Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/130510

A Novel K-Means Clustering-Based FPGA Parallel Processing in Big Data Analysis

Castro S.[1,*] *and R. Pushpalakshmi*[2]

[1] Department of Computer Science and Engineering, Solamalai College of Engineering Madurai, Tamilnadu, India
[2] Department of Information Technology, PSNA College of Engineering and Technology, Dindigul, Tamilnadu, India

Received: 18 Mar. 2019, Revised: 12 Jun. 2019, Accepted: 15 Jun. 2019
Published online: 1 Sep. 2019

Abstract: Nowadays an enormous amount of dynamic, heterogeneous, complex and unbounded data was obtained from various sectors like social networks, genomics, physics, health, and climatology. The process of operating and managing these data was significantly tedious, at the same, it is important to achieve the desired speed-performance in data processing. In the existing systems, hardware is more operative than the software. The processor-based software which processed earlier has a major disadvantage on the term of an algorithm, it is not effective on dealing with huge volume of data and also on achieving the overall efficiency. On the big data analyses, hardware support is important in order to overcome the real-time issues. The major data mining task to be performed in big data analytics is clustering. It makes the relationship between the object s by means of the similarity and categorizes the data into meaningful groups. In this work, a novel k-means algorithm is proposed to minimize the running time. This algorithm has simple and scalable parallel architecture, which is easy to implement on FPGA-based parallel processing architecture also. This implementation is more efficient for K-means Clustering system on dealing with the big data. It is also applicable for reconfigurable hardware platform such as FPGA, known for the real-time clustering applications.The proposed system is implemented on our hardware design with the benchmark dataset, in order to prove its feasibility and efficiency. Our proposed hardware architecture is more prominent in dealing with different kinds of datasets, with the varying number of clusters as well as a huge volume of data.

Keywords: K-NN,FPGA,Big-data

1 Introduction

In recent years, there is a rapid growth in the evolution of data acquisition techniques and data storage media. It results in the generation of the huge volume of heterogeneous, dynamic, complex and unbounded data from various sources like genomics, physics, climatology, health, social networks, etc. From the last decade,in genomics, the generation of sequence data is doubled in every seven months [1]. On every year, several petabytes of data were generated by the sequencing centers. Still, it's a challenge in analyzing and managing such a vast amount of data. In big data analytics, there are several major data mining tasks that have to be processed. In this research, we concentrated on the most effective data mining task that is followed widely. They are clustering and classification. Clustering and classification involve in selecting certain data and categorize into meaningful groups, by means of the similarity or dissimilarity between the objects. These tasks are widely applied in image segmentation, genome classification according to its Deoxyribonucleic Acid (DNA) data, etc. Nowadays, including the clustering and classification, processing of most data mining tasks are more complex. In general, classification is the supervised form of learning whereas clustering is the unsupervised learning form [2]. On the dataset apart from automatic execution, human interference is important in assigning labels or classes of the sub-groups. Later there is an advance, such as automatic execution is developed in categorizing the data into sub-groups by applying its own assumptions based on the similarity/dissimilarity between the objects.

Most of the clustering and classification algorithms existing are of processor-based (software only) algorithms. It has a severe disadvantage in analyzing and processing vast data effectively. The overcome this issue a software program is required, on the way that processor interprets according to its program instructions and

[*] Corresponding author e-mail: suseelcastro@gmail.com

executes the operations. The program instructions are stored, from which the processor fetches each instruction decoded and then process execution. In case of handling a huge volume of data, an external memory is implemented for fetching, analyzing and processing. In some cases, in order to attain its actual benefit, these data needs to process in real-time. A survey [3] explains the insufficiency in handling a large volume of data. It explains the insufficiency of processor-based computing platforms, including multi-processor, multi-core, GPGPU (General Purpose Graphics Processing Unit) in detail. In big data analytics, it is necessary to develop a new design technique, architectures, and computing platforms in order to overcome these existing issues. In big data analytics, as per the real-time aspects to satisfy the constraints and requirements it is important to implement some kind of hardware support. In this research work, we analyze several special-purpose hard wares. Special-purpose or customized hardware is nothing but specific optimization that is done on the hardware to perform a specific application. The main motto of this customization in processor-based software-only designs is to achieve the high execution on fetch/decode/execute instructions. This customized hardware achieves excellence in speed performance, lower power consumption [4], and area-efficiency comparing to the software running on a general-purpose processor. We concentrated on one of the effective clustering algorithms in this domain, known as K-means clustering. We research and provide an effective proposed hardware architecture which is perfect for K-means clustering. Our hardware design is excellent in performing parallel multiple computations and improvise the algorithms speed performance, by implementing inherent parallelism and pipeline nature of the operations. The Field Programmable Gate Array (FPGA)-based development platform is used in designing our proposed hardware architecture. This FPGAs enable the ease to design, develop, and implement. It is more effective in compute- and data-intensive algorithms in the hardware. This research represents that our proposed FPGA-based hardware design with K-means clustering algorithm improvises rapidly in minimizing the execution time comparing to other processor-based software. Thus our proposed architecture typically enhances the overall speed performances.

