Cuprins

Conceptul de performanţă

Performanţa poate avea mai multe accepţiuni:
(1) la un PC, suntem interesaţi de **timpul de răspuns** al unui program: este timpul fizic scurs între momentul când lansăm cererea şi momentul când primim rezultatul; el trebuie să fie cât mai mic;
(2) la un centru de calcul, suntem interesaţi de **productivitate** (**throughput**): numărul de sarcini executate în unitatea de timp; el trebuie să fie cât mai mare.

Obs: (1) $\Longrightarrow$ (2) dar (2) $\not\Longrightarrow$ (1).
De exemplu, centrul de calcul poate avea calculatoare lente dar multe; atunci pe fiecare calculator vom avea timpi de răspuns mari, dar numărul de sarcini executate în unitatea de timp în centrul de calcul va fi tot mare.

Ambele accepţiuni ale performanţei necesită abordări complexe, aşa că pentru simplitate trebuie să ne limităm la una din ele.

În cele ce urmează, **performanţa** se va referi la **timpul de răspuns**.
El priveşte rularea unui program cu nişte date şi depinde de mai mulţi factori: codul programului, datele cu care se rulează, promptitudinea cu care utilizatorul introduce datele când sunt cerute, sistemul de operare, încărcarea sistemului (activitatea concomitentă a altor utilizatori), arhitectura hardware, etc.

Conceptul de performanţă

Pentru a da o dimensiune numerică performanţei (P) şi a descrie dependenţa ei descrescătoare în raport cu timpul de răspuns (T), adoptăm formula:

$$P = \frac{1}{T} \quad (1)$$

Deja putem să comparăm performanţele sau să calculăm un raport al performanţelor unui program cu nişte date fixate, pe două maşini X şi Y:

$$P_X > P_Y \overset{\text{def}}{\Longleftrightarrow} T_X < T_Y, \quad \frac{P_X}{P_Y} = n \overset{\text{def}}{\Longleftrightarrow} \frac{T_Y}{T_X} = n \quad (2)$$

(am notat cu P_X, P_Y, T_X, T_Y, performanţele respectiv timpii de răspuns, pe maşinile X şi Y).

Conceptul de performanţă

Timpul de răspuns (al unui program cu nişte date) are următoarele componente:
(1) Timpul consumat cu executarea unor operaţii din program;
(2) Timpul consumat cu executarea unor operaţii din sistemul de operare, dar legate de programul nostru (ex: apeluri sistem cerute de program);
(3) Timpul consumat de program în aşteaptarea unor evenimente/resurse (ex: programul este adormit în "scanf()" aşteptând ca utilizatorul să introducă date de la consolă);
(4) Timpul în care programul este suspendat de sistemul de operare, pentru a executa operaţii adminstrative şi sarcini legate de alte programe (care rulează în paralel cu al nostru).

Notăm (1) + (2) = **Timpul CPU**
Deci timpul CPU este timpul consumat de program cu executarea unor operaţii (calcule), fie ele din program sau din sistemul de operare dar legate de program. El este inclus în timpul de răspuns, iar ceea ce rămâne este timpul consumat de program în aşteptare, fie că aşteaptă un eveniment/resursă, fie că este suspendat de sistemul de operare pentru a executa operaţii adminstrative şi sarcini legate de alte programe.

Conceptul de performanţă

Pe anumite sisteme există comenzi cu ajutorul cărora se pot măsura şi afişa diferite componente ale timpului de răspuns pentru o anumită cerere. De exemplu, în Linux comanda:

$$time \quad cmd$$

afişază diferite componente ale timpului de răspuns obţinute la executarea comenzii cmd. De exemplu, poate afişa:

$real$	$2m39.000s$	(timpul de răspuns)
$user$	$1m30.700s$	(timpul utilizator, (1))
sys	$0m12.900s$	(timpul sistem, (2))

Astfel, putem calcula procentajul timpului CPU din timpul de răspuns:

$$\frac{90.7s + 12.9s}{159s} = 65\%$$

Deci $> 1/3$ din timpul de răspuns a fost consumat în aşteptare, ceea ce ne spune ceva despre mediul în care a fost rulat programul - resurse greu disponibile, grad mare de încărcare a sistemului, etc.

