



HAL
open science

Control de balanceo de carga de un grupo de servidores de red

Miquel Talavera Foix, Carolina Albea-Sanchez

► **To cite this version:**

Miquel Talavera Foix, Carolina Albea-Sanchez. Control de balanceo de carga de un grupo de servidores de red. XXXV Jornadas de Automática, Sep 2014, Valencia, España. pp.478-483. hal-01068501

HAL Id: hal-01068501

<https://hal.science/hal-01068501>

Submitted on 25 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Control de balanceo de carga de un grupo de servidores de red

Miquel Talavera-Foix

CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse France, mtalaver@laas.fr

Carolina Albea

CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse France
Univ de Toulouse, UPS, LAAS, F-31400 Toulouse France, calbea@laas.fr

Resumen

Debido al incremento de servicios en Internet y del número de usuarios, se ha multiplicado la demanda de prestaciones de los servidores web. Si se usa adecuadamente los “clústeres”, se puede llegar a una potencia correcta de procesamiento que permita satisfacer los requerimientos. El equilibrado de carga, o “load balancing”, permite crear una red robusta y eficiente. En este trabajo se presenta una ley de control distribuido inspirada en la “teoría de consenso”. El bucle cerrado asegura la distribución equilibrada de la carga, además de garantizar una estabilidad asintótica. Gracias a un control descentralizado, la carga computacional se reduce y permite que a su vez el sistema sea escalable. Para garantizar la estabilidad local del modelo se hace una estimación de la cuenca atracción. El control distribuido se ha validado mediante simulaciones en Matlab & Simulink y Network Simulator 2.

Palabras clave: balanceo de carga, control por consensus, estabilidad de Lyapunov, dominio de atracción.

1. INTRODUCCIÓN

La mayoría de los servidores web deben funcionar veinticuatro horas al día, siete días a la semana, con total disponibilidad y fiabilidad. A veces, la potencia de cálculo de un solo servidor no es suficiente para tratar todas las solicitudes entrantes de una página web. Como consecuencia, los usuarios pueden esperar bastante para visualizar la página o incluso puede que no la visualicen a causa de las pérdidas de datos. Por tanto, interconectando un conjunto de servidores de una manera distribuida a través de una red compartida (“clústeres”) se puede mejorar significativamente el rendimiento global del sistema [4]. Asimismo, se mejora la escalabilidad de los servidores, resultando una red más segura.

En un sistema distribuido, los servidores trabajan independientemente los unos de los otros. Sus procesadores pueden tener diferentes capacidades así como distintas cargas iniciales. Por razones de

seguridad y robustez, se distribuye el flujo de solicitudes entrantes de una manera eficiente, como en el caso de una arquitectura en paralelo [10]. Esta concepción hace que ningún servidor pueda estar sobrecargado mientras haya otros con un nivel de carga inferior. La estrategia que distribuye equilibradamente la carga del sistema se conoce como “load balancing”.

La distribución de carga se puede implementar con un controlador descentralizado. Por lo que se requiere un conocimiento preciso del estado de cada servidor y de sus características. El equilibrio de carga evita las pérdidas de datos (sobrecargas) y garantiza que los recursos del sistema se utilicen adecuadamente.

Actualmente, existen diferentes estrategias para realizar una distribución de carga. Algunos métodos actuales se pueden encontrar en [8]. Un tutorial entre las diferentes estrategias tanto estáticas como dinámicas se puede hallar en [7]. Por otro lado, en [11] se encuentran diversas comparaciones entre métodos determinísticos.

Normalmente, la carga computacional de la mayoría de estas estrategias es muy elevada. “Shortest Queue Policy” [12] coloca la solicitud entrante al servidor que tenga una cola de espera inferior. “Shortest Expected Delay” [3] asigna la solicitud entrante al servidor que minimice el tiempo de proceso de la tarea. “Sender-Receiver Initiated Diffusion” [5] transfiere tareas de los servidores altamente cargados a otros más descargados.

