

Database - Bug #2350

BUFFER-COPY errors related with copying fields with extent

07/28/2014 04:31 PM - Vadim Nebogatov

Status:	Closed	Start date:	07/28/2014
Priority:	Normal	Due date:	
Assignee:	Vadim Nebogatov	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Cleanup and Stablization for Server Features	case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#1 - 07/28/2014 04:34 PM - Vadim Nebogatov

- File vmn_upd20140728a.zip added

Attached buffer-copy-dyn-static-tt-no-display.p with added new 22-29 tests.

Progress results

Test 22
Result 22: yes
Test 23
Result 23: yes
Test 24
Pair extents do not match for days1 and days. (11857)
Test 25
Result 25: yes
Test 26
Pair extents do not match for days and days1. (11857)
Test 27
Result 27: yes
Test 28
Result 28: yes
Test 29
Pair with schedule failed to have a member in each table. (9034)

P2J results

Test 22
Result 22: yes
Test 23
Result 23: yes
Test 24
** Array subscript 4 is out of range. (26)
** Error performing buffer copy. (-1)
Caused by: com.goldencode.p2j.NumberedException: Array subscript 4 is out of range
Test 25
Result 25: yes
Test 26
Result 26: yes
Test 27
** Error performing buffer copy. (-1)
java.lang.IllegalArgumentException: wrong number of arguments
Test 28
** Error performing buffer copy. (-1)

```
java.lang.IllegalArgumentException
Test 29
Pair with schedule failed to have a member in each table. (9034)
```

#2 - 07/28/2014 04:48 PM - Vadim Nebogatov

- Status changed from New to WIP

#3 - 07/29/2014 12:47 PM - Vadim Nebogatov

- File vmn_upd20140729a.zip added

I have attached update vmn_upd20140729a.zip fixing all these cases excepting one problem, it seems not directly related with issue.

Tests 27, 28 passed separately, but failed in whole attached buffer-copy-dyn-static-tt-no-display.p:

```
com.goldencode.p2j.persist.PersistenceException: org.hibernate.NonUniqueObjectException: a different object with the same identifier value was already associated with the session: [com.goldencode.p2j.persist.dynamic._temp.impl.DynamicRecordImpl#1]
    at com.goldencode.p2j.persist.RecordBuffer$Handler.invoke(RecordBuffer.java:9926)
    at com.goldencode.p2j.persist.$__Proxy3.setField5(Unknown Source)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3244)
    at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:4198)
    at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3351)
    at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1052)
    at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1137)
    at com.goldencode.testcases.Test$1$43.body(Test.java:1070)
    at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
    at com.goldencode.p2j.util.BlockManager.doBlockWorker(BlockManager.java:7903)
    at com.goldencode.p2j.util.BlockManager.doBlock(BlockManager.java:500)
    at com.goldencode.testcases.Test$1.body(Test.java:1065)
    at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
    at com.goldencode.p2j.util.BlockManager.topLevelBlock(BlockManager.java:6881)
    at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:221)
    at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:203)
    at com.goldencode.testcases.Test.execute(Test.java:39)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.Utils.invoke(Utils.java:1238)
    at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1701)
    at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1201)
    at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:364)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
    at java.lang.Thread.run(Thread.java:744)
Caused by: org.hibernate.NonUniqueObjectException: a different object with the same identifier value was already associated with the session: [com.goldencode.p2j.persist.dynamic._temp.impl.DynamicRecordImpl#1]
    at org.hibernate.engine.internal.StatefulPersistenceContext.checkUniqueness(StatefulPersistenceContext.java:691)
    at org.hibernate.event.internal.AbstractReassociateEventListener.reassociate(AbstractReassociateEventListener.java:72)
    at org.hibernate.event.internal.DefaultLockEventListener.onLock(DefaultLockEventListener.java:81)
    at org.hibernate.internal.SessionImpl.fireLock(SessionImpl.java:810)
    at org.hibernate.internal.SessionImpl.lock(SessionImpl.java:795)
    at com.goldencode.p2j.persist.RecordBuffer$Handler.invoke(RecordBuffer.java:9922)
    at com.goldencode.p2j.persist.$__Proxy3.setField5(Unknown Source)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

