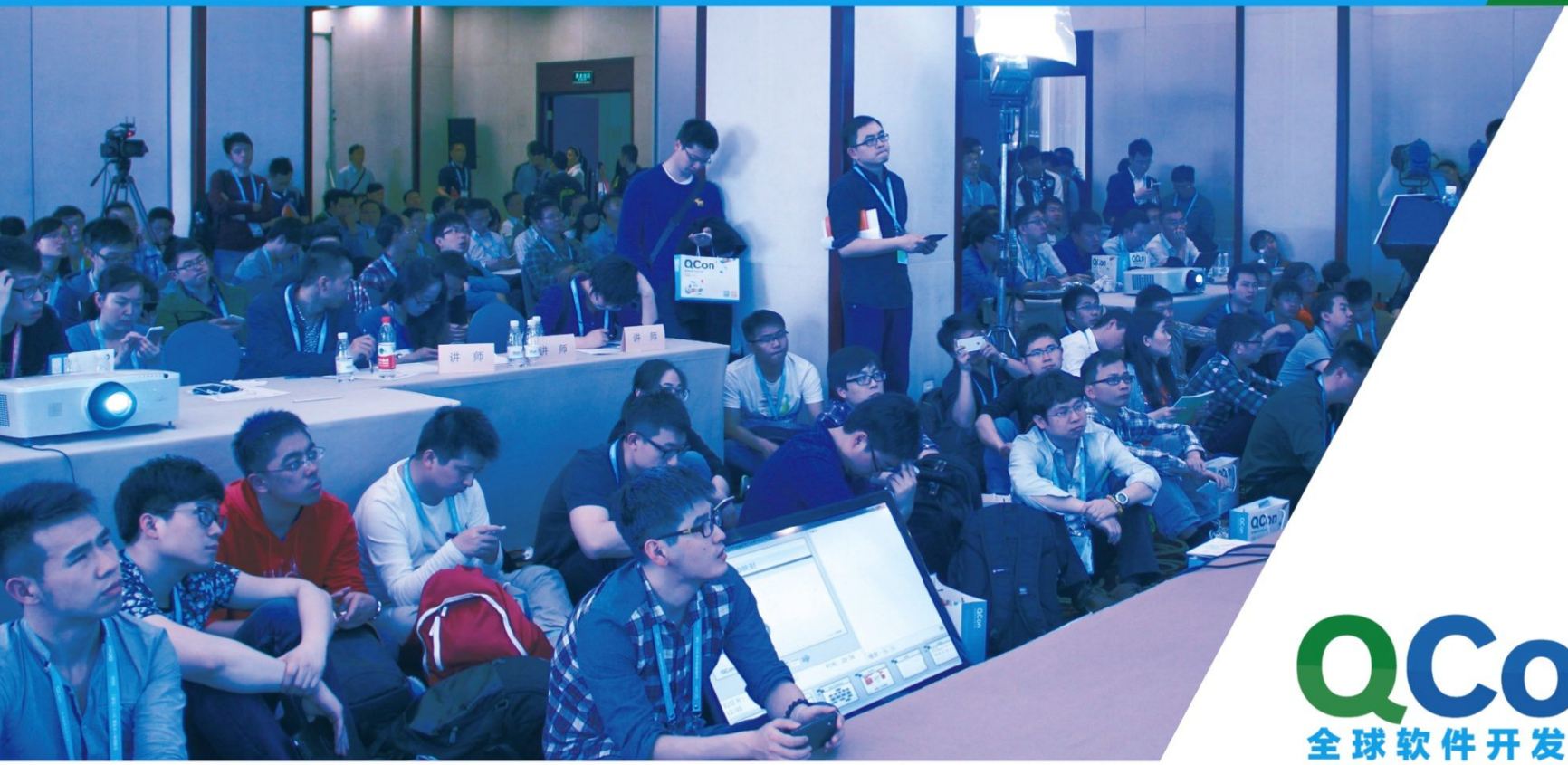
QCon全球软件开发大会

International Software Development Conference



QCon
全球软件开发大会

Elasticsearch集群中JVM问题的应对之策

我是谁

个推 首席搜索架构师 卞泽鑫.

