Optimization Of Software Transactional Memory Operations

Abstract

A software transactional memory system is described which performs optimizations on software transactional memory instructions to achieve efficient performance. Software transactional memory blocks are replaced by software transactional memory instructions, which are then further decomposed into decomposed software transactional memory instructions. The decomposed instructions allow a compiler with knowledge of the instruction semantics to perform optimizations which would be unavailable on traditional software transactional memory systems. High-level software transactional memory optimizations are performed such as code movement around procedure calls, addition of operations to provide strong atomicity, removal of unnecessary read-to-update upgrades, and removal of operations for newly-allocated objects.