Accelerating BFS Algorithm on a RISCV-based Many-Core Cluster

M. R. Ashuthosh[1], Aakarsh Vinay[1], Krishna K. Nagar[2], Madhura Purnaprajna[1]

[1]Centre for Heterogenous and Intelligent Processing System, PES University, Bengaluru, India
[2]Intel Corporation, USA

Motivation

+ High performance
+ Customisable, efficient
- High design effort

+Medium design effort
+ Customisable

+ Easy programming
- Fixed hardware

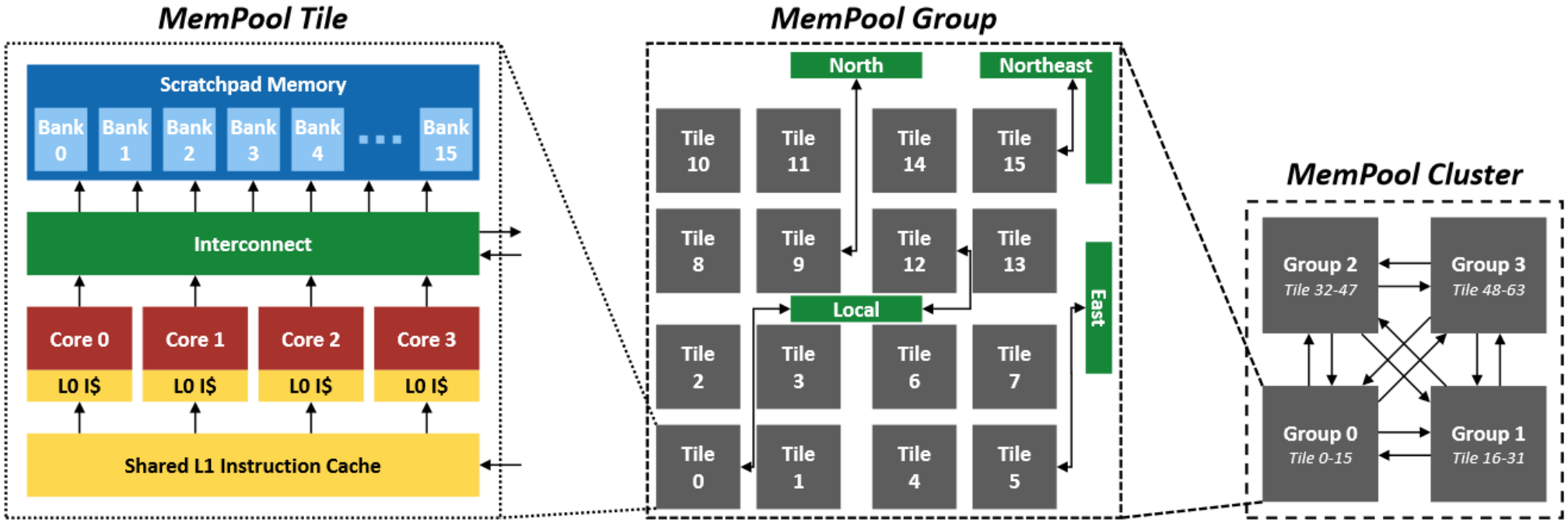
Acceleration of graph algorithms

- Graph algorithms are everywhere

- Memory bound, irregular memory accesses
- BFS traversal is the basic component in graph algorithms

Image source: Internet

Mempool [1]: A RISCV-based Manycore Cluster



Mempool Tile:

- 4 cores
- 16 memory banks
- Single-cycle latency

Mempool Group:

- 64 cores
- 256 memory banks
- 3 cycles latency

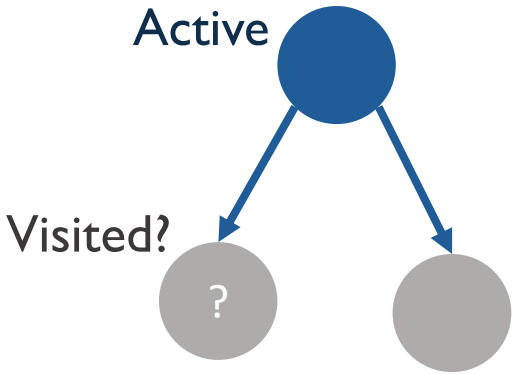
Mempool Cluster:

- 256 cores
- 1024 memory banks (1MiB)
- 5 cycles latency

[1] S. Riedel, M. Cavalcante, R. Andri, and L. Benini, "Mempool: A scalable manycore architecture with a low-latency shared L1 memory," 2023.

