Parallel and Real-time Analysis of Dense Structures from Graphs

Srikanta Tirthapura

Iowa State University

snt@iastate.edu

(based on joint work with Apurba Das, A. Pavan, Arko Mukherjee, Mike Svendsen, Kanat Tangwongsan, Kun-Lung Wu)

Big and Fast Graphs

- Social Media
  - Twitter: 600 Tweets/sec ≈ 200 billion Tweets/year
- Machine Generated Graphs as RDF triples
- Transportation
  - Wavetronix Sensors, 50KB/sec, 4-5GB/day
  - INRIX Sensors, 250KB/sec, 22GB/day
  - Image/Video data from Cameras

A (Simplistic) Framework for Analyzing Large Data

	Sequential	Parallel and Distributed
Batch Processing (store and process later)	External Memory Algorithms and Systems	Batch Parallel Algorithms Hadoop/Spark MapReduce MPI
Real-time Stream Processing (process data now)	Sequential Streaming Algorithms Some CEP/Streaming Systems	Parallel & Distributed Streaming Algorithms IBM Infosphere Streams Apache Spark Streaming Apache Flink

A (Simplistic) Framework for Algorithms and Software for Analyzing Large Data

	Sequential	**Parallel and Distributed**
Batch Processing (store and process later)	External Memory Algorithms and Systems	Batch Parallel Algorithms Hadoop/Spark MapReduce
Real-time Stream Processing (process data now)	Sequential Streaming Algorithms Some CEP/Streaming Systems	Parallel & Distributed Streaming Algorithms IBM Infosphere Streams Apache Spark Streaming Apache Flink

Our Research

- Enumerating complete and large dense structures
  - maximal cliques, maximal bicliques
- Counting and Enumerating small dense structures
  - triangles, small-sized cliques
- Incomplete dense structures
  - quasi-clique, densest subgraph, etc

Our Research

- Enumerating complete and large dense structures
  - maximal cliques, maximal bicliques
- Counting and Enumerating small dense structures
  - triangles, small-sized cliques
- Incomplete dense structures
  - quasi-clique, densest subgraph, etc

Maximal Clique Enumeration (MCE)

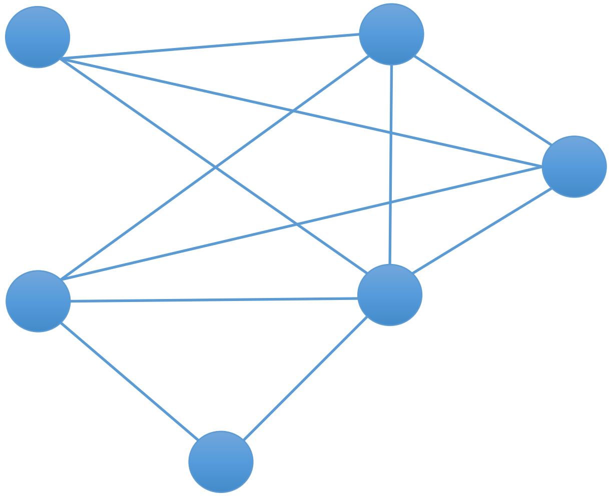
- A clique C in a graph G=(V,E) is a subset of V such that there is an edge between each pair of vertices in C

- A clique C is maximal if it is not contained within any other clique in G

- Maximal Clique Enumeration Problem:
  Given an undirected graph G = (V,E), enumerate all maximal cliques

