KADIR HAS UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

HYBRID KMEANS CLUSTERING ALGORITHM

MUSTAFA ALP ÇOLAKOĞLU

May, 2013

HYBRID KMEANS CLUSTERING ALGORITHM

MUSTAFA ALP ÇOLAKOĞLU
B.S., Computer Engineering, Kadir Has University, 2011
M.S., Computer Engineering, Kadir Has University, 2013

Submitted to the Graduate School of Science and Engineering

In partial fulfillment of the requirements for the degree of

Master of Science

In

Computer Engineering

KADIR HAS UNIVERSITY
May, 2013

KADIR HAS UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

HYBRID KMEANS CLUSTERING ALGORITHM

MUSTAFA ALP ÇOLAKOĞLU

APPROVED BY:

Asst. Prof. Dr. Zeki BOZKUŞ Kadir Has University _____
(Thesis Supervisor)

Asst. Prof. Dr. Taner ARSAN Kadir Has University _____

Prof. Dr. Selim AKYOKUŞ Doğuş University _____

APPROVED DATE: 14/05/2013

“I, Mustafa Alp Çolakoğlu, confirm that the work presented in this HYBRID KMEANS CLUSTERING ALGORITHM is my own. Where information has been derived from other sources, I confirm that this has been indicated in the HYBRID KMEANS CLUSTERING ALGORITHM.”

_____

MUSTAFA ALP ÇOLAKOĞLU

HYBRID KMEANS CLUSTERING ALGORITHM

Abstract

From the past up to the present size of the data is rapidly increasing day by day. Growing dimensions of this data can be held in databases is seen as a disadvantage. Companies have seen this information in databases as an excellent resource for increasing profitability. According to this source, the profiles of the customers can be clustering and new products can be presented for cluster customers. So data mining algorithms are needed for rapidly examine these sources of information and obtaining meaningful information from resources.This project has been implemented K-means clustering algorithm with the hybrid programming method. This project suggested that data grouped with hybrid programming takes less time. Algorithm accelerated with hybrid programming method. Parallel programming used to solve K-means problem with using multi-processor and threads used for running operations at the same time. Hybrid version of K-means clustering algorithm was written using the C programming language. Existing parallel K-means source code used thread structure is added. Message Passing Interface library and POSIX threads are used. Hybrid version of K-means algorithm and parallel K-means algorithm are run many times under the same conditions and comparisons were made. These comparisons were transferred to the tables and graphs.

Keywords: K-means Clustering Algorithm, Hybrid Programming, MPI, POSIX threads.

HİBRİT KMEANS KÜMELEME ALGORİTMASI

Özet

Geçmişten günümüze kadar olan süreçte veri boyutları günden güne hızla artmaktadır. Veritabanlarında tutulabilen bu verilerin işlenebilirliği artan boyutlardan dolayı dezavantaj olarak görülmektedir. Şirketlerin veritabanlarında bulunan bu bilgiler iyi kullanıldığı halde karlılığı arttırmaya yönelik bir mükemmel bir kaynaktır. Bu kaynakla müşterilerin profillerine göre bir kümeleme yapılabilir, yapılan kümelemelerle ilgili kümedeki müşteriye hitap edecek ürünler sunulabilmektedir. Bu kaynakların hızla incelenebilmesi ve kaynaklardan anlamlı bir bilgi çıkarabilmek için veri madenciliği algoritmalarına ihtiyaç duyulmaktadır. Bu projede K-means kümeleme algoritması hibrit programlama yöntemiyle implemente edilmiştir. Hibrit programlamayla kümeleri oluşturacak verilerin daha kısa sürede gruplanabileceği öne sürülmüştür. Algoritma hibrit programlama yöntemiyle hızlandırılmıştır. Hibriti oluşturan paralel programlamayla program parçacıklarının çoklu işlemciye sahip sistemlere dağıtılması ve işletilmesi, iş parçacıklarının yardımıyla birden fazla sürecin aynı anda yürütülmesi sağlanmıştır. Hibrit algoritma, C dili ile implemente edilmiştir. Var olan paralel K-means kaynak kodları, iş parcacıkları ile hibritleştirilmiştir. Paralelleştirme işlemi için Message Passing Interface kütüphanesi ve POSIX threads kullanılmıştır. Hibritleştirilen K-means algoritması, var olan algoritmayla aynı şartlar altında birden fazla kez çalıştırılarak sonuçlar elde edilmiş ve karşılaştırmalar yapılmıştır. Bu karşılaştırmalar tablolar ve grafiklere aktarılmıştır.

Anahtar Kelimeler: K-means Kümeleme Algoritması, Hibrit Programlama, Paralel Programlama, İş parçacığı

Acknowledgements

Firstly of all, I'd like to thank my supervisor Asst. Prof. Dr. Zeki BOZKUŞ who always supported and helped me during the thesis period. Thank you for your suggestions and valuable comments.

I would like to thank Research Professor Wei-keng Liao for Parallel K-means Data Clustering implementation.

I also wish to thank Asst. Prof. Dr. Taner ARSAN for support and advices helped me to reach this stage.

Finally, I would like to thank my family for without whom I could not go further. Words are not enough to express my thankfulness to you, as well as to my friends.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

MPI	Message Passing Interface
KDD	Knowledge Discovery in Databases
CRM	Customer Relationship Management
GPU	Graphics Processing Unit
POSIX	Portable Operating System Interface

Chapter 1

Introduction

K-means Clustering Algorithm is most used method for clustering objects and this project aim is improving execution time of K-means Clustering Algorithm. Parallel version of algorithm was written from Research Professor Wei-keng Liao Department of Electrical Engineering and Computer Science at Northwestern University [12]. Hybrid K-means algorithm implemented with existing codebase. Parallel and hybrid version of algorithm have been tested on the same environment and data. Data mining has been examined because clustering algorithm is a sub topic of data mining. K-means details, serial and hybrid version of K-means has been presented with tables and figures. Finally, hybrid version of K-means algorithm has presented with sample data and performance results are shown with tables and figures.

1.1. Objectives

Data size is rapidly increasing day by day. Companies have huge databases with growing dimensions for keeping information. This information has hidden data which is valuable for companies about customers. So data mining is very important for companies to obtain meaningful information from resources.

By the growing data classic clustering algorithms lack in performance with huge datasets and companies loose time to extract meaningful information from databases. So fast clustering algorithms are needed to improve performance and extract valuable information from data.

Aim of this study is to get same results with lower execution times by hybridization. Also, hybrid version of K-means algorithm not requires special hardware for execution.

1.2. Used Environments and Materials

Hybridization of K-means algorithm is written using parallel K-means source code so C Programming Language has been used for Hybrid K-means Clustering Algorithm. Both of algorithms have been used MPI libraries. This library helps programmer to coordinate computers which is in network for performing jobs in parallel. In addition, Pthreads are used in hybrid version of K-means, while Pthreads do not communicate but they share data and process it at the same time.

Large dataset has used for parallel and hybrid version of K-means algorithm. This dataset US Census 1990 dataset and has 2,458,285 records with 68 dimensions. Dataset collected from UCI KDD archive of California, Irvine. (http://kdd.ics.uci.edu/) This online repository includes large data sets with different data types and application areas. Main role of the repository is providing complex large dataset for researchers for data mining and reach to knowledge [1]. Versions of K-means algorithm has been performed Linux operating system of Kadir Has University.

1.3. Relational Works

Pthread Parallel K-means [2] is a paper, this work use Pthreads with parallel k-means algorithm. In this project a uniprocessor k-means algorithm taken and k-means parallel version implemented with Pthreads. Algorithm used from Normalized Cuts codebases. In this project 20 K clusters are used, data is texture with 39 dimensions. Pthread Parallel K-means codes run on a 4 katmai Linux kernel with Pentium 3 machine. 120x80, 240x160 and 480x320 pixel images are used as a data set. Uniprocessor and Pthread parallel code performances discussed. Speed up is shown with darker of the parallel code and the lighter shows uniprocessor performance. Parallel code running with 4 Pthreads and results shown below:

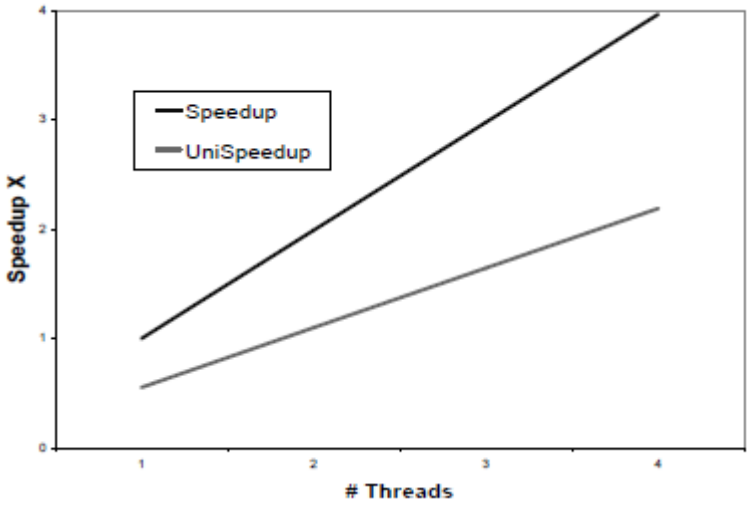

Figure 1: Speedup [2]

Performance analysis of PC-CLUMP based on SMP-Bus utilization [4] is another work for used hybrid programming model. PC-CLUMP is a HPC application for commodity based platforms. In this project, PC-CLUMP performance is analyzed with the SMP-bus access ratio. Shared memory and

Distributed memory programming is used for this project. These programming models are MPI-ONLY, Pthread + MPI, OpenMP + MPI. Threads interact with each other on shared memory with Pthread and MPI used for node communication. In this project, hybrid programming model Pthread and MPI gives a higher performance than MPI-only. Speedup figure is given below for Pthread + MPI performance 1000x1000 matrix size and number of P is 4. P indicates number of CPUs per node.

