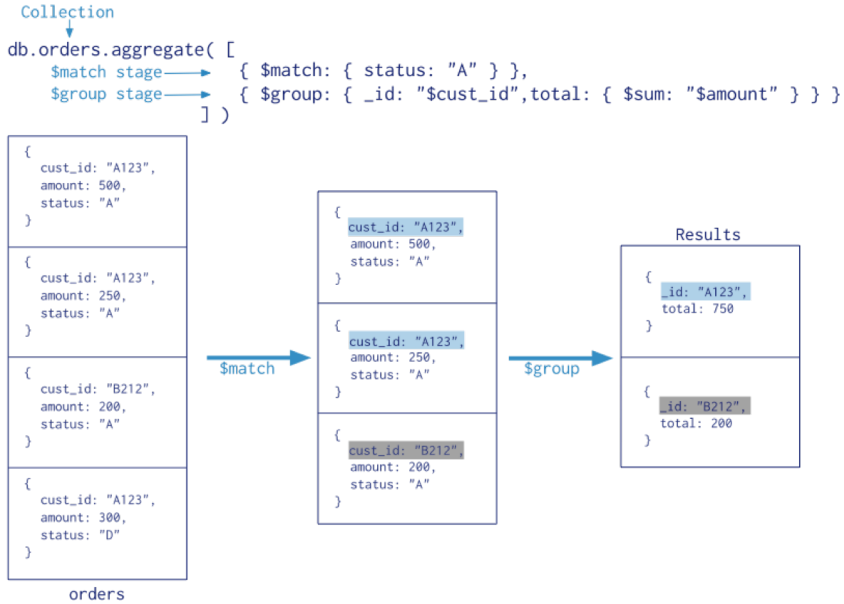
MongoDB

Operaciones de agregación

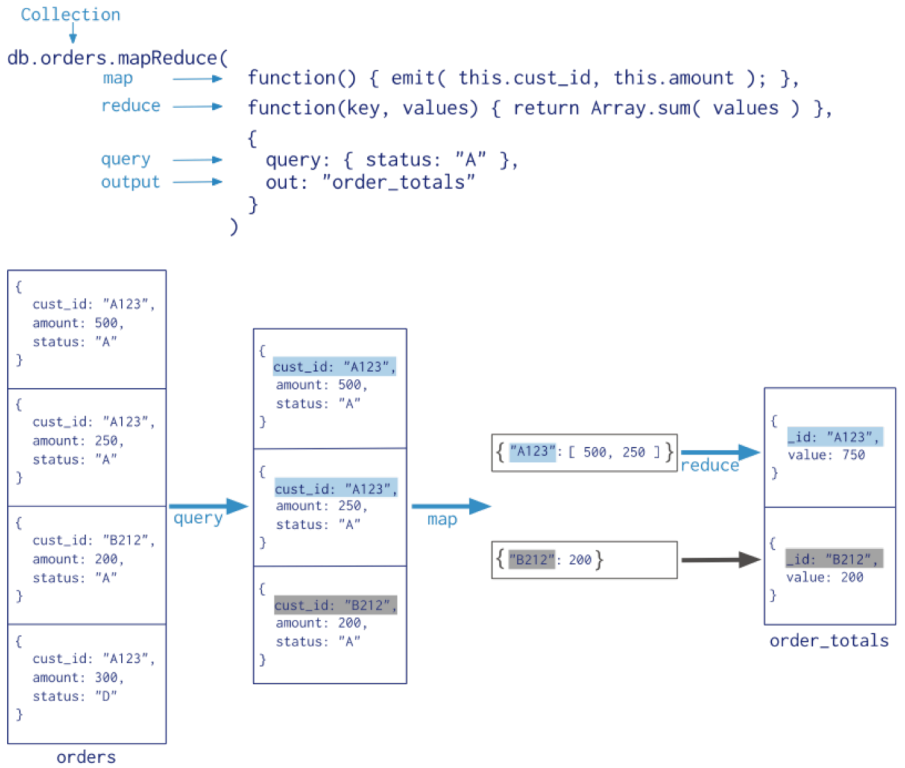
- Procesan datos y devuelven resultados



MongoDB

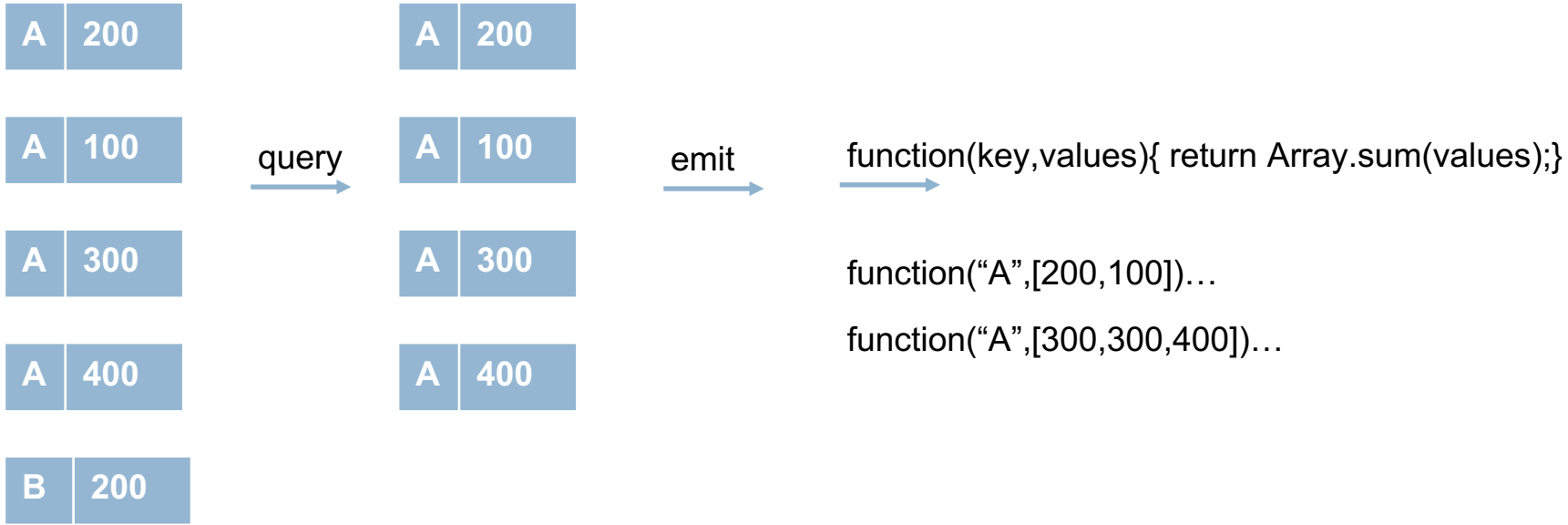
Operaciones MapReduce

- En general consta de dos fases
  - Una fase “map” en la cual se procesan los datos y se “emite” un resultado parcial
  - Una fase ”reduce” que combina el resultado en un solo resultado final



