# 中国联通XX软件研究院

# elasticsearch使用规范文档

# 

Elasticsearch是文档型数据库，不是关系型数据库，不具备严格的ACID事务特性，任何企图直接替代严格事务性场景的应用项目都会失败！

## 1 集群

* 集群状态为green时健康，yellow时副本异常，red时主分片异常
* 注意集群节点间网络通信波动较大持续较长时节点认为主节点死掉从而造成集群脑裂
* 集群规模较大时建议集群节点角色分离

## 2 节点

* 为了防止无意中破坏集群稳定的操作，ES数据节点管理的分片数默认不超过1000个
* ES数据节点会消耗2部分内存：1部分是 JVM 运行内存，起步设置就固定，内部划分为很多个应用块1部分是文件系统缓存，频繁挂载物理索引数据，会采用操作系统文件缓存，提升性能
* 单个节点JVM内存最大不超过32GB

## 3 索引

* 大索引需要拆分：增强性能，风险分散。
* 分片数量多，写入并行度增加，提升写入性能，但查询复杂度增加，降低查询性能
* 单分片数据容量在20~40GB最优，上限不超过50GB,数据条数限制上限21亿条
* 数据量未超过单分片容量尽量不增加分片数
* 聚合统计场景下，若单节点聚合性能低，可用考虑增加分片，提升聚合的并行度，提升性能，需要结合实际情况去测算设计
* 单个索引分片数量上限为128个
* 一个主分片可以有多个副本分片，原则是无上限限制
* 副本数量默认1，具备初级的高可用
* 副本数量多，查询可以提供查询并行度增加QPS，索引高可用大大提高
* 副本数据量，写入数据意味着写入多份，变相的降低了数据写入效率
* 可以在写入前后动态调整副本的数量
* 索引主分片尽量均衡分布在数据节点上
* 索引主分片若在某节点分布多个，尽量手动移动到其它节点
* 每个分片都有使用内存和 CPU 资源开销，在大多数情况下，一小组大分片比许多小分片使用更少的资源。

## 4 索引mapping

* mapping是定义文档及其包含的字段如何存储和索引的过程，因此在建立索引mapping时需要确定每个实体的属性、关系和约束等。
* 不要使用默认的动态字符串映射: 默认的[动态字符串映射](http://invalid.uri)会将字符串字段索引为[text](http://invalid.uri)和[keyword](http://invalid.uri)类型，如果您只需要其中一个，这将是一种浪费。
* 在索引中定义过多的字段会导致映射爆炸，这会导致内存不足错误。
* 使用[映射限制设置](http://invalid.uri)来限制字段映射的数量（手动或动态创建）
* 字段映mapping建立好后不可更改，如需更改请使用reindex操作
* 复杂的索引文档模型，查询与维护的代价更高
* 尽量简化索引文档模型，越简单越好
* 避免使用nested与join类型，或更高级的类型
* 字符串text与keyword优先选择keyword类型
* 数值类型尽量选择短精度的，节约存储空间
* 数值类型若无范围检索需求，若无计算需求，尽量使用 keyword 类型，倒排查询效率远远强于 bkd 树
* 创建索引尽量采用模板，固定住索引字段类型
* text 字段类型分词，分词器的选择对于写入性能影响很大，越复杂的分词器写入越慢
* 非text字段类型默认存储了2份，1个是source，1个是列式数据，依据场景需求可以选择禁用index，数据字段无需查询，可以禁用，节约索引信息资源
* doc\_value，数据字段支持排序与聚合，若无需求，可禁用，节约存储与内存
* fields，同一字段有多重查询需求，应用多字段类型，避免原始值存储多次，可节约存储与内存
* text 字段类型，有多个字段属性，若无此类需求，可以选择禁用，可大大节约存储、内存、CPU，如：norms 存储了词项的向量信息，查询时便于计算相关分值等。

## 5 refresh索引刷新

* ES的定位是准实时搜索引擎，刷新间隔默认是1s，表示写入后1秒后可被搜索到，所以这里的值取决于业务对实时性的要求，注意这里并不是越小越好，刷新频率高也意味着对ES的开销也大，通常业务类型在1-5s，日志型在30s-120s，如果集中导入数据可将其设置为-1，ES会自动完成数据刷新，完成后更改回来，否则后续会出现搜索不到数据
* 业务日志场景应用，若无实时查询要求，应该增大刷新间隔
* 大数据业务场景海量数据瞬间写入，数千万级数亿级，可以适当禁止刷新

