

Lock-based concurrency control suffers from programmability, scalability, and composability challenges. These challenges are exacerbated in emerging multicore architectures, on which improved software performance must be achieved by exposing greater concurrency. Transactional memory (TM) is an alternative synchronization model for shared memory objects that promises to alleviate these difficulties.

In this dissertation proposal, we consider software transactional memory (STM) for concurrency control in multicore real-time software, and present a suite of real-time STM contention managers for resolving transactional conflicts. The contention managers are called RCM, ECM, LCM, and PNF. RCM and ECM resolve conflicts using fixed and dynamic priorities of real-time tasks, respectively, and are naturally intended to be used with the fixed priority (e.g., G-RMA) and dynamic priority (e.g., G-EDF) multicore real-time schedulers, respectively. LCM resolves conflicts based on task priorities as well as atomic section lengths, and can be used with G-EDF or G-RMA. Transactions under ECM, RCM and LCM can retry due to non-shared objects with higher priority tasks. PNF avoids this problem.

We establish upper bounds on transactional retry costs and task response times under all the contention managers through schedulability analysis. Since ECM and RCM conserve the semantics of the underlying real-time scheduler, their maximum transactional retry cost is double the maximum atomic section length. This is improved in the the design of LCM, which achieves shorter retry costs. However, ECM, RCM, and LCM are affected by transitive retries when transactions access multiple objects. Transitive retry causes a transaction to abort and retry due to another non-conflicting transaction. PNF avoids transitive retry, and also optimizes processor usage by lowering the priority of retrying transactions, enabling other non-conflicting transactions to proceed.

We also formally compare the proposed contention managers with lock-free synchronization. Our comparison reveals that, for most cases, ECM, RCM, G-EDF(G-RMA)/LCM achieve higher schedulability than lock-free synchronization only when the atomic section length does not exceed half of the lock-free retry loop length. Under PNF, atomic section length can equal length of retry loop. With low contention, atomic section length under ECM can equal retry loop length while still achieving better schedulability. While in RCM, atomic section length can exceed retry loop length. By adjustment of LCM design parameters, atomic section length can be of twice length of retry loop under G-EDF/LCM. While under G-RMA/LCM, atomic section length can exceed length of retry loop.

We implement the contention managers in the Rochester STM framework and conduct experimental studies using a multicore real-time Linux kernel. Our studies confirm that, the contention managers achieve orders of magnitude shorter retry costs than lock-free synchronization. Among the contention managers, PNF performs the best.

Building upon these results, we propose real-time contention managers that allow nested atomic sections – an open problem – for which STM is the only viable non-blocking synchronization solution. Optimizations of LCM and PNF to obtain improved retry costs and greater schedulability advantages are also proposed.

Bibliography