BFS approaches

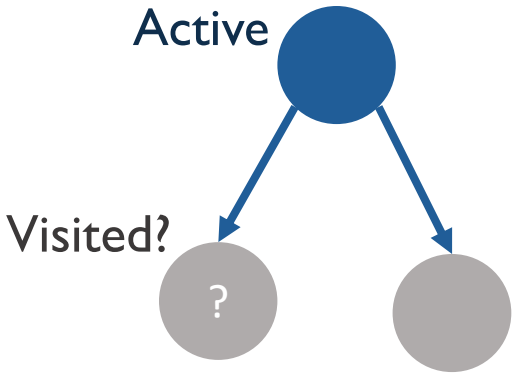
1. Top-down BFS



- Check neighbors of active vertex, if unvisited, update the unvisited vertex.

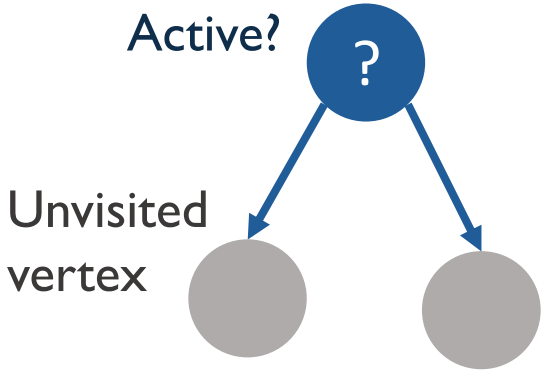
BFS approaches

1. Top-down BFS



- Check neighbors of active vertex, if unvisited, update the unvisited vertex.

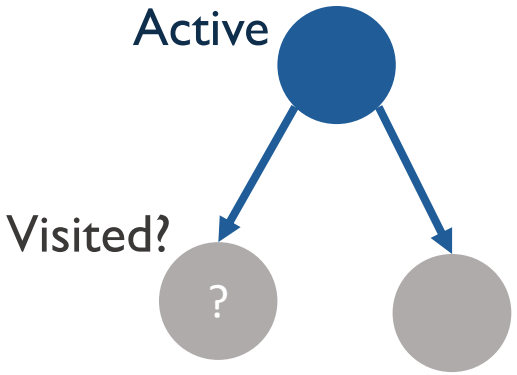
2. Bottom-up BFS



- Check neighbors of unvisited vertex, if active, update the unvisited vertex.

