

Comment collecter le vidage de segment et de thread à partir du serveur VXML CVP

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Problème](#)

[Dépannage](#)

[Étapes pour obtenir le vidage de tas](#)

[Étapes pour obtenir le vidage de thread](#)

Introduction

Ce document décrit comment collecter le vidage Heap et Thread pour Tomcat à partir du serveur VXML (Voice eXtensible Markup Language Server) CVP (Customer Voice Portal) de Cisco.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- CVP
- Serveur VXML CVP
- Applications VXML CVP

Components Used

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- CVP version 11.5
- Serveur VXML CVP 11.5

The information in this document was created from the devices in a specific lab environment. Tous les dispositifs utilisés dans ce document ont démarré par une configuration effacée (par défaut). Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

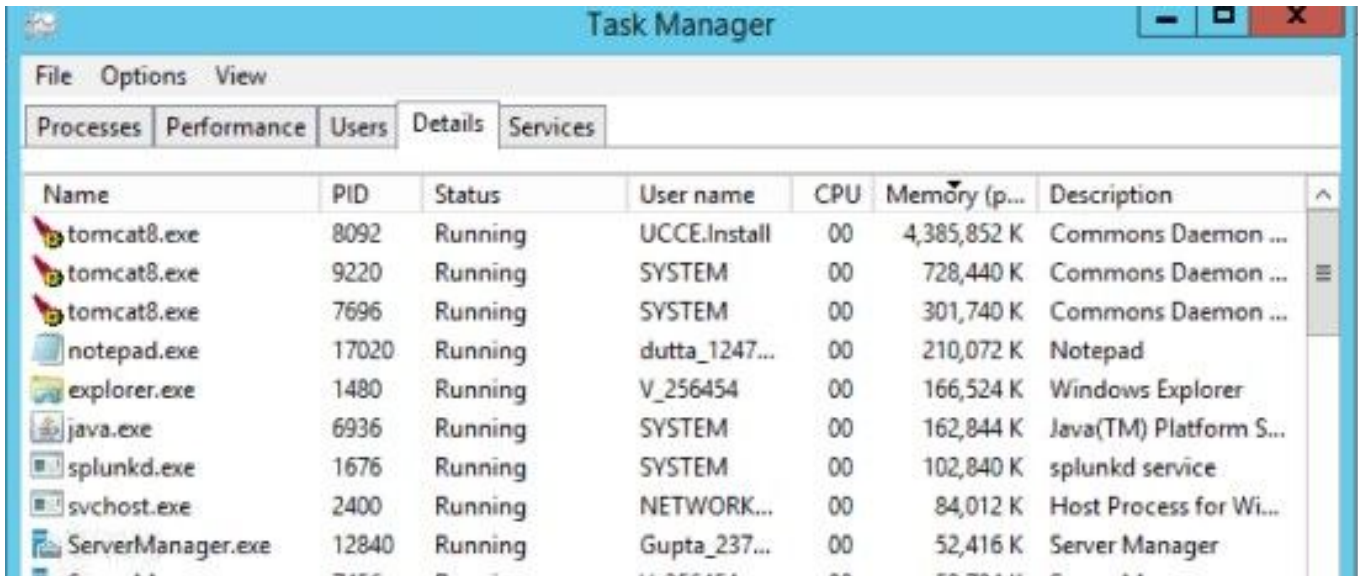
Problème

Scénario 1. Pendant que vous surveillez les performances du serveur VXML, vous avez découvert

qu'il utilise 4 Go de mémoire. Vous souhaitez connaître la cause première de la fuite de mémoire avant qu'elle ne s'écrase.

Scénario 2. Pendant que vous surveillez VXML Server, vous avez découvert que le serveur VXML utilise des threads élevés (~ 500), ce qui est assez inhabituel. Vous aimeriez savoir comment analyser l'utilisation du thread et connaître l'état du thread et le composant/application qui prend plus de threads.

Par exemple : VXML Tomcat consomme 4,5 Go de mémoire totale, comme l'illustre l'image.



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

Mémoire VXML à 4,5 Go

Dépannage

Étapes pour obtenir le vidage de tas

Étape 1. Télécharger la version JDK similaire à la version installée dans CVP

```
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)
```

