Altibase 7.1.0.6.1 Patch Notes

Table Of Contents

Fixed Bugs

BUG-47830 Altibase server abnormally terminates if an input/output variable (SQL_PARAM_INPUT_OUTPUT) is used for the type parameter of the SQLBindParameter function when binding a LOB data type.

- **module** : qp-dml-execute
- **Category** : Fatal
- **Reproducibility** : Always
- **Description** : Fixed a bug where the Altibase server abnormally terminates if an input/output variable (SQL_PARAM_INPUT_OUTPUT) is used for the type parameter of the SQLBindParameter function when binding LOB data types.
- **How to reproduce this bug**
  - **Reproduction conditions**

```
CREATE TABLE bug47830 (i1 integer, i2 clob );
INSERT INTO bug47830 VALUES(1, 'abc');
    /* prepares an SQL string for execution */
    rc = SQLPrepare(stmt, (SQLCHAR *)"INSERT INTO bug47830 VALUES(?,
?)", SQL_NTS);
    if (!SQL_SUCCEEDED(rc))
    {
        PRINT_DIAGNOSTIC(SQL_HANDLE_STMT, stmt, "SQLPrepare");
        goto EXIT_STMT;
    }
    /* binds a buffer to a parameter marker in an SQL statement */
    rc = SQLBindParameter(stmt, 1, SQL_PARAM_INPUT,
                          SQL_C_SLONG, SQL_INTEGER,
                          0, 0,
                          &i1,
                          0,
                          NULL);
    if (!SQL_SUCCEEDED(rc))
    {
        PRINT_DIAGNOSTIC(SQL_HANDLE_STMT, stmt, "SQLBindParameter");
        goto EXIT_STMT;
    }
    rc = SQLBindParameter(stmt,
                          2,
                          SQL_PARAM_INPUT_OUTPUT,
                          SQL_C_CLOB_LOCATOR, SQL_CLOB,
                          0,
                          0,
                          &c1_loc, 0, &c1_ind);
    if (!SQL_SUCCEEDED(rc))
...
```

  - **Actual Results**

        - Altibase client side

          The following error occurs in the application.

          ```
          Diagnostic Record 1
            SQLSTATE     : 08S01
            Message text : Communication link failure. Connection closed
            Message len  : 45
            Native error : 0x51043
          ```

        - Altibase server side

          The following call stack is printed in the altibase trace log altibase_error.log.

          ```
          BEGIN-STACK [CRASH] ==============================
          Caller[0] 0000000000418CE1      => mmmSignalHandler
          Caller[1] 0000003620C0F500      => not found
          Caller[2] 000000000042EB88      =>
          mmtCopyBindParamDataCallback(qciBindParam, unsigned char, unsigned
          int)
          Caller[3] 00000000004D2054      =>
          qci::setBindParamData(qciStatement, unsigned short, unsigned char,
          IDE_RC ()(qciBindParam, unsigned char, unsigned int))
          Caller[4] 0000000000430A78      =>
          mmtServiceThread::paramDataInProtocol(cmiProtocolContext,
          cmpProtocol, void, void)
          Caller[5] 0000000000F94017      => cmiRecv(cmiProtocolContext, void,
          PDL_Time_Value, void)
          Caller[6] 000000000042AA7F      =>
          mmtServiceThread::executeTask_READY(mmcTask, mmcTaskState)
          Caller[7] 000000000042C5F5      => mmtServiceThread::executeTask()
          Caller[8] 000000000042CBC2      =>
          mmtServiceThread::multiplexingAsShared()
          Caller[9] 000000000042D257      => mmtServiceThread::run()
          Caller[10] 0000000000FC9A72     => idtContainer::staticRunner(void)
          Caller[11] 0000003620C07851     => not found
          Caller[12] 00000036204E767D     => not found
          END-STACK ========================================
          ```

    - **Expected Results**

      ```
      Diagnostic Record 1
        SQLSTATE     : HY090
        Message text : Invalid binding to host variables
        Message len  : 33
        Native error : 0x31126
      Diagnostic Record 2
        SQLSTATE     : HY090
        Message text : Invalid binding to host variables
        Message len  : 33
        Native error : 0x31126
      ```

- **Workaround**
- **Changes**

- Performance view
- Property
- Compile Option
- Error Code

BUG-48603 If a reused TABLE OID is included in a replication gap, replication start fails with the error 'The row already exists in a unique index'.

