GraphBuilder: Collaborating to Construct Large-Scale Graphs

Ted Willke
Systems Architecture Lab

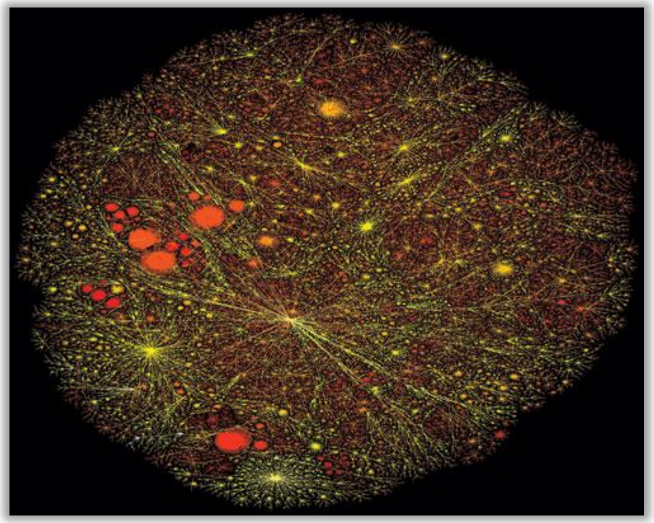
November 30th, 2012

http://www.istc-cc.cmu.edu/

Intel Science & Technology Center for Cloud Computing

10005

Petascale graphs: The end-to-end challenge



Full Internet Map
[Lumeta]

Social Graph
[Facebook]

- GraphLab is indeed promising
- But we struggled with feeding it and other practicalities
- Set out to study potential approaches…

But first…
a bit on how we got here.

Danny Bickson
Postdoc CMU

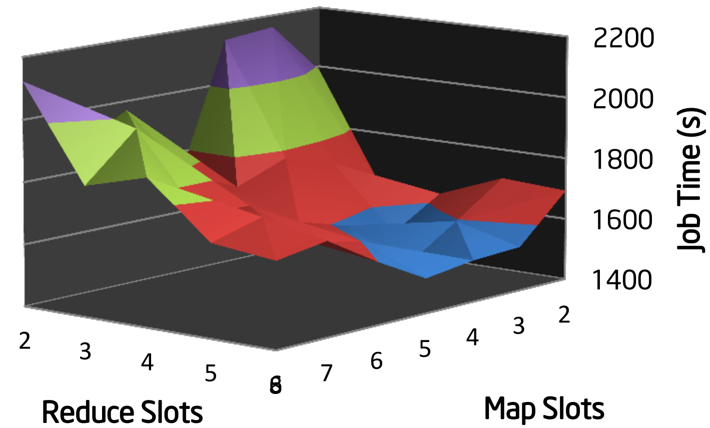
Haijie (Jay) Gu
PhD Candidate
CMU
and Summer 2012
Intern

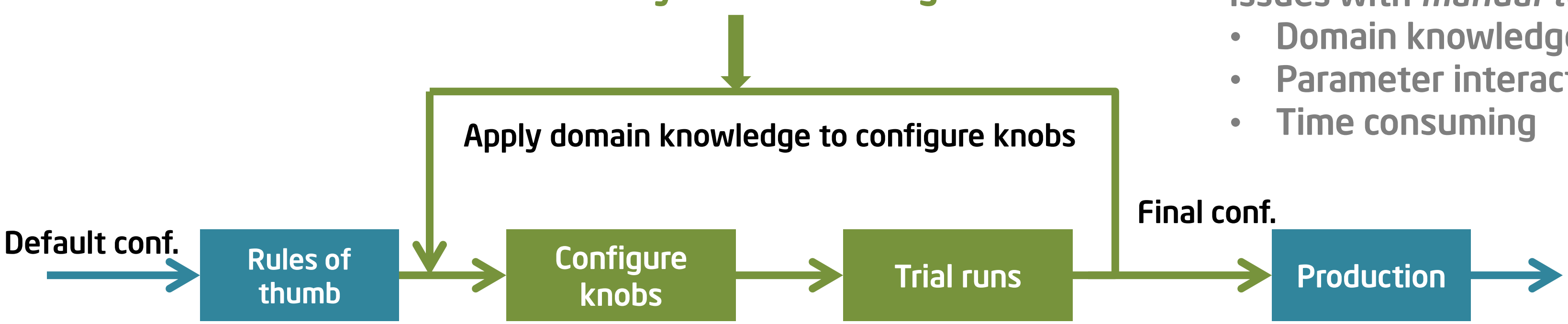
Joseph Gonzalez
Postdoc AMPLab

Yucheng Low
PhD Candidate CMU

Hadoop Research

- Evaluating new cluster technologies requires solid baselines
- But an exhaustive search for the best Hadoop configuration would take **7,257,600,000 lengthy trials!**
- Solve the challenge and accelerate our other work at the same time?





Issues with *manual tuning*:
- Domain knowledge
- Parameter interaction
- Time consuming

We spend weeks tuning clusters for a few days of experiments.

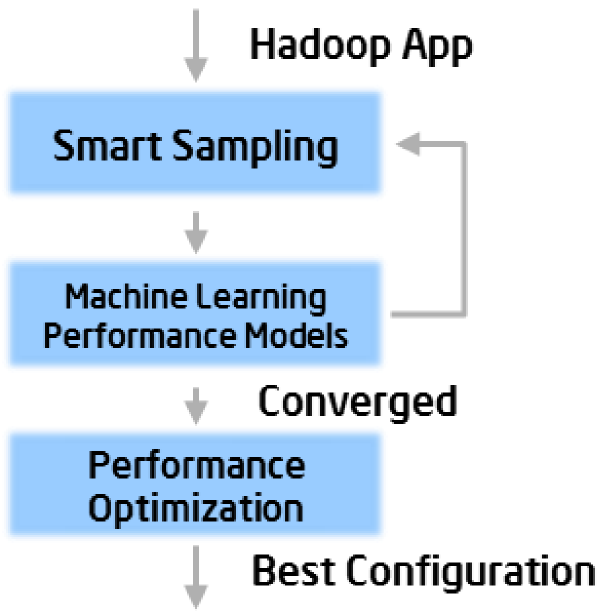
Our Approach

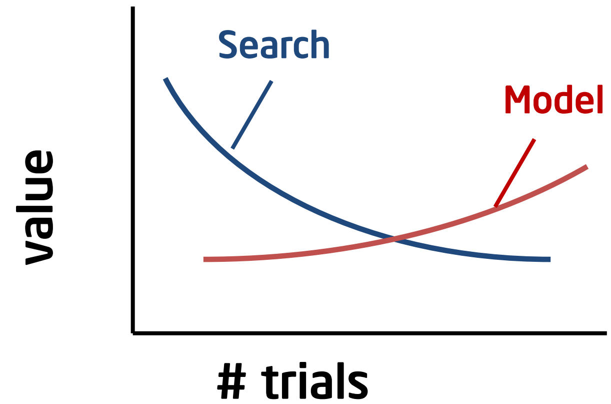
1. Focus on the most important parameters for each circumstance*
2. Apply generalized search algorithms to efficiently explore the parameter space
3. Model the system to reason about unexplored space

* Future work

Gunther: The Elephant Trainer

An Auto-tuner for Hadoop MapReduce

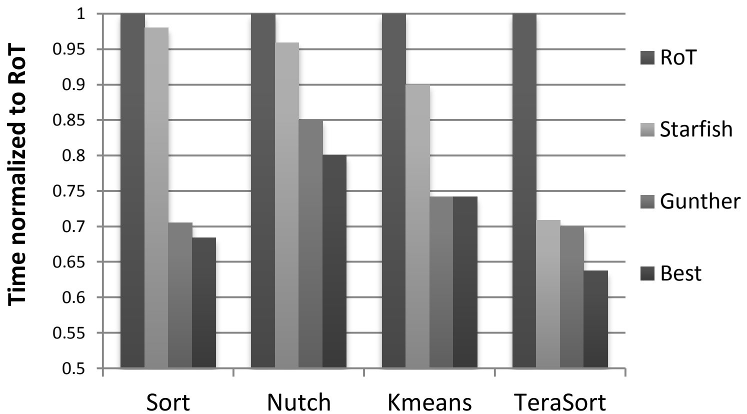




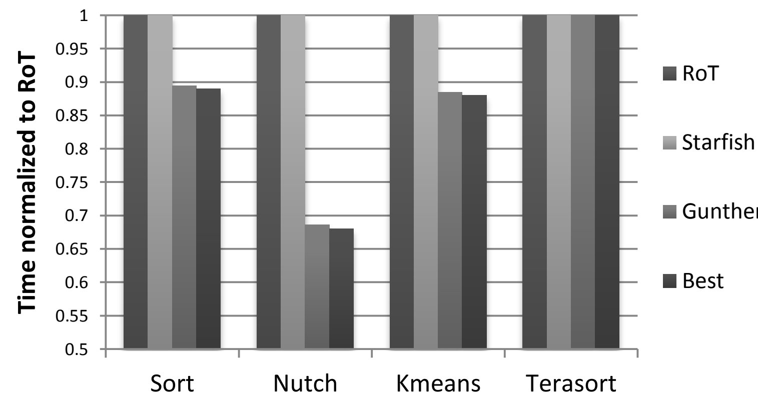
Key benefits:
- Little domain knowledge required
- Easily adapts to new datasets, workloads, frameworks, & clusters
- More effective and faster than manual approaches

Search + Model

Genetic search algorithm is ~95% *effective* in <30 trials



Storage bottlenecked cluster



Network bottlenecked cluster

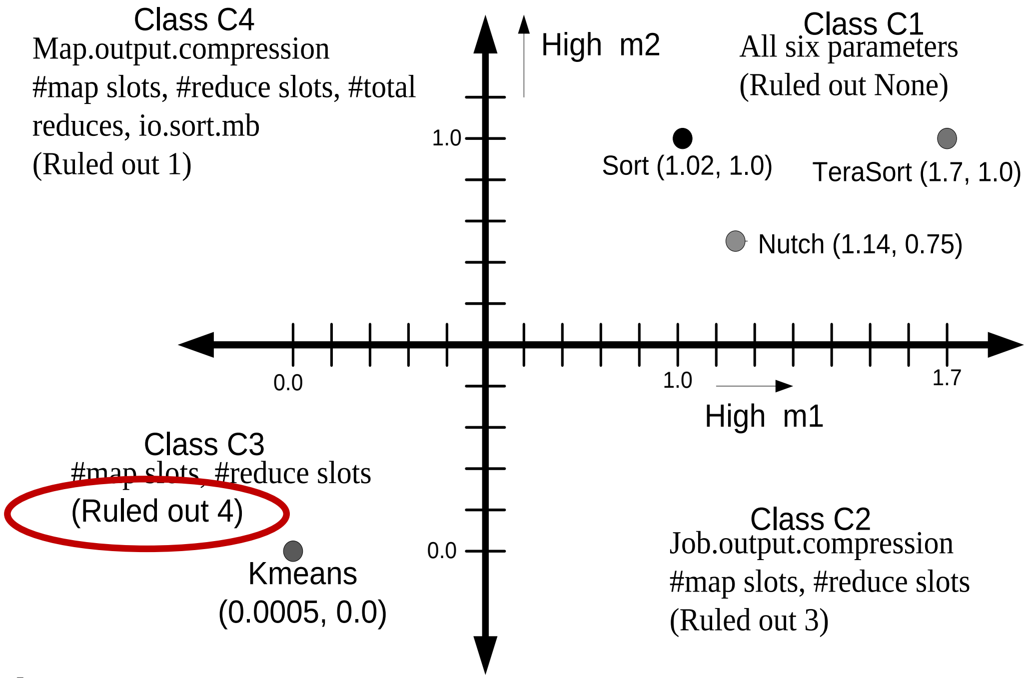
SVR model is very *accurate* but requires hundreds of trials (320 in this case)

Modeling Approach	SNB Cluster							ZT Cluster						
	min	Q1	median	mean	Q3	max	IQR	min	Q1	median	mean	Q3	max	IQR
MLR	0	7	15	17	23	68	16	0	8	18	23	33	117	25
MLR-I	0	6	14	16	22	69	16	0	8	17	22	31	129	23
MLR-Q	0	6	11	15	22	66	16	0	7	16	19	27	92	20
MLR-IQ	0	6	11	14	20	67	14	0	7	15	18	25	87	18
ANN	0	4	10	12	17	61	13	0	4	10	12	17	61	13
M5Tree	0	5	10	12	17	65	12	0	5	10	14	19	71	14
SVR	0	2	4	8	10	73	8	0	3	6	10	13	64	10

Apply model to predict perf and inform future searches.

Dimensionality Reduction

Rule out parameters *up front* that primarily affect resources that aren't likely to bottleneck



$$m1 = \frac{spilled}{map+reduce\ inputs}$$

$$m2 = \frac{HDFS\ bytes\ written}{HDFS\ bytes\ read}$$

Direction:

1. **Incorporate node- & cluster-level utilization observations (*m*) into model**
2. **Apply EV-based MV analysis offline to determine *what* params matter *when***
3. **Use 1st run to collect *m* and apply to search**

But tuned MapReduce is still MapReduce.

“Chance favors the *connected* mind.”