```

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3244)
at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:4198)
at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3351)
at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1052)
at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1137)
at com.goldencode.testcases.Test$1$43.body(Test.java:1070)
at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
at com.goldencode.p2j.util.BlockManager.doBlockWorker(BlockManager.java:7903)
at com.goldencode.p2j.util.BlockManager.doBlock(BlockManager.java:500)
at com.goldencode.testcases.Test$1.body(Test.java:1065)
at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
at com.goldencode.p2j.util.BlockManager.topLevelBlock(BlockManager.java:6881)
at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:221)
at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:203)
at com.goldencode.testcases.Test.execute(Test.java:39)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at com.goldencode.p2j.util.Utills.invoke(Utills.java:1238)
at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1701)
at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1201)
at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:364)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
at java.lang.Thread.run(Thread.java:744)

```

[07/29/2014 20:17:12 MSK] (com.goldencode.p2j.persist.RecordBuffer:SEVERE) Error performing buffer copy
java.lang.reflect.InvocationTargetException

```

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3244)
at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:4198)
at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3351)
at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1052)
at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1137)
at com.goldencode.testcases.Test$1$43.body(Test.java:1070)
at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
at com.goldencode.p2j.util.BlockManager.doBlockWorker(BlockManager.java:7903)
at com.goldencode.p2j.util.BlockManager.doBlock(BlockManager.java:500)
at com.goldencode.testcases.Test$1.body(Test.java:1065)
at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
at com.goldencode.p2j.util.BlockManager.topLevelBlock(BlockManager.java:6881)
at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:221)
at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:203)
at com.goldencode.testcases.Test.execute(Test.java:39)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at com.goldencode.p2j.util.Utills.invoke(Utills.java:1238)
at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1701)
at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1201)
at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:364)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
at java.lang.Thread.run(Thread.java:744)

```

Caused by: com.goldencode.p2j.util.ErrorConditionException: com.goldencode.p2j.persist.PersistenceException: o
rg.hibernate.NonUniqueObjectException: a different object with the same identifier value was already associate

```

d with the session: [com.goldencode.p2j.persist.dynamic._temp.impl.DynamicRecordImpl#1]
  at com.goldencode.p2j.util.ErrorManager.throwError(ErrorManager.java:1229)
  at com.goldencode.p2j.util.ErrorManager.recordOrThrowError(ErrorManager.java:1131)
  at com.goldencode.p2j.persist.RecordBuffer$Handler.invoke(RecordBuffer.java:10201)
  at com.goldencode.p2j.persist.$__Proxy3.setField5(Unknown Source)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:606)
  at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3244)
  at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:4198)
  at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3351)
  at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1052)
  at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1137)
  at com.goldencode.testcases.Test$1$43.body(Test.java:1070)
  at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
  at com.goldencode.p2j.util.BlockManager.doBlockWorker(BlockManager.java:7903)
  at com.goldencode.p2j.util.BlockManager.doBlock(BlockManager.java:500)
  at com.goldencode.testcases.Test$1.body(Test.java:1065)
  at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
  at com.goldencode.p2j.util.BlockManager.topLevelBlock(BlockManager.java:6881)
  at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:221)
  at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:203)
  at com.goldencode.testcases.Test.execute(Test.java:39)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:606)
  at com.goldencode.p2j.util.Utills.invoke(Utills.java:1238)
  at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1701)
  at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1201)
  at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:364)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:606)
  at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
  at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
  at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
  at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
  at java.lang.Thread.run(Thread.java:744)

```

Caused by: com.goldencode.p2j.persist.PersistenceException: org.hibernate.NonUniqueObjectException: a different object with the same identifier value was already associated with the session: [com.goldencode.p2j.persist.dynamic._temp.impl.DynamicRecordImpl#1]

```

  at com.goldencode.p2j.persist.RecordBuffer$Handler.invoke(RecordBuffer.java:9926)
  at com.goldencode.p2j.persist.$__Proxy3.setField5(Unknown Source)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:606)
  at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3244)
  at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:4198)
  at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3351)
  at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1052)
  at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1137)
  at com.goldencode.testcases.Test$1$43.body(Test.java:1070)
  at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
  at com.goldencode.p2j.util.BlockManager.doBlockWorker(BlockManager.java:7903)
  at com.goldencode.p2j.util.BlockManager.doBlock(BlockManager.java:500)
  at com.goldencode.testcases.Test$1.body(Test.java:1065)
  at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
  at com.goldencode.p2j.util.BlockManager.topLevelBlock(BlockManager.java:6881)
  at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:221)
  at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:203)
  at com.goldencode.testcases.Test.execute(Test.java:39)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:606)
  at com.goldencode.p2j.util.Utills.invoke(Utills.java:1238)
  at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1701)
  at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1201)
  at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:364)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)

```