- **module** : rp
- **Category** : Functional Error
- **Reproducibility** : Frequence
- **Description** :

  Fixed a bug where starting replication fails with an error in the Reproduction conditions.

  ```
  1. Replication is stopped or there is a replication gap
  2. Execute DDL on replication target table (A)
  3. Restart the Altibase server
  4. Execute DDL to replication target table (B)
  5. Delete the replication target table from the replication object and add it again
  6. Replication start
  ```

  This phenomenon occurs when information in the replication meta table (SYS_REPL_OLD_ITEMS_) conflicts due to the reused TABLE OID included in the replication gap after restarting the Altibase server. When this phenomenon occurs, altibase_rp.log The following log is output.

  ```
  ERR-11058(errno=0) The row already exists in a unique index.
  ERR-610f8(errno=0) [Sender] Failed to add a XLog [TID:-1603452059,
  SN:133280338892, Type:2, Log Type:1, Change Log Type:0]
  ERR-61014(errno=0) [Sender] Failed to make XLOG in a log file at
  SN[133280338892]
  ```

  Added meta table SYS_REPL_TABLE_OID_IN_USE_ while fixing this bug. The contents of this meta table can be found in the GeneralReference Manual

  Also the meta version has changed. If you need to roll back a patch, you will need to do a meta downgrade.

- **How to reproduce this bug**
  - **Reproduction conditions**

    ```
    1. Replication is stopped or there is a replication gap
    2. Execute DDL on replication target table (A)
    3. Restart the Altibase server
    4. Execute DDL to replication target table (B)
    5. Delete the replication target table from the replication object and add it again
    6. Replication start
    ```

  - **Actual Results**

    Replication start fails and the following log is output to $ALTIBASE_HOME/trc/altibase_rp.log.

    ```
    [2021/02/20 22:37:28] [Thread-140200585975552] [Level-0]
    [SenderXLog] Table Meta Log [TableOID:3096456->3098776, User Name:SYS,
    Table Name:T1, Partition Name:]
    [2021/02/20 22:37:28] [Thread-140200585975552] [Level-0]
    ERR-11058(errno=0) The row already exists in a unique index.
    [2021/02/20 22:37:28] [Thread-140200585975552] [Level-0]
    ERR-610f8(errno=0) [Sender] Failed to add a XLog [TID:23297,
    SN:31532356, Type:2, Log Type:1, Change Log Type:0]
    [2021/02/20 22:37:28] [Thread-140200585975552] [Level-0]
    ERR-61014(errno=0) [Sender] Failed to make XLOG in a log file at
    SN[31532356]
    [2021/02/20 22:37:28] [Thread-140200585975552] [Level-0]
    SN=<31532356>, MAGIC: 3320, TID: 23297, BE: N, REP: Y, ISVP: N,
    ISVP_DEPTH: 0, PLSN=<0, 480, 10071066>, LT: < 38 >, SZ: 49
    [2021/02/20 22:37:28] [Thread-140200585975552] [Level-0]
    ERR-61013(errno=0) [Sender] Replication failed
    ```

  - **Expected Results**

    ```
    Replication starts normally.
    ```

- **Workaround**

  ```
  Resolves the replication gap and executes DDL statements on the replication
  target table.
  ```

- **Changes**

  - Performance view
  - Property
  - Compile Option
  - Error Code

BUG-49168 Reflects the "PROJ-2749 Common Table Expression (CTE) Validation Phase Optimization" project.

- **module** : qp-dml-pvo
- **Category** : Enhancement
- **Reproducibility** : Unknown
- **Description** :

  Three things were applied in this bug.

  1. Fix the phenomenon that an error occurs when executing a long SQL statement using the WITH clause repeatedly and improve SQL processing performance

     By applying the COMPACT WITH technique, [ERR-3111D: There are too many DML statements in the stored procedure, or the SQL query is too long.] error does not occur and SQL processing performance is improved.

> COMPACT WITH technique
> One statement is shared to improve the problem of internally creating n+1 statements when using a view created with the WITH clause (excluding Recursive with) n times.

2. When the ORDER BY clause is used in the WITH statement, the unnecessary ORDER BY elimination (Order By Elimination, OBYE) technique is applied to improve SQL performance.
3. Apply view optimization function to views created with WITH clause

To apply the COMPACT WITH technique added in this bug to Altibase 7.1, the value of the private property must be changed. The following is the impact of applying the COMPACT WITH technique.

- Because the COMPACT WITH technique is not cost-based, it has the potential to generate inefficient execution plans.
- Applying the COMPACT WITH technique changes the SQL execution plan that meets the condition.
- If the COMPACT WITH technique is applied, memory usage may increase when executing SQL that meets the conditions.
  If you wish to apply the COMPACT WITH technique despite the above effects, contact the Altibase Technical Support Center.

- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

BUG-49250 (JDBC Driver) If a prepare error occurs while repeating Direct-Execute statements, the previously executed statement may be re-executed.

