International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 02, February-2015

Time Efficient Sentinel Data Mining using GPU

N. M. Sonawane
Computer Engineering Department
Late G. N. Sapkal College of Engineering,
Anjneri, Nashik
University of Pune,Maharastra,India

Prof. B. R. Nandwalkar
Computer Engineering Department
Late G. N. Sapkal College of Engineering,
Anjaneri, Nashik
University of Pune,Maharashtra,India

***Abstract*—Open Computing Language is used to implement the SentBit algorithm. For parallelization here we are using GPU (i.e. Graphics Processing Unit) The modules get executed on parallel processor and then store the result on shared memory. Due to this the execution time of existing system will get reduced. Graphics processing Unit use large blocks of data parallely. GPU make highly parallel structure which is more effective than general-purpose CPUs. The use of a GPU together with a CPU is nothing but GPU computing, which is used to accelerate general-purpose scientific and engineering applications. Few number of cores contains in CPU where as GPU contains 100s of cores to process data in parallel. We compare time required for execution of same data set for both i.e. sequential and parallel algorithm. The execution will give us Sentinels which are nothing but decisive points. So with help of these decisive points we can predict the behaviour of particular data set. For example since we had taken data of share market for particular companies so it's sentinels will predict the behavior of that particular share whether it is increasing or decreasing.This will help end user to take decision whether he has to buy or sell that particular share.**

Keywords- Sentinels,Predictive mining, cube based data mining, Pattern mining

I. INTRODUCTION

Data mining is defined as to discover pattern in large data set. 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.

This leads us to move Sentinel Mining approach. Sentinel Mining Approach can be used for discovering the relationship between measures of multidimensional data cube that is represented by Sentinels. The sentinel mining is nothing but to find out the sentinels. With the help of these sentinels we can predict the behavior of that particular data set.

In proposed system there are total eight modules. First module gives reading of data from files. Then second and third module displays encoding data into binary form and then create bitmap for each source measure. Fourth and fifth module express test sentinel which are created and calculate score for each sentinel. After that sort the sentinels, if they are not in the specified range of threshold then discard them. Then lastly predicting behavior of data set which will depend on output of sentinels. The rest of the paper is organized as follows. We present and discuss some related works in section II. Section III describes the System proposed in mentioned work. In section IV Result analysis is explained, Conclusion is given in section V and References are given in section VI.

Here sentinel mining is achieved by using indication stream. With help of this indication stream we can find out positive and negative sentinels. So these sentinels will act as decisive points and these decisive points will be used for finding out which share goes up or down.

If we refer sequential SentBit algorithm there CPU execute instructions sequentially. There is use of parallel execution of instruction which is done by GPU which will give less processing time of instruction. The Objective of this approach is to provide a generalized sentinel mining approach by reducing processing time of processor. In the proposed system we had implemented SentBit algorithm sequentially then this sequential algorithm is converted into parallel SentBit algorithm. So final output will show difference between time required by CPU and time required by GPU for this given data set.

II. LITERATURE SURVEY

Morten Middelfart, Torben Bach Pedersen have proposed that there is use of sequential sentinel mining algorithm. But using Graphics processing unit it should be change into parallel sentinel mining algorithm. So execution time of algorithm should be reduced[1].

M. Middelfart have proposed source measure and target measures and the causal relationship between them.The sentinels acts as a key due to the end user. So this sentinel can found during data mining process[2].

T.B. Pedersen proposed that one or more source measures changes in multidimensional data cube are expected to cause a change another measure critical to user. It provide good performance to user on real and operational data warehouse[3].

T. Imielinski, L. 'Khachiyan, and A. Abdulghani, have proposed that generalization of cube grades. The aggregation of measures of cube is done by MAX,MIN,SUM,AVG numerical attributes. The cube specialization, roll up, mutation are the operations performed on Cubegrades. In this paper cubegradescan not show to analyst as a final product[4].

J. Yang, W. Wang, P.S. Yu, and J. Han, suggest that the support which shows the pattern discovery on large sequences. During the noise the symbols may be misplaced with another symbols. For cancellation of noise there is use of biomedical study and consumer behavior. If the pattern length increases then the pruning techniques cannot work properly[5].

