CPU Performance Indicators

Contents

- Processor performance
- Memory performance

Processor speed

- Modern processors can have a high level of complexity, and detailed analytic modeling of their work is very difficult.
- Processor performance analysis is easy under the following simplifying assumptions:
  - Sequential execution of machine instructions
  - Instructions consists of an operation code followed by 0 to 3 addresses of arguments
  - Program execution is characterized by frequencies of opcodes and addressing modes (obtained using a dynamic trace program)

Dynamic trace program

- PC = program counter
- PSR = processor status register
  - N = negative flag (result of previous operation is negative)
  - Z = zero flag (the result of previous operation is zero)
  - V = overflow flag (overflow in previous operation)
  - C = carry flag (previous operation generates carry bit)
  - PRIO = current program priority code
  - T = trap flag (if 1, generate a trace interrupt at the end of each machine instruction)

Trace counters

- F[i] = frequency if i-th opcode
- Frequency of addressing modes
  - Direct
  - Indirect
  - Indexed
  - Register
  - Autoincremented register
  - Etc.
- Frequency of instruction size (bytes)
- Frequency of instruction formats
  - 0 address
  - 1 address (register or memory)
  - 2 addresses (R and/or M)
  - 3 addresses (R and/or M)

Dynamic trace program

Clear trace counters

Turn-on trace interrupt (T=1)

START: Fetch current instruction

PC++

Execute (current) machine instruction

Generate trace interrupt ----------→

If (not end of program) go to START ←----------

Print trace results

// Trace interrupt routine

Turn-off trace interrupt

Analyze instruction in [PC]

Increment trace counters

Turn-on trace interrupt

Return from interrupt

Typical overhead of the trace program

- 30 times!!!!

Probability of instructions

$$f_i = \text{ frequency of the } i^{\text{th}} \text{ opcode}$$

$$F = \sum_{i=1}^{n} f_i = \text{total number of executed instructions}$$

$$p_i = \frac{f_i}{F}, \quad \sum_{i=1}^{n} p_i = 1$$

Instruction mix time

$f_i =$ frequency of the i^{th} opcode

$t_i =$ mean time of the i^{th} instruction

$$T_{mix} = \sum_{i=1}^{n} p_i t_i = \text{mean instruction execution time}$$

Processor speed

$$\nu_{\mathrm{p}} = 1/T_{mix} \quad [\text{instructions per second}]$$

$$= 10^{-6}/T_{mix} \quad [\text{MIPS}]$$

What is MIPS?

- MIPS = Millions of Instructions Per Second

  or

- MIPS = Misleading Information about Processor Speed

Problems with MIPS

- No standardized workload (and the standard distribution of machine instruction frequencies)
- There are similar problems in finding the average instruction execution times (what is the distribution of addressing modes?)
- Meaning of comparison of processors having different architecture (e.g. different number of addresses per instruction)
- MIPS for hyperthreaded processors
- MIPS for multicore processors

Problems with MIPS (cont.)

- MIPS does not sufficiently reflect advanced instruction execution features
- MIPS and caching and paging
- The use of MIPS indicators is mostly restricted inside a specific architecture.
- Even with all these drawbacks MIPS is better than nothing (no processor speed indicator)

MIPS computation example

- Average control instruction takes 20ns and has probability 20%
- Average data transfer instruction takes 30ns and has probability 30%
- Average arithmetic instruction takes 50ns and has probability 50%
- Compute MIPS

MIPS computation example (cont.)

- Mean instruction time
  Tmix=0.2*20+0.3*30+0.5*50 = 4+9+25 = 38 ns = $38*10^{-9}$
- Proc speed = 10^{-6}/Tmix = $10^{-6}/(38*10^{-9})$ =1000/38 = **26.3 MIPS**