

- Altibase 7.1.0.2.0 Patch Notes
 - Fixed Bugs
 - BUG-46502 IPCDA PowerPC 안정성 개선
 - BUG-46594 이중화 핸드쉐이크 과정에서 불필요한 lock 이 잡힐 수 있습니다.
 - BUG-46644 XA에서 메모리 릭이 발생할 수 있습니다.
 - BUG-46666 LOB 업데이트시 메모리 테이블 사이즈가 계속 증가합니다.
 - BUG-46678 Return의 Precision을 명시하지 않은 function에서 return 값을 저장하는 buffer의 크기를 잘못 계산할 수 있습니다.
 - BUG-46679 disk table 에서 특정 계층쿼리 수행시 [ERR-311A4 : Loop in hierarchical query detected.]
 - BUG-46709 AIX에서 server stop시 비정상적으로 종료되는 문제 수정
 - Changes
 - Version Info
 - 호환성
 - 프로퍼티
 - 성능 뷰

Altibase 7.1.0.2.0 Patch Notes

Fixed Bugs

BUG-46502 IPCDA PowerPC 안정성 개선

- **module** : mm
- **Category** : Memory Error
- **재현 빈도** : Always
- **증상** : PowerPC 에서 IPCDA Connect Type으로 동작하는 경우의 안정성 개선
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property

- Compile Option
- Error Code

BUG-46594 이중화 핸드쉐이크 과정에서 불필요한 lock 이 잡힐 수 있습니다.

- **module** : rp
- **Category** : Other
- **재현 빈도** : Always
- **증상** : 이중화 핸드쉐이크 과정에서 파티션이 없는 경우에도 파티션 비교를 위한 락을 잡는 경우가 있습니다. 파티션이 없는 경우 파티션 비교를 하지 않도록 수정하여 불 필요한 락을 잡지 않도록 하였습니다.
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property
 - Compile Option
 - Error Code

BUG-46644 XA에서 메모리 릭이 발생할 수 있습니다.

- **module** : mm-xa
- **Category** : Memory Error
- **재현 빈도** : Rare
- **증상** : 서버 시작시 복구할 XA 트랜잭션이 있는 경우 트랜잭션 초기화를 2번 수행해 메모리 릭이 발생합니다. 초기화를 1번만 하도록 수정하였습니다.
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property

- Compile Option
- Error Code

BUG-46666 LOB 업데이트시 메모리 테이블 사이즈가 계속 증가합니다.

- **module** : sm
- **Category** : Other
- **재현 빈도** : Always
- **증상** : 메모리 테이블의 LOB컬럼 UPDATE시
메모리 테이블 사이즈가 계속 증가할 수 있는 문제가 있어서
수정하였습니다.
- **재현 방법**
 - 재현 절차

```
drop table test;
create table test (c1 clob);
insert into test values ('aaa');
insert into test values (lpad('b', 5000) );
select mem_page_cnt, mem_var_page_cnt from system.sys_tables_ a, v$membtbl_info b where
update test set c1 = CASE WHEN c1 = 'aaa' then lpad('b', 5000) else 'aaa' end;
update test set c1 = CASE WHEN c1 = 'aaa' then lpad('b', 5000) else 'aaa' end;
update test set c1 = CASE WHEN c1 = 'aaa' then lpad('b', 5000) else 'aaa' end;
exec sleep(4);
update test set c1 = CASE WHEN c1 = 'aaa' then lpad('b', 5000) else 'aaa' end;
update test set c1 = CASE WHEN c1 = 'aaa' then lpad('b', 5000) else 'aaa' end;
update test set c1 = CASE WHEN c1 = 'aaa' then lpad('b', 5000) else 'aaa' end;
exec sleep(4);
update test set c1 = CASE WHEN c1 = 'aaa' then lpad('b', 5000) else 'aaa' end;
update test set c1 = CASE WHEN c1 = 'aaa' then lpad('b', 5000) else 'aaa' end;
update test set c1 = CASE WHEN c1 = 'aaa' then lpad('b', 5000) else 'aaa' end;
exec sleep(4);
```

```
select mem_page_cnt, mem_var_page_cnt from system.sys_tables_ a, v$membtbl_info b where
```

◦ 수행 결과

마지막 SELECT 결과

==>

```
isQL> select mem_page_cnt, mem_var_page_cnt from system.sys_tables_ a, v$membtbl_info b
MEM_PAGE_CNT          MEM_VAR_PAGE_CNT
```

```
-----
1                      2
1 row selected.
```

◦ 예상 결과

마지막 SELECT 결과

===>

```
iSQL> select mem_page_cnt, mem_var_page_cnt from system_.sys_tables_ a, v$memtbl_info
MEM_PAGE_CNT          MEM_VAR_PAGE_CNT
-----
1                      1
1 row selected.
```

- **Workaround**

- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

BUG-46678 Return의 Precision을 명시하지 않은 function에서 return 값을 저장하는 buffer의 크기를 잘못 계산할 수 있습니다.

