TxSampler: A Lightweight Profiler for Hardware Transactional Memory

CLOMP-TM Details

Anonymous Author(s)

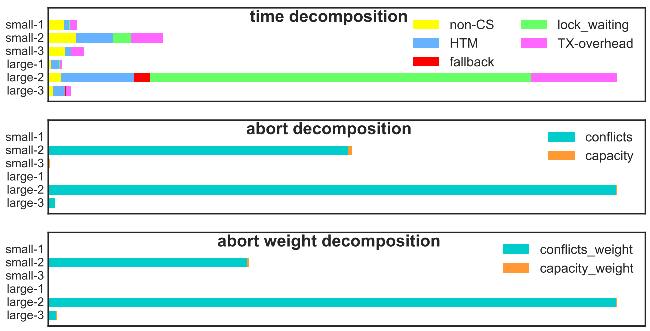

Figure 1. CLOMP-TM data from TxSampler.

Input #	ScatterMode	Expected Characteristics
1	Adjacent	Rare conflicts, cache prefetch friendly
2	FirstParts	High conflicts, cache prefetch friendly
3	Random	Rare conflicts, cache prefetch unfriendly

Table 1. Inputs for CLOMP-TM.

1 Correctness Verification: CLOMP-TM

CLOMP-TM [1] is a CORAL benchmark that is designed to characterize the behavior of HPC applications running with HTM. Originally, CLOMP-TM is implemented atop the HTM in Blue Gene/Q; we port the code to run with Intel TSX. CLOMP-TM provides different configurations and inputs for evaluating different HTM behaviors. Table 1 shows the three inputs with different characteristics, each of which runs with two configurations: small transactions and large transactions. With this benchmark, one can (1) compare the behaviors of the same configuration but different inputs, or (2) compare the performance of different configurations with the same input. Figure 1 shows the performance data collected by TxSampler for CLOMP-TM running with 14 threads, which can correctly explain the performance behavior of CLOMP-TM.

Different behaviors across inputs. The top of Figure 1 is the CPU cycles decomposition. For small transactions, regardless of the inputs, TxSampler reports high HTM overhead T_{oh}, which matches our intuition that small transactions have relatively higher overhead. For large transactions, we can see that with low conflicts and small footprint (input 1), most of the execution time is in transactions T_{tx} and there are nearly no aborts. For high conflicts (input 2), most of the time is spent in lock waiting T_{wait}; also it has a large number of conflict aborts and the weight associated with the abort as computed by TxSampler is high. For input 3 with large memory footprint, TxSampler reports a larger portion of capacity aborts compared to high conflicts.

Different performance between configurations. We obtain two insights from studying CLOMP-TM: (1) With low conflicts, large transactions perform better, and (2) with high conflicts, small transactions perform better. TxSampler provides the data to explain these observations. First, with low conflicts, both small and large transactions rarely fall into the slow paths, so they both have small T_{wait} and T_{fb}. Thus, the overhead T_{oh} has a high impact on the performance. As small transactions have larger T_{oh}, they perform worse than large ones. Second, with high conflicts, small transactions have a better performance than large ones. By causing a large number of aborts, large transactions have more execution time in T_{wait} and T_{fb} since large transactions incur higher abort penalty and serializing large transactions is more costly than small ones.

References

[1] Martin Schindewolf, Barna Bihari, John Gyllenhaal, Martin Schulz, Amy Wang, and Wolfgang Karl. 2012. What scientific applications can benefit from hardware transactional memory?. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE Computer Society Press, 90.

PL'17, January 01–03, 2017, New York, NY, USA
2017. ACM ISBN ...$15.00
https://doi.org/