

EEC 686/785
Modeling & Performance Evaluation of
Computer Systems

Lecture 24

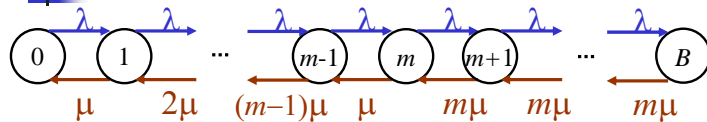
Wenbing Zhao

Department of Electrical and Computer Engineering
Cleveland State University
wenbing@ieee.org
(based on Dr. Raj Jain's lecture notes)

Outline

- Review of lecture 23
- Operational laws

Analysis of M/M/m/B Queue



- Birth-death process:

$$\lambda_n = \lambda \quad n = 0, 1, 2, \dots, B-1$$

$$\mu_n = \begin{cases} n\mu & n = 1, 2, \dots, m-1 \\ m\mu & n = m, m+1, \dots, B \end{cases}$$

Analysis of M/M/m/B Queue

- Probability of n jobs in the system:

$$p_n = \begin{cases} \frac{(m\rho)^n}{n!} p_0 & n = 1, 2, \dots, m-1 \\ \frac{\rho^n m^m}{m!} p_0 & n = m, m+1, \dots, B \end{cases}$$

- Probability of zero jobs in the system:

$$p_0 = \left[1 + \frac{(1 - \rho^{B-m+1})(m\rho)^m}{m!(1 - \rho)} + \sum_{n=1}^{m-1} \frac{(m\rho)^n}{n!} \right]^{-1}$$

Analysis of M/M/m/B Queue

- Mean number of jobs in the system: $E[n] = \sum_{n=1}^B np_n$
- Mean number of jobs in the queue: $E[n_q] = \sum_{n=m+1}^B (n-m)p_n$
- All arrivals occurring when the system is in the state $n=B$ are lost. Rate of the jobs actually entering the system, called **effective arrival rate**, is:

$$\lambda' = \sum_{n=0}^{B-1} \lambda p_n = \lambda \sum_{n=0}^{B-1} p_n = \lambda(1 - p_B)$$
- The difference $\lambda - \lambda' = \lambda p_B$ represents the **loss rate**

Analysis of M/M/m/B Queue

- Mean response time (using Little's law):

$$E[r] = \frac{E[n]}{\lambda'} = \frac{E[n]}{\lambda(1 - p_B)}$$

- Utilization of each server:

$$U = \frac{\text{Busy time per server}}{\text{Total time}} = \frac{(\lambda'T / \mu) / m}{T} = \frac{\lambda'}{m\mu} = \rho(1 - p_B)$$

- Probability of the system in full capacity = p_B

Queueing Networks

- Network = model in which jobs departing from one queue arrive at another queue (or possibly the same queue)
- Open queueing networks
 - External arrivals departures
- Closed queueing networks
 - No external arrivals or departures
- Mixed queueing networks
 - open for some workloads and closed for others

Product-Form Network

- Any queueing network in which:

$$P(n_1, n_2, n_3, \dots, n_M) = \frac{1}{G(N)} \prod_{i=1}^M f_i(n_i)$$

When $f_i(n_i)$ is some function of the number of jobs at the i th facility, $G(N)$ is a normalizing constant and is a function of the total number of jobs in the system

Non-Markovian Product Form Networks

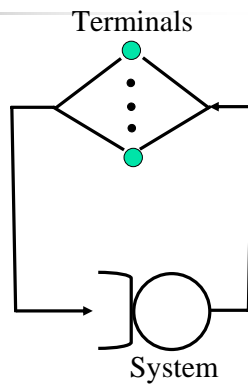
- **1. Job flow balance:** for each class, the number of arrivals to a device must equal the number of departures from the device
- **2. One step behavior:** no bulk arrivals and no simultaneous job-moves
- **3. Device homogeneity:** a device's service rate for a particular class does not depend on the state of the system in any way except for the total device queue length and the designated class's queue length

