode. The bigdata architecture provides a high-performance platform for data-intensive distributed computing, indexing, and high-level query on commodity clusters also run in a high-performance single-server mode. The big advantages of NewSQL databases are non-locking concurrency control mechanism, SQL, ACID support and provides shared-nothing architecture as well as capable of running on a large number of nodes without suffering bottlenecks. In recent trends New SQL comes into existence because of its better Performance and achieves scalability. In this research paper, we focuses on the Paradigm, Data model for Big Data, introduction to Batch layer, Serving layer, Speed layer, Data storage on the batch layer, Incremental batch processing. This paper also provides Lambda architecture in-depth, CouchDB architecture and defines how to handle big data through NewSQL Solutions (dbShards) as well as Future of NewSQL and Big Data processing.**

Index Terms - NoSQL, NewSQL, Big Data, MapReduce, Batch Processing etc.

I. INTRODUCTION

Since the inception of the relational database, Structured Query Language and relational database management systems have been the predominant force in the data management world, running our world's largest business systems. But in this time, the nature of application and data management has experienced significant and accelerating change. The driving forces of this type of shift are the increasing consumer as well as business acceptance, adoption, and proficiency of emerging technologies and paradigms including cloud, mobile devices, smartphones, agile development and social media. What this all types of data translates into is the creation and consumption of a new type of data called Big Data that is fast growing exponentially worldwide. Because the relational database management systems (RDBMS) is architecturally a centralized system, scaling the relational database management systems (RDBMS) to handle the volume, velocity and variety(3V) of Big Data requires employing a vertical scale approach using technologies such as caching, clustering and sharding. Managing big data at such scale, the relational database management systems (RDBMS) eventually hits a wall where it is no give longer feasible solutions.



Name of Column	Type
id	integer
user_id	integer
pageviews	bigint

Figure-1: Relational schema for simple analytics application

In 2009, next generation of databases called Not Only SQL emerged to answer these types of challenges associated with the volume, velocity and variety (3V) of Big Data. Not Only SQL (NoSQL) provides high-performance, highly available, distributed databases were designed to fit the needs of modern day big data structures as well as the cloud for elastic scale and performance. The most powerful features of Not Only SQL (NoSQL) databases are their low latency, linear performance, automatic replication of data, simple management, automatic sharding of data, continuous write availability, ability to deal with unstructured data, use of low-cost commodity nodes, economies of scale, simple scale, and in many cases, the ability to self-provision and self-heal supports. Not Only SQL (NoSQL) has some limitation such as includes a lack of standards, bad analytics performance, no support for indexes, no support for transactions and eventual consistency. NoSQL databases only solve some problem it do not address the limitations associated with the traditional relational database management systems and enterprise applications.

A new type of database systems have introduced to address Big Data as well as the limitations of traditional database systems by merging the capabilities of Structured Query Language and Not Only SQL systems into a technology platform more commonly referred to as NewSQL. NewSQL provides the full functionality of the traditional RDBMS with the scalability as well as performance of NoSQL systems. RDBMS based on a centralized systems architecture, but NewSQL based on a distributed systems architecture. NewSQL give best solution for the enterprise applications. The emergence of NewSQL providers is an indicator of a strong market demand for databases that are going far and beyond limitations of traditional database systems. NewSQL support ability to support OLTP transactions, provide consistency and reliability of a standardized SQL language.

II. PROPERTIES OF BIG DATA SYSTEM

1. Robust and fault-tolerant

Systems need to work correctly in the face of machines going down randomly, duplicated data, the complex semantics of consistency in distributed databases, concurrency, and more. In a production system, someone is going to make a mistake sometime, like by deploying incorrect code that corrupts values in a database.

2. Low latency reads and updates

The very large majority of applications require reads to be satisfied with very low latency, between a few milliseconds to a few hundred milliseconds. The update latency requirements vary a great deal between applications. Some applications require updates to propagate immediately.

3. Scalable

Big Data system supports Scalability properties which provides the ability to maintain performance in the face of increasing data or load by adding resources to the system.

4. General

A general system can support large number of applications. The Lambda Architecture generalizes to applications as financial management systems, social networking, scientific applications as well as social media analytics.