--Steven Johnson

Twister
Iterative MapReduce

HaLoop

A new country

Spark
Lightning-Fast Cluster Computing

Apache
Hama

BOOM
Berkeley Orders Of Magnitude

Twitter
Storm

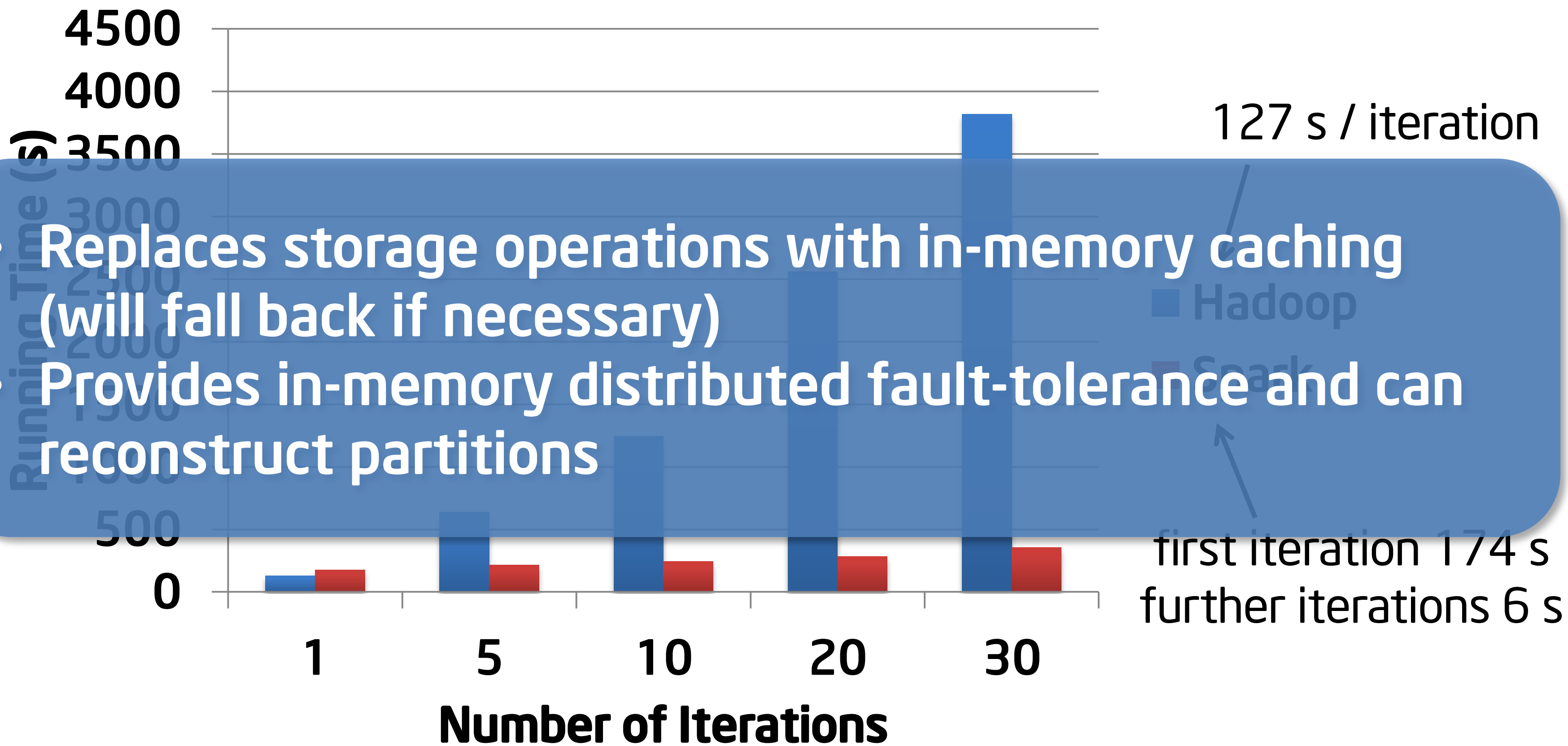
GraphLab

A new planet

Garth made the connection at the December 2011 ISTC retreat!

Spark*

Fast, Interactive, Language-Integrated Cluster Computing

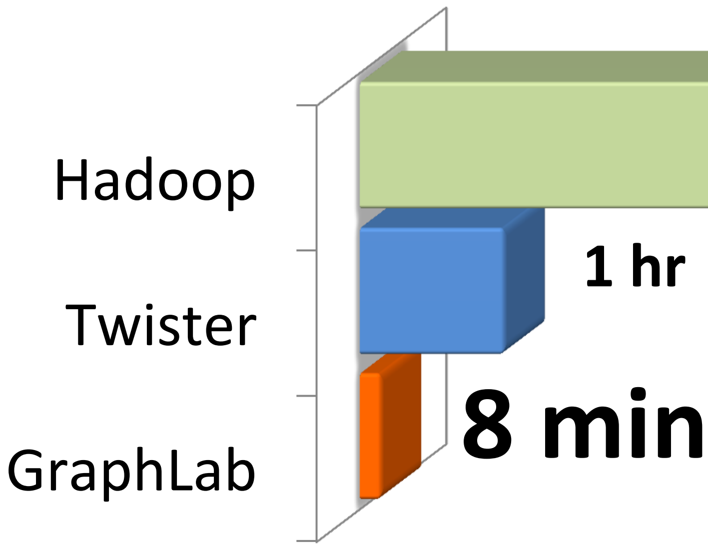


- Replaces storage operations with in-memory caching (will fall back if necessary)
- Provides in-memory distributed fault-tolerance and can reconstruct partitions

*Zaharia et al. UC Berkeley. Retrieved from www.spark-project.org.

PageRank on 64 DP Xeons

40M Webpages, 1.4 Billion Links



GraphLab is 41X faster. What's going on here?

Hadoop results from [Kang et al. '11]
Twister (in-memory MapReduce) [Ekanayake et al. '10]

Big graphs are a big deal!

100B Neuron
100T Relationships

Human Brain

1B Users
140B Friendships

Social Network

1Trillion Pages
100s T Links

Internet

Millions of Products & Users

e-commerce

27M Users
70K Movies

Online Services

Large Biological Cell Networks

Science

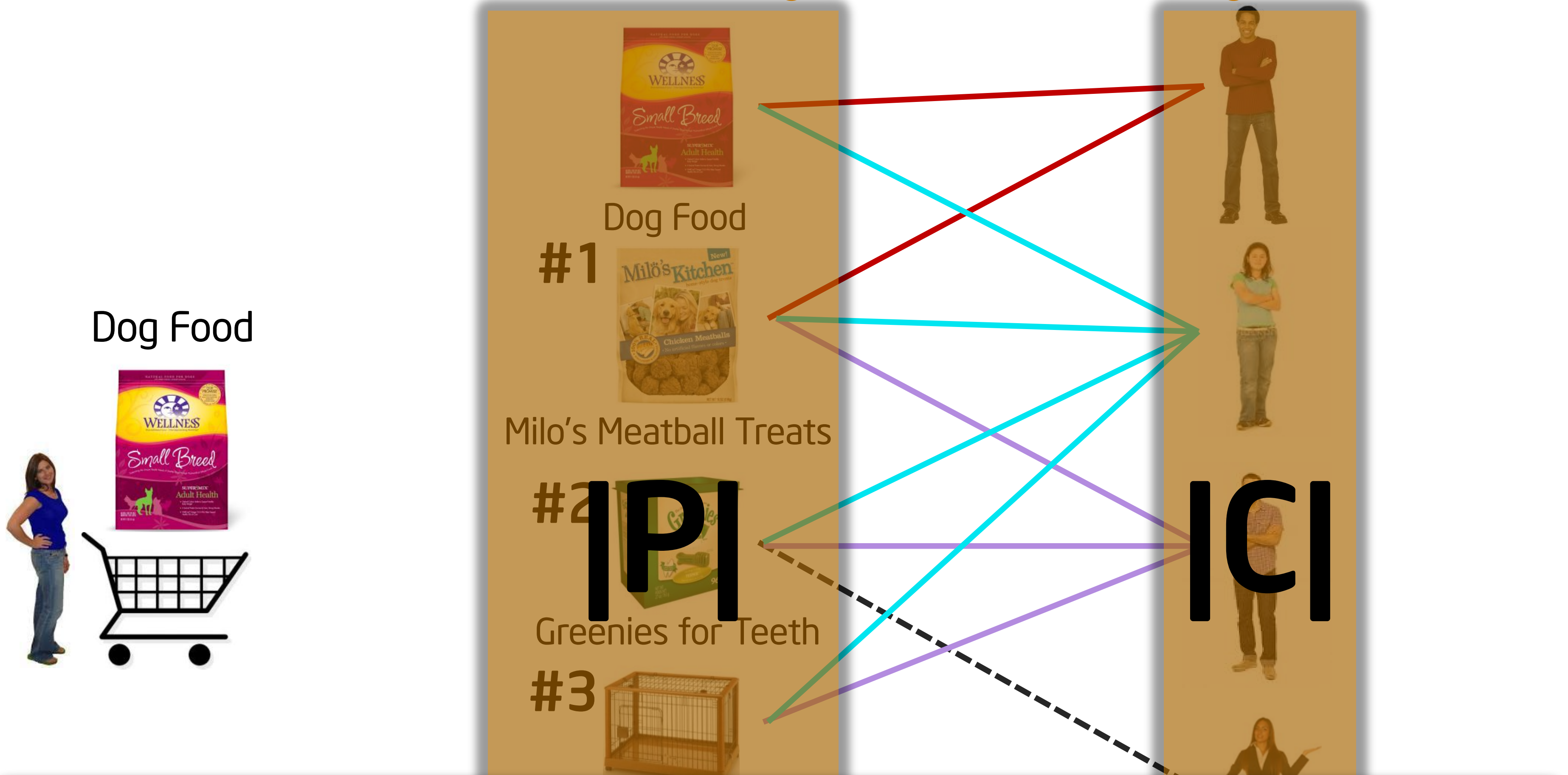
Many problems involve irregular data structures most naturally expressed as graphs, trees, and arbitrary sets

(paraphrasing Keshav Pingali)

Image source: [Wikipedia][alz.org] [Facebook]

Collaborative Filtering: Mining Relationships

Customers Who Bought This Item Also Bought What?



Time to find product similarities is

- **Days** worst case
- **Minutes** if algorithm **exploits data dependency structure**
- **Seconds** if **ideally partitioned** across M machines

Graph processing:
An extremely short history

Data-parallel

▼

Data-parallel + Iterative

▼

Graph-parallel + Iterative

▼

Asynchronous graph-parallel

Ship the **entire** graph structure…
… over and over …
or, better yet, pass the results …
… whenever you want.

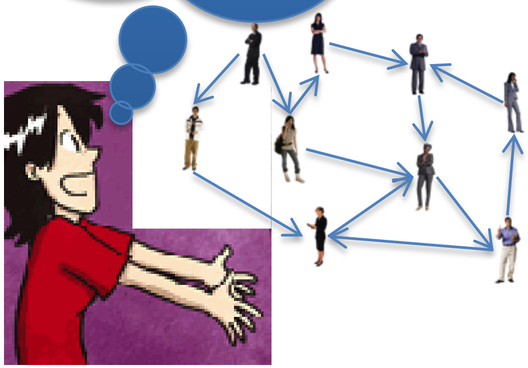
So we’re done, right?

"I spend more than half of my time integrating, cleansing and transforming data without doing any actual analysis. Most of the time I'm lucky if I get to do any *analysis* at all."

Anonymous Data Scientist
from Jeff Heer's (Stanford) interview study, 2012

Taking a Broader Perspective

I think I know how write
a custom script to construct a
graph on *1 machine*



… to run on my
totally awesome graph
processing cluster!!???

Semi-efficient parallel predictions ☹

* Adaptation of GraphLab team material

How do we construct the graph?

How do we store it? Query it?

Analyze it? ______ it?

Many of these challenges are solved for small problems… but what about Internet scale?

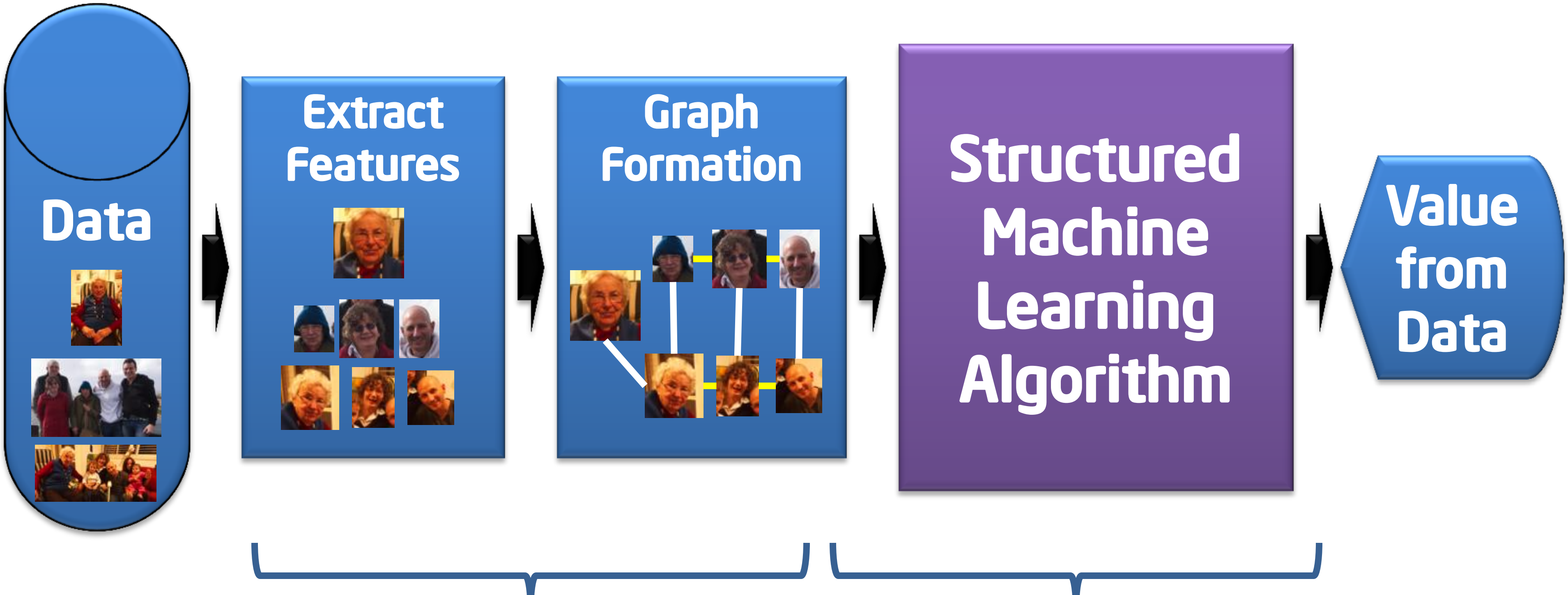
Challenges for Emerging Area

1. Few people skilled in the apps and algos
2. App frameworks emerging and evolving rapidly
3. Lack of tools to deploy systems and analyze system behavior

Lightly charted territory offers big opportunities for Intel and other companies.