```

    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
    at java.lang.Thread.run(Thread.java:744)
Caused by: org.hibernate.NonUniqueObjectException: a different object with the same identifier value was already associated with the session: [com.goldencode.p2j.persist.dynamic._temp.impl.DynamicRecord1Impl#1]
    at org.hibernate.engine.internal.StatefulPersistenceContext.checkUniqueness(StatefulPersistenceContext.java:691)
    at org.hibernate.event.internal.AbstractReassociateEventListener.reassociate(AbstractReassociateEventListener.java:72)
    at org.hibernate.event.internal.DefaultLockEventListener.onLock(DefaultLockEventListener.java:81)
    at org.hibernate.internal.SessionImpl.fireLock(SessionImpl.java:810)
    at org.hibernate.internal.SessionImpl.lock(SessionImpl.java:795)
    at com.goldencode.p2j.persist.RecordBuffer$Handler.invoke(RecordBuffer.java:9922)
    at com.goldencode.p2j.persist.$__Proxy3.setField5(Unknown Source)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3244)
    at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:4198)
    at com.goldencode.p2j.persist.RecordBuffer.copy(RecordBuffer.java:3351)
    at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1052)
    at com.goldencode.p2j.persist.BufferImpl.bufferCopy(BufferImpl.java:1137)
    at com.goldencode.testcases.Test$1$43.body(Test.java:1070)
    at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
    at com.goldencode.p2j.util.BlockManager.doBlockWorker(BlockManager.java:7903)
    at com.goldencode.p2j.util.BlockManager.doBlock(BlockManager.java:500)
    at com.goldencode.testcases.Test$1.body(Test.java:1065)
    at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:7048)
    at com.goldencode.p2j.util.BlockManager.topLevelBlock(BlockManager.java:6881)
    at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:221)
    at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:203)
    at com.goldencode.testcases.Test.execute(Test.java:39)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.Utills.invoke(Utills.java:1238)
    at com.goldencode.p2j.main.StandardServer$MainInvoker.execute(StandardServer.java:1701)
    at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1201)
    at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:364)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
    at java.lang.Thread.run(Thread.java:744)

```

It seems something wrong with primary key generator for Hibernate.

#4 - 07/30/2014 11:57 AM - Ovidiu Maxiniuc

See #2353, note 1.

#5 - 07/30/2014 03:14 PM - Eric Faulhaber

- *Project changed from Liberty to Database*

#6 - 07/30/2014 03:14 PM - Eric Faulhaber

- *Target version set to Milestone 11*

#7 - 07/30/2014 03:23 PM - Vadim Nebogatov

NonUniqueObjectException in note 3 thrown in code RecordBuffer, 9922:

```
        if (extent && createScope < 0 && !session.contains(currentRecord))
        {
            try
            {
                session.lock(currentRecord, LockMode.NONE);
            }
            catch (HibernateException exc)
            {
                throw new PersistenceException(exc);
            }
        }
    }
```

As far as I understood at this time there are several entity instances with the same id.
I am trying to localize this problem. BTW everything works as required if simply comment this code.

#8 - 07/30/2014 05:12 PM - Eric Faulhaber

Vadim Nebogatov wrote:

Attached buffer-copy-dyn-static-tt-no-display.p with added new 22-29 tests.

Please check this into the testcases project.

#9 - 07/30/2014 05:23 PM - Eric Faulhaber

Vadim Nebogatov wrote:

NonUniqueObjectException in note 3 thrown in code RecordBuffer, 9922:

[...]

As far as I understood at this time there are several entity instances with the same id.
I am trying to localize this problem. BTW everything works as required if simply comment this code.

This code was added to prevent a Hibernate LazyInitializationException, because in some cases, we were working with a record and the collection for the related extent field data was not initialized with data from the database. If we comment this out, I suspect it will result in regressions. However, it does not appear to be safe in other situations (as you are seeing here).

We actually have been wanting a simple recreate for this situation for some time! It has been reported in production periodically for years, but we never knew how to reproduce it.

Please work on the issue in note 4 first, then come back to this.

#10 - 07/30/2014 05:34 PM - Vadim Nebogatov

- Target version deleted (Milestone 11)

Eric Faulhaber wrote:

Vadim Nebogatov wrote:

Attached buffer-copy-dyn-static-tt-no-display.p with added new 22-29 tests.

Please check this into the testcases project.

Committed:

modified uast/buffer-copy-dyn-static-tt-no-display.p

modified uast/buffer/buffer-copy-dyn-static-tt-no-display.p

#11 - 08/01/2014 09:58 AM - Eric Faulhaber

- Target version set to Milestone 11

#12 - 08/01/2014 04:52 PM - Vadim Nebogatov

- File `vmn_upd20140801a.zip` added

`vmn_upd20140729a.zip` merged with latest bzt revision 10591 to `vmn_upd20140801a.zip`.

#13 - 08/01/2014 06:34 PM - Eric Faulhaber

Code review 20140801a:

- The use of `TempTableBuilder.getExtentSize` at lines 3812 and 3814 does not look correct. What if the source and destination buffers are not for dynamic temp-tables? You should be able simply to invoke the `size` methods instead (by this point, there should be a DMO available in each buffer on which to invoke the method). Please also add a comment at the top of the if statement (above line 3805) to describe the condition this block of code is meant to handle, if different than the situation described at line 3789.
- Also, just a minor formatting issue: please move the opening brace at line 3229 to the next line, to be consistent with the rest of the code.

#14 - 08/02/2014 08:19 AM - Vadim Nebogatov

- File `vmn_upd20140802a.zip` added

Attached `vmn_upd20140802a.zip` with corrections according review (note 13) merged with latest bzt revision 10592.

#15 - 08/02/2014 09:24 AM - Eric Faulhaber

Code review 20140802a:

Please log a warning inside the catch block at line 3829, rather than eating the exception. We really should never get here, but if we ever did, it would be a difficult issue to track down.

I noticed you prepended `opType == OperationType.COPY &...` to the if condition at line 3809. Two things:

- You are using the bitwise AND operator (`&`) here instead of logical AND (`&&`), which looks like a typo.
- This was not part of yesterday's update; did you find this situation applies only to `BUFFER-COPY` and not `BUFFER-COMPARE`?

#16 - 08/02/2014 10:05 AM - Vadim Nebogatov

Eric Faulhaber wrote:

I noticed you prepended `opType == OperationType.COPY &...` to the if condition at line 3809. Two things:

- You are using the bitwise AND operator (`&`) here instead of logical AND (`&&`), which looks like a typo.

& - it is typo, thanks

- This was not part of yesterday's update; did you find this situation applies only to `BUFFER-COPY` and not `BUFFER-COMPARE`?

I missed this from scratch. It happens only for BUFFER-COPY, when mapped two fields with extent have different extents.

#17 - 08/03/2014 07:27 AM - Vadim Nebogatov

- File `vmn_upd20140803a.zip` added

I have attached update `vmn_upd20140803a.zip` with fixes issues from Eric review and fix from Stas.

#18 - 08/03/2014 06:17 PM - Vadim Nebogatov

- File `buffer-copy-pair-list.p` added

Condition `RecordBuffer`, 3661

```
if (size > (Math.min(srcFieldMap.size(), dstFieldMap.size()) * 2) || (size % 2) != 0)
```

for error

```
Pairs list had too many pairs or odd number of entries. (9035)
```

seems not correct for BUFFER-COPY method.

I was able to reproduce this error for BUFFER-COPY only in case when pairs-list contains odd number of entries. Pairs-list for BUFFER-COPY seems has no size-related restrictions at all and could contain the same fields pairs several times.

There is only Progress restriction for statement size

```
** More than 15000 characters in a single statement--use -inp parm. (135)
```

I received it when tried to use several thousands of pairs.

Please check attached `buffer-copy-pair-list.p`. Execution results:

Progress