5. Extensible

Big Data system supports Extensible properties which allow functionality to be added with a minimal development cost. When a new feature is added to an existing feature requires a migration of old data into a new format.

6. Allows ad hoc queries

Being able to do ad hoc queries to mine a dataset arbitrarily gives opportunities for business optimization as well as new applications.

7. Minimal maintenance

Minimal Maintenance is most powerful properties of Big Data system which is the work required to keep a system running smoothly.

8. Debuggable

Big Data system supports the information necessary to debug the system when something go wrong. Big Data system also supports a key which is used to trace for each value in the system exactly what caused it to have that value contain.

III. PROBLEMS WITH TRADITIONAL ARCHITECTURE

1. Fault-tolerance is hard

As the number of machines in the backend increases, it became increasingly more likely that a machine would go to down. All the complexity of keeping the application working even under failures has to be managed manually. If your architecture support fault-tolerant: if the master node for a shard is down, you're unable to execute writes to that shard. On the other side if Making writes highly-available is a much more complex problem that your architecture doesn't begin to address.

2. Complexity pushed to application layer

The distributed nature of data is not abstracted away from you. Your application necessary needs to know which shard to look at for each key.

3. Lack of human fault-tolerance

When system gets more and more complex, it becomes more and more likely that a mistake will be made. Human fault-tolerance is not optional but it is essential especially when Big Data adds so many more complexities to building applications.

4. Maintenance is an enormous amount of work

Scaling your sharded database is a big problem such as time-consuming and error-prone.

IV. NOSQL DATABASES

Not Only SQL (NoSQL) is a non-relational database management system, fast information retrieval database and portable. Not Only SQL (NoSQL) databases are those types of databases that are non-relational, distributed in nature, open source as well as having high performance in a linear way that is horizontally scalable.

(A) Types of NoSQL systems

1. Key-Value based storage systems are basically associative with arrays, contains of keys and values. Each key is unique which provide non-ambiguous identification of values.

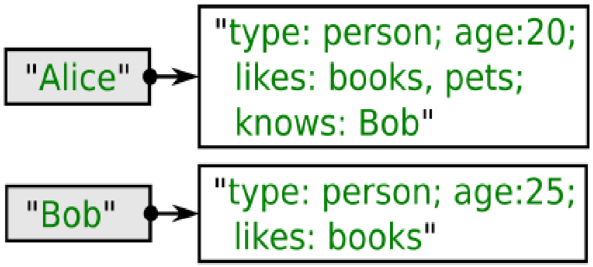


Figure-2: Key Value (KV) databases

2. Wide Column data stores are those types of databases that are used for processing of web, streaming of data and documents. Wide column data store can be seen as a Key-Value store, with a two-dimensional key: A stored value is referenced by a column key as well as a row key.

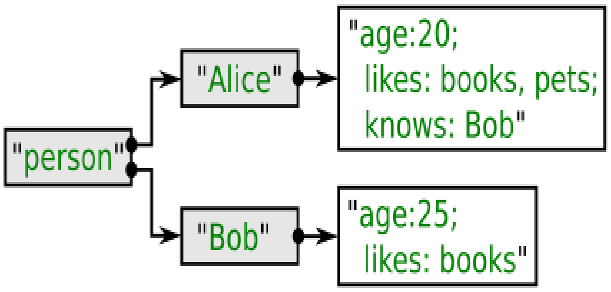


Figure-3: Structure of wide column store

3. Graph databases allow for all queries on the graph structure, such as on relations between nodes or shortest paths.

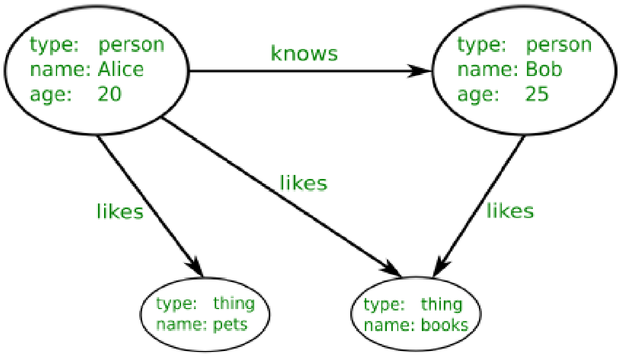


Figure-4: Structure of graph database

4. Document Stores databases are those type of NoSQL databases which use records as documents. Document Stores databases store unstructured or semi-structured documents which are usually hierarchal in nature. Document Stores Databases are schema free as well as not fixed in nature.

