

Performans Değerlendirmesi ve Anlaşılması

Giriş

Performans değerlendirmesinin amacı; Bir bilgisayar için aşağıda belirtilen sorulara cevap bulmaktır... Çünkü farklı tür bilgisayarlar arasında en iyi seçim yapabilmek için performans değerlendirmesi en iyi çözümdür.

Ayrıca yeni bir tasarım projesinde de en belirleyici kriter performans ölçümü ve değerlendirmesidir.

- Performans nasıl ölçülür?, raporlama ve özetleme nedir?
- Performansı hesaplamak için ana parametreler nelerdir?
- *Niçin bazı donanımlar farklı programlar için diğerlerinden daha iyidir?*
- *Sistem performansının hangi parametreleri donanımla alakalıdır?*
- *Komut seti, performansı nasıl etkiler?*

Performans

Airplane	Yolcu Sayısı	Uçuş menzili (miles)	Uçuş Hızı (m.p.h.)	yolcu taşıma hızı (Yolcu*m.p.h)
Boeing 777	375	4630	610	228,750
Boeing 747	470	4150	610	286,700
BAC/Sud Concorde	132	4000	1350	178,200
Douglas DC-8-50	146	8720	544	79,424

Bir bilgisayarın diğerinden daha iyi performansa sahip olması ne demektir? Bu soruya yolcu uçaklarına benzetme yaparak cevap verelim.

Tabloda tipik yolcu uçakları, seyir mesafeleri, seyir hızları ve kapasiteleri verilmiştir. Eğer bu tablodaki uçaklardan hangisinin en iyi performansa sahip olduğunu bulmak istersek ilk önce performans kelimesi ile ne kastettiğimizi bilmeliyiz. Örneğin en yüksek seyir hızına sahip olan uçak Concorde'dur, en uzun yolu DC-8 giderken en yüksek kapasiteye 747 sahiptir. (5.000 yolcu Newyork- Paris (1500km) arası taşımak için hangi uçağı seçersiniz?)

Performans, büyük oranda yapılacak işe bağlıdır.

Bireysel bir bilgisayar kullanıcısı olarak response (*cevap süresinin*) kısalmasını isteriz.

Datacenter yöneticileri ise genelde *birim zamanda üretilen veri miktarının* veya *bant genişliğinin* artmasını isterler.

Dolayısıyla çoğu zaman farklı performans metriklerine sahip oluruz.

Response time (Cevap Zamanı), Latency (Gecikme) veya Execution time (Elapsed - geçen süre -Yürütme zamanı – Yerine getirme zamanı- Çalışma zamanı)

Bir işi tamamlamak için harcanması gerekli toplam zamandır (Hafıza erişim zamanı, CPU execution time, I/O, v.b. gibi donanımlarla alakalıdır). Yani bir programın sonu ve başı arasındaki harcanan zamandır.

- İşimi çalıştırmak ne kadar sürer?

-Bir işi yürütmek ne kadar sürer?

-Veri tabanını sorgulamak için ne kadar beklemeliyim?

gibi sorular bu terimlerle ilgilidir.

- Response zamanını minimize etmek, performansı maximize etmektir.

- Throughput (Verimlilik – çıktı- Toplam iş bitirme hızı):

Verilen bir sabit zamanda yapılan toplam iş miktarıdır.

-Bir seferde kaç tane iş makinada çalıştırılabilir?

-Ortalama yürütme oranı nedir?

-Ne kadar iş yapıldı?

Gibi sorular bu terimle ilgilidir.

- Eğer yeni bir işlemci ile makinayı upgrade yaparsak neyi arttırmış oluruz? **Response zamanı düşer, verimlilik artar.**
- Eğer laboratuvara yeni bir makina eklersek neyi arttırmış oluruz? Hiçbir iş daha hızlı yapılmaz. **Sadece verimlilik (birim zamanda yapılan iş – Throughput) artar.**

CPU time (CPU execution time):

CPU'nun bir program için hesaplamaya harcadığı zamandır. Bu terim bekleme sürelerini veya giriş/çıkış işlemleri için harcanan zamanı içermez. Giriş çıkış sayısı veya başka programların harcadığı zaman değildir (program komutlarının ne kadar sürede işleneceğinin ölçüsüdür).

- Kullanıcı CPU zamanı: Bir program için CPU'nun harcadığı zamandır.
- Sistem CPU zamanı: işletim sisteminde görevleri yerine getirmek için *CPU 'nun harcadığı*
- Biz kullanıcı(user) CPU time' a odaklanacağız.

