

Herramienta: Simulación

Grupo de Redes, Sistemas y Servicios Telemáticos
<http://www.tlm.unavarra.es>

Programa de doctorado de
Tecnologías para la gestión distribuida de la información

Contenido

- Introducción
- Ejemplos
- Simulación de eventos discretos
- Variables aleatorias

Introducción

¿ Cómo evaluar un sistema ?

- Medirlo (experimentos)
 - ¿Y si no existe?
 - ¿Y si es muy caro o costoso hacer cambios en él?
(para preguntas “what if”)
- Análisis matemático
 - Solo para sistemas simples
 - Sistemas reales son complejos
- Simulación
 - Reconstruir el comportamiento del sistema en un programa
- Emulación
 - *Reproducir* el comportamiento mediante un programa

Simulación

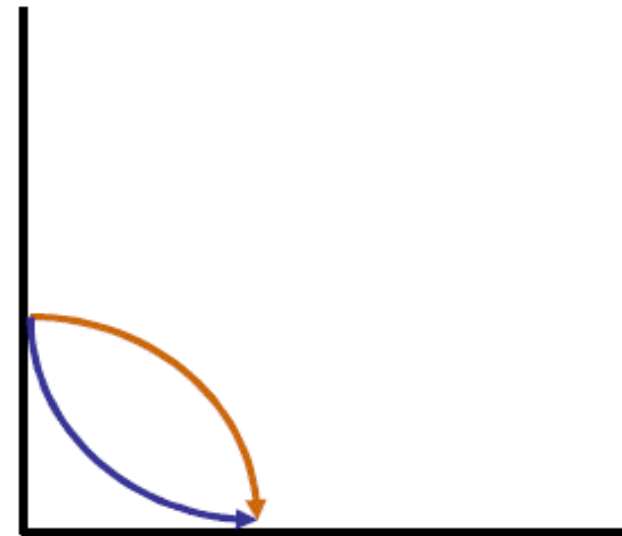
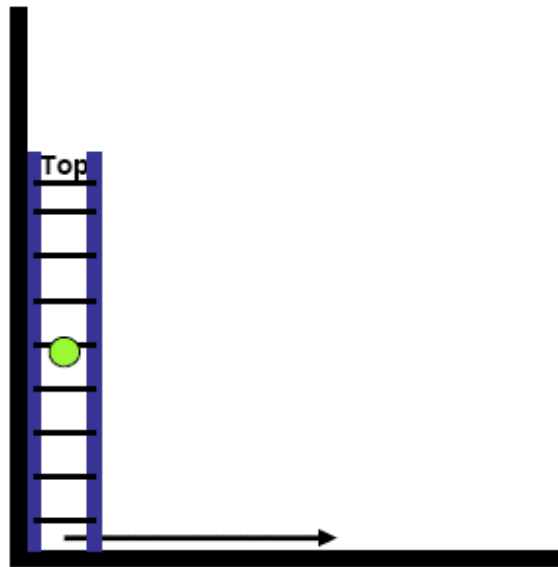
- Imitar el funcionamiento de un sistema real con el tiempo
- Esto no es nuevo...
- Trataremos simulaciones informáticas



- Se necesita un *modelo* del sistema real
- Se genera una historia artificial de sucesos en el sistema y sus repercusiones
- Se obtienen medidas (de prestaciones)
- Si el modelo es muy simple se puede resolver matemáticamente
- Modelos realistas son demasiado complejos para una solución analítica

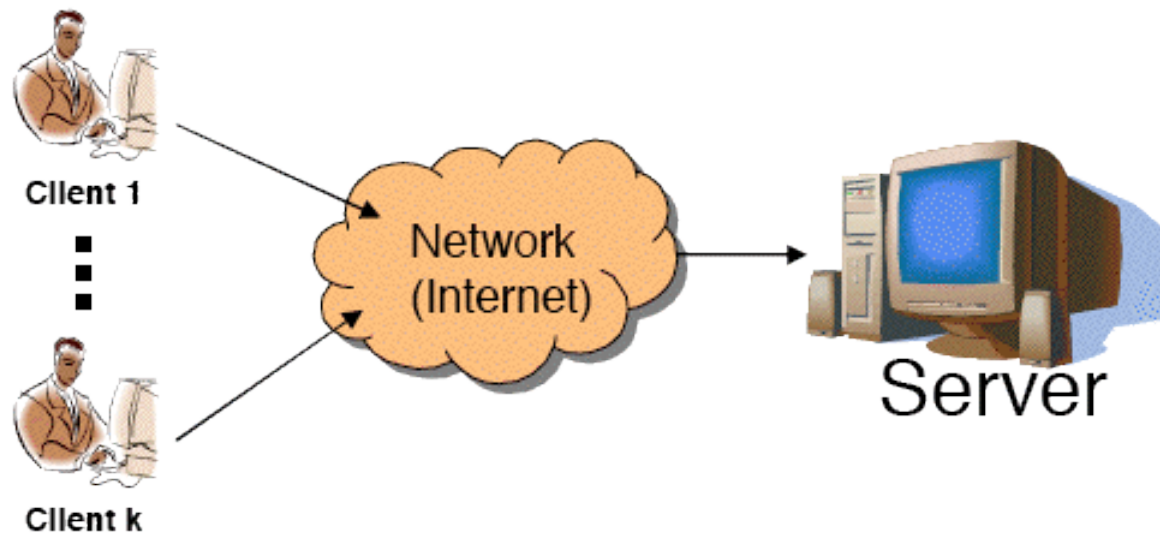
Ejemplo

- Desplazar una escalera de la pared tirando de la parte inferior
- ¿ Qué figura dibuja el centro de la escalera ?