Este artículo propone una estrategia de control distribuido para la gestión de la carga en un “clúster” de servidores modificando la tasa (“rate”) de entrada. Se ha realizado un estudio teórico y práctico para un caso general, de modo que sea fácilmente implementable en otras aplicaciones. Se ha propuesto un modelo dinámico-determinista y una ley de control distribuido inspirada en la teoría de consenso [9]. Se han realizado varias hipótesis para proponer una primera solución. Por ejemplo: se asume que todos los servidores tienen una misma estructura; que el volumen de trabajo es constante y que todas las solicitudes entrantes pueden ser procesadas por todos los servidores. El obje-

tivo principal de este artículo no es sólo asegurar que el sistema sea seguro, sino también evitar las pérdidas de datos. Se ha conseguido que la carga computacional del sistema disminuya y que el sistema sea escalable. Se comprueba que la carga total se mantiene y que las colas de cada servidor convergen asintóticamente a un nivel medio. Finalmente se ha validado el modelo mediante simulación en Matlab-Simulink y en Network Simulator 2.

Este artículo está organizado de la siguiente manera: en la sección II, se modeliza el sistema y se formula el problema; en la sección III, se presenta la ley de control basada en la teoría de consenso; se realiza una estimación de la cuenta de atracción en la sección IV; se presentan algunas simulaciones en la sección V y finalmente, el artículo termina con una sección de conclusiones y de trabajo futuro.

2. DISTRIBUCIÓN DE LA CARGA

2.1. MODELO

Vamos a considerar un sistema compuesto por una fuente S que distribuye una tasa entrante constante A sobre un conjunto de servidores N (ver Figura 1). La tasa de carga tratada Y , representa la capacidad de cada procesador y se supone constante. El objetivo de los N servidores no es otro que procesar la carga entrante. Las tasas A y Y se denotan respectivamente:

$$A = [a_1 \ a_2 \ \dots \ a_{N-1} \ a_N]^T$$

$$Y = [y_1 \ y_2 \ \dots \ y_{N-1} \ y_N]^T.$$

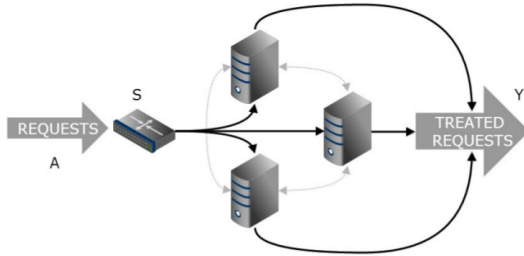


Figura 1: Sistema considerado con $N = 3$ servidores.

Hipótesis 1 *La carga global del sistema se conserva:*

$$\sum_{i=1}^N a_i = \sum_{i=1}^N y_i.$$

Para procesar correctamente la carga de trabajo de la fuente, se tiene en cuenta la cola de espera de los paquetes que aún no se han tratado. De esta manera, se introduce el estado: *tamaño de la cola*, $q_i \in \mathfrak{R}_{\geq 0}$, donde su variación \dot{q}_i corresponde a la diferencia entre la tasa de entrada, a_i , y la tasa de procesamiento máximo, y_i . Nota que toda solución $t \mapsto q_i(t)$ tiene que satisfacer que $q_i(t) \in \mathfrak{R}_{\geq 0}$ para todo $t \geq 0$.

Por otro lado, se supone que los servidores tienen diferentes cargas iniciales:

$$q_0 = [q_{0_1} \ q_{0_2} \ \dots \ q_{0_{N-1}} \ q_{0_N}]^T. \quad (1)$$

El modelo en forma compacta es:

$$\dot{q} = A - Y + u, \quad (2)$$

donde u es la ley de control que distribuye la carga.

