

Error: Bonita server not responding after an inactivity period



unai.gaston.
camino

Thursday, June 11, 2020 - 13:52

Blog Categories: Troubleshooting (/blog-categories/troubleshooting)

Component: Engine, Tomcat server | **Version:** 7.9+

Problem

You have set up your Bonita so it connects to your **database server** that is located in a private network **protected by a firewall**.

Everything works fine: you are able to connect to the Bonita portal; you can check your cases, you can execute your tasks; your processes move forward normally.

But then, if your **platform remains inactive for some time** (~1 hour, for example), when you get back, the server is not responding any more.

In your `bonita.yyyy-mm-dd.log` file, you can see this type of error:

```
// ---> first DB transaction after inactivity period (e.g. triggered by Login API
2020-04-29 18:46:20.335 +0000 FINEST (ajp-nio-8009-exec-5) org.bonitasoft.engine.
2020-04-29 18:46:20.336 +0000 FINEST (ajp-nio-8009-exec-5) org.bonitasoft.engine.
...
...
// ---> XA transaction timeout error
2020-04-29 18:49:20.337 +0000 WARN (Transaction Reaper) com.arjuna.ats.arjuna ARJ
2020-04-29 18:49:20.339 +0000 WARN (Transaction Reaper Worker 0) com.arjuna.ats.a
2020-04-29 18:49:20.342 +0000 WARN (Transaction Reaper Worker 0) com.arjuna.ats.a
java.base@11.0.7 (mailto:java.base@11.0.7)/java.net (https://java.net).SocketInput
java.base@11.0.7/java.net.SocketInputStream.socketRead(SocketInputStream.java:115)
java.base@11.0.7/java.net.SocketInputStream.read(SocketInputStream.java:168)
java.base@11.0.7/java.net.SocketInputStream.read(SocketInputStream.java:140)
...
org.postgresql.jdbc.PgPreparedStatement.executeQuery(PgPreparedStatement.java:106)
...
com.sun.proxy.$Proxy20.executeQuery(Unknown Source)
org.apache.tomcat.dbcp.dbcp2.PoolableConnection.validate(PoolableConnection.java:
org.apache.tomcat.dbcp.dbcp2.PoolableConnectionFactory.validateConnection(Poolabl
...
org.apache.tomcat.dbcp.pool2.impl.GenericObjectPool.borrowObject(GenericObjectPoc
...
org.apache.tomcat.dbcp.dbcp2.managed.ManagedDataSource.getConnection(ManagedDataS
org.apache.tomcat.dbcp.dbcp2.BasicDataSource.getConnection(BasicDataSource.java:7
org.hibernate.engine.jdbc.connections.internal.DatasourceConnectionProviderImpl.g
...
org.bonitasoft.engine.platform.impl.PlatformServiceImpl.getDefaultTenant(Platform
...
org.bonitasoft.engine.transaction.JTATransactionServiceImpl.executeInTransaction(
org.bonitasoft.engine.api.impl.LoginAPIImpl.getTenant(LoginAPIImpl.java:147)
org.bonitasoft.engine.api.impl.LoginAPIImpl.loginInternal(LoginAPIImpl.java:117)
org.bonitasoft.engine.api.impl.LoginAPIImpl.loginInternal(LoginAPIImpl.java:106)
com.bonitasoft.engine.api.impl.LoginAPIExt.login(LoginAPIExt.java:36)
...
org.keycloak.adapters.saml.servlet.SamlFilter.doFilter(SamlFilter.java:174)
org.bonitasoft.console.common.server.auth.impl.saml.BonitaSAML2Filter.processRequ
org.bonitasoft.console.common.server.auth.impl.saml.BonitaSAML2Filter.doFilter(Bc
...
java.base@11.0.7 (mailto:java.base@11.0.7)/java.lang.Thread.run(Thread.java:834)
```

Why do you have this error?

The **validation query** executed before borrowing one of the connections in Tomcat's datasource's connection pool, is **hanging**.

As a result, the **XA transaction is timing out**. Note that In the above example, the transaction timeout has been set to 3 minutes (180 seconds) in the TOMCAT_HOME/server/conf/jbossts-properties.xml file:

```
<!-- (default is 60 seconds) -->

<entry key (http://www.php.net/key)="com.arjuna.ats.arjuna.coordinator
```

(You may check this article (<https://community.bonitasoft.com/blog/how-increase-default-xa-transaction-timeout>) for further information on the XA Transaction Timeout, even though raising its value will not help in the present case.)

What is happening?

