GraphLab

Unleash Data Science

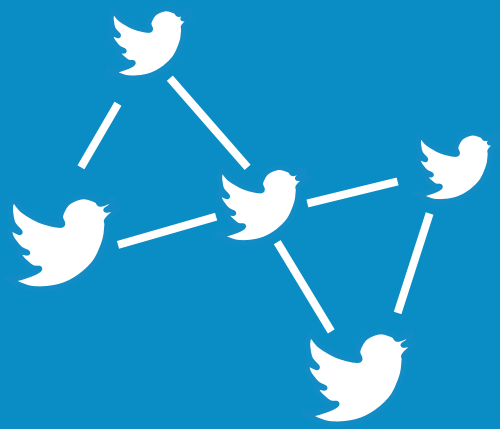
Danny Bickson

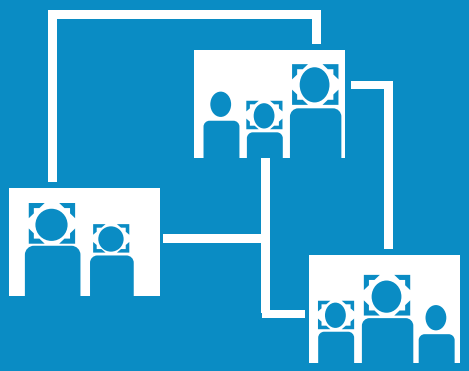
Co-Founder

GraphLab Project History

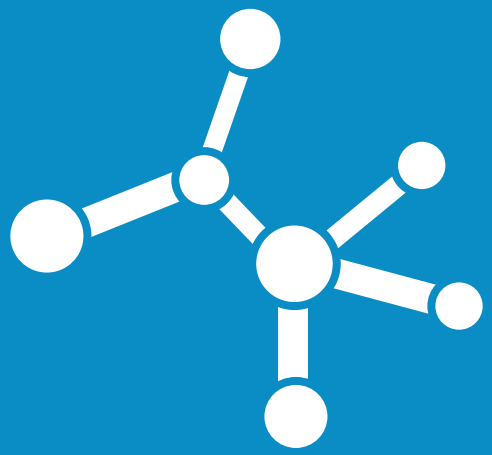
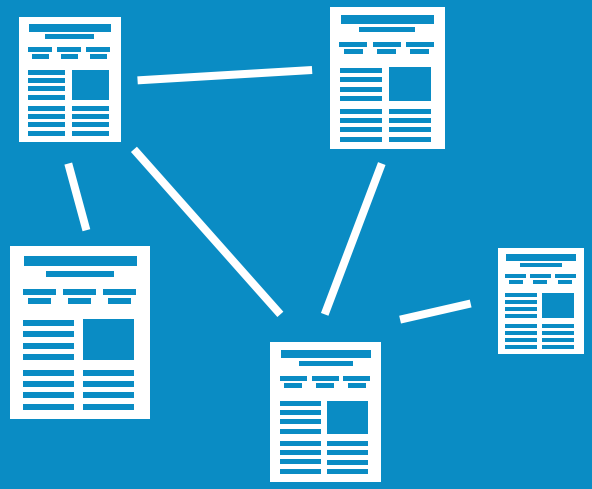
GraphLab (2009) → GraphChi (2011) → GraphLab Create (2014)

GraphLab Open Source (2009)





Graphs are Everywhere


Graphs are Essential to Data Mining and Machine Learning

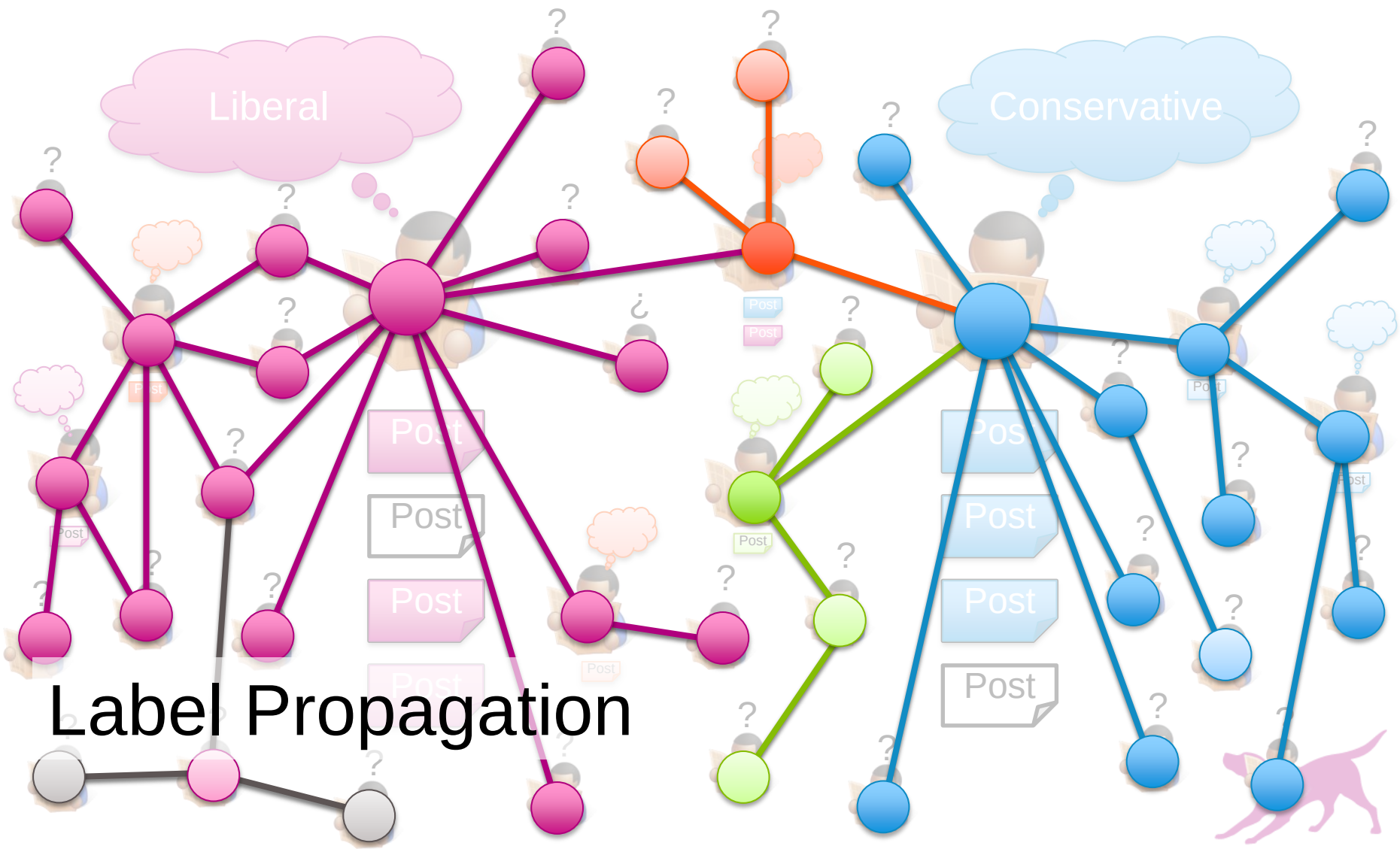
Identify influential information

Reason about latent properties

Model complex data dependencies

Examples of Graphs in Machine Learning

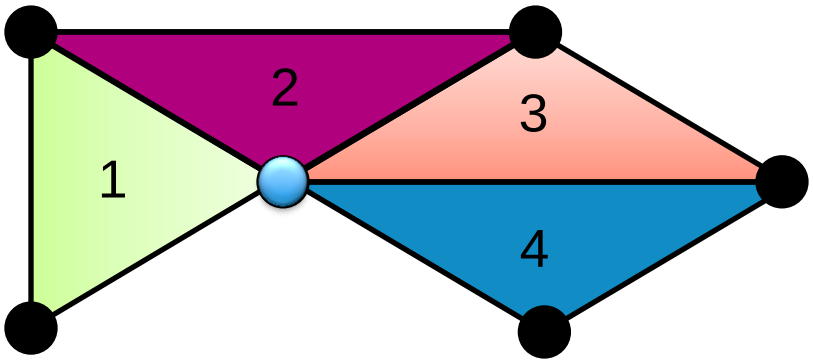
Predicting User Behavior



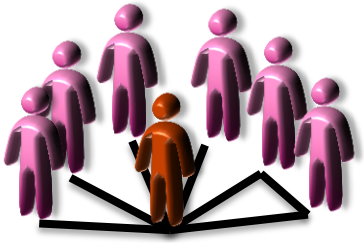
Label Propagation

Finding Communities

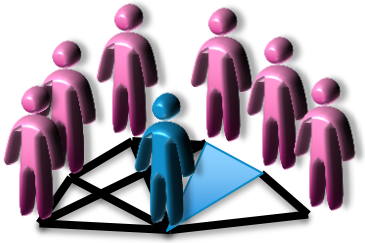
Count triangles passing through each vertex:



Measures “cohesiveness” of local community

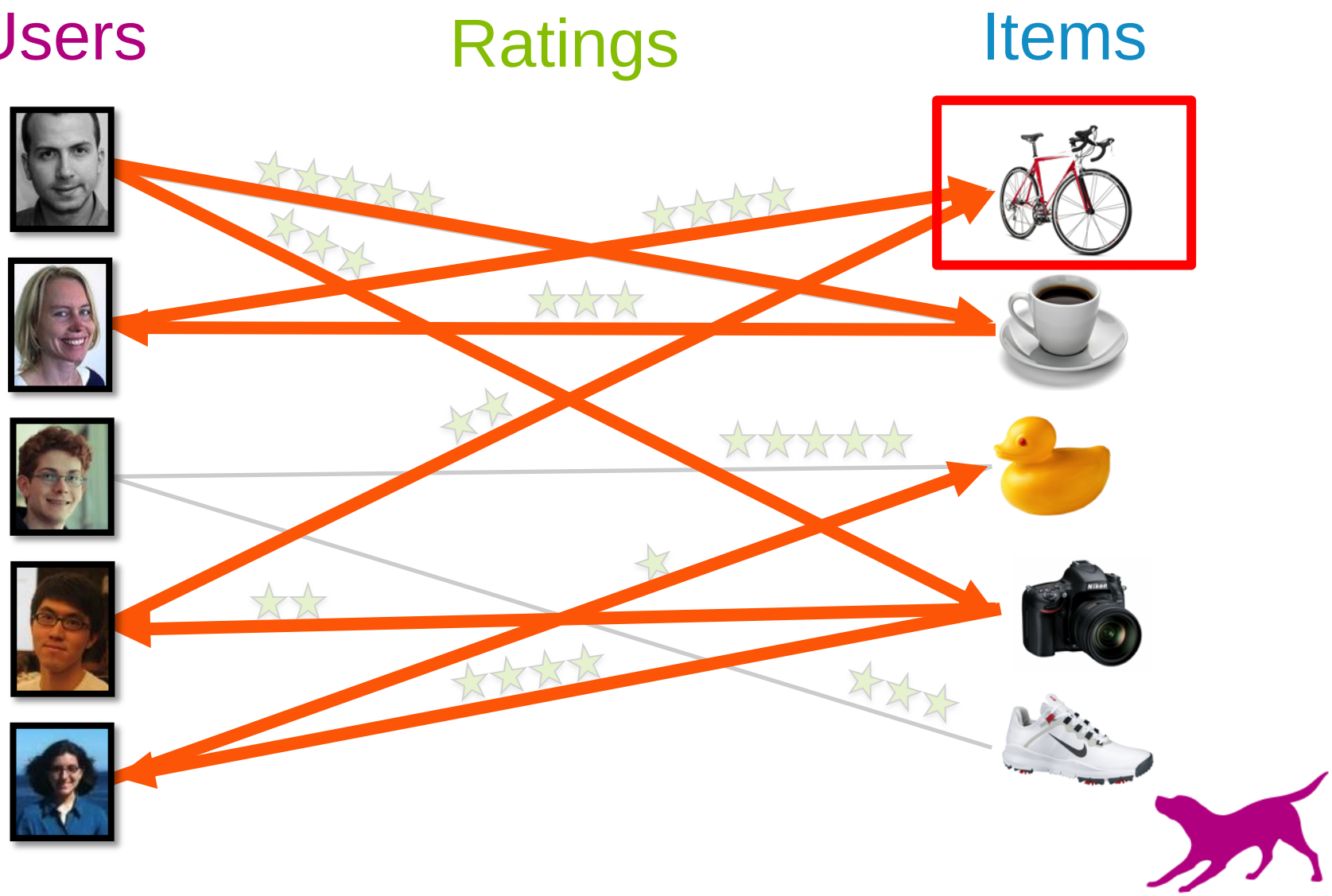


Fewer Triangles
Weaker Community



More Triangles
Stronger Community

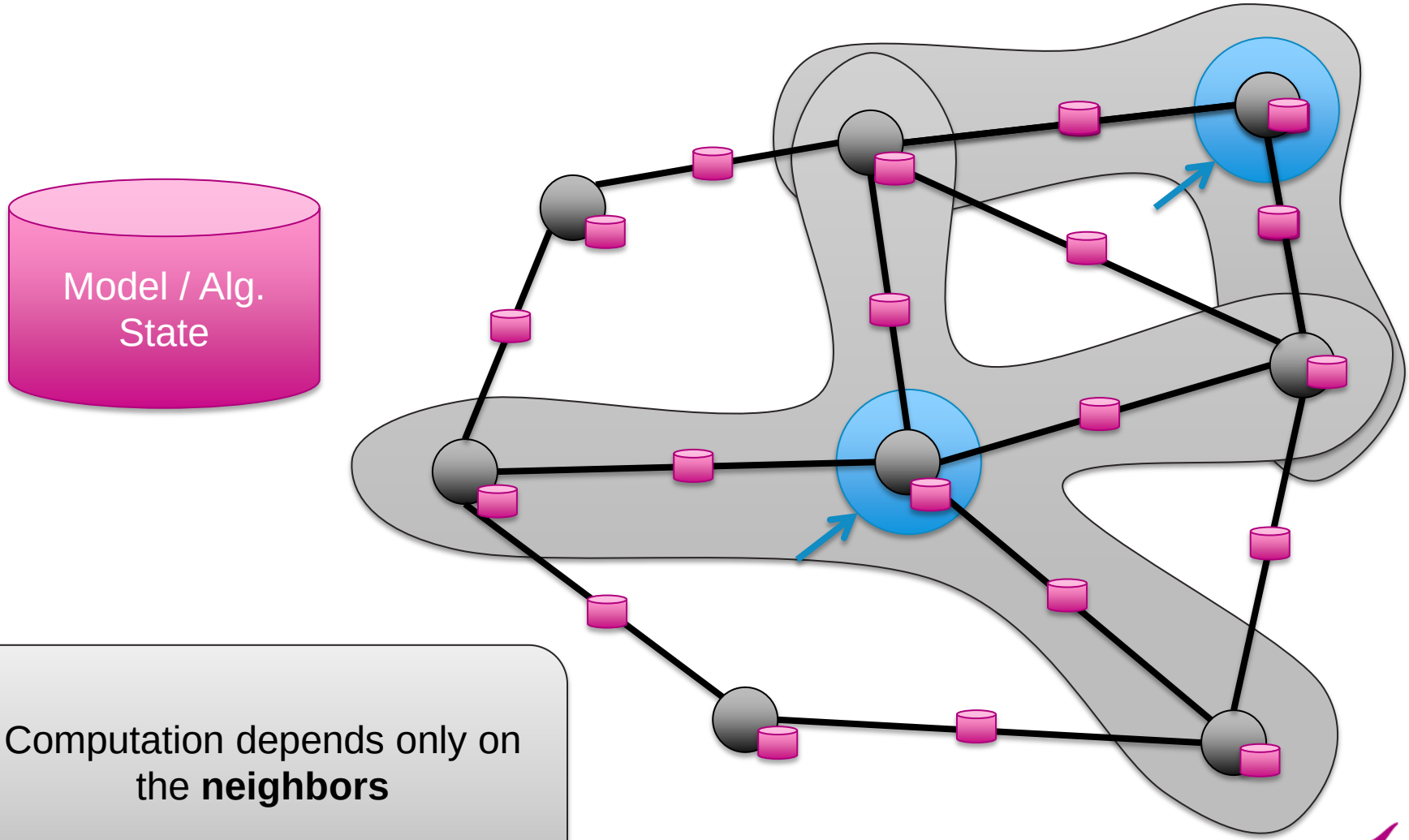
Recommending Products



Many More Applications

- Collaborative Filtering
  - Alternating Least Squares
  - Stochastic Gradient Descent
  - Tensor Factorization
- Structured Prediction
  - Loopy Belief Propagation
  - Max-Product Linear Programs
  - Gibbs Sampling
- Semi-supervised ML
  - Graph SSL
  - CoEM
- Community Detection
  - Triangle-Counting
  - K-core Decomposition
  - K-Truss
- Graph Analytics
  - PageRank
  - Personalized PageRank
  - Shortest Path
  - Graph Coloring
- Classification
  - Neural Networks

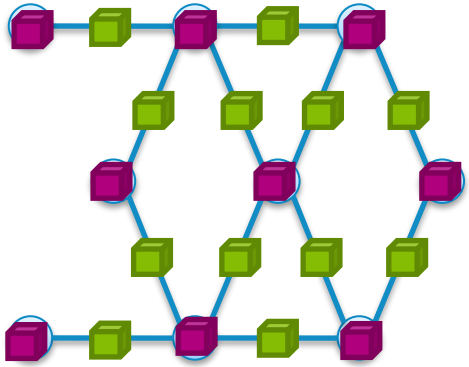
The Graph-Parallel Pattern



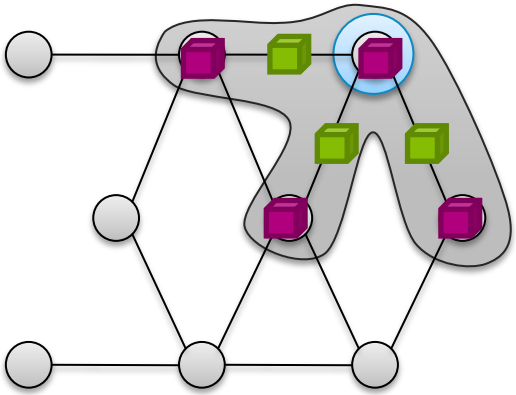
Computation depends only on the **neighbors**

The GraphLab Framework

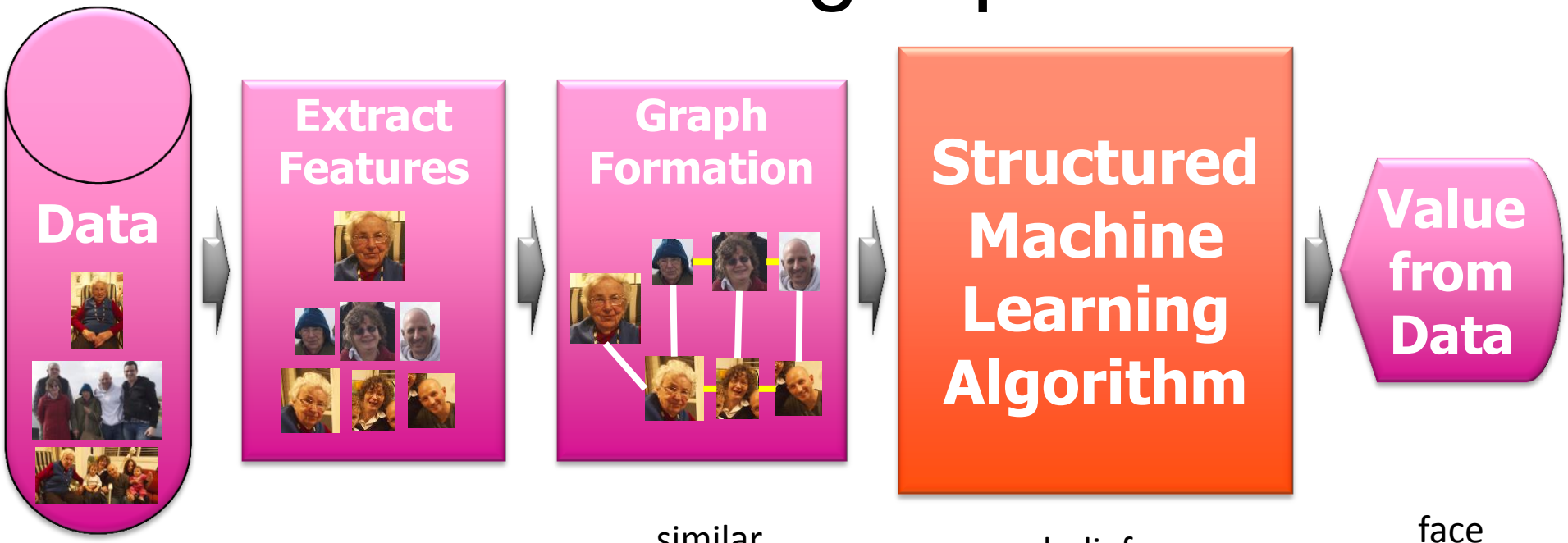
Data Model
Property Graph



Computation
Vertex Programs

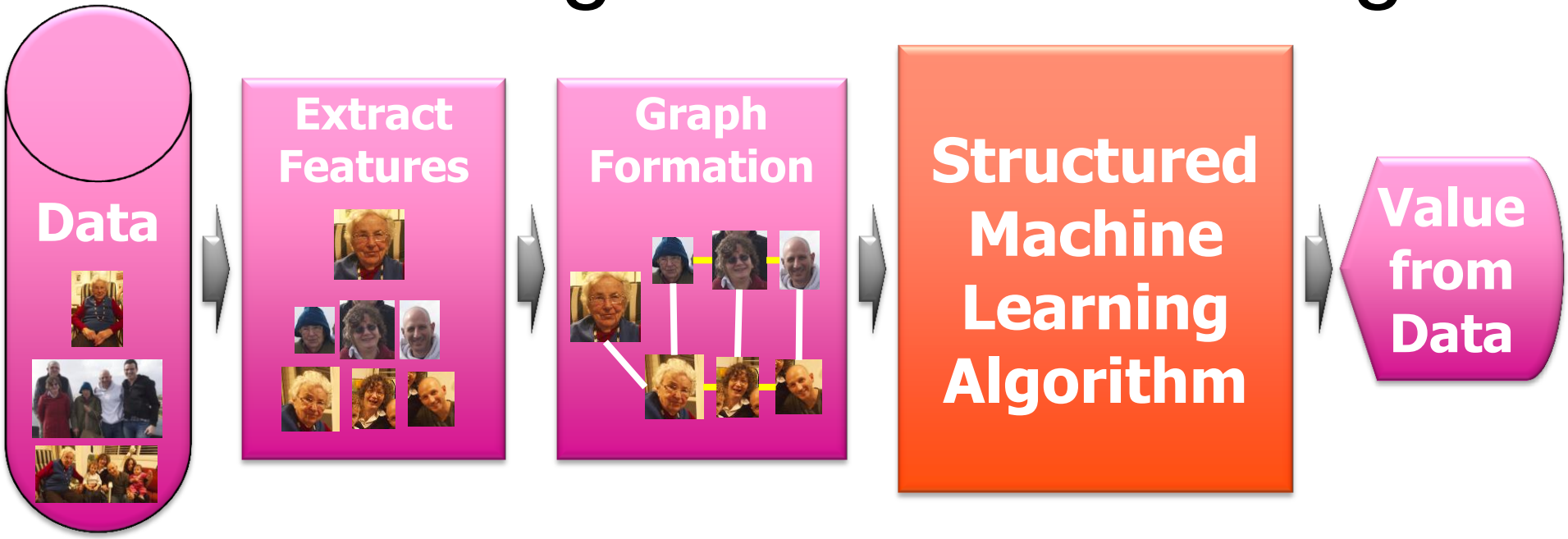


Machine Learning Pipeline



images	faces	similar faces	belief propagation	face labels
docs	important words	shared words	LDA	doc topics
movie ratings	side info	rated movies	collaborative filtering	movie recommend.

Parallelizing Machine Learning



Graph Ingress

mostly data-parallel

Graph-Structured Computation

graph-parallel

ML Tasks Beyond Data-Parallelism

Data-Parallel → Graph-Parallel

Map Reduce

Feature Extraction

Cross Validation

Computing Sufficient Statistics

Graphical Models
Gibbs Sampling
Belief Propagation
Variational Opt.

Semi-Supervised Learning
Label Propagation
CoEM

Collaborative Filtering
Tensor Factorization

Graph Analysis
PageRank
Triangle Counting

Example of a Graph-Parallel Algorithm

Flashback to 1998

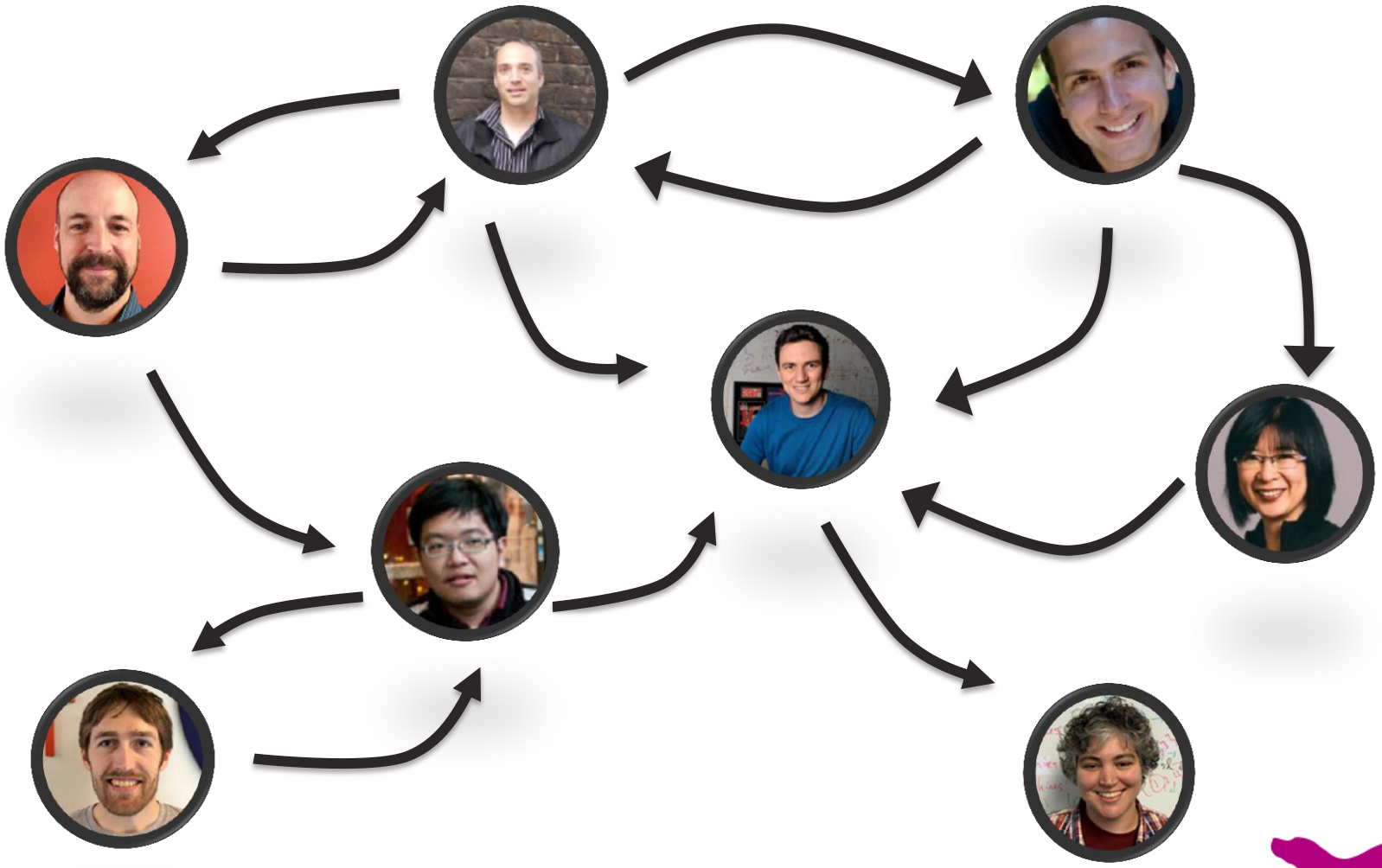
altavista™
SEARCH SOFTWARE

Why?

