HDagg: Hybrid Aggregation of Loop-carried Dependence Iterations in Sparse Matrix Computations

Behrooz Zarebavani[1], Kazem Cheshmi[1], Bangtian Liu[1], Michelle Mills Strout[2], and Maryam Mehri Dehnavi[1]

University of Toronto[1], University of Arizona[2]

Outline

- Overview
- HDagg: Hybrid aggregation of loop-carried dependence
  - HDagg's Objective
  - HDagg's Algorithm
    - Coarsening of densely connected vertices
    - Wavefront coarsening
- Results
- Conclusion

Outline

- Overview
- HDagg: Hybrid aggregation of loop-carried dependence
  - HDagg's Objective
  - HDagg's Algorithm
    - Coarsening of densely connected vertices
    - Wavefront coarsening
- Results
- Conclusion

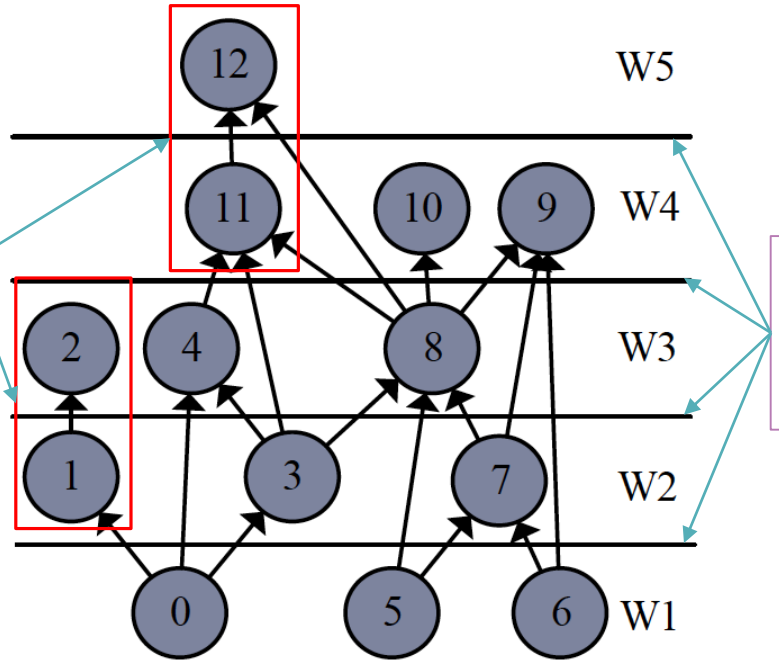
Loop Parallelization of Sparse Kernels

- Many sparse kernels have loop-carried dependences in their loop's iterations.
- Parallelizing these loops is difficult due to irregular memory accesses.
- State-of-the-art works, such as Wavefront techniques and LBC scheduling algorithms, do not consider the trade-off between locality, load balance, and synchronization.

Wavefront Technique

- Wavefront techniques provides a schedule which does not exploit reuse opportunities and has too many synchronization.

Data reuse opportunities between wavefronts are not exploited

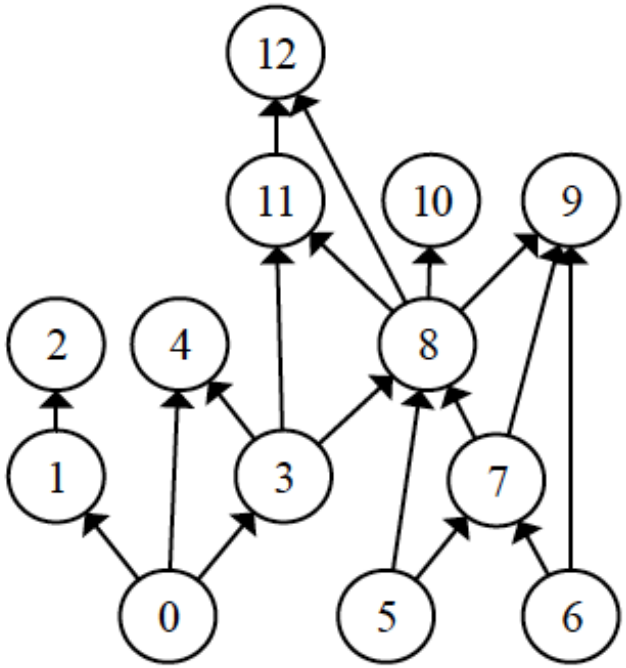


5 wavefronts causes too many synchronizations

The graph represents data dependency relations between the iterations in the sparse kernel.

LBC

- LBC schedule exploits some of the reuse opportunities. However, it reduces parallelism.

