

# Parallel Fringe Search

Christian Zeman & Lukas Mosimann

ETH Zürich

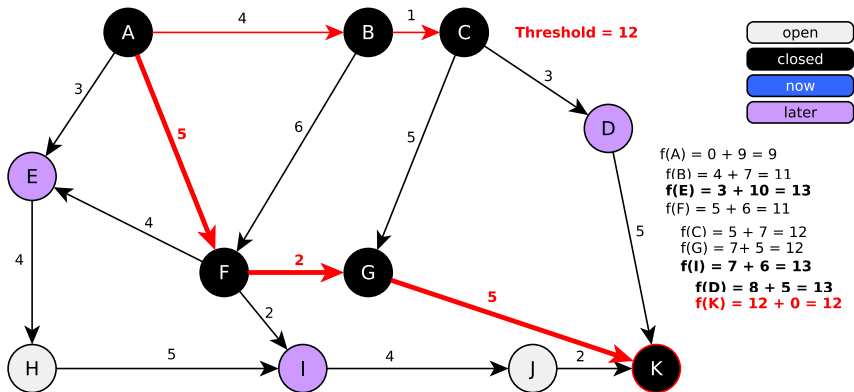
*Design of Parallel and High-Performance Computing*

December 14, 2013

# Overview

- 1 Algorithm
- 2 What we have done
- 3 Locking concepts
- 4 Benchmarks
  - Threshold vs. Path Length

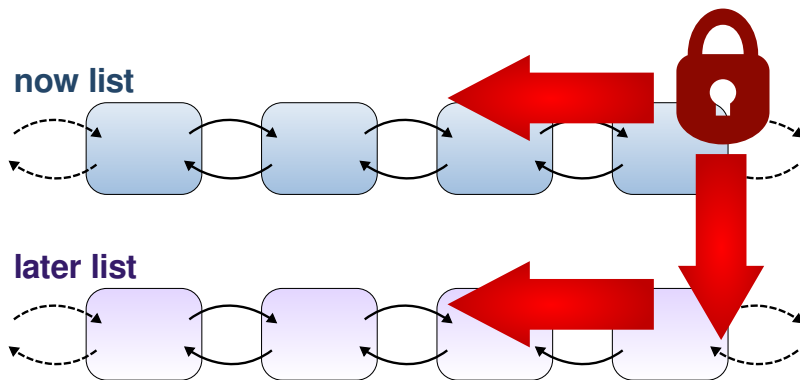
# Fringe Search



# What we have done

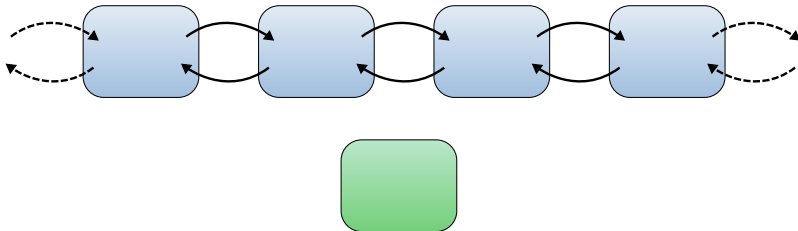
- Serial implementation of fringe search (much faster than Boost A\*)
- Parallel implementation with Open MP
  - 2 different locking concepts
  - Locks implemented using inline assembly (faster than Open MP locks)
- Benchmarking
  - Strong scaling
  - Weak scaling
  - Path quality

