

Elasticsearch 613.28.200 性能调优指导

文档版本
发布日期

02
2018-12-06



版权所有 © 华为技术有限公司 2018。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI 和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： <http://e.huawei.com>

目 录

1 禁用 swapping.....	1
1.1 概述.....	1
1.2 禁用所有的 swap 文件.....	1
1.2.1 概述.....	1
1.2.2 临时禁用 swap.....	1
1.2.3 永久禁用 swap.....	1
1.3 配置 swappiness.....	2
2 高性能写入(Indexing)调优.....	3
2.1 使用场景.....	3
2.2 如何调优.....	3
2.2.1 参数列表.....	3
2.2.2 参数修改.....	3
2.2.2.1 修改索引刷新时间及副本数.....	3
2.2.2.2 禁用 Doc Values.....	4
2.2.2.3 修改事务日志 translog 参数.....	4
2.2.2.4 禁用_source 字段.....	5
2.3 注意事项.....	5
3 快速数据迁移调优.....	6
3.1 使用场景.....	6
3.2 如何调优.....	6
3.2.1 参数列表.....	6
3.2.2 参数修改.....	6
3.2.2.1 根据 IP 或实例下线数据节点.....	6
3.2.2.2 控制磁盘高低水位.....	7
3.3 注意事项.....	7
4 快速副本修改.....	8
4.1 使用场景.....	8
4.2 如何调优.....	8
4.2.1 参数列表.....	8
4.2.2 参数修改.....	8

4.2.2.1 调整索引恢复参数	8
4.3 注意事项	9
5 快速重启.....	10
5.1 使用场景	10
5.2 如何调优	10
5.2.1 参数列表	10
5.2.2 详细步骤	10
5.3 注意事项	11
6 快速平衡.....	12
6.1 使用场景	12
6.2 如何调优	12
6.2.1 参数列表	12
6.2.2 参数修改	12
6.2.2.1 调整集群参数	12
6.3 注意事项	13

1 禁用 swapping

1.1 概述

大多数操作系统都尽可能多地为文件系统缓存使用内存，并切换出未使用的应用程序内存。这可能导致部分 JVM 堆被交换到磁盘上。

对于性能和节点的稳定性来说，这种交换是非常糟糕的，应该不惜一切代价避免。它可能导致垃圾收集持续几分钟而不是几毫秒，这可能导致节点响应缓慢，甚至脱离集群。

可以通过三种方式来禁用 swapping：禁用所有的 swap 文件、配置 swappiness 禁用 swapping、启用 bootstrap.memory_lock 禁用 swapping。

1.2 禁用所有的 swap 文件

1.2.1 概述

通常 Elasticsearch 是运行在一个独立的机器上，且内存使用是通过 JVM 来控制的。这可能不需要开启 swap。

禁用所有的 swap 包括临时禁用和永久禁用 swap。

1.2.2 临时禁用 swap

在 Linux 操作系统，可以 **root** 用户通过以下命令来临时禁用 swap：

```
sudo swapoff -a
```

1.2.3 永久禁用 swap

要永久的停用 swap，可以通过编辑/etc/fstab 文件，注释文件中所有包含 swap 单词的行。例如：

```
#/dev/mapper/vg_r160-lv_swap swap swap nosuid 0 0
```

1.3 配置 swappiness

可以在 Linux/Unix 系统中设置“vm.swappiness”的值为“1”，这样，正常情况下不会出现内核交换，但在紧急情况下仍允许整个系统交换。具体操作步骤如下：

步骤 1 执行以下命令，查询当前系统 vm.swappiness 的值：

```
sysctl -p |grep vm.swappiness
```

步骤 2 执行以下命令打开“sysctl.conf”文件：

```
vim /etc/sysctl.conf
```

步骤 3 按“Insert”键开始编辑，修改以下参数：

```
vm.swappiness=1
```

步骤 4 按“Esc”退出编辑，输入:wq 保存退出。

----结束

2 高性能写入(Indexing)调优

2.1 使用场景

适用于历史数据导入的场景。例如，用户的历史数据存放于 HBase 中，想要将 HBase 中的数据索引到 Elasticsearch 里。

2.2 如何调优

2.2.1 参数列表

参数	默认值	建议值
refresh_interval	1s	180s+
number_of_replicas	1	0
doc_value	true	false
translog.flush_threshold_size	512mb	1G+
translog.durability	request	async
translog.sync_interval	5s	180s+
_source	true	false

2.2.2 参数修改

2.2.2.1 修改索引刷新时间及副本数

默认“index.refresh_interval”为“1s”，即每秒都会强制生成 1 个新的 segments 文件，增大索引刷新时间，可以生成更大的 segments 文件，有效降低 IO 并减少 segments merge 的压力，该配置项可以建索引时指定（或者配置到 template 里去）。

