Challenges in Streaming Graph Analysis

Jonathan W. Berry
Cynthia A. Phillips
Steven Plimpton
C. Seshadhri
Sandia National Laboratories
Matthew Oster (Rutgers)

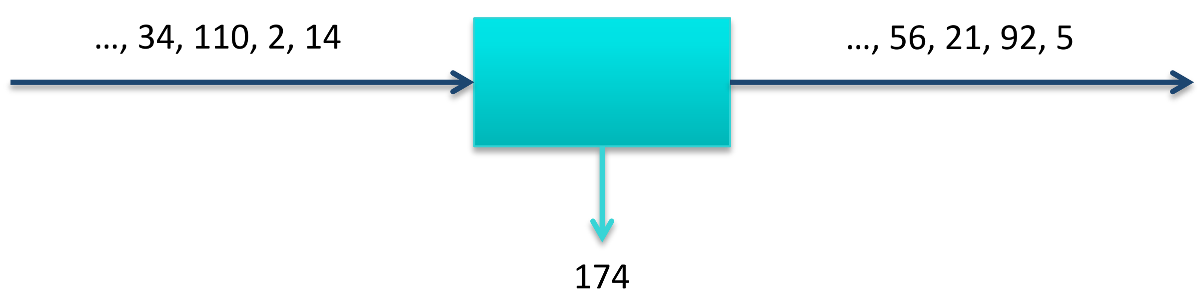
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Classic Streaming

- Information arrives piece by piece as generated in a (huge) stream
- Answer a question as the data set streams by
  - Use much less local space than the stream size

- Example: Watch a permutation of 1,…,n (n known) with one number missing. You have space for one number. Determine the missing number.

- Answer: store the sum of the numbers you have seen.



Streaming Relevance

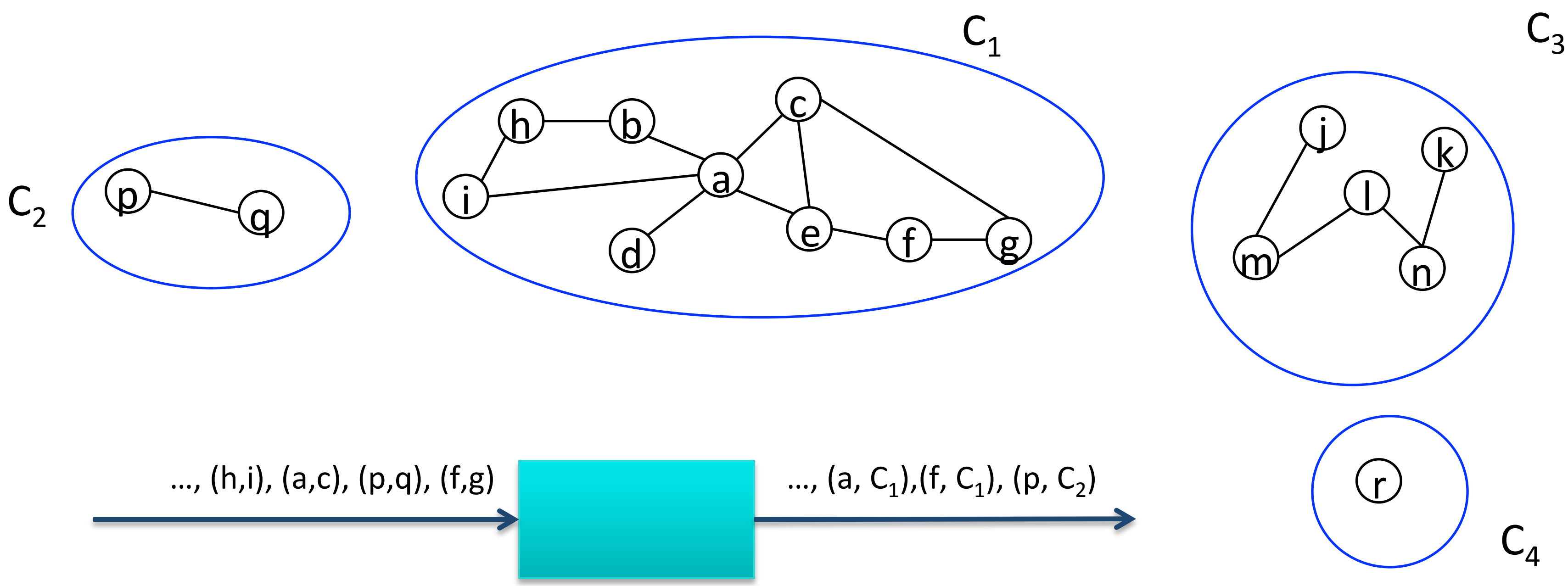
- Computer communication networks link entities
- Represent relationships with a graph

