Exploiting Parallel Processing Power of GPU for High Speed Frequent Pattern Mining

D.William Albert¹, Dr.K.Fayaz², D.Veerabhadra Babu³

¹Department of Computer Science & IT, Mahatma Gandhi University, Meghalaya, India
²Department of Computer Science & Technology, SK University, Anantapuramu, India
³Department of Computer Science & IT, Mahatma Gandhi University, Meghalaya, India

ABSTRACT:

Frequent pattern mining is one of the widely used data mining techniques for discovering trends or patterns from databases. As data is growing in exponential pace, data mining activities need more powerful computing. Fortunately modern GPUs (Graphics Processing Units) have specialized electronic circuits and support parallel processing. GPUs are capable of processing huge amount of data in short span of time. Data mining algorithms cannot be used directly with GPU. In this paper we proposed a new algorithm known as GPU Based Frequent Pattern Mining (GFPM) for high speed frequent pattern mining. The algorithm has additional mechanisms to handle memory and data transfer between traditional CPU and GPU. General Purpose Graphics Processing Unit (GPGPU) platform OpenCL with Java extensions is used to implement the algorithm. We built a prototype application to demonstrate the proof of concept. Datasets are obtained from UCI machine learning repository. The empirical results reveal that the GPU based FPM algorithm is able to leverage parallel processing power of GPU.

Index Terms – Data mining, frequent pattern mining, OpnCL, GPU, GFPM

I. INTRODUCTION

Association rule mining has been around in data mining to discover trends or patterns from huge amount of historical data. Generally enterprises get useful business intelligence with the help of association rule mining. Thus they can make expert decisions that help in reaching organizational goals. This mining approach can be applied to datasets in various fields like banking, insurance, finance, education, military, government, Wireless Sensor Networks (WSNs) and so on. In fact association rule mining is widely used across the fields globally for discovering actionable knowledge. With innovations in processing units used in computers, now
it is possible to achieve high speed association rule mining. Graphics Processing Unit (GPU) has become a part of the modern computer systems [5]. For a decade it has been showing its presence in the computing arena with improvements of its speed and capabilities. Modern GPU is very powerful and also supports parallel programming besides featuring bandwidth that can handle peak arithmetic operations. This kind of performance is not possible with traditional Central Processing Unit (CPU). The recent improvements in computing have positioned GPU as an alternative to leverage high speed computing. In other words, general purpose computing with GPU has become a common place that can deliver order-of-magnitude performance with respect to high speed computing. Our contributions in this paper are described here.

1. We have studied Modern GPU architecture and developed a special software component that can coordinate memory usage and data transfer between CPU and GPU. This component is used in data mining algorithms so as to use parallel power of GPU effectively.

2. We have proposed an algorithm named “GPU Based Frequent Pattern Mining Algorithm” for high speed association rule mining. This algorithm can exploit the power of modern GPU.

3. We have implemented the algorithm using the GPGPU platform OpenCL along with its Java extensions [3]. We built a prototype application to demonstrate the usefulness of the proposed algorithm using datasets collected from UCI machine learning repository [2].

The remainder of this paper is structured as follows. Section II provides a review of literature on frequent pattern mining, PGPGPU platforms, and high speed association rule mining with GPGPU platforms. Section III provides information about traditional frequent pattern mining. Section IV presents our approach to build an algorithm for high speed frequent pattern mining. Section V presents experimental results while section VI concludes the paper.

II. RELATED WORKS

Frequent Pattern Mining (FPM) has been around since the inception of data mining. The algorithms pertaining to FPM can be categorized into two types. Algorithms that are using pattern growth approach and algorithms that are using generate and test approach. FP-Growth [6] is best example for the first category while the second category are known as Apriori – like [7] algorithms. Apriori uses a bottom-up approach which extends frequent subsets. The important concept of Apriori is that if a pattern with length $k$ is not frequent then the $k+1$ pattern, its super pattern, is also not frequent. The candidates in this algorithm are independent. For this reason this algorithm is suitable for parallelization. Other examples for Apriori kind of
algorithms include [8] and [9]. There are some studies including [10], [11], and [12] that focused on distributed and parallel techniques for high speed computing.

