Vol II Issue IX March 2013

Impact Factor : 0.1870

ISSN No :2231-5063

Monthly Multidisciplinary Research Journal

Golden Research Thoughts

Chief Editor
Dr.Tukaram Narayan Shinde

Publisher
Mrs.Laxmi Ashok Yakkaldevi

Associate Editor
Dr.Rajani Dalvi

Honorary
Mr.Ashok Yakkaldevi

IMPACT FACTOR : 0.2105

Welcome to ISRJ

RNI MAHMUL/2011/38595

ISSN No.2230-7850

Indian Streams Research Journal is a multidisciplinary research journal, published monthly in English, Hindi & Marathi Language. All research papers submitted to the journal will be double - blind peer reviewed referred by members of the editorial Board readers will include investigator in universities, research institutes government and industry with research interest in the general subjects.

International Advisory Board

Flávio de São Pedro Filho Federal University of Rondonia, Brazil	Mohammad Hailat Dept. of Mathmatial Sciences, University of South Carolina Aiken, Aiken SC 29801	Hasan Baktir English Language and Literature Department, Kayseri
Kamani Perera Regional Centre For Strategic Studies, Sri Lanka	Abdullah Sabbagh Engineering Studies, Sydney	Ghayoor Abbas Chotana Department of Chemistry, Lahore University of Management Sciences [PK]
Janaki Sinnasamy Librarian, University of Malaya [Malaysia]	Catalina Neculai University of Coventry, UK	Anna Maria Constantinovici AL. I. Cuza University, Romania
Romona Mihaila Spiru Haret University, Romania	Ecaterina Patrascu Spiru Haret University, Bucharest	Horia Patrascu Spiru Haret University, Bucharest, Romania
Delia Serbescu Spiru Haret University, Bucharest, Romania	Loredana Bosca Spiru Haret University, Romania	Ilie Pintea, Spiru Haret University, Romania
Anurag Misra DBS College, Kanpur	Fabricio Moraes de Almeida Federal University of Rondonia, Brazil	Xiaohua Yang PhD, USA
Titus Pop	George - Calin SERITAN Postdoctoral Researcher	Nawab Ali Khan College of Business Administration

Editorial Board

Pratap Vyamktrao Naikwade ASP College Devrukh,Ratnagiri,MS India	Iresh Swami Ex - VC. Solapur University, Solapur	Rajendra Shendge Director, B.C.U.D. Solapur University, Solapur
R. R. Patil Head Geology Department Solapur University, Solapur	N.S. Dhaygude Ex. Prin. Dayanand College, Solapur	R. R. Yalikar Director Managment Institute, Solapur
Rama Bhosale Prin. and Jt. Director Higher Education, Panvel	Narendra Kadu Jt. Director Higher Education, Pune	Umesh Rajderkar Head Humanities & Social Science YCMOU, Nashik
Salve R. N. Department of Sociology, Shivaji University, Kolhapur	K. M. Bhandarkar Praful Patel College of Education, Gondia	S. R. Pandya Head Education Dept. Mumbai University, Mumbai
Govind P. Shinde Bharati Vidyapeeth School of Distance Education Center, Navi Mumbai	Sonal Singh Vikram University, Ujjain	Alka Darshan Shrivastava Shaskiya Snatkottar Mahavidyalaya, Dhar
Chakane Sanjay Dnyaneshwar Arts, Science & Commerce College, Indapur, Pune	G. P. Patankar S. D. M. Degree College, Honavar, Karnataka	Rahul Shriram Sudke Devi Ahilya Vishwavidyalaya, Indore
Awadhesh Kumar Shirotriya Secretary, Play India Play (Trust),Meerut	Maj. S. Bakhtiar Choudhary Director,Hyderabad AP India.	S.KANNAN Ph.D , Annamalai University,TN
	S.Parvathi Devi Ph.D.-University of Allahabad	Satish Kumar Kalhotra
	Sonal Singh	

**Address:-Ashok Yakkaldevi 258/34, Raviwar Peth, Solapur - 413 005 Maharashtra, India
Cell : 9595 359 435, Ph No: 02172372010 Email: ayisrj@yahoo.in Website: www.isrj.net**

A DETAILED STUDY ON PERFORMANCE OPTIMIZATION OF CLUSTERING ON GPU

VIJAL D. PATEL AND SUMITRA MENARIA

Computer Science & Engineering Department
Parul Institute of Technology,P.O. Limda, Tal . Waghodiya.Ahmedabad, India.
Computer Science & Engineering Department
Parul Institute of Engineering &Technology,P.O. Limda, Tal . Waghodiya.Baroda, India.

