

Mathematical modeling of the performance of a computer system.

Félix Armando Fermín Pérez

Universidad Nacional Mayor de San Marcos

Facultad de Ingeniería de Sistemas e Informática

Lima, Perú

fferminp@unmsm.edu.pe

Abstract

Generally the management of computer systems is manual; autonomic computing by self-management tries to minimize human intervention using autonomic controllers; for this, first the mathematical modeling of the system under study is performed, then is designed a controller that governs the behavior of the system. In this case, the determination of the mathematical model of a web server is based on a black box model by system identification, using data collected and stored in its own log during operation of the computer system under study.

Keywords: autonomic computing, self-management, system identification.

1 Introducción

La tecnología influye en cada aspecto de la vida cotidiana; las aplicaciones informáticas, por ejemplo, son cada vez más complejas, heterogéneas y dinámicas, pero también la infraestructura de información, como la Internet que incorpora grandes cantidades de recursos informáticos y de comunicación, almacenamiento de datos y redes de sensores, con el riesgo de que se tornen frágiles, inmanejables e inseguras. Así, se hace necesario contar con administradores de servicios y de servidores informáticos, experimentados y dedicados, además de herramientas software de monitoreo y supervisión, para asegurar los niveles de calidad de servicio pactados (Fermín, 2012).

Diao et al. (2005) promueven la utilización de la computación autónoma mediante sistemas de control automático en lazo cerrado, reduciendo la

intervención humana, que Fox y Patterson (2003) también han identificado como parte principal del problema, debido al uso de procedimientos ad hoc donde la gestión de recursos aún depende fuertemente del control y administración manual. Sobre la computación autónoma, Kephart y Chess (2003) mencionan que la idea es que un sistema informático funcione igual que el sistema nervioso autónomo humano cuando regula nuestra temperatura, respiración, ritmo cardíaco y otros sin que uno se halle siempre consciente de ello, esto es, promueve la menor intervención humana en la administración de la performance de los sistemas informáticos tendiendo hacia la auto-administración de los mismos. Tal auto-administración se caracteriza por las propiedades de auto-configuración (self-configuration), auto-curación (self-healing), auto-optimización (self-optimization) y auto-protección (self-protection).

En la visión de la computación autónoma los administradores humanos simplemente especifican los objetivos de alto nivel del negocio, los que sirven como guía para los procesos autónomos subyacentes. Así los administradores humanos se concentran más fácilmente en definir las políticas del negocio, a alto nivel, y se liberan de tratar permanentemente con los detalles técnicos de bajo nivel, necesarios para alcanzar los objetivos, ya que estas tareas son ahora realizadas por el sistema autónomo mediante un controlador autónomo en lazo cerrado, que monitorea el sistema permanentemente, utiliza los datos recolectados del propio sistema en funcionamiento, compara estas métricas con las propuestas por el administrador humano, y controla la performance del sistema, por ejemplo, manteniendo el tiempo de respuesta del sistema dentro de niveles prefijados.

De acuerdo con la teoría de control, el diseño de un controlador depende de un buen modelo matemático. En el caso de un sistema informático, primero debe tenerse un modelo matemático para luego diseñar un controlador en lazo cerrado o realimentado, pero sucede que los sistemas informáticos son bastante complicados de modelar ya que su comportamiento es altamente estocástico. Según Hellerstein et al (2004) se ha utilizado la teoría de colas para modelarlos, tratándolos como redes de colas y de servidores, bastante bien pero principalmente en el modelado del comportamiento estacionario y no cuando se trata de modelar el comportamiento muchas veces altamente dinámico de la respuesta temporal de un sistema informático en la zona transitoria, donde la tarea se complica.

De manera que en el presente artículo se trata el modelado matemático de un sistema informático mediante la identificación de sistemas, enfoque empírico donde según Lung (1987) deben identificarse los parámetros de entrada y salida del sistema en estudio, basándose en los datos recolectados del mismo sistema en funcionamiento, para luego construir un modelo paramétrico, como el ARX por ejemplo, con las técnicas estadísticas de autoregresión. La sección 2 describe alguna teoría básica sobre las métricas para el monitoreo de la performance de un sistema informático. En la sección 3 se trata el modelado matemático, y en la sección 4 se describe el experimento realizado; finalmente en la sección 5 se describen las conclusiones y trabajos futuros.

2 Monitoreo de la performance.

En la computación autónoma los datos obtenidos del monitoreo de la performance del sistema en estudio contribuye fundamentalmente en la representación del estado o del comportamiento del sistema, esto es, en el modelo matemático del mismo. Según Lalanda et al. (2013), conocer el estado del sistema desde las perspectivas funcionales y no funcionales es vital para llevar a cabo las operaciones necesarias que permitan lograr los objetivos en el nivel deseado y el monitoreo de la performance permite saber cuan bien lo está logrando. Generalmente los datos de la performance se consiguen vía el log del sistema en

estudio, con herramientas de análisis utilizando técnicas de análisis estadísticas, principalmente.

