ОЦЕНКА НА ПРОИЗВОДИТЕЛНОСТТА НА КОМПЮТЪРНИТЕ СИСТЕМИ

ИЗМЕРВАНЕ И ПРЕДСТАВЯНЕ НА ПРОИЗВОДИТЕЛНОСТТА

- **за потребители на настолни компютри** – намаляване ***на времето за отговор*** (времето между началото и завършването на събитие, наричано също ***време за изпълнение***)
- **за мениджърите на големи изчислителни центрове** – увеличаване на ***пропускателната способност*** (брой на задачите, завършени за единица време от машината)
  - **сравняване на производителността на два различни компютъра**:

X е n пъти по-бърз от Y

Време за изпълнение (Y) / Време за изпълнение (X)=n

- ***Времето за изпълнение е реципрочно на производителността !***
- **Увеличаването на производителността намалява времето за изпълнение →** **термини**: ***“подобряване на производителността”*** ***означава “намаляване на времето за изпълнение”*** ***или “увеличаване на пропускателната способност”***
- **Компютърът, който изпълнява същото количество работа за най-кратко време е най-бързият !**

- **Дефиниране на *време* : по стенен часовник, времето за отговор или изминалото време т.е. закъснението (латентността) за завършването на задача, включващо времето на CPU, времената за достъп до дисковете, достъпите до паметта, В/И операции, допълнителните разходи на OS, и др.**
- **Процесорно време :**

1. Потребителско процесорно време – времето, изразходвано за изпълнение на програмата;

2. Системно процесорно време – времето, изразходвано от OS да управлява изпълнението на задачите, заявени от програмата.

Избор на програми за оценка на производителността

Работен товар – програмната смес & командите на ОС, които потребителите използват за изпълнение от компютъра

5 нива на програми, представени в намаляващ ред относно точността на прогнозата:

1. ***Реални приложения*** – **компилатори на C, текстообработващи програми, Photoshop, др. (проблеми на преносимостта, дължащи се на зависимости от ОС или компилатора)**
2. ***Модифицирани (скриптирани) приложения*** – **фокусират върху зададен аспект на производителността**

Основни мерки за производителността

- ***Мерки за работен товар и скорост***
- *времето за изпълнение*
- *броят на изпълнените инструкции*
- *броят на изпълнените операции с плаваща точка*

- *Времето за изпълнение* е ***тясно обвързано със специфичната компютърна система***. То може да се промени при изпълнението на същата програма от друг компютър.

- *броят на изпълнените инструкции*, е *тясно свързан с архитектурата на системата инструкции* на процесора и не се променя, ако програмата се изпълнява на друг компютър със същата архитектура на системата инструкции
- *броят на изпълнените операции с плаваща точка, често е архитектурно независима*

Тип на работния товар	Единица за измерване на работния товар	Единица за измерване на скоростта
Време за изпълнение	Секунди (s), цикли на процесора	Приложение за секунда
Брой на инструкциите	Милиони инструкции или Милиарди инструкции	MIPS или BIPS
Брой на операциите с плаваща точка	Flop, Mflop, Gflop	Mflop/s, Gflop/s

Брой на инструкциите

- Може да се приложи за измерване на работния товар на *произволна програма*
- Работният товар се формира от броя на инструкциите, които машината е изпълнила (*динамичен брой инструкции на програмата*), а не броя на инструкциите, фигуриращи в текста на асемблерната програма (статичен брой инструкции).
- *Динамичният брой на инструкциите на програмата може да зависи от стойностите на входните данни.*

- За да се избегне зависимостта от входните данни, работният товар се оценява *за най-лошия случай на входни данни или за зададено специфично множество входни данни*.
- Дори при едни и същи входни данни, *броят на изпълнените инструкции може да варира при изпълнението на програмата на различни компютри.* Например, често RISC процесорите изпълняват от 50% до 150% повече инструкции от CISC процесорите при една и съща програма на език от високо ниво.

- *Броят на изпълнените инструкции може да варира* при изпълнението на програмата дори и на един и същ компютър, когато се използват *различни компилатори или различни оптимизационни техники*.
- По-голям брой изпълнени инструкции не означава задължително по-голямо време за изпълнение.

