Final Optimization（Phase Two）

106062128 陳頌恩
106062328 李思佑

1. **Optimization**

   目標：**Reduce Latency of Transactions with 3 servers**

   實作內容：

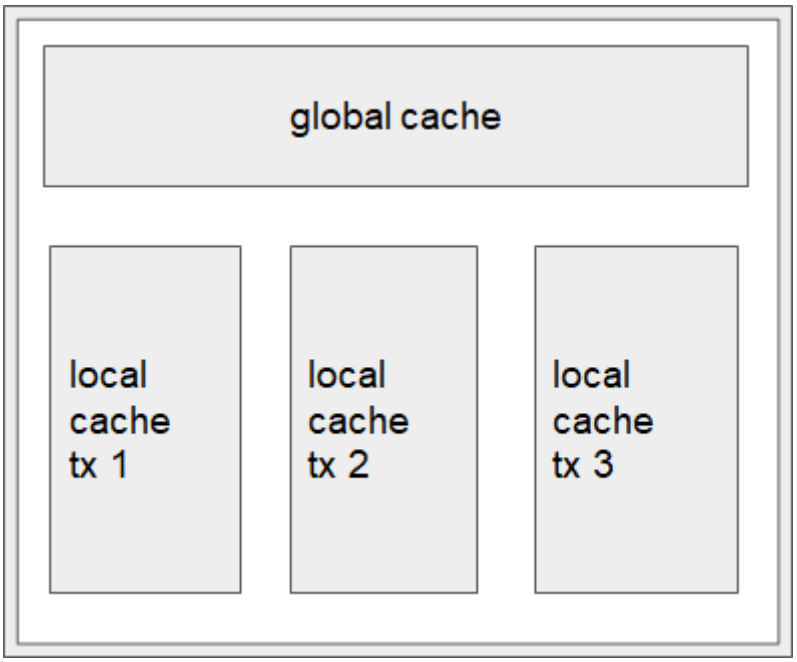
   (a). Find a master server for every transaction

   We choose a server as master, it will collect remote reads from other servers, and then send them to clients. We use this mechanism to avoid sending redundant data to clients. So we can reduce time spent on communication, thus lower latency.

   Note that in our experiment, we find this optimization sometimes lead to higher latency, because the master may finish the stored procedure slow. Therefore, we choose the master in the way it may execute faster. In particular, we choose the server with most local read/write, since it has less remote read and network latency.

   (b). Add global cache in each server

   When cacheMgr read records, we put records into global cache. When the same record is read, we get the record from global cache. This method can reduce I/O time, thus reduce latency.



2. **Experiment**

Environment :

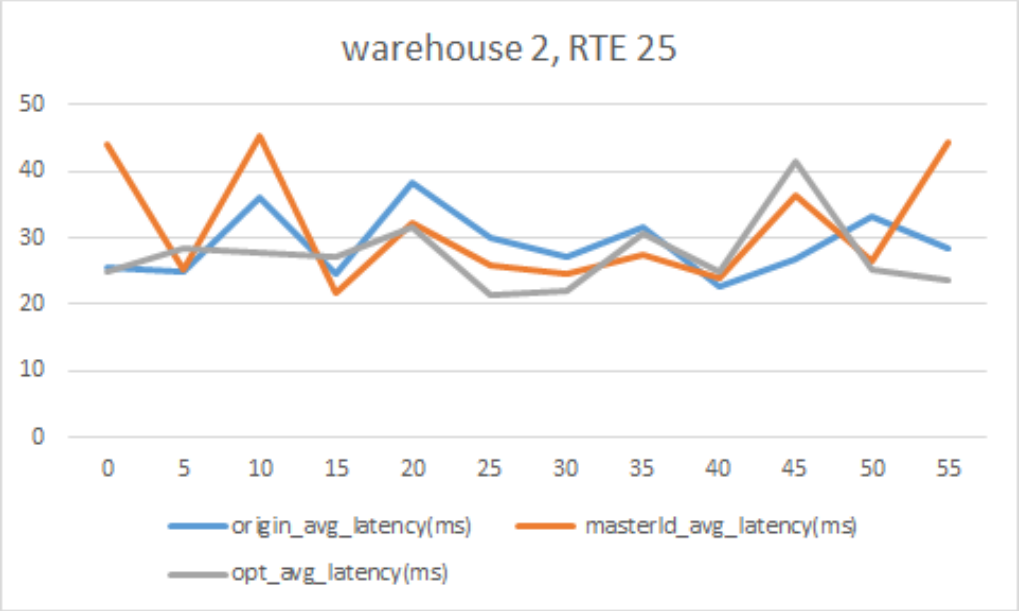
intel i7-8700@3.2GHz, 8GB RAM, 256GB SSD, Windows10

