Parallel Computing

Hands-on with MPI Programming and Spartan

Farzad Khodadadi
PhD Candidate @ CLOUDS Lab
The University of Melbourne
farzad.khodadadi@unimelb.edu.au

Outline

- MPI programming basics and common guidelines
- MPI4Py and MPJ Express demo
- Accessing Spartan, submitting jobs, and monitoring the results

MPI Programming Basics

Many parallel programs can be written using just these six functions:

```
MPI_INIT
MPI_FINALIZE
MPI_COMM_SIZE
MPI_COMM_RANK
MPI_SEND
MPI_RECV
```

- MPI_SEND and MPI_RECV functions can be substituted with collective operations such as MPI_BCAST and MPI_REDUCE

Collective Operations in MPI

- ★**`MPI_BCAST`** distributes data from one process (the root) to all others in a communicator.

- ★**`MPI_REDUCE`** combines data from all processes in communicator and returns it to one process.

- ★In many numerical algorithms, **`SEND/RECEIVE`** can be replaced by **`BCAST/REDUCE`**, improving both simplicity and efficiency.

MPI4Py Sample Programs

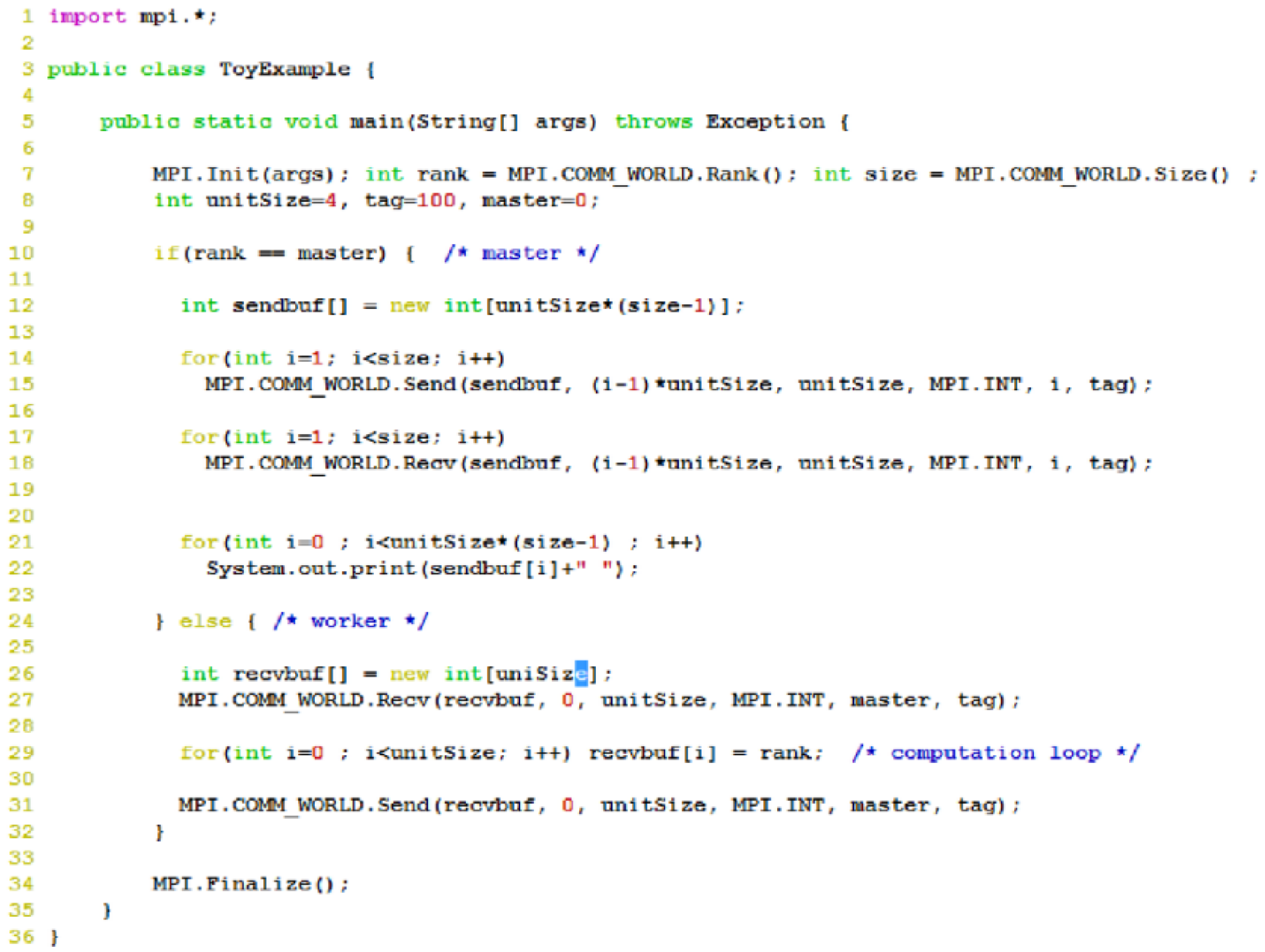
```
from mpi4py import MPI
import sys

size = MPI.COMM_WORLD.Get_size()
rank = MPI.COMM_WORLD.Get_rank()
print(“Helloworld! I am process %d of %d.\n” % (rank,
size))
```

```python
import numpy as np
from mpi4py import MPI

from parutils import pprint

comm = MPI.COMM_WORLD

pprint("-"*78)
pprint(" Running on %d cores" % comm.size)
pprint("-"*78)

comm.Barrier()

# Prepare a vector of N=5 elements to be broadcasted...
N = 5
if comm.rank == 0:
    A = np.arange(N, dtype=np.float64)    # rank 0 has proper data
else:
    A = np.empty(N, dtype=np.float64)     # all other just an empty array

# Broadcast A from rank 0 to everybody
comm.Bcast( [A, MPI.DOUBLE] )

# Everybody should now have the same...
print "[%02d] %s" % (comm.rank, A)
```

