

Sesión 5: Algoritmos de Equilibrio de Carga de Trabajo

19 de Mayo de 2006

Jose Luis Bosque

1



Índice

- 1.- Introducción
- 2.- Diseño del Algoritmo
- 3.- Implementación
- 4.- Resultados Experimentales
- 5.- Conclusiones

19 de Mayo de 2006

Jose Luis Bosque

2

1. Introducción

- El problema del equilibrio de carga de trabajo entre un conjunto de procesadores en un sistema paralelo es uno de los que más impacto tienen en el rendimiento del sistema.
- Hablando coloquialmente se puede decir que estos algoritmos tratan de que todos los nodos del cluster tengan en todo momento una carga de trabajo equilibrada con el resto.
- Esta correcta distribución mejora la eficiencia global de la aplicación y reduce el tiempo de ejecución, aprovechando al máximo los recursos disponibles en el sistema.
- Un mal reparto de esta carga puede suponer incrementos sustanciales en el tiempo de respuesta, debido a que haya procesadores sobrecargados que necesitan más tiempo de cómputo para realizar su trabajo, frente a procesadores infrautilizados.

19 de Mayo de 2006

Jose Luis Bosque

3

1. Introducción

- Estos algoritmos son muy complejos y constan de varias fases en su ejecución, cada una de las cuales puede atacarse como un problema independiente.
- Fuentes de desequilibrios de carga en un cluster:
 - La heterogeneidad del sistema permite que haya nodos más potentes que pueden computar una carga de trabajo mayor en el mismo tiempo.
 - La heterogeneidad de las tareas o los paquetes de datos que se asignen a los distintos nodos del sistema.
 - Las variaciones en la carga local en cada nodo provoca que su potencia de cómputo varíe a lo largo del tiempo, por lo que es necesario realizar ajustes dinámicos en la asignación de carga inicialmente planteada.
- Todos estos problemas provocan que no exista un algoritmo óptimo para todos los casos y que se hayan propuesto una gran cantidad de métodos.

19 de Mayo de 2006

Jose Luis Bosque

4

1.- Introducción

■ Definición tradicional del problema:

- Dado un conjunto de tareas y un conjunto de procesadores, encontrar una asignación de tareas a procesadores que consiga que cada nodo tenga la misma cantidad de carga de trabajo.

■ Definición para clusters:

- Dado un conjunto de tareas y un conjunto de procesadores encontrar una asignación de tareas a procesadores que consiga que cada procesador tenga en cada momento una carga de trabajo proporcional a su capacidad de cómputo, y dicha proporción sea constante para todos los nodos del sistema.

19 de Mayo de 2006

Jose Luis Bosque

5

1.- Introducción

■ Objetivo: equilibrar la carga de trabajo de un conjunto de procesadores de forma proporcional a su capacidad de cómputo en cada instante de tiempo.

- Mejorar la eficiencia de la aplicación.
- Maximizar la utilización de los recursos del sistema.

■ Problemas a considerar:

- Heterogeneidad del sistema.
- Carga local de cada nodo.

