Project Proposal

MATTHEW ROYLE

Supervisor: PROF. SHAUN BANGAY

Computer Science Department, Rhodes University

03 March 2009

1. Principle Investigator

MATTHEW GEORGE ROYLE

39 Worcester Mews, Worcester Street

Grahamstown, 6139

0726984925

g06r2001@campus.ru.ac.za

Supervised By: PROF. SHAUN BANGAY

2. Project Title

This project is proposed under the title of:

HIGH LEVEL OPENCL IMPLEMENTATION

3. Statement of the Problem

OpenCL has been developed for writing programs that run in parallel. While *OpenCL* has been engineered to take advantage of GPUs (General Processing Units) it could potentially be used on other architectures such as multi-core CPUs available in the latest generation of desktop PCs, or over networks of individual machines. While the standard is available at present, no implementation has yet been completed.

This project will investigate the implementation of *OpenCL* through the reuse of existing technologies. Rather than provide a dedicated *OpenCL* compiler producing architecture specific parallel code, this project will create a translator for *OpenCL* programs to existing parallel processing frameworks. This can could be as simple as a threads implementation on a PC, or a GPGPU based version using platforms such as CUDA.

The emphasis in this project will be on cunning software engineering, to achieve the most effective project with the least amount of effort. Secondary goals will be the evaluation of the *OpenCL* standard on various architectures.

4. Rationale and Motivation

Since the rendering of graphics is a mathematically intensive task, GPUs were originally developed to speed up the rendering process. To enable high speed rendering of graphics, GPUs have been developed with a graphics pipeline, which allows for mathematical calculations to be run in parallel. Because of their design, GPUs are suited for applications that run in parallel and are computationally intensive.[5]

GPUs have been adapted for use as general computational devices, this has led to them become known as General-Purpose Graphics Processing Units (GPGPUs). GPGPUs can now act as co-processors to CPUs, this can help reduce the computational load on CPUs. The ability for the two processors to act in conjunction has led to the need for a language that is able to function over both architectures, this is where *OpenCL* comes into play.[6]

The main focus for the *OpenCL* standard so far has been to implement it for use with GPUs. Apple, who originally conceived the idea, will include the *OpenCL* standard in its latest operating system for: Mac OS X 10.6 nicknamed “Snow Leopard”[2]. NVIDIA has indicated that it will include *OpenCL* on its GPUs[3], while AMD also plan to include *OpenCL* for use on its own GPUs[1]. Apple have also announced future plans to implement *OpenCL* on its popular iPhone[8]. *OpenCL* has been planned for launch on multiple platforms which include: mobile phones, handheld devices and multiple processor platforms[11]. So far, the most likely implementation is for its use on GPUs as announced by NVIDIA and AMD, as well as its implementation on mobile phones. This is most likely due to the popularity of these platforms and their widespread use. There is the possibility of expanding its implementation to other non x86 processor types such as the PS3's Cell processor[11]. The motivation for this project stems from the lack of implementation on CPU architecture.

5. Research Aims

The aims of this project can be divided into primary and secondary goals. Secondary goals will only be undertaken if the primary goals are met satisfactorily. These are:

- Primary Goals
  - Find a suitable high-level language to use as a translator
  - Create a high-level translator for the *OpenCL* specification on the x86 architecture
  - Create valid tests for the translator
  - Run created tests to ensure that the translator works correctly
- Secondary
  - Improve on the performance of the initial implementation
  - Evaluate the *OpenCL* standard on various architectures

Therefore a high-level translator for the x86 architecture will have to be created. The translator will be created using some sort of thread-based language to allow for the parallel nature of the standard. The purpose of the testing will initially involve testing to see whether the implementation works correctly, later testing could involve testing for speed and performance of the implementation. Some tests which could be used could include: generating a fractal image using parallel algorithms or implementing a version of John Conway's Game of Life using a grid based system for parallel computation.

