GPU Kernels

The gpu\_testfunction kernel performs the computation on the GPU. It takes three input arrays (a, b, and c) and an integer n representing the size of the arrays. Each thread in the kernel calculates the value of a[id], b[id], and c[id] where id is the unique thread index.1

The gpu\_testfunction\_uni\_mem kernel also performs the computation on the GPU, but it uses unified memory, which allows the GPU and CPU to access the same memory location without explicit data transfers.

CPU Function

The cpu\_testfunction function is a CPU-based implementation of the same mathematical operation, where the values are calculated in a loop on the CPU.

Main Function

The main function is the entry point of the program. It allocates memory for the input and output arrays on both the CPU and GPU, and then performs the following steps:

Calls the cpu\_testfunction to get the reference result. Copies the input data from the CPU to the GPU using cudaMemcpy. Launches the gpu\_testfunction kernel on the GPU. Copies the result from the GPU to the CPU using cudaMemcpy. Launches the gpu\_testfunction\_uni\_mem kernel on the GPU using unified memory. Frees the allocated memory on both the CPU and GPU.

The purpose of this code is to find the difference between GPU and CPU performance issues. And also we used unite memory here.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Observations-

 gpu\_testfunction\_uni\_mem kernel takes 83.6% of the total time (562.874 μs).

 Host-to-Device memory copy takes 10.4% of the total time (70.272 μs).

 Device-to-Host memory copy takes 4.8% of the total time (32.415 μs)

gpu\_testfunction\_uni\_mem:

* The kernel is executed once (1 instance).
* The executed instructions per cycle (IPC) is 0.4, indicating relatively low instruction-level parallelism.
* The SM (Streaming Multiprocessor) utilization is 20.2%, suggesting that the kernel is not fully utilizing the GPU's compute resources.
* The memory throughput is 77.67 MB/s, and the memory utilization (Mem Busy) is 22.26%, indicating moderate memory access activity.
* The L1/TEX cache hit rate is 25%, suggesting room for improvement in cache utilization.
* The L2 cache hit rate is 99.01%, which is good

gpu\_testfunction:

* The performance metrics are similar to gpu\_testfunction\_uni\_mem, with slightly higher SM utilization (20.28%) and memory throughput (77.67 MB/s).

Points to be considered and improvements needed:

*  The gpu\_testfunction\_uni\_mem kernel dominates the execution time, suggesting it is the primary computational kernel in your application.
*  Both kernels seem to have relatively low instruction-level parallelism (IPC) and moderate GPU utilization (SM Busy), indicating potential performance bottlenecks or opportunities for optimization.
*  Memory access patterns appear to be moderately efficient, with reasonable memory throughput and L2 cache hit rates. However, the low L1/TEX cache hit rates suggest potential improvements in data access patterns or cache utilization.
*  The memory copy operations (Host-to-Device and Device-to-Host) contribute a non-negligible portion of the total execution time, suggesting potential bottlenecks in data transfer between the host and device.