Време за изпълнение

- За дадена програма на определен компютър работният товар може да се дефинира като *цялото време, необходимо за изпълнение на програмата*.
- Времето за изпълнение трябва да се измерва *по стенен часовник*
- Основната измервателна единица в случая е *секунда*

Времето за изпълнение зависи от множество фактори:

- *Алгоритъмът* определя съществено времето за изпълнение
- *Структурирането на данните*
- *Входните данни*
- *Компютърната платформа:* очевидно, хардуерът и операционната система влияят върху производителността. Различно време за изпълнение може да се получи и при две компютърни платформи с еднакви процесори при една и съща тактова честота.

- Производителността не зависи само от процесора. Другите фактори са *йерархията на паметта, версията на операционната система и дали приложението използва всички компютърни ресурси или ги използва в режим на времеделене с други приложения*.
- *Езика за програмиране:* различни времена за изпълнение се получават при използване на различни езици за кодиране на приложението. Допълнително, *компилаторът и свързващият редактор също оказват значително влияние*.

Брой на операциите с плаваща точка

- ***Правила***: всяко събиране, изваждане или умножение се брои за 1 flop, обработката на индексите не се брои, присвояванията не се броят отделно, изолирано присвояване се брои като 1 flop, сравнението се брои за 1 flop, преобразуването на тип се брои за 1 flop, делението или квадратен корен се броят за 4 flop, изчисляването на синус, експонента, и др. се броят за 8 flop.

- *При използването на времето за изпълнение или броят на инструкциите като мярка за работния товар се получават следните странни ефекти: работният товар се променя при изпълнението на приложението на различни системи и още повече, работният товар може да бъде определен само чрез изпълнението на програмата.*
- Мярката с броя на операциите с плаваща точка е по-стабилна.

Скриптовете (scripts) ***се използват за симулация на приложни програми с цел да се възпроизведе интерактивното поведение на desktop система или да се симулира сложното взаимодействие на множество потребители в сървърна система***

Ядра *– извлечени малки ключови части от реални програми, използвани за оценка на производителността:* ***“Livermore Loops”, “Linpack”***

3. Еталонни програми “играчки” ***–типично обхващат 10-100 реда код и се получават предварително известни резултати: Eratosthenes, Puzzle, Quicksort***

4. Синтетични еталонни програми ***– изкуствено създадени за да възпроизведат среден изчислителен профил: “Whetstones”,“Dhrystones”***

КОМПЛЕКТИ ОТ ЕТАЛОННИ ПРОГРАМИ → *колекция от еталонни програми за измерване на производителността на процесори за широк спектър приложения*

SPEC (Standard Performance Evaluation Corporation) – *създава стандартизирани комплекти от еталонни приложни програми (www.spec.org)*

ЕТАЛОННИ ПРОГРАМИ ЗА ДЕСКТОП СИСТЕМИ

2 широки класа еталонни програми:

с интензивна процесорна обработка

& с интензивна графична обработка

SPEC фокусира върху производителността на процесора:

SPEC89, SPEC92, SPEC95, SPEC2000

SPEC2000 **включва 11 еталонни програми за цели числа (CINT2000) & 14 еталонни програми за плаваща точка (CFP2000)**

Еталоните SPEC са реални програми, модифицирани с оглед преносимост и ролята на В/И е минимизиран.

- **Еталоните за цели числа варират от част от компилатор на C до части от приложни програми за компютърна графика.**
- **Еталоните за числа с плаваща точка вкл. код за квантова динамика, моделиране с крайни елементи и динамика на флуидите.**

Комплектът еталони SPEC CPU се използва за оценка на производителността на ЦП както за desktop системи, така и за еднопроцесорни сървъри.

ЕТАЛОНИ ЗА СЪРВЪРИ

- Мярка SPECrate – прост еталон за пропускателна способност → изпълнение на множество копия на всеки еталон SPEC CPU на мултипроцесор (#copies=#CPU's)
- SPEC предлага SPECSFS (еталон за файлови сървъри), използващ *скрипт на заявки за файловия сървър*; тества производителността на В/И система (disk & I/O) както и на CPU
- SPECWeb е еталон за *Web server*, който симулира множество клиенти, подаващи заявки към сървъра за статични & динамични страници, както и клиенти, изпращащи данни към сървър

ЕТАЛОНИ ЗА ОБРАБОТКА НА ТРАНЗАКЦИИ (TP)

TPC (Transaction Processing Council) –

независим от търговците

www.tpc.org/

Всички TCP еталони измерват производителността в брой транзакции за секунда, като се включва изискване за времето за отговор!