Conceptul de performanţă

Distingem între:
performanţa sistemului = timpul de răspuns pe un sistem neîncărcat;
performanţa CPU = timpul CPU.

În cele ce urmează restrângem şi mai mult domeniul abordării şi vom considera că **performanţă** înseamnă **performanţa CPU** (deci în formula **(1)** este vorba de **timpul CPU**).

Cuprins

Măsurarea performanței

Timpul (CPU) se poate măsura în:
- **secunde** (s)
- **cicli de ceas** (c)

Legătura dintre cele două este dată de peroada de ceas (durata ciclului) D (măsurată în secunde (s), nanosecunde (ns), etc.) sau frecvența F (măsurată în herzi (Hz), megaherzi (MHz), etc.), după formula:

$$F = \frac{1}{D}$$

Când măsurăm în cicli nu mai contează fizica circuitelor (durata ciclului) ci doar împărțirea programului în operații mașina (logica programului) și a operațiilor mașină în ciclii (logica circuitelor) - avem astfel posibilitatea să comparăm d.p.v. logic o gamă mai largă de mașini.

Măsurarea performanței

O mărime care ne dă indicații asupra performanței este **numărul total de cicli executați** C.

El depinde de program, de date și de împărțirea instrucțiunilor în cicli (am presupus programul scris în limbaj mașină); nu depinde de durata ciclului. Performanța este cu atât mai mare cu cât C este mai mic.

Dacă presupunem programul scris într-un limbaj de nivel înalt, contează și împărțirea codului sursă în instrucțiuni mașină - deci intervine și performanța compilatorului.

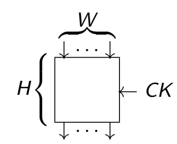
Atunci putem calcula timpul CPU T după formula:

$$T = C \times D = \frac{C}{F} \quad (3)$$

Această formulă arată că putem reduce timpul T (deci mări performanța) reducând numărul de cicli executați C sau/și durata ciclului D.
Este dificil însă de redus simultan ambii factori - de obicei reducerea unuia atrage creșterea celuilalt.

Măsurarea performanței

Pentru exemplificare, să considerăm următorul model simplificat de circuit, care execută câte o operație la fiecare ciclu:



El este construit prin conectarea altor circuite mai simple, în serie, în paralel, etc. (a se vedea capitolul de circuite logice), are o anumită lățime W, dată de numărul de intrări și de conectări în paralel, și o anumită înalțime H, dată de numărul de niveluri de conectare în serie; W este cu atât mai mare cu cât datele de prelucrat au o dimensiune mai mare (mai mulți biți), iar H este cu atât mai mare cu cât operația este mai complexă.

La fiecare tact de ceas (CK), datele de intrare sunt prelucrate prin circuit și transformate în niște date de ieșire, care la următorul tact vor fi preluate ca date intrare de un circuit conectat în continuare. Această prelucrare necesită un anumit timp, iar D trebuie să fie $\geq$ acest timp - altfel, dacă următorul tact vine prea devreme (înainte ca ieșirile să fie stabile), vor fi culese (și trimise mai departe) niște date eronate.

Măsurarea performanței

Creșterea D nu este influiențată de W, deoarece intrările sunt procesate în paralel, ci de H (lungimea traseelor ce trebuie parcurse); în capitolul de circuite logice vom vedea, de exemplu, că o adunare pe 32 biți se face la fel de repede ca o adunare pe 64 biți (deoarece biții operanzilor sunt procesați în paralel), dar o adunare se face mai repede decât o înmulțire.

Să considerăm acum că circuitul desenat mai sus execută o operație pe parcursul mai multor cicli - deci sunt necesare mai multe treceri prin el pentru a se efectua toate prelucrările necesare.
Întrucât efectuarea unei anumite operații necesită o anumită cantitate de prelucrări (care depinde de natura matematică a operației), proiectantul circuitului va avea de ales între două variante:

- va împărți prelucrarea în etape puține, realizate câte una la ficare ciclu, dar atunci la fiecare etapă trebuie făcute calcule multe, deci H va fi mare și astfel D va fi mare; atunci, la executarea programului vom avea C mic și D mare;

- va împărți prelucrarea în etape multe și atunci la fiecare etapă trebuie făcute calcule puține, deci H și D vor fi mici; atunci, la executarea programului vom avea C mare și D mic.

