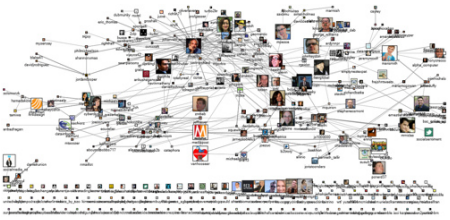
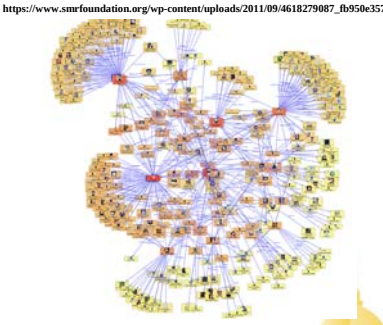
CSE-60742-01 Scalable Graph Algorithms

Peter M. Kogge
McCourtney Prof. of CSE
Univ. of Notre Dame
IBM Fellow (retired)



https://www.smrfoundation.org/wp-content/uploads/2011/09/4618279087_fb950e357d.jpg



http://mfs1.cdnsw.com/fs/Root/ck2ef-SocialNetworkAnalysis_Graph_Gradient.gif

Goals

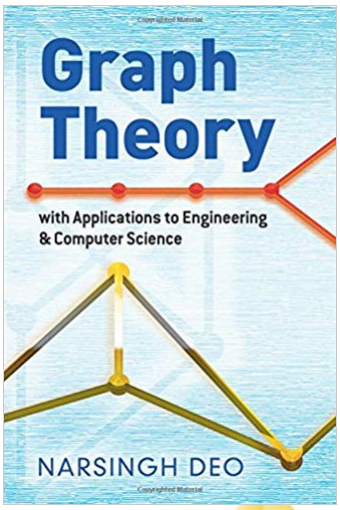
- Graphs have become a central part of:
  - social networks,
  - recommendation systems,
  - fraud detection,
  - national security.
- Course goal: explore graph processing
- Emphasis on "Scaling" of Computation
  - As graphs grow to large sizes
  - As parallelism in hardware increases
- Additional Goal: Hone your presentation & technical writing skills

Course Format

- Multiple "modules": each on different topic
- Each module:
  - Intro by professor
  - Presentations and summary papers by all students
    - Will be placed on web & integrated into technical reports
- Additionally each student will write and demonstrate a graph processing program
  - Ideally relevant to their particular research
- Grading:
  - Timely submission and presentation of all materials
  - Evaluation of presentations by class

Book

1. Intro
2. Paths & Circuits
3. Trees
4. Cut Sets
5. Planar & Dual Graphs
6. Vector Spaces
7. Matrix Representation
8. Coloring, Covering & Partitioning
9. Directed Graphs
10. Enumeration
11. Algorithms
12. Graphs in Switching & Coding
13. Electrical Networks
14. Operations Research
15. Survey



Expected Topics

- Key applications expressible as graphs
- Basic Graph Algorithms (sequential)
- Graph-oriented programming languages
- Parallel graph algorithms
- Student-selected graph projects

Presentation Evaluation Sheets

Your Name:				
Relevance of topic to Module 5: Absolutely Key 4: Better than average 3: Reasonable 2: Could have been better 1: Missing a lot				
Depth of Material: 5: Perfect 4: Missing small details 3: Reasonable 2: Could have been better 1: Missing a lot				
Presentation Organization: 5: Extraordinarily insightful 4: Better than average 3: Reasonable description 2: Could have been better 1: Missing a lot				
Key references: 1: Present 0: Missing				
Presentation Time: 3: Just right 2: Too short 1: Too long				
Overall presentation 5: Extraodrinary 4: Better than average 3: Good 2: Could have been better 1: Missing a lot				
Comments				

Student Papers

- I will post "blank" papers on Sharelatex
- Will "share" a separate copy with each student
- Ideally students will expand papers over semester
- Goal: at end, be able to quickly integrate contexts into a suite of overall reports
  - Graph Applications
  - Graph Benchmarks
  - Graph Programming Systems

