I J C T A, 9(20), 2016, pp. 9-13

Parallelism in Data Mining Techniques: Challenges and Opportunities

Swati Rustogi*, Manisha Sharma and Sudha Morwal*****

ABSTRACT

Data mining techniques are currently used in many fields like bioinformatics, education, banking sector, insurance sector, etc. Due to rapid increase in size of the data sets, there is a need to introduce parallelism in existing data mining techniques. Most of the data mining techniques are using multiple processors for parallelism. The main drawbacks of using multiprocessors are network communication and increase in cost. Due to advent of multi-core processors, capabilities of such processors can be utilized to execute a data mining technique in parallel in cores. In this paper, we study the current state of parallelism in data mining techniques, advantages of using multi-core environment and challenges in implementing data mining techniques in multi-core environment. Implementing data mining techniques in multi-core environment will be beneficial for

- Researchers – in carrying out their research activities and experiments, using highly available multi-core processors. Unavailability of expensive distributed infrastructure is an impediment in the innovations and discoveries to be made by researchers.
- Industry – in saving costs by not investing in the expensive infrastructure of distributed systems like grids, clouds, etc. and by implementing data mining solutions using multi-core processors.

Keywords: Multi-core, data mining, parallelism

1. INTRODUCTION

Data mining has been defined as "the non-trivial extraction of implicit, previously unknown, and potentially useful information from data" [1]. Data mining techniques extract useful information or pattern from a large data set. In recent years there has been a massive increase in size of the data set. The currently available serialized data mining algorithms are insufficient to process this large data. Serial computing has also become almost obsolete due to

- Diminishing of clock speed [2]
- Increase in the computational power of a single core processor due to multiple cores [3] and multiple processors [4].

To mine patterns from large data in lesser amount of time, parallelism has to be introduced in data mining techniques.

2. TYPES OF PARALLELISM

There are two ways of achieving parallelism:

- By using multiple processors [5], [6], [7], [8], [9]:

  In case of multiple processors, there is a cluster of machines which work together to solve a problem. They are said to work in master and slave mode. There is a cluster of machine and one machine

* Banasthali Vidyapith, Jaipur, India, *Email: swati.rustogi@gmail.com*

** Banasthali Vidyapith, Jaipur, India, *Email: manishasharma8@gmail.com*

*** Banasthali Vidyapith, Jaipur, India, *Email: sudha_morwal@yahoo.co.in*

becomes a master commanding rest of the machines. In case of data mining techniques, the master node divides the large data set among the slave nodes. Every slave node executes the same algorithm on the data set allocated to it. The data should be decomposed in such a manner that there is minimal communication between the slave nodes [10].

- By using a single processor in multi-core environment:

  Computational power of a single processor has been increased rapidly by use of multiple cores in a single processor. In the coming years, the numbers of cores used in a processor are going to increase [11]. Dual core chip executes an algorithm 1.5 times faster in comparison single core chip [12]. Execution of a serial algorithm in such an environment will require modifications in the algorithm.

3. NEED OF MULTI-CORE

Multi-core systems are replacements for multiprocessors, as at times expensive distributed systems like supercomputers are not available [13]. In case of multiprocessors, master slave parallelism is heavily used [14]. Speedup of this method is limited due to hardware requirements and memory requirements of each node and data distribution and communication between nodes [14].

By using multi-core processors, high performance software can be generated. They provide greater advantage in comparison to multiprocessor system, as they provide multithreading and parallelism on a massive scale [15]. Multi-core processors can be used as a substitute for the hardware like large distributed systems which are costly and have high maintenance cost [13].

4. CURRENT STATE OF PARALLELISM IN DATA MINING

- Algorithm proposed in [5], reduces complexity of serial K means by using data parallelism. It uses master slave approach and it is based on message passing model. Master partitions the data in K sets and then sends each subset to one of the K slaves. Slaves compute the mean and send this mean to the other slaves. New K subsets are formed on the basis of their distance to the mean. This process is repeated till the result does not become stable.
- Authors in [16] proposed an algorithm in which parallel implementation of K-means is done using GPU and CUDA model. Number of threads in an application is same as number of cores. In case of CUDA model since there is no overhead of creating threads, so number of threads to be created is independent of number of cores. In K means algorithm, calculating distance between data and centroid is most time consuming. So author has proposed number of threads to be equal to number of data points and then each thread computes distance between data point and centroid. This improves the efficiency of parallel K means algorithm.
- Algorithm proposed by author in [6] parallelizes K-means algorithm message passing and master and slave technique. Master creates slave processes and then sends the partitioned data set and list of centroids to each slave. Each slave then performs cluster assignment of each data point by calculating its distance from each centroid. This cluster assignment is forwarded to master. Master then recalculates the centroid. This process is continued till there are no changes in centroid values.
- Author in [17] modified K means algorithm from the point of view of Bayesian nonparametric model. K-means algorithm is a simple and easy to implement algorithm, while Bayesian nonparametric models are used for infinite mixtures and number of clusters need not be fixed. Author presents a hard clustering algorithm which is same as K means, except that new cluster is formed when point is farther away than a variable gamma, which means that it is away from all existing cluster.

