Programming of Parallel Computers

Project. KMeans clustering parallelization.

Aliaksandr Ivanou

Aliaksandr.Ivanou.1364@student.uu.se

March 28, 2014

1 Introduction

K-means clustering is one of the data mining algorithms that is aimed to partition data on several disjoint sets. Each set is represented by the centroid and the set of points that belong to this centroid.

The first step to partition data is to select centroids. Then, each iteration consists of two steps: Assignment step and Update step.

On the assignment step we are finding the nearest centroid for each point. On the update step we updating centroid coordinates(moving centroids closer to the points assigned to centroid).

The algorithm has covered when the centroids stop moving (coordinates stop updating).

Each row in the input data contains the set of features. In N dimensional space we can represent this set of features as a point. As a result, we have K points in N dimensional space and the task is find the centroid coordinates, that will represent the centers of the clusters.

2 Implementation

The input data is stored in the file mpi_input.bin. It is the binary file, the amount of records is written in the first 4 bytes, then the block are written. Each block has length 8000. Processor will read blocks starting from correct offset, as a result, each processor will collect unique set of points.

The data in text file is written in isolent.data.

There are several problems with k-means clustering algorithm implementation using MPI.

First, the tasks can distinct a lot from each other. For example, one task can have a huge amount of records, but each record will contain small amount of features. Another task can have small amount of records but large amount of features.

Second problem is that we cannot read all data in one processor and scatter it to another, we need to read data from the file, but the amount of records can be random, as a result, each processor can have different amount of records to process.

The cartesian grid topology is used in the implementation. Each row represents the group of processors with the same centroids. Each group computes data only for their centroids, then the data is shifted to the next row.

The algorithm consists of next steps:

```
Each processor reads its data chunk from the file
One processor scatters centroids to all processors
While centroids are changing their coordinates:

        for i in number of groups:
                assign nearest centroid to points
                shift points

        for i in numbe of groups:
                update centroid coordinates
                shift points

        for i in group_size:
                shift centroids
                update centroid coordinates

        check if coordinates have been changed

        gather centroids
```

3 Results

The task of clusterisation is quite hard to test. If the amount of features high, we cannot show them on the plot. The implementation currently prints the centroids to the standard output.

For testing, next dataset was used: http://archive.ics.uci.edu/ml/datasets/ISOLET.