- **module** : qp-psm-trigger-execute
- **Category** : Fatal
- **재현 빈도** : Always
- **증상** : Return의 precision을 명시하지 않은 function에서 return 값을 저장하는 buffer 크기를 잘못 계산해서 비정상 동작하는 문제를 수정했습니다.
- **재현 방법**
 - 재현 절차

```

drop table "T_TCCJ133_M01";
create table "T_TCCJ133_M01"
(
    "MRST_ID" NUMERIC(7) not null,
    "OPCP_HDQR_ID" VARCHAR(3) not null
)
tablespace sys_tbs_disk_data;
insert into "T_TCCJ133_M01" values ( 801901, 901);
insert into "T_TCCJ133_M01" values ( 801901, 901);
insert into "T_TCCJ133_M01" values ( 801701, 701);
insert into "T_TCCJ133_M01" values ( 199001, 001);
insert into "T_TCCJ133_M01" values ( 801701, 999);
drop table "T_TCCJ124_M01";
create table "T_TCCJ124_M01"
(
    "OPCP_HDQR_ID" VARCHAR(3) not null,
    "OPCP_HDQR_NM" VARCHAR(100) not null
)
tablespace sys_tbs_disk_data;
insert into "T_TCCJ124_M01" values ( 701,'Charging Station Group 1');
insert into "T_TCCJ124_M01" values ( 901,'M3 Group 1');
insert into "T_TCCJ124_M01" values ( 001,'c1');
insert into "T_TCCJ124_M01" values ( 002,'c2');
CREATE OR REPLACE FUNCTION FTCC_GET_HDQR_NM(P_HDQR_ID IN NUMBER)
RETURN VARCHAR2 DETERMINISTIC
IS
    V_HDQR_NM VARCHAR2(100) := NULL; --오류시 NULL을 반환한다.
BEGIN
    SELECT OPCP_HDQR_NM INTO V_HDQR_NM FROM T_TCCJ124_M01
    WHERE OPCP_HDQR_ID = P_HDQR_ID;
    RETURN V_HDQR_NM;
EXCEPTION
WHEN OTHERS THEN
    RETURN NULL;
END;
/
SELECT /*+ DISTINCT_SORT */ DISTINCT MRST_ID AS MRST_ID
    , OPCP_HDQR_ID AS OPCP_HDQR_ID
    , FTCC_GET_HDQR_NM(OPCP_HDQR_ID) AS OPCP_HDQR_NM
    FROM T_TCCJ133_M01;
[ERR-91015 : Communication failure.]
SELECT /*+ DISTINCT_HASH */ DISTINCT MRST_ID AS MRST_ID
    , OPCP_HDQR_ID AS OPCP_HDQR_ID
    , FTCC_GET_HDQR_NM(OPCP_HDQR_ID) AS OPCP_HDQR_NM
    FROM T_TCCJ133_M01;
[ERR-91015 : Communication failure.]

```

- 수행 결과
- 예상 결과

```

drop table "T_TCCJ133_M01";
create table "T_TCCJ133_M01"
(
    "MRST_ID" NUMERIC(7) not null,
    "OPCP_HDQR_ID" VARCHAR(3) not null
)
tablespace sys_tbs_disk_data;
insert into "T_TCCJ133_M01" values ( 801901, 901);
insert into "T_TCCJ133_M01" values ( 801901, 901);
insert into "T_TCCJ133_M01" values ( 801701, 701);
insert into "T_TCCJ133_M01" values ( 199001, 001);
insert into "T_TCCJ133_M01" values ( 801701, 999);
drop table "T_TCCJ124_M01";
create table "T_TCCJ124_M01"
(
    "OPCP_HDQR_ID" VARCHAR(3) not null,
    "OPCP_HDQR_NM" VARCHAR(100) not null
)
tablespace sys_tbs_disk_data;
insert into "T_TCCJ124_M01" values ( 701,'Charging Station Group 1');
insert into "T_TCCJ124_M01" values ( 901,'M3 Group 1');
insert into "T_TCCJ124_M01" values ( 001,'c1');
insert into "T_TCCJ124_M01" values ( 002,'c2');
CREATE OR REPLACE FUNCTION FTCC_GET_HDQR_NM(P_HDQR_ID IN NUMBER)
RETURN VARCHAR2 DETERMINISTIC
IS
    V_HDQR_NM VARCHAR2(100) := NULL; --오류시 NULL을 반환한다.
BEGIN
    SELECT OPCP_HDQR_NM INTO V_HDQR_NM FROM T_TCCJ124_M01
    WHERE OPCP_HDQR_ID = P_HDQR_ID;
    RETURN V_HDQR_NM;
EXCEPTION
WHEN OTHERS THEN
    RETURN NULL;
END;;
/
SELECT /*+ DISTINCT_SORT */ DISTINCT MRST_ID AS MRST_ID
    , OPCP_HDQR_ID AS OPCP_HDQR_ID
    , FTCC_GET_HDQR_NM(OPCP_HDQR_ID) AS OPCP_HDQR_NM
    FROM T_TCCJ133_M01;
4 rows selected.