如果只是单纯导入数据，不需要做实时查询，可以把 `refresh` 禁用（即设置 `index.refresh_interval` 为 -1），并设置 “`index.number_of_replicas`” 为 “0”，当然这样设置会有数据丢失风险。等到数据完成导入后，再把参数设置为合适的值。代码如下：

```
curl -XPUT "http://194.7.7.1:24100/myindex/_settings" -H 'Content-Type: application/json' -d'
{
  "number_of_replicas": 0,
  "refresh_interval": "180s"
}'
```

2.2.2.2 禁用 Doc Values

默认情况下，支持 `doc values` 的所有字段都是开启的。因为 `Doc Values` 默认启用，可以选择对数据集里面的大多数字段进行聚合和排序操作。但是如果确定不需要在字段上进行排序和聚合，或从脚本中访问字段值，则可以禁用 `doc values` 来节省磁盘空间。

要禁用 `Doc Values`，在字段的映射（`mapping`）设置 “`doc_values`” 为 “`false`” 即可。例如，这里我们创建了一个新的索引，字段 “`session_id`” 禁用了 `Doc Values`：

```
curl -XPUT "http://194.7.7.1:24100/myindex" -H 'Content-Type: application/json' -d'
{
  "mappings": {
    "my_type": {
      "properties": {
        "session id": {
          "type": "keyword",
          "doc_values": false
        }
      }
    }
  }
}'
```

2.2.2.3 修改事务日志 translog 参数

默认设置下，`translog` 的持久化策略是每个请求都 `flush`（`durability` 参数值为 `request`），这样能保证写操作的可靠性，但是对性能会有很严重的影响，实际测试发现如果使用默认设置进行导数据磁盘 IO 会持续占满。如果系统可以接受一定几率的数据丢失（或有手段补录丢失数据），可以通过调整 `translog` 持久化策略为周期性和一定大小的时候 `flush`，能大大提升导入性能。该配置项可以建索引时指定（或者配置到 `template` 里去）。代码如下：

```
curl -XPUT "http://194.7.7.1:24100/myindex/_settings" -H 'Content-Type: application/json' -d'
{
  "index": {
    "translog": {
      "flush threshold size": "1GB",
      "sync_interval": "180s",
      "durability": "async"
    }
  }
}'
```


2.2.2.4 禁用_source 字段

“_source” 字段包含在索引时传递的原始 JSON 文档正文。该 “_source” 字段本身不被索引（因此是不可搜索的），但它被存储，以便在执行获取请求时可以返回，例如 **get** 或 **search**。

虽然很方便，但是 “_source” 字段确实在索引中有不小的存储开销。因此，可以使用如下方式禁用：

```
curl -XPUT 'http://194.7.7.1:24100/tweets?pretty' -H 'Content-Type: application/json' -d'
{
  "mappings": {
    "tweet": {
      " _source": {
        "enabled": false
      }
    }
  }
}
```



说明

在禁用_source 字段之前请注意：如果_source 字段不可用，则不支持以下功能：

- update, update_by_query, reindex APIs.
- 高亮
- 将索引从一个 Elasticsearch 索引 reindex (重索引) 到另一个索引的能力，以便更改映射或分析，或将索引升级到新的主要版本。
- 通过查看索引时使用的原始文档来调试查询或聚合的能力。
- 潜在的未来可能会自动修复索引损坏的能力。

2.3 注意事项

参数使用完毕后，修改回默认值。“doc_value”、“_source” 不支持修改。

3 快速数据迁移调优

3.1 使用场景

扩容、迁移

3.2 如何调优

3.2.1 参数列表

参数	默认值	建议值
cluster.routing.allocation.exclude._ip	无	根据 ip 迁移
cluster.routing.allocation.exclude._name	无	根据实例迁移
cluster.routing.allocation.disk.watermark.low	85%	根据实际情况调整
cluster.routing.allocation.disk.watermark.high	90%	根据实际情况调整

3.2.2 参数修改

3.2.2.1 根据 IP 或实例下线数据节点

执行如下命令，使 Elasticsearch 集群原来的数据节点下线（退服，但暂未删除实例），数据自动从旧节点往其他节点上迁移。需要使用待下线节点的业务 IP 或实例名称，支持使用*通配符。

```
curl -XPUT "http://194.7.7.1:24100/_cluster/settings" -H 'Content-Type: application/json' -d'
{
  "transient": {
```

```
"cluster": {
  "routing": {
    "allocation.exclude": {
      "_ip": "194.7.7.1,194.7.7.2,194.7.7.3",
      "_name": "EsNode2"
    }
  }
}
```

3.2.2.2 控制磁盘高低水位

- **cluster.routing.allocation.disk.watermark.low**: 控制磁盘使用的低水位。默认为 85%，意味着如果节点磁盘使用超过 85%，则 Elasticsearch 不允许再分配新的分片。当配置具体的大小如 100MB 时，表示如果磁盘空间小于 100MB 不允许分配分片。
- **cluster.routing.allocation.disk.watermark.high**: 控制磁盘使用的高水位。默认为 90%，意味着如果磁盘空间使用高于 90% 时，Elasticsearch 将尝试分配分片到其他节点。
- **cluster.info.update.interval**: Elasticsearch 应该以什么频率检查集群中每个节点的磁盘使用情况。默认为 30 s。

