

java 下执行mysql 批量插入的几种方法及用时

java 下执行mysql 批量插入的几种方法及用时,1000次插入方法的比较。

方法1：

Java code

复制代码 代码如下:

```
conn = DriverManager.getConnection(JDBC_URL, JDBC_USER, JDBC_PASS);
pstmt = conn
    .prepareStatement("insert into loadtest (id, data) values (?, ?)");
for (int i = 1; i <= COUNT; i++) {
    pstmt.clearParameters();
    pstmt.setInt(1, i);
    pstmt.setString(2, DATA);
    pstmt.execute();
}
```

MyISAM：246.6秒、InnoDB：360.2秒

方法2：使用事务，不自动commit

Java code

复制代码 代码如下:

```
conn = DriverManager.getConnection(JDBC_URL, JDBC_USER, JDBC_PASS);
conn.setAutoCommit(false);
pstmt = conn
    .prepareStatement("insert into loadtest (id, data) values (?, ?)");
for (int i = 1; i <= COUNT; i++) {
    pstmt.clearParameters();
    pstmt.setInt(1, i);
    pstmt.setString(2, DATA);
    pstmt.execute();
    if (i % COMMIT_SIZE == 0) {
        conn.commit();
    }
}
conn.commit();
```

InnoDB：31.5秒

方法3：executeBatch

Java code

复制代码 代码如下:

```
conn = DriverManager.getConnection(JDBC_URL
    + "?rewriteBatchedStatements=true", JDBC_USER, JDBC_PASS);
conn.setAutoCommit(false);
pstmt = conn
    .prepareStatement("insert into loadtest (id, data) values (?, ?)");
for (int i = 1; i <= COUNT; i += BATCH_SIZE) {
    pstmt.clearBatch();
    for (int j = 0; j < BATCH_SIZE; j++) {
        pstmt.setInt(1, i + j);
        pstmt.setString(2, DATA);
        pstmt.addBatch();
    }
    pstmt.executeBatch();
    if ((i + BATCH_SIZE - 1) % COMMIT_SIZE == 0) {
        conn.commit();
    }
}
conn.commit();
```

InnoDB : 5.2秒

上面的使用时必须

1) rewriteBatchedStatements=true

2) useServerPrepStmts=true

方法4 : 先LOAD再COMMIT

Java code

复制代码 代码如下:

```
conn = DriverManager.getConnection(JDBC_URL, JDBC_USER, JDBC_PASS);
conn.setAutoCommit(false);
pstmt = conn.prepareStatement("load data local infile " "
    + "into table loadtest fields terminated by ','");
StringBuilder sb = new StringBuilder();
for (int i = 1; i <= COUNT; i++) {
    sb.append(i + "," + DATA + "\n");
    if (i % COMMIT_SIZE == 0) {
        InputStream is = new ByteArrayInputStream(sb.toString()
            .getBytes());
        ((com.mysql.jdbc.Statement) pstmt)
            .setLocalInfileInputStream(is);
        pstmt.execute();
        conn.commit();
        sb.setLength(0);
    }
}
InputStream is = new ByteArrayInputStream(sb.toString().getBytes());
((com.mysql.jdbc.Statement) pstmt).setLocalInfileInputStream(is);
pstmt.execute();
conn.commit();
```