CS 252 Graduate Computer Architecture

Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

Spring 1994 — Dave Patterson
Homework #4 — Remzi Arpaci

Due Monday, September 20, Noon in 634 Soda

You should work with your project partner(s) for this assignment.

In this problem, you will be asked to measure some performance characteristics (e.g. MFLOPS, MIPS) of a machine of your choice. The approach you will take is to measure the run time a small kernel program. Given the number of operations performed and the run time, you can find the operations/second. It will make you're life easier if you use a Unix machine, but you make use any machine which has an accurate clock and C-compiler.

On the instructional machines in `~cs252/f94/hw4/gauss.c` is a program where the inner loop is an L/U decomposition, which is where most of the time is spent in gaussian elimination.

The program is structured as two loops around the basic L/U decomposition. The inner loop on line 68 does a number of L/U solves. The outer loop starting at line 60 checks the reproducibility of the inner loop.

If the timer resolution exceeds 5% of the inner loop, the program displays a warning message. Increase the number of iterations of the inner loop.

Problem 1: Getting accurate time

System timers have a finite resolution, so you must find the resolution of the system you're on and adjust the inner loop starting on line 68 of gauss.c accordingly.

1.a. What system and clock rate (e.g. HP 720 @50 Mhz, i486@33Mhz) and compiler (e.g. native, gnu) are you running on? What is the resolution of the times() system call on you're system? How big does the matrix A have to get before the inner loop (lines 68-77) takes more than one clock resolution to execute?

Unfortunately, getting reproducible results is not as simple as it might seem. External events like context switches, I/O interrupts, and clock interrupts can effect the elapsed time of the program.

1.b. How many runs of the outer loop are required for the above size before the error drops bellow 5%

Problem 2: MFLOPS

Run the program with the following sizes for the matrix A and compiler optimization levels:

Sizes: 32x32, 128x128, 300x300

Optimizations: none, -O, -O4 (or whatever the maximum your compiler allows).

What is the MFLOPS of each size which each optimization level? Why do different sizes give different MFLOPS? Why might the MFLOPS you measured be different from the manufacturer's claim? (For the this and next problems turn in a little table with your results)

Problem 3: MAPS

MAPS is Millions of Array oPerations per Second. Thus, the L/U loop has 4 APs. For the same sizes and optimization levels as program 2, find the MAPS of your machine

Problem 4: MIntS

MIntS is Millions of Integer operations per Second. Our definition of an Int is and array indexing operation or loop increment operation. How may Int's are in the L/U loop? For the three sizes and optimization levels, how what are the MInts of your machine?

Problem 5: MIPS

For this and problem 6, get the assembly output of your compiler. (use the -S flag). Find the L/U loop in the assembly listing and count the number of instructions in the loop. If you have trouble finding the loop, you can insert some dummy function calls(e.g. `printf`) around the loop and look for the jumps or branches to the calls.

5.a For each optimization level, what is the total number of instructions in one loop iteration as well as the number of:

a. Floating point instructions?

b. Data instructions? (those that load/store into the arrays)

c. Integer Instructions? (Access to local variable, loop counters, etc.)

c. other (anything else)

5.b Again, for the same sizes and optimization levels as problem 2, what is the MIPS of your machine? Why might it be different from the manufacturer's claim?

6: CPI

For what size and optimization level is do you get the best CPI? The worst? What causes the difference, if any?