

Enl@ce: Revista Venezolana de Información,
Tecnología y Conocimiento
ISSN: 1690-7515
Depósito legal pp 200402ZU1624
Año 8: No. 3, Septiembre-Diciembre 2011, pp. 83-101

Cómo citar el artículo (Normas APA):
Barreto, A. y Cardinale Y. (2011). Modelo de balance de carga para un clúster computacional basado en la estabilidad de Lyapunov. *Enl@ce Revista Venezolana de Información, Tecnología y Conocimiento*, 8 (3), 83-101

Modelo de balance de carga para un clúster computacional basado en la estabilidad de Lyapunov

*Aquiles Barreto*¹
*Yudith Cardinale*²

Resumen

Balancear dinámicamente la carga en un clúster computacional, para evitar el desbalance de las asignaciones, es dividir equitativamente y en cualquier instante la carga de trabajo que se ha de repartir entre los nodos. El desbalance puede ser resuelto si uno o más nodos pueden migrar parte de sus trabajos. Una manera de decidir la migración es analizar, a través de una teoría, la estabilidad en el clúster. Para el análisis de estabilidad, los métodos basados en la teoría de Liapunov tienen un lugar destacado por la riqueza conceptual. El objetivo de este trabajo es formular un modelo matemático basado en ecuaciones diferenciales lineales, que compruebe la estabilidad del clúster y permitan la ejecución y migración de los trabajos. En este trabajo se concluye que si las cargas de los nodos tienden a sus puntos de equilibrio, las nuevas tareas de migración no necesitarán la comprobación de la estabilidad de los nodos para que puedan ser ejecutadas.

Palabras clave: clúster computacional, sistemas lineales, balance de carga, estabilidad de Lyapunov, método de Krasovskii

Recibido: 15-07-11 Aceptado: 10-10-11

¹ Ingeniero en Computación. Profesor Agregado del Departamento de Computación y Sistemas de la Universidad de Oriente. Doctorando en Computación de la Universidad Simón Bolívar. Dirección institucional: Departamento de Computación y Sistemas. UDO-Anzoátegui, Apdo. 6001. Barcelona, Venezuela.
Correo electrónico: abarreto@ldc.usb.ve.

² Ingeniera en Informática. Profesora Titular del Departamento de Computación y Tecnología de la Información de la Universidad Simón Bolívar. Magister en Ciencias de la Computación y Doctora en Computación. Dirección institucional: Ofic.: MYS-220A, Apdo. 89000. Caracas 1080-A, Venezuela.
Correo electrónico: yudith@ldc.usb.ve

Load Balancing Model for a Computational Cluster based on Lyapunov Stability

Abstract

Dynamic load balancing on a cluster computing consists on evenly dividing, at any moment of time, the workload to be distributed among the nodes, in order to avoid load unbalance. Load unbalance can be resolved if one or more nodes can migrate some of their work. Using a theory to allow the analysis of stability in the cluster is a way to decide the migration tasks. In the field of stability analysis, methods based on Lyapunov theory have a prominent place for the extent of stability concepts. The aim of this work is to develop a mathematical model based on linear differential equations representing the load, execution, and migration of task on a cluster computing. With the model, it is possible to determine the stability of the cluster. If the workload on the nodes tends to their equilibrium, new migration tasks do not need to check the stability of the nodes so they can be executed.

Key words: Computing Cluster, Linear Systems, Load Balancing, Liapunov Stability, Krasovskii Method

Introducción³

Un clúster computacional es un conjunto de computadoras independientes e interconectadas, trabajando juntas como si fueran un solo recurso, permitiendo procesamiento paralelo o distribuido. Este tipo de sistema provee una alternativa de multiprocesadores de bajo costo. Hoy día, los clústers computacionales son considerados como una plataforma estándar para la computación de alto desempeño (*high performance computing*) (Dongarra, 2011).

Como en toda red, en un clúster, el desbalance puede provocar la saturación de sus nodos y el posterior colapso del mismo, el colapso debe ser

evitado. El balance de carga en un clúster se puede mantener con un procedimiento que divide de la manera más equitativa posible el trabajo que se ha de repartir entre los procesadores, y así evitar el desbalance producto de la misma asignación. El desbalance puede ser resuelto si una o más máquinas pueden migrar parte de sus trabajos a otras máquinas que no estén usando su capacidad máxima en ese momento. Llevar la red de un estado de desbalance a uno de balance es una tarea que debe llevarse a cabo en el menor tiempo posible y una vez alcanzado el balance, este estado debe permanecer el mayor tiempo posible.

La eficiencia de la convergencia a un estado de carga balanceado dependerá básicamente

³ Especial agradecimiento al Prof. Blai Bonet (Departamento de Computación y Tecnología de la Información, Universidad Simón Bolívar, Caracas, Venezuela) por su valiosa contribución en el desarrollo de este artículo.

de dos factores, la dirección elegida para el envío del flujo de información entre los procesadores y la estructura de datos elegida para el sistema de balanceo de carga (Bronevich y Meyer, 2008). Para abordar el primer factor y decidir, a través de un algoritmo de migración eficiente, el origen y el destino de los trabajos a migrar, se necesita previamente resolver los problemas inherentes al segundo factor, la estructura de datos a través de la cual se decidirá la factibilidad de la migración. Para llevar a cabo el segundo factor en este trabajo, se modelará el clúster como un sistema dinámico que evoluciona a través del tiempo. Tal evolución viene dada por la razón de cambio (medida en unidades de trabajo por unidad de tiempo) de los trabajos que se generan en un nodo y se ejecutan en ese mismo nodo o la fracción de éstos que deban migrar a otro nodo para completar su ejecución y así balancear la carga.

Un modelo matemático de un sistema dinámico se define como un conjunto de ecuaciones que representan la dinámica del sistema con precisión, o al menos, bastante bien. Un modelo matemático no es único para un sistema determinado. Un sistema puede representarse de muchas formas diferentes, por lo que puede tener muchos modelos matemáticos, dependiendo de cada perspectiva. La dinámica de muchos sistemas se describe en términos de ecuaciones diferenciales (Blanchard, Devaney y Hall, 1999). Obtener un modelo matemático razonable es la parte más importante de este análisis.

El estudio de estabilidad para los sistemas lineales y no lineales provee varios métodos de análisis de estabilidad, uno de ellos es el método

de Krasovskii. El método de Krasovskii ofrece condiciones suficientes para los sistemas no lineales y condiciones necesarias y suficientes para los sistemas lineales para que el estado de equilibrio sea asintóticamente estable y difiere de otros métodos porque no está limitado a pequeñas salidas a partir del estado de equilibrio (Ogata, 2009).

Este trabajo propone un modelo matemático de balance de carga dinámico de un clúster de cómputo heterogéneo, basado en ecuaciones diferenciales lineales. Se utiliza el método de Krasovskii, que está basado en la teoría de estabilidad de Lyapunov, para determinar cómo es la estabilidad en el clúster ajustando los parámetros del modelo planteado. A través del ajuste de los parámetros del modelo se puede estudiar la posibilidad de obtener y mantener las condiciones necesarias y suficientes para alcanzar la estabilidad asintótica del sistema y así el balance de carga dinámico en el clúster o por el contrario concluir que el balance es inestable.

