Notes on 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 a JS-like query language
- Community edition is free, Enterprise edition = $$$
  - Enterprise users can purchase support, advanced features, cloud deployment

Notes on 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 performance

Notes on 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…

Notes on MongoDB

JSON

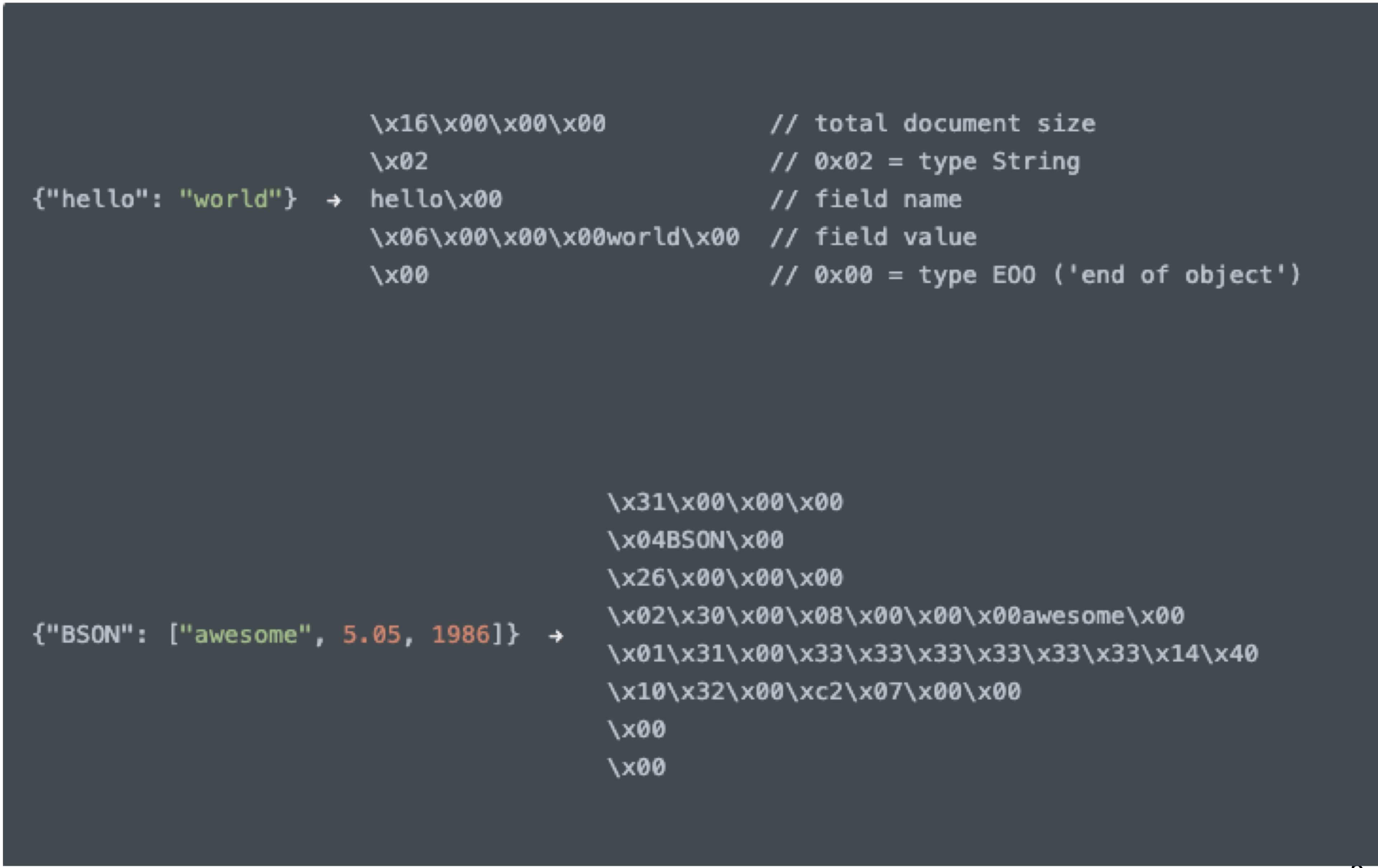
- Invented along with Java Script
- Pushed out and replaced XML
  - XML is more verbose
  - JSON is easier for humans to work with
- A string “key” is mapped to a “value”
  - The value can be a number, string, array
- Very widely used in web-based software development

