Assignment4\_reprot

Team3 105062548 胡裕人 105000039 林杰

# 想法

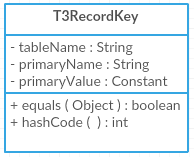
讓每一個txn可以一起競爭所有的LOCK，在要LOCK的過程中如果遇到collision，txn number比較小的優先權比較大，但是如果txn大的已經開始執行sql，那麼會讓該txn執行優先，

# 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，沒有wait，拿不到就釋出所有的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
   1. 新增T3sLock(Object,long)
   2. 新增T3xLock(Object,long)
3. 複製SerializableConcurrencyMgr.java在org.vanilladb.core.storage.tx.concurrency新增T3ConservativeConcurrencyMgr.java
   1. 新增readT3RecordKey(T3RecordKey)，對T3RecordKey做slock
   2. 新增modifyT3RecordKey(T3RecordKey)，對T3RecordKey做xlock
   3. 新增releaseLocks()，釋放tx所有的Lock
4. 在microbenchmarkProc 做些修改
   1. 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
   1. 在LockTable新增一個CopyOnWriteArrayList T3LockedList，這個List代表已經拿到所有lock的txn
   2. 新增registerT3LockedList()，把txn number放進List
   3. 新增leaveT3LockedList()，把txn number踢出List
   4. 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 |  |  |  |  |  |  |