On Graphs, GPUs, and Blind Dating

A Workload to Processor Matchmaking Quest

Abdullah Gharaibeh, Lauro Beltro Costa, Elizeu Santos-Neto, Matei Ripeanu

Department of Electrical and Computer Engineering
The University of British Columbia

presented by Ulla Aeschbacher

Institute of Computer Science, ETH Zürich

17.4.18

Goal of Paper

- Explore effectiveness of graph partitioning strategies and workload allocation schemes to maximize performance
- Give evaluation of said performance

Overview

1. Goal of Paper
2. Related Work and Background
3. Characteristics
4. Partitioning Strategy
5. Evaluation
6. Summary

Related Work

- September 2012: A Yoke of Oxen and a Thousand Chickens for Heavy Lifting Graph Processing

Related Work

- September 2012: A Yoke of Oxen and a Thousand Chickens for Heavy Lifting Graph Processing
- May 2013: On Graphs, GPUs, and Blind Dating

Related Work

- September 2012: A Yoke of Oxen and a Thousand Chickens for Heavy Lifting Graph Processing
- May 2013: On Graphs, GPUs, and Blind Dating
- November 2013: The Energy Case for Graph Processing on Hybrid CPU and GPU Systems

Totem Computation Model

- Processing divided into rounds
- Rounds consisting of three phases
  - Computation
  - Communication
  - Synchronization

Characteristics of (Most) Real-World Graph Workloads

- Modest processing per vertex per round

Characteristics of (Most) Real-World Graph Workloads

- Modest processing per vertex per round
- Poor locality

Characteristics of (Most) Real-World Graph Workloads

- Modest processing per vertex per round
- Poor locality
- Power-law degree distribution

Characteristics of Hybrid System

CPU Intel Nehalem Xeon X5650

- 2 $\times$ core frequency
- 24.6 $\times$ more memory
- 6 $\times$ more cache
- retailed at 999$

$\rightarrow$ Can process large graphs

GPU Nvidia Tesla C2075

- 16 $\times$ more hardware threads
- 4.5 $\times$ faster memory access
- retailed at 599$

$\rightarrow$ Can process graphs in parallel

Required Features for Graph Partitioning Strategies

- Low space and time complexity

Required Features for Graph Partitioning Strategies

- Low space and time complexity
- Handles scale-free graphs

Required Features for Graph Partitioning Strategies

- Low space and time complexity
- Handles scale-free graphs
- Handles large graphs

Required Features for Graph Partitioning Strategies

- Low space and time complexity
- Handles scale-free graphs
- Handles large graphs
- Reduces computation not communication

Partitioning Strategy

Placing high-degree vertices in one type of processor
and low-degree ones in the other type.

- Partitions have different degrees of parallelism
- Partitions are more homogenous in terms of vertex connectivity
- Low cost in terms of computational and space complexity

Three Different Partitioning Strategies

- HIGH: highest-degree vertices to CPU, lowest-degree vertices to GPU
- LOW: lowest-degree vertices to CPU, highest-degree vertices to GPU
- RAND: partitions the graph randomly

