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

Introduction in High Performance Computing
Sheet 1

Hints Please specify hardware characteristics for the nodes choosen, especially the speed of the CPU, the size of the cache and, if available, the speed of the DRAM and the cache topology (1st, 2nd, 3rd,....). See task to the topology of the system below.
Hand in the batch script for all exercises. After testing your programs you should finally run the benchmarks on a certain hardware (CONSTRAINT) and a node in exclusive mode (--exclusive option to sbatch script) to obtain reproducible results.

1. Determine the topology of the used systems

Find a way (*apropos topology* in a terminal window may help) to determine the memory and processor topology of the system used. Use the CONSTRAINT specification in a batch script to determine the topology of a compute node (not the head nodes).

2. First experience with benchmarks

Benchmarking is essential for High Performance Computing (HPC). This exercise gives a slight idea of writing your own benchmark.
Write your own benchmark with help of the supplementary file `Benchmark.h` as header file. Find out, how the time measurement is called. Benchmark the loop

```
for(i=0;i<N;i++) y[i]=2.3*x[i]+y[i];
```

after allocating and initializing the array `y` and `x` with random numbers. Both arrays are of type double precision. The size of the arrays should vary with `N` in the range of 10,000 to 100 Million. Choose the steps inbetween properly (Incrementing with fix increment is not a good idea). Use the number of FLOPS as measure for the output instead of time. Depict your results graphically and choose a proper scaling of axis. You might think of 2 plots - one for small N and one for larger ones.
In some cases the naive way of writing this exercise in a single function in a single file gives unrealistic results, especially with higher optimization levels to the compiler. Write the above for loop in a file by its own and perform the compilation in two steps (think about a `Makefile`. This will help you for future exercises and programming tasks).

Bonus

Experiment with the command `numactl`. Run your programs with memory bound to the same CPU and to a different CPU.

3. Tuning Code

Copy the source files in directory `/scratch/HPC_I/E01/`. Your tasks is to optimize (=reduce) the execution time of this program.
If you want to optimize the run time of program sources you should start with the function requiring most time. This is typically the task of a so called profiler. For the Intel compiler system we have licensed `vtune` as visual profiler. Please inform yourself about prerequisites to run and use a profiler meaningful (Tip: Compiler options). `vtune` is callable with a graphical interface `amplxe-gui`. You may start it with

```
rhrk-launch --title Vtune --module intel/latest --command amplxe-gui
```

Please close this window as soon as possible. The number of licenses is limited and other students are waiting.
After you have identified the hotspot, try to optimize the code. It is a bit tricky, as optimization of the hotspot function itself won't help very much.