- **module** : mm-jdbc
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Fixed a bug where a previously executed statement was re-executed if a prepare error occurred while executing a Direct-Execute statement repeatedly.

  This bug can occur when performing the following conditions sequentially.

```
1. direct execute #1
2. direct execute #2 (prepare error occurs)
3. prepare execute #3
4. direct execute #4 (The statement in #3 is executed)
```

- **How to reproduce this bug**
  - **Reproduction conditions**

```
Statement sStmt;
PreparedStatement sPstmt;
try {

    sStmt.execute("ALTER SESSION SET LOB_CACHE_THRESHOLD = 1048576");

} catch ( Exception e ) {

    System.out.println( "ERROR MESSAGE : " + e.getMessage()  );

}
try {
    sStmt.execute("ALTER SESSION CLOSE DATABASE LINK");
} catch ( Exception e ) {
    System.out.println( "ERROR MESSAGE : " + e.getMessage()  );
}

try {

    sPstmt = mConn.prepareStatement("ALTER SESSION SET 
LOB_CACHE_THRESHOLD = 1048576");
    sPstmt.execute();

} catch ( Exception e ) {

    System.out.println( "ERROR MESSAGE : " + e.getMessage() );

}
try {
    sStmt.execute("ALTER system CHECKPOINT");
} catch ( Exception e ) {

    System.out.println( "ERROR MESSAGE : " + e.getMessage() );
}
```

  - **Actual Results**

```
ALTER SESSION SET LOB_CACHE_THRESHOLD = 1048576
------------------------
ERROR MESSAGE : Property value overflow [LOB_CACHE_THRESHOLD]
ALTER SESSION CLOSE DATABASE LINK
------------------------
ERROR MESSAGE : SQL syntax error
line 1: parse error
ALTER SESSION CLOSE DATABASE LINK
                             ^
prepare-execute : ALTER SESSION SET LOB_CACHE_THRESHOLD = 1048576
------------------------
ERROR MESSAGE : Property value overflow [LOB_CACHE_THRESHOLD]
ALTER system CHECKPOINT
------------------------
```

```
    ERROR MESSAGE : Property value overflow [LOB_CACHE_THRESHOLD]
    ```

  - **Expected Results**

    ```
    ALTER SESSION SET LOB_CACHE_THRESHOLD = 1048576
    ------------------------
    ERROR MESSAGE : Property value overflow [LOB_CACHE_THRESHOLD]
    ALTER SESSION CLOSE DATABASE LINK
    ------------------------
    ERROR MESSAGE : SQL syntax error
    line 1: parse error
    ALTER SESSION CLOSE DATABASE LINK
                                 ^
    prepare-execute : ALTER SESSION SET LOB_CACHE_THRESHOLD = 1048576
    ------------------------
    ERROR MESSAGE : Property value overflow [LOB_CACHE_THRESHOLD]
    ALTER system CHECKPOINT
    ------------------------
    ```

- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

BUG-49263 [ERR-11041: A deadlock situation has been detected.] error may occur when continuously performing TRUNCATE on a partitioned table with multiple partitions included in replication.

- **module** : rp-control
- **Category** : Functional Error
- **Reproducibility** : Rare
- **Description** : Fixes a bug where an [ERR-11041: A deadlock situation has been detected.] error occurs when performing TRUNCATE continuously on a partitioned table with multiple partitions included in replication.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

BUG-49279 Adds the ability to extract object privileges for the QUEUE table.

- **module** : ux-aexport
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Adds the ability to extract object privileges for the QUEUE table.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

BUG-49281 If an alias is used for a column using ORDER BY in the OVER clause of an analytic function, and AND and OR conditions are used in the ON clause of a JOIN together, the Altibase server abnormally terminates.

- **module** : qp-select-execute
- **Category** : Fatal
- **Reproducibility** : Always
- **Description** : Fixes a bug where the Altibase server abnormally terminates when an alias is used for a column using ORDER BY in the OVER clause of an analytic function, and AND and OR conditions are used together in the JOIN ON clause.
- **How to reproduce this bug**
  - **Reproduction conditions**

```
CREATE TABLE bug49281_t1 ( i1 CHAR(3), i2 VARCHAR(16), i3 VARCHAR(12));
INSERT INTO bug49281_t1 SELECT LEVEL, LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL < 10;
CREATE TABLE bug49281_t2 ( i1 VARCHAR(17), i2 CHAR(2), i3 CHAR(3));
INSERT INTO bug49281_t2 SELECT LEVEL, LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL < 10;

SELECT B.i1
  FROM (SELECT i1, i2, ROW_NUMBER() OVER (PARTITION BY i3 ORDER BY i1) AS RN
          FROM bug49281_t1 ) A
  RIGHT OUTER JOIN
       (SELECT i1, '00' AS PCD FROM bug49281_t2) B
  ON (A.RN <> 1 OR ( A.RN = 1 AND A.i2 = B.PCD));
```

  - **Actual Results**

    Altibase server abnormally terminates.

  - **Expected Results**

    ```
    I1
    ---------------------
    1
    2
    3
    4
    5
    6
    7
    8
    9
    9 rows selected.
    ```

- **Workaround**

  You can avoid abnormal termination of the Altibase server by converting the queries in the ON clause.

