Application of Concurrency on Identifying Connected Components using Parallel Variants of Breadth First Search and Label Propagation

Group DEF: AJ Bulthuis, Andy Sauerbrei, and Tyrone Wu

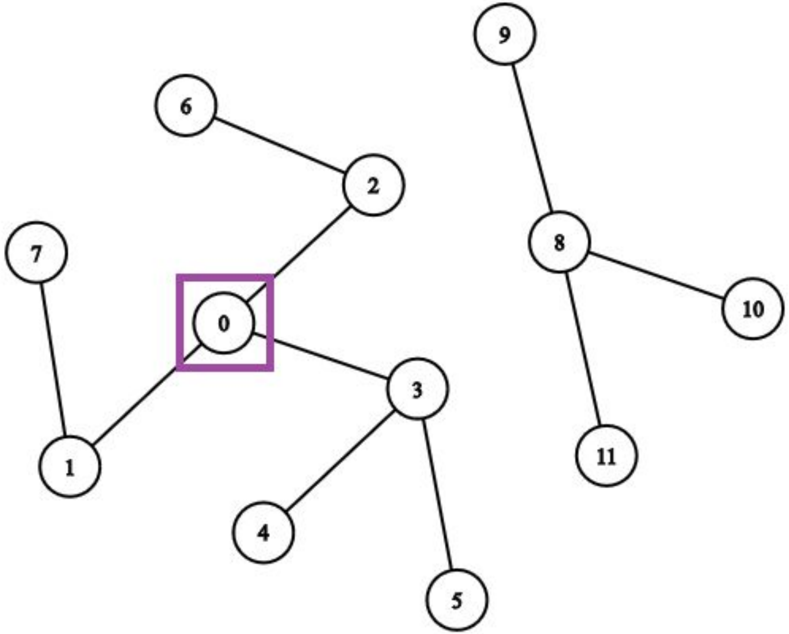
Walk-through of Parallel BFS

Parallel BFS (3 Threads)

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Connected Components:{0}

Frontier: {}

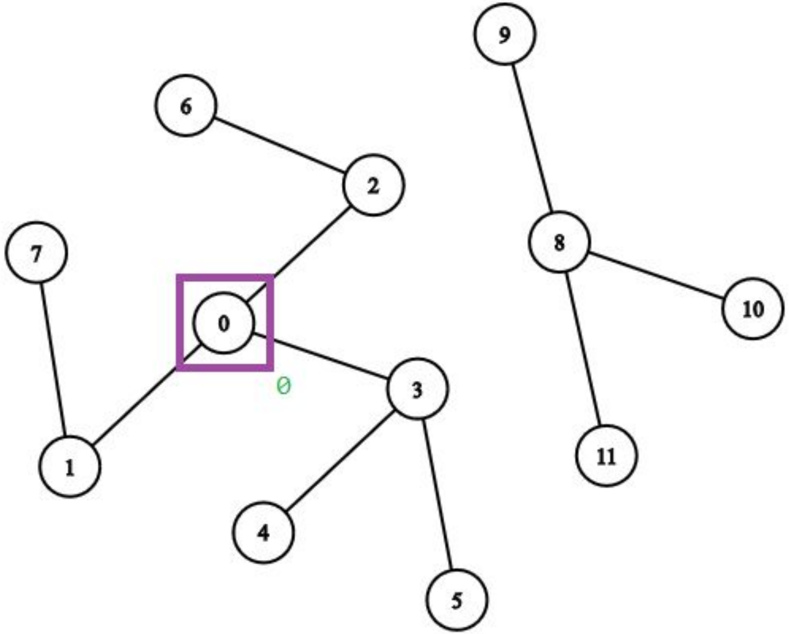


Parallel BFS (3 Threads)

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Connected Components:{0}

Frontier: {0}

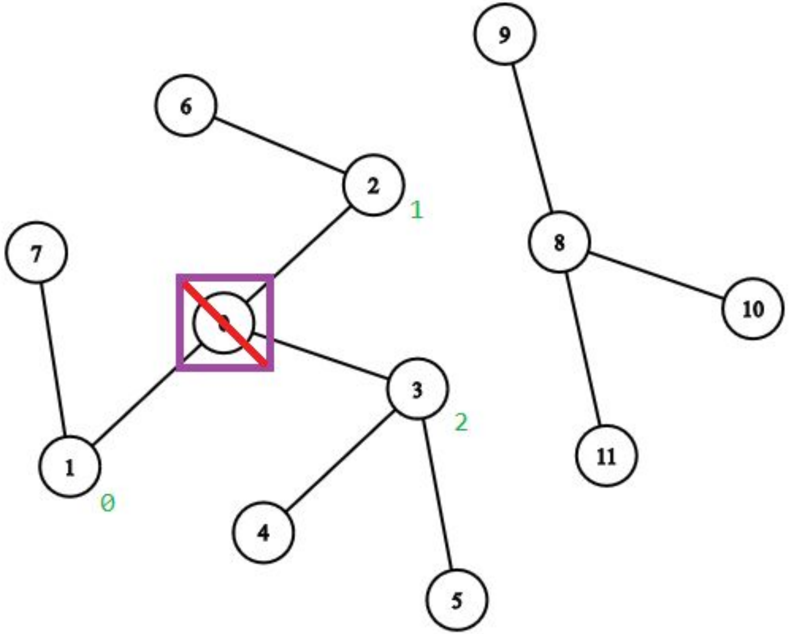


Parallel BFS (3 Threads)

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Connected Components:{0}

Frontier: {1,2,3}

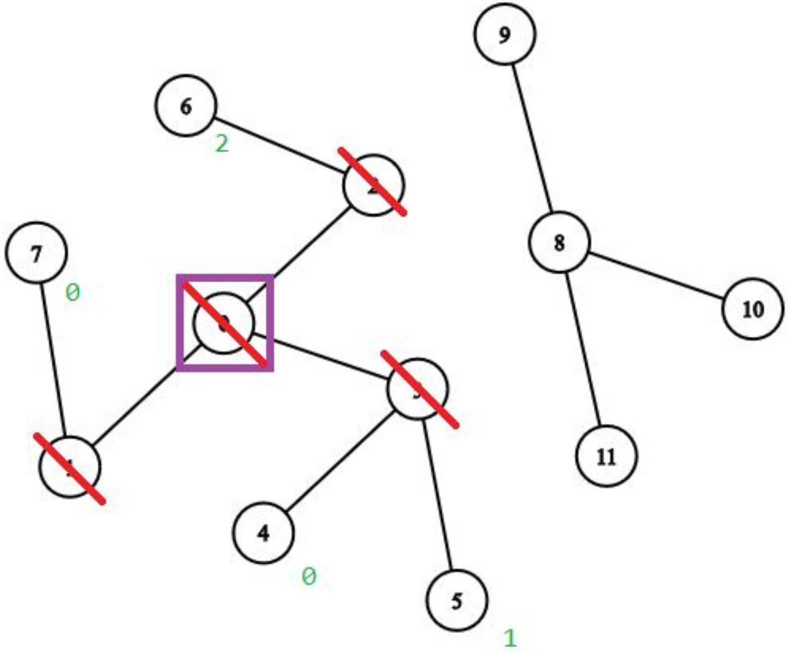


Parallel BFS (3 Threads)

