Outline for Today

- Three very simple, useful & timeless models
  - single queue, "machine repair model", "central server"
- Two types of systems
  - Open, transaction system
  - Closed, interactive (or batch) system
- Output Performance Measures
  - overall, per-queue
- Modeling methodology

Outline for Today

- Three very simple, useful & timeless models
  - single queue, "machine repair model", "central server"
- Two types of systems
  - Open, transaction system
  - Closed, interactive (or batch) system
- Output Performance Measures
  - overall, per-queue
- Modeling methodology

Model #1: Single-server Queue

A very general model of a shared resource

- Examples
  - a congested link in an Internet path
    - workload: arrival time & size for each arriving packet
    - system parameters: link scheduling discipline, link speed
  - a congested disk in a Web server
    - workload: read and write requests that miss in disk cache
    - system parameters: disk request scheduling discipline, disk speeds for seek, rotation, data transfers
  - a congested station in a "just in time" manufacturing system
    - workload spec: a system trace or a statistical model/generator

Model #1: Key Input Measures

The 2 most important workload/system parameters:

- arrival rate, \( \lambda \)
- average service time, \( S \)

Further input parameters:

- scheduling discipline, buffer size
- standard deviation of service time \( \sigma_S \) & of interarrival time
- more detailed measures of service times & interarrival times

\( S \) is derived from workload & system parameters

- e.g., \( S = \frac{\text{average packet size (bits)}}{\text{link speed (bits/second)}} \)

Model #1: Input Measures

- arrival rate, \( \lambda \)
- average service time, \( S \)

- in a statistical model, \( \lambda \) and \( S \) can be measured values, or larger values to study how performance will change for higher load
- \( \lambda \) and \( S \) should be measured during a peak stationary period
  - (e.g., when the averages are not varying and are at their peak values)
  - e.g., for the O2K job trace in assignment #1, measure \( \lambda \) and \( S \) during hours 12-18, but not over entire 24-hours
  - for simulation with trace input: use the peak period trace
Sources of Measures

- **Workload traces** (often incomplete):
  - System scheduler/accounting logs (e.g., NCSA O2K)
  - Application scheduler logs (e.g., Microsoft Media Server)
  - Software monitor (e.g., tcpdump)

- **System measures**:
  - Hardware specifications
  - Software monitors
  - Hardware monitors

(LZGS text, chapters 12-14 -- FYI)

**Model #2: “machine repair model”**

Another very general model of a shared resource

- *N* customers cycle in the network
  - e.g., *N* = 500

- Example:
  - The shared server performs machine repair
  - Fixed set of *N* clients
    - at delay center, client machine is operational
    - when the machine fails, the client moves to the queue

- Workload: operational times, repair times (trace or statistical)

- Delay center (each customer has their own server)

- Another very general model of a shared resource
  - Software monitor (e.g., tcpdump)

**Model #2: a second example**

A model of a shared resource

- *N* customers cycle in the network
  - e.g., *N* = 20

- Example:
  - The shared server is a bottleneck disk in a file server
  - That serves a fixed number of workstations

- Workload: times between disk accesses, disk read/write requests

- Workload spec: a set of traces or a statistical model/generator

**Model #2: a third example**

A model of a shared resource

- *N* customers cycle in the network
  - e.g., *N* = 8

- Example:
  - The shared server is a multiprocessor O.S. semaphore
  - Fixed set of *N* clients
    - at delay center, client machine is operational
    - when the machine fails, the client moves to the queue

- Workload: execution times in critical/non-critical mode (trace or model)

- Delay center (semaphore is a shared software resource)

**Model #2: Key Input Parameters**

The 3 most important system/workload parameters:

- *N* = average service time (seconds)
- *Z* = average delay (seconds)

- Further input parameters:
  - Scheduling discipline for the shared server
  - More detailed measures of delays & service times

In some cases, we can solve for general classes of these values

- Measure *N*, *Z*, *S* during peak load stationary periods

**Model #3: “Central Server”**

An server with a “transaction workload”:

- Example:
  - A Web server: request arrival generates one or more disk accesses; contention occurs at CPU & disks

- Key input parameters: *\( \lambda \)*, *(S)* and routing probabilities
  - (parameters of a server request trace or a statistical model)

- Use/measure these inputs for peak load stationary periods
Model #3: "Central Server"

A server with a finite client population & "interactive workload".

Example:
- A file server shared by N workstations
  - Z is the average time between a response from file server and when the workstation generates its next request to the server
  - Measure N, Z, (Sk) & routing during peak load stationary periods

Delay Centers

- Delay center can be represented in either of two ways
- Open or closed model can have any number of delay centers
- A delay center outside the "system" of interest is called a "think node" (due to early application in modeling timesharing systems)

Transactions with "think node"

- Transaction workload can have a "think node" Z = average think time (input measure)
- System is still open - sessions arrive at rate λ

"Batch Systems"

- No think node (but possibly other delay centers)
- A queue of customers waiting to enter the system is not shown, but it always has at least one customer waiting (in the model)
- Depoing customer is immediately replaced by a customer waiting to enter the system

Average Demands

- \( V_k = \) average no. of visits to node \( k \)
  - e.g., \( V_{cpu} = 100 \), \( V_{disk} = 40 \)
- \( D_k = V_k S_k = \) average total demand at node \( k \)
  - e.g., \( S_{cpu} = 10 \text{ msec}, D_{cpu} = 1 \text{ sec} \)

Open versus Closed: Question

- Need to know what to collect in workload trace
  - i.e., collect trace of server request arrival times (open)
  - Trace of "think times" for each client (closed)
- Need to know what to measure/specify for statistical model
  - i.e., time between requests to server or per client "think time"
- A common mistake is to use the wrong workload type
Open versus Closed Systems

