PageRank Algorithm : Multithreading Vs GraphX Spark Implementation

CS-527 Parallel Computer Architectures

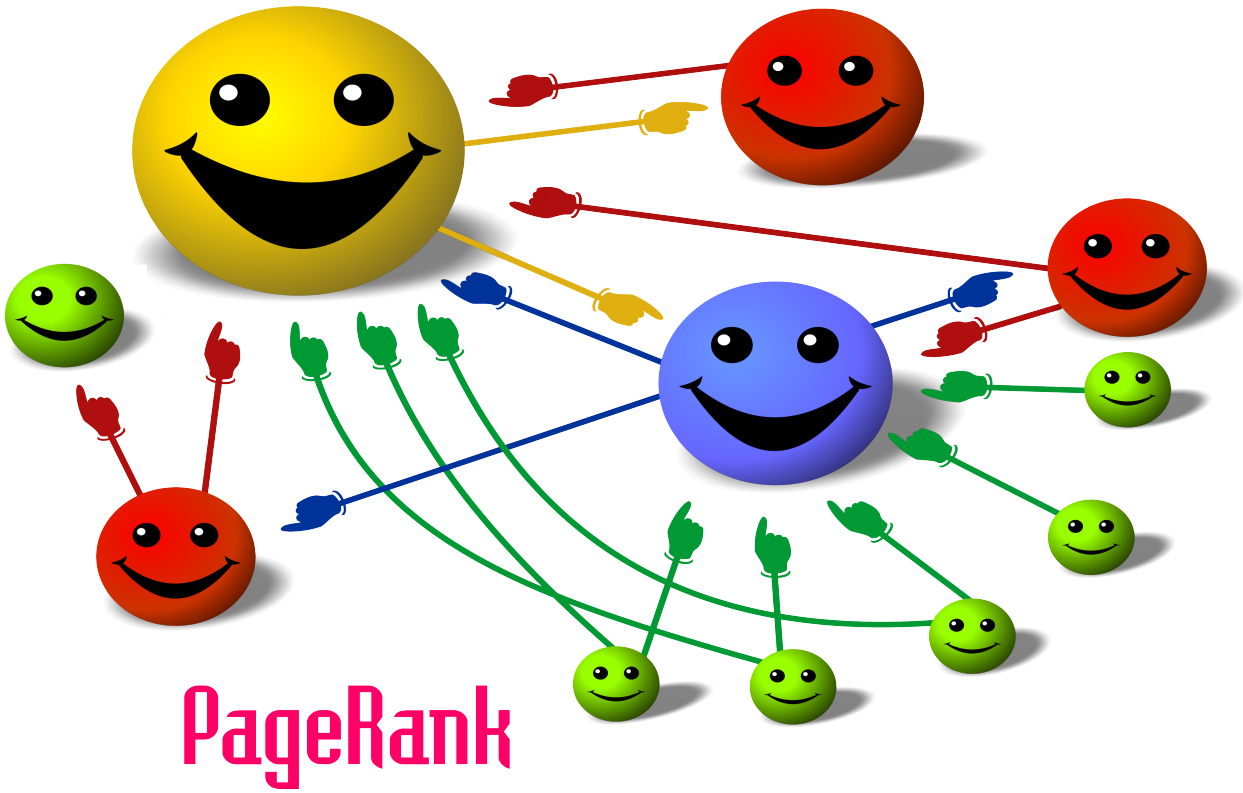
Jack Kolokasis

December 20, 2017

Outline

1. Motivation & Overview
2. Multithreading Implementation
3. GraphX Spark Implementation
4. Evaluation Results
5. Conclusions