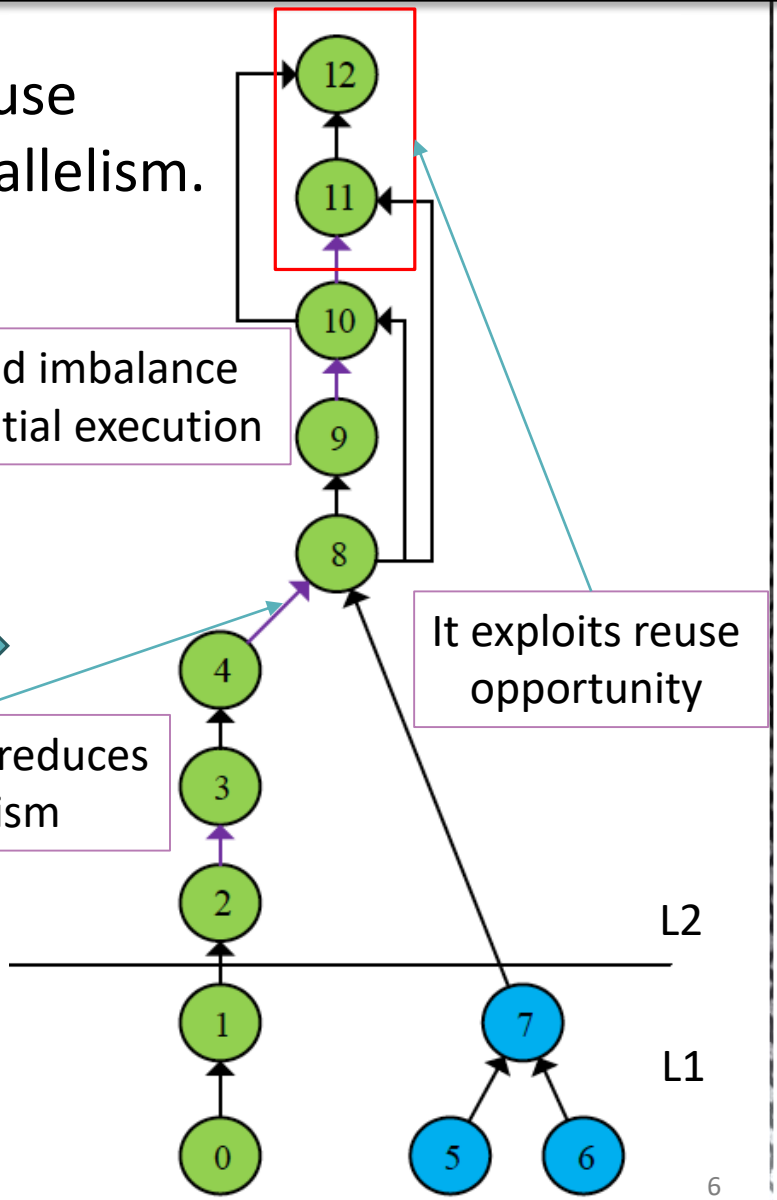


L2 causes load imbalance due to sequential execution

DAG to Tree

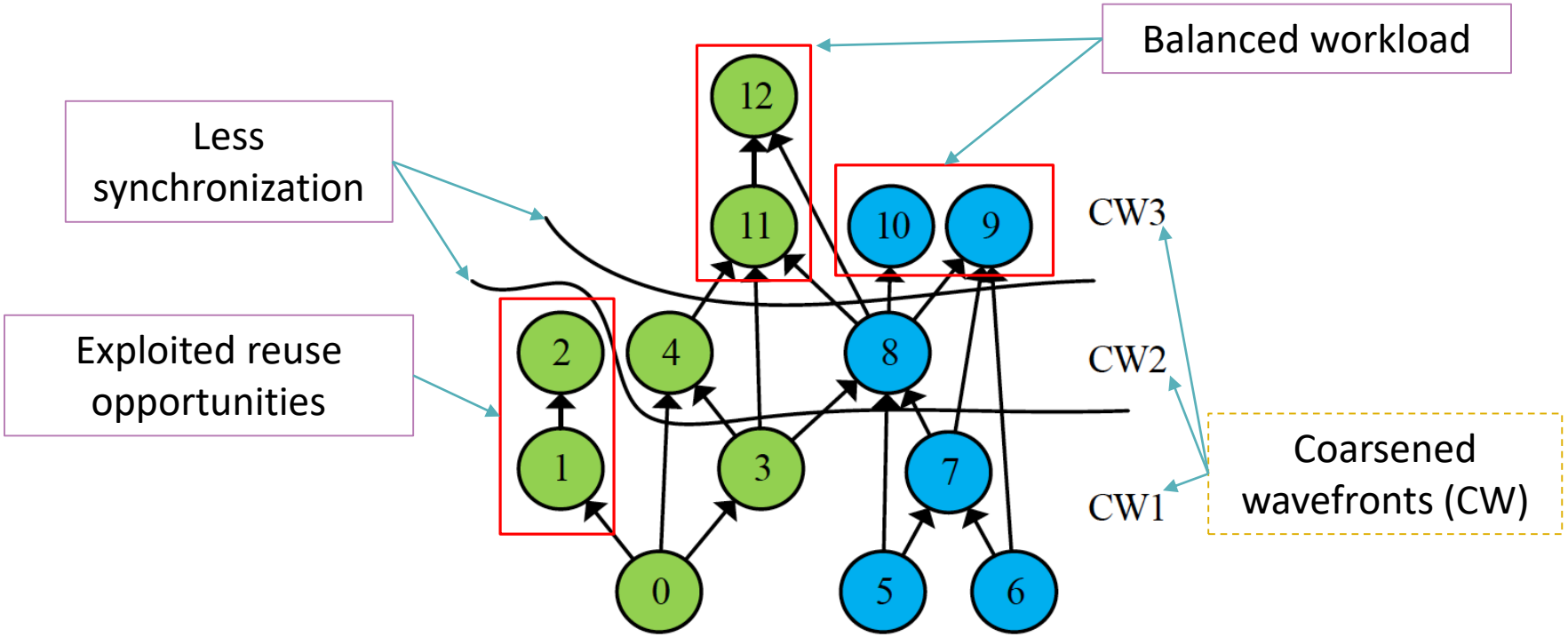
Extra edges reduces parallelism

It exploits reuse opportunity

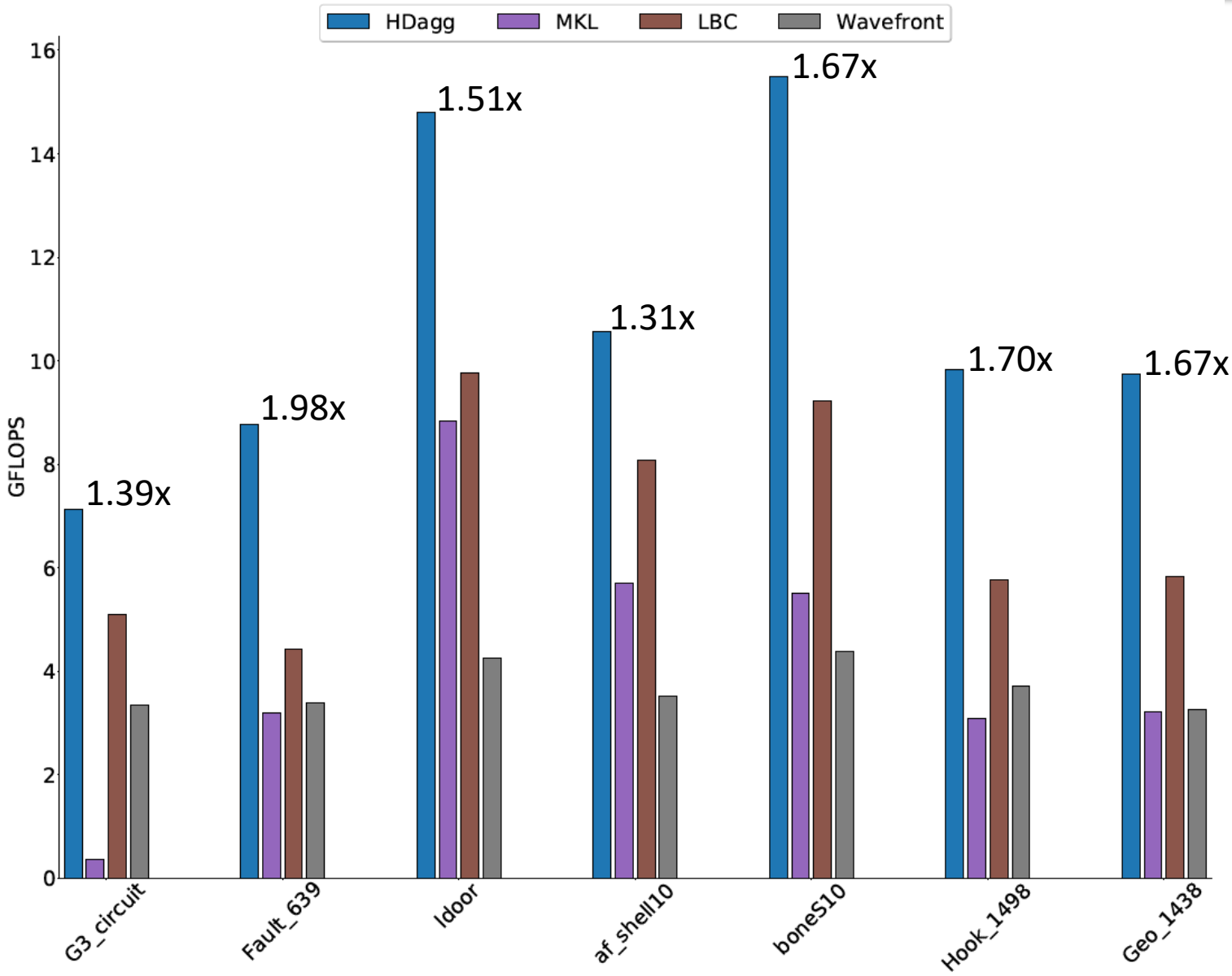


HDAGG

- HDagg provides a load balanced schedule with improved locality and synchronization overhead.



