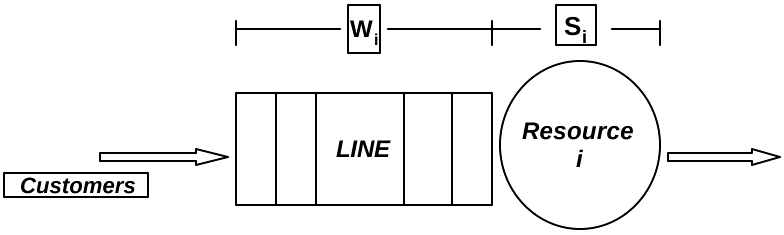
Quantifying Performance Models

Prof. Daniel A. Menascé
Department of Computer Science
George Mason University
www.cs.gmu.edu/faculty/menasce.html

Copyright Notice

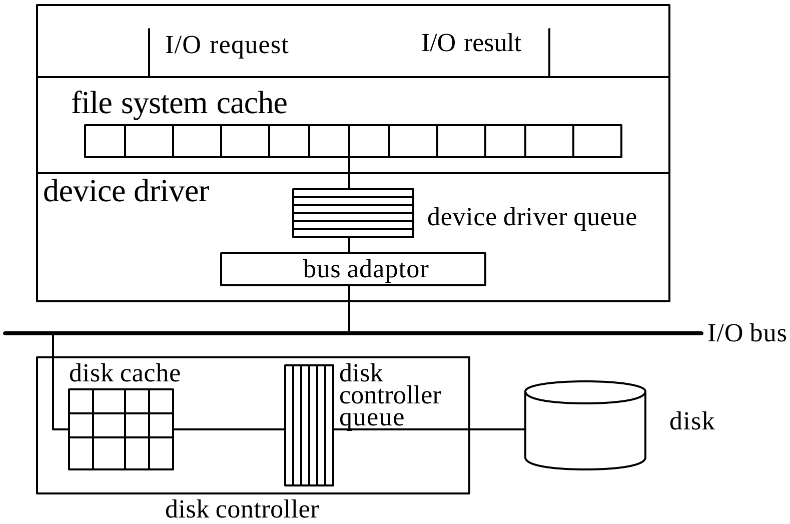
- Most of the figures in this set of slides come from the book “Performance by Design: computer capacity planning by example,” by Menascé, Almeida, and Dowdy, Prentice Hall, 2004. It is strictly forbidden to copy, post on a Web site, or distribute electronically, in part or entirely, any of the slides in this file.

A Resource and its Queue



S_i : service time
W_i : waiting time
resource: CPU, disk, network, etc.

Computing Disk Service Times



Computing Disk Service Times

$$s_d = ContrTime + P_{miss}(Seek + Latency + TransferT)$$

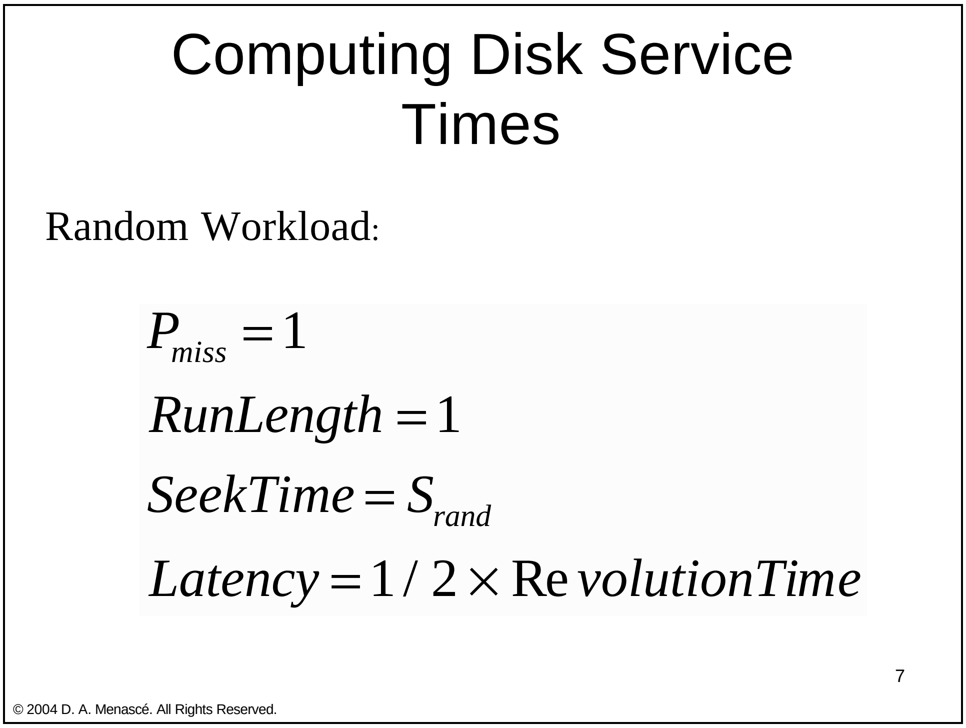
$$TransferT = \frac{BlockSize}{TransferRate}$$

Computing Disk Service Times Types of Workloads

Random Workload:
10, 201, 15, 1023, 45, 39, 782

Sequential Workload:

$$4, \underbrace{102, 103, 104, 105, 106}_{\text{run length= 5}}, 25, 88, \underbrace{32, 33, 34, 35, 36, 37, 38}_{\text{run length= 7}}, 29, 15$$



Computing Disk Service Times

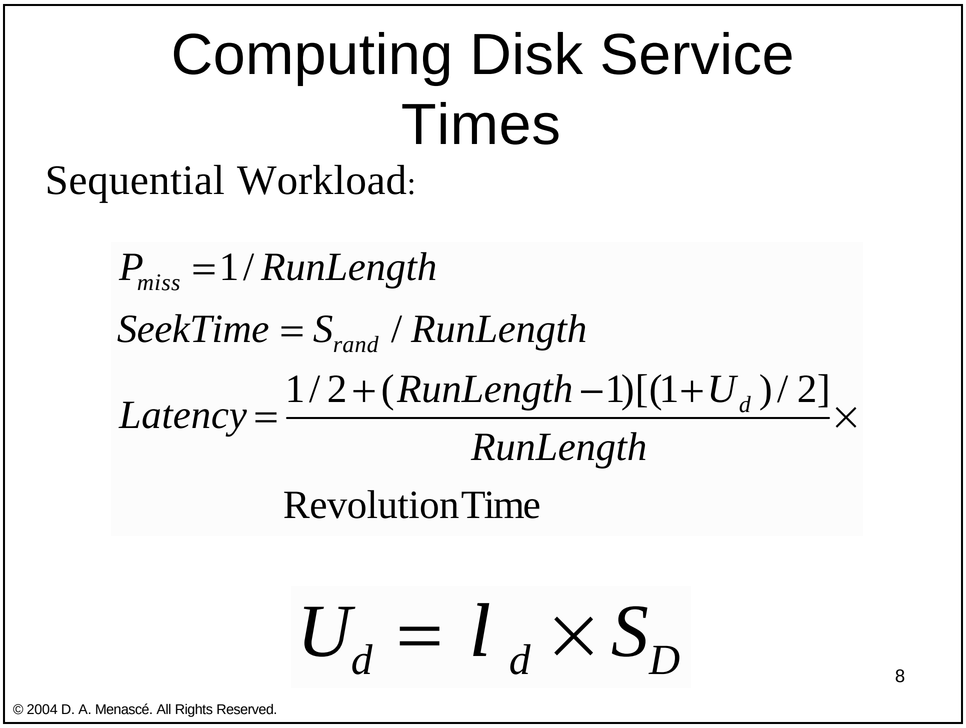
Random Workload:

$$P_{miss} = 1$$

$$RunLength = 1$$

$$SeekTime = S_{rand}$$

$$Latency = 1/2 \times \mathrm{Re}\, volutionTime$$



Computing Disk Service Times

Sequential Workload:

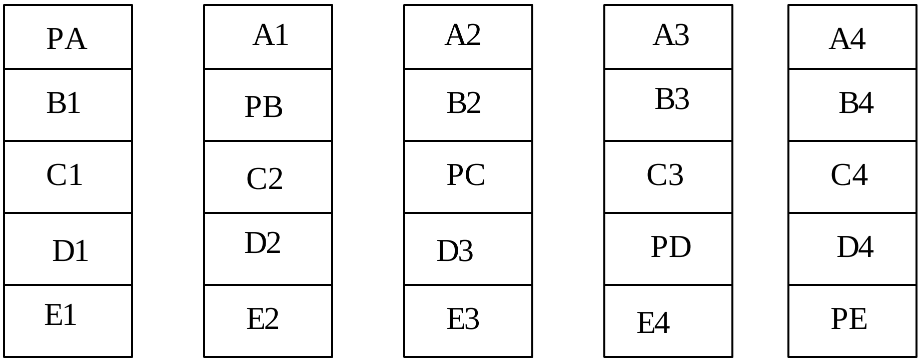
$$P_{miss} = 1/RunLength$$

$$SeekTime = S_{rand} \;/\; RunLength$$

$$Latency = \frac{1/2 + (RunLength - 1)[(1 + U_d)/2]}{RunLength} \times$$
$$\mathrm{RevolutionTime}$$

