

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 368 684**

51 Int. Cl.:

G06F 9/50 (2006.01)

G06F 9/455 (2006.01)

G06F 9/30 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **09701347 .8**

96 Fecha de presentación: **12.01.2009**

97 Número de publicación de la solicitud: **2223214**

97 Fecha de publicación de la solicitud: **01.09.2010**

54 Título: **DESCUBRIMIENTO DE TOPOLOGÍA VIRTUAL PARA CONFIGURACIÓN DE ORDENADOR.**

30 Prioridad:
11.01.2008 US 972802

45 Fecha de publicación de la mención BOPI:
21.11.2011

45 Fecha de la publicación del folleto de la patente:
21.11.2011

73 Titular/es:
**INTERNATIONAL BUSINESS MACHINES
CORPORATION
NEW ORCHARD ROAD
ARMONK, NY 10504, US**

72 Inventor/es:
**GAINEY JR., Charles;
FARRELL, Mark;
KUBALA, Jeffrey y
SCHMIDT, Donald**

74 Agente: **de Elzaburu Márquez, Alberto**

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

ES 2 368 684 T3

DESCRIPCIÓN

Descubrimiento de topología virtual para configuración de ordenador.

CAMPO DEL INVENTO

- 5 El presente invento se refiere en general a la virtualización de sistemas con múltiples procesadores. En particular, el presente invento se refiere a habilitar programas para descubrir la topología de su entorno virtual.

ANTECEDENTES

- 10 Entre las funciones de control del sistema está la capacidad para dividir el sistema en varias particiones lógicas (LPAR). Una LPAR es un subconjunto del hardware del procesador que está definido para soportar un sistema operativo. Una LPAR contiene recursos (procesadores, memoria y dispositivos de entrada/salida) y funciona como un sistema independiente. Pueden existir múltiples particiones lógicas dentro de un sistema de hardware de computadora central.

En los sistemas de ordenador de computadora central de IBM incluido el S/390®, durante muchos años hubo un límite de 15 LPAR. Las máquinas más recientes tienen 30 (y potencialmente más). Tales máquinas están ejemplificadas por las de la z/Architecture® (Arquitectura z). La z/Architecture® de IBM está descrita en los Principios de Funcionamiento de la z/Architecture SA22-7832-05 publicados en Abril del 2007 por IBM.

- 15 Las limitaciones prácticas del tamaño de la memoria, la disponibilidad de I/O, y la potencia de tratamiento o procesado disponible limitan usualmente el número de LPAR a menos de estos máximos.

- 20 El hardware y el firmware (soporte lógico inalterable) que proporciona la división en particiones es conocido como PR/SM™ (Recursos del Procesador/Gestor del Sistema). Son las funciones de PR/SM las que son usadas para crear y ejecutar las LPAR. La diferencia entre PR/SM (un equipo o dispositivo integrado) y las LPAR (el resultado de usar PR/SM) es a menudo ignorada y el término LPAR es usado colectivamente para el equipo y sus resultados.

- 25 Los administradores de sistemas asignan partes de memoria a cada LPAR y la memoria no puede ser compartida entre las LPAR. Los administradores pueden asignar procesadores (también conocidos como CP o CPU) para especificar las LPAR o pueden permitir que los controladores del sistema distribuyan alguno o la totalidad de los procesadores a todas las LPAR usando un algoritmo de equilibrado de carga interno. Los canales (CHPID) pueden ser asignados a LPAR específicas o pueden ser compartidos por múltiples LPAR, dependiendo de la naturaleza del dispositivo en cada canal.

Un sistema con un único procesador (procesador de CP) puede tener múltiples LPAR. Los PR/SM tienen un distribuidor interno que puede asignar una parte del procesador a cada LPAR, como un distribuidor de sistema operativo asigna una parte de su tiempo de procesador a cada proceso, desarrollo o tarea.

- 30 Las especificaciones de control de división en particiones están parcialmente contenidas en los IOCDS y están parcialmente contenidas en un perfil del sistema. Los IOCDS y el perfil residen ambos en el Elemento de Soporte (SE) que es simplemente un ordenador de bolsillo dentro del sistema. El SE puede estar conectado a una o más Consolas de Gestión de Hardware (HMC), que son ordenadores personales de mesa usados para vigilar y controlar el hardware tales como los microprocesadores de la computadora central. Una HMC es más conveniente de usar que un SE y puede controlar varios ordenadores centrales diferentes.

- 35 Trabajando desde una HMC (o desde un SE, en circunstancias inusuales), un operador prepara una computadora central para su uso seleccionando y cargando un perfil y un IOCDS. Estos crean las LPAR y configuran los canales con números de dispositivo, asignaciones de LPAR, información de trayecto múltiple y así sucesivamente. Esto es conocido como un Reinicio de Encendido (POR). Cargando un perfil y IOCDS diferentes, el operador puede cambiar completamente el número y naturaleza de las LPAR y la apariencia de la configuración I/O. Sin embargo, hacer esto interrumpe usualmente cualesquiera sistemas operativos y aplicaciones en ejecución y es por ello raras veces llevado a cabo sin una planificación previa.

Las particiones lógicas (LPAR) son, en la práctica, equivalentes a ordenadores centrales separados.

- 45 Cada LPAR ejecuta su propio sistema operativo. Este puede ser cualquier sistema operativo de computadora central; no hay necesidad de ejecutar z/OS®, por ejemplo, en cada LPAR. Los planificadores de instalaciones pueden elegir compartir dispositivos de I/O a través de varias LPAR, pero esta es una decisión local.

El administrador de sistema puede asignar uno o más procesadores de sistema para el uso exclusivo de una LPAR. Alternativamente, el administrador puede permitir que todos los procesadores sean usados en alguna o en todas las LPAR. Aquí, las funciones de control del sistema (a menudo conocidas como microcódigo o firmware) proporcionan un distribuidor para compartir los procesadores entre las LPAR seleccionadas. El administrador puede especificar un número

máximo de procesadores concurrentes que ejecutan en cada LPAR. El administrador puede también proporcionar ponderaciones para LPAR diferentes; por ejemplo, especificar que LPAR debería recibir dos veces como mucho el tiempo de procesador de lo que lo recibe LPAR2.

- 5 El sistema operativo en cada LPAR es IPLed separadamente, tiene su propia copia de su sistema operativo, tiene su propia consola de operador (si fuera necesario) y así sucesivamente. Si el sistema en una LPAR se destruye, no hay efecto en las otras LPAR.

- 10 En un sistema de computadora central con tres LPAR, por ejemplo, podría tener una producción z/OS en LPAR1, una versión de prueba de z/OS en LPAR2, y Linux® para S/390 en LPAR3. Si este sistema total tiene 8 GB de memoria, podría tener asignado 4 GB a LPAR1, 1 GB a LPAR2, 1 GB a LPAR3, y tener 2 GB en reserva. Las consolas del sistema operativo para las dos LPAR de z/OS podrían estar en posiciones completamente diferentes.

Para la mayor parte de los propósitos prácticos no hay diferencia entre, por ejemplo, tres ordenadores centrales separados que ejecutan z/OS (y que comparten la mayor parte de su configuración de I/O) y tres LPAR en la misma computadora central haciendo la misma cosa. Con excepciones menores z/OS, los operadores, y las aplicaciones no pueden detectar la diferencia.

- 15 Las diferencias menores incluyen la capacidad de z/OS (si se permitía cuando las LPAR eran definidas) para obtener rendimiento e información de utilización a través del sistema de computadora central completa y para desplazar dinámicamente recursos (procesadores y canales) entre las LPAR para mejorar el rendimiento.

Hoy día los ordenadores centrales de IBM® tienen un complejo de procesador central (CPC), que puede contener varios tipos diferentes de procesadores de z/Architecture® que pueden ser usados con propósitos ligeramente diferentes.

- 20 Varios de estos propósitos están relacionados con el control de coste del software, mientras que otros son más fundamentales. La totalidad de los procesadores en el CPC comienza como unidades procesadoras equivalentes (PU) o máquinas que no han sido caracterizadas para su uso. Cada procesador es caracterizado por IBM durante la instalación o posteriormente. Las caracterizaciones potenciales son:

- Procesador (CP)

- 25 Este tipo de procesador está disponible para un sistema operativo normal y un software de aplicación,

- Procesador de Asistencia de Sistema (SAP)

- 30 Cada computadora central moderna tiene al menos un SAP; los sistemas mayores pueden tener varios. Los SAP ejecutan un código interno para proporcionar el subsistema de I/O. Un SAP, por ejemplo, traslada números de dispositivo y direcciones reales de identificadores de trayecto de canal (CHPID), direcciones de unidad de control, y números de dispositivo. Gestiona múltiples trayectos para controlar unidades y realiza la recuperación de errores para errores temporales. Los sistemas operativos y aplicaciones no pueden detectar los SAP, y los SAP no usan ninguna memoria "normal".