Problema 1 *Nuestro objetivo es hacer un equilibrio de carga del sistema (1)-(2).*

Nota 1 *Una vez presentados los resultados de esta situación específica, se llevarán a cabo simulaciones para un caso más general donde la tasa entrante varíe en función del tiempo. Para diferenciarlas, la tasa entrante variante se denominará $A(t)$.*

2.2. CONTROL POR CONSENSUS

2.2.1. Conectividad entre servidores

Vamos a definir una matriz de conexión. Para ello se define una matriz adjunta

$$Ad = [ad_{ij}] = \begin{cases} 1 & \text{si } i \neq j, \\ 0 & \text{si } i = j, \end{cases}$$

y una matriz diagonal Δ ,

$$\Delta = \begin{cases} 0 & \text{si } i \neq j, \\ \sum_{k \in \mathbb{N}, k \neq i} ad_{ik} & \text{si } i = j. \end{cases}$$

Por tanto, la matriz Laplaciana del grafo indirecto resultante es:

$$L = \Delta - Ad. \quad (3)$$

La matriz Laplaciana de un grafo indirecto es simétrica y semi-definida positiva. Representa el grafo completo de la Figura 1.

2.2.2. Ley de control propuesta

El objetivo de este artículo es el de proponer una ley de control eficiente que asigne la entrada u_i a cada servidor i para cumplir la siguiente proposición:

Proposición 1 *Todas las longitudes de cola convergen a un punto promedio:*

$$\lim_{t \rightarrow +\infty} q_i = q^*,$$

donde q^* es el punto promedio.

Usando la ecuación del modelo (1)-(2) y la matriz Laplaciana (3), se diseña un controlador dinámico-distribuido que representa el flujo intercambiado entre los servidores.

$$u = -K_p Lq + q_c \quad (4)$$

$$\dot{q}_c = -K_i Lq, \quad (5)$$

donde K_p y $K_i \in \mathfrak{R}_{\geq 0}$ son las ganancias proporcional e integral, respectivamente. Este control es un algoritmo clásico de consensus para sistemas multi-agentes [9], que asegura que todos los tamaños de las colas convergen a un nivel promedio, q^* .

En el siguiente lema, se presenta una característica importante que define la elección de (4) y (5).

Lema 1 *La ley de control distribuido (4) y (5) satisface la Proposición 1, que remarca que todos los tamaños de las colas convergen a un mismo valor promedio.*

Prueba 1 *Se establece a partir de aplicar (3) en (4) y (5).*

2.2.3. Estabilidad en el sentido de Lyapunov

Sea V una función definida positiva:

$$V = q^T Lq + \frac{(q_c + A - Y)^2}{K_i}. \quad (6)$$

Su derivada correspondiente es:

$$\dot{V} = 2q^T L\dot{q} + 2\frac{\dot{q}_c(q_c + A - Y)}{K_i}.$$

Si se usa (2) y (5) se llega a:

$$\dot{V} = -2K_p q^T Lq \leq 0.$$

Esto implica estabilidad asintótica.

3. ESTIMACIÓN DE LA CUENCA DE ATRACCIÓN

La mayoría de las aplicaciones de control tienen restricciones del sistema (saturaciones, limitadores de la tasa de bits, especificaciones físicas...) que pueden hacer perder el carácter global de la estabilidad. En diseños de control, es usual aplicar la ley de control ignorando las restricciones. Por

ejemplo, si la ley de control es $u_d = \alpha(x)$, donde x es la variable de estado, se aplica realmente $u = \gamma(u_d) = \gamma(\alpha(x))$, donde γ es la función de saturación.

Si se trabaja en un dominio donde las restricciones no están activas $\gamma(\alpha(x)) = \alpha(x)$, se mantiene la estabilidad local. No obstante, si existe alguna región “prohibida” en el espacio de estado, se debe asegurar que el sistema no abandone la región “segura”. Para considerar estas limitaciones se estima una cuenca de atracción [1]. En [6] se pueden encontrar diversos métodos para la estimación de la cuenca de atracción. Un ejemplo de un inversor “boost” se halla en [2].

Hipótesis 2 *Existe una función de Lyapunov conocida y radialmente no acotada $V(x)$, tal que un conjunto invariante, positivo y compacto Ω , $\frac{\partial V}{\partial x} f(x, \alpha(x)) \geq 0$. M es subconjunto invariante del conjunto para el cual $\dot{V} = 0$ en Ω .*

Mediante el principio de invariancia de la LaSalle, la Hipótesis 2 garantiza que las trayectorias del modelo sin restricciones tienden a M . Note que la Hipótesis 2 también garantiza estabilidad local.

