

# Еталонни програми за оценка на производителността

---

проф. Пламенка Боровска

Факултет по Компютърни системи и управление

Катедра „Компютърни системи“

# Еталон (Бенчмарк)

---

- Изпълнението на компютърна програма, набор от такива или други операции с цел оценка на относителната производителност.
- Еталоните осигуряват метод за сравнение на производителността на множество подсистеми в различните архитектури.

# Защо еталони?

---

- Напоследък става все по-трудно да се сравни производителността на различните компютърни системи просто чрез техните спецификации.
- Процесор с по-малка тактова честота може да има същата производителност или дори по-голяма от тази на процесор с по-висока тактова честота.

# Митът за мегахерца

---

- ❑ Грешка е да се използва тактовата честота за сравнение на производителността на различни процесори.
- ❑ Тактовите честоти може да се използват за сравнение на производителността при различни скорости за един и същи модел и тип процесори.
  - Конвейерите и набора инструкции се отразяват в значителна степен върху производителността при различните архитектури.
- ❑ Даден процесор може да се нуждае от един цикъл да събере две числа и от още един за да ги умножи с трето, докато друг процесор може да направи всичко това за един цикъл.

# Производителност

---

- Намаляване на времето за отговор (времето за изпълнение) – времето между стартирането и завършването на дадена задача; общото време, необходимо на компютъра на завърши дадена задача, включително достъпите до диска и паметта, входно-изходните дейности, разходите на операционната система, времето за изпълнение от процесора и др.
- Увеличение на пропускателната способност – общото количество работа, извършена за определено време.

# Производителност

---

- Отношението между производителността и времето за изпълнение за даден компютър X е:

$$\text{Производителност X} = \frac{1}{\text{Време за изпълнение X}}$$

- Ако компютъра X е n пъти по-бърз от компютъра Y, тогава и времето за изпълнение на Y е n пъти по-дълго от това на X:

$$\frac{\text{Производителност X}}{\text{Производителност Y}} = \frac{\text{Време за изпълнение Y}}{\text{Време за изпълнение X}} = n$$

# Време за изпълнение

---

- Изминало време – общото време за изпълнение на дадена задача, включително достъпите до диска и паметта, входно-изходните дейности, разходите на операционната система и др.
- Процесорно време – времето, което процесора прекарва за изпълнение на задачата, като не включва времето за изчакване на входно-изходната система или изпълнението на други програми (времето за отговор = изминалото време, не процесорното време).
  - Потребителско процесорно време – процесорното време, прекарано в програмата.
  - Системно процесорно време – процесорното време, прекарано в операционната система за изпълнение на задачи свързани с програмата.

# Заблуди и капани

---

- MIPS се използва за точно измерване при сравнение на производителността на множество компютри.
  - Зависим от набора инструкции.
  - Варира измежду отделните програми на един и същи компютър, поради което не може да има един резултат за всички тях.
  - Може да варира обратнопропорционално на производителността.
  
- MFLOPS е последователен и полезен при измерването на производителността.
  - Зависим от конкретната машина и програма.
  - Неприложим извън измерването на производителността с плаваща запетая.
  - Наборът операции с плаваща запетая не е постоянен при различните машини.



# Видове еталони

---

## □ Реални програми

- Софтуер за текстова обработка
- CAD системи
- Потребителски софтуер

## □ Микроеталони

- Измерва се производителността чрез много малки специфични части от програмния код

## □ Ядра

- Съдържат цифрови кодове
- Обикновено взети от съществуваща програма
- Подходящи за фокусиране върху точно определен аспект
- Резултатите са в MFLOPS

## □ Компонентни

- Измерва производителността на основните компютърни компоненти

# Видове еталони

---

## □ Синтетични

- Процедура за програмиране на синтетичен еталон:
  - Статистика за всички видове операции от множество потребителски програми
  - Определяне съотношението на всяка една операция
  - Създаване на програма, базирана на по-горното съотношение

