주제

- description : 자바 프로그램 분석 방법 기술, 힙영역 분석, 쓰레드 분석
- author : 주레피
- email : dhan@repia.com
- lastupdate : 2022-07-10

Intro

Case Study

Tools

jps

JVM 프로세스 목록을 확인할 수 있음

```
$> jps -v
32128 Jps -Dapplication.home=/PROJECT/dge.repia.com/SearchApp/jdk1.8.0_202 -
Xms8m
5287 Server
24487 Bootstrap -Djava.util.logging.config.file=/WAS_APPS/tomcat-
instance/kigam.repia.com/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djava.awt.headless=true -Xms512m -Xmx1024m -XX:NewSize=256m -
XX:MaxNewSize=256m -XX:PermSize=256m -XX:MaxPermSize=512m -
XX:+DisableExplicitGC -Djdk.tls.ephemeralDHKeySize=2048 -
Djava.protocol.handler.pkgs=org.apache.catalina.webresources -
Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Denv=product -
Denv.servername=kigam.repia.com -Dignore.endorsed.dirs= -
Dcatalina.base=/WAS_APPS/tomcat-instance/kigam.repia.com -
Dcatalina.home=/WAS_APPS/apache-tomcat-8.5.76 -
Djava.io.tmpdir=/WAS_APPS/tomcat-instance/kigam.repia.com/temp
22156 Bootstrap -Djava.util.logging.config.file=/WAS_APPS/tomcat-
instance/dge.repia.com/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djava.awt.headless=true -Xms512m -Xmx1024m -XX:NewSize=256m -
XX:MaxNewSize=256m -XX:PermSize=256m -XX:MaxPermSize=512m -
XX:+DisableExplicitGC -Djdk.tls.ephemeralDHKeySize=2048 -
Djava.protocol.handler.pkgs=org.apache.catalina.webresources -
Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Denv=product -
Denv.servername=dge.repia.com -Dignore.endorsed.dirs= -
Dcatalina.base=/WAS_APPS/tomcat-instance/dge.repia.com -
```

```
Dcatalina.home=/WAS_APPS/apache-tomcat-8.5.76 -
Djava.io.tmpdir=/WAS_APPS/tomcat-instance/dge.repia.com/temp
26732 Server
27185 jar
14770 Server
26132 jar
18037 Server
20021 jar
25242 Bootstrap -
Djava.util.logging.config.file=/usr/local/tomcat8/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djdk.tls.ephemeralDHKeySize=2048 -
Djava.protocol.handler.pkgs=org.apache.catalina.webresources -
Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -
Dignore.endorsed.dirs= -Dcatalina.base=/usr/local/tomcat8 -
Dcatalina.home=/usr/local/tomcat8 -Djava.io.tmpdir=/usr/local/tomcat8/temp
32412 jar
11933 jar
26878 Bootstrap -Djava.util.logging.config.file=/WAS_APPS/tomcat-
instance/centos79.repia.com/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djava.awt.headless=true -Xms512m -Xmx1024m -XX:NewSize=256m -
XX:MaxNewSize=256m -XX:PermSize=256m -XX:MaxPermSize=512m -
XX:+DisableExplicitGC -Djdk.tls.ephemeralDHKeySize=2048 -
Djava.protocol.handler.pkgs=org.apache.catalina.webresources -
Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Denv=product -
Denv.servername=centos79.repia.com -Dignore.endorsed.dirs= -
Dcatalina.base=/WAS_APPS/tomcat-instance/centos79.repia.com -
Dcatalina.home=/WAS_APPS/apache-tomcat-8.5.76 -
Djava.io.tmpdir=/WAS_APPS/tomcat-instance/centos79.repia.com/temp
```

jmap

현재 실행 중인 JVM 프로세스의 메모리 맵(map)을 확인할 수 있음
JVM의 힙 메모리 정보를 얻어오거나 덤프를 떠서 분석해 볼 수 있다.

```
$> jmap -heap 200021
[root@sofo dump]# /PROJECT/dge.repia.com/SearchApp/java8/bin/jmap -heap
20021
Attaching to process ID 20021, please wait...
Debugger attached successfully.
Server compiler detected.
JVM version is 25.202-b08

using thread-local object allocation.
Parallel GC with 8 thread(s)

Heap Configuration:
   MinHeapFreeRatio         = 0
   MaxHeapFreeRatio         = 100
```

```
   MaxHeapSize               = 2009071616 (1916.0MB)
   NewSize                   = 41943040 (40.0MB)
   MaxNewSize                = 669515776 (638.5MB)
   OldSize                   = 83886080 (80.0MB)
   NewRatio                  = 2
   SurvivorRatio             = 8
   MetaspaceSize             = 21807104 (20.796875MB)
   CompressedClassSpaceSize  = 1073741824 (1024.0MB)
   MaxMetaspaceSize          = 17592186044415 MB
   G1HeapRegionSize          = 0 (0.0MB)

Heap Usage:
PS Young Generation
Eden Space:
   capacity = 125829120 (120.0MB)
   used     = 41686504 (39.755348205566406MB)
   free     = 84142616 (80.2446517944336MB)
   33.12945683797201% used
From Space:
   capacity = 5242880 (5.0MB)
   used     = 5231912 (4.989540100097656MB)
   free     = 10968 (0.01045989990234375MB)
   99.79080200195312% used
To Space:
   capacity = 5242880 (5.0MB)
   used     = 0 (0.0MB)
   free     = 5242880 (5.0MB)
   0.0% used
PS Old Generation
   capacity = 83886080 (80.0MB)
   used     = 6313904 (6.0214080810546875MB)
   free     = 77572176 (73.97859191894531MB)
   7.526760101318359% used

1711 interned Strings occupying 144344 bytes.

// JVM 메모리 통계
$> jmap -F -histo 20021

// 메모리 덤프
$> jmap -F -dump:format=b,file=heap.hprof 20021

==== jhat ====
메모리 덤프 파일 분석
<code bash>
$> jhat heap.hprof -port 7000
```

Term

Tip

Troubleshooting

Ref

- [Java] 자바 메모리 덤프 분석 - jps, jmap, jhat 사용법 및 예제
- [Java] Heap 모니터링 & Heap Dump 분석하기
- Thread Dump 분석
- HOW TO CAPTURE JAVA HEAP DUMPS? – 7 OPTIONS

heapdump, tda, 주레피, 메모리덤프, visualvm, TDA

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:java:analysis&rev=1657431812

Last update: **2022/07/10 14:43**