INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 3, Issue 6)
Available online at www.ijariit.com

Performance Evaluation of Multi-Core System through Mining Techniques

Ramdas Popat Jare
Dept. of Computer Engineering,
Dr. D. Y. Patil College of Engineering, Ambi,
Maharashtra
ramdas.jare55@gmail.com

Vrushali Desale
Dept. of Computer Engineering,
Dr. D. Y. Patil College of Engineering, Ambi,
Maharashtra
vrushalidesale@gmail.com

Abstract:* Apriori algorithm is a masterstroke algorithm of association rule mining. Increased possibility of the Multicore processors are impose us to upgrade the algorithm and applications so as to accomplishment the computational power from multiple cores finding frequent item sets is more upscale in terms of computing resources utilization and CPU power. Apriori Algorithm is used on very big data sets with high dimensionality of data. Therefore, parallel computing can be applied for mining using association rules. The process of association rule mining consists of finding frequent item sets and generating rules from the frequent item data sets. Finding frequent item sets are more expensive in terms of CPU power consumption and computing resources utilization. Thus, majority of parallel apriori algorithm focus on parallelizing the process of discovering frequent item set. The computation of frequent item sets mainly consist of creating the candidate's generation and counting items. In parallel frequent item sets mining algorithms addresses the issue of distributing the candidates among processors such that their creation and counting is effectively parallelized. Paper presents comparative study of the serial and parallel mining of data sets.

The Raspberry Pi is powerful, small computer having the dimensions of credit card which is invented with the hope of inspiring generation of learners to be creative. This computer uses ARM (Advanced RISC Machines) processor, the processor at the heart of the Raspberry Pi system is a Broadcom BCM2835 system on chip multimedia processor. Paper provides a description of the raspberry pi technology which is a very powerful computer. Also it introduces the overall system architecture and the design of hardware components are presented in details. We also uses Raspberry Pi 3 Model B 1.2 GHz 64-bit quad-core ARM Cortex-A53 and find the results using serial and parallel mining of data sets [14].

***Keywords:* Raspberry Pi, Parallel Data Mining, Frequent Itemsets, Association Rules, Apriori Algorithm.**

I. INTRODUCTION

Apriori algorithm is the classic algorithm of association rules, which finds the entire frequent item sets. When this algorithm is applied to dense data the algorithm performance declines due to the large number of long patterns emerge. In order to find more valuable rules, paper proposes an improved algorithm of association rules, the classical Apriori algorithm and mining of data using Raspberry Pi. Finally, the improved algorithm is verified, the results show that the improved algorithm is reasonable and effective, can extract more valuable information. Data mining techniques is one of the most dynamic emerging researches in today's database technology. To extract the valuable data from large databases, it is necessary to explore the databases efficiently. It is the analysis step of the Knowledge Discovery and Data Mining process. It is defined as the process of extracting Interesting non-trivial, implicit, previously unknown and useful information or patterns from large information repositories such as: relational database, data warehouses etc. The goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. The problem of mining association rules from transactional database was introduced in. The concept aims to find frequent patterns, interesting correlations, and associations among sets of items in the transaction databases or other data being used widely in various areas such as telecommunication networks, drug analysis, and risk and market management, inventory control.

Association rule are the statements that find the relationship between data in any database. Association rule has two parts "Antecedent" and "Consequent". For example {bread} => {butter}. Here bread is the antecedent and butter is the consequent. Antecedent is that item which is found in the database, and consequent is the item that is found in combination with the first i.e. the antecedent.

Formal definition: Let I = {i1,i2, …,in} be a set of items.
Let D be a set of task relevant data transactions where each transaction T is a set of items such that $T \subseteq I$. A unique TID is associated with each transaction.
Let A be a set of items. A transaction T is said to contain A if and only if $A \subseteq T$. An association rule is implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, and $A \cap B$ = null. The ARM technique is based on several threshold values which are explained below:

Minimum Support: Is defined as the percentage of records that contain XUY to the total number of records in the database. It is given as: Support (XUY) = Support (XY)/Total number of transactions.

