Project 2 grading policy :

For Checkpoint 1:

For getting any points:

- Your program should not incur a segmentation fault for any of the trace files.
- Your program should not get stuck (e.g., infinite loop) for any of the trace files.
- Your program should not crash for any of the trace files.

If your program meets the conditions mentioned above, the total points for Checkpoint 1 will be determined based on the number of trace files that you pass. Note that we are giving partial credits even if your program has certain types of bugs other than those mentioned above. E.g., your program may have overlapping blocks for some of the trace files and you will get partial credits for the trace files on which your program works properly.

How your program's performance results translate into grades for Checkpoints 2 and 3:

Below, I will try to explain this translation for **Checkpoint 3.** A similar logic applies to Checkpoint 2. Please refer to the image below.

You have to consider 3 types of quantities:

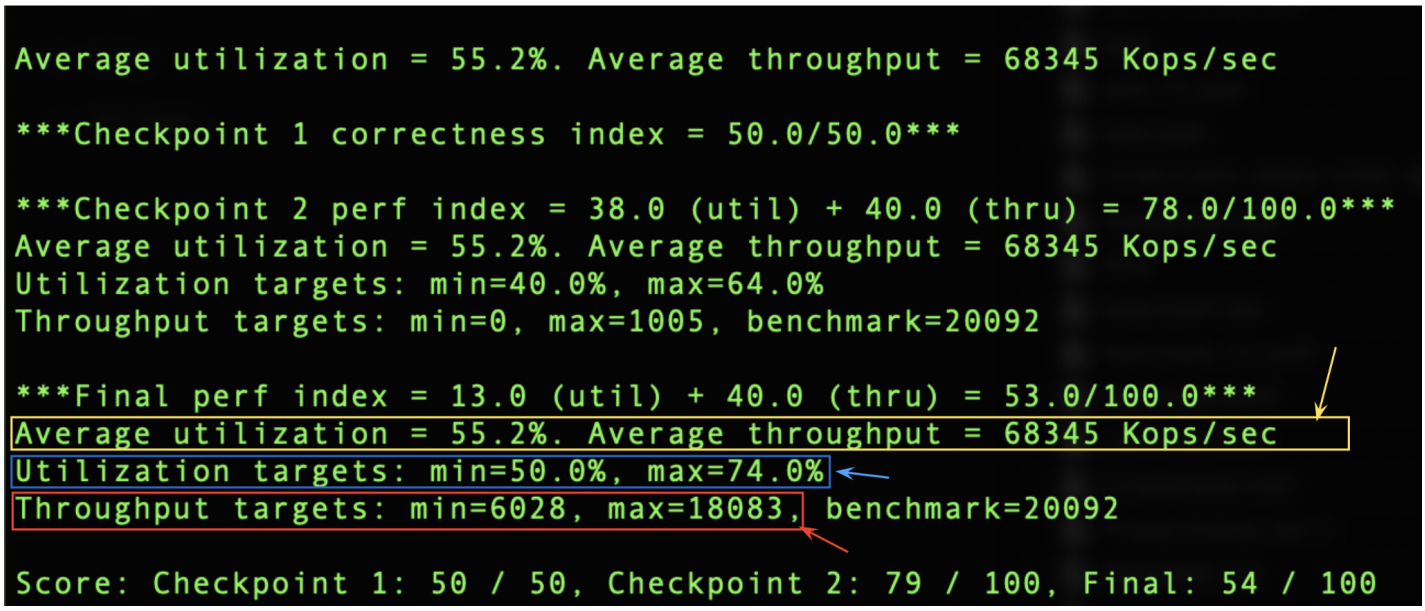
1. Average utilization and average throughput (boxed in yellow in the image below)
2. Utilization targets (boxed in blue in the image below)
3. Throughput targets (boxed in red in the image below)

The average utilization and the average throughput represent the performance offered by your program. The averages are calculated across the traces you are given.

Your total score comprises two parts: points for utilization and points for throughout.

The target minimum utilization is the minimum utilization that your program must achieve in order for you to get any points for utilization. So if your average utilization is less than the target minimum utilization, you will get 0 points for utilization. The target maximum utilization is the minimum value of utilization that your program must achieve in order for you to get full points for utilization. So if your average utilization is greater than the target maximum utilization, you will get full points for utilization. If your utilization lies between these two extremes, you will get points for utilization per the formula that appears below.

The target minimum throughput is the minimum value of throughput that your program must achieve in order for you to get any points for throughput. So if your average throughput is less than the target minimum throughput, you will get 0 points for throughput. The target maximum throughput is the minimum throughput that your program must achieve in order for you to get full points for throughput. So if your average throughput is greater than the target maximum throughput, you will get full points for throughput.



The following pseudocode describes how your points will be calculated for utilization and throughput:

```
if (AVG < MIN) {
    ZERO POINTS;
} else if (AVG > MAX) {
    Full Points;
} else {
    POINTS = 100*(AVG - MIN) / (MAX - MIN);
}
```

Let's use the above to calculate a sample points for utilization corresponding to the above image.

Average utilization (AVG) = 55.2%
Target minimum utilization (MIN) = 50%
Target maximum utilization (MAX) = 74.0%

Since the average utilization (55.2%) is greater than the target minimum utilization (50%) you will get some points for utilization. Since the average utilization (55.2%) is less than the target

maximum utilization (74%), you will not get full points for utilization. Your points for utilization will be:

POINTS = (AVG - MIN) / (MAX - MIN)
POINTS = (55.2-50) / (74-50)
POINTS = 5.2/24

So your score for utilization will be 0.216.

Now let's calculate your points for throughput:

Average throughput (AVG) = 68345
Target minimum throughput (MIN) = 6028
Target maximum throughput (MAX) = 18083

So you get 100% points for throughput.

Finally, since the weight for utilization is 60% and that for throughput is 40%, your final score will be: 0.216*60+40 = 53

Important note about throughput: unlike utilization, the throughput intimately depends on the resource contention on your machine as well as the machine itself. So if you are running on a mostly vacant machine, you will see a better throughput than if you are running an hour before the deadline with 100 other students also running their programs on the same machine. Even a highly efficient program may see poor throughput due to such contention. We will be doing our own testing on lightly loaded machines to assess your throughput.