Y. Zhu and D. Shasha,suggest that efficient method for solving problem based on discreate fourier transform and three level time interval hierarchy. It improve performance of previeousfourier transform algorithm. This algorithm is incremental and has fixed response time and can monitor the pairwise correlation of 10,000 streams on single PC[6].

J. Pei, J.Han, B.Mortazavi-Asl, H.Pinto, Q.Chen, U. Dayal proposed that the set of sequences in which each set consists of list of elements and each element contain set of items. So all this sequential pattern mining discovered frequent subsequences. This paper cannot show level by level and bi level projection[7].

III. PROPOSED SYSTEM

A. System Architecture

Fig.1 shows proposed system architecture. In first step there is reading of data from file. In second step there is generation of bitmap vector, so encode data in binary form. There is convert feature to one dimensional feature vector. This bitmap vector generator form number of bitmap vector. Then after that there is creation of bitmap vector for source measure. This bitmap vector is in 0 and 1 form. So with help of this there is creation of bitmap vector for each source measure. Test bitmap quality measure by parallel processing which is nothing but finding score of source measures. With the help of source measure we can calculate score, confidence, support and balance for bitmap. Then we have to find score of all sentinels which are source sentinel and target sentinels. There will be finding of score for each measure in parallel. Then we have to compare score value with these threshold value. After that if score is greater than threshold then this value should be accepted otherwise it must be rejected. After comparing the value the next step is to add bitmap to sentinels. There is formation of bitmaps and add this to sentinels.

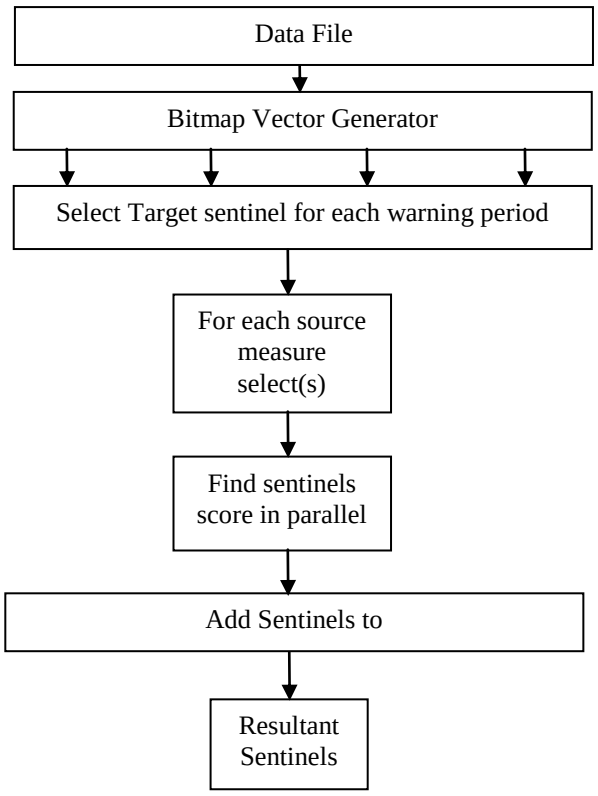

Fig.1 System architecture

The parallisation is done with help of sent bit parallel algorithm. Lastly get the resultant sentinels and to store the result in the CPU. After comparing score and threshold then there is addition of bitmap to the sentinels. So there is comparison of all process which takes part in execution of process.

B. *Algorithm*

Sentinel Mining Parallel algorithm:

Input: max number of sentinels to be returned, n,a data set,C, a set of source measures,S1...Sp, a target measures, thresholds (α,β,γ, σ), Maxw

1:for all S in {S1, . . . , Sp} do
// It shows set of source measures.
2: Allocate memory for Bitmap (M), fill(0)
//It shows allocation of memory for bitmap.
3: Allocate memory for MaxElimSupp
4: for w=1 to Maxw do
5: for x=1 to p do
5.1: for all S ∈ {$S_{AddSource}$; $S_{AddSource}$ } do
5.2: NewBits<-Bitmap(S) AND Bits.
//It shows AND operation for the Bitmaps.
5.3: if BitCount =BitNewBits+PI+NI<Bits+PI+NI
//It shows BitCount operation for the Bitmaps.
5.4: NewSource = Source U S
5.5: for all T do
5.5.a: SentSupp = SourceBits+PI+NI

