CS 3.307

Performance Modeling for Computer Systems

Tejas Bodas

Assistant Professor, IIIT Hyderabad

Logistics

Logistics

- Feel free to contact me anytime at tejas.bodas@iiit.ac.in.
- Office @ A5304.

Logistics

- Feel free to contact me anytime at tejas.bodas@iiit.ac.in.
- Office @ A5304.
- Book– Performance modeling and design of computer systems (Cambridge press) by Mor Harchol-Balter (Professor, CMU)
- Other books: 1) Stochastic processes by Sheldon Ross 2) Probabilistic modeling by Isi Mitrani.

Logistics

- Feel free to contact me anytime at tejas.bodas@iiit.ac.in.
- Office @ A5304.
- Book– Performance modeling and design of computer systems (Cambridge press) by Mor Harchol-Balter (Professor, CMU)
- Other books: 1) Stochastic processes by Sheldon Ross 2) Probabilistic modeling by Isi Mitrani.
- Assignment 1 : 15%. Midsem exam: 30%. Assignment 2: 15% Endsem 40 %.

Course Outline

- Module 1 (2 lectures)
  - Motivation, Probability refresher, Introduction to Stochastic Processes

Course Outline

- Module 1 (2 lectures)
  - Motivation, Probability refresher, Introduction to Stochastic Processes

- Module 2 (4 lectures)
  Poisson Process & Markov Chains

Course Outline

- Module 1 (2 lectures)
  - Motivation, Probability refresher, Introduction to Stochastic Processes
- Module 2 (4 lectures)
  Poisson Process & Markov Chains
- Module 3 (2 lectures) Elementary Queues

Course Outline

- Module 1 (2 lectures)
  - Motivation, Probability refresher, Introduction to Stochastic Processes
- Module 2 (4 lectures)
  Poisson Process & Markov Chains
- Module 3 (2 lectures) Elementary Queues
- Module 4 Renewal theorems and Busy period analysis (3 lectures)

Course Outline

- Module 1 (2 lectures)
  - Motivation, Probability refresher, Introduction to Stochastic Processes
- Module 2 (4 lectures)
  Poisson Process & Markov Chains
- Module 3 (2 lectures) Elementary Queues
- Module 4 Renewal theorems and Busy period analysis (3 lectures)
- Module 5 (3 lectures) Advanced Queues

Performance modeling for Computer systems

Performance modeling for Computer systems

- How do you measure the performance of your computer?

Performance modeling for Computer systems

- How do you measure the performance of your computer?

- Speed with which it runs programs.

Performance modeling for Computer systems

- How do you measure the performance of your computer?

- Speed with which it runs programs. RAM, clock speed,

Performance modeling for Computer systems

- How do you measure the performance of your computer?

- Speed with which it runs programs. RAM, clock speed, GPU, Cores.

Performance modeling for Computer systems

- How do you measure the performance of your computer?
- Speed with which it runs programs. RAM, clock speed, GPU, Cores.
- Storage space ?

Performance modeling for Computer systems

- How do you measure the performance of your computer?
- Speed with which it runs programs. RAM, clock speed, GPU, Cores.
- Storage space ? SSD or not ?

Performance modeling for Computer systems

- How do you measure the performance of your computer?
- Speed with which it runs programs. RAM, clock speed, GPU, Cores.
- Storage space ? SSD or not ?
- What is the key word here ?

Performance modeling for Computer systems

- How do you measure the performance of your computer?
- Speed with which it runs programs. RAM, clock speed, GPU, Cores.
- Storage space ? SSD or not ?
- What is the key word here ? LATENCY!

Performance modeling for Computer systems

- How do you measure the performance of your computer?
- Speed with which it runs programs. RAM, clock speed, GPU, Cores.
- Storage space ? SSD or not ?
- What is the key word here ? LATENCY!
- Performance metrics?

Performance modeling for Computer systems

- How do you measure the performance of your computer?
- Speed with which it runs programs. RAM, clock speed, GPU, Cores.
- Storage space ? SSD or not ?
- What is the key word here ? LATENCY!
- Performance metrics?
    - response time (run time, lag, delay, jitter)

Performance modeling for Computer systems

- How do you measure the performance of your computer?
- Speed with which it runs programs. RAM, clock speed, GPU, Cores.
- Storage space ? SSD or not ?
- What is the key word here ? LATENCY!
- Performance metrics?
  - response time (run time, lag, delay, jitter)
  - blocking probability (screen freeze, no disk space, packet loss, buffer full)

Modeling ?

Modeling ?

- Design for performance:

Modeling ?

- Design for performance: How many cores or GPU's?

Modeling ?

- Design for performance: How many cores or GPU's? which core to use?

Modeling ?

- Design for performance: How many cores or GPU's? which core to use? how to schedule instructions in a core?

Modeling ?

- Design for performance: How many cores or GPU's? which core to use? how to schedule instructions in a core?

- Routing (which core) and scheduling (which program/ instruction to execute)

Modeling ?

- Design for performance: How many cores or GPU's? which core to use? how to schedule instructions in a core?
- Routing (which core) and scheduling (which program/ instruction to execute)
- How do you know which is a good design?

Modeling ?

- Design for performance: How many cores or GPU's? which core to use? how to schedule instructions in a core?
- Routing (which core) and scheduling (which program/ instruction to execute)
- How do you know which is a good design? via experimentation?

Modeling ?

- Design for performance: How many cores or GPU's? which core to use? how to schedule instructions in a core?

- Routing (which core) and scheduling (which program/ instruction to execute)

- How do you know which is a good design? via experimentation?(costly!)

Modeling ?

- Design for performance: How many cores or GPU's? which core to use? how to schedule instructions in a core?
- Routing (which core) and scheduling (which program/ instruction to execute)
- How do you know which is a good design? via experimentation?(costly!)
- Performance analysis!

Modeling ?

- Design for performance: How many cores or GPU's? which core to use? how to schedule instructions in a core?
- Routing (which core) and scheduling (which program/ instruction to execute)
- How do you know which is a good design? via experimentation?(costly!)
- Performance analysis! via stochastic modeling

Applications Beyond Computers

Applications Beyond Computers

- Computer systems

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?
  - Scheduling operations, stocking of medicines, scheduling tests.

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?
  - Scheduling operations, stocking of medicines, scheduling tests.
- Hospitality industry

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?
  - Scheduling operations, stocking of medicines, scheduling tests.
- Hospitality industry
  - Designing hotel lobbies for faster checkin

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?
  - Scheduling operations, stocking of medicines, scheduling tests.
- Hospitality industry
  - Designing hotel lobbies for faster checkin
  - Restaurant seating!

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?
  - Scheduling operations, stocking of medicines, scheduling tests.
- Hospitality industry
  - Designing hotel lobbies for faster checkin
  - Restaurant seating! (How many tables of size 2,4,8?)

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?
  - Scheduling operations, stocking of medicines, scheduling tests.
- Hospitality industry
  - Designing hotel lobbies for faster checkin
  - Restaurant seating! (How many tables of size 2,4,8?)
- Transportation systems

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?
  - Scheduling operations, stocking of medicines, scheduling tests.
- Hospitality industry
  - Designing hotel lobbies for faster checkin
  - Restaurant seating! (How many tables of size 2,4,8?)
- Transportation systems
  - Airline or Railway scheduling

Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?
  - Scheduling operations, stocking of medicines, scheduling tests.
- Hospitality industry
  - Designing hotel lobbies for faster checkin
  - Restaurant seating! (How many tables of size 2,4,8?)
- Transportation systems
  - Airline or Railway scheduling
  - Priority scheduling, class differentiation

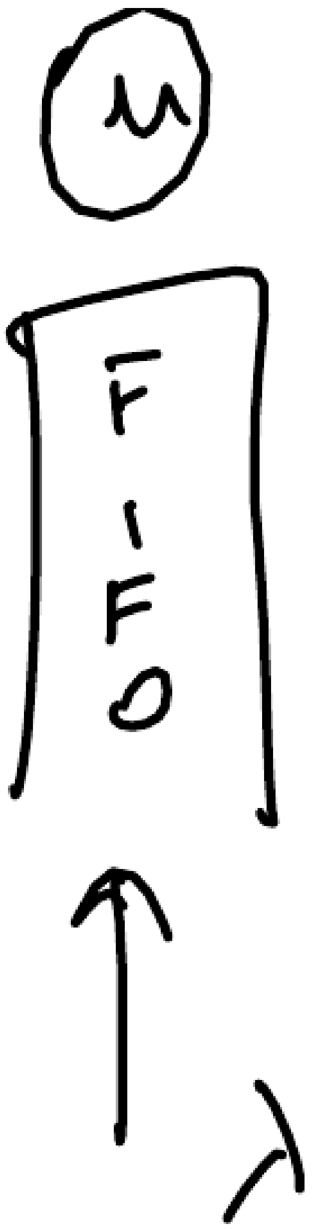
Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?
  - Scheduling operations, stocking of medicines, scheduling tests.
- Hospitality industry
  - Designing hotel lobbies for faster checkin
  - Restaurant seating! (How many tables of size 2,4,8?)
- Transportation systems
  - Airline or Railway scheduling
  - Priority scheduling, class differentiation
- Operation Research!

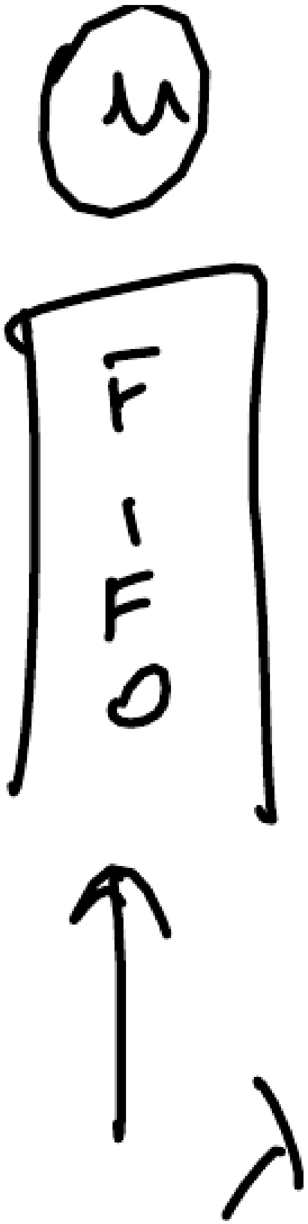
Applications Beyond Computers

- Computer systems
  - server farms, cloud computing, distributed storage systems
  - Communication systems, Wifi, Sensor networks.
- Heathcare
  - How many OT? How many Specialists or nurses?
  - Scheduling operations, stocking of medicines, scheduling tests.
- Hospitality industry
  - Designing hotel lobbies for faster checkin
  - Restaurant seating! (How many tables of size 2,4,8?)
- Transportation systems
  - Airline or Railway scheduling
  - Priority scheduling, class differentiation
- Operation Research!
- Henceforth use the term Queueing system!

A single server queue

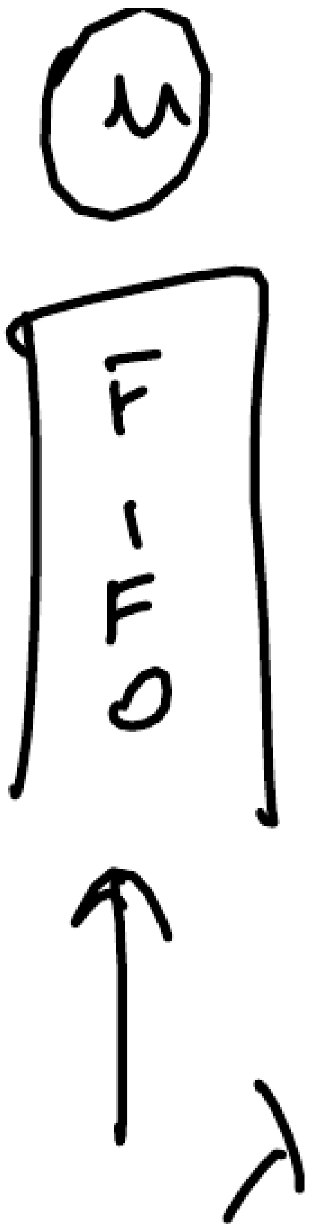


A single server queue



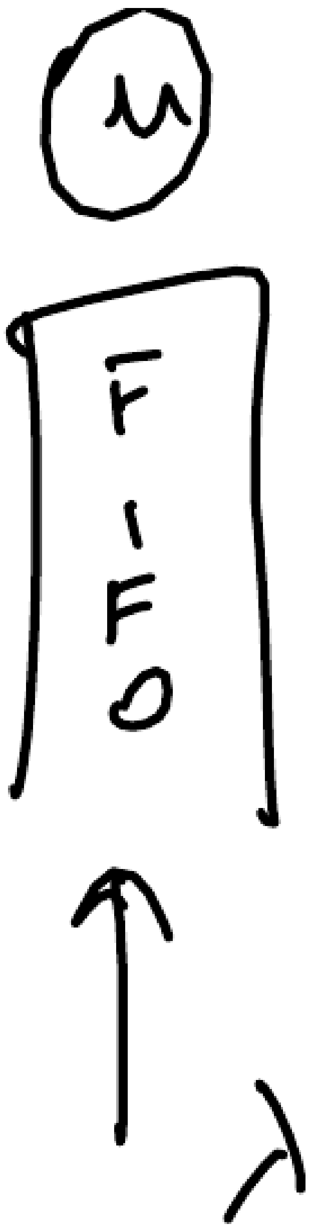
- One server, one FIFO queue for jobs to wait.

A single server queue



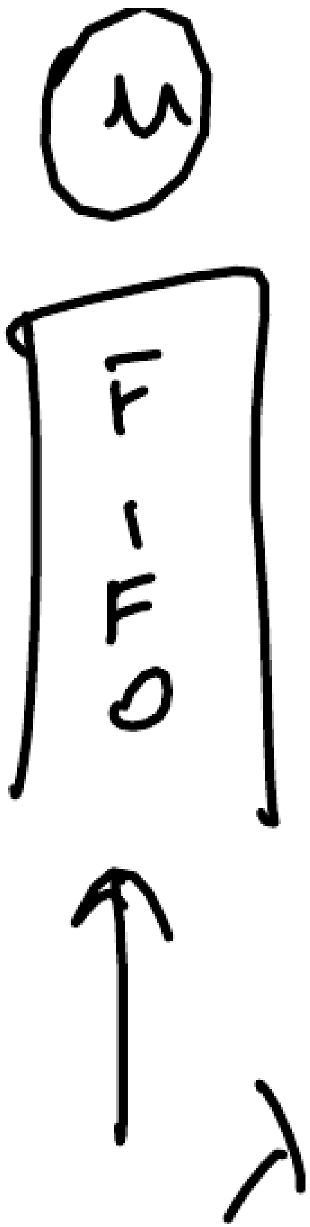
- One server, one FIFO queue for jobs to wait.
- μ denotes service rate, λ denotes the arrival rate.

A single server queue



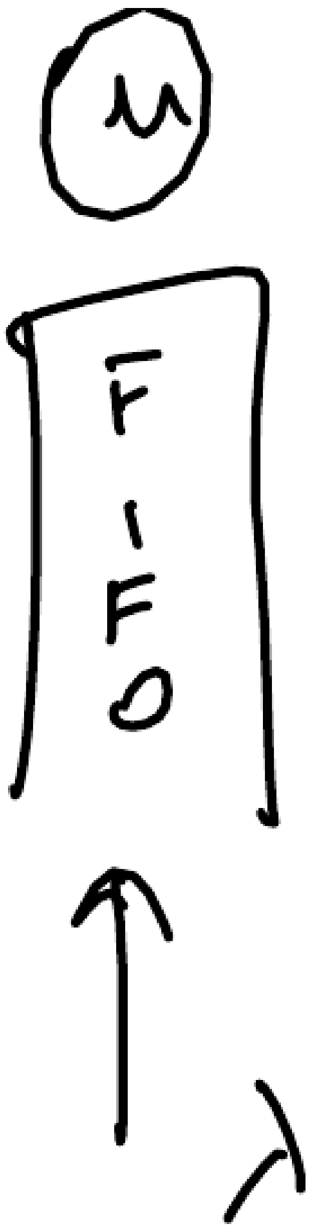
- One server, one FIFO queue for jobs to wait.
- μ denotes service rate, λ denotes the arrival rate.
- Service requirements S_n and inter-arrival times A_n are typically assumed to be i.i.d.

A single server queue



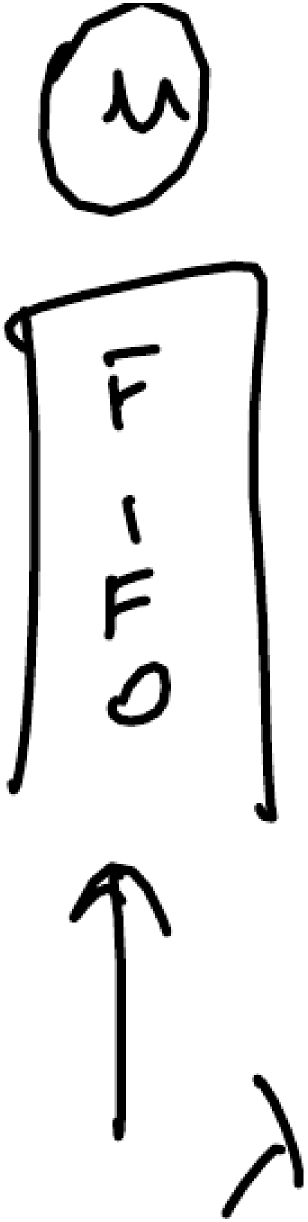
- One server, one FIFO queue for jobs to wait.
- μ denotes service rate, λ denotes the arrival rate.
- Service requirements S_n and inter-arrival times A_n are typically assumed to be i.i.d.
- In its simplest form, we will assume $S_n \sim Exp(\mu)$ and $A_n \sim Exp(\lambda)$.

A single server queue



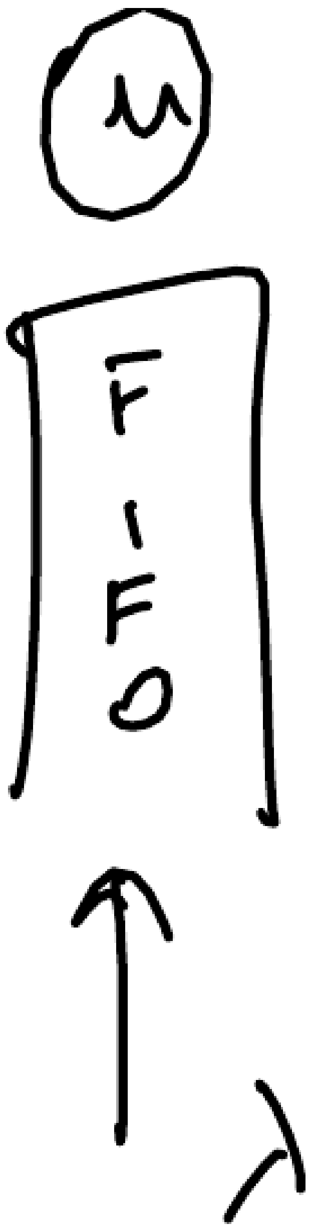
- One server, one FIFO queue for jobs to wait.
- μ denotes service rate, λ denotes the arrival rate.
- Service requirements S_n and inter-arrival times A_n are typically assumed to be i.i.d.
- In its simplest form, we will assume $S_n \sim Exp(\mu)$ and $A_n \sim Exp(\lambda)$.
- Jobs face queueing delay due to waiting for other jobs.

A single server queue



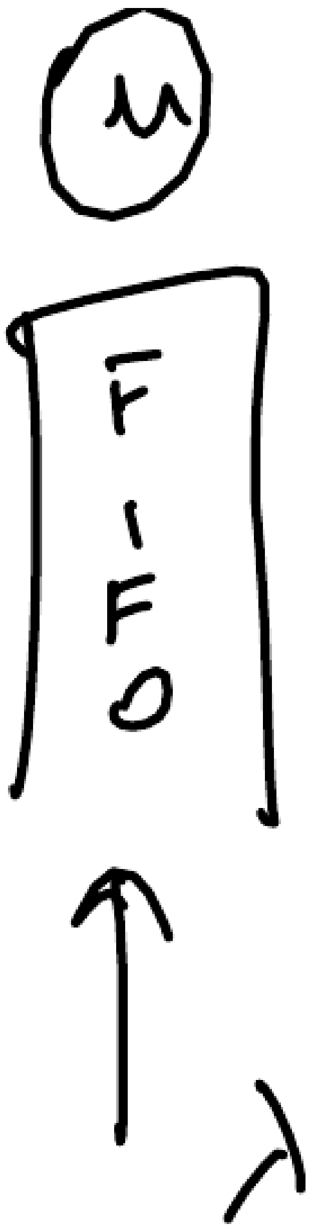
- One server, one FIFO queue for jobs to wait.
- μ denotes service rate, λ denotes the arrival rate.
- Service requirements S_n and inter-arrival times A_n are typically assumed to be i.i.d.
- In its simplest form, we will assume $S_n \sim Exp(\mu)$ and $A_n \sim Exp(\lambda)$.
- Jobs face queueing delay due to waiting for other jobs.
- This is the most basic $M/M/1$ queue.

A single server queue



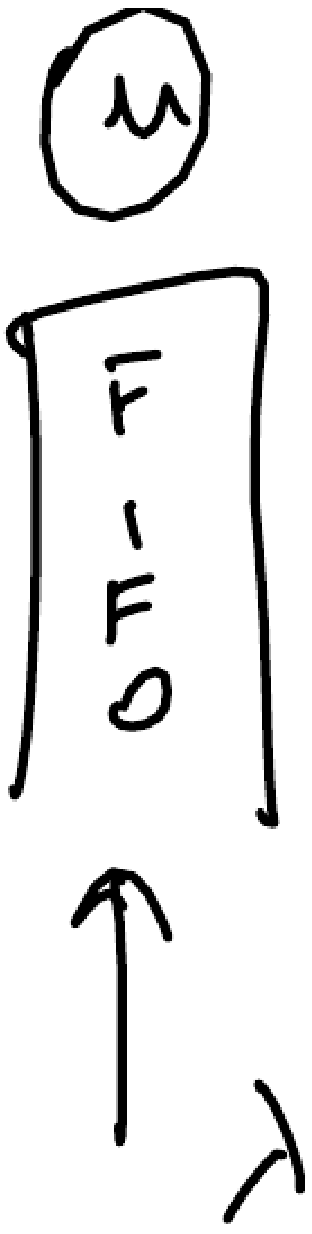
- One server, one FIFO queue for jobs to wait.
- μ denotes service rate, λ denotes the arrival rate.
- Service requirements S_n and inter-arrival times A_n are typically assumed to be i.i.d.
- In its simplest form, we will assume $S_n \sim Exp(\mu)$ and $A_n \sim Exp(\lambda)$.
- Jobs face queueing delay due to waiting for other jobs.
- This is the most basic $M/M/1$ queue. Modeling this as a Markov chain and solving its stationary distribution gives us mean response time (mean of service time + waiting time).

A single server queue

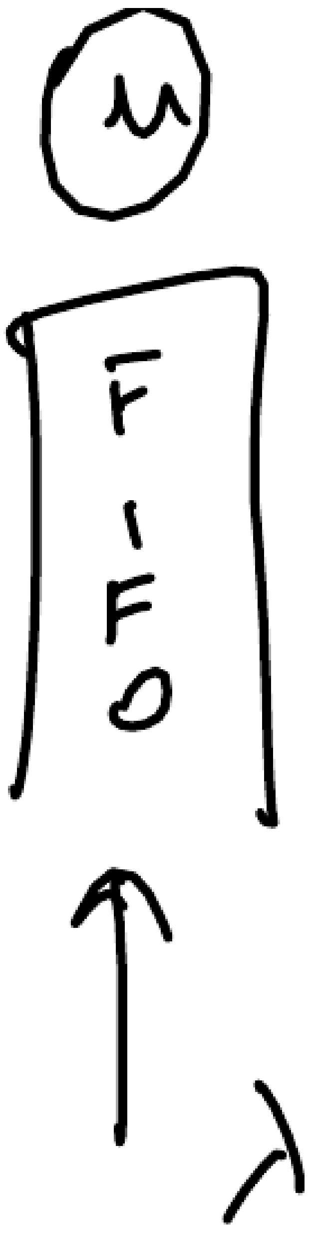


- One server, one FIFO queue for jobs to wait.
- μ denotes service rate, λ denotes the arrival rate.
- Service requirements S_n and inter-arrival times A_n are typically assumed to be i.i.d.
- In its simplest form, we will assume $S_n \sim Exp(\mu)$ and $A_n \sim Exp(\lambda)$.
- Jobs face queueing delay due to waiting for other jobs.
- This is the most basic $M/M/1$ queue. Modeling this as a Markov chain and solving its stationary distribution gives us mean response time (mean of service time + waiting time).
- $E[T] = \frac{1}{\mu - \lambda}$.

A single server queue

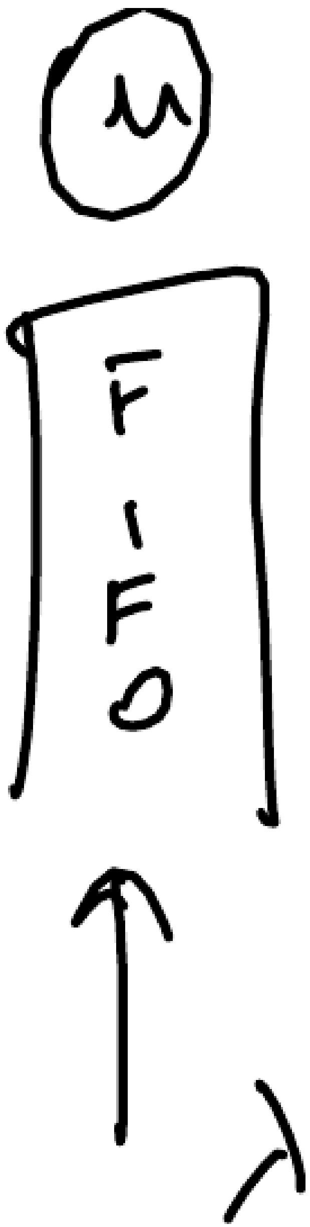


A single server queue



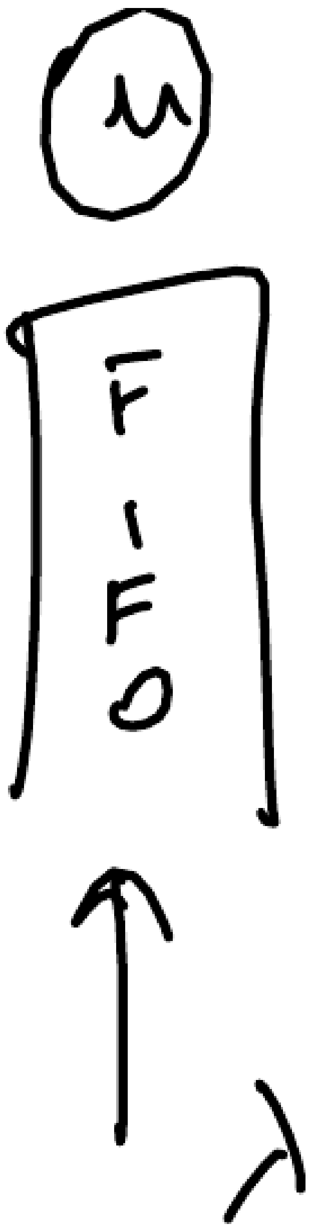
- $E[T] = \frac{1}{\mu - \lambda}$.

A single server queue



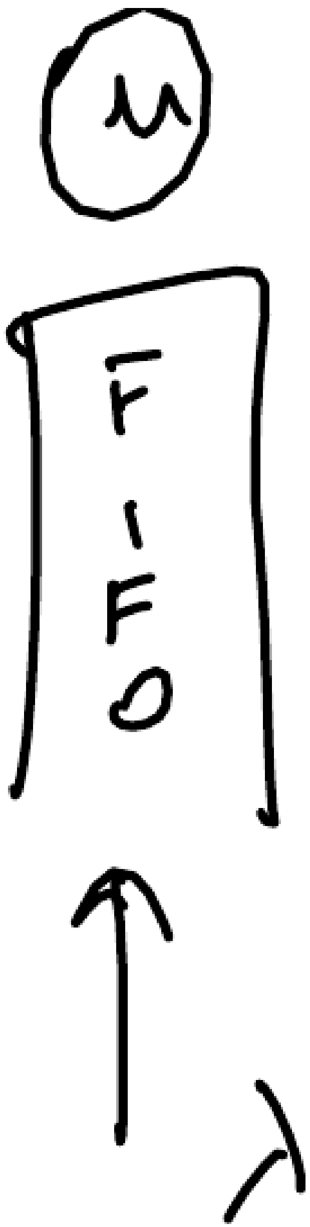
- $E[T] = \frac{1}{\mu - \lambda}$.
- Let N is the number of jobs in the system (Queue + server). Then what is $E[N]$?

A single server queue



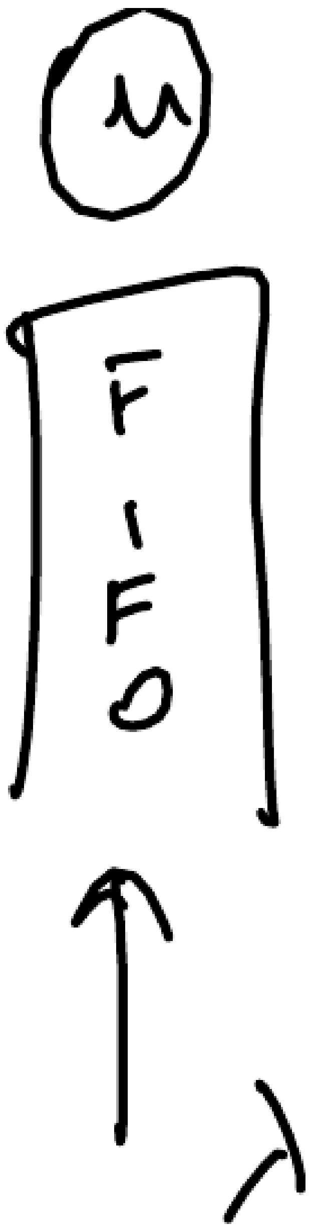
- $E[T] = \frac{1}{\mu - \lambda}$.
- Let N is the number of jobs in the system (Queue + server). Then what is $E[N]$?
- We will see Little's law that says that $E[N] = \lambda E[T]$.

A single server queue



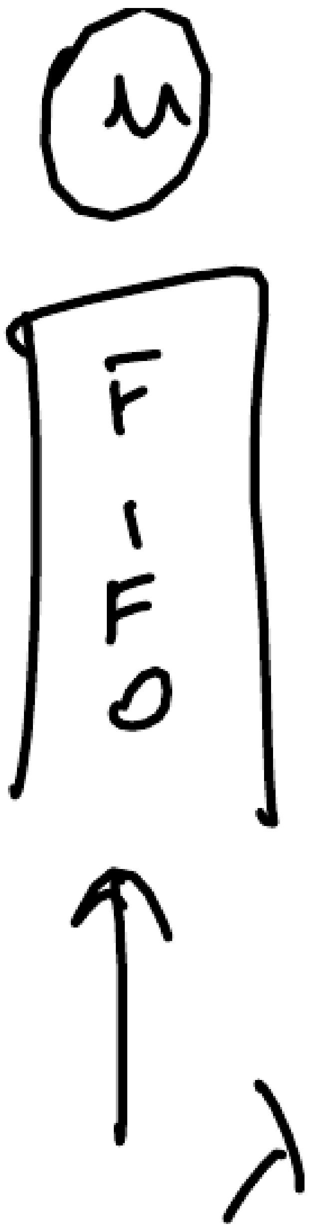
- $E[T] = \frac{1}{\mu - \lambda}$.
- Let N is the number of jobs in the system (Queue + server). Then what is $E[N]$?
- We will see Little's law that says that $E[N] = \lambda E[T]$.
- Mean number of jobs $E[N] = \frac{\lambda}{\mu - \lambda}$.

A single server queue

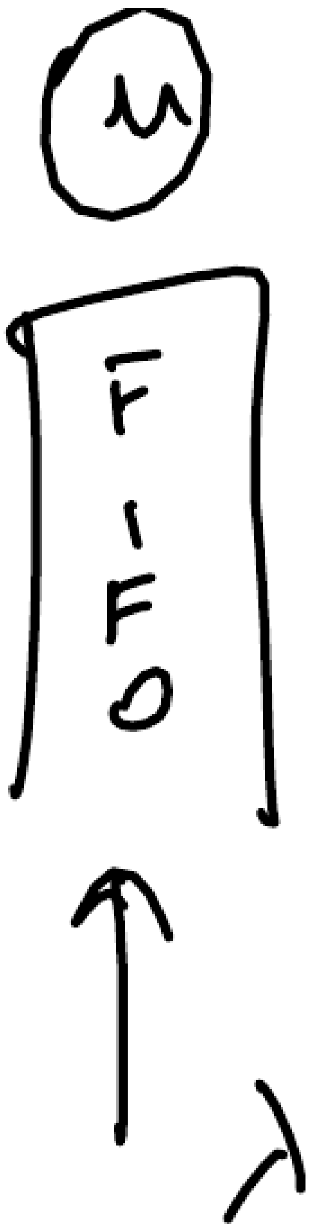


- $E[T] = \frac{1}{\mu - \lambda}$.
- Let N is the number of jobs in the system (Queue + server). Then what is $E[N]$?
- We will see Little's law that says that $E[N] = \lambda E[T]$.
- Mean number of jobs $E[N] = \frac{\lambda}{\mu - \lambda}$.
- This course is about Markov chain analysis to derive such formulas.

Example 1: Doubling the arrival rate

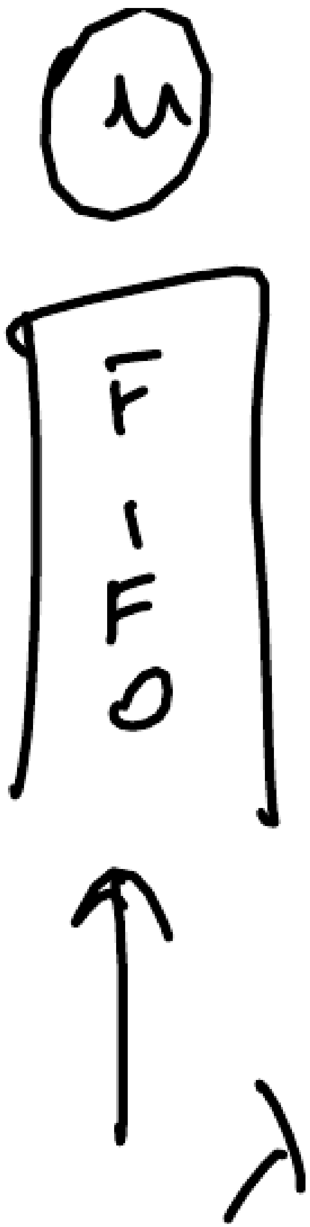


Example 1: Doubling the arrival rate



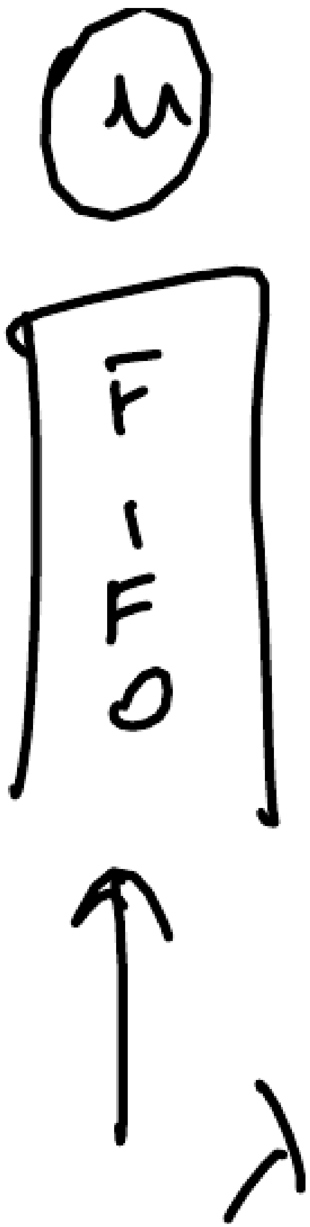
- $E[T] = \frac{1}{\mu - \lambda}$.

Example 1: Doubling the arrival rate



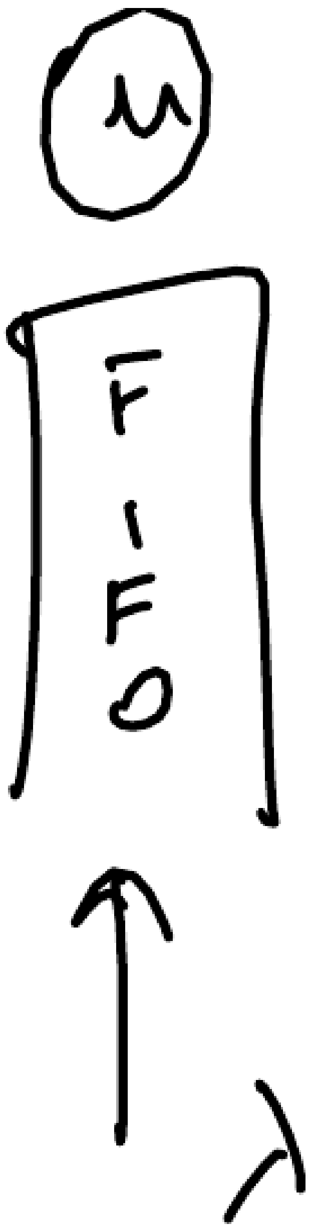
- $E[T] = \frac{1}{\mu - \lambda}$.
- What would happen to $E[T]$ if $\lambda \to 2\lambda$?

