

MongoDB 介绍

MongoDB 是一个基于分布式文件存储的数据库，由 C++ 语言编写。旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。

MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。它支持的数据结构非常松散，是类似 json 的 bson 格式，因此可以存储比较复杂的数据类型。Mongo 最大的特点是它支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

MongoDB 适用场景

MongoDB 的主要目标是在键/值存储方式（提供了高性能和高度伸缩性）和传统的 RDBMS 系统（具有丰富的功能）之间架起一座桥梁，它集两者的优势于一身。

Mongo 主要适用于以下场景：

- 网站数据：Mongo 非常适合实时的插入，更新与查询，并具备网站实时数据存储所需的复制及高度伸缩性。
- 缓存：由于性能很高，Mongo 也适合作为信息基础设施的缓存层。在系统重启之后，由 Mongo 搭建的持久化缓存层可以避免下层的数据源过载。
- 大尺寸、低价值的数据：使用传统的关系型数据库存储一些数据时可能会比较昂贵，在此之前，很多时候程序员往往会选择传统的文件进行存储。
- 高伸缩性的场景：Mongo 非常适合由数十或数百台服务器组成的数据库，Mongo 的路线图已经包含对 MapReduce 引擎的内置支持。

- 用于对象及 JSON 数据的存储：Mongo 的 BSON 数据格式非常适合文档化格式的存储及查询。

MongoDB 的使用也会有一些限制，它不适合于以下几个地方：

- 高度事务性的系统：例如，银行或会计系统。传统的关系型数据库目前还是更适用于需要大量原子性复杂事务的应用程序。
- 传统的商业智能应用：针对特定问题的 BI 数据库会产生高度优化的查询方式。对于此类应用，数据仓库可能是更合适的选择。
- 需要 SQL 的问题。

