Modelado y Simulación Numérica de Sistemas Informáticos. Un Enfoque Dinámico Continuo.

Jorge R. Vega

GIRED (Grupo de Investigación en Redes)
Universidad Tecnológica Nacional – Facultad Regional Santa Fe
Lavaise 610 (3000) Santa Fe – Argentina
E-mail: jvega@arcride.edu.ar

Resumen

Se presenta una metodología para el modelado y la simulación dinámica de un sistema informático (SI) arbitrario. El SI se representa gráficamente mediante la interconexión de tres clases de módulos, destinados a generar, atender, o encaminar trabajos. Cada módulo se comporta como un centro *dinámico* o *estático*. Los centros dinámicos se modelan matemáticamente por medio de ecuaciones diferenciales ordinarias; y los centros estáticos a través de ecuaciones algebraicas. Los modelos matemáticos propuestos predicen la evolución temporal del número de trabajos en cada dispositivo del SI, y proveen resultados de estado estacionario coincidentes con los obtenidos a través de algoritmos clásicos. La estructura gráfico/modular de cada centro permite construir módulos de mayor jerarquía para modelar componentes o dispositivos de un SI más complejo. La metodología es útil para evaluar la "performance" de un SI, y es fácil de implementar en un curso universitario de grado.

Palabras Clave: Modelado y simulación - Evaluación de performance - Informática educativa

1. Introducción

Debido al gran avance de los sistemas de comunicaciones y a su uso masivo e intensivo, la complejidad de los sistemas informáticos (SI) ha crecido notoriamente a lo largo del tiempo, tanto desde el punto de vista del *hardware* como del *software* utilizado. Actualmente, en lapsos de tiempo relativamente cortos se requiere actualizar y modificar estos sistemas, en busca de satisfacer la demanda pretenciosa de los usuarios: lograr tiempos de respuesta cada vez más pequeños para manipular transacciones con cargas de trabajo cada vez más elevadas. Indudablemente, para lograr estas mejoras no basta sólo con disponer de un SI moderno, con componentes cada vez más veloces; sino que también es importante administrar adecuadamente los recursos disponibles, evaluar la prestación que dichos recursos otorgan, sus debilidades o cuellos de botella, sus puntos de posible mejora, etc. En tal sentido, la evaluación de medidas de "performance" de un SI es una disciplina de gran importancia; y la simulación numérica de un modelo puede prestar una gran ayuda a la hora de evaluar potenciales modificaciones o rediseños del SI existente.

Desafortunadamente, el desarrollo de un modelo representativo de un SI no es una tarea sencilla. En efecto, las variables involucradas en el funcionamiento de un SI pueden ser continuas o discretas, evolucionan en el tiempo, y tienen normalmente atributos aleatorios. Desde un punto de vista estrictamente matemático, deberían utilizarse modelos derivados de la teoría general de los procesos estocásticos; y la obtención de resultados por esta vía requeriría entonces del planteo y de la resolución de sistemas de ecuaciones diferenciales (o en diferencias) estocásticas. Sin embargo, la complejidad involucrada en este tipo de modelos conduce muchas veces a recurrir a importantes simplificaciones. Por ejemplo, el modelo matemático de un proceso de "nacimiento/muerte", cuyos estados poseen atributos probabilísticos, puede representarse por las ecuaciones dinámicas de Kolmogorov; aunque por razones fundamentalmente de índole práctica, dichas ecuaciones suelen ser resueltas sólo en el estado estacionario. Los procesos de Markov, la teoría de colas, y especialmente los modelos de redes de cola, son posiblemente las herramientas teóricas más utilizadas para el modelado matemático de SI [1-4], incluso en aplicaciones más recientes destinadas a la evaluación de la "performance" de redes de computadoras [5,6].

En las últimas décadas han surgido otros enfoques alternativos, líneas de investigación y herramientas para el modelado de SI, tales como las redes de Petri y la teoría de los sistemas a eventos discretos. Si bien las redes de Petri permiten construir modelos en una forma relativamente sencilla, la complejidad descriptiva del modelo crece rápidamente, aun para sistemas no tan elaborados [7]. Por otra parte, los complejos conceptos matemáticos que sustentan la teoría de los sistemas a eventos discretos muchas veces desaniman a trabajar con ellos. En cierto sentido, existen similitudes entre los dos enfoques, y algunas herramientas algebraicas permiten describir ambos tipos de modelos [8]. Independientemente de la metodología de modelado utilizada, las características *dinámicas* intrínsecas de los SI son innegables; y estrictamente, el modelo de un SI debería permitir la predicción de la evolución temporal de las variables involucradas. La dinámica de los sistemas a eventos discretos ha sido discutida en la literatura, y comparada con la dinámica de los sistemas continuos [9,10], pero los modelos matemáticos desarrollados para SI no siempre la contemplan.

