Parallel Data Mining on Graphics Processing Unit with CUDA

Vandana Purohit[1] Piyush Raut[2] Sharda Reddy[3] Rishikesh Yadav[4] Prof. Yashwant Dongre[5]
[1,2,3,4]Student [5]Assistant Professor
[1,2,3,4,5]Department of Computer Engineering
[1,2,3,4,5]VIIT, PUNE India

Abstract— The traditional data mining algorithms work in sequential manner which increases their time of execution. These algorithms should use the parallel processing capabilities of the modern GPUs to execute parallel programs efficiently. Therefore a parallel data mining algorithm should be implemented that can utilize the processing power of GPUs to speed up the execution.

Key words: CUDA, Graphics Processing Unit

I. INTRODUCTION

The traditional data mining algorithms work in sequential manner which increases their time of execution. With serial processing in multicore systems, only one core does pro-cessing while other cores remain idle. A sequential data mining algorithm handling large data sets would potentially take a large amount of time. Data mining is the process of finding patterns in large databases and is also known as knowledge discovery.

Now-a-days, as computations and datasets are growing, fast algorithms are gaining importance. In recent years, the information available is growly tremendously. Hence the data mining algorithms should use the parallel processing capabilities for execution of parallel programs in an efficient manner. More memory is required due to use of multiple processors. To use the resources to their fullest parallel computing techniques are essential. The motivation behind usage of parallelism is low power as well as low consumption. It also provides faster execution when compared to serial techniques. Using parallel environment these data mining algorithms can be used for pattern discovery.

To speed up the execution, these parallel algorithms use the processing power of GPUs (Graphics Processing Unit).

The advantages of parallel data mining are parallel processing of data, less execution time and better resource utilization. This shows that on many domains parallel techniques are strong.

For practical implementation, we are using NVidia CUDA GeForce GTX series GPU with Cuda Toolkit 7.5 installed on Ubuntu 14.04[6][7]. The Nvidia GeForce GTX series contains a minimum of 640 cores and a maximum of 2048 cores.

We are using PostGre SQL[12] database as our primary database with PGStrom extension for GPU support. The algorithm which we would be parallelizing is the most widely used clustering algorithm i.e. K-Means clustering algorithm.

The Nvidia CUDA GPU[5] architecture is explained in section 2.

The PGStrom extension of the PostGre SQL is explained in section 4.5.

The CUDA implementation of the algorithm is explained in section 6.

II. NVIDIA CUDA ARCHITECTURE[5]

CUDA is an Application Program Interface (API) created by NVIDIA which provides a platform for parallel computing. It allows general purpose computing on Graphics Processing Unit (GPU). CUDA gives access to parallel computational elements and the virtual instruction set. It also has a unified virtual memory. CUDA can work with programming languages like C, C++ and Fortran. CUDA is compatible with all standard operating systems.

GPU is a specialized processor which works on high resolution tasks like 3D graphics. GPU allows manipulation of large block of data faster than CPU as GPU is evolution of parallel multicore systems. GPU architecture hides latency from computation.

A CUDA application will run serial code on host i.e CPU while the parallel code on GPU threads through multiple processing elements. GPU executes parallel portion as Kernel. Kernel is a function executed on GPU on request of host (CPU) as an array of threads which executes in parallel through different paths. Kernel is executed in a fashion of grid of block of threads. Block is collection of Threads and Grid is collection of Blocks.

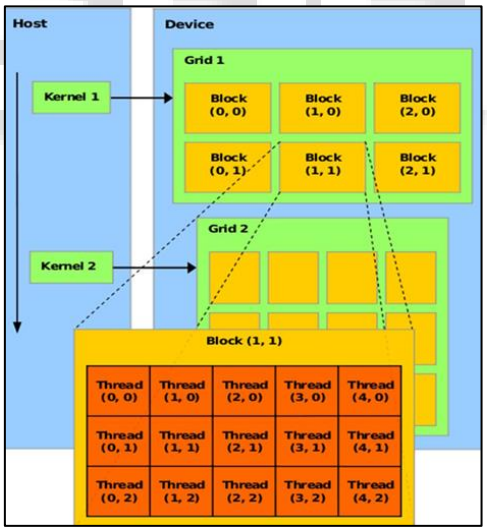

Fig. 1: Nvidia Cuda Architecture

III. DATA MINING TECHNIQUES

Data mining is the process of discovering pat-terns by analyzing information through differ-ent perspectives. The discovered knowledge is then used for generating revenue. It is used to find correlations among various large fields especially relational databases.

Data mining techniques includes association rule mining, clustering etc. Many algorithms have been developed like K-Nearest Neighbour classifier, Nave Bayesian classifier, FP Growth, Apriori, K-means for data mining purposes.

A. Parallel Data Mining[2][3]

When Data mining tools or algorithm implementations are done using parallel computers, it results into high performance computing which can analyze massive data in short time. Along with faster computations, complex data can be analyzed which would be in a greater quantity and thereby would provide improved results.

