

elasticsearch

加入尚学堂，一起进步！



- Elasticsearch分片查询
- 默认是randomize across shards
 - 随机选取，表示随机的从分片中取数据
- `_local`：指查询操作会优先在本地节点有的分片中查询，没有的话再在其它节点查询。
- `_primary`：指查询只在主分片中查询
- `_primary_first`：指查询会先在主分片中查询，如果主分片找不到（挂了），就会在副本中查询。
- `_only_node`：指在指定id的节点里面进行查询，如果该节点只有查询索引的部分分片，就只在这部分分片中查找，所以查询结果可能不完整。如 `_only_node:123` 在节点id为123的节点中查询。
- `_prefer_node:nodeid` 优先在指定的节点上执行查询
- `_shards:0,1,2,3,4`：查询指定分片的数据
- 自定义：`_only_nodes`：根据多个节点进行查询



- Elasticsearch脑裂问题
- 所谓脑裂问题（类似于精神分裂），就是同一个集群中的不同节点，对于集群的状态有了不一样的理解。
- <http://blog.csdn.net/cnweike/article/details/39083089>
- `discovery.zen.minimum_master_nodes`
 - 用于控制选举行为发生的最小集群节点数量。推荐设为大于1的数值，因为只有在2个以上节点的集群中，主节点才是有意义的



- Elasticsearch中脑裂问题
- 正常情况下，集群中的所有的节点，应该对集群中master的选择是一致的，这样获得的状态信息也应该是一致的，不一致的状态信息，说明不同的节点对master节点的选择出现了异常——也就是所谓的脑裂问题。这样的脑裂状态直接让节点失去了集群的正确状态，导致集群不能正常工作。



- Elasticsearch中脑裂产生的原因
- 1. 网络：由于是内网通信，网络通信问题造成某些节点认为master死掉，而另选master的可能性较小
- 2. 节点负载：由于master节点与data节点都是混合在一起的，所以当工作节点的负载较大时，导致对应的ES实例停止响应，而这台服务器如果正充当着master节点的身份，那么一部分节点就会认为这个master节点失效了，故重新选举新的节点，这时就出现了脑裂；同时由于data节点上ES进程占用的内存较大，较大规模的内存回收操作也能造成ES进程失去响应。



- Elasticsearch中脑裂解决
- 主节点
- `node.master: true`
- `node.data: false`
- 从节点
- `node.master: false`
- `node.data: true`
- 所有节点
- `discovery.zen.ping.multicast.enabled: false`
- `discovery.zen.ping.unicast.hosts: ["master" , "slave1" , "slave2"]`



• Elasticsearch的优化

- 调大系统的"最大打开文件数",建议32K甚至是64K
 - ulimit -a (查看)
 - ulimit -n 32000(设置)
- 修改配置文件调整ES的JVM内存大小
 - 1: 修改bin/elasticsearch.in.sh中ES_MIN_MEM和ES_MAX_MEM的大小, 建议设置一样大, 避免频繁的分配内存, 根据服务器内存大小, 一般分配60%左右(默认256M)
 - 2: 如果使用searchwrapper插件启动es的话则修改bin/service/elasticsearch.conf(默认1024M)
- 设置mlockall来锁定进程的物理内存地址
 - 避免交换 (swapped) 来提高性能
 - 修改文件conf/elasticsearch.yml
 - bootstrap.mlockall: true
- 分片多的话, 可以提升建立索引的能力, 5-20个比较合适。
- 如果分片数过少或过多, 都会导致检索比较慢。分片数过多会导致检索时打开比较多的文件, 另外也会导致多台服务器之间通讯。而分片数过少会导至单个分片索引过大, 所以检索速度慢。建议单个分片最多存储20G左右的索引数据, 所以, 分片数量=数据总量/20G
- 副本多的话, 可以提升搜索的能力, 但是如果设置很多副本的话也会对服务器造成额外的压力, 因为需要同步数据。所以建议设置2-3个即可。
- 要定时对索引进行优化, 不然segment越多, 查询的性能就越差
 - 索引量不是很大的话情况下可以将segment设置为1
 - curl -XPOST 'http://localhost:9200/bjsxt/_optimize?max_num_segments=1'
 - java代码: client.admin().indices().prepareOptimize("bjsxt").setMaxNumSegments(1).get();



- Elasticsearch的优化
- 删除文档：在Lucene中删除文档，数据不会马上在硬盘上除去，而是在lucene索引中产生一个.del的文件，而在检索过程中这部分数据也会参与检索，lucene在检索过程会判断是否删除了，如果删除了在过滤掉。这样也会降低检索效率。所以可以执行清除删除文档
 - `curl -XPOST 'http://localhost:9200/elasticsearch/_optimize?only_expunge_deletes=true'`
 - `client.admin().indices().prepareOptimize(" elasticsearch").setOnlyExpungeDeletes(true).get();`
- 如果在项目开始的时候需要批量入库大量数据的话，建议将副本数设置为0
 - 因为es在索引数据的时候，如果有副本存在，数据也会马上同步到副本中，这样会对es增加压力。待索引完成后将副本按需要改回来。这样可以提高索引效率



- Elasticsearch的优化
- 去掉mapping中 _all域，Index中默认会有 _all的域，(相当于solr配置文件中的拷贝字段text)，这个会给查询带来方便，但是会增加索引时间和索引尺寸
 - "_all":{"enabled":"false"}
- log输出的水平默认为trace，即查询超过500ms即为慢查询，就要打印日志，造成cpu和mem，io负载很高。把log输出水平改为info，可以减轻服务器的压力。
 - 修改ES_HOME/conf/logging.yaml文件
 - 或者修改ES_HOME/conf/elasticsearch.yaml



- 使用反射获取Elasticsearch客户端
- 可以使用前面讲的方式通过new获取client
- 使用反射方式：网上反映这种方式效率明显高于new客户端，并可避免线上环境内存溢出和超时等问题



- Elasticsearch使用经验谈
- 在使用java代码操作es集群的时候要保证本地使用的es的版本和集群上es的版本保持一致。
- 保证集群中每个节点的JDK版本和es配置一致



- Elasticsearch的分片规则
- elasticsearch在建立索引时，根据id或id，类型进行hash，得到hash值与该索引的文档数量取余，取余的值即为存入的分片。
 - 具体源码为：根据OperationRouting类的shardId方法进行分片