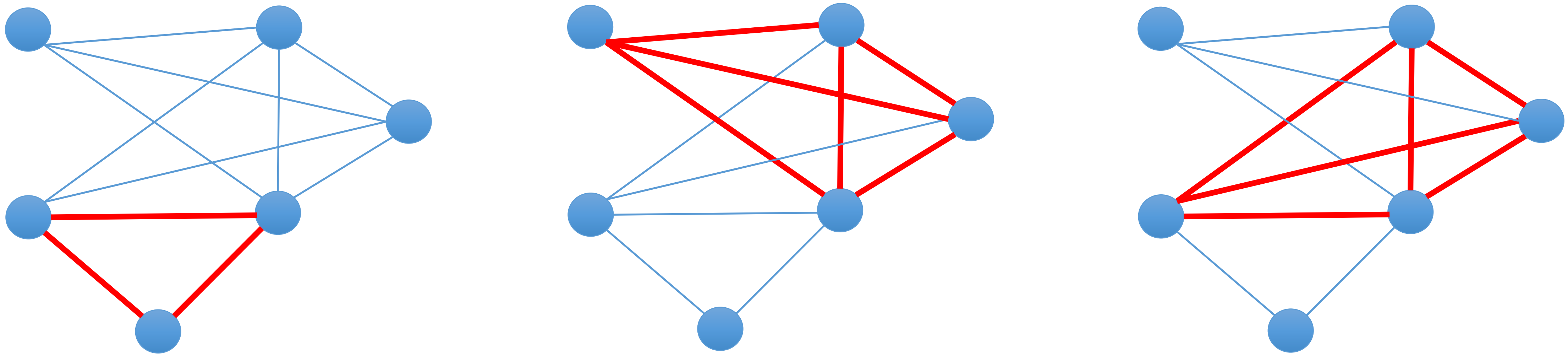
Maximal Clique Enumeration (MCE)

Given an undirected graph G = (V,E), enumerate all maximal cliques in G



Maximal Clique Enumeration (MCE)

