PREGEL: A SYSTEM FOR LARGE-SCALE GRAPH PROCESSING

What's a few billion graph nodes anyway?

CONTEXT

MapReduce

Sawzall

PigLatin Dryad

CGM

Scalable, fault-tolerant platform with a usable API that can express arbitrary graph algorithms.

KEY IDEAS

Graphs are weird

- Memory access locality
- Work per vertex
- Parallelism
- Relatively Sparsely Connected
- Failure!

Bulk Synchronous Parallel Model

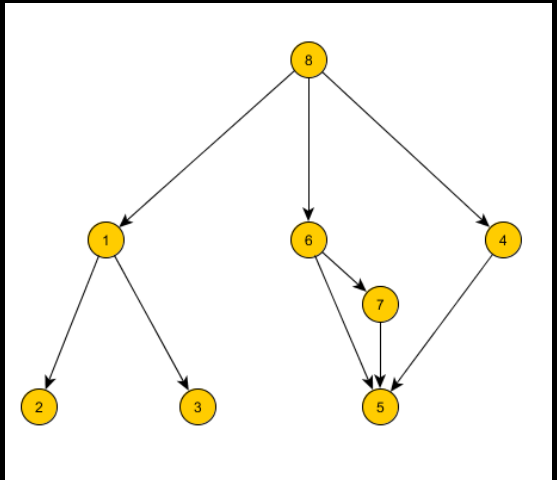
- Supersteps
- Compute at each vertex
- Messages

‘Think like a vertex’