Non-Markovian Product Form Networks

- The assumption 3 implies the following:
 - **Single resource possession**
 - **No blocking**
 - **Independent job behavior**
 - **Local information**
 - **Fair service**
 - **Routing homogeneity**

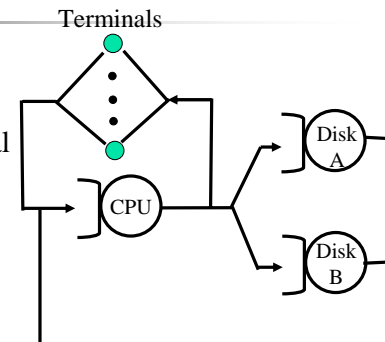
Machine Repairman Model

- One of the earliest model of computer systems
- Originally developed for modeling machine repair shops
 - A number of working machines
 - A repair facility with one or more servers (repairmen)
 - Whenever a machine breaks down, it is put in the queue for repair and serviced as soon as a repairman is available



Central Server Model

- Introduced by Buzen (1973)
- The CPU is the "central server" that schedules visits to other devices
- After service at the I/O devices the jobs return to the CPU



Types of Service Centers

- **Fixed-capacity service centers:** service time does not depend upon the number of jobs in the device
- **Delay centers** or IS (infinite server): no queueing. Jobs spend the same amount of time in the device regardless of the number of jobs in it
- **Load-dependent service centers:** service rates may depend upon the load or the number of jobs in the device

Operational Laws

- Relationships that do not require any assumptions about the distribution of service times or interarrival times
- Identified originally by Buzen (1976) and later extended by Denning and Buzen (1978)
- Operational => directly measured

Operational Laws

- **Operationally testable assumptions** => assumptions that can be verified by measurements
 - For example, whether number of arrivals is equal to the number of completions (i.e., job flow balance) is operationally testable
 - A set of observed service times is not operationally testable

Operational Laws

- **Operational quantities:** quantities that can be directly measured during a finite observation period
 - T = observation interval
 - A_i = number of arrivals
 - C_i = number of completions
 - B_i = busy time
- Operational quantities are *variables* that can change from one observation period to the next

Operational Laws

- Relationships that hold in every observation period are called **operational laws**

Operational quantities

$$\left\{ \begin{array}{l} \text{Arrival Rate } \lambda_i = \frac{\text{Number of arrivals}}{\text{Time}} = \frac{A_i}{T} \\ \text{Throughput } X_i = \frac{\text{Number of completions}}{\text{Time}} = \frac{C_i}{T} \\ \text{Utilization } U_i = \frac{\text{Busy Time}}{\text{Total Time}} = \frac{B_i}{T} \\ \text{Mean service time } S_i = \frac{\text{Total time served}}{\text{Number served}} = \frac{B_i}{C_i} \end{array} \right.$$

Utilization Law

$$\left\{ \begin{array}{l} U_i = \frac{B_i}{T} = \frac{C_i}{T} \times \frac{B_i}{C_i} \quad \text{or} \quad U_i = X_i S_i \end{array} \right.$$

Operational Laws

- Operational laws are similar to the elementary laws of motion
- For example, $d = \frac{1}{2} at^2$. Notice that distance d , acceleration a , and time t are operational quantities.
- There is no need to consider them as expected values of random variables or to assume a probability distribution for them

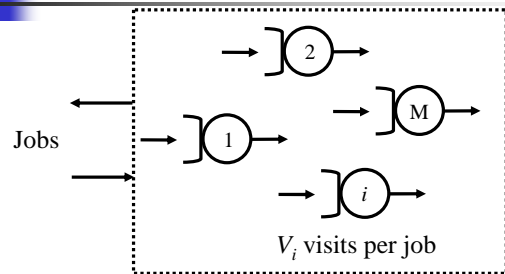
Example 33.1

- Consider a network gateway at which the packets arrive at a rate of 125 packets per second and the gateway takes an average of two milliseconds to forward them
 - Throughput $X_i = \text{exit rate} = \text{arrival rate} = 125 \text{ pps}$
 - Service time $S_i = 0.002 \text{ second}$
 - Utilization $U_i = X_i S_i = 125 \times 0.002 = 0.25 = 25\%$
 - This result is valid for any arrival or service process
 - Even if interarrival times and service times are not IID random variables with exponential distribution

