Printout

Wednesday, March 3, 2021 11:59 AM

Chapter 2
Computer Evolution and Performance – Part 09

Dr. Mostafa Elhosseini

1

Performance Assessment

Dr. Mostafa Elhosseini

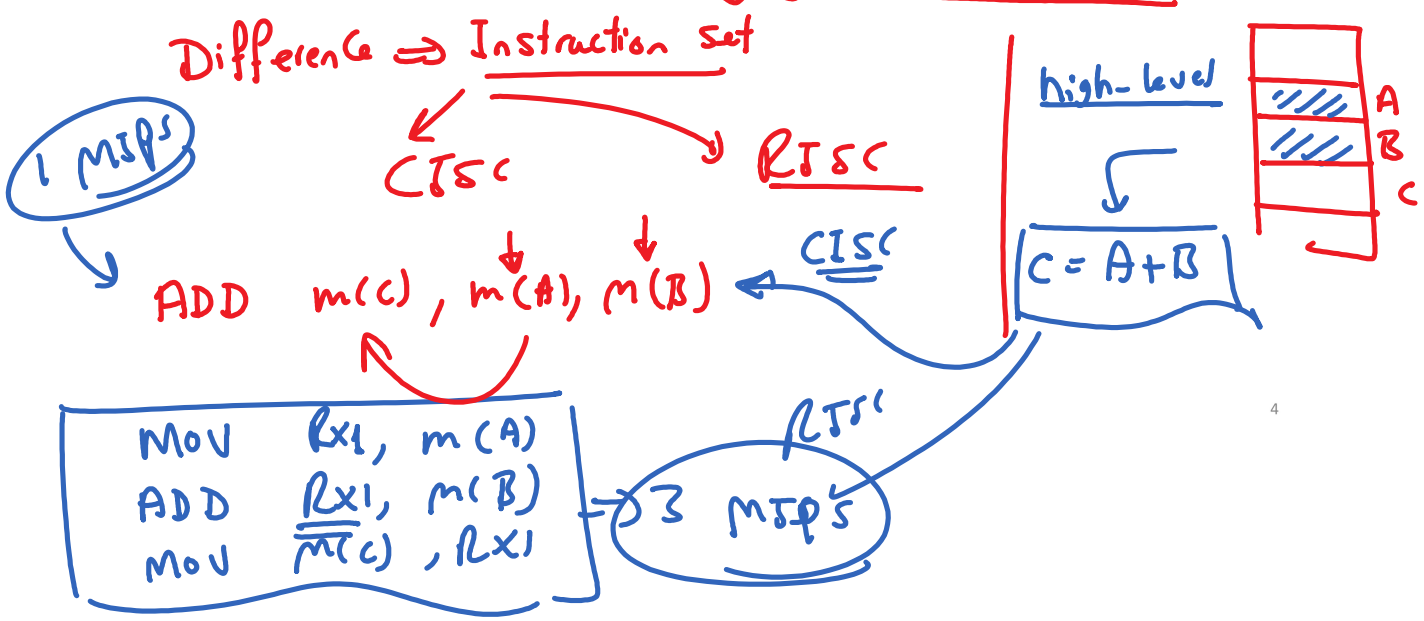
2

Agenda

- Intro
- Benchmarks
  - SPEC Benchmarks
  - Speed Metric
  - Rate Metric
- Amdahl's Law
  - Speedup

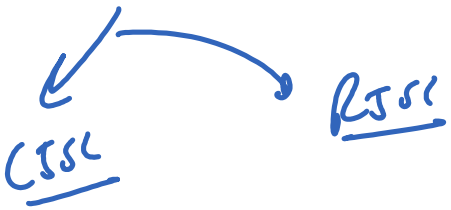
Intro...

- Measures such as MIPS and MFLOPS have proven inadequate to evaluating the performance of processors.