# Locking concept: Deadlock prevention

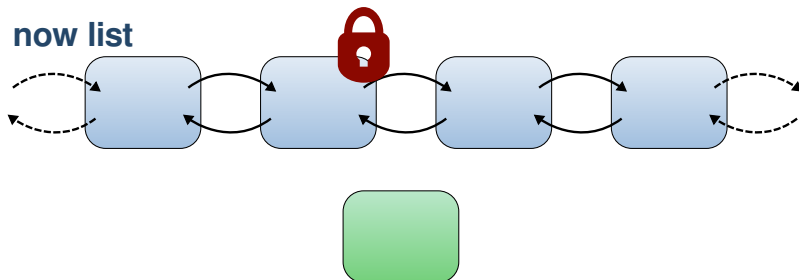


# Locking concept: Insert node

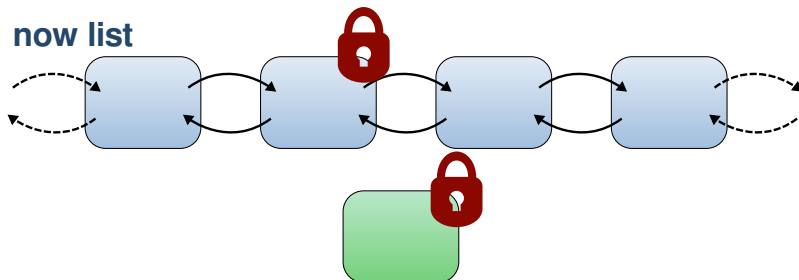
**now list**



# Locking concept: Insert node

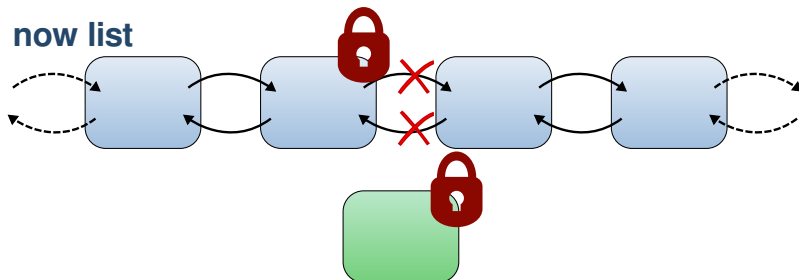


# Locking concept: Insert node

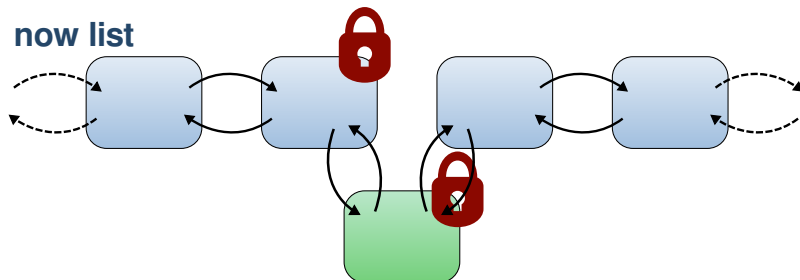




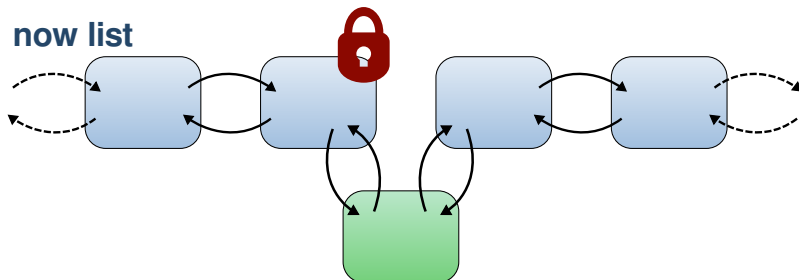
# Locking concept: Insert node



# Locking concept: Insert node

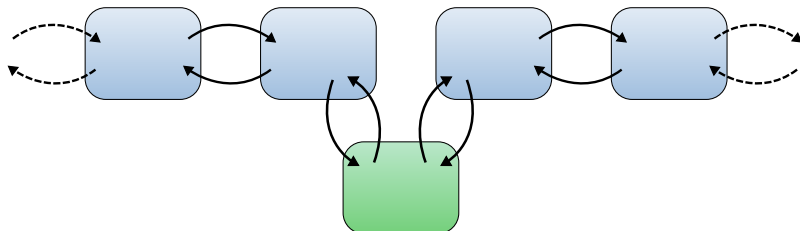


# Locking concept: Insert node



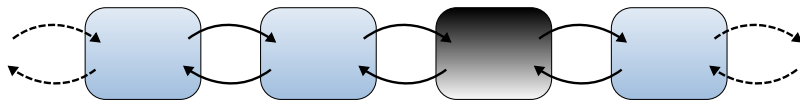
