

## Team 16 Assignment 4

109062123 曹瀚文 109062323 吳聲宏 109062330 黃鈺臻

實驗規格: Intel Core i5-10300H CPU @ 2.50GHz, 12GB RAM, 2TB SSD, Windows 11

### 1. 實作:

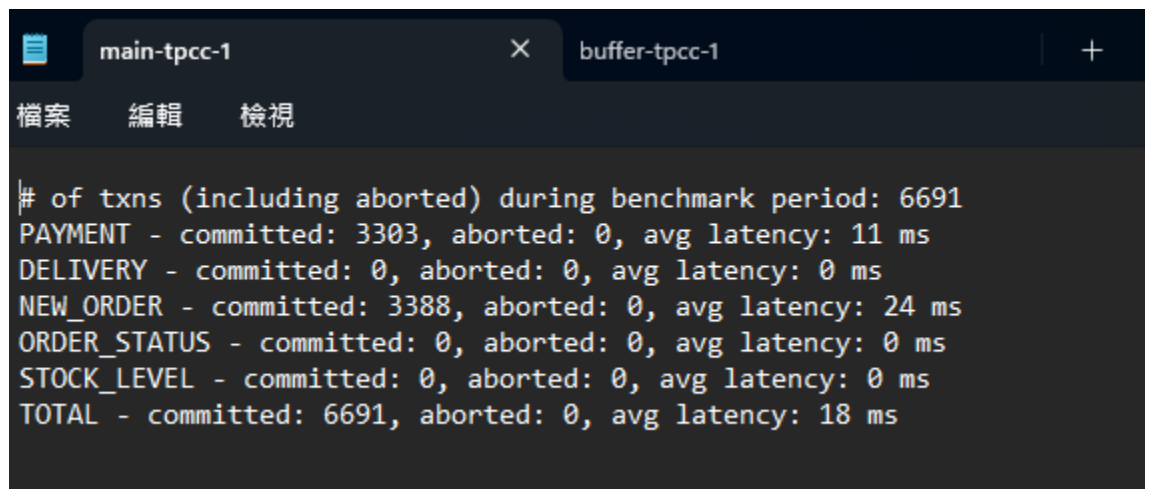
- 我們將Buffer以及BufferPoolMgr相關的synchronized functions的critical sections減少和縮短, 因為我們認為有可能因為threads之間要等待的話, 減少critical sections的大小也可以減少別的threads需要等待的時間。

以下是對Buffer的部分做optimize之後, 與原本的進行比較:

比較版本為tpc-c

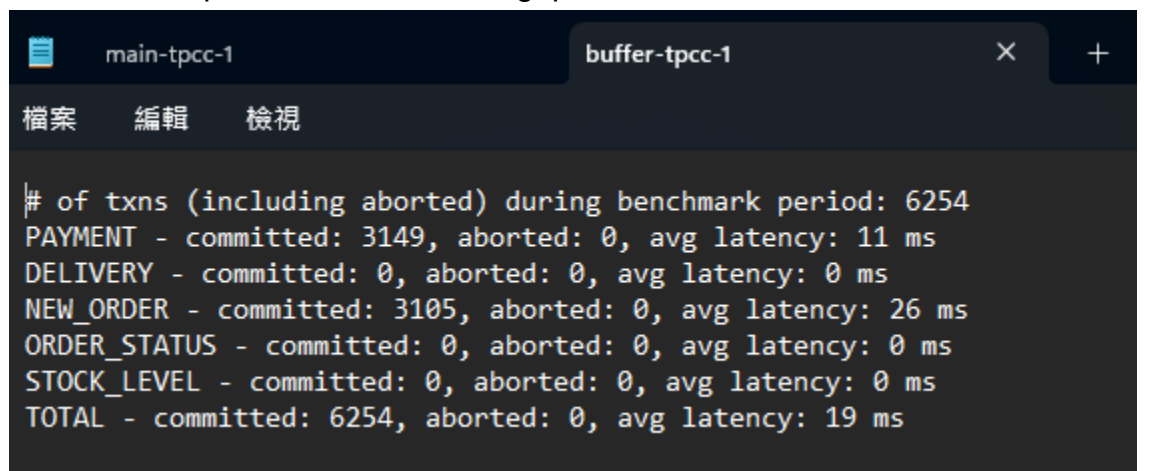
- BUFFER\_POOL\_SIZE=102400
- NUM\_WAREHOUSES=1

