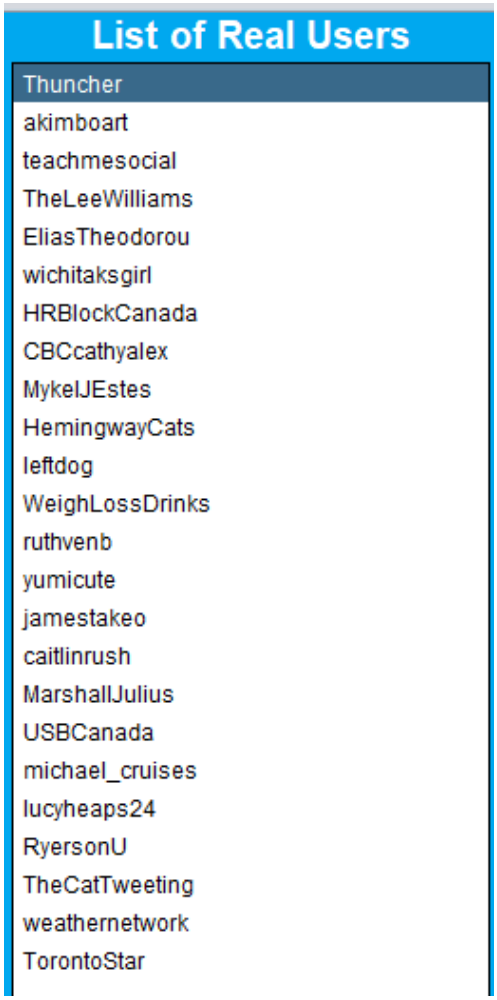
DSMP Project

Chosen Dataset: Reduced_57k.txt



For each algorithm, we ran the simulator on the dataset 3 times for each node. For the calculation, we took the average of the 3 runs to get a more accurate value.

Algorithm 1: K-Means

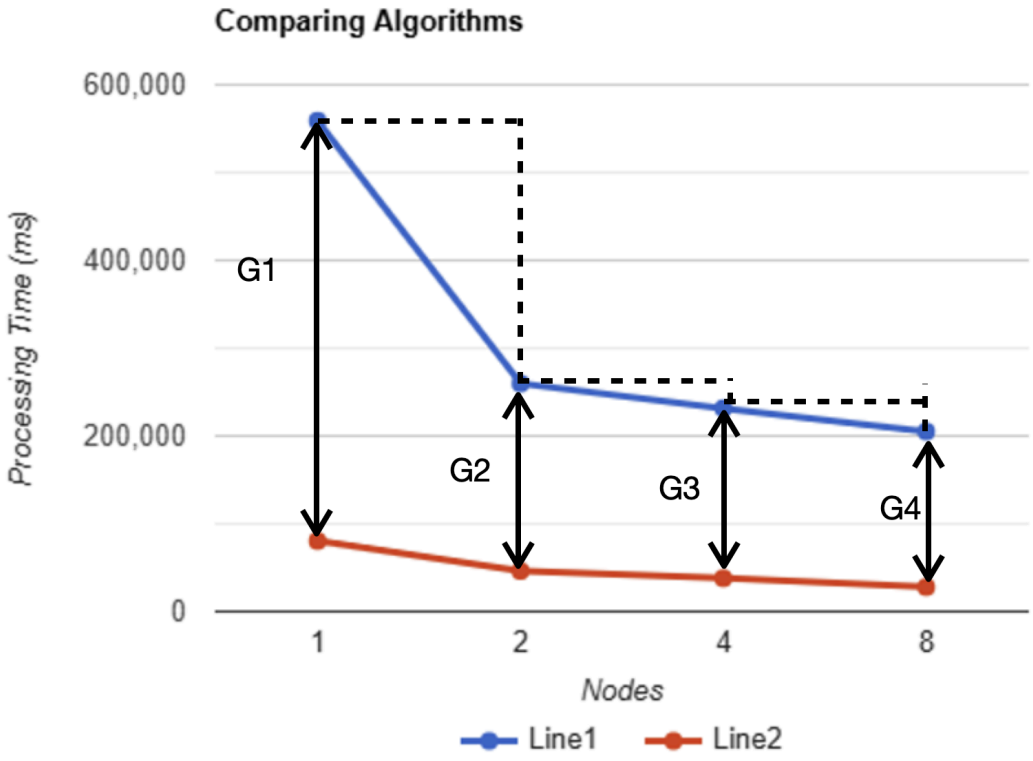
Node 1: (545283 ms + 563861 ms + 568141 ms) / 3 = **559095 ms**
Node 2: = (229088 ms + 264556 ms + 285418 ms) / 3 = **259687.3 ms ~ 259687 ms**
Node 4: (203888 ms + 231236 ms + 259480 ms) / 3 = **231534.7 ms ~ 231535 ms**
Node 8: = (212028 ms + 221504 ms + 181487 ms) / 3 = **205006.3 ms ~ 205006 ms**

Algorithm 2: Similarity

Node 1: (79382 ms + 81263 ms + 80495 ms) / 3 = **80380 ms**
Node 2: (45063 ms + 44852 ms + 48431 ms) / 3 = **46115.3 ms ~ 46115 ms**
Node 4: 36894 ms + 37789 ms +38982 ms) / 3 = **37888.3 ms ~ 37888 ms**
Node 8: (26788 ms + 29793 ms + 26985 ms) / 3 = **27855.3 ms ~ 27855 ms**

Questions

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



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

The processing time gain can be calculated as the percentage decrease in processing time when using multiple nodes compared to a single node.

The formula is:

$$\text{Gain}(\%)=((T_1-T_n)/T_1)\times 100$$

Where:

- T_1 is the processing time with 1 node.
- T_n is the processing time with n nodes.

For example:

T_1 = 559095 ms (K-Means with 1 node)

T_2 = 259687 ms (K-Means with 2 nodes)

$$G\%=((559095-259687)/559095)\times 100\%$$

$$=(299408/559095)\times 100\%$$

$$=(0.5335522)\times 100\%$$

$$=53.55\%$$

- So, the processing time gain for K-Means at 2 nodes is **53.55%**

<u>Comparison Table of Processing Time Gains (%)</u>

Nodes	**K-Means Gain (%)**	**Similarity Gain (%)**
2 Nodes	53.55%	42.63%
4 Nodes	58.59%	52.86%
8 Nodes	63.33%	65.35%

- This table shows that the **Similarity algorithm** has better scaling at 8 nodes compared to K-Means.

c. Which algorithm yields the best timing?

Based on the results,

- K-Means Algorithm: The processing time with 8 nodes is 205006 ms.
- Similarity Algorithm: The processing time with 8 nodes is 27855 ms.

Since the Similarity Algorithm has a significantly lower processing time compared to K-Means, it is the better algorithm in terms of timing.

Parallelization Gain (G1, G2, G3, G4)

K-Means Algorithm:

- G1 = 299408 ms (Reduction from 1 to 2 nodes)
- G2 = 28152 ms (Reduction from 2 to 4 nodes)
- G3 = 26529 ms (Reduction from 4 to 8 nodes)
- G4 = 205006 ms (Final execution time with 8 nodes)

Similarity Algorithm:

- G1 = 34265 ms (Reduction from 1 to 2 nodes)
- G2 = 8227 ms (Reduction from 2 to 4 nodes)
- G3 = 10033 ms (Reduction from 4 to 8 nodes)
- G4 = 27855 ms (Final execution time with 8 nodes)

Since Similarity has a lower final execution time and better scaling, it is the better algorithm in terms of performance.