## 6 限制数据大小

* 限制原始数据大小限制单条数据大小，单条数据勿超过 lkb
* 超过10kb，查询会明显变慢
* 限制单字段数据大小，单字段文本长度，限制最高 100mb，建议越短越好，1kb 以内最好。
* 若单条原始数据确实很大，建议使用 cassandra 列式数据库存储，ES 仅仅作为查询引擎，不存储原始数据

## 7 索引别名

* 在实际生产中，索引的设计可能不是一步到位的,随着业务的扩展，可能会在开发的中后期，调整索引Mapping结构，那么此时就需要借助reindex操作完成索引的迁移,如果要确保线上环境的可靠运行且用户无感知（无需告知用户，不影响用户的业务）的话,就要使用别名来指向更改前和更改后的索引

## 8 禁用source

[\_source](http://invalid.uri)字段存储文档的原始 JSON 正文。如果您不需要访问它，您可以禁用它以节约磁盘空间。但是，更新和重新索引等需要访问的 API\_source将不起作用。

## 9 避免大宽表

* ES索引字段数量默认最大1000，但建议不要超过100

## 10 数据查询注意

* 聚合查询的中间结果和最终结果都会在内存中进行，嵌套过多，会导致内存耗尽OOM

慎重使用模糊查询，ES计算都在JVM内存中完成

* 查询结果避免深度分页：ES每次默认最多获取10000条文档数据，获取更多分页存在深度分页问题，因为es的[http.max\_content\_length](http://invalid.uri)最大大小默认为100MB，es拒绝索引任何大于该值的文档，您可以决定通过修改该设置来增加大小，但 Lucene 仍然有大约 2GB 的限制。建议使用scroll或者searchAfter方式。scroll会把上一次查询结果缓存一定时间。
* 基数查询：尽量不要用基数查询去统计去重后的数据量大小（Unique Count，Distinct Count等）
* 慎重使用“索引名+\*”查询
* 尽量不使用script、update\_by\_query、delete\_by\_query，对ES性能影响的操作
* 慎重正则匹配查询

## 11 磁盘使用注意

为了确保所有节点都有足够的磁盘空间，而不会执行过多的分片移动。集群会根据节点磁盘使用水位去分配移动分片，磁盘使用水位主要有三个等级:

* 磁盘使用达到85%时停止副本分片分配: cluster.routing.allocation.disk.watermark.low：85%
* 磁盘使用达到90%时停止主分片分配：cluster.routing.allocation.disk.watermark.high ：90%
* 磁盘使用达到95%时停止数据写入，这会造成数据写入拒绝：cluster.routing.allocation.disk.watermark.flood\_stage：95%

## 12 codec 索引存储编码

* 索引压缩编码更换最好的压缩方式（[best\_compression](https://www.elastic.co/guide/en/elasticsearch/reference/7.9/tune-for-disk-usage.html)），.可大大节约磁盘空间,默认压缩选择的算法是LZ4， [best\_compression](https://www.elastic.co/guide/en/elasticsearch/reference/7.9/tune-for-disk-usage.html)选择的算法是DEFLATE

## 13 translog事务日志

* 事务日志达到默认阈值，会开始 flush，清空 translog
* 海量数据写入场景，会造成频繁的 flush，严重影响性能，可调大此值
* 若是固态硬盘，可以调大事务日志阈值，固态硬盘默认吞吐能力是机械硬盘的 4 倍
* 海量日志场景，也可以调整刷新 flush 的机制，改为异步；也可延长刷新 flush 时间，时间延长可能会有数据丢失，日志场景是可以容许数据丢失的

## 14 segment分段文件设置

* 海量数据写入场景，调大最小分段存储大小，可避免频繁 IO，大大减轻系统 IO 操作
* 固态硬盘条件，可以调大分段合并线程数，提升并发合并分段能力

## 15 集群慢日志

* 当集群写入或查询耗时久时可以打开集群慢日志查看具体请求