Final Project Report. Group 15.

(106060009 游智鈞，106060019 林威宇)

Explanation:

- Communication:
  這裡的實作其實跟 core 很像，我們只是在 client 端以及 server 端裡面多註冊了一個 vanillacomm client server 和 vanillacomm server，參照 vanillacomm 中 clientdemo 及 serverdemo 的使用方式，傳輸 total order message 和 p2pmessage。

- Scheduler:
  我們是按照 final project readme 上面的流程，主要是將 storedprocedure dispatch 到一個 新的 thread。

- Metadata:
  將所收到的資料做 partition，使得各 server 拿到相應的資料，partition 的方式是透過 hashcode ，我們有定義一個新的資料結構 calvinrecord ，裡面存放 tablename 及 record id。

- Recovery:
  Log stored procedure 的 requests (每個 tx 都有一個 Recovery Manager)

- Concurrency:
  利用 slock、xlock 預先 lock 所有在此 tx 內所需的 read / write record，也就是 Conservative Locking，最後當 tx commit 後，release 所有的 lock (每個 tx 都有一個 Concurrency Manager)。

- Stored Procedures:
  Store procedure 的部分，參考了 core 裡面的寫法，利用 prepare 傳入參數，並在 prepare 階段註冊 concurrencyMgr 和 recoveryMgr，另外我們也參照 paper 的做法，實做了 read write set analysis。Execute 的部分，按照 paper 的順序，先執行 local read 再 server remote read 最後 再將結果傳回 client 端。比較特別的是，我們放棄了 core 裡面的 executeSql 函式，取而代之，定義 executeLocalRead 和 executeLogic 函式。將 原本 executeSql 函式拆成兩部分執行。

- Cache:
  用來快取 local 與 remote 端的 records，利用 Map 和 List 的方式儲存 records 的資料，如 txNum、fieldName 和對應值。

- Benchmark:
  這次修改 MicroBenchmark，自訂義新的 MicroProcedure 以及 ParamHelper，並且利用 CalvinProcedure 實做不同 Procedure 所要執行的 function，例如在 txProcedure 中，利用 CacheMgr 得到 local 與 remote 的 records 後，來建立 ResultSet 以回傳 server 的 response。

- vanillacomm 傳輸的資料分為
  1. Storedprocedureobject : 顧名思義是傳輸一個用來辨識呼叫哪一個 storedprocedure 的 object. 裡面也包含了 storedprocedure 所需要的參數。
  2. serverResponse: 主要包含 server 端 resultset。
  3. TxRecordSet : 包含 cache record。

Experiments:

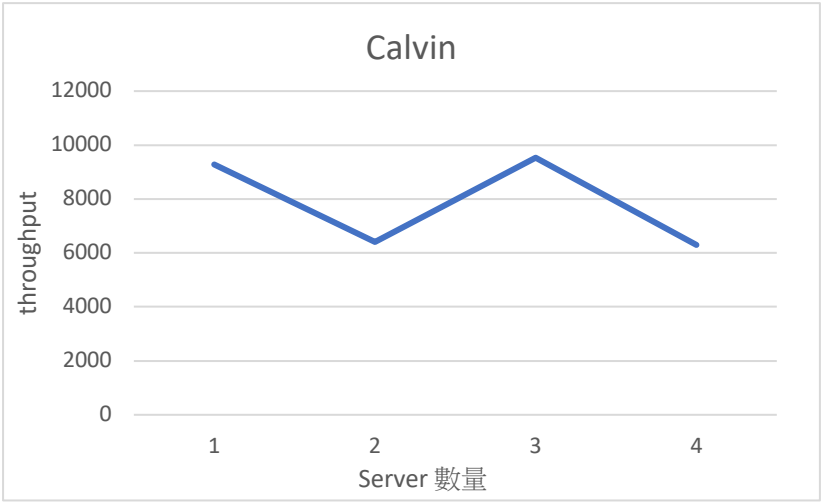
Environment:

1. Processor: intel core i5 2.3 GHZ
2. Ram: 16GB
3. Disk size (SSD): 512 GB
4. Operating system: Mac OS Catalina 10.5.4

Parameter:

1. Client Node: 1
2. Server Node: 1 ~ 4
3. Batch epoch: 5 ms

Result:



Discussion:

在這次實驗中，我們發現當只有 1 個 server 時，throughput 仍是很高，而我們認為這是因為 1 個 server 並不需 send 資料給其他 server，也不需要其它 server 資料寫入 remote cache record set 以節省時間，因此 throughput 較高，而 2 ~ 3 個 server 的 throughput 呈線性成長，不過再加入第 4 個 server 時，throughput 下降，我們觀察 CPU 的使用率已達 100%，使得每個 server 的效能下降，這也跟我們只使用一台電腦息息相關。