El modelo propuesto (2) tiene una saturación de la tasa total de entrada:

$$rate_i = u_i + a_i \geq 0, \quad (7)$$

donde $rate_i \in \mathfrak{R}_{>0}$. Toda solución $t \mapsto rate_i(t)$ tiene que cumplir $rate_i(t) \in \mathfrak{R}_{>0}$ para todo $t \geq 0$. Esta restricción se debe a que los servidores tienen una única entrada y una única salida, por tanto el balance de carga se efectuará en la entrada del servidor.

Asumiendo la saturación (7) en el modelo (2), una estimación (conservativa) del dominio de atracción se propone en el siguiente teorema:

Teorema 1 *Asumiendo ciertas las Hipótesis 1 y 2, existe una constante $c > 0$, tal que en el dominio $\Omega_c = \{x : V(x) \geq c^{-1}\}$, la restricción (7) se cumplen. Luego, todas las trayectorias del sistema con restricciones, empezando en el dominio Ω_c , tienden a $M \cap \Omega_c$.*

Prueba 2 *Dado que las restricciones se cumplen en Ω_c , los resultados en el sistema sin restricciones son válidos en Ω_c . Como consecuencia, $\dot{V} = 0$ en Ω_c y Ω_c son positivos e invariantes. Asimismo, puesto que $V(x)$ es radialmente no acotado, Ω_c es compacto. El teorema se puede validar aplicando el principio de invariancia de la LaSalle.*

Nota 2 *Dado que $M \cap \Omega_c \subset M$, el teorema garantiza el comportamiento asintótico en el sistema con restricciones.*

Usando el Teorema 1, el problema se reduce a encontrar un valor de $c > 0$ tal que (7) se cumple en los puntos donde $V(x) \geq 0$.

Problema 2 Maximizar c^{-1} :

$$\begin{aligned} s.a. : V(x) - c^{-1} &\leq 0 \\ u_i + a_i &\geq 0 \quad i = 1, \dots, N. \end{aligned} \quad (8)$$

Nota que las restricciones se cumplen tanto en los límites de Ω_c como en el conjunto $V(x) - c^{-1} < 0$.

Teorema 2 Considera el sistema (2), la ley de control (4) y (5), la función de Lyapunov (6) y la constante $c > 0$, entonces el Problema 2 es equivalente a resolver:

$$\begin{bmatrix} cY^T Y & -K_p Y^T L & Y^T \\ -K_p L Y & Y^T Y L & 0 \\ Y & 0 & \frac{Y^T Y I_n}{K_i} \end{bmatrix} \geq 0. \quad (9)$$

Prueba 3 Definimos:

$$x_1 \triangleq q \quad x_2 \triangleq q_c + A - Y, \quad (10)$$

donde $x_1, x_2 \in \mathfrak{R}^{N \times 1}$.

A partir de la ecuación (4), (5) y (10), se obtiene:

$$u = -K_p L x_1 + x_2 - A + Y \in \mathfrak{R}^{N \times 1}. \quad (11)$$

La relación (7) se puede reescribir de la siguiente forma usando (11):

$$K_p L x_1 - x_2 \leq Y.$$

Multiplicando por la izquierda por $2Y^T$:

$$2Y^T Y \geq 2Y^T K_p L x_1 - 2Y^T x_2. \quad (12)$$

Minimizando c , multiplicando por la izquierda (8) por $2Y^T Y$, y a partir de (12), la cuenca de atracción resulta:

$$\begin{aligned} 2Y^T Y &\geq cY^T Y x_1^T L x_1 + cY^T Y \frac{x_2^T I_n x_2}{K_i} \\ +Y^T Y &\geq 2Y^T K_p L x_1 - 2Y^T x_2. \end{aligned}$$

Reescribiendo la última inecuación en forma matricial:

$$\begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} Y^T Y & -K_p Y^T L & Y^T \\ -K_p L Y & cY^T Y L & 0 \\ Y & 0 & \frac{cY^T Y I_n}{K_i} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \geq 0,$$

y aplicando el complemento Schur se obtiene (9).