Confidence: Is defined as percentage of transaction in database containing X that also contains Y. It is given as:
Confidence = Support (XUY)/Support(X).

Lift: Is the ratio of the probability that X and Y occur together to the multiple of the two individual probabilities for X and Y. It is given as: lift = Pr(X, Y) / Pr(X).Pr(Y).

Conviction: Unlike lift, it measures the effect of the right-hand-side not being true. It inverts the ratio and is given as: conviction = Pr(X).Pr(not Y) / Pr(X,Y).

Problem Statement

To generate frequent item set using serial and parallel approach, reduce the memory space using garbage collector and also check performance on the basis of i3,i5,i7, raspberry pi. Parallelism is used to reduce time and increase performance, Multicore processor and raspberry pi commuter is used for parallelizing. Serial mining is consume time and reduce performance for mining. Apriori algorithm is implemented in serial and parallel manner and comparison of both on the basis of varying support count and time using parallel programming technique.

Limitation of Existing System

- Time Consuming Mining Procedure.
- Power consuming methodology.
- Insufficient resource utilization.
- Unutilization of multicore system.
- Single point load execution strategy.

Proposed System

Our objectives are measuring the serial and parallel performance of Apriori algorithm on a different core processor like Intel, AMD with respect to time and comparison between them. For our work we will mainly use different operating system like Windows, Linux with different datasets from Frequent Item set Mining Repository. We will also focus on the variation in different support counts from 30% to 70% for measuring performance.

Goals and Objectives

- To compare apriori algorithm using serial and parallel approach.
- maximize the utilization of CPU cores.
- To free the memory space using garbage collector.

System Design

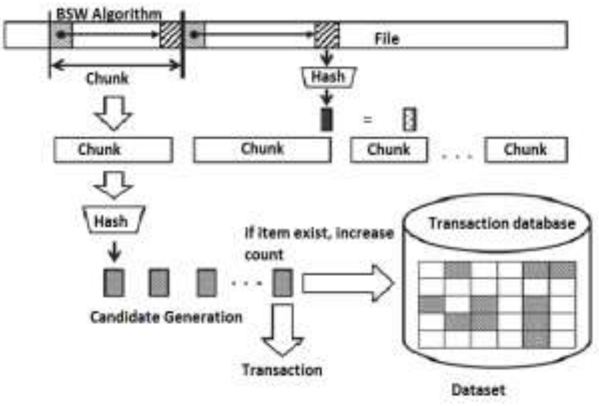


Fig.1: Architecture for Parallel Frequent Mining Using BSW

Above diagram is the architecture of our proposed system .In this architecture we focus on fixed size chunking and variable size chunking. In this diagram a large file is broken in to small chunks. Transaction database is shown in diagram, in that no of transaction files are stored. From that one file is selected and that file is break into small parts, that parts are called as chunks. As shown in architecture one file is selected and that files are break into small chunks. For that purpose we are going to use Basic sliding window algorithm. Apriori algorithm is used for fetching the data from dataset. After creation of chunks from file we are going to check for the candidate generation, if it exist then increment the count otherwise go for transaction. File Chunking Mechanism there are two approaches in partitioning a file into chunks:

- Fixed Size Chunking.
- Variable Size Chunking.

Fixed Size Chunking

In fixed size chunking, a file is partitioned into fixed size units, e.g., 8 Kbyte blocks. It is quick, and computationally very economical. A number of proceeding works have adopted fixed size chunking for backup applications and for large scale file systems. However, when a small amount of content is inserted to or deleted from the original file, the fixed size chunking may generate a set of chunks that are entirely different from the original ones even though most of the file contents remain intact.

Variable Size Chunking

Variable size chunking partitions a file based on the content of the file, not the offset. Variable size chunking is relatively prosperous against the insertion and deletion of the file. The Basic Sliding Window algorithm is widely used in variable size chunking. Above Fig 1. Explains the Basic Sliding Window algorithm. The Basic Sliding Window algorithm establishes a window of byte stream starting from the beginning of a file. It computes a signature, which is a hash value of byte stream in the window region. If the signature matches the predefined bit pattern, the algorithm sets the chunk boundary at the end of the window. After each comparison, the window slides one byte position and computes hash function again. In this report, an innovative multicore chunking algorithm is introduced, which parallelizes the variable size chunking. To date, most of the existing works on reduplications focus on expediting the redundancy detection process, while less attention has been paid on how to make the file chunking faster. We found that variable size chunking is computationally very expensive and is a significant bottleneck in the overall reduplications process.