source: https://github.com/jbornschein/mpi4py-examples/

```python
import numpy as np
from mpi4py import MPI

from parutils import pprint

comm = MPI.COMM_WORLD

pprint("-"*78)
pprint(" Running on %d cores" % comm.size)
pprint("-"*78)

my_N = 4
N = my_N * comm.size

if comm.rank == 0:
    A = np.arange(N, dtype=np.float64)
else:
    A = np.empty(N, dtype=np.float64)

my_A = np.empty(my_N, dtype=np.float64)

# Scatter data into my_A arrays
comm.Scatter( [A, MPI.DOUBLE], [my_A, MPI.DOUBLE] )

pprint("After Scatter:")
for r in xrange(comm.size):
    if comm.rank == r:
        print "[%d] %s" % (comm.rank, my_A)
    comm.Barrier()

# Everybody is multiplying by 2
my_A *= 2

# Allgather data into A again
comm.Allgather( [my_A, MPI.DOUBLE], [A, MPI.DOUBLE] )

pprint("After Allgather:")
for r in xrange(comm.size):
    if comm.rank == r:
        print "[%d] %s" % (comm.rank, A)
    comm.Barrier()
```

MPI Programming using MPJ Express



```java
import mpi.*;

public class ToyExample {

    public static void main(String[] args) throws Exception {

        MPI.Init(args); int rank = MPI.COMM_WORLD.Rank(); int size = MPI.COMM_WORLD.Size() ;
        int unitSize=4, tag=100, master=0;

        if(rank == master) {  /* master */

          int sendbuf[] = new int[unitSize*(size-1)];

          for(int i=1; i<size; i++)
            MPI.COMM_WORLD.Send(sendbuf, (i-1)*unitSize, unitSize, MPI.INT, i, tag);

          for(int i=1; i<size; i++)
            MPI.COMM_WORLD.Recv(sendbuf, (i-1)*unitSize, unitSize, MPI.INT, i, tag);


          for(int i=0 ; i<unitSize*(size-1) ; i++)
            System.out.print(sendbuf[i]+" ");

        } else { /* worker */

          int recvbuf[] = new int[uniSize];
          MPI.COMM_WORLD.Recv(recvbuf, 0, unitSize, MPI.INT, master, tag);

          for(int i=0 ; i<unitSize; i++) recvbuf[i] = rank;  /* computation loop */

          MPI.COMM_WORLD.Send(recvbuf, 0, unitSize, MPI.INT, master, tag);
        }

        MPI.Finalize();
    }
}
```

MPI4Py and MPJ Expres Demo

Parallel Programming using Spartan

- Login to Spartan
  - yourusername@spartan2.hpc.unimelb.edu.au
- Upload your data (for your assignment, the data has been already uploaded and you only need to create a symbolic link to it in your home directory)
- Write a script to automate execution of your tasks
- Use SLURM’s commands to submit your script, monitor your job’s execution, cancel it, and much more.

Spartan Demo

More on SLURM and it’s commands?

- https://www.vlsci.org.au/documentation/managing-x86-jobs-slurm

- https://rc.fas.harvard.edu/resources/documentation/convenient-slurm-commands/

thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
tapadh leat
merci
obrigado
bedankt
dziękuję
go raibh maith agat
sukriya
kop khun krap
terima kasih
mochchakkeram
dakujem
arigatô
grazie
takk
taiku
sagolun
mauruuru
hvala
спасибо
Баярлалаа
рахмат
mersi
kia ora
barka
welalin
tack
spas
dank je
misaotra
matondo
paldies
grazzi
mahalo
хвала
asante
manana
obrigada
murakoze
tenki
chokrane
mamnun
дякую
djiere dieuf
tau
faafetai lava
vinaka
спасибі
blagodaram
kiitos
dankie
nandri
nanni
dhanyavad
bayarlalaa
gràcie
köszönöm
dankon
aciū
akun
enkosi
sobodi
dēkuji
chnorakaloutioun
gratias ago
gràcies
sulpāy
ありがとう
tanemirt
rahmet
mèsi
didi madloba
najis tuke
kam sah hamnida
rahmat
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
diolch
dhanyavadagalu
shukriya
mercé
мерси
trugarez