LECT - 3

Amdahl's Law, MIPS & Practice Questions

Amdahl's Law

- Improving an aspect of a computer and expecting a proportional improvement in overall performance

$$T_{improved} = \frac{T_{affected}}{improvement\ factor} + T_{unaffected}$$

- Example: multiply accounts for 80s/100s
  - How much improvement in multiply performance to get 5× overall?

$$20 = \frac{80}{n} + 20$$

  - Can't be done!

- Corollary: make the common case fast

Amdahl's Law

- Amdahl's Law defines the speedup that can be gained by using a particular feature.
- Suppose that we can make an enhancement to a computer that will improve performance when it is used.

$$\text{Speedup} = \frac{\text{Performance for entire task using the enhancement when possible}}{\text{Performance for entire task without using the enhancement}}$$

$$\text{Speedup} = \frac{\text{Execution time for entire task without using the enhancement}}{\text{Execution time for entire task using the enhancement when possible}}$$

- **Two Factors**
- *The fraction of the computation time in the original computer that can be converted to take advantage of the enhancement*
- *The improvement gained by the enhanced execution mode; that is, how much faster the task would run if the enhanced mode were used for the entire program*

Execution Time & Speed Up

$$\text{Execution time}_{\text{new}} = \text{Execution time}_{\text{old}} \times \left((1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right)$$

$$\text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

Example

- Suppose that we want to enhance the processor used for Web serving. The new processor is 10 times faster on computation in the Web serving application than the original processor. Assuming that the original processor is busy with computation 40% of the time and is waiting for I/O 60% of the time, what is the overall speedup gained by incorporating the enhancement?

$$\text{Fraction}_{\text{enhanced}} = 0.4,\ \text{Speedup}_{\text{enhanced}} = 10,\ \text{Speedup}_{\text{overall}} = \frac{1}{0.6 + \frac{0.4}{10}} = \frac{1}{0.64} \approx 1.56$$

Practice Example

A common transformation required in graphics processors is square root. Implementations of floating-point (FP) square root vary significantly in performance, especially among processors designed for graphics. Suppose FP square root (FPSQR) is responsible for 20% of the execution time of a critical graphics benchmark. One proposal is to enhance the FPSQR hardware and speed up this operation by a factor of 10. The other alternative is just to try to make all FP instructions in the graphics processor run faster by a factor of 1.6; FP instructions are responsible for half of the execution time for the application. The design team believes that they can make all FP instructions run 1.6 times faster with the same effort as required for the fast square root. Compare these two desi

$$\text{Speedup}_{\text{FPSQR}} = \frac{1}{(1 - 0.2) + \frac{0.2}{10}} = \frac{1}{0.82} = 1.22$$

$$\text{Speedup}_{\text{FP}} = \frac{1}{(1 - 0.5) + \frac{0.5}{1.6}} = \frac{1}{0.8125} = 1.23$$

MIPS as a Performance Metric

- MIPS: Millions of Instructions Per Second
  - Doesn't account for
    - Differences in ISAs between computers
    - Differences in complexity between instructions

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6}$$

$$= \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

  - CPI varies between programs on a given CPU

Example

Consider two different implementations, M1 and M2, of the same instruction set. There are three classes of instructions (A, B, and C) in the instruction set. M1 has a clock rate of 80 MHz and M2 has a clock rate of 100 MHz. The average number of cycles for each instruction class and their frequencies (for a typical program) are as follows:

Instruction Class	Machine M1 – Cycles/Instruction Class	Machine M2 – Cycles/Instruction Class	Frequency
A	1	2	60%
B	2	3	30%
C	4	4	10%

- Calculate the average CPI for each machine, M1, and M2.
- Calculate the average MIPS ratings for each machine, M1 and M2.

For Machine M1:

Average MIPS rating = Clock Rate/(CPI * 106)

= (80 * 106) / (1.6*106)

= 50.0

For Machine M2:

Average MIPS rating = Clock Rate/(CPI * 106)

= (100 * 106) / (2.5*106)

= 40.0

The Problem With MIPS

- The problem with using MIPS as a measure of comparison is threefold:
  - MIPS is dependent on the instruction set, making it difficult to compare MIPS of computers with different instruction sets;
  - MIPS varies between programs on the same computer; and most importantly,
  - MIPS can vary inversely to performance!
- The classic example of the last case is the MIPS rating of a machine with optional floating-point hardware. Machines with the option yield faster executing programs yet have a **lower** MIPS rating.

PRACTICE SESSION

Concluding Remarks

- Cost/performance is improving
  - Due to underlying technology development
- Hierarchical layers of abstraction
  - In both hardware and software
- Instruction set architecture
  - The hardware/software interface
- Execution time: the best performance measure
- Power is a limiting factor
  - Use parallelism to improve performance

Readings from Text Book

- Chapter – 1

Acknowledgements

- Computer Organization and Design The Hardware/Software Interface, David Patternson & John Hennessy, 5th Edition, 2013
- Computer Architecture A Quantitative Approach, John Hennessy & David Patternson, 5th Edition, 2012
- Computer Architecture A Quantitative Approach, John Hennessy & David Patternson, 4th Edition, 2007

Material in these slides has been taken from, the following resources