Parallel Machine Learning (ML):
Joint work with the ISTC for CC (UW/CMU)

Machine Learning Pipeline

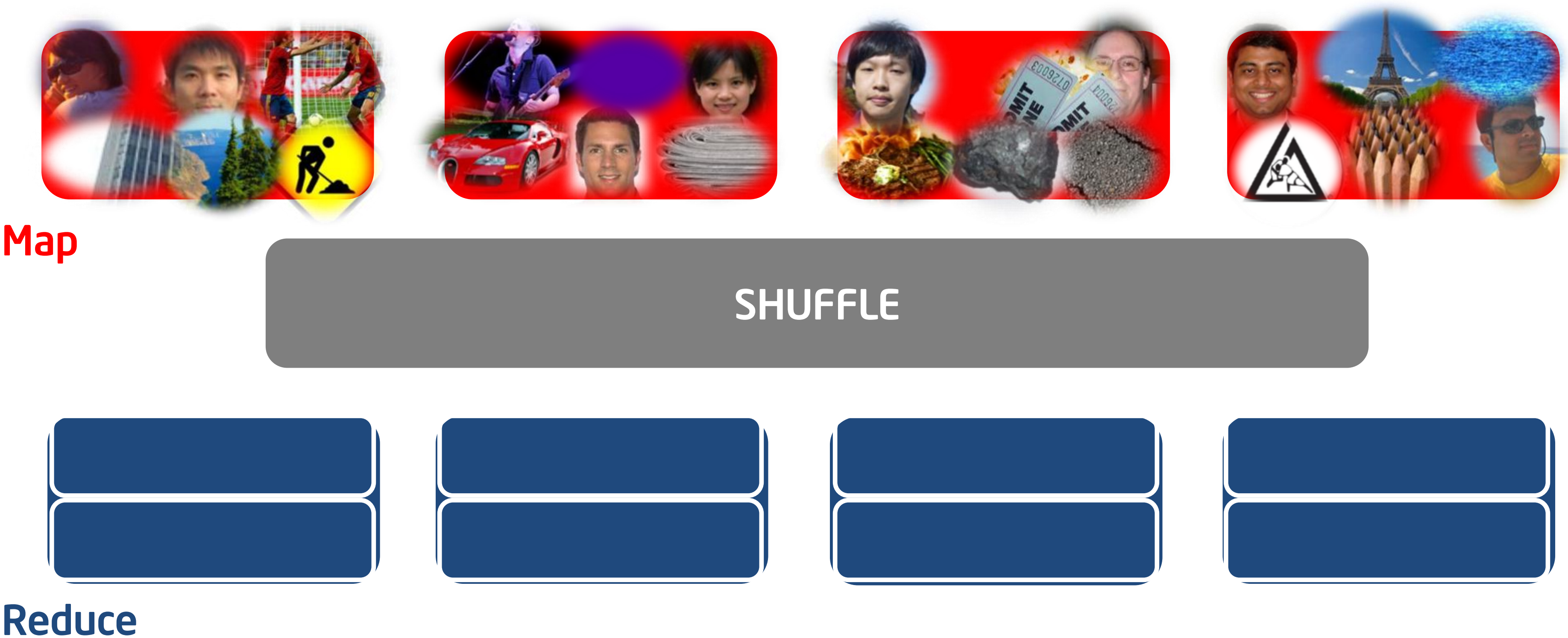


Graph Ingress
mostly data-parallel

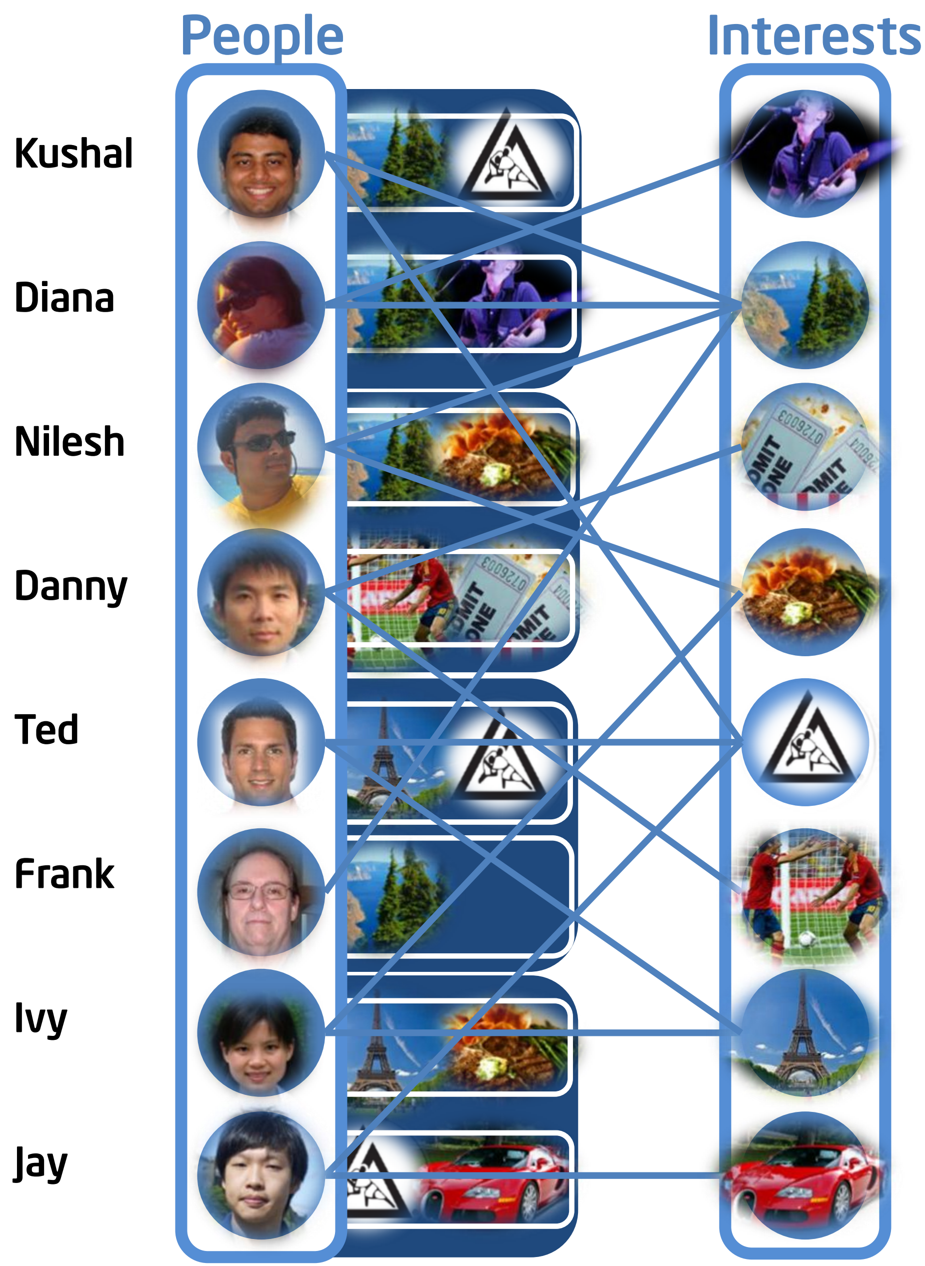
Graph-Structured Computation
graph-parallel

Hadoop for Graph Construction

- Intuitive Map and Reduce programming model (in Java)
- Framework takes care of resource provisioning
- Provides redundant storage and fault recovery



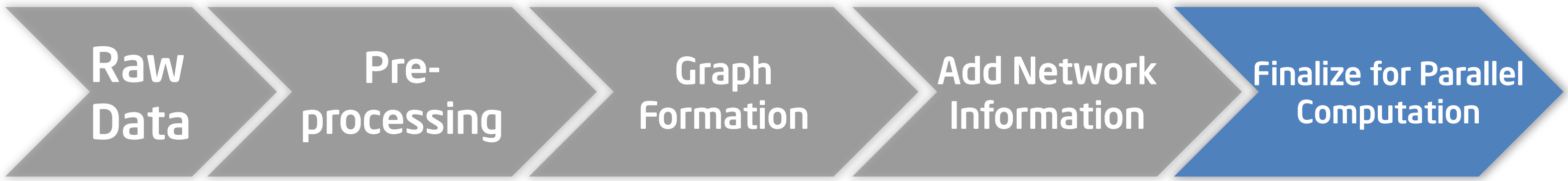
Cluster Computing Architecture


People
Interests
Kushal
Diana
Nilesh
Danny
Ted
Frank
Ivy
Jay

Building Graphs for Practical Apps

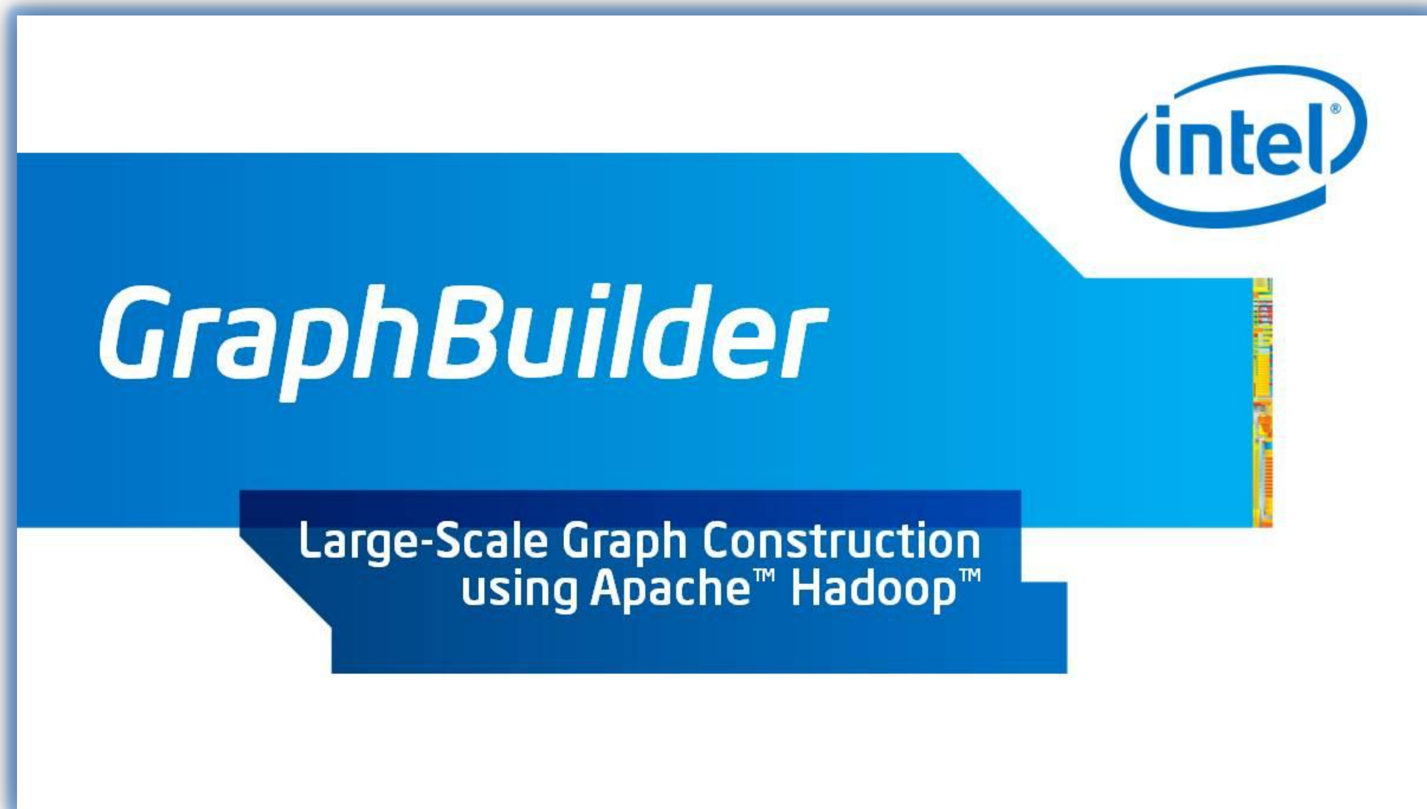
	Raw Data	Pre-processing	Graph Formation	Add Network Information
What **words** are most associated with what **(hidden) topics**?	XML Docs	Extract Doc Names and Words	Bipartite (Docs, Words)	Count Word Frequency
What does context tell me about the **type** (person, place, thing) of this noun?	News Feeds	Extract Noun Phrases and Contexts	Bipartite (NP, Context)	Count NP Frequency & Initialize **type** Distribution
What are the highest **ranked pages**?	Web Pages	Extract Page URLs and Links	Directed Graph	N/A

And, in practice and at scale we must:



- Minimize the use of system resources, like memory, storage, etc.
- Ensure GL's computational effort is load balanced for *power-law graphs*
- Do our best to ensure the graph we generated is the one we intended to

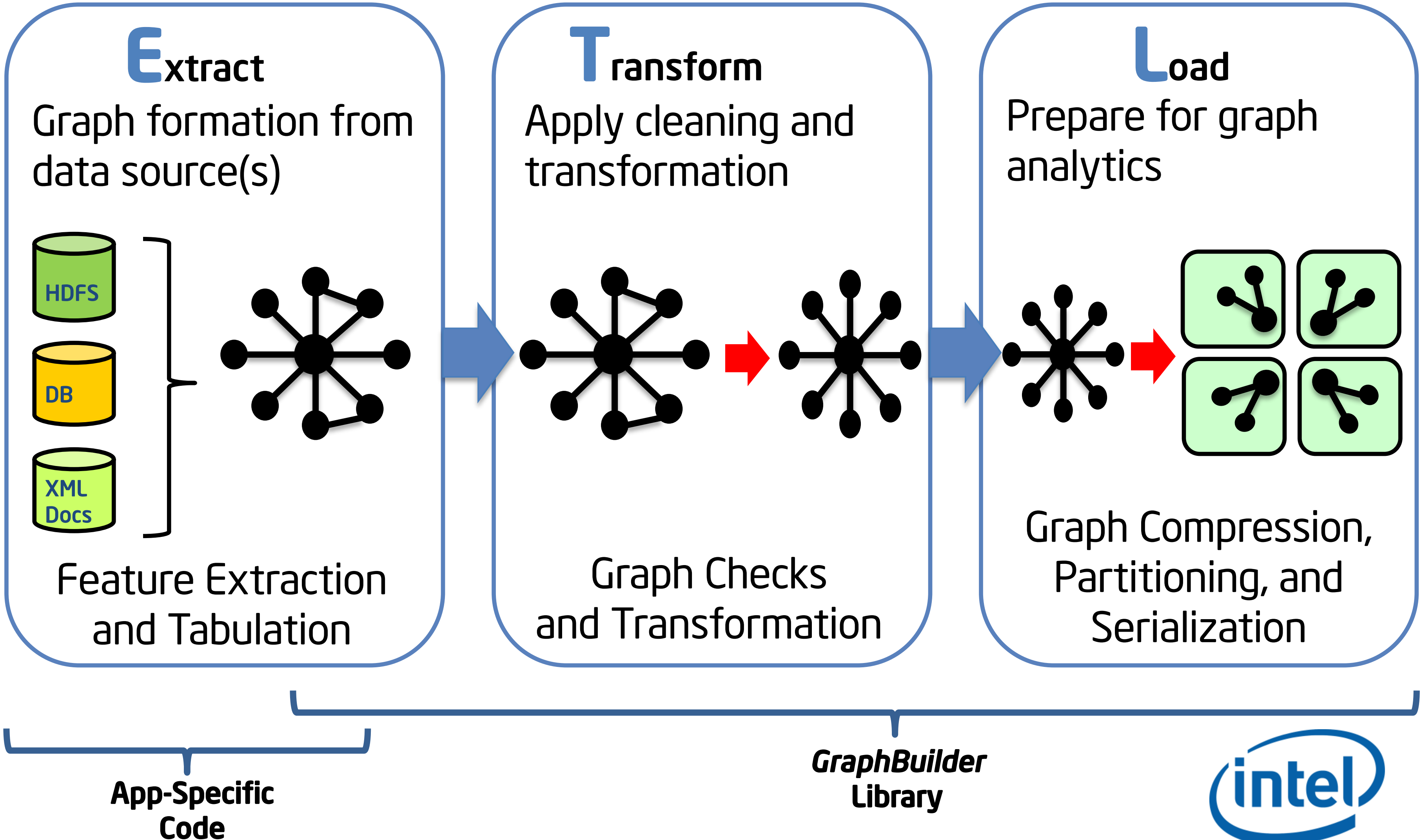
… but the application programmer shouldn't be responsible for this domain expertise!



GraphBuilder makes it easy.

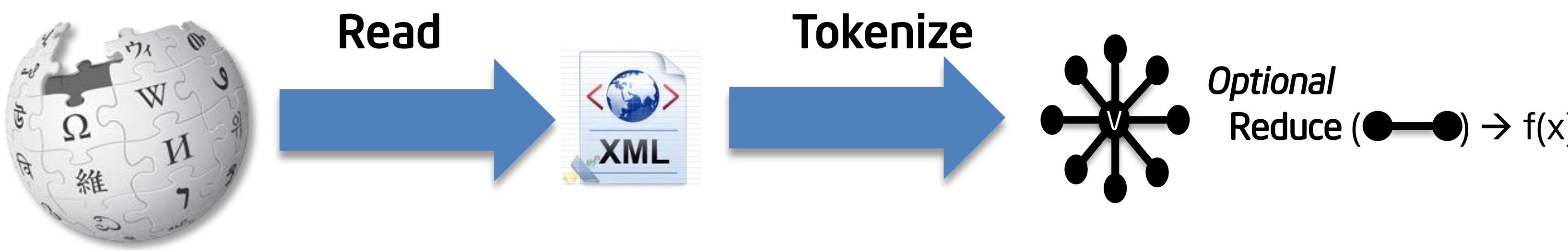
- Fills a hole in the ecosystem
- Written in Java for easy use in Hadoop MR and apps
- Offloads domain expertise

GraphBuilder Data flow



Extract - Graph Formation

Extract features from data to construct relationships




```
conf.set(XMLInputFormat.START_TAG_KEY, START_TAG);
conf.set(XMLInputFormat.END_TAG_KEY, END_TAG);
new XMLRecordReader((FileSplit) split, conf);
```

Read Records

```
Document doc = builder.parse(new
InputSource(new StringReader(s)));
title = xpath.evaluate("//page/title/text()",
doc);
```

Extract Features

```
title = title.replaceAll("\\s", "_");
id = xpath.evaluate("//page/id/text()", doc);
String text =
xpath.evaluate("//page/revision/text/text()",
doc);
parseLinks(text);
```

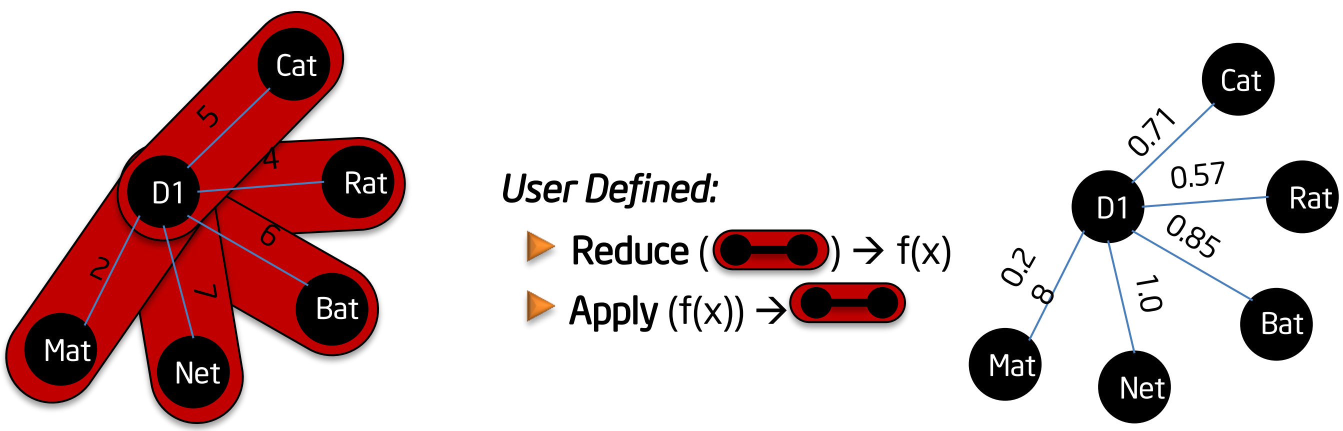
Parse Elements

- Write simple data-specific functions.
- Program sequential, not parallel!

Extract - Tabulation

Built-in tabulation functions for TF, TFIDF, WC, ADD, MUL, DIV.
Interface for custom tabulation on source and/or target vertex

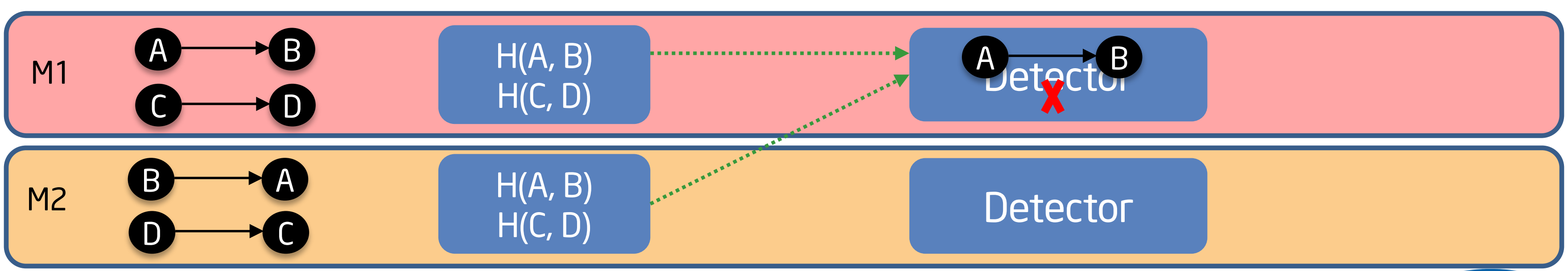
Example: Term Frequency



$$tf(t,d) = \frac{f(t,d)}{\max\{f(w,d) : w \in d\}}$$

Transform – Graph Transforms & Checks

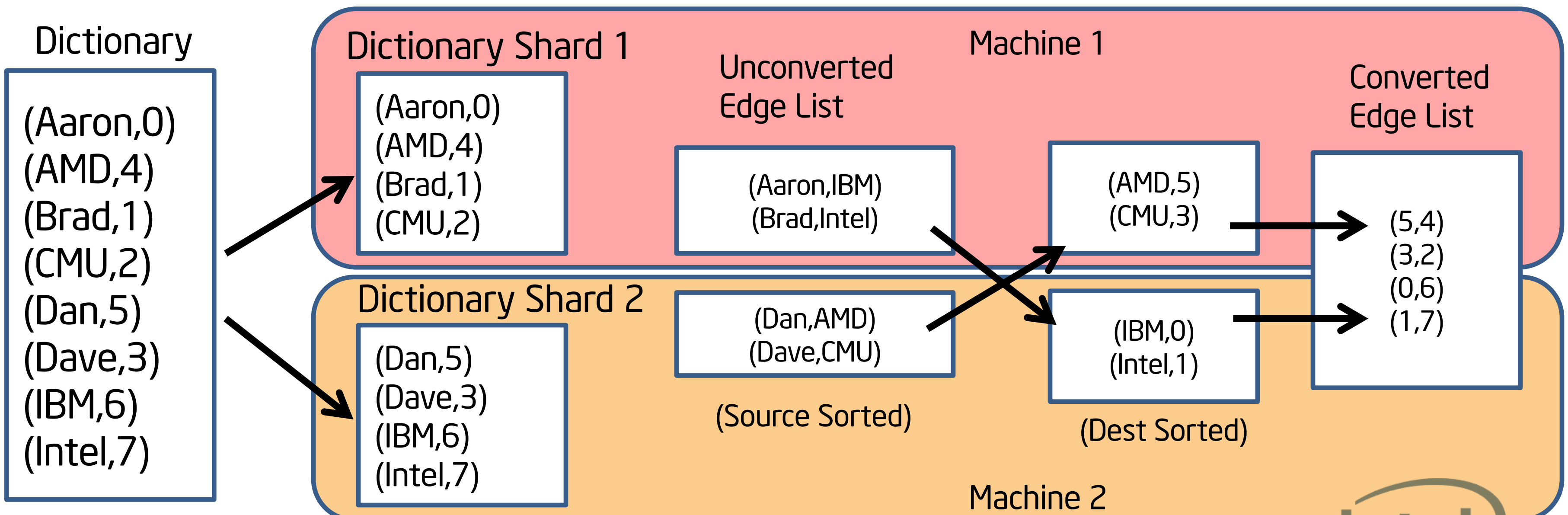
- Would like the ability to:
  - Optionally filter duplicate, dangling and/or self edges
  - Transform a directed graph into an undirected graph
  - Calculate graph statistics, compute sub-graphs, etc.
- The library provides:
  - Functions to perform self-, dangling- and duplicate-edge removal
  - Directionality transformation
- Solutions are based on a distributed hashing algorithm



Steering function

Load - Graph Compression

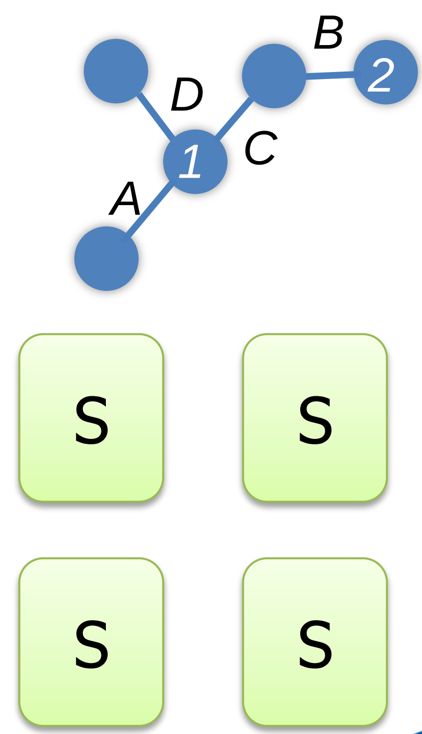
- We can save memory if we compress/normalize graph
- But, seems to call for global lookups in a framework that prefers *independent subproblems*
- A simple, scalable solution is to “shard” ordered lists:



Load - Graph Partitioning

"Cut quality varies inversely with cut balance." [Kevin Lang, '04]

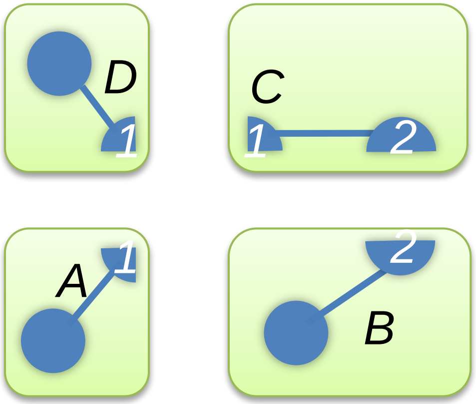
- Minimize communications by minimizing the number of machines *v* spans
- Place about the same number of *edges* on each machine



Load - Graph Partitioning

"Cut quality varies inversely with cut balance." [Kevin Lang, '04]

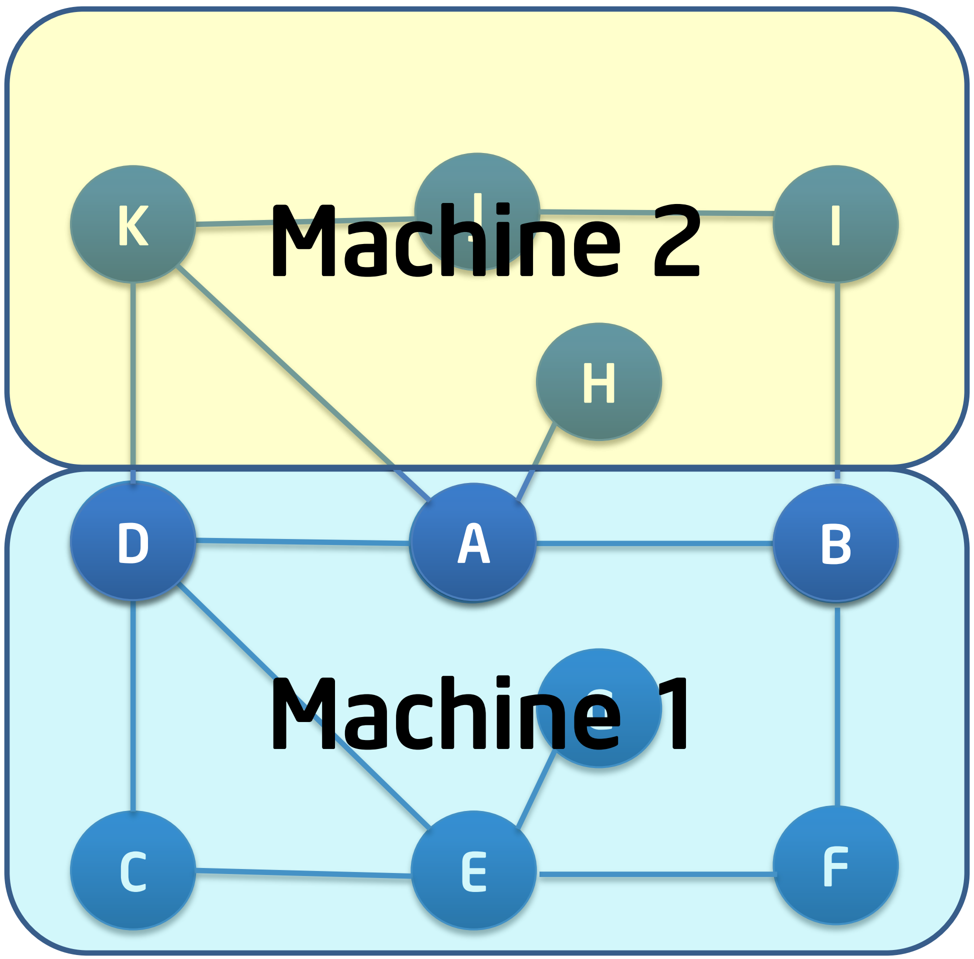
- Minimize communications by minimizing the number of machines *v* spans
- Place about the same number of *edges* on each machine



