Benchmarking graph database and processing systems for Agglomerative Clustering

Motivation

- In graph database space, there is an exciting development which combines both a storage and computational model.
- However, processing graph-like data is often mistakenly conflated with graph databases because they share same data model.
- No previous benchmark study between graph databases and graph processing systems on clustering algorithms.

Problem Statement

- Implement a clustering algorithm to find out clusters in social network like graph data.

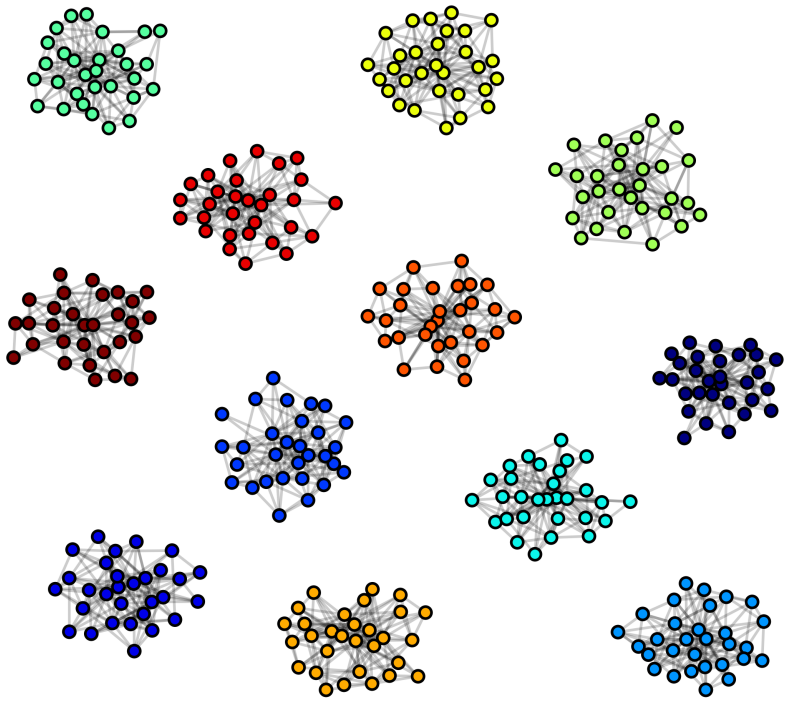
- Performance evaluation of graph databases (Neo4j and GraphLab) in terms of their time of execution on sparse to dense graphs from small to very large datasets.

Data generation

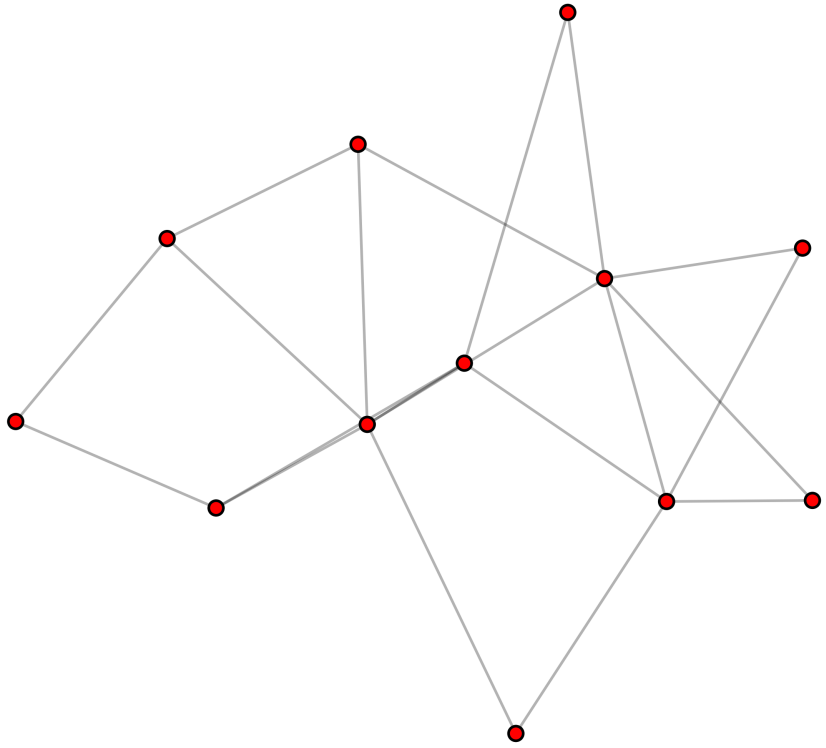
- We generated synthetic social network data using Networkx graph data generator.