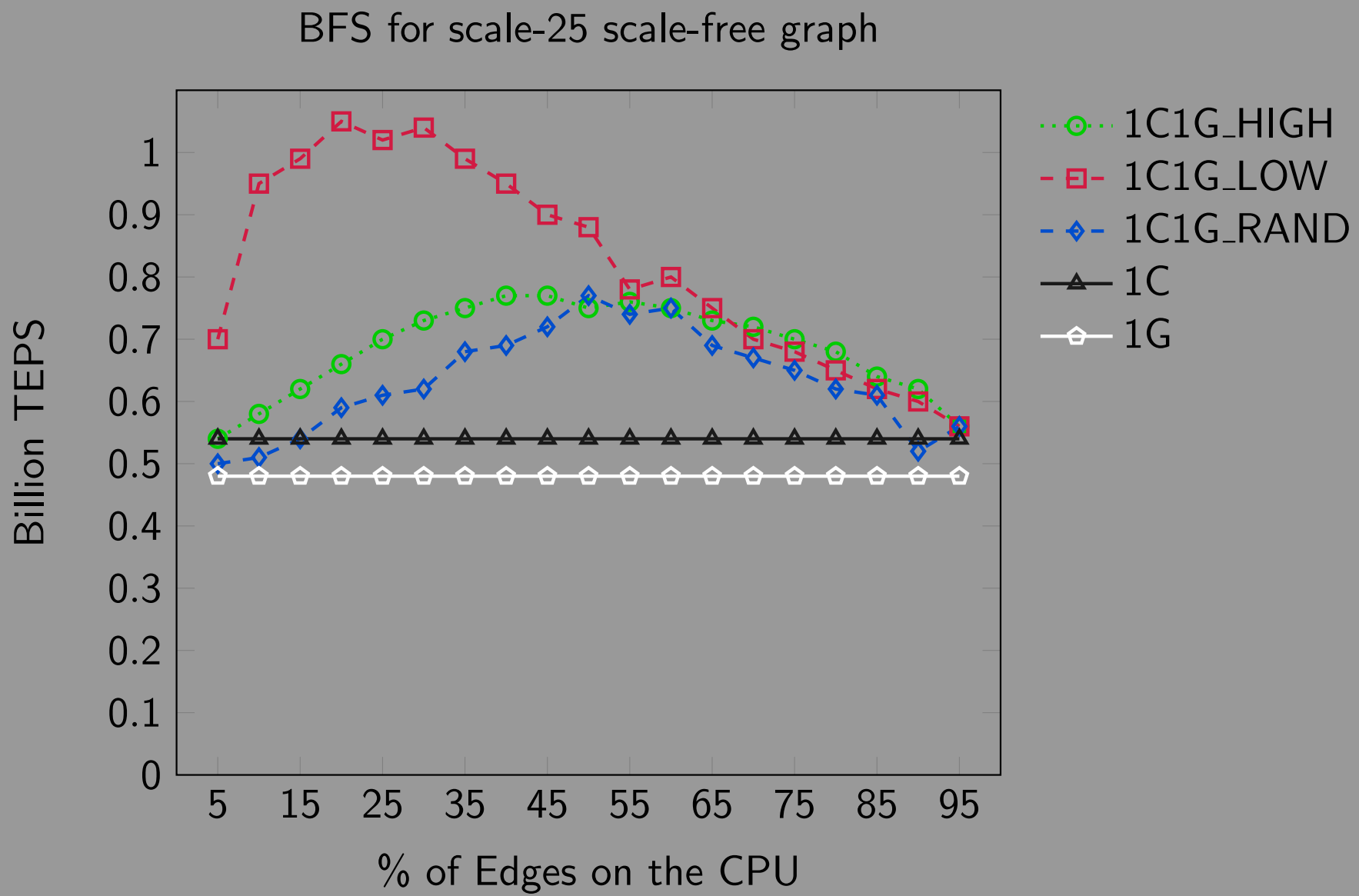
Evaluation

- Breadth-First Search (BFS): based on the level-synchronous one by Hong et al.
- PageRank: algorithm used by Google Search to rank websites in their search engine results by assigning weights to hyperlinks