4. RESULTADOS DE SIMULACIÓN Y DE IMPLEMENTACIÓN

Vamos a utilizar los siguientes parámetros:

$$N = 5$$

$$A = [3,8 \ 4,2 \ 2 \ 3 \ 2]^T \cdot 10^3 \text{ pqt/s}$$

$$Y = [3 \ 5 \ 2 \ 2,25 \ 2,75]^T \cdot 10^3 \text{ pqt/s}$$

$$q_0 = [4500 \ 4000 \ 3500 \ 4775 \ 4225]^T \text{ pqt.}$$

Por simplicidad, se toma:

$$K_p = K_i = 1.$$

Con los valores anteriores, la constante c^{-1} que maximiza la cuenca de atracción es:

$$c^{-1} = 2,9328 \cdot 10^7.$$

Con estos parámetros vamos a realizar simulaciones: en Matlab/Simulink, para validar la teoría desarrollada; y en NS2 para validar el controlador en un entorno más realista.

El modelo desarrollado se muestra escalable y robusto en la región segura. No obstante, no se puede asegurar su estabilidad dentro de la región prohibida.

4.1. MATLAB - SIMULINK

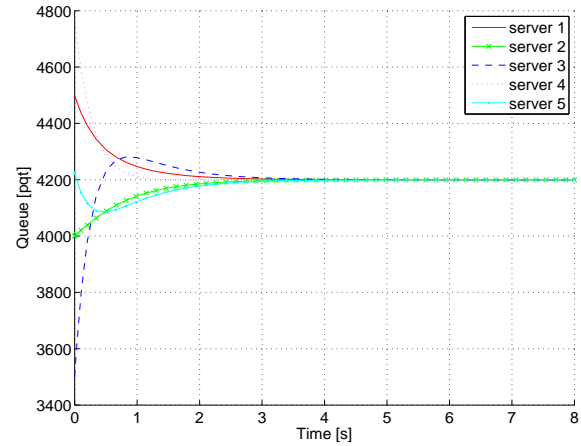


Figura 2: Tamaño de cola i en función del tiempo.

En la Figura 2 se muestra que todos los tamaños de las colas convergen asintóticamente al valor medio $q^* = 4200 \text{ pqt}$.

4.2. NETWORK SIMULATOR 2

NS2 es un simulador “open source” de redes tanto cableadas como inalámbricas. Consta de un núcleo principal escrito en C++ y de una interfaz en un lenguaje OTcl, una versión de Tcl orientada a objetos. El usuario describe una topología de red por medio de códigos en OTcl, y luego el programa principal de NS2 simular dicha topología utilizando los parámetros definidos.

4.2.1. Topología de la red

Se han creado $N + 1$ nodos. El nodo extra representa la fuente S .

Los enlaces son unidireccionales y con un ancho de banda que corresponde a la tasa de solicitudes tratadas, y_i . La política de gestión de cola se

caracteriza como “DropTail”, es decir, primero en entrar primero en salir. Por otro lado, los enlaces tienen un tamaño de cola inicial (1) y la capacidad máxima se fija en 10000 *pqt*.

El tráfico se modeliza como “Constant Bit Rate” (CBR), es decir, la tasa de bits es constante durante toda la simulación. Es muy útil para flujos de datos multimedia con canales de capacidad limitada. Este hecho, permite utilizar toda la capacidad. Se ha empleado el protocolo, “Use Datagram Protocol” (UDP), el cual es uno de los principales protocolos en los que se basa Internet. UDP permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión. Tampoco tiene control ni confirmación de flujo.

La fuente UDP es el nodo $N0$ (“Agent/UDP”) y las destinaciones son los N servidores (“Agent/Null”). El tamaño de paquetes se considera 125 *bytes*.

Para conseguir un tamaño de cola inicial de (1), el “rate” inicial se configura como:

$$rate_0 = [2,55 \ 2,5 \ 1,95 \ 2,6125 \ 2,3875]^T \cdot 10^4 \text{ pqt/s}.$$

El rate inicial se aplica durante 0,2 *s*. Una vez el tamaño de cola es el deseado, se aplica la ley de control de la forma (7).