- Steps to generate a **Social Network** graph:
  - First, we generate different communities with fixed number of nodes and edges with a given triangle formation probability.
  - Then we take a template of social network like graph and connect those communities based on it.
  - Edges between communities are also decided based on a probability.
  - These probabilities above decide whether the graph is sparse or dense.

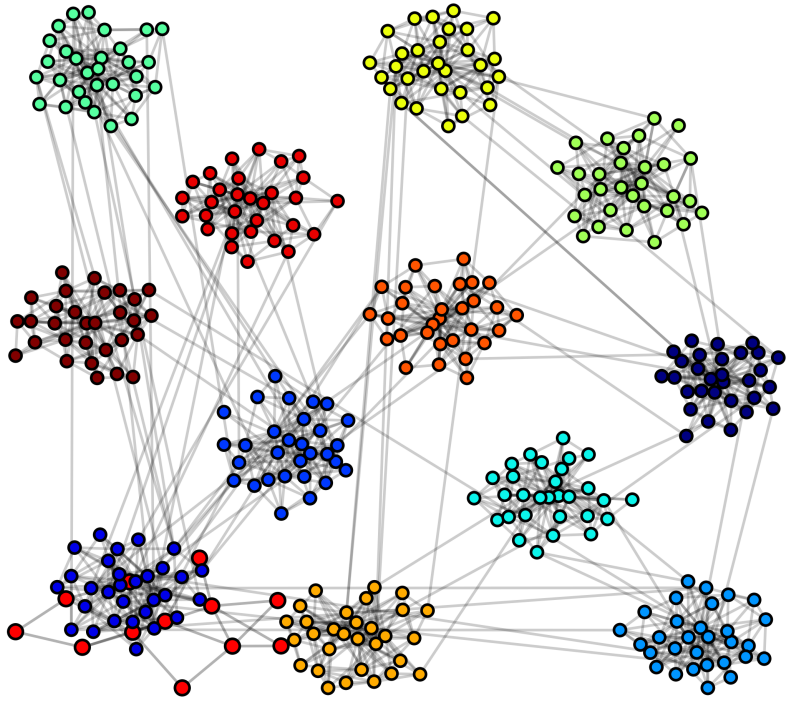
Data generation



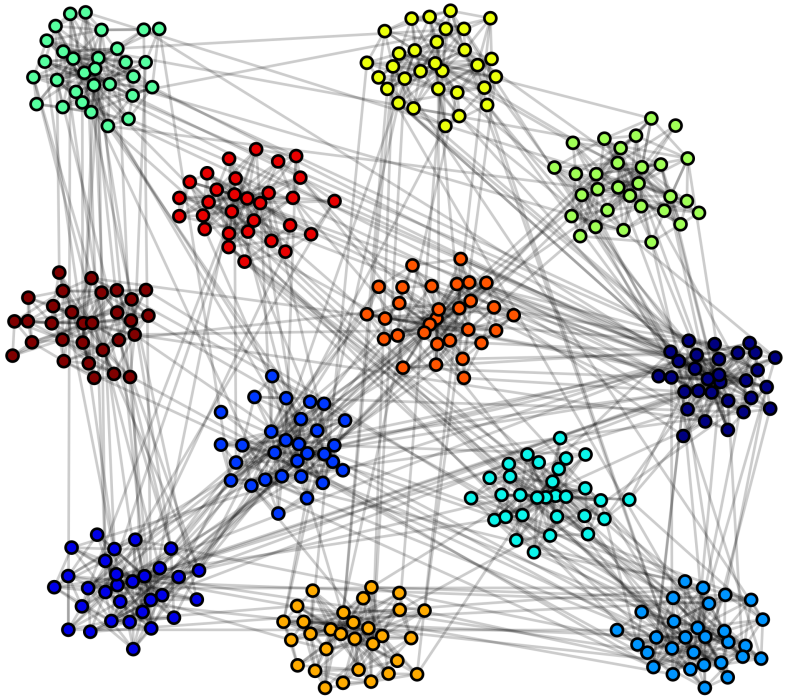
Data generation



Data generation



Data generation



Hierarchical Clustering Algorithm

- Two step clustering algorithm
  - **Step1**: Find out **structural similarity index** of each node from the graph and combine with a existing cluster

