OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Accelerating Real Time Applications on Heterogeneous Platforms

Vivek Vanpariya[1], Prof. S. Ramani[2]
[1, 2] *(SCSE, VIT University, Vellore, India)*

Abstract: *In this paper we describe about the novel implementations of depth estimation from a stereo images using feature extraction algorithms that run on the graphics processing unit (GPU) which is suitable for real time applications like analyzing video in real-time vision systems. Modern graphics cards contain large number of parallel processors and high-bandwidth memory for accelerating the processing of data computation operations. In this paper we give general idea of how to accelerate the real time application using heterogeneous platforms. We have proposed to use some added resources to grasp more computationally involved optimization methods. This proposed approach will indirectly accelerate a database by producing better plan quality.*

Index Terms: *Graphics processing unit (GPU), depth estimation, and video analysis.*

I. INTRODUCTION

General purpose CPU performance scaling has opened up a new direction for computer scientist and architect. Scaling performance of new generation GP-CPU has not as predicted by Gordon moor. After observing it closely the throughput for sequential code in GP-CPU is getting fluctuated between subsequent processor generations. This problem has motivated the scientist to innovate the high performance and cost effective hardware. As a solution computer scientist comes up with a low level GP-GPU core which has features like out of order execution and lack branch prediction to optimize the sequential code [1]. Khronos has a framework and standard set for heterogeneous parallel computing on cross-vendor and cross-platform hardware [7]. It provides an abstraction layer via which code can scale from simple embedded microcontrollers to GP-CPU's from Intel and AMD, up to massively-parallel GP-GPU hardware pipelines, all without modifying code [8]. Most important task of graphical processing unit (GPU) is to accelerate the graphical applications. It contains large amount of streaming processors for commutations when compared to Central Processing Unit (CPU). Also GPU have high-bandwidth for data transfer. Graphics processing unit's hasty increase in both programmability and capability has spawned a researcher's community that has effectively mapped a broad range of computationally demanding, complex problems to the GPU. In other words GPU computing is using it as to boost up central processing units for general-purpose scientific and engineering computations. It can be achieved by offloading some of the intensive and overwhelming portions of the code and the remaining part of the code will remain in the CPU. From user's view point, the application run more rapidly because of the extremely parallel processing power of the general purpose unit to boost performance which is well- known as "heterogeneous" or "hybrid" computing. Some critical factors like power, cost, bandwidth on which we have to focus when we working on GPU.

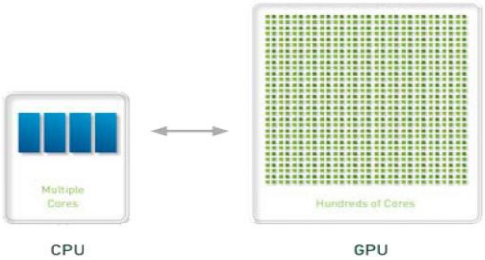

Figure 1: CPU vs. GPU

II. GPU ARCHITECTURE

Graphics processing units will be able to process independent fragments and vertices, but also can process many of them in parallel. This comes handy when the programmer needs to process many vertices or fragments in the same way. In this logic, GPUs are stream processors also called as stream at once. Stream can be said as collection of records that requires similar kind of computation.

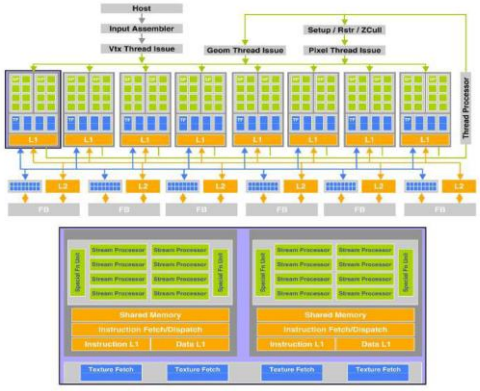

Figure 2: GPU Architecture

The 8800 is composed of 128 stream processors each rotating at the rate of 1350 Mhz. Each processor is capable to do calculation like MAD and MUL per clock cycle. Each processor need 4 cycles for executing specials instructions like EXP, LOG, RCP, RSQ, SIN, COS which is managed by an additional unit.

