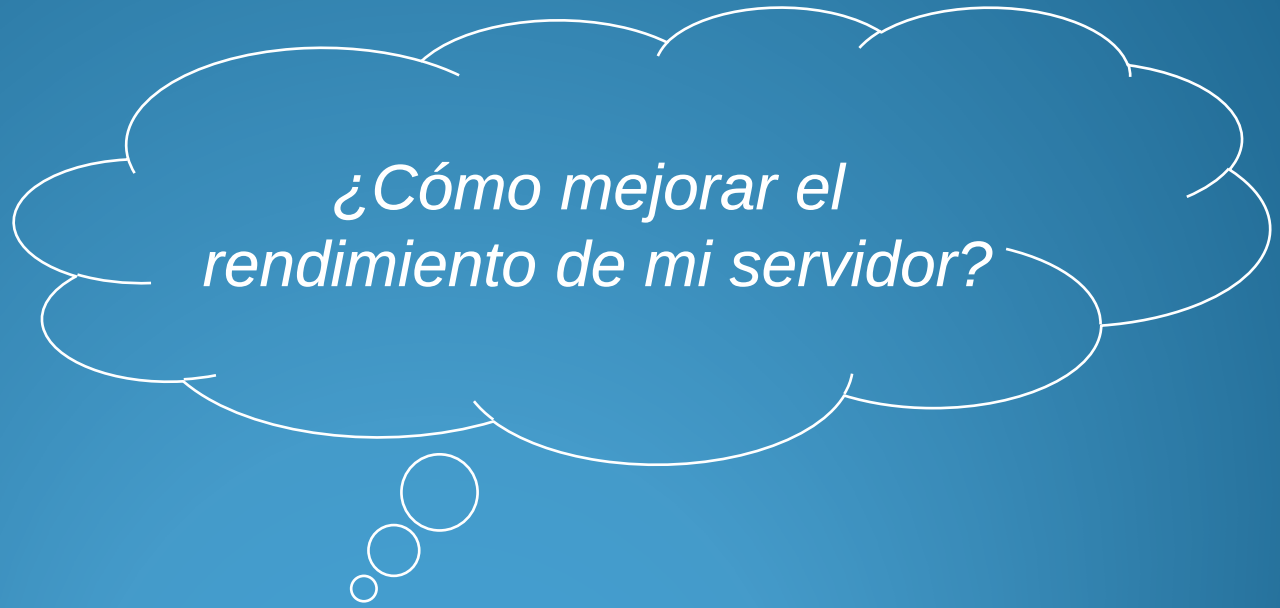
Tema 5: Optimización del rendimiento de un servidor mediante análisis operacional



Analistas, administradores y diseñadores

Objetivos del tema

- Proporcionar un modelo analítico de comportamiento de un sistema informático como punto de partida para obtener índices de rendimiento.
- Entender la importancia de los cuellos de botella como limitadores del rendimiento de los sistemas informáticos.
- Saber aplicar las leyes operacionales en ejemplos sencillos para obtener índices de rendimiento.
- Saber interpretar los límites optimistas del rendimiento que establece el análisis operacional.
- Saber evaluar de forma cuantitativa el efecto de diferentes terapias de mejora o estrategias de diseño sobre el rendimiento de un servidor.

Bibliografía

- *Evaluación y modelado del rendimiento de los sistemas informáticos*. X. Molero, C. Juiz, M. Rodeño. Pearson Educación, 2004. Capítulos 4 y 5.
- *The art of computer system performance analysis.* R.Jain. John Wiley & Sons, 1991. Capítulos 30, 32, 33 y 34.
- *Measuring computer performance: a practitioner's guide.* David J. Lilja, Cambridge University Press, 2000. Capítulo 11.

Contenido

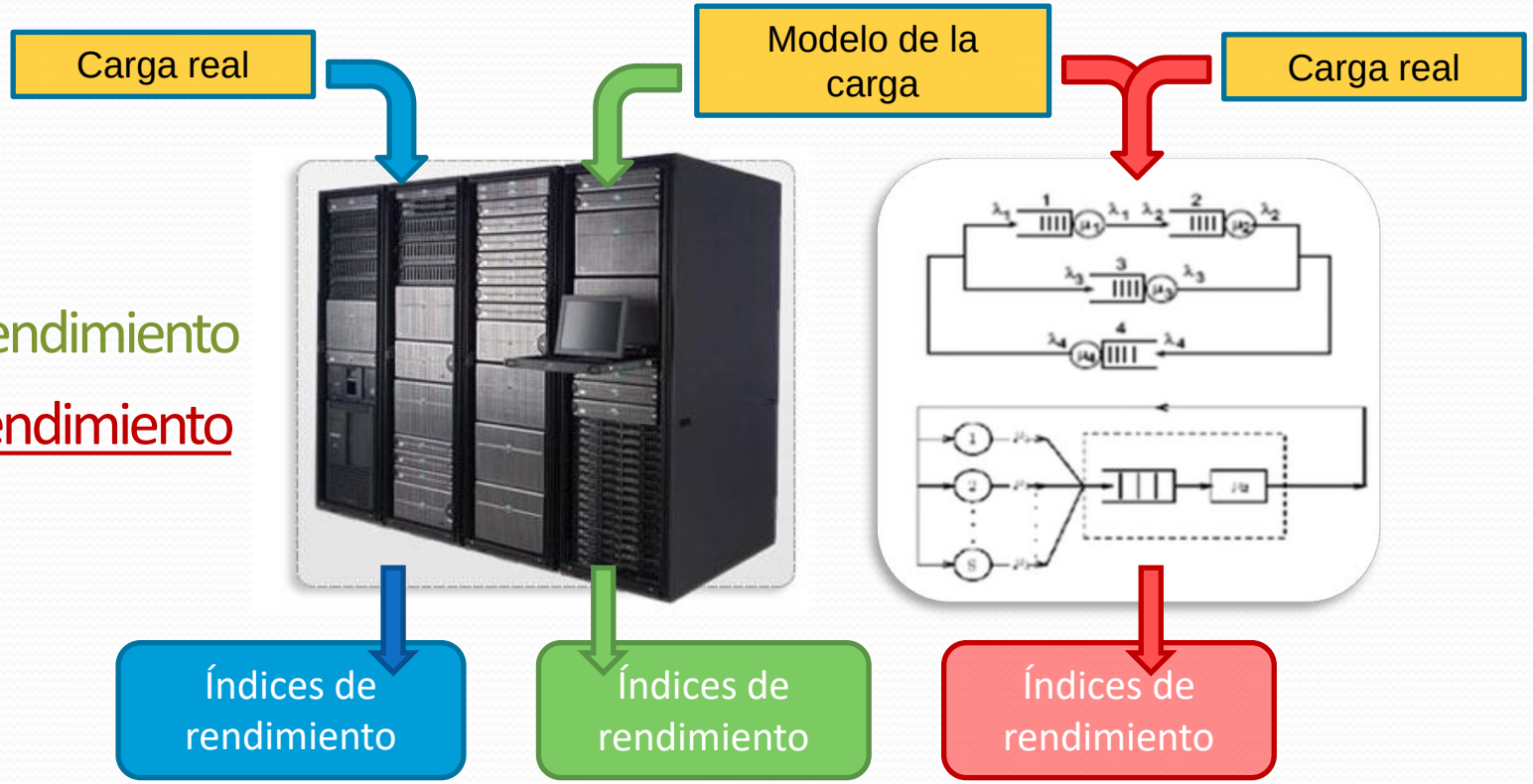
- Introducción: Redes de colas de espera.
- Variables y leyes operacionales.
- Límites optimistas del rendimiento.
- Técnicas de mejora del rendimiento.
- Algoritmos de resolución de redes de colas.

No os despistéis…

5.1. Introducción: Redes de Colas de Espera

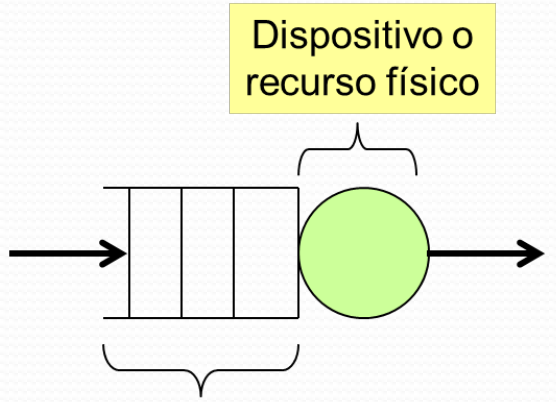
¿Cómo podemos mejorar el rendimiento de un servidor?

- Monitorización
- Comparación Rendimiento
- Optimización Rendimiento

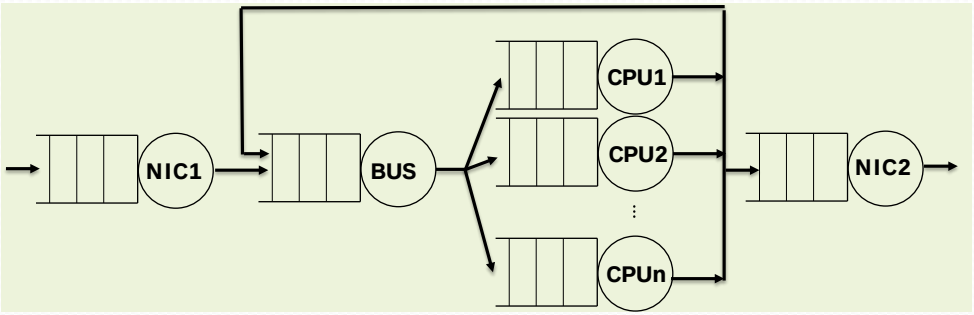


El modelo de un sistema informático

- Abstracción del sistema informático real.
  - Conjunto de dispositivos interrelacionados y trabajos que los usan (carga).
    - Dispositivos (*resources*): núcleos lógicos, unidades de almacenamiento permanente, tarjetas de red, etc.
    - Trabajos (*jobs*): procesos, accesos, peticiones, etc.
- Normalmente un dispositivo o recurso solo puede ser usado por un trabajo a la vez. El resto de trabajos tendrá que esperar.
- Modelos basados en **redes de colas** (*queueing networks*):
  - Una red de colas está formada por un conjunto de *estaciones de servicio* conectadas entre sí.
  - Estación de servicio (*service station*): Objeto compuesto por un dispositivo (recurso físico) que presta un servicio y una cola de espera para los trabajos (clientes) que demandan un servicio de él.



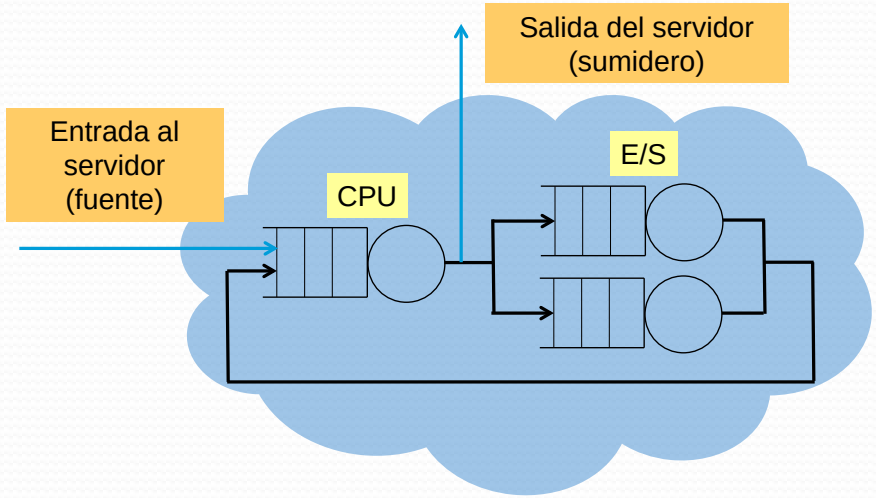
Concepto de estación de servicio

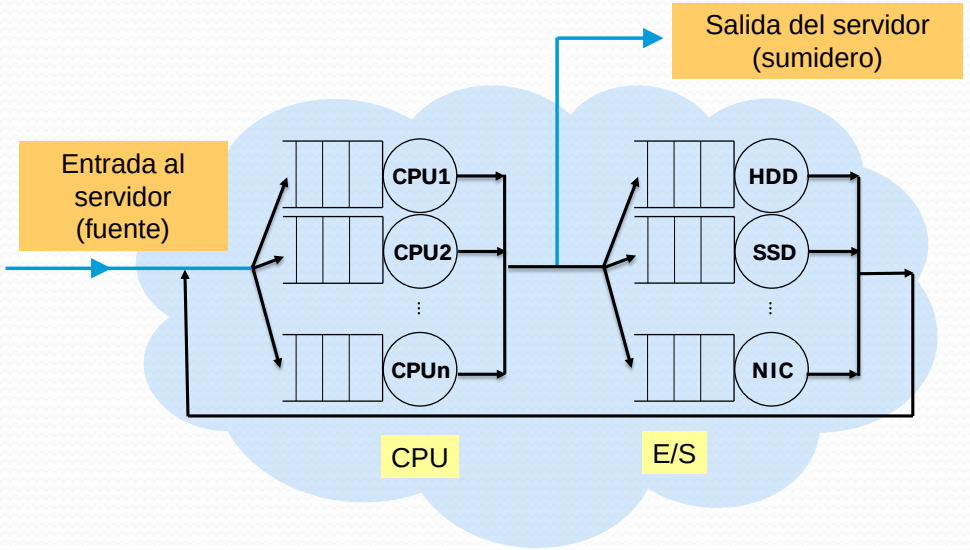


Ejemplo de red de colas

El modelo de servidor central

- Es la red de colas que más se ha utilizado para representar el comportamiento básico de los programas en un servidor de cara a extraer información sobre su rendimiento.
  - Un trabajo que "llega" al servidor comienza utilizando el procesador.
  - Después de "abandonar" el procesador, el trabajo puede:
    - terminar (sale del servidor), o bien
    - realizar un acceso a una unidad de entrada/salida (discos, red,...).
  - Después de una operación con una unidad de entrada/salida, el trabajo vuelve a "visitar" al procesador.



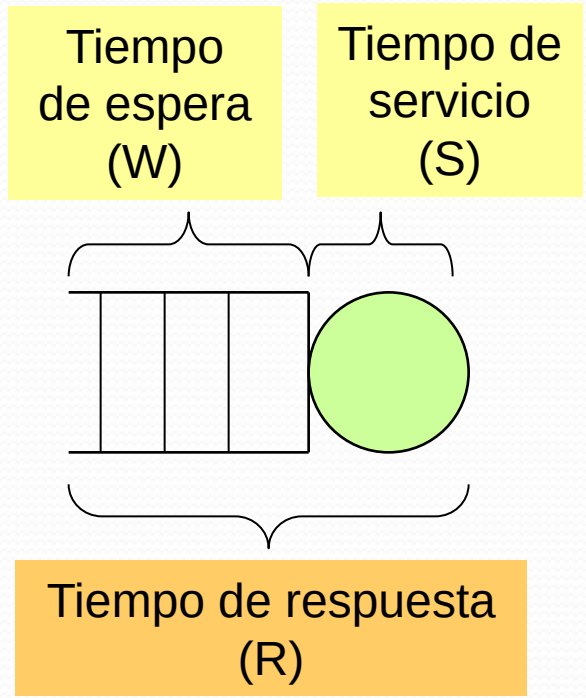


Algunas variables que caracterizan a un trabajo en una estación de servicio en un instante concreto

- Tiempo de espera en cola (W, *waiting time*)
  - Tiempo transcurrido desde que el trabajo solicita hacer uso del recurso físico (=se pone en la cola) hasta que realmente empieza a utilizarlo.
- Tiempo de servicio (S, *service time*)
  - Desde que el trabajo accede al recurso físico hasta que lo libera (=tiempo que tarda el recurso físico en procesar el trabajo).
- Tiempo de respuesta (R, *response time*)
  - Suma de los dos tiempos anteriores.

$$R = W + S$$

- Recopilando estas medidas para múltiples trabajos, podemos saber cómo se distribuyen estas variables en esa estación de servicio.



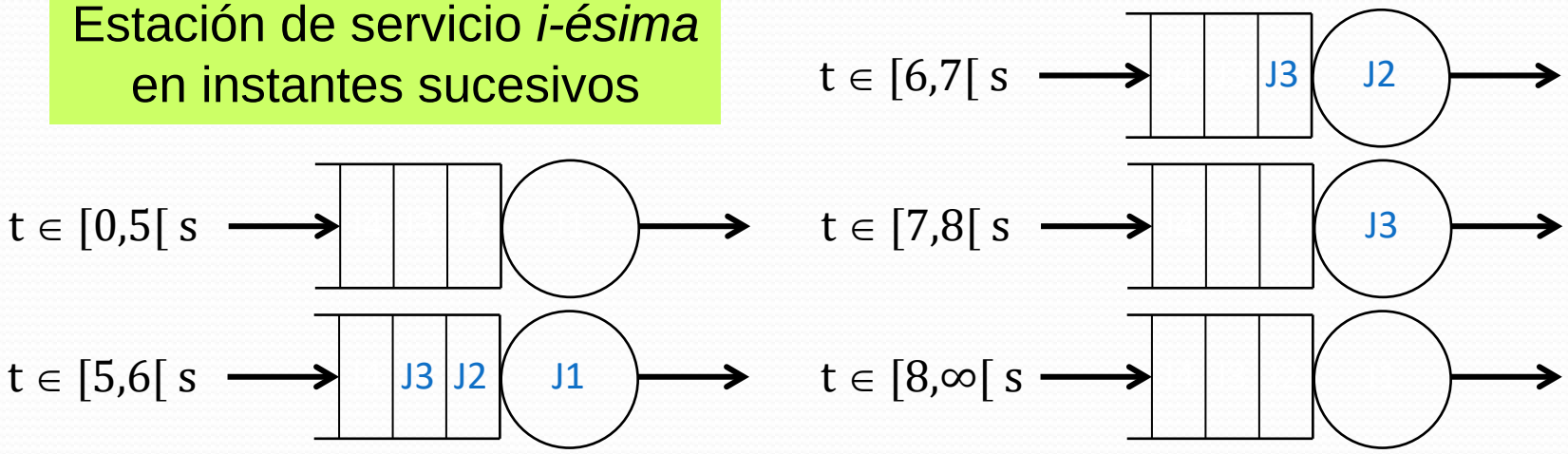
Ejercicio

Suponga que la estación de servicio i-ésima de una red de colas tiene un tiempo de servicio constante S_i=1s. Suponga que los trabajos (*jobs*) llegan con la siguiente distribución temporal:

- Durante los primeros 5 segundos no llega ningún trabajo.
- En t=5s llegan 3 trabajos: J1, J2 y J3 (por ese orden).

Calcule los tiempos de espera en la cola y los tiempos de respuesta que experimentan **cada uno** de los tres trabajos. Calcule finalmente los valores medios de W y R.

