Performance (I)

Hung-Wei Tseng
Recap: von Neumann Architecture

By loading different programs into memory, your computer can perform different functions.
Recap: The “abstraction”
Recap: Popular ISAs

- x86
- ARM
- MIPS
- RISC-V
- Apple A13 Bionic
- NVIDIA Tegra X1
- Qualcomm Snapdragon
- Swerv Core
- AMD Ryzen
- Intel Core i7
## Recap: MIPS v.s. x86

<table>
<thead>
<tr>
<th></th>
<th>MIPS</th>
<th>x86</th>
</tr>
</thead>
<tbody>
<tr>
<td>instruction width</td>
<td>32 bits</td>
<td>1 ~ 17 bytes</td>
</tr>
<tr>
<td>code size</td>
<td>larger</td>
<td>smaller</td>
</tr>
<tr>
<td>registers</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>addressing modes</td>
<td>reg+offset</td>
<td>base+offset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>base+index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>scaled+index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>scaled+index+offset</td>
</tr>
<tr>
<td>hardware</td>
<td>simple</td>
<td>complex</td>
</tr>
<tr>
<td>ISA type</td>
<td>Reduced Instruction Set Computers (RISC)</td>
<td>Complex Instruction Set Computers (CISC)</td>
</tr>
</tbody>
</table>
Recap: How many operations: CISC v.s. RISC

- **CISC (Complex Instruction Set Computing)**
  - Examples: x86, Motorola 68K
  - Provide many powerful/complex instructions
    - Many: more than 1503 instructions since 2016
    - Powerful/complex: an instruction can perform both ALU and memory operations

- **RISC (Reduced Instruction Set Computer)**
  - Examples: ARMv8, RISC-V, MIPS (the first RISC instruction, invented by the authors of our textbook)
  - Each instruction only performs simple tasks
  - Easy to decode
Outline

• Definition of “Performance”
• What affects each factor in “Performance Equation”
• Instruction Set Architecture & Performance
Definition of “Performance”
What is performance?

Dictionary

Search for a word

**performance**

/ˌpərˈfôrnəns/

Filter definitions by topic

All  Mechanics  Finance  Linguistics

**noun**

1. an act of staging or presenting a play, concert, or other form of entertainment.
   "Don Giovanni had its first performance in 1787"

   Similar:  show  production  showing  presentation  entertainment

2. the action or process of carrying out or accomplishing an action, task, or function.
   "the continual performance of a single task reduces a man to the level of a machine"

   Similar:  carrying out  execution  discharge  conducting  conduct

Translations, word origin, and more definitions

Definitions from Oxford Languages

Feedback
What do you want for a computer?
What do you want for a computer?

- Latency/Execution time
- Frame rate
- Responsiveness
- Real-time
- Throughput
- Cost
- Volume
- Weight
- Battery life
- Low power/low temperature
- Reliability

The most direct measurement of performance
• The simplest kind of performance
• Shorter execution time means better performance
• Usually measured in seconds

### Execution Time

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>120007a30</td>
<td>load address</td>
<td>gp, 15(t12)</td>
</tr>
<tr>
<td>120007a34</td>
<td>load address</td>
<td>gp, -25520(gp)</td>
</tr>
<tr>
<td>120007a38</td>
<td>load address</td>
<td>t1, 0(gp)</td>
</tr>
<tr>
<td>120007a3c</td>
<td>load address</td>
<td>t4, 0(gp)</td>
</tr>
<tr>
<td>120007a40</td>
<td>load data</td>
<td>t0, -23508(t1)</td>
</tr>
<tr>
<td>120007a44</td>
<td>branch</td>
<td>t0, 120007a94</td>
</tr>
<tr>
<td>120007a48</td>
<td>load address</td>
<td>t0, 0(gp)</td>
</tr>
<tr>
<td>120007a4c</td>
<td>store</td>
<td>zero, -23508(t1)</td>
</tr>
<tr>
<td>120007a50</td>
<td>load data</td>
<td>v0</td>
</tr>
<tr>
<td>120007a54</td>
<td>store</td>
<td>zero, -23512(t4)</td>
</tr>
<tr>
<td>120007a58</td>
<td>load data</td>
<td>t0, -23520(t0)</td>
</tr>
<tr>
<td>120007a5c</td>
<td>branch</td>
<td>t0, 120007a98</td>
</tr>
<tr>
<td>120007a60</td>
<td>move</td>
<td>t0, t1</td>
</tr>
<tr>
<td>120007a64</td>
<td>clear</td>
<td>t2</td>
</tr>
<tr>
<td>120007a68</td>
<td>branch</td>
<td>120007a80</td>
</tr>
</tbody>
</table>

**Instructions**

- **Processor**: CPU component that executes instructions.
- **PC**: Program Counter, keeps track of the next instruction to execute.

**How many of these?**

**How long is it take to execution each of these?**

**Cycles**

- **Instruction**: Number of instructions executed.
- **Seconds**: Time taken to execute the instructions.

**Seconds per Instruction**

- **Cycles × Seconds**
- **Instruction**

**Instruction Memory**

- Contains instructions that the processor will execute.
- Addressed by the PC.

**How to measure execution time?**

1. **Count Instructions**: Count the number of instructions executed.
2. **Time Measurement**: Measure the time taken to execute these instructions.
3. **Calculate Execution Time**: Divide the total time by the number of instructions.

**Execution Time Calculation**

- **Execution Time (in seconds)**: Execution Time = Cycles × Seconds / Number of Instructions
CPU Performance Equation

Performance = \frac{1}{\text{Execution Time}}

\text{Execution Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}

ET = IC \times CPI \times CT

1\text{GHz} = 10^9\text{Hz} = \frac{1}{10^9}\text{sec per cycle} = 1\text{ ns per cycle}

Frequency(i.e.,\text{clock rate})
Performance Example

• Assume that we have an application composed with a total of 500000 instructions, in which 20% of them are the load/store instructions with an average **CPI of 6 cycles**, and **the rest** instructions are integer instructions with average **CPI of 1 cycle**. If the processor runs at 2 GHz, how long is the execution time?

  A. 500000 ns
  B. 1000000 ns
  C. 2000000 ns
  D. 3500000 ns
  E. None of the above
Assume that we have an application composed with a total of 500000 instructions, in which 20% of them are the load/store instructions with an average CPI of 6 cycles, and the rest instructions are integer instructions with average CPI of 1 cycle. If the processor runs at 2 GHz, how long is the execution time?

A. 500000 ns
B. 1000000 ns
C. 2000000 ns
D. 3500000 ns
E. None of the above
Assume that we have an application composed with a total of 500000 instructions, in which 20% of them are the load/store instructions with an average CPI of 6 cycles, and the rest instructions are integer instructions with average CPI of 1 cycle. If the processor runs at 2 GHz, how long is the execution time?

\[ ET = (5 \times 10^5) \times (20\% \times 6 + 80\% \times 1) \times \frac{1}{2 \times 10^{-9}} \text{sec} = 5 \times 10^5 \text{ns} \]

A. 500000 ns
B. 1000000 ns
C. 2000000 ns
D. 3500000 ns
E. None of the above
Assume that we have an application composed with a total of 5000000000 instructions, in which 20% of them are “Type-A” instructions with an average CPI of 8 cycles, 20% of them are “Type-B” instructions with an average CPI of 4 cycles and the rest instructions are “Type-C” instructions with average CPI of 1 cycle. If the processor runs at 3 GHz, how long is the execution time?

A. 3.67 sec
B. 5 sec
C. 6.67 sec
D. 15 sec
E. 45 sec
Assume that we have an application composed with a total of 5000000000 instructions, in which 20% of them are “Type-A” instructions with an average CPI of 8 cycles, 20% of them are “Type-B” instructions with an average CPI of 4 cycles and the rest instructions are “Type-C” instructions with average CPI of 1 cycle. If the processor runs at 3 GHz, how long is the execution time?

A. 3.67 sec  
B. 5 sec  
C. 6.67 sec  
D. 15 sec  
E. 45 sec
Assume that we have an application composed with a total of 5000000000 instructions, in which 20% of them are “Type-A” instructions with an average CPI of 8 cycles, 20% of them are “Type-B” instructions with an average CPI of 4 cycles and the rest instructions are “Type-C” instructions with average CPI of 1 cycle. If the processor runs at 3 GHz, how long is the execution time?

A. 3.67 sec
B. 5 sec
C. 6.67 sec
D. 15 sec
E. 45 sec

\[
ET = (5 \times 10^9) \times (20\% \times 8 + 20\% \times 4 + 60\% \times 1) \times \frac{1}{3 \times 10^{-9}} \text{sec} = 5
\]

\[
ET = IC \times CPI \times CT
\]
Consider the same program on the following two machines, X and Y. By how much Y is faster than X?

<table>
<thead>
<tr>
<th></th>
<th>Clock Rate</th>
<th>Instructions</th>
<th>Percentage of Type-A</th>
<th>CPI of Type-A</th>
<th>Percentage of Type-B</th>
<th>CPI of Type-B</th>
<th>Percentage of Type-C</th>
<th>CPI of Type-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine X</td>
<td>3 GHz</td>
<td>500000</td>
<td>20%</td>
<td>8</td>
<td>20%</td>
<td>4</td>
<td>60%</td>
<td>1</td>
</tr>
<tr>
<td>Machine Y</td>
<td>5 GHz</td>
<td>500000</td>
<td>20%</td>
<td>13</td>
<td>20%</td>
<td>4</td>
<td>60%</td>
<td>1</td>
</tr>
</tbody>
</table>

A. 0.2  
B. 0.25  
C. 0.8  
D. 1.25  
E. No changes
Consider the same program on the following two machines, X and Y. By how much Y is faster than X?

<table>
<thead>
<tr>
<th></th>
<th>Clock Rate</th>
<th>Instruction s</th>
<th>Percentage of Type-A</th>
<th>CPI of Type-A</th>
<th>Percentage of Type-B</th>
<th>CPI of Type-B</th>
<th>Percentage of Type-C</th>
<th>CPI of Type-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine X</td>
<td>3 GHz</td>
<td>500000</td>
<td>20%</td>
<td>8</td>
<td>20%</td>
<td>4</td>
<td>60%</td>
<td>1</td>
</tr>
<tr>
<td>Machine Y</td>
<td>5 GHz</td>
<td>500000</td>
<td>20%</td>
<td>13</td>
<td>20%</td>
<td>4</td>
<td>60%</td>
<td>1</td>
</tr>
</tbody>
</table>

A. 0.2  
B. 0.25  
C. 0.8  
D. 1.25  
E. No changes
Speedup

• The relative performance between two machines, X and Y. Y is $n$ times faster than X

\[ n = \frac{\text{Execution Time}_X}{\text{Execution Time}_Y} \]

• The speedup of Y over X

\[ \text{Speedup} = \frac{\text{Execution Time}_X}{\text{Execution Time}_Y} \]
**Speedup of Y over X**

- Consider the same program on the following two machines, X and Y. By how much Y is faster than X?

<table>
<thead>
<tr>
<th></th>
<th>Clock Rate</th>
<th>Instructions</th>
<th>Percentage of Type-A Insts.</th>
<th>CPI of Type-A Insts.</th>
<th>Percentage of Type-B Insts.</th>
<th>CPI of Type-B Insts.</th>
<th>Percentage of Type-C Insts.</th>
<th>CPI of Type-C Insts.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Machine X</strong></td>
<td>3 GHz</td>
<td>500000</td>
<td>20%</td>
<td>8</td>
<td>20%</td>
<td>4</td>
<td>60%</td>
<td>1</td>
</tr>
<tr>
<td><strong>Machine Y</strong></td>
<td>5 GHz</td>
<td>500000</td>
<td>20%</td>
<td>13</td>
<td>20%</td>
<td>4</td>
<td>60%</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ ET_Y = (5 \times 10^9) \times (20\% \times 13 + 20\% \times 4 + 60\% \times 1) \times \frac{1}{5 \times 10^{-9}} \text{sec} = 4 \]

\[ \text{Speedup} = \frac{\text{Execution Time}_X}{\text{Execution Time}_Y} \]

A. 0.2
B. 0.25
C. 0.8
D. 1.25
E. No changes

\[ = \frac{5}{4} = 1.25 \]
What’s the limiting factor?
Identify the limiting factor

Why does an Intel Core i7 @ 3.5 GHz usually perform better than an Intel Core i5 @ 3.5 GHz or AMD FX-8350@4GHz?

- Because the instruction count of the program are different
- Because the clock rate of AMD FX is higher
- Because the CPI of Core i7 is better
- Because the clock rate of AMD FX is higher and CPI of Core i7 is better
- None of the above

Sysbench 2014 from http://www.anandtech.com/
**Identify the limiting factor**

- Why does an Intel Core i7 @ 3.5 GHz usually perform better than an Intel Core i5 @ 3.5 GHz or AMD FX-8350@4GHz?