主要职责：

- **前沿技术研究**
- **平台架构设计与开发**

演讲提纲

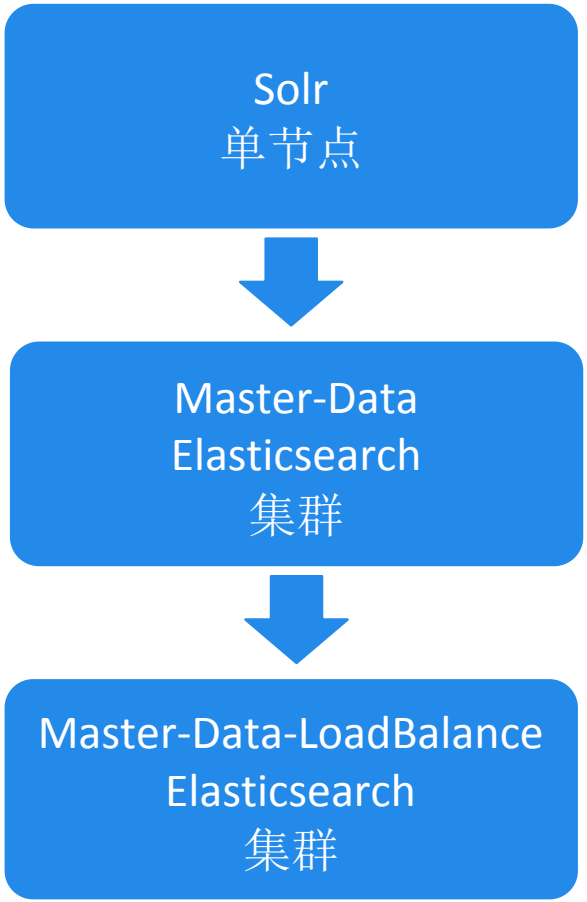
- 个推基于Elasticsearch的搜索引擎架构
- JVM在个推Elasticsearch集群的问题分析
- JVM在个推Elasticsearch集群中的调优

大家都在用个推

在第三方推送市场，个推占有率90%

- SDK接入用户超过50亿(海外5亿)
- 接入应用超过35万
- 开发者人数超过20万
- 独立覆盖8亿手机终端(海外1亿)
- 日均活跃用户近6亿
- 同时在线超过3亿

个推搜索引擎架构


Solr
单节点
Master-Data
Elasticsearch
集群
Master-Data-LoadBalance
Elasticsearch
集群

个推Elasticsearch演变

- 第一个Elasticsearch使用版本：0.20.6
- Index-Source分离架构使用的Elasticsearch版本：0.90.10
- 第一个Elasticsearch1.0以上的正式版本：1.2.2
- 目前使用的Elasticsearch版本：1.5.2

个推Elasticsearch踩过的坑

- 建立索引(index)慢
- Elasticsearch节点脱离集群
- Elasticsearch的get阻塞
- Elasticsearch备份无法恢复

个推Elasticsearch集群JVM分析工具

- gc日志输出
  - jconsole
  - jvisualvm
  - jstack
  - Eclipse内存分析器(eclipse memory analyzer)

个推Elasticsearch集群JVM问题分析

- [es-date-1224] [gc][young][3402090][244044] duration [887ms],collections [1]/[1.5s], total [887ms]/[3.3h], memory [4.5gb]->[4gb]/[6.9gb],all_pools {[young] [499.4mb]->[782.8kb]/[532.5mb]}{[survivor][32.7mb]->[30.2mb]/[66.5mb]}{[old] [3.9gb]->[3.9gb]/[6.3gb]}
- 上面这个例子的情况无须紧张，只是young gc，并且只用了887ms，对于Elasticsearch而言，没有啥影响。唯一需要留心的是，如果在日志中出现连续的和长时间的young gc，则需要引起警觉，可能是你的Heap内存分配不够。

