To design and implement the MPI Game of Life program.

Homework #4

By

Krishna Priya Kandru

CS 632 Parallel Computing

April 2, 2021

Gitlab: https://gitlab.cis.uab.edu/b0159342/hw4

**Problem Requirements**:

To design and implement a message-passing version of the "Game of Life" program using Message Passing Interface (MPI).

**Problem Process**:

The program takes the problem size, number of iterations as command line arguments. This is another process of parallelization. Open MP is a shared one whereas here we work on distributed.

command line arguments. After initializing we divide initial array into multiple processes using functions where each process pass its array to last row where again it is passed to top array Send and receive are from the processes at the end of array. Remapping of data is done on ghost cells using Scatter function. Open MP is an API which makes it easier to write shared memory program whereas MPI is API declarations. MPI runs on both shared and distributed memory architectures. Communication between nodes using MPI send receive is the difficult task in MPI programming. Communication between nodes is important for performace.

**Testing Plan:**

The program is tested for different number of processes and speedup, efficiency is calculated.

The testing is done by fixing no of iterations and board size to 5000. The number of processes is increased in steps. The program is tested for different number of processes in the scale to th power of 2. Speed up is calculated as time taken for serial execution to the parallel and efficiency as speedup is to number of processes.

**Test Cases**: (non-blocking version)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case number | Threads  (size is 5000X5000) constant | Time taken  DMC cluster  (iterations is 5000) | Speed Up  (Tserial/Tparallel) | Efficiency  (speedup/cores) |
| 1 | 1 | 1824.37 | 2.76 | 2.76 |
| 2 | 2 | 945.64 | 5.3 | 2.6 |
| 3 | 4 | 472.311 | 10.6 | 2.6 |
| 4 | 8 | 247.73 | 20.3 | 2.53 |
| 5 | 10 | 190.64 | 26.43 | 2.6 |
| 6 | 16 | 124.89 | 40.3 | 2.5 |
| 7 | 20 | 93.46 | 52.24 | 2.6 |

**Test Cases**: (blocking version)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case number | Threads  (size is 5000X5000) constant | Time taken  DMC cluster  (iterations is 5000) | Speed Up  (Tserial/Tparallel) | Efficiency  (speedup/cores) |
| 1 | 1 | 1930.91 | 2.6 | 2.6 |
| 2 | 2 | 954.17 | 5.2 | 2.6 |
| 3 | 4 | 491.04 | 10.26 | 2.56 |
| 4 | 8 | 251.25 | 20.05 | 2.53 |
| 5 | 10 | 202.54 | 26.43 | 2.50 |
| 6 | 16 | 127.02 | 39.6 | 2.47 |
| 7 | 20 | 104.29 | 48.3 | 2.4 |

**Analysis:**

Speedup = time for serial/ time for parallel

Efficiency = speedup/ Threads

Initially the program is divided into chunks and values. These are scattered into processes and checked if the scatter worked. The open MP performance is more advanced than MPI implementation. Since processes cannot share data is the main disadvantage associated with the message passing interface. Since open MP shares the data the speed up and execution time.

**Readme.md:**

# hw4

Game of life MPI

To compile: mpicc (blocking or nonblocking).c -o hw4

to run: mpiexec –n (no:of processes)