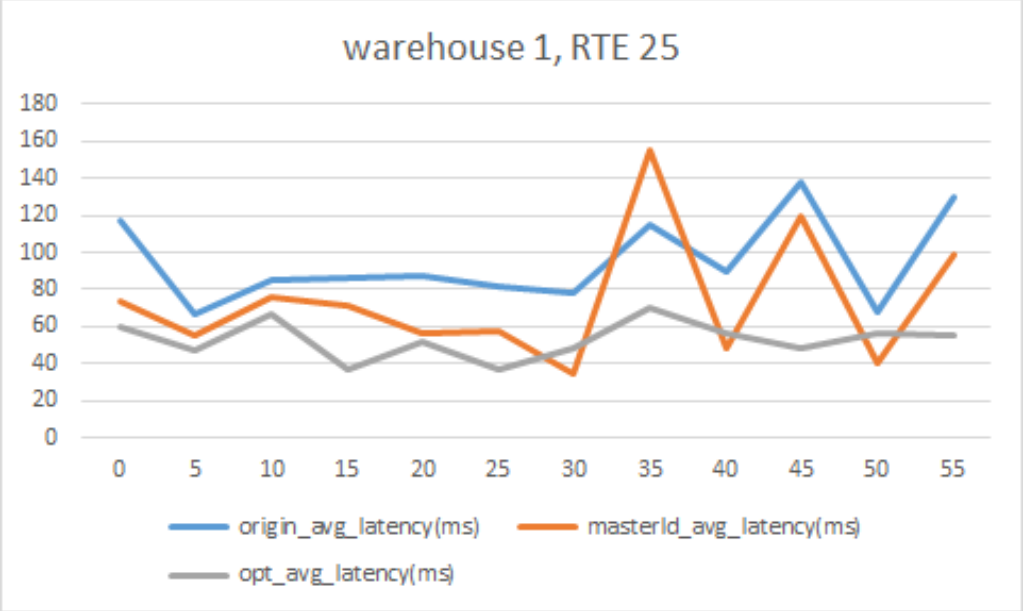
On three different computers in EECS 326

Setting :