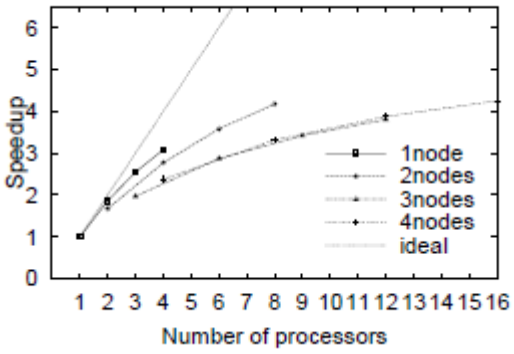

Figure 2: Speedup2 [4]

Another work is A Parallel Implementation of K-Means Clustering on GPUs [8]. In this project K-means Clustering Algorithm parallel version implemented using with

GPU. One dimensional array is used as a dataset and dataset contains 1 million elements which is unsigned long integer. 4000 clusters are chosen for implement algorithm. Performance results are shown below:

Platform	Time (s)	Performance Increase
Intel Pentium D, 3 Ghz	9.830	1 X
NVIDIA 8600 GT	0.724	13.57 X
NVIDIA 8800 Ultra GTX	0.144	68 X

Figure 3: Performance Increase [8]

1.4 Thesis Structure

Chapter 2 gives information about Data mining and Cluster Analysis. KDD and features of Cluster analysis examined. Then, K-means Clustering algorithm examined and steps of K-means Algorithms discussed in Chapter 3. After that, Chapter 4 explains Parallel programming models, reasons of parallel programming implementation.

Hybrid algorithm discussed, project environment is given and steps of Hybrid K-means Algorithm explained in Chapter 5. Last chapter contains dataset information for this project, algorithm execution strategy, parallel and hybrid algorithm performance results with tables and graphs.

Chapter 2

Data Mining and Cluster Analysis

In this section, data mining and KDD steps examined. Then, cluster analysis discussed and features of cluster analysis examined.

2.1. Data Mining

Data mining is an algorithm for extracting patterns from data [3]. Companies can extract valuable information from databases with data mining. Customer behaviors can be followed with data mining and companies can present special products for customers so they can increase their profits. Data mining is a step of Knowledge Discovery in different areas. We can show some widely used areas.

- Marketing: Following customers buying patterns, determination of campaign products, earning new customers with keeping current customers, CRM is most common data mining application areas for marketing.
- Banking and Insurance: Score analysis of customers for Credit card and loan applications, following customer profiles.
- Text Mining: Analysis of very large data set and obtains a relationship between data.
- Web Mining: Decreasing duration of data mining on the web with text, picture etc.

KDD is a method for discover knowledge from databases. KDD has five steps and data mining is fourth step. These steps are selection of data, preprocessing of data, data transformation, data mining and interpretation/evolution of data.

- Preprocessing of Data: Data are cleaned in this stage, noise and irregularities are cleaned from data. At this stage, normalizations are performed to collect various data in a single warehouse.
- Selection of Data: At this stage, some preprocessing operations are performed for selection of data. Determination of the data type, generalization of data, data presentation and visualization of data. Data Mining Query Language (DMQL) is used for these operations.
- Data Transformation: Preprocessed data is transmitted.
- Data Mining: At this stage, useful data extract with the aid of previous stages.
- Evolution / Interpretation: In this stage, evaluate and discuss data mining result

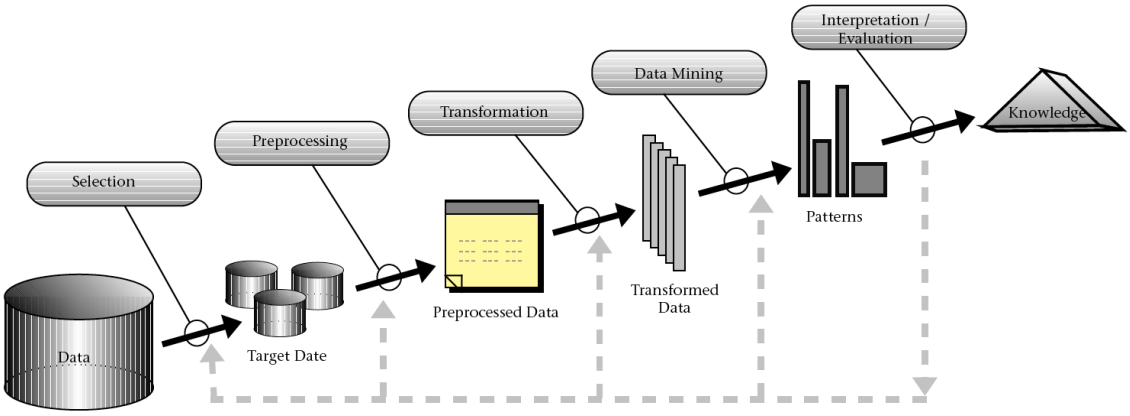

Figure 4: KDD [3]

2.2. Cluster Analysis

Clustering is a method for grouping similar objects. Different datasets can be used for a clustering process and similar data are involved into same cluster. Cluster analysis, used for statistics, biology, spatial data mining and machine learning, image recognition.

2.2.1. Features of Cluster Analysis

- It must be implemented for large data sets, Available for not properly shaped data.

- Supports for different data types , Both numerical and categorical data can be used.

- Number of the input variable must be minimum. If a method includes a less input variable, it is more independent from user.

- Cluster analysis must be available for data which contains noises.

- It must be independent from order of data; it must be work for different combinations of data.

- Must be available for multi-dimensional databases.

- Results must be clean and obvious.

These features are ideal features of a clustering algorithm. Cluster algorithm is a grooving topic and popular method.

Chapter 3

Clustering Algorithms

Similar objects are grouped with clustering algorithm methods. Different datasets can be used for a clustering process and similar data are involved into same cluster. There are many type of clustering algorithm but this project mention about K-means Clustering Algorithm. K-means Clustering Algorithm is most used and known clustering algorithm. In this work, K-means Clustering algorithm implemented with a hybrid programming which is id a mixture of MPI and Pthread. Firstly, classic K-means Algorithm will be explained and its working principle. Hybrid and parallel version of K-means Clustering algorithms will be explained next chapters.

3.1. K-means Algorithm

K-means Algorithm one of the best known and widely used algorithm for clustering. K-means algorithm take an initial parameter and this parameter is used to separate the pieces into classes. Points are placed around the closest data or placed which is similar to the centroid. K-means algorithm which is used in this study and developed by J. MacQueen, in 1967. This algorithm use Euclidean distance for calculations. This method is most common and used way for calculations . K-means become a clustering algorithm for scientific and industrial applications for many years. The name of K-means comes from algorithm initial parameter. This parameter indicated by number of cluster k. So k is an already known a constant positive integer number before the clustering process and unchanging value until the end of the clustering process.

This algorithm is commonly used for clustering but some weak aspects are available:

- There is a need initial k number as a parameter for algorithm. Results of the algorithm depend on this k parameter. If the number of cluster does not certain, this number must be found with trials.
- Excessive noise and exception data change the average algorithm so K-means algorithm is very sensitive to noise and exceptions. Algorithm data can be removed from exception and noise before starting process. K-means Clustering Algorithm is not suitable for overlapping clusters. Each element is a member of only one cluster at the same time.

- K-means Clustering Algorithm suitable for numerical attributes [3].

In this part, algorithm will be described with two dimensional diagrams. Figure 5 shows K-means working principle.

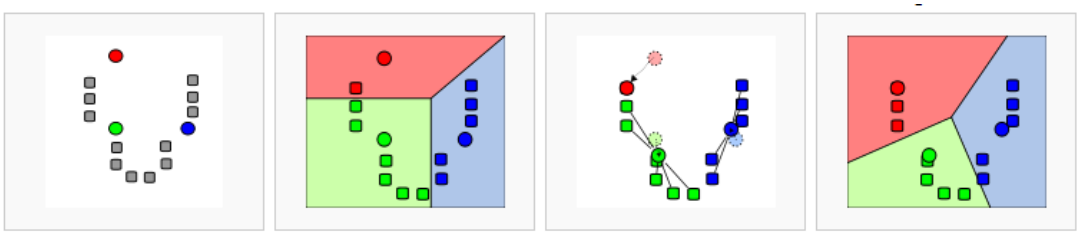

Figure 5: K-means Algorithm Working Principle [9]

In this example, k initial point is 3 and red, green, blue circles represent the cluster centers. The remaining gray points represent initial points. Firstly, gray points assigned red, green and blue circles and first clusters are generated. Average is calculated with elements in each cluster and the cluster centers calculated. Changing cluster centers is shown and the new cluster centers are indicated by arrows. The

same procedure has been repeated before center points become a constant point for each cluster. In this way, randomly selected cluster centers are recalculated and they come close to the real cluster centers. This process has stopped when centers are not changed.

