

**Modeling the performance of many-core programs  
on GPUs with advanced features**

by

**Pei Momo**



**Master of Science in Software Engineering**

**2012**



**Faculty of Science and Technology  
University of Macau**

Modeling the performance of many-core programs on GPUs  
with advanced features

by

Pei Momo

A thesis submitted in partial fulfillment of the  
requirements for the degree of

Master of Science

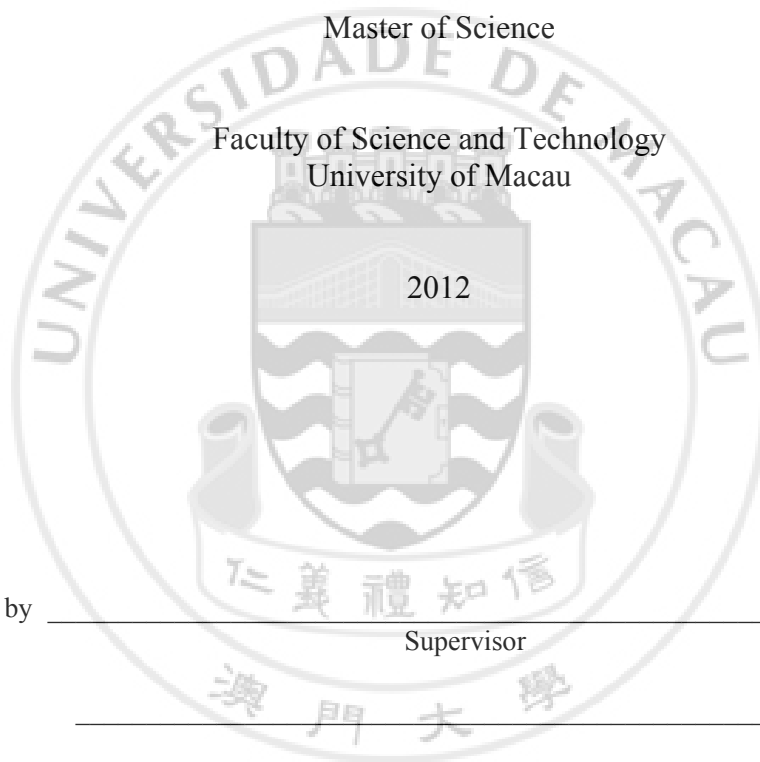
Faculty of Science and Technology  
University of Macau

2012

Approved by

Supervisor

Date



In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Macau, I agree that the Library and the Faculty of Science and Technology shall make its copies freely available for inspection. However, reproduction of this thesis for any purposes or by any means shall not be allowed without my written permission. Authorization is sought by contacting the author at

Address:

Telephone:

Fax:

E-mail:

Signature \_\_\_\_\_

Date \_\_\_\_\_



University of Macau

Abstract

MODELING THE PERFORMANCE OF MANY-CORE  
PROGRAMS ON GPUS WITH ADVANCED FEATURES

by Pei Momo

Thesis Supervisor: Dr. Xu Qiwen

Many-core graphics processing units (GPUs) as accelerators are increasingly used in scientific and parallel computing, but the performance optimization of general-purpose GPU computing is still a big challenge for developers. This paper provides a new analytical performance model in order to simulate all the main features of advanced GPU micro-architectures and evaluate performance of the features for specific GPGPU programs.

The proposed model can be used to estimate the overall execution time and find the potential bottlenecks of GPGPU applications. Developers can analyze the major GPU features because the latency of every GPU component can be calculated using the tool which based on the proposed model. Two latency relationships are provided in order to integrate the overlapping component latencies.

The latencies of global memory, shared memory and cache accesses also are analyzed in the proposed model. Two approaches are provided to calculate the number of global memory transactions, one (pattern method) is very simple and efficient, and the other (traversal method) can be used in all cases. Moreover, two corresponding approaches also are designed to estimate the number of the shared memory accesses (bank conflicts). In addition, a configurable trace-driven simulation is proposed for GPU caches so as to estimate the cache miss ratios and the latency.

