Homework – Preperation for Exams Answer Sheet

Questions

1. Consider program P, which runs on a 1 GHz machine M in 10 seconds. An optimization is made to P, replacing all instances of multiplying a value by 4 (mult X,X,4) with two instructions that set x to x + x twice (add X,X; add X,X). Call this new optimized program P'. The CPI of a multiply instruction is 4, and the CPI of an add is 1. After recompiling, the program now runs in 9 seconds on machine M. How many multiplies were replaced by the new compiler?

Answer 1

Program P running on machine M takes (10^9 cycles/seconds) * 10 seconds = 10^{10} cycles. P' takes (10^9 cycles/seconds) * 9 seconds = 9 x 10^9 cycles. This leaves 10^9 less cycles after the optimization.
Everytime we replace a mult with two adds, it takes 4 - 2 * 1 = 2 cycles less per replacement.
Thus, there must have been 10^9 cydes /(2 cydes/replacement) = 5 X 10^8 replacements to make P into P'.

2. Your company could speed up a Java program on their new computer by adding hardware support for garbage collection. Garbage collection currently comprises 20% of the cycles of the program. You have two possible changes to the machine. The first one would be to automatically handle garbage collection in hardware. This causes an increase in cycle time by a factor of 1.2. The second would be to provide for new hardware instructions to be added to the ISA that could be used during garbage collection. This would halve the number of instruction needed for garbage collections but increase the cycle time by 1.1. Which of these two options, if either, should you choose?

Answer 2

The first option reduces the number of instructions to 80%, but increases the time to 120%. Thus it will take: 0.8 * 1.2 = 0.96 as much time as the initial case.
The second option removes 20/2 = 10% of the instructions and increases the time taken to 110%. Therefore it will take 0.9 * 1.1 = 0.99 times as much time as the initial case.
Therefore, the first option is the faster of the two, and it is faster than the original, so you should have hardware automatically do garbage collection.

3. The table below shows the number of floating-point operations executed in three different programs and the runtime for those programs on three different computers:

Program	Floating-point operations	Execution time in seconds		
		Computer A	Computer B	Computer C
Program 1	$5x10^9$	2	5	10
Program 2	$20x10^9$	20	20	20
Program 3	$40x10^9$	200	50	15

One user has told you that three programs in the table constitute the bulk of his workload, but he does not run them equally. The user wants to determine how the three computers compare when the workload consists of different mixes of these three programs. (You know you can use the arithmetic mean to find the relative performance.)

Suppose the total number of floating-point operations (FLOPs) executed in the workload is equally divided among the three programs. That is, program 1 runs 8 times for every time program 3 runs, and program 2 runs twice for every time program 3 runs. Find which computer is fastest for this workload and by what factor. How does this compare with the total execution time with equal numbers of program executions?

Answer 3

With the new weighting the total execution time for the group of programs is:
Computer A = 8 * 2 + 2* 2 0 + 1 * 200 seconds = 256 seconds
Computer B =8*5 + 2* 2 0 + 1 * 50 seconds = 130 seconds
Computer C = 8 * 10+ 2* 2 0 + 1 * 15 seconds = 135 seconds
So with this workload, computer B is faster by a factor of 135/130 = 1.04 with respect to computer C and a factor of 256/130 = 1.97 with respect to computer A.
This new weighting reflects a bias from the previous results by a bias toward program 1 and program 2, which resulted in computer A and computer B looking comparatively better than before.

4. Your book cites a pitfall the utilization of a subset of the performance equation as a performance metric. To illustrate this, consider the following data for the execution of given instruction sequence of 10^6 instructions in different processors.

Processor	Clock rate	CPI
P1	4 GHz	1.25
P2	3 GHz	0.75

a) A fallacy is to consider that processor executing the largest number of instruction will need a larger CPU time. Considering that P1 is executing a sequence of 10^6 instructions and that the CPI of processors P1 and P2 do not change, determine the number of instructions that P2 can execute in the same time that P1 needs to execute 10^6 instructions.

b) A common fallacy is to use MIPS (millions of instructions per second) to compare the performance of two different processors, and consider that the processor with the largest MIPS has the largest performance. Check if this is true for P1 and P2.