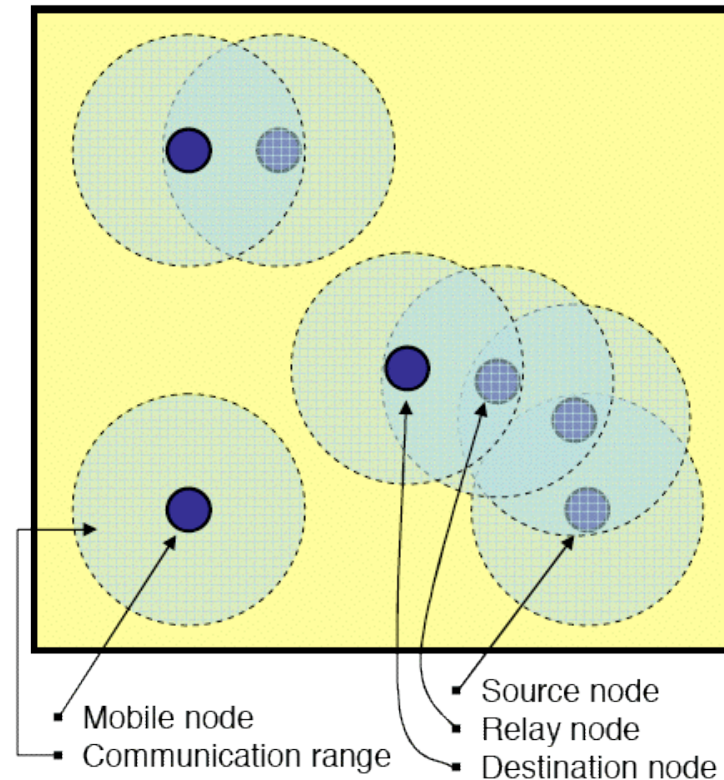
Ejemplo

- Clientes solicitan un servicio a través de una red
 - Clientes: usuario y navegador web
 - Servidor: servidor web
 - Servicio: página web
 - Red: Internet
- Analizar el rendimiento del servidor y de la red



Ejemplo

- Mobile Ad-hoc NETwork (MANET)
- Dos nodos se pueden comunicar si están dentro del alcance
- Los nodos pueden reenviar tráfico de otros
- Se mueven
- Modelar su movimiento
- Modelar el tráfico
- Resultados: prestaciones

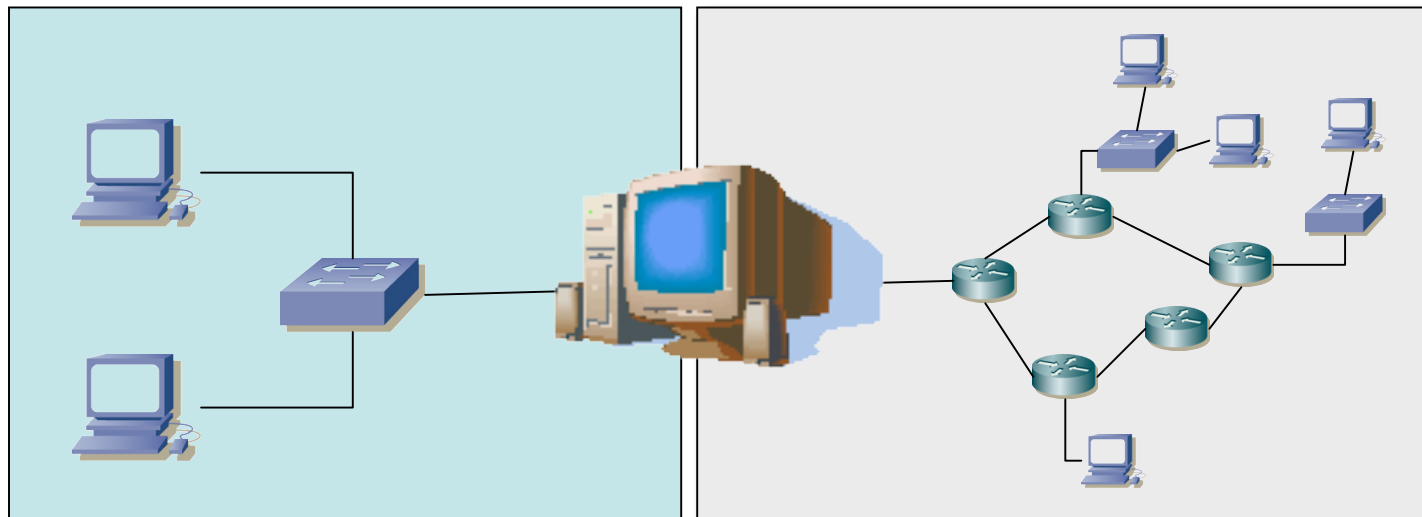
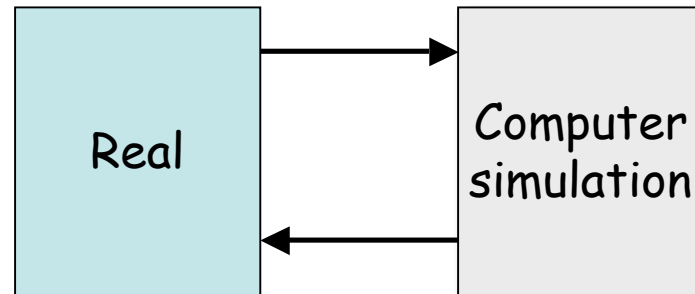


Cuándo no es apropiada

- Si el sentido común nos da la respuesta
- Si el problema se puede resolver analíticamente
- Si es más sencillo realizar experimentos
- Si el coste (€) del estudio de simulación es mayor que el ahorro posible con el conocimiento que se obtiene
- Si el sistema es demasiado complejo
- Si creemos que es la respuesta a cualquier problema.