La presentación del trabajo se estructura en las siguientes secciones. La primera sección describe los conceptos preliminares de la investigación y enuncia los teoremas de estabilidad a utilizar. La siguiente sección plantea nuestra propuesta de modelo matemático para el balance de carga dinámico de un clúster computacional de n nodos. Luego recreamos un caso de estudio para el modelo planteado considerando un clúster de 3 nodos, y más adelante se describen algunos trabajos relacionados con esta investigación. Por último se muestran algunas conclusiones alcanzadas en esta investigación y los trabajos futuros que desarrollaremos.

Preliminares

El objetivo fundamental de un clúster de computadoras es el de brindar una mayor potencia de cómputo que la lograda por una sola computadora, logrando menores tiempos de respuestas. Otro objetivo es el de permitir el compartimiento de recursos entre los nodos del clúster y el descubrimiento dinámico de los mismos por parte de dichos nodos. El manejo adecuado de los recursos garantiza alta disponibilidad y alto desempeño en el clúster. Un clúster que administra e identifica dinámicamente los recursos disponibles, libera al usuario de gestionar la disponibilidad, los métodos de acceso y cualquier otra política de seguridad y uso de los recursos, y la manera de seleccionar y asignar los recursos más apropiados para cada proceso.

Para lograr una eficiente gestión de los recursos y mejor desempeño en aplicaciones de cómputo científico, una tarea importante es la asignación de carga computacional al conjunto de procesadores. Después de algún tiempo de iniciada alguna tarea de cómputo, la asignación de trabajo a los procesadores puede ocasionar serios desbalances. Para mantener el balance en los siguientes pasos de cómputo, un procedimiento de repartir que hará movimientos periódicos de los datos entre los procesadores para mantener el volumen de comunicación bajo en las futuras iteraciones de las tareas (Catalyurek, Boman, Devine, Bozdog, Heaphy y Riesen, 2009).

El balance de carga entre los nodos de un clúster puede ser modelado a través de un sistema dinámico (Papachristodoulou, Peet, y Niculesco,

2007). En un sistema dinámico determinado, por lo general, el aspecto cuya determinación es más importante es la estabilidad. Si el sistema es lineal e invariante con el tiempo, existen muchos criterios de estabilidad. Entre ellos están el criterio de estabilidad de Nyquist y el criterio de estabilidad de Routh. Sin embargo si el sistema es no lineal o lineal pero variante con el tiempo, tales criterios de estabilidad no aplican. El segundo método de Lyapunov (también conocido como método directo de Lyapunov) es el método más general para la determinación de la estabilidad de los sistemas no lineales o variantes con el tiempo. Por supuesto, este método se aplica a la determinación de la estabilidad de los sistemas lineales e invariantes con el tiempo (Ogata, 2009).

a) Conceptos básicos de migración y balance de carga

El balance de carga estático se caracteriza por un conocimiento previo de la aplicación a ejecutar, las características del sistema y la carga total de trabajo que se ha de repartir entre los procesadores. Es de bastante utilidad en problemas de computación paralela y en técnicas de enrutamiento de paquetes. Una vez que el balance de carga se ha realizado, asignaciones posteriores de carga no son permitidas.

El balance de carga dinámico se puede adaptar a los cambios que se presenten en el sistema, acorde a un protocolo para detectar y enfrentar esos cambios, como por ejemplo la migración de trabajos. El balance de carga dinámico es requerido en una variedad de problemas de las ciencias de la computación, como sistemas de opera-

ción, problemas de optimización combinatoria y en general problemas de paralelismo en los que no se conoce a priori la naturaleza de los trabajos a paralelizar.

Existen dos esquemas bases en el balance de carga dinámico: orientado al servidor (también conocido como *work-sharing*) y orientado al receptor (*work-stealing*). El primero de ellos entrega las decisiones de migración al nodo que se encuentra sobrecargado de trabajo, y su objetivo es que todos los nodos conectados tengan la misma cantidad de trabajo. El segundo entrega las decisiones de migración al nodo que no posee carga de trabajo, y su objetivo es aprovechar los ciclos de CPU ociosos para realizar un trabajo en paralelo. El primero de estos esquemas ha sido fuertemente estudiado en la literatura, pero en la práctica su eficiencia decae al agregar más trabajo a un nodo saturado (Bustos, 2006). Por otro lado, según Berenbrink Friedetzky y Goldberg, (2001) “a pesar de su utilidad en la práctica, no existen resultados teóricos conocidos sobre *work-stealing*”. Más aún, todos los estudios teóricos de balance de carga dinámico se fundamentan sobre la migración fuerte de trabajos, en que los trabajos migran de un nodo a otro al momento de recibir el requerimiento de migración y no sobre la migración débil, utilizada en la práctica por su menor costo y en que los trabajos deben esperar por un estado estable en su ejecución antes de realizar la migración (Bustos, 2006).

b) Método directo de Lyapunov

A partir de la teoría clásica de la mecánica, sabemos que un sistema vibratorio es estable si su energía total (una función definida positiva) es

continuamente decreciente (lo que significa que la derivada de tiempo de la energía total debe ser negativa) hasta que alcanza un estado de equilibrio.

El método directo o segundo método de Lyapunov se basa en una generalización de este hecho: si el sistema tiene un estado de equilibrio asintóticamente estable, la energía almacenada en el sistema desplazada dentro del dominio de atracción se descompone conforme transcurre el tiempo, hasta que finalmente adopta su valor mínimo en el estado de equilibrio. Sin embargo, para los sistemas puramente matemáticos, no existe una forma simple de definir una “función de energía”. Para superar esta dificultad, Lyapunov introdujo la función de Lyapunov, una función de energía ficticia. No obstante, esta idea es más general que la de la energía y tiene más posibilidades de aplicación. De hecho, cualquier función escalar que satisfaga la hipótesis de los teoremas de estabilidad de Lyapunov sirve como función de Lyapunov. Para los sistemas simples (por ejemplo, sistemas lineales), podemos vislumbrar las funciones de Lyapunov convenientes; pero, para un sistema complicado (por ejemplo, un sistema no lineal), resulta bastante difícil encontrar una función de Lyapunov (Papachristodoulou et al., 2007).

Las funciones de Lyapunov dependen de x_1, x_2, \dots, x_n y t . Las representaremos mediante $V(x_1, x_2, \dots, x_n, t)$ o simplemente mediante $V(x, t)$. Si las funciones de Lyapunov no incluyen t explícitamente, las representamos mediante $V(x_1, x_2, \dots, x_n)$, o $V(x)$. En el segundo método de Lyapunov, el comportamiento del signo de $V(x, t)$ y el de su derivada con respecto al tiempo $V(x, t) = dV(x, t)/dt$ nos ofrece información con respecto a la esta-

bilidad, la estabilidad asintótica, o la inestabilidad de un estado de equilibrio sin que nos obliguen a que obtengamos directamente la solución. Esto se aplica tanto a los sistemas lineales como a los no lineales.

Se puede demostrar que si una función escalar $V(x)$, en donde x es un vector de dimensión n , es definida positiva, los estados de x que satisfacen $V(x) = C$, en donde C es una constante positiva, se encuentran en una hipersuperficie cerrada en el espacio de estados de n dimensiones, al menos en la cercanía del origen. Si $V(x) \rightarrow \infty$ conforme $\|x\| \rightarrow \infty$, entonces, las superficies cerradas se extienden sobre el espacio de estados completo. La hipersuperficie $V(x) = C_1$, se encuentra completamente dentro de la hipersuperficie $V(x) = C_2$, si $C_1 < C_2$ (Ogata, 2009).