III. OPEN COMPUTING LANGUAGE

Open CL is popularly known as one of the free of royalty standard for general purpose parallel programming across CPUs, GPUs and other processors, giving software developer's transportable and capable access to the power of these assorted processing platforms. OpenCL consists of an application programming interface for coordinating parallel computation across heterogeneous processors and a cross-platform programming language with a well précised computation background.

Standards of Open CL:

- It supports data-based and task-based parallel programming models.
- It utilizes a rift of ISO C99 with extensions for parallelism.
- It defines steady numerical supplies based on IEEE 754.
- It defines a configuration report for handheld and embedded devices.
- It efficiently interoperates with OpenGL, OpenGL ES and other graphics APIs.

OPENCL memory model

Open CL framework provides the distributed memory. As shown in figure 3, each processor has its own private memory and each core has its own local memory.

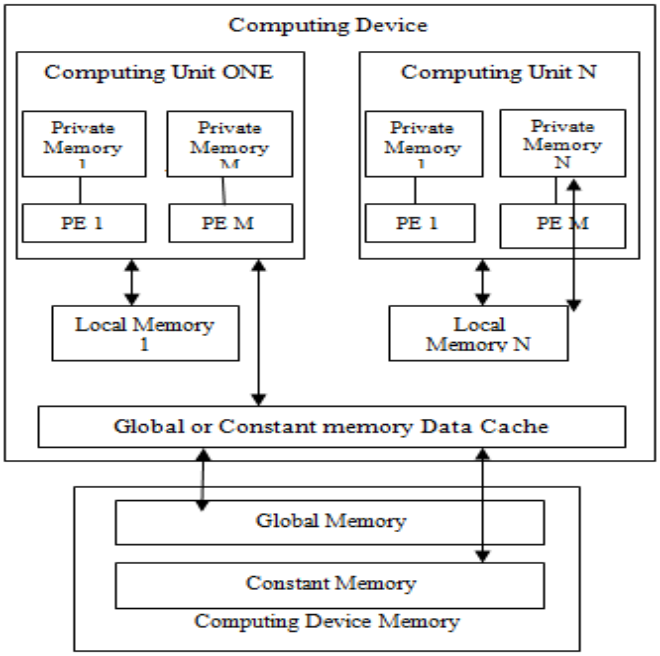

Figure 3: Architecture of Conceptual Open CL device having processing elements (PE), computing units and devices.

IV. TEST CASE FOR SERIAL VS PARALLEL CODE

In this section we give the basic comparison of serial and parallel code performance. If we take simple example of matrix multiplication in which each element is computed sequentially from the row of the first matrix and column of the second matrix. Same matrix multiplication can be done on parallel environment because there will be no previous data dependency. The below discussion give knowledge how to write parallel code using Open CL.

Parallel code of matrix multiplication is ported on Storm which is a tool of STMicroelectronics, which provide the parallel environment with 64 parallel processing units.

No. of Data Elements per matrix	No. of Cycles required for sequential code to run	No. of Cycles required for parallel code to run	Speed Up
16x16	1654850	32230	51
32x32	13185830	212402	62
48x48	44479008	701355	63
64x64	105413896	1653468	63
128x128	843161792	13180778	64
256x256	6744844288	105394568	64
512x512	53957111808	843086272	64

Table 2: Comparison table of Serial vs Parallel code of Matrix multiplication.

Tools have got 4 clusters and each cluster contains 16 processing elements (cores). Above table give comparison chart of matrix multiplication using Storm. If we take the advantage of local memory and memory bandwidth for accessing the data from host part to device part then we can get better speedup i.e., speed of data transfer from global memory to compute device is slower than the speed of data transfer from local memory to compute device. Because local memory is more near to clusters and it is local to each clusters. Memory coalscaling is another feature of parallel processing. Suppose my bit transfer capacity from host to device is 32 bit's then why should I transfer 8 bit at a time. Fetch 32 bit's at a time and store it on device local memory until four bytes process.

No. of Data Elements per matrix	No. of Cycles using Global Memory	No. of Cycles using Local Memory	Speed Up
16x16	32230	20477	1.57
32x32	212402	114949	1.85
48x48	701355	368612	1.90
64x64	1653468	860118	1.92

Table 3: Comparison table of global vs serial memory for Matrix multiplication.

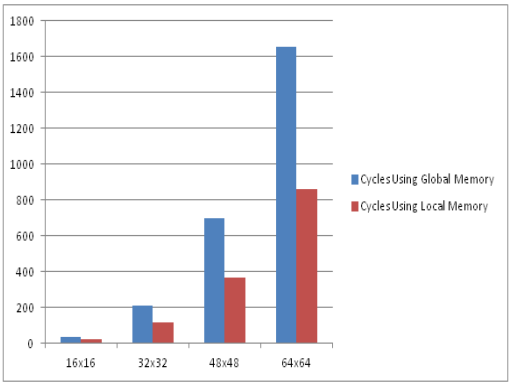

Figure 4: Cycles Needs for Global Vs. Local Memory for Matrix Multiplication

V. REAL TIME APPLICATION

Real time applications like depth estimation for stereo images, video analytics for survellience and etc. are the good applications on image processing technology where repetitive work on pixels is needed. For this we are concentrating on throughput rather than the latency. Filtering and/or convolution for images are one of the important modules for depth estimation or in video analytics. In below discussion we briefly discuss convolution and result analysis after porting it on NVIDIA's Quadro FX 4600 and Storm tool of STMicroelectronics.

Convolution

Kernel: It is a collection of numbers in the matrix form which is used in convolution of images. They are available in different size having different number patterns producing different results. The dimension of each kernel is subjective but most often 3x3 is preferred.

A convolution is usually performed by doing the multiplication of a pixel and its adjacent pixels color value by a matrix.

Convolve is likely used to Enhance, Sharpen or smoothing the image.

We have the formulae for doing convolution of image f(x, y) and kernel k(x, y) represented as

$$f(x,y) * k(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u,v)\, k(x-u,y-v)\,du\,dv \quad \ldots.(1)$$

In discrete form,

$$f(x,y)*k(x,y) = \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} f(i,j)\, k(x-i, y-j) \quad \ldots..(2)$$

W, H is the width and height of image respectively.

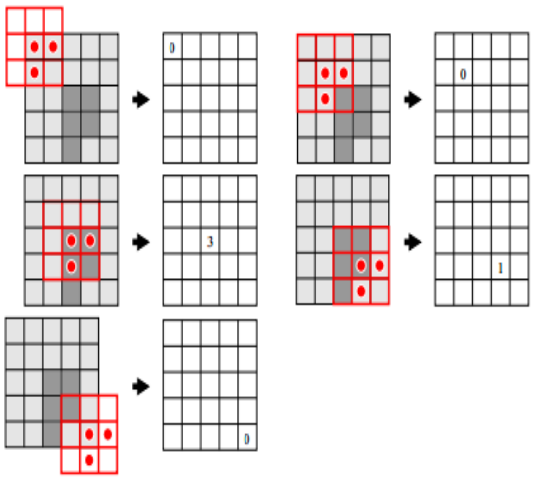

Figure 5: Masking on image.

VI. RESULT ANALYSIS

Function Name	Time require for assembly code (ms)	Time require for C/C++ code (ms)	Time require for Parallel code (ms)
convolve_cols_3x3	50	360	136
convolve_121_row_3x3_16bit	10	50	30
convolve_101_row_3x3_16bit	20	60	23

Table 4: Time analysis for assembly, serial and parallel code (using NVIDIA device).

