

BSM 420 – BİLGİSAYAR SİSTEMLERİNİN PERFORMANS DEĞERLENDİRMESİ

2.Hafta: Performans Değerlendirmesi Nedir?

Giriş

- Neden performans değerlendirilir?
- Performans değerlendirme Ölçütleri/Metrikleri
- Ölçütlerin uygulama alanlarıyla ilişkisi
- Performans değerlendirme teknikleri
- Performans değerlendirme tekniğini seçme
- Performans değerlendirmeye sistematik bir yaklaşım

Hedef

Performans Analizi = Analiz + Bilgisayar Sistemleri

Performans Analisti = Matematikçi + Bilgisayarcı

Amaç:

- Performans gereksinimlerinin belirlenmesi
- Tasarım alternatiflerinin değerlendirilmesi
- İki veya daha fazla sistemin karşılaştırılması
- Bir parametrenin optimum değerinin belirlenmesi (sistem ayarı)
- Performans darboğazının bulunması (darboğaz tespiti)
- Sistem üzerindeki yükün karakterizasyonu (iş yükü karakterizasyonu)
- Bileşenlerin sayısının ve boyutlarının belirlenmesi (kapasite planlaması)
- Gelecekte muhtemel yüklerdeki performansın tahmin edilmesi

Neden performans değerlendirilir?

- Sistemin amaçlanan uygulamalara uygun olduğundan emin olmak için,
- Sistemin belirli verimlilik ve güvenilirlik gereksinimlerini karşıladığından emin olmak için
- Belirli kaynak (zaman, bütçe) kısıtları altında sistemi optimum işlem gücü seviyesine yakın bir yerde tasarlamak/inşa etmek/işletmek
 - Sonuç: Sistem yeterli performansa ve uygun maliyete sahiptir

Performans değerlendirme ölçümleri

- **Duyarlılık(Responsiveness):** Belirli bir görevin sistem tarafından ne kadar hızlı gerçekleştirilebileceği
 - **Yanıt süresi:** bir isteği yerine getirme süresi; bir bilgisayar sistemindeki bir sorgunun ya da talebin sonu ile yanıtın başlangıcı arasında geçen süre (IBM Bilgi İşlem Sözlüğü)
 - Waiting time, processing time
- **Kullanım oranı:** sistemin çeşitli bileşenlerinin ne kadar iyi kullanıldığı
 - **Kullanım oranı (utilization):** Donanımın meşgul olduğu sürenin yüzdesini gösterir. En yüksek kullanım oranına sahip kaynak darboğazdır; Bu kaynaktaki performans iyileştirmesi en yüksek kazancı sağlar
- **Verimlilik:** Bir kullanıcının işi ne kadar etkili bir şekilde tamamlayabileceği
 - **Performans çıkışı (throughput):** birim zamanda tamamlanan iş miktarı

Performans değerlendirme ölçümleri

- **Görev Verebilirlik:** Sistemin bir görev süresince sürekli olarak çalışır durumda kalıp kalmayacağını gösterir
 - Güvenilirlik: Sistemin görev süresi boyunca doğru performans gösterme olasılığı
 - Onarım/ayarlama pratik olmadığında veya arıza davranışı felaket olduğunda kullanışlıdır
- **Güvenilirlik:** Sistemin uzun vadede ne kadar güvenilir olduğu
 - Arıza sayısı/gün, MTTF, MTTR, uzun süreli erişim
 - Onarımların mümkün olduğu ve arızaların tolere edilebilir olduğu durumlarda kullanışlıdır

Ölçülerin Uygulama Etki Alanlarıyla İlişkisi

- **Genel amaçlı hesaplama**

- Kelime işlemci
- Duyarlılık/Yanıt verebilirlik, kullanım oranı, verimlilik

- **Yüksek kullanılabilirlik**

- Görev işleme
- Yanıt verebilirlik, güvenilirlik, verimlilik

- **Gerçek zamanlı kontrol**

- Zaman kısıtlamaları
- Yüksek düzeyde yanıt verebilirlik, güvenilirlik

Ölçülerin Uygulama Etki Alanlarıyla İlişkisi

■ Görev odaklı

- Görev sırasında çok az onarım mümkün/hiç mümkün değildir, örneğin uçaklar
- Yüksek güvenilirlik, yanıt verebilirlik

