# springboot启动

top - 17:23:42 up 64 days, 7:20, 3 users, load average: 7.87, 8.48, 8.58

Threads: 15 total, 0 running, 15 sleeping, 0 stopped, 0 zombie

%Cpu(s): 91.7 us, 8.3 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

KiB Mem : 1882048 total, 120492 free, 1309588 used, 451968 buff/cache

KiB Swap: 0 total, 0 free, 0 used. 404672 avail Mem

4843 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.00 java

4844 root 20 0 2064792 116508 13324 S 0.0 6.2 0:03.20 java

4845 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.27 VM Thread

4846 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.00 Reference Handl

4847 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.00 Finalizer

4848 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.00 Signal Dispatch

4849 root 20 0 2064792 116508 13324 S 0.0 6.2 0:03.61 C2 CompilerThre

4850 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.89 C1 CompilerThre

4851 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.00 Service Thread

4852 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.02 VM Periodic Tas

4892 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.01 Catalina-utilit

4893 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.00 Catalina-utilit

4894 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.00 container-0

4906 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.00 http-nio-10001-

4907 root 20 0 2064792 116508 13324 S 0.0 6.2 0:00.00 http-nio-10001-

一个javabean实例估计在16bytes-64bytes

# Jmap分析

## 内存对象个数及内存分析

jmap -histo 714776|head -n 20

### 时段1

[meisapp@[开发]iz2ze8jj0t8w1dknxv614ez ~]$ jmap -histo 714776|head -n 20

num #instances #bytes class name

----------------------------------------------

1: 220267 40452440 [B

2: 261411 35265264 [C

3: 455688 14582016 java.util.concurrent.ConcurrentHashMap$Node

4: 381183 12197856 java.util.Hashtable$Entry

5: 140167 12068808 [Ljava.lang.Object;

6: 34395 10636928 [I

7: 382709 9244032 [Ljava.lang.String;

8: 135504 6504192 java.util.HashMap

9: 258310 6199440 java.lang.String

10: 157710 5046720 org.apache.hadoop.hbase.KeyValue

11: 77940 4364640 java.util.stream.ReferencePipeline$Head

12: 46579 3950176 [Ljava.util.HashMap$Node;

13: 69893 3914008 java.util.LinkedHashMap

14: 96143 3845720 java.util.LinkedHashMap$Entry

15: 59264 3792896 java.util.stream.ReferencePipeline$2

16: 2346 3253552 [Ljava.util.Hashtable$Entry;

17: 130235 3125640 java.util.ArrayList

### 时段2

### 时段3

### 时段4

## 内存活着的对象个数及内存分析

jmap -histo:live 714776|head -n 20

### 时段1

[meisapp@[开发]iz2ze8jj0t8w1dknxv614ez ~/services/meisoo-bigdata-open]$ jmap -histo:live 714776|head -n 20

num #instances #bytes class name

----------------------------------------------

1: 128070 10743568 [C

2: 19420 8019160 [I

3: 20374 3386272 [Ljava.lang.Object;

4: 98798 3161536 java.util.concurrent.ConcurrentHashMap$Node

5: 127676 3064224 java.lang.String

6: 16066 1780088 java.lang.Class

7: 17886 1573968 java.lang.reflect.Method

8: 8066 1504912 [B

9: 40154 1284928 java.util.HashMap$Node

10: 10708 968032 [Ljava.util.HashMap$Node;

11: 27737 887584 java.util.Hashtable$Entry

12: 21216 848640 java.util.LinkedHashMap$Entry

13: 1248 818688 org.apache.hbase.thirdparty.io.netty.util.internal.shaded.org.jctools.queues.MpscArrayQueue

14: 29436 752520 [Ljava.lang.String;

15: 558 727568 [Ljava.util.concurrent.ConcurrentHashMap$Node;

16: 11269 631064 java.util.LinkedHashMap

17: 36850 589600 java.lang.Object

### 时段2

[meisapp@[开发]iz2ze8jj0t8w1dknxv614ez ~/services/meisoo-bigdata-open]$ jmap -histo:live 714776|head -n 20

num #instances #bytes class name

----------------------------------------------

1: 146720 14751536 [C

2: 47304 13012896 [B

3: 335761 10744352 java.util.concurrent.ConcurrentHashMap$Node

4: 265919 8509408 java.util.Hashtable$Entry

5: 266334 6438072 [Ljava.lang.String;

6: 44809 4208424 [Ljava.lang.Object;

7: 146326 3511824 java.lang.String

8: 1674 2273104 [Ljava.util.Hashtable$Entry;

9: 2499 2157728 [Ljava.util.concurrent.ConcurrentHashMap$Node;

10: 22820 1825600 org.apache.hbase.thirdparty.com.google.common.cache.LocalCache$Segment

11: 16066 1780088 java.lang.Class

12: 17883 1573704 java.lang.reflect.Method

13: 10122 1436096 [I

14: 19801 1425672 java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask

15: 40246 1287872 java.util.HashMap$Node

16: 12079 1077808 [Ljava.util.HashMap$Node;

17: 33443 1070176 org.apache.hadoop.hbase.KeyValue

### 时段3

### 时段4

## jmap -dump:format=b,file=/root/dump.hprof 1716

### 时段1

### 时段2

### 时段3

### 时段4

# Jstat分析,

## jstat -gc -t 714776信息(单位应该是k)

jstat -gc -t 714776 10000 30

|  |  |
| --- | --- |
| OC | Old代的容量 (字节) |
| OU | Old代目前已使用空间 (字节) |

|  |  |
| --- | --- |
| FGC | 从应用程序启动到采样时old代(全gc)gc次数 |
| FGCT | 从应用程序启动到采样时old代(全gc)gc所用时间(s) |
| GCT | 从应用程序启动到采样时gc用的总时间(s) |

### 时段1

[meisapp@[开发]iz2ze8jj0t8w1dknxv614ez ~/services/meisoo-bigdata-open]$ jstat -gc -t 714776 10000 30

Timestamp S0C S1C S0U S1U EC EU OC OU MC MU CCSC CCSU YGC YGCT FGC FGCT GCT

3416.8 3584.0 3584.0 0.0 2240.0 167424.0 83788.0 349696.0 149029.4 91480.0 86290.0 11136.0 10154.4 7938 87.033 20 4.883 91.916

3426.8 4096.0 4096.0 0.0 2912.0 166400.0 160670.3 349696.0 166077.9 91480.0 86290.0 11136.0 10154.4 7993 87.574 20 4.883 92.456

### 时段2

### 时段3

### 时段4

## jstat -class 714776 这个用处很小

### 时段1

### 时段2

### 时段3

### 时段4

# jstack 1721

## 命令

线程死了,查出活着,不准确

### 时段1

### 时段2

### 时段3

### 时段4

# 模版分析

## 命令

### 时段1

### 时段2

### 时段3

### 时段4