可以根据实际情况调整：例如

```
curl -XPUT "http://194.7.7.1:24100/ cluster/settings" -H 'Content-Type:
application/json' -d'
{
  "transient": {
    "cluster.routing.allocation.disk.watermark.low": "80%",
    "cluster.routing.allocation.disk.watermark.high": "50gb",
    "cluster.info.update.interval": "1m"
  }
}'
```

3.3 注意事项

参数使用完毕后，修改回默认值。

4 快速副本修改

4.1 使用场景

副本个数修改，例如单改双改三。

4.2 如何调优

4.2.1 参数列表

参数	默认值	建议值
<code>indices.recovery.max_bytes_per_sec</code>	40mb	1G+
<code>cluster.routing.allocation.node_initial_primaries_recoveries</code>	4	实例个数+

4.2.2 参数修改

4.2.2.1 调整索引恢复参数

- **`indices.recovery.max_bytes_per_sec`**: 索引恢复每秒最大字节数，默认值为 40MB，调整此参数可以使索引恢复的网络带宽增加。
- **`cluster.routing.allocation.node_initial_primaries_recoveries`**: 在 recovery 时，同一个 node 上允许同时恢复的主分片个数，默认是 4 个。这个设置是为了防止同时进行的 recovery 进程太多影响节点负载，因为大多数情况下用的是 local gateway，速度相当快，所以可以同时执行多个 recovery 进程而不会造成太多的负荷。参考代码如下：

```
curl -XPUT "http://194.7.7.1:24100/_cluster/settings" -H 'Content-Type: application/json' -d'
{
  "transient": {
```

```
"cluster.routing": {  
  "allocation.node_initial_primaries_recoveries": 10  
},  
"indices.recovery.max_bytes_per_sec": "1GB"  
}'
```

4.3 注意事项

参数使用完毕后，修改回默认值。

5 快速重启

5.1 使用场景

如果 Elasticsearch 集群已经导入大量数据，在需要重启时，由于节点启动先后顺序不一致等原因，直接重启流程可能带来大量的主副本之间的数据复制，需要消耗大量资源，索引恢复需要很长时间，因此如果集群有重启需求时，建议先调整相关参数，并在重启后恢复。

5.2 如何调优

5.2.1 参数列表

无

5.2.2 详细步骤

步骤 1 暂停数据写入程序。

步骤 2 **root** 用户登录 Elasticsearch 任意数据节点执行以下命令关闭集群 **shard allocation**。

```
curl -XPUT "http://194.7.7.1:24100/_cluster/settings" -H 'Content-Type: application/json' -d'
{
  "transient": {
    "cluster": {
      "routing": {
        "allocation.enable": "none"
      }
    }
  }
}'
```

步骤 3 手动触发 flush。

curl -XPOST "http://194.7.7.1:24100/_flush/synced"

步骤 4 在 Manager 上重启 Elasticsearch 服务或者重启 Elasticsearch 任意节点。

步骤 5 重新开启集群 **shard allocation**。

```
curl -XPUT "http://194.7.7.1:24100/_cluster/settings" -H 'Content-Type: application/json' -d'
{
  "transient": {
    "cluster": {
      "routing": {
        "allocation.enable": "all"
      }
    }
  }
}
```

步骤 6 等待 recovery 完成，使用如下命令查看观察到集群 “health” 的 “status” 变成 “green”。

```
curl -XGET "http://194.7.7.1:24100/_cluster/health?pretty"
----结束
```

5.3 注意事项

无。

6 快速平衡

6.1 使用场景

扩容

6.2 如何调优

6.2.1 参数列表

参数	默认值	建议值
<code>cluster.routing.allocation.cluster_concurrent_rebalance</code>	2	实例个数+
<code>indices.recovery.max_bytes_per_sec</code>	40mb	1G+

6.2.2 参数修改

6.2.2.1 调整集群参数

- **`indices.recovery.max_bytes_per_sec`**: 索引恢复每秒最大字节数，默认值为 40MB，调整此参数可以使索引恢复的网络带宽增加。
- **`cluster.routing.allocation.cluster_concurrent_rebalance`**: 控制集群范围允许多少个并发碎片重新平衡。默认值为 2。请注意，此参数仅控制由于群集中的不平衡而导致的并发碎片重定位的数量。

```
curl -XPUT "http://194.7.7.1:24100/_cluster/settings" -H 'Content-Type: application/json' -d'
{
  "transient": {
    "cluster.routing": {
      "allocation.cluster_concurrent_rebalance": 10
    }
  },
}
```



```
"indices.recovery.max_bytes_per_sec": "1GB"  
}'
```

6.3 注意事项

参数使用完毕后，修改回默认值。