链滴

solr 性能优化

作者：llh

原文链接：https://ld246.com/article/1489212895750

来源网站：链滴

一.配置

1.schema.xml配置

schema配置不合理，往往会导致查询性能低，索引占用磁盘、内存空间大的问题。

1)合理设置字段属性

- 字段是否要检索(indexed)，是否要存储(stored)，按需配置，避免不必要的空间浪费。
- 段是否需要根据文本长度算分，是否需要在建索引时设置权重，如果不需要，设置omitNorms=true
- omitPositions、omitTermFreqAndPositions，词频信息和打分相关，位置信息和高亮显示相关当不需要这些功能，则可设置为true，节省磁盘空间，提升搜索速度。
- 对于需要排序的字段，使用docvalues，构造fieldCache会进行压缩，节省内存使用。

2)使用正确的数据类型

- 对于数值类型，使用能正确存储的最小数值类型，更小的数值类型占用更小的磁盘、内存、cpu缓，并且处理时的cpu周期也更少。
- 数值类型不要用string，一个整形占4字节，用string，大小为1000以上的整形就占了4个字节了。然，对于只有几个值(比如0、1、2、3）的可枚举的小整形，可以用string。
- 不需要分词的字符型，用string，不要用text，text默认用标准分词器分词。
- 需要范围查询的数值类型，需要使用tlong、tint等分精度索引的类型，范围查询性能是long、int不分精度索引的数值类型的10倍。当然，也不能滥用，分精度索引的数值类型比较占用空间，如果没范围查询的需求，则不需要使用。

2.solrconfig.xml配置

1)索引目录类型

- 采用NRTCachingDirectoryFactory，这种目录类型，小索引会缓存在内存中，减少磁盘IO。而且些小文件往往是频繁变化的，放在内存中，则reopen的时候，不需要读磁盘，性能会好很多。
- softcommit和hardcommit

hardcommit作用是使索引持久化，会flush当前正在索引的段到磁盘，比较重，影响查询性能，时间隔可以设置得较长。没有hardcommit，机器挂了，索引会否丢失？答案是不会，因为还有tlog，重后，会从tlog恢复。

<p>
softcommit作用是使索引可见，可根据实时性需要设置得适当的长度。需要注意，softcommit会致searcher层cache失效。索引实时性要求不高的情况下，频率尽可能设置长一点。

2)cache配置

缓存类型

● queryResultCache，查询结果缓存，key由q参数、fq参数、sort参数组成，value是docId和scor（score可能没有）的有序集合DocList。只要q、fq、sort参数有一个变化了，则属于不同的key，不命中结果。

● documentCache，document的缓存，key是docId，value是document。

● filterCache，filterQuery结果缓存，key是fq参数的值，value是docId的无序集合DocSet。

● fieldCache，正排索引缓存，可通过docId获取字段值。排序、facet、group等需要正排索引的查需要用到fieldCache。fieldCache是基于段的，一个字段的fieldCache是在第一次使用的时候加载到存中的。Lucene用一个weakHashMap存放已加载的段的fieldCache，key为段的indexReader。因，fieldCache是常驻内存，不会自动释放的，除非段被合并，不存在了，才会释放掉。使用时，要注内存的消耗，避免内存不够用，发生OOM。

缓存注意事项

● queryResultCache、documentCache、filterCache都是searcher级别的缓存，searcher重新打开(oftcommit会触发)，则缓存失效。其中queryResultCache、filterCache可以配置autowarn使searchr预热时重新加载。documentCache的key是docId，重新打开后，document的id已经变化，因此doumentCache不能进行autowarn。

● 对于实时查询（比如softcommit频率为1秒），最好不要配置cache，因为缓存失效太快，缓存命率可能比较低，如果配置了autowarn，还会导致不停的autowarn，加重服务器负担。除非查询语句比较集中，缓存条目很少。

● 对于有翻页的查询，可以适当调整queryResultWindowSize参数，比如一页的大小为10，则queryesultWindowSize设置为50，则后面4页，都会命中缓存，当然参数设置越大占用内存越多。

● 设置useFilterForSortedQuery=true，则对于没有按照score排序的查询，会用filterCache缓存主询结果（key为q参数），当然，如果主查询的命中数大多比较小，则没必要用这个缓存，可能还没有接查询快。

