2012 International Conference on Radar, Communication and Computing (ICRCC),
SKP Engineering College, Tiruvannamalai, TN., India. 21 - 22 December, 2012. pp.221-223.

Parallel K-Nearest Neighbor Implementation on Multicore Processors

P.P. Halkarnikar[1], Ananda P. Chougale[2], H.P. Khandagale[2] and P.P. Kulkarni[3]

Abstract— **As the industry moves from single chip processors to multi-core processors in the general purpose community, it is becoming increasingly important to develop techniques to find and expose enough parallelism in the application programs. Parallel programming is classified in to two major groups as code parallelism and data parallelism. In order to exploit the power of multi core processors it is essential to change programming of conventional application to parallel programming paradigms. Some compiler tools have been developed to help the programmer to develop parallel applications. However, it is still a challenging problem to programmer to extract full parallelism in general applications. Here we propose a case study of classification of huge database like electoral data of Kolhapur constituency in to age wise groups using popular technique of classification using K-Nearest Neighbor on multi core CPUs. Such a classification of data will predict the age group of constituency which will help the contestant to arrange their campaign accordingly. Also trend of voting can be associated to age groups for analysis. This application demonstrates how parallel programs can be developed using multi core processors to take full advantage of parallel programming on desktop.**

Keywords— **Parallel Programming, Multi core Processor, Data Mining, K-Nearest Neighbor**

I. INTRODUCTION

The deluge of available data for analysis demands the need to scale the performance of data mining implementations. CPU manufacturers like INTEL AMD have all ready started producing multi core processors for desktop and laptops giving the programmer the parallel computing paradigms. It is no longer possible to improve processor performance by simply increasing clock frequencies. As a result, multi-core architectures have become cost-effective means for scaling performance. Major Multi core architectures are of increasing importance and are impacting client, server and supercomputer systems [1]. Multi core CPU allow parallel programming to achieve high performance computing Thus, one of the major challenges today is achieving programmability and performance for data mining applications on multi-core machines and cluster of multi-core machines.

In this paper we proposed K-Nearest Neighbor parallel implementation on multi core processors . The K-Nearest Neighbor algorithm is a widely applied method for classification in machine learning and pattern recognition. However, we are not able to get a satisfactory performance in many applications, as the K-Nearest Neighbor algorithm has a high computational complexity [2]. The K-Nearest Neighbor algorithm is simple in calculation and can be applied to high-dimensional data sets. Nevertheless, when the test set, train set, and data dimension are larger than expected, the computational complexity will be huge and the operation time will be very long.

Here we propose a case study of classification of huge database like electoral data of Kolhapur constituency in to age wise groups using popular technique of classification using K-Nearest Neighbor on multi core CPUs. Electoral are classified as YOUNG, MIDDLE, OLD. Total Electoral of 5 Lakhs (approximately) are divided in to 72 wards of around 7000 electoral each. Out of that we have taken 4 wards for our study. We have processed 4 wards on Single CPU and Independently on Multi Core CPU for comparison. The result shows considerable improvement in processing time.

II. THE GENERAL ARCHITECTURE OF THE SYSTEM

As shown in figure 1, the system consists of four modules. The major modules are as follows

1. Voters Data Base
2. User Module
3. Data Mining Module
4. Presentation Module

A. Voters data base

The Data base contains Data from electoral list of Kolhapur constituency. Total Electoral of 5 Lakhs (approximately) are divided in to 72 wards of around 7000 electoral each. Out of that we have taken 4 wards for our study. This database is available publically on www.Kolhpurcollector.ernet.in. This data is stored using SQL server 2005.

[1] Dept. of CSE, D. Y. Patil College of Engineering, Kolhapur.
Email: pp_halkarnikar@rediffmail.com
[2] Department of Technology, Shivaji University, Kolhapur.
Email: k_hriday@yahoo.com, chouguleananda@yahoo.co.in
[3] Bharati Vidyapeeth, College of Engg, Kolhapur.
Email: pp_kulkarni@yahoo.com

