Single queue modeling

Basic definitions for performance predictions

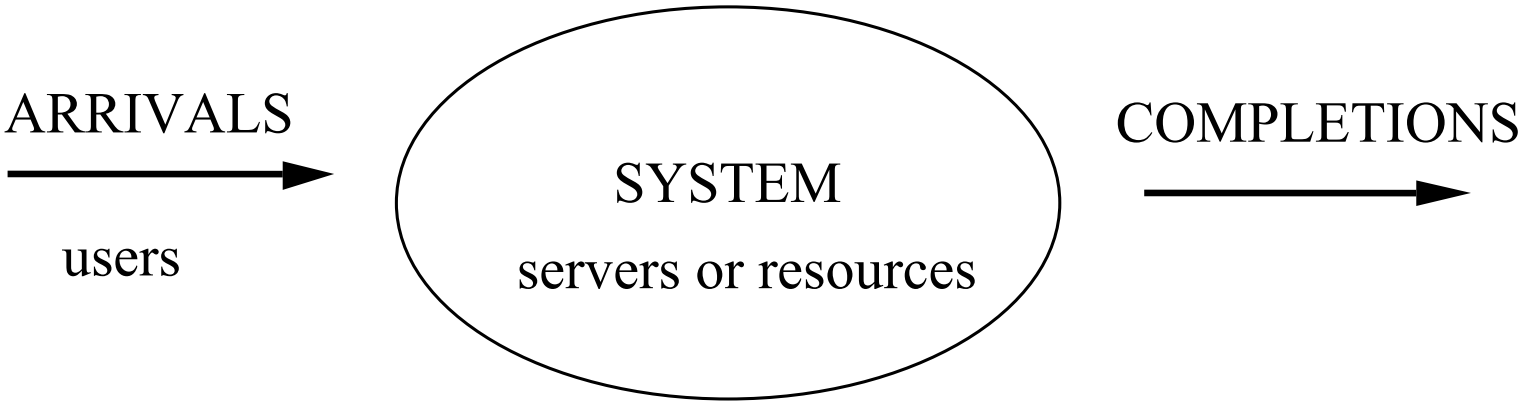
The performance of a system that gives services could be seen from two different viewpoints:

- **user**: the time to obtain a service or the waiting time before getting a service;

- **system**: the number of users served in the unit time or the resources utilization level.

Basic definitions for performance predictions

The model can be built analyzing the behaviour of the system.



Arrival rate and throughput

Principal measures:

- **T**: system observation interval;
- **A**: number of arrivals in the period T;
- **C**: number of completions in the period T
- **B**: busy period time in the period T

Derived quantities:

- λ: **arrival rate**, $\lambda = A/T$;
- X: **throughput**, $X = C/T$

Utilization factor

For a **single server** an interesting measure is also the period the server is busy (**B**).

Then we can obtain:

- **ρ**: server **utilization**, $\rho = B/T$;

- **S**: average **service completion per request**, B/C.

Now it is possible to get the **utilization law**:

$$\rho = B/T = S\,C/T = X\,S$$

a special case of the ***Little's law***

Little's law

Denoting with:

- **W** : the time spent in system by the all requests in T
- **N**: the **average number of requests** in the system,

$$N = W/T$$

- **R**: the **average** system **residence time** per request (average response time),

$$R=W/C$$

the **Little's law** is equal to: **N =X R**

given that N = W/T= R C/T

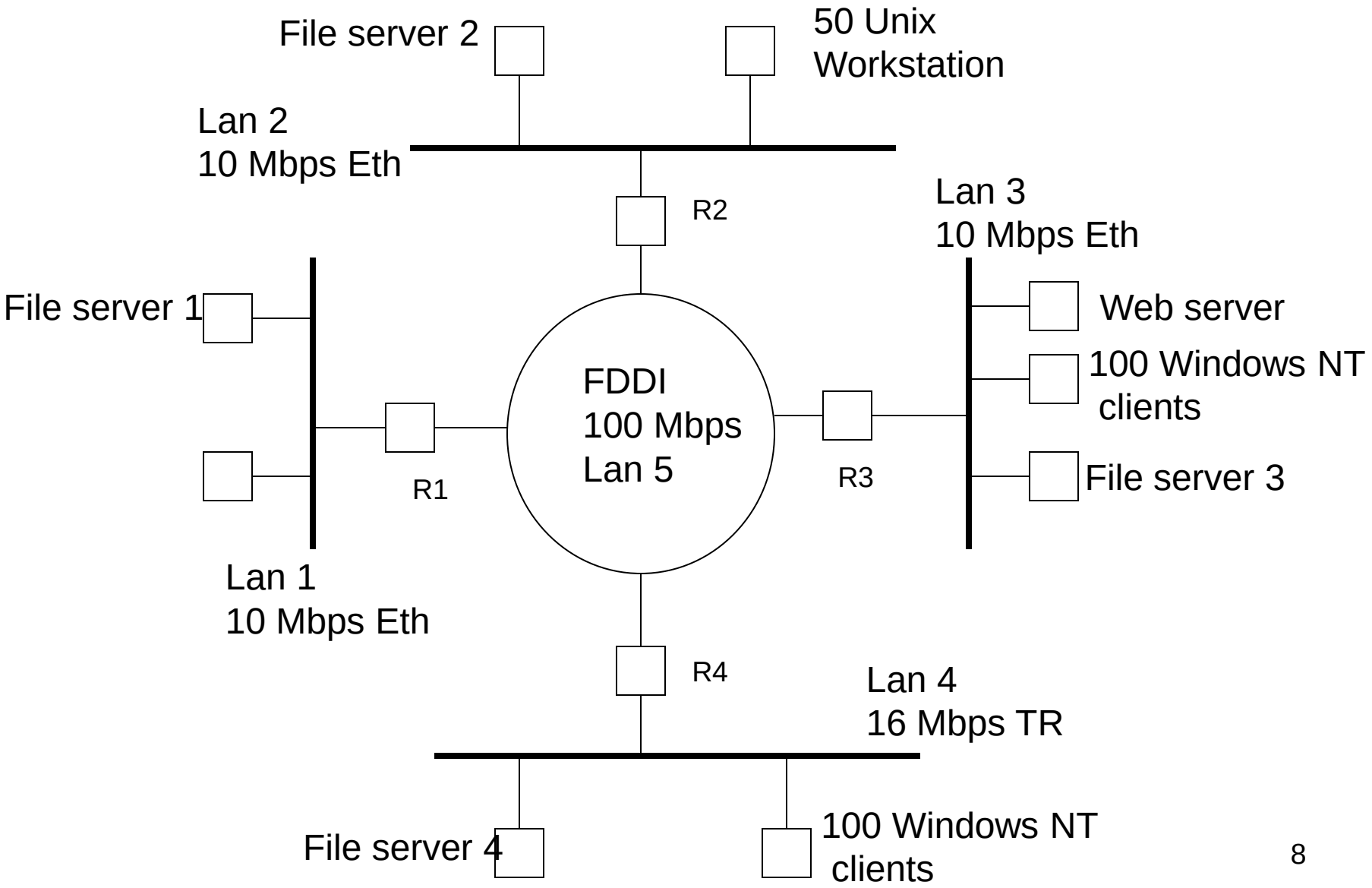
Little's law is very important in the performance evaluation of system consisting of a set of servers because it is widely applicable and does not require strong assumptions.

Queuing systems

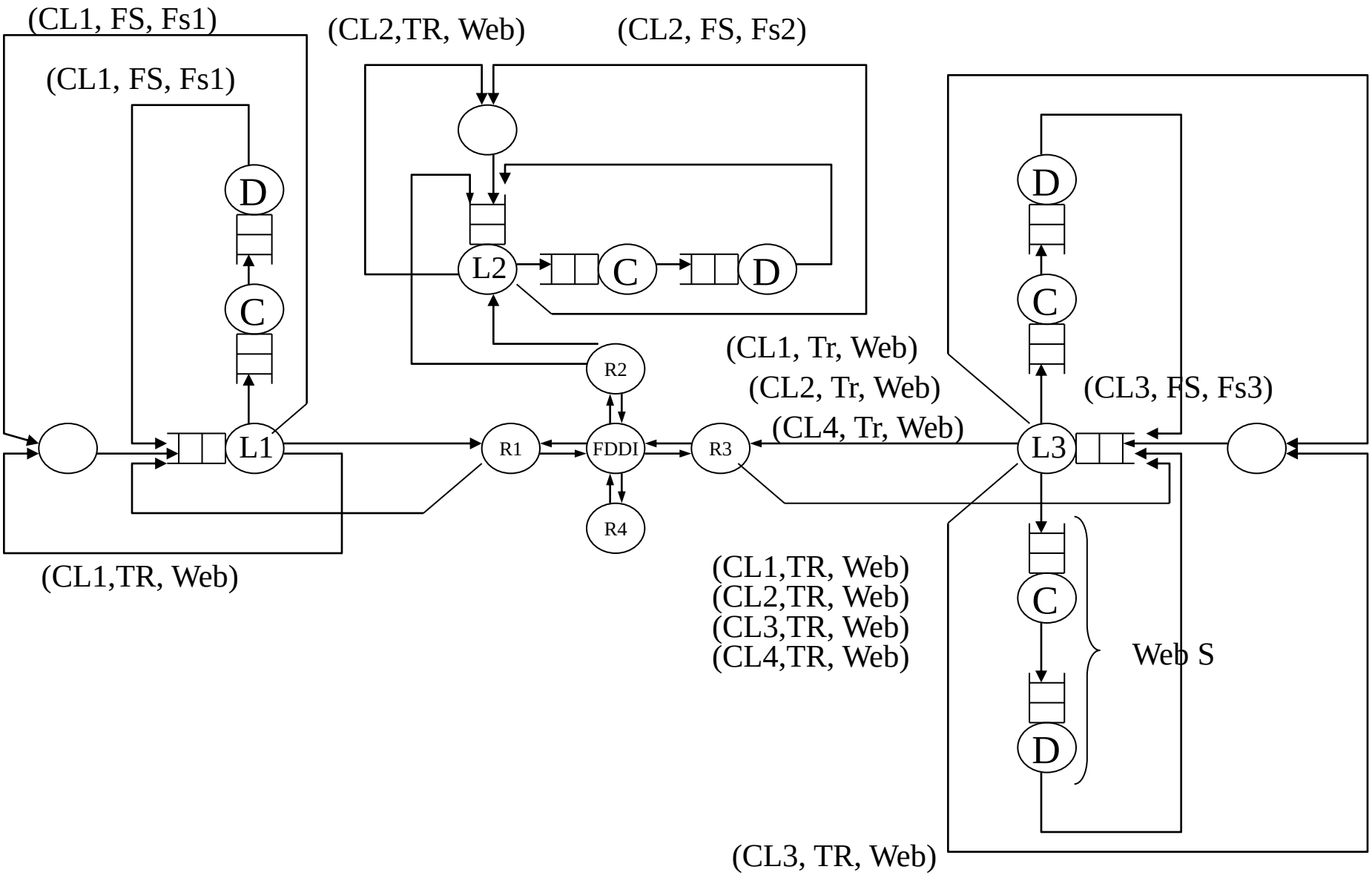
A system can be modelled by a set of interconnected queues.

The servers and the users have different characteristics/behaviors in each queue. If the queues have an ***independent*** each other behaviour then each queue can be **analysed separately**.

In particular conditions each queue can be modelled as a Markov chain.


File server 2
50 Unix
Workstation
Lan 2
10 Mbps Eth
R2
Lan 3
10 Mbps Eth
File server 1
Web server
100 Windows NT
clients
FDDI
100 Mbps
Lan 5
R1
R3
File server 3
Lan 1
10 Mbps Eth
R4
Lan 4
16 Mbps TR
File server 4
100 Windows NT
clients

A queuing network model



Queuing systems

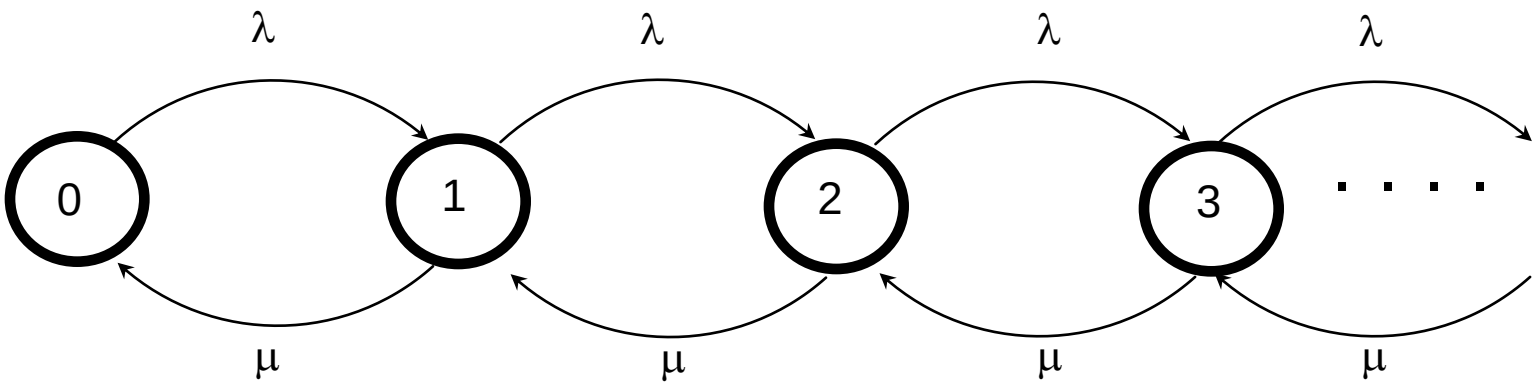
For example a ***birth/dead*** process can be used to represent a particular queue in which:

- the inter arrival time of the users is distributed in accordance to the exponential distribution, with arrival rate equal to **λ**;
- the presence of only **one** server;
- the service time is distributed in accordance to the exponential distribution, with service time equal to **1/μ**.