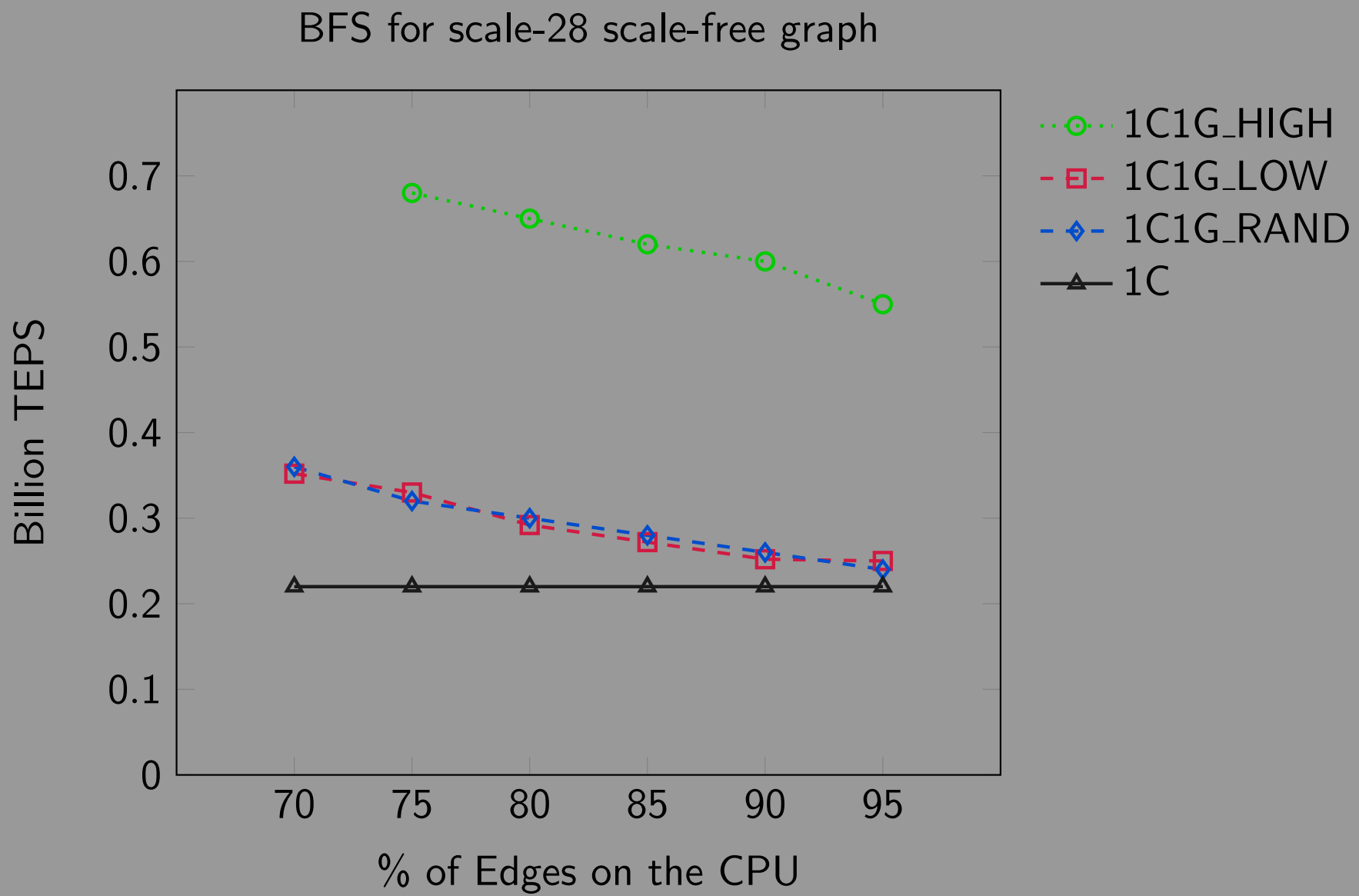
Evaluation

- Scale-25 and scale-28 power-law degree distribution graphs
- Scale-25 and scale-28 uniform degree distribution graphs
- Scale-25 to scale-29 graphs different hardware configurations

