Implementation of Mathematical Series Functions on CUDA Platform-A preliminary result

Dr. Raj.B. Kulkarni, Madhavi P. Patil

Abstract— High Performance GPUs originally designed for graphics processing. But GPUs also offer small to very high calculations with CUDA, the multithreaded approach of compute unified device architecture. In this paper we evaluate efficiency of computation in many core GPU platform such Quadro FX 1800 from NVIDIA and compare the performance with multi-core CPU execution. It presents introductory concepts of parallel computing from simple examples to debugging both logical and performance.

Index Terms— Minimum GPU, GPGPU, High performance computing, CUDA.

1. INTRODUCTION

Now a days the performance of general-purpose processors growing more rapidly and became strong in parallel computing. Parallel computing is more demanding in 3D games. Devices with multi-core processors give high performance in parallel vector computing in 3D graphics. With CPUs there is a limit to keep it up. An example of such parallel processors is the modern graphics cards, or Graphical Processing Units (GPUs), from NVIDIA.

The heterogeneous systems will rely on the integration of two major types of components: CPU (Multi- and many-core technology) and GPU (Special-purpose hardware and massively parallel accelerators).

General purpose graphic processor unit (GPGPU) is architecture for high performance computing that uses graphics processing units (GPU) for multicore processing of data. GPGPU exhibit two properties such as data parallelism and intensive throughput of data. Data parallelism is nothing but processor can execute operations on different data elements simultaneously.

On the other hand, throughput intensive process means an algorithm is going to process lots of data elements whose execution will be in parallel. GPGPU platforms available from NVIDIA, ATI, and Intel have a large number of processors (of the order of a few hundred) structured to allow multiple threads of execution.

Architecture of GPGPU is organized into an array of highly threaded streaming multiprocessors (SMs). It has number of streaming processors (SPs) that share control logic and instruction cache. e.g., Fermi and Kepler series from NVIDIA shows that as the number of cores per SMs increases, core speed also gets increased. The number of cores in Tesla C2075 is 448 whereas Tesla K20C Kepler series has 2496 cores. Therefore the memory bandwidth also increases in advanced generation of GPGPUs.

Instead of using matrix and vector operations, author tried other examples to compare GPU with CPU. This paper mainly focused on mathematical functions. Mathematical series functions such as functions that are core of trigonometry sine, cosec, etc. are very useful. Program is written in such way that data dependency in C code was avoided in CUDA code. The paper mainly focused on to evaluate the efficiency of computation in many core GPU platforms such as Quadro FX 1800 from NVIDIA and compare the performance with multi-core CPU execution. The objective of this project is to prove that GPUs are more effective than CPUs.

1.1 Trigonometry functions - introduction

There are six functions that are the core of trigonometry. There are three primary functions and other 3 functions can be derived from the primary functions.

Primary Trigonometric Functions:

1. Sine (sin)
2. Cosine (cos)
3. Tangent (tan)

The derived functions from the three primary functions:

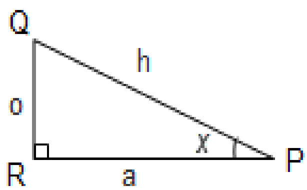
4. Secant (sec)
5. Cosecant (csc)
6. Cotangent (cot)

All six functions have three-letter abbreviations (shown in parentheses above). [4] Associate

Definitions of the six functions

- *Dr. R. B.Kulkarni,Associaet,Professor, Computer Sci. & Engi, WIT Solapur, Maharashtra, India, PH-+919822002072.*
  E-mail: raj_joy@yahoo.com
- *Madhavi P. Patil is currently pursuing master degree program in Computer Science and Egineering in WIT, Solapur University, Maharashtra, India, PH-+ 919545699315.*
  E-mail: madhavipatil1911@rediffmail.com

International Journal of Scientific & Engineering Research, Volume 6, Issue ħŸǜ . ÊÛÖÉÌÙ-2015
ISSN 2229-5518



- a - length of the side adjacent to the angle (x) in question.
- o - length of the side opposite the angle.
- h - the length of the hypotenuse.
- "x" represents the measure of the angle in either degrees or radians.
- P, Q and R are the angles of right triangle. [4]

Primary Trigonometric Functions	
Sine	$\sin x = \frac{o}{h}$
Cosec	$\cos x = \frac{a}{h}$
Tangent	$\tan x = \frac{o}{a}$
Derived Trigonometric Functions from Primary Functions	
Cosecant	$\csc x = \frac{1}{\sin x}$
Secant	$\sec x = \frac{1}{\cos x}$
Cotangent	$\cot x = \frac{1}{\tan x}$

1.2 Use of Trigonometric Series Functions [5]

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}$$

The most important applications of trigonometric series is the situation where there is a use of very small angles. E.g. calculating the period of a simple pendulum, and intensity minima in single slit diffraction. For such angles, only first term in the series can be approximated. This gives the useful small angle approximations:

Small angle approximations: $\sin x \approx x$ $\quad \cos x \approx 1$ $\quad \tan x = \frac{\sin x}{\cos x} \approx x$

1.3 Applications of Trigonometric Functions

The aim to develop trigonometric functions was for astronomy and geography, but then after scientists has been using it for other purposes also. Besides fields of mathematics, trigonometric function is used in physics, engineering, and chemistry. In mathematics, trigonometric is used in primarily in calculus, linear algebra, statistics, natural science and social science.