The EDN Embedded Microprocessor Benchmark Consortium (EEMBC, произнася се като “embassy”)

Комплектът еталони на EEMBC се състои от 34 ядра спадащи към 5 класа :

- Automotive/industrial
- Consumer
- Компютърни мрежи
- Автоматизация на офис
- Телекомуникации

- **Средно време за изпълнение** – средна аритметична стойност, където $Time_i$ е времето за изпълнение на $i^{тата}$ програма от общо n в работния товар

$$\frac{1}{n}\sum_{i=1}^{n} Time_i$$

- **Претеглено време за изпълнение:**
- Тегловен фактор w_i се присвоява на всяка програма за отразяване на относителната честота на програмата в работния товар
- **Претеглена аритметична средна стойност:**

$$\frac{1}{n}\sum_{i=1}^{n} Weight_i \, x \, Time_i$$

Закон на Amdahl

Дефинира ускорението (speedup), което може да се получи при използването на дадено качество

$$ускорение = \frac{производителност\_за\_цяла\_задача\_с\_подобрението\_при\_възможност}{производителност\_за\_цяла\_задача\_без\_подобрението}$$

Алтернативно,

$$Ускорение = \frac{Време\_за\_изпълнение\_на\_цяла\_задача\_с\_подобрението\_при\_възможност}{Време\_за\_изпълнение\_на\_цяла\_задача\_без\_подобрението}$$

Ускорението е отношението между времената за изпълнение:

$$Ускорение = \frac{1}{(1 - секция_{с\_подобрение}) + \frac{секция_{с\_подобрение}}{ускорение_{с\_подобрение}}}$$

Важен извод от закона на Amdahl е, че ако дадено подобрение може да се използва само за част от задача, тя не може да се ускори повече пъти от реципрочната стойност на (1 минус тази част)!

Законът на Amdahl може да се използва за прогнозиране на възможното ускорение, което може да се получи с дадено подобрение и как да се разпределят ресурсите за да се подобри отношението цена-производителност.

УРАВНЕНИЕ ЗА ПРОИЗВОДИТЕЛНОСТТА НА ЦП
The CPU Performance Equation

ЦП_време=

= ЦП_машинни_цикли_за_програмата
х
времетраенето_на_машинния_цикъл

или

ЦП_време = =
ЦП_машинни_цикли_за_програмата
/ тактова_честота

- ***Брой цикли за инструкция CPI***

(Clock cycles Per Instruction)

***броят цикли за инструкция* CPI =**

CPU цикли за програма / брой на инструкциите

- **Броят инструкции за цикъл IPC**

(Instructions Per Clock) е обратното на *броя цикли за инструкция* CPI

CPU време =

(IC x времетраенето на цикъла)/ тактовата честота

IC – instruction Count (брой инструкции)

- ***Броят на циклите може да се определи като IC x CPI !***

Циклите се наричат също ticks, clock ticks, clocks, cycles or clock cycles.

- Цикълът участва със своята продължителност (m.e., 1ns) или с честотата (e.g., 1GHz)

$$\frac{\text{инструкции}}{\text{програма}} x \frac{\text{машинни\_цикли}}{\text{инструкция}} x \frac{\text{секунди}}{\text{машинен\_цикъл}} =$$

$$= \frac{\text{секунди}}{\text{програма}} = CPU\_време$$

Параметрите за взаимозависими:

- Времетраенето на цикъла →*зависи от хардуерната технология и организацията*
- CPI →*организацията и ISA*
- IC →ISA & технологията на компилатора

$$CPU\_clock\_cycles = \sum_{i=1}^{n} IC_i \times CPI_i$$

Където IC_i представлява колко пъти инструкцията *i* е изпълнена в програмата и CPI_i представлява средния брой инструкции за цикъл за i-та инструкция.

$$CPU_{time} = \left(\sum_{i=1}^{n} IC_i x CPI_i\right) x Clock\_cycle\_time$$

Общият брой CPI е:

$$CPI = \frac{\sum_{i=1}^{n} IC_i x CPI_i}{брой\_инструкции} = \sum_{i=1}^{n} \frac{IC_i}{брой\_инструкции} x CPI_i$$