## TABLE OF CONTENTS

|                                                    |    |
|----------------------------------------------------|----|
| List of figures.....                               | iv |
| List of TABLES.....                                | vi |
| Chapter 1: Introduction.....                       | 1  |
| 1.1.    Background.....                            | 1  |
| 1.2.    Motivation.....                            | 1  |
| 1.3.    Proposed model.....                        | 2  |
| 1.3.1.    Brief introduction.....                  | 3  |
| 1.3.2.    Tool.....                                | 3  |
| 1.4.    Outline of the thesis.....                 | 4  |
| Chapter 2: GPGPU computing.....                    | 5  |
| 2.1.    CUDA.....                                  | 5  |
| 2.1.1.    Hardware architecture.....               | 5  |
| 2.1.2.    Software architecture.....               | 6  |
| 2.2.    Micro-architecture features.....           | 8  |
| 2.2.1.    Latency hiding.....                      | 8  |
| 2.2.2.    Computation units.....                   | 8  |
| 2.2.3.    Barrier synchronization.....             | 8  |
| 2.2.4.    Coalesced/uncoalesced memory access..... | 9  |
| 2.2.5.    Bank conflicts.....                      | 9  |
| 2.2.6.    Cache.....                               | 9  |
| 2.2.7.    Others.....                              | 9  |
| Chapter 3: Relative work.....                      | 10 |
| 3.1.    Valiant.....                               | 10 |
| 3.2.    Hong et al.....                            | 10 |
| 3.1.1.    Case 1: $CWP \geq MWP$ .....             | 11 |
| 3.1.2.    Case 2: $MWP > CWP$ .....                | 11 |
| 3.1.3.    Case 3: Not enough warps running.....    | 12 |
| 3.3.    Baghsorkhi et al.....                      | 12 |

|            |                                                                       |    |
|------------|-----------------------------------------------------------------------|----|
| 3.4.       | Others.....                                                           | 13 |
| 3.4.1.     | Sim et al. ....                                                       | 13 |
| 3.4.2.     | Resios.....                                                           | 14 |
| 3.4.3.     | Zhang et al.....                                                      | 14 |
| Chapter 4: | Analytical model .....                                                | 15 |
| 4.1.       | Latency relationships .....                                           | 16 |
| 4.2.       | Performance prediction.....                                           | 16 |
| 4.2.1.     | $\text{Latency}_{\text{mem}} \geq \text{Latency}_{\text{comp}}$ ..... | 17 |
| 4.2.2.     | $\text{Latency}_{\text{comp}} > \text{Latency}_{\text{mem}}$ .....    | 18 |
| 4.2.3.     | No enough active warps and ILPs .....                                 | 18 |
| 4.2.4.     | Overview of latencies .....                                           | 19 |
| 4.3.       | Computation time.....                                                 | 19 |
| 4.3.1.     | Active warps .....                                                    | 20 |
| 4.3.2.     | SP computation time.....                                              | 21 |
| 4.3.3.     | SFU computation time .....                                            | 23 |
| 4.3.4.     | Shared memory access time.....                                        | 25 |
| 4.3.5.     | Cache accesses time .....                                             | 26 |
| 4.3.6.     | Synchronization overhead.....                                         | 26 |
| 4.3.7.     | Total computation time.....                                           | 27 |
| 4.4.       | Memory access time .....                                              | 29 |
| 4.4.1.     | MWP .....                                                             | 29 |
| 4.4.2.     | Global memory access .....                                            | 30 |
| 4.4.3.     | Total memory access time .....                                        | 32 |
| 4.5.       | Overall execution time .....                                          | 32 |
| Chapter 5: | Memory access analysis.....                                           | 34 |
| 5.1.       | Memory Transaction.....                                               | 34 |
| 5.1.1.     | Index analysis.....                                                   | 34 |
| 5.1.2.     | Pattern method .....                                                  | 35 |
| 5.1.3.     | Traversal method .....                                                | 38 |
| 5.1.4.     | Examples.....                                                         | 39 |
| 5.2.       | Shared memory access.....                                             | 41 |

