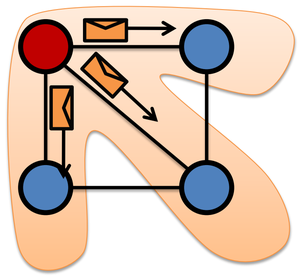
GraphLab2: Distributed Graph-Parallel Computation on Natural Graphs

Joseph Gonzalez, Yucheng Low, Haijie Gu, Danny Bickson, Carlos Guestrin

Graph Parallel Abstraction

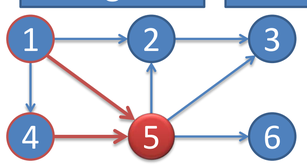
- **Vertex-Program** associated with each vertex
- **Graph** constrains the interaction along edges
  - **Pregel:** Programs interact through messages
  - **GraphLab 1:** Programs can read each-others state



Example: ***PageRank***

- Iterate: $R[i] = 0.15 + 0.85 \sum_{(j,i)\in E} \frac{R[j]}{L[j]}$

Rank of Page *i* | Random Reset Probability | Sum over in-links | Number of out-links form *j*

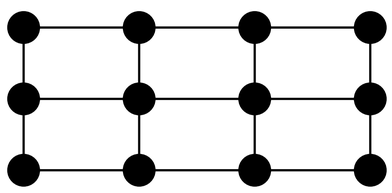


$$R[5] = 0.15 + 0.85\left(\frac{1}{3}R[1] + \frac{1}{1}R[4]\right)$$

Natural Graphs

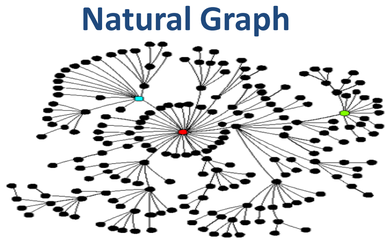
Graphs Encode Relationships between Entities

Assumed Structure



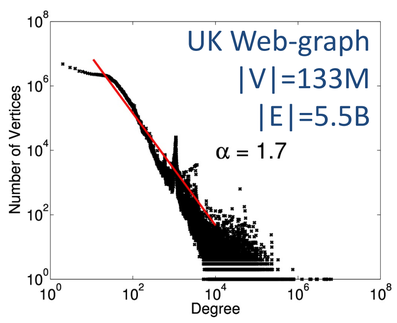
- ***Small*** neighborhoods
- ***Similar*** degree vertices
- Easy to ***partition***

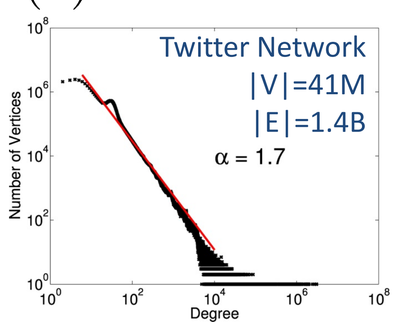
Natural Graph

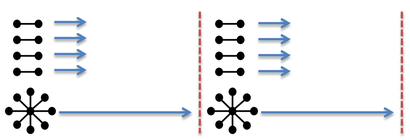


- ***Large*** Neighborhoods
- ***Power-Law Degree***
- Difficult to partition

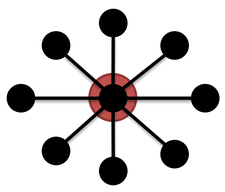
- **Power-Law Graphs**
  - *Most vertices have relatively few neighbors while a few vertices have many neighbors*
  - Probability of having degree *d*: $P(d) \propto d^{-\alpha}$



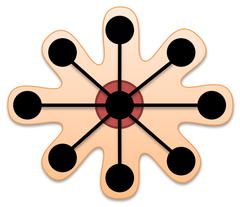




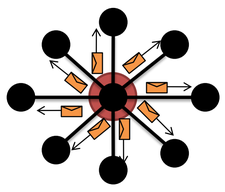
Synchronous execution is prone to stragglers.



