|  |  |
| --- | --- |
| **问题** | **答案** |
| 什么是脑裂？ | 一个es集群出现多个master节点构成多个分裂集群的情况。通过将discovery.zen.minimum\_master\_nodes设置为半数以上节点数避免脑裂的出现 |
| ES如何实现分布式存储和容错(shard/replica) | 分布式存储：将数据切分成多个shard以实现分布式扩展；  容错：每个shard可以设置1到多个replicate shard即副本，通过数据冗余实现容错 |
| Shard分片大小 | shard大小=索引总量/primary分片数  如何设定primary 分片数量？考虑如下因素：   * 预估所需分片数：索引总量 / 每个节点的处理上限（一般不超过ES的最大JVM对空间设置） * 在不超过处理上限的前提下，尽量设置小的primary shard分片数，以提高查询效率 |
| 解释一下index refresh 操作 | 作用： 每隔1秒将内存中的索引数据写入segment中，使数据可以被搜索到  如果设置为-1会如何？   * 数据写入index memory buffer和translog后不会刷新生成新的segment，只有在flush的时候才会将数据写入segment并保存到磁盘。可以显著提高索引性能，但是查询会有滞后 |
| 如何提高索引性能 | 1. 增加merge.policy.merge\_factor数：设置merge.policy.merge\_factor到30，初始是10，增加这个数需要更多的内存，bulk index可以调大这个值 2. 使用批量插入 3. 增加 segment refresh 间隔：index.refresh\_interval 4. 在索引时设置replicate shard为0，索引完成后再设置副本数量 5. 增加translog大小配置或者关闭translog功能（可能造成数据丢失） 6. 关闭 \_all 字段 |
| 如何提高查询性能？ | 1. 尽量使用最新的版本 2. 通过预查询将查询结果缓存 3. 尽量使用 filter 而不是 query 4. 增加replicate shard数量 5. 尽量减少返回的数据量（字段，文档数） |
| 解释一下fielddata | * 是什么：在执行某个查询的时候，如果需要基于某个字段排序、聚合计算等操作，会将包含这个字段的所有document加载到内存中。其内部结构是一个document id 到 document 字段的map。默认情况下其占用内存是没有上限的，即有多少用多少！ * 作用：加快获取值的操作，在所有需要查看一个文档里面某个字段值的时候都会用到，包括：排序，聚合，脚本查询 * 影响：   1. 加快排序，aggr等操作  2. 需要消耗大量的内存   * 最佳实践：   1. 配置预加载，在node启动时加载数据到内存，提高查询时的性能，有两种方式实现预加载：  1. 配置字段的 "fielddata"."loading" 属性为 "eager"：这样ES会在新的segment创建的时候加载 fielddata.  2. 预查询：在用户使用前执行预定的查询，促使ES加载fielddata  2. 设置内存占用的上限，以防止OOM  3. 对于不需要的字段将 index 属性设置为 no，降低内存占用  4. 尽量减少会被加载到fielddata中的字段的大小，如果字段值范围有限，可以将其映射为数值，降低内存占用 |
| Analyzer作用，由哪几部分组成？ | 作用：在索引和查询时将输入数据做分词处理  组成：0到n个char filter，一个tokenizer和0到n个token filter组成，其中char filter负责对原始输入数据进行替换、删减等操作，tokenizer负责对对数据切词，token filter负责对切分后的token进一步处理，如大小写、近似词等 |
| Mapping的作用是什么？ | Mapping相当于数据库中的表结构，指定了索引中的所有字段，每个字段的属性，analyzer等。如果在创建索引时不设定mapping则es将根据第一个索引的数据自动判断字段类型，为避免处理逻辑与预期不一致，通常建议创建索引前手动设定mapping。 |