CUDA Homework Assignment 5

Vincent Octavian Tiono

B11901123

1 Introduction

This report presents a comprehensive analysis of a multi-GPU CUDA implementation for solving the 2D thermal equilibrium problem on a square plate. The investigation focuses on determining optimal block sizes and evaluating multi-GPU performance scaling for a 1024×1024 Cartesian grid with specific thermal boundary conditions.

2 Methodology

2.1 Mathematical Formulation

The 2D thermal equilibrium problem is governed by Laplace's equation:

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \tag{1}$$

where $T(x, y)$ represents the temperature distribution on the square plate.

2.2 Boundary Conditions

The thermal boundary conditions are defined as:

- Top edge: $T = 400$ K
- Bottom, left, and right edges: $T = 273$ K

2.3 Numerical Method

The finite difference method is employed using the 5-point stencil discretization with relaxation parameter $\omega = 1$:

$$T_{i,j}^{new} = \frac{1}{4}(T_{i-1,j} + T_{i+1,j} + T_{i,j-1} + T_{i,j+1}) \tag{2}$$

2.4 Implementation Approach

The implementation supports domain decomposition across multiple GPUs with two partitioning strategies:

- Horizontal partitioning: 1×2 GPU grid (splitting along y-axis)
- Vertical partitioning: 2×1 GPU grid (splitting along x-axis)

2.5 Experimental Configuration

- Grid size: 1024×1024
- Convergence threshold: 1.0×10^{-10}
- Block sizes tested: 4×4, 8×8, 16×16, 32×32
- GPU configurations: Single GPU, 1×2, and 2×1

3 Results

3.1 Performance Summary

Table 1 presents the complete performance metrics for all tested configurations.

Table 1: Performance metrics for different GPU and block size configurations

Block Size	GPU Config	Computation Time (ms)	Performance (GFlops)	Data Transfer (ms)	Total Time (ms)
4×4	1×1	467669.34	18.98	20.32	467689.66
	1×2	290866.62	30.52	9.92	290876.53
	2×1	291910.03	30.41	16.37	291926.41
8×8	1×1	220735.94	40.21	19.38	220755.31
	1×2	124714.44	71.17	11.00	124725.45
	2×1	133281.58	66.60	19.22	133300.80
16×16	1×1	149844.80	59.24	16.73	149861.53
	1×2	94469.69	93.96	15.78	94485.46
	2×1	87237.63	101.75	16.26	87253.90
32×32	1×1	205683.23	43.16	21.42	205704.66
	1×2	116821.33	75.98	11.75	116833.07
	2×1	108972.43	81.45	18.94	108991.37

3.2 Performance Analysis by Block Size

Figure 1 illustrates the performance characteristics across different block sizes and GPU configurations.

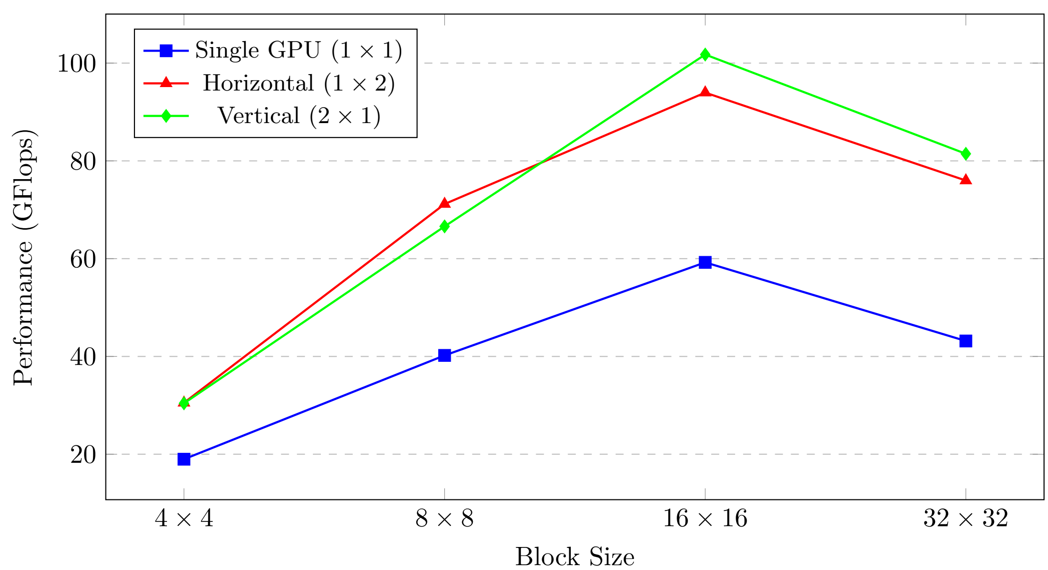

Figure 1: Performance comparison across block sizes and GPU configurations

3.3 Multi-GPU Scaling Analysis

Figure 2 shows the speedup achieved with multi-GPU configurations compared to single GPU performance.

