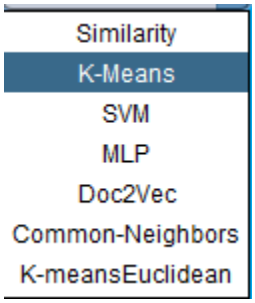
CPS 521 - DSMP Project

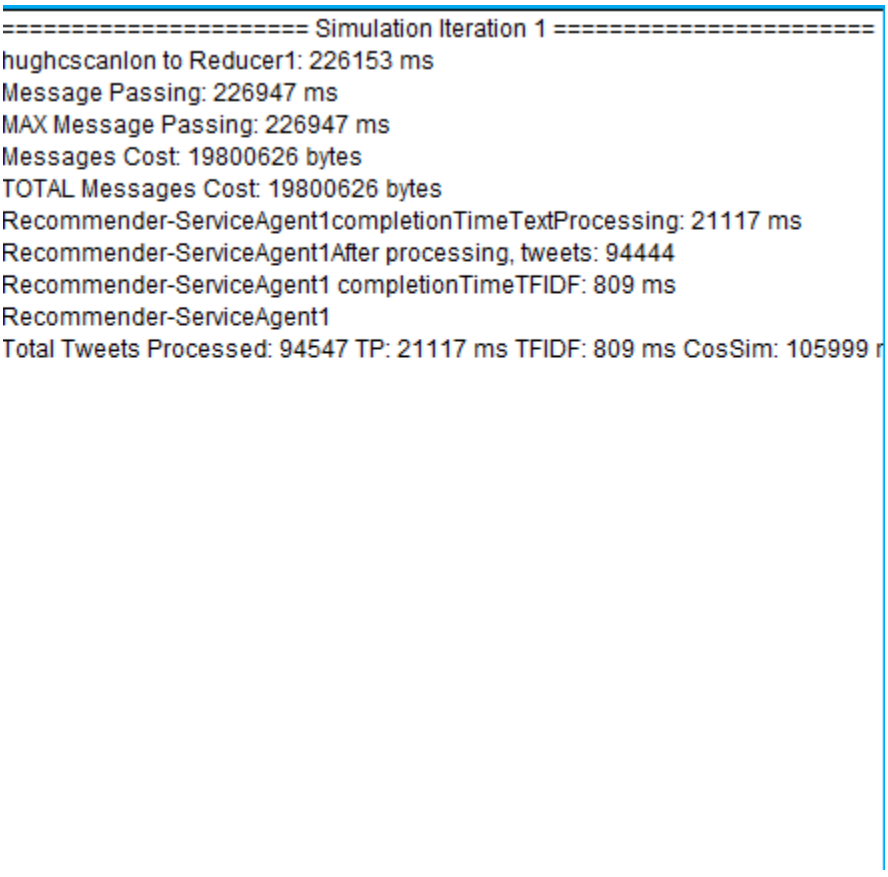
Mustafa Darras 50106030
Ekrem Yilmaz 501108034
Alden Shin-Culhane 501036054