Program Performansını Anlamak

- Bir programın performansını geliştirmek için öncelikle performans ölçütlerini açıkça belirlemek ve bundan sonra performansı kısıtlayan darboğazlara bakmak gereklidir.
- Bir bilgisayar kullanıcısı açısından yalnızca süre ile ilgili endişeler olsa da, daha detaylı bakıldığında diğer performans ölçütleri de göz önünde bulundurulmalıdır. Özellikle bilgisayar tasarımcıları, donanımın basit fonksiyonları ne kadar hızlı gerçekleştirdiklerini belirleyen bir ölçüt isteyebilirler.
- Neredeyse bütün bilgisayarlar, olayların donanım tarafından ne zaman gerçekleştirileceğini belirleyen bir saat kullanmak üzere tasarlanmışlardır. Bu ayırık zaman gecikmeleri saat döngüsü (*darbe*, saat darbesi, saat periyodu, clock periyodu, çevrimler (cycles)) olarak isimlendirilir.

Verim ve execution time (yürütme zamanı) ile ilgili

- X ve Y makinalarında çalışan programlar için verim,

$$\text{Performance}_x = 1 / \text{Execution time}_x$$

$$\text{Performance}_x > \text{Performance}_y$$

$$1 / \text{Execution time}_x > 1 / \text{Execution time}_y$$

$$\text{Execution time}_y > \text{Execution time}_x$$

- X, Y den n kez daha hızlıdır.

$$\text{Performance}_x / \text{Performance}_y = n$$

$$\text{Execution time}_y / \text{Execution time}_x = n$$

Problem: A makinası bir programı 20 sn'de çalıştırıyor. Aynı programı B 25 saniyede çalıştırıyor. A, B'den ne kadar hızlıdır.

Çözüm:

$$\text{Performance}_A / \text{Performance}_B = \text{Execution time}_B / \text{Execution time}_A = n$$

Buna göre performans oranı = $25 / 20 = 1.25 = n$

$$\text{Performance}_A = \text{Performance}_B \times 1.25$$

Yürütme zamanı (Execution-Response time) yerine sık sık biz cycle terimini kullanır (h/w tasarımcının görüşü).

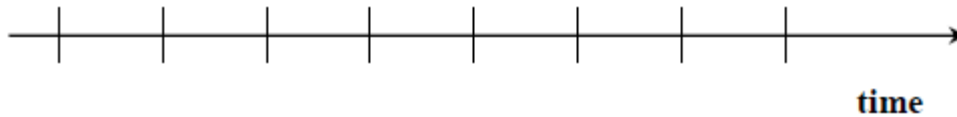
CPU execution time = CPU clock cycles sayısı x clock cycle time

(Bir program için)

(Bir program için)

$$Time = \frac{\text{seconds}}{\text{program}} = \frac{\text{clock cycles}}{\text{program}} \times \frac{\text{seconds}}{\text{clock cycle}}$$

- “Ticks” aktiviteler başladığı zamanki clock’ u gösterir.



- **cycle time (clock period) = time between ticks = seconds per cycle**
- **clock rate (clock frequency) = cycles per second = 1/ cycle time**

- (1 Hz. = 1 cycle/sec), 4 Ghz’ ‘lik clock’un cycle periyodu

$$\frac{1}{4 \times 10^9} \text{seconds} = \frac{1}{4 \times 10^9} \times 10^{12} = 250 \text{ picoseconds (ps)}$$

CPU zamanı

$$\frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \times \frac{\text{seconds}}{\text{cycle}}$$

- Performansı geliştirmek için yukarıdaki bağıntıya göre neler yapılabilir.?
 - 1- Bir program için gerekli Cycle'ların sayısının azaltılması
 - 2- Clock cycle time (Periyot) azlatmak
 - 3- Diğer ifadeyle clock rate (Frekans) arttırmak.

Örnek

- Programımızı A bilgisayarı 10 sn'de çalıştırıyor. A bilgisayarı 4 Ghz lik Clock'a sahiptir.

B bilgisayarı programı 6 sn de çalıştırıyor. B bilgisayarı A'nın clock cycles'ları sayısının 1.2 katına ihtiyaç duymaktadır.

B ne kadarlık bir clock zamanına (veya frekansına) sahiptir?

- Çözüm:

$$\text{CPU time}_A = \frac{\text{CPU clock cycles}_A}{\text{Clock rate}_A}$$

$$10 \text{ seconds} = \frac{\text{CPU clock cycles}_A}{4 \times 10^9 \frac{\text{cycles}}{\text{second}}}$$