● enableLazyFieldLoading配置为true，只读取需要的字段。这个属性要配合documentCache使用即开启了documentCache，才能发挥lazyLoading的作用。举个例子，比如查询出来的条目可能只示简单的信息，点击具体条目，则需要把整个条目的所有信息展示出来。则点击具体条目去查询时，中了cache，获取到的document就是有lazyField的document了，solr把它返回给调用者时，就会去载lazyField的真实的值。

● 可配置firstSearcher、newSearcher来预热耗时的查询，使查询结果缓存起来，使查询性能更平滑。

二.索引

建议采用离线方式构建索引，再拷贝索引到引擎中。离线构建索引的好处：

● 避免在线构建索引对在线服务的影响。

● 不用考虑记录更新的情况，省掉查询老记录是否存在的步骤，性能有提升。

● 可以使用内存大的机器，在内存中（RAMDirectory）进行构建（内存放不下，可考虑一个段一个的构建），速度往往是在线构建的好几倍。

● 离线构建可以打开多个indexWriter进行构建(最好多个进程，因为IndexWriter存在类上加锁的情，多个线程还是存在锁等待)，然后再合并到一个index里，虽然在线索引也可以多个线程并发写，但存在并发锁的问题。

● 离线索引追求的是吞吐量，对响应时间要求不高，可按照高吞吐量调配jvm参数，而在线索引还得虑jvm对在线查询的影响。

● 离线构建可减少中间的段的生成，可以调大ramBufferSzie和mergeFactor，避免生成小的段，避

在构建的过程中自动合并，在构建完成后再根据需要触发合并动作，省掉中间的合并过程，速度也有定的提高。

三.搜索

- 如果不是所有字段都需要返回，则明确指定需要返回的字段，减少系统开销。
- 一次返回的记录数不要太多，深度翻页使用cursorMark参数提升性能。
- 查询条件中包含路由字段的，可以先计算出分片，再从分片中获取机器，并指定shard来查询，只询相应的shard，避免服务端所有shard都查询。
- 合理使用filterquery，filterquery结合filtercache，能把常用的查询条件缓存起来，提升查询效率对于不使用filtercache的情况下，多个OR条件组合起来的查询也应该用filterquery，因为filterquery不算分的，性能会好很多。主查询中的OR查询，对于一个document并不是有一个OR查询条件命中返回，接着去查下一个document，而是所有的OR条件都去查一遍，记录命中的OR条件的数量，由来计算得分，所以会比在filterQuery中查询慢很多。
- facet query有几种实现方式，需要根据具体情况选择。
    - enum，通过filterCache缓存field中每个term的docset，然后和查询结果的docset进行交集来算term命中的数量，对于term比较少的情况比较合适。这种方式适用于multiValue。
    - fc，通过fieldcache计算，遍历查询结果的docset，根据docid从fieldcache获取值进行计算，里的fieldcache是searcher级别的，这样能避免每个segment的计算结果进行合并，但对于实时查询searcher频繁reopen，需要频繁构造fieldcache，性能会有波动，可以注册newSearcher监听器来解这个问题。
    - fcs，原来和fc一样，但用的是底层segment级别的fieldcache，需要对每个段的计算结果进行并，性能比fc差些，但适用于实时搜索。

四.系统

- 内存分配

合理设置jvm内存大小，solr缓存、lucene缓存、词典文件等都需要加载到内存中，内存设置太小，造成gc频繁。但也不要把内存用得太满，在jvm内存足够的情况下，多留点给操作系统做文件缓存，样索引文件能更多地被操作系统缓存起来，减少磁盘IO，提升搜索的性能。

- swap

系统内存不足时，操作系统会拿交换区(磁盘)来当做内存使用，交换区的访问速度和内存比非常慢，影响搜索的性能，对gc也有较大的影响，jvm收集交换区的垃圾对象时，常常速度很慢，造成应用停。建议内存足够的情况下，把swap关掉。如果内存比较紧张，则建议把swapness参数的值调小，让作系统尽量少使用swap，完全关掉可能会造成操作系统内存不足，进程而被操作系统kill掉。

- 磁盘

搜索引擎对磁盘的随机访问比较多，推荐使用ssd磁盘。