6. Workplan: Research Activities

Initial research activities will involve gaining an understanding of the *OpenCL* specification.[11][7] Once an understanding of the specification has been established, it is necessary to find a suitable high-level language to act as a translator.[10] Before comparing how parallel programs run on various architectures, it is important to have an understanding of creating parallel algorithms and how processors go about running code in parallel. Next it will be important to understand how *OpenCL* will interface with the x86 architecture. It will be necessary to understand the differences between running parallel code on a GPU versus running it on a CPU.

Once an understanding of how the systems will function has been achieved, an initial high-level translator will be implemented to emulate the *OpenCL* language for the x86 architecture. A test program will then be written, this will be used to test that the implementation works correctly. After the results of testing have been declared satisfactory, then a refinement of the original implementation will commence to improve its performance. If time permits, an evaluation of the *OpenCL* standard will be performed on various architectures.

7. Workplan: Research Approaches/Methods/Techniques

Research will be conducted by reviewing various papers in the relevant fields, which include: parallel computation, thread-based x86 languages, GPGPUs, NVIDIA's CUDA® language[9] and AMD's Brook+[4]. Furthermore it is essential to understand how *OpenCL* will be implemented by studying the *OpenCL* header files. It will be necessary to perform a review of high-level languages which support threading, from this review a suitable language will need to be selected to act as a translator. Implementation will be performed by using a thread-based language to implement a parallel processing environment, some of these languages could include: C/C++, Java, C# or Python. Functions specified by the *OpenCL* specification will be implemented by using *#define* or a similar concept in the high-level language.

8. Timeline

The following lists the progression of the project which includes the project milestones set out by the department. This timeline does not include any project work which involves gaining proficiency for using any languages involved or any coding required for the project. These will be decided at a later date.

- *3 March 2009*

  Hand in project Proposal

- *20 April 2009*

  Begin literature review

- *26 June 2009*

  Hand in literature review

- *24 August 2009*

  Complete poster

- *14 September 2009*

  Draft project paper

- *21 September 2009*

  Final project paper

- *28 September 2009*

  First chapter draft

- *2 November 2009*

  Oral presentation

- *9 November 2009*

  Hand in project

9. Possible Outcomes

The possible outcomes of this project will include a working translator based on the x86 system, a working test program that will help in the testing of *OpenCL* code and evaluation of the *OpenCL* standard on various architectures.

References

[1] AMD Drives Adoption of Industry Standards in GPGPU Software Development. $< http : //www.amd.com/us - en/Corporate/VirtualPressRoom/0,,51_104_543\ 127451,00.html >$, August 2008.

[2] Apple Previews Mac OS X Snow Leopard to Developers. $< http : //www.apple.com/pr/library/2008/06/09snowleopard.html >$, June 2008.

[3] NVIDIA Adds OpenCL To Its Industry Leading GPU Computing Toolkit. $< http : //www.nvidia.com/object/io_1228825271885.html >$, December 2008.

[4] AMD. *Brook+*, November 2007.

[5] K. BOGGAN and D.M. PRESSEL. GPUs: An Emerging Platform for General-Purpose Computation. August 2007.

[6] M. GLADDING, D. VILLMOW and B. ZHENG. Building a High Level Language Compiler for GPGPU, June 2008.

[7] Kronos OpenCL Working Group. *The OpenCL Specification*, February 2009.

[8] P. McLEAN. Future iPhones to wield OpenCL acceleration. *AppleInsider*, December 2008.

[9] NVIDIA Corporation, 2701 San Tomas Expressway Santa Clara, CA 95050. *NVIDIA CUDA Compute Unified Device Architecture*, June 2007.

[10] M. SALIH and O. TONCHEV. High-level programming languages translator. Master's thesis, Blekinge Institute of Technology, January 2008.

[11] N. TREVETT. OpenCL The Open Standard for Heterogeneous Parallel Processing. In *SIGGRAPH '08: ACM SIGGRAPH Courses*, Singapore, December 2008.