Estación de servicio *i-ésima* en instantes sucesivos

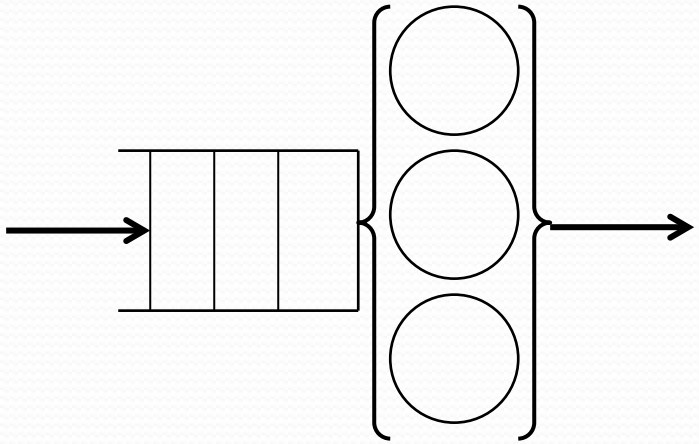


$W_i(J1) = 0s$ $W_i(J2) = 1s$ $W_i(J3) = 2s$ Valor medio $W_i = 1s$

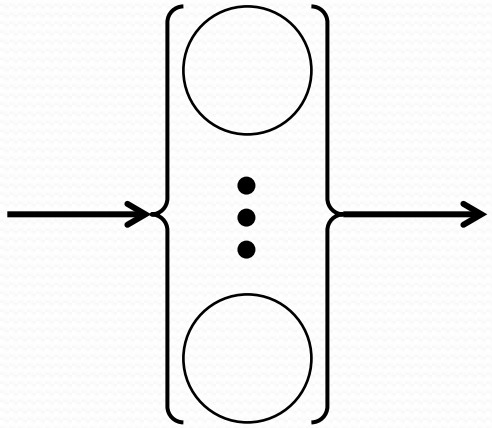
$R_i(J1) = 1s$ $R_i(J2) = 2s$ $R_i(J3) = 3s$ Valor medio $R_i = 2s$

Estaciones con más de un servidor

- Son capaces de atender a más de un trabajo en paralelo:



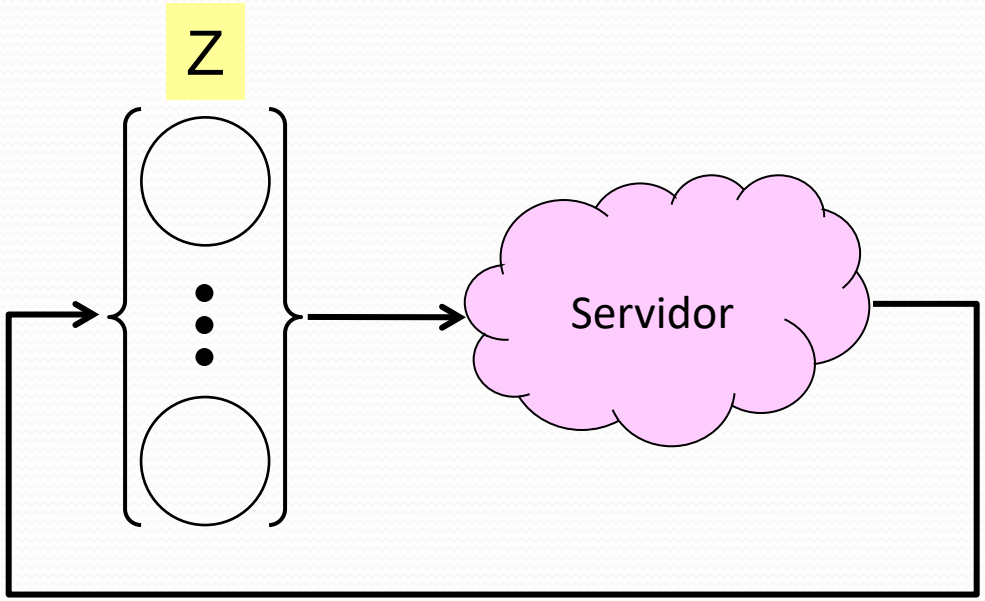
3 dispositivos
idénticos compartiendo
la misma cola de espera



Infinitos dispositivos:
no hay espera en cola.
R=S.
Estación tipo *retardo*

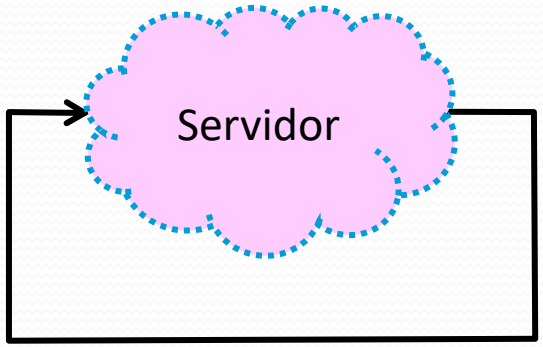
El tiempo de reflexión (Z, *think time*)

- Es un parámetro (Z) que representa el tiempo que requiere el cliente antes de volver a lanzar una petición al servidor tras la respuesta de éste.
- Se suele modelar mediante una estación de servicio tipo retardo con un tiempo de servicio = Z. Para ello, realizamos una hipótesis adicional: **cada cliente envía un único trabajo al servidor.**

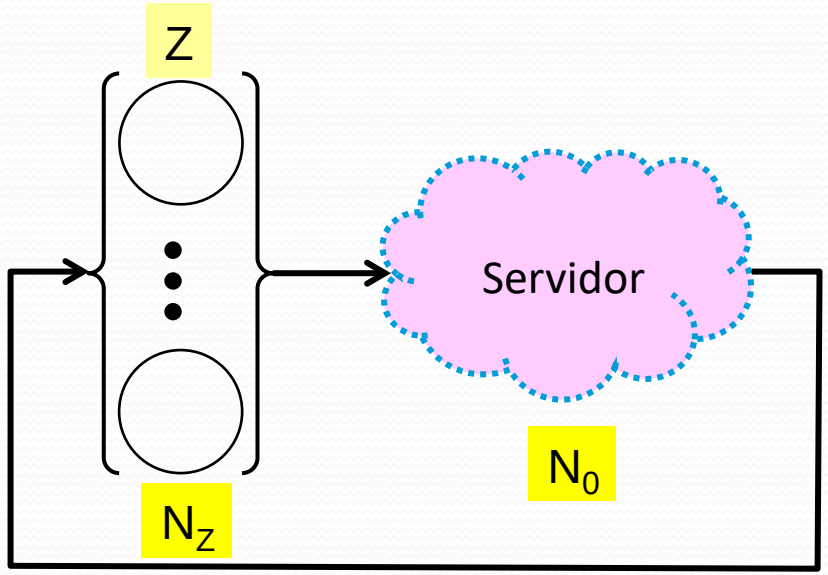


Redes de colas cerradas

- Presentan un número constante de trabajos que van **recirculando** por la red (N_T). Dependiendo de si hay o no interacción con los clientes se distingue entre redes de tipo **batch** (por lotes) o redes **interactivas**.



Red cerrada tipo batch
$N_T = N_0$

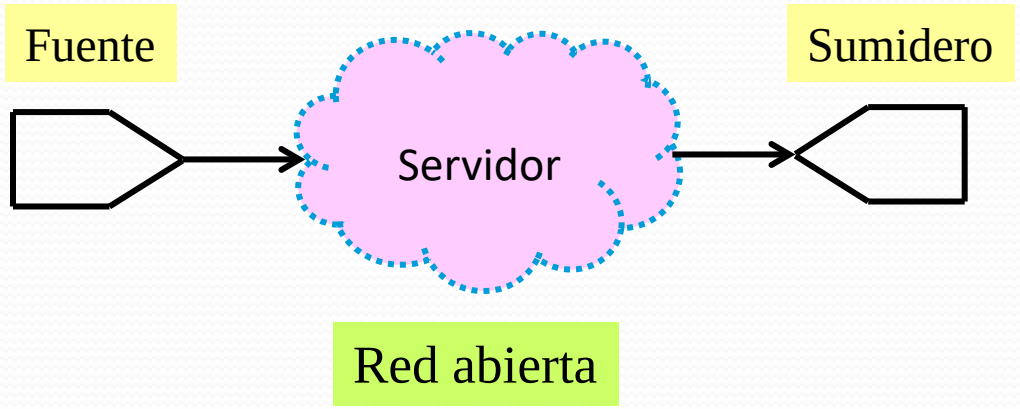


Red cerrada tipo interactivo
$N_T = N_0 + N_Z$

- Siempre supondremos 1 cliente = 1 trabajo.
  - N_o = Número de trabajos en el servidor = Número de clientes conectados al servidor.
  - N_Z = Número de clientes en reflexión = Número de trabajos en "reflexión" (= esperando a que los clientes vuelvan a introducirlos en el servidor).

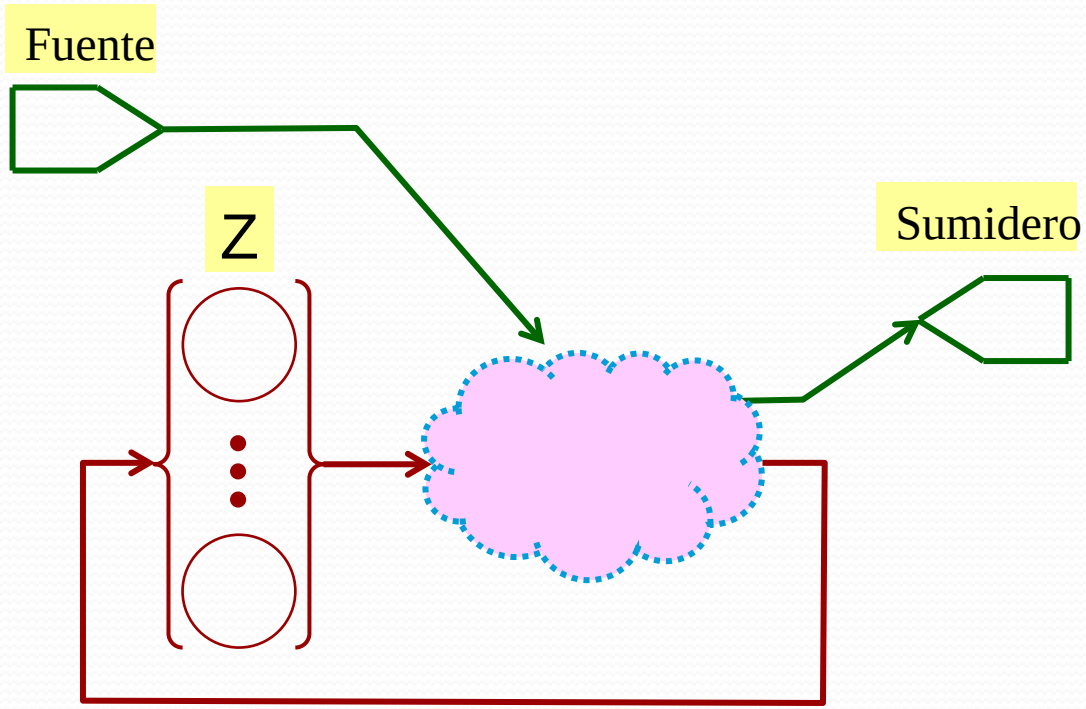
Redes de colas abiertas

- Los trabajos llegan a la red a través de una fuente externa que no controlamos. Tras ser procesados, salen de ella a través de uno o más sumideros. **No existe realimentación entre sumidero y fuente.**



Redes mixtas

- Cuando el modelo no corresponde a ninguno de los dos anteriores.



5.2. Variables y leyes operacionales

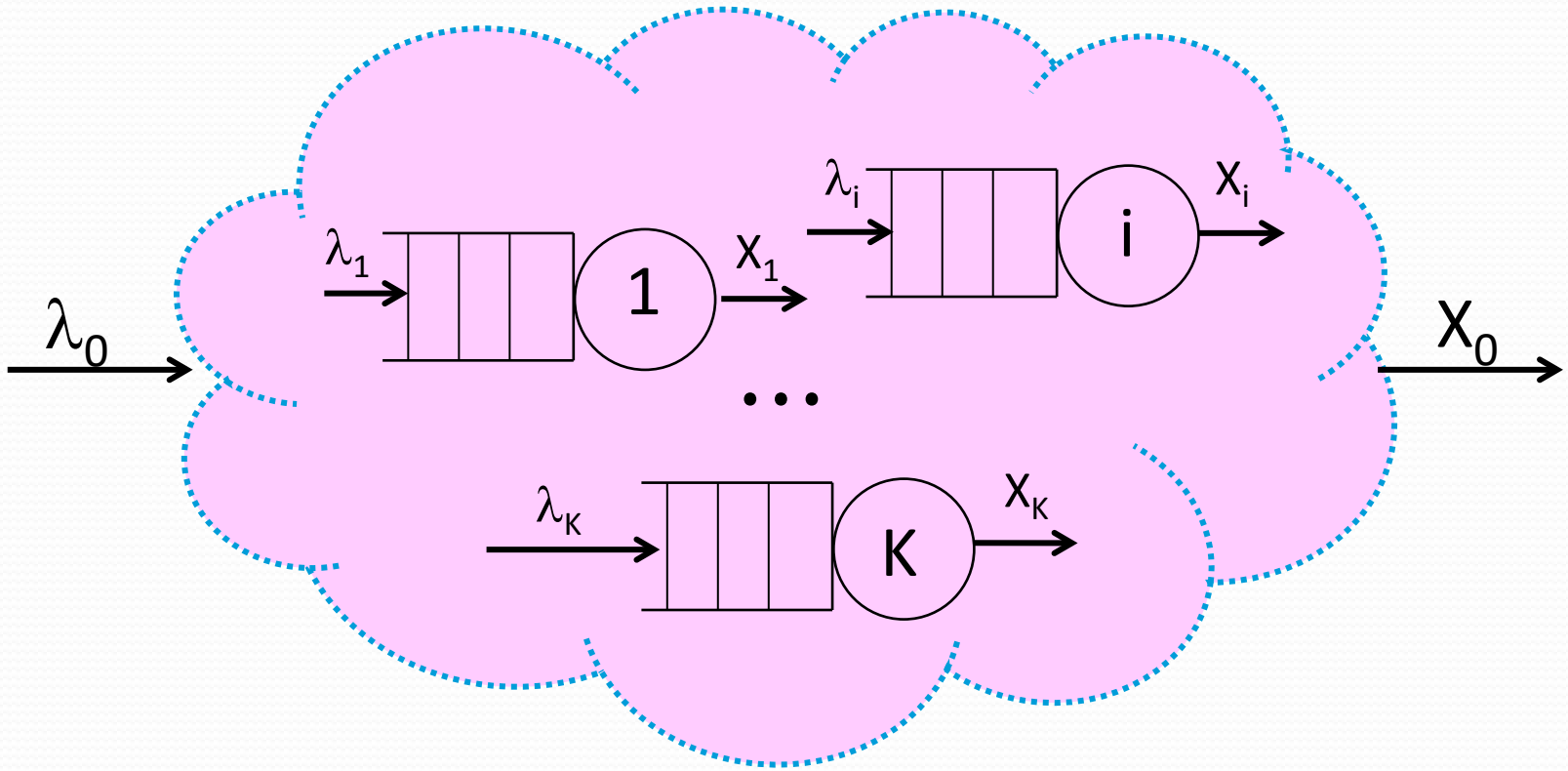
El análisis operacional

- Técnica de análisis de redes de colas basada en **valores medios** de diferentes variables medibles (*variables operacionales*) del servidor.

- Nos proporcionará relaciones generales entre las variables operacionales (*leyes operacionales*).
- Nos permitirá calcular las prestaciones del servidor para los casos de baja y alta carga por medio de cálculos muy sencillos.
- Nos permitirá evaluar los efectos en el rendimiento de diferentes modificaciones en los recursos del servidor.

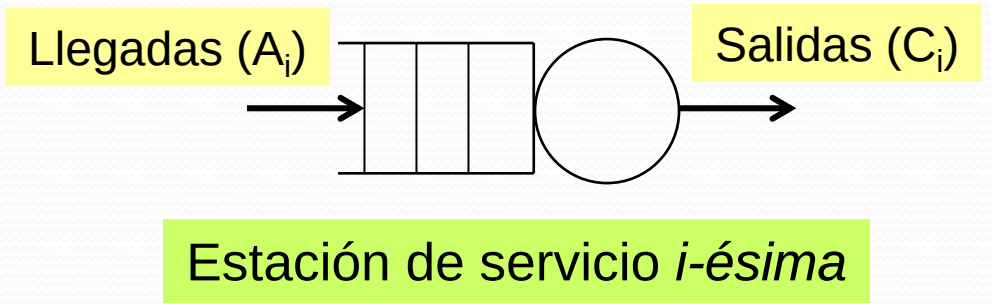
Variables del servidor y de cada estación de servicio

- El servidor contiene ***K*** estaciones de servicio (recursos o dispositivos).
- A todo el servidor en su globalidad lo denotamos como dispositivo "*cero*".



Variables operacionales básicas de una estación de servicio

- Variable global temporal:
  - $\boldsymbol{T}$ Duración del periodo de medida para el que se extrae el modelo.
- Variables operacionales básicas de la estación de servicio *i-ésima* **medidas durante el periodo de medida**:
  - $\boldsymbol{A_i}$ Número de trabajos solicitados a la estación (llegadas, *arrivals*).
  - $\boldsymbol{B_i}$ Tiempo que el dispositivo ha estado en uso (=ocupado) (*busy time*).
  - $\boldsymbol{C_i}$ Número de trabajos completados por la estación (salidas, *completions*).

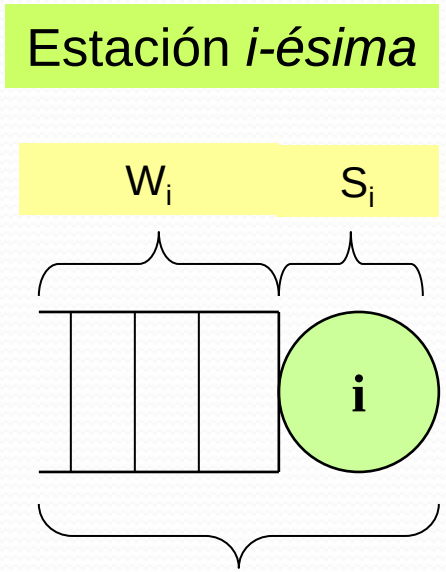


Variables operacionales deducidas

- **Todas ellas son valores medios y muchas se obtienen fácilmente a partir de las variables operacionales básicas:**
  - S_i Tiempo medio de servicio (*service time*): Cuánto tiempo necesita, de media, el recurso físico de la estación de servicio i-esima para atender cada petición que se le hace. Unidades: segundos/trabajo o simplemente segundos (s[/tr]).

$$S_i = \frac{B_i}{C_i}$$

  - W_i Tiempo medio de espera en cola (*waiting time*): Cuánto tiempo tiene que esperar, de media, un trabajo en la cola de la estación de servicio i-ésima para poder acceder al recurso físico. Unidades: segundos [/trabajo].
  - R_i Tiempo medio de respuesta *(response time*): Cuánto tiempo transcurre, de media, entre que un trabajo accede a la estación de servicio hasta que la abandona. Unidades: segundos [/trabajo].



$$R_i = W_i + S_i$$

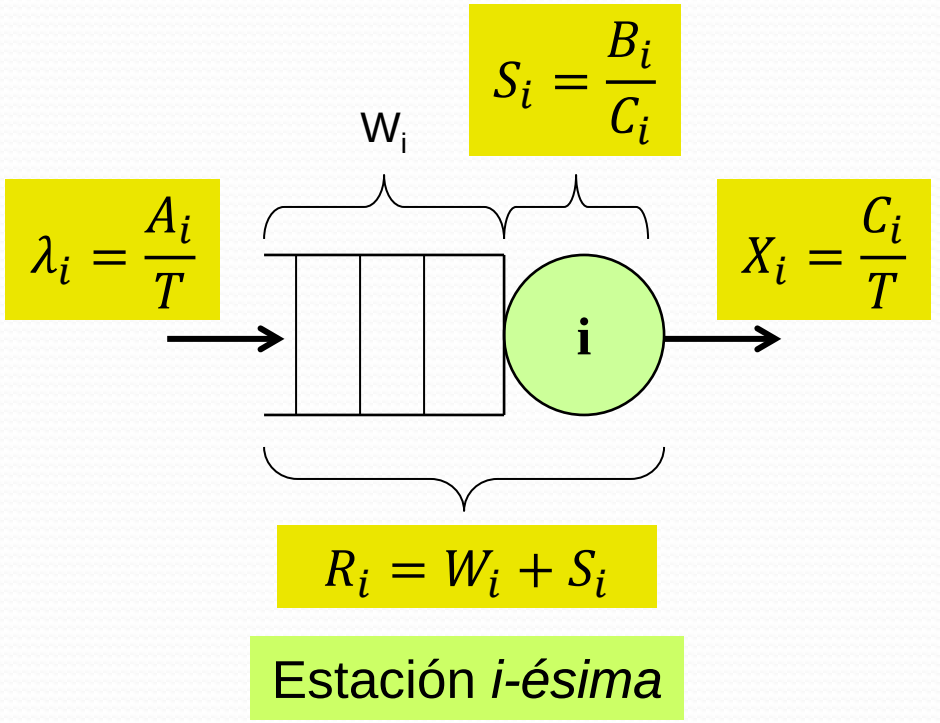
Variables operacionales deducidas (II)

- λ_i Tasa media de llegada (*arrival rate*): Cuántos trabajos por segundo llegan, de media, a la estación de servicio i-ésima. Unidades: tr/s.
- X_i Productividad media (*throughput*): Cuántos trabajos por segundo se completan, de media, por la estación de servicio i-ésima. Unidades: tr/s.
- U_i Utilización media *(utilization):* fracción del tiempo T que el dispositivo ha estado en uso (*busy*).

$$U_i = \frac{B_i}{T}$$

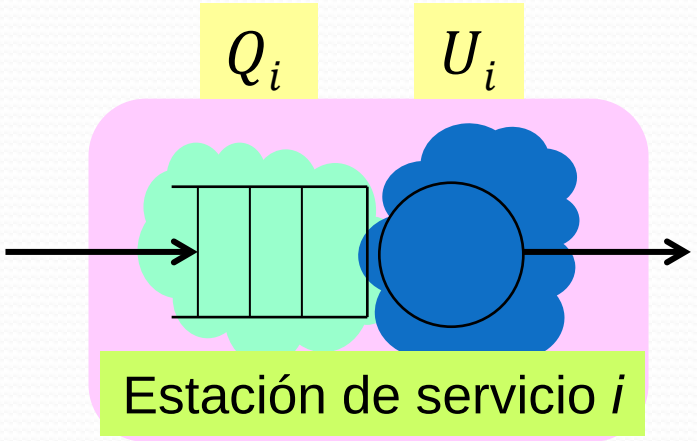
Unidades: No tiene unidades. A veces se expresa como tanto por ciento (%).