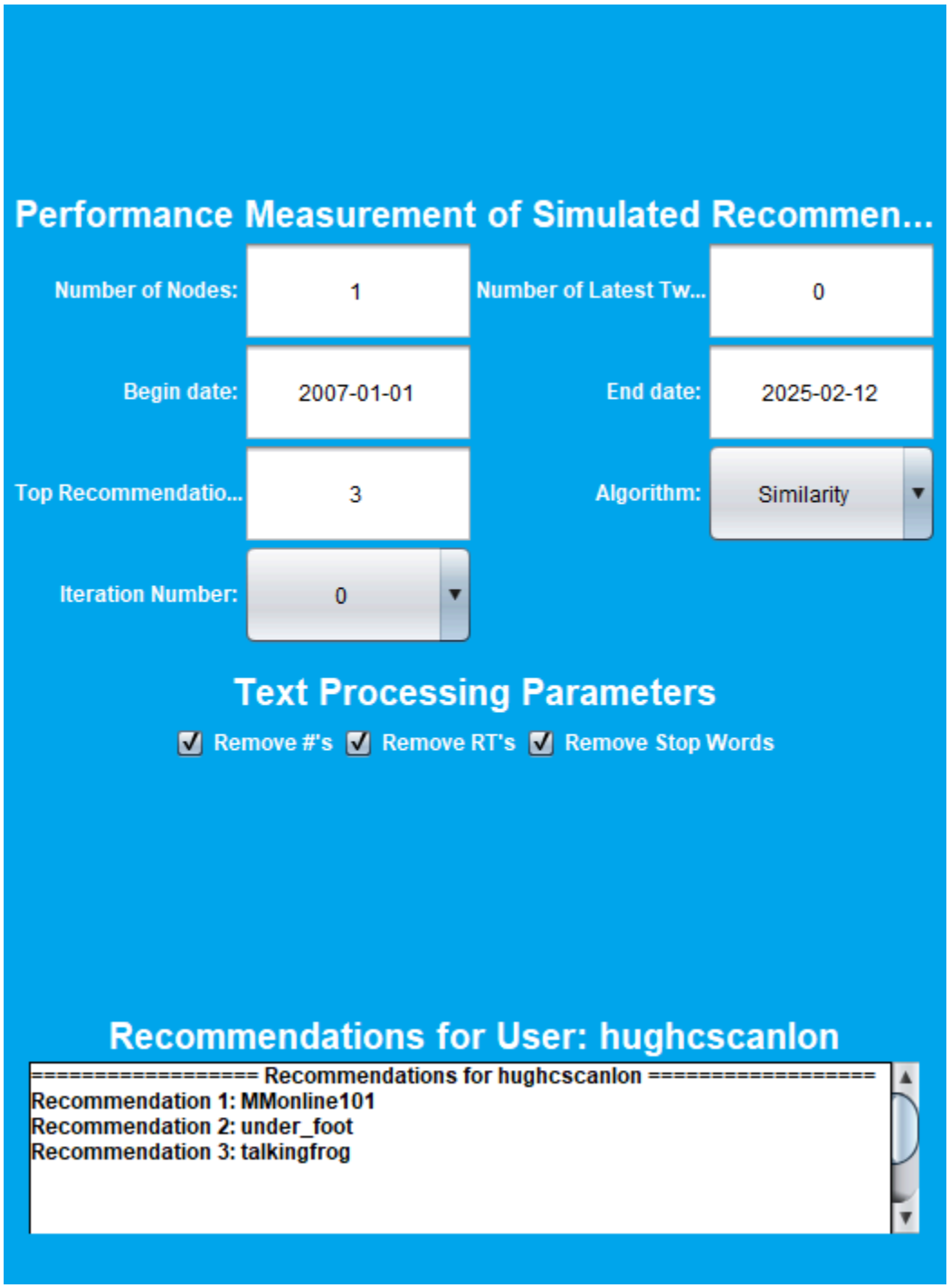
Given Algorithms



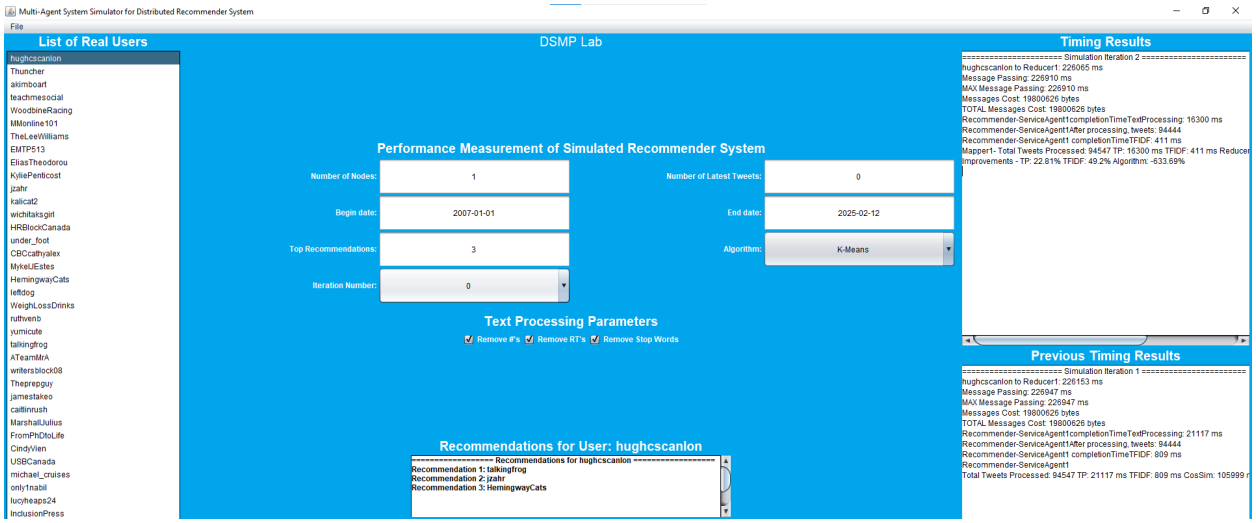
1 Node

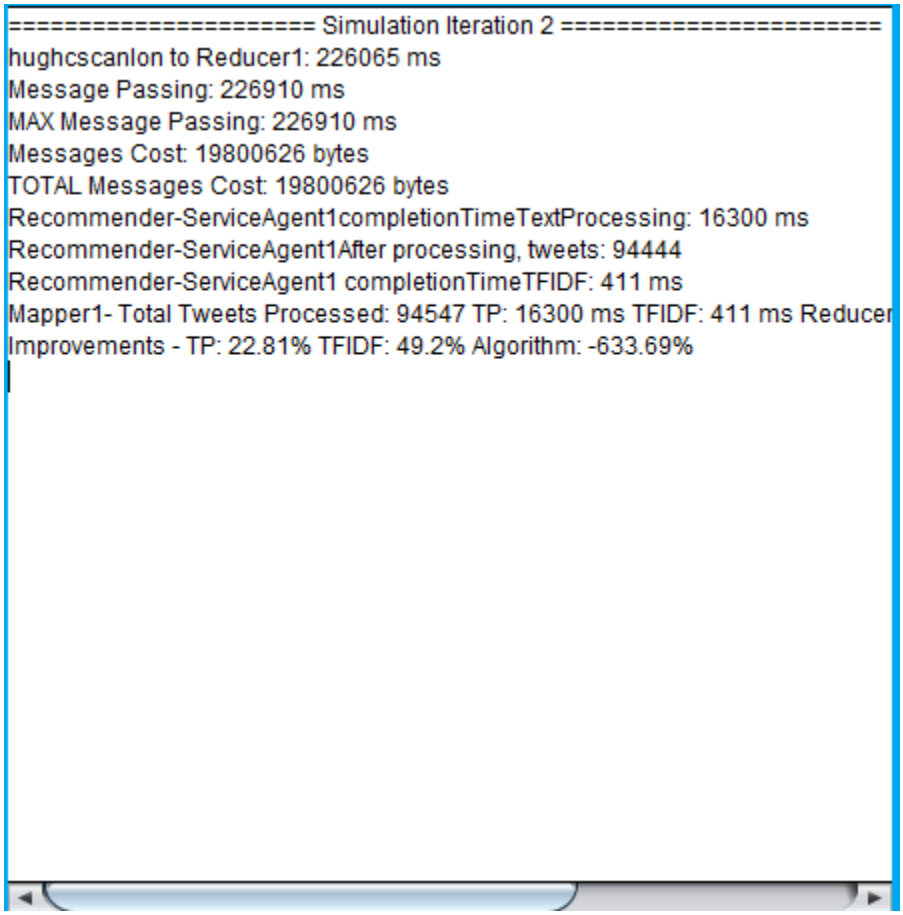
Algorithm 1: Similarity




Performance Measurement of Simulated Recommen...
Number of Nodes:
1
Number of Latest Tw...
0
Begin date:
2007-01-01
End date:
2025-02-12
Top Recommendatio...
3
Algorithm:
Similarity
Iteration Number:
0
Text Processing Parameters
Remove #'s
Remove RT's
Remove Stop Words
Recommendations for User: hughcscanlon
========== Recommendations for hughcscanlon ==========
Recommendation 1: MMonline101
Recommendation 2: under_foot
Recommendation 3: talkingfrog