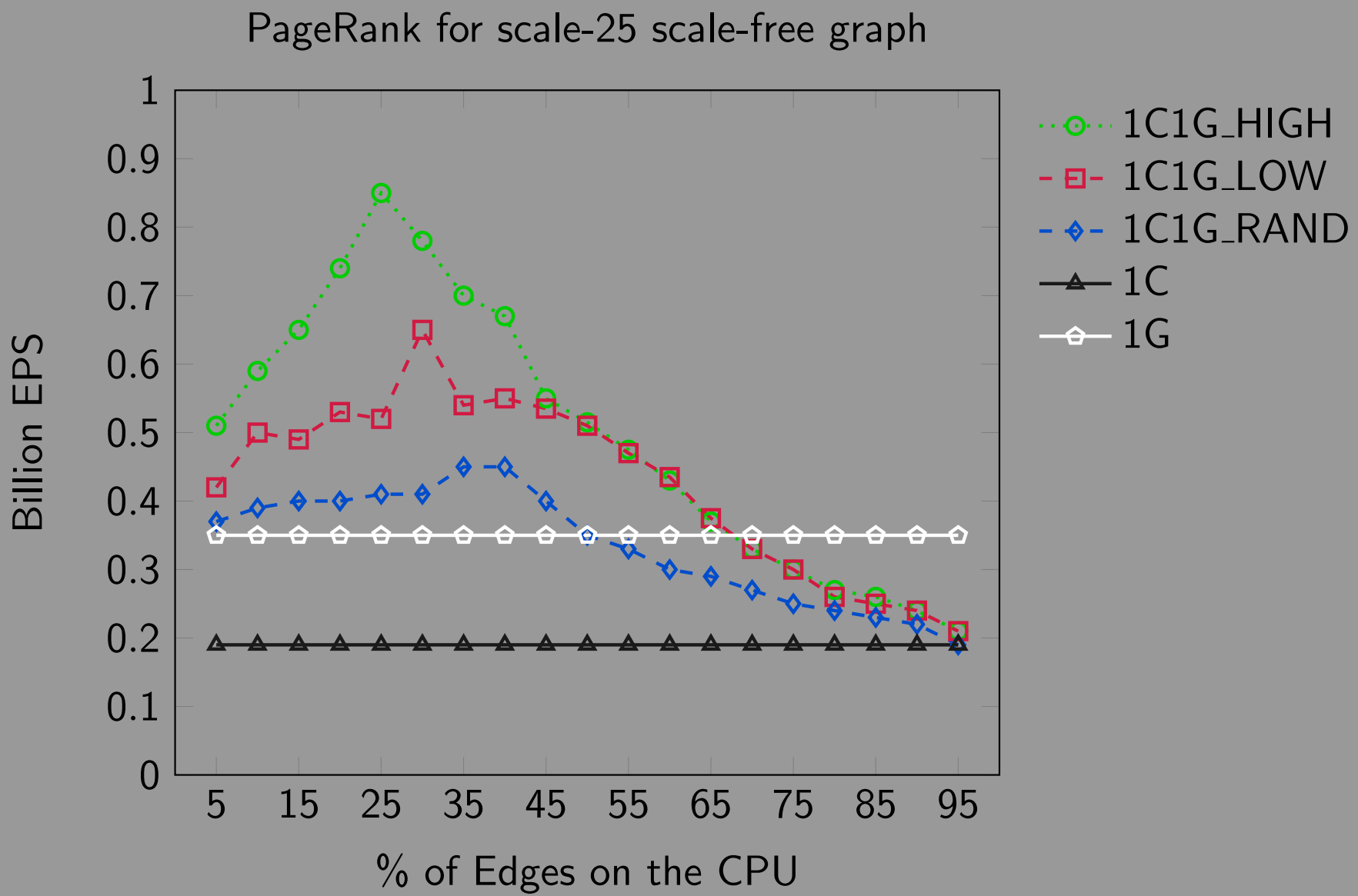
Evaluation


BFS for scale-25 scale-free graph
Billion TEPS
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
5
15
25
35
45
55
65
75
85
95
% of Edges on the CPU
1C1G_HIGH
1C1G_LOW
1C1G_RAND
1C
1G

