Client-Server Systems

Performance Modeling

Engineering Cloud Computing

Week 4

Client-Server Architecture

- **Clients:** Remote programs and users
- Servers can support multiple clients
- Servers implement one or more services
- Centralized services implemented on single servers
- Distributed: service implemented across multiple servers

Clients → Req → Server

Response

- Cloud: Pay-per-use
- Underprovisioned: traffic > capacity to handle
- Overprovisioned: More servers than needed to meet some performance requirement

Response time

GET stud_id.lastseen

Canvas

student list

Lastseen

Memcached-file

User → Load Balancer → HTTP Server-1 → Storage Server

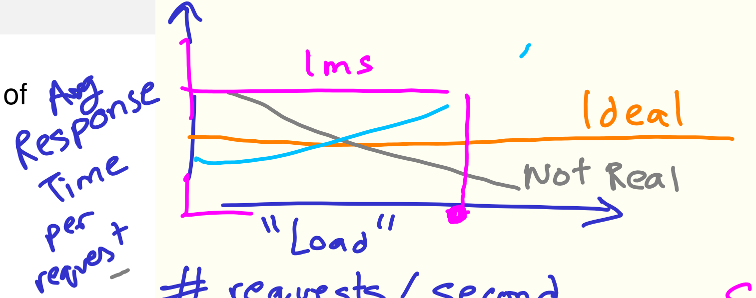
HTTP Server-2

⋮

2 tiers

Server Capacity

- Servers can become a bottleneck with increasing number of requests
  - Computational capacity, limited by the CPUs
  - Storage capacity: I/O transfer rate
  - Network between user and server



\# requests / second

Service Level Agreement (SLA)

Resp time < 1 ms

Computation Inside Servers

Network Interface Card

- Packet processing
  - Data from NIC to CPU via DMA
  - Interrupt handling
  - Packet travels up the network stack
  - Processes blocked on socket woken up
- Application level processing
  - Parse data from socket
  - Process/store data

Transp
Net
Link
Phy

Server performance

Key Metrics

Response time: Time between request initiation and response

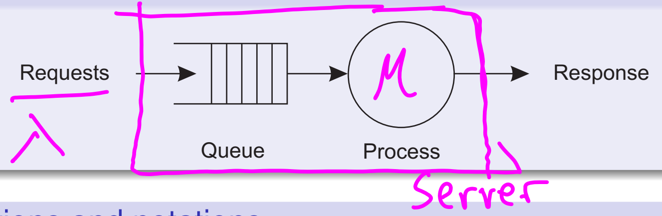
Throughput: Number of requests handled by server (per second)

Server Performance Considerations

- How many concurrent clients can be served?
- What is the maximum throughput that can be sustained?
- What is the response times for clients?

Queuing Theory Model

A centralized service can be modeled as a simple queuing system



Assumptions and notations

- The queue has infinite capacity $\Rightarrow$ arrival rate of requests is not influenced by current queue length or what is being processed.
- Arrival rate of requests: λ
- Processing capacity service: μ requests per second

① OS pkt process +
② Parse the req +
③ Retrieve / Store key / value pair

Quick Quiz

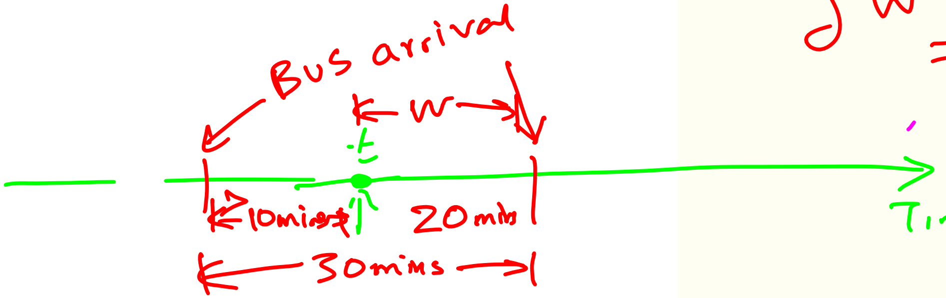
Bob finds his friend, Alice, at the bus-stop. It turns out both Alice and Bob are waiting for the same bus.
Alice has been waiting for the bus for 10 minutes. The bus is scheduled to arrive every 30 minutes. — arrival rate
Assume that there is no other information available about the bus (no real-time GPS etc.). What is Bob's expected waiting time for the bus?

