Graph Processing and applications

Xuemin Lin
DBG
UNSW, Australia

Outline

- Introduction of Graphs and Classic Problems
- Advanced Applications in Big Graphs
- Scalability
- Wrap-up

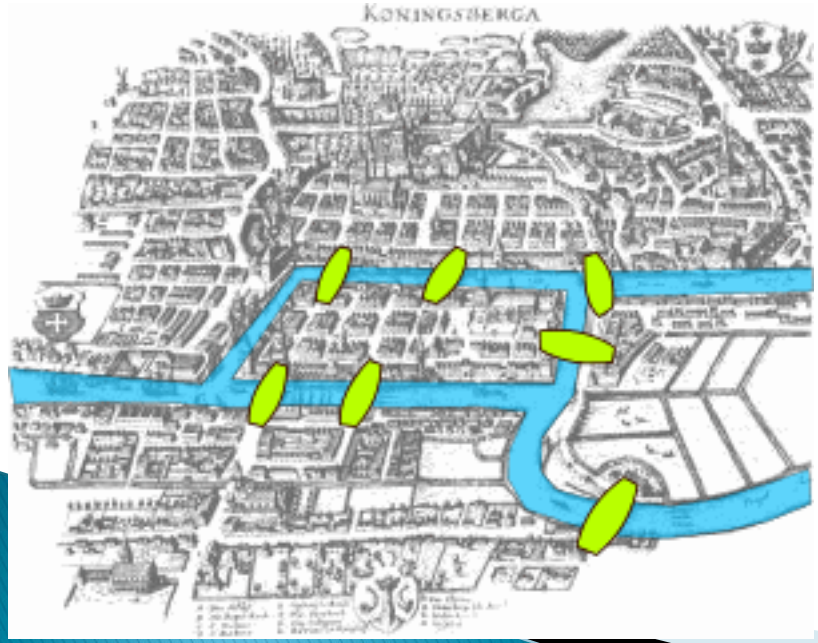
Introduction of

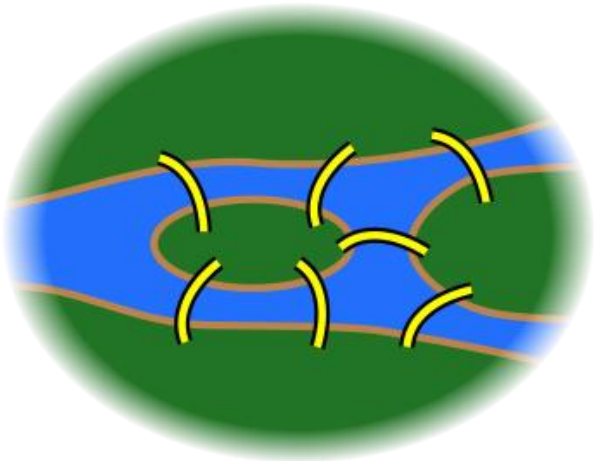
- Graphs and Classic Problems

1. Seven Bridges of Königsberg

Leonhard Euler in 1735:

- find a roundtrip through the city to cross each bridge once and only once
- earliest (published) work on networks/graphs





1. Seven Bridges of Königsberg(cont.)

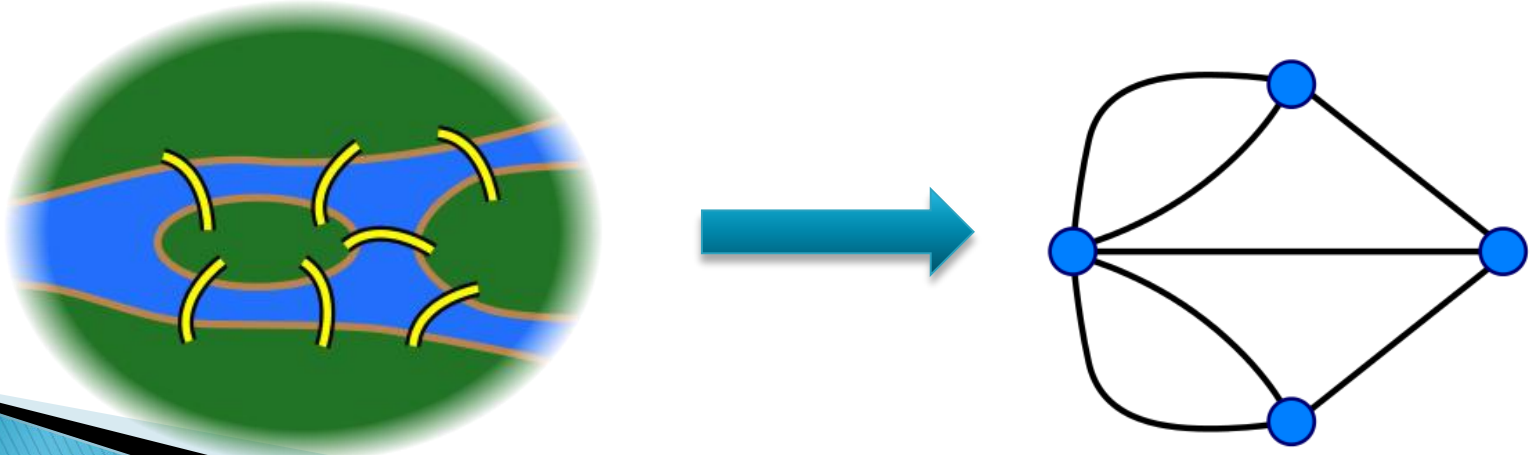
Abstraction:

- replaced each land mass with an abstract "vertex" or node, and each bridge with an abstract connection, an "edge"

Showed that this problem has no solution.

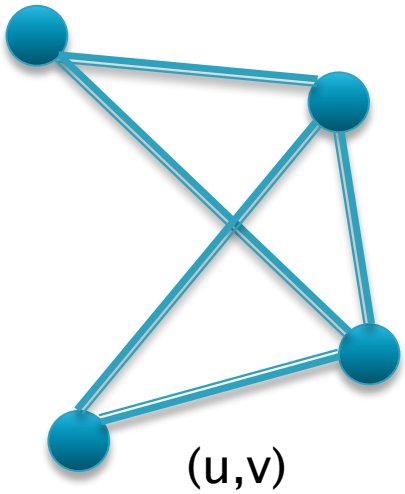
Euler Tour:

- necessary and sufficient conditions for the walk of the desired form: connected and all vertices with even degrees.



Graphs

- Definition:
  - A Graph is a collection of nodes joined by edges.



- G=(V,E)
  - V is a set of vertices, u, v in V are vertices
  - E is a set of edges, (u,v) in E is an edge.

Edges

- Undirected edges
  - u–v: (u,v)=(v,u)
  - e.g., u and v like each other; u and v are co-authors
- Directed edges(arcs)
  - u→v: (u, v) != (v, u)
  - (u, v) is an *in-link (in-edge)* of v, and an *out-link (out-edge)* of u.
  - e.g., u likes v; u is the father of v.
- Other attributes
  - weight (e.g. frequency of communication) – w(u, v)
  - rank (e.g., best friend, second best friend, ...)
  - type (e.g., friend, co-worker, activates, inhibits, ...)

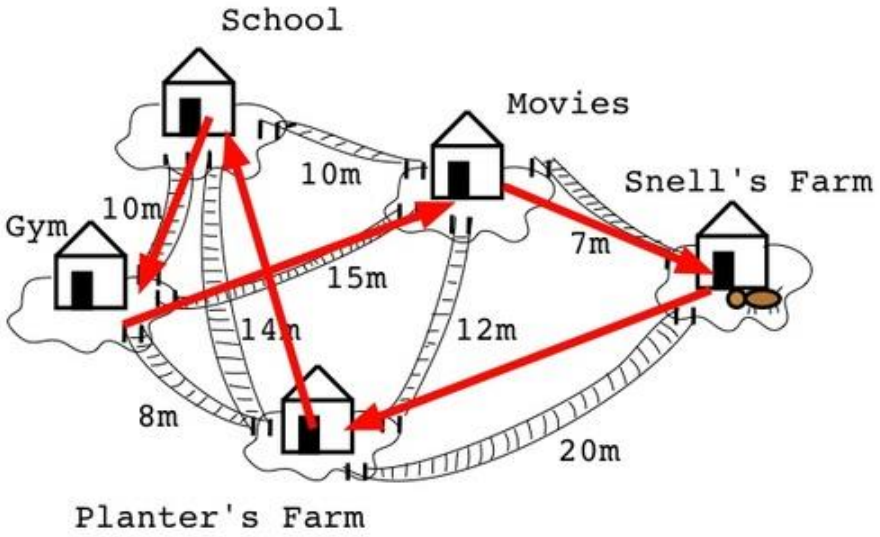
Vertices

- Degree
  - d(v): the number of edges connected to a vertex
  - In degree: number of incoming edges of a vertex
  - Out degree: number of outgoing edges of a vertex
- If edges are weighted, the *degree*, *in degree*, *out degree* are the total weights of *edges*, *incoming edges*, *outgoing edges* of a vertex.
- Other attributes:
  - Weight: (e.g., importance, authority of a person)
    - Type: (e.g., advisor, student, old people)
    - Label: (e.g., name)

2. Travelling salesman

- Problem:
  - Given a list of cities and the distances between each pair of cities, find the shortest route that visits each city exactly once and returns to the origin city.
  - In the theory of computational complexity, the TSP belongs to the class of NP-hard problems.

Thus, it is unlikely that the worst-case running time for any algorithm for the TSP is in PTIME.



