应用系统宕机问题

处理手册

版本：1.0

|  |  |  |
| --- | --- | --- |
| **版本修订记录** | | |
| 1.0 | 创建/2016-11-24 | 赵洲鹏 |
|  |  |  |

目录

[1 编写目的 5](#_Toc467766806)

[2 关于手册 6](#_Toc467766807)

[3 故障分类速查 7](#_Toc467766808)

[3.1 应用连接数据库故障 8](#_Toc467766809)

[3.1.1 从Websphere连接池获取连接超时 8](#_Toc467766810)

[3.1.2 Oracle数据库连接失败 8](#_Toc467766811)

[3.2 应用线程挂起 10](#_Toc467766812)

[3.2.1 Websphere WSVR0605W 10](#_Toc467766813)

[3.3 应用内存溢出 10](#_Toc467766814)

[3.3.1 堆内存溢出(Heap space) 10](#_Toc467766815)

[3.3.2 非堆内存溢出(PermGen space) 11](#_Toc467766816)

[3.4 IO句柄泄漏（Too Many Open Files） 11](#_Toc467766817)

[4 信息采集指引 12](#_Toc467766818)

[4.1 数据库信息采集（Oracle） 12](#_Toc467766819)

[4.1.1 Oracle processes设置 12](#_Toc467766820)

[4.1.2 Oracle 游标设置 12](#_Toc467766821)

[4.1.3 查看库表静态统计数据 13](#_Toc467766822)

[4.1.4 导出AWR/ADDM报告 13](#_Toc467766823)

[4.1.5 数据库告警日志 16](#_Toc467766824)

[4.1.6 Oracle锁定源查询 16](#_Toc467766825)

[4.2 应用服务器信息采集（Websphere） 17](#_Toc467766826)

[4.2.1 Websphere一般信息收集 17](#_Toc467766827)

[4.2.2 数据源连接池设置 18](#_Toc467766828)

[4.2.3 数据源（TYPE4）连接池设置 21](#_Toc467766829)

[4.3 操作系统信息采集（Linux） 23](#_Toc467766830)

[4.3.1 进程资源限制（ulimit） 23](#_Toc467766831)

[4.3.2 进程查询（ps） 24](#_Toc467766832)

[4.3.3 列出当前系统打开文件信息（lsof） 24](#_Toc467766833)

[4.3.4 内存信息(free) 25](#_Toc467766834)

[4.3.5 磁盘空间信息(df) 25](#_Toc467766835)

[4.4 JVM信息采集 25](#_Toc467766836)

[4.4.1 SUN/Oracle JVM信息采集 25](#_Toc467766837)

[4.4.2 IBM JVM信息采集 27](#_Toc467766838)

[4.5 应用日志采集 31](#_Toc467766839)

[4.5.1 Websphere 31](#_Toc467766840)

[5 问题诊断帮助 33](#_Toc467766841)

[5.1 从数据库连接池获取连接超时 33](#_Toc467766842)

[5.1.1 诊断步骤 33](#_Toc467766843)

[5.1.2 数据源监控和诊断（针对应用连接泄漏） 33](#_Toc467766844)

[5.1.3 其他可能造成问题的因素 37](#_Toc467766845)

[5.2 数据库连接失败 38](#_Toc467766846)

[5.2.1 诊断步骤 38](#_Toc467766847)

[5.2.2 其他说明 38](#_Toc467766848)

[5.3 应用线程挂起 39](#_Toc467766849)

[5.3.1 诊断步骤 39](#_Toc467766850)

[5.3.2 Java Core分析工具 40](#_Toc467766851)

[5.4 堆内存（Heap）溢出 41](#_Toc467766852)

[5.4.1 诊断步骤 42](#_Toc467766853)

[5.4.2 Heap Dump分析 42](#_Toc467766854)

[5.5 非堆内存（PermGen）溢出 46](#_Toc467766855)

[5.5.1 诊断步骤 46](#_Toc467766856)

[5.6 IO句柄泄漏（Too Many Open Files） 46](#_Toc467766857)

[5.6.1 诊断步骤 46](#_Toc467766858)

[5.6.2 代码review重点 46](#_Toc467766859)

[6 中间件已知问题汇总 48](#_Toc467766860)

[6.1 Websphere 48](#_Toc467766861)

[7 附录 49](#_Toc467766862)

[7.1 IBM JVM分析工具 49](#_Toc467766863)

[7.2 UltraEditor14 49](#_Toc467766864)

目录

[1 编写目的 5](#_Toc529540659)

[2 关于手册 5](#_Toc529540660)

[3 故障分类速查 5](#_Toc529540661)

[3.1 应用连接数据库故障 6](#_Toc529540662)

[3.1.1 从Websphere连接池获取连接超时 6](#_Toc529540663)

[3.1.2 Oracle数据库连接失败 7](#_Toc529540664)

[3.2 应用线程挂起 9](#_Toc529540665)

[3.2.1 Websphere WSVR0605W 9](#_Toc529540666)

[3.3 应用内存溢出 9](#_Toc529540667)

[3.3.1 堆内存溢出(Heap space) 9](#_Toc529540668)

[3.3.2 非堆内存溢出(PermGen space) 10](#_Toc529540669)

[3.4 IO句柄泄漏（Too Many Open Files） 10](#_Toc529540670)

[4 信息采集指引 10](#_Toc529540671)

[4.1 数据库信息采集（Oracle） 10](#_Toc529540672)

[4.1.1 Oracle processes设置 11](#_Toc529540673)

[4.1.2 Oracle 游标设置 11](#_Toc529540674)

[4.1.3 查看库表静态统计数据 11](#_Toc529540675)

[4.1.4 导出AWR/ADDM报告 12](#_Toc529540676)

[4.1.5 数据库告警日志 15](#_Toc529540677)

[4.1.6 Oracle锁定源查询 15](#_Toc529540678)

[4.2 应用服务器信息采集（Websphere） 16](#_Toc529540679)

[4.2.1 Websphere一般信息收集 16](#_Toc529540680)

[4.2.2 数据源连接池设置 17](#_Toc529540681)

[4.2.3 数据源（TYPE4）连接池设置 20](#_Toc529540682)

[4.3 操作系统信息采集（Linux） 22](#_Toc529540683)

[4.3.1 进程资源限制（ulimit） 22](#_Toc529540684)

[4.3.2 进程查询（ps） 22](#_Toc529540685)

[4.3.3 列出当前系统打开文件信息（lsof） 23](#_Toc529540686)

[4.3.4 内存信息(free) 23](#_Toc529540687)

[4.3.5 磁盘空间信息(df) 23](#_Toc529540688)

[4.3.6 列出进程中各线程的CPU占用情况 24](#_Toc529540689)

[4.4 JVM信息采集 24](#_Toc529540690)

[4.4.1 SUN/Oracle JVM信息采集 24](#_Toc529540691)

[4.4.2 IBM JVM信息采集 27](#_Toc529540692)

[4.5 应用日志采集 32](#_Toc529540693)

[4.5.1 Websphere 32](#_Toc529540694)

[5 问题诊断帮助 32](#_Toc529540695)

[5.1 从数据库连接池获取连接超时 32](#_Toc529540696)

[5.1.1 诊断步骤 32](#_Toc529540697)

[5.1.2 数据源监控和诊断（针对应用连接泄漏） 33](#_Toc529540698)

[5.1.3 其他可能造成问题的因素 37](#_Toc529540699)

[5.2 数据库连接失败 37](#_Toc529540700)

[5.2.1 诊断步骤 38](#_Toc529540701)

[5.2.2 其他说明 38](#_Toc529540702)

[5.3 应用线程挂起 39](#_Toc529540703)

[5.3.1 诊断步骤 39](#_Toc529540704)

[5.3.2 Java Core分析工具 39](#_Toc529540705)

[5.4 堆内存（Heap）溢出 41](#_Toc529540706)

[5.4.1 诊断步骤 41](#_Toc529540707)

[5.4.2 Heap Dump分析 41](#_Toc529540708)

[5.5 非堆内存（PermGen）溢出 44](#_Toc529540709)

[5.5.1 诊断步骤 45](#_Toc529540710)

[5.6 IO句柄泄漏（Too Many Open Files） 45](#_Toc529540711)

[5.6.1 诊断步骤 45](#_Toc529540712)

[5.6.2 代码review重点 45](#_Toc529540713)

[6 中间件已知问题汇总 45](#_Toc529540714)

[6.1 Websphere 46](#_Toc529540715)

[7 附录 46](#_Toc529540716)

[7.1 IBM JVM分析工具 46](#_Toc529540717)

[7.2 UltraEditor14 46](#_Toc529540718)

# 编写目的

针对应用系统出现宕机重大问题时，向一线现场实施/运维人员提供问题分析和线索收集指导，为下一步迅速定位故障原因、解决问题提供有力支持。

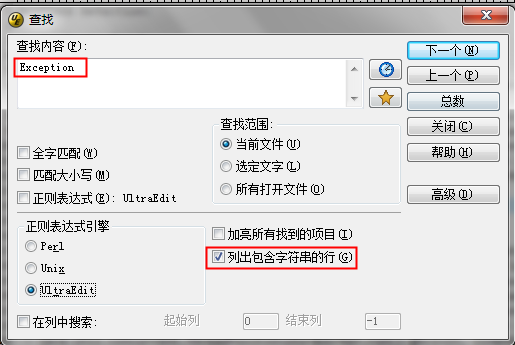
# 关于手册

* 由于硬件自身问题（断电/设备故障）造成的宕机及诊断恢复不在本文讨论范围内；
* 应用系统发生宕机时现场第一手的信息采集非常重要，如果特殊情况，请务必按本手册指引收集完成相关信息后，再进行应用系统重启，以保护问题相关线索；
* 手册主要针对宕机故障事后的一些处理与分析建议，但针对日常应用和数据库的关键性指标建议完善实时活动监控和预警措施，尽量在事前就能及时发现故障征兆，以采取积极措施；
* 手册会不断修订和更新，欢迎大家提供和补充新的案例。