***CPI_i* трябва да бъде измерен (не само изчислен) т.к. трябва да бъдат включени ефектите от конвейеризацията, липсите в кеша, и всички други неефективности на системата памет!**

Измерване & моделиране на компонентите на CPU

уравнение на производителността

СИМУЛАЦИОННИ ТЕХНИКИ

3 ОБЩИ КЛАСА:

1. Статично моделиране на базата на профили
2. Симулация по следи
3. Симулация по изпълнението

СТАТИЧНО МОДЕЛИРАНЕ

Базирано на профили, най-евтино

Динамичен профил на изпълнението **на програмата, който показва колко често всяка инструкция се изпълнява, може да се получи по един от 3-те метода:**

1. ***Чрез използване на хардуерни броячи*** **в процесора, чието съдържание периодически се записва → *тази техника дава приблизителен профил*.**

Повечето съвременни процесори съдържат броячи както за изпълнените инструкции, така и за машинните цикли

2. ***Изпълнение с инструментален код***, при което инструменталният код е компилиран в програмата. Този код се използва за инкрементиране на броячите, дава точен профил.
3. ***Посредством интерпретиране на програмата на ниво инструкции***, като се компилира броя на инструкциите в процеса.

След като се получи профила, програмата се анализира статично посредством преглед на кода. **Получаваме :**

- ***Общият брой на инструкциите***
- ***Подробна динамична програмна смес***

(типове инструкции & тяхната честота)

- ***Може да бъде изчислен приблизително броят на циклите за инструкция CPI***

Тази техника е много бърза ***и се използва за моделиране на производителността на къси целочислени конвейери, защото поведението на системата памет се игнориа.***

Този модел налага жестоки ограничения при моделирането на сложни конвейери.

СИМУЛАЦИЯ ПО СЛЕДИ

Полезна при моделирането на производителността на системата памет!

Изисква *създаването на следи* за достъп до паметта.

Следата **включва кои инструкции са изпълнени (определени чрез адреса на инструкцията), както и адресите на данните, до които е осъществен достъп.**

- ***Най-често се използват за симулиране на системата на паметта, вкл. кешовете и хардуера за управление на паметта***
- ***За прости конвейеризирани процесори*** **– комбиниране на симулацията на системата на паметта със статичен анализ на производителността на конвейерите**
- ***Симулация на сложните конвейери на процесора*** **по следи**

Симулация на изпълнението

- *Най-точната & най-скъпата техника*
- *Едновременна симулация на системата на паметта & конвейерите на процесора*
- *Точно моделиране на взаимодействието между системата на паметта & конвейерите на процесора*

Симулация по работния товар

Избират се програми или мултипрограмни работни товари от стандартни еталонни комплекти или представителен товар за конкретната машина.

Проектирането и настройката на точни симулатори е скъпо, и симулациите консумират огромно изчислително време!!!

ПРОИЗВОДИТЕЛНОСТ НА СЪРВЪРИ ЗА ОБРАБОТКА НА ТРАНЗАКЦИИ

OLTP – online transaction processing

Стандартен индустриален еталон → TPC-C (разчита на система с база данни за queries & updates):

- ***Разумна апроксимация на реално OLTP приложение***
- ***Измерва общата системна производителност включително на хардуера, ОС, В/И система & системата с база данни***
- ***Търговците на КС са задължени да обявяват както производителността (TPM-transactions per minute)& цена/производителност (TPM per dollar) →значителни усилия TPC-C да се изпълняват бързо.***

Производителност на вградени процесори

- *Процесорите са проектиране за определен клас приложения → често не се измерва производителността извън приложния спектър*
- *Цена & консумирана мощност (критични ограничения, системи с батерии)* **често са най-важните фактори**
- **За разлика от desktop или сървърните системи,** *повечето вградени системи нямат вторична памет (цялото приложение се съдържа или във FLASH или в DRAM)*
- *Ограничения както от цената, така и от физическия размер* **(клетъчни телефони)**
- *Производителността се представя в "Dhrystone MIPS"*

Мерки за производителност

- MIPS (Millions of Instructions Per Second) – милиони инструкции за секунда
- MOPS (Millions of Operations Per Second) – милиони операции за секунда
- FLOPS (Floating Point Operations Per Second) – брой операции с плаваща запетая за секунда