# Diseño y Evaluación de Configuraciones

## III Modelo de colas

J.M. Drake

Notas:

## Modelos de colas

- Los sistemas informáticos optimizan su rendimiento equilibrando la capacidad de los recursos de que disponen con los requerimientos de las aplicaciones que ejecutan.
- Para evaluar el comportamiento de las aplicaciones y configurar de forma óptima un sistema hay que saber evaluar características como:
  - Tiempo efectivo de acceso a un recurso.
  - Número máximo de invocaciones que puede atender un recurso.
  - Nivel de ocupación de un recurso.
  - Cuantos clientes pueden estar a la espera de un recurso.
- Los modelos de colas constituye una abstracción basada en modelos probabilísticos que permiten, describir, analizar y diseñar estos tipos de sistemas.
- Cada cola describe un recurso que genera contención, y en el que el tiempo efectivo de servicio (tiempo de estancia) es función del tiempo real de servicio (tiempo de servicio) y del tiempo de espera para acceder al mismo (tiempo de espera).



Notas:

In computer systems, many jobs share the system resources such as CPU, disks, and other devices. Since generally only one job can use the resource at any time, all other jobs wanting to use that resource wait in queues. Queueing theory helps in determining the time that the jobs spend in various queues in the system. These times can then be combined to predict the response time, which is basically the total time that the job spends inside the system. It is not, therefore, surprising that queueing models have become so popular among computer systems performance analysts.

Queueing theory is the key analytical modeling technique used for computer systems performance analysis. The literature on queueing theory is vast. Several hundred papers are published every year. Fortunately, a large percentage of day-to-day performance questions can be answered using only a few techniques. To begin with, you need a knowledge of the queueing notation and some results on single-queue systems. For systems with multiple queues, operational analysis, mean-value analysis, and convolution are very useful. In addition, the technique of hierarchical modeling is helpful in analyzing large systems. The discussion in this part has been limited to these techniques.

Following are examples of questions that you should be able to answer after reading this part:

**1.** What are the various types of queues?

**2.** What is meant by an M/M/*m/B/K* queue?

**3.** How should you obtain response time, queue lengths, and server utilizations?

**4.** How should you represent a system using a network of several queues?

**5.** How should you analyze simple queueing networks?

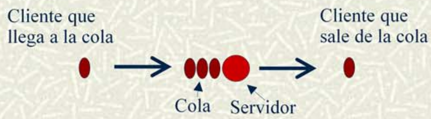
**6.** How should you obtain bounds on the system performance using queueing models?

**7.** How should you obtain variance and other statistics on system performance?

**8.** How should you subdivide a large queueing network model and solve it?

## Cola

- Es una abstracción que modela un ente capaz de prestar servicios a los cliente que lo requieren:
  - Cuando la cola está libre presta de forma inmediata (sin contención) el servicio que se le requiere.
  - Si la cola esta ocupada prestando el servicio a otro cliente, el cliente que invoca se suspende a la espera de que quede libre.
  - Cuando el servidor finaliza de atender a un cliente, el servidor elige uno de los clientes que se encuentran en espera, e inicia su atención. La política de planificación (*Scheduling Policy*) define que cliente escoge de entre los que están esperando.
  - Ningún cliente que ha invocado un servicio de una cola, puede abandonar la cola sin que se le haya prestado el servicio.






Notas:

A queue is a system to which customer arrive to receive a service. When the system is busy serving others customers, incoming customers wait for thry turn . Upon completion of a service, the customer that must be server next is selected according to some queueing policy . No customers are allowed to leave the queue before having received service.

## Magnitudes útiles en análisis de colas

- **Proceso de llegada**: Establece el patrón de llegada. Se caracteriza por la variable probabilística tiempo entre llegadas (*interarrival times*).
- **Tiempo de servicio**: Tiempo que el servidor realmente aplica a los clientes. Es una variable probabilistica.
- **Número de servidores**: Número de servidores que pueden atender simultáneamente a los clientes. Es un número fijo (aunque es posible que sólo algunos de ellos tengan trabajo)
- **Capacidad de la cola**: Número máximo de clientes que son admitidos en la espera a ser atendidos. Es un número entero o infinito.
- **Tamaño de la población**: Número de clientes que potencialmente pueden requerir servicios del servidor.
- **Política de atención**: Criterio con el que al finalizar de prestar un servicio es elegido el próximo cliente entre los que están esperando. Ejemplos: FIFO, LIFO, RR, etc...



Notas:

In order to analyze such a queue, the following characteristics of the system should be specified:

1. *Arrival Process:* If the students arrive at times $t_1,t_2,...,t_j$, the random variables $\tau_j=t_j-t_j-1$ are called the **interarrival times**. It is generally assumed that the$\tau_j$ form a sequence of Independent and Identically Distributed (IID) random variables. The most common arrival process is the so-called **Poisson arrivals**, which simply means that the interarrival times are IID and are exponentially distributed. Other distributions, such as the Erlang and hyperexponential, are also used. In fact, several queueing results are valid for all distributions of interarrival times. In such a case, the result is said to hold for a **general** distribution.

***2.Service Time Distribution****:* We also need to know the time each student spends at the terminal. This is called the service time. It is common to assume that the service times are random variables, which are IID. The distribution most commonly used is the exponential distribution. Other distributions, such as the Erlang, hyperexponential, and general, are also used. Again the results for a general distribution apply to all service time distributions.

**3. *Number of Servers****:* The terminal room may have one or more terminals, all of which are considered part of the same queueing system since they are all identical, and any terminal may be assigned to any student. If all the servers are not identical, they are usually divided into groups of identical servers with separate queues for each group. In this case, each group is a queueing system.

**4. *System Capacity****:* The maximum number of students who can stay may be limited due to space availability and also to avoid long waiting times. This number is called the system capacity. In most systems, the capacity is finite. However, if the number is large, it is easier to analyze if infinite capacity is assumed. The system capacity includes those waiting for service as well as those receiving service.

**5. *Population Size****:* The total number of potential students who can ever come to the computer center is the population size. In most real systems, the population size is finite. If this size is large, once again, it is easier to analyze if we assume that the size is infinite.

**6. *Service Discipline****:* The order in which the students are served is called the service discipline.

## Parámetros de una cola

- Tiempo de llegada de requerimientos:
  - α: Tipo de distribución estadística de los tiempos de acceso
  - σ: Tipo de distribución estadística de los tiempos de servicios
  - m: Número de servidores
  - β: Longitud de la cola.
  - N: Número de clientes potenciales.
  - Q: Política de atención
- Notación de Kendall:

$$\alpha / \sigma / m / \beta / N / Q$$

Si como es habitual β=∞, N=∞ y Q=FIFO, la notación se reduce a

$$\alpha / \sigma / m$$



Notas:

To specify a queueing system, we need to specify these six parameters. Queueing theorists, therefore, use a shorthand notation called the **Kendall notation** in the form α/σ/*m*/*β*/*N*/Q, where the letters correspond in order to the six parameters listed above. That is, α is the interarrival time distribution, σ is the service time distribution, *m* is the number of servers, *β* is the number of buffers (system capacity), *N* is the population size, and Q is the service discipline.

The distributions for interarrival time and service times are generally denoted by a one-letter symbol as follows:

M Exponential

E*k* Erlang with parameter *k*

H*k* Hyperexponential with parameter *k*

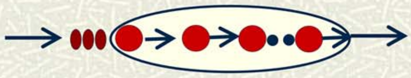
D Deterministic

G General

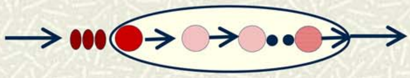
A deterministic distribution implies that the times are constant and there is no variance. A general distribution means that the distribution is not specified and the results are valid for all distributions.