partition: 3, server: 3, client: 1, TPCC dataset

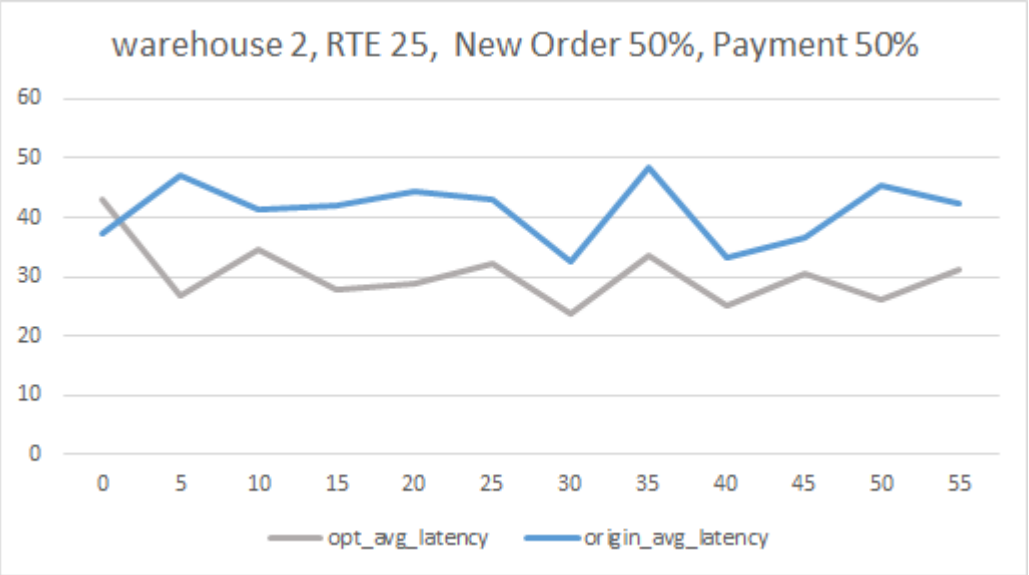
Controlled variable: # of warehouses

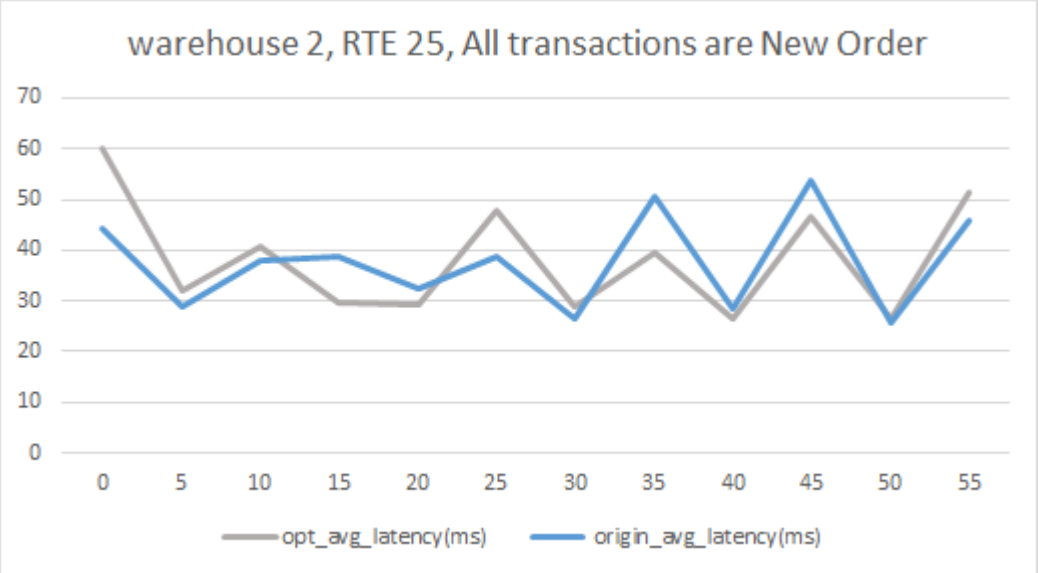


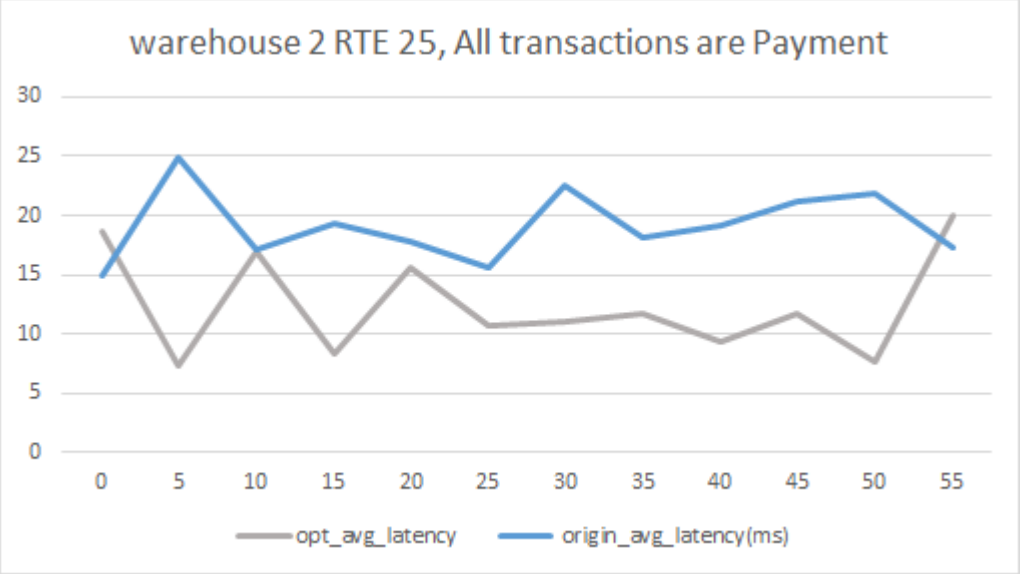


- When number of warehouses is lesser, the locality of dataset increases. Hit rate of global cache also increases, so we can save more I/O time. As a result, decrease of latency is more obvious.
- We can observe a high peak at 35s in orange line in second graph. Because master server may run slower than others, increasing the latency. But on average, adding master server leads to decrease of latency.

