

Re: Commons DBCP pooling problem

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.jdbcdriver/2005-11/msg00009.html>

- *From:* Joe Weinstein <joeNOSPAM@xxxxxxx>
 - *Date:* Fri, 04 Nov 2005 09:53:04 -0800
-

Joe Weinstein wrote:

downlode@xxxxxxxxxxx wrote:

Hello,
I am using Commons DBCP to pool my sqlserver database connections.
Sometimes there is just an hourglass cursor when trying to access my
application - looks like there are no free connections available.

Am I doing anything wrong in my code? Should I be closing the
datasource, for example? Please see below.

Hi. The JSP code is excellent. Very safe and sure to close
the connection.

Actually, the one extra thing I might try, is to make the
connection and it's subobjects method-level objects. If the
JSP is typically one instantiated class, any multiple threads
executing it could overwrite the connection reference and
leak connections instead of having them all closed.
Joe

Re: Commons DBCP pooling problem

I am not knowledgeable about that pool implementation. I would suggest getting a full thread dump of the JVM during that hang, and see what is waiting on what. Can you find out how many JSPs are running concurrently when such a hang is happening? If the pool implementation ever calls DriverManager calls, it may be the cause. DriverManager calls are all class-synchronized, and lots of JDBC drivers make those calls under the covers, so any one long-running DriverManager call (getConnection()) can halt all other JDBC in the whole JVM...

Joe Weinstein at BEA Systems

Class for retrieving datasource:

```
package mypackage.name.here

import javax.sql.*;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.*;
import org.apache.commons.dbcp.*;
import org.apache.commons.pool.*;
import org.apache.commons.pool.impl.*;

public class DatabaseUtil {

    static PoolingDataSource dataSource = null;
    public static final String DB_PROPERTIES =
"myextension.properties";

    // Returns a DataSource that implements connection pooling
    public static synchronized DataSource getDataSource(String
configDir) throws Exception {

        if (dataSource == null){

            Properties p = new Properties();
```

Re: Commons DBCP pooling problem

```
try{
    File cfg = new File(configDir, DB_PROPERTIES);
    p.load(new FileInputStream(cfg));
} catch (IOException e) {
    System.out.println(e);
}

String type = p.getProperty("dbType");
String url = p.getProperty("url");
String ip = p.getProperty("dbIP");
String name = p.getProperty("dbName");
String port = p.getProperty("dbPort");
String sid = p.getProperty("dbSid");
String fullUrl = null;
if (type.trim().equals("ms")){
    // MSSQL
    fullUrl = url+ip+"/"+name;
} else {
    // ORACLE
    fullUrl = url+ip+": "+port+": "+sid;
}

BasicDataSource ds = new BasicDataSource();
ds.setUrl(fullUrl);
ds.setDriverClassName(p.getProperty("driver"));
ds.setUsername(p.getProperty("usr"));
ds.setPassword(EncDoc.decode(p.getProperty("pass")));

ds.setMaxActive(100);

// Create a PoolableDataSource
ObjectPool connectionPool = new GenericObjectPool(null);
ConnectionFactory connectionFactory = new
DataSourceConnectionFactory(ds);
PoolableObjectFactory poolableConnectionFactory = new
PoolableConnectionFactory(connectionFactory, connectionPool, null,
null, false, true);
dataSource = new PoolingDataSource(connectionPool);
}

return dataSource;
}
```

Re: Commons DBCP pooling problem

Sample JSP page:

```
...
DataSource dataSource = DatabaseUtil.getDataSource(configDir);
Connection myConnect = null;
Statement myStatement = null;
ResultSet myResults = null;

try{
myConnect = dataSource.getConnection();
myStatement = myConnect.createStatement();
...
myQuery = "select count (*) as 'total' from admin where courseID =
'"+courseID+"'";

myStatement.executeQuery(myQuery);
myResults = myStatement.getResultSet();
while (myResults.next()) count = myResults.getInt("total");

...

}catch (SQLException e) {
    out.println("SQL Error: <br><br>"+e);
} finally {
    try { myResults.close(); } catch(Exception e) {}
    try { myStatement.close(); } catch(Exception e) { }
    try { myConnect.close(); } catch(Exception e) { }
}
```

Thanks all.

Re: Commons DBCP pooling problem