Queueing Theory

A Mathematically Rigorous yet Intuitive Guide

By
Rishabh Pomaje
&
Samyak Sanjay Parakh

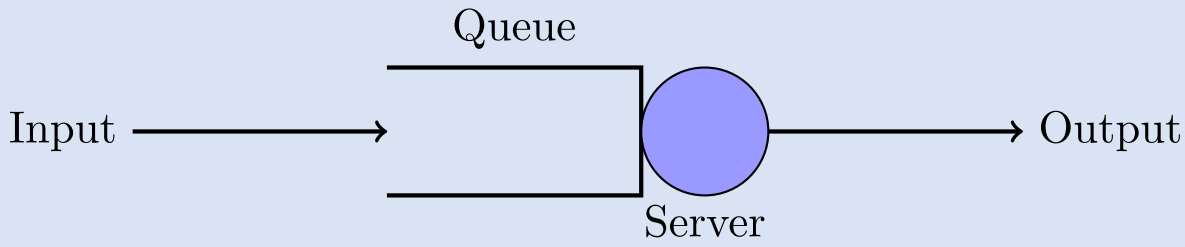
Motivation & Background

- Queueing Theory
  - *'Study of Waiting'.*
- Why do queues form?
  - Finite Resources
  - Finite efficiency
- Why to study queues?
  - Resource Planning
  - Decision Making
- What's there to study?
  - Randomness in Arrival times
  - Randomness in Demands.

Queue *Server*

Arrivals	Server	Resource
People	Doctor Bank Manager	Doctor's time Manager's time
Planes	Air Traffic Controller	Runways at an airport.
System Processes (Tasks)	Microprocessor Or Microcontroller	Memory + Compute resources (Cores)

Examples of Queues



Basic Assumptions

- Whenever a server is free, and there is a customer in the queue, the customer immediately goes into service.
- Whenever a server is busy if a customer arrives, the customer waits in the queue*.
- Whenever a server becomes free, the next customer is decided according to a *scheduling policy/ discipline.*
- Customers do not become impatient and leave the queue before their service completion.

Modeling Queues

- Simulate/ Model Randomness $\Longrightarrow$ Probability distributions
- Select statistics of
  - Arrival spacing
  - Service spacing
  - Both in temporal dimension.
- *Scheduling Policy or Discipline*
  - FIFO/ LIFO/ LIFOPR/ Priority/ SJB/ Round Robin.
  - Feedback into queue.

- Max capacity of the queue*.
  - Overflow.
  - Customers are "blocked/turned away/dropped."

Notation

- *Kendall Notation*

$$A/B/C/X/Y/Z$$

- A = Inter-arrival time distribution
- B = Service time distribution
- C = Number of servers
- X = System Capacity
- Y = Size of the customer population
- Z = Queue scheduling discipline

Notation

- State of the system at time t : $N(t)$ or simply N
  - Number in queue = Waiting + those being served.

- $P_{N(t)}(k)$ = probability(k customers in the queue).
  - p_n = probability of same but at equilibrium.
  - $p_{i,j}$ = probability of going from state i to state j.

- $\overline{N}$ = average (expectation) of variable N.

- var(N) = variance of the number of customers.

- Utilization = fraction of time the server is busy*.

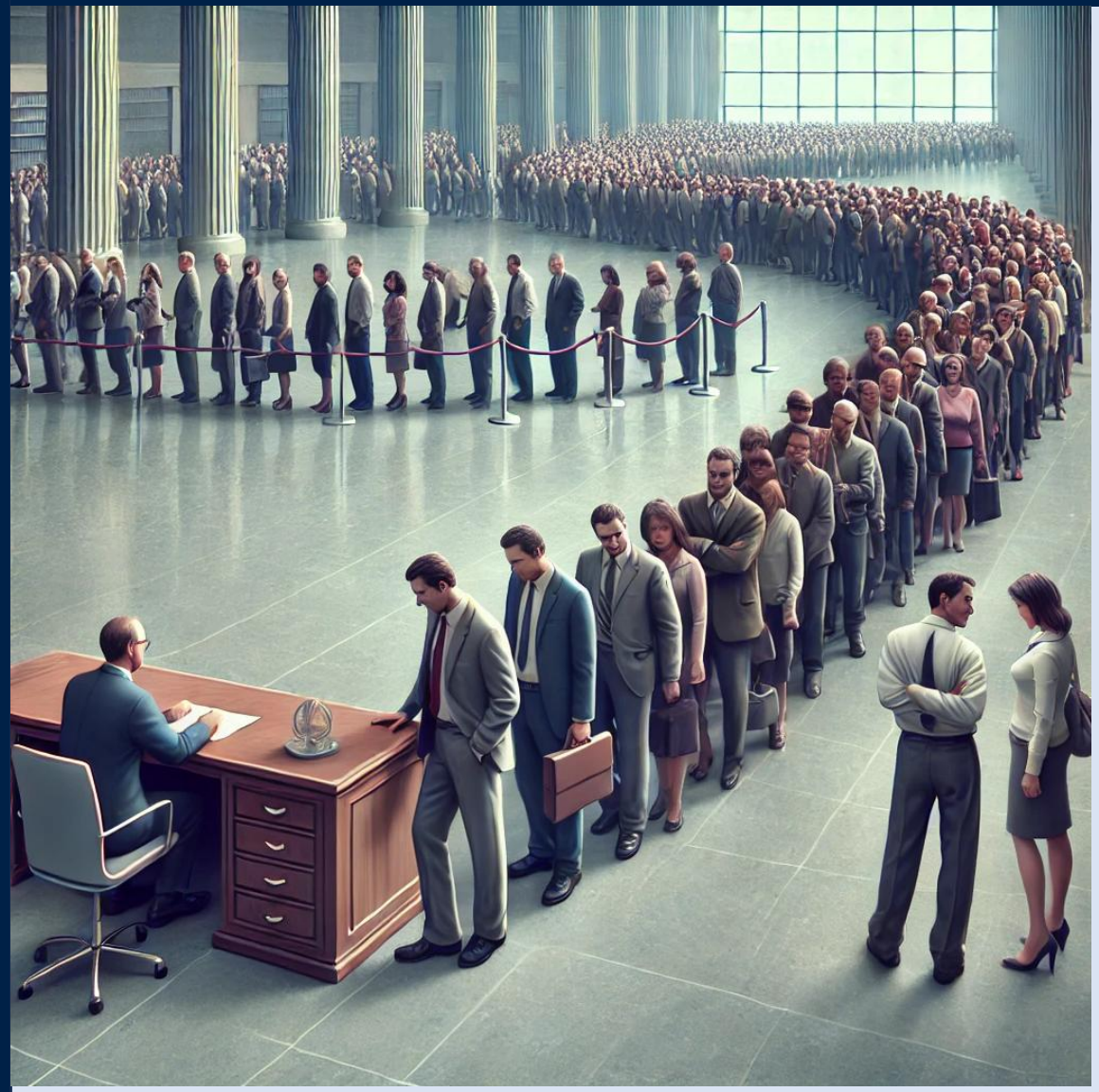
Objectives

- Preliminary analysis
  - First-order analysis.

- We will be looking at steady-state analysis.
  - The system is in equilibrium.
  - The state distributions, i.e., prob(system is in some state i), become time-invariant.

- For each of the cases, we have some measures/metrics of performance that depend on the state probabilities or state distribution.

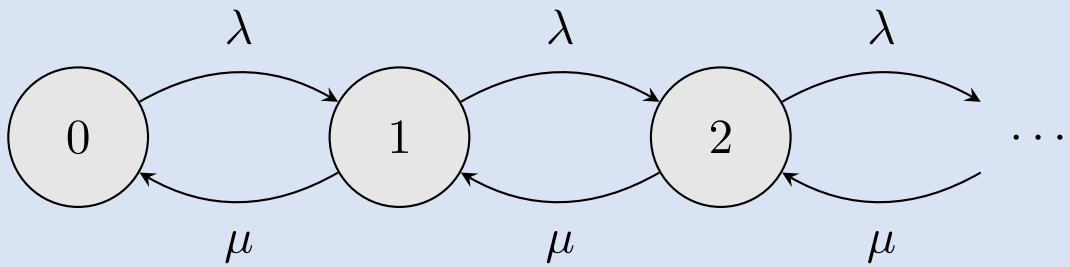
M/M/1 Queue – a recap



- Model description:
  1. Poisson arrival process with rate λ.
  2. Exponential service times with rate μ.
  3. Single server
  4. FIFO scheduling discipline.
  5. Infinite buffer size

Source: DALL · E

Overview of Derivation



We get the equations,

$$P_n(t+\delta t) = P_n(t)p_{n,n} + P_{n-1}(t)p_{n-1,n} + P_{n+1}(t)p_{n+1,n}$$
$$P_0(t+\delta t) = P_0(t)p_{0,0} + P_1(t)p_{1,0}$$

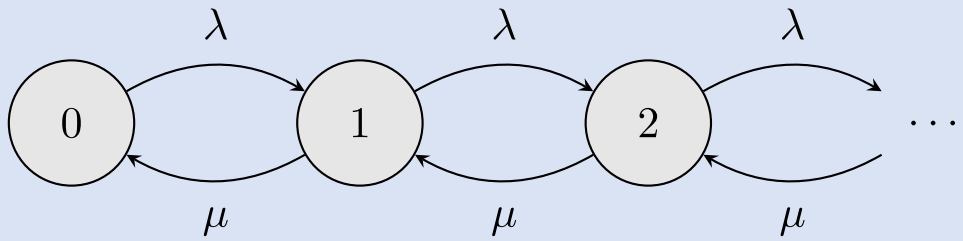
$\delta t \to 0$

$$\frac{dP_n(t)}{dt} = -(\lambda+\mu)P_n(t) + \lambda P_{n-1}(t) + \mu P_{n+1}(t)$$
$$\frac{dP_0(t)}{dt} = -\lambda P_0(t) + \mu P_1(t)$$

Digression – Probability Flux

- **Definition:** (Probability flux is defined along a transition.)
  - Φ_p = P(state where the transition originates) × transition rate of the transition.

$$\begin{aligned}\Phi_P(\text{node 1}) &= \text{Flow out of node 1} + \text{Flow into node 1} \\ &= -(\lambda+\mu)p_1 + \lambda p_0 + \mu p_2\end{aligned}$$



  - Looks familiar?

  - This is exactly the RHS of the differential equation. – *Global Balance Equation.*

  - *Local Balance Equations.*
    - Formed by equating flux across state boundaries.

