**MongoDB Sharding : A Literature Review**

[Sharding](https://docs.mongodb.com/manual/reference/glossary/#std-term-sharding) is a method for distributing data across multiple machines. MongoDB uses sharding to support deployments with very large data sets and high throughput operations. (MongoDB, 2008) MongoDB is mainly a distributive database which uses sharding to spit and store a single logical dataset in multiple databases. (Kim, 2014)

Sharding is also referred as **horizontal partitioning**. The distinction of **horizontal**vs **vertical** comes from the traditional tabular view of a database. A database can be split vertically — storing different tables & columns in a separate database, or horizontally — storing rows of a same table in multiple database nodes. (Kim, 2014)

In this literature review most of the information focuses on sharding and its architecture.

**Addressing System Growth**

There are two methods for addressing system growth: vertical and horizontal scaling.

Vertical Scaling: Vertical Scaling involves increasing the capacity of a single server, such as using a more powerful CPU, adding more RAM, or increasing the amount of storage space.

Horizontal Scaling: Horizontal Scaling involves dividing the system dataset and load over multiple servers, adding additional servers to increase capacity as required. While the overall speed or capacity of a single machine may not be high, each machine handles a subset of the overall workload, potentially providing better efficiency than a single high-speed high-capacity server.

**Sharding in MongoDB**

There are mainly 2 types of sharding in MongoDB.

Range Based Sharding: The shard key is used in range-based partitioning to determine the range of data in the collection. For example, if the shard key is an integer, then consider a range from negative infinity to infinity. Then this range is divided into smaller partitions called chunks. Each chunk has its maximum and minimum value. (Sivaraman, 2014)

Hash Based Sharding: In hash-based partitioning, hash is computed for each shard key in the collection. Then these hashes are used to create the chunks. This method is used to distribute data randomly. (Sivaraman, 2014)

It is a well-known and proven fact that sharding is important in MongoDB.

**Sharding Implementation**

To implement sharding, three things have to be configured – query router, config server and shard servers.

Starting Config Server Instances: Config server stores the metadata information about the data stored in shards. Generally it is advised to have more than one config server.

Starting Shard Server Instances: Shard servers are the ones that store the actual data. The data is divided into chunks and is distributed evenly across all the configured shards.

Starting Query Router Instances: The mongos instances act as query routers. They are light weight and do not need data directories. The mongos instances can even run on the same server which runs the config server.

We had hard time understanding Config Server instances but after this review, most of the confusion got cleared.This literature review actually helped us finding more about sharding and we got to know in depth.

**Conclusion**

MongoDB offers built-in sharding to implement a large database without compromising the performance. In this review, we mainly focused on MongoDB sharding and its basic concepts. We hope the above helps to setup MongoDB sharding. (Ali, 2020)

# Works Cited

Ali, A. (2020). Retrieved from https://geekflare.com/mongodb-sharding/

Kim, j. (2014, December 5). *Sharding*. Retrieved from https://medium.com/@jeeyoungk/how-sharding-works-b4dec46b3f6#:~:text=Sharding%20is%20a%20method%20of,dataset%20and%20handle%20additional%20requests.&text=Sharding%20allows%20a%20database%20cluster,its%20data%20and%20traffic%20growth.

*MongoDB*. (2008). Retrieved from https://docs.mongodb.com/manual/sharding/

Sivaraman, k. (2014). Retrieved from Semantic Scholar: https://www.semanticscholar.org/paper/2-.-3-Sharding-in-MongoDB-Sivaraman/a3ac2a94a94fed5a44236373ef1ec42900f6a6f0

*Troubleshooting a MongoDB Sharded Cluster*. (2020, June 12). Retrieved from https://severalnines.com/database-blog/troubleshooting-mongodb-sharded-cluster