



Lab Parallelization · Summer Semester 2017

Prof. Dr. Ulrich Meyer
Dipl. Inf. Gerhard Leuck

Assignment 1

Hand out: 25.04.2017

Hand in: 8.05.2017, till 4:00 pm at ppva-tut@cs.uni-frankfurt.de

Task 1

Create your own system environment for MPICH3 like explained in the lab.

Copy the example program `hellompi.c` from `/home/lab/2017/src` or download it from <http://www-intern.rbi.cs.uni-frankfurt.de/rbi/parallelization-lab>

Compile the program.

- a) Execute the program so that it runs on all compute nodes using ethernet.
- b) Execute the program so that it runs on all compute nodes using infiniband.

Modify the program so that only the root process (process with rank 0) prints the output of all processes in the order of the ranks. By use of the flag `-h` the program should give information about how it can be started and which program parameters are available.

Task 2

a)

Write a MPI program which calculates the minimum, maximum and sum of a random set of integers. Each process generates a set of m random integers there m is given as command line parameter. Process with rank 0 sends the results to process 1. Process 1 recalculates the values and sends the results to process 2. Then process 2 recalculates and forwards the results to process 3 and so on. The process with the highest rank sends its results to process 0. Finally process 0 prints out the overall result.

b)

If p processes are started, the algorithm in task 2a needs p steps. That is very inefficient. Write a program which solves the problem in $\log_2(p)$ steps implementing a tree-like communication pattern. Measure the total time needed for computation and communication in task 2a and task 2b and compare the results. You can assume that the number of processes is $p = 2^n$.