Estimating the Impact of Communication Schemes for Distributed Graph Processing

Tian Ye[1], Sanmukh R. Kuppannagari[1], Ceasr A. F. De Rose[2],
Sasindu Wijeratne[1], Rajgopal Kannan[3], Viktor K. Prasanna[1]
[1]University of Southern California, [2]PUCRS, [3]US Army Research Lab
tye69227@usc.edu
https://fpga.usc.edu

ISPDC 2022

Background and Motivation

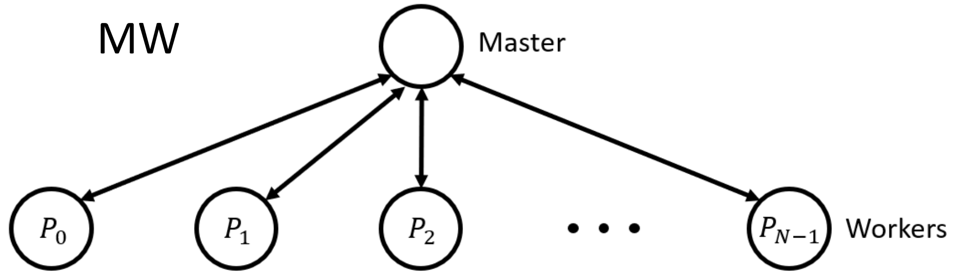
- Extreme scale graph analytics require distributed graph processing on cloud/clusters
- Graph $G = (V, E)$ is partitioned and allocated to N computing nodes
- Communication cost has significant impact on the performance

- This work
  - Identify and define communication schemes in graph analytics
  - Develop performance models to estimate communication time that enable trade-off analysis before graph analytics run on cloud/clusters

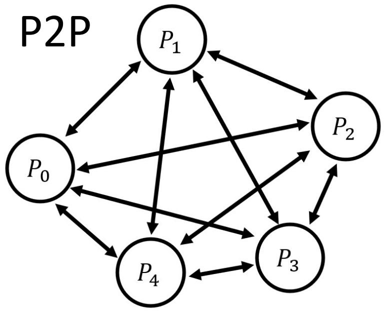
Communication Schemes

- Type of data being communicated
  - Vertex Proportional Communication (VPC)
    - Each node broadcasts vertex attributes to its neighbors
  - Edge Proportional Communication (EPC)
    - Each node sends edge-specific messages along outgoing edges
- Underlying virtual communication network
  - Master-Worker (MW)
  - Ring
  - Peer-to-Peer (P2P)

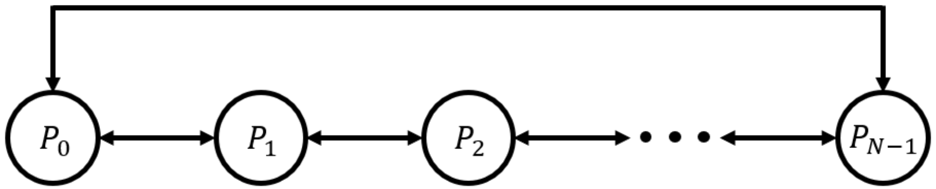
MW



P2P



Ring



Vertex Proportional Communication (VPC)

Example of algorithm using VPC

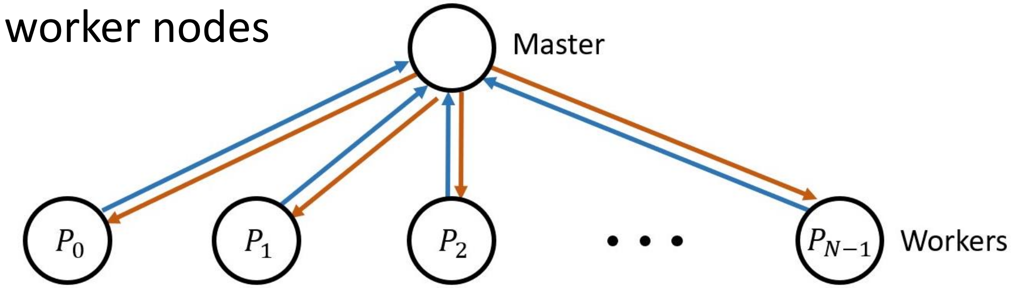
Algorithm 1: VPC BASED PAGERANK

Input: A graph $G^* = (V^*, E^*)$
Output: PageRank results for all vertices $PR[:]$

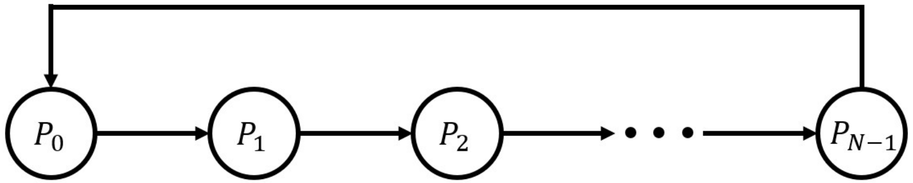
$PR[:] \leftarrow 1/|V|$
while *Convergence* $>$ *Expected Convergence* **do**
  for *each vertex* $u \in V^*$ **do**
    $sum \leftarrow 0$
    for *each* $v \in Adj(u)$ **do**
      $sum \leftarrow sum + PR[v]/OutDeg(v)$
    $PR[u] \leftarrow (1 - df)/|V| + df \times sum$
    `// df = damping factor`
  All_to_All_Broadcast(PR) Communication Phase

