GPU Programming, Spring 2019

Patric Manner (37208)

February 7, 2019

Hardware

Asus N56VZ laptop with a Nvidia GeForce GT 650M graphics card; 384 CUDA cores and a compute capability of 3.0 (Kepler architecture).

Design and execution

General

- Removed the 100000 from the beginning of each file

CPU side

- Allocated unified memory with cudaMallocManaged() for the value arrays and histograms
- Read values from the files, convert to radians and place into 4 arrays (ascension and declination separately for both files)
- Initialize the histogram arrays with zeros
- Start a kernel for each histogram (DD, DR, RR)

GPU side

- Find out global thread index and check if it's below N
- Each thread (with index below N) multiplies one value from the first galaxy list (it's index) with all of the values in the second galaxy list
- Use given formula to calculate angle between two galaxies
- Convert to degrees, determine bin number and add to bin with atomicAdd

Back to the CPU side

- Wait for everything to finish calculating (cudaDeviceSynchronize())

- Calculate the omega values, which is the differences between two equally large sets of galaxies, with the three histograms DD, DR and RR
- Print out the 15 first omega values for analysis
- (Optional) Include the debug()-function to write everything to a file and run a python script to visualize the histograms

Program specs and results

- Threads in block: 512
- Blocks in grid: 196
- Average running time: 35s
- Registers used: 22
- Omega results point towards the real galaxies being non-randomly distributed

Problems encountered

- Got bus errors due to the acos function getting input values greater than 1, which led to that the program tried to access a negative array index in the histogram
- Got bus error due to not synchronizing the devices
- Took a long time to realize that I was printing %d even though the result of the aggregation was correct when checking if the histograms were filled properly