CSCI596 Assignment 4—Parallel Molecular Dynamics

Due: October 1 (Fri), 2021

The purpose of this assignment is to gain hands-on experience in practical use of message passing interface (MPI) in real-world applications, thereby consolidating your understanding of asynchronous message passing and communicators. In addition, you will get familiar with the message-passing scheme used in common spatial-decomposition applications, using the parallel molecular dynamics (MD) program, `pmd.c`, as an example.

(Part I—Asynchronous Messages)

Modify `pmd.c` such that, for each message exchange, it first calls `MPI_Irecv`, then `MPI_Send`, and finally `MPI_Wait`. The asynchronous messages make the deadlock-avoidance scheme unnecessary, and thus there is no need to use different orders of send and receive calls for even- and odd-parity processes. In addition to just `MPI_Send`, insert other computations that do not depend on the received messages between `MPI_Irecv` and `MPI_Wait`.

- ***Submit the modified source code, with your modifications clearly marked***.
- Run both the original `pmd.c` and the modified program on 16 cores (requesting 4 nodes with 4 cores per node in your Slurm script), and compare the execution time for `InitUcell` = {3,3,3}, `StepLimit` = 1000, and `StepAvg` = 1001 in `pmd.in` (keep all the other parameter values as they are as downloaded from the course home page) and `vproc` = {2,2,4} (*i.e.*, `nproc` = 16) in `pmd.h`. Which program runs faster? Repeat the comparison three times and report the average runtime of both programs. ***Submit the timing data***.

(Part II—Communicators)

Following the lecture note on "*In situ* analysis of molecular dynamics simulation data using communicators," modify `pmd.c` such that as many number of processes as that for MD simulations is spawned to calculate the probability density function (PDF) for the atomic velocity.

- ***Submit the modified source code (name it pmd_split.c), with your modifications clearly marked***.
- Run the modified program on 16 cores (requesting 2 nodes with 8 cores per node in your Slurm script), with which 8 cores perform MD simulation and the other 8 cores calculate PDF. In `pmd.h`, choose `vproc[3] = {2,2,2}` and `nproc = 8`. Also, specify `InitUcell` = {5,5,5}, `StepLimit` = 30, and `StepAvg` = 10 in `pmd.in`. ***Submit the plot of calculated PDFs at time steps 10, 20, and 30***.