CSCI596 Assignment 5—Hybrid MPI+OpenMP MD—Answer

1. Major changes in the hybrid MPI+OpenMP program, `hmd.c`, from the MPI program, `pmd.c`.

In the header file, `hmd.h`

```
#include "omp.h"
...
int vthrd[3] = {2,2,1}, nthrd = 4;
int thbk[3];
```

In `hmd.c`

```
int main() {
  ...
  omp_set_num_threads(nthrd);
  ...
}

void init_params() {
  ...
  for (a=0; a<3; a++) {
    lc[a] = al[a]/RCUT;
    thbk[a] = lc[a]/vthrd[a];
    lc[a] = thbk[a]*vthrd[a];
    rc[a] = al[a]/lc[a];
  }
  ...
}

void compute_accel() {
  int i,j,a,lc2[3],lcyz2,lcxyz2,mc[3],c,mc1[3],c1;
  double lpe_td[nthrd];
  ...
  for (i=0; i<nthrd; i++) lpe_td[i] = 0.0;
  ...
#pragma omp parallel private(mc,c,mc1,c1,i,j,a)
  {
    double dr[3],rr,ri2,ri6,r1,fcVal,f,vVal;
    int std,vtd[3],mofst[3];
    std = omp_get_thread_num();
    vtd[0] = std/(vthrd[1]*vthrd[2]);
    vtd[1] = (std/vthrd[2])%vthrd[1];
    vtd[2] = std%vthrd[2];
    for (a=0; a<3; a++)
      mofst[a] = vtd[a]*thbk[a];
    for (mc[0]=mofst[0]+1; mc[0]<=mofst[0]+thbk[0]; (mc[0])++)
    for (mc[1]=mofst[1]+1; mc[1]<=mofst[1]+thbk[1]; (mc[1])++)
    for (mc[2]=mofst[2]+1; mc[2]<=mofst[2]+thbk[2]; (mc[2])++) {
      ...
            if (rr<rrCut) {
              ...
              lpe_td[std] += 0.5*vVal;
              for (a=0; a<3; a++) {
                f = fcVal*dr[a];
                ra[i][a] += f;
              }
            }
      ...
    } /* Endfor central cell, c */
  } /* End OpenMP parallel block */
  for (i=0; i<nthrd; i++)
    lpe += lpe_td[i];
```

2. Output from `hmd.c` (2 MPI processes each with 4 OpenMP threads on 8 processors), along with the corresponding Slurm script file, is given below.

(Output from `hmd.c`)
```
al = 4.103942e+01 4.103942e+01 2.051971e+01
lc   = 16 16 8
rc   = 2.564964e+00 2.564964e+00 2.564964e+00
thbk = 8 8 8
nglob = 55296
 0.050000  0.877345 -5.137153 -3.821136
 0.100000  0.462056 -4.513097 -3.820013
 0.150000  0.510836 -4.587287 -3.821033
 0.200000  0.527457 -4.611958 -3.820772
 0.250000  0.518668 -4.598798 -3.820796
 0.300000  0.529023 -4.614343 -3.820808
 0.350000  0.532890 -4.620133 -3.820798
 0.400000  0.536070 -4.624899 -3.820794
 0.450000  0.539725 -4.630387 -3.820799
 0.500000  0.538481 -4.628514 -3.820792
CPU & COMT = 8.546544e+00 1.181345e-01
```

3. Strong-scaling parallel efficiency for `hmd.c` with the number of threads 1, 2, 4, and 8 on one compute node at CARC.
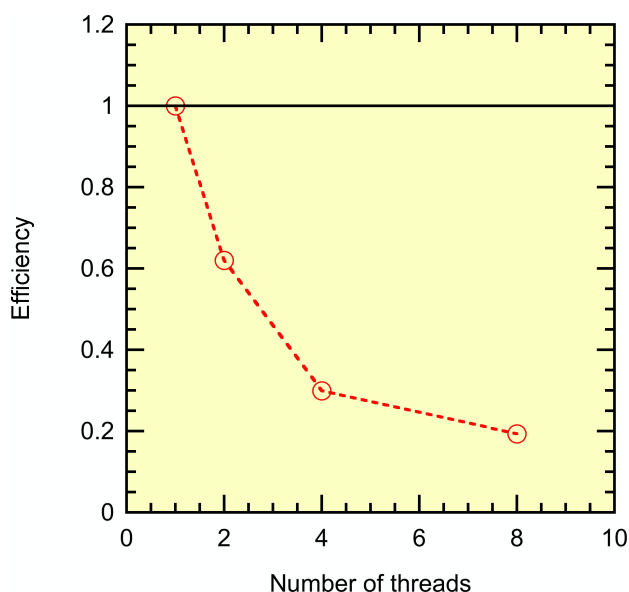
For the performance degradation of this program due to "false sharing", please see the lecture on "hybrid MPI+OpenMP parallel MD".