UTILIZATION LAW & LITTLE’S LAW

Software Performance Engineering

Our analysis thus far

- We define metrics for each system to measure performance
- We use the exponential distribution
  - To analyze inter-arrival times in a Markovian stochastic system
  - For a single random variable, e.g. “customers arriving”
- …but real systems have multiple random variables interacting!
  - Servers behave randomly
  - Customers behave randomly

Arrival Rate & Service Rate

- Arrival rate: λ
  - The rate at which jobs (“customers”) are entering the system
  - Can be constant, or a parameter to exponential distribution
  - e.g. “We get 60 customers per hour”
- Mean inter-arrival time: $1/\lambda$
  - e.g. “We get a new customer every minute”
- Service rate: μ
  - The rate at which our server handles jobs
  - Can be constant, or a parameter to exponential distribution
  - e.g. “We can serve 120 customers per hour”
- Mean service time: $1/\mu$ or S
  - e.g. “We take 2 minutes for every customer”

Traffic Intensity & Utilization

- Traffic intensity ρ
  - $\rho = {}^{\lambda}/_{\mu} = \lambda S$
  - % measure of load on the overall system
  - e.g. “Traffic intensity of 50%”
  - If $\rho \geq 1$, the system cannot keep up with demand

    (a lot of our math breaks down in this situation)
- Utilization U
  - The proportion of time the server is busy
  - Maxes out at 100%, or *saturated*
  - If jobs are never lost, $U = \rho$
  - If jobs can be lost, $U \leq \rho$

Throughput & Queue Length

- Throughput: X
  - The rate at which the entire system processes jobs
  - (Becomes more complex in multi-service systems)
  - If jobs can be lost, $X \leq \lambda$
- Queue Length: n
  - The number of jobs present in the system
  - The number of jobs present at a given server

Response Time & Waiting Time

- Response time: $\bar{R}$
  - The total period of time from a job's arrival to its final processing
- Waiting time
  - The time that a job must wait in the queue until it is served

Queuing Discipline

- FCFS: First Come First Serve
- LCFS: Last Come First Serve
  - e.g. a stack
- LCFSPR: Last Come First Served Preemptive Resume
  - The most recently arriving job preempts the job
  - That job is served to completion, unless preempted itself
- Time Slicing or Round Robin
  - Each job is given a fixed period of time before it is interrupted and switches to another job in the queue

Utilization Law

- These laws are helpful for:
  - Determining if your measurements are sane
  - Examining how each your metrics relate to each other

- Utilization Law: $U = XS = {}^{X}/_{\mu}$
  - Utilization is the product of throughput and mean service time
  - This is true *regardless* of your queuing discipline
  - e.g. Coffee Shop
    S = 2 m/c, “e.g. “We take 2 minutes for every customer”
    X = 1/4 c/m, “We get 1 customers every 4 minutes”
    U = 50%

More Utilization Law eg’s

- e.g. Processors
  - Mean service time for a job is 10ms
  - What is the maximum expected throughput if we want our maximum utilization at 80%?
  - X = U/S = 0.8/(10 ms/j)= .08 j/ms = 8 jobs/sec

Little's Law

- Mean Queue length $\bar{n}$ is the product of Throughput X and Mean Response Time R
- $\bar{n} = X\bar{R}$
- e.g. Rollercoaster
  - On average it takes 15 minutes from getting to the back of the line to riding to exit.
  - The rollercoaster handles 20 riders/hour (1 rider/3 minutes)
  - Thus, the mean queue length is: $\bar{n} = X\bar{R} = \frac{15}{3} = 5$ riders
- e.g. Rollercoaster – should we ride?
  - The rollercoaster handles 20 riders/hour (1 rider/3 minutes)
  - Queue length is currently 30
  - Current line is 30 people. (Assume that is average)
  - How long will we wait?
  - $\bar{R} = \frac{n}{X} = \frac{30}{0.333} = 90$ minutes

Applied to Single Server Systems

- M/M/1 queues
  - Customer arrival rate is Markovian
  - Service Rate is Markovian
  - 1 Server
  - No maximum capacity, infinite customers
- $X = \lambda \quad \text{if } \rho \leq 1$
  - With a single server, throughput of the entire system is basically just customer arrival rate
- $\bar{R} = \frac{\bar{n}}{\lambda}$
  - Application of Little's law with the above assumption

Queue Length of M/M/1

- $\bar{n} = \frac{\rho}{1-\rho}$ eq 3.8 from the book, we'll skip the derivation
- e.g. Web app
  - Customers arrive 1 per second, λ=1 c/s
  - Webapp processes them at 8 per second, $\mu = 8$ c/s
  - Traffic intensity: $\rho = \frac{1}{8} = 12.5\%$
  - Mean queue length: $\bar{n} = \frac{0.125}{0.875} = 0.142$
- e.g. Busy Web app
  - Customers arrive 6 per second, λ=6 c/s
  - Webapp processes them at 8 per second, $\mu = 8$ c/s
  - Traffic intensity: $\rho = \frac{6}{8} = 75\%$
  - Mean queue length: $\bar{n} = \frac{0.75}{0.25} = 3$
  - So our λ increased by 6x, but our $\bar{n}$ increased by 21x!!