19 de Mayo de 2006

Jose Luis Bosque

6

2. Clasificación Jerárquica

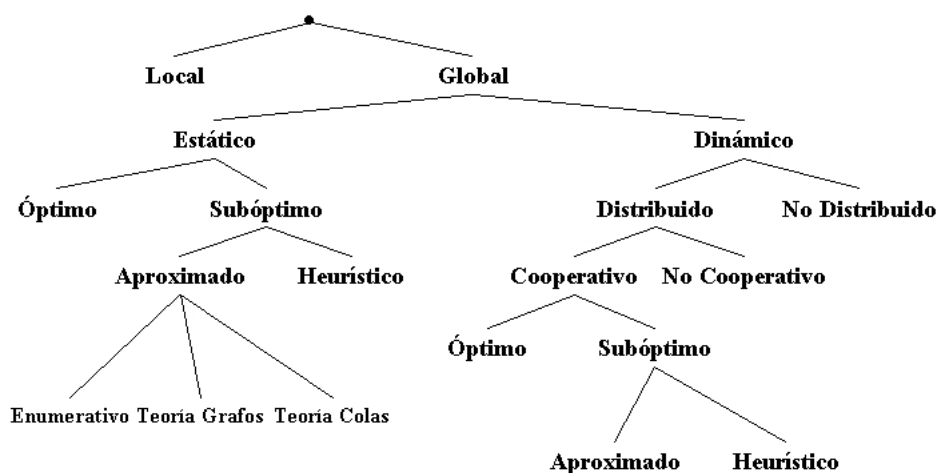
- Pretende proporcionar un medio adecuado para describir los aspectos más importantes de un método en función de un grupo reducido de características.
- Propone una visión del problema como una política de gestión de recursos para gestionar de forma eficiente su acceso y utilización. de los recursos por parte de varios consumidores.
- Cada instancia del problema de equilibrio tiene tres componentes principales: consumidores, recursos y políticas, con dos propiedades:
 - Rendimiento: satisfacción de los consumidores con la gestión de los recursos.
 - Eficiencia: satisfacción de los consumidores en términos de dificultad o coste en el acceso al gestor de recursos mismo.
- Los consumidores quieren acceder al recurso rápida y eficientemente, pero no quieren verse molestados por problemas de sobrecarga del gestor.

19 de Mayo de 2006

Jose Luis Bosque

7

2. Clasificación



19 de Mayo de 2006

Jose Luis Bosque

8

2. Clasificación

- Estática vs. dinámica: Esta elección indica el tiempo en el cual se toman las decisiones de asignación.
- Óptimos vs. subóptimos: Si se conoce toda la información sobre el estado del sistema y sobre los recursos que necesitan los procesos, se puede hacer una asignación óptima. Generalmente, este problema es computacionalmente inabordable (NP- completo) y se tienen que buscar soluciones subóptimas.
- Aproximados vs. heurísticos: Los primeros buscan una buena solución, en lugar de evaluar todas las posibles para encontrar la solución óptima. Los heurísticos son algoritmos estáticos que hacen las hipótesis más realistas sobre el conocimiento a priori tanto de los procesos como de la carga del sistema. Requieren una cantidad de recursos y tiempo razonable y son más sencillos de calcular y monitorizar.

2. Clasificación

- Soluciones dinámicas: Suponen muy escaso conocimiento a priori sobre las necesidades de los recursos de los procesos y del entorno de ejecución. No se toman decisiones hasta que un proceso empieza su vida en un entorno dinámico.
- Distribuidos vs. centralizados: Decide si la responsabilidad de la planificación dinámica global debería residir físicamente en un único procesador o si el trabajo de toma de decisiones debe hacerse de forma distribuida entre todos los procesadores.
- Cooperativo vs. no cooperativo: En los primeros cada procesador desarrolla una parte del trabajo de planificación, pero todos trabajan juntos con el mismo objetivo. En los no cooperativos los procesadores toman decisiones independientes e independientemente de su efecto en el sistema.

3. Estructura general del algoritmo

- La ejecución de un algoritmo de equilibrio de carga dinámico requiere una serie de medios para mantener una vista consistente del estado del sistema y una política de negociación para migrar carga de trabajo (procesos o datos) entre los procesadores.
- Fases del algoritmo de distribución de carga de trabajo:
 - Regla de medida del estado
 - Regla de intercambio de información
 - Regla de iniciación
 - Operación de equilibrado de la carga

19 de Mayo de 2006

Jose Luis Bosque

11

3. Estructura general del algoritmo

- Etapas de un algoritmo de equilibrio de carga:
 - Regla de medida del estado:
 - Mide la capacidad de cómputo de un nodo en un instante de tiempo dado.
 - El estado depende de la capacidad de proceso del nodo y de la carga local que tenga en un instante dado.
 - Es importante en algoritmos dinámicos ya que determina las decisiones de equilibrio de carga.

