International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 04 | April-2016 www.irjet.net p-ISSN: 2395-0072

MACH: Performance Enhancement in Multi-core Processor using Apriori Algorithm with file Chunking

Prantik Pancholi[1], Shital Khairnar[2], Jyoti kamble[3] , Amol Jadhao[3]

[1] *B.E Student, Computer Engineering, Dr.D.Y.Patil Institute of engineering and technology, Maharashtra, India*
[2] *B.E Student, Computer Engineering, Dr.D.Y.Patil Institute of engineering and technology, Maharashtra, India*
[3] *B.E Student, Computer Engineering, Dr.D.Y.Patil Institute of engineering and technology, Maharashtra, India*
[4] *M.E.Project Guide, Computer Engineering, Dr.D.Y.Patil Institute of engineering and technology, Maharashtra, India*

Abstract - *Apriori Algorithms are used on very huge amount of data sets with high dimensionality. This paper presents the performance of serial mining and parallel mining with the help of Apriori algorithm also we are used load balancing concept and BSW algorithms in this paper. The size of database is in GB and TB. It needs a fast processor. Multi-core processor is used for parallelism. Parallelism is used to reduce time, increase performance and fast processing. Serial mining can takes more time for mining process. To solve this issue we are proposing a parallel processing in which load balancing is done among processors. In proposed system it implements Apriori algorithm in serial and parallel manner also we show the comparison between serial and parallel mining on the basis of varying support-count and time using parallel programming technique. Garbage collector is used to free the memory space. This paper presents a load balancing technique designed specifically for parallel publications applications running on multicore applications. This architecture provides a hardware parallelism through cores inside the CPU. It increased performance low cost as compare to single core machines attracts HPC high performance computing connectivity.*

Key Words: *Parallel data mining, Serial data mining, frequent item set, Association rules, Apriori algorithm, BSW Algorithms.*

1. INTRODUCTION

Accumulation of large data from different sources of the society but a little knowledge situation has lead to knowledge discovery from large databases which is known as data mining. The Data mining techniques are use the existing data and retrieve the useful information from it which is not directly visible in the original data. As data mining algorithms deal with large data, the primary goals are how to store the data in the main memory at run time and how to enhance the run time performance. Sequential algorithms cannot support the scalability, in terms of the size, data dimension or runtime performance for such large databases. Because the size of the data are increasing to a large quantity, high-performance parallel and distributed computing used to get the advantage of more than one processor to handle these quantities of data. Also sequential algorithms take more and more time for scanning frequent item set because it works sequentially. Association Rule Mining or frequent itemset mining is an important functionality of data mining. The Apriori algorithm helps to finding frequent item sets from a large transaction database. As data mining mainly deals with the large volumes of data, the main concern is how to improve the performance of the algorithm. Modern Java applications employ multithreading to increase the performance by harnessing execution parallelism available in today's multi core processors. Parallelism achieves great performance .However, as the numbers of threads and processing cores are scaled up, many applications do not achieve the desired level of performance improvement. Multi-core system works parallely means it executes more tasks at same time. Multi-core Architecture support hardware parallelism through cores inside the CPU. It increases the performance and low cost as compared to single-core machines Association rule is the technique to determine the frequent Item set and Candidate generation in data mining

2. Objective

Take the advantage of newly launch multi-core system for the implementation of parallel Apriori algorithm. Performance Enhancement of array based Apriori algorithm using BSW Algorithm. Parallelism is a technique to achieve more and more speed for the processing. With this functionality, as specified in system architecture master node will also interact with another distributed system and will Parallely execute the mining operations on those systems. Utilize the available resources for execution. Achieve peak performance of the processor through parallelism. Generate high level throughputs using HPC environment.

