

Architektura komputerów - wydajność, porównywanie

Przemek Błaśkiewicz

9 października 2024

- W języku wysokiego poziomu

- W języku wysokiego poziomu
- Reprezentatywny dla jakiegoś przypadku

- W języku wysokiego poziomu
- Reprezentatywny dla jakiegoś przypadku
- Mierzalny

- W języku wysokiego poziomu
- Reprezentatywny dla jakiegoś przypadku
- Mierzalny
- Powszechny

- W języku wysokiego poziomu
- Reprezentatywny dla jakiegoś przypadku
- Mierzalny
- Powszechny
- ?

SPECS

	SPEC2017	SPEC2006	SPEC2000	SPEC95	SPEC92	SPEC89
GNU C compiler	←					gcc
Perl interpreter	←			perl		espresso li
Route planning	←		mcf			li eqntott
General data compression	XZ		bzip2		compress	
Discrete Event simulation - computer network	←	omnetpp	vortex	go	sc	
XML to HTML conversion via XSLT	←	xalancbmk	gzip	ijpeg		
Video compression	X264	h264ref	eon	m88ksim		
Artificial Intelligence: alpha-beta tree search (Chess)	deepsjeng	sjeng	twolf			
Artificial Intelligence: Monte Carlo tree search (Go)	leela	gobmk	vortex			
Artificial Intelligence: recursive solution generator (Sudoku)	exchange2	astar	vpr			
		hammer	crafty			
		libquantum	parser			
Explosion modeling	←	bwaves				fpppp
Physics: relativity	←	cactuBSSN				tomcatv
Molecular dynamics	←	namd				doduc
Ray tracing	←	povray				nasa7
Fluid dynamics	←	lbm				spice
Weather forecasting	←	wrf			swim	matrix300
Biomedical imaging: optical tomography with finite elements	parest	games		apsi	hydro2d	
3D rendering and animation	blender			mgrid	su2cor	
Atmosphere modeling	cam4	milc	wupwise	applu	wave5	
Image manipulation	imagick	zeusmp	apply	turb3d		
Molecular dynamics	nab	gromacs	galgel			
Computational Electromagnetics	fotonik3d	leslie3d	mesa			
Regional ocean modeling	roms	deall	art			
		soplex	equake			
		calculix	facerec			

Dla zestawu n testów:

Dla zestawu n testów:

- 1 Skompiluj, uruchom trzykrotnie, weź medianę czasu wykonania (T_i^s)

Dla zestawu n testów:

- 1 Skompiluj, uruchom trzykrotnie, weź medianę czasu wykonania (T_i^s)
- 2 Znormalizuj względem maszyny wzorcowej (wsk. referencyjny):

$$r_i = \frac{T_i^r}{T_i^s}$$

Dla zestawu n testów:

- 1 Skompiluj, uruchom trzykrotnie, weź medianę czasu wykonania (T_i^s)
- 2 Znormalizuj względem maszyny wzorcowej (wsk. referencyjny):

$$r_i = \frac{T_i^r}{T_i^s}$$

- 3 Oblicz średnią geometryczną wszystkich wskaźników:

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

- phoronix-test-suite / openbenchmark.org
- Dhrystone (int) / Whetstone (float)
- Livermore loops (parallel)
- Linpack (top-500, $Ax=B$)

Niezawodność

Miara długości nieprzerwanego dostarczania usługi.

MTTF - *Mean Time To Failure*.

$1/\text{MTTF}$ = częstotl. awarii; FIT (*Failures in Time*): w ujęciu miliarda godzin działania.

Niezawodność

Miara długości nieprzerwanego dostarczania usługi.

MTTF - *Mean Time To Failure*.

$1/\text{MTTF}$ = częstotl. awarii; FIT (*Failures in Time*): w ujęciu miliarda godzin działania.

- Zał.: żywotności komponentów mają $\sim \text{Exp}$ (niezależne od czasu działania)

Niezawodność

Miara długości nieprzerwanego dostarczania usługi.