Solution

- Local Balance equations:

$$\lambda p_0 = \mu p_1 \implies p_1 = \frac{\lambda}{\mu} p_0$$

$$\lambda p_1 = \mu p_2 \implies p_2 = \frac{\lambda}{\mu} p_1$$

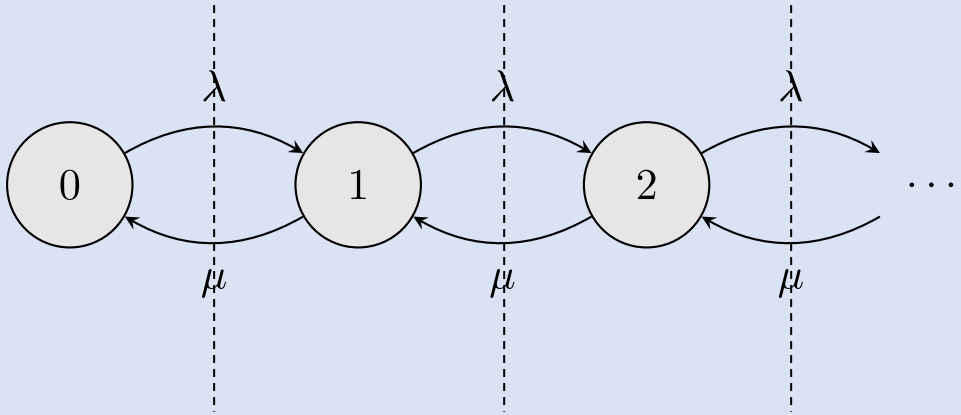
$$\vdots$$

$$\lambda p_{n-1} = \mu p_n \implies p_n = \frac{\lambda}{\mu} p_{n-1}$$

$$\vdots$$

$$p_n = \left(\frac{\lambda}{\mu}\right)^n p_0$$

$$\sum_{i=0}^{\text{num}_{\text{states}}} p_i = 1.$$



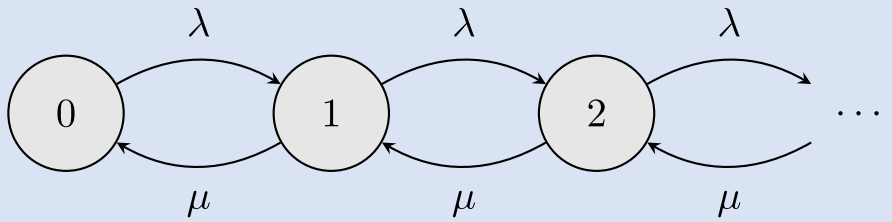
Combining all of them we get the following solution:

$$p_0 = 1 - \rho$$

$$p_n = \rho^n p_0$$

$$\rho = \lambda / \mu$$

M/M/1 - Summary



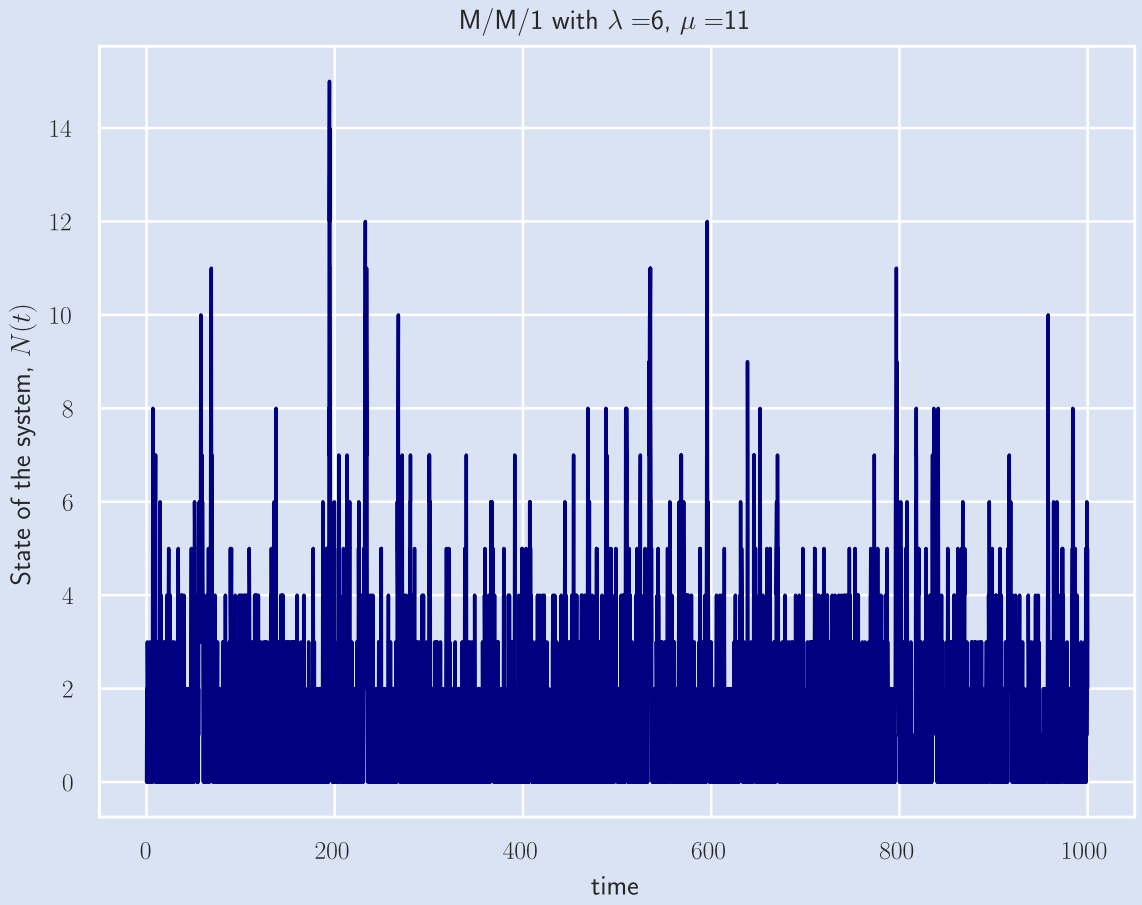
Random Variable	Distribution
Number of arrivals in interval (0, t]	Poisson(λt)
Inter-arrival times	Exponential(λ)
Number of departures in the interval (0, t]	Poisson(μt)
Inter-departure times	Exponential(μ)

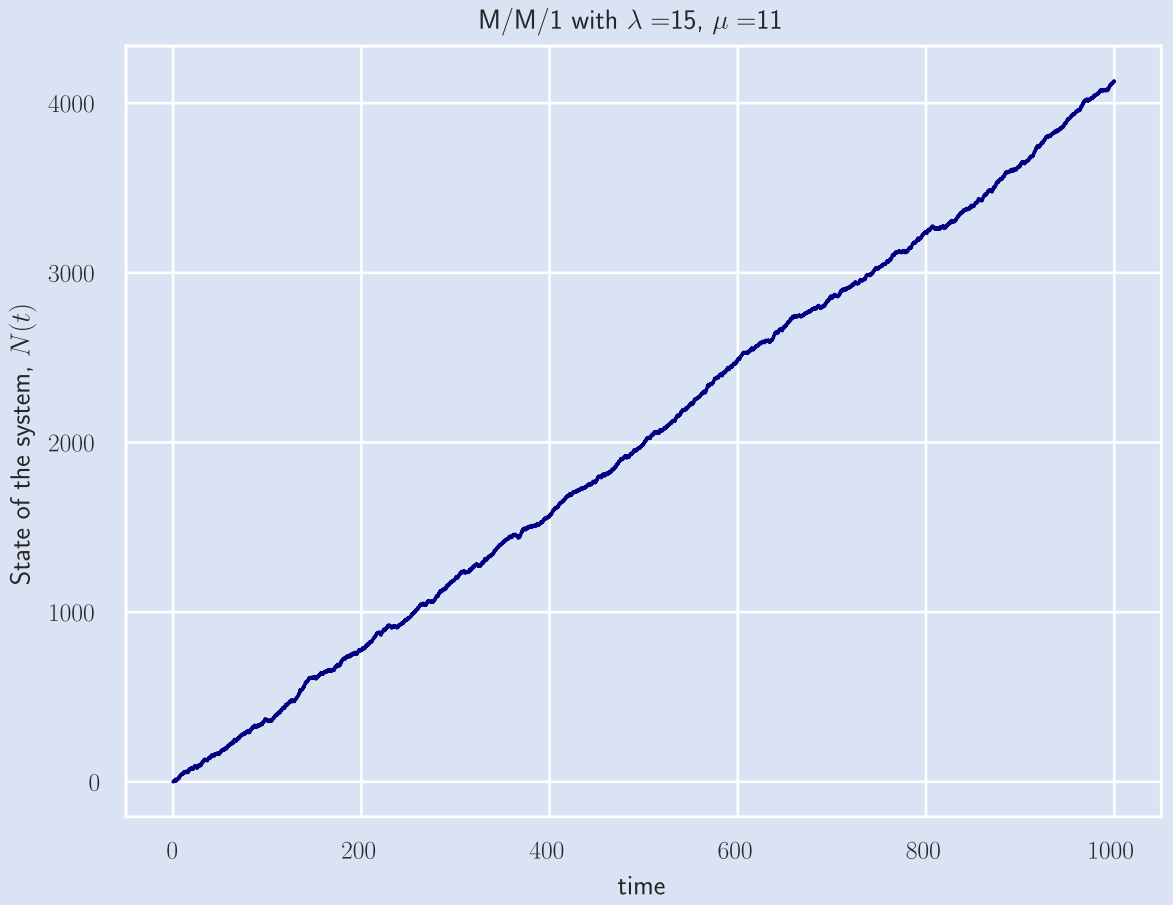
At equilibrium, $\lambda < \mu$,

State Probabilities	$p_0 = 1 - \rho$ $p_n = \rho^n p_0$
Average number of customers in the queue	$\rho/(1-\rho)$
Variance of number of customers in the queue	$\rho/(1-\rho)^2$