3.2. K-means Algorithm Steps

The first step of the algorithm cluster centers or in other words K points are selected. K must be greater than 1. In MacQueen cluster algorithm k cluster centers are selected from first k elements. But, if k element values are close to each other, the selection of first k points made by randomly. Each of the points referred to as the prototype. These selected elements are determined by initial clusters and they are cluster centers. Weighted average value of the cluster or closest value of this average called by cluster center. Determining the boundaries between the members with closer to the center of the cluster. Firstly, center of two clusters are connected with a straight. Center point of this straight is a boundary of the two clusters. Boundary line is a reference point and show that elements which will be included in the cluster. The most widely used method of calculating the distance between points is Euclidean equation. Two dimensional information need straight to joining of the two cluster center while plane is used when dimensions increase.

The third step is a new element added to a cluster and each cluster weighted average is calculated and defined a new cluster center. Weighted average is calculated with the each of the average values of all the elements. While at the beginning algorithm initial k points are a cluster center, as a result of the second loop of the new cluster centers are no longer a cluster member, just an average value. The next selection procedures the cluster center represented by an average value. Different elements can be included for each cycle.

Clustering process continues with the elements of a cluster until to participate same or different cluster. Iterations continue until no object moves to another cluster.

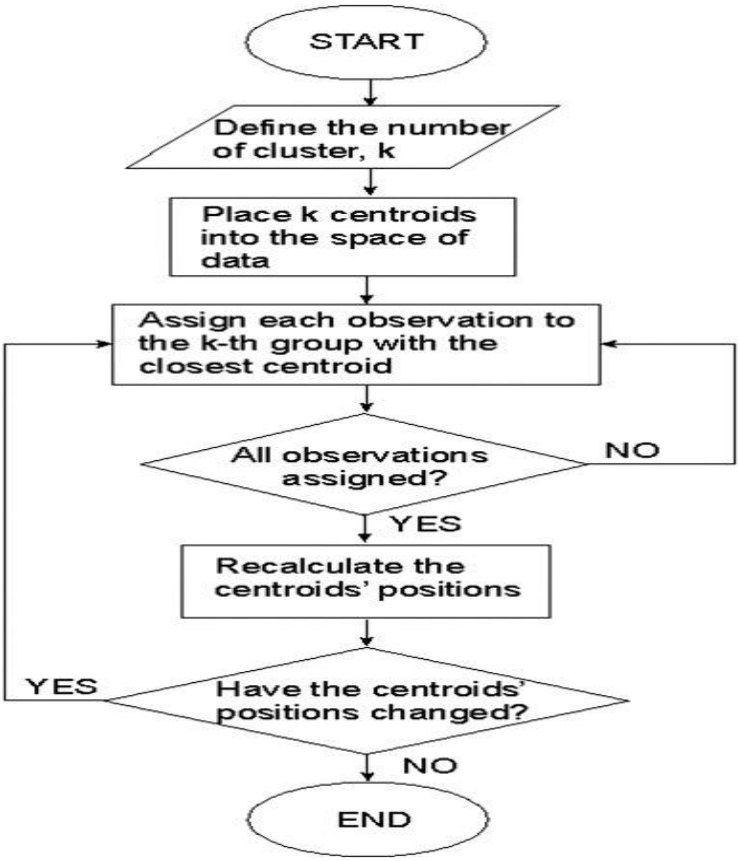

Figure 6: K-means Algorithm Steps [10]

3.3. Effects of the K Number

In clustering algorithms, K is indicated by the number and it show number of clusters. K is known before processing and clustering and not changes until the process finish.

K-means clustering algorithms and other clustering algorithms cannot offer a solution to determine the number k. But in many cases, determine a specific K value is not necessary. In analysis phase, preliminary study are made to determine the value of k. Clustering algorithm is executed with an estimated k value and results are discussed if there is no clustering occur, k value is changed and algorithm run again.

K-means algorithm uses geometric or arithmetic calculation methods to separate the data into clusters. These methods are:

3.3.1 Geometric Calculation :

In this method, cluster data shown as a point in coordinate system. Euclidean distance, most widely used method for calculating distance between points. In two-dimensional systems uses the right for defining cluster edges. When the number of dimensions increases, plane is used. An example explained for geometric calculation with 20 elements and 3 clusters [11].

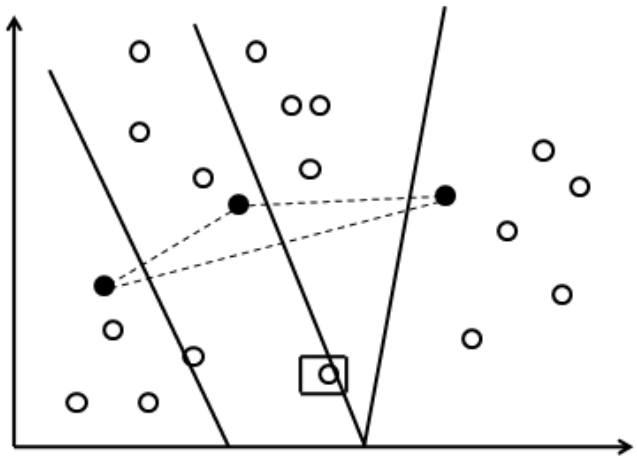

Figure 7: Cluster Center Points [11]

In first step, 3 elements are selected for initial cluster centers. Filled circles show cluster centers in Figure 7. In second step, edges of cluster determined to join objects nearest cluster. In the third step, taking the average value of each set of elements and cluster centers recalculated.

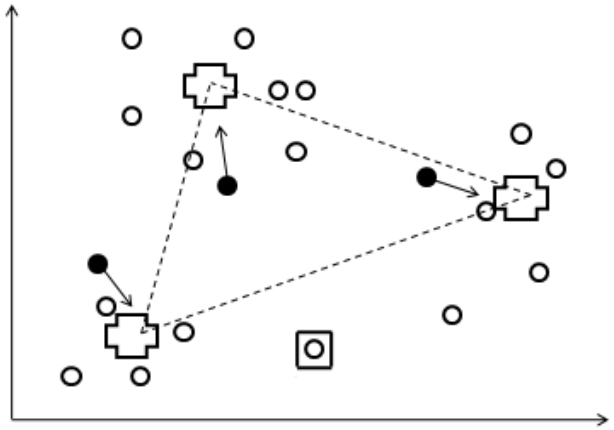

Figure 8: New Clusters [11]

In Figure 8, plus symbols show the new cluster symbols. Circle is drawn previous cluster centers to show changes of new cluster centers. Square marked element in Figure 7 included second cluster after first loop. Then, in second loop element become a member of first cluster [11]. Cluster boundaries changes with cluster centers modification in each cycle. Figure 8 shows the boundaries of the new clusters. These processes continue when cluster centers become a constant point.

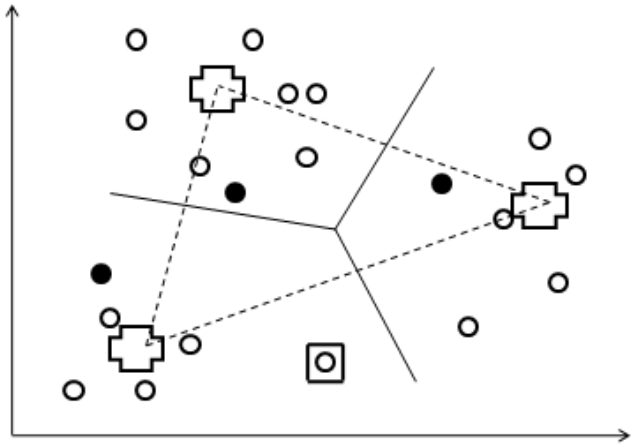

Figure 9: Cluster Boundaries [11]

3.3.2 Arithmetic Calculation

Data used to explain arithmetic calculation method. The data in the Figure 10 are four kinds of points. Each point has x and y value. In this example K is taken 2[7].

Object	X	Y
Point A	1	1
Point B	2	1
Point C	4	3
Point D	5	4

Figure 10: Cluster Points [7]

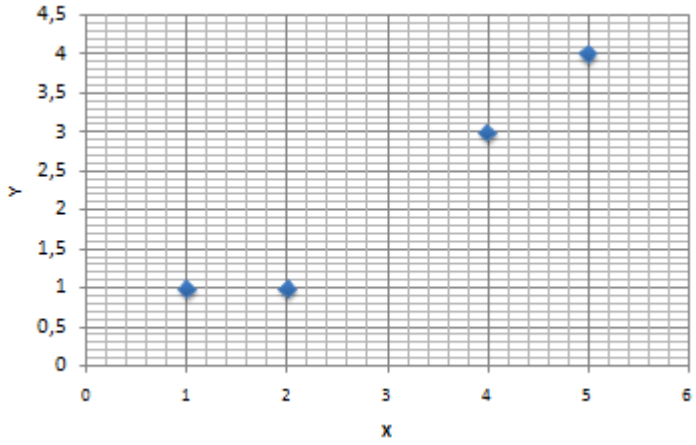

Figure 11: Cluster in Two Dimensional Coordinate System [7]

Figure 11 shows points in two dimensional coordinate system. In the first step, the first cluster centers taken A and B points because they are nearest point to (0.0) .