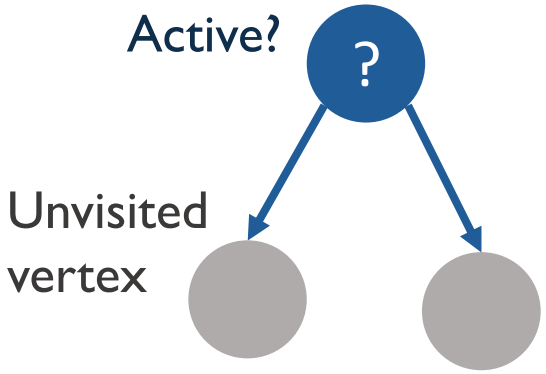
BFS approaches

1. Top-down BFS



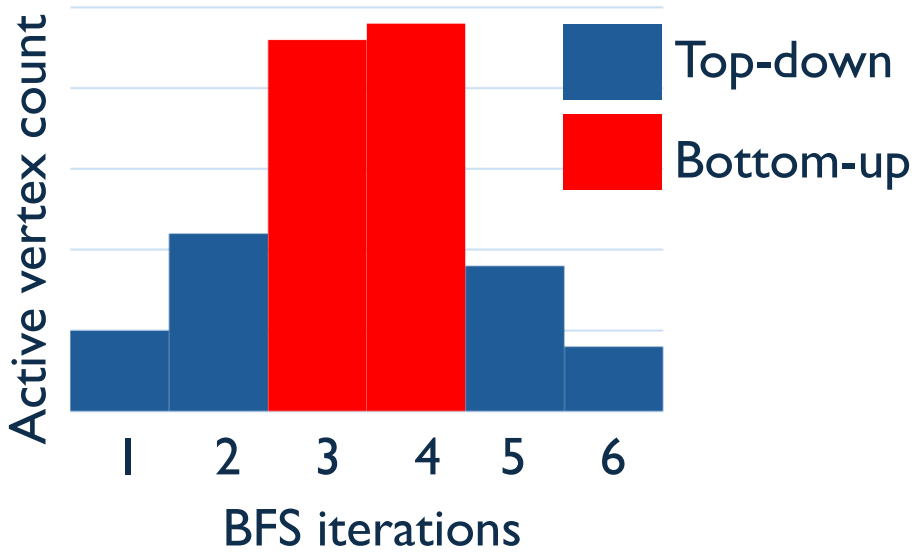
- Check neighbors of active vertex, if unvisited, update the unvisited vertex.

2. Bottom-up BFS



- Check neighbors of unvisited vertex, if active, update the unvisited vertex.

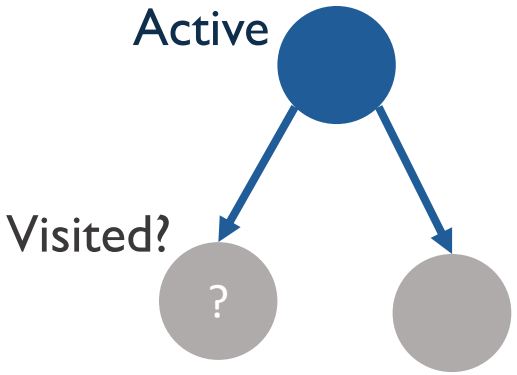
3. Hybrid BFS [3], an optimal approach



[3] S. Beamer, K. Asanovi´c, and D. Patterson, "Direction-optimizing breadth-first search," SC '12, (Washington, DC, USA), IEEE Computer Society Press, 2012.

