Università
della
Svizzera
italiana

Facoltà
di scienze
informatiche

High Performance Computing **2017**

Student:Timon Willi

Solution for Assignment 5

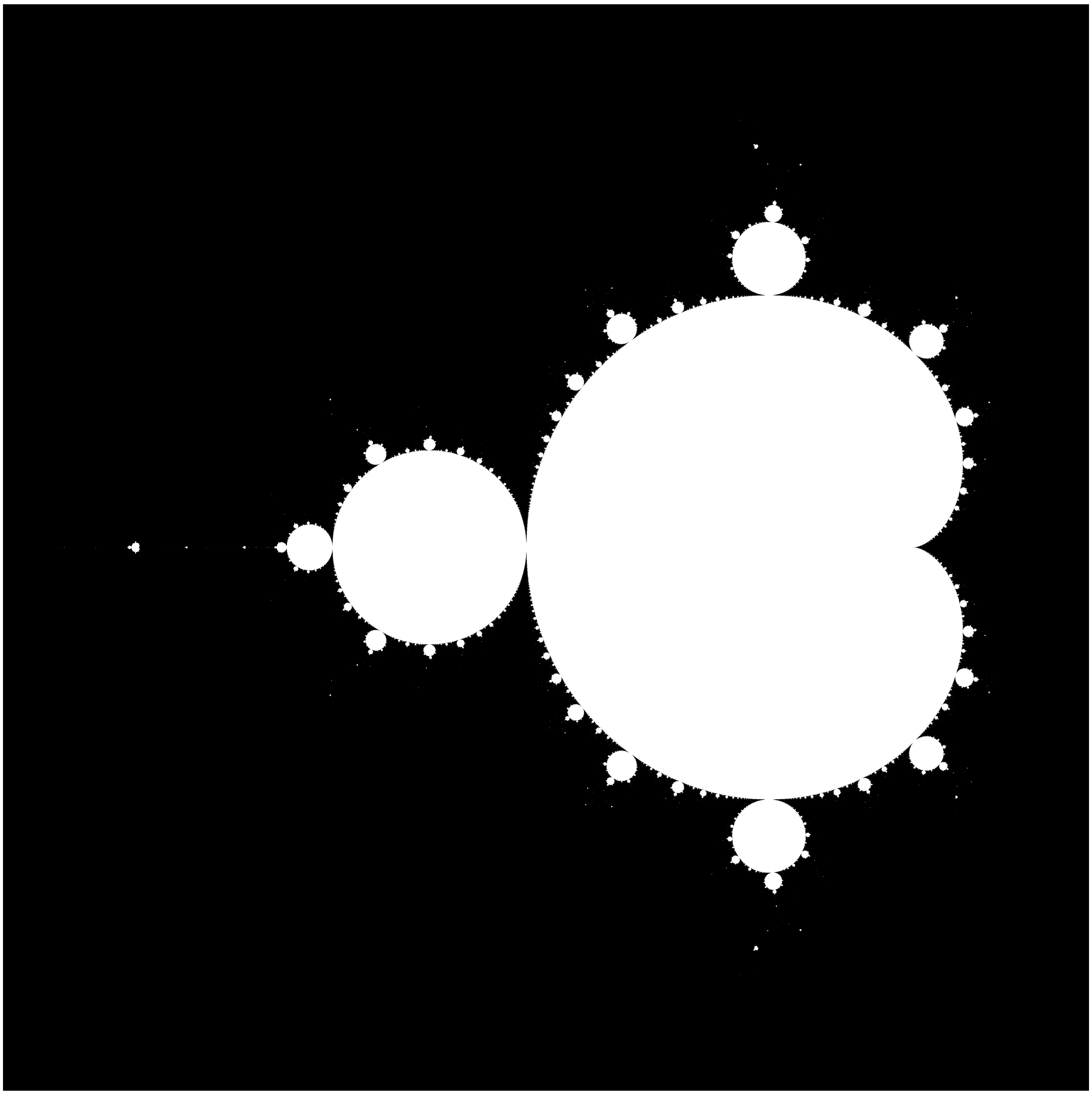
Due date: 25 October 2017, 13:30

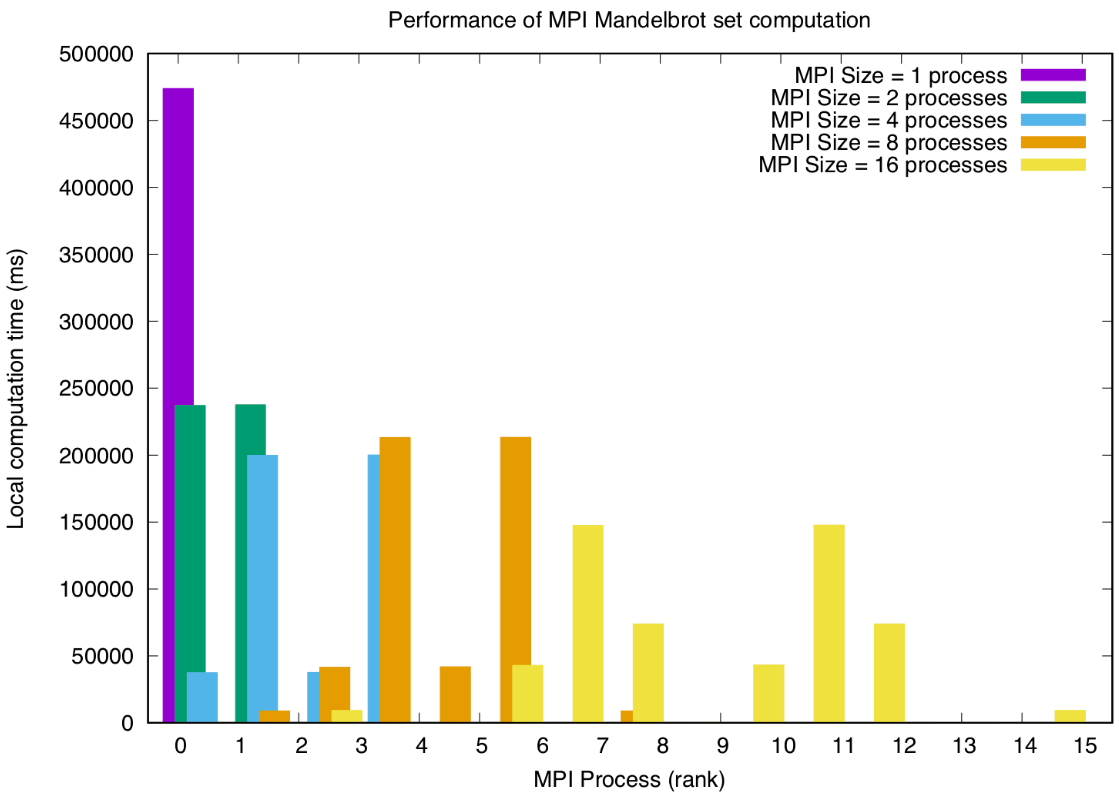
1. Parallel Programming with OpenMP *(40 Points)*

```
[stud26@icsnode21 ghost] mpirun -np 16 ./ghost data of rank 9 after communication
9.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 9.0
8.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 10.0
8.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 10.0
8.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 10.0
8.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 10.0
8.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 10.0
8.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 10.0
8.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 10.0
8.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 10.0
8.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 10.0
8.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 9.0 10.0
9.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0 9.0
```

2. Parallel reduction operations using OpenMP *(60 Points)*

The more processes are used, the less time it takes to compute. The work is not distributed completely regularly since some areas require more computation than others. We can see for example that 16 processes is not using most resources provided, which leads to the suggestion that a number between 8 and 16 processes is the most resource efficient version. Another possibility would be to choose a partitioning of the picture, such that the work is distributed more equally. However, this would have to be problem specific.




Performance of MPI Mandelbrot set computation
Local computation time (ms)
MPI Process (rank)
MPI Size = 1 process
MPI Size = 2 processes
MPI Size = 4 processes
MPI Size = 8 processes
MPI Size = 16 processes
0
50000
100000
150000
200000
250000
300000
350000
400000
450000
500000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15