<table>
<thead>
<tr>
<th>Processor</th>
<th>Model</th>
<th>Cores</th>
<th>Clock Rate</th>
<th>L3 Cache</th>
<th>CPI</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Core i7</td>
<td>4770K (54W)</td>
<td>4/8T</td>
<td>3.5 GHz</td>
<td>1MB L3, 8MB L2</td>
<td></td>
<td>2382</td>
</tr>
<tr>
<td>Intel Core i5</td>
<td>4690K (85W)</td>
<td>4/4T</td>
<td>3.5 GHz</td>
<td>1MB L3, 6MB L3</td>
<td></td>
<td>2234</td>
</tr>
<tr>
<td>AMD FX-8350</td>
<td>8350 (25W)</td>
<td>4/8T</td>
<td>4.0 GHz</td>
<td>8MB L2, 3MB L3</td>
<td></td>
<td>1889</td>
</tr>
</tbody>
</table>


A. Because the instruction count of the program are different  
B. Because the clock rate of AMD FX is higher  
C. Because the CPI of Core i7 is better  
D. Because the clock rate of AMD FX is higher and CPI of Core i7 is better  
E. None of the above
Identify the limiting factor

Why does an Intel Core i7 @ 3.5 GHz usually perform better than an Intel Core i5 @ 3.5 GHz or AMD FX-8350@4GHz?

A. Because the instruction count of the program are different
B. Because the clock rate of AMD FX is higher
C. Because the CPI of Core i7 is better
D. Because the clock rate of AMD FX is higher and CPI of Core i7 is better
E. None of the above
Identify the limiting factor

Why does an Intel Core i7 @ 3.5 GHz usually perform better than an Intel Core i5 @ 3.5 GHz or AMD FX-8350@4GHz?

A. Because the instruction count of the program are different
B. Because the clock rate of AMD FX is higher
C. Because the CPI of Core i7 is better
D. Because the clock rate of AMD FX is higher and CPI of Core i7 is better
E. None of the above

Every time when the question ask you about the “performance”, thinking about the performance equation first!
What Affects Each Factor in Performance Equation
How programmer affects performance?

- Performance equation consists of the following three factors:
  ① IC
  ② CPI
  ③ CT

How many can a programmer affect?

A. 0
B. 1
C. 2
D. 3
How programmer affects performance?

- Performance equation consists of the following three factors
  ① IC
  ② CPI
  ③ CT

How many can a programmer affect?

A. 0
B. 1
C. 2
D. 3
• By adding the “sort” in the following code snippet, what the programmer changes in the performance equation to achieve better performance?

```cpp
std::sort(data, data + arraySize);

for (unsigned c = 0; c < arraySize*1000; ++c) {
    if (data[c%arraySize] >= INT_MAX/2)
        sum ++;
}
```

A. CPI  
B. IC  
C. CT  
D. IC & CPI
By adding the “sort” in the following code snippet, what the programmer changes in the performance equation to achieve better performance?

```cpp
std::sort(data, data + arraySize);
for (unsigned c = 0; c < arraySize*1000; ++c) {
    if (data[c%arraySize] >= INT_MAX/2)
        sum ++;
}
```

A. CPI
B. IC
C. CT
D. IC & CPI
Programmer’s impact

By adding the “sort” in the following code snippet, what the programmer changes in the performance equation to achieve **better** performance?

```cpp
std::sort(data, data + arraySize);

for (unsigned c = 0; c < arraySize*1000; ++c) {
    if (data[c%arraySize] >= INT_MAX/2)
        sum ++;
}
```

A. CPI

B. IC — we increased IC, suppose to make the

C. CT performance worse

D. IC & CPI
Announcements

• Assignment #1 is up!
  • Due next Monday before the lecture
  • Check the website for the template
  • Submit a pdf through Canvas
  • TA’s discussion today @ 5pm will give you hints on these questions — use the lecture link

• Reading quizzes
  • Due next Tuesday

• Resources
  • Ask questions — piazza
  • Reading quizzes, turning in assignments — Canvas
  • Slides, schedule, assignment questions — Check our website
  • Video archive — Prof. Usagi’s Youtube channel
Computer Science & Engineering