Unit 1. Quantitative analysis of computer performance

Computer Architecture

Area of Computer Architecture and Technology
Department of Computer Science and Engineering
University of Oviedo

Fall, 2015

The concept of performance

You want to buy a computer

If your budget is 500€, which computer would you choose?

- **Apple iPad Air 2**
- ARMv8 (1.5GHz)
- 2 GB LPDDR3 RAM
- PowerVR GXA6850 GPU
- 16 GB NAND storage
- 2048×1536 IPS display
- Price: 499€

- **Samsung Galaxy Tab 2**
- Krait 400 (2.3 GHz)
- 3 GB RAM
- Adreno 330 GPU
- 32 GB NAND storage
- 2560×1600 IPS display
- Price: 500€

The concept of performance

You want to buy a computer

If your budget is 500€, which computer would you choose?

- **Apple iPad Air 2**
- ARMv8 (1.5GHz)
- 2 GB LPDDR3 RAM
- PowerVR GXA6850 GPU
- 16 GB NAND storage
- 2048×1536 IPS display
- Price: 499€

- **Samsung Galaxy Tab 2**
- Krait 400 (2.3 GHz)
- 3 GB RAM
- Adreno 330 GPU
- 32 GB NAND storage
- 2560×1600 IPS display
- Price: 500€

The concept of performance

You want to buy a computer

If your budget is 500€, which computer would you choose?

- **Apple iPad Air 2**
- ARMv8 (1.5GHz)
- 2 GB LPDDR3 RAM
- PowerVR GXA6850 GPU
- 16 GB NAND storage
- 2048×1536 IPS display
- Price: 499€

- **Samsung Galaxy Tab 2**
- Krait 400 (2.3 GHz)
- 3 GB RAM
- Adreno 330 GPU
- 32 GB NAND storage
- 2560×1600 IPS display
- Price: 500€

The concept of performance

You want to buy a computer

If your budget is 500€, which computer would you choose?

- **Apple iPad Air 2**
- ARMv8 (1.5GHz)
- 2 GB LPDDR3 RAM
- PowerVR GXA6850 GPU
- 16 GB NAND storage
- 2048×1536 IPS display
- Price: 499€

- **Samsung Galaxy Tab 2**
- Krait 400 (2.3 GHz)
- 3 GB RAM
- Adreno 330 GPU
- 32 GB NAND storage
- 2560×1600 IPS display
- Price: 500€

The concept of performance

You want to buy a computer

If your budget is 500€, which computer would you choose?

- **Apple iPad Air 2**
- ARMv8 (1.5GHz)
- 2 GB LPDDR3 RAM
- PowerVR GXA6850 GPU
- 16 GB NAND storage
- 2048×1536 IPS display
- Price: 499€

- **Samsung Galaxy Tab 2**
- Krait 400 (2.3 GHz)
- 3 GB RAM
- Adreno 330 GPU
- 32 GB NAND storage
- 2560×1600 IPS display
- Price: 500€

The concept of performance

You want to buy a computer

If your budget is 500€, which computer would you choose?

- **Apple iPad Air 2**
- ARMv8 (1.5GHz)
- 2 GB LPDDR3 RAM
- PowerVR GXA6850 GPU
- 16 GB NAND storage
- 2048×1536 IPS display
- Price: 499€

- **Samsung Galaxy Tab 2**
- Krait 400 (2.3 GHz)
- 3 GB RAM
- Adreno 330 GPU
- 32 GB NAND storage
- 2560×1600 IPS display
- Price: 500€

The concept of performance

You want to buy a computer

If your budget is 500€, which computer would you choose?

- **Apple iPad Air 2**
- ARMv8 (1.5GHz)
- 2 GB LPDDR3 RAM
- PowerVR GXA6850 GPU
- 16 GB NAND storage
- 2048×1536 IPS display
- Price: 499€

- **Samsung Galaxy Tab 2**
- Krait 400 (2.3 GHz)
- 3 GB RAM
- Adreno 330 GPU
- 32 GB NAND storage
- 2560×1600 IPS display
- Price: 500€

The concept of performance

There is a need to compare computers

Several criteria can be used

- Price
- Technical service
- Manufacturer's reputation
- Warranty
- Weight
- Appearance
- Performance

Often the performance is related to the cost

- $\uparrow$ cost $\Rightarrow \uparrow$ performance
- However, this relationship is not always true

The concept of performance

How is performance measured?

Example




Vehicle	Passengers	Top speed	Fuel consumption	Price
Ferrari F430	2	320 km/h	18.3 l/100 km	186 000
Audi R8	2	300 km/h	14.2 l/100 km	130 000
Renault Scenic	5	170 km/h	4.5 l/100 km	17 000
Citroën C4	7	180 km/h	10.7 l/100 km	19 000

The concept of performance

How is performance measured?

Example




Vehicle	Passengers	Top speed	Fuel consumption	Price
Ferrari F430	2	320 km/h	18.3 l/100 km	186 000
Audi R8	2	300 km/h	14.2 l/100 km	130 000
Renault Scenic	5	170 km/h	4.5 l/100 km	17 000
Citroën C4	7	180 km/h	10.7 l/100 km	19 000

1. Which is the fastest car?
2. Which car consumes less fuel?
3. Which car provides less cost per passenger and per km?

The concept of performance

How is the performance of a computer measured?

- The performance of a computer is a very broad concept
  - Execution time of a task
  - Number of tasks completed per time unit
  - Power consumption
  - etc.

It is necessary to define performance metrics

Metric

Unit of measure to quantify a particular feature of a system

- A performance metric quantifies the performance of a computer
- A value for the comparison is needed

Performance metrics

Response time, elapsed time or wall-clock time

Time taken to complete a task

- Time between the start and the completion of a program, response to a query on a database, etc.
- Latency to complete a task
- Simile with speed: $\uparrow$ speed $\Rightarrow \downarrow$ response time

$$\text{Performance} = \frac{1}{\text{Response time}}$$

Throughput

Number of tasks completed per time unit

- Number of instructions executed per second, number of queries served by a data base per second, etc.
- In memory systems it is measured as bandwidth
- It can be measured in several ways

Performance metrics

Response time, elapsed time or wall-clock time

Time taken to complete a task

- Time between the start and the completion of a program, response to a query on a database, etc.
- Latency to complete a task
- Simile with speed: $\uparrow$ speed $\Rightarrow \downarrow$ response time

$$\text{Performance} = \frac{1}{\text{Response time}}$$

Throughput

Number of tasks completed per time unit

- Number of instructions executed per second, number of queries served by a data base per second, etc.
- In memory systems it is measured as bandwidth
- It can be measured in several ways

Performance metrics

Example

Given computer A that can complete a task in 20 seconds and computer B that can complete the same task in 30 seconds, what is the difference in performance between both computers? That is, how much faster is computer A than computer B?

$$\frac{\text{Performance}_\text{A}}{\text{Performance}_\text{B}} = \frac{\text{Response time}_\text{B}}{\text{Response time}_\text{A}} = \frac{30}{20} = 1.5$$

Computer A has a *speedup* equal to 1.5 $\Leftrightarrow$ A is 50% faster than B:

$$\frac{\text{Response time}_\text{B} - \text{Response time}_\text{A}}{\text{Response time}_\text{A}} \times 100 = \frac{30 - 20}{20} \times 100 = 50\%$$

Performance metrics

Example

Throughput, ρ, is expressed in tasks per second as follows:

$$\rho_{\mathrm{A}} = \frac{1}{\text{Response time}_{\mathrm{A}}} = \frac{1}{20} = 0.05 \text{ tasks/s}$$

$$\rho_{\mathrm{B}} = \frac{1}{\text{Response time}_{\mathrm{B}}} = \frac{1}{30} = 0.0\widehat{3} \text{ tasks/s}$$

This relationship is not always true

- Only when a computer cannot execute more than one task at a time

Performance metrics

Response time and throughput are not constant values

- They vary among measurements
- They must be approximated by random variables
- These variables are characterized by a mean and a standard deviation
- Several measurements must be taken for significant results to be obtained

Example

The response time in the execution of a program is measured 10 times, $n = 10$:

Measurement	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Response time (s)	3.2	2.9	3.1	3.0	2.8	3.1	3.2	3.0	3.3	2.7

What is the response time of the computer for this program?