## Los tipos de distribución habituales (1)

- (D) Determinista o periódica: La llegada de requerimientos es periódica y los periodo entre llegadas es constante $T=1/\lambda$, y sin variación estadística entre ellas.
- (M) Exponencial: La llegada de requerimientos es de Poisson. La distribución de los tiempos es de tipo exponencial y esta caracterizada por el tiempo medio entre llegadas $T_m=1/\lambda$.
- (U) Uniforme: La distribución de los tiempos entre llegadas es uniforme con valores en el rango $T_m$-Rango/2<t<$T_m$+Rango/2. Es un tipo de distribución hipoexponencial.
- ($E_r$) Erlang (r): Es un tipo de distribución hipoexponencial de tipo Gamma. Se pueden considerar como una composición de r etapas exponenciales de igual frecuencia media colocadas en cascada.



- Hipoexponencial: Tiene una variabilidad inferior a la exponencial. Se puede considerar como una composición en cascada de r etapas exponenciales cada una de una frecuencia media diferentes.



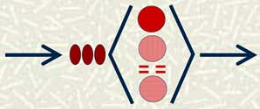


Notas:

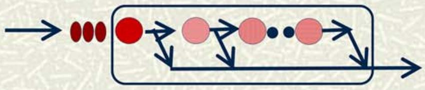
Las características de las diferentes distribuciones se pueden consultar p.e. en Raj Jain:"The art of Computer System Performance Analysis"

## Los tipos de distribución habituales (2)

- ($H_r$) Hiperexponencial: Tiene una variabilidad superior a la exponencial. Se puede considerar como la composición en paralelo de un conjunto de r servidores exponenciales de diferente frecuencia media.



- Coaxian: Es un tipo de distribución hipoexponencial equivalente a una composición en cascada de r servidores exponenciales, pero con la posibilidad de que en cualquier etapa se pueda abandona el servicio.



- G General: Es un tipo de distribución arbitraria.



Notas:

## Políticas de atención al cliente en espera

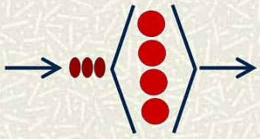
- FIFO: Se atiende el cliente que lleva mas tiempo esperando en la cola.
- LIFO: Se atiende al último cliente que ha realizado la invocación del servicio.
- RoundRobin: Se atienden concurrentemente a todos los cliente dedicando a cada una de ellos sucesivas ventanas de tiempo.
- Procesador compartido: Se atienden concurrentemente a todos los clientes dedicando una fracción de la capacidad de procesamientos a cada cliente en espera.
- Prioridad: Se atienden a los clientes de acuerdo con la prioridad que a cada uno se le ha asignado.
- EDF: Se elige aquel requerimiento al que le falta el menor tiempo para que termine su plazo



Notas:

The most common discipline is First Come, First Served (FCFS). Other possibilities are Last Come, First Served (LCFS) and Last Come, First Served with Preempt and Resume (LCFS-PR). Computer system CPUs generally use Round-Robin (RR) with a fixed-size quantum. If the quantum size is small compared to average service time, it is called Processor Sharing (PS) since each of the *n* waiting jobs would then receive 1/*n*th of the processor's time. A system with a fixed delay, for example, a satellite communication link, is called an **Infinite Server** (IS) or a **delay center**. Terminals in timesharing systems are usually modeled as delay centers. Sometimes the scheduling is based on the service time required. Examples of such disciplines are Shortest Processing Time first (SPT), Shortest Remaining Processing Time first (SRPT), Shortest Expected Processing Time first (SEPT), and Shortest Expected Remaining Pro-first (SERPT). In the real world, occasionally one may encounter Biggest In, First Served (BIFS) or Loudest Voice, First Served (LVFS).

## Ejemplo de especificación de una cola



**M/G/4/50/2000/LIFO**

- La entradas de requerimientos es de naturaleza exponencial
- Los tiempos de servicios son de naturaleza General
- Tiene 4 servidores
- El tamaño de la cola es de 50
- La población de clientes que requieren su servicios es de 2000
- Se atienden a los clientes utilizando una política LIFO



Notas:

Example **M/M/3/20/1500/FCFS** denotes a single-queue system with the following parameters:

**1.** The time between successive arrivals is exponentially distributed.

**2.** The service times are exponentially distributed.

**3.** There are three servers.

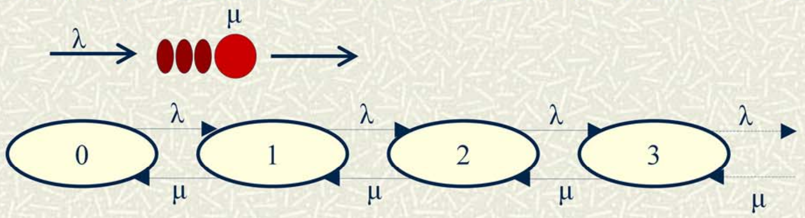
**4.** The queue has buffers for 20 jobs. This consists of three places for jobs being served and 17 buffers for jobs waiting for service. After the number of jobs reaches 20, all arriving jobs are lost until the queue size decreases.

**5.** There is a total of 1500 jobs that can be serviced.

**6.** The service discipline is first come, first served.

## Una cola y procesos probabilístico

- Una cola es una forma condensada de un proceso probabilístico representable mediante un máquina de estados.
- Por ejemplo una cola M/M/1 corresponde a la representación del proceso




Es estable si μ>λ

$$\left.\begin{array}{l}\lambda P_j = \mu P_{j+1} \\ \sum_{\forall j} P_j = 1\end{array}\right. \xrightarrow{t\to\infty} \Rightarrow \left\{P_j = \left(\frac{\lambda}{\mu}\right)^j\left(1-\frac{\lambda}{\mu}\right) \Rightarrow \left\{\begin{array}{l} P_0 = 1-\frac{\lambda}{\mu} \\ P_1 = \frac{\lambda}{\mu}\left(1-\frac{\lambda}{\mu}\right) \\ \ldots \\ P_\infty \to 0 \end{array}\right.\right.$$

- Probabilidad de que esté vacía => $P_0=1-\lambda/\mu$
- Número medio de ocupación => $Q=\Sigma\, j\, P_j = (\lambda/\mu)/(1-\lambda/\mu)$



Notas:

An M/M/1 queue, which is the most commonly used type of queue, can be used to model single–processor systems or to model individual devices in a computer system. It is assumed that the interarrival times and the service times are exponentially distributed and there is only one server. There are no buffer or population size limitations and the service discipline is FCFS. Th analyze this type of queue, we need to know only the mean arrival rate » and the mean service rate $\mu$.

The state of this queue is given by the number of jobs in the system. A state transition diagram for the system is shown in the figure. It is similar to that of the birth-death processes.

## Magnitudes observables y métricas

- **Magnitudes observables**
  - A => Número de requerimientos que llegan en la ventana T.
  - C => Número de requerimientos que son servidos durante T.
  - T => Ventana de tiempo de medida.
  - B => Tiempo que el servidor ha estado ocupado en T
- **Métricas**
  - $\lambda$=A/T => Tasa media de entrada
  - X= C/T => Througput: Tasa de requerimientos servidos
  - S= B/C => Tiempo medio de servicio
  - U= B/T => Tanto por ciento de utilización
  - W=S Q => Tiempo medio de espera
  - R=S+W => Tiempo medio de estancia
  - Q= $\lambda$ R => Numero medio de req. en cola. Fórmula de LITTLE
  - Q=X R => Fórmula de LITTLE
  - U=X S => Utilización en función del throughput



Notas:

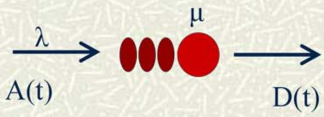
Arrival rate($\lambda$): Whereas A counts the number of request arriving at a queueing center, $\lambda$ express the rate at which they arrive. It is the number of arrivals per units of time. For example, if an operating system is instrumented such that it counts the number of request for service at some resource, then the total number of the counts during the measurement period is the arrival rate.