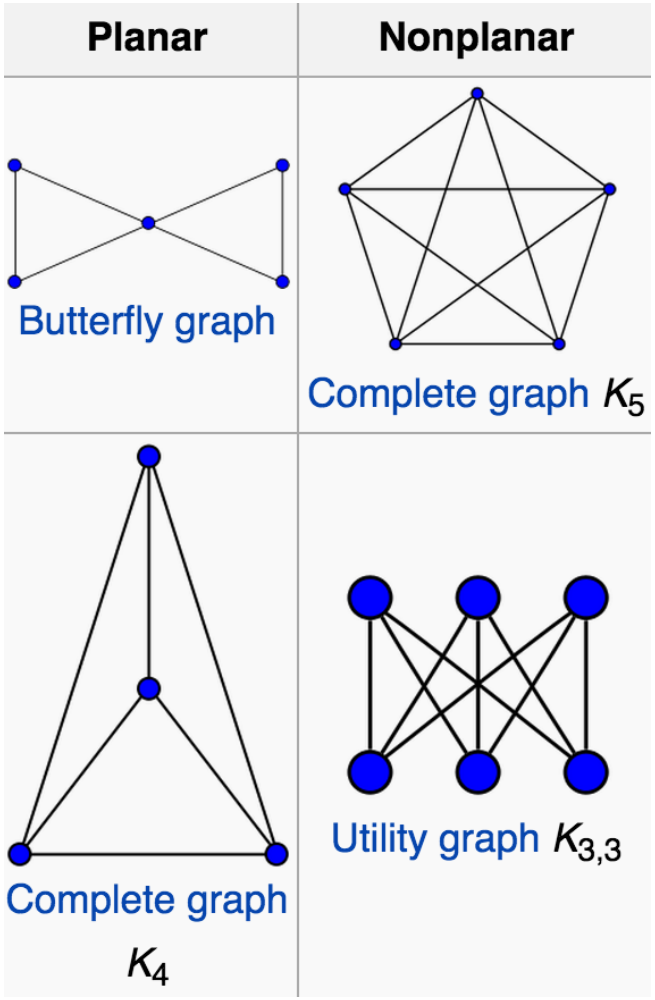
2. Travelling salesman (cont.)

- Heuristic Solutions:
  - Constructive heuristics: Nearest-neighbors (greedy), spanning tree, bitonic tour, etc.
  - for extremely large problems (millions of cities), within a reasonable time which are with a high probability just 2-3% away from the optimal solution

Widely used in several applications:
planning, logistics
the manufacture of microchips.
DNA sequencing, etc.

3. Planar graph

- Definition: can be embedded in the plane
  - i.e., it can be drawn on the plane in such a way that its edges intersect only at their endpoints.
- Determination:
  - does not contain a subgraph that is a subdivision of K5 (the complete graph on five vertices) or K3,3((complete bipartite graph on six vertices)

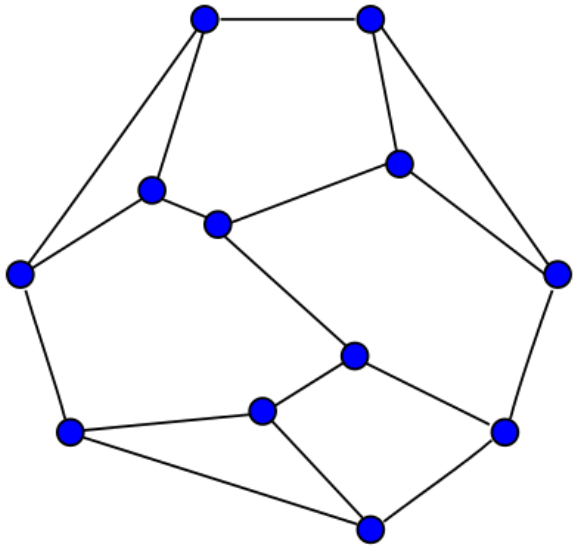


4. Euler's Formula

- Euler's Formula:

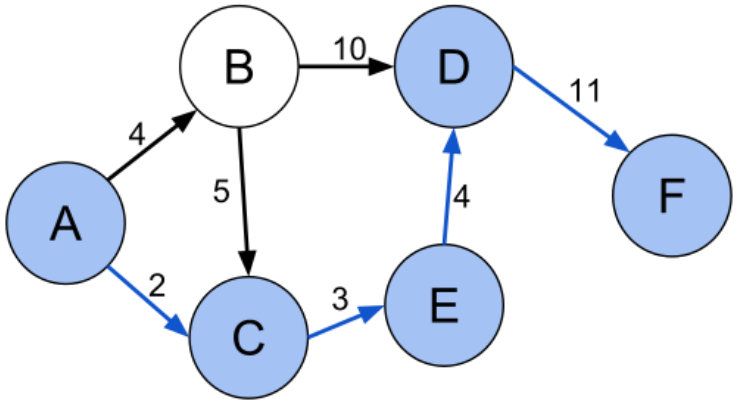
  if a finite, connected, planar graph is drawn in the plane without any edge intersections

  $v - e + f = 2$. (v is the number of vertices, e is the number of edges and f is the number of faces)



5. Shortest Path

- a.k.a. Geodesic paths:
  - finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized.
- Solution:
  - Single Source: Dijkstra
  - All Pair: Floyd-Warshall.



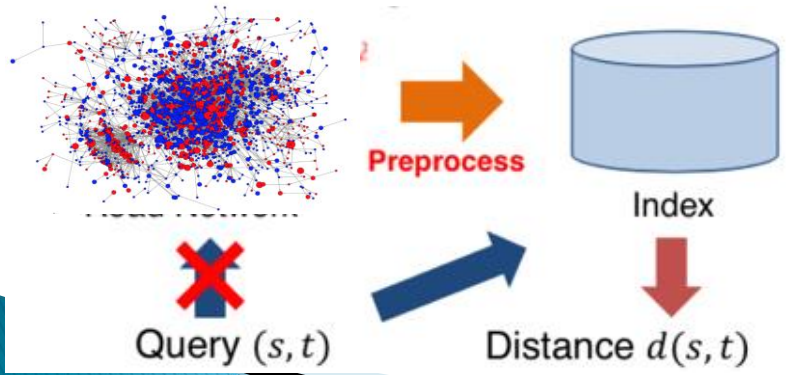
5. Shortest Path(cont.)

- Applications:
  - Road Network Navigation
  - n-Degree Separation in Social Networks

Six Degree of Separation
http://entitycube.research.microsoft.com/6dumapsvg.aspx

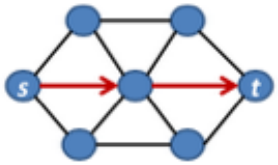
5. Shortest Path(cont.)

- How to fast get shortest path between two nodes in a large graph?
  - Pre-computed small structure to maintain distances.
  - Distance Oracle:
    - Hop cover for each nodes' reachable node list
      - Intersect of two lists.
    - Landmark based global summary structure.
      - Tree traversal.



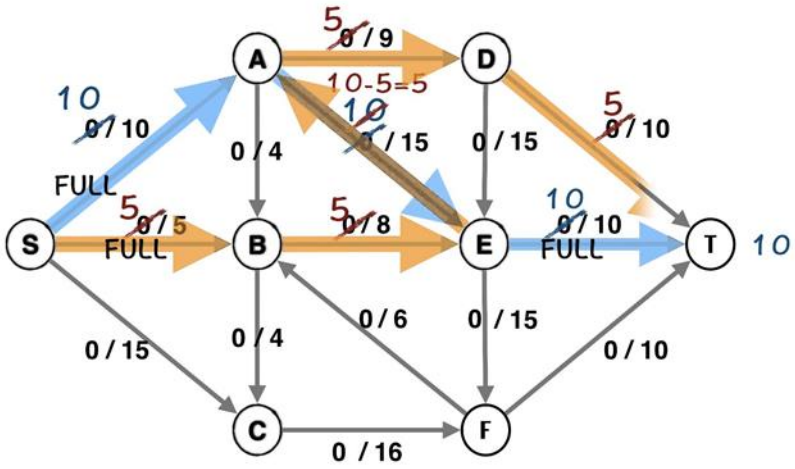
Given a graph $G = (V, E)$

1. construct an index to
2. answer distance $d_G(s,t)$



6. Max-Flow

- Problem:
  - finding a feasible flow through a single-source, single-sink flow network that is maximum.
- Applications:
  - Airline/pipeline/network scheduling
  - Circulation-demand/logistics planning
  - Social network Sybil/Spammer detection

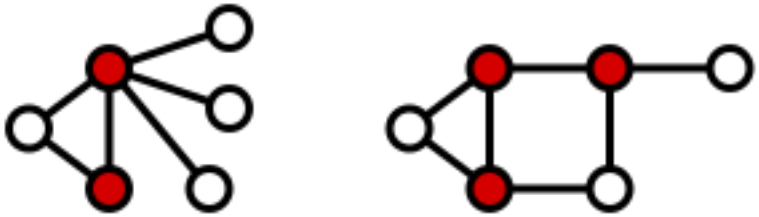


6. Max-Flow (cont.)

- Common Solutions:
  - Linear Programming.
  - Ford-Fulkerson: O(mn)
  - Recent result: close to O (m)

