**Parallel implementation of K-Means**

Final project of the course Parallel and Distributed Computation, 2016 Summer Semester

# Problem Definition

A big number of points move on XOY plane. For each given moment **t** we are interested to define **K** clusters and find the minimum distances **d** between centers of the clusters.

Generally speaking, after finding K clusters (and there for K centers) in each given t, we need to give as a result the minimum distance between K centers considering all of the clusters found in each t, “step” i.e. :

Find the min dis between any given K clusters(local min).

Find the min from all calculated local mins(global min).

Each point Pi moves around its center (ai, bi) as follows:

ai, bi represent the circle center, each point has its own center which it moves on its perimeter.

xi = ai + Ri \* cos(2t/T)

yi = bi + Ri \* sin(2t/T)

Where:

Ri is a radius of the circle of the point Pi,

T defines the time interval [0, T] to check the system.

Input data and Output Result of the project

You will be supplied with the following data:

* Number of points of the set and number of clusters to be found for each ~~point~~  “step” t
* t – the time step between calculation of centers of the clusters
* T – time interval
* LIMIT – the maximum number of iterations for K-MEAN algorithm.

**Input File format**

The first line of the file contains the number of **given points** and **K** - number of clusters to find, time increment **t**, time interval **T**, and maximum number of iterations **LIMIT**.

Following lines contain **point ID**, coordinates of the circle (ai, bi) and its radius Ri .

For example:

**250000 3 0.1 150.0 2000**

**0 12.5 16.2** **10.23**

**1 34.4 100.2 14.2**

**2 96.7 3.777 5.5**

**…**

**249999 55.2 11.01 60.14**

**Output File format**

The output file contains

* The minimum value of **d** during the time interval [0, T], time **t** when this occurred and coordinates of all centers of the clusters at this moment.

For example:

**d = 14.5**

**t = 120.3**

Centers of the clusters:

**12.4 55.6**

**2.4 90.15**

**66.0 45.2**

# Requirements

* Implement the K-MEANS algorithm explained in the class (see the appendix).

Use first K points as initial positions of the centers of the clusters.

In case that in some iteration there will be no points in cluster – keep its center for the next iteration.

* The input file initially is known for one machine only.

The output file has to be written to the file on the same machine.

* The computation time of the parallel program have to be faster than the one for sequential solution. For example, the run time of the sequential program that implements very naïve approach in case of **250000** points and **3** clusters will take approximately on i7-2600 CPU with 8GB RAM (checked in lab #251, Ficus).
* Be ready to demonstrate your solution running on at least **three computers**.
* No code sharing between students is allowed. The parts of code, if any, which was incorporated to your project, have to be referenced according to the academic rules.
* Be able to explain each line of the project code.
* The set contains at least 1000 but not more than **300,000** points. The number of clusters K will be less than **100**.

# Grade Policy:

* **60 points** for the effective proper parallel implementation of the problem with two components: ***MPI+OpenMP*** or ***OpenMP+ CUDA*** or ***MPI+CUDA***. The project that produce wrong results will not be accepted
* **10 points** for implementation in ***MPI+OpenMP+CUDA*** configuration.
* **10 points** for the documentation of your solution – clear explanation what and how the problem was parallelized, what is a rational of choosing the specific architecture, complexity evaluation.
* **10 points** for the code quality – modularity, generality, self-explanatory, organization.
* **10 points** for the Load Balancing.

***Additional Bonus for the project grade***

**5 points** for implementation under LINUX OS

**5 points** for implementation with OpenCL

**5 points** for implementation of sophisticated variation of the K-MEANS algorithm(have to be approved by lecturer).

**5 points** for your own proposal (have to be approved by lecturer).

# הפרויקט יוגדר כמטלת הקורס. הגשת התוכנה והתיעוד רק דרך מערכת Moodle.

# יישום והגשת הפרויקט ביחידים בלבד.

בהצלחה