

# MongoDB Sharding



document author: Yongjie Lyu    Yongjie.L@outlook.com

通过在多个MongoDB实例之间拆分数据进行分片来实现水平扩展，通过分片来支持海量数据集和高吞吐量操作的部署方案。

- 分片将数据分布在一个集群的机器上，读写操作可以分发给各个分片来提高性能。
- MongoDB支持基于分片键创建数据区域

## ▼ 分片优势：

读写负载

存储容量

高可用性

每个shard包含被分片的数据集中的一个子集。每个分片可以被部署为副本集。

**mongos**：客户端接入路由器，在客户端应用程序和分片集群之间提供接口，前端应用可以透明使用。

**config服务器**：config servers存储了分片集群的元数据和配置信息。

## 分片键

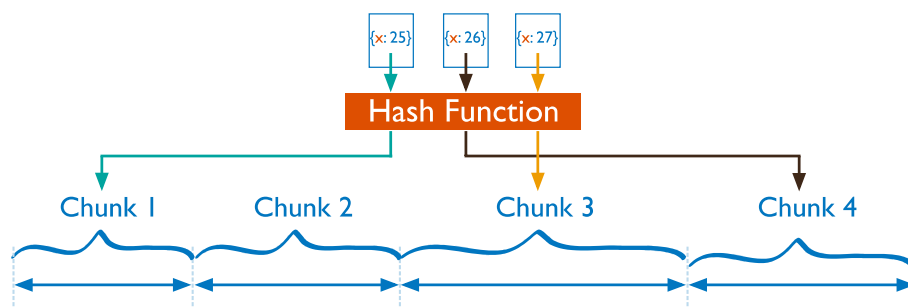
- MongoDB使用分片键在各个分片中分发集合中的文档。分片键由文档中的一个或多个字段组成。
- 在需要将数据分片的时候需要选取分片键，在选取分片键并分片之后就无法更改分片键了。
- 分片键的选择会影响分片群集的性能，效率和可伸缩性。

## Chunks块

MongoDB将分片数据拆分成块。每个分块都有一个基于分片键的上下限范围。

## 分片策略

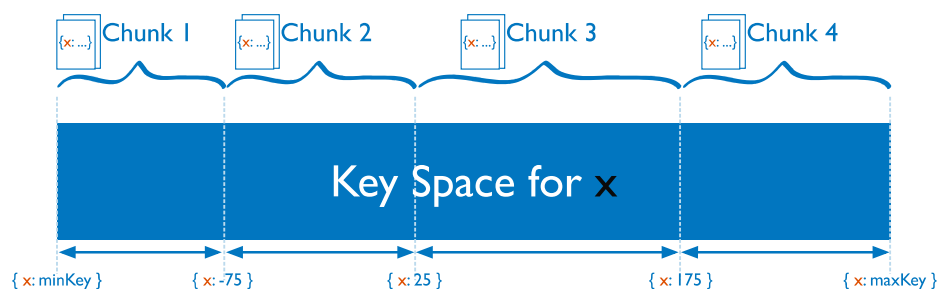
### 哈希分片



哈希分片涉及计算分片键字段值的哈希值。然后，根据散列的分片键值为每个块分配一个范围。

- 优：哈希值的数据分发有助于更均匀的数据分发，尤其是在分片键单调更改的数据集中。
- 劣：哈希分布意味着对分片键的基于范围的查询不太可能针对单个分片，从而导致更多集群范围的广播操作。

### 范围分片



范围分片根据分片键的值将数据划分为多个范围，然后基于分片键的值分配每个块的范围。

往往导致值“接近”的一系列分片键更有可能分布在同一块上。

- 优：便于mongos执行针对性的操作，可以仅将操作路由到包含所需数据的分片上。

- 劣：分片键考虑不周会导致数据分布不均，这可能会削弱分片的某些优势或导致性能瓶颈。

## 生产环境配置

- 配置服务器config部署为拥有3个成员的副本集
- 将每个分片（Shard）部署为拥有3个成员的副本集
- 部署一个或多个mongos路由