- Equipo integrado para Linux® (IFL)

- 35 Este es un procesador normal con una o dos instrucciones inhabilitadas que son usadas solamente por z/OS®. Linux no usa estas instrucciones y puede ser ejecutado por un IFL. Linux puede ser ejecutado por un CP también. La diferencia es que un IFL no es tenida en cuenta cuando se especifica el número de modelo del sistema. Esto puede suponer una diferencia sustancial en costes de software.

- zAAP

- 40 Este es un procesador con varias funciones inhabilitadas (manejo de interrupción, algunas instrucciones) de tal modo que no puede ser ejecutado todo el sistema operativo en el procesador. Sin embargo, z/OS puede detectar la presencia de procesadores zAAP y los usará para ejecutar código Java™. El mismo código Java puede ser ejecutado sobre un CP estándar. De nuevo, los procesadores zAAP no son tenidos en cuenta cuando se especifica el número de modelo del sistema. Como los IFL, existen solamente para controlar los costes de software.

- zIIP

- 45 El sistema Procesador de Información Integrado z9™ (zIIP) es una máquina especializada para tratar cargas de trabajo de bases de datos elegibles. El zIIP está diseñado para ayudar a reducir los costes de software para cargas de trabajo seleccionadas en la computadora central, tales como inteligencia de negocios (BI), planificación de recursos de empresa (ERT) y gestión de relaciones con clientes (CRM). El zIIP refuerza la misión de la computadora central como el cubo de

datos de la empresa ayudando a hacer un acceso directo a DB2® más efectivo en coste y a reducir la necesidad de múltiples copias de los datos.

- Equipo de Acoplamiento Integrado (ICF)

- 5 Los procesadores ejecutan sólo Código Interno Licenciado. No son visibles los sistemas o aplicaciones operativos normales. Un equipo de acoplamiento es, en efecto una gran memoria de apuntes usada por sistemas múltiples para coordinar trabajos. Los ICF deben ser asignados a los LPAR que entonces resultan equipos de acoplamiento.

Repuesto

- 10 Una PU sin caracterizar funciona como un “repuesto”. Si los controladores del sistema detectan un CP o SAP que falla, puede ser reemplazado con una PU de repuesto. En la mayor parte de los casos esto puede hacerse sin interrupción de ningún sistema, incluso para la aplicación que se está ejecutando en el procesador que falla.

- 15 Además de estas caracterizaciones de procesadores, algunos ordenadores centrales tienen modelos o versiones que están configurados para funcionar más lentos que la velocidad potencial de sus CP. Esto es ampliamente conocido como “cubrir un cambio brusco” aunque IBM prefiere el término configuración de capacidad, o algo similar. Esto se hace utilizando microcódigo para insertar ciclos nulos en la corriente de instrucciones del procesador. El propósito, de nuevo, es controlar los costes de software teniendo el mínimo modelo o versión de computadora central que satisfaga los requisitos de aplicación. Los IFL, SAP, zAAP, zIIP e ICF funcionan siempre a la velocidad total del procesador debido a que estos procesadores “no cuentan” en los cálculos de precio de software.

- 20 El procesador y la CPU pueden referirse bien a la caja del sistema completo, o a uno de los procesadores (CPU) dentro de la caja del sistema. Aunque el significado puede ser claro a partir del contexto de una exposición, incluso los profesionales de ordenadores centrales deben clarificar qué significado de procesador o CPU están usando en una exposición. IBM usa el término complejo de procesador central (CPC) para hacer referencia a la colección física de hardware que incluye el almacenamiento principal, uno o más procesador centrales, temporizadores, y canales. (Algunos programadores de sistemas usan el término complejo de electrónica central (CEC) para referirse a la “caja” de la computadora central, pero el término preferido es CPC).

- 25 Brevemente, todos los procesadores de S/390 o zArchitecture dentro de un CPC son unidades de tratamiento (PU). Cuando IBM entrega el CPC, las PU están caracterizadas como CP (para trabajo normal), Equipo Integrado para Linux (IFL), Equipo de Acoplamiento Integrado (ICF) para configuraciones Sysplex Paralelas, y así sucesivamente.

- 30 Los profesionales de computadora central usan típicamente sistema para indicar la caja de hardware, un entorno de hardware completo (con dispositivos de I/O), o un entorno operativo (con software), dependiendo del contexto. Típicamente usan procesador para significar un solo procesador (CP) dentro del CPC.

- 35 El z/VM® HYPERVISOR™ está diseñado para ayudar a los clientes a extender el valor de negocio de la tecnología de computadora central a través de la empresa integrando aplicaciones y de otros al tiempo que proporciona niveles de disponibilidad, seguridad, y facilidad operativas excepcionales. La tecnología de virtualización z/VM está diseñada para permitir que la posibilidad de que los clientes ejecuten centenares a millares de servidores Linux en una única computadora central que funciona con otros sistemas operativos de System z, tales como z/OS®, o una solución de servidor de empresa sólo con Linux a gran escala. z/VM V5.3 puede también ayudar a mejorar la productividad alojando cargas de trabajo que no son de Linux tales como z/OS, z/VSE, y z/TPF.

- 40 z/VM proporciona a cada usuario un entorno de trabajo individual conocido como una máquina virtual. La máquina virtual simular la existencia de una máquina real dedicada, incluyendo funciones de procesador, memoria, redes, y recursos de entrada/salida (I/O). Los sistemas operativos y los programas de aplicación pueden ejecutarse en máquinas virtuales como invitados. Por ejemplar se pueden ejecutar múltiples Linux e imágenes de z/OS. En el mismo sistema z/VM que está también soportando distintas aplicaciones y usuarios finales. Como resultado, el desarrollo, pruebas, y entornos de producción pueden compartir un único ordenador físico.

- 45 Con referencia a las figs. 15A-15D, la división en particiones y la virtualización implican un desplazamiento en la forma de pensar de físico a lógico tratando recursos de IT como recursos comunes lógicos en vez de como entidades físicas separadas. Esto implica consolidar y compartir recursos de IT, y proporcionar una “ilusión de un sistema único” tanto para servidores homogéneos como heterogéneos, almacenamiento, sistemas distribuidos, y redes.

- 50 La división en particiones del hardware implica CPU separadas para separar sistemas operativos, cada uno de los cuales ejecuta sus aplicaciones específicas. La división en particiones del software emplea un “hipervisor” basado en software para permitir que sistemas operativos individuales se ejecuten en cualquiera o en todas las CPU.

Los hipervisores permiten que múltiples sistemas operativos sean ejecutados en un ordenador anfitrión al mismo tiempo. La tecnología de hipervisor originada en el IBM VM/370, el predecesor del z/VM que existe hoy día. La división en particiones lógica (LPAR) implica la división en particiones del firmware (un hipervisor basado en hardware) para aislar el sistema operativo de las CPU.

- 5 La virtualización permite o explota cuatro capacidades fundamentales: compartir recursos, agregar recursos, emular funciones, y aislar. Se exploran estos tópicos con más detalle en las siguientes secciones.

z/VM es un sistema operativo para la plataforma de System z de IBM que proporciona un entorno de prueba y producción muy flexible. La puesta en práctica de z/VM de la tecnología de virtualización de IBM proporciona la capacidad de ejecutar sistemas operativos de función completa tales como Linux en el System z, z/OS, y otros como "invitados" de z/VM. z/VM
10 soporta invitados de z/Architecture de IBM de 64 bits de IBM e invitados de Architecture/390 de Sistema de Empresa de IBM de 31 bits.

z/VM proporciona a cada usuario un entorno de trabajo individual conocido como una máquina virtual. La máquina virtual simula la existencia de una máquina real dedicada, que incluye funciones de procesador, memoria, redes, y recursos de entrada/salida (I/O). Los sistemas operativos y los programas de aplicación pueden ser ejecutados en máquinas virtuales como invitados. Por ejemplo se pueden ejecutar múltiples imágenes de Linux o de z/OS® en el mismo sistema z/VM que
15 está también soportando distintas aplicaciones y usuarios finales. Como resultado, el desarrollo, las pruebas y los entornos de producción pueden compartir un único ordenador físico.

Una máquina virtual usa recursos de hardware reales, pero incluso con dispositivos dedicados (como una unidad de cinta) La dirección virtual de la unidad de cinta puede ser o no la misma que la dirección real de la unidad de cinta. Por ello, una
20 máquina virtual sólo conoce "hardware virtual" que puede existir o no en el mundo real.