Tomcat's datasource defines a minimum number of idle connections (`minIdle`) that must be kept in the pool at all time (where possible, that is, provided all `maxTotal` connections are not already active); for instance, in your `TOMCAT_HOME/setup/tomcat-templates/bonita.xml`:

```
Nombre del recurso = "bonitaDS"
    uniqueName = "jdbc / bonitaDSXA"
    auth = "Container"
    factory = "org.jboss.narayana.tomcat.jta.TransactionalDataSou
    transactionManager = "TransactionManager"
    type = "javax.sql.XADataSource"
    initialSize = " 8 "
    maxTotal = " 50 " minIdle = " 8 "
    maxIdle = " 16 "
    defaultAutoCommit = " false "
    validationQuery = " @@ ds1.database_test_query @@ "
    removeAbandonedOnBorrow = " true "
    removeAbandonedOnMaintenance ="true "
    logAbandoned = " true "
    testOnBorrow = " true "

    xaDataSource = "RawBonitaDS"
```

Esto significa que después de un período de inactividad, al menos las conexiones mínimas al servidor de la base de datos permanecerán abiertas.

Sin embargo, si el período de inactividad se extiende más allá de la *duración máxima de la sesión TCP* del servidor de seguridad de su base de datos (por ejemplo, 3.600 segundos = 1 hora), el servidor de seguridad cortará la conexión.

Ahora, cuando regrese a su servidor Bonita después de este período de inactividad, Tomcat intentará tomar prestada una de esas conexiones inactivas para evitar tener que establecer una nueva (ya que esta última puede ser relativamente costosa), por lo que comenzará iniciando el consulta de validación (definida en el parámetro `validationQuery`) a través de la conexión.

El problema proviene del hecho de que el grupo de conexiones de Tomcat no parece hacer frente a este evento inesperado :

- no es consciente de que la conexión se ha caído por su igual, por lo que **la consulta de validación se bloquea**

- el tiempo de espera de la consulta de validación (`validationQueryTimeout`, si fuera a definirse) no tiene poder: no logra abortar la consulta de validación suspendida

Este es un problema conocido común a algunos grupos de conexiones, como explica el siguiente artículo (<https://github.com/brettwooldridge/HikariCP/wiki/Bad-Behavior:-Handling-Database-Down>) :

"(...) Varios grupos diferentes proporcionan tales mecanismos de tiempo de espera, pero obtener tiempos de espera correctos en condiciones de red adversas es más difícil de lo que parece.

"(...)

" Dbcp2 no devolvió una conexión y tampoco expiró. La ejecución de la consulta de validación está bloqueada debido al tráfico TCP no reconocido. Posteriormente, la instrucción SQL ejecutada en la conexión (incorrecta) por el arnés de prueba se bloquea (debido a TCP no reconocido). `setMaxWait` (5000) es aparentemente inútil para manejar las interrupciones de la red. No hay otras configuraciones de tiempo de espera significativas que se apliquen a una falla de red.

"(...)

" Dbcp2: (D) - Aparentemente se cuelga hasta el límite de tiempo de espera TCP del sistema operativo.

(...) "

Puede leer más sobre esta *falla* en los siguientes artículos, por ejemplo: aquí (<https://blog.cloudflare.com/when-tcp-sockets-refuse-to-die/>) y allá (<https://stackoverflow.com/questions/13085676/tcp-socket-no-connection-timeout>)

Solución

Básicamente, hay dos formas de superar este error:

- Opción A (**desaconsejada**): aumentando el valor de la *duración máxima de la sesión TCP* (o similar) del firewall . Pero, de nuevo, puede ser difícil estimar cuál podría ser el período de inactividad más largo de su servidor Bonita, por lo que será difícil saber qué valor sería suficiente. Además, puede haber otras aplicaciones además del servidor Bonita que acceden al mismo servidor de base de datos que inevitablemente se verán afectadas por esta opción.
- Opción B (**recomendada**): ajustando los parámetros del grupo de conexiones de Tomcat para que todas las conexiones inactivas se renueven de vez en cuando.