MTTF - *Mean Time To Failure*.

$1/\text{MTTF}$ = częstotl. awarii; FIT (*Failures in Time*): w ujęciu miliarda godzin działania.

- Zał.: żywotności komponentów mają $\sim \text{Exp}$ (niezależne od czasu działania)
- częstotl. awarii systemu = \sum częstotl. awarii komponentów

- 10 dysków: 1 000 000 MTTF
- 1 kontroler ATA: 500 000 MTTF
- 1 zasilacz: 200 000 MTTF
- 1 chłodzenie: 200 000 MTTF
- 1 kabel ATA: 1 000 000 MTTF

Niech I_c będzie liczbą operacji wykonanych przez program, I_i liczbą operacji i -tego typu, CPI_i – liczbą taktów potrzebną do wykonania instrukcji i -tego typu. Czas wykonania programu oznaczmy przez T , częstotliwość procesora przez f a okres taktu przez τ ($= 1/f$).

Niech I_c będzie liczbą operacji wykonanych przez program, I_i liczbą operacji i -tego typu, CPI_i – liczbą taktów potrzebną do wykonania instrukcji i -tego typu. Czas wykonania programu oznaczmy przez T , częstotliwość procesora przez f a okres taktu przez τ ($= 1/f$).

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

Niech I_c będzie liczbą operacji wykonanych przez program, l_i liczbą operacji i -tego typu, CPI_i – liczbą taktów potrzebną do wykonania instrukcji i -tego typu. Czas wykonania programu oznaczmy przez T , częstotliwość procesora przez f a okres taktu przez τ ($= 1/f$).

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

op	CPI	częstość
ALU	1	0,6
LD/STO	2	0,18
BRA	4	0,12
MEM	8	0,1

Niech I_c będzie liczbą operacji wykonanych przez program, l_i liczbą operacji i -tego typu, CPI_i – liczbą taktów potrzebną do wykonania instrukcji i -tego typu. Czas wykonania programu oznaczmy przez T , częstotliwość procesora przez f a okres taktu przez τ ($= 1/f$).

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

op	CPI	częstość
ALU	1	0,6
LD/STO	2	0,18
BRA	4	0,12
MEM	8	0,1

$$CPI = 1 \cdot 0,6 + 2 \cdot 0,18 + 4 \cdot 0,12 + 8 \cdot 0,1 = 2,24$$

Niech I_c będzie liczbą operacji wykonanych przez program, l_i liczbą operacji i -tego typu, CPI_i – liczbą taktów potrzebną do wykonania instrukcji i -tego typu. Czas wykonania programu oznaczmy przez T , częstotliwość procesora przez f a okres taktu przez τ ($= 1/f$).

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

op	CPI	częstość
ALU	1	0,6
LD/STO	2	0,18
BRA	4	0,12
MEM	8	0,1

$$CPI = 1 \cdot 0,6 + 2 \cdot 0,18 + 4 \cdot 0,12 + 8 \cdot 0,1 = 2,24$$

CPI: cykle procesora + cykle pamięci

Miliony instrukcji na sekundę (MIPS)

Miliony instrukcji na sekundę (MIPS)

$$\text{MIPS} = \frac{I_c}{T \cdot 10^6}$$

Miliony instrukcji na sekundę (MIPS)

$$\text{MIPS} = \frac{l_c}{T \cdot 10^6}$$

$$T = \text{CPI} \cdot l_c \cdot \tau = \text{CPI} \cdot l_c \cdot 1/f$$

Miliony instrukcji na sekundę (MIPS)

$$\text{MIPS} = \frac{l_c}{T \cdot 10^6}$$

$$T = \text{CPI} \cdot l_c \cdot \tau = \text{CPI} \cdot l_c \cdot 1/f$$

$$\text{MIPS} = \frac{f}{\text{CPI} \times 10^6}$$

Miliony instrukcji na sekundę (MIPS)