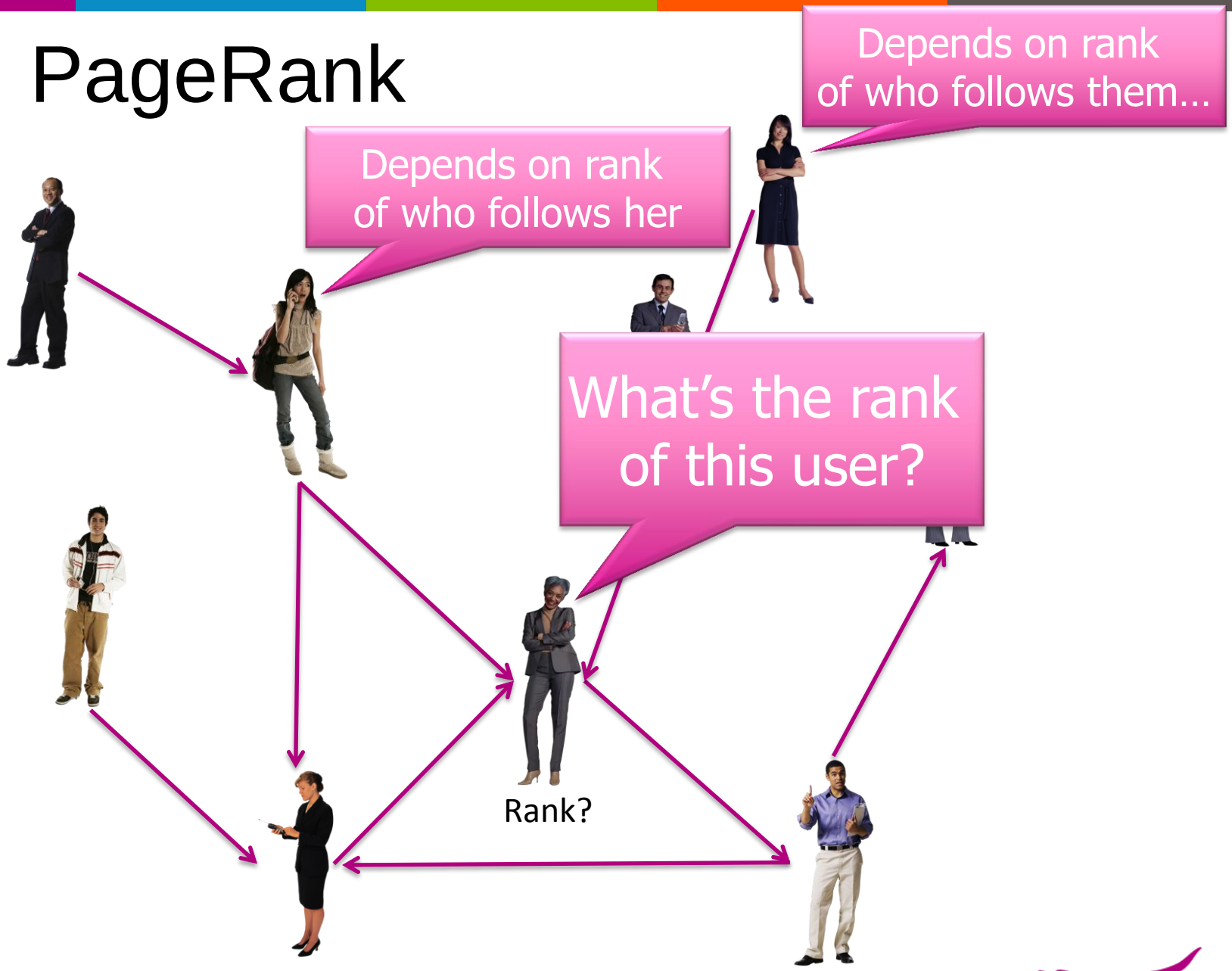
Google!
BETA

First Google advantage:
a Graph Algorithm & System to Support it!

PageRank: Identifying Leaders

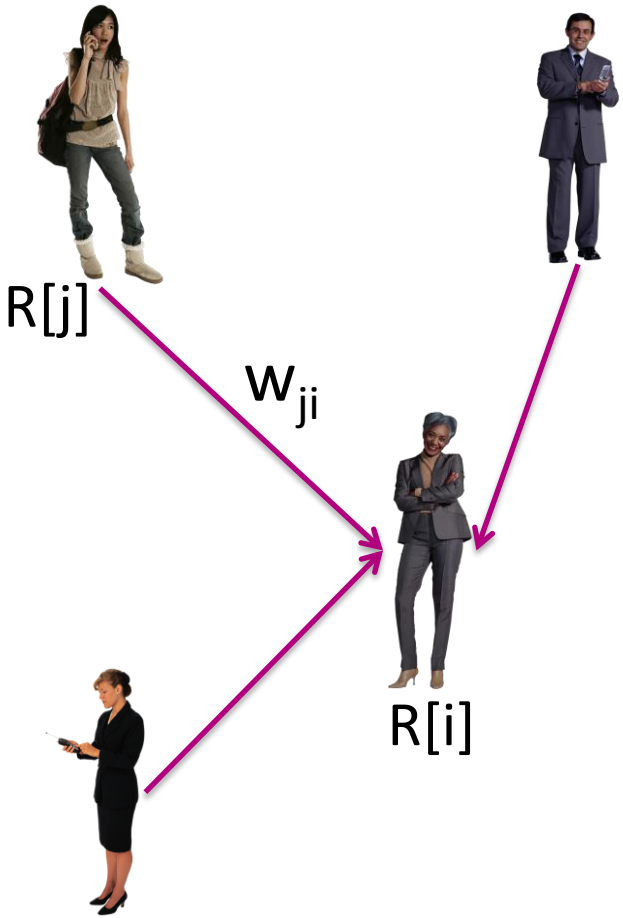


PageRank



Loops in graph → Must iterate!

PageRank Iteration



Iterate until convergence:

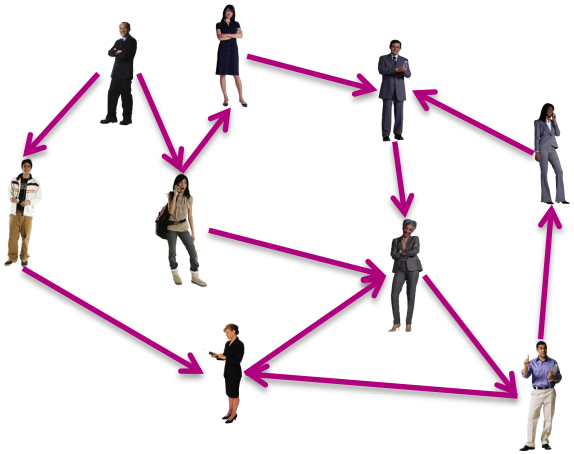
"My rank is weighted average of my friends' ranks"

$$R[i] = \alpha + (1-\alpha) \sum_{(j,i)\in E} w_{ji} R[j]$$

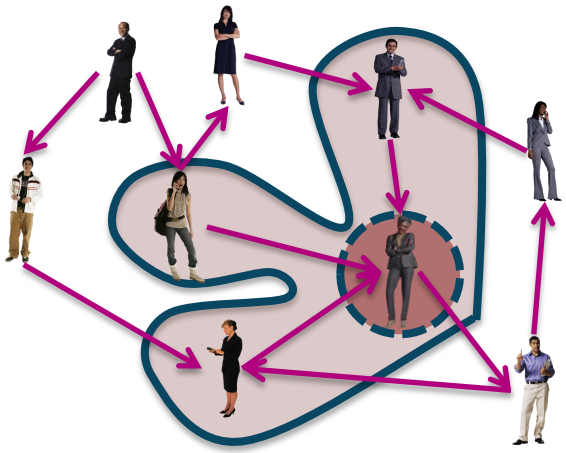
- α is the random reset probability
- w_{ji} is the prob. transitioning (similarity) from j to i

Properties of Graph Parallel Algorithms

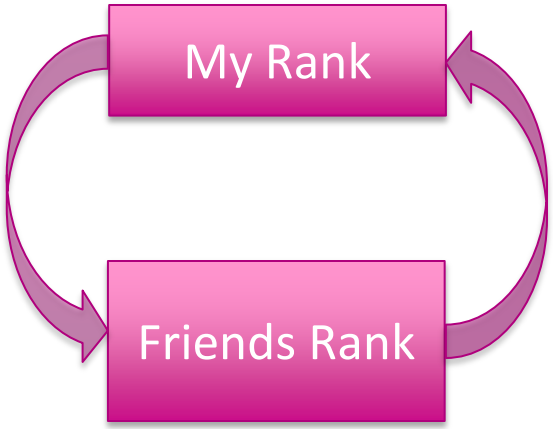
Dependency Graph



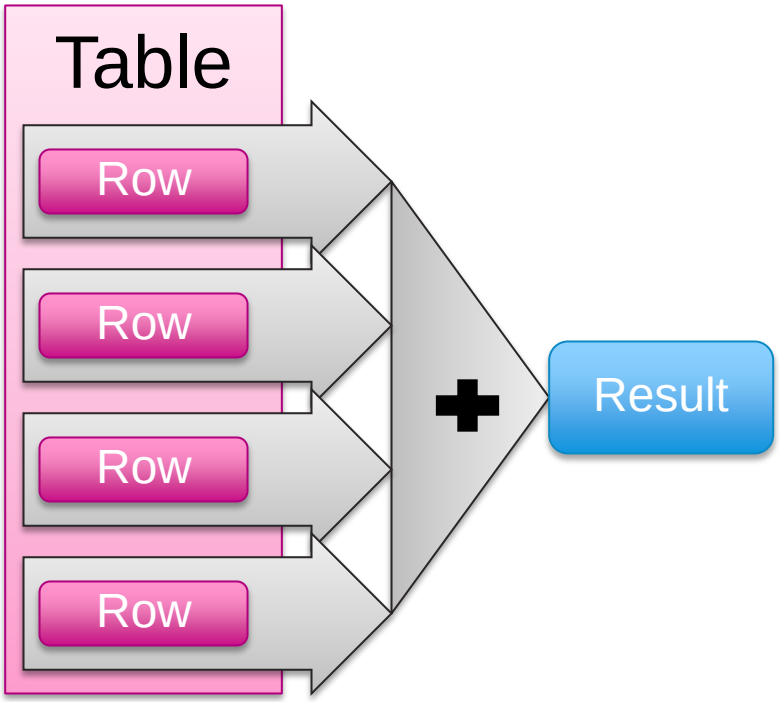
Local Updates

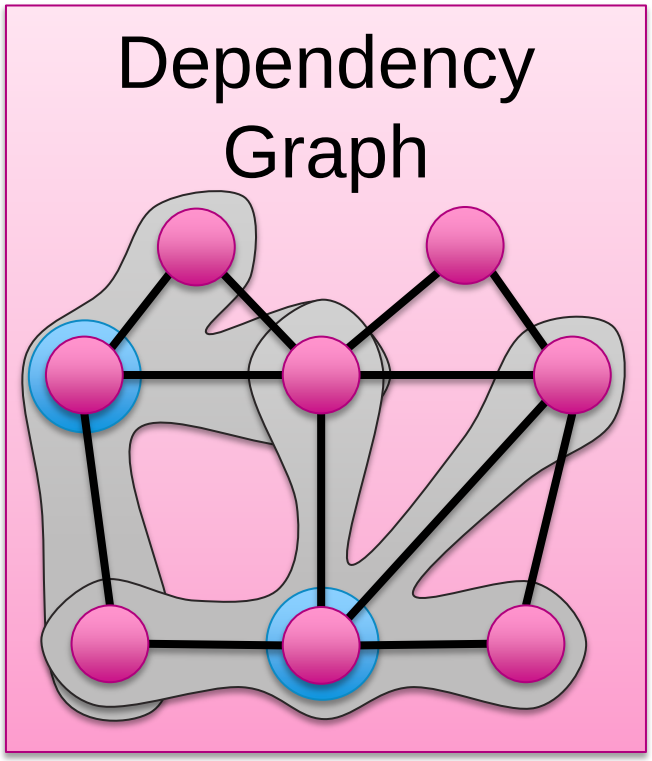


Iterative Computation



Data-Parallel vs Graph Parallel





hadoop MapReduce

GraphLab

Addressing Graph-Parallel ML

Data-Parallel ⟷ Graph-Parallel

Map Reduce

Feature Extraction

Cross Validation

Computing Sufficient Statistics

Graph-Parallel Abstraction

Graphical Models
Gibbs Sampling
Belief Propagation
Variational Opt.