This kind of queue is denoted as **M/M/1**

M/M/1 queue analysis

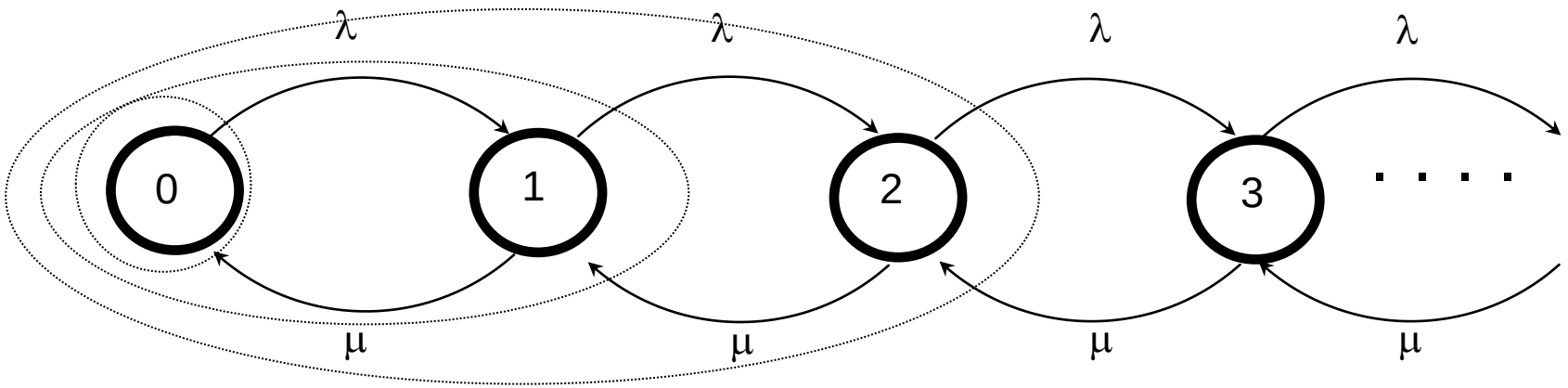
Infinite queue



State transition diagram – infinite population

M/M/1 queue analysis

Infinite queue



State transition diagram **with boundaries**

M/M/1 queue analysis

At every state, in a balanced condition the incoming flow is equal to the outgoing flow.

$$flow - in = flow - out$$

$$\lambda p_0 = \mu p_1$$

$$\lambda p_1 + \mu p_1 = \lambda p_0 + \mu p_2 ----> \lambda p_1 = \mu p_2$$

.

.

.

$$\mu p_k = \lambda p_{k-1}$$

M/M/1 queue analysis

At every state, in a balanced condition the incoming flow is equal to the outgoing flow.

$$flow - in = flow - out$$

$$\mu p_1 = \lambda p_0$$

$$\mu p_2 = \lambda p_1$$

.

.

.

$$\mu p_k = \lambda p_{k-1}$$

M/M/1 queue analysis

Solving the system:

$$p_k = \frac{\lambda}{\mu} p_{k-1} = \frac{\lambda}{\mu}\left(\frac{\lambda}{\mu} p_{k-2}\right) = \ldots = p_0\left(\frac{\lambda}{\mu}\right)^k, k = 1,2,\ldots$$

The sum of the time fractions the system being in all possible states, from 0 to ∞, equals to 1 (probability):

$$p_0 + p_1 + p_2 + \ldots + p_k + \ldots = \sum_{k=0}^{\infty} p_k = \sum_{k=0}^{\infty} p_0\left(\frac{\lambda}{\mu}\right)^k = 1$$

M/M/1 queue analysis

$$p_0 + p_1 + p_2 + \ldots + p_k + \ldots = \sum_{k=0}^{\infty} p_k = \sum_{k=0}^{\infty} p_0 \left(\frac{\lambda}{\mu}\right)^k = 1$$

we obtain

↓

$$p_0 = \left[\sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k\right]^{-1} = 1 - \frac{\lambda}{\mu}$$

M/M/1 queue analysis

It is now possible to calculate the following quantities:

Probability to stay in state k (fraction of time server has k requests):

$$p_k = p_0 \rho^k \quad \text{where} \quad \rho = \frac{\lambda}{\mu}$$

$$\text{then} \quad p_0 = \left[\sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu} \right)^k \right]^{-1} = 1 - \frac{\lambda}{\mu} = 1 - U$$

M/M/1 queue analysis

The utilization factor:

$$U = \rho = 1 - p_0 = \frac{\lambda}{\mu}$$

The expected number of users in the system

Using the “average” definition,

$$\overline{N} = \sum_{k=0}^{\infty} k * p_k = \sum_{k=0}^{\infty} k * (1-U)U^k = (1-U)\sum_{k=0}^{\infty} k * U^k$$

The last sum is equals to $U/(1-U)^2$ for $U<1$,

$$\overline{N} = U/(1-U)$$

M/M/1 queue analysis

The average response time

Using the **Little's Law** (**the average throughput equals to λ**),

$$R = \overline{N} / X = (U / \lambda)(1-U) = (1/\mu)/(1-U) = S/(1-U)$$

where $S=1/\mu$ is the average service time of request at the server.

The expected number of users in the queue

$$E[L] = \sum_{k=1}^{\infty} (k-1) p_k = \frac{\rho^2}{1-\rho}$$

The average time in the queue and in the system

$$E[W] = \frac{E[L]}{\lambda} = \frac{\rho}{\mu(1-\rho)} \qquad E[T] = \frac{E[N]}{\lambda} = \frac{1}{\mu(1-\rho)}$$

Example (infinite queue)

Arrival rate at the Web server: λ**=30** requests/sec

Average service time: **0.02** seconds

Solution

Average service rate $\rightarrow \mu = 1/0.02 = 50$ requests/sec

$U=\lambda/\mu \rightarrow U=30/50=0.6$ (60%)

$p_0=1-U=1-0.6=40\%$

Average number of requests at the server:

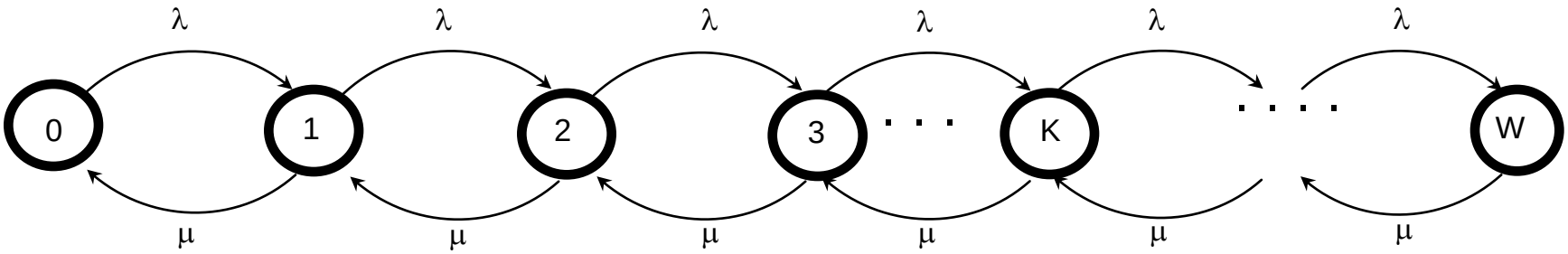
$$N=U/(1-U)=0.6/(1-0.6)=1.5$$

Average response time:

$$R=S/(1-U)=(1/\mu)/(1-U)=(1/50)/(1-0.6)=0.05\text{sec}$$

M/M/1 finite queue analysis

Finite queue



State transition diagram – infinite population/finite queue

M/M/1 finite queue analysis

Using the flow-in=flow-out equations,

$$p_k = p_0\left(\frac{\lambda}{\mu}\right)^k, k = 1,....W$$

We have a finite number of states,

$$p_0 + p_1 + \ldots + p_W = p_0 \sum_{k=0}^{W} p_0\left(\frac{\lambda}{\mu}\right)^k =$$

$$= p_0\left[\frac{1-(\lambda/\mu)^{W+1}}{1-\lambda/\mu}\right] = 1$$

M/M/1 finite queue analysis

Hence,

$$p_0 = \frac{1 - \lambda / \mu}{1 - (\lambda / \mu)^{W+1}}$$

The utilization factor:

U=1-p_0 $\rightarrow$

$$U = p_0 \frac{(\lambda / \mu)\left[1 - (\lambda / \mu)^{W}\right]}{1 - (\lambda / \mu)^{W+1}}$$

M/M/1 finite queue analysis

The expected number of users in the system

Using the “average” definition:

$$\overline{N} = \sum_{k=0}^{W} k * p_k = p_0 \sum_{k=0}^{W} k(\lambda/\mu)^k$$

$$\overline{N} = \frac{(\lambda/\mu)\left[W(\lambda/\mu)^{W+1} - (W+1)(\lambda/\mu)^W + 1\right]}{\left[1-(\lambda/\mu)^W\right](1-(\lambda/\mu))}$$

M/M/1 finite queue analysis

The average response time

Using the Little's Law and S=1/ μ:

$$R = \overline{N} / X = \frac{S\left[W(\lambda/\mu)^{W+1} - (W+1)(\lambda/\mu)^{W} + 1\right]}{\left[1-(\lambda/\mu)^{W}\right](1-(\lambda/\mu))}$$

M/M/1 finite queue analysis

Fraction of time server has k requests:

$$p_k = \begin{cases} \frac{1-\lambda/\mu}{1-(\lambda/\mu)^{W+1}} \left(\frac{\lambda}{\mu}\right)^k & k = 0, \cdots, W \quad \lambda \neq \mu \\ 1/(W+1) & k = 0, \cdots, W \quad \lambda = \mu \end{cases}$$

Server utilization:

$$U = \begin{cases} \frac{(\lambda/\mu)\ [1-(\lambda/\mu)^W]}{1-(\lambda/\mu)^{W+1}} & \lambda \neq \mu \\ W/(W+1) & \lambda = \mu \end{cases}$$

Average server throughput:

$$X = U \times \mu$$

Average number of requests in the server:

$$\overline{N} = \begin{cases} \frac{(\lambda/\mu)[W(\lambda/\mu)^{W+1} - (W+1)\ (\lambda/\mu)^W + 1]}{[1-(\lambda/\mu)^{W+1}](1-\lambda/\mu)} & \lambda \neq \mu \\ W/2 & \lambda = \mu \end{cases}$$

Average response time:

$$R = \overline{N}/X$$

Example (finite queue)

Arrival rate at the Web server: λ**=30** requests/sec

Average service time: **0.02** seconds

Identify the minimum queue length so that less than 1% of requests are rejected

Solution

Average service rate $\rightarrow \mu = 1/0.02 = 50$ request/sec

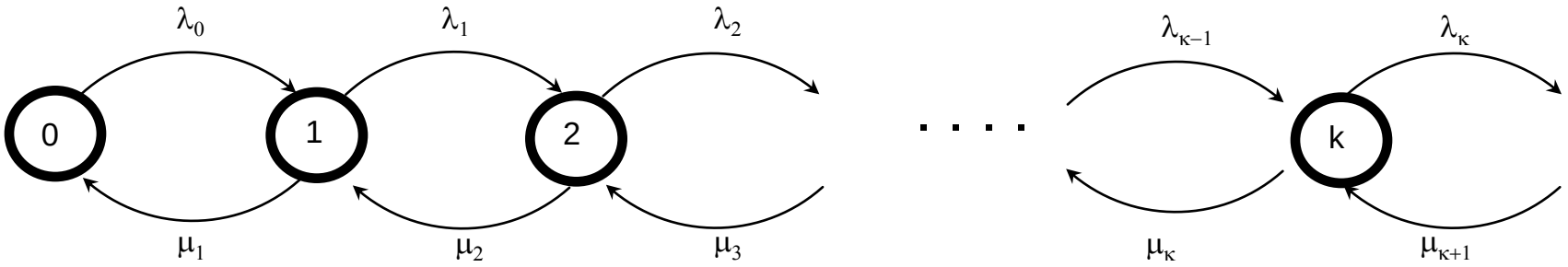
$U=\lambda/\mu \rightarrow U=30/50=0.6$ (60%)

$\mathbf{p_0}=0.4/(1-0.6^{W+1})$

$p_W=p_0(\lambda/\mu)^W<0.01 \rightarrow 0.4* 0.6^W/(1-0.6^{W+1})<0.01 \rightarrow$

W≥8

Generalized System-Level Models



Using the flow-in=flow-out equations,

$$\lambda_{k-1} p_{k-1} = \mu_k p_k, k = 1,2,...$$

Generalized System-Level Models

By applying recursively,

$$p_1 = \frac{\lambda_0}{\mu_1} p_0$$

$$p_2 = \frac{\lambda_1}{\mu_2} p_1 = \frac{\lambda_1}{\mu_2} \frac{\lambda_0}{\mu_1} p_0$$

.

.

.

$$p_k = \frac{\lambda_{k-1}}{\mu_k} p_{k-1} = \frac{\lambda_{k-1}}{\mu_k} \cdots \frac{\lambda_1}{\mu_2} \frac{\lambda_0}{\mu_1} p_0$$

Generalized System-Level Models

So,

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}$$

Since

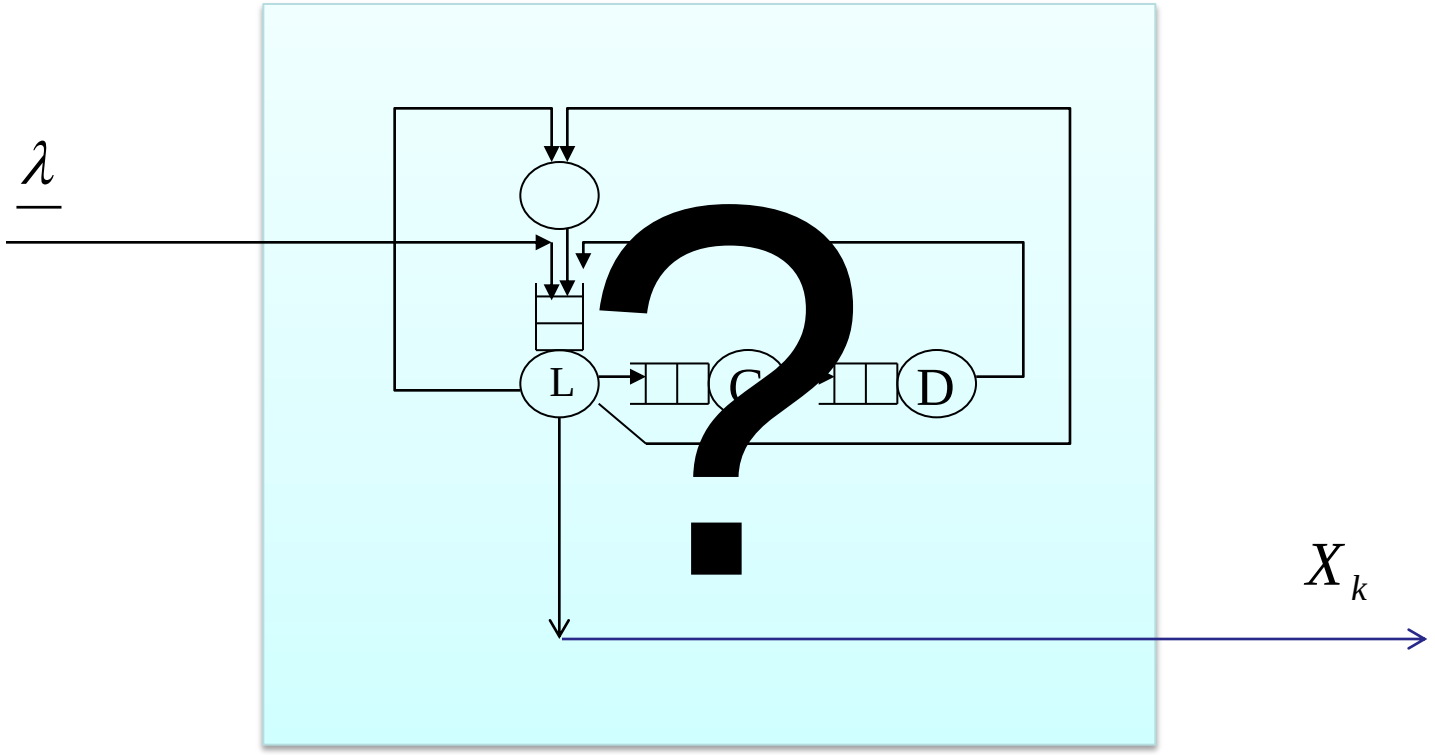
$$\sum_{k=0}^{\infty} p_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} = 1$$

Generalized System-Level Models

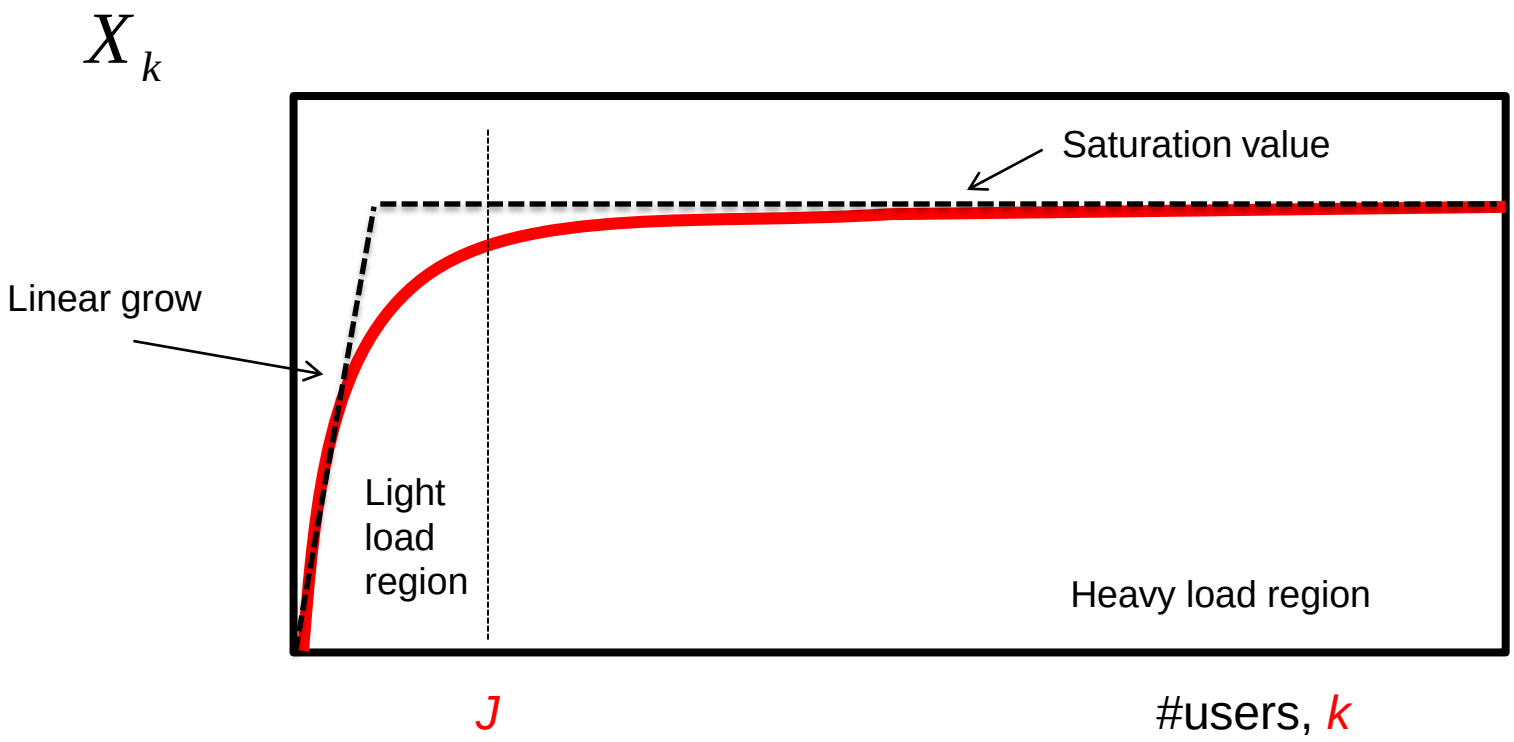
this implies that

$$p_0 = \left[\sum_{k=0}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right]^{-1}$$

OPEN generalized system model with constant arrival rate and variable service rate
page 330

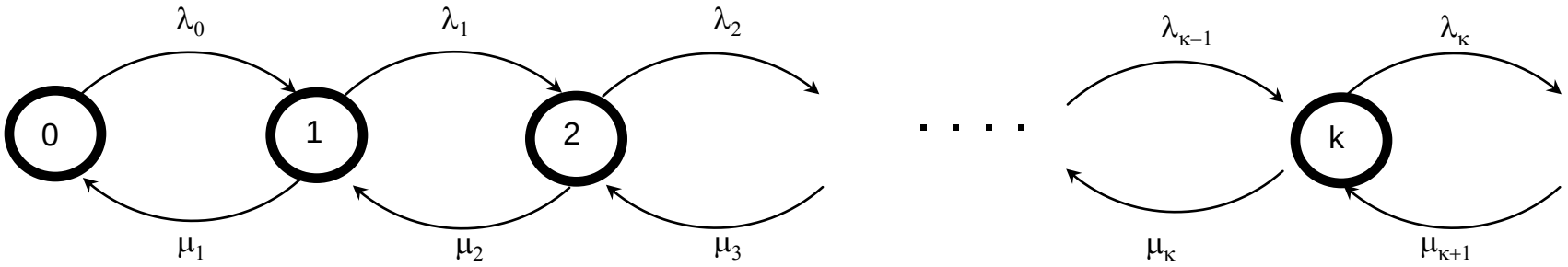


Typical throughput curve

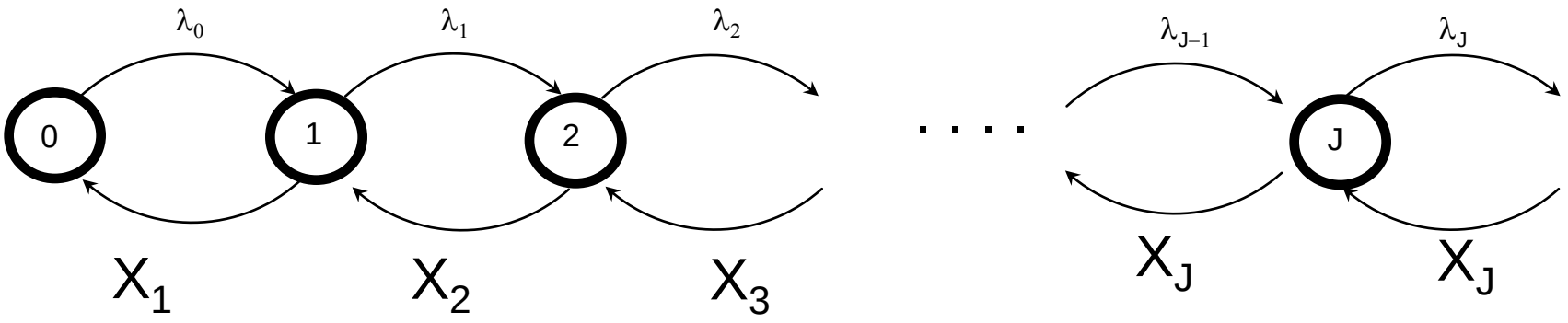


Approximated service rate

$$\mu_k = \begin{cases} X(k) & for\ k \le J \\ X(J) & for\ k > J \end{cases}$$



$$p_0 = \left[\sum_{k=0}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \right]^{-1}$$



$$\beta(k) = X_1 X_2 \ldots X_k$$

$\beta(k) = X_1 X_2 \ldots X_k$

$$p_0 = \left[1 + \sum_{k=1}^{J} \frac{\lambda^k}{\beta(k)} + \frac{\lambda^J}{\beta(J)} \frac{\rho}{1-\rho}\right]^{-1}$$

$$p_k = \begin{cases} p_0 \dfrac{\lambda^k}{\beta(k)} & for\ k \leq J \\ p_0 \dfrac{X(J)^J \rho^k}{\beta(J)} & for\ k > J \end{cases}$$

Kendall's notation

AD/STD/N_1/ N_2/ N_3

(arrival distribution, service time distribution, number of servers, max number of users in the system (queue + server), number of potential users, if not indicated then it is equal to infinitive number)