Example

1. Compute the average (or mean) response time ($\overline{x}$)

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{30.3}{10} = 3.03\,\mathrm{s}$$

2. Compute the standard deviation (s)

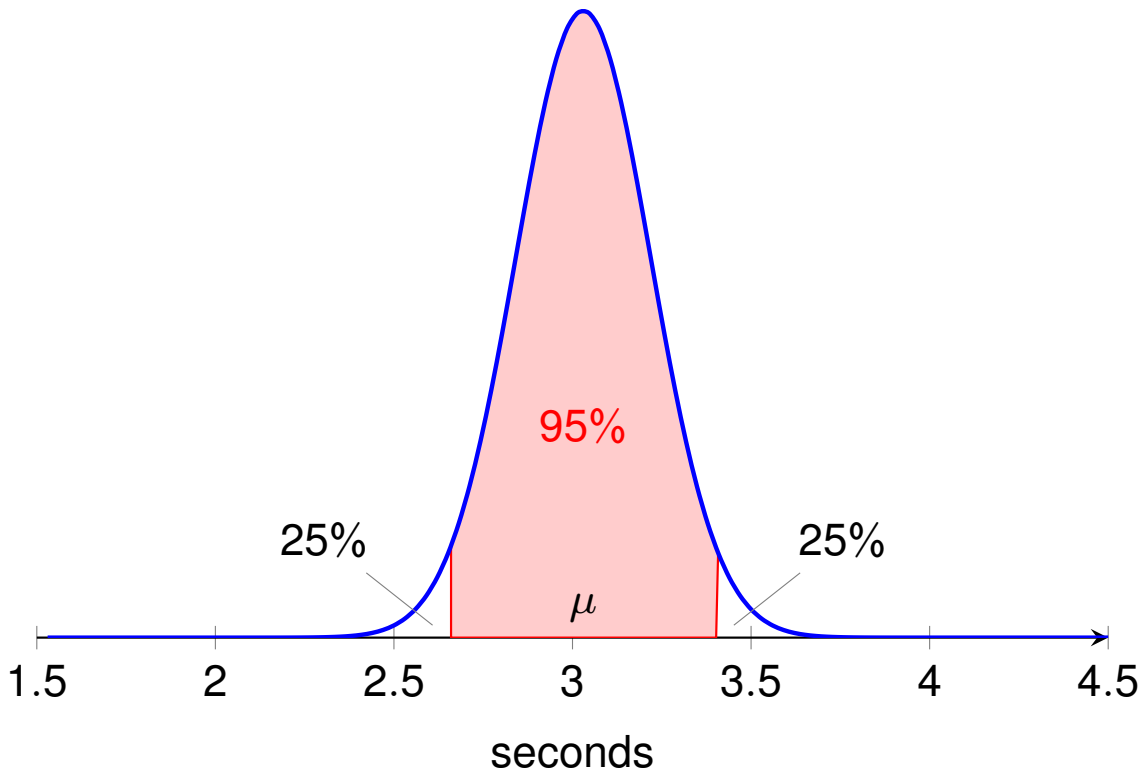
$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}} = 0.19\,\mathrm{s}$$

3. It is assumed that the random variable follows a normal distribution, so we use the sample mean and sample standard deviation (computed above) as estimators of the population mean (μ) and the population standard deviation (σ) respectively

Performance metrics

Response time as a random variable

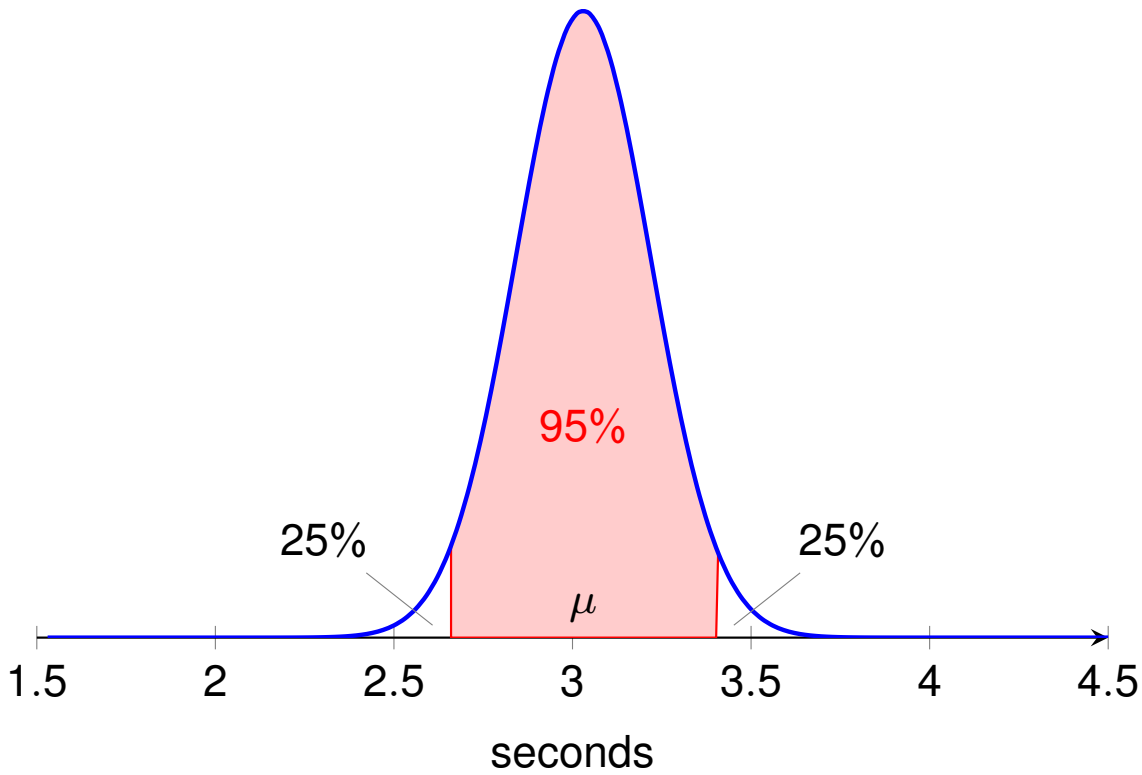
- Mean $\mu \approx \overline{x}$
- Standard deviation $\sigma \approx s \Rightarrow$ response time between $[\mu - 2\sigma, \mu + 2\sigma]$ with 95% likelihood $\Rightarrow [2.62, 3.38]$



Performance metrics

Response time as a random variable

- Mean $\mu \approx \overline{x}$
- Standard deviation $\sigma \approx s \Rightarrow$ response time between $[\mu - 2\sigma, \mu + 2\sigma]$ with 95% likelihood $\Rightarrow [2.62, 3.38]$



Performance improvements

Several types of improvements can be proposed

- Improve the response time and throughput
- Improve the throughput only

Example

Reduce the computing time of the task to 15 s in computer A

- Throughput is also increased $\Rightarrow 0.0\widehat{6}$ tasks/s

Add a second processor to computer A

- The throughput is doubled but the response time is not modified

Performance improvements

Techniques for improving the performance

1. Reduce the response time
   - Organizational improvements of functional components
   - Technological improvements
   - Throughput is also increased
2. Increase the throughput
   - Increase the parallelism
   - Usually, the response time is not reduced, sometimes it is even increased

Amdahl's law

- A computer consists of several components
  - CPU, memory, I/O system and peripheral devices
- Improve computer performance $\Rightarrow$ improve some portion of the computer
- Improving the performance of a component by a factor p, does not imply an enhancement of the performance of the computer by the same factor

Statement

The speedup or gain that can be obtained in the performance of a system due to the fact of improving one of its components is limited by the fraction of the computation time that can take advance of the enhancement

Amdahl's law

The response time of a computer is computed adding

- The fraction of the computation time of the original portion of the computer
- The fraction of the computation time of the enhanced portion of the computer

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

Amdahl's law defines the speedup that can be gained by using a particular feature

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

Amdahl's law

Exercise

Suppose that a computer needs 100 seconds to execute a program: 20 seconds of CPU execution, 35 of memory accesses, 40 due to input/output operations in the hard disk, and the rest due to operations in the interconnection mechanisms of the computer

1. Which is the speedup in the execution of the program if the CPU is replaced by another one three times faster?

2. Which is the speedup if the hard disk is replaced by another one two times faster?

3. Which are the new response times?

Amdahl's law

Exercise

Suppose that a computer needs 100 seconds to execute a program: 20 seconds of CPU execution, 35 of memory accesses, 40 due to input/output operations in the hard disk, and the rest due to operations in the interconnection mechanisms of the computer

1. Which is the speedup in the execution of the program if the CPU is replaced by another one three times faster?

$$\text{Fraction}_{\text{CPU}} = \frac{20}{100} = 0.2 \qquad \textit{Speedup} = \frac{1}{(1 - 0.2) + \frac{0.2}{3})} \approx$$

$$1.15 \Rightarrow 15\%$$

2. Which is the speedup if the hard disk is replaced by another one two times faster?

3. Which are the new response times?

Amdahl's law

Exercise

Suppose that a computer needs 100 seconds to execute a program: 20 seconds of CPU execution, 35 of memory accesses, 40 due to input/output operations in the hard disk, and the rest due to operations in the interconnection mechanisms of the computer

1. Which is the speedup in the execution of the program if the CPU is replaced by another one three times faster?

$$\text{Fraction}_{\text{CPU}} = \frac{20}{100} = 0.2 \qquad \textit{Speedup} = \frac{1}{(1 - 0.2) + \frac{0.2}{3})} \approx$$

$$1.15 \Rightarrow 15\%$$

2. Which is the speedup if the hard disk is replaced by another one two times faster? $\text{Fraction}_{\text{HD}} = \frac{40}{100} = 0.4$ $A = \frac{1}{(1 - 0.4) + \frac{0.4}{2})} = 1.25 \Rightarrow 25\%$

3. Which are the new response times?

Amdahl's law

Exercise

Suppose that a computer needs 100 seconds to execute a program: 20 seconds of CPU execution, 35 of memory accesses, 40 due to input/output operations in the hard disk, and the rest due to operations in the interconnection mechanisms of the computer

1. Which is the speedup in the execution of the program if the CPU is replaced by another one three times faster?

$$\text{Fraction}_{\text{CPU}} = \frac{20}{100} = 0.2 \qquad \textit{Speedup} = \frac{1}{(1 - 0.2) + \frac{0.2}{3})} \approx$$

$$1.15 \Rightarrow 15\%$$

2. Which is the speedup if the hard disk is replaced by another one two times faster? $\text{Fraction}_{\text{HD}} = \frac{40}{100} = 0.4$ $A = \frac{1}{(1 - 0.4) + \frac{0.4}{2})} = 1.25 \Rightarrow 25\%$

3. Which are the new response times? $\text{Response time}_{\text{new}} = \frac{\text{Response time}_{\text{old}}}{\textit{Speedup}}$

$$\text{Response time}_{\text{CPU enhanced}} = \frac{100}{1.15} = 86.96 \text{ s} \qquad \text{Response time}_{\text{HD enhanced}} = \frac{100}{1.25} = 80 \text{ s}$$

CPU performance

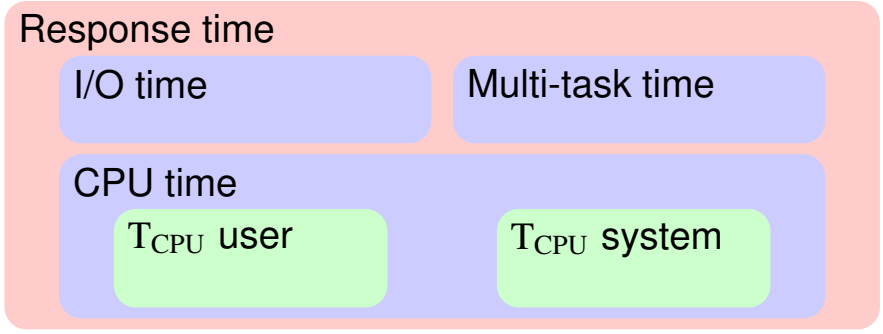
How to measure the real performance of a CPU

The above metrics can be used: response time and throughput

- ✔ Easy to understand, intuitive
- ✘ Difficult to compare different CPUs

Response time

- It depends on several factors
- It varies a lot in multitasking operating systems



Is MIPS useful (million instructions per second)?

It is not useful while comparing CPU with different instruction sets

There is a need for extrapolating metrics

CPU time

Response time

- The time between the start and the completion o a task
- Includes
  - CPU time
  - Input/Output time
  - Execution of other tasks
  - etc.

CPU time

- ✔ User time: time invested in executing the task code
- System time: time invested in executing the operating system while executing the task

CPU time

In UNIX there exists a command for retrieving the response time of a task

```
$>  time ./my-task

real 0m2.822s
user 0m2.760s
sys 0m0.008s
```

`real` Total elapsed time, that is, response time

`user` CPU user time

`sys` CPU system time

- Total CPU time: $2.76 + 0.008 = 2.768 \Rightarrow \frac{2.768}{2.822} \times 100 = 98.09\%$
- This is a CPU-intensive task

CPU time

The CPU time of a program (monothread) is computed based on the clock frequency, f, or on the clock cycle time, T:

$$\begin{aligned} \mathrm{T_{CPU}} &= \frac{\text{CPU clock cycles for a program}}{f} \\ &= \text{CPU clock cycles for a program} \times T \end{aligned}$$