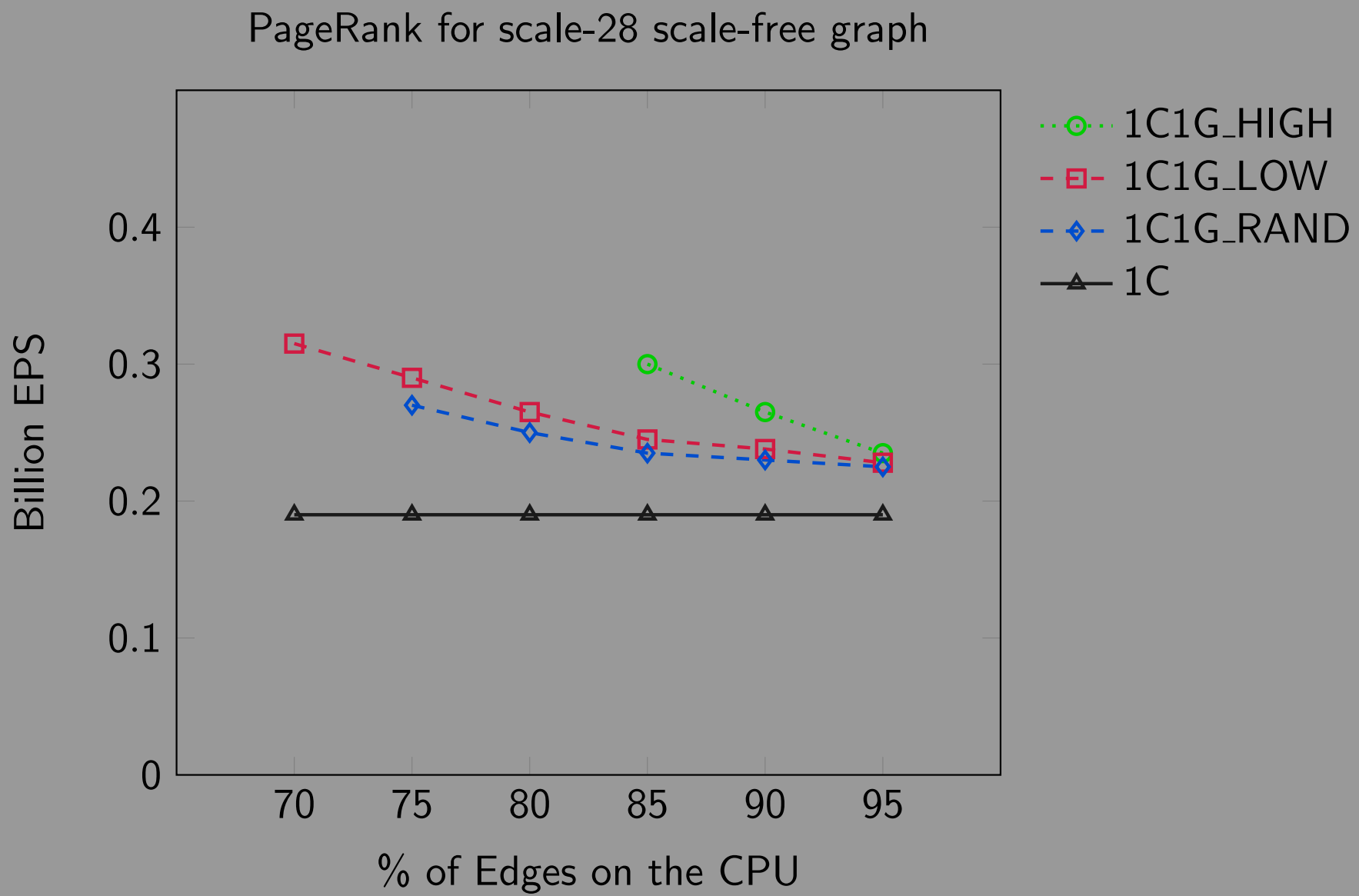
Evaluation


BFS for scale-28 scale-free graph
Billion TEPS
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
70
75
80
85
90
95
% of Edges on the CPU
1C1G_HIGH
1C1G_LOW
1C1G_RAND
1C

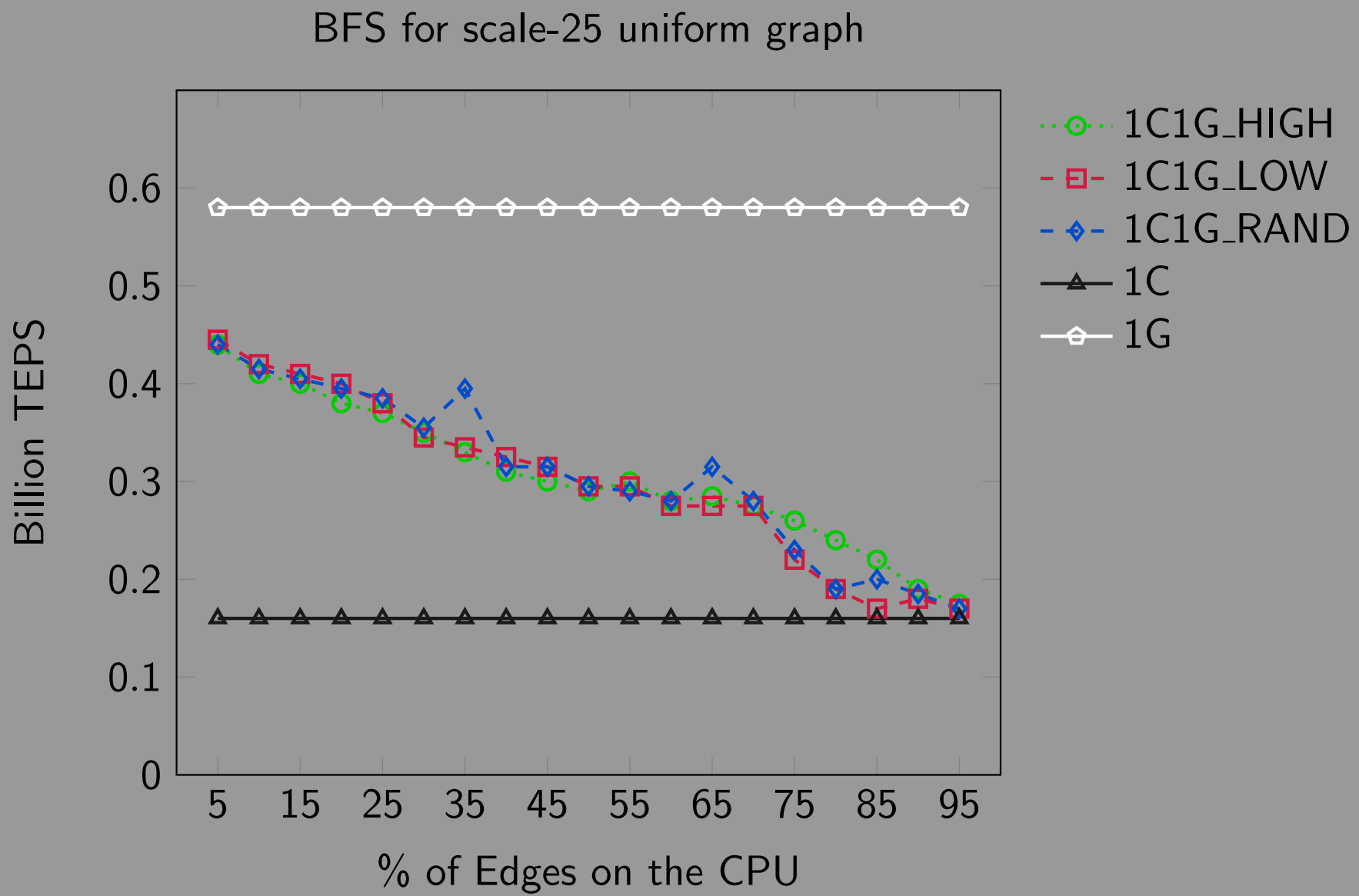
Evaluation


PageRank for scale-25 scale-free graph
Billion EPS
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
15
25
35
45
55
65
75
85
95
% of Edges on the CPU
1C1G_HIGH
1C1G_LOW
1C1G_RAND
1C
1G