Valor máximo: $U_{i,max} = 1$ (=100%) cuando B_i=T.



Variables operacionales deducidas (III)

- **También hay variables operaciones deducidas que hacen referencia al número medio de trabajos, durante el periodo de medida T, en cada parte de una estación de servicio:**
  - $\mathbf{N_i}$: Número medio de trabajos en la estación de servicio (cola más recurso).
  - $\mathbf{Q_i}$: Número medio de trabajos en cola de espera (*jobs in queue*).
  - $\mathbf{U_i}$: Número medio de trabajos siendo servidos por el dispositivo, $U_i = N_i - Q_i$. Coincide numéricamente con la utilización media: $U_i = \frac{B_i}{T}$.



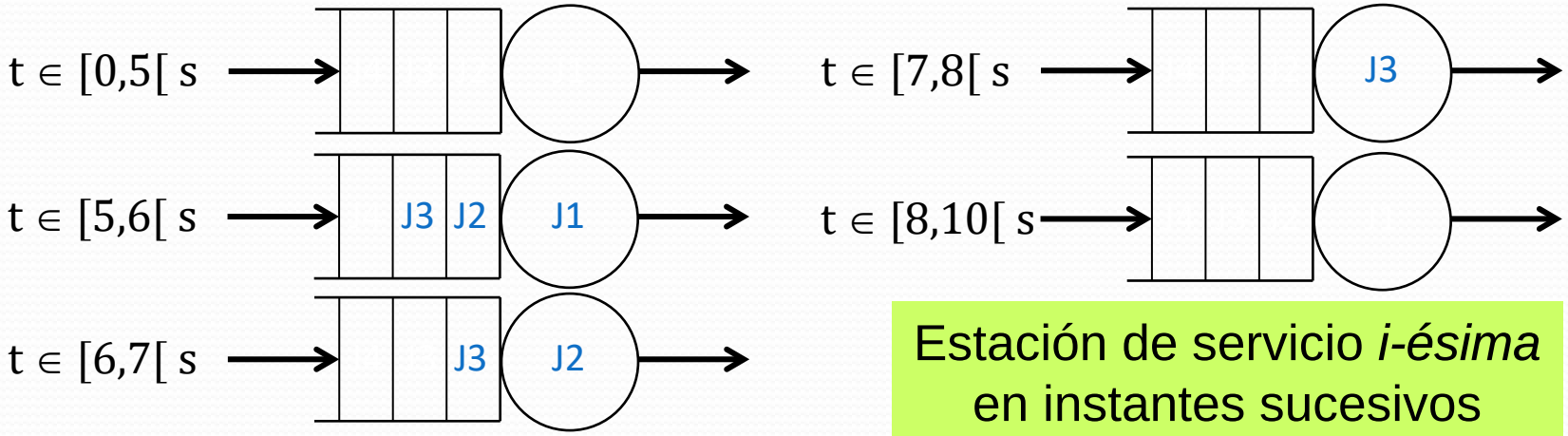
Ejercicio

Suponga que la estación de servicio i-ésima de una red de colas tiene un tiempo de servicio constante S_i=1s. Suponga que los trabajos llegan con la siguiente distribución temporal:

- Durante los primeros 5 segundos no llega ningún trabajo.
- En t=5s llegan 3 trabajos: J1, J2 y J3 (por ese orden).

Para el intervalo de medida [0, 10[s, calcule A_i, B_i, C_i, λ_i, X_i, U_i, Q_i, N_i.

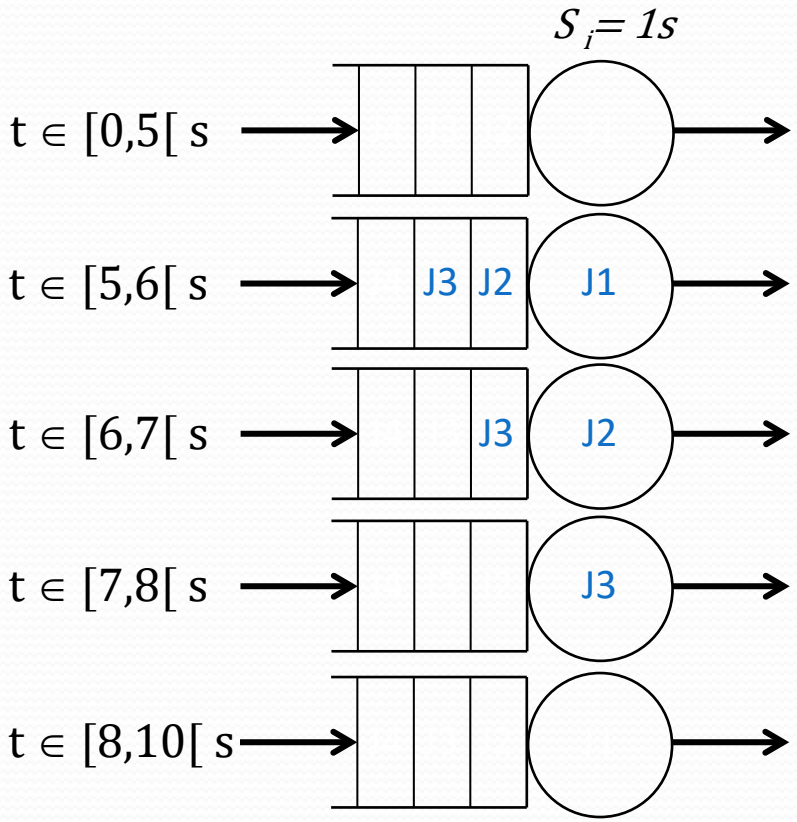
T = 10 s



A_i = 3 trabajos, B_i = 3s, C_i = 3 trabajos, $\lambda_i = A_i/T = 3/10 = 0{,}3$ trabajos/s = $Ci/T = X_i$

Ejercicio (cont.)

Cálculo de la utilización media (U_i) y del número medio de trabajos en la cola (Q_i) y en la estación (N_i).



Estación de servicio *i-ésima* en instantes sucesivos

$$U_i = \frac{B_i}{T} = \frac{3}{10} = 0{,}3$$

$$Q_i = \frac{0 \times 5s + 2 \times 1s + 1 \times 1s + 0 \times 3s}{10s} = 0{,}3\ tr.$$

$$N_i = Q_i + U_i = 0{,}6\ tr.$$

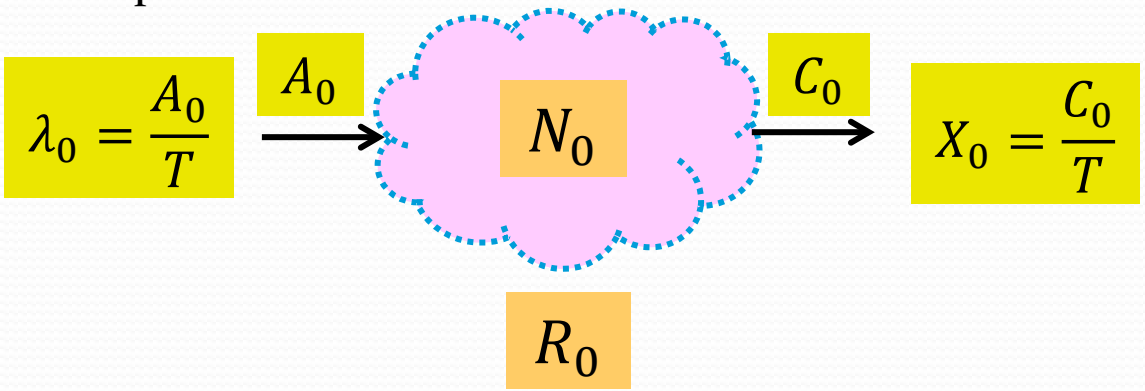
Otras alternativas para calcular N_i y U_i:

$$N_i = \frac{0 \times 5s + 3 \times 1s + 2 \times 1s + 1 \times 1s + 0 \times 2s}{10s} = 0{,}6\ tr.$$

$$U_i = \frac{0 \times 5s + 1 \times 3s + 0 \times 2s}{10s} = 0{,}3\ tr.$$

Variables operacionales de un servidor

- Variables operacionales básicas de un servidor:
  - $\boldsymbol{A_o}$ Número de trabajos solicitados al servidor (*arrivals*).
  - $\boldsymbol{C_o}$ Número de trabajos completados por el servidor (*completions*).
- Variables operacionales deducidas de un servidor:
  - $\boldsymbol{\lambda_o}$ Tasa media de llegada al servidor (*arrival rate*).
  - $\boldsymbol{X_o}$ Productividad media del servidor (*throughput*).
  - $\mathbf{N_o}$ Número medio de trabajos en el servidor (= número medio de clientes conectados al servidor). $\mathbf{N_o} = N_1+N_2+\ldots+N_K$.
  - $\mathbf{R_o}$ Tiempo medio de respuesta del servidor (*response time*) ≡ tiempo que tarda, de media, el servidor en procesar una petición.



Razón de visita y demanda de servicio

- Razón media de visita $\boldsymbol{V_i}$ (*visit ratio*). Representa la proporción entre el número de trabajos completados por el servidor y el número de trabajos completados por la estación de servicio i-ésima. Nos indica el número de veces que, de media, un trabajo "visita" la estación de servicio i-ésima antes de abandonar el servidor.

$$V_i = \frac{C_i}{C_0}$$

- Demanda media de servicio $\boldsymbol{D_i}$ (*service demand*). Cantidad de tiempo que, por término medio, el dispositivo de la estación de servicio i-ésima le ha dedicado a cada trabajo que abandona el servidor (= que ha sido procesado por completo por el servidor).

$$D_i = \frac{B_i}{C_0} = V_i \times S_i$$

Ejercicio

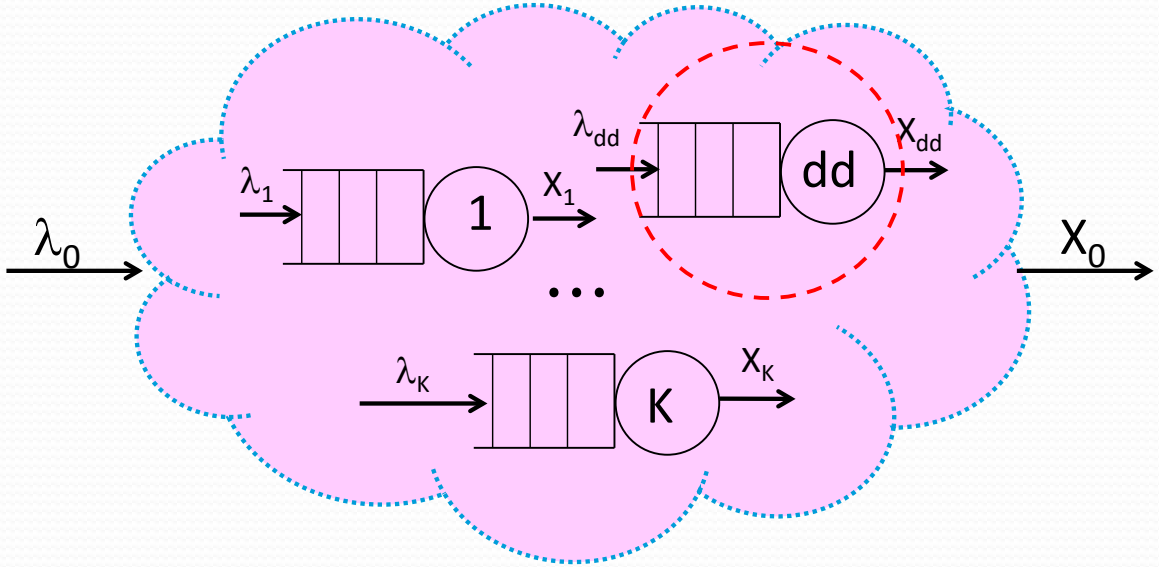
Después de monitorizar el **disco duro** de un servidor web durante un periodo de **24horas**, se sabe que ha estado en uso (=ocupado) un total de **6 horas**. Asimismo, se han contabilizado durante ese periodo un total de **98590** peticiones de lectura/escritura al disco duro y un total de **98591** peticiones completadas. Se ha estimado que cada petición atendida por el servidor web ha requerido una media de **9,5** peticiones de lectura/escritura al disco duro. Calcule, para ese periodo de monitorización:

a) La tasa media de llegada y la productividad media del disco duro.
b) La utilización media del disco duro.
c) El tiempo medio de servicio y la demanda media de servicio del disco duro.
d) ¿Cuál es la productividad media del servidor web?

Nota: Todas las variables operacionales deducidas que se usan en este tema son valores medios por lo que normalmente no se indicará de forma explícita la palabra "medio" al referirnos a ellas.

Solución

- Datos de partida:
  - T = 24h
  - B_{dd} = 6h
  - A_{dd} = 98590 pet. de lectura-escritura = 98590 tr
  - C_{dd} = 98591 pet. de lectura-escritura = 98591 tr
  - $V_{dd} = \frac{C_{dd}}{C_0} = 9{,}5 \frac{\text{pet. de lectura−escritura}}{\text{página web servida}} = 9{,}5$



a) $\lambda_{dd} = \frac{A_{dd}}{T} = \frac{98590\ tr}{24h} = 4108 \frac{tr}{h} = 1{,}14 \frac{tr}{s} = 1{,}14$ pet. de lectura − escritura/s

$X_{dd} = \frac{C_{dd}}{T} = \frac{98591\ tr}{24h} = 4108 \frac{tr}{h} = 1{,}14 \frac{tr}{s} = 1{,}14$ pet. de lectura − escritura/s

b) $U_{dd} = \frac{B_{dd}}{T} = \frac{6h}{24h} = 0{,}25\ (25\%)$.

c) $S_{dd} = \frac{B_{dd}}{C_{dd}} = \frac{6h}{98591\ tr} = \frac{6h \times 3600s/h}{98591\ tr} = 0{,}22s[/tr]$

$D_{dd} = V_{dd} \times S_{dd} = 9{,}5 \times 0{,}22s = 2{,}1s[/tr]$

d) Como: $C_0 = \frac{C_{dd}}{V_{dd}} = \frac{98591\ tr}{9{,}5} = 10378$ tr, tenemos que:

$X_0 = \frac{C_0}{T} = \frac{10378\ tr}{24h} = 432 \frac{tr}{h} = 0{,}12 \frac{tr}{s} =$ 0,12 pág. web/s

Leyes operacionales y equilibrio de flujo

- El valor de todas las variables utilizadas en el análisis operacional dependerá del intervalo de observación T.
- Existen, sin embargo, una serie de relaciones entre algunas variables operacionales que se mantienen válidas para cualquier intervalo de observación y que no dependen de suposiciones sobre la distribución de los tiempos de servicio o de la forma en la que llegan los trabajos. Estas relaciones se denominan **leyes operacionales**.
- Estas leyes se pueden expresar de una forma alternativa cuando el servidor se encuentra en **equilibrio de flujo.** Si se escoge un intervalo de observación T suficientemente largo (con respecto a lo que suele tardar el servidor en servir una petición, R_o), diremos que un servidor está en **equilibrio de flujo** si se cumple que:
  - El número de trabajos que completa el servidor coincide aproximadamente con los solicitados ($C_o \approx A_o$). Dicho de otra manera, la productividad media coincide aproximadamente con la tasa media de llegada ($X_o \approx \lambda_o$).
  - El número de trabajos que completa cada estación de servicio coincide aproximadamente con los que se solicitan: ($C_i \approx A_i \Rightarrow X_i \approx \lambda_i$, $\forall i=1...K$).

Ley de la Utilización

- Relaciona la utilización media de un dispositivo con el número de trabajos que completa por unidad de tiempo (=su productividad media) y el tiempo que le dedica, de media, a cada uno de ellos (=su tiempo de servicio medio).