MongoDB

Un ejemplo mas complicado



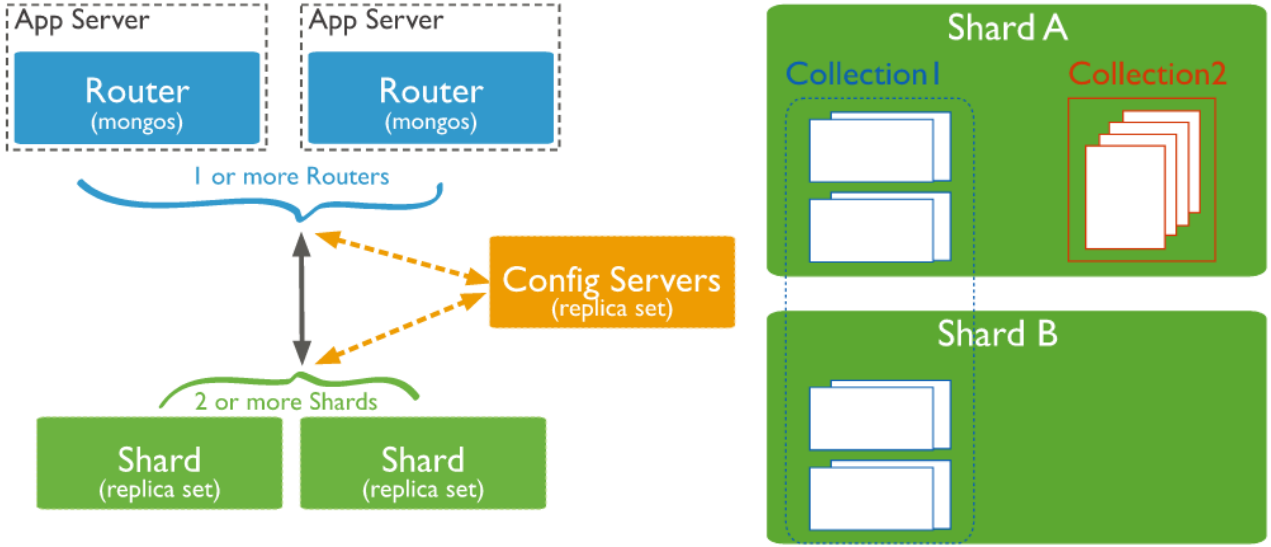
MongoDB

Sharding

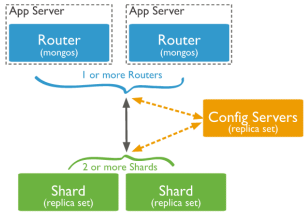
- Es un método para distribuir datos en múltiples máquinas.
- Existen dos maneras de enfrentar crecimiento de un sistema:
  - Verticalmente: implica aumentar las capacidades de un solo servidor.
  - Horizontalmente: implica dividir la información en porciones más pequeñas y distribuir la carga en múltiples equipos.

MongoDB

Sharding



MongoDB



Sharding

- Shard
  - Cada “shard” contiene un subconjunto de la información partida.
  - A cada parte se la debería distribuir como ”replica set”.

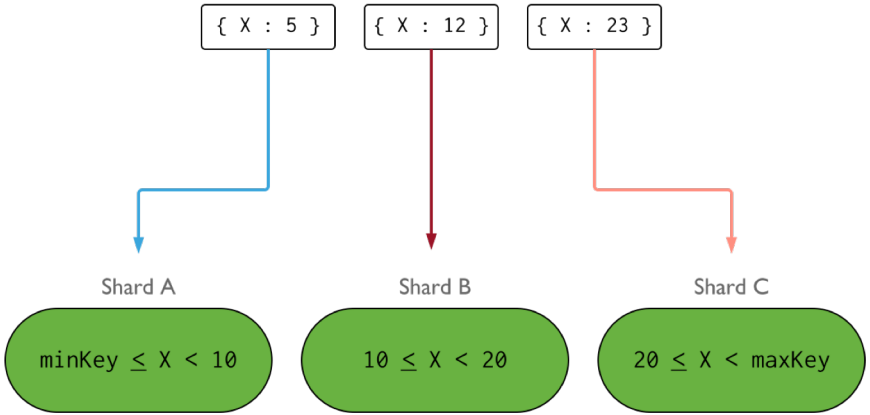
- Mongos: actúan como los puntos de entrada de los clientes para rutear las consultas.

- Config servers: almacenan la configuración y metadata relacionada con el cluster.

MongoDB

Shards key

- Mongo realiza la partición a nivel de colecciones utilizando una “shard key”.
- Está compuesta de uno o más campos inmutables.
- Solamente puede existir una shard key por colección y no se puede cambiar.



