

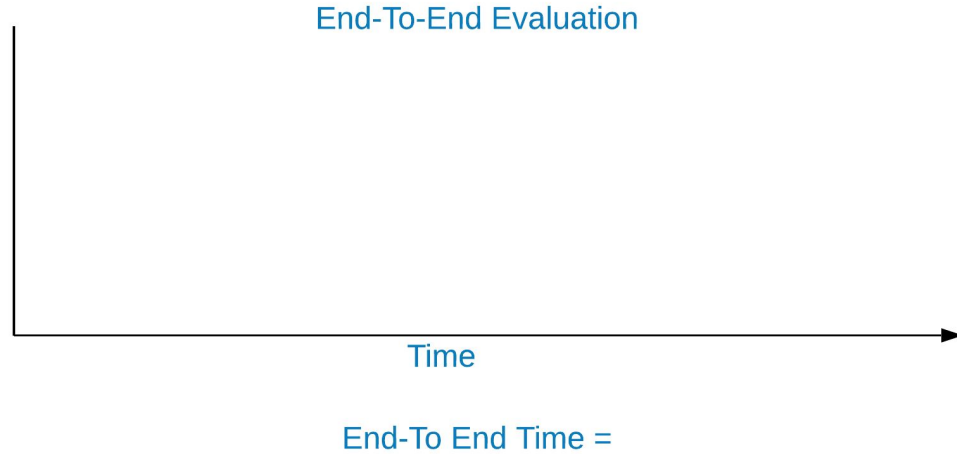


Accelerating **Graph Processing**

Presenter : Abdullah Mughrabi

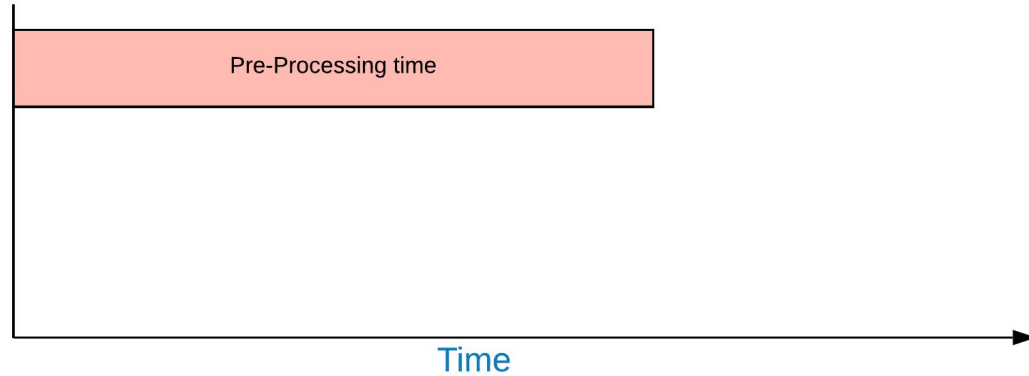


End to end acceleration



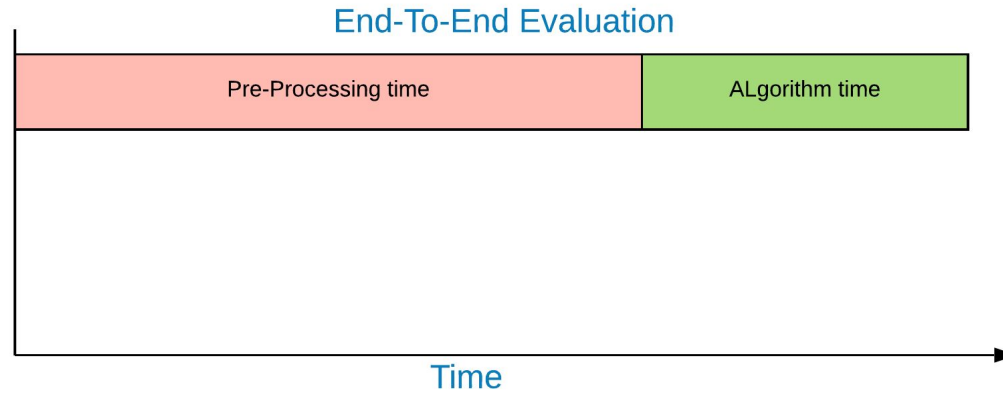


End to end acceleration



End-To End Time = Pre-processing

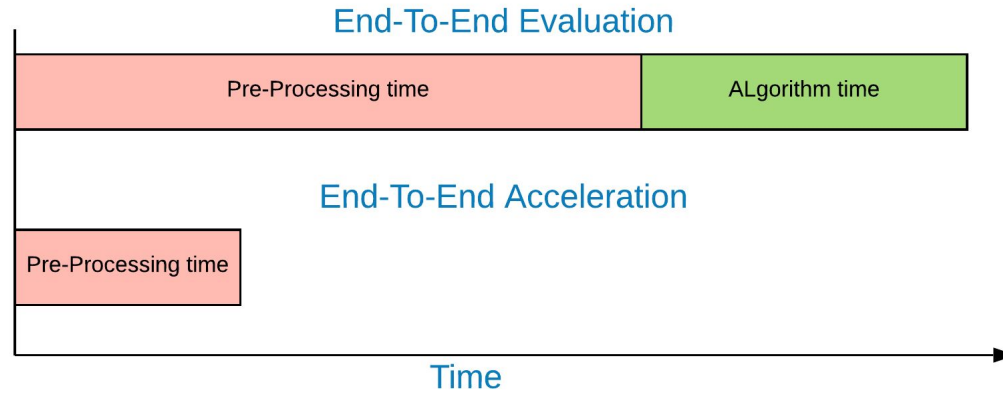
End to end acceleration



End-To End Time = Pre-processing + Algorithm time

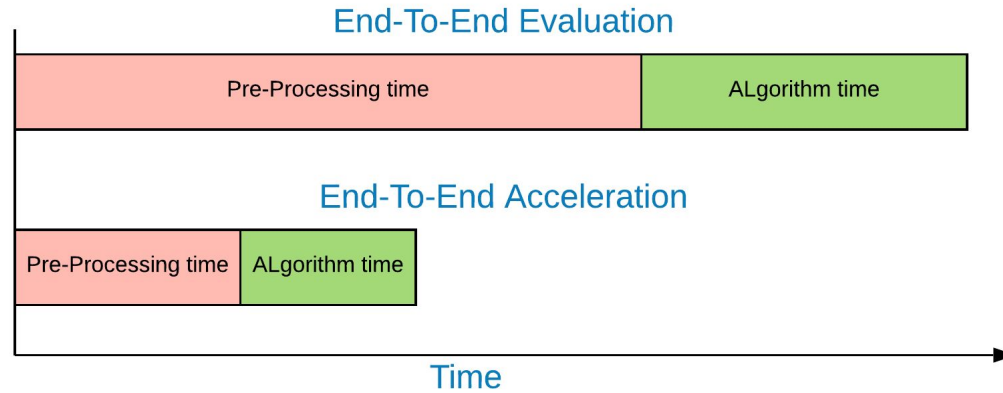


End to end acceleration



End-To End Time = Pre-processing + Algorithm time

End to end acceleration



End-To End Time = Pre-processing + Algorithm time



Survey

Semih Salihoglu recently led his students (with help by Tamer Özsu and me).

They conducted a survey of graph processing systems that focused on four major questions:

- What types of graph data do users have?
- What computations do users run on their graphs?
- What software do users use to perform their computations?
- What are the major challenges users face when processing their graph data?



Results

Scalability is unequivocally the most pressing challenge faced by the survey participants. The ability to process very large graphs efficiently appears to be the biggest limitation of existing software.

After scalability, participants indicated visualization as their second most pressing challenge.



Size of graph

One of the questions in the survey asked participants about the size of their graphs.

Of the 89 responses, 20 participants (8 researchers and 12 practitioners) indicated working with graphs containing more than one billion edges. Another 21 participants (8 researchers and 13 practitioners) indicated working with graphs containing between 100 million and one billion edges.

In terms of the number of vertices, 27 participants (10 researchers and 17 practitioners) indicated that their graphs contained more than 100 million vertices.