Si equilibrio de flujo

$$\forall\, i = 1,..,K \qquad U_i = X_i \times S_i = \lambda_i \times S_i$$

- Demostración:

$$\forall\, i = 1,..,K \qquad S_i = \frac{B_i}{C_i} = \frac{B_i/T}{C_i/T} = \frac{U_i}{X_i}$$

- Una consecuencia inmediata de esta ley es que la productividad media de un dispositivo viene limitada por la inversa de su tiempo de servicio:

$$U_i \leq 1 \quad \Rightarrow \quad X_i \leq \frac{1}{S_i} \quad \forall i = 1, \ldots, K$$

Ley del flujo forzado y relación utilización-demanda

- Las productividades (=flujos de salida) de cada estación de servicio tienen que ser proporcionales a la productividad global del servidor. La **ley del flujo forzado** relaciona la productividad del servidor con la de cada uno de los dispositivos que integran el mismo:

Si equilibrio de flujo

$$\forall\, i = 1,..,K \qquad X_i = X_0 \times V_i = \lambda_0 \times V_i = \lambda_i$$

Demostración: $V_i = \frac{C_i}{C_0} = \frac{C_i/T}{C_0/T} = \frac{X_i}{X_0}$

- Las utilizaciones de cada dispositivo son proporcionales a las demandas de servicio del mismo, siendo la constante de proporcionalidad precisamente la productividad global del servidor (**relación utilización-demanda de servicio**):

Si equilibrio de flujo

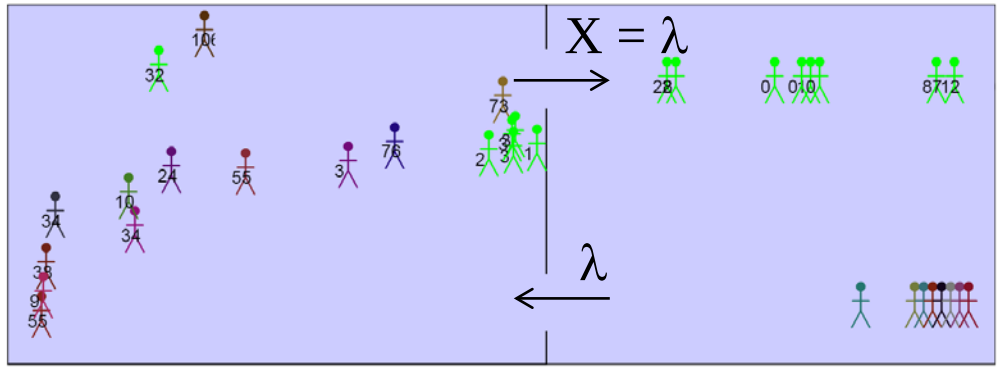
$$\forall\, i = 1,..,K \qquad U_i = X_0 \times D_i = \lambda_0 \times D_i$$

Demostración: $D_i = \frac{B_i}{C_0} = \frac{B_i/T}{C_0/T} = \frac{U_i}{X_0}$

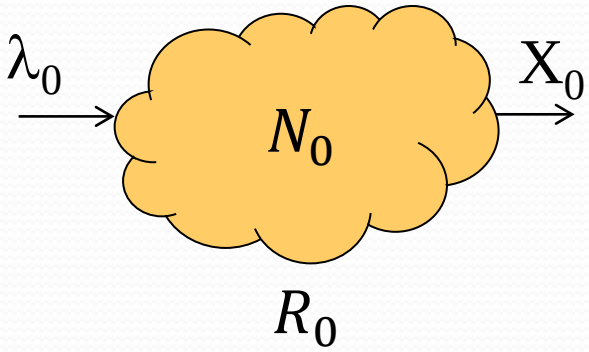
$$U_i \leq 1 \quad \Rightarrow \quad X_0 \leq \frac{1}{D_i} \quad \forall i = 1, \dots, K$$

Ley de Little

"The long-term average number of customers in a stationary system is equal to the long-term average arrival rate multiplied by the average time a customer spends in the system"



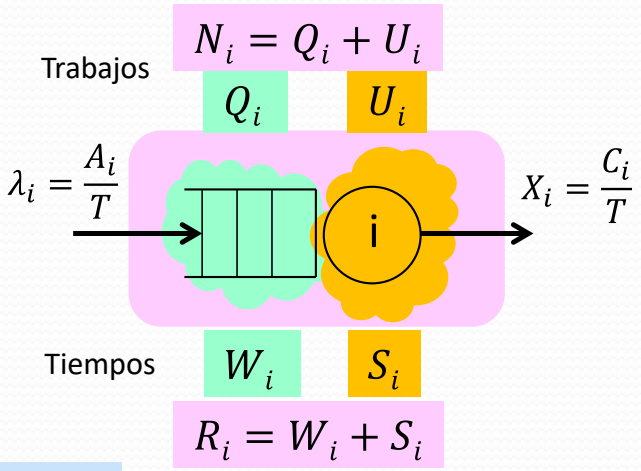
- Aplicada a un servidor, esta ley relaciona las dos variables más importantes que reflejan el rendimiento de un servidor: su productividad media (X_o) y su tiempo medio de respuesta (R_o).



$$N_0 = \lambda_0 \times R_0 = X_0 \times R_0$$

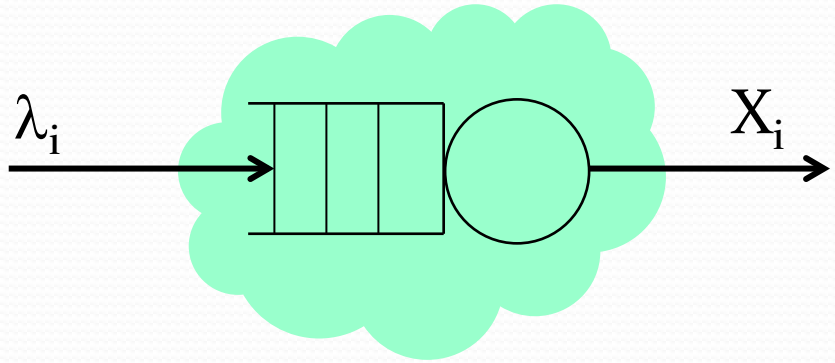
- Esta ley solo es válida cuando el servidor está en equilibrio de flujo.

Ley de Little (cont.)



La ley de Little puede ser aplicada no solo al servidor en su totalidad, sino a cada estación de servicio y a cada uno de los diferentes sub-niveles de una estación de servicio.

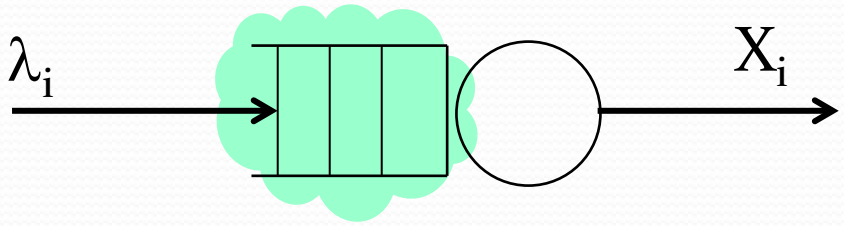
- Aplicación a toda una estación de servicio:



$$N_i = \lambda_i \times R_i = X_i \times R_i$$

Tiempo medio de respuesta: R_i
Nº medio de trabajos en la estación: N_i

- Aplicación a la cola de una estación de servicio:



$$Q_i = \lambda_i \times W_i = X_i \times W_i$$

Tiempo medio de espera en cola: W_i
Nº medio de trabajos en la cola: Q_i

Ejemplo de aplicación

- Un servidor de base de datos **en equilibrio de flujo** recibe una media de 120 consultas por minuto. Sabemos que su disco duro tarda, de media, 30ms en atender cada petición de lectura/escritura (abreviado *R/W*) que le llega (48ms si incluimos la espera en la cola) y que su productividad es 25 peticiones de R/W completadas por segundo. Calcule:

  a) El número medio de peticiones de R/W en la cola de espera del disco duro.

  - Solución (podemos usar la Ley de Little ya que el servidor está en equilibrio de flujo):

$$\boldsymbol{Q_{dd} = \lambda_{dd} \times W_{dd} = X_{dd} \times (R_{dd} - S_{dd}) = 25 \frac{tr}{s} \times 0,018s = 0,45\ tr = 0,45\ pet.de\ R/W}$$

  - Solución alternativa:

$$N_{dd} = \lambda_{dd} \times R_{dd} = X_{dd} \times R_{dd} = 25 \text{ tr/s} \times 0{,}048 \text{ s} = 1{,}2 \text{ tr}$$

$$U_{dd} = X_{dd} \times S_{dd} = 25 \text{ tr/s} \times 0{,}03 \text{ s/tr} = 0{,}75 \ (75\%)$$

$$\boldsymbol{Q_{dd} = N_{dd} - U_{dd} = 1,2 - 0,75 = 0,45\ tr = 0,45\ pet.de\ R/W}$$

  b) ¿Cuánto tiempo de disco duro requiere, de media, cada consulta que se realiza al servidor?

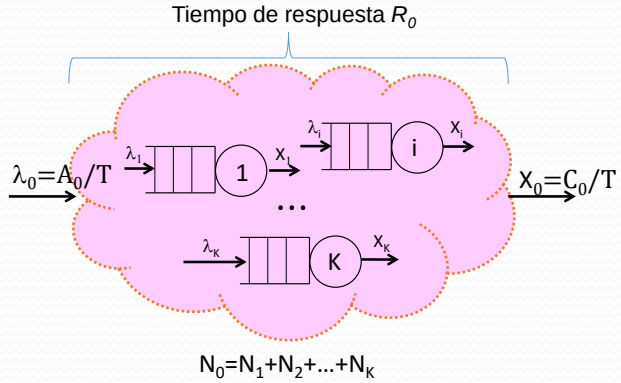
$$\boldsymbol{D_{dd} = \frac{B_{dd}}{C_0} = \frac{B_{dd}/_T}{C_0/_T} = \frac{U_{dd}}{X_0} = \frac{U_{dd}}{\lambda_0} = \frac{0,75}{120\ tr/min} = \frac{0,75}{2\ tr/s} = 0,375\ s[/tr] = 0,375\ s[/consulta]}$$

Ley general del tiempo de respuesta

- El tiempo medio de respuesta que experimenta, de media, una petición a un servidor en equilibrio de flujo se puede calcular teniendo en cuenta que cada una de ellas ha tenido que "visitar" V_i veces al dispositivo i-ésimo, requiriendo cada visita una media de R_i segundos:

$$R_0 = V_1 \times R_1 + V_2 \times R_2 + \cdots + V_K \times R_K = \sum_{i=1}^{K} V_i \times R_i$$

Ley general del tiempo de respuesta



- Demostración:

Ley de Little ↓

$$N_0 = N_1 + N_2 + \cdots + N_K \Rightarrow X_0 \times R_0 = \boldsymbol{X_1} \times R_1 + \boldsymbol{X_2} \times R_2 + \cdots + \boldsymbol{X_K} \times R_K$$

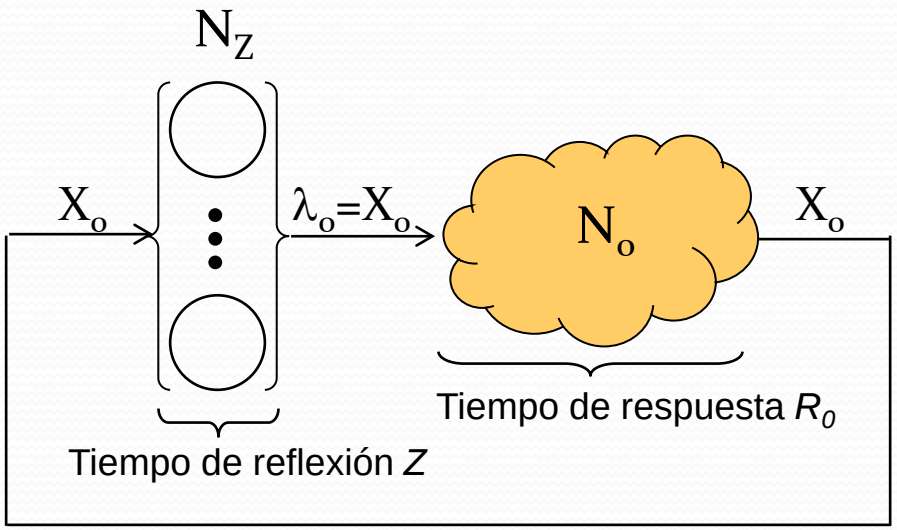
Ley del Flujo Forzado →

$$\Rightarrow X_0 \times R_0 = \boldsymbol{X_0} \times \boldsymbol{V_1} \times R_1 + \boldsymbol{X_0} \times \boldsymbol{V_2} \times R_2 + \cdots + \boldsymbol{X_0} \times \boldsymbol{V_K} \times R_K$$

Nótese que, en general: $R_0 \neq R_1 + R_2 + \cdots + R_K = \sum_{i=1}^{K} R_i$

Ley del tiempo de respuesta interactivo

- Un servidor en una red cerrada siempre está en equilibrio de flujo, es decir, siempre supondremos que el tamaño de las colas es suficientemente grande para asegurarlo ($\geq N_T$).
- Al ser una red cerrada, el número total de trabajos (=clientes) en la red ($N_T=N_Z+N_0$), es constante.
- Aplicamos la ley de Little a diversas partes de la red de colas:
  - Ley de Little aplicada a los clientes en reflexión: $N_Z = X_0 \times Z$, donde N_Z = Número **medio** de clientes (=trabajos) en reflexión.
  - Ley de Little aplicada al servidor: $N_0 = X_0 \times R_0$



$$N_T = N_Z + N_0 = X_0 \times Z + X_0 \times R_0 = \\ = X_0 \times (Z + R_0)$$

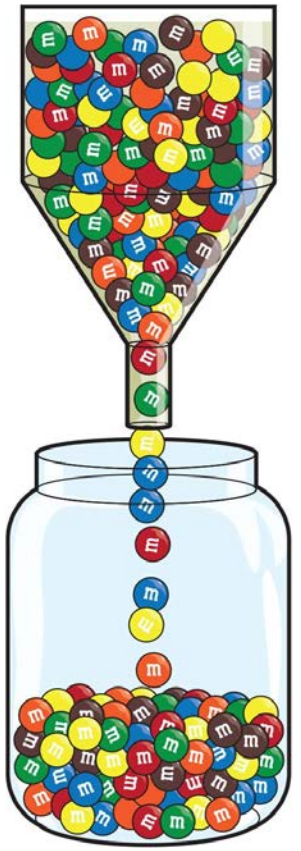
$$R_0 = \frac{N_T}{X_0} - Z$$

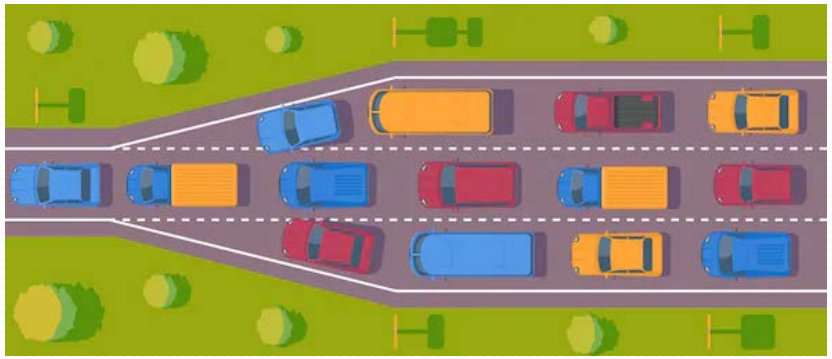
que se conoce como la *Ley del tiempo de respuesta interactivo*.

5.3. Límites optimistas del rendimiento

Limitaciones en el rendimiento: cuello de botella

- Todo servidor presenta alguna limitación en su rendimiento.
- La localización del elemento limitador no solo depende del servidor sino también del tipo carga.
  - Al elemento limitador del rendimiento del servidor se le denomina cuello de botella (*bottleneck*).
  - Además, puede haber más de uno de estos elementos limitadores.
- Veremos que la única manera de mejorar las prestaciones de un servidor de manera significativa es actuando sobre el cuello de botella.





Identificación del cuello de botella

- El cuello de botella es el dispositivo que primero llegará a **saturarse** (utilización media = 1) cuando aumente la carga (λ_0 mayor).

Si equilibrio de flujo

$$U_i = X_0 \times D_i = \lambda_0 \times D_i \qquad \forall\, i = 1,..,K$$

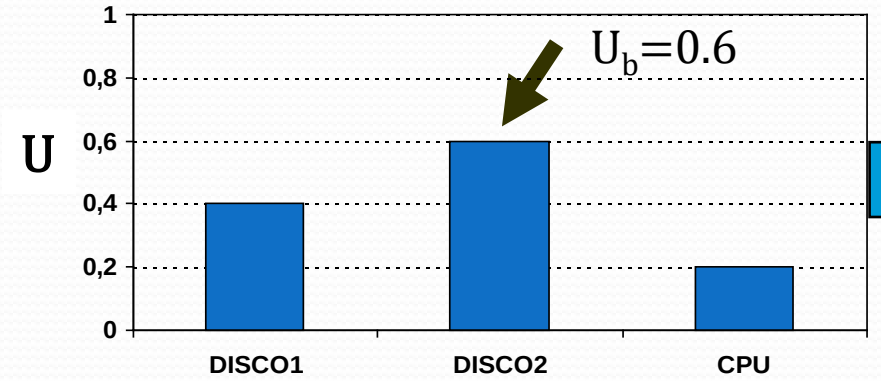
- Como U_i es proporcional a D_i, podemos identificar fácilmente el cuello de botella de un servidor simplemente identificando el dispositivo con **mayor demanda de servicio o con mayor utilización.**
- No hace falta llevar el servidor al límite para identificar el cuello de botella.
- Como $D_i = V_i \times S_i$, concluimos que la localización del cuello de botella no solo depende de lo rápido que sea el dispositivo (S_i) sino también del tipo de carga a la que está sometido (V_i).
- Denotaremos por "b" (*bottleneck*) al índice del dispositivo cuello de botella. Su demanda de servicio y su utilización vendrán dadas por.

$$D_b = \max_{i=1\cdots K}\{D_i\} = V_b \times S_b$$

$$U_b = \max_{i=1\cdots K}\{U_i\} = X_0 \times D_b$$

Saturación del servidor

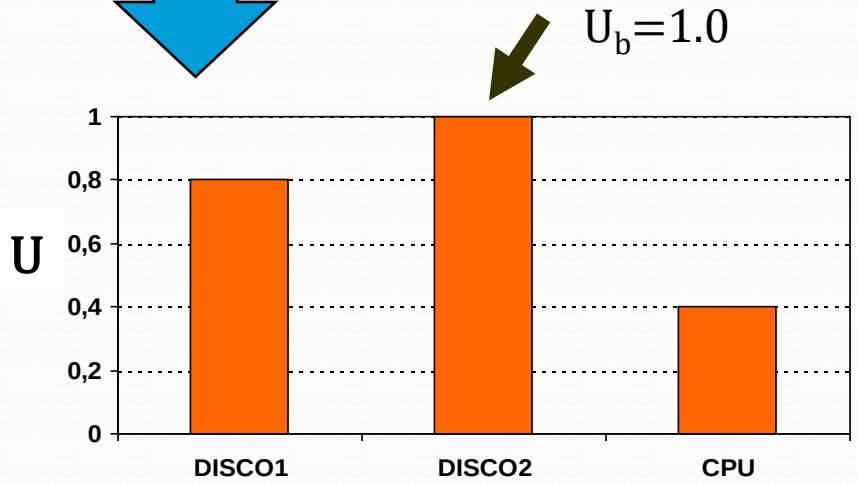
- El servidor se satura cuando lo hace el cuello de botella ya que éste será el primer dispositivo en alcanzar una utilización media = 1 cuando aumente la carga.



Operación “normal”

$$U_b = X_0 \times D_b \overset{\text{Si eq. flujo}}{=} \lambda_0 \times D_b$$

Aumento de la carga: $\lambda_0 \Uparrow\Uparrow$



Servidor saturado

- En saturación, el cuello de botella está al máximo de su productividad:

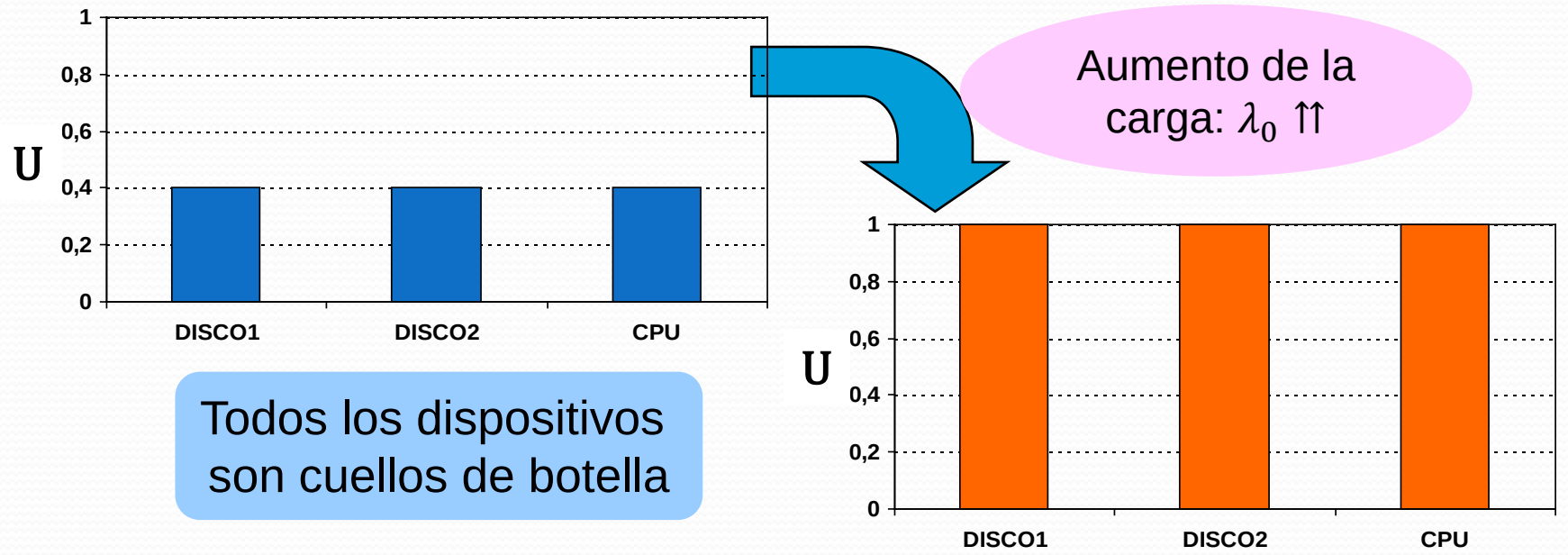
$$1 = U_b = X_b \times S_b \rightarrow X_b = {}^{1}/_{S_b}$$

- El equilibrio de flujo siempre se cumple mientras el servidor no esté saturado.