# 故障分类速查

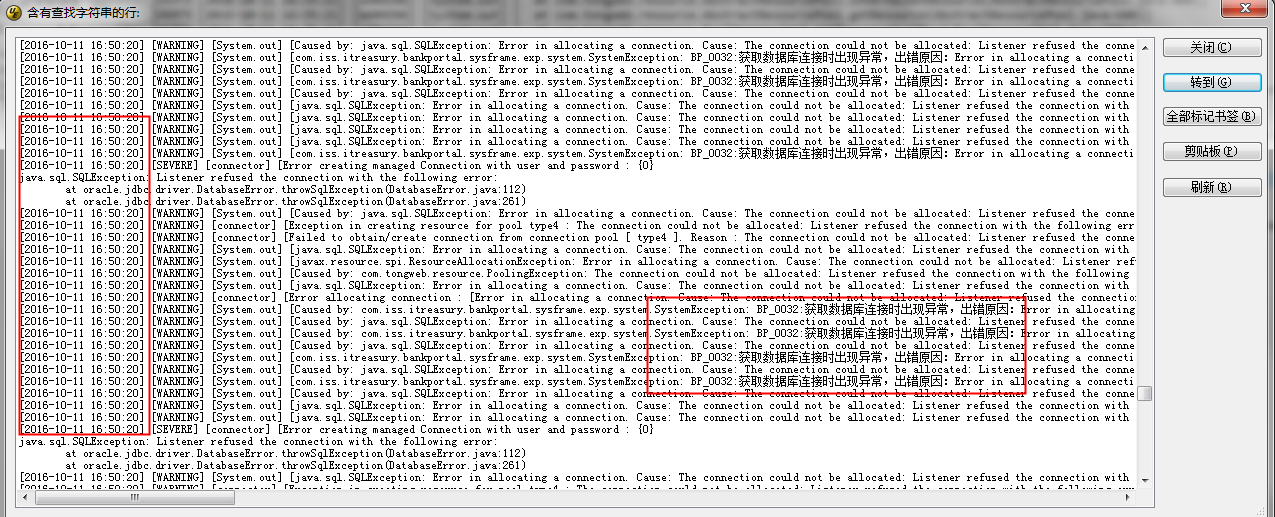
[获取应用日志](#_应用日志采集)后，通过检索包含"Exception"或者"Error"的所有行，来快速筛查当前应用中出现的问题；

日志查看和分析工具推荐使用[UltraEditor](#_UltraEditor14)：



通过日志中连续时段出现的大量相似异常信息，来初步定位到故障的大致分类；

比如通过下图可以明显看到在16:50:20左右输出了大量连接数据库方面的异常信息：



常见导致应用宕机的故障有：

* **应用连接数据库故障（如应用连接泄漏、数据库连接不足等）**
* **应用内存问题（内存溢出）**
* **应用线程挂起或死锁**
* **IO句柄泄漏（后台一直提示打开文件过多）**

下面将对这些故障的典型表现、相关信息收集及诊断步骤进行分别介绍：

## 应用连接数据库故障

### 从Websphere连接池获取连接超时

#### J2CA0045E

* **典型故障信息：**

|  |
| --- |
| **J2CA0045E**: 对资源 jdbc/cpf/type4 调用方法 createOrWaitForConnection 时，没有可用的连接。  com.ibm.websphere.ce.cm.ConnectionWaitTimeoutException:  Connection not available, Timed out waiting for XXXXX |

* **信息收集指引：**

|  |  |
| --- | --- |
| **1.确定数据源类型：（type4?）** | |
| **2.取Websphere数据源设置** | [传送门](#_数据源连接池设置) |
| **2.取Websphere Type4数据源设置** | [传送门](#_数据源（TYPE4）连接池设置) |
| **3.取应用日志** | [传送门](#_Websphere应用日志) |
| **4.取FFDC日志** | [传送门](#_Websphere_FFDC日志) |

* **问题诊断：**

**参考**[**5.1 从数据库连接池获取连接超时**](#_从数据库连接池获取连接超时)

### Oracle数据库连接失败

#### ORA-12519

* **典型故障信息：**

|  |
| --- |
| **connect db failed by type4 driver. java.sql.SQLException: Listener refused the connection with the following error:**  **ORA-12519, TNS:no appropriate service handler found**  **DSRA0010E: SQL 状态：66000，错误码：12,519。** |

* **信息收集指引：**

|  |  |
| --- | --- |
| **1.获取Oracle processes设置** | [传送门](#_Oracle_processes设置) |
| **2.统计应用数据源设置中的最大连接数** | [was数据源](#_Websphere数据源)  [was type4数据源](#_Websphere_Type4数据源) |
| **3.获取Oracle告警日志** | [传送门](#_数据库告警日志) |

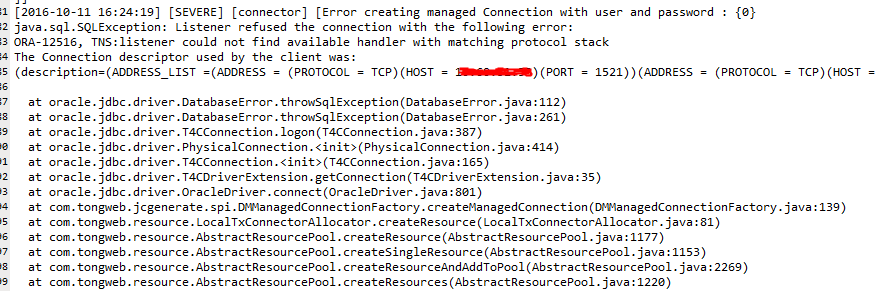
* **问题诊断：**

参考[5.2数据库连接失败](#_诊断步骤)诊断步骤

#### ORA-12516/ORA-12521

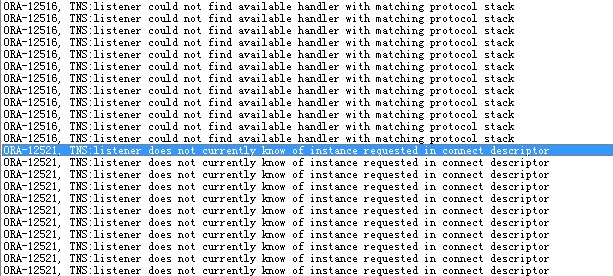
* **典型故障信息：**

|  |
| --- |
| **java.sql.SQLException: Listener refused the connection with the following error:**  **ORA-12516, TNS:listener could not find available handler with matching protocol stack** |

****

|  |
| --- |
| **java.sql.SQLException: Listener refused the connection with the following error:**  **ORA-12521, TNS:listener does not currently know of instance requested in connect descriptor** |

**日志中交互出现大量ORA-12516和ORA-12521错误：**

****

* **信息收集指引：**

|  |  |
| --- | --- |
| **1.逐个RAC实例，获取Oracle processes设置** | [传送门](#_Oracle_processes设置) |
| **2.统计应用数据源设置中的最大连接数** | [was数据源](#_Websphere数据源)  [was type4数据源](#_Websphere_Type4数据源) |
| **3.获取Oracle告警日志** | [传送门](#_数据库告警日志) |

* **问题诊断：**

参考[5.2数据库连接失败](#_诊断步骤)诊断步骤

## 应用线程挂起

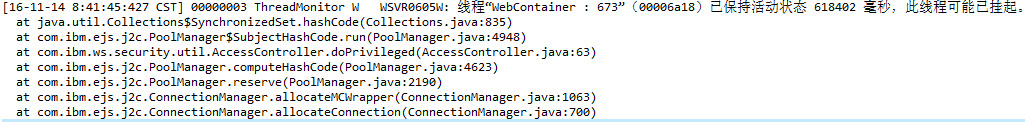
### Websphere WSVR0605W

* **典型故障信息：**

|  |
| --- |
| **WSVR0605W: 线程“WebContainer : 673”（00006a18）已保持活动状态 618402 毫秒，此线程可能已挂起。在服务器中共有 9 个线程可能处于挂起状态。**  **<Hung Thread Stack Trace>** |

* **信息收集指引：**

**1.上述故障信息后面附带的挂起线程的堆栈信息非常重要，显示了该挂起线程当前执行到的地方；例如：**

****

**2.获取**[**当前的Java Core**](#_Java_Core)**；**

**3.检查数据库**[**是否有锁定发生**](#_Oracle锁定源查询)**。**

* **问题诊断：**

参考[5.3 应用线程挂起](#_应用线程挂起)

## 应用内存溢出

### 堆内存溢出(Heap space)

* **典型故障信息：**

|  |
| --- |
| **#堆内存溢出**  **java.lang.OutOfMemoryError: Java heap space** |

* **信息收集指引：**

|  |  |
| --- | --- |
| **1.收集JVM虚拟机启动参数** | [**IBM JVM**](#_虚拟机启动参数)  [**SUN/Oracle JVM**](#_虚拟机启动参数_1) |
| **2.收集操作系统内存信息** | [**传送门**](#_内存信息(free)) |
| **3.收集堆内存转储文件** | [**IBM JVM**](#_堆内存转储（Heap_Dump）)  [**SUN/Oracle JVM**](#_堆内存转储（Heap_Dump）) |
| **4.收集Java Core线程dump** | [**IBM JVM**](#_Java_Core)  [**SUN/Oracle JVM**](#_Java_Core/Thread_dump) |

* **问题诊断：**

参考[5.4 堆内存溢出](#_堆内存（Heap）溢出)

### 非堆内存溢出(PermGen space)

* **典型故障信息：**

|  |
| --- |
| **java.lang.OutOfMemoryError: PermGen space** |

* **信息收集指引：**

|  |  |
| --- | --- |
| **收集JVM虚拟机启动参数** | [**IBM JVM**](#_虚拟机启动参数)  [**SUN/Oracle JVM**](#_虚拟机启动参数_1) |

* **问题诊断：**

参考[5.5 JVM非堆内存溢出](#_永久保存区溢出)

## IO句柄泄漏（Too Many Open Files）

* **典型故障信息：**

|  |
| --- |
| **java.net.SocketException: Too many open files**  **java.io.IOException：打开的文件过多** |

* **信息收集指引：**

|  |  |
| --- | --- |
| **1.取服务器操作系统进程资源限制** | [传送门](#_进程资源限制（ulimit）) |
| **2.查找Java进程和进程的打开文件信息** | [传送门](#_列出当前系统打开文件信息（lsof）) |

* **问题诊断：**

请参考[5.6 IO句柄泄漏](#_IO句柄泄漏（Too_Many_Open)

# 信息采集指引

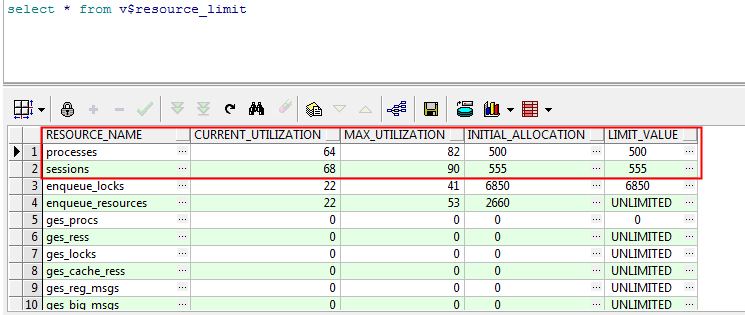
## 数据库信息采集（Oracle）

### Oracle processes设置

#查看processes/sessions设置以及统计（需sys/dba权限）：

|  |
| --- |
| #在SQL窗口执行  select \* from v$resource\_limit |

例如：



注：CURRENT\_UTILIZATION为当前占用统计；

MAX\_UTILIZATION为数据库曾达到的最大值；

LIMIT\_VALUE为目前数据库设置的限制值。

### Oracle 游标设置

#查看open cursor设置：

|  |
| --- |
| **#通过PL/SQL Developer命令窗口或者SQLPLUS执行**  **show parameter cursor;** |

