Diseño y Evaluación de Configuraciones

VI Simuladores

J.M. Drake

Notas:

Prototipos, emuladores y simuladores

- El análisis del comportamiento de un sistema se puede realizar básicamente con dos métodos:
  - Realizando medidas sobre el sistema que se analiza.
  - Formulando un modelo matemático que describa el comportamiento del sistema y resolviendo analíticamente el comportamiento del modelo
- La medida directa sobre el sistema no puede aplicarse si:
  - El sistema aún no existe (está en fase de diseño).
  - No se puede generar el workload que se quiere analizar
  - La naturaleza del sistema no permite hacer pruebas.
- Prototipo es un sistema de la misma naturaleza que el que se estudia, pero en el que se ha reducido la escala o la funcionalidad.
- Emulador es un sistema construido con una tecnología diferentes que se tiene el mismo comportamiento en base a que las ecuaciones que rigen el comportamiento son las mismas.
- Simulador es un sistema construido con la misma o con diferentes tecnología en base a que ambos son implementaciones del mismo modelo de comportamiento.

Notas:

Criterios para seleccionar la técnica de evaluación

Criterio	Modelo	Simulador	Medida
Etapa de desarrollo	Cualquiera	Cualquiera	Después de prototipo
Tiempo de evaluación	Pequeño	Medio	Según el caso
Herramientas	Herramientas matemáticas de análisis	Herramientas específica de simulación	Instrumentación o entornos de evaluación
Exactitud	Baja	Alta	Según el caso
Comparación entre sistemas	Fácil	Moderada	Dificil
Costo	Bajo	Medio	Alto
Escalabilidad	Baja	Media	Alta

Notas:

Verificación y validación

- Cuando se estudia un sistema mediante un modelo, prototipo o simulador hay que comprobar que conduce a resultados correctos. Esto requiere dos procesos:
  - Verificación: Hace referencia a la correctitud de la implementación
  - Validación: hace referencia a los errores que se cometen como consecuencia de la representatividad de las suposiciones que se hacen para establecer su correspondencia con el sistema.
- Hasta que un sistema no se haya validado los resultados de la evaluación deben estar bajo sospecha:
  - No dar fe a los resultados de simulación hasta que hayan sido validados por modelos analíticos o medidas directas.
  - No dar fe a los resultados analíticos hasta que no hayan sido validados por simulación o medida directa.
  - No dar fe a las medida directas hasta que hayan sido validadas por simulación o modelo analíticos.
- El conocimiento del experto es una técnica importante de validación.
- Es útil utilizar sucesivamente varias técnicas para complementar sus aprovechar sus ventajas complementarias

Notas:

Tipos de simuladores

- Simulación de Monte Carlo:
  - Se utilizan en sistemas en los que no existe evolución temporal, y se utilizan variaciones aleatorias para que el sistema evolucione hacia el valor correcto.
  - Habitualmente corresponde a la aplicación de métodos aleatorios para la resolución de sistemas deterministas que por su complejidad no pueden ser tratados como tales.
- Simulación conducida por trazas: Se sustituyen la respuestas de parte del sistema por trazas obtenidas de otras simulaciones o del sistema real (nunca del propio simulador)
  - Ventajas: Credibilidad, facilidad de validación, workload representativos, facilidad de comparación, independencia de las opciones, semejanza con el sistema que se estudia
  - Inconvenientes: Complejidad y falta de completitud..
- Simuladores de eventos discretos: La simulación basada en eventos discretos se basa en sistemas con relojes virtuales, que avanzan sólo a los instantes en que ocurre algo (evento). Son los mas utilizados.

Notas:

Errores en simulación: Nivel inadecuado de detalle

- El nivel de detalle de un análisis por simulación sólo está limitado por el tiempo que se le dedique:
  - A mayor nivel de detalle:
    - Mayor tiempo y costo de desarrollo
    - Mayor dificultad de depurar los errores.
    - Mayor tiempo de realización del análisis de resultados.
- Se supone que un mayor nivel de detalle es mejor porque requiere realizar menos suposiciones que validar. Esto no es siempre cierto:
  - A mayor detalle se requieren establecer mas parámetros. Si no se dispone de los datos y hay que suponerlos, se pueden incrementar los errores.
- Es conveniente realizar una estrategia en espiral:
  - Comenzar por un modelo simple, obtener resultados y sensibilidades.
  - Incrementar el nivel de detalle en las áreas que mejoren los resultados.

Notas:

Errores en simulación: Herramienta inadecuada

- La elección de la herramienta de simulación tiene un impacto relevante sobre el tiempo de desarrollo del modelo y sobre la facilidad de validación:
  - Una herramienta mas específica:
    - Es mas directa y requiere menos tiempo de desarrollo.
    - Es mas fácil de verificar (por ejemplo mediante trazas)
    - Ofrece resultados mas específicos (Estadísticas apropiadas)
  - Una herramienta mas general
    - Están mas disponibles y son mas portables y eficientes.

Notas:

Errores en simulación: Modelos no validos

- El simulador no reproduce el comportamiento del sistema bajo estudio, sino el comportamiento del modelo que implementa.
  - Errores en el modelado del sistema conducen a errores de la simulación.
- Los análisis por simulación siempre requieren análisis de validación de los modelos y de los resultados.

Notas:

Errores en simulación: Condiciones iniciales inadecuados

- El análisis por simulación requiere iniciar el estudio cada vez que se realiza un análisis, y por tanto se obtienen dos tipos de respuesta:
  - Respuesta transitoria: Es función de las condiciones iniciales de ejecución. Son irrelevantes en el estudio.
  - Respuesta estacionaria: Es función del workload y no depende de las condiciones iniciales. Es la información que se analiza.
- Unas condiciones iniciales inadecuadas puede producir una respuesta transitoria de mucha duración, o incluso que no se anula con el tiempo.
- Estrategias que pueden utilizarse son:
  - Ejecuciones muy prolongadas que hacen irrelevantes el peso del transitorio,
  - Inicialización apropiada: El estado inicial se toma de ejecuciones previas
  - Eliminación de los datos iniciales
  - Promediar múltiples ejecuciones con condiciones iniciales independientes.

Notas:

Errores en simulación: Simulaciones cortas.

- El tiempo de simulación es clave para garantizar los niveles de confianza de los resultados estadísticos.
- Es crítica en sistemas con gran variabilidad en los tiempos de los fenómenos que se producen.
- Formas de evitarlos:
  - Basar la finalización en medidas que evalúen los niveles de confianza.
  - Realizar simulaciones que describan el comportamiento del sistema en diferentes escalas de tiempo.
    - Simulaciones de fenómenos rápidos, en los que los fenómenos lentos se consideran como offsets.
    - Simulaciones de fenómenos lentos en los que los fenómenos rápidos se consideran como ruido.

Notas:

Erores en simuladores: Generadores aleatorios pobres.

- Uno de los principales objetivos del análisis por simulación es el anásis de la dependencia de los resultados de los tipos de distribuciones de los datos y parámetros del sistema.
  - Estos dependen de la capacidad de reproducir fenómenos con esas distribuciones.
- La generación de números aleatorios están muy bien estudiados por los matemáticos. El problema es que todos son deterministas, y requieren introducir una semilla inicial para conseguir que las diferentes generaciones sean diferentes.

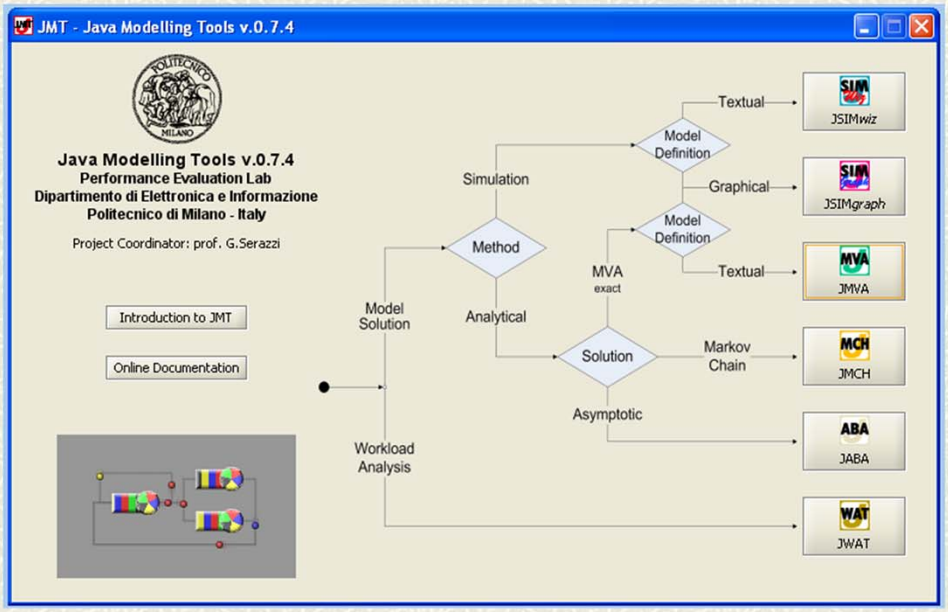
Notas:

Errores en simulación: No existencia de objetivos

- Analizar el comportamiento de un sistema no es sólo conseguir una traza neutra de la evolución del sistema. Sino conseguir una traza en la que se manifiesten los fenómenos que interesan analizar.
- Antes de plantear un análisis por simulación hay que formular claramente los objetivos del análisis.
- El proyecto de simulación debe diseñarse en base a los objetivos que se tienen establecidos.

Notas:

Java Modelling Tools (Politécnico de Milán)



Notas:

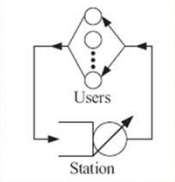
JSIM Graph: Interfaz gráfica del simulador

- A través de una interfaz gráfica se permite la introducción de los elementos de la red (layout) y los parámetros de los elementos.
- El simulador automatiza:
  - Detección y eliminación de los transitorios
  - Finalización automática cuando se alcanzan los niveles de confianza especcificados.
  - Capacidad de múltiples simulaciones barriendo un parámetro. (What if)

Notas:

Características (1):

- Tasas de entradas (Arrival rates): Burst (General), Constant, Erlang, Exponential, Gamma, Hyperexponential, Normal, Pareto, Poisson, Student-T, Uniform.
- Políticas de cola: First Come First Served, FCFS with priority, Last Come First Served, LCFS with priority
- Estrategias de encaminamiento:
  - Definida mediante probabilidad, Fastest service,Least utilization, Random, Round robin, Join the Shortest Queue, Shortest response time.
- Tiempos de servicios dependientes de la carga.

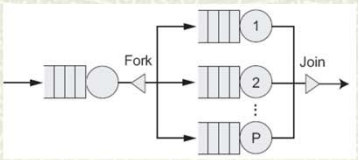


n	1	2	3	4	5	6	7	8
$D(n)$ [s]	2.00	2.50	3.33	4.25	5.20	6.17	7.14	8.13

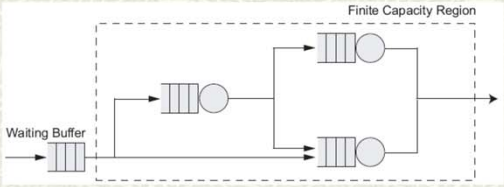
Notas:

Características (2)

- Modela paralelismo mediante estructuras Fork and Join.

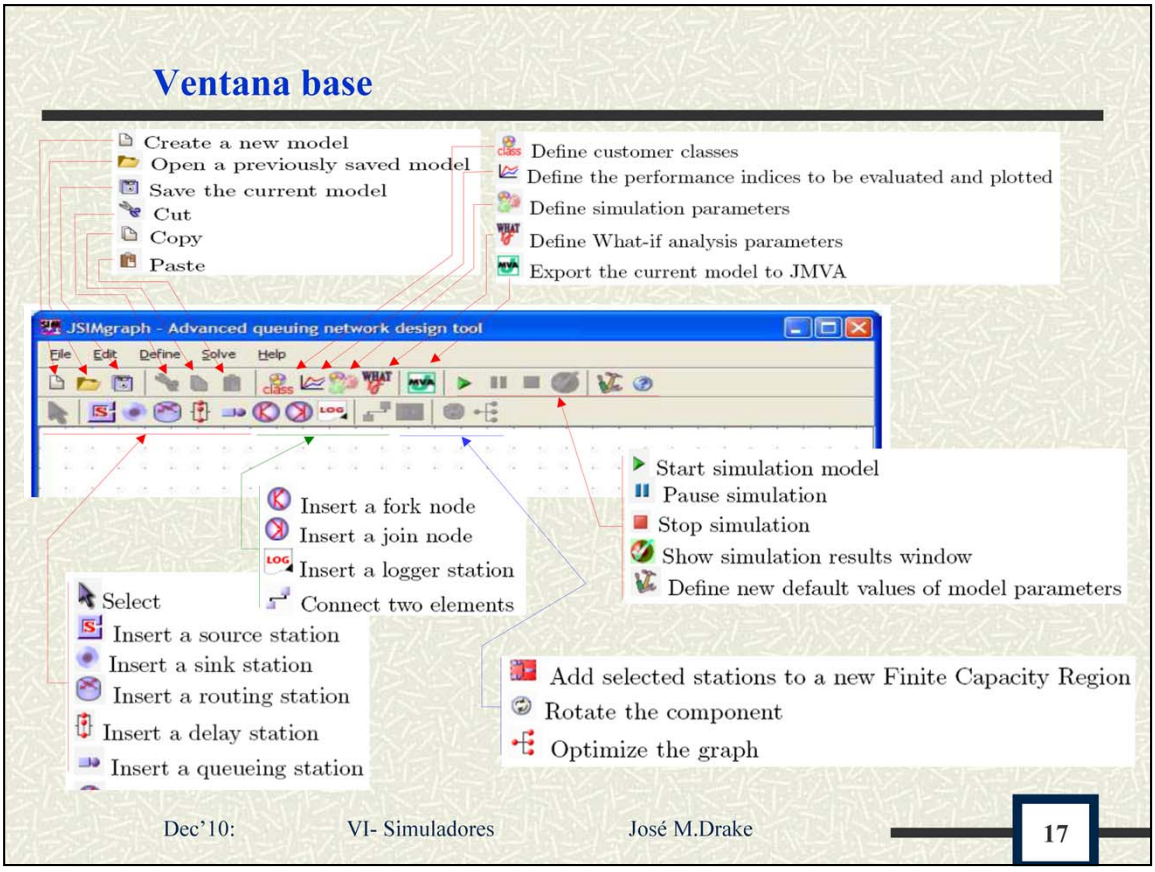


- Regiones de bloqueo(en el que el número de clientes esta limitado)



- Análisis What if.
- Registro de trazas
- Visualización gráfica de las métricas seleccionadas.

Notas:


Ventana base
Create a new model
Open a previously saved model
Save the current model
Cut
Copy
Paste
Define customer classes
Define the performance indices to be evaluated and plotted
Define simulation parameters
Define What-if analysis parameters
Export the current model to JMVA
JSIMgraph - Advanced queuing network design tool
File
Edit
Define
Solve
Help
Start simulation model
Pause simulation
Stop simulation
Show simulation results window
Define new default values of model parameters
Insert a fork node
Insert a join node
Insert a logger station
Connect two elements
Select
Insert a source station
Insert a sink station
Insert a routing station
Insert a delay station
Insert a queueing station
Add selected stations to a new Finite Capacity Region
Rotate the component
Optimize the graph
Dec'10:
VI- Simuladores
José M.Drake
17

Notas:

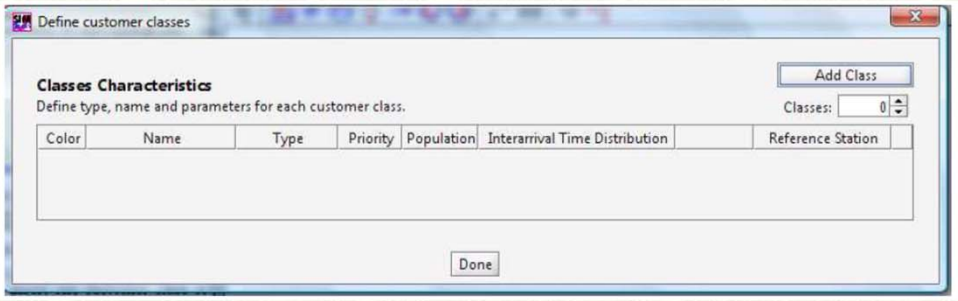
Proceso de simulación

1. Dibujar el circuito de colas del sistema.
2. Definir las clases de clientes y asociarlas a una cola de referencia.
3. Establecer los parámetros de cada elemento del modelo.
4. Seleccionar los índices de performance que se evalúan.
5. Si se necesitan se introducen las regiones con capacidad finita.
6. Establecer los parámetros de control de la simulación.
7. Habilitar el barrido What if si se utiliza.
8. Lanzar la simulación
9. Si se generan errores de diagnóstico, corregirlos.

Notas:

Definir las clases de clientes

- Definir el workload:
  - (Open) => λ Tasa de llegada.
  - (Closed) => N número de la población
- Definir los parámetros de los clientes:
  - (Open) => Prioridad, distribución estadística de intervalos entre llegadas, Estación de referencia, distribución de tiempos de servicios.
  - (Closed) => Prioridad, población, estación de referencia, distribución de los tiempos de servicios.



Notas:

Indices de performances: Métricas (1)

- Number of Customers (of a station): number of customers N at a station, both waiting and receiving service.
- Queue Time (of a station): average awaiting time in a station queue. It does not include the Service Time.
- Residence Time (of a station): total time spent at a station by a customer, both queueing and receiving service, considering all the visits at the station performed during its complete execution.
- Response Time (of a station): average time spent in a station by a customer for a single visit (Queue time + Service time).
- Response Time per Sink : average time spent in system by a customer before dropping at the selected sink.
- Utilization (of a station): percentage of time a station is used (i.e., busy) evaluated over all the simulation run. It ranges from 0 (0%), when the station is always idle, to a maximum of 1 (100%), when the station is constantly busy servicing customers for the entire simulation run. Queueing stations may have more than one server, their number is a parameter to be specified (default is 1). It is important to point out that the utilization U of a queueing station with a single server is given by U = S, and in a station with mservers the utilization of any individual server is given by U = S/m. In delay stations, for consistency with Little's law, the utilization is computed as the average number of customers in the station, and thus it may be greater than 1.

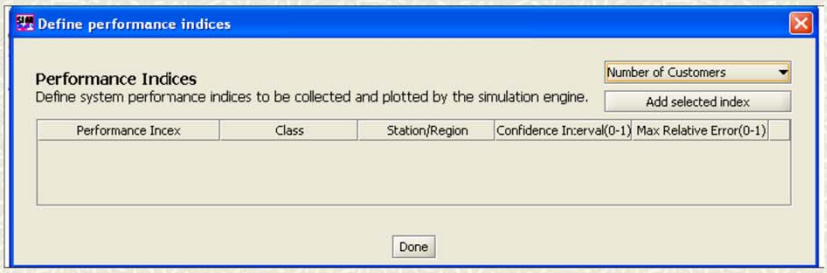
Notas:

Índices de performance: Métricas (2)

- Throughput (of a station): rate at which customers departs from a station, i.e., the number of requests completed in a time unit.
- Throughput per Sink : rate at which customers departs from the system with respect to the selected sink, i.e., the number of requests completed in a time unit that reach a given sink.
- System Throughput [X] (of the entire system): rate at which customers departs from the system.
- System Response Time [R](of the entire system): average time a customer spends in the system in order to receive service from the various stations it visits. It corresponds to the intuitive notion of response time, as the interval between the submission of a request and the reception of the response.
- System Number of Customers (of the entire system): average number of customers in the system. If the index is associated with a closed class, then it is equal to the number of customers in the class.
- System Power (of the entire system): The optimal operational point of a system is the point corresponding to the maximum System Throughput X with the minimum System Response time R, i.e., the value of the ratio X/R is maximized in this point. This ratio is known as System Power .

Notas:

Ventana para establecer las métricas



- Como salida de la simulación se puede establecer cualquier subconjunto de métricas:
  - Para cada métrica asociada a una clase de clientes y a una cola se le evalúa su intervalo de confianza y su máximo error relativo (ambos en el intervalo 0-1).
  - Por cada métrica que se quiera evaluar, se deben realizar los siguientes pasos:
    1. Seleccionar la métrica que se va aañadir.
    2. Seleccionar la clase (bien una individual, o todas agregadas)
    3. Seleccionar la cola
    4. Establecer el intervalo de confidencia y el máximo error admitido.

Notas:

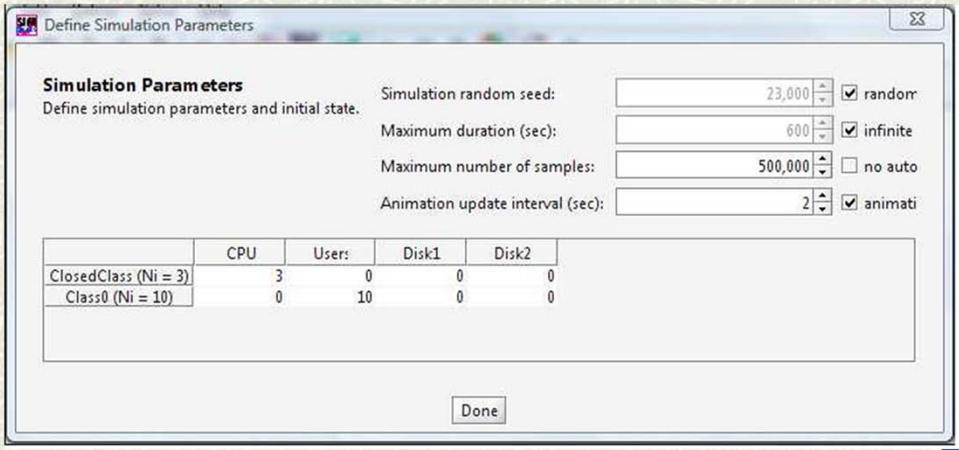
Intervalo de confianza y máximo error

- Intervalo de confianza : es el rango de valores alrededor del valor obtenido para el cual se han producido el (1-α)% de los valores. (1- α) es el nivel de confianza, y α el error aleatorio i nivel de significación.
- El error relativo es el el valor medio de las desviaciones con respecto al valor medio, normalizado por el valor medio.

Notas:

Parámetros de simulación

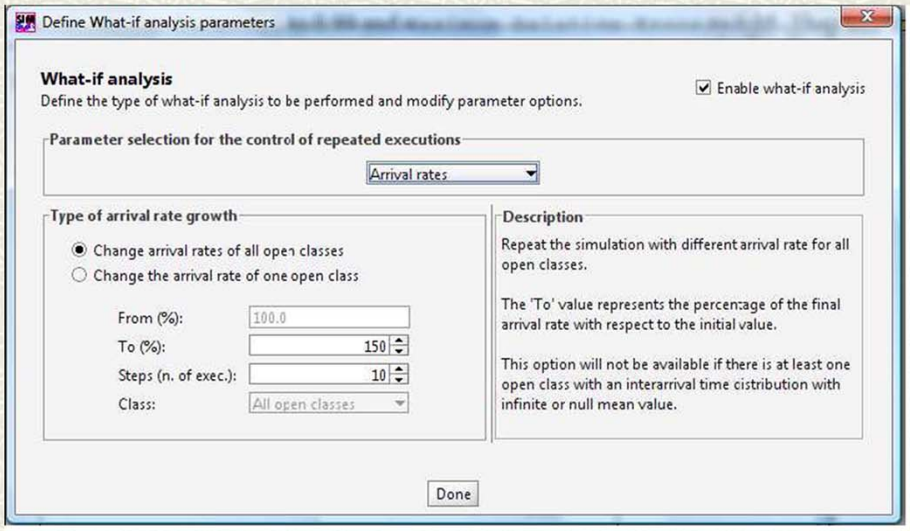
- Máxima duración: Tiempo máximo que la simulación debe durar. Valor por defecto infinito)
- Máximo número de muestras: Es el número de muestras para cada indice antes de que se finalice la simulación.
- Intervalo de representación (s): Es el intervalo de tiempo con el que se representan los resultados en los gráficos.
- Estado inicial: Distribución de los clientes en el instante 0.



Notas:

Análisis *What if*

- Consiste en un conjunto de simulaciones en los que un parámetro es modificado dentro de un rango y con un paso establecidos



Notas:

Regiones de capacidad limitada (FCR)

- Es un conjunto de colas en el que está controlado el número total de cliente en ellas.
- Hay dos tipo de FCR:
  - Shared: Se limita el número de clientes que pueden estar en el conjunto de las colas de la región, sin considerar el tipo de cliente de que se trata.
  - Dedicated: Se limita el número de clientes que pueden estar en el conjunto de las colas de la región de una determinada clase de clientes.
- Drop: Criterio de rechazo.
  - Drop=true => El cliente que trata de acceder a una FCR y la encuentra completa, es eliminado.
  - Drop=false =>El cliente que trata de acceder a una FCR y la encuentra completa, es mantenido en una cola asociada al FCR hasta que pueda ser admitido.

Notas:

Source Station

- Son bloques para introducir los clientes en el caso de sistemas abiertos:
  - La tasa de generación y el tipo de distribución se define sobre la clase.
  - A que colas se envían se define por la conectividad
  - Se puede definir la estrategia de encaminamiento:
    - Random: Los clientes generados se distribuyen aleatoriamente y con igual prioridad entre todos las colas de salida.
    - Round robin: Son circularmente enrutados entre todas las colas de salidas.
    - Probabilities: Se define la probabilidad de cada clase de salida.
    - Join the shortest queue (JSQ): El cliente generado se envía hacia el cliente que tiene menos elementos en espera en el tiempo en que se distribuye.
    - Shorstest response time: El cliente se envía hacia la cola donde el tiempo medio de respuesta es mas bajo en el tiempo en que se distribuye.
    - Least utilization: Se envía a la estación que tiene una utilización media mas baja en el tiempo en que se distribuye.
    - Fastest service: Se envía a la estación con tiemo medio de servicio mas bajo en el instante en que se distribuye.

Notas:

Sink y Delay

- En los sistemas abiertos define los puntos por los que los clientes salen del sistema.
  - Los Sink no tienen parámetros, solo conexiónes de entrada.
- Los delay son elementos con un número ilimitado de servidores (no tienen colas).
  - Son caracterizados por su tiempo de servicio, que fija la duración del retraso que los clientes sufren en él.
  - Su utilización es igual al numero de clientes en el centro: $Q_i=R_iX_i=S_iX_i=U_i$
  - Service section:
    - Load independent: Distribución de los tiempos de servicio
    - Load dpendent: Se define el tiempo de servicio por rangos.
    - Por cada rango se especifica: Distribución tiempo medio y coeficiente de variación (hiperexponencial >1, exponencial=1 e hipoexonencial <1).

Notas:

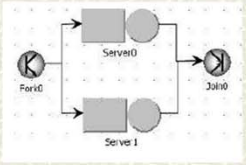
Queueing station: Cola

- Una cola tiene dos componentes la cola y el servicio:
  - Queue section:
    - Capacidad=> infinita o finita con una longitud finita definida.
    - Política de cola:
      - FCFS: Política FIFO estricta
      - FCFS(Priority): Se sirven por prioridades, y FIFO entre los de igual prioridad.
      - LCFS: Política LIFO estricta.
      - LCFS(Priority): Se sirven por prioridades, y LIFO entre los de igual prioridad.
    - Drop: Política de rechazo para las colas con longitud finita.
  - Service Section: Igual que en Delay (Load independent, load dependent)
  - Routing section: Igual que en Source (Random, Round robin, Probabilities, Join the shortest queue, Shortest response time, Least utilization, Fastest Service)

Notas:

Fork y Join Stations

- Permiten formular el paralelismo de colas.



- Un Fork station está caracterizado por dos parámetros:
  - Forking degree: Numero de clientes que son generados por cada salida y por cada clientes que llega.
  - Calpacity: Limita el numero de clientes que pueden estar operando en un grupo Fork-Join. En este caso el fork tiene una cola de espera. Con la correspondiente queue section (igual que en colas).
- Un Join station hace esperar los cliente que llegan procedentes de un mismo fork a que sean tantos como se generaron en él por cada cliente de entrada. Los que no proceden del fork se enrutan independientemente.
  - Tiene una sección de encaminamiento

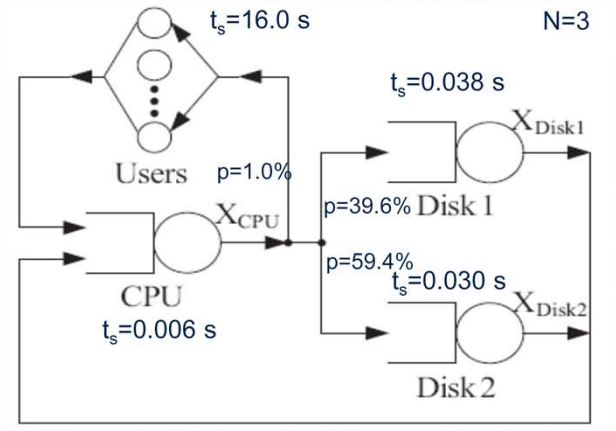
Notas:

Routing station Logger Station

- Es una estación con tiempo de ejecución nulo, que permite un enrutado con estrategia definida en él. Se introduce para proporcionar estrategias de encaminamiento mas complejos.
  - Se pueden combinar varias para mezclar estrategias.
- Una Logger Station permite capturar el flujo de clientes que pasa por él y registrarlo en un fichero.
  - Logging option: Permite definir donde y que información se registra.
  - Logfile option: Permite definirla gestión de los ficheros en los que se registran los eventos.

Notas:

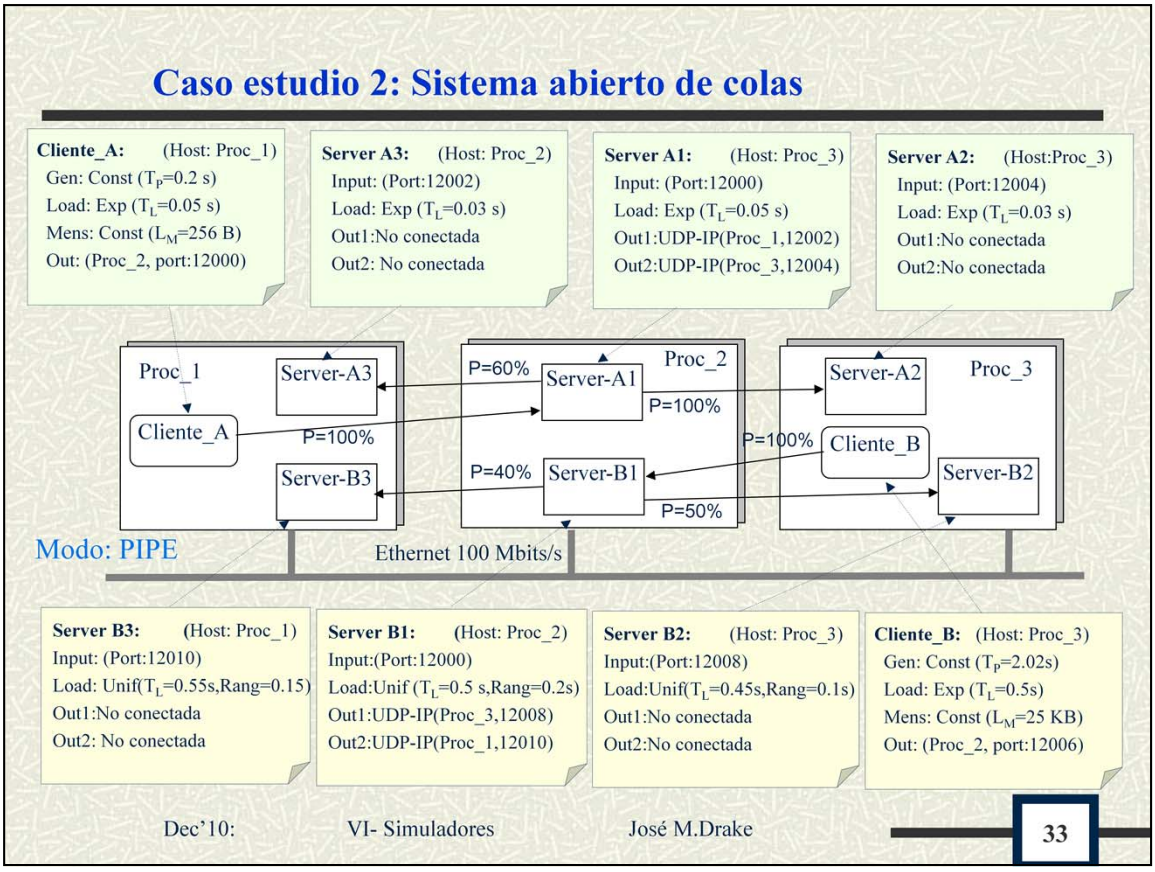
Caso estudio 1: Sistema cerrado



Analizar el sistema para N=3

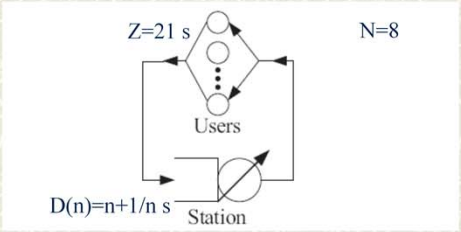
Analizar el throughput del sistema cuando N varía de 1 a 10

Notas:


Caso estudio 2: Sistema abierto de colas
Cliente_A: (Host: Proc_1)
Gen: Const (T_P=0.2 s)
Load: Exp (T_L=0.05 s)
Mens: Const (L_M=256 B)
Out: (Proc_2, port:12000)
Server A3: (Host: Proc_2)
Input: (Port:12002)
Load: Exp (T_L=0.03 s)
Out1:No conectada
Out2: No conectada
Server A1: (Host: Proc_3)
Input: (Port:12000)
Load: Exp (T_L=0.05 s)
Out1:UDP-IP(Proc_1,12002)
Out2:UDP-IP(Proc_3,12004)
Server A2: (Host:Proc_3)
Input: (Port:12004)
Load: Exp (T_L=0.03 s)
Out1:No conectada
Out2:No conectada
Proc_1
Server-A3
Cliente_A
P=100%
Server-B3
P=60%
Server-A1
Proc_2
P=100%
P=40%
Server-B1
P=50%
Server-A2
Proc_3
P=100%
Cliente_B
Server-B2
Modo: PIPE
Ethernet 100 Mbits/s
Server B3: (Host: Proc_1)
Input: (Port:12010)
Load: Unif(T_L=0.55s,Rang=0.15)
Out1:No conectada
Out2: No conectada
Server B1: (Host: Proc_2)
Input:(Port:12000)
Load:Unif (T_L=0.5 s,Rang=0.2s)
Out1:UDP-IP(Proc_3,12008)
Out2:UDP-IP(Proc_1,12010)
Server B2: (Host: Proc_3)
Input:(Port:12008)
Load:Unif(T_L=0.45s,Rang=0.1s)
Out1:No conectada
Out2:No conectada
Cliente_B: (Host: Proc_3)
Gen: Const (T_P=2.02s)
Load: Exp (T_L=0.5s)
Mens: Const (L_M=25 KB)
Out: (Proc_2, port:12006)
Dec'10:
VI- Simuladores
José M.Drake
33

Notas:

Caso de estudio 3: Clase con demanda función de la carga



n	1	2	3	4	5	6	7	8
$D(n)$ [s]	2.00	2.50	3.33	4.25	5.20	6.17	7.14	8.13

Notas: