CSC 428 Operating Systems

Professor. Scott Thede

Masayuki Nagai (Moon)

Project 3: Data Report

Estimated Pi values

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **2** | **5** | **10** | **20** |
| **100** | 3.200 | 3.152 | 3.116 | 3.144 |
| **1000** | 3.1100 | 3.1432 | 3.1456 | 3.1584 |
| **10000** | 3.1396 | 3.1372 | 3.1497 | 3.1416 |

Root squared error of each estimation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **2** | **5** | **10** | **20** |
| **100** | 0.058407 | 0.010407 | 0.025592 | 0.002407 |
| **1000** | 0.031592 | 0.001607 | 0.004007 | 0.016807 |
| **10000** | 0.001992 | 0.004392 | 0.008127 | 0.000027 |

In the two tables above, the top row above represents the number of threads to run and the left column represents the number of arrows to throw in each thread. To see how close each estimation is to the true pi, I also calculated the root squared error of each estimation. As I expected, the worst estimation of pi is when using 2 threads with 100 arrows whereas the best estimation of pi is when using 20 threads with 10000 arrows. In general, increasing the total number of arrows to throw (the number of threads \* the number of arrows to throw in each thread) improves the estimation. However, it’s important to note that increasing the total number of arrows to throw doesn’t always improve the estimation of pi as you can see the estimation when using 5 threads with 1000 arrows (5000 arrows to throw in total) is better than (10, 1000), (20, 1000), (2, 10000) and (5, 10000).