# Locking concept: Insert node

**now list**

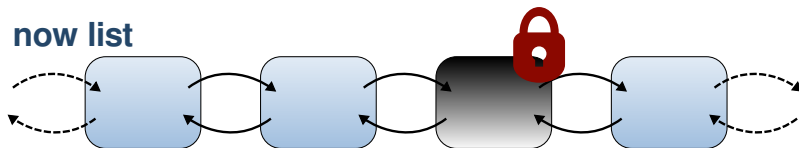


# Locking concept: Remove node

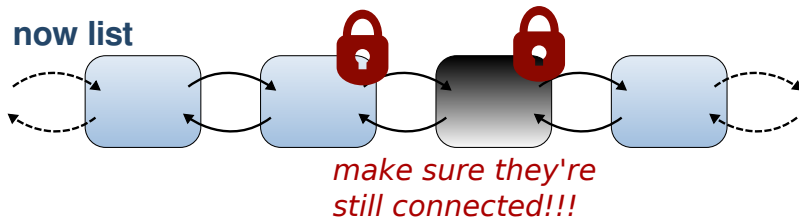
now list



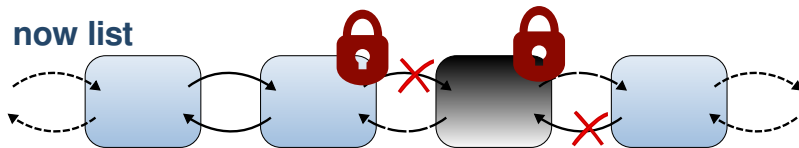
# Locking concept: Remove node



# Locking concept: Remove node

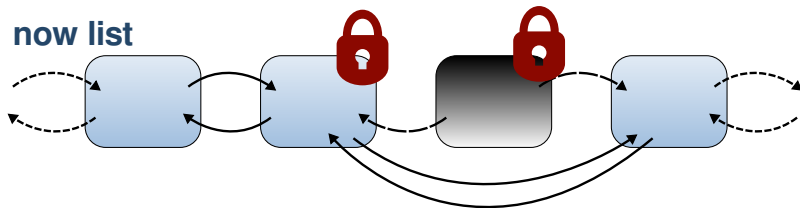


# Locking concept: Remove node

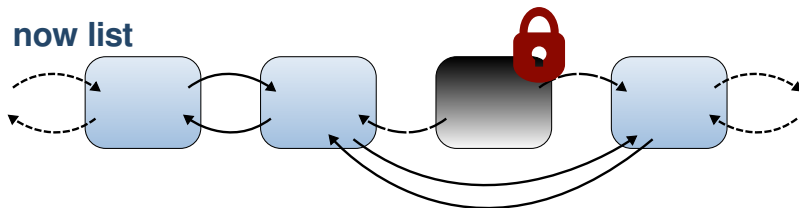




# Locking concept: Remove node

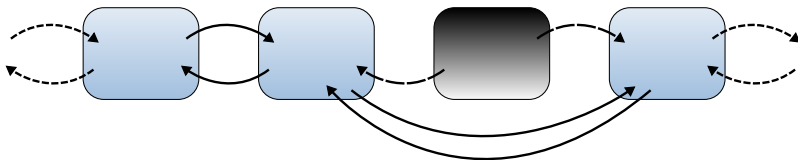


# Locking concept: Remove node



# Locking concept: Remove node

now list



# Locking concept: 2 concepts for removing nodes

Standard:

- Lock node and predecessor as shown before and remove it right away

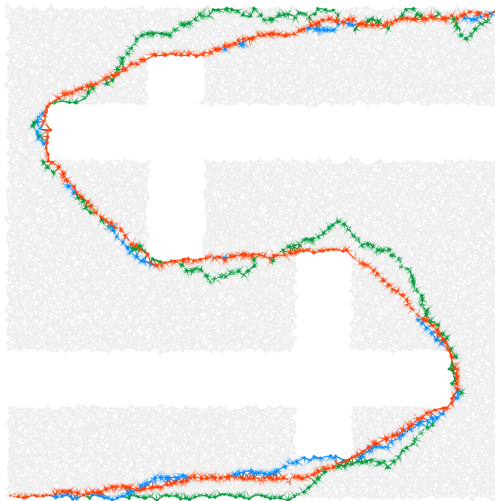
Lazy locking:

- Don't lock anything and just mark the node as removed
- Other threads will clean up and remove it later

# Threshold vs. Path Length

## Threshold update

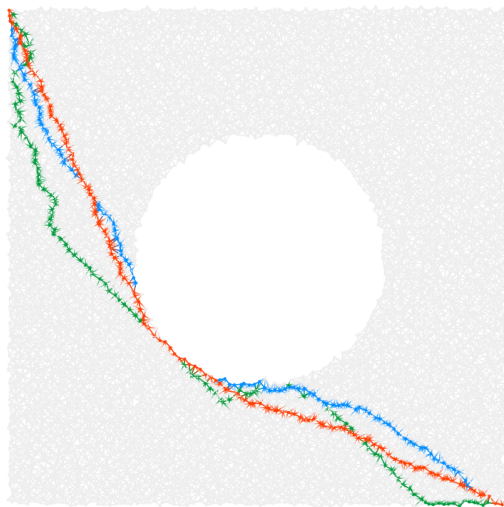
- Threshold  $+= 0.1$
- Threshold  $+= 1$
- Threshold  $+= 10$



# Threshold vs. Path Length

## Threshold update

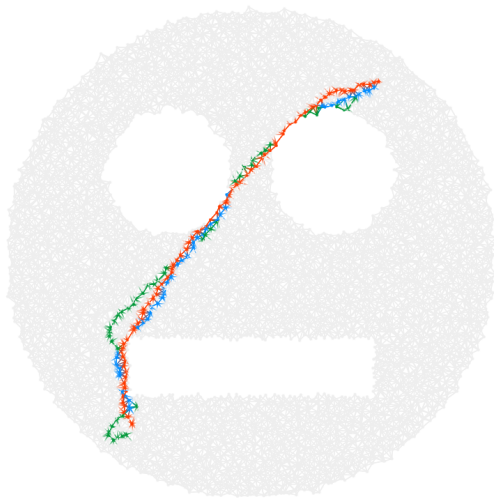
- Threshold  $+= 0.1$
- Threshold  $+= 1$
- Threshold  $+= 10$