Simulation & Emulation

- La emulación obtiene eventos del sistema real
- Devuelve eventos tras una simulación de lo que les ha sucedido
- Esos eventos de salida deben producirse en el instante real que les corresponda



Componentes

- Sistema (*system*)
 - Grupo de objetos con una interacción o interdependencia orientada hacia un propósito
- Componentes
 - Entidad (*entity*)
 - Un objeto de interés en el sistema
 - Atributo (*attribute*)
 - Propiedad de una entidad
 - Actividad (*activity*)
 - Un periodo de tiempo de una longitud especificada
- Ejemplo: Sucursal de un banco
 - Los clientes podrían ser entidades
 - Su saldo en cuenta sería un atributo
 - Hacer depósitos una actividad

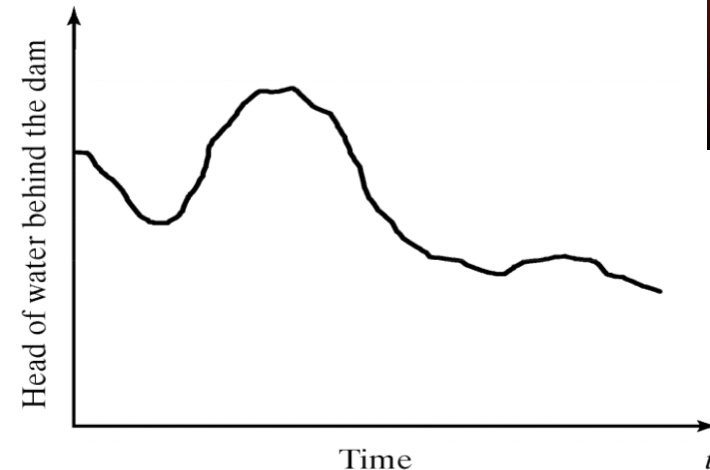
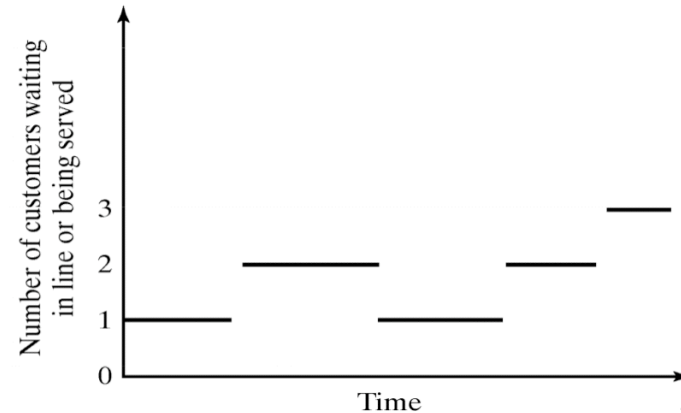


Estado del sistema

- Grupo de variables necesarias para describir el sistema en un momento cualquiera, en relación con los objetivos del estudio
- Ejemplo: banco
 - Número de cajeros ocupados
 - Número de clientes esperando en cola
 - Instante en que llegará el siguiente cliente
- **Evento:** suceso instantáneo que puede cambiar el estado del sistema (endógenos u exógenos)
- Ejemplo:
 - Llegada de un nuevo cliente (exógeno)
 - Cliente termina de ser atendido (endógeno)

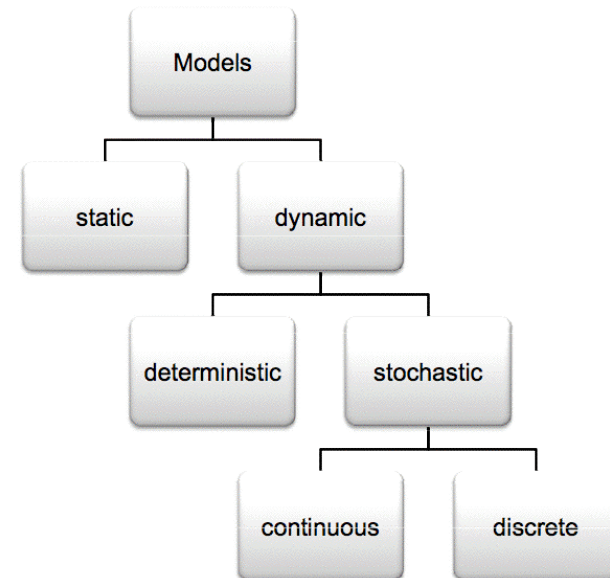
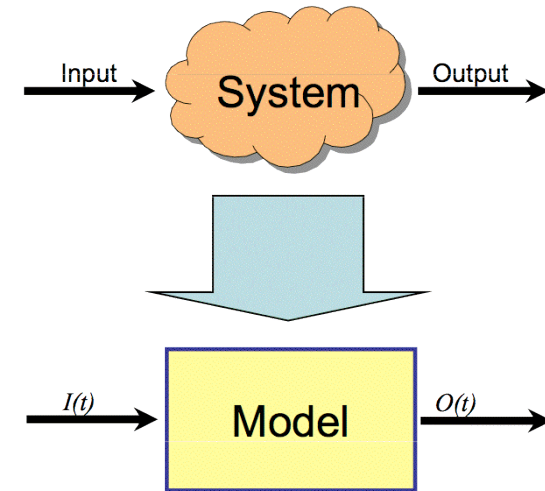
Tipos de sistemas

- Discretos o continuos
- Sistema **discreto**
 - Las variables de estado cambian solo en un conjunto discreto de puntos en el tiempo
 - Ejemplo: banco con llegadas y salidas
- Sistema **continuo**
 - Las variables de estado cambian de forma continua con el tiempo
 - Ejemplo: nivel de agua en un pantano
- Difícil ser solo de un tipo pero normalmente suelen predominar los cambios de uno de los dos tipos



Modelo del sistema

- Representación de un sistema para estudiarlo
- Simplifica el sistema
- Considera solo los aspectos que afectan al problema en estudio
- Debe ser lo suficientemente detallado para poderse obtener conclusiones que apliquen al sistema real
- Tipos:
 - Matemático/Físico
 - Estático (Monte Carlo)/Dinámico
 - Determinista/Estocástico
 - Discreto/Continuo
- Nos interesan los estocásticos, dinámicos y discretos.



Ejemplos de simulación

Simulation table

- Método para seguir el estado del sistema con el tiempo
- Metodología
 1. Determinar las características de los *inputs* a la simulación (normalmente distribuciones de probabilidad)
 2. Construir una tabla de simulación
 3. Para cada iteración i generar un valor de cada uno de los p *inputs* y evaluar la función calculando la respuesta y_i que normalmente depende de los *inputs* y de respuestas previas
- Tabla:
 - p *inputs* x_{ij} , $j = 1, 2, \dots, p$
 - Una respuesta y_i
 - Para cada iteración i

Repetitions	Inputs						Response
	x_{i1}	x_{i2}	...	x_{ij}	...	x_{ip}	
i							y_i
1							
2							
3							
...							
n							

Ejemplo: Servidor Web

- Un servidor web *single-threaded*
- Recibe peticiones de ficheros que debe obtener del disco duro
- El S.O. atiende las peticiones en serie, completando una antes de atender la siguiente
- Si el disco está ocupado el hilo del servidor web se bloqueará a la espera de que el disco finalice
- El disco es capaz de servir datos a 80 Mbps (aprox. 10MBps)



Ejemplo: Servidor Web

- Número infinito de clientes (las llegadas no cambian porque estemos atendiendo a varios)
- Entre cada par de peticiones consecutivas pasa un tiempo aleatorio (uniforme) entre 10 y 90 milisegundos
 - Media 50 mseg \Rightarrow 20 peticiones/seg
- Independientes
- Los ficheros que se solicitan son de $100 \times N$ KBytes donde N está entre 1 y 5 (igual probabilidad, independientes)
 - Media 300 KBytes \approx 2.4 Mbits/petición
- En media se solicitan 48 Mbps
- Caso peor: 500 KBytes a 80 Mbps \Rightarrow 51 mseg $>$ 10 mseg
- Se formará una cola de peticiones en el servidor



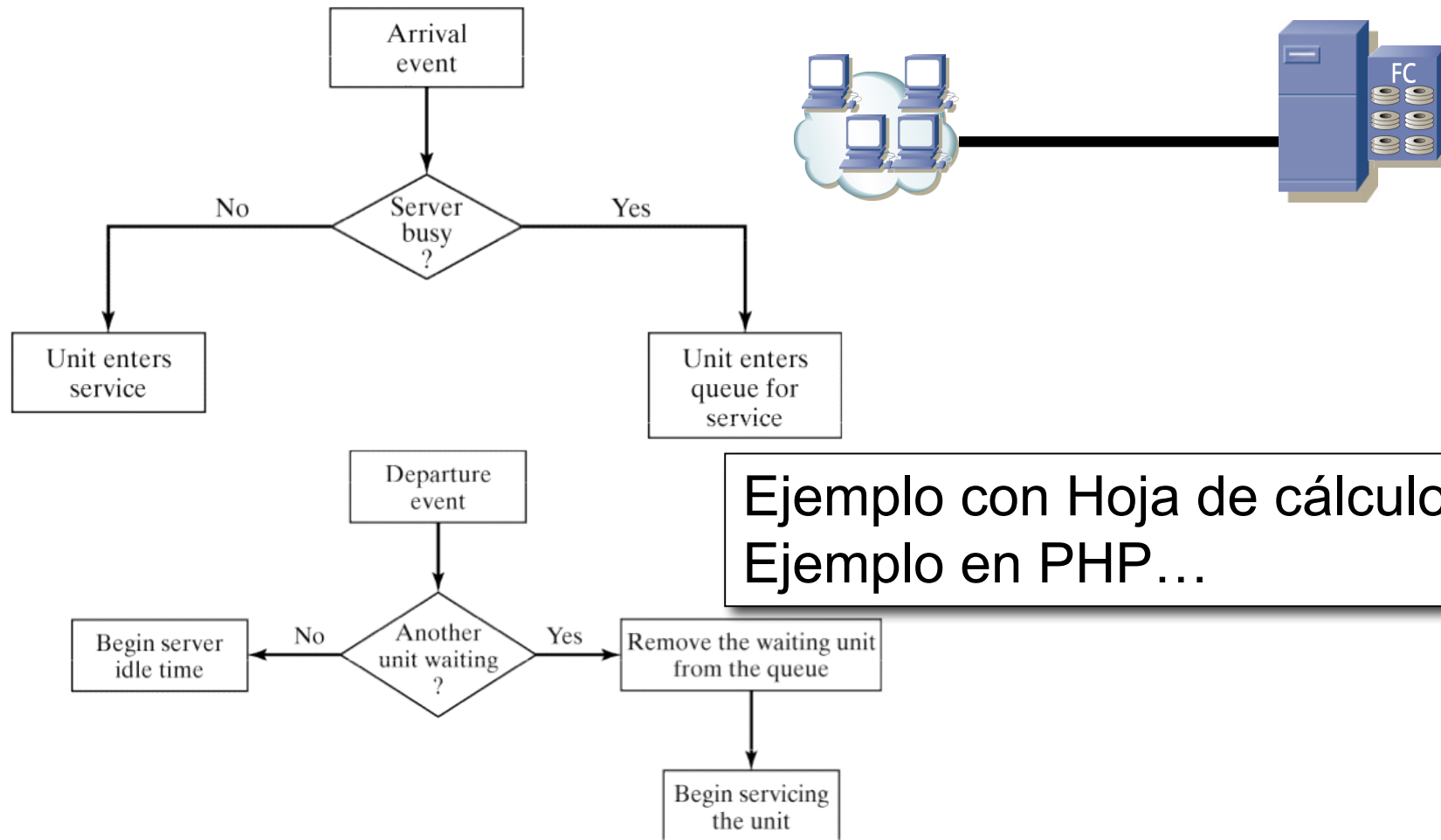
Ejemplo: Servidor Web

- Más hipótesis:
 - Ignoramos los efectos de la red
 - Servidor no es multi-hilo
 - El disco no atiende a varias peticiones a la vez
 - No contamos tiempos de búsqueda en el disco
 - Puede mantener a la espera tantas peticiones como necesite
- Preguntas:
 - ¿ Cuántas peticiones tienen que esperar a que se atienda otra ?
 - ¿ Cuánto tiene que esperar un usuario a que empiecen a servirle el fichero que ha solicitado ? ¿ Caso peor ? ¿ Media ? ¿ El 95% ?
 - ¿ Cuánto podría aumentar la carga y seguir “funcionando” el sistema ?
 - ¿ Qué velocidad de discos se necesita para una “calidad” ojetivo ?



Ejemplo: Servidor Web

- Pregunta simple: ¿ Cuánto tiene que esperar un usuario a que empiecen a servirle el fichero que ha solicitado ?



Ejemplo con Hoja de cálculo...
 Ejemplo en PHP...

Ejemplo: Servidor Web

- ¿ Es realista el modelo de usuario ?
 - ¿Uniforme el tiempo entre llegadas?
 - ¿Y peticiones a ráfagas? (html + imágenes)
 - ¿Ficheros tamaños uniformes?
 - ¿A partir de cuántos usuarios es razonable suponer una población “infinita”?
- ¿ De verdad puedo ignorar la red ?
 - ¡ Es un flujo en media de 48 Mbps y con picos de 240 Mbps !
 - TCP: RTT, pérdidas, control de flujo



Simulación de eventos discretos

Simulación de eventos discretos

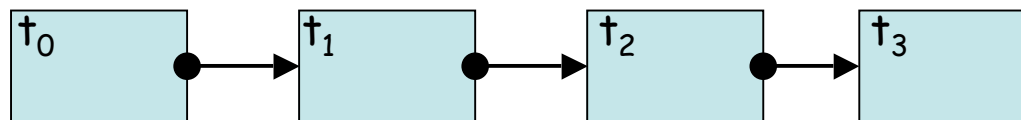
- Modelado con el tiempo de un sistema en el que todos los cambios de estado se producen en un conjunto discreto de puntos en el tiempo
- Empleo de métodos numéricos
 - En vez de métodos analíticos
 - El modelo se “corre” en vez de se “resuelve”
- Se lleva a cabo produciendo una secuencia de *snapshots* del sistema con el tiempo
- El *snapshot* en un instante t incluye
 - El estado del sistema en el instante t
 - Una lista de las actividades en progreso y cuándo terminarán
 - El estado de todas las entidades
 - Los valores de todos los contadores estadísticos

Future (pending) Events List (FEL)

- Es el mecanismo para hacer avanzar la simulación
- La FEL contiene los eventos planificados para este instante o posteriores aún sin procesar
- Cada evento contiene el instante de tiempo en que sucede
- Ordenados por instante de tiempo de menor a mayor
- Garantiza que los eventos tienen lugar en orden cronológico

Gestión de actividades con una FEL

- Duración actividad se conoce al comenzar (determinista o aleatoria)
- En algunos entornos existe la posibilidad de cancelar
- Al comenzar la actividad se introduce un evento de finalización de la actividad en la FEL



CLOCK = $t < t_0$

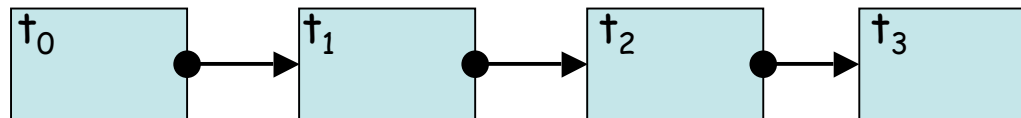
$$t_0 \leq t_1 \leq t_2 \leq t_3$$

Información fundamental a definir

- ¿Efectos de cada tipo de evento?
 - Cambios de estado
 - Cambios de atributos de entidades
- ¿Cómo se definen las actividades?
 - Deterministas, probabilísticas, ecuaciones
 - Qué tipo de evento marca su principio/final
 - Su comienzo es condicional al estado?
- ¿Cómo comienza la simulación?
 - Primeros eventos
- ¿Cuándo finaliza la simulación?

Avance de la simulación

- *Snapshots* del sistema con el tiempo
- *Snapshot* incluye el estado del sistema y la FEL
- Esa FEL contiene las actividades en progreso y cuándo finalizan
- CLOCK = t = instante actual en la simulación
- Evento en t_0 = Evento inminente
- Se actualiza CLOCK = t_0
- Se retira el evento inminente de la FEL
- Se “ejecuta” el evento
- Eso crea un nuevo *snapshot* del sistema



CLOCK = t < t_0

$$t_0 \leq t_1 \leq t_2 \leq t_3$$

Event-scheduling/Time-advance

Mientras queden eventos en la FEL

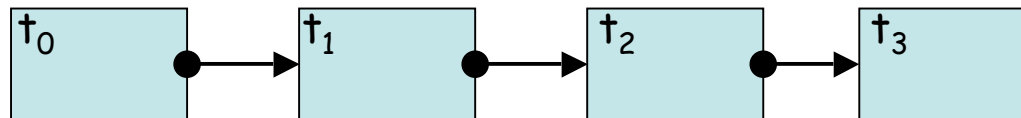
Retirar el primero (evento inminente)

Avanzar la variable de CLOCK hasta el instante del evento

Procesar el evento: puede modificar el estado del sistema e introducir otros eventos futuros en la FEL

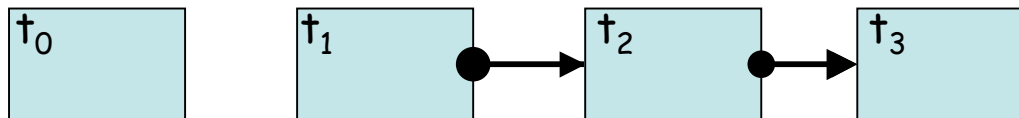
Generar eventos futuros (si es necesario) y colocarlos en la FEL manteniéndola ordenada

Actualizar los contadores y estadísticos



Ejemplo: Avance con Llegadas

- Evento inicial: una llegada
- Se procesa
 - Introduce eventos consecuencia de ella (...)
 - Se introduce un nuevo evento que es la siguiente llegada (...)



¿ Fin de la simulación ?

- Cuando no queden eventos en la FEL
- En la inicialización introducir un evento futuro de finalización
 - Limita el tiempo simulado
 - No limita el tiempo real
- Detenerla al alcanzar una duración (tiempo real)
- Detenerla al alcanzar unas medidas una cierta precisión
- Número máximo de eventos (de algún tipo) a procesar

Gestión de la FEL

- Su longitud cambia durante toda la simulación
- Su gestión eficiente es vital
- Operaciones más frecuentes:
 - Retirar el primero
 - Insertar manteniendo el orden
- Puede soportar el eliminar un evento en concreto

Simulation Tools

- Librerías de utilidades
- Simuladores programables
- Simuladores controlables (gráfico, script)
- Simuladores de redes (ns2, OMNeT++, SSFNet, Parsec, Qualnet, OPNET, JiST/SWANS ...)

Ejemplo: Servidor Web (again)

- El disco es capaz de servir datos a 10 MBps (80 Mbps)
- Tiempo entre llegadas uniforme [10, 90] milisegundos
 - Media 50 mseg \Rightarrow 20 peticiones/seg
- Independientes
- Ficheros de [1, 5]x100 KBytes
 - Media 300 KBytes \approx 2.4 Mbits/petición
- Nueva pregunta:
 - ¿Cuántas peticiones hay esperando?

Ejemplo en PHP...
Ejemplo con OMNeT++...



Variables aleatorias

¿ Por qué ?

- El mundo es más fácil de describir de forma probabilística que determinista
- ¡ Es que hay demasiados factores !
- Recordar conceptos

Probability

- A **random variable** (r.v.) X is the outcome of a random event expressed as a numeric value
- The *Cummulative Distribution Function (CDF)* provides the fixed probability that the r.v. will not exceed a value x

$$CDF(X) \equiv F_X(x) \equiv P(X \leq x)$$

- The *Complementary Cummulative Distribution Function (CCDF)*:

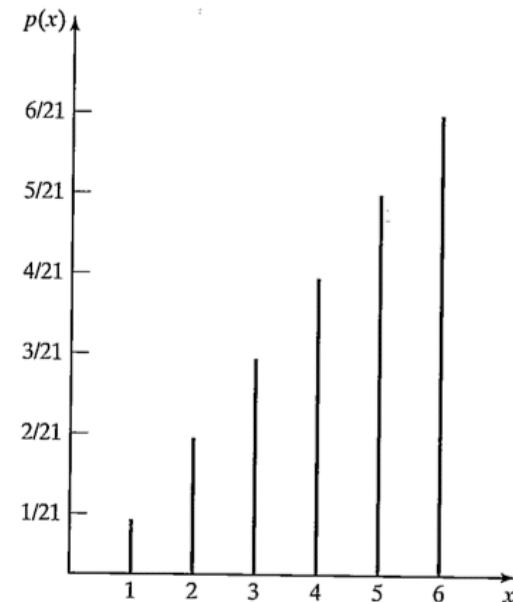
$$CCDF(X) \equiv \bar{F}_X(x) \equiv 1 - F_X(x) \equiv P(X > x)$$

- **Discrete** r.v. : takes values from a finite or a countably infinite set of values
- *Probability Distribution Function (PDF)* o *Probability Mass Function* of a discrete r.v. :

$$p_X[x_i] \equiv P(X = x_i)$$

$$p_X[x_i] \geq 0$$

$$\sum_{i=1}^{\infty} p_X[x_i] = 1$$



Continuous r.v.

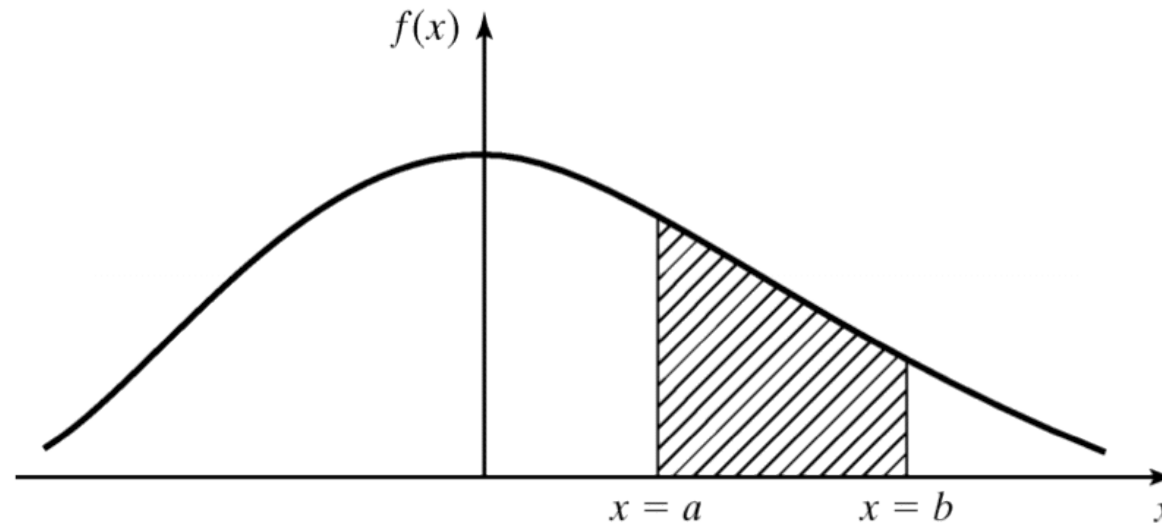
- **Continuous** r.v. : takes values from an uncountably infinite set of values R_X
- *Probability Distribution Function (PDF)* of a continuous r.v. :

$$f_X(x) \equiv \frac{dF_X(x)}{dx} = \frac{dP(X \leq x)}{dx}$$

$$P(x_1 < X \leq x_2) = F_X(x_2) - F_X(x_1) = \int_{x_1}^{x_2} f_X(u) du$$

$$f_X(x) \geq 0 \quad (x \in R_X) \quad \int_{R_X} f_X(x) dx = 1$$

$$P(x_1 < X \leq x_2) = P(x_1 \leq X \leq x_2) = P(x_1 \leq X < x_2) = P(x_1 < X < x_2)$$



Moments

- *Expected value* of a continuous random variable X (mean):

$$E[X] \equiv \mu_X \equiv \int_{-\infty}^{\infty} u f_X(u) du$$

- *nth moment* of X :

$$E[X^n] \equiv \int_{-\infty}^{\infty} u^n f_X(u) du$$

- Related with the variability is the *variance* :

$$\text{Var}(X) \equiv \sigma_X^2 = E[(X - \mu_X)^2] = \int_{-\infty}^{\infty} (u - \mu_X)^2 p(u) du = E[X^2] - (E[X])^2 = E[X^2] - \mu_X^2$$

- *Standard deviation*:

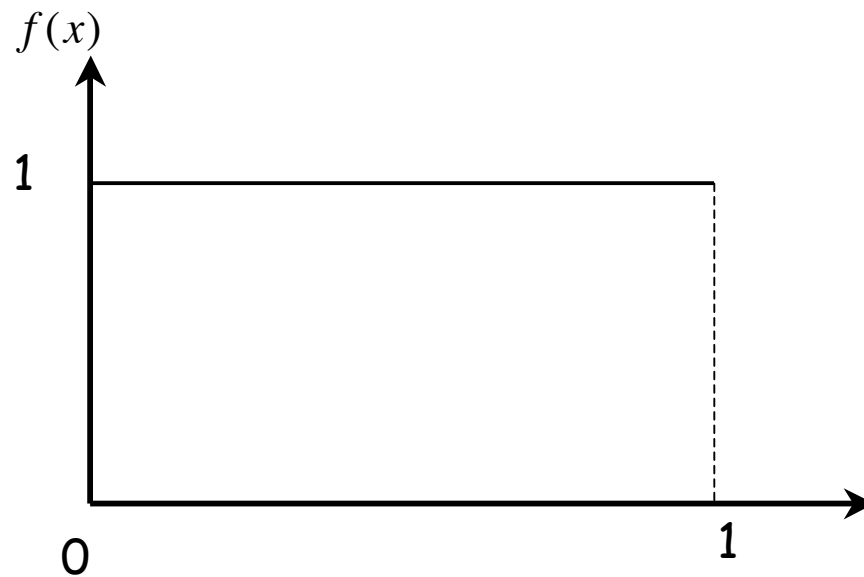
$$\sigma_X \equiv \sqrt{\text{Var}(X)}$$

Commonly Encountered Distributions

Distribution	Definition	Domain
Exponential	$p(x) = \lambda e^{-\lambda x}$	$x > 0$
Normal	$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$	$-\infty < x < \infty$
Gamma	$p(x) = \frac{(x-\gamma)^{\alpha-1} \exp[-(x-\gamma)/\beta]}{\beta^\alpha \Gamma(\alpha)}$	$x > \gamma$
Extreme	$F(x) = \exp\left[-\exp\left(-\frac{(x-\alpha)}{\beta}\right)\right]$	$-\infty < x < \infty$
Lognormal	$p(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\log x - \mu}{\sigma}\right)^2\right]$	$x > 0$
Pareto	$p(x) = \alpha k^\alpha x^{-\alpha-1}$	$x > k$
Weibull	$p(x) = \frac{bx^{b-1}}{a^b} \exp\left[-\left(\frac{x}{a}\right)^b\right]$	$x > 0$

Generación de números aleatorios

- Se busca generarlos con distribución uniforme
- Independientes



Números pseudo-aleatorios

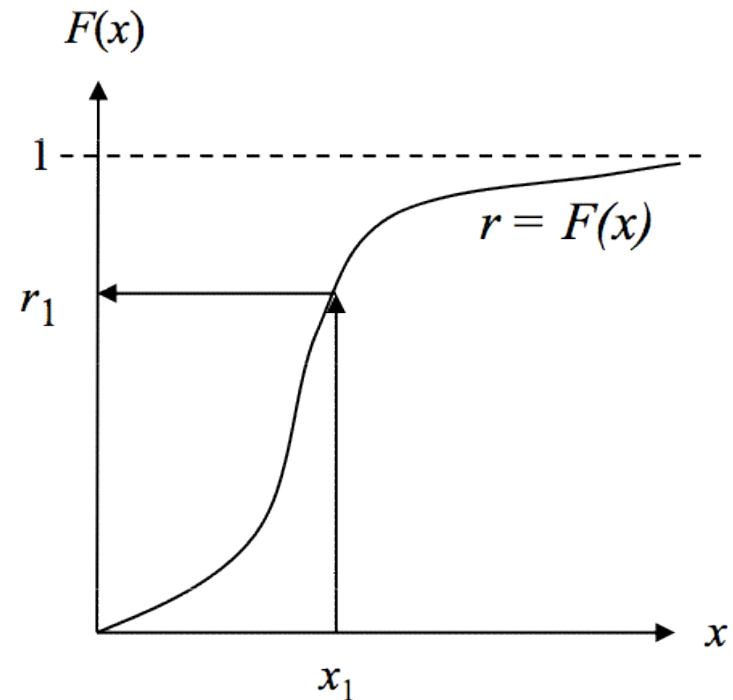
- Completamente predecibles
- Generar números en $[0,1]$ que imitan el comportamiento de los aleatorios
- Tienen un periodo de repetición
- Ejemplo: Linear Congruential Method

$$X_{i+1} = (aX_i + c) \bmod m$$

- ¿ Y una variable no uniforme ?

