# ES核心概念和原理

## 1、什么是搜索：百度、垂直搜索（站内搜索）

搜索：通过一个**关键词**或一段描述，得到你想要的（相关度高）结果。

## 2、如何实现搜索功能?

关系型数据库：性能差、不可靠、结果不准确（相关度低）

### 例子

想要查询 “小米NFC智能手机”

|  |  |  |
| --- | --- | --- |
| id | brandName | describe |
| 1 | 小米手机 | 手机中的战斗机 |
| 2 | 小米NFC手机 | 小米手机,支持全功能NFC,手机中的战斗机 |
| 3 | NFC手机 |  |
| 4 | 小米耳机 |  |
| 5 | 红米耳机 |  |
| ...... |  |  |
| ...... |  |  |
| 10000000 | Xxx机器人 | 扫地好帮手 |
| ...... |  |  |

如果是sql查询，select \* from product where brandName like “%小米NFC智能手机%”，这种查询不走索引，会进行全表扫描，时间复杂度O(n) ，如果有1000w条数据就要有1000w次IO。再牛B的数据库也顶不住。

而且brandName字段数据也没有“小米NFC智能手机”导致搜不出来。

## 倒排索引、Lucene和全文检索

### 引入倒排索引

倒排索引是基于document做的，这里为了方便理解用表做。

倒排索引是Lucene封装的一个功能，分词的第一步就是把词拆开，把上面的商品的brandName进行分词：

小米手机 ----------- 小米，手机

小米NFC手机 ----------- 小米，NFC，手机

NFC手机 ----------- NFC ，手机

小米耳机 ---------- 小米，耳机

红米耳机 ---------- 红米，耳机

这一行类似es的document list

|  |  |  |
| --- | --- | --- |
| 是否匹配到 | 索引 | 出现id |
| √ | 小米 | 1,2,4 |
| √ | 手机 | 1,2,3 |
| √ | NFC | 2，3 |
|  | 耳机 | 4，5 |
|  | 红米 | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| id | brandName | describe | 在匹配到结果中索引中出现次数  （匹配相关度） |
| 1 | 小米手机 | 手机中的战斗机 | 2 |
| 2 | 小米NFC手机 | 小米手机,支持全功能NFC,手机中的战斗机 | 3 |
| 3 | NFC手机 |  | 2 |
| 4 | 小米耳机 |  | 1 |
| 5 | 红米耳机 |  | 0 |
| ...... |  |  |  |
| ...... |  |  |  |
| 10000000 | Xxx机器人 | 扫地好帮手 |  |
| ...... |  |  |  |

### 倒排索引数据结构

#### 关系数据库和ES类比

关系型数据库 -> Databases -> Tables -> Rows -> Columns

Elasticsearch -> Indices -> Types -> Documents -> Fields

#### ES数据结构

1. 包含这个关键词的document list
2. 关键词在每个doc中出现的次数 TF term frequency （词频）
3. 关键词在整个索引中出现的次数 IDF inverse doc frequency
4. 关键词在当前doc中出现的次数
5. 每个doc的长度，越长相关度越低
6. 包含这个关键词的所有doc的平均长度

