論文の内容の要旨

# Research on Performance Optimization Methods based on Performance Analytical Modeling and Communication Latency Hiding in GPU
# （GPU の性能分析モデリングと通信レイテンシの隠蔽に基づく性能の最適化方法に関する研究）

氏名　羅成

Stream processing has been widely used since the emergence of stream applications such as time varying visualization and audio/video processing. Stream processing can exploit the inherent parallelism of the pipeline while the different stream elements also can be processed simultaneously to achieve data parallelism. Graphic processing unit (GPU) is one of the most successful stream architectures in recent years which is originally designed for acceleration of graphics applications. Now, it is widely used as General-purpose computing on graphics processing units (GPGPU) to accelerate many scientific applications with more than 10 times speedup over CPU platform. There are many new programming languages that help programmers to write parallel applications with GPUs such as Brook+, CUDA and OpenCL. With these programming and architectural features, programmers can quickly port their programs to a GPU based platform. However, if programmers want to have a better performance, they need to have a further understanding at various features of the low-level architecture and associated bottlenecks in their applications which will increase their burden in writing parallel applications. Therefore, there are many researches working on performance optimization methods from many aspects for programmers without much knowledge of GPU.

The motivation behind this work was caused by the emerging of high computational potential GPU along with the difficulty of writing high performance parallel programs on GPU based system. Our interests focus on performance prediction problem and communication latency between the host and the device problem. For performance prediction problem, it is difficult to

predict the performance of kernel codes on GPU without enough knowledge about the low level architecture. There- fore, programmers may use unsuitable configuration to run their applications on GPUs which may lead to poor performance. Therefore, performance analytical model is needed to help programmers better understand the performance of their applications on GPU and find out the performance bottlenecks.

On the other hand, the communication latency between the host and device also can greatly affect the performance. CUDA programs include two parts: host code running on CPU and device code running on GPU. The host code invokes the device code to execute the kernel operation while the input and output streams are stored in device memory. As the device memory is separated from the host memory, streams are required to be transferred between them which leads to the communication latency between the host and device. According to different application types, the communication latency overhead between the host and device will account from very little to very high proportion of the total execution time cost. It is difficult for programmers to achieve high performance without awareness of the communication latency. CUDA supports concurrent execution for kernel execution and data transfer. Notice that some latest NVIDIA Tesla series GPUs begin to support two copy engines for bi-directional data transfer which enables to launch data send, kernel execution and data receive simultaneously. With this new feature, it is possible to use three streams respectively for data send, kernel execution and data receive to hide the communication latency by overlapping the three streams.

In this thesis I am proposing performance optimization methods based on performance analytical modeling and communication latency hiding to solve the performance prediction problem and communication latency problem respectively. For performance prediction problem, I propose a performance analytical model which can help programmers have a better insight into their applications and give a better configuration to execute application based on the predicted results. For communication latency problem, I propose a task partitioning and scheduling method named TPSM to help programmers achieve to hide the communication latency between the host and the device in GPU based system.

The performance analytical model can estimate the execution time of massively parallel programs which take the instruction-level and thread-level parallelism into consideration. The model contains two components: memory submodel and computation submodel. The memory submodel can estimate the cost of memory instructions by considering the number of active threads and GPU memory bandwidth. Correspondingly, the computation submodel can estimate the cost of computation instructions by considering the number of active threads and application’s arithmetic intensity. I use ocelot to analyze PTX codes to obtain several input parameters for the two submodels such as memory transaction number and data size. Based on the two submodels, the analytical model can estimate the cost of each instruction while considering instruction-level and thread-level parallelism, thereby estimating the overall execution time of an application. With the predicted results, programmers can choose a suitable configuration to execute their applications with better performance.

I also propose a Task Partition and Scheduling Method (TPSM) which can help programmers to partition individual GPU application into subtasks and improve the performance of individual application with three streams by overlapping data send, kernel execution and data receive. With two copy engines, the work support simultaneous data send, kernel execution and data receive while previous work can only support simultaneous unidirectional data transfer and kernel execution. To extract the features of application, I classify GPU applications into several basic types from computation-to-communication ratio aspect and send-to-receive ratio aspect. With the classification, I design corresponding task partitioning and scheduling sub-methods. I also design a time optimal data transfer algorithm to achieve optimal data transfer between host and device in multiple GPU architecture. TPSM can be applicable in single GPU architecture, multiple GPU symmetric architecture and multiple GPU non-symmetric architecture.

I use four benchmarks to test the performance analytical model and tasking partitioning and scheduling method on various GPUs. The results show that the performance analytical model can achieve on average around 90% accurate ratio for the prediction of kernel execution. The results of TPSM show that the work proposed in this thesis can successfully hide the communication latency between for individual application to achieve high performance which is very close to the lower bound time cost.