Atitit 大批量百万数据循环oom对解决方案jdbc 流式处理

目录

[1.1. 流式处理这个方案性能最好了。。 1](#_Toc3994)

[2. 服务器游标模式useCursorFetch=true 性能也还凑活 2](#_Toc6111)

[2.1. 性能对比 2](#_Toc7746)

可能需要配合调整net read timeout参数，长连接性能最好，短连接吞吐量大。。

## 流式处理这个方案性能最好了。。

需要满足三个条件

ResultSet.***TYPE\_FORWARD\_ONLY***,

ResultSet.***CONCUR\_READ\_ONLY***

statement.setFetchSize(Integer.***MIN\_VALUE***);

**private** **static** **void** queryStream(Connection conn, String sql, Consumer<Map> consumer)

**throws** SQLException {

PreparedStatement statement = conn.prepareStatement(sql, ResultSet.***TYPE\_FORWARD\_ONLY***,

ResultSet.***CONCUR\_READ\_ONLY***);

statement.setFetchSize(Integer.***MIN\_VALUE***);

**long** begin = System.*currentTimeMillis*();

ResultSet resultSet = statement.executeQuery();

**while** (resultSet.next()) {

Map<String, Object> hm = *rs2map*(resultSet);

consumer.accept(hm);

// System.out.println(resultSet.getString("number"));

}

}

# 服务器游标模式useCursorFetch=true 性能也还凑活

需要在url连接处增加&useCursorFetch=true，以及客户端设置setFetchSize

//neeed url add usecursor ,,,and client use setFetchSize ...if not use fetchsize will oom...

**private** **static** **void** queryStreamCurser(Connection conn, String sql, Consumer<Map> consumer)

**throws** SQLException {

PreparedStatement statement = conn.prepareStatement(sql);

statement.setFetchSize(500); //

**long** begin = System.*currentTimeMillis*();

ResultSet resultSet = statement.executeQuery();

**while** (resultSet.next()) {

Map<String, Object> hm = *rs2map*(resultSet);

consumer.accept(hm);

// System.out.println(resultSet.getString("number"));

}

}

## 性能对比

百万数据循环，20s内，stream 450 rec..... server curser mode,400 rec...