



Christoph Amrein



Timothy Markiewicz

Graduate Candidates	Christoph Amrein, Timothy Markiewicz
Examiner	Prof. Dr. Luc Bläser
Co-Examiner	Dr. Felix Friedrich, ETH Zürich, Zürich, ZH
Subject Area	Software

Software Transactional Memory for .NET

A runtime system for simple and efficient transaction-based concurrent programming in .NET

```
public class BankAccount {
    public int Balance { get; set; }

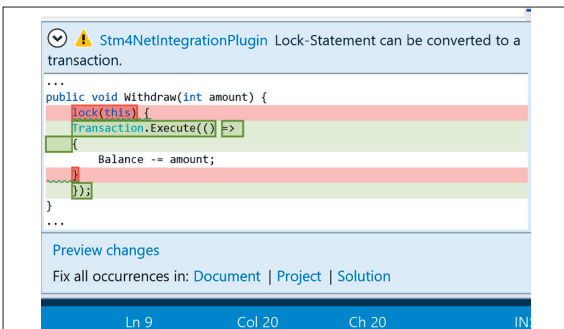
    public void Withdraw(int amount) {
        Transaction.Execute(() => {
            if (Balance < amount) {
                Transaction.Retry();
            }
            Balance -= amount;
        });
    }
}
```

Transaction block in C#.

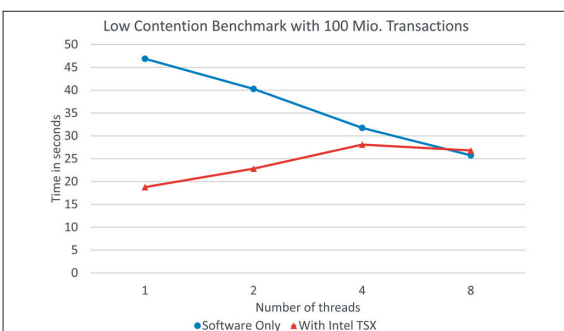
Introduction: Classical multi-threaded programming requires explicit synchronization of shared resources, being a highly challenging task for many software developers: On the one hand, identifying shared resources is non-trivial, easily leading to race conditions in case of under-synchronization. On the other hand, synchronization naturally bears the risk of deadlocks and starvation.

Approach/Technologies: Transactional memory is an alternative concept, significantly simplifying concurrent programming. It employs a descriptive programming model using the notion of transactions, inspired by database systems. A transaction constitutes an atomic sequential execution that is automatically isolated to other concurrent transactions. The runtime system guarantees the correct transactional execution, typically by using an optimistic concurrency control scheme.

Result: We have developed a practical transactional memory programming model and runtime system for the .NET framework. Our experimental evaluation shows that the solution is superior to existing .NET transaction frameworks in terms of performance, correctness, and ease of use. Moreover, it employs Intel TSX hardware transactional memory to increase performance. Last but not least, a refactoring tool for Visual Studio C# assists programmers on migrating existing code to the transactional model.



Refactoring assistance in Visual Studio.



Execution time of 100 million transactions, with and without Intel TSX hardware support.