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Connected Components:{0}

Frontier: {4,5,6,7}

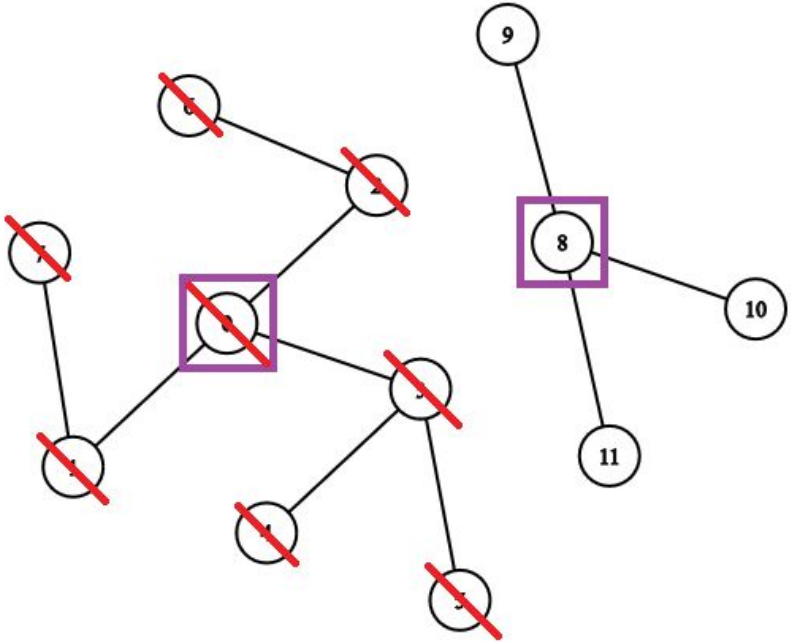


Parallel BFS (3 Threads)

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Connected Components:{0, 8}

Frontier: {}

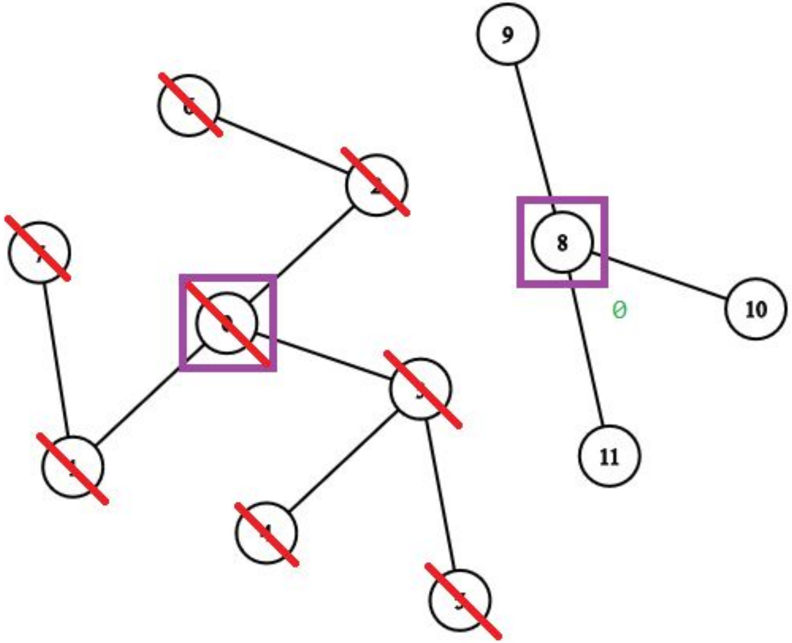


Parallel BFS (3 Threads)

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Connected Components:{0, 8}

Frontier: {8}

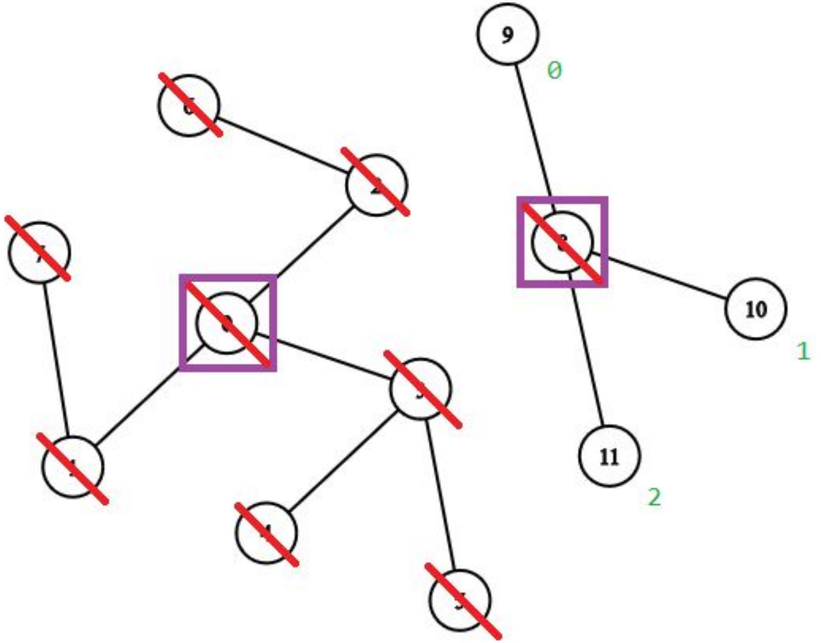


Parallel BFS (3 Threads)

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Connected Components:{0, 8}

Frontier: {9,10,11}

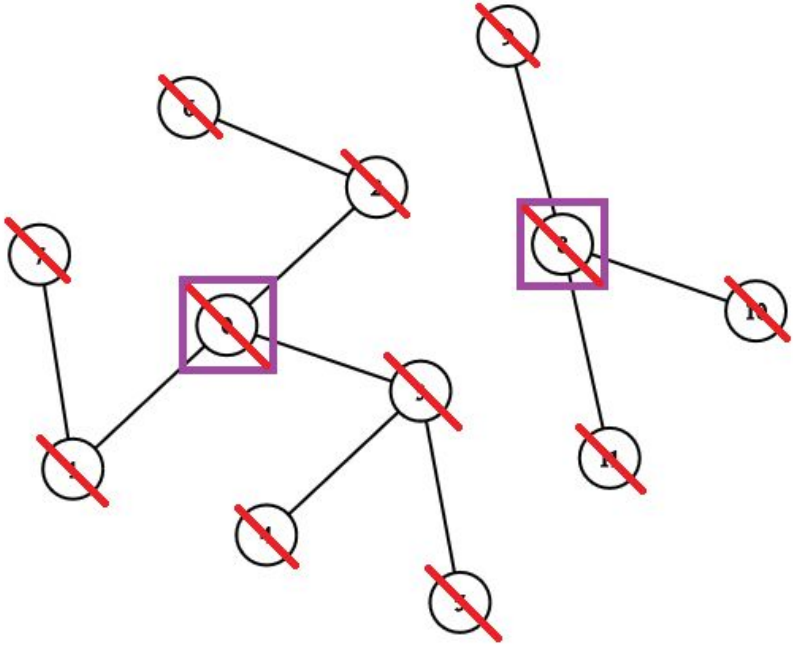


Parallel BFS (3 Threads)

