Explanation for K means:

In the modified code using the multiprocessing package, we parallelized the following parts of the k-means algorithm:

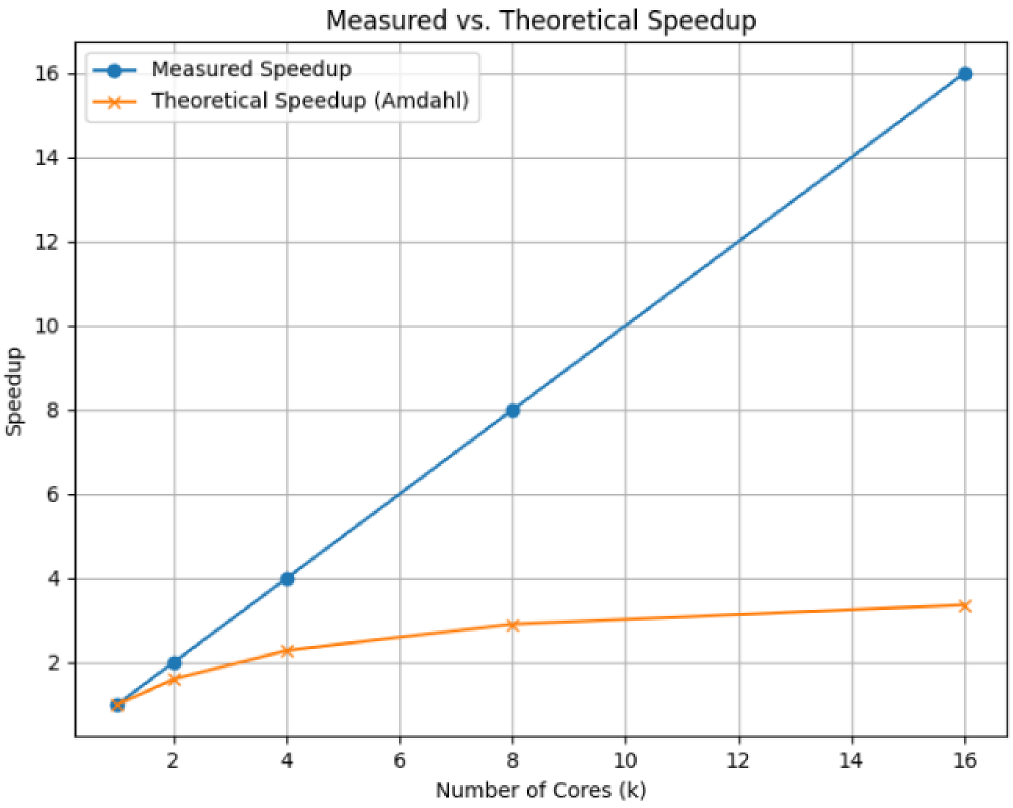
Nearest Centroid Computation:

The nearestCentroid_parallel function parallelizes the computation of the nearest centroid for each data point. It uses a pool of worker processes to distribute the data among processes and calculate the nearest centroid in parallel.

K-Means Iterations:

The kmeans function parallelizes the assignment of data points to nearest centroids during each iteration. It divides the data among processes, and each process computes the nearest centroids for its subset of data. The results are then aggregated to update the centroids.

Graph:



Part a)

To find the probability that for an item x, all bits $h_1(x)$,..., hk (x) are contained in one cache line, we'll consider the sample set S and the set of favorable outcomes E.

Let n be the number of bits in the Bloom filter, and c be the number of bits in one cache line (which is 512 bits in this case).

For an item x, the probability that all k hash functions hash to the same cache line can be calculated as follows:

The sample set S consists of all possible placements of k hash functions of a within the m cache lines:

$$|S| = m^k$$

The set of favourable outcomes E is when all k hash functions of a fall into the same cache line:

$$|E| = 1$$

Therefore, the probability that all k hash functions of item a are contained in one cache line is:

P(all hi(x) in one cache line) =

$$\frac{|E|}{|S|} = \frac{1}{m^k}$$

Part b)

Using the multiplication rule:

The probability that each hash function hi(x) falls into the same cache line for a specific x is 1/m

As there are k independent hash functions, the probability that all k hash functions fall into the same cache line is calculated by multiplying these probabilities:

P(all hi (x) in one cache line) =

$$\left(\frac{1}{m}\right)^k = \frac{1}{m^k}$$

Part c)

For this part, let's find the expected number of items for which the bits h₁ (x),..., hk (x) are contained in two or more cache lines.

Given k = 3, m = 128, and n = 100000, the expected number of items for which the bits are contained in two or more cache lines can be found using the formula for expected value:

E(items in two or more lines) = n × P(all hi(x) in two or more cache lines)

From part a, the probability that all k hash functions of an item a fall into two or more cache lines is 1 – P(all hi (x) in one cache line) =

$$1 - \frac{1}{m^k}$$

Now, plug in the values:

E(items in two or more lines) =

$$100000 \times \left(1 - \frac{1}{m^k}\right) = 100000 \times \left(1 - \frac{1}{128^3}\right)$$

Calculate the numerical value to find the expected number of items for which the bits h₁(x),..., hk (x) are contained in two or more cache lines.

WORK DISTRIBUTION:

Problem 1 (k-means): Bhavanishankar Ramesh Hakari

Problem 2: Akshay Kalson

review and documentation: Anuj Biswal