```
Test 1
Result 1:  yes
Test 2
Pair extents do not match for a1 and b1. (11857)
Test 3
Result 3:  yes
Test 4
Pairs list had too many pairs or odd number of entries. (9035)
Test 5
Pair extents do not match for a1 and b1. (11857)
Test 6
Result 6:  yes
Test 7
Result 7:  yes
Test 8
Result 8:  yes
```

P2J

```
Test 1
Result 1:  yes
Test 2
Pair extents do not match for a1 and b1. (11857)
Test 3
Pairs list had too many pairs or odd number of entries. (9035)
```

```
Test 4
Pairs list had too many pairs or odd number of entries. (9035)
Test 5
Pairs list had too many pairs or odd number of entries. (9035)
Test 6
Pairs list had too many pairs or odd number of entries. (9035)
Test 7
Pairs list had too many pairs or odd number of entries. (9035)
Test 8
Pairs list had too many pairs or odd number of entries. (9035)
```

#19 - 08/03/2014 10:58 PM - Eric Faulhaber

Code review 20140803a:

It seems this code will work, but I'm wondering about some choices:

- By adding `getSize` as a separate method, it introduces the possibility that other callers will use it. However, it lacks safety code to make sure there is a record in the buffer before invoking the sizer. In the previous situation, we knew there was a record there, but as a general purpose method, there should be a check.
- Why is `getSize` static? If we have to pass in the `RecordBuffer` object as a parameter in order to invoke the method, why not just make this an instance method instead?
- It would be more efficient to invoke the sizer method on the `currentRecord` object directly, rather than on the DMO proxy. The latter forces us through the higher overhead of the invocation handler, but there is no benefit to doing this.
- There are other areas you've touched in `RecordBuffer` which invoke the sizer method directly (e.g., `RecordBuffer.copy`). Why not re-use the new method in those places to be consistent?

#20 - 08/04/2014 10:42 AM - Eric Faulhaber

Vadim Nebogatov wrote:

Condition `RecordBuffer`, 3661

```
if (size > (Math.min(srcFieldMap.size(), dstFieldMap.size()) * 2) || (size % 2) != 0)
```

for error

```
Pairs list had too many pairs or odd number of entries. (9035)
```

seems not correct for BUFFER-COPY method.

[...]

Yes, it looks like that length restriction is wrong; I'm not sure why I came up with that. It seems checking instead for an odd number of pairs and testing the overall length limit of 15K characters should fix this, right?

Might as well roll this fix into the next update, unless you think it is more complicated than that.

#21 - 08/04/2014 11:12 AM - Vadim Nebogatov

Eric Faulhaber wrote:

Yes, it looks like that length restriction is wrong; I'm not sure why I came up with that.

Maybe it was taken from BUFFER-COMPARE testing? I did not check similar tests for BUFFER-COMPARE, but possible it has another restrictions.

It seems checking instead for an odd number of pairs and testing the overall length limit of 15K characters should fix this, right?

Yes, maybe do this change only for BUFFER-COPY, and remain current condition for BUFFER-COMPARE?

limit of 15K characters - it is common restriction, possible should be checked both for BUFFER-COPY and BUFFER-COMPARE, maybe for all other methods – not tested so far.

Might as well roll this fix into the next update, unless you think it is more complicated than that.

I think it could be incorporated in the same update. I already tested #2353 with only odd number of pairs restriction and received promising results: some returned fields are already not null (earlier all fields returned as nulls). Only issue will remain: search results contain not one but several (10+) equal records.

#22 - 08/04/2014 07:11 PM - Vadim Nebogatov

[ECF: moved content of this note to #2353, note 13]

#23 - 08/05/2014 04:09 PM - Vadim Nebogatov

I just noticed that message

`** More than 15000 characters in a single statement--use -inp parm. (135)`

is compiler time message.

I am not sure if I received it earlier as runtime message. To be sure I tried to reach very long pairs-list dynamically with the test like :

```
...
DO TRANSACTION ON ERROR UNDO, LEAVE:
  def var i as integer.
  def var s as character.
  s = "a4,b2".
  DO i=1 TO 1800:
    s = s + ",a4,b2".
  END.
  bool = hba:buffer-copy(hbb, "", s).
END.
```

The customer's testing system does not like such code very much and it exits immediately. I checked several times.

```
Test 8
SYSTEM ERROR: stkpush: stack overflow. Increase the -s parameter. (279)
** Save file named core for analysis by Progress Software Corporation (439).
Quit (core dumped)
```

I think we don't need to check 15K restriction evidently.

#24 - 08/05/2014 05:48 PM - Vadim Nebogatov

- File *vmn_upd20140805a.zip* added

I have attached next update *vmn_upd20140805a.zip* tested with #2353 and *buffer-copy-pair-list.p* from note 18. Both passed.

#25 - 08/05/2014 08:29 PM - Eric Faulhaber

Vadim Nebogatov wrote:

Eric Faulhaber wrote:

Yes, it looks like that length restriction is wrong; I'm not sure why I came up with that.

Maybe it was taken from BUFFER-COMPARE testing? I did not check similar tests for BUFFER-COMPARE, but possible it has another restrictions.

I don't think so -- I didn't work on BUFFER-COMPARE when I put this code in, and the condition does not exist in versions previous to that. I think it was just an incorrect assumption on my part. To be sure, please write a Progress test case to confirm there is no such limit under BUFFER-COMPARE.

It seems checking instead for an odd number of pairs and testing the overall length limit of 15K characters should fix this, right?

Yes, maybe do this change only for BUFFER-COPY, and remain current condition for BUFFER-COMPARE?

Only leave the condition in for BUFFER-COMPARE if the test case confirms it is needed, otherwise remove it.

#26 - 08/05/2014 09:08 PM - Eric Faulhaber

Code review 20140805a:

- As noted above, the if condition at line 3662 should drop the BUFFER-COMPARE-specific part, if a test case confirms it is not consistent with Progress.
- I just discovered that the new getSize method is redundant with an existing method, getExtent. Please use getExtent instead, but improve it to invoke the sizer method on currentRecord instead of dmoProxy. Please also add a @throws tag to the javadoc for that method to indicate that NullPointerException is thrown if there is no record in the buffer. I think this is OK, since it would represent a programming error to call the method in a situation where currentRecord can be null.

If you make no other changes besides these, please post your update, but go ahead and begin regression testing without waiting for a code review.

#27 - 08/06/2014 09:09 AM - Vadim Nebogatov

- File vmn_upd20140806a.zip added

Please check attached buffer-compare-pair-list.p. Execution results:

Progress

```
Test 1
Result 1:  no
Test 2
Pair extents do not match for a1 and b1. (11857)
Result 2:  no
Test 3
Result 3:  no
Test 4
Pairs list had too many pairs or odd number of entries. (9035)
Result 4:  no
Test 5
Pair extents do not match for a1 and b1. (11857)
Result 5:  no
Test 6
Result 6:  no
Test 7
Result 7:  no
Test 8
Result 8:  no
```

P2J

```
Test 1
Result 1:  no
Test 2
Result 2:  no
Test 3
Pairs list had too many pairs or odd number of entries. (9035)
Test 4
Pairs list had too many pairs or odd number of entries. (9035)
Test 5
Pairs list had too many pairs or odd number of entries. (9035)
Test 6
Pairs list had too many pairs or odd number of entries. (9035)
Test 7
Pairs list had too many pairs or odd number of entries. (9035)
Test 8
Pairs list had too many pairs or odd number of entries. (9035)
```

I have attached update with corrected getExtent(), removed condition for BUFFER-COMPARE and merged with latest b2r rev. 10593.

After fixes P2J results