Given an undirected graph G = (V,E), enumerate all maximal cliques in G

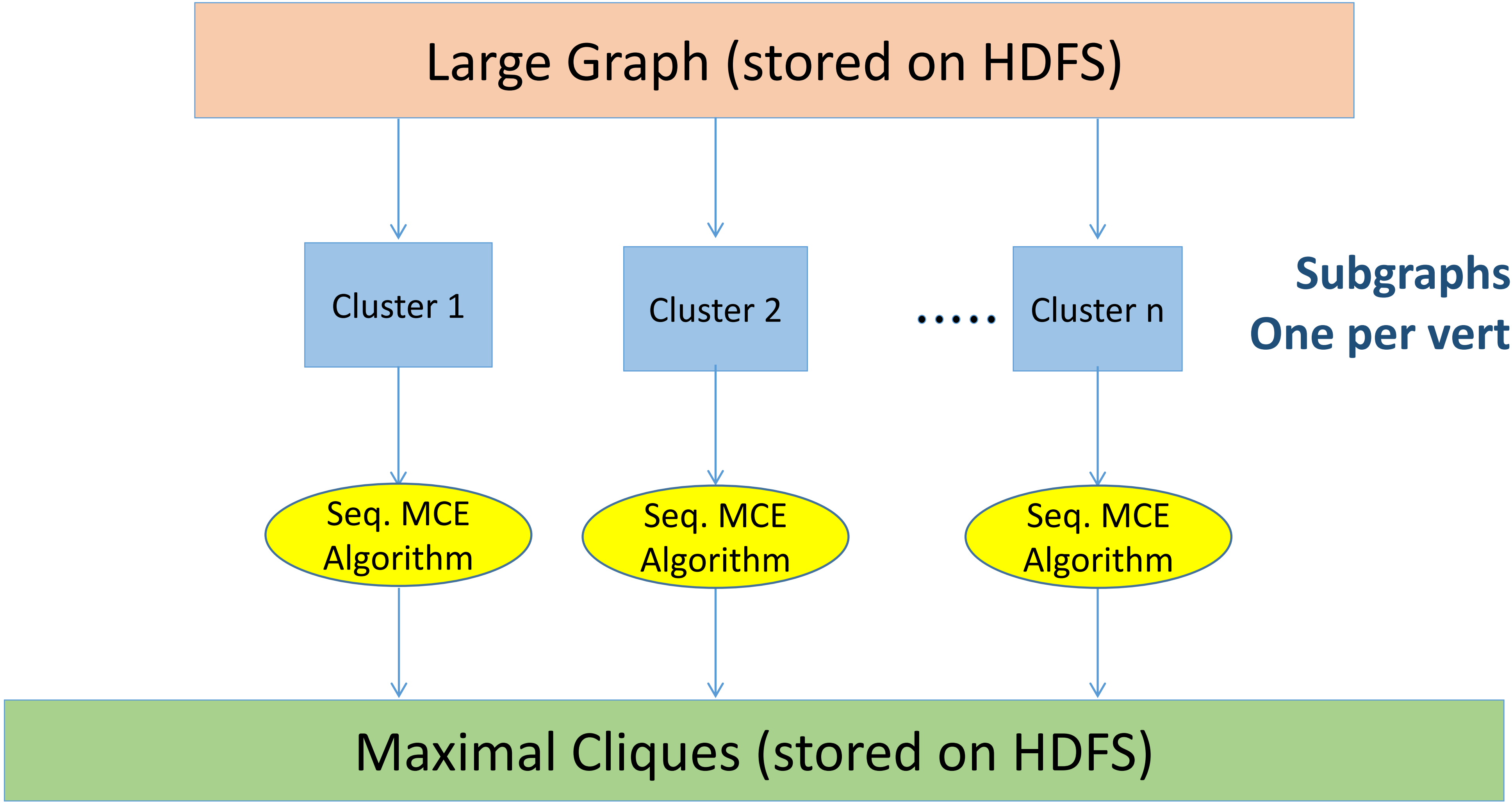


Problem 1: Batch MCE from a Large Graph

- Challenges:
  - Graphs may be small and fit within memory of a single machine, but the number of structures can be large
  - Even moderate graphs, beyond the capacity of a single processor

- Approach:
  - Use parallel computing through Mapreduce (Hadoop or Spark)
    - Can be easily extended to a similar platform (Giraph)
  - Challenges:
    - How to divide into subproblems?
    - How to balance load across processors?
    - How to avoid redundant computations among processors?

PECO: Parallel Enumeration of Cliques using Ordering



Subgraphs
One per vertex

Ideas for PECO Parallel MCE (1)

- Multiple subproblems processed in parallel, one per vertex
  - Done naively, can lead to severe skew in subproblem sizes (experiments show 400:1 skew in subproblem sizes)

- Total order among all vertices in the graph. Design a function "rank" such that: if u is part of more maximal cliques than v, then rank(u) > rank(v)

- Load balancing: assign responsibilities for a vertex so that higher rank vertices are responsible for fewer maximal cliques they are part of

Ideas for PECO Parallel MCE (2)

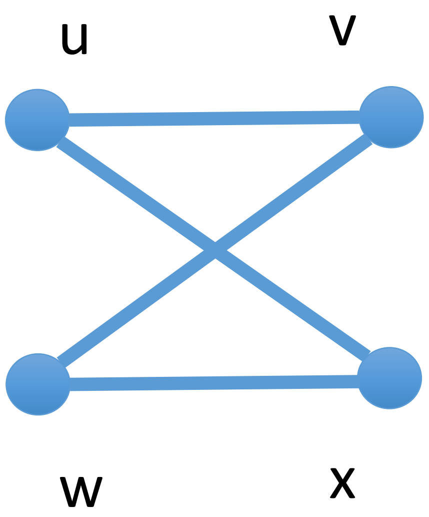
- Each subproblem solved using a sequential Algorithm for MCE:
  - Based on DFS and Pivoting due to Tomita et al. (TTT 2006)
  - Avoid overlap among subproblems by using the ordering in conjunction with a variant of TTT
- Our Parallel Algorithm is work-efficient i.e. its total computation cost is equal to that of a sequential execution

Results on Parallel MCE (PECO)

- **Mining Maximal Cliques from a Large Graph using MapReduce: Tackling Highly Uneven Subproblem Sizes**
  Michael Svendsen, Arko Mukherjee, Srikanta Tirthapura
  Journal Parallel and Distributed Computing (Special Issue for Big Data), 79: pages 104-114, 2015

