Altibase 7.1.0.6.0 Patch Notes

Table of Contents

Fixed Bugs

BUG-49063 If DELETE and DEQUEUE occur simultaneously on the same record in the queue table, the Altibase server abnormally terminates.

- **module** : sm

- **Category** : Fatal

- **Reproducibility** : Rare

- **Description** : BUG-48230 (Altibase 7.1.0.4.8) has been modified to not catch X Latch on pages to improve QEUEUE performance. Later, it was discovered that the Altibase server abnormally terminated when DELETE and DEQUEUE were executed simultaneously. To avoid this problem, add a statement to set whether to allow DELETE statements in the queue table.

  - Added DELETE clause to CREATE QUEUE and ALTER QUEUE statements

    - DELETE ON

      - Allows DELETE statements on QUEUE tables.
      - DEQUEUE parallel execution performance is lower than when the DELETE statement is not allowed.

    - DELETE OFF

      - Do not allow DELETE statements on QUEUE tables.
      - DEQUEUE parallel execution performance improve than when the DELETE statement is allowed.

  A related V$QEUEUE performance view has been added and the error message when specifying a disk tablespace when creating a queue table has been changed.

- **How to reproduce this bug**

  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**

- **Changes**

  - Performance view

    - Added V$QEUEUE performance view.

    - ```
      iSQL> DESC V$QUEUE
      [ ATTRIBUTE ]

      ------------------------------------------------------------------
      ----------
      NAME                              TYPE

      ------------------------------------------------------------------
      ----------
      TABLE_OID                         BIGINT
      DELETE_ON                         VARCHAR(1)
      QUEUED                            BIGINT
      ```

- - - DELETE_ON
    - - T : DELETE can be executed. Due to lock wait during DEQUEUE, performance may be slower than when DELETE cannot be performed.
      - F : Unable to execute DELETE. DEQUEUE performance is faster than DELETE possible.
    - - QUEUED
      - - Number of items in queue
  - Property
  - Compile Option
  - Error Code
    - The error message when creating a queue table in a disk tablespace has been changed.

```
CREATE QUEUE Q1 ( 7 ) TABLESPACE SYS_TBS_DISK_DATA;
-[ERR-311E5 : The table is not a memory or volatile table.]
+[ERR-3147B : The QUEUE table cannot be used on disk tablespace.]
```

BUG-49242 QUEUE objects are not extracted in aexport object mode.

- **module** : ux-aexport
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Fixes a bug where QUEUE object is not extracted in aexport object mode.
- **How to reproduce this bug**
  - **Reproduction conditions**

```
iSQL> CREATE QUEUE q1(10);
Create success.
```

  - **Actual Results**

```
% aexport -s 127.0.0.1 -u sys -p manager -object sys.q1
The SYS.Q1 object not exist
```

  - **Expected Results**

```
% aexport -s 127.0.0.1 -u sys -p manager -object sys.q1
##### QUEUE #####
** "SYS"."Q1"
create queue "Q1" (10);
```

- **Workaround**
  - 전체 DB 모드/사용자 모드로 export
- **Changes**
  - Performance view

- Property
- Compile Option
- Error Code

BUG-49243 The call stack is not output to the altibase_error.log when using long SQL statements over 20000 bytes in length.

- **module** : id

- **Category** : Testcase Fail

- **Reproducibility** : Always

- **Description** : Fixes a bug where the call stack is truncated in altibase_error.log when a long SQL statement of more than 20000 bytes is used. This bug occurs from Altibase 7.1.0.5.2 with BUG-48433 reflected.

- **How to reproduce this bug**

    - **Reproduction conditions**
    - **Actual Results**
    - **Expected Results**
- **Workaround**

- **Changes**

    - Performance view
    - Property
    - Compile Option
    - Error Code

BUG-49271 The Altibase server may abnormally terminate in the process of reusing the hash temp table during transaction processing including disk tables.

- **module** : sm

- **Category** : Fatal

- **Reproducibility** : Always

- **Description** : Fixed a bug where an uninitialized page was accessed and the Altibase server abnormally terminated when reusing hash temp tables for temporary data processing during transaction processing including disk tables.

- **How to reproduce this bug**

    - **Reproduction conditions**

```
CREATE TABLE T10 (CIF int, AGE int) TABLESPACE SYS_TBS_DISK_DATA;

CREATE TABLE T1 (CIF int, ACNO int, DPNT_CLSN_Q int, DPNT_CNPR int,
ORD_DT int, BNS_TCD int ) TABLESPACE SYS_TBS_DISK_DATA;
CREATE TABLE T2 (CIF int, ACNO int) TABLESPACE SYS_TBS_DISK_DATA;
CREATE TABLE T3 (BAS_DT int, BIZDT_YN int, EXG_COD int ) TABLESPACE
SYS_TBS_DISK_DATA;
```

```sql
INSERT INTO T10 SELECT  MOD(LEVEL,50000), MOD(LEVEL,100) FROM DUAL
CONNECT BY LEVEL <=50000;
INSERT INTO T1  SELECT  7, 9, 1, 1, 5, 2 FROM DUAL CONNECT BY LEVEL
<=100;
INSERT INTO T2  SELECT  MOD(LEVEL,50000), 9  FROM DUAL CONNECT BY LEVEL
<=50000;
INSERT INTO T3  SELECT  1, 1, 0 FROM DUAL CONNECT BY LEVEL <=10;


ALTER SYSTEM SET HASH_AREA_SIZE = 3145728;


SELECT V1.CIF
     , CASE WHEN V1.AGE < 10 THEN '0' ELSE SUBSTR(V1.AGE, 1, 1) END AS
AGE_GROUP
     , V2.OPENRANK_TYPE
  FROM (
        SELECT CIF, AGE
          FROM T10
       ) V1
      ,(
        SELECT V98.CIF  AS CIF
             , CASE WHEN V98.CUT_LINE = 0 AND RANK = 1 THEN '10'
                    WHEN RANK <= V98.CUT_LINE THEN '10'
               ELSE NULL END AS OPENRANK_TYPE
          FROM
             (
               SELECT V99.CIF AS CIF
                    , V99.DPNT_CNPR AS DPNT_ACBK_PR
                    , TRUNC(COUNT(DISTINCT V99.CIF) OVER() * 0.1) AS
CUT_LINE
                    , RANK() OVER(ORDER BY V99.DPNT_CNPR DESC) AS RANK
                 FROM
                    (
                      SELECT T2.CIF  AS CIF
                           , SUM(T1.DPNT_CLSN_Q * T1.DPNT_CNPR) AS
DPNT_CNPR
                        FROM T1, T2
                       WHERE T1.ACNO= T2.ACNO
                         AND  T1.BNS_TCD= '2'
                         AND  T1.ORD_DT >= ( SELECT BAS_DT
                                              FROM T3
                                             WHERE BAS_DT <= 3
                                               AND BIZDT_YN = 1
                                               AND EXG_COD = 0
                                             ORDER BY BAS_DT DESC
                                           )
                       WHERE RK = 5
                    )
                      GROUP  BY T2.CIF
                ) V99
             ) V98
        ) V2
   WHERE V1.CIF = V2.CIF
     AND V2.OPENRANK_TYPE IS NOT NULL  ;
```

- **Actual Results**

```
[ERR-91015 : Communication failure.]
```

- **Expected Results**

```
50000 rows 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
7.1.0.6.0	6.5.1	8.9.1	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

- V$QEUEUE

Changed performance views

Deleted preformance views