Declarative Concurrent Data Structures

Ioanna Tsakalidou, Hamish Nicholson, Aunn Raza, and Anastasia Ailamaki

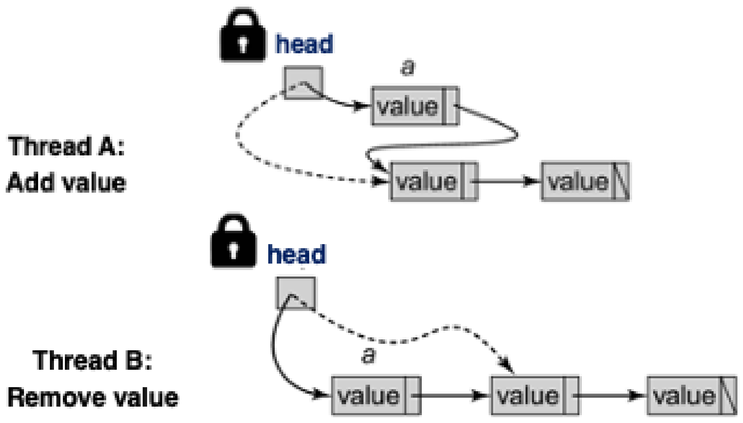
EPFL

Design Of Concurrent Data Structures Is Workload-Dependent

Use Case: List Used For A Scheduling Mechanism Criteria

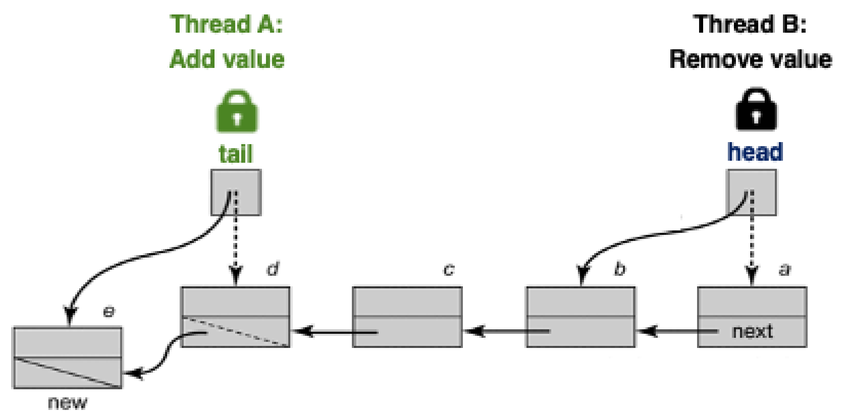
Stack/LIFO

Prioritize Tasks Arriving First



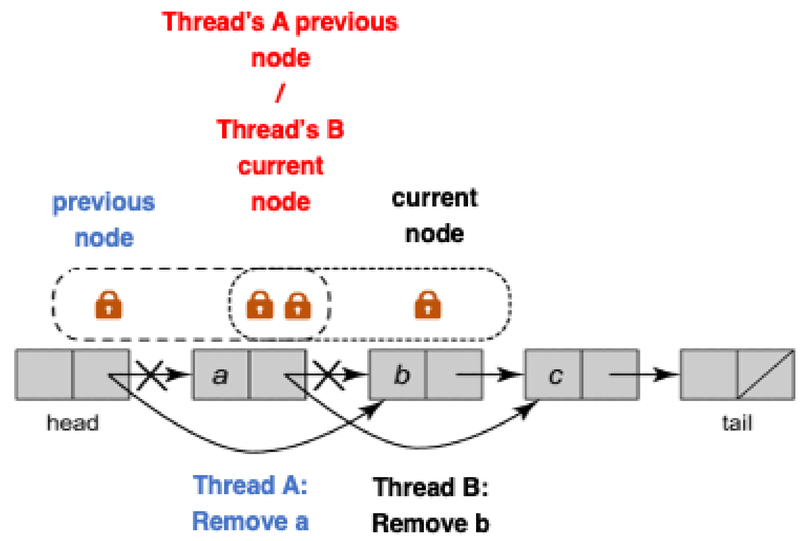
Queue/FIFO

Prioritize Tasks Arriving Last

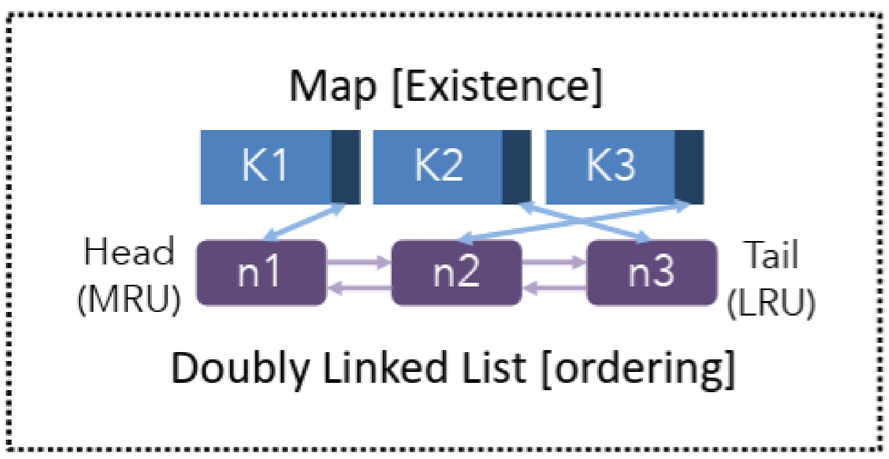


