Lab Session 3

Submission deadline: Feb 20, 11:59pm

Please submit your lab results through CatCourses, including Makefile, source code and a short report (up to one page). You may find the OpenMP tutorial from Lawrence Livermore National Lab useful (https://computing.llnl.gov/tutorials/openMP/).

1.Parallelization of matrix vector multiplication

1.1 Create a program that computes a simple matrix vector multiplication b=Ax, Use OpenMP directives to make it run in parallel. Change the number of threads and measure performance.

1.2 Fix the number of threads (e.g., using 4 threads), and try static and dynamic scheduling strategies ("dynamic" and "guided"). Measure performance and explain your results.

2. Producer/consumer

Parallelize the "prod_cons.c" program (see lab3/prod_cons.c at CatCourses). This is a well-known pattern called the producer consumer pattern: One thread produces values that another thread consumes. This pattern is often used with a stream of produced values to implement "pipeline parallelism".

Use OpenMP directives to parallel prod_cons.c. Report the execution time.

Hint: the key to parallel this program with OpenMP is to employ synchronization between threads.

3. Linked list

Consider the "linked.c" program (see lab3/linked.c at CatCourses). This program traverses a linked list computing a sequence of Fibonacci numbers at each node.

3.1 Parallelize this program using OpenMP tasks.

Change the list size (i.e., N) and the number of threads, and report the execution times.