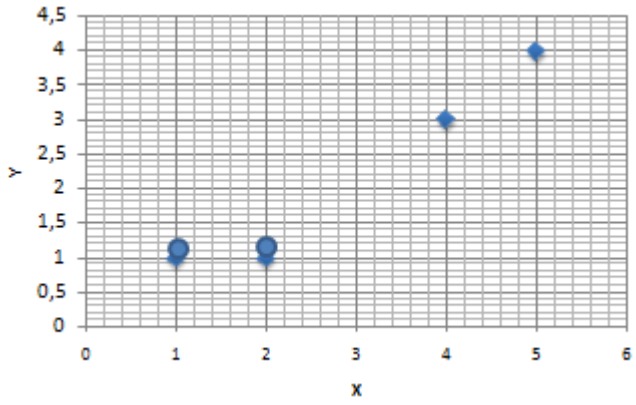

Figure 12: Cluster Centers [7]

Figure 12 shows cluster centers (1, 1) and (2, 1). In the second step, the distance between selected first cluster centers and other elements are calculated. Each column

of matrix shows object coordinates for Euclidean distance calculation. The distance from the center of the first cluster of objects in the first row of the matrix the second line, the distance of the objects located in the center of the second set. For example, on the first line distance between A and B, C, D values will be given. In the same way on the second line between B and A, C, D values will be given. Calculated values of first matrix will be given below:

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) \\ \mathbf{c}_2 = (2,1) \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

Figure 13: Cluster Distances [7]

Cluster centers distances to them (1, 1) and (2, 1) index values are zero. Euclidean calculation using for calculation. For example, C distance is from A and B calculated as 3.61 and 2.83 [7].

In the third step, the distance to a cluster objects are member of a cluster based on the value in the matrix. The smallest values are found for every object. For example, the first and second lines of the object values of D 5 and 4.24. Smallest value 4.24 D object assigns second cluster. At the same way, the first and second lines of the object values of C 3.61 and 2.83. Smallest value 2.83 C objects assigns second cluster. A and B not assigned because they are cluster centers. The matrix G is a cluster; the first line of matrix shows first cluster, the second row shows the second cluster. If objects included in cluster takes value of 1, if not takes value of 0. Object

A is cluster center and single element of cluster. Object B is cluster center of second cluster and C and D objects are elements of second cluster.

$$\mathbf{G}^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \\ A \quad B \quad C \quad D$$

Figure 14: Matrix G [7]

First cluster has only one element and cluster center does not change its (1, 1). Second cluster has 3 elements and new cluster center calculation with x and y values objects by arithmetic average and average is (2.66,3.66).

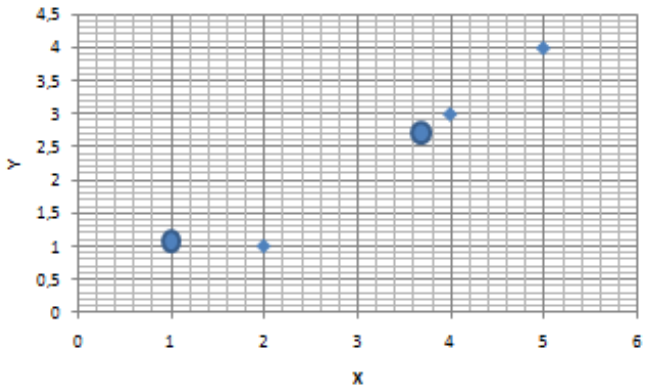

Figure 15: New Cluster Centers [7]

New cluster centers are shown in Figure 15. In figure, second cluster center is a point not an object. After forming the new cluster centers, object distances are recalculated with Euclidean distance.

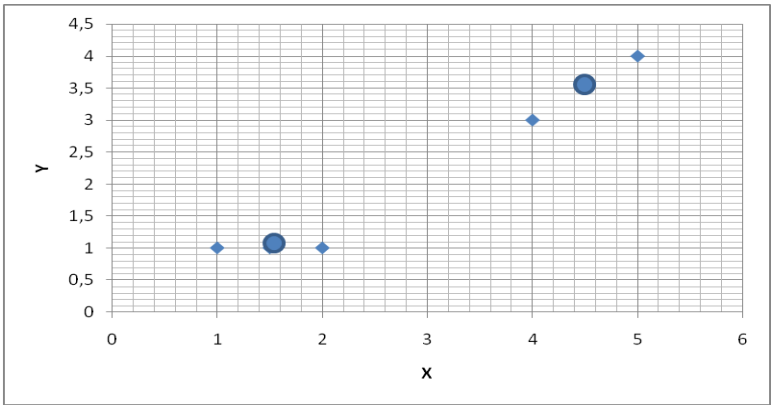

Figure 16: Final Clusters [7]

The smallest objects are included in a cluster with distance matrix. This time the elements of the first cluster become the objects A and B, the second cluster objects are C and D. Cluster centers are recalculated and Figure 16 shows new cluster centers. These calculations continue until cluster edges do not change.

Chapter 4

Parallelization of Algorithms

In this chapter, parallel programming, reasons of parallel programming, parallel programming models are examined.

4.1 Parallel Programming

Parallel programming is a methodology to solving a computational problem with using of multiple compute resources. Parallel programming is efficient and useful model to remedy difficulties in many areas. Some of these areas:

- Science (Genetics, Geology, Biotechnology etc.),
- Engineering (Mathematics, Circuit Design, Microelectronics etc.),
- Industrial (Advanced graphics and virtual reality etc.),
- Commercial (Web search engines, oil explorations etc.).

4.2 Reasons of Parallel Programming

We introduce parallel programming, but now why parallel programming is so popular explained and it has widely application area. Some of main reasons:

- Time & Money Saving: Only one job can process with serial programming while parallel programming assign more resources for a job and it decrease its completion time and it helps reduce costs.
- Big Problems Solving: In many areas problem are complex and big, so solving these problems with a single computer is very difficult. For example: Web Search Engines operate million of queries at a same time.
- Concurrency: Single computing makes only one thing while multiple computing doing many things at a same time.
- Transmission speeds: Serial Computers data transmission speeds depends on their hardware and these environment does not meet computing with increasing speeds.
- Economic limitations: Making a single processor faster is more expensive than using large a number of processor with same or better performance.
- Future: Distributed systems, faster networks and supercomputer performances point to Parallel Programming.

4.3 Parallel Programming Models

Parallel Programming model is a strategy that how can be problems can be expressed.[5] Some of the models:

- Shared Memory Model,
- Threads,
- Distributed Memory Model,

- Data Parallel Model,
- Hybrid Model,
- Single Program Multiple Data(SPMD) ,
- Multiple Program Multiple Data (MPMD).

Shared Memory Model (Serial Programming): Tasks are sharing an address space to read/write operations. Development is very easy with this model because you don't need to communicate tasks. Shared Memory Model is poor for performance issues when same data usage from multiple processors.

Threads Model: Single process has shared and processing with multiple tasks. Every thread has local data and they are responsible from execute and share them. Global memory is used for threat communication with each other. In this model programmer controls all parallelism operations. POSIX Threads and OPENMP are most used implementation of threads. Detailed information next chapters with threads.

Distributed Memory / Message Passing Model: In this model ,local memories are used for computation. Tasks communicated with each other for data exchange. For implementation side MPI is a standard for message passing implementations. I will give detailed information next chapters with MPI.

Data Parallel Model: A common data set is used by tasks. Each task process different part of common data. Fortran 90 and 95, High Performance Fortran (HPF) is most common parallel platforms.

Hybrid Model: A hybrid model is a technique which uses more than one of the parallel programming models. For example, MPI with OpenMP(Thread Model) is a common and most used hybrid model. In this example, Threads process local data on node with OpenMP and these processes communicate with different nodes with MPI. Also, MPI with GPU is another popular Hybrid Model.

Single Program Multiple Data (SPMD) : This model can be combination of parallel programming models. Tasks operate on copy of the program with a different data. Advantage of this model tasks not execute all program they can work only a part. This model is mostly used for multi-node clusters.

Multiple Program Multiple Data (MPMD): This model like SPMD, can be combination of parallel programming models. Tasks operate on different programs with a different data. MPMD is not as common as SPMD; this model is special for certain problems.

MPI is a standard for message passing implementations. Message passing programs uses this interface because of flexibility, portability, efficiently and practically. MPI defined for FORTRAN and C++ programs.

MPI is founded in 1994 with a 2 year working time. Today, MPI is a mixture of MPI-1(released in May, 1994) and MP-2(released in 1996) [6].