19 de Mayo de 2006

Jose Luis Bosque

12

3. Estructura general del algoritmo

Regla de medida del estado:

- Se basa en un índice de carga con las siguientes propiedades:
 - Dinámico.
 - Reflejar el estado actual.
 - Sencillo y rápido de evaluar.
 - Resumir el estado global del nodo.
 - Estable.
 - Adaptable.

3. Estructura general del algoritmo

■ Regla de información:

- Especifica como recoger y mantener información de carga de todos los procesadores
- Bajo demanda.
- Periódica.
- Por eventos.
- Globales vs. Basada en dominios.

3. Estructura general del algoritmo

■ Regla de iniciación:

- Determina cuando debe iniciarse una operación de equilibrio de carga.
- Iniciada por el emisor.
- Iniciada por el receptor.
- Simétrica.
- Periódicas.

3. Estructura general del algoritmo

■ Operaciones de equilibrio de carga:

- Regla de localización: determina el socio de la operación de equilibrio.
 - Algoritmos directos.
 - Algoritmos de vecino más próximo.
- Regla de distribución: Cómo distribuir la carga entre los nodos que participan en la operación de equilibrio.
- Regla de selección: selecciona el/los procesos a intercambiar.
 - Con interrupción.
 - Sin interrupción.

4.- Diseño del Algoritmo

- Distribución de la carga de trabajo proporcional a la capacidad de cómputo de cada nodo.
- Características generales:
 - Dinámicos vs. Estático.
 - Distribuidos vs centralizados.
 - Globales vs locales.
 - Basados en estados.
 - Se autoeliminan en situaciones de equilibrio.

19 de Mayo de 2006

Jose Luis Bosque

17

4.- Diseño del Algoritmo

- Regla de medida del estado (I):
 - Índice de carga basado en la asignación por la potencia de cómputo del nodo.

$$A = \begin{cases} \frac{1}{N+1} & \text{if } U \geq \frac{1}{N} \\ 1-U & \text{Otherwise} \end{cases}$$

- Expresión del índice de carga:

$$I_i = \frac{P_i \cdot A_i}{P_{MAX}}$$

19 de Mayo de 2006

Jose Luis Bosque

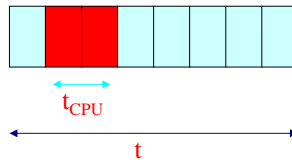
18

4.- Diseño del Algoritmo

■ Modelo RTP

- Las tareas intensivas en CPU siempre ejecutan hasta el final de su rodaja de tiempo
- Las tareas no intensivas en cómputo algunas veces no finalizan completamente su rodaja de tiempo.
- La fracción de tiempo de CPU consumido por tareas no intensivas en cómputo, denotado por X , viene dado por la siguiente expresión:

$$X = \frac{t_{CPU}}{t}$$



$$X=0.25$$

4.- Diseño del Algoritmo

■ Modelo RTP

- La asignación predicha para una tarea no intensiva en cómputo que llega a un nodo es:

$$A_i = \frac{X_i}{(n+1) + \sum_{k=0}^m X_k} \quad 1 \leq i \leq m$$

- n : Número de tareas intensivas en cómputo.
- m : Número de tareas no intensivas en cómputo.

4.- Diseño del Algoritmo

Modelo RTP

- Las nuevas tareas se supone que son siempre intensivas, pues es el peor caso. Entonces el número de tareas intensivas pasa a ser $n+1$
- La asignación para la nueva tarea puede calcularse como todo el tiempo no consumido por las tareas no intensivas compartida mediante round robin con el resto de tareas intensivas:

$$A = \frac{1 - \sum_{i=1}^m A_i}{n + 1}$$

19 de Mayo de 2006

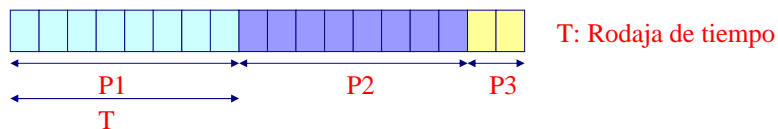
Jose Luis Bosque

21

4.- Diseño del Algoritmo

Modelo RTP:

- Ejemplo con dos tareas intensivas (P1 y P2) y una no intensiva con $X=0.25$ (P3):



$$A(P3) = \frac{0.25}{(2+1)+0.25} = 0.0769$$

Tiempo de respuesta predicho para una nueva tareas:

$$A = A(P1) = A(P2) = \frac{1 - A(P3)}{3} = 0.308$$

$$t_{pred} = \frac{t}{A} = \frac{t}{0.308}$$

19 de Mayo de 2006

Jose Luis Bosque

22

4.- Diseño del Algoritmo

- Regla de medida del estado (II):
 - Basado en tres estados: Estos estados se basan en un índice de carga y determinan la capacidad de un nodo para manejar nuevas trabajos:
 - Receptor.
 - Neutro.
 - Emisor.
 - Determinan el comportamiento de un nodo en una operación de equilibrio.
 - Los cambios de estado dependen de las variaciones en el índice.

4.- Diseño del Algoritmo

- Regla de información:
 - Determina el conocimiento que cada nodo del estado global del sistema.
 - Política Global basada en eventos.
 - Elimina puntos de sincronización.
 - Mantiene actualizado la información del sistema.
 - Sólo se comunican los cambios de o a receptor.
 - Cada nodo mantiene una cola de receptores.

4.- Diseño del Algoritmo

■ Regla de Iniciación:

- Regla iniciada por el emisor.
- Las decisiones son completamente locales.
- Los nodos neutros no pueden tomar parte en las operaciones de equilibrio.
- El ancho del neutro proporciona un mecanismo para controlar el máximo nivel de desequilibrio en el sistema.
- Se fija un mínimo de asignación para cada tarea local.

19 de Mayo de 2006

Jose Luis Bosque

25

4.- Diseño del Algoritmo

■ Operación de Equilibrio de carga:

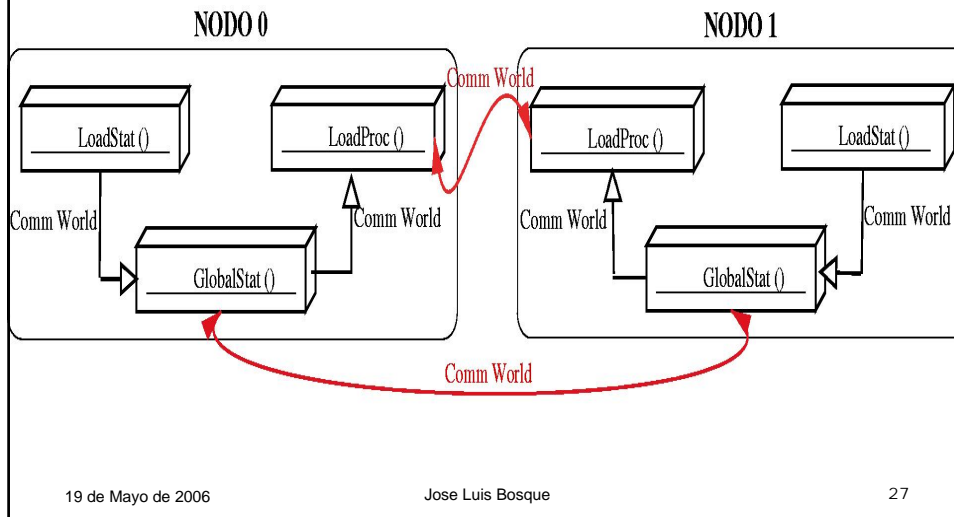
- Regla de localización: se selecciona el primer nodo de la cola de receptores y se inicia una negociación.
 - Es el nodo más antiguo en la cola y por lo tanto el más estable.
- El candidato puede aceptar o rechazar la tarea.
- El algoritmo se auto-elimina si la cola está vacía.
- Si no se encuentran socios la tarea se bloquea hasta que se produce un cambio de estado.

