

CVP VXMLサーバからヒープダンプとスレッドダンプを収集する方法

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[問題](#)

[トラブルシューティング](#)

[ヒープダンプの取得手順](#)

[スレッドダンプの取得手順](#)

概要

このドキュメントでは、Cisco Customer Voice Portal(CVP)Voice eXtensible Markup Language Server(VXML)からTomcatのヒープダンプおよびスレッドダンプを収集する方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- CVP
- CVP VXMLサーバ
- CVP VXMLアプリケーション

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- CVP バージョン 11.5
- CVP VXMLサーバ11.5

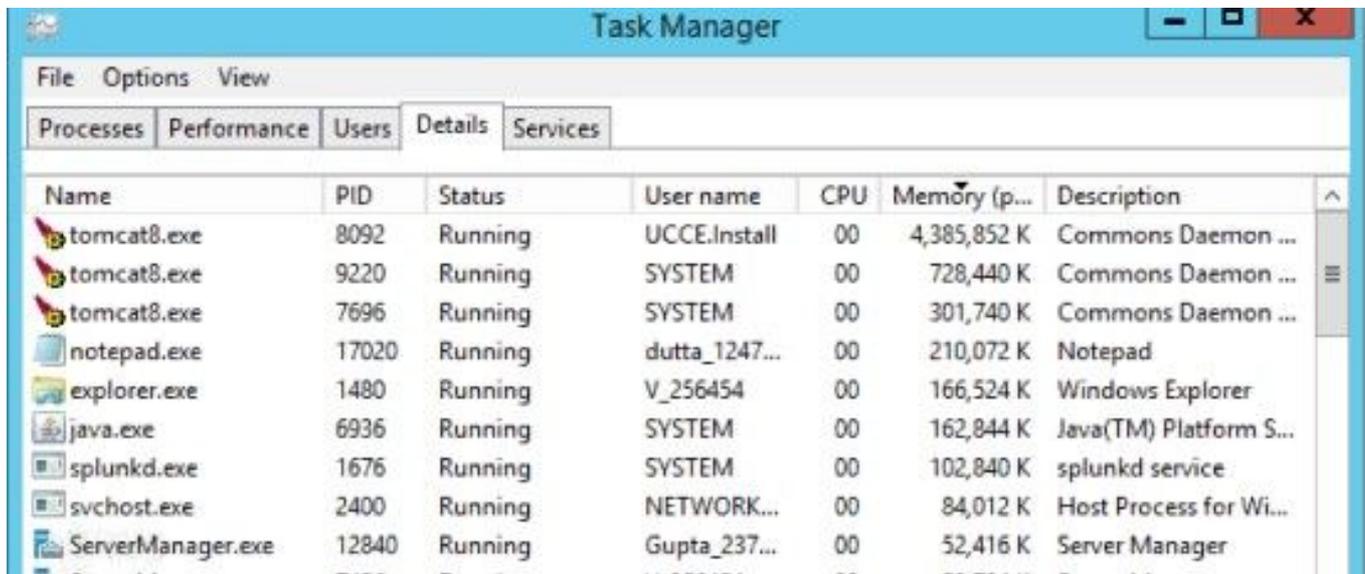
このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメント内で使用されているデバイスはすべて、クリアな設定（デフォルト）から作業を始めています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

問題

シナリオ1. VXMLサーバのパフォーマンスを監視している間に、4GBのメモリが使用されていることが判明しました。メモリリークの根本的な原因を、クラッシュする前に知りたいと思います。

シナリオ2. VXMLサーバをモニタしている間に、VXMLサーバが高いスレッド(~ 500)を使用していることがわかりました。これは非常に珍しいことです。スレッドの使用状況を分析し、スレッドの状態やコンポーネント/アプリケーションを調べる方法を知りたいと思います。

次に例を示します。図に示すように、VXML Tomcatは合計4.5 GBのメモリを消費します。



Name	PID	Status	User name	CPU	Memory (p...	Description
tomcat8.exe	8092	Running	UCCE.Install	00	4,385,852 K	Commons Daemon ...
tomcat8.exe	9220	Running	SYSTEM	00	728,440 K	Commons Daemon ...
tomcat8.exe	7696	Running	SYSTEM	00	301,740 K	Commons Daemon ...
notepad.exe	17020	Running	dutta_1247...	00	210,072 K	Notepad
explorer.exe	1480	Running	V_256454	00	166,524 K	Windows Explorer
java.exe	6936	Running	SYSTEM	00	162,844 K	Java(TM) Platform S...
splunkd.exe	1676	Running	SYSTEM	00	102,840 K	splunkd service
svchost.exe	2400	Running	NETWORK...	00	84,012 K	Host Process for Wi...
ServerManager.exe	12840	Running	Gupta_237...	00	52,416 K	Server Manager

4.5 GBのVXMLメモリ

トラブルシューティング

ヒープダンプの取得手順

ステップ1: CVPにインストールされているバージョンと同様のJDKバージョンをダウンロードします

```
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)
```