Some reasons to use MPI:

- Standard: MPI is library which is a specific for message passing library and it supports all HPC platforms.
- Portable: MPI does not need source code modifying operations, you can built your application with MPI standard supported platforms.

- Performance: Hardware vendors can customize own routines with using MPI core routines.
- Functionality and Availability: MPI has 105 + routines and it is open to access.

MPI firstly suitable for distributed memory but changing trends change MPI libraries. MPI available for Distributed Memory, Shared Memory and Hybrid hardware platforms. Today, parallelism is clear for programmers. They are identified and implement parallelism with MPI libraries.

Thread is an independent procedure which is set to works independently. Hardware vendors have own threads and there is no common and portable development platform. IEEE establish POSIX as a standard for UNIX systems and POSIX Threads or Pthreads is implementation standard for UNIX systems. Pthreads uses C programing language properties and pthread.h using as a header file.

Pthreads are very efficient for implementations, Pthread creation and management costs are need less system resources. Management of threads more powerful then managing a process. Threads can on work on same address space and thread communications are easier than process communication.

Chapter 5

Hybrid K-means Algorithm

Hybrid K-means Algorithm has been implemented by using parallel K-means algorithm. Hybrization of K-means algorithm provide a faster solution for clustering large datasets. Also, this solution provide same results with parallel version of clustering algorithm.

Parallel algorithm redesigned for hybrid programming with a usage of Pthreads. Pthreads are used for these reasons:

- First reason of the usage Pthread is performance issue, Pthreads are increase the performance of the program,
- Pthread use less system resources than processes. When processes and Pthreads are compared creating and managing operations for Pthreads has less operating system overhead.
- Address space is shared from all threads and thread communications is more efficient from process. Inter- process communications are difficult from thread communications.

Applications with threaded has advantages with non-threaded applications.

- I/O Operations: Assume that program has parts and it is performing I/O operations. So processes are waiting for I/O operations for performing CPU operations. But this situation is different for Pthreads. Only one thread can wait for I/O operations and other threads can perform CPU operations at the same time.

- Important tasks can be scheduled with priority order and they can be stopped the lower priority tasks.
- Pthreads can make operations with asynchronously. For example, while Pthreads operates tasks, they can take new jobs for execution.
- Uni- processor is enough for multi-threaded applications.
- Discrete and independent tasks can be executed concurrently with Pthreads.

In this project, hybrid k-means algorithm has been implemented with existed parallel K-means algorithm codes and K points selected randomly and distance calculations are made in same way.

Next sections are presents hybrid K-means algorithm steps and implementation jobs.

5.1. Hybrid K-means Algorithm Steps

Some steps of existing algorithm are redesigned for Hybrid programming. Hybrid model has common steps with parallel model. Data set shared by memory and each task share this dataset. So tasks are needed to communicate with each other for performing jobs.

Implementation steps of hybrid model are given below:

Assume that N data points are given, k shows cluster numbers and each data point is a vector with P processor. There is a root processor and this processor starts the algorithm steps with initial points. K clusters determined by taking first k points.

First of all, objects are read by the program and initial means are determined by this central processor and central processor broadcast k initial means to other processor

with MPI. This broadcasted means are used as a cluster center for other processors and iterations are starting for K-means algorithm. Every processor delivers internal calculations and number of items to central process with MPI. Central process takes number and sum of items from each process with MPI. So root process has all data, root process and it calculates new cluster means with Pthread and MPI. After that, root process broadcast calculated cluster means with MPI. Each process again do own job until clusters become a constant. In this project there is a control variable &. This variable is incremented by 1 if data object change membership between clusters. Running of algorithm stops when ratio of &/N equals to threshold [12]. Threshold .0001 for this project [12]. Pthreads are used for average the sums and replacing old cluster centers with new cluster centers. This operation is the most time-consuming part in K-means clustering steps. So Pthreads are used for the most time consuming part efficiently. Figure 17 shows hybrid model structure. In hybrid model Pthreads used in Shared Memory with MPI because of decreasing MPI buffering times in memory. Figure 18 shows K-means Algorithm steps with MPI and Pthread.

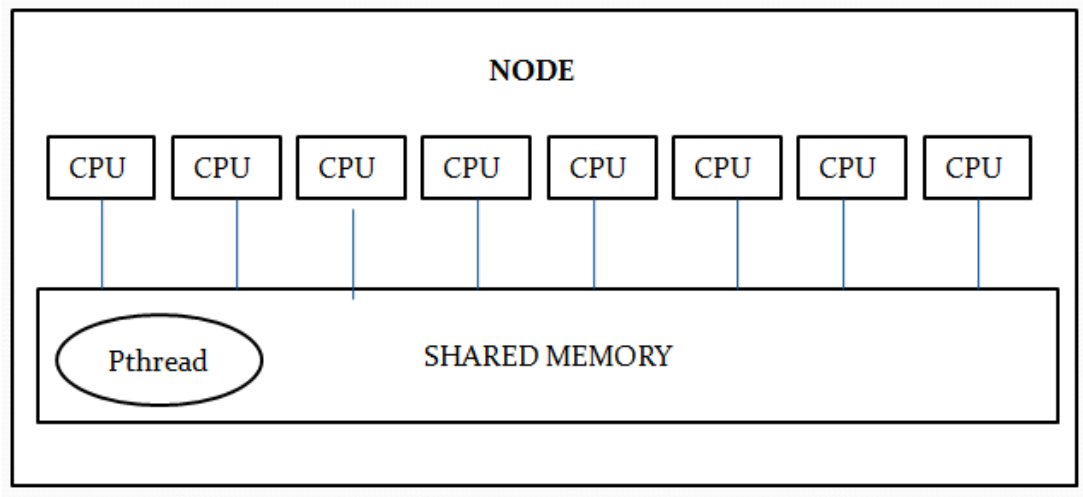

Figure 17: Hybrid Model Structure

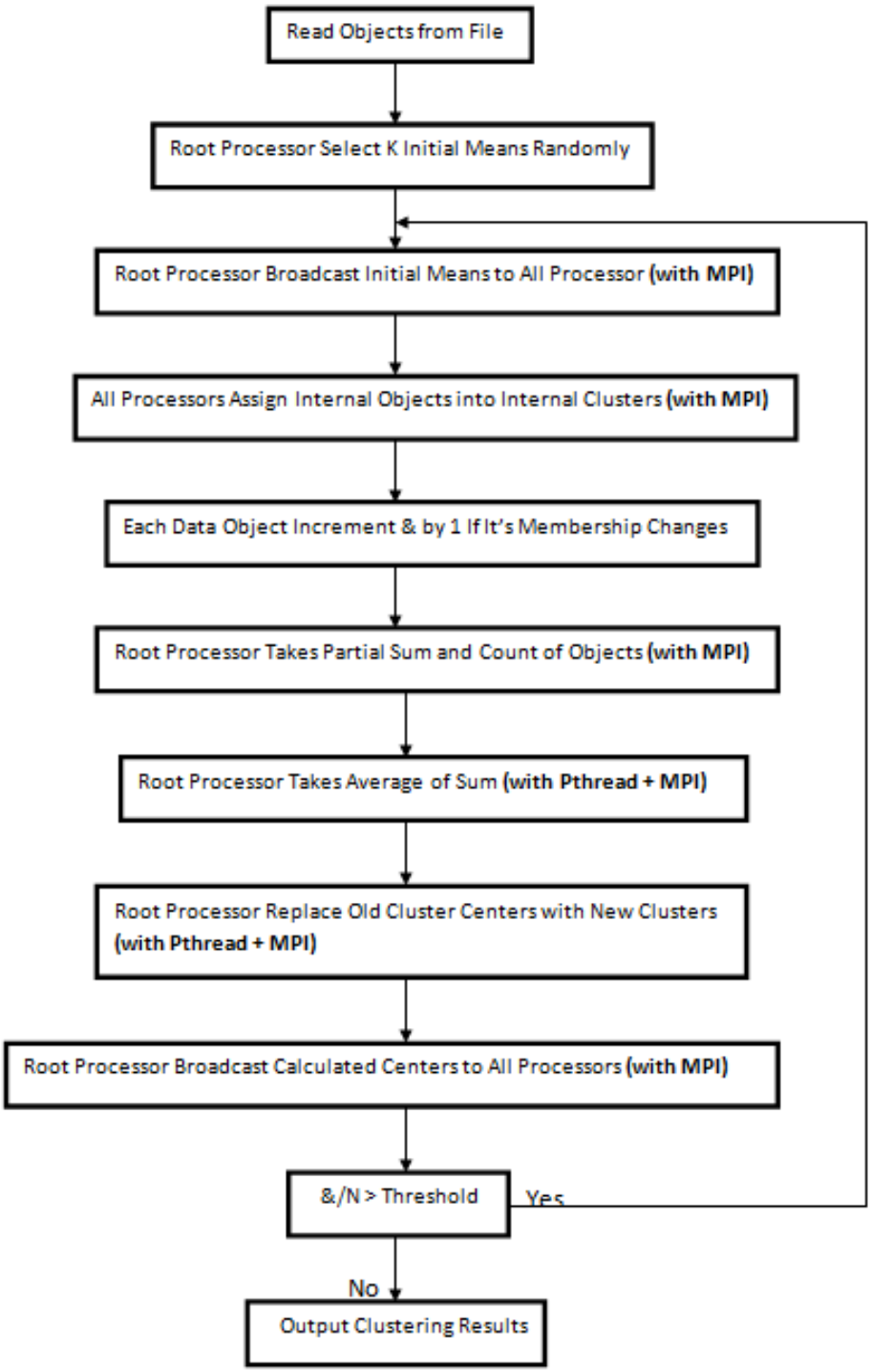

Figure 18: Conversion of Parallel K-means to Hybrid K-means

5.2. Environment for Testing

Experiments made at Linux operating system. Machine has a node with 8 processor and processor model is Intel Xeon(R) CPU X5550 with 2,67 GHz and its cache size is 8192 KB This machine located at Kadir Has University.