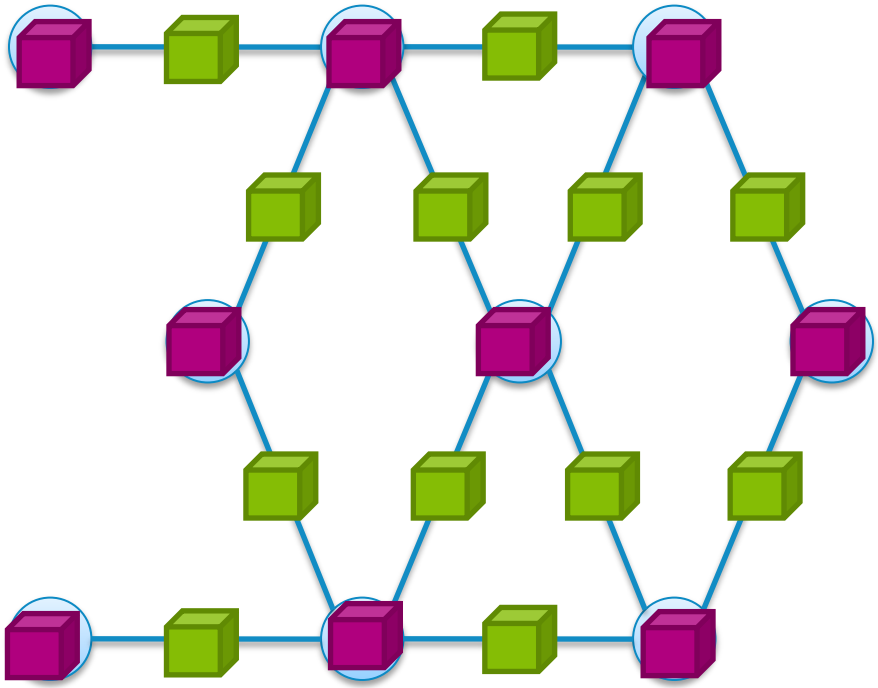
Semi-Supervised Learning
Label Propagation
CoEM

Collaborative Filtering
Tensor Factorization

Data-Mining
PageRank
Triangle Counting

Data Graph

Data associated with vertices and edges



Graph:

- Social Network

Vertex Data:

- User profile text
- Current interests estimates

Edge Data:

- Similarity weights

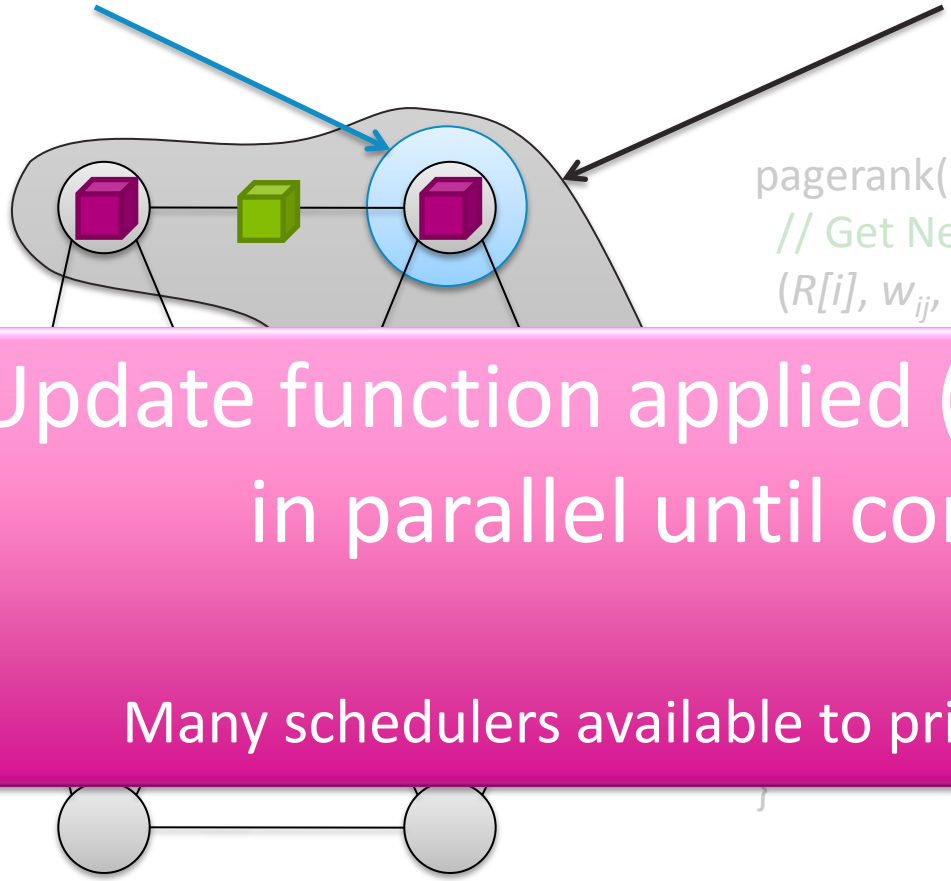
How do we *program* graph computation?

“Think like a Vertex.”

-Malewicz et al. [SIGMOD’10]

Update Functions

User-defined program: applied to **vertex** transforms data in **scope** of vertex



```
pagerank(i, scope){
  // Get Neighborhood data
  (R[i], w_ij, R[j]) ←scope;
```

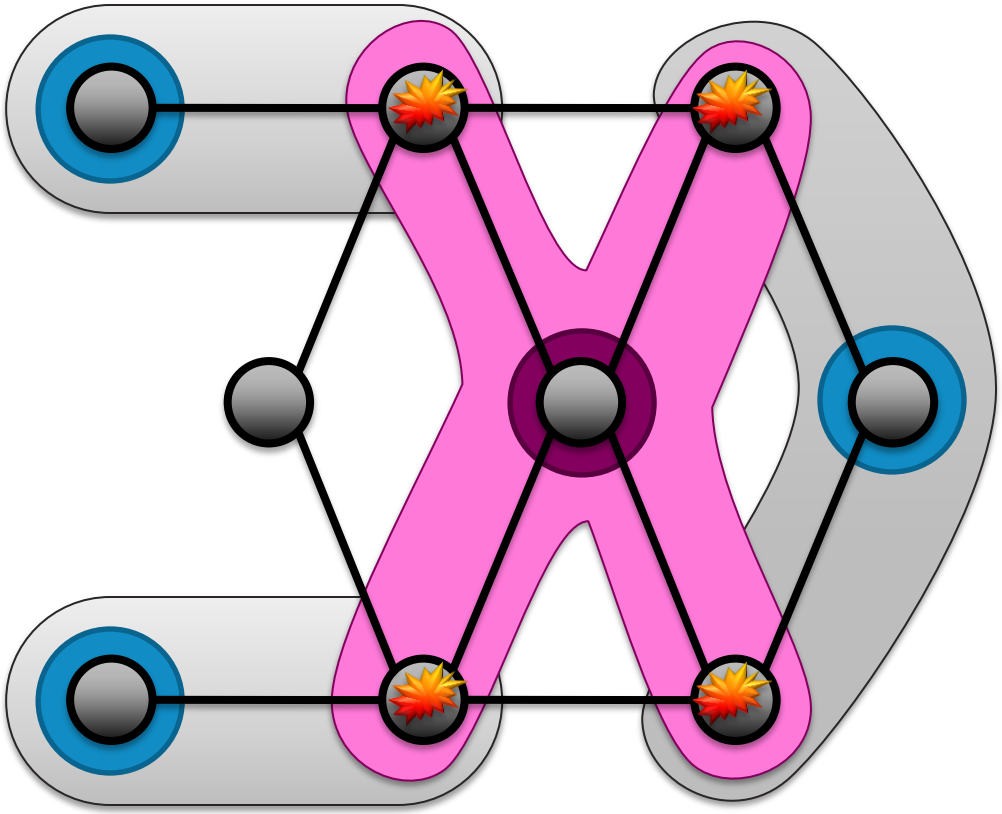
Update function applied (asynchronously) in parallel until convergence

Many schedulers available to prioritize computation

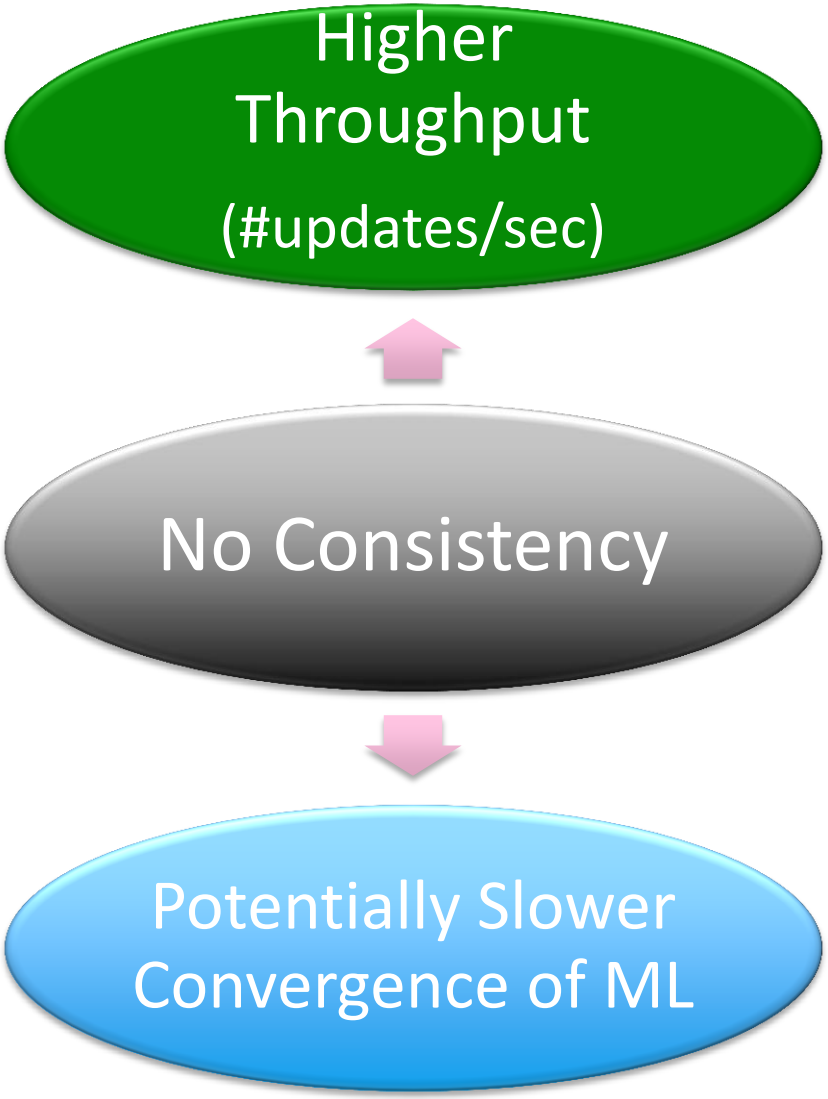
Dynamic computation

Ensuring Race-Free Code

How much can computation **overlap**?



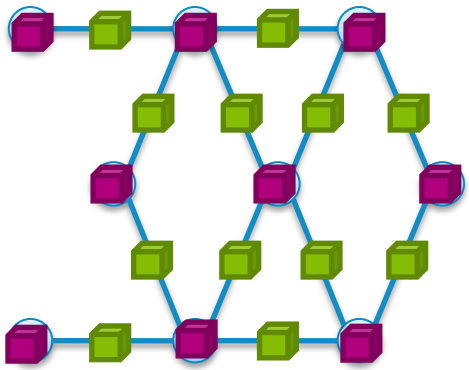
Need for Consistency?



The GraphLab Framework

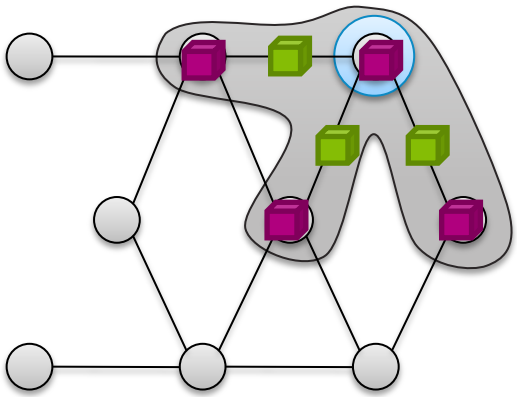
Graph Based

Data Representation

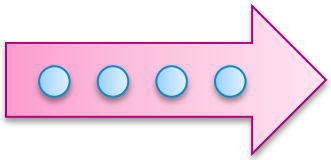


Update Functions

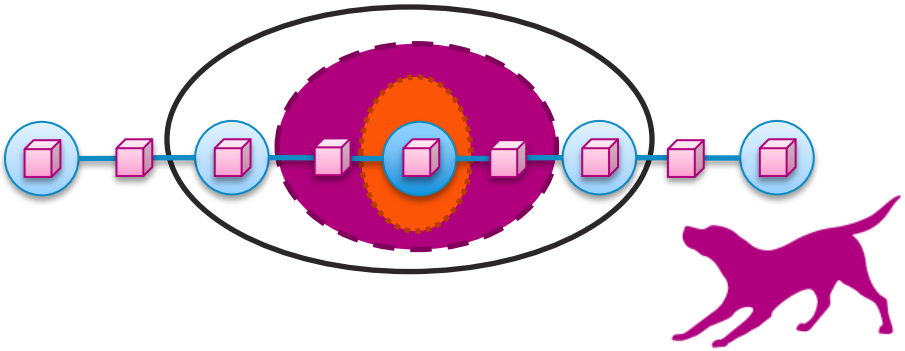
User Computation



Scheduler



Consistency Model



Never Ending Learner Project (CoEM)

Hadoop	95 Cores	7.5 hrs
Distributed GraphLab	**32 EC2 machines**	**80 secs**

0.3% of Hadoop time

2 orders of mag faster →
2 orders of mag cheaper

Thus far…

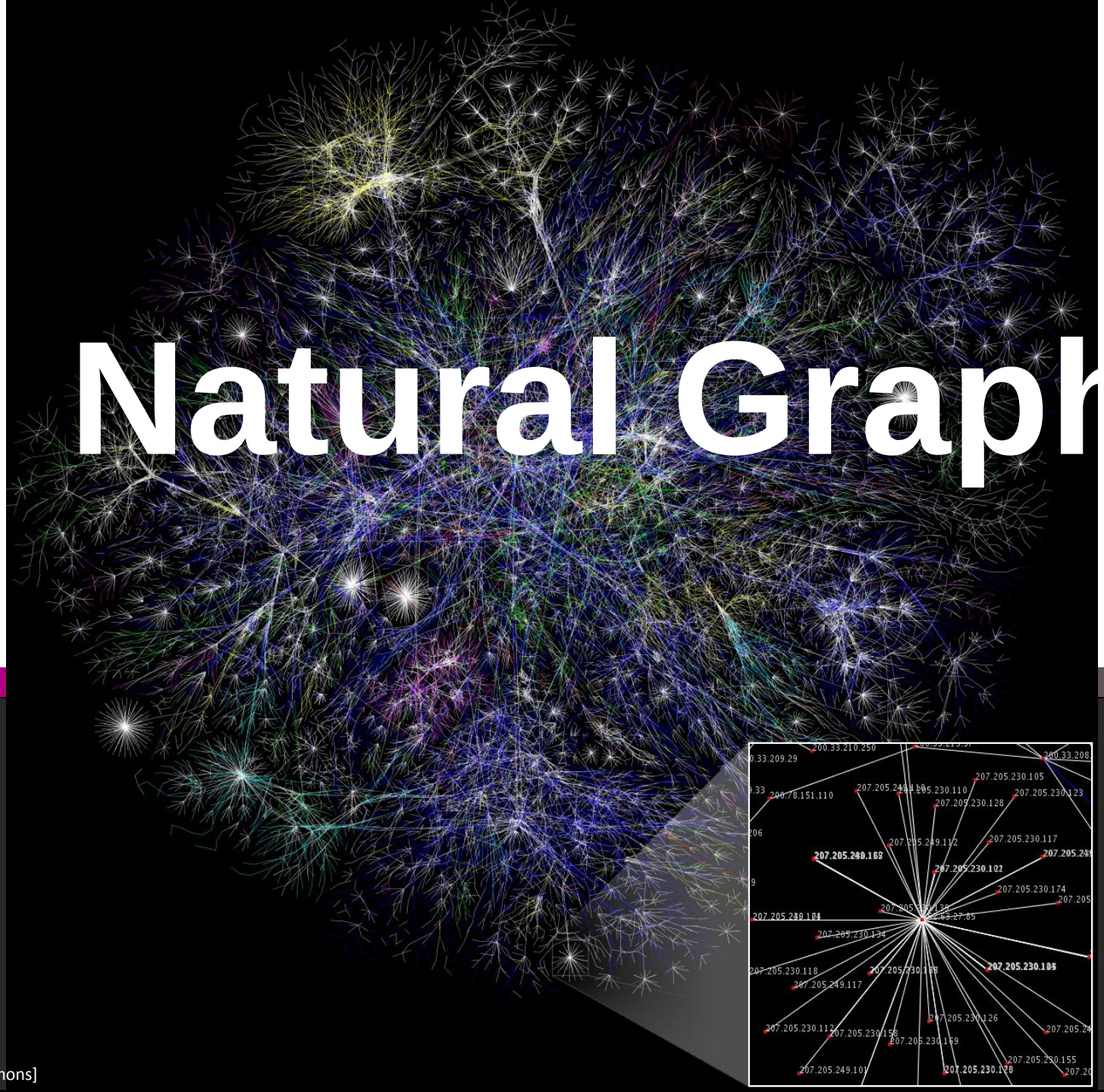
GraphLab 1 provided exciting scaling performance