沒有任何optimization: total throughput = 6691



```
# of txns (including aborted) during benchmark period: 6691
PAYMENT - committed: 3303, aborted: 0, avg latency: 11 ms
DELIVERY - committed: 0, aborted: 0, avg latency: 0 ms
NEW_ORDER - committed: 3388, aborted: 0, avg latency: 24 ms
ORDER_STATUS - committed: 0, aborted: 0, avg latency: 0 ms
STOCK_LEVEL - committed: 0, aborted: 0, avg latency: 0 ms
TOTAL - committed: 6691, aborted: 0, avg latency: 18 ms
```

Buffer相關的optimization: total throughput = 6254



```
# of txns (including aborted) during benchmark period: 6254
PAYMENT - committed: 3149, aborted: 0, avg latency: 11 ms
DELIVERY - committed: 0, aborted: 0, avg latency: 0 ms
NEW_ORDER - committed: 3105, aborted: 0, avg latency: 26 ms
ORDER_STATUS - committed: 0, aborted: 0, avg latency: 0 ms
STOCK_LEVEL - committed: 0, aborted: 0, avg latency: 0 ms
TOTAL - committed: 6254, aborted: 0, avg latency: 19 ms
```

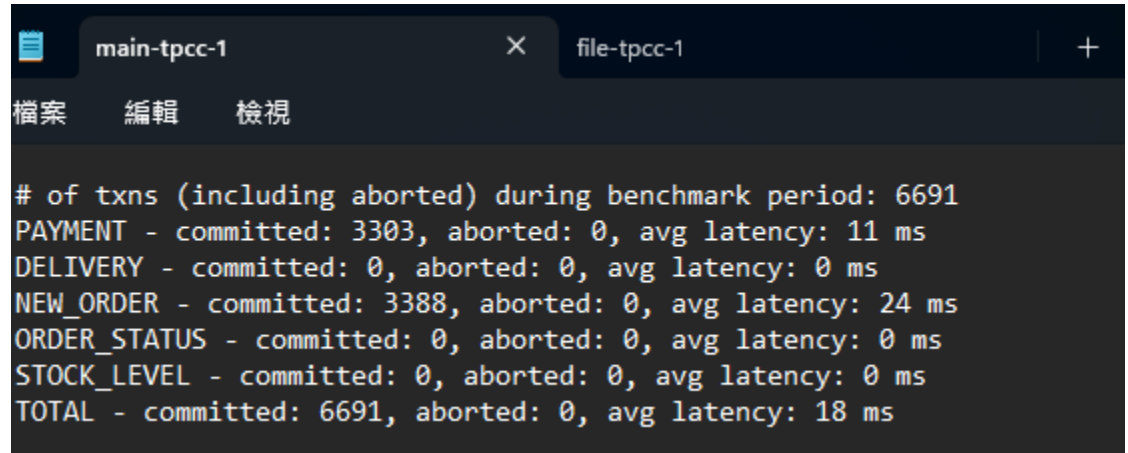
- 在FileMgr我們增加lock的數量。利用reentrant lock和lock stripping讓不同檔案可以使用不同的lock。因為之前的FileMgr是共用一個lock, 只要有人在使用, 就沒辦法讓其他人使用。但假如只是要用不同的檔案, 會造成不需要的等