7. Vertex Cover

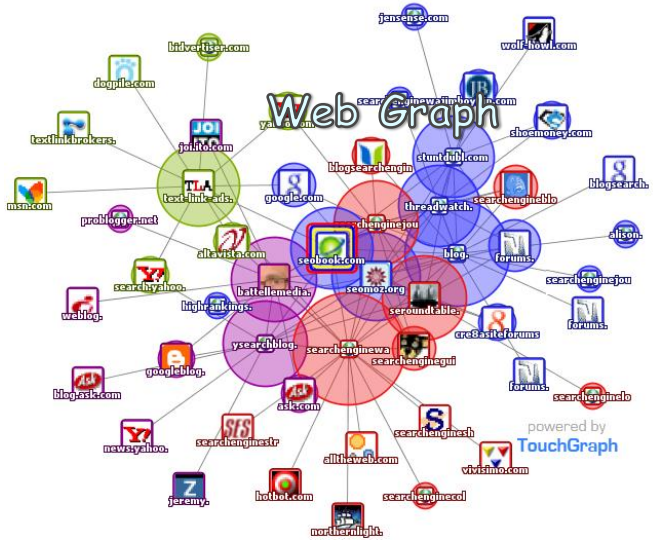
- Problem:
  - A vertex-cover of an undirected graph G=(V,E) is a subset V' of V, for edge (u,v) in G, u or v is in V'.
  - Minimum Vertex Cover:
    - NP-hard optimization problem.
- Solution:
  - Approximation algorithm:
    - Repeatedly taking both endpoints of an edge into the vertex cover and remove.
    - Maximum Matching

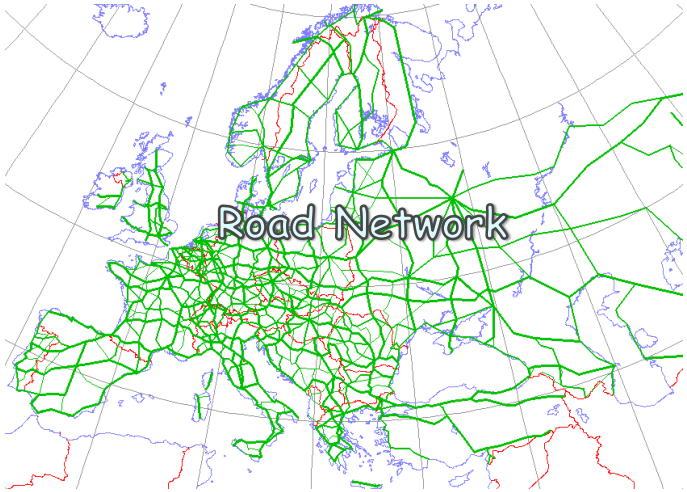


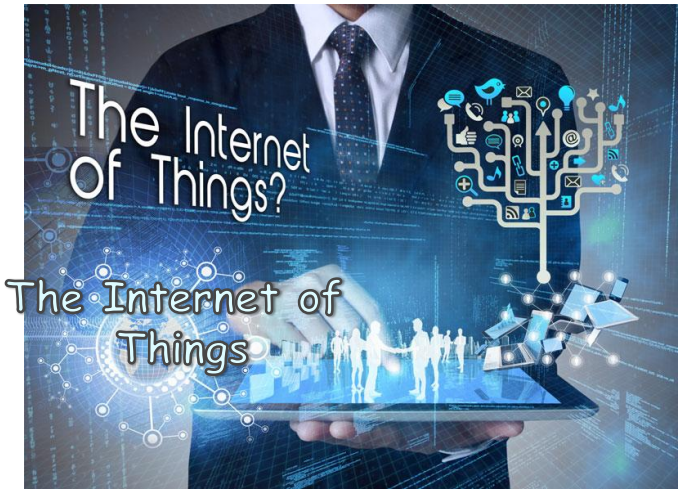
ADVANCED APPLICATIONS

- Big Graphs

Graphs are Everywhere!





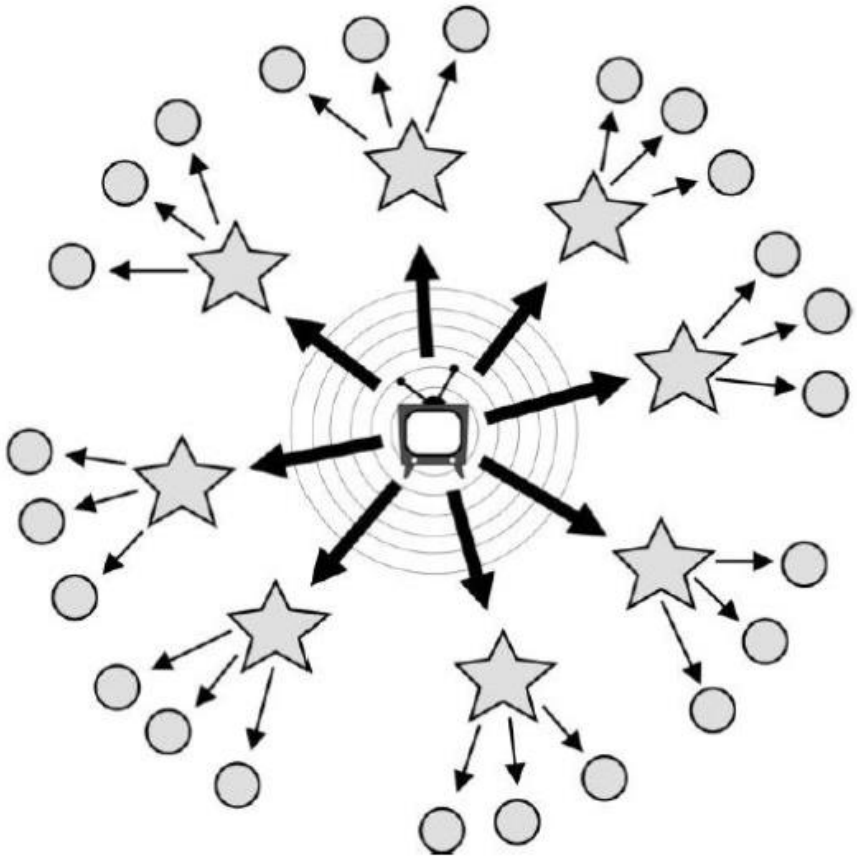


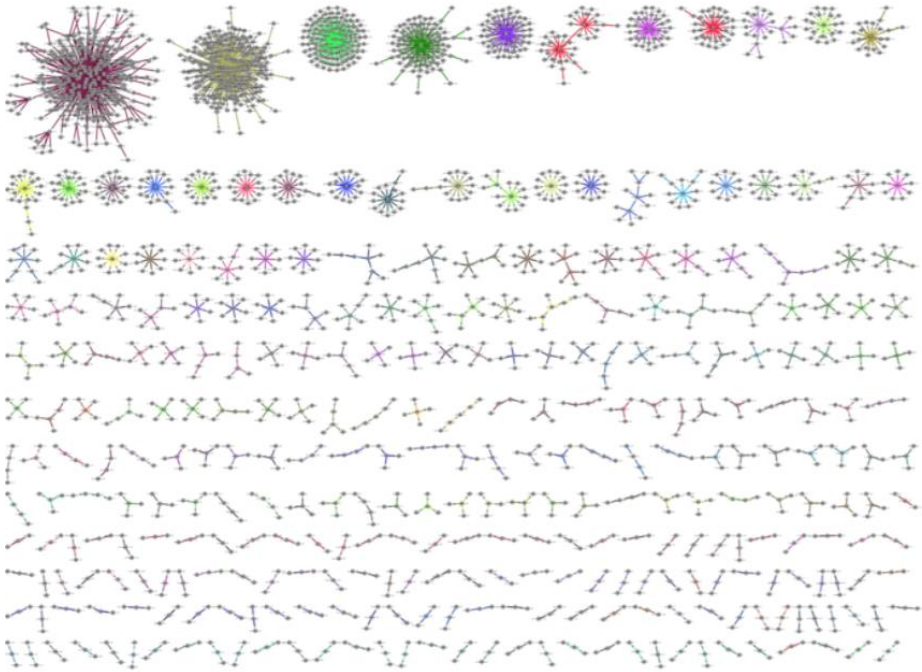
Graph is Abstract

Transportation network

Graph is Abstract (cont.)

Two-Step Flow in
Communication Research

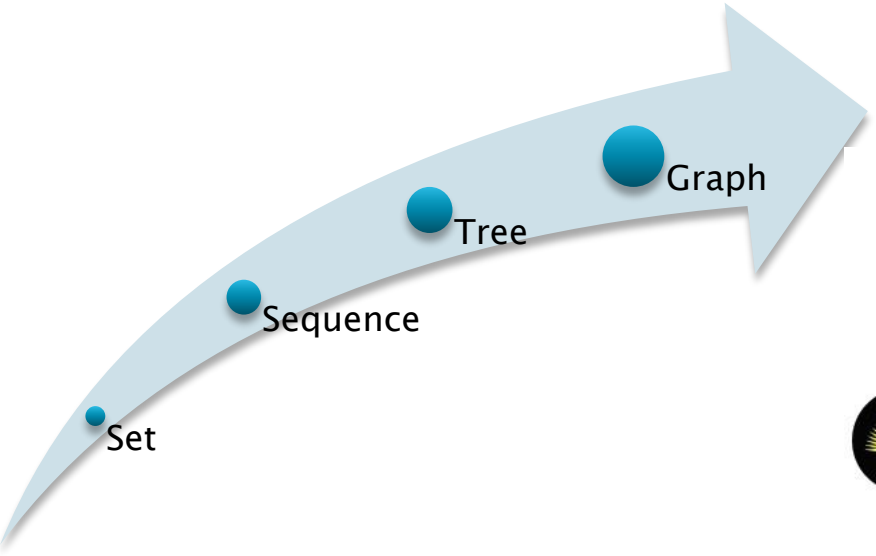




Word of Mouth
Message Spreading Examples

Graph is Intuitive

Graphs Can Model Complex Data Structures in Real World!

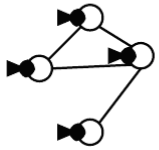


Evolution of Data Types

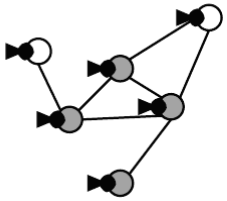
Taste Graph with Users and Products

Problem 1: Graph Structure Search

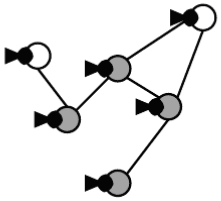
- **Substructure Search**
  - Retrieve graphs in DB which contain the query graph
- **Substructure Similarity Search**
  - Retrieve graphs in DB which have some components of the query graph
- **Superstructure Search**
  - Retrieve graphs in DB which are contained in the query graph
- **Pair-wise Structure Search**
  - Rank node-pairs in a graph based on link structure



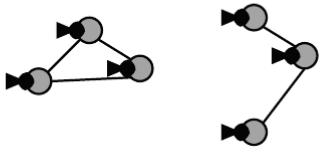
Query Graph



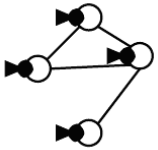
Substructure Search



Substructure Similarity Search



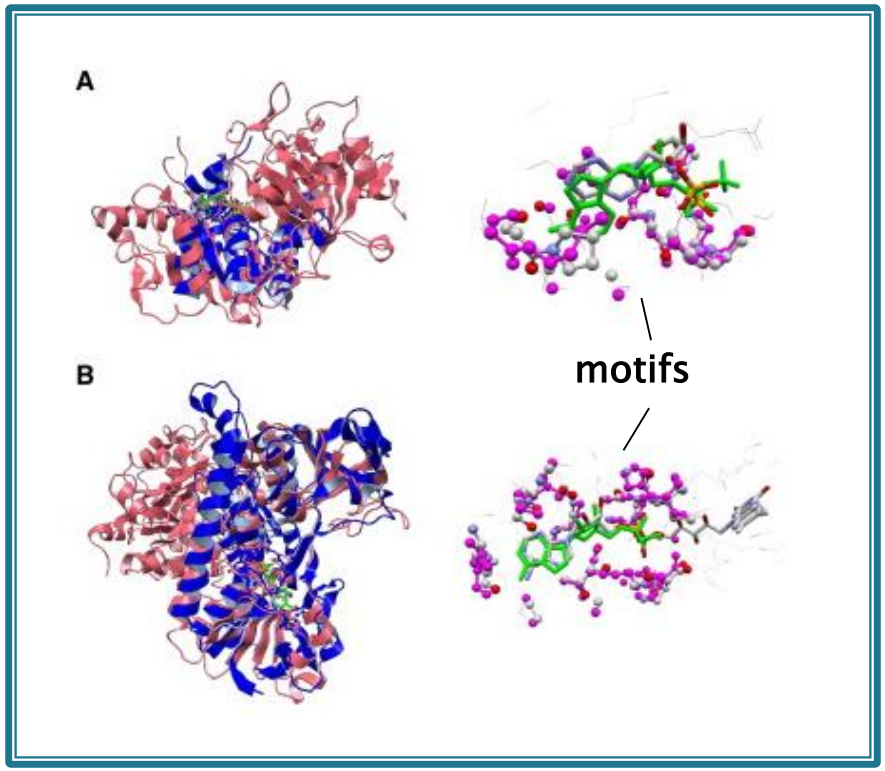
Superstructure Search



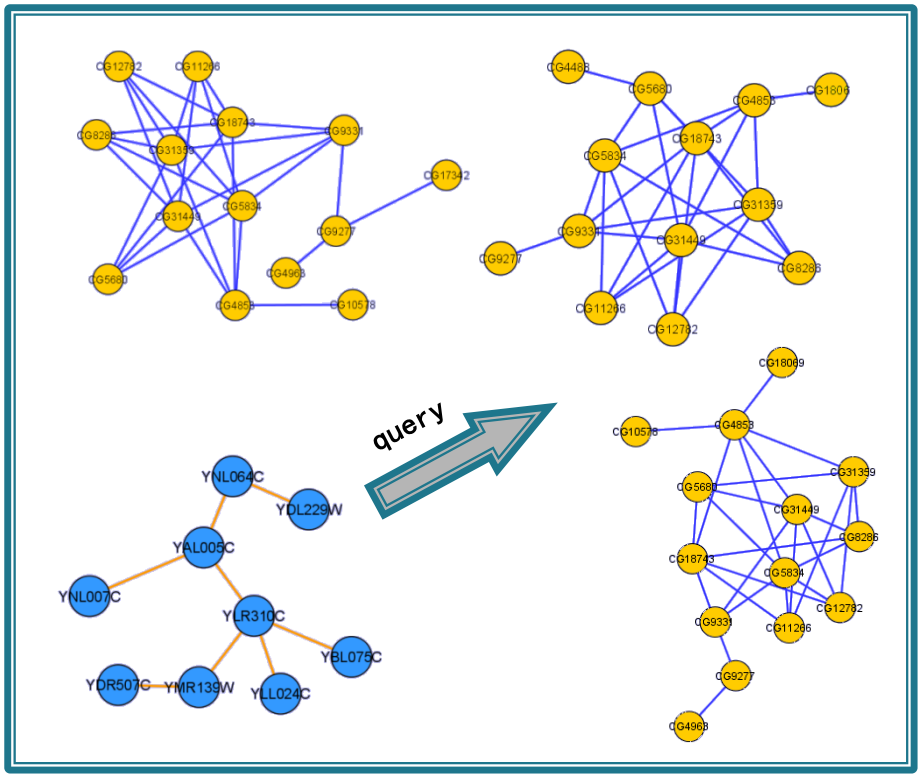
Pair-wise Structure Search

NP-Complete!

In Real World: Substructure Search (1,2)

