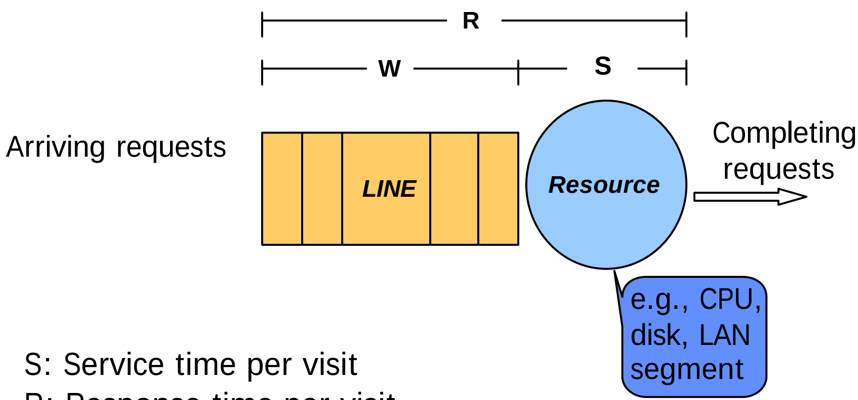
Performance of Multiprogrammed Operating Systems

Daniel A. Menascé
Department of Computer Science
George Mason University

Typical Issues

- What are the relevant performance metrics?
- How does one measure the load of a computer system?
- How does performance varies with the load?
- How can congestion be predicted?

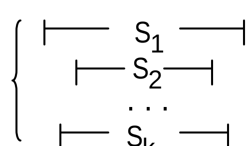
A Resource and its Queue

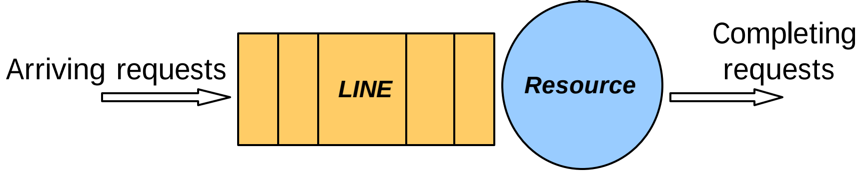


S: Service time per visit
R: Response time per visit
W: Waiting time per visit

Service Demand (D)

Service demand =
Total average service time over all visits





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

Service Demand Example

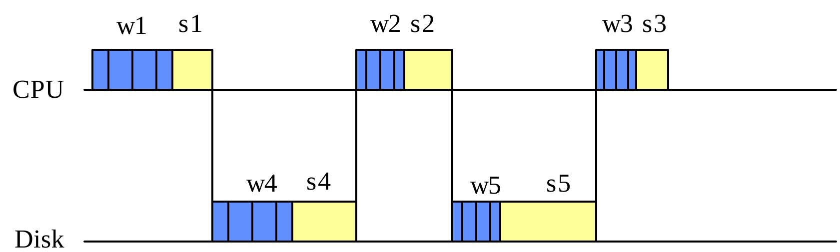
- Requests to a Web site use two disks. The service times at each of the disks for each I/O carried out by a single request are

	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

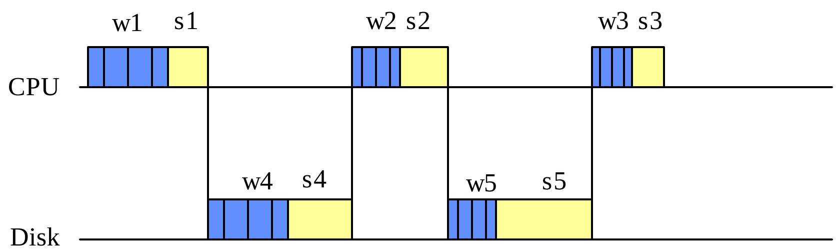
Service demand at disk 2

Service Demand



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

Residence Time

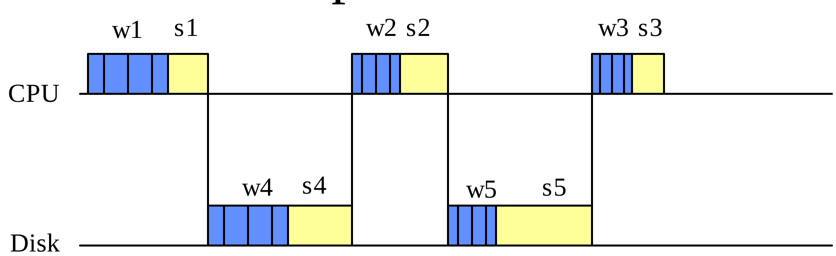


Residence time at the CPU = w1 + s1 + w2 + s2 + w3 + s3
Residence time at the disk = w4 + s4 + w5 + s5

