OpenMp-Approximating Pi

By: Rabindra Khadka

Overview:

The objective of this exercise was to write an OpenMP program to compute the value of pi by performing integration between 0 to 1 with 10^8 steps of a function f(x)=4/(1+x*x) .The code demonstrates the use of critical clause, atomic clause and reduction clause.

Outcome :

After turning the serial code into parallel with OpenMp using critical clause, atomic clause and reduction clause elapsed time for each run were recorded as shown below in the table.

No of threads	Serial time(s)	Critical time(s)	Atomic time(s)	Reduction time(s)
1	1.34	1.34	1.34	1.36
2	1.34	0.67	0.67	0.57
4	1.34	0.35	0.35	0.16
8	1.34	0.18	0.18	0.08
16	1.36	0.10	0.10	0.04
20	1.36	0.17	0.16	0.06

Table 1: Elapsed Run time with different OpenMp clauses

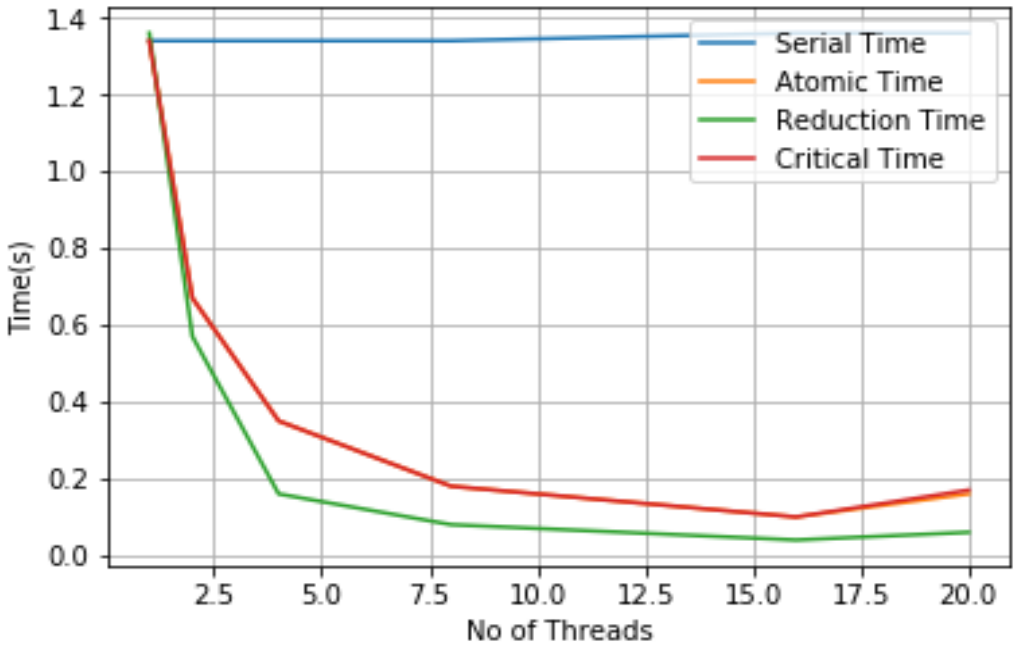

Fig1: No of threads vs elapsed time in seconds

As seen in figure 1 above, the elapsed time of the serial pi program with 10^8 steps ran in 1.34 sec and the performance did not scale up even when threads were added. The elapsed run time of the parallel program decreased significantly as the threads were increased .The parallel program performance with reduction clause scaled up efficiently in compare to the program ran with critical and atomic clauses. The atomic and critical clauses depicted similar level of performance.

Build Instructions:

The file named ***'OMP_Exercise_1.1.c'*** available in github repo can be executed with the help of the attached **makefile** and the result can be obtained by submitting the code in Ulysses's node with the following command line:

qsub -l nodes=1:ppn=20,walltime=3600 -q regular calculate_pi.sh