CS698F Advanced Data Management

Instructor: Medha Atre

Announcements

- Assignment-2 paper selection due **tonight**: 20-Oct-2017, 23:59
- Assignment-2 presentations:
  - ~~25, 27 Oct 2017~~ Nov 1, 3 in class
  - Course project report and code due: 14-Nov-2017, 23:59
  - In class presentations and demo on 15, 17 Nov, 2017
  - **No extension to this allowed!**
- Endsem written exam: 18-Nov-2017 16:00-19:00

Reachability queries

- Given a graph G(V, E), where V is the set of nodes and E is the set of edges, a reachability query asks:
  - Does there exist "any" path between nodes *x* and *y,* where x and y are two nodes in the graph.
- In case of directed graph, the path is directed and other for undirected graphs the path is undirected.
- The graphs may have cycles too, giving rise to “strongly connected components”.

Challenges

- Identify strongly connected components (SCC) and collapse them
  - SCC is where every node in a given subgraph can reach every other node.
- Efficient traversal over large graphs.
- High computational complexity $O(V^3)$ for standard algorithms like all-pair shortest path, which also gives us transitive clousure for answering reachability.
- High storage space requirement.

Brute-force methods

- After collapsing all the SCCs, do a DFS walk over the graph and maintain reachability "map".
- Optimizations on how to maintain this "map"?
  - Compressed bit-vectors [SIGMOD 2011]
  - Partial maps (e.g., 2-hop cover) [SODA 2002]
  - Other index structures, such as Interval Labeling, hyperdimensional interval labeling etc.

Compressed bit-vectors

- Do a DFS walk on the graph.
- Assign nodes to bit-positions as they are visited.
- This ensures that reachability bit-vector is "densely packed" for most real-life practical graphs.

Interval Labeling

- Each graph node has a an interval [x, y] associated with it.
  - This interval is decided after traversing the graph first.
- To decide reachability
  - Node "t" is reachable from "s" *iff* $[x_t, y_t]$ is completely contained in $[x_s, y_s]$
    - e.g., let t's interval be [3, 5] and s's interval be [1, 10], then node 't' is reachable from 's'
    - Works only for "undirected graphs"!
    - Does NOT work for directed graphs! Why?

Key aspects to consider

- Index creation time.
- Index size.
- Query answering time.
- High index creation time – one time cost.
- High index size – can be disk-resident.
- High query answering time – *needs to be mitigated*.