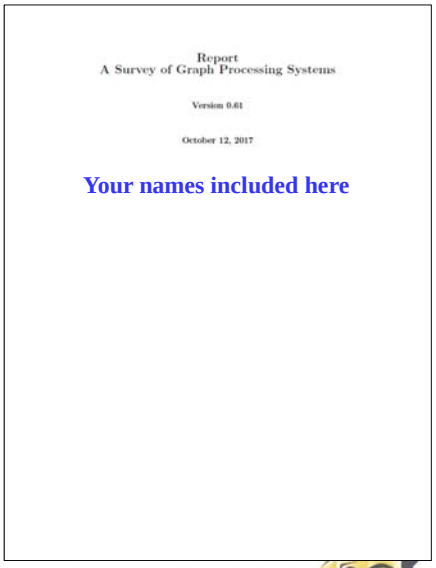
Group Report on Graph Programming Paradigms

Sections

- Introduction
- Graph Kernels
- Graph Languages
- Graph Libraries
- Graph Systems

Throughout Semester students will add sections, & be included as Contributors

Only 3 of 26 identified variants have initial text



Report
A Survey of Graph Processing Systems

Version 0.61

October 12, 2017

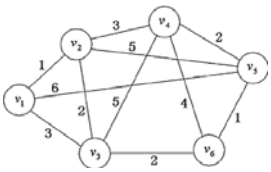
Your names included here

This Week Let's Talk About

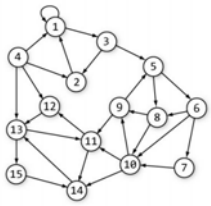
- Graph Basics
- Some well-known graph problems
- Existing graph benchmarks
- Spectrum of graph programming systems
- Each student's
  - Research Topics
  - Interests
  - Programming experience

Graph

- **Graph** G = (V,E) – pair of sets
  - V = set of ***vertices***
  - E = set of *edges*
- **Edge** = pair of vertices (u,v)
  - Undirected: no "direction" to edge
    - Edge (u,v) same as (v,u)
  - Directed: u is source, v is destination
    - Edge (u,v) is not same as (v,u)
- **Hypergraph**: edge can connect any number of vertices



https://www.researchgate.net/profile/Chenggpei_Tang/publication/250922991/figure/fig2/AS:294240128778241@1447163721109/An-undirected-graph-G-with-7-vertex-and-12-edges.png



http://www.greatandlittle.com/studios/public/blowup-images/Dart/.directed_graph_m.jpg

https://i.stack.imgur.com/pWMWS.png

Basic Terms

- Edge e is **incident** on vertex v if e = (u,v)
- **Loops**: source & destination is same
- **Planar**: can be drawn so no edges cross
- **Face**: region fenced by a set of edges
- **Isolated vertex**: no edge sources or sinks
- **Labeling**: "Value" assigned to vertex or edge
- **Subgraph**: subset of vertices and edges where all subset edges connect only vertices from vertex subset

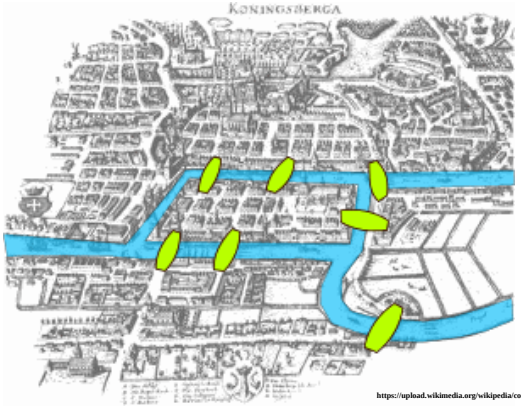
Degrees

- Directed graphs:
  - **In-Degree** of v = # of edges incident on v
  - **Out-Degree** of u = # of edges leaving u
- Undirected graphs:
  - Degree of x d(x) = # of edges sourced or sinked at x
  - Book counts self-loop as 2
  - $\Sigma_{x \text{ in } V} d(x) = 2|E|$
- **k-Regular Graph**: all vertices have degree k