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Connected Components:{0, 8}

Frontier: {}

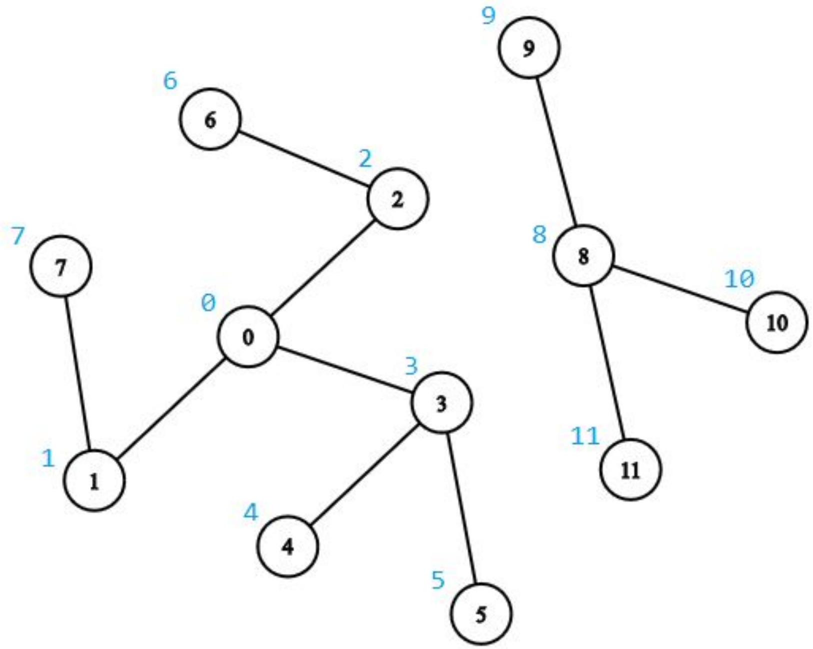


Walk-through of a Modified Label Propagation (Serialized)