待。於是我們採取了這種假如沒有被lock, 就可以使用其他檔案。此外我們也一樣縮小了一些function的critical section。

比較版本為tpcc-c

- BUFFER\_POOL\_SIZE=102400
- NUM\_WAREHOUSES=1

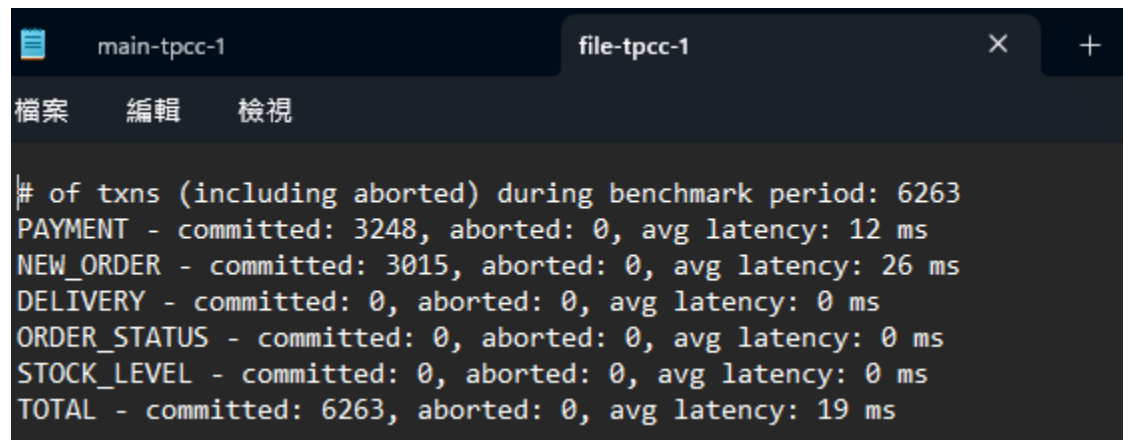
沒有任何optimization: total throughput = 6691



```
main-tpcc-1 file-tpcc-1
檔案 編輯 檢視

# of txns (including aborted) during benchmark period: 6691
PAYMENT - committed: 3303, aborted: 0, avg latency: 11 ms
DELIVERY - committed: 0, aborted: 0, avg latency: 0 ms
NEW_ORDER - committed: 3388, aborted: 0, avg latency: 24 ms
ORDER_STATUS - committed: 0, aborted: 0, avg latency: 0 ms
STOCK_LEVEL - committed: 0, aborted: 0, avg latency: 0 ms
TOTAL - committed: 6691, aborted: 0, avg latency: 18 ms
```

File相關的optimization: total throughput = 6263



```
main-tpcc-1 file-tpcc-1
檔案 編輯 檢視

# of txns (including aborted) during benchmark period: 6263
PAYMENT - committed: 3248, aborted: 0, avg latency: 12 ms
NEW_ORDER - committed: 3015, aborted: 0, avg latency: 26 ms
DELIVERY - committed: 0, aborted: 0, avg latency: 0 ms
ORDER_STATUS - committed: 0, aborted: 0, avg latency: 0 ms
STOCK_LEVEL - committed: 0, aborted: 0, avg latency: 0 ms
TOTAL - committed: 6263, aborted: 0, avg latency: 19 ms
```

