Assignment #7

Build on Assignment #6 (NOrec) to build HyNOrec Hybrid TM algorithm. Use the following algorithm,

```
padded unsigned seqlock
padded unsigned counter[]
thread local unsigned id

HW_POST_BEGIN
  if (seqlock & 1)
    while (true) // await abort

HW_PRE_COMMIT
  counter[id] = counter[id] + 1
```

HTM code is provided in as a separate standalone global lock implementation.

We have some bank accounts (represented by a number) initialized with $1,000, implement a program to do 100,000 transactions of 10 transfers between 20 randomly chosen accounts (in each transaction). The amount of money transferred is $50 and no negative accounts' balances are allowed. You need to run the program using 1, 4, 8, 16, 24, 32, 48, 56, 64 threads. The number of transfers should be split on the number of threads. For example, 25,000 transfers per thread when we are running 4 threads. The total amount of money before and after the transfers must be the same. You program should print the sum before and after executing all transfers. Implement the program using HyNOrec. You should build a special purpose HyTM for this problem.

Note: HTM code will not run of your laptop if the processor does not support TSX extension.

You can use the following server which has 72 cores and supports TSX extension.

```
ssh -p 2002 nile.ece.vt.edu -l <with your account>
For example:
ssh -p 2002 nile.ece.vt.edu -l student1
```

Account will be sent by email directly

Deliverables:

Run your program with 4 different configurations and 1, 4, 8, 16, 24, 32, 48, 56, 64 threads.

1. 1,000,000 accounts.
2. 1,000 accounts.
3. 1,000,000 accounts with disjoint access (which means each thread access a unique set of the accounts. E.g., with two threads, one will access the first 500,000 accounts and the other will access the following 500,000 accounts).
4. 1,000 accounts with disjoint access.

In each case, plot also statistics about how many transactions finished in HTM and STM.

Dead line:

May 2nd, 2018 at 12:00 PM