Investigating the scale-invariance of graph algorithm performance

Tim van Zalingen

UvA

July 5, 2018

Supervisors (UvA):
Merijn Verstraaten & dr. ir. A.L. (Ana) Varbanescu

Research Project 19

Introduction

- Graph processing
  - Breadth First Search (BFS)
    - Parallel
    - Edge-centric vs vertex-centric implementation
- GPU
- Scaling

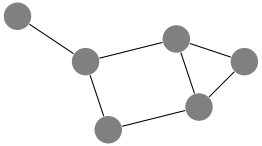

Figure 1: Example of a simple graph.

Figure 2: Left: CPU architecture, right: GPU architecture. (source: nvidia.com)

Scaling mechanism

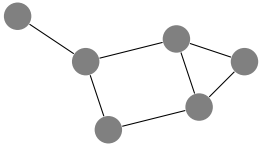

Figure 3: Original graph.

Scaling mechanism

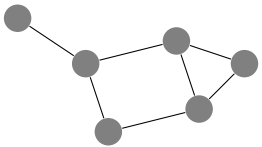

Figure 3: Original graph.

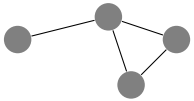

Figure 4: Sample of graph in figure 3.

Scaling mechanism

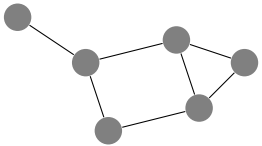

Figure 3: Original graph.

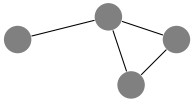

Figure 4: Sample of graph in figure 3.

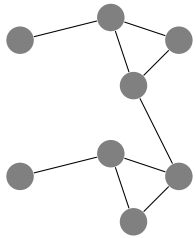

Figure 5: Scaled graph comprised of two samples as shown in figure 6.

Scaling parameters

- Number of interconnections
- High degree or random vertex bridges
- Sample size
- Topology:

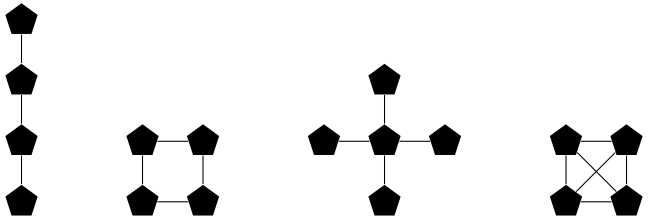

Figure 6: Illustration of different topologies

Research Question

Is the relative performance of graph algorithms scale-invariant?

- What are the effects of tuning the scaling parameters?
- Do implementations show similar behaviour under scaling?

Method

- Scaling parameters
  - What parameters?
- Comparison of graphs
  - Diverse set of graphs
  - Scaled versions of this set

Results: Scaling parameters - actor-collaboration

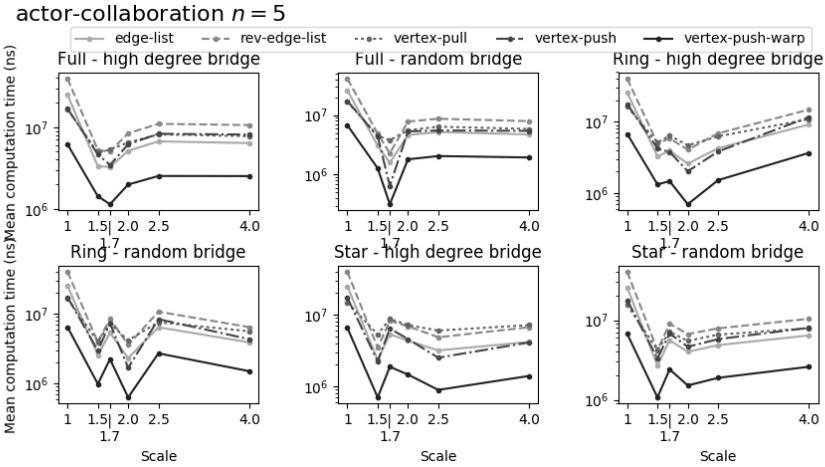

Figure 7: Comparison between scaling parameters for the actor-collaboration graph.