个推Elasticsearch集群JVM问题分析

- [es-data-1224] [gc][old][76581][22] duration [3.1m], collections[2]/[3.1m], total [3.1m]/[3.1m], memory [3gb]->[1.2gb]/[3.4gb], all_pools{[young] [251mb]->[74.9mb]/[266.2mb]}{[survivor][25.8mb]->[0b]/[33.2mb]}{[old] [2.8gb]->[1.1gb]/[3.1gb]}
- 如果这种JVM出现，则你的节点一定被踢出了集群。old gc是比较耗时，上面这个例子用了3.1分钟，一定是出了啥大事，要不是然“世界”不会停转这么久的，呵呵！

个推Elasticsearch集群JVM问题分析

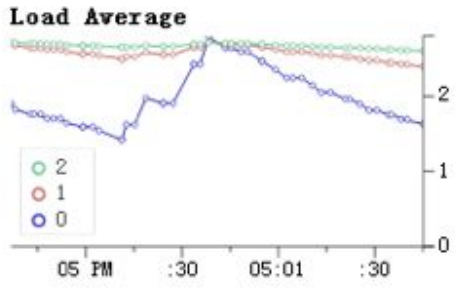
节点频繁GC，节点丢失

调大参数

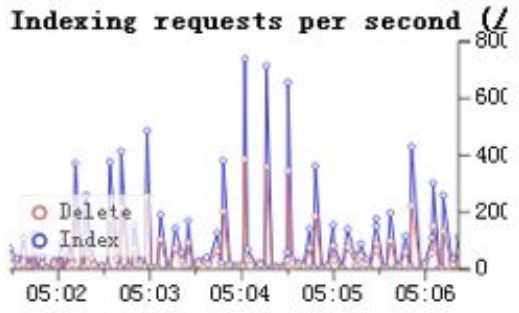
- discovery.zen.fd.ping_interval
- discovery.zen.fd.ping_timeout
- discovery.zen.fd.ping_retries

个推Elasticsearch集群JVM问题分析

客户端建立索引很慢，但是从监控上看到ES的load并不高



2: 2.6
1: 2.39
0: 1.63



Delete: 1055377851
Index: 1286760524

个推Elasticsearch集群JVM问题分析

客户端JStack分析，客户端实际上是阻塞的

```
[redacted] 000061d22000 nid=0x5c89 waiting on condition [0x0000000055c3d000]
java.lang.Thread.State: WAITING (parking)
 at sun.misc.Unsafe.park(Native Method)
 - parking to wait for  <0x00000007003248f0> (a org.elasticsearch.common.util.concurrent.BaseFuture$Sync)
 at java.util.concurrent.locks.LockSupport.park(LockSupport.java:158)
 at java.util.concurrent.locks.AbstractQueuedSynchronizer.parkAndCheckInterrupt(AbstractQueuedSynchronizer.java:811)
 at java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireSharedInterruptibly(AbstractQueuedSynchronizer.java:969)
 at java.util.concurrent.locks.AbstractQueuedSynchronizer.acquireSharedInterruptibly(AbstractQueuedSynchronizer.java:1281)
 at org.elasticsearch.common.util.concurrent.BaseFuture$Sync.get(BaseFuture.java:274)
 at org.elasticsearch.common.util.concurrent.BaseFuture.get(BaseFuture.java:113)
 at org.elasticsearch.action.support.AdapterActionFuture.actionGet(AdapterActionFuture.java:45)
 [redacted]
 at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886)
 at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908)
 at java.lang.Thread.run(Thread.java:662)

Locked ownable synchronizers:
 - <0x000000074c2bb770> (a java.util.concurrent.locks.ReentrantLock$NonfairSync)
```