Exercițiu rezolvat

Exercițiu rezolvat:

Un program (cu niște date fixate) durează pe mașina A, având frecvența $400\ MHz$, 10 secunde. Dorim o mașină B pe care rularea să dureze 6 secunde. Proiectantul spune că poate crește frecvența, dar programul va necesita de 1.2 ori mai mulți cicli. Care va fi frecvența mașinii B ?

Explicație: proiectantul dorește reducerea duratei ciclului prin micșorarea înalțimii circuitelor (H din desenul anterior); atunci instrucțiunile vor necesita mai multe cicluri pentru a fi executate, factorul mediu de multiplicare fiind 1.2.

Exercițiu rezolvat

Rezolvare:

Din enunț ni se dau:
$F_A = 400\ MHz$ (frecvența mașinii A, în MHz);
$T_A = 10\ s$ (timpul pe mașina A, în secunde s);
$T_B = 6\ s$ (timpul pe mașina B, în secunde s);
$C_B/C_A = 1.2$ (raportul numerelor de cicluri pe mașinile A, B);
Se cere:
$F_B = ?$ (frecvența mașinii B, în MHz).

Vom avea:

$$F_B \overset{(3)}{=} \frac{C_B}{T_B} \overset{ip}{=} \frac{1.2\times C_A}{T_B} \overset{(3)}{=} \frac{1.2\times F_A\times T_A}{T_B} \overset{ip}{=} \frac{1.2\times 400\,MHz\times 10\,s}{6\,s} = 800\,MHz$$

Comentariu: deși s-a dublat frecvența, performanța (raportul timpilor) nu s-a dublat; deci, performanța se obține greu.

Exercițiu rezolvat

Sfat:

În redactarea rezolvărilor, la efectuarea calculelor, menționați la fiecare pas și unitățile de măsură; utilitatea este următoarea:

- unele unități de măsură pot conține multipli care trebuie luați în considerare (de exemplu, 1 Mega $= 10^6$);
- avem posibilitatea să ne verificăm corectitudinea aplicării formulelor (de exemplu, dacă ne așteptăm să obținem secunde și rezultă cicli, nu este bine).

Convenție:

În cele ce urmează, mărimea fizică va fi notată cu literă mare iar unitatea de măsură cu literă mică, pe cât posibil aceeași literă.
De exemplu, $C = 10\ c$ înseamnă că numărul total de cicli executați este 10 (cicli).

Măsurarea performanței

Alte două mărimi care ne dau indicații asupra performanței sunt **numărul total de instrucțiuni executate** I și **numărul mediu ce cicli per instrucțiune executați** (engl: **cycles per instruction**) CPI, calculat după formula:

$$CPI = \frac{C}{I} \quad (4)$$

I depinde de program și de date; nu depinde de împărțirea instrucțiunilor în cicli și nici de durata ciclului.
Performanța este cu atât mai mare cu cât I este mai mic.
CPI depinde de program, de date și de împărțirea instrucțiunilor în cicli; nu depinde de durata ciclului.
Performanța este cu atât mai mare cu cât CPI este mai mic.

Măsurarea performanței

CPI ne permite să comparăm două implementări hardware diferite ale unui același set de instrucțiuni.

De exemplu, să presupunem că avem de implementat o listă ce conține instrucțiunile $+$ (adunare) și $*$ (înmulțire).

Presupunem că avem două implementări, în care numărul de cicli necesari fiecărei instrucțiuni este dat de tabelul:

	Nr. cicli $+$	Nr. cicli $*$
Implementare 1	10 c	100 c
Implementare 2	12 c	70 c

Se pune problema care implementare este mai bună. Răspunsul depinde de procentajul cu care sunt executate fiecare dintre cele două instrucțiuni.
Daca pentru categoria de programe și de date avute în vedere, la rulare se execută în medie $> 93.75\%$ $+$ și restul $*$, este preferabilă prima implementare; dacă se execută în medie $< 93.75\%$ $+$ și restul $*$, este preferabilă a doua implementare (exercițiu !).
Numărul mediu de cicli per instrucțiune calculat ținând cont de ponderea apariției fiecărui tip de instrucțiuni este tocmai CPI. Așasar, cu cât CPI este mai mic, performanța este mai mare (pentru categoria de programe și de date vizată).