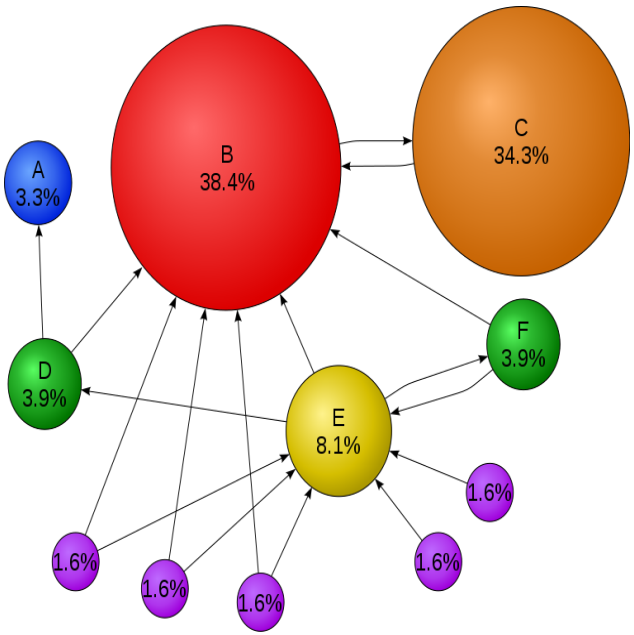
PageRank Is Fun



PageRank if fun.... but there are a LOT of Pages with a LOT of Links and it becomes a LOT of work to calculate.

Overview

How Does PageRank Work



- Directed Graph
  (nodes point to other nodes but it's one way street)
- Initialize all the node with a default probability:

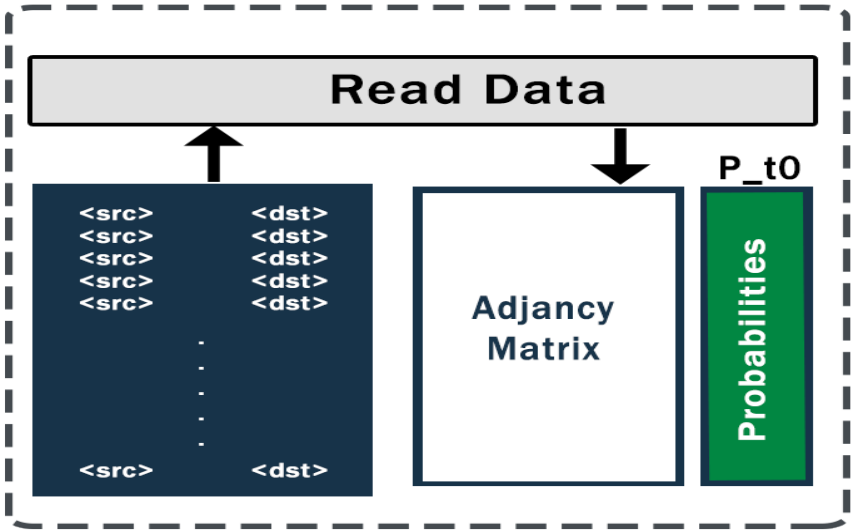
$$PR(p_i; 0) = \frac{1}{N}$$

- For every node in the graph calculate a rank on every iteration:

