Transparency CPU-GPU Collaboration for Data-Parallel Kernels on Heterogeneous Systems

1 Summary

This paper presented the single kernel multiple devices (SKMD) system, a framework that transparently orchestrates collaborative execution of a single data-parallel kernel across multiple asymmetric CPUs and GPUs.

SKMD is an abstraction layer located between applications and the OpenCL library. It uses OpenCL as the intermediate language. SKMD transparently partitions an OpenCL kernel across multiple devices being aware of the transfer cost and performance variation on the workload, launches parallel kernels, and merges the partial results into the final output automatically. Their system not only eliminates the tedious process of re-engineering applications when the hardware changes, but also makes efficient partitioning decisions based on application characteristics, input sizes, and the underlying hardware.

The dynamic compiler of SKMD has three main components: Kernel Transformer, Buffer Manager, and Partitioner. The kernel transformer changes the original kernel to Partition-Ready kernel, which enables the kernel to work only on a subset of work-groups. After kernel transformation, the buffer manager performs static analysis on kernels to determine memory access pattern of each work-group. If memory access range of each work-group can be analyzed statically, the buffer manager will transfer only necessary data back; if memory access range cannot be analyzed, entire input should be transferred to each device and output must be merged. In order to merge irregular locations of output from different devices, the kernel transformer generates Merge Kernel, and SKMD launches it on CPU device.

2 Intellectual Merit

In the technique aspects, this paper first proposed three central challenges that hinder most parallel kernels to benefit speedup from multiple computing devices: 1) Data-parallel kernels with irregular memory access patterns are hard to partition over multiple devices; 2) The partitioning decision becomes more complicated when systems are equipped with several types of devices; 3) The performance of a GPU is often not constant to the amount of data it operates upon, and this variation will affect the partitioning decision.

To solve these problems, this paper makes the following contributions: 1) The SKMD runtime system accomplished transparent collaborative execution of a data-parallel kernel. 2) Designed a code transformation methodology that distributes data and merges results in a seamless and efficient manner regardless of the data access pattern. 3) Proposed a partitioning algorithm that balance the workload among multiple asymmetric CPUs and GPUs considering the performance variation of each device.

The key point of this paper is assigning a subset of work-groups to each device exclusively and merge partial results to final results properly. To achieve this goal they introduced a novel idea of partition original data-parallel computing kernel into Partition Ready Kernel and Merge Kernel, which can guarantee the partial output are merged correctly and efficiently.

3 Weakness

1. This paper is too technique-oriented and avoid of handling complex kernel mapping. It focuses on the engineer details about how to assign more fine grained work group to devices, and how to merge partial output correctly and efficiently with regarding to the memory address space.

2. The scope of this paper is also single application, it didn’t consider the multiple application scenario.