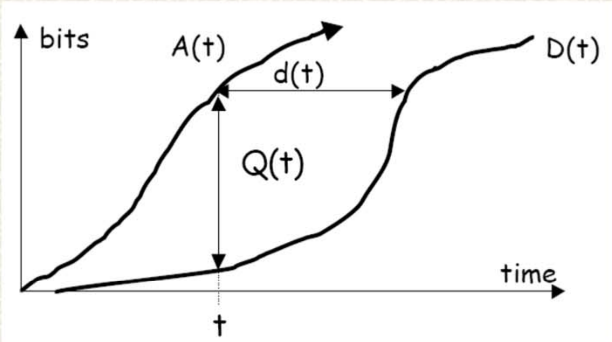
Throughput (X): This quantity is also a rate. Since it is a direct measure of the rate of completions, it is a counterpart of the arrival rate. As we shall see shortly, $\lambda$ and X can be equal for certain types of queues. In the event that they are not equal, we need to be able to distinguish them.

Service time (S): It is not a rate. It expresses the average amount of time required to complete the servicing of a single request.

Mean utilization (U): expressed the average amount of time the server or resource was busy during the measurement time T. Since U is a ratio of two quantities, it has not units and is offten expressed as a percentage.

## Funciones acumulativas en una cola




- A(t) Función acumulativa de llegada: Número de llegadas en [0,t]
- D(t) Función acumulativa de salida; Número de clientes servidos en [0,t]
- $Q(t) := A(t) - D(t)$ is the backlog (unfinished work) at time $t$.
- $d(t) = \min\{u \geq 0 : A(t) \leq (D(t + u)\}$ *Tiempo de respuesa a un cliente que llega en t si la política de atención es FIFO.*



Notas:

Consider a queue which is viewed as a black box. We make no specific assumptions about its operation; it may be a network node, an information system, etc. The cumulative functions are:

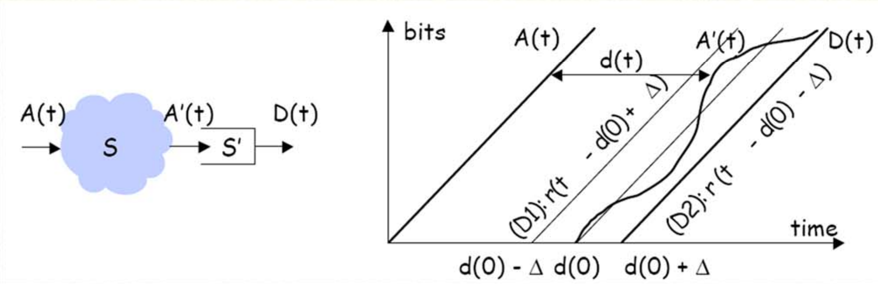
- $A(t)$ (*input function*): amount of work that arrives into the system in the time interval $[0, t]$
- $D(t)$ (*output function*): amount of work done in the time interval $[0, t]$

Assume that there is some time $t0 \leq 0$ at which $A(t0) = D(t0) = 0$. We interpret $t0$ as an instant at which the system is empty. The main observations are:

- $Q(t) := A(t) - D(t)$ is the backlog (unfinished work) at time $t$.
- Define $d(t) = \min\{u \geq 0 : A(t) \leq (D(t + u)\}$ (horizontal deviation on Figure 8.1). If there is no loss of work (no incoming item is rejected) and if the system is first in, first out (*FIFO*), then $d(t)$ is the response time for a hypothetical atom of work that would arrive at time $t$.

## Ejemplo de aplicación de las curvas acumulativas

- Una red de comunicación transfiere señal de vídeo. La entrada a la red es a frecuencia r (paquetes/s) constante. La red introduce un retraso aleatorio pero con jitter acotado por Δ.
- Para conseguir la reproducción correcta se requiere consumir los paquetes a la misma frecuencia constante.
- La solución es introducir un buffer para eliminar el jitter.




- Se debe establecer un retraso d(0)+2Δ.
- La longitud del buffer debe ser Q= r x (d(0)+2 Δ)



Notas:

PLAYOUT BUFFER. Consider a packet switched network that carries bits of information from a source with a constant bit rate *r* (Figure 8.2) as is the case for example, with circuit emulation. We have a first system *S*, the network, with input function $A(t) = rt$. The network imposes some variable delay, because of queuing points, therefore the output *A*() does not have a constant rate *r*. What can be done to re-create a constant bit stream ? A standard mechanism is to smooth the delay variation in a playout buffer. It operates as follows. When the first bit of data arrives, at time $d(0)$, it is stored in the buffer until some initial delay has elapsed. Then the buffer is served at a constant rate *r* whenever it is not empty. This gives us a second system *S*, with input *A*() and output *D*(). What initial delay should we take ? We give an intuitive, graphical solution.

The second part of the figure shows that if the variable part of the network delay (called *delay jitter*) is bounded by some number Δ, then the output $A(t)$ is bounded by the two lines (D1) and (D2). Let the output $D(t)$ of the playout buffer be the function represented by (D2), namely $D(t) = rt - d(0) - \Delta$. This means that we read data from the playout buffer at a constant rate *r*, starting at time $d(0) + \Delta$. The fact that $A(t)$ lies above (D2) means that there is never underflow.

Thus the playout buffer should delay the first bit of data by an amount equal to Δ, a bound on delay jitter.

## Fórmula de LITTLE

- Es una ecuación entre magnitudes de performance que se verifica para régimen estacionario (magnitudes promedio) y con independencia de los tipos de distribución probabilística de acceso o de servicio:
- El número medio de clientes en una cola (Q) es en régimen estacionario igual al producto de la frecuencia de acceso (λ) por el tiempo de estancia (R):

$$Q = \lambda\ R$$

- O equivalentemente, igual al producto del throughput (X) por el tiempo de residencia (R).

$$En\ régimen\ estacionario\ X = \lambda \quad \Rightarrow \quad Q = \lambda\ R = X\ R$$



Notas:

One of the most commonly used theorems in queueing theory is Little's law, which allows us to relate the mean number of jobs in any system with the mean time spent in the system as follows:

Mean number in the system = arrival rate × mean response time

This relationship applies to all systems or parts of systems in which the number of jobs entering the system is equal to those completing service. Little's law, which was first proven by Little (1961), is based on a black-box view of the system. The law applies as long as the number of jobs entering the system is equal to those completing service, so that no new jobs are created in the system and no jobs are lost forever inside the system. Even in systems in which some jobs are lost due to finite buffers, the law can be applied to the part of the system consisting of the waiting and serving positions because once a job finds a waiting position (buffer), it is not lost. The arrival rate in this case should be adjusted to exclude jobs lost before finding a buffer. In other words, the effective arrival rate of jobs entering the system should be used.