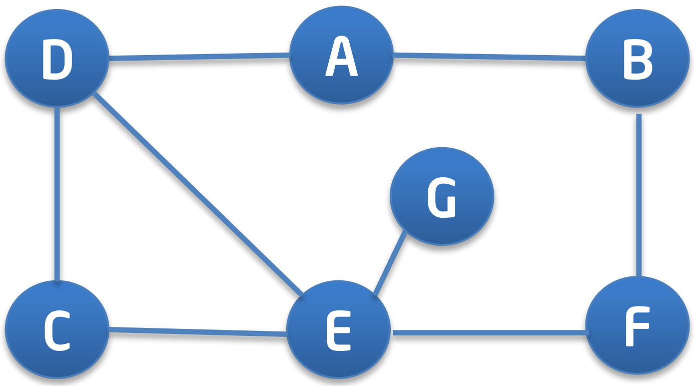
Heuristic-Based Partitioning Strategies

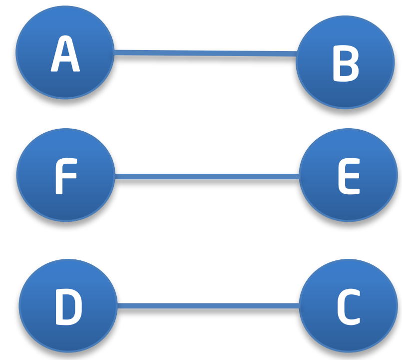
- Random edge placement: Edges are placed randomly by each system
- Greedy edge placement: Global coordination for edge placement to minimizes the vertex spanned
- Oblivious greedy placement: Implements a local version of the Greedy without global coordination

Oblivious Algorithm



Machine 1's Shard





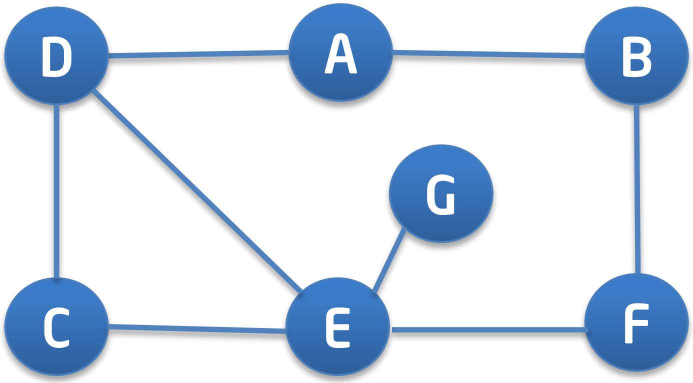
Partition 1

CASE 1:
Both end points have never been seen before

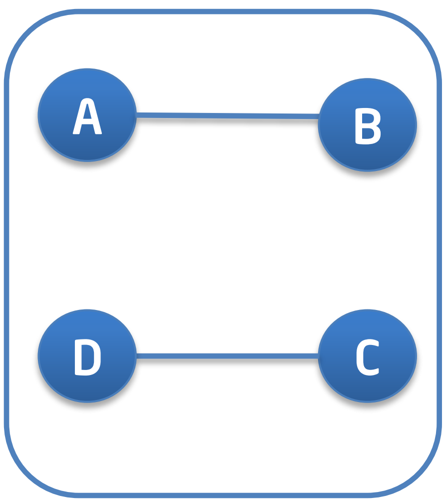
→ Randomly assign

Partition 2

Machine 1's Shard



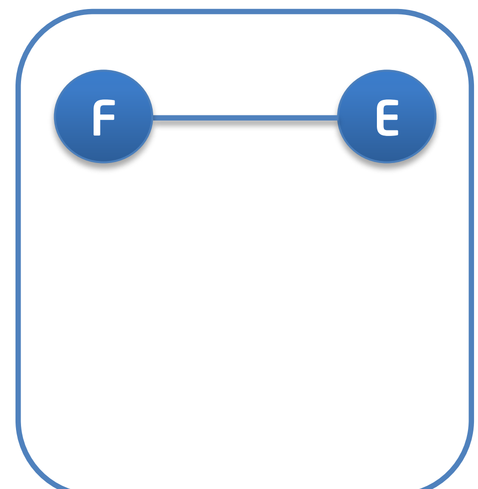
Partition 1



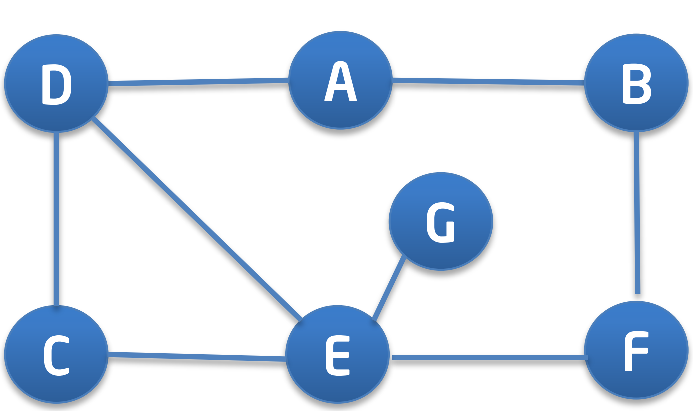
CASE 2:
Both end points have been seen before on the same partition

→ Assign to a partition which contains both endpoints

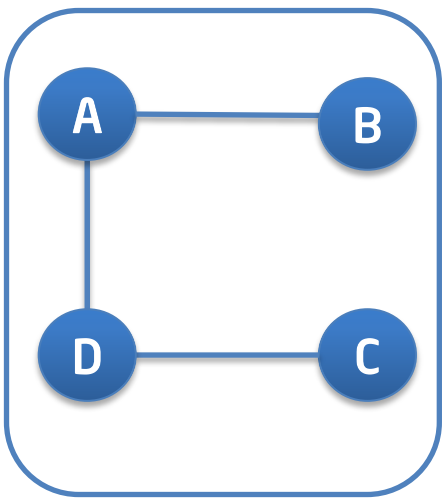
Partition 2



Machine 1's Shard



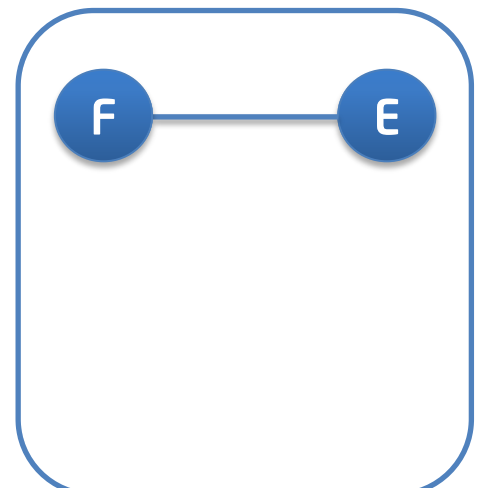
Partition 1



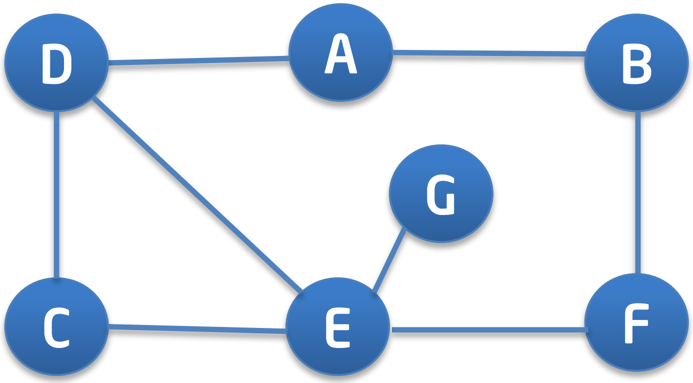
CASE 3:
Both end points have been seen before but on different partitions

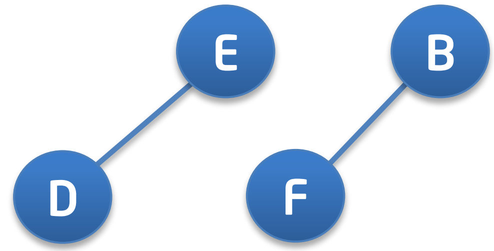
→ Assign to any partition that contains an endpoint

Partition 2

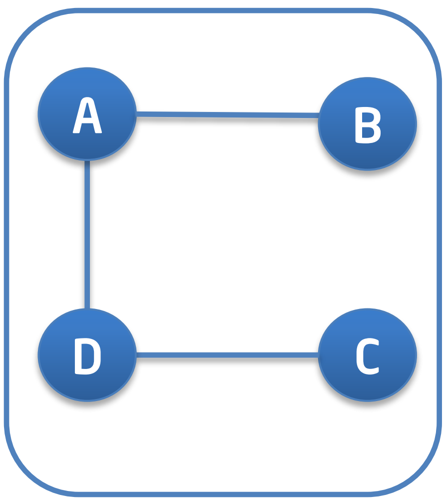


Machine 1's Shard





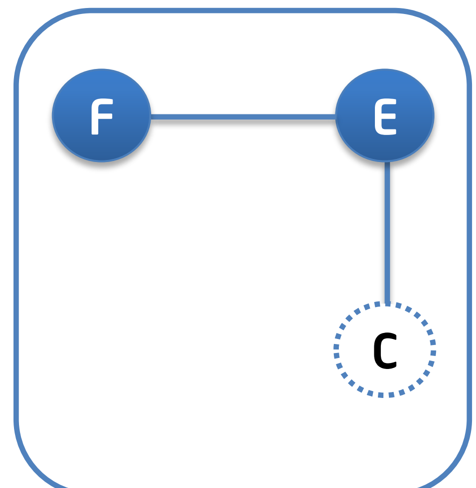
Partition 1



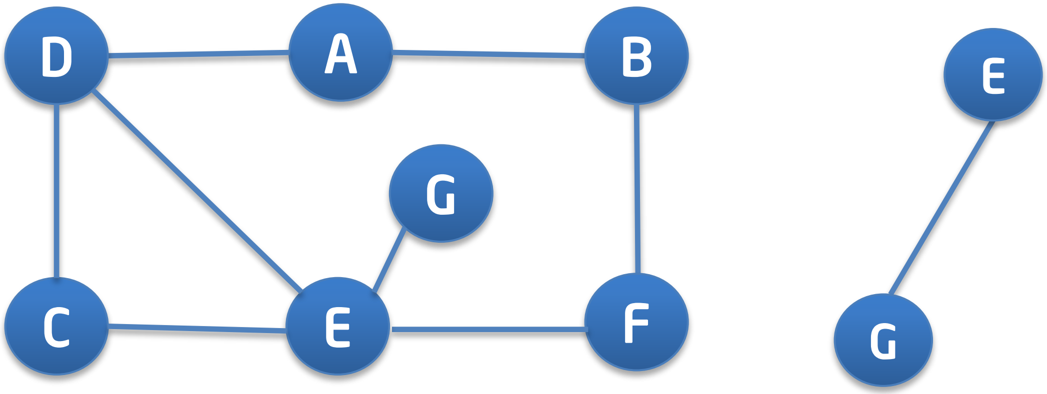
CASE 3:
Both end points have been seen before but on different partitions

→ Assign to any partition that contains an endpoint

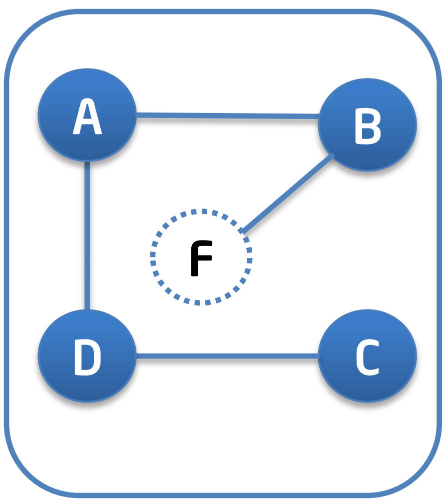
Partition 2



Machine 1's Shard



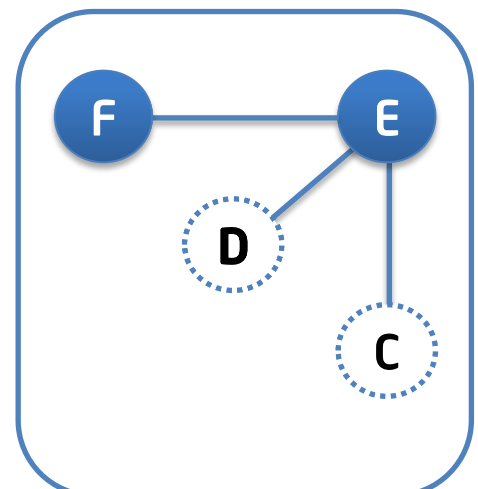
Partition1



CASE 4:
Only one end point has been seen before

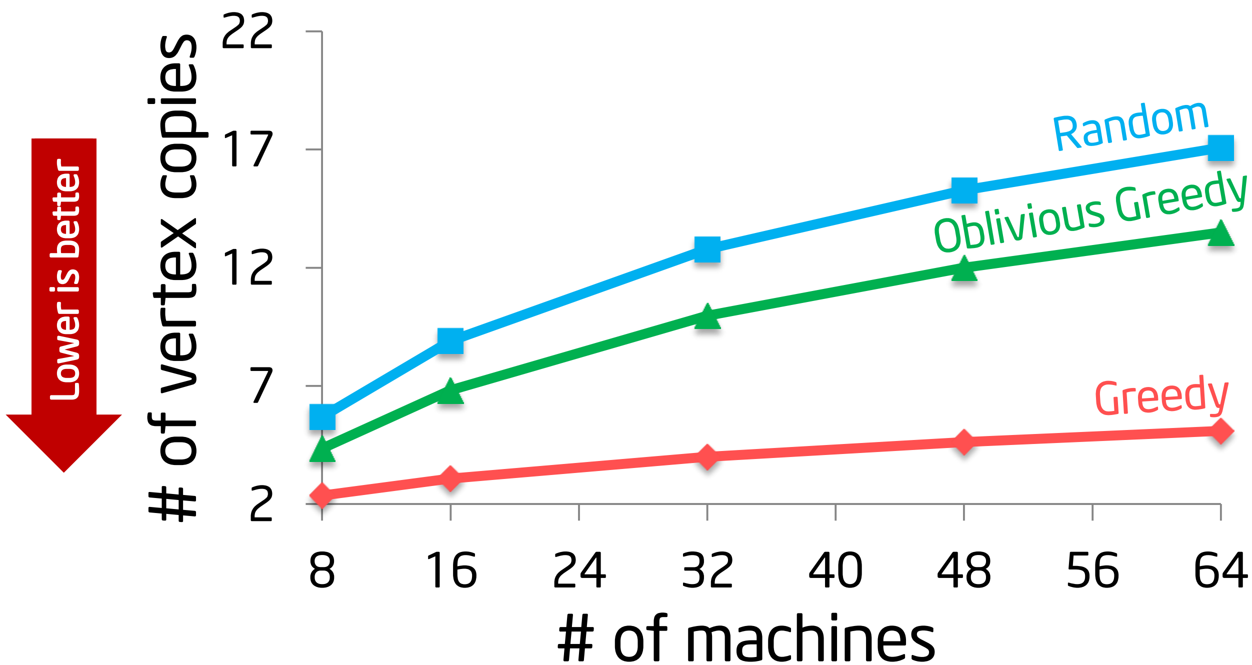
→ Assign to a partition that contains the endpoint