2.1 Related Work

Jaehong Min, Youjip Won and Kyeongyeol Lim posed multithread variable size chunking mechanism on

multicore architecture system.[10] MUCH partitions a file into small segment and Master thread distributes them to chunker thread. Ke Zhang, Yi Chai, Yi Li posed In Order to Avoid Multiple Times Data Base Scanning Migrate Candidate generation and Frequent Item set Together[12 15]. S. N. Tirumala Rao, E. V. Prasad, Jntuk posed A Critical Performance Study of Memory Mapping on Multi-Core Processors. [2, 3] Sheila A. Abaya posed Association Rule Mining based on Apriori Algorithm in Minimizing Candidate Generation. This Algorithm is not efficient and effective so it needs to use modified Apriori Algorithm. Agrawa and other scholars analyzed the basket firstly in 1993 and given the association rules mining in data mining. Furthermore, the classical Apriori Algorithm was been proposed in the following year [6]. Subsequently, many researchers studied the problem of mining association rules in algorithm optimization. Savasere et al proposed a method based on Partition [7]. Park et al proposed a method based on Hash [8]. Mannila and Toivonen proposed a method on Sampling [9, 10] based.

3. Existing System

3.1 Core

The smaller and independent processing units of central Processing Units (CPU) are called cores. A core is the processing unit which receives instructions and performs actions based on those instructions. Processor can have single core or multiple cores.

3.2 Single-Core

A chip with one CPU is called as single core .In single core system only one core running at the same time this is the sequential execution system all threads are running sequentially. It's great for web surfing, checking e-mail, word processing, etc. but it's usually not suitable for frequent items calculations. In data mining the task of finding frequent pattern in large database is very important. In single core architecture find frequent item sets or candidate generation by using single thread.

3.3 Single-Core Architecture

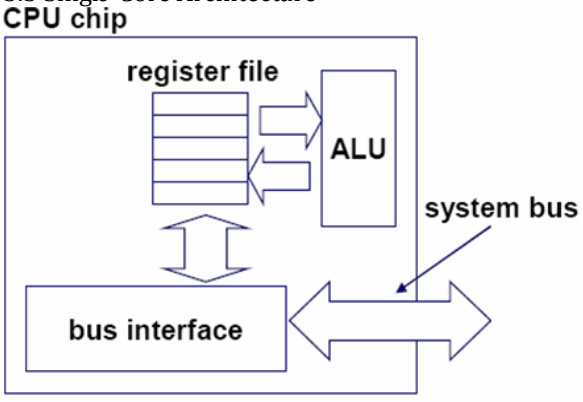

Fig 1: Single core architecture

4. Proposed Framework

4.1 Multi-core architecture

The term multi-core used to describe two or more CPU working together on the same chip. A processor that uses more than one core is called multi-core processor. Typically, a processor that uses two cores is called dual-core processor. Parallel computing is a type of computation in which many calculations are carried out simultaneously operating on the principle that large problems can often divided into smaller ones, which are then solved at the same time .Multi-core system support the parallel computing called as" parallelism". If the number of threads are less than or equal to the number of cores, separate core is allocated to each thread and threads run independently on multiple cores. If the number of threads is more than the number of cores, the cores are shared among the threads. In multi core architecture find frequent item sets or candidate generation by using multiple threads.

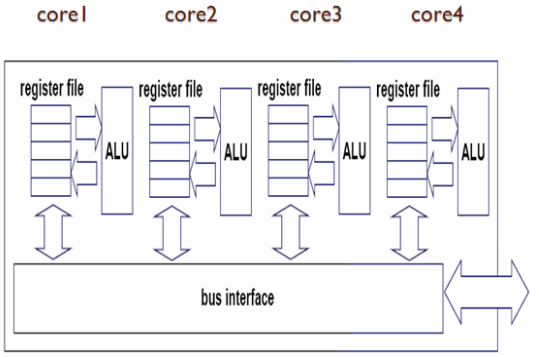

Fig 2: Multiple core architecture

In figure (3) Multiple thread execute on multiple core in parallel. Each core executes independent threads. Load can no longer be considered symmetric across the cores. Source code will often be unavailable, preventing compilation against the specific hardware configuration.