## La ley de Littel se corresponde con nuestra experiencia

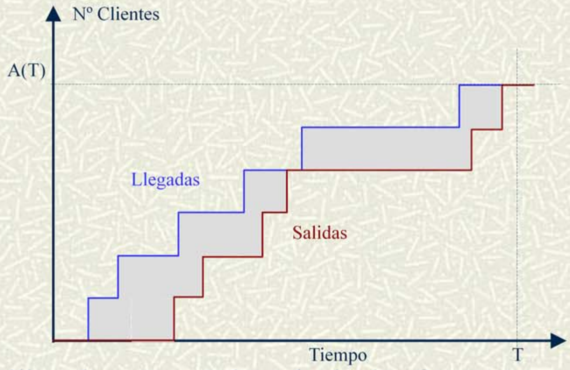
- Cuando en un día lluvioso se incrementa el tiempo de tránsito de cada coche (se conduce mas despacio), la calle se llena de coches, aunque el tráfico sea el mismo.
- En un comedor de comidas rápidas, el espacio que se necesita para atender el mismo número de clientes es menor que en un restaurante en el que se come con tranquilidad.



Notas:

## Demostración de la fórmula de Little

- Considérese una cola en la que se analiza una ventana en la que todos los clientes que han llegado se han atendido (esto es lo que se puede considerar válido en régimen estacionario). En ella se observan las siguientes entradas y salidas.



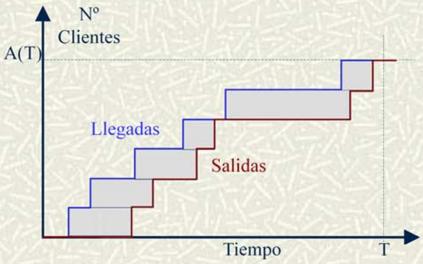

- Vamos a contabilizar la zona sombreada de dos formas: por rectángulos horizontales y por rectángulos verticales.



Notas:

## Demostración fórmula de Little (2)

- Contabilidad horizontal




| Anchura | Altura | Area |
|---|---|---|
| 4,5 | 2,0 | 9,0 |
| 4,5 | 2,0 | 9,0 |
| 4,5 | 2,0 | 9,0 |
| 2,0 | 2,0 | 4,0 |
| 5,0 | 2,0 | 10,0 |
| 2,0 | 2,0 | 4,0 |
| 22,0 | 12,0 | 44,0 |

$$Area = \sum rectangulos = 44$$

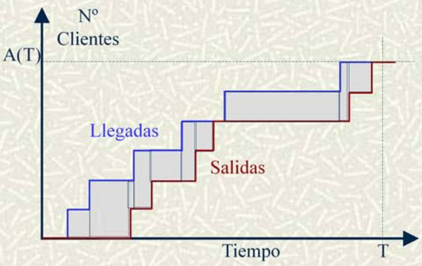
$$Area = 6 \times Altura\ Media \times Anchura\ Media = 6 \times \frac{A}{6} \times Tiempo\ estancia\ medio = A \times R$$



Notas:

## Demostración fórmula de Little (3)

Suma vertical:




| Anchura | Altura | Area |
|---|---|---|
| 1,0 | 2,0 | 2,0 |
| 3,0 | 4,0 | 12,0 |
| 0,5 | 2,0 | 1,0 |
| 1,0 | 4,0 | 4,0 |
| 2,5 | 2,0 | 5,0 |
| 1,0 | 4,0 | 4,0 |
| 1,0 | 2,0 | 2,0 |
| 4,5 | 2,0 | 9,0 |
| 0,5 | 4,0 | 2,0 |
| 1,5 | 2,0 | 3,0 |
| 16,5 | 28,0 | 44,0 |

$$Area = \sum rectangulos = 44$$

$$Area = 13 \times Altura\ Media \times Anchura\ Media = 13 \times Anchura\ media \times Altura\ media = T \times Q$$

Igualado ambas sumas:

$$Area = R \times A = T \times Q \quad \Rightarrow \quad Q = \frac{A}{T} \times R = \lambda \times R$$



Notas:

## Tanto por ciento de utilización del servidor

- Teniendo en cuenta la definición de:
  - % Utilización: U=B/T = Tiempo usando servicio/Tiempo medida
  - Throughput: X= C/T= Nº de clientes servidos/Tiempo medida
  - Tiempo servicio: S=B/C= Tiempo usando serv./Nº clientes servidos
- Se verifica:

$$U = \frac{B}{T} = \frac{C}{T} \times \frac{B}{C} = X \times S$$

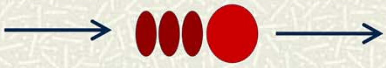
- Ejemplo: *Considérese un disco SCSI que es utilizado con una tasa de 50 operacionesIO/s por una aplicación. Si el tiempo medio de cada operación de lectura es de 10 ms. Cual es el tanto por ciento de utilización:*

$$U = X \times S = 50(oper/s) \times 0.01(s) = 0.5 = 50\,\%\ utilización$$



Notas:

## Ecuaciones para una cola simple



- Tiempo de residencia R:

$$R = S + S \times Q = S \times (1+Q) = S \times (1+XR) \Rightarrow \boxed{R = \frac{S}{1-SX} = \frac{S}{1-U}}$$

- Número medio de clientes en la cola Q

$$R = \frac{S}{1-U} \xrightarrow[\times X]{} \boxed{Q = \frac{U}{1-U}}$$

- Tiempo medio de espera W

$$Q = \frac{U}{1-U} \xrightarrow[\times S]{} \boxed{W = Q \times S = \frac{S \times U}{1-U}}$$

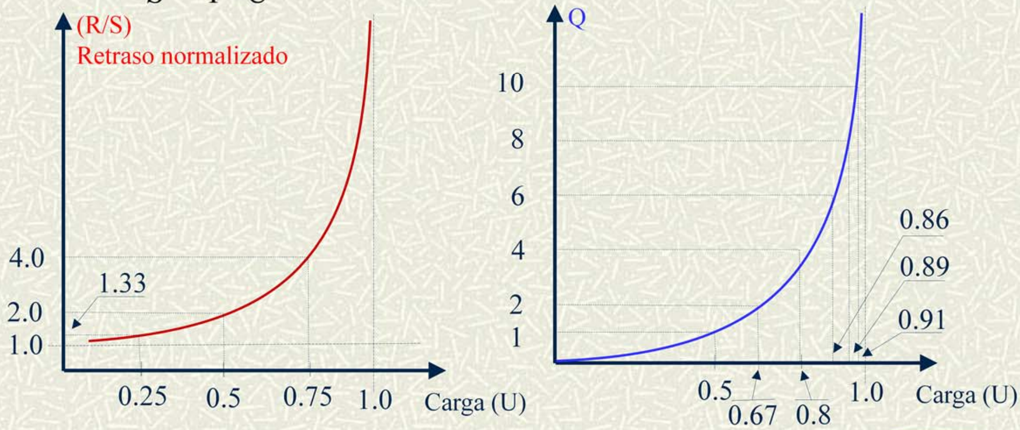


Notas:

## Retraso normalizado frente a la carga

- El retraso normalizado por el tiempo de servicio, frente a la carga (% utilización del servicio) es una hipérbola equilátera:

$$\frac{R}{S} = \frac{1}{1-U} \qquad Q = \frac{U}{1-U}$$






Notas:

## Ejemplo: Respuesta de un puerto de una red

- Las medidas sobre el puerto de acceso a una red, revela que los paquetes llegan al puerto con una tasa de 125 paquetes/s, y tarda aproximadamente 2 ms en ser transmitidos.

