International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 11
November 2017

Optimization of Databases for Distributed Internet of Things Analytical applications

Akula.V.S.Siva Rama Rao
Research Scholar, Department of CSE ,
Hindustan Institute of Technology and Science

Jinil Persis Devarajan
Associate Professor
Hindustan University, Chennai

Abstract— *One of the major challenge in the domain of distributed internet of things analytics is choosing the appropriate database for the applications, because Internet Of Things sensors devices generate torrents of data every day from billions of devices throughout the world. IoT generated data have three important characteristics which are Volume, Velocity and Variety to process and analyze this data traditional Relational Data Base Management systems does not support, it requires NoSQL database. There are about 150 NoSQL databases available in market among these selecting one of database for distributed internet of things analytical application is a challenge. The CAP theorem (Consistency, Availability and Partition tolerance) evaluates and suggest suitable database for distributed Internet Of Things analytical applications.*

Keywords—NoSQL, CAP, Distributed System, IoT, Analitics

I. INTRODUCTION

Today we find digital data in and around in all fields of the world. The digital data is being generated from various sources such sensors, actuator, RFID, scanned, satellite, aero plane, health, insurance etc. The Internet of Things transmitting data on the network increasing exponentially.[1] The relational database system are sufficient for structured data applications but in the area of IoT, billion of devices generate variety of data ranging from structured, semi-structured, unstructured, image, audio data and video from difference sources of device from different parts of world[2]. Devices generated data not only transmitted but also should be analyzed and provide information. Data analytics is major aspect in the area of Internet of Things. IoT data analytics completely different from stored data in warehouses, IoT analytics is real-time analytics of huge, high-speed and variety data. Example the aircraft A350 model has 6,000 sensors and produces 2.5 TB data per day. Engines are equipped with various sensors use to collect data of every aspect of the operation such as impact of air-pressure, humidity and temperature etc. In aviation industry every element of an aircraft performance is being checked, which could save billions of fuel expenses and improves the safety of the passengers.

The proposed CAP theorem optimizes database to capture, store, retrieve and analyze the huge, high speed and variety data for distributed Internet Of Things Analytical applications[9,11].

II.BACKGROUND

In order to analyze IoT data effectively, it is very important to choose the right database. Choosing an appropriate database for IoT application is the challenge because IoT environment is not same in always. There are many factors plays important role in choosing the right database for IoT applications[4].

Generally scalability, capable of processing huge amounts of data at adequate speeds, flexible schema, portability with various analytical tools, security aspects and costs, apart from these fault-tolerance and high availability feature are required as if any node in the system goes down it must be in position to accepting read and write requests. In distributed database systems servers holding replica copies of data, if one of the server fails then other server storing and respond to the queries till failed server gets up. The distributed computing has been using many years, distributed computing concept become major hurdle for the success for IoT initiatives[5].

Now a days customer-centric services are being performed throughout the world. So there need of high availability systems built withstand failure. For example in smart city project every traffic light equipped with sensors and these must communicate with near by traffic lights. The system would have to employ several nodes around the traffic grid to gather the data and make available to the IoT application. If any one of the node fails then the data gathering and process should be available to the remaining system. The other nodes should feel their data will always be available to the IoT application. Every developed Internet of things system must be robust[5].

The size of analyzing data grows immensely, making it necessary to find more efficient scalable solution than the so far existing Relational Database Management Systems. A as result new methodologies were developed under the Basically Available, Soft-state and Eventual consistency. Brewer proposed CAP theorem, which help to select the appropriate database for distributed system IoT[6].

III.RELATED WORK

In the paper "NoSQL Hands On", Mr. Rebika Rai states that NoSQL encompasses a wide variety of various database technologies were developed in response to the demands in the application like IoT. There is a need to develop to work on applications that create massive volumes, rapidly changing data types like unstructured, semi-structured and multimedia data. This the importance of NoSQL database stressed for unstructured, semi-structure and multimedia data is concern[6].

In paper "Data modeling in the NoSQL world", Mr. Paolo Atzeni supported scalability, performance, and consistency, as needed by future web applications[7].

Mr. Deka Ganesh Chandra discussed on BASE features of NoSQL ie BASE (Basically Available, Soft State, Eventual consistency) analysis of NoSQL. ACID (Atomicity, Consistency, Isolation, and Durability) properties of RDBMS versus BASE in his paper "BASE analysis of NoSQL database"[8,10].

In the paper "A scalable generic transaction model scenario for distributed NoSQL databases", Mr. Ramesh Dharavath Chiranjeev Kumar Presents a scalable three-tier architecture along with a distributed middle-ware protocol to support atomic transactions across heterogeneous NoSQL databases. Their methodology does not compromise on any assumption on the accuracy of failure modalities. Hence, it is suitable for a class of heterogeneous distributed systems[9].

