On Scheduling Best-Effort HTM Transactions

Mohamed Mohamedin, Roberto Palmieri and Binoy Ravindran

Systems Software Research Group

Virginia Tech

Why Scheduling

- When the contention is high → Live-lock
  - Lot of work is wasted
- Old problem
  - Studied in many fields
- What is different with current HTM?

Scheduling in HTM

- No control on conflict resolution

- A transaction is aborted first, and then the program is notified
  - Programmer cannot decide which transaction to abort

- Speculative
  - All written objects are discarded upon abort
    - Cannot get information from aborted transactions
  - No non-transactional instructions in Intel Haswell HTM

How to detect conflict before HTM?

- Repeat what HTM already does very efficiently

- Any added meta-data will consume HTM precious resources

  - Meta-data itself increase conflicts

    - E.g., Read-Write Locks

Proposed Solution

- Requires a priori knowledge of the working set
- Conflicting transactions are not scheduled together
  - Why we still need a TM?
    - Conflict estimation may not be exact
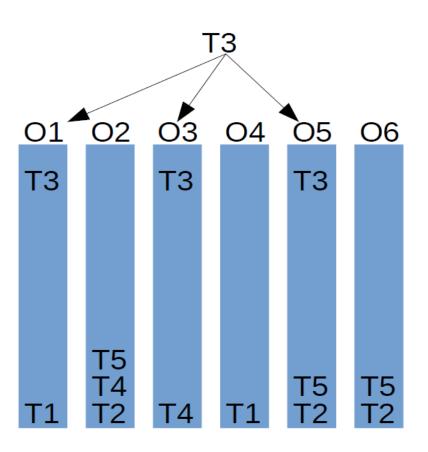   - HTM will solve conflicts due to wrong estimation

Proposed Solution: Octonauts

- Scheduling queues
  - Each queue protects a group of objects
- A transaction subscribe to all required queues
- Lightweight signature based read/write sets
  - Collecting information from committed transactions
  - Provided by the programmer
- Queue are efficiently implemented as counters
  - Tickets

Octonauts

- T1 and T2 are executing now

- T3 is subscribing to O1, O3, and O5 atomically

- T4 waiting on O2 queue (although It is on top of O3 queue)
- T5 is waiting for O2, O5, and O6 queues
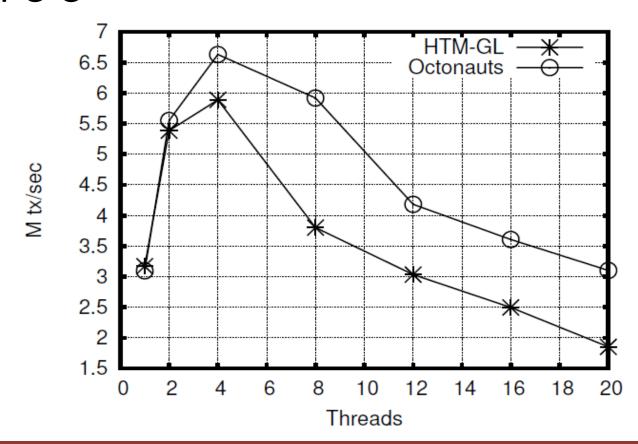
Octonauts: HTM-STM communication

- HTM is a best-effort
  - An HTM scheduler should also provide an efficient fallback path
- Using Phasing
  - Plain HTM
    - All workload is in HTM
  - Instrumented HTM
    - An STM transaction exists
- Communication is via write-set signatures in a Ring

Evaluation

- TPC-C

Other Benefits of Octonauts

- Select which transactions cannot run together
  - Based on conflict (default)
  - Other possible criteria:
    - Capacity
    - Time
    - User specific reason

Thanks!

Questions?

Research project's web-site: www.hyflow.org