例如：

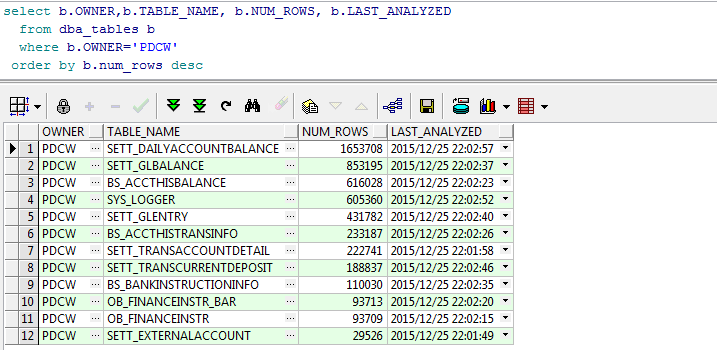


### 查看库表静态统计数据

#显示当前数据库表（按行数排序）：

|  |
| --- |
| **select b.OWNER,b.TABLE\_NAME, b.NUM\_ROWS, b.LAST\_ANALYZED**  **from dba\_tables b**  **where b.OWNER='<用户名>'**  **order by b.num\_rows desc** |

例如：



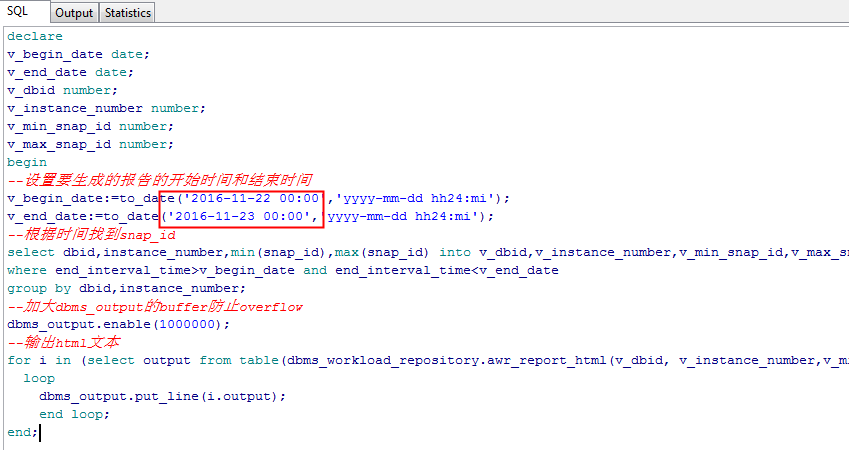
### 导出AWR/ADDM报告

#1. 通过PL/SQL Developer连接Oralce数据库（使用sys或者dba权限的用户）

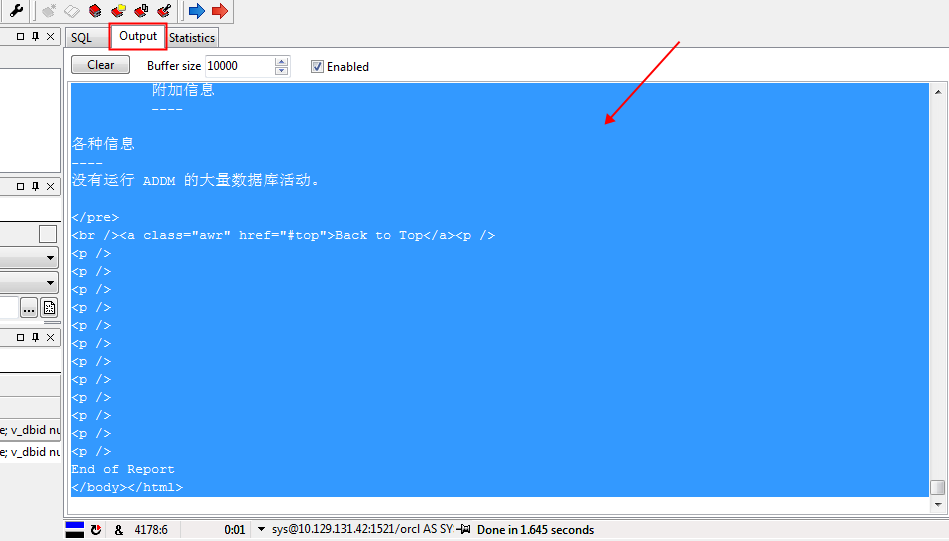
#2. 在SQL窗口中贴入：



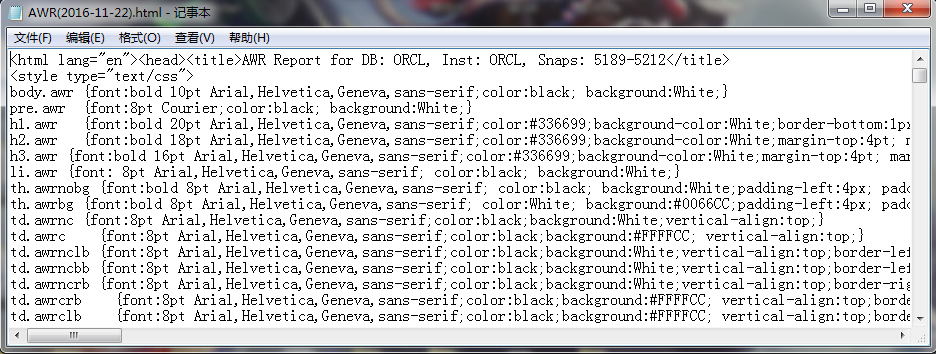
#3. 调整其中开始时间和截止时间为欲分析的时间段：



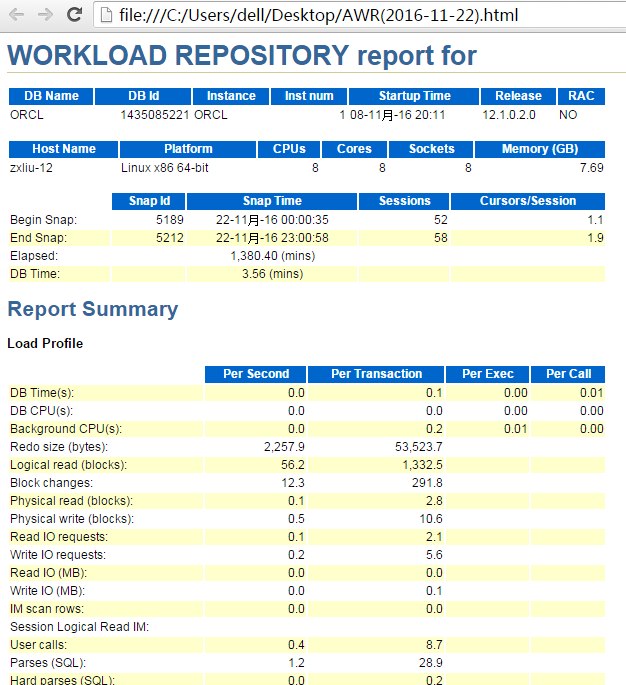
#4. 执行；完毕后将Output中的内容全部拷贝；



#5. 新建一html文件，将内容粘贴入其中；

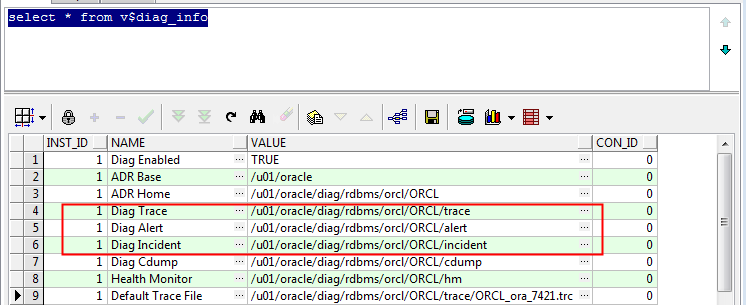


#6. 保存即可

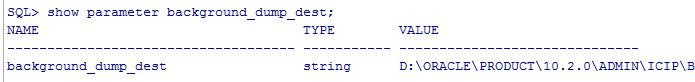


### 数据库告警日志

* **oracle11**以上，通过查询v$diag\_info来获取告警日志所在目录，如下图所示：



* **oracle10g**，通过查询background\_dump\_dest参数来获取告警日志所在目录：



* 通过数据库操作系统用户，登录数据库服务器，并到相应目录获取日志

### Oracle锁定源查询

注：下述SQL需要具备dba权限的用户执行

|  |
| --- |
| **--列出锁定已超过60秒的对象**  SELECT SUBSTR(TO\_CHAR(w.session\_id), 1, 5) WSID,  p1.spid WPID,  SUBSTR(s1.username, 1, 12) "WAITING User",  SUBSTR(s1.osuser, 1, 8) "OS User",  SUBSTR(s1.program, 1, 20) "WAITING Program",  s1.client\_info "WAITING Client",  w.last\_convert "WAITING TIME(second)",  SUBSTR(TO\_CHAR(h.session\_id), 1, 5) HSID,  p2.spid HPID,  SUBSTR(s2.username, 1, 12) "HOLDING User",  SUBSTR(s2.osuser, 1, 8) "OS User",  SUBSTR(s2.program, 1, 20) "HOLDING Program",  s2.client\_info "HOLDING Client",  o.object\_name "HOLDING Object"  FROM gv$process p1,  gv$process p2,  gv$session s1,  gv$session s2,  dba\_locks w,  dba\_locks h,  dba\_objects o  WHERE w.last\_convert > 60  AND h.mode\_held != 'None'  AND h.mode\_held != 'Null'  AND w.mode\_requested != 'None'  AND s1.row\_wait\_obj# = o.object\_id  AND w.lock\_type(+) = h.lock\_type  AND w.lock\_id1(+) = h.lock\_id1  AND w.lock\_id2(+) = h.lock\_id2  AND w.session\_id = s1.sid(+)  AND h.session\_id = s2.sid(+)  AND s1.paddr = p1.addr(+)  AND s2.paddr = p2.addr(+)  ORDER BY w.last\_convert DESC; |

|  |
| --- |
| **--根据SID查询相关SQL**  select sql\_text  from v$sql s  where hash\_value in (  select sql\_hash\_value from v$session where sid = ?) |