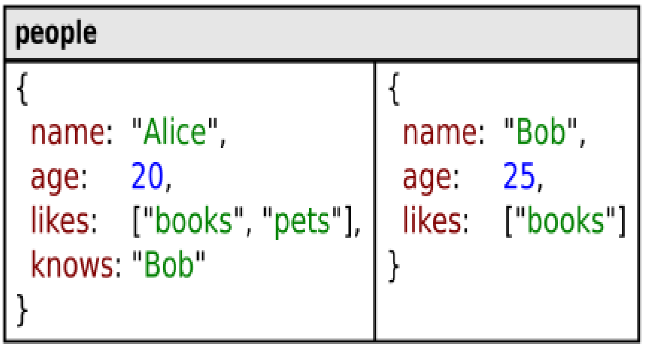


Figure-5: Structure of document stores database

(B) Characteristics of NoSQL

- NoSQL does not use SQL language.
- NoSQL stores huge amount of data.
- In distributed environment, we use NoSQL without any inconsistency.
- If any faults exist in any machine, then there will be no discontinuation of any work.
- NoSQL is open source database.
- NoSQL allows data to store which is not having any fixed schema.
- NoSQL does not support ACID properties.
- NoSQL is horizontally scalable leading to high performance.
- NoSQL is having more flexible structure.

(C) MapReduce

MapReduce is a framework, as well as a programming model, designed to process large amounts of data using user-defined logic.

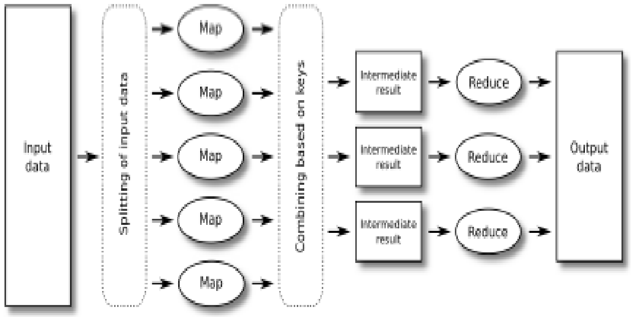


Figure-6: Model of the MapReduce framework

MapReduce framework was to provide an abstraction for processing data, without having to deal with the demands coming with scalability. Input to a MapReduce execution is a dictionary (dataset consisting of key/value pairs) as well as output is again a dictionary. The user has to implement two functions only MAP and REDUCE. MAP function take a key and a value and returns a list of key/value pairs as intermediate result. In MapReduce all input data is split into several chunks, each handled by a separate MAP process.

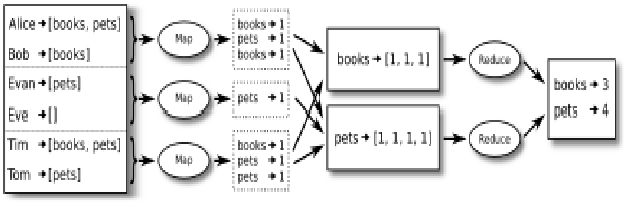


Figure-7: Example of MapReduce

This code are used for MapReduce through MongoDB

```
map = function () {
for (i in this.likes) {
emit (this.likes[i], {count: 1});
}
};
reduce = function (key, values) {
var total = 0;
for (i in values) {
total += values[i].count
}
return {count: total}
};
db.people.mapReduce (map, reduce)
// [{books: {count: 3} },
// {pets: {count: 4} }]
```

NoSQL solve the scalability and flexibility problems of a traditional database, but introduced new type of problems such as lack of ubiquitous access and consistency options, especially for OLTP workload, for schema-less data stores. Hence, this recent time developed a new type of data-management solutions which are also concerns the address large data OLTP concerns, without sacrificing ACID and SQL interfaces. Figure 8 shows the architecture of CouchDB to how to work and store the big data.