Size of the graph

A graph with 100 billion edges, even with a moderate amount of metadata, could comfortably fit in memory on a commodity server today.

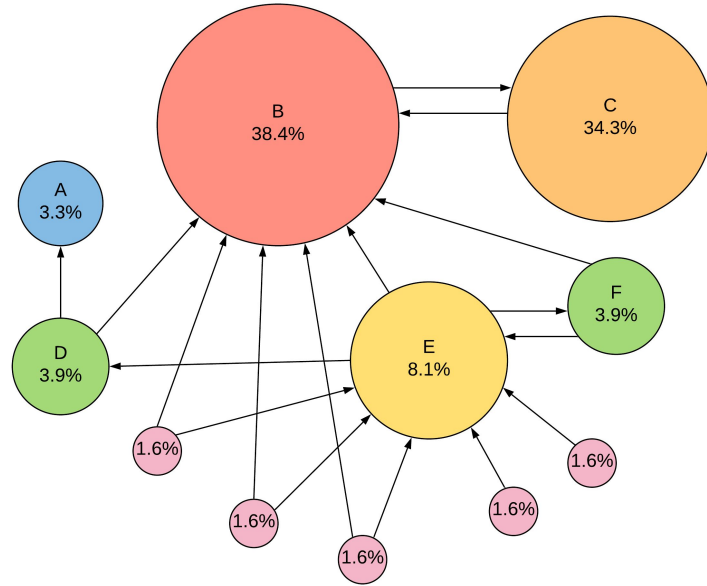
The critical question was not “How fast is the graph growing?” but “How fast is the graph growing compared to Moore’s Law.”



Challenges in graph processing

- Poor locality.
- Frequent cache misses
- Frequent inter-core communication
- Memory bandwidth saturation

PageRank



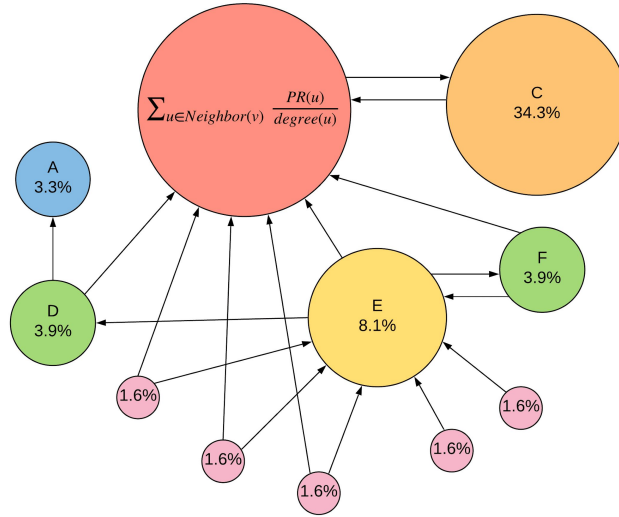


PageRank

$$PR(v) := \alpha \left(\frac{1}{N} \right) + (1 - \alpha) \sum_{u \in neighbors(v)} \frac{PR(u)}{degree(u)}$$

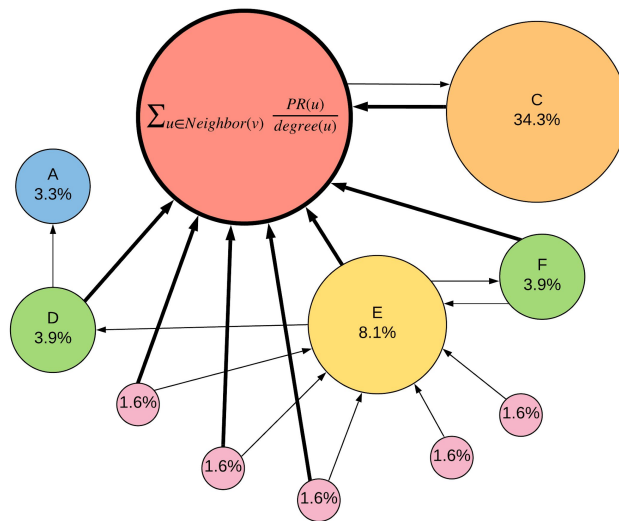
PageRank

$$PR(v) := \alpha \left(\frac{1}{N} \right) + (1 - \alpha) \sum_{u \in \text{Neighbor}(v)} \frac{PR(u)}{\text{degree}(u)}$$

