

如何收集顶头转储和线程请从CVP VXML服务器转存

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[问题](#)

[故障排除](#)

[获得堆转储的步骤](#)

[获得线程转储的步骤](#)

简介

本文描述如何从Cisco用户语音门户(CVP)语音扩展标记语言服务器(VXML)收集堆转储和线程转储Tomcat的。

先决条件

要求

Cisco 建议您了解以下主题：

- CVP
- CVP VXML服务器
- CVP VXML应用程序

使用的组件

本文档中的信息基于以下软件和硬件版本：

- CVP版本11.5
- CVP VXML服务器11.5

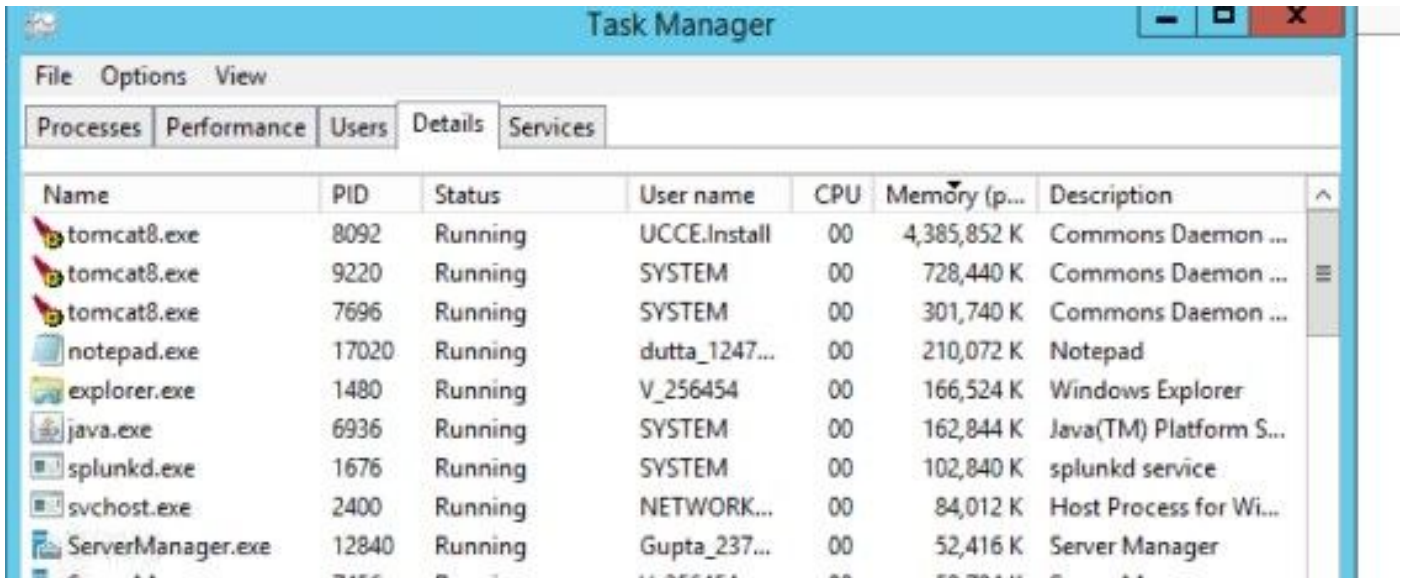
本文档中的信息都是基于特定实验室环境中的设备编写的。用于本文的所有设备从原始启动了。如果您的网络实际，请保证您了解所有命令的潜在影响。

问题

方案1。当您监控VXML服务器性能时您发现使用内存4GB。在失败前，您希望认识存储器泄漏的根本原因。

方案2。当您监控VXML服务器时，您发现VXML服务器使用高线程(1 500)，是相当异常的。您希望会分析线程使用方法和发现采取更多线程的线程状态和组件/应用程序。