$$PR(p_i; t+1) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{out(p_j)}$$

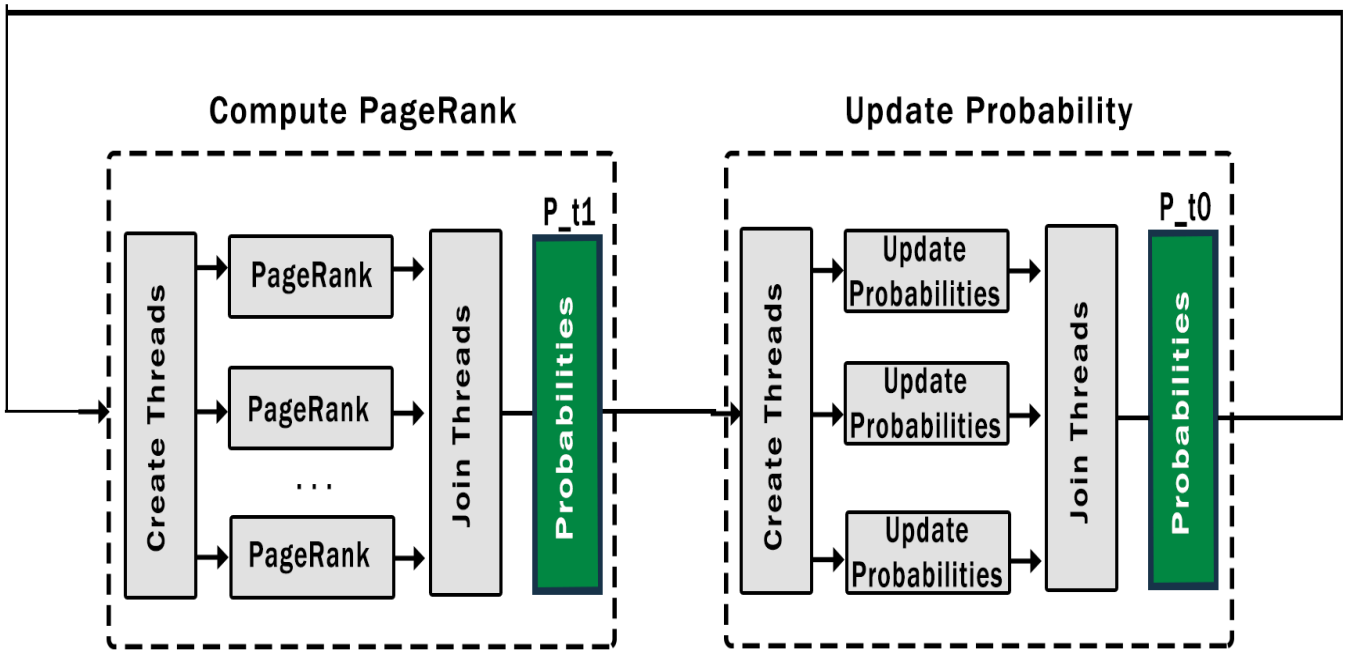
Multithreading Implementation

Read Input Datase and Construct the Graph

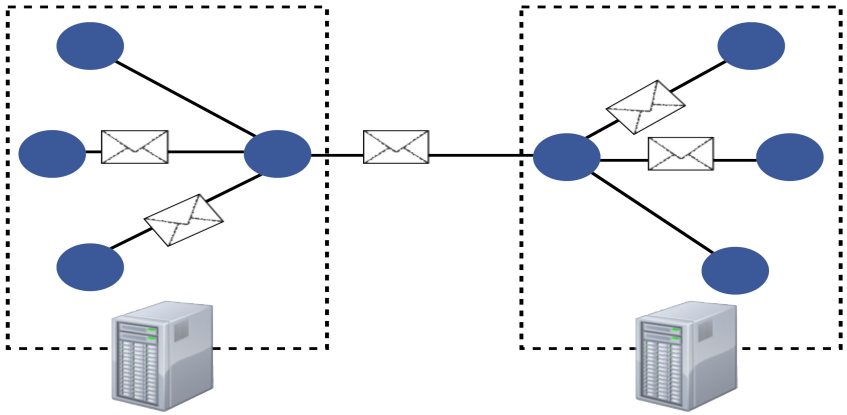


Multithreading Implementation

Computation of PageRank Algorithm



GraphX Overview



Vertex Abstraction

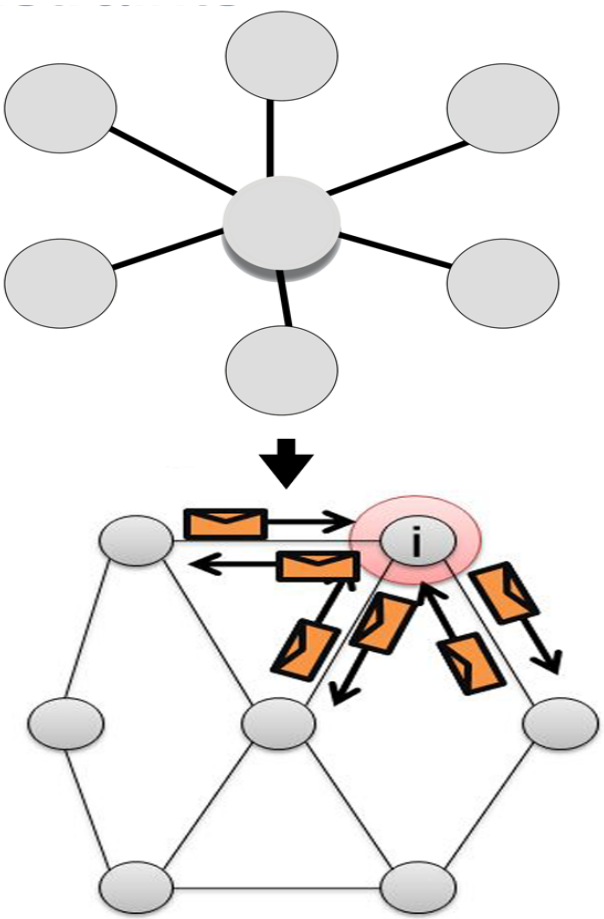
- Vertices exchange messages

System Representation :

- Graph is partitioned accross cluster nodes
- Machines keeps vertices in memmory
- Messages are local or accross network

GraphX PageRank Implementation

- Load Graph from text file
- Initialize graph vertices
- Insert a weight on the edges
- Run Pagerank
  - Update vertex probabilities using messages



Evaluation