Waiting time

Service time

Response Time

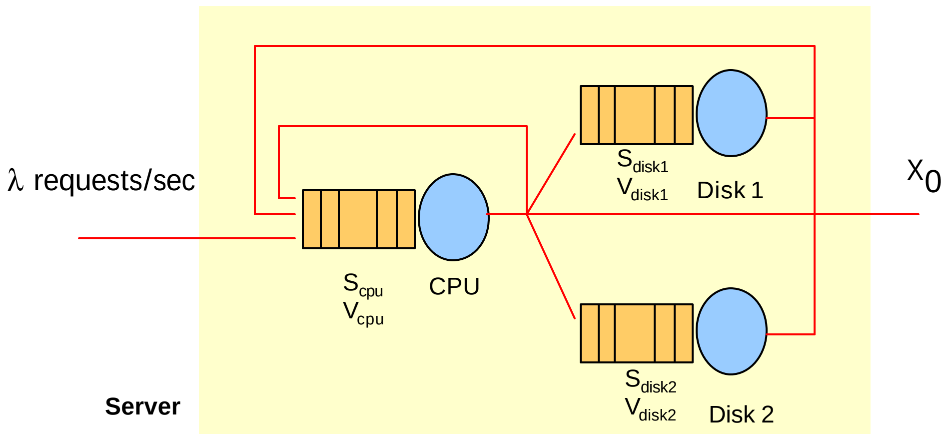


Response time = Residence time at the CPU + Residence time at the disk

Waiting time

Service time

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)
- 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.

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

Example of Utilization Law: iostat in Unix

r/s	w/s	Kr/s	Kw/s	svc_t_(msec)
0.8	7.4	6.2	131.2	136.7
0.2	4.4	1.6	113.6	61
1	14.8	8	438.4	61.3
13	1.2	128	134.4	16.8
0.2	0	1.6	0	12.4
0	0.2	0	25.6	40.9
0	0	0	0	0
0	4	0	28.6	116
0	0	0	0	0
0	0	0	0	0
3	0	24	0	11.4
0	0.6	0	35.2	35.2
0	0	0	0	0
0	0.2	0	1.6	17.3
1.30	2.34	12.10	64.90	36.36

$X_{disk} = 1.3 + 2.34 = 3.64$ IOs/sec

$$U_{disk} = X_{disk} \times S_{disk} = 3.64 \times 0.03636 = 13.24\%$$

Utilization Law: example

- 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%

Basic Performance Results

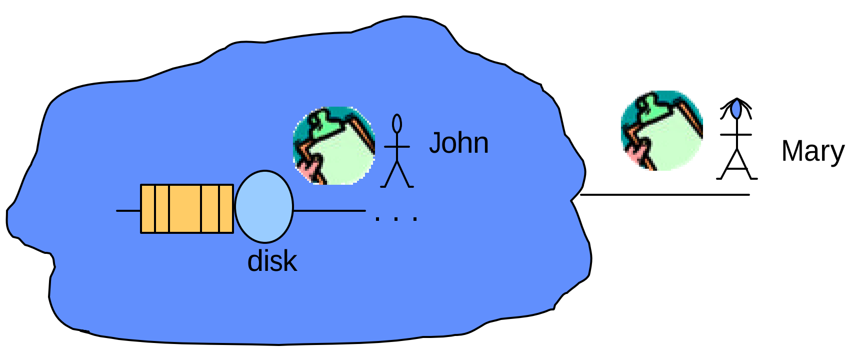
<u>Forced Flow Law</u>

- 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*.

$$X_i = V_i * X_o$$

Basic Performance Results

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: example

- **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?**

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

Basic Performance Results

Service Demand Law

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

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

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!

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?

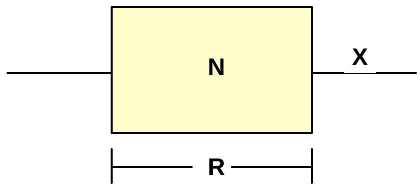
$$U_{cpu} = 90\%$$

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

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

Basic Performance Results

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".

$$N = R * X$$

Little's Law Example

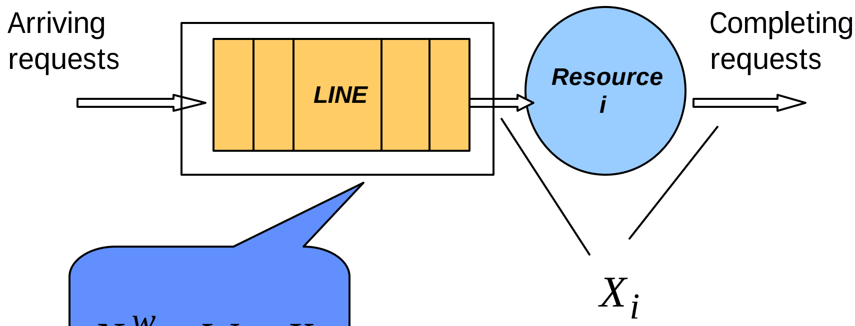
- An file server was monitored during 30 min and the number of I/O operations performed during this period was found to be 32,400. The average number of active requests (N_{req}) was 9.
- What was the average response time per file server request?