- Author in [18], proposed optimistic concurrency approach which can be used for solving the concurrency issues, which occur due to parallelization of Bayesian nonparametric models. Optimistic concurrency control does not require locks and it is used in places where chances of conflicts are rare.
- Author in [7] parallelizes Stochastic Gradient Descent algorithm. Author proposes a parallel algorithm which does not uses locks. Processors access shared memory and they can update the memory areas without using locks.
- Author in [8] proposed parallel algorithm for association rule mining, which is executed in grid computing environment. Algorithm uses hybrid parallelism as it does both data and task parallelism. Algorithm also uses work load balancing to allocate jobs to each process.
- Author in [19], proposed middleware for parallelizing Apriori association mining algorithm and K-nearest neighbor algorithm, Author has proposed techniques like full locking, partial locking, complete replication and partial replication for concurrency. Author in [20] used map reduce for Apriori algorithm and performed parallelization using multiple cores.
- Author in [21] proposed an algorithm which selects centers for K-means in more than one iteration, such that in a single iteration multiple centers are selected. This algorithm is an improvement over K means algorithm which uses single iteration to select K centers and K-means++ algorithm [22], which selects only a single center in a single iteration.
- Author in [23] proposed an algorithm which uses concept of mini-batch to make the existing K-means algorithm more scalable. Mini-batches are the samples from the data set.

Not much effort has been done in executing data mining algorithms in multi-core environment. Moreover,

- Higher cost is involved in setting up a networked system
- There is a delay in total execution time in case of networked environment as the data has to be divided among the multiple processors and the final result has to be collated from all the processors.
- Expensive infrastructure like grids, super computers are not accessible to everyone.
- Multi-core technology is available to all the users.

5. HOW TO ACHIEVE PARALLELISM IN MULTI-CORE ENVIRONMENT

Multiple threads should be used in multi-core environment to take advantage of multiple processing units. These threads can communicate with each other easily as they share memory [24].

For achieving parallelism in multi-core processor, authors in [25] mentioned the current solutions available for implementing parallelism in multi-core. OpenMP is a programming approach used for shared memory. Threads update the shared memory so that the updated data can be read by other threads. In case of shared memory, programmer needs to take care of memory conflicts. MPI is the programming approach, in which threads use message passing to communicate.

6. CHALLENGES

Not much effort has been done in the area of implementation of data mining techniques in multi-core environment. Following issues needs to be addressed [26]:

1. Select appropriate paradigm for parallel programming [27]: Task parallelism or SPMD (Single program multiple data) parallelism.
2. Selecting suitable programming constructs [28]: Message based programming, Scatter and gather based programming or Event based programming.

3. Selecting suitable APIs for implementing the programming constructs [29]: MPI or OpenMP.
4. Which approach to use to make data sharable between multiple cores [30]: Declaring data as shared data. Data can be accessed by multiple processors at a given time by making the data as sharable data or data that can be shared at execution time with some input data.
5. Select the mechanism for resolving conflicts due to concurrent access to the shared data [31]: Optimistic approach or Pessimistic approach

Table 1
Challenges in Parallelism of Data Mining Techniques

Challenges	*Solutions*	
	Alternative 1	*Alternative 2*
Paradigm for parallel programming	Task parallelism	SPMD parallelism
Programming constructs	Message based programming	Event based programming
APIs	MPI	OpenMP
Data sharing among cores	Sharing of data	Send and receive data
Conflict resolution	Optimistic approach	Pessimistic approach

7. CONCLUSION

For achieving parallelism in data mining techniques multi –core environment should be used. It will be beneficial, especially to those researchers and industry experts, who do not have access to large distributed environments like clusters, grids, etc. While using multi-core environments there are certain challenges which needs to be looked into.

First of all, it should be decided whether data parallelism or task parallelism will be used. Then the segment of code has to be identified, in which parallelism using threads will be applied.

Then communication mode between threads should be decided, whether to use shared memory or message passing. A parallel algorithm in which issues like load balancing between cores, data decomposition, concurrent access and proper decomposition of data are resolved, will be more efficient and useful to researchers to mine vast data, without investing in costly infrastructure.

REFERENCES

[1] W.J. Frawley, G. Piatetsky-Shapiro and C.J. Matheus, “Knowledge discovery in databases: An overview”,AI magazine 13, no. 3, pp.57, 1992.

[2] J. Parkhurst, J. Darringer and B. Grundmann ,” From single core to multi-core: preparing for a new exponential”, in Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design,ACM, pp. 67-72, 2006.

[3] W Feng and P Balaji , P., “Tools and environments for multicore and many-core architectures”, IEEE Computer 42, No. 12, pp. 26-27, 2009.

[4] I. S. Dhillon and D.S. Modha , “A data-clustering algorithm on distributed memory multiprocessors”, Large-Scale Parallel Data Mining, Springer , pp. 245-260, 2000.