## □ Входно-изходни

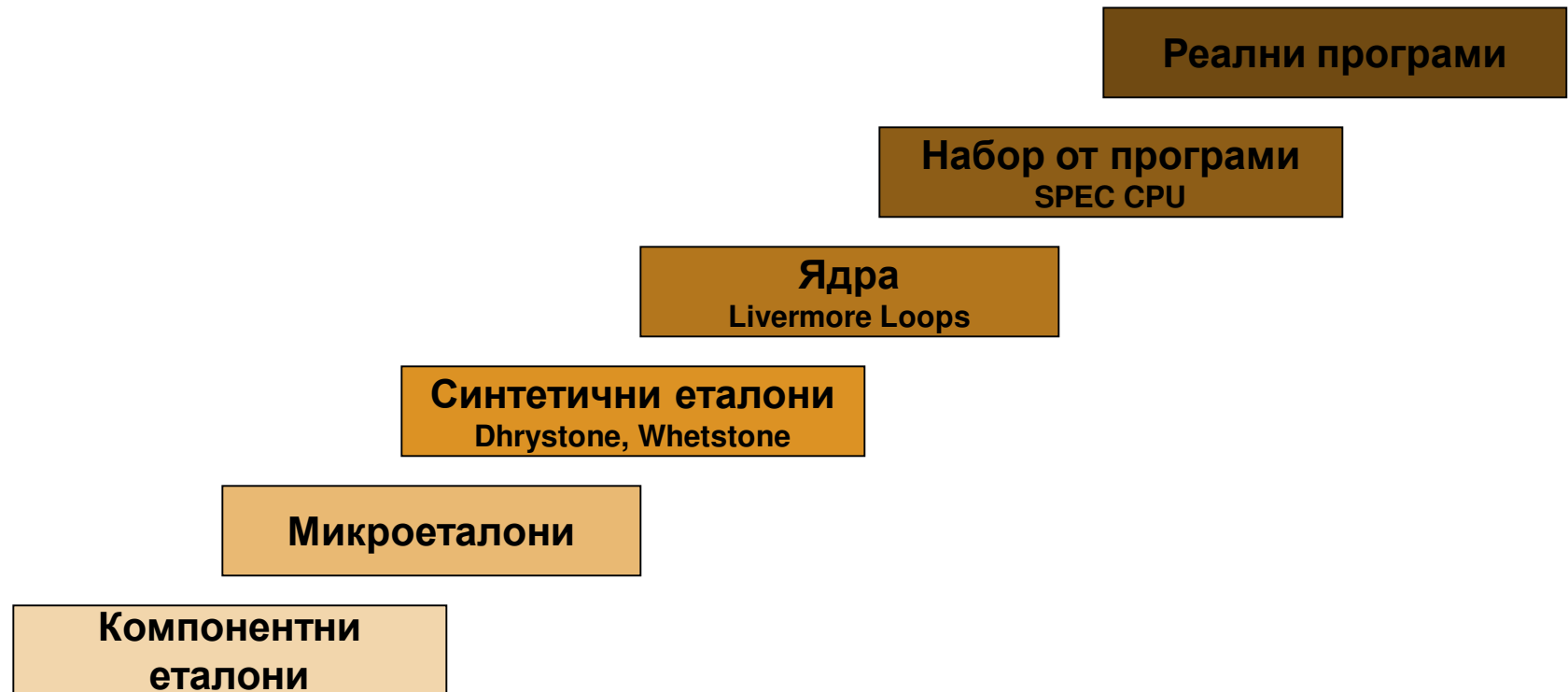
## □ Бази данни

- Измерва пропускателната способност и времето за отговор на системите за управление на бази данни

## □ Паралелни

- Използва се на машини с множество процесори или системи, изградени от множество машини

# Спектър еталони



← По-лесни за усъвършенстване

По-лесни за скалиране

По-лесни за поддръжка

По-малко показателни

→ По-трудни за усъвършенстване

По-трудни за скалиране

По-трудни за поддръжка

По-показателни

# Ядра: Плюсове и минуси

---

- + Взети от реални програми
- + Лесни за разбиране и анализиране
- + Лесно преносими – дори може да се  
конвертират в други езици
- + Работи на принципа бързо да се  
извърши общия случай
- Резултатите са като MIPS, ако  
програмата не е същата
- Не е представителен, ако програмата е  
сложна и има много различни части

