Exercises for Architectures of Supercomputers

7th Exercise, 8./9.1.2020

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Cache hit rate in practice

- After **simulating** an Intel processor's L1 cache, in this exercise the **hardware performance counters** will be used to investigate the processor's behavior **in practice**
  - We can then compare these findings to our theoretical results from the previous exercise

- Refresher (see 4th exercise for details)
  - Modern processors feature **hardware performance counters**, which enable logging of certain **events** that occur on the processor
  - Mode of operation
    - Performance counters can be programmed with a particular event (e.g., the number of branch instructions executed)
    - Once programmed, the counter is incremented whenever the event (e.g., a branch instruction) occurs

Cache hit rate in practice

- Write a program that takes the size of an array as argument
  - Allocate memory for the array (data type does not matter, although `float` or `double` is recommended
  - Initialize the array with the value 1.0
- Recreate data-access pattern used in simulator
  - A regular streaming access to an array with a stride of 64 bytes
  - Implement the access in a separate function
  - Execute the data-access loop 10,000 times so the cache misses encountered in the first iteration will not significantly bias the result
- Measure the `GROUP MEM_UOPS_RETIRED_LOADS` and `MEM_LOAD_UOPS_RETIRED_L1_HIT` events for the function
  - Hint: Reuse code from 4th exercise to save time
    - Surround your function call with the appropriate `LIKWID_MARKER_START` and `LIKWID_MARKER_STOP` macros
    - `#include likwid.h` and use the appropriate compiler flags (`-I`, `-L`, …)
    - Make sure to use `likwid-perfctr`'s marker API (`-m`)

Exercise 1: Cache Simulator

- Use `likwid-perfctr` to determine the cache hit rate for arrays of different sizes (20kB – 40kB)

- Extend last week's figure with a graph of the obtained data
  - *x*-axis (linear): array size [kB]
  - *y*-axis (linear): cache hit rate [%]

- Compare the results of the simulator to those of the processor