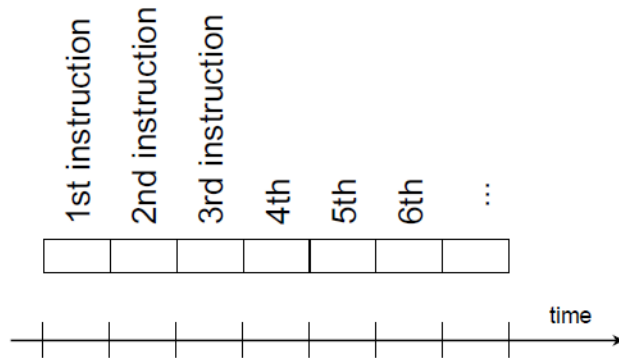
$$\text{CPU clock cycles}_A = 10 \text{ seconds} \times 4 \times 10^9 \frac{\text{cycles}}{\text{second}} = 40 \times 10^9 \text{ cycles}$$

$$\text{CPU time}_B = \frac{1.2 \times \text{CPU clock cycles}_A}{\text{Clock rate}_B}$$

$$6 \text{ seconds} = \frac{1.2 \times 40 \times 10^9 \text{ cycles}}{\text{Clock rate}_B}$$

$$\text{Clock rate}_B = \frac{1.2 \times 40 \times 10^9 \text{ cycles}}{6 \text{ seconds}} = \frac{8 \times 10^9 \text{ cycles}}{\text{second}} = 8 \text{ GHz}$$

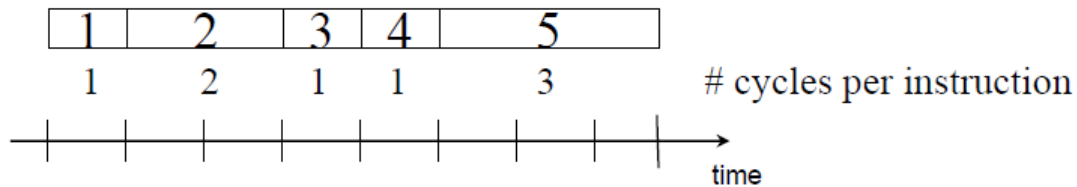
Komutların sayısını clock'ların sayısına eşit farz edebilir miyiz?



Bu düşünce yanlıştır. Çünkü farklı komutlar farklı makinalarda farklı zaman alırlar.

Niçin? Makine kodlarını hatırlayın!

- Çarpma toplamadan daha çok zaman alır.
- Floating point sayılarla işlem tamsayılarla işlemlerden daha uzundur.
- Hafızaya erişim registerlere erişimden daha uzun zaman alır.



Ortalama Clock Cycle'ı

Clock cycles per instruction (CPI); Bir program için bir komut başına clock cycle'ının ortalama sayısıdır. Buna göre bir programın harcayacağı clock sayısı:

$$\text{CPU Clock cycles} = \text{Programdaki komutların sayısı} \times \text{CPI}$$

- Bir programda **execution time** (yürütme zamanı) doğrudan doğruya komutların sayısına bağlıdır.
- Buna göre, bir programın yürütme zamanı hesabı için başka bir yolda; yürütülen komutların sayısı ile, her komut için harcanan ortalama zamanın çarpılmasıdır.

$$\text{CPU execution time} = \text{CPU clock cycles} \times \text{clock cycle time}$$

$$(\text{CPU yürütme zamanı} = \text{CPU clock sayısı} \times \text{clock periyodu})$$

Veya;

$$\text{CPU execution time} = \text{Programdaki komut sayısı} \times \text{CPI} \times \text{clock periyodu}$$

- Verilen bir program için
Bazı komutların sayısı (makine komutları)
Bazı cycle'lar sayıları
Bazı seconds sayıları gereklidir.
- Bu nicelikler ile ilgili kelimeler;
cycle zamanı (cycle başına saniye- periyot)
clock rate (saniye başına cycle -Frekans)
CPI (komut başına ortalama cycle'lar)
- *Yoğun floating point uygulamaları daha yüksek CPI gerektirir.*

- Performans, yürütme (execution) zamanı tarafından belirlenir.

Diğer değişkenlerin hangisi performansa eşittir.

- Yürütülen program için Cycle'lerin sayısı?
- Programda komutların sayısı?
- Saniye başına cycle'ların sayısı (Frekans)?
- Komut başına cycle'ların ortalama sayısı?(CPI)
- Saniye başına komutların ortalama sayısı?(MIPS)

Yaygın hata: değişkenlerin birini düşünmek performansı gösterir ancak gerçekçi değildir.

CPI aynı ISA (komut seti)' nin farklı gerçekleştirmelerini karşılaştırmak için iyi bir ölçümdür.

- Farzedin ki aynı komut seti mimarisi ile gerçekleştirilmiş iki farklı makineye sahibiz. (Ama programlama için farklı komutlar kullanılabilir)
- A makinasının Clock cycle süresi (Clock Periyodu) = 250 ps ve aynı program için 2.0 CPI ya sahiptir.
- B makinasının Clock cycle süresi (Clock periyodu) = 500 ps ve 1.2 CPI ya sahiptir.
- Bu program için hangi makine daha hızlıdır. Ne kadar hızlıdır?
- Eğer iki makine aynı ISA'ya sahipse (clock rate, CPI, yürütme zamanı, komutların sayısı, MIPS) daima benzer olacaklardır?

