

Team14 - Database Assignment 5 Report

106070038 杜葳葳 106030019 吳岱容

■ Explanation

■ implement private workspace per Tx

◆ 每個 Tx 新增一個 private workspace 的物件 (WorkSpaceForTx 由兩個 HashMap 構成)，來存 commit 時要 flush 到 buffer 的資料。

■ add CLock

◆ 在 LockTable 中依據 2V2PL 的規則實作 CLock，在 tx.commit 的時候針對每個存在 private workspace (WorkSpaceForTx) 的紀錄來判斷要不要加上 Clock，加上 lock 後將裡面的內容真正藉由 BufferMgr 寫到底層。

■ Setval

◆ 在 recordPage 的 setVal 實作放入 private workSpace (因為在這邊之後會接到 Buffer)，然後不往下做 buffer 的 setVal (變成到 Transaction commit 後才做)。

■ GetVal

◆ 也是在 recordPage 下的 GetVal 檢查，檢查現在讀取的資料是不是在同一個 transaction 內有拿過 XLock，如果有資料直接從 private workspace 拿取，如果沒有則照原本的方法。

■ HeaderFilePage

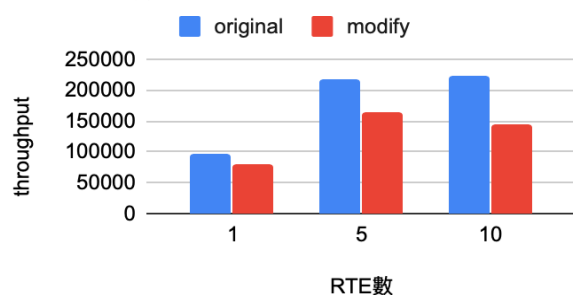
◆ 在這邊裡面也有 getVal 及 setVal，處理方式和上面大概相同。

■ **Environment:** Dual-Core Intel Core i5 CPU @ 2.7 GHz, 8 GB RAM, 128 GB SSD, MacOS

■ Experiment

■ NUM_RTES

RTE個數

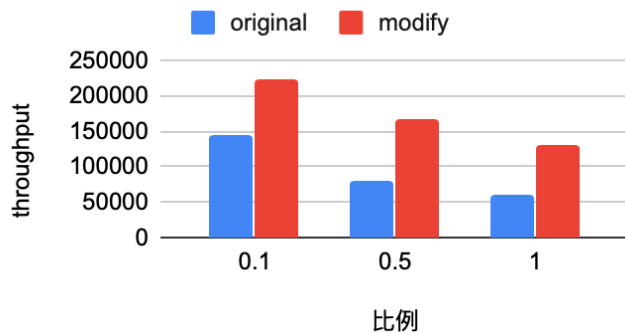


RTE 數越高，throughput 越高，但修改後的效能不如原本的，推測是因為實作 2V2PL 使得 write 要等到 commit 才能真正寫入。

	1	5	10
origin	97415	219010	222042
modify	78690	164904	144051

■ RW_TX_RATE

Read-Write Tx Rate

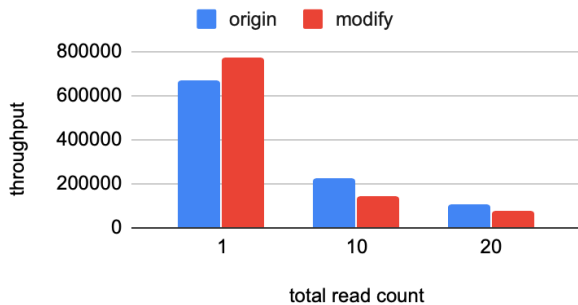


Tx Rate 小代表 Read Tx 多，因此 throughput 較高。另外，能從圖中發現實作 2V2PL 的效能明顯較原本的好。

	0.1	0.5	1
origin	144051	79892	59960
modify	222042	166548	129795

■ TOTAL_READ_COUNT

total read count

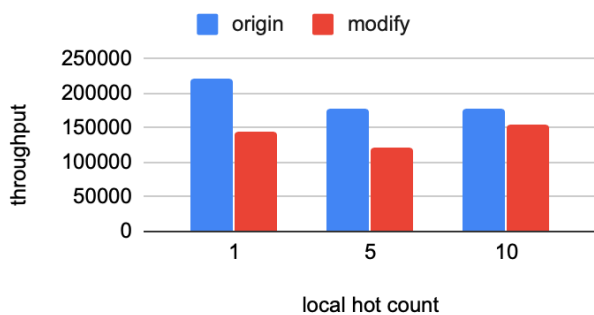


Read_count 越高，每個 Tx 讀的 record 數越多，因此效能有明顯下降的趨勢。

	1	10	20
origin	667858	222042	108350
modify	771145	144051	79354

■ LOCAL_HOT_COUNT

Local hot count

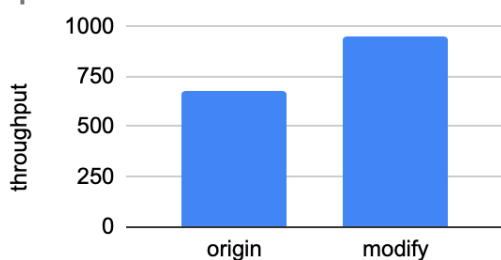


LOCAL_HOT_COUNT 數越高，越易發生碰撞，因此效能變差。

	1	5	10
origin	222042	176016	178497
modify	144051	121874	153495

■ TPCC Benchmark

tpcc-WareHouse1



實作 2V2PL 可以大幅減少 Write 阻擋 Read 的情形，因此 modify 後的效能明顯高於原本的。

	1
origin	674
modify	951

■ Summary

- 從上面的實驗可以發現，其中有幾個參數在改動後效能並沒有提升，反而下降，因此推測仍取決於 **benchmark** 的設計。另外，2V2PL 雖然主要目的是為了減少 **share lock** 以及 **exclusive lock** 的互斥來進行優化，但是 **certify lock** 的存在也會影響到其他 **transaction**，整體來看對效能的影響或許不是那麼好推論結果。
- 總結來說，2V2PL 可能較適合 **Read Tx** 多的 **benchmark**。