Table of Contents

Altibase®

New Features Guide

Altibase New Features Guide

Release 7.2

㈜알티베이스

08378 서울시 구로구 디지털로 306 대륭포스트타워II 10층

전화: 02-2082-1114 팩스: 02-2082-1099

고객서비스포털: http://support.altibase.com

homepage: http://www.altibase.com

Altibase 7.2.0.0.1의 새로운 기능 및 특징

Altibase 버전 7.2.0.0.1에서 사용할 수 있는 새로운 기능들을 소개한다.

기능 개선

SQL 확장

CREATE QUEUE 및 ALTER QUEUE 구문에 DELETE 절 추가

큐(QUEUE) 테이블에 DELETE 문 허용 여부를 설정하는 DELETE 절이 추가되었다.

DELETE OFF로 DELETE 문을 허용하지 않으면 DELETE 문을 허용한 경우보다 DEQUEUE 병렬 수행 성능이 향상된다. 구문 사용 방법은 Altibase 7.2 SQL Reference 매뉴얼 을 참고한다. 관련하여 성능 뷰 V$QUEUE_DELETE_OFF가 추가되었다.

범위 파티션드 객체에 파티션 추가 연산 지원

범위 파티셔드 테이블에 파티션 추가(ADD PARTITION) 구문을 지원한다. 이 기능 추가로 기본 파티션 없는 범위 파티션드 테이블 생성이 가능하다.

Altibase 7.2 에서 범위 파티션드 테이블 생성 시 주의 사항

- Altibase 7.2 에서는 기본 파티션이 없는 범위 파티션드 테이블을 생성할 수 있다.

  참고로 기본 파티션이 없는 범위 파티션드 테이블을 생성하면 SYS_TABLE_PARTITIONS_에서 PARTITION_NAME 이 없는 파티션이 추가로 생성된다.
- 범위 파티션드 객체에서 파티션 추가는 기본 파티션이 없는 범위 파티션드 테이블에서만 사용할 수 있다.
- 기본 파티션이 없는 범위 파티션드 테이블은 기본 파티션 추가 연산을 수행할 수 없다.
- 기본 파티션이 있는 범위 파티션드 테이블은 기본 파티션 삭제 연산을 수행할 수 없다.

- 기본 파티션이 없는 범위 파티션드 테이블은 CONJOIN/DISJOIN 구문을 사용할 수 없다.
- 범위 파티션드 테이블이 이중화 대상 테이블인 경우 파티션 추가 연산을 수행할 수 없다.

응용 프로그램 개발 인터페이스

JDBC API Specification 4.2 부분 지원 (PROJ-2707)

Altibase 7.2 에서 JDBC API Specification 4.2를 부분적으로 지원한다.

Altibase 7.2 JDBC 드라이버는 JRE 1.8 이상에서 동작한다. Altibase 7.2 JDBC 드라이버에서 지원하는 JDBC 4.2 API는 Altibase 7.2 JDBC User's Manual 에서 확인할 수 있다. 변경 사항 및 호환성 이슈는 Altibase JDBC 7.2 변경 사항 및 호환성 이슈에서 확인할 수 있다.

- **Auto-loading of JDBC driver class**

  명시적으로 Class.forName() 클래스를 로딩할 필요없이 META-INF/services/java.sql.Driver 파일을 이용한 자동 드라이버 로딩 기능 지원

- **Wrapper Pattern Support**

  프록시에서 구현 객체에 대한 참조를 얻는 JDBC 4.0 표준 인터페이스를 지원한다. 커넥션풀 등에서 생성하는 프록시 객체에서 JDBC 객체를 획득할 수 있다.

  ```
  try (Connection sWrappedCon = dbPool.getConnection()) {
      if (sWrappedCon.isWrapperFor(AltibaseConnection.class)) {
          AltibaseConnection connection =
  sWrappedCon.unwrap(AltibaseConnection.class);
          ...
          ...
  }
  ```

- **National Character Set Support**

  JDBC 4.0 스펙인 표준 다국어 처리 인터페이스 지원

- **Aborting Connections**

  비동기적으로 데이터베이스와의 물리적 연결을 종료하는 Connection.abort() 인터페이스 지원

- **Standard Socket Network Timeout API Support**

  데이터베이스 서버로부터 소켓 응답 대기 시간을 설정하는 표준 인퍼페이스 Connection.setNetworkTimeout() 지원

- **Connection Management Enhancements**

  Validation Query없이 Connection 객체에서 유효성 검사를 수행하는 Connection.isValid() 지원

- **Large Update Counts Support**

  대용량 레코드 업데이트를 위한 executeLargeUpdate(), executeLargeBatch() 지원

- **Set Client Information Support**

  Connection.setClientInfo()를 이용한 클라이언트 어플리케이션 속성(name) 설정 지원

- **java.sql.SQLType interface Support**

  JDBC 4.2 표준 인터페이스 java.sql.SQLType을 구현한 AltibaseJDBCType 지원

JDK 레벨에서 향상된 기능들은 Altibase JDBC 7.2 에서도 대부분 사용할 수 있다.

