Combining-Based Concurrent Double-Ended Queues

Fatourou Panagiota, Papadogiannakis Petros, Nikolaos Arvanitakis, Nikolaos Giachoudis

- Stacks and queues are widely used in many settings: OS kernels, memory management, synchronization, scheduling, load balancing, resource sharing.
- Deques have enhanced functionality. They can be used as stacks or queues, or as a combination of both.
- Additional applications. They are used when implementing a web browser's history, application's list of undo operations, in graph traversal algorithms, task management, queuing and caching systems.

Motivation

- Synchronization is required between threads operating on the same and on different sides of the deque.
- Deques are more complex than queues and stacks: they support additional functionality, and thus require implementing more operations.

Implementation

- Employs SoTA software combining algorithms: CC-Synch (Fatourou and Kallimanis PPoPP 2012).
- It uses one instance of CC-Synch for each of its two endpoint.

Software Combining

- Every thread that requires mutual exclusion inserts a new node with the wait bit marked as TRUE and completed marked as FALSE. It is assigned the node pointed to by the previous value of Tail.
- Thread announces its request in the fields of its assigned node.

Double-Ended Queues

- Two algorithms have been designed and tested.
- The proposed deques use elimination: concurrent enqueues & dequeues on the same endpoint get eliminated.

Experiments

- Deques with elimination outperform by far CCQueue, the state-of-the-art FIFO Queue.
- Deques without elimination exhibit almost as good performance as CCQueue, despite the additional functionality they support.

Contact Info

Petros Papadogiannakis
csd4485@csd.uoc.gr

This project has received funding from the Hellenic Foundation for Research and Innovation under the 2nd Call for H.F.R.I.’s Research Projects to Support Faculty Members & Researchers, Agreement No 03684.