Recently GPU has emerged suitable for high speed computing. GPU renders devices for games and other computing tasks [13]. The main difference between CPU and GPU is that CPU can execute simple instructions and functions but GPU can execute many instructions at a time in parallel as it has plethora of computing units. GPU computing can help save time and also cost in scientific and other high computing tasks. To leverage GPU power GPGPU platforms came into existence. They include Stream [14], CUDA [15], and OpenCL [16] and so on. These platforms can be used to develop data mining algorithms that can exploit the parallel processing power of modern GPUs to achieve high speed mining of patterns.

III. ASSOCIATION RULE MINING

Before going to high speed association rule mining, this section introduces association rule mining. Association rule mining problem was initially introduced by Agrawal et al. [4] in 1993. Let D be the database with plethora of transactions. Each transaction is represented by T. Let I be a set of attributes denoted as \( I = \{I_1, I_2, I_3, \ldots, I_n\} \). An association rule is the implication that can be represented in the form of \( X \Rightarrow Y \). It does mean that \( X \) implies \( Y \). \( X \) is an item which is having association with \( Y \). To measure the association statistically two measures are being used. They are known as support and confidence. The threshold for support and confidence helps to retrieve only the patterns that satisfy these measures. Thus it is possible to extract user-interested patterns that will be more useful to domain experts to make well informed decisions. The support and confidence are computed as follows.

\[
\text{Support} = \frac{\text{number of records of } A \text{ with } B}{\text{total number of records}}
\]

\[
\text{Confidence} = \frac{\text{number of records of } A \text{ with } B}{\text{the total number of records with } A}
\]

Considering \( A \) and \( B \) are items in the given dataset, the frequent item sets can be retrieved based on the given threshold and confidence. The measures like support and confidence can filter out unfit item sets that do not have required frequency. This will improve quality in association rule mining. Generally a mining program gets all associations resulting in huge number of rules. In order to obtain only interesting rules as per the guidelines suggested by domain experts, the measures like support and confidence are used. Thus the support and confidence are essential for statistical analysis of latent associations present in databases.
IV. FREQUENT PATTERN MINING WITH GPU

When algorithms are defined for frequent pattern mining they are to be different for CPU and GPU platforms. The reason behind this is that GPU has many massive processing units and they are to be exploited with parallel programming paradigm. Therefore developers who build applications for GPU platforms need to have certain specialized knowledge. In order to use the huge computational resources associated with GPU, developers need to have knowledge on underlying hardware and graphics programming. Actually programmers are to encode data to a graphics vector prior to using OpenGL or DirectX functions for rendering purposes. The data which has been rendered has to be decoded again. Thus it needs know how of graphics programming. To make such programming simple OpenCL framework came into existence. OpenCL supports programming for GPU which is cross platform in nature besides supporting parallel programming in heterogeneous environments.

When frequent pattern mining needs to be done on GPU which has many processing units the GPU will speed up the mining procedures. Each GPU has several processing units and each unit can perform simple instructions only. There are other issues like access latency and memory size. Keeping all these things in mind an algorithm needs to be written for GPU that can leverage its computational resources fully. There should be a provision for data handling between the traditional CPU and GPU. The algorithm for frequent pattern mining also should have features like highly parallel and compact data structure. The data structure needs to have compactness in order to improve speed-up ratio while using the power of GPU. As the GPU has more latency time comparatively, it is essential that the data has to be loaded into a data structure in memory in order to improve performance. When whose data set is scanned it proves to be costly and time consuming. To overcome this problem, transaction identification set has been introduced. The transaction identification set helps choosing transactions directly without the need for searching entire database. Thus he compact data structure helps in processing data faster. Frequent pattern mining needs to take care of memory access between CPU and GPU. In order to make this communication simple a class is defined in Java programming language which in storing data and converting as per the requirements of the application at runtime. Another Java class is defined for handling number of threads and handling underlying parameters.

A. Program Architecture Needed for Leveraging Power of GPU
The frequent pattern mining needs to have the architecture as illustrated in Figure 1.

![Figure 1 - Proposed Program Architecture for Frequent Pattern Mining on GPU](image)

In tune with the architecture presented in Figure 1 the algorithm is defined specifically for leveraging the parallel programming supported by the GPU. The algorithm is as presented in Figure 2 below which makes use of the architectural components described in [1].

