HW03 - Integer Sort
CS5500

Chelsi Gupta

September 20, 2018

1 Steps for Implementation

- I have implemented this code in python3 as I am more comfortable with it and also chunking of data into segments is much easier in python.
- First I declared a list to be sorted with random integers in it.
- Calculated the number of slave processes which is "total processes(size) - 1".
- The data segment which will be sent to each slave is "len(listToSort)/numSlaves".
- The process with rank 0 is the master and will send chunks of data to each slave for sorting.
- If the length of list to be sorted is not evenly divisible by number of slaves then the last slave process gets 1 extra element then the other slaves which have equal sized chunks.
- Each slave process sorts its chunk of data and sends it back to the master process.
- As soon as master receives a chunk, it appends it to a new list(called sortedList) and sorts the combined list.
- The master then displays the final sorted list after receiving, merging and sorting the last chunk.
- I compiled and ran the program using "mpirun -np 3 python master_slave.py".

2 Code

```
from mpi4py import MPI
comm = MPI.COMM_WORLD   # Defines the default communicator

listToSort = [9,5,3,8,2,6,1,7,4,0,11,34,23]
```

```
sortedList = []
size = comm.Get_size()  # Stores the number of processes in size.
rank = comm.Get_rank()  # Stores the rank (pid) of the current process

numSlaves = size-1
chunk_size = len(listToSort)//numSlaves
l = 0
r = chunk_size

if rank == 0: # I am the master
    print ("The list to be sorted is: ",listToSort,'\n')
    for i in range(1,size,1):
        if len(listToSort) % numSlaves!= 0 and i == size-1:
            lstChunk =  listToSort[l:]
            comm.send(lstChunk, dest=i, tag=0)
            print ("The master is directing slave process ",i," to sort",lstChunk,'\n')
        else:
            lstChunk =  listToSort[l:r]
            print ("The master is directing slave process ",i," to sort",lstChunk,'\n')
            comm.send(lstChunk, dest=i, tag=0)
            l += chunk_size
            r += chunk_size

    while numSlaves>0:
        lstChunk = comm.recv(source=MPI.ANY_SOURCE, tag=0)
        for i in lstChunk:
            sortedList.append(i)
        sortedList.sort()
        numSlaves -= 1

    print ("The sorted list is ",sortedList,'\n')

else: # I am a slave
    lstChunk = comm.recv(source=0, tag=0)
    print ("Slave process %s is sorting its chunk" %rank,'\n')
    lstChunk.sort()
    comm.send(lstChunk, dest=0, tag=0)
```

3 Output

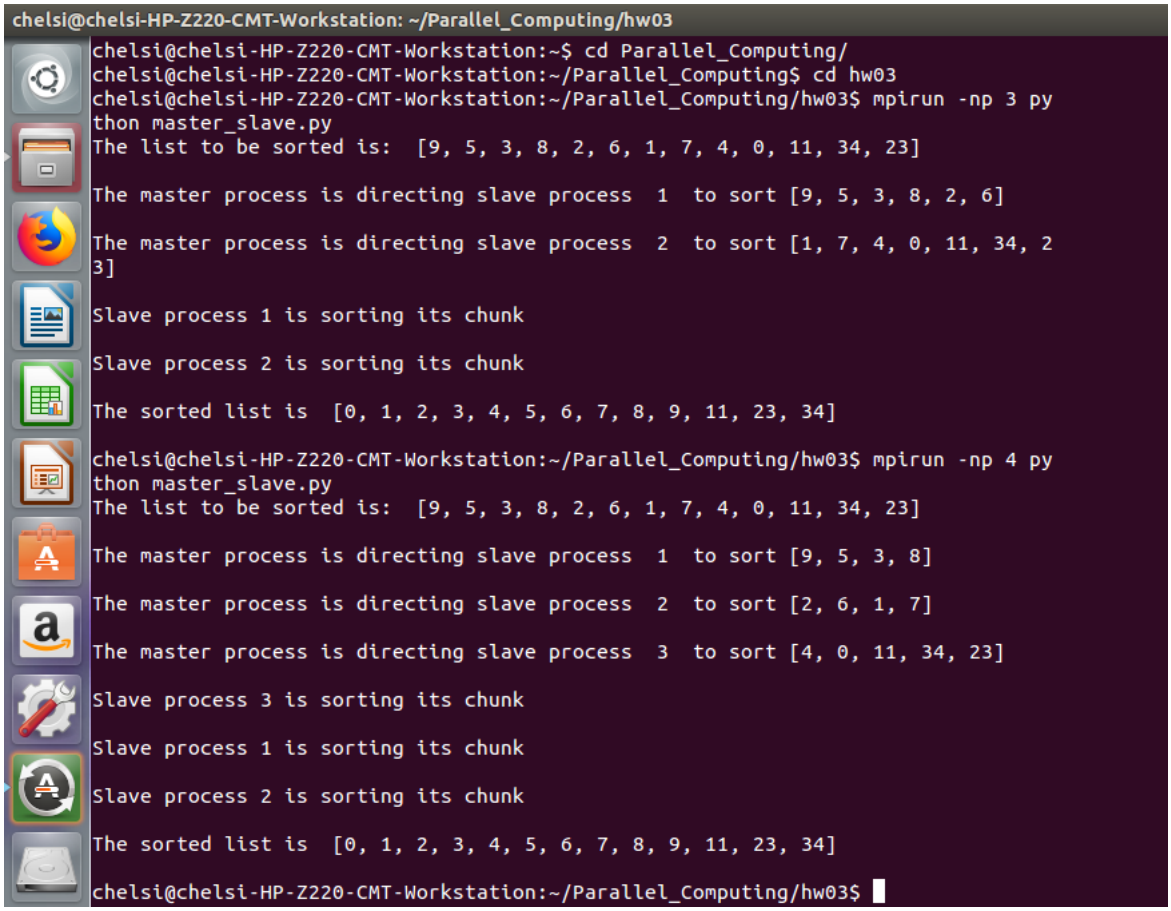

Figure 1: My Output