

Actions in the Twilight: Eine Erweiterung zu Software Transactional Memory

Annette Bieniusa (Universität Freiburg)
gemeinsam mit Peter Thiemann (Universität Freiburg)
und Arie Middelkoop (Universiteit Utrecht)

Transactional Memory (TM) hat sich zu einer vielversprechende Alternative bei multi-threaded Anwendungen entwickelt. Im Vergleich zu traditionellen locking-basierten Synchronisationsverfahren verbessert es die Gesamtproduktivität des Programmierers, indem es feingranular Daten vor nebenläufigem Zugriff sichert und zugleich Modularität, Skalierbarkeit und Wiederverwendung von Code in anderem Kontext bietet. Erste industriereife Implementierungen sind mittlerweile verfügbar, die sich kompetitiv zu anderen Synchronisationsparadigmen verhalten.

Programmierer, die Transactional Memory in ihren Anwendungen einsetzen wollen, stehen jedoch weiterhin vor einem Problem: Die Verwendung von bestehenden APIs ist häufig nicht möglich, da diese in der Regel auf Protokollen mit Handshake oder Locking basieren, um I/O und andere Systemservices bzw. nebenläufige Datenstrukturen zu nutzen. Dies wiederum steht im Konflikt zur Nutzung von TM Operationen.

In letzter Zeit wurden Vorschläge zur sicheren Integration dieser Operationen in das TM Paradigma unterbreitet. Diese stufen Transaktionen, die unumkehrbare Seiteneffekte beinhalten, als *irreversible* und *inevitable* ein. Ähnlich wie schon in früheren Arbeiten wird dabei ein globales Lock eingeführt, das beim Ausführen solcher Transaktionen eine Serialisierung der laufenden Transaktionen erzwingt und so Konflikte und Rollback vermeidet. Diese Serialisierung verhindert allerdings, dass Handshake-Protokolle innerhalb von Transaktionen ausgeführt werden können. Ein weiterer Nachteil besteht in der Verringerung der dem Programm inhärenten Nebenläufigkeit, was zu einer niedrigeren Auslastung des Systems führt.

Dieser Vortrag stellt *Twilight STM* vor. Diese STM Implementierung umfasst neben den typischen STM Operationen eine zweigeteilte Commitphase. Zu Beginn des Commits wird zunächst die Gültigkeit der Speicheroperationen auf Konsistenz geprüft, bevor die geschriebenen Werte für andere Transaktionen zum Lesen und Schreiben freigegeben werden. Im Vergleich zu anderen Zweiphasen-Commitprotokollen kann der Programmierer in der sogenannten *Twilight Zone* zwischen Prüfung und tatsächlichem Commit flexibel und in Abhängigkeit der jeweiligen Anwendung auf Konflikte reagieren. Durch Korrektur der zu schreibenden Werte kann dann beispielsweise auf ein Rollback verzichtet werden. Eine erweiterte API unterstützt die Inspektionen von Konflikten und deren Korrektur.

Darüberhinaus können in der Twilight Zone I/O Operationen, Systemservices und Handshake-Protokolle ausgeführt werden. Diese externe Operationen sind direkt global sichtbar und erlauben es Transaktionen miteinander zu kommunizieren. Obgleich wir Garantien (Freiheit von Deadlocks, weak atomicity, progress) für die Twilight API geben können, obliegt es dem Programmierer, diese für seinen Twilight Code sicherzustellen.

Erste Benchmark-Ergebnisse zeigen, dass durch Twilight STM anwendungsspezifische Strategien zum contention management einfach und erfolgreich umgesetzt werden können.