G: General

M: Markovian

D: Deterministic

M/M/3/5

- Markovian arrival distribution
- Markovian service distribution
- 3 servers
- Max 5 number of users in the system
- Infinite number of potential users

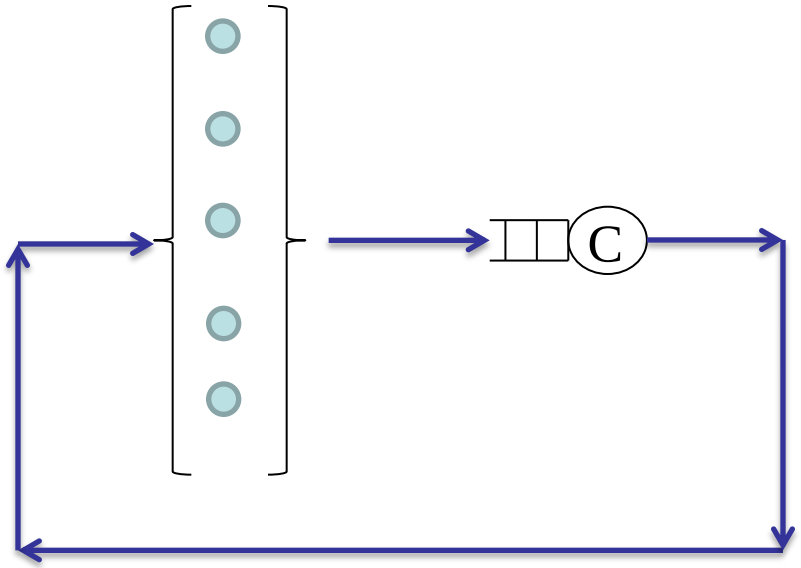
M/M/1//5

Interactive users (finite number)

- 1 server
- 5 interactive users with average ***thinking*** time equal to Z and exponentially distributed time,
  i.e. after an user received a service he/she waits fo a while - the thinking time

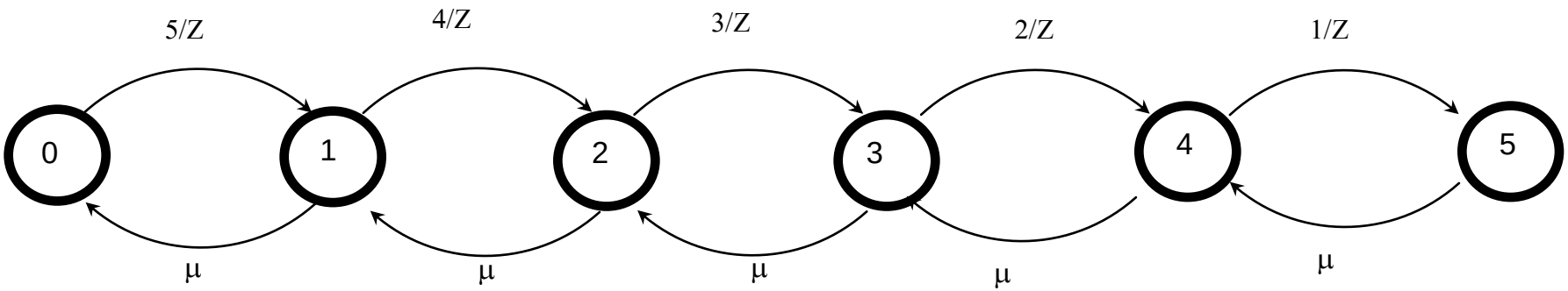
$\lambda_{\kappa} = (5-K)/Z =$ arrival rate

M/M/1//5



M/M/1//5

Interactive users (finite number) cont.

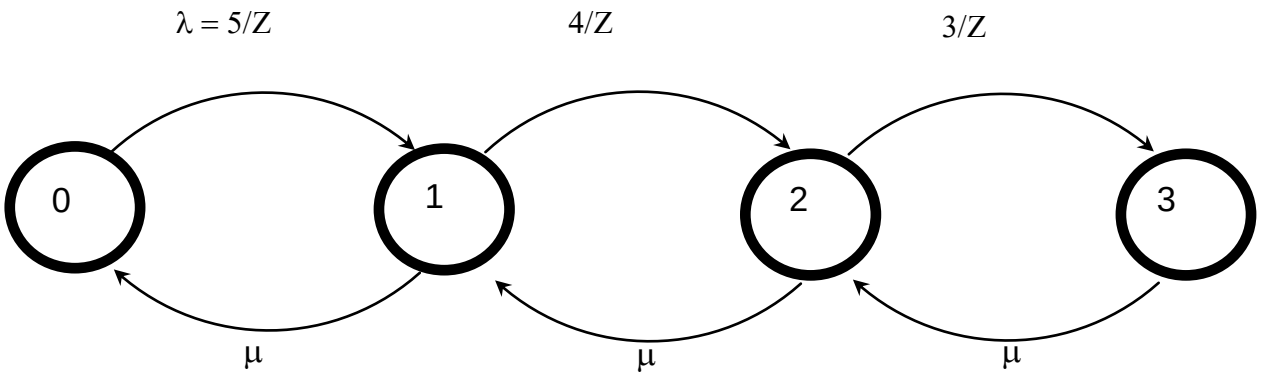


M/M/1/3/5

1 server, 3 max users in the system, 5 potential users, with average thinking time equal to Z

M/M/1/3/5

1 server, 3 max users in the system, 5 potential users, with average thinking time equal to Z

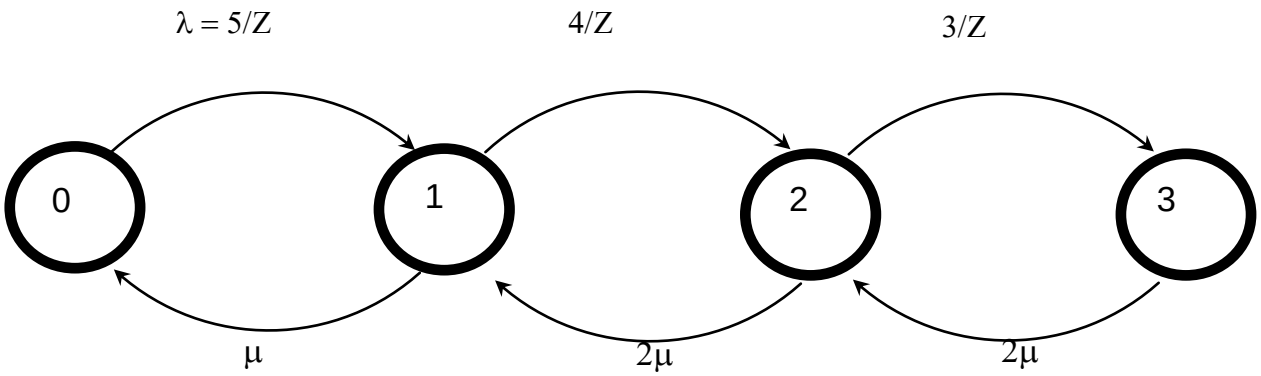


M/M/2/3/5

2 servers, 3 max users in the system, 5 potential users, with average thinking time equal to Z

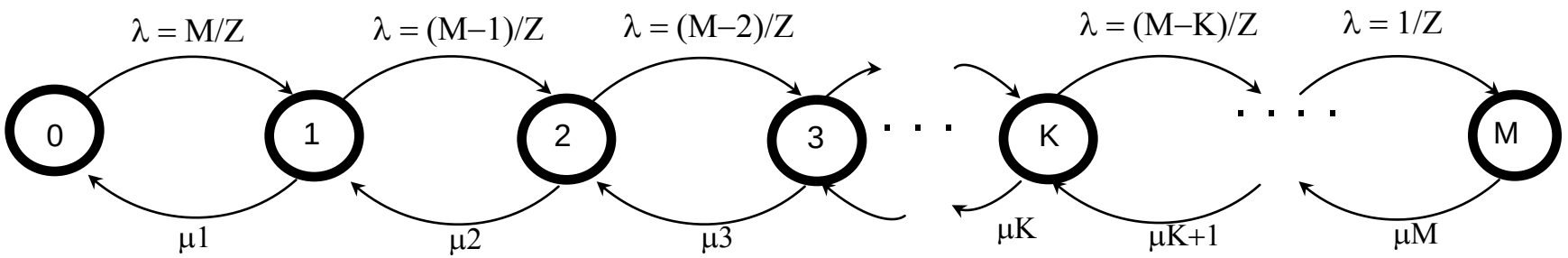
M/M/2/3/5

2 servers, 3 max users in the system, 5 potential users, with average thinking time equal to Z



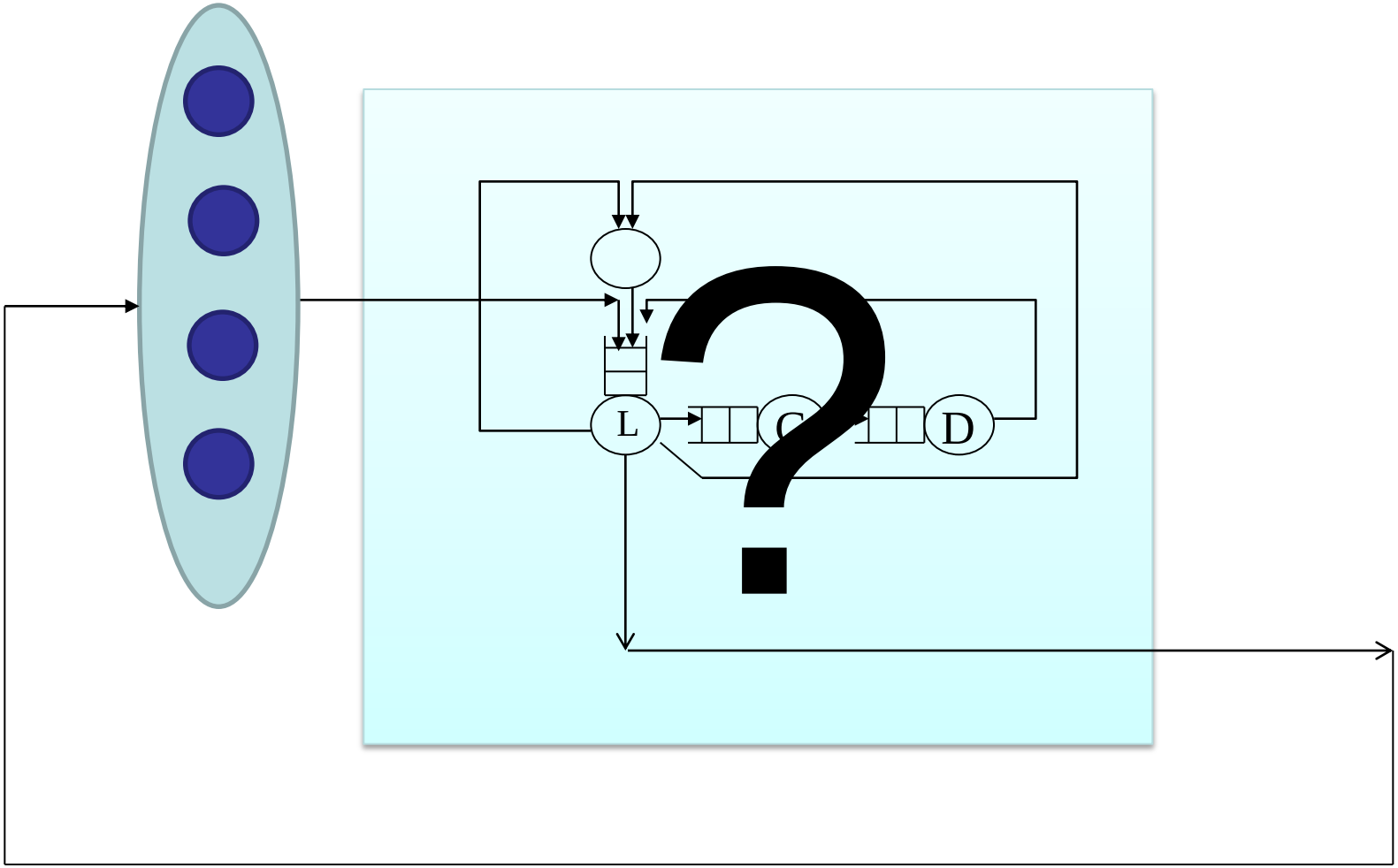
CLOSED

generalized system model with constant arrival rate and variable service rate if the «room» is enough for all the users