Notes on MongoDB

BSON

- 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 folks 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

Notes on MongoDB



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

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

Notes on MongoDB

MongoDB Documents

- They are “polymorphic” (i.e. “multiple shapes”…)
    - i.e. 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
    - 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

Notes on MongoDB

MongoDB relies on REPLICATION

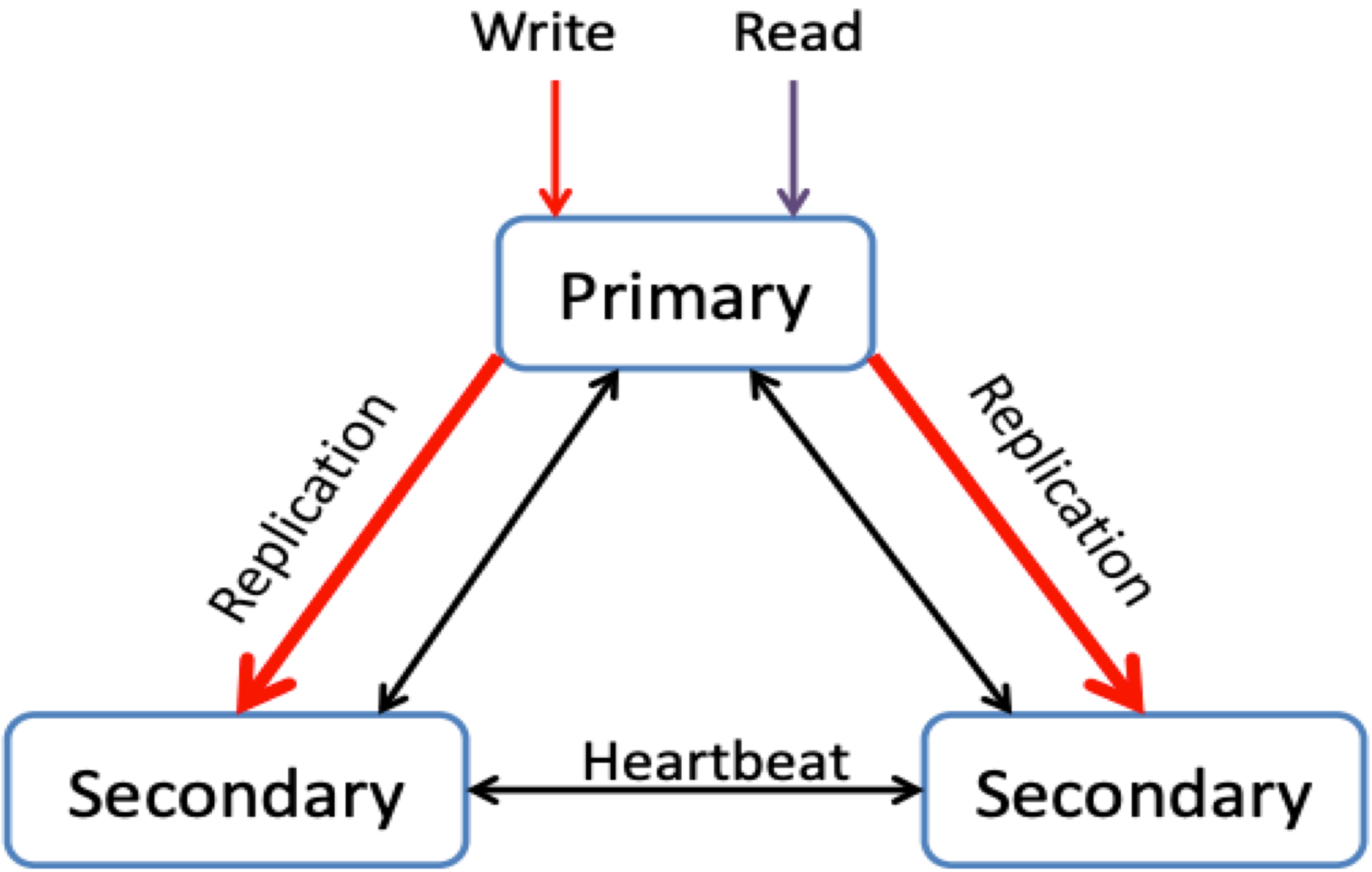
- MongoDB provides horizontal scaling
  - You can configure a scalable number of nodes in a cluster
  - The cluster can be spread across data centers and geography
  - MongoDB can easily scale READ operations across the cluster (parallelization)

Notes on MongoDB

MongoDB relies on 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

Replication



Notes on 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 for Upgrades

Distribution via Replication allows administrators to

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

Notes on MongoDB

MongoDB relies on SHARDING

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

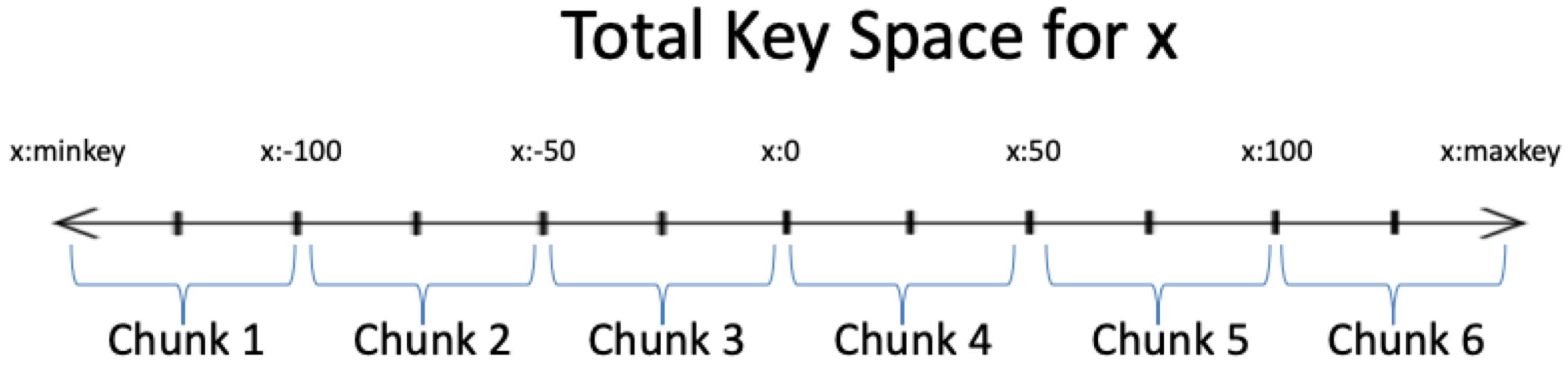
Notes on MongoDB