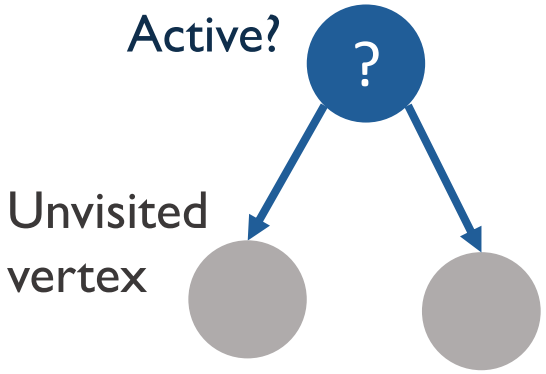
BFS approaches

1. Top-down BFS



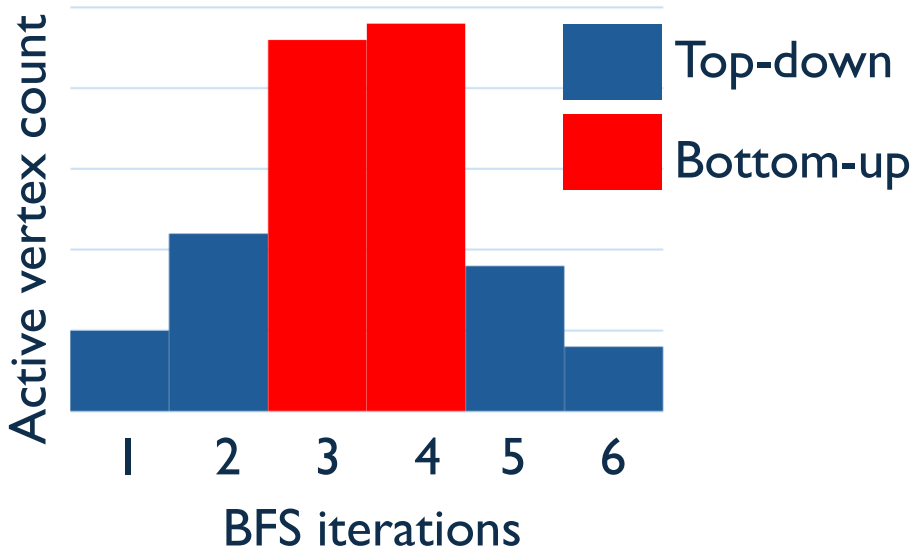
- Check neighbors of active vertex, if unvisited, update the unvisited vertex.

2. Bottom-up BFS



- Check neighbors of unvisited vertex, if active, update the unvisited vertex.

