

决赛题目：性能优化

题目描述：

本题目要求参赛队伍在系统运行正确的基础上，对查询执行、存取管理、事务处理等技术进行优化，尽可能提高系统在 TPC-C 测试基准下的吞吐。测试包括两个部分：(1) 正确性测试。参赛队伍提交的系统必须通过正确性测试。(2) 性能测试。性能测试为加载数据测试和并发事务测试。

1、在决赛过程中允许参赛队伍对提交的系统进行优化，这种优化可以不局限于功能题目中要求的实现技术，例如：可以使用其他死锁预防/死锁检测与消除策略来替换 wait-die。注意：TPC-C 可能会运行较长的时间，在整个测试过程中，要求系统不能出错且事务执行要求满足可串行化隔离级别，否则该题性能分数不得分。

2、在完成性能测试之前，参赛队伍需要完成 load 指令，把表的数据插入到系统中。load 指令格式如下：

```
load file_name into table_name;
```

示例：load ../../src/test/performance_test/table_data/warehouse.csv into warehouse;

其中，file_name 为数据文件的相对于数据库文件夹的相对路径，数据文件在测试时存放于 src/test/performance_test/table_data 文件夹下，在测试脚本中，通过 ./bin/rmdb 启动服务器，因此数据库文件夹位于 build 文件夹下。

load_data 指令不需要包含建表操作。

3、在完成性能测试之前，参赛队伍需要补充 update 语句功能，使得 update 语句支持如下表达式操作：

```
update student set score=score+5.5 where id<3;
```

4、为了减少将结果集写入文件带来的成本开销，参赛队伍需要在接收到“set output_file off”的命令后，停止向 output.txt 文件写入输出结果，在未接收到该命令时，默认需要开启向 output.txt 文件写入结果集的功能。

注意：“set output_file off”命令没有分号。

5、测试程序中包含如下九张表：

- warehouse 表，记录仓库相关信息，表中有 W 条记录；
- item 表，记录商品信息，表中有 10 万条记录；
- stock 表，记录每个仓库的商品库存数据，表中有 W*10 万条记录；
- district 表，记录每个仓库提供服务的地区，表中有 W*10 条记录；
- customer 表，记录每个地区的客户信息，表中有 W*10*3000 条记录；
- history 表，记录客户的交易历史，表中有 W*10*3000 条记录；
- order 表，记录每个地区的订单数据，表中有 W*10*3000 条记录；
- new_orders 表，记录新订单信息，Order 表中每个地区的最后 900 条订单被添加到 NewOrder 表中；

- order_line 表，每个订单随机生成 5-15 条 OrderLine 记录；
- 其中 $W \leq 50$ ；

6、测试程序包含如下五种事务：

- NewOrder：新订单的生成
- Payment：订单付款
- OrderStatus：最近订单查询
- Delivery：配送
- StockLevel：库存缺货状态分析

在测试程序中，会对每张表的主键建立索引，测试事务中包含主键字段上的单点查询、范围查询和非主键字段上的单点查询、范围查询。

各个事务中可能出现的 SQL 参考大赛仓库 db2024 中 rmdb/决赛性能测试 SQL 示例.pdf 文件。

7、性能测试包含两个阶段：

- 第一阶段：加载数据，通过 load 指令，把测试程序中包含的九张表的数据导入到系统中。**此阶段限时 240 秒。**
- 第二阶段：并发执行读写事务，事务占比如下：10/23 new_order，10/23payment，1/23order-status，1/23delivery，1/23stock-level。受制于平台资源，每支队伍提交的系统，其 TPC-C 负载测试最长持续时间为 1200 秒。

测试说明：

1、最终返回三个结果，分别为 new_order_txn，running_time，tpmC，分别代表第二阶段完成的 new order 事务数量，第二阶段事务的总运行时间，TPC-C 测试中系统最大有效吞吐；

2、排名按 TPC-C 测试吞吐指标 tpmC 来进行排名；

3、在进行性能测试之前，会对功能正确性进行测试，通过正确性测试后再进行性能测试，正确性测试包括题目二、题目三、题目四、题目七、题目十(不涉及 checkpoint 测试样例)；

4、第一阶段 load data 命令测试时，首先使用 select count(*) 语句对表中记录的数量进行测试，然后对表中数据进行一致性检验，通过这两个检验后才能通过 load 命令的测试；

5、第一阶段 load data 为性能通过性测试，在规定时间内完成数据加载并通过测试才可以进行第二阶段 TPC-C 性能测试。load data 实际执行的时间不计入排名。

6、第二阶段测试程序会运行一定数量的事务(假设 new order 事务为 M)，如果在规定时间内完成所有事务，则以实际完成时间(假设完成时间为 t 分钟)计算吞吐，吞吐计算公式为 $tpmc=M/t$ ；如果超时(假设超时时间为 T 分钟)，则以实际完成事务数量(假设 new order 事务为 m)计算吞吐，吞吐计算公式为 $tpmc=m/T$ 。

7、第二阶段 TPC-C 负载测试执行完成后，测试程序会对数据一致性进行检验，数据一致性的检验规则参考大赛仓库 db2024 中 rmdb/数据一致性检验规则.pdf

文件;

8、评测机的内存为 8GB，测试程序最大运行时间为 2400s;

9、评测机 CPU 为 8 核，进行性能评测时，同时有 16 个客户端向服务端发起连接;

10、如果在第一阶段 load data 超时，系统会输出 "In performance test, server timeout in load_data phase";

11、如果在第一阶段或第二阶段的事务运行过程中发生代码崩溃，系统会输出 "In performance test, server stops running when processing tpcc transactions in the load_data/tpcc phase.";

12、如果成功执行了第二阶段的事务但一致性检验阶段发生代码崩溃，系统会输出 "In performance test, server stops running when checking consistency." 并打印一致性检验的相关信息;

13、如果成功执行了第二阶段的事务但未通过一致性检验（并且未发生代码崩溃），系统会输出 "Consistency check for xxxx failed";

14、只有在规定的时间内完成数据加载，并且在第二阶段正确执行事务，通过数据一致性检验后，才能获得本题目的分数。