M/M/1 - Simulations

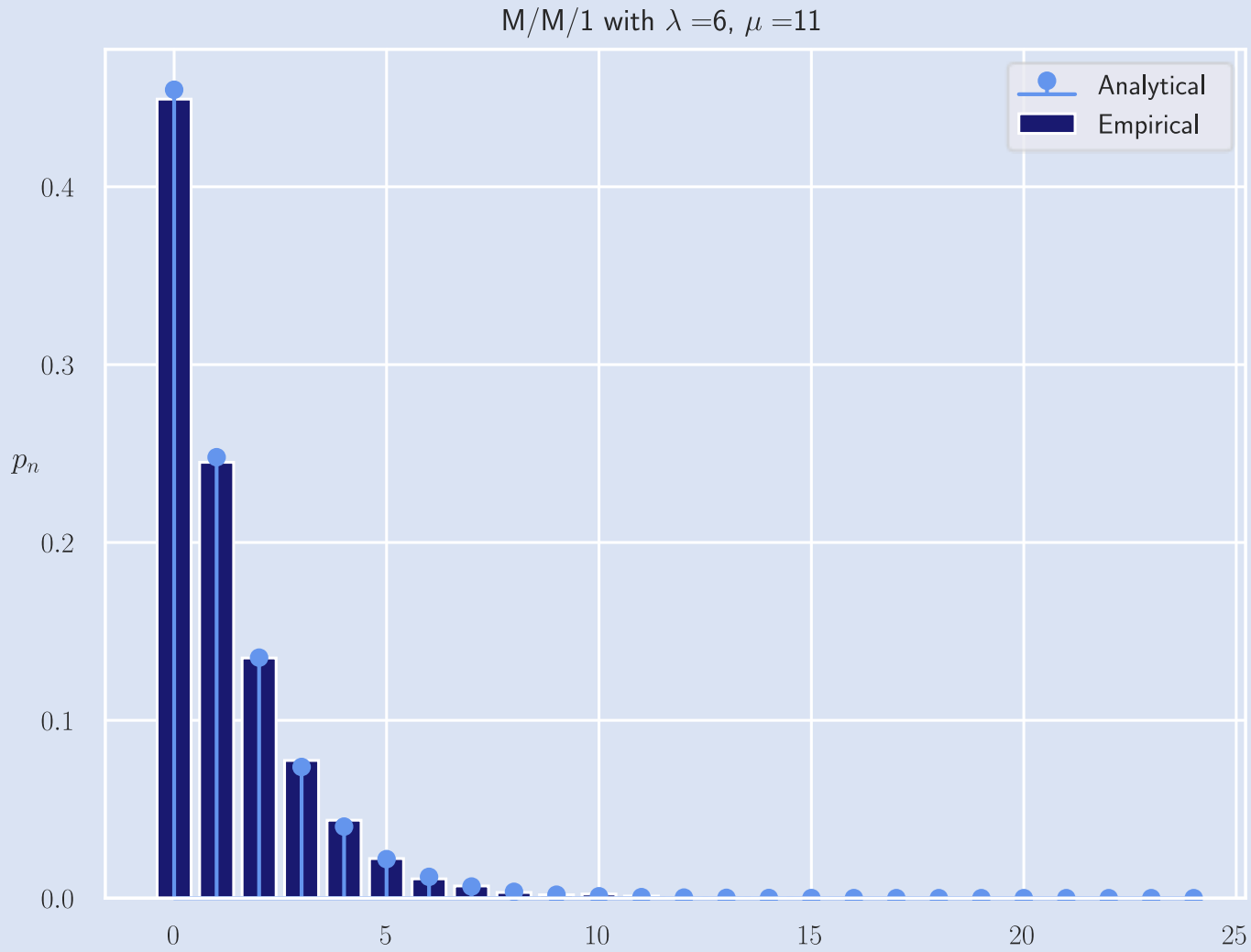
- Evolution of the state with time $\lambda \in \{6, 15\}, \mu = 11.$





M/M/1 - Simulations

- Stationary State Distribution: $\lambda = 6, \mu = 11$.



M/M/1 - Insights

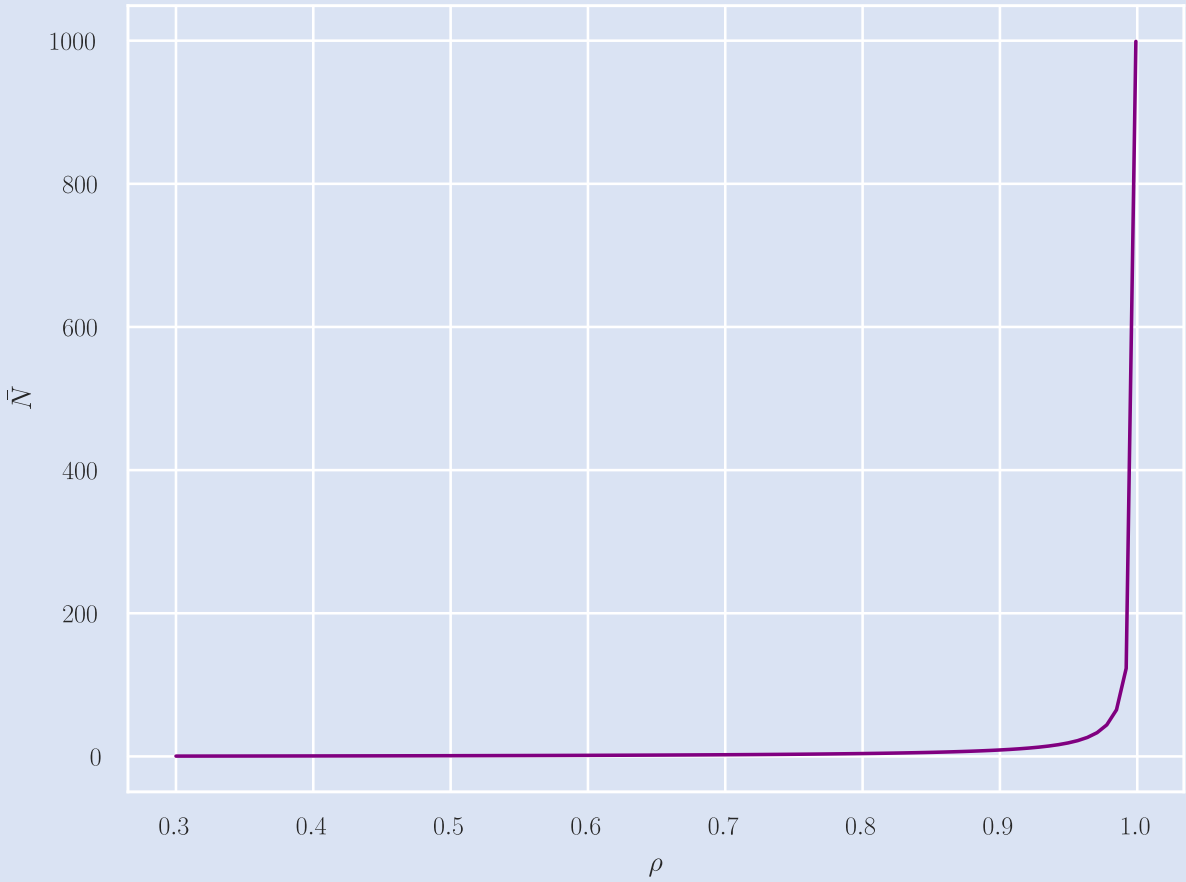
- For equilibrium, we must have $\lambda < \mu$.

Performance metrics: Mean and Variance.

1. As the utilization tends to 1, mean and variance of the system state blow up to ∞.

   $$\mathbb{E}[N] = \frac{\rho}{1-\rho}, var(N) = \frac{\rho^2}{1-\rho}.$$

2. Randomness grows without bounds
3. *Response time ≠ Service time*
   1. Waiting + Service
   2. Any relation between Response time and arrival rate?
   3. *Little's Law* : $\bar{N} = \lambda \bar{T}$



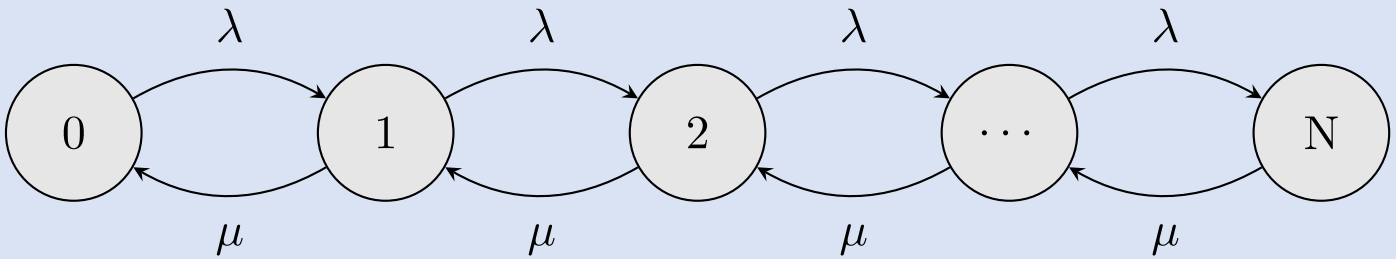
Mean v/s ρ (Utilization)

M/M/1/N – Finite buffer size

- Why finite buffer size?
  - Finite capacity.
  - *Example*. An airport is unlikely to be able to accommodate infinite airplanes.
- Model description:
  1. (1) – (5) from M/M/1.
  2. Finite buffer size of N.
  3. Any excess/ additional arrivals
     - Turned down/ blocked/ dropped.
     - Do not return if blocked once.

Source: DALL · E

M/M/1/N - Summary



- ! Caution: Now we have FSM. So same equations, different limits on sum.
- In equilibrium, $\sum_{n=0}^{\infty} \rightarrow \sum_{n=0}^{N}$

$$p_0 + \rho p_0 + \rho^2 p_0 + \ldots + p_N = 1$$

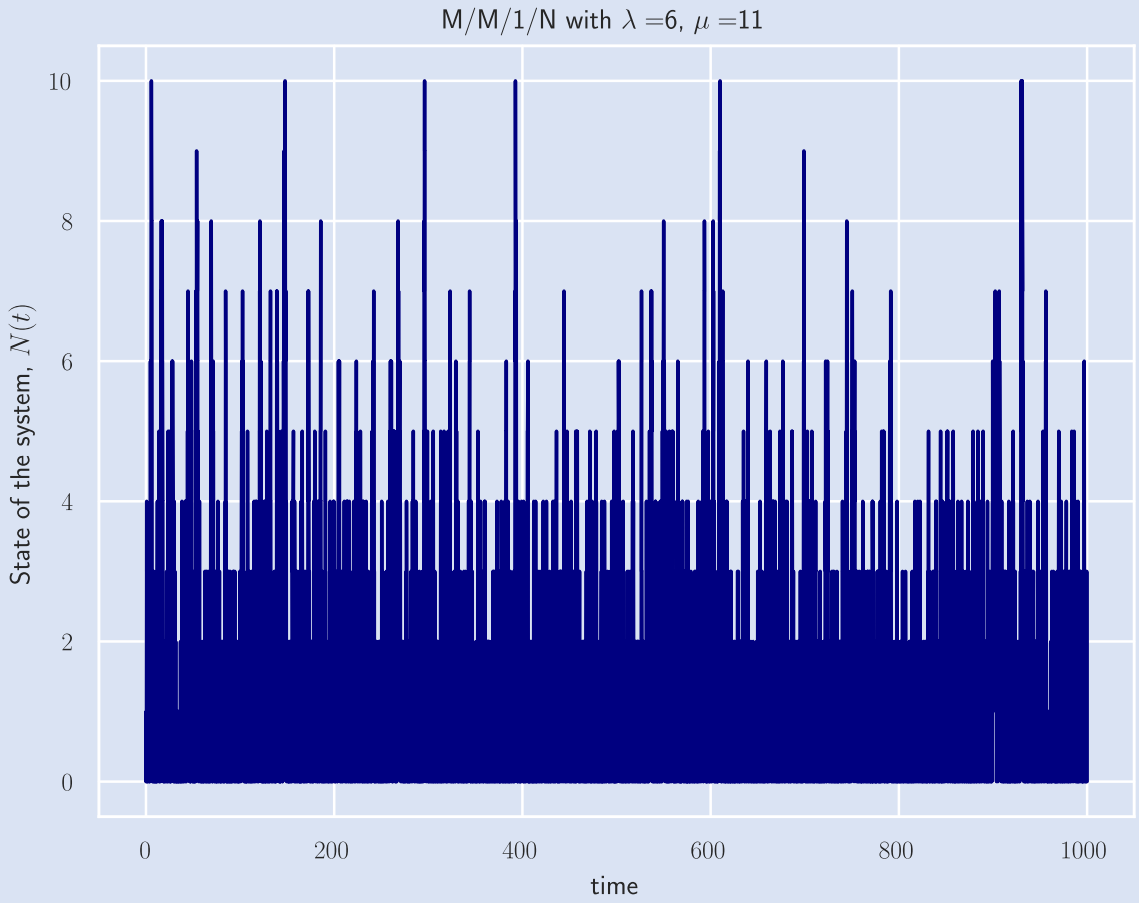
$$p_0 = \frac{1}{\sum_{i=0}^{N} \rho^i}$$