MongoDB 原理

1. MongoDB 复（副）本集与分片：

每个复（副）本集中存储的数据是相同的，相当于主备方式的数据冗余，目的是为了容灾。

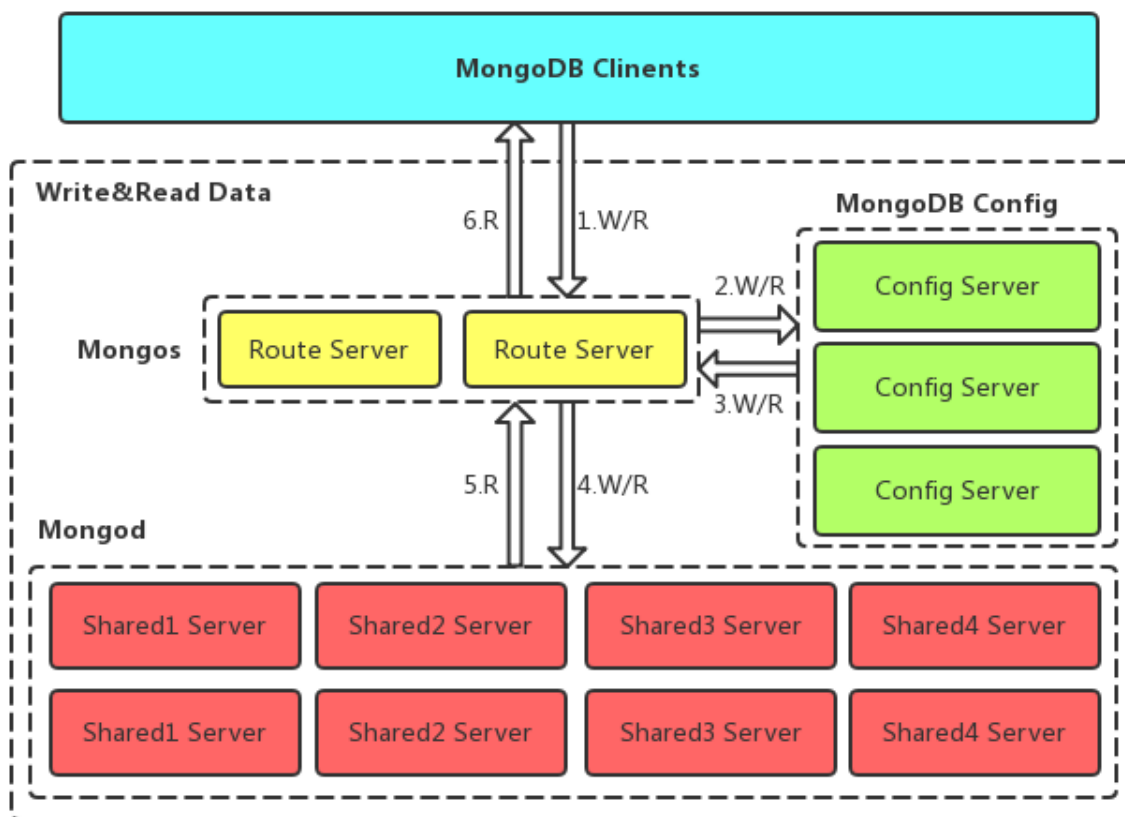
分片是为了数据的拓展，按照片键进行结点划分，数据根据片键存储到对应的服务器上。

2. MongoDB 部署方案

MongoDB 的集群部署方案中有三类角色：实际数据存储结点（Shared Sever）、配置文件存储结点（Config Sever）和路由接入结点（Route Server）。

客户端直接与路由结点相连，从配置结点上查询数据，根据查询结果到实际的存储结点上查询和存储数据。

MongoDB 读写数据的流程如图：

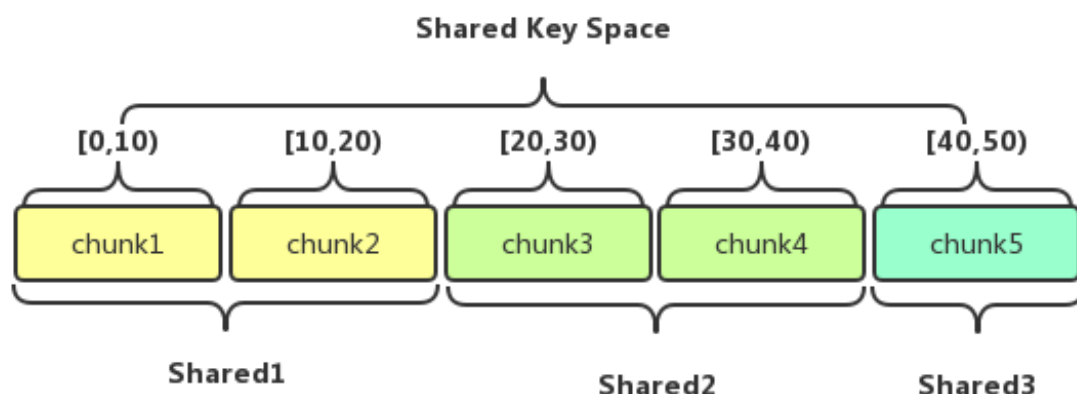


写入数据时，先由客户端访问路由结点，向配置结点获取相关信息后，将数据写入存储结点。

读取数据时，由客户端访问路由结点，向配置结点获取相关信息后，对存储结点进行访问，获取到相应数据，再经由路由结点将数据返回至客户端。

对于副本集，又有主和从两种角色，写数据和读数据也是不同，写数据的过程是只写到主结点中，由主结点以异步的方式同步到从结点中。而读数据则只要从任一结点中读取，具体到哪个结点读取是可以指定的。

对于 MongoDB 的分片，假设我们以某一索引键（ID）为片键，ID 的区间[0,50)，划分成 5 个 chunk，分别存储到 3 个片服务器中，可以进行如下分片：



