bytea column, getting "ERROR: invalid memory alloc request size" when loading data

Last updated by | Francisco Javier Pardillo Martin | Feb 10, 2022 at 3:38 AM PST

Contents

Issue

Getting "**ERROR: invalid memory alloc request size**" when trying to upload data (less than 1Gb) in a table with **bytea** column type.

Customer example output from a Java application:

```
Exception in thread 'main' org.postgresql.util.PSQLException: ERROR: invalid memory alloc request size 1677722
        at org.postgresql.core.v3.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2675)
        at org.postgresql.core.v3.QueryExecutorImpl.processResults(QueryExecutorImpl.java:2365)
        at org.postgresql.core.v3.QueryExecutorImpl.execute(QueryExecutorImpl.java:355)
        at org.postgresql.jdbc.PgSt
```

Investigation/Analysis

bytea column types should allow up to 1Gb of data (per old documentation: PostgreSQL: Documentation: 7.4: Storing Binary Data)

Example Java code provided by cx to test the issue:

Create test table:

```
create table content_packs_test (a text, b bytea);
```

Java code

```
package com.company;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.PrintStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Properties;

public class Main
{
  public static final String CONTENT_PACKS_TABLE = "CONTENT_PACKS_TEST";
  public static final String CONTENT_PACKS_COLUMN_VERSION = "VERSION";
  public static final String CONTENT_PACKS_COLUMN_BYTES = "CP_BYTES";

  public static void main(String[] args)
    throws SQLException, FileNotFoundException
  {
    if (args.length != 7)
    {
      System.out.println(" please provide 7 parameters");
      System.out.println(" cpFileName cpVersion DbServerName  dbPort dbName dbuser and dbpass");
      System.out.println("example:  CP_2021.05.zip 2021.05.96 localhost 5432 testDB  postgres ****");
      System.out.println(" The cpFileName should to be inserted as test should be in the same folder as this j
      System.out.println(" cpVersion can be retrieved from CP_2021.05.zip/version.dat");
      System.exit(-1);
    }
    String currentPath = new File(".").getAbsolutePath();
    System.out.println("Current dir:" + currentPath);
    String cpFileName = args[0];
    String cpVersion = args[1];
    String dbServerName = args[2];
    String dbPort = args[3];
    String dbName = args[4];
    String dbUser = args[5];
    String dbPass = args[6];
    File cpFile = new File(currentPath + File.separator + args[0]);

    Connection conn = getDbConnection(dbUser, dbPass, dbName, dbServerName, dbPort);

    FileInputStream fis = new FileInputStream(cpFile);
    String sql = "INSERT INTO CONTENT_PACKS_TEST VALUES (?, ?)";
    PreparedStatement ps = conn.prepareStatement(sql);
    ps.setString(1, cpVersion);
    int size = (int)cpFile.length();
    System.out.println(" writing to table CONTENT_PACKS_TEST version " + cpVersion + " from zip " + cpFileName
    ps.setBinaryStream(2, fis, size);
    System.out.println("Started the insert " + sql);
    ps.executeUpdate();
    System.out.println("END");
    ps.close();
    try
    {
      fis.close();
    }
    catch (IOException e)
    {
      e.printStackTrace();
      System.out.println("error" + e.getLocalizedMessage());
    }
  }

  public static Connection getDbConnection(String dbUser, String dbPass, String dbName, String dbServerName, S
    throws SQLException
  {
    String url = "jdbc:postgresql://" + dbServerName + ":" + dbPort + "/" + dbName;
    System.out.println("Connecting to " + url);
    Properties props = new Properties();
    props.setProperty("user", dbUser);
    props.setProperty("password", dbPass);

    return DriverManager.getConnection(url, props);
  }
}
```

Mitigation

Change parameter **log_min_duration_statement** to default value "-1" or a value greater than the insert statement duration.

Repro Steps

Issue can be reproduced in an Azure Database for PostgreSQL server and community deployments when setting parameter **log_min_duration_statement** to a small value different than default "-1", and insert operation is taking longer than this value.

Example Java test ouptut when log_min_duration_statement is set to default value "-1" or a value greater than the insert statement, loading a file from near 1Gb:

```
java com.company.test <filename> 1 xxx.postgres.database.azure.com 5432 postgres <password>
Current dir:/.
Connecting to jdbc:postgresql://xxx.postgres.database.azure.com:5432/postgres
writing to table CONTENT_PACKS_TEST version 1 from zip dump.test.sql with size 1058051654
Started the insert INSERT INTO CONTENT_PACKS_TEST VALUES (?, ?)
END
```

Example Java test ouptut when log_min_duration_statement is set to default value "3000" (3 seconds), insert statement is taking longer than 3 seconds:

```
java com.company.test <filename> 1 xxx.postgres.database.azure.com 5432 postgres <password>
Current dir:/.
Connecting to jdbc:postgresql://xxx.postgres.database.azure.com:5432/postgres
 writing to table CONTENT_PACKS_TEST version 1 from zip dump.test.sql with size 1058051654
Started the insert INSERT INTO CONTENT_PACKS_TEST VALUES (?, ?)
Exception in thread "main" org.postgresql.util.PSQLException: ERROR: invalid memory alloc request size 2116103
        at org.postgresql.core.v3.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2532)
        at org.postgresql.core.v3.QueryExecutorImpl.processResults(QueryExecutorImpl.java:2267)
        at org.postgresql.core.v3.QueryExecutorImpl.execute(QueryExecutorImpl.java:312)
        at org.postgresql.jdbc.PgStatement.executeInternal(PgStatement.java:448)
        at org.postgresql.jdbc.PgStatement.execute(PgStatement.java:369)
        at org.postgresql.jdbc.PgPreparedStatement.executeWithFlags(PgPreparedStatement.java:153)
        at org.postgresql.jdbc.PgPreparedStatement.executeUpdate(PgPreparedStatement.java:119)
        at com.company.test.main(test.java:53)


<div id='csswikifeedback-start'></div>
<br/>
```

How good have you found this content?