Label Propagation

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

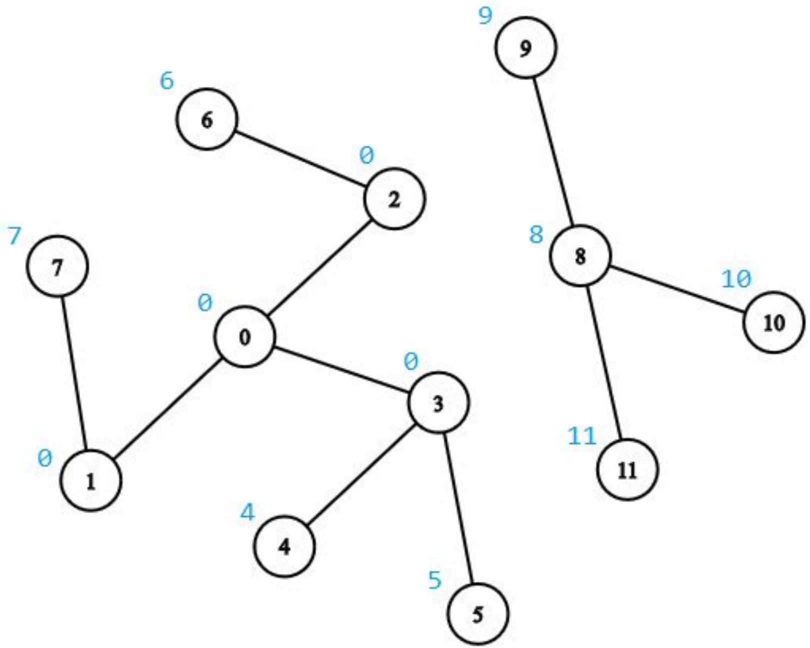
Label: {0,1,2,3,4,5,6,7,8,9,10,11}



Label Propagation

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

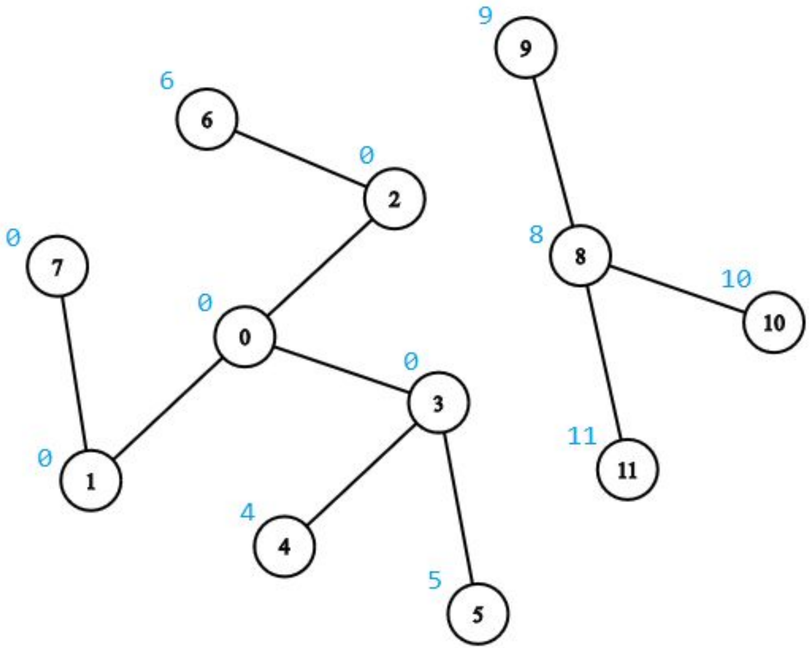
Label: {0,0,0,0,4,5,6,7,8,9,10,11}



Label Propagation

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

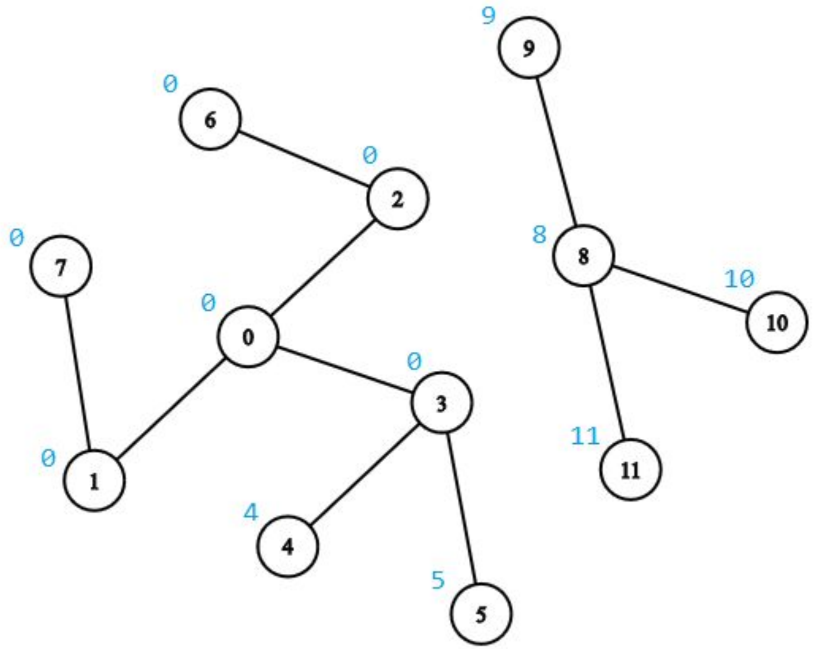
Label: {0,0,0,0,4,5,7,0,8,9,10,11}



Label Propagation

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

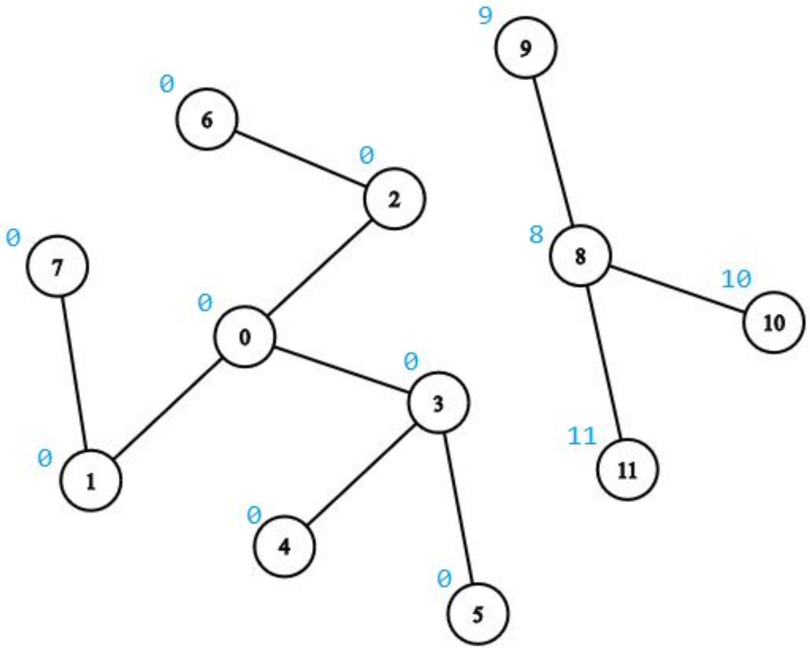
Label: {0,0,0,0,4,5,0,0,8,9,10,11}



Label Propagation

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Label: {0,0,0,0,0,0,0,0,8,9,10,11}

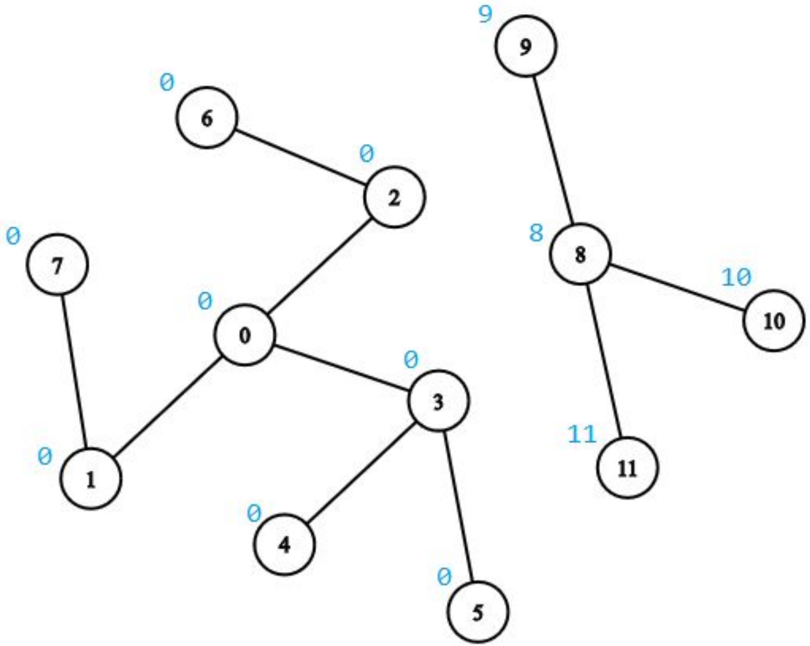


Label Propagation

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Label: {0,0,0,0,0,0,0,0,8,9,10,11}

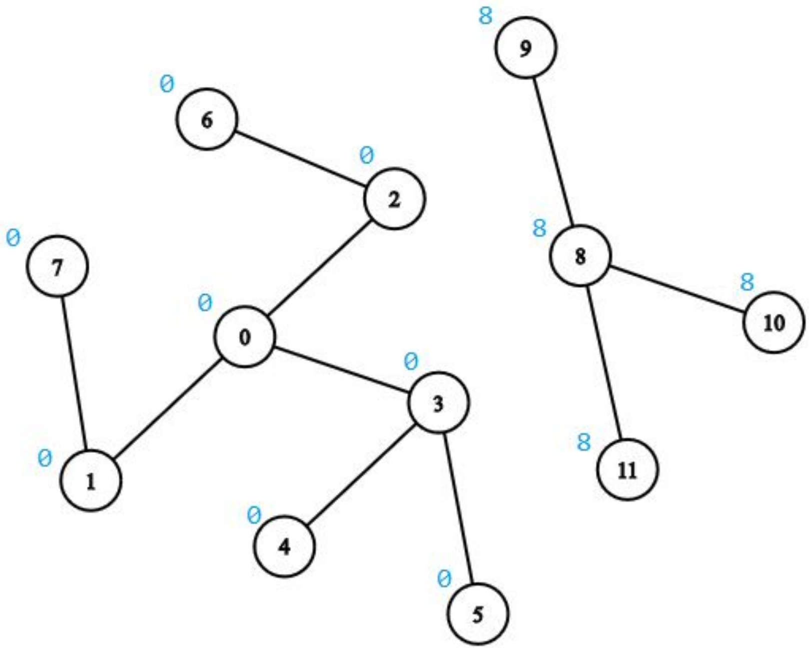
There is not change for the next 4 nodes: 4, 5, 6, and 7.