Para un sistema determinado, si se encuentra una función escalar definida positiva $V(x)$ tal que su derivada con respecto al tiempo, tomada a lo largo de una trayectoria, sea siempre negativa, entonces, conforme se incrementa el tiempo, $V(x)$ adopta valores cada vez más de C . Conforme se incrementa el tiempo, $V(x)$ finalmente se reduce a cero y, por tanto, x también se reduce a cero. Esto implica la estabilidad asintótica del origen del espacio de estados. El teorema principal de Lyapunov, que es una generalización del hecho anterior, proporciona una condición suficiente para la estabilidad asintótica. Este teorema se enuncia a continuación.

c) Teorema principal de Lyapunov

Suponga un sistema que se describe mediante $\dot{x} = f(x, t)$ en donde $f(0, t) = 0$, para toda t .

Si existe una función escalar $V(x, t)$ con primeras derivadas parciales continuas, que satisface las condiciones:

1. $V(x, t)$ es definida positiva
2. $\dot{V}(x, t)$ es definida negativa

entonces el estado de equilibrio en el origen es uniforme y asintóticamente estable. Si, además, $V(x, t) \rightarrow \infty$ conforme $\|x\| \rightarrow \infty$, entonces el estado de equilibrio en el origen es uniforme y asintóticamente estable en general.

d) Método de Krasovskii

El estudio de la estabilidad de los sistemas lineales y no lineales provee varios métodos. El teorema o método de Krasovskii para la estabilidad asintótica en general ofrece condiciones suficientes para los sistemas no lineales y condiciones necesarias y suficientes para los sistemas lineales y difiere de otros métodos porque no está limitado a pequeñas salidas a partir del estado de equilibrio (Ogata, 2009). El teorema o método de Krasovskii es una aplicación de la teoría de estabilidad de Lyapunov (Papachristodoulou et al., 2007), y en particular del método directo de Lyapunov, que a su vez es el método más general para la determinación de la estabilidad de los sistemas lineales y no lineales variantes o no en el tiempo.

Un estado de equilibrio de un sistema no lineal puede ser estable incluso si no se satisfacen las condiciones especificadas en este teorema. Por tanto, al usar este teorema se debe tener cuidado de no llegar a conclusiones erróneas con respecto a la estabilidad de un sistema no lineal determinado.

e) Teorema de Krasovskii

Suponga que $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ y que $\mathbf{f}(\mathbf{x})$ es diferenciable con respecto a x_i , $i = 1, 2, \dots, n$. Se define la matriz jacobiana para el sistema del modo siguiente:

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \frac{\partial(f_1, f_2, \dots, f_n)}{\partial(x_1, x_2, \dots, x_n)} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Así mismo se define:

$$\hat{\mathbf{F}}(\mathbf{x}) = \mathbf{F}^*(\mathbf{x}) + \mathbf{F}(\mathbf{x})$$

En donde $\mathbf{F}^*(\mathbf{x})$ es la transpuesta conjugada de $\mathbf{F}(\mathbf{x})$. Si $\mathbf{f}(\mathbf{x})$ es real, entonces $\mathbf{F}(\mathbf{x})$ es real y $\mathbf{F}^*(\mathbf{x})$ se escribe como $\mathbf{F}'(\mathbf{x})$. Es evidente que $\hat{\mathbf{F}}(\mathbf{x})$ es claramente hermitiana (simétrica si $\mathbf{F}(\mathbf{x})$ es real). Si $\hat{\mathbf{F}}(\mathbf{x})$ es definida negativa, el estado de equilibrio es asintóticamente estable. Una función de Lyapunov (Li y Gu, 2010; Nikraves y Vali, 2007) para este sistema es:

$$V(\mathbf{x}) = \mathbf{f}^*(\mathbf{x}) \mathbf{f}(\mathbf{x})$$

Además, si $\mathbf{f}^*(\mathbf{x}) \mathbf{f}(\mathbf{x}) \rightarrow \infty$ conforme $\|\mathbf{x}\| \rightarrow \infty$, entonces el estado de equilibrio es asintóticamente estable en general. Para saber si $\hat{\mathbf{F}}(\mathbf{x})$ es definida negativa, aplicamos el criterio de Sylvester (Ikramov y Savel'eva, 1998) para definidad negativa sobre todas las menores principales de $\hat{\mathbf{F}}(\mathbf{x})$, esto es, sea $\hat{\mathbf{F}}(\mathbf{x})$ una forma cuadrática en-

tonces $\hat{\mathbf{F}}(\mathbf{x})$ es definida negativa $\Leftrightarrow (-1)^k$ determinante $(\hat{\mathbf{F}}(\mathbf{x}))_k > 0, k = 1, 2, \dots, n$. Luego al aplicar el criterio de Sylvester a todas las menores principales de $\hat{\mathbf{F}}(\mathbf{x})$ se obtiene un sistema expresado en forma de matrices de desigualdades (Seuret, 2009; Ariba y Gouaisbaut, 2008; Fridman y Niculescu, 2007).

Modelo de balance de carga dinámico para clústeres computacionales: la propuesta

Para nuestro modelo matemático, se considera un clúster de n nodos, donde:

- $n > 1$.
- El balance de carga que regirá este modelo es el balance de carga dinámico centralizado, es decir, se seguirá una estructura típica maestro/esclavo o de programación por demanda, en la que el nodo maestro tiene la colección completa de la carga a ejecutar. La carga de trabajos es repartida entre los nodos esclavos y cuando alguno de éstos finaliza la ejecución de su carga, solicita una nueva carga al nodo maestro. El nodo maestro no se involucra en la ejecución de trabajos, sólo distribuye la carga que a través de él ingresa al clúster, de allí que el valor mínimo de nodos en el clúster es 2. Este enfoque es ideal para clústeres de alto desempeño.

Un modelo matemático para un clúster de cómputo, basado en la ley de equilibrio o tasa de cambio para problemas de flujo, se puede definir considerando la siguiente hipótesis:

- Sea $Carga_i(t)$ la cantidad de trabajos a ejecutar en cualquier momento t en el nodo i , la variación del volumen de trabajos a ejecutar o tasa de

cambio en el nodo i viene dada por la diferencia de las cargas de entrada y salida expresada en

$$\frac{dCarga_i}{dt} = \text{Carga de entrada}_{\text{Nodo } i} - \text{Carga de salida}_{\text{Nodo } i}$$

La cantidad de trabajo de un nodo en cualquier instante de tiempo es la unidad fundamental de este modelo y que a su vez representa la carga del nodo. La naturaleza de las aplicaciones que generan carga no está contemplada en este

cantidad de trabajos por unidad de tiempo, para $1 \leq i \leq n$, esto es:

ejemplo por tratarse de un enfoque a bajo nivel. Una ecuación algebraica para este modelo está dada por la carga en unidades de trabajo para un nodo en cualquier instante de tiempo [3], para $1 \leq i, j \leq n, i \neq j$, es decir:

$$Carga_i^{(t+1)} = Carga_i^{(t)} + CargaObtenida_de_OtrosNodos_{j \rightarrow i}^{(t)} - CargaMigrada_a_OtrosNodos_{i \rightarrow j}^{(t)} - CargaEjecutada_en_el_Nodo_i^{(t)}$$

donde:

$Carga_i^{(t+1)}$: es la carga del nodo i en el instante $t+1$.

$Carga_i^{(t)}$: representa la carga del nodo i en el instante t .

$CargaObtenida_de_OtrosNodos_{j \rightarrow i}^{(t)}$: es la carga ganada por el nodo i proveniente de otros nodos en el instante t .

$CargaMigrada_a_OtrosNodos_{i \rightarrow j}^{(t)}$: representa la carga migrada del nodo i a otros nodos en el instante t .

$CargaEjecutada_en_el_Nodo_i^{(t)}$: es la carga de trabajo finalizada en el nodo i en el instante t .

Para este modelo se usarán las siguientes variables y parámetros para $1 \leq i \leq n$:

- t = Tiempo. Es la variable independiente.

- $Carga_i$ = Carga en unidades de trabajo del nodo i generada desde el mismo nodo o proveniente de otros nodos en cualquier instante de tiempo t . Representa la variable dependiente.

- VC_i = Velocidad promedio de carga en el nodo i expresada en Mb/seg (sólo para el nodo maestro, para los restantes nodos es igual a cero). Es un parámetro.

- TTC_i = Tamaño promedio de los trabajos expresado en Mb que se cargan en el nodo i (sólo para el nodo maestro, para los restantes nodos es igual a cero). Representa un parámetro.

- VM_i = Velocidad promedio de migración desde el nodo i expresada en Mb/seg. Es un parámetro.

- TM_i = Tamaño del espacio de memoria para trabajos del nodo i expresada en Mb. Representa un parámetro.

- VE_i = Velocidad promedio de ejecución de trabajos en el nodo i expresada en Mb/seg. Es un parámetro.
- $Carga_a_migrar_i$ = Carga a migrar del nodo i a otros nodos. La carga migrada a un nodo será aceptada si su carga actual sumada a la que recibirá por migración es menor que su capacidad de memoria. Representa un parámetro.

Luego a partir de las ecuaciones (1) y (2) se puede expresar la hipótesis del modelo a través de la siguiente ecuación diferencial lineal, donde sólo uno de los nodos es el nodo maestro y para los otros $n - 1$ nodos restantes $VC = 0$ y $TTC = 0$ (el clúster carga trabajos sólo a través del nodo maestro). De igual forma $VE = 0$ para el nodo maestro, por su condición de no ejecutar y sólo distribuir carga de trabajo entre los nodos esclavos. Luego para $1 \leq i, j \leq n, i \neq j$, tenemos:

$$\frac{dCarga_i}{dt} = VC_i * TTC_i - \frac{(n-1)VM_i}{TM_i} Carga_i - \frac{VE_i}{TM_i} Carga_i + \sum_{j \neq i}^n \frac{VM_j}{TM_j} Carga_j \quad (3)$$

$$Carga_a_migrar_i = Carga_i - TM_i \text{ si } Carga_i \geq TM_i$$

$$Carga_a_migrar_i = 0 \text{ en cualquier otro caso}$$

Donde:

$VC_i * TTC_i$: representa la cantidad de trabajos por segundo que están ingresando al clúster a través del maestro $\left(VC_i * TTC_i = \frac{Mb}{seg} * \frac{Trabajos}{Mb} = \frac{Trabajos}{seg} \right)$.

$\frac{(n-1)VM_i}{TM_i} Carga_i$: es la cantidad de trabajos por segundo que será migrada desde el nodo i a los $n-1$ nodos restantes $\left(\frac{(n-1)VM_i}{TM_i} Carga_i = \frac{\frac{Mb}{Seg}}{Mb} * Trabajos = \frac{Trabajos}{seg} \right)$.

$\frac{VE_i}{TM_i} Carga_i$: representa la cantidad de trabajos por segundo ejecutados o finalizados en el nodo i

$$\left(\frac{VE_i}{TM_i} Carga_i = \frac{\frac{Mb}{Seg}}{Mb} * Trabajos = \frac{Trabajos}{seg} \right)$$

$\sum_{j=1}^{i-1} \frac{VM_j}{TM_j} Carga_j + \sum_{j=i+1}^n \frac{VM_j}{TM_j}$: es la carga de trabajos por segundo que será obtenida por el nodo i proveniente de los restantes nodos $\left(\frac{VM_j}{TM_j} Carga_j = \frac{\frac{Mb}{Seg}}{Mb} * Trabajos = \frac{Trabajos}{seg} \right)$.

Los parámetros son cantidades que no cambian con el tiempo (o con la variable independiente) pero pueden ajustarse por causas naturales o a través de experimentos. El aspecto más importante de este estudio consiste en determinar la manera cómo cambian las variables dependientes cuando se ajustan los parámetros. Un buen ajuste de parámetros garantizará que el modelo representa fielmente la característica real que se quiere modelar del clúster, el balance de carga dinámico. Luego, si los datos reales concuerdan con el modelo, se tendrá confianza en que la hipótesis hecha al crear el modelo es razonable y éste puede usarse para hacer predicciones; si no concuerdan, entonces debemos estudiar y mejorar nuestras suposiciones. En todo caso, aprendemos más acerca del sistema al compararlo con el modelo.

Con este modelo, se prueba si hay condiciones para una estabilidad asintótica del sistema. Esto es, se prueba si hay condiciones necesarias y suficientes para garantizar el balance de carga dinámico en el clúster, con los parámetros ya establecidos. La idea es emplear algún método, basado en el segundo método de Lyapunov (también conocido como método directo de Lyapunov), para examinar la estabilidad del sistema. Un ejemplo de este tipo de métodos es el método de Krasovskii. Si con estos parámetros se cumple con el teorema de Krasovskii, entonces el estado de equilibrio es asintóticamente estable en general, de lo contrario se debe volver a iterar intentando un nuevo ajuste de parámetros. El resultado obtenido es de vital importancia a la hora de decidir una migración a través de un algoritmo de migración, es decir, si

se puede concluir que el equilibrio en el clúster de cómputo es asintóticamente estable, tendremos certeza de que el funcionamiento del clúster es adecuado para realizar tareas de migración en las que no se requiera conocer previamente la estabilidad del sistema.

La siguiente sección muestra a través de un ejemplo sencillo de un clúster de 3 nodos, cómo se usa el modelo propuesto para determinar la estabilidad asintótica del sistema. Esto es, probar si hay condiciones necesarias y suficientes para garantizar el balance de carga dinámico en el clúster, con los parámetros establecidos.

Caso de estudio: ejemplo de uso del modelo propuesto

Esta sección ilustra la aplicabilidad del modelo con un caso de estudio simple que consta de un clúster computacional de 3 nodos, usando datos sintéticos para instanciar las variables y parámetros del modelo. Estos valores son los mostrados en la **Tabla 1**, y representan el primer ajuste de valores a parámetros. Si el modelo concluye que el clúster es inestable en sentido de Liapunov-Krasovskii, se pueden intentar nuevos ajustes de parámetros al modelo hasta concluir con la estabilidad o inestabilidad definitiva del sistema.

a) Escenario

Se modela un clúster computacional con $n = 3$ nodos, el nodo 1 representando al nodo maestro y cuyos valores de variables y parámetros se muestran en la **Tabla 1**.

La **Figura 1** muestra gráficamente el esquema del clúster.

A partir de la ecuación general (3) y los valores mostrados en la **Tabla 1**, se puede expresar el modelo a través del siguiente sistema de ecuaciones diferenciales representado por (4), (5) y (6). Se usó GNU Octave-3.2.2 para recrear el modelo.

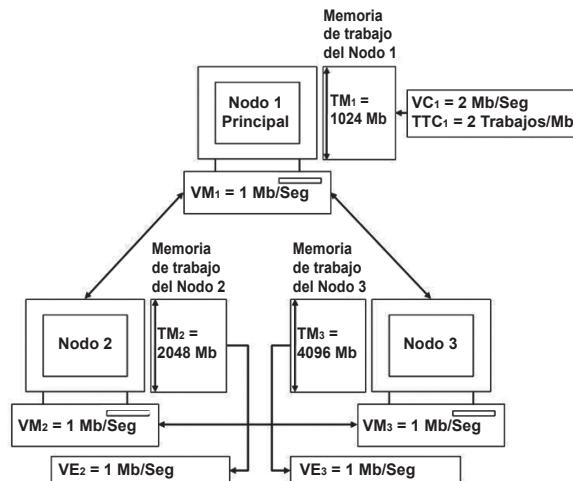
Se integra para un intervalo de 6 horas, es decir, desde $t = 0$ hasta $t = 21600$ segundos te-

$$\frac{dCarga_1}{dt} = 4 - \frac{2Carga_1}{1024} + \frac{Carga_2}{2048} + \frac{Carga_3}{4096} \quad (4)$$

$$\frac{dCarga_2}{dt} = -\frac{3Carga_2}{2048} + \frac{Carga_1}{1024} + \frac{Carga_3}{4096} \quad (5)$$

$$\frac{dCarga_3}{dt} = -\frac{3Carga_3}{4096} + \frac{Carga_1}{1024} + \frac{Carga_2}{2048} \quad (6)$$

Figura 1
Esquema del clúster



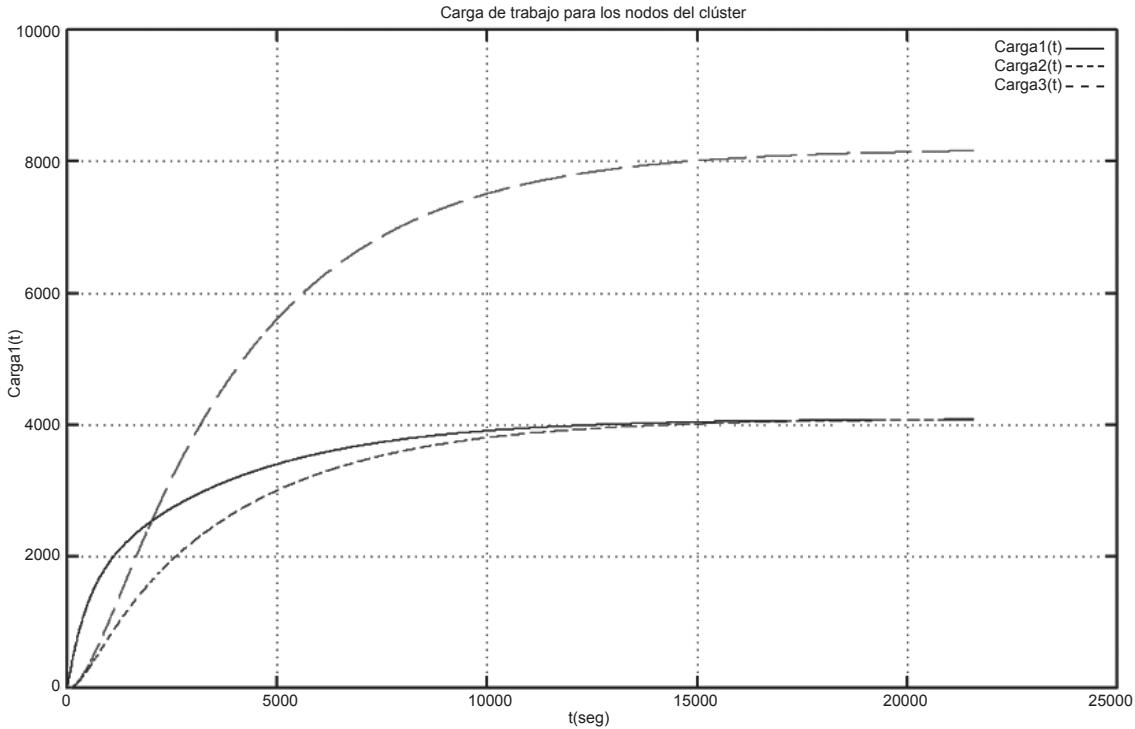
niendo como condiciones iniciales, $Carga_1(0) = 1$, $Carga_2(0) = 1$ y $Carga_3(0) = 1$. Observe la gráfica resultante en la **Figura 2**.

Para esta solución, la carga de trabajo de cada nodo tiende a estabilizarse transcurridos los primeros 15000 segundos y pareciera permanecer estable cuando $t \rightarrow \infty$. Si esto es cierto, para este ajuste de parámetros, cada nodo tiende a su punto de equilibrio. Observe que la curva de $Carga_3(t)$ crece más que las otras dos, esto es debido a que el nodo 3 tiene mayor capacidad de memoria para almacenar trabajos, como lo muestra el parámetro en la **Tabla 1**.

Tabla 1
Valores iniciales de parámetros y variables

| | |
|-------------------------------|---------------------------------|
| Nodo 1 (Maestro) | Valor |
| VC_1 (Promedio) | 2 Mb/seg |
| TTC_1 (Promedio) | 2 Trabajos/Mb |
| VM_1 (Promedio) | 1 Mb/seg |
| TM_1 | 1024 Mb |
| VE_1 | 0 |
| $Carga_1(t=0)$ | 1 Trabajo |
| Nodo 2 (Esclavo) | |
| VC_2 | 0 |
| TTC_2 | 0 |
| VM_2 (Promedio) | 1 Mb/seg |
| TM_2 | 2048 Mb |
| VE_2 (Promedio) | 1 Mb/seg |
| $Carga_2(t=0)$ | 1 Trabajo |
| Nodo 3 (Esclavo) | |
| VC_3 | 0 |
| TTC_3 | 0 |
| VM_3 (Promedio) | 1 Mb/seg |
| TM_3 | 4096 Mb |
| VE_3 (Promedio) | 1 Mb/seg |
| $Carga_3(t=0)$ | 1 Trabajo |
| Intervalo de experimentación: | Desde $t=0$ Hasta $t=21600$ seg |
| 6 horas | |