**B. Frequent Pattern Mining Algorithm for Parallel Processing using GPU (GFPM Algorithm)**

<table>
<thead>
<tr>
<th>Algorithm: Frequent Pattern Mining on GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Dataset $D$ and minimum support $\text{minsup}$</td>
</tr>
<tr>
<td><strong>Output:</strong> Frequent patterns $FP$ that satisfy $\text{minsup}$</td>
</tr>
<tr>
<td>1. Obtain $D$ from disk</td>
</tr>
<tr>
<td>2. Scan $D$ and using compact structure to represent it</td>
</tr>
<tr>
<td>3. Use memory handler component to send data to GPU</td>
</tr>
<tr>
<td>4. Generate candidate itemset</td>
</tr>
<tr>
<td>5. Use memory handler to fetch storing candidates</td>
</tr>
<tr>
<td>6. Use memory handler to send generated candidates to GPU</td>
</tr>
<tr>
<td>7. Use memory handler to store results</td>
</tr>
<tr>
<td>8. Use thread and parameter handler to store required details and parameters</td>
</tr>
<tr>
<td>9. The thread handler has to perform the following on GPU</td>
</tr>
<tr>
<td>10. Allocate set of candidate itemsets to each processing unit</td>
</tr>
<tr>
<td>11. Choose itemsets that has $\text{minsup}$</td>
</tr>
<tr>
<td>12. Use memory handler to get the results back</td>
</tr>
<tr>
<td>13. Return $FP$</td>
</tr>
</tbody>
</table>

![Figure 2 – Illustrates proposed algorithm for frequent pattern mining on GPU](image)

OpenCL platform is used to implement this algorithm that makes use of program components that are specially built to handle communication between CPU and GPU besides handling parameters and threads to leverage parallel processing power of GPU. Since GPU has plenty of processing units, each processing unit is given a set of candidate item set where the threshold is verified and the results are chosen that satisfy the minimum support value. Thus the whole processing is done in very short span of time.

D.William Albert, Dr.K.Fayaz and D.Veerabhadra Babu  
75
V. EXPERIMENTS AND RESULTS

This section provides details of the experiments made including the environment used, the datasets used for experiments and the performance of the proposed algorithm besides comparing the CPU and GPU performances. The software and hardware environment used for experiments is presented in Table 1.

<table>
<thead>
<tr>
<th>ITEM</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>AMD Phenom II X4 965 3.4 GHz</td>
</tr>
<tr>
<td>GPU</td>
<td>ATI Radeon HD 5850 with 1440 stream processing units and 1 G DDR5 memory</td>
</tr>
<tr>
<td>Memory</td>
<td>8G DDR3 memory</td>
</tr>
<tr>
<td>OS</td>
<td>Microsoft Windows 7</td>
</tr>
<tr>
<td>GPGPU PLATFORM</td>
<td>OpenCL</td>
</tr>
<tr>
<td>JAVA BINDINGS</td>
<td>Java OpenCL</td>
</tr>
</tbody>
</table>

Table 1 – Environment used for Experiments

Many datasets are used for extracting frequent patterns using GPU platform. They are collected from UCI machine learning repository [2]. However, the results obtained using chess dataset are presented in this paper. For testing the performance of the algorithm only 100 transactions are considered for experiments. Experiments are made in terms of computational time taken, speed up ratio, and the performance differences between CPU and GPU. The results obtained from chess dataset are presented as a series of graphs in this section. Different block sizes, thresholds, execution time and speedup ratio are observed besides testing the performance of CPU and GPU.

![Computation Time: Various Threshold](image)

Figure 3 – Computation time of different thresholds

D.William Albert, Dr.K.Fayaz and D.Veerabhadra Babu
As can be seen in the Figure 3, it is understood that the number of threads have their impact on the computational time. The thresholds also can have their impact on the computation time for that matter. As threshold increases, the computational time also increases. As number of threads increase, computation time decreases.

![Speed-up Ratio: Various Threshold](image1)

**Figure 4 – Speed up ratio of different thresholds**

As can be seen in the Figure 4, it is understood that the number of threads have their impact on the speed up ratio. The thresholds also can have their impact on the speed up ratio for that matter. As threshold increases, the speed up ratio decreases. As number of threads increase, speed up ratio increases.