Performance Comparison for Triangular Solve

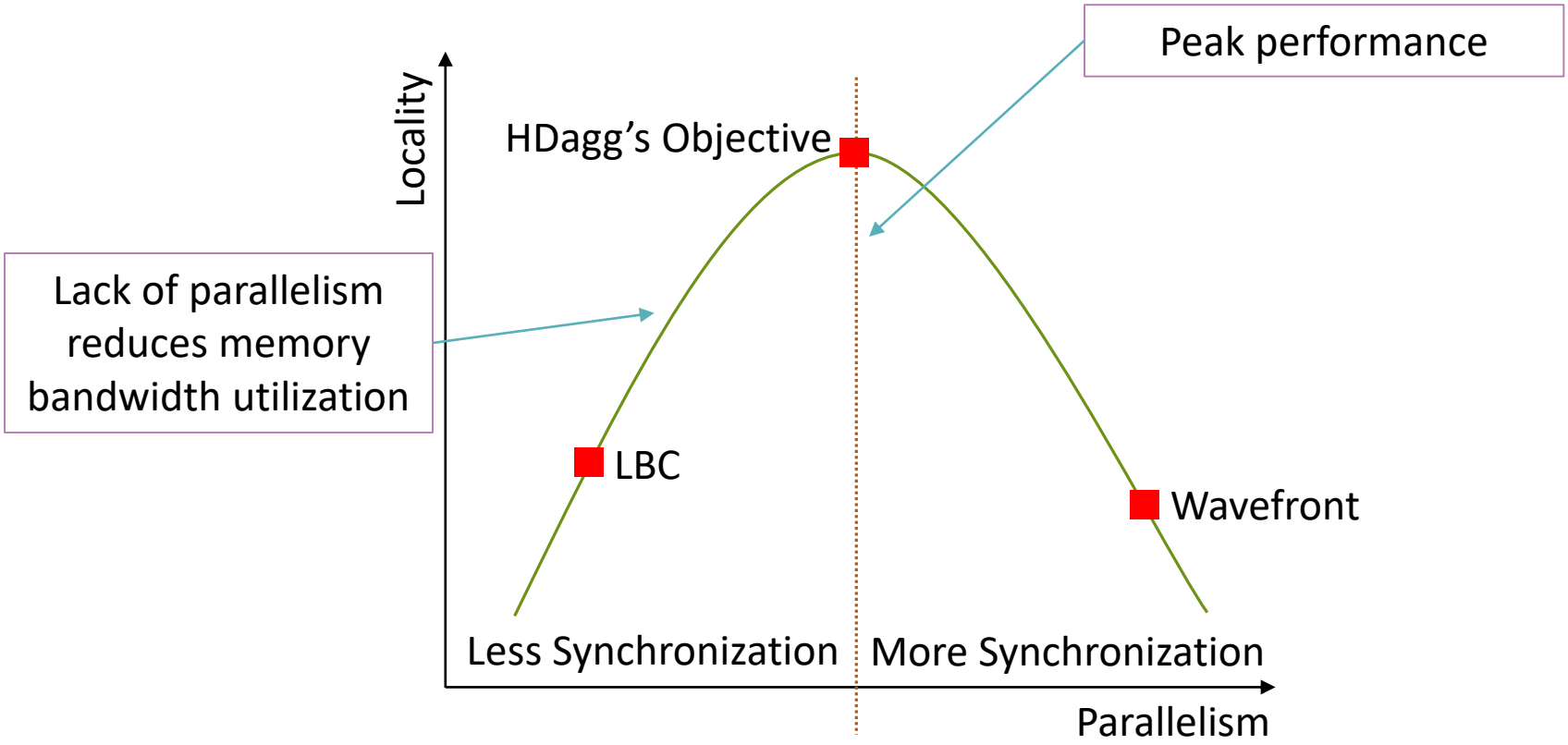


Outline

- Overview
- HDagg: Hybrid aggregation of loop-carried dependence
  - HDagg's Objective
  - HDagg's Algorithm
    - Coarsening of densely connected vertices
    - Wavefront coarsening
- Results
- Conclusion

HDagg Objective

- HDagg finds a trade-off between locality, load balance, and synchronization.

