CPE 631 Advanced Computer Systems Architecture: Homework #4

Computer System Simulation

In this homework, we will focus on Multi2Sim - a simulation framework for CPU-GPU heterogeneous computing platforms. Multi2Sim includes models for superscalar, multithreaded, and multicore processors, as well as GPU architectures. It supports the following ISAs: X86, ARM, MIPS, NVIDIA Fermi, AMD Southern Islands. The ultimate source of information (manuals, downloads, benchmarks) is the Multi2Sim web site at https://www.multi2sim.org/. To get starting use the LaCASA Multi2Sim tutorials that can be found at: http://lacasa.uah.edu/portal/index.php/tutorials/36-multi2sim.
Multi2Sim@blackhawk: Mult2Sim is available @blackhawk in the /apps/arch/multi2sim-4.2. Add /apps/arch/multi2sim-4.2/bin in your path. The benchmarks that are precompiled to run on multi2sim can be found at /apps/arch/m2s_benchmarks. To run your own benchmarks, read the LaCASA tutorials at the following web page http://lacasa.uah.edu/portal/index.php/tutorials/36-multi2sim. Please read all four tutorials that explain how to get started with multi2sim, how to configure memory, how to configure x86 model, and how to configure a system.

Question #1 (60 points) Explore effectiveness of the blocking optimization for matrix multiplication
Use the single-threaded matrix-multiplication example code to perform detailed (cycle-accurate) simulation with the default x86 processor configuration with the following memory hierarchy. The processor has split instruction (L1I) and data (L1D) caches. The L1I caches and L1D caches are 4KB, with cache blocks 64 bytes, 4-way set associative. Assume a unified L2 cache (L2U) with the following parameters: 32 KB, 4-way set associative, 64-byte blocks. All caches use the LRU replacement policy. The latency for L1I and L1D is 4 clock cycles and the latency for L2U is 10 clock cycles. Use squared matrix multiplication 256x256. Will blocking optimization improve performance? If yes, what block size appear to give the best performance?

Turn in: your answers, source code, simulator outputs, and plots.

Question #2. (40 points)
Write a C program that performs a DAXPY kernel (Double aX+Y => Y). Let the size of the array be an input parameter to your program. The X and Y are initialized as follows: X(i)=i; Y(i)=i/2; a=3.0.
What is program execution time for arrays of 2048 elements (assume the default x86 configuration)?
Evaluate the effectiveness of the loop unrolling (unroll the loop 4 times). What is the program execution time with loop unrolling?

Turn in: your answers, source code, simulator outputs, and plots.

Note: In addition to submission through canvas, you should upload in your work on blackhawk at the following directory: /home/work/cpe631/<your username>/hw4. The directory should contain source codes, simulator outputs, and reports in either text or pdf format.