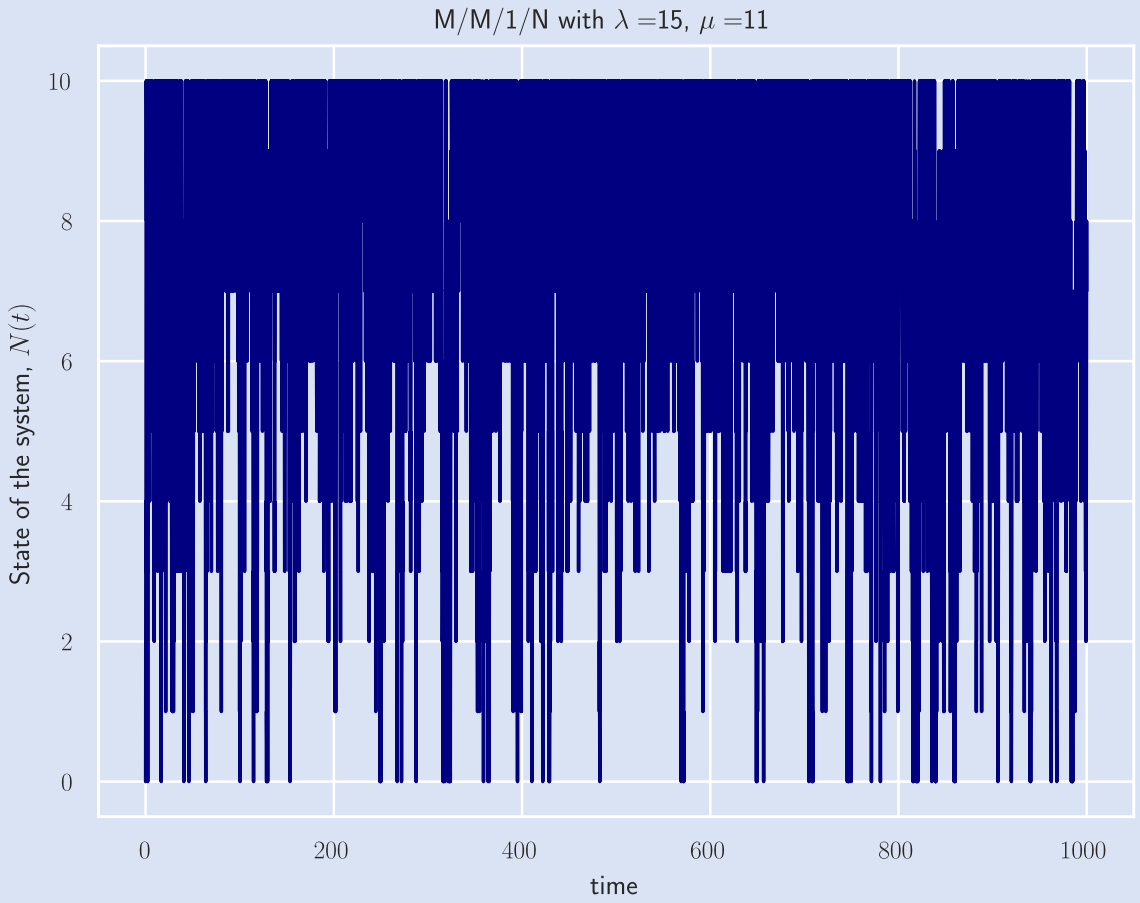
$$p_n = \left(\frac{1-\rho}{1-\rho^{N+1}}\right)\rho^n \qquad \ldots \; 0 \leq n \leq N.$$

- Note: Here, λ need not be greater than μ. Why?
  - Finite buffer size $\Longrightarrow$ Finite max queue length.

M/M/1/N - Simulations

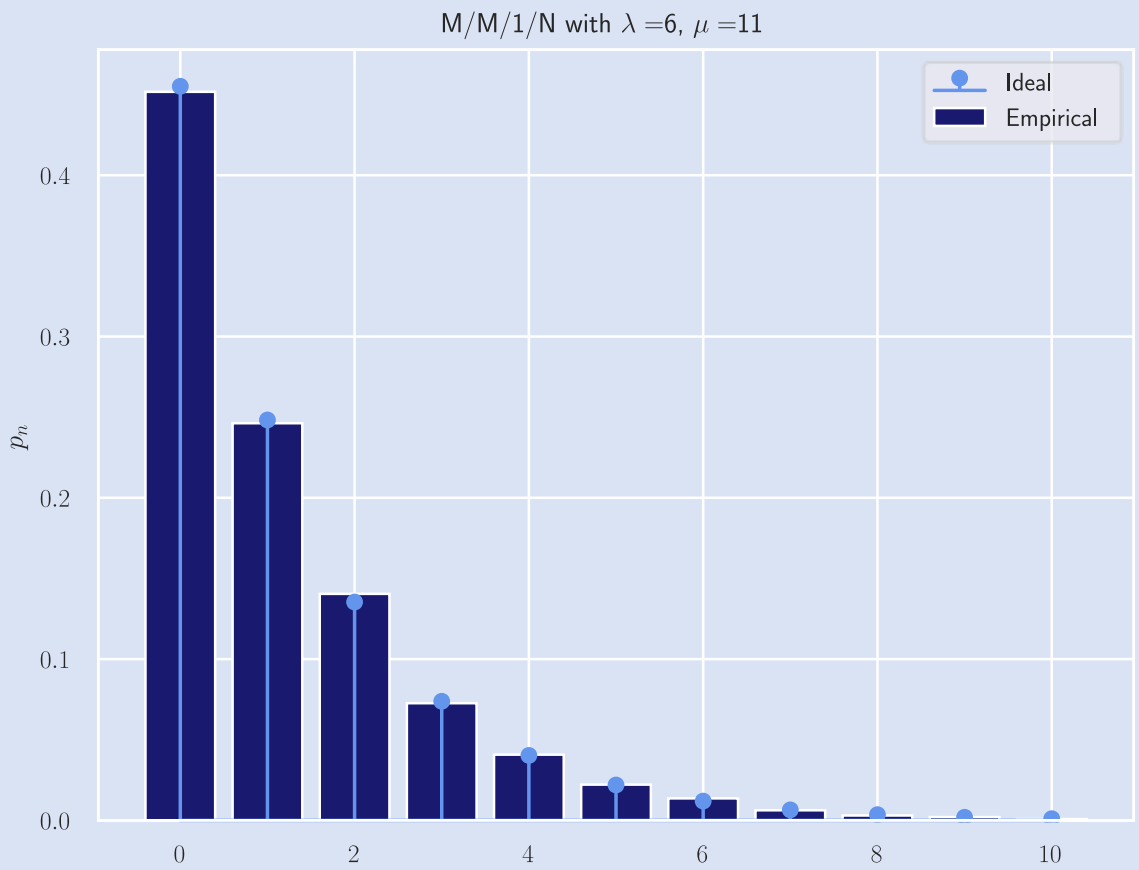
- Evolution of the queue with $\lambda \in \{6, 15\}, \mu = 11$.

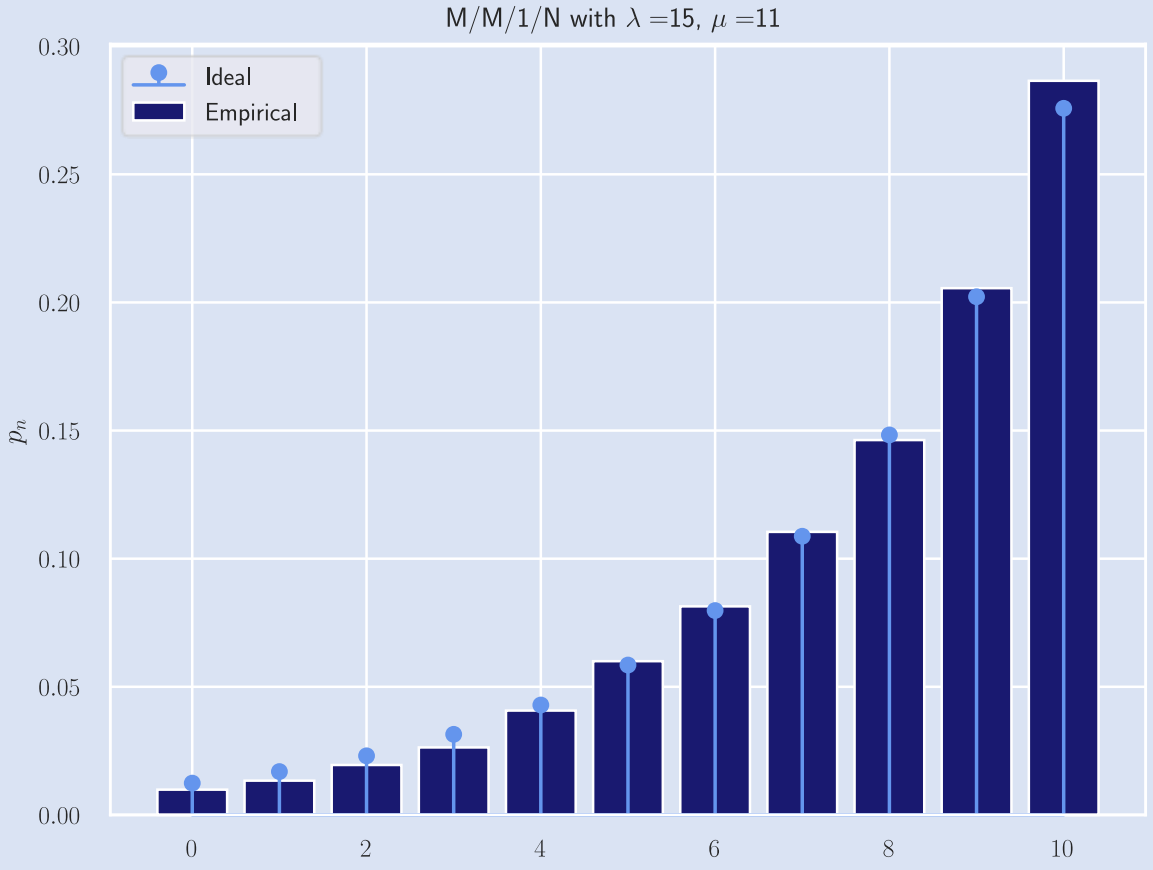




M/M/1/N – Simulations

- Stationary State Distribution when $\lambda \in \{6, 15\}, \mu = 11$.





