Database Systems:

Module 13, Lecture 2 – MongoDB

Instructor: Alan Paradise

MongoDB

LESSON OBJECTIVES

- Be able to describe the MongoDB architecture
- Understand how the MongoDB software embodies many key components of the NoSQL model

MongoDB

MongoDB is a NoSQL “Document” database.

- Stores collections of documents in a key:value pair format
- MongoDB is NOT Relational
- MongoDB does NOT store data in tables
- MongoDB does NOT use the SQL query language
- MongoDB uses its own JS-like query language
- Community edition is free, Enterprise edition = $$$
- Enterprise users can purchase support, advanced features, cloud deployment

MongoDB

MongoDB Concepts

- A MongoDB DATABASE can contain one or more COLLECTIONS
- A MongoDB **COLLECTION** can contain many **DOCUMENTS**
  - Each **document** has a primary key
  - Primary keys (and any other field) can be **indexed** for faster query performance

MongoDB

MongoDB Concepts

- Documents are structured and stored in JSON-like format
  - MongoDB stores data in BSON
    - Binary JSON
  - BSON was invented by the MongoDB folks…

MongoDB

JSON – Java Script Object Notation

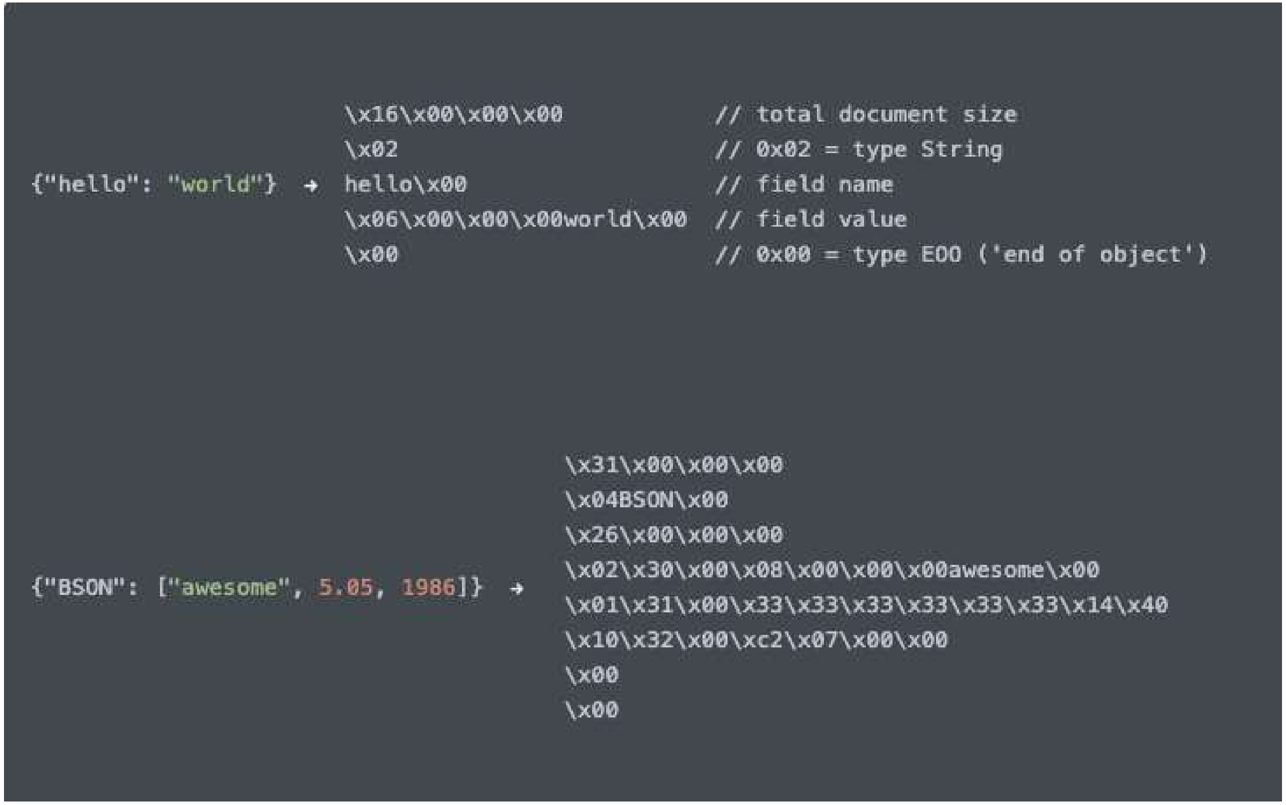
- Invented along with Java Script
- Pushed out and replaced XML
  - XML is more verbose, and older
  - JSON is easier for humans to work with and newer
- A string “key” is mapped to a “value”
  - The value can be a number, string, array
- Very widely used in web-based software development