Forced Flow Law

- Forced Flow Law:** relates the system throughput to individual device throughputs
- In an **open model**, system throughput = number of jobs leaving the system per unit time
- In a **closed model**, system throughput = number of jobs traversing the OUT to IN link per unit time
- If observation period T is such that $A_i = C_i$
=> Device satisfies the assumption of job flow balance

Forced Flow Law



- Each job makes V_i requests for i th device in the system:

$$C_i = C_0 V_i \quad \text{or} \quad V_i = \frac{C_i}{C_0}$$

V_i is called **visit ratio**

Forced Flow Law

- System throughput:

$$\text{System throughput } X = \frac{\text{Jobs completed}}{\text{Total time}} = \frac{C_0}{T}$$

- Throughput of i th device:

$$\text{Device throughput } X_i = \frac{C_i}{T} = \frac{C_i}{C_0} \times \frac{C_0}{T} \quad \text{or} \quad X_i = X V_i$$

This is the **forced flow law**

Forced Flow Law

- Combining the forced flow law and the utilization law, we get:

$$\text{Utilization of } i\text{th device } U_i = X_i S_i = X V_i S_i$$

$$\text{or } U_i = X D_i$$

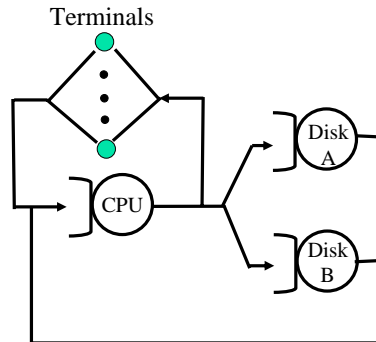
- Here $D_i = V_i S_i$ is the total service demand on the device for all visits of a job
- The device with the highest D_i has the highest utilization and is the **bottleneck device**

Example 33.2

- In a timesharing system, accounting log data produced the following profile for user programs
 - Each program requires five seconds of CPU time, makes 80 I/O requests to the disk A and 100 I/O requests to disk B
 - Average think-time of the users was 18 seconds
 - From the device specifications, it was determined that disk A takes 50 ms to satisfy an I/O request and the disk B takes 30 ms per request
 - With 17 active terminals, disk A throughput was observed to be 15.70 I/O requests per second
 - We want to find the system throughput and device utilization**

Example 33.2

- $D_{CPU} = 5$ seconds
- $V_A = 80$
- $V_B = 100$
- $Z = 18$ seconds
- $S_A = 0.050$ seconds
- $S_B = 0.030$ seconds
- $N = 17$
- $X_A = 15.70$ jobs/second



Example 33.2

- Since the jobs must visit the CPU before going to the disks or terminals, the CPU visit ratio is:

$$V_{CPU} = V_A + V_B + 1 = 181$$

- Total service demand on the disk A and disk B

$$D_A = S_A V_A = 0.050 \times 80 = 4 \text{ seconds}$$

$$D_B = S_B V_B = 0.030 \times 100 = 3 \text{ seconds}$$

Example 33.2

- Using the forced flow law, the throughputs are:

$$X = \frac{X_A}{V_A} = \frac{15.70}{80} = 0.1963 \text{ jobs/second}$$

$$X_{CPU} = X V_{CPU} = 0.1963 \times 181 = 35.48 \text{ requests/second}$$

$$X_B = X V_B = 0.1963 \times 100 = 19.6 \text{ requests/second}$$

- Using the utilization law, the device utilizations are:

$$U_{CPU} = X D_{CPU} = 0.1963 \times 5 = 98\%$$

$$U_A = X D_A = 0.1963 \times 4 = 78.4\%$$

$$U_B = X D_B = 0.1963 \times 3 = 58.8\%$$

Transition Probabilities

- p_{ij} = probability a job moving to j th queue after service completion at i th queue
- Visit ratios and transition probabilities are equivalent in the sense that given one we can always find the other
- In a system with job flow balance, the number of completions at j th queue:

$$C_j = \sum_{i=0}^M C_i p_{ij}$$

- Subscript 0 is used to denote visits to the outside link
- p_{i0} = probability of a job exiting from the system after completion of service at i th device