## 应用服务器信息采集（Websphere）

### Websphere一般信息收集

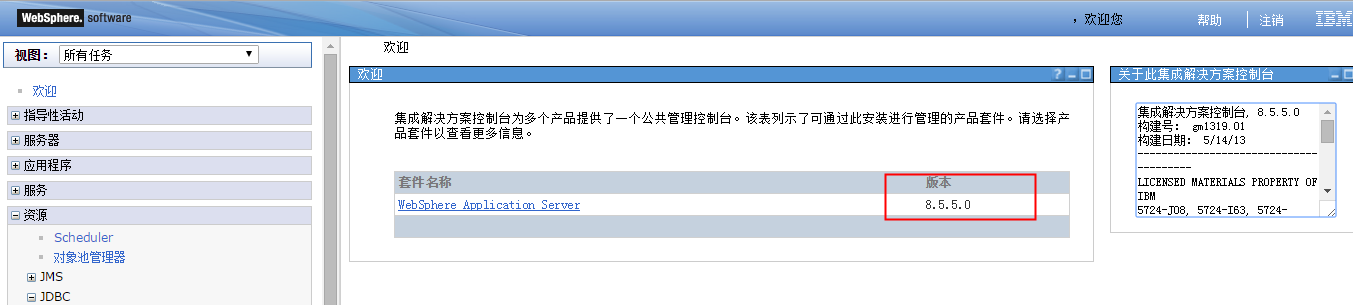
* Websphere详细版本号：

#1. 登入Websphere控制台（通常控制台地址为https://服务器IP:9043/ibm/console）



#2. 输入was控制台管理用户标识和密码，点击登录；

图中显示版本信息



### 数据源连接池设置

***注：下面示例以was8.5为参照，其他版本变动不大；***

#1. 登入Websphere控制台（通常控制台地址为https://服务器IP:9043/ibm/console）



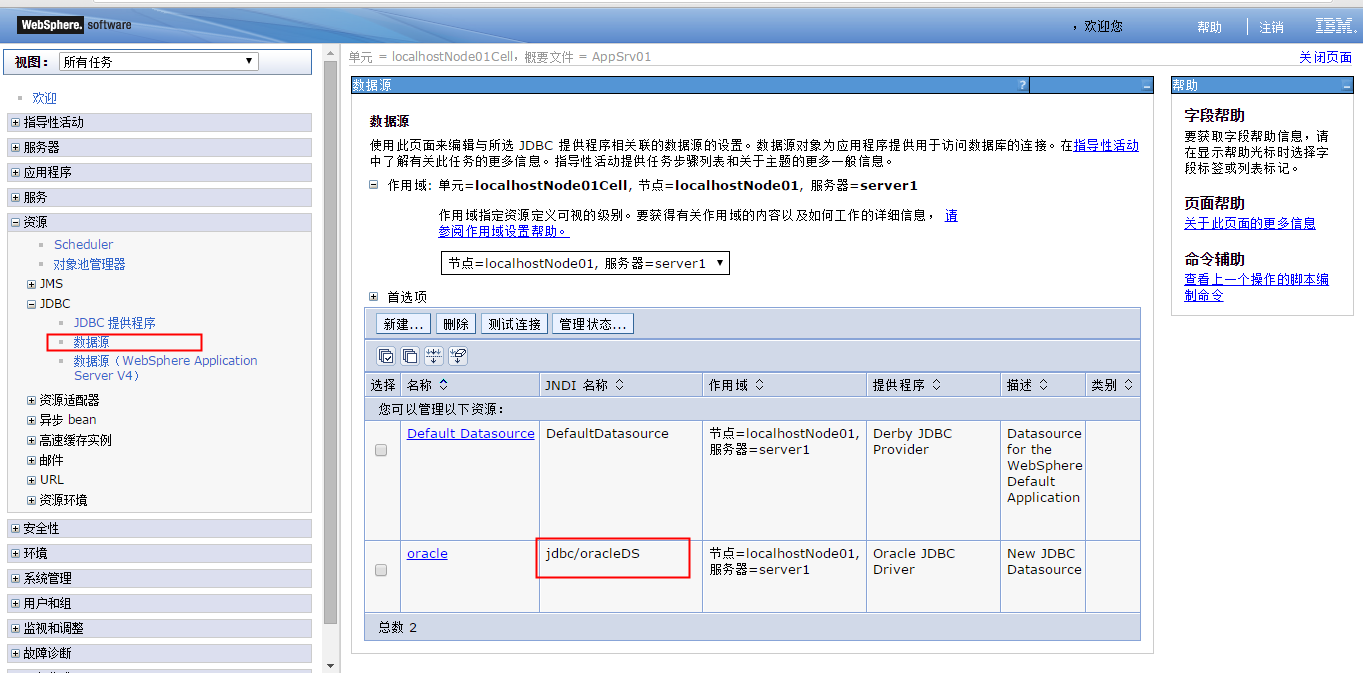
#2. 输入was控制台管理用户标识和密码，点击登录；

#3. 进入【资源】>【JDBC】>【数据源】：



#4. 根据JNDI Name选择目前系统所用的数据源：

（注：财务公司系统通常为**jdbc/cpf/type4**）



#5. 点击链接打开详细，然后点击【连接池属性】：



#6 .截图：



### 数据源（TYPE4）连接池设置

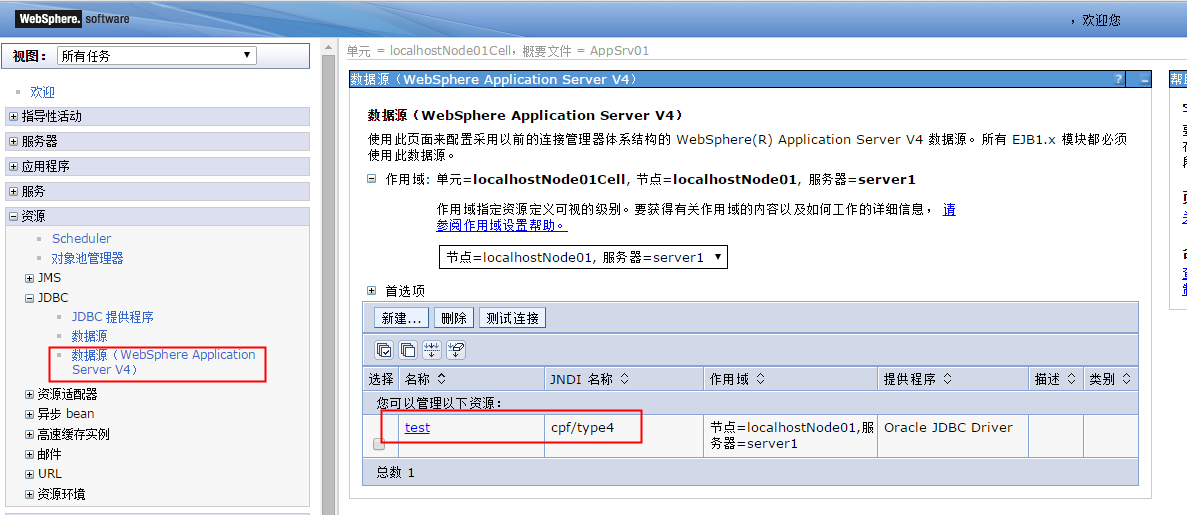
***注：下面示例以was8.5为参照，其他版本变动不大；***

#1. 登入Websphere控制台（通常控制台地址为https://服务器IP:9043/ibm/console）



#2. 输入was控制台管理用户标识和密码，点击登录；

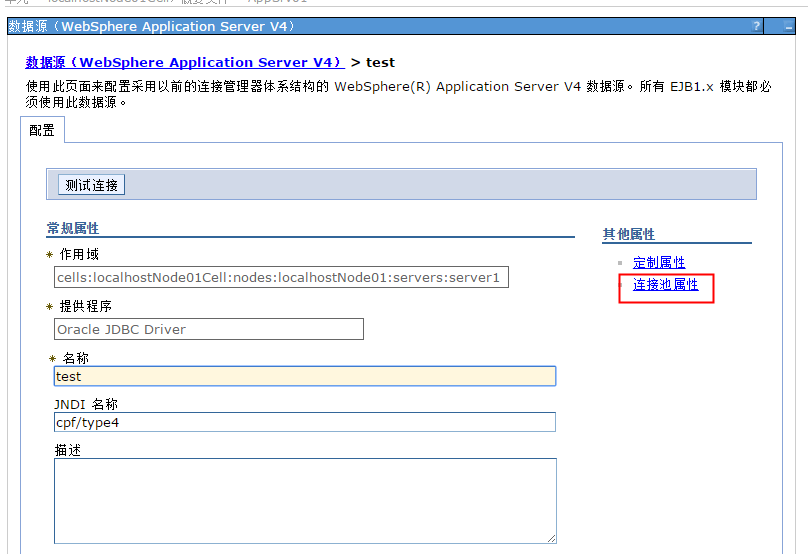
#3. 进入【资源】>【JDBC】>【数据源(WebSphere Application Server V4)】：



#4. 根据JNDI Name选择目前系统所用的数据源：

（注：财务公司系统通常为**jdbc/cpf/type4**）

#5. 点击链接打开详细，然后点击【连接池属性】：



#6 .截图：



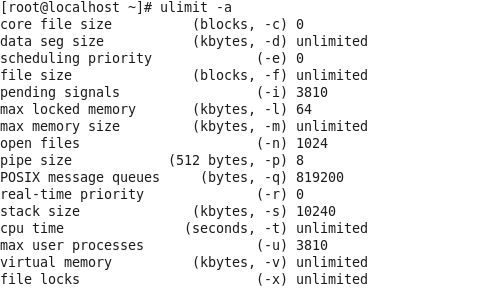
## 操作系统信息采集（Linux）

### 进程资源限制（ulimit）

查看当前用户ulimit设置的所有信息：

|  |
| --- |
| ulimit -a |

示例：



### 进程查询（ps）

#显示所有进程的UID、PPID、C、STIME

|  |
| --- |
| ps -ef |

#显示所有java进程：

|  |
| --- |
| ps -ef | grep java |

示例：



注：上图中箭头所指即当前系统中java进程（was进程）对应的进程号（PID）；

### 列出当前系统打开文件信息（lsof）

#列出当前系统所有打开文件的信息（输出到当前目录的all\_openfile.txt中）

|  |
| --- |
| **lsof > all\_openfile.txt** |

#显示当前系统中open file数最多的TOP5进程：

|  |
| --- |
| **lsof -n | awk '{print $2}' | sort | uniq -c | sort -nr | head -n 5 > openfile\_top5.txt** |

#根据已知的进程号（通过3.3.2进程查询），获取其全部open file信息：

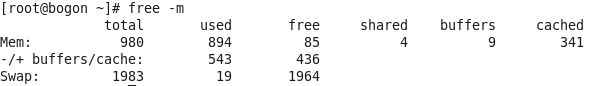
|  |
| --- |
| **#例如查询进程29680的全部open file信息**  **lsof -p 29680 > lsof\_pid.txt** |

### 内存信息(free)

#查看内存信息（以m为单位）

|  |
| --- |
| **free -m** |

例如：

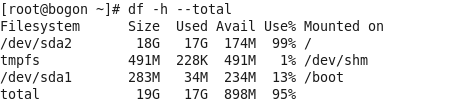


### 磁盘空间信息(df)

#显示所有文件系统的磁盘使用情况以及汇总

|  |
| --- |
| **df -h --total** |

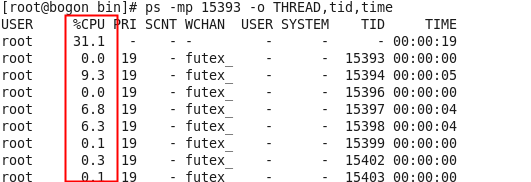
例如：



### 列出进程中各线程的CPU占用情况

|  |
| --- |
| **ps -mp 进程号 -o THREAD,tid,time** |

例如：



|  |
| --- |
| **#转换tid为16进制**  **printf "%x\n" tid** |

## JVM信息采集

### SUN/Oracle JVM信息采集

#### 虚拟机启动参数

#导出java进程以及其启动参数：

|  |
| --- |
| **ps -ef|grep java > java\_process.txt** |

例如：



/bin/java 后面跟的-Dxxx=xxx即为相关启动参数

#### 堆内存转储（Heap Dump）



* 在OOM时自动生成转储文件：

|  |
| --- |
| #在java启动参数中添加：  **-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=</path/to/file>** |

