# Computer Networks Performance Evaluation

# Chapter 3
# Quantifying Performance Models

## Performance by Design:
## Computer Capacity Planning by Example

Daniel A. Menascé, Virgilio A.F. Almeida, Lawrence W. Dowdy
Prentice Hall, 2004

# Outline

1. Introduction
2. Stochastic Modeling vs. Operational Analysis
3. Basic Performance Results
    1. Utilization Law
    2. Service Demand Law
    3. The Forced Flow Law
    4. Little's Law
    5. Interactive Response Time Law
4. Bounds on Performance
5. Using QN Models
6. Concluding Remarks
7. Exercises
8. Bibliography

# Introduction

- Chapter 2 introduced the basic framework that will be used throughout the book to think about performance issues in computer systems:
  - queuing networks.
- Chapter 2 concentrated on the
  - qualitative aspects of these models and
  - looked at how a computer system can be mapped into a network of queues.
- Chapter 3 focuses on the quantitative aspects of these models and

# Introduction.

- Introduce the input parameters and performance metrics that can be obtained from the QN models. The notions of
  - service times,
  - arrival rates,
  - service demands,
  - utilization,
  - response time,
  - queue lengths,
  - throughput, and
  - waiting time

  are discussed here in more precise terms.

# Stochastic Modeling vs. Operational Analysis

- *SM*
  - Ergodic stationary Markov process in equilibrium.
  - Coxian (phase) distributions of service times. *http://cnx.org/content/m10854/latest/*
  - Independence in service times and routing.
- *OA*
  - Finite time interval
  - Measurable quantities
  - Testable assumptions

*OA* made analytic modeling accessible to capacity planners in large computing environments.

# Application and Analysis of QN

- Applications
  - System Sizing; Capacity Planning; Tuning
- Analysis Techniques
  - Global Balance Solution
    - » Massive sets of Simultaneous Linear Equations
  - Bounds Analysis
    - » Asymptotic Bounds (ABA), Balanced System Bounds (BSB)
  - Solutions of "Separable" Models
    - Exact (Convolution, eMVA)
    - Approximate (aMVA)
  - Generalizations beyond "Separable" Models
    - » aMVA with extended equations

# System Sizing

- Process of determining the capacity requirements of a planned system in order to meet a given future workload and service level requirement.

- This may include, for servers:
  - Processing capacity,
  - Amount of memory and
  - I/O capacity and
- For Network:
  - Network bandwidth and
  - network hardware.

# Capacity Planning

- Determine the optimum hardware solution to meet the future workload and service level requirements.
- Ensure delivery of insufficient capacity does not lead to service impacting performance problems.
- Avoid over-provisioning, potentially resulting in stranded capital
- The capacity planner is especially receptive to products that are scalable and stable and predictable in terms of support and upgrades over the life of the product.

# Challenges

- Poorly defined service level requirements.
- Lack of understanding of system workloads.
- Difficulty obtaining meaningful test results
- System architecture not fully defined or understood.
- Target infrastructure not fully defined or understood.
- Cost of a permanently manned team or contractors.

# Basic Performance Results

- This section presents the approach known as operational analysis [1], used to establish relationships among quantities based
  - on measured or
  - known data about computer systems.
- To see how the operational approach might be applied, consider the following motivating problem.

# Motivating Problem

- Motivating problem: Suppose that during an observation period of 1 minute,
- a single resource (e.g., the CPU) is observed to be busy for 36 sec.
- A total of 1800 transactions are observed to arrive to the system.
- The total number of observed completions is 1800 transactions (i.e., as many completions as arrivals occurred in the observation period).
- What is the performance of the system (e.g.,
  - the mean service time per transaction,
  - the utilization of the resource,
  - the system throughput)?

# Operational Variables (Measured Quantities)

- The following is a partial list of such measured quantities:
  - $T$ : length of time in the observation period
  - $K$ : number of resources in the system
  - $B_i$ : total busy time of resource $i$ in the observation period $T$
  - $A_i$ : total number of service requests (i.e., arrivals) to resource $i$ in the observation period $T$
  - $A_0$ : total number of requests submitted to the system in the observation period $T$
  - $C_i$ : total number of service completions from resource $i$ in the observation period $T$
  - $C_0$ : total number of requests completed by the system in the observation period $T$

# Derived Variables

- From these known measurable quantities, called operational variables, a set of derived quantities can be obtained. A partial list includes the following:
  - $S_i$ : mean service time per completion at resource $i$ ; $S_i = B_i / C_i$
  - $U_i$ : utilization of resource $i$ ; $U_i = B_i / T$
  - $X_i$ : throughput (i.e., completions per unit time) of resource $i$ ; $X_i = C_i / T$
  - $\lambda_i$ : arrival rate (i.e., arrivals per unit time) at resource $i$ ; $\lambda_i = A_i / T$
  - $X_0$ : system throughput; $X_0 = C_0 / T$
  - $V_i$ : average number of visits (i.e., the visit count) per request to resource $i$ ; $V_i = C_i / C_0$

# Operational Analysis of motivating problem

- Using the notation above, the motivating problem can be formally stated and solved in a straightforward manner using operational analysis.
- The measured quantities are:

$$K = 1\ resource$$

$$T = 60\ \text{sec}$$

$$B_1 = 36\ \text{sec}$$

$$A_1 = A_0 = 1800\ transactions$$

$$C_1 = C_0 = 1800\ transactions$$

# Operational Analysis Motivating Problem.

- Thus, the derived quantities are :

$$S_1 = \frac{B_1}{C_1} = \frac{36}{1800} = \frac{1}{50} \; second \;\; per \;\; transaction$$

$$U_1 = \frac{B_1}{T} = \frac{36}{60} = 60\%$$

$$\lambda_1 = \frac{A_1}{T} = \frac{1800}{60} = 30tps$$

$$X_0 = \frac{C_0}{\tau} = \frac{1800}{60} = 30tps$$

# Multiple Class

- The notation presented above can be easily extended to the multiple class case by considering that $R$ is the number of classes and by adding the class number $r$ ($r = 1, \cdots, R$) to the subscript.
- For example,
  - $U_{i,r}$ is the utilization of resource $i$ due to requests of class $r$ and
  - $X_{0,r}$ is the throughput of class $r$ requests.

# Operational Law

- The subsections that follow discuss several useful relationships called:

operational laws between operational variables.

  - Utilization Law,
  - Service Demand Law,
  - The Forced Flow Law,
  - Little's Law,
  - Interactive Response Time Law,

# Utilization Law

- As seen above, the utilization of a resource is defined as $U_i = B_i/T$
- Dividing the numerator and denominator of this ratio by the number of completions from resource $i$, $C_i$, during the observation interval, yields

$$U_i = \frac{B_i}{T} = \frac{B_i/C_i}{T/C_i} \qquad (3.2.1)$$

# Utilization Law and Throughput

- The ratio $B_i/C_i$ is simply the average time that the resource was busy for each completion from resource $i$, i.e., the average service time $S_i$ per visit to the resource.

- The ratio $T/C_i$ is just the inverse of the resource throughput $X_i$.

- Thus, the relation known as the Utilization Law can be written as:

$$U_i = S_i \times X_i \qquad (3.2.2)$$

# Utilization Law.

- If the number of completions from resource $i$ during the observation interval $T$ is equal to the number of arrivals in that interval, i.e., if $C_i = A_i$, then $X_i = \lambda_i$ and the relationship given by the Utilization Law becomes

$$U_i = S_i \times \lambda_i.$$

- If resource $i$ has $m$ servers, as in a multiprocessor,
- the Utilization Law becomes $U_i = (S_i \times X_i)/m$.
- The multiclass version of the Utilization Law is $U_{i,r} = S_{i,r} \times X_{i,r}$.

# Example 3.1

- 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/second.
- What is the utilization of the link?
- Start by identifying the operational variables provided or that can be obtained from the measured data.
- The link is the resource ($K = 1$) for which the utilization is to be computed.
- The throughput of that resource, $X_1$, is 3 packets/second.
- What is the average service time per packet?