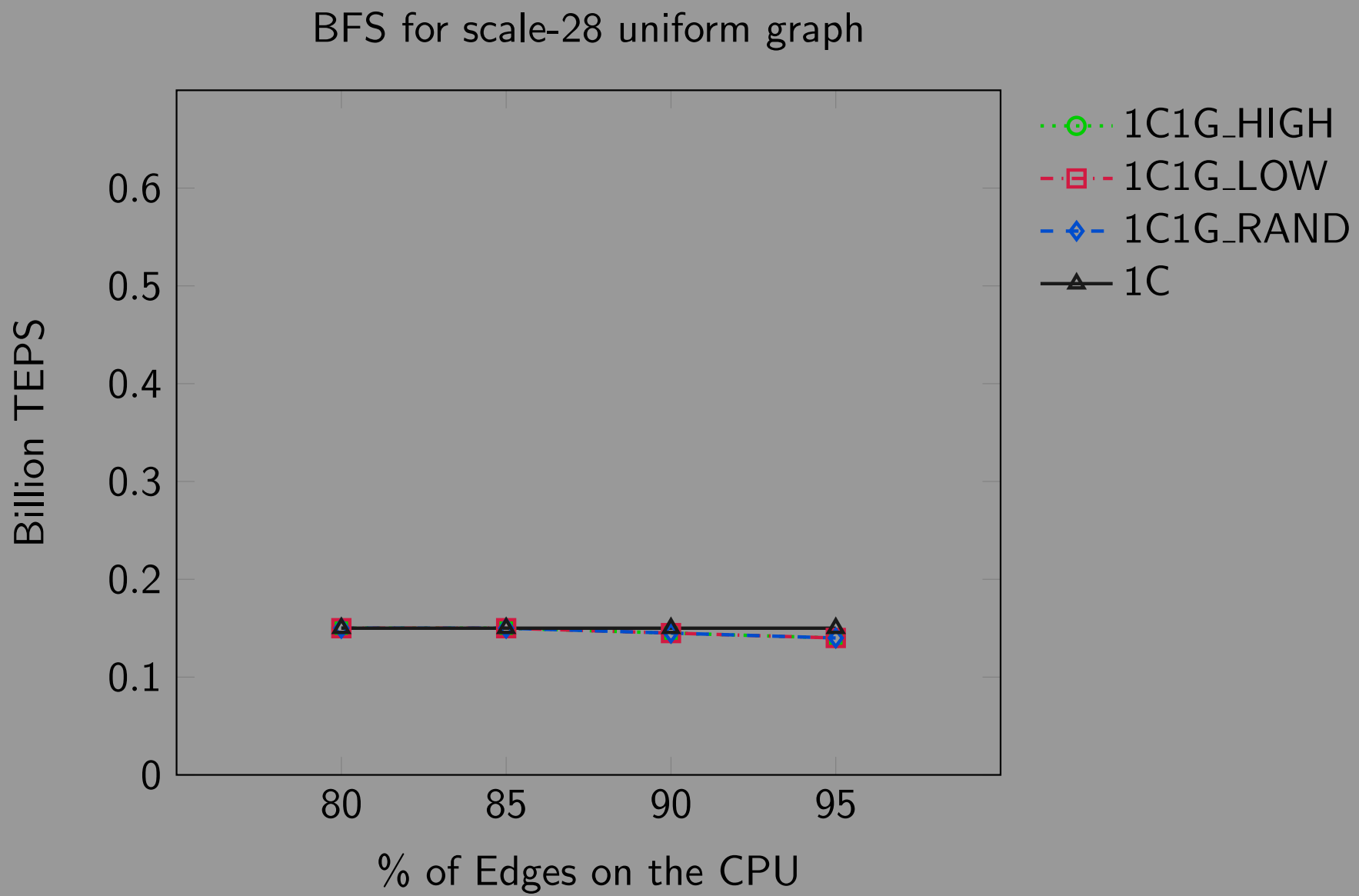
Evaluation


PageRank for scale-28 scale-free graph
Billion EPS
0
0.1
0.2
0.3
0.4
70
75
80
85
90
95
% of Edges on the CPU
1C1G_HIGH
1C1G_LOW
1C1G_RAND
1C