- Synchronous
- Compute at each vertex
- Messages

IMPLEMENTATION

Vertex = String identifiers + Mutable Value + edges



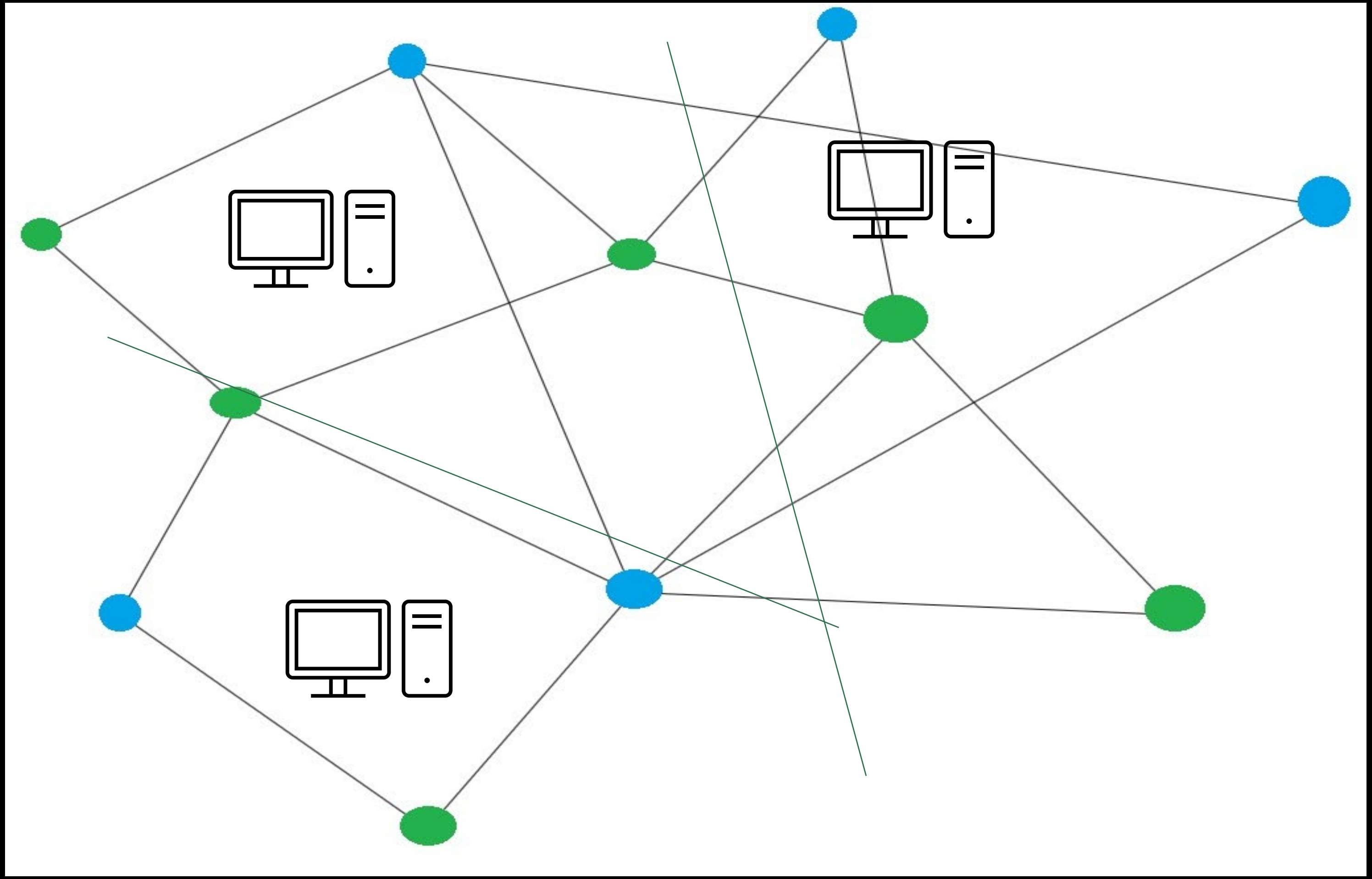
Input

Program

- Written in C++ API
- Each vertex is typed
- Override Compute()
- Message Passing
- Lots of file formats allowed

Output = Set of Values Determined by Vertices

IMPLEMENTATION



- Pure Message Passing
- Supersteps
- Halt votes
- Combiners
- Aggregators
- Partial Ordering for Topology Mutations

EVALUATION

- Experiments on binary trees for scaling, log-normal random graphs to represent real life
- Single-Source Shortest Paths
- > 1 billion vertices and edges
- In 2010 there were 256 million websites, chrome only had 14.9% market share
- Now 30-50 billion are pages indexed by google
- More diverse tests?

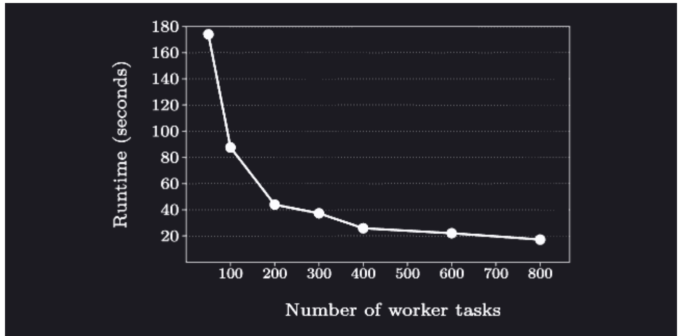


Figure 7: SSSP—1 billion vertex binary tree: varying number of worker tasks scheduled on 300 multicore machines

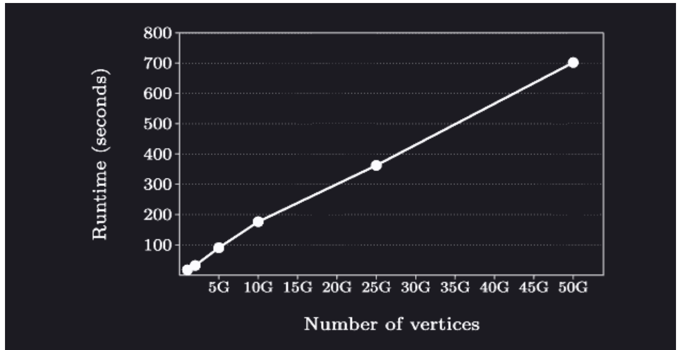


Figure 8: SSSP—binary trees: varying graph sizes on 800 worker tasks scheduled on 300 multicore machines

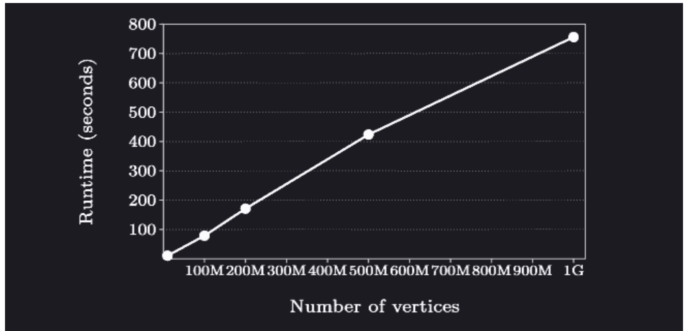


Figure 9: SSSP—log-normal random graphs, mean out-degree 127.1 (thus over 127 billion edges in the largest case): varying graph sizes on 800 worker tasks scheduled on 300 multicore machines

CONTRIBUTIONS

- Scalable
- Model description
- Usability
- Performance satisfactory for billions of vertices
- PageRank, Shortest Path Finding, Bipartite Matching, Semi-Clustering

ANALYSIS

- What about asynchronous programs and concurrency issues?
- What about more dense graphs?
- What better ways of partitioning the graph are there?
- Pregel took off very quickly, this meant they couldn’t make large changes to their API, was this bad?
- Think like a vertex – good?
- Real graphs have skewed power law distributions!