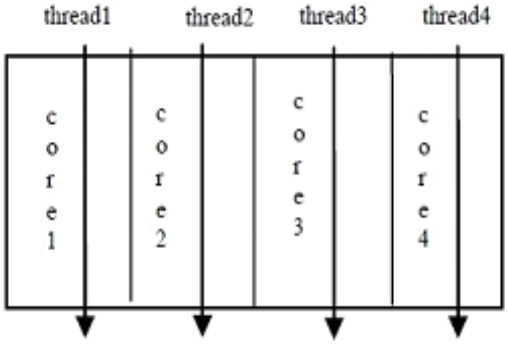

Fig 3: Independent Threads on each Core

4.2 Advantages of Multi-core

Processing speed: As core behaves as a processing unit, the use of multiple cores has enhanced processing speed.
Clocking optimization: Due to the use of multiple cores, the

CPUs are capable of processing more data with same clock frequency as that of single core processors.
Multithreading: Multi-core processors have increased the easy of implementing multithreading with single core.
High Performance
Parallel processing: As every core executes the instructions as an individual processing unit, parallel processing is achieved.
Cache Sharing: Multi-core processors share the cache memory of CPU hence reducing separate use of cache for every core.

5. Mathematical module

Support
I = {i1, i2, i3,....., im} is a collection of items.
T is a collection of transactions with the items.
TID is a transaction identifier TID.
Support 'S' is proportion of transactions in the data set which contains the Itemset.
Support(X=>Y) = Support (XUY) = P (XUY).
Association rule
A=>B is such that,
A Є I, B Є I.
A is called as Premise
B is called as Conclusion
Confidence
It is defined as a conditional probability
Confidence (X=>Y) = Support (XUY) / Support(X) = P(Y/X).
Support Count Refer as a Number of Transaction that contains a particular Item set.
Support Count = ϭ(x) for item set "x"

$$\text{ϭ(x)} = \{t_i | x \subseteq t_i, t_i \in T\} \quad \text{.....(1)}$$

Whereas,
T → Total Transaction
t_i → Current Transaction
x → Item set
ϭ(x) → Support Count
Support Determine how often a rule to a given data set.

$$\text{Support } \{S (x \rightarrow y)\} = \frac{\text{ϭ(xUy)}}{N} \quad \text{.....(2)}$$

Confidence determine how frequently Item in Y appear in transaction that contain x

$$\text{Confidence } \{C (x \rightarrow y)\} = \frac{\text{ϭ(xUy)}}{\text{ϭ(x)}} \quad \text{.....(3)}$$

Total Count Time

$$T^{k}_{Apriori-Tid} = \sum_{t=t_1}^{t_{N_{T_{k-1}}}} \sum_{c=c_1}^{c_{N_{c_k}}} fun(N_t, N_c)$$

..... (4)

Whereas,

NT_{k-1}= no. of transaction used in k^{th} round

N_{ck} = no of K dimensional candidate item set

Nt = no of item in transaction t

Nc = no of item in candidate item c

fun (Nt, Nc) = Gives count time of determining whether 't' support 'c" or not.

6. Apriori Algorithm

Apriori algorithms are mostly used in data mining technique. This algorithm is used to find frequent item set and calculate Candidate generation.
Apriori Algorithm working on two step process.
1. Prune step
2. Join step
Example: In this example a database has a four transaction. Let the minimum support =2 transaction. As it shows the transaction in Table 1
Table 1: Database

Transaction Id	Item sets
100	1, 3, 4
200	2 ,3, 5
300	1 ,2, 3, 5
400	2 ,5

Table 2: frequent item set

Set of item sets	Transaction Id
1	{100, 300}
2	{200, 300, 400}
3	{100, 200, 300}
4	{100}
5	{200, 300, 400}

Table 3: Support count

Itemset	Support
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

Table 4: L_1

Itemset	Support
{1}	2
{2}	3
{3}	3
{5}	3

Table 5: L_2

Items	Support
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

Table 6: L_3 Frequent item sent

Itemset	Support
{2 3 5}	2

7. INPUT-OUTPUT PROCESSING

In figure (4) we show the working of our system by tacking example also we have to observer that which platform is better for frequent item calculation. In this figure user first login the mining system. System required password which is generated for security purpose. There are two platforms available serial mining and parallel mining. User selects the database file and gives the mining system. After selection of 5database file mining calculate the frequent item set and show the analysis the system. There is one difference in serial and parallel mining, parallel mining works fast scanning for large database file. It reduces the execution time.

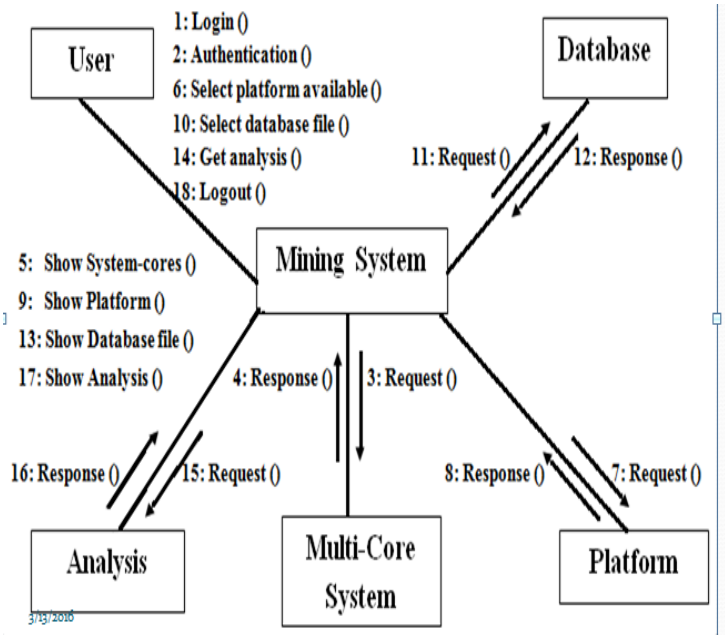

Fig 4: Input output processing system

8. Module

8.1.1 User Interaction Module

This module provides various interfaces for accessing system features as per user activity. User interaction module handles the all functions of user also this module specially designed for interaction between user and system.

8.1.2 Authentication Module

Authentication- Establishing the authenticity of a person or other entity. This module designed for security purpose which generates unique username and password. After enter the login mining systems check the username and password if correct it gives the response to the multicore system. Also user can change the password by using old password. The following frames show the how to authenticate the user.

8.1.3 Database construction

In database constriction we are taking database file from the user side. In database is nothing but total transactions and Item set.

7.1.4 Serial Mining module

Mining system provides two platforms to the user serial mining and parallel mining. In serial mining the items are calculating one by one. In serial mining item set convert into binary format using format generator. Serial mining module executes candidate generation and frequent Item mining in sequence manner. Serial mining give their response after long time. This is the main disadvantages of serial mining.

8.1.5 Parallel Mining Module

There are two options available to the user if he wants to select the parallel mining, mining system send request to the platform. Platform give the response and system show the platform which is requested. User selects the database file and calculates the items on parallel platform. After calculating the frequent items set on two platforms system show the analysis.

8.1.6 Performance Analyzer

One of the main concepts of paper is comparison between serial and parallel mining. How they work on different processor also we check on different processor and we have observe that the performance of serial mining and parallel mining. We perform on windows 7, 64bit, Intel® Core™ i3-2330M CPU, RAM 3 GB.

8. Experiment and Result Analysis

We are used to 5 to 6 system for test or time comparison between serial mining parallel mining. The average results for both the execution time and the CPU usage are different after checking on different system. One of the examples of system is explained below that we perform our project.
System Details:-
Processor: - Intel® Core™ i3-2330M CPU
Speed: - 2.20 GHz
RAM: - 3GB
O.S/ Java version: - 64 Bit
In table database is 10 20 transaction. That mean we have a 10 item set or 20 transactions. Other one is 20 25 transactions that mean we have 20 item set or 25 transactions. We are compare time both in serial mining and parallel mining. In Table (7) minimum support is 30% .

Table 7: Time comparison between serial and parallel mining with minimum support 30

	Serial mining		Parallel mining	
Tns. Item	Freq_Item	Time	Freq_Item	Time
10 20	119	109 m/s	119	78 m/s
20 25	2003	2043 m/s	2003	223 m/s
20 100	21151	197596 m/s	21151	2031 m/s

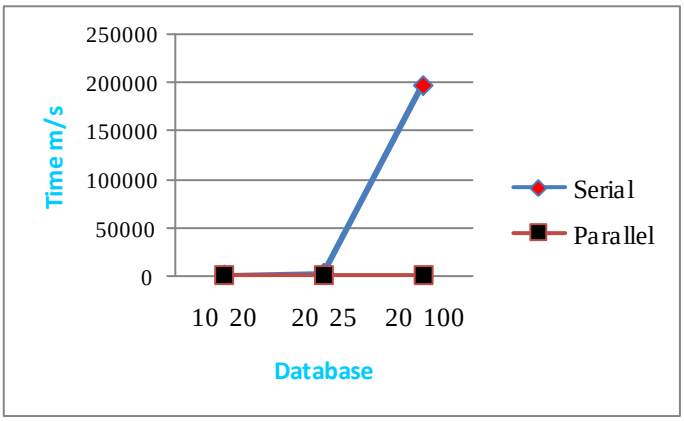

Fig 6: Time comparison between serial and parallel mining with minimum support 30

We are taking same database with different minimum support, minimum support is 45

Table 8: Time comparison between serial and parallel mining with minimum support 45

	Serial mining		Parallel	
Tns. Item	Freq_Item	Time	Freq_Item	Time
10 20	70	63 m/s	70	78 m/s
20 25	372	243 m/s	372	94 m/s
20 100	3689	6324 m/s	3689	449 m/s

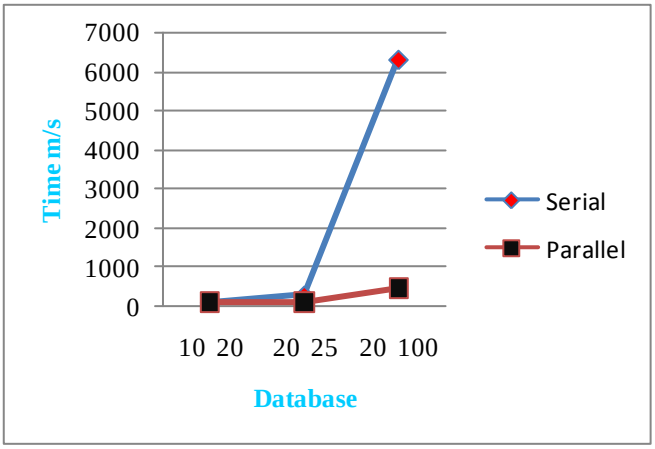

Fig 7: Time comparison between serial and parallel mining with minimum support 45

9. CONCLUSIONS

In this work we overcome the drawbacks of sequential algorithms which is ineffective for frequent items calculation. A parallel mining system effectively work on frequent items calculation it reduce the scanning time and also it gives the better performance as compare to sequential algorithm. In experimental result show that proposed module is feasible for large database file and all the processes finish within less time. We have to test our proposed framework on different operating system and we notify that our system is successfully implemented as per the our goals and it works better for frequent items calculations as compare to current system. Improvement of our system we can use graphics processor in future. We can also distribute mining processing load in network, In future we will also focus on Security mechanism.

10. References

[1] Khadidja Belbachir, Hafida Belbachir, "The Parallelization of Algorithm Based on Partition Principle for Association Rules Discovery", In Proceedings of International Conference on Multimedia Computing and Systems(ICMCS), IEEE, May 2012.

[2] Ruowu Zhong, Huiping Wang, "Research of Commonly Used Association Rules Mining Algorithm in Data Mining", In Proceedings of International Conference on Internet Computing and Information Services(ICICIS), IEEE, September 2011.

[3] V.Umarani, Dr.M.Punithavalli, "A Study On Effective Mining Of Association Rules From Huge Databases", International Journal of Computer Science and Research (IJCR), Vol. 1 Issue 1, 2010.

[4] Xindong Wu , Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang ,Hiroshi Motoda, "Top 10 Algorithms in Data Mining", Knowledge and Information Systems, Volume 14, Issue 1, pp 1-37, Springer, January 2008.

[5] Eui-Hong (Sam) Han, George Karypis, Vipin Kumar, "Scalable Parallel Data Mining for Association Rules", IEEE Transactions on Knowledge and Data Engineering, Volume:12 , Issue: 3, May/June 2000.

[6] Mohammed J. Zaki, "Parallel and Distributed Association Mining: A Survey", IEEE Concurrency, Vol 7, Issue 4, pp 14-25, October 1999.

[7] Mohammed J. Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, Wei Li, "Parallel Algorithms for Discovery of Association Rules", Data Mining and Knowledge Discovery, Vol 1, Issue 4, pp 343-373, Springer, December 1997.

[8] Mohammed J. Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, Wei Li, "A Localized Algorithm for Parallel Association Mining", Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures, ACM 1997.

[9] Rakesh Agrawal, John C. Shafer, "Parallel Mining of Association Rules", IEEE Transactions on Knowledge and Data Engineering, December 1996.

[10] Youjip Won, Kyeongyeol Lim, and Jaehong Min," MUCH: Multithreaded Content-Based File Chunking" IEEE TRANSACTIONS ON COMPUTERS, VOL. 64, NO. 5, MAY 2015.

[11] F. Douglis and A. Iyengar, "Application-specific delta-encoding via resemblance detection," presented at the USENIX, San Antonio,TX, USA, Jun. 2003.

[12] S. Hofmeyr, C. Iancu and F. Blagojevic, "Load balancing on speed," ACM Symposium on Principles and Practice of Parallel Programming, Bangalore, India, Jan 2010, pp. 147-158.

[13] C. Liao, Z. Liu, L. Huang, , and B. Chapman. Evaluating OpenMP on Chip Multithreading Platforms. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008.

[14] Rui Chang; Zhiyi Liu; , "An improved apriori algorithm," Electronics and Optoelectronics (ICEOE), 2011 International Conference on, vol.1, no., pp.V1-476-V1-478, 29-31 July 2011

[15] Du Ping; Gao Yongping; , "A new improvement of Apriori Algorithm for mining association rules," Computer Application andSystem Modeling (ICCASM), 2010 International Conference on , vol.2, no., pp.V2-529- V2-532, 22-24 Oct. 2010.

Jyoti kamble currently studying in BE (Final Year) Computer Engineering in DYPIET, Pune University. Completed Diploma in computer engineering in 2013. She currently works on data mining. She interesting in hacking.

Prantik Pancholi studying BE (FY) in DYPIET, Pune University. He has completed 12th in Rajasthan. His interesting topic is data mining.

Shital Khairnar studying BE(FY) in DYPIET, Pune University. She completed Diploma in Computer Engineering. She interesting in Data mining also hardware related topics include.

Amol Jadhao (ME) project Guide Sir. He received the degree. His interests in also data mining. His knowledge is very good about android.