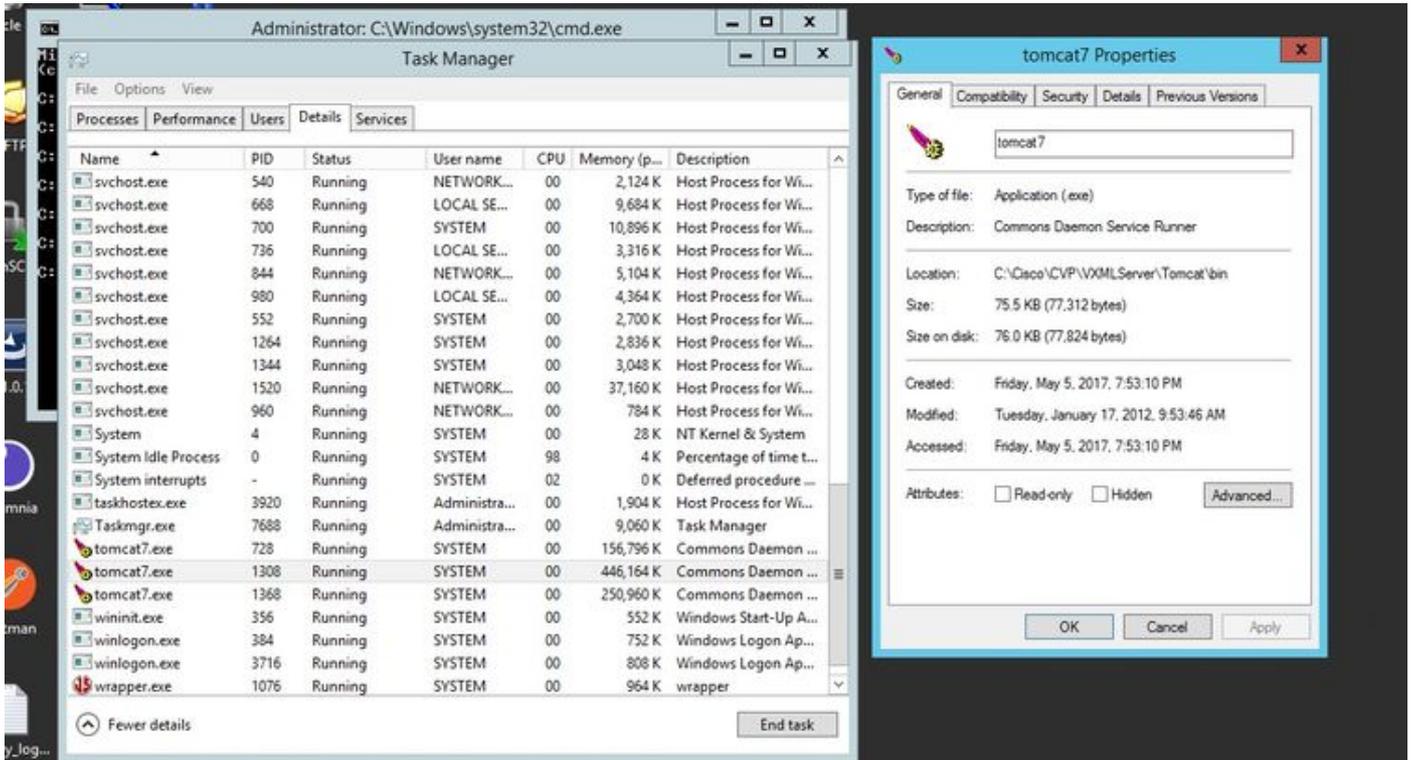
ステップ2 : デスクトップからCVPサーバにJDKをコピーします。

```
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"
```

注 : <http://www.oracle.com/technetwork/java/javaee/downloads/index.html>からJDKをダウン

ロードし、ローカルフォルダにEXEをインストールし、ローカルマシンからCVPにJDKをコピーするだけです。

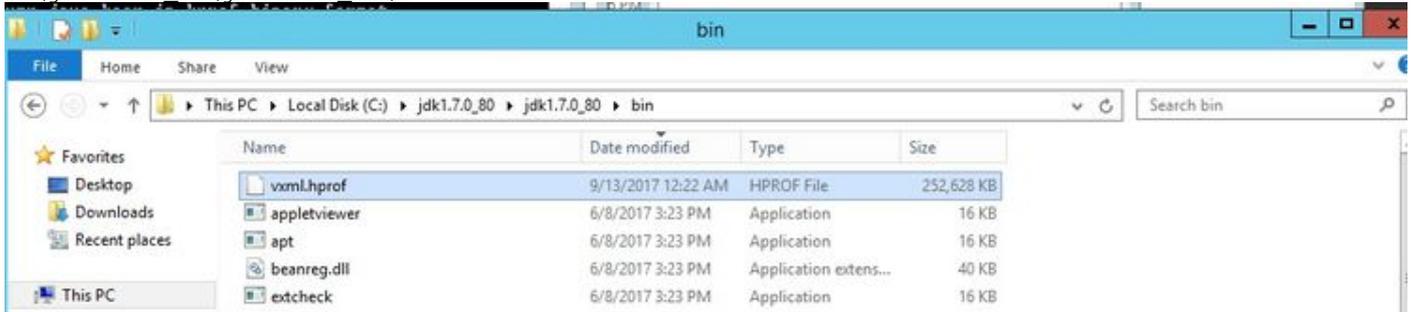
ステップ3：タスクマネージャからVXMLサーバのプロセスID(PID)を収集します。