CPI (clock cycles per instruction)

Clock cycles for completing each instruction of the program

$$\mathrm{CPI} = \frac{\text{CPU clock cycles for a program}}{\text{\# of program instructions}}$$

Substituting...

$$\mathrm{T_{CPU}} = \frac{\text{\# of program instructions} \times \mathrm{CPI}}{f} = \frac{\text{Instructions}}{\text{Program}} \times \mathrm{CPI} \times T$$

CPU time: Iron Law

$$T_{CPU} = \frac{\text{Instructions}}{\text{Program}} \times \text{CPI} \times T$$

CPU time depends on

1. # of instructions per program
   - Abstraction level of the programming language
   - Compiler technology
   - Code optimization
2. Clock rate
   - Indicates the speed of the CPU
   - It depends on the processor manufacturing technology (40 *nm*, 32 *nm*)
   - Frequency is usually increased improving the manufacturing technology
   - ✘ Heat dissipation (and power consumption)
   - It can also be achieved with organizational improvements (*pipeline*)
3. Clock cycles per instruction (CPI)...

CPU time: Iron Law

$$T_{CPU} = \frac{\text{Instructions}}{\text{Program}} \times \text{CPI} \times T$$

CPU time depends on

1. # of instructions per program
   - Abstraction level of the programming language
   - Compiler technology
   - Code optimization
2. Clock rate
   - Indicates the speed of the CPU
   - It depends on the processor manufacturing technology (40 *nm*, 32 *nm*)
   - Frequency is usually increased improving the manufacturing technology
   - ✘ Heat dissipation (and power consumption)
   - It can also be achieved with organizational improvements (*pipeline*)
3. Clock cycles per instruction (CPI)...

CPU time: Iron Law

$$T_{CPU} = \frac{\text{Instructions}}{\text{Program}} \times \text{CPI} \times T$$

CPU time depends on

1. # of instructions per program
   - Abstraction level of the programming language
   - Compiler technology
   - Code optimization
2. Clock rate
   - Indicates the speed of the CPU
   - It depends on the processor manufacturing technology (40 *nm*, 32 *nm*)
   - Frequency is usually increased improving the manufacturing technology
   - ✘ Heat dissipation (and power consumption)
   - It can also be achieved with organizational improvements (*pipeline*)
3. Clock cycles per instruction (CPI)...

Clock cycles per instruction

CPI value depends on instruction complexity

- CISC: complex instruction set
  - ✔ Compact programs, with a small number of instructions
  - ✘ CPI increases
- RISC: reduced instruction set computer
  - ✔ Low CPI
  - ✘ More instructions are needed in the program

Current tendency

Follow the principle stated in Amdahl's law when designing instruction sets

- Frequently used instructions $\Rightarrow$ Low CPI
- Less used instructions carrying out complex tasks $\Rightarrow$ High CPI. Their weight in final performance will be less
- Modern CPUs follow a hybrid RISC/CISC model

Benchmarks

The computer performance cannot be computed, but measured

- Measurement results vary from measure to measure
- Depends on computer conditions: workload

Benchmark

A program, or a set of programs, used to assess the performance of a computer. It can also be oriented to evaluate a portion of the computer: CPU, memory, hard disk, GPU, etc.

- A portion of the computer is subjected to a workload
- The results vary from benchmark to benchmark

Workload

It must represent the real work of the computer

- Desktop computer $\Rightarrow$ office tasks
- Workstation $\Rightarrow$ technical or scientific applications (compilers, video editors, etc.)
- Performance varies according to the workload

Benchmarks

Types of workload

- Real workload: real applications run in the computer
  - ✔ Performance evaluated under realistic work conditions
  - ✘ They are difficult to reproduce
  - ✘ High variability $\Rightarrow$ need of a high number of repetitions for getting statistically significant results
- Synthetic workload: fake programs invented to try to match the profile and behaviour of real applications
  - ✔ Easy to reproduce
  - ✔ They are not real applications
  - ✘ Sometimes their results are not correlated with the real observed performance
- Analytic workload: mathematical models to predict the performance of the computer
  - ✔ Useful when the computer is not available (it does not exist or it has not been acquired yet) or when the computer cannot run tests due to its operation
  - ✘ Low correlation with observed performance

Unit 1. Quantitative analysis of computer performance

Computer Architecture

Area of Computer Architecture and Technology
Department of Computer Science and Engineering
University of Oviedo

Fall, 2015