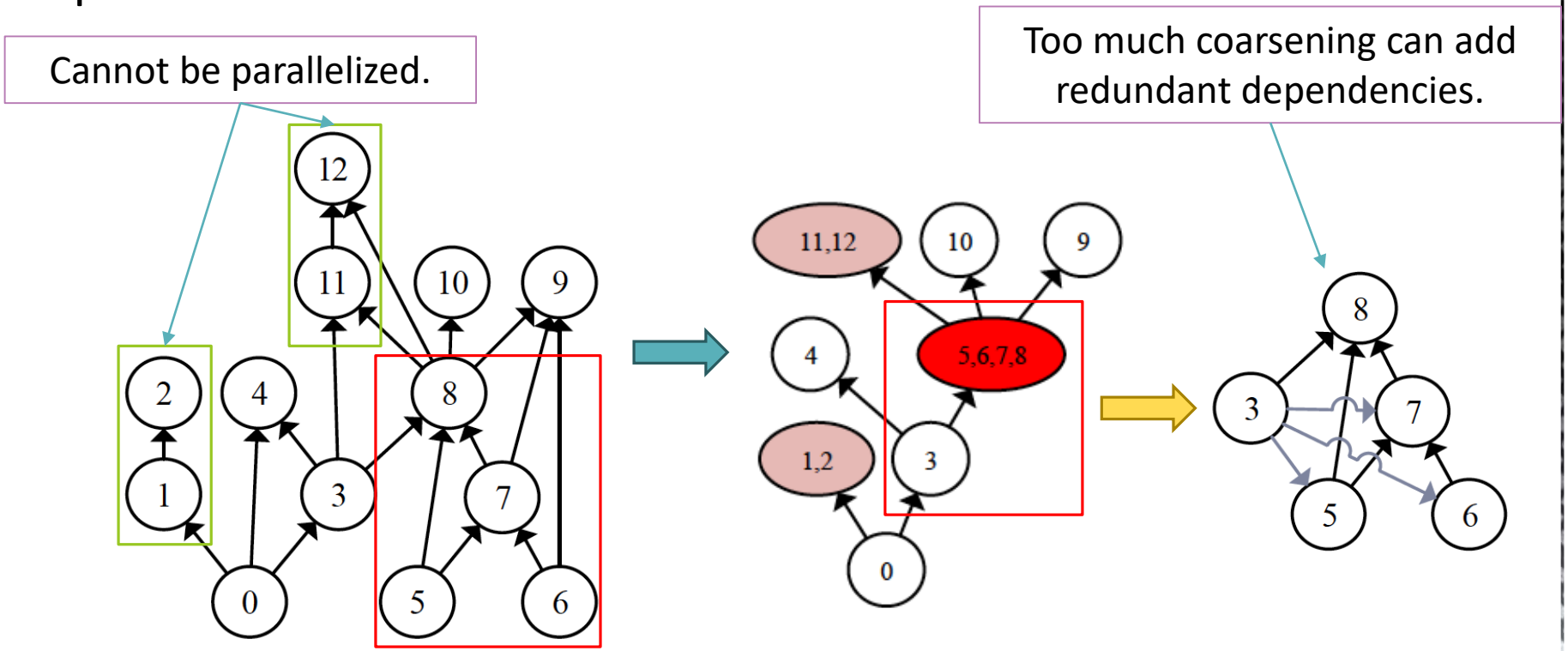


Outline

- Overview
- HDagg: Hybrid aggregation of loop-carried dependence
  - HDagg's Objective
  - HDagg's Algorithm
    - Coarsening of densely connected vertices
    - Wavefront coarsening
- Results
- Conclusion

Local Coarsening

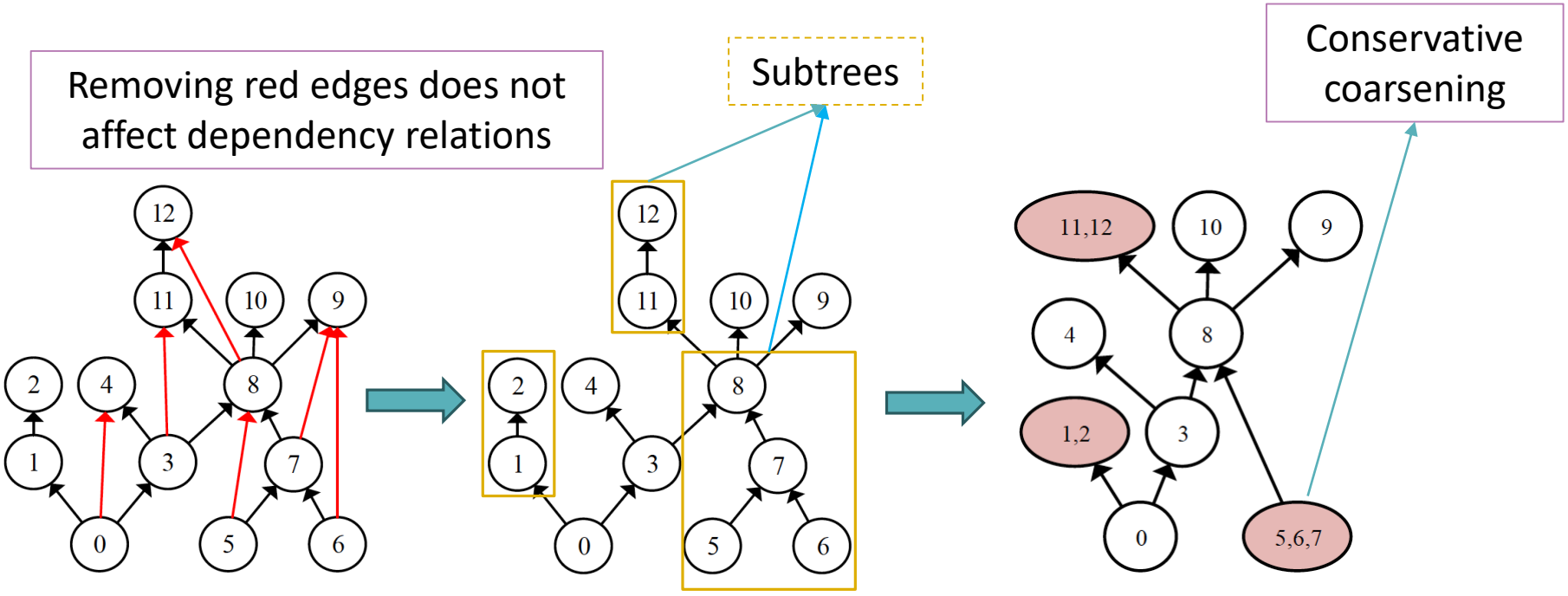
- Densely connected vertices have lots of shared data.
- Coarsening them improves locality. However, it can reduce parallelism.



Coarsening Densely Connected Vertices

- HDagg deletes transitive edges to reveal densely connected components as subtrees.

- Subtrees with small effect in parallelism are eligible to be coarsened.