# Example 3.1.

- In other words, what is the average transmission time?
- Each packet has

  1,500 bytes/packet x 8 bits/byte = 12,000 bits/packet.

- Thus, it takes 12,000 bits/56,000 bits/sec = 0.214 sec to transmit a packet over this link.
- Therefore, $S_1$ = 0.214 sec/packet.
- Using the Utilization Law, we compute the utilization of the link as $S_1 \times X_1$ = 0.214 x 3 = 0.642 = 64.2%.

# Example 3.2

- Consider a computer system with one CPU and three disks used to support a database server.
- Assume that all database transactions have similar resource demands and that the database server is under a constant load of transactions.
- Thus, the system is modelled using a single-class closed QN, as indicated in Fig. 3.1.
- The CPU is resource 1 and the disks are numbered from 2 to 4.
- Measurements taken during one hour provide the number of transactions executed (13,680),
- the number of reads and writes per second on each disk and their utilization, as indicated in Table 3.1.

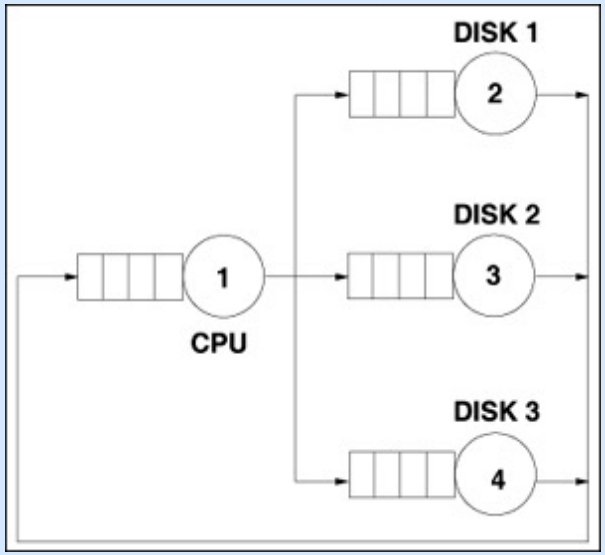
## Table 3.1. Data for Example 3.2

| Disk | Reads Per Second | Writes Per Second | Total I/Os Per Second | utilization |
|---|---|---|---|---|
| 1 | 24 | 8 | 32 | 0.30 |
| 2 | 28 | 8 | 36 | 0.41 |
| 3 | 40 | 10 | 50 | 0.54 |

# Example 3.2.

- What is the average service time per request on each disk?
- What is the database server's throughput?

Figure 3.1. Closed QN model of a database server.



# Example 3.2..

- The throughput of each disk, denoted by $X_i$ (i = 2, 3, 4), is the total number of I/Os per second, i.e., the sum of the number of reads and writes per second.
- This value is indicated in the fourth column of the table.
- Using the Utilization Law, the average service time is computed as $S_i$ as $U_i/Xi$. Thus,
  - $S_2 = U_2/X_2 = 0.30/32 = 0.0094$ sec,
  - $S_3 = U_3/X_3 = 0.41/36 = 0.0114$ sec, and
  - $S_4 = U_4/X_4 = 0.54/50 = 0.0108$ sec.
- The throughput, $X_0$, of the database server is given by $X_0 = C_0/T = 13{,}680$ transactions/3,600 seconds = 3.8 tps.

# Service Demand Law

- The service demand, $D_i$, is defined as the total average time spent by a typical request of a given type obtaining service from resource $i$.
- A request may visit several devices, possibly multiple times.
- For any given request, its service demand is the sum of all service times during all visits to a given resource.
- Service demand does not include queuing time since it is the sum of service times.

# Service Demand Law.

- Define $D_{i,r}$, as the service demand of requests of class $r$ at resource $i$.
- To illustrate the concept of service demand, consider that six transactions perform three I/Os on a disk.
- The service time, in msec, for each I/O and each transaction is given in Table 3.2.
- The last line shows the sum of the service times over all I/Os for each transaction.
- The average of these sums is 36.2 msec.
- This is the service demand on this disk due to the workload generated by the six transactions.

Table 3.2. Service times in msec for six requests. Each transactions performs three I/Os on a disk.

| | Transaction No. | | | | | |
|---|---|---|---|---|---|---|
| I/O No. | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 10 | 15 | 13 | 10 | 12 | 14 |
| 2 | 12 | 12 | 12 | 11 | 13 | 12 |
| 3 | 11 | 14 | 11 | 11 | 11 | 13 |
| Sum | 33 | 41 | 36 | 32 | 36 | 39 |

Service demand on this disk due to the workload generated by the six transactions.
(33+41+36+32+36+39)/6=36.2 msec.

# Service Demand Law..

- By multiplying the utilization $U_i$ of a resource by the measurement interval $T$ one obtains the total time the resource was busy.
- If this time is divided by the total number of completed requests, $C_0$, the average amount of time that the resource was busy serving each request is derived.
- This is precisely the service demand. So,

$$D_i = \frac{U_i \times T}{C_0} = \frac{U_i}{C_0/T} = \frac{U_i}{X_0} \qquad (3.2.3)$$

- This relationship is called the Service Demand Law, which can also be written as $D_i = V_i \times S_i$ .

# Service Demand Law...

- By definition of the service demand (and since
  $D_i = U_i/X_0 = (B_i/T)/(C_0/T)$
  $= B_i/C_0 = (C_i \times S_i)/C_0$
  $= (C_i/C_0) \times S_i = V_i \times S_i$ ).

- In many cases, Eq. (3.2.3) indicates that the service demand can be computed directly from the device utilization and system throughput.

- The multiclass version of the Service Demand Law is $D_{i,r} = U_{i,r}/X_{0,r} = V_{i,r} \times S_{i,r}$.

# Example 3.3

- A Web server is monitored for 10 minutes and its CPU is observed to be busy 90% of the monitoring period.
- The Web server log reveals that 30,000 requests are processed in that interval.
- What is the CPU service demand of requests to the Web server?
- The observation period $T$ is 600 (= 10 x 60) seconds.

# Example 3.3.

- The Web server throughput, $X_0$, is equal to the number of completed requests $C_0$ divided by the observation interval;
  - $X_0$ = 30,000/600 = 50 requests/sec.
  - The CPU utilization is $U_{CPU}$ = 0.9.
- Thus, the service demand at the CPU is
- $D_{CPU} = U_{CPU}/X_0$ = 0.9/50 = 0.018 seconds/request.

# Example 3.4

- What are the service demands at the CPU and the three disks for the database server of Example 3.2
- assuming that the CPU utilization is 35% measured during the same one-hour interval?
- Remember that the database server's throughput was computed to be 3.8 tps.
- Using the Service Demand Law and the utilization values for the three disks shown in Table 3.1, yields:
  - $D_{CPU}$ = 0.35/3.8 = 0.092 sec/transaction,
  - $D_{disk1}$ = 0.30/3.8 = 0.079 sec/transaction,
  - $D_{disk2}$ = 0.41/3.8 = 0.108 sec/transaction, and
  - $D_{disk3}$ = 0.54/3.8 = 0.142 sec/transaction.

# The Forced Flow Law

- There is an easy way to relate the
  - throughput of resource i, $X_i$,
  - to the system throughput, $X_0$.
- Assume for the moment that every transaction that completes from the database server of Example 3.2 performs an average of two I/Os on disk 1.
- That is, suppose that for every one visit that the transaction makes to the database server, it visits disk 1 an average of two times.
- What is the throughput of that disk in I/Os per second?

# The Forced Flow Law (2)

- Since 3.8 transactions complete per second (i.e., the system throughput, $X_0$) and each one performs two I/Os on average on disk 1,
- the throughput of disk 1 is 7.6 (= 2.0 x 3.8) I/Os per second.
- In other words, the throughput of a resource ($X_i$) is equal to the average number of visits ($V_i$) made by a request to that resource multiplied by the system throughput ($X_0$).
- This relation is called the Forced Flow Law:

$$X_i = V_i \times X_0 \quad (3.2.4)$$

- The multiclass version of the Forced Flow Law is: $X_{i,r} = V_{i,r} \times X_{0,r}$.