Servidor equilibrado (*balanced system*)

- Servidor en el que todos los dispositivos, de media, tienen la misma demanda de servicio y utilización (la carga se absorbe equitativamente):

$$U_i \approx U_j \ \ \forall i,j = 1 \cdots K \quad \Rightarrow \quad D_i \approx D_j \ \ \forall i,j = 1 \cdots K$$

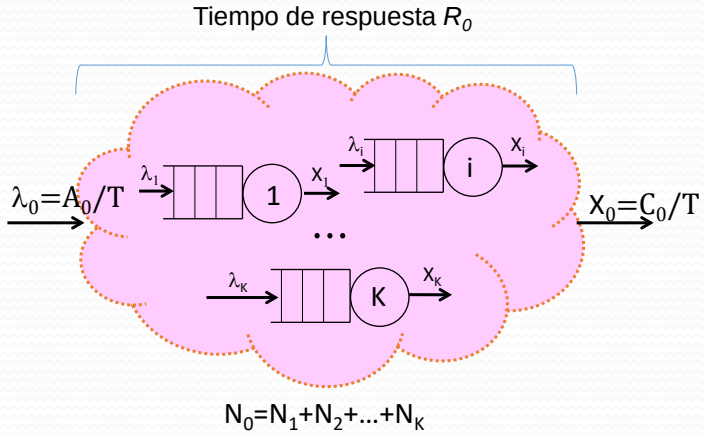


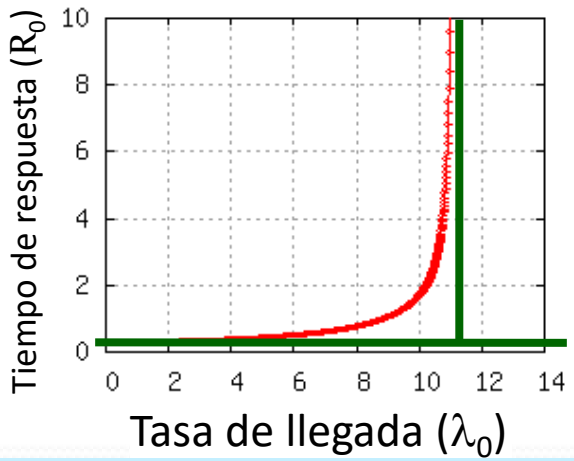
Límites del rendimiento de un servidor

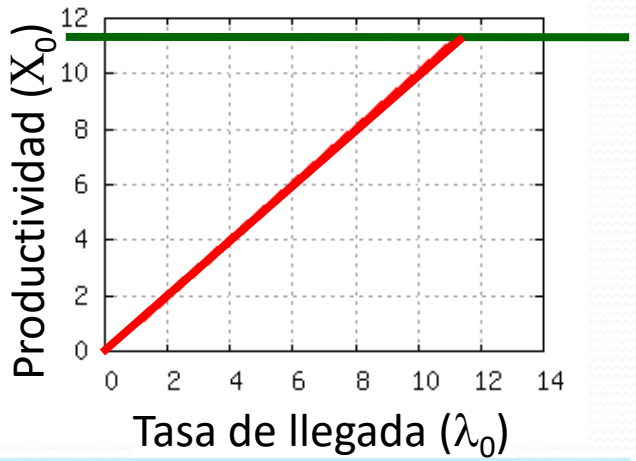
- Vamos a estimar los límites en las prestaciones de un servidor (R_o, X_o) en los casos extremos de alta y baja cargas.
- Esencialmente, se trata de estimar una cota superior de la productividad e inferior para el tiempo de respuesta del servidor por lo que a estos límites se les suele denominar ***límites optimistas*** del rendimiento. En particular, debemos preguntarnos:
  - ¿Cuál es la mejor "productividad media" que puede alcanzar del servidor? X_0^{max}.
  - ¿Cuál es el mejor "tiempo de respuesta medio" que puede proporcionar del servidor? R_0^{min}.
- ¿Para qué me sirve esto?:
  - Para poder estimar la *capacidad* del servidor.
  - Para poder estimar la mejora potencial de prestaciones que pueden reportar ciertas acciones sobre el servidor.

Estimación de los límites del rendimiento del servidor

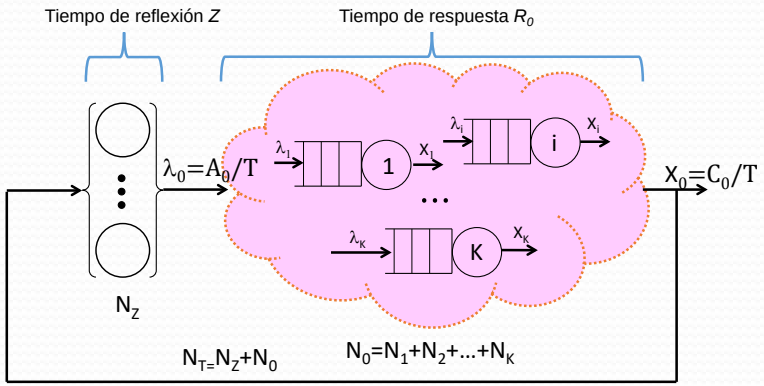
Redes abiertas

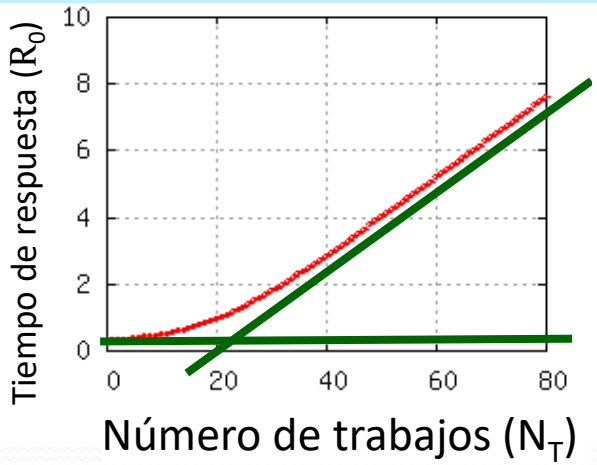


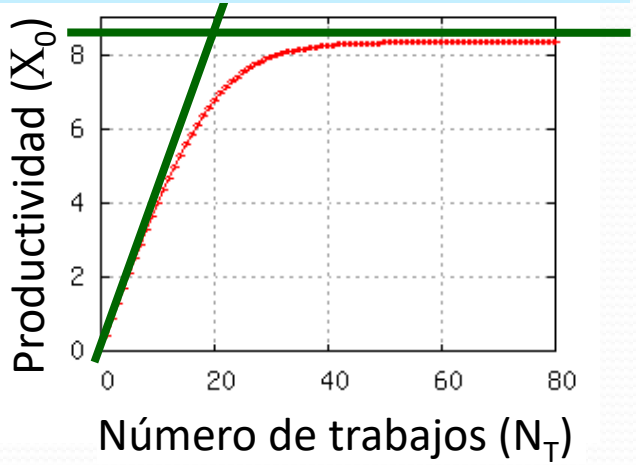




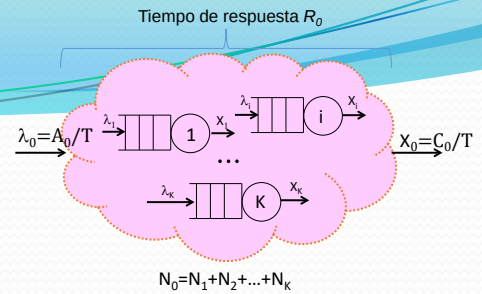
Redes cerradas







Límites optimistas: redes abiertas



- El valor máximo de la productividad media del servidor será aquél producido por una tasa de llegada que sature el dispositivo cuello de botella (U_b=1):

Si eq. flujo

$$U_b = X_0 \times D_b = \lambda_0 \times D_b \qquad Si\ U_b = 1 \Rightarrow X_o^{max} = \frac{1}{D_b}$$

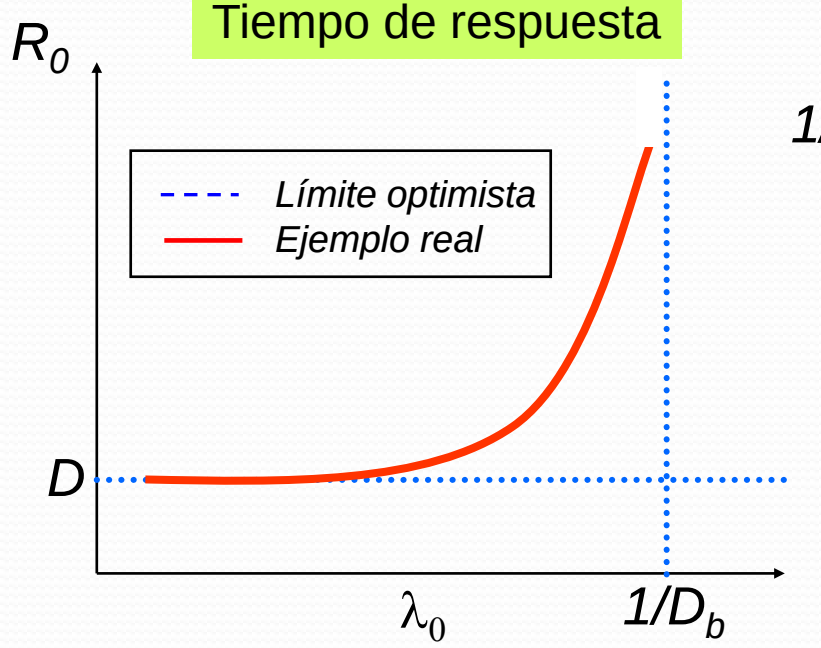
Una tasa de llegada mayor provocaría un aumento descontrolado de la cola del cuello de botella hasta su desbordamiento final y, por tanto, dejaría de cumplirse la hipótesis del equilibro de flujo (X_o ya no podría seguir a λ_o). R_o crecería igualmente sin control.

- El valor más optimista (= el valor mínimo) del tiempo medio de respuesta del servidor (R_o^{min}) será el que experimenta un trabajo cuando llega al servidor sin que haya otros trabajos previamente ($\mathbf{W_i = 0\ \forall i=1,..,K}$):

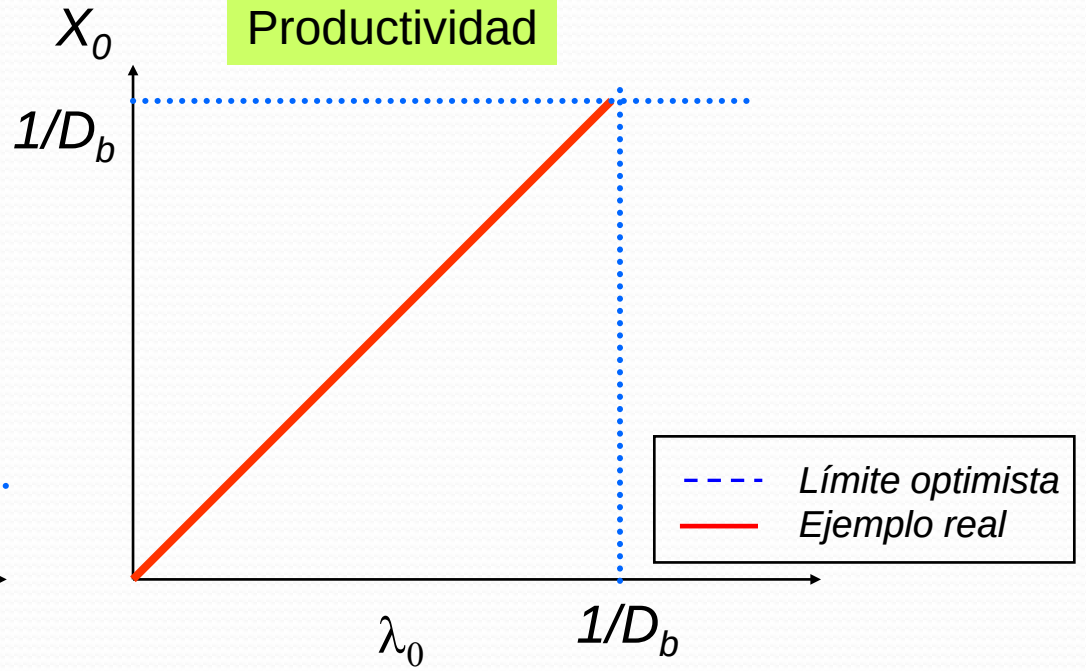
$$R_0 = \sum_{i=1}^{K} V_i \times R_i = \sum_{i=1}^{K} V_i \times (W_i + S_i) \quad \Rightarrow \quad R_0 \rightarrow R_o^{min} = \sum_{i=1}^{K} V_i \times S_i = \sum_{i=1}^{K} D_i \equiv D$$

Resumen: Límites optimistas en redes abiertas

Tiempo de respuesta



Productividad



$$R_o^{min} = D \equiv \sum_{i=1}^{K} D_i$$

$$X_o^{max} = \frac{1}{D_b}$$

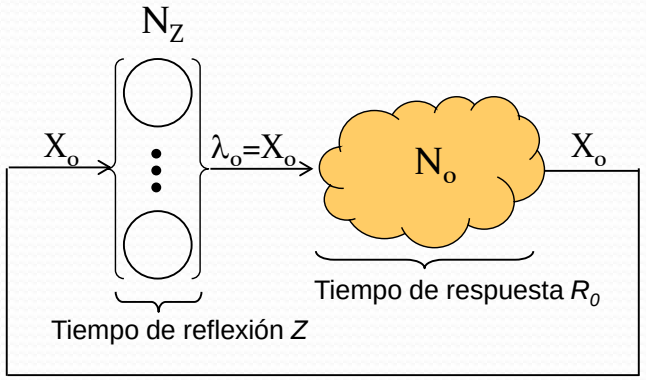
Ejemplo de red abierta

Dispositivo	V_i	S_i (ms)	D_i (s)
CPU	16	10	0,16
DISCO A	7	20	0,14
DISCO B	8	30	**0,24**

- Tiempo de respuesta medio mínimo:
  - $R_0^{min} = 0{,}16 + 0{,}14 + 0{,}24 = 0{,}54s.$
- Productividad media máxima:
  - $X_0^{max} = \frac{1}{D_b} = \frac{1}{0{,}24} = 4{,}2\ tr/s.$
- Utilización media máxima de la CPU:
  - $U_{CPU}^{max} = X_0^{max} \times D_{CPU} = 0{,}67.$
- Productividad media máxima de la CPU:
  - $X_{CPU}^{max} = X_0^{max} \times V_{CPU} = 67\ tr/s.$
- U_i con λ_o = 2 trabajos/s :
  - Como $\lambda_o < X_0^{max}$ estamos en equilibrio de flujo: $X_o = \lambda_o$
  - $U_{CPU} = X_o \times D_{CPU} = 0{,}32.$
  - $U_A = X_o \times D_A = 0{,}28.$
  - $U_B = X_o \times D_B = 0{,}48.$

Nota: En análisis operacional siempre hablamos de **valores medios**. En este caso, nos referimos a valores máximos y mínimos de esos valores medios que se podrían alcanzar variando la tasa de llegada de peticiones al servidor.

Límites optimistas: redes cerradas



Ley de Little a la red completa ($N_T=N_o+N_Z$):

$$N_T = X_0 \times (R_0 + Z)$$

$$R_0 = \frac{N_T}{X_0} - Z$$

- a) Para valores de carga altos (N_T grande):
  - Valor optimista de la productividad media: Cuando el dispositivo cuello de botella esté cerca de la saturación:

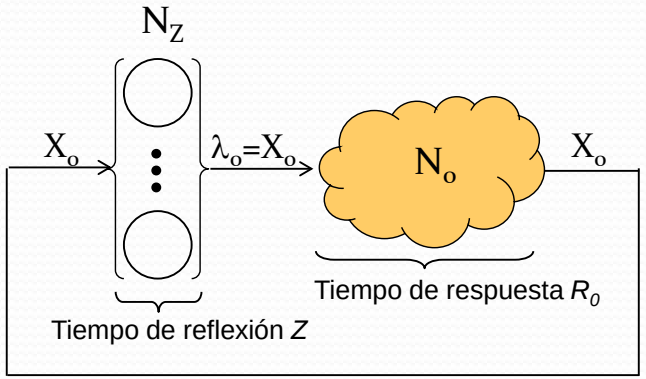
$$U_b = X_0 \times D_b$$

$$Si\ U_b \longrightarrow 1 \Rightarrow X_0 \longrightarrow X_o^{max} = \frac{1}{D_b}$$

  - Valor optimista del tiempo de respuesta medio: A partir del valor optimista de la productividad media (sin más que reemplazar ese valor de X_o en la ley de Little a la red completa):

$$R_0 \longrightarrow \left(\frac{N_T}{X_o^{max}}\right) - Z = D_b \times N_T - Z$$

Límites optimistas: redes cerradas (II)



Ley de Little a la red completa ($N_T=N_o+N_Z$):

$$N_T = X_0 \times (R_0 + Z)$$

$$X_0 = \frac{N_T}{R_0 + Z}$$

- b) Para valores de carga bajos (N_T pequeño):
  - Valor optimista del tiempo de respuesta medio: cuando los trabajos siempre encuentran los dispositivos sin ocupar ($\mathbf{W_i = 0\ \forall i=1..K}$):

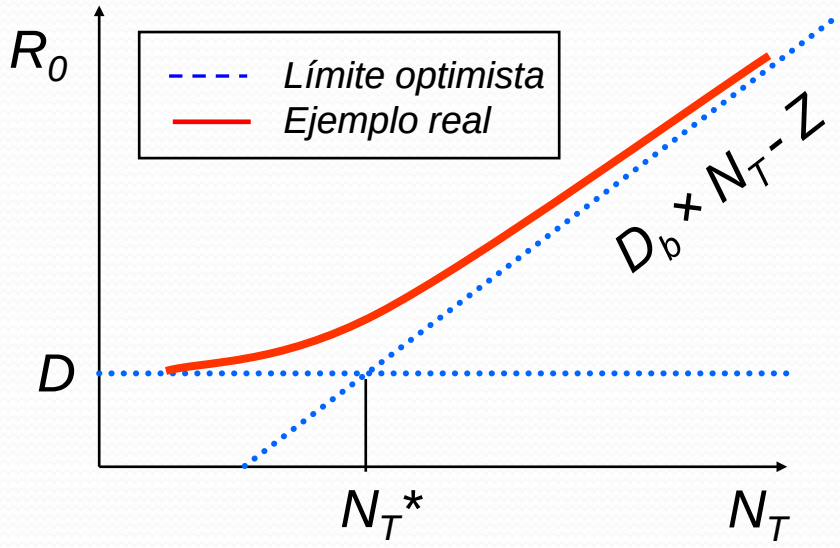
$$R_0 = \sum_{i=1}^{K} V_i \times R_i = \sum_{i=1}^{K} V_i \times (W_i + S_i) \quad \rightarrow \quad R_0 \rightarrow R_o^{min} = \sum_{i=1}^{K} V_i \times S_i = \sum_{i=1}^{K} D_i \equiv D$$

