(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 639 781**

(51) Int. CI.:

G06F 11/34 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **13.02.2014 E 14155094 (7)**

(97) Fecha y número de publicación de la concesión europea: **21.06.2017 EP 2879055**

(54) Título: **Sistema y método que facilitan la predicción del rendimiento de la aplicación multi-hilo en presencia de cuellos de botella de recursos**

(30) Prioridad:

25.11.2013 IN MU36992013

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
30.10.2017

(73) Titular/es:

TATA CONSULTANCY SERVICES LIMITED (100.0%)
Nirmal Building, 9th Floor Nariman Point
Maharashtra
400021 Mumbay, IN

(72) Inventor/es:

DUTTAGUPTA, SUBHASRI;
VIRK, RUPINDER SINGH y
NAMBIAR, MANOJ KARUNA KARAN

(74) Agente/Representante:

ELZABURU, S.L.P

DESCRIPCIÓN

Sistema y método que facilitan la predicción del rendimiento de la aplicación multi-hilo en presencia de cuellos de botella de recursos

Campo Técnico

La presente descripción se refiere en general a un método y sistema que predicen el rendimiento de una aplicación multi-hilo. Más particularmente, la presente descripción se refiere a la predicción del rendimiento de la aplicación multi-hilo en presencia de cuellos de botella de recursos.

Antecedentes

En el sistema empresarial de múltiples niveles típico, una aplicación de software es accedida normalmente por un gran número de usuarios. Debido al gran número de usuarios, la probabilidad de cuellos de botella de recursos ha aumentado drásticamente. La presencia de cuellos de botella de recursos obstaculiza típicamente el rendimiento de la aplicación de software. El cuello de botella puede ocurrir bien debido a los recursos de software o bien a los recursos de hardware. Con el fin de aumentar la escalabilidad del sistema empresarial de múltiples niveles, los cuellos de botella de recursos deberían ser detectados mientras la aplicación de software está siendo probada.

La mayoría de los sistemas empresariales de múltiples niveles han empleado distintos métodos para identificar los cuellos de botella de recursos ya que los cuellos de botella de recursos pueden limitar el rendimiento total del sistema empresarial de múltiples niveles. Los cuellos de botella de recursos son identificados solamente si los cuellos de botella de recursos ocurren explícitamente durante la propia prueba de rendimiento o en una etapa muy posterior en un entorno de producción. El modelado de la simulación de eventos discretos es una de las aproximaciones bien conocidas utilizadas para predecir el rendimiento de la aplicación de software. Sin embargo, el desarrollo de tal modelo de simulación es un proceso que consume tiempo.

Un documento de la técnica anterior (XP58231385) de Daniel A. Menascé titulado "Simple analytic modeling of software contention" describe una aproximación general al modelado de contención de software usando redes de filas de espera. La aproximación es llamada como proceso iterativo de nivel dos de SQN HQN. Las redes de filas de espera representan recursos de software y recursos de hardware. Este documento proporciona una técnica de modelado analítico simple para estimar el retraso de contención de software. Sin embargo, cuando un gran número de hilos están accediendo a la aplicación multi-hilos, la técnica de este documento falla ya que éste algoritmo entra en bucle infinito y falla para determinar el rendimiento de la aplicación multi-hilos.

Otro documento de la técnica anterior titulado "Bridging the gap between queuing based performance evaluation" describe el uso de redes de espera en cola de múltiples clases durante la fase de especificación del flujo de diseño para modelar sistemas orientados al flujo de datos. Se ha desarrollado un marco de evaluación de arquitectura de plataforma llamado "systemQ" que se basa en redes de filas de espera, en donde SystemQ está dirigido a la modelización y evaluación de sistemas a partir de dominios orientados a flujos de datos tales como tratamiento de red.

Otra solicitud de patente US20110218786 describe un modelo de red de filas de espera de entorno de servicios web. El comportamiento de servicio se resume en tres fases en serie, en paralelo e inactivas. Se ha proporcionado un método para estimar los parámetros del modelo basándose en la técnica de aproximación estocástica.

Otra solicitud de patente JP2000298593 describe la predicción de un índice de rendimiento para el grado paralelo de ordenadores paralelos en un entorno multitarea. Cuando la especificación de un grado u ordenadores paralelos dado como entrada al sistema de predicción de rendimiento, se genera un modelo sobre la base de esta especificación, se calculan los valores de predicción del índice de rendimiento tales como el grado de mejora e índices de rendimiento más detallados tales como el rendimiento, una respuesta y un recurso que utiliza la tasa a partir del modelo generado. Sin embargo, esta técnica anterior falla al predecir el rendimiento cuando un gran número de hilos comienza a acceder a la sección crítica de la aplicación multi-hilos.

Este compendio es proporcionado para introducir aspectos relacionados con el sistema o sistemas y métodos que facilitan la predicción del rendimiento de una aplicación multi-hilos en presencia de cuellos de botella de recursos y los aspectos son además descritos después en la descripción detallada.

La presente descripción se refiere a un sistema según la reivindicación 1 independiente adjunta, para facilitar la predicción de rendimiento de una aplicación de software multi-hilos en presencia de cuellos de botella de recursos. El sistema comprende un procesador y una memoria acoplada al procesador. El procesador es capaz de ejecutar una pluralidad de módulos almacenados en la memoria. La pluralidad de módulos comprende un módulo de representación configurado para representar una o más redes de filas de espera para recursos empleados para ejecutar la aplicación de software multi-hilos. Los recursos comprenden recursos de hardware y recursos de software. Las redes de filas de espera son una de una red de puesta en colas de hardware o una red de puesta en colas de software. Las redes de puesta en colas son utilizadas para detectar el nivel de concurrencia en el que se encuentra el cuello de botella de recursos mientras que accede a la aplicación de software multi-hilos, en donde el

cuello de botella de recursos es el primer cuello de botella de recursos, en donde el cuello de botella de recursos es el primer cuello de botella de recursos, y en donde se detecta la concurrencia comprobando el número de hilos que acceden a la aplicación de software multi-hilos. La memoria almacena un módulo de computación configurado para calcular unas métricas de rendimiento para la aplicación de software multi-hilos. Las métricas de rendimiento comprenden uno o más parámetros. El módulo de computación es configurado para utilizar una técnica iterativa con un valor predeterminado de solicitud de servicio para desarrollar una relación entre la red de filas de espera de software y la red de filas de espera de hardware, en donde la relación comprende la utilización o compartición de los recursos de software con respecto a los recursos de hardware con respecto al valor predeterminado de la solicitud de servicio. La ejecución de la técnica iterativa comprende, la obtención de un número de hilos bloqueados en una sección crítica de los recursos de software basándose en el valor predeterminado de la solicitud de servicio, en donde los hilos bloqueados en la sección crítica son hilos que esperan a entrar en la sección crítica y obtener un tiempo de residencia en cada uno de los recursos de hardware basándose en el valor predeterminado de la solicitud de servicio. La ejecución de la técnica iterativa comprende además actualizar una solicitud de servicio de la sección crítica de una iteración actual a una solicitud de servicio de la sección crítica actualizada para reflejar una espera para los recursos de hardware utilizando el tiempo de residencia. La solicitud de servicio de la sección crítica es actualizada de manera iterativa para tener en cuenta la contención de recursos en los recursos de hardware basándose en el tiempo de residencia. La ejecución de la técnica iterativa comprende además la comparación del número de hilos bloqueados de la iteración actual con el número de hilos bloqueados de una iteración previa con el fin de comprobar si el número de hilos bloqueados de la iteración actual son más elevados que el número de hilos bloqueados de la iteración previa, y comprobar si la solicitud de servicio de sección crítica actualizada es mayor que la solicitud de servicio de sección crítica de la iteración previa. La ejecución de la técnica iterativa comprende además identificar una diferencia entre el número de hilos bloqueados en una sección crítica de la iteración actual y el número de hilos bloqueados en la sección crítica en la iteración previa.

La ejecución de la técnica iterativa comprende además ejecutar de manera repetida la técnica iterativa 1) si la diferencia en el número de hilos bloqueados es mayor que un límite predefinido, y 2) si el número de hilos bloqueados de la iteración actual es mayor que el número de hilos bloqueados de la iteración previa y 3) si la solicitud de servicio de sección crítica actualizada es mayor que la solicitud de servicio de sección crítica de la iteración previa; de otro modo obtener las métricas de rendimiento con la solicitud de servicio de sección crítica actualizada para la aplicación de software multi-hilos cuando la diferencia en el número de hilos bloqueados es inferior al límite predefinido, en donde el límite predefinido indica que la diferencia en el número de hilos bloqueados ha convergido sobre varias iteraciones de la técnica iterativa, determinando por lo tanto un rendimiento de la aplicación de software multi-hilos. El módulo de computación está configurado para determinar uno o más parámetros en forma de un valor de rendimiento y un valor de tiempo de respuesta para la aplicación de software multi-hilos según la relación, determinando por tanto el rendimiento de la aplicación de software multi-hilos.

La presente descripción se refiere también a un método según la reivindicación 9 independiente adjunta, para facilitar la predicción de rendimiento de una aplicación de software multi-hilos en presencia de cuellos de botella de recursos. El método comprende representar una o más redes de filas de espera de recursos empleadas para ejecutar la aplicación de software multi-hilos. Los recursos comprenden recursos de hardware y recursos de software. Las redes de filas de espera son de una red de filas de espera de hardware o una red de filas de espera de software. Las redes de espera en colas ayudan a detectar un nivel de concurrencia en el que se encuentra el cuello de botella de recursos mientras que accede a la aplicación de software multi-hilos. El cuello de botella de recursos es el primer cuello de botella de recursos, y en donde la concurrencia es detectada comprobando el número de hilos que acceden a la aplicación de software multi-hilos. El método proporciona además métricas de rendimiento de computación para la aplicación de software multi-hilos. Las métricas de rendimiento comprenden uno o más parámetros. La computación comprende utilizar una técnica iterativa con un valor predeterminado de solicitud de servicio para desarrollar una relación entre la red de filas de espera de software y la red de filas de espera de hardware, en donde la relación comprende la utilización o compartición de los recursos de software con respecto a los recursos de hardware con respecto al valor predeterminado de la solicitud de servicio y determinar uno o más parámetros en forma de un valor de rendimiento y un valor de tiempo de respuesta para la aplicación de software multi-hilos según la relación, determinando por tanto el rendimiento de la aplicación de software multi-hilos.