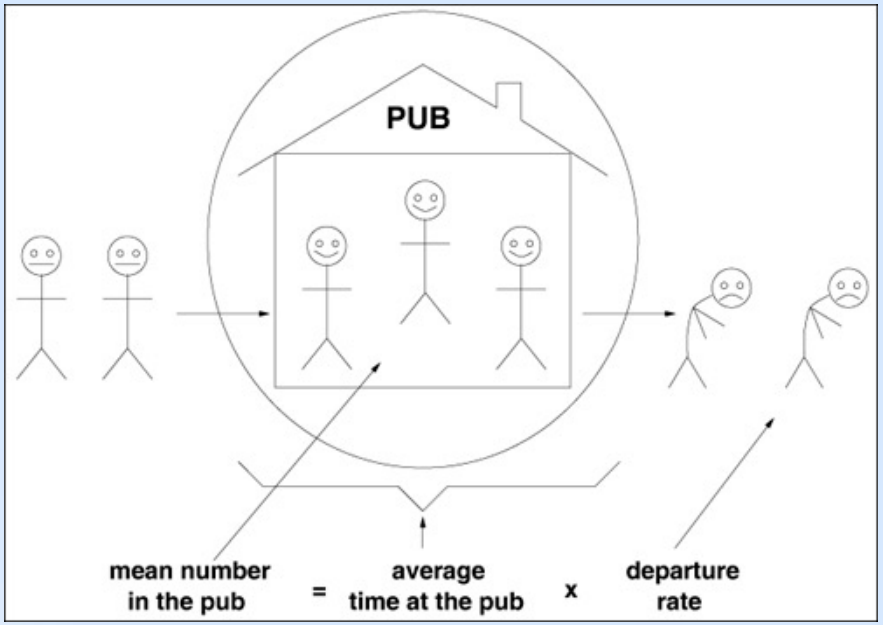
# Example 3.5

- What is the average number of I/Os on each disk in Example 3.2?
- The value of $V_i$ for each disk $i$, according to the Forced Flow Law, can be obtained as $X_i/X_0$.
- The database server throughput is 3.8 tps and the throughput of each disk in I/Os per second is given in the fourth column of Table 3.1.
- Thus, $V_1 = X_1/X_0$ = 32/3.8 = 8.4 visits to disk 1 per database transaction.
- Similarly, $V_2 = X_2/X_0$ = 36/3.8 = 9.5 and
- $V_3 = X_3/X_0$ = 50/3.8 = 13.2.

# Little's Law

- Little's result states that the average number of folks in the pub (i.e., the queue length) is equal to the departure rate of customers from the pub times the average time each customer stays in the pub (see Fig. 3.2).

Figure 3.2. Little's Law.



# Little's Law.

- This result applies across a wide range of assumptions.
- For instance, consider a deterministic situation where a new customer walks into the pub every hour on the hour.
- Upon entering the pub, suppose that there are three other customers in the pub.
- Suppose that the bartender regularly kicks out the customer who has been there the longest, every hour at the half hour.
- Thus, a new customer will enter at 9:00, 10:00, 11:00, ..., and
- the oldest remaining customer will be booted out at 9:30, 10:30, 11:30, ....

# Little's Law..

- It is clear that the average number of persons in the pub will be $3\frac{1}{2}$ ,
- since 4 customers will be in the pub for the first half hour of every hour and
- only 3 customers will be in the pub for the second half hour of every hour.
- The departure rate of customers at the pub is one customer per hour.
- The time spent in the pub by any customer is $3\frac{1}{2}$ hours. Thus, via Little's Law:

$$avg.number\ in\ pub = departure\ rate\ at\ pub \times avg.time\ spent\ in\ pub$$

$$3\frac{1}{2} = 1 \times 3\frac{1}{2}$$

# Little's Law…

- Also, it does not matter which customer the bartender kicks out.
- For instance, suppose that the bartender chooses a customer at random to kick out.
- We leave it as an exercise to show that the average time spent in the pub in this case would also be  hours.
  - [Hint: the average time a customer spends in the pub is one half hour with probability 0.25, one and a half hours with probability (0.75)(0.25) = 0.1875 (i.e., the customer avoided the bartender the first time around, but was chosen the second), two and a half hours with probability (0.75)(0.75)(0.25), and so on.]

# Little's Law....

- Little's Law applies to any "black box", which may contain an arbitrary set of components.
- If the box contains a single resource (e.g., a single CPU, a single pub) or if the box contains a complex system (e.g., the Internet, a city full of pubs and shops), Little's Law holds.

- Thus, Little's Law can be restated as

*average number of customers in a box =*

*departure rate from the box × average time spent in the box*

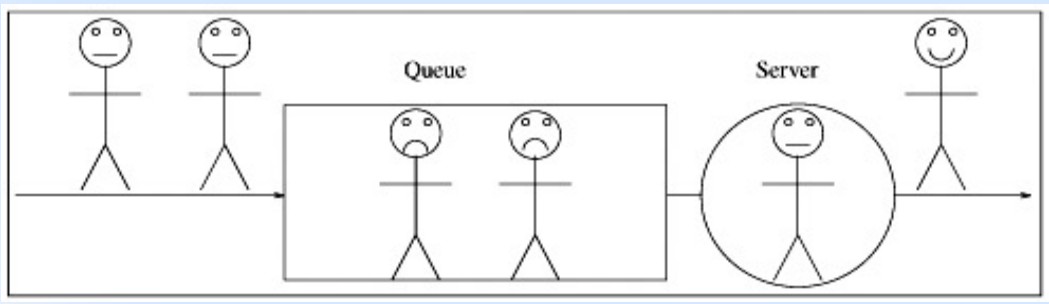
$$N_i = R_i \times X_i \qquad (3.2.5)$$

# Little's Law.....

- For example, consider the single server queue of Fig. 3.3.
- Let the designated box be the server only, excluding the queue.
- Applying Little's Law, the average number of customers in the box is interpreted as the average number of customers in the server.
- The server will either have a single customer who is utilizing the server, or the server will have no customer present.
- The probability that a single customer is utilizing the server is equal to the server utilization.
- The probability that no customer is present is equal to the probability that the server is idle.

# Single Server

Figure 3.3. Single server.



# Single Server: Little's Law

- The average number of customers in the server equals:

$$N^s = 1 \times prob[single\ customer\ present] + 0 \times prob[no\ customer\ present] \qquad (3.2.6)$$

- This simply equals the server's utilization.
- Therefore, the average number of customers in the server, $N^s$, equals the server's utilization.
- Thus, with this interpretation of Little's Law,

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

- This result is simply the Utilization Law!
- Now consider that the box includes both the waiting queue and the server.

# Single Queue: Little's Law

- The average number of customers in the box (waiting queue + server), denoted by $N_i$, is equal, according to Little's Law, to the average time spent in the box, which is the response time $R_i$, times the throughput $X_i$.

- Thus, $N_i = R_i \times X_i$.
- Little's Law indicates that

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

- ,where $N_i^w$ is the average number of customers in the queue and
- $W_i$ the average waiting time in the queue prior to receiving service.