4.2.2. Discretización del modelo

Como NS2 es un simulador de eventos discretos, se aplica una aproximación discretizada de la ley de control (4) y (5) mediante el Método de Euler hacia atrás.

$$u[kT] = u[(k-1)T] - K_p Lq[kT] + (K_p - K_i T) Lq[(k-1)T],$$

donde T es el periodo de muestreo, el cual se fija para las simulaciones en 50 *ms*.

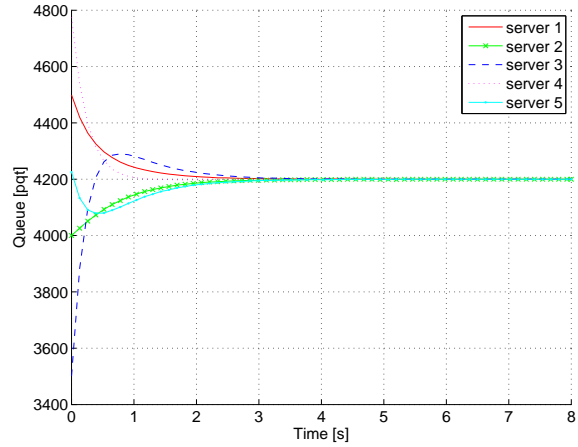
Con la Figura 3a se puede comprobar que la discretización no varia el comportamiento del modelo. Se puede comprobar que se obtiene prácticamente la misma simulación en Matlab/Simulink que en NS2. La Figura 3b muestra como el “rate” es positivo para todo $t \geq 0$, $rate_i(t) \in \mathfrak{R}_{>0}$.

Para crear un entorno más real, se aplica ruido en la aplicación CBR. En este caso, la tasa de entrada es $A(t)$. El sistema converge en cada momento al valor medio de todos los tamaños de cola $q^*(t)$.

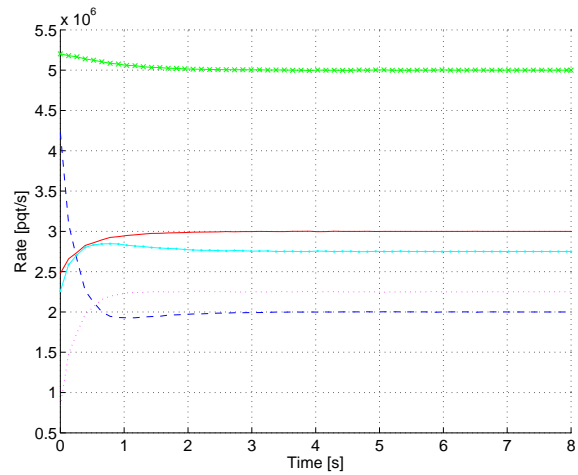
La Figura 4a y la Figura 4b muestra que el sistema es robusto.

5. CONCLUSIONES Y TRABAJO FUTURO

Este artículo propone un control distribuido para un sistema de N servidores. La ley de control ba-



(a) Tamaño de cola i en función del tiempo.



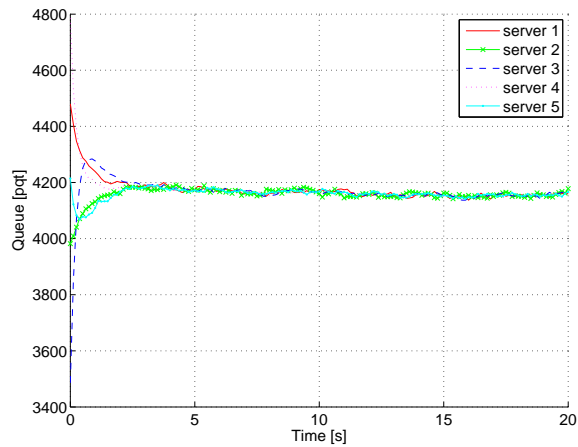
(b) “Rate” i en función del tiempo.