La ejecución de la técnica iterativa comprende, obtener un número de hilos bloqueados en una sección crítica de los recursos de software basándose en el valor predeterminado de la solicitud de servicio, en donde los hilos bloqueados en la sección crítica son hilos que esperan a entrar en la sección crítica y obtener un tiempo de residencia en cada uno de los recursos de hardware basándose en el valor predeterminado de la solicitud de servicio. La ejecución de la técnica iterativa comprende además actualizar una solicitud de servicio de sección crítica de una iteración actual a una solicitud de servicio de sección crítica actualizada para reflejar una espera para los recursos de hardware utilizando el tiempo de residencia. La solicitud de servicio de sección crítica es actualizada de manera iterativa para tener en cuenta la contención de recursos en los recursos de hardware basándose en el tiempo de residencia. La ejecución de la técnica iterativa comprende además comparar el número de hilos bloqueados de la iteración actual con el número de hilos bloqueados de una iteración previa con el fin de comprobar si el número de hilos bloqueados de la iteración actual es mayor que el número de hilos bloqueados de la iteración previa, y comprobar si la solicitud de servicio de sección crítica actualizada es mayor que la solicitud de servicio de sección crítica de la iteración previa. La ejecución de la técnica iterativa comprende además identificar una diferencia

entre el número de hilos bloqueados en una sección crítica de la iteración actual y el número de hilos bloqueados en la sección crítica en la iteración previa.

La ejecución de la técnica iterativa comprende además ejecutar de manera repetida la técnica iterativa 1) si la diferencia en el número de hilos bloqueados es mayor que un límite predefinido, y 2) si el número de hilos bloqueados de la iteración actual es mayor que el número de hilos bloqueados de la iteración previa y 3) si la solicitud de servicio de sección crítica actualizada es mayor que la solicitud de servicio de sección crítica de la iteración previa; de otro modo obtener las métricas de rendimiento con la solicitud de servicio de sección crítica actualizada para la aplicación de software multi-hilos cuando la diferencia en el número de hilos bloqueados es inferior que el límite predefinido, en donde el límite predefinido indica que la diferencia en el número de hilos bloqueados ha convergido sobre una serie de iteraciones de la técnica iterativa, determinando por tanto un rendimiento de la aplicación de software multi-hilos.

La presente descripción se refiere también a un producto de programa informático según la reivindicación 11 independiente adjunta que ha incorporado en él un programa informático para facilitar la predicción de rendimiento de una aplicación de software multi-hilos en presencia de cuellos de botella de recursos. El producto de programa informático comprende un código de programa para representar una o más redes de filas de espera para recursos empleados para ejecutar la aplicación de software multi-hilos. Los recursos comprenden recursos de hardware y recursos de software y las redes de filas de espera es una de una red de filas de espera de hardware o una red de filas de espera de software. Las redes de filas de espera son utilizadas para detectar un nivel de concurrencia en el que se encuentra el cuello de botella de recursos mientras que accede a la aplicación de software multi-hilos en donde el cuello de botella de recursos es el primer cuello de botella de recursos, y en donde la concurrencia es detectada comprobando el número de hilos que acceden a la aplicación de software multi-hilos. El producto de programa informático comprende un código de programa para calcular unas métricas de rendimiento para una aplicación de software multi-hilos. Las métricas de rendimiento comprenden uno o más parámetros. El código de programa para calcular comprende un código de programa para utilizar n técnica iterativa con un valor predeterminado de solicitud de servicio para desarrollar una relación entre la red de filas de espera de software y la red de filas de espera de hardware, en donde la relación comprende la utilización o compartición de los recursos de software con respecto a los recursos de hardware con respecto al valor predeterminado de la solicitud de servicio. El producto de programa informático comprende un código de programa para determinar uno o más parámetros en forma de un valor de rendimiento y un valor de tiempo de respuesta para la aplicación de software multi-hilos, determinando por tanto el rendimiento de la aplicación de software multi-hilos.

Breve Descripción de los Dibujos

La descripción detallada es descrita con referencia a las figuras adjuntas. En las figuras, el dígito o dígitos más a la izquierda de un número de referencia identifica la figura en la cual el número de referencia aparece en primer lugar. Los mismos números son utilizados a lo largo de los dibujos para referirse a características y componentes similares.

La fig. 1 ilustra una implementación de red de un sistema que facilita la predicción de rendimiento de una aplicación multi-hilos en presencia de cuellos de botella de recursos mostrados, según una realización del presente objeto.

La fig. 2 ilustra el sistema que facilita la predicción de rendimiento de una aplicación multi-hilos en presencia de cuellos de botella de recursos, según una realización del presente objeto.

La fig. 3 ilustra un método que facilita la predicción de rendimiento de una aplicación multi-hilos en presencia de cuellos de botella de recursos, según una realización del presente objeto.

La fig. 4 ilustra una red de filas de espera de software y una red de filas de espera de hardware según una realización del presente objeto.

La fig. 5 ilustra unas métricas de rendimiento en el caso en donde las secciones crítica y no crítica son del mismo tamaño según una realización del presente objeto.

La fig. 6 ilustra unas métricas de rendimiento en un caso en donde la solicitud de servicio de la sección no crítica es más de dos veces la solicitud de servicio de la sección crítica según una realización ejemplar del presente objeto.

La fig. 7 ilustra unas métricas de rendimiento para solicitudes de múltiples clases según una realización ejemplar del presente objeto.

La fig. 8 ilustra unas métricas de rendimiento para la agrupación de recursos según una realización ejemplar del presente objeto.

Descripción Detallada

Aunque los aspectos del sistema y métodos descritos para facilitar la predicción de rendimiento de una aplicación multi-hilos en presencia de cuellos de botella de recursos pueden ser implementados en cualquier número de

diferentes sistemas informáticos, entornos, y/o configuraciones, las realizaciones son descritas en el contexto del siguiente sistema ejemplar.

Con referencia ahora a la fig. 1, se muestra una implementación de red 100 del sistema 102 para facilitar la predicción de rendimiento de una aplicación multi-hilos en presencia de cuellos de botella de recursos. Múltiples redes de filas de espera son representadas por el sistema 102. Las múltiples redes de filas de espera comprenden una red de filas de espera de hardware y una red de filas de espera de software. Las redes de filas de espera son utilizadas para detectar un nivel de concurrencia en el que se encuentran los cuellos de botella de recursos mientras que accede a la aplicación multi-hilos. Utilizando un valor de solicitud de servicio predeterminado, unas métricas de rendimiento son calculadas según la relación entre la red de filas de espera de hardware y la red de filas de espera de software. Las métricas de rendimiento comprenden uno o más parámetros. Uno o más parámetros predicen el rendimiento de la aplicación multi-hilos.

Aunque el objeto actual es explicado considerando que el sistema 102 es implementado como una aplicación sobre un servidor, puede entenderse que el sistema 102 puede ser implementado también en una variedad de sistemas informáticos, tales como un ordenador portátil, un ordenador de sobremesa, un portátil, una estación de trabajo, un ordenador central, un servidor, un servidor de red, y similares. En una implementación, el sistema 102 puede ser implementado en un entorno basado en la nube. Se entenderá que el sistema 102 puede ser accedido por múltiples usuarios a través de uno o más dispositivos de usuario 104-1, 104-2… 104-N, referidos colectivamente como usuario 104 en lo sucesivo, o aplicaciones que residen sobre los dispositivos de usuario 104. Ejemplos de los dispositivos de usuario 104 pueden incluir, pero no están limitados a, un ordenador portátil, un asistente digital personal, un dispositivo manual, y una estación de trabajo. Los dispositivos de usuario 104 son acoplados de manera comunicativa al sistema 102 a través de una red 106.

En una implementación, la red 106 puede ser una red inalámbrica, una red cableada, o una combinación de las mismas. La red 106 puede ser implementada como uno de los diferentes tipos de redes, tal como intranet, red de área local (LAN), red de área amplia (WAN), Internet, y similares. La red 106 puede o bien ser una red dedicada o bien una red compartida. La red compartida representa una asociación de diferentes tipos de redes que utiliza una variedad de protocolos, por ejemplo, Protocolo de Transferencia de Hipertexto (HTTP), Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP), Protocolo de Aplicación Inalámbrica (WAP), y similares, para comunicar entre sí. Además la red 106 puede incluir una variedad de dispositivos de red, incluyendo routers, puentes, servidores, dispositivos informáticos, dispositivos de almacenamiento, y similares.

Con referencia ahora a la fig. 2, el sistema 102 es ilustrado según una realización del presente objeto. En una realización, el sistema 102 puede incluir al menos un procesador 202, una interfaz 204 de entrada/salida (I/O), una memoria 208. Al menos un procesador 202 puede ser implementado como uno o más microprocesadores, microordenadores, micro-controladores, procesadores de señal digital, unidades de tratamiento central, máquinas de Estado, circuitos lógicos, y/o cualesquiera dispositivos que manipulan señales basándose en instrucciones operacionales. Entre otras capacidades, al menos un procesador 202 es configurado para buscar y ejecutar instrucciones legibles por ordenador almacenadas en la memoria 208.