In paper "Relational Databases and "Big Data : The Alternative of a NoSQL Solution " Mr. Jan L. Harrington demonstrated major categories of NoSQL data models, hardware architecture differences between relational and NoSQL databases and NoSQL transaction control ie BASE transactions[10].

IV.PROPOSED METHODOLOGY

As the society moving towards digitization there is an exponential increase of volume, velocity and variety of the data, at the same time there is a need adapt new technologies like NoSQL to store and retrieve data for analytics[12]..

A NoSQL referring to Non-SQL or Non-relational, it is database which provides mechanism to store and retrieval data. NoSQL databases are used in bigdata real-time applications. It is also known as "Not only SQL" as it is used as SQL query language for many applications[6,7].

NoSQL database offer much more flexibility than the traditional relational database. NoSQL can store, process and analyze data including social media, sensor data, online activity, images, location-based information and more. Store, retrieving, process, and analyze all of this unstructured data leads to the development of schema-less database. The term NoSQL encompasses a broad range databases. NoSQL databases document oriented, which can stored in a single document that can be easily found but it not necessarily categorized into fields like a relational database does. NoSQL Database capable of processing huge amount of in the Hadoop platform[6].

NoSQL characteristics includes schema-free, easy replication support, simple API, eventually consistent BASE**,** a huge amount of data[8].

Different categories of NoSQL databases [13]

Wide column store/column

Ex:HBase,MapR, Hortonworks, Cloudera, Cassandra, Scylla, Hypertable, Accumulo, Amazon SimpleDB, Cloudata, MonetDB, HPCC, ApacheFlink, IBM informix Splice Machine, eXtremeDB Financial Edition, ConcourseDB, Druid, KUDU, Elassandra

Document store

Ex: Elastic, ArangoDB, OrientDB, gunDB, MongoDB, Cloud Datastore,Azure, DocumentDB, RethinkDB, Couchbase Server, CouchDB, ToroDB, SequoiaDB, NosDB

Key value/Tuple store

Ex :DynamoDB, Azure Table Storage, Riak,Redis ,Aerospike, LevelDB,RocksDB

Graph Databases

Ex:Neo4J,ArangoDB, OrientDB,Infinite Graph, Sparksee, TITAN, Multimodel, Databases, ArangoDB, OrientDB, Datomic, gunDB, CortexDB, Oracle NOSQL Database

Object Databases

Ex: Vesant,db4o, Objectivity, GemStone/S, Starcounter, Perst, VelocityDB, HSS Database, ZODB, Magma, Smalltalk DB, optimistic locking, Transactions, etc.

Grid and cloud databases

Ex: GridGain, Crate Data , Oracle Coherence, GigaSpaces, GemFire, Infinispan, Queplix

XML Databases

Ex: EMC DocumentumxDB, eXist, Sedna, BaseX, Qizx, Berkeley DB XML

Multidimenstinal databases

Ex:Globals, Intersystems Cache, GT.M, SciDB, MiniM DB, rasdaman

Multivalue databases

Ex: U2, OpenInsight, TigerLogic PICK, Reality, OpenQM, Model 204 Database, Tieto

Event sourcing:

Ex: Event Store, Eventsourcing for Java (es4j)

Time series/streaming databases

Ex :Axibase, Riak TS

BASE: BASE full form is Basically Available, Soft-state and Eventual consistency, ie basic accessibility (Basically Available), flexibility (Soft state), and the final alignment (Eventual consistency). This means that for every request guaranteed to send one of node despite failure, the system state may change, even without the appearance of a data in the system, and that data will be compatible, although there still can be inconsistency[8].

CAP Theorem : With the immense increase, change and implementation of the data in area of data analytics Brewer proposed CAP-Theorem (Consistency, Availability and Partition tolerance), theorem had huge impact on choosing suitable database for the distributed applications[3].

The CAP theorem states that the distributed system is a set of interconnected nodes that share data. According the CAP theorem distributed system has three properties which are Consistency,. Availability and Partition Tolerance.

Consistency **:** A read operation is guaranteed to return the most recent write for a given client.

Availability : A non-failing node should return a reasonable response within a reasonable amount of time.

Partition Tolerance **:** The system will continue to function when network partitions occur.

In the case of failure of node, it is impossible to maintain consistent data on all the replicas on node and still be available for write operations. If it is disallows to write, it gets consistency but availability is compromised because no node can process any write request. If it allows writes, it gets availability, but consistency is compromised because updates will lead to divergence in replicas on node.

The choice is between Consistency and Availability provide two types of systems :

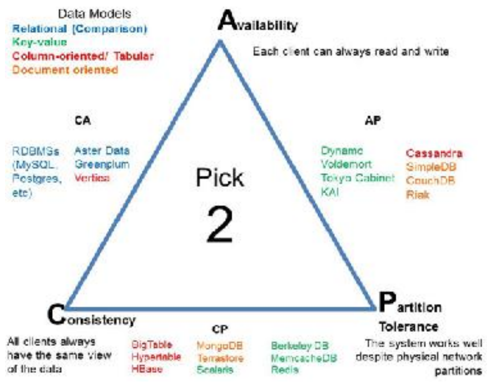


Fig 1: Brewer's CAP Theorem Triangle

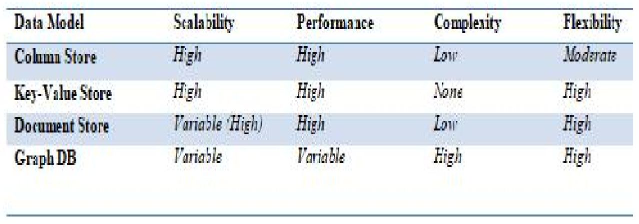
CP (Consistency and Partition Tolerance) Choose Consistency over Availability when the application requirements dictate atomic reads and writes.

Suggested DB : MongoDB, Terrastore, Scalaris, BigTable, Hypertable, Hbase, emcacheDB, BerkelyDB, Redis, *AP (Availability and Partition Tolerance):* Choose Availability over Consistency when the application requirements allow for some flexibility around when the data in the system synchronizes.

Suggested DB : Cassandra, SimpleDB, CouchDB, Riak, Dynamo, Voldemort, Tokyo Cabinet, KAI

.And the third choice CA (Consistency and Availability) Single state cluster, therefore all node are always in the contact, if a partition occurs then the system blocks.

Suggested DB : MySQL, Postgres, Aster Data, Greenplum, Vertica



Data Model	Scalability	Performance	Complexity	Flexibility
Column Store	*High*	*High*	*Low*	*Moderate*
Key-Value Store	*High*	*High*	*None*	*High*
Document Store	*Variable (High)*	*High*	*Low*	*High*
Graph DB	*Variable*	*Variable*	*High*	*High*

Table 1: Evaluation of NoSQL Databases

V.CONCLUSION

This paper helps the developers to choose the suitable NoSQL database from the available NoSQL databases in market for their distributed Internet Of Things Analytical applications.

This paper suggested a few of NoSQL databases to choose from the available about 150 databases, the CAP theorem can extended to remaining NoSQL databases.

VI. REFERENCES

[1] Designing a data management pipeline for pervasive **sensor** communication systems, JussiRonkainen, AnttiIivari 2015 Elsevier

[2] Towards NoSQL-based Data Warehouse Solutions, Zane Bicevskaa, Ivo Oditisa© 2016 Elsevier, ICTE 2016.

[3] A classification of NoSQL data stores based on key design Characteristics Antonios Makrisa, Konstantinos Tserpesa,b,Vassiliki Andronikoub, Dimosthenis, Anagnostopoulosa © 2016 Published by Elsevier .

[4] A Framework for Migrating Relational Datasets to NoSQL, Leonardo Rocha, Fernando Vale, Elder Cirilo, D´arlinton Barbosa, and Fernando Mour˜ao, Elsevier ICCS 2015

[5] IoT Data Provenance Implementation Challenges Adel Alkhalila, Rabie A. Ramadan, © 2014. Published by Elsevier

[6] NoSQL Hands on, Rebika Rai, Praqshant Chettri, . © 2017, Published by Elsevier

[7] Data modeling in the NoSQL world ,Paolo Atzeni, © 2016, Published by Elsevier

[8] BASE analysis of NoSQL databse, Deka Ganesh Chandra, © 2015, Published by Elsevier https:/doi.org/10.1016/j.future.2015.05.003

[9] A scalable generic transaction model scenario for distributed NoSQL databases, Ramesh Dharavath Chiranjeev Kumar, © 2016, Published by Elsevier

[10] Relational Databases and "Big Data : The Alternative of a NoSQL Solution " Mr. Jan L. Harrington, © 2015, Published by Elsevier

[11] Enabling distributed intelligence assisted Future Internet of Things Controller (FITC), Hasibur Rahman, Rahim Rahmani, 2210-8327/_ 2017, Production and hosting by Elsevier . on behalf of King Saud University, (http://creativecommons.org/licenses/by-nc-nd/4.0/).

[12] Industrial Big Data as a result of IoT adoption in Manufacturing, D. Mourtzis, E. Vlachou, N. Milas © 2017 . Published by Elsevier

[13] Persisting big-data:The NoSQL land scape Alejandro Corbellini n, CristianMateos, Alejandro Zunino, Daniela Godoy, Silvia Schiaffino Information Systems63(2017)1–23 2016 Elsevier