  ```
  ON (A.RN <> 1 OR ( A.RN = 1 AND A.i2 = B.PCD))
  ->
  ON (A.RN <> 1 OR  A.RN = 1 ) AND ( A.RN <> 1 OR A.i2 = B.PCD)
  ```

- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

BUG-49283 LOCK WAIT occurred due to inserting a unique key with the same value in multiple sessions, but a no wait event occurred. Wait events are not collected.

- **module** : sm
- **Category** : Functional Error
- **Reproducibility** : Unknown
- **Description** : Fixes a bug where wait events are not collected in V$SESSION_WAIT or V$STATEMENT when LOCK WAIT occurs by inserting a unique key with the same value in multiple sessions. This bug is specific to disk tables. In the case of memory tables, it is normal for no wait event to occur under the same condition because there is no wait event.
- **How to reproduce this bug**
  - **Reproduction conditions**

    ```
    Session A
    iSQL> CREATE TABLE bug49283 (c1 INTEGER PRIMARY KEY) TABLESPACE
    SYS_TBS_DISK_DATA;
    ```

```
iSQL> ALTER SYSTEM SET TIMED_STATISTICS = 1;
iSQL> AUTOCOMMIT OFF;
iSQL> INSERT INTO bug49283 VALUES(1);

Session B
iSQL> AUTOCOMMIT OFF;
iSQL> INSERT INTO bug49283 VALUES(1);

Session A
iSQL> SELECT * FROM V$LOCK_WAIT;
iSQL> SELECT SID, EVENT FROM V$SESSION_WAIT WHERE SID <> SESSION_ID();
iSQL> SELECT TX_ID, SESSION_ID, EVENT, SUBSTR(QUERY, 1, 60) QRY FROM
V$STATEMENT WHERE SESSION_ID <> SESSION_ID();
```

- **Actual Results**

```

iSQL> SELECT * FROM V$LOCK_WAIT;
TRANS_ID             WAIT_FOR_TRANS_ID
---------------------------------------------
1408                 21633
1 row selected.
iSQL> SELECT SID, EVENT FROM V$SESSION_WAIT WHERE SID <> SESSION_ID();
SID                  EVENT
------------------------------------------------------------------
2                    no wait event
1 row selected.
iSQL> SELECT TX_ID, SESSION_ID, EVENT, SUBSTR(QUERY, 1, 60) QRY FROM
V$STATEMENT WHERE SESSION_ID <> SESSION_ID();
TX_ID                SESSION_ID  EVENT
  QRY
------------------------------------------------------------------------
----------
1408                 2           no wait event
  INSERT INTO bug49283_m VALUES(1)
1 row selected.
```

- **Expected Results**

```
iSQL> SELECT * FROM V$LOCK_WAIT;
TRANS_ID             WAIT_FOR_TRANS_ID
---------------------------------------------
1408                 21633
1 row selected.
iSQL> SELECT SID, EVENT FROM V$SESSION_WAIT WHERE SID <> SESSION_ID();
SID                  EVENT
------------------------------------------------------------------
2                    enq: TX - row lock contention, data row
1 row selected.
iSQL> SELECT TX_ID, SESSION_ID, EVENT, SUBSTR(QUERY, 1, 60) QRY FROM
V$STATEMENT WHERE SESSION_ID <> SESSION_ID();
TX_ID                SESSION_ID  EVENT
  QRY
```

```
------------------------------------------------------------------------
----------
1408                 2           enq: TX - row lock contention, data row
  INSERT INTO bug49283 VALUES(1)
1 row selected
```

- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

Changes

Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
7.1.0.6.1	6.5.1	8.10.1	7.1.7	7.4.6

> You can check the module version change history in Version_Histories.

Compatibility

Database binary version

The database binary version has not changed.

> The database binary version indicates the compatibility of database image files and log files. If this version needs to be patched to a different version, the database must be reorganized.

Meta Version

The meta version has not changed.

> If you want to roll back the patch after patching to a version with a changed meta version, see Meta Downgrade.

CM protocol Version

The cm protocol version has not changed.

Replication protocol Version

The replication protocol version has not changed.

Altibase Server Properties

Added properties

Changed properties

Deleted properties

Performance Views

Added performance views

Changed performance views

Deleted performance views