

## 1、es 如何进行分页搜索的语法

关键字 size、from

size: 每页大小      from: 分页开始位置

```
1 GET /_search?size=10           //查出前十条
2 GET /_search?size=10&from=0    //从第 0 条开始，查出size为 10 的分页数据
3 GET /_search?size=10&from=20   //从第 20 页开始，查出 size 为10的分页数据
```

## 2、什么是 deep paging 问题？为什么会产生这个问题，它的底层原理是什么？ deep paging 性能问题，以及原理深度图解

### 1、分页查询原理：

你的请求首先可能是发送到一个不包含这个index的shard的node上去，这个node就是一个coordinate node，那么多个coordinate node就会将搜索请求转发到一index的三个shard所在的node上面去。

比如说我们刚才说的那个情况下，要搜索60000条数据数据中的1000页，实际上每个shard都要将内部的20000条数据中的第10001~10010条数据拿出来，不是才10条是10010条数据。3个shard每个shard都要返回10010条数据给coordinate node，coordinate node会将收到的总共30030条数据给coordinate node，coordinate node会将收到的30030条数据，然后再这些数据中按照\_score, 相关度分数，然后去到排位10001~10010条数据，其实就是我们要的结果。。

### 2、带来的性能问题

搜索的过深的时候，就需要在coordinate node上保存大量的数据，还要进行大量的数据的排序，再取出对应的那一页，所以这个过程，即耗费网络的带宽，耗费内存，还耗费cpu。所以deep paging的性能问题。

