Technische Universität Kaiserslautern
Fachbereich Informatik
RHRK
Priv.–Doz. Dr. Josef Schüle

Introduction in High Performance Computing
Exercise Sheet 4

MPI Benchmark
Write an MPI program to benchmark the network bandwidth for a `MPI_Send` and `MPI_Recv` pair. Measure the time with help of `MPI_Wtime`. Use 2 tasks for the message sizes

$$\{1K, 10K, 100K, 1M, 10M, 100M\}$$

(all Bytes). Calculate and depict the bandwidth for

- 2 tasks on the same node (but the node dedicated to you)
- 2 tasks on two nodes (both nodes dedicated)

Extrapolate the graph to a message size of length 0 and determine that way the latency of the network. Hints: Measure the time on the sender site and assure with a small message returned (even size 0 is possible) that the message was actually received. Don't forget, to subtract the time for this short message from your results.

Classical Conjugate Gradient Method (again)
Back to sheet 3 resp. sheet 1 exercise 3. Now start in the main program and the main iteration loop. Assuming that the master task initializes all data draw a flow graph for all data to be exchanged. Please do not skip this step - otherwise you will pay for it later.
Now that you have a proper data flow graph think about it again and write into your graph (not the code) the necessary MPI routines and functionalities. (By the way - a reduction is available via a MPI_Reduce call or in case all need the result a MPI_Allreduce.) Take special care about the call to `sgemv` which performs a multiplication of a matrix with a vector.
Now that you have the latency and the bandwitdt from the previous exercise you may think within the SPM-Model about the possible speedup you may achieve for you data layout. Adjust your data flow accordingly to achieve best results in theory.
Now finally start to program and compare your measured times with theory for different sizes `n` and different degrees of parallelism.