EVAL PROBLEM 1

a) CPU clock speed using the SLEEP method

WaitMethod: SLEEP WaitTime: 1 0 ClocksElapsed: 1113877651 ClockSpeed: 1113.877651 MHz	WaitMethod: SLEEP WaitTime: 6 0 ClocksElapsed: 6679553916 ClockSpeed: 1113.258986 MHz
WaitMethod: SLEEP WaitTime: 2 0 ClocksElapsed: 2227485027 ClockSpeed: 1113.742514 MHz	WaitMethod: SLEEP WaitTime: 7 0 ClocksElapsed: 7793940907 ClockSpeed: 1113.420130 MHz
WaitMethod: SLEEP WaitTime: 3 0 ClocksElapsed: 3340930716 ClockSpeed: 1113.643572 MHz	WaitMethod: SLEEP WaitTime: 8 0 ClocksElapsed: 8909895277 ClockSpeed: 1113.736910 MHz
WaitMethod: SLEEP WaitTime: 4 0 ClocksElapsed: 4453027754 ClockSpeed: 1113.256938 MHz	WaitMethod: SLEEP WaitTime: 9 0 ClocksElapsed: 10022689628 ClockSpeed: 1113.632181 MHz
WaitMethod: SLEEP WaitTime: 5 0 ClocksElapsed: 5567974837 ClockSpeed: 1113.594967 MHz	WaitMethod: SLEEP WaitTime: 10 0 ClocksElapsed: 11130390437 ClockSpeed: 1113.039044 MHz

STATISTICS

Minimum: WaitMethod: SLEEP WaitTime: 10 0 ClocksElapsed: 11130390437 ClockSpeed: 1113.039044 MHz	Maximum: WaitMethod: SLEEP WaitTime: 1 0 ClocksElapsed: 1113877651 ClockSpeed: 1113.877651 MHz
Average: 1113.5202893 MHz	SD: 0.25230400533127

b) CPU clock speed using the BUSYWAIT method

WaitMethod: BUSYWAIT WaitTime: 1 0 ClocksElapsed: 1113718151 ClockSpeed: 1113.718151 MHz WaitMethod: BUSYWAIT WaitTime: 2 0 ClocksElapsed: 2227264679 ClockSpeed: 1113.632339 MHz WaitMethod: BUSYWAIT WaitTime: 3 0 ClocksElapsed: 3340797007 ClockSpeed: 1113.599002 MHz WaitMethod: BUSYWAIT WaitTime: 4 0 ClocksElapsed: 4454386867 ClockSpeed: 1113.596717 MHz WaitMethod: BUSYWAIT WaitTime: 5 0 ClocksElapsed: 5567987520 ClockSpeed: 1113.597504 MHz	WaitMethod: BUSYWAIT WaitTime: 6 0 ClocksElapsed: 6681527874 ClockSpeed: 1113.587979 MHz WaitMethod: BUSYWAIT WaitTime: 7 0 ClocksElapsed: 7795016741 ClockSpeed: 1113.573820 MHz WaitMethod: BUSYWAIT WaitTime: 8 0 ClocksElapsed: 8908845603 ClockSpeed: 1113.605700 MHz WaitMethod: BUSYWAIT WaitTime: 9 0 ClocksElapsed: 10021173266 ClockSpeed: 1113.463696 MHz WaitMethod: BUSYWAIT WaitTime: 10 0 ClocksElapsed: 11135219867 ClockSpeed: 1113.521987 MHz

STATISTICS

Minimum: WaitMethod: BUSYWAIT WaitTime: 9 0 ClocksElapsed: 10021173266 ClockSpeed: 1113.463696 MHz	Maximum: WaitMethod: BUSYWAIT WaitTime: 1 0 ClocksElapsed: 1113718151 ClockSpeed: 1113.718151 MHz
Average: 1113.5896895 MHz	SD: 0.062839163866565

c) Overall results

Which method yielded more precise and stable results?

- Busy wait yielded more precise and stable results as the standard deviation was 0.062839163866565 whereas for sleep the standard deviation was 0.25230400533127.

Which method produced a result that is closer to the specifications provided by the manufacturer of your CPU?

- The busy wait method produced a result closer to the specifications provided by the manufacturer of my CPU. The busy wait method has a larger standard deviation than the specifications for my CPU but it is closer then the sleep method.

EVAL PROBLEM 2

a)
After running the server and connecting to the client, I calculated the time window by using the receipt timestamp of the first request and the completion timestamp of the last request which yielded these numbers:

Receipt Timestamp: 1695075549.377295
Completion Timestamp: 1695075600.941988

In finding the difference between the two I found that the time window is 51.5646929741.

To calculate the average throughput I took the total amount of requests run and divided it by the time window.

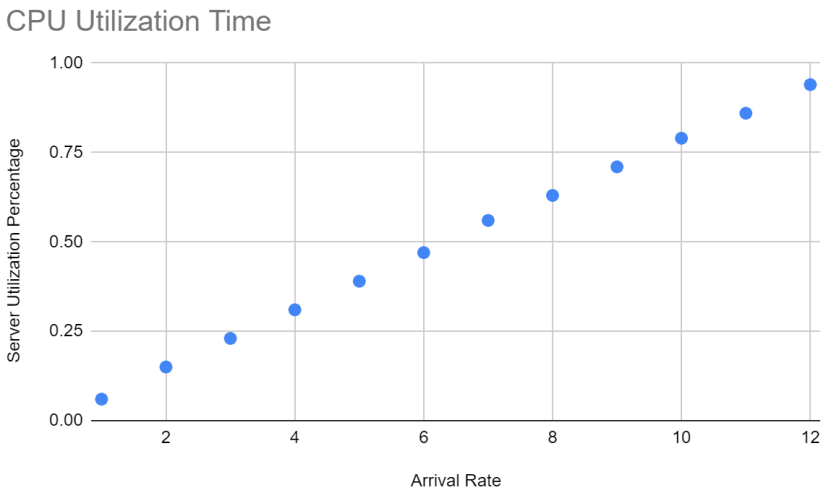
500/51.5646929741= 9.69655729844 requests per second

b)
In order to calculate the total utilization time, I summed the request timestamp of each of the 500 requests. To do this, I used R studio and imported the data. After using R, I found that the total time was 40.6641 between all the requests.

To calculate the total utilization time, I divided the total time and the time window.
40.6641/51.5646929741 = 0.78860355127 = 78.860355127%

c)
While working in the build directory and calling the given command,
I got that the percent of CPU of this job was 79% which matches with my calculated utilization time of 78.860355127%.

d) There is a trend that as the arrival rate increases so does the server utilization percentage. According to the data the increase occurs at almost a linear rate.



e) Using a Python script and outputting the data into the server-output.txt file, I calculated the average difference, maximum difference, minimum difference and the standard deviation of differences.

Average Difference: 1695142531.4651315
Maximum Difference: 1695142532.576828
Minimum Difference: 1695142531.129072
Standard Deviation of Differences: 0.3134776309260052

f) In using the same Python script from part E for each of the 12 arrival rates, I recorded the data of each of the CPU Utilization percentages and then calculated the average response time for each. The data I found is mapped in the graph below. The overall trend is that there is an exponential increase relationship between the server utilization percentage and the average response time.