MongoDB

BSON – Binary Java Script Object Notation

- JSON is string based
- String parsing is relatively slow
- JSON is human-readable, but less efficient for storage and movement of data across networks
- BSON was invented by MongoDB developers to replace JSON with a data format/structure that provides
- Faster data movement
- More efficient storage
- More flexible – can store more types of data (like integer versus floating point)
- BSON encodes data item TYPE and LENGTH in binary notation

MongoDB



```
                      \x16\x00\x00\x00          // total document size
                      \x02                      // 0x02 = type String
{"hello": "world"} →  hello\x00                 // field name
                      \x06\x00\x00\x00world\x00 // field value
                      \x00                      // 0x00 = type EOO ('end of object')
```

```
                                      \x31\x00\x00\x00
                                      \x04BSON\x00
                                      \x26\x00\x00\x00
                                      \x02\x30\x00\x08\x00\x00\x00awesome\x00
{"BSON": ["awesome", 5.05, 1986]} →   \x01\x31\x00\x33\x33\x33\x33\x33\x33\x14\x40
                                      \x10\x32\x00\xc2\x07\x00\x00
                                      \x00
                                      \x00
```

MongoDB

MongoDB Documents

- Are “polymorphic” (i.e. “multiple shapes”…)
  - Not all documents in a collection must have all

the same key:value pairs.

- There is no need to declare the structure of a document
- It is “self-describing”

- The database takes the JSON you provide, converts it to BSON, and stores/retrieves it efficiently , then converts back to JSON on retrieval

- You can add fields to one document in a collection without modifying any other documents in the same collection

MongoDB

MongoDB relies on REPLICATION

- MongoDB provides horizontal scaling
- You can configure a scalable number of nodes in a cluster
- The cluster can be spread across data centers and geography
- MongoDB can easily scale READ operations across the cluster (parallelization)

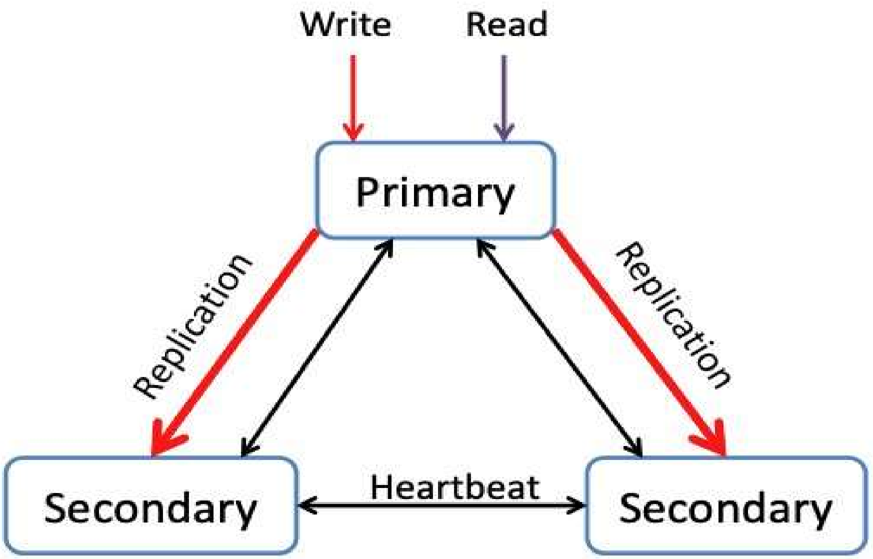
MongoDB

MongoDB relies on Primary-to-Secondary REPLICATION

- You can configure a number of replica sets
- Each set is replicated across the cluster
- Provides High Availability
- If MongoDB detects that it has lost a node, it will shift processing over to another replica

