A4 Report

Author: Kareem El Assad

@March 19, 2022

101107739

A4 Report Database

Number of Observations (n)	Elapsed Time (Instant) in Milliseconds (M1)	Elapsed Time in Nanoseconds (tmxb.getThreadCpuTime())	Elapsed Time in Milliseconds (M2) (tmxb.getThreadCpuTime())	Variance σ2 (M1)	Variance σ2 (M2)
1	62	62935000	62	19.36	18.7489
2	64	64356300	64	5.76	5.4289
3	67	67665300	67	0.36	0.4489
4	69	68838600	68	6.76	2.7889
5	68	67908500	67	2.56	0.4489
6	68	68120600	68	2.56	2.7889
7	63	64299000	64	11.56	5.4289
8	67	66737200	66	0.36	0.1089
9	63	64131500	64	11.56	5.4289
10	68	68092600	68	2.56	2.7889
11	68	69619500	69	2.56	7.1289
12	67	66325500	66	0.36	0.1089
13	69	68903400	68	6.76	2.7889
14	65	66416200	66	1.96	0.1089
15	68	68503500	68	2.56	2.7889

Calculations

Calculations for Part 3

meanM1 = sum(Elapsed Time (Instant) in Milliseconds (M1)) / 15

= 996 / 15 = 66.4

meanM2 = sum(Elapsed Time in Milliseconds (M2)(tmxb.getThreadCpuTime())) / 15

= 995 / 15 = 66.33

totalVarianceM1 = sum(Variance σ2 (M1)) = 77.6

totalVarianceM2 = sum(Variance σ2 (M2)) = 57.4

standardDeviationM1 = sqrt(totalVarianceM1) = 8.809

standardDeviationM2 = sqrt(totalVarianceM2) = 7.576

We want to find the “Z” value for a confidence interval of 95%.

From the site provided by the professor, Z = 1.960

Then, we can apply the following formula to find the confidence interval:

Confidence Interval = $mean \pm Z(\frac{std.div}{\sqrt{n}})$

So,

confidenceIntervalM1 = $66.4 \pm (1.960)(\frac{8.809}{\sqrt{15}})$ = 66.4 ± 4.458

confidenceIntervalM2 = $66.33 \pm (1.960)(\frac{7.576}{\sqrt{15}})$ = 66.33 ± 3.834

Conclusion

While the results were extremely similar, I believe that the 2nd method (Not using Instant) yields better results and is therefore preferred. The standard deviation for M1 was higher than that found for M2 (8.809 > 7.576). This indicates that the M1 values are more widely spread than those of M2 and are therefore, less reliable. This is also reflected in the confidence interval calculations above.

Since we have the mean time for the program to send 1000 packets in both directions, we can calculate the time required to send 1 packet by dividing by 1000. This would mean that M1 would estimate $0.0664ms$ while M2 would estimate $0.0663ms$. I don’t think these time scales are realistic as it does not accurately factor in server start-up time. I believe that it would take at least between $5ms - 7ms$.