Hello 又见面了，中华石杉，

14，33，60，66，74

框架第一步：特性

1. 分布式
2. 高性能
3. 高可用
4. 可伸缩

Q:ES应该是针对文本的吧

Q：他是怎么提高搜索速度的

Q:一般搜索有哪些方式？

A:数据库写sql

Q：数据库检索有什么问题

A:就拿like来说，如果你用生化机，你是搜不出生化危机的

Q:idea find in path是怎么实现的

Q:拆词

Q：倒排索引，这个倒排是怎么体现的？

Q:ES需要代码开发么？

Q:倒排索引的有哪些基本概念？可以通俗的给我讲一下么？

Q：Elasticsearch和Lucene是什么关系？

A:简单的lucene只有建立索引和搜索功能，无法提供大数据量，容错性，高可用性，Elastic就是做的后者

Q:ES除了搜索还可以做什么

A:搜索包括全文检索，结构化搜索，数据分析，对海量数据进行分析P3 <https://www.bilibili.com/video/av29521652?p=3>

K:Index（库） 》 type（表） 》document（行）

Q:primary shard可以扩展么？

A:不可以，创建索引的时候就定了

Q：什么是面向文档的搜索引擎

Q：集群管理，需要整理

Q:常用api需要搜集

Q:聚合api可以嵌套下专，很厉害的（嵌套聚合，下专分析，聚合分析重要）

P8

K:shard 可以理解为kafka里面的分区

Q:扩容方案

Q:平等节点是什么？

A:所有的节点都会接受请求，自己有就自己给，没有就帮忙去拉一下

Q:shard 和replica不是一样的？

A:不一样的，还是可以像理解kafka那样

K:新节点加入会rebalance

K：如果只有一个节点，但是需要三个副本，你会发现副本无法分配，这个时候状态就是yellow，那么要是两个node节点的，三个副本会又会怎样分配呢？三个都会放入新加入的节点

Q:扩容有极限，难道不可以无限扩容么？

K：编程语言只是实现自己解决问题思路的工具，那么问题就是如何培养自己解决问题的思路

Q:全量替换，创建（其实是强制创建），delete

A：首先需要知道是，dcoument都是不可变的，所以当你对于同一个id创建的时候发生的其实是全量替换，原document会被标记delete，在es空间不足的时候会将其删除掉，当然你也可以不使用全量替换，使用强制创建，这个时候会有一些警告，需要在参数上加上一个参数，这个是可以查的。删除的原理跟全量替换很像，都不会真的删除都是先标记

P17该看P18

POST和PUT两种更新方式前者其实是创建了一个新的，用POST才是修改，一定要注意哦

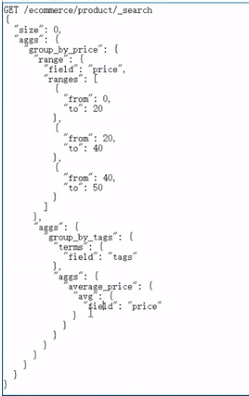
按照区间分组：

字段分组和区间分组都可以在ES中简单实现，聚合是一层层一层传递结果的

Q:乐观锁丢弃不对吧，如果丢弃的话，数据还是会出错的呀

A:这个时候会报错，必须重试，其实ES支持不使用其内部的乐观锁机制，external，实现版本号大于es就可以成功，这样便于编码，用内部的需要重试，而且不知道重试多少次