Example 1: Doubling the arrival rate



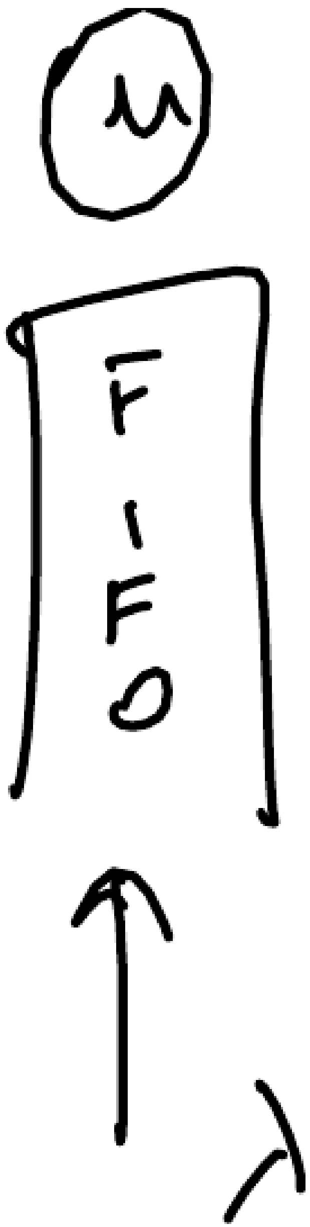
- $E[T] = \frac{1}{\mu-\lambda}$.
- What would happen to $E[T]$ if $\lambda \to 2\lambda$?
- It could blow up if $\mu < 2\lambda$.

Example 1: Doubling the arrival rate



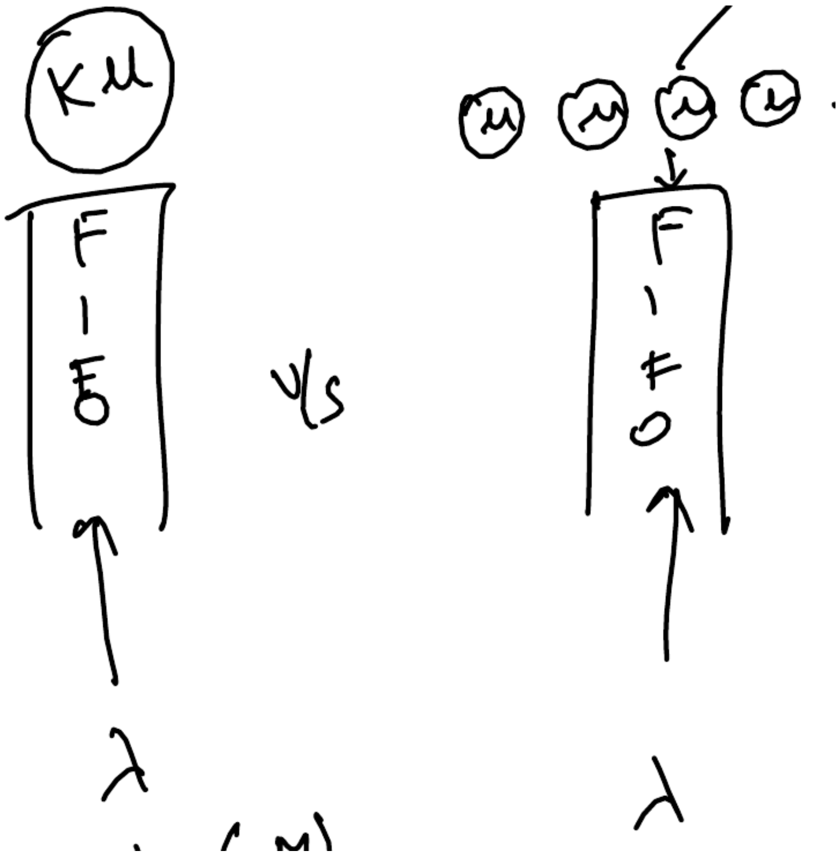
- $E[T] = \frac{1}{\mu - \lambda}$.
- What would happen to $E[T]$ if $\lambda \to 2\lambda$?
- It could blow up if $\mu < 2\lambda$.
- If you want to maintain the same level of response time then do you need to double μ?

Example 1: Doubling the arrival rate



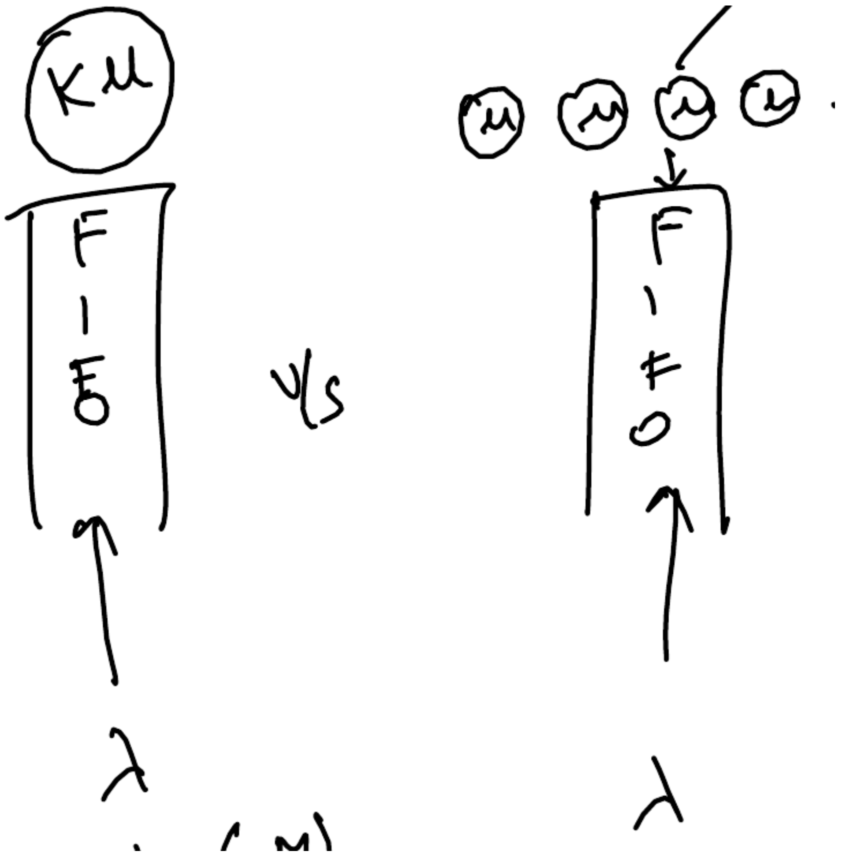
- $E[T] = \frac{1}{\mu - \lambda}$.
- What would happen to $E[T]$ if $\lambda \to 2\lambda$?
- It could blow up if $\mu < 2\lambda$.
- If you want to maintain the same level of response time then do you need to double μ?
- This course is about making such design choices!

Example 2: A fast server versus many slow servers



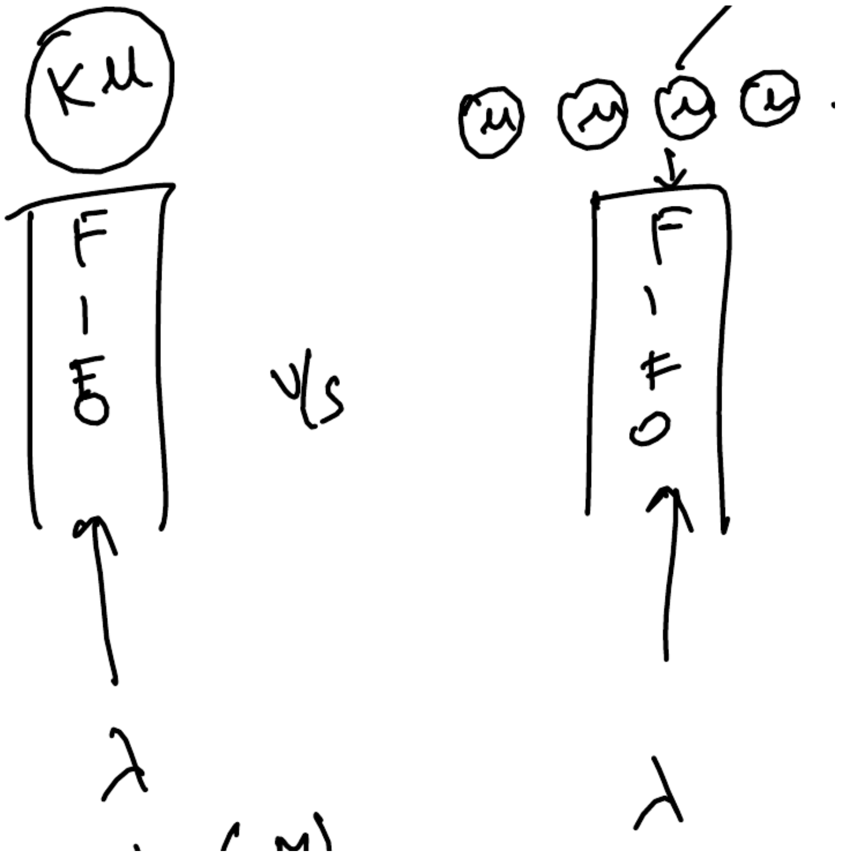
- Which system will have lower $E[T]$?

Example 2: A fast server versus many slow servers



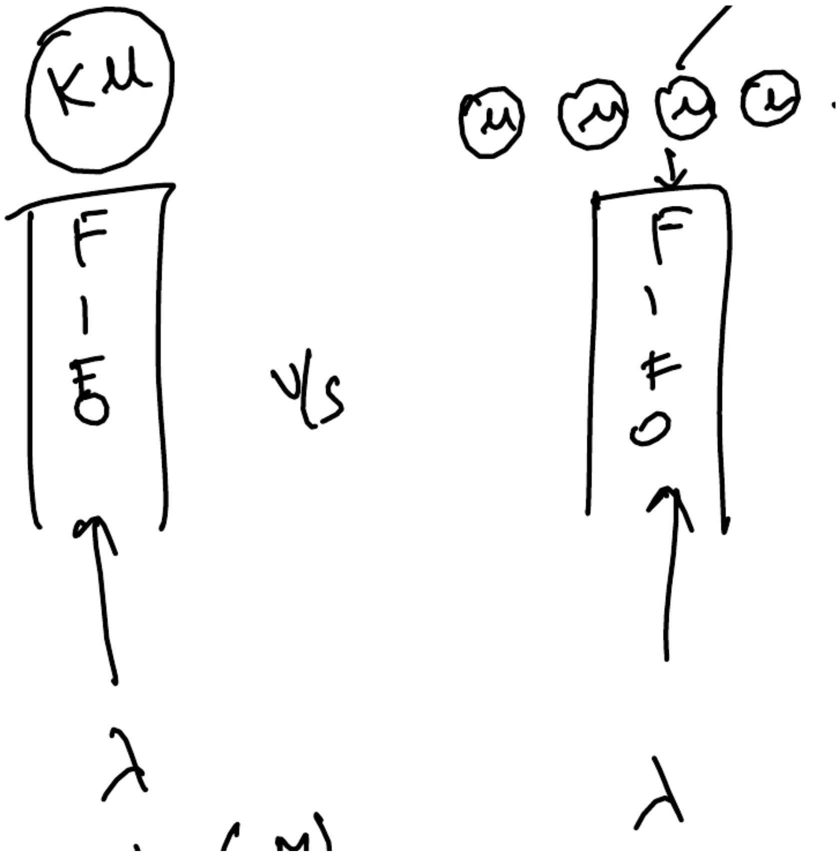
- Which system will have lower $E[T]$?
- Is a fast server ($K\mu$) better that K normal servers (μ)?

Example 2: A fast server versus many slow servers



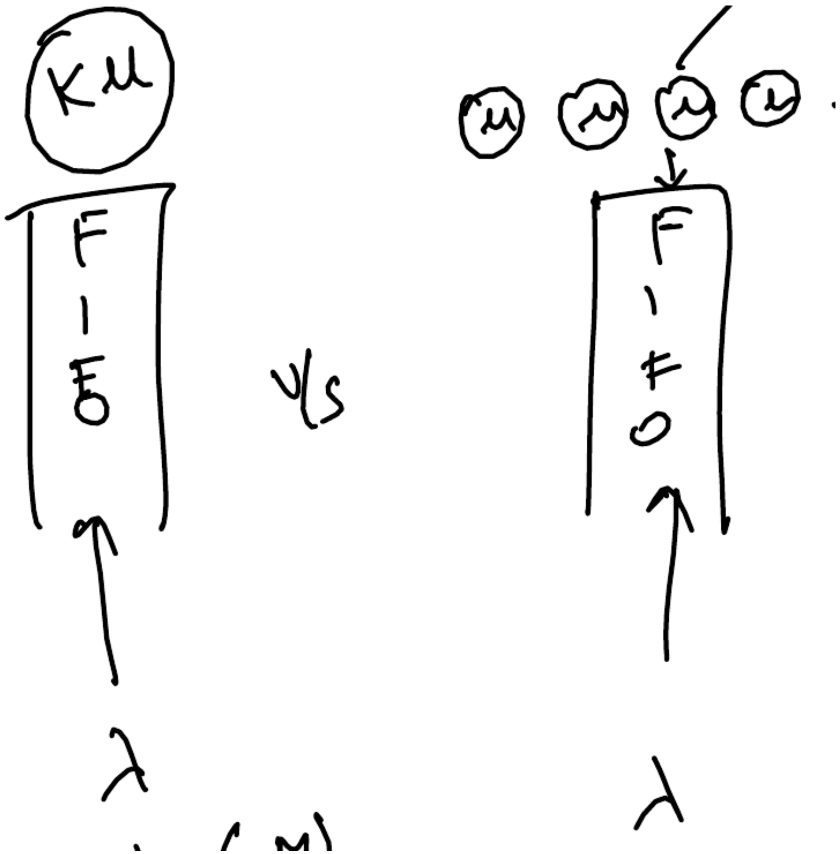
- Which system will have lower $E[T]$?
- Is a fast server ($K\mu$) better that K normal servers (μ)?
- Does job variability impact this decision?

Example 2: A fast server versus many slow servers



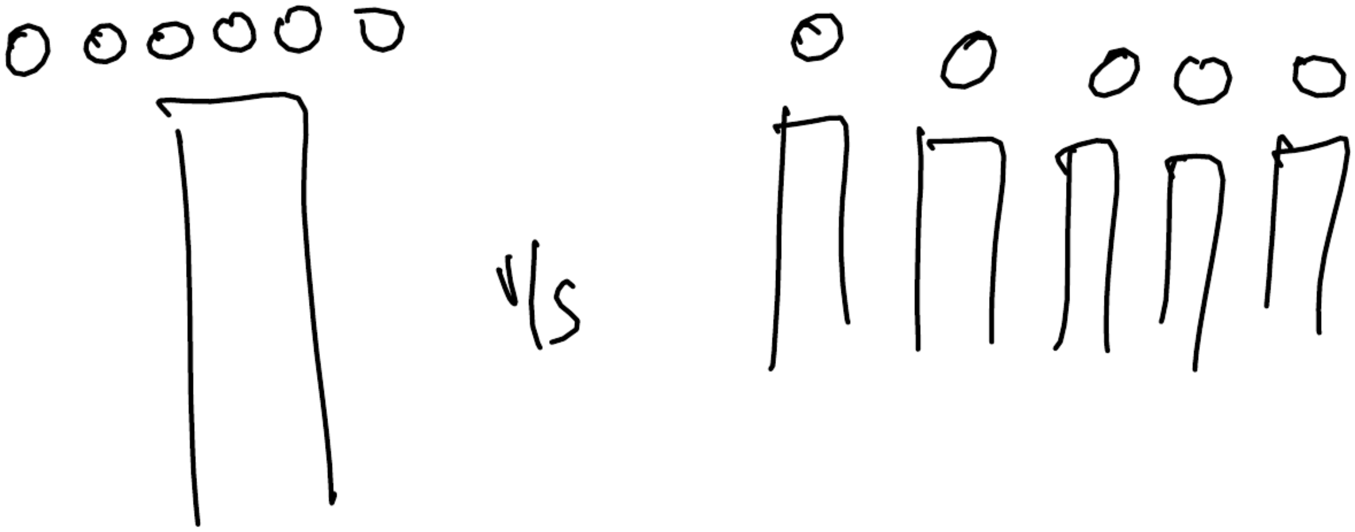
- Which system will have lower $E[T]$?
- Is a fast server $(K\mu)$ better that K normal servers (μ)?
- Does job variability impact this decision? Suppose job sizes were XS, S, M, L, XL.

Example 2: A fast server versus many slow servers



- Which system will have lower $E[T]$?
- Is a fast server ($K\mu$) better that K normal servers (μ)?
- Does job variability impact this decision? Suppose job sizes were XS, S, M, L, XL.
- In the first model, an S, or M job has to possibly wait behind XL. This is avoided in the second scenario.

Example 3: Central queue or individual



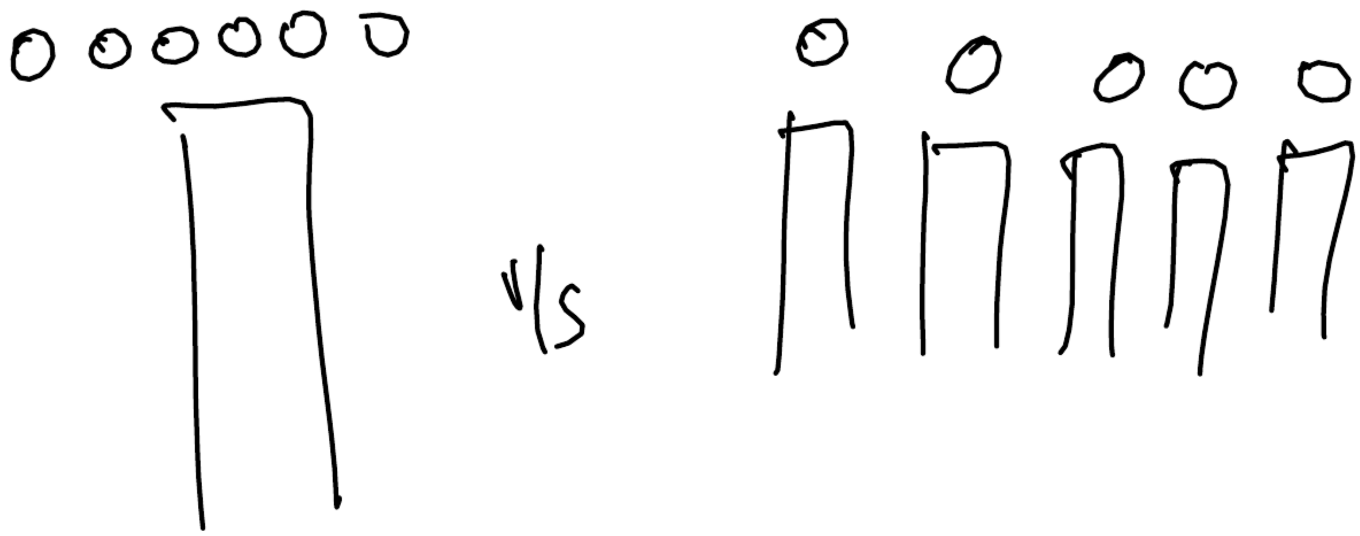
Example 3: Central queue or individual

- At Airport immigration, Hotel check-ins you often see central queues.

Example 3: Central queue or individual

- At Airport immigration, Hotel check-ins you often see central queues.
- But at movie theatres, metro/train ticket counters, you see the second model.

Example 3: Central queue or individual

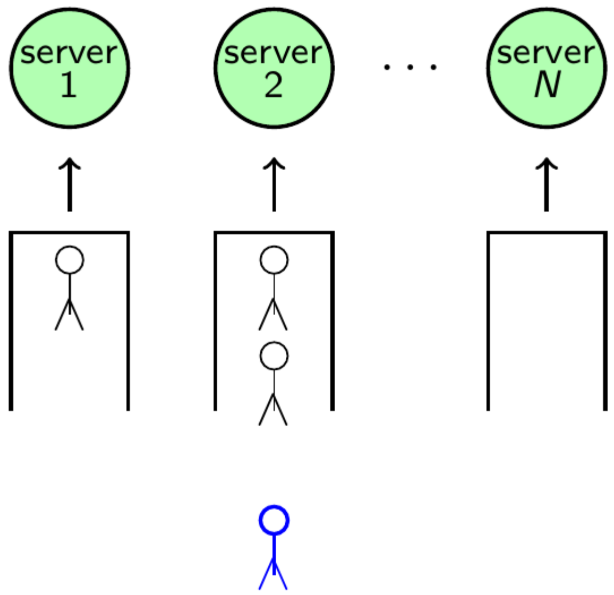


- At Airport immigration, Hotel check-ins you often see central queues.
- But at movie theatres, metro/train ticket counters, you see the second model.
- Which setting has a lower $E[T]$?

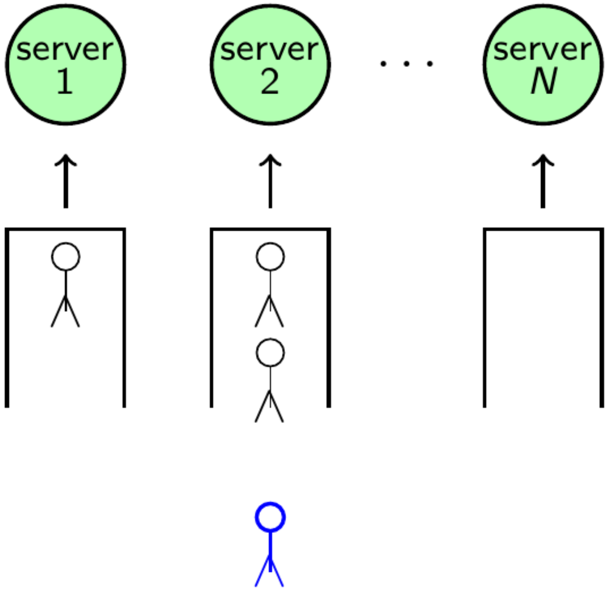
Example 3: Central queue or individual

- At Airport immigration, Hotel check-ins you often see central queues.
- But at movie theatres, metro/train ticket counters, you see the second model.
- Which setting has a lower $E[T]$?
- This course will help you answer such performance modeling questions.

Example 4: Supermarket queue and load balancing

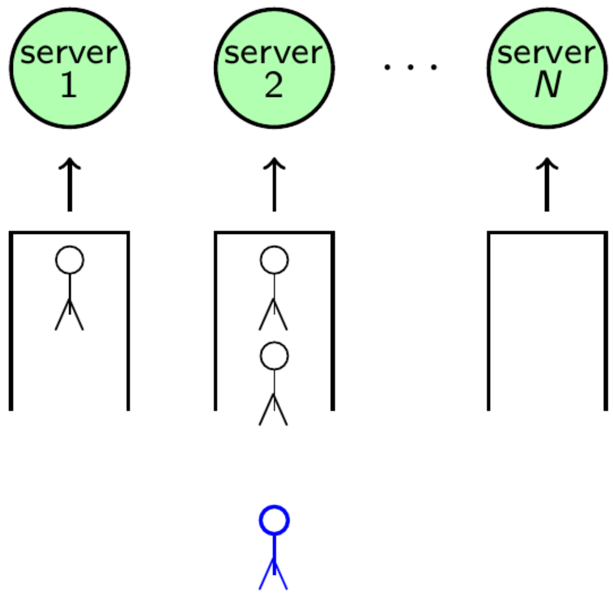


Example 4: Supermarket queue and load balancing



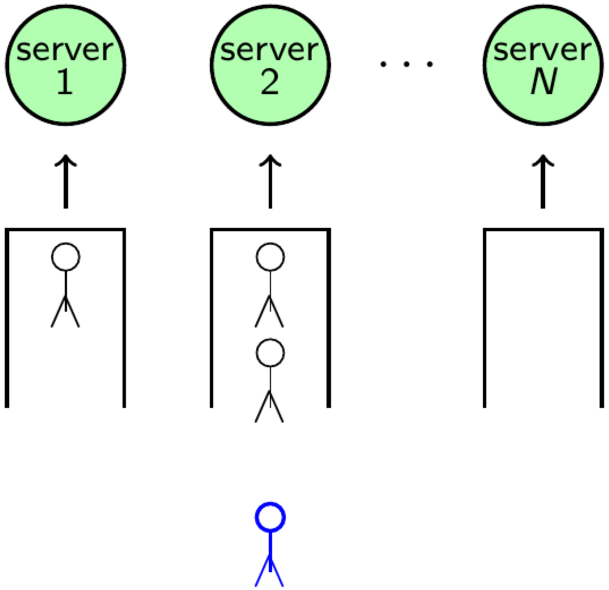
- Load balancing concerns the questions which queue to join/assign?

Example 4: Supermarket queue and load balancing



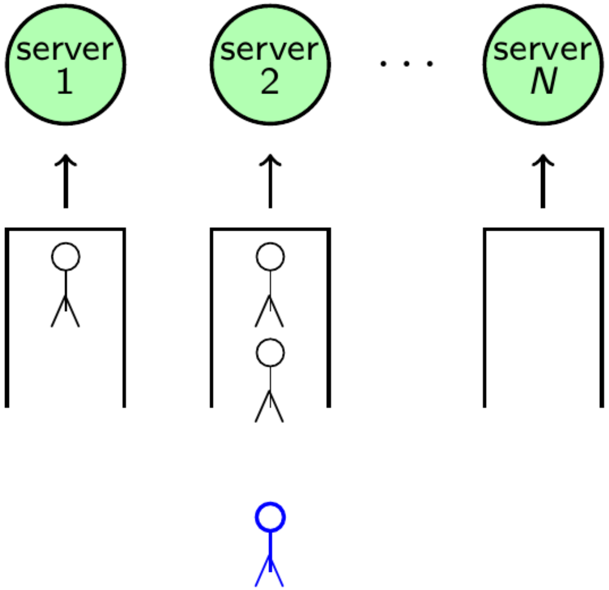
- Load balancing concerns the questions which queue to join/assign?
- Popular policy is Join shortest Queue (JSQ).

Example 4: Supermarket queue and load balancing



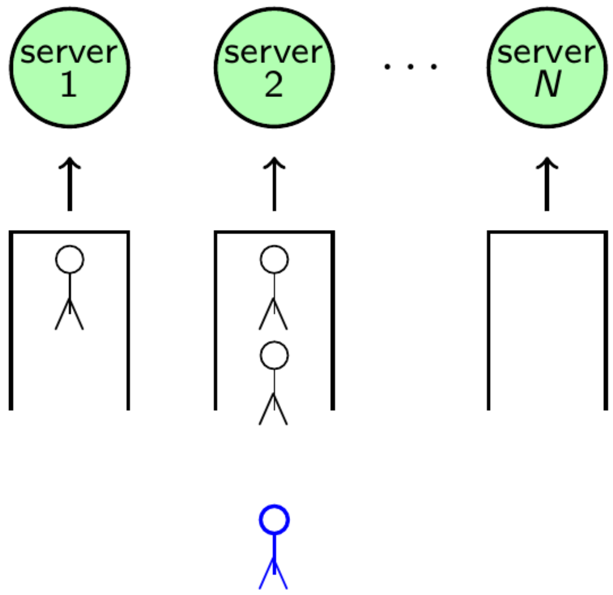
- Load balancing concerns the questions which queue to join/assign?
- Popular policy is Join shortest Queue (JSQ).
- What should be ideally done is Join smallest work (JSW).

Example 4: Supermarket queue and load balancing

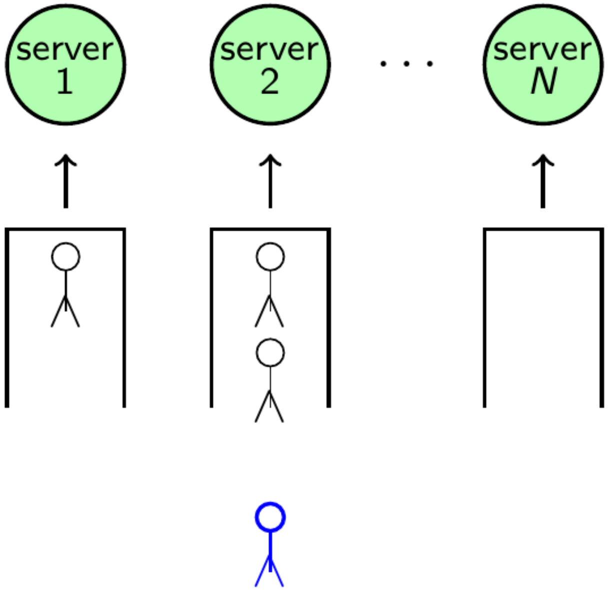


- Load balancing concerns the questions which queue to join/assign?
- Popular policy is Join shortest Queue (JSQ).
- What should be ideally done is Join smallest work (JSW).
- N is typically large and the overhead in obtaining queue length information is huge ($2N$).

Example 4: Supermarket queue and load balancing

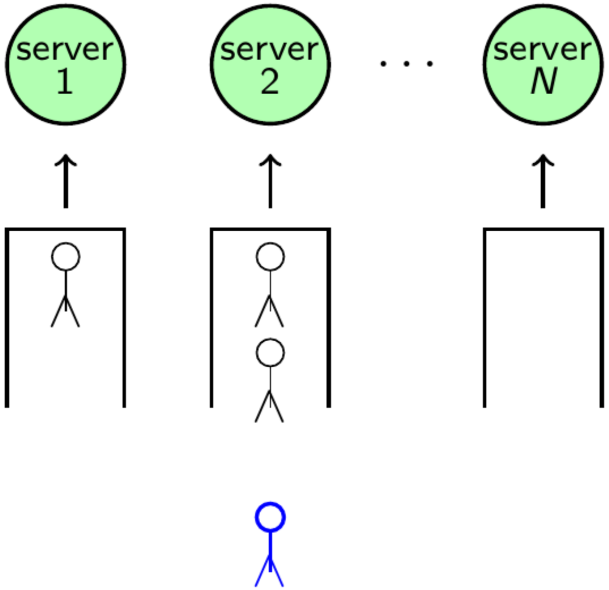


Example 4: Supermarket queue and load balancing



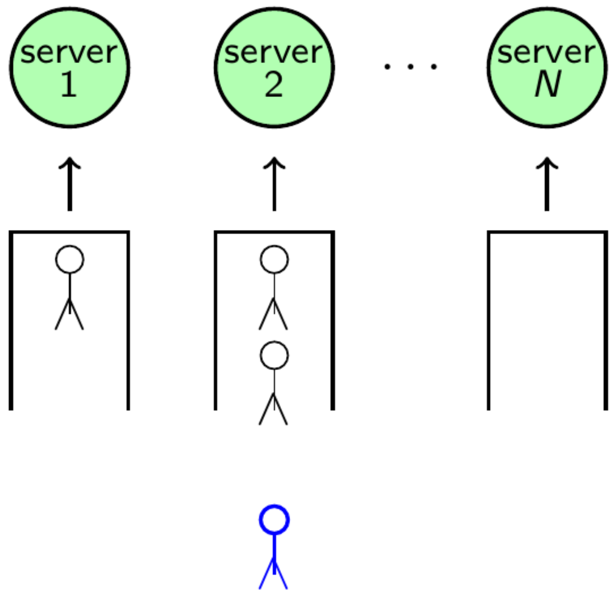
- In that case, sample d servers randomly and join appropriate queue using $JSQ(d)$ or $JSW(d)$.

Example 4: Supermarket queue and load balancing



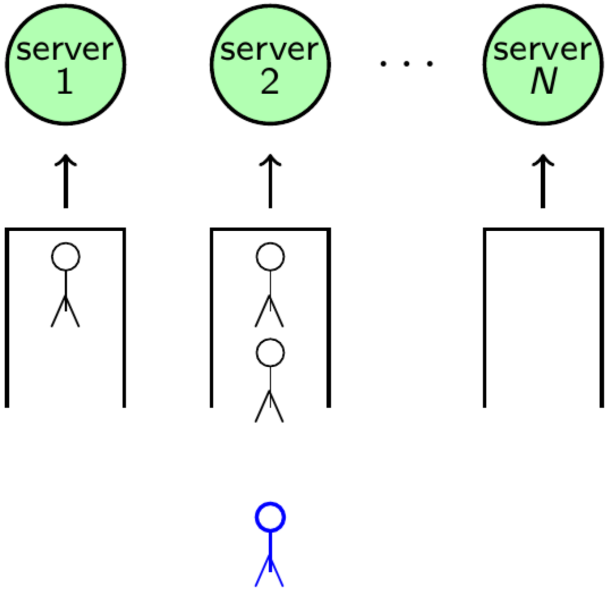
- In that case, sample d servers randomly and join appropriate queue using $JSQ(d)$ or $JSW(d)$.
- Problem with JSW or $JSW(d)$ is that the workload information is typically unknown.

Example 4: Supermarket queue and load balancing



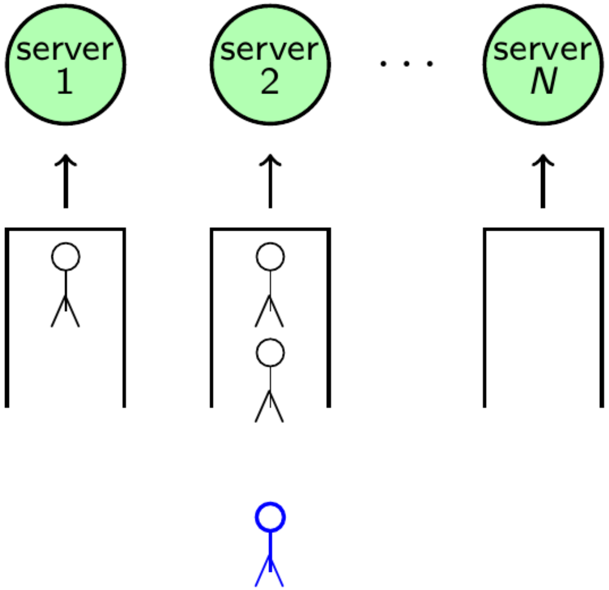
- In that case, sample d servers randomly and join appropriate queue using $JSQ(d)$ or $JSW(d)$.
- Problem with JSW or $JSW(d)$ is that the workload information is typically unknown. How to implement it then?

Example 4: Supermarket queue and load balancing



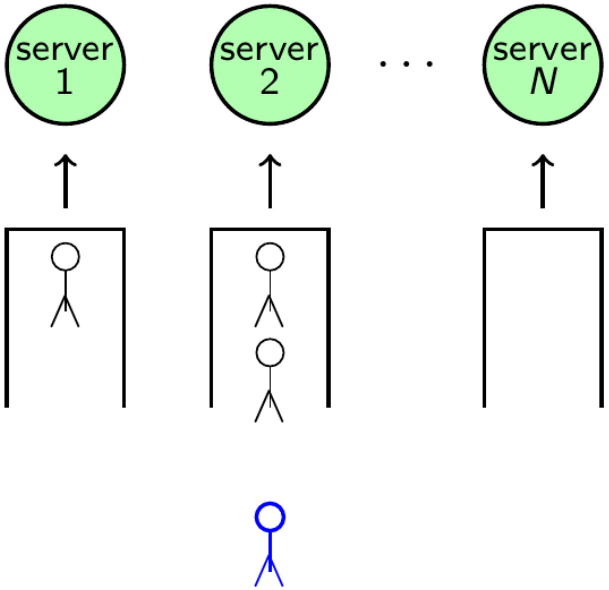
- In that case, sample d servers randomly and join appropriate queue using $JSQ(d)$ or $JSW(d)$.
- Problem with JSW or $JSW(d)$ is that the workload information is typically unknown. How to implement it then?
- How about replicating jobs on d servers and cancelling copies when one copy starts service ?

Example 4: Supermarket queue and load balancing



- In that case, sample d servers randomly and join appropriate queue using $JSQ(d)$ or $JSW(d)$.
- Problem with JSW or $JSW(d)$ is that the workload information is typically unknown. How to implement it then?
- How about replicating jobs on d servers and cancelling copies when one copy starts service ?
- This is redundancy-d with cancel on start.

Example 4: Supermarket queue and load balancing



- In that case, sample d servers randomly and join appropriate queue using $JSQ(d)$ or $JSW(d)$.
- Problem with JSW or $JSW(d)$ is that the workload information is typically unknown. How to implement it then?
- How about replicating jobs on d servers and cancelling copies when one copy starts service ?
- This is redundancy-d with cancel on start.
- We do this at super-markets all the time!