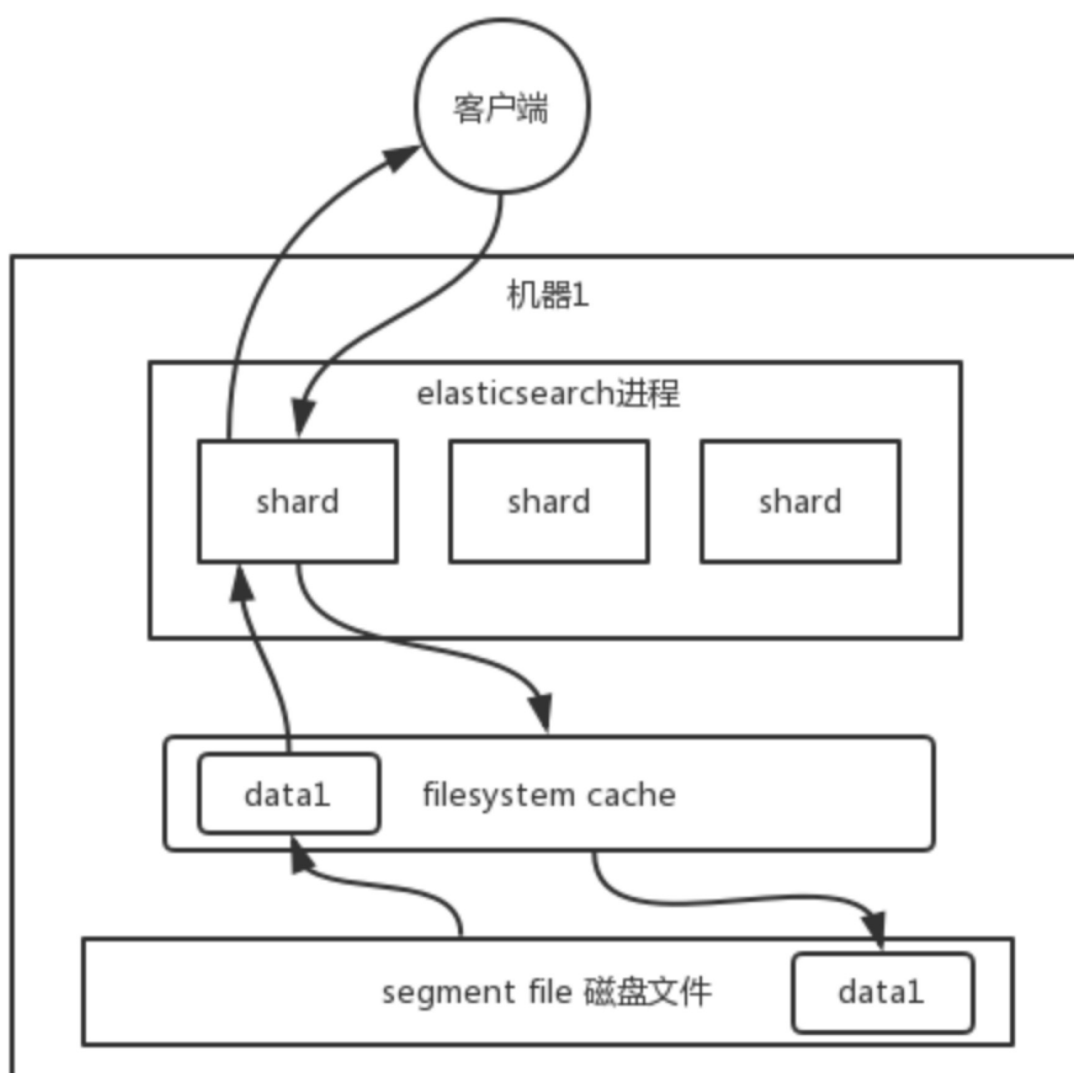


es在数据量很大的情况下，如何提升查询效率？

很多时候数据量大了，特别是几亿条数据的时候，第一次搜索的时候，是5~10s，后面反而就快了，可能就几百ms了。

对于es的调参，不需要想着能随手调一个参数就能起到万能的作用。

性能优化的杀手锏---filesystem cache



es的索引严重依赖于底层的filesystem cache，如果给filesystem cache更多的内存，尽量让内存可以容纳所有的ids segment file索引数据文件，那么搜索的时候就基本都是走内存的，性能会非常高。

性能差距究竟有多大？？如果走磁盘一般肯定得上秒，搜索性能绝对是秒级别的，但如果是走filesystem cache，是走纯内存的，那么一般来说性能比走磁盘要高一个数量级，基本就是毫秒级的。

es性能最佳的情况是机器的内存至少可以容纳总数据量的一半。

假如有一行数据。id, name, age...30个字段，但是现在搜索，只需要根据id, name, age三个字段来搜索。这样就是90%的字段是不用来搜索的，结果占据了filesystem cache的空间，单条数据的数据量越大，就会导致filesystem cache能缓存的数据就越少。其实，仅仅写入es中要用来检索的少数几个字段就可以了，比如说写入es id, name, age三个字段，然后你可以把其他的字段数据存在mysql/hbase里，一般建议是用es+hbase这么一个架构。通过三个字段在es中搜索出doc id，然后去hbase或者mysql中把完整的数据全部查出来。

数据预热：

假如说无论怎样es集群中写入的数据量还是超过了filesystem cache的一倍，那么就可以做数据预热。

看一些大V，平时看的人很多，可以搞一个后台系统，每隔一会儿，自己的后台系统去搜索一下热数据，然后刷到filesystem cache中，后面用户实际上来看这个热数据的时候，是直接从内存里搜索的，速度很快。

冷热分离：

es可以做mysql似的水平拆分，就是说将大量的访问很少、频率很低的数据，单独写一个索引，然后将访问很频繁的热数据单独写一个索引。最好是将冷数据写入一个索引中，然后热数据写入另外一个索引中，这样可以确保热数据在被预热之后，尽量都让他们留在filesystem os cache里，别让冷数据给冲刷掉。

看，假设你有 6 台机器，2 个索引，一个放冷数据，一个放热数据，每个索引 3 个 shard。3 台机器放热数据 index，另外 3 台机器放冷数据 index。然后这样的话，你大量的时间是在访问热数据 index，热数据可能就占总数据量的 10%，此时数据量很少，几乎全都保留在filesystem cache里面了，就可以确保热数据的访问性能是很高的。但是对于冷数据而言，是在别的 index 里的，跟热数据 index 不在相同的机器上，大家互相之间都没什么联系了。如果有人访问冷数据，可能大量数据是在磁盘上的，此时性能差点，就 10% 的人去访问冷数据，90% 的人在访问热数据，也无所谓了。

document 模型设计

对于 MySQL，我们经常有一些复杂的关联查询。在 es 里该怎么玩儿，es 里面的复杂的关联查询尽量别用，一旦用了性能一般都不太好。

最好是先在 Java 系统里就完成关联，将关联好的数据直接写入 es 中。搜索的时候，就不需要利用 es 的搜索语法来完成 join 之类的关联搜索了。

document 模型设计是非常重要的，很多操作，不要在搜索的时候才想去执行各种复杂的乱七八糟的操作。es 能支持的操作就是那么多，不要考虑用 es 做一些它不好操作的事情。如果真的有那种操作，尽量在 document 模型设计的时候，写入的时候就完成。另外对于一些太复杂的操作，比如 join/nested/parent-child 搜索都要尽量避免，性能都很差的。