Label Propagation

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Label: {0,0,0,0,0,0,0,0,8,8,8,8}

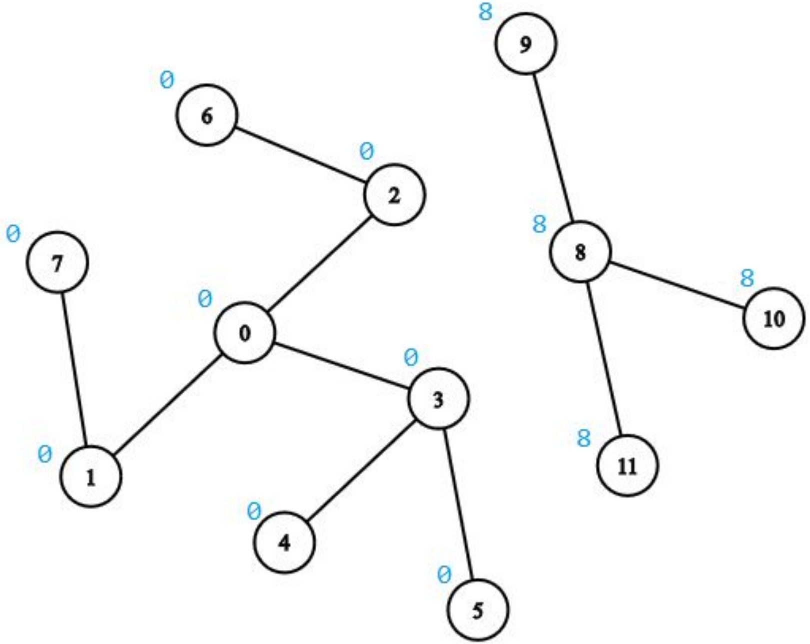


Label Propagation

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Label: {0,0,0,0,0,0,0,0,8,8,8,8}

There is not change for the next 3 nodes: 9, 10, and 11.

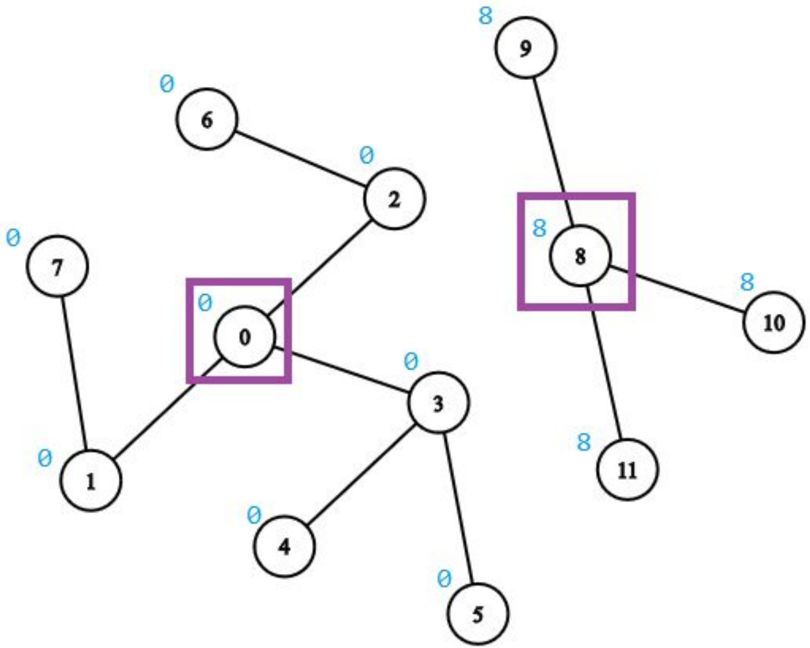


Label Propagation

Vertices: {0,1,2,3,4,5,6,7,8,9,10,11}

Label: {0,0,0,0,0,0,0,0,8,8,8,8}

Connected Components:{0, 8}



Discussion of Experiment

Experimental Design

Dataset

- 1 Connected Component of 102,000 Vertices
- 3 Connected Component each with 34,000 Vertices
- 6 Connected Component each with 17,000 Vertices

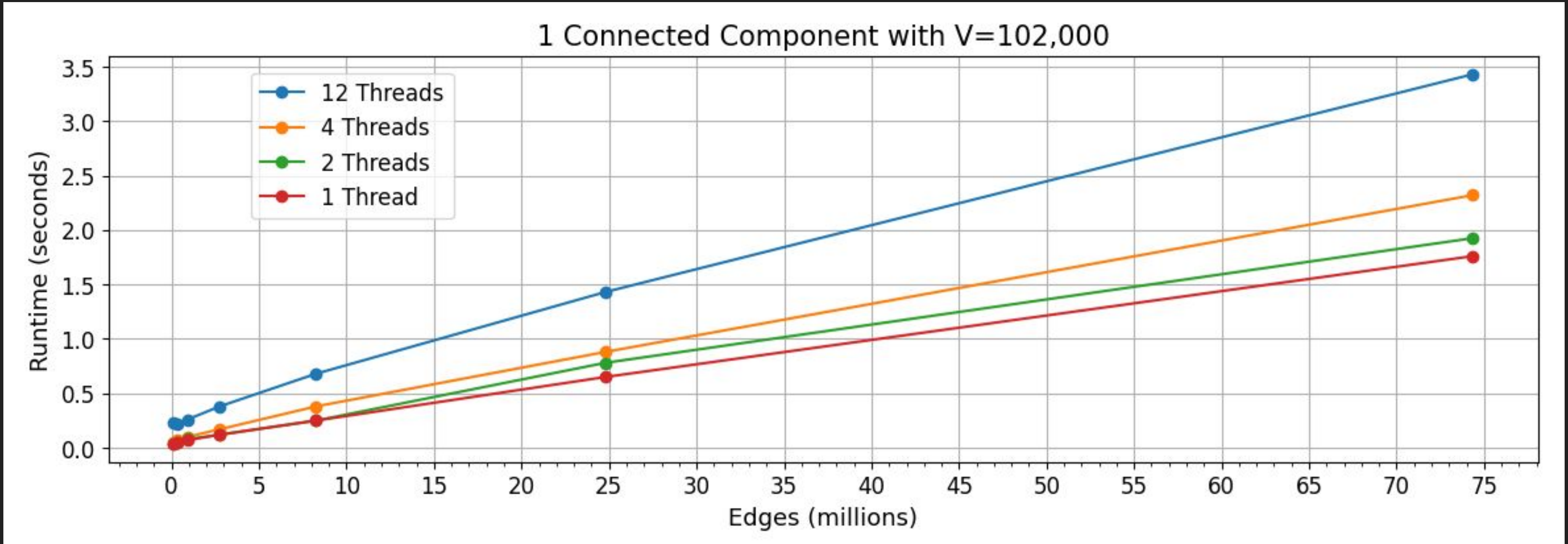
Questions:

- How does number of threads affect general runtime?
- How does density scale with runtime?
- How does parallel BFS and LP compare?

