JVM第十五天-JVM调优-arthas的使用

**Arthas简介**

arthas是啥？ 一款阿里出品的在线JVM监测工具

**为什么需要在线排查？**

在生产上我们经常会碰到一些不好排查的问题，例如线程安全问题，

用最简单的threaddump或者heapdump不好查到问题原因。

为了排查这些问题，有时我们会临时加一些日志，比如在一些关键的函数里打印出入参，

然后重新打包发布，如果打了日志还是没找到问题，继续加日志，重新打包发布。

对于上线流程复杂而且审核比较严的公司，从改代码到上线需要层层的流转，会大大影响问题排查的进度。

**下载和安装**

访问 https://github.com/alibaba/arthas

进行jar包的下载安装，这里不进行赘述。

**启动**

这里还是使用我们上次的案例程序

|  |
| --- |
| package com.peng.jvm.gc;  import java.math.BigDecimal;  import java.util.ArrayList;  import java.util.Date;  import java.util.List;  import java.util.concurrent.ScheduledThreadPoolExecutor;  import java.util.concurrent.ThreadPoolExecutor;  import java.util.concurrent.TimeUnit;  /\*\*  \* 从数据库中读取信用数据，套用模型，并把结果进行记录和传输  \*/  public class T15\_FullGC\_Problem01 {  private static class CardInfo {  BigDecimal price = new BigDecimal(0.0);  String name = "张三";  int age = 5;  Date birthdate = new Date();  public void m() {}  }  private static ScheduledThreadPoolExecutor executor = new ScheduledThreadPoolExecutor(50,  new ThreadPoolExecutor.DiscardOldestPolicy());  public static void main(String[] args) throws Exception {  executor.setMaximumPoolSize(50);  for (;;){  modelFit();  Thread.sleep(100);  }  }  private static void modelFit(){  List<CardInfo> taskList = getAllCardInfo();  taskList.forEach(info -> {  // do something  executor.scheduleWithFixedDelay(() -> {  //do sth with info  info.m();  }, 2, 3, TimeUnit.SECONDS);  });  }  private static List<CardInfo> getAllCardInfo(){  List<CardInfo> taskList = new ArrayList<>();  for (int i = 0; i < 100; i++) {  CardInfo ci = new CardInfo();  taskList.add(ci);  }  return taskList;  }  } |

先启动这个程序：

|  |
| --- |
| java -Xms20M -Xmx20M -XX:+UseParallelGC -XX:+HeapDumpOnOutOfMemoryError com.peng.jvm.gc.T15\_FullGC\_Problem01 |

接下来移动到arthas根目录，启动arthas：

|  |
| --- |
| java -jar arthas-agent.jar |

|  |
| --- |
| [root@dream01 arthas]# java -jar arthas-agent.jar  arthas-agent.jar中没有主清单属性  [root@dream01 arthas]# java -jar arthas-boot.jar  [INFO] arthas-boot version: 3.1.7  [INFO] Found existing java process, please choose one and hit RETURN.  \* [1]: 2169 T15\_FullGC\_Problem01 |

此时，可以看到，arthas检测到了所有的java运行程序，其中1号就是我们刚启动的程序。

输入1：

|  |
| --- |
| 1  [INFO] arthas home: /usr/local/arthas  [INFO] Try to attach process 2363  [INFO] Attach process 2363 success.  [INFO] arthas-client connect 127.0.0.1 3658  ,---. ,------. ,--------.,--. ,--. ,---. ,---.  / O \ | .--. ''--. .--'| '--' | / O \ ' .-'  | .-. || '--'.' | | | .--. || .-. |`. `-.  | | | || |\ \ | | | | | || | | |.-' |  `--' `--'`--' '--' `--' `--' `--'`--' `--'`-----'    wiki https://alibaba.github.io/arthas  tutorials https://alibaba.github.io/arthas/arthas-tutorials  version 3.1.7  pid 2363  time 2020-03-23 00:35:55 |

此时代表已经成功进入检测我们程序的界面了。

**帮助**

可以通过help查看有哪些命令

|  |
| --- |
| [arthas@2874]$ help  NAME DESCRIPTION  help Display Arthas Help  keymap Display all the available keymap for the specified connection.  sc Search all the classes loaded by JVM  sm Search the method of classes loaded by JVM  classloader Show classloader info  jad Decompile class  getstatic Show the static field of a class  monitor Monitor method execution statistics, e.g. total/success/failure count, average rt, fail rate, etc.  stack Display the stack trace for the specified class and method  thread Display thread info, thread stack  trace Trace the execution time of specified method invocation.  watch Display the input/output parameter, return object, and thrown exception of specified method invocation  tt Time Tunnel  jvm Display the target JVM information  ognl Execute ognl expression.  mc Memory compiler, compiles java files into bytecode and class files in memory.  redefine Redefine classes. @see Instrumentation#redefineClasses(ClassDefinition...)  dashboard Overview of target jvm's thread, memory, gc, vm, tomcat info.  dump Dump class byte array from JVM  heapdump Heap dump  options View and change various Arthas options  cls Clear the screen  reset Reset all the enhanced classes  version Display Arthas version  shutdown Shutdown Arthas server and exit the console  stop Stop/Shutdown Arthas server and exit the console. Alias for shutdown.  session Display current session information  sysprop Display, and change the system properties.  sysenv Display the system env.  vmoption Display, and update the vm diagnostic options.  logger Print logger info, and update the logger level  history Display command history  cat Concatenate and print files  pwd Return working directory name  mbean Display the mbean information  grep grep command for pipes.  profiler Async Profiler. https://github.com/jvm-profiling-tools/async-profiler |

**常用命令**

**jvm**

打印JVM内的一些基本信息

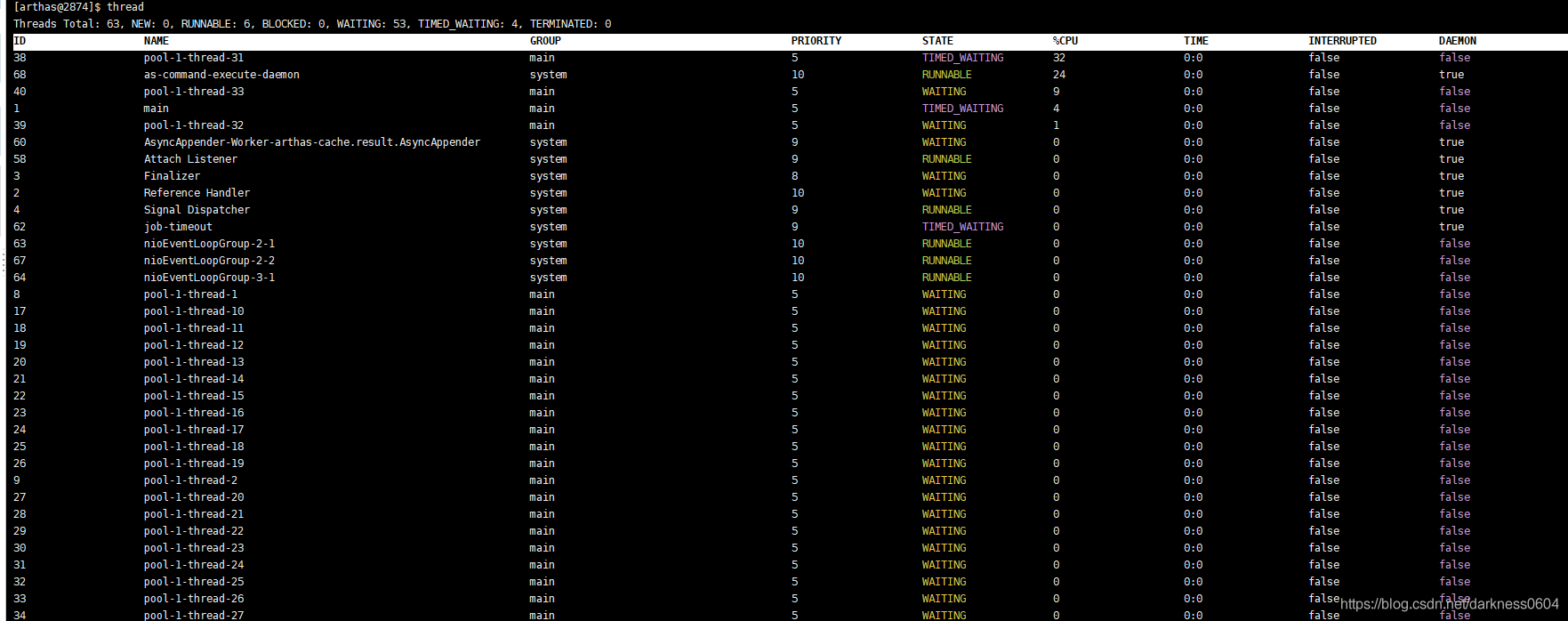
|  |
| --- |
| [arthas@2496]$ jvm  RUNTIME  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  MACHINE-NAME 2496@dream01  JVM-START-TIME 2020-03-23 00:37:11  MANAGEMENT-SPEC-VERSION 1.2  SPEC-NAME Java Virtual Machine Specification  SPEC-VENDOR Oracle Corporation  SPEC-VERSION 1.8  VM-NAME Java HotSpot(TM) 64-Bit Server VM  VM-VENDOR Oracle Corporation  VM-VERSION 25.211-b12  INPUT-ARGUMENTS -Xms20M  -Xmx20M  -XX:+UseParallelGC  -XX:+HeapDumpOnOutOfMemoryError    CLASS-PATH .  BOOT-CLASS-PATH /usr/java/jdk1.8.0\_211-amd64/jre/lib/resources.jar:/usr/java/jdk1.8.0\_211-amd64/jre/lib/rt.jar:/usr/java/jdk1.8.0\_211-amd64/jre/lib/sunrsasign.jar:/usr/java/jdk1.8.0\_211-amd64/j  re/lib/jsse.jar:/usr/java/jdk1.8.0\_211-amd64/jre/lib/jce.jar:/usr/java/jdk1.8.0\_211-amd64/jre/lib/charsets.jar:/usr/java/jdk1.8.0\_211-amd64/jre/lib/jfr.jar:/usr/java/jdk1.8.0\_21  1-amd64/jre/classes  LIBRARY-PATH /usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib    ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  CLASS-LOADING  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  LOADED-CLASS-COUNT 2288  TOTAL-LOADED-CLASS-COUNT 2293  UNLOADED-CLASS-COUNT 5  IS-VERBOSE false    ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  COMPILATION  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  NAME HotSpot 64-Bit Tiered Compilers  TOTAL-COMPILE-TIME 2130(ms)    ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  GARBAGE-COLLECTORS  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  PS Scavenge 37/117(ms)  [count/time]  PS MarkSweep 6/589(ms)  [count/time]    ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  MEMORY-MANAGERS  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  CodeCacheManager Code Cache    Metaspace Manager Metaspace  Compressed Class Space    PS Scavenge PS Eden Space  PS Survivor Space    PS MarkSweep PS Eden Space  PS Survivor Space  PS Old Gen      ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  MEMORY  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  HEAP-MEMORY-USAGE 19922944(19.00 MiB)/20971520(20.00 MiB)/19922944(19.00 MiB)/17337072(16.53 MiB)  [committed/init/max/used]  NO-HEAP-MEMORY-USAGE 19988480(19.06 MiB)/2555904(2.44 MiB)/-1(-1 B)/19155400(18.27 MiB)  [committed/init/max/used]  PENDING-FINALIZE-COUNT 0    ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  OPERATING-SYSTEM  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  OS Linux  ARCH amd64  PROCESSORS-COUNT 1  LOAD-AVERAGE 0.69  VERSION 2.6.32-431.el6.x86\_64    ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  THREAD  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  COUNT 63  DAEMON-COUNT 7  PEAK-COUNT 63  STARTED-COUNT 64  DEADLOCK-COUNT 0    ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  FILE-DESCRIPTOR  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  MAX-FILE-DESCRIPTOR-COUNT 4096  OPEN-FILE-DESCRIPTOR-COUNT 36  Affect(row-cnt:0) cost in 259 ms. |