M/M/1/N - Insights

- Blocking Probability:

  The blocking probability is the probability that the queue buffer is full, i.e., $P_N(t)$. Using the derived equations, Blocking Probability is $\frac{(1-\rho)}{(1-\rho^{N+1})}\rho^N$ when the system is in equilibrium.

- Increases with ρ, decreases N. See table.
- Average number of customers blocked per unit time
  - $N_{block} = \lambda P_N$

- What happens if $\rho = 1$?, $N \to \infty$?
  - L'Hospital's rule:
    - $\lim_{\rho \to 1} p_n = \frac{1}{N+1}$
  - When N $\to \infty$,
    - M/M/1/N $\to$ M/M/1

λ/μ	$P_{blocking}$	N	$P_{blocking}$
0.10	0.000009	1	0.474
0.5	0.016	2	0.299
0.75	0.072	3	0.212
1.00	0.166	5	0.126
2.00	0.508	20	0.014
5.00	0.800	100	0.0000027

Source: [2]

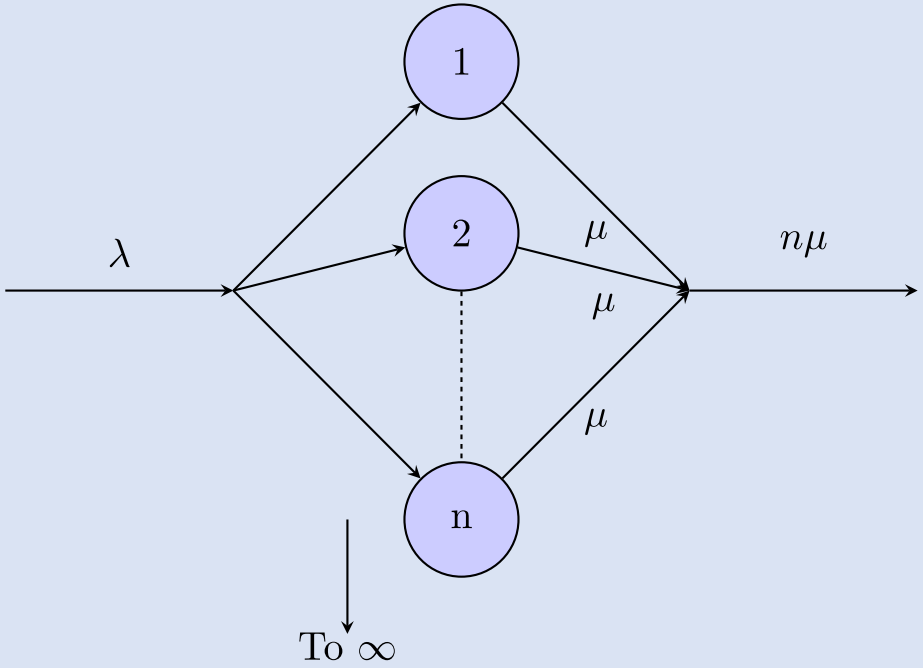
M / M / ∞ - Infinite servers

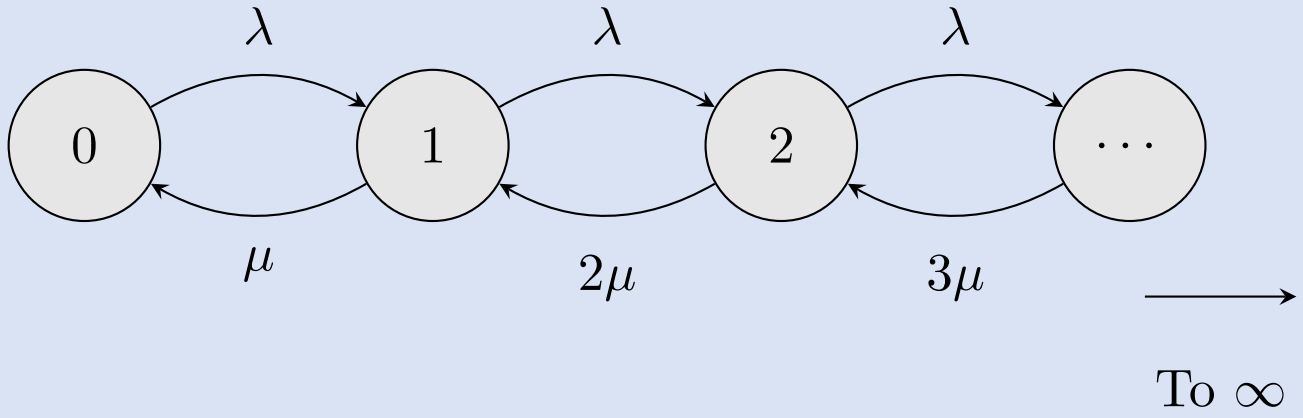
- Model:
  1. Most of the assumptions carry over.
  2. Now there are ∞ servers, no need to wait!
  3. No limit on queue length.

- Dream scenario
  - The supermarket billing section.
  - Job queues at a workstation server/ mainframe/ HPC.

Source: DALL · E

M / M / ∞ - Summary



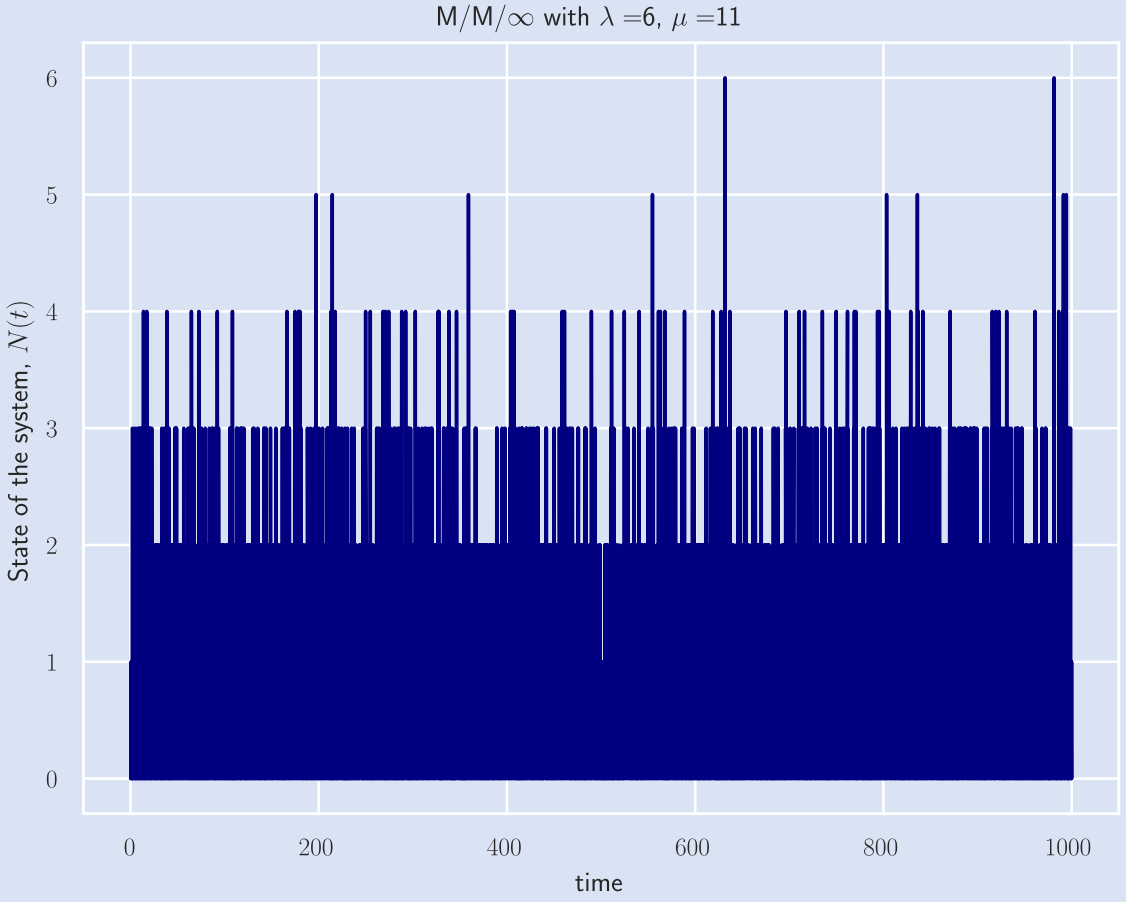


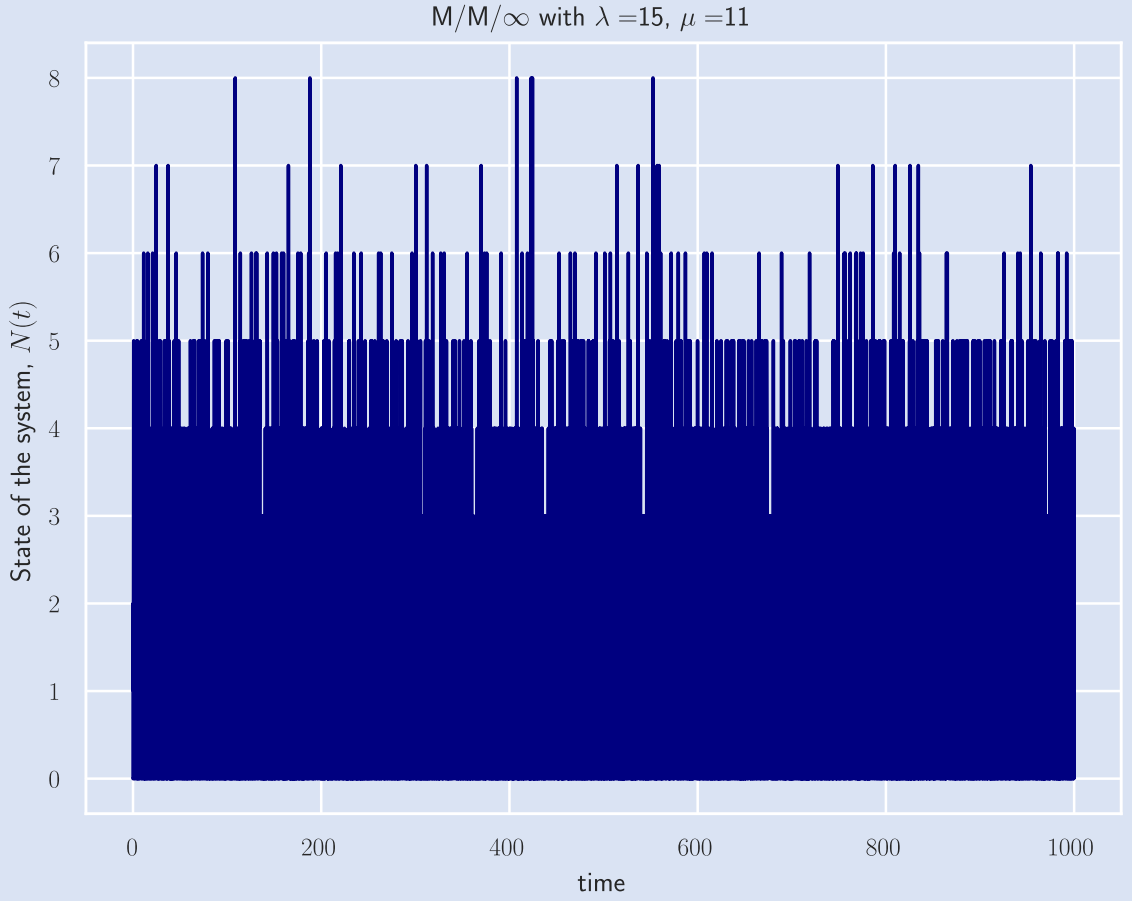
- In equilibrium,

$$p_0 = e^{-\lambda/\mu}$$
$$p_n = \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n e^{-\lambda/\mu} \quad \Longrightarrow \quad p_n = \frac{\rho^n}{n!}e^{-\rho}$$
$$N \sim \text{Poisson}(\rho)$$