* 手工对目前运行中的JVM生成heap dump：

#1. 查找Java进程号(pid)：

|  |
| --- |
| **ps -ef|grep java** |

#2. 生成Java进程的dump：

|  |
| --- |
| **jmap -dump:format=b,file=文件名 [pid]**  **例如：**  **Jmap -dump:format=b,file=jvm.dmp 2331** |

**注意：对运行中的系统手工生成heap dump会造成系统卡顿，不要在业务繁忙的时刻生成heap dump**

**注：在solars系统的****SDK 1.5.0\_16 及更高情况下，默认kill -3只会生成javacore文件，不会生成堆内存转储文件，需要添加参数:**

**javacore：**

**/soft/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1/native\_stdout.log中.**

**cat >/soft/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1/native\_stdout.log**

**heapdump:**

**默认存放在AppSrv/profiles/<...>的根目录下面**

**java\_pid5907.hprof.20170601.130500**

**ls -l /soft/IBM/WebSphere/AppServer/profiles/AppSrv01/ | grep java**

**配置下面参数可以使用SIGQUIT - kill -3 [JVM PID] 生成heapdump：  
  -XX:+HeapDumpOnCtrlBreak**

**为了使JVM能够在发生内存溢出问题的时候生成 heapdump ，我们需要配置如下JVM通用参数：  
  -XX:+HeapDumpOnOutOfMemoryError**

**增加下面的参数后，可以用SIGQUIT - kill -3 [JVM PID]生成类柱状图，该信息会给写入native\_stdout.log而无须使用jmap工具：  
  -XX:+PrintClassHistogram**

#### Java Core/Thread dump



#1. 查找Java进程号(pid)：

|  |
| --- |
| **ps -ef|grep java** |

#2. 生成Java进程的thread dump：

|  |
| --- |
| **jstack -F pid > stack.txt**  **例如：**  **jstack -F 2331 > stack.txt** |

#### GC日志采集

#1. 查找Java进程号(pid)：

|  |
| --- |
| **ps -ef|grep java** |

#2. 查看进程的GC统计信息

|  |
| --- |
| **jstat -gc pid > gc.txt**  **例如：**  **jstat -gc 2331 > gc.txt** |

### IBM JVM信息采集

#### 虚拟机启动参数

#导出java进程以及其启动参数：

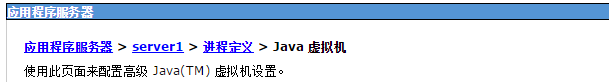
|  |
| --- |
| **ps -ef|grep java > java\_process.txt** |

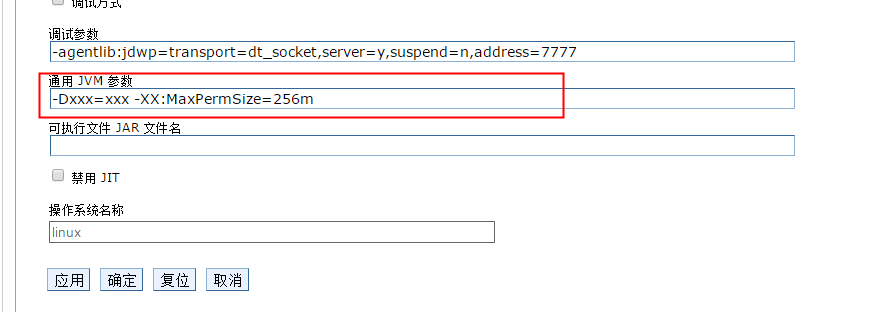
例如：



/bin/java 后面跟的-Dxxx=xxx即为相关启动参数

#websphere7设置JVM参数：





（设置完毕后需要重启websphere才能生效）

#### 堆内存转储（Heap Dump）

* 自动生成转储文件：

Websphere中间件在内存溢出(OutOfMemoryError)时默认会自动在：

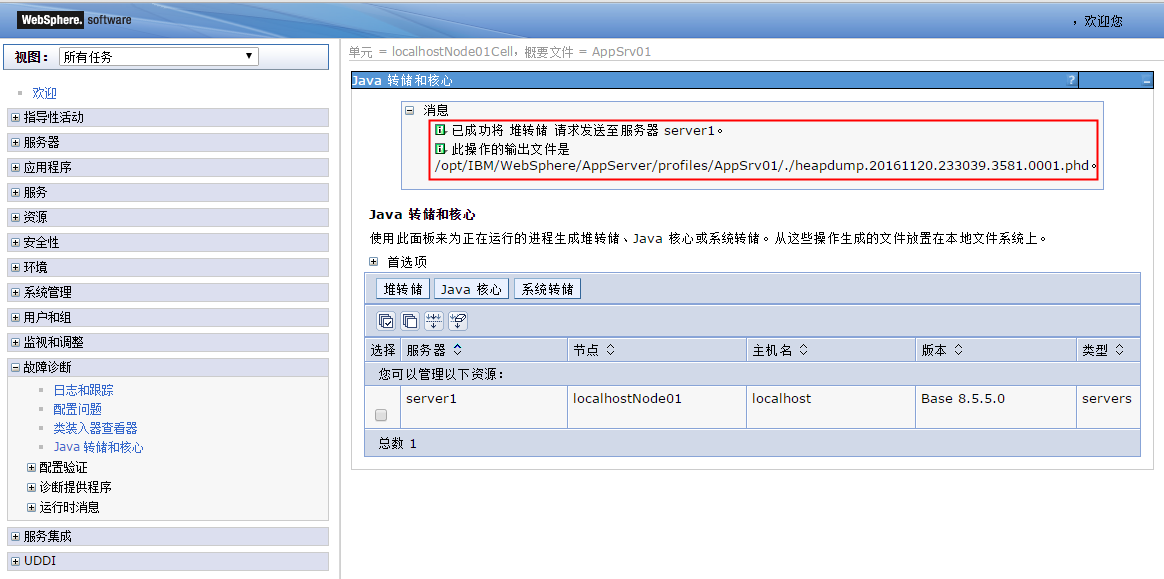
**<WAS\_HOME>/AppServer/profiles/AppSrv01**

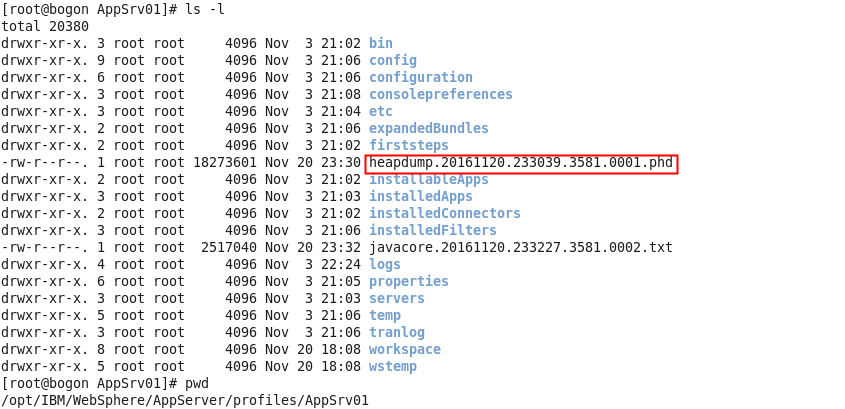
下生成堆内存转储文件(**\*.phd**)和Java Core文件(**javacore.\*.txt**)，在由于内存溢出或者线程挂起导致宕机时，请务必等待dump文件生成完毕后再杀进程和重启应用；

* 手工生成内存转储文件：
* **WAS 8或者以上版本**

**a）控制台生成**







b）命令行执行:

cd /sett/IBM/WebSphere/AppServer/profiles/AppSrv03/bin

./wsadmin.sh

wsadmin>set jvm [$AdminControl completeObjectName type=JVM,process=server1,\*]

wsadmin>$AdminControl invoke $jvm generateHeapDump

wsadmin>$AdminControl invoke $jvm dumpThreads

* **WAS 8以下版本**

（1）添加环境变量：IBM\_HEAPDUMP=true

|  |
| --- |
| **#Linux**  **echo "export IBM\_HEAPDUMP=true" >> /etc/profile**  **source /etc/profile** |

