A Survey on Query Processing With Graphic Processing In CUDA

Payal Bhatt
M.E. Student
Department of Computer Engineering
BHGCET, Rajkot
G.T.U. Gujarat, India

Aakansha Saxena
Assistant Professor
Department of Computer Engineering
BHGCET, Rajkot
G.T.U. Gujarat, India

ABSTRACT

Initially introduced as special-purpose accelerators for graphics applications, GPUs have now emerged as general purpose computing platforms for a wide range of applications. To address the requirements of these applications, modern GPUs include sizable hardware- managed caches. However, several factors, such as unique architecture of GPU, rise of

CPU-GPU heterogeneous computing etc., demand effective management of caches to achieve high performance and energy efficiency. Recently, several techniques have been proposed for this purpose. The main aim of this survey is to an image of various techniques that can be used in GPU for general purpose programming, henceforth to improve the overall performance of any simple programming. This paper also states Challenges and applications of GPU.

Keywords

Query processing, CUDA, GPU(General Purpose Programming).

1. INTRODUCTION

Structured Query Language (SQL)

Firstly let us know what exactly SQL query processing is? Basically, **Structured Query Language** is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS). It is a process of taking the SQL statements we write and converting them into something that can make requests to the storage engine and retrieve the results needed. Basically Query processing translates high level query (relational) into equivalent lower level query. Processing a single SQL statement is the most basic way that SQL server executes the SQL statement.

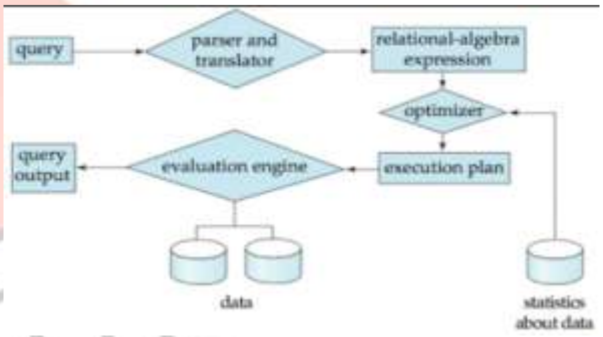
SQL Processing is the parsing, optimization, row source generation, and execution of a SQL statement. Out of all these four steps of query processing the 'heart' of the entire processing is Query Optimization. So we aim here to focus on the core part of the SQL processing.

Steps for Query processing[1]:

1.Parsing andtranslation

2.Optimization

3. Evaluation



Steps for Query Processing[1]

2. General Purpose Computing with Graphics Processing Unit

Let us now move towards introducing a platform for performing the general purpose tasks in Graphic Processing Unit. The platform for performing our SQL queries is CUDA [COMPUTE UNIFIED DEVICE ARCHITECTURE].

CUDA is a platform for performing massively parallel computations on graphics accelerators. It allows software developers to use a CUDA- enabled graphics processing unit (GPU) for general purpose processing – an approach known as GPGPU.

CUDA was developed by NVIDIA. CUDA presents a unique opportunity to develop widely-deployed parallel applications. The current CUDA GPU Architecture is branded Te

In early days of computing, the central processing unit (CPU) performed the calculations. As more graphics- intensive applications were developed, however, their demands put strain on the CPU and degraded performance. The GPU came about as a way to offload those tasks from the CPU, freeing up its processing power. A graphics processing unit(GPU) is a computer chip that performs rapid mathematical calculations, primarily for the purpose of rendering images more quickly because of its parallel processing architecture, which allows it to perform multiple calculations at the same time. The resulting performance improvements have made GPU's popular chips for other resource- intensive tasks unrelated to graphics.

In the last _few years, two factors stimulated a renewed attention respectively in dis-tributed and parallel methods for data mining, and in particular for high performance Frequent Pattern Mining methods. The _rst is the wide availability of commercial, Elastic and Distributed computing facilities, such as the Amazon Elastic Compute Cloud; the second is the Microprocessors with increasing number of cores availabili-ty, for example the recent NVIDIA Kepler microprocessor, the one used in GeForce GTX 770 and GeForce GTX 780 cards. Features 2880 core with 6GB total Global Memory[3]. These innovations open new scalable opportunities on the one hand and demand for further care to handle their peculiarities on the other. Thus, to selctively utilize these opportunities, there is a need for a new generation and modi_cation of the existing data mining algorithms, it is suited for coherent and predictable mem-ory access, in able to exploit this particular General- Purpose computing paradigm on Graphics Processing Units GPGPU [1]. This is due to the fundamentally di_eren-t architecture programming model and design of many core GPUs with respect to traditional multi-core CPUs.

CUDA is a massively parallel computing platform and as well as programming model developed by NVIDIA Corporation, increasing computing performance for par-allel problems with use of Graphics Processing Units. It also provides low and high level APIs (application programming interfaces) with debuggers, shared memory ac-cess, scattered reads and fast read backs. NVIDIA CUDA platform come with C programming extension and C++ integration too. Recently NVIDIA CUDA releases CUDA 6.0 which enable unifed memory, in other word developers are free from manually memory copy (CudaMemcpy) from host to device and visa versa. Now CUDA support such complex data structure like link list and tree.[4]

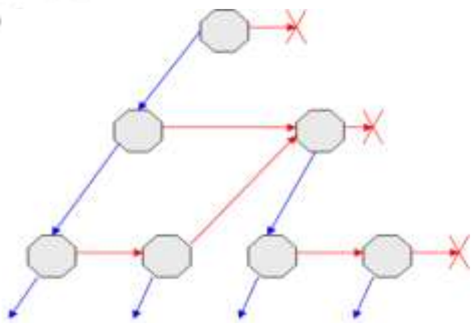
3. Techniques

3.1 Map Reduce Paradigm

A new and active area of data mining research is in the MapReduce paradigm. Originally pioneered by Google, it gives the programmer a new paradigm for data mining based on the functional primitives map and reduce [3]. This paradigm has a fundamentally parallel nature, and is used extensively by Google and many other companies for large scale distributed data processing. Though essentially just a name for using two of the primitives mentioned in the previous section, MapReduce has become a major topic itself. Re- search in this area has shown that MapReduce frameworks can be accelerated on multicore machines [16] and on GPUs

3.2 K-D tree

Functions:- Coarsening
Interpolation
Force Calculation



3.3 K Nearest Neighbor

First, all the distances from the target query to the elements on the database are evaluated and then a second stage sorts these distances to obtain the nearest elements.

3.4 Sparse Spatial Selection(SSS)

During construction, this pivot-based index [2] selects some objects as pivots from the collection and then computes the distance between these pivots and the rest of the database.
The result is a table of distances where columns are the pivots and rows the objects.
Each cell in the table contains the distance between the object and the respective pivot, These distances are used to solve queries.

4. R-tree algorithm

Traversing R-Trees on CPU needs a loop to iterate the array of minimum bounding boxes[17]of child tree nodes.
To exploit massively parallel CUDA architecture and to alleviate the I/O resource competition, this loop is parallelized with a number of threads.
In MPTS search algorithm, a set of threads in a CUDA block cooperate to check in parallel if the child nodes overlap a given query, i.e., the number of threads in each block is set equal to the number of node fan-outs (the maximum number of child nodes).

5. Prefix tree

The idea is that a tuple consists of a key Ki and a payload Pi, where the key Ki may be of any type of integer (32 or 64 bit) or a string with variable length.
A key Ki is split into N equally sized sub-keys Kn i consisting of b = jKij N bits with jKij being the number of bits in the key.
Each node contains an array of 2b child node pointers. Hence, Depth of the Tree can be known also the keys are found in sorted order and thus can be efficiently managed.

6. Speculative tree traversal

Our concept of speculative tree traversal consists of two steps:
1. Parallel traversal of (all) partitions of the tree, and
2. Aggregation of intermediate results to the final result.
The parallel traversal of the tree uses the high number of cores on a GPU.
In its initial implementation, all possible partial results are created.
Then, the intermediate results are used to determine the final result. The second step is, again, a traversal of the intermediate results, which is implemented as a serial traversal.

	Techniques	Reference Paper	Pros	Cons
1.	Map Reduce Pardigram	Accelerating SQL Database Operations on a GPU with CUDA[14], 2010	Reduces the effort, simpler and effective	Memory size, Tranfser time, lacks in implementation of Multi-core query
2.	K-D Tree	Rapid Multipole Graph Drawing on GPU[15], 2013	Impressive speedup over previous methods, also used in Web Designing, S/W Eng.	Load balancing, Median Finding, Band width
3.	KNN Search	KNN Query Processing in Metric Spaces using GPUs, 2014	Increasing in Efficiency	Memory lose
4.	SSS Index	kNN Query Processing in Metric Spaces using GPUs, 2014	Performance speed - up	Increasing Cost
5.	R-Tree Search Algorithm	Parallel Multi-dimensional Range Query Processing with R-Trees on GPU, 2013	Prunning out unnecessary nodes, Performance increase	Dimensionality Problem
6.	Prefix Tree Traversal	GPUBased Speculative Query Processing for Database Operations, 2011	Optimization	Limited Bandwidth and GPU memory
7.	Spetaculative Traversal	GPUBased Speculative Query Processing for Database operations, Operations, 2011	Minimization instruction to increase Efficiency	Limited Speed

7. APPLICATIONS

With the increase usage of internet, huge amount of opinionated data is available over various data resources. This data has increased the usage of general purpose programming. Some of its applications are as follows [4]:

- **Computational Geoscience:** The performance of geosciences numerical computations is often constrained by memory bandwidth. This makes programmable GPUs attractive, as they have the potential to deliver several times more memory bandwidth than a conventional CPU, for algorithms that contain enough inherent data parallelism.
- **Computational Medicine**: The medical scientific research to diagnostics and treatment planning. However, medical imaging procedures are often computationally demanding due to the large three-dimensional medical datasets to process in practical clinical applications. With the rapidly enhancing performances of graphics processor, improved programming support, and excellent price –to performance ratio.
- **Computational Finance:** The GPU accelerator offers financial service firms the ability to drive their business faster, with better analytics at lower cost. GPUs allow complex risk calculations at the trader level to run in seconds, allowing real-time risk to be business usual. Value at risk, counterparty risk and initial and lifetime margining calculations are some of the types of calculations benefiting from GPUs.
- **Computational Modelling :**
- **Frequency Estimation:** A good turing frequency estimation is a statiscal technique for estimating the probability of encountering an object, given a set of past observations of objects from different species.
- **Quantile Estimation:** Quantile regression is a type regression analysis used in statistics and econometrics.

8. CHALLENGES

●**Hybrid query processing:** A hybrid approach of mixing both cpu and gpu is one of the rising challenge. Hence the aim here is to speed up the query execution by combining cpu and gpu. Thus hybrid optimization can be achieved. A hybrid system is created and implemented by map reduce strategies.

●**Concurrent query processing with gpu's:** In current databases, gpus are used as dedicated accelerators to process each individual query. Gpus among concurrent queries for high throughput, the major challenge is concurrent query processing. A multiqx-gpu improves the throughput by running multiple queries at same time.

●**Forensics on gpu co processing in database:** Recently, using gpus for co processing in database system has shown to be beneficial. However, information system processing confidential data cannot benefit from gpu acceleration yet because knowledge of security issues and forensic examinations on gpus are still fragmentary. This can be done by creating a memory dump of device memory.

●**Gpu- based speculative query processing for database operations:** With an increasing amount of data and user demands for fast query processing, the optimization of database operations continues to be a challenging task. A common optimization method is to leverage parallel hardware has become available within commodity hardware. Here it is essential to develop pruning methods, to efficiently leverage the hardware for scalable problems.

●**Metadata challenge for query processing over heterogeneous wireless senor network:** Wireless sensor networks become integral part of our life. These networks can be used for monitoring the data in various domain due to their flexibility and functionality.

Query processing and optimization in the wireless sensor network is a very challenging task because of their energy and memory constraint, hence these can be overcome with developing query processing techniques for wireless sensor networks.

9.CONCLUSION

In this survey paper, we provide the overview graphic processing for simple queries and complex queries along with type of data sources are used in it. We have also listed various techniques, CUDA tools, its library functions, its API etc. Survey results show that the performance and speed using GPU are increased as compared to CPU for general purpose programming rather that special purpose programming.

10.REFERENCES

[1] silberschatz, korth and sudarshan @1997

[2] ramez elmasri and shankar b. Navathe

[3] computer unified device architecture) supercomputing for masses by peter zalutski

[4] amol deshpande, zachary ives and vijayshankar raman, "foundations and trendsr *in* databases" vol. 1, no. 1 (2007) 1–140 c 2007 a. Deshpande, z. Ives and v. Raman doi: 10.1561/1900000001

[5] mr.p.karthik1, prof.g.thippa reddy2, mr.e.kaari vanan3, "ijcsi international journal of computer science issues", vol. 9, issue 4, no 3, july 2012 issn (online): 1694-0814 www.ijcsi.org

[6] peter bakkum and kevin skadron, gpgpu-3 march 14, 2010, pittsburg, pa, usa.

[7] kaibo wang kai zhangz yuan yuan siyuan ma rubao lee, xiaoning dingy xiaodong zhang conference on "very large data bases", september 1st 5th 2014, hangzhou, china

[8] alpa jain #1, anhai doan 2, luis gravano, *department of computer sciences, university of wisconsin-madison*

[9] apeksha godiyal, jared hoberock, michael garlandy, and john c hart, 2007

[10] peter benjamin volk, dirk habich, wolfgang lehner, the original idea was submitted to the "first annual sigmod programming contest { main memory transactional index" by the dexter team.

[11] ricardo j. Barrientos, jos´e i. G´omez, christian tenllado, manuel prieto matias, and mauricio marin, 2011

[12] paramjeet kaur et al, / (ijcsit) international journal of computer science and information technologies, vol. 5 (2) , 2014, 2210-2214

[13] ruchin gupta, narendra teotia " *international journal of computer applications "(0975 – 8887) volume 84 – no 2, december 2013* .

[14] indu bala* ochin sharma aftab alam , volume 4, issue 9, september 2014 issn: 2277 128x "international journal of advanced research in computer science and software engineering".

[15] muhammad waqar, anas bilal, ambreen mallik, imran anwar, "*european journal of engineering and technology "vol. 4 no. 5, 2016*

[16] pooja khulbe , shruti pant, "*international journal of computer applications "(0975 – 8887) volume 91 – no.16, april 2014*

[17] k. Ravindra babu*, g. Pavan kumar volume 6, issue 1, january 2016 issn: 2277 128x "international journal of advanced research in computer science and software engineering"

[18] jinwoong kim, sumin hong, and beomseok nam. A performance study of traversing spatial indexing structures in parallel on gpu. In *3rd international workshop on frontier of gpu computing (in conjunction with hpcc), 2012.*