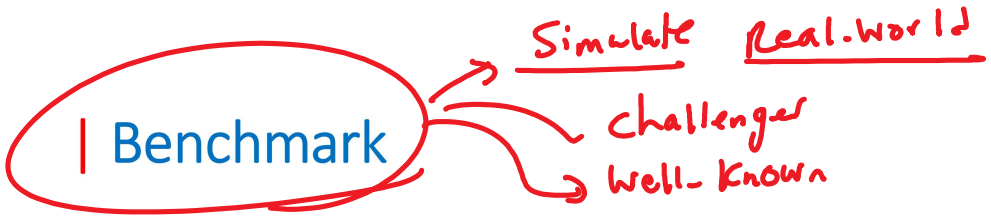
Intro…

- For example, consider this high-level language statement:
  - A = B + C /* assume all quantities in main memory */



Intro…

- Further, the performance of a given processor on a given program may not be useful in determining how that processor will perform on a very different type of application



- beginning in the late 1980s and early 1990s, industry and academic interest shifted to measuring the performance of systems using a set of benchmark programs.
- The same set of programs can be run on different machines and the execution times compared

| Benchmark C/Cs

- Written in a high-level language
- Representative of a particular kind of programming style,
- Measured easily.
- Wide distribution

Portable across different m/c

- System programming
- Commercial
- I/O - floating-point

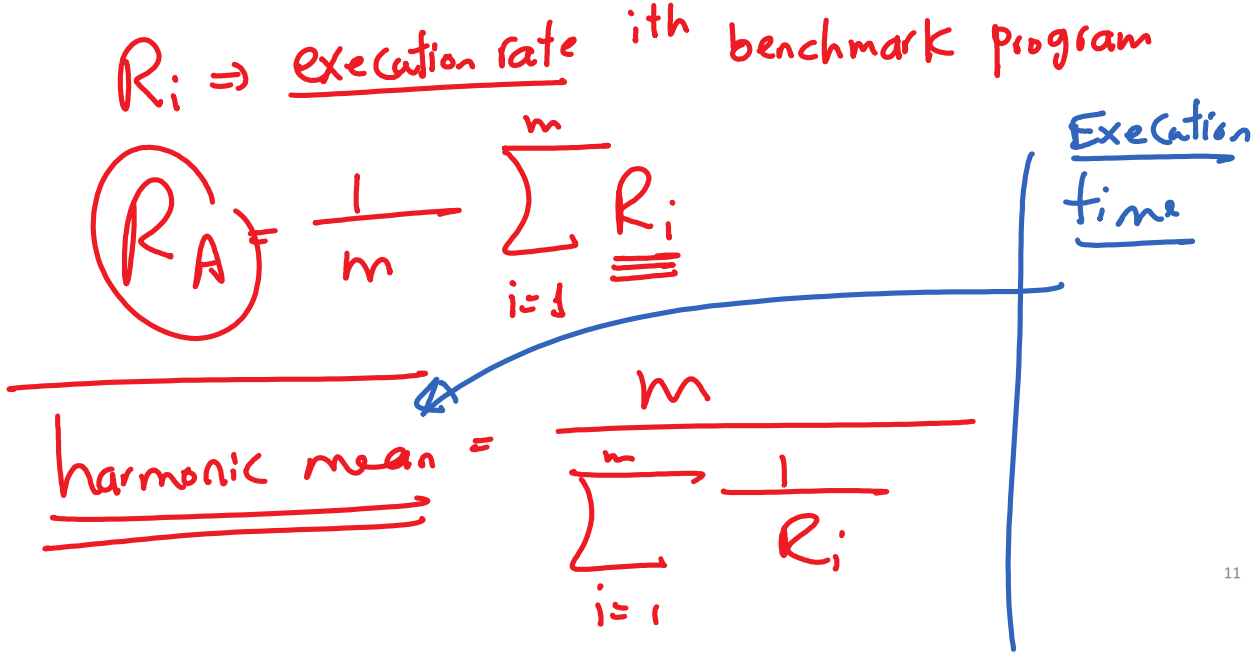
SPEC Benchmarks

- The best known such collection of benchmark suites is defined and maintained by the System Performance Evaluation Corporation (SPEC), an industry consortium.
  - Widely used for comparison and research purposes
  - The best known of the SPEC benchmark suites is SPEC CPU2006.
  - This is the industry standard suite for processor-intensive applications.

