Prof. Dr. C. Stamm christoph.stamm@fhnw.ch Tel.: 056 202 78 32

Exercise 6

1 Shellsort

In Exercise 4 you have studied and modified an MPI based parallel version of odd-even transposition sort. Now you should implement the parallel version of Shellsort described on Slide 10.8, which uses in its second phase odd-even transposition sort. Make use of MPI and change the data type of the array elements from `int` to `float`, that you can compare the performance with your Quicksort implementation of task 2. Measure the performance of your implementation for varying array sizes $n = 2^{3k}$, $k \in [5, 9]$.

2 Quicksort

Given are the files “main.cpp” and “quicksort.cpp”. “main.cpp” contains a test environment and “quicksort.cpp” contains an already implemented sequential version of Quicksort (e.g. Slide 10.25).

a) Implement an efficient parallel version of Quicksort for shared memory systems. Let the number of processing elements p be an additional user defined parameter.

b) Compare the performance of your parallel Quicksort implementation with the performance of the serial and parallel standard sorting algorithms, e.g. `std::sort(…)`, `parallel_sort(…)`, and with your MPI implementation of Shellsort.

   Plot the performance for varying array sizes $n = 2^{3k}$, $k \in [5, 9]$, analyze and discuss the plotted curves.