Partition 2



Partitioning Quality

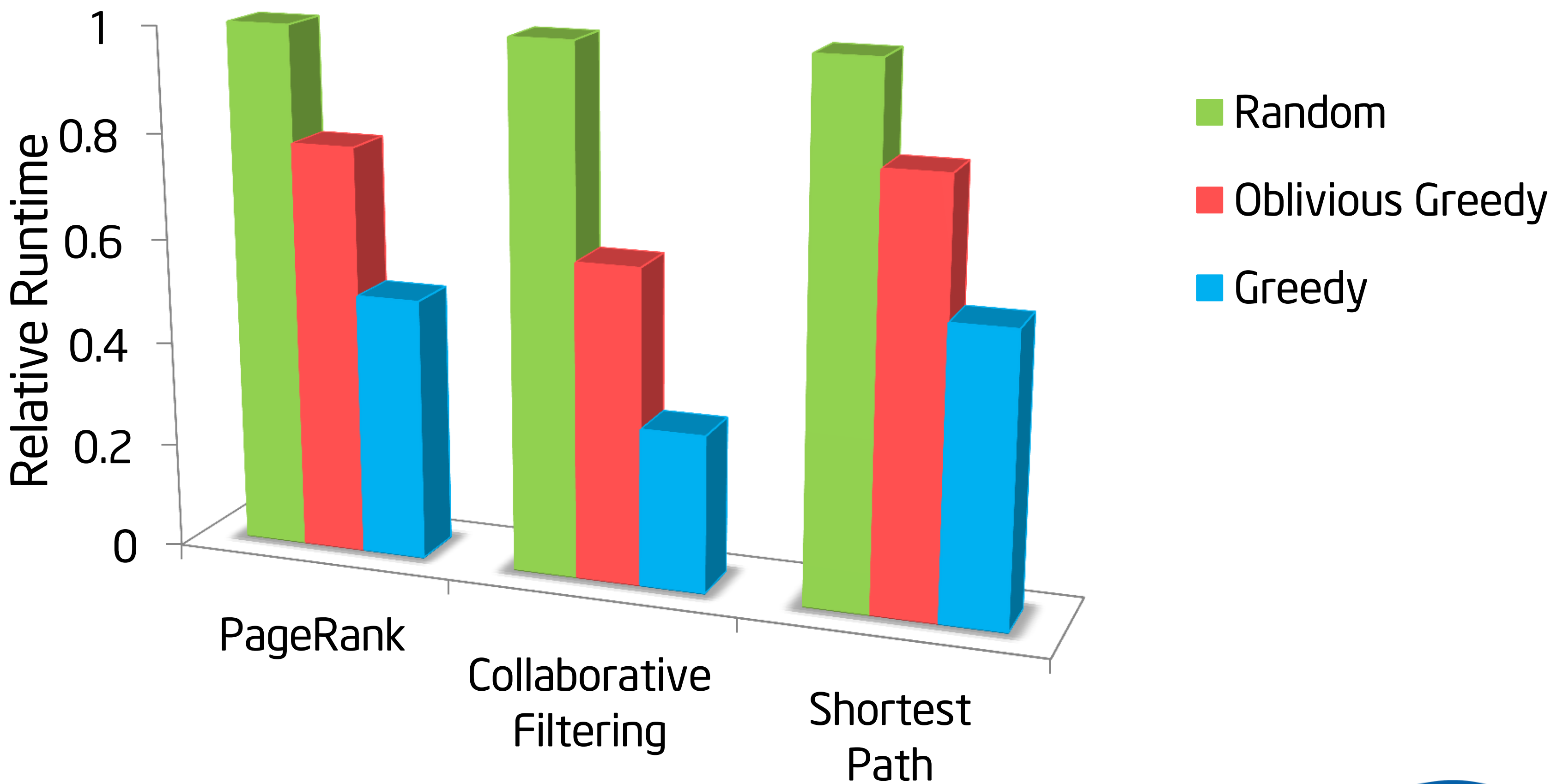
Twitter Graph: 41M vertices, 1.4B edges



Greedy yields a quality cut and the best performance….

*Gonzalez et al., "PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs," [OSDI'12]

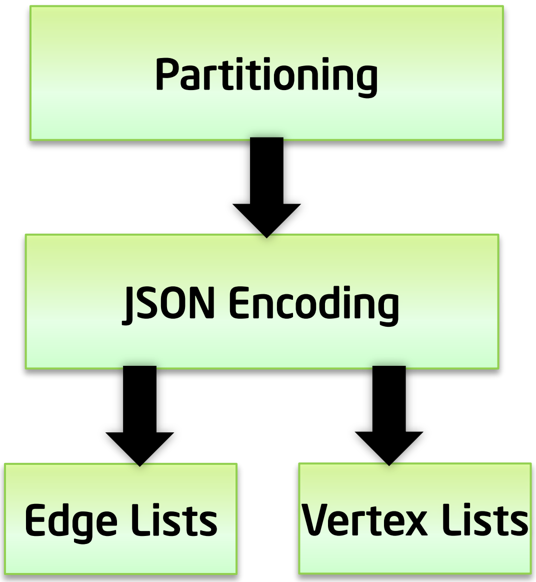
Performance for Partitioning



Performance is inversely proportional to replication.

*Gonzalez et al., “PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs,” [OSDI’12]

Load - Graph Serialization

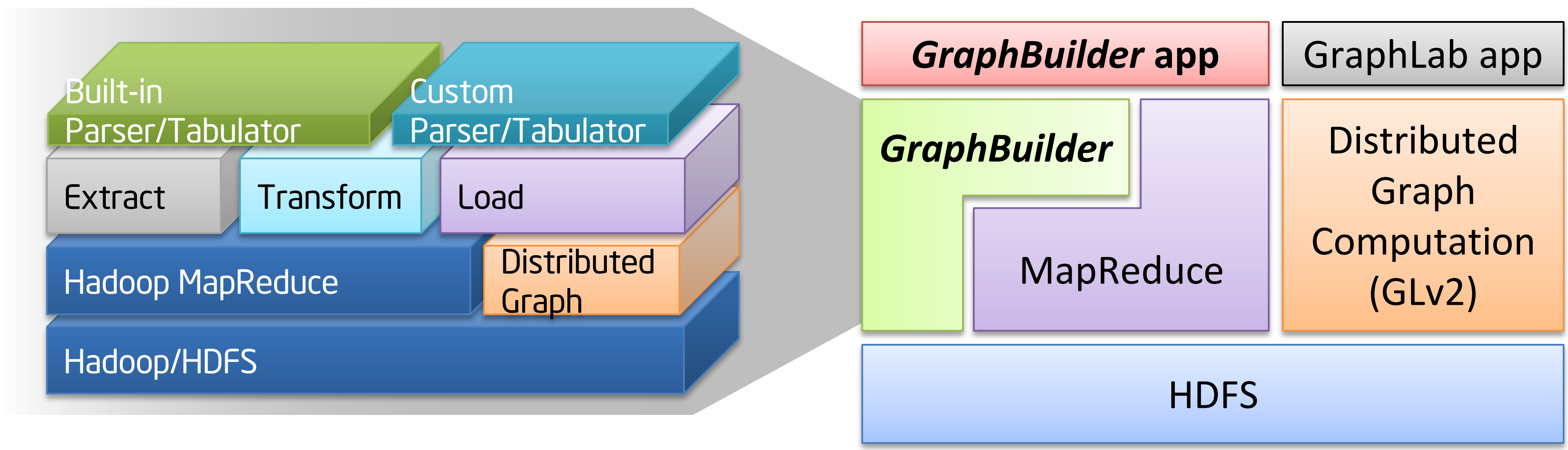



```
{
    "src_id": 34,
    "dest_id": 45
    "e-data": 30
}
```

```
{
    "ver_id": 34,
    "v-data": 56,
    "mirror": [1,2,3],
    "owner": 1
}
```

- Self-describing data format
  - JSON +/- compression
- Extensible
  - Easy to connect with Graph Databases
  - Plug-in Graph Visualizers

GraphBuilder Stack

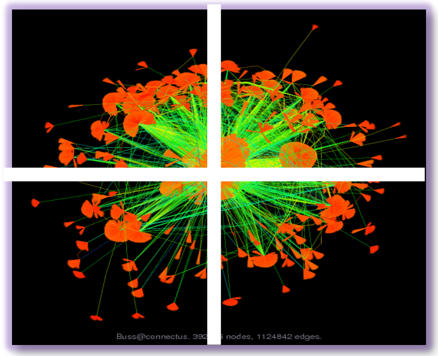


GraphBuilder Demo

WIKIPEDIA

GraphBuilder

Partitioned Bipartite graph



Latent Dirichlet Allocation (LDA) Algorithm

GraphLab

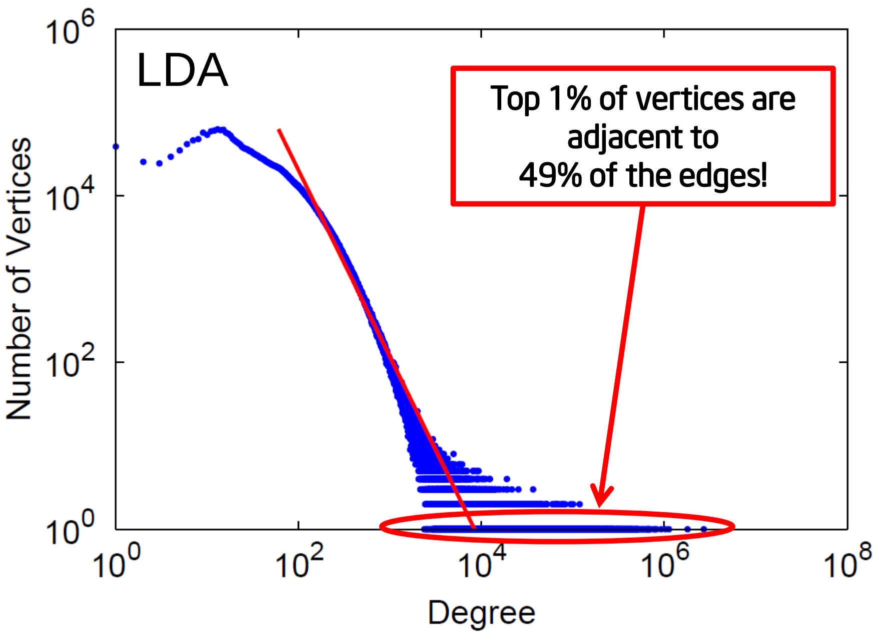
Knowledge Extraction

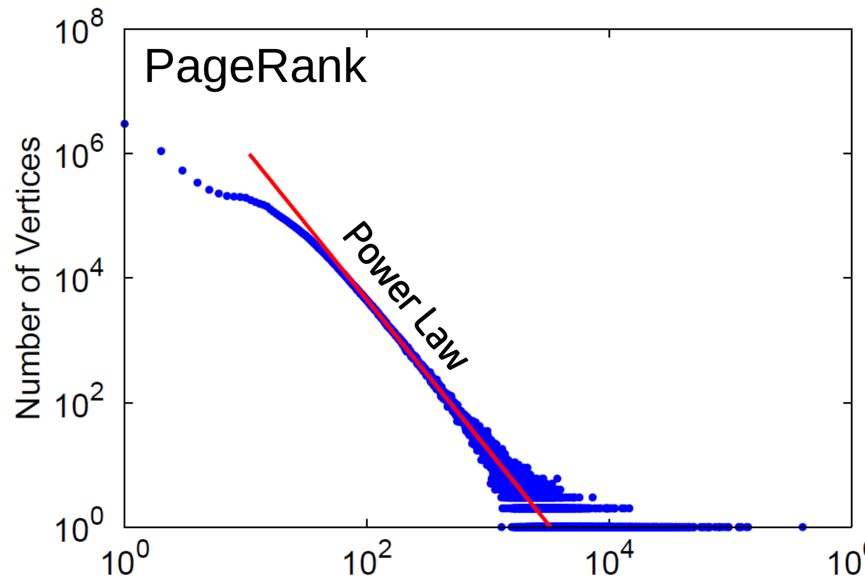
WordCloud Visualizer

NTopics	NWords	NDocs	NTokens	Alpha	Beta
50	296248	4003417	824408165	0.5	0.1

Topic Modeling

Our Wikipedia Graphs





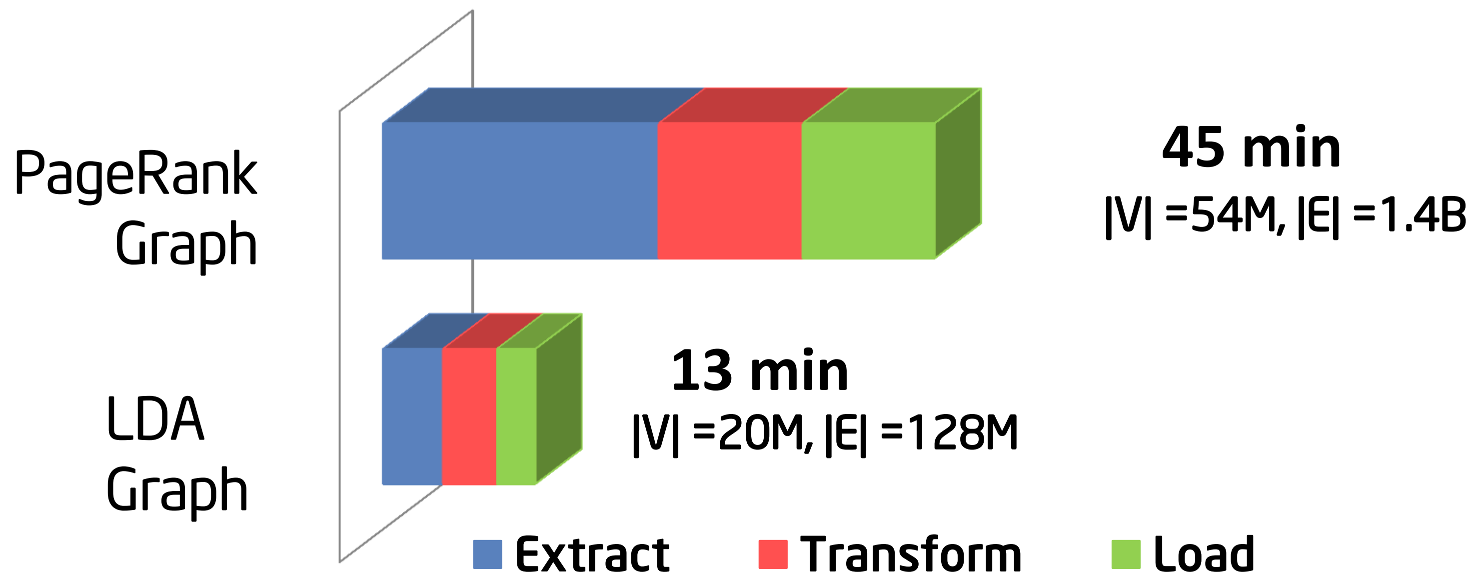
Graph	|V|	|E|	α
LDA	4.9M	478M	2.23
PageRank	9.7M	107M	2.41