Transition Probabilities

- Dividing both sides by C_0 we get:

$$V_j = \sum_{i=0}^M V_i p_{ij}$$

- Since each visit to the outside link is defined as the completion of the job, we have:

$$V_0 = 1$$

- The above two equations are called **visit ratio equations**

Transition Probabilities

- In central server models, after completion of service at every queue, the jobs always move back to the CPU queue:

$$p_{i1} = 1 \quad \forall i \neq 1$$

$$p_{ij} = 0 \quad \forall i, j \neq 1$$

- The above probabilities apply to exit and entrance from the system ($i=0$) also. Therefore, the visit ratio equations become:

$$1 = V_1 p_{10}$$

$$V_j = V_1 p_{1j} = \frac{p_{1j}}{p_{10}} \quad j = 2, 3, \dots, M$$

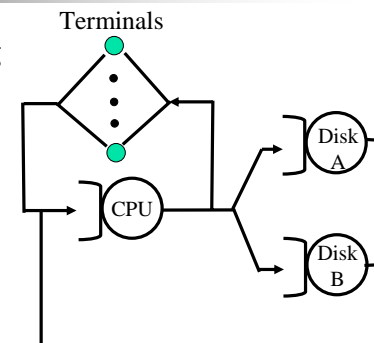
Transition Probabilities

- Thus, we can find the visit ratios by dividing the probability p_{1j} of moving to j th queue from CPU by the exit probability p_{10}

Example 33.3

- Consider the queueing network:

- The visit ratios are $V_A = 80$, $V_B = 100$, and $V_{CPU} = 181$



Example 33.3

- After completion of service at the CPU, the probabilities of the job moving to disk A, disk B, or terminals are 80/181, 100/181, and 1/181, respectively. Thus, the transition probabilities are 0.4420, 0.5525, and 0.005525
- Given the transition probabilities, we can find the visit ratios by dividing these probabilities by the exit probability (0.005525)

$$V_A = \frac{0.4420}{0.005525} = 80$$

$$V_B = \frac{0.5525}{0.005525} = 100$$

$$V_{CPU} = 1 + V_A + V_B = 1 + 80 + 100 = 181$$

Little's Law

- Mean number in the device = arrival rate × mean time in the device

$$Q_i = \lambda_i R_i$$

- If the job flow is balanced, the arrival rate is equal to the throughput and we can write:

$$Q_i = X_i R_i$$

Example 33.4

- The average queue length in the computer system of example 33.2 was observed to be: 8.88, 3.19, and 1.40 jobs at the CPU, disk A, and disk B, respectively. What were the response times of these devices?

- In example 33.2, the device throughputs were determined to be

$$X_{CPU} = 35.48, X_A = 15.70, \text{ and } X_B = 19.6$$

- The new information given in this example is:

$$Q_{CPU} = 8.88, Q_A = 3.19, \text{ and } Q_B = 1.40$$

Example 33.4

- Using Little's law, the device response times are:

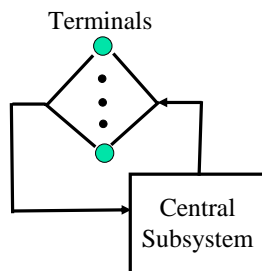
$$R_{CPU} = Q_{CPU} / X_{CPU} = 8.88 / 35.48 = 0.250 \text{ seconds}$$

$$R_A = Q_A / X_A = 3.19 / 15.70 = 0.203 \text{ seconds}$$

$$R_B = Q_B / X_B = 1.40 / 19.6 = 0.071 \text{ seconds}$$

General Response Time Law

- There is one terminal per user and the rest of the system is shared by all users
- Applying Little's law to the central subsystem:
 $Q = XR$
 Here, Q = total number of jobs in the system
 R = system response time
 X = system throughput