- Compare Computation Time of PageRank Multithreading against Spark GraphX Pagerank Implementation executing on single node
  - Machine Specifications:
    - **CPU:** Intel(R) Xeon(R) CPU E5-2630 V3 @ 2.40GHz
    - **Number of Cores:** 32 cores
    - **Memmory Size:** 256GB
    - **OS:** CentOS Linux **Version:** 7 (Core)
  - Multithreading Pagerank
    - Use 1, 2, 4, 8, 16, 32 number of threads respectively. One Thread per core !

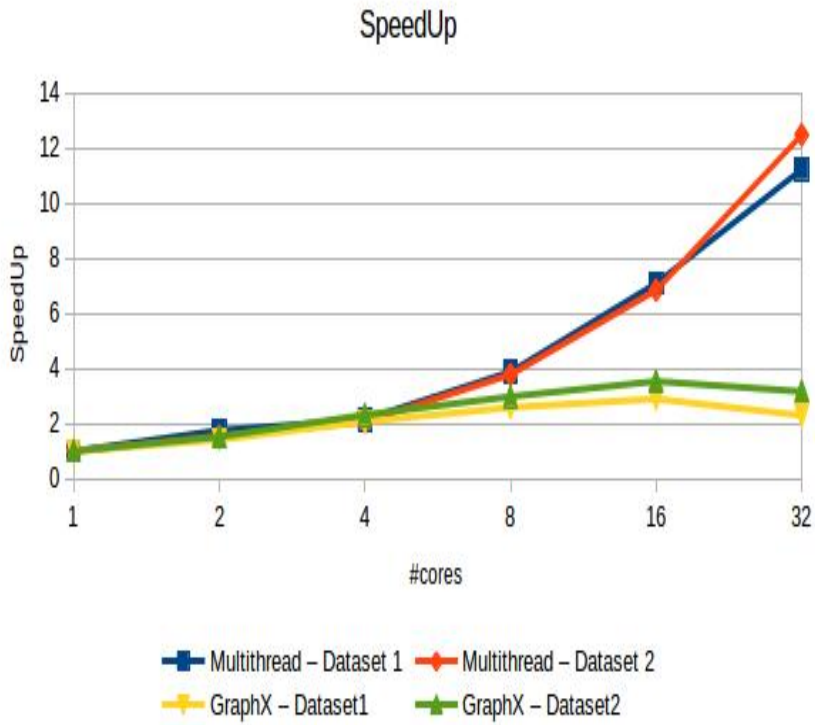
Evaluation

- Spark GraphX Pagerank (1 worker)
  - 1 worker: 1, 2, 4 ,8, 16, 32 cores respectively
- Spark GraphX Pagerank (Multi Workers)
  - 1 worker: 32 cores and 256GB memory
  - 2 worker: 16 cores and 125GB memory
  - 4 worker: 8 cores and 60GB memory
  - 8 worker: 4 cores and 30GB memory
  - 16 worker: 2 cores and 15GB memory
- Average of 5 runs
- Graphs Datasets