Muchas veces, el análisis de la "performance" de un SI se efectúa a partir de un modelo operacional, matemáticamente representado por expresiones *algebraicas* recursivas derivadas de la teoría de colas. Bajo este enfoque, el modelo del SI es *estático*, y según las características de procesamiento y el tipo de carga, suele clasificarse en: 1) cerrado, abierto o mixto; 2) de clase simple o múltiple; y 3) con generación de cargas de trabajos por colas "batch", por terminales, o por accesos transaccionales [2,3]. La resolución de las expresiones matemáticas del modelo y la

evaluación de las medidas de “performance” del SI se efectúan por medio de algoritmos algebraicos que resuelven el problema *estacionario*. Entre los algoritmos típicamente utilizados, podemos mencionar los denominados MVA (“*mean value analysis*”) exactos y aproximados; y normalmente se describe un algoritmo diferente para cada clase de SI a analizar. En el caso de los modelos de los SI más simples (“separables” [2]), estos algoritmos son relativamente sencillos de implementar (basta para ello una simple planilla de cálculo). Sin embargo, adolecen de al menos dos limitaciones importantes: a) su complejidad conceptual, en el sentido que es difícil interpretar el significado físico de cada uno de sus pasos; y b) sólo proveen resultados de estado estacionario.

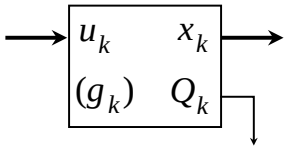
Un conocimiento exhaustivo sobre modelado matemático de SI escapa usualmente a la formación académica recibida por un estudiante de grado; y también –posiblemente– a la preparación media de un profesional de la ingeniería en el área de los sistemas informáticos. Sería de interés entonces disponer de alguna herramienta que permita con cierta facilidad modelar un SI y evaluar su “performance”, logrando predicciones razonablemente aproximadas, aun a costa de sacrificar el rigor científico que proveerían los modelos más precisos. En este trabajo, se propone una metodología para modelar y simular el comportamiento dinámico de un SI. La idea general perseguida es desarrollar una herramienta computacional que permita –con facilidad y en forma intuitiva–, analizar el funcionamiento del SI, practicar modificaciones en su configuración, determinar medidas de “performance”, identificar dispositivos conflictivos, etc. Además, resultará de interés proveer a esta herramienta de: a) un carácter gráfico/modular; y b) una capacidad de modelado jerárquico.

2. Consideraciones Preliminares

Entenderemos por SI a un conjunto de dispositivos interconectados –a los que genéricamente denominaremos *módulos*–, y capacitados para recibir, generar (o consumir), atender, encaminar, y/o transferir trabajos de cómputo. Cada uno de estos módulos intercambia información con el resto del sistema (es decir, con otros módulos) a través de las variables instantáneas “flujo de ingreso de trabajos” [$u(t)$], y “flujo de egreso de trabajos” [$x(t)$]. El funcionamiento de cada módulo se describirá mediante un modelo matemático determinístico de tiempo continuo; básicamente, una expresión matemática que vinculará ambos flujos. El modelo matemático del SI será la concatenación de los modelos matemáticos individuales de cada módulo, respetando la topología de su interconexión.

Denominaremos *centros* a los módulos más simples (de menor jerarquía) que componen un SI. La combinación de dos o más centros dará lugar a la formación de un módulo de mayor jerarquía. En este trabajo, consideraremos los siguientes centros: a) dos centros de *generación de trabajos* (por terminales y por colas “batch”); b) un centro de *atención de trabajos*, que genéricamente denominaremos centro de cola; y c) dos centros de *encaminamiento de trabajos* (nodos de derivación y de confluencia).

La clasificación de los centros presentada en el párrafo anterior es meramente funcional, pero a los efectos del modelado matemático es conveniente clasificarlos según su comportamiento en el SI como *centros dinámicos* y *centros estáticos*. En un SI, un *centro dinámico* (CD) es un conjunto de dispositivos que procesa y/o demora los trabajos (Q) que instantáneamente residen en él. En el caso más general, un CD puede además generar, consumir, recibir y/o entregar flujos de trabajos. En la Figura 1, se representa en forma esquemática un CD ‘k’. Admitiremos que todas las variables asociadas con el CD ‘k’ pueden evolucionar en forma continua en el tiempo (t).



Variables del CD '*k*'	
Nombre	Descripción
u_k [trab./s]	Flujo de arribo de trabajos
g_k [trab./s]	Flujo de generación (o consumo) de trabajos
Q_k [trab.]	Número de trabajos en el CD
x_k [trab./s]	Flujo de procesamiento de trabajos

Figura 1: Esquema modular de un CD. Descripción de las variables asociadas.

En un intervalo de tiempo suficientemente pequeño, un balance global de los trabajos en el CD puede expresarse cualitativamente de la siguiente manera:

$$\left[\begin{array}{c}\text{Variación temporal}\\ \text{de trabajos en el CD}\end{array}\right] = \left[\begin{array}{c}\text{Flujo de arribo}\\ \text{de trabajos}\end{array}\right] - \left[\begin{array}{c}\text{Flujo de salida}\\ \text{de trabajos}\end{array}\right] + \left[\begin{array}{c}\text{Flujo de generación}\\ \text{(consumo) de trab.}\end{array}\right]$$

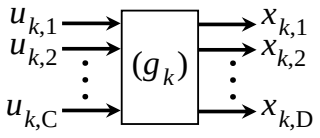
Este balance puede representarse matemáticamente a través de la siguiente ecuación diferencial ordinaria:

$$\frac{dQ_k(t)}{dt} = u_k(t) - x_k(t) + g_k(t) \;;\quad Q_k(t_0) = Q_{k,0} \tag{1}$$

donde $Q_{k,0}$ representa al número de trabajos en el CD, en un dado tiempo t_0. (Nótese que g_k es positivo cuando se generan trabajos, y negativo si se consumen o pierden.)

Para resolver la ec. (1), hay que proponer un modelo matemático que permita calcular la velocidad de procesamiento x_k; y esta velocidad dependerá del tipo de CD en cuestión. En este trabajo, consideraremos dos clases de CD: los centros de cola, y los centros de generación de trabajos por terminales. En el punto 3, se presentan las características operativas y los modelos matemáticos propuestos para ambos centros.

En un SI, un *centro estático* (CE) es un conjunto de dispositivos que intercambia información entre sus entradas y sus salidas, en un tiempo infinitamente pequeño (en la práctica, en un tiempo mucho menor que la menor constante de tiempo del SI). Desde este punto de vista, un CE puede ser considerado como un CD ficticio, con una dinámica *infinitamente rápida*. En el caso más general, un CE puede recibir C flujos de entrada, entregar D flujos de salida, y generar (o consumir) trabajos. En la Figura 2, se representa en forma esquemática un CE '*k*'. Admitiremos que todas las variables asociadas con el CE '*k*' pueden evolucionar en forma continua en el tiempo (*t*).



Variables del CE '*k*'	
Nombre	Descripción
$u_{k,i}$ [trab./s]	Flujos de arribo de trabajos
g_k [trab./s]	Flujo de generación (o consumo) de trabajos
$x_{k,i}$ [trab./s]	Flujos de partida de trabajos

Figura 2: Esquema modular de un CE. Descripción de las variables asociadas.

Los CE se encuentran permanentemente en estado estacionario, y entonces el balance de la ec (1) se reduce a:

$$0 = \sum_{i=1}^{C} u_{k,i}(t) - \sum_{i=1}^{D} x_{k,i}(t) + g_k(t) \qquad (2)$$

En este trabajo consideraremos tres clases de CE: los centros de generación de trabajos "batch", los nodos de confluencia de trabajos, y los nodos de derivación de trabajos. En el punto 4, se presentan las características operativas y los modelos matemáticos propuestos para estos centros.

3. Modelado Matemático de los Centros Dinámicos

3.1. Centro de Cola

Un *centro de cola* (CC) es un CD en el que mientras algunos trabajos están siendo atendidos o procesados, los restantes esperan en cola su turno. Ejemplos de CC son: una CPU, un sistema de discos rígidos, un canal de transmisión de datos, etc. En una primera instancia, se admitirá que en un CC: i) todos los trabajos son similares, y requerirán entonces el mismo tiempo medio de servicio o atención (S_k = cte.); y ii) no se produce generación ni consumo de trabajos (es decir, $g_k = 0$). En la Figura 3, se representa un esquema operacional de un CC, típicamente utilizado para diferenciar los trabajos en servicio de los ubicados en cola de espera. Sin embargo, debe notarse que con relación al modelo matemático sólo interesará computar el número total de trabajos en el CC (Q_k).

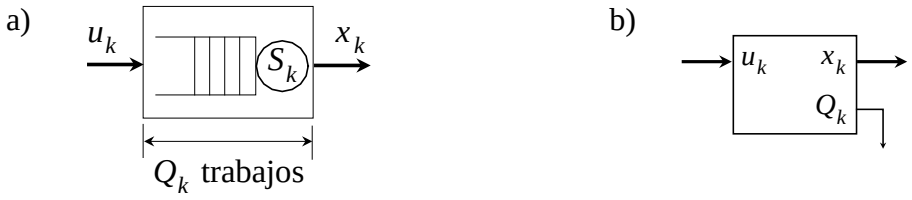


Figura 3: Representación de un CC: a) esquema operacional; b) estructura modular.

Veamos ahora cómo evaluar la velocidad de procesamiento de trabajos, x_k, en un CC. Intuitivamente, es de esperar que x_k dependa al menos de los siguientes factores: i) la tasa de servicio, S_k; ii) la cantidad de trabajos en el CC, Q_k; y iii) la cantidad total de trabajos residentes en el SI, N. Por un lado, es bastante evidente que x_k será inversamente proporcional a S_k. Por otra parte, es de esperar que x_k sea proporcional a Q_k, dado que al aumentar Q_k mayor utilización tendrá el CC (hay más disponibilidad de trabajos para ser atendidos y resueltos); pero los incrementos marginales de x_k con Q_k serán cada vez menores, porque el CC tenderá gradualmente a saturarse. Por último, la dependencia de x_k con N es más indirecta, porque no se puede ahora considerar al CC en forma aislada, dado que los N trabajos se reparten entre todos los recursos del SI. En general, al aumentar N, se espera una mayor utilización de cada uno de los recursos, y un incremento de x_k como consecuencia de un aumento en Q_k.

No existe evidentemente una expresión analítica única que permita calcular fielmente la velocidad de procesamiento x_k en un CC. A efectos del modelo, en este trabajo se propone que la expresión a utilizar satisfaga los requerimientos comentados en el párrafo anterior, y además que permita reproducir los valores estacionarios provistos por la técnica MVA aproximada [2]. Concretamente, se propone calcular $x_k(t)$ a través de la siguiente expresión:

$$x_k(t) = \frac{Q_k(t)}{R_k(S_k, Q_k, N)} = \frac{Q_k(t)}{S_k\left[1 + Q_k(t) - \frac{Q_k(t)}{N}\right]} \tag{3}$$

donde R_k es un tiempo de residencia característico por cada visita de un trabajo al CC. Nótese que si $N = 1$, entonces $R_k = S_k$; y si $N \to \infty$, resulta $R_k = S_k + S_k\, Q_k$; es decir, la ec. (3) es exacta para $N = 1$, y asintóticamente correcta para valores elevados de N.

Reemplazando la ecn. (3) en la (1), y recordando que $g_k = 0$, resulta:

$$\frac{dQ_k}{dt} = u_k(t) - \frac{Q_k(t)}{S_k\left[1 + Q_k(t) - \frac{Q_k(t)}{N}\right]} \; ; \; Q_k(t_0) = Q_{k,0} \tag{4}$$

Esta expresión puede extenderse al caso de un CC que atienda trabajos de distinto requerimiento medio (clases múltiples). Un CC '*k*' que atienda *CL* clases de trabajos, tendrá un tiempo característico para la clase '*c*', que se calculará como:

$$R_{k,c}(t) = S_{k,c}\left[1 + \sum_{j=1}^{CL} Q_{k,j}(t) - \frac{Q_{k,c}(t)}{N_c}\right] \quad ; \quad (c = 1, ..., CL) \tag{5}$$

donde $S_{k,c}$ y $Q_{k,c}$ son el tiempo de servicio y el número de trabajos de la clase *c* en el CC '*k*', respectivamente. Entonces, para el CC '*k*' deberá plantearse una EDO por cada clase:

$$\frac{dQ_{k,c}}{dt} = u_{k,c}(t) - \frac{Q_{k,c}(t)}{S_{k,c}\left[1 + \sum_{j=1}^{CL} Q_{k,j}(t) - \frac{Q_{k,c}(t)}{N_c}\right]} \; ; \; Q_{k,c}(0) = 0 \; ; \quad (c = 1, ..., CL) \tag{6}$$

donde $u_{k,c}$ es el flujo de trabajos de clase *c* que arriba al CC '*k*'. Nótese que para una única clase de trabajos, la ec. (6) se reduce a la ec. (4).

3.2. Centro de Generación por Terminales

En un *centro de generación por terminales* (CGT), cada usuario (ubicado en una terminal) origina un trabajo nuevo para el SI, cuando recibe la información de que su trabajo anterior ha concluido. Nótese entonces que la velocidad neta de generación de trabajos (generación menos consumo) es nula (g_k=0). Supongamos que hay N_T terminales activas, y que Z es el tiempo medio de pensado o reflexión utilizado por cada usuario para decidir la generación de un nuevo trabajo. Entonces, el CGT puede ser considerado como un CC con $Q_k = Q_T$, y $R_k = Z$. En tal caso, la velocidad de "procesamiento" en las terminales se modela como $x_T = Q_T / Z$; y a partir de la ecn. (1), resulta:

$$\frac{dQ_T}{dt} = u_T(t) - \frac{Q_T(t)}{Z} \; ; \quad Q_T(0) = N_T \tag{7}$$

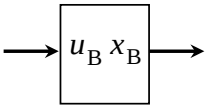
donde u_T es el flujo de trabajos que arriba al CGT; y $Q_T(0)=N_T$ indica que, a t=0, todos los trabajos residen en las terminales antes de ser remitidos para su procesamiento. Debe notarse que en un CGT $Q_T(t) \le N_T$. En la Fig. 4, se presentan el modelo operacional y el esquema modular de un CGT.

Figura 4: Representación de un CGT: a) esquema operacional; b) estructura modular

4. Modelado Matemático de los Centros Estáticos

4.1. Centro de Generación "batch"

Un *centro de generación "batch"* (CGB) puede interpretarse como una cola de trabajos que están permanentemente disponibles por el SI. Dicha cola no es un 'dispositivo' en el que puedan acumularse trabajos, sino que provee en forma instantánea: a) un trabajo nuevo por cada trabajo resuelto en el SI; y b) N_B nuevos trabajos, al aumentarse en N_B el grado de multiprogramación del SI. En la Fig. 5, se presenta el esquema modular de un CGB para una única clase de trabajos, y su correspondiente modelo matemático derivado a partir de la ec. (2), con C=D=1.



$$x_B(t) = u_B(t) + \delta_0 N_B \qquad (8)$$

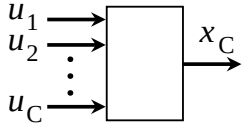
Figura 5: Estructura modular y modelo matemático de un CGB.

En la ec. (8), $u_B(t)$ y $x_B(t)$ son, respectivamente, las velocidades de arribo y de partida de los trabajos de clase simple, asociadas al CGB; $\delta_0 N_B$ representa la velocidad de generación de N_B nuevos trabajos de la misma clase al SI, en el tiempo t_0; y δ_0 es una "delta de Dirac" en el tiempo t_0. Nótese que, $x_B(t) = u_B(t)$, $\forall\ t \neq t_0$.

4.2. Nodos de Confluencia y de Derivación de Trabajos

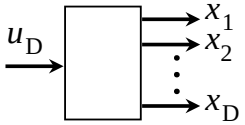
Un *nodo de confluencia de trabajos* (NCT) es un punto de un SI al que concurren simultáneamente C (>2) flujos de trabajos, $u_i(t)$, i=1,...C; y del que parte un único flujo, $x_C(t)$. Un *nodo de derivación de trabajos* (NDT) es un punto del SI al que arriba un único flujo de trabajos, $u_D(t)$, y del que parten simultáneamente D (>2) flujos de trabajos, $x_i(t)$, i=1,...D. En cualquiera de estos nodos admitiremos que: 1) no se generan ni se destruyen o pierden trabajos (g=0); y 2) el encaminamiento de esa información se efectúa en forma instantánea, sin acumularse trabajos (CE). En un NDT se requiere conocer en que proporción se encamina el flujo de trabajos hacia cada una de las D salidas. Llamaremos $p_i(t)$, i=1,...D, a la probabilidad instantánea de que un trabajo que arriba al NDT se encamine hacia la salida 'i' (usualmente, dichas probabilidades se asumirán constantes). En la Fig. 6, se muestran los esquemas modulares de ambos nodos, y sus correspondientes modelos matemáticos derivados a partir de la ec. (2). Nótese que: 1) D=1, en un NCT; y 2) C=1, en un NDT.

NCT:



$$x_C(t) = \sum_{i=1}^{C} u_i(t) \qquad (9)$$

NDT:



$$x_i(t) = p_i(t)\, u_D(t)\ ; \qquad i = 1, ..., D \qquad (10.a)$$

$$\sum_{i=1}^{D} p_i(t) = 1 \qquad (10.b)$$

Figura 6: Estructura modular y modelo matemático de los NCT y NDT.

5. Ejemplos de Aplicación

Se construyó una "biblioteca" de centros básicos para clase simple de trabajos, con los siguientes componentes: 1) un CC [ec. (4)]; 2) un CGT [ec. (7)]; 3) un CGB [ec. (8)]; 4) un NCT [ec. (9)]; y 5) un NDT [ec. (10)]. Estos centros permiten modelar módulos más complejos y SI. A efectos de mostrar la metodología propuesta, se presentan a continuación algunos ejemplos tomados de la referencia [2]. Los modelos matemáticos propuestos se implementaron en la interface gráfica Simulink, de Matlab [11].

Ejemplo 1. Un servidor central (1 CPU / 2 discos) es accedido desde 3 terminales, con trabajos de requerimientos similares (clase simple). En la Fig. 7 se muestra la configuración del SI, implementada con los centros básicos disponibles. En el módulo jerárquico desarrollado para el servidor central, se seleccionaron arbitrariamente algunas variables internas para disponerlas como variables de salida; pero sólo la velocidad de procesamiento global (X) podrá utilizarse para la interconexión con otros módulos. A partir de las salidas del modelo, podrían calcularse otras medidas de "performance" (por ejemplo, las velocidades de procesamiento en la CPU y en los discos, sus tiempos de respuesta, etc.). En la Tabla 1, se presentan los datos del ejemplo, y se comparan los resultados estacionarios de algunas variables obtenidos aplicando la metodología propuesta (en negrita), con los resultados provistos por los algoritmos MVA exacto y aproximado. Las probabilidades del NDT se calcularon en base a las visitas de los trabajos a los distintos recursos, según: $p_1 = 1/V_{CPU} = 1/121$; $p_2 = V_{d,1}/V_{CPU} = 70/121$; $p_3 = V_{d,2}/V_{CPU} = 50/121$. El tiempo total de respuesta R se calculó por aplicación directa de la ley de Little ($R=N_T/X–Z$). Nótese que los estados estacionarios provistos por el modelo coinciden con las predicciones del algoritmo MVA aproximado. En la Fig. 8, se muestran las evoluciones temporales de las colas de trabajo y de las velocidades de procesamiento en cada CC y en el CGT, correspondientes al proceso de arranque del sistema. Al inicio del proceso (t≈0), todos los trabajos están en las terminales (Q_T≈3); y luego se va cargando cada dispositivo hasta alcanzar el estado estacionario, después de unos 20 s.

Ejemplo 2. Se considera el mismo servidor central del *Ejemplo 1*, reemplazando la generación de trabajos con terminales por una carga externa (transaccional), con una velocidad de acceso de trabajos λ=0.3 trab./s. Siguiendo la aproximación para modelos abiertos [2], el tiempo de respuesta $R_k(\lambda)$ en un CC arbitrario es $R_k(\lambda) = S_k\ [1+ Q_k(\lambda)]$. Para emular este comportamiento mediante los CC ya definidos, se impone $N\rightarrow\infty$ en la ec. (4). El esquema de simulación es el mismo de la Fig. 7, con el CGT reemplazado por una entrada constante λ. Alternativamente, puede utilizarse el módulo jerárquico del servidor central (Fig. 7), con $u=\lambda$. Los resultados obtenidos no se muestran por

razones de brevedad; pero los valores estacionarios coinciden con la resolución del mismo SI por el algoritmo para modelos abiertos, detallado en [2].

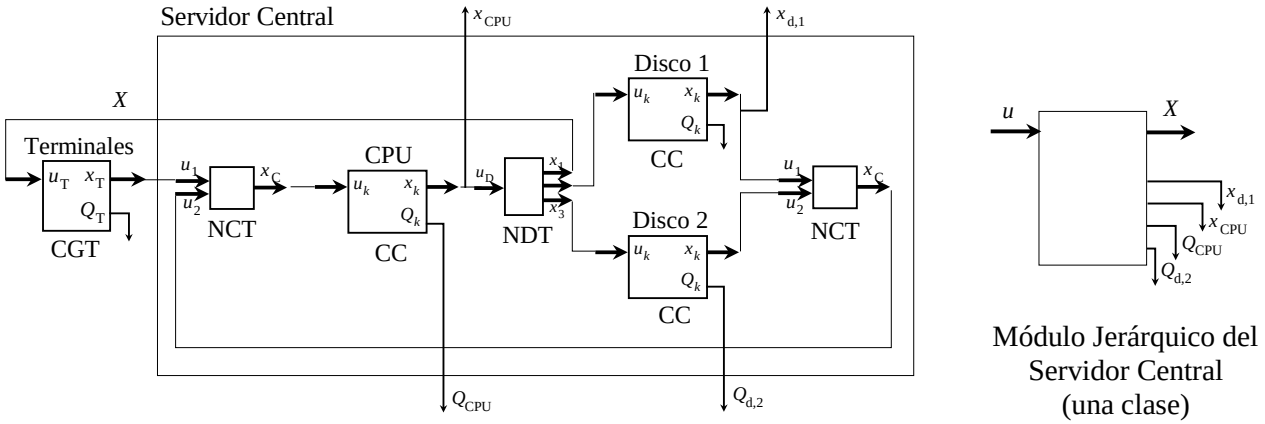


Figura 7. Esquema modular para la simulación numérica del Ejemplo 1, y módulo jerárquico equivalente del servidor central (salidas arbitrariamente seleccionadas).

TABLA 1: DATOS Y RESULTADOS DE ESTADO ESTACIONARIO DEL EJEMPLO 1

Datos	Resultados							
S_{CPU}=5 ms ; V_{CPU}=121		Q_T	Q_{CPU}	$Q_{d,1}$	$Q_{d,2}$	x_{CPU}	$X=u_T$	R
$S_{d,1}$=30 ms $V_{d,1}$=70 ;	Método Propuesto	**2.265**	**0.0972**	**0.4021**	**0.2359**	**18.26**	**0.1509**	**4.881**
$S_{d,2}$=27 ms $V_{d,2}$=50 ;	MVA Exacto [2]	2.273	0.0976	0.3947	0.2350	18.39	0.1520	4.737
Z=15 s ; N_T=3	MVA Aprox. [2]	2.265	0.0972	0.4021	0.2359	18.26	0.1509	4.881

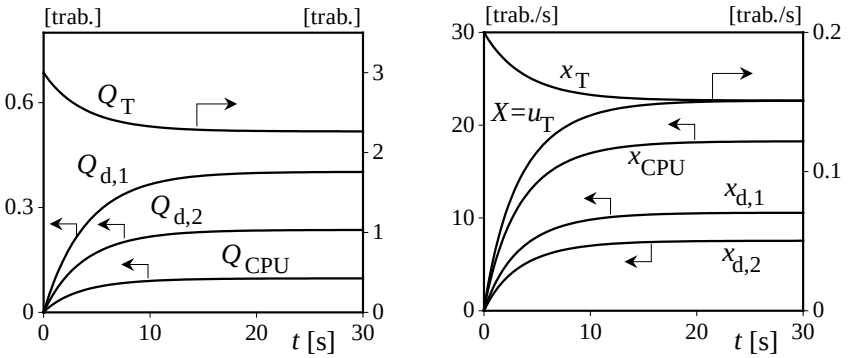


Figura 8. Evoluciones temporales de algunas variables del SI del Ejemplo 1.

Ejemplo 3. Un servidor (1 CPU / 1 disco) atiende dos clases de trabajos (A y B), que arriban al SI como carga transaccional con λ_A=3/19 trab./s, y λ_B=2/19 trab/s. Para cada clase de trabajo, los tiempos de servicio y las visitas a cada recurso son: $S_{A,CPU}$=1/10 s; $S_{A,d}$=1/3 s; $S_{B,CPU}$=2/5 s; $S_{B,d}$=1 s; $V_{A,CPU}$=10; $V_{A,d}$=9; $V_{B,CPU}$=5; $V_{B,d}$=4. Las probabilidades de los NDT se calcularon como: $p_{1,A}$=1/10; $p_{2,A}$=9/10; $p_{1,B}$=1/5; y $p_{2,B}$=4/5. Utilizando la ec. (6) con CL=2 (c=A, B), se construyó un módulo de CC para dos clases de trabajo, y se lo utilizó para representar tanto a la CPU como al

disco. Se adoptaron como variables de salida las colas de trabajos en cada recurso y la velocidad de procesamiento, para cada clase. En la Fig. 9, se muestra el esquema modular utilizado, y el módulo jerárquico posteriormente construido para un servidor central con una CPU y un disco, y para dos clases de trabajos. Los resultados dinámicos se presentan en la Fig. 10. Puede observarse los elevados tiempos requeridos para la estabilización de las colas de trabajo en los discos (> 500 s), con respecto a las otras variables del SI. Los valores estacionarios coinciden con los resultados obtenidos por el algoritmo MVA para sistemas abiertos con clases múltiples [2]. Como era de esperar, al alcanzarse el estado estacionario se verifica: $x_A \to \lambda_A$; y $x_B \to \lambda_B$.

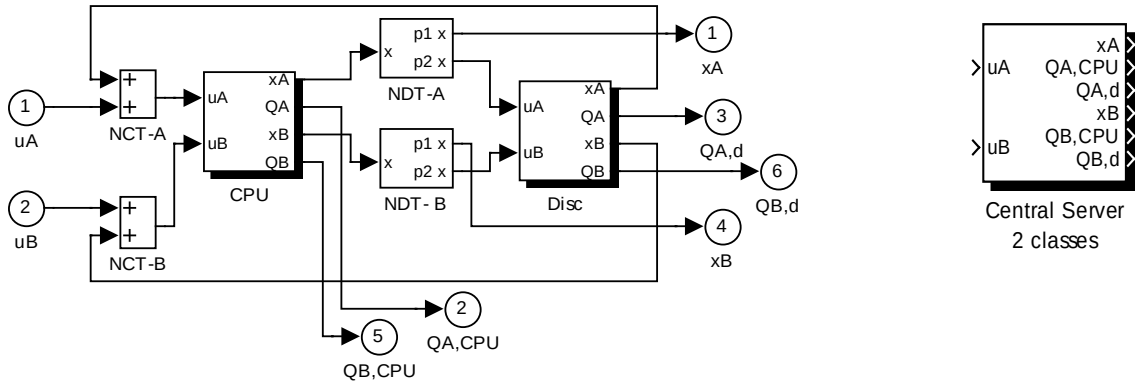


Figura 9. Esquema modular para la simulación numérica del Ejemplo 3, y módulo jerárquico equivalente del servidor central con dos clases de trabajos (salidas arbitrariamente seleccionadas).

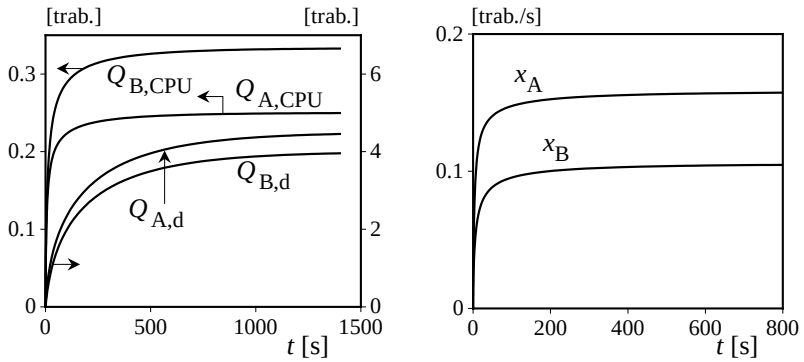


Figura 10. *Evoluciones dinámicas de algunas variables del SI* del Ejemplo 3.

Ejemplo 4. Se utiliza el servidor del *Ejemplo 3* para resolver dos colas de trabajos "batch" (clases A y B), con N_A=N_B=1. Después de los primeros 4 s, se incrementa el número de trabajos de clase A procesados (N_A=2). En la Fig 11, se muestra el esquema modular utilizado para la simulación, y los resultados dinámicos correspondientes. La discontinuidad observada a t=4 s en $Q_{A,CPU}$ corresponde al ingreso del nuevo trabajo de la clase A directamente a la cola de la CPU. La variable de estado estacionario más afectada después del ingreso del nuevo trabajo es el número de trabajos en cola en el disco, correspondiente a la clase A. Por otra parte, puede observarse que los tiempos de estabilización de todas las variables en el caso "batch", resultaron notoriamente inferiores a los observados en la operación transaccional (*Ejemplo 3*). En efecto, dicha operación resultó más exigente como consecuencia del mayor número de trabajos (≈9) residentes en el SI.

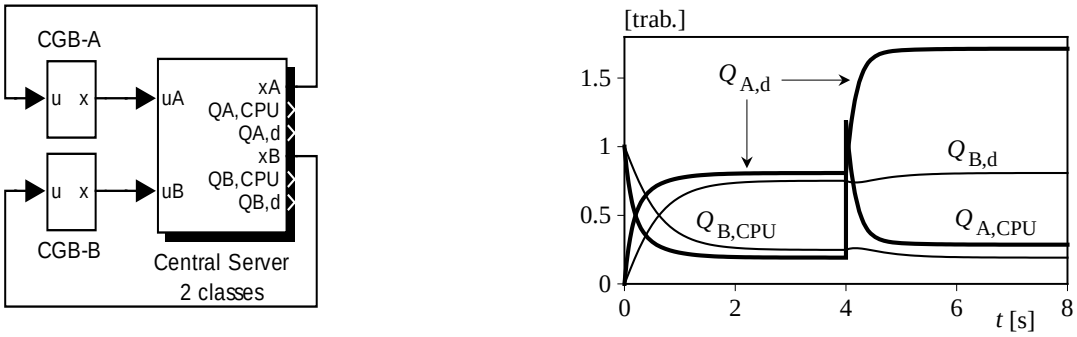


Figura 11. Esquema modular y simulación numérica dinámica del Ejemplo 4.

6. Discusión Final

Se desarrollaron modelos matemáticos continuos y determinísticos para algunos dispositivos básicos, comúnmente utilizados en la descripción de un SI a través de un modelo operacional. El carácter gráfico/modular conseguido, permite fácilmente implementar el modelo del SI, y simular su comportamiento dinámico. La variable de interconexión de los módulos propuestos es el flujo de trabajos. Sin embargo, al desarrollar diferentes módulos con el objeto de confeccionar una "biblioteca" de dispositivos, es conveniente dejar disponibles otras variables de salida (típicamente, las colas de trabajo), para posibilitar la evaluación de las medidas de "performance". Si bien no se presentaron resultados dinámicos de variables de "performance" típicas (como tiempos de respuesta), su evaluación a partir de las variables disponibles es inmediata.

Se derivó una expresión dinámica para un CC que atiende una clase única de trabajos, y se la extendió al caso de múltiples clases. La validez de estas ecuaciones parece más justificada en un SI con un nivel de carga elevado; dado que en tal caso, el funcionamiento de cada dispositivo puede aproximarse mejor a través de variables continuas en el tiempo. Pero, independientemente del nivel de carga del SI, la tendencia asintótica de las ecuaciones provee resultados de estado estacionario coincidentes con los obtenidos a partir de los algoritmos clásicos del análisis de un SI (técnicas algebraicas de resolución de sistemas abiertos, MVA aproximados para los sistemas cerrados, etc.).

Las aplicaciones desarrolladas hasta la fecha de la metodología de simulación propuesta, se reducen a una serie de ejemplos básicos como los comentados. Sin embargo, el carácter modular de los dispositivos, la facilidad de la construcción de los modelos, y la posibilidad de elaborar módulos de mayor jerarquía, lo presentan como una herramienta interesante a la hora de evaluar un SI de mayor complejidad, como puede ser una red informática. A tales efectos, se requerirá el desarrollo de nuevos módulos para dispositivos típicos de red (por ej., un "router"); o de módulos que tengan en cuenta efectos no contemplados (por ej., manipulación de trabajos en cola con prioridad).

El carácter dinámico de los módulos presentados permite detectar y evaluar comportamientos críticos del SI durante un transitorio de magnitud importante (como podría ser el provocada por la salida de servicio de un dispositivo en una red), que quizás no sería detectado en un análisis estacionario (por ej., una oscilación o un sobrevalor de una variable que produzca la saturación temporaria de un dispositivo). También, podrían evaluarse políticas de control y optimización dinámica, manipulando tráfico y/o encaminamiento de cargas de trabajos, tendientes a mejorar la "performance" global del SI. En la tal caso, se requeriría probablemente la definición de nuevas métricas de "performance" dinámica.

Por último, la herramienta propuesta permite intuitivamente la construcción del modelo de simulación a partir de un modelo operacional clásico. Por tal motivo, puede resultar de interés su utilización en cursos universitarios de grado, dedicados al estudio de SI y a la determinación de sus medidas de "performance".

Referencias

1. Iglehart, D., G. Shedler (1980). "Regenerative Simulation of Response Times in Networks of Queues". Lectures Notes in Control and Information Sciences, Vol. 26, Balakrishnan y Thoma Ed. Springer-Verlag, Berlin.
2. Lazowska, E.D., J. Zahorjan, G.Scott Graham, K.C. Sevcik (1984). "Quantitative System Performance. Computer System Analysis Using Queueing Network Models". Prentice Hall, Inc., Englewood Cliffs, New Jersey.
3. Conway, A., N. Georganas (1989). "Queueing Networks. Exact Computational Algorithms". The MIT Press, Massachusetts.
4. Puigjaner Trepat, R., J. Serrano, A. Rubio (1992). "Evaluación y Explotación de Sistemas Informáticos". ISBN 84-7738-316, Ed. Síntesis, Madrid.
5. Kleinrock, L. (1993). "On the Modeling and Analysis of Computer Networks". *Proc. IEEE*, vol. 81, N° 8, 1179-1191.
6. Menascé, D., V. Almeida (1998). "Capacity Planning for Web Performance. Metrics, Models, and Methods". Prentice Hall, Inc., Englewood Cliffs, New Jersey.
7. Murata, T. (1989). "Petri Nets: Properties, Analysis and Applications". *Proc. IEEE*, vol. 77, N° 4, 541-580.
8. Cohen, G., P. Moller, J. Quadrat, M. Viot (1989). "Algebraic Tools for the Performance Evaluation of Discrete Event Systems". *Proc. IEEE*, vol. 77, N° 1, 39-58.
9. Ho, Y. (1989). "Dynamics of Discrete Event Systems". *Proc. IEEE*, vol. 77, N° 1, 3-6.
10. Cao, X. (1989). "A Comparison of the Dynamics of Continuous and Discrete Event Systems". *Proc. IEEE*, vol. 77, N° 1, 7-13.
11. — Matlab V. 5.3 / Simulink V. 3.0. Licencia No 155062 – CERET – Fac. Regional Santa Fe (Univ. Tecnológica Nacional).