MongoDB vs Redis: Critical Analysis and Comparison

Preeti Bhatt*
** D.C.S.A. Panjab University sector-14 Chandigarh*
Preeti.gcg@gmail.com

Abstract- In the past, large number of applications uses relational database reason being having vast set of features, transaction management, concurrency support, and query capabilities. But if the data is big and lots of query processing is required then relational databases fails to store and process big data effectively and then can't perform joins and transaction operations efficiently. After the emergence of non-relational databases, some of these problems of relational database are overcome. In this paper, we describe a document based database named MongDB and a key-store database named Redis. This paper focuses on the comparison and analyses of two popular NoSQL databases: MongoDB and Redis.

Keywords- Non-Relational, NoSQL, MongoDB, Redis, Scalability, Distributive Databases.

I.INTRODUCTION

These days people using hybrid databases respective to their applications requirements. Today's applications demands flexibility in choosing database according to their applications components. In this paper, we are comparing features of two popular NoSQL databases MongoDB [13] and Redis [12] and deciding which database is better at which situation or can we say one among them.

II.MONGODB VS REDIS

2.1. MONGODB:

MongoDB is a NoSQL document-oriented database supports JSON format. It is schema free, open-source and cross-platform. It allow us to access and analyze the data through real time aggregations, indexing and Ad hoc queries. It provides high availability, geographic distribution, and horizontal scaling as a distributive database.

Key customers:

ADP, Adobe, Amadeus, AstraZeneca, Barclays, BBVA, Bond, Bosch, Cisco, CERN, City of Chicago, Department of Veteran Affairs, Department of Works and Pensions, eBay, Epic Games.

Typical application scenarios:

- Internet of Things (Bosch, Silver Spring Networks)
- Mobile (The Weather Channel, ADP, O2)
- Single_View (MetLife)
- Real-Time Analytics (Buzzfeed, City of Chicago, Crittercism)
- Personalization (Expedia, eHarmony, Gilt)
- Catalogs (Under Armour, Otto)

2.2. REDIS:

REmote DIctionary Server (REDIS) is a key-value store database that can be used as a cache or message broker.

Redis, benchmarked as the world's fastest database, reduces application complexity, simplifies development, accelerates time to market and provides unprecedented flexibility to developers with its visionary data structures and modules.

Redis Labs is in-memory multi-model database platform for transactional, analytics and hybrid deployments. It is the home of open source Redis and commercial provider of Redis Enterprise. Redis Enterprise is available, both as a service in major public, private and hybrid clouds, as well as downloadable software.

Key customers:

Verizon, Vodafone, Atlassian, Trip Advisor, Jet.com, Nokia, Samsung, HTC, Docker, Staples, Intuit, Groupon, Shutterfly, KPMG, TD Bank, UnitedHealthcare, RingCentral, The Motley Fool, Bleacher Report, HipChat, Salesforce.

Typical application scenarios

Common use cases include (but are not limited to):

- Caching
- Session management
- High-speed transactions
- Time-series logs
- Message broker
- Real-time analytics
- Data ingestion
- Leaderboards
- Job and queue management

III.RELATED WORK

[1] Perform comparison for streaming applications among qualitative and quantitative features using MongoDB and CouchDB. Amazon S3 [2] as a storage technology provides availability, scaling and reliability by synthesizing technologies. [4] perform comparative analysis of NoSQL databases based on some selected featured and mentioned qualitative evaluation criteria in two different sets named as common installations size measurement and performance measurement. [3] Summarized main features of three databases named BigTable, DynamoDB, and Cassandra and do comparison and contrast between them. [8] Did performance comparison between MongoDB, Redis, couchbase, Cassandra and HBase. The author introduced YCSB [5] benchmarking for the performance comparison of various key-store and document store databases. YCSB++ [7] benchmarking is introduced by the author which do performance debugging against highly scalable and semi-structured table stores. [6] Introduced YCSB as NoSQL performance measurement tool and select an optimal NoSQL database store where YCSB perform simple read write operations on primary key. [9] Perform experimentation and evaluation and tried to access feasibility of non-relational databases on handling tree data structure with heterogeneous nodes and handling high volume of data efficiently that was not achieved by relational databases. [10] Selected MongoDB and couchbase as document store databases and tried to compare time and analyses performance of these databases on inserting and retrieving various size images on databases. [11] Characterized various features of mongoDB and Cassandra and compare and evaluate their operational principles.

IV.COMPARITIVE ANALYSIS OF KEY-VALUE STORE NOSQL DATABASES

Table1: The Comparison and contrast between MongoDB and Redis

	MongoDB	**Redis**
Development language	C++	C
Primary Database Model	Document store	Key-value store
Developer	MongoDB, Inc. (2009)	Salvatore Sanfilippo (developer at Redis Lab) (2009)
Transactions	Multi-document ACID Transactions with snapshot isolation	Optimistic locking, atomic execution of commands blocks and scripts
Replication	Master-slave replication Single master replication	Master-slave replication Multi-master replication
Concurrency Control	Two-Phase Locking (Deadlock Prevention) Optimistic Concurrency Control (OCC)	Not Supported
Checkpoints	Consistent	Non-blocking
Triggers	No	No
CAP Theorem	Consistency, Partition tolerance	Consistency, Partition tolerance
Operating System/Platform	Cross Platform	POSIX Systems