5.5. b:Confidence = $\frac{|A|+|B|}{Sent\ Supp}$

5.5.c:Balance= $\frac{4*|A|*|B|}{(|A|+|B|)2}$

5.5.d:NewScore= { 1- wp + $\frac{[1+Maxw-w]*wp}{Maxw}$} *

$\{\frac{1}{2}+\frac{1+MaxSource-SourceCnt}{MaxSource*2}\}$ *

(A+B) * Confidence * ($\frac{1}{2}+\frac{Balance}{2}$)

//From above formulae's we can calculate SentSupp, Confidence,Balance and NewScore.
6: if NewScore>Score then
7: if
SentSupp ≥ σ ^ Confidence ≥ γ ^ Balance ≥ β
8: Append NewSource
9: return the top n sentinels from SentList

So from above algorithm it must be concluded that the performance of the system is improved.

Where α is Indication
β is Balance
γ is Confidence
σ is SentSupp

We take number of source measure like S1,S2,……..Sp. Then we define target measure, thresholds (α,β,γ, σ). After this we have to allocate memory for bitmap vector. Then these bitmaps will get added and get stored on new bit. With the help of bit count in the bit map we will get score.

C. *Related Work*

GPU Architecture: Following GPU architecture shows N compute unit which are included in compute device. In compute device consists of number of private memories. This compute unit is communicated with local memory and Global/Constant memory data cache. Compute device

memory consists of global memory and constant memory. This global and constant memory also communicated with global/constant memory data cache.

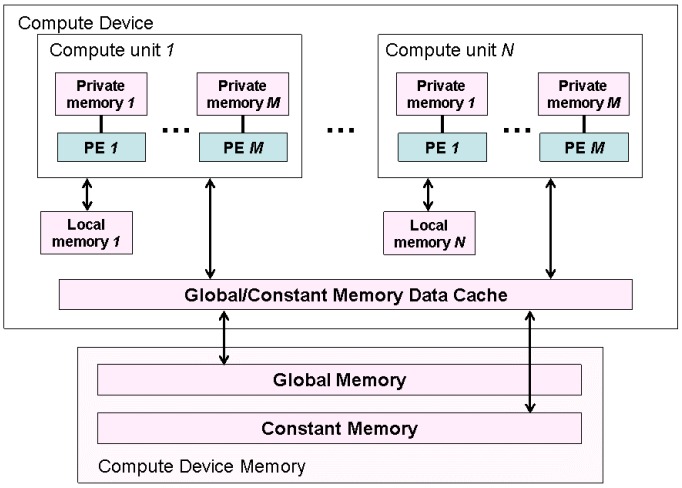

Fig.2 GPU Architecture

IV. RESULT ANALYSIS

Module Results: Module result consists of bitmap vectors for each feature, which are stored in intermediate file on disk. Each bitmap vector consists of bits based on value of feature instance and value of indicator function.

Data Set : In the proposed system there is use of AXIS,SBI & RELIANCE dataset. The data set consists of following features, the description of features are as follows:

Open: It is the stock price when market starts at 9:00 AM IST, It's a float value.

High: It is the highest stock price for the day, it's a numeric value.

Low: It is the lowest stock price for the day, it's a float value.

Close: It is the Price of stock at the time of market closure for the day, it's a float value.

Volume: It is the total number of stock available for transaction for the day, it's an integer value.

Following graph shows time required for analysis of sequential & parallel algorithms for considered data set.