Dataset	Vertex	Edges
Dataset1	10,000	10,000,000
Dataset2	100,000	20,000,000

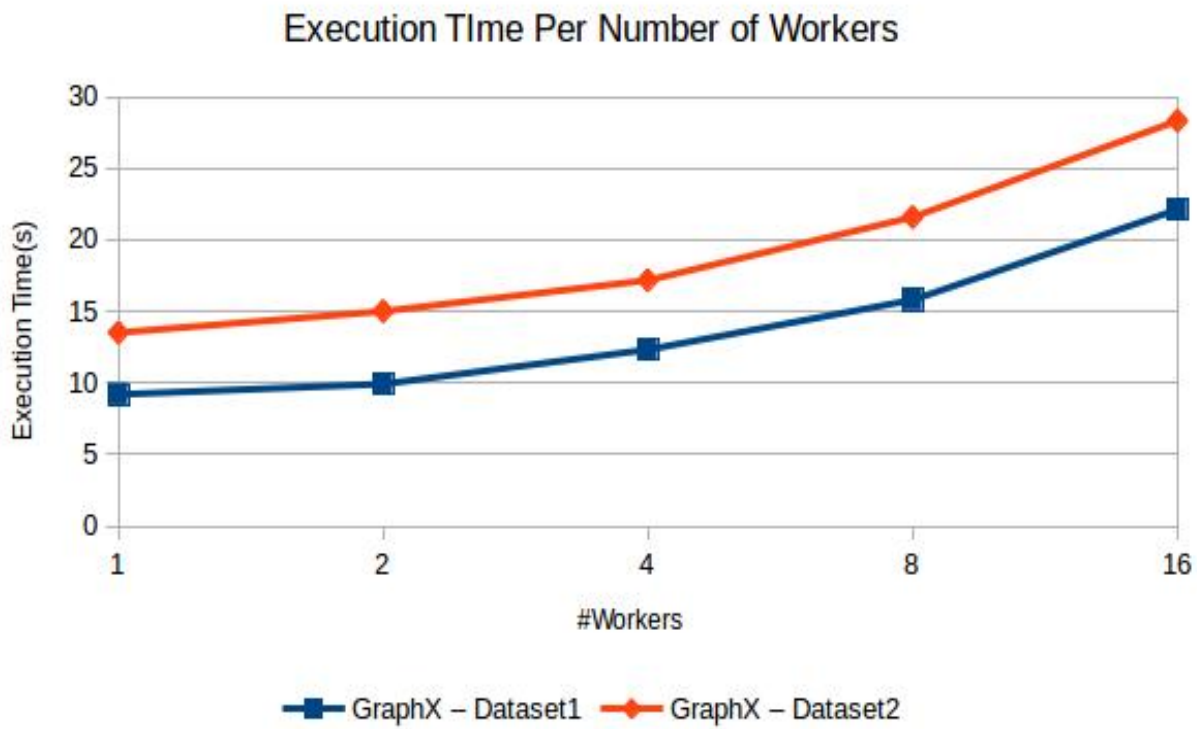
Results

Execution Time and SpeedUp



Results

Execution Time Spark GraphX Using Multiple Workers



Conclusions

- Multithreading PageRank achieve speedUp up to 10% when running on 32 cores according to GraphX implementation
- Spark GraphX introduce scalability but with cost on computation overhead (eg. JVM)
- Using multiple workers the execution time is higher, because of the communication cost between the workers.
- Better to use one worker per machine on Spark