2 Related work

In his research, T. Kucukyilmaz [5] implement the standard K-means algorithm parallel for shared memory multiprocessors. The author uses standard Euclidean distance as the chosen metric. Initially, the implementation started by parallelizing, even though it reflects in a low impact on final speed-up. On the aspect of computation, the assignment step is considered as the most demanding step as it to parallelized between the processors. In which each processor carries the task of assignment step on the subset of the dataset and the error results such as partial mean square is stored. The shared memory holds the results of all data points, based on that the update step can also perform parallelized among the processors. For each processor, a new center of a subset of clusters can be achieved by these parallelized update steps. Finally, from all the slave processors, the master processor collects the partial mean squared errors. This solution is perfect for handling huge datasets with various dimensions, the increasing computational time of the algorithm can be also balanced by the parallel implementations. By applying a suitable dataset, the implementation achieves maximum speed-up of 4, using 8 Intel Nehalem-EX Xeon 7550 cores at 2.0 GHz.

The PAM (Partitioning Around Medoids) algorithm is the widely uses partitioning-based methods of clustering. The major drawback of PAM is not effective on dealing with large datasets or any embedded or real-time application.This makes the researchers do some modifications and overcome the complexities of PAM resulting in speed up of the algorithms. To achieve this a basic multi-core implementation of K-Medoids is proposed. By means, it divides the algorithm into several sub-tasks and each individual sub-task is carried out on a separate core. This enhanced architecture increase of speedup (4 while utilizing 16 cores), In the same aspect Rechkalov in [6] proposed a multi-core solution for the Intel Xeon Phi Many-Core Coprocessor.The author implements OpenMP parallelizing technology and loop vectorization in the algorithm additionally with tiling approach.The result obtained by the optimized version of the algorithm achieves overall performance according to the data nature to be clustered [6].

The proposed architecture has homogenous processing cores, which are executed parallel as per the independent data subsets.To share these cores are linked by means of a bus. The proposed design is operated on a collaborative working environment, which enables independent execution of cores and data sharing among themselves to get the results.This design can speed up by using only 10.31 percent of the total device slice registers and 33 percent of total slice LUTs of a Spartan 6 FPGA.For processing the real-time image processingmulti-processor architecture having heterogeneous tiles was represented in [7]. On that, every tile of the architecture is built with computational and memory capabilities. A novel NoC structure named Spidergon is implemented for connecting the tiles. This proposed architecture is applicable for various algorithmic classes and runs at 400 MHz as per the real-time processing of up to 30 VGA frames/s [8].Several studies on optimizing the Particle Swarm Optimization (PSO)hardware design and its algorithm were proposed byMehmood et al. in [9].

3 Proposed Design Flow

The main motto of the proposed work implementation is to achieve clustering by means of K-means algorithm.K-means algorithm is most popularly known simplest algorithms for computing the clustering task. Its easy implementation and rapid execution time make this algorithm as most widely-used clustering algorithms among the research community. The algorithm computes with centroid model, which segments data into cluster sets and each segment is demonstrated by the mean of vector data points among the class. All data points are represented as a cluster formation which form of nearest center. To measure the distance between the data points Euclidean distance is used as a metric along with some other kind of matrices that can also be applied. At each center, the initial position is attributed and then the algorithm updates its position by its iterative fashion.