Çözüm:

- Verilere göre her iki bilgisayarda, aynı program için benzer sayıda komut işler. Bu sayıya **I** (Instruction sayısı) diyelim.
- Her bir bilgisayar için *clock cycle* (Clock periyot) sayısını bulalım.
 - **CPU clock cycles_A = I x 2.0**
 - **CPU clock cycles_B = I x 1.2**

Buradan her bir bilgisayar için CPU time'ı hesaplayabiliriz.

$$\begin{aligned}\text{CPU time}_A &= \text{CPU clock cycles}_A \times \text{Clock cycles time}_A \\ &= I \times 2.0 \times 250 \text{ ps} = I \times 500 \text{ ps}\end{aligned}$$

$$\begin{aligned}\text{CPU time}_B &= \text{CPU clock cycles}_B \times \text{Clock cycles time}_B \\ &= I \times 1.2 \times 500 \text{ ps} = I \times 600 \text{ ps}\end{aligned}$$

Anlaşılabacağı gibi A bilgisayarı daha hızlıdır.

$$\text{Performance}_A / \text{Performance}_B = \text{Execution time}_B / \text{Execution time}_A = n$$

$$= I \times 600 \text{ ps} / I \times 500 \text{ ps} = 1.2$$

$$\text{CPU time} = \text{komut sayısı} \times \text{CPI} \times \text{Clock cycle (Clok periyodu) time}$$

Kod Segmentlerinin karşılaştırılması

- Bir derleyici tasarımcısı iki kod dizileri arasında karar vermeyi deniyor. Donanım aşağıdaki mimarileri desteklemektedir.
- 3 farklı komut sınıfı vardır. A, 1 cycle, B, 2 cycle, C 3 cycle ihtiyaç duyuyor.
- İlk kod dizisi 5 komuta,(A=2, B=1,ve C=2)
- İkinci kod dizisi 6 komuta sahiptir. (A=4, B=1 ve C=1.)

Hangi size göre daha hızlıdır. Ne kadar?

Herbir dizi için CPI?

	CPI for this instruction class		
	A	B	C
CPI	1	2	3

Code sequence	Instruction counts for instruction class		
	A	B	C
1	2	1	2
2	4	1	1

Çözüm:

- 1.kod dizisi (2 + 1 + 2 = 5 komut)
- 2.kod dizisi (4 + 1 + 1 = 6 komut) kullanır.

$$\text{CPU clock cycles} = \sum_{i=1}^n (\text{CPI}_i \times C_i),$$

- n komut sınıflarının sayısı, C_i i. Sınıfın komutlarının sayısı ise yukarıdaki formüle göre, her iki komut dizisine göre CPU clock cycles'ı (Clock periyodu sayısını) bulabiliriz.

$$\text{CPU clock cycles}_1 = (2 \times 1) + (1 \times 2) + (2 \times 3) = 2 + 2 + 6 = 10 \text{ cycles}$$

$$\text{CPU clock cycles}_2 = (4 \times 1) + (1 \times 2) + (1 \times 3) = 4 + 2 + 3 = 9 \text{ cycles}$$

Buna göre 2.kod dizisi daha hızlı olsa bile fazladan 1 komut işleyecektir. Dolayısıyla daha düşük bir CPI'ye sahip olmalıdır.

$$\text{CPI} = \frac{\text{CPU clock cycles}}{\text{Instruction count}}$$

$$\text{CPI}_1 = \frac{\text{CPU clock cycles}_1}{\text{Instruction count}_1} = \frac{10}{5} = 2$$

$$\text{CPI}_2 = \frac{\text{CPU clock cycles}_2}{\text{Instruction count}_2} = \frac{9}{6} = 1.5$$

Örneğe devam:

- Bu örnek gösterir ki; performans değerlendirme için yalnızca bir kıstas (Burada komut sayısı kullanılsaydı) kullanmak yanıltıcı olabilir.
- İki bilgisayarı performans açısından karşılaştırmak için, (yürütme zamanı- execution time) her üç bileşeni (Komut sayısı, CPI, Clock Cycle time (periyot)) dikkate almak gerekir.
- Eğer bileşenlerden bir kısmı özdeş ise, performans karşılaştırması özdeş olmayan bileşenlere göre yapılır. Örneğin her iki bilgisayarın frekansları aynı ise karşılaştırma; komut sayısı ve CPI'ya göre yapılabilir.
- Performans'a ve CPI'ye etki eden en büyük faktör ise PIPELINE mimarisidir ki bu konu daha ileride detaylandırılacaktır.