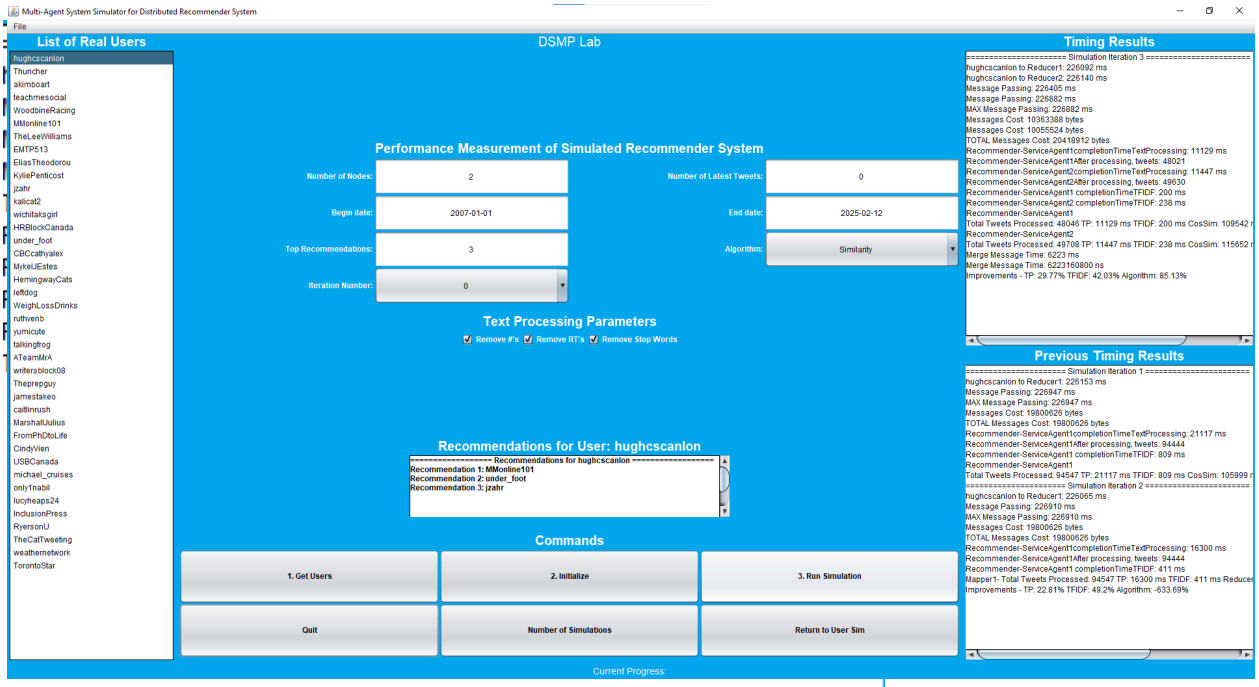
Algorithm 2: K-Means



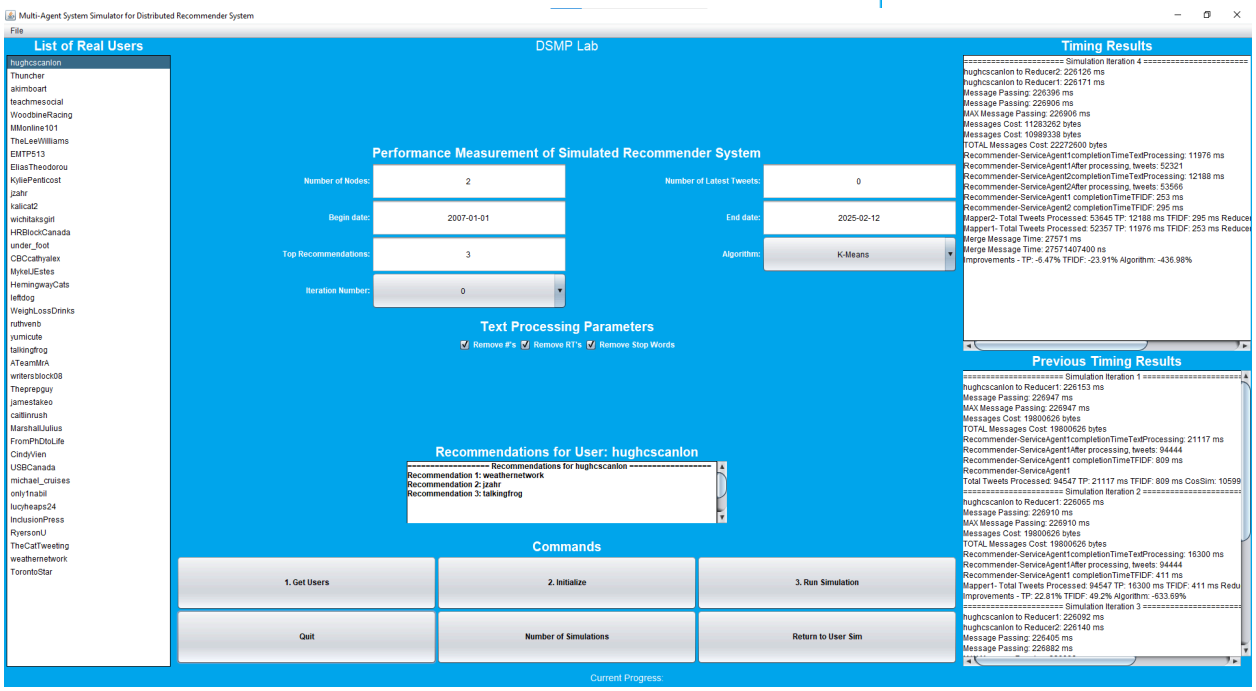


2 Nodes

Algorithm 1: Similarity

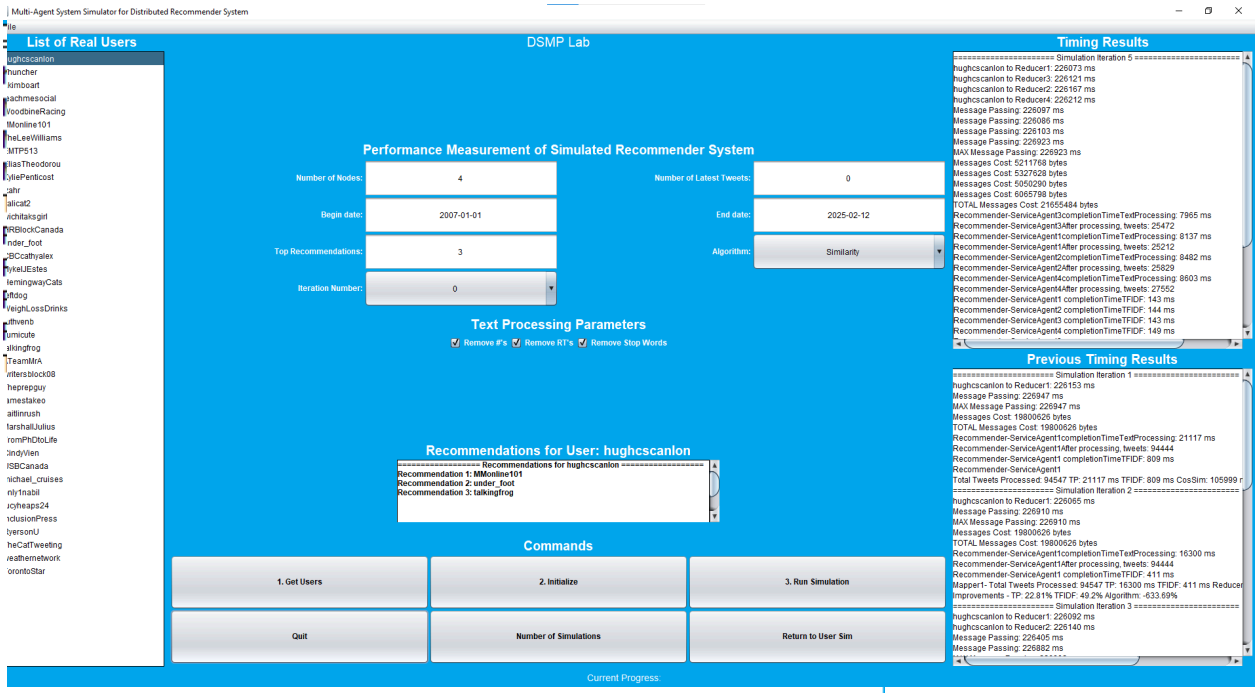


Algorithm 2: K-Means

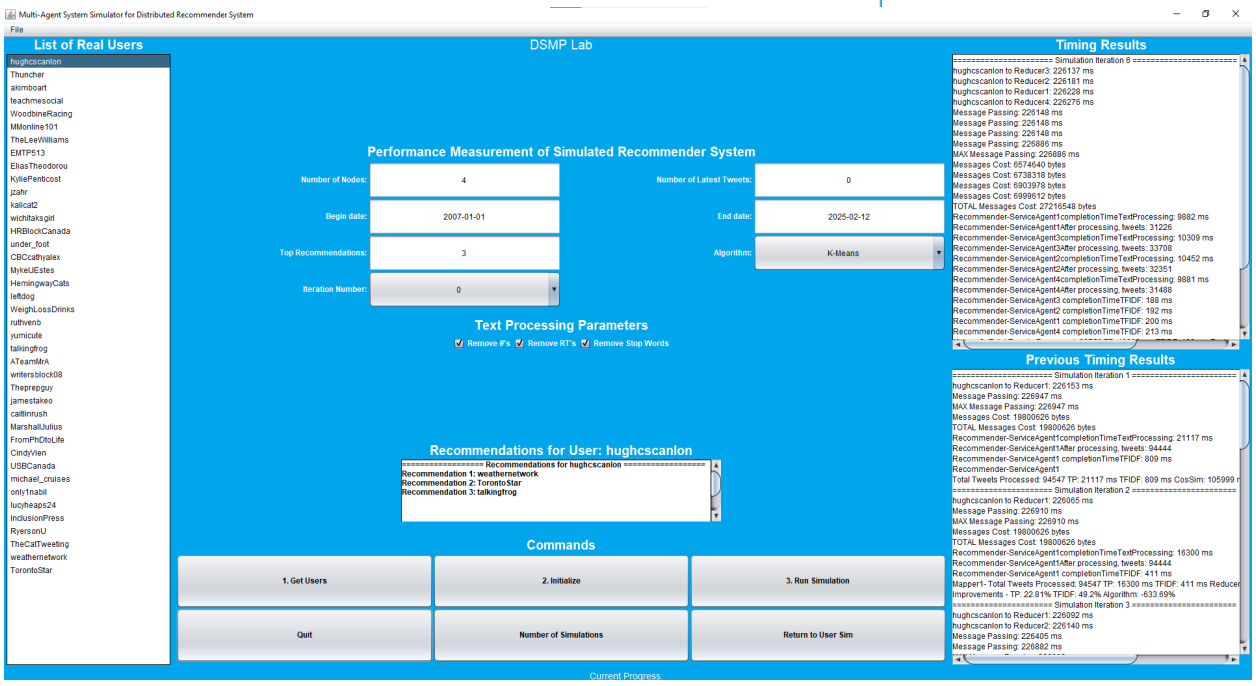


4 Nodes

Algorithm 1: Similarity



Algorithm 2: K-Means



8 Nodes

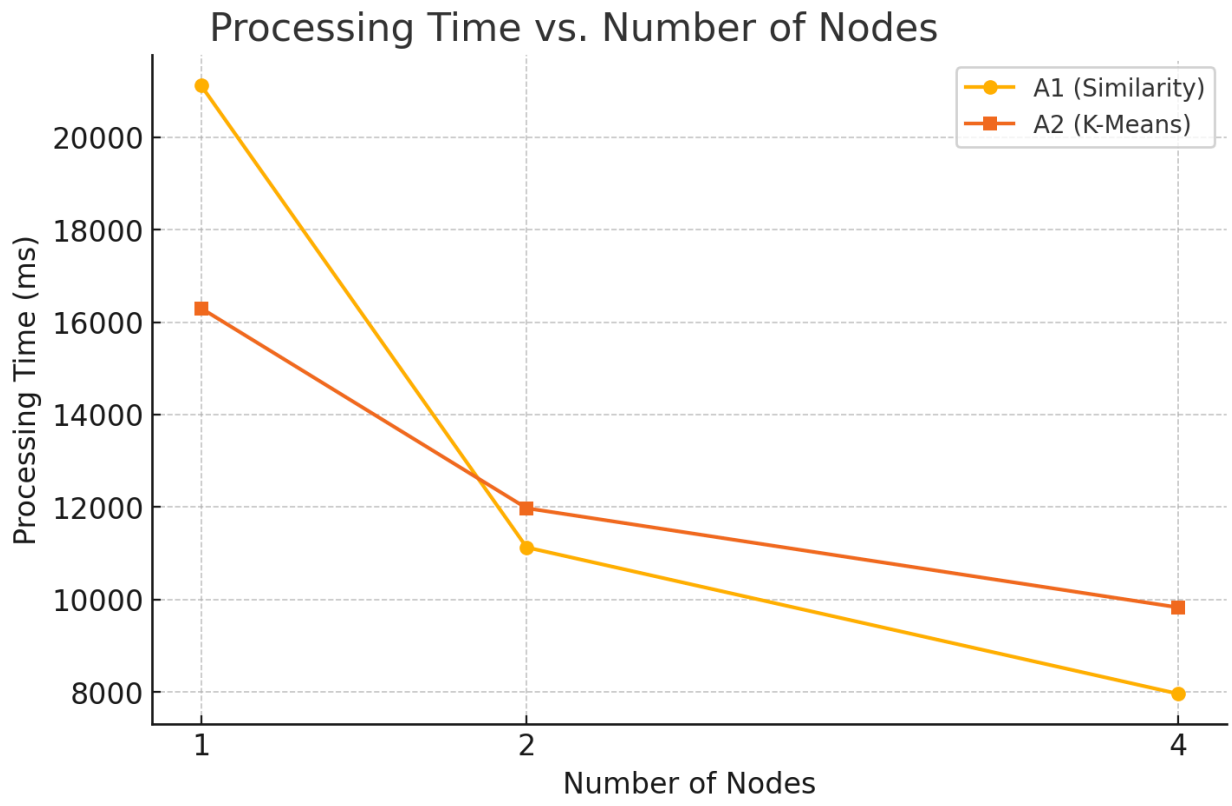
Algorithm 1: Similarity

APP DID NOT WORK

Algorithm 2: K-Means

APP DID NOT WORK

a. Plot the timing vs. nodes for each algorithm (you will plot 2 curves).



The code used to plot the table:

```
import matplotlib.pyplot as plt

