**Summary**

- **Goal:** to compactly represent graphs with tens or hundreds of billions of edges
- **Previous Work:**
  - Graph summarization: a promising graph-compression technique
  - Algorithms for summarizing graphs that are small enough to fit in main memory
- **Proposed Algorithm (SWeG):**
  - a parallel and distributed algorithm for compactly summarizing large graphs
  - scales near linearly with the size of the input graph and requires sub-linear memory
- **Results:**
  - Speed: up to 5,400× faster than competitors, with similarly compact representations
  - Scalability: scales to graphs with over 20 billions of edges
  - Compression: achieves up to 3.4× additional compression when combined with other advanced graph-compression techniques

**Motivation**

- **Graph summarization** is a promising graph-compression technique
- Existing algorithms are not satisfactory in terms of speed and compression rates
- Existing algorithms assume that the input graph is small enough to fit in main memory
- **Question:** How can we concisely summarize graphs with tens or hundreds of billions of edges that are too large to fit in main memory or even on a disk?

**Problem Definition: Graph Summarization**

- Example of graph summarization: Notice that it is a lossless process!

### Parallel & Distributed Implementation

- **Map Stage:** compute min hashes in parallel
- **Shuffle Stage:** group supernodes using min hashes
- **Reduce Stage:** process groups independently in parallel

**Complexity Analysis**

- Let the input graph be $G = (V, E)$. For each node group $V_i$ in the grouping step, consider subgraph $G_i = (V'_i, E'_i)$, which induced from $V_i$ and their neighbors.
- **Time Complexity:** $O(T \times \sum_{i \in |V|} |E_i|) = O(T \times |E|)$ if we divide groups finely
- **Memory Requirements:** $O(|V'_i| + \max_{i \in |E|})$

**Further Compression: SWeG+**

- **Main Idea:** further compress the outputs of SWeG, which are three graphs, using any off-the-shelf graph-compression techniques, such as BFS [4], BP [5], and VNMin [6].

**Experimental Results**

- **Dataset:** 13 real graphs (with 10K – 20B edges)
- **Competitors:** summarization algorithms: Greedy [1], Randomized [1], and SAGS [3]
- **Q1. Speed and Compression:** SWeG significantly outperforms its competitors
- **Q2. Scalability:** SWeG is linear in the number of edges & SWeG scales up to 20 billion edges
- **Q3. Compression:** SWeG+ achieves unprecedented compression rates
- **Q4. Memory Requirements:** SWeG loads ≤ 0.1 – 4% of edges in main memory at once
- **Q5. Effects of Iterations (Figure 7 of the paper):** ~20 iterations (i.e., $T = 20$) are enough for obtaining concise outputs

**References**