Figura 3: Simulación en NS2

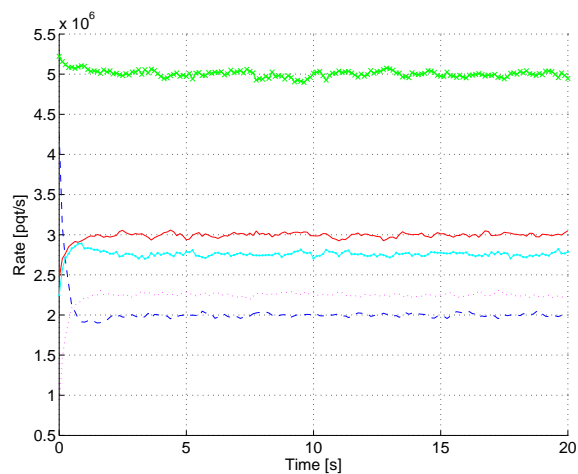
sada en la teoría de consenso permite manipular el tamaño de cola de cada servidor con la finalidad de realizar un equilibrado de carga. La carga computacional del sistema es notablemente inferior comparado con otras estrategias de balanceo de carga, por tanto, se reducirán las pérdidas de datos. Asimismo, el sistema se convierte en robusto y escalable. El control distribuido se ha diseñado de forma general para poder ser implementado en grandes estructuras de servidores.

Por otro lado, se ha presentado un método para estimar la cuenca de atracción que tiene en cuenta las restricciones físicas del sistema. El método se basa en estimar una superficie de Lyapunov donde se cumplen las restricciones.

Un trabajo futuro consistirá en considerar los retardos en la transmisión de la información del estado de cada servidor, así como los retardos de la distribución de los paquetes en la red.



(a) Tamaño de cola i en función del tiempo.



(b) "Rate" i en función del tiempo.

Figura 4: Simulación con ruido en NS2

Agradecimientos

Los autores desean agradecer al proyecto CNES la financiación de este trabajo.

Referencias

- [1] C. Albea and F. Gordillo. Control of the boost DC-AC converter with RL load by energy shaping. In *Decision and Control, In Proc. of the 46th IEEE Conference on*, pages 2417–2422, Dec 2007.
- [2] C. Albea and F. Gordillo. Estimating the attraction domain for the boost inverter. *Asian Journal of Control*, 15(1):169–176, 2013.
- [3] S. A. Banawan and J. Zahorjan. Load sharing in heterogeneous queueing systems. In *IN-FOCOM Proc. of the 8th Annual Joint Conference of the IEEE Computer and Communications Societies. Technology: Emerging or Converging, IEEE*, pages 731–739 vol.2, Apr 1989.

- [4] W. Becker. Dynamic balancing complex workload in workstation networks-challenge, concepts and experience. In *High-Performance Computing and Networking*, pages 407–412. Springer, 1995.
- [5] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7(2):279–301, 1989.
- [6] R. Genesio, M. Tartaglia, and A. Vicino. On the estimation of asymptotic stability regions: State of the art and new proposals. *Automatic Control, In trans. on IEEE*, 30(8):747–755, Aug 1985.
- [7] H. Kameda, El-Z.S. Fathy, I. Ryu, and J. Li. A performance comparison of dynamic vs. static load balancing policies in a mainframe-personal computer network model. In *Decision and Control, In Proc. of the 39th IEEE Conference on*, volume 2, pages 1415–1420 vol.2, 2000.
- [8] H. Kameda, J. Li, C. Kim, and Y. Zhang. *Optimal Load Balancing in Distributed Computer Systems*. Springer Publishing Company, Incorporated, 1st edition, 2011.
- [9] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. In *Proc. of the IEEE*, 95(1):215–233, Jan 2007.
- [10] Z. Tang, J. D. Birdwell, J. Chiasson, C. T. Abdallah, and M. M. Hayat. Closed loop control of a load balancing network with time delays and processor resource constraints. In *Advances in Communication Control Networks*, pages 245–268. Springer, 2005.
- [11] M. H. Willebeek-LeMair and A. P. Reeves. Strategies for dynamic load balancing on highly parallel computers. In *Trans. on IEEE Transactions Parallel and Distributed Systems*, 4(9):979–993, 1993.
- [12] W. Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, 14(1):181–189, 1977.