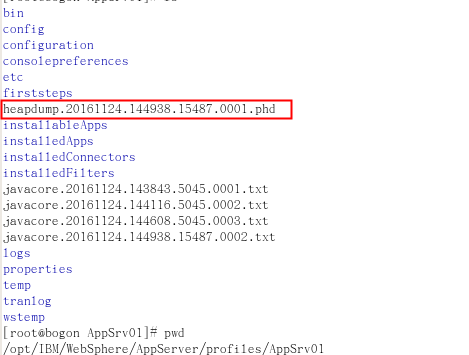
或者在JVM参数中添加：

|  |
| --- |
| **#针对IBM JVM**  **-Xdump:heap:events=user** |

（2）重启WAS；

（3）需要手工生成HEAPDUMP时，执行**kill -3 进程号**，默认生成在AppSrv01/下。

**注意：不要与kill -9 进程号 （立即杀死进程）混淆**

****

#### Java Core

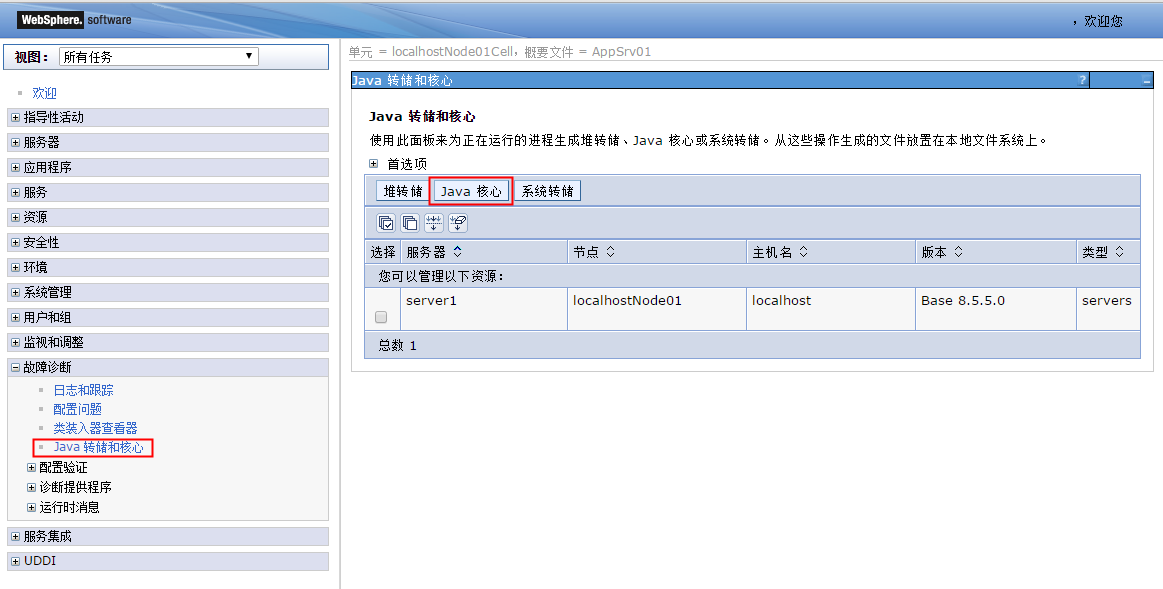
* 自动生成转储文件：

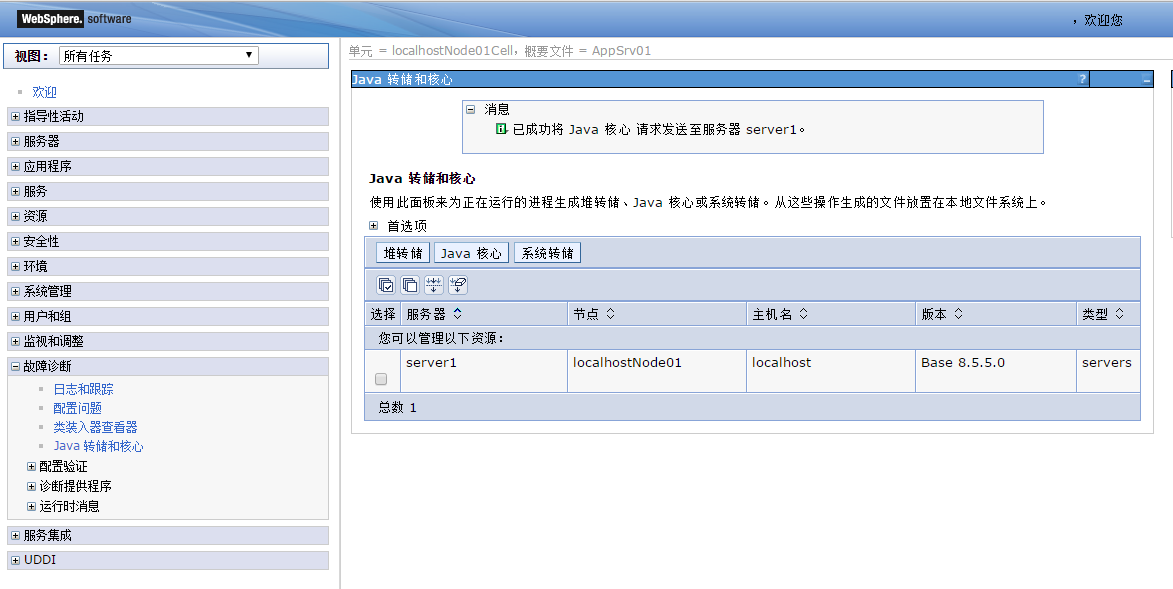
Websphere中间件在内存溢出(OutOfMemoryError)或者其他异常宕机时默认会自动在：

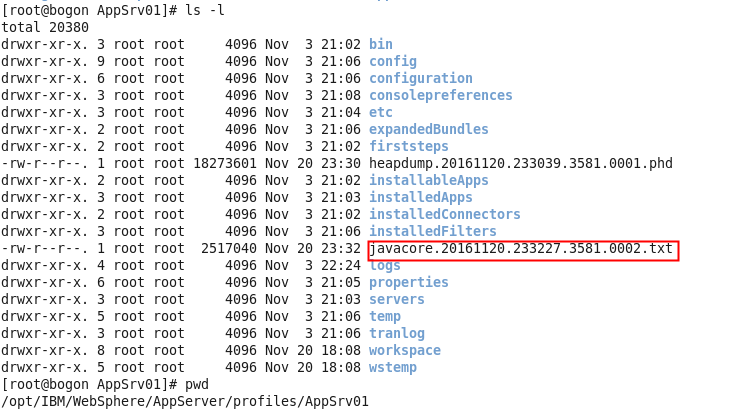
**<WAS\_HOME>/AppServer/profiles/AppSrv01**

下生成堆内存转储文件(**\*.phd**)和Java Core文件(**javacore.\*.txt**)，在由于内存溢出或者线程挂起导致宕机时，请务必等待dump文件生成完毕后再杀进程和重启应用；

* **WAS 8或者以上版本**







* **WAS 8 以下版本**

|  |
| --- |
| **kill -3 <PID>** |

**#例如java进程号为5045**

**kill -3 5045**

**将在AppSrv01下生成java core：**

****

#### GC日志采集

|  |
| --- |
| **-verbose:gc**  **-Xverbosegclog** |

## 应用日志采集

### Websphere

#### 应用日志

通常关注的是SystemOut.log（控制台日志）、SystemErr.log（异常日志）和trace.log（诊断日志）

|  |
| --- |
| #SystemOut.log、SystemErr.log、trace.log日志所在目录：  **<WAS安装目录>/AppServer/profiles/AppSrv01/logs/server1**  #例如：  **/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1** |

#### FFDC日志

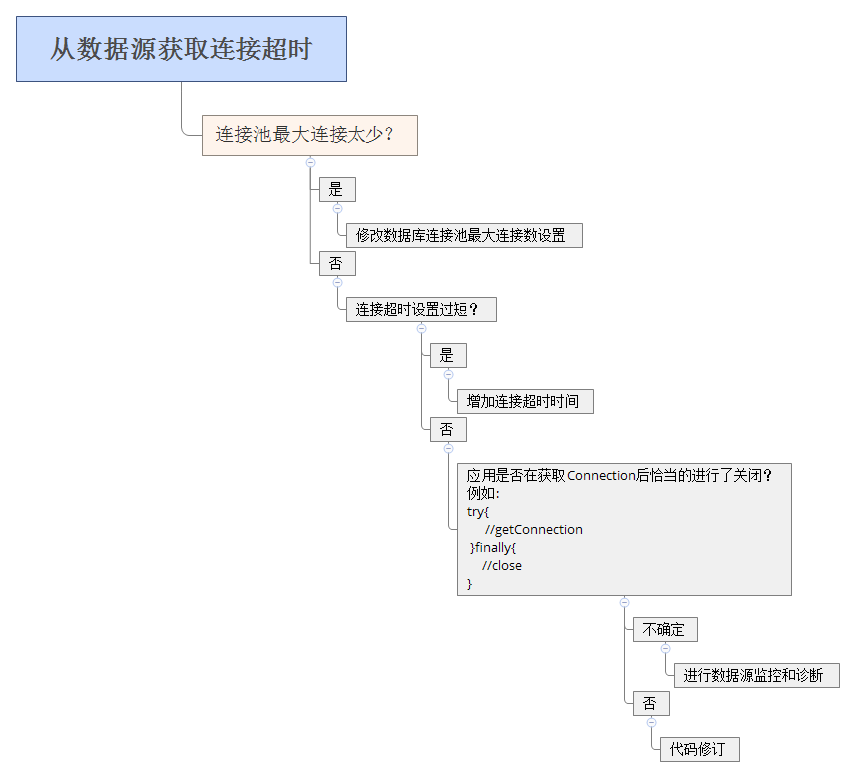
FFDC（首次失败数据获取）工具会尝试在发生可能的非正常情况时，自动捕获并保存关键信息，其中包含堆栈跟踪、异常发生时的环境，以及对生成这一异常的服务器的组件状态所做的一个快照；发生故障时可以通过FFDC深入了解问题的详细情况。

|  |
| --- |
| #FFDC日志所在目录：  **<WAS 安装目录>/AppServer/profiles/AppSrv01/logs/ffdc**  #例如：  **/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/ffdc** |

# 问题诊断帮助

## 从数据库连接池获取连接超时

### 诊断步骤



### 数据源监控和诊断（针对应用连接泄漏）

#### Websphere数据源

（1）进入【日志和跟踪】>【诊断跟踪】



（2）将最大文件大小适当调大（比如200M），然后更改日志详细信息级别

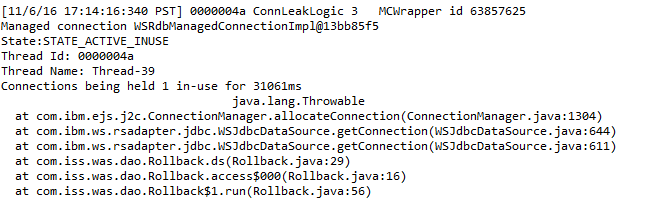


（3）在原来\*=info后面添加 **: WAS.j2c=all**



（4）保存后重启Websphere生效；

当再次发生从数据源获取连接超时时，was会在trace.log日志（跟SystemOut.log同级目录）中输出当前连接池内部连接的详细信息（包括应用getConnection时的线程堆栈，以及持有时间）：



通过连接持有时间定位到可疑代码，进行代码review；

#### Websphere Type4数据源

(1) 进入type4数据源的定制属性页面：





(2)新增自定义属性 diagOptions=4：



(3)保存后重启Websphere服务器，当再次发生获取连接超时时，Websphere在SystemOut.log和trace.log日志中会将当前连接池的全部连接线程以及应用getConnection时的线程堆栈信息进行输出；通过连接持有时间定位到可疑代码，进行代码review；