M / M / ∞ - Simulations

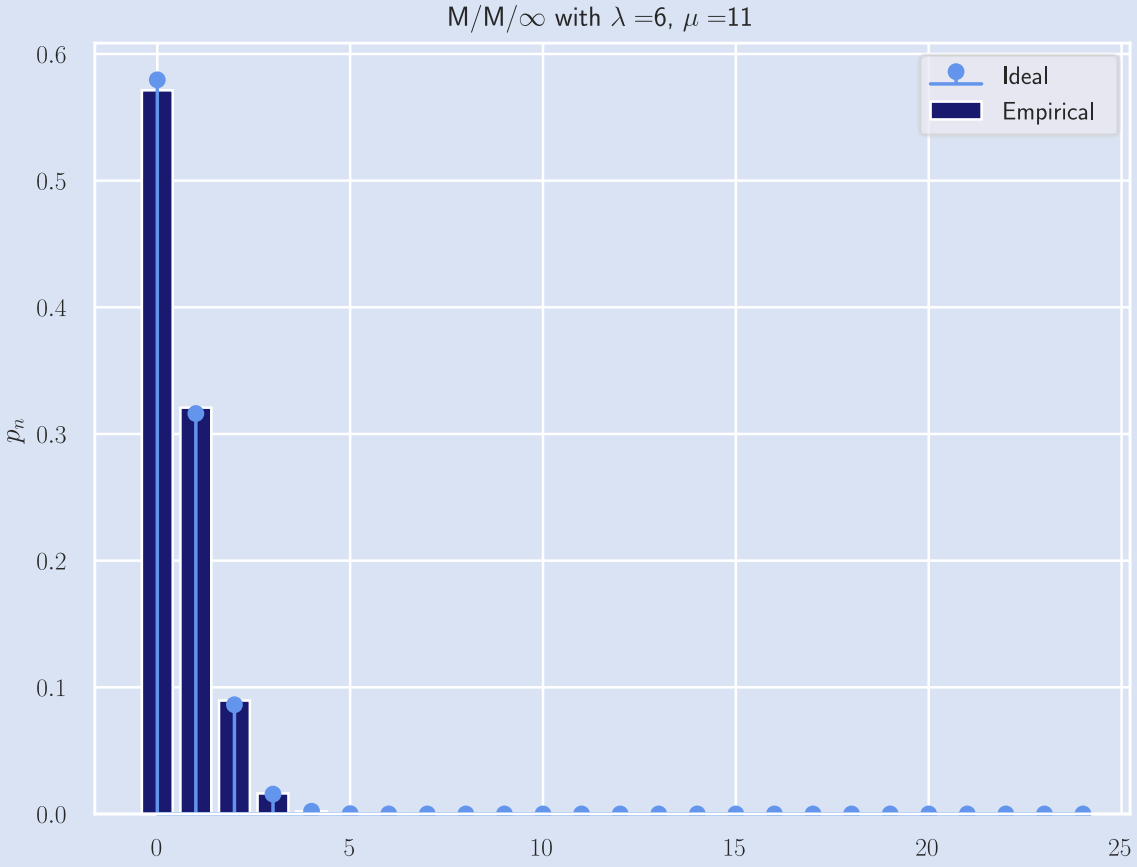
- Evolution of state with $\lambda \in \{6, 15\}, \mu = 11$.

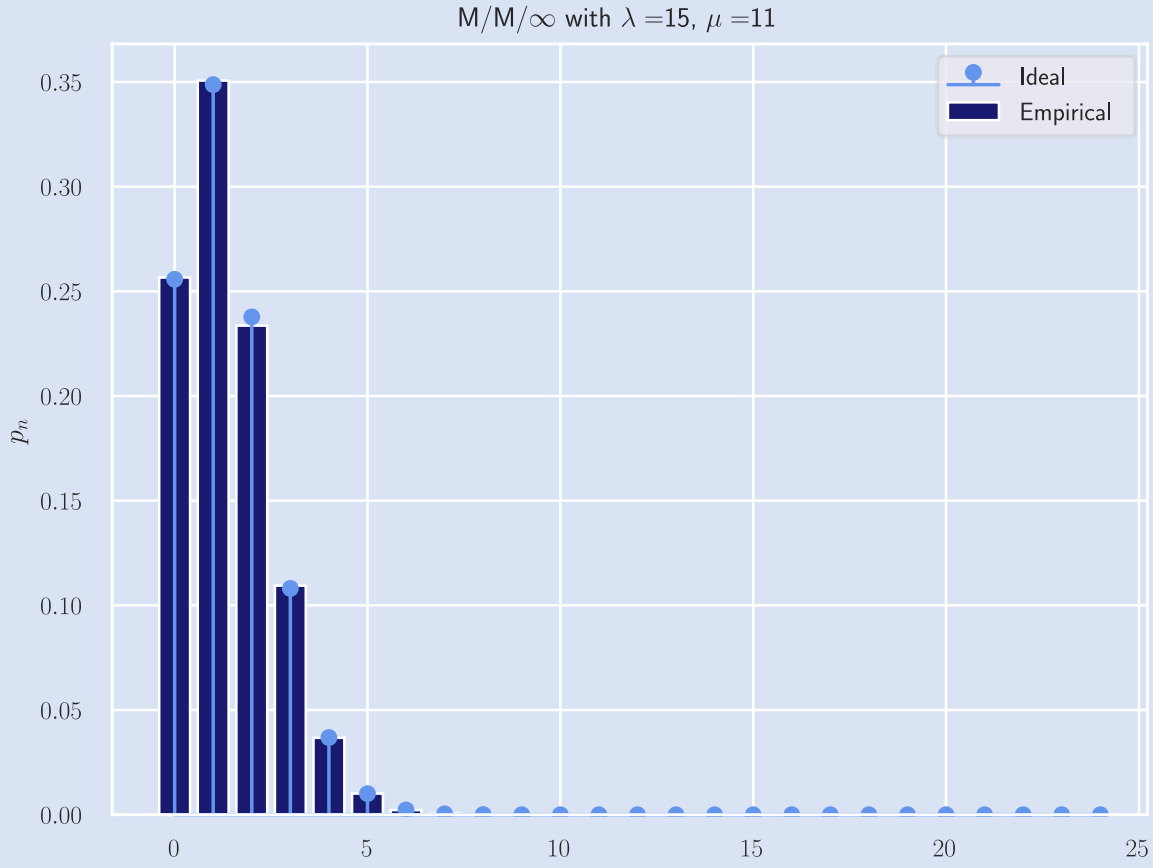




M / M / ∞ - Simulations

- Stationary State Distributions with $\lambda \in \{6, 15\}, \mu = 11$.



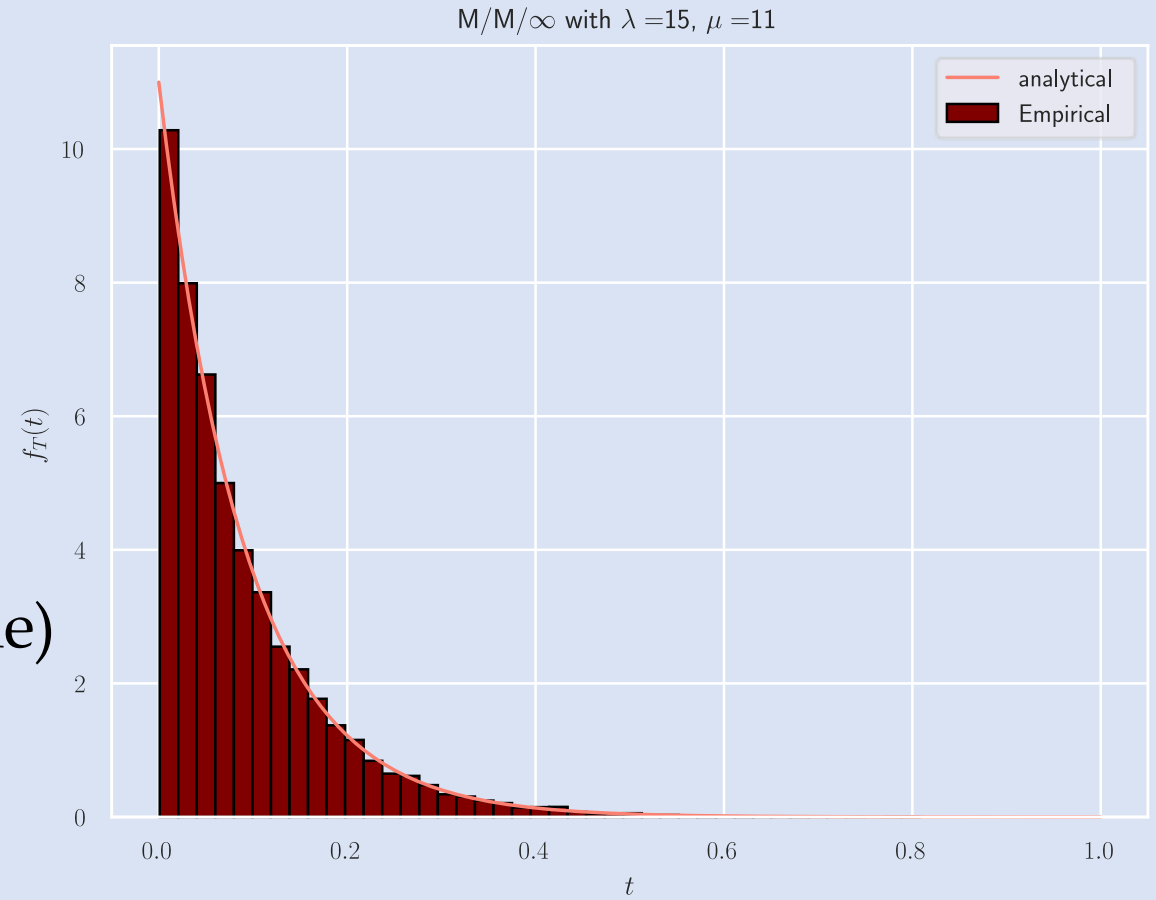


M / M / ∞ - Simulations

- Timing statistics
  - Response time = Service Time + Waiting Time.
  - But no need to wait
  - Response time → Service time
  - Mean = $1/\mu$.
  - Is it so?

