CS 475/575 -- Spring Quarter 2022

Project #5

CUDA: Monte Carlo Simulation

1. Tell what machine you ran this on

I ran the code on the Rabbit server. The probability is 22.24%

1. Show the table and the two graphs

|  |  |  |  |
| --- | --- | --- | --- |
| NumberofTrials | Blocksize | MegaTrials/Second | Probability |
| 1024 | 8 | 8.9586 | 24.22% |
| 1024 | 32 | 12.3124 | 21.78% |
| 1024 | 128 | 10.4473 | 25.00% |
| 4096 | 8 | 50.7132 | 22.71% |
| 4096 | 32 | 39.4453 | 24.10% |
| 4096 | 128 | 47.6723 | 21.46% |
| 16384 | 8 | 137.4128 | 22.16% |
| 16384 | 32 | 101.4665 | 22.46% |
| 16384 | 128 | 211.047 | 22.22% |
| 65536 | 8 | 350.0855 | 22.68% |
| 65536 | 32 | 450.9027 | 22.70% |
| 65536 | 128 | 544.6809 | 22.62% |
| 262144 | 8 | 735.1701 | 22.30% |
| 262144 | 32 | 1671.1546 | 22.50% |
| 262144 | 128 | 1948.6204 | 22.61% |
| 1048576 | 8 | 978.6459 | 22.56% |
| 1048576 | 32 | 2720.2391 | 22.50% |
| 1048576 | 128 | 3751.7747 | 22.53% |
| 2097152 | 8 | 1050.3237 | 22.51% |
| 2097152 | 32 | 3319.9594 | 22.51% |
| 2097152 | 128 | 4966.73 | 22.53% |
| 4194304 | 8 | 1092.0119 | 22.49% |
| 4194304 | 32 | 3688.0135 | 22.50% |
| 4194304 | 128 | 6279.8005 | 22.53% |
| 8388608 | 8 | 1126.9485 | 22.49% |
| 8388608 | 32 | 4076.1924 | 22.50% |
| 8388608 | 128 | 7427.4381 | 22.51% |
| 16777216 | 8 | 1134.3536 | 22.51% |
| 16777216 | 32 | 4134.209 | 22.50% |
| 16777216 | 128 | 7919.8783 | 22.50% |

1. What patterns are you seeing in the performance curves?

In the performance vs Block size graph, there was a kind of a minute growth in the performance till 65536 NumTries and after those block sizes there was constant growth in performances.

In the performance vs Num Tries graph, there was a slow increase in all the performances for every block size and remains constant once they have reached a saturation point.

1. Why do you think the patterns look this way?

Based on the block size, the number of threads allocated. So, as the block size increases, threads also increase, which will exponentially improve the overall performance.

1. Why is a BLOCKSIZE of 8 so much worse than the others?

Based on the block size itself number of GPU threads are created, so if we consider a large block size less number threads are created but if there are small block size then a large number of GPU threads are created and there may be waiting for their turn to get executed.

1. How do these performance results compare with what you got in Project #1? Why?

During our project#1, the max performance was near 200 MegaTrials/Second but now the performance has raised exponentially like the current max performance using GPU has gone higher than 7000 MegaTrials/Second. This is because of the individual cache line which helps in working independently without memory locks and helps in faster processing.

1. What does this mean for the proper use of GPU parallel computing?

Using the GPU accelerates the parallel computation by dividing the whole process into blocks and starts working as an individual code like a processor while also has an individual cache line. The block computation is optimized so that overall performance using the GPU is faster.