Linked List With Mutable Nodes

Prioritize Tasks Based On Duration



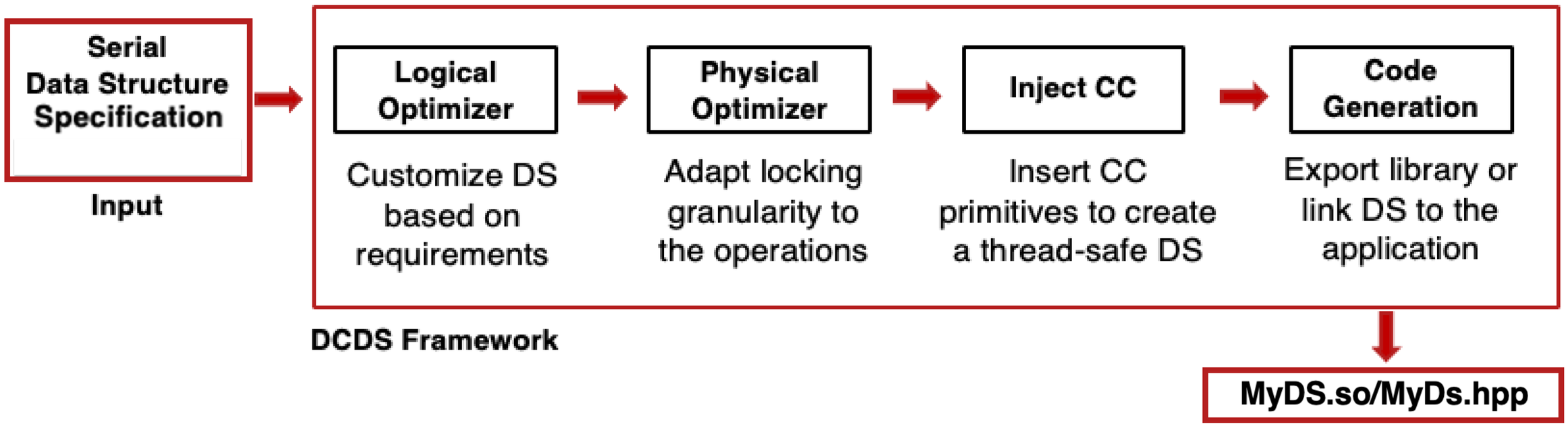
Composing Concurrent Data Structures Requires A Synchronization Wrapper For Atomicity



The Synchronization Wrapper

- serializes operations with a lock guard
- gives up performance benefits of individual data structures

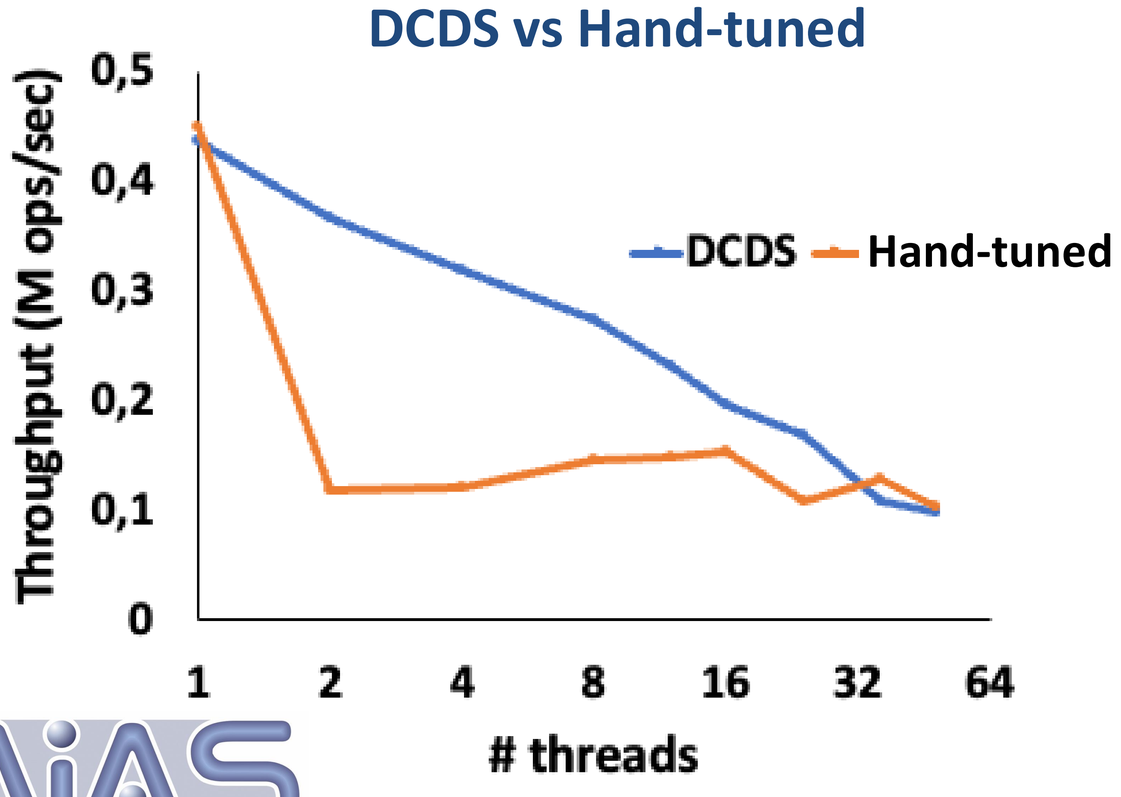
Automatic Development Of Concurrent From Sequential Code Through Declarativity



