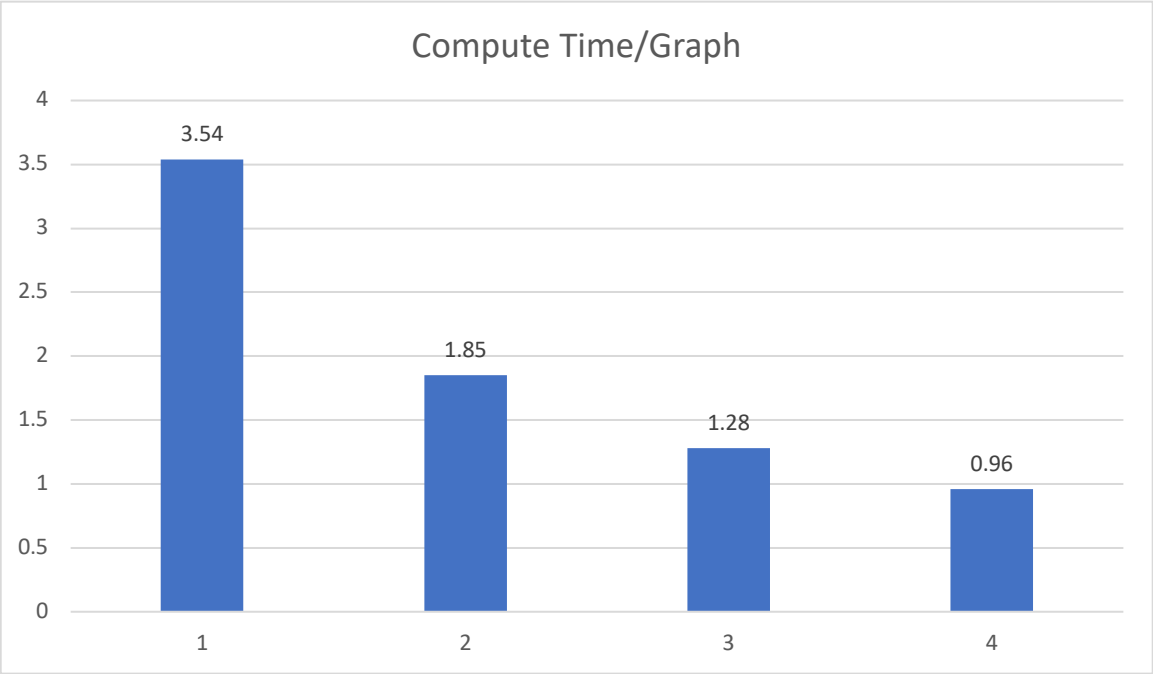
Noah del Angel

Question 1a) What compute times do you get? Present them (in seconds) in a bar graph with the GPU count along the x-axis (increasing from left to right) and the time along the y-axis. Include a label on top of each bar showing the compute time with two digits after the decimal point.



Question 1b) How much faster in terms of speedup is the code on 4 GPUs compared to 1 GPU?

Speed up from 4 GPUs to one is 3.6875.

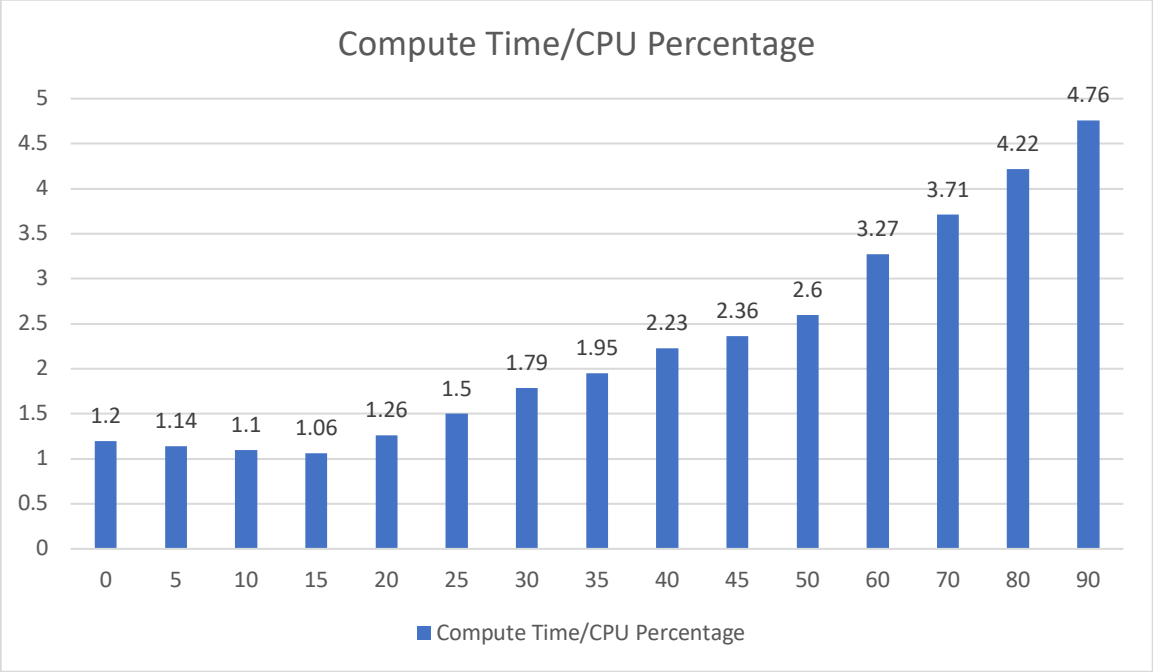
Question 1c) Explain why it is better to use a cyclic distribution of the work across the GPUs than a blocked distribution.

Cyclic distributions do a better job of load balancing, compared to blocked distribution, when the size of the work is unknown.

Question 1d) How many CPU sockets does a GPU compute node of Frontera have? How many GPUs does a GPU compute node of Frontera have?

GPU compute nodes have 16 CPU sockets and with 4 GPU's per compute node.

Question 2a) What compute times do you get? Present them (in seconds) in a bar graph with the CPU workload percentage along the x-axis (increasing from left to right) and the time along the y-axis. Include a label on top of each bar showing the compute time with two digits after the decimal point.



Question 2b) Which CPU percentage results in the highest performance?

The CPU percentage with the highest performance is 15 percent, with a compute time of 1.06 seconds.

Question 2c) With a CPU percentage of 10, how many frames are assigned to the CPU of each compute node and how many to the GPU?

Each CPU on the compute will receive 12 frames, and each GPU will receive 116 frames. In total, the CPU will compute 48 frames and the GPU will compute 464 frames.

Question 2d) For inputs with relatively few frames, it is better to parallelize the middle (for-row) loop than the outer (for-frame) loop. Explain why that is the case.

When paralleling the for-row low, assuming there is a small number of frames, there is little overhead, and we can parallelize more of the work. If there is many frames, parallelizing the for-row produces too much overhead.

Question 2e) Does the code use a blocked, cyclic, or block-cyclic distribution when assigning work to the CPUs and GPUs?

For both the CPU & the GPU we use the blocked distribution when assigning work.