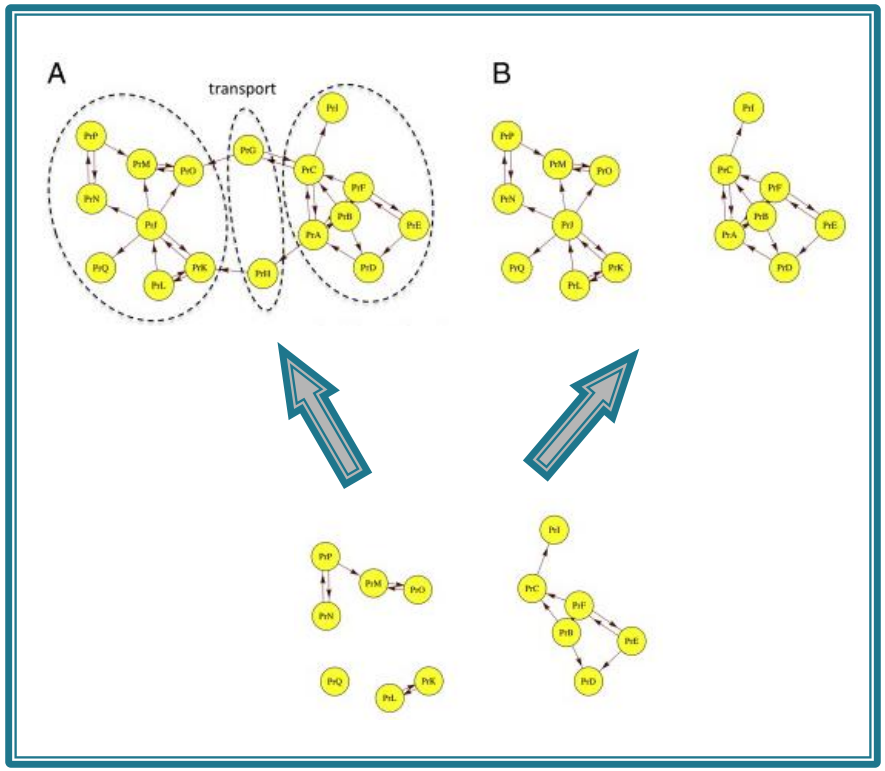


Find Motifs Shared by Different Proteins

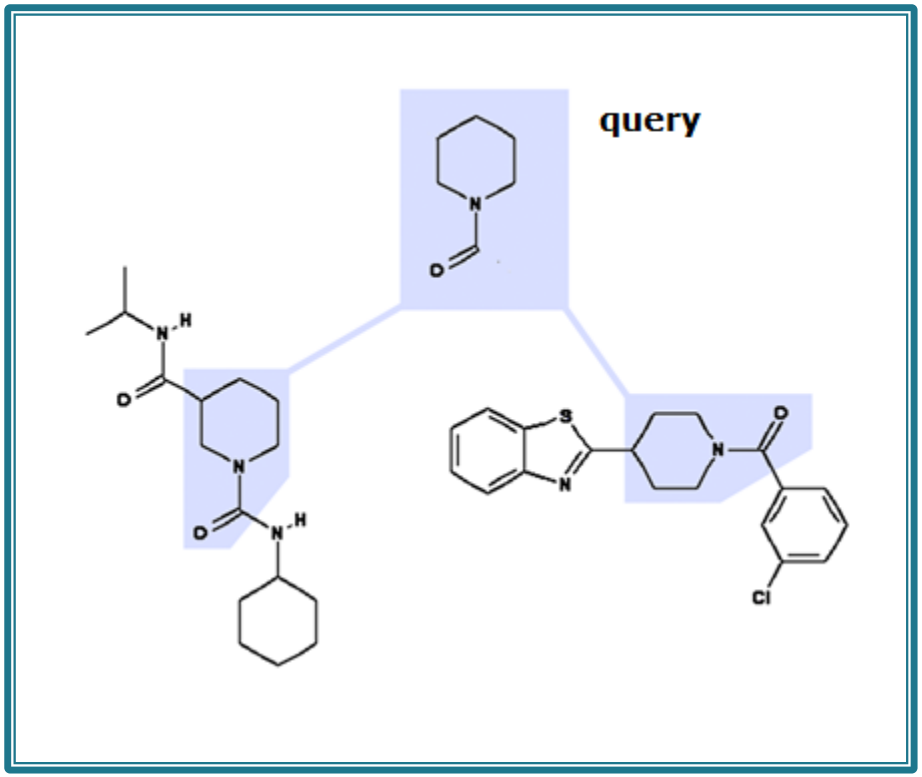


Query a Complex Species

In Real World: Substructure Search

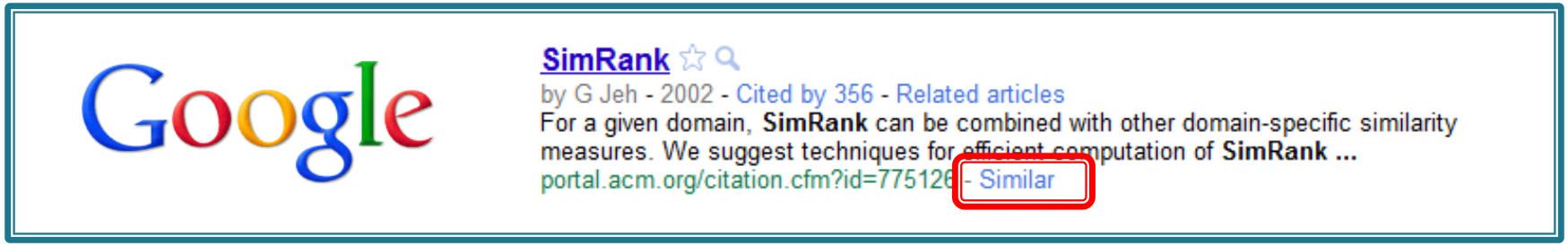


Drug Design



Chemical Compounds Identification

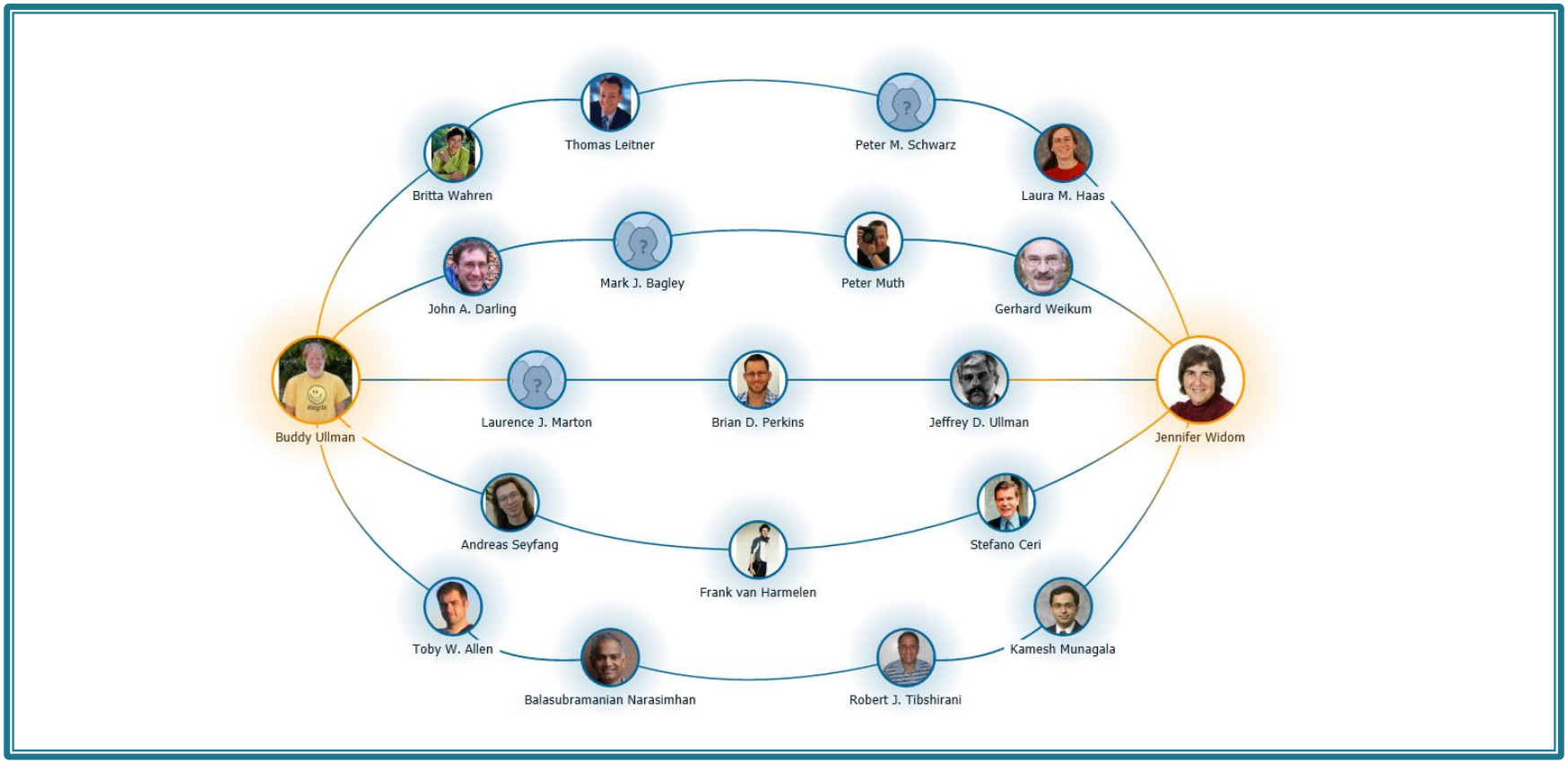
In Real World: Pair-wise Structure Search (1,2)



"Find-Similar-Document" Query in Search Engine

Collaborative Filtering in a Recommender System

In Real World: Pair-wise Structure Search (3)

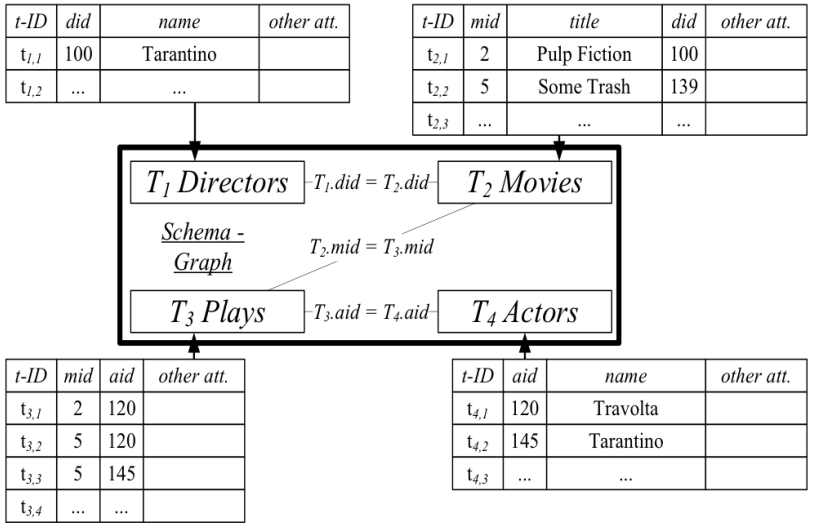


Detect Co-author Closeness of 2 Researchers via link aggregations

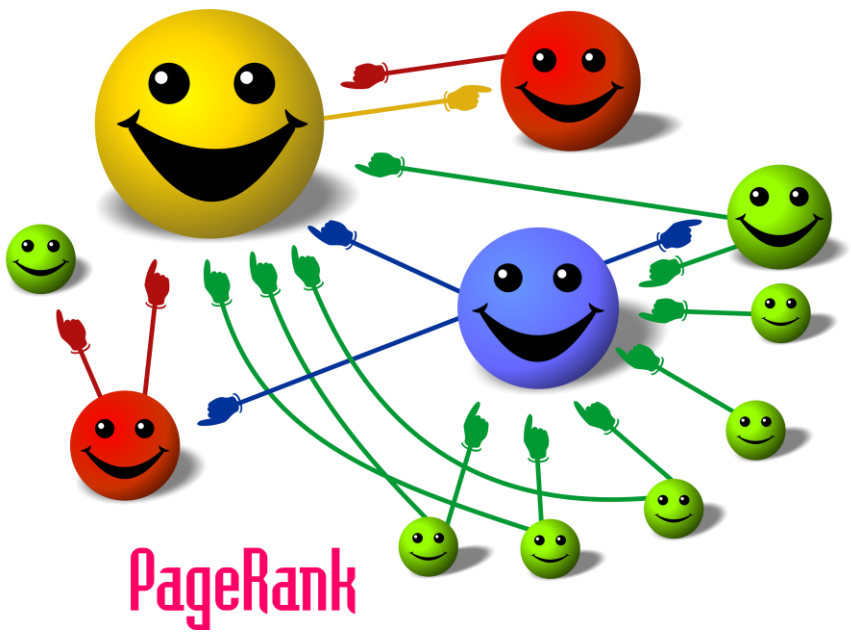
Problem 2: Keyword Search

- Question:

  Given documents and keywords, retrieve documents most related to the keywords. i.e., Google search style.
- Challenge: Model user intention, Computation.
- Solution:
  - Implement joins in relational DB.
  - Materialize all relationships by a big graph and compute the top-k minimum Steiner-Trees.