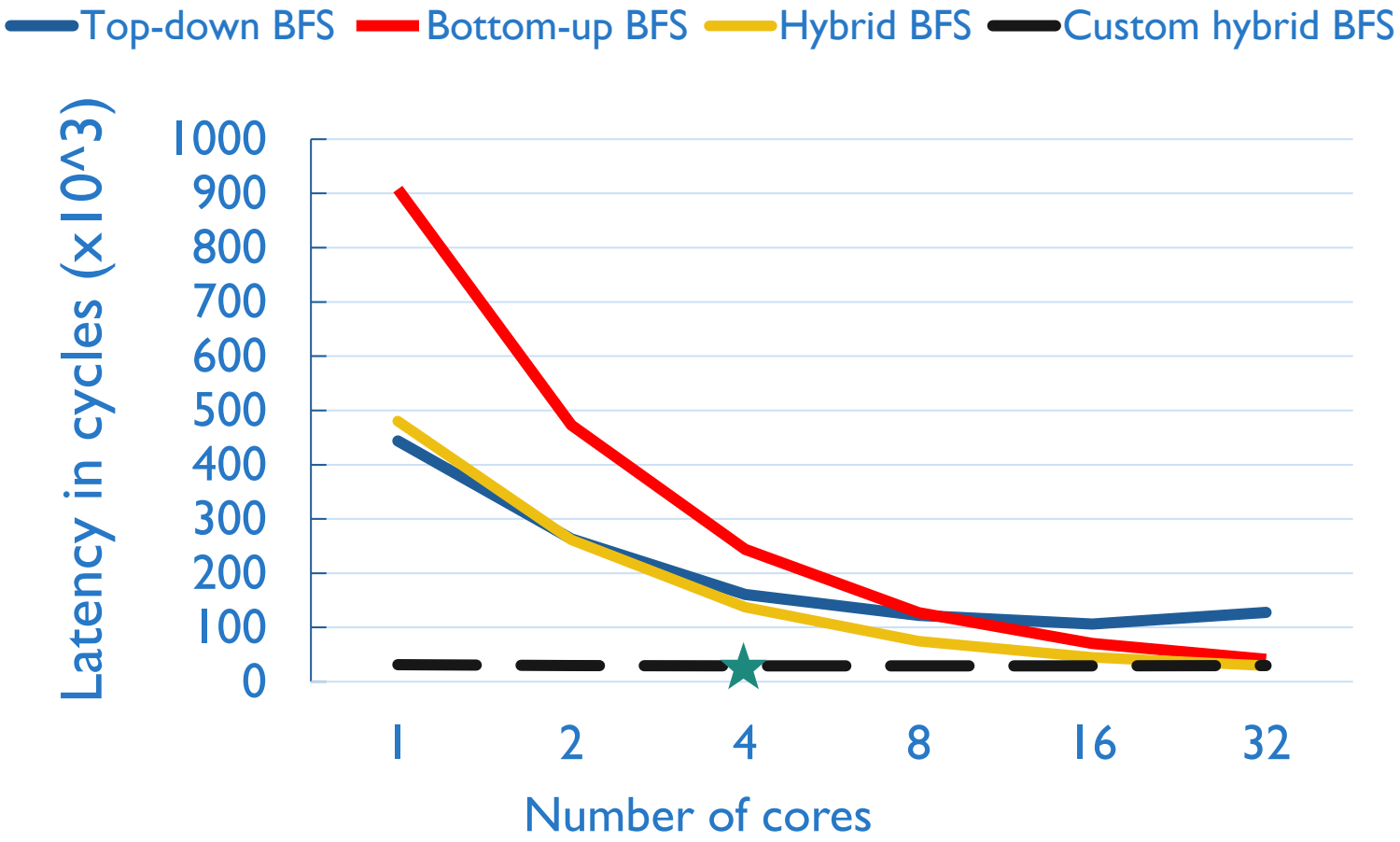
3. Hybrid BFS [3], an optimal approach



4. Custom hybrid BFS (our approach): Hybrid with optimal number of cores for iterations of top-down and bottom-up.

[3] S. Beamer, K. Asanovi´c, and D. Patterson, "Direction-optimizing breadth-first search," SC '12, (Washington, DC, USA), IEEE Computer Society Press, 2012.

How many cores are enough?