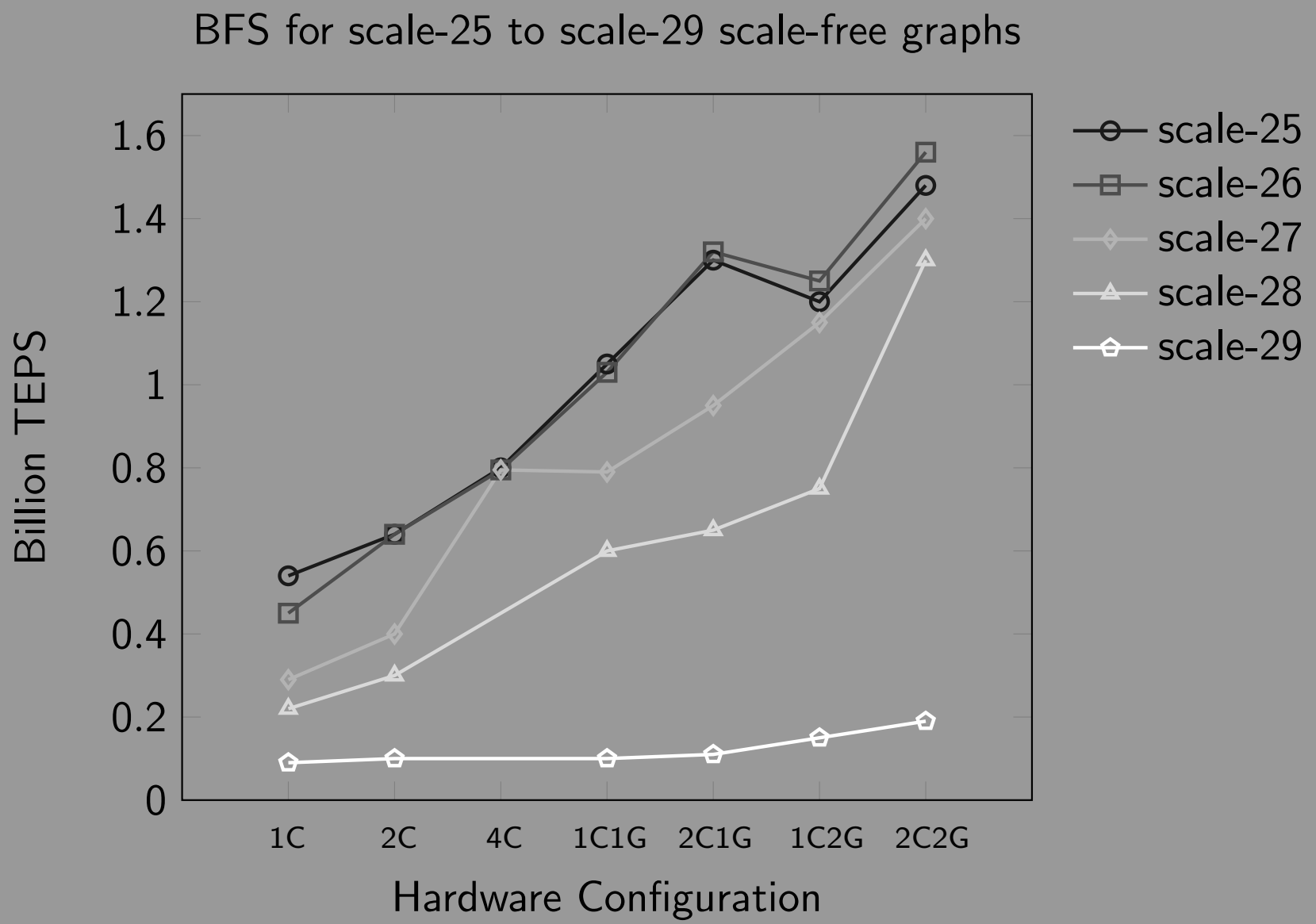
Evaluation


BFS for scale-25 uniform graph
Billion TEPS
0
0.1
0.2
0.3
0.4
0.5
0.6
5
15
25
35
45
55
65
75
85
95
% of Edges on the CPU
1C1G_HIGH
1C1G_LOW
1C1G_RAND
1C
1G

