ACCELERATING SQL QUERIES USING GPU

Krutika Paradkar[1] | Rucha Borane[2] | Pallavee Deshmukh[3] | Akshay Kale[4]
Ashwini Bhugul[5] | Vandana Rupnar[6]

[1](Computer Department, SPPU, Pune, India, krutikaparadkar.comp@mmcoe.edu.in)
[2](Computer Department, SPPU, Pune, India, ruchaborane.comp@mmcoe.edu.in)
[3](Computer Department, SPPU, Pune, India, pallaveedeshmukh.comp@mmcoe.edu.in)
[4](Computer Department, SPPU, Pune, India, akshaykale.comp@mmcoe.edu.in)
[5](Computer Department, SPPU, Pune, India, ashwinibhugul@mmcoe.edu.in)
[6](Computer Department, SPPU, Pune, India, vandanarupnar@mmcoe.edu.in)

***Abstract*— *Day by day there is a rapid growth in the generation of data. So, to handle such a huge amount of data there is a need of some effective techniques which avail to store, retrieve and analyze such massive data. Graphics Processing Unit (GPU) were conventionally utilized to optimize image filtering and video processing but it additionally avails to handle critical business applications where the computational power of GPU can provide significant benefits. In this paper, we propose the approach of analyzing the speed up of query execution. In GPU, the data is divided into smaller parts of data with help of threads of the GPU, SQL query is fired parallelly. The proposed approach is implemented on a SQLite database using the CUDA framework. The execution time of both CPU and GPU is analyzed.*

Keywords*— *Parallel Database; Parallel Programming; Distributed Database; Database Management

1. INTRODUCTION

GPU has proved to be an efficient co-processor in the field of conventional computer. In this advancing world of computation the demand for high speed processing is increasing day by day. Web based application and system which uses database on back end required high speed processing to access the data over databases. Most of the top supercomputers in the world contain GPUs to increase performance. Research in applying GPUs to solve parallel problems has been done on numerous applications. The acceleration of SQL queries would enable programmers to do little or no change to their source code.

In database system the computational power of GPU can provide significant benefit to critical business applications for instance Bank's Transaction Process. A database system performs a significant amount of repeated calculations on different data. Traditional databases used to perform the steps of a query sequentially using the CPU. CPU has less but more powerful cores and GPU has more but less powerful cores. Though parallelism is possible in CPUs by using multiple cores, but attempting to achieve greater parallelism requires the addition of costly additional CPUs, and communication between large numbers of CPUs is problematic. A GPU database is feasible due to the considerable inbuilt parallelism in the way databases operate. A database query consists of a set of operations that are performed on the rows of one or more tables. Each of these operations is repeated, potentially once for each row in the table. Within the query, the execution of these operations is largely data independent; the execution of operations on one row usually does not affect the operations being executed on another row. This introduces to a large amount of data parallelism. Each set of operations could be executed simultaneously on every row.

The proposed approach has been conducted on an NVIDIA GeForce GTX TITAN X which is an ultimate graphics card. It is a combination of the latest technologies and performance of the new NVIDIA Maxwell™ architecture. It is has 3072 CUDA cores inside and 12 GB memory.

2. LITERATURE REVIEW

There has been a lot of development in the past few years in the field of general-purpose graphics processing unit databases. In 2010, Peter Bakkum implemented SQLite database management system on a GPU using CUDA framework. In his work he demonstrated core principles of GPU database implementation with the avail of a database capable of performing SELECT queries on integer values. Also, it uses GPU-mapped memory with efficient caching. Consequently, it can compute a larger size of data which exceeds GPU physical memory size.

Glen's explored new ways of handling complex data types and managing data and workloads. Their work focused on three critical areas for developing a database system on a GPU: data caching to manage the data on the GPU, processing table joins and managing the resultant workload on the GPU, and handling irregular and complex data types.

In 2015, Patta's included a query parser which accepts SQL queries in the form of input and the output will be in the form of intermediate code or opcode. They used Virtual Machine to execute the opcode program, to process data records and give a result set as output. Also, they used the Table Functionality for managing groups of tablets in tables. The speed comparison between the GPU and CPU was done with the help of a SQL profiler.