To broadcast the vertex attribute (PR)

- Master-Worker Network (VPC-MW)
  - Each worker node sends the PR values it possesses to the master node
  - The master node broadcasts all PR values to all worker nodes



- Ring Network (VPC-Ring)
  - Each node sends data to right neighbor and receives data from left neighbor
  - Repeat $(N-1)$ iterations



Edge Proportional Communication (EPC)

Example of algorithm using EPC

Algorithm 2: EPC BASED PAGERANK

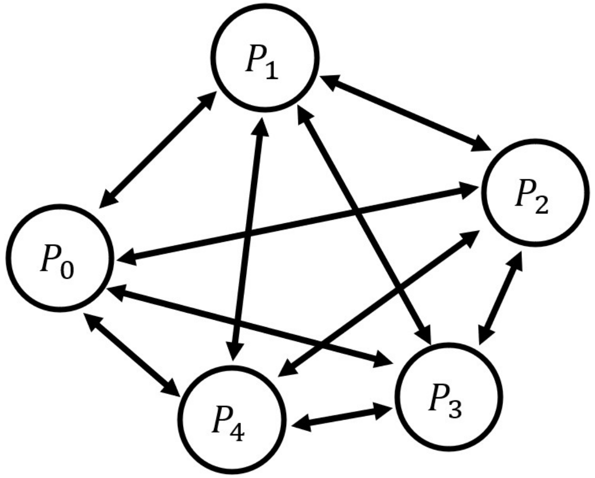
Input: A graph $G^* = (V^*, E^*)$
Output: PageRank results for all vertices $PR[:]$

$PR[:] \leftarrow 1/|V|$
while *Convergence* > *Expected Convergence* **do**
  $sum\_iter[:] \leftarrow 0$
  for *each vertex* $u \in V^*$ **do**
    $contribute \leftarrow PR[u]/OutDeg(u)$
    for *each destination* $v \in Adj(u)$ **do**
      $sum\_iter[v] \leftarrow sum\_iter[v] + contribute$
  All_to_All_Personalized_Communication(sum_iter)
  for *each vertex* $u \in V^*$ **do**
    $PR[u] \leftarrow (1 - df)/|V| + df \times sum$

Communication Phase

To implement All-to-all Personalized Comm.

- Peer-to-Peer Network (EPC)
  - In Iteration i, Node P_i sends its data to all other nodes



Performance Modeling (1)

- Motivation
  - Design space exploration for graph analytics is large
  - Sub-optimal choices lead to long running time and high monetary costs

- Benefits of performance modeling
  - Enable quick trade-off analysis
  - Help to understand the impact of various parameters (e.g., communication schemes, number of nodes) on the performance

Performance Modeling (2)

- t_s = Average communication latency between two nodes
- t_w = Average communication time to transfer a word

- To estimate t_s and t_w
  - Communicate data of L words and measure the round-trip time (RTT)
  - Repeat with different values of L, and apply linear regression

$$RTT = 2(t_s + t_w \cdot L)$$

Performance Modeling (3)

- VPC-MW Communication Time

$$T_{VPC-MW} = \underbrace{N \cdot \left(t_s + \frac{V}{N} \cdot t_w\right)}_{N \text{ workers send data to the master sequentially}} + \underbrace{N \cdot (t_s + V \cdot t_w)}_{N \text{ workers receives data from the master sequentially}} = 2Nt_s + (N+1)Vt_w$$

- VPC-Ring Communication Time

$$T_{VPC-Ring} = (N-1)\left(t_s + \frac{V}{N} \cdot t_w\right)$$

  - For each node, sending and receiving data are non-blocking, i.e., happening simultaneously

Performance Modeling (4)

- EPC Communication Time

$$T_{EPC} = \sum_{i=1}^{N}\left(t_s + t_w \sum_{j \neq i} \eta_{ij}\alpha_{ij}\right) = Nt_s + t_w \sum_{i=1}^{N}\sum_{j \neq i} \eta_{ij}\alpha_{ij}$$

- η_{ij} is average size of message for a destination vertex
- α_{ij} is # vertices in Partition j with at least one incoming edge from Partition i
- $\alpha_{ij} = \left\|\mathbf{A}_{ji}\mathbf{1}\right\|_0$, $\mathbf{A}_{ji}$ is a sub-matrix in the graph's adjacency matrix with rows for Partition j and columns for Partition i

Experimental Evaluation (1)

- Platforms
  - High-Performance Cluster (HPC)
    - Dual Intel Xeon 10-core 2.4 GHz processors, up to 64 GB memory
  - Chameleon Cloud's MPICH3 Bare-Metal Cluster
    - Each node has 24 Intel Xeon E5-2670 v3 2.3 GHz CPUs, 128 GB memory
    - Machines connected with InfiniBand
- Datasets