■ Uzun ömür

- Akıllı yerleşik teşhis ve onarım tesisine sahip insansız uzay gemileri
- Son derece güvenilir, yanıt verebilirlik

Performans Ölçüm Teknikleri

■ Ölçme

- Deney tasarlama
- Performans parametrelerinin ölçülmesi
- Anlamlı sonuçlar çıkarmak için istatistiksel tekniklerle analiz edilen veri

■ Analitik modelleme

- Matematiksel bir model oluşturma (Kuyruk, Markov zinciri, Petri ağları, vb.)
- Daha sonra da modeli çözme

■ Simülasyon modelleme

- Bir sistem davranış modeli oluşturma
- İş yükünün uygun bir soyutlamasını girdi olarak
- Deney tasarımı, veri toplama ve analizini içerir

Performans Ölçüm Tekniği Seçme

Criterion	Measurements	Analytical Modeling	Simulation
Life-cycle stage	Post-prototype	Any	Any
Time required	Varies	Small	Medium
Tools	Instrumentation	Analysts	Computer Languages
Accuracy	Varies	Low	Moderate
Trade-off analysis	Difficult	Easy	Moderate
Cost	High	Small	Medium
Salability	High	Low	Medium

İki veya daha fazla teknik kullanılabilir

- aynı anda her birinin sonuçlarını onaylamak ve doğrulamak için
- farklı değerlendirme aşamalarında sırayla

Sistematik Yaklaşım

1. Çalışmanın amaçlarını belirt ve sistemi tanımla
2. Sistem servislerini ve sonuçlarını listele
3. Performans metriklerini seç
4. Performansı etkileyen parametreleri listele
5. Çalışılacak faktörleri (çalışma sırasında değişen parametreler) ve değerlerini seç
6. Değerlendirme tekniğini seç
7. İş yükünü seç (sistemi sıkıştıran/zorlayan şey, gerçek hayatta sistem kullanımını temsil etmelidir)
8. Deneyler tasarla
9. Verileri analiz et ve yorumla
10. Sonuçları sun. Gerekirse baştan başla

Performans Ölçümü

İki temel metrik:

- duvar saati zamanı (bir programın cevap zamanı)
- çıkış (birim zamanda yerine getirilen görevler)

** çıkışı optimize etmek için, asgari kaynak israfı olmalıdır

Performans (Başarım)

- Bir işi yaparken geçen süre:
 - çalışma zamanı, cevap zamanı, gecikme vs.
- Bir günde, saatte, dakikada, saniyede yapılan iş
 - çıkış(throughput), band genişliği vs.

Plane	DC to Paris	Speed	Passengers	Throughput (pmp)
Boeing 747	6.5 hours	610 mph	470	286,700
BAD/Sud Concode	3 hours	1350 mph	132	178,200

Bilgisayar Performansı: zaman

- *Cevap Zamanı (Response Time, elapsed time, latency):*
 - Bir görevin çalışması için ne kadar süre gerekli?
 - Başlangıçtan sona kadar tamamlanması için ne kadar süre gerekli?
 - Bir veritabanı sorgusu için ne kadar süre beklenmeli?
- *Performans çıkışı(Throughput):*
 - Tek seferde makine ne kadar işi çalıştırabilir?
 - Ortalama çalışma oranı ne?
 - Bir işin tamamlanabilmesi için ne kadar çalışma gerekli ?
- *Bir makineyi yeni bir işlemci ile yükselttiğimiz zaman ne kadar bir iyileşme sağlanır?*
- *Bir laba yeni bir makine konduğunda ne kadar iyileşme olur?*

Çalışma Zamanı (Execution Time)

- *Geçen Süre (Elapsed Time)*
 - *Baştan sona herşeyi dikkate alır (disk bellek erişimleri, I/O beklemeleri, diğer programların çalışması vs.)*
 $\text{geçen süre} = \text{CPU zamanı} + \text{bekleme zamanı (I/O, diğer programlar vs.)}$
- *CPU zamanı*
 - *I/O beklemesi ve diğer programların çalışması için geçen zamanı dikkate almaz*
 - *Kullanıcı CPU zamanı ve sistem CPU zamanı (OS çağrıları) olarak ikiye ayrılır*

