Contents

Chapter 1

Contention Management

Concurrent programs that make strong progress guarantees are scalable but those that require centralised transaction management to ensure progress are not. Transactional Memory systems centralise the responsibility for scheduling and contention management, at the expense of scalability. This chapter explains why scalable concurrent programs should make strong progress guarantees. It also explains how a load-balancing scheduler, intended for use with a parallel workload, can be used to schedule Memory Transactions.

Section 6.1 identifies the choice of contention management mechanism as one of the most significant decisions taken when designing a concurrent system.

Section 6.2 describes how Immutable Data Structures can be used to implement non-blocking algorithms.

Section 6.3 compares a non-blocking Producer Consumer Queue with its blocking counterpart.

Section 6.4 describes how Memory Transactions can be load-balanced by a scheduler intended for a parallel workload.

Bibliography