

El sistema es tan lento hoy... pero ¿qué está haciendo?



Cómo supervisar QueueMetrics a través de JMX.

A veces nos encontramos con problemas de rendimiento en los sistemas que operamos: no siempre es fácil entender lo que realmente está pasando. Una aplicación se siente lenta.

Pero, ¿por qué? Muchas veces, la resolución se basa en el ensayo y error; si tenemos suerte, el problema se soluciona. Por supuesto, esta no es la mejor manera de trabajar. Se oye decir que hay que dar más memoria al sistema, o un disco más rápido, o más CPU, y a veces funciona.

Pero sin duda podemos hacerlo mejor.

Una de las cosas más agradables del entorno Java, el que se encuentra detrás de QueueMetrics y WombatDialer, es que ofrece una visión muy precisa y detallada de lo que realmente está sucediendo. No es necesario utilizar una compilación especial del software destinado a recopilar estadísticas, y no hay que pagar un alto precio de rendimiento para entender lo que el sistema está haciendo, por lo que es perfectamente aceptable recopilar estadísticas en vivo en un entorno de producción.

Incluso mejor, las JVMs están construidas para ser monitorizadas, de modo que hay un ecosistema estandarizado de herramientas para ayudarte a entender lo que está pasando realmente. Este enfoque se llama JMX.

El inconveniente -como puedes imaginar, siempre hay un inconveniente- es que la configuración de JMX no es tan sencilla. Especialmente si necesitas monitorizar un sistema desde una máquina diferente, por ejemplo, un servidor de producción desde tu estación de trabajo, la configuración te hará tirarte de los pelos antes de entender todas las opciones que necesitas para establecer una conexión que funcione.

Para facilitarte la vida, hemos hecho posible la configuración de una conexión JMX completamente funcional con sólo deshacer el comentario de una sola línea, y crear un túnel SSH que nos permitirá acceder a los datos de forma segura.

Necesitarás un RPM de Tomcat reciente (8.5.64 o más reciente).

Así, una vez instalado, se edita `/etc/sysconfig/qm-tomcat6`, y donde dice:

```
# Remove the comments below to activate JMX monitoring  
  
# JAVA_RMI_ADDRESS=localhost  
  
JAVA_JMX_PORT="30000"
```

Se deshace el comentario de la línea con `JAVA_RMI_ADDRESS`; guardar el archivo y reiniciar. Sólo tienes que hacer esto una vez, y esto se puede hacer con seguridad incluso en los sistemas de producción.

Puedes cambiar el puerto JMX si lo necesitas - no importa el puerto que elijas, siempre que sea el mismo en tu PC y en el servidor.

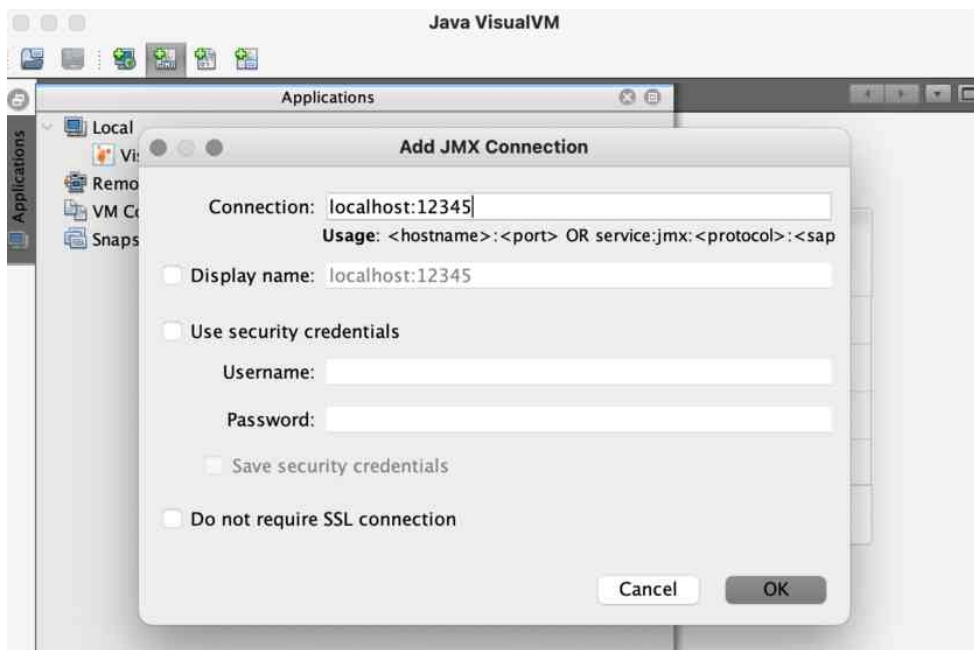
Ahora, vas a tu estación de trabajo e inicias una nueva sesión SSH con el siguiente encantamiento, para crear un túnel local para el puerto 30000 y el siguiente también:

```
ssh root@my.queuemetrics.server \  
-L 30000:localhost:30000 \  
-L 30001:localhost:30001
```