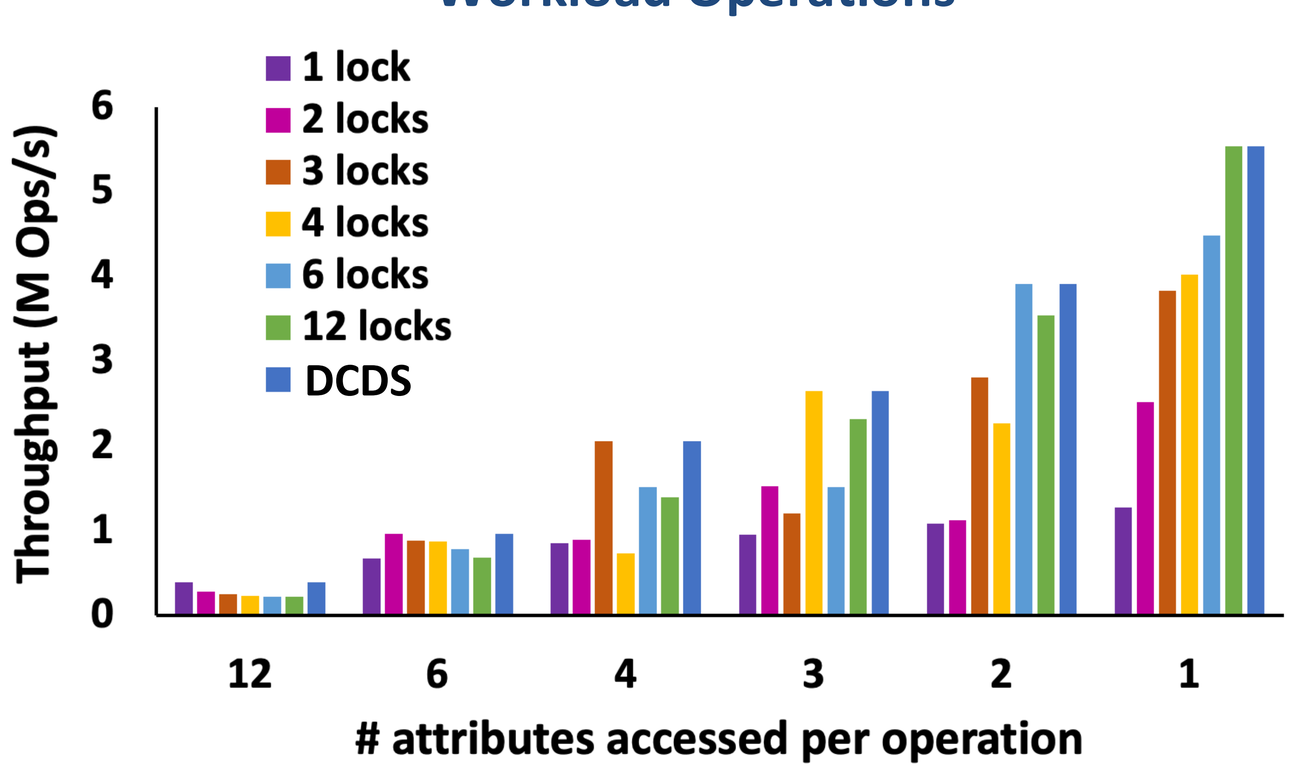
DCDS achieves **ease of development** and **performance** through declarativity and build-time specialization

Up to 4.5x Speedup With Build-Time Specialization In DCDS

Scalability of LRU Container Generated By DCDS vs Hand-tuned



Best Locking Granularity Depends On Workload Operations



DATA-INTENSIVE APPLICATIONS AND SYSTEMS