# Little's Law and Class Aggregation - 1

$$N = N_1 + N_2$$

$$N = RX,$$

$$N_1 = R_1 X_1,$$

$$N_2 = R_2 X_2$$

$$X = X_1 + X_2 = \lambda_1 + \lambda_2$$

$$T^{agr} = R = \frac{R_1 X_1 + R_2 X_2}{X_1 + X_2} = \frac{N}{X}$$

# Little's Law and Class Aggregation - 2

$$N^s = N_1^s + N_2^s$$

$$N^s = SX,$$

$$N_1^s = S_1 X_1,$$

$$N_2^s = S_2 X_2$$

$$X = X_1 + X_2$$

$$S = \frac{S_1 X_1 + S_2 X_2}{X_1 + X_2} = \frac{N^s}{X}$$

$$W = \frac{W_1 X_1 + W_2 X_2}{X_1 + X_2} = \frac{N^w}{X}$$

# Example 3.6

- Consider the database server of Example 3.2 and assume that during the same measurement interval the average number of database transactions in execution was 16.
- What was the response time of database transactions during that measurement interval?
- The throughput of the database server was already determined as being 3.8 tps.
- Apply Little's Law and consider the entire database server as the box.

# Example 3.6.

- The average number in the box is the average number $N$ of concurrent database transactions in execution (i.e., 16).
- The average time in the box is the average response time $R$ desired.
- Thus, $R = N/X_0$ = 16/3.8 = 4.2 sec.

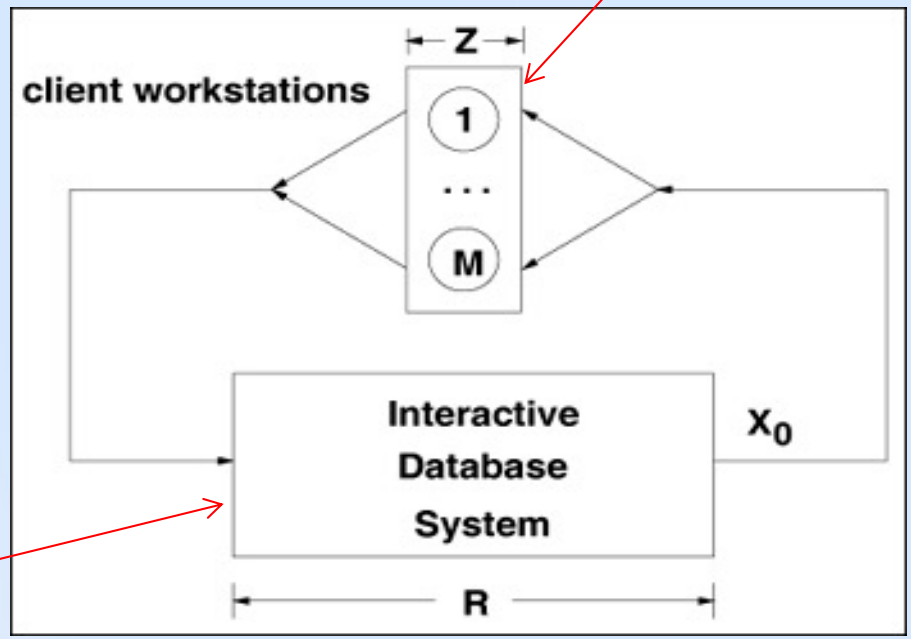
# Interactive Response Time Law

- Consider an interactive system composed of
  - $M$ clients (sources of requests),
  - average think time is denoted by $Z$ and
  - average response time is $R$.

- See Fig. 3.4.
- The think time is defined as the time elapsed since a customer receives a reply to a request until a subsequent request is submitted.
- The response time is the time elapsed between successive think times by a client.

Figure 3.4. Interactive computer system.

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




$$\overline{M} + \overline{N} = M$$

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

# Interactive Response Time Law.

- Let $\overline{M}$ and $\overline{N}$ be the average number of clients thinking and waiting for a response, respectively.
- By viewing clients as moving between workstations and the database server, depending upon whether or not they are in the think state, $\overline{M}$ and $\overline{N}$ represent the average number of clients at the workstations and at the database server, respectively.
- Clearly, $\overline{M} + \overline{N} = M$ since a client is either in the think state or waiting for a reply to a submitted request.
- By applying Little's Law to the box containing just the workstations, $\overline{M} = X_O \times Z$ (3.2.7)
- Since the average number of requests submitted per unit time (throughput of the set of clients) must equal the number of completed requests per unit time (system throughput $X_0$).

# Interactive Response Time Law..

- Similarly, by applying Little's Law to the box containing just the database server,

$$\overline{N} = X_O \times R \qquad (3.2.8)$$

- where $R$ is the average response time. By adding Eqs. (3.2.7) and (3.2.8),

$$\overline{M} + \overline{N} = M = X_O(Z + R) \qquad (3.2.9)$$

- With a bit of algebra,

$$R = \frac{M}{X_O} - Z \qquad (3.2.10)$$

# Example 3.7

- If 7,200 requests are processed during one hour by an interactive computer system with 40 clients and an average think time of 15 sec, the average response time is

$$R = \frac{40}{7200/3600} - 15 = 5\,\text{sec} \qquad (3.2.11)$$

# Example 3.8

- A client/server system is monitored for one hour. During this time, the utilization of a certain disk is measured to be 50%.
- Each request makes an average of two accesses to this disk, which has an average service time equal to 25 msec.
- Considering that there are 150 clients and that the average think time is 10 sec,
- What is the average response time?
- The known quantities are:
  - $U_{disk}$ = 0.5,
  - $V_{disk}$ = 2,
  - $S_{disk}$ = 0.025 sec,
  - $M$ = 150, and
  - $Z$ = 10 sec.

# Example 3.8.

- From the Utilization Law,

$$U_{disk} = S_{disk} \times X_{disk}$$

- Thus, $X_{disk}$ = 0.5/0.025 = 20 requests/sec.
- From the Forced Flow Law,

$$X_O = \frac{X_{disk}}{V_{disk}} = \frac{20}{2} = 10\ requests/\text{sec}$$

- Finally, from the Interactive Response Time Law,

$$R = \frac{M}{X_O} - Z = \frac{150}{10} - 10 = 5\,\text{sec}$$

# Interactive Response Time Law…

- The multiclass version of the Interactive Response Time Law is $R_r = M_r/X_{0,r} - Z_r$.
- Figure 3.5 summarizes the main relationships discussed in the previous sections.

# Figure 3.5. Summary of Operation Laws.

$$\text{Utilization Law}: \; U_i = X_i \times S_i = \lambda_i \times S_i \qquad (3.2.12)$$

$$\text{Forced Flow Law}: \; X_i = V_i \times X_o \qquad (3.2.13)$$

$$\text{Service Demand Law}: \; D_i = V_i \times S_i = U_i / X_o \qquad (3.2.14)$$

$$\text{Little's Law}: \; N = X \times R \qquad (3.2.15)$$

$$\text{Interactive Response Time Law}: \; R = \frac{M}{X_o} - Z \qquad (3.2.16)$$

# Bounds on Performance

- Upper bounds on throughput and lower bounds on response time can be obtained by considering the service demands only (i.e., without solving any underlying model).
- This type of bounding analysis can be quite useful since it provides the analyst with the best possible performance one could hope from a system.
- The bounding behaviour of a computer system is determined by its bottleneck resource.
- The bottleneck of a system is that resource with the highest utilization (or, equivalently, the resource with the largest service demand).

# Example 3.9

- Consider again the database server of Example 3.2 and the service demands for the CPU and the three disks computed in Example 3.4.
- The service demands were computed to be:
  - $D_{CPU}$ = 0.092 sec,
  - $D_{disk1}$ = 0.079 sec,
  - $D_{disk2}$ = 0.108 sec, and
  - $D_{disk3}$ = 0.142 sec.

$$\sum D_i = 0.421 sec$$

- Correspondingly, the utilization of these devices are 35%, 30%, 41%, and 54%, respectively (from Example 3.4 and Table 1.1).
- What is the maximum throughput $X_0^{\max}$ of the database server?

