CUDA Homework Assignment 5

Hao-Tse , Hsiao

B12902100

Abstract. In Homework Assignment 5, we compute the thermal equilibrium temperature distribution on a square plate. We evaluate the performance of the Jacobi iterative method for solving the steady-state heat distribution on a 1024×1024 square plate, using CPU, single GPU, and dual GPU implementations. Additionally, this experiment investigates the optimal block size configuration to maximize execution speedup.

1 Task Description

We begin by initializing the plate of size 1024×1024. The top edge of the plate is set to 400 K, while the rest of the boundary is held at 273 K. The thermal equilibrium temperature distribution on a square plate is computed using both CPU and GPU implementations.

After completing the simulation, the results were exported to a ‘.txt‘ file and visualized using the ‘plot_heatmap.py‘ script.

2 Task Result

After performing the vector computation on both the CPU and two GPUs, we compiled the results in the following table. CPU execution time is used as the baseline for calculating speedup. Block size refers to the number of threads per dimension; for example, a block size of 4 corresponds to $4 \times 4 = 16$ threads per block. Each value in the table represents the execution speedup. The relative difference between the GPU and CPU computed results is illustrated in Figures 1 and 2. The simulation results of the CPU and GPU implementations are highly consistent, as illustrated in Figures 1 and 2.

Table 1. GPU Execution Speedup Relative to CPU (Higher is Better)

Block size	single GPU	two GPUs
4	1.6123	8.7068
8	2.2758	31.8418
16	2.2215	**37.8705**
32	**2.2931**	x

Table 1 summarizes the execution speedup of various block and grid configurations on the GPU, normalized against CPU execution time. Each value represents the relative performance improvement. The best performance on a single GPU was observed with a block size of 32 × 32 threads per block. The best performance on the dual-GPU setup was observed with a block size of 16 × 16 threads per block.

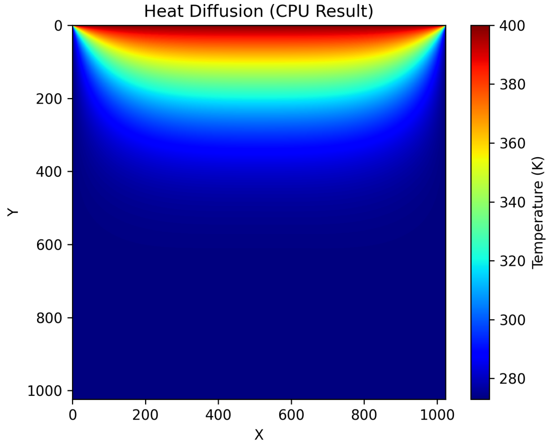

Fig. 1. Heatmap of CPU-computed steady-state temperature distribution

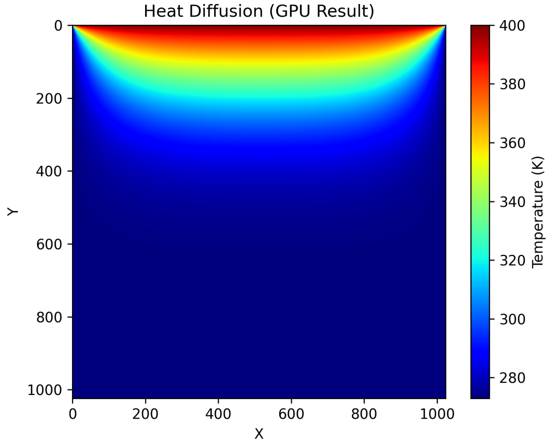

Fig. 2. Heatmap of GPU-computed temperature distribution using 16 × 16 blocks

3 Discussion

The first part of the computation utilizes only a single GPU, while the second part distributes the workload across two GPUs. Each GPU is assigned half of the plate to compute in parallel using CUDA kernels.

The results show that GPU-based computation is significantly faster than CPU-based computation. In particular, the dual-GPU implementation achieves even greater speedup due to increased parallelism.

We also observed that the choice of block size affects performance. If the block size is too large, it may reduce parallelism due to thread divergence or shared memory contention. On the other hand, if it is too small, the overhead of thread management may dominate. Optimal performance was achieved using a block size of 16×16 for two GPUs and 32×32 for a single GPU.

These findings highlight the effectiveness of GPU parallelism for solving large-scale numerical problems like heat diffusion.

4 References

1. Introduction to CUDA Parallel Programming — Homework Assignment 5