1.3.1 Astronomy and geography

Trigonometric tables were created over two thousand years ago for computations in astronomy. The kind of trigonometry needed to understand positions on a sphere is called spherical trigonometry. Now the job of Spherical trigonometry has been taken over by linear algebra and so it is rarely taught. However, one application of trigonometry is astronomy.

1.3.2 Engineering and physics

Even trigonometry was first useful to spheres; there are larger applications to planes. All engineers, even military engineers have used trigonometry since long time.
Trigonometry has heavy demands in Physics. In early days, two fields Optics and statics of physics use trigonometry, but now a day all branches of physics use trigonometry as trigonometry aids in understanding space and so related fields such as physical chemistry naturally use trigonometry.

1.3.3 Mathematics and its applications

Trigonometry is also used in mathematics, such as Calculus, linear algebra, statistics, etc. It has many applications all over the natural and social sciences. [2]
Following are some fields where trigonometry is use:
electrical engineering, electronics, acoustics, civil engineering, mechanical engineering, machining, probability theory, visual perception medical imaging , architecture, number theory, pharmacology,astronomy, cartography,geophysics, crystallography, land surveying and geodesy, many physical sciences, oceanography, optics, seismology, and statistics.[3]

2. BACKGROUND

The main purpose of a graphics processing unit (GPU) is for 3D or 2D graphics rendering by taking load from the microprocessor and frees it for another work. GPU is used in embedded systems, mobile phones, personal computers, workstations, and game consoles. In a personal computer, a GPU can be present on a video card, or it can be mounted on the motherboard.
The GPU is especially compatible to data-parallel computations in which the same program is executed on many data elements in parallel. There is a less flow control and high strength of arithmetic because the same program is executed for each data element.
To speed up the computations, applications that process large data sets can use a data-parallel programming model.
In 3D rendering, large sets of pixels and vertices are mapped to parallel threads. On the contrary, image processing and

International Journal of Scientific & Engineering Research, Volume 6, Issue ŀƔŀ . ÊÛÖÉÌÙ-2015
ISSN 2229-5518

media processing applications e.g video encoding and decoding, post-processing of rendered images, image scaling, stereo vision, pattern recognition etc. Such application can map image blocks and pixels to parallel processing threads. Algorithm other than image rendering and procseeing are speeded by data-parallel processing. Like modern CPUs, there are lots of arithmetic operations performed on GPU. More transistors are used in GPUs which process data arrays instead of flow control of several sequential computing threads. Figure 1 shows how much room is occupied by various circuits in CPUs and GPUs:

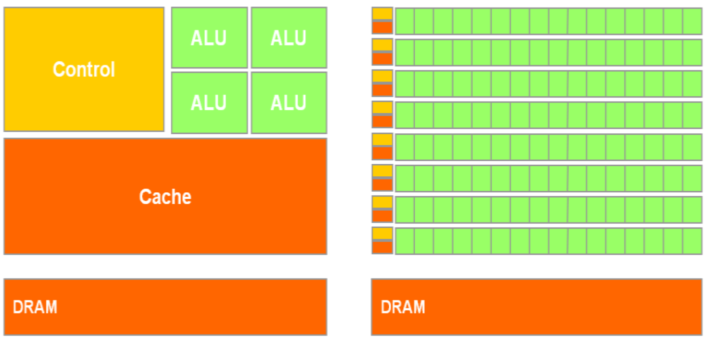


Fig.1 Fundamentaly design philosopies of CPU and GPU

Applications such as molecular modelling which are convenient for parallel computing are changed effortlessly for GPU computing, which require high processing power. As well as, use of several GPUs gives even more computing power for the same tasks.

General Purpose Graphics Processing Unit (GPGPU)
The main intension of using GPU is for graphics processing tasks but many computations such as matrix and vector operations, the use of GPUs for non-graphical calculations has increased. A concept is to use a General Purpose Graphics Processing Unit with modified form of stream processor. Use of GPGPU turns the massive floating-point computational power of a modern graphics card into general-purpose computing power, as opposed to being hard wired exclusively to do graphical operations. This can yield several orders of magnitude higher performance than a conventional CPU.

Compute Unified Device Architecture (CUDA)
CUDA is an API extension to the C programming and supports specified functions from a normal C program to run on the GPU's stream processors. The C program is capable of taking advantage of a GPU's ability to operate on large terms in sine series in parallel, by appropriately using the CPU whenever necessary. CUDA is also the first API to allow CPU-based applications to access directly the resources of a GPU for more general purpose computing without the limitations of using a graphics API.

3 IMPLEMENTATION OF MATHEMATICAL FUNCTION ON CPU AND GPU

Several mathematical calculations have been developed for checking the time required running the same program on CPU and GPU and final conclusion have been given that GPU is more faster than CPU. Here in this paper something different mathematical calculations that are mathematical series functions have been developed to compare CPU and GPU.
For this comparision Intel® Xeon® Processor E5-2620 (15M Cache, 2.00 GHz, 7.20 GT/s Intel® QPI) 6 core and 12 Threads CPU and NVIDIA QUADRO FX 1800 GPU have been used.
There are several parts to creating a complete program using CUDA. To compute sine series there are three parts: the main file, computation of sine on the machine (CPU), and computation of sine on the device (GPU).
Algorithm:

1. Program asked for angle in degree and number of terms for accuracy purpose.
2. Then on CPU the angle is converted into radian.
3. Converted angle and number of terms along with some required data send to GPU.
4. On GPU actual sine of the angle is computed
5. Result is transfer to CPU again.
6. With some computation final result has been displayed to user.

4 RESULT

After creating and running the program, the results I received showed that the GPU computed the sine of the given angle faster than the CPU. As the angle and number of terms are increased the amount of time it took to compute on the CPU increased. However, on the GPU even with the increase of the angle and number of terms, the computation time stayed pretty much the same (Figure 1 and 2).

International Journal of Scientific & Engineering Research, Volume 6, Issue ħŶŵ . ÊÛÖÉÌÙ-2015
ISSN 2229-5518

CPU: Intel® Xeon® Processor E5-2620		GPU: NVIDIA QUADRO FX 1800
Angle = 30 , No. of terms = 300		Angle = 30, No. of terms = 300
Sr.No	Computational Timing (ms)	Computational Timing (ms)
1	31	1.3
2	16	1.3
3	31	1.3
4	31	1.3
5	31	1.3
6	31	1.3
7	31	1.3
8	16	1.3
9	31	1.3
10	15	1.3
11	31	1.3
12	32	1.3
13	31	1.3
14	32	1.3
15	16	1.3
16	31	1.3
17	31	1.3
18	31	1.3
19	32	1.3
20	31	1.3
21	32	1.3
22	31	1.3
23	16	1.3
24	32	1.3
25	31	1.3
26	31	1.3
27	31	1.3
28	15	1.3
29	16	1.3
30	32	1.3
Sum	829	40.3
Avg.(ms)	27.63333333	1.3

Fig. 1 Computational comparision in CPU and GPU for Sine Series Expansion

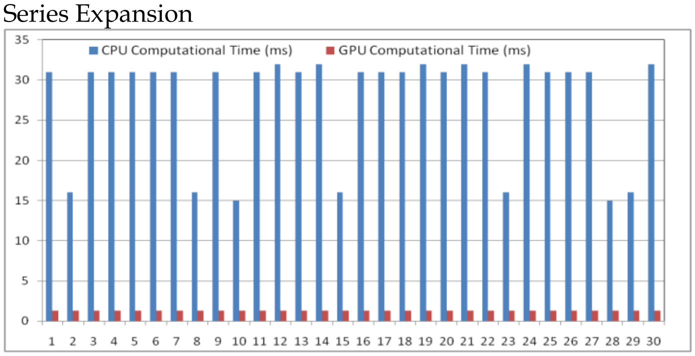


Fig. 2 Graph showing Computational time between CPU and GPU

5 CONCLUSION

Detailed study of performance of GPGPU families as an application to trigonometric series functions is done in the paper. Since series functions may or may not involves computation over number of terms for more accuracy, the multithreaded execution provided by CUDAGPGPU environment provides a viable method to calculate trigonometric value of given angle via such high performance computing platforms.

ACKNOWLEDGEMENT

I wish to acknowledge Head, Department of Electronics Engineering, DIAT, Pune for providing research facilities related to CUDA platform

REFERENCE

[1] David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors, Second Edition: A Hands-on Approach, Morgan Kaufmann, 2010
[2] Jessica Brazelton; Tuskegee University; Matrix Multiplication using Graphics Processing Unit; CSCI 452 Computer Science Seminar II.
[3] http://www.clarku.edu/~djoyce/trig/apps.html
[4] http://en.wikipedia.org/wiki/Uses_of_trigonometry
[5] http://www.mathopenref.com/trigfunctions.html
[6] http://hyperphysics.phyastr.gsu.edu/hbase/trgser.html#c2
[7] Sheetal Lahabar, P J Narayanan; Singular Value Decomposition on GPU using CUDA; IPDPS 2009 (IEEE International Parallel Distributed Processing Symposium); Report No: IIIT/TR/2009/163
[8] Kruger, J. and Westermann, R. 2003. Linear Algebra Operators for GPU implementation of Numerical Algorithms. In Proc. of SIGGRAPH
[9] Fujimoto, N. 2008. Faster Matrix-Vector Multiplication on GeForce 8800 GTX. In Proc. of IEEE International Parallel and Distributed Processing Symposium
[10] NVIDIA CUFFT library user's guide 6.0, Feb 2014.
[11] NVIDIA CUDA C Programming Guide 6.0, 2014.