The working procedure of the proposed algorithm can be demonstrated by the following steps

1. By the chosen distance metric, the data point is assigned to the closest center.

2. All the assigned data points are arranged and centers are re-calculated. Within the cluster, the new position as per the mean of all the data points is calculated

3. The algorithm portions to execute parallel is identified.

4. In order to form the final cluster, all the intermediate results from different PEs were combined.

```
$Step-1$
$Centralization();$
$InitializethedatasetD_ix;$
$For$
$Calculatethedistancepointfromcentredata$
$point[CD]ix; \quad Findmin_distancefromd = C-> D_{ix}(currentdistance < mic-distancedatapoint);$
$StoreBuffer(K):d$
$Return();$
$Endfor;$
$Step-2$
$Foreach(datapoint)//clusterformation$
$Calculatedistanced;$
$update-Buffer(K):[CD]-ix(d);$
$Grouping-datapoint(C,D-ix)->G-c;$
$Grouping-nearestdatapoint(G,c)->cluster$
$formation(C-l);$
$clusterformation(C-l)Calculatemin-distance$
$(d)(C-l->d);$
$updatebuffer(K)<-d;$
$endfor$
$return();$
$Step-3$
$Forclassification$
$Classification(C-l);$
$Storebuffer(K)//classificationstoretheresults$
$Endfor;$
$Return();$
```

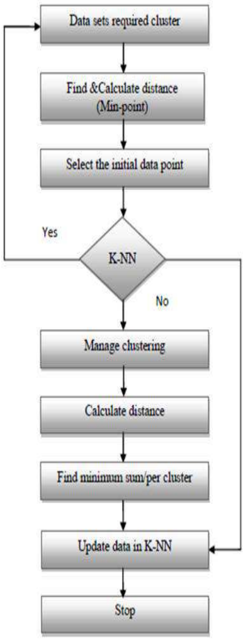
Data sets required cluster
Find &Calculate distance (Min-point)
Select the initial data point
Yes
K-NN
No
Manage clustering
Calculate distance
Find minimum sum/per cluster
Update data in K-NN
Stop

Fig. 1: Algorithm Novel K-nearest neighbor selection

4 Clustering and Classification

In most of the data mining applications, clustering and classification are the major tasks to be processed [10]. It categorizes the data set and groups into sub-groups based on the similarity among the objects. The grouping of sub-groups is based on both similar as well as dissimilar as possible. Clustering is done as per its inter-pattern similarities and classification labels the dataset into smaller sample/training set. Then the rules are applied to the labeled groups to map the data [11,12,13]. Both the process needs to be effective and the pattern which is not very apparent is also high lightened. the sample training and testing Both task results in the grouping as is different in its own way. In clustering, on the whole dataset, the grouping is executed by the algorithm. In classification, based on the training set grouping/labeling pre-exists and the algorithm is applied to the labeled ones to map new vectors. In our research work, we concentrated on clustering algorithms.

5 Parallel and Distributed in K-Means

In our proposed architecture, Processing Elements (PEs) are linked together by means of an interconnecting network known as the bus. Figure 2 clearly portraits the interconnected network and each PE has the capability of

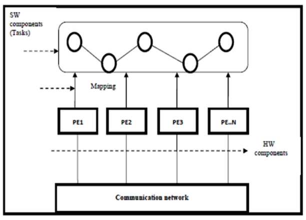

Fig. 2: Top-level block diagram of the proposed architecture

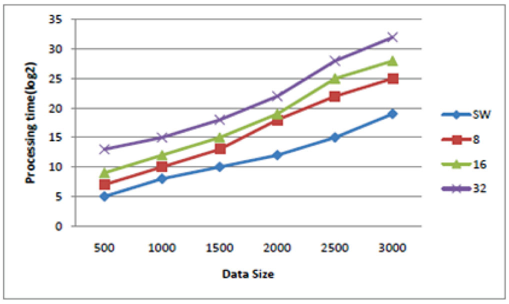

Fig. 3: Datasize vs processing time

accessing all data points which can be operated parallel with others to obtain high convergence and eventually an increased throughput. The overall flow of the algorithm is controlled by the communication network. In the proposed design the interconnecting network is of bus-based, point-to-point or network-on-chip-based interface.Based on the application complexities and requirements the choice of interconnection is done. For example to achieve concurrent message passing at the cost of area and power overhead the NoC-based interface is applied.

1. The size of the data set is N, which is segmented into several available cores PE. Each core is appended by zero at end of data set. These subsets are equally assigned to the available cores.