nodes = [1, 2, 4]

A1_timing = [21117, 11129, 7965]  # A1 (Similarity) Processing Times
A2_timing = [16300, 9196, 9832]   # A2 (K-Means) Text Processing Times

plt.figure(figsize=(8, 5))
plt.plot(nodes, A1_timing, marker='o', linestyle='-', label="A1 (Similarity)")
plt.plot(nodes, A2_timing, marker='s', linestyle='-', label="A2 (K-Means)")

plt.xlabel("Number of Nodes")
plt.ylabel("Processing Time (ms)")
plt.title("Processing Time vs. Number of Nodes")
plt.xticks(nodes)
plt.legend()
plt.grid(True)

plt.show()
```

b. How do you determine the processing time gain for any number of nodes used in any algorithm? Write this in a formula.

$T(N_1)$ is the processing time with N_1 nodes.
$T(N_2)$ is the processing time with N_2 nodes.
G represents the percentage gain in processing efficiency

$$G = \left(\frac{T(N_1) - T(N_2)}{T(N_1)} \right) \times 100$$

Nodes	A1 (Similarity) Processing Time (ms)	A1 Speedup	A1 Gain (%)	A2 (K-Means) Processing Time (ms)	A2 Speedup	A2 Gain (%)
1	21117	1.00	0.00	16300	1.00	0.00
2	11129	1.90	47.29	11976	1.36	26.51
4	7965	2.65	62.26	9832	1.66	39.67

The total time gain is calculated as:

Total Gain = G1+G2+G3+G4

Where:

- Gn represents the parallelization gain at each node count compared to the previous count.
- Since we only have data for 1, 2, and 4 nodes, we calculate:

Total Gain (so far) = G1+G2+G3

Since we do not have the 8-node data, G4 is missing, meaning our total time gain is only approximate up to 4 nodes.

Algorithm	Total Gain (%)
A1 (Similarity)	131.15460132369276
A2 (K-Means)	141.29559537845648

c. Which algorithm yields the best timing?
A2 (K-Means) has a slightly higher overall time gain (141.30%) than A1 (Similarity) (131.15%).

A2 (K-Means) yields the best timing.