# 批量新增优化

在单线程下测试了10w条bulk插入性能，约40s。则大数据60e数据量在此效率下大约要666min。

故需要尽量优化bulk插入性能。

## 1.refresh间隔

调整refresh时间间隔，优化点： 减少刷新频率，降低潜在的写磁盘性能损耗， 默认的刷新时间间隔是1s，对于写入量很大的场景，这样的配置会导致写入吞吐量很低，适当提高刷新间隔，可以提升写入量，代价就是让新写入的数据在60s之后可以被搜索，新数据可见的及时性有所下降。

## 2.replica数目

调整replica数目，在bulk大量数据到ES集群的可以把副本数设置为0，在数据导入完成之后再设置为1或者你集群的适合的数目。

## 3. 去掉\_all字段

Index中默认会有\_all这个字段（es6.x已经禁用），默认会把所有字段的内容都拷贝到这一个字段里面，这样会给查询带来方便，但是会增加索引时间和索引尺寸。

## 4.translog优化

Elasticsearch 2.0之后为了保证不丢数据，translog的持久化策略是，对于每个 index、bulk、delete、update请求都做一次flush（刷新translog数据到磁盘上）。

translog默认为512MB，flush操作达到512MB fsync刷新到硬盘（这个问题不大），而translog是默认是每次index请求都写磁盘，优化点： 减少写磁盘的频率，调整为index.translog.durability=async，index.translog.sync\_interval=30s（默认值是5s）

## 5.segment合并

Lucene的index由多个segment组成，lucene会不断把一些小segment合并成一个大segment。执行索引创建时，es会先生成小 segment，离线时会有逻辑对小segment进行合并，优化查询性能，但合并过程消耗较多磁盘IO，影响查询。

可以通过设置\_settings中的"index.merge.scheduler.max\_thread\_count": 2来调整merge过程的并发度。

## 6.mapping优化

由于es自动检测和添加新字段称为动态映射，我见过由于环境问题发生mapping漏设置，这时有300~400个字段自动映射为text类型（太多），导致每批次（5000条）插入响应时间都大于3分钟。总之，检查mapping的合理性很重要。

## 7.ES的JVM内存大小

修改配置文件调整ES的JVM内存大小，这个值不能超过32g，一般机器好点设置成十几个g速度就非常快了。具体要看自己机器的内存。