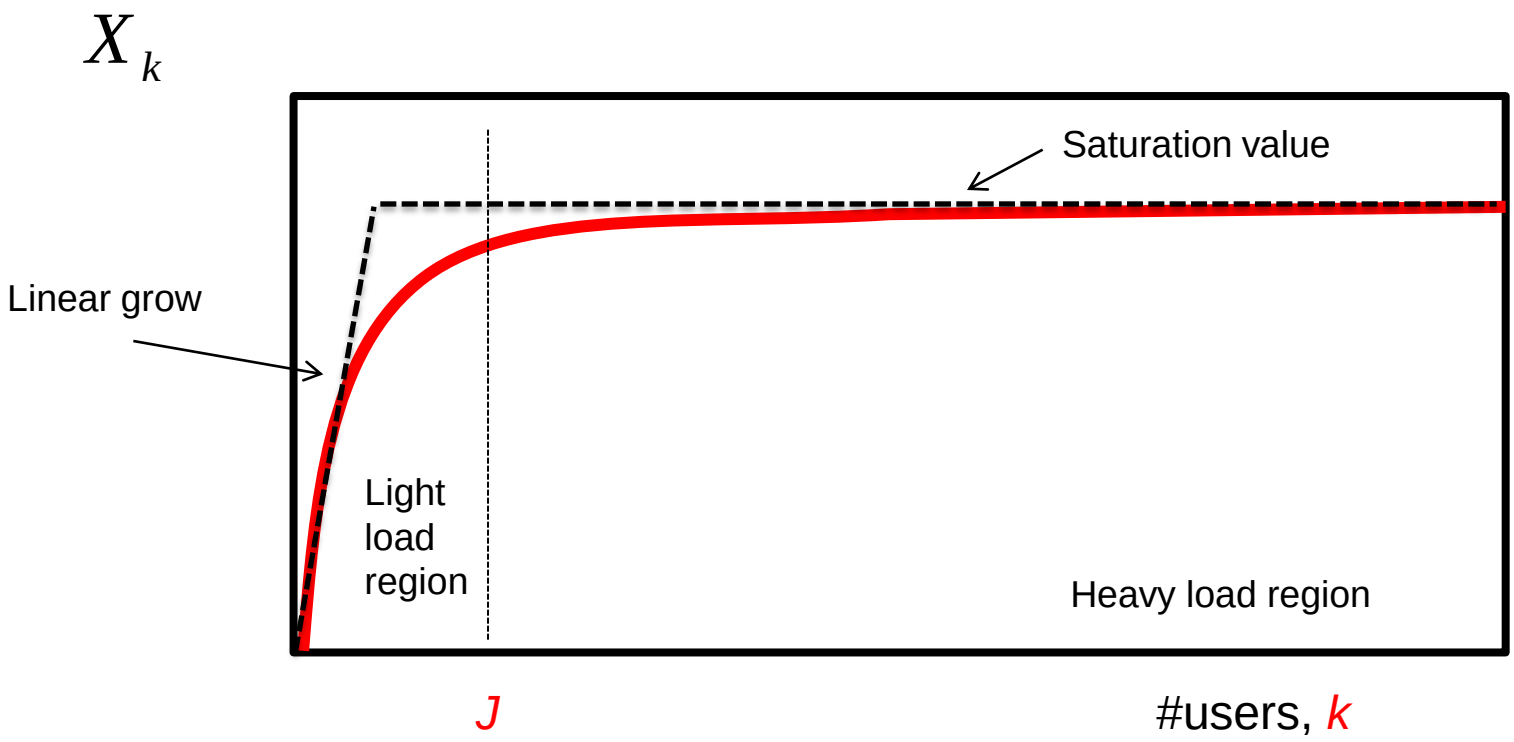


CLOSED

generalized system model with constant arrival rate and variable service rate



Typical throughput curve

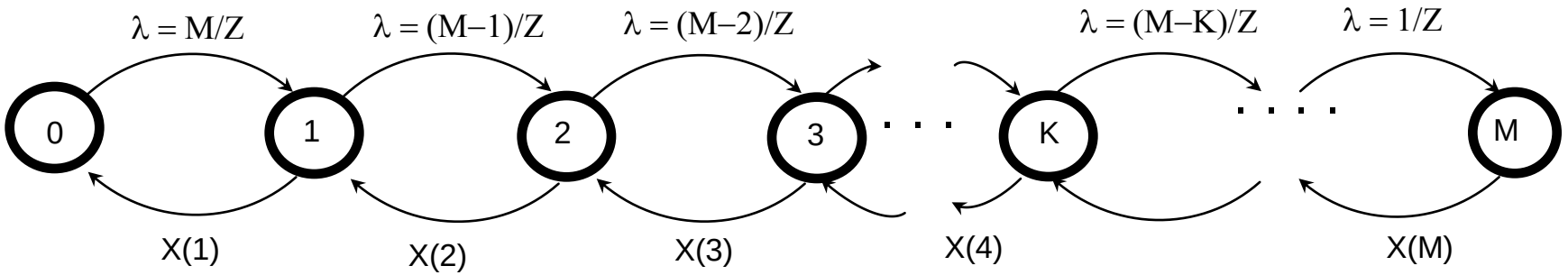


Approximated service rate

$$\mu_k = \begin{cases} X(k) & for\ k \le J \\ X(J) & for\ k > J \end{cases}$$

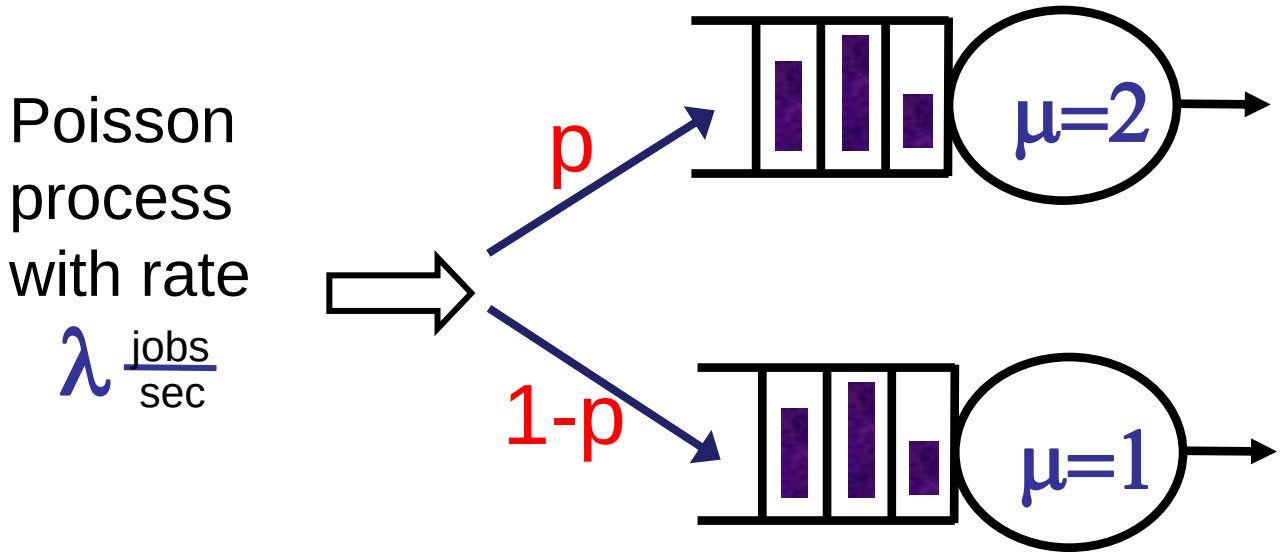
Closed model

Generalized system model with constant arrival rate and variable service rae rate if the«room» is enough for all the users



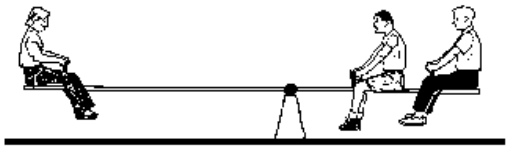
MULTI SERVER QUEUES

Load Balancing

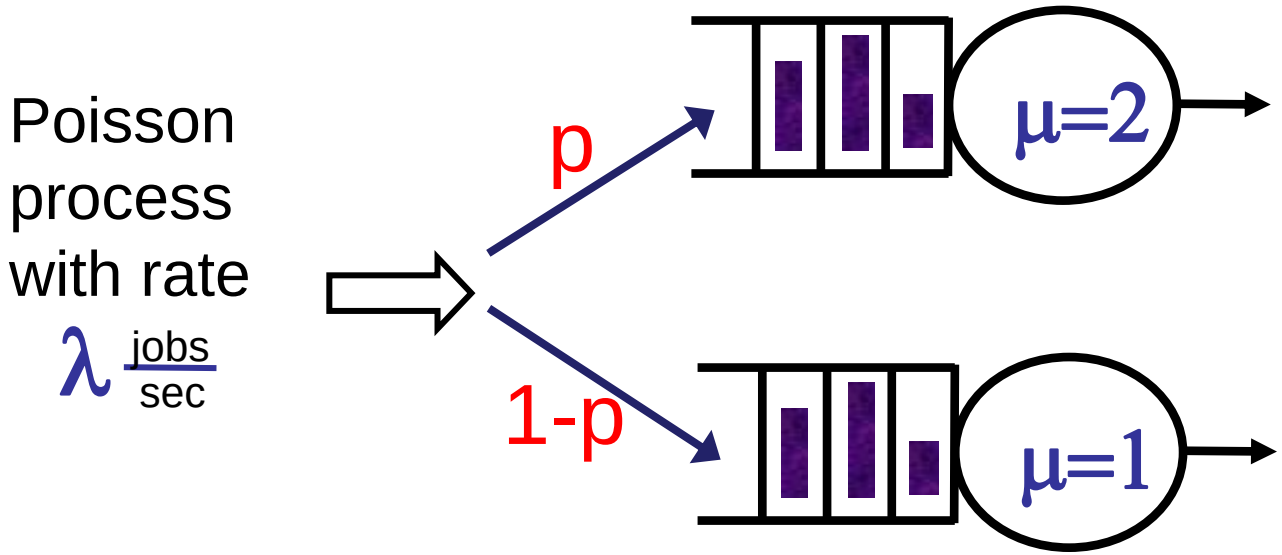


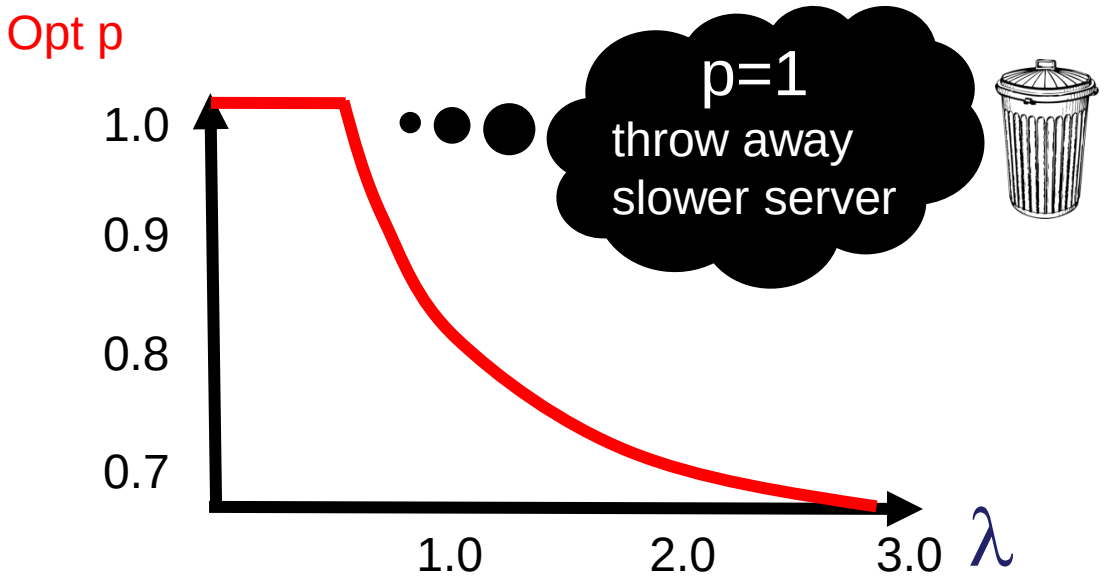
QUESTION: What is the optimal p to minimize **E[T]**?