Abstract:

Clustering can be considered the most important unsupervised learning technique. Clustering is “the process of organizing objects into groups whose members are similar in some way”. It is used for simplification, pattern detection and data concept construction. As clustering is prime component for many scientific and engineering applications using Data mining, This paper focus on optimizing performance of clustering using parallelization. Many Clustering algorithms give best performance on small dataset. But when large dataset will apply then performance will main issue so that will optimize using GPU. Because GPU giving much Higher Performance. parallelization of clustering will be implemented on GPU based various programming modal like CUDA, Open MP, and Open Acc. Comparison of result will be used to suggest best platform for clustering.

KEY-WORDS:

Clustering, data mining, GPU Computing, CUDA, Scalable parallel programming, Performance Evaluation.

1.INTRODUCTION

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

A "clustering" is essentially a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example a hierarchy of clusters embedded in each other. Clustering can be roughly distinguished in:

Hard clustering: each object belongs to a cluster or not

Soft clustering (fuzzy clustering): each object belongs to each cluster to a certain degree (e.g. a likelihood of belonging to the cluster)

Classification Of Clustering

Traditionally clustering techniques are broadly divided in [1]:

Title :A DETAILED STUDY ON PERFORMANCE OPTIMIZATION OF CLUSTERING ON GPU
Source:Golden Research Thoughts [2231-5063] VIJAL D. PATEL AND SUMITRA MENARIA yr:2013 vol:2 iss:9

Hierarchical methods
Partitioning Methods
Density Based Algorithms
Grid Based Clustering

Hierarchical Methods

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. Hierarchical clustering generally falls into two types:

Agglomerative Algorithm
Division Algorithm

In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place, which may run from a single cluster containing all objects to n clusters each containing a single object. Hierarchical Clustering is subdivided into agglomerative methods, which proceed by series of fusions of the n objects into groups, and divisive methods, which separate n objects successively into finer groupings. Hierarchical clustering may be represented by a two dimensional diagram known as dendrogram which illustrates the fusions or divisions made at each successive stage of analysis.

Partitioning Methods

The partitioning methods generally result in a set of M clusters, each object belonging to one cluster. Each cluster may be represented by a centroid or a cluster representative; this is some sort of summary description of all the objects contained in a cluster. The precise form of this description will depend on the type of the object which is being clustered. In case where real-valued data is available, the arithmetic mean of the attribute vectors for all objects within a cluster provides an appropriate representative; alternative types of centroid may be required in other cases, e.g., a cluster of documents can be represented by a list of those keywords that occur in some minimum number of documents within a cluster. If the number of the clusters is large, the centroids can be further clustered to produce hierarchy within a dataset.

Density-Based Algorithms

Density-based algorithms are capable of discovering clusters of arbitrary shapes. Also this provides a natural protection against outliers. These algorithms group objects according to specific density objective functions. Density is usually defined as the number of objects in a particular neighborhood of a data object. In these approaches a given cluster continues growing as long as the number of objects in the neighborhood exceeds some parameter.

Grid Based Clustering

These focus on spatial data i.e. the data that model the geometric structure of objects in the space, their relationships, properties and operations. This technique quantizes the data set into a no. of cells and then work with objects belonging to these cells. They do not relocate points but rather builds several hierarchical levels of groups of objects. The merging of grids and consequently clusters, does not depend on a distance measure. It is determined by a predefined parameter.

1.1 Issues in Clustering

·Proper Classification as per the Application.
Performance is the major issue when data increases in mess.

1.2 Mode of Performance Optimization

Performance optimization is foremost objective when we are performing many computations on processor. So following techniques are used for performance optimization:

Efficient Resources (Devices)
Optimized computing Algorithm

Best utilization of resources with less cost (Parallel processing)
This paper proposes a mechanism for High Performance Computing.

2. GPU ARCHITECTURE

GPU (Graphics Processing Unit) have been of large interest in last decade to achieve high performance computing through massive data parallelism model. GPU has hundreds of core processors. GPUs are designed for graphics. GPU is used to process similar operation on independent vertices & pixels in parallel. GPU is more suitable for stream computations like SIMD (single Instruction multiple data) programming model. For higher efficiency, GPU processes many data elements in parallel with same program module.

Last decade, to satisfy demand of high performance computing a tremendous rise in computing power has been observed. Although CPU has been constantly increasing their performance, the growth has been over shadowed by the improvement in performance of GPU. This trend has led to draw the interest of researchers with high computational requirements to look for GPUs as a possible solution. This has led to the emergence of a new term GPGPU (General Purpose computing using Graphical Processing Units).