La interfaz 204 de I/O puede incluir una variedad de interfaces de software y hardware, por ejemplo, una interfaz web, una interfaz de usuario gráfica, y similares. La interfaz 204 de I/O puede permitir al sistema 102 interactuar con un usuario directamente o a través del dispositivo de cliente 104. Además, la interfaz 204 de I/O puede habilitar el sistema 102 para comunicar con otros dispositivos informáticos, tales como servidores web y servidores de datos externos (no mostrados). La interfaz 204 de I/O puede facilitar múltiples comunicaciones dentro de una amplia variedad de tipos de redes y protocolos, incluyendo redes cableadas, por ejemplo, LAN, cable, etc., y redes inalámbricas, tales como WLAN, celular, o satélite. La interfaz 204 de I/O puede incluir uno o más puertos para conectar un número de dispositivos entre sí o a otro servidor.

La memoria 208 puede incluir cualquier medio legible por ordenador conocido en la técnica incluyendo, por ejemplo, memoria volátil, tal como memoria de acceso aleatorio estática (SRAM) y memoria de acceso aleatorio dinámica (DRAM); y/o memoria no volátil, tal como memoria de sólo lectura (ROM), ROM programable que se puede borrar, memorias flash, discos duros, discos ópticos, y cintas magnéticas. La memoria 208 puede incluir módulos 210 y datos 212.

Los módulos 210 incluyen rutinas, programas, objetos, componentes, estructuras de datos, etc., que realizan tareas particulares, funciones o implementan tipos de datos abstractos particulares. En una implementación, los módulos 210 pueden incluir un módulo de representación 212 y un módulo de computación 214. Los otros módulos 216 pueden incluir programas o instrucciones codificadas que complementan aplicaciones y funciones del sistema 102.

Los datos 218, entre otras cosas, sirven como un repositorio para almacenar datos procesados, recibidos, y generados por uno o más de los módulos 216. Los datos 218 pueden incluir también una base de datos 220, y otros datos 224. Los otros datos 224 pueden incluir datos generados como resultado de la ejecución de uno o más módulos en el otro módulo 216.

La presente descripción se refiere a un sistema o sistemas y métodos que facilitan la predicción de rendimiento de la

aplicación multi-hilos en presencia de cuellos de botella de recursos. La red de filas de espera de software y una red de filas de espera de hardware son utilizadas para detectar la presencia de cualquier cuello de botella mientras que se accede a la aplicación. Un valor de servicio predeterminado es utilizado para calcular unas métricas de rendimiento. Uno o más parámetros de métricas de rendimiento como el relativo a la aplicación multi-hilos, tiempo de respuesta y utilización de recursos individuales predice el rendimiento de la aplicación multi-hilos.

Las redes de filas de espera así representadas por el módulo de representación 112 se refieren a modelos analíticos. Los modelos analíticos son aproximaciones matemáticas de un sistema mundial real y pueden ser utilizados para predecir ciertos comportamientos de uno o más sistemas que son modelados. La red de filas de espera (o modelos de filas de espera) recibida y utilizada aquí predice el comportamiento de la aplicación multi-hilos en términos de diferentes medidas relacionadas con el rendimiento (métricas de rendimiento). Estas medidas pueden incluir pero no están limitadas a una cantidad esperada de tiempo de espera. El tiempo esperado se refiere a que un objeto dado se gastará dentro de una cola antes de ser procesado. Las medidas relacionadas con el rendimiento incluyen además un número de objetos esperado dentro de una cola en un instante particular, una probabilidad de que cualquier cola este vacía, un tiempo de servicio para un tipo particular de objeto, o similares. Los modelos analíticos (modelos de filas de espera o redes de filas de espera) son utilizados frecuentemente para predecir si el sistema particular (aquí la aplicación multi-hilos) será capaz de cumplir con la calidad establecida de las métricas de servicio, tal como el tiempo de respuesta.

El sistema y métodos presentes predicen el rendimiento de una aplicación multi-hilos que es accedida por múltiples usuarios. La aplicación multi-hilos puede estar dividida en una sección crítica y una sección no crítica. Una sección de un código que puede ser ejecutado solamente por uno de muchos hilos concurrentes se conoce como sección crítica. En general, un acceso a una sección crítica es controlado por un bloqueo o semáforo. La sección restante de la aplicación multi-hilos que puede ser ejecutada simultáneamente por cualquier número de hilos concurrentes se conoce como sección no crítica.

Los cuellos de botella de recursos pueden ocurrir bien debido a recursos de hardware o bien a recursos de software. El sistema y método facilitan la búsqueda del número de usuarios que acceden a la aplicación multi-hilos cuando es encontrado el primer cuello de botella. El primer cuello de botella puede ocurrir bien debido a la limitación de recursos de hardware o bien a la restricción de recursos de software. Las aplicaciones multi-hilos pueden tener comunicaciones de mensajes síncrona o asíncrona.

El sistema 102 toma las siguientes presunciones mientras que predice el rendimiento de la aplicación multi-hilos:

- El sistema 102 está en estado estacionario. El análisis de acceso de la aplicación multi-hilos no es válido para ningún comportamiento transitorio del sistema.

- Todo los que espera para un recurso de software se implementa solamente a través de construcciones de sincronización explícitas.

- La aplicación multi-hilos puede tener un número de secciones críticas pero puede tener solamente una sección no crítica. La sección no crítica representa toda la parte del código sin ninguna de las secciones críticas.

Ejemplos no limitativos de la aplicación multi-hilos comprenden las aplicaciones escritas en lenguajes de programación con construcciones de sincronización explícitas como en C, C++, Java. En aplicaciones escritas en lenguajes de programación es posible conocer la porción exacta del código que es sincronizado.

La memoria 108 almacena el módulo 112 de representación configurado para representar las redes de filas de espera para recursos empleados para ejecutar la aplicación multi-hilos. Las redes de filas de espera están formadas para modelar la contención para recursos de software y hardware. Las redes de filas de espera están formadas para detectar un nivel de concurrencia en el que el cuello de botella (primer cuello de botella de recursos) es encontrado mientras se accede a la aplicación multi-hilos. La concurrencia es detectada comprobando el número de hilos que acceden a la aplicación de software multi-hilos.

El módulo 112 de representación representa la red de filas de espera de hardware y la red de filas de espera de software. La red de filas de espera de hardware que representa todos los recursos hardware. Los recursos de hardware comprenden una CPU (Unidad de Tratamiento Central), una memoria, un disco o una combinación de los mismos. La red de filas de espera de software representa módulos de software correspondientes a las secciones crítica y no crítica. Los recursos de software comprenden los módulos de software.

En las redes de filas de espera, los usuarios son representados como clientes y denominados como hilos. La red de filas de espera de software consiste de dos tipos de recursos de software: recursos de demora y recursos de filas de espera. El recurso de demora corresponde con el código de sección no crítica (NCS). Como en la sección no crítica, no existe contención ni suceden colas antes de que se ejecute el módulo de software. Los recursos de filas de espera corresponden con las secciones críticas (CS) que son delimitadas por construcciones de sincronización. La aplicación multi-hilos puede tener cualquier número de secciones críticas. La SQN puede consistir de una sección no crítica y un número de secciones críticas.

Mientras un hilo (es decir el usuario) está utilizando un recurso de software, utiliza también recursos físicos tales como CPU y discos. La red de filas de espera asociada con los recursos físicos es denominada red de filas de espera de hardware (HQN). Así los usuarios en la HQN son usuarios que están utilizando los recursos físicos debido a la ejecución de módulos de software.

La fig. 4 ilustra la red de filas de espera de software que consiste de una sección o crítica y dos componentes de sección crítica. Todos los módulos de software son representados utilizando rectángulos mientras que todos los servidores son representados utilizando círculos. En la HQN, los usuarios pueden acceder a un servidor de CPU y cualquiera de los dos servidores de disco. Las dos capas de la red de filas de espera conducen a las siguientes observaciones importantes:

1. El tiempo gastado en las secciones no críticas y crítica depende de la contención en los recursos físicos, por ejemplo, CPU y disco.

2. El número de usuarios en la red de filas de espera de hardware, que luchan por los recursos físicos, es igual al número de hilos concurrentes que no están bloqueados esperando a entrar en una sección crítica. Como los hilos bloqueados están en reposo, no están presentes en la cola HQN.

Con referencia a una la fig. 4, con el fin de reducir la complejidad, se han hecho presunciones. A modo de un ejemplo no limitativo, la CPU de recursos de hardware y disco son considerados con la asunción de que están dentro de las secciones críticas, solamente las operaciones de cálculo y IO son realizadas y no tiene lugar comunicación de mensajes.

El módulo 112 de computación es configurado para calcular unas métricas de rendimiento para la aplicación multi-hilos resolviendo la red de filas de espera de software y la red de filas de espera de hardware utilizando una técnica iterativa con una solicitud de servicio predeterminada. Las métricas de rendimiento comprenden uno o más parámetros. Los uno o más parámetros predicen el rendimiento de la aplicación multi-hilos en presencia de cuellos de botella de recursos.

El módulo 112 de computación utiliza la solicitud de servicio predeterminada para desarrollar una relación entre la red de filas de espera de software y la red de filas de espera de hardware. La relación comprende la utilización o compartición de los recursos de software con respecto a los recursos de hardware con respecto al valor predeterminado de la solicitud de servicio.

El módulo 112 de computación ejecuta un teorema de análisis de valor medio (metodología) como la técnica iterativa sobre la solicitud de servicio predeterminada. El teorema de análisis de valor medio es utilizado para derivar las características de rendimiento en forma de las métricas de rendimiento según la relación entre la red de filas de espera de hardware y la red de filas de espera de software.

El módulo 112 de computación ejecuta el teorema de análisis de valor medio para estimar el rendimiento de la aplicación multi-hilos en presencia de N usuarios concurrentes.

Mientras se resuelven las redes de filas de espera, se han hecho las siguientes presunciones:

1. La solicitud de servicio predeterminada de todos los recursos de hardware en cada uno de los niveles de una aplicación multi-hilos es capturada. La solicitud de servicio predeterminada de un recurso de hardware (por ejemplo, CPU) para un módulo de software se refiere a la cantidad de tiempo de CPU requerido para ejecutar el módulo específico.