# Синтетични еталони: Плюсове и минуси

---

- + По-реалистични от ядрата
- + Все още лесни за разбиране и анализиране
- + Все още лесно преносими
  
- Да се разчита на един еталон  
изкривява възприятието
- Никой не би се съгласил само на един еталон
- Лесно се злоупотребява – проектантите се фокусират върху подобряването на еталона, а не върху реалните приложения

# Реални приложения: Плюсове и минуси

---

- + По-близки до приложенията – по-реалистични
- + По-добри за откриване на слабостите в машината и точките, където има загуба на производителност
- По-трудно се сравняват различни машини, освен ако не се използва общоприет стандарт (напр. ANSI C)
- Трудно се определя защо определена програма работи по-бързо или слабо поради сложността
- Кой точно еталон? (Винаги може да се намери еталон, който да направи машината да изглежда най-добрата)
- Изисква се много време за симулацията (проблем за изследователите)

# Как да изберем правилния еталон?

---

- Необходимо е да си зададен няколко въпроса преди избора на даден еталон.
- Ще закупувам ли някоя от системите, които ще сравнявам?
- Опитвам ли се да подобря съществуващата система?
- Опитвам ли се да продам тази система?
- Търся ли насоки за развитие?
- Други.

# Отчет за производителността

---

- Трябва да съдържа всичко, което е необходимо за да се повтори точно същия експеримент на същата машина, но от друг човек, така че да се получат същите резултати.
- Когато се интерпретират получените резултати от дадена машина, изключително важно е да се знаят всички детайли за конфигурацията ѝ.
- Поради тази причина, отчета за производителността трябва да съдържа следните елементи:



# Отчет за производителността

---

- ❑ Производител на машината и модел.
- ❑ Производител на процесора, модел, ревизия и брой.
- ❑ Тактова честота на процесора.
- ❑ Архитектура на шината, големина и скорост.
- ❑ Размер на RAM паметта.
- ❑ Размер на първичните и вторичните кешове.
- ❑ Производител на дисковата памет, модел, размер и интерфейс.
- ❑ Производител на операционната система, версия и ревизия.
- ❑ Производител на компилатора, версия, ревизия и кои опции са използвани.

# Whetstone

---

- ❑ Първият съзнателно написан еталон за измерване на компютърната производителност.
- ❑ Симулира цифрови приложения с плаваща запетая.
- ❑ Съдържа висок процент данни и инструкции с плаваща запетая.
- ❑ Около 50% от времето за изпълнение се прекарва в математически изчисления.
- ❑ Болшинството променливи са глобални, поради което не се показват предимствата на архитектурите като RISC, където при по-голям брой процесорни регистри, се подобрява работата с локалните променливи.
- ❑ Съдържа малки цикли, поради което използването на малки инструкционни кешове би увеличило значително производителността.

# Dhrystone

---

- ❑ Създаден да тества факторите на производителността, важни при програмирането на нецифрови системи:
  - Операционни системи
  - Компилатори
  - Програми за текстова обработка
- ❑ Не съдържа операции с плаваща запетая.
- ❑ Значителна част от времето се прекарва в стрингови функции, като по този начин теста става силно зависим от начина на изпълнение на тези операции – чрез in-line код, програми написани на асемблер и др.
- ❑ Податлив за оптимизация на критичните части от кода от производителите.
- ❑ Използва се малко количество глобални данни (за разлика от Whetstone).

# Lawrence Livermore FORTRAN Kernels (Lawrence Livermore Loops)

---

- Набор от 24 отделни теста, извлечени от приложенията, които често се изпълняват в Националната лаборатория на САЩ Лоурънс Ливърмор.
- Оценяват аритметичната производителност на машините и техните компилатори.
- Получават се общо 72 резултата чрез времето изпълнение на 24-те ядра 3 пъти с различни дължини на цикъла DO.
- Отчетът от изпълнението дава 7 параметъра за производителността в MFLOPS – минималната, максималната и средната производителност, хармоничното и геометричното значение, стандартното отклонение и значещата точност.