GPGPU is defined as General Purpose computation using GPU and graphics API in applications other than 3D graphics. In order to achieve higher efficiency, GPGPU is designed as a combination between hardware components and software that allows the use of a traditional GPU (Graphics processor) to perform non-graphical computing tasks with higher processing power. [2]

Figure 1: GPU Architecture [2]

Fig. 1 describes GPU architecture. Each GPU has number of Streaming Multiprocessors(SM) i.e.30 SMs and each streaming multiprocessor has 8 streaming core processor. This architecture has characteristics to perform same operation on collection of records (Data parallelism). Hence, GPU uses streaming processor as core processor to achieve massive data parallelism. Each NVIDIA GPU has a hundreds of parallel cores and within each core has floating point unit, logic unit (add, sub, mul), move and compare unit, and a branch unit.

Each core is managed by a thread manager. A thread manager can spawn and manage hundreds of threads per core. A very important feature is that it uses lightweight threads that have little creation overhead and zero-overhead scheduling. The GPU is highly parallel compute coprocessor that has its own device memory with high bandwidth.

3. CUDA

Compute Unified Device Architecture (CUDA) is a Programming model and the APIs required for performing computation tasks on NVIDIA GPUs. CUDA is a scalable highly parallel and multithreaded API to execute a program on any number of processor without recompiling. CUDA allow a program written for a platform to be executed in other platform without recompilation. CUDA provides efficient programming environment for regular applications. CUDA is designed to support many programming languages as shown in Fig.2. It provides extension to programming languages for performing computation on NVIDIA GPUs. CUDA C is extension to C language with CUDA API to operate on NVIDIA GPU. It

allows a programmer to write a program for one thread, and it instantiate for multiple parallel threads.

Figure 2: CUDA Interface

3.1 CUDA Programming Execution model

Figure 3: CUDA execution model

Programming model provided for designing and executing general purpose computation kernel on NVIDIA GPU is named as Compute Unified Device Architecture (CUDA). CUDA programming modal is available for most known operating systems Windows, and Linux. CUDA programming modal allows programmer to use GPU for parallel programming without knowledge of graphics. Fig.3 describes CUDA computational model to be executed on CPU

and GPU both. A CUDA program is sequential host program and one or many kernel code as shown in Fig.3. Serial code will be executed by a CPU thread, and Kernel code will be executed by GPU threads grouped in a block. Kernel here specifies portion of computational code to be executed on NVIDIA GPU device. CUDA kernel executes on a grid of thread blocks, where a thread block is a group of threads that work in SIMT (Single Instruction Multiple Thread) model. Threads of each block execute cooperates each other using shared memory and synchronizing its execution. Threads from different blocks operate independently. A grid can be of 1 or 2 dimensions, and a thread block can be of 1, 2, or 3 dimension as shown in Fig.3. A grid can have up to 65535 blocks of threads in each dimension and a thread block can support 768 or 1024 threads depending on GPU architecture. Each thread on a block and a block on grid has a unique identification.

4. LITERATURE REVIEW

Present two efficient Apriori implementations of Frequent Item set Mining (FIM) that utilize new-generation graphics processing units (GPUs). Here implementations take advantage of the GPU's massively multi-threaded SIMD (Single Instruction, Multiple Data) architecture. One implementation runs entirely on the GPU and eliminates intermediate data transfer between the GPU memory and the CPU memory. The other implementation employs both the GPU and the CPU for processing. It represents item sets in a trie, and uses the CPU for trie traversing and incremental maintenance. Here preliminary results show that both implementations achieve a speedup of up to two orders of magnitude over optimized CPU

Apriori implementations on a PC with an NVIDIA GTX 280 GPU and a quad-core CPU. So in future we also plan to explore other mining algorithm with GPU acceleration [5].

A frequently used method of clustering is a technique called k-means clustering. The k-means algorithm consists of two steps: A map step, which is simple to execute on a GPU, and a reduce Step, which is more problematic. Here provide experimental comparisons using OpenCL. Here compare the GPU code with the standard OpenMP benchmark, MineBench. In that implementation, both the map and reduce steps are computed on the CPU. For large data sizes, the new GPU scheme shows great promise, with performance up to 35 times faster than MineBench on a four core Intel i7 CPU. The performance results for OpenCL are likely to improve through OpenCL tools. Here we have developed a very high speed-implementation of k-means. Given that the CPU is doing very little when both the map and reduce executes on the GPU, one possible future research direction would be to use the CPU to process part of the computation. Another potential area of research is implementing k-means on multiple GPUs. Since we have eliminated most of the memory copy overhead, we should get almost linear performance improvement as we increase the number of GPUs [6].

Scalable clustering using Graphics Processors. Basic approach is based on k-means, but it reorders the way of determining object labels, and exploits the high computational power and pipeline of graphics processing units (GPUs). Here implement algorithm in a PC with a Pentium IV 3.4G CPU and a NVIDIA GeForce 6800 GT graphics card. Comprehensive performance study shows that the common GPU in desktop computers could be an efficient co-processor of CPU in traditional and data stream clustering [7].

Using a popular clustering algorithm, K-Means, as an example, result have been very positive. On a data set with a billion data points, GPU-accelerated implementation achieved an order of magnitude performance gain over a highly optimized CPU-only version running on 8 cores, and more than two orders of magnitude gain over a popular benchmark, MineBench, running on a single core. Note that GPU is still evolving at a fast pace. The latest GPU from Nvidia, GTX 295, has doubled the capacity of the GTX 280. More than one GPU board can be installed into same machine. These can translate into greater performance potential, and potentially enable us to tackle even bigger problems [8].

Modern GPUs offer much computing power at a very modest cost. Even though CUDA and other related recent developments are accelerating the use of GPUs for general purpose applications, several challenges still remain in programming the GPUs. Thus, it is clearly desirable to be able to program GPUs using a higher-level interface. Here use program analysis and code generation to map the applications to a GPU. Several additional optimizations are also performed by the system. Here evaluated system using three popular data mining applications, k-means clustering, EM clustering, and Principal Component Analysis (PCA). The main observations of experiments are as follows. The speedup that each of these applications achieves over a sequential CPU version ranges between 20 and 50. the automatically generated version did not have any noticeable overheads compared to hand written codes. Finally, the optimization performed in the system resulted in significant performance improvements [9].

K-mean clustering has been widely used in processing large datasets in many fields of studies. Using General Purpose processor (GPP) to process large datasets may take a long time, therefore many acceleration methods have been proposed in the literature to speed-up the processing of such large datasets. Here propose a parameterized Field Programmable Gate Array (FPGA) implementation of the K-means algorithm and compare it with previous FPGA implementation as well as recent implementation on Graphics processing Units (GPUs) and with GPPs. So this implementation has shown higher performance in terms of speed-up over previous GPU and GPP implementations, and is more energy efficient [10].

Graphics processing units (GPUs) are powerful computational devices tailored towards the needs of the 3-D gaming industry for high-performance, real-time graphics engines. Nvidia Corporation provides a programming language called CUDA for general-purpose GPU programming. Hierarchical clustering is a common method used to determine clusters of similar data points in multidimensional spaces; if there are n data points, it can be computed in $O(n^2)$ to $O(n^2 \log n)$ sequential time, depending on the distance metrics employed. The present work explores parallel computation of hierarchical clustering with CUDA/GPU, and obtains an overall speed-up of up to 48 times over sequential computation with an Intel Pentium CPU. Further work is needed to write CUDA/GPU code for other linkage methods, and it will benefit the reserach community if fast CUDA/GPU hierarchical clustering is made callable from popular data analysis platforms [11].

Markov clustering (MCL) is becoming a key algorithm within bioinformatics for determining clusters in networks. However, with increasing vast amount of data on biological networks, performance and scalability issues are becoming a critical limiting factor in applications. Meanwhile, GPU computing, which uses CUDA tool for implementing a massively parallel computing environment in the GPU card, is becoming a very powerful, efficient, and low-cost option to achieve substantial performance gains over

CPU approaches. The use of on-chip memory on the GPU is efficiently lowering the latency time, thus, circumventing a major issue in other parallel computing environments, such as MPI. Here introduce a very fast Markov clustering algorithm using CUDA (CUDA-MCL) to perform parallel sparse matrix-matrix computations and parallel sparse Markov matrix normalizations, which are at the heart of MCL. Here utilized ELLPACK-R sparse format to allow the effective and fine-grain massively parallel processing to cope with the sparse nature of interaction networks data sets in bioinformatics applications. As the results show, CUDA-MCL is significantly faster than the original MCL running on CPU. Thus, large-scale parallel computation on off-the-shelf desktop-machines, that were previously only possible on supercomputing architectures, can significantly change the way bioinformaticians and biologists deal with their data.

Extension of CUDA-MCL capability, here considered hybrid CUDA and OpenMP implementations (hybrid CUDA/OpenMP) which enable the exploitation of multicore CPU and many-core GPUs in multi-GPU cards using OpenMP and CUDA, respectively. OpenMP might be used to manage each thread in a multicore CPU to split the parallel task locally into each GPU unit in multi-GPUs. Furthermore, CUDA could be employed in all GPU units to exploit the power of their massively parallel threads to do each local task in each GPU unit in parallel.

This approach will give us more accessibility, computing power, and capacity with more cores and more VRAM from all of GPUs cards, all locally in a single desktop machine. These CUDA/OpenMP approaches can be considered as a cheaper, more scalable, efficient hybrid method to solve the large data set and memory problems of traditional hybrid OpenMP/MPI in supercomputers or distributed computing machines [12].

5. CONCLUSION AND FUTURE WORK

- As it has been Clustering Algorithm giving best performance on small data set so we will implement it on large data set and different GPU platform and optimize the Performance.
- After Observation Parallel Computing is most important mode of Performance Optimization.
- For high Performance we will implement Many Clustering Algorithm on Different Platform of GPU and find out which platform is best according to Application.

6. REFERENCES

- [1] Pradeep Rai, Shubha Singh, "A Survey of Clustering Techniques" in proceedings of the International journal of Computer Applications (0975-8887) Volume 7-No.12, October 2010.
- [2] NVIDIA Corporation, NVIDIA Programming Guide, Version 3.1.1, July 2010.
- [3] Osama Abu Abbas, "Comparison Between Data Clustering Algorithm" in proceedings of the International Arab Journal of Information Technology, Vol. 5, No. 3, July 2008.
- [4] A.K. Jain, M.N. Murty, P.J. Flynn, "Data Clustering: A Review" ACM computing Surveys, Vol. 31, No. 3, September 1999.
- [5] Wenbin Fang, Milan Lu, Xiangye Xiao, Bingsheng He, Qiong Luo, "Frequent Itemset Mining on Graphics Processors" in proceeding of the Fifth International Workshop on Data management on New Hardware, June 28, 2009.
- [6] Balaji Dhanasekaran, Norman Rubin, "A new method for GPU based Irregular Reductions and Its Application to K-Means Clustering" in proceedings of the International Conference on GPGPU-4, March 2011.
- [7] Feng Cao, A K.H. Tung, A Zhou, "Scalable Clustering Using Graphics Processors" School of Computing, National University of Singapore, 2004.
- [8] Ren Wu, Bin Zhang, Meichun Hsu, "Clustering Billions of Data Points Using GPUs" UCHPC-MAW'09, May 2009, Ischia, Italy.
- [9] Wenjing Ma, Gagan Agrawal, "A Translation System for Enabling Data Mining Applications On GPUs", ICS'09, June 2009, New York, USA, pp. 400-409.
- [10] Hanaa M. Hussain, K Benkrid, A Erdogan, H Seker, "Highly Parameterized K-means Clustering on FPGAs: Comparative Results with GPPs and GPUs", in proceeding of the International Conference on Reconfigurable Computing and FPGAs, IEEE 2011.
- [11] Dar-Jen Chang, M Kantardzic, M Ouyang, " Hierarchical clustering with CUDA/GPU", UCHPC-MAW, 2010.
- [12] Alhadi Bustamam, Kevin Burrage, Nicholas A. Hamilton, "Fast Parallel Markov Clustering in Bioinformatics Using Massively Parallel Computing on GPU with CUDA and ELLPACK-R Sparse Format", in proceeding of the IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, VOL. 9, NO. 3, MAY/JUNE 2012.

Publish Research Article International Level Multidisciplinary Research Journal For All Subjects

Dear Sir/Mam,

We invite unpublished research paper.Summary of Research Project,Theses,Books and Books Review of publication,you will be pleased to know that our journals are

Associated and Indexed,India

- ★ International Scientific Journal Consortium Scientific
- ★ OPEN J-GATE

Associated and Indexed,USA

- EBSCO
- Index Copernicus
- Publication Index
- Academic Journal Database
- Contemporary Research Index
- Academic Paper Databse
- Digital Journals Database
- Current Index to Scholarly Journals
- Elite Scientific Journal Archive
- Directory Of Academic Resources
- Scholar Journal Index
- Recent Science Index
- Scientific Resources Database

Golden Research Thoughts
258/34 Raviwar Peth Solapur-413005,Maharashtra
Contact-9595359435
E-Mail-ayisrj@yahoo.in/ayisrj2011@gmail.com
Website : www.isrj.net