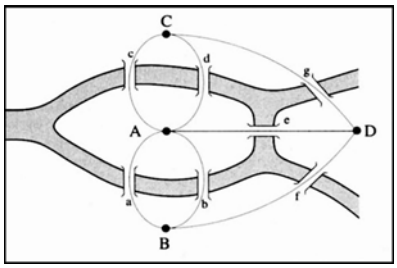
Following the Edges

- **Path** of length k from u to v: sequences of k edges (u_i,v_i) where $u = u_1$, $v_i=u_{i+1}$, $v_k = v$
  - **Simple path**: no vertices repeated
- **Connected Graph:** path between every 2 vertices
  - **Strongly connected:** paths follow edges
- **Cycle** of length k: path from u back to u
  - **Simple cycle**: no vertex touched by >2 edges
- **Tree:** Connected graph with no simple cycles
  - **Leaves**: vertices with no outgoing edges
- **Circuit**: cycle that includes all vertices

7 Bridges of Konigsberg



https://upload.wikimedia.org/wikipedia/commons/5/5d/Konigsberg_bridges.png

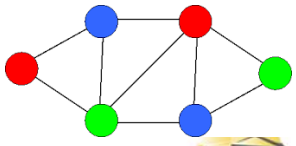


https://i.stack.imgur.com/Ufw1m.jpg

- Problem: circuit thru city to cross all 7 bridges exactly once each (4 vertices, 7 edges)
- Euler: Circuit needs **even** degree on each vertex

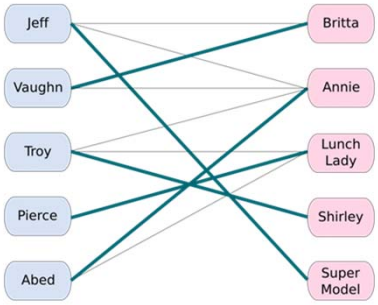
Graph Coloring

- **Coloring**: label from a small set of values
- **Vertex coloring**: no two vertices connected by an edge have same color
- **Edge coloring**: no two edges that share a vertex have same color
- **Face coloring**: no two faces that share an edge have same color
- **4-color conjecture:** solvable for planar graphs
- **3-color problem**: NP-complete



Bipartite Matching (aka. Marriage Problem)

- Given 2 sets of vertices L & R
- and set E of edges between them
- Is there a subset of edges where every vertex has at most 1 edge?

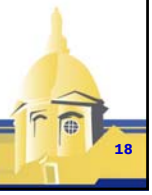


Classes of Application Computations

- **Batch**: function applied to entire graph of major subgraph as it exists at some time
- **Streaming**:
  - Incoming sequence of small-scale updates
    - New vertices or edges
    - Modification of a property of specific vertex or edge
    - Deletions
  - Sequence of localized queries

General Classes of Graph Computation

- Characteristics of individual vertices
  - E.g. "properties" such as degree
- Characteristics of graph as a whole
  - E.g. diameter, max distance, covering
- Characteristics of pairs of vertices
  - E.g. Shortest paths
- Characteristics of subgraphs
  - E.g. Connected components, spanning tree
  - Similarities of subgraphs, …



Current Benchmark Suites

Kernel	Kernel Class: Connectedness	Path Analysis	Centrality	Clustering	Subgraph Isomorphism	Other	Benchmarking Efforts: Standalone	Firehose	Graph500	GraphBLAS	Graph Challenge	Graph Algorithm Platform	HPC Graph Analysis	Kepner & Gilbert	Stinger	VAST	Outputs: Graph Modification	Compute Vertex Property	Output Global Value	Output O(1) Events	Output O(|V|) List	Output O(|V|^k) List (k>1)
Anomaly - Fixed Key						X		S												X		
Anomaly - Unbounded Key						X		S												X		
Anomaly - Two-level Key						X		S												X		
BC: Betweeness Centrality			X							B		B		B	S			X				
BFS: Breadth First Search	X								B	B		B	B	B	B			X			X	
Search for "Largest"						X						B									X	
CCW: Weakly Connected Components	X												B	B	S			X			X	
CCS: Strongly Connected Components	X												B	B							X	
CCO: Clustering Coefficients				X										B	S			X				
CD: Community Detection			X	X											S			X			X	
GC: Graph Contraction				X									B	B							X	
GP: Graph Partitioning				X							B/S			B							X	
GTC: Global Triangle Counting					X							B							X			
Insert/Delete						X									S		X					
Jaccard				X			B/S															X
MIS: Maximally Independent Set										B				B								
PR: PageRank			X										B					X				
SSSP: Single Source Shortest Path		X							B				B/S	B				X			X	
APSP: All pairs Shortest Path		X												B								X
SI: General Subgraph Isomorphism						X					B/S											
TL: Triangle Listing						X					B/S											X
Geo & Temporal Correlation						X										B/S				X		

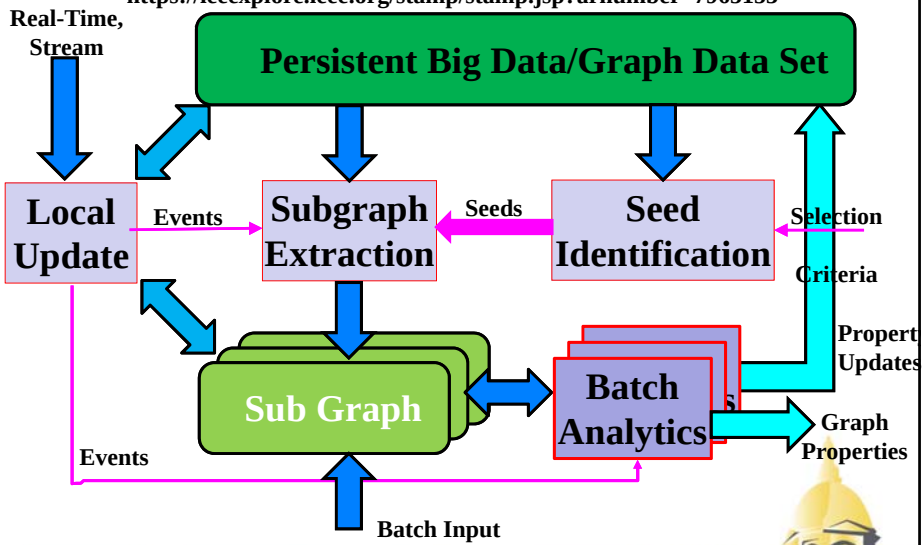
Kernel Class: what class of computing kernel performs

Benchmarking Efforts
- S => Streaming
- B => Batch
- B/S => Both

Outputs: what is size or structure of result of kernel execution?

Canonical Graph Processing

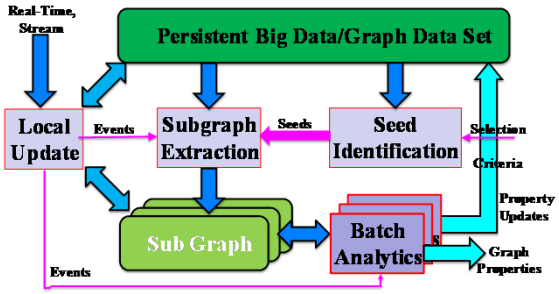
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7965153



Streaming Characteristics

- Two kinds
  - Streams of queries
  - Updates to persistent data
- Both typically localized to start with
- Streaming updates often multi-step
  - Perform update (use atomics)
  - Perform some local computations
  - Compare to threshold
  - If threshold passed, extract some larger subset
  - And perform a bigger analytic

Observations



- Data sets live in persistent memory
- Streaming updates trigger threshold tests
- Streaming queries result in local graph traversals
- Batch analytics used primarily for analysis & new property computation

For Next Class

- Think about problems relevant to you that have relevance to graphs
  - Look thru book
- Be ready to discuss for 2-5 min in class
  - No detail or prepared presentation needed
- OK to have >1 interesting problem
- Goal: find a problem that you want to pursue in more detail over semester

First Student Presentations

15-20 min in class on selected problem

- What is driving application
- How does this problem convert into a graph
- What are properties of real-world graphs
  - Nature, # of vertices, edges
  - How might the size of such graphs grow in future
- Are there available on-line sample data sets
- What is metric by which solutions would be compared?
- From 20,000 ft, what are the key algorithms?

Presentations will be posted right after class

Written paper to be prepared a week later