Implementation Modules

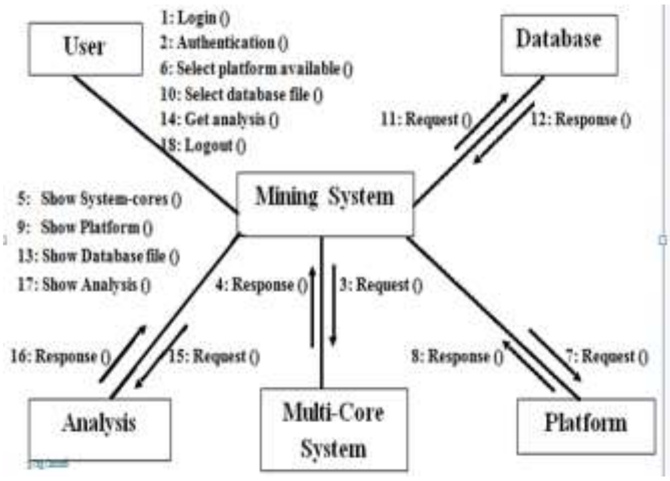


Fig.2

In this Input output processing user first login system and authenticate with password. It sends a request to multicore system and get response. Figure 2: Input Output Processing with Implementation Modules.

1. User Interaction Module

This module is to provide 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.

2. Authentication Module

Establishing the authenticity of a person or other entity. Not to be confused with authorization defining access rights to resources. This module designed for security purpose which generates unique username and password. User can login by using password and username system check the username and password and give the response.
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.

3. 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 so serial mining give the response after long time and also it works slowly the performance of serial mining is low so this is the main disadvantages of serial mining.

4. Parallel Mining Module

There are two options available to the user if he wants to select the parallel 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.

5. Performance Analyzer

One of the main concepts of the project is comparison between serial and parallel mining. How they work on different processor and raspberry pi computer and we have find the performance of serial mining and parallel mining.

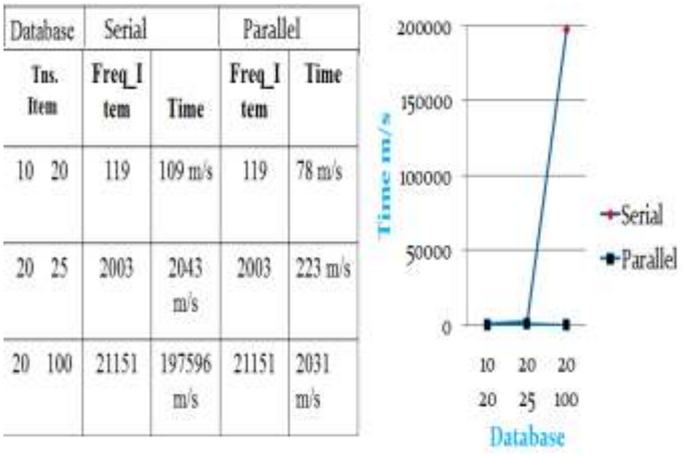
Result:

There are numbers of systems are used to test and observe 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.

