Queueing Theory-1

R. Jain book + M. Claypool [WPI]

Introduction

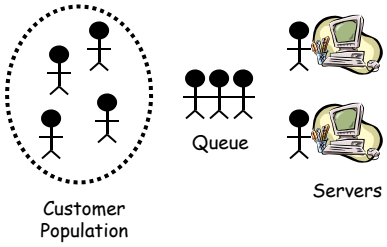
- In computers, jobs share many resources: CPU, disks, devices
- Only one can access at a time, and others must wait in queues
- Queuing theory helps determine time jobs spend in queue
  - Can help predict response time

Outline

- Introduction
- Notation and Rules
- Little's Law
- Utilization Law
- Types of Stochastic Processes
- Analysis of a Single Queue, Single Server
- Analysis of a Single Queue, Multiple Servers

3

Notation-1

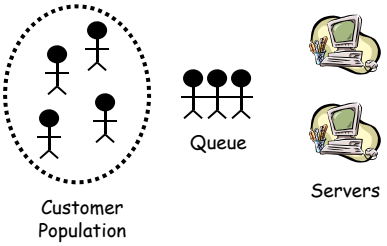


- Imagine waiting for a service in a Bank (or checking out at a grocery store, or …)
  - Resources are "servers"
  - People are "customers"
- If all servers busy, customers wait in a "queue"

- For queuing analysis, need to specify:
  - Population size
  - Number of servers
  - System capacity
  - Arrival process
  - Service time distribution
  - Service discipline

4

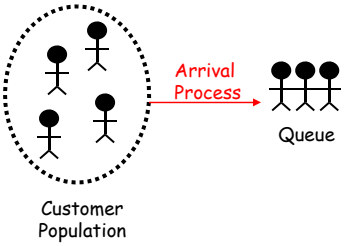
Notation-2



- Population size
  - Potential customers who can enter
  - Most real systems: size is finite but easier to analyze if infinite
- Number of servers
  - Can be one or more
  - Assume identical, but if not then separate queuing system for each
- System capacity
  - Number that can wait plus be served
  - Most systems have finite queue length, but easier to analyze if infinite

5

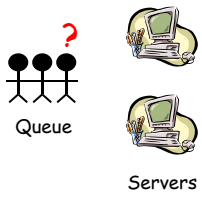
Notation-3



- Arrival process
  - Students arrive at $t_1, t_2, \ldots, t_j$
  - Interarrival times are $\tau_j = t_j - t_{j-1}$
  - Usually assume independent, identically distributed (IID)
- Arrival process (cont.)
  - Most common are Poisson arrivals
    - IID and exponentially distributed ($f(x)=\lambda e^{-\lambda x}$)
- Service time distribution
  - Amount of time each customer at server
  - Again, usually IID
  - Most common are exponential

6

Notation-4



- Service discipline
  - Order customers called for servicing
  - Most common is FCFS

- Kendall notation
  - A/S/m/B/K/SD
  - A is Arrival time distr.
  - S is Service time distr.
  - m is number of servers
  - B is number of buffers (queue size + servers)
  - K is population size
  - SD is service discipline
- Some typical times used:
  - M Exponential
    - M means “memoryless” in that current arrival independent of past
  - D Deterministic
  - G General
    - Valid for all

7

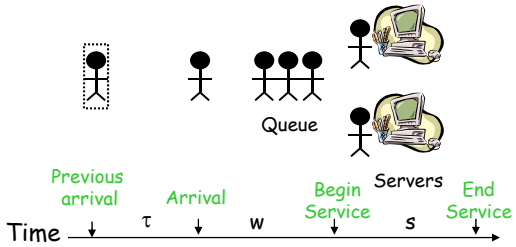
Notation Example

- M/M/3/20/1500/FCFS – single queue system with:
  - Exponentially distributed arrivals
  - Exponentially distributed service times
  - Three servers
  - Capacity 20 (queue size is 20 – 3 = 17)
  - Population is 1500 total
  - Service discipline is FCFS

- Often, assume infinite queue and infinite population and FCFS, so just → M/M/3

8

Variables for All Queues



- τ = interarrival time
- λ = mean arrival rate
  - = 1/E[τ]
  - Can sometimes depend upon jobs in system
- s = service time per job
- μ = mean service rate per server
  - = 1/E[s], total rate mμ
- n_q = number of jobs waiting in queue
- n_s = number of jobs receiving service
- n = number of jobs in system
  - $n = n_q + n_s$
- r = response time
- w = waiting time

Note, all except μ and λ are random

9

Rules for All Queues-1

- *Stability Condition*
  - If the number of jobs becomes infinite, system unstable. For stability, mean arrival rate should be less than mean service rate

    $$\lambda < m\mu$$

  - Does not apply to finite queue or finite population systems
    - Finite population cannot have infinite queue
    - Finite queue drops if too many arrive so never has infinite queue

10

Rules for All Queues-2

- *Number in System versus Number in Queue*
  - Number of jobs is equal to waiting and servicing

$$n = n_q + n_s$$

  - Also means:

$$E[n] = E[n_q]+E[n_s]$$

  - So mean number of jobs is equal to mean number in queue plus mean number being serviced

$$Var[n] = Var[n_q]+Var[n_s]$$

  - Variance of jobs equal to variance of queue + svc
- Also, service rate of servers independent of jobs in queue

$$Cov(n_q,n_s) = 0$$

11

Rules for All Queues-3

- *Number versus Time*
  - If jobs not lost due to buffer overflow the mean jobs is related to response time as:

mean jobs in system = arrival rate x mean response time

  - Similarly

mean jobs in queue = arrival rate x mean waiting time

  - Above equations known as “Little’s Law” (derivation: later)
  - For finite buffers, we can use effective arrival rate (ignoring drops)

12

Rules for All Queues-4

- *Time in System versus Time in Queue*
  - Time spent in system is sum of queue and service time

$$r = w + s$$

  - In particular:

$$E[r] = E[w] + E[s]$$

  - If service rate independent of jobs in queue

$$Cov(w,s) = 0$$

$$Var[r] = Var[w] + Var[s]$$

13

Outline

- Introduction
- Notation and Rules
- Little's Law
- Utilization Law
- Types of Stochastic Processes
- Analysis of a Single Queue, Single Server
- Analysis of a Single Queue, Multiple Servers

14

Little's Law

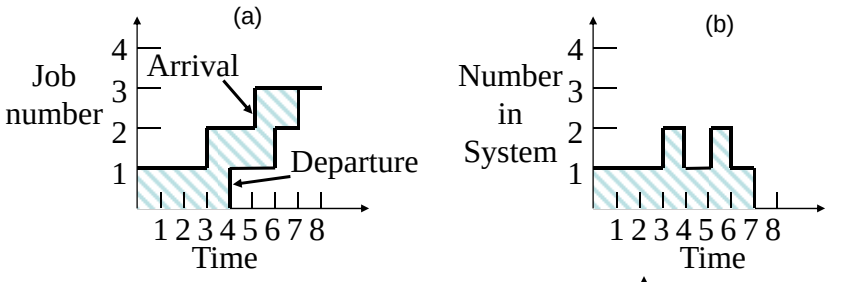
Mean jobs in system = arrival rate x mean response time

- Very commonly used in theorems
- Applies if jobs entering equals jobs serviced
  - No new jobs created, no new jobs lost
  - If lost, can adjust arrival rate to mean only those not lost
- Intuition: suppose monitor system and keep log of arrival and departures. If long enough, arrivals about the same as departures.
  - Let there be N arrivals in long time T. Then:

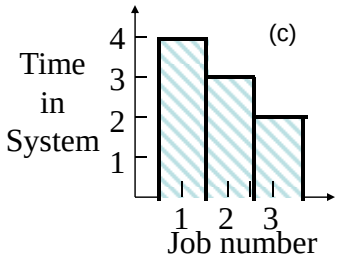
    arrival rate = total arrivals / total time = N/T

15

Proof of Little's Law

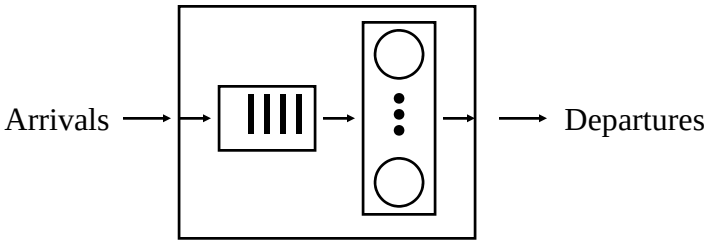


- If *T* is large, arrivals = departures = *N*
- Arrival rate = Total arrivals/Total time= *N/T*
- Hatched areas = total time spent inside the system by all jobs = *J*
- (c) $\Rightarrow$ Mean time in the system= *J/N*
- (b) $\Rightarrow$ Mean Number in the system
  = *J/T* = $\frac{N}{T} \times \frac{J}{N}$
  = Arrival rate£ Mean time in the system



16

Application of Little's Law



- Applying to just the waiting facility of a service center

  Mean number in the queue = Arrival rate × Mean waiting time
- Similarly, for those currently receiving the service, we have:

  Mean number in service = Arrival rate × Mean service time

17

Example

- A monitor on a disk server showed that the average time to satisfy an I/O request was 100 milliseconds. The I/O rate was about 100 requests per second. What was the mean number of requests at the disk server?

- Using Little's law:

  Mean number in the disk server

  = Arrival rate × Response time

  = 100 (requests/second) ×(0.1 seconds)

  = 10 requests

18

Outline

- Introduction
- Notation and Rules
- Little's Law
- Utilization Law
- Types of Stochastic Processes
- Analysis of a Single Queue, Single Server
- Analysis of a Single Queue, Multiple Servers

19

Utilization Law-1

- Given average arrival rate λ.
- Average utilization of a system is time busy over total time

$$U = b/T$$

- Factor into:

$$U = b/T = (b/d)\ (d/T)$$

  where d is number of departures and arrivals during time T
- Notice, (b/d) is average time spent servicing each of the d jobs. Call it s (s = b/d)
- Since balanced (in == out), $\lambda = d/T$
- So:

$$U = \lambda s \quad (\textit{Utilization Law})$$

20

Utilization Law-2

- Consider I/O system with one disk and one controller. If average time required to service each request is 6 msec, what is maximum request rate it can tolerate?
- Maximum will occur when 100% utilized, so U=1
- Substituting $U = \lambda s$, we get:

$$1 = \lambda_{max} s$$

- So, $\lambda_{max} = 1 / (6 \times 10^{-3}) = 167$ requests/sec

21

Utilization Law-3

- Notice, utilization law $U = \lambda s$ can be written as:

$$U = \lambda / \mu$$

  where μ is the average service rate
- Ratio λ/μ is often called *traffic intensity*
  - Given own symbol $\rho = \lambda/\mu$
- If ($\rho > 1$) then $\lambda > \mu$ (arrival rate greater than service rate)
  - Jobs arrive faster than can be processed
  - Queue grows to infinity
  - Unstable
- Must have ($\rho < 1$) for stability (so U never > 100%)

22

Operational Analysis

- Using Little's Law and Utilization Law can say things about average behavior
  - Requires no assumptions about distribution times of arrivals or servicing
  - High level view
- But can not say things about, say, maximum or worst case
- For example, cannot use it to determine needed buffer space to enqueue incoming requests
- Will use stochastic distributions and queuing theory to get more detailed analysis

23

Outline

- Introduction
- Notation and Rules
- Little's Law
- Utilization Law
- Types of Stochastic Processes
- Analysis of a Single Queue, Single Server
- Analysis of a Single Queue, Multiple Servers

24

Stochastic Processes

- **Process**: Function of time
- **Stochastic Process**: Random variables, which are functions of time
- ***Example 1:***
  - *n(t)* = number of jobs at the CPU of a computer system
  - Take several identical systems and observe n(t)
  - The number *n(t)* is a random variable.
  - Can find the probability distribution functions for *n(t)* at each possible value of t.
- ***Example 2:***
  - *w(t)* = waiting time in a queue

25

Types of Stochastic Processes

- Discrete or Continuous State Processes
- Markov Processes
- Birth-death Processes
- Poisson Processes

26

Discrete/Continuous State Processes

- Discrete = Finite or Countable
- Number of jobs in a system *n(t) = 0, 1, 2, ...*
- *n(t)* is a discrete state process
- The waiting time *w(t)* is a continuous state process.
- **Stochastic Chain**: discrete state stochastic process