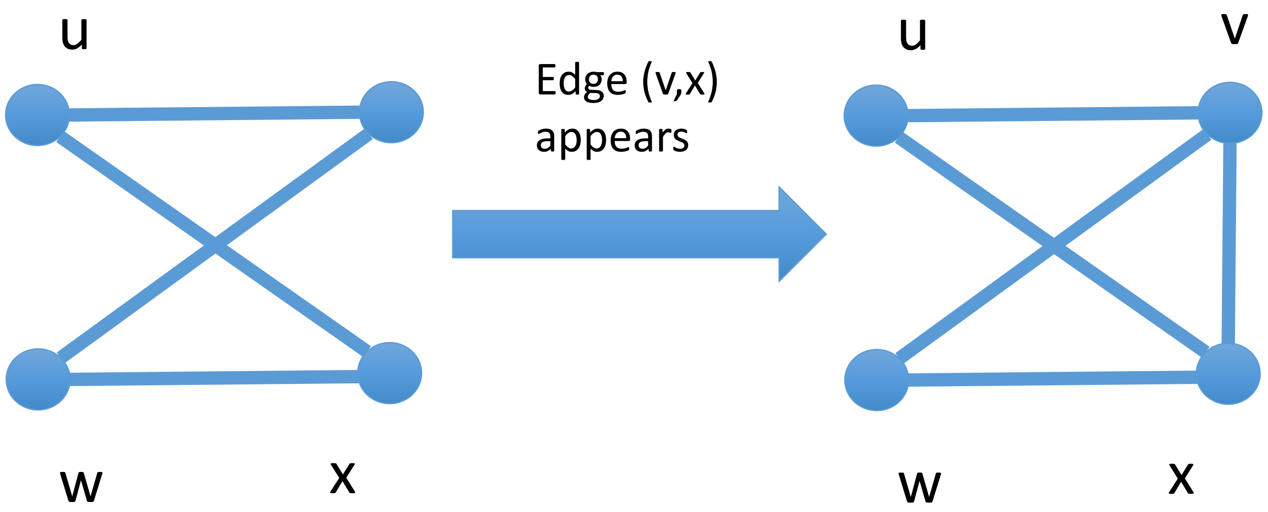
Problem 2: Dynamic MCE

Track Emerging (and Disappearing) dense structures in a large dynamic graph



Dynamic MCE

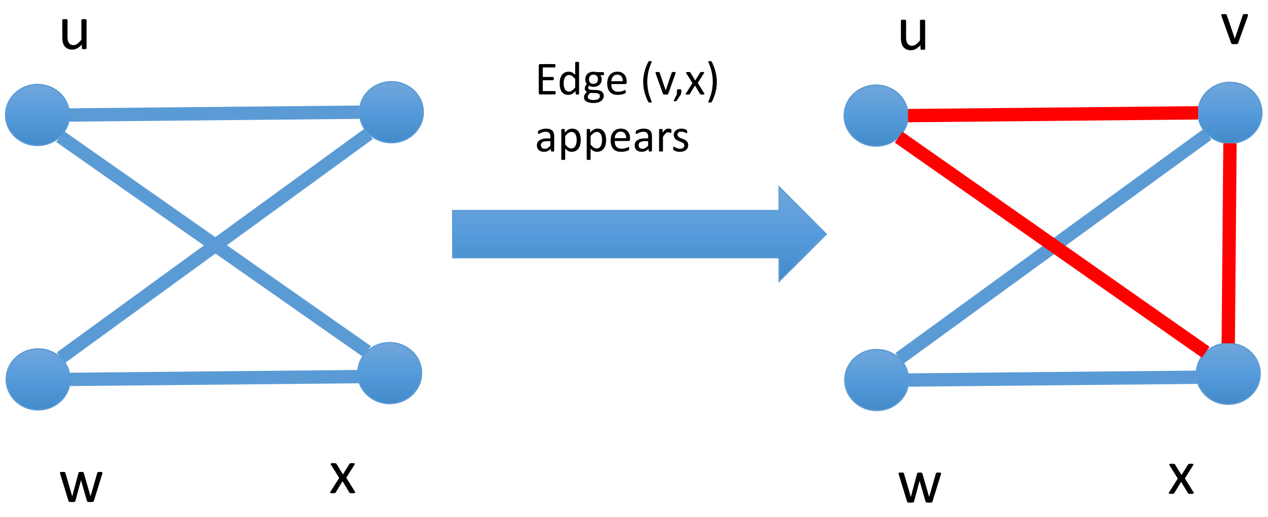
Track Emerging (and Disappearing) dense substructures in a large dynamic graph