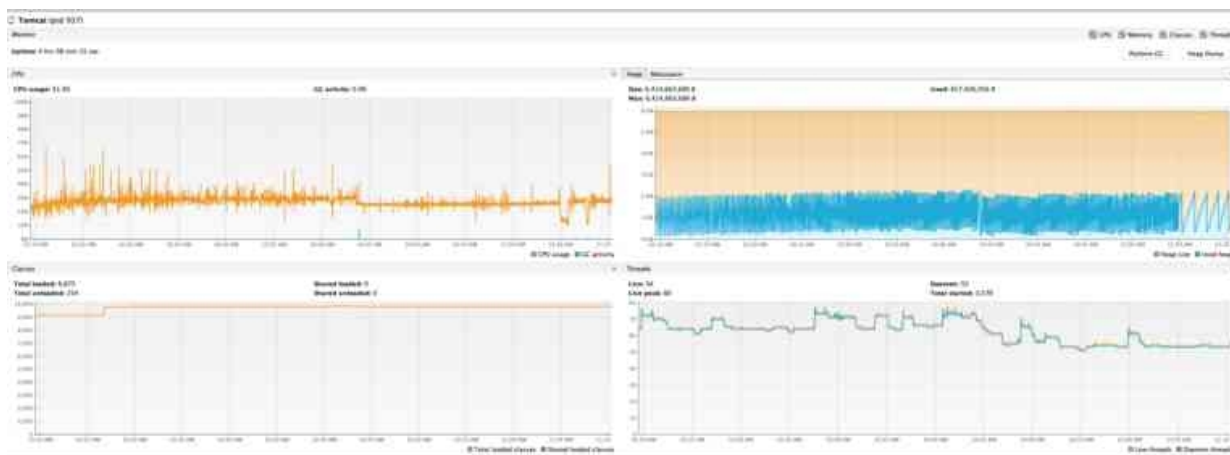
Ahora puedes ejecutar VisualVM en su estación de trabajo; lo más probable es que ya esté instalado si tienes un SDK de Java; si no, puedes descargarlo desde: <https://visualvm.github.io>

Ten en cuenta que mientras el ejemplo anterior se conecta a través de SSH como 'root', cualquier usuario servirá.

Cuando lances VisualVM, deberás hacer click en el botón 'Añadir conexión JMX', y crear una nueva conexión a 'localhost:30000', sin contraseña y sin encriptación (ya que usamos SSH para la seguridad y la encriptación).



Cuando hayas terminado, haz click en la conexión recién creada para iniciar la supervisión.



Por ejemplo, en los gráficos de arriba, podemos ver que:

- El uso de la CPU está en torno al 30%, y es estable (gráfico de la izquierda). Muy pocas recolecciones de basura, que suelen ser las principales culpables de los problemas de rendimiento de Java.
- No tenemos presión de memoria. El sistema ronda los 2G de uso (gráfico de la derecha)
- El sistema estuvo un poco inactivo en los últimos minutos - no asignó mucha memoria, pero está computando algo.
- Podemos saber lo que está haciendo obteniendo un volcado de hilos.

Se pueden hacer varias cosas interesantes con VisualVM:

- 'Conoce tu JVM': puedes ver la configuración de la JVM desde 'Visión general' / 'Argumentos de la JVM'.

- 'Monitoreo de la memoria': puedes ver el uso actual de la CPU, la memoria y los hilos desde la página 'Monitor'. Ten en cuenta que con la mayoría de las configuraciones, es normal que toda la memoria sea utilizada antes de que se realice una recolección de basura; por lo que esperarías ver picos y caídas en el gráfico. También puedes forzar una recolección de basura si quieres ver el uso "verdadero" de la memoria, pero esto puede ser imprudente en servidores de producción muy cargados.
- 'Monitoreo de hilos': puedes obtener un volcado de hilos textual como el comentado anteriormente seleccionando 'Hilos' / 'Volcado de hilos'.
- Puedes utilizar el 'Sampler' para adquirir un desglose de la memoria y el uso de la CPU por clase (primero necesitas instalar el plugin 'VisualVM-Sampler' desde el menú de Plugins, si no está ya disponible).
- Puedes mantener un servidor abierto con varias instancias de VisualVM para supervisar varios servidores QM.

¡Feliz seguimiento!

Referencias de QueueMetrics

El software QueueMetrics está disponible en las instalaciones o como un servicio alojado en la nube para FreePBX, Yeastar S PBX, Grandstream, Issabel, FusionPBX y muchas otras distribuciones de Asterisk.

Para más información técnica, consulte el [Manual del Usuario](#) .

Visite www.queuemetrics.com para una prueba gratuita de 30 días con todas las funciones.

Asiste a nuestros [Free Webinars](#) para una demostración en vivo de QueueMetrics.