个推Elasticsearch集群JVM问题分析

```
"elasticsearch[                                    index][T#2]" - Thread t@199
   java.lang.Thread.State: RUNNABLE
    at org.elasticsearch.index.store.Store$StoreDirectory.deleteFile(Store.java:398)
    - locked <16e6f4cb> (a java.lang.Object)
    at org.apache.lucene.index.IndexFileDeleter.deleteFile(IndexFileDeleter.java:584)
    at org.apache.lucene.index.IndexFileDeleter.deleteNewFiles(IndexFileDeleter.java:572)
    at org.apache.lucene.index.IndexWriter.deleteNewFiles(IndexWriter.java:4622)
    - locked <59883a5a> (a org.apache.lucene.index.IndexWriter)
    at org.apache.lucene.index.DocumentsWriter$DeleteNewFilesEvent.process(DocumentsWriter.java:727)
```

```
doFlush(DocumentsWriterPerThread) : boolean - org.apache.lucene.index.DocumentsWriter
  flushAllThreads(IndexWriter) : boolean - org.apache.lucene.index.DocumentsWriter
  postUpdate(DocumentsWriterPerThread, boolean) : boolean - org.apache.lucene.index.DocumentsWriter (2 matches)
  preUpdate() : boolean - org.apache.lucene.index.DocumentsWriter
```