System Details:

1. Processor: Intel® Core™ i3 2330M CPU Speed: 2.20 GHz RAM: 3 GB O.S/ Java version: 64 Bit

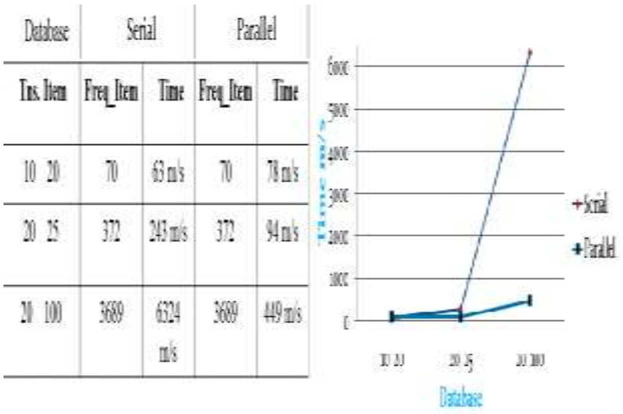
1. Minimum Support -25



Database	Serial		Parallel	
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

Figure: Comparison of Serial and Parallel Process for Min Support -25

2. Minimum Support -40

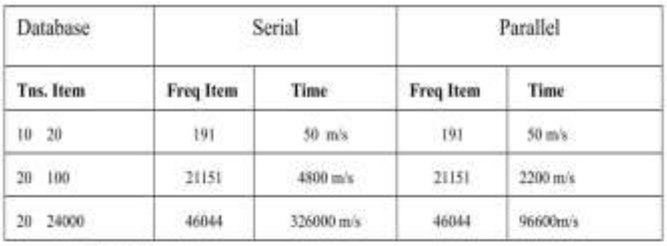


Database	Serial		Parallel	
Tns. Item	Freq_Item	Time	Freq_Item	Time
10 20	70	63 m/s	70	78 m/s
20 25	372	243 m/s	372	94 m/s
20 100	3689	6324 m/s	3689	449 m/s

Figure: Comparison of Serial and Parallel Process for min Support- 40

2. Processor: Intel® Core™ i74710 HQ CPU
Speed: 2.50 GHz RAM: 16 GB
O.S/ Java version: 64 Bit

1. Minimum Support 25



Database	Serial		Parallel	
Tns. Item	Freq Item	Time	Freq Item	Time
10 20	191	50 m/s	191	50 m/s
20 100	21151	4800 m/s	21151	2200 m/s
20 24000	46044	326000 m/s	46044	96600m/s

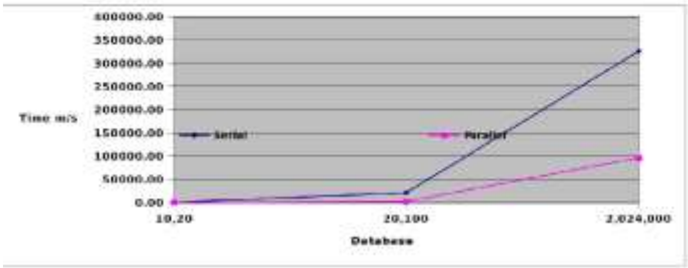
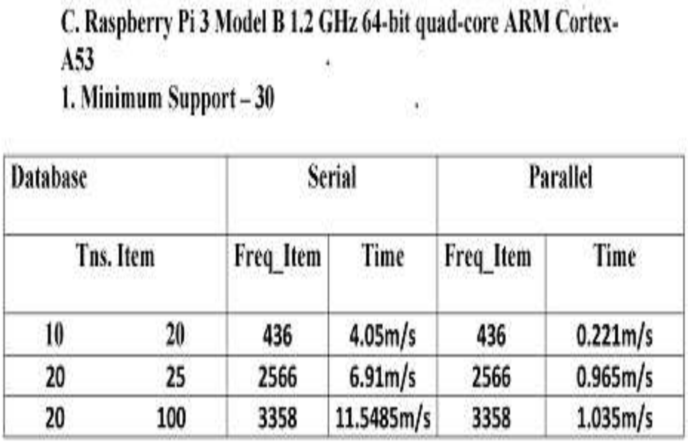


Fig: Comparison of Serial and Parallel Process for min support 25 (Processor: - Intel® Core™ i7-4710 HQ CPU)

3. Raspberry Pi 3 Model B 1.2 GHz 64-bit quad-core ARM Cortex-A53
1. Minimum Support -30



C. Raspberry Pi 3 Model B 1.2 GHz 64-bit quad-core ARM Cortex-A53
1. Minimum Support – 30

Database		Serial		Parallel	
Tns. Item		Freq_Item	Time	Freq_Item	Time
10	20	436	4.05m/s	436	0.221m/s
20	25	2566	6.91m/s	2566	0.965m/s
20	100	3358	11.5485m/s	3358	1.035m/s

REFERENCES

1. R. Agrawal and R. Srikant. "Fast algorithms for mining association rules". IBM Research Report RJ9839, IBM Almaden Research Center, San Jose, California, June 1994.