[5] S. Kantabutra and A.L. Couch, “Parallel K-means clustering algorithm on NOWs”, NECTEC Technical journal 1, no. 6, pp. 243-247, 2000.

[6] K. Kerdprasop and N.Kerdprasop, “Concurrent Data Mining and Genetic Computing Implemented with Erlang Language”, International Journal of Software Engineering and Its Applications 7, no. 3, pp. 63-76, 2013.

[7] B. Recht, C. Re, S. Wright, and F. Niu, “ Hogwild: A lock-free approach to parallelizing stochastic gradient descent”, Advances in Neural Information Processing Systems, pp. 693-701, 2011.

[8] R. Tlili, and Y. Slimani, “A hierarchical dynamic load balancing strategy for distributed data mining”, International Journal of Advanced Science and Technology, Volume 39, pp. 21-48, 2012.

[9] D. Foti, D. Lipari, C. Pizzuti and D. Talia, “Scalable parallel clustering for data mining on multicomputers”, In *International Parallel and Distributed Processing Symposium*, pp. 390-398, 2000.

[10] J. Li, Y. Liu, W. K. Liao and A. Choudhary, "Parallel data mining algorithms for association rules and clustering", In International Conference on Management of Data, 2008

[11] R. Oshana, D. Stewart and M. Domeika, "Introduction to multicore programming guide", 2013.

[12] D. Geer, "Chip makers turn to multicore processors", Computer, Voulme 38, Issue 5, pp. 11-13, 2005.

[13] J. Li, W. Guo and H. Zheng, "An Undergraduate Parallel and Distributed Computing Course in Multi-Core Era", In 9th International Conference for Young Computer Scientists, pp. 2412-2416, 2008.

[14] J.M. Kraus and H.A. Kestler, "A highly efficient multi-core algorithm for clustering extremely large datasets", BMC bioinformatics, 11(1), pp. 169, 2010.

[15] P. Gepner and M.F. Kowalik, "Multi-core processors: New way to achieve high system performance", In International Symposium on Parallel Computing in Electrical Engineering, pp. 9-13, 2006

[16] R. Farivar, D. Rebolledo, E. Chan and R.H. Campbell , "A Parallel Implementation of K-Means Clustering on GPUs", In PDPTA, Vol. 13, No. 2, pp. 212-312, 2008.

[17] B. Kulis and M.I. Jordan, "Revisiting k-means: New algorithms via Bayesian nonparametrics",arXiv preprint arXiv:1111.0352, 2011.

[18] X. Pan, J.E. Gonzalez, S. Jegelka, T. Broderick and M.I. Jordan, "Optimistic concurrency control for distributed unsupervised learning", In Advances in Neural Information Processing Systems, pp.1403-1411, 2013.

[19] R. Jin and G. Agrawal, "A middleware for developing parallel data mining implementations", In Proceedings of the first SIAM conference on Data Mining, 2001.

[20] N. Li, L. Zeng, Q. He and Z. Shi, "Parallel implementation of apriori algorithm based on MapReduce", In 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), pp. 236-241, 2012.

[21] B. Bahmani, B. Moseley, A. Vattani, R. Kumar and S. Vassilvitskii, "Scalable k-means++", In Proceedings of the VLDB Endowment, Volume 5 Issue 7, pp. 622-633, 2012.

[22] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding", In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms , pp. 1027- 1035, 2007.

[23] D. Sculley, "Web-scale k-means clustering", In Proceedings of the 19th international conference on World wide web, pp. 1177-1178, 2010.

[24] V. Packirisamy and H. Barathvajasankar, "Openmp in multicore architectures", University of Minnesota, Tech. Rep, 2005.

[25] J. Y. Mignolet, R.Baert, T.J. Ashby, P. Avasare, H.O. Jang and J. C. Son,." Mpa: Parallelizing an application onto a multicore platform made easy".IEEE micro, volume 3, no. 29, pp. 31-39, 2009.

[26] D. Patterson, "The trouble with multi-core", Spectrum, IEEE, vlume 47, issue 7, pp. 28-32, 2010.

[27] A. Sbîrlea, K. Agrawal and V. Sarkar, "Elastic Tasks: Unifying Task Parallelism and SPMD Parallelism with an Adaptive Runtime", In Euro-Par 2015: Parallel Processing, pp. 491-503, 2015.

[28] D. W. Holmes, J. R. Williams and P. Tilke, "An events based algorithm for distributing concurrent tasks on multi-core architectures", Computer Physics Communications, volume 181, issue 2, pp. 341-354, 2010.

[29] G. Krawezik, "Performance comparison of MPI and three OpenMP programming styles on shared memory multiprocessors", In Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures, pp. 118-127, 2003.

[30] J. P. Singh, W. D. Weber and A. Gupta, "SPLASH: Stanford parallel applications for shared-memory", ACM SIGARCH Computer Architecture News, volume 20, no. 1, pp. 5-44, 1992

[31] D.R. Selvarani and T.N. Ravi, " A survey on data and transaction management in mobile databases", arXiv preprint arXiv:1211.5418, 2012.