Transactional Tasks: Parallelism in Software Transactions (Artifact)

Janwillem Swalens[1]

1 **Software Languages Lab, Vrije Universiteit Brussel**
jswalens@vub.ac.be

Abstract

Many programming languages support different concurrency models. In practice these models are often combined, however the semantics of the combinations are not always well-defined. We studied the combination of futures and Software Transactional Memory. We introduce *transactional tasks*, a mechanism to create futures in a transaction. Transactional tasks allow the parallelism in a transaction to be exploited, while providing safe access to the state of their encapsulating transaction. We created Clojure$^{\text{TxTk}}$, a fork of Clojure with support for transactional tasks. Furthermore, we ported two applications from the STAMP benchmark suite, and extended these to use transactional tasks: Labyrinth$^{\text{TxTk}}$ and Bayes$^{\text{TxTk}}$. Lastly, $^{\text{TxTk}}$Redex is a machine-executable implementation of the operational semantics, in PLT Redex.

1998 ACM Subject Classification D.1.3 [Concurrent Programming] Parallel Programming; D.3.2 [Language Classifications] Concurrent, distributed, and parallel languages

Keywords and phrases Concurrency, Parallelism, Futures, Threads, Fork/Join, Software Transactional Memory

Digital Object Identifier 10.4230/DARTS.2.1.13

Related Article Janwillem Swalens, Joeri De Koster, and Wolfgang De Muter, "Transactional Tasks: Parallelism in Software Transactions", in Proceedings of the 30th European Conference on Object-Oriented Programming (ECOOP 2016), LIPIcs, Vol. 56, pp. 23:1–23:28, 2016.
http://dx.doi.org/10.4230/LIPIcs.ECOOP.2016.23

Related Conference 30th European Conference on Object-Oriented Programming (ECOOP 2016), July 18–22, 2016, Rome, Italy

1 Scope

This artifact aims to provide the necessary material to reproduce the experiments in the companion paper. Firstly, Clojure$^{\text{TxTk}}$ is a fork of Clojure supporting transactional tasks. Labyrinth$^{\text{TxTk}}$ and Bayes$^{\text{TxTk}}$ are two applications built using Clojure$^{\text{TxTk}}$. These applications were used to evaluate the performance of transactional tasks in the paper. We encourage the reader to experiment with Clojure$^{\text{TxTk}}$, and develop other applications using transactional tasks.

In addition, we provide $^{\text{TxTk}}$Redex, a machine-executable implementation of the operational semantics described in the paper. $^{\text{TxTk}}$Redex can be used to interactively explore the operational semantics: you can implement small example programs and see which results it may produce. With $^{\text{TxTk}}$Redex you can 'play around' with the semantics, to gain a better insight into the concepts described in the paper.


© Janwillem Swalens;
licensed under Creative Commons Attribution 3.0 Germany (CC BY 3.0 DE)

2 Content

This artifact contains:

Clojure^{TxTk} An implementation of transactional tasks, as a fork of Clojure.

Labyrinth^{TxTk} and Bayes^{TxTk} Two example applications that use transactional tasks. These applications are implemented using Clojure^{TxTk} and were used to evaluate our approach. They should allow the experiments in the paper to be reproduced.

^{TxTk}Redex A machine-executable implementation of the operational semantics described in the paper.

The file `index.html` contains detailed instructions on the contents of this artifact. In particular, it contains the necessary instructions to build and run all parts of the artifact.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). The latest version of this artifact is available at `http://soft.vub.ac.be/~jswalens/ecoop-2016-artifact`. The latest version of Clojure^{TxTk} is also available at `https://github.com/jswalens/transactional-futures`.

4 Tested platforms

Clojure^{TxTk} is expected to work on any platform running Java 8. The results in the paper were gathered on a machine with two Intel Xeon E5520 processors (for a total of 8 physical cores or 16 hardware threads) and 8 GB of memory.

For ^{TxTk}Redex, we recommend Racket v6 and 4 GB of memory.

5 License

Clojure^{TxTk}: Eclipse Public License 1.0 (`https://opensource.org/licenses/EPL-1.0`)
Labyrinth^{TxTk} and Bayes^{TxTk}: BSD 3-Clause License (`https://opensource.org/licenses/BSD-3-Clause`)
^{TxTk}Redex: MIT License (`https://opensource.org/licenses/MIT`)

6 MD5 sum of the artifact

e9d1951304299497e1a1942d9926fd0b

7 Size of the artifact

48 MB