Entre las métricas de performance inicialmente se encontraba la velocidad de procesamiento, pero al agregarse más componentes a la infraestructura informática, surgieron nuevas métricas, siendo las principales las que proporcionan una idea del rendimiento o trabajo realizado en un periodo de tiempo, la utilización de un componente, o el tiempo en realizar una tarea en particular como por ejemplo el tiempo de respuesta. Lalanda et al. (2013) mencionan que las métricas de performance más populares son las siguientes:

- Número de operaciones en punto flotante por segundo (FLOPS), representa una idea del rendimiento del procesamiento, realiza comparaciones entre máquinas que procesan complejos algoritmos matemáticos con punto flotante en aplicaciones científicas.
- Tiempo de respuesta, representa la duración en tiempo que le toma a un sistema llevar a cabo una unidad de procesamiento funcional. Se le considera como una medición de la duración en tiempo de la reacción a una entrada determinada y es utilizada principalmente en sistemas interactivos. La sensibilidad es también una métrica utilizada especialmente en la medición de sistemas en tiempo real, consiste en el tiempo transcurrido entre el inicio y fin de la ejecución de una tarea o hilo.
- Latencia, medida del retardo experimentado en un sistema, generalmente se le utiliza en la descripción de los elementos de comunicación de datos, para tener una idea de la performance de la red. Toma en cuenta no solo el tiempo de procesamiento de la CPU sino también los retardos de las colas durante el transporte de un paquete de datos, por ejemplo.
- Utilización y carga, métricas entrelazadas y utilizadas para comprender la función de administración de recursos y proporcionan una medida de cuan bien se están utilizando los componentes de un sistema y se describe como un porcentaje de utilidad. La carga mide el trabajo realizado por el sistema y usualmente es representado como una carga promedio en un periodo de tiempo.

Existen muchas otras métricas de performance:

- número de transacciones por unidad de costo.

- función de confiabilidad, tiempo en el que un sistema ha estado funcionando sin fallar.
- función de disponibilidad, indica que el sistema está listo para ser utilizado cuando sea necesario.
- tamaño o peso del sistema, indica la portabilidad.
- performance por vatio, representa la tasa de cómputo por vatio consumido.
- calor generado por los componentes ya que en sistemas grandes es costoso un sistema de refrigeración.

Todas ellas entre otras más permiten conocer mejor el estado no funcional de un sistema o proporcionar un medio para detectar un evento que ha ocurrido y ha modificado el comportamiento no funcional. Así, en el presente caso se ha elegido inicialmente como métrica al tiempo de respuesta, ya que el sistema informático en estudio es un servidor web, por esencia de comportamiento interactivo, de manera que lo que se mide es la duración de la reacción a una entrada determinada.

3 Modelado matemático.

En la computación autónoma, basado en la teoría de control realimentado, para diseñar un controlador autónomo primero debe hallarse el modelo matemático del sistema en estudio, tal como se muestra en la Figura N° 1. El modelo matemático de un servidor informático se puede hallar mediante dos enfoques: uno basado en las leyes básicas, y otro en un enfoque empírico denominado identificación de sistemas. Parekh et al. (2002) indican que en trabajos previos se ha tratado de utilizar principios, axiomas, postulados, leyes o teorías básicas para determinar el modelo matemático de sistemas informáticos, pero sin éxito ya que es difícil construir un modelo debido a su naturaleza compleja y estocástica; además es necesario tener un conocimiento detallado del sistema en estudio, más aún, cuando cada cierto tiempo se va actualizando las versiones del software, y finalmente que en este enfoque no se considera la validación del modelo.

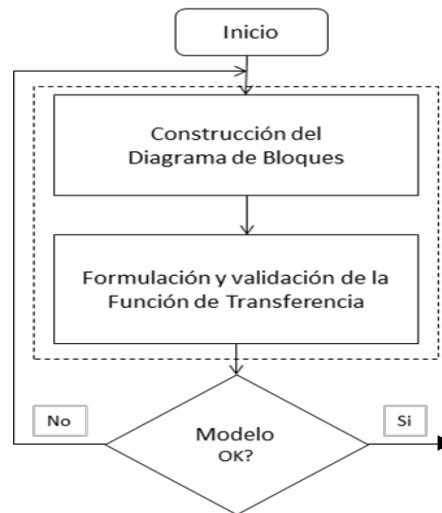


Figura N° 1. Modelado matemático basado en la teoría de control. (Adaptado de Hellerstein, 2004).