SPEC Benchmarks

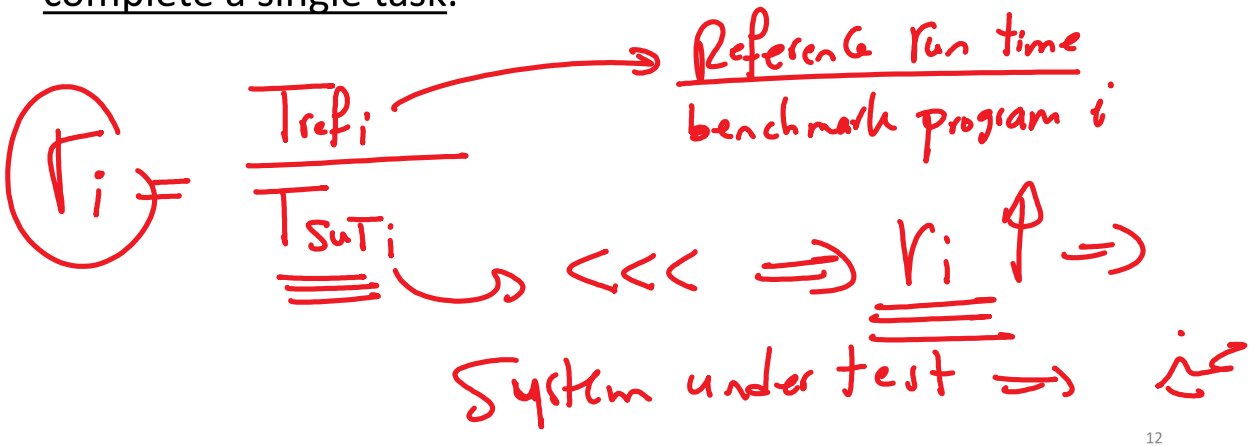
- The CPU2006 suite is based on existing applications that have already been ported to a wide variety of platforms by SPEC industry members.
- It consists of 17 floating-point programs written in C, C++, and Fortran; and 12 integer programs written in C and C++.
- The suite contains over 3 million lines of code. This is the fifth generation of processor-intensive suites from SPEC, replacing SPEC CPU2000, SPEC CPU95, SPEC CPU92, and SPEC CPU89

How does it work?



Speed Metric – Rate Metric

- The speed metric measures the ability of a computer to complete a single task.



| Speed Metric – Rate Metric

$$r_i = \frac{T_{ref\,i}}{T_{SUT\,i}}$$

$$r_G = \left(\prod_{i=1}^{n} r_i \right)^{1/n}$$

| Example

- As an example of the calculation and reporting, consider the Sun Blade 6250, which consists of two chips with four cores, or processors, per chip. One of the SPEC CPU2006 integer benchmark is 464.h264ref. This is a reference implementation of H.264/AVC (Advanced Video Coding), the latest state-of-the-art video compression standard.
- The Sun system executes this program in 934 seconds. The reference implementation requires 22,135 seconds

$$r_i = \frac{22,135}{934} \simeq 23.7 \uparrow \Rightarrow \text{higher Speed}$$

Example

Benchmark	Ratio
400.perlbench	17.5
401.bzip2	14.0
403.gcc	13.7
429.mcf	17.6
445.gobmk	14.7
456.hmmer	18.6

Benchmark	Ratio
458.sjeng	17.0
462.libquantum	31.3
464.h264ref	23.7
471.omnetpp	9.23
473.astar	10.9
483.xalancbmk	14.7

- $(17.5 \times 14 \times 13.7 \times 17.6 \times 14.7 \times 18.6 \times 17 \times 31.3 \times 23.7 \times 9.23 \times 10.9 \times 14.7)^{1/12} = 18.5$