MongoDB

Replication



MongoDB

MongoDB node role management

- If the Primary node fails for any reason, the other members vote to elect a new primary from among the secondary nodes

- No Downtime required for Upgrades

Distribution via Replication allows administrators to

- take a node offline
- upgrade hardware or software
- bring it back online
- NO DOWNTIME needed

MongoDB

MongoDB relies on SHARDING

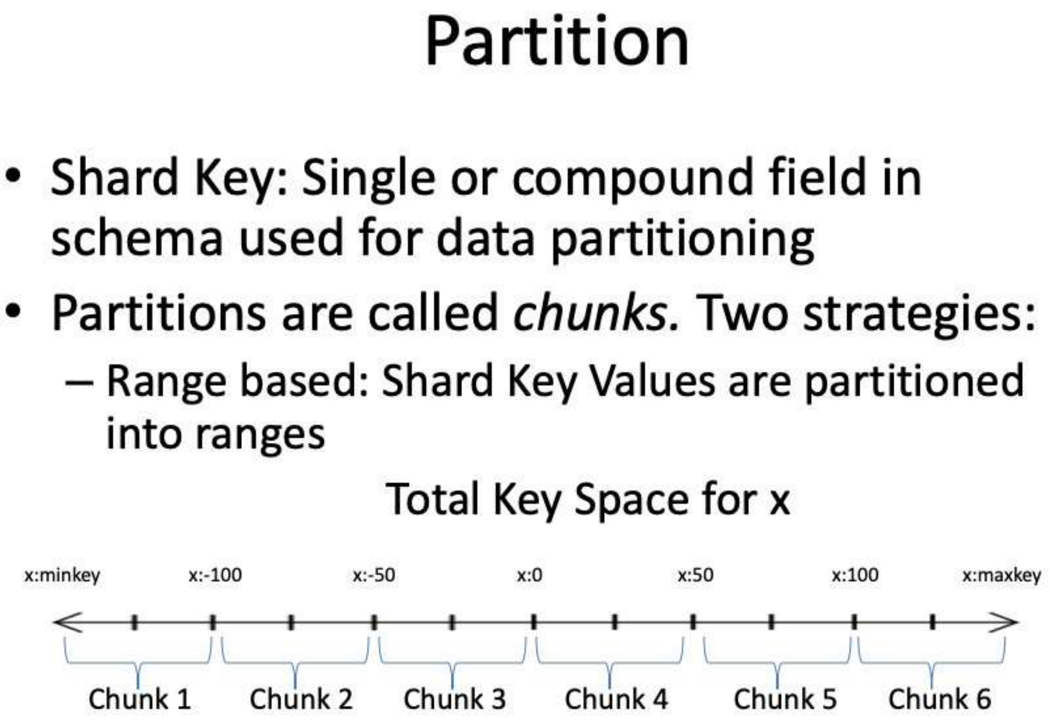
- MongoDB uses horizontal scaling for sharding
- You can spread your data out across multiple nodes
- A Mongo shard is referred to as a “partition”
- Distributes WRITES across multiple nodes/partitions (parallelization)
- Data can be distributed based on user query patterns

MongoDB

MongoDB relies on SHARDING

- MongoDB offers architects options regarding sharding
- Each document has a primary key
- Partition by key ranges
  - Co-locating documents based on geography
- Partition by hash
  - A more random but uniform spread of data

