Institut für Informatik — TU München WS 2013/14
Scientific **C**omputing in **C**omputer **S**cience
Michael Bader, Alexander Heinecke, Alexander Breuer

Masterpraktikum Scientific Computing (High Performance Computing) Exercise Sheet 2: Shared-Memory-Parallelization with OpenMP

Tutorial on 05.11.2013

Exercise 5 „Shared-Memory π-calculation“(2 Points)

With $\phi(x) = \frac{1}{1+x^2}$ we have $\int \phi(x)dx = arctan(x)$.

Hence, π can be calculated through integration of $\phi(x)$!

- Develop a serial implementation that integrates function $\phi(x)$ over $[0, 1]$!
- Parallelize your application using `OpenMP`. Please implement two different versions using both *critical directive* and *reduction clause* in order to ensure the correct summation order!
- Use `OMP_NUM_THREADS` in order to start your application with different thread counts (up to 8 or 16, depending on your machine)! Calculate the achieved speed-up and provide interpretations. Please perform weak and strong scaling studies! Weak scaling: if you double the number of threads you also double the problem size! Strong scaling: You keep the problem size constant but increase the number of threads!

Hint: Use the mid-point rule for integration! Therefore, split the unit-interval into n equal sized sub-intervals with length $h = \frac{1}{n}$. For each mid-point $\check{x}_i$ of each sub-interval, calculate the value of function $\phi(\check{x}_i)$. Afterwards, sum up all function values! In the end multiply the result with $4 \cdot h$!

Explain, why this method works!

Exercise 6 „Matrix-Matrix-Multiplication II“(2 Points)

Parallelize your application of exercise 4 with OpenMP! Please ensure that each cache-block is handled by only one thread. Provide weak- and strong-scaling results by using sufficient figures and plots!

Exercise 7 „Quicksort“(2 Points)

Parallelize the quicksort implementation given in `quicksort.c`. Please employ the task-concept of OpenMP 3.1. Use the `final` clause for stopping the parallelization of the recursion at sufficient level of the recursion. Examine the scalability (strong scaling) for different problem sizes and plot your results!

Have fun!

Deadline: 18.11.2013, 9.00 am! Please mail to **heinecke@in.tum.de**. If there is no submission until this deadline, the exercise sheet is graded with 0 points!

The next tutorial is on Nov. 20 in room 02.06.020 (please check TUM-online for change prior to the meeting)!

Please download applications-frames from
`http://www5.in.tum.de/wiki/index.php/Masterpraktikum_Scientific_Computing_-_High_Performance_Computing_-_Winter_13`