Rate Metric

- The rate metric measures the throughput or rate of a machine carrying out a number of tasks.
- For the rate metrics, multiple copies of the benchmarks are run simultaneously.
- Typically, the number of copies is the same as the number of processors on the machine

$$r_i = \frac{N \times Tref_i}{Tsut_i}$$

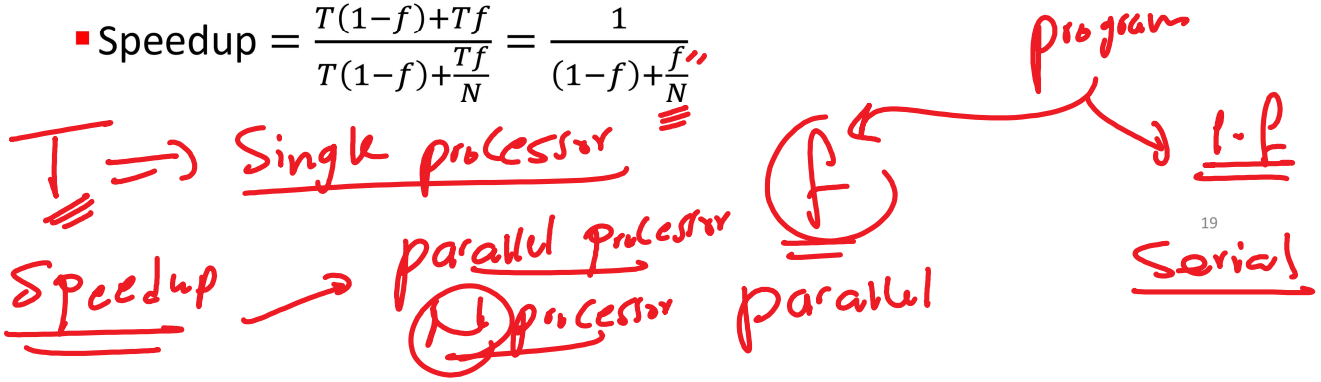
Rate Metric

Amdahl's Law

- speedup in one aspect of the technology or design does not result in a corresponding improvement in performance
- Amdahl's law was first proposed by Gene Amdahl and deals with the potential speedup of a program using multiple processors compared to a single processor

Amdahl's Law

- Let T be the total execution time of the program using a single processor. Then the speedup using a parallel processor with N processors that fully exploits the parallel portion of the program is
- Speedup = $\frac{time\ to\ execute\ program\ on\ a\ single\ processor}{time\ to\ execute\ program\ on\ N\ parallel\ processors}$
- Speedup = $\frac{T(1-f)+Tf}{T(1-f)+\frac{Tf}{N}} = \frac{1}{(1-f)+\frac{f}{N}}$



Amdahl's Law

- Speedup $= \frac{T(1-f)+Tf}{T(1-f)+\frac{Tf}{N}} = \frac{1}{(1-f)+\frac{f}{N}}$

$\frac{1}{1-f}$

Two important conclusions can be drawn:

- When f is small, the use of parallel processors has little effect
- As N approaches infinity, speedup is bound by $1/(1-f)$, so that there are diminishing returns for using more processors

Amdahl's Law

- Amdahl's law can be generalized to evaluate any design or technical improvement in a computer system.
- Consider any enhancement to a feature of a system that results in a speedup. The speedup can be expressed as
- Speedup $= \frac{Performance\ after\ enhancement}{performance\ before\ engancement}$
- Speedup $= \frac{Execution\ time\ before\ enhancement}{Execution\ time\ after\ enhancement}$

Amdahl's Law

- Suppose that a feature of the system is used during execution a fraction of the time f, before enhancement, and that the speedup of that feature after enhancement is SU_f, Then the overall speedup of the system is Speedup $= \frac{1}{(1-f)+\frac{f}{SU_f}}$

$$\frac{1}{(1-f)+\frac{f}{N}}$$

$$\frac{1}{(1-f)+\frac{f}{K}}$$

22

| Amdahl's Law

- For example, suppose that a task makes extensive use of floating-point operations, with 40% of the time is consumed by floating-point operations. With a new hardware design, the floating-point module is speeded up by a factor of K. Then the overall speedup is:

$$\text{Speedup} = \frac{1}{(1-f)+\frac{f}{SU_f}}$$

$$\text{Speedup} = \frac{1}{0.6+\frac{0.4}{K}}$$

Thus independent of K, the maximum speed is 1.67

$$\frac{1}{0.6} : \frac{10}{6} : 1\frac{4}{6} = 1.67$$