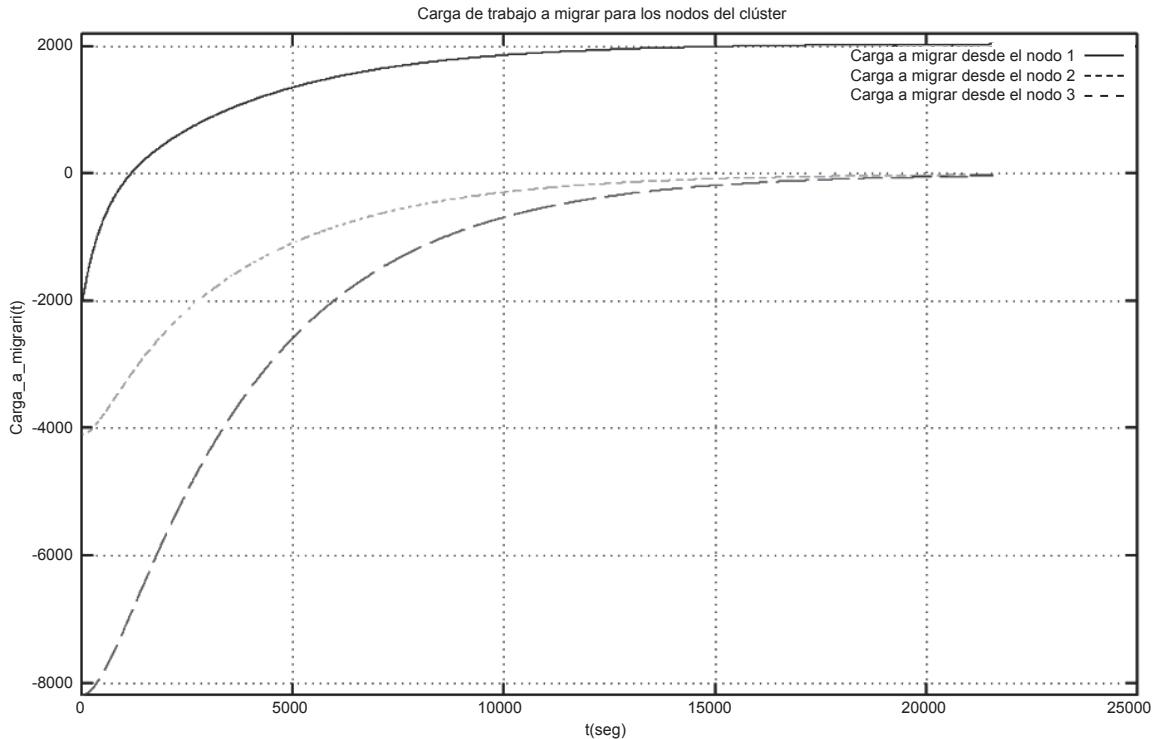
Figura 2
Solución gráfica del modelo para $0 \leq t \leq 21600$ seg (6 horas)



También se aprecia que $Carga_1(t)$ se incrementa al inicio más rápidamente que en los otros nodos dado que es a través del nodo 1 que el clúster adquiere los trabajos a ejecutar (nodo 1 es el nodo maestro); así puede observarse que esta supera los 2048 trabajos (la capacidad máxima de memoria para trabajos del nodo 1 es de 1024 Mb,

los trabajos se almacenan a razón de 2 trabajos/Mb de acuerdo a los parámetros mostrados en la **Tabla 1**) rebasando su capacidad de memoria y generando rápidamente un desbalance en el clúster. En **Figura 3** se muestra la necesidad de migración de los nodos para balancear la carga del clúster.

Figura 3
Necesidad de migración de cada nodo del clúster



Adicionalmente, puede observarse en la **Figura 3** que los nodos 2 y 3 no requieren migrar carga, sin embargo, el nodo 1 requiere migrar al menos 1024 Mb de carga (se debe recordar que el eje y de la gráfica representa la carga en trabajos, y que el tamaño promedio de un trabajo es 512 Mb, es decir, 2 trabajos/Mb) que representa justo el exceso de carga observado en la Figura 2, pero si el nodo 1 migra su exceso, el clúster aún per-

manecerá desbalanceado porque los nodos 2 y 3 tienen su capacidad de memoria al máximo como se aprecia en la Figura 2. Esta situación amerita un nuevo ajuste de parámetros en el modelo del clúster. Para esta segunda iteración se disminuirá la velocidad de carga de trabajos en el nodo maestro (VC_m) de 2Mb/seg a 1Mb/Seg, luego se efectúa el cambio en el sistema de ecuaciones y se genera una nueva solución. Las soluciones gráficas son

mostradas en las **Figuras 4 y 5**. Se observa ahora que disminuyendo a la mitad VC_1 , la carga del nodo 1 no excede su capacidad de memoria y la carga de los nodos 2 y 3 sólo ocupa la mitad de la capacidad de memoria para ambos nodos. Si este comportamiento se mantiene así a través del tiempo, el clúster puede trabajar balanceadamente. Para validar si la carga en los nodos del clúster tiende a su punto de equilibrio, se usará el método de Krasovskii. Si se concluye que el sistema de ecuaciones que describe el clúster es asintóticamente estable en general, entonces la carga en los nodos del clúster si tiende a su punto de equilibrio y para cualquier otra necesidad de migración por parte de los nodos esclavos, está podrá llevarse a cabo sin requerirse previamente la comprobación de la estabilidad del sistema.

Seguidamente partiendo del sistema de ecuaciones diferenciales y aplicando el teorema o método de Krasovskii se buscan las condiciones necesarias y suficientes para una estabilidad asintótica del sistema, esto es, probar si hay condiciones necesarias y suficientes para garantizar el balance de carga en el clúster con los parámetros ya establecidos en cualquier instante de tiempo t , para ello debe reescribirse el sistema de ecuaciones diferenciales (4), (5) y (6) de la forma siguiente:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

en donde \mathbf{x} es un vector de dimensión n (vector de estado) y $\mathbf{f}(\mathbf{x})$ es un vector de dimensión n cuyos elementos son funciones lineales de x_1, x_2, \dots, x_n . Para este caso, $[x_1, x_2, x_3] = [\text{Carga}_1, \text{Carga}_2, \text{Carga}_3]$ y $\dot{\mathbf{x}} = [\dot{x}_1, \dot{x}_2, \dot{x}_3] = \left[\frac{d\text{Carga}_1}{dt}, \frac{d\text{Carga}_2}{dt}, \frac{d\text{Carga}_3}{dt} \right]$. Se

asume además que $\mathbf{f}^*(\mathbf{x})\mathbf{f}(\mathbf{x}) \rightarrow \infty$ conforme $\|\mathbf{x}\| \rightarrow \infty$. Luego nuestro sistema con el cambio efectuado a VC_1 , nos queda:

$$\dot{x}_1 = 2 - \frac{2x_1}{1024} + \frac{x_2}{2048} + \frac{x_3}{4096} \quad (7)$$

$$\dot{x}_2 = \frac{x_1}{1024} - \frac{3x_2}{2048} + \frac{x_3}{4096} \quad (8)$$

$$\dot{x}_3 = \frac{x_1}{1024} + \frac{x_2}{2048} - \frac{3x_3}{4096} \quad (9)$$

A partir de las ecuaciones (7), (8) y (9) y aplicando el teorema de Krasovskii descrito anteriormente se obtiene:

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} -2/1024 & 1/2048 & 1/4096 \\ 1/1024 & -3/2048 & 1/4096 \\ 1/1024 & 1/2048 & -3/4096 \end{bmatrix}$$

$$\mathbf{F}'(\mathbf{x}) = \begin{bmatrix} -2/1024 & 1/1024 & 1/1024 \\ 1/2048 & -3/2048 & 1/2048 \\ 1/4096 & 1/4096 & -3/4096 \end{bmatrix}$$

$$\hat{\mathbf{F}}(\mathbf{x}) = \begin{bmatrix} -4/1024 & 3/2048 & 5/4096 \\ 3/2048 & -6/2048 & 3/4096 \\ 5/4096 & 3/4096 & -6/4096 \end{bmatrix}$$