2. La solicitud de servicio predeterminada de todos los eventos de sincronización es capturada. Esto puede ser conseguido a través de la instrumentación de las creaciones de sincronización utilizadas por la aplicación. Los eventos de Sincronización incluyen métodos sincronizados, bloque de código con palabras clave sincronizadas, barreras, que adquieren seguido por la liberación de un bloqueo, variables condicionales. Un método que pertenece a una clase puede ser declarado sincronizado. De manera similar un bloque de código puede ser prefijado por una palabra clave sincronizada para marcar como sección sincronizada. Para una barrera, si un hilo llega a la barrera, esperará hasta que todos los hilos lleguen a ese punto de sincronización. En caso de una variable de condición, un hilo espera hasta que otro hilo llama a la señal sobre la misma variable de condición.

Esto supone además que sólo hay dos módulos en la capa de software, una sección crítica y una sección no crítica. Las solicitudes de CPU de estas secciones son indicadas por $D_{CS,CPU}$ y $D_{NCS,CPU}$ respectivamente. La solicitud de servicio de CPU total es obtenida añadiendo estos dos valores. Esta solicitud es utilizada en la red de filas de espera de hardware.

a. $D_{CPU} = D_{CS,CPU} + D_{NCS,CPU}$ (1a)

En una máquina multi-núcleo con C núcleos, la sección crítica puede ejecutarse solamente sobre un núcleo a la vez mientras que la sección no crítica puede ejecutarse sobre varios núcleos. Por tanto, la solicitud de CPU para la red de filas de espera de hardware es modificada como sigue:

a. $D_{CPU} = D_{CS,CPU} + D_{NCS,CPU/C}$ (1b)

Además, durante la ejecución en la sección crítica, un hilo puede acceder también al disco para hacer alguna operación de lectura y escritura. Así, la solicitud para la sección crítica y la sección no crítica incluye tanto solicitud debida a CPU, como a disco. La solicitud de recursos total para la sección crítica es indicada por DCS y consiste de la solicitud debida a CPU y a disco.

a. $D_{CS} = D_{CS,CPU} + D_{CS,Disk}$ (2)

Existe una relación similar para la solicitud de servicio de la sección no crítica. El módulo 112 de computación considera también un tiempo de residencia. El tiempo de residencia es el tiempo total gastado por un hilo en un recurso físico y es indicado por R_{CPU} (para la CPU) y R_{Disk} (para el disco).

Los pasos realizados por el módulo de computación mientras que se ejecuta el teorema de análisis del valor medio son explicados a continuación:

El módulo 112 de computación está configurado para utilizar la solicitud de servicio predeterminada de la sección crítica y la sección no crítica y guardar los valores iniciales como $D^i_{CS,CPU}$ y $D^i_{NCS,CPU}$. Indican el número de hilos bloqueados en CS por B^i_{CS} y se inicializan a cero.

El módulo 112 de computación está configurado para resolver la red de filas de espera de software con las solicitudes de servicio predeterminadas de todas las secciones críticas y secciones no críticas. El número de clientes (usuarios o hilos) en la capa de software es el mismo que el número de usuarios en el sistema de acceso.

El módulo 112 de computación está configurado para obtener el número de hilos bloqueados en la sección crítica BCS. Retener este valor para compararlo con el número de hilos bloqueados en la siguiente iteración.

En el siguiente paso, el número de clientes en la red de filas de espera de hardware es tomado como el número de hilos que no están bloqueados en la sección crítica ($N-B_{CS}$). Resolver la red de filas de espera de hardware y obtener el tiempo de residencia de cada uno de los recursos de hardware tales como R_{CPU} y R_{Disk}.

El módulo 112 de computación está configurado para actualizar la solicitud de servicio predeterminada de los módulos de software para reflejar la espera para los recursos de hardware como sigue:

a. $D_{CS,CPU} = R_{CPU} * D^i_{CS,CPU}/(D^i_{CS,CPU} + D^i_{NCS,CPU})$

b. $D_{NCS,CPU} = R_{CPU} * D^i_{NCS,CPU}/(D^i_{CS,CPU} + D^i_{NCS,CPU})$

Así, la solicitud de servicio de la sección crítica es ajustada de forma iterativa para tener en cuenta la contención de recursos en los recursos de hardware.

El módulo de computación comprueba si $D_{CS,CPU} < D^i_{CS,CPU}$. Este paso impide que la solicitud de sección crítica sea inferior que la solicitud inicial. Si es cierto, entonces se obtiene la solución final.

El módulo 112 de computación encuentra una diferencia entre el número de hilos bloqueados en la sección crítica en esta iteración BCS y en la iteración previa B^i_{CS}. Si la diferencia entre B_{CS} y B^i_{CS} es menor que un límite específico (o un valor predeterminado) ($|B_{CS} - B^i_{CS}|<\varepsilon$) donde ε es un número pequeño o $B^i_{CS} > B_{CS}$, a continuación se obtiene la solución final; de otro modo asigna $B^i_{CS} = B_{CS}$, y vuelve atrás al Paso 2 es decir otra vez a resolver la red de filas de espera de software con las solicitudes de servicio predeterminadas de todas las secciones críticas y las secciones no críticas. Épsilon (el valor predeterminado) incluye un número decimal pequeño menor que 1. Épsilon indica que el error en el número de hilos bloqueados se ha estabilizado o ha convergido sobre un número de iteraciones. El valor de épsilon puede ser tomado como 0,005.

El módulo 112 de computación calcula un valor X_{EST} de rendimiento final de la red de filas de espera de software y de la red de filas de espera de hardware. El tiempo de respuesta medio del sistema individual es obtenido utilizando la ley de Little. La ley de Little dice que si N es el número de clientes en los sistemas y X es el rendimiento, R es el tiempo de respuesta y Z es el tiempo de pensar utilizado para las solicitudes, entonces existe la siguiente relación entre estos parámetros.

$$N = X * (R+Z)$$

A continuación se estima la utilización de la CPU, el disco y cualquier otro de los recursos de hardware. Para una máquina de múltiples núcleos, la utilización de la CPU media por un núcleo es obtenida desde el rendimiento estimado y la solicitud de servicio de CPU total como se ha dado en la ecuación (1a).

$$U_{CPU} = (D_{CPU} \times X_{EST})/C \quad (3)$$

Dada la solicitud de servicio predeterminada de todas las secciones críticas y la sección no crítica, el algoritmo del análisis del valor medio anterior es utilizado para obtener el rendimiento del sistema hasta que el rendimiento se

satura.

El sistema 102 es configurado para validar con solicitudes de múltiples clases donde las solicitudes de usuarios pertenecen a múltiples clases y cada una de las clases de solicitudes tiene su propio módulo de software que puede contener una sección crítica. En la predicción de rendimientos de múltiples clases, dos hilos pueden estar simultáneamente dentro de sus secciones críticas. Además, las secciones críticas que pertenecen a diferentes clases pueden tener diferentes solicitudes de servicios. Además en una máquina de múltiples núcleos, las secciones no críticas de todas las clases pueden ejecutarse simultáneamente sobre diferentes núcleos.

El sistema 102 es configurado para predecir el rendimiento de la aplicación multi-hilos donde se utiliza la agrupación de recursos. La agrupación de recursos se refiere a casos donde un recurso compartido es un grupo de recursos. Ejemplos no limitativos son un grupo de conexión de DB o un grupo de hilos. El grupo de hilos se refiere a un número de hilos creados por un servidor de aplicación para manejar solicitudes de usuarios concurrentes. En un instante un único hilo sirve exclusivamente a un usuario. De manera similar el grupo de conexión de base de datos se refiere a una memoria caché de conexiones de base de datos creada para ejecutar comandos sobre DB de manera más eficiente.

El tamaño del grupo indica el número máximo de recursos compartidos que pueden ser utilizados en un momento dado. Por lo tanto, el número máximo de hilos que puede utilizar el recurso es igual al tamaño del grupo. Mientras se configura el sistema para la predicción de rendimiento de la aplicación multi-hilos para recursos agrupados, el grupo de recursos es modelado como un multi-servidor en la red de filas de espera de software y la solicitud de recurso corresponde a la de una sección crítica de un único hilo.

El sistema 102 proporciona una solución para predecir el rendimiento de la aplicación multi-hilos en presencia de la aplicación calculando unas métricas de rendimiento. La solución en términos de métricas de rendimiento converge. Las tres condiciones para las cuales la solución proporcionada por el sistema 102 puede converger son:

1. La solicitud de servicio predeterminada de la sección crítica no puede ser menor que la solicitud de servicio inicial antes de pasar a través de los pasos iterativos.

2. Después de unas pocas iteraciones iniciales, el número de hilos bloqueados puede no reducirse a partir de las iteraciones previas.

3. La diferencia en el número de hilos bloqueados entre una nueva iteración y una iteración previa es menor que un número pequeño.

Si se satisfacen estas condiciones, el módulo 112 de computación no puede realizar los pasos iterativos y puede realizar un paso final de cálculo de métricas de rendimiento.

El sistema 102 está configurado para predecir el rendimiento del programa de java multi-hilos utilizado como la aplicación multi-hilos con secciones críticas. Se asume que el programa de java tiene una única sección crítica y una única sección no crítica. La solicitud de servicio predeterminada ha sido variada para cada sección y se observa la escalabilidad de la aplicación multi-hilos con el número de usuarios.

Los resultados obtenidos para distintas configuraciones de servicios están recogidos a continuación:

Tabla 1 Categorías de servidor para aplicación de muestreo

Categoría de servidor	Características
Servicios de rango elevado	8 núcleos CPU 2,66 GHz Xeon con caché L2 de 1Mb, RAM física de 8 Gb
Servicios de rango medio	Cuatro núcleos AMD Opteron CPU 2,19 GHz con caché L2 de 2MB, RAM de 4 GB
Servicios de rango bajo	Intel ® Doble Núcleo CPU 2,33 GHz con Caché de 4MB, RAM de 2 GB

La predicción de rendimiento implicaba módulos de software que hacen uso principalmente de la CPU como recurso de hardware lo que significa que los módulos de software realizan el cálculo y no son realizadas actividades de disco o de red. Se consideran tres diferentes combinaciones de secciones crítica y no crítica

Escenario 1: La sección crítica y la sección no crítica están realizando operaciones similares de tal manera que sus solicitudes de servicio (solicitud de servicio predeterminada) son casi las mismas.