# Example 3.9.

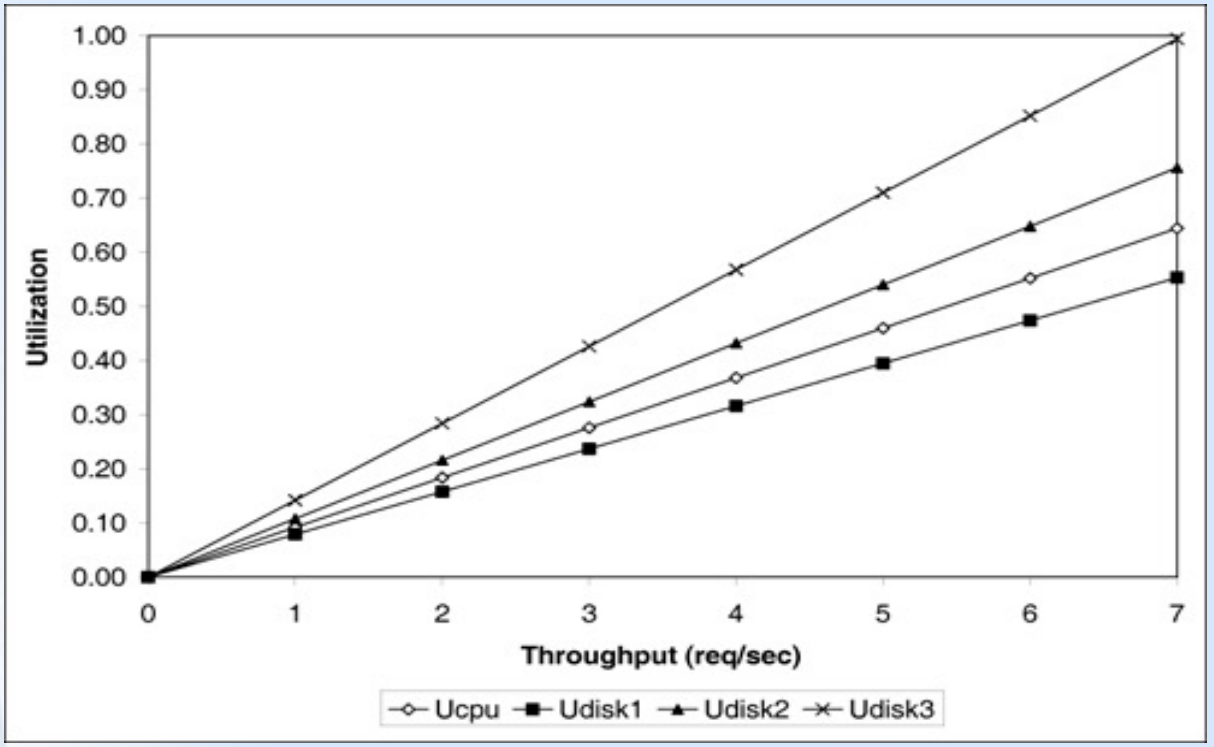
- Using the Service Demand Law, it follows that
  - $U_{CPU} = D_{CPU} \times X_0 = 0.092 \times X_0$,
  - $U_{disk1} = D_{disk1} \times X_0 = 0.079 \times X_0$,
  - $U_{disk2} = D_{disk2} \times X_0 = 0.108 \times X_0$,
  - $U_{disk3} = D_{disk3} \times X_0 = 0.142 \times X_0$.

- The service demands do not include any queuing time, only the total service required by a transaction at the device.
- Therefore, as the load (i.e., as the throughput, $X_0$) increases on the database server, each of the device utilizations also increases linearly as a function of their individual $D_i$'s.

# Example 3.9..

- See Fig. 3.6. As indicated in the figure, the utilization of disk 3 will reach 100% before any other resource, because the utilization of this disk is always greater than that of other resources.
- That is, disk3 is the system's bottleneck. When the system load increases to a point where disk 3's utilization reaches 100%, the throughput cannot be increased any further.
- Since $X_0 = U_{disk3}/D_{disk3}$, $X_0 \leq 1/D_{disk3}$.
- Therefore, the maximum throughput, $X_0^{max} = 1/D_{disk3}$ = 1/0.142 = 7.04 tps.

Figure 3.6. Utilization vs. Throughput for Example 3.9.



# Bounds on Performance

- This example demonstrates that

$$X_O = \frac{U_i}{D_i} \le \frac{1}{D_i} \quad for\ all\ resources\ i \qquad (3.3.17)$$

- The resource with the largest service demand will have the highest utilization and is, therefore, the system's bottleneck.
- This bottleneck device yields the lowest (upper bound) value for the ratio *1/Di*. Therefore,

$$X_O \le \frac{1}{\max\{D_i\}} \qquad (3.3.18)$$

# Bounds on Performance.

- Now consider Little's Law applied to the same database server and let $N$ be the number of concurrent transactions in execution. Via Little's Law, $N = R \times X_0$.
- But, for a system with $K$ resources, the response time $R$ is at least equal to the sum of service demands, $\sum_{i=1}^{k} D_i$ , when there is no queuing. Thus,

$$N = R \times X_o \geq \left( \sum_{i=1}^{k} D_i \right) \times X_O \qquad (3.3.19)$$

- which can be rewritten as

$$X_O \leq \frac{N}{\sum_{i=1}^{k} D_i} \qquad (3.3.20)$$

# Bounds on Performance..

- the upper asymptotic bounds are:

$$X_O \le min\left[\frac{1}{max\{D_i\}}, \frac{N}{\sum_{i=1}^{k} D_i}\right] \quad (3.3.21)$$

- To illustrate these bounds, consider the same database server in Examples 3.2 and 3.4.
- Consider the two lines (i.e., from Eq. (3.3.21)) that bound its throughput as shown in Fig. 3.7.
- The line that corresponds to the light load bound is the line $N$ / 0.421 (solid line with solid diamonds).
- The horizontal line at 7.04 tps (solid line with unfilled diamonds) is the heavy load bound for this case.

