



KIẾN TRÚC MÁY TÍNH

KHOA HỌC & KỸ THUẬT MÁY TÍNH



Võ Tấn Phương

<http://www.cse.hcmut.edu.vn/~vtphuong>

Chương 2

Hiệu suất

Hiệu suất ???

- ❖ Chúng ta lựa chọn máy tính theo tiêu chí gì?
- ❖ Tại sao máy tính chạy chương trình này tốt nhưng không tốt đối với chương trình khác?
- ❖ Làm sao để đo hiệu suất của một máy tính?
- ❖ Phần cứng và phần mềm ảnh hưởng như thế nào?
- ❖ Tập lệnh của máy tính ảnh hưởng như thế nào?
- ❖ Hiểu hiệu suất sẽ biết được động cơ của việc cải tiến sẽ tập trung vào công việc gì

Thời gian đáp ứng & Thông lượng

❖ Thời gian đáp ứng (response time)

- ✧ Khoản thời gian từ lúc bắt đầu đến khi kết thúc một công việc
- ✧ $\text{Response Time} = \text{CPU Time} + \text{Waiting Time (I/O, OS scheduling, etc.)}$

❖ Thông lượng (throughput)

- ✧ Số lượng công việc giải quyết trong một khoảng thời gian

❖ Giảm thời gian thực thi sẽ cải thiện thông lượng

- ✧ Ví dụ: sử dụng bộ xử lý nhanh hơn
- ✧ Thời gian thực hiện một công việc ít \Rightarrow nhiều công việc được thực hiện

❖ Tăng thông lượng cũng giảm thời gian đáp ứng

- ✧ Ví dụ: Tăng số lượng nhân của bộ xử lý
- ✧ Nhiều công việc thực thi song song
- ✧ Thời gian thực thi của một công việc (CPU Time) không thay đổi
- ✧ Thời gian chờ trong hàng đợi định thời giảm (OS scheduling)

Định nghĩa hiệu suất (performance)

❖ Một chương trình A chạy trên máy tính X

$$\text{Performance}_X = \frac{1}{\text{Execution time}_X}$$

❖ X nhanh hơn Y *n* lần (cùng chạy chương trình A)

$$\frac{\text{Performance}_X}{\text{Performance}_Y} = \frac{\text{Execution time}_Y}{\text{Execution time}_X} = n$$

Thời gian thực thi “Execution Time”?

❖ Thời gian trôi qua (elapsed time)

✧ Tính tất cả:

- CPU time, Waiting time, Input/output, disk access, OS scheduling, ... etc.

✧ Con số hữu ích, nhưng không phù hợp cho mục đích đánh giá CPU

❖ Execution Time ~ CPU Execution Time

✧ Chỉ tính thời gian thực thi các lệnh của chương trình

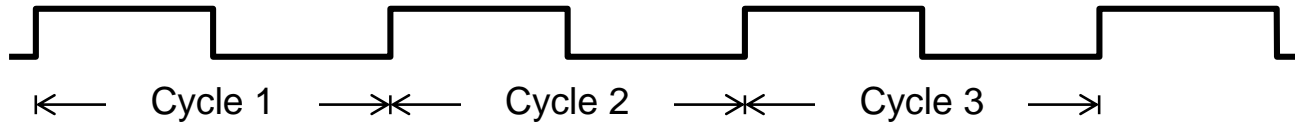
✧ Không tính: thời gian chờ I/O hoặc OS scheduling

✧ Được tính bằng “giây”, hoặc

✧ Thể hiện thông qua số lượng chu kỳ xung nhịp khi CPU thực thi chương trình A (CPU clock cycles)

Số chu kỳ xung nhịp (Clock Cycles)

❖ Clock cycle = Clock period = $1 / \text{Clock rate}$



❖ Clock rate = Clock frequency = Cycles per second

✧ 1 Hz = 1 cycle/sec 1 KHz = 10^3 cycles/sec

✧ 1 MHz = 10^6 cycles/sec 1 GHz = 10^9 cycles/sec

✧ 2 GHz clock has a cycle time = $1/(2 \times 10^9) = 0.5$ nanosecond (ns)

❖ Dùng số chu kỳ xung nhịp thể hiện CPU execution time

$$\text{CPU Execution Time} = \text{Clock cycles} \times \text{cycle time} = \frac{\text{Clock cycles}}{\text{Clock rate}}$$

Cải tiến hiệu suất

❖ Để cải tiến hiệu suất:

- ✧ Giảm số chu kỳ xung nhịp cần thiết để thực thi một chương trình, hoặc
- ✧ Giảm thời gian một chu kỳ (tăng tần số xung nhịp)

❖ Ví dụ:

- ✧ Chương trình A chạy mất 10 giây trên máy tính X với tần số 2 GHz
- ✧ **Số lượng chu kỳ xung nhịp để chạy chương trình A trên máy X ?**
- ✧ Yêu cầu thiết kế máy tính Y chạy cùng chương trình A trong 6 giây
- ✧ Máy tính Y cần thêm 10% số chu kỳ xung nhịp
- ✧ **Hỏi máy tính Y cần xung nhịp có tần số bao nhiêu ?**

❖ Lời giải:

- ✧ Clock cycles trên máy X = $10 \text{ sec} \times 2 \times 10^9 \text{ cycles/s} = 20 \times 10^9$
- ✧ Clock cycles trên máy Y = $1.1 \times 20 \times 10^9 = 22 \times 10^9 \text{ cycles}$
- ✧ Clock rate cho máy Y = $22 \times 10^9 \text{ cycles} / 6 \text{ sec} = 3.67 \text{ GHz}$

Clock Cycles per Instruction (CPI)

❖ Chương trình là một chuỗi các lệnh

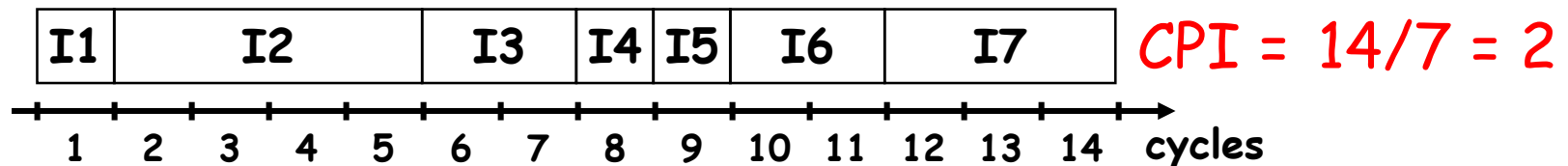
Program (C Language):

```
swap(int v[], int k) {  
    int temp;  
    temp = v[k];  
    v[k] = v[k+1];  
    v[k+1] = temp;  
}
```

MIPS Assembly Language:

```
sll $2,$5, 2  
add $2,$4,$2  
lw  $15,0($2)  
lw  $16,4($2)  
sw  $16,0($2)  
sw  $15,4($2)  
jr  $31
```

❖ CPI là **số trung bình** của số chu kỳ xung nhịp trên lệnh



Thời gian thực thi

- ❖ Thông tin từ chương trình A được thực thi ...
 - ✧ Số lượng lệnh (lệnh dạng ngôn ngữ máy hoặc hợp ngữ)
 - ✧ Số lượng chu kỳ xung nhịp CPU thực thi
 - ✧ Thời gian thực thi
- ❖ Liên hệ giữa CPU clock cycles đến Instruction Count

$$\text{CPU clock cycles} = \text{Instruction Count} \times \text{CPI}$$

- ❖ Thời gian thực thi: (liên quan đến số lượng lệnh)

$$\text{Time} = \text{Instruction Count} \times \text{CPI} \times \text{cycle time}$$

Ví dụ

❖ Máy A và B có chung kiến trúc tập lệnh (ISA)

❖ Chương trình P chạy trên A và B

✧ Máy A có clock cycle time: 250 ps và CPI: 2.0

✧ Máy B có clock cycle time: 500 ps và CPI: 1.2

✧ Máy nào chạy P nhanh hơn và bao nhiêu lần?

❖ Lời giải:

✧ Chung ISA => chung số lệnh IC

✧ CPU execution time (A) = $IC \times 2.0 \times 250 \text{ ps} = 500 \times IC \text{ ps}$

✧ CPU execution time (B) = $IC \times 1.2 \times 500 \text{ ps} = 600 \times IC \text{ ps}$

✧ Máy A nhanh hơn máy B = $\frac{600 \times IC}{500 \times IC} = 1.2$

Xác định CPI

❖ Phân biệt CPI trung bình của chương trình và lệnh

CPI_i = số chu kỳ xung nhịp của loại lệnh i

C_i = số lệnh của loại lệnh i

$$\text{CPU cycles} = \sum_{i=1}^n (CPI_i \times C_i)$$

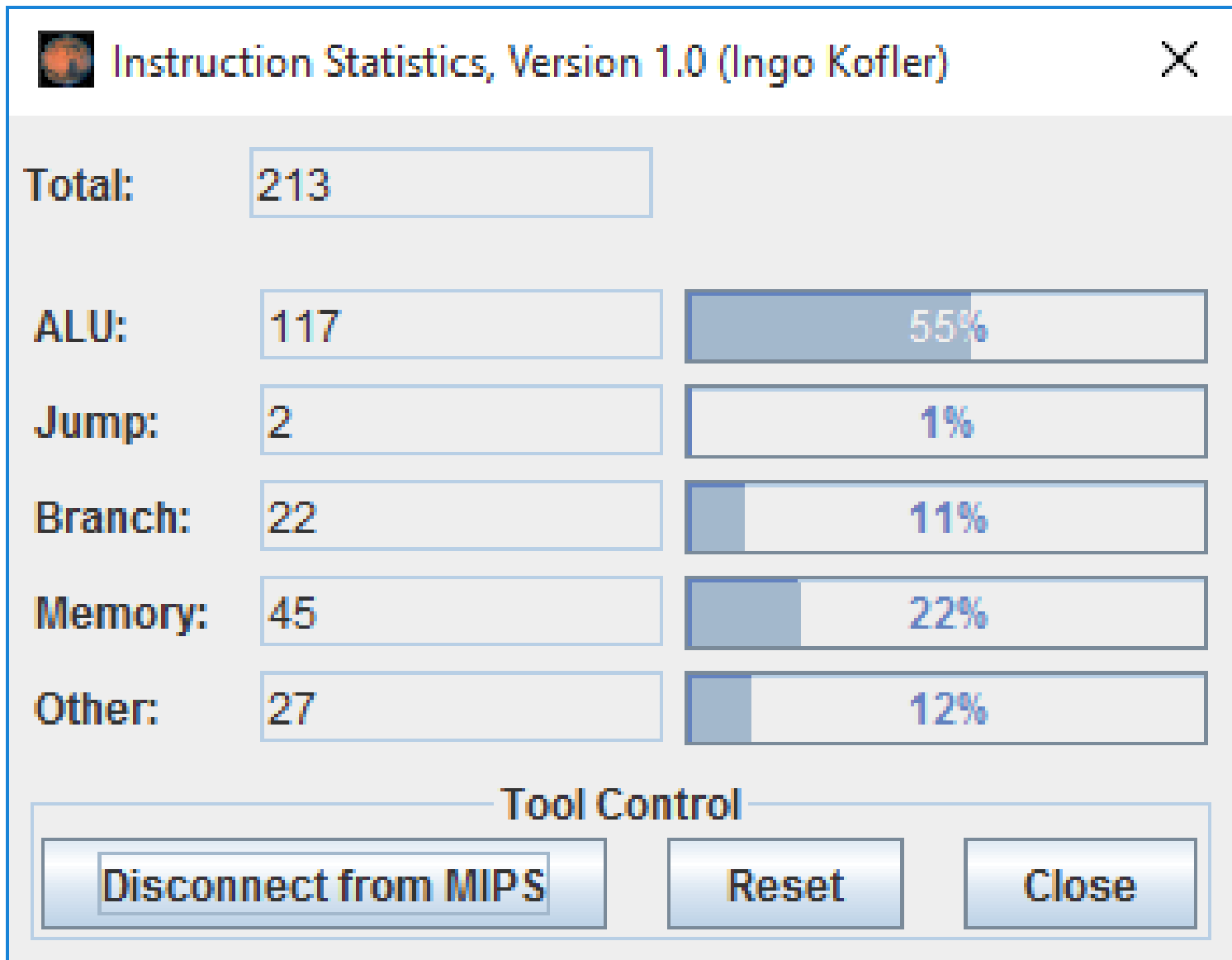
$$CPI = \frac{\sum_{i=1}^n (CPI_i \times C_i)}{\sum_{i=1}^n C_i}$$