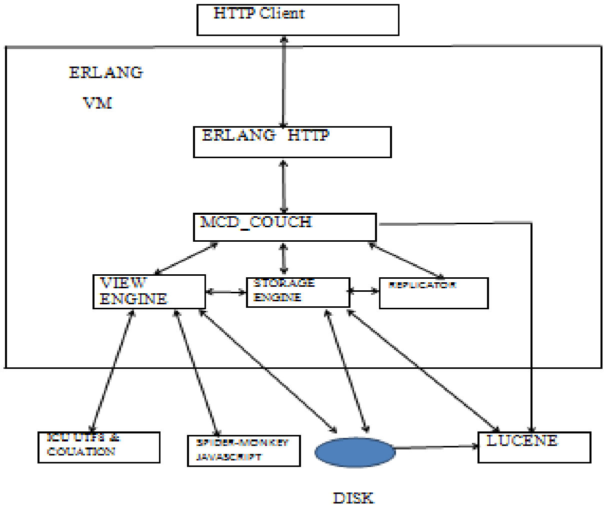


Figure-8: CouchDB architecture

V. LAMBDA ARCHITECTURE

The Lambda Architecture solve the problem of computing arbitrary functions on arbitrary data in real-time by decomposing the problem into three layers such as batch layer, serving layer and speed layer.

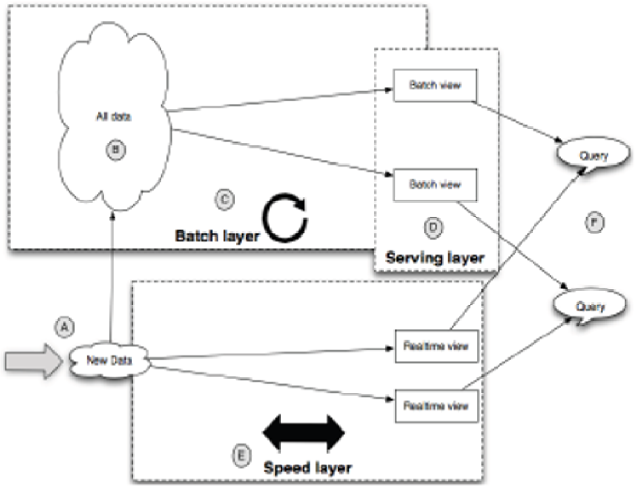


Figure-9: Lambda Architecture Diagram

(A): All new type of data is sent to both the batch layer as well as the speed layer. Through the batch layer, new type of data is appended to the master dataset, but through the speed layer, the new type of data is consumed to do incremental updates of the real-time views.

(B): The master dataset contains properties such as it is immutable, append-only set of data type. The master dataset contains the rawest information or data that is not derived from any other information.

(C): In Lambda Architecture, the batch layer precomputes query functions from scratch. The batch layer outputs are called "batch views." The power of the batch layer is its ability to compute arbitrary functions on arbitrary data, which gives it the power to support any application.

(D): The serving layer is most important layer of Lambda Architecture which indexes the batch views produced by the batch layer and makes it possible to get particular values out of a batch view very quickly. The serving layer is a scalable database that swaps in new batch views.

(E): In Lambda Architecture, contains speed layer which provides facilities for the high latency of updates to the serving layer. Speed layer uses fast incremental algorithms as well as read/write databases to produce real-time views that are always up to date. The speed layer only manage recent type of data, because any data older than that has been absorbed into the batch layer and accounted for in the serving layer.

(F): Queries are solved by getting results from both the batch layer and real-time views and merging them together also.

VI. NEWSQL DATABASE

NewSQL is a different type of relational database management systems that is provide the same scalable performance of NoSQL systems for OLTP workloads and still maintaining the ACID guarantees of a traditional single-node database system.

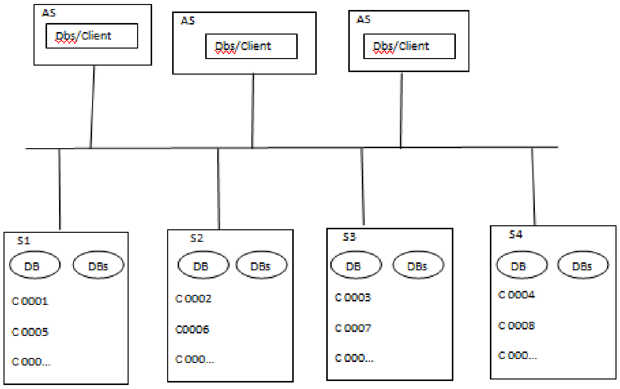


Figure-10: architecture of dbShards (NewSQL solutions)

VII. Important Characteristics of NewSQL Solutions

- NewSQL provides feature SQL as the primary mechanism for application interaction.
- NewSQL support ACID properties for transactions.
- NewSQL controls a non-locking concurrency control mechanism which is helpful for the real-time reads will not conflict with writes.
- NewSQL (dbShards) architecture providing much higher per-node performance than available from traditional RDBMS solutions.
- NewSQL support a scale-out, parallel, shared-nothing architecture, capable of running on a large number of nodes without suffering bottlenecks.
- NewSQL systems are approximately 50 times faster than traditional OLTP RDBMS.

VIII. CONCLUSION AND FUTURE WORK

In this paper shows that what can go wrong when scaling a relational system with traditional techniques like sharding. The problems faced went beyond scaling as the system became more complex to manage. The benefits of data systems built using the Lambda Architecture go beyond just scaling as well as how much more robust your applications made. In this Paper also explore, that there are many other reasons why Big Data applications will be more robust. Although the Lambda Architecture as a whole is generic and flexible which solve the problem of computing arbitrary functions on arbitrary data in real-time by decomposing the problem into several layers.

NewSQL system is a new generation of information management systems, which is apt for businesses that are future planning to:

1. Migrate existing applications to adapt to new trends of data growth rapidly,
2. Develop new and powerful applications on highly scalable OLTP systems.

Hence, New SQL system should be considered as an alternative to NoSQL (Not Only SQL) for New OLTP applications. If New OLTP system is as big and vastly used in market as I foresee, I expect we will see many New SQL engines employing a variety of architectures in the future.

IX. ACKNOWLEDGMENT

This paper work was supported by Jaipur Engineering College and Research Centre (JECRC), Jaipur. We are thankful to Mr. Arpit Agarwal (Director of JECRC, Jaipur) for valuable suggestions, kind support as well as encouragement. Further, also want to convey thanks to Dr. Prof. K. K. Agrawal (IT Department of JECRC, Jaipur) and last but not least all faculty member of JECRC Foundation for their time to time suggestions and technical support.

REFERENCES

[1] Big Data; Principles and best practices of scalable real-time data systems; Nathan Marz, James Warren

[2] An Oracle White Paper in Enterprise Architecture, August 2012; Oracle Information Architecture: An Architect's Guide to Big Data

[3] On the inequality of the 3V's of Big Data Architectural Para-digms: A case for heterogeneity, Todor Ivanov, Nikolaos Korfiatis, Roberto V. Zicari(ivanov@dbis.cs.uni-frankfurt.de; korfiatis@em.uni frankfurt.de; zicari@informatik.uni-frankfurt.de) Frankfurt Big Data Laboratory Chair for Databases and Information Systems Institute for Informatics and Mathematics Goethe University Frankfurt Robert-Mayer-Str. 10, 60325,Bockenheim Frankfurt am Main, Germany http://www.bigdata.uni-frankfurt.de

[4] An Oracle White Paper, June 2013 ;Oracle: Big Data for the Enterprise

[5] Gaining Value From Big Data: Integrating Relational Systems with Hadoop, Colin White, BI Research, May 2013; Sponsored by ParAccel

[6] Bigdatawhitepaper.pdf

[7] Big-Data-Strategy-Issues-Paper1.pdf

[8] History Repeats Itself: Sensible and NonsenSQL Aspects of the NoSQL Hoopla C. Mohan; IBM Almaden Research Center 650 Harry Road San Jose, CA 95120, USA +1 408 927 1733, cmohan@us.ibm.com

[9] Johannes Zollmann "NoSQL Databases"

[10] Katarina Grolinger, Wilson A Higashino, Abhinav Tiwari1 and Miriam AM Capretz, "Data management in cloud environments: NoSQL and NewSQL data stores", Grolinger et al. Journal of Cloud Computing: Advances, Systems and Applications 2013, 2:22; http://www.journalofcloudcomputing.com/content/2/1/22