(a) $p = 2/3$ (b) $p > 2/3$ (c) $p < 2/3$

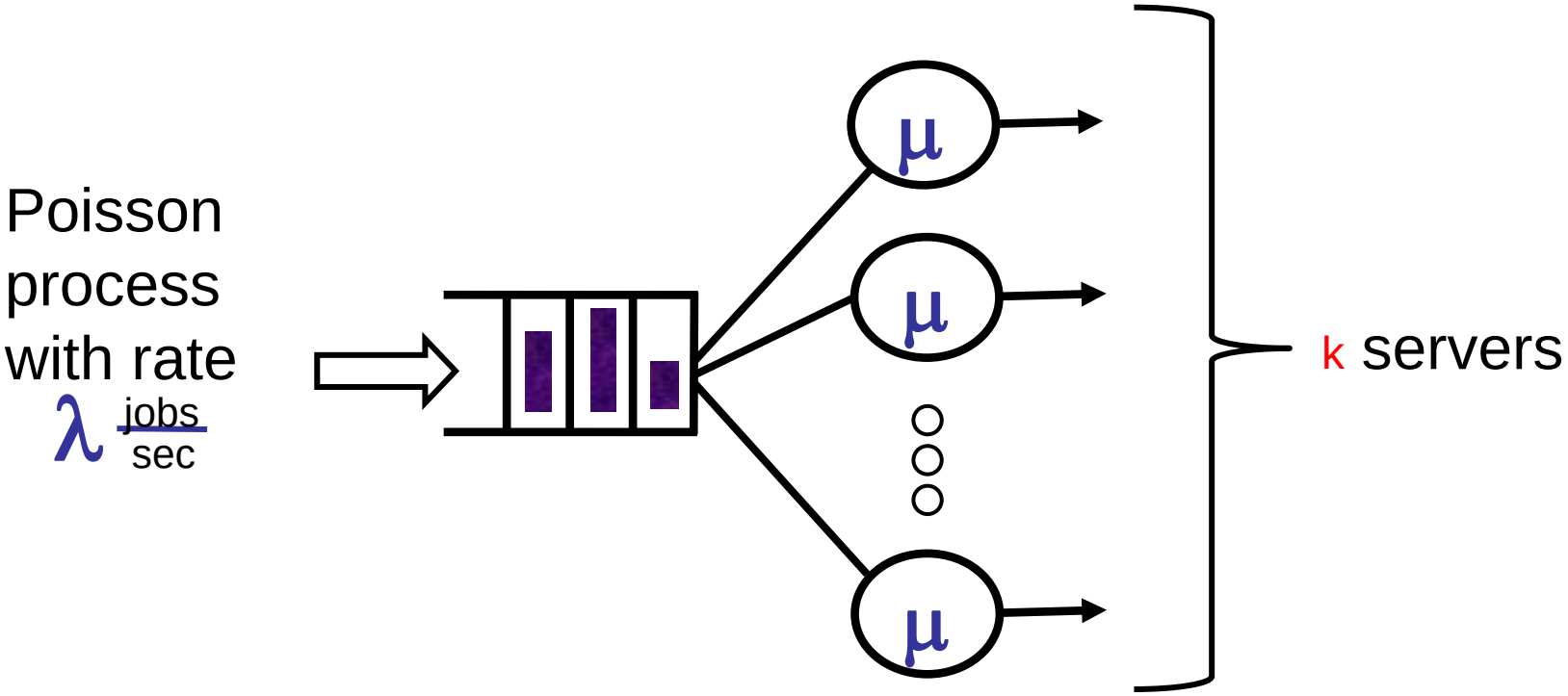


Load Balancing





M/M/k



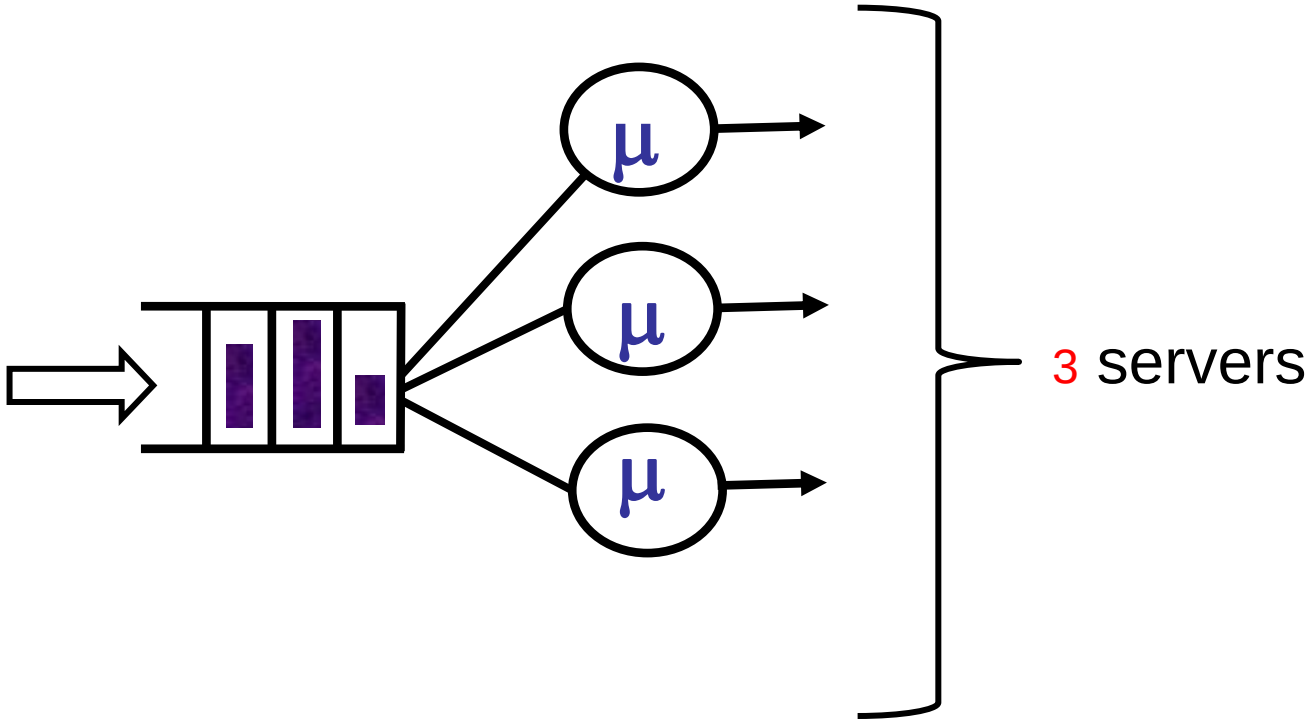
Central queue. Server takes job when free.

Service time $S \sim \text{Exp}(\mu)$

$$\rho \equiv \text{System Load} \equiv \frac{\lambda}{k\mu}$$

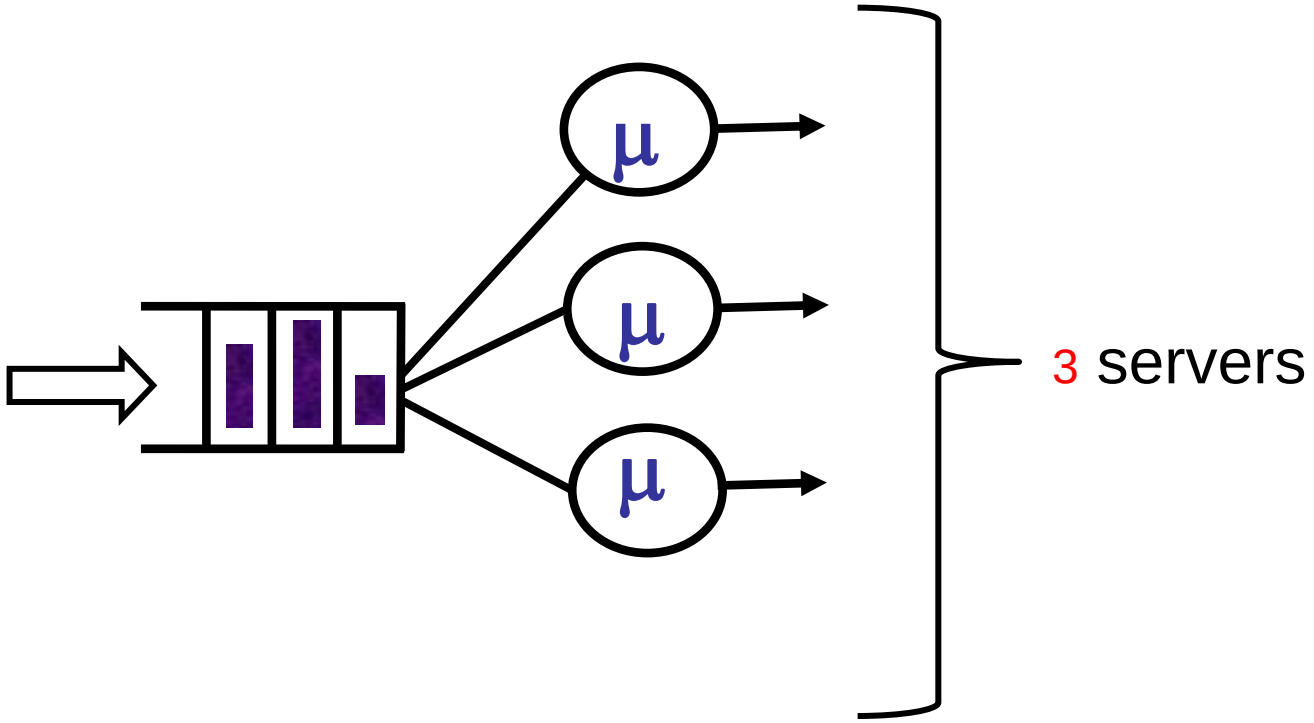
M/M/3

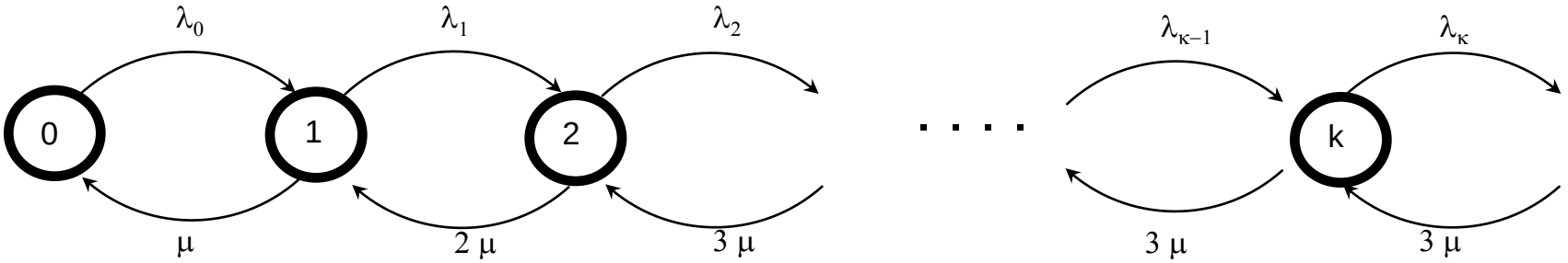
Poisson process with rate λ $\frac{\text{jobs}}{\text{sec}}$