Cyber traffic/activity is a stream through time

- Stream are huge
  - Humans cannot keep up
  - Gap will only increase
- It's time to develop fundamental streaming graph algorithms to partially automate the analyst's tasks

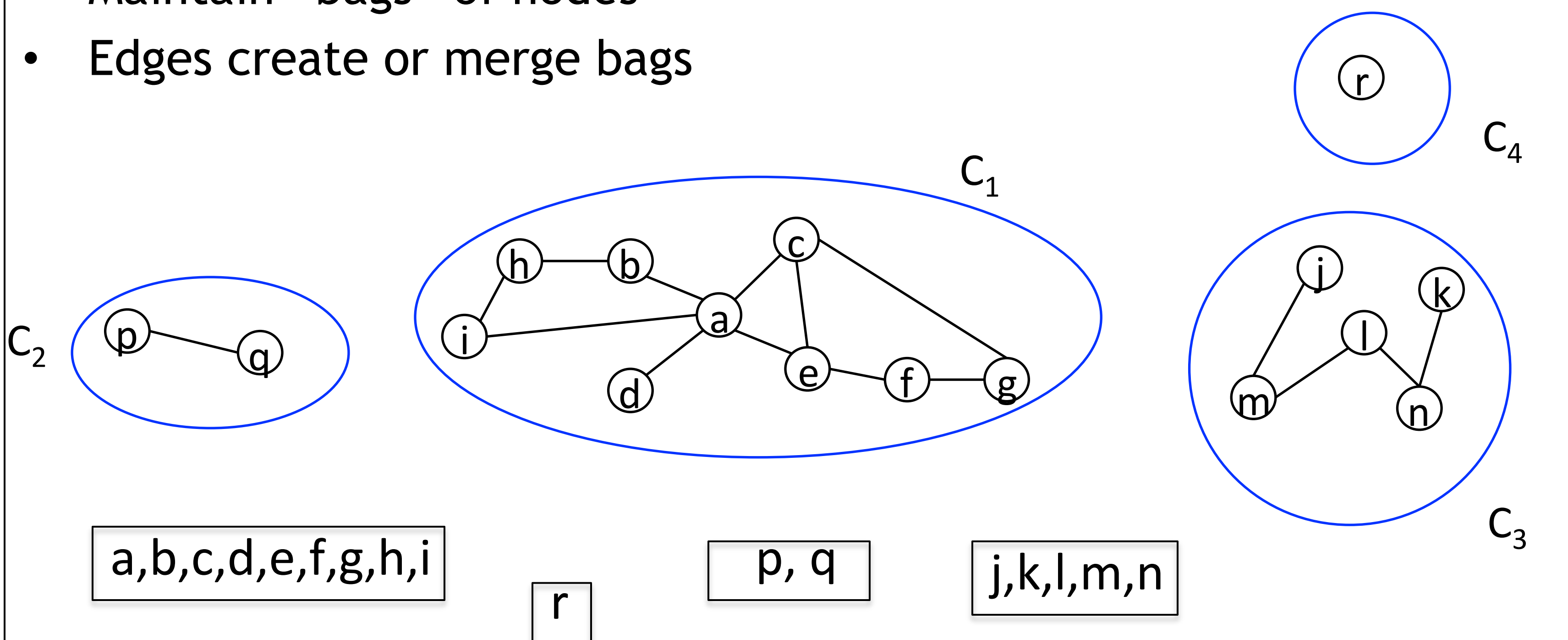
Connected Components

- Input: stream of edges (learn nodes from edge)
- Output: (node, label) pairs
- Two nodes have the same label if there is a path between them
- Can't output 2 pairs with different labels until seen all of (finite) graph