Chapter 6

Parallel and Hybrid K-Means Algorithm Results

Parallel and Hybrid versions of algorithm had been executed many times. After that, results are compared, both algorithm produce same results with a different times. Hybrid algorithm execution takes less time than the parallel algorithm and execution results will given with tables and graphs.

6.1 Data Features

Large dataset has used for parallel and hybrid version of K-means algorithm. This dataset US Census 1990 dataset and has 2,458,285 records with 68 dimensions. Dataset is used from UCI KDD archive of California, Irvine. (http://kdd.ics.uci.edu/). This is not an original dataset less useful attributes dropped.

This repository includes large data sets for research studies with different data types and application areas. Main role of the repository is providing complex large dataset for researchers for data mining and reach to knowledge [1]. Dataset includes US Census information of 1990 and dataset has 2,458,285 lines and each line represent a person and row id is given for information Table 1 shows data about person. A person has 68 different properties in this data.

dAge	dAncstry1	dAncstry2	iAvail	iCitizen	iClass	dDepart
iDisabl1	iDisabl2	iEnglish	iFeb55	iFertil	dHispanic	dHour89
dHours	iImmigr	dIncome1	dIncome2	dIncome3	dIncome4	dIncome5
dIncome6	dIncome7	dIncome8	dIndustry	iKorean	iLang1	iLooking
iMarital	iMay75880	iMeans	iMilitary	iMobility	iMobillim	dOccup
iOthrserv	iPerscare	dPOB	dPoverty	dPwgt1	iRagechld	dRearning
iRelat1	iRelat2	iRemplpar	iRiders	iRlabor	iRownchld	dRpincome
iRPOB	iRrelchld	iRspouse	iRvetserv	iSchool	iSept80	iSex
iSubfam1	iSubfam2	iTmpabsnt	dTravtime	iVietnam	dWeek89	iWork89
iWorklwk	iWWII	iYearsch	iYearwrk	dYrsserv		

Table 1: Attribute Names

6.2 Algorithm Execution Methodology

Dataset has been executed 5 times for each condition for getting stable results. Hybrid and Parallel version of K-means Clustering Algorithm select K initial points randomly at the start of the process. Different randomly points had been obtained with 5 times execution and arithmetic means are used for showing performance results. Graphics are used arithmetic means.

Parallel and Hybrid K-means algorithms had been executed with different number of processes for showing performance results. Also, different numbers of Phreads are used. When number of processes and number of Pthreads increases, execution time of the Hybrid algorithm decreases. Several performance results obtained with different combination of number of processes, K initial number and number of Pthread.

6.3 Execution Results

First of all, K cluster number selected as 30, and algorithm run from 1 to 8 processor on a node and all steps executed 5 times, every execution listed in Table 2 and means are calculated with taking average of executions.

# of Processor	**Execution 1**	**Execution 2**	**Execution 3**	**Execution 4**	**Execution 5**	**Mean**
1	691,700 sec	690,255 sec	690,119 sec	691,122 sec	691,568 sec	690,952 sec
2	518,896 sec	518,784 sec	518,608 sec	517,123 sec	518,417 sec	518,365 sec
3	403,927 sec	404,983 sec	403,783 sec	404,702 sec	404,847 sec	404,448 sec
4	322,456 sec	322,469 sec	322,347 sec	322,080 sec	323,851 sec	322,640 sec
5	260,159 sec	260,143 sec	261,094 sec	260,857 sec	260,462 sec	260,543 sec
6	216,848 sec	215,654 sec	216,519 sec	216,042 sec	216,367 sec	216,286 sec
7	184,920 sec	185,651 sec	185,651 sec	185,342 sec	185,692 sec	185,451 sec
8	163,085 sec	163,486 sec	163,042 sec	163,758 sec	163,956 sec	163,465 sec

Table 2: MPI K-means Algorithm, K=30

Figure 19 shows K-means Algorithm MPI version graph and execution times (sec) given for every processor in a node. Execution times are decrease 690,952 to 163,465 when number of processor increases.

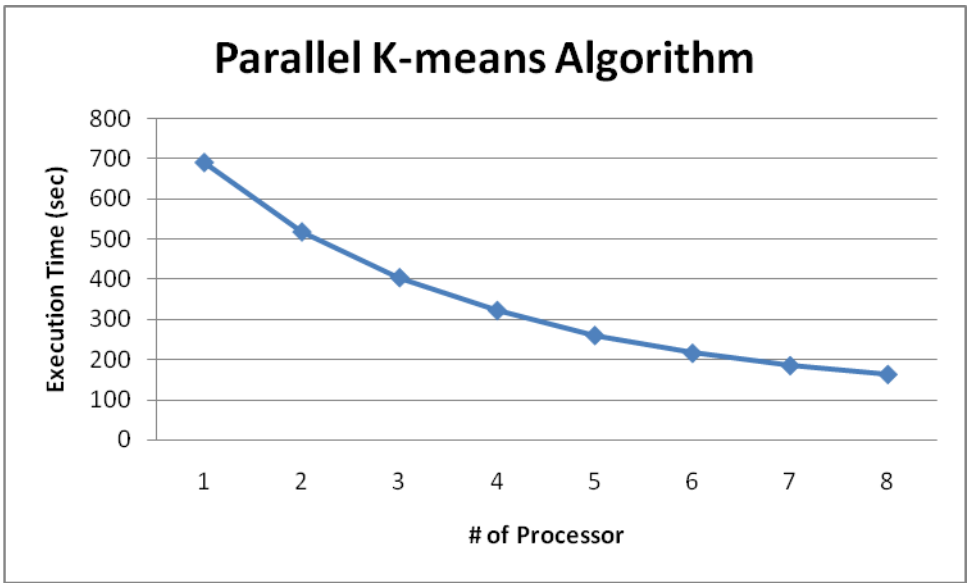

Figure 19: MPI K-means Algorithm, K=30

Table 3 shows means and speedups of MPI K-means algorithm for each processor. Speedup is calculated for every processor with 30 cluster center by using mean values.

# of Processor	**Mean**	**Speedup**
1	690,952 sec	1
2	518,365 sec	1,332
3	404,448 sec	1,708
4	322,640 sec	2,141
5	260,543 sec	2,651
6	216,286 sec	3,194
7	185,451 sec	3,725
8	163,465 sec	4,226

Table 3: Speedup Table for Parallel Version

Figure 20 show speedup graph of MPI K-means algorithm for each processor. Speedup is calculated for every processor to show that when number of processor increases algorithm gain speed for every step and speed increase 1 to 4,226.

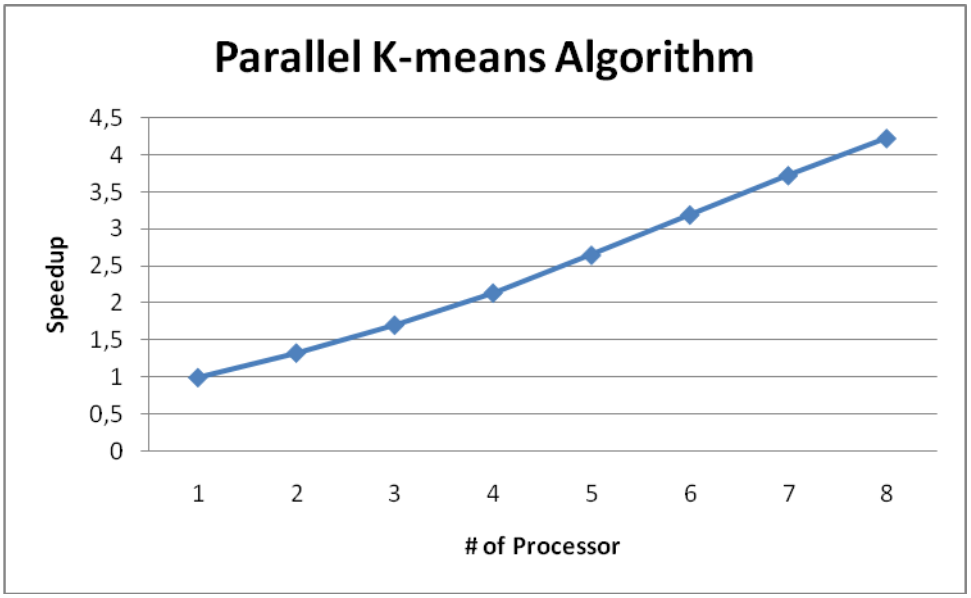

Figure 20: Speedup Graph for Parallel Version

K cluster number selected as 30, and algorithm run from 1 to 8 processor on a node with 16 Pthreads and all steps executed 5 times, every execution listed in Table 4 and means are calculated with taking average of executions.