\*\*\*\*\*  
THE LIVERMORE FORTRAN KERNELS: \* SUMMARY \*  
\*\*\*\*\*

---

Computer : CRAY-YMP C90 (240 MHz)  
System : UNICOS 7.C, loaded  
Compiler : CFT77 5.0.1.17  
Date : 92.02.18  
Testor : Charles Grassl, CRI

MFLOPS RANGE: REPORT ALL RANGE STATISTICS:

Mean DO Span = 167  
Code Samples = 72  
Maximum Rate = 826.0859 Mega-Flops/Sec.  
Average Rate = 190.5636 Mega-Flops/Sec.  
Geometric Mean= 86.2649 Mega-Flops/Sec.  
Median Q2 = 83.5138 Mega-Flops/Sec.  
Harmonic Mean = 40.7302 Mega-Flops/Sec.  
Minimum Rate = 6.7925 Mega-Flops/Sec.  
Mean Precision = 11.07 Decimal Digits

---

# Lawrence Livermore FORTRAN Kernels (Lawrence Livermore Loops)

---

- ❑ Kernel 1: an excerpt from a hydrodynamic code
- ❑ Kernel 2: an excerpt from an Incomplete Cholesky-Conjugate Gradient code
- ❑ Kernel 3: the standard Inner Product function of linear algebra
- ❑ Kernel 4: an excerpt from a Banded Linear Equations routine
- ❑ Kernel 5: an excerpt from a Tridiagonal Elimination routine
- ❑ Kernel 6: an example of a general linear recurrence equation
- ❑ Kernel 7: an Equation of State fragment
- ❑ Kernel 8: an excerpt of an Alternating Direction, Implicit Integration code
- ❑ Kernel 9: an Integrate Predictor code
- ❑ Kernel 10: a Difference Predictor code
- ❑ Kernel 11: a First Sum
- ❑ Kernel 12: a First Difference

# Lawrence Livermore FORTRAN Kernels (Lawrence Livermore Loops)

---

- ❑ Kernel 13: an excerpt from a 2-D Particle-in-Cell code.
- ❑ Kernel 14: an excerpt of a 1-D Particle-in-Cell code
- ❑ Kernel 15: a sample of how casually FORTRAN can be written
- ❑ Kernel 16: a search loop from a Monte Carlo code
- ❑ Kernel 17: an example of an implicit conditional computation
- ❑ Kernel 18: an excerpt from a 2-D Explicit Hydrodynamic code
- ❑ Kernel 19: a general Linear Recurrence Equation
- ❑ Kernel 20: an excerpt from a Discrete Ordinate Transport program
- ❑ Kernel 21: a matrix X matrix product calculation
- ❑ Kernel 22: a Planckian Distribution procedure
- ❑ Kernel 23: an excerpt from 2-D Implicit Hydrodynamics
- ❑ Kernel 24: finds the location of the first minimum in X

# LINPACK

---

- ❑ Създаден от съществуващо приложение, произлизащо от колекция подпрограми за линейна алгебра на Fortran.
- ❑ Тества производителността на операции с плаваща запетая, като не се използва деление.
- ❑ Не се използват математически функции (за разлика от Whetstone).
- ❑ Няма глобални променливи.
- ❑ Операциите се осъществяват чрез локални променливи или чрез масив, предаващ се на подпрограмите като параметър.
- ❑ Работи с двумерен масив, като когато се сравняват резултати трябва да се използва същия по големина масив.



# LINPACK

---

- ❑ Резултатите могат да бъдат с единична или двойна точност, като това се задава допълнително.
- ❑ Около 70% от времето за изпълнение се прекарва в една функция, като дори и малък инструкционен кеш може да промени значително резултатите.
- ❑ Еталонът разчита главно на пакета с основни подпрограми за линейна алгебра BLAS, който трябва да е кодиран на Fortran.
- ❑ Някои производители представят резултати, за чието получаване са се използвали същите библиотеки, но пренаписани на Асемблер, което може да доведе до съществени различия.