$$\sigma(u,v) = \frac{\sum_{x \epsilon \Gamma(u) \cap \Gamma(v)} w(u,x).w(v,x))}{\sqrt{\sum_{x \epsilon \Gamma(u)} w^2(u,x)} \sqrt{\sum_{x \epsilon \Gamma(v)} w^2(v,x)}}$$

  - **Step2**: Compute **average clustering coefficient** of each cluster which is a measure of the degree to which nodes in a graph are clustered.

Clustering Coefficient

- For a graph G(V,E), edge e_{ij} connects vertex v_j with vertex v_j.
- N_i is the neighborhood for a vertex v_i .
- Local clustering coefficient of a node k for undirected graph is defined as

$$C_i = \frac{2|e_{jk} : v_j, v_k \epsilon N_i, e_{jk} \epsilon E|}{k_i(k_i - 1)}$$

- Average clustering coefficient for the whole network is given by:

$$C_{avg} = \frac{1}{n} \sum_{i=1}^{n} C_i$$

Graph Traversal

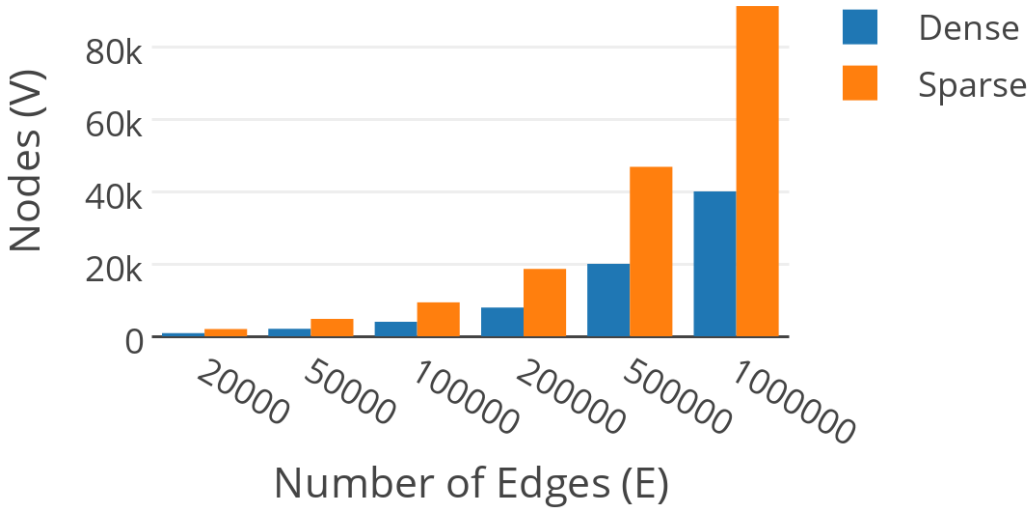
- We will start breadth first search traversal of the graph and combine nodes to form clusters based on its structural similarity index.

- In order to remove biasness induced by the starting access point of the graph, we randomly sample 20 points.

- This will give us the generalized communities rather than skewed set of clusters.