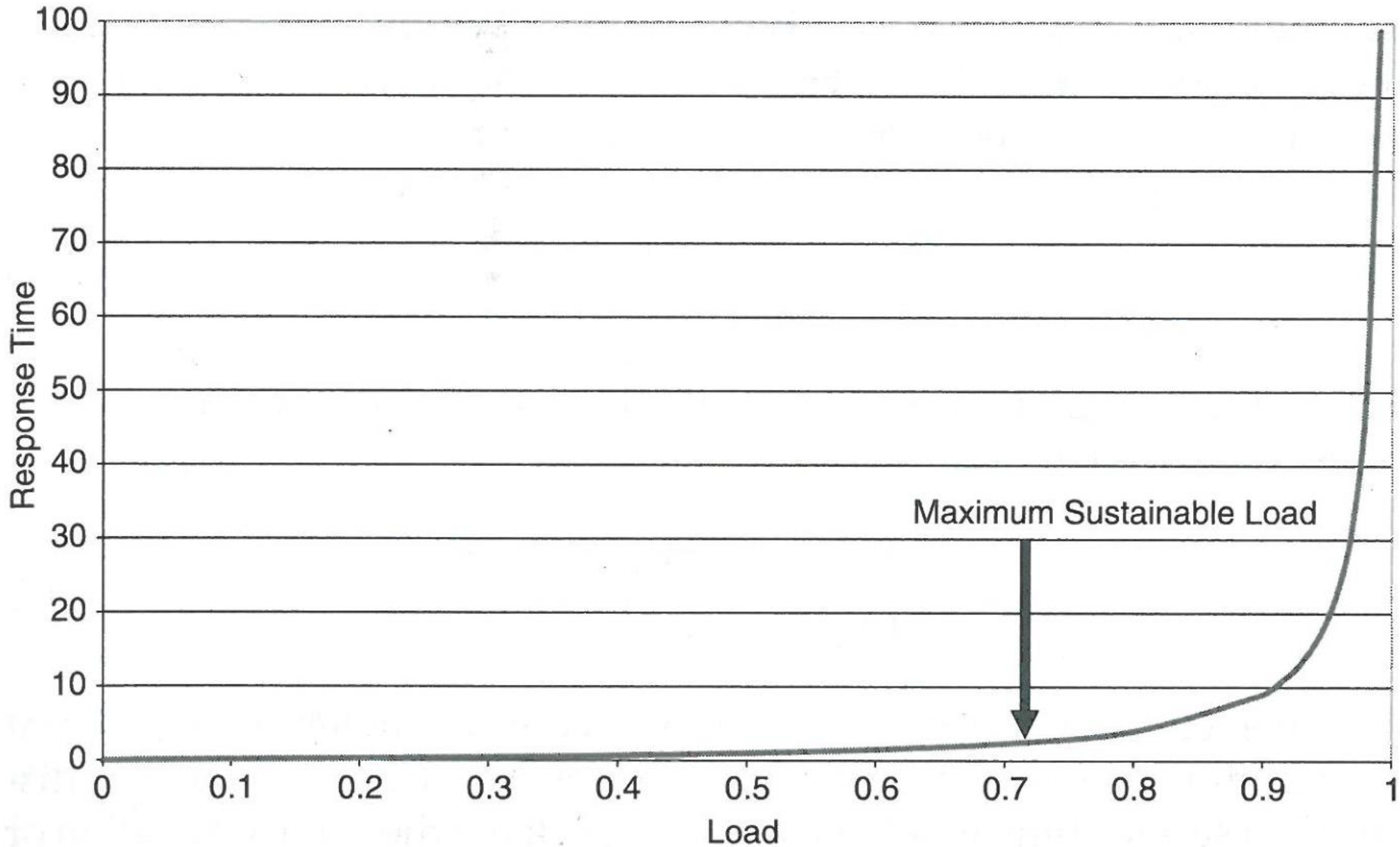

Figure 3.8 *Mean response time of an M/M/1 queue*

- For this chart, assume that λ=1 so that $\lambda = \bar{R}$
- Load is ρ, or U if jobs are never lost