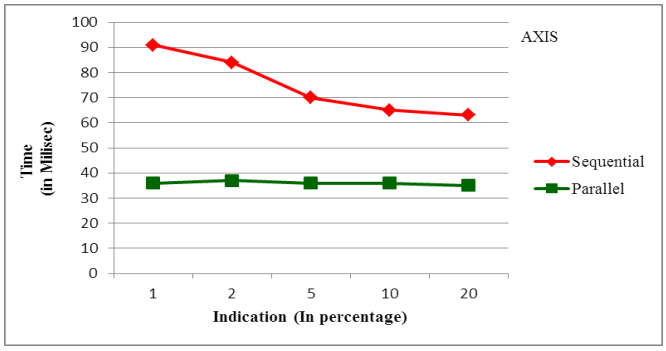

Fig.3 Time Comparison Graph for Axis Data Set

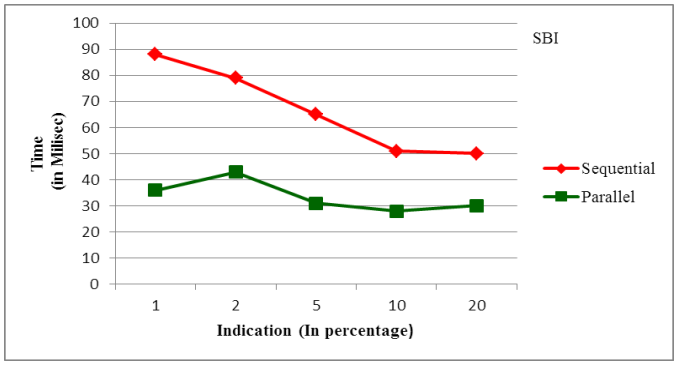

Fig.4 Time Comparison Graph for SBI Data Set

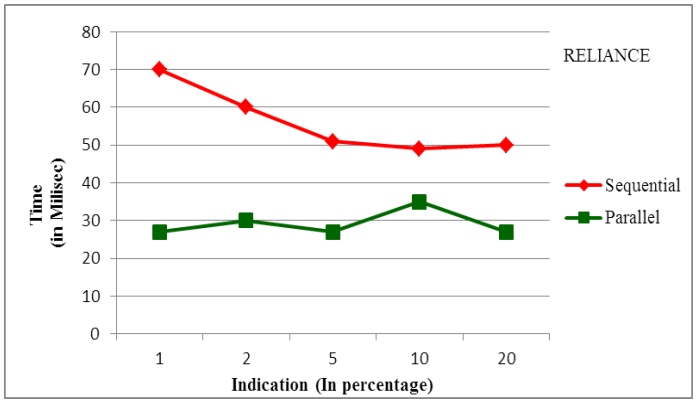

Fig.5 Time Comparison Graph for Reliance Data Set

In Fig.No.3,4 & 5 time required for analysis of data on sequential & parallel algorithm is compared, where X-axis represents value of Indication and Y-axis represents time required for the analysis on sequential & parallel algorithm respectively.

Red graph line shows time required for the analysis of sequential algorithm and green graph line shows time required for the analysis of parallel algorithm. If we see the graph lines, which shows different values of indication (i.e. 1,2,5,10 & 20 percent) we can see the considerable variation in the time required for analysis of sequential algorithm.

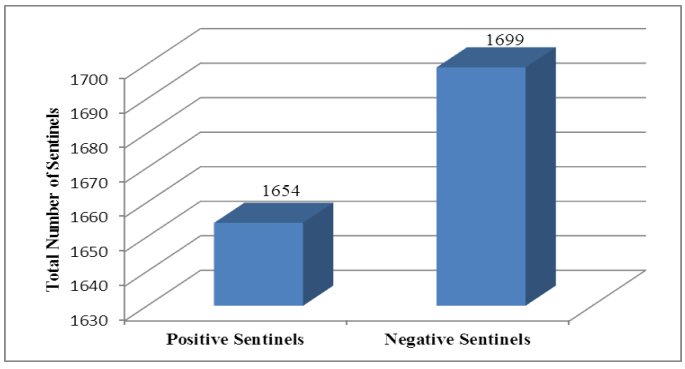

Fig.6 Sentiel Mining Graph for AXIS Data Set

Fig No.6 shows decision about the particular dataset from number of positive & negative sentinels and we get these sentinels after the execution of the algorithm.

So for above case we can say that this share should be sold out since number of negative sentinels are more.