2. Youjip Won, Kyeongyeol Lim, and Jaehong Min. "MUCH: Multithreaded Content- Based File Chunking". IEEE Transaction on Computers,VOL. 64, NO. 5, ISSN: 0018-9340, May 2015.

3. Jugendra Dongre, S. V. Tokekar, and Gend Lal Prajapati. "The Role of Apriori Algo-rithm for finding the Association Rules in Data Mining". International Conference on
Issues and Challenges in Intelligent Computing Techniques (ICICT), IEEE Catalogue Number: CFP1463W-DVD ISBN: 978-1-4799-2899-6, 2014.

4. OSheila A. Abaya. "Association Rule Mining based on Apriori Algorithm in Minimizing Candidate Generation". In:International Journal of Scienti_c Engineering Research Volume 3, Issue 7, July-2012.

5. Mamta Dhanda, "An Approach to Extract E_cient Frequent Patterns from Transactional Database", In: International Journal of Engineering Science and Technology
(IJEST), Vol.3 No.7 July 2011, ISSN: 0975-546.

6. Junjie Qian, Du Li, Witawas Srisa-an, Hong Jiang and Sharad Seth, "Factors Affecting Scalability of Multithreaded Java Applications on Manycore Systems". IEEE
Transaction on Computers, International Symposium on Performance Analysis of Systems and Software (ISPASS) 2015, pp. 167-168, doi:10.1109/ISPASS.2015.7095800,2015.

7. Jaishree Singh*, Hari Ram**, Dr. J.S. Sodhi, "Improving the efficiency of Apriori algorithm by transaction reduction", In: International Journal of Scientific and Research Publications, Volume 3, Issue 1, ISSN 2250-3153, January 2013.

8. Zaki MJ (1999), "Parallel and distributed association mining: a survey. Concurrency",
In: IEEE (Volume: 7, Issue: 4), Page(s): 1425, Special issue on Parallel Mechanisms for Data Mining, ISSN: 1092-3063.
G. S. Moze College, Balewadi, Pune 59 Dept of Computer Engg Project Dissertation Parallel Frequent Item Mining Using BSW Chunking Mechanism.
9. Bala Dhandayuthapani Veerasamy, Dr. G.M. Nasira, "Setting CPU A_nity in Windows Based SMP Systems Using Java", In: IJSER (Volume: 3, Issue: 4) April-2012,
ISSN: 2229-5518.
10. N. Baddal, S. Bagga, "Implementation of Apriori Algorithm in MATLAB using Attribute A_nity M8atrix", In: IJARCSSE, vol. 4, issue 1, jan 2013.
11. A. Chatterjee and A. Rakshit, "Inuential Rule Search Scheme (IRSS) -A New Fuzzy Pattern Classi_er", In: IEEE Transaction on Knowledge and Data Engineering, vol.
16, no. 8, pp. 881-893Aug. 2004.
12. Yan Zhang, Jing Chen, "AVI: Based on the vertical and intersection operation of the improved Apriori algorithm," In: IEEE 2010.
13. Lingjuan Li, Min Zhang, "The Strategy of Mining Association Rule Based on Cloud Computing," In: 2011 International Conference on Business Computing and Global Informatization.
14. Raspberry pi education manual, Version 1.0 December 2012 [2] Pritish Sachdeva and Shrutik Katchii, “A Review Paper on Raspberry Pi”, Vol.4, No.6, Dec 2014
15. NEWS BRIEF, Published by the “IEEE Computer Society”, 0018-9162/12 © 2012
16. Eben Upton, Gareth Halfacree, “Raspberry Pi User Guide manual” 2012.
17. Raspberry Pi Start Guide.RS Components, Vsn 1.0 3/2012