- **Open single-server queue**
  - Arrival can occur at any queue length
  - Throughput, $X = \min(\lambda, 1/S)$
  - $X = \frac{1}{S} < \lambda \Rightarrow$ shared server is in overload
  - $X = \frac{\lambda}{\lambda} \Rightarrow$ throughput/arrival rate to queue is independent of $R$

Closed system

- **Closed system**
  - Arrival won’t occur when the queue contains $N$ customers
  - Queue arrival rate = queue departure rate $\leq \frac{1}{S}$
  - If the shared resource is upgraded (i.e., $S$ decreases)
    - Throughput decreases
    - Average queue length decreases
    - Throughput stays the same, since throughput $= \lambda$

Example:
- Web server with no mirror sites
- A disk in a file server
- A system semaphore

Open versus Closed - Summary

- **Open system**
  - Arrival can always occur
  - System can be overloaded
  - If throughput $= \lambda$, upgrading a system resource does not increase throughput
  - The behavior is qualitatively different; consider which is the most appropriate workload

- **Closed system**
  - Arrival won’t occur when the queue contains $N$ customers
  - Queue arrival rate = queue departure rate $\leq \frac{1}{S}$
  - If the shared resource is upgraded (i.e., $S$ decreases)
    - Throughput increases, since throughput $= \frac{1}{1/R}$

Open versus Closed - “Pop Quiz”

- **Open system**
  - Arrival can always occur
  - System can be overloaded
  - If throughput $= \lambda$, upgrading a system resource does not increase throughput
  - Closed system
  - No arrivals when $N$ in system
  - System overload infeasible
  - Throughput = $N(1/R - 1/R) = 1/R$

- **Closed system**
  - No arrivals when $N$ in system
  - System overload infeasible
  - Throughput = $N(1/R - 1/R) = 1/R$

Example:
1. Jobs arriving to a compute server?
2. A bottleneck link in the Internet?
Open versus Closed - Wrap-up

- Usually the appropriate decision is clear:
  - Shared semaphore: closed (fixed N)
  - Shared memory system resources: closed (fixed N)
  - Bottleneck link in the Internet: open
  - Web server: open (perhaps project a single arrival rate change)

- It's important to make a thoughtful decision:
  - Incorrect workload can lead to incorrect performance projections
  - If the simulation workload is a trace of request arrival times, check that the request completion rate is the request arrival rate (otherwise, requests are getting stuck in the system)

Output Measures: open systems

- $X = \lambda$: system throughput (customers/sec)
- $R$: average total system residence time (sec)
- $Q$: average number of customers in the system (summed over all the queues & delay centers)

Output Measures: single-server queue

- $X = \lambda$: throughput (customers/sec)
- $U$: server utilization (fraction of time server is busy)
- $R = w(S)$: average total queue residence time (sec)
- $Q$: average queue length (incl. the customer in service)

Output Measures: closed systems

- $X$: system throughput (at designated departure pt.)
- $R$: average system residence time (for designated collection of queues & delay centers)
- $Q$: average number in the "system" (summed over the designated queues & delay centers)

Per-queue Output Measures

- $U_k$: utilization of the server at node $k$
- $X_k$: throughput (or departure rate) at node $k$
- $Q_k$: average queue length at node $k$

Average Residence Time: $R_k$

- $R_k$: average total residence time at node $k$
- $R_k / V_k$: average residence time per visit to $k$
- $R_k / V_k$: average residence time per visit to $k$
Modeling Methodology

- Measure system & workload
- Collect traces or count events
- Compute measures ($\lambda$, $S_{c,k}$, $R_c$, etc)
- Model system (workload resource requirements)
- Compute performance from model
- Validate results for observed workload parameters
- Compute new results for system design mods & for wider range of workload parameters

Modeling Caveats

- Analytic models are an important system design tool but simulation is also important
- [LZGS]: methodology is a way of thinking, not a substitute for thinking
- Successful modeling requires more than methods & software packages; need to learn how to:
  - Define the model - what features to represent
  - Customize the model - non-standard solutions
  - Select ranges of model parameter values

Summary

Key Model Inputs:
- $\lambda$ = customer arrival rate to system
- $N$ = fixed number of customers in the system
- $(S_j)$; $(Z_j)$ = average service times & average delays
- $(p_{c,j})$ = routing probabilities
- $(V_i)$ = average number of visits to each center

Key Model Outputs:
- $X$ = system throughput (at departure pt.)
- $R$ = average system residence time
- $Q$ = average number of customers in the system

Per-class Measures

- $S_{c,k}$ = average class c service time at center k
- $U_{c,k}$ = utilization of node k by class c
e.g., $S_{X,cpu}$, $U_{X,cpu}$, $U_{X,mem}$

CS 547: next steps

- Read LZGS, Chapter 3 for Wednesday
- Homework #1 due Friday
- Keep up with reading & homeworks

Per-class Measures

- $X_{c,k}$ = throughput of class c at node k
- $Q_{c,k}$ = average class c queue length at node k
Per-class Measures

- $X_c$ = system throughput for class $c$
- $R_c$ = average system residence time
- $Q_c, N_c$ = average no. of class $c$ in the system

$$Q_A = Q_{A,cpu} + Q_{A,disk1} + Q_{A,disk2}$$

Per-class & Overall Measures

$$Q_{cpu} = Q_{A,cpu} + Q_{B,cpu}$$
$$Q = Q_A + Q_B = Q_{cpu} + Q_{disk1} + Q_{disk2}$$
$$X = X_A + X_B; \quad X_{cpu} = X_{cpu,A} + X_{cpu,B}$$

Questions?