- Try-with-resources 구문을 통한 자동 JDBC 리소스 해제

```
try (Statement stmt = con.createStatement()) {
      ResultSet rs = stmt.executeQuery(query);
      while (rs.next()) {
        String coffeeName = rs.getString("aaa");
        int supplierID = rs.getInt("bbb");
      }
    }
}
```

- SQLException에 Enhanced for-each loop 사용

```
catch(SQLException ex) {
     for(Throwable e : ex ) {
        LOG.error("Error occurred: " + e);
     }
}
```

- 커넥션풀 등에서 생성되는 proxy객체에서 실체 JDBC 객체 획득

```
try (Connection sWrappedCon = dbPool.getConnection()) {
    if (sWrappedCon.isWrapperFor(AltibaseConnection.class)) {
        AltibaseConnection connection =
sWrappedCon.unwrap(AltibaseConnection.class);
        ...
        ...
}
```

유틸리티

altiComp 커밋 카운트 설정 기능 추가

커밋(commit) 카운트를 설정할 수 있는 프로퍼티 COUNT_TO_COMMIT가 추가되었다. 관련 내용은 Altibase 7.2 Utilities Manual 에서 확인할 수 있다.

성능 및 안정성 향상

OLTP Scalability 성능 향상(TASK-7073)

- Linux x86-64 CPU 코어 수 24코어 이상에서 조회 트랜잭션 성능 저하 현상 개선
- 메모리 DB 삭제(DELETE) 트랜잭션 성능 향상을 위해 로깅 구조 개선
- 디스크 DB 변경 트랜잭션 성능 향상을 위해 In-place MVCC 동작 방식 개선
- 테이블 잠금(TABLE LOCK) 병목 개선
- INSERT/UPDATE 트랜잭션 처리 시 불필요한 트랜잭션 로그 기록을 제거
- 트랜잭션 로그파일 압축 시 메모리 할당/해제 병목 개선
  - 이와 관련한 영향도 확인
- 휘발성(Volatile) 메모리 DB 트랜잭션 성능 향상
- 커밋 병목 및 가비지 콜렉션 쓰레드 병목 개선
  - 트랜잭션 커밋 후 테이블 정보 업데이트 병목 개선

- 메모리 DB 트랜잭션 성능 향상
    - 디스크 읽기를 유발하는 함수의 병목을 제거
    - Group Commit Log 기능 추가

트랜잭션 로그 기록 성능 향상(TASK-6983)

로그 압축 알고리즘을 압축 속도가 빠른 LZ4 로 변경하였다.

휘발성/비휘발성 메모리 DB 트랜잭션 성능 향상

메모리 테이블 객체 식별자 추적 단계를 간소화하여 휘발성/비휘발성 메모리 DB 트랜잭션 성능이 향상되었다.

언두(undo) 테이블스페이스 재사용 안정성 향상

언두 테이블스페이스와 디스크 인덱스의 불필요한 관계를 제거하여 버그 발생 위험 요소 제거하였다. 디스크 페이지 공간 효율 개선으로 관련 프로퍼티들의 기본값 및 최대값이 변경되었다.

- INDEX_INITTRANS 최대값이 30에서 50으로 변경
- INDEX_MAXTRANS 기본값과 최대값이 30에서 50으로 변경

PARTITIONED TABLE에 대한 LIMIT FOR UPDATE 성능 개선

서브쿼리의 인라인 뷰에 ORDER BY절 사용 시 SQL 성능 개선

조건절(WHERE, HAVING 절)에서 사용한 서브쿼리의 인라인뷰에 ORDER BY절이 있는 경우 OBYE(Order By Elimination, 불필요한 ORDER BY 제거) 쿼리 변환을 적용하여 SQL 성능이 향상되었다.

- SQL 사용 예

```
SELECT *
  FROM T1
 WHERE I1 IN (SELECT /*+ NO_UNNEST */I1
                FROM (SELECT *
                        FROM T2
                       ORDER BY I2, I3));
```

> 이 영향을 받는 *SQL*의 실행 계획에 변화가 있다. *SUBQUERY FILTER* 안에 *SORT* 플랜 노드 없어진다.

스칼라 서브쿼리(Scalar subquery) 성능 개선

스칼라 서브쿼리 결과가 단일 레코드인지 확인 과정에서 발생하는 오버헤드를 제거하여 성능을 개선하였다.

Altibase 이중화 성능 향상

이중화 Sender 성능 향상

- 압축 로그에서 이중화에 필요한 로그만 압축 해제하는 기능 추가
- xLog 압축 알고리즘을 LZO에서 LZ4로 변경

기타

프로퍼티 추가

- DBLINK_GLOBAL_TRANSACTION_LEVEL

메타 테이블 추가

- SYS_REPL_TABLE_OID_IN_USE_

성능 뷰 추가

- V$REPL_REMOTE_META_INDEX_COLUMNS
- V$QUEUE_DELETE_OFF