Outline of Lecture

1. The Role of Computer Performance

2. Measuring Performance

Summary

The CPU time can be decomposed as follows:

$$CPU\ \ time\ =\ \frac{Instructions}{Program} \times \frac{Clock\ \ cycles}{Instructions}$$

$$\times \frac{Seconds}{Clock\ \ cycle} = \frac{Seconds}{Program}$$

- The basic components of performance of a computer are measured as follows:

Components of performance	Units of measure
CPU execution time for a program	Seconds for the program
Instruction count	Instructions executed for the program
Clock cycles per instruction (CPI)	Average clock cycles / Instruction
Clock time	Seconds / Clock cycle

The Challenge

There are trade-offs between the various factors that affect performance (i.e., clock cycle, CPI, IC) - optimizing one of them can lead to worsening the performance of the other.

Example

A compiler designer is trying to decide between two code sequences for a particular computer. The hardware designers have supplied the following facts:

Instruction class	CPI for this instruction class
A	1
B	2
C	3

For a particular high-level-language statement, the compiler writer is considering two code sequences that require the following instruction counts:

Code sequence	IC for instruction class		
	A	B	C
1	2	1	2
2	4	1	1

Which code sequence executes the most instructions? Which will be faster? What is the CPI for each sequence?

Answer

Sequence 1 executes 2 + 1 + 2 = 5 instructions.

Sequence 2 executes 4 + 1 + 1 = 6 instructions.

∴ Sequence 1 executes fewer instructions

To find which one is faster, we should find the CPU clock cycles needed for each sequence.

$$CPU\ clock\ cycles = \sum_{i=1}^{n} CPI_i \times IC_i$$

CPU clock cycles_1 = (2x1) + (1x2) + (2x3) = 10 cycles

CPU clock cycles_2 = (4x1) + (1x2) + (1x3) = 9 cycles

∴ Sequence 2 executes faster than sequence 1.

$$CPI_1 = \frac{CPU\ clock\ cycles_1}{IC_1} = \frac{10}{5} = 2$$

$$CPI_2 = \frac{CPU\ clock\ cycles_2}{IC_2} = \frac{9}{6} = 1.5$$

Some "Misleading" Performance Measures

There are certain computer performance measures which are famous with computer manufacturers and sellers - but are misleading.

MIPS: **Meaningless Indicator of Processor Speed**

- ***MIPS*** (**M**illion **I**nstructions **P**er **S**econd) ***depend*** on the instruction set - cannot compare two computers with different instruction sets.
- MIPS varies between programs on the same computer - different programs use different instruction mixes.
- MIPS can vary inversely to performance (See textbook for concrete example).

MFLOPS: Focus on one type of work

- MFLOPS (**M**illion **F**loating-point **O**perations **P**er **S**econd) depends on the program. Must be FP intensive.
- MFLOPS depends on the computer as well.
- The floating point operations vary in complexity (e.g., add & divide).
- Assumes other type of instructions are not "work".

Peak Performance: Performance the manufacturer guarantees you won't exceed.

- Difference between peak performance and average performance is huge.
- Peak performance occurs only for meaningless codes (e.g., executing the same instruction over and over).

Benchmarks

Benchmarking are the programs to use to measure performance

- **Real programs:** CAD tools, text processing (Latex), compilers (C) - have inputs, outputs, and options when the user wants to use them.
  - ➜ The most accurate way to characterize performance.
- **Kernels:** key pieces from real programs. Typically used to extract specific features of the machine.
  - ➜ They are good for focusing on individual features.
  - ➜ Often useful for processor designer.
  - ➜ Have little value.

Benchmark Suites

- These are a collection of programs that try to explore and capture all the strengths and weaknesses of a computer system (real programs, kernels).
  - ➜ Good benchmarks accelerate progress
    - ∗ good target for development
  - ➜ Bad benchmarks hurt progress
    - ∗ help real programs vs. sell machines (optimize the machine for the benchmark).
    - ∗ Inventions that help real programs may not help benchmark.

Some Useful benchmarks

- One good benchmark suite is **SPEC92**. It is used to evaluate workstations and servers.

- SPEC benchmark requires a complete description of the machine (CPU, memory, FPU, cache, etc.), compiler, operating system — any user can duplicate the exact experiment under the same conditions.

Comparing Performance

- Comparing the performance by looking at individ-

	Computer A	Computer B	Computer C
Program P1 (secs)	1	10	20
Program P2 (secs)	1000	100	20
Total time (secs)	1001	110	40

ual programs is not fair.

- A better measure would be the ***arithmetic mean***:

$$\frac{1}{n}\sum_{i=1}^{n} Time_i$$

where $Time_i$ is the execution time of program *i* of a total of *n* in the workload.

- An even better measure would be the ***weighted arithmetic mean***:

$$\sum_{i=1}^{n} Weight_i \times Time_i = WAM$$

where $Weight_i$ is the frequency of program i of a total of n in the workload.

- An another approach is to measure the performance of a new computer using a certain program by normalizing it to a reference machine.