Prototype Overview

- Hardware: 8 node cluster
  - 1U Dual CPU (Intel SNB) Amazon build ZT systems
  - 64 GB Memory, Four SATA Hard Drives
  - Intel 10G Adapter and Switch
- Software:
  - Apache Hadoop 1.0.1
  - GraphLab v2.1
  - *GraphBuilder* beta

Preliminary Results

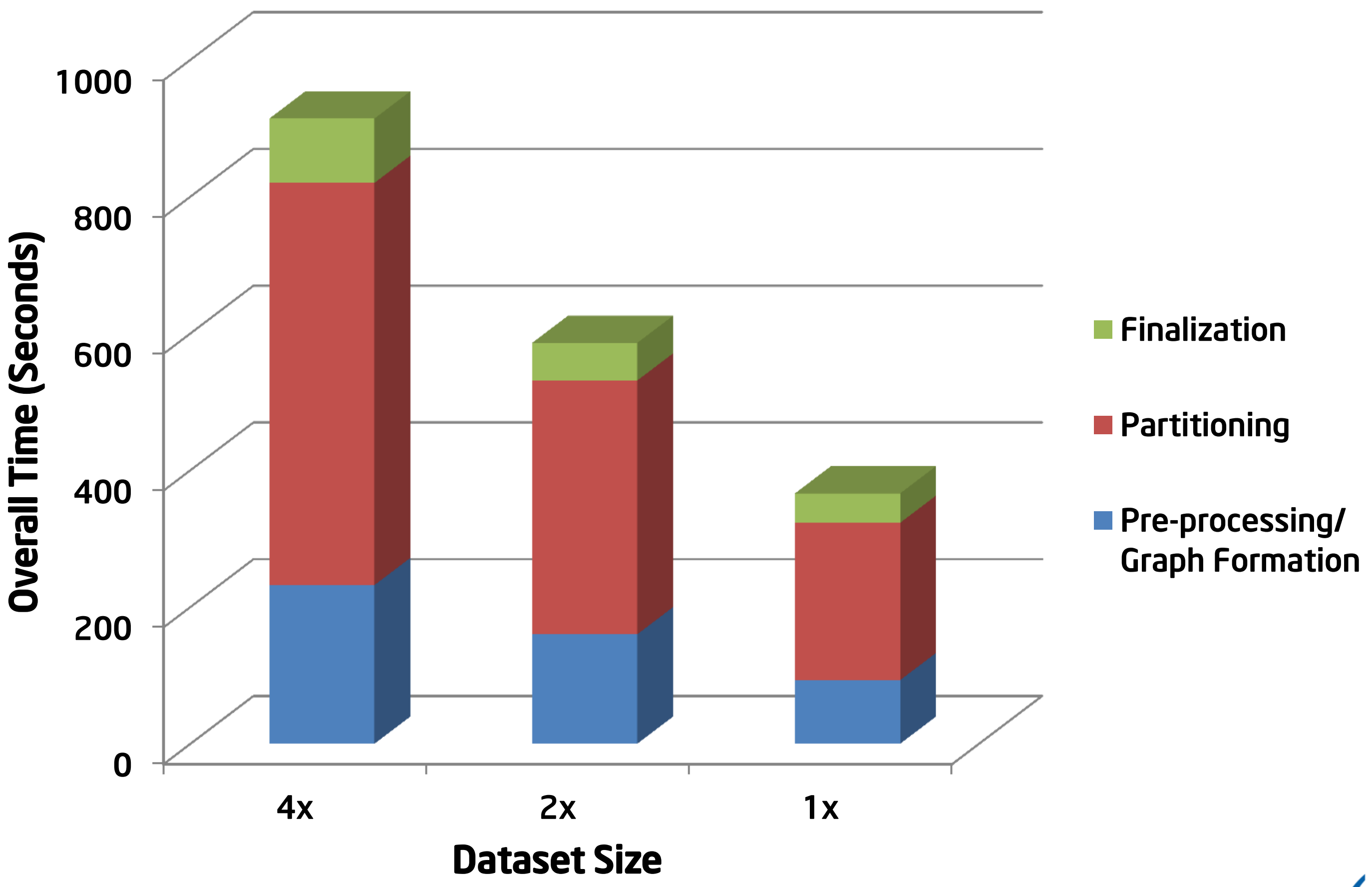
Graph	Custom plug-in code	Graph Compression	Partitioning Improvement (vs. Random)
PageRank	100 lines	60%	17%
LDA	130 lines	5%	32%



Hardware: 8 node cluster
1U Dual CPU (Intel SNB) Amazon build ZT systems
64 GB Memory, Four SATA Hard Drives
Intel 10G Adapter and Switch

Software:
Apache Hadoop 1.0.1
GraphLab v2.1
GraphBuilder beta

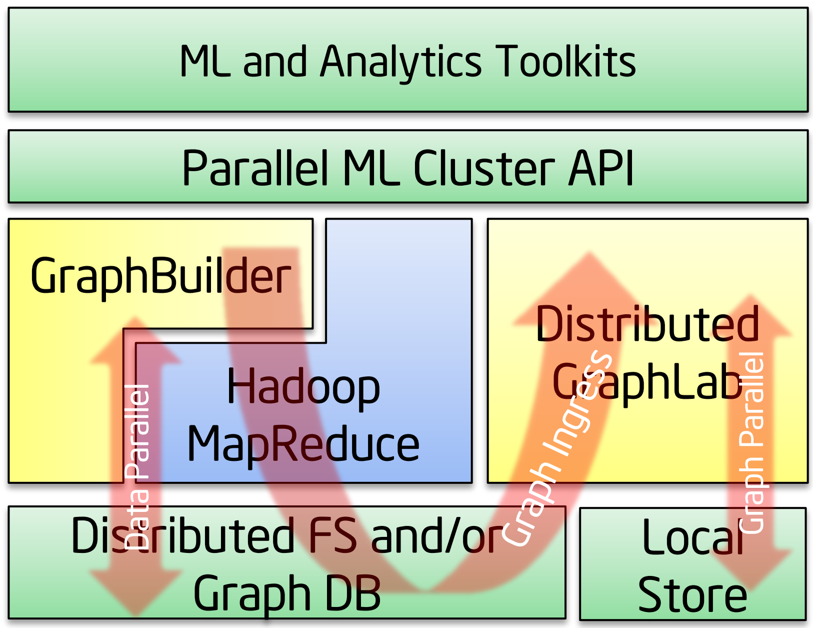
Scaling Experiment



Collaboration ahead!

All Together Now

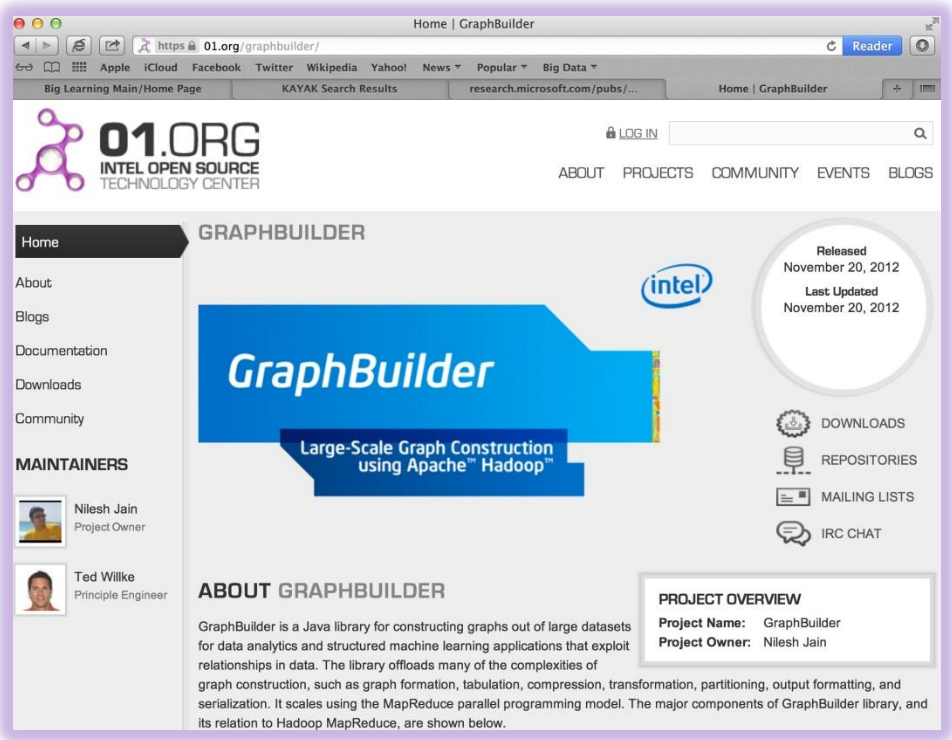
Parallel Machine Learning Pipeline



Future areas for ISTC collaboration:

1. Improve usability and data wrangling
2. Research GL fault tolerance and local storage support
3. Advance GB + GL for streaming and time-evolving apps

Launches today!



Intel open source portal at http://www.01.org

GraphLab2 at http://graphlab.org

Both under Apache 2.0 licensing.

Looking for research partners and committers.

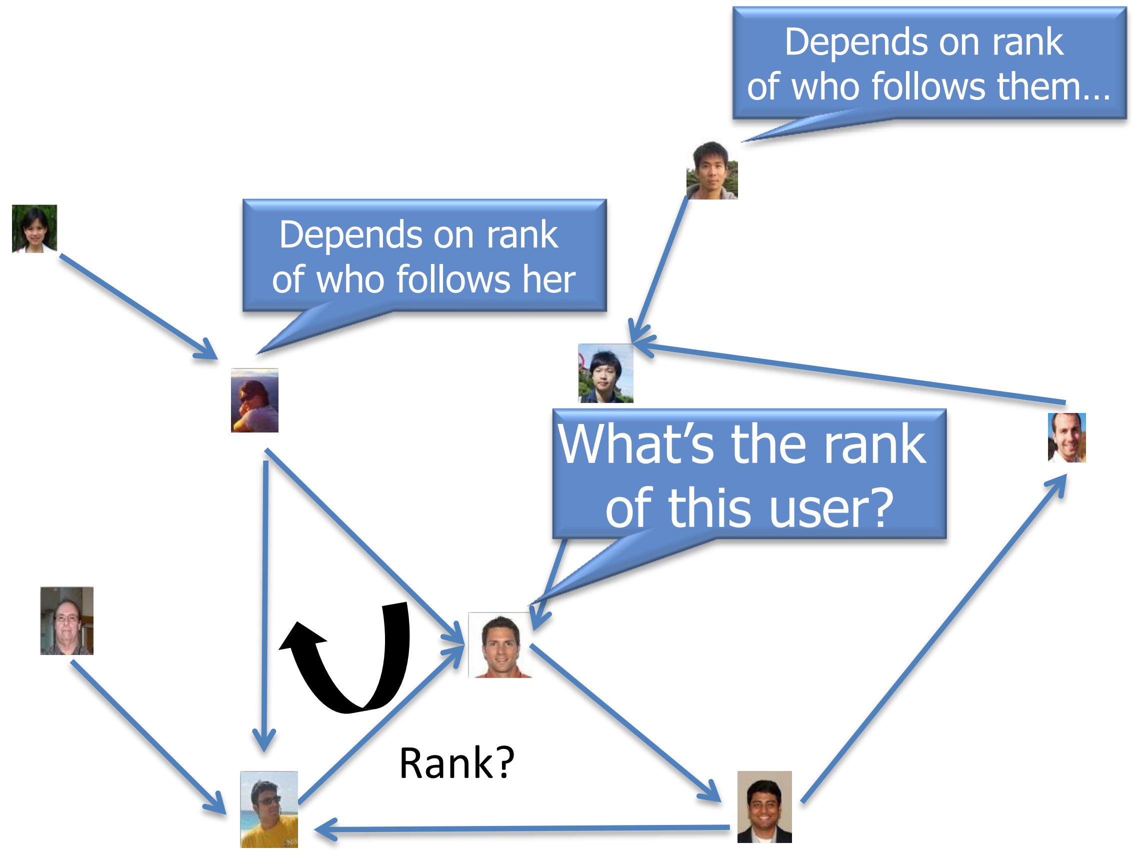
Contacts:

nilesh.jain@intel.com
theodore.l.willke@intel.com

intel®

Leap ahead™

How many people are pointing to you and what's their relative importance?

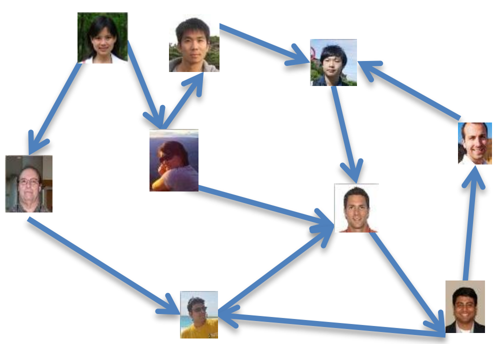


Loops in graph - Must iterate!