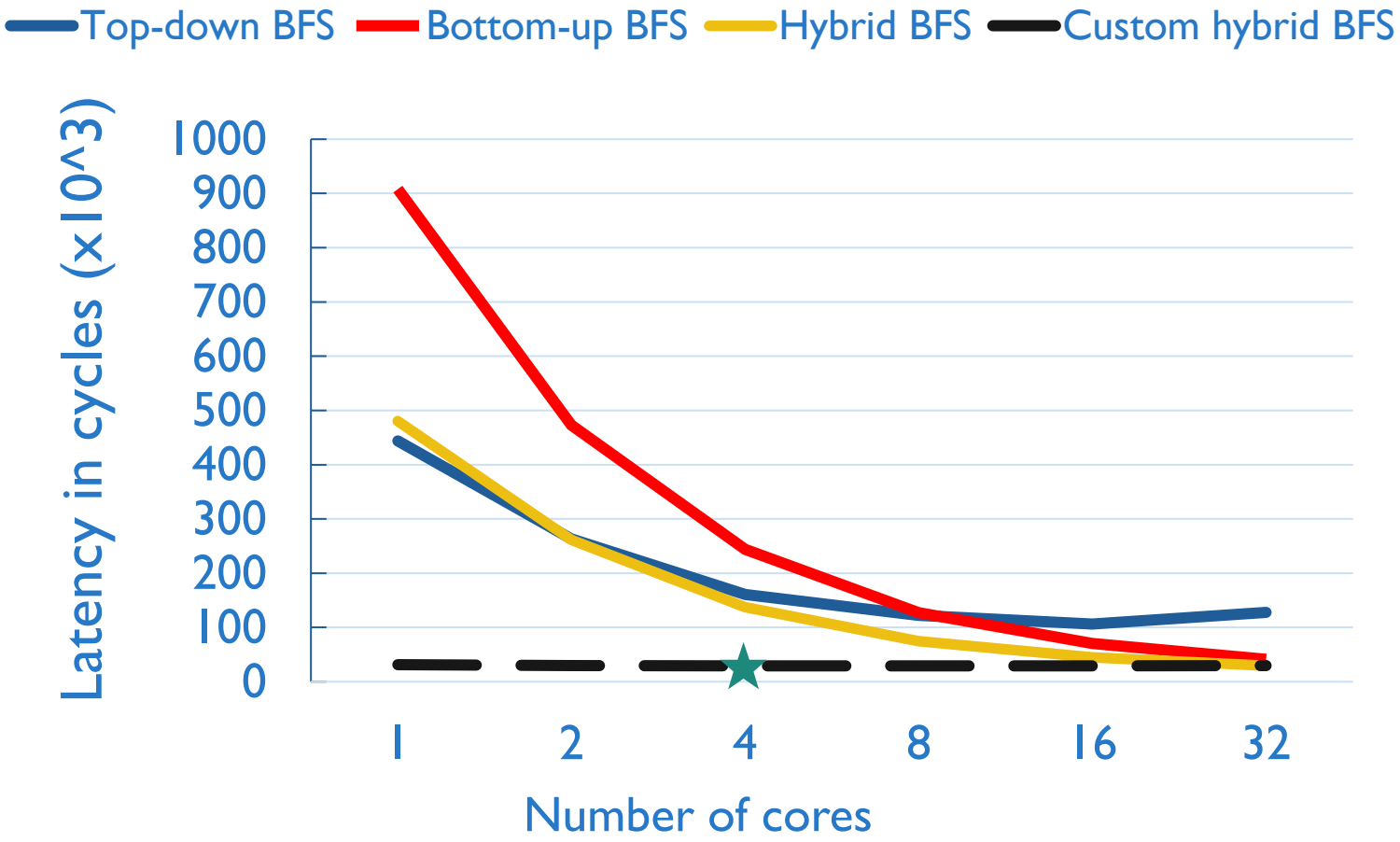


On traversing the tech-routers-rf [2], the speedups:

- Top-down BFS: 4.2× on 16 cores.
- Bottom-up BFS: 22.1× on 32 cores.
- Hybrid BFS: 16.02× on 32 cores.
- ★ Custom hybrid BFS: 16.5× when top-down=4 cores, bottom-up=32 cores.

[2] R.A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in AAAI.

How many cores are enough?



On traversing the tech-routers-rf [2], the speedups:

- Top-down BFS: 4.2× on 16 cores.
- Bottom-up BFS: 22.1× on 32 cores.
- Hybrid BFS: 16.02× on 32 cores.
- ★ Custom hybrid BFS: 16.5× when top-down=4 cores, bottom-up=32 cores.

- Varying the number of cores for top-down and fixing 32 cores for bottom-up provides no more than 7.4% performance improvement.

[2] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in AAAI.

Conclusion & Ongoing efforts

- Custom hybrid BFS(our approach) performs better than Hybrid BFS with maximum number of cores.

- These results underscore the importance of performance modeling for many-core architectures in rapid design-space exploration.

- Similar analysis for popular graph algorithms such as Pagerank.

Scan to reach us!

