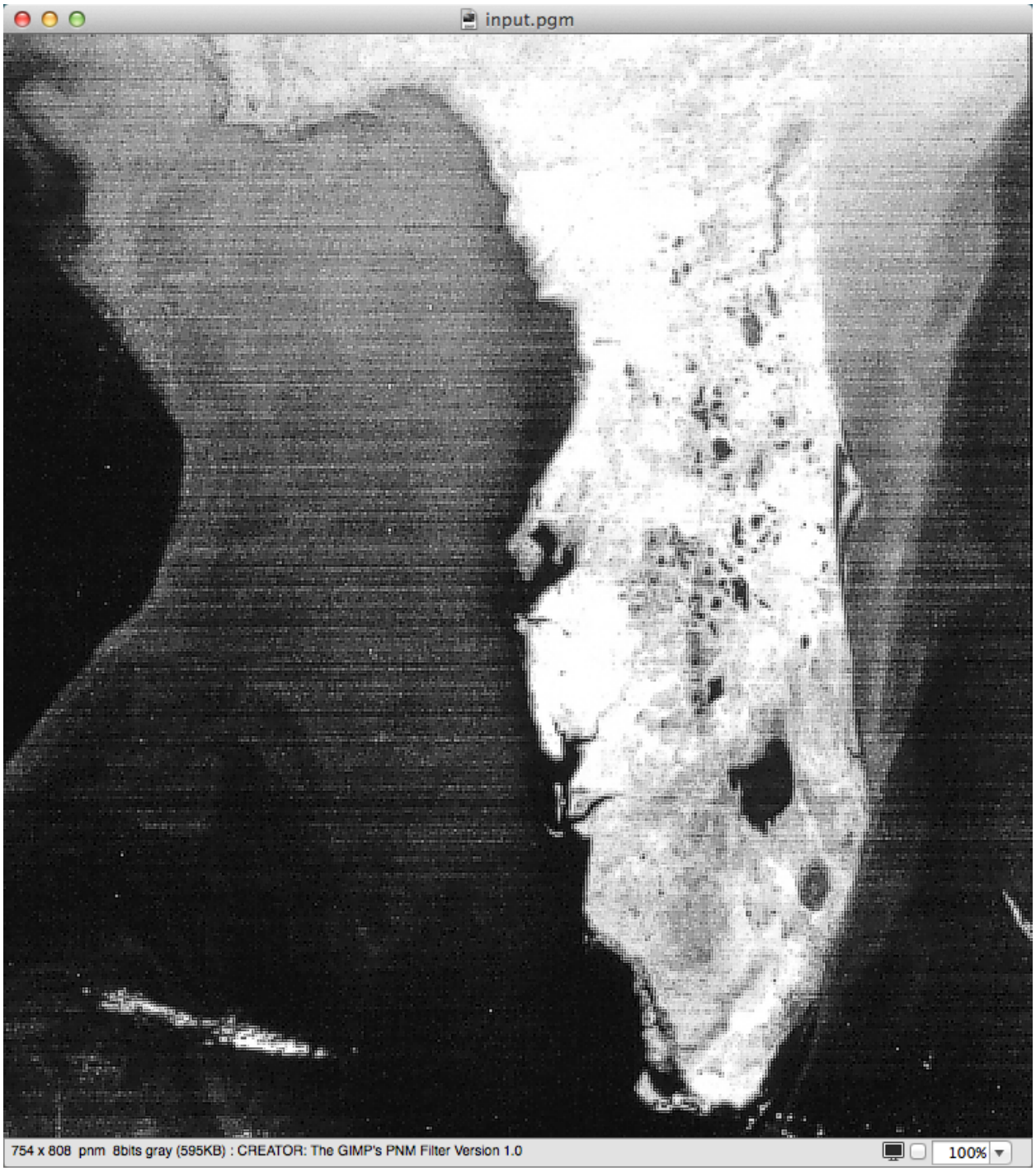
The Parallel Implementation of High-path Filter in Frequency Domain Using OpenMPI

Ye Guo & Di Zu

1. Instruction on how to compile and run my code on cluster
   1) Copy the all files in the folder named "source codes" into cluster.
   2) In the file 2DDFT.c, the argument radius, which is the radius of low-pass filter, could be modified where I put a note "you can modify the parameter radius here !!!" in the end of a line of code. The default value of radius is sqrt((width/2)^2 + (height/2)^2)/2.
   3) Modify number of processes to be used in file named "mpi–run.sbatch". For example, you can change the last line of code in this file, which is "mpirun -n 2 DFT florida_satellite.pgm result.pgm". The number "2" in this sentence is corresponding to the number of processes that will be assigned in this task.
      Note that the number of processes here should be at least 2 because the process 0 only take care of assigning jobs to other processes, while the other processes only compute the 1D DFT of the data block passed in. So it needs at least 2 processes to complete this task.
   4) Run –sbatch mpi–build.sbatch
      –sbatch mpi–run.sbatch
      And there will be two slurm–XXXX.out (XXXX is the ID of task running on cluster) files generated, the second .out file will show the result of execution.
      Note that it might take a few seconds for the cluster to run the program, so the user is better to wait for a few seconds to check the .out file after running the command "-sbatch mpi-run.sbatch".

2. Comparison of running time for different number of processes being used (Radius = sqrt((width/2)^2 + (height/2)^2)/2)

   Source Image:



Result image:

Comparison of speed with different number of processes:

a) # of processes = 2

```
begin reading PGM....
Width=754, Height=808
Maximum=255
[compute-001:41200] 1 more process has sent help message help-mpi-btl-tcp.txt / invalid if_inexclude
[compute-001:41200] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages
The total time cost is 171.474634 seconds
Begin writing PGM....
```

The total time consumed is 171.47s

b) # of processes = 5

```
begin reading PGM....
Width=754, Height=808
Maximum=255
[compute-001:41229] 4 more processes have sent help message help-mpi-btl-tcp.txt / invalid if_inexclude
[compute-001:41229] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages
The total time cost is 43.356911 seconds
Begin writing PGM....
```

The total time consumed is 43.35s

c) # of processes = 9

```
begin reading PGM....
Width=754, Height=808
Maximum=255
[compute-001:41274] 8 more processes have sent help message help-mpi-btl-tcp.txt / invalid if_inexclude
[compute-001:41274] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages
The total time cost is 21.809193 seconds
Begin writing PGM....
```

The total time consumed is 21.80s

d) # of processes = 17

```
begin reading PGM....
Width=754, Height=808
Maximum=255
[compute-001:41308] 16 more processes have sent help message help-mpi-btl-tcp.txt / invalid if_inexclude
[compute-001:41308] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages
The total time cost is 20.686781 seconds
Begin writing PGM....
```

The total time consumed is 20.68s

3. Conclusion

With more number of processes being used, the faster the speed could be. However, if the number of processes is up to 9, the rate of speed increasing become very small, in another word, there is no necessary to involve more than 9 processes in running this program.