$\text{CPU zamanı} = \text{kullanıcı CPU zamanı} + \text{sistem CPU zamanı}$

$\Rightarrow \text{geçen süre} = \text{kullanıcı CPU zamanı} + \text{sistem CPU zamanı} + \text{bekleme zamanı}$

- *kullanıcı CPU zamanı dikkate alınır (CPU çalışma zamanı veya sadece çalışma zamanı)*
 - *Program içindeki kod satırlarının çalışması için geçen süre*

Performans Tanımı

- X makinesi üzerinde çalışan bir program için:

$$\text{Performans}_X = 1 / \text{Çalışma Zamanı}_X$$

- *X makinesi Y makinesinden n kat hızlıdır:*

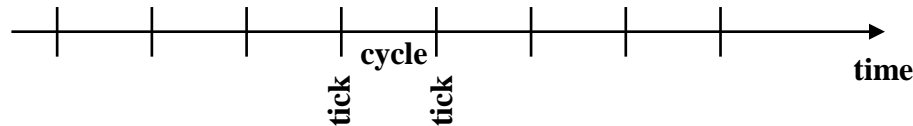
$$\text{Performance}_X / \text{Performance}_Y = n$$

Saat Çevrimleri

- Çalışma zamanını saniyeler cinsinden sunmak yerine, çoğu kez saat çevrimlerini kullanırız.
- Modern bilgisayarlarda, olaylar çevrim çevrim ilerler : yani çarpma, toplama gibi her olay bir dizi çevrimde gerçekleşir.

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

- *Saat tiklemeleri çevrimin başı ve sonunu gösterir:*



- *Çevrim zamanı* = tikler arası zaman = çevrim başına saniye
- *Saat oranı (frekans)* = saniye başına çevrim
(1 Hz. = 1 çevrim/saniye, 1 MHz. = 10^6 çevrim / saniye)

- *Örnek* : 200 Mhz $\frac{1}{200 \times 10^6} \times 10^9 = 5$ nanoseconds *çevrim süresi eder*

Performans Denklemi I

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



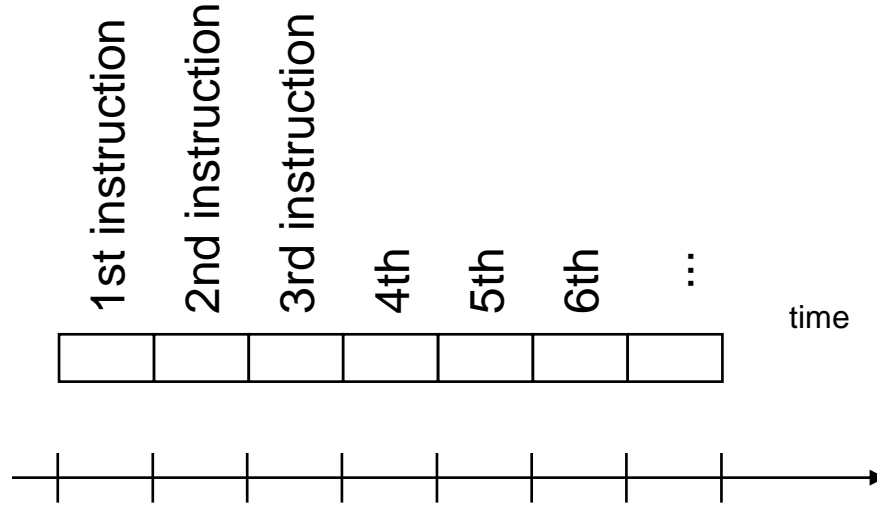
Bir program için:

$$\text{CPU çalışma zamanı} = \text{CPU saat çevrim sayısı} \times \text{Saat çevrimi zamanı}$$

- Dolayısıyla, performansı artırmak için:
 - Bir programın çevrim sayısını düşür veya
 - Saat çevrim zamanını düşür veya
 - Saat frekansını artır (overclock)

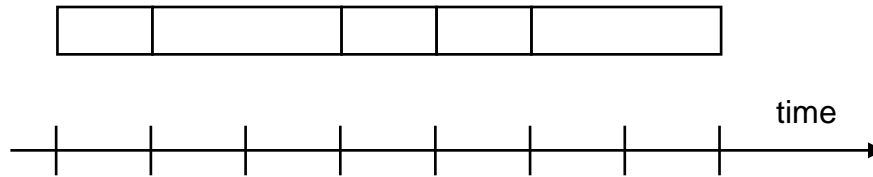
Bir program için ne kadar çevrim gerekli?

- Çevrim sayısı = komut sayısı ??



- **Bu varsayım kesinlikle yanlıştır! Çünkü:**
 - Farklı komutların işlenmesi farklı zamanlar(çevrimler) alır
 - Neden...?

Bir program için ne kadar çevrim gerekli?



- Çarpma toplamadan daha çok zaman alır
- Virgüllü (kayan noktalı) işlemler tamsayılardan daha çok zaman alır.
- Belleğe erişim kaydedicilere erişimden daha çok zaman alır
- *Çok önemli:* çevrim zamanını değiştirmek donanım tasarımının değişmesine neden olduğundan komutların çalışması için gerekli çevrim sayısını değiştirir.

Örnek I

- En sevdiğimiz program 400Mhz olan A bilgisayarında 10 saniye çalışıyor.
- Bir bilgisayar tasarımcısına, bu programı 6 saniye içinde çalıştıracak yeni bir B makinesi tasarlamasına yardımcı olmaya çalışıyoruz.
- Tasarımcı, saat hızını önemli ölçüde artırmak için yeni (veya belki de daha pahalı) teknolojiyi kullanabilir, ancak bu artışın CPU tasarımının geri kalanını etkileyeceğini ve aynı program için B makinesinin A makinesinden 1.2 kat daha fazla saat çevrimi gerektirmesine neden olacağını bize bildirdi..
- Tasarımcıya hangi saat hızını hedef olarak bildirmeliyiz?

Çözüm:

Terminoloji

- Bir program çalışmak için aşağıdakileri gerektirir:
 - Komutlar
 - Çevrimler
 - Saniyeler
- Bu büyüklükleri aşağıdaki terimler ile açıklarız:
 - *Çevrim zamanı* (çevrim / saniye)
 - *Saat frekansı* (çevrim / saniye)
 - *(ortalama) CPI* (çevrim / komut - cycles per instruction)
 - Kayan nokta yoğun bir uygulamada yüksektir
 - *IPC* (komut / çevrim)
 - *MIPS* (milyon komut / saniye)
 - Basit komutlardan oluşan bir programda yüksek

Hızlanma Vs. Oran

- ⑩ “Hızlanma” bir orandır. «**kat**», «**kere**» ile ifade edilir

= eski çalışma zamanı / yeni çalışma zamanı

- ⑩ “İyileşme- Bağlı Değişim ”, “**artış**”, “**düşüş**” genellikle belirli bir referansa göre yüzdelik değeri verir

- ⑩ «**daha**» fazla, hızlı, yavaş, çok...

= (yeni perf – eski perf) / eski perf

Bağlı değişim = hızlanma - 1

ÖRNEK 2

Bir program eski laptopta 100 saniye, yenisinde 70 saniye çalışıyor

Hızlanma ne kadar?

$$\text{Hızlanma} = (1/70) / (1/100) = 1.42 \text{ kat}$$

- Performanstaki iyileşme /bağıl değişim oranı ne?

$$(1/70 - 1/100) / (1/100) = 42\% \text{ daha fazla}$$

- Çalışma zamanındaki düşüm ne kadar?

30% daha az

Etki faktörleri

- Saat çevrim zamanı : teknoloji ve pipeline
- CPI: mimari ve komut seti tasarımı
- Komut sayısı: komut seti tasarımı ve derleyici
- CPI (cycles per instruction) veya IPC (instructions per cycle) matematiksel olarak doğru hesaplanamaz

CPU Performansını Etkileyen Unsurlar

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Inst Count	CPI	Clock Rate
Program	X		
Compiler	X	(X)	
Inst. Set.	X	X	
Organization		X	X
Technology			X

Örnek CPI hesabı

Base Machine (Reg / Reg)

Op	Freq	Cycles	CPI(i)	(% Time)
ALU	50%	1	.5	(33%)
Load	20%	2	.4	(27%)
Store	10%	2	.2	(13%)
Branch	20%	2	.4	(27%)
			<hr/> 1.5	

Typical Mix

Ortalama CPI = 1.5

Performans Ölçümü

- *Performans çalışma zamanı ile ölçülür*
- Diğer değişkenler de performans ölçümünde kullanılabilir mi?
 - Programın çalışması için gerekli çevrim sayısı?
 - Programdaki komut sayısı ?
 - Saniye başına çevrim sayısı ?
 - Komut başına ortalama çevrim sayısı ?
 - Saniye başına ortalama komut sayısı
- *ortak bir tuzak:* Değişkenlerden birini kabul etmek, başka bir veri yoksa performans göstergesidir.

Performans Denklemi II

$$\begin{array}{l} \text{CPU execution time} \\ \text{for a program} \end{array} = \begin{array}{l} \text{Instruction count} \\ \text{for a program} \end{array} \times \text{average CPI} \times \text{Clock cycle time}$$

- *Bu denklemi 1 nolu denklemden türetiniz*

Performans Denklemi II

CPU execution time = Instruction count × average CPI × Clock cycle time
for a program for a program

- *Bu denklemi 1 nolu denklemden türetiniz*

$$\frac{\text{Komut Sayısı}_{\text{prog}}}{\text{Komut}} \times \frac{\text{Gevrim}}{\text{Gevrim}} = \frac{\text{Gevrim}}{\text{prog}}$$

↓
CPI

Örnek III (CPI)

- Aynı komut seti mimarisinin (ISA) iki farklı uyarlaması olsun. Bir program için:
 - A makinesi 10ns saat çevrimi zamanına sahip ve $CPI=2.0$
 - B makinesi 20ns saat çevrim zamanı ve $CPI = 1.2$
- *Hangi makine ne kadar hızlıdır?*
- *Eğer iki makine de aynı ISA'ya sahip ise, saat hızı, CPI, çalışma zamanı, komut sayısı ve MIPS gibi büyüklüklerden hangileri benzer olur?*

Çözüm

Örnek IV(CPI)

- Bir derleyici tasarımcısı belirli bir makine için iki farklı kod arasında karar vermeye çalışmaktadır.
- Donanıma bağlı olarak, 3 farklı komut kümesi vardır: A, B ve C sınıfı, ve sırasıyla 1,2,3 çevrim değerlerine sahiptirler.
- İlk kod dizisi 5 komuta sahip:
2 adet A, 1 adet B, ve 2 adet C
- İkinci kod dizisi 6 komuta sahip:
4 adet A, 1 adet B, and 1 adet C.
- *Hangi kod daha hızlıdır? Ne kadar hızlıdır? Her kod dizisinin CPI değeri nedir?*

Örnek IV (CPI)

- Bir derleyici tasarımcısı belirli bir makine için iki farklı kod arasında karar vermeye çalışmaktadır.
- Donanıma bağlı olarak, 3 farklı komut kümesi vardır: A, B ve C sınıfı, ve sırasıyla 1,2,3 çevrim değerlerine sahiptirler.
- İlk kod dizisi 5 komuta sahip:
2 adet A, 1 adet B, ve 2 adet C
- İkinci kod dizisi 6 komuta sahip:
4 adet A, 1 adet B, and 1 adet C.
- *Hangi kod daha hızlıdır? Ne kadar hızlıdır? Her kod dizisinin CPI değeri nedir?*

1,1 kat

Örnek V(MIPS)

- Üç farklı komut sınıfına sahip bir 500 MHz lik makine üzerinde iki farklı derleyici test edilmektedir. Komut setleri Sınıf A, Sınıf B ve Sınıf C ve sırasıyla 1, 2 ve 3 çevrim gerektirmektedir.
- Her iki derleyici de büyük bir yazılım kodu üretmek için kullanılır.
- Derleyici 1, 5 milyar A Sınıfı komutu, 1 milyar B Sınıfı komutu ve 1 milyar Sınıf C komutu ile kod üretir.
- Derleyici 2, 10 milyar A Sınıfı komutu, 1 milyar B Sınıfı komutu ve 1 milyar Sınıf C komutu ile kod üretir.
- *Hangi derleyici MIPS e göre daha hızlıdır?*
- *Çalışma zamanına göre hangi dizi daha hızlıdır?*

Çözüm