# NAS Parallel Benchmark

---

- ❑ NAS – Numerical Aerodynamic Simulation
- ❑ Създаден от NASA в изследователския център Еймс, посветен на напредъка на изчислителната аеродинамика.
- ❑ Първоначалната цел била да се докаже, че до 2000 г. ще съществува компютърна система, способна да симулира цялостно космическо транспортно средство за един до няколко часа.
- ❑ Първоначално съдържа 5 ядра и 3 псевдо-приложения.
- ❑ Днес съдържа общо 11 теста:

# NAS Parallel Benchmark

---

- ❑ IS – Integer Sort, random memory access
- ❑ EP – Embarrassingly Parallel
- ❑ CG – Conjugate Gradient, irregular memory access and communication
- ❑ MG – Multi-Grid on a sequence of meshes, long- and short-distance communication, memory intensive
- ❑ FT – Discrete 3D fast Fourier Transform, all-to-all communication
- ❑ BT – Block Tri-diagonal solver
- ❑ BT-IO – Test of different parallel I/O techniques
- ❑ SP – Scalar Penta-diagonal solver
- ❑ LU – Lower-Upper Gauss-Seidel solver
- ❑ UA – Unstructured Adaptive mesh, dynamic and irregular memory access
- ❑ DC – Data Cube
- ❑ DT – Data Traffic

# SPEC

---

- ❑ Standard Performance Evaluation Corporation – организация с идеална цел, създадена през 1988 г.
- ❑ Основната ѝ цел е да се създадат справедливи, безпристрастни и съдържателни еталони.
- ❑ Финансира се от своите членове – над 40 водещи производители на софтуер и хардуер.
- ❑ В състава ѝ влизат допълнително десетки асоциирани членове и изследователски групи от целия свят.
- ❑ Еталоните са написани на платформено неутрален език – C или Fortran.
- ❑ Еталоните могат да се получат единствено от SPEC на физически носител след заплащане на лиценз.
  - Забранено е качването на еталоните на общодостъпни сървъри.

# SPEC

---

- Резултатите от измерванията се публикуват в интернет сайта на SPEC.
- Кодът на еталоните може да се компилира от произволни компилатори, но самия код не може да се променя.
  - Много производители съзнателно оптимизират своите компилатори и/или системи за да подобрят своите резултати.
- Еталоните целят да симулират реални ситуации:
  - SPEC CPU тества производителността на процесора чрез измерване времето за изпълнение на няколко програми, като напр. компилатора GCC и програмата за шах Crafty.
  - SPEC WEB тества производителността на WEB сървъри чрез изпълнението на множество различни типове паралелни HTTP заявки.
  - Отделните задачи имат различна тежест в зависимост от тяхната значимост.
  - Отделните тежести се използват при изчислението на резултата от еталона.

# SPEC CPU 2006

---

- Фокусира се върху интензивността на изчислителната производителност, като набляга на:
  - Процесора
  - Архитектурата на паметта
  - Компилаторите
- Съдържа два компонента:
  - CINT2006 – измерва изчислителната интензивност на производителността при работа с цели числа
  - CFP2006 – измерва изчислителната интензивност на производителността при работа с числа с плаваща запетая
- Програмите в еталона са взети от съществуващи приложения, използвани от крайни потребители, а не са синтетични.

# SPEC CPU 2006

---

- ❑ Не се натоварват останалите компоненти на машината като мрежата, операционната система, графичното ускорение и входно-изходната система.
- ❑ При тестването на еднопроцесорни машини, ефектите върху производителността от тези компоненти обикновено са незначителни.
- ❑ При тестването на високопроизводителни машини, ефект върху производителността може да окажат стартираните процеси от операционната система, както и входно-изходната система – броя дискове, скоростта им и тяхната организация.

