

Concurrency control is important for real-time systems. Real-time tasks may conflict on shared objects. Without proper real-time concurrency control, tasks can miss their deadlines. Past work on real-time concurrency control is mostly based on locking and lock-free algorithms- with little attention to transactional memory (TM)- for real-time concurrency control. Lock-based concurrency control suffers from programmability, scalability, and compositionality challenges. These challenges are exacerbated in emerging multicore architectures, on which improved software performance must be achieved by exposing greater concurrency. Lock-free offer numerous advantages over locks (e.g., deadlock-freedom), but their programmability has remained a challenge. Past studies show that they are best suited for simple data structures where their retry cost is competitive to the cost of lock-based synchronization.

In this dissertation proposal, we consider software transactional memory (STM) for concurrency control in multicore real-time software. Doing so will require bounding transactional retries. Retry bounds in STM are dependent on the contention manager (CM) policy at hand. We investigate and design a number of real-time CMs. The first two CMs are directly based on dynamic and static priority of underlying tasks. Earliest Deadline-First CM with G-EDF scheduler (ECM) resolves conflicts based on absolute deadline of the underlying instances. Rate Monotonic Assignment with G-RMA scheduler (RCM) resolves conflicts based on period of underlying instances.

ECM and RCM conserve the semantics of the underlying real-time scheduler. This conservative approach can result in a maximum transaction retry cost due to another transaction of double the maximum atomic section length. So, we design another CM named Length-based CM (LCM). LCM not only considers priority of transactions, but also relative length of the interfering transaction to remaining length of interfered transaction. LCM is used with G-EDF and G-RMA. By proper choice of different parameters, retry cost under LCM is kept lower than retry cost under ECM and RCM. ECM, RCM and LCM are affected by transitive retry in case of multiple objects per transaction. Transitive retry enforces a transaction to abort and retry due to another non-conflicting transaction. So, we develop the Priority-based with Negative value and First access (PNF) CM to avoid transitive retry. PNF also optimizes processor usage by lowering priority of the job holding a retrying transaction. Thus, other jobs can proceed if there is no conflict. For each contention manager, we upper bound retry cost and response time. We analytically compare CMs' schedulability against each other, and against lock-free algorithm. We identify the conditions under which each CM is preferred. We experimentally measure retry cost and response time of mentioned contention managers, as well as lock-free, on our testbed.

We have focused on the design of real-time contention managers for non-nested transactions on multicore systems. Fundamental and open problems that we propose to solve (post-prelim) include designing real-time contention managers for nested transactions in multicore systems. In addition, analytical and experimental comparison between developed contention managers and locking protocols is also proposed.

Bibliography