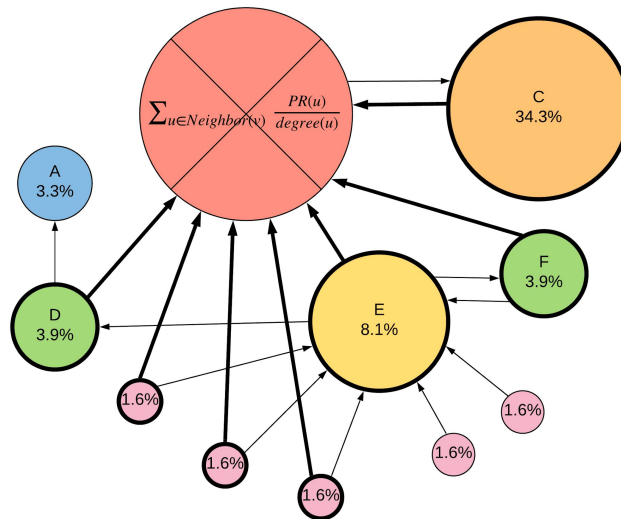


PageRank

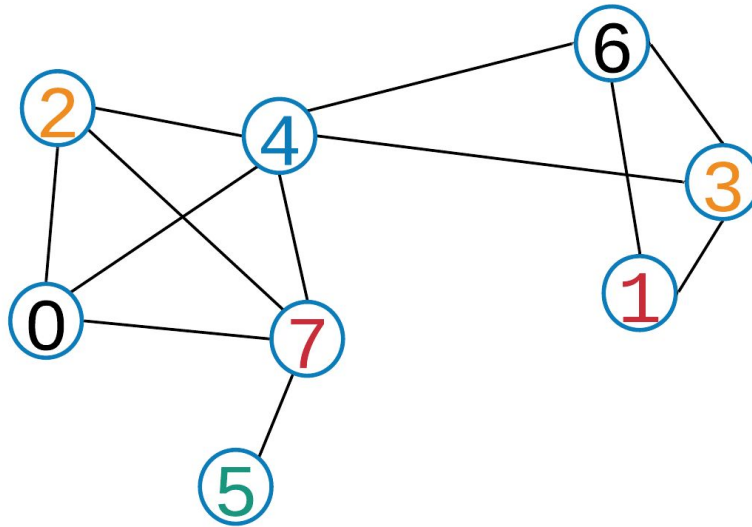
$$PR(v) := \alpha \left(\frac{1}{N} \right) + (1 - \alpha) \sum_{u \in \text{Neighbor}(v)} \frac{PR(u)}{\text{degree}(u)}$$



$$PR(v) := \alpha \left(\frac{1}{N} \right) + (1 - \alpha) \sum_{u \in Neighbor(v)} \frac{PR(u)}{degree(u)}$$