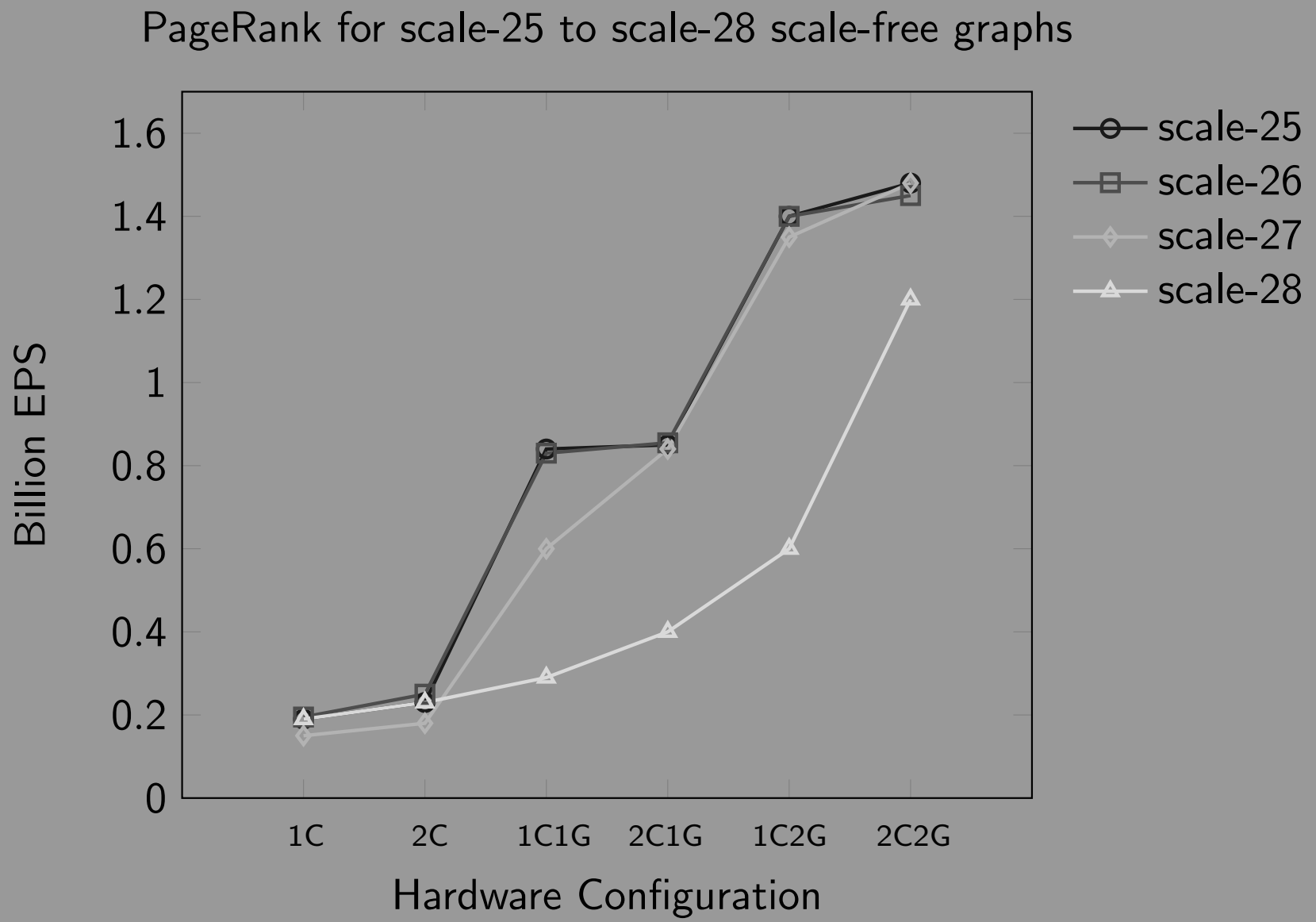
Evaluation


BFS for scale-28 uniform graph
Billion TEPS
0
0.1
0.2
0.3
0.4
0.5
0.6
80
85
90
95
% of Edges on the CPU
1C1G_HIGH
1C1G_LOW
1C1G_RAND
1C

Evaluation


BFS for scale-25 to scale-29 scale-free graphs
Billion TEPS
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1C
2C
4C
1C1G
2C1G
1C2G
2C2G
Hardware Configuration
scale-25
scale-26
scale-27
scale-28
scale-29

Evaluation


PageRank for scale-25 to scale-28 scale-free graphs
Billion EPS
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1C
2C
1C1G
2C1G
1C2G
2C2G
Hardware Configuration
scale-25
scale-26
scale-27
scale-28

Summary

- Cache-sensitive algorithms:
  - Large graphs: HIGH
  - Small graphs: LOW
- More compute-intensive algorithms:
  - Large graphs: LOW
  - Small graphs: HIGH
- Hybrid-system makes sense
- Graph topology is important

Personal Review

- Simple partition strategy
- Show in great detail how it improves performance
- Focus is on BFS
- Code is available

Questions?