19 de Mayo de 2006

Jose Luis Bosque

26

5.- Implementación del algoritmo

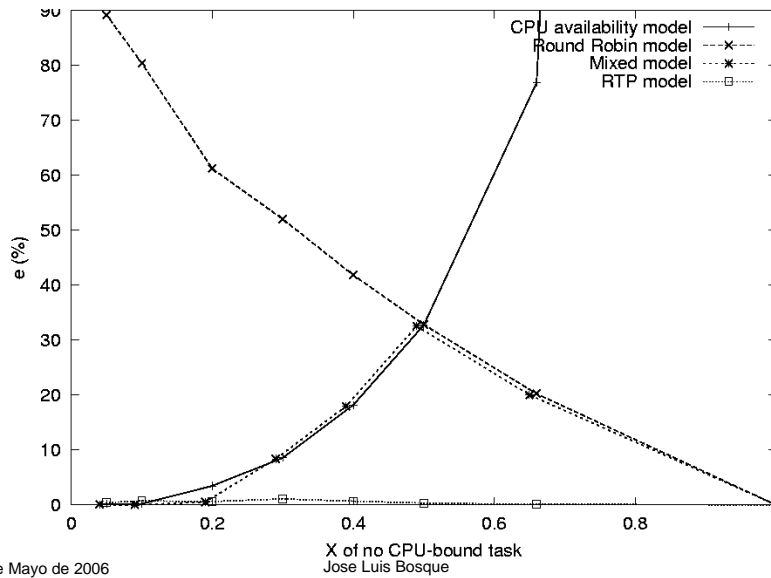


6.- Resultados Experimentales

■ Arquitectura del sistema:

- Cluster heterogéneo con 8 PC's.
- 4 nodos Pentium III a 550 MHz.
- 4 nodos Pentium III a 733 MHz.
- Red Fast Ethernet a 100 Mbps.
- Conjunto de tareas intensivas en cómputo.