Măsurarea performanței

Din formulele (3) și (4) obținem **ecuația elementară a performanței**:

$$T = I \times CPI \times D = \frac{I \times CPI}{F} \quad (5)$$

Ea leagă cei trei **factori cheie** ai performanței: numărul de instrucțiuni executate I, numărul mediu de cicli pe instrucțiune executați CPI și durata ciclului D (sau frecvența F).

Există interdependențe între factori; de exemplu, scăderea CPI poate atrage creșterea D (deoarece la fiecare ciclu se vor executa operații mai multe și/sau mai complexe).

Acești factori trebuie considerați simultan atunci când analizăm un sistem, altfel putem trage concluzii eronate privind performanța - a se vedea următorul exercițiu rezolvat.

Măsurarea performanței

Ne punem problema cum putem măsura T folosind formulele (3) sau (5).

C este greu de măsurat - el depinde de program, de date și de arhitectură (împărțirea instrucțiunilor în cicli).

I este mai ușor de măsurat - depinde doar de program și de date; unele procesoare au chiar contoare hardware care numără instrucțiunile executate - ele se folosesc la simulări.

CPI este însă greu de măsurat - el depinde de program, de date și de arhitectură (împărțirea instrucțiunilor în cicli).

Măsurarea performanţei

Uneori poate fi suficientă o aproximare a lui C, obţinută urmărind execuţia programului pe clase de instrucţiuni. Mai exact:

Considerăm n clase de instrucţiuni.
De exemplu: clasa instrucţiunilor de salt (din care face parte 'goto'), clasa instrucţiunilor aritmetice aditive (din care fac parte adunarea, scăderea), clasa instrucţiunilor aritmetice multiplicative (din care fac parte înmulţirea, împărţirea).

Pentru fiecare clasa $k \in \{1, \ldots, n\}$ notăm cu CPI_k media numerelor de cicli necesitaţi de instrucţiunile din clasa k.
De exemplu, $CPI_{aditive}$ este media dintre numărul de cicli necesitaţi de o adunare şi numărul de cicli necesitaţi de o scădere.
Notăm deci că la calcularea lui CPI_k se ia câte un singur exemplar din fiecare instrucţiune a clasei k, fără ponderi/procentaje.

Pentru un program şi nişte date considerate, notăm cu I_k numărul de instrucţiuni executate din clasa k, $k \in \{1, \ldots, n\}$.

Măsurarea performanţei

Atunci pentru programul, datele şi arhitectura considerate, vom avea:

$$C \simeq \sum_{k=1}^{n} (CPI_k \times I_k) \quad (6)$$

Valorile n şi CPI_k, $k \in \{1, \ldots, n\}$, depind doar de arhitectură - pentru o maşină dată ele sunt fixate şi pot fi comunicate prin documentaţia maşinii.

Valorile I_k, $k \in \{1, \ldots, n\}$, depind de program şi de date, nu şi de arhitectură.

Astfel, pe o maşină dată, dacă vrem să aflăm (cu aproximaţie) numărul de cicli C consumaţi la rularea unui anumit program cu anumite date, este suficient să aflăm valorile I_k, $k \in \{1, \ldots, n\}$, iar acestea sunt mai uşor de aflat, deoarece instrucţiunile se numără mai uşor decât ciclii.

Măsurarea performanţei

Valoarea C calculată cu formula (6) este însă doar una aproximativă.

De exemplu, presupunem că pe o maşină dată avem:

Clasa	Instrucţiuni	Nr. de cicli necesitaţi
aditive	adunare	1
	scădere	3
multiplicative	înmulţire	10
	împărţire	30

Rezultă $CPI_{aditive} = 2$, $CPI_{multiplicative} = 20$.

Considerăm un program şi nişte date a.î. la rulare se execută o adunare şi o înmulţire.
Valoarea exactă a lui C este: $1 + 10 = 11$.
Valoarea lui C calculată cu formula (6) este: $2 + 20 = 22$.