Answer 4

a)

P1: 10^6 instructions, $T_{cpu}(P1) = 0.315 \times 10^{-3}$ s

P2: $T_{cpu}(P2) = N \times 0.75/3 \times 10^9$ then $N = 1.26 \times 10^6$

b)

$MIPS(P1) = 4 \times 10^9 \times 10^{-6}/1.25 = 3200$

$MIPS(P2) = 3 \times 10^9 \times 10^{-6}/0.75 = 4000$

$MIPS(P1) < MIPS(P2)$, performance(P1) < performance(P2) in this case

5. Another pitfall, relating to the execution of programs in multiprocessors systems, is expecting improvement in performance by improving only the execution time of part of the routines. The following table shows the execution time of five routines of a program running on different numbers of processors.

	# Processors	Routine A (ms)	Routine B (ms)	Routine C (ms)	Routine D (ms)	Routine E (ms)
a.	2	20	80	10	70	5
b.	16	4	14	2	12	2

a) Find the total execution time and by how much it is reduced if the time of routines A,C, and E is improved by 15%.

b) By how much is the total time reduced if routine B is improved by 10%?

c) By how much is the total time reduced if routine D is improved by 10% ?

Answer 5

a)
Without reduction in any routine:
a. total time 2 proc = 185 ns
b. total time 16 proc = 34 ns
Reducing time in routines A, C and E:
a. 2 proc: T(A) = 17 ns, T(C) = 8.5 ns, T(E) = 4.1 ns, total time = 179.6 ns ==> reduction = 2.9%
b. 16 proc: T(A) = 3.4 ns, T(C) = 1.7 ns, T(E) = 1.7 ns, total time = 32.8 ns ==> reduction = 3.5%
b)
a. 2 proc: T(B) = 72 ns, total time = 177 ns ==> reduction = 4.3%
b. 16 proc: T(B) = 12.6 ns, total time = 32.6 ns ==> reduction = 4.1%
c)
a. 2 proc: T(D) = 63 ns, total time = 178 ns ==> reduction = 3.7%
b. 16 proc: T(D) = 10.8 ns, total time = 32.8 ns ==> reduction = 3.5%

6. Assume for arithmetic, load/store, and branch instructions, a processor has CPIs of 1, 12, and 5, respectively. Also assume that on a single processor a program requires the execution of 2.56E9 arithmetic instructions, 1.28E9 load/store instructions, and 256 million branch instructions. Assume that each processor has a 2 GHz clock frequency.

Assume that, as the program is parallelized to run over multiple cores, the number of arithmetic and load/store instructions per processor is divided by 0.7 x p (where p is the number of processors) but the number of branch instructions per processor remains the same.

a) Find the total execution time for this program on 1, 2, 4, and 8 processors, and show the relative speedup of the 2, 4, and 8 processor result relative to the single processor result.

b) If the CPI of the arithmetic instructions was doubled, what would the impact be on the execution time of the program on 1, 2, 4, or 8 processors?

c) To what should the CPI of load/store instructions be reduced in order for a single processor to match the performance of four processors using the original CPI values?

Answer 6

a)

p	# arith inst.	# L/S inst.	# branch inst.	cycles	ex. time	speedup
1	2.56E9	1.28E9	2.56E8	7.94E10	39.7	1
2	1.83E9	9.14E8	2.56E8	5.67E10	28.3	1.4
4	9.12E8	4.57E8	2.56E8	2.83E10	14.2	2.8
8	4.57E8	2.29E8	2.56E8	1.42E10	7.10	5.6

b)

p	ex. time
1	41.0
2	29.3
4	14.6
8	7.33

c) 3