978-1-4673-2758-9/12/$31.00 ©2012 IEEE

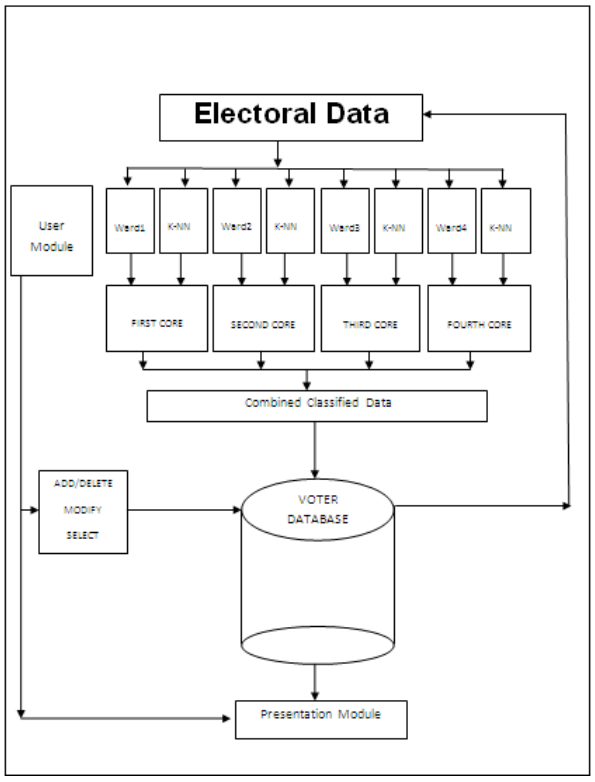

Fig. 1. General architecture of the system

B. User module

This module helps to an administrator for management of the voters data base. The administrator is able to perform various operations on the voter's database such as "Insert a new record", "Update an existing record" or "Delete the record ". These modifications will be again stored back in voter's database. It has a facility to add, Modify, Update data in SQL server. GUI is developed under this module to add, modify, update database. GUI has also a facility to select the training dataset, no. of Cores and output display format.

C. Data mining module

This module implements K-Nearest Neighbor Algorithm for classification of voters data base. In this we have got voter's database which consists of about 250 entries of voters of an area and the training list which consists of 16 sample records which will be used by the algorithm. The algorithm will use the voter's database and the training set for classifying the voters into the class such as young, middle, or old on the basis of "age" attribute of the voter. After the successful execution of the algorithm, a class would be assigned to each and every voter present in the database i.e. the voter either belongs to class young or middle or old depending upon the "age" of the person. After the assignment of the class to the voters, the result will be reflected in the database which will be further used in the presentation module for generating the result.

D. Presentation module

This is the last module of the project where the result of the algorithm execution is presented to the users. After the successful implementation of K-Nearest Neighbor Algorithm in Data Mining module, classes of the voters are stored in the voter's database. Now in this module the result is shown using various visualization techniques such as Pie chart, Bar, Lines etc. This graphical output would be generated from the information of voters' classes stored in the voter's database.

III. Flow Chart of the System

The flow chart of the system is shown in figure 2.

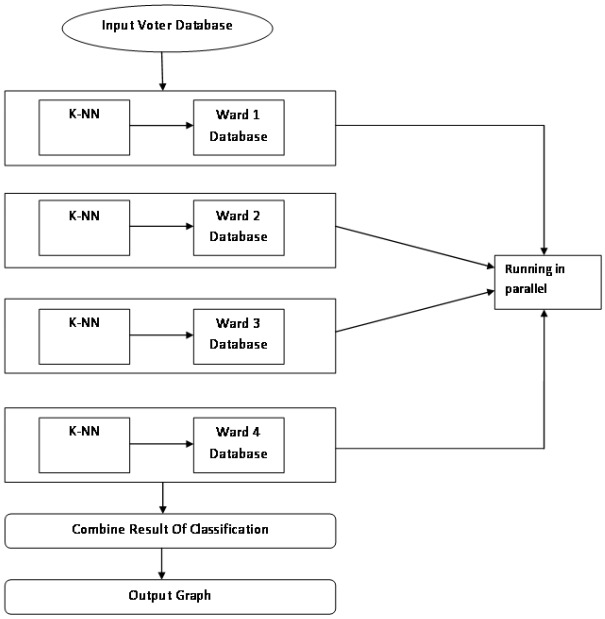

Fig. 2. Flow chart of the system

In this application concurrency is achieved by data decomposition ward wise. Ward wise data is put on each core for task parallelism. The ward wise classified data is combined together for entire voter list of the constituency. The classified data is displayed in pie chart format for understanding of the user.

