

Metacat - Bug #3416

ArrayIndexOutOfBoundsException in DBConnectionPool

06/25/2008 11:15 AM - Duane Costa

Status:	New	Start date:	06/25/2008
Priority:	Normal	Due date:	
Assignee:	Jing Tao	% Done:	0%
Category:	metacat	Estimated time:	0.00 hour
Target version:	Unspecified	Spent time:	0.00 hour
Bugzilla-Id:	3416		

Description

The following error has appeared in the Tomcat log files at LTER several times:

```
Jun 25, 2008 9:55:54 AM org.apache.catalina.core.StandardWrapperValve invoke
SEVERE: Servlet.service() for servlet metacat threw exception
java.lang.ArrayIndexOutOfBoundsException: 9 >= 9
at java.util.Vector.elementAt(Vector.java:432)
at edu.ucsb.nceas.metacat.DBConnectionPool.printMethodNameHavingBusyDBConnection(DBConnectionPool.java:617)
at edu.ucsb.nceas.metacat.MetaCatServlet.handleGetOrPost(MetaCatServlet.java:481)
at edu.ucsb.nceas.metacat.MetaCatServlet.doPost(MetaCatServlet.java:312)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:710)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:269)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:188)
at org.vfny.geoserver.filters.SetCharacterEncodingFilter.doFilter(SetCharacterEncodingFilter.java:122)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:215)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:188)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:210)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:174)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:127)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:117)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:108)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:151)
at org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:870)
at org.apache.coyote.http11.Http11BaseProtocol$Http11ConnectionHandler.processConnection(Http11BaseProtocol.java:665)
at org.apache.tomcat.util.net.PoolTcpEndpoint.processSocket(PoolTcpEndpoint.java:528)
at org.apache.tomcat.util.net.LeaderFollowerWorkerThread.runIt(LeaderFollowerWorkerThread.java:81)
at org.apache.tomcat.util.threads.ThreadPool$ControlRunnable.run(ThreadPool.java:685)
at java.lang.Thread.run(Thread.java:595)
```

The relevant source code in DBConnectionPool.java is:

```
/**
 * Method to print out the method name which have busy DBconnection
 */
public void printMethodNameHavingBusyDBConnection()
{
DBConnection db = null; //single DBConnection
    int poolSize = 0; //size of connection pool
    //get the size of DBConnection pool
    poolSize = connectionPool.size();
    //Check every DBConnection in the pool
    for ( int i=0; i<poolSize; i++)
    {
db = (DBConnection) connectionPool.elementAt(i);
        //check the status of db. If it is free, count it
        if (db.getStatus() == BUSY)
        {
            logMetacat.warn("This method having a busy DBConnection: "
                +db.getCheckOutMethodName());
        }
    }
}
```

```
logMetacat.warn("The busy DBConnection tag is: "  
                +db.getTag());  
    }//if  
}//for
```

```
}//printMethodNameHavingBusyDBConnection
```

It looks like this could be a thread safety issue. Perhaps the poolSize changes between the time that it is assigned:

```
poolSize = connectionPool.size();
```

and the time that the Vector is accessed:

```
db = (DBConnection) connectionPool.elementAt(i);
```

If this is the case, then it seems that a synchronized() block might be needed.

Thanks,
Duane

History

#1 - 03/27/2013 02:23 PM - Redmine Admin

Original Bugzilla ID was 3416