例如：如镜像所显示，VXML Tomcat消耗总内存4.5GB。



在4.5 GB的VXML内存

故障排除

获得堆转储的步骤

步骤1. 下载JDK版本类似于在CVP上安装的版本

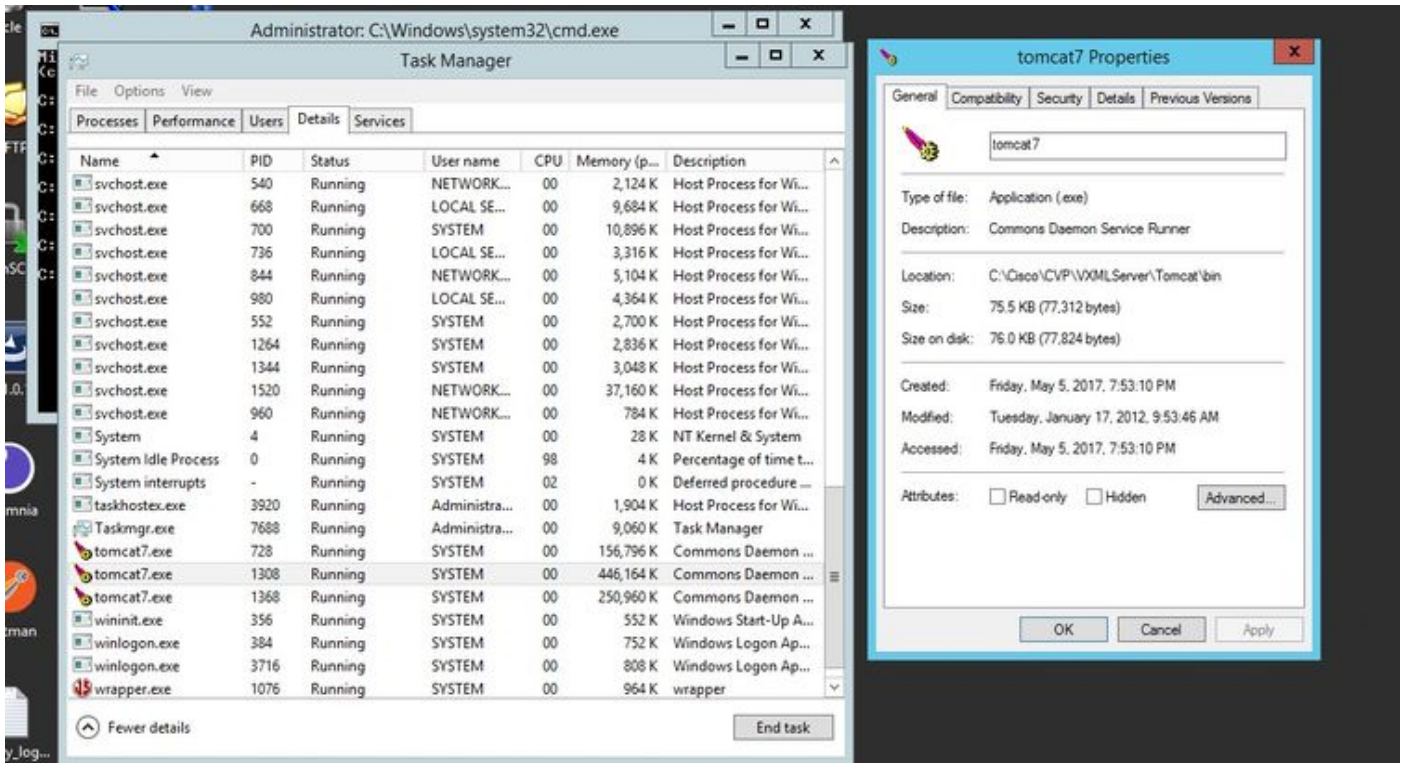
```
C:\Cisco\CVP\jre\bin>java -version
java version "1.7.0_51"
Java(TM) SE Runtime Environment (build 1.7.0_51-b13)
Java HotSpot(TM) Client VM (build 24.51-b03, mixed mode)
```

步骤2. 从桌面复制JDK到CVP服务器。

```
Download windows JDK exe
Open with 7-Zip
Dump contents into a directory %JDK-EXE%
cmd: cd %JDK-EXE%.rsrc\1033\JAVA_CAB10
cmd: extrac32 111
Now have a tools.zip in directory, open it in 7-Zip
Extract contents into a new directory %JDK-VERSION%
cmd: cd %JDK-VERSION%
cmd: for /r %x in (*.pack) do .\bin\unpack200 -r "%x" "%~dx%~px%~nx.jar"
```

Note:从<http://www.oracle.com/technetwork/java/javase/downloads/index.html>请下载JDK并且在本地文件夹上安装EXE并且从您的本地设备复制JDK到CVP。

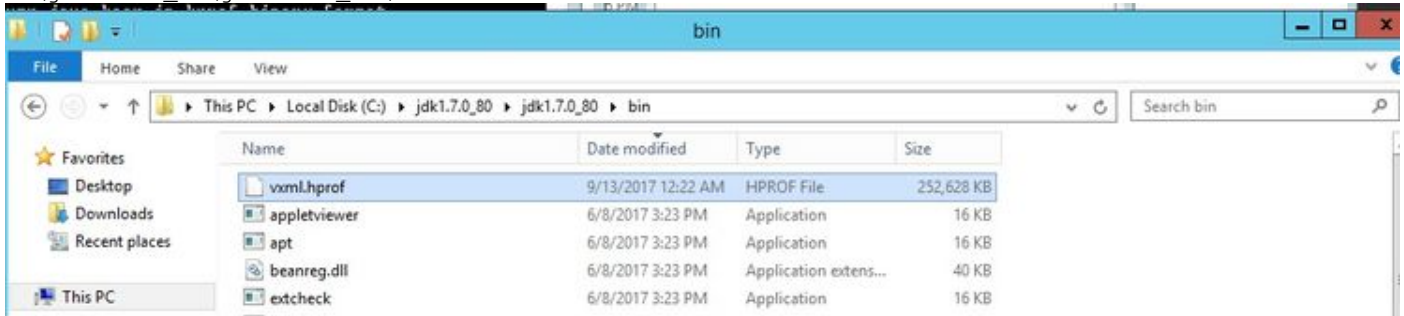
步骤3. 从任务管理器收集进程ID (PID) VXML服务器。



步骤4.执行此命令为了收集HeapDump。(前：jmap -转储：file=vxml.hprof TOMCAT实例<PID>。)

```
C:\jdk1.7.0_80\jdk1.7.0_80\bin>jmap -dump:file=vxml.hprof 1308
Dumping heap to C:\jdk1.7.0_80\jdk1.7.0_80\bin\vxml.hprof ...
Heap dump file created
```

```
C:\jdk1.7.0_80\jdk1.7.0_80\bin>
```