Inverse-transform Technique

- $F(x)$ CDF de la variable a generar
- X v.a. Uniforme $[0, 1]$
- Generar r_1 con X
- Invertir el valor con $F^{-1}(x)$
- Más sencillo si F es invertible



Ejemplo: Exponencial

$$f(x) = \lambda e^{-\lambda x}$$

$$F(x) = 1 - e^{-\lambda x}$$

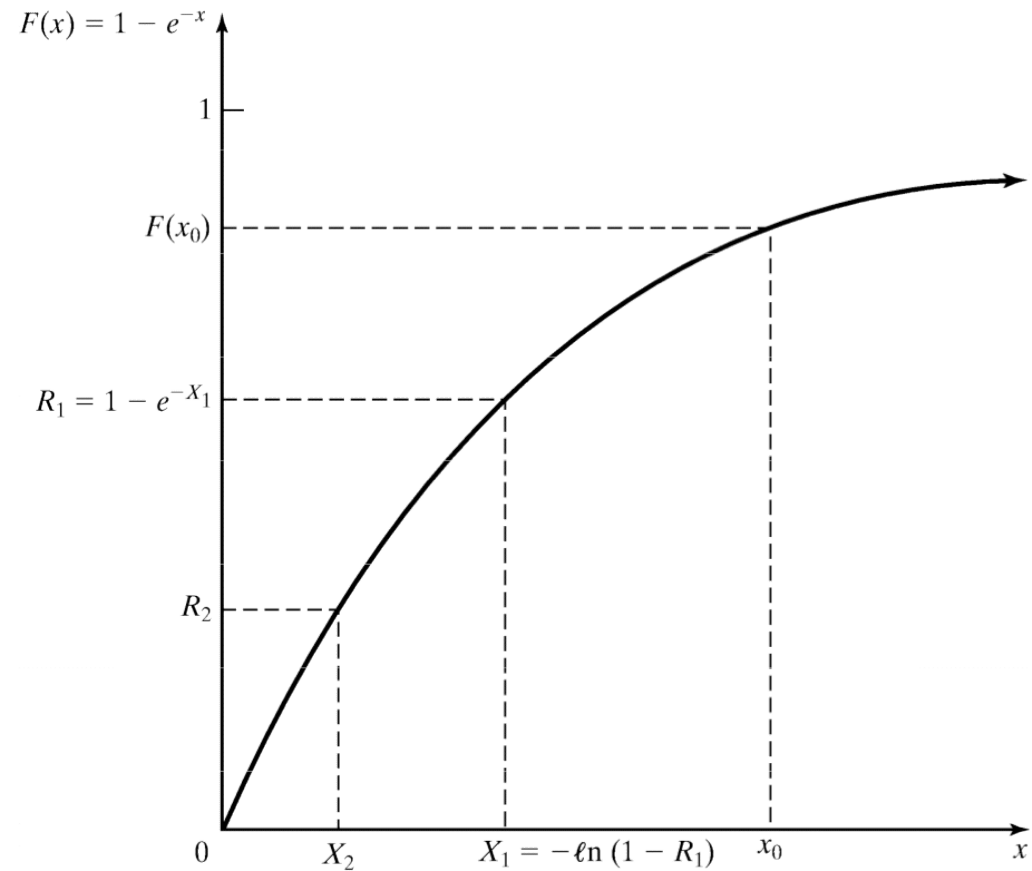
$$R = 1 - e^{-\lambda X}$$

$$1 - R = e^{-\lambda X}$$

$$\ln(1 - R) = -\lambda X$$

$$X = -\frac{\ln(1 - R)}{\lambda} = F^{-1}(R)$$

$$X = -\frac{\ln(R)}{\lambda} = F^{-1}(R)$$



Inverse-transform Technique

- Otras fáciles: Triangular, Weibull, Pareto
- También con $F(x)$ sacada de valores experimentales
 - Interpolar entre datos
- Si la variable es discreta es una simple tabla
- Difíciles: Gamma, Normal, Beta
- Aproximaciones numéricas a la CDF

Basarse en propiedades

Ejemplo: Distribución Normal

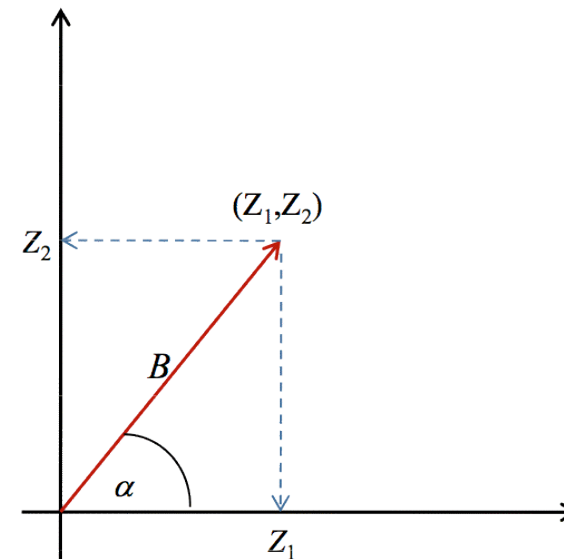
- Z_1 y Z_2 vv.aa. $N(0,1)$
- En coordenadas polares:
$$\begin{cases} Z_1 = B \cos(\alpha) \\ Z_2 = B \sin(\alpha) \end{cases}$$
- El radio B sigue una distribución exponencial
- El ángulo una distribución uniforme
- Son independientes
- Generar 2 valores de $N(0,1)$ con 2 valores de uniforme:

$$Z_1 = \sqrt{-2 \ln(R_1)} \cos(2\pi R_2)$$

$$Z_2 = \sqrt{-2 \ln(R_1)} \sin(2\pi R_2)$$

- Luego $N(\mu, \sigma)$:

$$Y = \mu + \sigma Z_i$$



Otros temas

Otros temas

- Verificación y validación del modelo
- Input Modeling
 - Data Collection
 - Identifying the distribution
 - Parameter estimation
 - Goodness-of-Fit tests
- Output Analysis
 - Estadísticos
 - Intervalos de confianza
 - Steady-state vs Transient
 - Batch Means