Project proposal:
Porting the K-means algorithm on GPUs

The K-means problem is one of the oldest problem in computational geometry, it can be formulated as follows: given an integer K and a set of n data points in R^{n_c} ($n_c \ll n$), the objective is to choose K centers which minimize the total squared distance between each point and its closest center. This problem is also known as data partition or data clustering in a euclidean space [1].

The K-means algorithm is a local search optimization method. It seeks to find a partition $V_1, \ldots, V_K$ of K circular sets with centers $\mathbf{c}_1, \ldots, \mathbf{c}_K$ which minimizes the sum of the squared euclidean distance between $\mathbf{x}_i \in R^{n_c}$ and the center of the set to which it is assigned:

$$\sum_{k=1}^{K} \sum_{\mathbf{x}_i \in V_k} \|\mathbf{x}_i - \mathbf{c}_k\|^2$$

The K-means algorithm, described in Algorithm 1, starts with K arbitrary centers, typically chosen uniformly at random from the data points. Each point is then assigned to the nearest center, and each center is recomputed as the center of mass of all points assigned to it: $(\mathbf{c}_k)_j = 1/|V_k| \sum_{\mathbf{x}_i \in V_k} (\mathbf{x}_i)_j \ \forall j = 1, \ldots, n_c$.

Algorithm 1 *K-means*

Data: A set of points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with $\mathbf{x}_i \in \mathcal{R}^{n_c}$, an integer $K < n$, an integer *maxiter*
Result: A partition $\{V_k\}$ of X into K nonempty clusters
Initialize K centers $\mathbf{c}_k$ by sampling K random points from X
while $\mathbf{c}_k$ is changing and *iter* $<$ *maxiter* **do**
  $V_k = \emptyset$
  for $\mathbf{x}_i \in X$ **do**
    find $k = \arg\min_{1 \le k \le K} \|\mathbf{x}_i - \mathbf{c}_k\|^2$
    add $\mathbf{x}_i$ to V_k
  end for
  compute new centers: $(\mathbf{c}_k)_j = 1/|V_k| \sum_{\mathbf{x}_i \in V_k} (\mathbf{x}_i)_j \ \forall j = 1, \ldots, n_c$
end while

The objective function is monotonically decreasing, then no configuration is repeated during the algorithm; furthermore, since there are only K^n possible cluster configurations, the process will always terminate. This version of the algorithm provides a simple and fast method for spatial clustering, although it offers no approximation guarantees; indeed, the final result largely depends on the initial centers and it can lead to a poor approximation of the global minimum of the objective function, however the method is still among the widely used ones in practice.

A version of this algorithm is implemented in C in the software framework available at https://github.com/bootcmatch/BCMatch4Graphs An implemen-

tation of K-means is also available in the Statistics and Machine Learning Matlab Toolbox.

Main objective of this project is the design, development and testing of a version of K-means for Nvidia GPU architectures by using CUDA. For testing and results analysis, the following guidelines can be considered:

- 2D or 3D randomly generated data sets can be used to graphically illustrate results;
- performance indicators, such as execution times and speedup when using GPU versus CPU, can be used to analyse efficiency of the GPU implementation.

References

[1] Hastie T., Tibshirani R., Friedman J. *The element of statistical learning*, New York, USA: Springer; 2001.