Considerăm acum un program şi nişte date a.î. la rulare se execută o scădere şi o împărţire.
Valoarea exactă a lui C este: $3 + 30 = 33$.
Valoarea lui C calculată cu formula (6) este tot: $2 + 20 = 22$.

Măsurarea performanţei

Într-adevăr, cu formula (6) am considerat de fiecare dată că se execută o instrucţiune aditivă şi o instrucţiune multiplicativă, nesesizând că la primul program se execută cele mai rapide instrucţiuni din cele două clase, iar la al doilea program cele mai lente.

Valoarea lui C calculată cu formula (6) se apropie de valoarea exactă cu atât mai mult cu cât din fiecare clasă instrucţiunile se execută cu ponderi mai apropiate (cam tot atâtea adunări câte scăderi, cam tot atâtea înmulţiri câte împărţiri).

Exerciţiu rezolvat

Exerciţiu rezolvat:
Pe o maşină dată, avem următoarele clase de instrucţiuni şi CPI asociate:

clasa	CPI_{clasa}
A	1
B	2
C	3

Considerăm două programe şi nişte date, pentru care s-au măsurat:

Program	Nr. de instrucţiuni executate din fiecare clasă		
	A	B	C
programul 1	2	1	2
programul 2	4	1	1

Se cer: a) Care program execută mai multe instrucţiuni ?
b) Care program este mai rapid ?
c) Cât este CPI pentru fiecare program ?

Exerciţiu rezolvat

Rezolvare:
a) Notăm:
$I_k^p =$ numărul de instrucţiuni executate de programul $p \in \{1, 2\}$ din clasa $k \in \{A, B, C\}$;
$I^p =$ numărul total de instrucţiuni executate de programul $p \in \{1, 2\}$.
Avem:
$I^1 = I_A^1 + I_B^1 + I_C^1 = 2i + 1i + 2i = 5i$
$I^2 = I_A^2 + I_B^2 + I_C^2 = 4i + 1i + 1i = 6i$
Comentariu: am obţinut $I^1 < I^2$, ceea ce pare să arate că programul 1 este mai performant.

Exercițiu rezolvat

b) Notăm:

$C^p =$ numărul total de cicli executați de programul $p \in \{1, 2\}$.

Folosind formula (6), avem:

$$C^1 = CPI_A \times I_A^1 + CPI_B \times I_B^1 + CPI_C \times I_C^1$$
$$= 1\tfrac{c}{i} \times 2i + 2\tfrac{c}{i} \times 1i + 3\tfrac{c}{i} \times 2i = 10c$$
$$C^2 = CPI_A \times I_A^2 + CPI_B \times I_B^2 + CPI_C \times I_C^2$$
$$= 1\tfrac{c}{i} \times 4i + 2\tfrac{c}{i} \times 1i + 3\tfrac{c}{i} \times 1i = 9c$$

Comentariu:

Am obținut $C^1 > C^2$, ceea ce pare să arate că programul 2 este mai performant, în contradicție cu a).

Dintre mărimile I și C, cea care măsoară realist durata execuției este C, deoarece folosește aceeași unitate de măsură, ciclul (instrucțiunile nu durează toate la fel de mult). Deci răspunsul corect este cel dat de b).

Explicația erorii de la a): programul 1 execută instrucțiuni mai puține, dar mai lungi (execută două din clasa C, care durează în medie câte 3 cicli).

Exercițiu rezolvat

c) Notăm:

$CPI^p =$ numărul mediu de cicli pe instrucțiune executați de programul $p \in \{1, 2\}$.

Folosind formula (4), avem:

$$CPI^1 = \frac{C^1}{I^1} = \frac{10c}{5i} = 2\tfrac{c}{i} \text{ (sau } 2cpi\text{)}.$$
$$CPI^2 = \frac{C^2}{I^2} = \frac{9c}{6i} = 1.5\tfrac{c}{i} \text{ (sau } 1.5cpi\text{)}.$$

Comentariu:

Am obținut $CPI^1 > CPI^2$, ceea ce pare să arate că programul 2 este mai performant, ca la b) (și în contradicție cu a)).

Întrucât am văzut că b) oferă răspunsul corect, rezultă că, cel puțin în acest caz, mărimea CPI descrie performanța mai bine decât mărimea I.

