# elasticsearch常见问题文档

## 1 7.x集群规模⼤且磁盘IO较差的集群节点掉线后可能处于进进出出的不稳定状态

原因：磁盘io不给力，元数据更新超时

方案：调整超时时间

cluster.follower\_lag.timeout: "600s"

cluster.join.timeout: "600s"

cluster.publish.timeout: "600s"

## 2 集群OGC严重，甚⾄掉节点

方案：

1.取消search

POST \_tasks/\_cancel?action=\*search

2.分析slowlog

打开slowlog

PUT /\_all/\_settings

{

"index.serach.slowlog.threshold.query.warn": "1s",

"index.serach.slowlog.threshold.query.info": "500ms",

"index.serach.slowlog.threshold.fetch.warn": "1s",

"index.serach.slowlog.threshold.fetch.info": "500ms",

"index.serach.slowlog.level": "info",

}

<https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules->

slow.html

## 3 分片迁移缓慢

方案：适当提升分片迁移速度，可根据实际情况使用

PUT \_cluster/settings

{

"persistent": {

"cluster.routing.allocation.node\_concurrent\_recoveries": "10",

"cluster.routing.allocation.node\_initial\_primaries\_recoveries": "30",

"cluster.routing.allocation.cluster\_concurrent\_rebalance": "10",

"indices.recovery.max\_bytes\_per\_sec": "100mb"

}

}

<https://www.elastic.co/guide/en/elasticsearch/reference/current/modules->

cluster.html

## 4 集群状态为yellow

方案：

检查集群状态：GET \_cluster/health?v

检查未分配原因：GET \_cat/allocation/explain

查看分配进度：GET \_cat/recovery?active\_only=true?v

重试分配失败的分片：POST \_cluster/reroute?retry\_failed

## 5 搜索超时

原因:

1.常见原因是磁盘 I/O 速度慢，导致搜索排队或在某些情况下 CPU 完全饱和。

2.大型数据集或占用大量内存的查询。

3.我们有时候写查询，为了图方便，经常使用通配符\*来查询，这有可能会匹配到多个索引，由于索引下分片太多，超过了集群中的核心数。就会在搜索线程池中造成排队任务，从而导致搜索拒绝。

方案：

1.elasticsearch支持同时查询多个索引，为了提高查询效率，避免使用通配符查询

2.不要将请求的 size 值设置太大，结合：from、size 深度翻页机制实现,但要注意from/to查询带来的深度分页问题。全量遍历借助 scroll API 实现。

3.缩小时间范围, 请求时间范围越长（比如 时间跨度周期 1 年以上的数据），请求数据量越大，超时的可能性越高。

4.调整内存设置,通过配置查询的内存断路器来限制查询的内存使用量。

如：将 index.breaker.request.limit 限制为 40％，默认是 60%。

通过将search.max\_buckets设置为 5000 （默认值：10000）来限制用于聚合的存储桶数。

5.通过在Elasticsearch中启用慢查询日志，我们可以在日志中看到，哪些查询语句耗时长，从而制定优化措施如使用filter上下文查询等。

6.集群中应该有独立的协调节点，专门用于数据请求，并给它们设置足够的内存。通过数据节点与协调节点分离(必要时可将协调节点读写分离)，可以避免节点挂掉之后，导致整个集群不可用，或者长时间响应迟钝。

## 6 写入拒绝或查询拒绝

方案：

查看拒绝个数

GET \_cat/thread\_pool/bulk?s=queue:desc&v

GET \_cat/thread\_pool/search?s=queue:desc&v

1.检查 bulk 请求的 body 大小是否不合理。单个 bulk 请求的大小在10MB以内比较合适，如过大，则会导致单个 bulk 请求处理时间过长，导致队列排满；如过小，则会导致 bulk 请求数过多，导致队列排满。

2.检查写入 QPS 和集群配置是否匹配，经验值为在4C16G 3节点集群上分片分布均衡时可以承担约2W - 3W QPS 的写入，但如果还有较多的查询请求时 QPS 会更低，具体可以通过压测确定集群最高能承受的 QPS 写入量，选择合适的配置。

3.检查分片（shard）数据量是否过大。分片数据量过大，有可能引起 Bulk Reject，建议单个分片大小控制在20GB - 50GB左右。可在 kibana 控制台，通过命令查看索引各个分片的大小。

GET \_cat/shards?index={index\_name}&v

## 7 集群磁盘使用率高

原因：

当集群磁盘使用率超过85%：会导致新的分片无法分配。

当集群磁盘使用率超过90%：Elasticsearch 会尝试将对应节点中的分片迁移到其他磁盘使用率比较低的数据节点中。

当集群磁盘使用率超过95%：系统会对 Elasticsearch 集群中对应节点里每个索引强制设置 read\_only\_allow\_delete 属性，此时该节点上的所有索引将无法写入数据，只能读取和删除对应索引。

方案：

1.删除历史数据

DELETE index-name-\*

2.关闭索引只读状态

PUT \_all/\_settings

{

"index.blocks.read\_only\_allow\_delete": null

}

## 8 bulk过大导致写入ES失败

原因：

一次bulk写入数据量大于100M ES就会报错并拒绝写入

方案：

建议一次bulk写入在100M以内，也可以修改ES参数用以支持一次写入更多数据量

调整参数：http.max\_content\_length，默认100M