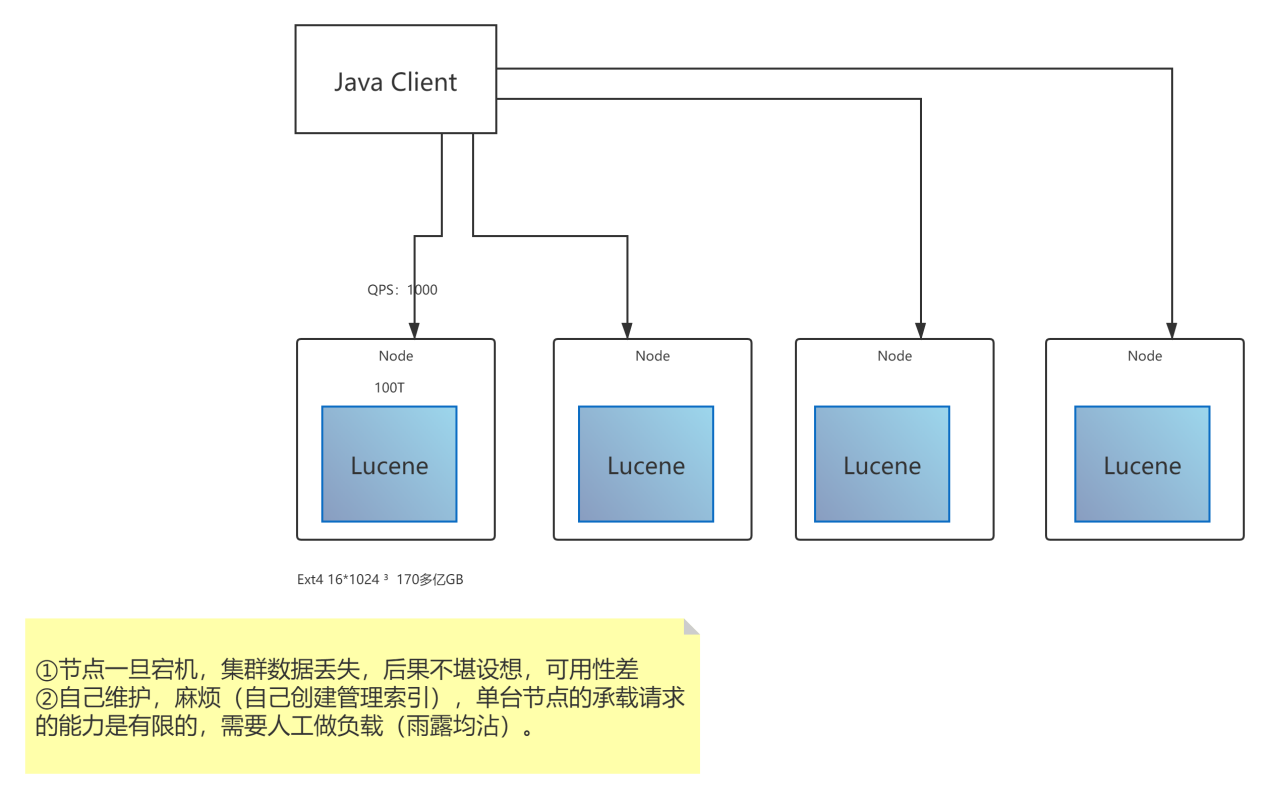
### Lucene

#### 概念：

Lucene是一个jar包。帮我们创建了倒排索引，提供了复杂的API

#### Lucene做集群实现搜索存在问题

1. 节点一旦宕机，节点数据丢失，后果不堪设想，可用性差。
2. 自己维护，麻烦（自己创建管理索引），单台节点的承载请求的能力是有限的，需要人工做负载（雨露均沾）。



## Elasticsearch：

### 概念

**分布式、高性能、高可用、可伸缩、易维护** ES≠搜索引擎

### 优点：

面向开发者友好，屏蔽了Lucene的复杂特性，集群自动发现（cluster discovery）

自动维护数据在多个节点上的建立

会帮我做搜索请求的负载均衡

自动维护冗余副本，保证了部分节点宕机的情况下仍然不会有任何数据丢失

ES基于Lucene提供了很多高级功能：复合查询、聚合分析、基于地理位置等。

对于大公司，可以构建几百台服务器的大型分布式集群，处理PB级别数据；对于小公司，开箱即用，门槛低上手简单。

相遇传统数据库，提供了全文检索，同义词处理（美丽的cls>漂亮的cls），相关度排名。聚合分析以及海量数据的近实时（NTR）处理，这些传统数据库完全做不到。