Week 14: TPCC benchmark on SQLite DB Engine

Siyeol Choi
2019315530

1. INTRODUCTION

In this lab experiment, we aim to experiment with SQLite and TPCC benchmark. By manipulating its journal mode, we would conduct experiments on TPS variation along each transations including DELIVERY, NEW_ORDER, ORDER_STATUS, PAYMENT, STOCK_LEVEL.

2. METHODS

First, we start with setting up the TPCC benchmark environment. Then we vary the journal mode varying between WAL and DELETE with fixed warehouse of 10 and duration of 1800. Upon completion of each iteration, we should re prepare the database file and flush all cache. After finishing the experiment, we compare the TPS changes and analyze the executed time/ time spent/ and rate of each journal mode.

Performance Evaluation

3.1 Experimental Setup

[System Setup]

OS Ubuntu 20.04.1 LTS (GNU/Linux 4.19.104-microsoft-standard

cpu cores 4

Vendor: Msft

Product: Virtual Disk

Revision: 1.0

Compliance: SPC-3

User Capacity: 274,877,906,944 bytes [274 GB]

Logical block size: 512 bytes

Physical block size: 4096 bytes

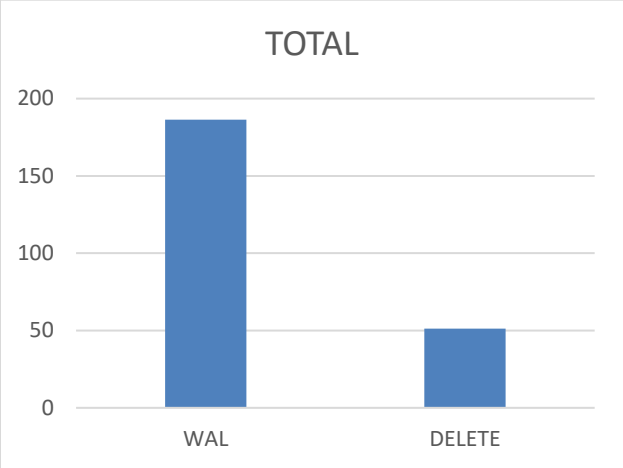
Type Configuration

3.2 Experimental Results

[Table]

	WAL	DELETE
DELIVERY	115.91	36.21
NEW_ORDER	156.66	43.95
ORDER_STATUS	1920.79	839.2
PAYMENT	206.17	53.79
STOCK_LEVEL	870.1	553.79
TOTAL	186.3	51.33

[Graph]



4. Conclusion

TPS (txn/s) is much higher in WAL (Write-Ahead Logging) journal mode compared to DELETE (which is RBJ). The overall performance along each transaction DELIVERY, NEW_ORDER, ORDER_STATUS, PAYMENT, STOCK_LEVEL is higher in WAL compared to DELETE.

The reason is as follows, Deleting RBJ file is expensive, so WAL is faster in most cases since it uses significantly fewer fsync() operations and Disk I/O operations become more sequential. However, WAL is not the best optimal when it is read intensive due to its searching overhead. Therefore, in STOCK_LEVEL where reading consists the most part, the difference between DELETE and WAL is comparably low.

5. REFERENCES

[1] PyTPCC https://github.com/apavlo/py-tpcc

[2] SQLite Pragma Cache Size https://www.sqlite.org/pragma.html#pragma_cache_size