MPI Sample Analysis

Operating Systems Lab Exercise 3

Skyelar Craver & Steven Pitts

February 21, 2019

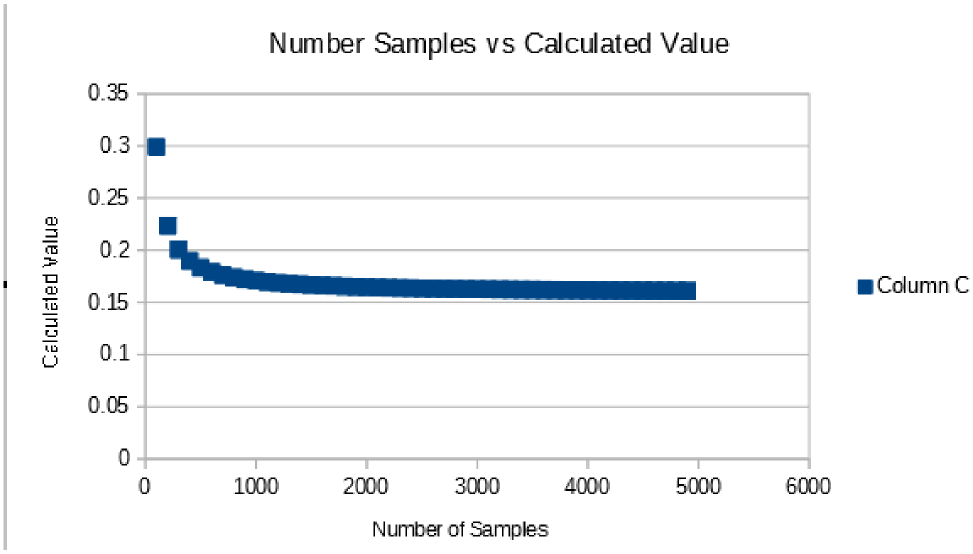
By submitting this assessment, we hereby declare that we have neither given nor received any unauthorized help or used any unauthorized materials during this assessment, and that we have adhered to any additional policies regarding Academic Honesty set by Wentworth Institute of Technology.

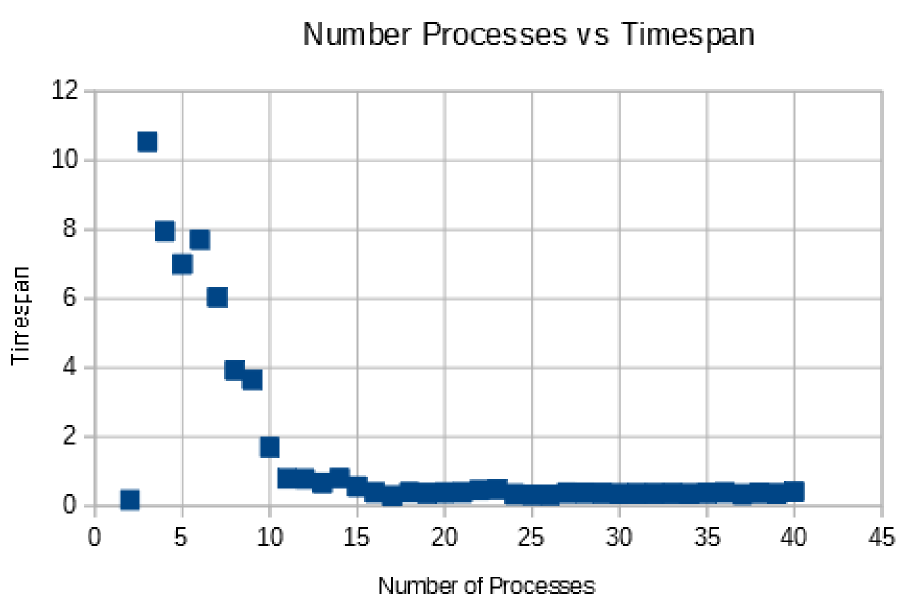
Part 1:

The goal of this experiment is to determine what optimal number of samples should be used when using the Monte Carlo Integration on a sequence of values. It will be assumed that the function f does not drastically change the answer, and it will be assumed that the Turing system in use is a reasonably accurate model for a generic 40-core host. The number of samples will be increased until a value is clearly being approached, and the lowest value of N that gives a decent representation of the true value will be chosen for future analysis.

