Parallel K-Means Clustering Algorithm on DNA Dataset

Fazilah Othman, Rosni Abdullah, Nur'Aini Abdul Rashid, and Rosalina Abdul Salam

School of Computer Science, Universiti Sains Malaysia, 118[illegible] Malaysia
{fazot, rosni, nuraini, rosalina}@

Abstract. Clustering is a division of data into groups of similar objects. K-means has been used in many clustering work because of the ease of the algorithm. Our main effort is to parallelize the k-means clustering algorithm. The parallel version is implemented based on the inherent parallelism during the Distance Calculation and Centroid Update phases. The parallel K-means algorithm is designed in such a way that each **P** participating node is responsible for handling **n/P** data points. We run the program on a Linux Cluster with a maximum of eight nodes using message-passing programming model. We examined the performance based on the percentage of correct answers and its *speed-up* performance. The outcome shows that our parallel K-means program performs relatively well on large datasets.

1 Introduction

The objective of this work is to partition data into groups of similar items. Given a set of meaningless data and sets of representatives, our work will group the data according to the nearest representative. This work is useful in helping scientists explore new data and lead them to new discovery in relationships between data. It is widely employed in different disciplines which involve grouping massive data such as computational biology, botany, medicine, astronomy, marketing and image processing. A survey [1][2][3] on clustering algorithm reported that K-means is a popular, effective and practically feasible method widely applied by scientists. However, the rapid growth of data makes the processing time increase due to large computation time. [2] has implemented the K-means algorithm on DNA data using positional weight matrices (PWM) training The decreasing prices of personal computers make parallel implementation a practical approach. In this paper, we propose a parallel implementation of **K**-means clustering algorithm on a cluster of personal computers. This is to provide a practical and economically feasible solution.

2 K-Means Clustering Algorithm

K-means algorithm works conveniently with numerical values and offers clear geometric representations. The basic K-means algorithm requires time proportionate to number of patterns and number of cluster per iteration. This is computationally expensive especially for large datasets [4]. To address these problems, parallelization

K.-M. Liew et al. (Eds.): PDCAT 2004, LNCS 3320, pp. 248–251, 2004.

has become a popular alternative which exploits the inherent data parallelism within sequential K-means algorithm. Efforts in parallelizing the K-means algorithm has been done by [1][5][6][7][8] in areas such as image processing, medicine, astronomy marketing and biology. As our contribution we propose a parallel K-means clustering algorithm for DNA dataset running on a cluster of personal computers.

3 Parallel K-Means Clustering Algorithm

The sequential algorithm spends much of it's time calculating new centroid and calculating the distances between **n** data points and **k** centroids. We can cut down the execution time by parallelizing these two operations. Our parallel **K-means** algorithm is parallelized based on the inherent data-parallelism especially in the *Distance Calculation* and *Centroid Update* operations. The *Distance Calculation* operation can be executed asynchronously and in parallel for each data point (x_i for $1 \leq \mathbf{i} \leq \mathbf{n}$).

We designed the parallel program in such a way that each participating **P** processor is responsible for handling **n/P** data points. The basic idea is to divide the **n** data points into **P** parts which are the approximate size for the portion of data which will be processed by the **P** independent nodes.

However, each of the **P** nodes must update and store the mean and **k** latest centroid in the local cache. The master node will accumulate new assigned data points from each worker node and broadcast new global mean to all. The **k** centroids allow each node to perform distance calculation operation in parallel while the global mean permit each node to decide on the convergence condition independently.

The *Centroid Update* is performed in parallel. It is operated before the new iteration of K-means begins. New centroids will be recomputed based on the newly assigned data points in k centroids. Each node that performs *Centroid Update* need to communicate simultaneously since the computation requires the global mean accumulated by the master node. The parallel K-means algorithm design is shown in Figure 1.

4 Implementation and Result

We run the program on Aurora Linux Cluster with a maximum of 8 nodes using message passing programming model. Each node has CPU speed of 1396 MHz, swap memory of 500 Mb and total memory of 1 GB for clients and swap memory of 520 Mb and total memory of 2.0 GB for the server. We tested on three datasets which have been statistically analyzed and published by P.Chaudhuri and S.Das[9] to benchmark our cluster result. The three datasets are ribosomal RNA for twenty four organisms, vertebrate mitochondrial DNA sequences and complete genomes of roundworm. Next we are interested in studying the impact of parallelizing the sequential K-means algorithm in terms of performance. Thereby we assume our cluster result is acceptable. Figure 2(a) shows the result of executing parallel K-means algorithm on ribosomal RNA sequences of 24 organisms and Figure 2(b) shows the result

of executing parallel K-means algorithm on the artificial dataset of 15.7 MB, which consist of 16 sequences, each of length 1 million base pair. We examined the performance based on the percentage of correct answers and highlight the speed-up. The outcome shows that our parallel K-means algorithm performs relatively well on large dataset.