Problem 3: PageRank



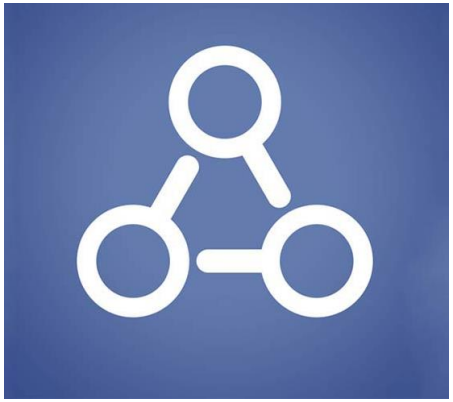
PageRank: Node Quality/Importance Ranking
It counts the number and quality of links to a page to determine a rough estimate of how important the website is.

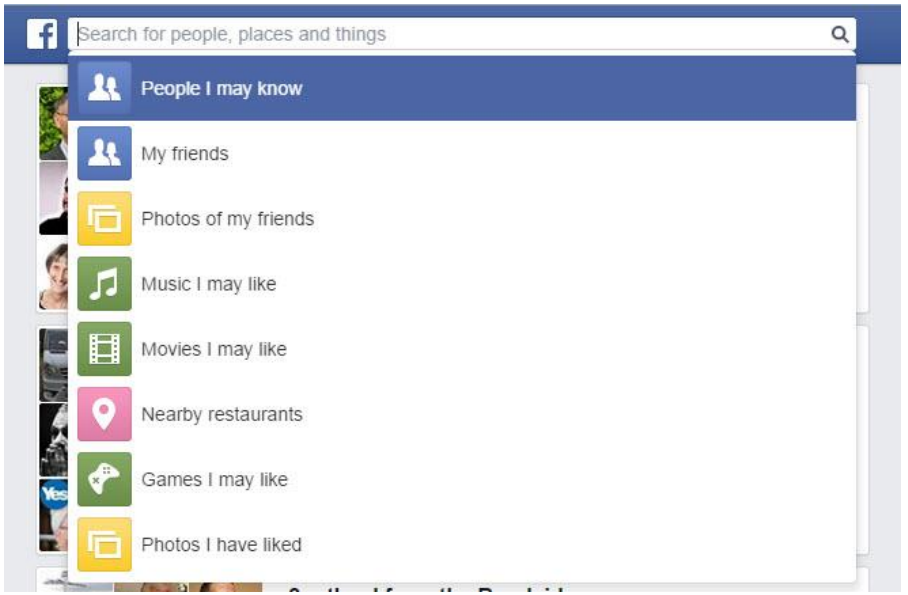
The underlying assumption is that more important websites are likely to receive more links from other websites.
Power Iteration:

- $PR(j) = \frac{1-d}{N} + d \cdot \sum_{i:i \to j} \frac{PR(i)}{D(i)}$

Problem 4: General Node Ranking

- Facebook's Graph Search
  - Semantic search engine to answer natural language queries.
  - Finds information from within a user's network of friends.
  - Build graph and correlations from users' network and contents.
  - Ranking and similarity are proceeded with inverted index and promoted with social proximity.

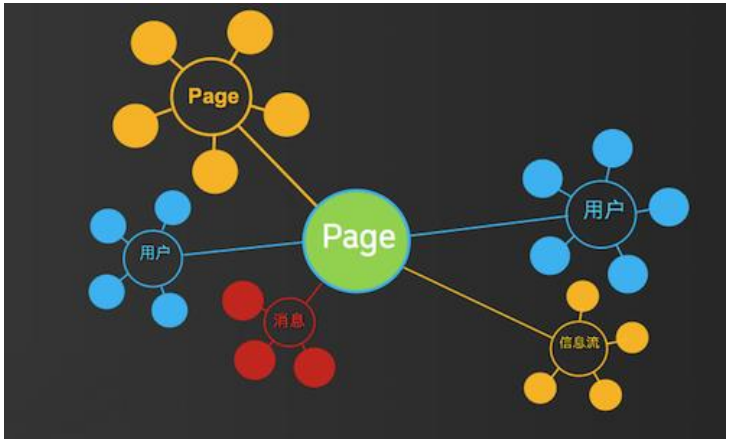




Problem 4: General Node Ranking

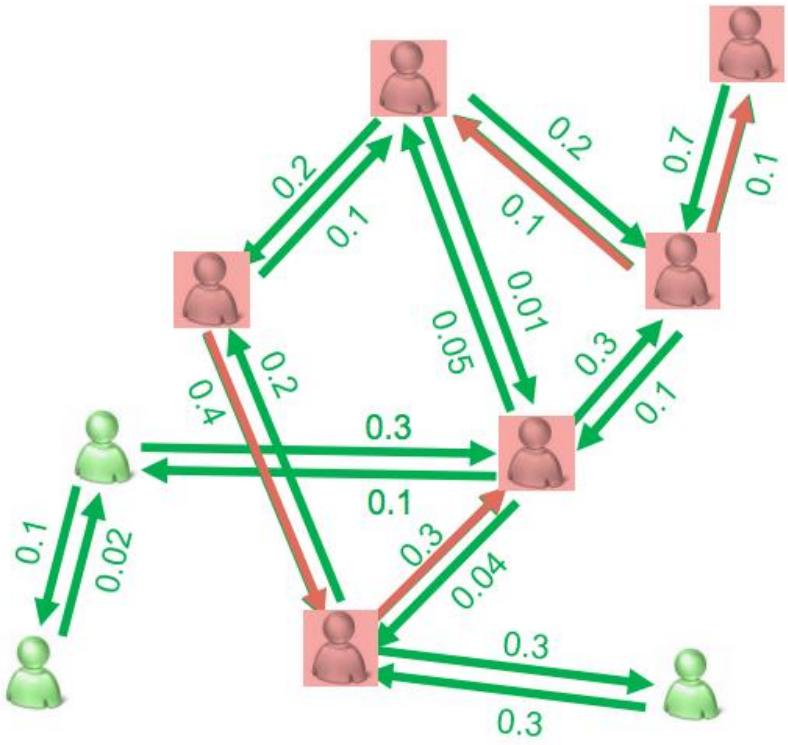
- Weibo's Page
  - Integration: connect location/post/picture with people.
  - Unified solution: retrieve interests, tastes.

- New Nodes:
  - Location, Images,Posts
  - Connections:
    - Transparency, temporary
    - Content connection



Problem 5: Influence Maximization

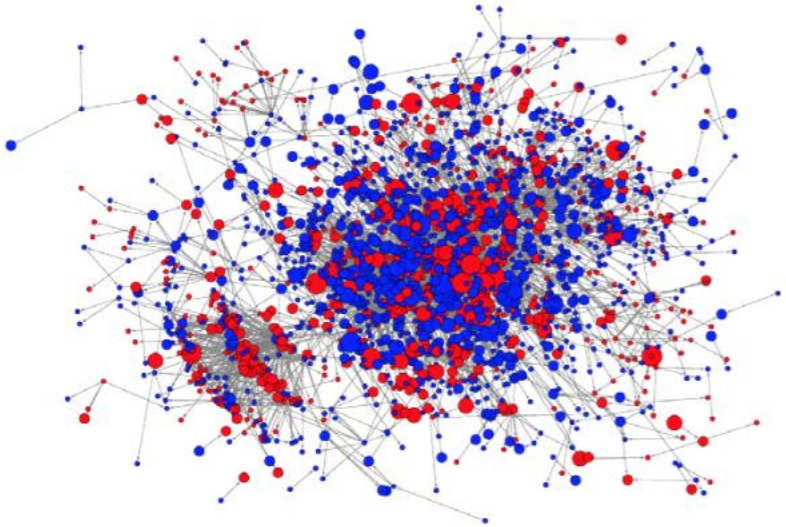
- Social influence graph
  - vertices are individuals ,links are social relationships
  - number p(u,v) on a directed link from u to v is the probability that v is activated by u after u is activated
- Independent cascade model
  - initially some seed nodes are activated
  - At each step, each newly activated node u activates its neighbor v with probability p(u,v)
- **Definition:**
  - **find k seeds that generate the largest influence spreads**



Influence Maximization (cont)

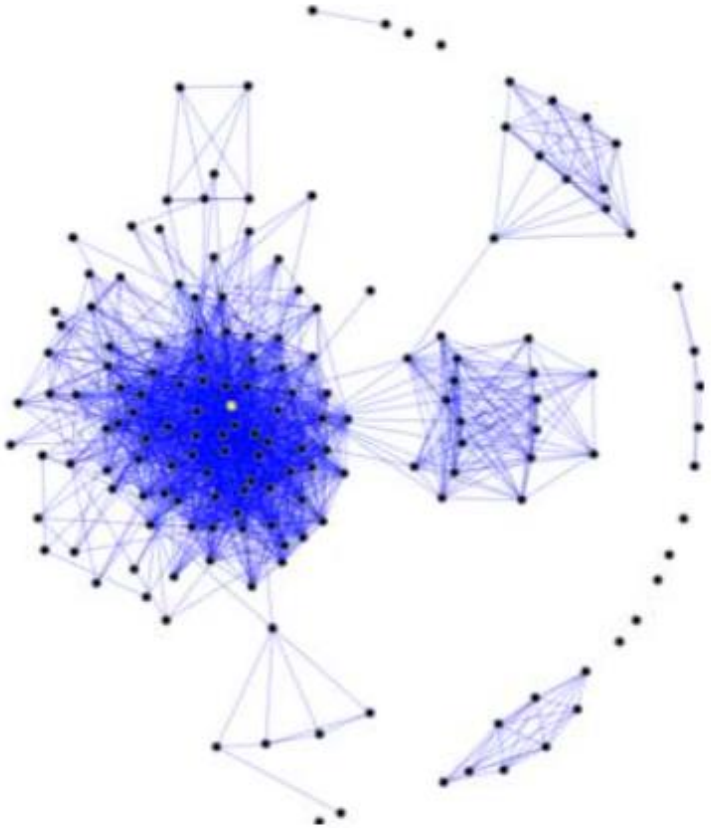
- Difficulties:
  - Finding most influential set is at least as hard as vertex cover. NP-Hard.