Performans değerlendirilmesi

- İki veya daha fazla bilgisayar arasındaki performans değerlendirmesini nasıl yaparız.?
- İki bilgisayarın performansını değerlendirmek için; bu iki bilgisayarda *bir workload '(tanımlı bir işlemler kümesi programı)'un* yürütme (execution) zamanları karşılaştırılır.
- **Benchmark** “ Bir ölçme veya değerlendirme standardı "dır. Bir bilgisayar benchmarkı; tipik bir şekilde tanımlı **workload**'u gerçekleştiren bilgisayar programıdır.
- Bilgisayar benchmark için ölçülen metrikler;
Hız: Bir workload'ın ne kadar hızlı tamamlandığı.
Throughput: Bir birim zamanda tamamlanan workload birimi'dir.
- Birden fazla bilgisayarda aynı bilgisayar benchmarkı çalıştırılarak bir karşılaştırma yapılmasını sağlar.

- Performans en iyi gerçek bir uygulamanın çalıştırılmasıyla belirlenir.
- Küçük kriterler
 - Mimarlar ve tasarımcılar için güzel
 - Standardize etmek kolay
 - Kötüye kullanılabilir.
- **SPEC (System Performance Evaluation Cooperative)** şirketler gerçek program ve giriş kümeleri hakkında anlaşmaya varmışlardır.
 - CPU- yoğunluklu kriterleri kullanarak sunucular ve iş istasyonları üzerinde duruluyor.
- Şimdilerde CPU performansı, grafik, yüksek performanslı hesaplama, nesneye yönelik hesaplama, Java uygulamaları, client server modelleri, mail sistemleri , Dosya sistemleri ve web serverler. İçeren performans testleri önem kazanmıştır.

Performans Raporu

- Hazırlıktan sonra performans ölçümleri seçilen uygun kriterler kümesinden elde edilir.
- **Tekrarlanabilirlik** :rapordaki diğer herşey dahil, deneyler rapor edilen sonuçları çoğaltmak için tekrarlanabilirlik gerekir.
(O/S version, derleyici, giriş, bilgisayar konfigürasyonu, gibi)
- **Girişlerin seçimi** çok önemlidir. Büyük girişler bellek sistemini değerlendirmek için gereklidir. Ayrıca büyük iş yükü gereklidir.

- Kriterlerin bir grubunun performansını özetlemek için nasıl karar verildiği kritiktir.
- Özet hesaplama nasıl yapılmalıdır.
- Aşağıdaki örneği göz önüne alınız.

Program	Computer A execution time	Computer B execution time
1	1 sec	10 secs
2	1000 secs	100 secs
Total	1001 secs	110 secs

- Program 1 için A, B'den 10 kat daha hızlıdır.
- Program 2 için B, A'dan 10 kat daha hızlıdır.
- Toplam sonucu nedir?

- Sürekli özet ölçümü
- Böylece bağıl performansı özetlemek için 2 programın toplam yürütme zamanını kullanabiliriz.

$$\frac{\text{Performance}_B}{\text{Performance}_A} = \frac{\text{Execution Time}_A}{\text{Execution Time}_B} = \frac{1001}{110} = 9.1$$

- Böylece B, A'dan 9.1 kat daha hızlıdır.
- Yürütme zamanının ortalaması, workloaddaki programların herbirinin n defa çalıştırılarak elde edilen toplam yürütme zamanına oranıdır. Bu ise aritmetik ortalama (**Arithmetic Mean**) dir.

$$AM = \frac{1}{n} \sum_{i=1}^n \text{Execution Time}_i$$

- Farzedinki 4 farklı desktop seçiyorsunuz. (apple Macintosh, Pentium 4, AMD, Pentium 5)
- En hızlı bilgisayar en yüksek clock rate (Clock Frekansa) sahip olan olacaktır.
- Bütün bilgisayarlar aynı komut setini kullandığı için(intel ISA) komutların sayısı aynıdır. En hızlısı en yüksek *clock frekansına* sahip olan olacaktır.
- AMD işlemci farklı CPI'lara sahip olabilir. Ancak biz clock frekansına bakarak iki pentium işlemcinin daha hızlı olduğunu söyleyebiliriz.
- Benzer işler için kriterlerin sonuçlarına bakarak tam bir performans elde edebiliriz.

- Aşağıdakilerden hangisi doğrudur.
- Program 1 için A B'den daha hızlıdır.
- Program 2 için A B'dan daha hızlıdır.
- Program 1 ve 2'nin yürütmelerinin sayısı eşit workload için A B'den daha hızlıdır.
- Program 2'nin program 1'den 2 kat daha workload için A B'den daha hızlıdır.

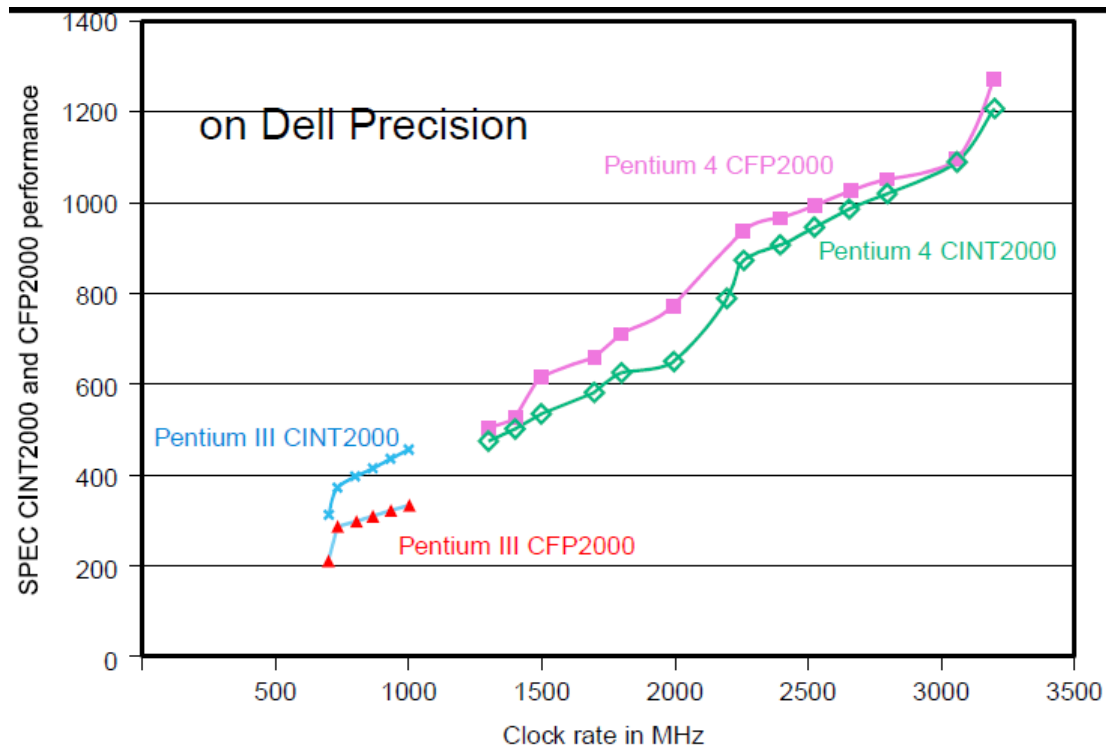
Program	Computer A execution time	Computer B execution time
1	2 secs	4 secs
2	5 secs	2 secs

- 12 integer 14 floating point program
- Herbir grup için farklı özetler.
- Clock rapor edilir ancak Çoğunlukla harcanan zaman CPU'dadır. CPU performansı ölçülür.

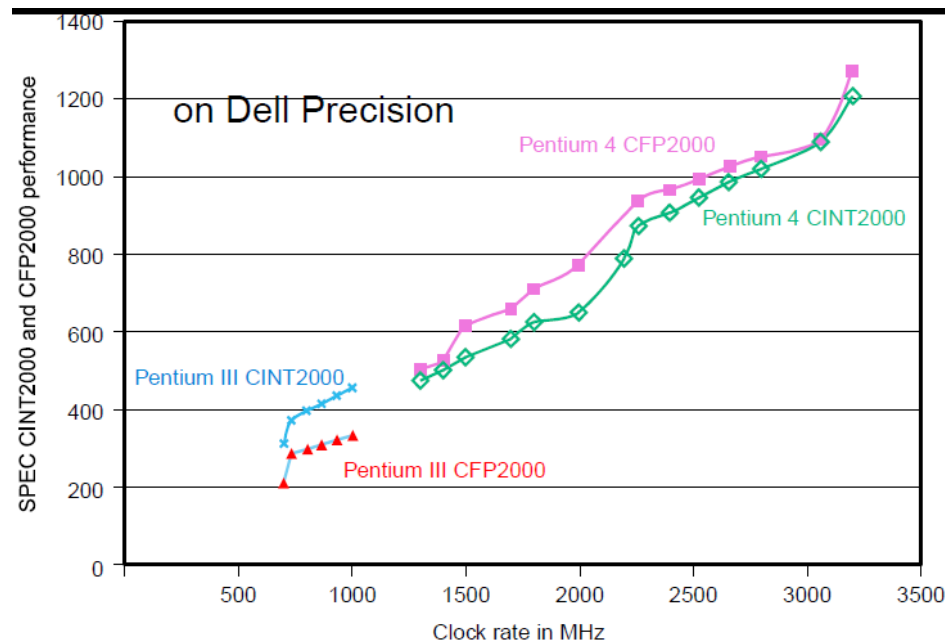
Integer benchmarks		FP benchmarks	
Name	Description	Name	Type
gzip	Compression	wupwise	Quantum chromodynamics
vpr	FPGA circuit placement and routing	swim	Shallow water model
gcc	The Gnu C compiler	mgrid	Multigrid solver in 3-D potential field
mcf	Combinatorial optimization	applu	Parabolic/elliptic partial differential equation
crafty	Chess program	mesa	Three-dimensional graphics library
parser	Word processing program	galgel	Computational fluid dynamics
eon	Computer visualization	art	Image recognition using neural networks
perlbnk	perl application	equake	Seismic wave propagation simulation
gap	Group theory, Interpreter	facerec	Image recognition of faces
vortex	Object-oriented database	ammp	Computational chemistry
bzip2	Compression	lucas	Primality testing
twolf	Place and rote simulator	fma3d	Crash simulation using finite-element method
		sixtrack	High-energy nuclear physics accelerator design
		apsl	Meteorology: pollutant distribution