Edge (v,x) appears

Emerging Maximal Cliques

Track Emerging (and Disappearing) dense substructures in a large dynamic graph

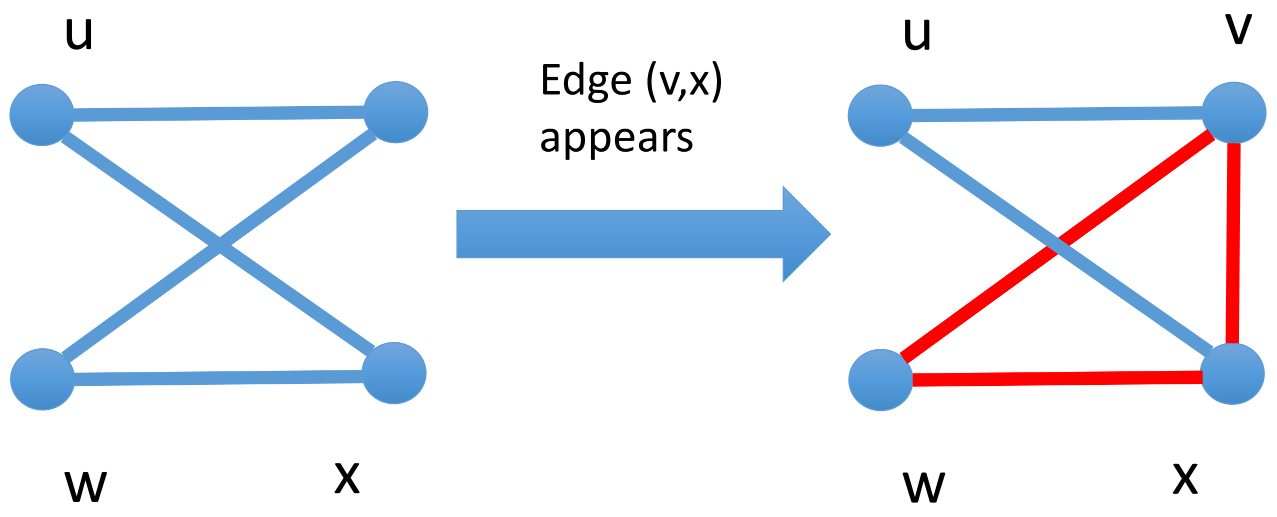


Edge (v,x) appears

Clique (u,v,x) has emerged

Emerging Maximal Cliques

Track Emerging (and Disappearing) dense substructures in a large dynamic graph

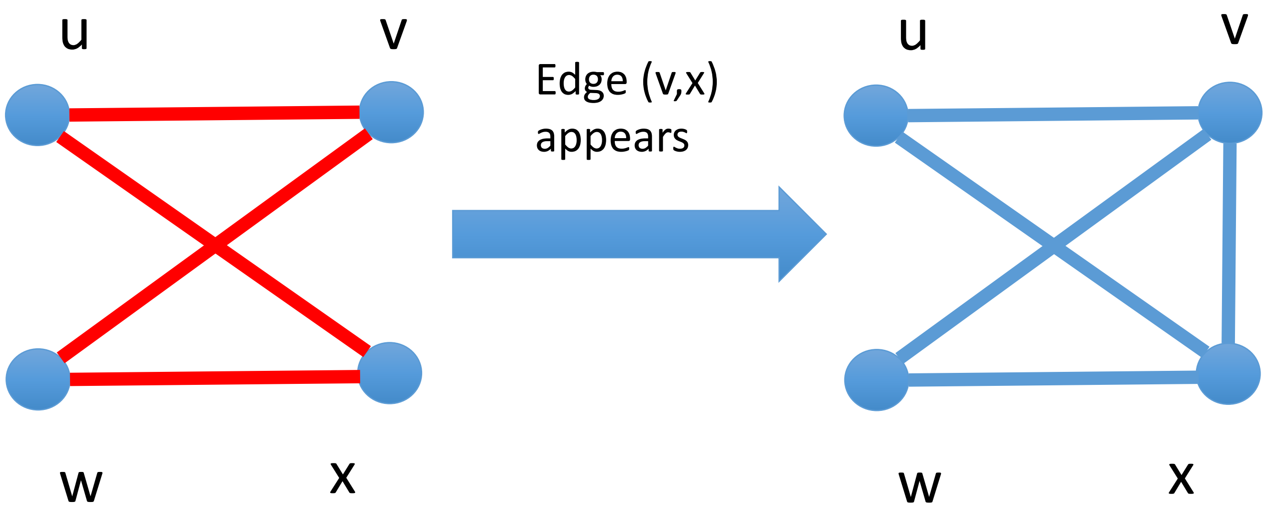


Clique (u,v,x) has emerged

Clique (w,v,x) has emerged

Subsumed Maximal Cliques

Track Emerging (and Disappearing) dense substructures in a large dynamic graph



Clique (u,v,x) has emerged

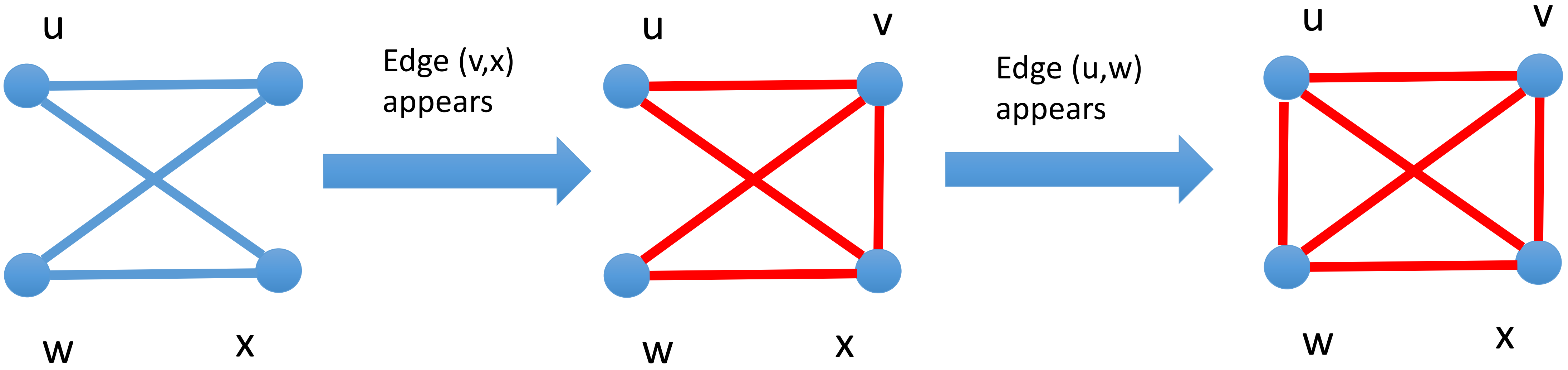
Clique (w,v,x) has emerged

Cliques (v,w) (u,v) (u,x) (w,x) are subsumed by other cliques

New and Subsumed Maximal Cliques

Track Emerging (and Disappearing) dense substructures in a large dynamic graph

(u,v,x) and (v,w,x) subsumed
(u,v,w,x) has emerged



Edge (v,x) appears

Edge (u,w) appears

Dynamic MCE

Suppose we went from graph G to graph G+H through the addition of edge set H.

Let N(G,G+H) = set of new cliques, and
S(G,G+H) = set of cliques that are subsumed

1. Magnitude of Change: How large can N(G,G+H), S(G,G+H) be?
2. Enumeration of Change: How to enumerate N and S without enumerating Cliques(G) and Cliques(G+H)?
3. Can we enumerate in a change-sensitive manner (time proportional to size of change)?

Our Results on Dynamic MCE (Magnitude)

Let f(n) denote the maximum number of maximal cliques in a graph on n vertices (Moon and Moser 1965 provide tight bounds on f(n))

1. When a single edge is added to a graph, maximum change in maximal cliques can be as large as c.f(n) where c > 1
    - Bound is tight (for a single edge)

2. Near-tight results for arbitrary edge additions showing that the maximum change due in maximal cliques can be as large as $\approxeq$ 2f(n)

3. Found an error in the 50 year old result of Moon and Moser on graphs containing the maximum number of maximal cliques

Results: Change Sensitive Enumeration Algorithm

Suppose g(G,H) new maximal cliques were formed through the addition of edge set H to graph G. An algorithm to:

1. Enumerate new cliques in time $O(d^3m\ g(G,H))$
2. Enumerate subsumed cliques in time $O(d^2 2^m\ g(G,H))$

   where d is the maximum vertex degree in G, and m is the number of edges

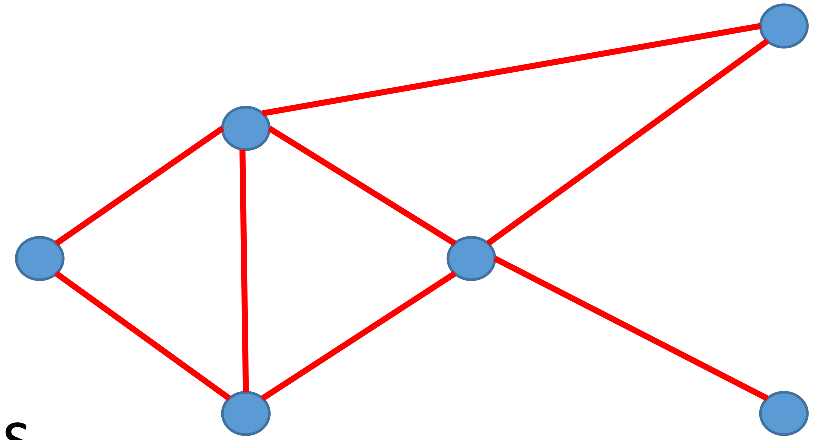
3. Experimental results show these are better than prior work (Stix 2004 and Ottosen-Vomlel 2010) by a factor of 1000

Results on Dynamic MCE

- Apurba Das, Michael Svendsen, Srikanta Tirthapura: **Change-Sensitive Algorithms for Maintaining Maximal Cliques in a Dynamic Graph.** CoRR abs/1601.06311 (2016)

Triangle Sampling and Counting

- A triangle is a triple of vertices (u,v,w) such that {u,v} {v,w} and {u,w} are all adjacent to each other



- Problem #1: Sample from the set of all triangles in a graph

- Problem #2: Count the number of triangles in a simple undirected graph

Neighborhood (Chain) Sampling

Two edges are adjacent
if they share a vertex

- Choose a random edge r_1 in the graph (using reservoir sampling on the entire stream of edges)

- Choose a random edge r_2, that appears after r_1, and is adjacent to r_1 (using reservoir sampling on substream decided by choice of r_1)

- See if triangle defined by $r_{1,}$ r_2 is completed by a third edge

- Produces a biased sample, but bias can be handled using rejection sampling

Streaming Graph Sampling and Applications

- Presented a fast and accurate method (Neighborhood Sampling) to count triangles in a Streaming Graph
  - 100 times faster than previous methods
  - Relative error much smaller, using same memory

- Memory does not increase with the size of the graph, only with desired accuracy, and with graph structure

- Effective Use of Parallelism
  - Process a 167GB graph in 1000 seconds, on 12 core machine
  - **Counting and Sampling Triangles from a Graph Stream**
    A. Pavan, Kanat Tangwongsan, Srikanta Tirthapura, Kun-Lung Wu
    In *Proc. 40th International Conference on Very Large Databases (VLDB) 2014*
  - **Triangle Counting in a Massive Streaming Graph Using a Multicore Machine**
    Kanat Tangwongsan, A. Pavan, and Srikanta Tirthapura, "
    in Proc. *ACM Conference of Information and Knowledge Management* (**CIKM**) pages 781—78

Other Structures / Extensions

- **Enumerating Maximal Bicliques from a Large Graph using MapReduce**
  Arko Mukherjee and Srikanta Tirthapura
  IEEE Transactions on Services Computing (to appear)

- Combining Parallel and Streaming in Enumeration
  Prior work on counting small dense structures (triangles and relatives)
  Parallel Triangle Counting in Massive Streaming Graphs
  Kanat Tangwongsan, A. Pavan, Srikanta Tirthapura
  Proc. ACM Conference on Information and Knowledge Management (CIKM)
  The full version is available at http://arxiv.org/abs/1308.2166

Conclusion

- Algorithms for Enumerating the Change in the set of complete dense structures in a dynamic graph
  - Bounds on the Magnitude of Change
  - Change-Sensitive Algorithms
- Work-Efficient Parallel Algorithms for enumerating dense structures from a large graph
- Orders of magnitude speedup compared to prior work

Questions for Future Research

- Maximal Cliques
    - Improved Change-Sensitive Algorithms for Dynamic Clique Enumeration
- Other Structures (especially incomplete dense structures)
- Uncertain/noisy graphs
- Parallel combined with Streaming
- Counting without Enumerating