Un z/VM de primer nivel es el sistema operativo base que está instalado en lo alto del hardware real de la fig. 16. Un sistema operativo de segundo nivel es un sistema que es creado sobre la base del sistema operativo z/VM. Por ello, z/VM como sistema operativo de base se ejecuta en el hardware, mientras que un sistema operativo invitado se ejecuta en la tecnología de virtualización. En la fig. 14, se ilustra un z/VM OS invitado de segundo nivel cargado en una partición de
25 invitado de primer nivel (invitado-1).

En otras palabras, hay un sistema operativo z/VM de primer nivel que se asienta directamente en el hardware, pero los invitados de este sistema z/VM de primer nivel están virtualizados. Virtualizando el hardware a partir de los invitados, se es capaz de crear y usar tantos invitados como sea necesario con una pequeña cantidad de hardware.

Como se ha mencionado previamente, los sistemas operativos que se ejecutan en máquinas virtuales son a menudo denominados "invitados". Otros términos y frases que podrían encontrarse son:

- "Ejecución de primer nivel" que significa ejecutar directamente sobre el hardware (que es lo que hace z/VM).
- "Ejecución de segundo nivel", "ejecución bajo VM", o "ejecución sobre (lo alto de) VM" significa ejecutar como un invitado.

Un ejemplo de la funcionalidad de z/VM es, si se tiene un sistema z/VM de primer nivel y un sistema z/VM de segundo nivel, se podría continuar creando más sistemas operativos sobre el sistema de segundo nivel. Este tipo de entorno es particularmente útil para probar la instalación del sistema operativo antes de su despliegue, o para probar o depurar sistemas operativos. Los recursos virtuales pueden tener funciones y características que no están disponibles en sus recursos físicos subyacentes. La fig. 14 ilustra la virtualización por emulación de recursos.

Ejemplos incluyen software de emulación de arquitectura que pone en práctica una arquitectura de procesador usando otra; iSCSI, y ponen practican un bus virtual de SCSI en una red IP; y almacenamiento en cinta virtual puesto en práctica en almacenamiento físico en disco.

Además, el empaquetado de unidades de tratamiento centrales (CPU) en tecnología moderna es a menudo jerárquico. Múltiples núcleos pueden ser colocados en un solo chip. Múltiples chips pueden ser colocados en un único módulo. Múltiples módulos pueden ser empaquetados en una placa a menudo definida como un libro, y múltiples libros pueden ser
45 distribuidos a través de múltiples cuadros.

Las CPU tienen a menudo varios niveles de caché, por ejemplo cada procesador puede tener una caché (o posiblemente una caché de Instrucción dividida y una caché de datos) y puede haber mayores caché adicionales entre cada procesador y la interfaz de memoria principal. Dependiendo del nivel de la jerarquía, las caché pueden ser compartidas entre más de una sola CPU. Las decisiones de ingeniería relativas a tal colocación se relacionan con espacio, potencia/térmico,
50 distancia de cableado, frecuencia de CPU, rendimiento del sistema, y otros aspectos. Esta colocación de elementos de la

CPU crea una estructura interna que puede ser más o menos favorable a una partición lógica particular, dependiendo de donde reside la colocación de cada CPU de la partición. Una partición lógica da la apariencia a un sistema operativo, de propiedad de ciertos recursos incluyendo la utilización del procesador en la que en la actualidad, el sistema operativo está compartiendo los recursos con otros sistemas operativos en otras particiones. Normalmente, el software no es conocedor de la situación y, en una configuración de múltiples tratamientos simétrica (SMP), observa un conjunto de CPU donde cada una proporciona el mismo nivel de rendimiento. El problema es que la ignorancia del empaquetado interno y de la "distancia" entre cualesquiera dos CPU puede dar como resultado un software que hace elecciones menores que óptimas sobre como se le puede asignar el trabajo a las CPU. Por ello, el potencial completo de la configuración SMP no es conseguido.

El ejemplo de computadora central de virtualización presentado esta destinado a enseñar distintas topologías posibles al virtualizar una máquina. Como se ha mencionado, los programas que se ejecutan en una partición (incluyendo los sistemas operativos) probablemente tienen una visión de los recursos que están disponibles para ellos, incluyendo los procesadores, la memoria y los I/O que están dedicados a la partición. De hecho, los programas no tienen ninguna idea de que se están ejecutando en una partición. Tales programas tampoco son conscientes de la topología de su partición y por ello no pueden hacer elecciones basadas en tal topología. Lo que se necesita es un modo para programas que optimicen la topología de configuración sobre la que se están ejecutando.

El documento EP-A-1.674.987 describe un método de acuerdo con la parte de caracterización previa de la reivindicación 1.

EXPOSICIÓN DEL INVENTO

El invento proporciona un método según la reivindicación 1 y un sistema y un programa de ordenador correspondientes.

Los inconvenientes de la técnica anterior son superados, y se proporcionan ventajas adicionales, a través de la previsión de un método, sistema, y producto de programa de ordenador para habilitar un subconjunto de recursos de hardware de ordenador durmientes en un sistema de ordenador actualizable que tiene un conjunto de recursos de hardware durmientes.

Un ordenador anfitrión que comprende CPU anfitrionas puede estar dividido en particiones lógicas/virtuales que tienen CPU invitadas. La división en particiones es realizada preferiblemente por firmware o por software como podría proporcionarse por un sistema operativo tal como z/VM de IBM. Cada CPU invitada es una CPU virtual en la que los programas de invitado ven las CPU invitadas como procesadores de CPU reales, pero de hecho, el anfitrión subyacente está haciendo corresponder cada CPU invitada a recursos de CPU anfitrionas. En una realización, una CPU invitada es puesta en práctica usando una parte de una CPU anfitriona por el anfitrión designando una parte de tiempo de CPU para que la CPU invitada utilice la CPU anfitriona. Se ha considerado que una pluralidad de CPU invitadas podrían ser soportadas por una sola CPU anfitriona pero también puede aplicarse lo opuesto.

En otra realización, las CPU invitadas son emuladas por software por lo que, las rutinas de emulación convierten funciones de la CPU invitada (incluyendo la decodificación y la ejecución de instrucciones) a rutinas que se ejecutan sobre las CPU anfitrionas. Las CPU anfitrionas están aprovisionadas para soportar las CPU invitadas.

En otra realización, una primera imagen de invitado puede ser el anfitrión de una segunda imagen de invitado. En cuyo caso las segundas CPU invitadas están aprovisionadas por las primeras CPU invitadas que son aprovisionadas por sí mismas por CPU anfitrionas. La topología de las configuraciones es una anidación de niveles de CPU invitadas y una o más CPU anfitrionas.

Una nueva instrucción de PERFORM TOPOLOGY FACILITY (EJECUTAR EQUIPO DE TOPOLOGÍA) (PTF) es proporcionada y la instrucción de STORE SYSTEM INFORMATION (STSI) es mejorada para proporcionar un nuevo SYSIB (identificador SYSIB 15.1.2) que proporciona información de afinidad de componentes y de empaquetado lógico al software. Esto permite que el software aplique una selección informada e inteligente sobre como los elementos individuales, tales como unidades de tratamiento de los múltiples procesadores, son asignados a diferentes usos y cargas de trabajo. Así proporcionar información a un programa (OS) para mejorar el rendimiento aumentando las relaciones de acierto de caché compartida por ejemplo.

Una nueva instrucción de PERFORM TOPOLOGY FUNCTION (PTF) es usada por un programa privilegiado tal como un supervisor, un OS, un núcleo y similares para solicitar que se cambie la topología de configuración de la CPU dentro de la cual se está ejecutando el programa. En una realización, la topología de la CPU invitada es conmutada entre polarización horizontal y vertical.

Teniendo la capacidad de enseñar la información de topología de CPU, el programa comprende la "distancia" entre cualesquiera dos o más CPU arbitrarias de una configuración de multitratamiento simétrica.

La capacidad prevista para minimizar la distancia agregada de todas las CPU en una configuración, y cómo las tareas de programa de aplicación particular son distribuidas en las CPU individuales proporciona programas de supervisión con la capacidad de mejorar el rendimiento. El rendimiento mejorado puede resultar a partir de uno o más de los siguientes atributos que son mejorados por un mejor conocimiento de la topología:

5 Acortamiento de los trayectos de señalización entre CPU

El almacenamiento compartido, accedido por múltiples CPU es más probable que esté en caché que están más próximas al conjunto de las CPU. Por ello, el uso de almacenamiento entre caché está confinado a un subconjunto menor de la máquina total y configuración que permite una transferencia más rápida de caché a caché. La presencia de una posición de almacenamiento en la caché más próxima de una CPU (L1) es significativamente más probable que ocurra.