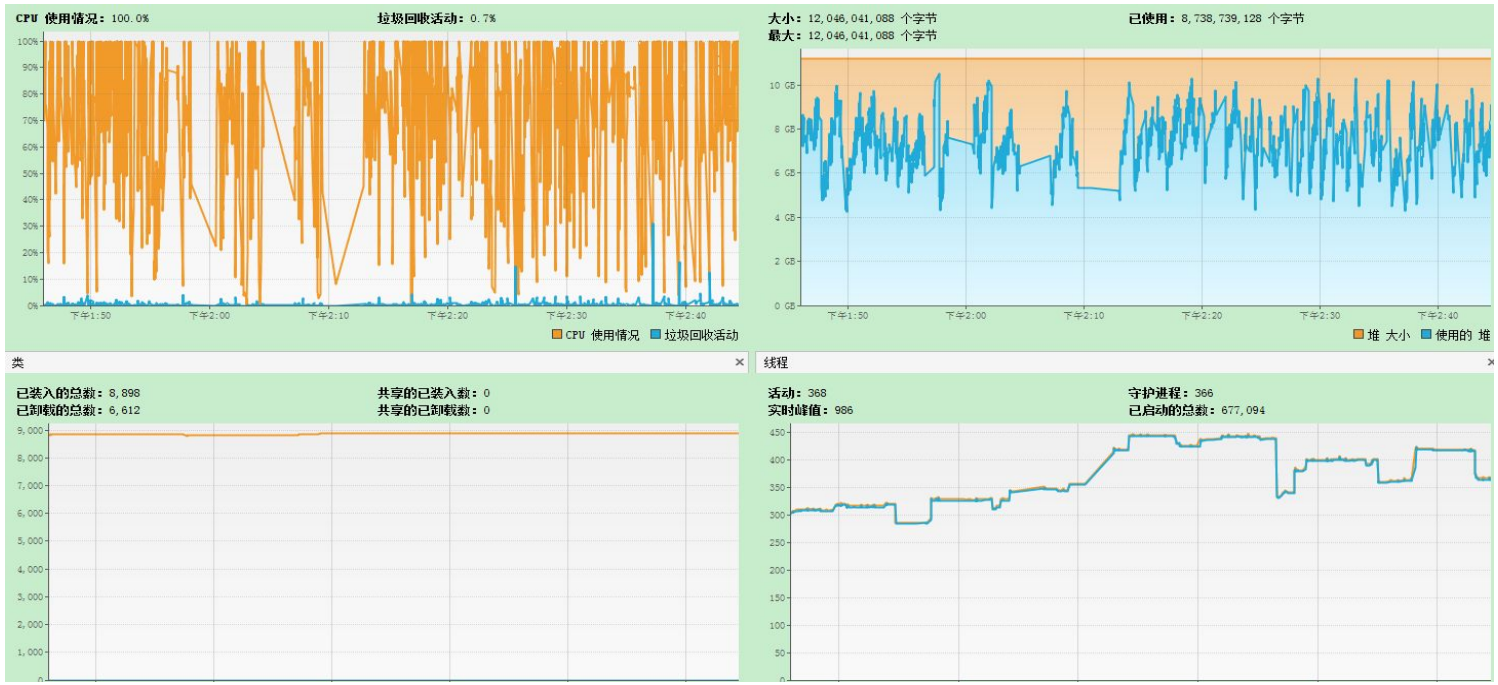
从ES的jstack信息的分析结果来看，应该是flush操作导致ES建索引慢

index.translog.flush_threshold_ops ＝ 5000

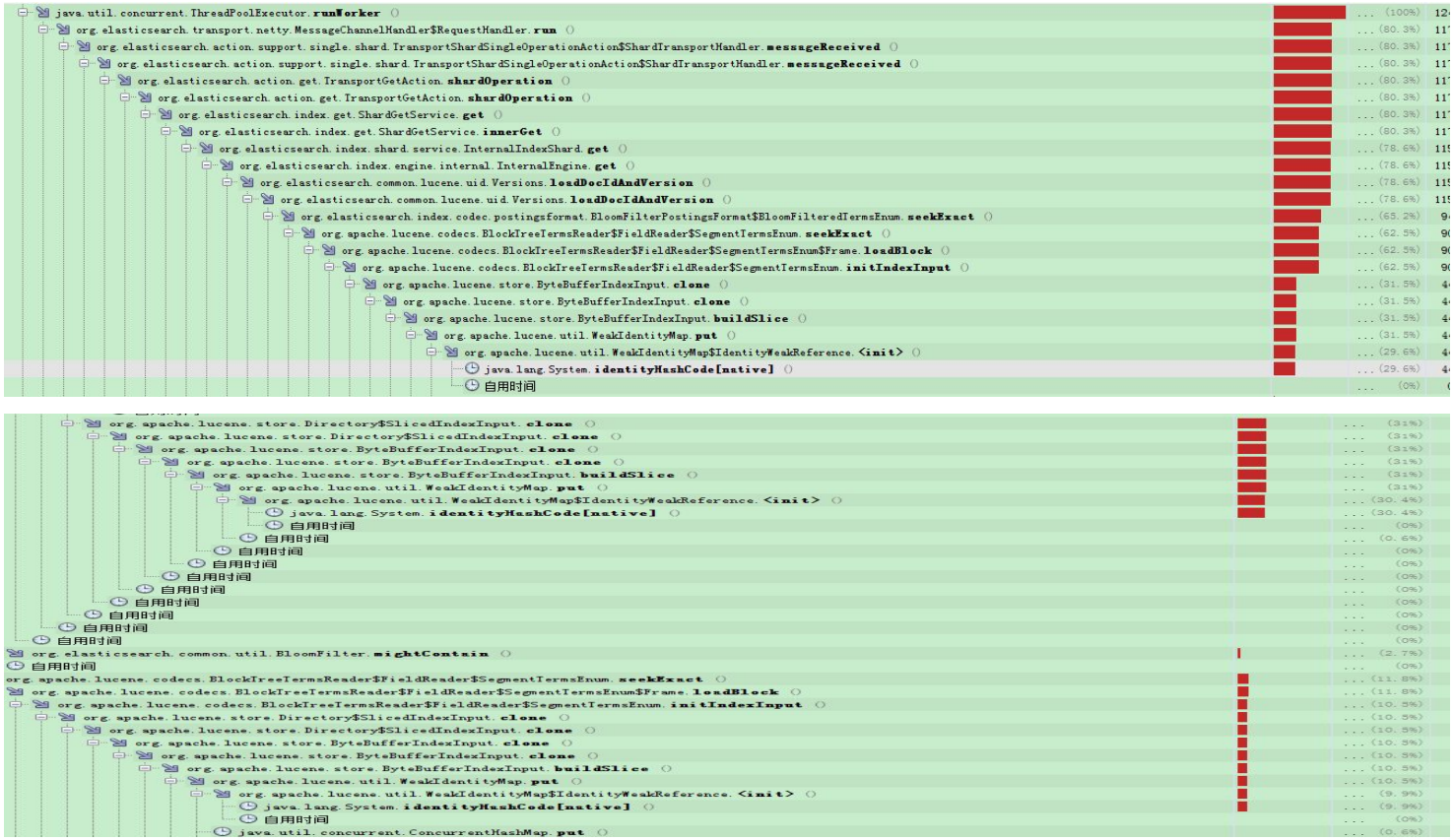
ndex.translog.flush_threshold_size ＝ 200mb