|                                           |                                    |    |
|-------------------------------------------|------------------------------------|----|
| 5.2.1.                                    | Pattern method .....               | 41 |
| 5.2.2.                                    | Traversal method .....             | 43 |
| 5.2.3.                                    | Examples.....                      | 43 |
| Chapter 6: Cache simulation.....          |                                    | 45 |
| 6.1.                                      | Cache.....                         | 45 |
| 6.2.                                      | Trace-driven simulation.....       | 46 |
| 6.3.                                      | Problems for many-core GPUs .....  | 47 |
| 6.4.                                      | Proposed simulation.....           | 48 |
| 6.4.1.                                    | Two layer structure .....          | 48 |
| 6.4.2.                                    | Cache simulator .....              | 49 |
| 6.5.                                      | Examples.....                      | 51 |
| Chapter 7: Experimental methodology ..... |                                    | 52 |
| 7.1.                                      | Case study: vector addition.....   | 52 |
| 7.2.                                      | Case study: matrix transpose ..... | 55 |
| Chapter 8: Conclusion.....                |                                    | 57 |
| 8.1.                                      | Future works .....                 | 57 |
| bibliography .....                        |                                    | 59 |

## LIST OF FIGURES

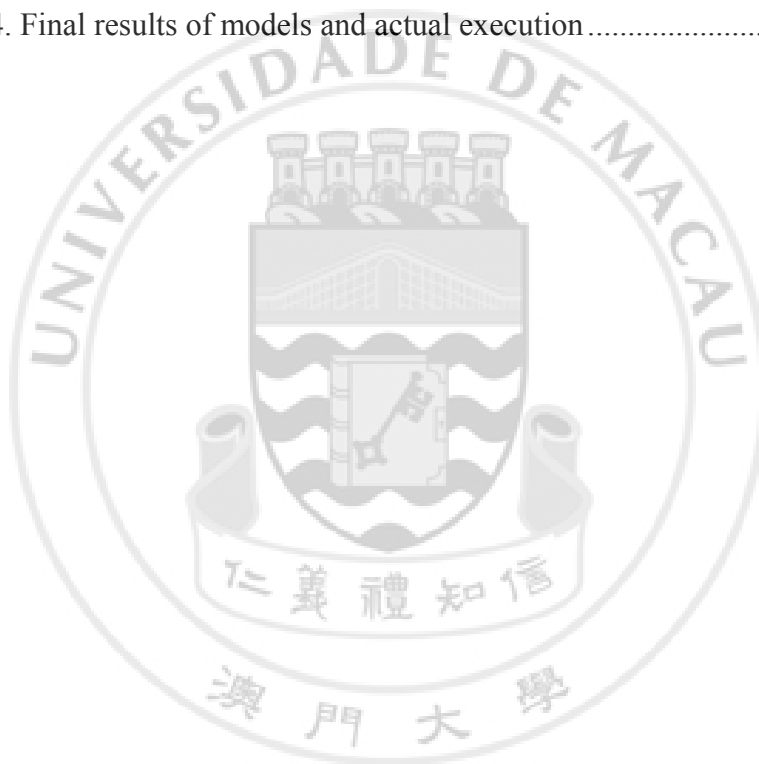
| <i>Number</i>                                                                                          | <i>Page</i> |
|--------------------------------------------------------------------------------------------------------|-------------|
| Figure 2.1. Hardware architecture of an advanced GPU .....                                             | 6           |
| Figure 2.2. The thread hierarchy in the CUDA platform.....                                             | 7           |
| Figure 2.3. A simple kernel definition .....                                                           | 7           |
| Figure 2.4. A simple main function definition.....                                                     | 8           |
| Figure 3.1. A example of CWP is greater than MWP .....                                                 | 11          |
| Figure 3.2. A example of MWP is greater than CWP .....                                                 | 12          |
| Figure 3.3. A WFG for statement $x = A[\text{threadIdx.x}] + y$ .....                                  | 13          |
| Figure 4.1. An example for $\text{Latency}_{\text{mem}} > \text{Latency}_{\text{comp}}$ .....          | 17          |
| Figure 4.2. An example for $\text{Latency}_{\text{comp}} > \text{Latency}_{\text{mem}}$ .....          | 18          |
| Figure 4.3. An example for no enough warps and ILPs .....                                              | 19          |
| Figure 4.4. The relationship between latencies on an advanced GPU .....                                | 19          |
| Figure 4.5. Instructions in the SP computation pipelines .....                                         | 23          |
| Figure 4.6. A case which SPs dominated the latency .....                                               | 24          |
| Figure 4.7. A case which SFUs dominated the latency .....                                              | 25          |
| Figure 4.8. A case for no enough active warps and ILPs .....                                           | 25          |
| Figure 4.9. A synchronization with $\text{Latency}_{\text{mem}} > \text{Latency}_{\text{comp}}$ .....  | 27          |
| Figure 4.10. A synchronization with $\text{Latency}_{\text{comp}} > \text{Latency}_{\text{mem}}$ ..... | 27          |
| Figure 4.11. An example for $\text{Latency}_{\text{units}} > \text{Latency}_{\text{shared}}$ .....     | 28          |
| Figure 4.12. An example for $\text{Latency}_{\text{units}} < \text{Latency}_{\text{shared}}$ .....     | 28          |
| Figure 4.13. (a) Global memory system model (b) time line of memory warps .....                        | 29          |
| Figure 4.14. A possibility on a Fermi-based GPU in the example. ....                                   | 30          |
| Figure 4.15. Over part for the global memory access time.....                                          | 31          |
| Figure 4.16. The relationship between two batches.....                                                 | 33          |
| Figure 5.1. The global memory accesses of the first warp in example 1 .....                            | 40          |
| Figure 5.2. The global memory accesses of the first warp in example 2.....                             | 40          |
| Figure 5.3. The global memory accesses in example 3 .....                                              | 41          |
| Figure 6.1. 1, 2, 8-way set associative in caches .....                                                | 46          |