- 10 Debido al rendimiento mejorado, el número de CPU realmente en la configuración puede ser menor en número, aunque realizando aún el mismo trabajo hecho en el mismo tiempo o en un tiempo de ejecución menor. Tal reducción de las CPU reduce el número de trayectos de comunicación que cada CPU debe usar para comunicarse con las otras CPU de la configuración, contribuyendo así además a una mejora del rendimiento total.

- 15 Por ejemplo, si 10 CPU necesitan ejecutar un programa particular, el tráfico entre caché es sustancial mientras que si el mismo programa puede ser ejecutado en una CPU, no hay tráfico entre caché. Esto indica que la presencia de caché de posiciones de almacenamiento deseadas está garantizada para estar en la caché de la única CPU, si dicho almacenamiento está en cualquier caché.

- 20 Cuando el almacenamiento y la jerarquía de caché asociada es local, en oposición a estar distribuida a través de múltiples cuadros físicos (es decir, cajas, etc.), los trayectos de señalización son más cortos. El conocimiento de la topología indica la distancia relativa en seleccionar el subconjunto apropiado de CPU para asignar a un programa de aplicación de tal modo que, incluso dentro de un conjunto mayor de CPU en una configuración SMP, el subconjunto optimiza la distancia minimizada entre ellas. Esto es a veces denominado un grupo de afinidad.

Las nociones de reducción de cómputo de CPU y la distancia entre las CPU son informadas por información de topología que permite que el programa optimice la asignación de CPU a un grupo de afinidad.

- 25 En una realización (fig. 20), en un sistema de ordenador de anfitrión dividido en particiones lógicamente que comprende procesadores anfitriones (CPU anfitrionas), se ha descubierto información de topología de uno o más procesadores invitados (CPU invitadas) de una configuración de invitado y almacenada en una tabla. Preferiblemente, un procesador invitado de la configuración de invitado busca una instrucción de STORE SYSTEM INFORMATION (ALMACENAR INFORMACIÓN DEL SISTEMA) para ejecución. Cuando se ejecuta la instrucción STORE SYSTEM INFORMATION, basada en una solicitud de información de topología de la instrucción de STORE SYSTEM INFORMATION, obtiene información de topología de la configuración de ordenador comprendiendo la información de topología información de anidado de procesadores de la configuración. La información de topología obtenida es almacenada en una tabla de topología de configuración preferiblemente en memoria.

- 35 En una realización, la información de topología comprende la identificación de procesadores anfitriones y los niveles de anidado asociados con la topología de los procesadores anfitriones.

En otra realización, la información de topología comprende información acerca de procesadores anfitriones de configuración de invitado y niveles de anidado asociados con la topología de los procesadores anfitriones de la configuración de invitado.

- 40 En una realización, la instrucción de STORE SYSTEM INFORMATION comprende un campo de código de operación, un campo de registro base, un campo de desplazamiento con signo, en los que la instrucción de descubrimiento de topología comprende además un primer registro general virtualmente implicado que contiene un campo de código de función y un campo selector-1 y un segundo registro general virtualmente implicado que contiene un campo selector-2, especificando el campo de código de función la solicitud de información de topología, el campo de registro base y el campo de desplazamiento con signo que identifica una posición en la memoria de un bloque de información de sistema (SYSIB) que contiene la tabla de topología de configuración, en la que los valores del campo selector-1 y del campo selector-2, en combinación, determinan la solicitud de información de topología que se ha de realizar.

En otra realización, la tabla incluye una entrada de procesador de lista de topología para cada grupo de procesadores anidados de los procesadores.

- 50 En aun otra realización, la entrada de procesador de lista de topología comprende además un indicador que indica cómo de dedicados están los procesadores del grupo de procesadores anidados a la configuración de invitado de división en particiones lógicas.

En una realización, la tabla incluye además una entrada de recipiente de lista de topología de cada nivel de anidado para una jerarquía de uno o más niveles de anidado que tienen dichos procesadores anidados.

En una realización, la ejecución de STORE SYSTEM INFORMATION es realizada por emulación en un procesador ajeno.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

- 5 El sujeto en cuestión que es considerado como el invento está indicado particularmente y reivindicada de forma distintiva en la parte concluyente de la memoria. El invento, sin embargo, tanto en cuanto a organización como a método de práctica, junto con otros objetos y ventajas del mismo, puede ser mejor comprendido por referencia a la descripción siguiente tomada en conexión con los dibujos adjuntos en los que:

La fig. 1 representa un sistema de Ordenador Anfitrión de la técnica anterior;
- 10 La fig. 2 representa un sistema de Ordenador Anfitrión emulado de la técnica anterior;

La fig. 3 representa un formato de instrucción de una instrucción de máquina STSI de la técnica anterior;

La fig. 4 representa un registro virtualmente implicado de la instrucción STSI de la técnica anterior;

La fig. 5 representa una tabla de código de función;

La fig. 6 representa una tabla SYSIB 1.1.1 de la técnica anterior;
- 15 La fig. 7 representa una tabla SYSIB 1.2.2 de la técnica anterior;

La fig. 8 representa una tabla SYSIB 1.2.2 de Formato-1 de la técnica anterior;

La fig. 9 representa una tabla SYSIB 1.2.2 de Formato-2 de la técnica anterior;

La fig. 10 representa una tabla SYSIB 15.1.2 de acuerdo con el invento;

La fig. 11 representa un TLE de tipo contenedor;
- 20 La fig. 12 representa un TLE de tipo CPU;

La fig. 13 representa un formato de instrucción de una instrucción de máquina PTF de acuerdo con el invento;

La fig. 14 representa un formato de registro de la instrucción PTF;

Las figs. 15A-15D representan elementos de la técnica anterior de sistemas de ordenador divididos en particiones;

La fig. 16 representa un sistema de ordenador ejemplar;
- 25 Las figs. 17-19 representan contenedores del sistema de ordenador ejemplar; y

La fig. 20 representa un flujo de una realización del invento.

DESCRIPCIÓN DETALLADA

- 30 En una computadora central, las instrucciones de máquina organizadas en arquitectura son usadas por programadores (típicamente escribiendo aplicaciones en "C" pero también en Java®, COBOL, PL/I, PL/X, Fortran y otros lenguajes de nivel alto), a menudo a modo de una aplicación compiladora. Estas instrucciones almacenadas en el medio de almacenamiento pueden ser ejecutadas nativamente en un Servidor IBM de z/Architecture, o alternativamente en máquinas que ejecutan otras arquitecturas. Pueden ser emuladas en los servidores de computadora central de IBM existentes y futuros y en otras máquinas de IBM (por ejemplo, Servidores de pSeries® y Servidores de xSeries®). Pueden ser ejecutadas en máquinas que ejecutan Linux en una amplia variedad de máquinas que usan un hardware fabricado por
- 35 IBM®, Intel®, AMD™, Sun Microsystems y otros. Además de la ejecución en ese hardware bajo una z/Architecture®, puede usarse Linux así como máquinas que usan emulación por Hércules, UMX, FSI (Fundamental Software, Inc) o Platform Solutions, Inc. (PSI) donde generalmente la ejecución es un modo de emulación. En el modo de emulación, el software de emulación es ejecutado por un procesador nativo para emular la arquitectura de un procesador emulado.
- 40 El procesador nativo ejecuta típicamente el software de emulación que comprende tanto firmware como un sistema operativo nativo para realizar la emulación del procesador emulado. El software de emulación es responsable de buscar y ejecutar instrucciones de la arquitectura de procesador emulado. El software de emulación mantiene un contador de programa emulado para guardar la pista de límites de instrucción. El software de emulación puede buscar una o más

instrucciones de máquina emulada en un instante y convertir la una o más instrucciones de máquina emulada a un grupo correspondiente de instrucciones de máquina nativa para su ejecución por el procesador nativo. Estas instrucciones convertidas pueden ser memorizadas en caché de tal modo que puede lograrse una conversión más rápida. Sin oponerse, el software de emulación debe mantener las reglas de arquitectura de la arquitectura de procesador emulado de modo que asegure que los sistemas operativos y aplicaciones escritas para el procesador emulado funcionan correctamente. Además el software de emulación debe proporcionar recursos identificados por la arquitectura de procesador emulado que incluyen, pero no están limitados a registros de control, registros de propósito general (que incluyen a menudo registros de coma flotante), función de traslación de dirección dinámica que incluye tablas de segmento y tablas de página por ejemplo, mecanismos de interrupción, mecanismos de cambio de contexto, relojes de Hora del Día (TOD) e interfaces de arquitectura para subsistemas I/O de tal modo que un sistema operativo o un programa de aplicación diseñado para ejecutarse en el procesador emulado, puede ser ejecutado en el procesador nativo que tiene el software de emulación.