## 2. 實驗:

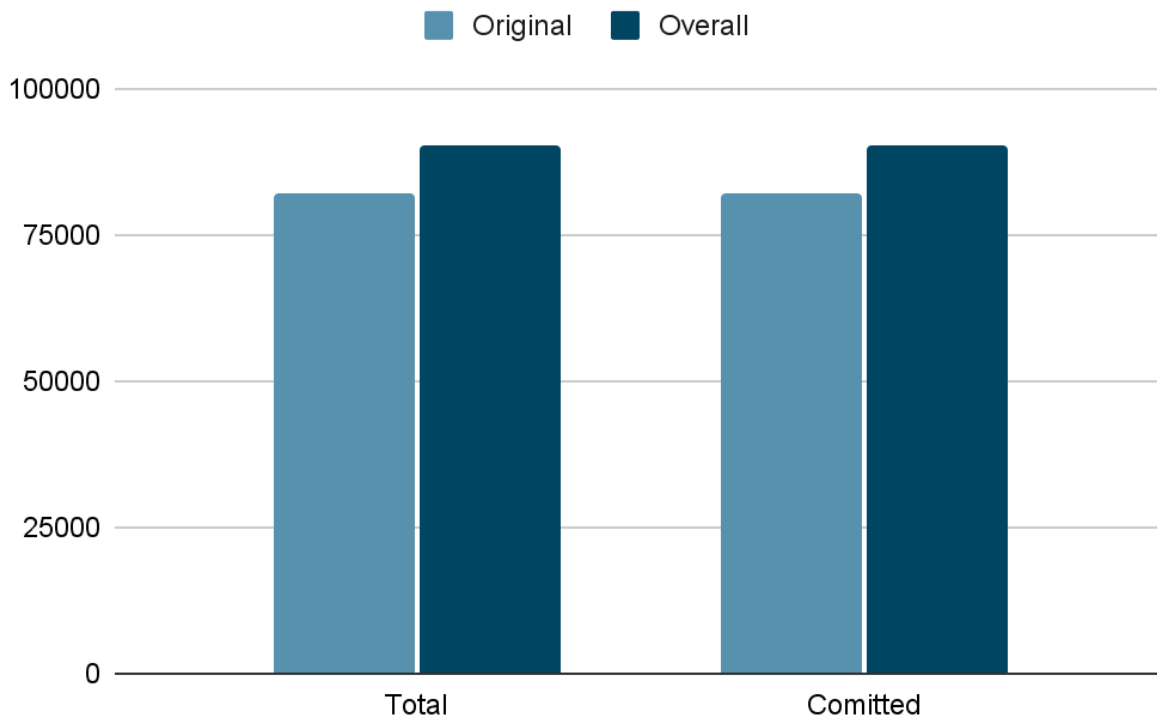
- Micro

不管怎麼調參數latency都沒有太大的變化, 但我們發現我們的版本會比原來的還要多做transactions, aborted的數量也比較少

改變的參數:

- NUM\_ITEMS=10000
- RW\_TX\_RATE=0.5
- WRITE\_RATIO\_IN\_RW\_TX=0.8

## # of txns during benchmark period



Original: total=82092, total throughput=committed=82035, aborted=57

Overall: total=90217, total throughput=committed=90166, aborted=51

- tpc-c

對於這個benchmark我們一樣也測試了原本版本與最終我們優化的code進行比較

比較版本為tpc-c

- BUFFER\_POOL\_SIZE=102400
- NUM\_WOREHOUSES=1

沒有任何optimization: total throughput = 6691

```
main-tpcc-1  overall-tpcc-1
檔案 編輯 檢視
# of txns (including aborted) during benchmark period: 6691
PAYMENT - committed: 3303, aborted: 0, avg latency: 11 ms
DELIVERY - committed: 0, aborted: 0, avg latency: 0 ms
NEW_ORDER - committed: 3388, aborted: 0, avg latency: 24 ms
ORDER_STATUS - committed: 0, aborted: 0, avg latency: 0 ms
STOCK_LEVEL - committed: 0, aborted: 0, avg latency: 0 ms
TOTAL - committed: 6691, aborted: 0, avg latency: 18 ms
```

Overall Optimization: Total throughput = 5986

```
main-tpcc-1  overall-tpcc-1  X  +
檔案  編輯  檢視

# of txns (including aborted) during benchmark period: 5986
PAYMENT - committed: 2976, aborted: 0, avg latency: 12 ms
DELIVERY - committed: 0, aborted: 0, avg latency: 0 ms
NEW_ORDER - committed: 3010, aborted: 0, avg latency: 27 ms
ORDER_STATUS - committed: 0, aborted: 0, avg latency: 0 ms
STOCK_LEVEL - committed: 0, aborted: 0, avg latency: 0 ms
TOTAL - committed: 5986, aborted: 0, avg latency: 20 ms
```

以下是我們做過的其他測試，可以看到相對於原本版本，並沒有增加太多的latency或是其他的結果

TPC-C Latency