# Figure 3.7. Bounds on throughput example. Original system

$$\frac{1}{max\{D_i\}} = 7.04$$

$$\frac{N}{\sum_{i=1}^{k} D_i} = \frac{N}{0.421}$$

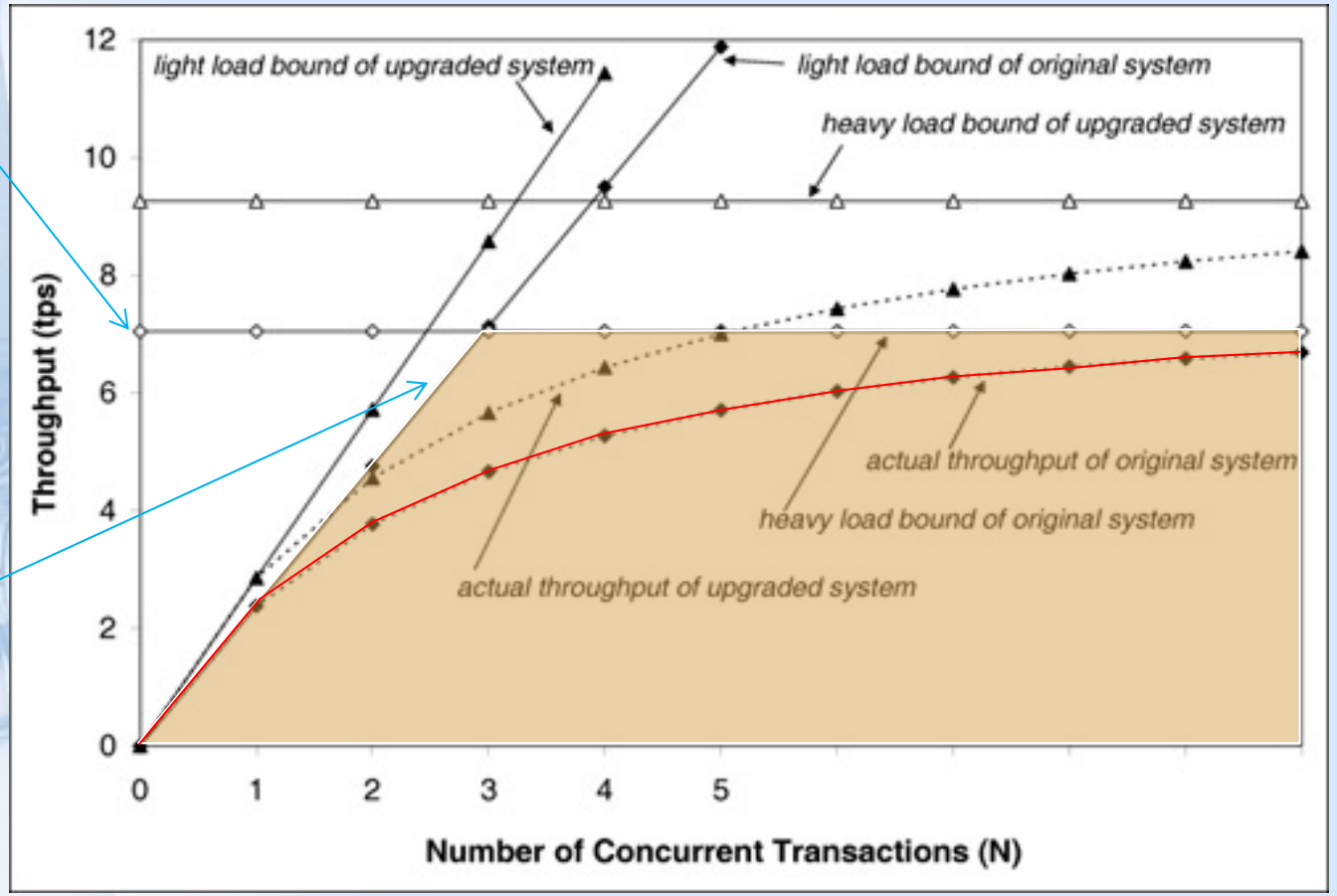
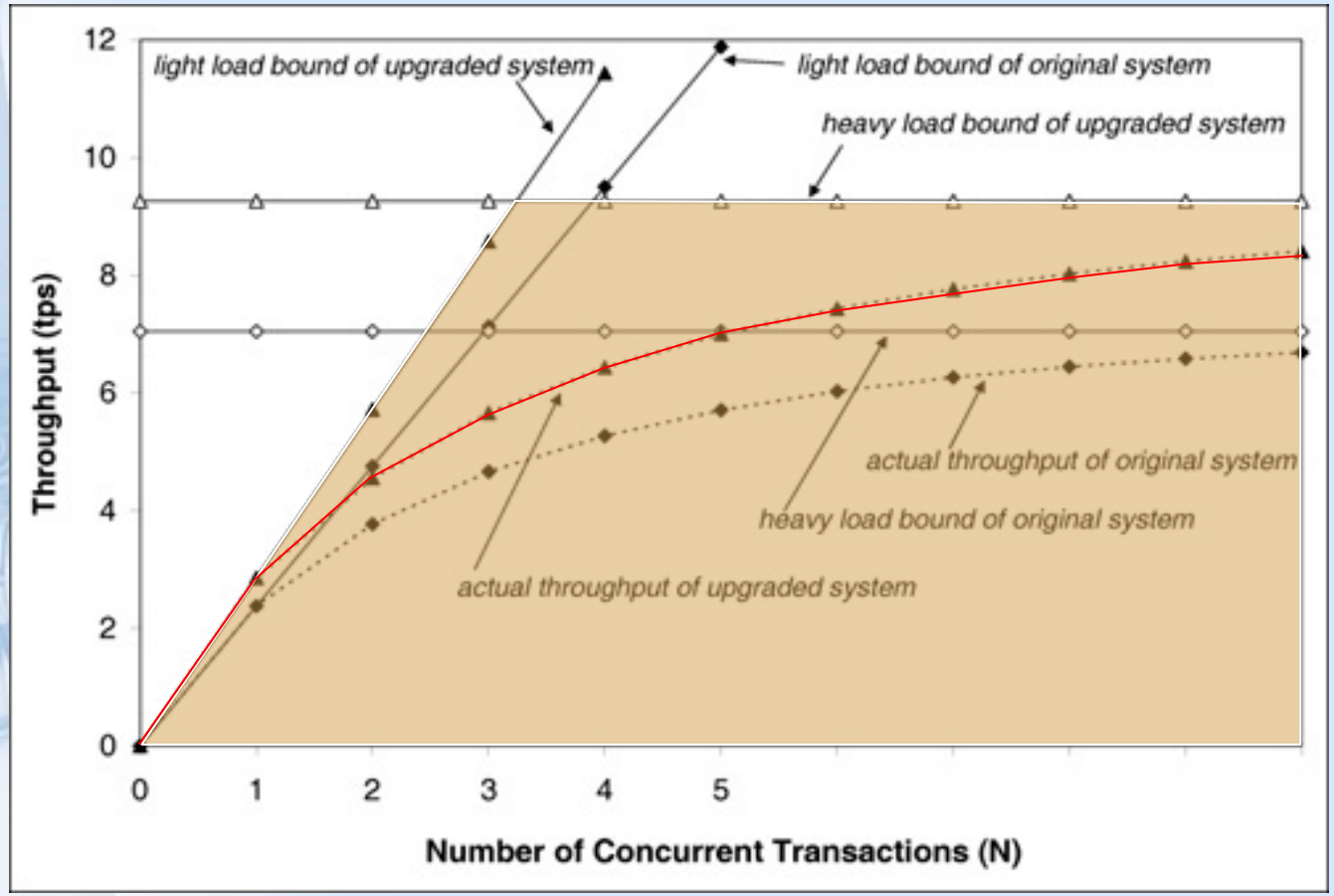


Figure 3.7. Bounds on throughput example. Upgraded system.



# Bounds on Performance...

- Then, the sum of the service demands becomes 0.35 (= 0.092 + 0.079 + 0.108 + 0.071) sec.
- The maximum service demand is now that of disk 2, the new bottleneck, and the new heavy load bound (i.e., the inverse of the maximum service demand) is now 9.26 (= 1/0.108) tps.
- Note that when the bottleneck resource was upgraded by a factor of two, the maximum throughput improved only by 32% (from 7.04 tps to 9.26 tps).

- Disk 2 became the new bottleneck.
- It would have been sufficient to upgrade disk 3 by a factor of 1.32 (= 0.142/0.108) instead of 2 to make its service demand equal to that of disk 2.

# Example 3.10

- Consider the same database server of Examples 3.2 and 3.4. Let the service demand at the CPU be fixed at 0.092 sec.
- What should be the values of the service demands of the three disks to obtain the maximum possible throughput, while maintaining constant the sum of the service demands at the three disks?
- Note that this is a load balancing problem (i.e., the goal is to maximize the throughput by simply shifting the load among the three disks).
- As demonstrated, the maximum service demand determines the maximum throughput.

# Example 3.10.

- Since the CPU is not the bottleneck, the maximum throughput is obtained when the service demands on all three disks is the same and equal to the average of the three original values.
- This is the balanced disk solution.
- In other words, the optimal solution occurs when $D_{disk1} = D_{disk2} = D_{disk3}$ = (0.079 + 0.108 + 0.142)/3 = 0.1097 sec.
- In this case, the maximum throughput is 9.12 (= 1/0.1097) tps.
- Maximum throughput can be expanded to increase 29.5% (i.e., from 7.04 tps to 9.12 tps) simply by balancing the load on the three existing disks.

# Example 3.10.

- To be convinced that the balanced disk solution is the optimal solution, assume that all disks have a service demand equal to D seconds.
- Now, increase the service demand of one of them by $\epsilon$ seconds, for $\epsilon > 0$.
- Since the sum of the service demands is to be kept constant, the service demand of at least one other disk has to be reduced in such a way that the sum remains the same.
- The disk that had its service demand increased will now have the largest service demand and becomes the bottleneck.
- new maximum throughput would be $1/(D + \epsilon) < 1/D$.