#### 开源数据源（c3p0 / dhcp /tomcat pool）等

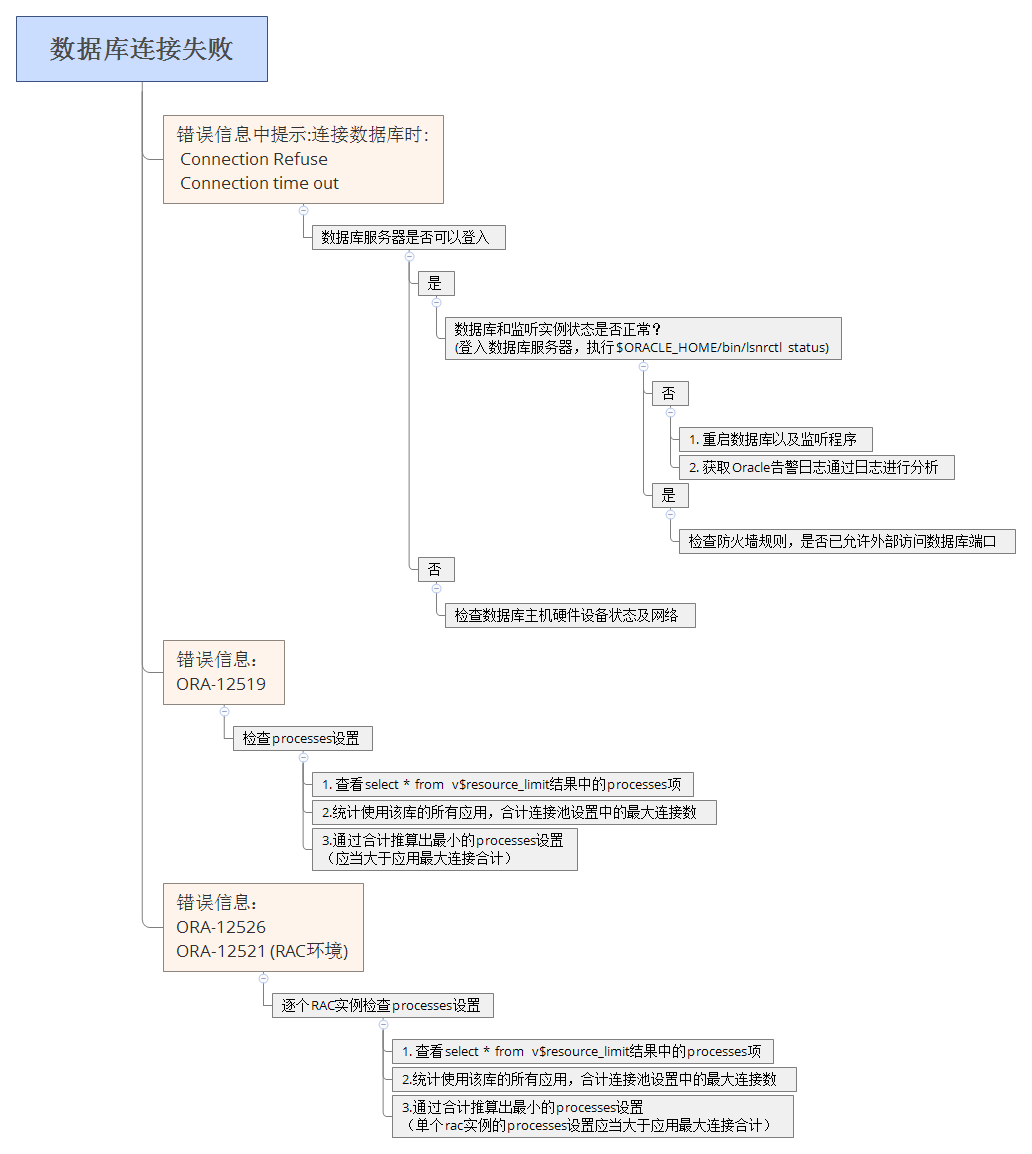
推荐替换为阿里巴巴开源的druid数据源(https://github.com/alibaba/druid)，druid提供了应用使用数据源时非常丰富的监控统计数据，例如SQL执行耗时统计、连接池中每个连接打开时的线程堆栈等信息，非常方便排查应用程序的连接泄漏问题。

### 其他可能造成问题的因素

* 不合理的应用程序设计（比如过大的事务、非常低效的SQL操作（例如对于大表的全表扫描）也会造成应用程序长时间持有连接，而在并发业务量较大的时候产生获取连接超时的问题
* 对于上述建议取故障时段前后一小时的[数据库AWR](#_导出AWR/ADDM报告)进行辅助分析

## 数据库连接失败

### 诊断步骤

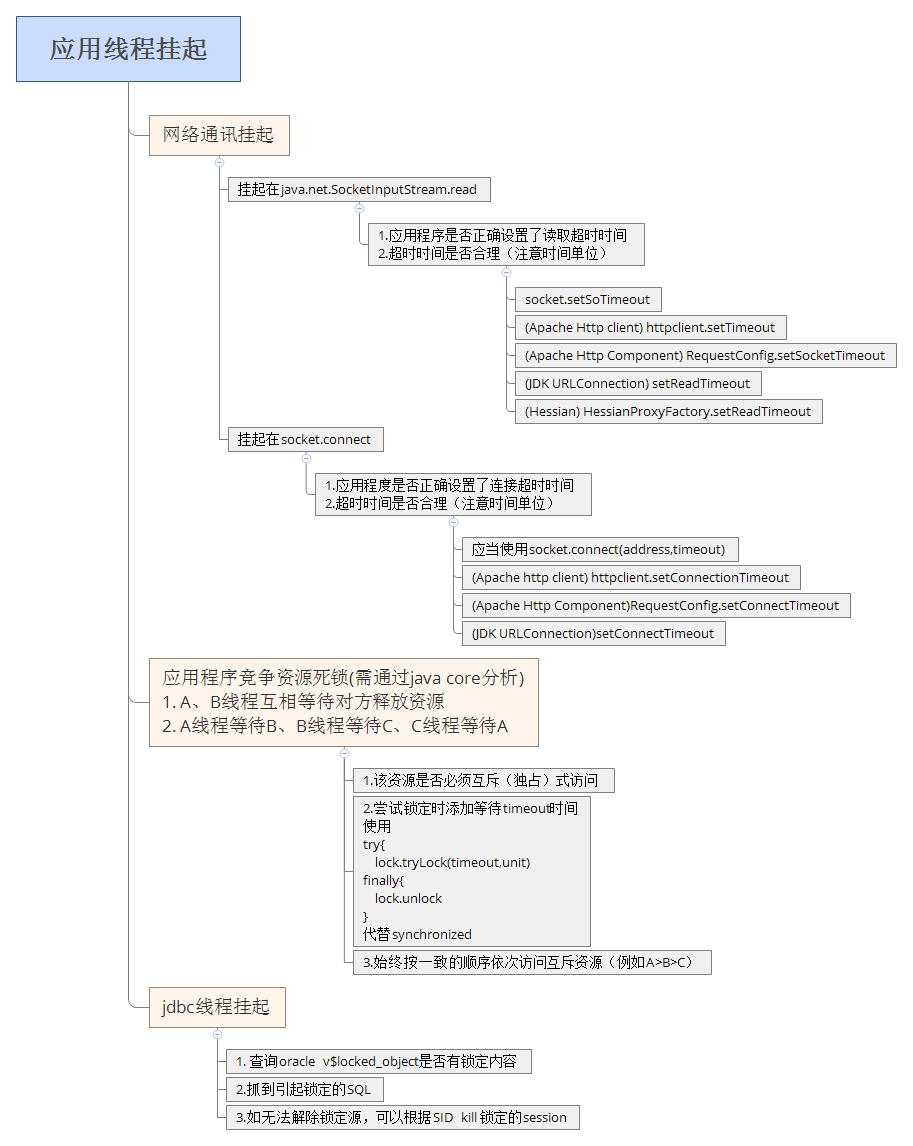


### 其他说明

* 应用程序连接池的最大连接设置应当在合理的范围，正常情况应当通过压力测试得出当前环境应用使用数据库的连接峰值；
* 如果连接池的最大连接已经足够了，但应用程序仍不够用，应当进行应用程序连接泄漏问题诊断。