MongoDB



Partition

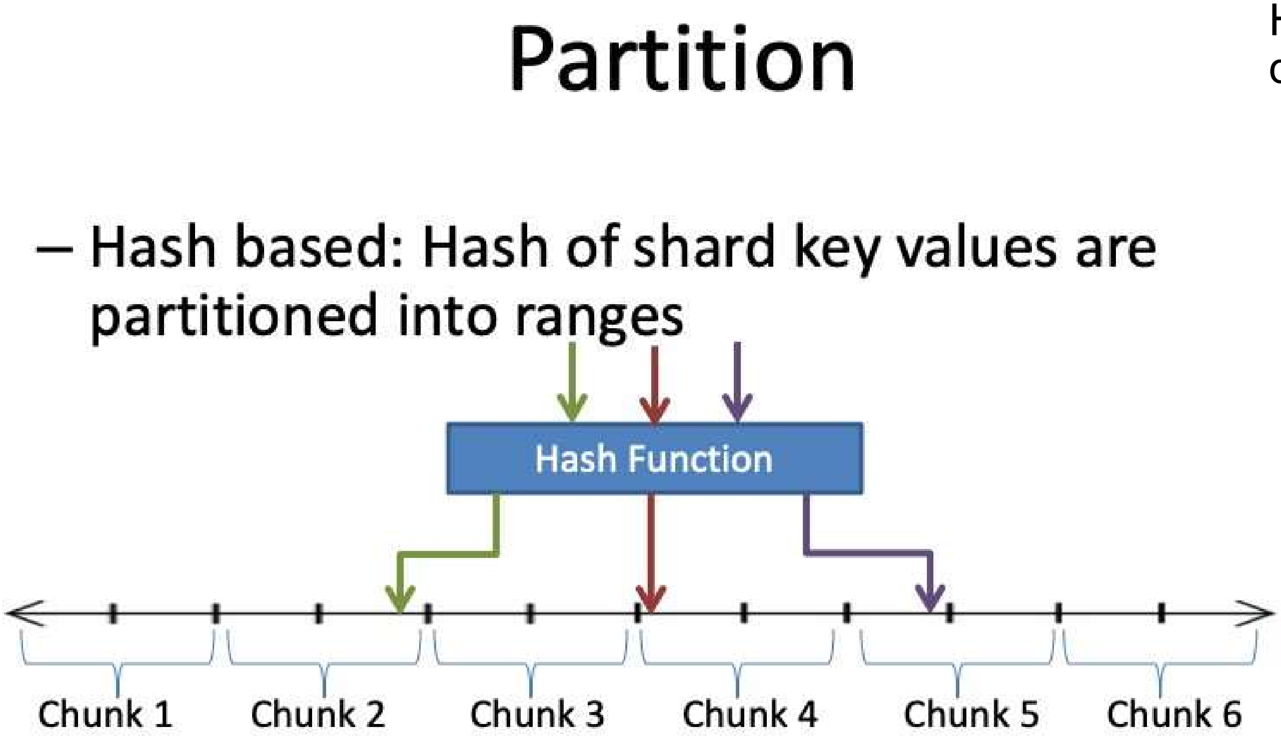
- Shard Key: Single or compound field in schema used for data partitioning
- Partitions are called *chunks.* Two strategies:
  - Range based: Shard Key Values are partitioned into ranges

Total Key Space for x

x:minkey
x:-100
x:-50
x:0
x:50
x:100
x:maxkey
Chunk 1
Chunk 2
Chunk 3
Chunk 4
Chunk 5
Chunk 6

MongoDB

Hash Scheme leads to better data balancing across nodes



Partition

- Hash based: Hash of shard key values are partitioned into ranges

MongoDB

Balancing

- Splitting: Background process which splits the partition when a chunk grows beyond a threshold
- Balancing: Migrates chunks among shards if there is an uneven distribution

MongoDB

More Resources

- http://mongodb.org
- https://docs.mongodb.com/manual/tutorial/getting-started/

MongoDB

Next Topic: The MongoDB Query Language