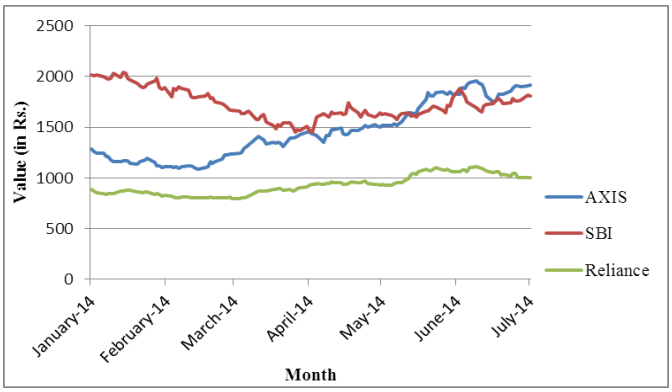

Fig.7 Different Shares Comparison Graph

FigNo.7 shows comparison of shares in monthwise format.X-axis represents month & Y-axis represents value of that particular share in rupees.

So from table No. 1 if we see the value for indication we can see the considerable difference in the time required for the analysis of different data set. For example if we consider indication 10 percent for Axis data set then it takes 65 Milisecond for analysis by sequential algorithm and 36 Milisecond for analysis by parallel algorithm which is our proposed system. So we can say that the proposed system is faster by 29 Milisecond.

V. CONCLUSION AND FUTURE SCOPE

GPU (Graphics Processing Unit) is used for parallel algorithm to reduce processing time.So we propose parallel sentbit algorithm to reduce time required for analysis.
With help of this sentbit algorithm we can mine sentinels and these mined sentinels will have positive & negative sentinels which will help to predict result.

For future work we can used multiple GPU processors.

Table 1: Time (Milisec) Comparison table for Axis, SBI & Reliance

INDICATION	AXIS			SBI			RELIANCE		
	Sequential	Parallel (Proposed System)	Difference	Sequential	Parallel (Proposed System)	Difference	Sequential	Parallel (Proposed System)	Difference
1	91	36	55	88	36	52	70	27	43
2	84	37	47	79	43	36	60	30	30
5	70	36	34	65	31	34	51	27	24
10	65	36	29	51	28	23	49	35	14
20	79	36	43	50	30	20	50	27	23

VI. REFERENCES

[1] Morten Middlefart,"*Efficient sentinel mining using bitmap on modern processor*,"IEEE transactions on knowledge and data engineering,vol 25,no 10,October 2013.

[2] M. Middelfart,"*Using Sentinel Technology in the TARGIT BI Suite,"* Proc. VLDB Endowment,vol. 3,no. 2, pp. 1629-1632, September 2010.

[3] T.B. Pedersen, "*Implementing Sentinel Technology in the TARGIT BI Suite,"* Proc. IEEE 27th Int'l Conf. Data Eng. (ICDE), pp. 1187-1198, September 2011.

[4] T. Imielinski,"*Cubegrades: Association Rules,"*Data Mining Knowledge Discovery,vol.6,no.3,pp 219-257,September 2002.

[5] J. Yang,"*Mining Long Sequential Patterns in a Noisy Environment*,"Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 406-417, June 2002.

[6] Y. Zhu and D. Shasha, "*StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time*," Proc. 28th Int'l Conf. Very Large Data Bases (VLDB), pp. 358-369,

[7] June 2002.

[8] J. Pei, "*PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth*," Proc. 17th Int'l Conf. Data Eng. (ICDE), pp. 215-224, 2001.

[9] Intel, Intel SSE4 Programming Reference, July 2007.

[10] "Advanced Micro Devices," Software Optimization Guide for AMD Family 10h Processors, November 2008.

[11] J. Han and M. Kamber, "*Data Mining Concepts and Techniques,"* second ed. Morgan Kaufmann Publishers, 2006.

[12] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu, "FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 355-359, 2000.

[13] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, *"Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach,"* IEEE Trans. Knowledge Data Eng., vol. 16, no. 11, pp. 1424-1440, November. 2004.

[14] F. Nakagaito, T. Ozaki, and T. Ohkawa, *"Discovery of Quantita-tive Sequential Patterns from Event Sequences*," Proc. IEEE Int'l Conf. Data Mining Workshops (ICDM), pp. 31-36, 2009.

[15] P. Shenoy, J.R. Haritsa, S. Sudarshan, G. Bhalotia, M. Bawa, and D Turbo-Charging ,"*Vertical Mining of Large Databases,"* Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 22-33, 2000.

[16] J. Han and M. Kamber,"*Data Mining Concepts and Techniques*, "second ed. Morgan Kaufmann Publishers, 2006.