Discussion of Results

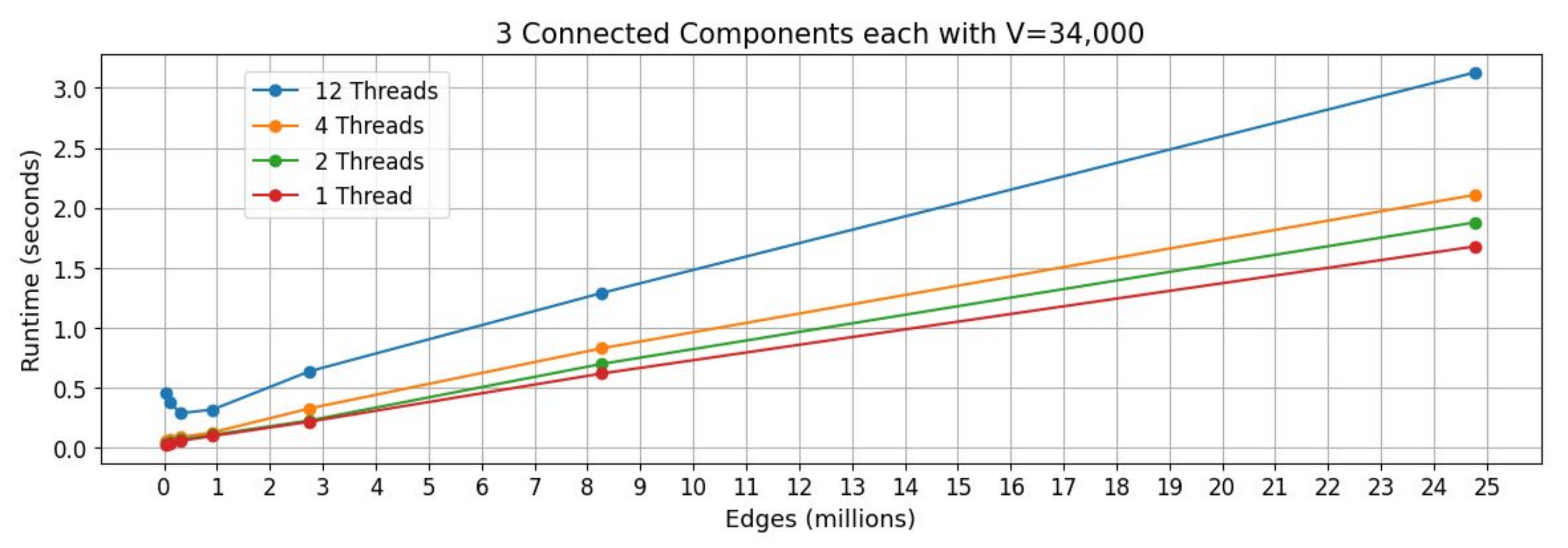
Frontier-Based BFS

- 1 Component with 102,000 Vertices



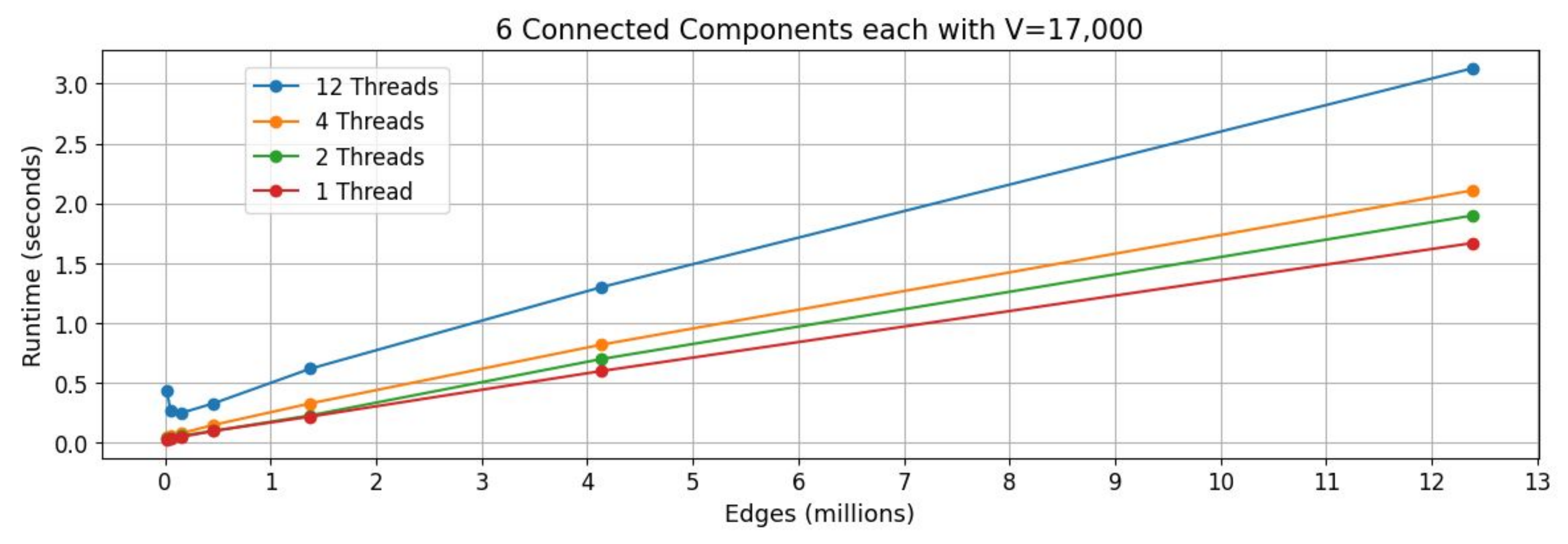
Frontier-Based BFS

- 3 Components each with 34,000 Vertices



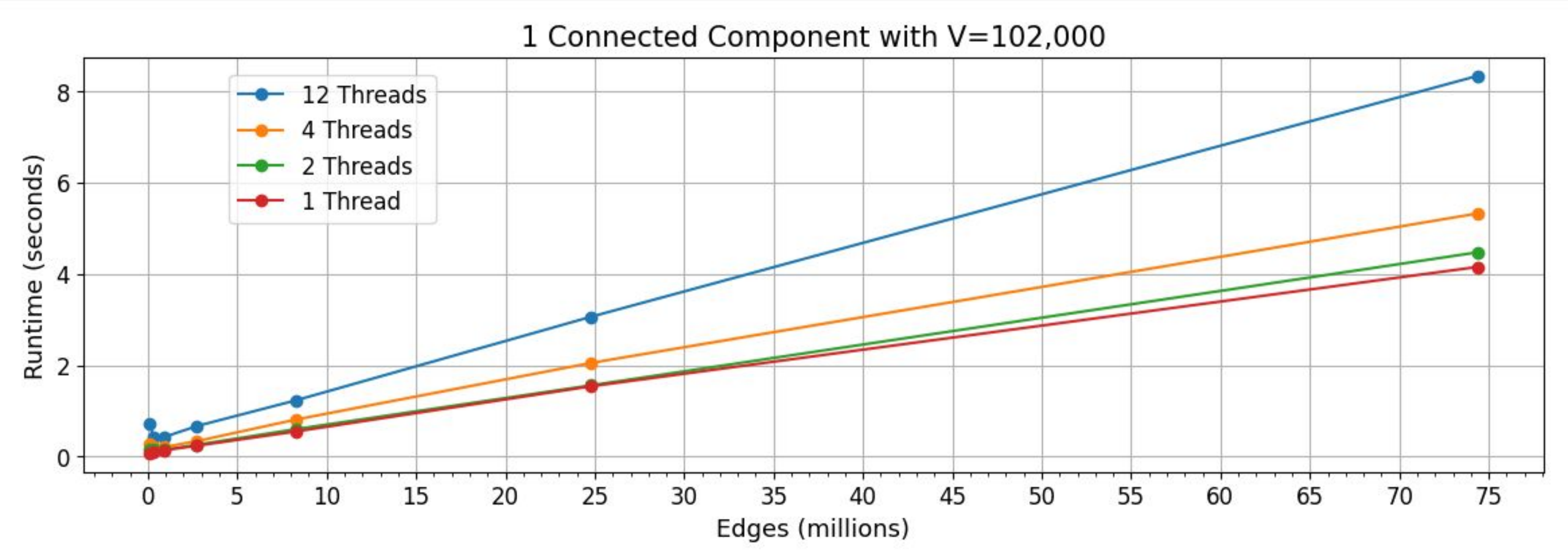
Frontier-Based BFS

- 6 Components each with 17,000 Vertices