```

- **Workaround**

```

CREATE OR REPLACE FUNCTION FTCC_GET_HDQR_NM(P_HDQR_ID IN NUMBER)
RETURN VARCHAR2 DETERMINISTIC
IS
    V_HDQR_NM VARCHAR2(100) := NULL; --오류시 NULL을 반환한다.
BEGIN
    SELECT OPCP_HDQR_NM INTO V_HDQR_NM FROM T_TCCJ124_M01
    WHERE OPCP_HDQR_ID = P_HDQR_ID;
    RETURN V_HDQR_NM;
EXCEPTION
WHEN OTHERS THEN
    RETURN V_HDQR_NM; -- NULL 대신 NULL 값을 갖는 변수를 return 한다.
END;
/

```

- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

BUG-46679 disk table 에서 특정 계층쿼리 수행시 [ERR-311A4 : Loop in hierarchical query detected.]

- **module** : qp
- **Category** : Functional Error
- **재현 빈도** : Always
- **증상** : disk table에서 join 과 같이 계층 쿼리가 사용될 경우 발생하는 오류 수정
- **재현 방법**
 - 재현 절차

```

create table t1 ( i1 varchar(10), i2 varchar(10), i3 varchar(10) ) tablespace sys_tbs_d
Insert into t1 values( 'a', 2, null );
insert into t1 values( 'b', 3, 'a' );
insert into t1 values( 'c', 4, 'b' );
insert into t1 values( 'd', 5, 'c' );
create table t2 ( i1 varchar(10), i2 varchar(10), i3 varchar(10) ) tablespace sys_tbs_d
Insert into t2 values( null, 1, 'a' );
insert into t2 values( 'a', 2, 'b' );
insert into t2 values( 'b', 3, 'c' );
insert into t2 values( 'c', 4, 'd' );
select t1.i1, cast( max(sys_connect_by_path(t1.i1, '/')) as varchar(20)) path, max(conn
from t1 left outer join t2 on t1.i2 = t2.i2 + 1
connect by prior t2.i1 = t1.i1
group by t1.i1;
select t1.i1, cast( max(sys_connect_by_path(t1.i1, '/')) as varchar(20)) path, max(conn
from t1 left outer join t2 on t1.i2 = t2.i2 + 1
connect by prior t2.i1 = t1.i1
group by t1.i1
order by t1.i1;
select t1.i1, cast( max(sys_connect_by_path(t1.i1 || ( select t1.i1 from dual ), '/'))
, max(connect_by_root(t1.i1 || ( select t1.i1 from dual)))
from t1 left outer join t2 on t1.i2 = t2.i2 + 1
connect by prior t2.i1 = t1.i1
group by t1.i1
order by t1.i1;
select t1.i1, cast( sys_connect_by_path(t1.i1 || ( select t1.i1 from dual ), '/') as va
, connect_by_root(t1.i1 || ( select t1.i1 from dual)), connect_by_isleaf
from t1 left outer join t2 on t1.i2 = t2.i2 + 1
connect by prior t2.i1 = t1.i1
order by t1.i1;

```

- 수행 결과
- 예상 결과

• Workaround

TEMP_TBS_MEMORY hint 사용

• 변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-46709 AIX에서 server stop시 비정상적으로 종료되는 문제 수정

- **module** : id
- **Category** : Fatal
- **재현 빈도** : Always
- **증상** : Altibase 7.1.0.1.5 버전 이후부터 발생했던 문제로, AIX에서 server stop시 비정상적으로 종료되던 문제를 수정하였습니다.
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property
 - Compile Option
 - Error Code

Changes

Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version	sharding version
7.1.0.2.0	6.5.1	8.7.1	7.1.6	7.4.4	2.2.1

Altibase 7.1 패치 버전별 히스토리는

[Version_Histories](#)

에서 확인할 수 있다.

호환성

Database binary version

데이터베이스 바이너리 버전은 변경되지 않았다.

데이터베이스 바이너리 버전은 데이터베이스 이미지 파일과 로그파일의 호환성을 나타낸다. 이 버전이 다른 경우의 패치(업그레이드 포함)는 데이터베이스를 재구성해야 한다.

Meta Version

메타 버전은 변경되지 않았다.

패치를 롤백하려는 경우,
[메타다운그레이드](#)를
참고한다.

CM protocol Version

통신 프로토콜 버전은 변경되지 않았다.

IPCDА 연결 속성을 이용하는 경우, 7.1.0.2.0 이하버전과 호환성을 보장하지 않는다. 따라서,
IPCDА 를 이용하는 경우, 서버와 클라이언트를 7.1.0.2.0 이상 버전으로 패치해야 한다.

Replication protocol Version

Replication 프로토콜 버전은 변경되지 않았다.

Sharding Version

샤딩 버전이 2.1.0 에서 2.2.1로 변경되었다.

알티베이스 샤딩 프로토콜 및 메타는 상위, 하위 호환성을 보장하지
않는다. 즉, 샤딩 버전이 다른 경우, 재구성해야 한다. *샤딩 설정은 [Altibase Sharding 설치와 설정](#)
을 참고한다.*

프로퍼티

추가/변경/삭제된 프로퍼티 없음

성능 뷰

추가/변경/삭제된 성능 뷰 없음