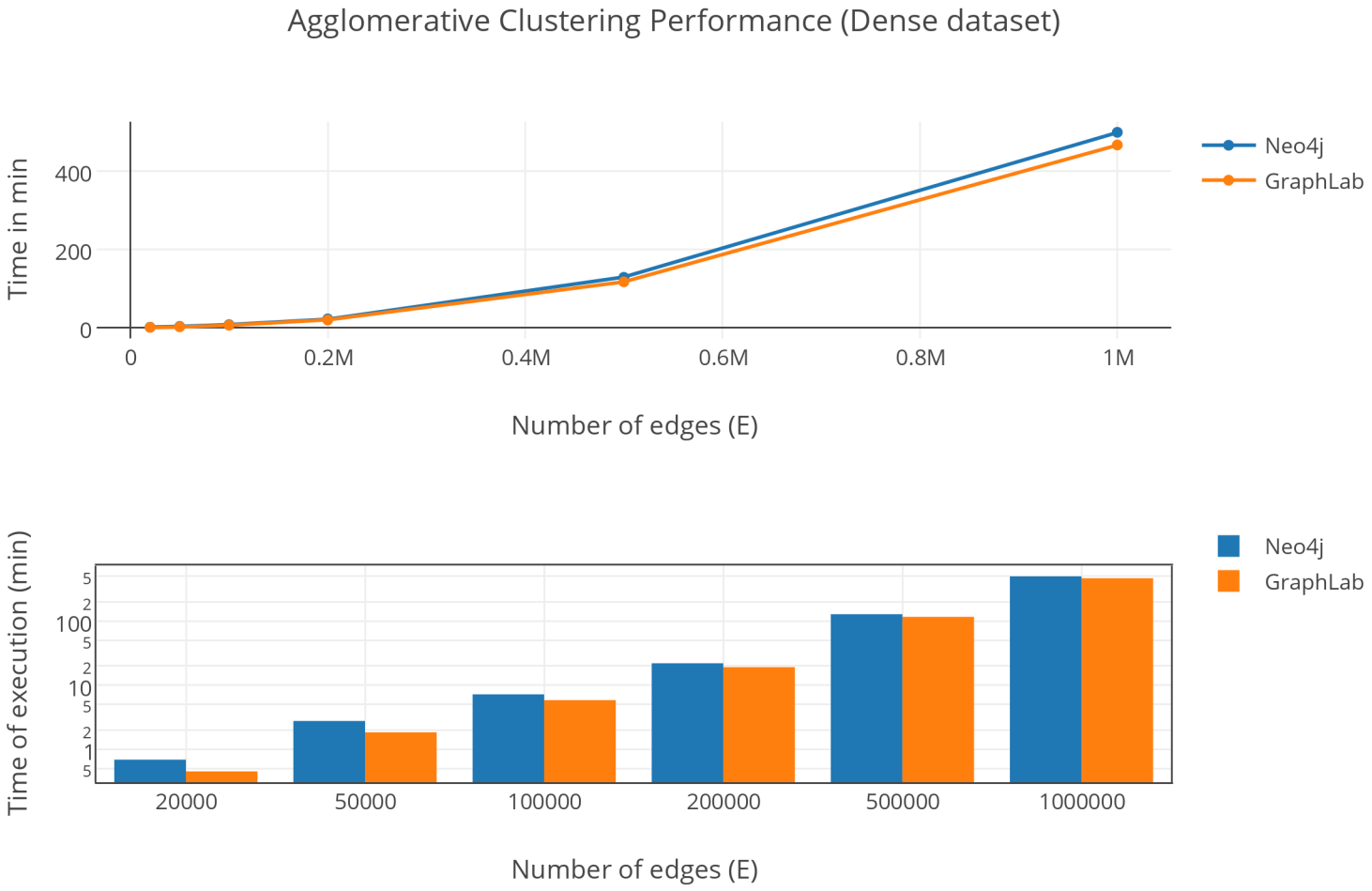
Dataset Statistics

Dataset	Edges/dataset	Dense (Nodes)	Sparse (Nodes)
1	20K	900	2000
2	50K	2100	5000
3	100K	4000	9500
4	200K	8000	19000
5	500K	20000	47000
6	1000K	40000	91000