其中的GARBAGE-COLLECTORS 项，可以看到我们使用的垃圾回收器是 PS Scavenge：

|  |
| --- |
| GARBAGE-COLLECTORS  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  PS Scavenge 37/117(ms)  [count/time]  PS MarkSweep 6/589(ms)  [count/time] |

**thread**

定位线程问题



|  |
| --- |
| thread 线程号，可以看到具体线程内的堆栈调用信息：  [arthas@2874]$ thread 1  "main" Id=1 TIMED\_WAITING  at java.lang.Thread.sleep(Native Method)  at T15\_FullGC\_Problem01.main(T15\_FullGC\_Problem01.java:32)  Affect(row-cnt:0) cost in 23 ms. |

**dashboard**

类似于top命令，观察系统情况

**heapdump**

类似于jmap dump，会导出当前的堆存储文件，和jmap dump一样，也是非常影响系统性能的。

**以上都是其他工具也能实现的比较通用的功能，接下来说一说arthas比较独有的功能。**

**jad反编译**

可以对该进程内的代码进行反编译。

可用于：

1、动态代理生成类的问题定位

2、第三方的类（观察代码）

3、版本问题（确定自己最新提交的版本是不是被使用）

对Object进行反编译：

|  |
| --- |
| [arthas@3142]$ jad java.lang.Object  ClassLoader:  Location:    /\*  \* Decompiled with CFR.  \*/  package java.lang;  public class Object {  static {  Object.registerNatives();  }  protected void finalize() throws Throwable {  }  public final void wait() throws InterruptedException {  this.wait(0L);  }  public final void wait(long l, int n) throws InterruptedException {  if (l < 0L) {  throw new IllegalArgumentException("timeout value is negative");  }  if (n < 0 || n > 999999) {  throw new IllegalArgumentException("nanosecond timeout value out of range");  }  if (n > 0) {  ++l;  }  this.wait(l);  }  public final native void wait(long var1) throws InterruptedException;  public boolean equals(Object object) {  return this == object;  }  public String toString() {  return this.getClass().getName() + "@" + Integer.toHexString(this.hashCode());  }  public native int hashCode();  public final native Class<?> getClass();  protected native Object clone() throws CloneNotSupportedException;  public final native void notify();  public final native void notifyAll();  private static native void registerNatives();  }  Affect(row-cnt:1) cost in 423 ms. |

**redefine 热替换**

可以实现类的热替换，但目前有些限制条件：只能改方法实现（方法已经运行完成），不能改方法名， 不能改属性

**sc**

使用通配符搜索class名称。

**watch**

watch 让你能方便的观察到指定方法的调用情况。

能观察到的范围为：返回值、抛出异常、入参，通过编写 OGNL 表达式进行对应变量的查看。

**没有包含的功能**

它没有包含jmap的功能

**参考文档**

https://www.jianshu.com/p/507f7e0cc3a3