- *Little's Law:*
  - Mean number of customers =
  - Rate of arrivals (arrivals per unit time)

  × Average response(=service) time
  - $\rho = \frac{\lambda}{\mu}$

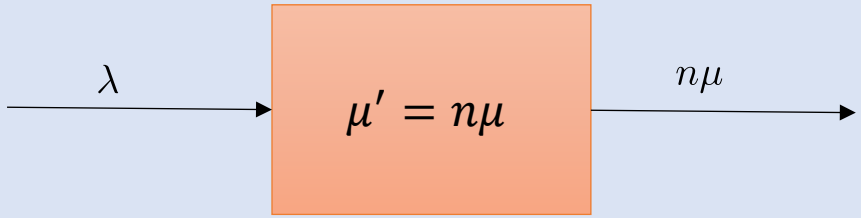


M / M / ∞ - Insights

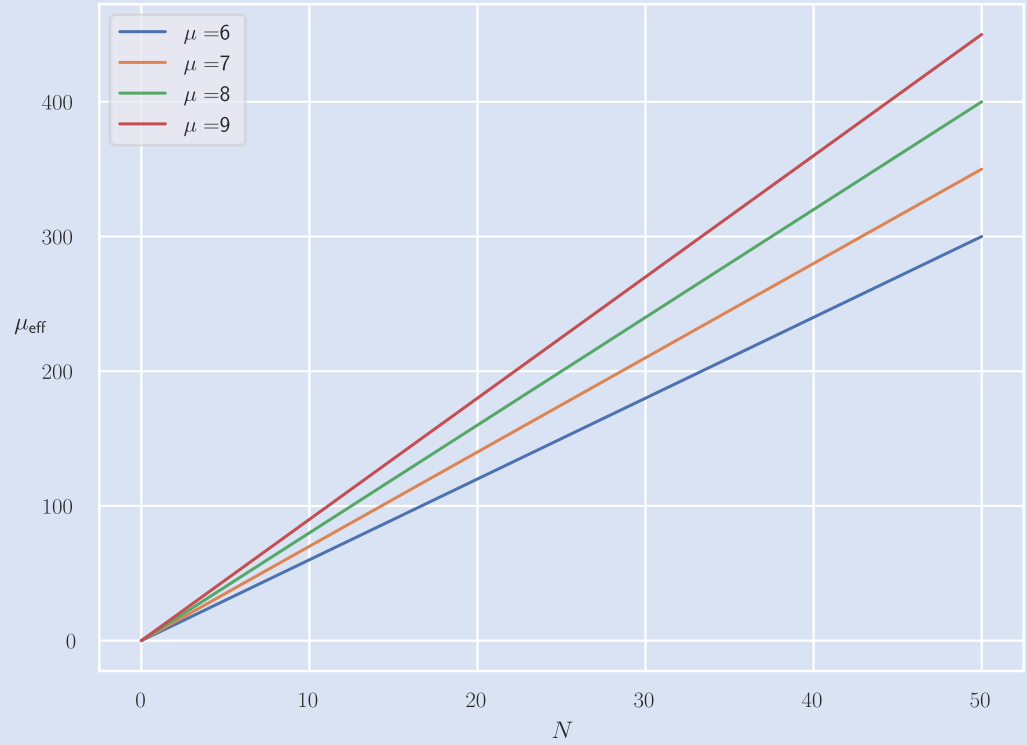
- Infinite number of servers.
  - $\implies \lambda \geq \mu$ is possible.
- State statistics
  - $\mathbb{E}[N] = \text{var}(N) = \rho$

- Alternative Interpretation
  - Responsive Server

- Blocking Probability is zero.
- Applications:
  - Large cloud computing facilities.
  - Large call centers.
  - Self-service type situations



Responsive Server



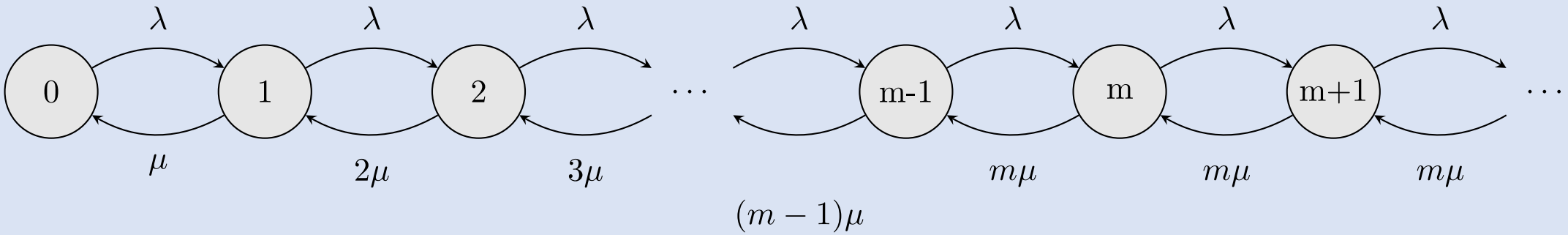
M / M / m – Multi-servers (finite)

- Model:
  - Same statistics for arrivals and departures.
  - But fall back into reality.
    - Finite servers
  - Queue size has no limits.

- More realistic assumptions than previous model(s).