High Availability	No	No
License type	Open Source (Server side public license v1) Commercial license are not available	Open Source (BSD 3-clause) Commercial license are available
Map reduce	yes	No
Consistency	Eventual Consistency Immediate Consistency	Strong eventual consistency with CRDTs Eventual Consistency
Querying	Custom API Complex query support	Using API calls
Data Model	Document / XML	Key/Value
Partitioning scheme	Sharding	Sharding
Scalability	Highly scalable	Liner scalability
API	proprietary protocol using JSON	proprietary protocol using RESP-Redis Serializable Protocol
Storage Architecture	Hybrid	Hybrid
Query Compilation	JIT Compilation	Not Supported
Logging	Physiological Logging	Command Logging
Data Scheme	Scheme free	Scheme free
Foreign keys	Not Supported	Not Supported
Indexes	B+Tree	Hash Table
Joins	Not Supported	Not Supported
Stored Procedures	Not Supported	Not Supported
Views	Virtual Views	Not Supported
Cloud-based only	No	No
User Concept	Access rights for users and roles	Simple password-based access control
Durability	Yes	Yes
Server-side scripts	JavaScript	Lua
SQL Support	Read-only SQL queries via the MongoDB Connector for BI	No
Predefined Data Type Support	String, integer, decimal, double, Boolean, sate, object_id, geospatial	Strings, hashes, lists, sets and sorted sets, bit arrays, hyperloglogs and geospatial indexes
Programming language support	Actionscript info C C# C++ Clojure info ColdFusion info D info Dart info Delphi info Erlang Go info Groovy info Haskell Java JavaScript Lisp info Lua info MatLab info Perl PHP PowerShell info Prolog info Python R info Ruby Scala Smalltalk info	C C# C++ Clojure Crystal D Dart Elixir Erlang Fancy Go Haskell Haxe Java JavaScript (Node.js) Lisp Lua MatLab Objective-C OCaml Pascal Perl PHP Prolog Pure Data Python R Rebol

		Ruby Rust Scala Scheme Smalltalk Swift Tcl Visual Basic

V.CONCLUSION

In this paper we perform comparative analysis of features between MongoDB and Redis NoSQL database. If speed is the primary concern of your applications then MongoDB may not be the best choice. Redis is faster database compared to MongoDB as it is an in-memory database. MongoDB is more flexible compared to Redis, hence it scale better than Redis. Redis have more complex syntax and data storage structure than SQL so it is very difficult to learn and querying. MongoDB allow us to learn new query language but it’s easy to learn compared to Redis. MongoDB is companies only database (single standalone database) but Redis is not companies only database and used when they need more speed in existing database. MongoDB uses document based data storage and can be modified easily and on the other hand Redis uses basic key-value data storage but have support for additional data types. After this analysis I found that, if data structure is not clearly defined and have unstructured and/or structured data then MongoDB is a great choice but if data structure is clearly defined and we need speed and fast access and size of the data is stable then Redis is great choice.

REFERENCES

[1] Kumar KS, Mohanavalli S. A performance comparison of document oriented NoSQL databases. In2017 International Conference on Computer, Communication and Signal Processing (ICCCSP) 2017 Jan 10 (pp. 1-6). IEEE.

[2] DeCandia G, Hastorun D, Jampani M, Kakulapati G, Lakshman A, Pilchin A, Sivasubramanian S, Vosshall P, Vogels W. Dynamo: amazon's highly available key-value store. InACM SIGOPS operating systems review 2007 Oct 14 (Vol. 41, No. 6, pp. 205-220). ACM.

[3] Kalid S, Syed A, Mohammad A, Halgamuge MN. Big-data NoSQL databases: A comparison and analysis of “Big-Table”,“DynamoDB”, and “Cassandra”. In2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(2017 Mar 10 (pp. 89-93). IEEE.

[4] Tudorica BG, Bucur C. A comparison between several NoSQL databases with comments and notes. In2011 RoEduNet international conference 10th edition: Networking in education and research 2011 Jun 23 (pp. 1-5). IEEE.

[5] Cooper BF, Silberstein A, Tam E, Ramakrishnan R, Sears R. Benchmarking cloud serving systems with YCSB. InProceedings of the 1st ACM symposium on Cloud computing 2010 Jun 10 (pp. 143-154). ACM.

[6] Reniers V, Van Landuyt D, Rafique A, Joosen W. On the state of nosql benchmarks. InProceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion 2017 Apr 18 (pp. 107-112). ACM.

[7] Patil S, Polte M, Ren K, Tantisiriroj W, Xiao L, López J, Gibson G, Fuchs A, Rinaldi B. YCSB++: benchmarking and performance debugging advanced features in scalable table stores. InProceedings of the 2nd ACM Symposium on Cloud Computing 2011 Oct 26 (p. 9). ACM.

[8] Tang E, Fan Y. Performance comparison between five NoSQL databases. In2016 7th International Conference on Cloud Computing and Big Data (CCBD) 2016 Nov 16 (pp. 105-109). IEEE

[9] Jayathilake D, Sooriaarachchi C, Gunawardena T, Kulasuriya B, Dayaratne T. A study into the capabilities of NoSQL databases in handling a highly heterogeneous tree. In2012 IEEE 6th International Conference on Information and Automation for Sustainability 2012 Sep 27 (pp. 106-111). IEEE.

[10] Chopade MR, Dhavase NS. Mongodb, couchbase: performance comparison for image dataset. In2017 2nd International Conference for Convergence in Technology (I2CT) 2017 Apr 7 (pp. 255-258). IEEE.

[11] Abramova V, Bernardino J. NoSQL databases: MongoDB vs cassandra. InProceedings of the international C* conference on computer science and software engineering 2013 Jul 10 (pp. 14-22). ACM.

[12] https://redis.io/

[13] https://www.mongodb.com/