FIGURE 4.5 The SPEC CPU2000 benchmarks. The 12 integer benchmarks in the left half of the table are written in C and C++, while the floating-point benchmarks in the right half are written in Fortran (77 or 90) and C. For more information on SPEC and on the SPEC benchmarks, see www.spec.org. The SPEC CPU benchmarks use wall clock time as the metric, but because there is little I/O, they measure CPU performance.

- Performans skalası clock rate ile hemen hemen lineerdir.
- Diğer gözlemler?



- Bir clock rate tarafından bölünen CFP2000 ve CINT2000'in ortalama değerlerine bakınız.
- $\text{CINT2000/Clock rate Pentium III} = 0.47$ $\text{Pentium 4} = 0.36$
- $\text{CFP2000/Clock rate Pentium III} = 0.34$ $\text{Pentium 4} = 0.39$



SPECweb99

- Verime odaklanır. Bir web server bağlantılarının maksimum sayısının ölçümünü destekleyebilir.
- Çoklu işlemciler sık sık kullanılır.
- Web server s/w sistem ölçümlerinin bir bölümüdür.
- SPECweb99 performansı bir çok sistem karakteristiğine bağlıdır. (disk sürücülerinin sayısı, CPU'ların sayısı, ağların sayısı, clock rate)
- Örnek: Pentium III Xeon w/ 7 disk drives, 8 CPUs, 8 networks, at 0.7GHz, Pentium Xeon 4 w/ 5 disk drives, 2 CPUs, 4 networks, at 3.06GHz 'dan çok daha hızlıdır.

- Güç tüketimi , işlemci performansı için anahtar sınırlama oluyor.
- Batarya güçleri yavaş geliyor
- İşlemciler etkili işletilebilmek için tasarlanmalıdır. Farklı clock rate'ler arasında anahtarlatabilmeli ve güç korunabilmelidir.
- 3 işletim modu varsayılır.

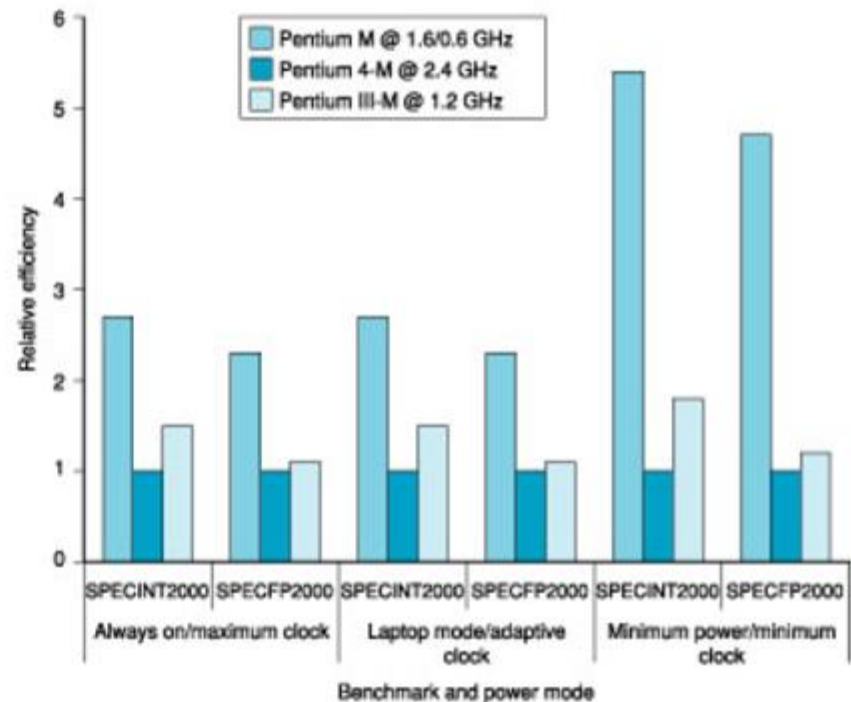
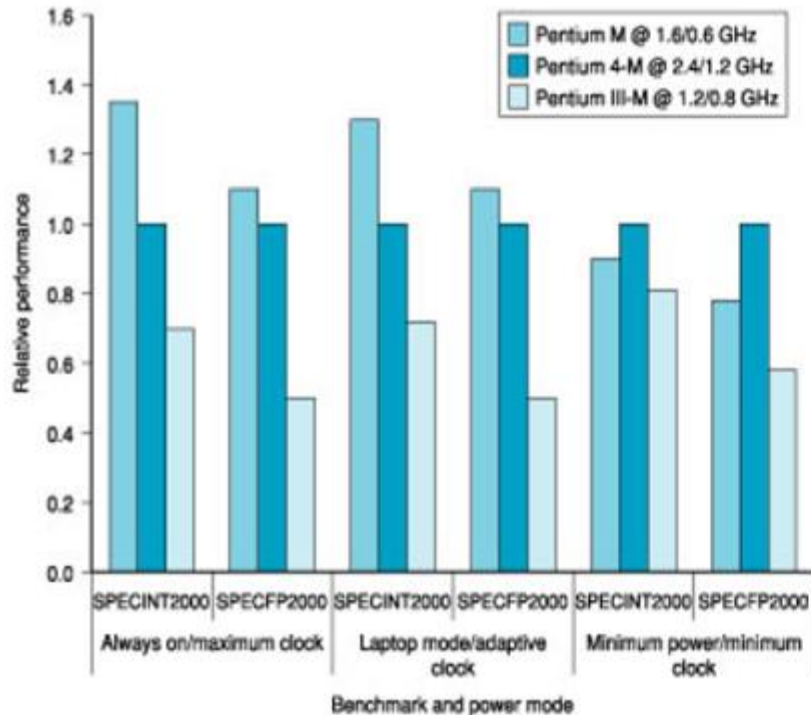
Maximum güç: maximum clock rate → en iyi performans