Semi-streaming Connected Components

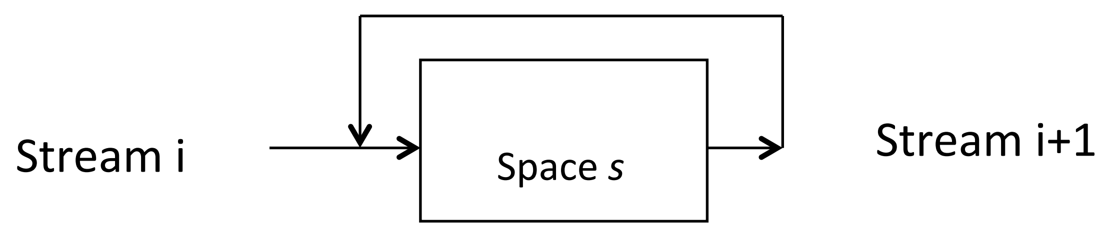
- Graph with |V|=n. Allowed O(n polylog(n)) space
- Can store all the nodes
- Maintain "bags" of nodes
- Edges create or merge bags



W-Stream Model

- Read a stream, write a stream for another pass
  - Finite Stream
  - The rewrite stream is "in the air."
  - Trade off space vs # passes
- Demestrescu, Finocchi, Ribichini: s space, $O\left(\frac{n \lg n}{s}\right)$ passes

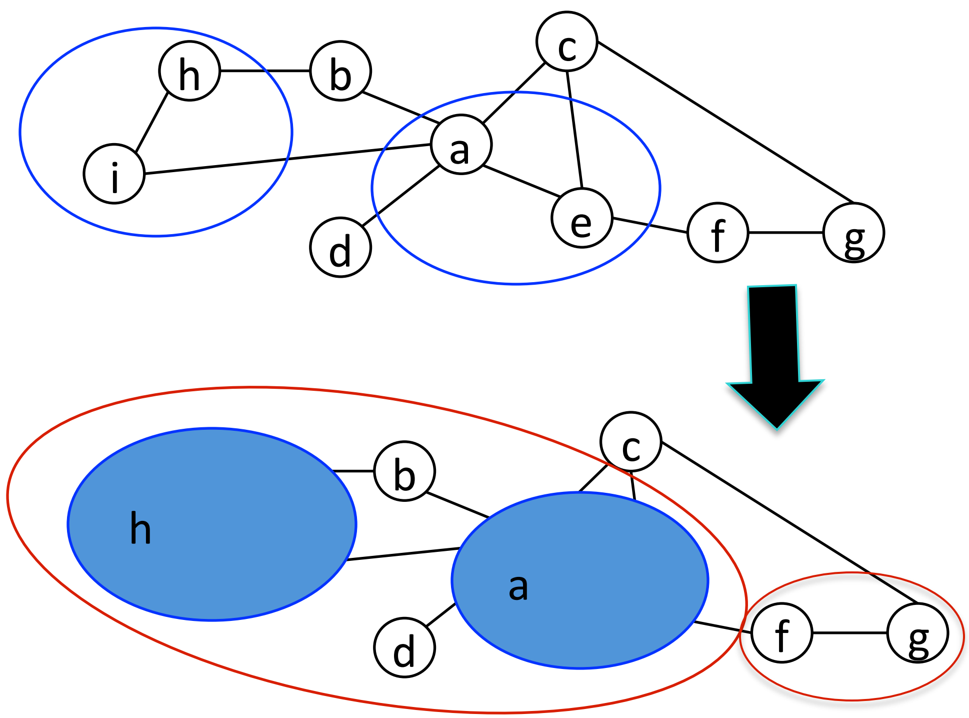
n = # nodes



C. Demetrescu, I. Finocchi, and A. Ribichini. Trading off space for passes In graph streaming problems, ACM Transactions on Algorithms, Vol. 6, No 1, Dec 2009.

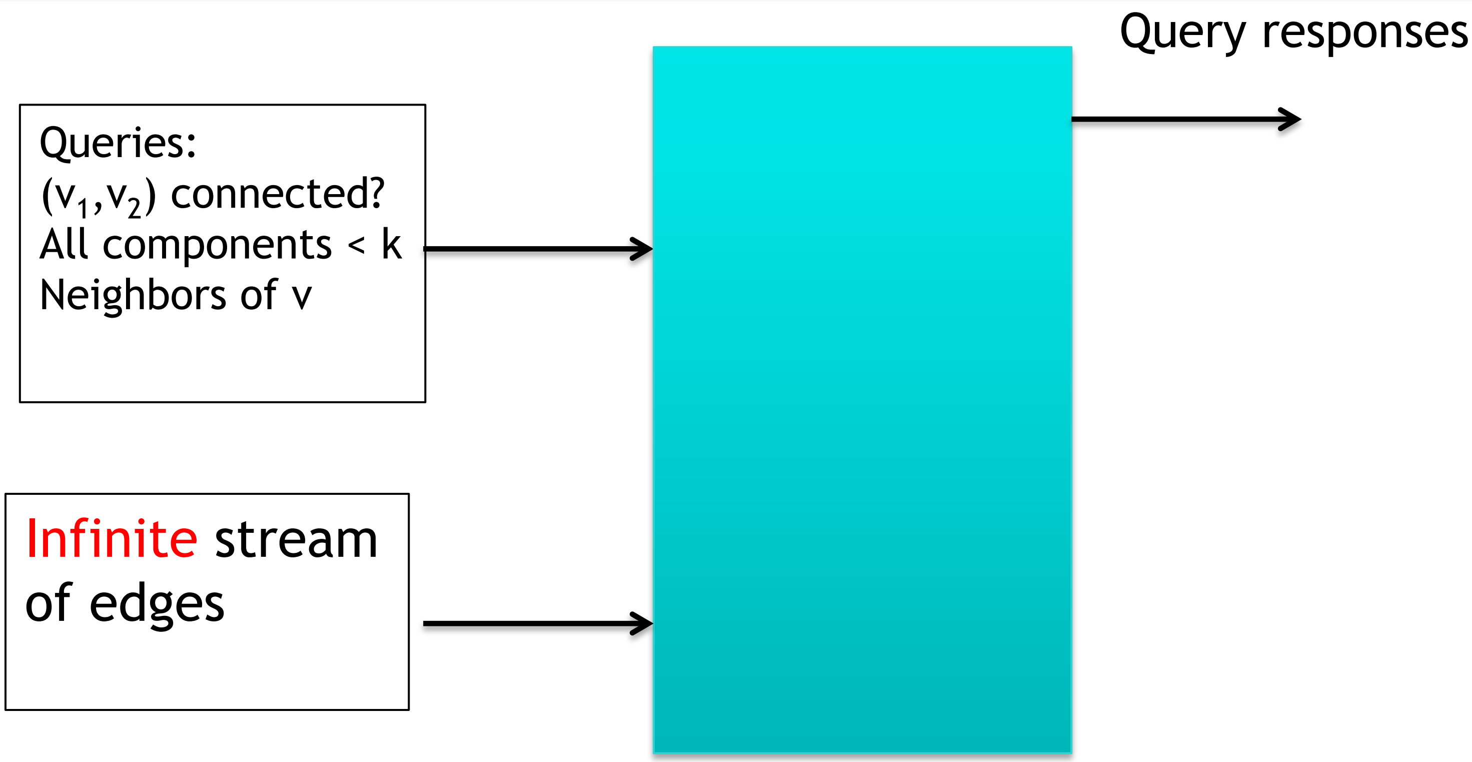
W-Stream Connected Components

- Idea: each phase/pass, contract connected components:



Each stream 2 parts: A: contracted graph edges, B: substructure of contracted components, (node, label)

What Analysts Want/Need



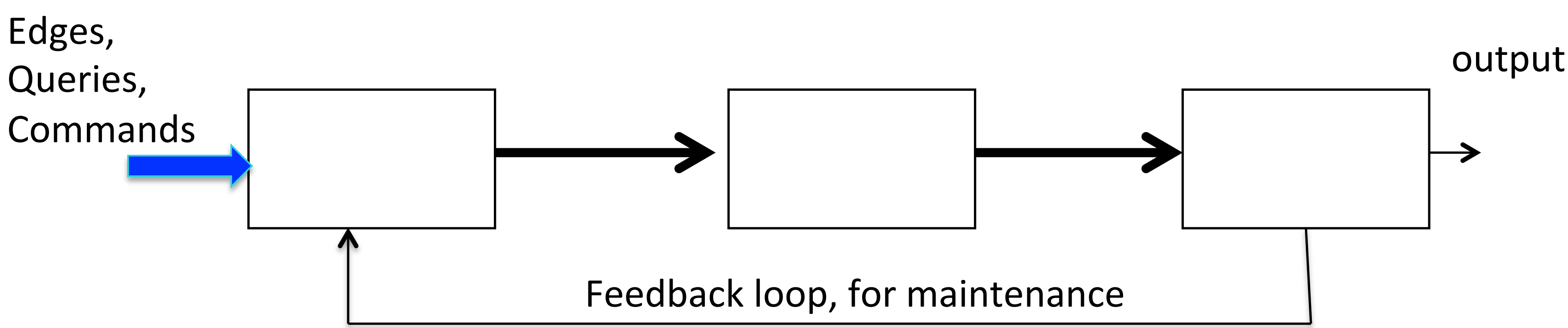
- State of art: D. Ediger, J. Riedy, D. Bader, H. Meyerhenke, MTAAP 2011
  – Dynamic connectivity for scale-free graphs, shared memory, process input edges in batches, achieves 240,000 updates/second if most are insertions

