

ES的分布式架构原理？

在搜索这块，lucene是最流行的搜索库。现在很多项目都是直接用基于lucene的分布式搜索引擎-ElasticSearch，简称为ES。现在分布式搜索基本已经成为大部分互联网行业的java系统的标配。

ElasticSearch设计的理念就是分布式搜索引擎，底层其实还是基于lucene的。核心思想就是在多台机器上启动多个es进程实例，组成了一个es集群。

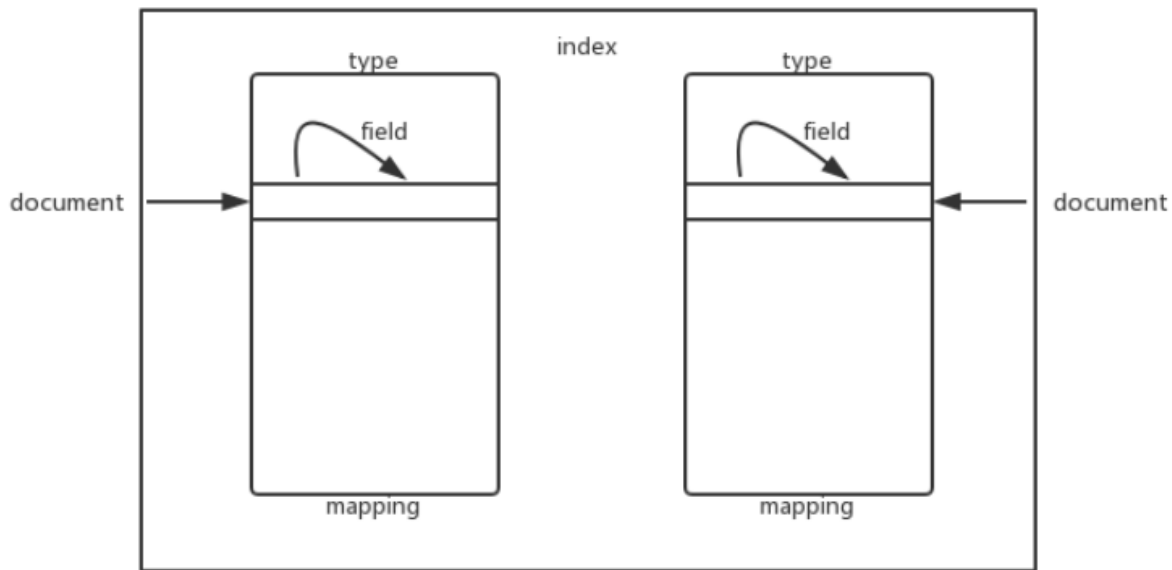
es中存储数据的基本单位是索引，比如说你现在要在es中存储一些订单数据，你就应该在es中创建一个索引order\_idx，所有的订单就都写到这个索引里面去，一个索引差不多就是相当于mysql里的一张表。

index -> type -> mapping -> document -> field

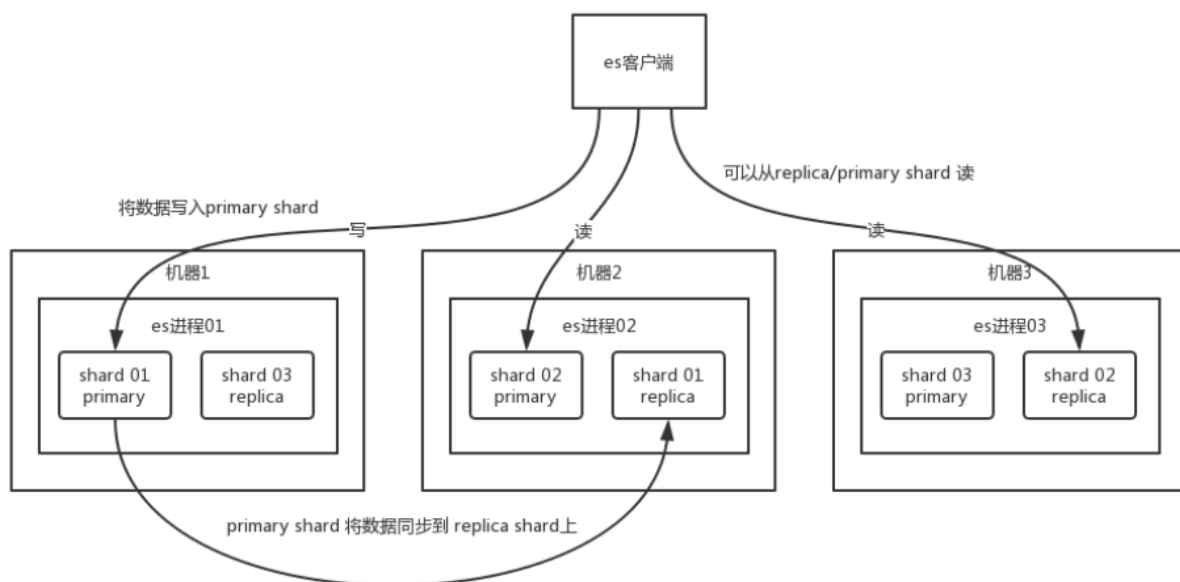
index相当于MySQL里的一张表。而type没法跟mysql里去对比，一个index里可以有多个type，每个type的字段都是差不多的，但是有一些略微的差别。假设有一个index，是订单index，里面专门是放订单数据的。就好比说在mysql中建表，有些订单是实物商品的订单，比如一件衣服；有些订单是虚拟商品的订单，比如游戏点卡。就两种订单大部分字段是一样的，但少部分字段可能有略微的一些差别。

所以就会在订单index里，建两个type，一个是实物商品订单type，一个是虚拟商品type，这两个type大部分字段都是一样的，少部分字段是不一样的。

很多情况下，一个index里可能就一个type，但是确实如果说是一个index里有多个type的情况，可以认为index是一个类别的表，具体的每个type代表了mysql中的一个表。每个type有一个mapping，如果认为一个type是具体的一个表，index就代表多个type同属于一个类型，而mapping就是这个type的表结构定义，你在mysql中创建一个表，肯定是要定义表结构的，里面有什么字段，每个字段是什么类型。实际上往index里的一个type里面写的一条数据，叫做一条document，一条document就代表了mysql中某个表里的一行，每个document有多个field，每个field就代表了这个document中的一个字段的一个值。



如果做一个索引，这个索引可以拆分为多个shard，每个shard存储部分数据。  
接着就是这个shard的数据实际是有多个备份，就是说每个shard都有一个primary shard，负责写入数据，但是还有几个replica shard。primary shard写入数据之后，会将数据同步到其他几个replica shard上去。



通过这个replica的方案，每个shard的数据都有多个备份，如果某个机器宕机了，还有别的数据副本在别的机器上。

es集群多个节点，会自动选举一个节点为master节点，这个master节点其实就是干一些管理工作的，比如维护索引元数据、负责切换primary shard和replica shard身份等。要是master节点宕机了，那么会重新选举一个节点为master节点。

如果是非master节点宕机了，那么会由master节点，让那个宕机节点上的primary shard的身份转移到其他机器上的replica shard。接着你要是修复了那个宕机机器，重启了之后，master节点会控制将缺失的replica shard分配过去，同步后续修改的数据之类的，让集群恢复正常。

上述就是ElasticSearch作为分布式搜索引擎最基本的一个架构设计。

