Distributed Exact Subgraph Matching in Small Diameter Dynamic Graphs

Charith Wickramaarachchi†, Rajgopal Kannan†
Charalampos Chelmis*, and Viktor K. Prasanna†

†University of Southern California
*University at Albany SUNY

cchelmis@albany.edu
Outline

• Subgraph Matching
• Dynamic Graphs
• Exact Subgraph Matching in Dynamic Graphs
• Graph Pruning
• Evaluation
• Conclusion and Future Work
Subgraph Matching

Input: Data graph G (V, E, L), Query graph Q (V^q, E^q, L^q)
Output: All subgraphs in G that “match” Q

• Applications
  – Search in OSN, Knowledge graphs, plagiarism detection, …

• Matching Models
  – Subgraph Isomorphism
    • Exact matching
  – Graph Simulation
    • Relaxed matching
Subgraph Isomorphism

Q matches data graph G iff there exists subgraph \( G^s \subseteq G \) and bijective function \( f: V^Q \rightarrow V^S \) such that

- any node \( v \in V^q \) and \( f(v) \in V^S \) have the same label
- An edge \((v_i, v_j) \in E^q \) exists iff \((f(v_i), f(v_j)) \in E^S\)
Graph Simulation

Subgraph Simulation:
• Q matches data graph G if a binary relation $R \subseteq V^q \times V$ exists such that
  – 1) if $(u, u') \in R$ then $l^q(u) = l(u')$;
  – 2) $\forall (u, v) \in E^q, \exists (u', v') \in E: (u, u') \in R$;
  – 3) $\forall u \in V^q, \exists u' \in V: (u, u') \in R$

Subgraph Dual Simulation:
• Q matches data graph G if
  – 1) Q matches G via graph simulation under a match relation $R_D \subseteq V^q \times V$;
  – 2) $\forall (u, u') \in R_D [(w, u) \in E^q \Rightarrow \exists w' \in V: (w, w') \in R_D \wedge (w', u') \in E]$
Dynamic Graphs

- Graphs that change with time
  - Dynamic Network, e.g., $G=(V,E,W)$, $W = \{w_1, w_2, \ldots, w_T\}$
  - Evolving Graph, e.g., $G^t = (V^t,E^t)$
  - Temporal Graph, e.g., graph sequence $\{G_1, \ldots, G_T\}$
  - Graph Stream, e.g., $G^{(s)} := \{G^{(t_a)}, G^{(t_a+1)}, \ldots, G^{(t_{a+1}-1)}\}$
  - Networks Sequence, i.e., $\{G^t\}_{t=0,\ldots,T}$
  - Time-Series Graphs

- Usually, used for event detection

Source: NetSpot: Spotting Significant Anomalous Regions on Dynamic Networks
Source: GED: the method for group evolution discovery in social networks
**Subgraph Matching in Dynamic Graphs**

- **Focus:** How to maintain a set of subgraph matches in a dynamic graph?

- Let $M_t$ be the set of subgraphs in $G_t$ that match query graph $Q$ via subgraph isomorphism

- An **incremental** subgraph matching algorithm takes $G_t$, $\Delta e_u$ and $M_t$ as input, to produce $M_{t+1}$ for $G_{t+1}$ by computing the changes $\Delta M$ to set $M_t$

- **Observation:**
  - The set of subgraphs that can be potentially affected by an edge update is within a **radius** from the edge update
  - This “neighborhood” is bounded by the query **diameter**

- **Assumption:** $D_Q \ll D_G$
A Distributed Incremental Algorithm

- **Goal:** Leverage already computed results to minimize unnecessary re-computations
- **Solution:** Framework that re-uses legacy SIM libraries developed for small static graphs
- **Edge updates** are processed in batches
- **Practitioner** assigns edge updates to workers
- Each worker is responsible for maintaining a disjoint partition of $G$
- When an update arrives at a worker
  - Distributed depth limited BFS (may span to graph portions stored in other workers)
  - Affected subgraph is copied to worker
  - Matches are made on the affected subgraph
  - Matches are sent to the Reducer

Existing subgraph isomorphism libraries can be used to find matches

Parallelism at multiple levels
- Edge updates are processed in parallel
- Distributed subgraph construction for each edge update
Graph Pruning

• Performance can degrade with increasing query graph diameter
• Even more so in **small diameter** graphs
  – An edge update can affect subgraph matches in a major portion of the graph
  – The size of the subgraph constructed by the distributed depth limited BFS can grow fast with increasing \( D_Q \)
• Solution:
  – Distributed algorithm to maintain a pruned graph based on **dual simulation** in a dynamic graph and the notion of **safe edges**
  – Reduces the size of \( G \) to be searched for matches
  – Reduces communication overhead of subgraph construction
• Algorithm follows the BSP model
• Complexity
  – \( O(E_{\text{SCC}}) \) super-steps, where \( E_{\text{SCC}} \) is the number of edges in the largest strongly connected component
Evaluation - Setup

- Implementation
  - C++, MPI (MPICH2)
  - VF2 is used for subgraph matching within neighborhood subgraphs
  - Lemon graph library for efficient implementation of graph data structure

- Platform
  - Amazon EC2 – c3.2xlarge
  - 5 nodes

- Datasets

| Dataset                      | | V | | E | Type             |
|------------------------------|---|----|----|------------------|
| California R/N (CA)          |   | 1,965,206 | 2,766,607 | Large diameter   |
| Central USA R/N (CTR)        |   | 14,081,816 | 34,292,496 | Large diameter   |
| Full USA R/N (USA)           |   | 23,947,347 | 58,333,344 | Large diameter   |
| DBLP network (DBLP)          |   | 317,080    | 1,049,866  | Large diameter   |
| YouTube (YT)                 |   | 4,945,382  | 49,445,382 | Small-diameter   |
| Live Journal (LJ)            |   | 5,284,457  | 77,402,652 | Small-diameter   |

- Query graphs: D = 1,2,3 (|V| = 5,12,17)
Evaluation Results (1)

- Latency comparison with/without graph pruning
- Baseline: **Sequential exact** subgraph isomorphism algorithm [TODS2013]

**Facilitated** incremental exact matching on large diameter networks
Pruning **significantly improves** performance on small world networks
Evaluation Results (2)

- Average % reduction in graph size
Evaluation Results (3)

- Latency with increasing number of workers

Latency (sec)

- Log

d=1
- d=2
- d=3

DBLP  YT  LJ  CA  CTR  USA  DBLP  YT  LJ  CA  CTR  USA  DBLP  YT  LJ  CA  CTR  USA

W=8  W=16  W=32

UCS
School of Engineering
University of Southern California
Conclusion and Future Work

• Proposed a query preserving distributed graph pruning approach (D-IDS) to enable **exact subgraph matching** in **small diameter dynamic** graphs
• Graph pruning resulted in over 60% reduction in graph size in real-world networks
• Significantly improved the performance of neighborhood search based subgraph matching for small diameter graphs
• Future work
  – Examine impact of graph partitioning strategies
  – Study effect of update rate in a variety of dynamic settings
Questions?

• Thank you!