## 应用线程挂起

### 诊断步骤



### Java Core分析工具

（1）通过jstack导出的线程DUMP可以直接通过文本编辑/查看器进行查看：

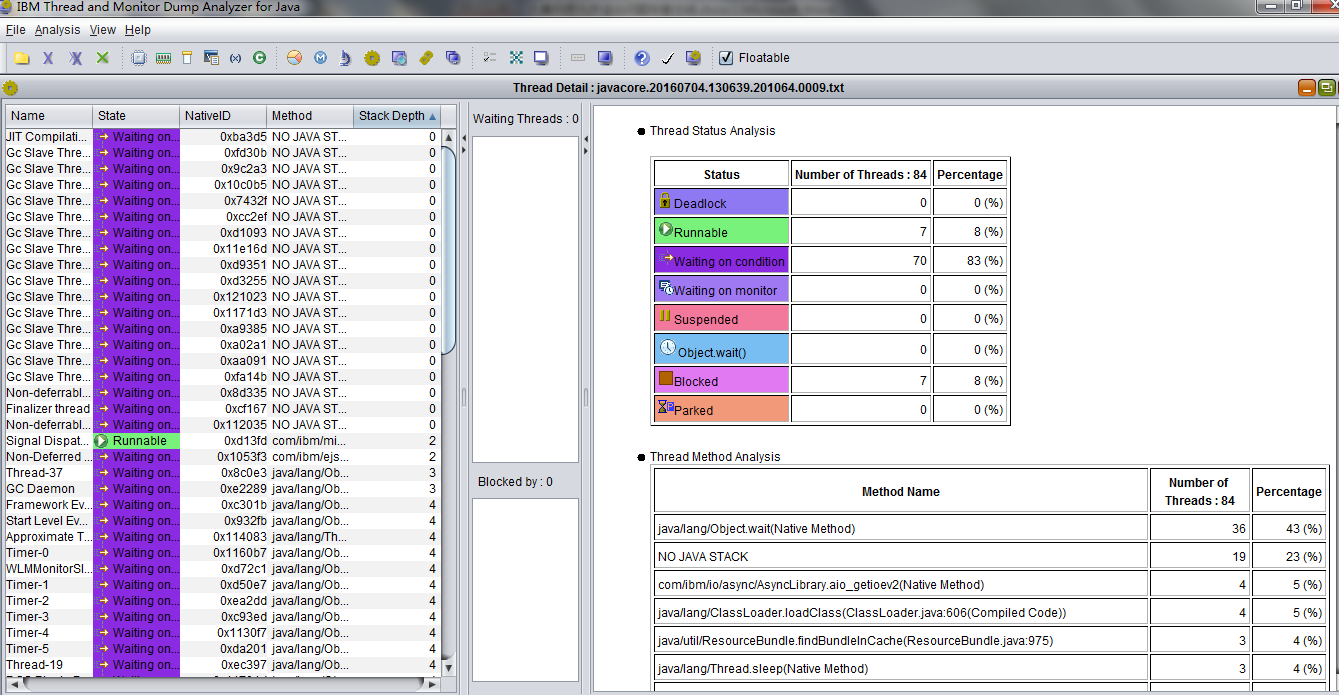


jstack可以自动侦测和提示应用程序中的死锁；对于正常的IO阻塞等待可以通过查看线程栈辅助进行定位分析；

（2）对于was生成的javacore文件，可以通过IBM Thread and Monitor Dump Analyzer for Java工具分析；

[附录7.1中的jca457.jar](#_IBM_JVM分析工具)（通过 java -jar jca457.jar 运行）

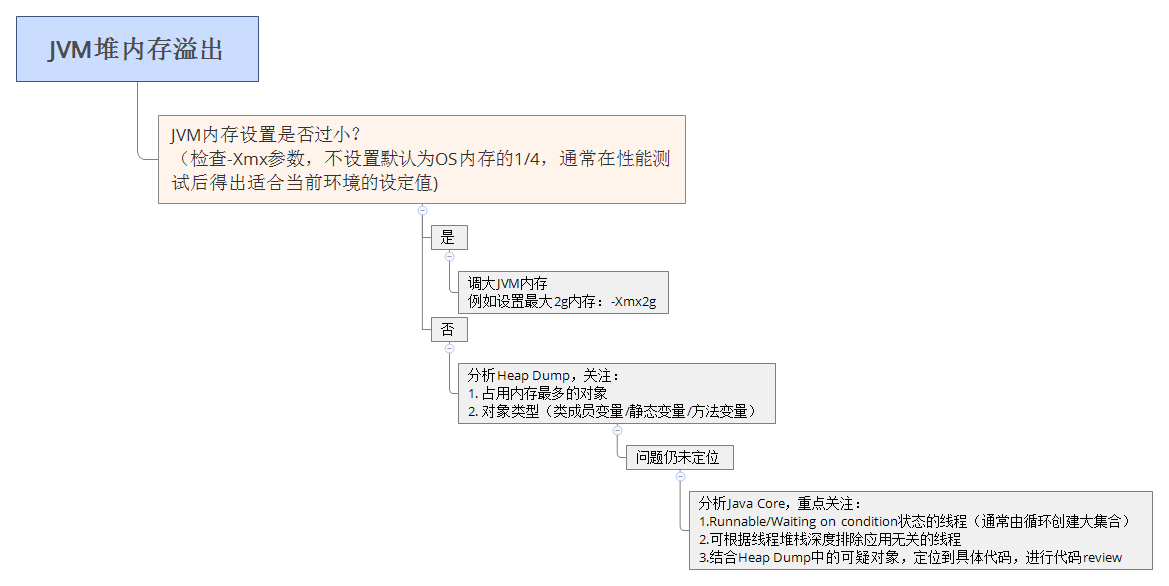




## 堆内存（Heap）溢出

问题分析人员应当对JVM结构、常见参数和垃圾回收机制有一定了解。

### 诊断步骤



### Heap Dump分析

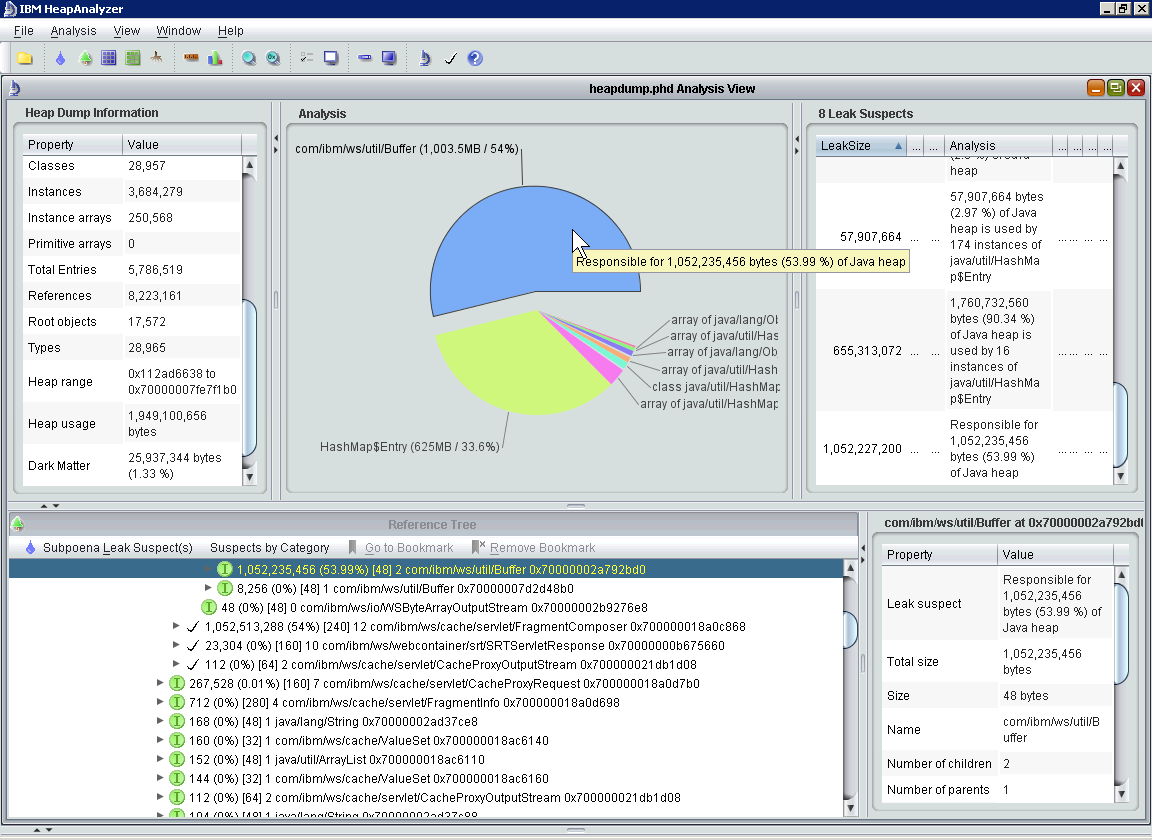
#### 工具

（1）[MAT(Memory Analyze Tool)](http://www.eclipse.org/mat/)：

主要针对Sun/Oracle的Hotspot JVM的内存dump进行分析

（2）[IBM Heap Analyzer](https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=4544bafe-c7a2-455f-9d43-eb866ea60091)：

主要针对IBM JVM的内存dump进行分析（[附录7.1 IBM JVM分析工具](#_IBM_JVM分析工具)ha456.jar）；



通过

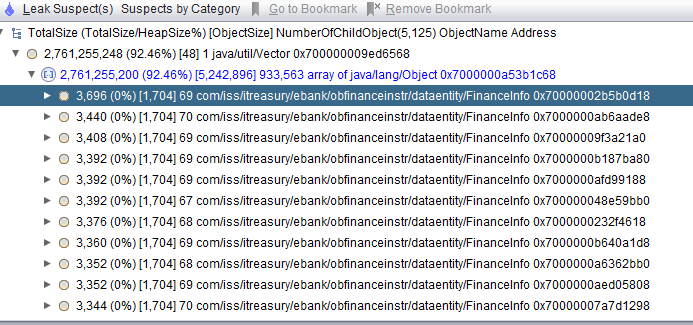
java -jar ha456.jar运行，如果内存不够打开dump文件，通过-Xmx参数调大内存

（例如java -Xmx4g -jar ha456.jar）

#### DUMP分析示例

上述两个工具都提供了对内存中的对象按占用空间大小进行分组统计和排序等功能，可以比较直观的查看当前JVM内存的使用情况；

例如下面是比较典型的应用程序导致的内存泄漏问题，通过IBM Heap Analyzer打开dump后：

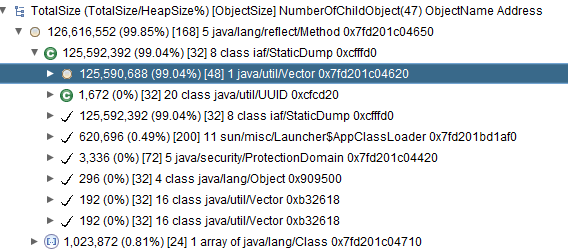


内存中92.46%的空间被一个Vector集合占用，内部全是业务系统创建的FinanceInfo对象。

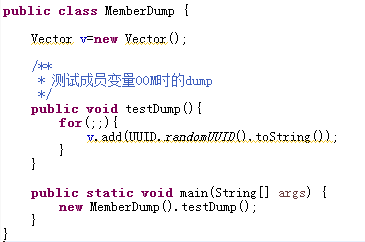
#### 变量类型和在Heap Dump中的体现

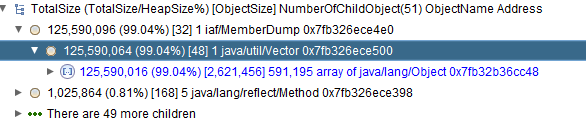
* 类变量（静态变量）



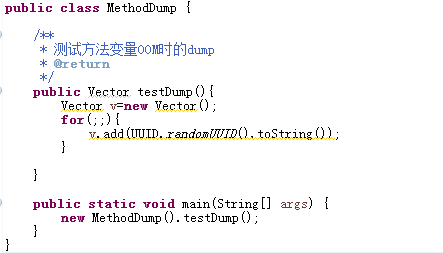


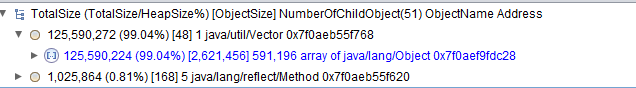
* 成员变量（实例变量）





* 局部变量（方法变量）





## 非堆内存（PermGen）溢出

### 诊断步骤

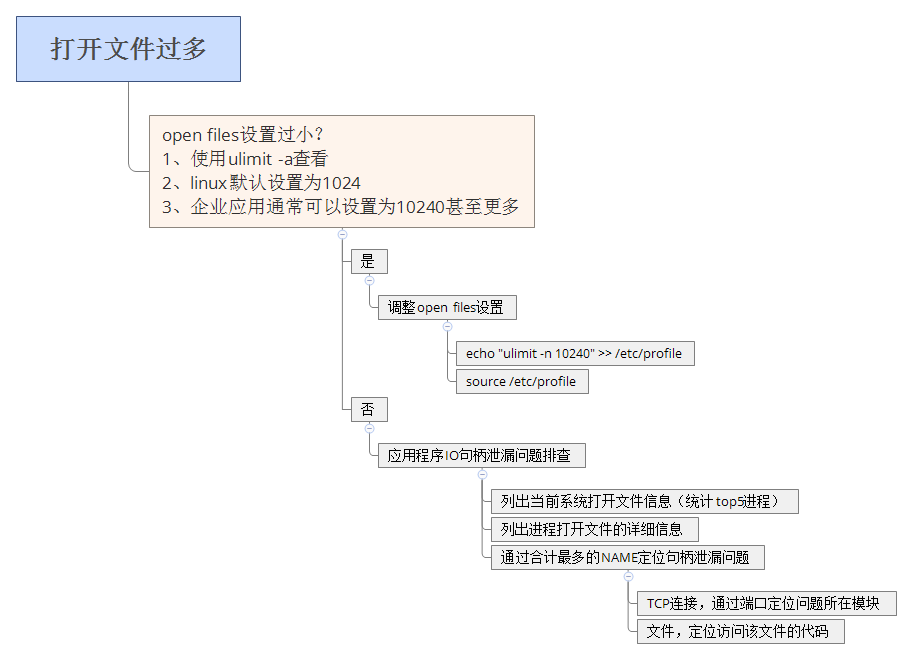
**java.lang.OutOfMemoryError: PermGen space**常见由JVM非堆内存（永久区）分配过小（32位JDK默认为64M、64位JDK默认为85M），可以通过在Java虚拟机启动参数中添加：

-XX:MaxPermSize=256m （设定为256M），来进行调整；

**注意：只针对JDK7以前，从JDK8开始Java已取消PermGen，改为Metaspace**

## IO句柄泄漏（Too Many Open Files）

### 诊断步骤



### 代码review重点

* 对于网络通讯，使用短连接交互时，是否在通讯完毕后正确关闭了连接以及IO流；
* 对于文件操作，是否在读写文件完毕后正确关闭了IO流。

# 中间件已知问题汇总

## Websphere

|  |  |  |
| --- | --- | --- |
| **已知问题** | **影响** | **解决措施** |
| 数据源j2c认证死锁  （线程挂起在  com.ibm.ejs.j2c.PoolManager$SubjectHashCode） | 7.0.0.23之前的版本 | 1.升级到7.0.0.23之后的版本  或者  2.在数据源添加定制属性  useRRASetEquals =true |
|  |  |  |

# 附录

## IBM JVM分析工具



运行该工具需要JDK7，通过java -jar ha456.jar或者java -jar jca457.jar运行

## UltraEditor14

http://yunpan.isscloud.com/s/IKV4cGRmTXQ 密码：nxe3 过期时间：2018-01-01

注：比较清爽的版本，最新版本AD太多