6.- Resultados Experimentales

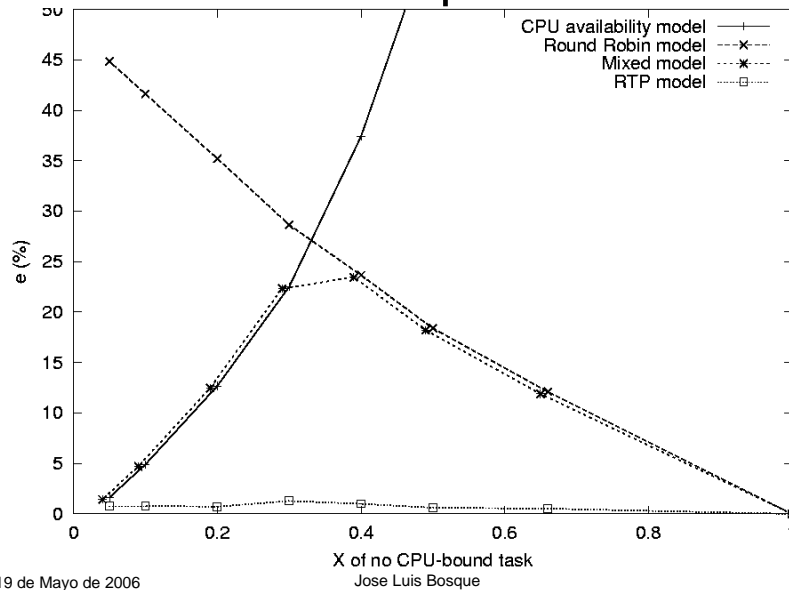


19 de Mayo de 2006

Jose Luis Bosque

29

6.- Resultados Experimentales

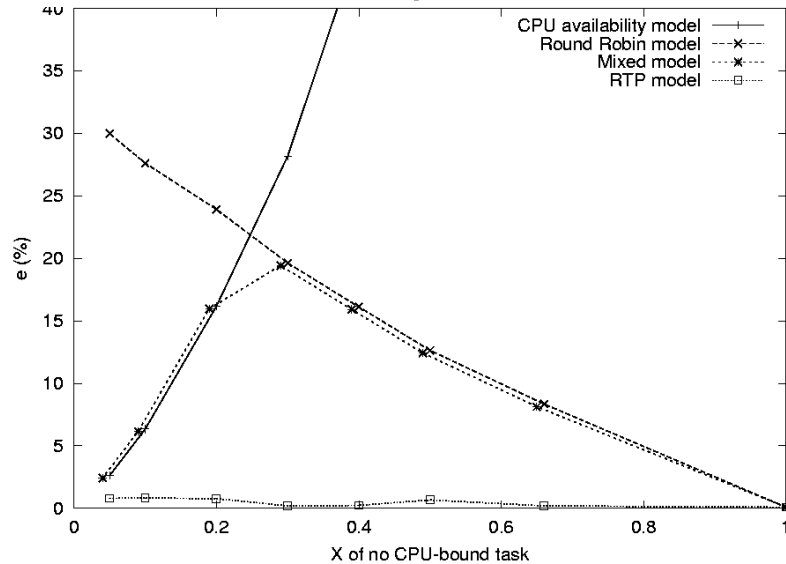


19 de Mayo de 2006

Jose Luis Bosque

30

6.- Resultados Experimentales



19 de Mayo de 2006

Jose Luis Bosque

31

6.- Resultados Experimentales

■ Resultados sin carga local:

- Objetivo: comprobar la sobrecarga del algoritmo.
- Distribución inicial óptima.
- Los tiempos con algoritmo son siempre mejores que sin él.
 - Mejor aprovechamiento de los recursos.
 - Asignación mínima de CPU a cada tarea.
 - Ancho del neutro = 0.1 \Rightarrow 25% de asignación por tarea.
- No se intenta realizar ninguna operación de equilibrio.

19 de Mayo de 2006

Jose Luis Bosque

32

6.- Resultados Experimentales

■ Experimentos con carga externa:

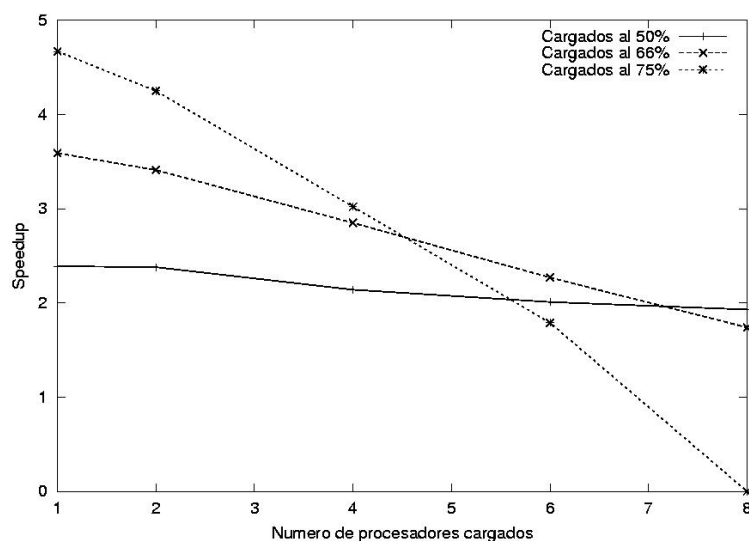
- Distribución inicial homogénea.
- Se cargan los nodos localmente al 50%, 66% y 75%.
- Se incrementa el número de nodos cargados.
- La ganancia aumenta con la carga local, pero disminuye al aumentar el número de nodos cargados.
- El número de operaciones realizadas aumenta tanto con la carga local como con el número de nodos cargados.
- Las pendientes de las curvas de ganancia son mayores al aumentar la carga local de cada nodo.