Una instrucción específica que es emulada es descodificada, y una subrutina llamada para realizar la función de la instrucción individual. Una función de software de emulación que emula una función de un procesador emulado es puesta en práctica, por ejemplo, en una subrutina "C" o unidad, o algún otro método de proporcionar una unidad para el hardware específico como será comprendido por la experiencia de aquellos técnicos expertos después de comprender la descripción de la realización preferida. Distintas patentes de emulación de hardware y software que incluyen, pero no están limitadas a la patente norteamericana nº 5.551.013 para un "Multiprocesador para emulación de hardware" de Beausoleil y col., y la patente norteamericana nº 6.009.261: "Tratamiento previo de rutinas de objetivo almacenado para emular instrucciones incompatibles en un procesador objetivo" de Scalzi y col.; y la patente norteamericana nº 5.574.873: "Descodificar la instrucción de invitado para acceder directamente a rutinas de emulación que emulan las instrucciones de invitado, de Davidian y col; la patente norteamericana nº 6.308.255: "Bus de multitratamiento simétrico y conjunto de chips usado para el coprocesador que soporta la autorización de ejecutar un código no nativo en un sistema", de Gorishek y col.; y la patente norteamericana nº 6.463.582: "Traductor de código de objeto de optimización dinámica para emulación de arquitectura y método de traducción de código de objeto de optimización dinámica" de Lethin y col.; y la patente norteamericana nº 5.790.825: "Método para emular instrucciones de invitado en un ordenador anfitrión a través de una recopilación dinámica de instrucciones de anfitrión" de Eric Traut; y muchas otras, ilustran la variedad de formas conocidas para conseguir la emulación de un formato de instrucción de arquitectura para una máquina diferente para una máquina objetivo disponible para los expertos en la técnica, así como aquellas técnicas de software comercial usadas por aquellos a los que se ha hecho referencia antes.

Con referencia a la fig. 1, se han retratado componentes representativos de un sistema 100 de ordenador anfitrión. También pueden emplearse otras disposiciones en un sistema de ordenador que son bien conocidas en la técnica.

El entorno de ordenadores anfitriones está basado preferiblemente en la z/Architecture ofrecida por Internacional Business Machines Corporation (IBM®), Armonk, Nueva York. La z/Architecture está descrita más completamente en: *Principios de Funcionamiento de z/Architecture*, IBM® Publicación Nº SA22-7832-05 6ª Edición, (Abril de 2007). Entornos de ordenadores basados en la z/Architecture incluyen, por ejemplo eServer™ y zSeries®, ambas por IBM®.

El ordenador anfitrión representativo 100 comprende una o más CPU 101 en comunicación con el almacén principal (memoria de ordenador 102) así como las interfaces de I/O a dispositivos de almacenamiento 111 y redes 101 para comunicar con otros ordenadores o SAN y similares. La CPU puede tener una Traducción de Dirección Dinámica (DAT) 103 para transformar direcciones de programa (direcciones virtuales) en la dirección real de memoria. Una DAT incluye típicamente una Memoria Tampón de Traslación de Búsqueda a un Lado (TLB) 109 para memorizar traslaciones de modo que los últimos accesos al bloque de la memoria de ordenador 102 no requieren el retraso de la traslación de dirección. Típicamente una memoria caché 109 es empleada entre una memoria de ordenador 102 y el procesador 101. La caché 109 puede ser jerárquica con una caché grande disponible para más de una CPU y caché menores, más rápidas (nivel inferior) entre la caché grande y cada CPU. En algunas puestas en práctica las caché inferiores están divididas para proporcionar caché de bajo nivel separadas para buscar la instrucción y los accesos de datos. En una realización, una instrucción es buscada desde la memoria 102 por una unidad 104 de búsqueda de instrucción mediante una caché 109. La instrucción es descodificada en una unidad (706) de descodificación de instrucción y distribuida (con otras instrucciones en algunas realizaciones) a unidades 108 de ejecución de instrucción. Típicamente varias unidades 108 de ejecución son empleadas, por ejemplo una unidad de ejecución aritmética, una unidad de ejecución de coma flotante y una unidad de ejecución de instrucción de rama. La instrucción es ejecutada por la unidad de ejecución, accediendo a operandos desde los registros de instrucción especificados o a la memoria cuando sea necesario. Si se ha de acceder a un operando (cargado o almacenado) desde la memoria 102, una unidad 105 de almacenamiento de carga maneja típicamente el acceso bajo control de la instrucción que se está ejecutando.

En una realización, el invento puede ser puesto en práctica por un software (a veces denominado Código Interno Autorizado (LIC), firmware, micro-código, mili-código, pico-código y similares, cualquiera de los cuales sería consistente

con el presente invento). El código de programa de software que pone en práctica el presente invento es típicamente accedido por el procesador también conocido como una CPU (Unidad de Tratamiento o Procesado Central) 101 del sistema de ordenador 100 desde el medio 111 de almacenamiento a largo plazo, tal como una unidad de CD-ROM, una unidad de cinta o un disco duro. El código de programa del software puede ser puesto en práctica sobre cualquiera de una
5 variedad de medios conocidos para usar con un sistema de tratamiento de datos, tal como un disquete, un disco duro, o un CD-ROM. El código puede ser distribuido en tal medio, o puede ser distribuido a usuarios desde la memoria del ordenador 102 o el almacenamiento de un sistema de ordenador sobre una red 110 a otros sistemas de ordenador para usar por los usuarios de tales otros sistemas.

Alternativamente, el código de programa puede ser puesto en práctica en la memoria 102, y accedido por el procesador
10 101 que usa el bus del procesador. Tal código de programa incluye un sistema operativo que controla la función e interacción de los distintos componentes del ordenador y uno o más programas de aplicación. El código de programa está normalmente paginado a partir de medios 111 de almacenamiento densos a la memoria 102 de alta velocidad en la que está disponible para su tratamiento por el procesador 101. Las técnicas y métodos para poner en práctica el código de programa de software en la memoria, sobre un medios físico, y/o distribuir el código del software a través de las redes son
15 bien conocidos y no serán descritos aquí adicionalmente. El código de programa, cuando es creado y almacenado en un medio tangible (que incluye pero no está limitado a módulos de memoria electrónica (RAM), memoria flash, discos compactos (CD), DVD, cinta magnética y similar es a menudo denominado como un "producto de programa de ordenador". El medio de producto de programa de ordenador es típicamente legible por un circuito de tratamiento preferiblemente en un sistema de ordenador para su ejecución por el circuito de tratamiento.

En la fig. 2, se ha proporcionado un sistema 201 de ordenador anfitrión emulado ejemplar que emula un sistema de
20 ordenador anfitrión 100 de una arquitectura de anfitrión. En el sistema 201 de ordenador anfitrión emulado 201, el procesador anfitrión (CPU) 208 es un procesador anfitrión emulado (o procesador anfitrión virtual) y comprende un procesador de emulación 207 que tiene una arquitectura de conjunto de instrucción nativa diferente que la usada por el procesador 101 del ordenador anfitrión 100. El sistema 201 de ordenador anfitrión emulado tiene una memoria 202
25 accesible al procesador de emulación 207. En la realización ejemplar, la memoria 207 está dividida en particiones en una parte de la memoria 102 de ordenador anfitrión y una parte de rutinas de emulación 203. La memoria 102 de ordenador anfitrión está disponible para programas del ordenador anfitrión emulado 201 de acuerdo con una arquitectura de ordenador anfitrión. El procesador de emulación 207 ejecuta instrucciones nativas de un conjunto de instrucciones de arquitectura de una arquitectura distinta de la del procesador emulado 208, las instrucciones nativas obtenidas a partir de
30 la memoria 203 de rutinas de emulación, y puede acceder a una instrucción de anfitrión para su ejecución desde un programa en la memoria 102 de ordenador anfitrión empleando una o más instrucciones obtenidas en una rutina de Secuencia y Acceso/Descodificación que puede descodificar la instrucción o instrucciones accedidas para determinar una rutina de ejecución de instrucción nativa para emular la función de la instrucción de anfitrión a la que se ha accedido.

Otros equipos que están definidos para la arquitectura del sistema de ordenador anfitrión 100 pueden ser emulados por
35 Rutinas de Equipos de Arquitectura, que incluyen tales equipos como Registros de Propósito General, Registros de Control, Traslación de Dirección Dinámica, y soporte del Subsistema de I/O y caché del procesador por ejemplo. Las rutinas de emulación pueden beneficiarse también de la función disponible en el procesador de emulación 207 (tal como Registros Generales y traslación dinámica de dirección virtual) para mejorar el rendimiento de las rutinas de emulación. Pueden también preverse un hardware especial y Máquinas de Carga para ayudar al procesador 207 en la emulación de
40 la función del ordenador anfitrión 100.