当数据量很大，需要增加片服务器时，只需移动 chunk 来均分数据即可。

配置结点：存储配置文件的服务器其实存储的是片键与 chunk 以及 chunk 与 server 的映射关系。

路由结点：路由角色的结点在分片的情况下起到负载均衡的作用。

MongoDB 和 MySQL 的区别

1. MySQL 是关系型数据库

- 优点：

在不同的引擎上有不同的存储方式。

查询语句是传统的 SQL 语句，拥有成熟的体系，成熟度很高。

- 缺点：

在处理海量数据时，效率会显著变慢。

2. MongoDB 介于关系数据库和非关系数据库之间，面向集合的、模式自由的文档型数据库。

多个键值对有序地放置在一起即是文档，文档是 MongoDB 中数据的基本单元，类似关系数据库的行。MongoDB 中的查询语法是独特的，有点类似 Javascript 面向对象的查询语言。

存储方式：虚拟内存+持久化。

查询语句：独特的 MongoDB 的查询方式。

架构特点：可以通过副本集，以及分片来实现高可用。

数据处理：数据存储存储在硬盘上，需要经常读取的数据(热数据)会被加载到物理内存中，从而达到高速读写的效果。

成熟度与广泛度：新兴数据库，成熟度较低，Nosql 数据库中最为接近关系型数据库，比较完善的数据库之一，适用人群不断增长。

- 优点：

读写速度快。在适量级的内存的 MongoDB 的速度是十分快速的，通过将热数据存储存储在物理内存中，使得热数据的读写变得十分快速。

高扩展性。存储的数据格式是类似 json 的 bson 格式。

- 缺点：

不支持事务。

3. MongoDB 与 MySQL 命令对比

传统的关系数据库一般由数据库（database）、表（table）、记录（record）三个层次概念组成，MongoDB 是由数据库（database）、集合（collection）、文档对象（document）三个层次组成。

MongoDB 对于关系型数据库里的表，但是集合中不包括列、行和关系概念，这体现了模式自由的特点。

MongoDB 备份方式

1. 文件快照备份

文件快照备份基于操作系统，通过 Linux 的 LVM 特性对 MongoDB 数据文件所在的设备进行备

份。正是由于这种备份方式使用操作系统级别的工具，故而兼备了高可靠性和高效率。不过这样也导致了宿主环境必须要给予支持。文件快照备份的特点：

- 优点：

- 1) 支持时间点
- 2) 高可靠性、高效率
- 3) 增量备份

- 缺点：

- 1) 需要操作系统支持
- 2) 需要开启日志功能，否则无法保证数据的有效性和一致性

- 其他：

- 1) 可以获取分片集群的一致快照，需要关闭集群均衡器并在大致相同的时刻从集群的每个分片和配置服务器上抓取快照。

- 2) 没有开启日志功能时，备份前需要完全完成写操作，备份过程中将停止任何写操作 (MMAPv1)。

- 3) 开启日志功能后，在快照时 MMAPv1 无需停止写操作达到一致状态，Wired Tiger 通过最后的检查点达到一致状态（检查点为每 2G 数据或每分钟一次）。

- 4) 文件快照创建的是整个磁盘的镜像，所以把数据文件、配置、日志放在一个逻辑磁盘上会节约空间。

2. mongodump 备份/mongorestore 恢复

执行 `mongodump` 命令读取 MongoDB 数据库，可以生成一个备份文件，该备份文件格式为 BSON，执行 `mongorestore` 命令可以通过读取该备份文件将 MongoDB 数据库恢复。

使用这种方式进行备份时，对于小型的 MongoDB 集群而言，占用空间更少，也更高效；但是

对于大型的 MongoDB 集群，效率则不是很高。

在备份的过程中，可以继续修改数据，会记录与 oplog 中；备份的数据也不包括本地数据库、索引，故而恢复数据库时必须重建索引；此外备份还会产生网络流量，影响当前 mongod 性能。一旦备份数据超过系统内存，查询操作就会产生错误。执行备份时一定要有的访问控制权限。mongodump 不同版本之间不能互相兼容，无法使用较新的版本备份老数据。