Laptop mode: adaptif clock rate

Minimum güç: minimum clock rate → en iyi güç verimliliği

- Enerji verimliliği: kriterler çalışırken performans ortalama güç tüketimi tarafından bölünür.

- Hangi makine daha düşük clock rate ile daha yüksek performansa sahiptir.



- Performans özel programlara özgüdür.
- Toplam yürütme zamanı performansın tutarlı bir özetidir.
- Mimari performansı arttırmak için; Clock rate'yi arttırmak
İşlemci organizasyonunu geliştirmek, düşük CPI
Compiler geliştirme, düşük CPI ve düşük komut sayısı
Komut sayısını etkileyecek algoritma/dil seçmek
- Birçok tuzak,
Makine performansının bir yönünü geliştirmeyi ummak
toplam performansı etkiler. (Amdahl kanunu)
Performans metriği olarak bir alt performans denklemi
kullanılarak (MIPS ölçümleri)

Örnek; bir makinada, bir program 100 saniyede çalıştığını farzedelim. Bu zamanın 80 sn'si çarpma içindir. Biz programı 2 kat daha hızlı çalıştırmak istersek çarpma hızını ne kadar çok geliştirmeliyiz.?

- Programı, 5 kat daha hızlı nasıl yapabiliriz.?

Amdahl kanunu

$$\begin{array}{lcl} \text{Execution Time} & = & \text{Execution Time} + \frac{\text{Execution Time Affected}}{\text{Amount of Improvement}} \\ \text{After Improvement} & & \text{Unaffected} \\ \text{(İyileştirmeden sonra)} & & \text{(etkilenmemiş değer)} \end{array}$$

$$50\text{sn} = (100 - 80 \text{ sn}) + 80\text{sn}/n$$

$$50 - 20 = 80 / n \quad = 30 \quad n = 80$$

$$n = 80/30 = 2.666 \text{ kadar çarpma hızını geliştirmeliyiz.}$$

- MIPS = Million Instructions Per Second
- Anlamak kolaydır,belli varsayımlar altında performans belirlemekte kullanılır.
- “whole picture” vermediği için Çoğu kez yanlış olabilir.
- Yürütme oranı tanımlar ancak komut yetenekleri görünmez
- Farklı ISA’lar ile bilgisayar karşılaştırmada kullanılamaz. Niçin?

- İki farklı compiler 4 Ghz lik bir makinada 3 farklı komut sınıfı için test ediliyor. Class A, Class B, ve C Class, sırasıyla 1, 2, ve 3 cycle gereklidir.
- Her iki compiler yazılımın büyük bir parçası için kod üretmede kullanılır.
- İlk compilerin kodu 5 million Class A komut, 1 million Class B komut, and 1 million Class C komut kullanıyor.
- İkinci compiler kodu 10 million Class A komut, 1 million Class B komut, ve 1 million Class C komut kullanıyor.
- Hangi dizi MIPS'e göre daha hızlıdır.
- Yürütme zamanına göre hangi dizi daha hızlıdır.