Altibase 7.1.0.6.4 Patch Notes

Table Of Contents

Fixed Bugs

BUG-49331 The replication sender terminates due to contention in the SYS_REPL_TABLE_OID_IN_USE meta table when executing a DDL statement on a replication target table.

- **module** : rp-control
- **Category** : Testcase Fail
- **Reproducibility** : Frequence
- **Description** : Fixes a bug where replication senders are terminated due to contention in the SYS_REPL_TABLE_OID_IN_USE meta table when executing DDL statements on replication target tables.

  When a table is included in multiple replication objects or when a DDL statement is executed on a replication target table, a contention occurs, and the following message is output to $ALTIBASE_HOME/trc/altibase_rp.log.

  ```
  [2021/10/12 14:31:14 79E][PID:193728][Thread-140046648854272][LWP-194294]
  ERR-1304c(errno=16) A record has already been updated.
  ```

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

BUG-49356 When performing aexport in full DB mode, indexes with different index owners than table owners are not script extracted.

- **module** : ux-aexport
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Fixes a bug where scripts are not extracted for indexes whose index owner is different from the table owner when performing aexport in Full DB mode.
- **How to reproduce this bug**
  - **Reproduction conditions**

```
    CREATE USER alti IDENTIFIED BY test;
    CONNECT alti/test
    DROP TABLE t1;
    CREATE TABLE t1 (c1 INTEGER, c2 CHAR(10), c3 CHAR(10));
    CREATE INDEX t1_idx1 ON t1 (c1);
    CONNECT sys/manager
    CREATE INDEX alti.t1_idx2 ON alti.t1 (c2);
    CREATE INDEX t1_idx3 ON alti.t1 (c3);
    ```

  - **Actual Results**

    ```
    $ aexport -s 127.0.0.1 -u sys -p manager...
    $ grep T1_IDX3 *
    $
    ```

  - **Expected Results**

    ```
    $ grep T1_IDX3 *ALL_CRT_INDEX.sql:create index "T1_IDX3" on "ALTI"."T1"
    ("C3") tablespace "SYS_TBS_MEM_DATA";
    ```

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

BUG-49376 When using DBLink, there is a problem that cannot be processed when a pending transaction occurs in a specific node at the Two-Phase Commit Level.

- **module** : dblink
- **Category** : Functional Error
- **Reproducibility** : Frequence
- **Description** : When using DBLink, when a transaction is being performed on multiple nodes at the Two-Phase Commit Level (DBLINK_GLOBAL_TRANSACTION_LEVEL = 2), if a specific node abnormally terminates during the commit/rollback process, Fixed a bug where transactions were left in a pending state.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option

  - Error Code

BUG-49378 Avoid abnormal shutdown of Altibase server during hash (HASH) operation in DISK TEMP TABLE and add logs for cause analysis.

- **module** : sm
- **Category** : Fatal
- **Reproducibility** : Impossible
- **Description** : Avoid abnormal shutdown of Altibase server during hash (HASH) operation in DISK TEMP TABLE and add logs for cause analysis. After reflecting this bug, when a problem occurs, the following log is left in $ALTIBASE_HOME/trc/altibase_dump.log. (However, __TEMPDUMP_LEVEL = 1 setting is required)

  ```
  DUMP HASH WASEGMENT:
  WASegPtr              : 0x7fb19e2031d8
  SpaceID               : 4
  Type                  : 1 HASH
  HashSlotCount         : 65536
  InMemory              : OutMemory
  Unique                : 0
  OpenCursorType        : 0
  EndWPID               : 384
  InsertGroup BeginWPID : 128
  InsertGroup EndWPID   : 320
  InsertGroup ReuseWPID : 131
  FetchGroup BeginWPID  : 384
  FetchGroup EndWPID    : 384
  FetchGroup ReuseWPID  : 384
  SubHashGroup BeginWPID : 320
  SubHashGroup EndWPID   : 384
  SubHashGroup ReuseWPID : 352
  WAExtentListCount     : 8
  MaxWAExtentCount      : 6
  AllocWAPageCount      : 384
  NPageHashBucketCnt    : 384
  NPageCount            : 386
  SubHashPageCount      : 64
  SubHashBuildCount     : 3
  HashSlotPageCount     : 128
  ```

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

  - Compile Option
  - Error Code

BUG-49402 [ERR-11041: A deadlock situation has been detected.] error may occur when executing a DDL statement in a triple or higher replication environment.

- **module** : rp-control
- **Category** : Functional Error
- **Reproducibility** : Rare
- **Description** : Fixes a bug where [ERR-11041 : A deadlock situation has been detected.] error occurs when executing DDL statements in a redundant environment with triple replication or higher.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

BUG-49408 In a replication environment using the DDL replication function, the Altibase server may abnormally terminate if a DDL statement is rolled back in the local server.

- **module** : rp
- **Category** : Fatal
- **Reproducibility** : Frequence
- **Description** : In a duplication environment that satisfies the conditions below, when a DDL statement executed on the local server is rolled back on the local server due to a problem with the remote server, an abnormal shutdown of the Altibase server caused by a memory error has been fixed.
  1. At least one of View, Trigger, Stored Procedure, and Package objects including replication target table exists
  2. 2. Use the DDL replication function (altibase server property REPLICATION_DDL_ENABLE setting value 1)
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**

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

BUG-49412 Reflects "PROJ-2751 Relaxation of Maximum Number of Host Variables".

- **module** : qp
- **Category** : Enhancement
- **Reproducibility** : Always
- **Description** : Fixed a bug where the error [The number of host variables exceed the maximum limit (1024).] occurs when the number of host variables used exceeds 1024.
- **How to reproduce this bug**
    - **Reproduction conditions**
    - **Actual Results**
    - **Expected Results**
- **Workaround**
- **Changes**
    - Performance view
    - Property
    - Compile Option
    - Error Code

BUG-49420 [ERR-11058: The row already exists in a unique index.] error occurs when executing a DDL statement.

- **module** : rp
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Fixed bug that [ERR-11058: The row already exists in a unique index.] error occurs when executing DDL statements when all of the conditions below are satisfied in a redundant environment.
    1. Start redundancy at least once
    2. Execute DDL statement on replication target table
    3. . Re-execution of the DDL statement before reading the DDL log previously performed by the replication sender
    4. The re-executed DDL statement reuses the table object identifier (TABLE OID) before executing the first DDL statement.
- **How to reproduce this bug**
    - **Reproduction conditions**

```
Create redundancy gap
Execute DDL statements on replication tables (truncate, etc.)
```

- **Actual Results**

  ```
  DDL statement failed with the following error message.
  [ERR-11058 : The row already exists in a unique index.]
  ```

- **Expected Results**

  ```
  DDL syntax success
  ```

- **Workaround**

```
Eliminate a redundancy gap by executing a redundant FLUSH statement before
executing a DDL statement
or
Before executing the DDL statement, check V$REPGAP to confirm that there is
no duplication gap before executing the DDL statement.
```

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

Changes

Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
7.1.0.6.4	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