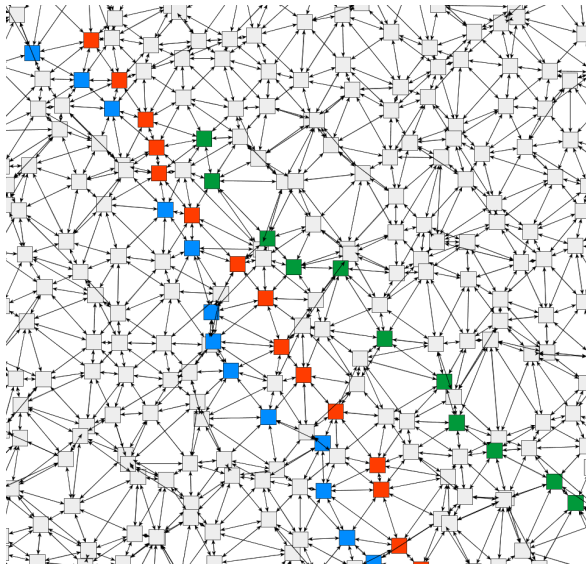
# Threshold vs. Path Length

## Threshold update

- Threshold  $+= 0.1$
- Threshold  $+= 1$
- Threshold  $+= 10$



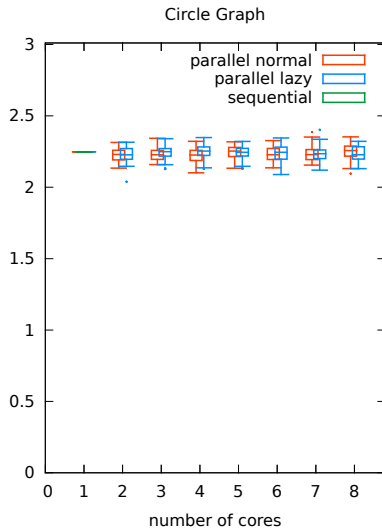
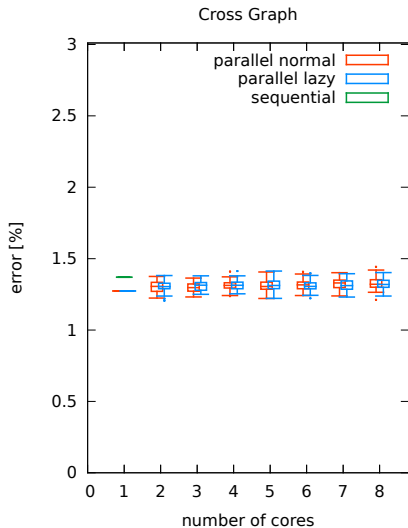
# Threshold vs. Path Length





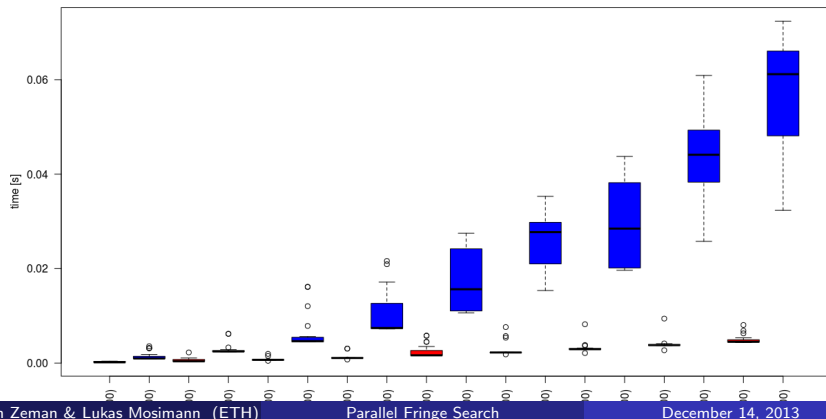
# Relative error path length (compared to $A^*$ )

relative Error, 2048 x 2048, threshold=1



# Benchmark runtime

- Runtime vs. A\* from Boost Graph Library
- Using "cross graph" and 400 - 10'000 nodes
- Threshold  $\pm = 1$
- 20 runs per size
- Intel Core 2 P9600 @ 2.53 GHZ



# Benchmark path length

- Runtime vs. A\* from Boost Graph Library
- Using "cross graph" and 400 - 10'000 nodes
- Threshold  $\pm 1$
- 20 runs per size
- Intel Core 2 P9600 @ 2.53 GHZ

# Benchmark path length

# Nodes	A-Star	Fringe	Difference	Difference in %
400	69.731	70.6903	0.9593	1.37 %
900	101.071	102.638	1.567	1.55 %
1600	133.304	136.326	3.022	2.27 %
2500	165.47	167.891	2.421	1.46 %
3600	195.363	198.115	2.752	1.41 %
4900	227.957	231.554	3.597	1.58 %
6400	258.5	261.49	2.99	1.16 %
8100	288.961	293.483	4.522	1.56 %
10000	321.263	325.853	4.59	1.43 %

What we still have to do:

- Implement parallel version
- Benchmarking (strong/weak scaling)
- Threshold fine-tuning
- Write report

# The End