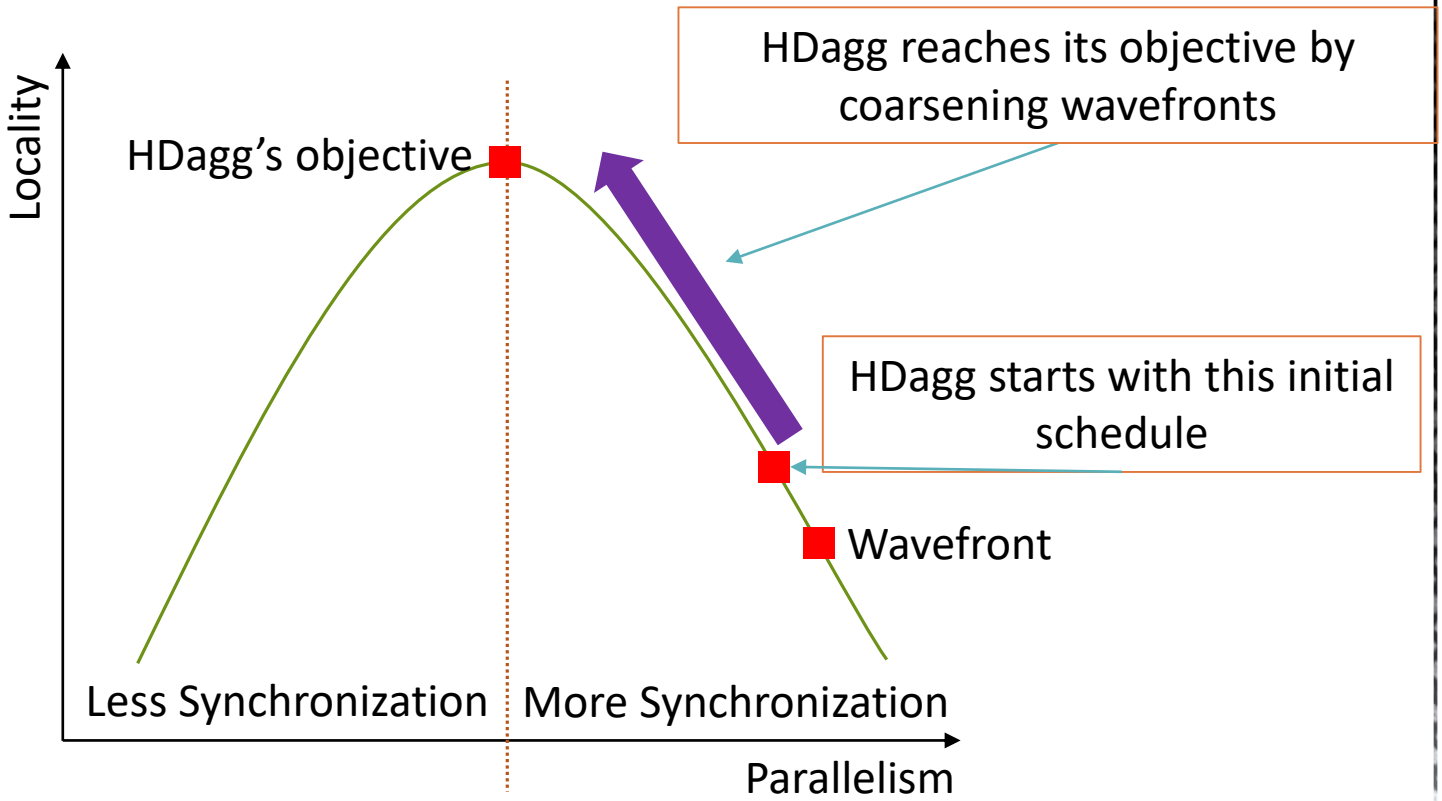
OUTLINE

- Overview
- HDagg: Hybrid aggregation of loop-carried dependence
  - HDagg's Objective
  - HDagg's Algorithm
    - Coarsening of densely connected vertices
    - Wavefront coarsening
- Results
- Conclusion

HDagg's Wavefront Coarsening Overview

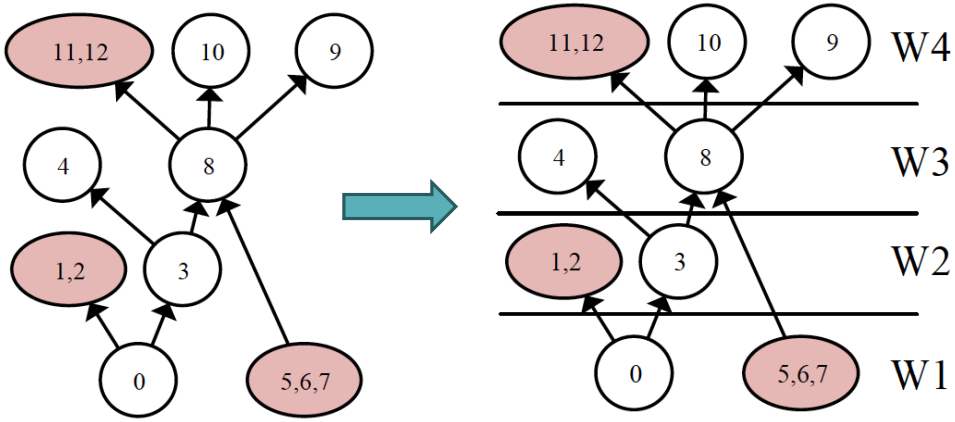
- HDagg uses Wavefront techniques as an initial schedule.

- HDagg improves locality by coarsening wavefronts while maintaining load balance.

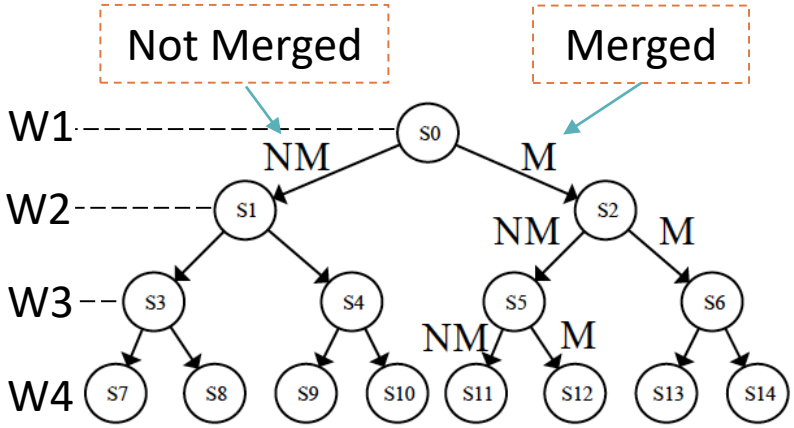


Decision Space for Wavefront Coarsening

- Coarsening wavefronts in a DAG can reduce parallelism.
- HDagg controls the parallelism reduction with its load-balance preserving (LBP) algorithm.
- LBP forms a binary tree to choose one of possible merging scenarios.



HDagg's initial schedule.



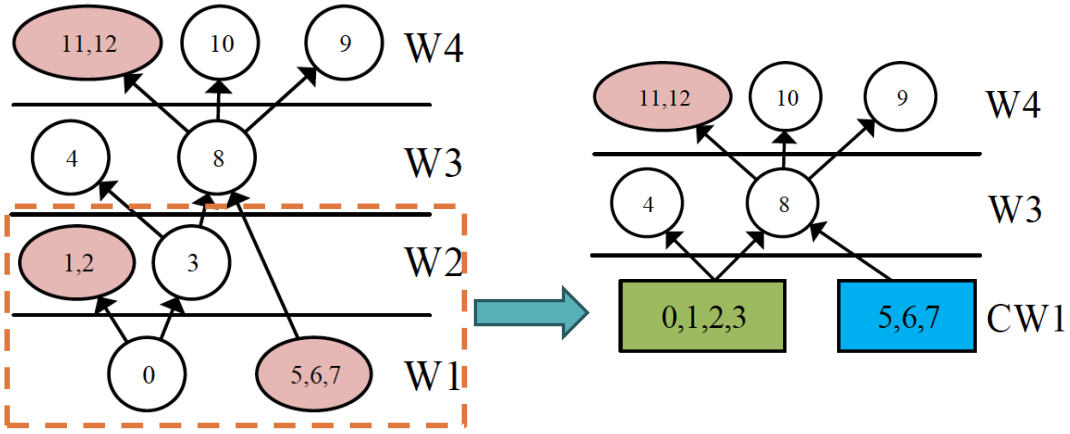
Decision space

S1: Not merging the first two wavefronts ([[W1], [W2]]).
S2: Merging the first two wavefronts ([W1, W2]).