  Características del puerto que se puede obtener de esta información son:
  - Throughput => X=125 pack/s
  - Tiempo de servicio => S= 2 ms

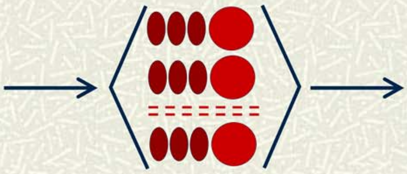
  - Utilización de la red (U=XS) => U=0,25 = 25 %
  - Tiempo medio de permanencia en puerto R=S/(1-U) => R= 2.66 ms
  - Número medio de paquetes en el puerto Q=U/(1-U) => Q=0.33 paquetes



Notas:

## Tiempo de residencia en un sistema con N colas paralelas

p(M/M/m)



- Se considera un sistema que atiende un throughput X que es atendido por p servicios dotados con su respectiva cola. Los tiempos de servicio (S) de las N colas son idénticos.

  En este caso la longitud de las colas se reduce en un factor p:

$$R = S + S \times \left(\frac{Q}{p}\right) \Rightarrow R = \frac{S}{1-\frac{X \times S}{p}} \Rightarrow R = \frac{S}{1-\frac{U}{p}} = \frac{S}{1-\rho} \quad \left(siendo\ \rho = \frac{U}{p}\right)$$

- ρ es la utilización U dividido por el número de servidores, puede varia en el rango 0< ρ <1, y representa la carga normalizada del sistema. El resultado se puede interpretar como si todo el flujo pasara secuencialmente por los N servidores, pero requiriendo sólo un tiempo de servicio S/N.



Notas:

The figure show the flow of identical customers into a q-parallel queueing center where each queue has its own server and each server has the same service time S. The stream of arrivals is split into two separa streams. Therefore from the wiewpoint of a newly arriving customer, each queue appears shorter (by half) than it would if there was just a single queue center.

The only difference with the simple service case is the factor p appearing with the utilization. When the total utilization is divided by the number of servers, it is called the server utilization and is denoted ρ. Since 0< ρ<1, it represents the probability that the server is busy.

## Ejemplo de Servicio WEB

- Un servicio de atención por WEB distribuye su trabajo entre los 10 técnicos que atienden los clientes utilizando una estrategia de intercambios de mensajes en sesiones de *chatting*. La asignación de clientes a técnicos se hace aleatóriamente en la llamada y se mantiene asignado durante la sesión.

  Se mide las sesiones de atención, y se comprueba que:
  - Se atienden clientes con una tiempo medio entre llamadas de 3 min.
  - Cada cliente espera aproximadamente 3 minutos antes de que sea atendido.
  - Y mantiene el dialogo con el técnico durante un tiempo medio de 2 min.

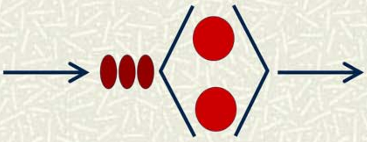
  ¿En cuantos técnicos hay que ampliar la plantilla si se quiere reducir los tiempos de espera de los clientes a sólo 1 min?.

| Situación actual | | | | Nueva sit. |
|---|---|---|---|---|
| Parámetro | Valor | | | Valor |
| Número de servicos (N) | 10 | | | 18 |
| Throughput (X) | 3,0 | CPM | | 3,0 |
| TiempoServicio (S) | 2,0 | s | | 2,0 |
| Utilización (U=XS) | 600,0 | % | | 600,0 |
| Carga normalizada (ρ=U/N) | 60,0 | % | | 33,3 |
| Tiempo de residencia (R=S/(1-U)) | 5,0 | s | | 3,0 |
| Tiempo de espera(W=R-S) | 3,0 | s | | 1,0 |



Notas:

## Sistema con 2 servidores que atiende una cola



- En este caso el tiempo de residencia se acorta por dos razones:
  - La existencia de 2 servers acorta los tiempos de servicios a la mitad
  - En los tiempos de servicio debe repercutirse la probabilidad de que alguno de los servidores esté ocupado. Este peso es exactamente ρ=(U/2).

Q=XR  SX=U

$$R = S + \frac{1}{2} S \times \rho \times Q = S + \frac{S \times X}{2} \times \rho \times R = S + \rho^2 R \quad \Rightarrow \quad \boxed{R = \frac{S}{1-\rho^2}}$$

  - De igual modo se puede deducir que el número medio de clientes en la cola Q es :

$$\boxed{Q = \frac{2\rho}{1-\rho^2}}$$

Dec'11: III- Queue model José M.Drake 25

Notas:

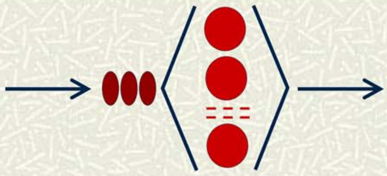
Suppose we add an identical server to a common queue. Nothing else is changed. With two servers available to service customers, we expect the mean residence time to be reduced. The question is, by how much?.

Naively, you might expect the residence time is halved because the capacity of the uniserver has been doubled. Things are more subtle than that, however, by virtue of what happens during the time you arrive and wait for service. When you finally reach one of the server , it still takes the same time S to be serviced. Like the twin center example, thos aspect is not different from the uniserver case. It's what happens ahead of you in the queue that is diferent. The shorter residence time (as been by you) comes from a shorter effective service time for those customers ahead of you. The effective service time is composed of two factors:

- The existence of anhother server halves the service time.
- The service time is weighted by the probability that a server is busy.

The first factor corresponds to the expected service time S/2 as in twin queueing center. The second factor is new. The probability that the server is busy is given the server utilization ρ=U/2. During those periods when both servers are busy , the queue length will grow. In this sense, the effective server time also depends on the load ρ. Combining these two factors, we write the effective service time as S(ρ)=(S/2) ρ.

## Sistema con N servidores que atiende una cola



- En este caso el tiempo de residencia se acorta por dos razones:
  - La existencia de m server acorta los tiempos de servicios en un factor m
  - En los tiempos de servicio debe repercutirse la probabilidad de que alguno de los servidores esté ocupado. Este peso **se puede aproximar** por $\rho^{m-1}=(U/m)^{m-1}$.

$$R = S + \frac{1}{m} S \times \rho^{m-1} \times Q = S + \frac{S \times X}{N} \times \rho^{m-1} \times R = S + \rho^{m} R \quad \Rightarrow \quad \boxed{R = \frac{S}{1-\rho^{m}}}$$

  - De igual modo se puede deducir que el número medio de clientes en la cola Q es :

$$\boxed{Q = \frac{m\rho}{1-\rho^{m}}}$$

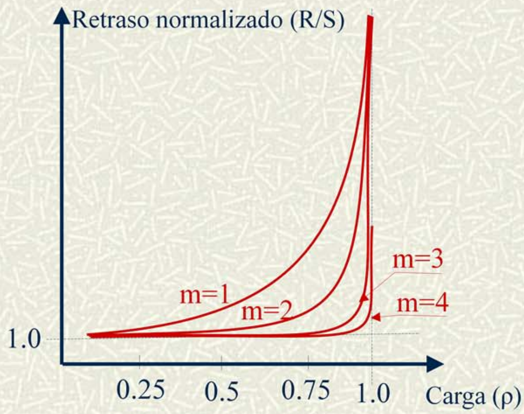


Notas:

## Tiempo de respuesta normalizado frente a la carga

- El retraso normalizado por el tiempo de servicio, frente a la carga (% utilización del servicio) ya no es una hipérbola equilátera:

$$\frac{R}{S} = \frac{m}{1-\rho^m}$$






Notas:

The figure show the relative effect on response time, as more servers are added to an M/M/m system. The general trend is to push the knee of the curve toward the lower right cornet of the plot. The uppermost curve corresponds to the single server case.