"black box" = NFS server

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

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

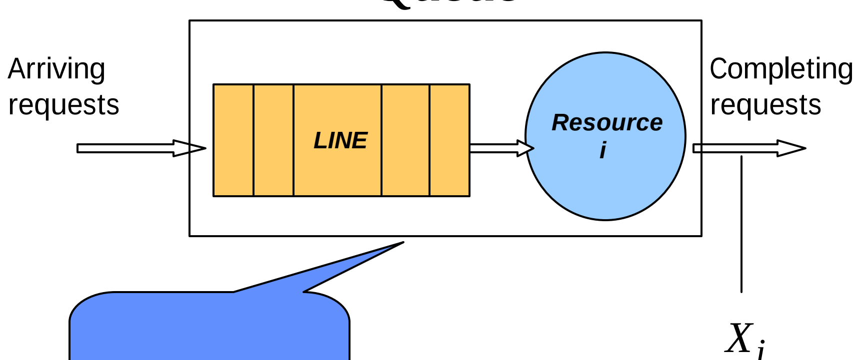
Applying Little's Law to the Waiting Line



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

23

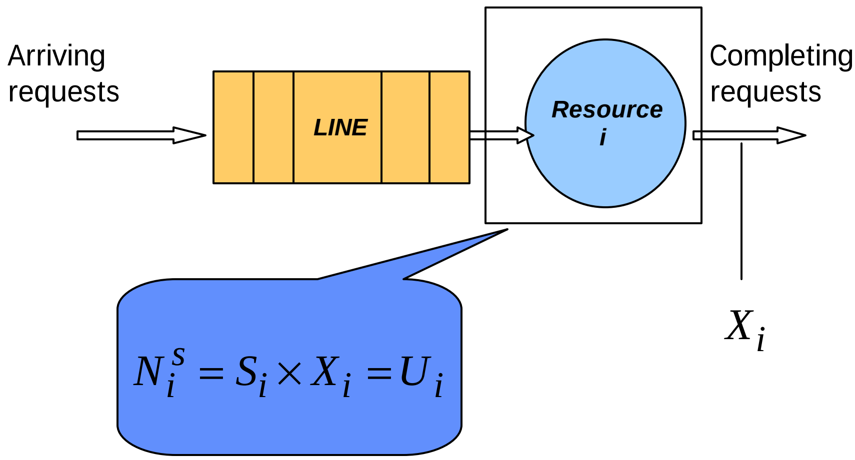
Applying Little's Law to the Queue



$$N_i = R_i \times X_i$$

24

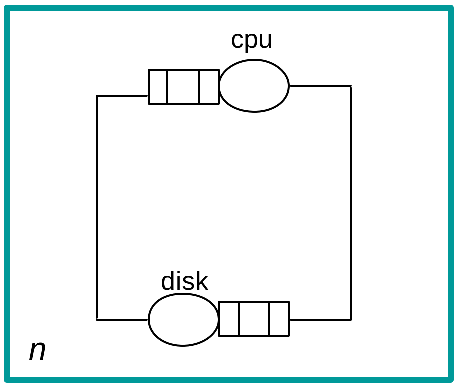
Applying Little's Law to the Server



Closed QN Model: example

- A multi-programmed computer system has one CPU and one disk. Assume that *5* processes are in execution concurrently. Each process takes 3 msec of CPU and 10 msec of disk time. What are the throughput and response time of the computer system?

Closed QN Model



Computer system

Closed QN Model: Mean Value Analysis (MVA)

$$R_i(n) = S_i + S_i \times \overline{n}_i^A(n)$$

"My response time is equal to my service time plus my waiting time (i.e, the service time of all those who arrived ahead of me)."

Notation:
(n) means "a function of n."

Closed QN Model: Mean Value Analysis (MVA)

$$R_i\,(n) = S_i + S_i \times \overline{n}_i^A(n)$$

"My response time is equal to my service time plus my waiting time (i.e, the service time of all those who arrived ahead of me)."

Arrival theorem:

$$\overline{n}_i^A(n) = \overline{n}_i(n-1)$$

"I cannot find myself in the queue, thus the n-1."

Avg. # people I find in the queue. *Avg. # people in the queue.*

Notation:
(n) means "a function of n."

