High Performance Computing Assignment 4

Yijian Xie

April 15, 2019

1 Matrix-vector operations on a GPU

I implemented both the vector-vector and matrix-vector multiplication.

Vector-vector multiplication

The result of GPU version program on TITAN X and TITAN Z were not correct and I did not figure out why. The length of vectors are 2^{25}.

CPU	Bandwidth	GPU	Bandwidth
Xeon E5-2680 v3 @ 2.50GHz	27.16GB/s	GTX TITAN Black	58.76GB/s
Xeon E5-2660 v3 @ 2.60GHz	23.66GB/s	RTX 2080 Ti	98.07GB/s
Xeon Gold 5118 @ 2.30GHz	9.59GB/s	GTX TITAN V	389.59GB/s
Xeon Gold 5118 @ 2.30GHz	7.25GB/s	GTX TITAN X	N/A
Xeon E5-2650 v2 @ 2.60GHz	13.40GB/s	GTX TITAN Z	N/A

Matrix-vector multiplication

The size of matrix is $2^{11} * 2^{11}$, and the length of vector is 2^{11}. Because there are 2^{11} vector-vector multiplication, the overhead becomes much larger than simply calculate one vector-vector multiplication. There were significant differences on different servers, maybe due to different workloads on each server.

CPU	Bandwidth	GPU	Bandwidth
Xeon E5-2680 v3 @ 2.50GHz	0.91GB/s	GTX TITAN Black	0.51GB/s
Xeon E5-2660 v3 @ 2.60GHz	0.08GB/s	RTX 2080 Ti	0.66GB/s
Xeon Gold 5118 @ 2.30GHz	0.46GB/s	GTX TITAN V	0.92GB/s
Xeon Gold 5118 @ 2.30GHz	0.004GB/s	GTX TITAN X	0.012GB/s
Xeon E5-2650 v2 @ 2.60GHz	0.005GB/s	GTX TITAN Z	0.009GB/s

2 2D Jacobi method on a GPU

The size of 2D-matrix is $2000 * 2000$, and the matrix will be iterated 2000 times.

GPU	Time
GTX TITAN Black	6.67s
RTX 2080 Ti	5.89s
GTX TITAN V	5.80s
GTX TITAN X	8.53s
GTX TITAN Z	11.91s

3 Final project

My teammate Wenjun Qu and I will implement the "Game of Life". The simple idea is to split the board into $m*n$ pieces and calculate them parallelly, while the information of the edge of each piece needs to be passed via MPI. We are also interested in the circumstances where the board has an irregular shape or the board is formed by triangles or hexagons.