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

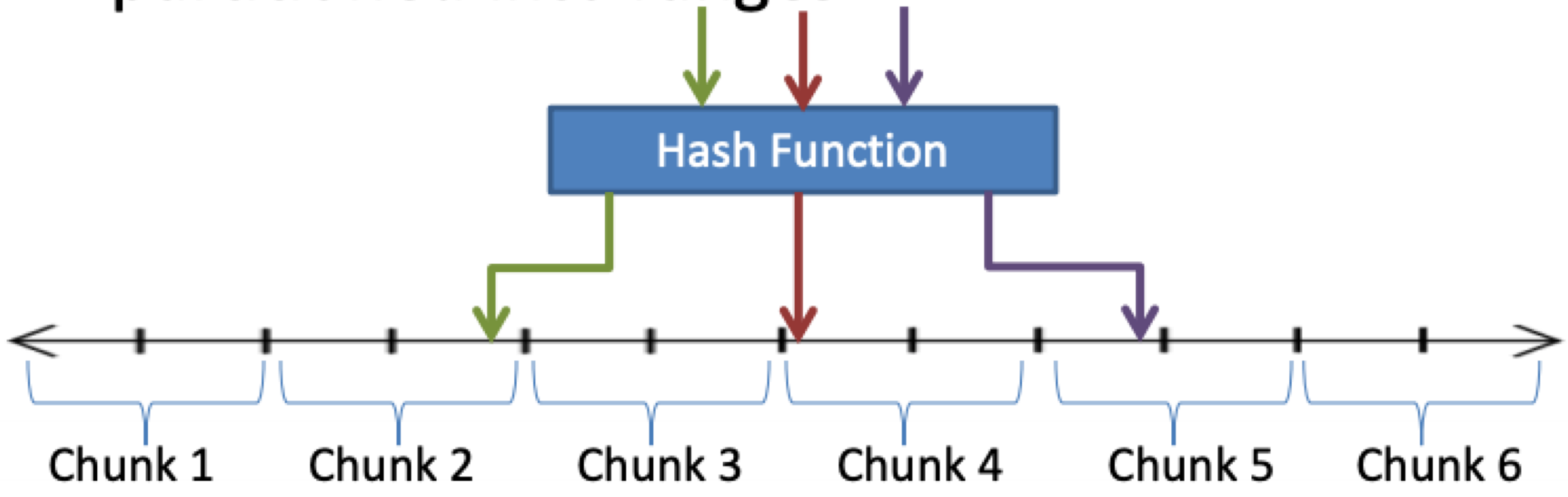
Partition

- 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

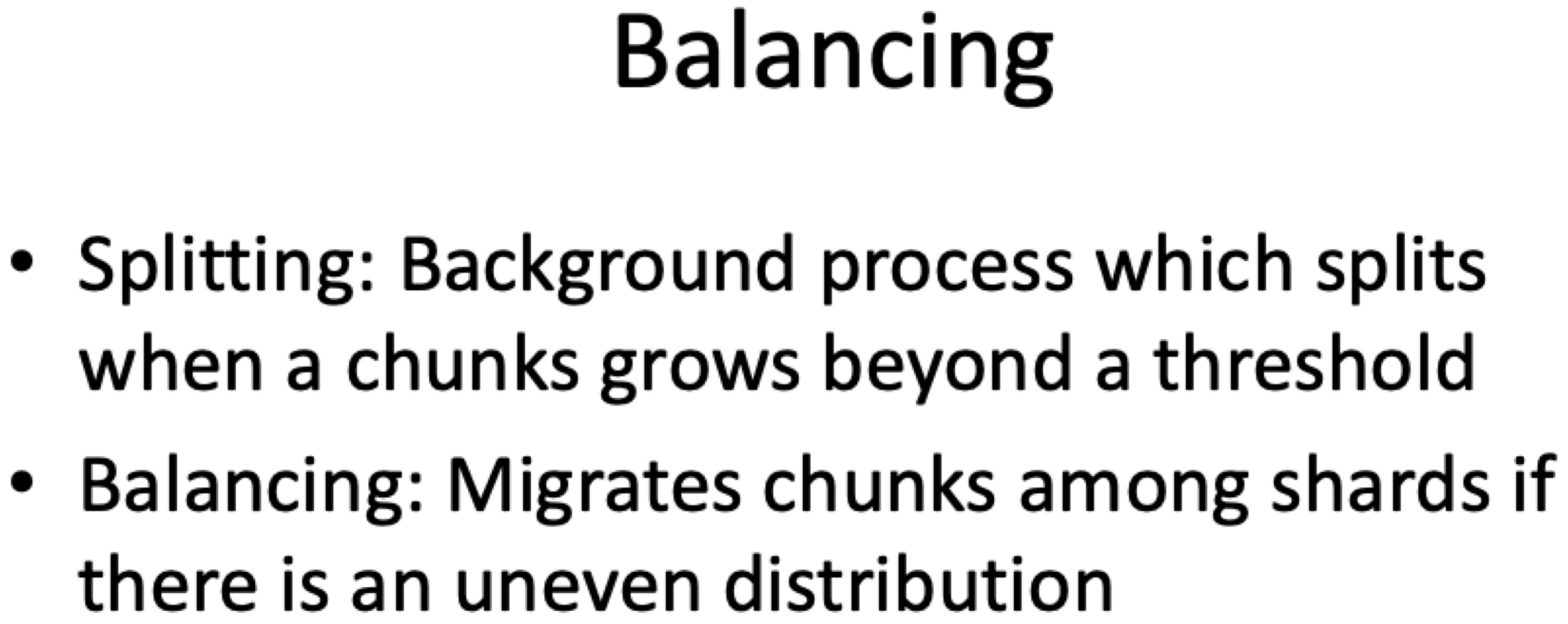


Partition

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



- Hash Scheme leads to better data balancing



Balancing

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

More Resources

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