Advanced Computer Architecture

Assessing and Understanding Performance

Computer Science

Performance Metrics

- Possible measures:
  - response time – time elapsed between start and end of a program
  - throughput – amount of work done in a fixed time
- The two measures are usually linked
  - A faster processor will improve both
  - More processors will likely only improve throughput

System Execution Time

Consider a system X executing a fixed workload W

System Performance$_X$ = 1 / System Execution time$_X$

where System Execution time = response time

CPU Execution Time

- System Execution time is the total elapsed time from start till the end.
- CPU Execution time on the other hand is the time CPU spends computing for this task and does not include time spent waiting for I/O or running other programs.
- Similarly we also have CPU performance .

CPU Performance Equation - I

- CPU execution time for a program =
  CPU clock cycles for the program x Clock cycle time
- Clock cycle time = 1 / Clock rate
- then CPU execution time for a program =
  CPU clock cycles for the program / Clock rate

Example

- If a processor has a frequency of 3 GHz, the clock ticks 3 billion times ($3x\ 10^9$) in a second
- If a program runs for 10 seconds on a 3 GHz processor, how many clock cycles did it run for?
- If a program runs for 2 billion clock cycles on a 1.5 GHz processor, what is the execution time in seconds?
- If a program runs for 10 seconds on a 4 GHz processor, what clock rate is required for a new processor to run the same program in 6 seconds?

CPU Performance Equation - II

- CPU clock cycles for a program = No. of instrs. in the program x avg clock Cycles Per Instruction (CPI)

Substituting in equation - I,

- Execution time = clock cycle time x No. of instrs x avg CPI

Factors Influencing CPU Performance

CPU Execution time = clock cycle time x number of instrs x avg CPI

- Clock cycle time: from the frequency of processor
- Number of instructions: the quality of the compiler and the instruction set architecture
- CPI: the nature of each instruction and the quality of the architecture implementation

Example

Execution time = clock cycle time x number of instrs x avg CPI

Which of the following two systems is better?

- A program is converted into 4 billion MIPS instructions by a compiler ; the MIPS processor is implemented such that each instruction completes in an average of 1.5 cycles and the clock speed is 1 GHz

- The same program is converted into 2 billion x86 instructions; the x86 processor is implemented such that each instruction completes in an average of 6 cycles and the clock speed is 1.5 GHz

Amdahl’s Law

- Amdahl’s Law: performance improvements through an enhancement is limited by the fraction of time the enhancement comes into play

- Example: a web server spends 40% of time in the CPU and 60% of time doing I/O – a new processor that is ten times faster results in a 36% reduction in execution time – Amdahl’s Law states that maximum execution time reduction is 40%

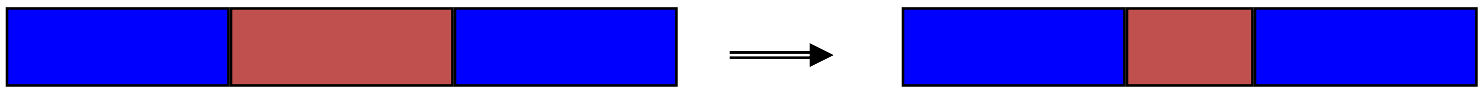
Amdahl's Law

$$ExTime_{new} = ExTime_{old} \times \left[(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}} \right]$$

$$Speedup_{overall} = \frac{ExTime_{old}}{ExTime_{new}} = \frac{1}{(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

Best you could ever hope to do:

$$Speedup_{maximum} = \frac{1}{(1 - Fraction_{enhanced})}$$



Amdahl's Law example

- New CPU 10X faster
- I/O bound server, so 60% time waiting for I/O

$$\text{Speedup}_{\text{overall}} = \frac{1}{\left(1 - \text{Fraction}_{\text{enhanced}}\right) + \dfrac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

$$= \frac{1}{\left(1 - 0.4\right) + \dfrac{0.4}{10}} = \frac{1}{0.64} = 1.56$$

- Apparently, its human nature to be attracted by 10X faster, vs. keeping in perspective its just 1.6X faster