

Decryption of Main Types of NoSQL Databases

MongoDB

Found by Dwight Merriman, Eliot Horowitz and Kevin Ryan, MongoDB was first released in 2009 and soon became a popular NoSQL database. It is a document-oriented database that is perfect for web applications. It uses a storage format called “Binary Serialized Document Format” (BSON). The strength of MongoDB is that it allows users to store their data faster compared to relational models. Also with less constraints than that of RDBMS, MongoDB gives its users more flexibility on data structure. It supports a variety of programming languages, including but not limiting to C++, C#, Ruby, Python and Java. It is easy to name some famous users of MongoDB, including New York Times, FourSquare, Craigslist and Esty (DB-Engines, 2021).

Amazon DynamoDB

Developed by the mega company Amazon, AWS DynamoDB is a cloud database first introduced in 2012 as an evolution of its predecessor Amazon SimpleDB. It is a key-valued oriented database system that helps its users manage scalable amount of data (they can store as much data as they need) while the price is based on throughput by the users. Amazon provides full service so its users do not need to worry about any administration issues. Notable company users include Duolingo, Major League Baseball (MLB), Netflix, Airbnb, and Docomo (DB-Engines, 2021).

Redis

Another popular NoSQL database, Redis is an in-memory key-valued database that simultaneously runs as a store and a cache. Developed by Salvatore Sanfilippo and first released in 2009, Redis have been supported by numerous programming languages, including but not limiting to C, C++, C#, Python, R and PHP. Thanks to its in-memory characteristic, It provides outstanding performance in speed of data writing and reading without the need of on-disk storage. However, the down side is the limitation of data size (no larger than memory). Well known clients of Redis include Twitter, GitHub, Snapchat, Pinterest and StackOverflow (DB-Engines, 2021).

Cassandra

Based on BigTable and DynamoDB, Cassandra is a wide column database that fulfills a variety of purposes, such as recommendation engines and product catalogs. Initially released by Facebook in 2008, Cassandra is a schema-free database that allows flexible deployments. Similar to other NoSQL, Cassandra is extremely scalable, making it a useful

tool in large data management. Supported programming languages include but not limit to C#, C++, Java, JavaScript and Python. Its major customers include Uber, Walmart, eBay, McDonalds and Microsoft (DB-Engines, 2021).

Elasticsearch

Based on Apache Lucene, Elasticsearch is a full-text search and analytics engine that was first introduced 11 years ago. It is ideal for enterprise search, observability and security use by its quick and powerful search function. Users can store a large amount of text from any website and lookup for key value relatively easy. Like most NoSQL, it has a schema-free feature. Supported programming languages are not as many as those of the above databases. Some featured users include NASA, Cisco, Goldman Sachs and Verizon (DB-Engines, 2021).

Compare PostgreSQL to MongoDB

This is not simply a RDBMS vs NoSQL comparison. There are many websites telling us the differences between these two types of database.

PostgreSQL	MongoDB
Schema to Start	Schema-free
SQL	BSON
ACID	Potential ACID
Monolithic Architecture	Distributed Architecture

We can compare them directly by “CAP Theorem”. It tells us that Availability, Consistency and Partition Tolerance can not be all satisfied by a system at the same time since there is always a tradeoff. This can be exemplified by our two types of systems: PostgreSQL satisfies both Availability and Consistency, while MongoDB satisfies Consistency and Partition Tolerance.

Some suggest RDBMS will be replaced by NoSQL sooner rather than later. In my opinion, there is no real Swiss army knife in our comparison. It all depends on the actual needs: If we are looking up join queries, PostgreSQL would absolutely be our choice for join queries since it has absolute advantage over any NoSQL. But what if we are looking up for other sort of queries? How much resource/budget do we have for our project? How are we going to scale up our data in the future? Those are only some of the questions we need to answer before we understand which kind of system best serve our

purposes.

Syntax are different on PostgreSQL and MongoDB when they are referring to similar items. For instance, they have TABLES vs COLLECTIONS, ROWS vs DOCUMENTS, INDEXES vs JOINS respectively.

Reference List

1. *MongoDB System Properties*. (2021). DB-Engines. <https://db-engines.com/en/system/MongoDB>
 2. *Amazon DynamoDB System Properties*. (2021). DB-Engines. <https://db-engines.com/en/system/Amazon+DynamoDB>
 3. *Redis System Properties*. (2021). DB-Engines. <https://db-engines.com/en/system/Redis>
 4. *Cassandra System Properties*. (2021). DB-Engines. <https://db-engines.com/en/system/Cassandra>
- Elasticsearch System Properties*. (2021). DB-Engines. <https://db-engines.com/en/system/Elasticsearch>