# CINT 2006

---

400.perlbench	C	PERL Programming Lang
401.bzip2	C	Compression
403.Gcc	C	C Compiler
429.Mcf	C	Combinatorial Optimization
445.Gobmk	C	Artificial Intelligence: go
456.Hmmer	C	Search Gene Sequence
458.Sjeng	C	Artificial Intelligence: chess
462.Libquantum	C	Physics: Quantum Computing
464.h264ref	C	Video Compression
471.Omnetpp	C++	Discrete Event Simulation
473.Astar	C++	Path-finding Algorithms
483.Xalancbmk	C++	XML Processing



# CFP 2006

---

410.Bwaves	Fortran	Fluid Dynamics
416.Gamess	Fortran	Quantum Chemistry
433.Milc	C	Physics: Quantum Chromodynamics
434.Zeusmp	Fortran	Physics/CFD
435.Gromacs	C/Fortran	Biochemistry/Molecular Dynamics
436.cactusADM	C/Fortran	Physics/General Relativity
437.leslie3d	Fortran	Fluid Dynamics
444.Namd	C++	Biology/Molecular Dynamics

---

# CFP 2006

---

447.dealII	C++	Finite Element Analysis
450.Soplex	C++	Linear Programming, Optimization
453.Povray	C++	Image Ray-tracing
454.Calculix	C/Fortran	Structural Mechanics
459.GemsFDTD	Fortran	Computational Electromagnetics
465.Tonto	Fortran	Quantum Chemistry
470.Lbm	C	Fluid Dynamics
481.Wrf	C/Fortran	Weather Prediction
482.sphinx3	C	Speech recognition

# Други SPEC еталони

---

- ❑ SPECparc – няколко широко използвани интензивни 3D приложения.
- ❑ SPEC HPC – високопроизводителни паралелни тестове, използващи приложения за квантова химия, моделиране на времето, откриване на промишлени находища на петрол и др.
- ❑ SPECjbb – измерва производителността на сървъри, изпълняващи бизнес приложения на Java.
- ❑ SPEC MPI – измерва MPI паралелната производителност на клъстери и SMP хардуер.
- ❑ SPEC OMP – измерва производителността на OpenMP приложения.
- ❑ SPECsfs – измерва производителността на файлови сървъри; използва скрип на заявки за файловия сървър; тества производителността както на централния процесор, така и на дисковата и входно-изходната системи.
- ❑ SPECviewperf – измерва производителността чрез OpenGL на 3D графични системи.

# TSP

---

- ❑ Transaction Processing Performance Council – организация с идеална цел, създадена през 1988 г.
- ❑ Създава еталони за обработка на транзакции и бази данни.
- ❑ Финансира се от своите членове.
- ❑ Два типа членство:
  - Пълноправно – в момента такава имат 19 водещи производители на софтуер и хардуер.
  - Асоциирано – в момента такава имат 4 фирми.
- ❑ Производителността се измерва в брой транзакции.
- ❑ Изискване за максимално допустимо време за отговор.
- ❑ Три действащи еталона – TSP-C, TSP-E, TSP-N.

# TSP-C

---

- ❑ Предлага множество различни типове транзакции, сложна база данни и цялостна изпълнима структура.
- ❑ Смес от 5 равностойни, но различни по тип и сложност транзакции.
- ❑ База данни, включваща 9 типа таблици с широк обхват и размер.
- ❑ Симулира цялостна изчислителна среда, в която определен брой потребители изпълняват транзакции в база данни.
- ❑ Резултатите са в брой транзакции за минута.

# TSP-E

---

- Използва база данни по модел на брокерска фирма с клиенти, които генерират транзакции, свързани с търговия, запитвания по сметки и маркетингови проучвания.
- Брокерската фирма взаимодейства с финансовите пазари за да изпълнява поръчки в полза на клиентите си и обновява информацията в сметките им.
- Мащабируем еталон – броя на клиентите на фирмата може да варира, така че да отразява натоварването при различни по големина бизнеси.
- Определя достатъчния микс от транзакции, които трябва да се изпълнят.
- Резултатите са в брой транзакции за секунда.

