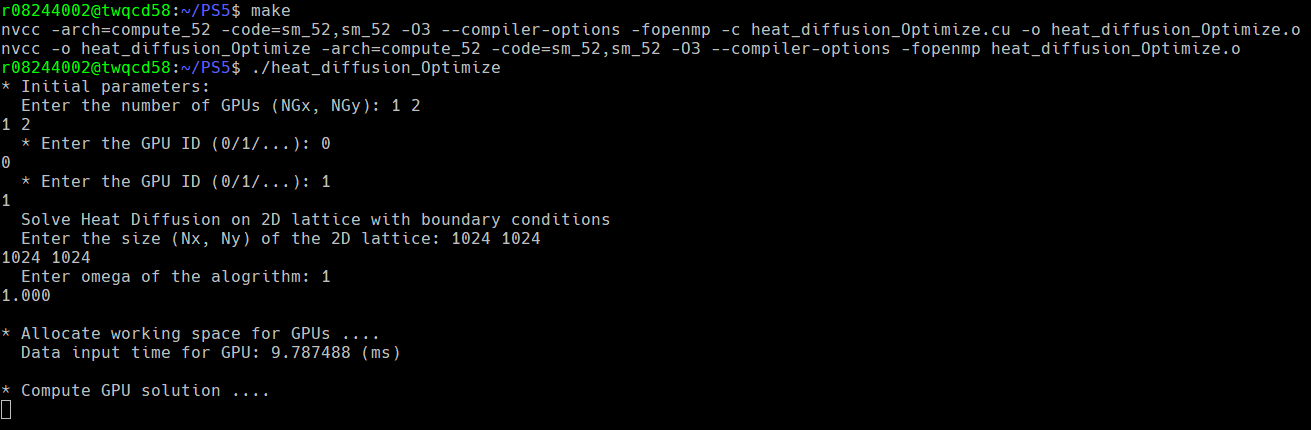
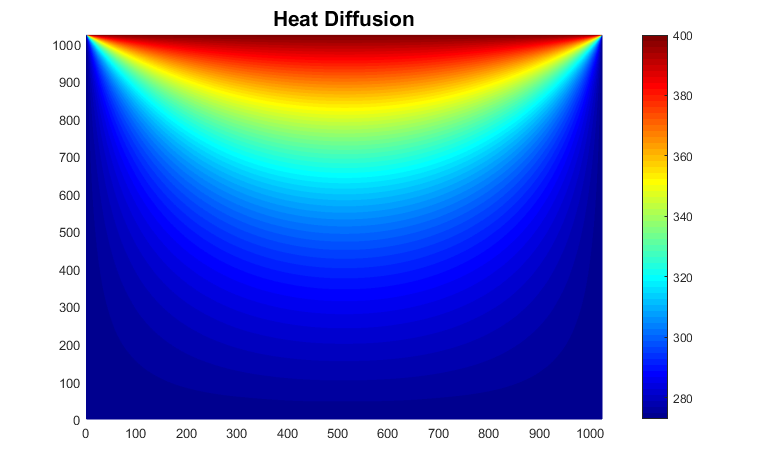
**CUDA Parallel Programming Problem Set 5**

R08244002 Shin-Rong Tsai

* **Heat Diffusion with**

1. **Result**

“Slightly” change the code *laplace2D\_NGPU.cu* given in the course and saved it as *heat\_diffusion.cu*, then compiled and run. To find the optimal block size, add a loop to loop through all the block size, saved in file *heat\_diffusion\_Optimize.cu*.

Which gives the final result:

Compare total time used of 2 GPU with 1 GPU, unit in second:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Block Size  Time Used (s) | (4, 4) | (8, 8) | (16, 16) | (32, 32) |
| 2 GPU | 485.80 | 222.54 | 233.91 | 242.52 |
| 1 GPU | 791.82 | 355.68 | 338.89 | 405.44 |
| Speed Up | 1.63 | 1.60 | 1.45 | 1.67 |

1. **Discussion**
2. The optimal block size for 2 GPU is (8,8), 1 GPU is (16,16). It is quite interesting that all the algorithm about lattice has optimal block size either be 16 or near 16.
3. The speed up rate between 2 GPU and 1 GPU is far from 2, which means effects like direct access to other GPU’s memory may take up some time.
4. Compare different with block size (16, 16) and 2 GPU:

|  |  |  |
| --- | --- | --- |
|  | Iteration | Total Time Used (sec) |
| 1.5 |  |  |
| 1 | 1214037 | 233.91 |
| 0.9 | 1274320 | 178.92 |

Though has more iterations, it uses less time. I think it is just because of how the hardware performed then. And since

if we fixed and , bigger means bigger velocity, and faster diffusion rate. So it needs less iteration to reach equilibrium. and give the same lattice result, even though they used different iterations.

But if the diffusion rate is too big that the neighbor coordinate cannot cover, then the whole algorithm breaks down. Just like case, which gives a wrong simulation.