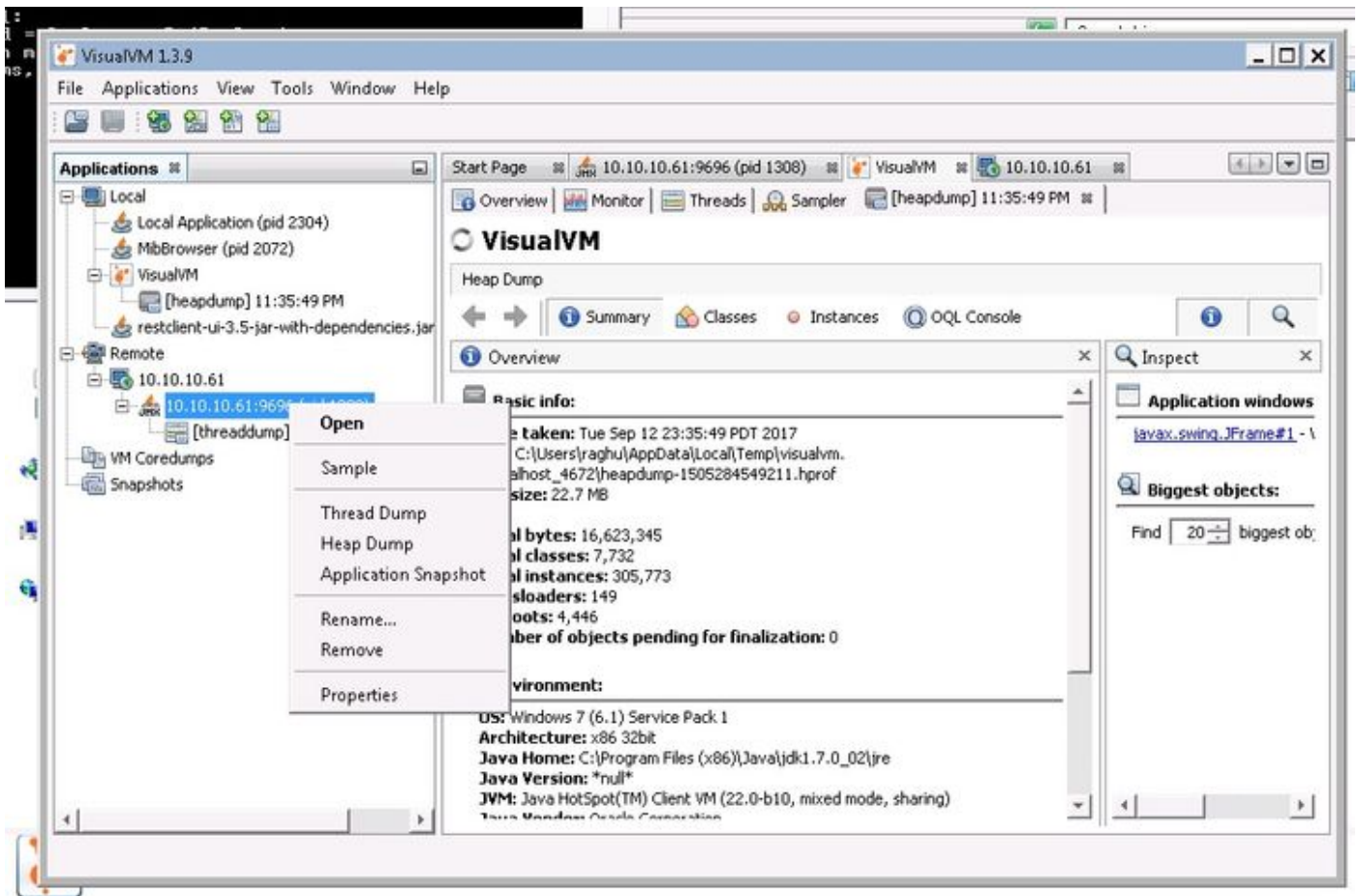
Note:您的hProf被创建，并且您能复制到本地系统和监控脱机。

获得线程转储的步骤

线程是相对容易比较的转储收藏堆积转储。

步骤1.连接到CVP VXML在<CVP VXML Server>的服务器：9696 (9696是VXML服务器的默认JMX端口)该使用jVisualVM。

步骤2.在JXM连接的Righ-clik和收集线程转储。



第3步。这里在远程服务器转存的线程转储，可以被保存作为文件和用于进一步分析。