ステップ4:HeapDumpを収集するには、このコマンドを実行します。(例：jmap -dump:file=vxml.hprof <PID OF TOMCAT Instance)。

```
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>
```



注:hProfが作成され、ローカルシステムにコピーしてオフラインでモニタできます。

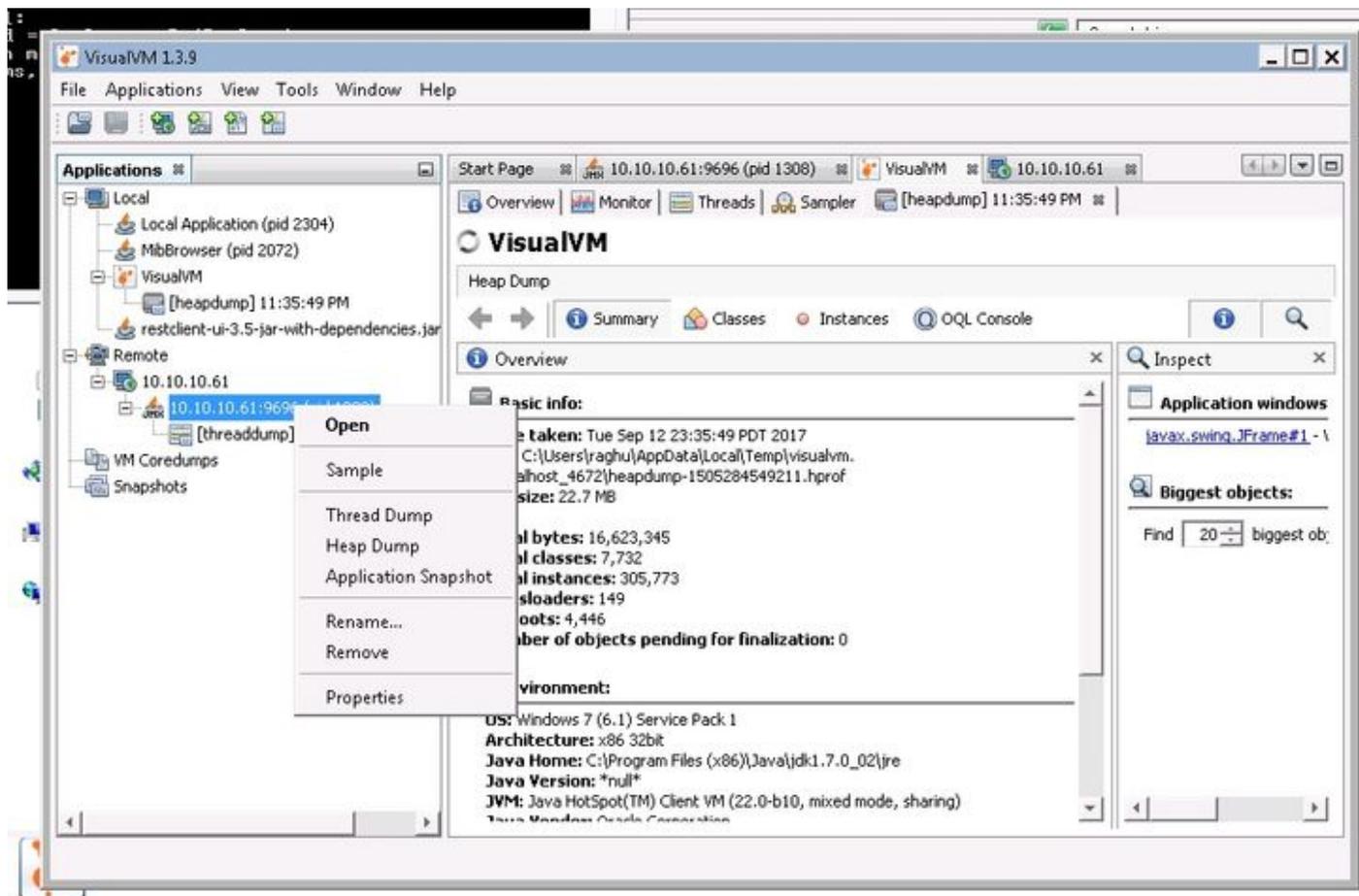
スレッドダンプの取得手順

スレッドダンプの収集は、ヒープダンプと比較して比較的簡単です。

ステップ1:jVisualVMを使用する<CVP VXML Server>:9696 (9696はVXML Serverのデフォルトの

JMXポート)。

ステップ2:JMX接続でRigh-cklを実行し、スレッドダンプを収集します。



ステップ3 : ここでは、リモートサーバにスレッドダンプをダンプし、ファイルとして保存して詳細な分析に使用できます。