General Response Time Law

$$Q = Q_1 + Q_2 + \dots + Q_M$$

$$XR = X_1R_1 + X_2R_2 + \dots + X_MR_M$$

- Dividing both sides by X and using forced flow law:

$$R = V_1R_1 + V_2R_2 + \dots + V_MR_M$$

$$\text{or, } R = \sum_{i=1}^M R_i V_i$$

This is called the **general response time law**. This law holds even if the job flow is not balanced

Example 33.5

- Let us compute the response time for the timesharing system of example 33.2 and 33.4
- For this system:
 $V_{CPU} = 181, V_A = 80, V_B = 100$
 $R_{CPU} = 0.250, R_A = 0.203, R_B = 0.071$
- The system response time is:
 $R = R_{CPU}V_{CPU} + R_A V_A + R_B V_B$
 $= 0.250 \times 181 + 0.203 \times 80 + 0.071 \times 100 = 68.6$
 The system response time is 68.6 seconds

Interactive Response Time Law

- If Z = think-time, R = response time, the total cycle time of requests is $R + Z$
- Each user generates about $T/(R + Z)$ requests in T
- If there are N users, the system throughput X = total number of requests / total time
 $= N[T/(R+Z)]/T$
 $= N/(R+Z)$
 or

$$R = (N/X) - Z$$

This is the **interactive response time law**

Example 33.6

- For the timesharing system of example 33.2, we can compute the response time using the interactive response time law as follows:

$$X = 0.1963, N = 17, Z = 18$$

Therefore:

$$R = \frac{N}{X} - Z = \frac{17}{0.1963} - 18 = 86.8 - 18 = 68.6 \text{ seconds}$$

This is the same as that obtained earlier in example 33.5

Bottleneck Analysis

- From forced flow law: $U_i \propto D_i$
the device with the highest total service demand D_i has the highest utilization and is called the bottleneck device
 - Note: delay centers can have utilizations more than one without any stability problems. Therefore, delay centers cannot be a bottleneck device
 - Only queueing centers used in computing D_{max}
- The bottleneck device is the key limiting factor in achieving higher throughput

Bottleneck Analysis

- Improving the bottleneck device will provide the highest payoff in terms of system throughput
- Improving other devices will have little effect on the system performance
- Identifying the bottleneck device should be the first step in any performance improvement project

Bottleneck Analysis

- Throughput and response times of the system are bound as follows:

$$X(N) \leq \min \left\{ \frac{1}{D_{max}}, \frac{N}{D + Z} \right\}$$

$$R(N) \geq \max \{ D, ND_{max} - Z \}$$

Here, $D = \sum D_i$ is the sum of total service demands on all devices except terminals

- These are known as **asymptotic bounds**

Proof

- The asymptotic bounds are based on the following observations:
 - The utilization of any device cannot exceed one. This puts a limit on the maximum obtainable throughput
 - The response time of the system with N users cannot be less than a system with just one user. This puts a limit on the minimum response time
 - The interactive response time formula can be used to convert the bound on throughput to that on response time and vice versa

Proof

- For the bottleneck device b we have:

$$U_b = XD_{\max}$$

Since U_b cannot be more than 1, we have:

$$XD_{\max} \leq 1 \quad \text{or} \quad X \leq \frac{1}{D_{\max}}$$

- With just one job in the system, there is no queuing and the system response time is simply the sum of the service demands:

$$R(1) = D_1 + D_2 + \dots + D_M = D$$

Proof

- Here, D is defined as the sum of all service demands. With more than one user there may be some queuing and so the response time will be higher. Therefore:

$$R(N) \geq D$$

- Applying the interactive response time law to the bounds:

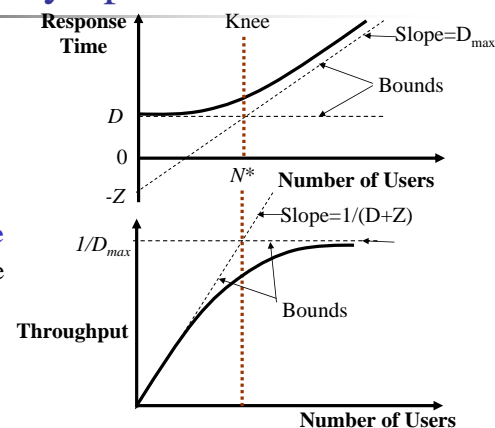
$$R(N) = \frac{N}{X(N)} - Z \geq ND_{\max} - Z$$

$$X(N) = \frac{N}{R(N) + Z} \leq \frac{N}{D + Z}$$

Combining these bounds we get the asymptotic bounds

Typical Asymptotic Bounds

- Both *throughput* and *response time* bounds consist of two straight lines
- The point of intersection of the two lines is called the **knee**
- For both response time and throughput, the **knee** occurs at the same value of number of users



Typical Asymptotic Bounds

- The number of jobs N^* at the knee is given by:

$$D = N^* D_{\max} - Z$$

$$\Rightarrow N^* = \frac{D + Z}{D_{\max}}$$

- If the number of jobs is more than N^* , then we can say with certainty that there is queueing somewhere in the system
- The asymptotic bounds can be easily explained to people who do not have any background in queueing theory or performance analysis

Example 33.7

- For the timesharing system considered in example 33.2:

$$D_{CPU} = 5, D_A = 4, D_B = 3, Z = 18$$

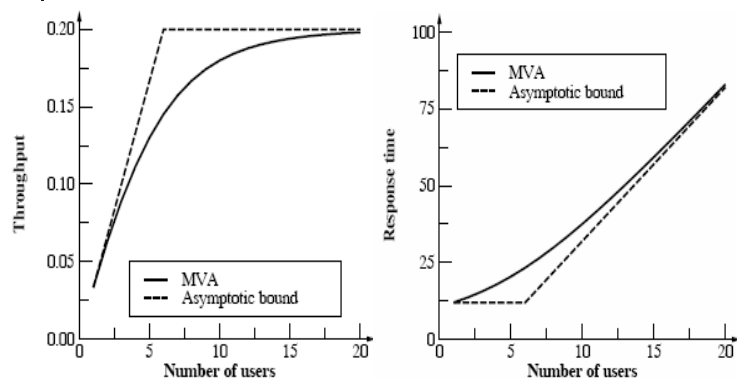
$$D = D_{CPU} + D_A + D_B = 5 + 4 + 3 = 12, D_{\max} = D_{CPU} = 5$$

- The asymptotic bounds are:

$$X(N) \leq \min \left\{ \frac{N}{D + Z}, \frac{1}{D_{\max}} \right\} = \min \left\{ \frac{N}{30}, \frac{1}{5} \right\}$$

$$R(N) \geq \max \{ D, N D_{\max} - Z \} = \max \{ 12, 5N - 18 \}$$

Example 33.7



Example 33.7

- The knee occurs at:

$$12 = 5N^* - 18$$

$$\Rightarrow N^* = \frac{12 + 18}{5} = \frac{30}{5} = 6$$

- Thus, if there are more than 6 users on the system, there will certainly be queueing in the system



Example 33.8

- How many terminals can be supported on the timesharing system of example 33.2 if the response time has to be kept below 100 seconds
 - Using the asymptotic bounds on the response time we get
$$R(N) \geq \max\{12, 5N - 18\}$$
 - The response time will be more than 100, if:
$$5N - 18 \geq 100 \Rightarrow N \geq 23.6$$
 - The response time is bound to be more than 100. Thus, the system cannot support more than 23 users if a response time of less than 100 is required