```
Test 1
Result 1:  no
Test 2
Pair extents do not match for a1 and b1. (11857)
Test 3
Result 3:  no
Test 4
Pairs list had too many pairs or odd number of entries. (9035)
Test 5
Pair extents do not match for a1 and b1. (11857)
Test 6
Result 6:  no
Test 7
Result 7:  no
Test 8
Result 8:  no
```

Results are the the same as in Progress excepting that method result is not printed after error. Progress catches error and returns boolean value.

#28 - 08/06/2014 09:09 AM - Vadim Nebogatov

- File *buffer-compar-pair-list.p* added

#29 - 08/06/2014 10:07 AM - Eric Faulhaber

Your test results, where we are not returning a result for 2, 4, and 5 (while Progress returns false) tells me that we are raising an error condition and leaving the transaction block for these errors, while Progress is not. This is appropriate for BUFFER-COPY, but not BUFFER-COMPARE. I think we should catch the `ErrorConditionException` in the BUFFER-COMPARE case and return false.

#30 - 08/06/2014 10:13 AM - Eric Faulhaber

Code review 20140806a:

Please note the path of the source file in the zip archive is missing the leading `src/` directory. Please fix that and make the adjustment for BUFFER-COMPARE mentioned in note 29, then regression test.

#31 - 08/06/2014 02:02 PM - Vadim Nebogatov

- File *vmn_upd20140806b.zip* added

Corrected update *vmn_upd20140806b.zip* is attached, test results for note 27 are like in Progress. Regression started.

#32 - 08/06/2014 03:41 PM - Eric Faulhaber

Code review 20140806b:

Line 2790 catches `ConditionException` instead of `ErrorConditionException`, as I requested in note 29. This will improperly catch other conditions besides `ERROR`. We don't want to do this; `STOP` and `QUIT` conditions, for instance, should be allowed to unwind the stack and are meant to be handled by the block manager.

Please correct this and confirm that your test case still works, then restart regression testing.

#33 - 08/06/2014 06:05 PM - Vadim Nebogatov

- File *vmn_upd20140807a.zip* added

Corrected update *vmn_upd20140807a.zip* is attached. Tests still work properly, search API (#2353) also works. Regression stopped, will restart regression tomorrow.

#34 - 08/07/2014 06:15 PM - Vadim Nebogatov

Regression 20140807_101619: failed `tc_job_002`, `tc_job_clock_002`, `gso178_session1`, `gso178_session2`, `gso242_session1`, `gso242_session2`, `gso_364`, `gso_243` - all timeout related. Will start regression one more time.

#35 - 08/08/2014 02:07 PM - Vadim Nebogatov

Regression 20140808_092455: failed `tc_job_002`, `tc_job_clock_002` from previous regression. Will repeat "main-regression" target one more time.

#36 - 08/09/2014 06:42 PM - Vadim Nebogatov

Regression 20140809_123827: failed `tc_job_002`, `tc_job_clock_002` from previous regression. Will repeat regression one more time.

#37 - 08/10/2014 04:05 PM - Vadim Nebogatov

Regression 20140810_120027: failed tc_job_002, tc_job_clock_002 from previous regression. Will repeat regression one more time.

#38 - 08/11/2014 05:30 AM - Vadim Nebogatov

Regression completed: tc_job_clock_002 finally passed in 20140810_163259.

#39 - 08/11/2014 07:45 AM - Eric Faulhaber

Please commit and distribute the 0807a update.

#40 - 08/11/2014 08:44 AM - Vadim Nebogatov

update [vmn_upd20140807a.zip] committed to bzd rev. 10595 and distributed.

#41 - 08/11/2014 09:03 AM - Eric Faulhaber

- % Done changed from 0 to 100

- Status changed from WIP to Closed

#42 - 11/16/2016 12:06 PM - Greg Shah

- Target version changed from Milestone 11 to Cleanup and Stabilization for Server Features

Files

vmn_upd20140728a.zip	1.69 KB	07/28/2014	Vadim Nebogatov
vmn_upd20140729a.zip	88.7 KB	07/29/2014	Vadim Nebogatov
vmn_upd20140801a.zip	88.7 KB	08/01/2014	Vadim Nebogatov
vmn_upd20140802a.zip	88.8 KB	08/02/2014	Vadim Nebogatov
vmn_upd20140803a.zip	88.9 KB	08/03/2014	Vadim Nebogatov
buffer-copy-pair-list.p	2.22 KB	08/03/2014	Vadim Nebogatov
vmn_upd20140805a.zip	88.9 KB	08/05/2014	Vadim Nebogatov
vmn_upd20140806a.zip	88.5 KB	08/06/2014	Vadim Nebogatov
buffer-compar-pair-list.p	2.39 KB	08/06/2014	Vadim Nebogatov
vmn_upd20140806b.zip	88.7 KB	08/06/2014	Vadim Nebogatov
vmn_upd20140807a.zip	88.7 KB	08/06/2014	Vadim Nebogatov