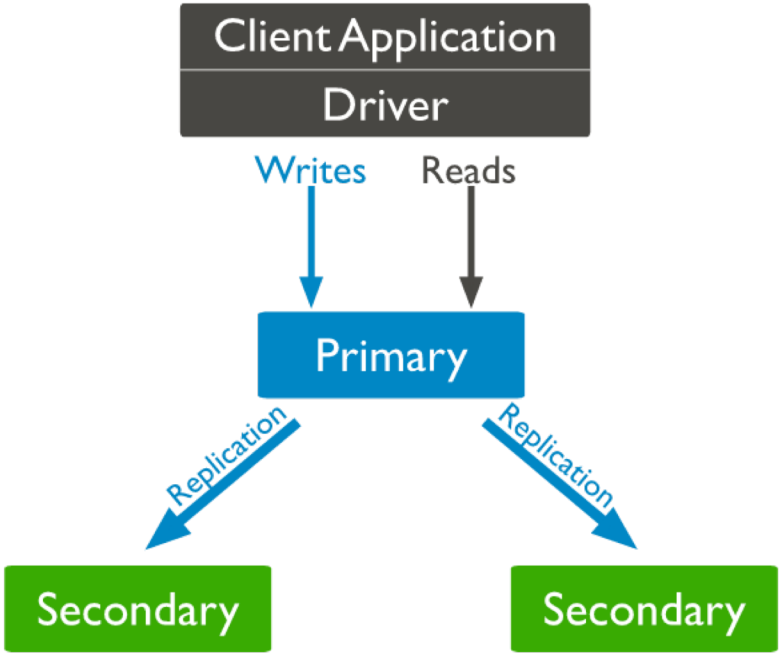
MongoDB

Ventajas y beneficios

- Lecturas / escrituras
  - La carga se distribuye horizontalmente en todo el cluster.
- Capacidad de almacenamiento
  - Si el espacio requerido aumenta, simplemente se agregan nuevos nodos al cluster.
- Alta disponibilidad
  - Aún cuando algún nodo no funcione, se puede seguir resolviendo parcialmente el pedido.

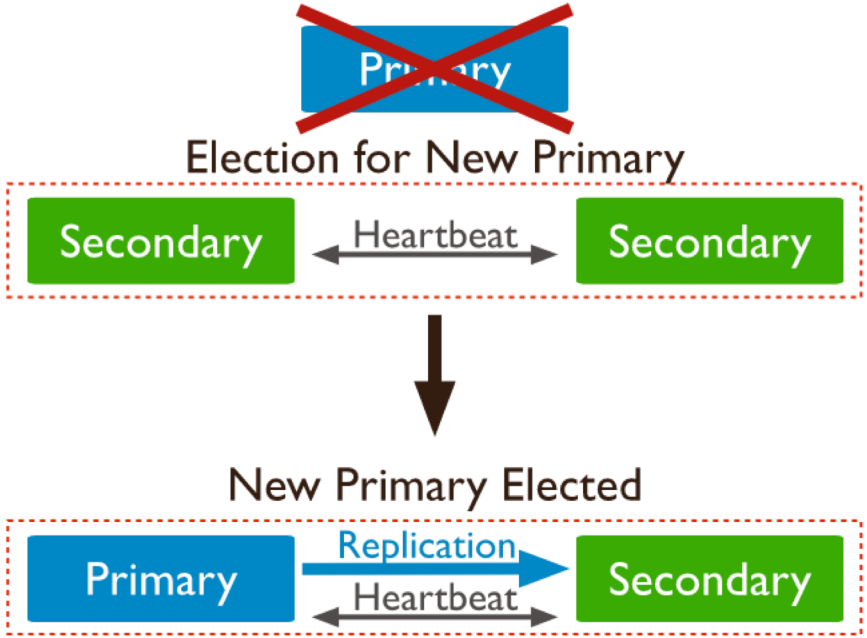
MongoDB

Replication



MongoDB

Automatic failover



MongoDB

Trabajo con MongoDB desde Java

Dependencias de Maven

```
<dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongo-java-driver</artifactId>
    <version>3.4.1</version>
</dependency>
```

Conectarse a un servidor

```
MongoClient mongoClient = new MongoClient("localhost", 27017);
```

Acceder a una base de datos

```
DB database = mongoClient.getDB("myMongoDb");
```

Obtener todas las bases de datos

```
mongoClient.getDatabaseNames().forEach(System.out::println);
```

Crear una nueva colección

```
database.createCollection("customers", null);
```

MongoDB

Trabajo con MongoDB desde Java

Guardar

```
DBCollection collection = database.getCollection("customers");
BasicDBObject document = new BasicDBObject();
document.put("name", "Shubham");
document.put("company", "Baeldung");
collection.insert(document);
```

Realizar una consulta

```
BasicDBObject searchQuery = new BasicDBObject();
searchQuery.put("name", "John");
DBCursor cursor = collection.find(searchQuery);

while (cursor.hasNext()) {
    System.out.println(cursor.next());
}
```