Aplicamos el criterio de Sylvester a todas las menores principales de $\hat{\mathbf{F}}(\mathbf{x})$:

$$-\frac{4}{1024} < 0$$

$$\begin{vmatrix} -4/1024 & 3/2048 \\ 3/2048 & -6/2048 \end{vmatrix} > 0$$

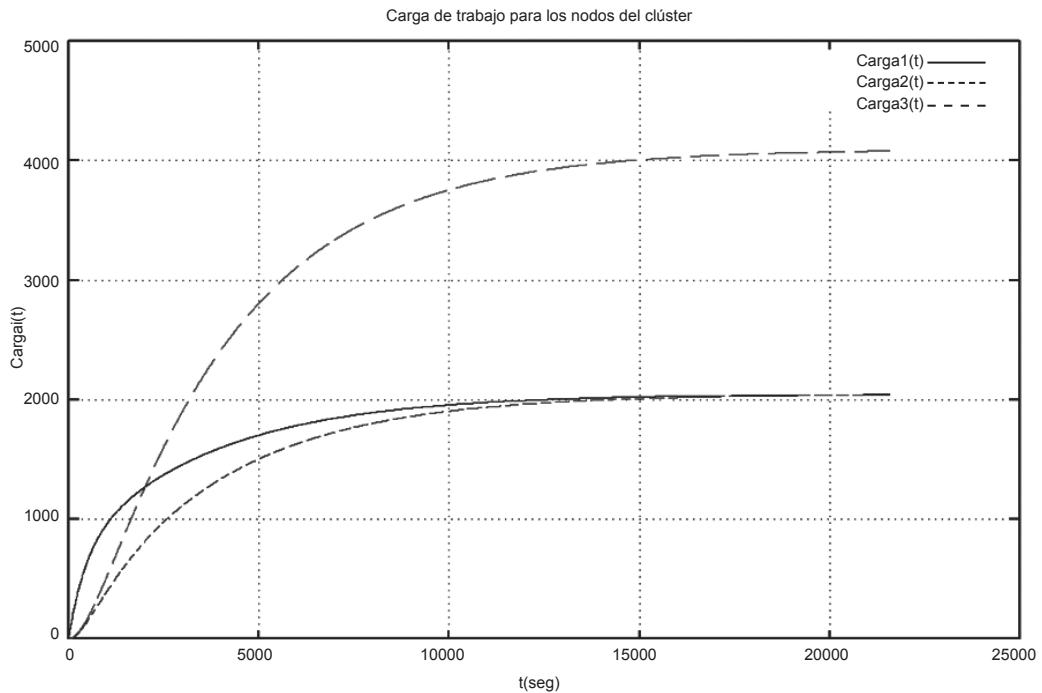
$$\begin{vmatrix} -4/1024 & 3/2048 & 5/4096 \\ 3/2048 & -6/2048 & 3/4096 \\ 5/4096 & 3/4096 & -6/4096 \end{vmatrix} < 0$$

Para cada menor principal se cumple el criterio de Sylvester para definidad negativa sobre $\hat{F}(\mathbf{x})$. Entonces si $\hat{F}(\mathbf{x})$ es definida negativa, el estado de equilibrio es asintóticamente estable en general. Estas conclusiones justifican el comportamiento observado en las **Figuras 4 y 5**. Luego, se puede concluir que los tres nodos trabajarán balanceadamente para el modelo planteado.

Trabajos relacionados

La teoría de estabilidad de Lyapunov y los métodos que son aplicaciones de ésta, han ganado cada vez más terreno en el campo del balance de carga dinámico asociado a sistemas de múltiples computadoras. En la literatura se han desarrollado muchos algoritmos de protocolo de consenso para redes a lo largo de los últimos años. La ro-

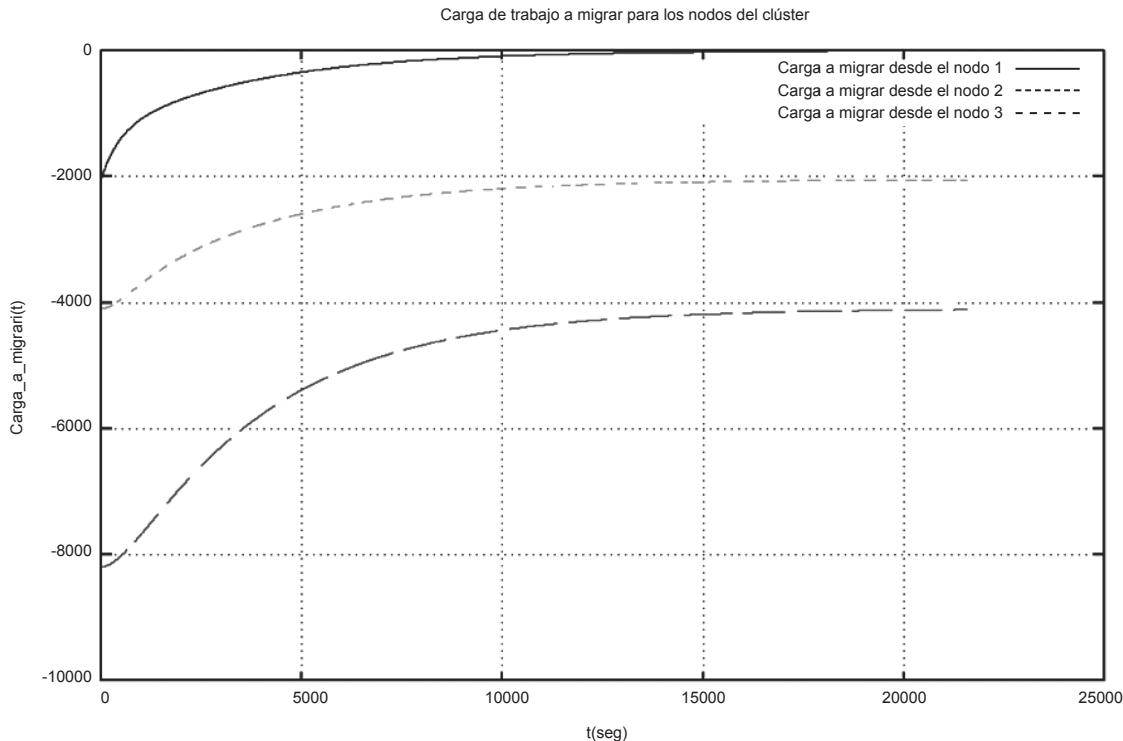
Figura 4
Solución gráfica del modelo con $VC_1 = 1\text{Mb/seg}$ para $0 \leq t \leq 21600$ seg (6 horas)



bustez de esos algoritmos que implica la dinámica no lineal, ha sido ignorada (Hui y Haddad, 2009). Por robustez se entiende la capacidad del algoritmo de alcanzar la semiestabilidad bajo un modelo con incertidumbre. En este trabajo se examina la robustez de algunos algoritmos de protocolo de consenso y se desarrollan condiciones suficientes para la estabilidad utilizando el método directo de

Lyapunov. La semi estabilidad es la propiedad de que cada trayectoria del sistema que se inicia en un punto de equilibrio estable de Lyapunov converge a un posiblemente diferente punto de equilibrio estable de Lyapunov (Hui y Haddad, 2009). Por tanto, esta semi estabilidad implica estabilidad de Lyapunov, y es lo que significa estabilidad asintótica.

Figura 5
Necesidad de migración de cada nodo del clúster con VC1 = 1Mb/seg



En (Barth y Bournez, 2009) se expone un enfoque dinámico de balance de carga basado en teoría de juegos, particularmente en el equilibrio de Nash. Aquí se plantea un escenario donde las tareas o trabajos son jugadores cuyo objetivo es minimizar su propio costo. El costo de una máquina es una función de su propia carga. La carga de una máquina es la suma de los pesos de los trabajos que estén ejecutando en ella. El objetivo del trabajo es probar que a partir de información incompleta sobre el juego se puede aprender y probar que se consigue un equilibrio de Nash a través de algunas funciones de Lyapunov. En (Seuret, 2009) se propone un método basado en funciones de Lyapunov-Krasovskii para el análisis de estabilidad en sistemas lineales utilizando parámetros polinomiales. Las condiciones de estabilidad son derivadas de una matriz lineal de inecuaciones.