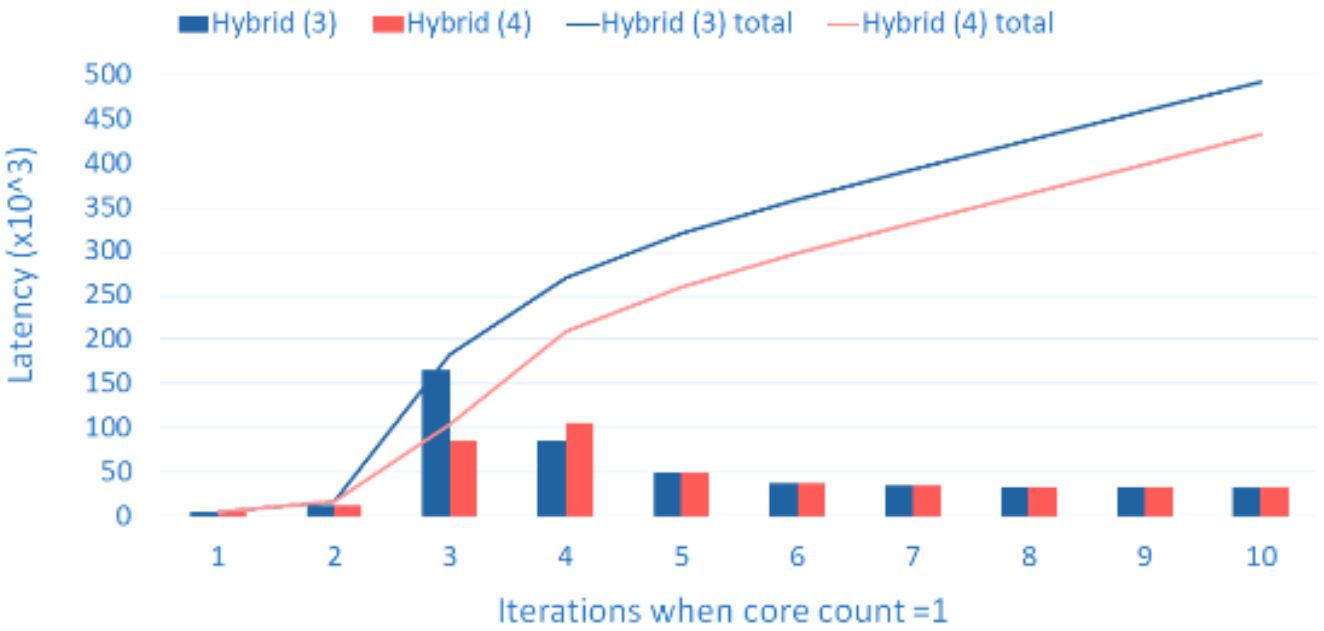
Thank you

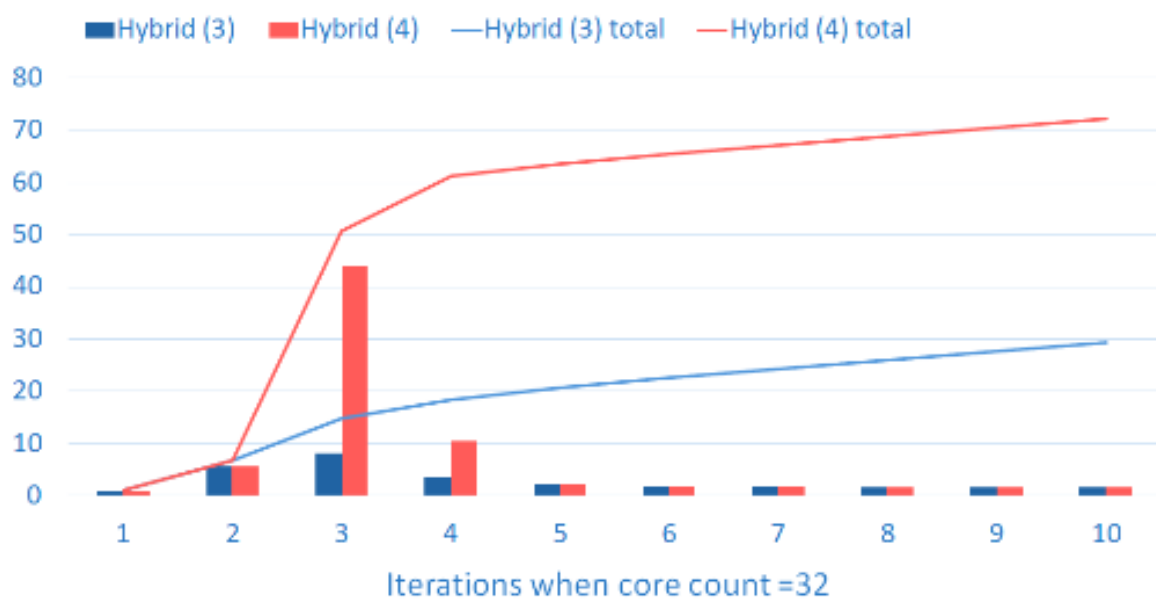
Contact:

Mail: ashuthoshmr@pes.edu
Website: https://ashuthosh.de/

Backup

Why hybrid is slower than top-down when #core=1





Why?	Edges explored	Edges explored
Core count	Switch at 3 (bu at 3)	Switch at 4 (td at 3)
1	6716	1964
2	3423	1010
4	1731	612
8	916	298
16	540	438
32	293	721