Răspunsul de la c) era de așteptat, deoarece programul 2 este mai rapid deși execută mai multe instrucțiuni și de aceea trebuie ca instrucțiunile executate de el să fie mai rapide.

Oricum, acest exercițiu ilustrează ce am spus mai devreme, că factorii care influiențează performanța trebuie considerați simultan atunci când analizăm un sistem, altfel putem trage concluzii eronate privind performanța.

Măsurarea performanței

O altă mărime care ne dă indicații asupra performanței este **frecvența de executare a instrucțiunilor**:

$$FI = \frac{I}{T} \quad (7)$$

Întrucât de obicei valorile lui FI (măsurate în instrucțiuni pe secundă) sunt mari, se obișnuiește introducerea încă unui factor 10^6 la numitor, obținându-se mărimea:

$$MIPS = \frac{I}{T \times 10^6} \quad (8)$$

măsurată în milioane de instrucțiuni pe secundă (**millions instructions per second**).

FI și $MIPS$ depind de program, de date, de împărțirea instrucțiunilor în cicli și de durata ciclului.

Performanța este cu atât mai mare cu cât FI și $MIPS$ sunt mai mari.

Măsurarea performanței

Ca și în cazul altor mărimi discutate mai devreme, FI și $MIPS$ trebuie considerate împreună cu ceilalți factori atunci când analizăm un sistem, altfel putem trage concluzii eronate privind performanța - a se vedea următorul exercițiu rezolvat.

Exercițiu rezolvat

Exercițiu rezolvat:

Pe o mașină dată, având frecvența de $500\ MHz$, avem următoarele clase de instrucțiuni și CPI asociate:

clasa	CPI_{clasa}
A	1
B	2
C	3

Considerăm două programe și niște date, pentru care s-au măsurat:

Program	Nr. de instrucțiuni executate din fiecare clasă		
	A	B	C
programul 1	5	1	1
programul 2	10	1	1

Se cer: a) Care sunt timpii CPU pentru cele două programe ?

b) Care sunt valorile MIPS pentru cele două programe ?

Exercițiu rezolvat

Rezolvare:

Situația seamănă cu cea din exercițiul precedent și atunci la fel ca acolo putem afla numărul total de cicli executați. Acum știm în plus și frecvența, iar aceasta ne va permite ca din numărul de cicli să aflăm timpii CPU.

Notăm:

$I_k^p =$ numărul de instrucțiuni executate de programul $p \in \{1, 2\}$ din clasa $k \in \{A, B, C\}$;

$I^p =$ numărul total de instrucțiuni executate de programul $p \in \{1, 2\}$.

$C^p =$ numărul total de cicli executați de programul $p \in \{1, 2\}$.

$T^p =$ timpul CPU consumat de programul $p \in \{1, 2\}$.

$MIPS^p =$ valoarea MIPS pentru programul $p \in \{1, 2\}$.

Exerciţiu rezolvat

a) Folosind formula (6), avem:

$$C^1 = CPI_A \times I_A^1 + CPI_B \times I_B^1 + CPI_C \times I_C^1$$
$$= 1\tfrac{c}{i} \times 5i + 2\tfrac{c}{i} \times 1i + 3\tfrac{c}{i} \times 1i = 10c$$
$$C^2 = CPI_A \times I_A^2 + CPI_B \times I_B^2 + CPI_C \times I_C^2$$
$$= 1\tfrac{c}{i} \times 10i + 2\tfrac{c}{i} \times 1i + 3\tfrac{c}{i} \times 1i = 15c$$

Atunci, folosind formula (3), avem:

$$T^1 = \frac{C^1}{F} = \frac{10c}{500\,MHz} = \frac{10c}{500 \times 10^6\,Hz} = \frac{10c}{500 \times 10^6\,\frac{c}{s}}$$
$$= 2 \times 10^{-8}s$$
$$T^2 = \frac{C^2}{F} = \frac{15c}{500\,MHz} = \frac{15c}{500 \times 10^6\,\frac{c}{s}} = 3 \times 10^{-8}s$$

Comentariu: am obţinut $T^1 < T^2$, ceea ce pare să arate că programul 1 este mai performant. Mărimea T măsoară realist performanţa, deoarece foloseşte acelaşi etalon pentru ambele programe (toate secundele sunt la fel de lungi).