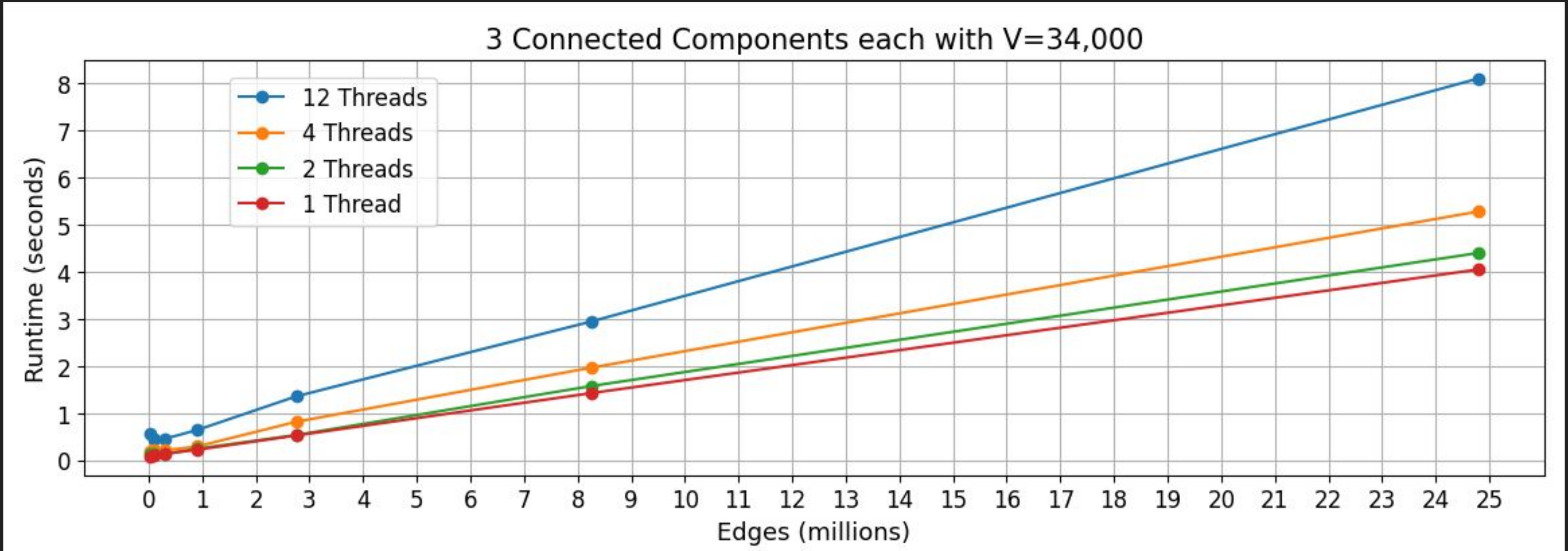
Label Propagation

- 1 Component with 102,000 Vertices



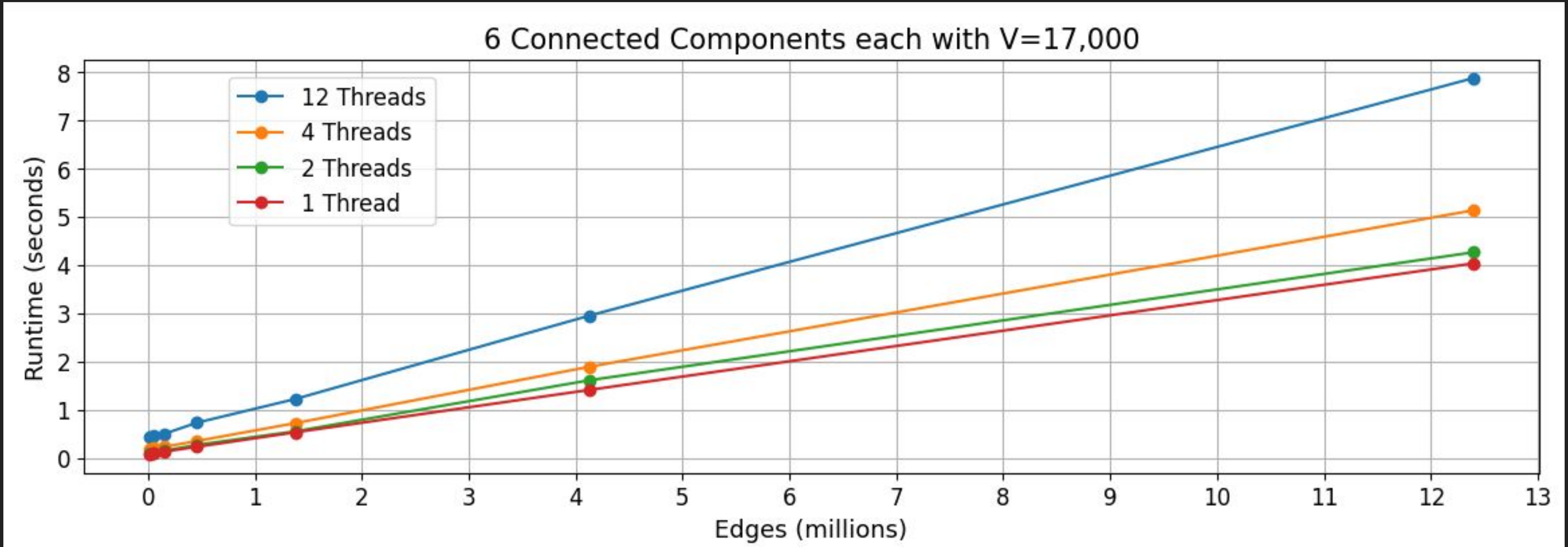
Label Propagation

- 3 Components each with 34,000 Vertices



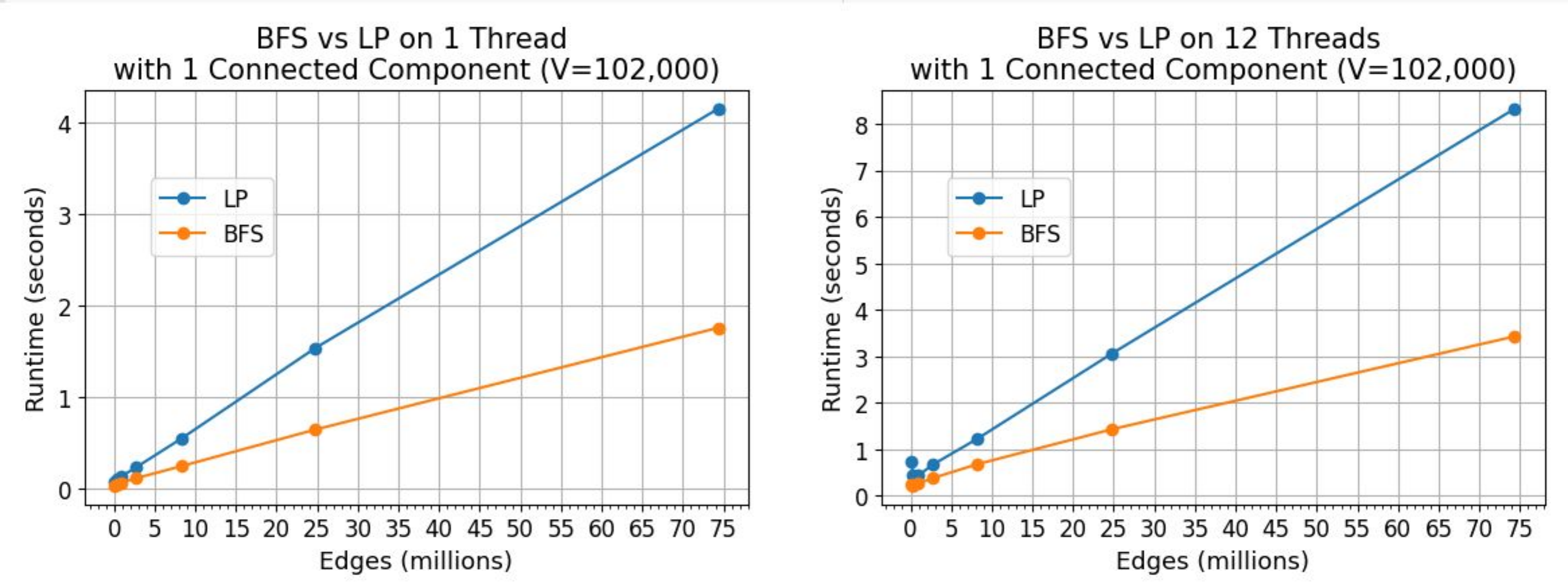
Label Propagation

- 6 Components each with 17,000 Vertices



Frontier BFS vs LP