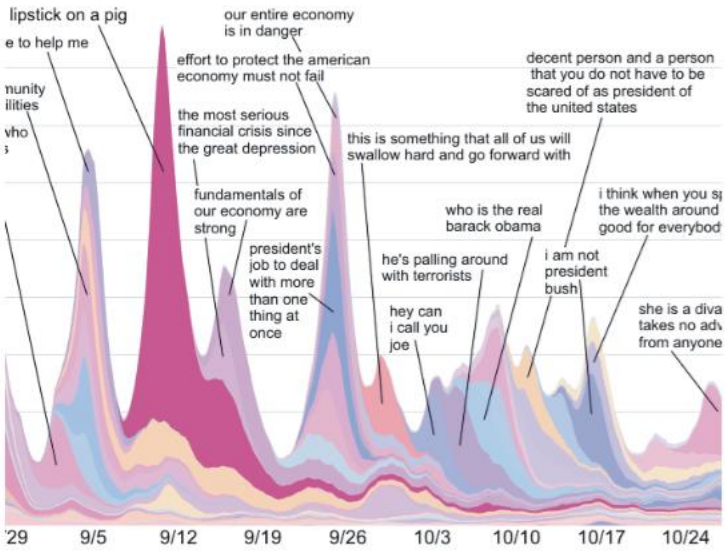
- Solutions:
  - Greedy approximation, hill climbing produces a solution ~63% of optimal value.
    - Monotone, submodular
  - Others subsequent studies improved the running time and extended scenarios.



Influence Maximization(cont.)

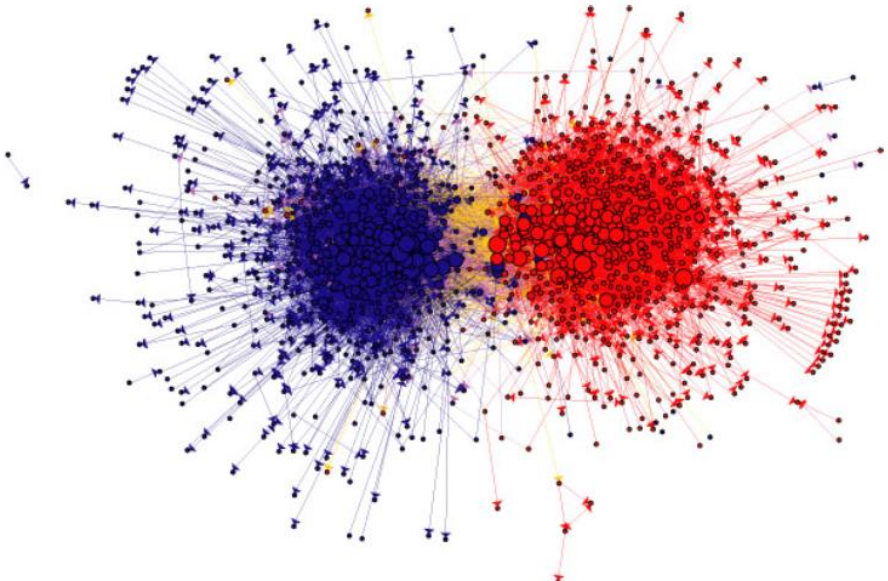
- Viral markets:
  - Influential initialization.
- Rumor Detection:
  - Misinformation control.





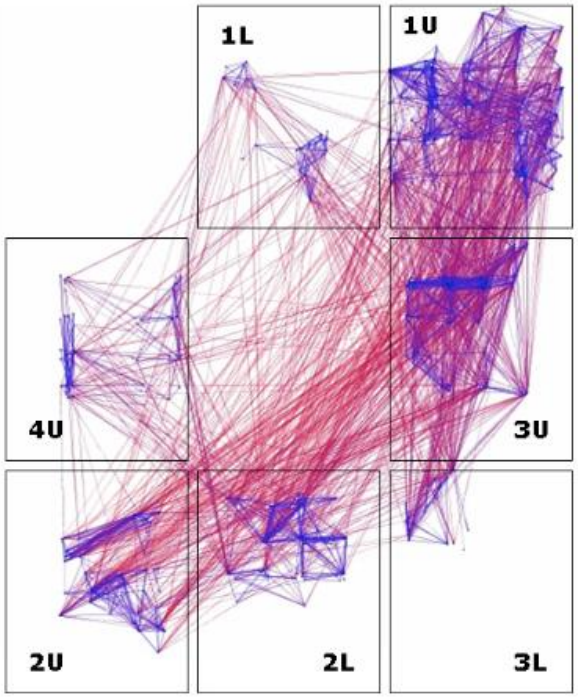
Problem 6: Community Detection

- Definition:
  - Groups of vertices with dense connections within groups and sparser connections between groups.
    - Within-group(intra-group)edges. – High density
    - Between-group(inter-group)edges. – Low density.



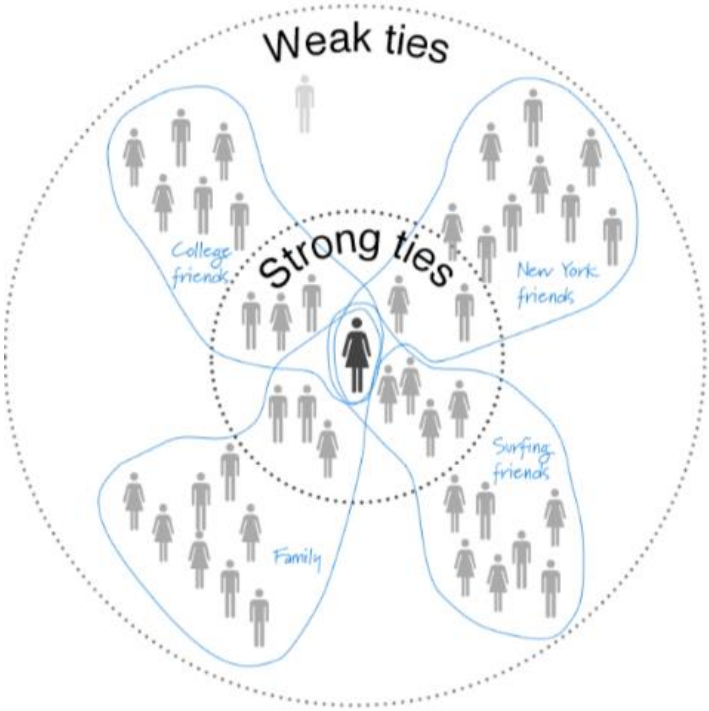
Community Detection(cont.)

- Criteria:
  - How can we evaluate whether the communities found are real communities? Group truth: user labeling.
  - Clustering: purity, accuracy.
  - Modularity, cut value.
- Difficulties(vary on different implementation):
  - Hierarchical Clustering: $O(n^2 \log n)$
  - Betweenness Clustering: $O(m^2 n)$
  - Newman, 2003: $O((m+n)n)$; $O(n^2)$ for sparse.



Community Detection(cont.)

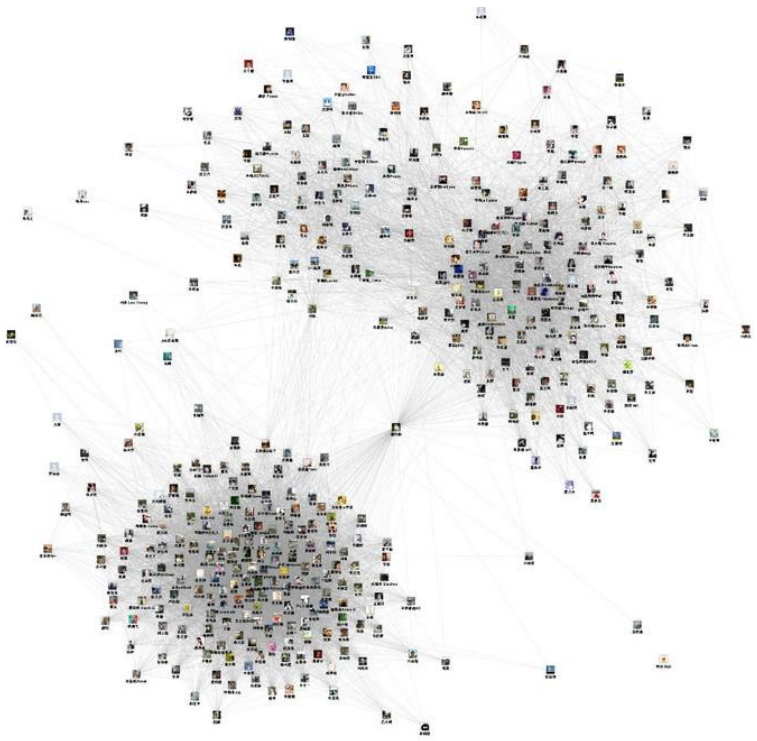
- The Real Life Social Network
  - Strong Ties: 4-6 (interact regularly)
  - Weak Ties: 130~150
    - Limitation of our brain.
    - Be consistent throughout history.
    - Farming village, Roman army
      Virtual online game groups.
  - Circles: multiple scenarios.



Community Detection(cont.)

- Visualize your Renren network
  - Renren Friend list fetch
  - Fruchterman-Reingold algorithm

  to layout the nodes $O(n^3)$
- Separate life circles.
  - Layout visualization
  - Density circles/groups.



https://donghaoren.org/projects/xnanalysis

Community Detection(cont.)

- Get your QQ friendship circles.
  - Download beta version, enable Quan feature.
  - http://quan.qq.com

- Categorize friendship

in temporal and group circles.

- Annotate friends' name with

Collective neighbors' census.

