Parallelizing Sequential Tasks:

Need to efficiently support the parallelization of tasks with partially or totally ordered data. Pipeline-parallel organizations avoid contention and stalls caused by intra-thread synchronization other organizations.

Proof Sketch

The references [1] prove that “in the program order of the consumer, the consumer dequeues values in the same order that they were enqueued in the producer’s program order,” for strong to weakly ordered consistency models, showing that FastForward works even on relaxed memory models.


FastForward for Efficient Pipeline Parallelism

Parallelization of Sequential Tasks:

Need to efficiently support the parallelization of tasks with partially or totally ordered data. Pipeline-parallel organizations avoid contention and stalls caused by intra-thread synchronization other organizations.

The references [1] prove that “in the program order of the consumer, the consumer dequeues values in the same order that they were enqueued in the producer’s program order,” for strong to weakly ordered consistency models, showing that FastForward works even on relaxed memory models.