EG4387 – Parallel Programming/EG6370 Parallel Processing

Project

If a sequence of numbers follows a pattern of adding a fixed amount from one term to the next, it is referred to as an arithmetic sequence. The sum of its terms is called an arithmetic series. The following examples are all arithmetic series:

1) $1 + 2 + 3 + 4 + 5 + \ldots. + (n - 2) + (n - 1) + n = \sum_{k=1}^{n}(k)$
2) $3 + 10 + 17 + 24 + 31 + \ldots + (3 + 7(n-1)) = \sum_{k=1}^{n}(3 + 7(k - 1))$

In General

$S_n = a_1 + a_2 + a_3 + a_4 + \ldots + a_k + \ldots + a_n = \sum_{k=1}^{n} a_k$

$= a_1 + (a_1 + d) + (a_1 + 2d) + (a_1 + 3d) + (a_1 + 4d) + \ldots + (a_1 + (n - 1)d) = \sum_{k=1}^{n}(\mathrm{a1} + (\mathrm{k} - 1)\mathrm{d})$ (1)

$a_k = a_1 + (k - 1)d$

a_1 is the first element in the arithmetic sequence.

d is the constant difference between consecutive elements in the arithmetic sequence.

S_n is the sum of the all the elements in the arithmetic sequence.

$$s_n = \frac{1}{2}n(a_1 + a_n) = \frac{1}{2}n(2a_1 + d(n - 1)) \qquad (2)$$

1) Write a parallel program using MPI that would compute the arithmetic series given by equation (1) above. Your program should read the first term, the constant difference between consecutive elements, and the number of terms in the arithmetic sequence using the following command line:
   mpirun –n <number of processes> ProgramName < a_1> <d> <number of terms>
   Your program should use block allocation of terms of the arithmetic sequence. Each processor should compute a local sum of the terms allocated to it. The processors should then perform a sum reduction. A way of double-checking the result, processor 0 should also compute and print the value the arithmetic series using equation (2).
2) Rewrite the same program using OpenMP, use cyclic allocation of terms of the arithmetic sequence.

Due date: April 28, 2020