  - Valor optimista de la productividad media: A partir del valor optimista del tiempo de respuesta medio (sin más que reemplazar ese valor de R_o en la ley de Little a la red completa):

$$X_0 \rightarrow \frac{N_T}{R_o^{min} + Z} = \frac{N_T}{D + Z}$$

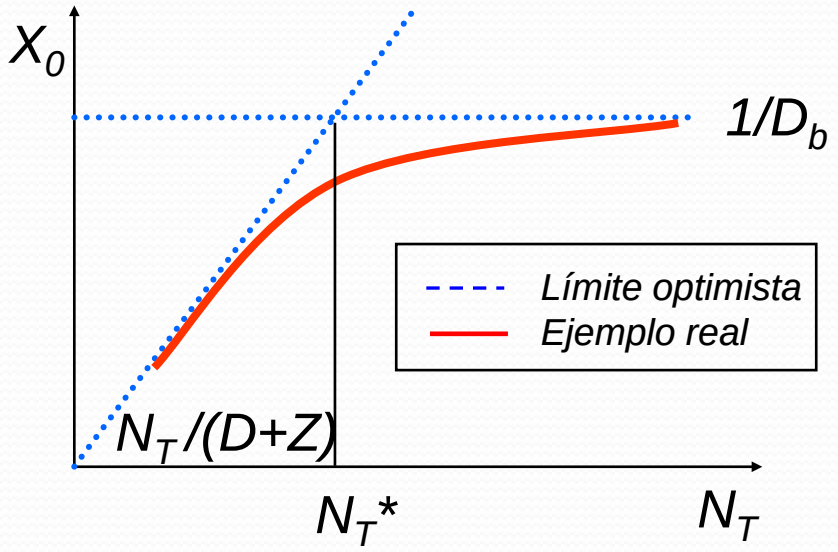
Resumen: límites optimistas en redes cerradas

Tiempo de respuesta



$$R_0 \geq max\{D, D_b \times N_T - Z\}$$

Productividad



$$X_0 \leq min\left\{\frac{N_T}{D+Z}, \frac{1}{D_b}\right\}$$

Punto teórico de “saturación” (knee point)

- Es el valor de N_T en donde las asíntotas coinciden:

$$D = D_b \times N_T^* - Z \Rightarrow N_T^* = \frac{D + Z}{D_b}$$

- Propiedades del punto teórico de “saturación” N_T^*:
  - Para un número total de trabajos $N_T > N_T^*$, los límites en las prestaciones vienen impuestos únicamente por el cuello de botella del servidor.
  - A partir de N_T^* trabajos ya no se puede conseguir el tiempo de respuesta mínimo ya que se empiezan a formar colas de espera en, al menos, el dispositivo cuello de botella (en la práctica, esto sucede bastante antes).
  - En principio, podría parecer el número ideal de trabajos en la red ya que, al menos teóricamente, para $N_T = N_T^*$ se podría conseguir la productividad máxima y el tiempo de respuesta mínimo absolutos del servidor (en la práctica esto nunca se puede conseguir de forma simultánea): $N_T^* = X_o^{max} \times \left(R_o^{min} + Z\right) = \frac{D+Z}{D_b}$

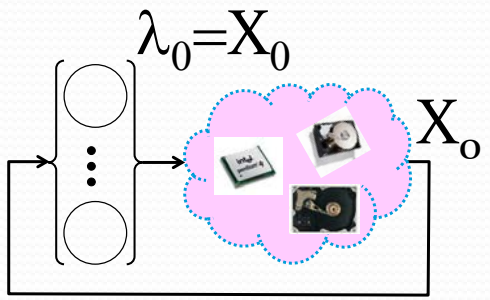
Ejercicio: cálculo de límites optimistas de una red cerrada

Tiempo de reflexión (Z)			18 s
Dispositivo	V_i	S_i (s)	D_i (s)
CPU	5	1	5
DISCO A	2	2	4
DISCO B	2	1,5	3

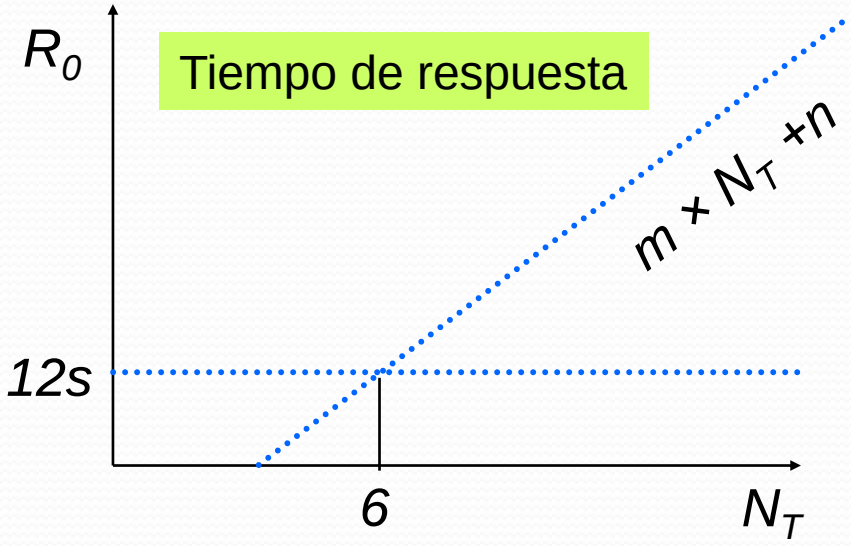
Cuello de botella: CPU. $D_b = D_{CPU} = 5s$

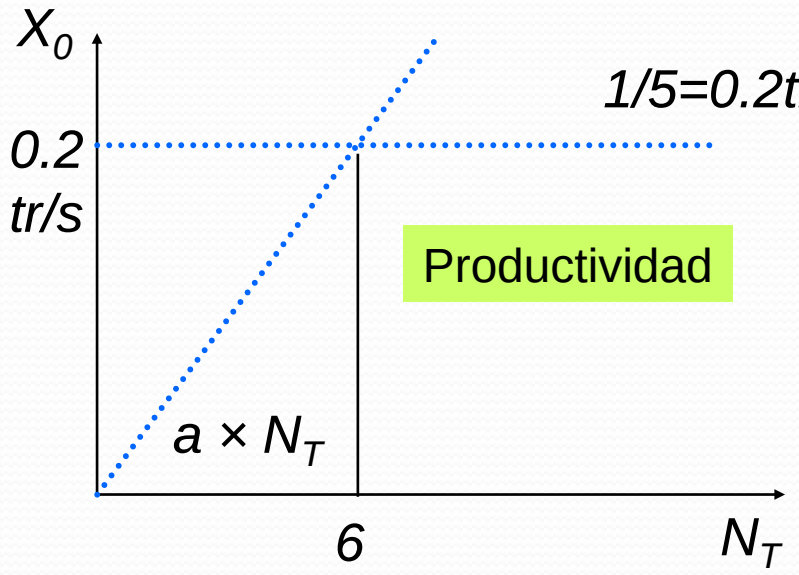
$$X_o^{max} = \frac{1}{D_b} = 1/5 = 0.2\ tr/s$$

$$R_o^{min} = D \equiv D_{CPU}+D_{DISCOA}+D_{DISCOB} = 12s$$



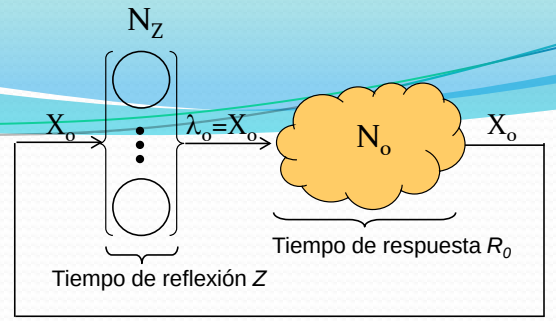
Punto teórico de saturación: $N_T^* = \frac{D+Z}{D_b} = \frac{12+18}{5} = 6\ tr$





Solo nos queda calcular los parámetros que definen a las dos asíntotas que nos quedan: **m**, **n** y **a**

Ejercicio (continuación)

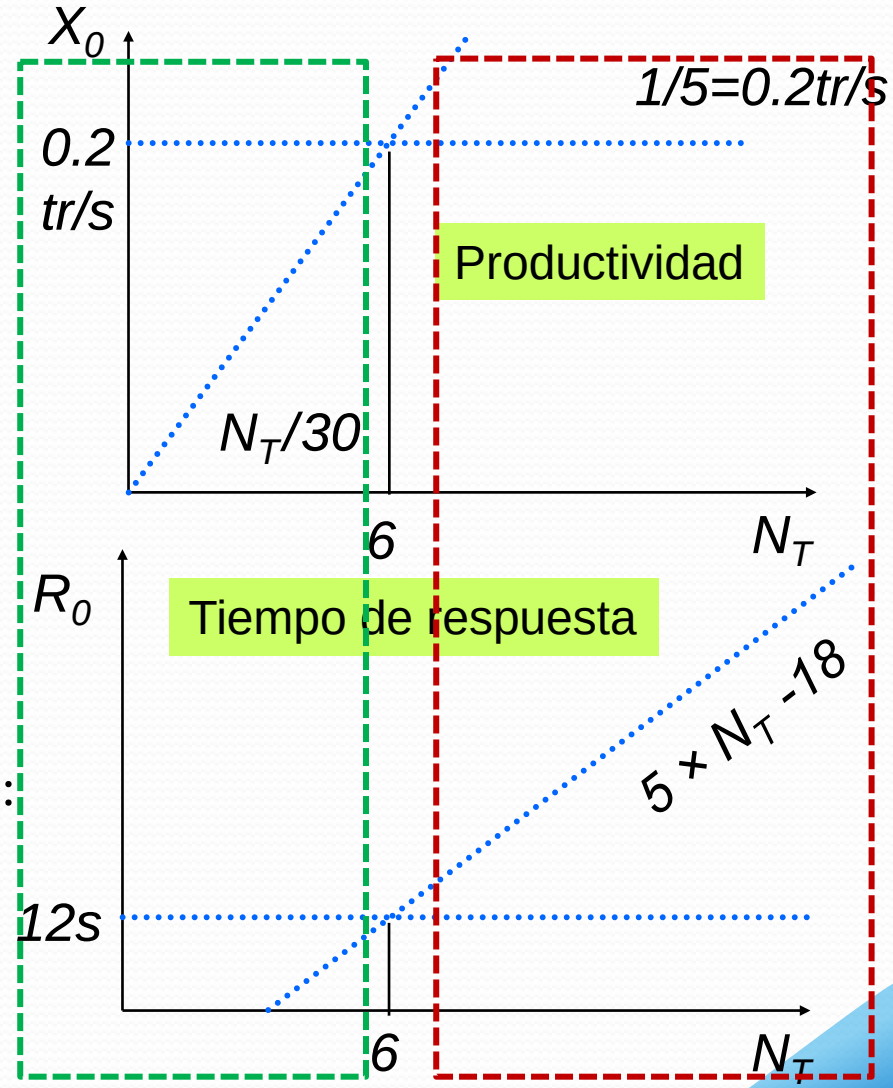


Ley de Little a la red completa:

$$N_T = X_0 \times (R_0 + Z)$$

$$X_0 = \frac{N_T}{R_0 + Z} = \frac{N_T}{R_0 + 18} \qquad R_0 = \frac{N_T}{X_0} - Z = \frac{N_T}{X_0} - 18$$

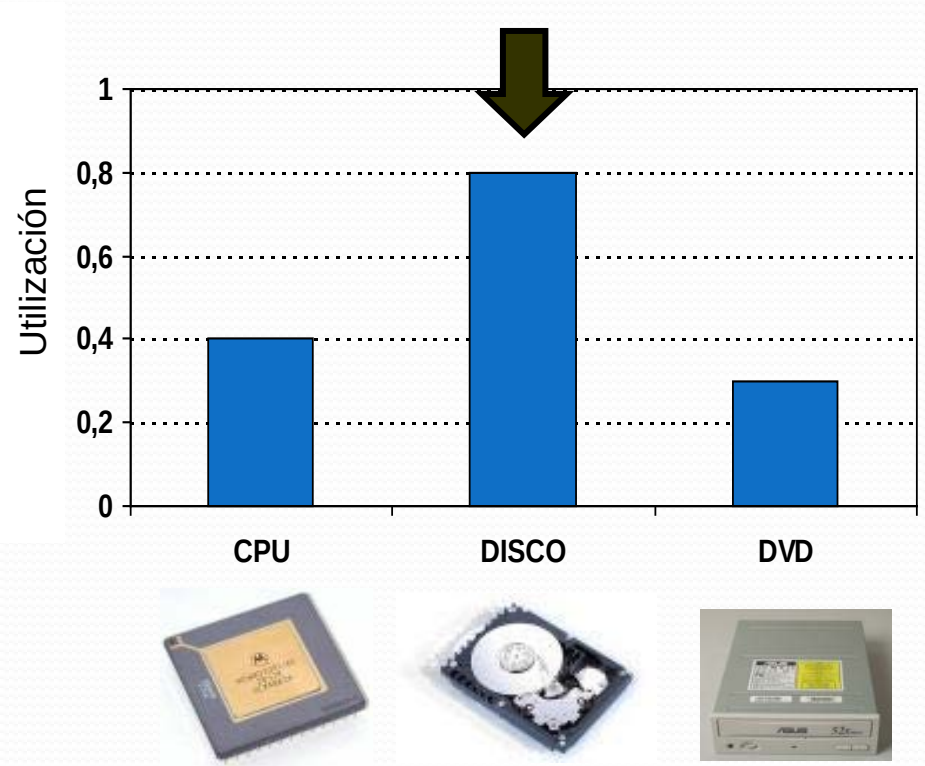
- a) Para valores de carga altos (N_T grande, **cuadro rojo**):
  - $X_0 \longrightarrow X_o^{max} = \frac{1}{D_b} = \frac{1}{5} = 0.2 \text{ tr/s}$
  - $R_0 \longrightarrow \left(\frac{N_T}{X_o^{max}}\right) - Z = D_b \times N_T - Z = 5 \times N_T - 18$
- b) Para valores de carga bajos (N_T pequeño, **cuadro verde**):
  - $R_0 \rightarrow R_o^{min} = D \equiv D_{CPU}$+DDISCOA+DDISCOB = 12s
  - $X_0 \rightarrow \frac{N_T}{R_o^{min}+Z} = \frac{N_T}{12+18} = \frac{N_T}{30}$



5.4. Técnicas de mejora

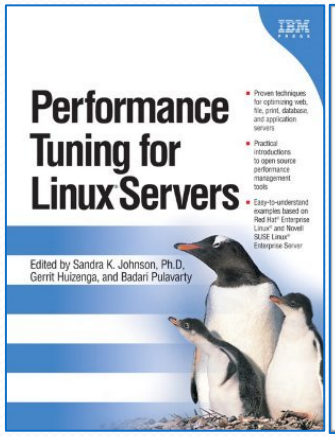
Técnicas para mejorar las prestaciones

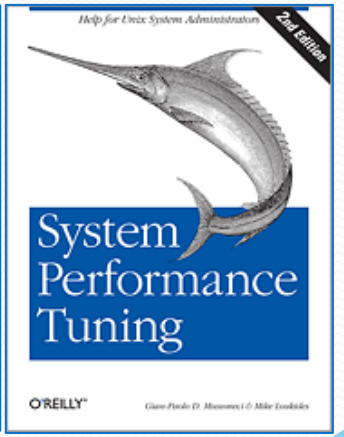
- Para mejorar las prestaciones de manera significativa hay que actuar sobre el cuello de botella del servidor: **hay que reducir** $\boldsymbol{D_b = V_b \times S_b}$. ¿Qué opciones tenemos?
  - Sintonización o ajuste.
  - Actualización y/o ampliación.



Sintonización o ajuste (*tuning*)

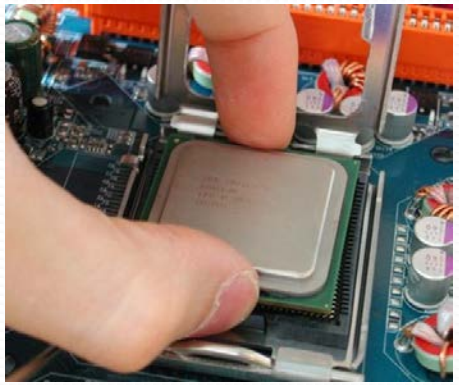
- Optimización del funcionamiento de componentes existentes:
  - A nivel hardware: Ajuste de los parámetros de la placa base (frecuencias, voltajes, RAID por hardware con mayores prestaciones), módulos DRAM en canales diferentes, etc.
  - Aplicaciones: ficheros de configuración, uso de *profilers* (si código fuente disponible). Mejora, en general, de la distribución de carga entre los recursos existentes para que se **igualen las demandas de servicio**.
  - Sistema operativo: políticas de gestión de procesos, memoria, almacenamiento y red: *sysctl, /sys, nice, renice, taskset, ulimit, chcpu, tune2fs, ionice, hdparm, ethtool, tc, route*, ...
- Algunos inconvenientes:
  - Posible alteración de la fiabilidad.
  - Hay que conocer muy bien el funcionamiento de los componentes hardware, las aplicaciones y el S.O.
  - Si hay aleatoriedad, debemos realizar tests estadísticos para ver qué factores realmente influyen en las prestaciones.





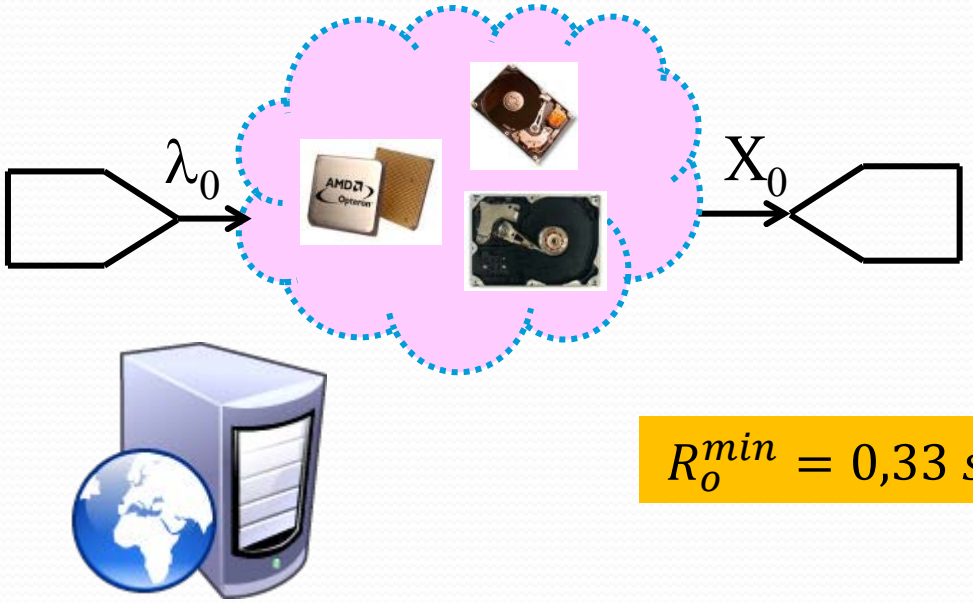
Actualización y/o ampliación

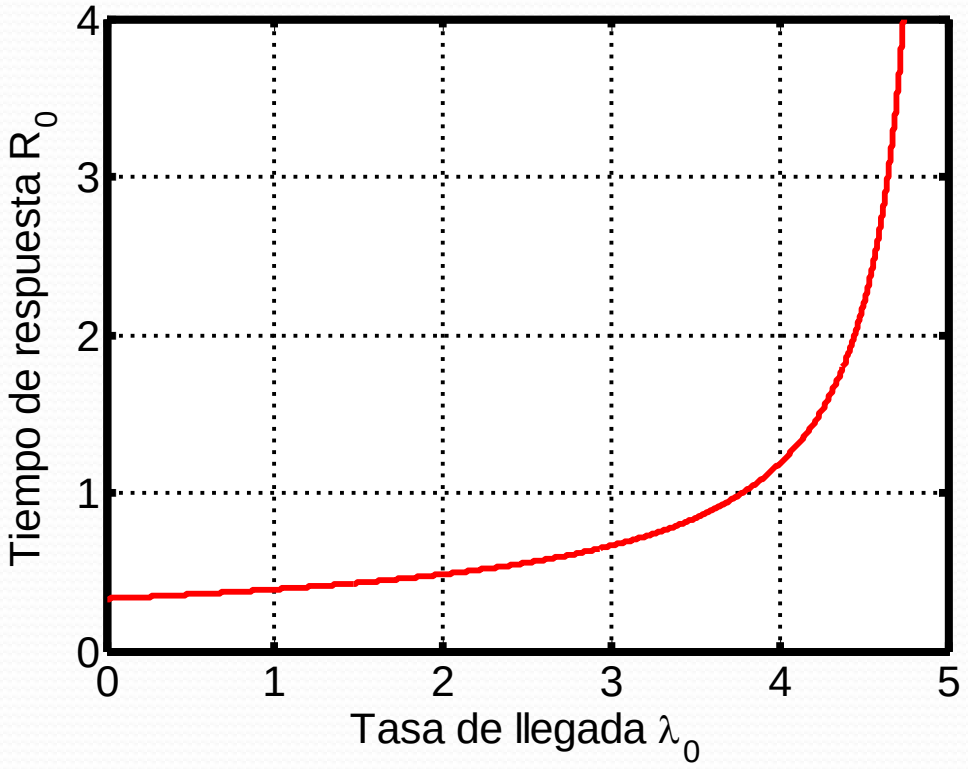
- Reemplazar dispositivos por otros más rápidos (disminuimos S_b de forma directa).
  - Procesador, memoria, disco, conexión de red, …
- Añadir dispositivos para poder realizar más tareas en paralelo (disminuimos V_b de forma directa). Idealmente, debemos re-distribuir la carga entre los componentes del mismo tipo que el dispositivo añadido para que se **igualen las demandas de servicio**.
  - Ejemplo: añadimos un procesador a la placa base, un módulo de DRAM, una nueva unidad al RAID, etc.
- Algunos problemas:
  - Facilidad del servidor para dejarse actualizar (extensibilidad/escalabilidad).
  - Compatibilidad de los nuevos elementos con los existentes.