Escenario 2: La sección crítica realiza dos veces la cantidad de operaciones comparado con la sección no crítica. Por lo tanto la solicitud de servicio de la sección crítica es más de dos veces la de la solicitud de servicio de la sección no crítica.

Escenario 3: La solicitud de servicio de la sección no crítica es el doble de la cantidad que la solicitud de servicio de la sección crítica.

En cada uno de los escenarios listados anteriormente, el programa de java es accedido por usuarios concurrentes y el rendimiento del programa es observado cuando la concurrencia es variada. El rendimiento es expresado como el número de iteraciones completadas por unidad de tiempo (por segundo). En iteración, el programa ejecuta una vez la sección crítica y la sección no crítica. En cada uno de los casos, la CPU es el único recurso de hardware que es considerado. Durante el intervalo de prueba, se miden el rendimiento medio, el tiempo de respuesta de la aplicación y la utilización de la CPU. El rendimiento es medido durante todo el intervalo de prueba y el tiempo de respuesta es tomado como el tiempo medio tomado para completar una iteración del programa que consiste de una sección crítica y una sección no crítica.

A modo de ejemplo no limitativo, la fig. 5 ilustra la predicción de rendimiento es explicada para un caso donde las secciones crítica y no críticas son del mismo tamaño. El módulo de computación calcula unas métricas de rendimiento calculando un valor de rendimiento. El valor de rendimiento se satura debido al cuello de botella de la sección crítica a 2000 usuarios y la utilización de la CPU media no va más allá del 67%. A 3000 usuarios, el valor de rendimiento predicho así es 1327 iteraciones por segundo mientras que el valor de rendimiento real es 1360 iteraciones por segundo.

A modo de otro ejemplo no limitativo, la fig. 6 ilustra la predicción de rendimiento de la aplicación multi-hilos donde las secciones críticas y no críticas son de diferentes tamaños. Aquí, los hilos (usuarios o clientes) gastan diferente cantidad de tiempo en la sección crítica y en la sección no crítica. Considerando el escenario 3 (como se ha descrito anteriormente), sobre un servidor de rango medio donde la solicitud de servicio de la sección no crítica es dos veces el de la sección crítica. La fig. 6 ilustra el valor de rendimiento predicho y real de la aplicación multi-hilos y la utilización de la CPU.

Se observa que el rendimiento predicho está dentro de una imprecisión del 10-15% del rendimiento real. Para 2000 usuarios, el rendimiento se satura a 890 iteraciones/s y la utilización de la CPU es del 73%. Los valores predichos son 825 iteraciones/s y del 67%. La diferencia en los valores observados y predichos es atribuida a la solicitud de servicio elevada estimada utilizando una única prueba de usuario de la aplicación. El sistema es configurado también para predecir el rendimiento de la aplicación multi-hilos en situaciones donde las operaciones dentro de la sección crítica no solamente utilizan la CPU sino también realizan operaciones de I/O sobre el disco.

A modo de otro ejemplo no limitativo, la fig. 7 ilustra la predicción de rendimiento de la aplicación multi-hilos donde las secciones críticas tienen solicitud de múltiples clases. En este caso, las solicitudes procedentes de usuarios pertenecen a múltiples clases y cada clase de solicitud tiene su propio módulo de software que puede contener una sección crítica. En este caso solamente dos clases de solicitudes son consideradas y las secciones críticas que pertenecen a diferentes clases tienen diferentes solicitudes. La fig. 7 ilustra el rendimiento predicho para clases individuales y la utilización de CPU total desde diferentes solicitudes.

En este escenario, ambas solicitudes utilizan el mismo número de usuarios. Por ejemplo, cuando 500 usuarios están accediendo a la solicitud de clase 1, otros 500 usuarios están accediendo a la solicitud de clase 2. La sección crítica CS1 de clase 1 tiene solicitud de servicio de 0,50 ms y la sección crítica CS2 de clase 2 tiene solicitud de servicio de 1,9 ms. Ya que la sección crítica de clase 2 tiene solicitud más elevada, alcanza la saturación más pronto a 500 usuarios mientras que la sección crítica de clase 1 continúa para tener un rendimiento más elevado hasta 1200 usuarios. A partir de la fig. 7, puede verificarse que en el escenario de múltiples clases también, el sistema 102 es capaz de predecir el rendimiento para clases individuales y la utilización total de la CPU. Sin embargo, el rendimiento predicho y la utilización para la clase 2 son aproximadamente el 10% más elevados que el rendimiento y la utilización reales. Esto es debido a una mayor solicitud de servicio estimada utilizando la rutina de temporizador alrededor de una sección crítica. El sistema 102 se atribuye a la solicitud de servicio más elevada calculada para secciones críticas. La solicitud de servicios obtenida a través de la instrumentación del código de programa correspondiente con el bloque de sincronización. Uno o más módulos capturan el tiempo de inicio y el tiempo de fin del bloque de sincronización para obtener la solicitud de servicios.

A modo de otro ejemplo no limitativo, la fig. 8 ilustra la predicción de rendimiento de la aplicación multi-hilos donde las secciones críticas tienen agrupación de recursos. El tamaño del grupo de recursos de 50 y 100 son utilizados y el número de usuarios es variado. La solicitud de servicios para la agrupación de recursos es tomada como 13 ms que es relativamente más elevado comparado con los experimentos anteriores. Esto es para restringir el número de usuarios, se incrementa.

La fig. 8 muestra el rendimiento total de la agrupación de recursos completa y por utilización de núcleo de CPU cuando el número de usuarios es variado sobre un servidor de rango medio. Puede observarse que el rendimiento y la utilización predichos por el sistema 102 indican los valores reales. Debido a múltiples recursos en la agrupación, el rendimiento es encontrado mucho más elevado en este caso y la aplicación aumenta también a un número más elevado de usuarios.

Además, el método puede ser implementado en cualquier hardware, software, firmware o combinación de los

mismos adecuada. Sin embargo, para facilitar la explicación, en las realizaciones descritas a continuación, el método 300 puede ser considerado para ser implementado en el sistema 102 descrito anteriormente.

En el bloque 302, una o más redes de filas de espera son representadas para recursos empleados para ejecutar la aplicación multi-hilos para detectar un nivel de concurrencia en el cual los cuellos de botella de recursos son encontrados mientras se accede a la aplicación de software multi-hilos.

En el bloque 304, unas métricas de rendimiento son calculadas para la aplicación de software multi-hilos. Las métricas de rendimiento comprenden uno o más parámetros.

En el bloque 306, utilizando una tecnología iterativa con un valor predeterminado de solicitud de servicio para identificar una relación entre la red de filas de espera de software y la red de filas de espera de hardware.

En el bloque 308, el valor de uno o más parámetros es determinado en forma de un valor de rendimiento y un valor de tiempo de respuesta para la aplicación de software multi-hilos, determinando por tanto el rendimiento de la aplicación de software multi-hilos.

REIVINDICACIONES

1.- Un sistema para determinar el rendimiento de una aplicación de software multi-hilos en presencia de uno o más cuellos de botella de recursos, comprendiendo sistema:

un procesador (202); y

una memoria (208) acoplada al procesador (202), en donde el procesador (202) es capaz de ejecutar una pluralidad de módulos almacenados en la memoria (208), y en donde la pluralidad de módulos comprende:

un módulo (212) de representación configurado para representar una red de filas de espera de hardware y una red de filas de espera de software para recursos empleados para ejecutar la aplicación de software multi-hilos, en donde las redes de filas de espera representa la contención para los recursos, y en donde los recursos comprenden recursos de hardware y recursos de software, en donde las redes de filas de espera son utilizadas para comprobar el número de hilos que acceden a la aplicación de software multi-hilos cuando se encuentra un cuello de botella de recursos;

un módulo (214) de computación configurado para ejecutar una técnica iterativa con un valor predeterminado de la solicitud de servicio para desarrollar una relación entre la red de filas de espera de software y la red de filas de espera de hardware, en donde la relación comprende la utilización de los recursos de software con respecto a los recursos de hardware basándose en el valor predeterminado de la solicitud de servicio, y en donde la ejecución de la técnica iterativa comprende:

obtener un número de hilos bloqueados en una sección crítica de los recursos de software basándose en el valor predeterminado de la solicitud de servicio, en donde los hilos bloqueados en la sección crítica son hilos que esperan entrar en la sección crítica;

obtener un tiempo de residencia en cada uno de los recursos de hardware basándose en el valor predeterminado de la solicitud de servicio;

actualizar una solicitud de servicio de sección crítica de una interacción actual para una solicitud de servicio de sección crítica actualizada para reflejar una espera para los recursos de hardware utilizando el tiempo de residencia, en donde la solicitud de servicio de sección crítica es actualizada de forma iterativa para tener en cuenta la contención de recursos en los recursos de hardware basándose en el tiempo de residencia;

comparar el número de hilos bloqueados de la iteración actual con el número de hilos bloqueados de una iteración previa con el fin de comprobar si el número de hilos bloqueados de la interacción actual es más elevado que el número de hilos bloqueados de la iteración previa, y comprobar si la solicitud de servicio de sección crítica actualizada es más elevada que la solicitud de servicio de sección crítica de la iteración previa; e

identificar una diferencia entre el número de hilos bloqueados en una sección crítica para la iteración actual y el número de hilos bloqueados en la sección crítica en la iteración previa; y

ejecutar repetidamente la técnica iterativa 1) si la diferencia en el número de hilos bloqueados es mayor que un límite predefinido, 2) si el número de hilos bloqueados de la iteración actual es más elevado que el número de hilos bloqueados de la iteración previa y 3) si la solicitud de servicio de sección crítica actualizada es mayor que la solicitud de servicio de sección crítica de la iteración previa; de otro modo

obtener las métricas de rendimiento con la solicitud de servicio de sección crítica actualizada para la aplicación de software multi-hilos cuando la diferencia en el número de hilos bloqueados es inferior que el límite predefinido, en donde el límite predefinido indica que la diferencia en el número de hilos bloqueados ha convergido sobre un número de iteraciones de la técnica iterativa, determinando por tanto un rendimiento de la aplicación de software multi-hilos.

2.- El sistema de la reivindicación 1, en donde los recursos de software comprenden módulos de software y los recursos de hardware comprenden una CPU (Unidad de Tratamiento Central), una memoria, un disco o una combinación de los mismos.

3.- El sistema de la reivindicación 1, en donde la red de filas de espera de software representa los módulos de software correspondientes a una o más secciones críticas y a una o más secciones no críticas, y la red de filas de espera de hardware representa la pluralidad de recursos de hardware, y en donde una sección crítica corresponde a un único recurso compartido y una sección no crítica corresponde a una porción del código que puede ser ejecutado simultáneamente por ‘n’ números de hilos.

4.- El sistema de la reivindicación 1, en donde uno o más cuellos de botella de recursos comprenden uno o más cuellos de botella de software, uno o más cuellos de botella de hardware, y uno o más cuellos de botella debido a cualquiera de una agrupación de hilos, una agrupación de conexión de base de datos, bases o bloqueos sobre elementos de datos, o una combinación de los mismos.

5.- El sistema de la reivindicación 1, en donde el valor predeterminado de la solicitud de servicio comprende, una

solicitud de servicio de los recursos de hardware en uno o más niveles de la aplicación de software multi-hilos; y una solicitud de servicio de uno o más eventos de sincronización asociados con la aplicación de software multi-hilos.

6.- El sistema de la reivindicación 3, en donde el módulo de computación calcula las métricas de rendimiento para la aplicación de software para el 'n' números de hilos, en donde 'n' está en un rango de uno a millares.

7.- El sistema de la reivindicación 1, en donde uno o más parámetros de las métricas de rendimiento comprenden un rendimiento, un tiempo de respuesta, una cantidad de tiempo de espera de los hilos, un número de objetos en una fila de espera, y una probabilidad de que una fila de espera se vacíe, una utilización de la CPU media (Unidad de Tratamiento Central) por núcleo a partir del rendimiento y una solicitud de servicio de CPU total.

8.- El sistema de la reivindicación 1, en donde el módulo de computación calcula las métricas de rendimiento cuando la diferencia en el número de hilos bloqueados se estabiliza.

9.- Un método para determinar el rendimiento de una aplicación de software multi-hilos en presencia de uno o más cuellos de botella de recursos, el método comprende:

representar, por un procesador, una red de filas de espera de hardware y una red de filas de espera de software, para recursos empleados para ejecutar la aplicación de software multi-hilos, en donde las redes de filas de espera representan la contención para los recursos, y en donde los recursos comprenden recursos de hardware y recursos de software, en donde las redes de filas de espera son utilizadas para comprobar el número de hilos que acceden a la aplicación de software multi-hilos cuando se encuentra un cuello de botella de recursos;

en donde el procesador es utilizado para desarrollar una relación entre la red de filas de espera de software y la red de filas de espera de hardware ejecutando, una técnica iterativa con un valor predeterminado de la solicitud de servicio, en donde la relación comprende la utilización de los recursos de software con respecto a los recursos de hardware basándose en el valor predeterminado de la solicitud de servicio; y en donde ejecutar la técnica iterativa comprende:

obtener un número de hilos bloqueados en una sección crítica de los recursos de software basándose en el valor predeterminado de la solicitud de servicio, en donde los hilos bloqueados en la sección crítica son hilos que esperan entrar en la sección crítica;

obtener un tiempo de residencia en cada uno de los recursos de hardware basándose en el valor predeterminado de la solicitud de servicio;

actualizar una solicitud de servicio de sección crítica de una iteración actual para una solicitud de servicio de sección crítica actualizada para reflejar una espera para los recursos de hardware utilizando el tiempo de residencia, en donde la solicitud de servicio de sección crítica actual es actualizada de manera iterativa para tener en cuenta la contención de recursos en los recursos de hardware basándose en el tiempo de residencia;

comparar el número de hilos bloqueados de la iteración actual con el número de hilos bloqueados de un iteración previa con el fin de comprobar si el número de hilos bloqueados de la iteración actual es más elevado que el número de hilos bloqueados de la iteración previa, y comprobar si la solicitud de servicio de sección crítica actualizada es mayor que la solicitud de servicio de sección crítica de la iteración previa; e

identificar una diferencia entre el número de hilos bloqueados en una sección crítica para la iteración actual y el número de hilos bloqueados en la sección crítica en la iteración previa; y

ejecutar repetidamente la técnica iterativa 1) si la diferencia en el número de hilos bloqueados es más elevada que un límite predeterminado, y 2) si el número de hilos bloqueados de la iteración actual es más elevado que el número de hilos bloqueados de la iteración previa y 3) si la solicitud de servicio de sección crítica actualizada es mayor que la solicitud de servicio de sección crítica de la iteración previa; de otro modo

obtener las métricas de rendimiento con la solicitud de servicio de sección crítica actualizada para la aplicación de software multi-hilos cuando la diferencia en el número de hilos bloqueados es inferior que el límite predeterminado, en donde el límite predeterminado indica que la diferencia en el número de hilos bloqueados ha convergido sobre un número de iteraciones de la técnica iterativa determinando por ello un rendimiento de la aplicación de software multi-hilos.

10.- El método de la reivindicación 9, en donde el valor predeterminado de la solicitud de servicio comprende una solicitud de servicio de todos los recursos de hardware a cada uno de los niveles de la aplicación de software; y una solicitud de servicio de todos los eventos de sincronización.

11.- Un producto de programa informático que tiene incorporado en él un programa informático para determinar el rendimiento de una aplicación de software multi-hilos en presencia de uno o más cuellos de botella de recursos, comprendiendo el producto de programa informático un código de programa para realizar un método según la reivindicación 9.

12.- El producto de programa informático de la reivindicación 11, en donde la red de filas de espera de hardware representa módulos de software correspondientes con una o más secciones críticas y una o más secciones no críticas, y la red de filas de espera de hardware representa la pluralidad de recursos de hardware, y en donde una sección crítica corresponde con un único recurso compartido y una sección no critica corresponde con una porción del código que puede ser ejecutada simultáneamente por 'n' número de hilos.

13.- El método de la reivindicación 9, en donde uno o más parámetros de las métricas de rendimiento comprenden un rendimiento, un tiempo de respuesta, una cantidad de tiempo de espera de los hilos, un número de objetos en una fila de espera, y una probabilidad de que una fila de espera se vacíe.

14.- El método de la reivindicación 9, en donde la técnica iterativa es un método de análisis de valor medio.

15.- El método de la reivindicación 9, en donde la red de filas de espera de software representa módulos de software correspondientes con una o más secciones críticas y una o más secciones no críticas, y la red de filas de espera de hardware representa la pluralidad de recursos de hardware, y en donde una sección crítica corresponde con un único recurso compartido y una sección no crítica corresponde con una porción de código que puede ser ejecutado simultáneamente por 'n' número de hilos.