7. Distance Distribution

- Definition:
  - The length distribution of shortest distance d(u,v) between u to v in graph G.
  - Interesting global graph feature.
- Solution:
  - Naïve: All pair Shortest distance
    - Floyd-Warshall : $O(m^3)$
    - Thorup: O(mn), JACM, 1999
  - Sampling
    - O(mlogn), FOCS 1994, JCSS 1997

Distance Distribution

- ANF: diffusion-based approximation.
  - Approximate Neighbor Function
    - Incrementally maintain node's neighbors with distance lower than *t.*
    - Time Cost: O(m log n)
    - Memory: O(n log n)
- HyperLoglog counter: count approximately the number of distinct elements in a stream
  - Memory: O(n loglog n)
- HyperANF: the combination
  - Billions of nodes with small errors in a few hours using standard workstation. (Facebook)

Scalable

- Computation

Graphs are Big!

- 2.1 billion webpages in 2000
- 15 billion links in 2000

(already over trillion)

- 1.23 billon active users in 2013
- 117 billion friendships in 2013

- 645 million users in 2013
- 1.7 billion tweets/month in 2013

- 1.4 billion webpages in 2002
- 6.6 billion links in 2002

Challenges in Big Graph Processing

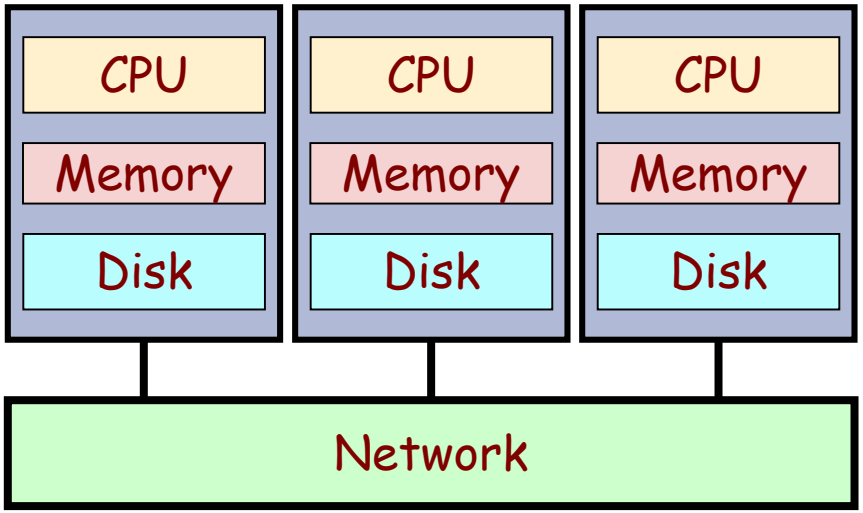
Graph Storage: Single Machine is not Enough

Scalability: Speedup with More Machines

Stability: Finish in Limited Time

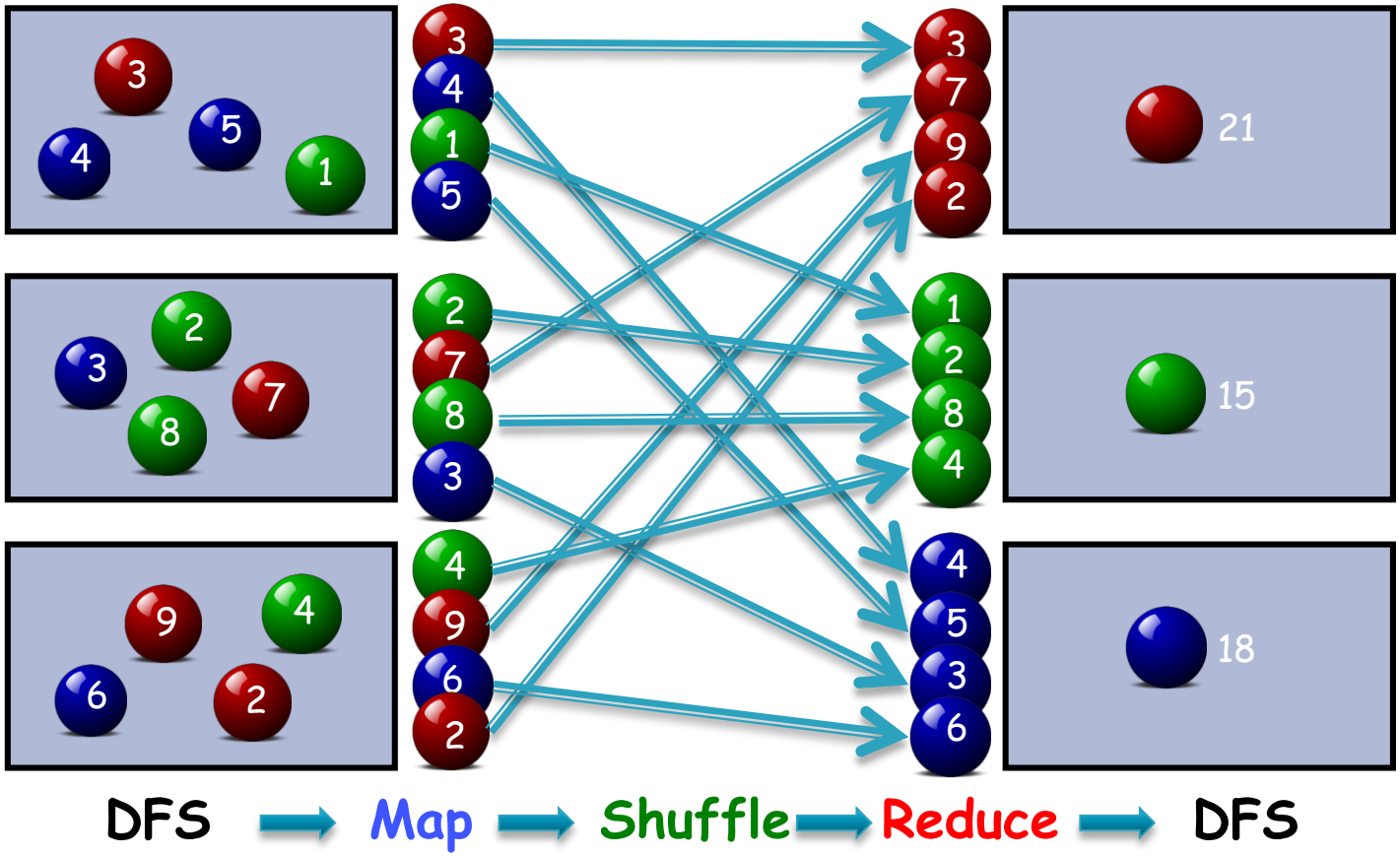
Robustness: No Crash due to Memory Limitation

What is MapReduce?



- Distributed shared-nothing environment
- Data are stored in a distributed file system (DFS) in forms of key-value pairs
- Data are transferred through network
- Process in rounds. Each round has three phases: Map, Shuffle, Reduce

MapReduce: An Example



- Calculate the sum for each color
- Key: colors  Value: numbers 1,2,3,…

Why MapReduce?

- **Scalability for big data**
  - Widely used in Google, Facebook, Yahoo, Amazon, etc.
  - Scalable to hundreds/thousands of machines
- **Stability**
  - Fundamental algorithms for scans, joins, and sorts
  - Usually small number of rounds
- **Simplicity**
  - Only two functions need to be implemented: Map and Reduce
  - Easy to be integrated with various information (e.g., structure, text, location, etc.)

Why MapReduce?

- **Robustness**
  - Data are passed to Map and Reduce in a streaming manner
  - No need to keep intermediate results in memory
- **Fault tolerance**
  - Automatic
- **Workload balancing**
  - Automatic
- **Elastic**
  - On-demand computation
  - Freely add/remove machines

Existing Positive Results

- Real-time computation class over PRAM against relational data (VLDB2013)
- MapReduce Class (MRC) (SODA2010)
  - Class with essential benefits compared to a central environment
- Minimal MapReduce Class (MMC) for processing spatial data (SIGMOD2013)
- None of these is suitable for processing big graph data in real-time.

A New Class for Graph: *SGC (SIGMOD2014)*

- **Disk:** O(m/t)/machine, O(m) total
  - Same as $\mathcal{MMC}$
- **Memory:** O(1)/machine, O(t) total
  - To ensure the robustness requirement
  - Only consider the memory used in map and reduce phases
  - Even stronger than $\mathcal{MMC}$
- **Communication/round:** Õ(m/t)/machine, O(m) total
  - Relax $\mathcal{MMC}$
- **CPU/round:** Õ(m/t)/machine, O(m) total
  - Relax $\mathcal{MMC}$
  - Only consider the CPU used in map and reduce phases
- **Number of Rounds:** O(log(n))
  - Relax $\mathcal{MMC}$
  - Given the above constraints, O(1) round is impossible for most graph problems.

Problems in *SGC*

- **Breadth First Search (BFS)**
- **Pagerank**
- **Graph Keyword Search (KWS)**
  - Each node in the graph contains text information
  - Input a list of keywords
  - Output the set of nodes that can reach all keywords within distance r_{max}
  - Can be processed similar to BFS by expanding from multiple keyword nodes simultaneously
- **Connected Component (CC) Computation**
  - Finish in O(log(n)) rounds
  - Share similar idea with O(log(n)) CRCW PRAM algorithms
- **Minimum Spanning Forest (MSF) Computation**
  - Finish in O(log(n)) rounds
  - Based on the Sollin's Algorithm.

Wrap UP

Conclusion and Remark

Summary:

- Ubiquitous (Big) Graph Data
- Problems and Application
- Unique Computational Challenges

Remark:

- Wide open
- Distributed/Cloud computing
- Approximate techniques
- Semantics

Q&A