Ejemplo: servidor web modelado mediante una red abierta

Dispositivo	V_i	S_i (s)	D_i (s)
CPU	10	0,02	**0,2**
DISCO A	4	0,02	0,08
DISCO B	5	0,01	0,05





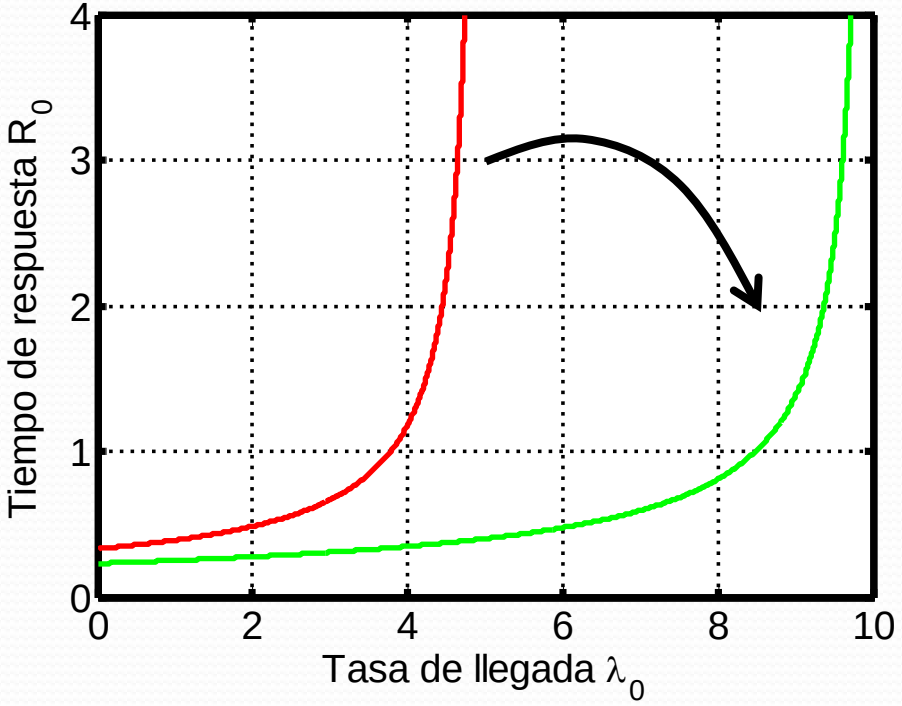
$$R_o^{min} = 0{,}33\ s$$

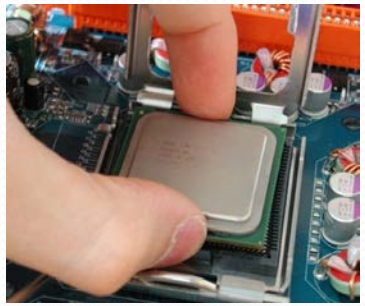
$$X_o^{max} = \frac{1}{D_b} = \frac{1}{0{,}2} = 5\ tr/s$$

Actualización: CPU doble de rápida

Dispositivo	V_i	S_i (s)	D_i (s)
CPU	10	0,01	**0,1**
DISCO A	4	0,02	0,08
DISCO B	5	0,01	0,05

La CPU se mantiene como cuello de botella pero con menor demanda





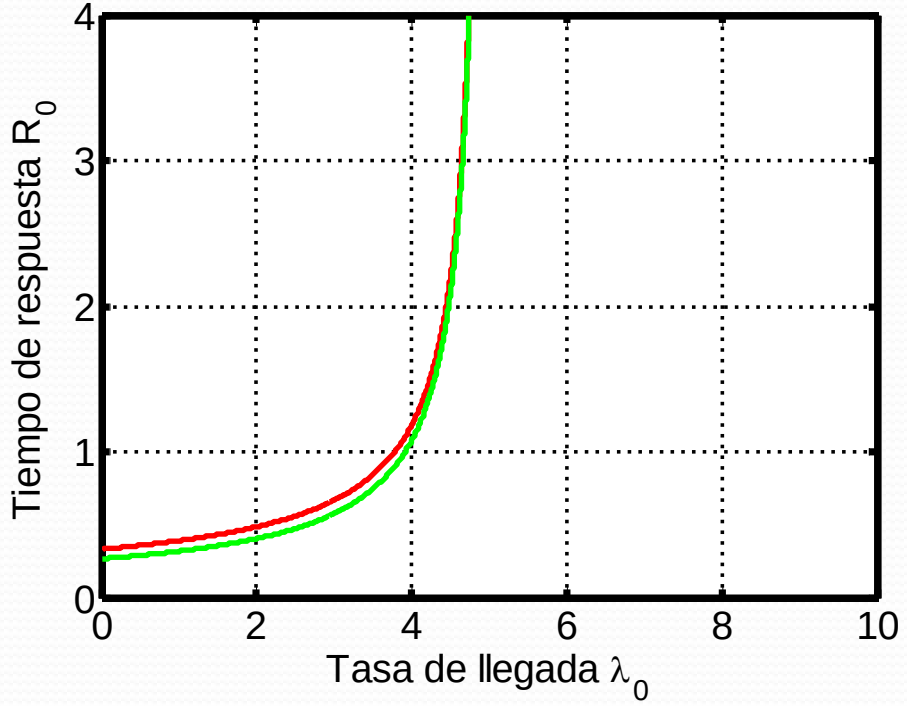
$$R_o^{min} = 0{,}23\ s$$

$$X_o^{max} = \frac{1}{D_b} = \frac{1}{0{,}1} = 10\ tr/s$$

Actualización: discos doble de rápidos

Dispositivo	V_i	S_i (s)	D_i (s)
CPU	10	0,02	**0,2**
DISCO A	4	0,01	0,04
DISCO B	5	0,005	0,025

La CPU se mantiene como cuello de botella

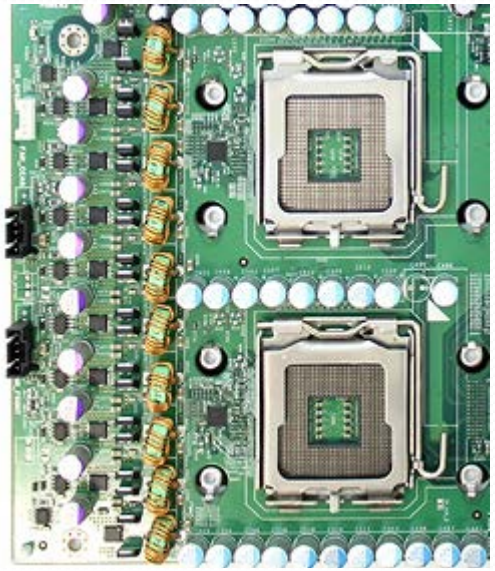


$$R_o^{min} = 0{,}265\ s$$

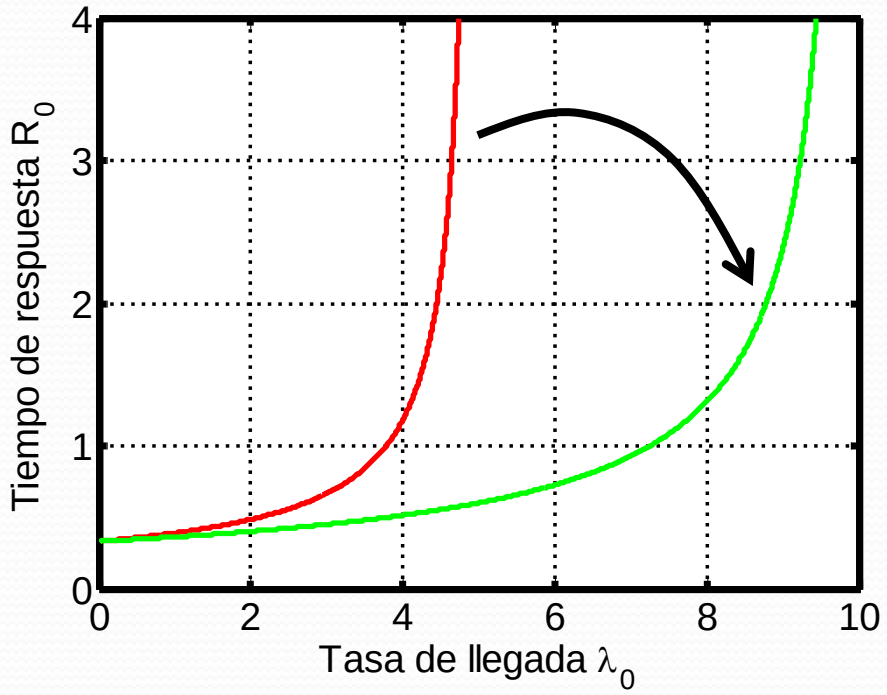
$$X_o^{max} = \frac{1}{D_b} = \frac{1}{0{,}2} = 5\ tr/s$$

Ampliación: Añadimos una segunda CPU

Dispositivo	V_i	S_i (s)	D_i (s)
CPU	5	0,02	**0,1**
DISCO A	4	0,02	0,08
DISCO B	5	0,01	0,05
CPU 2	**5**	**0,02**	**0,1**



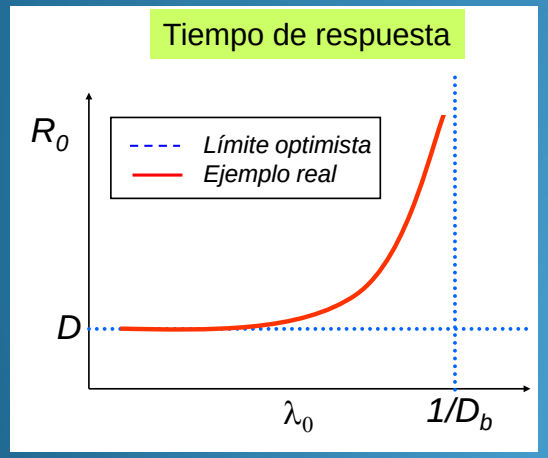
Suponemos que el S.O. equilibra la carga entre ambas CPU

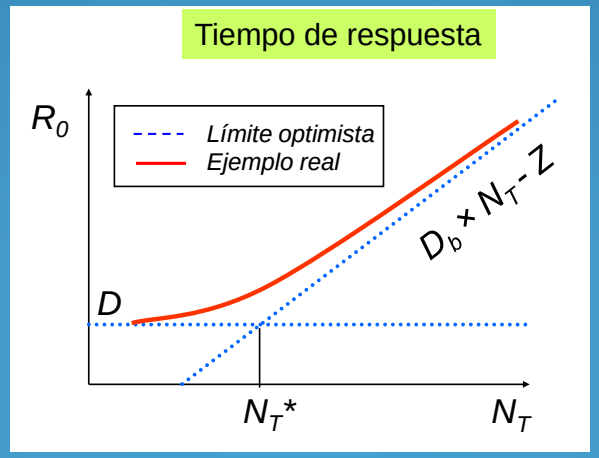


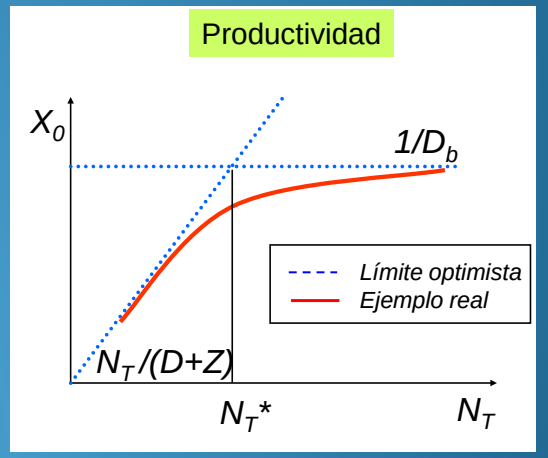
$$R_o^{min} = 0{,}33\ s$$

$$X_o^{max} = \frac{1}{D_b} = \frac{1}{0{,}1} = 10\ tr/s$$

5.5. Algoritmos de resolución de modelos de redes de colas

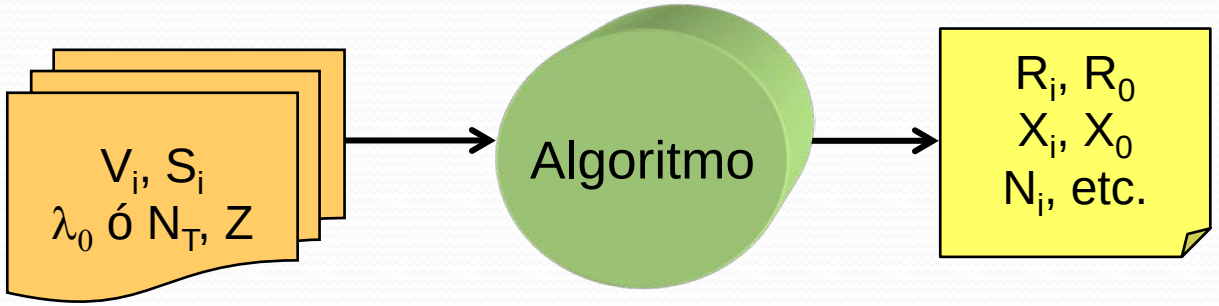




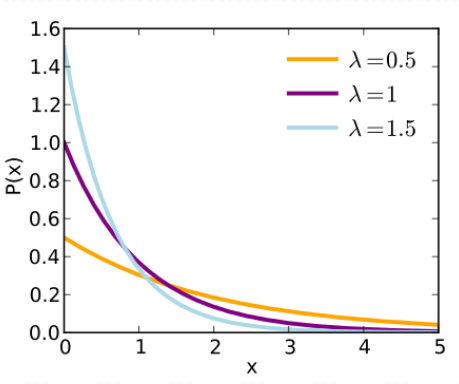


Algoritmos de resolución de redes de colas

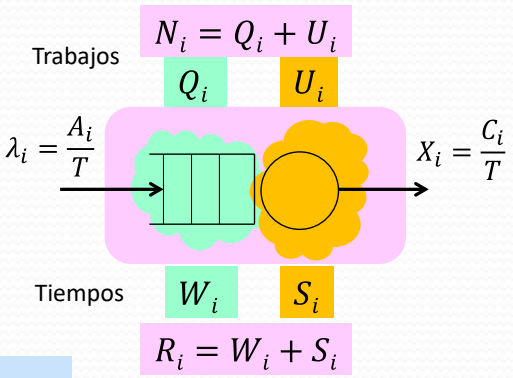
- En este apartado vamos a proporcionar una metodología (algoritmo) para resolver modelos de redes de colas. Supondremos conocido:
  - El número de estaciones de servicio (K).
  - Por cada estación:
    - Razón de visita medio de cada estación ($V_i = C_i/C_o$).
    - Tiempo de servicio medio de cada estación ($S_i = B_i/C_i$).
  - Si la red es abierta: Tasa de llegada al servidor ($\lambda_o = A_o/T$).
  - Si la red es cerrada:
    - Número total de trabajos en el sistema servidor+clientes (N_T).
    - Tiempo medio de reflexión de los clientes (Z).



Redes abiertas: hipótesis de la independencia en la llegada de trabajos



- Para **redes abiertas en equilibrio de flujo**, vamos a suponer que el momento en el que llega un trabajo es independiente de cuándo llegó el trabajo anterior (*memoryless property*). Esto se puede conseguir si suponemos que todas las distribuciones de probabilidad en la red de colas son de tipo exponencial $P(x)=\lambda e^{-\lambda x}$.
- En ese caso, se puede demostrar que cuando un trabajo llega a la estación de servicio i-ésima tiene que esperar, de media, a que se procesen los N_i trabajos que, de media, hay en la estación, uno comenzando a ser servido y el resto esperando:



$$W_i = N_i \times S_i$$

- Por lo tanto, el tiempo de respuesta medio vendrá dado por: $R_i = W_i + S_i = N_i \times S_i + S_i$
- Aplicando la ley de Little, ya que estamos en equilibrio de flujo, $N_i = X_i \times R_i$:

$$\boldsymbol{R_i} = X_i \times \boldsymbol{R_i} \times S_i + S_i \Rightarrow (1 - X_i \times S_i) \times \boldsymbol{R_i} = S_i \Rightarrow$$

$$\boldsymbol{R_i} = \frac{S_i}{1 - X_i \times S_i} = \frac{S_i}{1 - U_i} = \frac{S_i}{1 - X_0 \times D_i} = \frac{S_i}{1 - \lambda_0 \times V_i \times S_i}$$

Algoritmo de resolución de redes de colas abiertas

- Suponemos conocidos: λ_o, V_i y S_i $\forall i=1..K$, y que el servidor está en equilibrio de flujo ($X_o = \lambda_o$).

  - Paso 1.- Calculamos la demanda media de servicio de cada estación:

$$D_i = V_i \times S_i$$

  - Paso 2.- Calculamos el tiempo medio de respuesta de cada estación usando la hipótesis $W_i = N_i \times S_i$:

$$R_i = \frac{S_i}{1 - X_i \times S_i} = \frac{S_i}{1 - U_i} = \frac{S_i}{1 - X_0 \times D_i}$$

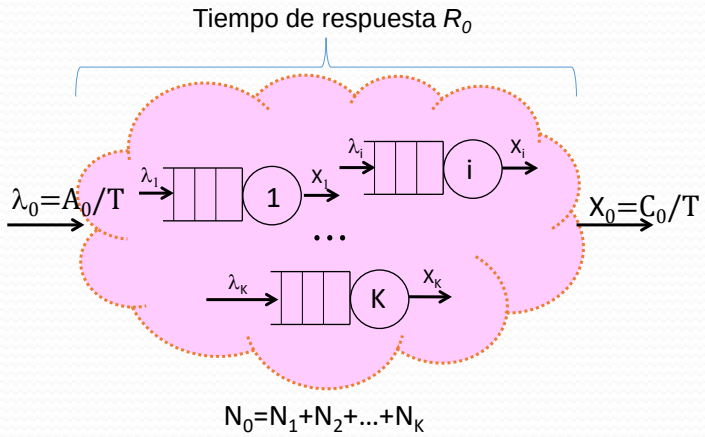
  - Paso 3.- Calculamos el tiempo medio de respuesta del servidor:

$$R_0 = \sum_{i=1}^{K} V_i \times R_i = \sum_{i=1}^{K} \frac{V_i \times S_i}{1 - X_0 \times D_i} = \sum_{i=1}^{K} \frac{D_i}{1 - X_0 \times D_i}$$

- El resto de las variables operacionales (X_i, U_i, N_i , N_o, W_i, Q_i,...) se pueden calcular usando sus expresiones habituales.