## Calculo exacto de la respuesta de sistema multi servicio

- El cálculo intuitivo de la probabilidad de que alguno de los servidores esté ocupado $\rho^{m-1}$ no es exacta. El resultado exacto para el caso de la cola M/M/m lo describe la función C Erlang:

$$C(m,\rho)=\frac{\frac{(m\rho)^m}{m!}}{(1-\rho)\sum_{k=0}^{m-1}\frac{(m\rho)^m}{k!}+\frac{(m\rho)^m}{m!}}$$

- El tiempo de estancia que resulta es en este caso

$$R=S\left(1+\frac{C(m,\rho)}{m(1-\rho)}\right)$$

- Para el caso de una cola M/M/m/0 (sin contención) lo describe la función B de Erlang

$$B(m,\rho)=\frac{\frac{(m\rho)^m}{m!}}{\sum_{k=0}^{m-1}\frac{(m\rho)^m}{k!}}$$



Notas:

## Comparación de los calculos de los tiempos de estancia normalizados

Tiempo de respuesta normalizada (aproximada) (exacta)

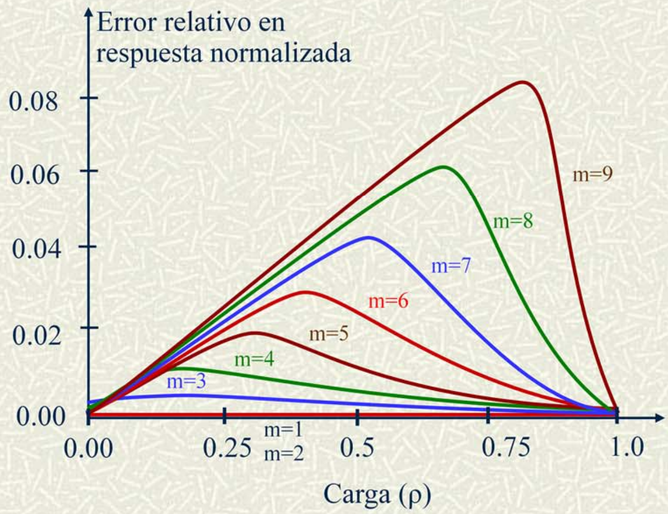
| m,ρ | 0,333 | 0,5 | 0,666 | 0,75 | 0,9 |
|---|---|---|---|---|---|
| 1 | 1,499 | 2,000 | 2,999 | 4,000 | 10,000 |
| 1 | 1,499 | 2,000 | 2,999 | 4,000 | 10,000 |
| 2 | 1,125 | 1,333 | 1,800 | 2,286 | 5,263 |
| 2 | 1,125 | 1,333 | 1,800 | 2,286 | 5,263 |
| 4 | 1,012 | 1,067 | 1,246 | 1,463 | 2,908 |
| 4 | 1,019 | 1,087 | 1,284 | 1,509 | 2,969 |
| 6 | 1,001 | 1,016 | 1,096 | 1,216 | 2,134 |
| 6 | 1,004 | 1,033 | 1,142 | 1,281 | 2,233 |
| 8 | 1,000 | 1,004 | 1,041 | 1,111 | 1,756 |
| 8 | 1,001 | 1,015 | 1,083 | 1,178 | 1,877 |



Notas:

The approximate equation generally tends to understimated the exact response time because it is missing some of the coefficients in the Erlang C formula. The relative error as plotted as a surfacein the next page and as a table in this page. For light loads ($\rho$<0.33) is a excelent approximation.The maximum error being less than 1% at m=4 servers. Under heavy loads ($\rho$~0.9) the error increases from about 1% at 3 servers to 10% at 32 servers

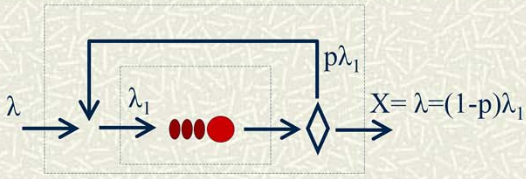
## Errores entre los tiempo de respuesta aproximado y exacto






Notas:

## Colas con realimentación




$$\lambda_1 = \lambda + p\lambda_1 \quad \Leftrightarrow \quad \lambda_1 = \frac{\lambda}{1-p} \qquad \Rightarrow \quad U = \lambda_1 S = \frac{\lambda\, S}{1-p}$$

$$\textit{Factor incremento visitas } V = \frac{1}{1-p}$$

$$\textit{Demanda media efectiva } D = V\ S$$

$$R = \frac{D}{1-\lambda D} = \frac{\frac{S}{1-p}}{1-\frac{\lambda S}{1-p}}$$



Notas:

So far, we have been discussing queueing center where a customer arrives at random from external source, queues for service, , receives service, and then departs the center, never tu return. Clearly, there are cases in which a customer who has already received service retuns for futher service: a customer forgot to purchase a item and must return to the grocery store, children form a line for repeat slides in a playground; packets must be retrasmitted on a communication network and so on. This effect is called feedback.

A general feedback mechanism of this type is represented in a simple queueing center . External arrivals occur at a rate λ. Customers who have received service return to the quue with branching probability p. The stream of retiurning customers $p\lambda_1$ combines with new arrivals λ such tha effective arrival rate at the queue is $\lambda_1 = \lambda + p\,\lambda_1$.

## Ejemplo de canal de comunicación con tasa de fallos

- Considérese una canal de comunicación por el que se transmite un paquete cada 2 s, y que la transmisión requiere un tiempo de 0,75 s. El canal presenta una tasa de fallo del 30%, y los mensajes fallidos tienen que ser retransmitidos.

| | | | |
|---|---|---|---|
| Tasa de acceso al sistema | $\lambda$= | 0,5 | mensajes/s |
| Tiempo de transmisión | S= | 0,75 | s |
| Probabilidad de fallo | p= | 0,3 | |
| Tasa efectiva de transmisión $\lambda_1=\lambda/(1-p)$ | $\lambda_1$= | 0,714 | mensajes/s |
| % de utilización $U=\lambda_1 S$ | U= | 0,536 | |
| Tiempo de espera $W=U\,S/(1-U)$ | W= | 0,866 | s |
| Tiempo de servicio en la cola $R_1=W+S$ | $R_1$= | 1,818 | s |
| Factor incremento de visitas $V=1/(1-p)$ | V= | 1.429 | |
| Tiempo de servicio en el sistema $R=VR_1$ | R= | 2.308 | s |

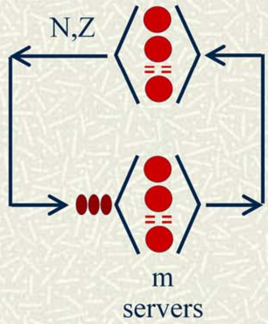


Notas:

## Colas con número finito de clientes

- Hay muchos sistemas que procesan los requerimientos de un número finito de clientes que pueden considerarse como sistemas cerrados. Tienen una gran importancia porque representan estructuras de auto regulación.

- Se consideran que hay N clientes que tras salir de la cola permanecen fuera durante un tiempo medio Z, y luego se reintegran a la cola de nuevo
- Razonamiento:
  - Si los N clientes están en la cola, se alcanza la máxima ocupación, y en este estado no puede incrementarse porque no hay mas clientes.
  - Caben congestiones mas bajas n=0,1,2,3,..,N. Y en función de ellas se pueden expresar el throughput medio X(n) y el tiempo de respuesta medio R(n).
- La ventaja es que es una máquina de estados finita, y por tanto analizable. Aunque como N puede ser muy grande su análisis puede ser muy tedioso (en la práctica imposible).