Program (C Language):

```
swap(int v[], int k) {  
    int temp;  
    temp = v[k];  
    v[k] = v[k+1];  
    v[k+1] = temp;  
}
```

MIPS Assembly Language:

```
sll $2,$5, 2  
add $2,$4,$2  
lw $15,0($2)  
lw $16,4($2)  
sw $16,0($2)  
sw $15,4($2)  
jr $31
```



Ví dụ: CPI trung bình

❖ Sau khi biên dịch 1 chương trình với 3 loại lệnh A, B, C cho kết quả:

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

■ Kết quả biên dịch 1: IC = 5

- Clock Cycles
 $= 2 \times 1 + 1 \times 2 + 2 \times 3$
 $= 10$

- Avg. CPI = $10/5 = 2.0$

■ Kết quả biên dịch 2: IC = 6

- Clock Cycles
 $= 4 \times 1 + 1 \times 2 + 1 \times 3$
 $= 9$

- Avg. CPI = $9/6 = 1.5$

Ví dụ 2

Cho các thông số của chương trình như bảng dưới
CPI trung bình?

Tỉ lệ % thời gian của từng nhóm lệnh?

Class _i	Freq _i	CPI _i	CPI _i × Freq _i	%Time
ALU	50%	1	$0.5 \times 1 = 0.5$	$0.5/2.2 = 23\%$
Load	20%	5	$0.2 \times 5 = 1.0$	$1.0/2.2 = 45\%$
Store	10%	3	$0.1 \times 3 = 0.3$	$0.3/2.2 = 14\%$
Branch	20%	2	$0.2 \times 2 = 0.4$	$0.4/2.2 = 18\%$

$$\text{Average CPI} = 0.5 + 1.0 + 0.3 + 0.4 = 2.2$$

Tính speed up trong trường hợp CPI của lệnh load = 2?

Tính speed up trong trường hợp 2 lệnh ALU thực thi trong
1 chu kỳ xung nhịp?

Thông số MIPS – Triệu lệnh trên giây

- ❖ MIPS: Millions Instructions Per Second
- ❖ Được dùng làm thông số đo hiệu suất
 - ✧ Máy chạy nhanh hơn \Rightarrow MIPS lớn hơn
- ❖ MIPS đo tốc độ xử lý lệnh

$$\text{MIPS} = \frac{\text{Instruction Count}}{\text{Execution Time} \times 10^6} = \frac{\text{Clock Rate}}{\text{CPI} \times 10^6}$$

- ❖ Thời gian thực thi tính theo MIPS

$$\text{Execution Time} = \frac{\text{Inst Count}}{\text{MIPS} \times 10^6} = \frac{\text{Inst Count} \times \text{CPI}}{\text{Clock Rate}}$$

Hạn chế của thông số MIPS

Ba vấn đề của thông số MIPS

1. Không tính đến số lượng lệnh

- ✧ Không thể dung MIPS để so sánh các máy tính có tập lệnh khác nhau vì IC sẽ khác