But…

We couldn’t scale up to Altavista Webgraph 2002

1.4B vertices, 6.7B edges

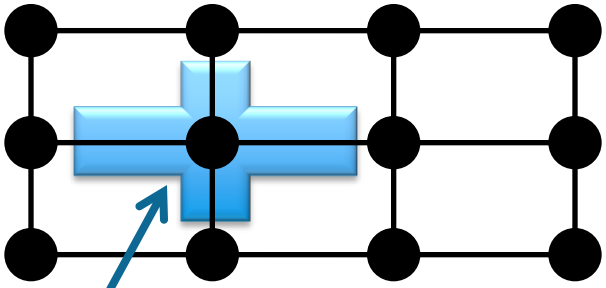
Natural Graph



[Image from WikiCommons]

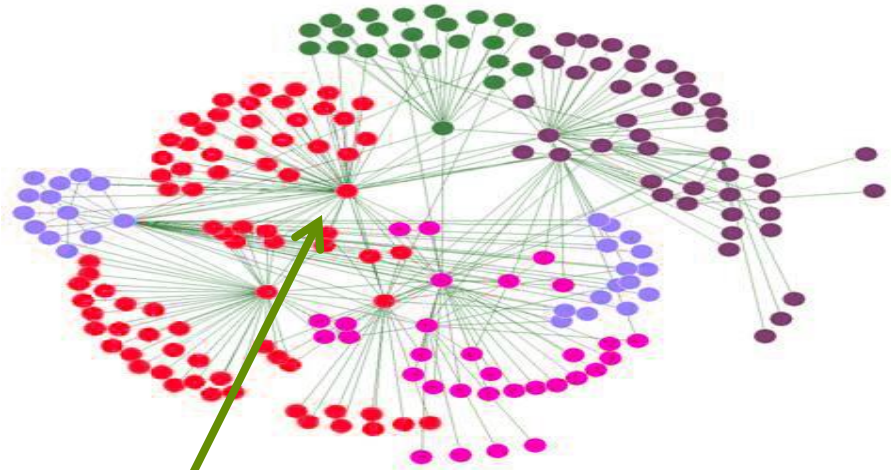
Achilles Heel: Idealized Graph Assumption

Assumed…



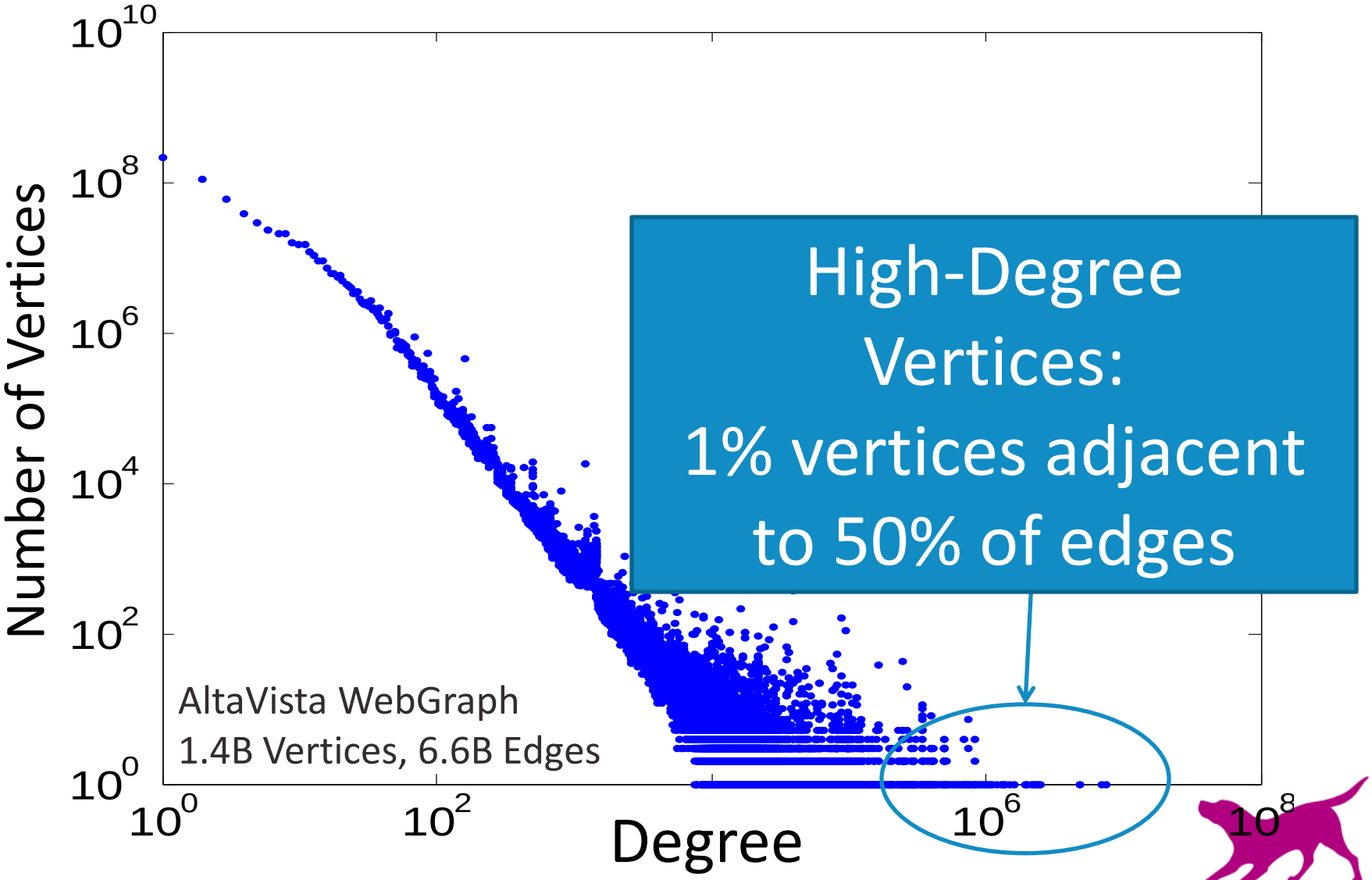
Small degree ➔
Easy to partition

But, Natural Graphs…



Many high degree vertices
(power-law degree distribution)
➔
Very hard to partition

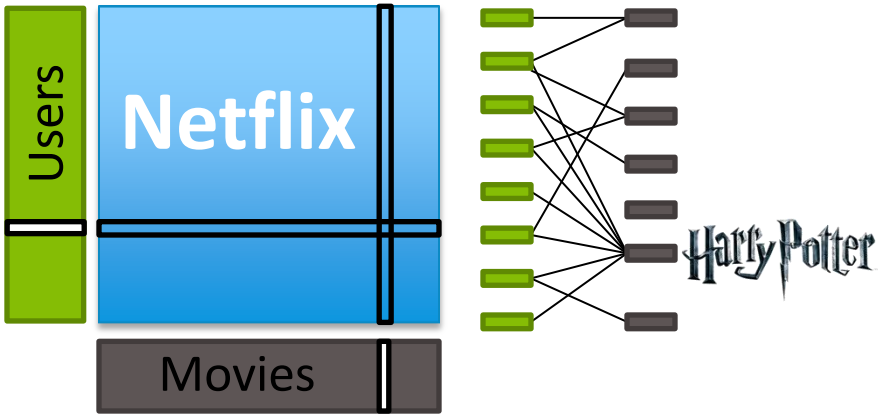
Power-Law Degree Distribution



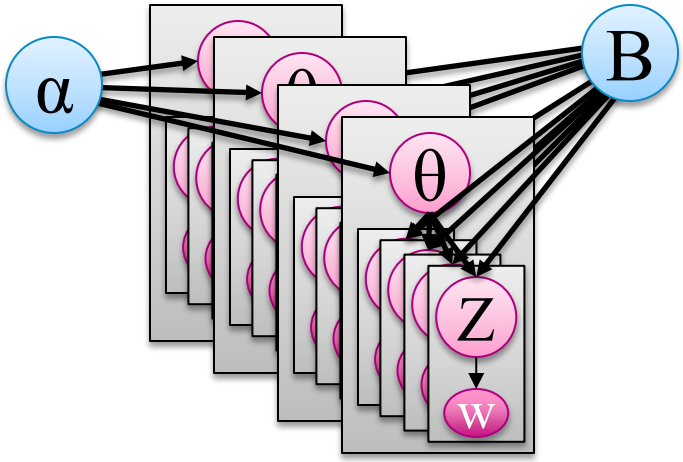
High Degree Vertices are Common

"Social" People

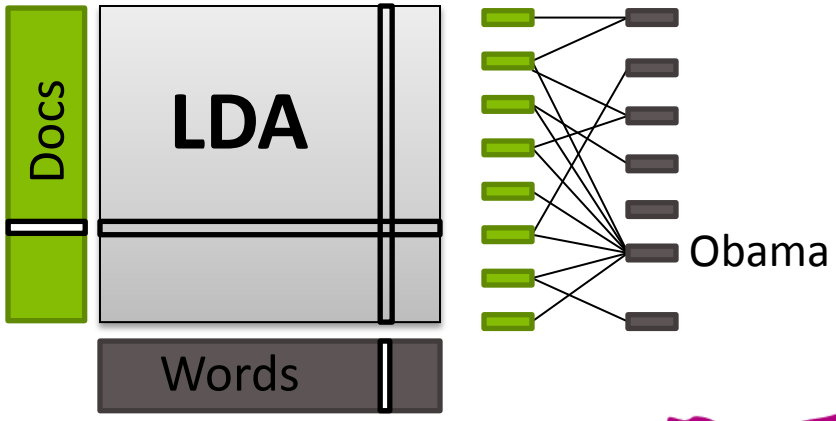
Popular Movies



Hyper Parameters



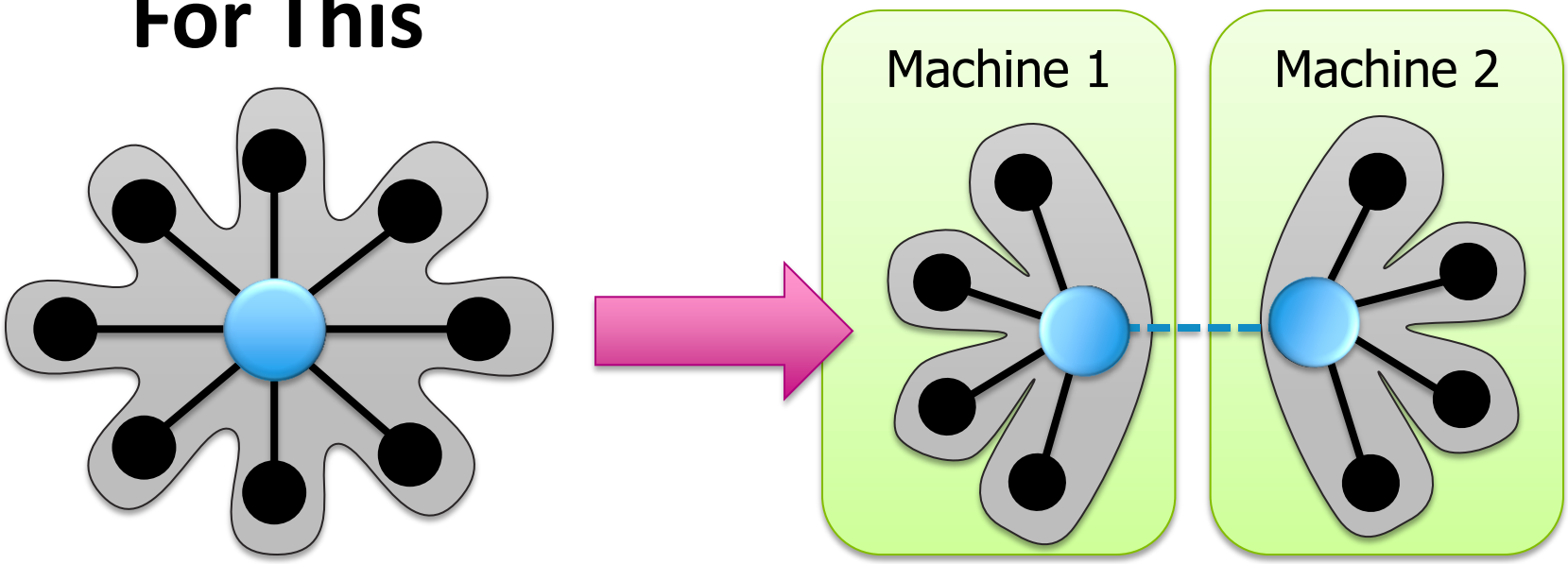
Common Words



GraphLab 2 Solution

Program For This

Run on This

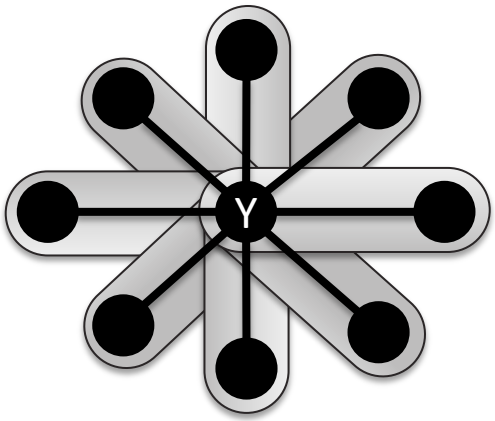


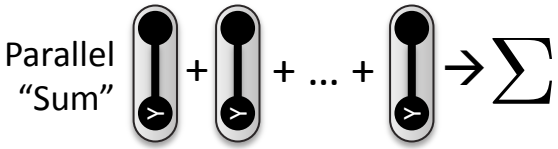
- Split **High-Degree** vertices
- **New Abstraction →** *Leads to this Split Vertex Strategy*

GAS Decomposition

Gather (Reduce)

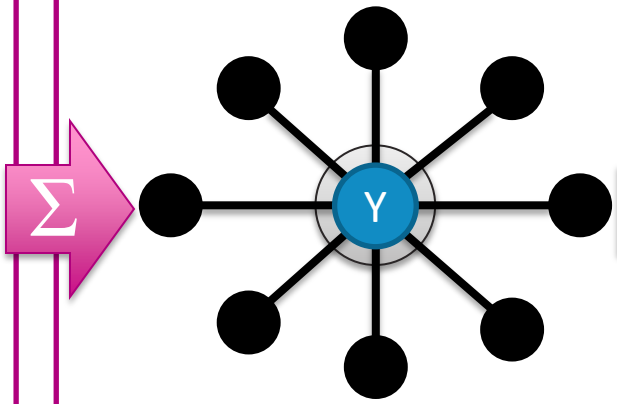
Accumulate information about neighborhood





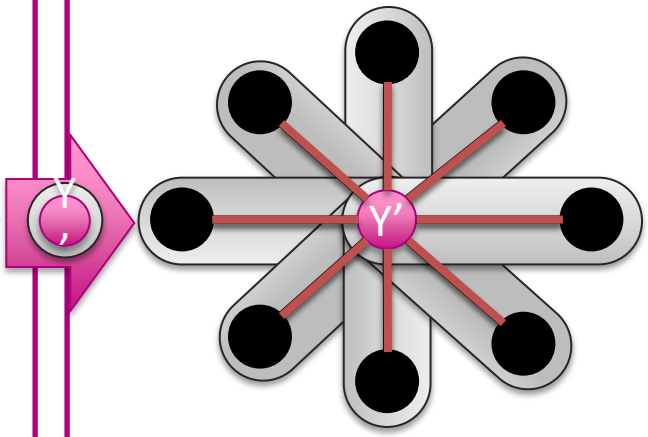
Apply

Apply the accumulated value to center vertex

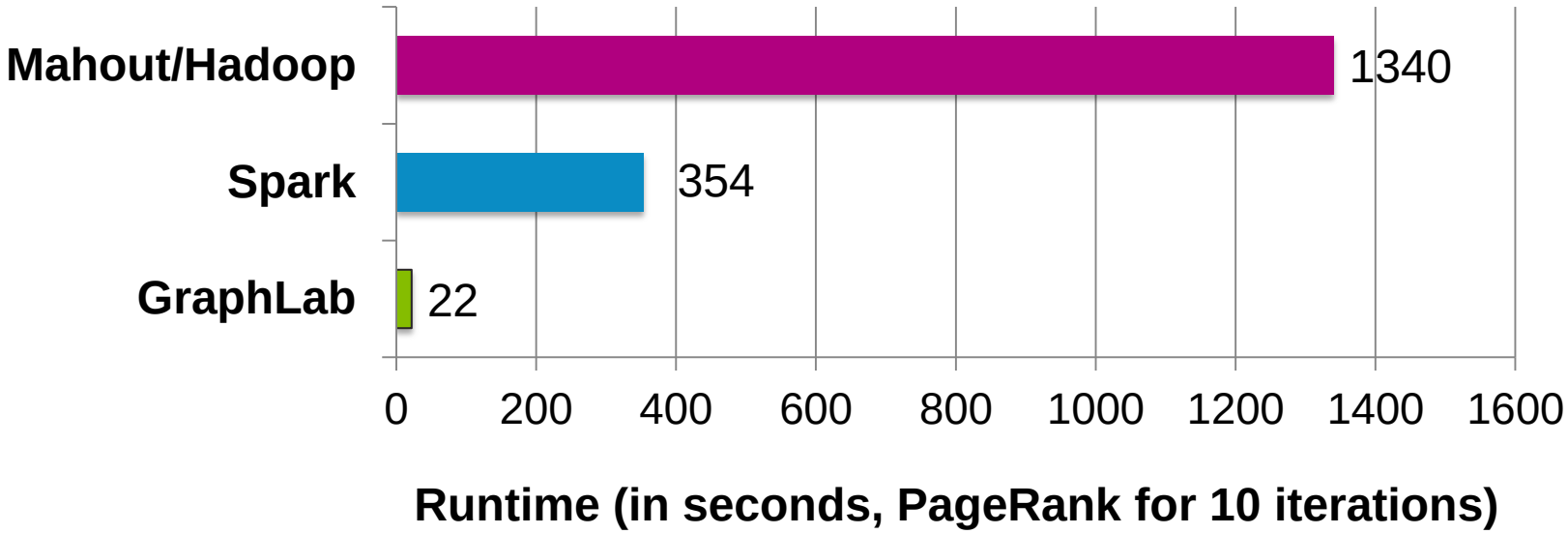


Scatter

Update adjacent edges and vertices.

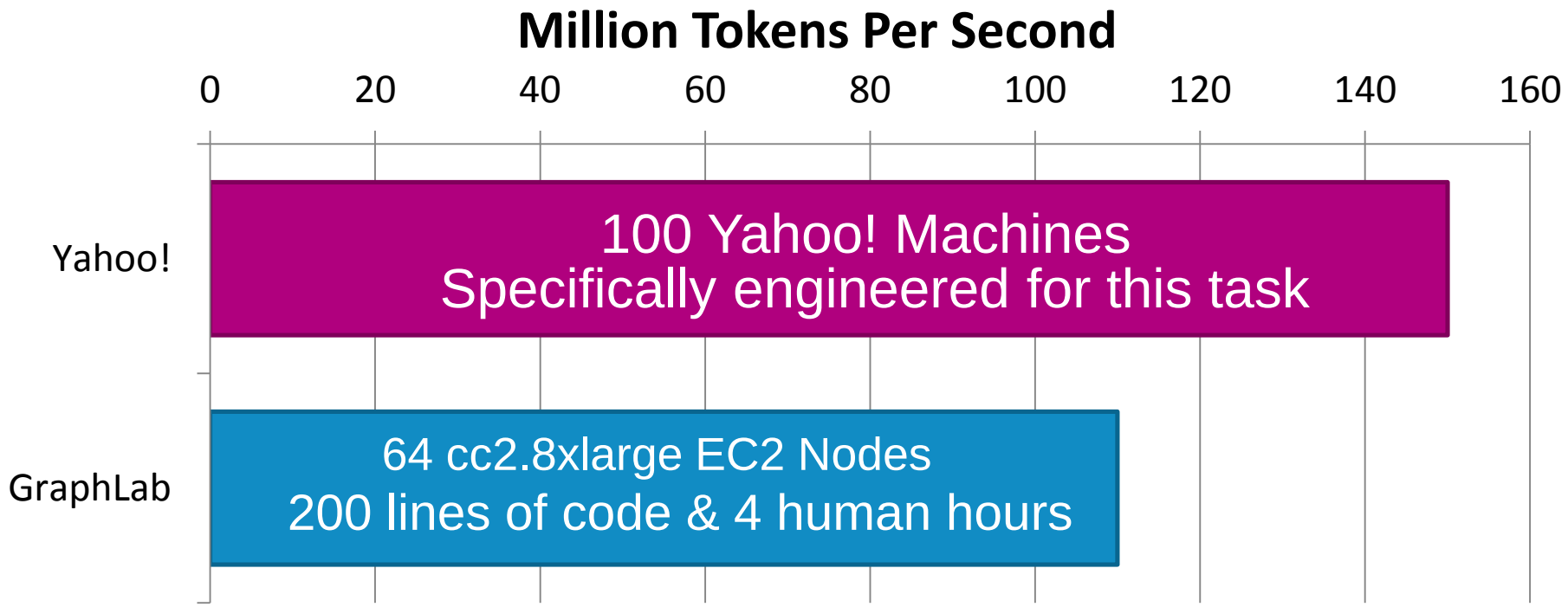


PageRank on the Live-Journal Graph



GraphLab is 60x faster than Hadoop
GraphLab is 16x faster than Spark

Topic Modeling (LDA)



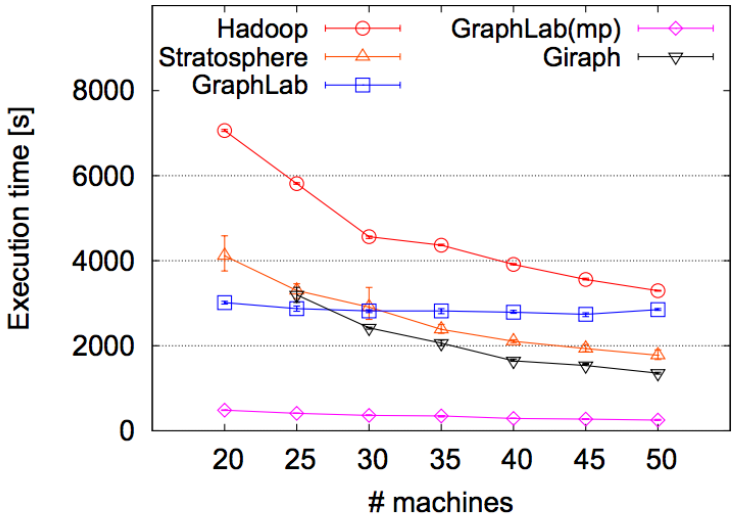
English language Wikipedia

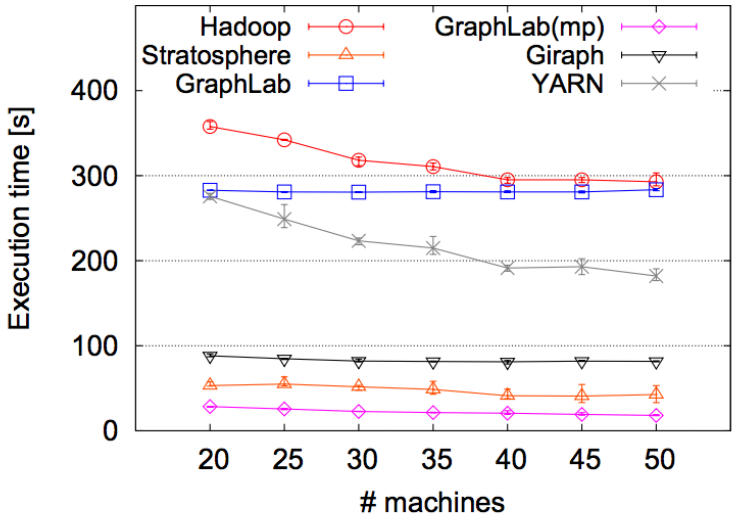
- 2.6M Documents, 8.3M Words, 500M Tokens

Computationally intensive

GraphLab vs. Giraph

Figure 1: The execution time of algorithm BFS of all datasets of all platforms.





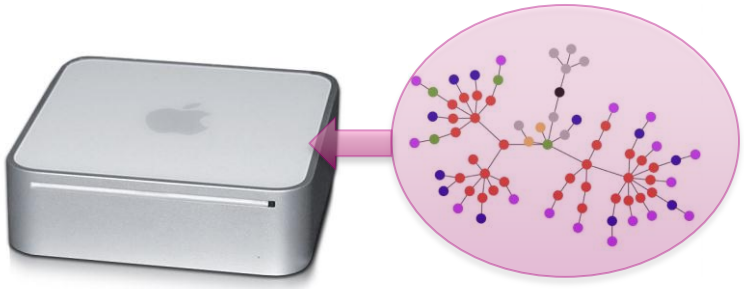
Source: SC13 paper

GraphChi (2011)

GraphChi: Going small with GraphLab

GraphLab

Solve huge problems on small or embedded devices?