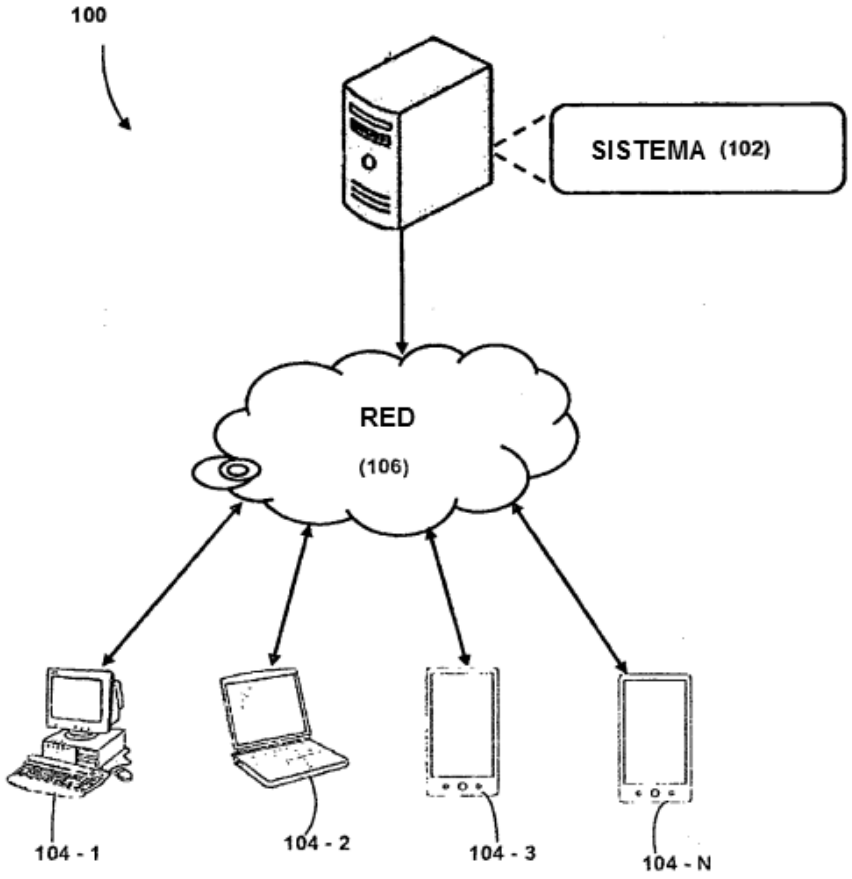

Figura 1

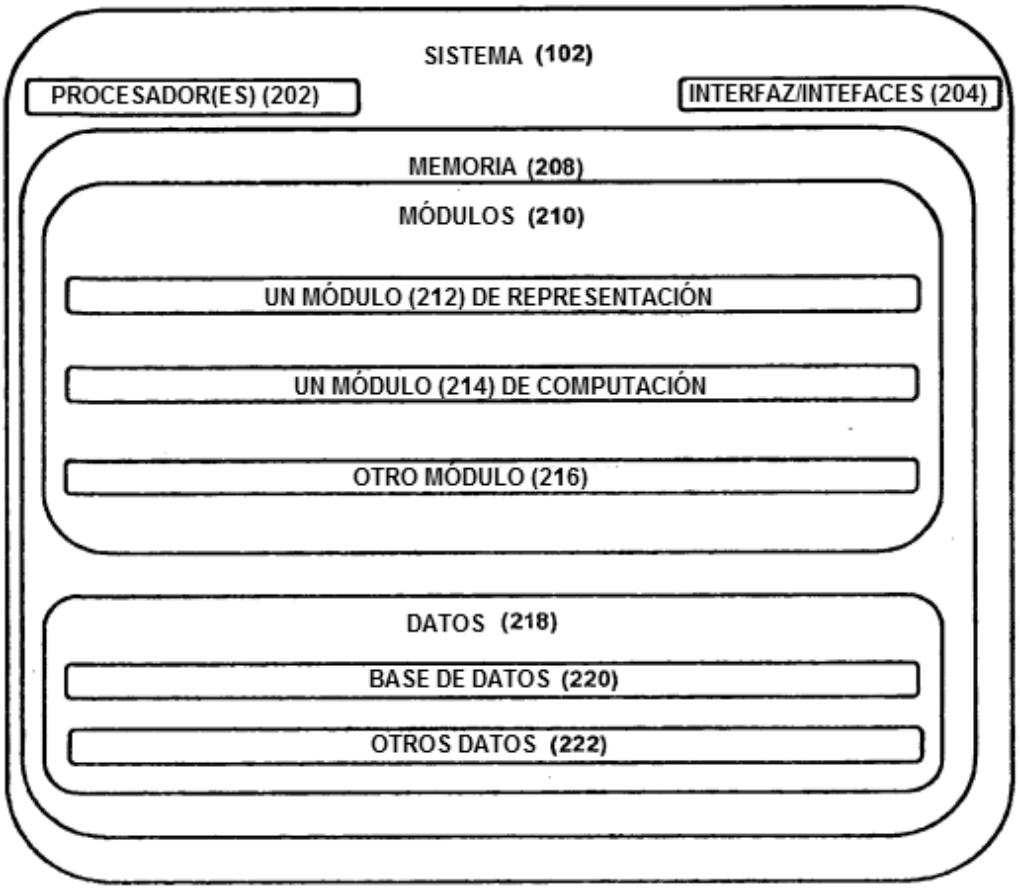
SISTEMA (102)
PROCESADOR(ES) (202)
INTERFAZ/INTEFACES (204)
MEMORIA (208)
MÓDULOS (210)
UN MÓDULO (212) DE REPRESENTACIÓN
UN MÓDULO (214) DE COMPUTACIÓN
OTRO MÓDULO (216)
DATOS (218)
BASE DE DATOS (220)
OTROS DATOS (222)

Figura 2

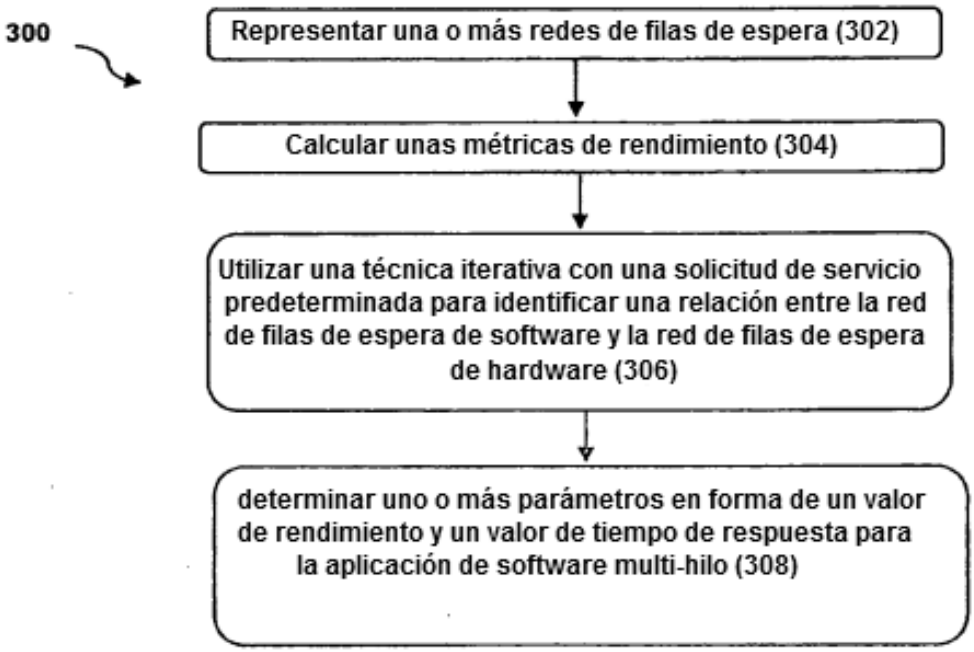
300
Representar una o más redes de filas de espera (302)
Calcular unas métricas de rendimiento (304)
Utilizar una técnica iterativa con una solicitud de servicio predeterminada para identificar una relación entre la red de filas de espera de software y la red de filas de espera de hardware (306)
determinar uno o más parámetros en forma de un valor de rendimiento y un valor de tiempo de respuesta para la aplicación de software multi-hilo (308)

Figura 3

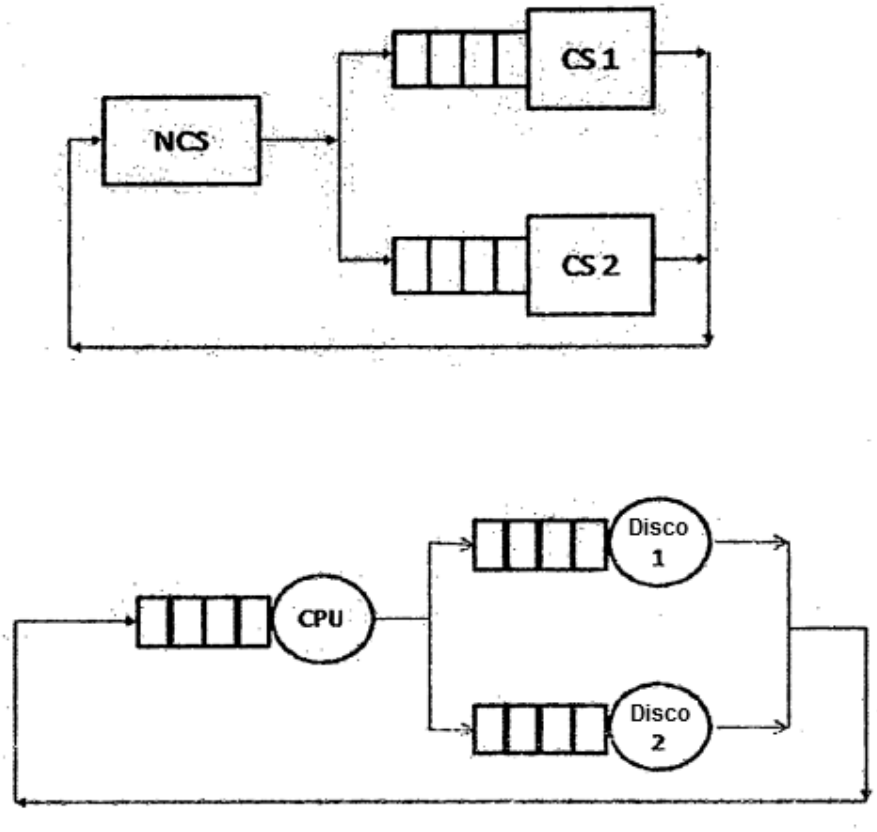

Figura 4

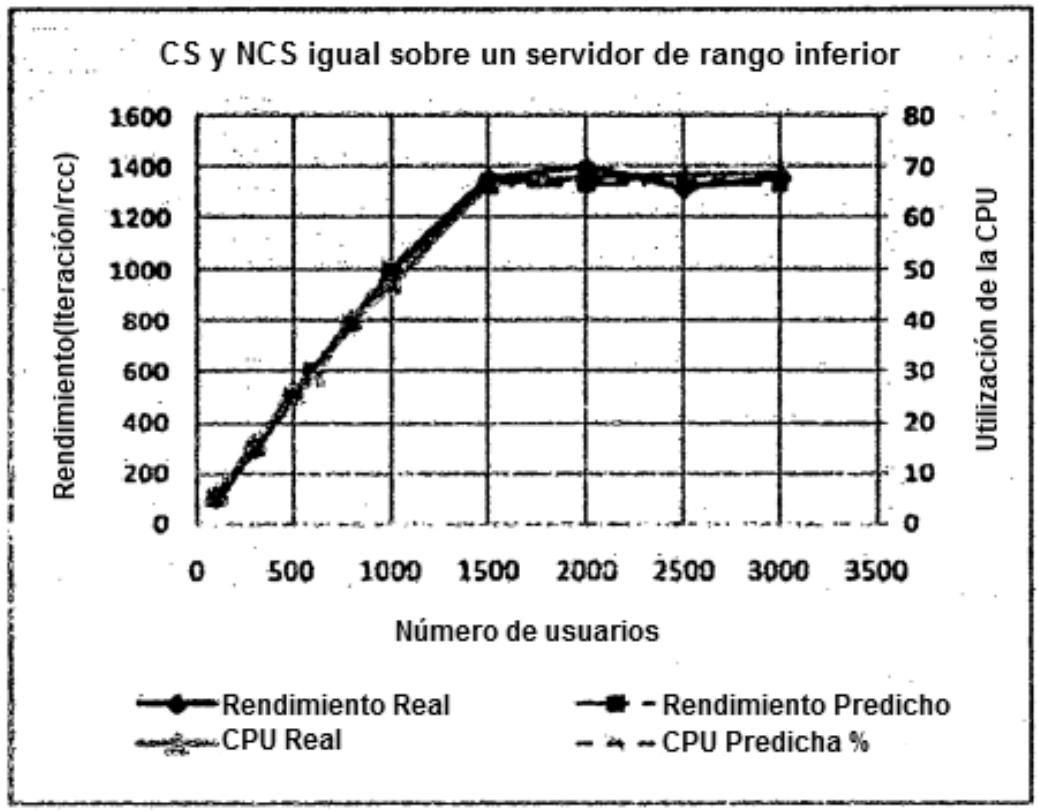
CS y NCS igual sobre un servidor de rango inferior
Rendimiento(Iteración/rcc)
1600
1400
1200
1000
800
600
400
200
0
0
500
1000
1500
2000
2500
3000
3500
Número de usuarios
80
70
60
50
40
30
20
10
0
Utilización de la CPU
Rendimiento Real
Rendimiento Predicho
CPU Real
CPU Predicha %