19 de Mayo de 2006

Jose Luis Bosque

33

6.- Resultados Experimentales

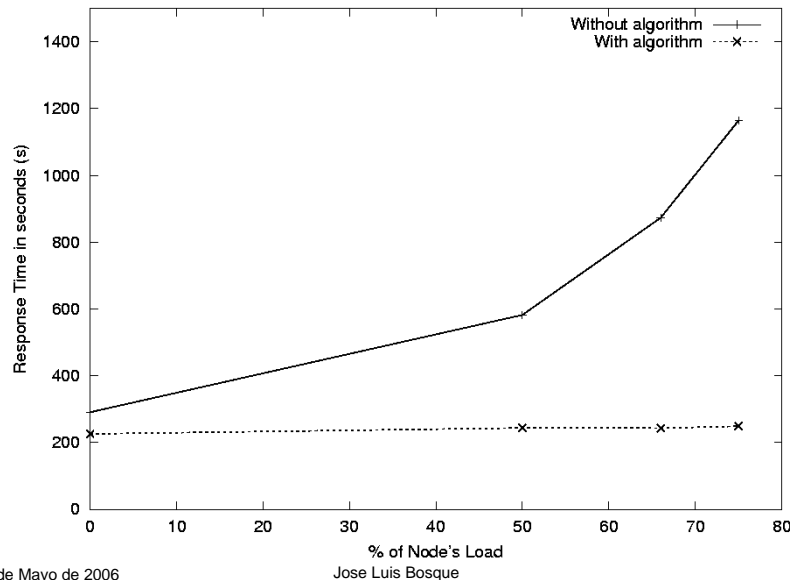


19 de Mayo de 2006

Jose Luis Bosque

34

6.- Resultados Experimentales



19 de Mayo de 2006

Jose Luis Bosque

35

6.- Resultados Experimentales

■ Variación del ancho del neutro:

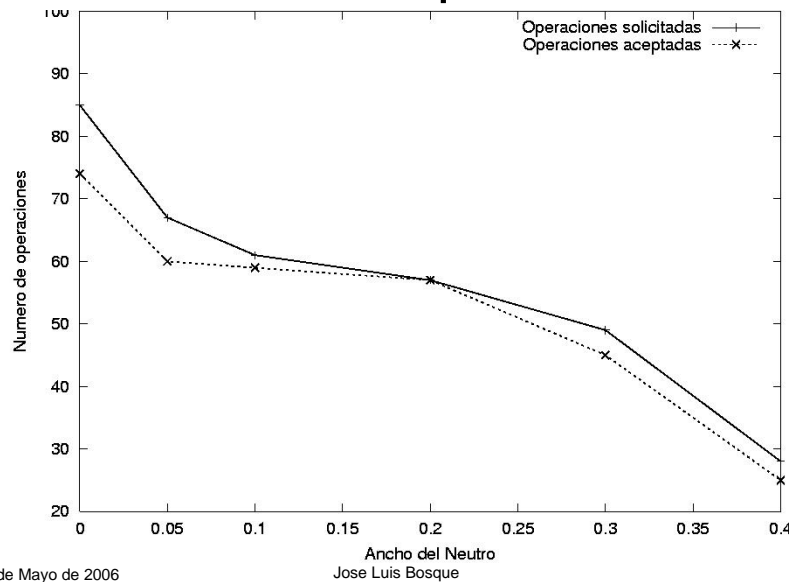
- Objetivo: evaluar la variación del desequilibrio del sistema en función del ancho del neutro.
- Métrica: diferencia entre operaciones intentadas y aceptadas.
- Al aumentar el ancho del neutro el desequilibrio siempre aumenta.
- El número de operaciones de equilibrio disminuye al incrementarse el ancho del neutro.
- Es necesario un equilibrio entre el número de operaciones realizadas y el desequilibrio permitido en el sistema.

19 de Mayo de 2006

Jose Luis Bosque

36

6.- Resultados Experimentales



19 de Mayo de 2006

Jose Luis Bosque

37