2. MIPS thay đổi trên cùng một máy tính

- ✧ Một máy tính không thể có thông số MIPS chung cho tất cả chương trình

3. MIPS có thể đối lập với thông số hiệu suất

- ✧ MIPS cao không đồng nghĩa hiệu suất tốt hơn
- ✧ Xem ví dụ ở slide kế

Ví dụ về thông số MIPS

- ❖ Hai compiler được so sánh khi biên dịch cùng một chương trình chạy trên một máy tính 4 GHz với 3 nhóm lệnh: Class A, Class B và Class C; CPI tương ứng 1, 2 và 3 chu kỳ xung nhịp
- ❖ Compiler 1 sinh ra 5 tỉ lệnh Class A, 1 tỉ lệnh Class B và 1 tỉ lệnh Class C.
- ❖ Compiler 2 sinh ra 10 tỉ lệnh Class A, 0.5 tỉ lệnh Class B và 1 tỉ lệnh Class C.
- ❖ Compiler nào sinh ra chương trình có thông số MIPS cao hơn?
- ❖ Compiler nào sinh ra chương trình có thời gian thực thi tốt hơn?

Lời giải

❖ Số chu kỳ xung nhịp tương ứng

✧ CPU cycles (compiler 1) = $(5 \times 1 + 1 \times 2 + 1 \times 3) \times 10^9 = 10 \times 10^9$

✧ CPU cycles (compiler 2) = $(10 \times 1 + 0.5 \times 2 + 1 \times 3) \times 10^9 = 14 \times 10^9$

❖ Thời gian thực thi tương ứng

✧ Execution time (compiler 1) = $10 \times 10^9 \text{ cycles} / 4 \times 10^9 \text{ Hz} = 2.5 \text{ sec}$

✧ Execution time (compiler 2) = $14 \times 10^9 \text{ cycles} / 4 \times 10^9 \text{ Hz} = 3.5 \text{ sec}$

❖ Compiler1 sinh ra chương trình chạy nhanh hơn

❖ Thông số MIPS tương ứng

✧ MIPS = $\text{Instruction Count} / (\text{Execution Time} \times 10^6)$

✧ MIPS (compiler 1) = $(5 + 1 + 1) \times 10^9 / (2.5 \times 10^6) = 2800$

✧ MIPS (compiler 2) = $(10 + 0.5 + 1) \times 10^9 / (3.5 \times 10^6) = 3286$

❖ Compiler2 sinh ra chương trình có thông số MIPS cao hơn!!!

Các yếu tố tác động đến thời gian thực thi

$$\text{Time} = \text{Instruction Count} \times \text{CPI} \times \text{cycle time}$$

	I-Count	CPI	Cycle
Program	X		
Compiler	X	X	
ISA	X	X	X
Organization		X	X
Technology			X

Amdahl's Law

❖ Amdahl's Law dùng để đo Speedup

✧ So sánh máy tính trước và sau khi áp dụng cải tiến E

$$\text{Speedup}(E) = \frac{\text{Performance with } E}{\text{Performance before}} = \frac{\text{ExTime before}}{\text{ExTime with } E}$$

❖ Cải tiến E chiếm tỉ lệ f về thời gian thực thi và được cải thiện s lần và các phần khác có thời gian không đổi

$$\text{ExTime with } E = \text{ExTime before} \times (f / s + (1 - f))$$

$$\text{Speedup}(E) = \frac{1}{(f / s + (1 - f))}$$

Ví dụ về Amdahl's Law

- ❖ Một chương trình chạy mất 100 giây, phần phép nhân tốn 80 giây. Cần cải tiến phép nhân bao nhiêu lần để chương trình chạy nhanh hơn 4 lần?

Lời giải: giả sử phép nhân được cải thiện s lần

$$25 \text{ sec (4 times faster)} = 80 \text{ sec} / s + 20 \text{ sec}$$

$$s = 80 / (25 - 20) = 80 / 5 = 16$$

Cần cải tiến phép nhân nhanh hơn $s = 16$ lần

Chương trình chạy nhanh hơn 5 lần?

$$20 \text{ sec (5 times faster)} = 80 \text{ sec} / s + 20 \text{ sec}$$

$$s = 80 / (20 - 20) = \infty \text{ Không thể!}$$