Figura 5

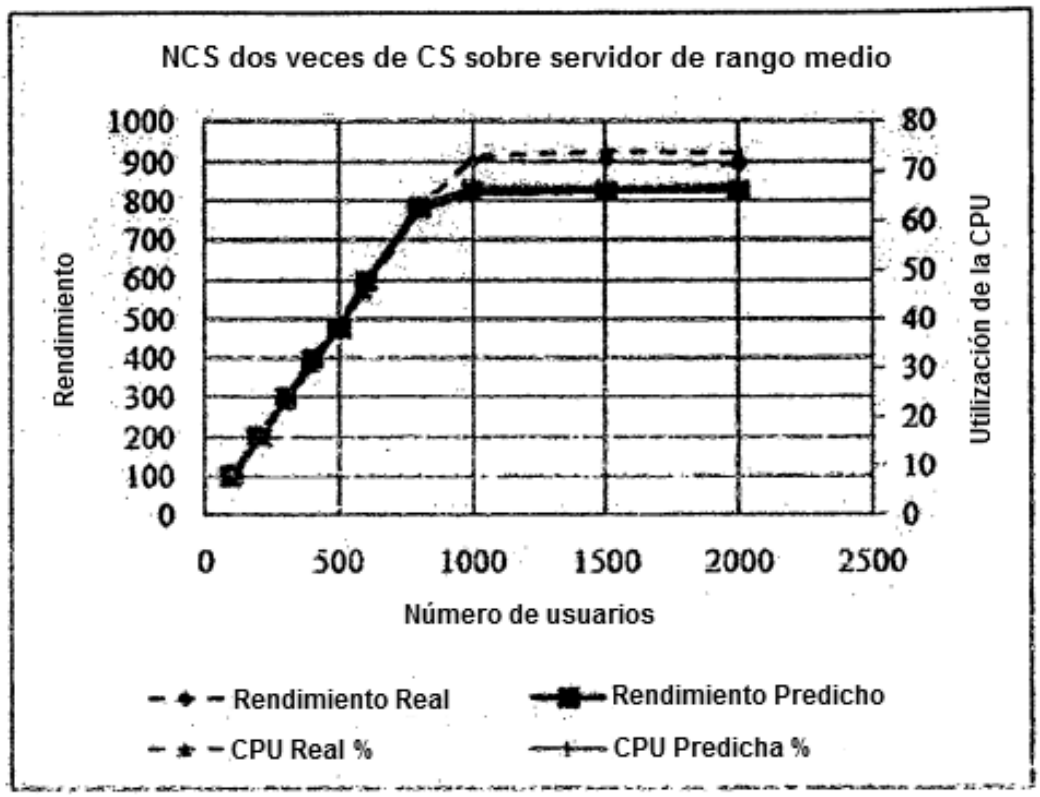
NCS dos veces de CS sobre servidor de rango medio
Rendimiento
Utilización de la CPU
1000
900
800
700
600
500
400
300
200
100
0
80
70
60
50
40
30
20
10
0
0
500
1000
1500
2000
2500
Número de usuarios
Rendimiento Real
Rendimiento Predicho
CPU Real %
CPU Predicha %

Figura 6

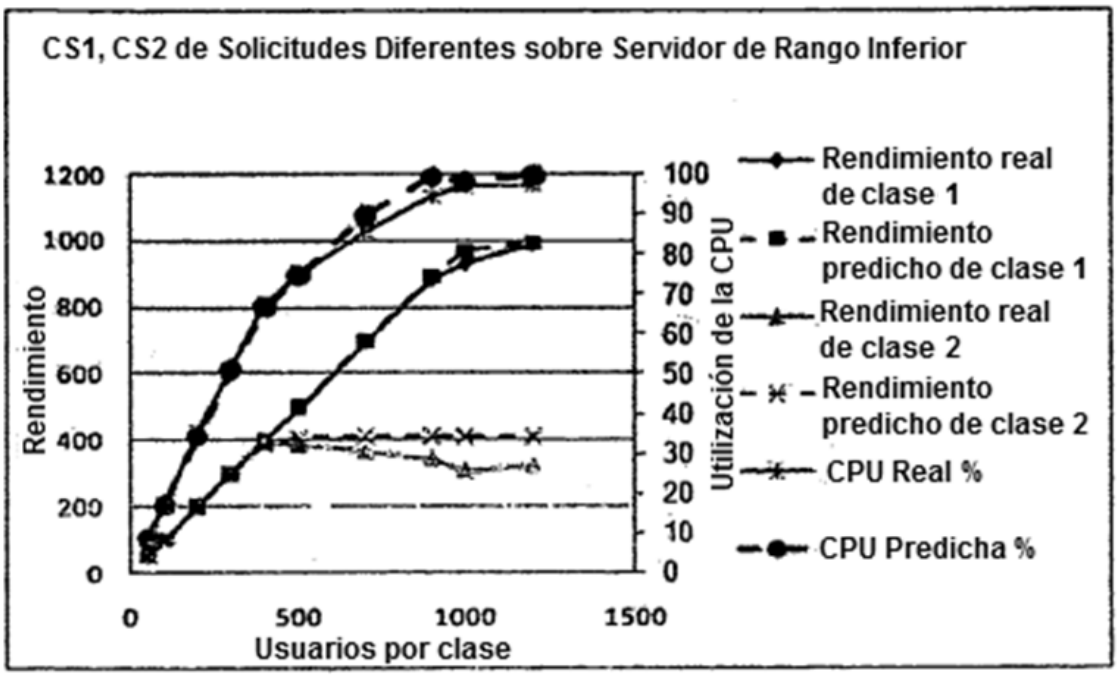
CS1, CS2 de Solicitudes Diferentes sobre Servidor de Rango Inferior
Rendimiento
1200
1000
800
600
400
200
0
0
500
1000
1500
Usuarios por clase
Utilización de la CPU
100
90
80
70
60
50
40
30
20
10
0
Rendimiento real de clase 1
Rendimiento predicho de clase 1
Rendimiento real de clase 2
Rendimiento predicho de clase 2
CPU Real %
CPU Predicha %

Figura 7

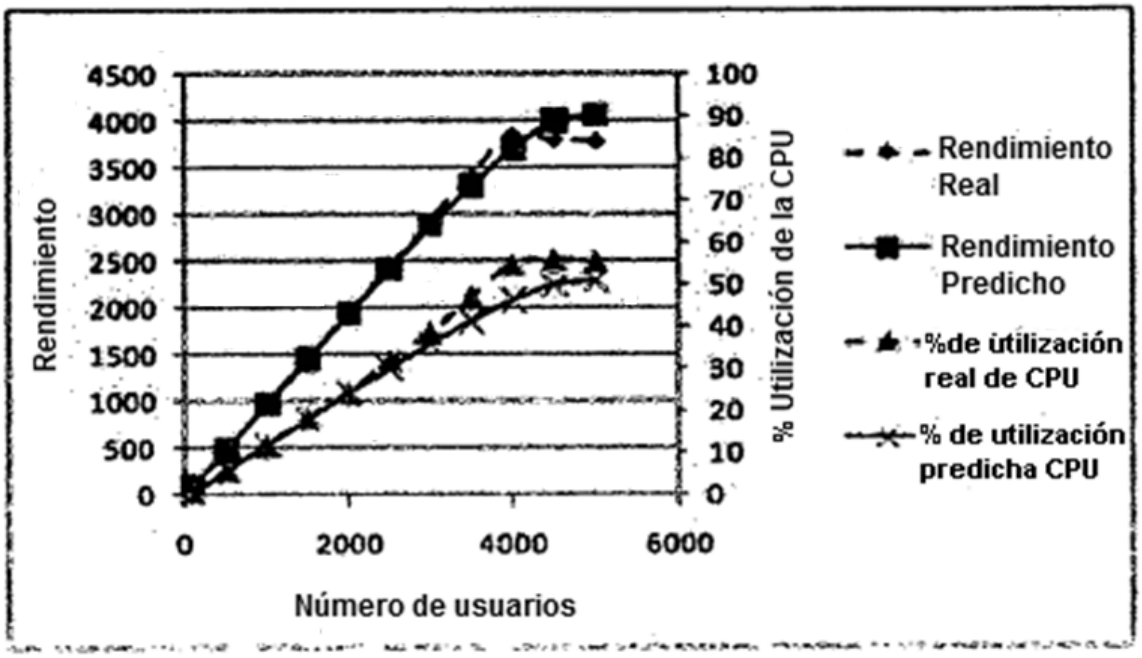
Rendimiento
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
Número de usuarios
100
90
80
70
60
50
40
30
20
10
0
% Utilización de la CPU
Rendimiento Real
Rendimiento Predicho
%de utilización real de CPU
% de utilización predicha CPU

Figura 8