IV. Experimental Results

This application is developed using C# language with OpenMP environment. The execution is carried out on core 2 duo Intel processor. The presentation module displays the pie chart of result by combing the out put from each core as shown in figure 3.

For performance comparison we executed the application on Intel core 2 duo processor with 2.1 MHz clock frequency and 2 GB RAM. The number of core selected pragmatically where 1, 2, 4 respectively. Result of execution is listed in table 1. It is observed that 30 % less time is required from single core to 2 Core and 22 %

less execution time is observed from 2 Core to 4 Core parallel executions. Overall 55% execution time saving is observed from sequential core to quad core parallel program.

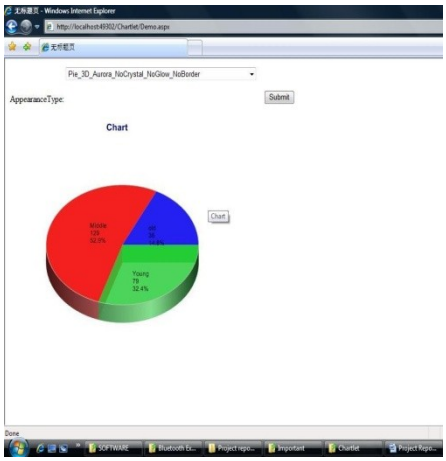

Fig. 3. Age wise Classification result

Table 1. Execution Time in ms with different CPU Core

	Sequential Program	*Parallel Program*
Case-I On Dual Core	13600	9600
Case-2 On Quad Core	12300	7500

The performance of the system with the different core is shown in the figure 4. The graph is plot for execution time required against number of cores. There is considerable reduction in execution time when number of core is increased as expected.

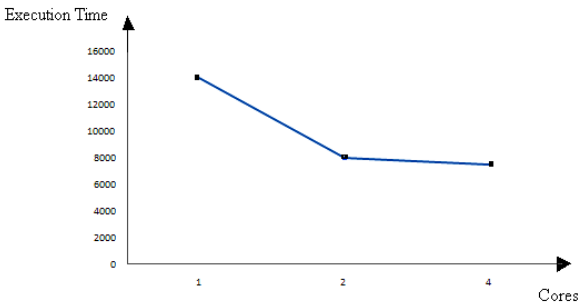

Fig. 4. Execution Performance on number of Cores

V. CONCLUSION

In this paper we have demonstrated the parallel programming on multi core processor. Data mining application like electoral classification shows less execution time when parallel programming is used. Here we applied data parallelism on multi core processor. K-Nearest Neighbor algorithm for data classification execute parallelly on quad processor to classify data of electoral of Kolhapur constituency. Results of all processors are combined to display total electoral classification. We found that with the same data base and same computing platform, the parallel version program will gain better performance than sequential version program. It is clear from the graph, that performance gain come from the data decomposition and the concurrency of the tasks. Another observation is that, with the same computing platform and parallel version program, the program running on quad-core processor based platform will achieve better performance than duo-core platform linearly. Obviously, in a sense, this is a good scale up performance for numbers of cores.

REFERENCES

[1] X. Qiu, G. Fox, H. Yuan, S.-H. Bae, G. Chrysanthakopoulos, and H. Nielsen. " Parallel datamining on multicore clusters", In Seventh International Conference on Grid and CooperativeComputing 2008. GCC '08, pages 41–49, 2008.

[2] Quansheng Kuang, and Lei Zhao, "A Practical GPU Based KNN Algorithm", Proceedings of the Second Symposium International Computer Science and Computational Technology,(ISCSCT '09) Huangshan, P. R China, 26-28,Dec. 2009, pp. 151-15.

[3] Xiaopeng Yu, Xiaogao yu, "The research on an adaptive k-nearest neighbors classifier", Proc.5th IEEE International conference on Cognitive Informatiacs (ICCI 06), pp 535-540.

[4] Honggang Wang, Jide Zhao, Hongguang Li and Jianguo Wang," Parellel Clustering Algorithms for Image Processing on Multi-Core CPU", International Conference on Computer Science and Software Engineering, Hong Kong 2008

[5] Li Xiong, S. Chitti, Ling Liu, "KNearest Neighbor Classification across Multiple Private Databases", ACM - CIKM'06 Arlington, Virginia, USA, November 5–11, 2006.