En contraste, según Ljung (1987) la identificación de sistemas es un enfoque empírico donde debe identificarse los parámetros de entrada y salida del sistema en estudio, para luego construir un modelo paramétrico, como el ARX por ejemplo, mediante técnicas estadísticas de autoregresión. Este modelo o ecuación paramétrica relaciona los parámetros de entrada y de salida del sistema de acuerdo a la siguiente ecuación:

$$y(k+1) = Ay(k) + Bu(k) \quad (1)$$

- donde
- $y(k)$: variable de salida
 - $u(k)$: variable de entrada
 - A, B : parámetros de autoregresión
 - k : muestra k -ésima.

Este enfoque empírico trata al sistema en estudio como una caja negra, de manera que no afecta la complejidad del sistema o la falta de conocimiento experto, incluso cuando se actualicen las versiones del software bastaría con estimar nuevamente los parámetros del modelo. Así, para un servidor web Apache, la ecuación paramétrica relaciona el parámetro entrada, Max Clients (MC), un parámetro de configuración del servidor web Apache que determina el número máximo de conexiones simultáneas de clientes que pueden ser servidos; y el parámetro de salida Tiempo de respuesta (TR), que indica lo rápido que se

responde a las solicitudes de servicio de los clientes del servidor, ver Figura N° 2.



Figura N° 2. Entrada y salida del sistema a modelar. (Elaboración propia).

En (Hellerstein et al, 2004) se propone realizar la identificación de sistemas informáticos, como los servidores web, de la siguiente manera:

- 1.- Especificar el alcance de lo que se va a modelar en base a las entradas y salidas consideradas.
- 2.- Diseñar experimentos y recopilar datos que sean suficientes para estimar los parámetros de la ecuación diferencial lineal del orden deseado.
- 3.- Estimar los parámetros del modelo utilizando las técnicas de mínimos cuadrados.
- 4.- Evaluar la calidad de ajuste del modelo y si la calidad del modelo debe mejorarse, entonces debe revisarse uno o más de los pasos anteriores.

4 Experimento.

La arquitectura del experimento implementado, se muestra en la Figura N° 3.

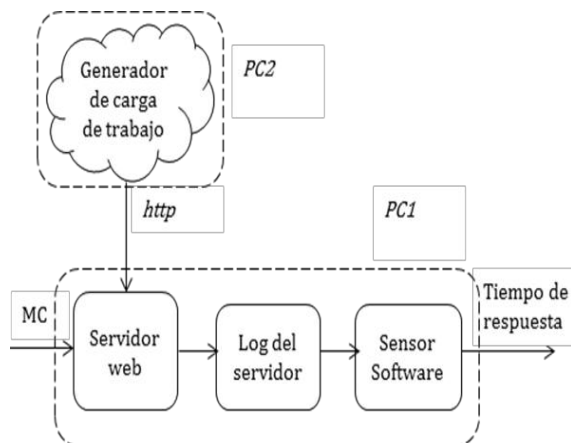


Figura N° 3. Arquitectura para la identificación del modelo de un servidor web (Elaboración propia).

La computadora personal PC1 es el servidor web Apache, asimismo contiene el sensor software que

recoge los tiempos de inicio y fin de cada http que ingresa al servidor web, datos que se almacenan en el log del mismo servidor, y luego se realiza el cálculo del tiempo de respuesta de cada http completado en una unidad de tiempo y el tiempo de respuesta promedio del servidor. El servidor web por sí mismo no hace nada, por ello, la computadora personal PC2 contiene un generador de carga de trabajo, que simula la actividad de los usuarios que desean acceder al servidor web; en este caso se utilizó el JMeter una aplicación generadora de carga de trabajo y que forma parte del proyecto Apache.

La operación del servidor web, no solo depende de la actividad de los usuarios, sino también de la señal de entrada MaxClients (MC), que toma forma de una sinusoide discreta variable que excita al servidor web junto con las solicitudes http del generador de carga de trabajo (PC2). De manera que con la actividad del servidor web almacenada en su log, un sensor software calcula los valores de la señal de salida Tiempo de Respuesta (TR), mostrados en la Figura N° 4.

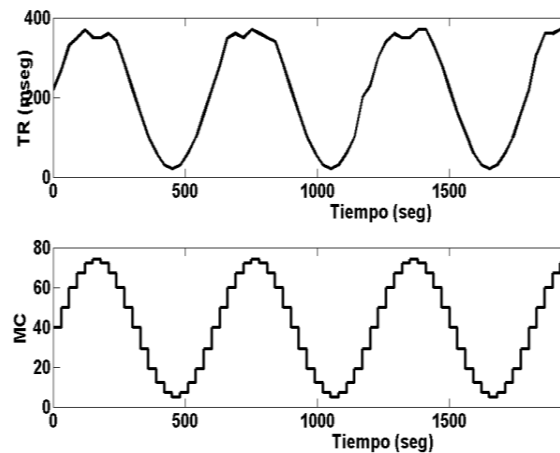


Figura N° 4. Señal de entrada MaxClients MC y señal de salida Tiempo de respuesta TR.