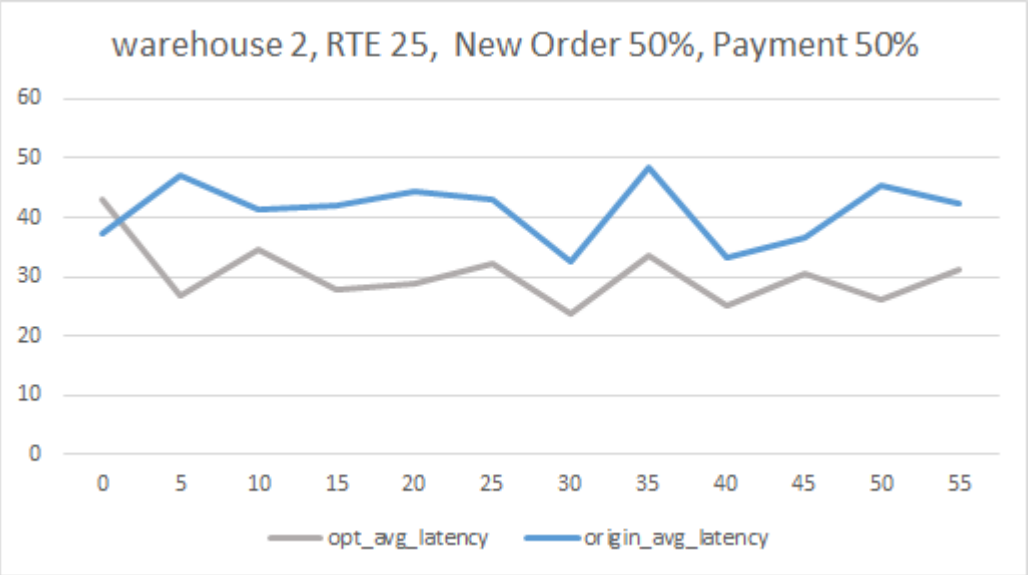
Controlled variable: Type of Transactions

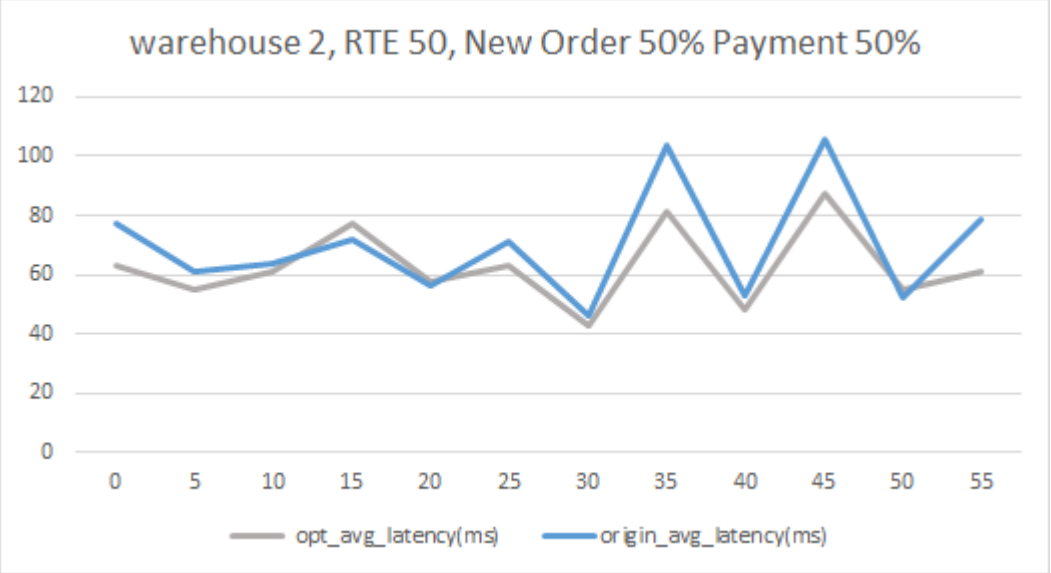






Controlled variable: # of RTEs





- When there are more RTEs, the improvement of latency is lesser.
- **Reason:** When there are more RTEs, there will be more threads to execute stored procedure, which makes the master server more unlikely to be the first one to finish.

  And there's a scene might happen: some active nodes(but not master) have finished the stored procedure before master but still need to wait for the end of master's execution.

  So the optimization of choosing master server may increase latency , which offsets the improvement made by global cache.