Con el fin de proporcionar información de topología a los programas, se han proporcionado dos instrucciones. La primera es una mejora para una instrucción de la técnica anterior STSI (STORE SYSTEM INFORMATION) y la segunda es una instrucción PTF (PERFORM TOPOLOGY FUNCTION (EJECUTAR FUNCIÓN DE TOPOLOGÍA)).

Perspectiva de la Topología de la CPU:

Con la llegada de los nuevos ordenadores centrales de eSeries de IBM, e incluso previamente, la organización de las
45 máquinas en estructuras nodales ha dado como resultado un comportamiento de acceso de memoria no uniforme (NUMA) (a veces también llamado "irregular"). El propósito de la nueva función SYSIB 15.1.2 de la instrucción de la técnica anterior STSI (STORE SYSTEM INFORMATION) y la nueva instrucción de PERFORM TOPOLOGY FUNCTION (PTF) es proporcionar conocimiento de topología de máquina adicional al programa de modo que puedan realizarse
50 ciertas optimizaciones (incluyendo relaciones de caché-hit) y mejorar por ello el rendimiento total. La cantidad de recursos de la CPU anfitriona asignada a una configuración de invitado de multitratamiento (MP) ha sido distribuida generalmente de modo uniforme a través del número de CPU invitadas configuradas. (Una CPU invitada es una CPU lógica proporcionada a un programa, todas las CPU invitadas están soportadas por software/hardware dividido en particiones en las CPU anfitrionas reales). Tal distribución uniforme implica que ninguna CPU invitada particular está autorizada a ningún
55 aprovisionamiento de CPU anfitriona extraordinaria que cualesquiera CPU invitadas determinadas arbitrariamente. Este

estado de la configuración de invitado, que afecta a todas las CPU de la configuración, es denominado "polarización horizontal". Bajo la polarización horizontal, la asignación de una CPU anfitriona a una CPU invitada es aproximadamente la misma cantidad de aprovisionamiento para cada CPU invitada. Cuando el aprovisionamiento no está dedicado, las mismas CPU anfitrionas que aprovisionan a las CPU invitadas también pueden ser usadas para aprovisionar CPU invitadas de otro invitado, o incluso otras CPU invitadas de la misma configuración de invitado.

Cuando la otra configuración de invitado es una partición lógica diferente, una CPU anfitriona, cuando está activa en cada partición, típicamente debe tener acceso más al almacenamiento principal porque la relación caché-acierto es reducida teniendo que compartir las caché a través múltiples zonas de reubicación. Si el aprovisionamiento de la CPU anfitriona puede alterar el equilibrio de tal modo que algunas CPU anfitrionas están en su mayor parte, o incluso exclusivamente, asignadas a una configuración de anfitrión dada (y que resulta el comportamiento normal), entonces las relaciones de caché-acierto mejoran, como lo hace el funcionamiento. Tal distribución desigual implica que una o más CPU invitadas están autorizadas para aprovisionar adicionalmente la CPU anfitriona frente a otras, CPU invitadas determinadas arbitrariamente que están autorizadas para un menor aprovisionamiento de CPU anfitriona. Este estado de la configuración de invitado, que afecta a todas las CPU de la configuración, es denominado "polarización vertical". La arquitectura presentada aquí clasifica la polarización vertical en tres niveles de autorización de aprovisionamiento, alto, medio, y bajo:

- La autorización alta garantiza aproximadamente que el 100% de una CPU anfitriona es asignada a una CPU lógica/virtual, y la afinidad es mantenida como una fuerte correspondencia entre las dos. Con respecto al aprovisionamiento de una partición lógica, cuando la polarización vertical es en efecto, la autorización de una CPU dedicada es definida como alta.

- La autorización media garantiza que una cantidad sin especificar de recursos de una CPU anfitriona (una o más CPU anfitrionas) es asignada a una CPU virtual/lógica, y cualquier capacidad restante de la CPU anfitriona es considerada como un espacio libre que puede ser asignado a otra parte. El mejor caso para el espacio libre disponible sería asignarlo como espacio libre local si esto es posible. Un resultado menos beneficioso ocurre si ese espacio libre disponible es asignado como espacio libre remoto. Es también el caso de que el porcentaje de recursos asignado a una CPU lógica de autorización media es una aproximación más suave si se la compara con la aproximación del 100% de un ajuste de autorización alta.

- La autorización baja garantiza que aproximadamente el 0% de una CPU anfitriona es asignada a una CPU virtual/lógica. Sin embargo, si hay disponible espacio libre, tal CPU virtual/lógica puede recibir aun algún recurso de CPU. Un modelo de contenedores anidados que usa polarización está destinado a proporcionar un nivel de inteligencia acerca de la estructura nodal de la máquina cuando se aplica a la configuración de solicitud, de modo que, generalmente, grupos de CPU anfitrionas pueden estar asignados a grupos de CPU invitadas, mejorando así tanto como sea posible el hecho de compartir almacenamiento y la minimización de configuraciones diferentes que colisionan esencialmente en las mismas CPU anfitrionas. La polarización y la autorización indican la relación de CPU físicas a CPU lógicas o CPU lógicas a CPU virtuales en una configuración de invitado, y cómo la capacidad asignada a la configuración de invitado es repartida a través de las CPU que comprenden la configuración.

Históricamente, una configuración de invitado ha sido polarizada horizontalmente. Sin embargo, muchas CPU invitadas fueron definidas para la configuración, el recurso de CPU anfitriona asignado fue distribuido uniformemente a través de todas las CPU invitadas de una forma equitativa, no autorizada. Puede decirse que el peso de una única CPU lógica en una partición lógica cuando la polarización horizontal es en efecto es aproximadamente igual al peso de configuración total dividido por el número de CPU.

Sin embargo, con la introducción de la 2097 y los modelos de familia, resulta imperativo ser capaz de extender el recurso de la CPU anfitriona de una forma diferente, que es denominada polarización vertical de una configuración, y estando indicado entonces el grado de aprovisionamiento de CPU invitadas con CPU anfitrionas como autorización alta, media o baja. La autorización alta es en efecto cuando una CPU virtual/lógica de una configuración polarizada verticalmente está completamente soportada por la misma CPU anfitriona. La autorización media es en efecto cuando una CPU virtual/lógica de una configuración polarizada verticalmente está parcialmente soportada por una CPU anfitriona. La autorización baja es en efecto cuando una CPU virtual/lógica de una configuración polarizada verticalmente no tiene garantizado ningún recurso de CPU anfitriona, diferente que el que podría resultar disponible debido a que el recurso de espacio libre resulta disponible.

Espacio Libre de CPU:

Recurso de CPU, hay dos clases de recurso de espacio libre de CPU:

- El espacio libre local resulta disponible cuando una CPU virtual/lógica de una configuración no está usando todos los recursos a los que está autorizada y dicho espacio libre es a continuación usado dentro de la configuración de dicha CPU. El espacio libre local es preferido sobre el espacio libre remoto cuando se esperan mejores relaciones de acierto sobre las caché cuando el espacio libre es usado dentro de la configuración.

- 5
- El espacio libre remoto resulta disponible cuando una CPU virtual/lógica de una configuración no está usando todos los recursos a los que está autorizada y dicho espacio libre es a continuación usado fuera de la configuración de dicha CPU. El espacio libre remoto se espera que exhiba relaciones de acierto inferiores sobre las caché, pero es aún mejor que no ejecutar una CPU virtual/lógica del todo.