# Example 3.10..

- Thus, by increasing the service demand on one of the disks the maximum throughput decreases.
- Similarly, suppose that the service demand of one of the disks is decreased.
- Then, the service demand of at least one of the other disks will have to increase so that the sum remains constant.
- The service demand of the disk that has the largest increase limits the throughput.
- Let $D + \delta$, for $\delta > 0$, be the service demand for the disk with the new largest demand.
- Then, the maximum throughput is now equal to $1/(D + \delta) < 1/D$.

# Response Time Lower Bound

- Now consider a lower bound on the response time.
- According to Little's Law, the response time R is related to the throughput as $R = N/X_0$.
- By replacing $X_0$ by its upper bound given in Eq. (3.3.21), the following lower bounds for the response time can be obtained.

$$R = \frac{N}{X_o} \geq \frac{N}{\min\left[\frac{1}{\max\{D_i\}}, \frac{N}{\sum_{i=1}^{k} D_i}\right]} = \tag{3.3.22}$$

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

# Example 3.11

- Consider the same database server as before. What is the lower bound on response time?
- The sum of the service demands is 0.421 (= 0.092+0.079+0.108+0.142) and the maximum service demand is 0.142 sec. Therefore, the response time bounds are given by

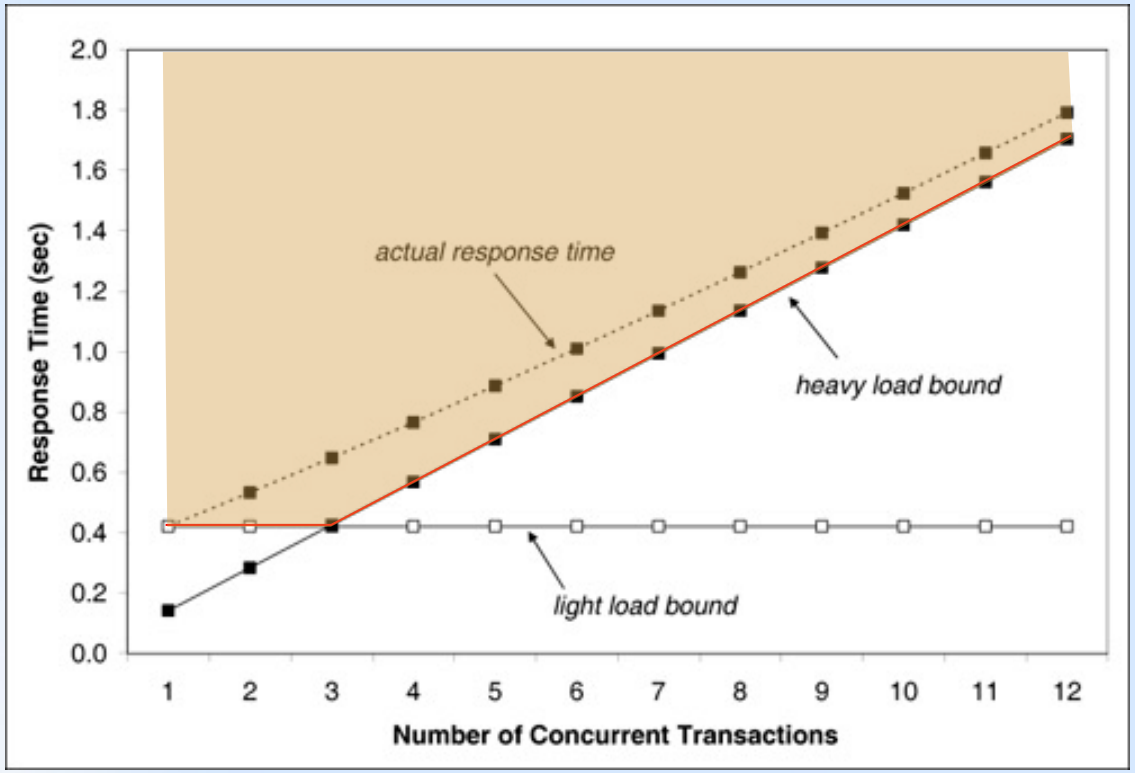
$$R \geq max[0.142 \times N, 0.421] \quad (3.3.23)$$

- These bounds are illustrated in Fig. 3.8, which also shows the actual response time curve.

# Example 3.11.

- The actual values of the response time are obtained by solving a closed QN model (see Chapter 12) with the help of the enclosed ClosedQN.XLS MS Excel workbook.
- As seen, as the load on the system increases, the actual response time approaches the heavy load response time bound quickly.

Figure 3.8. Bounds on response time example.



# Using QN Models

- One of the most important aspects in using QN models to predict performance is to understand
  - what models to use and
  - how to obtain the data for the model.

- In Chapter 2, different types and uses of QN models (open, closed, single class, or multiclass) were discussed.
- Numerical examples that illustrate the process are provided here.

# Example 3.12

- A Web server, composed of a single CPU and single disk, was monitored for one hour.
- The main workload of the server can be divided into HTML files and requests for image files.
- During the measurement interval 14,040 requests for HTML files and 1,034 requests for image files are processed.
- An analysis of the Web server log shows that HTML files are 3,000-bytes long and image files are 15,000-bytes long on average.
- The average disk service time is 12 msec for 1,000-byte blocks.

# Example 3.12.

- The CPU demand, in seconds, per HTTP request, is given by
- the expression $CPUDemand = 0.008 + 0.002 \times RequestSize$, where $RequestSize$ is given in the number of 1000-byte blocks processed.
- This expression for the CPU demand indicates that there is a constant time associated to processing a request (i.e., 0.008 seconds) regardless of the size of the file being requested.

# Example 3.12..

- This constant time involves
  - opening a TCP connection,
  - analyzing the HTTP request, and
  - opening the requested file.
- The second component of the CPU demand is proportional to the file size since the CPU is involved in each I/O operation.
- What is the response time for HTML and image file requests for the current load and for a load five times larger? Since the workload is characterized as being composed of two types of requests, a two-class queuing network model is required.

# Example 3.12...

- Should an open or closed model be used?
- The answer depends on how the workload intensity is specified.
- In this example, the load is specified by the number of requests of each type processed during the measurement interval. In other words, the arrival rate for each type of requests is:

(3.4.24)

$$\lambda_{HTML} = 14{,}040/3{,}600 = 3.9\, requests/\text{sec}, \text{ and}$$

$$\lambda_{image} = 1{,}034/3{,}600 = 0.29\, requests/\text{sec}.$$

# Example 3.12....

- This workload intensity is constant and does not depend on a fixed number of customers.
- Therefore, an open QN model as described in Chapter 13 is chosen
- The next step is to compute the
  - service demands for the CPU and
  - disk for HTML and image file requests.
- Using the expression for CPU time, the service demand for the CPU for HTML and image requests can be computed by using the corresponding file sizes in 1,000-byte blocks for each case as:

# Example 3.12.....

- $D_{CPU,HTML}$ = 0.008 + 0.002 x 3 = 0.014 sec and
- $D_{CPU,\ image}$ = 0.008 + 0.002 x 15 = 0.038 sec.
- The disk service demand is computed by multiplying the number of blocks read for each type of request by the service time per block.
- That is, $D_{disk,HTML}$ = 3x0.012 = 0.036 sec and
- $D_{disk,image}$ = 15 x 0.012 = 0.18 sec.
- By entering this data into the MS Excel OpenQN.XLS workbook that comes with this book and
- solving the model, the results in Table 3.3 are obtained.

# Example 3.12……

- In the case of open models, the throughput is equal to the arrival rate.
- Consider what happens under a five-fold increase in the load.
- The arrival rates become
- $\lambda_{HTML}$ = 5 x 3.9 = 19.5 requests/sec and
- $\lambda_{image}$ = 5 x 0.29 = 1.45 requests/sec.