Dealing with an Infinite Stream

- Efficiently use aggregate local memory across processors
  - O(1) space per edge
  - Edges not dropped unless the system is storing Ω(ps) edges ("full")
- Aging
  - Command (in stream) to remove all edges older than t
    - Reduce space when the system is filling up
    - Newer edges likely more interesting
  - Must recompute components, so must store all edges
  - No queries till recomputed, output token when OK to resume
  - No edges dropped during recomputation
- Queries
  - Answer relative to graph at time of query
  - Constant-sized answer immediate, non-constant as able

New (Challenging) Model: X-Stream

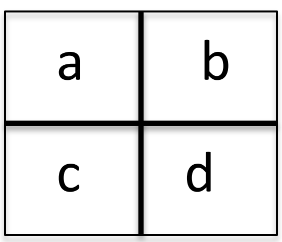
- Parallel ring architecture with systolic synchronous communication
  - Input is 1/k of bandwidth
- For handling infinite streams (though graph finite)
  - W-stream (finite) can fill and spill, reduced latency
- Stream entry point can (must) move around the loop
- Queries stream through in one pass, possible latency
- Theoretical measures: k, worst-case/avg update/message, post-aging stabilization speed, constant in memory usage guarantee
- Practical measures: streaming rate, query response latency

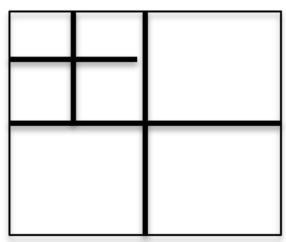


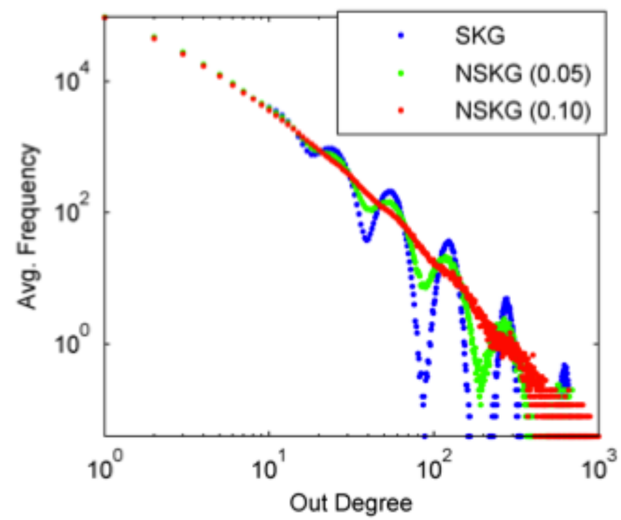
Sandia National Laboratories

Benchmark Generation

- Usual benchmark generator: R-MAT (Chakrabarti, Zhan, Faloutos 2004)
  - R-MAT was an original GRAPH 500 benchmark
  - Special case of Stochastic Kronecker Graphs [Leskovec et al 2005,7,10]
  - Parallel Edge generation, Suitable for streaming
  - Advertised heavy-tailed degree sequence for appropriate a,b,c
- But R-MAT has issues with degree distribution,# isolated vertices, k-core [Seshadhri, Pinar, Kolda ’11]







(b) WEBNotreDame

Benchmark Generation

- They proposed new social graph generation model BTER [Seshadhri, Kolda, Pinar 2012]
  - Provable community structural requirements assuming only heavy tailed degree distribution and high clustering coefficient.
  - Still issues with low-degree nodes (Poisson), joint-degree distribution, etc.

This search may never end as researchers discover properties to reproduce.

How well can we

- Generate “representative” graphs in place in parallel
- Generate “representative” data for streaming graph algorithms

Streaming and Exascale

- Communication squeeze between hierarchies
  - Energy costs, bandwidth
- Data is finite, resident, but huge
  - E.g. data from scientific simulation
- Compute globally with local summaries
  - Statistical approximations
  - Sublinear-time summaries
    - E.g. Sesh Commandur: finding a triangle
    - Tend to be sampling based (parallel)
- Stream data to processors
  - Does this need a new model for algorithm development and analysis?