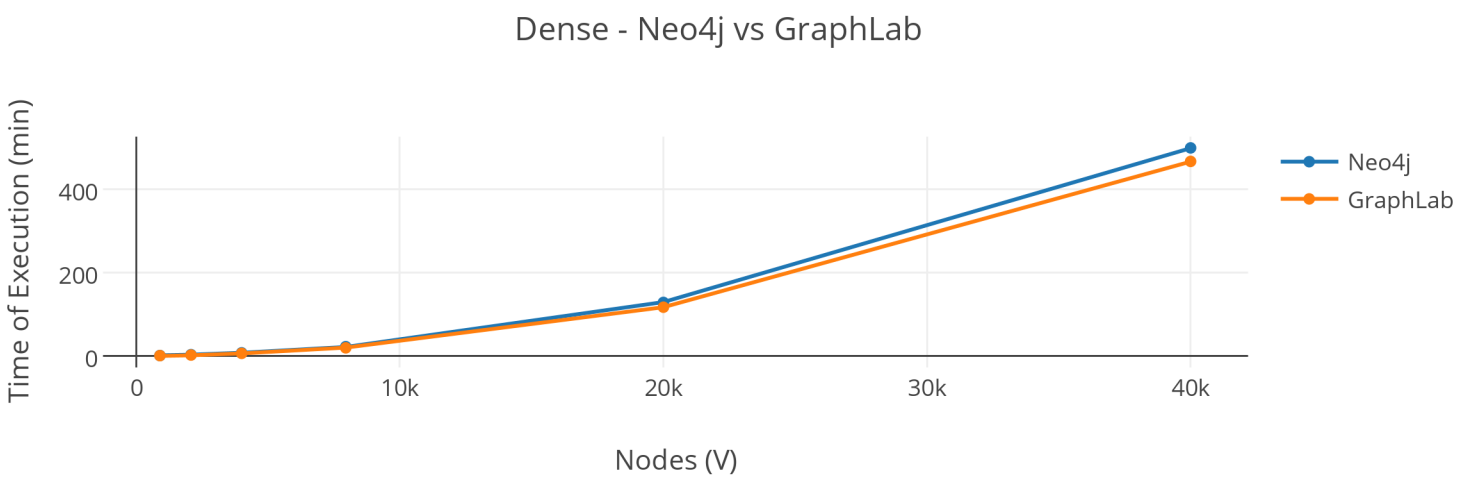


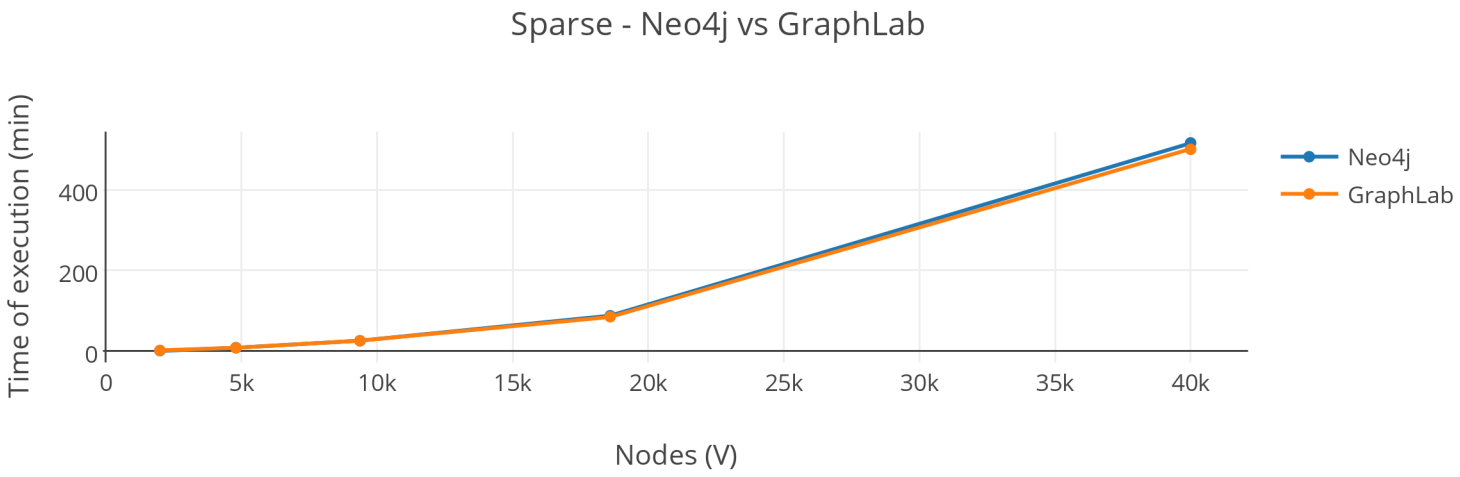
Degree:
Sparse = 12 edges/node
Dense = 25 edges/node

Results - Based on edges



Neo4j vs GraphLab - Based on nodes





Dense vs Sparse datasets