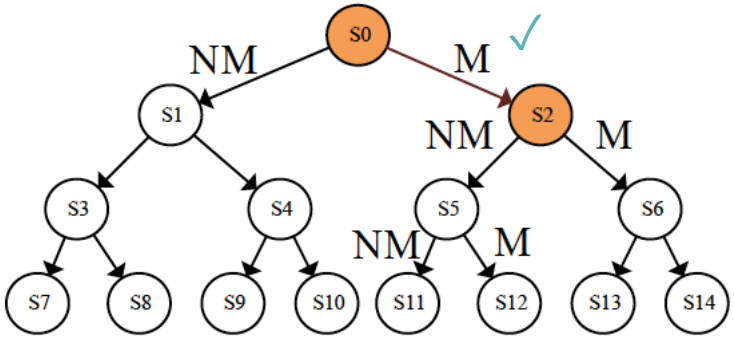
HDAGG'S WAVEFRONT COARSENING

- Moving from S0 to one of S1 or S2, indicates whether W1 should be merged with W2 or not.

- It always starts with merged branch in the tree.



Merging [W1, W2]

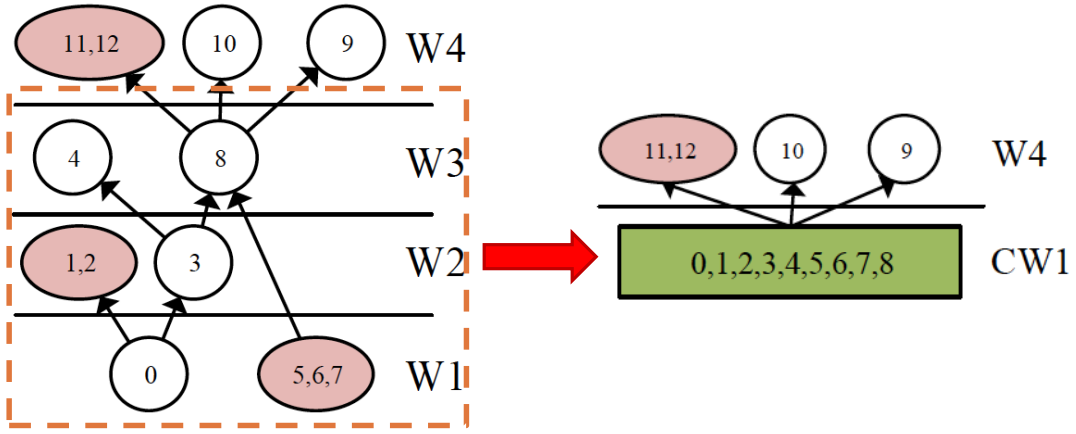


Decision space

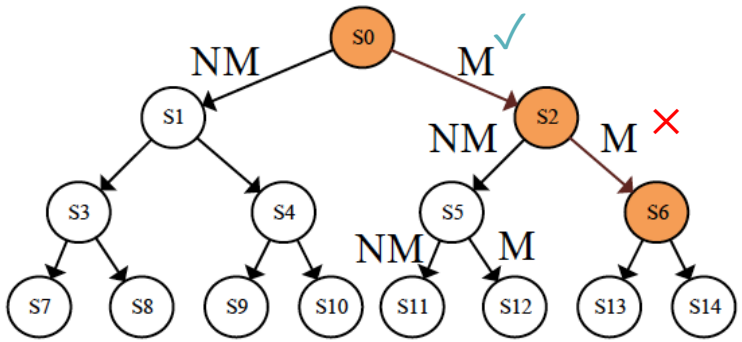
CW is a coarsened wavefront with its tasks assigned to the cores. Green and blue are the tasks assigned to core 0 and core 1.

HDAGG'S WAVEFRONT COARSENING

- If merging causes load imbalance, LBP takes not-merged branch.



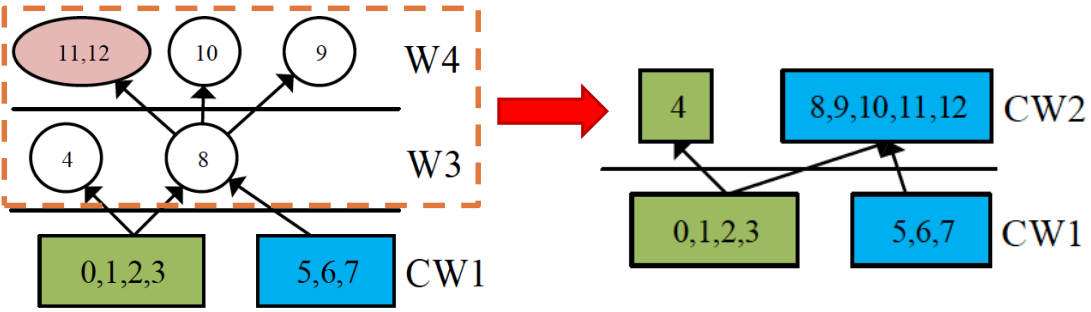
Merging [W1, W2, W3]



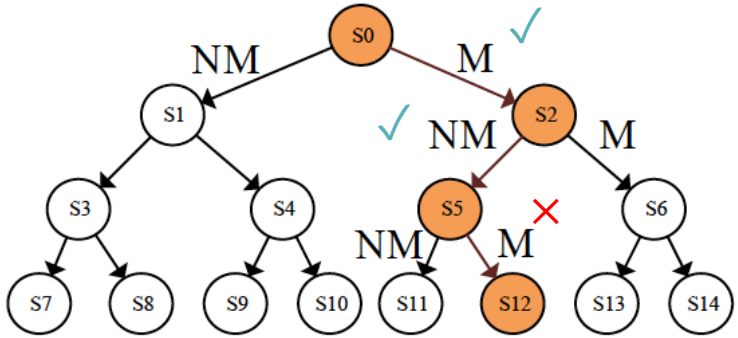
Decision space

HDAGG'S WAVEFRONT COARSENING

- It continues with the last acceptable coarsening scenario.



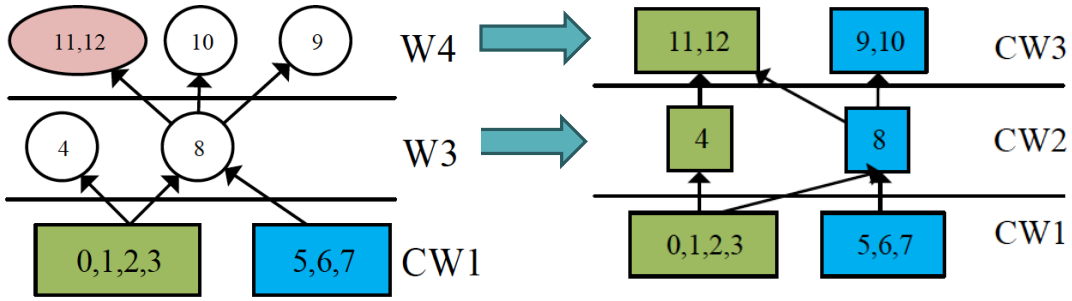
Merging [W3, W4]



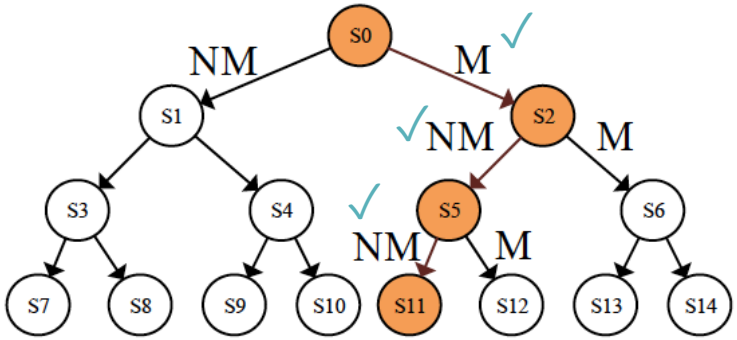
Decision space

HDAGG'S WAVEFRONT COARSENING

- LBP also assigns tasks inside an unmerged wavefronts to the cores.