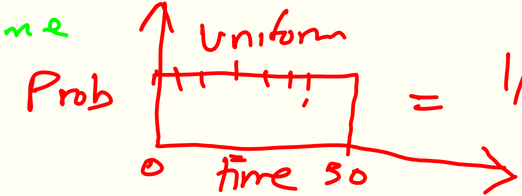
$P(W)$

$E[W] = \int_0^\infty w P(w)\, dw$

$W \leq 30$ mins

$$\frac{\int w \, P(W \mid \text{no bus for 10 mins})}{\int w P(w)}$$





$= 1/30$

Distribution of Requests and Service Times

- Requests arrive according to a random process
- Typically, arrival process is modeled as a **Poisson distribution**
- Arrival rate: λ per second
- Request service rate: μ per second

$$P(\text{n arrivals in interval T}) = \frac{(\lambda T)^n e^{-\lambda T}}{n!}$$

$$E[n] = \lambda T$$

Inter-arrival time: Time between successive events

CDF

$$
\begin{aligned}
P(IA \le t) &= 1 - P(IA > t) && (1)\\
&= 1 - P(\text{0 arrivals in time t}) && (2)\\
&= 1 - e^{-\lambda t} && (3)\\
& && (4)
\end{aligned}
$$

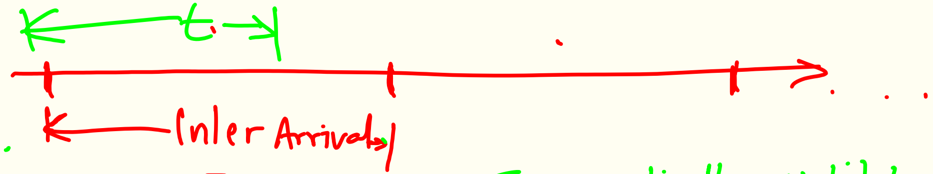
$1 - e^{-\lambda t}$ is the CDF of the **exponential distribution**!

Service Time: Exponentially distributed with parameter μ → $CDF = 1 - e^{-\mu t}$

$\lambda = 2$ req/second

$$\underset{= n}{P(5}\text{ reqs in } \underbrace{10 \text{ seconds}}_{T}) = \frac{(2 \cdot 10)^5 e^{-2 \cdot 10}}{5!}$$

$E[\text{Reqs in 1 minute}] = 2 \times 60 = 120$



Time → Exponentially distribution

Exponential Distribution

- CDF: $F(t) = 1 - e^{-\lambda t}$
- Probability distribution: $f(t) = \lambda e^{-\lambda t}$
- **Memoryless**: $P(X \leq T + a | X > a) = P(X \leq T)$

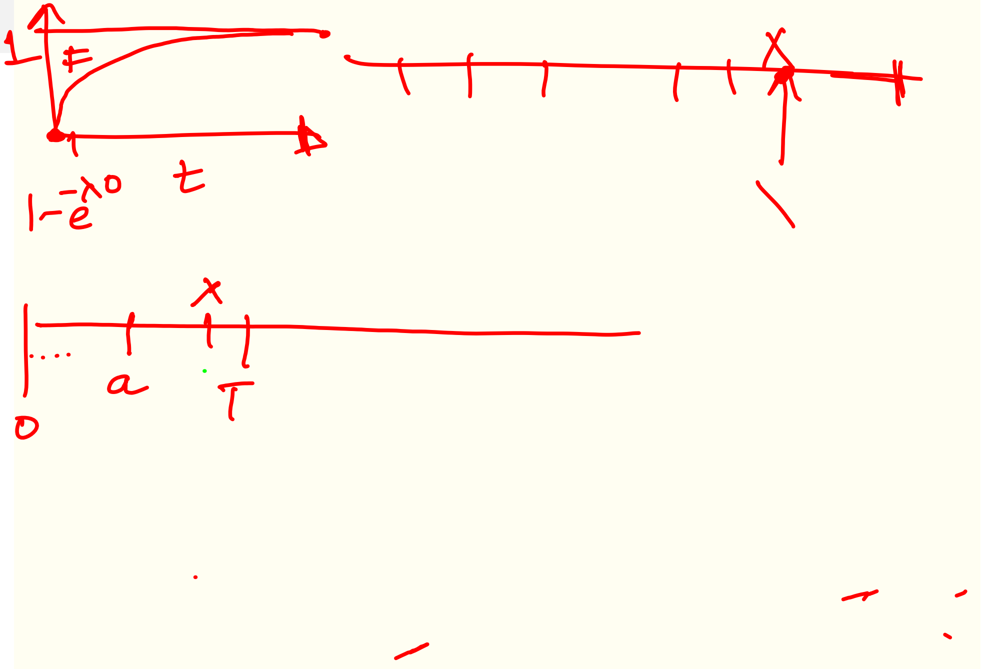
Proof:

$$P(X \leq T + a | X > a) = \frac{P(a \leq X \leq T + a)}{P(X > a)} \tag{5}$$

$$= \frac{\int_a^{T+a} \lambda e^{-\lambda t} dt}{\int_a^{\infty} \lambda e^{-\lambda t} dt} \tag{6}$$

$$= 1 - e^{-\lambda T} = P(X \leq T) \tag{7}$$

- Previous history does not help in predicting future events
- Waiting for the bus example: Waiting time of others doesn't matter if bus arrivals are exponentially distributed.