![Speed-up Ratio: Various Block Sizes](image2)

**Figure 5 – Speed up ratio of different blocks**

As can be seen in the Figure 5, it is understood that there is no significant impact of the block sizes on the speedup ratio. The block sizes 2, 4, and 6 are considered for experiments. The horizontal axis represents number of threads while the vertical axis shows speed up ratio.
As can be seen in Figure 6, it is evident that GPU’s processing units’ occupational time is more when compared to that of CPU. This is quite natural that GPUs are to be utilized in parallel programming and CPU does less work. This is revealed in the results. One interesting observation here is that the GPU’s occupation time reduced when number of threads is increased. It does mean that more threads will finish the job faster.

With another similar experiment, as can be seen in Figure 7, it is evident that GPU’s processing units occupational time is more when compared to that of CPU. This is quite natural that GPUs are to be utilized in parallel programming and CPU does less work. This is revealed in the results. One interesting observation here is that the GPU’s occupation time reduced when number of threads is increased. It does mean that more threads will finish the job faster.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we studied frequent pattern mining process with GPU. Traditional data mining algorithms for frequent pattern mining cannot be used with modern GPU directly. We proposed an algorithm known as GPU Based Frequent Pattern Mining Algorithm to leverage the parallel
processing power of GPU. OpenCL with Java extensions is the GPGPU platform used to implement the algorithm. Many datasets are collected from UCI machine learning repository for experiments. The results reveal that when number of threads is more, computation time is reduced. When threshold is increased, the computation time is increased. The overall occupational time of GPU is more and CPU is less as expected. The algorithm lets all processors of GPU to be busy with the data mining task. Thus huge amount of data can be processed in short span of time. Proposing Apriori-based algorithm for high speed association rule mining with GPU and developing data mining algorithms for Big Data mining with MapReduce programming are our future endeavors.
REFERENCES


[2] UCI machine learning repository


[6] [1]. Han, Y. Pei, Y. Yin et al., "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," Data Mining and Knowledge Discovery, vol. 8,


AUTHORS BRIEF INTRODUCTION:

D.William Albert, Dr.K.Fayaz and D.Veerabhadra Babu
D. William Albert, is a Research Scholar in Mahatma Gandhi University, Meghalaya in Data Mining specialization. Basically a B.Sc. Science Graduate and did his Post-graduation Masters in Computer Science in 2004 and Master of Technology in Computer Science & Engineering in 2006. He is a Life Member of professional body such as Indian Society for Technical Education, New Delhi. More than 15 National & International technical papers published and guided M.Tech, M.C.A. and B.Tech Students in their Project works. The area of interests includes Data Warehousing & Mining, Software Engineering, Software Testing Methodologies, Software Project Management, Distributed Operating System, Human Computer Interface, E-commerce, Enterprise Resource Planning, etc. e-mail Id: albertdwgtl@gmail.com

Dr. K. Fayaz is working as a faculty of system (System In-charge) in Sri Krishnadevaraya Institute of Management (SKIM), Sri Krishnadevaraya University, Anantapuramu. He acquired B.Sc., from S.K.University, Post Graduate Diploma in Computer Application from JNTU, Anantapuramu and Master of Information Technology from Manipal Academy of Higher Education (MAHE) Deemed University, Manipal. Subsequently has done M.Phil and Ph.D from S.K.University in Computer Science & Technology. He has published more than 10 papers in Computer Science International Journals. He has attended and presented papers in several International, National Conferences, Seminars, Workshops. E-mail Id: fayazk16@gmail.com

D. Veerabhadra Babu graduated in Master of Computer Applications in 2005 and Master of Technology in Information Technology in 2011 from K.S.O.University, Karnataka and is a Research Scholar in Mahatma Gandhi University, Meghalaya in Data Mining specialization. The area of interests includes Data Warehousing & Mining, Software Engineering, Software Testing Methodologies, Software Project Management, Human Computer Interface, Management Information Systems, etc. E-mail Id: veerabhadra.durgam@gmail.com

Address for Communication:
D.WILLIAM ALBERT,
H.No:6/776-G2, UPSTAIRS,
RAJIV COLONY,
GUNTAKAL (POST) – 515 801,
ANANTAPURAM (DIST),
ANDHRA PRADESH (STATE).
+91 9440 424 763.

D. William Albert, Dr. K. Fayaz and D. Veerabhadra Babu 81