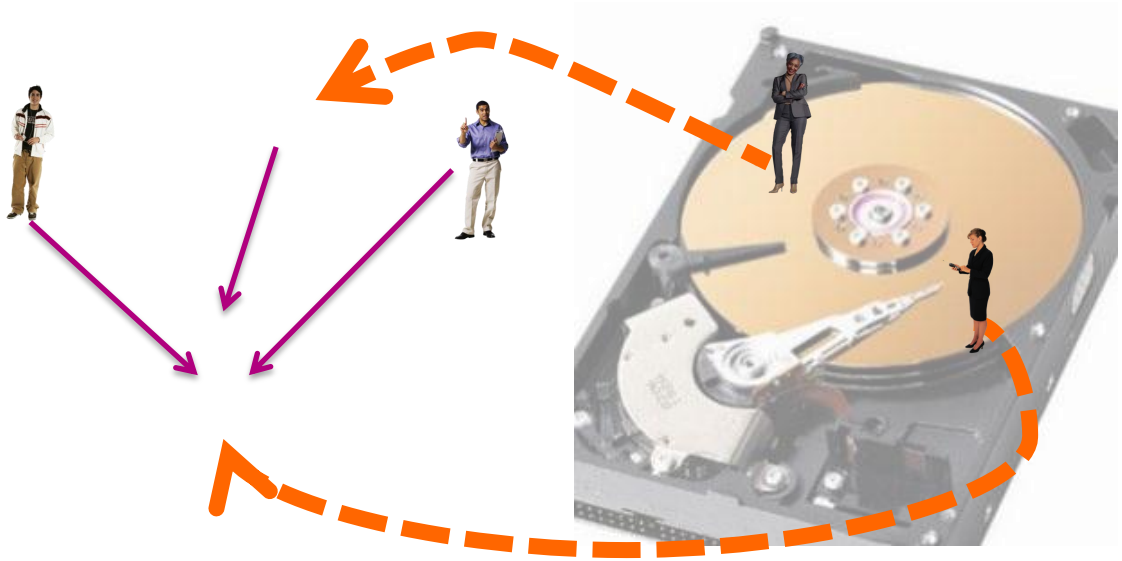


Key: Exploit non-volatile memory (starting with SSDs and HDs)

GraphChi – disk-based GraphLab

Challenge:

Random Accesses



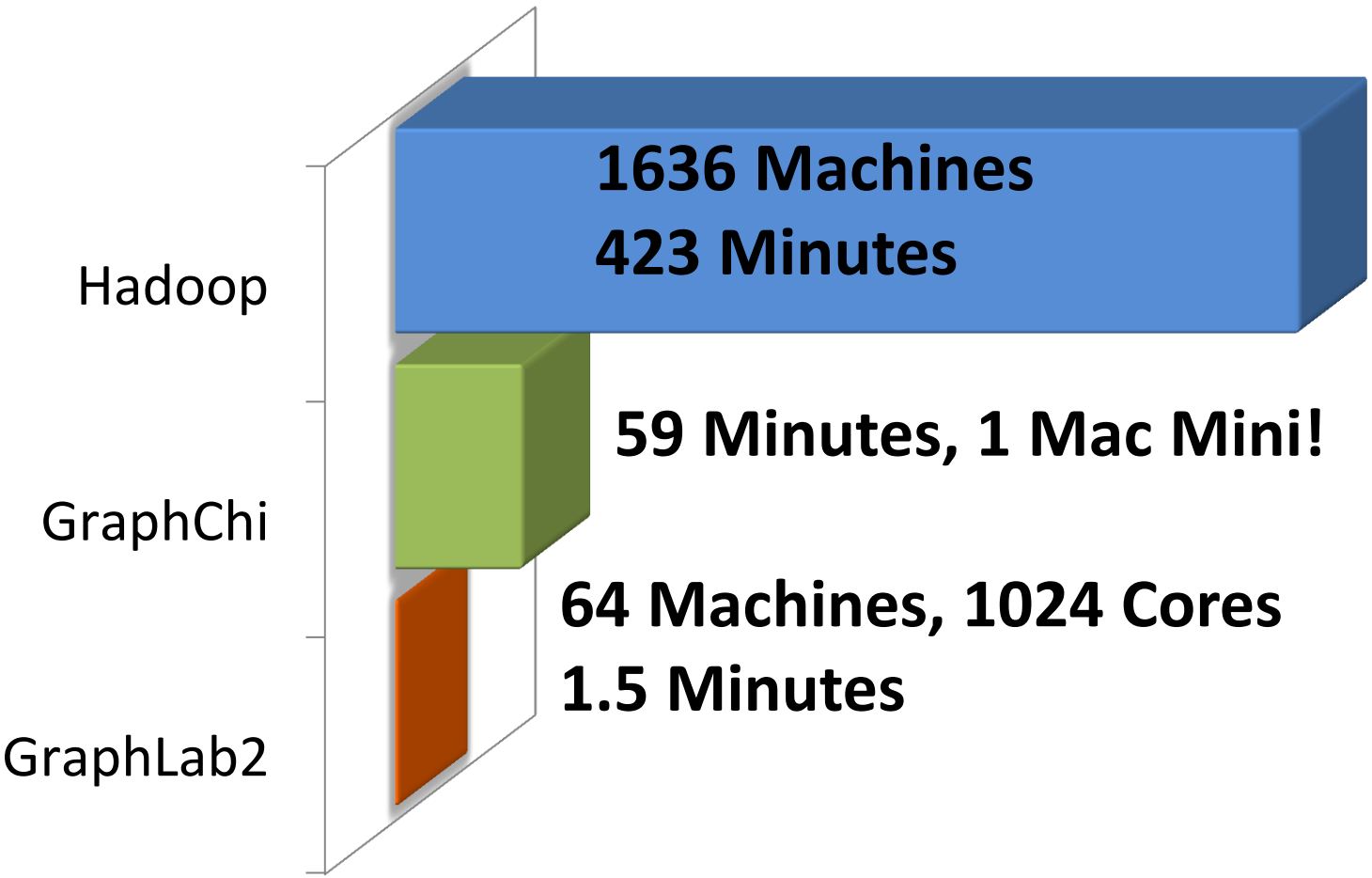
Novel GraphChi solution:

Parallel sliding windows method ➔ *minimizes number of random accesses*

Triangle Counting on Twitter Graph

40M Users
1.2B Edges

Total: 34.8 Billion Triangles

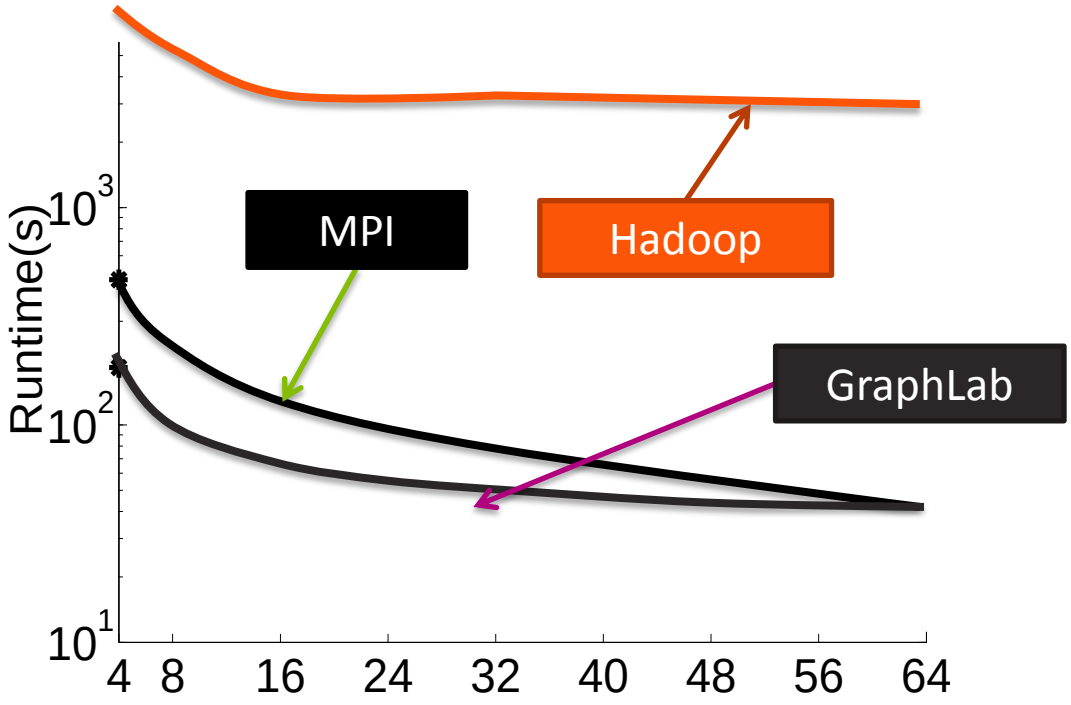


Hadoop results from [Suri & Vassilvitskii WWW '11]

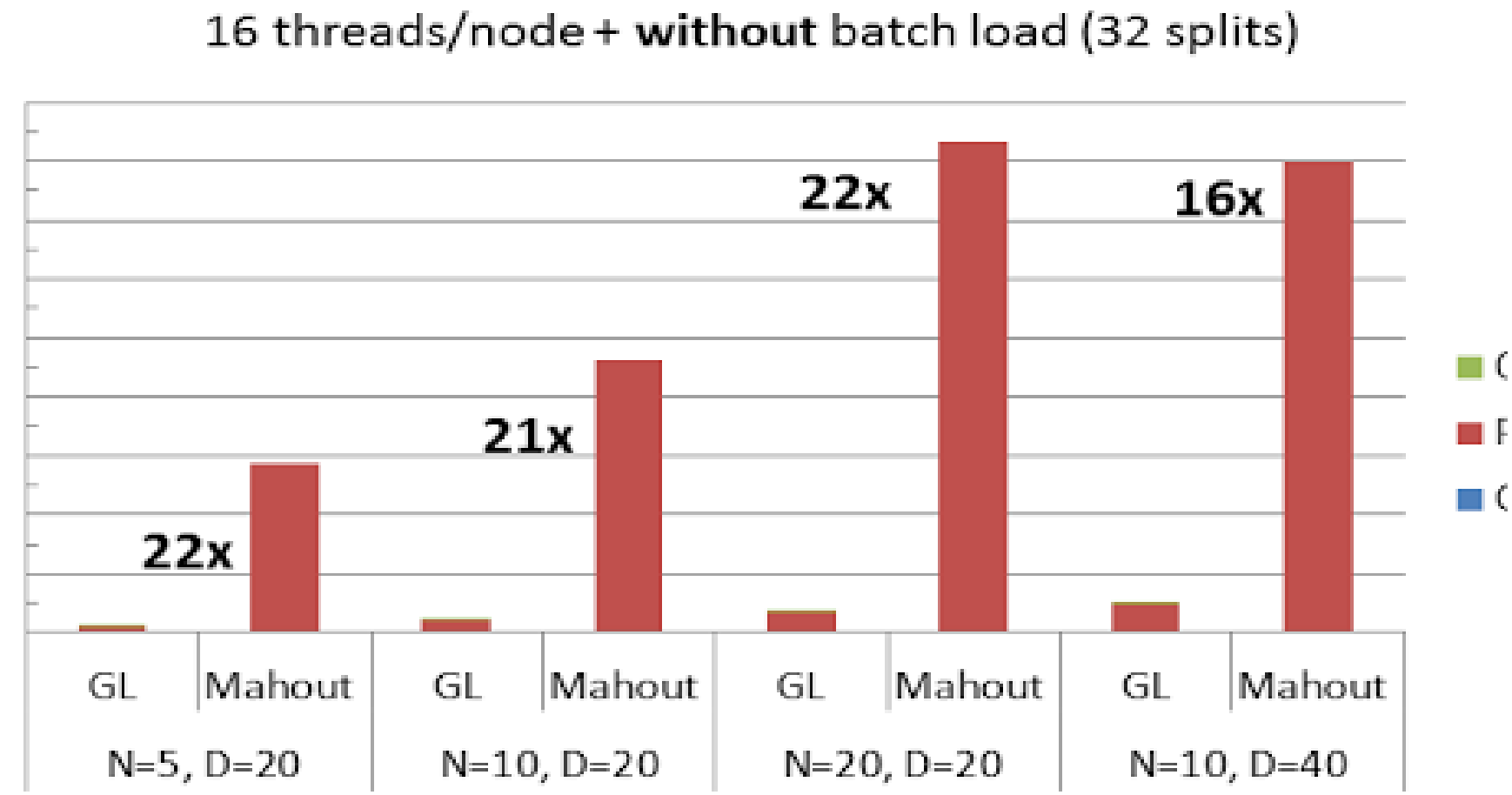
Netflix Collaborative Filtering

- Alternating Least Squares Matrix Factorization

Model: 0.5 million nodes, 99 million edges

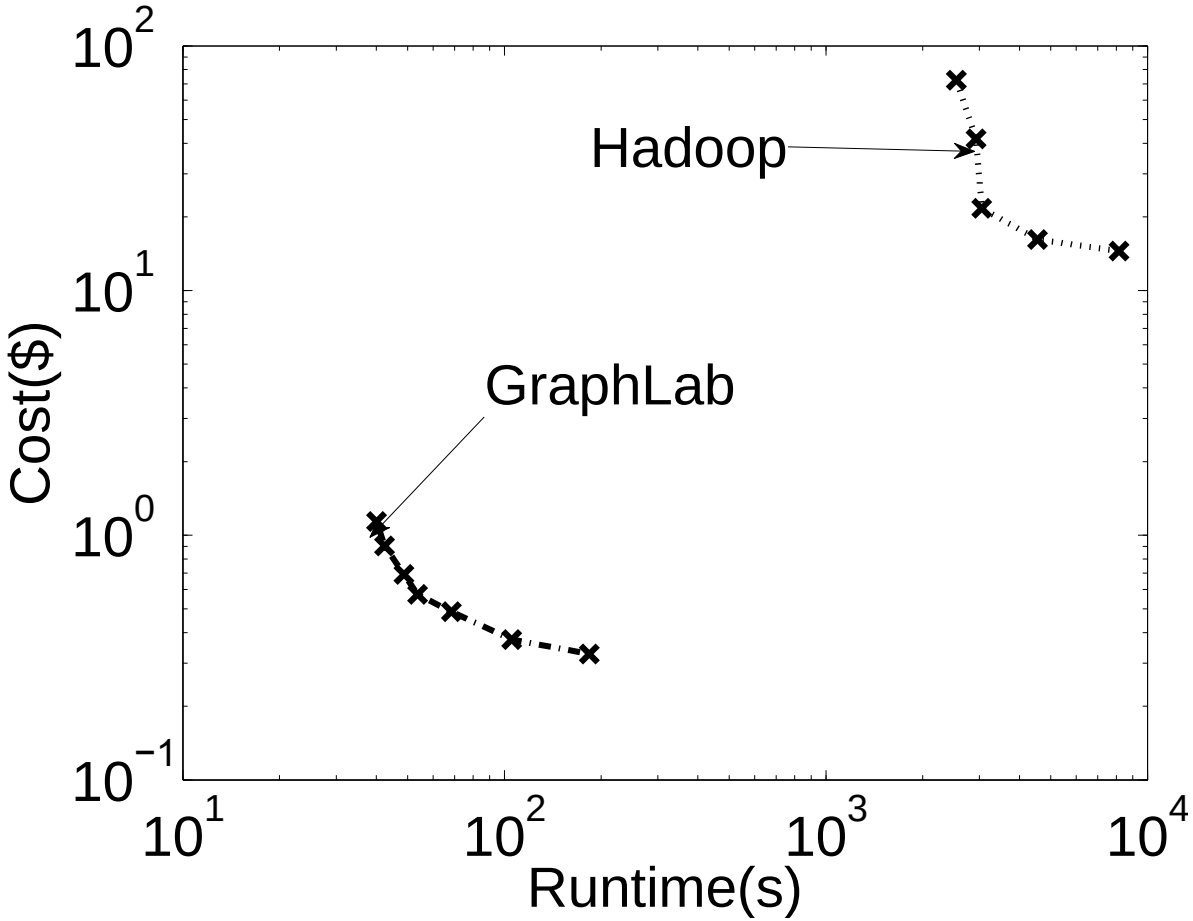


Intel Labs Report on GraphLab



Data source: Nezih Yigitbasi, Intel Labs

The Cost of Hadoop



Growing User Community and Adoption

GraphLab Conferences

2012 → 2013

3RD CONFERENCE JULY 21 2014

Growing community contribution

Kobo contrib
GraphLab

London
for Grap

DARI
Grap

collabo
Kobo r
GraphL

Drug repurposing using GraphLab

Recently I learned about an interesting work by Murat Can Cobangolu, a graduate student at the
CMU-
intera
The ba
that w
were
as a st

Virginia Tech CloudCV project contributes ADMM co
GraphLab

Some additional GraphLab open source co
contributions announced today. Dhruv Bhat
Tech Lab contributed today the recently ma
algorithm by Boyd: alternating directions m
multipliers (ADMM). The algorithms are no
graphical models toolkit.

