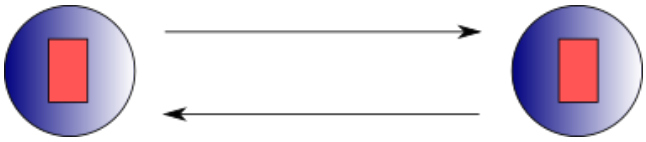
MPI tutorials with solution

Sarith P Sathian

April 5, 2019

1 MPI point to point communication

Write a python script to send a *numpy array [0,1,2,3,4,5,6,7,8]* from process with **rank 0** to process with **rank 1**.

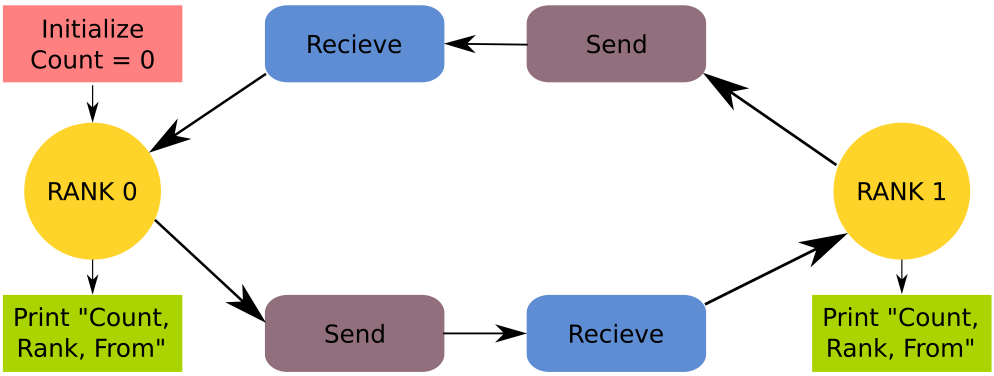


Solution:

```
from mpi4py import MPI
import numpy as np

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

if rank == 0:
  data = np.arange(9, dtype='int')
  comm.Send([data, MPI.INT], dest = 1, tag = 11)
  print('Data sent from',rank)
if rank==1:
  data = np.zeros(9, dtype='int')
  comm.Recv([data, MPI.INT], source = 0, tag = 11)
  print(rank,'Recieved',data)
```

2 P2P example (ping pong game)

Write a python script to send a *numpy array "count* [0]*"* in a loop starting from process with **rank 0** through processes with **ranks 1,2,..n** and finally back to **rank 0** from **rank n**. The value of *count* is incremented in every sent and the loop terminates when the value exceeds *max-count*.

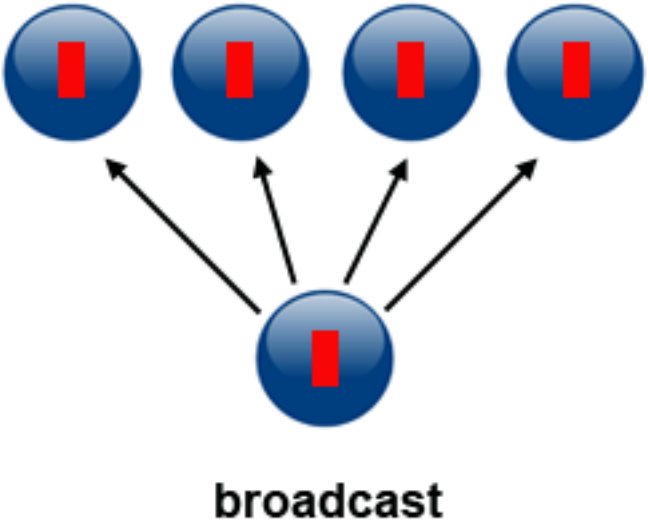


Solution:

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

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

max_count = 5
partner = (rank+1)%size
count = np.array([0], dtype = 'i')

while count[0] < max_count:
  if rank == (count[0] % 2):
    count[0] = count[0] + 1
    comm.Send([count, MPI.INT], dest = partner, tag = 11)
    print('Count', count[0],'send from',rank,'to',partner)
  else:
    comm.Recv([count, MPI.INT], source = partner, tag = 11)
    print('Count',count[0],'recieved at',rank,'from',partner)
```

3 MPI Broadcast

Write a python script to send an *numpy array (size 100)* from root process to all other processes.

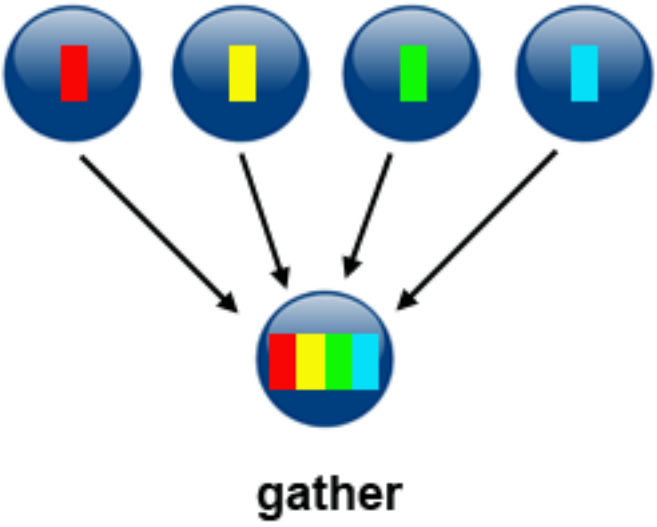


Solution:

```
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)
print(data, rank)
```

4 MPI Gather

Write a python script to gather *numpy arrays (size 10)* from all processors to **rank 0**.

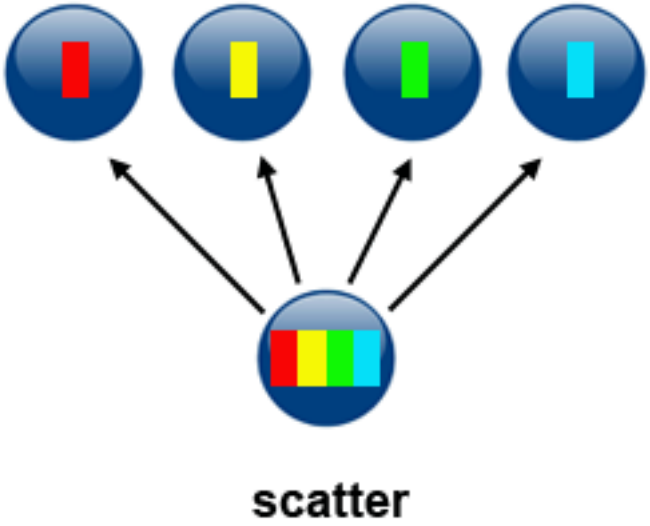


Solution:

```
from mpi4py import MPI
import numpy as np

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

sendbuf = np.zeros(10, dtype='i') + rank
recvbuf = None
if rank == 0:
    recvbuf = np.empty([size, 10], dtype='i')
comm.Gather(sendbuf, recvbuf, root=0)
if rank == 0:
    print(recvbuf)
```

5 MPI Scatter

Write a pyhton script to scatter a piece of *numpy array (size 100)* to all processes from **rank 0**.

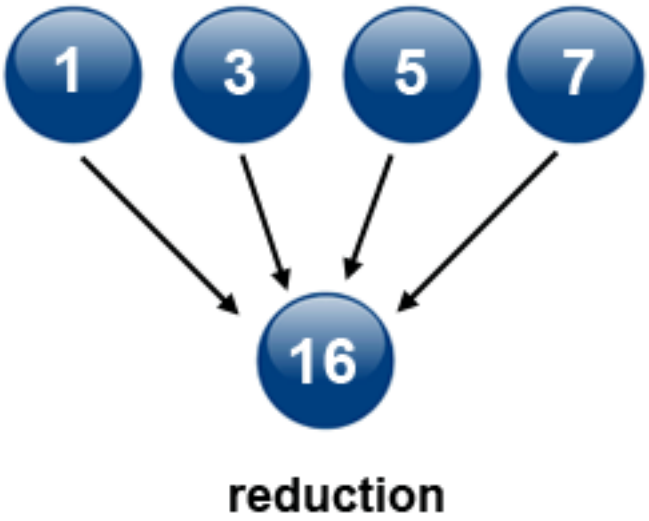


Solution:

```
from mpi4py import MPI
import numpy as np

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

sendbuf = None
if rank == 0:
    sendbuf = np.empty([size, 100], dtype='i')
    sendbuf.T[:,:] = range(size)
recvbuf = np.empty(100, dtype='i')
comm.Scatter(sendbuf, recvbuf, root=0)
print(recvbuf, rank)
```

6 MPI Reduce

Write a python script to *get the sum* of vlaues in all the processes on **rank 0**.



Solution:

```
from mpi4py import MPI
import numpy as np

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

sendbuf = np.zeros(10, dtype='i') + rank
recvbuf = None
if rank == 0:
    recvbuf = np.empty([10], dtype='i')
comm.Reduce(sendbuf, recvbuf, op=MPI.SUM, root=0)
if rank == 0:
    print(recvbuf)
```