CMSC416: Introduction to Parallel Computing

Topic: MPI Performance Analysis, Modeling, and Tools

Date: 02/15/2024

1. Assignment 1 has been “officially” posted, has been up for a week technically
2. Assignment 0
    a. Best way is to create tarball in zaratan, then scp it over into your local computer then upload
    b. You should be comfortable with linux commands by now
3. For the scribe notes
    a. Use the template online in gradescope
4. Assignment 1
    a. Implement game of life in parallel
    b. Certain ruleset the 2D array has to follow
    c. No wraparound
        i. You can just put 0 for the “ghost” cells
    d. Board can be rectangular, not just a square
    e. You get a makefile, but you have to modify and adapt it to our program
    f. Batch script is similar to the same one as assignment 0
    g. Must first parse the input file and assign the alive cells, then divide the board up after
    h. The executable takes (Command Line Args)
        i. File name
        ii. Number of generations in this game of life
        iii. X limit
        iv. Y limit
    i. At least 4 processes but will be up to 16 processes
    j. 2 ways
        i. Have each process read the entire board
        ii. Have one process read the entire board and then distribute the portions accordingly
    k. Output
        i. Make a csv file
        ii. Each coordinate is its own line
        iii. Sort by x and y coordinates
        iv. Single unified csv file
        v. Best way is to get the entire output to process 0 and write it to csv
    l. Tested on both output as well as performance
        i. Print the timings to stdout
            1. Calculate min time, max time, avg time
        ii. Parallel version should be faster than sequential version
        iii. Check website for grading split for this assignment
    m. 2 versions of the code
        i. 1D decomposition with blocking send and recv

        ii. 1D decomposition with non-blocking Isend and Irecv
    n. You have to submit a report pdf as well
        i. 2 line plots of the performance on the 2 parallel versions
        ii. Check the assignment page for more information
5. Other MPI Commands
    a. MPI_Waitany
    b. MPI_Test
    c. MPI_Wait
        i. To link an operation with the completion event
        ii. For non-blocking related operations
    d. MPI_Waitall
        i. Waits for all mpi requests to finish
6. Collective Operations
    a. Don't worry about how they are implemented
    b. MPI_Barrier
        i. Blocks until all processors have reached this point
    c. MPI_Bcast
        i. For sending to all processors from communicator
    d. MPI_Reduce
        i. Reduce data for all processes to the root
        ii. Recv is only on root
        iii. Sendbuf is on all processes, even root
    e. MPI_Allreduce
        i. For when you want to send the result back to all processes
        ii. For sending output to all processes
    f. MPI_Scatter
        i. Send from root to all processes
        ii. Different data to diff processes
        iii. Scatterv is for when you want to send different amounts of data to different processes
    g. MPI_Gather
        i. Gathers data from all processes to root
        ii. Different data
        iii. Gatherv is for gathering different amounts of data for different processes
    h. MPI_Scan
7. Other MPI Calls
    a. MPI_Wtime
        i. Returns elapsed time
        ii. Just use this instead of clock_gettime()
8. Calculating π
    a. Best way is to change process with iterations
        i. Rank 0 does the first iteration, Rank 1 does the second iteration, etc…
    b. Use MPI_Bcast
    c. Use MPI_Reduce

    d. Go for n/p from index
    e. “Embarrassingly” parallel
    f. Bcast is for sending the value to all processes
    g. Everyone is on the same page for amount of iterations
    h. Reduce is for gathering partial sums and then computing the final answer.
        i. Ensure the semantics do not change when writing in parallel
9. MPI runtime will assign ranks for each process