

# Assignment4\_report

Team3 105062548 胡裕人 105000039 林杰

## 想法

---

讓每一個 txn 可以一起競爭所有的 LOCK，在要 LOCK 的過程中如果遇到 collision，txn number 比較小的優先權比較大，但是如果 txn 大的已經開始執行 sql，那麼會讓該 txn 執行優先，並且再嘗試拿該 LOCK。

## 步驟

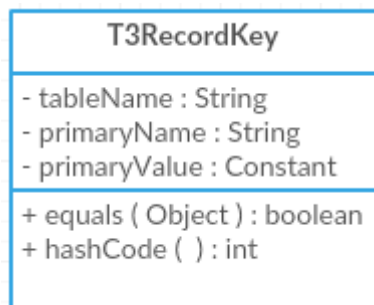
---

### 想法

利用 T3ConservativeConcurrencyMgr 取得 T3RecordKey 的 Lock

T3RecordKey 代表一個 record 的物件

1. 在 package org.vanilladb.core.storage.tx 新增 T3RecordKey.java



代表一個 record 的 LOCK

2. 因為在 LockTable 裡的 lock 都有 wait 的功能，不符合我們的設計理念，沒能有 wait，拿不到 LOCK 就釋出所有的 LOCK 或是等待可以 LOCK 的時機，自訂 lock 的方法，如果拿不到 lock 就回傳 0=拿不到 LOCK、1=成功拿到 LOCK、2=搶走 LOCK 的 TXN 正在執行 SQL

甲、新增 T3sLock(Object, long)

乙、新增 T3xLock(Object, long)

3. 複製 SerializableConcurrencyMgr.java 在  
 org.vanilladb.core.storage.tx.concurrency 新增  
 T3ConservativeConcurrencyMgr.java
  - 甲、新增 readT3RecordKey(T3RecordKey) · 對 T3RecordKey 做 slock
  - 乙、新增 modifyT3RecordKey(T3RecordKey) · 對 T3RecordKey 做 xlock
  - 丙、新增 releaseLocks() · 釋放 tx 所有的 Lock
4. 在 microbenchmarkProc 做些修改
  - 甲、microbenchmarkProc 有兩次試圖抓取 LOCK 的動作，第一次是嘗試抓取，  
 第二次是確定都有抓到 LOCK，如果都有抓到 LOCK 才能執行 SQL
5. 為了做出 txn 順序性，在 LockTable 做一些修改
  - 甲、在 LockTable 新增一個 CopyOnWriteArrayList T3LockedList，這個  
 List 代表已經拿到所有 lock 的 txn
  - 乙、新增 registerT3LockedList()，把 txn number 放進 List
  - 丙、新增 leaveT3LockedList()，把 txn number 踢出 List
  - 丁、如果在做 T3slock 時有 lock 衝突到，比 txNum 大小，小的優先

## 實驗

Pool Size	50000		100000		200000	
	Total	Avg(ms)	Total	Avg(ms)	Total	Avg(ms)
1	64885	9	64729	9	74415	7
2	67585	8	71851	8	74255	7
3	68388	8	70255	8	74614	7
AVG	66952.67	8.33	68945	8.33	74428	7

version	Origin		Team 3	
	Total	Avg(ms)	Total	Avg(ms)
1	6002	8	64729	9
2	6063	8	71851	8
3	7835	8	70255	8
AVG	6633.33	8	68945	8.33