En (Papachristodoulou et al., 2007) abordan el problema del análisis de estabilidad para los sistemas lineales con retardos variables en el tiempo. La estabilidad de este tipo de sistemas es de gran interés en redes de comunicación como también en otras áreas. Ellos desarrollaron condiciones suficientes para la estabilidad robusta usando descomposición de sumas de cuadrados en polinomios multivariados y el teorema Lyapunov-Krasovskii.

En (Farias, Moallemi y Prabhakar, 2005) se considera un modelo donde los trabajos se distribuyen a procesadores paralelos siguiendo una distribución de Poisson y cadenas de Markov. La idea es que un trabajo que llega a un procesador pueda ser ejecutado o migrado a otro procesador, en este caso este crea $K \geq 1$ trabajos independien-

tes. Cuando $K = 1$ no se impone ningún costo extra por trabajos de migración. El interés de este trabajo está dado para casos en donde $K > 1$, porque tiene que decidirse si se migra o no. Por un lado la migración persigue balancear la carga, pero por otro lado, la migración en sí es un trabajo extra que hará disminuir el rendimiento del procesador. En el trabajo de Farias et al. se plantean la interrogante: ¿Existe migración débil que contribuya a balancear la carga mientras provee alto desempeño? Ellos concluyen que migrar al procesador menos cargado hace inestable al sistema y encuentran que una leve variación de esta regla lo hace estable. La estabilidad es estudiada en el sentido de Lyapunov.

En este trabajo se desarrolla un modelo matemático basado en ecuaciones diferenciales que describe el comportamiento de un clúster de cómputo. De forma análoga a los trabajos anteriores, se usa la teoría de Estabilidad de Liapunov y en particular el método de Krasovskii para analizar la estabilidad en el flujo de carga de trabajo entre los nodos. Si la carga de los nodos tiende a sus puntos de equilibrio y no se aleja de ellos, puede llegar a concluirse que el comportamiento del clúster es asintóticamente estable, de allí que, las próximas tareas de migración no requerirán de la comprobación previa de la estabilidad de los nodos para que éstas puedan ejecutarse.

Conclusiones y trabajos futuros

Para el conjunto de ecuaciones diferenciales que describen el funcionamiento del clúster de cómputo junto con los valores dados a los pa-

rámetros, el clúster mantendrá su estado balanceado (la trayectoria del clúster, que partió de las condiciones iniciales tiende a su estado de equilibrio) siempre que se cumplan las condiciones de las desigualdades (criterio de Sylvester a todas las menores principales de $\hat{F}(\mathbf{x})$) que son necesarias y suficientes para una estabilidad asintótica. Luego, al tener la certeza de que el clúster está asintóticamente estable, es decir, que alcanzó su estado de equilibrio, se pueden realizar tareas de migración sin necesidad de comprobar la estabilidad de los nodos antes de tales migraciones, esto indudablemente se traduce en un incremento en el desempeño del clúster computacional. Este trabajo abarcó sólo algunos parámetros relacionados con la asignación, ejecución y migración de tareas en un clúster computacional de tres nodos, estos parámetros se ajustan sólo al inicio de cada experimento. Si no se producen los resultados esperados, debe repetirse el proceso.

Como trabajo futuro inmediato se desarrollará una aplicación para ajustar dinámicamente los parámetros al modelo, de esta forma se conocerá de forma inmediata la configuración que debe tener el clúster para garantizar su estado balanceado, todo esto aplicado a un enfoque de migración débil.

Bibliografía

- Ariba, Y. y Gouaisbaut, F. (2008). Construction of Lyapunov-Krasovskii functional for time-varying delay systems. Manuscript, published in *47th IEEE Conference on Decision and Control*, Cancun: Mexico. DOI: 10.1109/CDC.2008.4738942.
- Barth, D. y Bournez, O. (2009). A Dynamic Approach for Load Balancing. Inria-00435160, version 1. Manuscript, published in *The Third International Workshop on Game Theory in Communication Networks – GAMECOMM 2009*.
- Berenbrink, P., Friedetzky, T. y Goldberg, L. (2001). The natural work-stealing algorithm is stable. En *IEEE Symposium on Foundations of Computer Science*, pp. 178-187.
- Blanchard, P., Devaney, R. y Hall, C. (1999). *Differential Equations*. International Thomson Publishing Company. Boston University.
- Bronevich, A. y Meyer, W. (2008). Load balancing algorithms based on gradient methods and their analysis through algebraic graph theory. *Journal of Parallel and Distributed Computing*. Issue 2, 68 (2008), pp. 209-220.
- Bustos, J. (2006). Balance de Carga Dinámico para Objetos Activos Móviles. *Proyecto de Investigación del grupo OASIS*, perteneciente a INRIA Sophia-Antipolis, Francia.
- Catalyurek, U., Boman, E., Devine, K., Bozdog, D., Heaphy, R. y Riesen, L. (2009). A repartitioning hypergraph model for dynamic load balancing. *Journal of Parallel and Distributed Computing*, 69, pp. 711-724.
- Dongarra, J. (2011). *Japan Reclaims Top Ranking on Latest TOP500 List of World's Supercomputers*. Más reciente consulta: 17 de julio 2011. Disponible en el sitio: Top500 Supercomputer Site. <http://www.top500.org/>
- Farias, V., Moallemi, C., y Prabhakar, B. (2005). *Load Balancing with Migration Penalties*. Electrical Engineering, Computer Science, Stanford University, Stanford, CA 94305, USA.

- Fridman, E. y Niculescu, S. (2007). On complete Lyapunov-Krasovskii functional techniques for uncertain systems with fast-varying delays. *Int. J. Robust Nonlinear Control* (en prensa). *InterScience* (www.interscience.wiley.com). Más reciente consulta: 12 de abril de 2011. DOI: 10.1002/rnc.1230.
- Hui, Q. y Haddad, W. (2009). On robust control algorithms for nonlinear network consensus protocols. Department of Mechanical Engineering, Texas Tech University, Lubbock, TX 79409-1021, U.S.A. *Int. J. Robust Nonlinear Control*; 20:269–284 Publicado en línea el 27 de marzo de 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/rnc.1426.
- Ikramov, K. y Savel'eva, N. (1998). Conditionally definite matrices. *Journal of Mathematical Sciences*.. Volume 98, Number 1, 1-50, DOI: 10.1007/BF02355379.
- Li, H. y Gu, K. (2010). Discretized Lyapunov-Krasovskii functional for coupled differential-difference equations with multiple delay channels. *Automatica*. 902-909.
- Nikraves, S. y Vali, A. (2007). Lyapunov-Krasovskii Approach for Delay-Dependent Stability Analysis of Nonlinear Time-Delay Systems. *Scientia Iranica*, Vol. 14, No. 6, pp. 586.
- Ogata, K. (2009). *Modern Control Engineering*. Fifth Edition. Prentice Hall. University of Minnesota. ISBN 13: 978-0-13-615673-8.
- Papachristodoulou, A., Peet, M. y Niculescu, S. (2007). Stability Analysis of Linear Systems with Time-Varying Delays: Delay Uncertainty and Quenching. *Proceedings of the 46th IEEE Conference on Decision and Control New Orleans, LA, USA, Dec. 12-14, 2007*.
- Seuret, A. (2009). Lyapunov-Krasovskii Functionals Parameterized with polynomials. Manuscript, published in "ROCOND'09, Haifa: Israel (2009)".