## Table 3.3. Service Demands, Arrival Rates, and Performance Metrics for Ex. 3.12

| | html | image |
|---|---|---|
| Arrival rate (req/sec) | 3.90 | 0.29 |
| Service demands(sec) | | |
| Cpu | 0.014 | 0.038 |
| Disk | 0.036 | 0.180 |
| Utilizations(%) | | |
| cpu | 5.5 | 1.1 |
| disk | 14 | 5.2 |
| Residence times (sec) | | |
| Cpu | 0.015 | 0.041 |
| disk | 0.045 | 0.223 |
| Response time (sec) | 0.060 | 0.264 |

# Example 3.12.....

- Solving the model with these values of the arrival rates, new response times of 0.93 sec for HTML and 4.61 sec for image requests are obtained.
- Thus, image file requests experience an increase in their response time by a factor of 17.5 and requests for HTML files experience a response time increased by a factor of 15.5.
- At the new load level, the disk utilization reaches 96% as indicated by the model, up from its previous 19.2% utilization (i.e., 14% + 5.2%).
- This indicates that the original system has excess capacity, but a five-fold load increase is nearing its maximum capacity.

# Example 3.13

- Reconsider the Web server of Example 3.12.
- What is the response time and throughput of HTML and image file requests when there is an average of 14 HTML requests and 6 image file requests being executed concurrently at all times?
- In this case, the workload is specified by a number of concurrent requests in execution and not by an arrival rate.
- In this situation, a closed multiclass QN model (described in Chapter 13) is now appropriate. This model can be solved using the MS Excel workbook ClosedQN.XLS.

# Example 3.13.

- The service demands are the same as in Example 3.12.
- Solving the model,
  - $R_{HTML}$ = 0.72 sec,
  - $R_{image}$ = 3.57 sec,
  - $X_{HTML}$ = 19.3 requests/sec, and
  - $X_{image}$ = 1.7 requests/sec.
- By comparing these results against these in Example 3.12,
- when the workload is increased five-fold, similar performance magnitudes are observed.

# Concluding Remarks

- Chapter 2 described the various types of performance models from a qualitative point of view.
- In this chapter, these models are quantified.
- A set of very important relationships between performance variables is introduced.
- These relationships, called Operational Laws, are quite general (i.e., robust) and are extremely useful because:
  - i) they are very simple,
  - ii) they are based on readily available measurement data, and
  - iii) they can be used to obtain helpful performance metrics.

# Concluding Remarks.

- Simple bounding techniques were introduced and used to obtain upper bounds on throughput and lower bounds on response time from service demands.
- Examples were presented of applying QN models to various performance situations.
- In the following chapters of Part I the set of applications of performance models is expanded.
- The models used here are described in Part II and are implemented using the tools included.

# Exercises (1)

- A computer system is monitored for one hour. During this period, 7200 transactions were executed and the average multiprogramming level is measured to be equal to 5 jobs.

  What is the average time spent by a job in the system once it is in the multiprogramming mix (i.e., the average time spent by the job in the system once it is memory resident)?

# Exercises (2)

- Measurements taken during one hour from a Web server indicate that the utilization of the CPU and the two disks are: $U_{CPU}$ = 0.25, $U_{disk1}$ = 0.35, and $U_{disk2}$ = 0.30. The Web server log shows that 21,600 requests were processed during the measurement interval. What are the service demands at the CPU and both disks, what is the maximum throughput, and what was the response time of the Web server during the measurement interval?

# Exercises (3, 4)

- 3. Consider the Web server of Exercise 3.2. Draw a graph of the Web server's throughput as a function of the number of concurrent requests. Comment on observations.
- 4. A computer system is measured for 30 minutes. During this time, 5,400 transactions are completed and 18,900 I/O operations are executed on a certain disk that is 40% utilized. What is the average number of I/O operations per transaction on this disk? What is the average service time per transaction on this disk?

# Exercises(5, 6)

- 5. A transaction processing system is monitored for one hour. During this period, 5,400 transactions are processed. What is the utilization of a disk if its average service time is equal to 30 msec per visit and the disk is visited three times on average by every transaction?
- 6. The average delay experienced by a packet when traversing a computer network is 100 msec. The average number of packets that cross the network per second is 128 packets/sec. What is the average number of concurrent packets in transit in the network at any time?

# Exercises (7, 8)

- 7. A file server is monitored for 60 minutes, during which time 7,200 requests are completed. The disk utilization is measured to be 30%. The average service time at this disk is 30 msec per file operation request. What is the average number of accesses to this disk per file request?
- 8. Consider the database server of Example 3.2. Using ClosedQN.XLS, what is the throughput of the database server, its response time, and the utilization of the CPU and the three disks, when there are 5 concurrent transactions in execution?

# Exercises (9)

- 9. A computer system has one CPU and two disks: disk 1 and disk 2. The system is monitored for one hour and the utilization of the CPU and of disk 1 are measured to be 32% and 60%, respectively. Each transaction makes 5 I/O requests to disk 1 and 8 to disk 2. The average service time at disk 1 is 30 msec and at disk 2 is 25 msec.

# Exercises (9)

- Find the system throughput.
- Find the utilization of disk 2.
- Find the average service demands at the CPU, disk 1, and disk 2.
- Use ClosedQN.XLS to find the system throughput, response time, and average queue length at the CPU and the disks when the degree of multiprogramming is n, for n = 0, ..., 4.
- Based on the above results, what is a good approximation for the average degree of multiprogramming during the measurement interval?

# Exercises (10)

- An interactive system has 50 terminals and the user's think time is equal to 5 seconds. The utilization of one of the system's disk was measured to be 60%. The average service time at the disk is equal to 30 msec. Each user interaction requires, on average, 4 I/Os on this disk. What is the average response time of the interactive system?

# Exercises (11)

- Obtain access to a UNIX or Linux machine and become acquainted with the command iostat, which displays information on disk and CPU activity. The data in Table 3.4 shows a typical output report from iostat. Each line displays values averaged over 5-second intervals. The first three columns show activity on disk sd0. The kps column reports KB transferred per second, the tps column shows the number of I/Os per second, and the serv column shows average disk service time in milliseconds. The next four columns display CPU activity. The us column shows the percent of time the CPU spent in user mode. The next column shows the percent of time the CPU was in system mode followed by the percent of time the CPU was waiting for I/O. The last column is the percent of time the CPU was idle. Compute the disk and CPU utilizations.

# Exercises (9)

3.11

Table 3.4. Data for Exercise 3.9

| sd0 | | | cpu | | | |
|---|---|---|---|---|---|---|
| kps | tps | serv | us | sy | wt | id |
| 25 | 3 | 6 | 19 | 3 | 0 | 78 |
| 32 | 4 | 7 | 13 | 4 | 0 | 83 |
| 28 | 2 | 7 | 20 | 3 | 0 | 77 |
| 18 | 2 | 8 | 24 | 2 | 0 | 74 |
| 29 | 3 | 9 | 18 | 5 | 0 | 77 |
| 33 | 4 | 12 | 23 | 3 | 0 | 74 |
| 35 | 4 | 8 | 25 | 5 | 0 | 70 |
| 25 | 4 | 10 | 32 | 4 | 0 | 64 |
| 26 | 3 | 11 | 28 | 4 | 0 | 68 |
| 34 | 4 | 12 | 22 | 6 | 0 | 72 |

# Exercise (12)

- You are planning a load testing experiment of an e-commerce site. During the experiment, virtual users (i.e., programs that behave like real users and submit requests to the site) send requests to the site with an average think time of 5 seconds. How many virtual users you should have in the experiment to achieve a throughput of 20 requests/sec with an average response time of 2 seconds?

# Bibliography


- [1] P. J. Denning and J. P. Buzen, "The Operational analysis of queueing network models," Computing Surveys, vol. 10, No. 3, September 1978, pp. 225-261.
- [2] J. C. Little, "A Proof of the queueing formula L = lW," Operations Research, vol. 9, 1961, pp. 383–387.
- [3] R. R. Muntz and J. W. Wong, "Asymptotic properties of closed queuing network models," Proc. 8th Princeton Conf. Information Sciences and Systems, 1974.