Markov Chains

- States with transition probabilities P_{ij} between state i and j
- Represented by a matrix $\mathbf{P}$
- P_{ij} is probability of going from j to i in 1 step
- $(\mathbf{P}^2)_{ij}$ denotes probability of being in j after starting at i after 2 steps.
- We are interested in P^n for $n \to \infty$
- For markov chains, P^n_{ij} is in a row and column
- That is, the limiting probability of being in a state doesn't depend on where you start.
- Limiting distribution of being in state j: $\pi_j = lim_{n \to \infty} P^n_{ij}$
- $\sum \pi_i = 1$

M/M/1 Queue Markov Chains

- Balance equations:
- $\lambda\pi_0 = \mu\pi_1$
- $(\lambda + \mu)\pi_1 = \lambda\pi_0 + \mu\pi_2$
- $(\lambda + \mu)\pi_n = \lambda\pi_{n-1} + \mu\pi_{n+1}$

$\pi_1 = \frac{\lambda}{\mu}\pi_0$
$\pi_n = \left(\frac{\lambda}{\mu}\right)^n \pi_0$
Let $\rho = \lambda/\mu$ and $\rho < 1$

$$1 = \sum_{i=0}^{\infty} \pi_i = \pi_0 \sum_{0}^{\infty} \rho^i = \frac{\pi_0}{1-\rho} \tag{8}$$

Last step is using the geometric series sum for ρ
$\pi_0 = 1 - \rho$

Properties

Mean number of objects in the system = $\rho/(1-\rho)$
Utilization = ρ

Fraction of time having k requests in the system

$$p_k = (1 - \frac{\lambda}{\mu})(\frac{\lambda}{\mu})^k$$

Little's Law

- N: Number of items in the system
- S: Response time (Time to leave the system)

Little's Law

$E[N] = \lambda E[T]$

Proof outline

1. Plot N vs. time, for a total time T
2. Area of the 'ribbon', A = Time spent by all items
3. $\lambda = N/T$
4. Mean time spent in the system, $E[T] = A/N$
5. Mean number in the system, $E[N] = A/T$
6. Counting the area in two ways = $E[N] = \lambda E[T]$

Little's Law Applications

- Very general. Multiple queueing disciplines, network of queues, etc!
- Average response time of server, $E[S] = E[T] = E[N]/\lambda$
- $E[N] = \sum_0^\infty kP_k = \rho/1 - \rho$
- $E[S] = \dfrac{1}{\mu - \lambda}$

Scenarios

1. Is it better to have one queue with 2 servers or 2 separate queues?
2. What happens when processing power is doubled?

Server Performance Implications

- Average Response Time = $1/\mu - \lambda$
- Useful to identify saturating load
- What to do if incoming traffic rate (λ) is close to μ?
  - Scaling techniques. Next class!

Throughput

- X: Rate at which events are processed
- C events processed in total time T
- $X = \frac{C}{T}$
- Events are only processed if system is busy
- Rate at which events are processed when system is busy = μ
- $X = \rho \cdot \mu = \lambda$
- Independent of μ !
- Throughput of server doesn't improve if its performance improves!?!
  Why do we want faster servers?

Closed-loop Systems

- Looked at **open** systems so far
- Many systems are closed, or atleast have some feedback
- Processed items feed back into the queue
- Web server example: people view a sequence of web-pages, based on what is served

Queue Networks

- Can represent system as a network of queues
- One queue for CPU, one for disk, etc.
- Or for different parts of the application
- Little's law is applicable!

Closed Networks

- Items feed back into queue after some “thinking time” $E[Z]$
- Total number of items = N
- $E[R]$ is the response time
- Little’s law: $N = XE[T]$
- But $E[T] = E[R] + E[Z]$
- Throughput $X \leq \dfrac{N}{E[R] + E[Z]}$
- For small N, the equality holds
- In practice, throughput converges to $1/E[R]$ for high N. (system is saturated)
- Closed systems useful for measuring the service rate μ