Results: Scaling parameters - dbpedia-starring

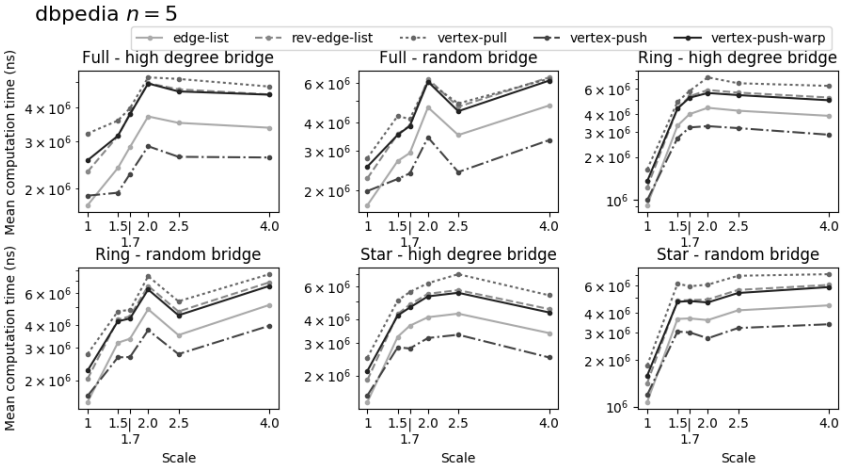

Figure 8: Comparison between scaling parameters for the dbpedia-starring graph.

Results: Graph comparison

Mean algorithm performance on different scales for given graphs

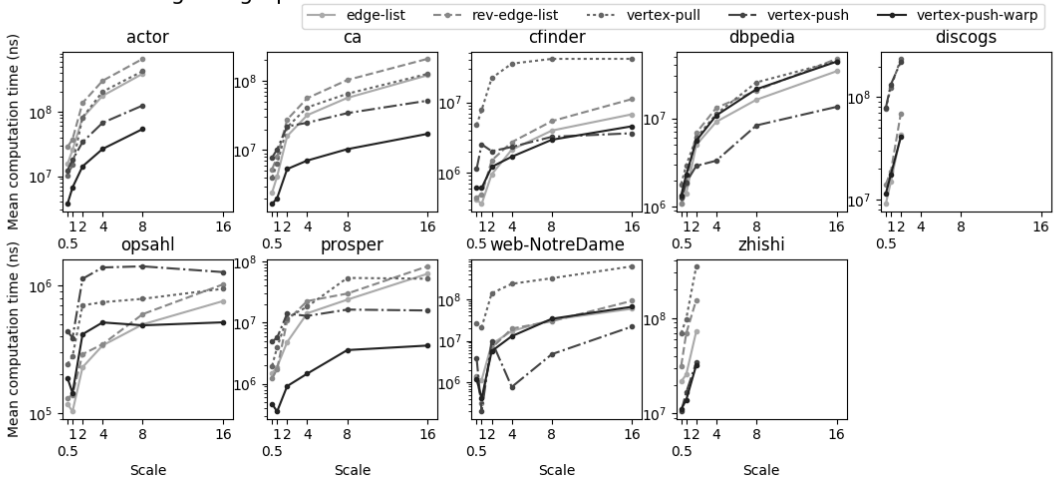

Figure 9: Comparison between algorithm mean computation time for different graphs.

Vertex-push transition point

Observation: The point where vertex-push starts to outperform other algorithm implementations, is in the hundred thousands of vertices.

For similar graphs, is this transition point similar as well?

Results: Vertex-push transition point

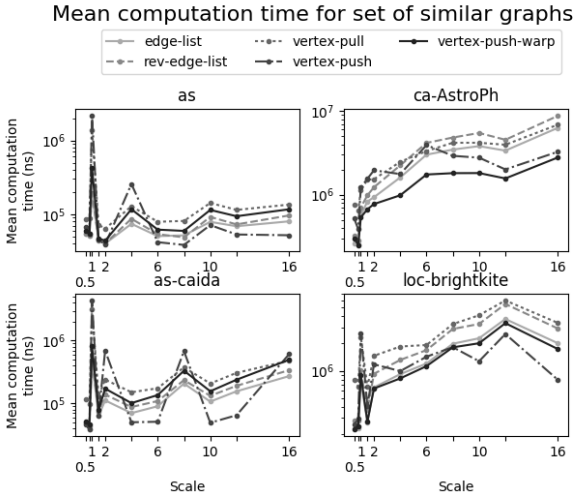

Figure 10: Mean execution time over scale on similar graphs.

All graphs are/have:

- Undirected
- Unweighted
- Average degree around 5

vertices/edges:
ca: 6K/13K
as-caida: 26K/53K
AstroPh: 19K/198K
loc: 58K/214K

Discussion: Conclusion

The relative performance of BFS implementations can be stable under scaling. However, it is not fully scale-invariant.

- Tuning scaling parameters has no great effect.
- Transition points and stability depends on the graph.
- The vertex-push implementation scales better. Results hint to a predicable transition point. Appears to depend on number of edges per vertex.

Discussion: Limitations

- Effects of scaling parameters only investigated on two graphs.
- Set of graphs diverse and limited.
- Conclusions only valid for current implementation of scaling and BFS algorithms.

Future work

- Investigate more graph algorithms.
- Compare similar graphs.
- Investigate variants of BFS implementations.
- Can transition points be determined?

Summary

The scaling parameters have low impact on how algorithm implementations scale.

Relative performance is stable around a size.

When scaling to multiple times the original size, algorithms can switch in ranking.

Vertex-push appears to scale best.