10 El objetivo es maximizar las relaciones de acierto sobre la caché de la CPU. Para una partición lógica, la cantidad de recursos de CPU física está determinada por las ponderaciones del sistema total que determina los recursos de CPU asignados a cada partición lógica. Por ejemplo, en una MP lógica de 3 vías que tiene asignado recursos de CPU física equivalentes a una única CPU, y está polarizada horizontalmente, cada CPU lógica podría ser distribuida independientemente y recibir así aproximadamente el 33% de los recursos de CPU física. Si la misma configuración estuviera polarizada verticalmente, sólo una única CPU lógica sería ejecutada y recibiría aproximadamente el 100% de los recursos de CPU física (autorización alta) mientras las dos CPU lógicas restantes no serían distribuidas normalmente (autorización baja). Tal asignación de recursos es normalmente una aproximación. Incluso una CPU de autorización baja puede recibir alguna cantidad de recursos si es sólo para ayudar a asegurar que un programa no resulta pegado en tal CPU. Previendo unos medios para que un programa de control indique si comprende la polarización, y para recibir una indicación para cada CPU de su polarización y, si la polarización es vertical, el grado de autorización, el programa de control puede hacer un uso más inteligente de estructuras de datos que generalmente son pensados para ser locales para una CPU frente a disponible para todas las CPU de una configuración. También, tal programa de control puede evitar dirigir trabajo a cualquier CPU de autorización baja. Los recursos de CPU física reales asignados podrían no constituir un número entero de CPU, así hay también la posibilidad de una o más CPU en una configuración polarizada verticalmente MP que está autorizada pero no a un grado alto, dando como resultado por ello en tal CPU que tiene autorización vertical, bien media o baja. Es posible para cualesquiera de las CPU de autorización baja restantes recibir alguna cantidad de recursos de CPU anfitriona. Por ejemplo, esto puede ocurrir cuando tal CPU es un objetivo, tal como mediante una orden de SIGP y el recurso de CPU anfitriona de espacio libre está disponible. De otra manera, tal CPU virtual/lógica podría permanecer en un estado no distribuido, incluso si es de otra manera capaz de ser distribuida.

20 Preferiblemente, de acuerdo con el invento, un campo de polarización de 2 bits es definido para la nueva "entrada de lista-topología" (TLE) de tipo CPU de la instrucción de STORE SYSTEM INFORMATION (STSI). El grado de autorización de polarización vertical para cada CPU está indicado como alta, media, o baja. La asignación no es un porcentaje preciso sino que en vez de ello es algo casual y heurístico.

25 Además de la polarización vertical como un medio de volver a asignar ponderación a CPU invitadas, existe otro concepto, que es la creación y gestión de capacidad de dejar espacio libre (también denominado "espacio en blanco"). La capacidad de dejar espacio libre es creada en las siguientes circunstancias:

- 35
- Una CPU vertical alta contribuye a dejar espacio libre cuando su utilización media (AU) cae por debajo del 100 por ciento (100-AU).
 - Una CPU vertical media que tiene un aprovisionamiento asignado de M por ciento de una CPU anfitriona contribuye a dejar espacio libre cuando su utilización media (AU) cae por debajo de M por ciento ($M-AU > 0$).
 - Una CPU vertical baja no contribuye a dejar espacio libre.
 - Una CPU vertical alta no es un consumidor de espacio libre.
- 40

Instrucción de STORE SYSTEM INFORMATION:

Una realización ejemplar de un formato de una instrucción de STORE SYSTEM INFORMATION en la fig. 3 comprende un campo de código de operación 'B27D', un campo B2 de registro base y un campo D2 de desplazamiento con signo. El código de operación de instrucción informa a la máquina que ejecuta la instrucción que hay registros generales '0' y '1' implicados asociados con la instrucción. Una dirección es obtenida de un segundo operando añadiendo el valor de campo de desplazamiento con signo al contenido del registro general especificado por el campo base. En una realización, cuando el campo de registro base es "0", el valor de signo extendido del campo de desplazamiento es usado directamente para especificar el segundo operando. Cuando la instrucción de STORE SYSTEM INFORMATION es buscada y ejecutada, dependiendo de un código de función en el registro general 0, o bien una identificación del nivel de la configuración que ejecuta el programa es situada en el registro general 0 o bien una información acerca de un componente o componentes de una configuración es almacenada en un bloque de información del sistema (SYSIB). Cuando la información acerca de

un componente o componentes es solicitada, la información es especificada por un contenido adicional del registro general 0 y por el contenido del registro general 1. El SYSIB, si hay, está designado por la dirección del segundo operando. La máquina se considera que proporciona uno, dos, o tres niveles de configuración. Los niveles son:

1. La máquina básica, que es la máquina como si estuviera operando en el modo básico.
2. Una partición lógica, que es proporcionada si la máquina está operando en el modo LPAR (dividida en particiones lógicamente). Una partición lógica es proporcionada por el hipervisor LPAR, que es una parte de la máquina. Una máquina básica existe incluso cuando la máquina está operando en el modo LPAR.
3. Una máquina virtual, que es proporcionada por un programa de control de máquina virtual (VM) que es ejecutado o bien por la máquina básica o bien en una partición lógica. Una máquina virtual puede ejecutar por sí misma un programa de control VM que proporciona una máquina virtual de nivel más elevado (más retirada de la máquina básica), que también es considerada una configuración de nivel-3. Los términos modo básico, modo LPAR, partición lógica, hipervisor, y máquina virtual, y cualesquiera otros términos referidos específicamente a esos términos, no están definidos en esta publicación; están definidos en los manuales de las máquinas. Un programa que es ejecutado por una configuración de nivel-1 (la máquina básica) puede solicitar información acerca de esa configuración. Un programa que es ejecutado por una configuración de nivel-2 (en una partición lógica) puede solicitar información acerca de la partición lógica y acerca de la máquina básica subyacente. Un programa que es ejecutado por una configuración de nivel-3 (una máquina virtual) puede solicitar información acerca de la máquina virtual y acerca del uno o dos niveles subyacentes; una máquina básica es siempre subyacente, y una partición lógica puede o no estar entre la máquina básica y la máquina virtual. Cuando la información acerca de una máquina virtual es solicitada, es proporcionada información acerca de la configuración que ejecuta el programa y acerca de cualquier nivel o niveles subyacentes de la máquina virtual. En cualquiera de estos casos, es proporcionada información acerca de un nivel sólo si el nivel pone en práctica la instrucción.

El código de función que determina la operación es un número entero binario sin signo en posiciones de bit 32-35 del registro general 0 y es como sigue:

Código de función Información Solicitada:

0	Número de nivel de configuración actual
1	Información acerca del nivel 1 (la máquina básica)
2	Información acerca del nivel 2 (una partición lógica)
3	Información acerca del nivel 3 (una máquina virtual)
4-14	Ninguna; los códigos están reservados
15	información del nivel de configuración actual

Código de Función Inválido:

El nivel de la configuración que ejecuta el programa es denominado el nivel actual. La configuración de nivel especificada por un código de función no-cero es denominada el nivel especificado. Cuando es nivel especificado está numerado más alto que el nivel actual, entonces el código de función es denominado inválido, el código de estado es ajustado a 3, y ninguna otra acción (incluyendo la comprobación) es realizada.

Código de Función Válido:

Cuando el código de función es igual o menor que el número del nivel actual, es denominado válido. En este caso, los bits 36-55 del registro general 0 y los bits 32-47 del registro general 1 deben ser cero o 15; de otra manera, es reconocida una excepción de especificación. Los bits 0-31 de los registros generales 0 y 1 siempre son ignorados. Cuando el código de función es 0, un entero binario sin signo que identifica el nivel de configuración actual (1 para la máquina básica, 2 para la partición lógica, o 3 para la máquina virtual) es situado en las posiciones de bit 32-35 del registro general 0, el código de condición es ajustado a 0, y ninguna otra acción es realizada. Cuando el código de función es válido y no-cero, los registros generales 0 y 1 contienen especificaciones adicionales acerca de la información solicitada, como sigue:

- Las posiciones de bit 56-63 del registro general 0 contienen un entero binario sin signo, denominado selector 1, que especifica un componente o componentes de la configuración especificada.
- Las posiciones de bit 48-63 del registro general 1 contienen un entero binario sin signo, denominado selector 2, que especifica el tipo de información solicitada.

5 El contenido de los registros generales 0 y 1 han sido mostrado en la fig. 4.

Cuando el código de función es válido y no-cero, la información puede ser almacenada en un bloque de información del sistema (SYSIB) que comienza en la posición del segundo operando. El SYSIB es de 4K bytes y debe comenzar en un límite de 4K-bytes; de otra manera, una excepción de especificación puede ser reconocida, dependiendo del selector 1 y del selector 2 y de si las excepciones de acceso son reconocidas debido a referencias al SYSIB.

10 El selector 1 puede tener los valores siguientes:

Selector 1	Información Solicitada
------------	------------------------

0	Ninguna; el selector está reservado
1	La información acerca del nivel de configuración especificado por el código de función
2	La información acerca de una o más CPU en el nivel de configuración especificado
3-255	Ninguna; los selectores están reservados

Cuando el selector 1 es 1, el selector 2 puede tener los valores siguientes:

Selector 2 cuando

Selector 1 es 1	Información Solicitada
-----------------	------------------------

0	Ninguna; el selector está reservado
1	Información acerca del nivel de configuración especificado
2	Información de topología acerca del nivel de configuración especificado
3-65.535	Ninguna; los selectores están reservados

25 Cuando el selector 1 es 2, el selector 2 puede tener los valores siguientes:

Selector 2 cuando

Selector 1 es 2	Información Solicitada
-----------------	------------------------

0	Ninguna; el selector está reservado
1	Información acerca de la CPU que ejecuta el programa en el nivel de configuración especificado
2	Información acerca de todas las CPU en el nivel de configuración especificado
3-65.535	Ninguna; los selectores están reservados

Sólo algunas combinaciones del código de función, selector 1, y selector 2 son válidas, como se ha mostrado en la fig. 5.