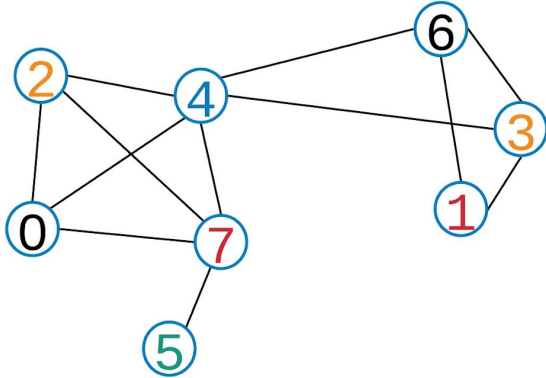


PageRank



PageRank

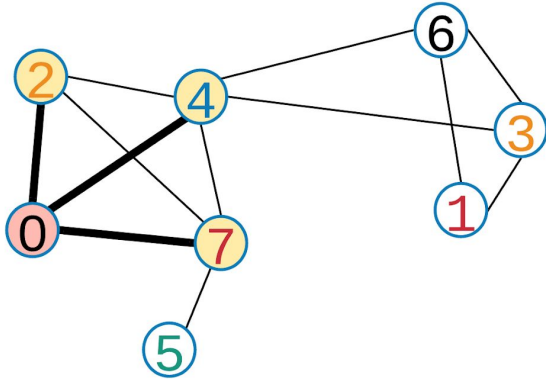
$$\sum_{u \in \text{Neighbor}(v)} \frac{PR(u)}{\text{degree}(u)}$$



PR	0	1	2	3	4	5	6	7

PageRank

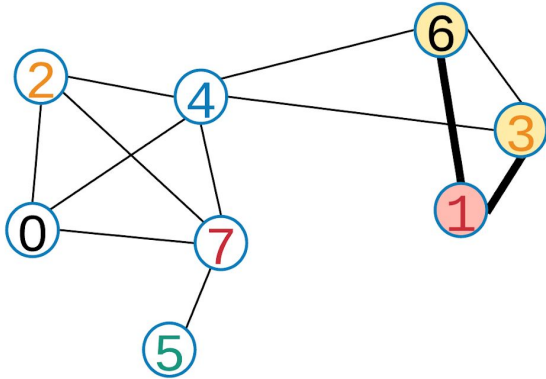
$$\sum_{u \in \text{Neighbor}(v)} \frac{PR(u)}{\text{degree}(u)}$$



PR	0	1	2	3	4	5	6	7
	■		■		■			■

PageRank

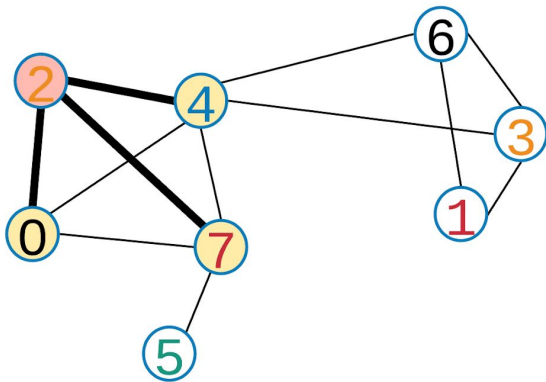
$$\sum_{u \in \text{Neighbor}(v)} \frac{PR(u)}{\text{degree}(u)}$$



PR	0	1	2	3	4	5	6	7

PageRank

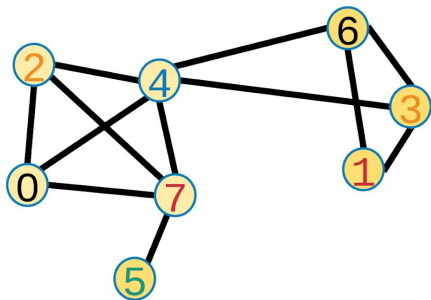
$$\sum_{u \in \text{Neighbor}(v)} \frac{PR(u)}{\text{degree}(u)}$$



PR	0	1	2	3	4	5	6	7
	■		■		■			■
		■		■			■	
	■		■		■			■

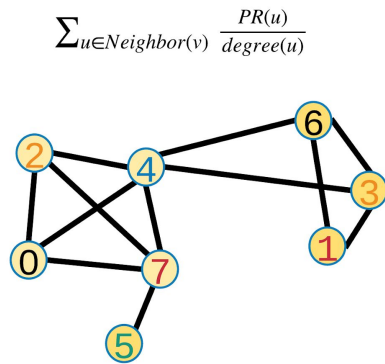
PageRank

$$\sum_{u \in \text{Neighbor}(v)} \frac{PR(u)}{\text{degree}(u)}$$



PR	0	1	2	3	4	5	6	7
	■	□	■	□	■	□	□	■
	□	■	□	■	□	□	■	□
	■	□	■	□	■	□	□	■
	□	■	□	■	■	□	■	□
	■	□	■	■	■	□	■	■
	□	□	□	□	□	■	□	■
	□	■	□	■	■	□	■	□
	■	□	■	□	■	■	□	■