Con los datos de MC y TR obtenidos se estiman los parámetros de regresión A y B de la ecuación paramétrica ARX, haciendo uso de las técnicas de mínimos cuadrados, implementadas en el ToolBox Identificación de Sistemas del Matlab, logrando la siguiente ecuación paramétrica de primer orden:

$$TR(k+1) = 0.06545TR(k) + 4.984MC(k+1)$$

Se puede observar que el Tiempo de respuesta actual depende del Tiempo de respuesta anterior y del parámetro de entrada MaxClients. El modelo hallado es evaluado utilizando la métrica r^2 , calidad de ajuste, del ToolBox utilizado, que indica el porcentaje de variación respecto a la señal original. En el caso de estudio, el modelo hallado tiene una calidad de ajuste del 87%, lo que se puede considerar como un modelo aceptable. En la Figura N° 5 puede observarse la gran similitud entre la señal de salida medida y la señal de salida estimada.

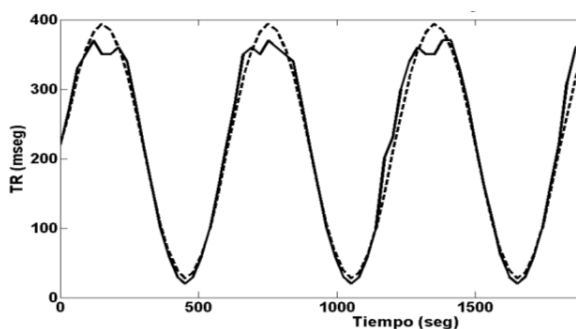


Figura N° 5. TR medida y TR estimada.

5 Conclusiones.

La identificación del comportamiento de un sistema informático tratado como una caja negra es posible, para ello debe simularse el funcionamiento del mismo con hardware y herramientas software como el Jmeter y Matlab, por ejemplo.

El modelo matemático determinado en base a los datos recopilados del log del sistema en estudio se aproxima bastante al modelo real, en este caso se obtuvo un 87% de calidad de ajuste.

El sensor software implementado ha permitido calcular el tiempo de respuesta en base a los datos almacenados en el log del mismo servidor.

De los datos tomados se observa que los sistemas informáticos poseen un comportamiento cercano al lineal solo en tramos por lo que se sugiere experimentar con modelos matemáticos no lineales para comparar la calidad de ajuste de ambos.

Como trabajos futuros se plantea diseñar un controlador autónomo basado en el modelo lineal

ARX, aunque más adelante se planteará el diseño de controladores autónomos no lineales para sistemas informáticos en general, ya que el comportamiento temporal no lineal, hace adecuada la utilización de técnicas de inteligencia artificial como la lógica difusa y las redes neuronales.

6 Referencias bibliográficas.

Diao, Y.; Hellerstein, J.; Parekh, S.; Griffith, R.; Kaiser, G.; Phung, D. (2005) *A Control Theory Foundation for Self-Managing Computing Systems*, IEEE Journal on Selected Areas in Communications, Vol. 23, Issue 12, pp. 2213 – 2222. IEEE. DOI: 10.1109/JSAC.2005.857206.

Fermín F. (2012). *La Teoría de Control y la Gestión Autónoma de Servidores Web*. Memorias del IV Congreso Internacional de Computación y Telecomunicaciones. ISBN 978-612-4050-57-2.

Fox, A.; Patterson, D. (2003). *Self-repairing computers*, Scientific American, Jun2003, Vol. 288 Issue 6, pp. 54-61.

Hellerstein, J.; Diao, Y.; Parekh, S.; Tilbury, D. (2004). *Feedback Control of Computing Systems*. Hoboken, New Jersey: John Wiley & Sons, Inc. ISBN 0-471-26637-X

Kephart, J.; Chess, D. (2003). *The vision of autonomic computing*. Computer, Jan2003, Vol. 36, Issue 1, pp. 41-50. IEEE. DOI: 10.1109/MC.2003.1160055

Lalanda, P.; McCann, J.; Diaconescu, A. (2013). *Autonomic Computing. Principles, Design and Implementation*. London: Springer-Verlag. ISBN 978-1-4471-5006-0

Ljung L. (1987). *System Identification: Theory for the User*. Englewood Cliffs, New Jersey: Prentice Hall.

Parekh, S.; Gandhi, N.; Hellerstein, J.; Tilbury, D.; Jayram, T.; Bigus, J. (2002). *Using control theory to achieve service level objectives in performance management*. Real-Time Systems. Jul2002, Vol. 23, Issue 1/2, pp. 127-141. Norwell, Massachusetts: Kluwer Academic Publishers. DOI: 10.1023/A:1015350520175.