

## Presto 超大内存泄漏问题排查

最近有一个线上 Presto 集群的 Coordinator 节点内存一直处于90%以上，如下图所示：



如果想及时了解Spark、Hadoop或者HBase相关的文章，欢迎关注微信公众号：过往记忆大数据

日志里面可以看到几乎每分钟都在 Full GC，但是内存仍然占用很高，所以第一个想法就是存在内存泄漏。但是线上的集群内存是180GB，所以分析这些内存就不能用常规的方法了。本文主要是简单介绍对于 Java 超大内存如何进行内存泄漏排查。

### Dump 内存

为了分析内存的使用情况，我们不得不把 Coordinator 节点的内存 dump 下来，因为内存很大，所以我们开启了压缩，具体命令如下：

```
jmap -dump:live,gz=1,file=iteblog.hprof.gz <pid>
```

注意：因为内存很大，上面命令在执行的过程中会导致集群不可用，所以建议在集群使用量很小的情况下进行。

### 使用 ParseHeapDump.sh 脚本分析内存使用情况

在我们的场景下，dump 下来的内存压缩后的文件都有36GB，解压后更是达到了160多GB，所以我们无法把它下载到本地，然后使用图形化的 Eclipse Memory Analyzer 进行分析。不过 Eclipse Memory Analyzer 给我们提供了命令行分析内存的方法，也就是通过 ParseHeapDump.sh 来分析。可以到[这里](#)下载对应版本的 Memory Analyzer。

下载好之后，我们就可以使用 ParseHeapDump.sh 来分析内存，具体如下：

```
./ParseHeapDump.sh /home/iteblog/iteblog.hprof.gz -discard_ratio=80 -discard_pattern="char  
W[W]|byteW[W]|javaW.langW.String|javaW.langW.Long|java.lang.ObjectW[W]|javaW.lang  
W.Integer" org.eclipse.mat.api:suspects org.eclipse.mat.api:overview org.eclipse.mat.api:top_c  
omponents
```

关于 Memory Analyzer 的使用可以参见[这里](#)。

执行完上面命令之后，可以得到三个文件：

```
[root@11.11.11.11 iteblog]# ll
总用量 12575904
-rw-r--r-- 1 root root 146914 12月 29 10:47 iteblog_Leak_Suspects.zip
-rw-r--r-- 1 root root 85080 12月 29 10:47 iteblog_System_Overview.zip
-rw-r--r-- 1 root root 764025 12月 29 11:23 iteblog_Top_Components.zip
```

我们可以把上面三个文件进行解压，之后，我们就可以到 iteblog\_Leak\_Suspects 解压包里面得到内存泄漏的相关信息。

**本博客文章除特别声明，全部都是原创！**  
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。  
本文链接: **【】** ( )