PageRank



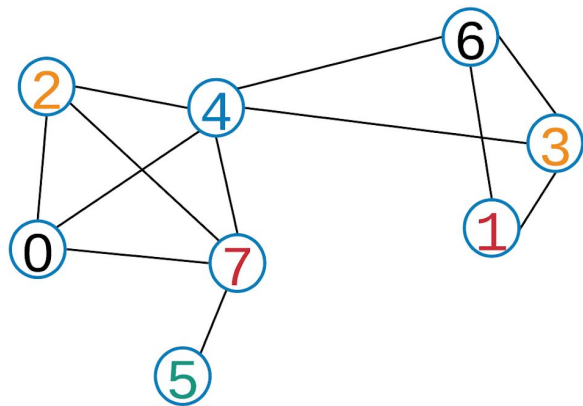
PR	0	1	2	3	4	5	6	7
	■		■		■			■
		■			■			■
	■		■		■		■	
			■	■	■		■	
			■	■	■		■	
			■	■	■		■	
	■		■		■		■	

Poor spatial locality

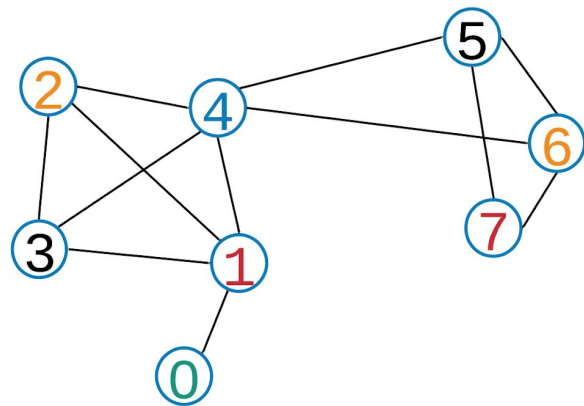
Poor temporal locality

PageRank

Random Ordering



High-locality



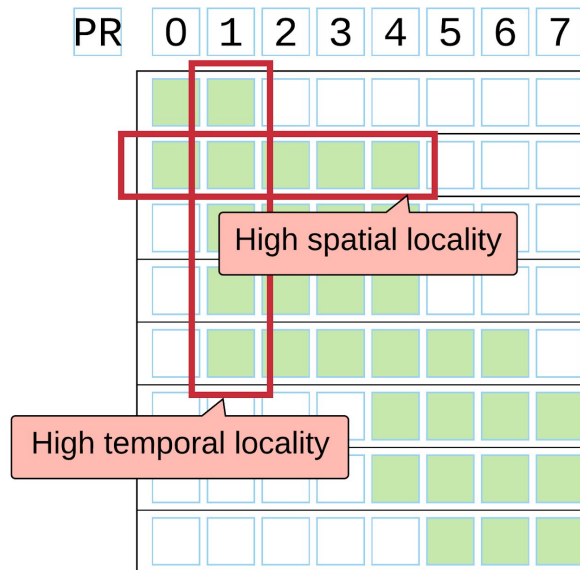
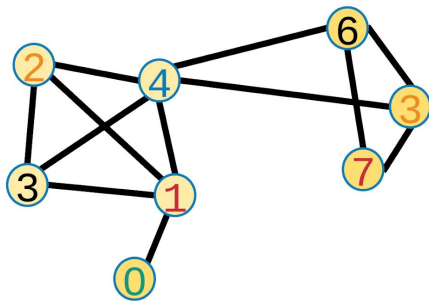
PageRank

$$\sum_{u \in Neighbor(v)} \frac{PR(u)}{degree(u)}$$

PR	0	1	2	3	4	5	6	7
	■	■						
	■	■	■	■	■			
		■	■	■	■			
		■	■	■	■			
		■	■	■	■	■	■	
					■	■	■	■
					■	■	■	■
						■	■	■

PageRank

$$\sum_{u \in \text{Neighbor}(v)} \frac{PR(u)}{\text{degree}(u)}$$



CAPI Challenges

