NoSQL Database Model for Big Health Data Analysis

Bryant Cornwell
D-590 SQL/noSQL
Summer 2021

What is Health Data?

- Characteristics of health data:
  - Mostly document based
  - Varying form/structure
  - Requires frequent read/write tasks
- Due to the tremendous amount of health data and various forms, it would be difficult to keep up with changes to a relational schema in a database.
- The data is used to find connections and relationships that may help healthcare providers make important decisions.

SQL vs NoSQL Databases for Health Data

SQL

- Required relational schema and normalization can create downtime when making schema changes or database development.
- Has been an industry standard at many companies and commercial software.

NoSQL (MongoDB)

- No structure required, so document-based data with any or no structure can be added without any downtime.
- Relatively new, but its flexibility allows quicker development.
- MongoDB has the capability to allocate data and computation across multiple machines through a process called Sharding.

Methods

- Databases compared:
  - SQL Server
  - MongoDB
  - MongoDB-sharded (cloud network of machines to extend data storage and computation across them)
- To compare read and write speeds of each database, four queries were used.
- Response times of SQL Server, MongoDB and Mongo-sharded were compared on four queries of varying complexity.
- The first query focused on writing data while the other three queries read data. The data written in the first query is the same for each database.
- Each query was performed 4 times and each time the number of records were increased. The response time for each query on each database configuration was recorded and compared.

Table 4 – The variation of queries.

Query	Query description
I	To write health data (Drugs and their effects)
II	To retrieve results containing one textual value ("Drug-name = Co-amoxiclav")
III	To retrieve results containing tow textual value ("Drug-name = Co-amoxiclav" and "Side effect = dizziness")
IV	To retrieve results containing tow textual value and one numerical value ("Drug-name = Co-amoxiclav" , "Milligrams = 500" and "Side effect = dizziness")

Figure 1. Description of the queries used for each database to gauge performance.
Image obtained from reference: Goli-Malekabadi et al. p. 79

Results

- Write query favored the MongoDB configuration.
- MongoDB showed long response times for queries that involved mass retrieval of data.
- However, the read queries favored the Shard version of MongoDB for retrieving large number of results.
- The spread of data and processing power that cloud computing provides had reduced the response times equivalent to the SQL Server database.

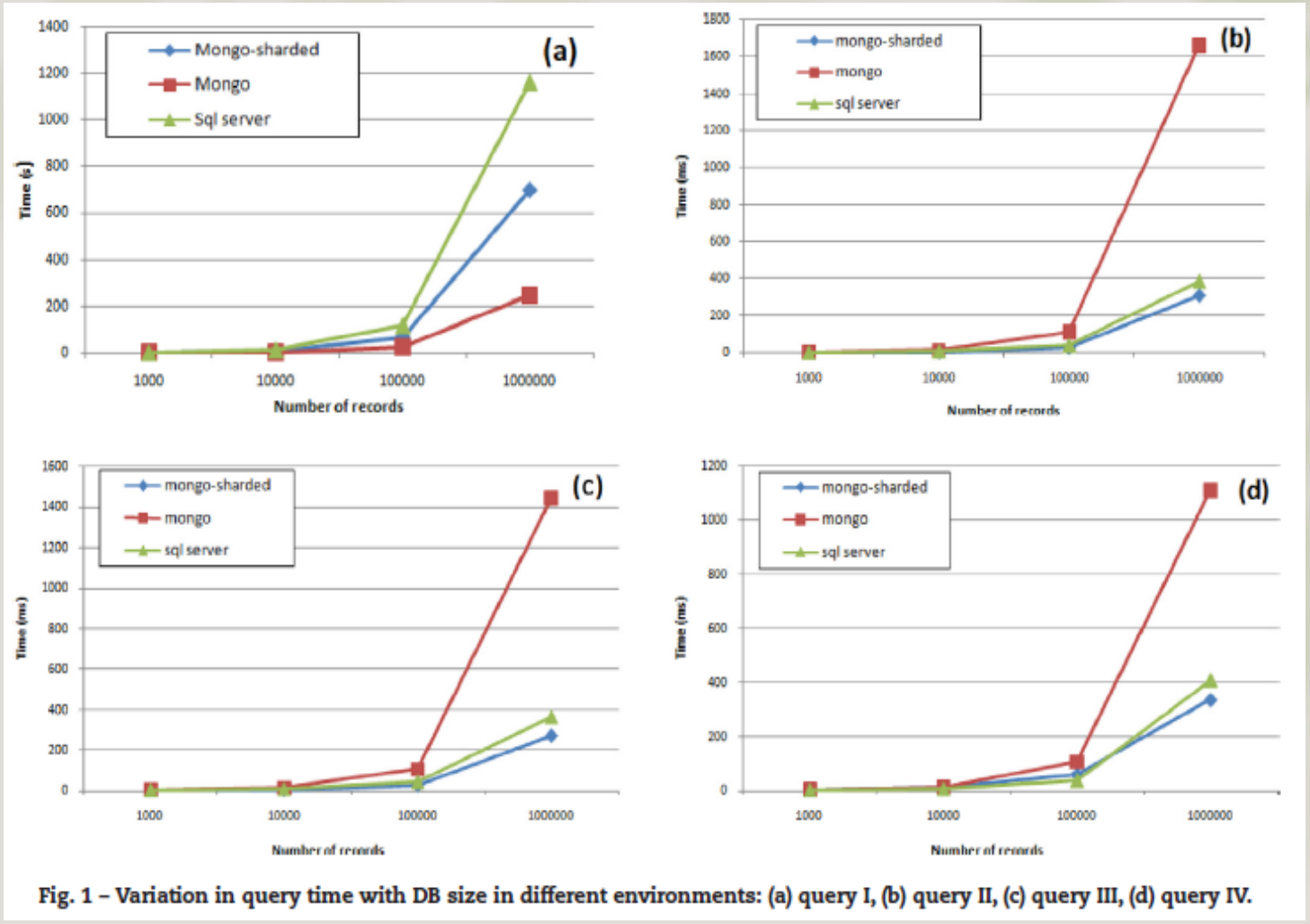

Fig. 1 – Variation in query time with DB size in different environments: (a) query I, (b) query II, (c) query III, (d) query IV.

Figure 2. Charts above display the relationship between response time and number of result for each database configuration on each query.
Image obtained from reference: Goli-Malekabadi et al. p. 80

Results

- Mongo-sharded took more time to write data since the data was not written to one single machine.

- Reading fewer records took longer for Mongo-sharded compared to SQL Server, but for large number of records the response time was shorter than SQL Server.

Table 5 – Query performance of different databases with various sizes: query I.

Number of records	Response time(s) in various database implementations		
	SQL Server	Mongo	Mongo-sharded
1000	1.13	0.24	1.15
10,000	12.34	2.41	7.29
100,000	117.9	23.8	68.6
1,000,000	1161.31	248.46	700.49

Table 8 – Query performance of different databases with various sizes: query IV.

Number of records	Response time(ms) in various database implementations		
	SQL Server	Mongo	Mongo-sharded
1000	1.44	1.3	2.46
10,000	5.79	10.45	9.34
100,000	38.13	103.83	60
1,000,000	408.65	1111.06	336.33

Figure 3. The tables above display the comparison of read and write response times for each database.
Image obtained from reference: Goli-Malekabadi et al. p. 79

Challenges and Future Direction

- Future development of this model may include applying current data storing and processing tools. Example: map-reduce programming models.
- Since the health data consists of sensitive information, security and privacy issues plague this model.
- The security in Cloud settings must be studied to develop a method to handle these security issues.

Personal Reflection

- MongoDB provides more accessibility and eases the database development process for new companies or companies considering switching databases from SQL.
- Sharding slowed down the writing speed more than the base MongoDB configuration but performs read operations much quicker than the base MongoDB. There is not a set configuration that is great for both read/write queries, but the Mongo-sharded configuration seems the best route to go for the continued expansion of health data.

References

- Goli-Malekabadi, Z., Sargolzaei-Javan, M., & Akbari, M. K. (2016). An effective model for store and retrieve big health data in cloud computing. *Computer Methods and Programs in Biomedicine, 132*(1), 75-82. http://dx.doi.org/10.1016/j.cmpb.2016.04.016

- MongoDB, Inc. (n.d.). *Sharding in MongoDB*. https://www.mongodb.com/basics/sharding