$$U_d = l_d \times S_D$$

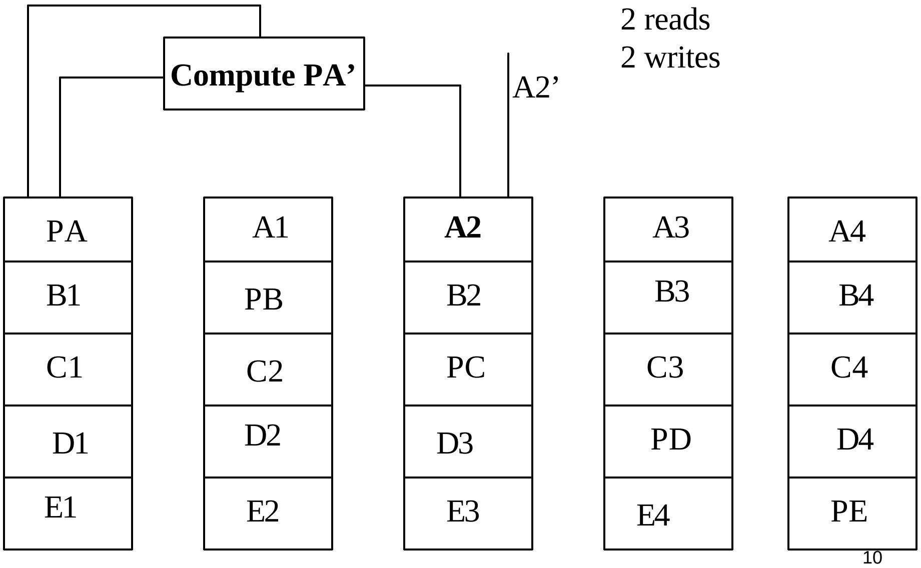
Disk Arrays



PA	A1	A2	A3	A4
B1	PB	B2	B3	B4
C1	C2	PC	C3	C4
D1	D2	D3	PD	D4
E1	E2	E3	E4	PE

Disk Arrays - Write One Stripe Unit

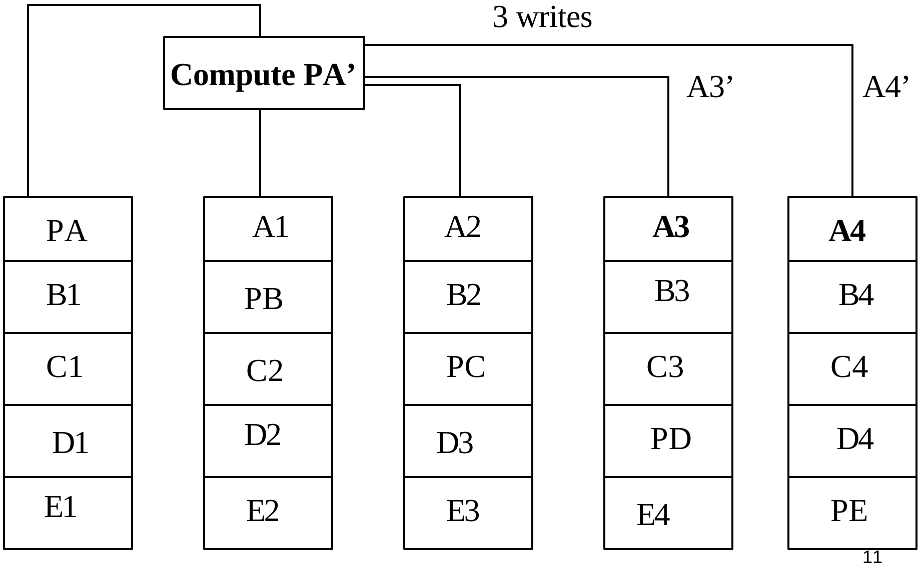
2 reads
2 writes



PA	A1	**A2**	A3	A4
B1	PB	B2	B3	B4
C1	C2	PC	C3	C4
D1	D2	D3	PD	D4
E1	E2	E3	E4	PE

Disk Arrays - Write Two Stripe Units

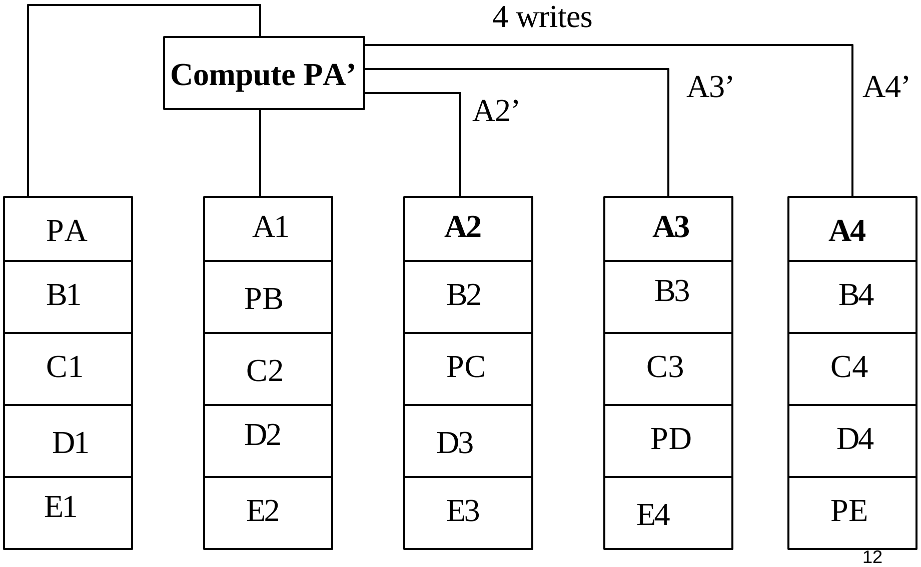
2 reads
3 writes



PA	A1	A2	**A3**	**A4**
B1	PB	B2	B3	B4
C1	C2	PC	C3	C4
D1	D2	D3	PD	D4
E1	E2	E3	E4	PE

Disk Arrays - Write Three Stripe Units

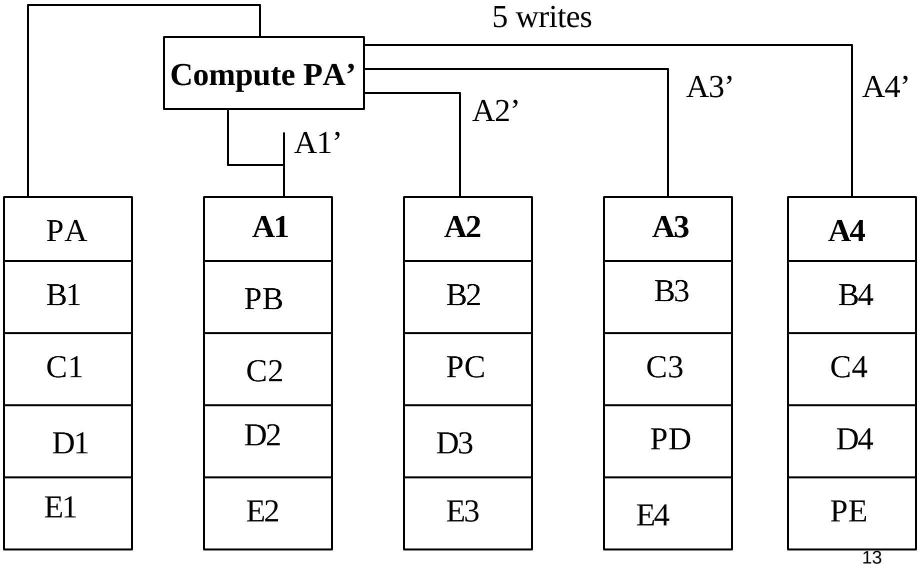
1 read
4 writes



PA	A1	**A2**	**A3**	**A4**
B1	PB	B2	B3	B4
C1	C2	PC	C3	C4
D1	D2	D3	PD	D4
E1	E2	E3	E4	PE

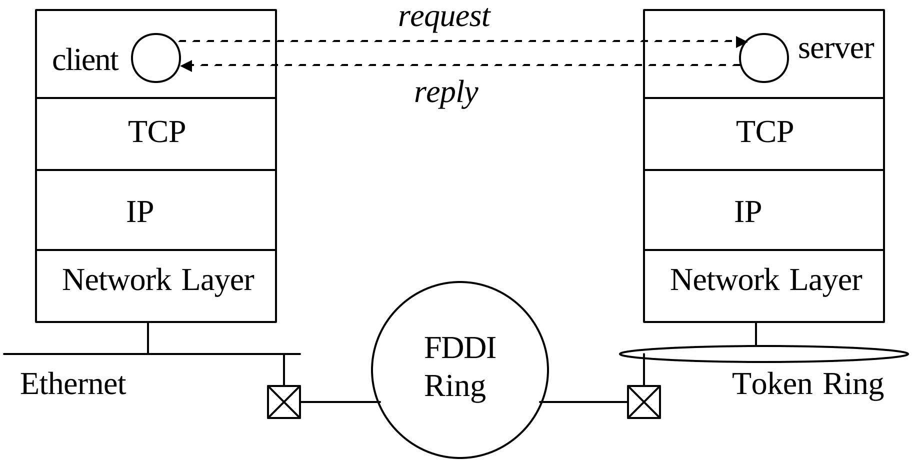
Disk Arrays - Write Four Stripe Units

0 reads
5 writes



PA	**A1**	**A2**	**A3**	**A4**
B1	PB	B2	B3	B4
C1	C2	PC	C3	C4
D1	D2	D3	PD	D4
E1	E2	E3	E4	PE

Network Service Times



Network Service Times

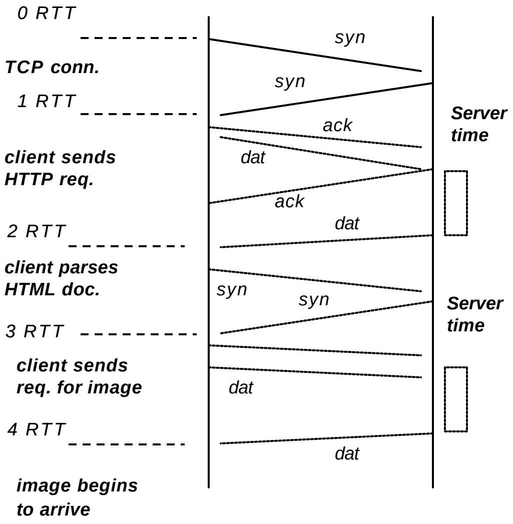
18 B (with trailer)	20 B	20 B		
Frame Header	IP Header	TCP Header	Client Request	Frame Trailer

MTU=1500 bytes
Client Message Size = 2500 bytes
No Datagrams = $\lceil 2500 / (1500–20–20) \rceil = 2$
Total Overhead = 2 * (18+20+20)=116 bytes
Message Service Time = [2500+116]*8/10,000,000=0.02098 sec

Web Page Download Times

- Depend on
  - type of HTTP protocol used
  - page parameters
  - network parameters
  - TCP parameters

HTTP 1.0 interaction



0 RTT

TCP conn.

1 RTT

client sends
HTTP req.

2 RTT

client parses
HTML doc.

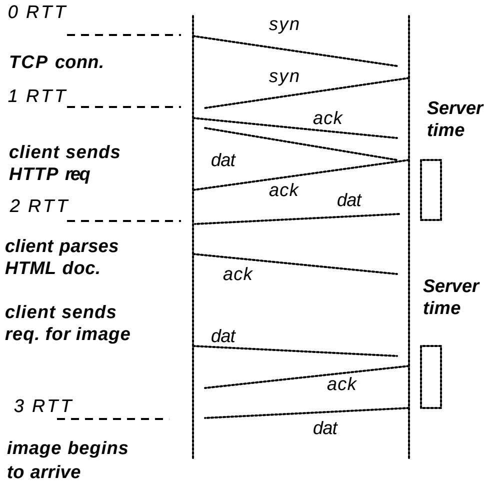
3 RTT

client sends
req. for image

4 RTT

image begins
to arrive

HTTP 1.1 interaction



0 RTT

TCP conn.

1 RTT

client sends
HTTP req

2 RTT

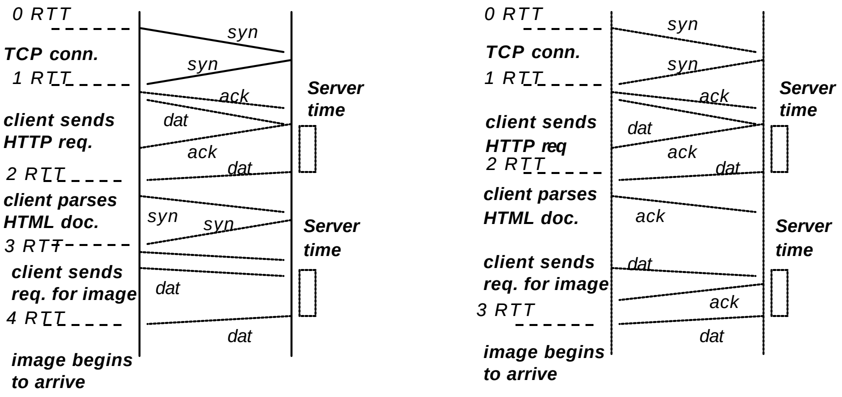
client parses
HTML doc.

client sends
req. for image

3 RTT

image begins
to arrive

HTTP 1.0 and 1.1



HTTP 1.0

HTTP 1.1

Lower Bound on Page Download Time

- PageSize: size, in bytes, of all objects of a page, including the HTTP header (290 bytes).
- B: effective network bandwidth (in bps)
- RTT: network round trip time (in sec)
- NObj: Number of embedded objects in a page.

Lower Bound on Page Download Time

- Non-persistent connection

$$PDT_{NP} > (NObj + 1) \times (2 \times RTT) + \frac{PageSize}{B}$$

- Persistent connection

$$PDT_{P} > RTT + (NObj + 1) \times RTT + \frac{PageSize}{B}$$

Page Download Time Example: Simple Page

- HTML page = 15,650 bytes
- HTTP header = 290 bytes
- 10 images of 4,200 bytes each
- RTT = 0.05 sec
- B = 125,000 bytes/sec

$$PDT_{NP} > 11 \times 2 \times 0.05 + \frac{15{,}650 + 11 \times 290 + 10 \times 4{,}200}{125{,}000} = 1.59 \quad \text{sec}$$

$$PDT_{P} > 0.05 + 11 \times 0.05 + \frac{15{,}650 + 11 \times 290 + 10 \times 4{,}200}{125{,}000} = 1.09 \quad \text{sec}$$

Page Download Time Example: Elaborate Page

- HTML page = 15,650 bytes
- HTTP header = 290 bytes
- 20 images of 20,000 bytes each
- RTT = 0.05 sec
- B = 125,000 bytes/sec

$$PDT_{NP} > 21 \times 2 \times 0.05 + \frac{15{,}650 + 21 \times 290 + 20 \times 20{,}000}{125{,}000} = 5.47 \quad \text{sec}$$

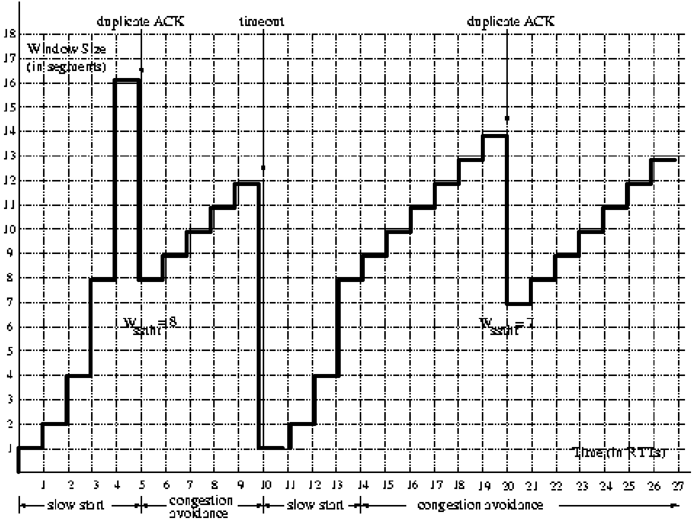
$$PDT_{P} > 0.05 + 21 \times 0.05 + \frac{15{,}650 + 21 \times 290 + 20 \times 20{,}000}{125{,}000} = 4.47 \quad \text{sec}$$

TCP Throughput

- Depends on:
  - Packet Loss Ratio
  - Round Trip Time
  - Wm: Maximum Receiver Window Size (advertised by the receiver at connection establishment time)
  - TCP timeout
  - Network Bandwith
  - Maximum Segment Size

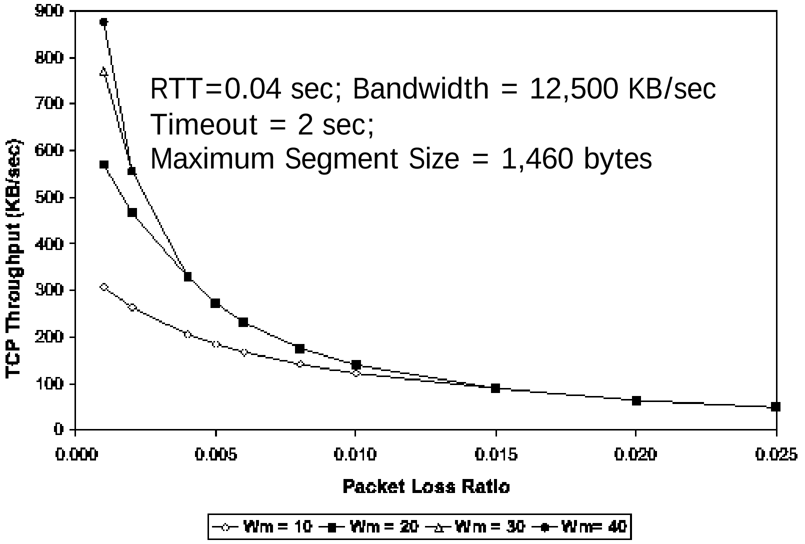
TCP: Window size vs time (in RTTs)