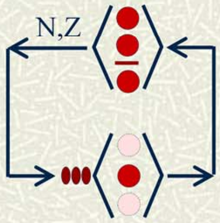

Notas:

The constraint on the number of customers is tantamount to a form of negative feedback or self-regulation. The finite population of N customers is either preparing to join the queue (this is sometimes called the thinking state with think-time denoted by Z), or they are already at a quueueing center either enqueueed or eing serviced.

The feedback property is easily understood as follows. If, during some period, all N customers are at the service center , then there can not be any new arrivals at the service center. The system is maximally busy. In such circustances the response time will be at its worst. In general, as the service center gets busy, the rate at which itgets busier decreases, thus lowering any futher congestion at the service center.

## Ejemplo Gestión de un comedor

- Considérese un comedor servido por m camareros que sirven un comedor con N mesas a ser atendidas:
  - Z Tiempo de estancia del cliente. Incluye el tiempo de propiamente comer y pensar a la espera de que el camarero atienda.
  - S Tiempo de atención del camarero: Incluye el tiempo que tarda el camarero en detectar una mesa que requiere su servicio y accede a ella.
- El modelo se formula considerando los camareros como los servidores y los clientes son sólo el medio con el que se formula el problema. Cuando un cliente termina, es sustituido inmediatamente por otro equivalente.




m=1
servers

- Consideremos el caso de que sólo haya un camarero m=1. El throuhput X(N) se puede formular como:

$$X(N) = \frac{N-Q}{Z}$$

Esto representa que el throughput es igual a la frecuencia de acceso

- Aplicando la formula de Little:

$$Z\,X(N) = N - X(N)\,R \quad \Rightarrow X(N) = \frac{N}{R+Z} \quad \Rightarrow \quad R = \frac{N}{X(N)} - Z$$

- En este caso X(N) es una salida del análisis del modelo y no una magnitud estadística medida como entrada para establecer la carga



Notas:

Considérese el caso de un comedor donde cada cliente es servido por el primer camarero que está disponible de la plantilla de m camareros de que dispone. El tiempo de servicio Z representa el tiempo medio que los clientes tardan en comer un plato ( en este caso el genérico “tiempo en pensar” corresponde al específico “tiempo en comer”). Como consecuencia de que los camareros está continuamente sirviendo a algún cliente, el tiempo de servicio S corresponde al tiempo medio que tarda un camarero en descubrir que un cliente está esperando y servir la comida. Este sistema puede ser representado por un modelo de colas y analizado su comportamiento.

En la práctica cada camarero es un servidor, pero de forma diferente a la interpretación habitual, el camararo viene al cliente y no al revés. Sin embargo, este es un detalle irrelevante en término del modelo de performance. La realidad y el modelo son lógicamente equivalentes: Cada cliente solo requiere un número determinado de platos (por ejemplo 3 veces), y luego abandona el comedor, pero es inmediatamente sustituido por otro cliente que estaba esperando en la entrada. El comedor con una determinada capacidad de N puestos, representa la atención de un número finito de clientes N y puede representarse con una red cerrada de colas y una población de clientes N.

Un sistema de colas cerrado y si por simplificar suponemos que el número de camareros es m=1, se puede expresar el throughput como X(N)= (N-Q)/Z. Esra expresión simplemente representa que el throughput es una función del número de clientes que llegan al sistema, el cual ocurre con una frecuencia proporcional a la inversa de al tiempo de comer Z, reducida al numero (N-Q) de clientes que están esperando (han dejado de comer su plato).

Suponemos que no puede ser atendido nada mas que un cliente cada vez, por lo que usando la fórmula de Little (Q=XR), resulta X(N)=(N-XR)/Z , de donde se puede deducir el valor del Throughput X(N) y del tiempo de estancia medio R

## Ejemplo Computador compartido por desarrollador de aplicaciones.

- Considérese un computador que se utiliza en tiempo compartido por múltiples desarrolladores de aplicaciones para compilar sus programa. Analizando su evolución se comprueba:
  - Número de desarrolladores = 230
  - Tiempo medio entre compilaciones Z=300 s
  - Tanto por ciento de utilización de la CPU: U= 48%
  - Tiempo medio de compilación: S=0.63 s
- Cual es el througput del sistema bajo estas condiciones de carga:

$$U_{CPU} = X_{CPU} S_{CPU} \quad \Rightarrow \quad X_{CPU} = \frac{U_{CPU}}{S_{CPU}} = \frac{0.48}{0.63} = 0.7636 \quad Compilaciones / s$$

- ¿Cual es el tiempo medio efectivo de compilación?

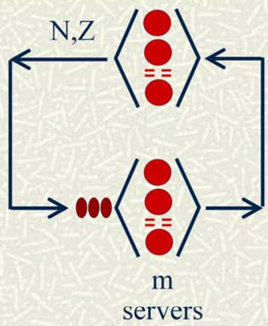
$$R = \frac{N}{X_{CPU}} - Z = \frac{230}{0,7636} - 300 = 1.21\, s$$



Notas:

## Cola con un número clientes finitos y servidor múltiple

- El calculo del throughput en un circuito cerrado con una cola de múltiple servidores es en general tediosa.
- El cálculo del tiempo máximo de respuesta (para Z=0) es fácil de calcular:




$$R = \frac{N}{X} - Z \xrightarrow[\text{por } S]{\text{Dividiendo}} \frac{R}{S} = \frac{N}{X\,S} - \frac{Z}{S} = \frac{N}{U} - \frac{Z}{S} \xrightarrow[Z=0]{U=m} \left(\frac{R}{S}\right)_{\max} = \frac{N}{m}$$

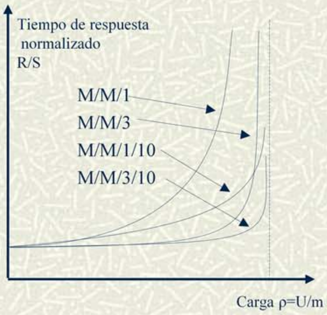


Notas:

## Cuando M/M/m/N/N se puede aproximar por M/M/m

- Ha sido propuesto que el efecto del tamaño de la cola no tiene efecto si es mayor que 10 L, siendo L=Q-m el número medio de clientes esperando en la cola.

  *Aproximación válida si* N> 10 L = 10(Q-m)




- Esto nos permite determinar la carga ρ máxima para la que la aproximación es válida.

$$L = \frac{N}{10} \qquad L = \frac{m\rho}{1-\rho^m} - m\rho \qquad \Rightarrow \quad m\rho^{m+1} + L\rho^m - L = 0$$

Ejemplo: Supóngase una cola M/M/m/N/N: En este caso la aproximación por M/M/m puede realizarse siempre que:

- N=10 => L≤1
  - m=1 => ρ≤0,618
  - m=3 => ρ≤0,688
  - m=6 => ρ≤0,752

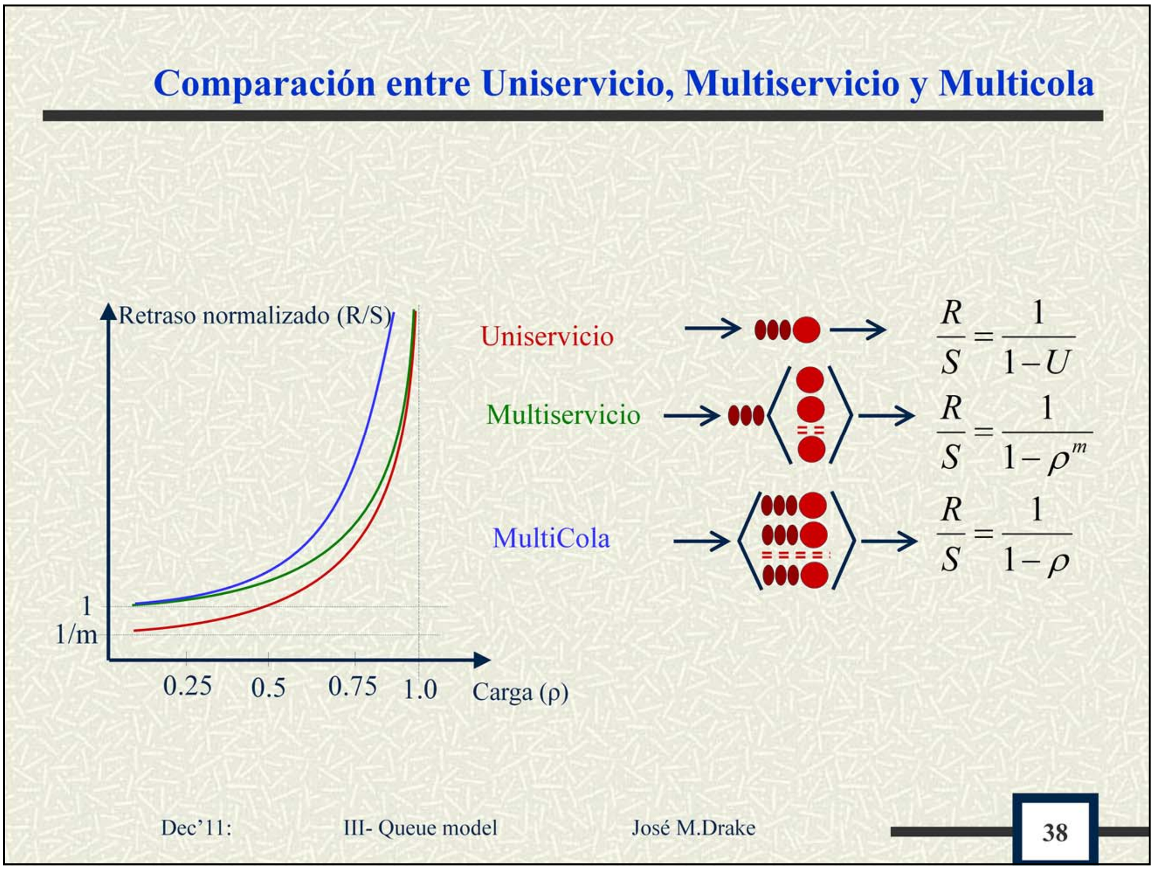
N=25 => L≤2.5
- m=1 => ρ≤0,747
- m=3 => ρ≤0,785
- m=6 => ρ≤0,82



Notas:

Se esperar que el caso de una población de N clientes en una cola M/M/m/N/N, se aproxime con la cola M/M/m ⇔ M/M/m/∞/∞ sean parecida cuando N => ∞ (el interés de la sustitución se deriva de que M/M/m tiene expresiones cerradas sencillas).

Highleymann (1989) ha demostrado que el comportamiento es equivalente si la relación entre el número de clientes de la población N y el número medio de clientes que espera en la cola Q-m, es mayor de 10.




Notas:

Si comparamos colas mutiservidores con colas mono servidores con baja carga, esto es cuando no hay encolamiento y el tiempo de servicio S es el dominante los tiempos de respuesta son siempre equivalentes. Puesto que la cola simple (univervicio) tiene un tiempo de servicio mas bajo (1/m) que el mutiservicio (1). Cuando la carga es pesada (ρ->1) el tiempo de espera se hace dominante en el tiempo de respuesta y no aparecen diferencias entre el caso uniservicio y multicola.

Si comparamos los sistemas multicola y multiservicio bajo carga baja, el comportamiento es similar. Bajo carga pesada los tiempos de servicio del sistema multicola es mucho mas alto que en el caso multiservidor. La razón es que el caso mutiservidor aprovecha mejor los posibles tiempos en los que los otros servidores no están ocupados.

Como conclusión los Uniservicios son mejores que los multiservicios, y entre estos el multiservidor es mejor que el multicola. El problema está que los tiempos de servicio del uniservicio no se pueden reducir por límites físicos, mientras que los tiempos de servicio de los multiservidores se puede disminuir incrementando el número de servidores.

¿Por que se utilizan en los bancos el multiservicio y en los supermercados la multicola?. La razón está en temas que no son tratados en este estudio. En los bancos las variabilidad de los tiempos de servicios es mayor, y esto hace que si se utilizara multicola, algunos clientes tendrían tienpos de espera muy altos. La segunda razón es logística, una única cola en un supermercado sería dificil de implementar, y se prefiere por espacio disponer de colas (espacialmente ubicadas en sitios distribuidos).

## Colas no Markovianas Relación de Pollanzek-Khinchine

- Hasta ahora muchas de las ecuaciones se han evaluado para el caso distribuciones estadísticas no markovianas.
- El efecto de una variabilidad superior o inferior a la exponencial se puede modelar considerando el tiempo de residencia se incrementa en una fracción κ que es función de la variabilidad de los tiempos de servicio:

$$\kappa = \frac{1}{2}\left(1 + COV^2\right) \qquad \begin{array}{l} M/M/1 \quad \Rightarrow COV^2 = 1 \\ M/U/1 \quad \Rightarrow COV^2 = 1/3 \\ M/G/1 \quad \Rightarrow COV^2 = 0 \end{array}$$

- Los tiempos de estancia se pueden evaluar como:

$$R_{PK} = S + SQ - (1-\kappa)\rho S = S + S\ X\ R - (1-\kappa)\rho S \quad =>$$

$$R_{PK} = \frac{S\left(1-(1-\kappa)\rho\right)}{1-\rho} = S + \frac{\rho S(1-COV^2)}{2(1-\rho)}$$



Notas:

Hasta ahora hemos supuesto que la distribución de los tiempos de servicios tiene una distribución estadística de tipo exponencial, esto se ha hecho porque en estos casos existen soluciones estadísticas. Pero en la realidad las distribuciones no son exponenciales sino que tienen otras muchas formas.

Hasta ahora se supone que un cliente que llega ve a un conjunto de clientes en la cola y uno que está siendo servido. Cada uno va a requerir un tiempo de servicio medio S. Es mas realista suponer que como consecuencia de la variabilidad respecto de la exponencial, hay una fracción de los clientes para los que el tiempo de servicio se decrementa en un factor 1-κ, siendo κ función de la variabilidad de ls distribución de los tiempos de servicio ½(1+$COV^{2)}$

## Conclusiones

- Hemos dado una panorámica sobre los elementos y las métricas que se utilizan para evaluar el comportamiento d sistema informáticos.
- Hemos tratado colas M/M/1, q(M/M/1), M/M/m, M/M/m/N/N y M/X/1
- Hemos tratado colas independientes salvo algún caso basado en M/M/1.
- Hemos tratados de hacer sólo estudios de promedios para evitar los conocimientos estadísticos. Esto tiene como consecuencia que no permite evaluar dispersiones.
- No hemos tratado los casos en los que los tiempos de servicio son función de la carga.
- No hemos estudiado los efectos de las políticas de planificación. Sólo se han estudiado FIFO.
- En el próximo capítulo estudiamos modelos basados en redes de colas bajo es estudios promedios. En el siguiente utilizaremos la aplicación de simuladores.



Notas: