

Exercise Sheet 8

Discussion starts from Monday, June 29, 2020

8.1 NUMA (Theory)

- Refresh your knowledge about the concept of Non-Uniform-Memory-Access (NUMA) and explain how different CPU cores can face different access latencies for different regions of the main memory.
- Our test system is equipped with hyperthreading, i.e. each physical core provides 2 logical compute threads. Which NUMA latencies do you expect for the logical threads, compared to the physical cores?
- Go through the code template in your SVN folder (numa_test.c). In this template, a memory buffer is allocated for each available NUMA region. Afterwards, a pthread is started on each CPU thread, where the threads run synchronized. Why does the program access the allocated NUMA buffers entirely once before the measurement begins?

8.2 Determining NUMA latencies

- Perform an access time measurement on each pthread on each thread, such that the access time to each NUMA region from each thread is measured. Perform random accesses within each NUMA region and measure the total access time for each NUMA region.
Hint: Perform round about 1.000.000 memory accesses per thread and measure the total time
- Print out the measurement results in an understandable way, such that the access time for every thread / region pair can be easily seen.

8.3 Evaluation of the measurement

- Have a closer look now at the measured access latencies. What insights can you infer about the NUMA architecture on our system?
Hint: It is useful to sort the measured data and determine the range, i.e to sort the NUMA regions along their access latency.
- Run the tool `lstopo` (`lstopo --of pdf > topo.pdf`) to get an overview about the NUMA architecture on the test system. Can you verify the displayed topology with you measurement results?

General Information: Further information can be found under <https://ls12-www.cs.tu-dortmund.de/daes/de/lehre/lehrveranstaltungen/summersemester-2020/rechnerarchitektur-deutsch.html>. Submitting solutions to the exercise sheets is not required.