

Final Project Report

Implement Explain

1. Bench:

- Procedure for Calvin
 - 將 stored procedure request 傳遞給 calvin，為此將原本 bench 中舊有的 StoredProcedure 改為 CalvinStoredProcedure 以便傳遞至 calvinDB 中做接收。而在內容方面，幾乎與原本 bench 中的 procedure 類似，中間差異為 implements 的不同，以及在 performTransactionLogic() 函式中須取得的資料來源為 class 自身所持有的資料。

```
public class TpccStoredProcFactory implements CalvinStoredProcedureFactory {  
    @Override  
    public CalvinStoredProcedure<?> getStoredProcedure(int pid, long txNum) {  
        CalvinStoredProcedure<?> sp;  
    }  
}
```

2. Calvin:

- Communication
 - 這部分的 client side 我們放在 bench 實作。這邊實作將 total-order 送給 vanilla comm，接收到 P2pmessage 的時候把它存起來。
- Scheduler
 - 在收到 run() 的訊息時，scheduler 將會利用 FACTORY_CLASS 產生一個 stored procedure 以生成一個新的 task 加入 mutiple thread 中實做。
- Metadata
 - 僅做了 PartitionMetaMgr() 的部份，用於取得 record 中的 partitionID。
- Recovery
 - 尚未實做。
- Concurrency
 - 這邊我們是用跟 core 一樣的 Concurrency。
- Stored procedures
 - 主體以 core 中的 stored procedure 下去做更改，不同的是在接收 bench 的 request 後去實做 performTransactionLogic()。
- Cache
 - 我們實作了兩個 class 第一個是 CacheRecord 裡面包含的東西，有 table name, field name, predicate。第二個是 Cache 裡面會用 RecExisting 去存 record，並且實現了助教的兩個要求，第一個是從 vanilla comm 存傳過來的 record，第二個是實現跟 vanilla core 的互動。

Experiments

這次的作業我們的架構還是存在許多問題來不及解決，只能等助教公布基礎版 calvin 我們再去 trace code 。希望最後那次能做出來！