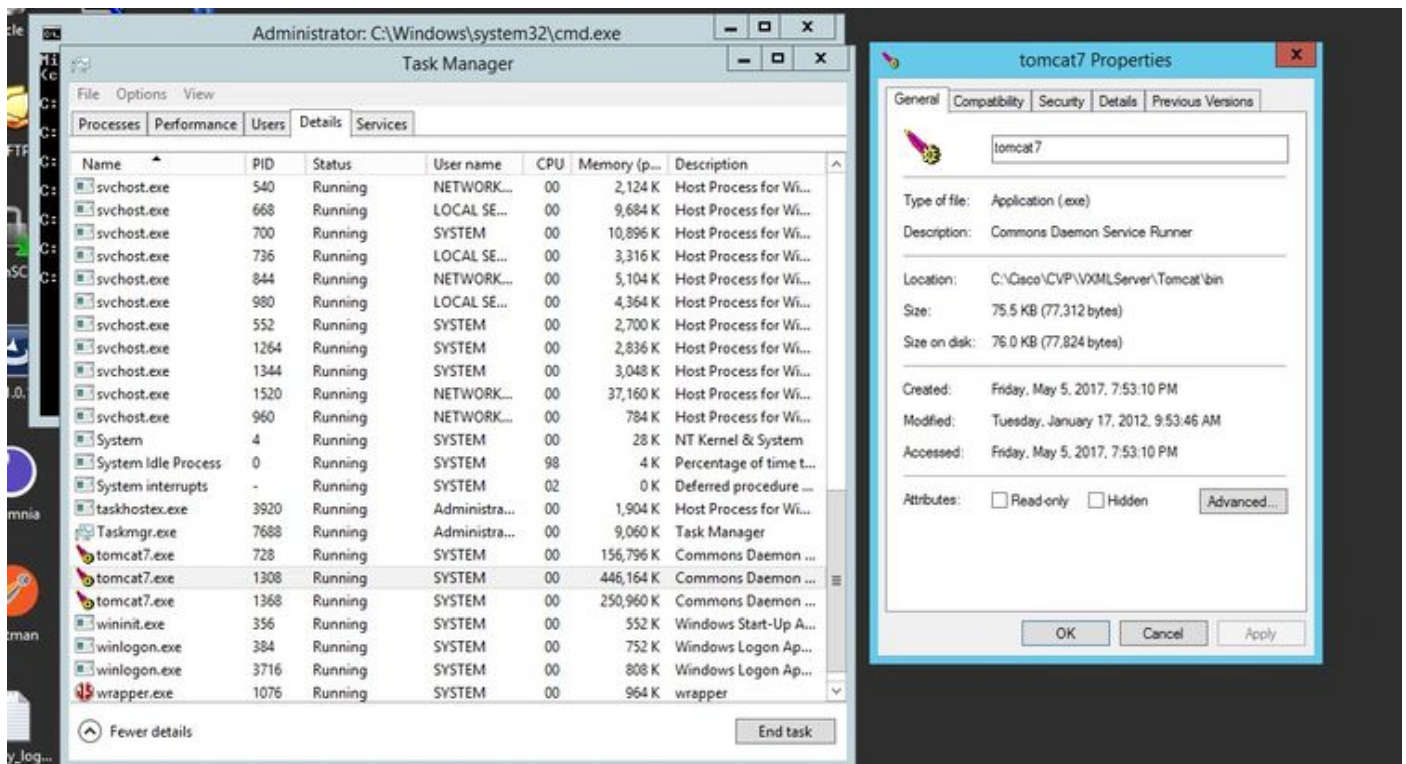
Étape 2. Copiez JDK du bureau vers le serveur CVP.

```
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: Il vous suffit de télécharger JDK à partir de <http://www.oracle.com/technetwork/java/javaee/downloads/index.html> et d'installer EXE sur

le dossier local et de copier le JDK de votre machine locale vers CVP .

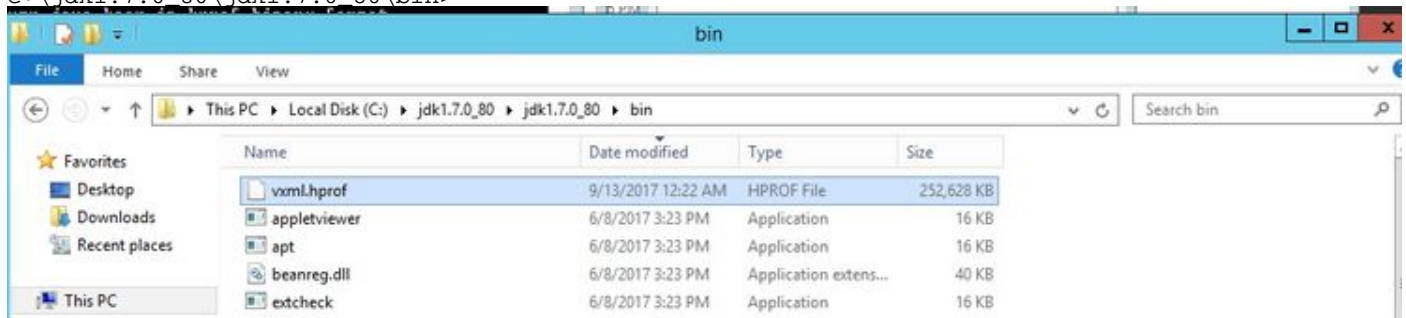
Étape 3. Collectez l'ID de processus (PID) du serveur VXML à partir du **Gestionnaire des tâches**.



Étape 4. Exécutez cette commande afin de collecter HeapDump. (Ex : 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>
```