# of Pthreads : 16, K=30						
# of Processor	**Execution 1**	**Execution 2**	**Execution 3**	**Execution 4**	**Execution 5**	**Mean**
1	99,398 sec	99,588 sec	99,378 sec	99,512 sec	99,552 sec	99,485 sec
2	68,073 sec	68,587 sec	68,179 sec	68,630 sec	68,330 sec	68,359 sec
3	56,598 sec	56,916 sec	56,089 sec	56,018 sec	56,596 sec	56,443 sec
4	45,581 sec	45,844 sec	46,015 sec	45,983 sec	45,729 sec	45,830 sec
5	37,564 sec	38,677 sec	37,956 sec	37,014 sec	37,123 sec	37,666 sec
6	31,081 sec	31,982 sec	31,457 sec	31,569 sec	31,703 sec	31,558 sec
7	27,755 sec	26,693 sec	26,138 sec	26,731 sec	26,830 sec	26,829 sec
8	23,461 sec	23,241 sec	23,092 sec	23,571 sec	23,213 sec	23,315 sec

Table 4: Pthread + MPI K-means Algorithm, Pthread=16

Figure 21 shows K-means Algorithm Pthread + MPI version graph and execution times (sec) given for every processor in a node. Execution times are decrease 99,485 to 23,315 when number of processor increases with using 16 Pthreads.

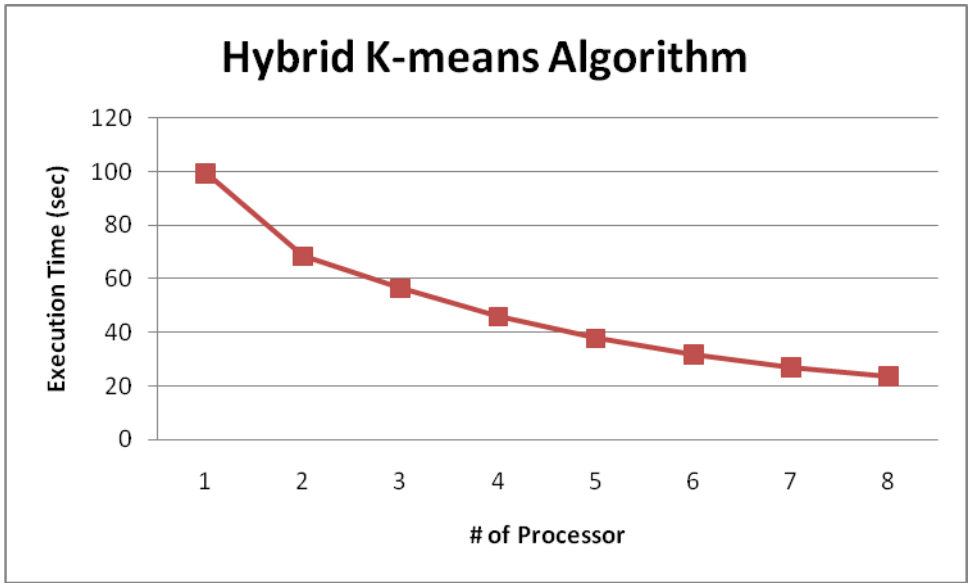

Figure 21: Pthread + MPI K-means Algorithm, Pthread=16

Table 5 shows means and speedups of Pthread + MPI K-means algorithm for each processor. Speedup is calculated for every processor with 30 cluster center and 16 Pthreads by using mean values.

# of Processor	Mean	Speedup
1	99,485 sec	1
2	68,359 sec	1,455
3	56,443 sec	1,762
4	45,830 sec	2,170
5	37,666 sec	2,641
6	31,558 sec	3,152
7	26,829 sec	3,708
8	23,315 sec	4,266

Table 5: Pthread + MPI K-means Algorithm, Pthread=16

Figure 22 shows speedup graph of Pthread + MPI K-means algorithm for each processor. Speedup is calculated for every processor to show that when number of processor increases algorithm gain speed for every step and speed increase 1 to 4,266.

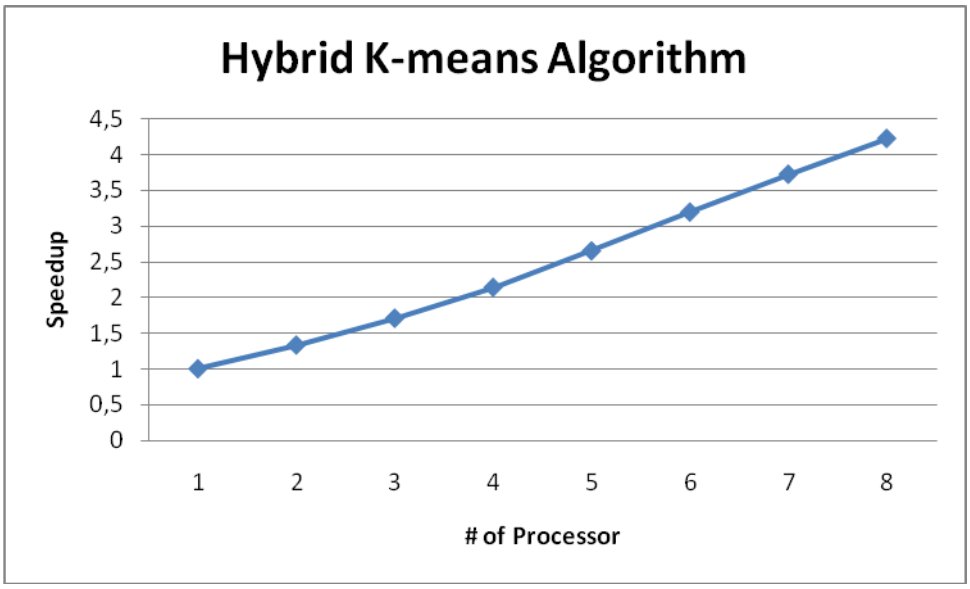

Figure 22: Pthread + MPI K-means Algorithm, Speedup

Figure 23 shows both Parallel and Hybrid K-means Algorithm graphs and execution times (sec) given for every processor in a node. Execution times are decrease for both algorithm. Hybrid algorithm has less execution times because of Pthread usage.

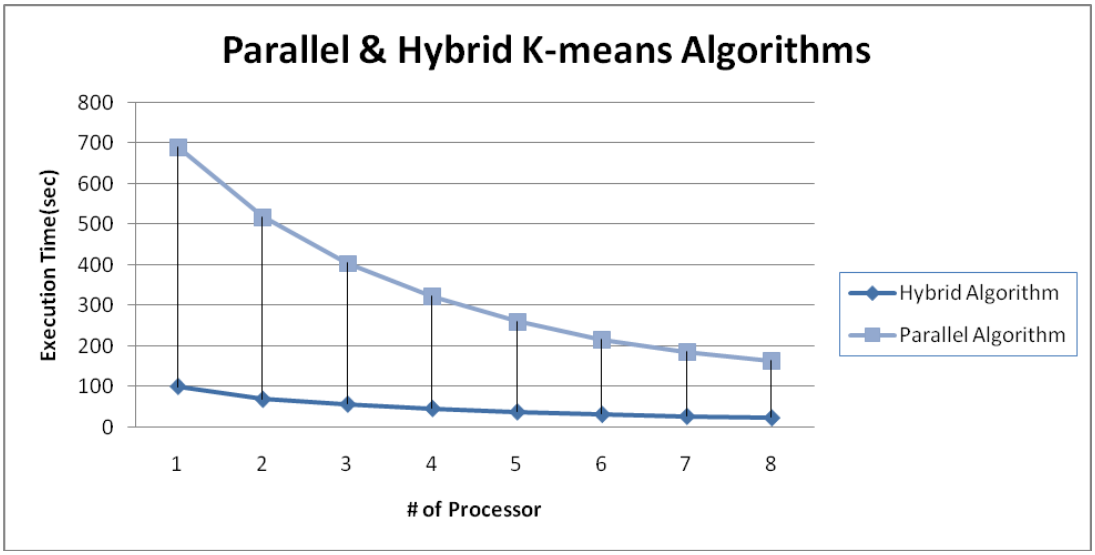

Figure 23: Hybrid-Parallel K-means Algorithm

This time K cluster number selected as 15, and algorithm run on 8 to 16 Pthreads with a 8 processor and all steps executed 5 times, every execution listed in Table 6 and means are calculated with taking average of executions.

