High Performance Computing

Term 4 2018/2019

Lecture 5

Hybrid MPI+OpenMP

```
#include <mpi.h>
#include <omp.h>
```

Hybrid MPI+OpenMP

```
#include <mpi.h>
#include <omp.h>

  .....
MPI_Init_thread(&argc, &argv, MPI_THREAD_MULTIPLE, &required);
  .....
MPI_Comm_rank(MPI_COMM_WORLD, &prank);
MPI_Comm_size(MPI_COMM_WORLD, &psize);
  .....
#pragma omp parallel
    {
        int tid = omp_get_thread_num();
        printf("Hello from thread[%d] from process[%d]\n", tid, prank);
    }
```

Hybrid MPI+OpenMP

http://tiny.cc/u8xy4y

MPI in Python

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

if rank == 0:
    data = {'a': 7, 'b': 3.14}
    comm.send(data, dest=1, tag=11)
elif rank == 1:
    data = comm.recv(source=0, tag=11)
```

MPI in Python

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

if rank == 0:
    data = {'a': 7, 'b': 3.14}
    req = comm.isend(data, dest=1, tag=11)
    req.wait()
elif rank == 1:
    req = comm.irecv(source=0, tag=11)
    data = req.wait()
```

MPI in Python (numpy arrays)

```
from mpi4py import MPI
import numpy

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

# passing MPI datatypes explicitly
if rank == 0:
    data = numpy.arange(1000, dtype='i')
    comm.Send([data, MPI.INT], dest=1, tag=77)
elif rank == 1:
    data = numpy.empty(1000, dtype='i')
    comm.Recv([data, MPI.INT], source=0, tag=77)

# automatic MPI datatype discovery
if rank == 0:
    data = numpy.arange(100, dtype=numpy.float64)
    comm.Send(data, dest=1, tag=13)
elif rank == 1:
    data = numpy.empty(100, dtype=numpy.float64)
    comm.Recv(data, source=0, tag=13)
```

MPI in Python

Python object

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

if rank == 0:
    data = {'key1' : [7, 2.72, 2+3j],
            'key2' : ( 'abc', 'xyz')}
else:
    data = None
data = comm.bcast(data, root=0)
```

Numpy array

```
from mpi4py import MPI
import numpy as np

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

if rank == 0:
    data = np.arange(100, dtype='i')
else:
    data = np.empty(100, dtype='i')
comm.Bcast(data, root=0)
for i in range(100):
    assert data[i] == i
```

Tasks

- Parallelize Hadamard product of two vectors (http://tiny.cc/4o0y4y) using MPI+OpenMP
- Write A*X+Y (axpy) where A-scalar, X and Y are vectors using mpi4py
- Continue with Schelling model