Various challenges are to be taken into consideration for parallel data mining like Communication, Synchronization and Data De-composition. If ignored, these can degrade the quality of data mining results. Dynamic load balancing is the important factor to be considered as parallel database servers has transient loads and multiple users.

B. K-Means Algorithm[1]

Clustering is the process of grouping the object having similarity into clusters and dissimilar objects are grouped in to a another cluster.

It is an supervised learning algorithm. K de-fines number of clusters. K centroids are generated for k clusters. Objects which are nearer to the centroid are grouped together.

Then again a new centroid is generated is and same is repeated until no object is moving from one cluster to another where two consecutive steps are generating the same result.

Calculation of pairwise distance is an independent task so it can be parallelized.

But all the objects have to wait for centroid updation until all the pair distances are been calculated.

IV. GPU ACCELERATED DATABASES

Today there are a plethora of popular and functional databases to choose from (ex. Oracle, MYSQL, MongoDB etc.) while starting a fresh database management project.

All these databases provide exceptional functionality in various fields like RDBMS/NoSQL, ACID properties, Maintaining redundancy, Preserving dependencies, Providing security, Providing a rich and strong query language and so on.

However, in todays era of parallelism, some of these databases lack the need of providing parallelism for execution of database operations.

Massive parallelism can be achieved if GPUs are used in conjunction with databases to accelerate the operations and save precious time which may later be used for any other computation.

There are a few GPU accelerated databases available in the market which use the massive computation power of the GPU for general purpose database operations and speedup the execution in comparison to traditional databases.

Some of these GPU Accelerated Databases are:

A. MapD Core[8]

MadD Core is a recent startup which uses GPUs to provide high speed acceleration on SQL queries. It is an In-Memory RDBMS solution that compiles the queries using LLVM compiler.

MapD claims to execute billions of records in merely milliseconds by using standard SQL.

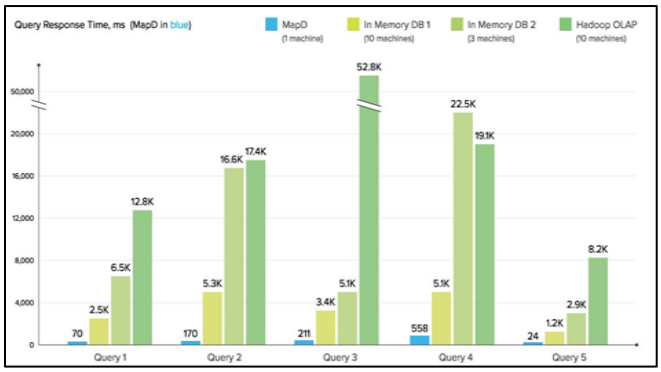

Fig. 2: MapD

B. SQream[9]

SQReam Database, released in 2014, is a solution from SQReam Technologies which is a database that specifically utilizes NVidia CUDA GPUs to accelerate SQL queries. SQReam Technologies claim to provide 100 times speedup execution on terabytes or petabytes of datasets.

C. Kinetica[10]

1) Legacy Name: GPU-DB

Kinetica DB is an in-memory database product from Kinetica Inc. which uses GPU processing to enable faster and flexible OLAP operations and was developed to meet the requirements of the US Army.

Kinetica DB claims to decrease analytical processing time for more than billions of records by more than 100 times compared to the recent in memory databases.

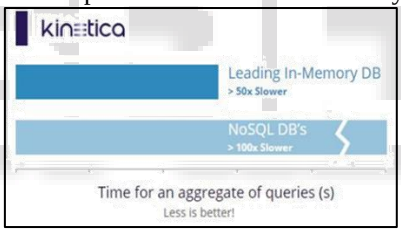

Fig. 3: Kinetica DB

D. Blazing DB[11]

Blazing DB is a platform which enables the users to run huge processes and jobs through Python, R, and SQL on super-charged GPU servers.

BlazingDB is an high performance SQL database written in C/C++ able to handle petabyte scale.

	Blazing	Redshift	PostgreSQL	MySQL
Join	✔	30x	40x	350x
Aggregations*	✔	5x	80x	140x
Predicates	✔	2.5x	36x	130x

Fig. 4: BlazingDB

E. PGStrom: PostGre SQL[12][13][14]

PostGre SQL is one of the most popular open source database management systems available in the market.

Like any traditional RDBMS system, it does not support GPU acceleration but it is possible to add GPU acceleration to this database.

PG-Strom is an additional extension for PostGre SQL database which is designed to utilize NVidia CUDA GPUs massive parallelism capability to perform intensive database operations.

PG-Strom uses JIT compiler to decide whether a query can be successfully paralleized and executed on GPU.

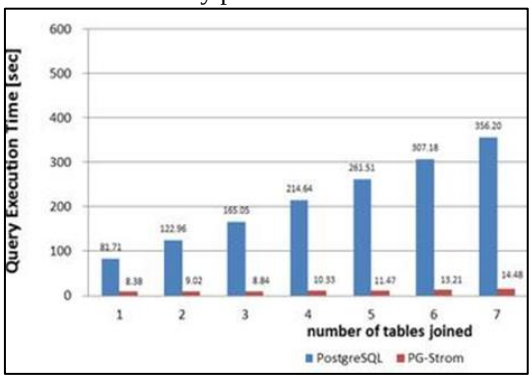

Fig. 5: PG-Strom

V. ALGORITHMS TO BE PARALLELIZED

There are many data mining algorithms avail-able today which provide accurate results de-pending on the needs of the users.