$$\text{MIPS} = \frac{l_c}{T \cdot 10^6}$$

$$T = \text{CPI} \cdot l_c \cdot \tau = \text{CPI} \cdot l_c \cdot 1/f$$

$$\text{MIPS} = \frac{f}{\text{CPI} \times 10^6}$$

$$\text{FLOPS} = \frac{\text{liczba op. zmn. przez}}{\text{czas wykonania}}$$

$$\text{czas CPU} = \frac{\text{liczba cykli zegara w programie}}{\text{częstotliwość}}$$

$$\text{czas CPU} = \frac{\text{liczba cykli zegara w programie}}{\text{częstotliwość}}$$

$$\text{CPI} = \frac{\text{liczba cykli zegara w programie}}{\text{liczba instrukcji}}$$

$$\text{czas CPU} = \frac{\text{liczba cykli zegara w programie}}{\text{częstotliwość}}$$

$$\text{CPI} = \frac{\text{liczba cykli zegara w programie}}{\text{liczba instrukcji}}$$

$$\text{czas CPU} = \text{liczba instrukcji} \times \text{CPI} \times \text{częstotliwość}^{-1}$$

$$\text{czas CPU} = \frac{\text{liczba cykli zegara w programie}}{\text{częstotliwość}}$$

$$\text{CPI} = \frac{\text{liczba cykli zegara w programie}}{\text{liczba instrukcji}}$$

$$\text{czas CPU} = \text{liczba instrukcji} \times \text{CPI} \times \text{częstotliwość}^{-1}$$

$$\text{czas CPU} = \sum_{i=1}^m (\text{liczba instrukcji}_i \times \text{CPI}_i) \times \text{częstotliwość}^{-1}$$

- częstotl. operacji zmn. przec. : 25%
- średnie CPI op. zmn. przec. : 4,0
- średnie CPI pozost. op.: 1,33
- częstotl. operacji FSQRT: 2%
- CPI FSQRT: 20

Co jest bardziej opłacalne: zmniejszyć CPI FSQRT do 2 (10-krotnie!), czy wszystkich operacji zmn. przec. do 2,5?

Poprawa wydajności wynikająca z użycia szybszej metody realizacji zadania jest ograniczona ułamkiem czasu, przez jaki ta metoda może być wykorzystana w odniesieniu do całego czasu wykonania.

Poprawa wydajności wynikająca z użycia szybszej metody realizacji zadania jest ograniczona ułamkiem czasu, przez jaki ta metoda może być wykorzystana w odniesieniu do całego czasu wykonania.

- f — część czasu, dla której możliwe jest zastosowanie ulepszenia (≤ 1)

Poprawa wydajności wynikająca z użycia szybszej metody realizacji zadania jest ograniczona ułamkiem czasu, przez jaki ta metoda może być wykorzystana w odniesieniu do całego czasu wykonania.

- f — część czasu, dla której możliwe jest zastosowanie ulepszenia (≤ 1)
- S — przyspieszenie wynikające z zastosowania ulepszenia (> 1)

Poprawa wydajności wynikająca z użycia szybszej metody realizacji zadania jest ograniczona ułamkiem czasu, przez jaki ta metoda może być wykorzystana w odniesieniu do całego czasu wykonania.

- f — część czasu, dla której możliwe jest zastosowanie ulepszenia (≤ 1)
- S — przyspieszenie wynikające z zastosowania ulepszenia (> 1)

Poprawa wydajności wynikająca z użycia szybszej metody realizacji zadania jest ograniczona ułamkiem czasu, przez jaki ta metoda może być wykorzystana w odniesieniu do całego czasu wykonania.

- f — część czasu, dla której możliwe jest zastosowanie ulepszenia (≤ 1)
- S — przyspieszenie wynikające z zastosowania ulepszenia (> 1)

$$\text{przyspieszenie} = \frac{1}{(1 - f) + \frac{f}{S}}$$