Window Size (segments)



Time (in RTTs)

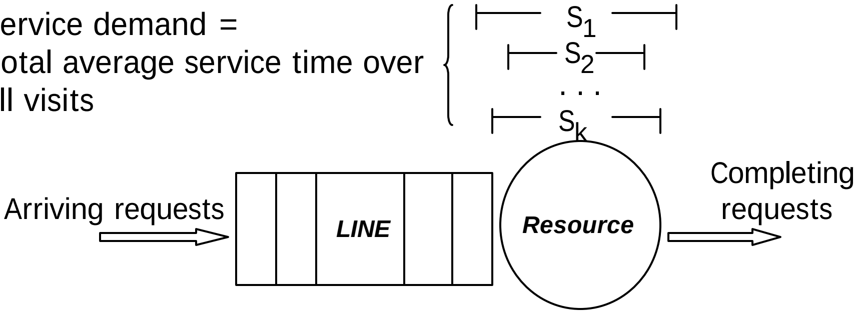
TCP Throughput



RTT=0.04 sec; Bandwidth = 12,500 KB/sec
Timeout = 2 sec;
Maximum Segment Size = 1,460 bytes

Service Demand (D)

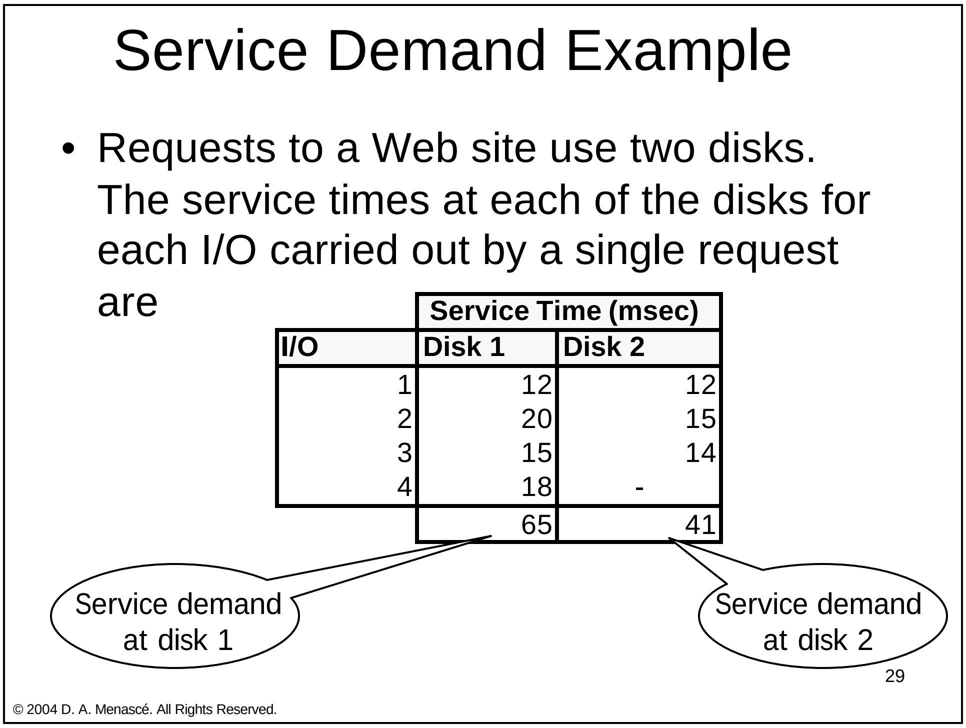
Service demand =
Total average service time over
all visits



Si: Service time at visit i
D: Service demand = S1 + S2 + ...+ Sk

Important take home!

- Service demands are <u>important</u> parameters for performance models
- Service demands are easy to measure. Service times are much harder to obtain!
- Service demands are associated with a type of request and a resource.
- Service demands are measured in time units (e.g., sec, msec)
- Service demands are load independent!
- More on this to come …



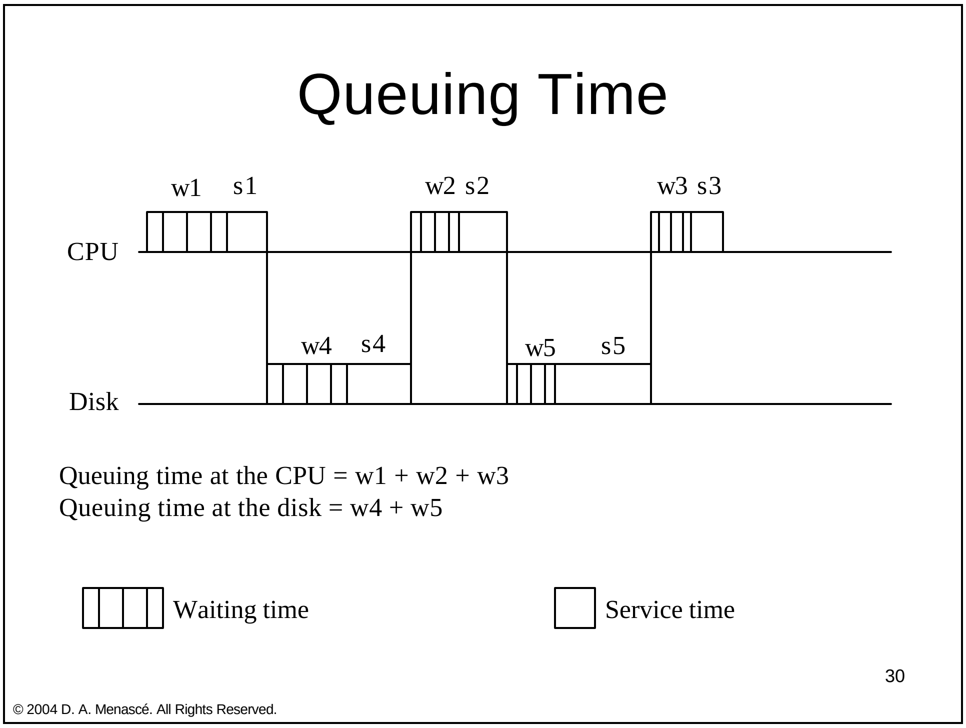
Service Demand Example

- Requests to a Web site use two disks. The service times at each of the disks for each I/O carried out by a single request are

	Service Time (msec)	
I/O	Disk 1	Disk 2
1	12	12
2	20	15
3	15	14
4	18	-
	65	41

Service demand at disk 1

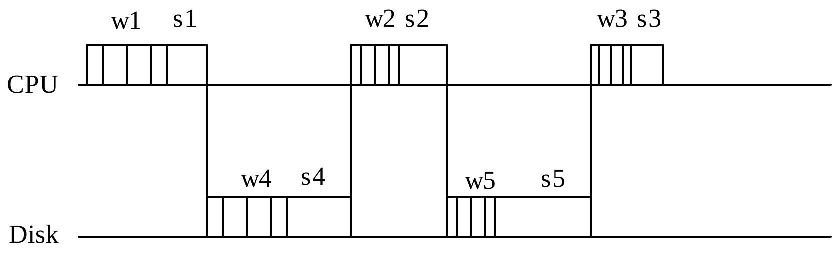
Service demand at disk 2



Queuing Time

Queuing time at the CPU = w1 + w2 + w3
Queuing time at the disk = w4 + w5

Service Demand

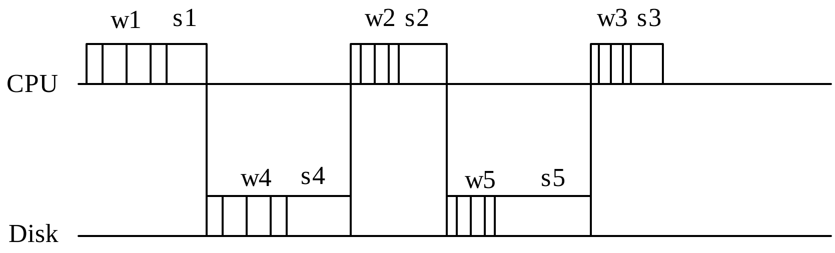


Service demand at the CPU = s1 + s2 + s3
Service demand at the disk = s4 + s5

▥ Waiting time

□ Service time

Residence Time

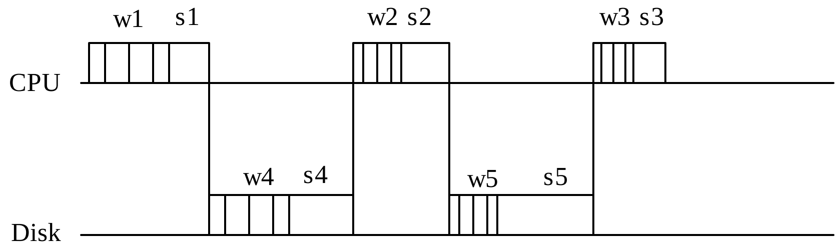


Residence time at the CPU = w1 + s1 + w2 + s2 + w3 + s3
Residence time at the disk = w4 + s4 + w5 + s5