- 1 Component with 102,000 Vertices
- 1 Thread vs 12 Threads

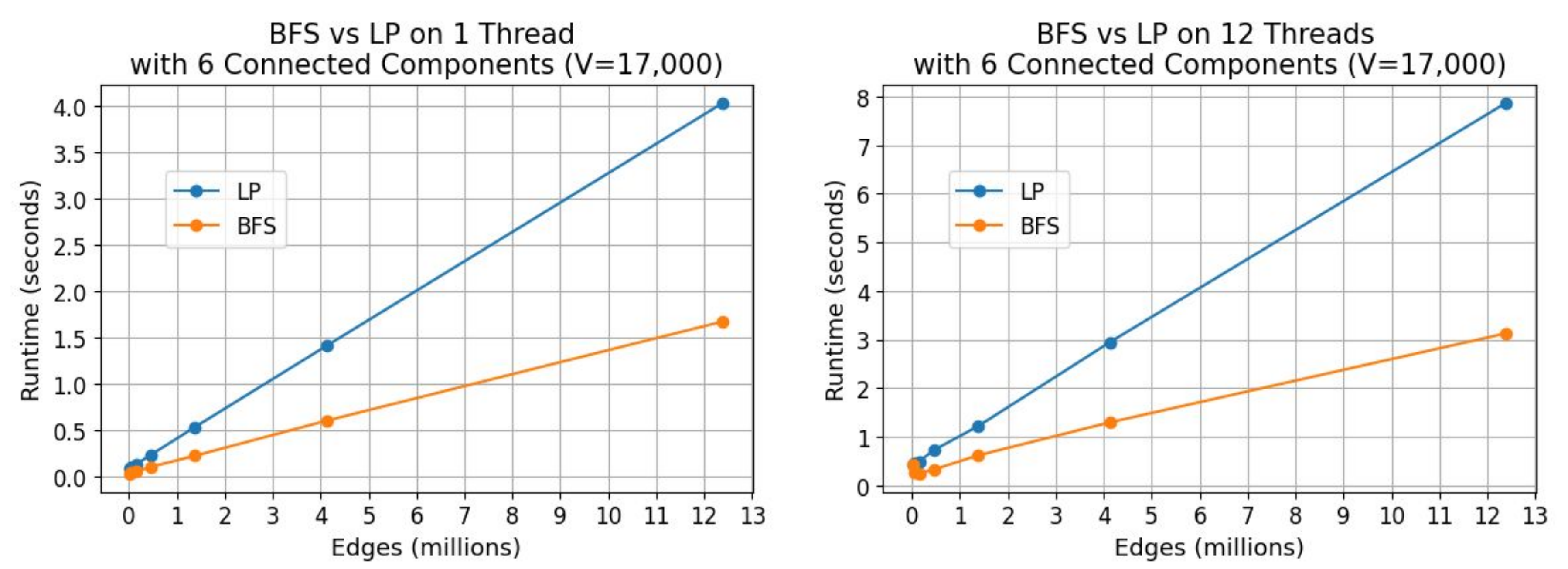


BFS vs LP on 1 Thread with 1 Connected Component (V=102,000)

BFS vs LP on 12 Threads with 1 Connected Component (V=102,000)

Frontier BFS vs LP

- 6 Components each with 17,000 Vertices
- 1 Thread vs 12 Threads



Questions?