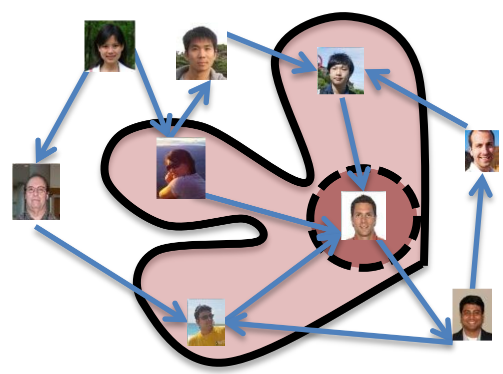
Graphics source: [Joseph Gonzalez (CMU)]

Properties of Graph-Structured Computation

Dependency Graph



Local Update



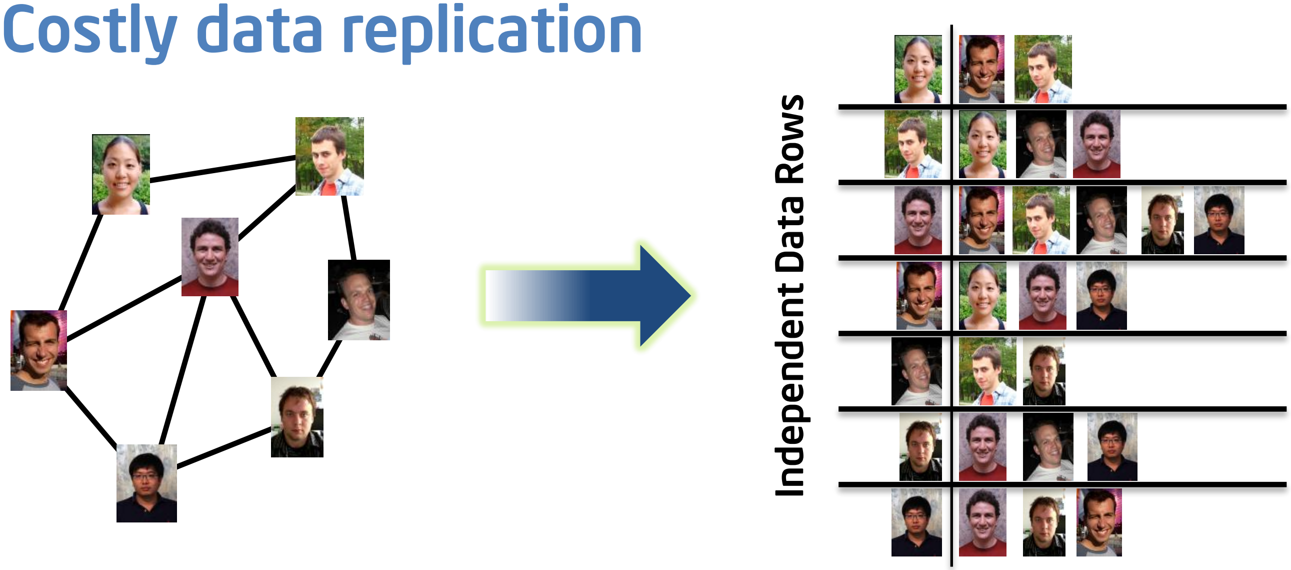
Iterative Computations

Similar properties for many other problems!

Graphics source: [Joseph Gonzalez (CMU)]

Data Dependencies

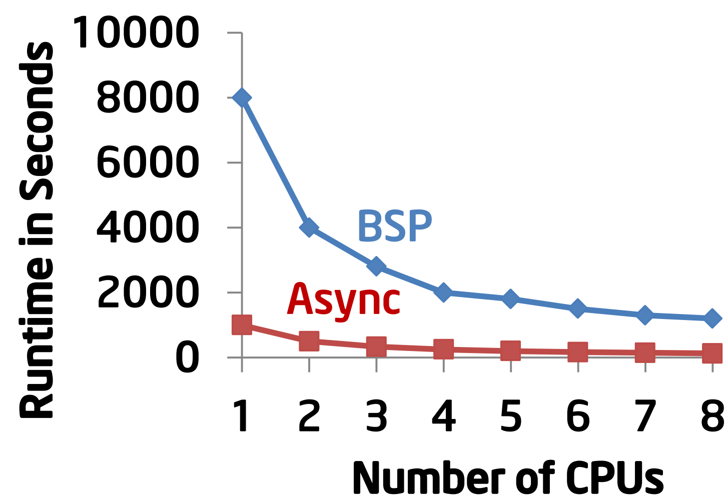
- MapReduce does not efficiently express data dependencies
  - User must code substantial data transformations
  - Costly data replication



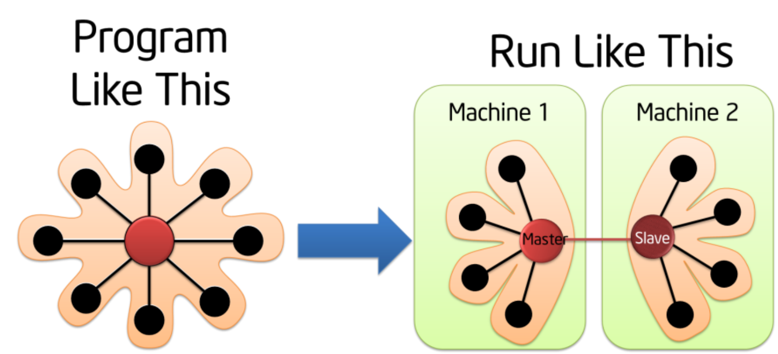
- MapReduce does not efficiently express iterative algorithms

Approaches to Graph-Structured Computation

- **Bulk Synchronous Processing (BSP)**
  - Giraph on Hadoop (Inspired by Google Pregel)
  - Dryad (Microsoft Research)
  - Apache Hama on Hadoop (Twitter)



- **Asynchronous Graph-Parallel**
  - Galois (UT Austin) → Edge partitioning
  - **GraphLab (CMU) → Vertex partitioning**



Split **High-Degree** Vertices

GraphLab has an edge.

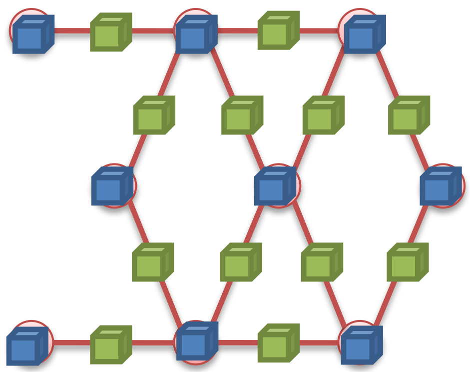
GraphLab Goals

- Designed specifically for ML
  - Graph dependencies
  - Iterative
  - Asynchronous
  - Dynamic
- Simplifies design of parallel programs
  - Abstracts away hardware issues
  - Automatic data synchronization
  - Addresses multiple hardware architectures

The GraphLab Framework

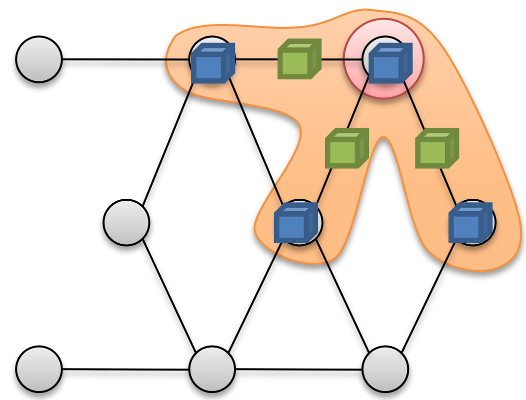
Graph Based

Data Representation

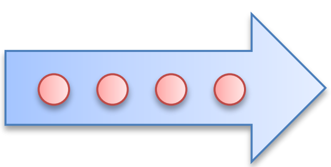


Update Functions

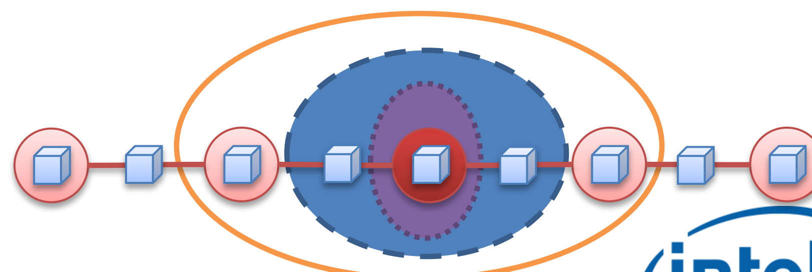
User Computation



Scheduler

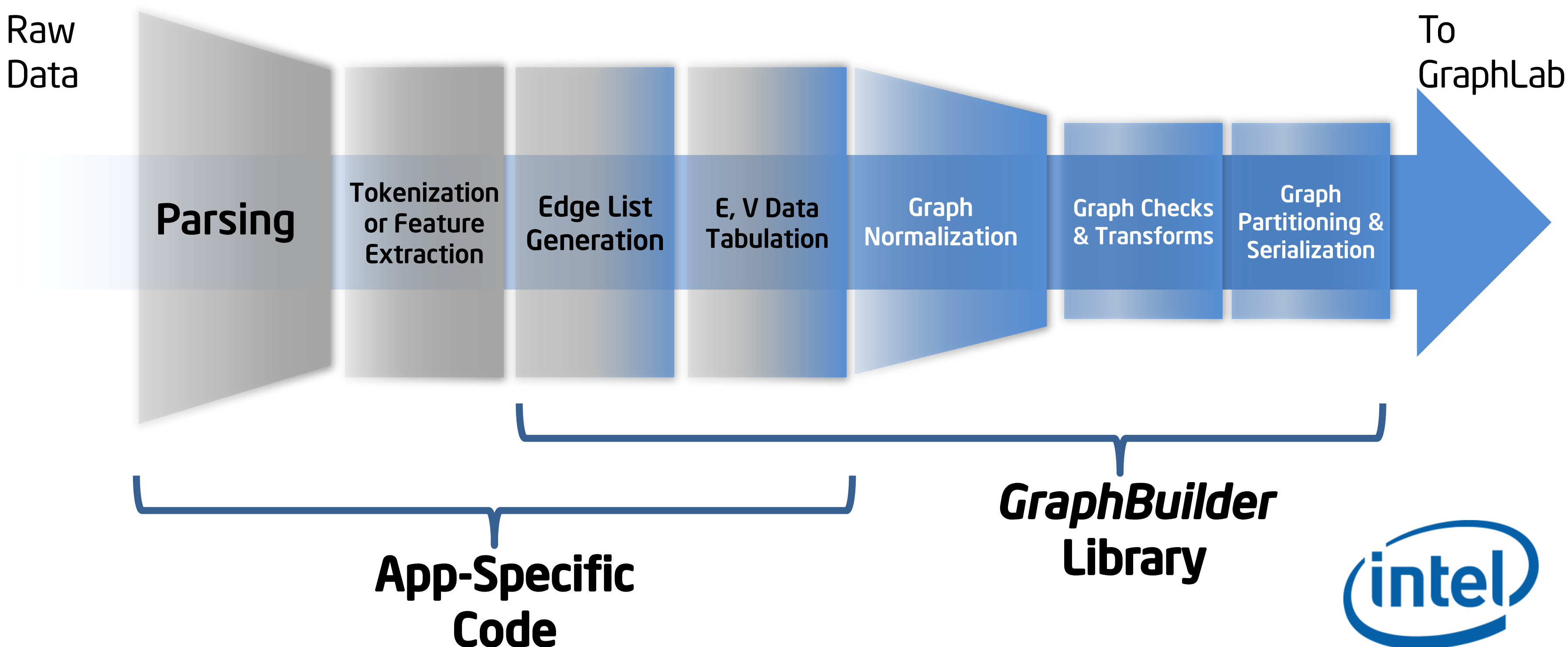


Consistency Model



GraphBuilder makes it easy.

- Fills a hole in the ecosystem
- Written in Java for easy use in Hadoop MapReduce and applications
- Offloads domain expertise



Prototype Overview

- Hardware: 8 node cluster
  - 1U Dual CPU (Intel SNB) Amazon build ZT systems
  - 64 GB Memory, Four SATA Hard Drives
  - Intel 10G Adapter and Switch
- Software:
  - Apache Hadoop 1.0.1
  - GraphLab v2.1
  - *GraphBuilder* beta

Example Topics Discovered from Wikipedia

party law government election court president elected council general minister political national members committee united office federal member house parliament vote public elections democratic held rights candidate congress senate district seat constitution secretary republican campaign commission supreme votes conservative parties bill police

york county american united city washington john texas served virginia pennsylvania war moved ohio chicago william carolina north florida illinois george james died massachusetts president named jersey born boston south union west company georgia smith began michigan fort years philadelphia white missouri maryland america black tennessee

century king roman empire greek bc ancient emperor ii kingdom period battle city time great war ad early reign kings iii son rule power greece army centuries dynasty rome modern history imperial medieval death ottoman years led byzantine defeated ruled year throne athens capital castle military late iv middle control

engine car design model cars production built engines vehicle class models speed vehicles designed produced power front system version type series motor rear standard gun company introduced range ford sold fuel drive wheel tank fitted factory machine developed based replaced wheels time powered small high weight electric body mounted early

war army military forces battle force british command general navy ship division ships troops corps service naval regiment commander infantry attack men officer fleet soldiers units officers operations unit june august brigade july fire training march battalion april operation captain september three enemy united october sea royal german marine major

son died married family king daughter john death william father born wife royal ireland irish henry house lord charles sir prince brother children england queen duke thomas years marriage george earl edward english second elizabeth sons mary james mother appointed year dublin lady title great succeeded robert ii member castle

season team game league games played coach football record teams baseball field year second career play basketball hockey three yards won bowl points win series player head conference championship seasons players draft high time named national led nfl third major finished stadium division lead playing ncaa history runs touchdown signed

species family birds small long large animals bird plants genus plant natural habitat tree fish tropical white black order leaves brown common forests trees animal flowers eggs worldwide feed occur subtropical wild length male breeding habitats range food female fruit short insects endemic forest group including include moist threatened tail

art museum work works artists collection design arts painting artist gallery paintings exhibition style fine including painted architecture york fashion painter life early created sculpture artistic history contemporary collections years museums worked images time photography figures academy exhibitions modern portrait photographs began studio drawing include exhibited produced designed period visual

white red black blue called color will head green gold side small hand long arms top flag horse wear silver common light dog wood body type large yellow form worn dogs cut popular left generally traditional ball front horses shape hair feet colors time coat three typically modern face cross

school students university high college schools education year program student campus community programs training center members science national years public academic association courses arts educational include class institute department teachers colleges classes offers activities universities district engineering learning founded faculty girls sports children boys international board teaching academy secondary established

album band song released music songs single records recorded rock bands release live tour video record albums label group recording guitar track cover version tracks number featured time chart hit uk top performed studio played singles sound love pop artist solo cd debut singer artists members included early second bass

radio station news television channel broadcast stations network media tv broadcasting time format local program bbc programming live fm morning host began sports fox air cable call hosted coverage music pm sunday daily channels digital abc aired changed current launched communications programme day broadcasts moved cbs years saturday talk night

age 18 population income average years median living 65 males females households 100 family people families older town size city household miles density american township total area county races census 2000 square 45 25 64 children 24 44 white female land including units housing bureau individuals located poverty united village

music musical opera festival orchestra dance performed jazz piano theatre performance works concert symphony composer played performances instruments musicians classical including work composed major singing songs folk instrument ballet composition composers play performing concerts playing stage years include popular choir ensemble sound style time violin hall piece chamber recordings string