In 2017, Shehab et al proposed a new hybrid query processing technique that makes use of capabilities of CPUs and GPUs where they proposed approach which breaks down each SQL statement into smaller parts during the parsing process. Then in which it automatically manages the distribution of different query parts to be executed either on the CPU only on the GPU co-processor only or on both CPU and GPU. Their main focus was on how to automatically decide which part of the query should be executed on CPUs and which others should be executed on the GPU co-processor for that they demonstrated how to control and dispatch the query parts making a decision about which query part should be executed on the CPU and which should be executed on the GPU co-processor.

3. PROPOSED SYSTEM

The proposed approach used in this paper is used to analyze the speed up of query execution. In this system, the large data is taken for processing. The SQL query is then executed first on CPU. The execution time is calculated for the CPU. The same data is used with the GPU. In GPU, the data is divided into small parts of data. On each part of data with the help of threads of the GPU, SQL query is fired parallelly. The result of each thread is stored in a file. The execution time of both CPU and GPU is analyzed.

In this proposed approach we present the implementation of SQL operations using CUDA framework on the GPU platform. A programming language like CUDA helps to directly interact with the GPU. We propose the optimization of selection queries and aggregation functions with where-clause operations. Using the CUDA programming the data is stored on the GPU in row-column

Fig. 1. NVIDIA Titan X Card

The NVIDIA GeForce GTX TITAN X card is used which is an ultimate graphics card which combines the latest technologies and performance of the new NVIDIA Maxwell™ architecture. It has 3072 CUDA cores and 12 GB memory with memory bandwidth of 336.5 GB/sec and memory type of GDDR5 SDRAM placed on the card

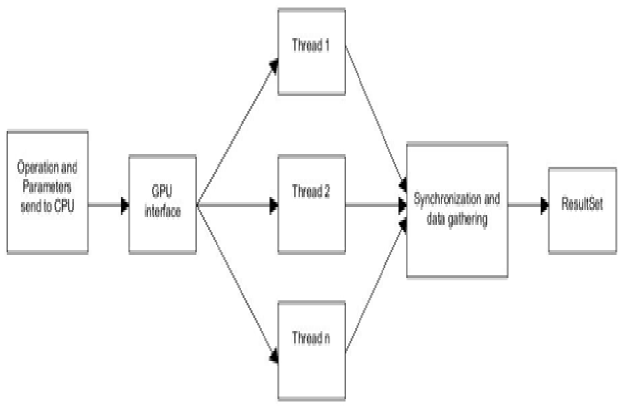
Operation and Parameters send to CPU
GPU interface
Thread 1
Thread 2
Thread n
Synchronization and data gathering
ResultSet

Fig. 2. Overview of Execution Flow

4. ADVANTAGES

The main objective is to retrieve data quickly. Most of the databases are built and filled with data due to which leads to slow speed. The time taken for executing a query results in exponential growth as the amount of data increases in the database leading to more waiting times on the user, as well as on the application side. Sometimes the wait time could range from minutes, to hours, and days as well in worst cases. Query optimization is used in database administration. The Queries allow the user to extract significant information from a database. Instead of scanning the whole table, they can predefine the categories of information which will be retrieved.

5. CONCLUSION

The SQLite database was used as a platform for executing the queries. SQL parsing mechanism and switching between CPU and GPU execution was useful and enhancing. Execution on the GPU was supported by re-implementing the SQLite virtual machine as a CUDA kernel. In this paper database queries are offloaded on the GPU to achieve effective acceleration of database operations. In the upcoming years the size of NVIDIA GPUs will increase and it will be able to handle large amount of data and performance will also increase. Insert, Update which are purely handled by CPU can also be handle by GPU to enhance the performance and get results in less time.

REFERENCES

[1] Esraa Shehab, Alsayed Algergaway, Amany Sarhan "Accelerating relational database operations using both CPU and GPU co-processor

[2] Rajendra A. patta, Anuraj R. Kurup, Sandip M.Walunj "Enhancing Speed of SQL Database Operations using GPU" International Conference on Pervasive Computing:IEEE-978-1-4799-6272-3

[3] Kalle Karkkainen "Parallelism in Database Operation" Helsinki University October 2, 2012

[4] Peter Bakkum and Kevin Skadron "Accelerating SQL Database Operation on GPU with CUDA" Department of Computer Science University of Virginia,Charlottesville,VA22904, March 14,2010

[5] Glen Hordemann, Jong Kwan Lee, Andries H. Smith \Accelerated SQLite Database using GPUs", Department of Computer Science and Engineering Texas AM University, 2010