MPI Ex. 3 Analysis
Brea Koenes / cs374 / 09/23/21

On lab time, the probability the burn-percentage begins to accelerate is at about 0.36. The same is true of 20x20 on Borg. Borg 40x40 begins to accelerate around 0.44 and Borg 80x80 begins to accelerate around 0.45. On lab time, the probability the burn-percentage begins to decelerate at is about 0.64. 20x20 on Borg decelerates around 0.67. Borg 40x40 begins to decelerate around 0.65 and Borg 80x80 begins to decelerate around 0.64. The reason why this is relatively unchanging is because the increments that the numbers increase in, AKA the curve of the line, should not vary much. The numeric value of the points should change as it's run in different places, but they should increase and decrease in relatively similar amounts; it is the same program with the same time complexity run on Borg and the lab. The inflection point where the 20x20 lab time begins to increase rapidly is 0.45. Borg's 20x20 and 40x40 is 0.45 as well. Borg's 80x80's inflection point is 0.48. This is of interest because it shows how the program's burn percentage accelerates in different conditions and even estimates its inflection point with higher forest sizes.

On lab time, 0.56 is the burn-probability that the fire burns the longest. For Borg 20x20, it is 0.55. For Borg 40x40, it is 0.53. For Borg 80x80, it is 0.52. At the same time, burn probabilities are also increasing. Looking at the chart, it is at the midpoint of the probabilities—around 0.5—that the burn times are the longest. Burn probabilities cause the burn time to increase or level out. Also, the burn percentage burned may be correlated to burn time. This is just an idea from what I observed. At the peak of the burn times of each graph, the burn percent is rapidly increasing to its peak. For example, Borg 20x20 peaks around 0.56 within 32 seconds and the burn percentage at 0.56 is 62% burned and increasing quickly. The burn % on the lab and Borg is also leveling out to 1 and the burn times are decreasing as the probabilities increase. For example, on lab time 20x20, the burn time decreases at 0.6 and the % burned decreases at 0.64. Simply put, they accelerate at similar probabilities.

The performance of my program on the cluster increases as the number of processes increases. Specifically, the runtime with 1 process on Borg 20x20 is 27.5 seconds. The runtime with 256 processes is 0.17 seconds. As the forest size increases, it follows this same pattern. For example, 1 process is 155.8 seconds on Borg 40x40 and 256 processes is 0.80 seconds. Due to the beautiful MPI reduction pattern and parallel programming, the runtimes decrease steadily as the number of processes increase. I intentionally reordered firestarter.c in order to make it more parallel, shifting "for" loops and using MPI functions.

The size of the forest makes the runtimes increase. As the example above states, Borg 20x20 with 1 process is 27.5 seconds. 1 process on Borg 40x40 is 155.8 seconds. This is expected because it takes more time to run processes with a larger forest. The slope of the runtime compared to processes is unchanging for all of them, just the points differ—also as expected.

The same goes as above for the burn percentages with probability and burn times with probability. The shape of the curves between Borg 20x20, 40x40, and 80x80 are basically the same. However, their points differ; as the forest increases, burn time is longer. On 20x20, the highest burn time is about 32 seconds. On 80x80, it is about 135 seconds. Also, the acceleration of the percent burned becomes steeper as the forest size increases. Borg 20x20's curve ranges from 0.28-0.76. Borg 80x80's ranges from 0.44-0.72. All of this is expected, because a bigger forest takes longer to burn and the percentage burned increases quicker on a larger forest.

My model collaborates with Amdahl's and Gustafson's Laws. My program collaborates with Amdahl's law because adding more processors may not make the program run faster. My program is not perfectly parallel, the non-parallel part of my program may cause a latency in the speedup. Gustafson's Law is also true as the speedup increases as the forest size increases. Specific comparisons are given in the paragraphs above. It speeds up also due to the parallel parts of my program.