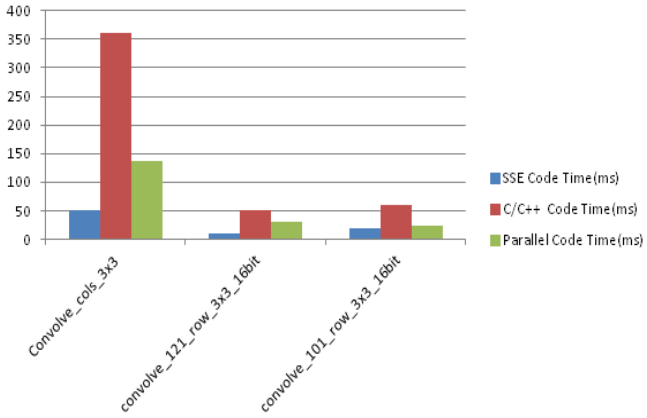

Figure 6: Comparison graph for table 4

Function Name	Time require for Assembly Code (ms)	Time require for C/C++ Code (ms)	Time require for Parallel Code on xp70(ms)
convolve_cols_3x3	50	360	128
convolve_121_row_ 3x3_16bit	10	50	28
convolve_101_row_ 3x3_16bit	20	60	20

Table 6: Time analysis for assembly, serial and parallel code (using Sthorm device).

VII. CONCLUSION AND FUTURE WORK

In this paper we have discussed the behavior of serial and parallel code with respect to different heterogeneous platforms. OpenCL has lot of task like getting a platform, kernel setting, memory/buffer initialization, filling the read buffer with input data, taking the device information, setting the work group and work items etc.. Due to this, performance results are not expected for small size data. By analyzing the comparison tables, we can accelerate the sequential code with the help of parallel computing elements on heterogeneous platforms.

In future related work, one can apply the optimization technique like memory co-aliasing, vector data types and pipeline futures on complex real time applications.

REFERENCES

[1] I. Blthoff, H. Blthoff, and P. Sinha. “Topdown influences on stereoscopic depth-perception”. Nature Neuroscience, 1:254.257.

[2] S. Das and N. Ahuja. “Performance analysis of stereo, vergence, and focus as depth cues for active vision”. IEEE Trans PAMI, 17:1213.1219.

[3] A. Smolic and P. Kauff, “Interactive 3-D video representation and coding technologies,” Proc. IEEE, Special Issue on Advances in Video Coding and Delivery, vol. 93, no. 1, pp. 98–110, Jan. 2005.

[4] K. Mueller, P.Merkle, H. Schwarz, T. Hinz, A. Smolic, T. Oelbaum, and T. Wiegand, “Multi-view video coding based on H.264.MPEG4-AVC using hierarchical B pictures,” in *Proc*. PCS’ 2006, Apr. 2006, paper SS3–3. [Online.] http://www.engineeringvillage2.org.

[5] Niem W, Buschmann R, ”Automatic Modelling of 3D Natural Objects fromMultiple Views”, European Workshop on Combined real and synthetic image processing for broadcast and video productions, 23–24. 11. 1994, Hamburg, Germany.

[6] L. Falkenhagen, A. Kopernik, M. Strintzis, ”Disparity Estimation based on 3D Arbi- trarily Shaped Regions”, RACE Project Deliverable, R2045/UH/DS/P/023/b1.

[7] Y. Yakimovski, R. Cunningham, ”A System for Extracting 3D Measurements from a Stereo Pair of TV Cameras”, CGVIP, Vol. 7, 1978, pp. 195 – 210.

[8] Koch, R., ”3–D Scene Modelling from Stereoscopic Image Sequences”, European Workshop on Combined real and synthetic image processing for broadcast and video productions, 23–24. 11. 1994, Hamburg, Germany.

Vivek Vanpariya received his B.E degree from A.D Patel institute of Technology affiliated to Sardar Patel University, Gujrat, India. He did his M. Tech in Computer Science and Engineering from VIT University Vellore, India. His area of interests includes Computer Network, Algorithms and Operating Systems.

Ramani S received his M.Tech degree from Bharathidasan Institute of Technology affiliated to Bharathidasan University, TamilNadu, India. He is pursuing his Ph. D from VIT University Vellore, India. His area of interests includes Database, Web Technologies and Data Mining.