**elasticsearch 分片(Shards)的理解**

2018年04月25日 13:19:40 [云中鲸](https://me.csdn.net/qq_38486203) 阅读数：2665

* + [分片重要性](https://blog.csdn.net/qq_38486203/article/details/80077844#%E5%88%86%E7%89%87%E9%87%8D%E8%A6%81%E6%80%A7)
  + [分片是什么？](https://blog.csdn.net/qq_38486203/article/details/80077844#%E5%88%86%E7%89%87%E6%98%AF%E4%BB%80%E4%B9%88)
  + [分片的设置](https://blog.csdn.net/qq_38486203/article/details/80077844#%E5%88%86%E7%89%87%E7%9A%84%E8%AE%BE%E7%BD%AE)
  + [分片个数（数据节点计算）](https://blog.csdn.net/qq_38486203/article/details/80077844#%E5%88%86%E7%89%87%E4%B8%AA%E6%95%B0%E6%95%B0%E6%8D%AE%E8%8A%82%E7%82%B9%E8%AE%A1%E7%AE%97)
  + [分片查询](https://blog.csdn.net/qq_38486203/article/details/80077844#%E5%88%86%E7%89%87%E6%9F%A5%E8%AF%A2)

**分片重要性**

ES中所有数据**均衡的存储**在集群中**各个节点的分片中**，会影响ES的性能、安全和稳定性， 所以很有必要了解一下它。

**分片是什么？**

简单来讲就是咱们在ES中所有数据的文件块，也是**数据的最小单元块**，整个ES集群的核心就是对所有分片的分布、索引、负载、路由等达到惊人的速度

实列场景：

假设 IndexA 有2个分片，我们向 IndexA 中插入10条数据 (10个文档)，那么这10条数据会尽可能平均的分为5条存储在第一个分片，剩下的5条会存储在另一个分片中。

和主流关系型数据库的表分区的概念有点类似，如果你比较熟悉关系型数据库的话。

**分片的设置**

创建 IndexName 索引时候，在 Mapping 中可以如下设置分片 ([curl](https://www.baidu.com/s?wd=curl&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd))

PUT indexName

{

"settings": {

"number\_of\_shards": 5

}

}

* 1
* 2
* 3
* 4
* 5
* 6

注意

**索引建立后，分片个数是不可以更改的**

**分片个数（数据节点计算）**

分片个数是越多越好，还是越少越好了？根据整个索引的数据量来判断。

实列场景：

如果 IndexA 所有数据文件大小是300G，改怎么定制方案了？(可以通过Head插件来查看)

建议：（仅参考）

1、每一个分片数据文件小于30GB

2、每一个索引中的一个分片对应一个节点

3、节点数大于等于分片数

* 1
* 2
* 3
* 4
* 5
* 6
* 7

根据建议，至少需要 10 个分片。

结果： 建10个节点 (Node)，Mapping 指定分片数为 10，满足每一个节点一个分片，每一个分片数据[带下](https://www.baidu.com/s?wd=%E5%B8%A6%E4%B8%8B&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)在30G左右。

[**SN**](https://www.baidu.com/s?wd=SN&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)**(分片数) = IS(索引大小) / 30**

**NN(节点数) = SN(分片数) + MNN(主节点数[无数据]) + NNN(负载节点数)**

**分片查询**

我们可以指定es去具体的分片查询从而进一步的实现es极速查询。

1. **randomizeacross shards**

随机选择分片查询数据，es的默认方式

1. **\_local**

优先在本地节点上的分片查询数据然后再去其他节点上的分片查询，本地节点没有IO问题但有可能造成负载不均问题。数据量是完整的。

1. **\_primary**

只在主分片中查询不去副本查，一般数据完整。

1. **\_primary\_first**

优先在主分片中查，如果主分片挂了则去副本查，一般数据完整。

1. **\_only\_node**

只在指定id的节点中的分片中查询，数据可能不完整。

1. **\_prefer\_node**

优先在指定你给节点中查询，一般数据完整。

1. **\_shards**

在指定分片中查询，数据可能不完整。

1. **\_only\_nodes**

可以自定义去指定的多个节点查询，es不提供此方式需要改源码。