Source: DALL · E

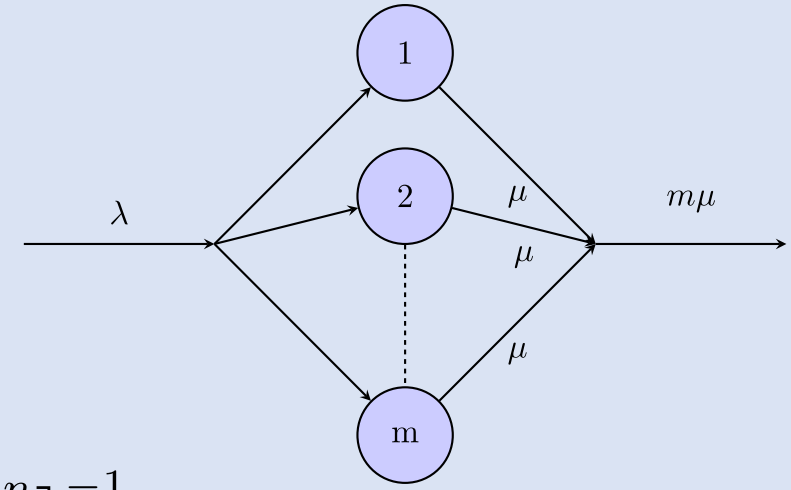
M / M / m – Summary



- In equilibrium,

$$p_n = \frac{\lambda^n}{n!\mu^n} p_0 \qquad \text{if } 1 \le n < m$$

$$= \frac{\lambda^n}{m^{n-m} m!\mu^n} p_0 \qquad \text{if } n \ge m.$$

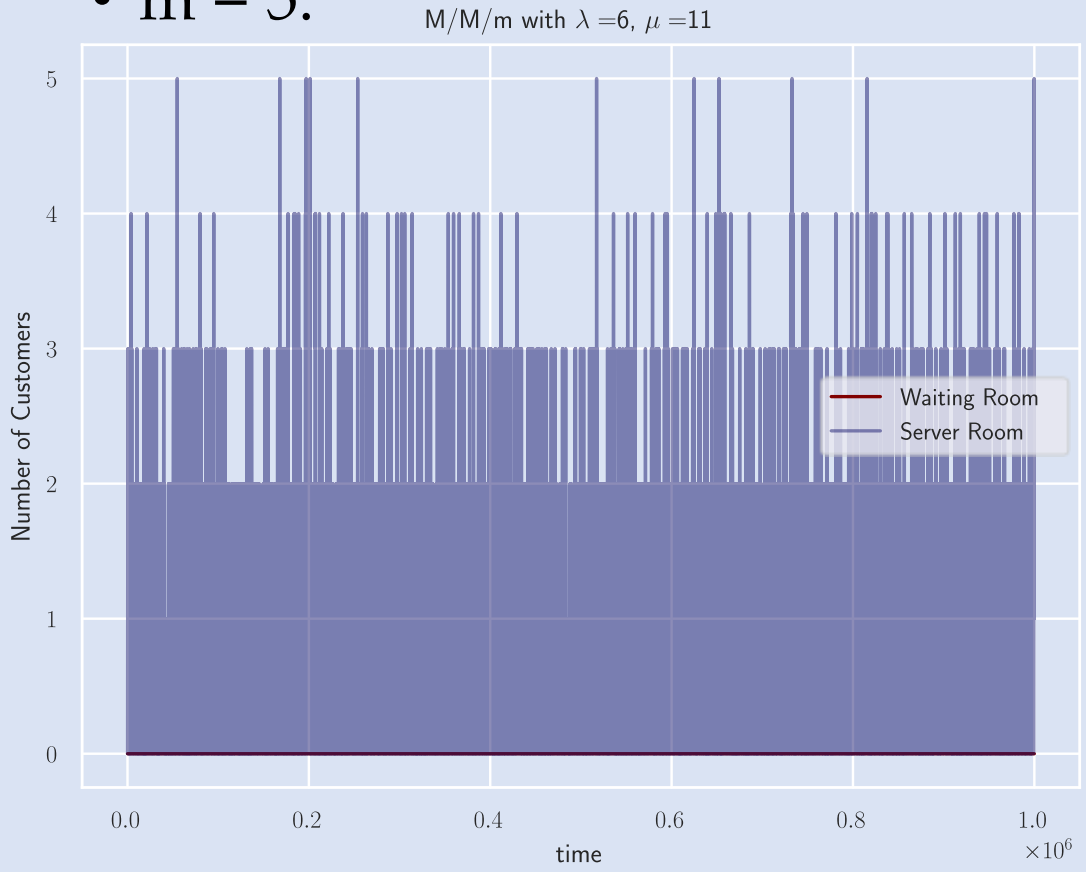


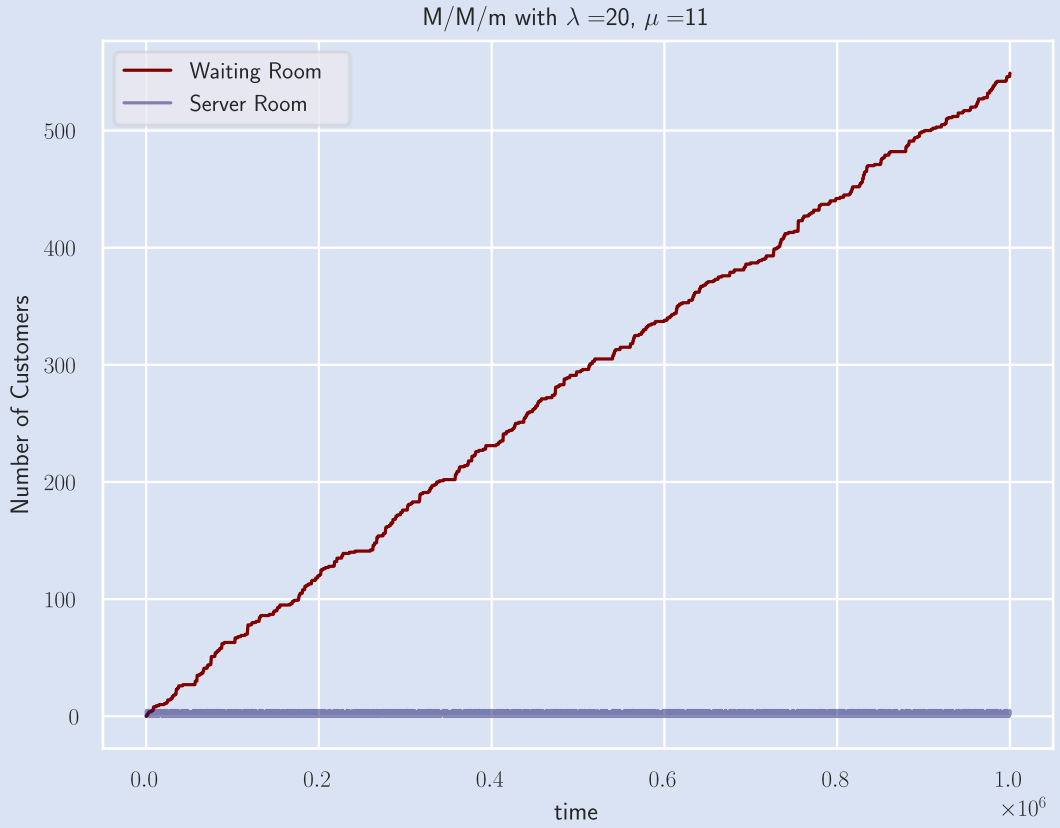
where,

$$p_0 = \left[1 + \sum_{n=1}^{m-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n + \sum_{n=m}^{\infty} \frac{1}{m^{n-m}} \frac{1}{m!} \left(\frac{\lambda}{\mu}\right)^n\right]^{-1}$$

M / M / m - Simulations

- Let's see some other plots
  - Visualize the state of the 'waiting room' and 'server room'
  - m = 5.





M / M / m - Insights

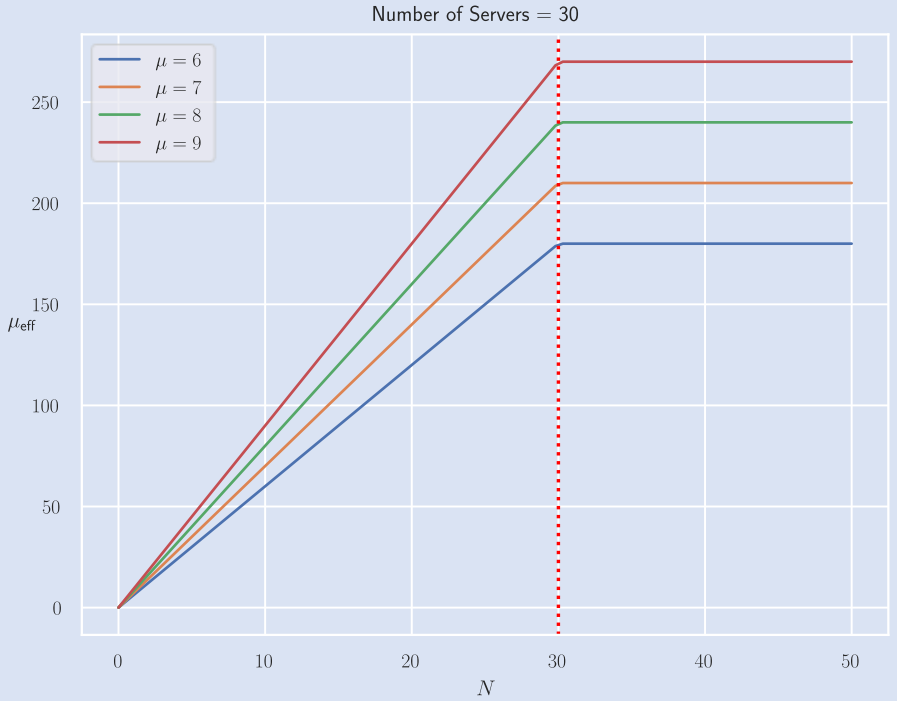
- This and previous model
  - *Load-dependent servers*
  - $\mu_{\text{eff}} = \mu \times \min\{N, m\}$
  - N = number of customers

- Queueing Probability
  - *Erlang – C formula*

$$\rho = \frac{\lambda}{m\mu}$$

$$\mathbb{P}(\text{queueing}) = \frac{1}{m!}\left(\frac{\lambda}{\mu}\right)^m \frac{1}{1-\rho} p_0$$

  - Widely used in telecommunication



λ	$P_{queueing}$
5	0.00008
10	0.0032
15	0.027
20	0.130
25	0.463

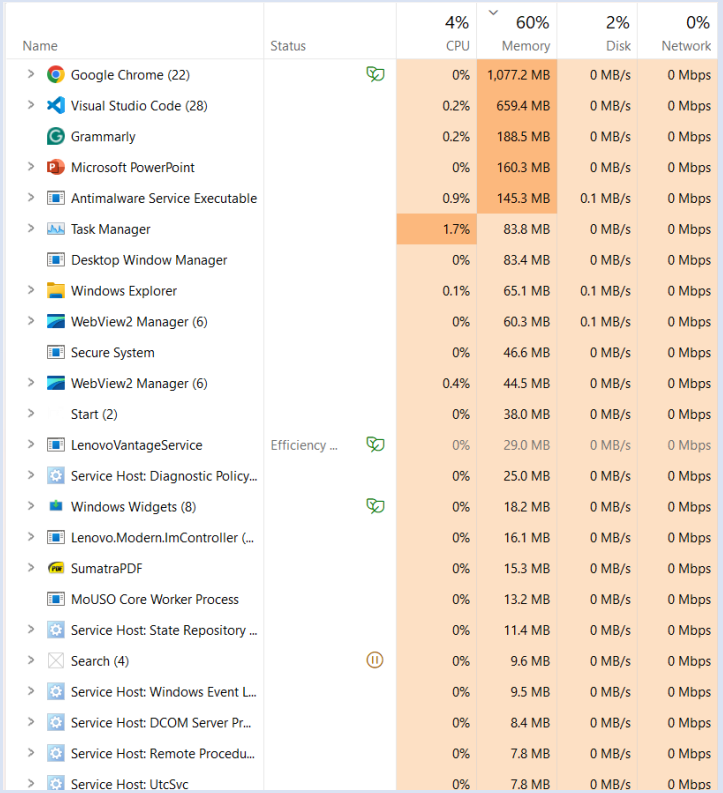
Other models

- So far, customers' demands were *i.i.d.*, we considered single class of customers.
  - Multi-class arrivals $\Longrightarrow$ Variable statistics.
- More general and flexible models
  - M/G/1
  - G/G/1 and so on.
- Complexity of models increases
  - Difficult to get analytical results.
  - Lack of meaningful insights.
  - Trade off:

    Accuracy of model v/s Mathematical tractability, Valuable insights.

Applications

- Intuitive ones:
  - Previously mentioned: Banks, Hospitals, Supermarkets, etc.
  - Fast food chains
  - Ticket vendors (~fixed service time $\Longrightarrow$ M/D/m).
- Non-intuitive ones:
  - Communications:
    - Managing and routing of different packets.
  - Different Operating System processes, programs
    - Single-core CPU – M/M/1
    - Multi-core CPU – M/M/m
  - Interrupt Handling in embedded systems
    - Single (Multi)- core MCU – M/M/1(m)/LIFOPR



Critical Application – A case study

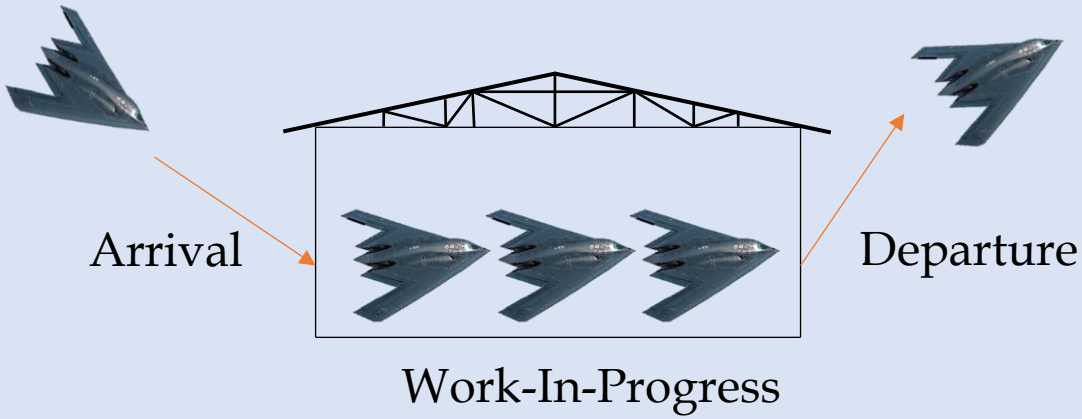
- US Military – Scheduling B-2 Bomber maintenance
  - Limited Resource – Only 20 of B2s.
  - Required at a moment's notice.
    - FHP, Wartime posture
  - Frequent maintenance period of 18–45 days!
- Issue: Unpredictable scheduling, downtime.
  - Due to asymmetric usage
- Result:
  - Decreased aircraft availability (AA).
  - ~4.75 in heavy maintenance, ~3 in light maintenance.
  - AA = ~60% of total capacity
  - Unacceptable.

Northrop B-2 Spirit

Source: https://en.wikipedia.org/wiki/Northrop_B-2_Spirit#/media/File:B-2_Spirits_on_Deployment_to_Indo-Asia-Pacific.jpg

Critical Application – A case study

- Acceptable AA = 80-85%
- Model: Queue at the maintenance hangar.
- Data analysis observations:
  - ~3 B2s are in maintenance at a time ($\bar{N}$).
  - Every ~7 days (λ), one of the B2s goes into maintenance .



- Queueing theory analysis:
  - Little's Law:

Response time = $\bar{T} = \frac{\bar{N}}{\lambda} = \frac{3}{\frac{1}{7}} = 21$ days.

❖ Conclusion

The heavy LO maintenance routine must have a lead time of ~21 days to maintain acceptable AA.

References

1. H. Pishro-Nik. *Introduction to Probability, Statistics, and Random Processes*. Kappa Research,LLC, 2014.
2. Thomas G. Robertazzi. *Computer Networks and Systems: Queueing Theory and Performance Evaluation*. Springer-Verlag, Berlin, Heidelberg, 3rd edition, 2000.
3. William J. Stewart. Probability, *Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton University Press, 2009.
4. https://www.isixsigma.com/case-studies/u-s-a-f-uses-continuous-process-improvement-on-the-b-2-bomber-part-1/

Thank You

Rishabh Pomaje

&

Samyak Sanjay Parakh