个推Elasticsearch集群JVM问题分析

表现： Elasticsearch集群get操作缓慢



个推Elasticsearch集群JVM问题分析



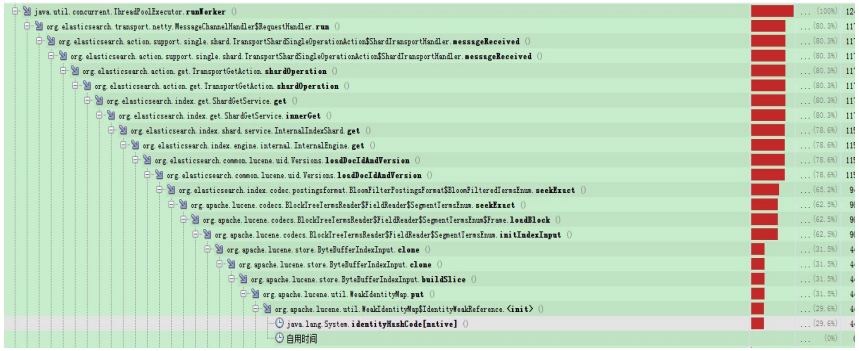
个推Elasticsearch集群JVM问题分析

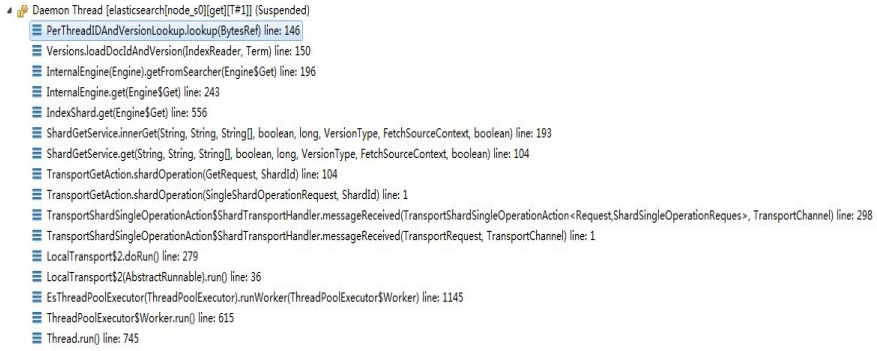
通过工具的分析结果，我们可以看到节点服务器的cpu基本全部跑满，主要是因为每次es的get请求的条件都不一样，在ES集群的queryResult caches中获取不到结果，所以每次都得调用identityHashCode来重新计算hash值，然后放入WeakIdentityMap中，这些操作占用了绝大部分cpu时间，导致服务器的cpu占用率非常高。

个推Elasticsearch集群JVM问题分析

- 解决Elasticsearch集群get操作慢的问题
  - Elasticsearch-1.2.2升级到Elasticsearch-1.5.2

Elasticsearch-1.2.2与Elasticsearch-1.5.2get操作对比





个推Elasticsearch集群JVM优化

- 使用JDK1.7+
- ES_HEAP_SIZE
- ES_HEAP_NEWSIZE
- 禁用jdk7默认Garbage-First(G1)

个推Elasticsearch集群JVM优化

- CMS garbage collector
  - 增大： CMSInitatingOccupancyFraction
  - 降小： cache expire time
  - 调小： index.merge.policy.segments_per_tier

No Duang, No Push!

个推真诚期待与您携手！

微信关注

官网注册

个推官网：www.GeTui.com

客服电话：4006-808-606

商务咨询：support@getui.com

客服 QQ：2117094763

THANKS