# TSP-N

---

- ❑ Еталон за вземане на решения.
- ❑ Съдържа набор от ориентирани към бизнеса заявки и едновременни промени на информацията.
- ❑ Илюстрира системите за подпомагане вземането на решения, които изследват големи масиви от данни, изпълняват запитвания с висока сложност и дават отговор на критични за бизнеса въпроси.
- ❑ Матрицата на ускорението отразява множество аспекти на способността на системата да обработва заявки.
- ❑ Резултатите са в транзакции за час.

# EEMBC

---

- ❑ Embedded Microprocessor Benchmark Consortium – организация с идеална цел, създадена през 1997 г.
- ❑ Разработва значими еталони за измерване на производителността на софтуера и хардуера във вградените системи.
- ❑ Състои се от над 40 водещи производители на софтуер и хардуер.
- ❑ Индустриален стандарт за оценка на възможностите на вградените микропроцесори, компилаторите и свързаните с вградената система реализации в съответствие с обективни и ясно определени критерии.



# EEMBC

---

- ❑ Симулират реалните приложения и изискванията, които вградените системи срещат в тези среди.
- ❑ Еталоните се разпространяват в пакети според изследваната сфера – телекомуникации, мрежи, дигитални развлечения, Java, автомобили/индустрия и офис продукти.
- ❑ Допълнителен пакет, който позволява да се следи енергията, консумирана от процесора, докато се изпълняват алгоритмите и приложенията.
- ❑ Има специфични еталони за многоядрени системи, които обхващат няколко области на приложение.

# Еталони за измерване на консумираната ел. енергия

---

- Измерва какво количество ел. енергия е консумирал даден компютър за изпълнението на дадена задача или набор от такива.
- Има 3 основни области, които представляват интерес при изследването на характеристиките на системи с ниска мощност, прилагащи техники на управление на захранването, имащи за цел постигането на ниска консумация на ел. енергия:
  - Първо - действителната консумация на ел. енергия при нормална работа и управление на захранването.
  - Второ – оперативността и използваемостта на системата в условията на управление на захранването; ясно е, че може да се постигнат забележителни резултати при пестенето на енергия, но това е за сметка на производителността на системата и времето за отговор.
  - Трето – отражението на техниките за управление на захранването върху надеждността на системата.

# Еталони за измерване на консумираната ел. енергия

---

- Подходящата стратегия при изследването на системи с управление на хранването трябва да вземе в предвид тези три области, така че да се оцени по достойнство цялата система и то без жертване на оперативността и надеждността на системата.
- Консумацията на ел. енергия се оценява най-добре докато се изпълняват едновременно множество задачи или значими често използвани приложения.
- Системата и отделните компоненти имат множество режими на работа в зависимост от това какво е моментното натоварване и очакваното такова в близко бъдеще.
- Ако даден компонент не се използва и не се очаква да се използва в близко бъдеще, с цел понижение на консумираната ел. енергия нивото му на готовност може дори да се компрометира. Възможно е дори и пълното изключване на този компонент.

# Компютърни игри

---

- ❑ Много компютърни игри, благодарение на тяхната взискателност към хардуера и зависимостта на скоростта на работа на играта от мощността на компютрите, успешно се използват като еталони.
- ❑ Стартира се предварително записан участък от играта и се измерва количеството кадри в секунда (FPS), което е способна да предаде тестваната система.
- ❑ Резултатите от тестовете с различни разделителни способности и настройки на качеството на картината може да се публикуват в Интернет според желанието на потребителя, на специално определени сайтове за различните игри.

# Компютърни игри

---

- 3DMark 2011
- 3DMark Vantage
- Aliens vs. Predator
- Battlefield 3
- Crysis 2
- Left 4 Dead 2
- Metro 2033: The Last Refuge
- StarCraft II: Wings of Liberty
- S.T.A.L.K.E.R.: Call of Pripyat
- Tom Clancy's H.A.W.X. 2
- Total War: Shogun 2
- Unigine Heaven Demo