Programming Project: Design and Implementation of a Concurrent Data Structure Using Transactional Memory

1 Project Specification

Transactions can be used for designing concurrent data structures starting from their sequential design. This project consists of the implementation of a concurrent red-black tree data structure using two techniques:

- Consistency Oblivious Programming;
- Transactional Memory.

Consistency Oblivious Programming (COP) (link below) has been introduced by Yehuda Afek, Hillel Avni, and Nir Shavit in the 15th International Conference of Principles of Distributed Systems (2011) and aims at deriving a broad class of algorithms for concurrent data structures that execute without verifying consistency (the technical details of COP are included in the project specification).

The student is expected to provide the following implementations:

- an implementation of a sequential red-black tree where elements are integer values.
- the above implementation is expected to be made concurrent using transactional memory by wrapping each data structure operation into an atomic block. It is left to the student deciding which transactional memory library to use. Both software and hardware implementations of transactional memory are acceptable.
- an optimized implementation of red-black tree using the COP approach, which uses transactions in its internal design.

The three implementations should expose the same APIs that a standard red-black tree provides. Any Transactional Memory implementation detail should be decoupled from the benchmarking application.

A performance comparison among the three implementations is required to assess their effectiveness. The project is supplied with a test application, which consists of an on-line reservation system that uses a red-black tree to look-up and manipulate reservation items (e.g., cars, hotels, flights). A comprehensive report showing the performance of the three implementations under different contention levels and number of parallel worker threads.

Evaluation is expected to be performed on the multicore server machines made available by the instructor for the course work.

2 Supplementary Material