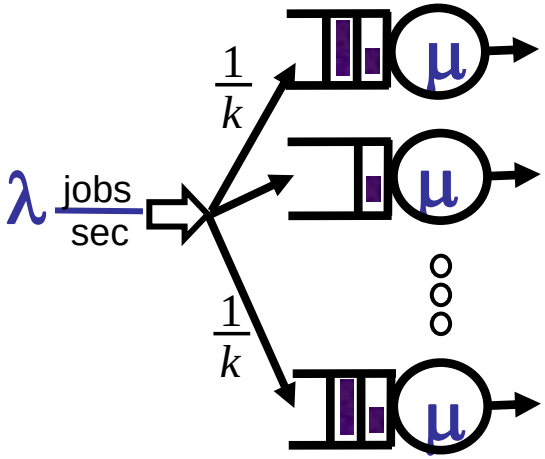
M/M/3

Poisson process with rate λ $\frac{\text{jobs}}{\text{sec}}$

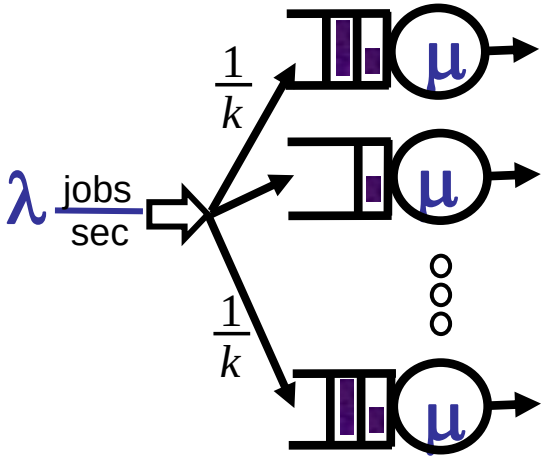


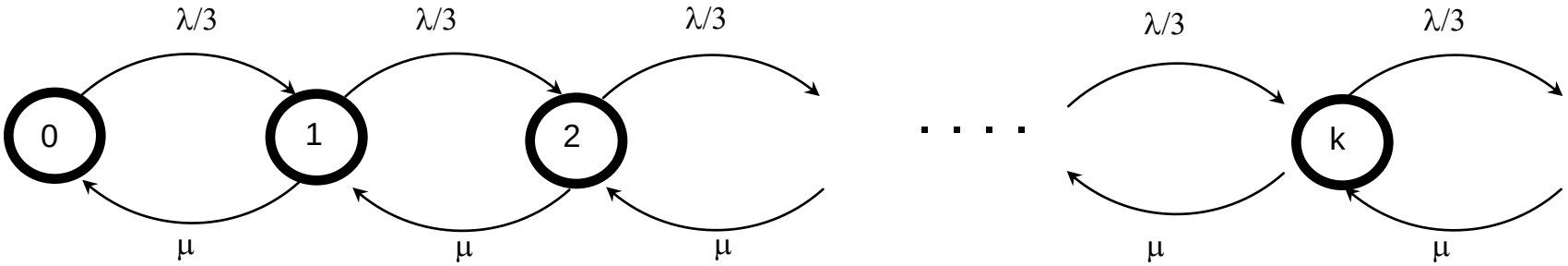


3 M/M/1



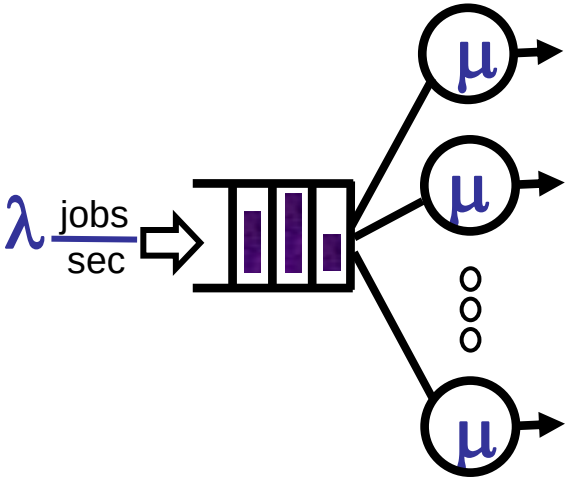
3 M/M/1





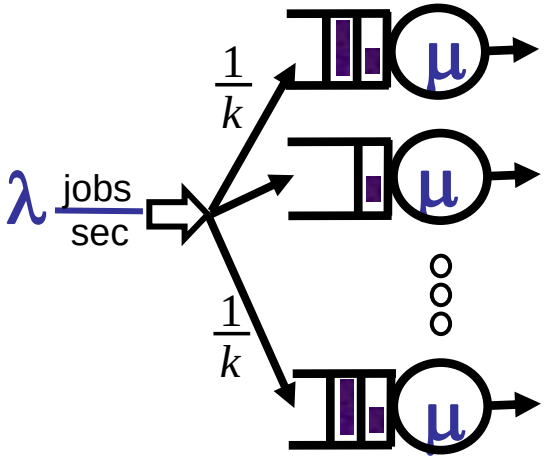
3 Architectures

M/M/k



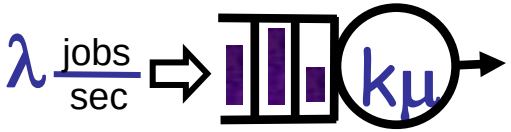
$$\rho = \frac{\lambda}{k\mu}$$

Splitting



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

M/M/1fast

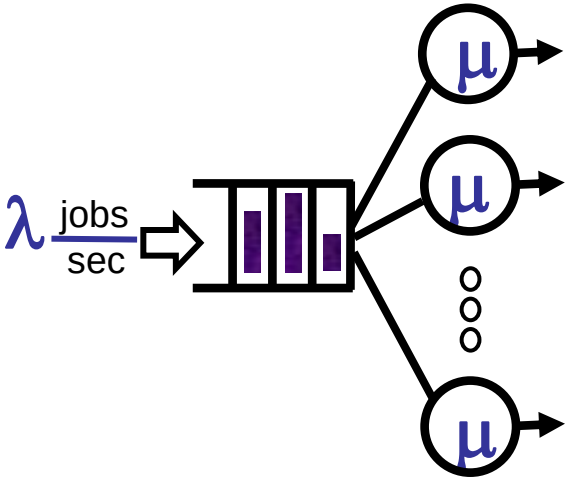


$$\rho = \frac{\lambda}{k\mu}$$

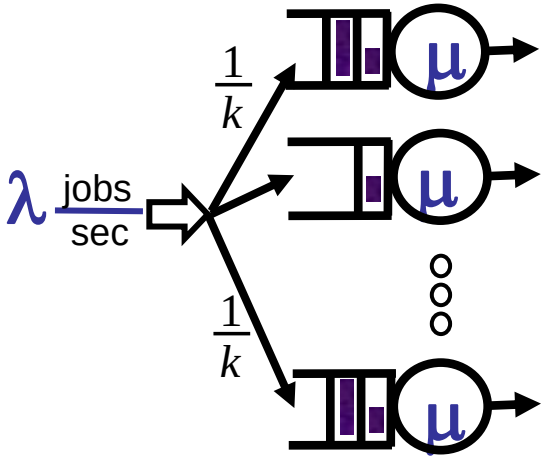
Q: Which is best for minimizing E[T]?

3 Architectures

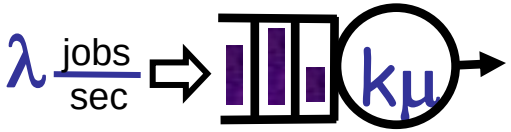
M/M/k



Splitting



M/M/1fast



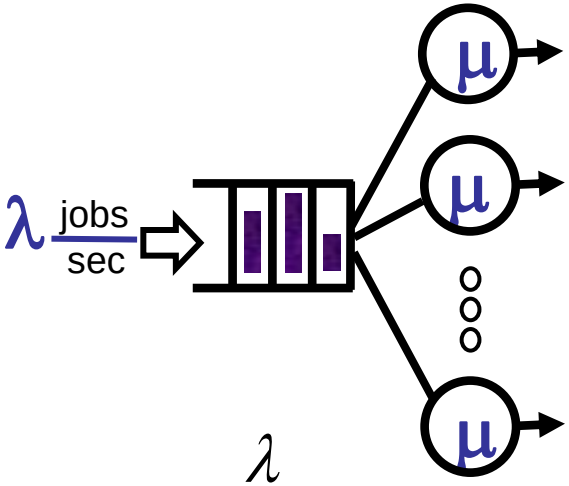
?

$$E[T_Q]^{M/M/1} = \frac{\rho}{1-\rho} \cdot E[S]$$

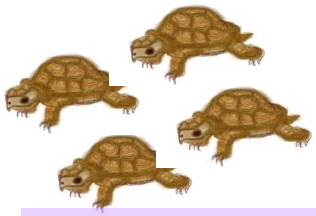
ρ is the same for both
E[S] of the second is K times faster than the first one

Many slow or 1 fast?

M/M/k

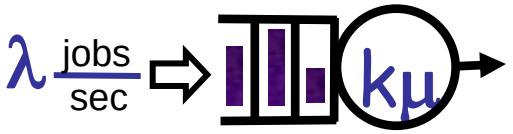


$$\rho = \frac{\lambda}{k\mu}$$



vs.

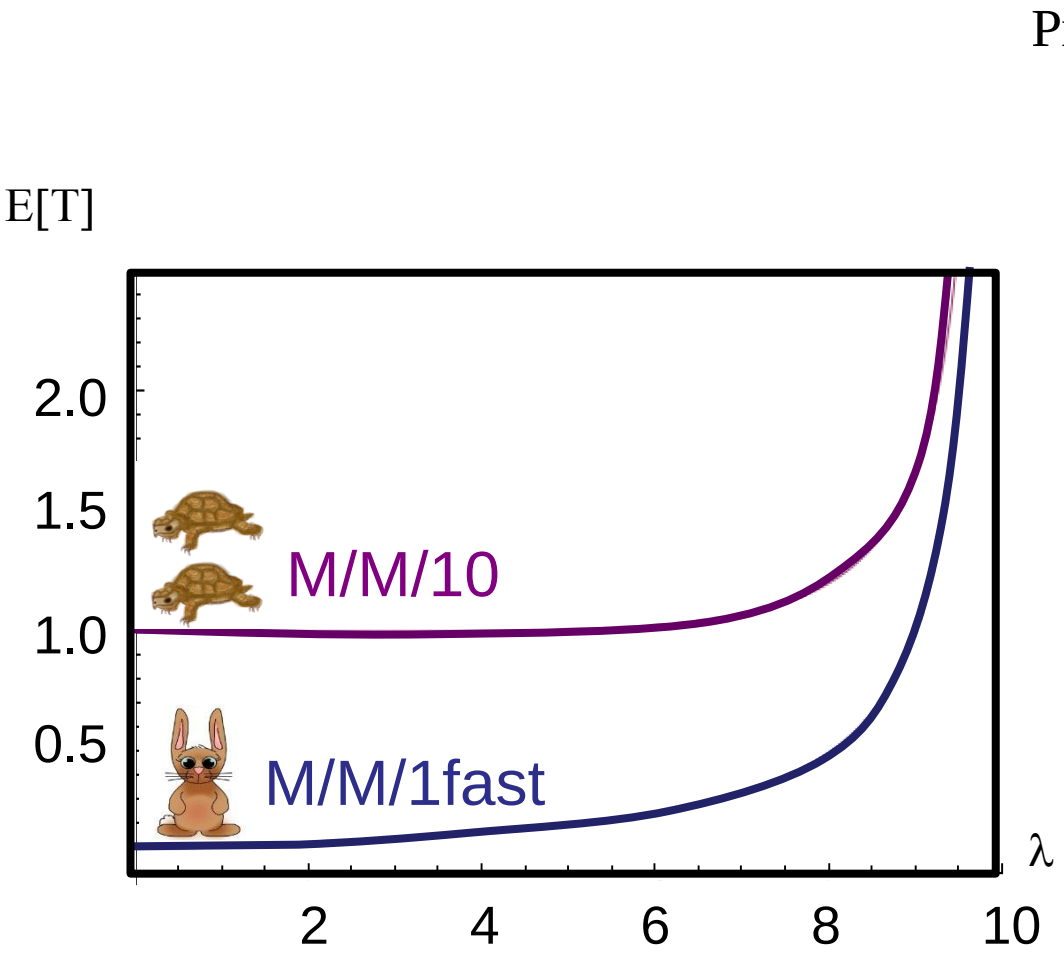
M/M/1fast

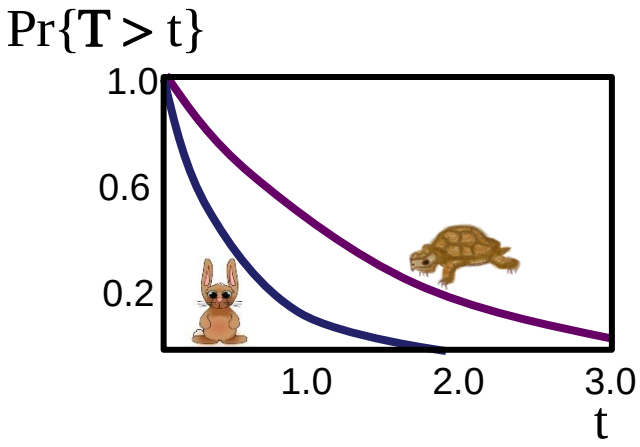


$$\rho = \frac{\lambda}{k\mu}$$

QUESTION: Which is best for minimizing $E[T]$?