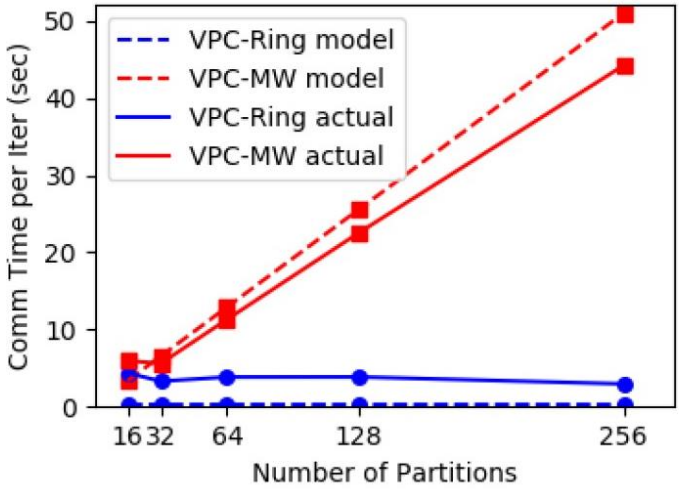
PROPERTIES OF DATASETS

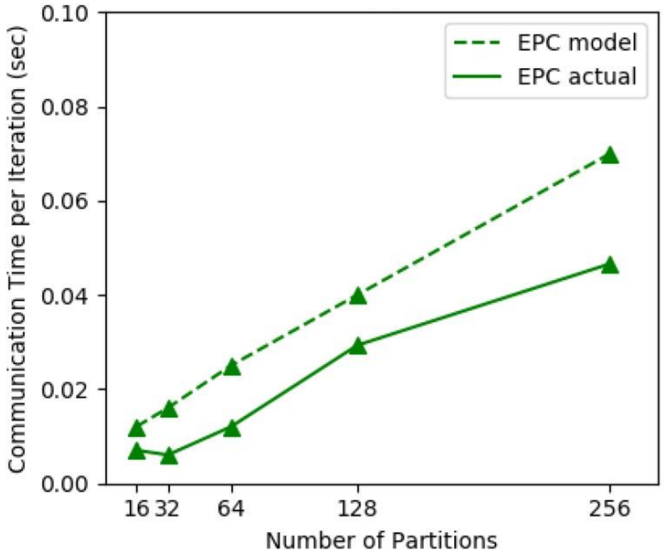
Graph	Edges	Vertices	Avg. degree
uk-union-2006-06-2007-05 [19]	5 507 679 822	133 633 040	41.215
twitter-2010 [20]	1 468 365 182	41 652 230	35.253
webbase-2001	1 019 903 190	118 142 155	8.633

- Benchmarks
  - PageRank (PR)
  - Weakly Connected Components (WCC)

Experimental Evaluation (2)

- Results for *uk-union-2006-06-2007-05* dataset and PageRank on HPC





- Predictions are close to actual evaluations / have similar trends
- Congestions may occur as the data center is public

Experimental Evaluation (3)

- Insight 1: VPC-Ring and EPC consistently outperform VPC-MW
- Insight 2: VPC-Ring has the best scalability
  - For VPC-Ring, communication time almost stays constant when N increases
  - For VPC-MW and EPC, higher N leads to longer communication time but lower storage at each node

Experimental Evaluation (4)

- Insight 3: In most practical cases, EPC outperforms VPC-Ring

$$T_{VPC-Ring} \approx V t_w$$

$$T_{EPC} \approx t_w \sum_{i=1}^{N} \sum_{j \neq i} \alpha_{ij} = N d_{po} t_w = \left(\frac{N d_{po}}{V} \right) \cdot V t_w$$

d_{po}: average out-degree of all partitions

  - Graph partitionings usually have high intra-partition connectivity and low inter-partition connectivity such that $\frac{N d_{po}}{V} < 1$
- Insight 4: Hypothetical scenario exists where VPC-Ring will outperform EPC (d_{po} is high)
  - Partitioned graph has low locality
  - Few vertices in the same node share common destinations

Experimental Evaluation (5)

- Insight 5: Impact of partitioning schemes on communication time
  - For VPC-Ring and VPC-MW
    - $T_{VPC-Ring}$ and T_{VPC-MW} only depend on V and N, irrelevant to how graph is partitioned
    - Applications using VPC should focus on partitioning that optimizes computation loads
  - For EPC
    - Partitioning is optimal with

$$\min \sum_{i}^{N} \sum_{j \neq i} \alpha_{ij} = \min \sum_{i}^{N} \sum_{j \neq i} \left\| \mathbf{A}_{ji} \mathbf{1} \right\|_0$$

    - Heuristics should be developed to optimize this target

Conclusion

- We developed and validated performance estimation models for communication schemes for distributed graph processing frameworks
- Our models enable the analysis of trade-offs between partitioning schemes and communication schemes in early development stages

Thanks for your listening!

https://fpga.usc.edu