Part 2:

The first necessary step is choosing a and b. For purposes of this experiment, it was decided that a will be 1 and b will be 100. The number of samples along with the value calculated from them is shown below:





Num Samples	Result
100	0.298888
200	0.223525
300	0.200784
400	0.189837
500	0.183402
600	0.179167
700	0.17617
800	0.173936
900	0.172208
1000	0.17083
1100	0.169707
1200	0.168774
1300	0.167986
1400	0.167312
1500	0.166728
1600	0.166218
1700	0.165769
1800	0.16537
1900	0.165014
2000	0.164693
2100	0.164404
2200	0.16414
2300	0.1639
2400	0.16368
2500	0.163478
2600	0.163291
2700	0.163118
2800	0.162958
2900	0.162809
3000	0.16267
3100	0.162539
3200	0.162418
3300	0.162303
3400	0.162195
3500	0.162094
3600	0.161998
3700	0.161907
3800	0.161821
3900	0.161739
4000	0.161662
4100	0.161588
4200	0.161518
4300	0.161451
4400	0.161388
4500	0.161327
4600	0.161268
4700	0.161213
4800	0.161159
4900	0.161108

Num Processors, Timespan

2	0.17591
3	10.54272
4	7.95493
5	7.00438
6	7.71559
7	6.04355
8	3.93449
9	3.65489
10	1.70714
11	0.7957
12	0.79105
13	0.66726
14	0.81301
15	0.56204
16	0.39852
17	0.29337
18	0.40296
19	0.37225
20	0.39189
21	0.39796
22	0.46283
23	0.48705
24	0.34382
25	0.32853
26	0.32371
27	0.38729
28	0.38605
29	0.37434
30	0.35425
31	0.35938
32	0.36197
33	0.36653
34	0.35435
35	0.3808
36	0.39508
37	0.34063
38	0.38347
39	0.35882
40	0.41667

Given the approximation of 0.16, a value of 2000 was chosen to be a reasonable representation of the number of samples, but a value of 100000 samples was used to be safe and to get a reasonable timing diagram. Using that, we can now analyze execution time for different numbers of processes. The relation between number of processes and execution time is found above.

Part 3:

Given the asymptote seen around a value of 31 it can be gathered that increasing the number of processes above that will do either nothing or be detrimental to the execution time. Therefore, since the lowest execution time was found to occur at 31 processes, it can be determined that 31 is the optimal number of processes for the given program. Since the execution time for a single process was 10.54272, we can calculate that the calculated speedup with 31 processes would result in execution time of $\frac{10.54272}{31} = 0.34$. However, the true execution time with 31 processes was 0.359, due to the fact that the master process must be synchronous, and the overhead associated with multiple processes. This overhead includes moving variables around between threads not on the same core (although this matters little in this program), and extra calculations/assignments that must be performed in order to set up the multiple processes. We are able to go above 40 processes since a processor core can run multiple processes in threads, but this is of little help to us, as it provides no improvement. Using batch processing, the program took 0.359 seconds to run, while without it took 10 seconds (over a minute counting wait time) to run. The results are very different, since batch processing allows for waiting for the computer to be ready with all cores needed, which can separate wait times from actual execution time.

Part 4:

Given the analysis of number of samples in part 2, the optimal number of samples was determined to be 100000, as it gives a close enough approximation of the integral without being so large that it would take a long period of time to process. Using the execution times mapped to the number of processes, the optimal processes count was determined to be 31, due to its lowest execution time.