De acuerdo con la Opción B, estos son los parámetros que necesitaría establecer en las fuentes de datos de Tomcat en su `TOMCAT_HOME / setup / tomcat-templates / bonita.xml`:

(siempre que inicie su servidor con el script `start-bonita.sh/.bat`; de lo contrario, necesitaría actualizar directamente el archivo `TOMCAT_HOME / conf / Catalina / localhost / bonita.xml`):

```
testWhileIdle = "false" (valor predeterminado)
timeBetweenEvictionRunsMillis = "" minEvictableIdleTimeMillis = "
"
softMinEvictableIdleTimeMillis = " - 1 "(valor por defecto)
```

El objetivo es garantizar que todas las conexiones inactivas se renueven después de un tiempo, incluso si eso significa ir por debajo del valor mínimo. Por lo tanto, este es el razonamiento detrás de esos valores:

- `testWhileIdle = "false"` ---> por lo que la decisión de desalojar o no la conexión inactiva no depende del resultado de la consulta de validación
- `softMinEvictableIdleTimeMillis = "- 1"` ---> para que el desalojo de la conexión inactiva no esté condicionado por el valor de `minIdle`, y todas las conexiones pueden cerrarse independientemente (de todos modos, el grupo de conexiones creará automáticamente tantas conexiones nuevas como sea necesario para otorgar el valor mínimo)
- `timeBetweenEvictionRunsMillis` y `minEvictableIdleTimeMillis > 0` ---> por lo que se está ejecutando un subproceso desalojador

NOTA: tenga cuidado al elegir los valores para estos dos parámetros, ya que definirán el tiempo que una conexión inactiva permanecerá abierta; no desea que ese tiempo sea demasiado corto para que las conexiones inactivas permanezcan abiertas durante algún tiempo, ya que tener que crear nuevas conexiones con demasiada frecuencia puede tener un ligero impacto en el rendimiento de Bonita (establecer una nueva conexión puede ser relativamente costoso).

Para obtener más información sobre los parámetros de configuración del origen de datos de Tomcat, consulte la documentación de DBCP de Apache Commons (<https://commons.apache.org/proper/commons-dbcp/configuration.html>).
303 lecturas

Añadir nuevo comentario

Tu nombre antonio.gonzalez

Comentario *

Fuente HTML	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	
Formato	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	

Cambiar al editor de texto sin formato



I'm not a robot

reCAPTCHA
Privacy - Terms



Salvar

Popular

- Una forma efectiva de combatir las bibliotecas duplicadas y las clases conflictivas de versiones con la herramienta Memory Analyzer Tool (/effective-way-fight-duplicated-libs-and-version-conflicting-classes-using-memory-analyzer-tool) : 25.822 lecturas
- Adquirir volcado de pila de MAT (herramienta de analizador de memoria) (/acquire-heap-dump-mat-memory-analyzer-tool) : 24.895 lecturas
- Cómo agregar Javascript personalizado en sus formularios Bonita BPM (/how-add-custom-javascript-your-bonita-bpm-forms) - 23,222 lecturas
- Ejemplos y videos tutoriales para Bonita BPM 7 (/examples-and-video-tutorials-bonita-bpm-7) - 16.616 lecturas
- Cómo usar Bonita BPM 6 Web REST API en Python: crear cuentas (/how-use-bonita-bpm-6-web-rest-api-python-create-accounts) - 15,582 lecturas
- Una primera muestra de AngularJS (/first-taste-angularjs) : 14.095 lecturas
- Cómo crear y personalizar elementos de límite en Eclipse GMF (/how-create-and-customize-boundary-items-eclipse-gmf) - 13,073 lecturas
- El juego "Bonita Escape" (/bonita-escape-game) - 12,705 lecturas
- CommitStrip: cuando tengo que trabajar con sistemas heredados (/commitstrip-when-i-have-work-legacy-systems) : 11.542 lecturas
- ¡Lo habéis conseguido! - ¡Tu lo tienes! (/C2%A1lo-habC3%A9is-conseguido-%E2%80%93-youve-got-it) - 8,593 lecturas

¡Obtenga actualizaciones de la comunidad mundial de Bonita!

Subscribe

(/)

Bonitasoft ayuda a las empresas innovadoras de todo el mundo a reinventar, automatizar y obtener una visibilidad profunda de sus procesos comerciales. La plataforma de automatización de procesos digitales Bonitasoft de Bonitasoft permite la colaboración entre desarrolladores profesionales y ciudadanos para entregar rápidamente proyectos y aplicaciones de automatización utilizando las mejores metodologías y herramientas DevOps.

Con un ecosistema de más de 150,000 miembros y clientes en más de 75 países, Bonitasoft ofrece la plataforma de código abierto y código bajo más utilizada para la automatización de procesos comerciales.

(<https://twitter.com/bonitasoft>)

(<https://www.facebook.com/bonitasoftbpm>)

(<https://www.linkedin.com/company/bonitasoft/>)

([/blog.xml](#))

(<https://www.youtube.com/user/BonitaSoft>)

© Bonitasoft Community tiene licencia bajo una licencia internacional CC by-sa 4.0.