Edge information overwhelms single machine



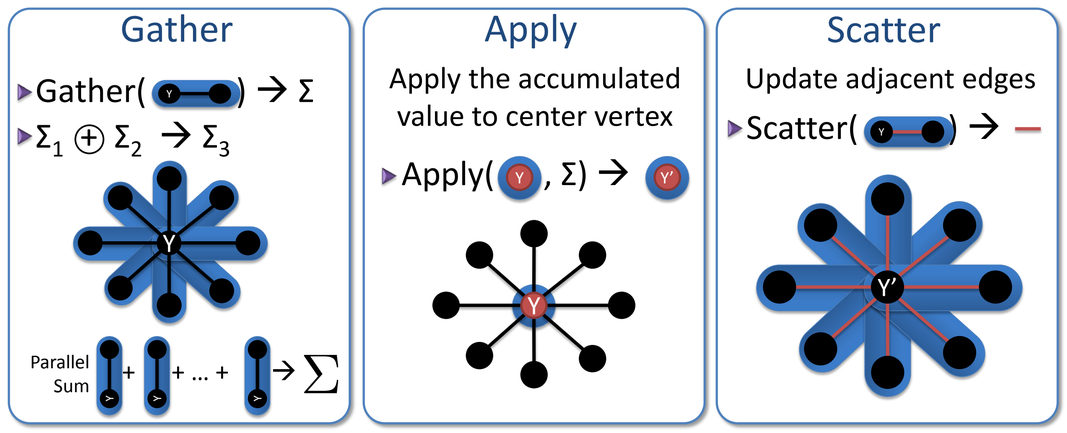
Touches a large fraction of graph (GraphLab 1)



Produces many messages (Pregel)

PowerGraph (OSDI '12)

- **Factorized** Vertex-Programs:



- Distribute vertex-program over several machines and move computation to data

PowerGraph_PageRank(v)

Gather(u→v) : return R[u] / #outlinks[u]

sum(a, b) : return a + b;

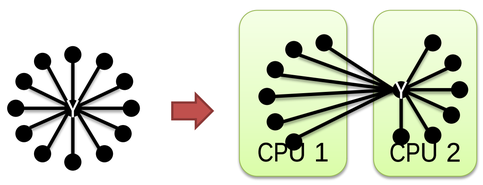
Apply(v, Σ_v) : R[v] = 0.15 + 0.85 Σ_v

Scatter(v → w) :

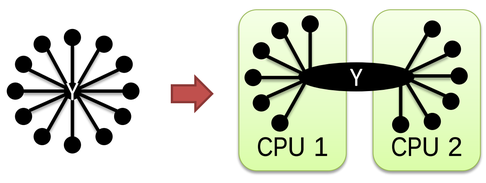
if (R[v] changes by ε) then *activate*(w)

Vertex Seperators

Cut vertices instead of edges!

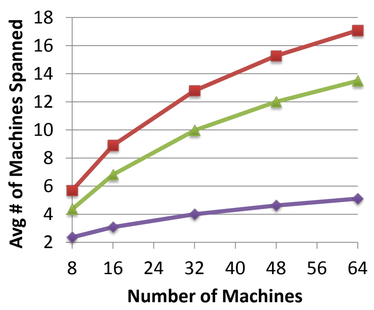


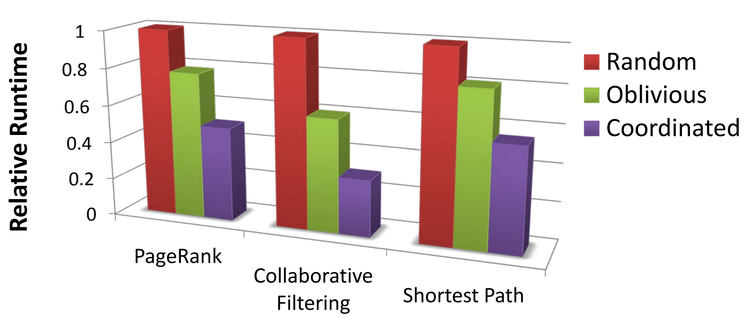
Must synchronize **many** edges



Must synchronize a **single** vertex

Three Streaming Partitioning Strategies

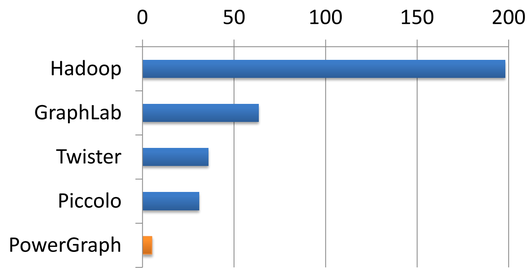




Results

PageRank (per Iteration):

40M Webpages, 1.4B Links



Triangle Counting:

Twitter Graph:
40 Million Vertices
1.4 Billion Edges
34.8 Billion Triangles

Hadoop	**1536 Machines 423 Minutes**
PowerGraph	**64 Machines, 1024 Cores 1 min 30 seconds**

Latent Dirichlet Allocation

Wikipedia:
8M documents
315M tokens

Smola et al	**150M Tokens /sec 100 nodes x 8 cores**
PowerGraph	**110M Tokens /sec 64 nodes x 16 cores**

Better

Available Now!

GraphLab

http://**GraphLab**.org

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University

Georgia Tech

intel

PRINCETON UNIVERSITY

UC Berkeley