*"We implemented ADMM and Bethe-ADMM
inference in MRFs.*

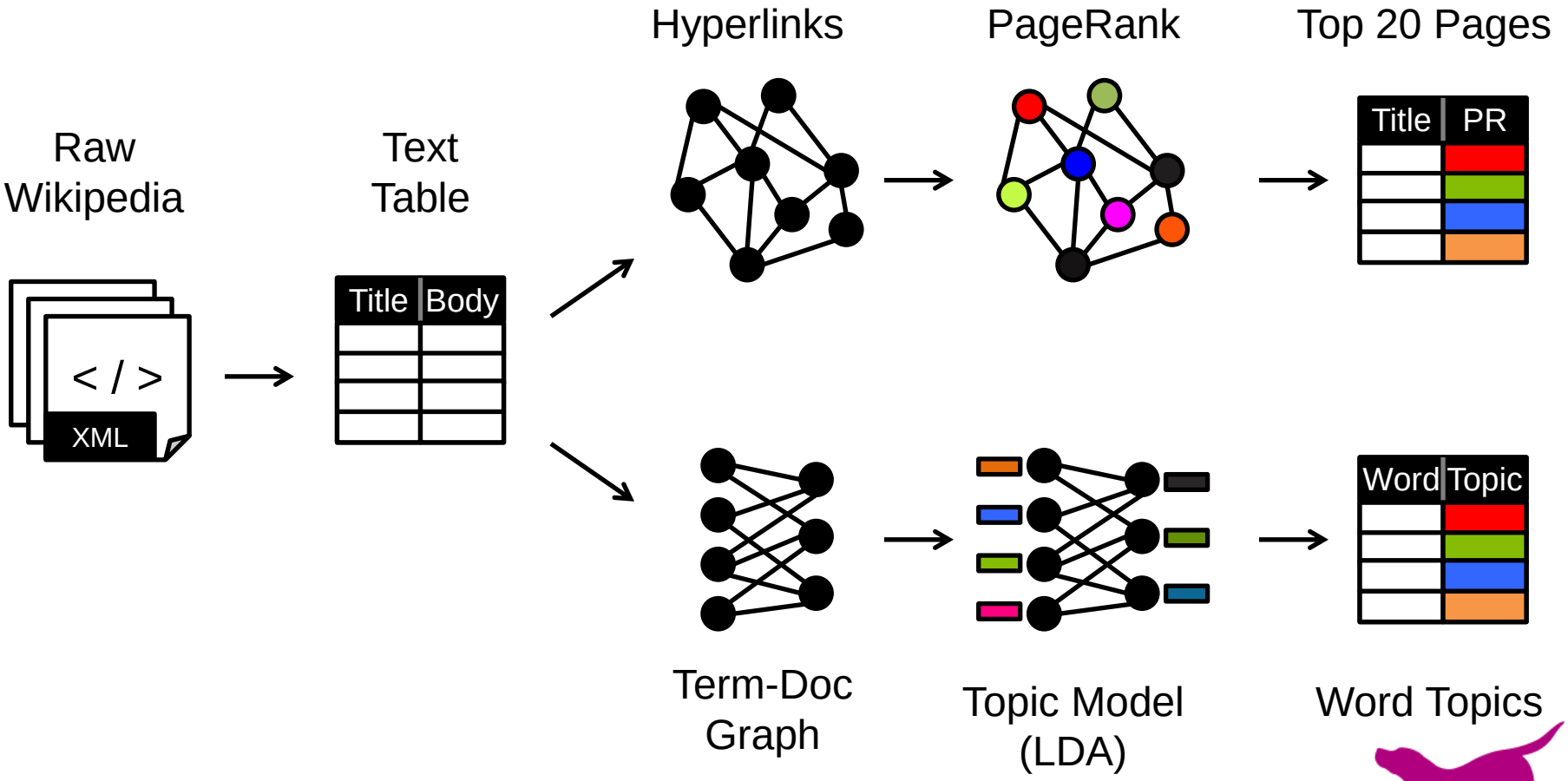
The algorithms are reported in the following

GraphLab

Unleash Data Science

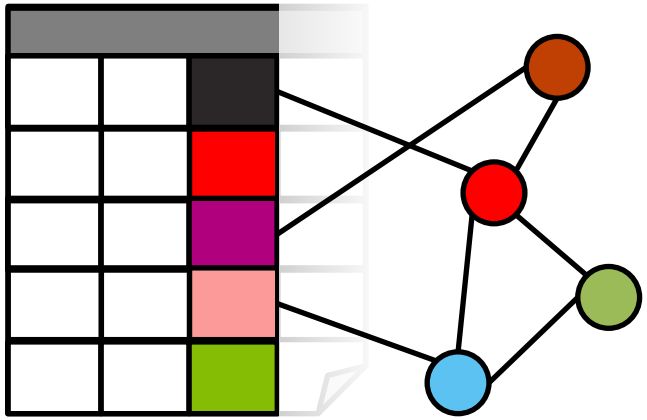
Power + Simplicity

Real-World Pipelines Combine Graphs & Tables



GraphLab Create:
Blend Graphs & Tables

Enabling users to easily and efficiently express the entire graph analytics pipelines



within a simple Python API.

Machine Learning is a powerful tool but …

even basic applications can be challenging.

6 months from R/Matlab to production (at best).

state-of-art algorithms are trapped in research papers.

Goal of GraphLab:
Make large-scale machine learning accessible to all! ☺

Now with GraphLab: Learn/Prototype/Deploy

Even basics of scalable ML can be challenging

Learn ML with GraphLab Notebook

6 months from R/Matlab to production, at best

pip install graphlab then deploy on EC2

State-of-art ML algorithms trapped in research papers

Fully integrated via GraphLab Toolkits

Value Proposition

"Data scientists tend to use a **variety of tools**, often across **different programming languages**… require a lot of **context-switching** which **affects productivity** and **impedes reproducibility**."

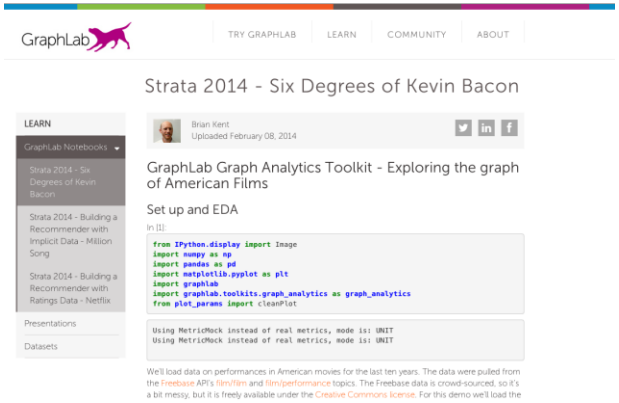
Ben Lorica, O'Reilly Media

GraphLab Create: From prototyping to production without context switching

Three Steps to Simplicity

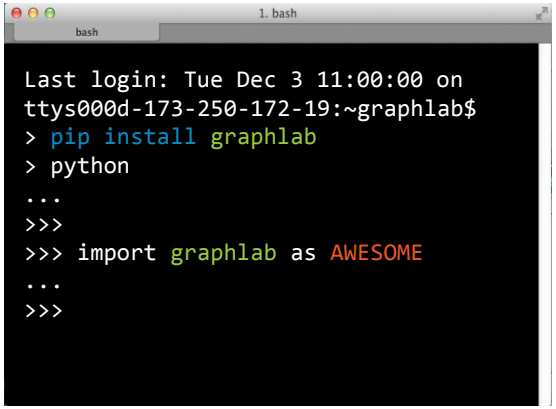
Learn

Learn ML with GraphLab Notebook



Prototype

Easy Install graphlab



Deploy

Easily Scale GraphLab with EC2 or GraphLab Platform

```
>>> import graphlab
>>> graphlab.launch("cc2.8xlarge")

or

Publish Notebook to Collaborators
```

Learn: GraphLab Notebooks

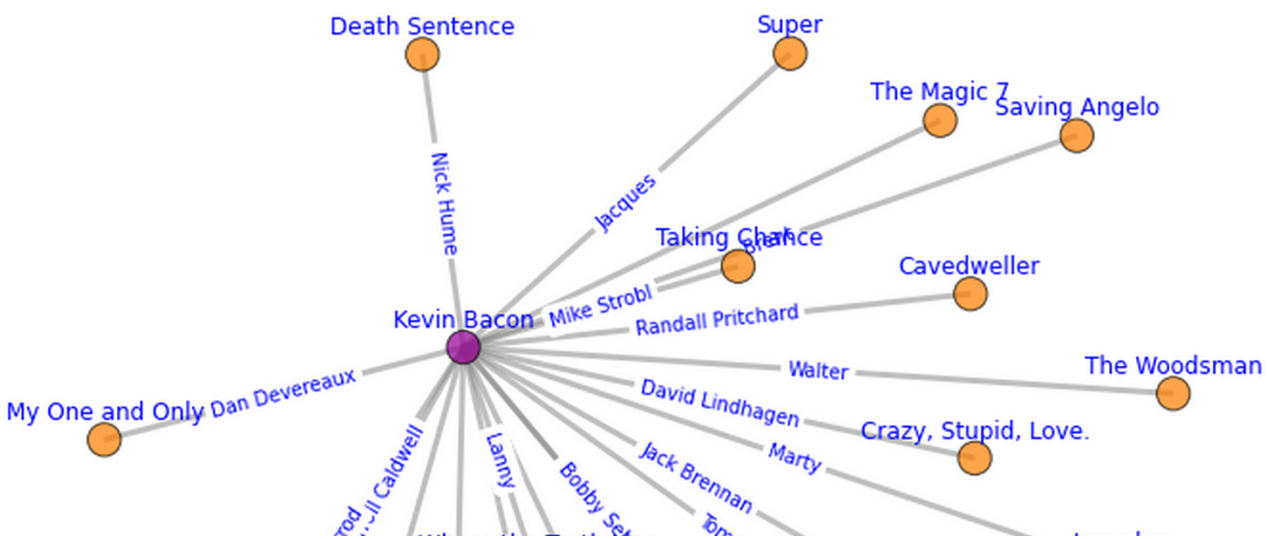
The Kevin Bacon game

OK, let's play the Kevin Bacon game. First, let's see what movies he's been in over the last decade...

In [15]:

```
bacon_films = g.get_edges(src_ids=['Kevin Bacon'])

subgraph = graphlab.Graph()
subgraph = subgraph.add_edges(bacon_films, src_field='__src_id',
                              dst_field='__dst_id')
subgraph.show(vlabel='id', elabel='character', highlight=['Kevin Bacon'])
```



Prototype: GraphLab Create (beta)

GraphLab Create is a Python package that enables developers and data scientists to apply machine learning to build state of the art data products.

- **Build recommenders fast.** Don't waste time coding from scratch.
- **Code in Python.** Do more in one system with tools you love.
- **Iterate more.** Don’t wait for tomorrow to improve results.
- **Scale with ease.** Create on your laptop, deploy to the Cloud.

Build an end-to-end recommender in **six** lines of Python

```
>>> import graphlab

>>> data = graphlab.SFrame("s3://my_data.csv")

>>> model = graphlab.toolkits.recommender.Model()
>>> model = model.train(data, user="user_id", item="item_id")
>>> model.recommend(users=["Sasha", "Zoe", "Juan"], k=100)

>>> model.save("s3://my_model.gl")
```

Deploy: **GraphLab Create**:

Easily **install** & **prototype** locally with new Python API

jegonzal@osmium2: ~ — ~...

```
jegonzal at osmium2 in ~
$ pip install graphlab
```

Deploy to the cluster in **one step**

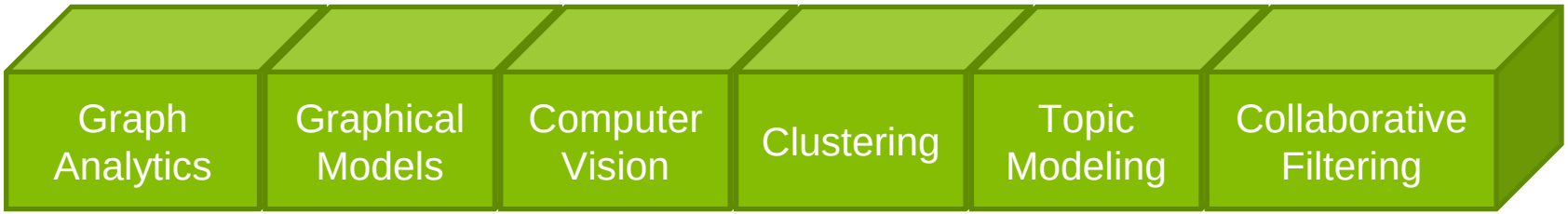
movielens_cloud.py

```
import graphlab

graphlab.launch("tcp://128.95.73.109:12000")  # point to server
dataserver = "http://testdatasets.s3-website-us-west-2.amazonaws.com/"
```

GraphLab Toolkits

Highly scalable, state-of-the-art machine learning straight from python



continually growing with external contributors across industry and academia.

Collaborative Filtering Vertical

- Award winning software for collaborative filtering
  - We ranked top places in several high profile competitions: ACM KDD CUP 2011, ACM KDD CUP 2012, WCSD 2013

- GraphLab software is used by thousands of companies, Some examples:
  - Pandora uses GraphLab for recommending music
  - Adobe uses GraphLab for recommending designers in their social network
  - King is using GraphLab for recommending game moves
  - References from the above companies will be given upon request

Unmatched functionality

- Side features
- Cold start support (new users)
- High dimensional models
- RESTful API (in the works)

GraphLab License

- Open source: Apache 2
- Python: closed source. Licensed (currently free)

Build scalable data products fast

GraphLab

Join our community at GraphLab.com

Follow us @graphlabteam