Computer Science Honours Project Proposal

Parallel processing with GPU's

Simon Naudé

Department of Computer Science, Rhodes University

Supervisor:
Karen Bradshaw
February 28, 2014

Project Title

The project is proposed under the title of:
Applications for parallel processing using GPU's

Statement of the project

Graphical Processing Unit's (GPU's) have increased substantially in performance over the years, allowing them to out perform your Central Processing Unit (CPU) in certain aspects. [2] Not only have GPU's started to out perform CPU's, but they have also become cheaper in regards to Flops[1]/dollar. [3] This has made research into trying to do other operations very popular. The problem with General Purpose GPU's (GPGPU's) is that GPU's must executed in lock-step (where a group of cores execute the same function). This adds limitations to programming the GPU. [2] However certain processes can be performed well under these conditions, such as having multiple decryption algorithms run on the same piece of information, and once they are all done, take the output data of which ever process you want.

Objective of Research

Based on the above statement, the objectives of this research are:

- Primary Goals:
  - Create a application that utilizes parallel processing with decryption using OpenGL. First working on the CPU
  - Port working application onto the GPU
  - Benchmark the application between the GPU and the CPU
- Secondary Goals and future possible additions:
  - add support for multiple GPU's
  - add support for multiple GPU's in Crossfire

History and Background

The processing power of a Graphic Processing Unit (GPU) is greater than that of a normal processor on a given die, however it contains a constraint where a group of cores must execute the same instruction (Lock-Step execution). This causes the GPU to be faster but less versatile then normal processors (Such as the Central processing Unit (CPU)). [2]

[1] The number of FLoating-point Operations Per Second - Giving an accurate measure of the processors performance

Recently attention has been drawn towards using GPU's to accelerate non-graphic performance; this recent interest is due to two factors of the GPU: Its Price / Performance Ratio; and its Evolution of speed. [3] The Evolution of speed is mainly due to the gaming industry demanding better graphics, which in result requires faster Graphics cards. This demand has made the GPU speed double in speed approximately twice a year. [3] Since there is such a high demand in the gaming industry the price of the actual graphics cards have been able to stay relatively cheap in comparison to a CPU. [3]

Rendering a graphical scene with a GPU is made up of a sequence of processing stages. These stages run in parallel and in a fixed order, which is known as the graphics hardware pipeline. [3] With a simple view there are four main stages in this process:
1) In this stage (vertex processing) the GPU gets a 3D polygraph mesh, which is then converted into a 2D screen position render. During this, Color and texture coordinates associated with the vertex is evaluated. [3]
2) Within the second stage the vertices are grouped into triangles, and then scan-converted which generates a set of fragments. These fragments stores the state information needed to update a pixel. [3]
3) During the third stage (Fragment Processing) texture coordinates of the fragments are used to get colors from one or more textures. Mathematical operations are then performed to determine the ultimate color for the fragment. [3]
4) The final stage is comprised of various tests (Such as depth and alpha tests) which will determine whether or not the fragment should be used to update the pixel. [3]

There are two major producers of GPUs, Nvidia and AMD(Radeon), and each use their own programming structure. Nvidia use their Cuda, and AMD use OpenGL. Some of Major operations an OpenGL application would perform (the pipelined process) to render an image include: Specifying data for constructing shapes; Executing various shaders to determine the position, color, and other attributes, of the inputs; rasterization; executing a fragment shader for each fragment generated during the rasterization phase; any additional pre-fragment operations. [4]

This gives the possibility for accelerated computing, which involves offloading any computation intensive portions of a program to the GPU. [1] Allowing certain types of applications, such as: Volume rendering; flow visualization; ray-tracing; and simulations, to be run at a greater speed.

Approach

The first phase would be to get familiar with the different aspects of the project, such as the architecture of the GPU's I am going to be using. It will also include getting used to OpenGL and how it works, and on the different decryption algorithms I could use
The second phase will include doing research to see if there have been any other experiments on decryption using GPGPU's, and reading up on what methods they may have taken. The third phase will include implementing a solution for a GPGPU application, but using the

CPU (As CPU's allow for easier debugging), after this I will port the application over onto the GPU The final phase will include running benchmark tests to see if my application was able to decrease the time taken to run the decryption algorithms using the GPGPU's over the time taken to run the algorithms on a CPU

Requirements

The hardware requirements for this project are as follows:

- Access to a x64 based personal computer
- A multi-core CPU
- An AMD HD 7970OC Graphics card
- (Extension) An addition AMD HD 7970OC Graphics card (With crossfire capabilities)

The software requirements for this project are as follows:

- Windows 7 or higher OS
- Visual Studio 2010

Project Progression Time-line

03 March 2014 - Formal written proposal completed
04 March 2014 - Project Presentation
30 May 2014 - Literature review and plan of action
6 - 20 June 2014 - First Semester Exams
29 July 2014 - Seminar Series 2 – Part 1
5 August 2014 - Seminar Series 2 – Part 2
12 August 2014 - Seminar Series 2 – Part 3
15 September 2014 - Short Paper Submitted
27 - 29 October 2014 - Seminar Series 3
31 October 2014 - Project deadline
7 November 2014 - Research Website Complete
19 November 2014 - Final Oral Examination

References

[1] What is gpu accelerated computing? Online: http://www.nvidia.com/object/what-is-gpu-computing.html.

[2] P. Chongstitvatana. Putting general purpose into a gpu-style softcore. In *Int. Conf. on Embedded Systems and Intelligent Technology*, 2013.

[3] Qiu F. Kaufman A. Yoakum-Stover S. Fan, Z. Gpu cluster for high performance computing. In *Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, page 47, 2004.

[4] Sellers G. Kessenich J. Licea-Kane B. Shreiner, D. *OpenGL Programming Guide*. Addison-Wesley, 8 edition, 2013.