Few of these algorithms are widely used in the industry sectors and also have a great scope for parallelism. These algorithms have been successfully parallelized and implemented on multi-processor CPU architectures.

With the limitation of the number of avail-able cores, the multi-processor architecture turns out as a bottleneck for the parallelized algorithms limiting the degree of parallelism that may be achieved.

To reduce this bottleneck, these algorithms can be implemented on GPUs and utilize thou-sands of cores available on the GPUs resulting in the massive degree of parallelism.

NVidia CUDA GPUs provide GPU access to general purpose computing and algorithms can be implemented on CUDA rather than multiprocessor CPUs. This will result into a highly parallelized and functional data mining algorithm and better resource utilization of the underlying hardware on the system.

The algorithms which are widely used and have a great scope of parallelism are [1]

- K-Means Clustering KNN Algorithm
- Apriori Algorithm

VI. CU-K-MEANS ALGORITHM[1]

Clustering is a process of grouping the data items into clusters so that data items within the same clusters are similar to each other and data items in different clusters are dissimilar to each other.

K-Means Clustering works purely on the basis of Euclidian Distance between the data items. The Euclidian distance of all the data items is calculated and the data item is sent to the cluster whose centroid is closest to the data item.

The distance calculation, centroid update and centroid movement detection are the major tasks which consume most of the time of the serial algorithm and these tasks can be parallelized successfully.

The distance calculation of one data item is independent of the distance calculation of the other data item. So the task of Distance Calculation can be massively parallelized for every data item with the use of GPU cores and threads.

Next, as clusters are an independent entity in itself, the centroid calculation can be done parallely by taking the average of each data item within a cluster. This task also has a high scope of parallelism and can be easily implemented on the GPU.

Lastly, the newly calculated centroids have to be compared to the old centroids to detect centroid movement if any. This task can also be performed parallely as the centroids of each cluster are independent of each other.

The algorithm finally terminates when there is not centroid movement detected within the clusters.

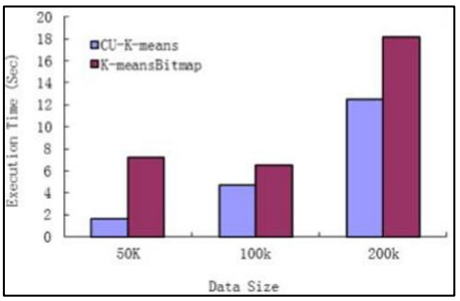

Fig. 6: K-Means Clustering

VII. CONCLUSION

The paper focuses on the parallelization of data mining techniques. For different data mining applications, GPU with CUDA provides us benefits for parallelization.

The open-source PostgreSQL database is used for this purpose. It took around 120 seconds for execution of two tables joined together using this database. The query execution time increased roughly by 40 seconds each time a new table was joined. For three tables joined it took approximately 160 sec, for four 220 sec and so on.

Thus applications that used distributed environment or supercomputers can be now solved using a single desktop having NVIDIA graphics card for parallel computing using CUDA.

REFERENCES

[1] Parallel data mining techniques on Graphics Processing Unit with Compute Unified Device Architecture(CUDA) LihengJian ChengWang YingLiu ShenshenLiang Weidong YiYongShi Published online: 26 August 2011
[2] A highly efficient multi-core algorithm for clustering extremely large datasets. JohannMKraus HansAKestler
[3] Paralle Data Mining on Multicore Clusters XiaohongQiu1 GeoffreyFox Huapeng Yuan Seung HeeBae2
[4] NVIDIA (2008) CUDA programming guide 2.1. http://www.nvidia.com/object/cuda develop.html
[5] CUDA C Programming Guide https://docs.nvidia.com/cuda/cuda-c-programming-guide/
[6] CUDA Tooklit 7.5 https://developer.nvidia.com/cuda-75-downloads-archive
[7] Installing CUDA 7.5 with Ubuntu 16.04 or Ubuntu 14.04 http://kislayabhi.github.io/Installing CUDA with Ubuntu/
[8] MapD - The world's fastest GPU database and visual analytics platform. https://www.mapd.com/
[9] SQream Technologies - GPU based SQL database http://sqream.com/solutions/products/sqream-db/

[10] Kinetica - Fast, scalable and proven GPU database for massive datasets. http://www.kinetica.com/product/
[11] BlazingDB: High Performance GPU Database for Big Data SQL. https://blazingdb.com/
[12] PostgreSQL: The world's most advanced open source database. https://www.postgresql.org/
[13] PostgreSQL tutorial - TutorialsPoint http://www.tutorialspoint.com/postgresql/
[14] PGStrom - PostgreSQL wiki https://wiki.postgresql.org/wiki/PGStrom