Ejemplo: resolución de redes abiertas

Recurso	V_i	S_i (s)
CPU	9	0,010
DISCO	3	0,020
RED	5	0,016



Suponiendo que la tasa de llegada de peticiones al servidor es de 5 peticiones/s:

a) Calcule las demandas de servicio de cada recurso.

b) ¿Qué recurso es el cuello de botella? ¿Cuál es la productividad máxima del servidor? ¿Está el servidor saturado?

Partiendo de la hipótesis de la independencia en la llegada de trabajos:

c) Calcule el tiempo de respuesta de cada recurso y del servidor.

d) Calcule el nº medio de clientes conectados al servidor (=trabajos en el servidor).

e) Calcule el tiempo medio de espera en la cola y el número medio de trabajos en la cola de cada recurso.

Solución del ejemplo

a)

Recurso	V_i	S_i (s)	D_i (s)	U_i
CPU	9	0,010	0,09	0,45
DISCO	3	0,020	0,06	0,30
RED	5	0,016	0,08	0,40

b) La CPU es el cuello de botella (el recurso de mayor demanda de servicio).
- Productividad máxima del servidor:

$$X_0^{max} = {}^{1}/_{D_b} = {}^{1}/_{0,09} = 11,1\ tr/s$$

Como $\lambda_0 = 5\ {}^{tr}/_{s} < X_0^{max}$ el servidor está en equilibrio de flujo (X_o=λ_0) y no saturado. Como comprobación, calculamos $U_i = X_0 \times D_i = 5\ {}^{tr}/_{s} \times D_i$ y verificamos que $U_b = U_{CPU} = 0,45 < 1$.

c) Hipótesis de la independencia en las llegadas de trabajos: $R_i = N_i \times S_i + S_i \Rightarrow R_i = \frac{S_i}{1-X_0 \times D_i}$

$$R_{CPU} = \frac{S_{CPU}}{1 - X_0 \times D_{CPU}} = \frac{0,01\ s}{1 - 5tr/s \times 0,09s} = 0,018s$$

Igualmente, $R_{DISCO} = 0,029s,\ R_{RED} = 0,027s.$

Finalmente, $R_0 = V_{CPU} \times R_{CPU} + V_{DISCO} \times R_{DISCO} + V_{RED} \times R_{RED} = 0,38\ s$

d) $N_0 = X_0 \times R_0 = 5\ {}^{tr}/_{s} \times 0,38s = 1,9\ tr = 1,9\ clientes$

Solución del ejemplo (cont.)

e)

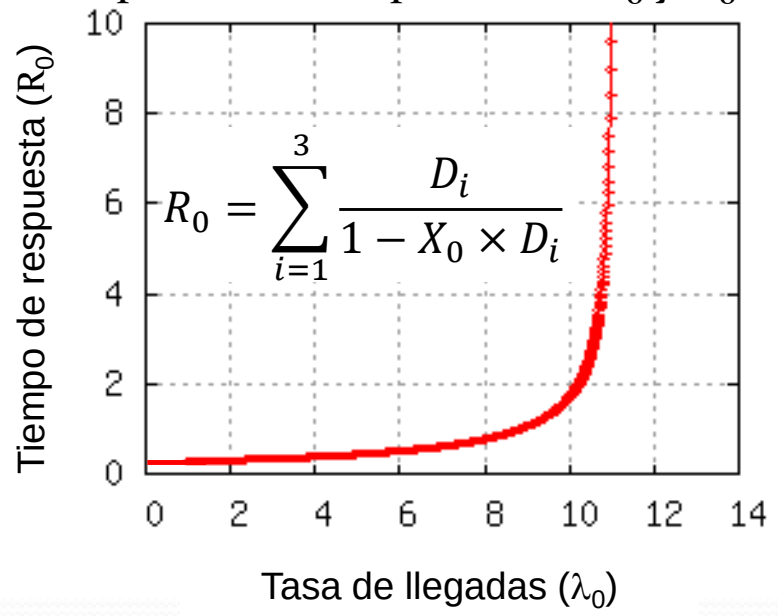
Recurso	V_i	S_i (s)	D_i (s)	U_i	R_i (s)	W_i (s)	Q_i (tr.)
CPU	9	0,010	0,09	0,45	0,018	0,008	0,37
DISCO	3	0,020	0,06	0,30	0,029	0,009	0,13
RED	5	0,016	0,08	0,40	0,027	0,011	0,27

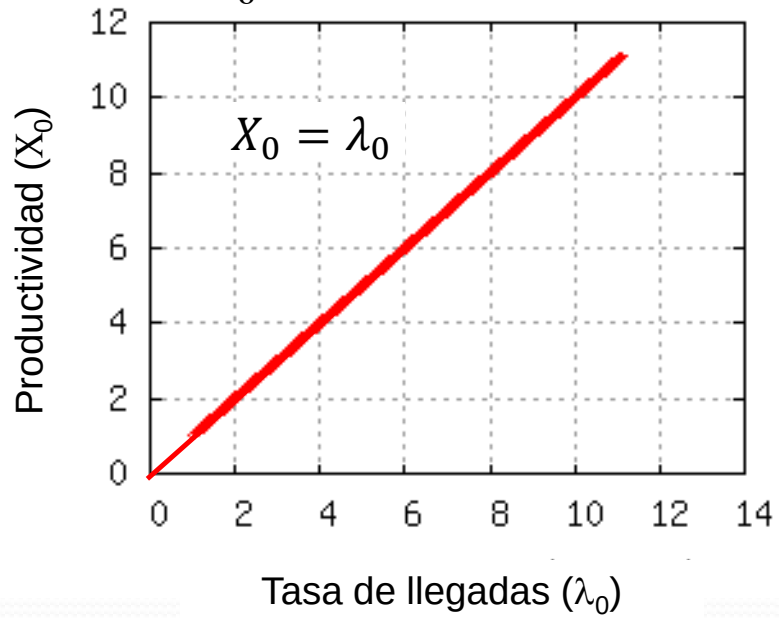
- $R_i = \frac{S_i}{1 - X_0 \times D_i}$
- $W_i = R_i - S_i$
- $Q_i = \lambda_i \times W_i = X_i \times W_i = X_0 \times V_i \times W_i$

Otra forma (sólo si independencia en las llegadas de trabajos):

- $Q_i = N_i - U_i = \frac{W_i}{S_i} - U_i$

Adicionalmente, podríamos representar R_o y X_o en función de λ_o:





Resolución con `solvenet`

- Programa muy sencillo que resuelve redes de colas utilizando los algoritmos de esta sección.
  - Disponible el código fuente en lenguaje C (SWAD).
  - Los parámetros del modelo se indican en la línea de comandos.

```
Usage: solvenet [0|1] [lambda0| NT Z] K S1 V1...SK VK
  With no parameters, shows this message
  network: 0 (open) and 1 (closed)
  lambda0: arrival rate = throughput(only open networks)
  NT:      total number of jobs in the net (only closed nets)
  Z:       think time (only interactive closed networks)
  K:       number of service stations
  Si:      service time of device i
  Vi:      ratio visit of device i
```

Resolución con `solvenet`: redes abiertas

Recurso	V_i	S_i (s)	D_i (s)
CPU	9	0.010	0.09
DISCO	3	0.020	0.06
RED	5	0.016	0.08

λ_o	5.0 trabajos/s

```
solvenet 0 5.0 3 0.01 9 0.02 3 0.016 5
```

```
*****************************************************************
*    NAME    *     Ui     *     Ni     *     Ri     *     Xi     *     Wi     *
*****************************************************************
*            *            *            *            *            *            *
* DEV 1      *     0.4500*     0.8182*     0.0182*    45.0000*     0.0082*
*            *            *            *            *            *            *
* DEV 2      *     0.3000*     0.4286*     0.0286*    15.0000*     0.0086*
*            *            *            *            *            *            *
* DEV 3      *     0.4000*     0.6667*     0.0267*    25.0000*     0.0107*
*            *            *            *            *            *            *
*****************************************************************
```

Resolución con `solvenet`: redes abiertas (II)

Recurso	V_i	S_i (s)	D_i (s)
CPU	9	0.010	0.09
DISCO	3	0.020	0.06
RED	5	0.016	0.08

λ_o	5.0 trabajos/s

```
solvenet 0 5.0 3 0.01 9 0.02 3 0.016 5
```

```
*****************************************************
*    NAME    *     Vi     *     Si     *     Di     *     Qi     *
*****************************************************
*            *            *            *            *            *
*   DEV 1    *     9.0000*     0.0100*     0.0900*     0.3682*
*            *            *            *            *            *
*   DEV 2    *     3.0000*     0.0200*     0.0600*     0.1286*
*            *            *            *            *            *
*   DEV 3    *     5.0000*     0.0160*     0.0800*     0.2667*
*            *            *            *            *            *
*****************************************************
```

Resolución con `solvenet`: redes abiertas (III)

Recurso	V_i	S_i (s)	D_i (s)
CPU	9	0.010	0.09
DISCO	3	0.020	0.06
RED	5	0.016	0.08

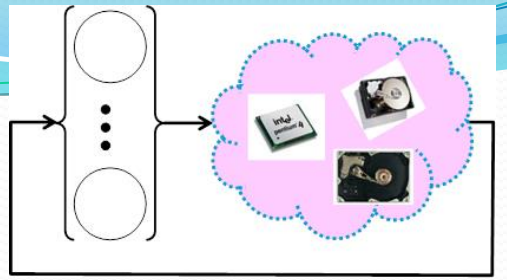
λ_o	5.0 trabajos/s

```
solvenet 0 5.0 3 0.01 9 0.02 3 0.016 5
```

```
**************************************
*          SYSTEM VARIABLES          *
**************************************
*                          *         *
* #JOBS IN SYSTEM (N0)     *    1.9134*
*                          *         *
* RESPONSE TIME (R0)       *    0.3827*
* MINIMUM RESPONSE TIME    *    0.2300*
*                          *         *
* THROUGHPUT (X0)          *    5.0000*
* MAXIMUM THROUGHPUT       *    11.111*
*                          *         *
**************************************
```

```
**************************************
*        OPTIMISTIC BOUNDS           *
**************************************
*                                    *
* R0_min =       0.2300              *
* X0_max =       11.1111             *
*                                    *
**************************************
```

Resolución de redes cerradas



- Suponemos conocidos: V_i, S_i, N_T y Z.
  - Método: Debemos ir resolviendo la red para valores incrementales del número de trabajos en la red hasta alcanzar N_T: n_T=1,…,N_T.
  - Notación: $N_i(n_T)$: Número de trabajos en la estación de servicio i-ésima si en la red hubiese n_T trabajos. Ídem para los tiempos de respuesta $R_i(n_T)$ y las productividades $X_i(n_T)$.
  - **Hipótesis de la independencia en la llegada de trabajos para redes cerradas:** $W_i(n_T) = N_i(n_T - 1) \times S_i \Longrightarrow \boldsymbol{R_i(n_T) = (N_i(n_T - 1) + 1) \times S_i}$

$For\ i = 1\ to\ K\ do\ N_i(0) = 0$ ← Inicialización del nº de trabajos en cada estación

$For\ n_T = 1\ to\ N_T\ do$

$For\ i = 1\ to\ K\ do\ \boldsymbol{R_i(n_T) = (N_i(n_T - 1) + 1) \times S_i}$ ← **Hipótesis de la independencia en la llegada de trabajos (redes cerradas)**

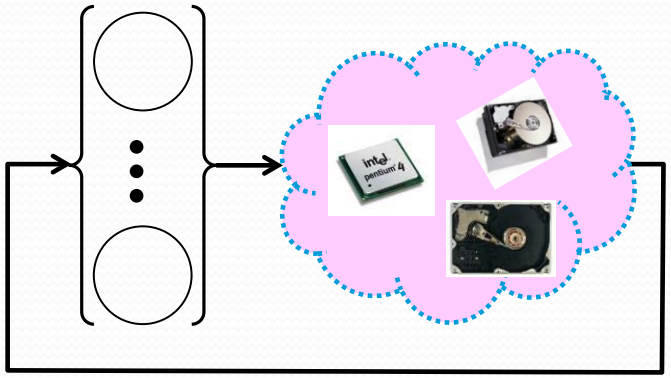
$$R_0(n_T) = \sum_{i=1}^{K} V_i \times R_i(n_T)\,, X_0(n_T) = \frac{n_T}{Z + R_0(n_T)}$$ ← Tiempo de respuesta y productividad del servidor

$For\ i = 1\ to\ K\ do\ N_i(n_T) = X_0(n_T) \times V_i \times R_i(n_T)$

Actualización del número de trabajos en cada estación

Ejemplo: resolución de redes cerradas

T. reflexión (Z)		2 s
Recurso	V_i	S_i (s)
CPU	10	0,01
DISCO1	5	0,02
DISCO2	4	0,03



$N_{CPU}(0) = N_{DISCO1}(0) = N_{DISCO2}(0) = 0$

$n_T = 1$

$R_{CPU}(1) = S_{CPU} = 0{,}01s$

$R_{DISCO1}(1) = S_{DISCO1} = 0{,}02s$

$R_{DISCO2}(1) = S_{DISCO2} = 0{,}03s$

$R_0(1) = 10 \times 0{,}01 + 5 \times 0{,}02 + 4 \times 0{,}03 = 0{,}32s$

$$X_0(1) = \frac{1}{2 + 0{,}32} = 0{,}43\ trabajos/s$$

$N_{CPU}(1) = 0{,}43 \times 10 \times 0{,}01 = 0{,}043$

$N_{DISCO1}(1) = 0{,}43 \times 5 \times 0{,}02 = 0{,}043$

$N_{DISCO1}(1) = 0{,}43 \times 4 \times 0{,}03 = 0{,}052$

Ejemplo: resolución de redes cerradas (II)

$N_{CPU}(1) = 0{,}043; N_{DISCO1}(1) = 0{,}043$

$N_{DISCO2}(1) = 0{,}052$

$n_T = 2$

$R_{CPU}(2) = (0{,}043 + 1) \times S_{CPU} = 0{,}01043s$

$R_{DISCO1}(2) = (0{,}043 + 1) \times S_{DISCO1} = 0{,}0209s$

$R_{DISCO2}(2) = (0{,}052 + 1) \times S_{DISCO2} = 0{,}0316s$

$$R_0(2) = \sum_{i=1}^{3} V_i \times R_i(2) = 0{,}3348s$$

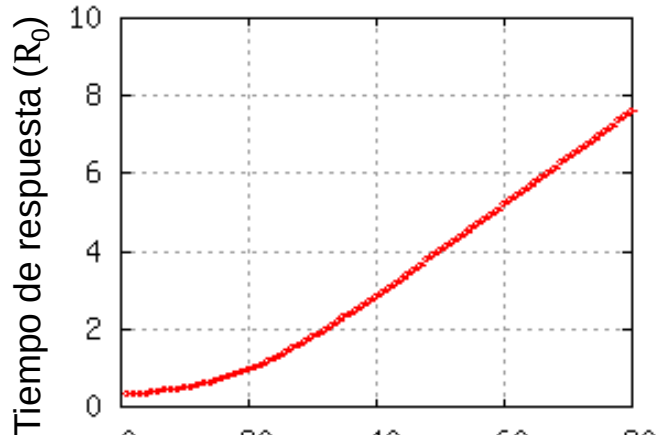
$$X_0(2) = \frac{2}{2 + 0{,}3348} = 0{,}857\ trabajos/s$$

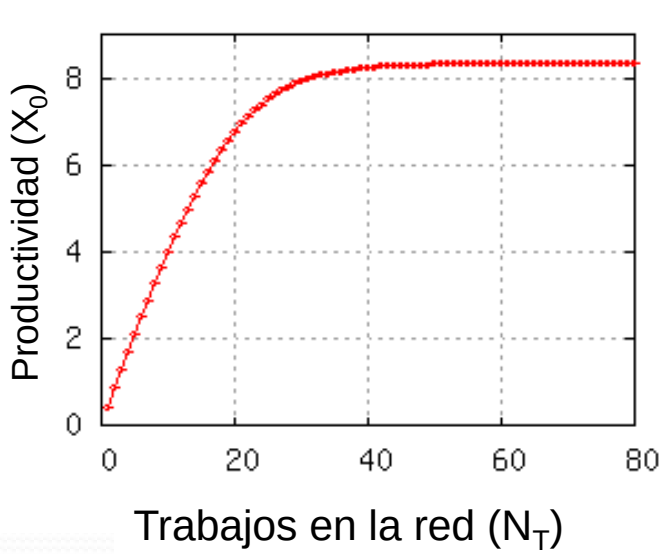
$N_{CPU}(2) = 0{,}857 \times 10 \times 0{,}01043 = 0{,}0894$

$N_{DISCO1}(2) = 0{,}857 \times 5 \times 0{,}0209 = 0{,}0894$

$N_{DISCO1}(2) = 0{,}857 \times 4 \times 0{,}03348 = 0{,}1081$

etc. hasta llegar al valor n_t=N_T que nos pidan





Resolución con `solvenet`: redes cerradas

Recurso	V_i	S_i (s)
CPU	10	0.01
DISCO1	5	0.02
DISCO2	4	0.03

T. reflexión (Z)	2 s
Nº Trabajos en la red cerrada (N_T)	2

```
solvenet 1 2 2 3 0.01 10 0.02 5 0.03 4
```

```
*****************************************************************
*    NAME    *     Ui     *     Ni     *     Ri     *     Xi     *     Wi     *
*****************************************************************
*            *            *            *            *            *            *
* DEV 1      *     0.0857*     0.0894*     0.0104*     8.5659*     0.0004*
*            *            *            *            *            *            *
* DEV 2      *     0.0857*     0.0894*     0.0209*     4.2830*     0.0009*
*            *            *            *            *            *            *
* DEV 3      *     0.1028*     0.1081*     0.0316*     3.4264*     0.0016*
*            *            *            *            *            *            *
*****************************************************************
```

Resolución con `solvenet`: redes cerradas (II)

Recurso	V_i	S_i (s)
CPU	10	0.01
DISCO1	5	0.02
DISCO2	4	0.03

T. reflexión (Z)	2 s
Nº Trabajos en la red cerrada (N_T)	2

```
solvenet 1 2 2 3 0.01 10 0.02 5 0.03 4
```

```
*****************************************************
*    NAME    *     Vi     *     Si     *     Di     *     Qi     *
*****************************************************
*            *            *            *            *            *
*   DEV 1    *    10.0000*     0.0100*     0.1000*     0.0037*
*            *            *            *            *            *
*   DEV 2    *     5.0000*     0.0200*     0.1000*     0.0037*
*            *            *            *            *            *
*   DEV 3    *     4.0000*     0.0300*     0.1200*     0.0053*
*            *            *            *            *            *
*****************************************************
```

Resolución con `solvenet`: redes cerradas (III)

Recurso	V_i	S_i (s)
CPU	10	0.01
DISCO1	5	0.02
DISCO2	4	0.03

T. reflexión (Z)	2 s
Nº Trabajos en la red cerrada (N_T)	2

```
solvenet 1 2 2 3 0.01 10 0.02 5 0.03 4
```

```
************************************
*         SYSTEM VARIABLES         *
************************************
*                        *         *
* #JOBS IN SYSTEM (N0)   *    0.2868*
* #INTERACTIVE USERS(NZ)*    1.7132*
* #JOBS IN THE NET (NT) *         2*
* KNEE POINT (NT*)       *   19.3333*
*                        *         *
* RESPONSE TIME (R0)     *    0.3348*
* MINIMUM RESPONSE TIME *    0.3200*
*                        *         *
* THROUGHPUT (X0)        *    0.8566*
* MAXIMUM THROUGHPUT     *    8.3333*
*                        *         *
************************************
```

```
************************************
*       OPTIMISTIC BOUNDS          *
************************************
*                                  *
* R0 >= max{0.32,  0.12*NT-2.00}   *
*                                  *
* X0 <= min {NT/2.32,  8.33}       *
*                                  *
*                                  *
************************************
```