Closed QN Model: Mean Value Analysis (MVA)

$$R_i\,(n) = S_i + S_i \times \overline{n}_i^A(n)$$

"My response time is equal to my service time plus my waiting time (i.e, the service time of all those who arrived ahead of me)."

Arrival theorem:

$$\overline{n}_i^A(n) = \overline{n}_i(n-1)$$

"I cannot find myself in the queue, thus the n-1."

Avg. # people I find in the queue. *Avg. # people in the queue.*

So:

$$R_i\,(n) = S_i[1 + \overline{n}_i\,(n-1)]$$

Notation:
(n) means "a function of n."

Closed QN Model: Mean Value Analysis (MVA)

But:

Avg. # visits — *Avg. response time per visit*

$$R_i^{'}(n) = V_i R_i(n) = V_i S_i [1 + \bar{n}_i\ (n-1)]$$

"The residence time is equal to the response time per visit times the average number of visits to resource i per transaction."

Closed QN Model: Mean Value Analysis (MVA)

But:

Avg. # visits — *Avg. response time per visit*

$$R_i^{'}(n) = V_i R_i(n) = V_i S_i [1 + \bar{n}_i\ (n-1)]$$

"The residence time is equal to the response time per visit times the average number of visits to resource i per transaction."

Finally, we get equation (1) of MVA: *Avg. service demand.*

$$R_i^{'}(n) = D_i [1 + \bar{n}_i\ (n-1)]$$

Closed QN Model: Mean Value Analysis (MVA)

no. of resources

Applying Little's Law to the entire system:

$$n = X_0(n) \times R_o(n) = X_0(n) \times \sum_{i=1}^{K} R_i^{'}(n)$$

Remember that the response time is the sum of all residence times?

Closed QN Model: Mean Value Analysis (MVA)

no. of resources

Applying Little's Law to the entire system:

$$n = X_0(n) \times R_o(n) = X_0(n) \times \sum_{i=1}^{K} R_i^{'}(n)$$

Remember that the response time is the sum of all residence times?

Finally, we get equation (2) of MVA:

System throughput

$$X_0(n) = n / \sum_{i=1}^{K} R_i^{'}(n)$$

Closed QN Model: Mean Value Analysis (MVA)

Applying Little's Law to resource i:

$$\bar{n}_i(n) = X_i(n) \times R_i(n)$$

Avg. queue length at resource i

Closed QN Model: Mean Value Analysis (MVA)

Applying Little's Law to resource i:

$$\bar{n}_i(n) = X_i(n) \times R_i(n)$$

Avg. queue length at resource i

Avg. # visits to resource i

Response time at resource i.

Using the Force Flow Law:

$$\bar{n}_i(n) = X_i(n) \times R_i(n) = V_i X_0(n) \times R_i(n)$$

Closed QN Model: Mean Value Analysis (MVA)

Applying Little's Law to resource i:

$$\bar{n}_i(n) = X_i(n) \times R_i(n)$$

Avg. queue length at resource i

Using the Force Flow Law:

Avg. # visits to resource i *Response time at resource i.*

$$\bar{n}_i(n) = X_i(n) \times R_i(n) = V_i X_0(n) \times R_i(n)$$

Finally, we get equation (2) of MVA:

System throughput *Residence time at resource i.*

$$\bar{n}_i(n) = X_o(n) R_i^{'}(n)$$

Mean Value Analysis (MVA): putting it all together

Residence Time Equation:

$$R_i^{'}(n) = D_i[1 + \bar{n}_i\ (n-1)]$$

Throughput Equation:

$$X_0(n) = n \Big/ \sum_{i=1}^{K} R_i^{'}(n)$$

Queue length equation:

$$\bar{n}_i(n) = X_o(n) R_i^{'}(n)$$

MVA Example Revisited

- D_{cpu} = 3 msec; D_{disk} = 10 msec.
- n = 0, 1, 2, 3, 4 , 5
- Look at the MVA equations and think how you would use them to solve the problem? [Hint: the queue length at all resources is 0 when n = 0]. In other words:

$$\bar{n}_i(0) = 0$$

Solution to the Closed QN Model

n	Rcpu	Rdisk	Ro	Xo	ncpu	ndisk
0			0.000	0.000	0.000	0.000
1	3.000	10.000	13.000	0.077	0.231	0.769
2	3.692	17.692	21.385	0.094	0.345	1.655
3	4.036	26.547	30.583	0.098	0.396	2.604
4	4.188	36.041	40.229	0.099	0.416	3.584
5	4.249	45.836	50.085	0.100	0.424	4.576
6	4.273	55.758	60.031	0.100	0.427	5.573
7	4.281	65.730	70.011	0.100	0.428	6.572
8	4.284	75.720	80.004	0.100	0.428	7.572

Can be easily computed using a spreadsheet!