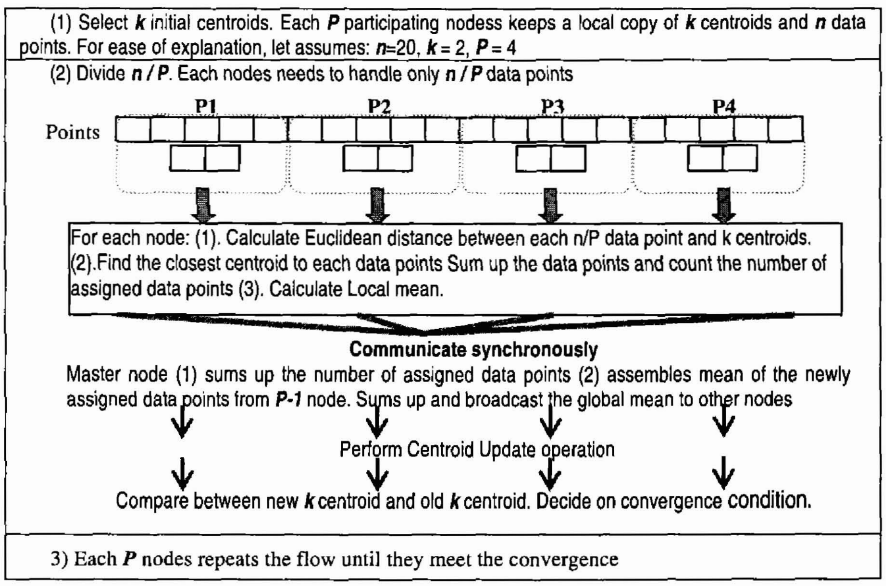
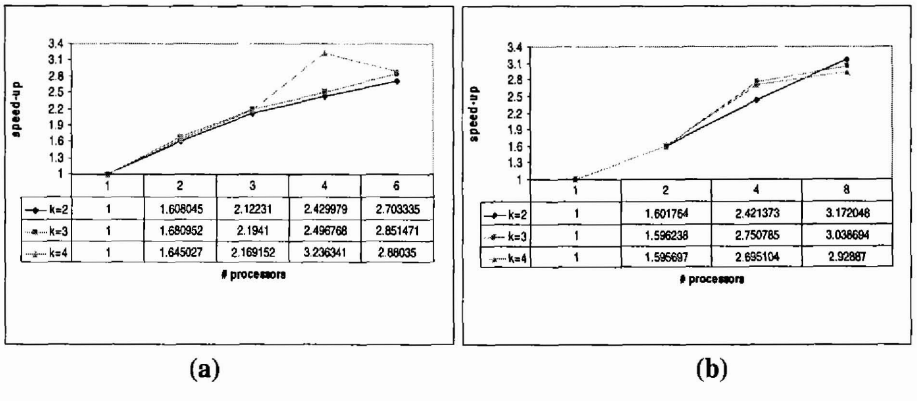


Fig. 1. The diagram of Parallel K-means Algorithm



(a) (b)

Fig. 2. (a) Speed-up performance for parallel K-means using ribosomal RNA sequences of 24 organisms; (b) Speed-up performance for parallel K-means using artificial dataset

5 Conclusion and Future Work

The experiments carried out showed that the parallel K-means algorithm starts making progress on a large dataset. In order to improve the accuracy of the cluster results, we observed that attention should be given to the data training phase. In our program, we applied the PWM method where we calculated the frequency of nucleotides A, T, C and G for each position in the sequences. However, the DNA sequence is very rich with gene information and the arrangement within the nucleotides gives crucial information. It is very interesting to employ other method called the distribution of DNA words that focus on the word frequency-based approach as reported in [9]. We hope to port our work to a high performance cluster of Sun machine. The SUN Cluster is a new facility provided by the Parallel and Distributed Computing Centre, School of Computer Science, USM. With 2 GB memory space on the server machine alone and a total hard disk external storage of 70 GB on the clusters machines, it is hoped that it will produce more encouraging results.

References

1. Inderjit S. Dillon and Dharmendra S. Modha, "A Data-Clustering on Distributed Memory Multiprocessors" in ACM SIGKDD Workshop on Large-Scale Parallel KDD System (KDD 99), August 1999.
2. Xiufeng Wan, Susan M. Bridges, John Boyle and Alan Boyle, "Interactive Clustering for Exploration of Genomic Data", Xiufeng Wan, Susan M. Bridges, John Boyle and Alan Boyle, Mississippi State University, Mississippi State, MS USA,2002
3. K.Alsabti, S.Ranka and V.Singh. An Efficient K-Means Clustering Algorithm. http://www.cise.ufl.edu/~ranka/, 1997
4. K.Murakami and T.Takagi, "Clustering and Detectionof 5' Splices Sitesof mRNA by K Wight Matrices Model",Pac Symp BioComputing,1999, pp 171-181.
5. Kantabutra S. and Couch A.L, "Parallel K-means Clustering Algorithm on NOWs",NECTEC Technical Journal,vol 1, no.6 (February 2002),pp 243-248.
6. Killian Stoffel and Abdelkader Belkoniene, "Parallel K/H-means Clustering for Large Data Sets",Proceedings of the European Conference on Parallel Processing EuroPar'99, 1999.
7. Kantabutra S., Naramittakapong, C. and Kornpitak, P, "Pipeline K-means Algorithm on NOWs," Proceeding of the Third International Symposium on Communication and Information Technology (ISCIT2003),Hatyai,Songkla,Thailand,2003.
8. Forman,G and Zhang , B., "Linear Speed-Up for a parallel Non-Approximate Recasting of Center-Based Clustering Algorithm,including K-Means,K-Harmonic Means and EM," ACM SIGKDD Workshop on Distributed and Parallel Knowledge Discovery (KDD2000), Boston, MA, 2000.
9. Probal Chaudari and Sandip Dass,"Statistical Analysis of Large DNA sequences using distribution of DNA words", CURREBT SCIENCE, vol. 80, no. 9(10 may 2001) pp 1161 – 1166.