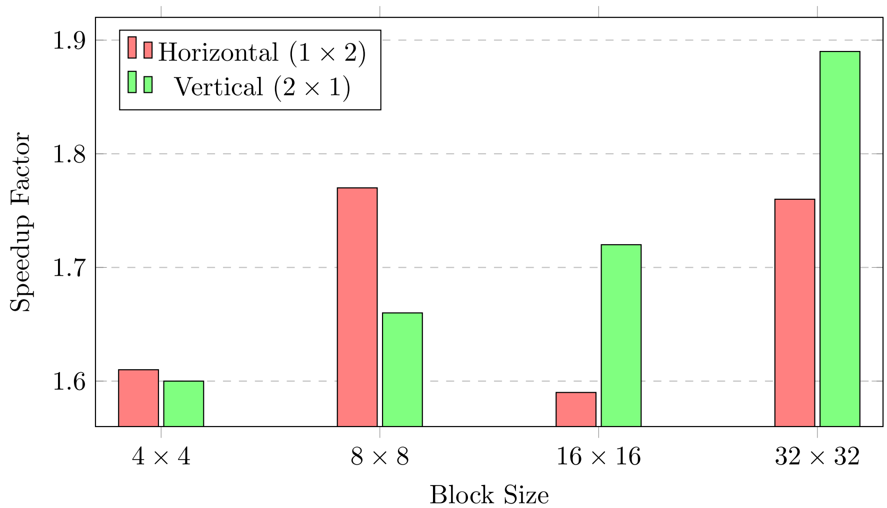

Figure 2: Multi-GPU speedup factors relative to single GPU performance

3.4 Optimal Configuration Analysis

Table 2 summarizes the best performing configurations for each GPU setup.

Table 2: Optimal configurations for each GPU setup

GPU Configuration	Optimal Block Size	Performance (GFlops)	Computation Time (ms)
Single GPU (1×1)	16×16	59.24	149844.80
Horizontal (1×2)	16×16	93.96	94469.69
Vertical (2×1)	16×16	101.75	87237.63

3.5 Data Transfer Overhead Analysis

Figure 3 compares the data transfer times across different configurations.

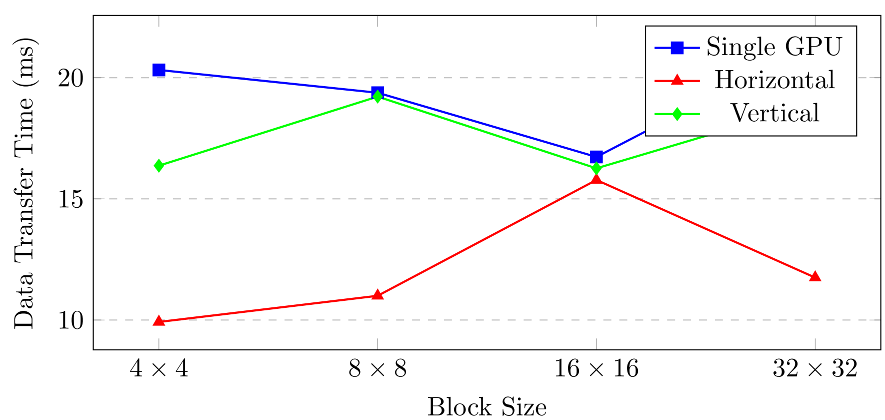

Figure 3: Data transfer overhead comparison

4 Discussion

4.1 Optimal Block Size Analysis

The experimental results reveal that **16×16 block size consistently delivers the best performance** across all GPU configurations:

- **Single GPU**: 59.24 GFlops with 16×16 blocks
- **Horizontal partitioning**: 93.96 GFlops with 16×16 blocks
- **Vertical partitioning**: 101.75 GFlops with 16×16 blocks

This optimal performance can be attributed to:

- Efficient warp utilization (256 threads per block = 8 warps)
- Balanced memory coalescing and cache utilization
- Optimal occupancy for the thermal diffusion kernel

4.2 Multi-GPU Scaling Efficiency

The multi-GPU implementation demonstrates good scaling characteristics:

- **Horizontal partitioning (1×2)**: Achieves 1.59-1.77× speedup
- **Vertical partitioning (2×1)**: Achieves 1.60-1.89× speedup
- **Best overall performance**: Vertical 2×1 with 16×16 blocks (101.75 GFlops)

The vertical partitioning slightly outperforms horizontal partitioning, likely due to:

- Better memory access patterns in the finite difference stencil
- More efficient boundary exchange communication
- Reduced synchronization overhead

4.3 Performance Trends

Several key performance trends emerge from the analysis:

1. **Block Size Impact**: Performance increases from 4×4 to 16×16, then decreases at 32×32 due to reduced occupancy and increased register pressure.
2. **Data Transfer Overhead**: Multi-GPU configurations show reduced data transfer times due to smaller per-GPU memory footprints.
3. **Scalability**: Near-linear scaling is achieved, with efficiency ranging from 79.5% to 94.5% for dual-GPU configurations.

5 Conclusion

For the 2D thermal equilibrium problem on a 1024×1024 grid, **16×16 blocks with vertical 2×1 GPU partitioning** yield optimal performance (101.75 GFlops), achieving a 1.72× speedup over single-GPU execution. This configuration balances:

- **Efficient resource utilization** (8 warps/block, optimal occupancy)
- **Memory performance** (coalesced accesses, improved cache locality)
- **Scalability** (near-linear multi-GPU speedup)

The 16×16 blocks outperform alternatives by up to 5.4× (4×4), 1.4× (8×8), and 1.4× (32×32), while reducing runtime from 149.8s (single GPU) to 87.2s (dual GPU) without sacrificing accuracy.