P22https://www.bilibili.com/video/av29521652/?p=23



K:使用\_update更新

K:使用groovy脚本

Q:\_update retry策略

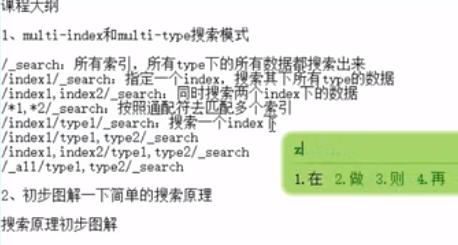
Q:bulk操作，bulk操作一个失败不会引起所有操作失效，但是会有异常

Q:bulk大小会对性能有所影响？什么东西大了也会有性能影响呀

Q:写一致性 quorum

<https://www.bilibili.com/video/av29521652/?p=34>

Q:timeout是如何保证快速查询的



还有这么多查询的方式

Q:shard到底有几种

Q:deep paging

Q：像MongoDB和es这样的应该是不适合做专门的分页查询

Q：+和-号搜索，-号就是不包含什么的数据

Q：\_allmetadata这个很重要，所有的字段值其实都会变成索引，在没有指定某个字段的时候，这个如果应用在结构化检索里面应该是可以的，当然这个用于?q=….查询语句

A:可通过q=“value”，不管字段

Q:问号查询？好像还强大些！关键知识点Mapping

K:同义词，缩写也可以，不仅仅是包含哟---🡪分词器很重要

回炉重造mapping，43要在仔细下，maping，dynamic mapping

K:bool –must should查询，知道一下应该很有用

Q:term query,terms query exist query（这三个其实不常用）分别是什么（exist甚至已经被弃用）

Q:bool和bulk有啥区别，我现在已经记不清了

Q:String 类型的排序因为分词的原因，可能很难达到预期的排序效果

Q:搜索流程

Q:每个shard都要分配同样长度的queue，每个shard要是没有这么长呢？

Q：查询路由规则

Q:滚动搜索的实现思路

Q:elatsearch的搜索原理到底是什么哟？

K:dynamic mapping =🡺可以手动设置类型以及是否分词，分词的方式

K：设置为keyword可以让其不被分词

Q:有哪几种query的方式，query dsl，？参数，bulk，bool(must,should)

Q：query和filter区别

A:filter只会过滤数据不会根据相关度去排序，而query会。

# Q：查询含有filter过滤查询和不含有filter查询

# A:可以看到，不使用filter依然可以实现一定的filter功能，但两者性能是不一样的，前者通过纯query实现，是会计算sores，而后者不会，只是单纯的过滤数据。

|  |
| --- |
| GET gather-033\_112000/\_search{  "size": 0,  "query": {    "range": {      "memrate": {        "gte": 0,        "lte": 100      }    }  }}GET gather-033\_112000/\_search{  "size": 0,  "query": {    "bool": {      "filter": {        "range": {          "memrate": {            "gte": 0,            "lte": 100          }        }      }    }  }} |

Q：string类型的排序会有什么问题？

A:受分词的影响，如果一个value中是多个word，那么就不知道是哪个word影响了sore导致最后query的结果不是自己的预期。解决的方式一般为：手动设置mappig，增加field配置



K：doc value 正派索引，用来排序，一般索引创建会同时创建正排索引和倒排索引

K:搜索关键组件：coordinate node，请求路由组件（kafka里也有这个东西）

Q:replica也会处理请求么？

A:replica只会承载读请求

|  |
| --- |
| 1）index包含多个shard  2）每个shard都是一个最小工作单元，承载部分数据，lucene实例，完整的建立索引和处理请求的能力  3）增减节点时，shard会自动在nodes中负载均衡  4）primary shard和replica shard，每个document肯定只存在于某一个primary shard以及其对应的replica shard中，不可能存在于多个primary shard  5）replica shard是primary shard的副本，负责容错，以及承担读请求负载  6）primary shard的数量在创建索引的时候就固定了，replica shard的数量可以随时修改  7）primary shard的默认数量是5，replica默认是1，默认有10个shard，5个primary shard，5个replica shard  8）primary shard不能和自己的replica shard放在同一个节点上（否则节点宕机，primary shard和副本都丢失，起不到容错的作用），但是可以和其他primary shard的replica shard放在同一个节点上 |

Q:preference

<https://elasticsearch.cn/article/334>