# of Processes : 8, K=15						
# of Pthreads	**Execution 1**	**Execution 2**	**Execution 3**	**Execution 4**	**Execution 5**	**Mean**
8	20,931 sec	20,993 sec	20,934 sec	20,972 sec	20,965 sec	20,959 sec
9	20,579 sec	20,573 sec	20,521 sec	20,513 sec	20,537 sec	20,544 sec
10	20,053 sec	20,009 sec	20,073 sec	20,057 sec	20,039 sec	20,046 sec
11	19,369 sec	19,327 sec	19,391 sec	19,323 sec	19,314 sec	19,344 sec
12	18,236 sec	18,257 sec	18,269 sec	18,225 sec	18,293 sec	18,256 sec
13	16,720 sec	16,796 sec	16,768 sec	16,708 sec	16,758 sec	16,750 sec
14	15,069 sec	15,089 sec	15,057 sec	15,083 sec	15,021 sec	15,063 sec
15	13,147 sec	13,145 sec	13,151 sec	13,175 sec	13,186 sec	13,160 sec
16	10,896 sec	10,869 sec	10,807 sec	10,827 sec	10,837 sec	10,847 sec

Table 6: Pthread + MPI K-means Algorithm, P=8

Figure 24 shows K-means Algorithm Pthread + MPI version graph and execution times (sec) given for every Pthread with a constant processor number. Execution times are decrease 20,959 to 10,847 when number of Pthread increases with using 8 processor.

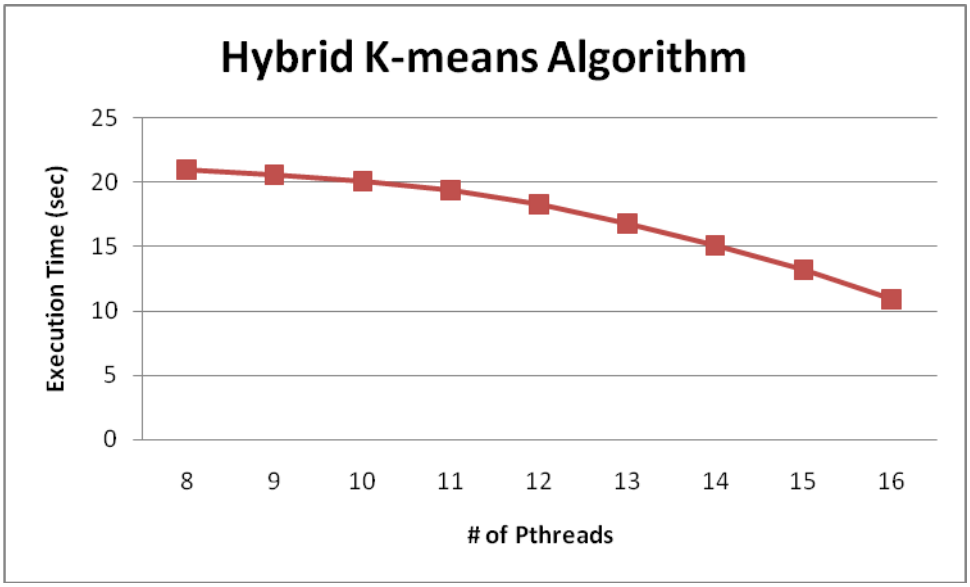

Figure 24: Pthread + MPI K-means Algorithm, P=8

Table 7 shows means and speedups of Pthread + MPI K-means algorithm for each Pthread. Speedup is calculated for every Pthread with 15 cluster center and 8 processor by using mean values.

# of Pthreads	**Mean**	**Speedup**
8	20,959 sec	1
9	20,544 sec	1,02
10	20,046 sec	1,045
11	19,344 sec	1,083
12	18,256 sec	1,148
13	16,750 sec	1,251
14	15,063	1,391
15	13,160 sec	1,592
16	10,847 sec	1,932

Table 7: Pthread + MPI K-means Algorithm, P=8, Speedup

Figure 25 shows speedup graph of Pthread + MPI K-means algorithm for each Pthread. Speedup is calculated for every Pthread to show that when number of Pthread increases algorithm gain speed for every step and speed increase 1 to 1,932.

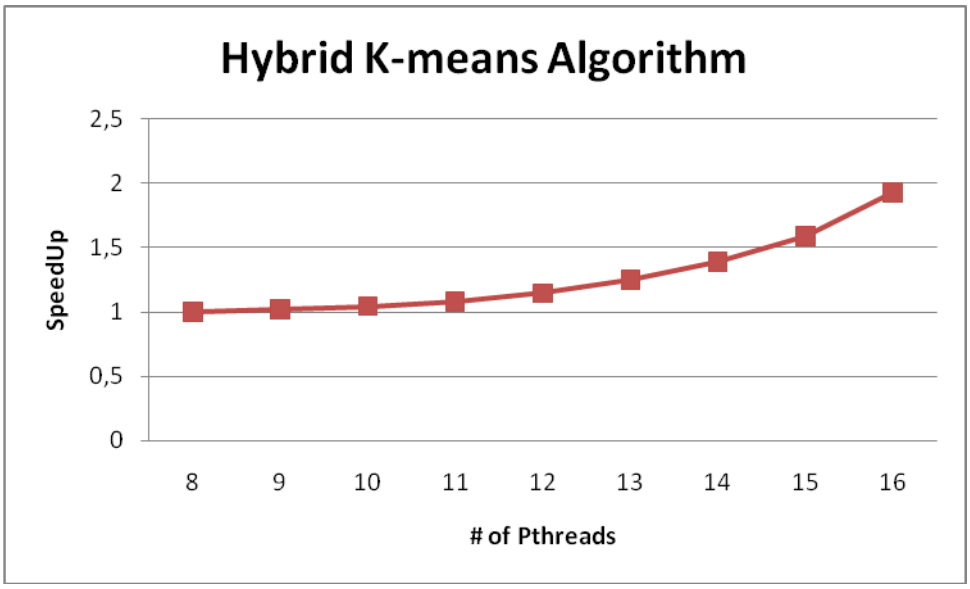

Figure 25: Pthread + MPI K-means Algorithm, P=8, Speedup

Conclusion

In this thesis, Hybrid version of Parallel K-means Clustering had been implemented with using MPI and Pthreads. First of all, serial K-means algorithm learned than parallel version of this algorithm examined.

Hybrization of algorithm is very important because program has parts and it is performing I/O operations. So processes are waiting for I/O operations for performing CPU operations. But this situation is different for Pthreads. Only one thread can wait for I/O operations and other Pthreads can perform CPU operations at the same time. Also, Pthread use less system resources than processes. When processes and Pthreads are compared creating and managing operations for Pthreads has less operating system overhead. Furthermore, address space is shared from all threads and thread communications is more efficient from process. Inter- process communications are difficult from thread communications. These are benefits of Pthreads and only using MPI is not a good solution for large data sets. Hybrid programming is a good alternative.

Finally, Hybrid K-means algorithm explained step by step and existed parallel K-means algorithm source code examined and most time consuming part of the parallel K-means Algorithm had been found. Calculating average of the sums and replacing old cluster centers with new Cluster centers most time consuming steps. So Pthreads are used for this most time consuming parts. Results are compared both of hybrid and parallel K-means algorithm; same results are obtained with same data and same environment. Execution times are compared and hybrid model reduce execution time. This work supported with graphs and tables.

There are some limitations for this work. First of all, existing code need some refactoring for Hybrid Programming and used data set can be larger. Existing dataset

has 2,458,285 person information. K-means Clustering Algorithm does not give information about optimal K number.

As a further research, hybrid programming will use for whole of the existing parallel code and optimal K cluster number can be selected with analysis of data.

References

[1] KDD (April 2013), Web site: http://kdd.ics.uci.edu/

[2] Hohlt, Barbara. *Pthread Parallel K-means.* CS267 Applications of Parallel Computing, UC Berkeley, Web Site: (April 2013) http://barbara. stattenfield. org/papers/cs267paper. pdf .

[3] Fayyad, Usama; *"From Data Mining to Knowledge Discovery in Databases*".

Piatetsky-Shapiro, Gregory; Smyth, Padhraic (1996).

[4] Berkhin, Pavel.: "*Survey of Clustering Data Mining Techniques*", Accrue

Software Inc., San Jose, California, USA (2002).

[5] (April 2013), Website:
http://en.wikipedia.org/wiki/Parallel_programming_model.

[6] (April 2013), Web site: http://en.wikipedia.org/wiki/Message_Passing_Interface

[7] (April 2013), Web site :
http://people.revoledu.com/kardi/tutorial/kMean/NumericalExample.htm

[8] Reza Farivar, Daniel Rebolledo, Ellick Chan, Roy Campbell "*A Parallel Implementation of K-Means Clusteringon GPUs*"

[9] (April 2013), Web site: http://en.wikipedia.org/wiki/K-means_clustering

[10] Sülün, Erhan *"Improvements in K-Means algorithm to execute on large amounts of data"*(2004)

[11] (April, 2013) http://people.revoledu.com/kardi/tutorial/kMean

[12] (January, 2013) Parallel K-means, avaliable at
http://www.ece.northwestern.edu/ wkliao

Curriculum Vitae

Mustafa Alp Çolakoğlu was born in September 26th, 1988, in Istanbul. He received his BS in Computer Engineering in 2011 from Kadir Has University. From 2011 to 2012, he worked as a Assistant Software Engineer at Huawei. Since 2012 he is Development Project Expert in Turkish Economy Bank.