Parallel Computing

Hands-on with MPI Programming and Spartan

Farzad Khodadadi
PhD Candidate @ Melbourne eResearch Group (MeG)
The University of Melbourne
farzad.khodadadi@unimelb.edu.au

Outline

- MPI programming basics and common guidelines
- MPI4Py 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

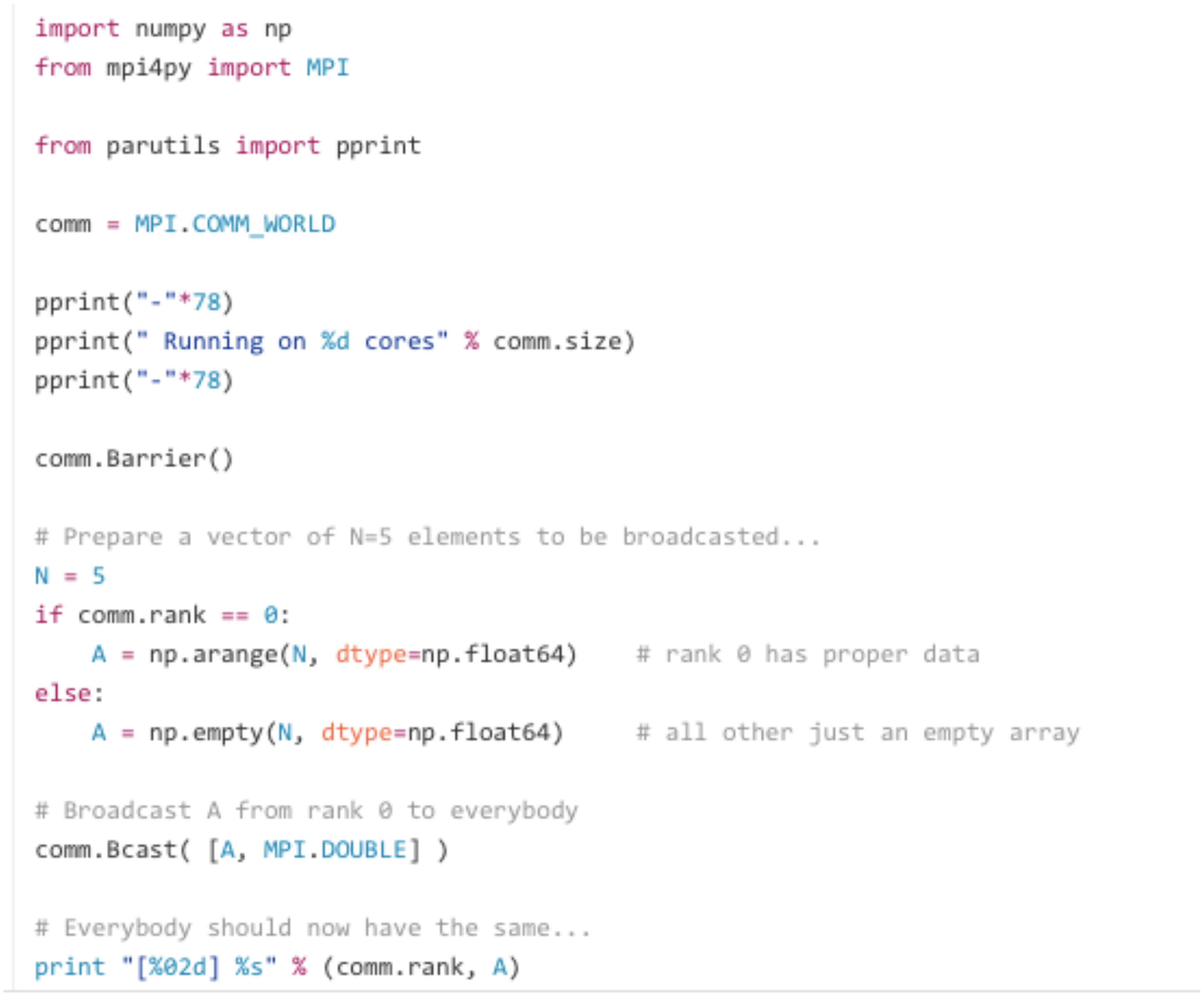
Collective Operations in MPI

- ★**`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()
```

MPI4Py Demo

Parallel Programming using Spartan

- Login to Spartan
    - yourusername@spartan.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://dashboard.hpc.unimelb.edu.au/getting_started/
- https://rc.fas.harvard.edu/resources/documentation/convenient-slurm-commands/

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