2. The entire data set X is simulated in all available PEs and the data set X is equally partitioned which are assigned to each PE.

3. The subtask “Find Minimum Sum” builds phase for respective data subset of size in buffer (k) parallel which is executed by each PE. All PEs are homogenous which function as master PE, to perform as master PEany processing element that can be assigned. From each PE all results of the first subtask are collected and execute the subtask “Select Initial Medoid”.As per the algorithm,In this step carried out still the K medoids get initialized.

4. The PEs collect the final results of Build Phase and send to the algorithm for executing the next phase.

5. In the dataset, all the assigned data objects with its nearest cluster numbers are tagged with appropriate PEs. All the tags connected with its data object are stored in the local memory. The PEs broadcast their tags one by one by means of the interconnecting network. By doing this each PE obtain complete clustering results.

6 Experimental results

To examine the efficiency of our proposed system “Wholesale Customer Data” [2] benchmark dataset is taken with the embedded hardware architecture for K-Means clustering algorithm. The size of the dataset is 3520 and there are 440 8-attribute vectors. Our proposed methodology is applicable for processing data set regardless of the size. The varying data size processes its scalability, in which the number of attributes as well as the number of bits of the attributes remain the same only size of the data or the number of vectors differs. To ensure the speed performance, the experiment is conducted with a different number of clusters for the given dataset. A parallel processing architecture is implemented to test the hardware advantages. Here the multiple Euclidian distance computations on parallel are conducted by using multiple PEs. The experiment is conducted with various hardware configurations with a varying number of PEs such as 1PE, PEs, 16PEs, and 32PEs. In this work, we design and apply embedded software for the K-Means clustering algorithm in order to examine the performance of our embedded hardware designs. In the same development platform, the software model is processed on the MicroBlaze processor. A hardware timer is used for obtaining the execution time of both hardware and software designs. On every clock cycles, the execution time is calculated by a standard unit with time/speedup of K-Means Clustering operated on various platforms. The execution time is represented for 8, 16, 24, and 32 clusters in the observations. The observations based on the four different hardware configurations (Hw) along with the software design (Sw) for varied data sizes (i.e., a varying number of vectors) as well as a varied number of clusters are plotted in the graph

Figure 7 represent the results of execution times versus the data sizes(number of vectors). Here the results are obtained for various hardware configurations(with varying number of PEs) and the software on MicroBlaze for 32 clusters. figure 8 represents result of execution times versus the number of iterations. Here also various hardware configurations (with varying number of PEs) and the software on MicroBlaze for 32 clusters is implemented. The obtained results are plotted on the graph, in which the top lines show the software on MicroBlaze execution time. Where the bottom line

© 2019 NSP

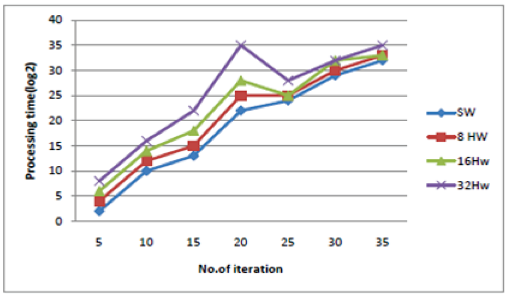

Fig. 4: No.of iteration vs processing time

represents hardware configuration execution time with 32 PEs.

From fig-8 and with the obtained results it is clear that the execution time for K-Means clustering is increased but not linear. It varies for different hardware configurations (with varying number of PEs) and for software on MicroBlaze. In the execution time it is only based on the data size but also the number of iterations taken by the k-means to form the clusters. There is a similar cluster behavior among the 8, 16, and 24 clusters.

7 CONCLUSION AND FUTURE WORK

In this paper, an enhanced hardware architecture for K-Means clustering algorithm is proposed for big data analysis. In section IIA, We analyze the various problems in existing hardware designs for the K-Means Clustering. Our proposed hardware architecture is well standard, scalable and parameterized. It is applicable for executing the data of various sizes (any number of vectors and any number of attributes), various clusters of different applications as well as various hardware platforms.In order to achieve the speed performance, the proposed design is applicable for customization as per the number of parallel PEs. In comparison to the software counterpart, our implementation of hardware configuration with 32PEs is executed up to 368 times greater in speed. To minimize memory access latency, several advanced methods are incorporated. In our previous studies on data mining, the memory access latency on hardware support is the severe issue. These experimental results boost up the implementation of K-Means clustering algorithm in big data analysis using FPGA-based parallel hardware.Our current work as about analyzing a method which integrates the novel and efficiently-embedded architectures to larger systems. Along with the larger systems integration includes big data centers and genomic sequencing centers in order to minimize the computation complexities as well as maximizing the system's efficiencies.