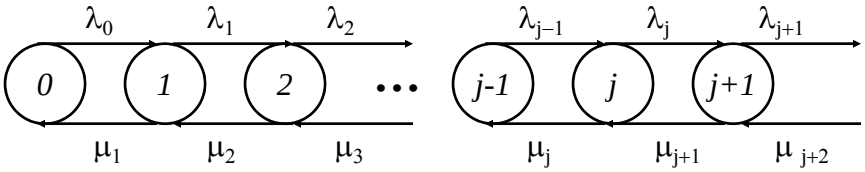
27

Markov Processes

- Future states are independent of the past and depend only on the present.
- Named after A. A. Markov who defined and analyzed them in 1907.
- **Markov Chain**: discrete state Markov process
- Markov ⇒ It is not necessary to know how long the process has been in the current state ⇒ State time has a memoryless (exponential) distribution
- *M/M/m* queues can be modeled using Markov processes.
- The time spent by a job in such a queue is a Markov process and the number of jobs in the queue is a Markov chain.

28

Birth-Death Processes

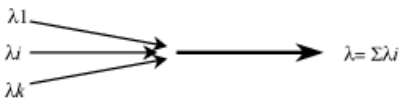


- The discrete space Markov processes in which the transitions are restricted to neighboring states
- Process in state *n* can change only to state *n+1* or *n-1*.
- Example: the number of jobs in a queue with a single server and individual arrivals (not bulk arrivals)

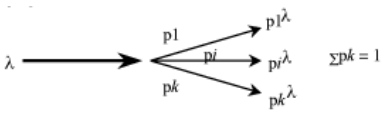
29

Poisson Processes-1

- Interarrival time *s* = IID and exponential
  ⇒ number of arrivals *n* over a given interval *(t, t+x)* has a Poisson distribution
  ⇒ arrival = Poisson process or Poisson stream
- Properties:
  - 1.Merging: $\lambda = \sum_{i=1}^{k} \lambda_i$



  - 2.Splitting: If the probability of a job going to *ith* substream is p_i, each substream is also Poisson with a mean rate of $p_i \lambda$



30

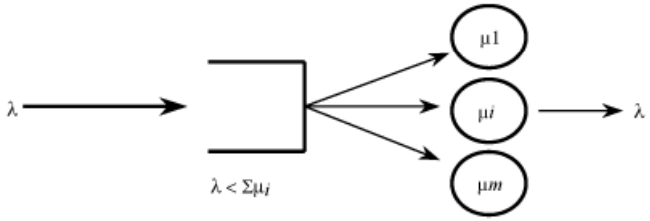
Poisson Processes-2

– 3.If the arrivals to a single server with exponential service time are Poisson with mean rate λ, the departures are also Poisson with the same rate λ provided $\lambda < \mu$.

31

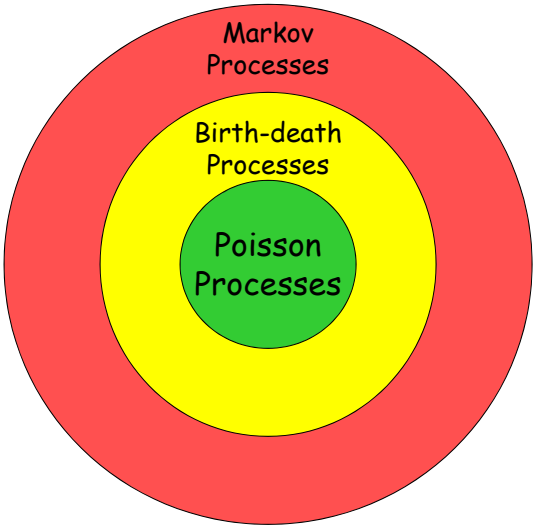
Poisson Processes-3

– 4. If the arrivals to a service facility with m service centers are Poisson with a mean rate λ, the departures also constitute a Poisson stream with the same rate λ, provided $\lambda < \sum_i \mu_i$. Here, the servers are assumed to have exponentially distributed service times.



32

Types of Stochastic Processes



33

Example

- During a one-hour observation interval, the name server of a distributed system received *10,800* requests. The mean response time of these requests was observed to be one-third of a second. What is the mean number of queries in the server?

- Answer:

Throughput X = 10800/3600 = 3 requests/sec

Q = XR = 3× (1/3) = 1 request

34