IMPACT

- Foundational for Google PageRank
- Inspired Facebook’s Apache Giraph and Apache Spark’s Graph library
- Useful for GNN processing (or at least what it inspired)
- Used for graph databases
- Showed scalability with simplicity
- Distribution

FURTHER WORK

- Not stored all in ram
- Relax synchronity
- Assigning vertices to machines – dynamic repartitioning

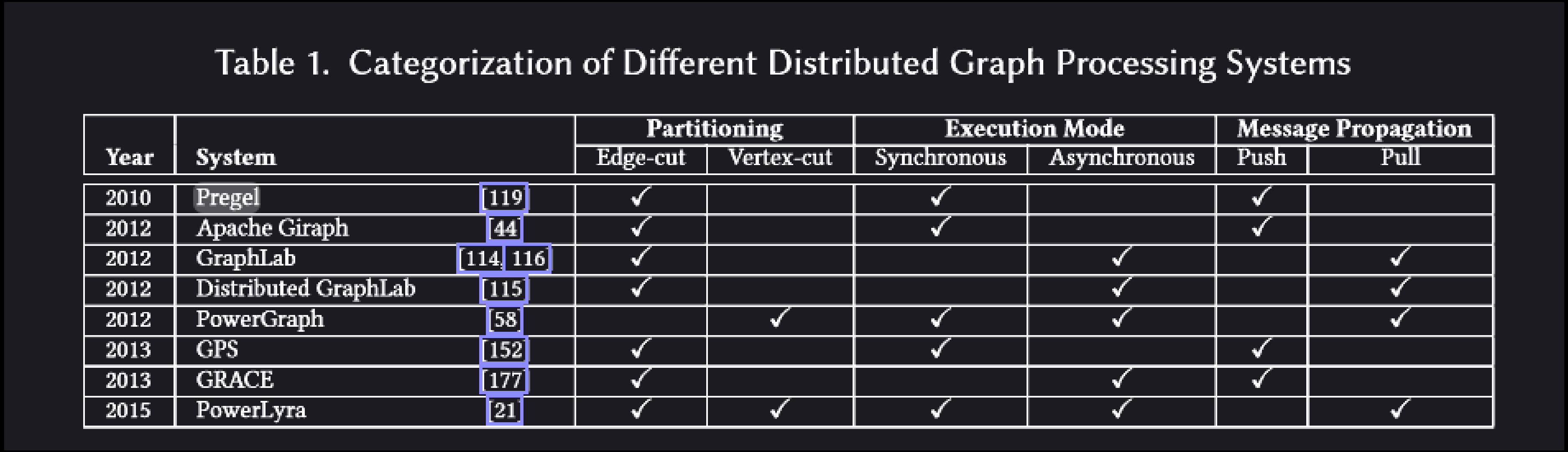


Table 1. Categorization of Different Distributed Graph Processing Systems

Year	System	Partitioning		Execution Mode		Message Propagation	
		Edge-cut	Vertex-cut	Synchronous	Asynchronous	Push	Pull
2010	Pregel [119]	✓		✓		✓	
2012	Apache Giraph [44]	✓		✓		✓	
2012	GraphLab [114, 116]	✓			✓		✓
2012	Distributed GraphLab [115]	✓			✓		✓
2012	PowerGraph [58]		✓	✓	✓		✓
2013	GPS [152]	✓		✓		✓	
2013	GRACE [177]	✓			✓	✓	
2015	PowerLyra [21]	✓	✓	✓	✓		✓

https://dl.acm.org/doi/pdf/10.1145/3597428

REFERENCES:

[1] Grzegorz Malewicz, et al. 2010. Pregel: a system for large-scale graph processing. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (SIGMOD '10), 135–146. https://doi.org/10.1145/1807167.1807184

[2] Besta M, Gerstenberger R, Peter E, Fischer M, Podstawski M, Barthels C, Alonso G and Hoefler T. (2023). Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries. ACM Computing Surveys. 56:2. (1-40). Online publication date: 29-Feb-2024.

[3] Vatter J, Mayer R and Jacobsen H. (2023). The Evolution of Distributed Systems for Graph Neural Networks and Their Origin in Graph Processing and Deep Learning: A Survey. ACM Computing Surveys. 56:1. (1-37). Online publication date: 31-Jan-2024.

[4] J. Shun, G.E. Blelloch, Ligra: a lightweight graph processing framework for shared memory, in Proceedings of the 18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (2013), pp. 135–146

[5] Luiz Barroso, Jeffrey Dean, and Urs Hoelzle, Web search for a planet: The Google Cluster Architecture. IEEE Micro 23(2), 2003, 22–28.