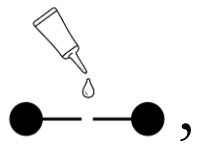
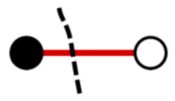
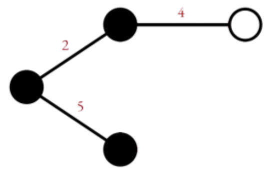
Concurrent Link-Cut Trees

Mihail Stoian | Advised by Jana Giceva and Philipp Fent

Technische Universität München

Introduction

Efficient , , and  at heart of

- maximum flow
- online minimum spanning tree

Motivation

Link-cut trees: *fastest* known solution.

But: considered unparallelizable.

Highlights

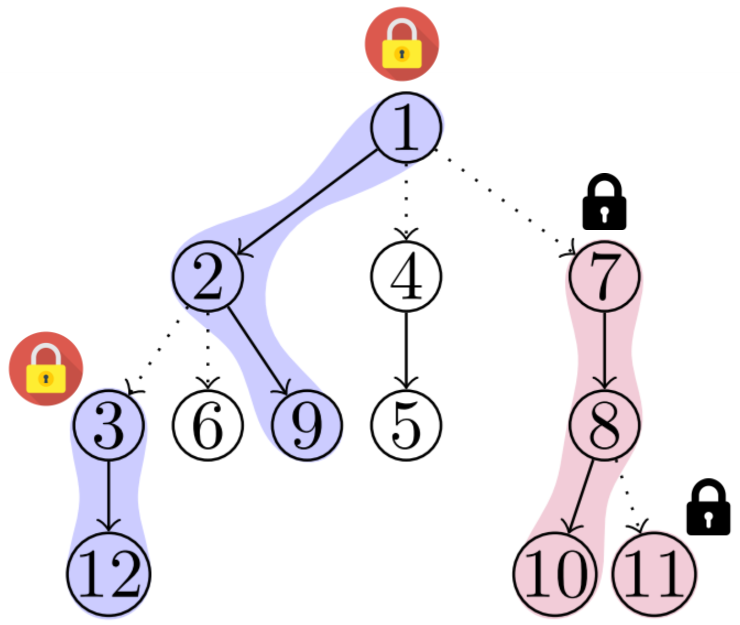
First parallel implementation since 40 years.

10x speedup.

Approach

Parallelize *only* the logic, not the auxiliary splay trees.

Lock preferred paths by locking the topmost node:

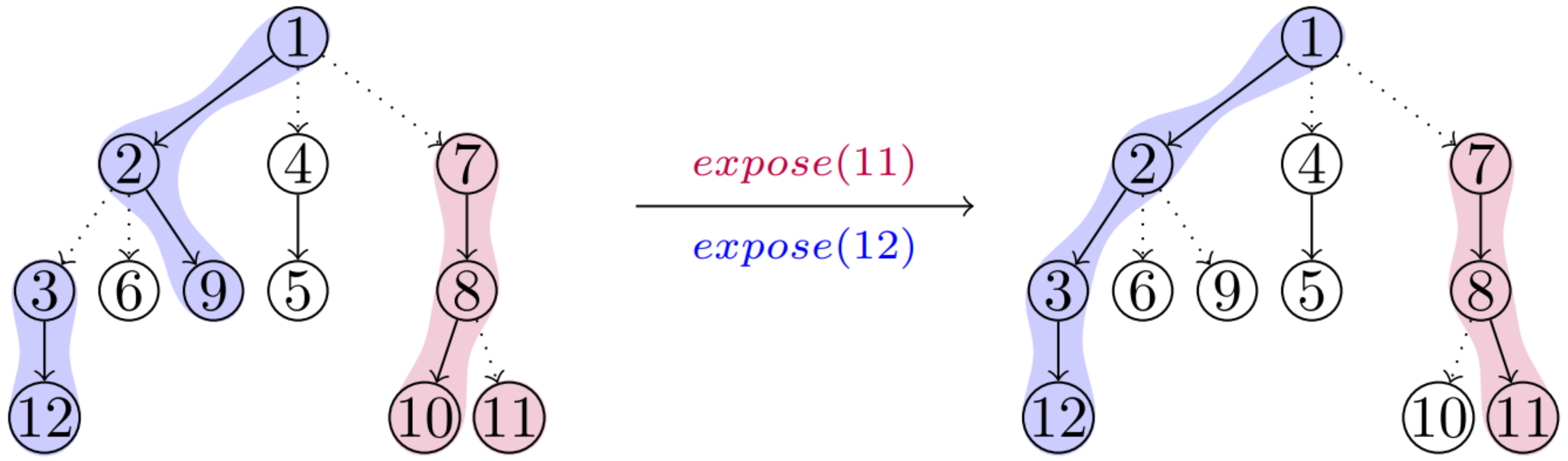


Parallelization scheme:

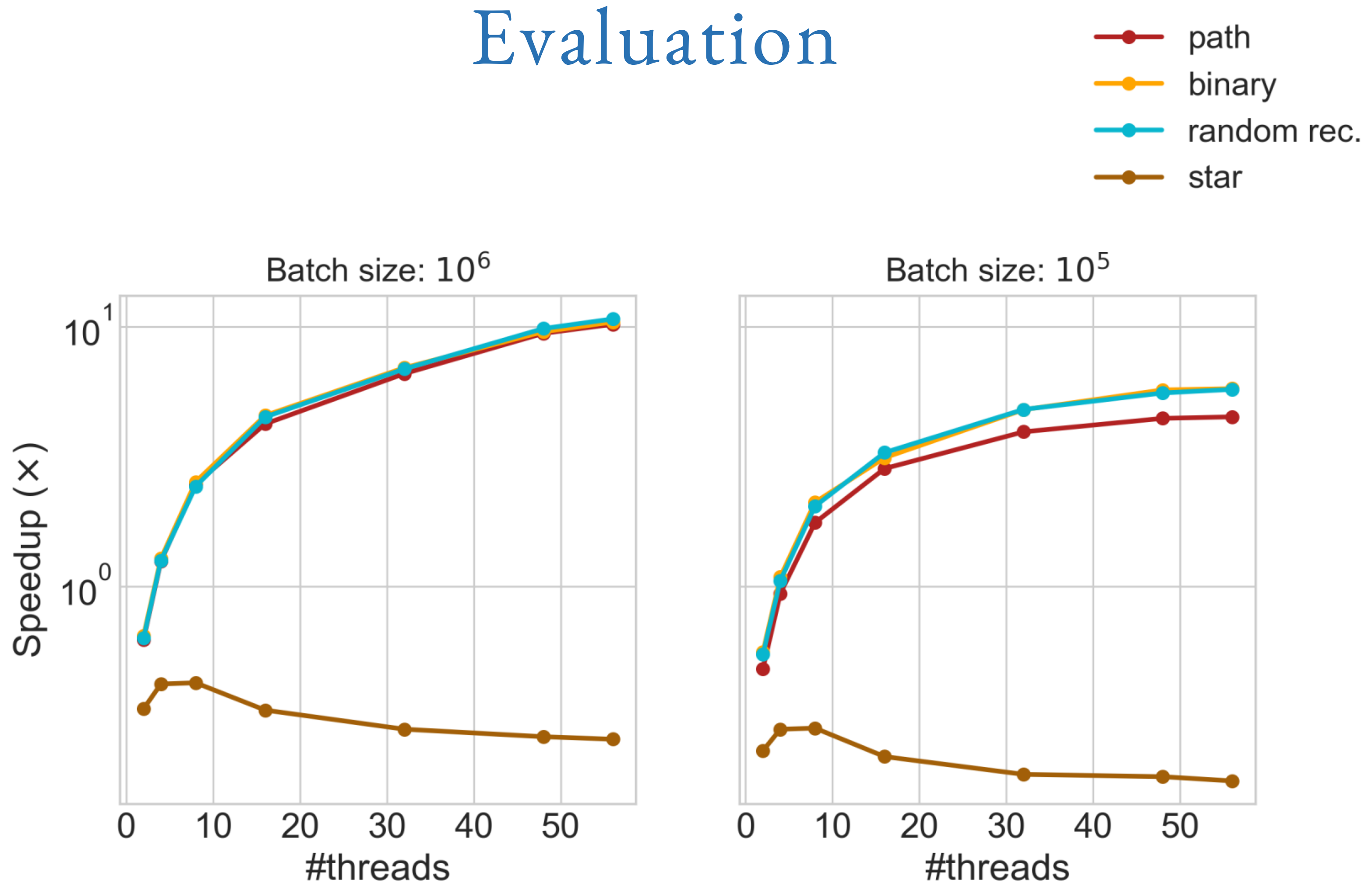
- use an auxiliary lock-free pointer array.
- once locked, check that topmost node did not change.
- perform specific operations on respective splay tree.
- release locks once the new preferred path has been formed.

Independent operations are performed in parallel.

Synchronize only when preferred paths cross:



Evaluation



10x speedup over sequential link-cut trees.

Star trees: root is a hotspot.

Future Work

Theoretical analysis.

Support for binary operations, e.g., lca.

Learned Concurrent Data Structures.

stoianmihail/concurrent-link-cut-trees

mihail.stoian@tum.de