References

[1] D. Stephens, S. Y. Lee, F. Faghri, R. Campbell, C. Zhai, M. Efron, S. Sinha, R. Iyer and R. Gene ”Big Data: Astronomical or Genomical?,” PLoS Biol.**14** 13 (7) (2015)

[2] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Prentice Hall Advanced Reference Series **21** (1) 137-138(1988).

[3] D. Singh and C. Reddy, ”A survey on platforms for big data analytics,” Journal of Big Data **2**(1)1-20 (2015).

[4] H. M. Hussain, K. Benkrid, H. Seker and A. T. Erdogan, ”FPGA Implementation of Kmeans Algorithm for Bioinformatics Application: An Accelerated Approach to Clustering Microarray Data,” in Proc NASA/ESA Conference on Adaptive Hardware and Systems.**2**48-255(2011)

[5] T. Kucukyilmaz, “Parallel k-means algorithm for shared memory multiprocessors,” Journal of Computer and Communications **399**,2(11)15-23 (2014).

[6] Rechkalov, T.V. Partition Around Medoids Clustering on the Intel Xeon Phi Many-Core Coprocessor. in Proc of the 1st UralWorkshop on Parallel, Distributed, and Cloud Computing for Young Scientists (Ural-PDC 2015), Yekaterinburg, Russia **,** 29-41 (2015).

[7] Mehmood, S. Cagnoni, S. Mordonini, M. Farooq, M. Particle swarm optimisation as a hardware-oriented meta-heuristic for image Analysis. in Proc of the Workshops on Applications of Evolutionary Computation, Tübingen, Germany **,** 369-374 (2009).

[8] Saponara, S.; Fanucci, L.; Petri, E. A multi-processor NoC-based architecture for real-time image/video enhancement. J. Real-Time Image Process **,** 8(1) 111-125 (2013).

[9] Mehmood, S.; Cagnoni, S.; Mordonini, M.; Khan, S.A. An embedded architecture for real-time object detection in digital images based on niching particle swarm optimization. J. Real-Time Image Process **,** 10(1) 75-89 (2015).

[10] D. Hand, H. Mannila and P. Smyth, Principles of Data Mining, The MIT Press Nature **,** 20-150 (2001).

[11] B.Sc thesis EVALUATION OF DIFFERENT CLUSTERING AND CLASSIFICATION ALGORITHMS FOR CONTINOUS AND DISCRETE DATA SETS **,** (2012)

[12] B. K. Lavine, ”Clustering and Classification of Analytical Data”.Encyclopedia of Analytical Chemistry. **,**1-21 (2006)

[13] A. K. Jain, M. N. Murty and P. J. Flynn, ”Data Clustering: A Review,” ACM Computing Surveys**,** 31(3) 264-323 (1999).

Castro S. is an Assistant Professor of Computer Science and Engineering at Solamalai College of Engineering (Approved by Anna University), Madurai, Tamil Nadu, India. He received the B.Tech degree in Information Technology from PSNA College of Engineering and Technology, (Approved by Anna University),Dindigul, TamilNadu, India, in 2010, and the M.Tech. degree in Computer Science and Engineering from Karunya University, Coimbatore, TamilNadu, India, in 2012. Hir main areas of research interest are Data Mining, Big Data, Cloud Computing and Network security. He published many papers in international and national journals.

R. PushpaLakshmi is a Professor of Information Technology at PSNA College of Engineering & Technology (Anna University), TamilNadu, India. She received her PhD in Information and Communication Engineering from Anna University, Chennai, India, in 2014. She received the B.E. degree in Computer Science and Engineering from Madurai Kamaraj University, TamilNadu, India, in 2001, and the M.E. degree in Computer Science and Engineering from Anna University, Chennai, India, in 2004. Her main areas of research interest are Wireless Networks, Network Security, Soft Computing and Data Mining. She is a Life Member of the Indian Society for Technical Education (ISTE).

© 2019 NSP
Natural Sciences Publishing Cor.