

SE292:AUG2014: Assignment 3: 60 Points

Due by email (simmhan@serc.iisc.in) by Tue Nov 18 2014, 7AM

PThreads & Concurrency using the Producer Consumer Problem

- (1) Run the program [ProdCons.c](#)¹ with and without CONCURRENCY enabled, for input number of items $n=1, 100, 10000, 100000, \text{ and } 1000000$. Discuss the correctness and time taken based on your observations? (10 points)
- (2) Modify the concurrent program from (1) so that the producer and consumer share a buffer of size “m” items ($m < n$), i.e., the producer can produce up to ‘m’ items that have not yet been consumed, before it waits for the buffer capacity to get freed up. Test this program for $m=10$ and $m=1000$ for $n=100, 10000, 100000$ and 1000000 . The values of m and n should be taken as a commandline input, e.g., `ProdConsBuffered 10 100000`
Report your results for different values of m and n, and discuss your observations. Submit the code as `ProdConsBuffered.c`. (20 points)
- (3) Modify the program from (2) such that the producer code now generates items that are each matrices of size 1024×1024 random floating point numbers. Modify the consumer such that it generates the square of this item (i.e., multiplies the matrix with itself) stored as a separate result. Run the program with 1 producer thread and 1, 2, 3, and 4 consumer threads, with $m=10$ and $n=10, 50$ and 100 . The number of consumer threads and values of m and n should be taken from commandline, e.g., `ProdConsMatMult 2 10 50`
Report the average time taken over 5 runs each, when the number consumer threads is 1, 2, 4 and 8. Plot a curve where the X Axis is the number of threads and Y axis is the speedup relative to 1 consumer thread. Discuss the scalability, speedup and efficiency of the program. Submit the code as `ProdConsMatMult.c` (30 points)



NOTE:

- i) Document your code. Make sure the code compiles on a Linux environment using gcc and standard pthread library, e.g.,
`gcc -pthread -o ProdConsBuffered ProdConsBuffered.c`
- ii) Use the file naming and the input parameter conventions listed in the question.
- iii) List the gcc version and machine specifications (e.g., OS, version, CPU architecture CPU processors/cores, core speed, memory available) when reporting your results.
- iv) Your submission should contain a single PDF document of the observations/report, and two C programs named `ProdConsBuffered.c` and `ProdConsMatMult.c`. All files you submit should contain your name and email.
- v) Email your submission to simmhan@serc.iisc.in by the deadline.

¹ <http://www.serc.iisc.ernet.in/~simmhan/SE292/assignments/ProdCons.c>