Assigning tasks to two cores



Decision space

Outline

- Overview
- HDagg: Hybrid aggregation of loop-carried dependence
  - HDagg's Objective
  - HDagg's Algorithm
    - Coarsening of densely connected vertices
    - Wavefront coarsening
- Results
- Conclusion

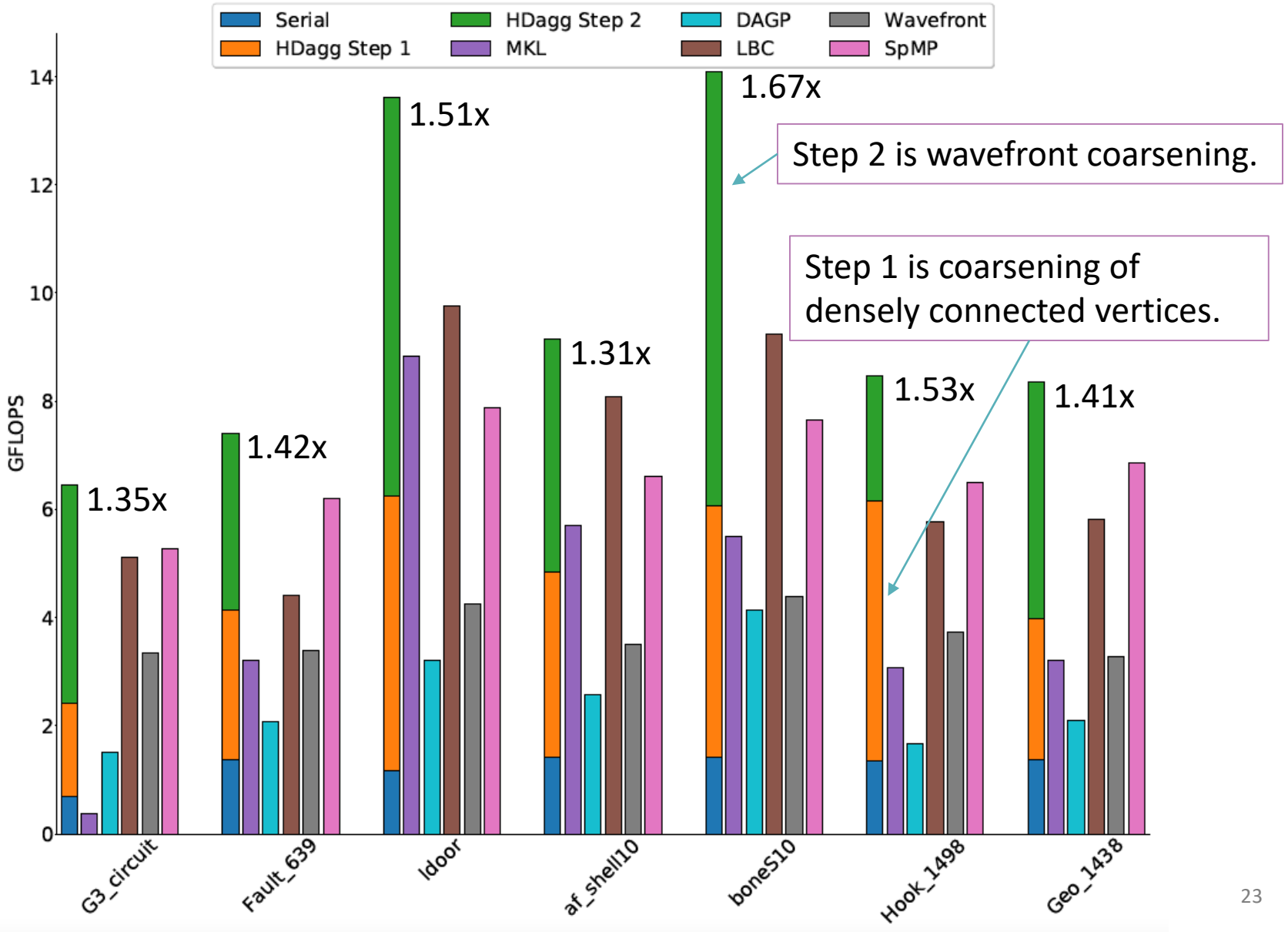
Experimental Setup

Target processor: Intel® Xeon® Gold 6248 CPU (Cascade);

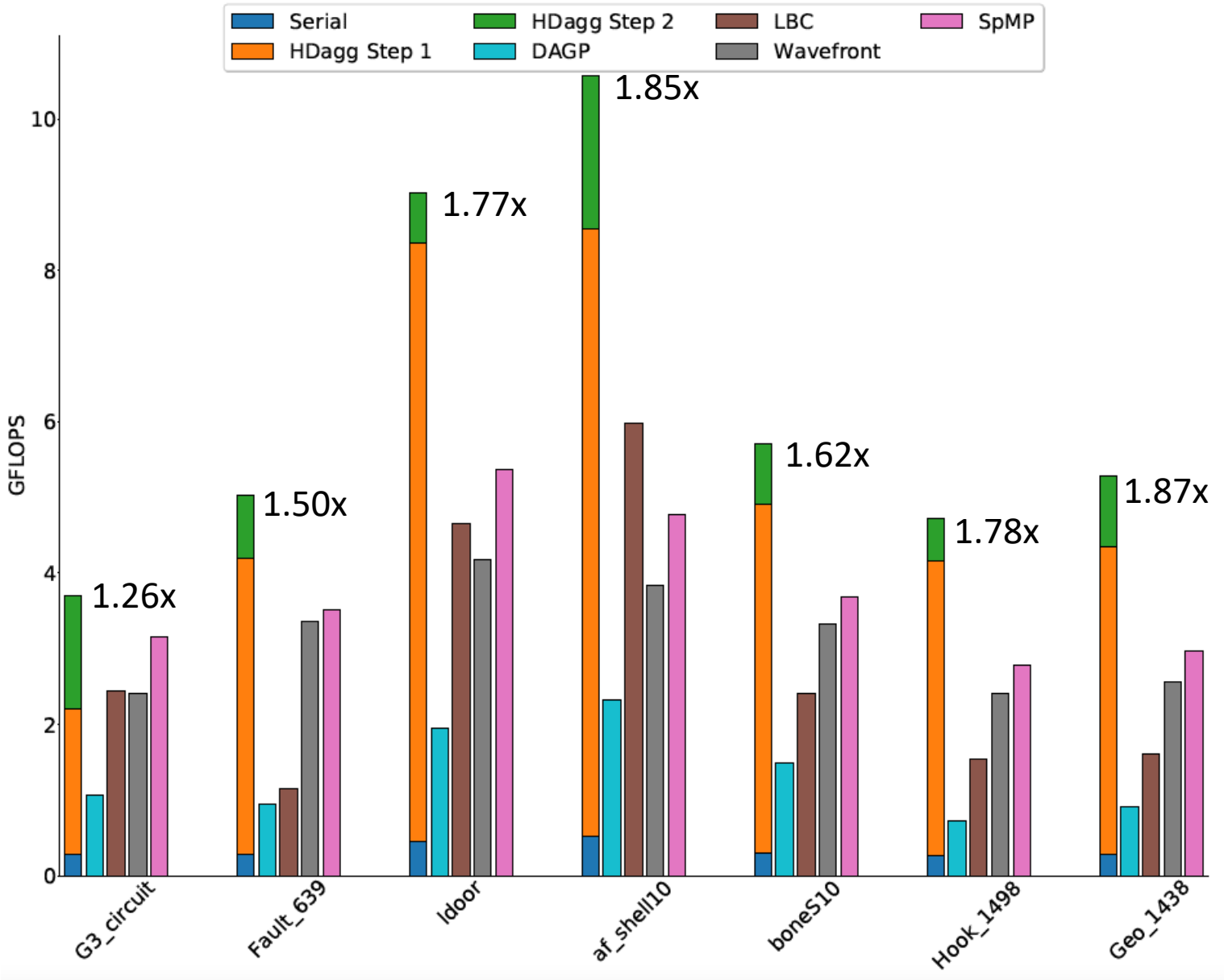
Benchmarks: Suitesparse matrix collection