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| Figure 6.2. The two-layer structure for caches in an advanced GPU ..... | 49 |
| Figure 6.3. a node for a cache line .....                               | 49 |
| Figure 6.4. The structure of the trace-driven simulator .....           | 50 |
| Figure 7.1. A CUDA kernel for vector addition .....                     | 52 |
| Figure 7.2. Estimated execution times for these kernels on GT 430 ..... | 55 |
| Figure 7.3. Trend comparison between models and actual execution .....  | 56 |



## LIST OF TABLES

| <i>Number</i>                                                  | <i>Page</i> |
|----------------------------------------------------------------|-------------|
| Table 4.1. The number of available resources on each SM.....   | 20          |
| Table 7.1. Hardware parameters for GT 430 .....                | 54          |
| Table 7.2. Parameters in this case for the proposed model..... | 54          |
| Table 7.3. Parameters in this case for Hong model .....        | 55          |
| Table 7.4. Final results of models and actual execution .....  | 56          |



## ACKNOWLEDGMENTS

First of all, the author would like to thank his supervisor, Dr. Xu Qiwen for always being there to support him academically and personally, guiding him through the difficulties and recognize the nature of object of study, so for that and more the author is always grateful.

Of equal importance is that thank Prof. Jeff Sanders for his help sincerely. He gives the author a lot of constructive suggestions, including not only how to complete the final year paper but also how to become a qualified researcher and author.

Moreover, the author also greatly appreciates Prof. Chen Yifeng, Dr. Zhang Yu, Mr. Yin Jiangling, Mr. Wang Peng and else who encourage him a lot during the design and composition of this thesis.

The author has been supported by Macau Science and Technology Development PEARL project (No. 041/2007/A3).

The author also would like to give his acknowledgement to his university, University of Macau. Thanks for providing him a good environment to study.

Finally, the author would like to say that I love you, my parents.