▥ Waiting time

□ Service time

Response Time



Response time = Residence time at the CPU + Residence time at the disk

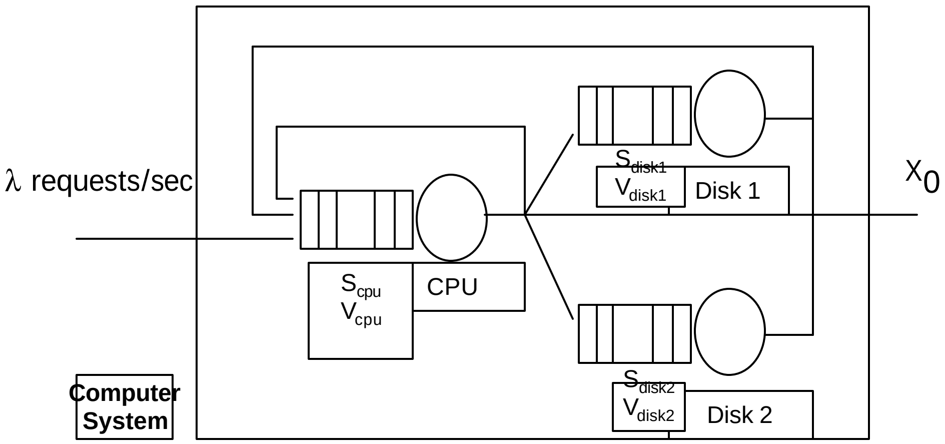
▥ Waiting time

□ Service time

Practice Questions

- What units are used to measure service demands?
- Is the service demand a function of the workload intensity?
- What is the relationship between service time and service demand?
- What is the relationship between response time, service time, and waiting time?
- What is the relationship between residence time and response time?
- What is the relationship between response time and residence time?

Computer Systems Have Many Resources!



Some Notation

- V_i: average number of visits to queue *i* by a request (e.g., avg. no. of I/Os to a disk)
- $S_{i:}$ average service time of a request at queue *i* per visit to the resource; (e.g., avg. disk service time)
- λ_i : average arrival rate of requests to queue *i* (e.g., number of I/O requests per second arriving at a disk).
- D_i : service demand of a request at queue *i*, (e.g., avg. total I/O time of a request at a given disk)

Notation (cont'd)

- $N_{i:}$ average number of requests at queue *i*, waiting or receiving service from the resource (e.g., avg. no. of I/O requests using or in the waiting queue of a give disk)
- $X_{i:}$ average throughput of queue *i*, i.e. average number of requests that complete from queue *i* per unit of time (e.g., avg. no. completed I/O requests/sec at a given disk)
- $X_{o:}$ average system throughput, defined as the number of requests that complete per unit of time. (e.g., avg. no. of completed HTTP requests/sec)

Basic Performance Results

Utilization Law

- The utilization (U_i) of resource *i* is the fraction of time that the resource is busy.

$$U_i = X_i * S_i = \lambda_i * S_i$$

Utilization Law: example 1

- The bandwidth of a communication link is 56,000 bps and it is used to transmit 1500-byte packets that flow through the link at a rate of 3 packets/sec. What is the utilization of the link?

Utilization Law: example 1

- The bandwidth of a communication link is 56,000 bps and it is used to transmit 1500-byte packets that flow through the link at a rate of 3 packets/sec. What is the utilization of the link?
- Avg Packet Service (transmission) Time =

  (1500 x 8) / 56000 = 0.214 sec/packet
- Link Throughput = 3 packets/sec
- Link Utilization = 0.214 sec/packet x 3 packets/sec = 0.642 = 64.2%

Utilization Law: example 2

- A computer system has one CPU and 3 disks and supports a DB server. All DB transactions have similar resource demands and the server is under a constant load. Measurements taken during one hour show that 13,680 transactions were executed. The number of reads and writes and the disk utilizations are shown in the table.
- What is the average service time per request on each disk?
- What is the DB server's throughput?

Disk	Reads/sec	Writes/sec	I/Os/sec	Util.
1	24	8	32	0.30
2	28	8	36	0.41
3	40	10	50	0.54

Utilization Law: example 2

- A computer system has one CPU and 3 disks and supports a DB server. All DB transactions have similar resource demands and the server is under a constant load. Measurements taken during one hour show that 13,680 transactions were executed. The number of reads and writes and the disk utilizations are shown in the table.
- What is the average service time per request on each disk?
- What is the DB server's throughput?

Disk	Reads/sec	Writes/sec	I/Os/sec	Util.
1	24	8	32	0.30
2	28	8	36	0.41
3	40	10	50	0.54

- Si = Ui / Xi
- S1 = 0.3/32 = 0.0094 sec
- S2 = 0.41/36 = 0.0114 sec
- S3 = 0.54 / 50 = 0.0108 sec
- X0 = 13680 /3600 = 3.8 tps

Utilization Law: example 3

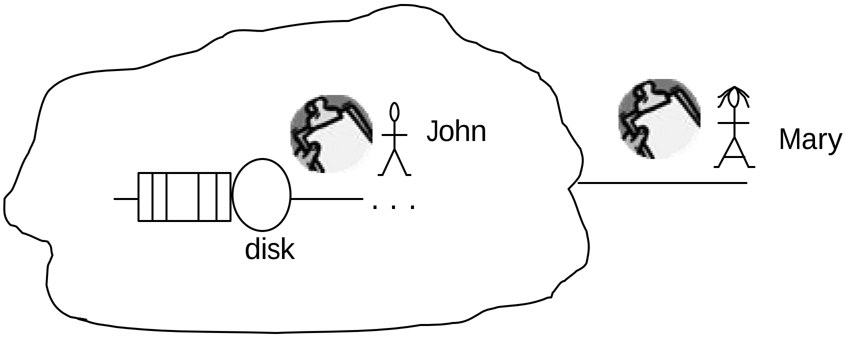
- A network segment transmits 1,000 packets/sec. Each packet has an average transmission time equal to 0.15 msec.
- What is the utilization of the LAN segment?

Utilization Law: example 3

- A network segment transmits 1,000 packets/sec. Each packet has an average transmission time equal to 0.15 msec.
- What is the utilization of the LAN segment?

$$U_{LAN} = X_{LAN} * S_{LAN} = 1,000 * 0.00015 = 0.15 = 15\%$$

Forced Flow Law



Each transaction does 3 I/Os on average and Mary measures a throughput equal to 12 tps. How many I/Os per second are seen by John?

Forced Flow Law

- By definition of the average number of visits V_i, each completing request has to pass V_i times, on the average, by queue *i*. So, if X_o requests complete per unit of time, V_i*X_o requests will visit queue *i.*

$$X_i = V_i * X_o$$

Forced Flow Law: example 1

- **Database transactions perform an average of 4.5 I/O operations on the database server. During a one-hour monitoring period, 7,200 transactions were executed.**
- **What is the average throughput of the disk?**
- **If each I/O takes 20 msec on the average, what is the disk utilization?**

Forced Flow Law: example 1

- **Database transactions perform an average of 4.5 I/O operations on the database server. During a one-hour monitoring period, 7,200 transactions were executed.**
- **What is the average throughput of the disk?**
- **If each I/O takes 20 msec on the average, what is the disk utilization?**

$$X_{server} = 7{,}200 / 3{,}600 = 2 \text{ tps}$$
$$X_{disk} = V_{disk} * X_{server} = 4.5 * 2 = 9 \text{ tps}$$
$$U_{disk} = X_{disk} * S_{disk} = 9 * 0.02 = 0.18 = 18\%$$

Forced Flow Law: example 2

- X0 = 13680 /3600 = 3.8 tps
- What is the average number of I/Os made by a transaction I/Os on each disk?
- Vi = Xi/X0
- V1 = 32/3.8 = 8.4 I/Os
- V2 = 36/3.8 = 9.5 I/Os
- V3 = 50/3.8 = 13.2 I/Os

Disk	Reads/sec	Writes/sec	I/Os/sec	Util.
1	24	8	32	0.30
2	28	8	36	0.41
3	40	10	50	0.54

Service Demand Law

- The service demand D_i is given by:

$$D_i = V_i * S_i = (X_i/X_o)(U_i/X_i) = U_i / X_o$$

Measuring Service Demands

- The service demand D_i is related to the system throughput and utilization by:

$$D_i = U_i / X_o$$

where U_i is the utilization of resource i and X_o the system throughput. Easy to get!

Example of Service Demand Law: vmstat

in	sy	cs	us	sy	idle
119	65	24	1	0	99
296	2491	289	13	6	81
260	5586	213	44	7	49
326	2822	474	21	7	72
352	1913	271	13	4	83
304	2058	280	17	5	78
275	3072	506	21	7	72
322	3340	417	18	8	74
301	2000	201	9	3	87
261	1952	282	10	4	86
251	1870	220	9	4	87
412	4646	763	33	12	54
					76.83

Interval:
12*5sec= 60 sec

Number of Requests:
20

$$U_{cpu} = 1 - 0.7683 = 0.232 = 23.2\%$$

$$X_0 = 20 / 60 = 0.333 \text{ requests/sec}$$

$$D_{cpu} = \frac{U_{cpu}}{X_0} = 0.232 / 0.333 = 0.695 \text{ sec}$$

Service Demand Law: example

- A Web server running on top of a Unix system was monitored for 10 minutes. It was observed that the CPU was 90% busy during the monitoring period. The number of HTTP requests counted in the log was 30,000.
- What is the CPU service demand of an HTTP request?

53

Service Demand Law: example

- A Web server running on top of a Unix system was monitored for 10 minutes. It was observed that the CPU was 90% busy during the monitoring period. The number of HTTP requests counted in the log was 30,000.
- What is the CPU service demand of an HTTP request?

$$U_{cpu} = 90\%$$

$$X_{server} = 30{,}000 / (10*60) = 50 \text{ requests/sec}$$

$$D_{cpu} = V_{cpu} * S_{cpu} = U_{cpu} / X_{server} = 0.90 / 50 = 0.018 \text{ sec}$$

54

Service Demand law: example 3

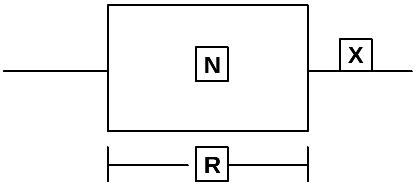
- X0 = 13680 /3600 = 3.8 tps
- Ucpu = 35%

- What are the service demands at the CPU and disks?

- Di = Ui / X0
- Dcpu = 0.35 / 3.8 = 0.092 sec
- Ddisk1 = 0.3 / 3.8 = 0.079
- Ddisk2 = 0.41 / 3.8 = 0.108
- Ddisk3 = 0.54 / 3.8 = 0.142

Disk	Reads/sec	Writes/sec	I/Os/sec	Util.
1	24	8	32	0.30
2	28	8	36	0.41
3	40	10	50	0.54

Little’s Law



- The average number of customers in a “black box” is equal to the average time each customer spends in the “box” times the throughput of the “box”.

$$\boxed{N = R * X}$$

Little’s Law: example 1

- An NFS server was monitored during 30 min and the number of I/O operations performed during this period was found to be 32,400. The average number of active requests (N_{req}) was 9.
- What was the average response time per NFS request at the server?

Little’s Law: example 1

- An NFS server was monitored during 30 min and the number of I/O operations performed during this period was found to be 32,400. The average number of active requests (N_{req}) was 9.
- What was the average response time per NFS request at the server?

“black box” = NFS server

$$X_{server} = 32{,}400 / 1{,}800 = 18 \text{ requests/sec}$$

$$R_{req} = N_{req} / X_{server} = 9 / 18 = 0.5 \text{ sec}$$

Little’s Law: example 2

- A large portal service offers free email service. The number of registered users is two million and 30% of them send send mail through the portal during the peak hour. Each mail takes 5.0 sec on average to be processed and delivered to the destination mailbox. During the busy period, each user sends 3.5 mail messages on average. The log file indicates that the average size of an e-mail message is 7,120 bytes.
- What should be the capacity of the spool for outgoing mails during the peak period?

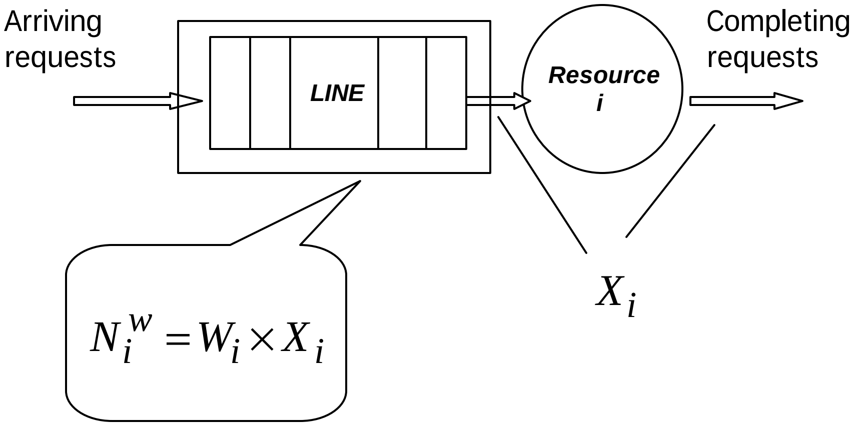
Little’s Law: example 2

- A large portal service offers free email service. The number of registered users is two million and 30% of them send send mail through the portal during the peak hour. Each mail takes 5.0 sec on average to be processed and delivered to the destination mailbox. During the busy period, each user sends 3.5 mail messages on average. The log file indicates that the average size of an e-mail message is 7,120 bytes.
- What should be the capacity of the spool for outgoing mails during the peak period?

AvgNumberOfMails = Throughput x ResponseTime
= (2,000,000 x 0.30 x 3.5 x 5.0) / 3,600 =
2,916.7 mails

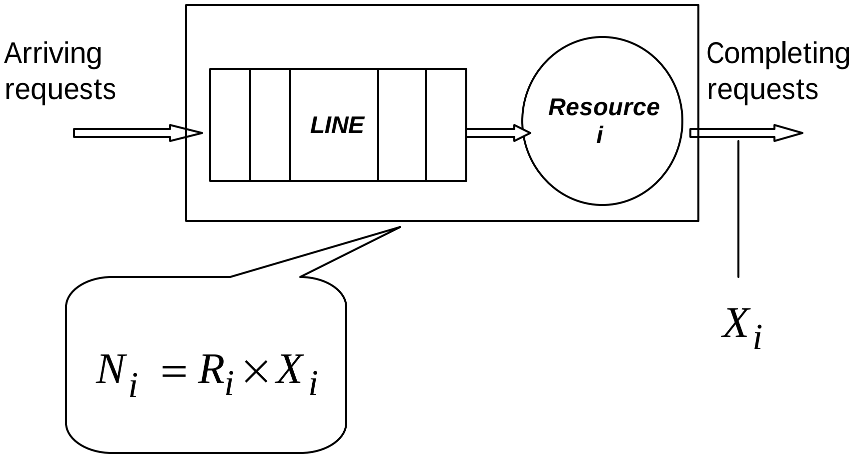
AvgSpoolFile = 2,916.7 x 7,120 bytes = 19.8 MBytes

Applying Little’s Law to the Waiting Line



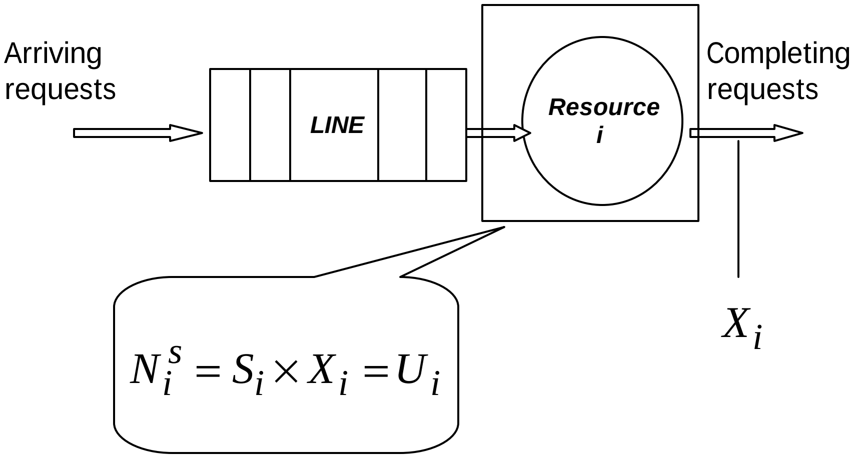
$$N_i^w = W_i \times X_i$$

Applying Little’s Law to the Queue



$$N_i = R_i \times X_i$$

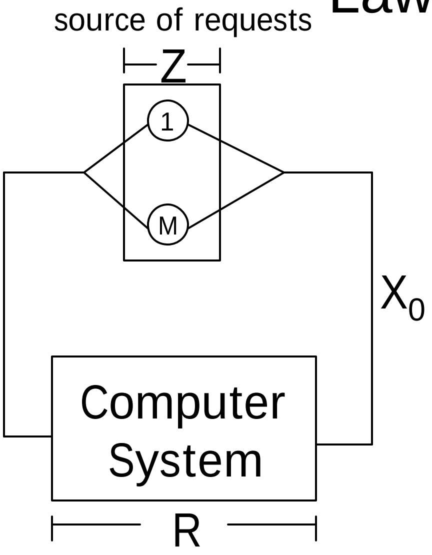
Applying Little's Law to the Server



$$N_i^s = S_i \times X_i = U_i$$

X_i

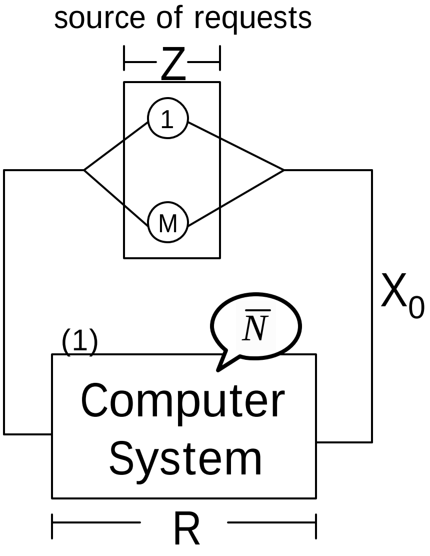
Interactive Response Time Law



R = M/X_0 - Z

R: avg. response time
Z: avg. think time
X_0: avg. throughput
M: number of sources of requests.

Interactive Response Time Law



R: avg. response time

Z: avg. think time

X_0: avg. throughput

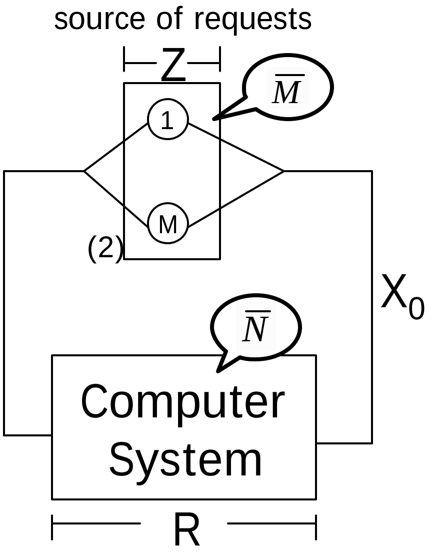
M: number of sources of requests.

Apply Little's Law to the box (1):

$$\overline{N} = X_0 \times R$$

65

Interactive Response Time Law



R: avg. response time

Z: avg. think time

X_0: avg. throughput

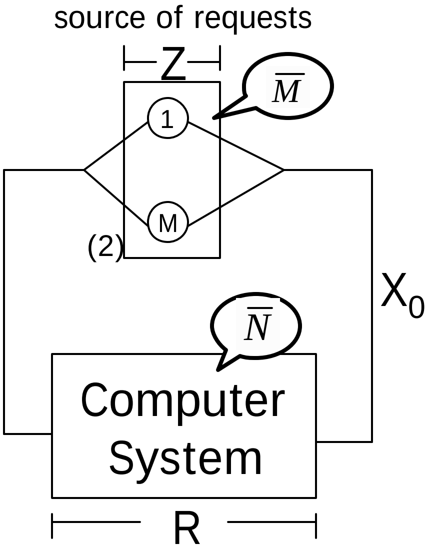
M: number of sources of requests.

Apply Little's Law to box (2):

$$\overline{M} = X_0 \times Z$$

66

Interactive Response Time Law



R: avg. response time
Z: avg. think time
X_0: avg. throughput
M: number of sources of requests.

Combining the results:

$$\overline{N} = X_0 \times R$$

$$\overline{M} = X_0 \times Z$$

––––––––––––––

$$\overline{N} + \overline{M} = M = X_0 (R + Z)$$

$$\Rightarrow \boxed{R = \frac{M}{X_0} - Z}$$

Interactive Response Time Law Example

- A database server is capable of processing 20 requests/sec. The average think time is 15 sec. What is the maximum number of client machines that can be supported so that the average response time does not exceed 2 seconds?

Interactive Response Time Law Example

- A database server is capable of processing 20 requests/sec. The average think time is 15 sec. What is the maximum number of client machines that can be supported so that the average response time does not exceed 2 seconds?
- $Z = 15$ sec, $X_0 = 20$ req/sec. So,
- $M = (R + 15) * 20 \leq (2 + 15) * 20 = 340$

Summary of Basic Results

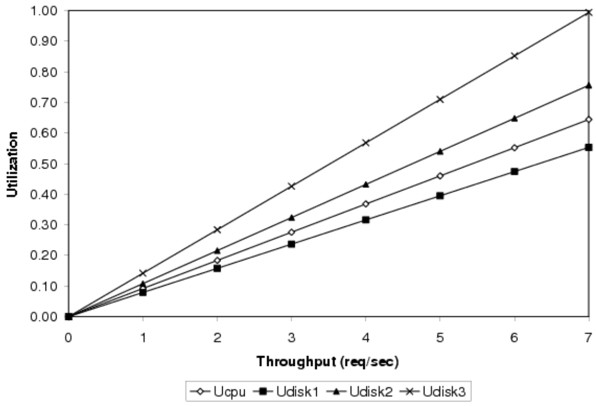
- Basic Concept of Queuing Theory and Operational Analysis
  - terminology and notation
  - service time and service demand
  - waiting time and queuing time
- Basic Performance Results and Examples
  - utilization law: $U_i = X_i * S_i$
  - forced flow law: $X_i = V_i * X_0$
  - service demand law: $D_i = V_i * S_i = U_i / X_0$
  - Little’s Law: $N = R * X$
  - Interactive Response Time Law: $R = M/X_0 - Z$

Practice Questions

- What is service demand?
- What units are used to measure service demands?
- Is the service demand a function of the workload intensity?
- What is the relationship between service time and service demand?
- What is the relationship between response time, service time, and waiting time?
- What is the relationship between residence time and response time?
- What is the relationship between response time and residence time?

Bounds on Performance

- Bounds on response time and throughput can be computed from the service demands only.



$$U_i = D_i \times X_0$$

The resource with the largest service demand reaches 100% utilization before all others.
This resource is the bottleneck.

Throughput Bound

- The utilization of a resource cannot exceed 100%:

$$X_0 = \frac{U_i}{D_i} \leq \frac{1}{D_i}$$

This is the upper asymptotic bound on throughput under heavy load conditions.

Throughput Bound

- Apply Little’s Law to the entire system:

$$N = R \times X_0 \geq \left(\sum_{i=1}^{K} D_i \right) \times X_0$$

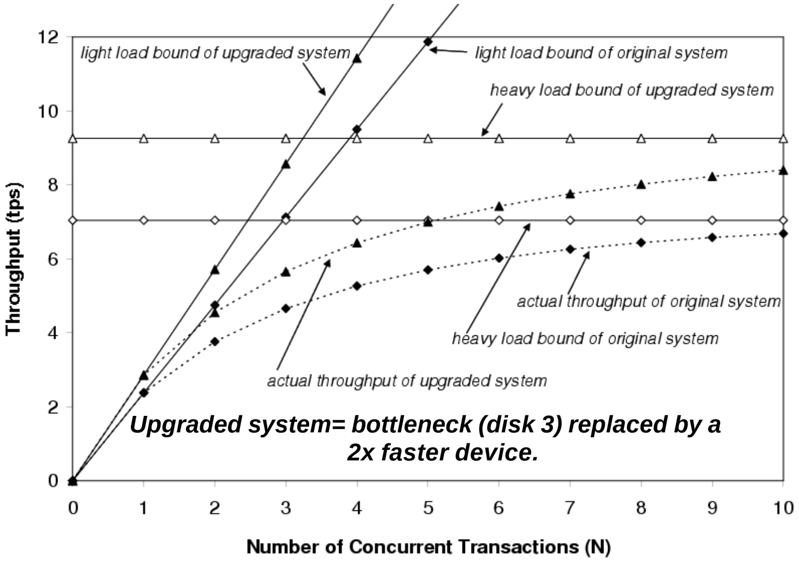
$$\Rightarrow X_0 \leq \frac{N}{\sum_{i=1}^{K} D_i}$$

This is the upper asymptotic bound on throughput under light load conditions.

Throughput Asymptotic Bounds

$$X_0 \le \min\left[\frac{1}{\max\{D_i\}}, \frac{N}{\sum_{i=1}^{K} D_i}\right]$$

Throughput Asymptotic Bound



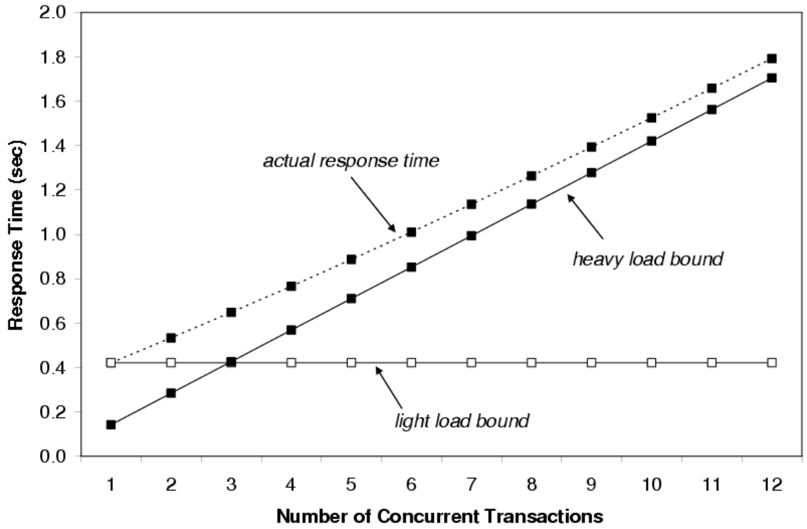
Upgraded system= bottleneck (disk 3) replaced by a 2x faster device.

Lower Bound on Response Time

$$R = \frac{N}{X_0} \geq \frac{N}{\min\left[\frac{1}{\max\{D_i\}}, \frac{N}{\sum_{i=1}^{K} D_i}\right]}$$

$$= \max\left[N \times \max\{D_i\}, \sum_{i=1}^{K} D_i\right]$$

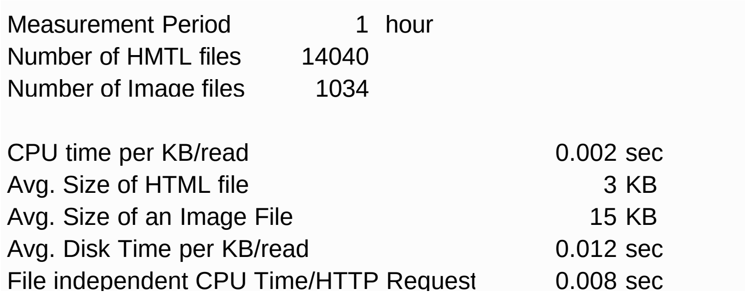
Response Time Lower Bound



Using QNs to Predict Performance

Using QNs to Predict Performance

- The following measurements were taken from a Web server. Compute the service demands and response times for HTML and image files for the current load and for a load 5 times bigger.



Measurement Period	1	hour
Number of HMTL files	14040	
Number of Image files	1034	
CPU time per KB/read	0.002	sec
Avg. Size of HTML file	3	KB
Avg. Size of an Image File	15	KB
Avg. Disk Time per KB/read	0.012	sec
File independent CPU Time/HTTP Request	0.008	sec

Using QNs to Predict Performance

- What kind of model? Open or closed? Single-class or multiclass?
  - Open since the workload intensity is given as the number of requests processed during a measurement interval.
  - Two-class model: HMTL and images (significantly different sizes)
- Arrival rates:

$$l_{HTML} = 14040 / 3600 = 3.9 \quad \text{req/sec}$$

$$l_{images} = 1034 / 3600 = 0.29 \quad \text{req/sec}$$

Using QNs to Predict Performance

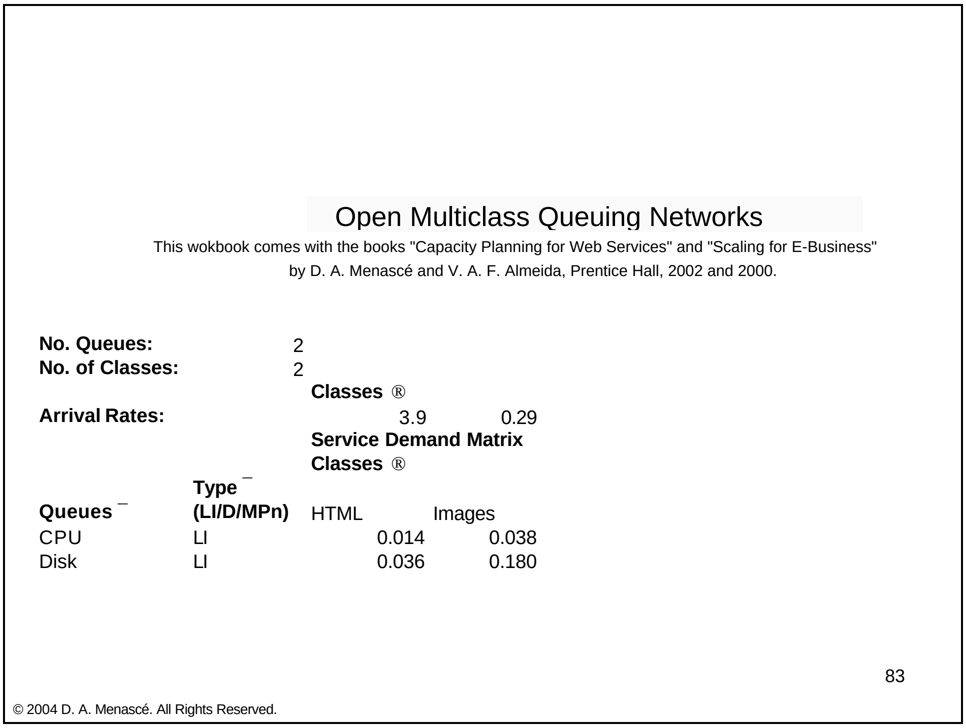
- Service demands:

$$D_{CPU,HTML} = 0.008 + 0.02 \times 3 = 0.014 \quad \text{sec}$$

$$D_{CPU,images} = 0.08 + 0.002 \times 15 = 0.038 \quad \text{sec}$$

$$D_{disk,HTML} = 3 \times 0.012 = 0.036 \quad \text{sec}$$

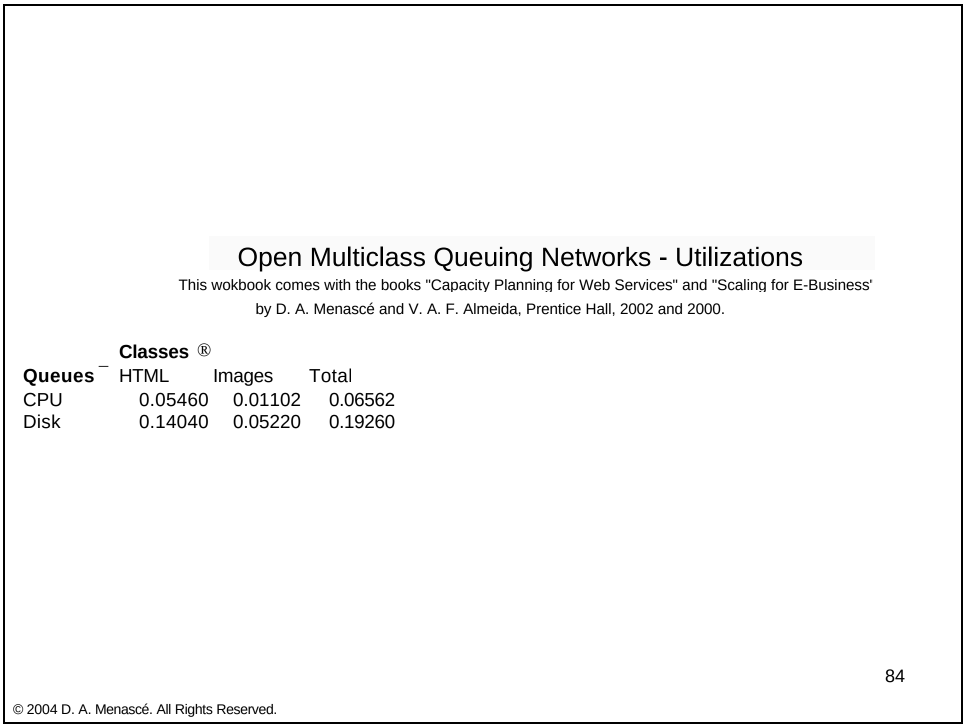
$$D_{disk,images} = 15 \times 0.012 = 0.18 \quad \text{sec}$$



Open Multiclass Queuing Networks

This wokbook comes with the books "Capacity Planning for Web Services" and "Scaling for E-Business"
by D. A. Menascé and V. A. F. Almeida, Prentice Hall, 2002 and 2000.

No. Queues:		2	
No. of Classes:		2	
		Classes ®	
Arrival Rates:		3.9	0.29
		Service Demand Matrix	
		Classes ®	
Queues ¯	**Type ¯** **(LI/D/MPn)**	HTML	Images
CPU	LI	0.014	0.038
Disk	LI	0.036	0.180



Open Multiclass Queuing Networks - Utilizations

This wokbook comes with the books "Capacity Planning for Web Services" and "Scaling for E-Business'
by D. A. Menascé and V. A. F. Almeida, Prentice Hall, 2002 and 2000.

	Classes ®		
Queues ¯	HTML	Images	Total
CPU	0.05460	0.01102	0.06562
Disk	0.14040	0.05220	0.19260

Open Multiclass Queuing Networks - Residence Times

This wokbook comes with the books "Capacity Planning for Web Services" and "Scaling for E-Business"
by D. A. Menascé and V. A. F. Almeida, Prentice Hall, 2002 and 2000.

	Classes ®	
Queues ¯	HTML	Images
CPU	0.01498	0.04067
Disk	0.04459	0.22294
Response Time	0.05957	0.26361