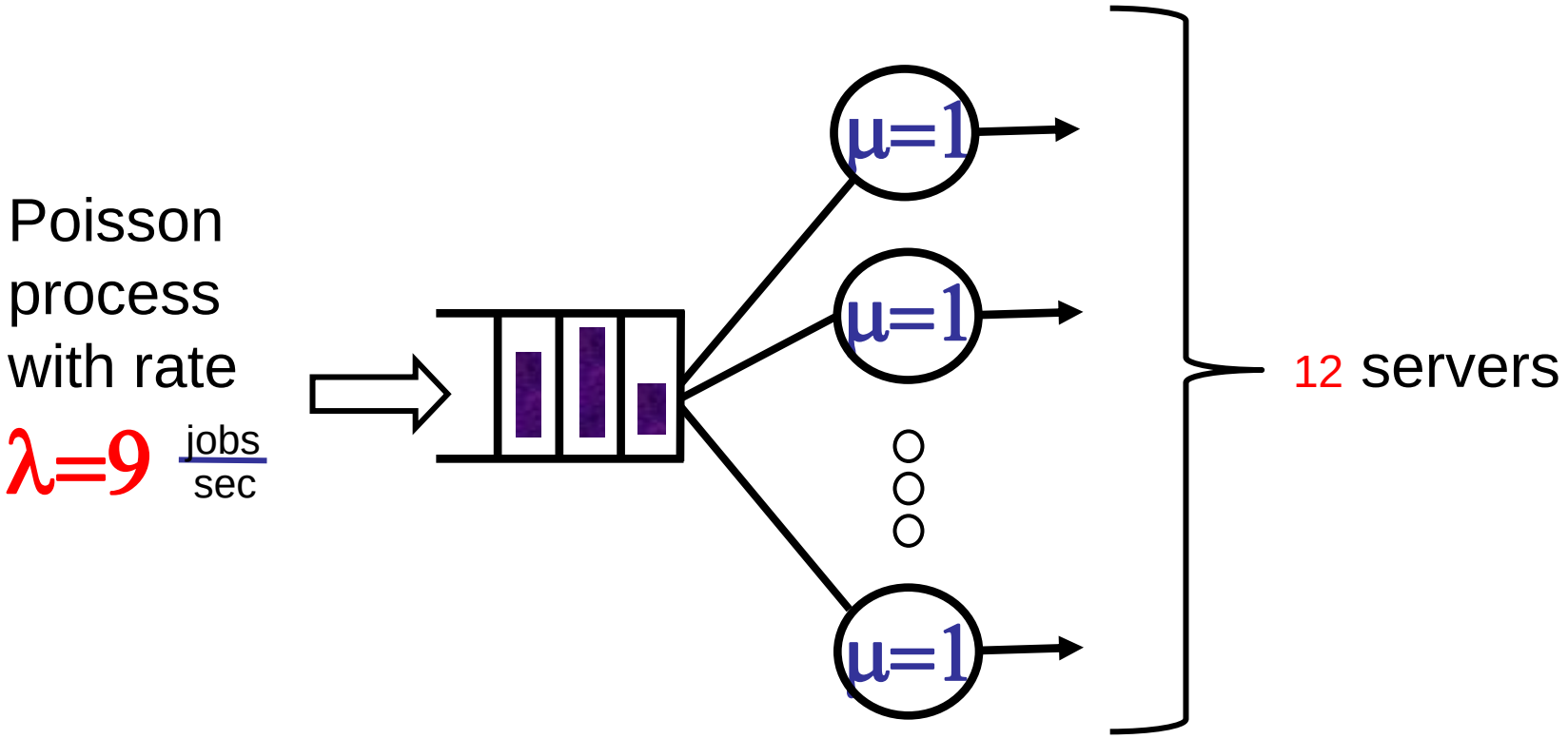
Many slow or 1 fast?





Capacity Provisioning & Scaling

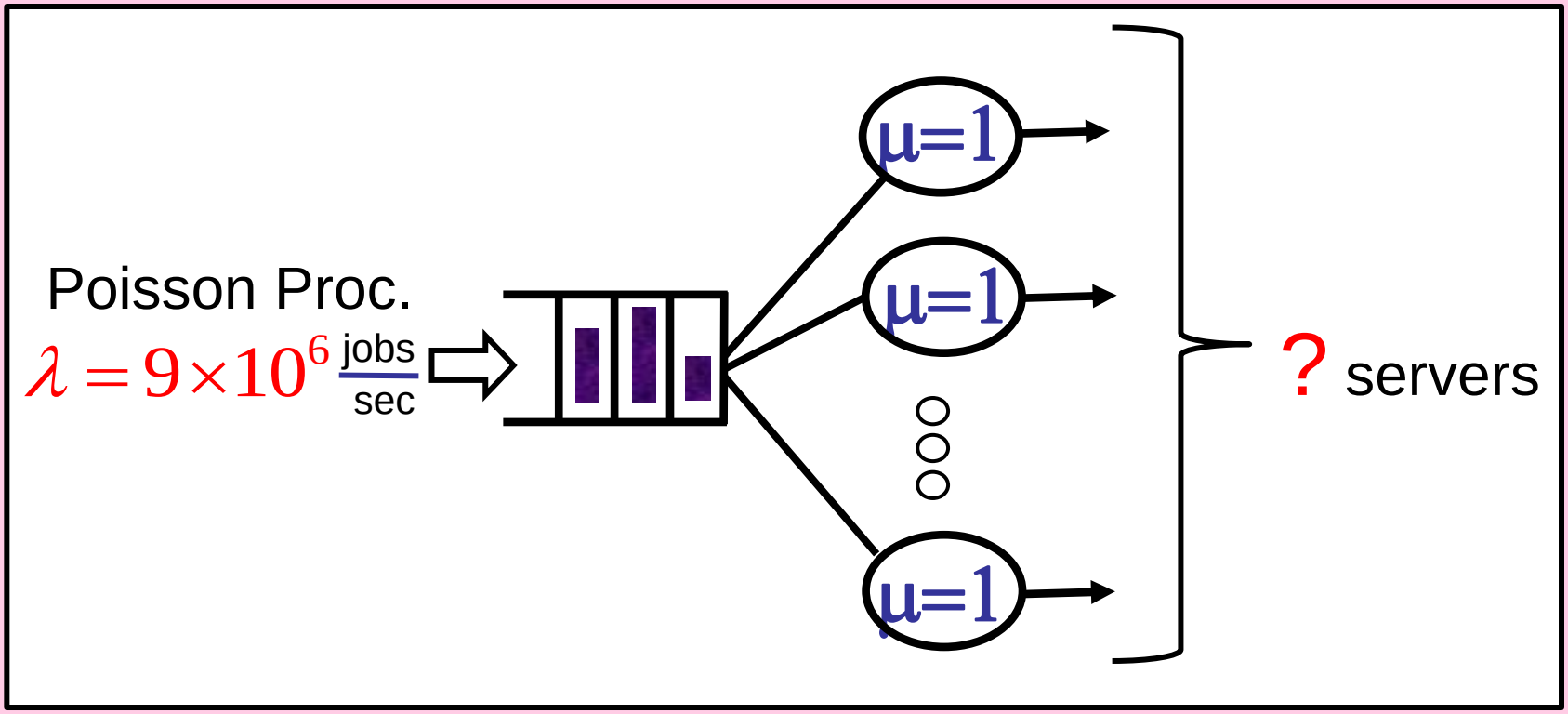
Consider the following example:



$P_Q =$ Probability an arrival has to queue $= 20\%$

Capacity Provisioning & Scaling

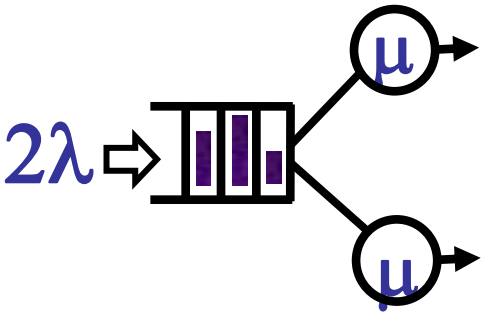
QUESTION: If arrival rate becomes 10^6 times higher, how many servers do we need to keep P_Q the same?



(a) 9.1×10^6
(b) 10×10^6
(c) 11×10^6
(d) 12×10^6
(e) 13×10^6
(f) none

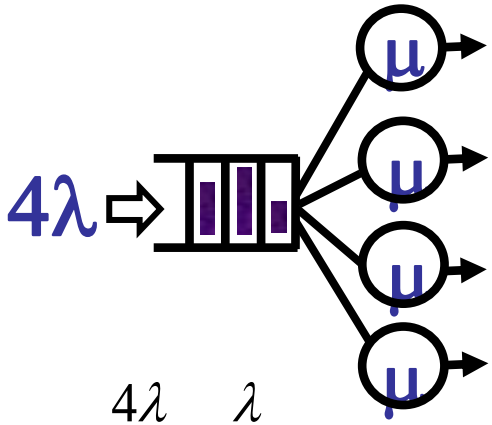
Proportional Scaling is Overkill

M/M/2



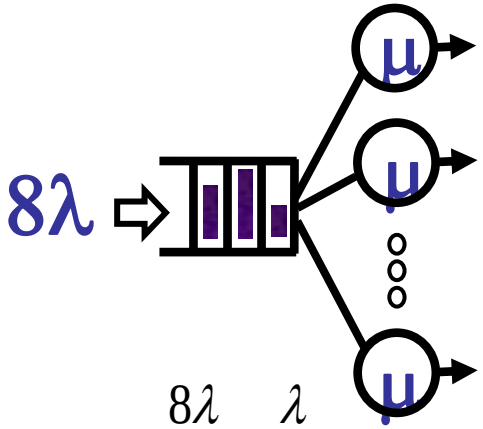
$$\rho = \frac{2\lambda}{2\mu} = \frac{\lambda}{\mu}$$

M/M/4

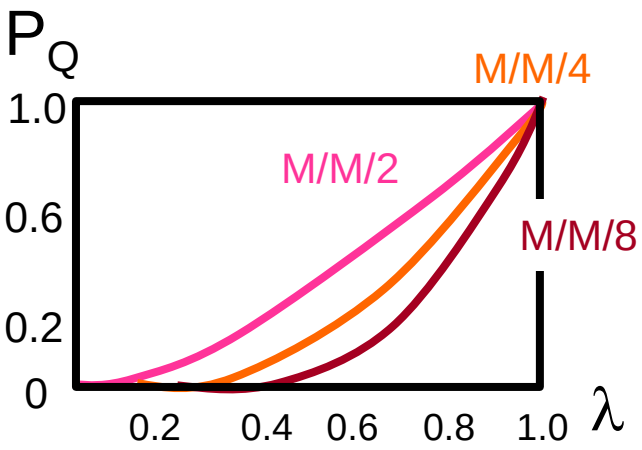


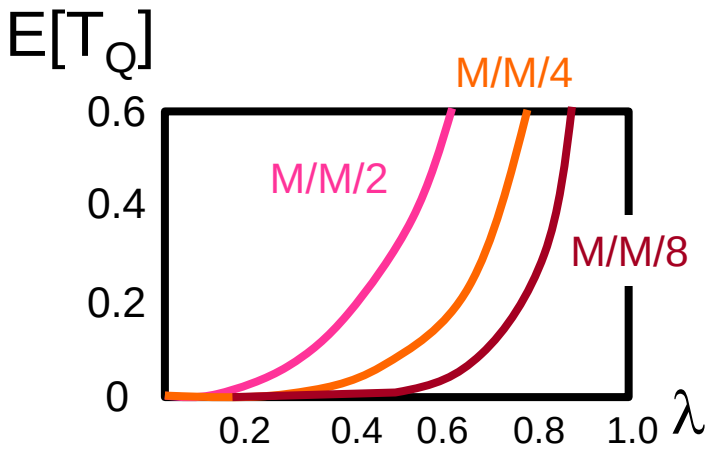
$$\rho = \frac{4\lambda}{4\mu} = \frac{\lambda}{\mu}$$

M/M/8



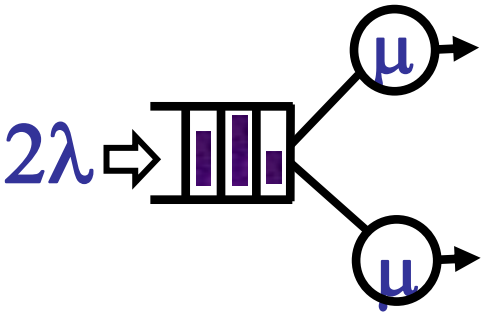
$$\rho = \frac{8\lambda}{8\mu} = \frac{\lambda}{\mu}$$





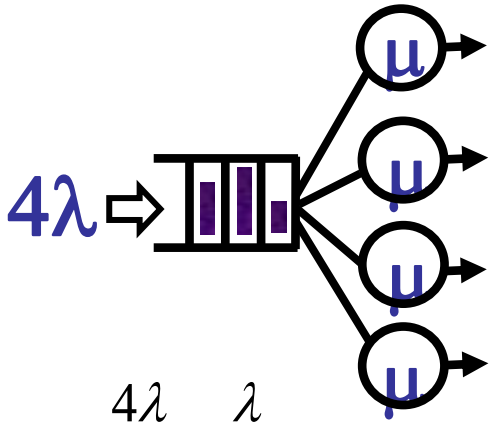
Proportional Scaling is Overkill

M/M/2



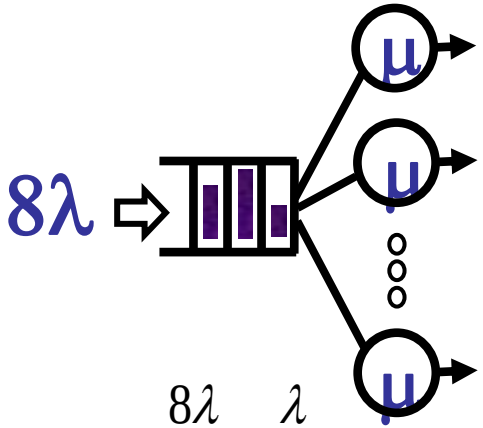
$$\rho = \frac{2\lambda}{2\mu} = \frac{\lambda}{\mu}$$

M/M/4



$$\rho = \frac{4\lambda}{4\mu} = \frac{\lambda}{\mu}$$

M/M/8



$$\rho = \frac{8\lambda}{8\mu} = \frac{\lambda}{\mu}$$

More servers at same system load ➔ lower P_Q ➔ lower $E[T_Q]$

high ρ ⇏ high $E[T_Q]$, ➔ given enough servers