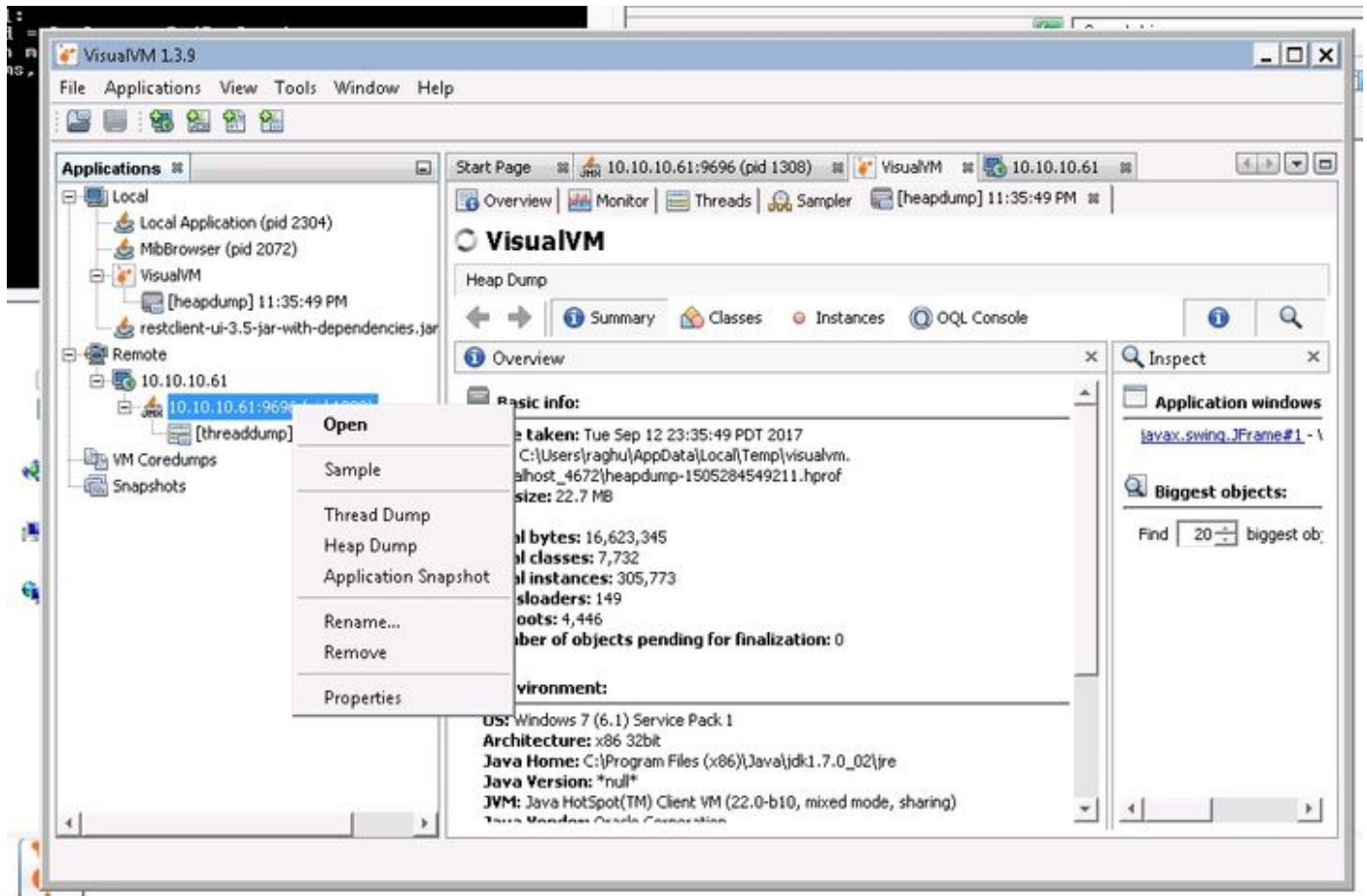
Remarque : Votre Prof est créé et vous pouvez le copier sur le système local et le surveiller hors connexion.

Étapes pour obtenir le vidage de thread

La collecte des décharges de thread est relativement facile par rapport à la collecte des décharges de tas.

Étape 1. Connectez-vous à CVP VXML Server sur <CVP VXML Server> : **9696** (9696 est le port JMX par défaut pour VXML Server) qui utilise jVisualVM.

Étape 2. Faites un clic droit sur **Connexion JMX** et **Collectez le vidage de thread**.



Étape 3. Ici, le vidage de thread est vidé sur le serveur distant, où il peut être enregistré en tant que fichier et utilisé pour une analyse plus approfondie.