Benjy Katz

HW 3

Benchmark (image) (scaleFactor) Mode Cnt Score Error Units

Benchmarks.testSerial earth.png 2 avgt 10 134.392 ± 5.192 ms/op

Benchmarks.testSerial earth.png 3 avgt 10 297.144 ± 16.531 ms/op

Benchmarks.testSerial earth.png 4 avgt 10 533.611 ± 11.922 ms/op

Benchmarks.testSerial earth.png 10 avgt 10 3438.660 ± 160.148 ms/op

The first parallelization I did was to break the image into rows and expand going through every pixel of the row. This leads to hundreds of tasks, each one short. To parallelize the brightening I performed the parallelization the same way after the initial resizing was complete. This ended up with a significant speed up for images to that only needed to double or triple in size but no speed ups when the image needed to be significantly larger. I believe this has to do with the fact that this large new image takes up much more space in memory which causes the hand off from one thread to the other to be very costly.

Benchmarks.testParallel earth.png 2 avgt 10 53.732 ± 1.077 ms/op

Benchmarks.testParallel earth.png 3 avgt 10 147.818 ± 3.476 ms/op

Benchmarks.testParallel earth.png 4 avgt 10 308.245 ± 19.782 ms/op

Benchmarks.testParallel earth.png 10 avgt 10 3653.386 ± 555.571 ms/op

I thought to change the sub problem by breaking the image into quadrants. I thought that if I changed the problem to the same number of subregions as the number of cores on my computer I could speed up the processing. Unfortunately, this solution performed roughly the same as the previous solution.

Benchmarks.testParallelRegions earth.png 2 avgt 10 58.032 ± 4.777 ms/op

Benchmarks.testParallelRegions earth.png 3 avgt 10 149.778 ± 4.440 ms/op

Benchmarks.testParallelRegions earth.png 4 avgt 10 314.573 ± 6.261 ms/op

Benchmarks.testParallelRegions earth.png 10 avgt 10 3685.541 ± 112.570 ms/op

Here I changed the thread pool to a Cache as opposed to a fixed to try and benefit from work stealing.

Benchmarks.testParallelCache earth.png 2 avgt 10 64.368 ± 5.016 ms/op

Benchmarks.testParallelCache earth.png 3 avgt 10 153.253 ± 2.521 ms/op

Benchmarks.testParallelCache earth.png 4 avgt 10 310.963 ± 5.541 ms/op

Benchmarks.testParallelCache earth.png 10 avgt 10 3632.226 ± 163.580 ms/op

Here I first brighten the smaller image and then expand it which allows for much less work to be done since we do not need to go over every pixel of the larger image to brighten it. When attempting to perform them at the same time there is disaster since the image cannot be resized and brightened at the same time without knowing which will come first. This was seen in the ParallelTransformSameTime. The image is an absolute mess.

Benchmarks.testParallelBrightenFirst earth.png 2 avgt 10 38.262 ± 10.310 ms/op

Benchmarks.testParallelBrightenFirst earth.png 3 avgt 10 97.523 ± 1.202 ms/op

Benchmarks.testParallelBrightenFirst earth.png 4 avgt 10 190.959 ± 2.422 ms/op

Benchmarks.testParallelBrightenFirst earth.png 10 avgt 10 2753.487 ± 33.149 ms/op