Name	Application	Order (10^3)	Non-zeros (10^6)
G3_circuit	Circuit simulation	1585	7.6
af_shell10	sheet metal forming	1508	52.3
Hook_1498	3D mechanical problem	1498	59.3
Geo_1438	geomechanically model of earth crust	1413	60.2
ldoor	INDEED Test Matrix	952	42.4
boneS10	Model reduction problem	914.8	40.8
Fault_639	Contact mechanics	638.8	27.2

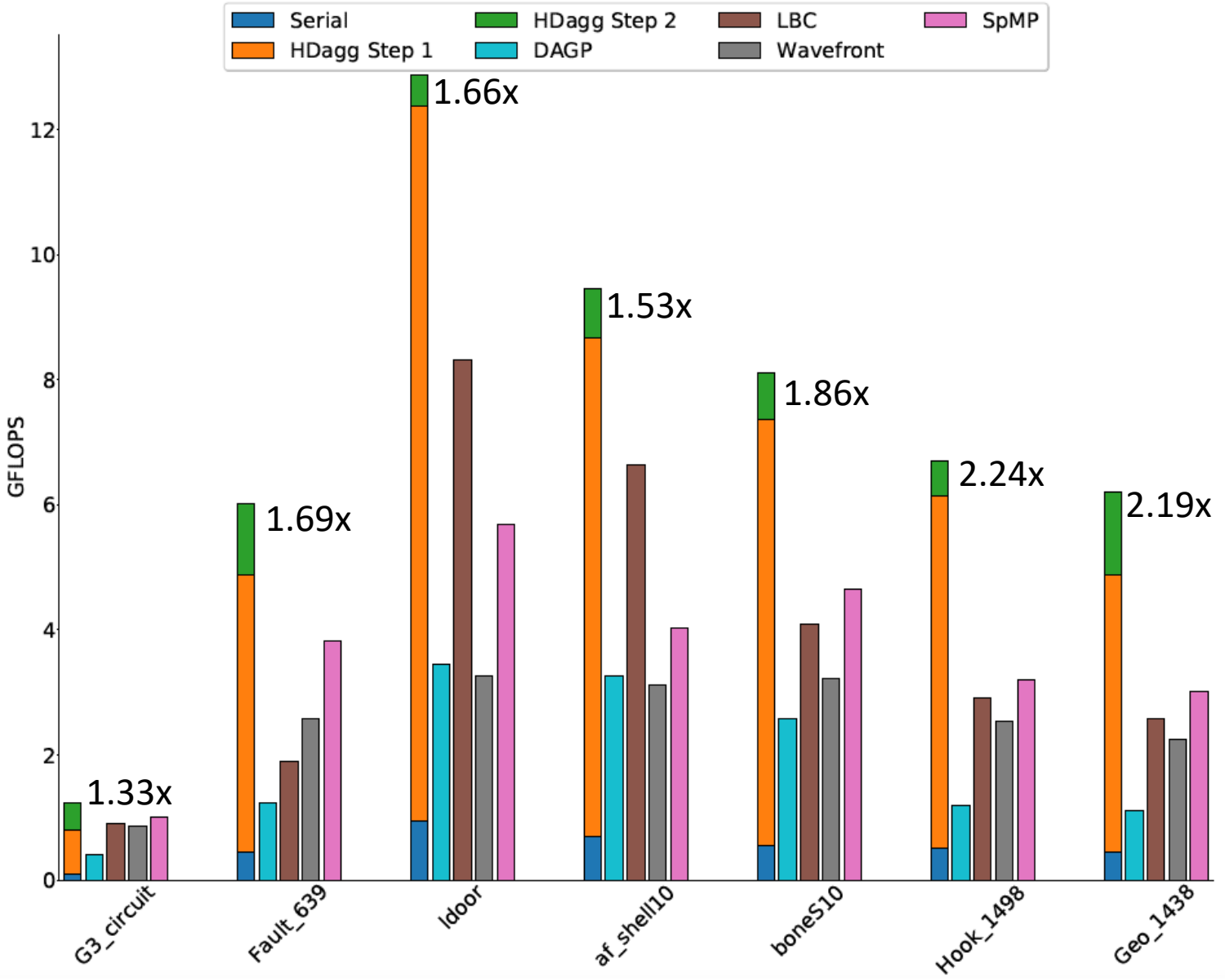
HDagg vs Libraries: Triangular Solve



HDagg vs Libraries: Incomplete Cholesky



HDagg vs Libraries: Incomplete LU



Outline

- Overview
- HDagg: Hybrid aggregation of loop-carried dependence
  - HDagg's Objective
  - HDagg's Algorithm
    - Coarsening of densely connected vertices
    - Wavefront coarsening
- Results
- Conclusion

Conclusion

- HDagg is a novel iteration aggregation technique that creates an efficient schedule for sparse numerical methods.
- HDagg uses two coarsening algorithms to generate load-balanced schedule while increasing locality and reducing synchronization.
- HDagg's schedule outperforms state-of-the-arts such as Wavefront, SpMP, DAGP, and LBC algorithms.
- HDagg's source code is publicly available from:

Benchmark: https://github.com/BehroozZare/HDagg-benchmark
Tool: https://github.com/sympiler/aggregation