TLDS: Transactional Lockless Data Structures

Damian Dechev

Computer Software Engineering
Scalable & Secure Systems Lab

ucf-cs.github.io/tlds/

Abstract

Traditionally, non-blocking data structures provide linearizable operations, but these operations are not composable. Transactional data structures can perform a sequence of operations that appears to execute atomically, which facilitates modular design and software reuse. TLDS encompasses:

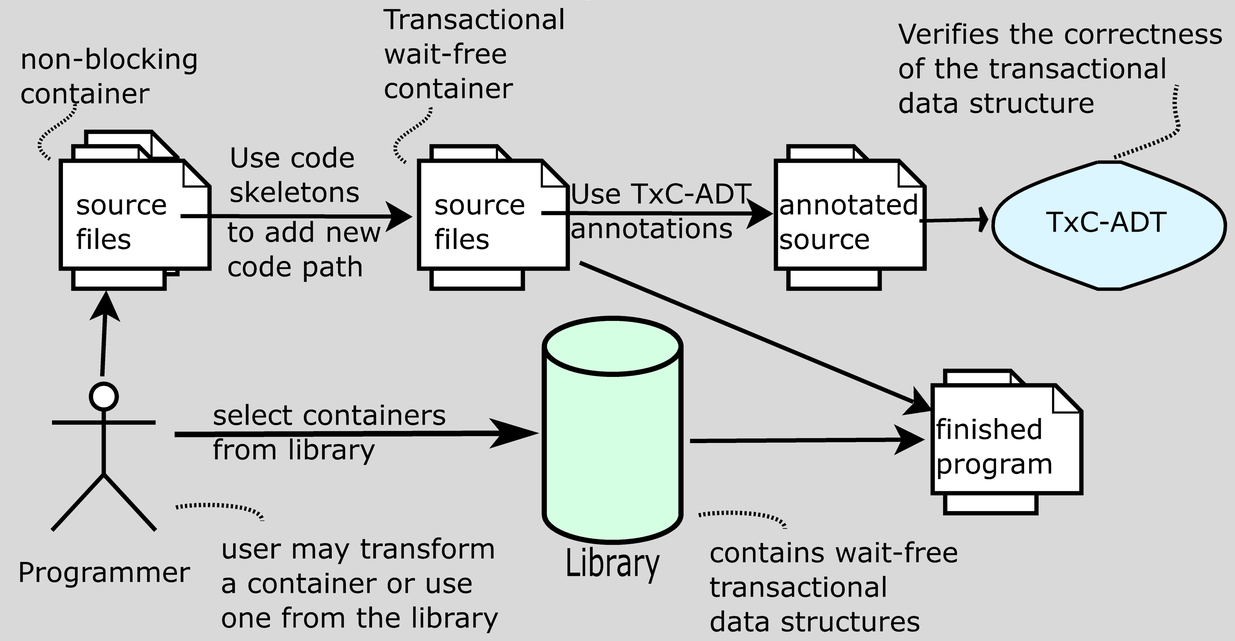
- A scalable methodology for transforming non-blocking data structures into transactional containers
- A library of transactional data structures
- A tool to validate their correctness

Correctness Tool

- We present TxC-ADT, the first tool that can check the correctness of transactional data structures.
- TxC-ADT recasts the standard definitions of transactional correctness in terms of an abstract data type.
- TxC-ADT verifies correctness conditions including serializability, strict serializability, opacity, and causal consistency.

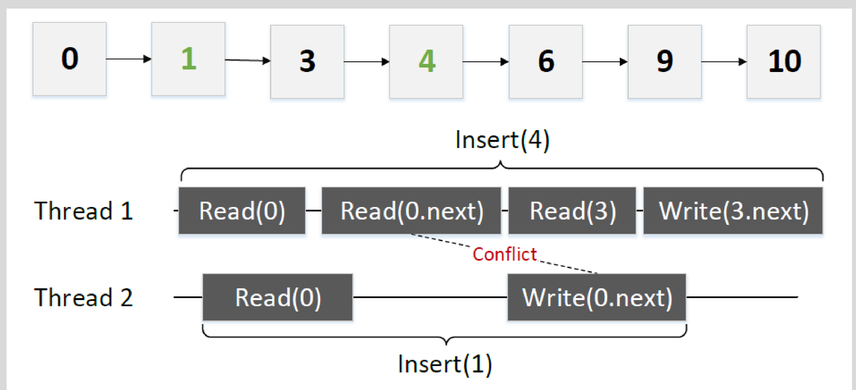
Overview

We present our work on transforming non-blocking containers into transactional ones, and verifying those transactional containers.

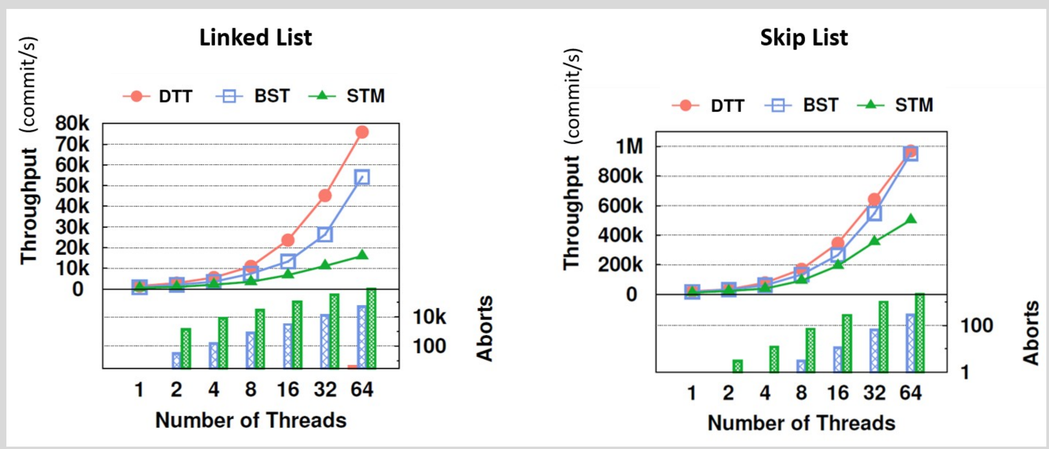


Dynamic Transactional Transformation

- We have developed a framework that transforms non-blocking data structures into wait-free transactional containers.
- This framework is open source: http://ucf-cs.github.io/tlds/
- We use semantic conflict-detection to allow commutative operations to execute without aborting, which improves upon STM.



- Results:



Library

- We provide a library of verified transactional data structures.
- Programmers can integrate our data structures to easily create fully-fledged parallel programs.

Conventional Transactional Code

```
void ThreadWork() {
    mutex.lock();
    if( skiplist.find(3) )
        skiplist.insert(5, 100);
    mutex.unlock();
}
```

Transactional Code Using TLDS

```
void ThreadWork() {
    ExecuteTransaction(TxFunction);
}

void TxFunction() {
    if( CallOp(skiplist, Find, 3) )
        CallOp(skiplist, Insert, 5, 100);
}
```

Impact

By providing the source code of our work, in addition to publishing our results, we enable developers to take advantage of our existing verified data structures. We also provide a framework that can be used to create additional containers.

Our approach will advance the state of the art in our field, which will impact the design and implementation of parallel programs. The impact on software in scientific and commercial applications will spur further research work, and increase market productivity.