35 Cuando la combinación del código de función especificado, selector-1, y selector-2 es inválida (es distinta de la mostrada en la fig. 5) o si es válida pero la información solicitada no está disponible debido al nivel especificado no se pone en práctica o no se pone en práctica completamente la instrucción o debido a que una parte necesaria del nivel está desinstalada o no inicializada, y dado que una excepción no es reconocida, el código de estado es ajustado a 3. Cuando el código de función no es cero, la combinación es válida, la información solicitada está disponible, y no hay excepción, la

información solicitada es almacenada en un bloque de información del sistema (SYSIB) en la dirección del segundo operando.

Algunos o todos los SYSIB puede ser buscados antes de ser almacenados.

5 Un SYSIB puede estar identificado en referencias por medio de "SYSIB fc.s1.s2", donde "fc", "s1", y "s2" son los valores de un código de función, selector 1, y selector 2, respectivamente.

Las secciones siguientes describen los SYSIB definidos por medio de figuras y texto relacionado. En las figuras, los desplazamientos mostrados en la izquierda son valores de palabra (una palabra que comprende 4 bytes). "La configuración" se refiere al nivel de configuración especificado por el código de función (el nivel de configuración acerca del que es solicitada información).

10 SYSIB 1.1.1 (Configuración de Máquina Base)

SISIB 1.1.1 tiene el formato mostrado en la fig. 6, donde los campos tienen los siguientes significados:

Reservado: El contenido de las palabras 0-7, 13-15, y 29-63 está reservado y es almacenado como ceros. El contenido de las palabras 64-1023 está reservado y puede ser almacenado como ceros o muchos permanecen sin cambios.

15 Fabricante: Las palabras 8-11 contienen los 16-caracteres (0-9 o A-Z mayúsculas) nombre EBCDIC del fabricante de la configuración. El nombre es justificado a la izquierda con espacios en blanco posteriores si es necesario.

Tipo: La palabra 12 contiene cuatro caracteres (0-9) de número de tipo EBCDIC de la configuración. (Esto es denominado el número de tipo de máquina en la definición de la ID de la CPU de STORE (ALMACÉN)).

20 Identificador de la Capacidad del Modelo: Las palabras 16-19 contienen LOS 16 caracteres (0-9 o A-Z mayúsculas) el identificador de capacidad del modelo EBCDIC de la configuración. El identificador de capacidad del modelo es justificado a la izquierda con espacios en blanco posteriores si es necesario.

25 Código de Secuencia: Las palabras 20-23 contienen los 16 caracteres (0-9 o A-Z mayúsculas) de código de secuencia EBCDIC de la configuración. El código de secuencia es justificado a la derecha con ceros que preceden al EBCDIC si es necesario.

Planta de Fabricación: La palabra 24 contiene los cuatro caracteres (0-9 o A-Z mayúsculas) de código EBCDIC que identifica la planta de fabricación para la configuración. El código es justificado a la izquierda con espacios en blanco posteriores si es necesario.

30 Modelo: Cuando la palabra 25 no son ceros binarios, las palabras 25-28 contienen los 16 caracteres (0-9 o A-Z mayúsculas) la identificación del modelo EBCDIC de la configuración. La identificación del modelo es justificada a la izquierda con espacios en blanco posteriores si es necesario. (Esto es denominado el número de modelo en la nota 4 de programación en la página 10-111 de STORE CPU ID). Cuando la palabra 25 son ceros binarios, el contenido de las palabras 16-19 representa tanto el identificador de capacidad del modelo como el modelo.

Notas de programación:

35 1 Los campos del SYSIB 1.1.1 son similares a los del descriptor de nodo descrito en la publicación Comandos de Dispositivo de I/O Comunes y Auto-Descripción. Sin embargo, el contenido de los campos de SYSIB puede no ser idéntico al contenido de los campos del descriptor de nodo correspondientes porque los campos de SYSIB:

- Permiten más caracteres.

- Son más flexibles con respecto al tipo de caracteres permitidos.

40 • Proporcionan información que está justificada de modo diferente dentro del campo.

- Puede no usar el mismo método para determinar el contenido de campos tales como el campo de código de secuencia.

2. El campo de modelo en un descriptor de nodos corresponde al contenido del campo de modelo de STSI y no al campo identificador de capacidad de modelo de STSI.

45 3. El campo de modelo especifica el modelo de la máquina (es decir el modelo físico); el campo identificador de

capacidad de modelo específica una credencial que puede ser usada para situar un estado de capacidad o rendimiento en la publicación de Biblioteca del Sistema para el modelo.

SYSIB 1.2.1 (CPU de Máquina Básica)

SYSIB 1.2.1 tiene el formato mostrado en la fig. 7 en el que los campos tienen el siguiente significado:

- 5 Reservado: el contenido de las palabras 0-19, bytes 0 y 1 de la palabra 25 y las palabras 26, 63 está reservado y es almacenado como ceros. El contenido de las palabras 64-1023 es reservado y puede ser almacenado como ceros o puede permanecer sin cambios.
- 10 Código de Secuencia: las palabras 20-23 contienen los 16 caracteres (0-9 o mayúsculas A-Z) de código de secuencia de EBCDIC de la configuración. El código está justificado a la derecha con ceros anteriores a EBCDIC si fuera necesario.
- Planta de fabricante: La palabra 24 contiene los cuatro caracteres (0-9 o mayúsculas A-Z) de código de EBCDIC que identifica la planta de fabricación para la configuración. El código es justificado a la izquierda con espacios en blanco posteriores si fuera necesario.
- 15 Dirección de CPU: Los bytes 2 y 3 de la palabra 25 contienen la dirección de CPU por la que esta CPU es identificada en una configuración de multitratamiento. La dirección de CPU es un entero binario sin signo de 16 bits. La dirección de CPU es la misma que se ha almacenado por STORE CPU ADDRESS (ALMACENAR DIRECCIÓN DE CPU) cuando el programa es ejecutado por una máquina que funciona en el modo básico.

Nota de programación:

- 20 Múltiples CPU en la misma configuración tienen el mismo código de secuencia, y es necesario usar otra información, tal como la dirección de CPU para establecer una única identidad de CPU. El código de secuencia devuelto para una CPU de máquina básica y una CPU de partición lógica es idéntico y tiene el mismo valor que el código de secuencia devuelto para la configuración de máquina básica.

SYSIB 1.2.2 (CPU de Máquina Básica)

- 25 El campo de formato en el byte 0 de la palabra 0 determina el formato del SYSIB. Cuando el campo de formato tiene un valor de cero, el SYSIB 1.2.2 tiene una implantación de formato 0 como se ha mostrado en la fig. 8. Cuando el campo de formato tiene un valor de uno, el SYSIB 1.2.2 tiene una implantación de formato 1 como se ha mostrado en la fig. 9.

Reservado:

- 30 Cuando el campo de formato contiene un valor no-cero, el contenido de los bytes 1-3 de la palabra 0 y de las palabras 1-6 es reservado y almacenado como ceros. Cuando el campo de formato contiene un valor de uno, el contenido del byte 1 de la palabra 0 y de las palabras 1-6 es reservado y almacenado como ceros. Cuando se necesitan menos de 64 palabras para contener la información para todas las CPU, la parte de la SYSIB que sigue a la lista de factor de ajuste en un SYSIB de formato 0 o la lista de factor de ajuste alternativo en un SYSIB de formato 1, hasta la palabra 63 son reservadas y almacenadas como ceros. El contenido de las palabras 64-1023 es reservado y puede ser almacenado como ceros o puede permanecer sin cambios. Cuando se necesitan 64 o más palabras para contener la información para todas las CPU, la parte del SYSIB que sigue a la lista del factor de ajuste en un SYSIB de formato 0 o a la lista de factor de ajuste alternativo en un SYSIB de formato 1, hasta la palabra 1023 son reservadas y pueden ser almacenadas como ceros o pueden permanecer sin cambios.

Formato:

- 40 El byte 0 de la palabra 0 contiene un entero binario sin signo de 8 bits que especifica el formato de SYSIB 1.2.2.

Desplazamiento de Capacidad de CPU Alternativa:

Cuando el campo de formato tiene un valor de 1, los bytes 2-3 de la palabra 0 contienen un entero binario sin signo de 16 bits que especifica el desplazamiento en bytes del campo de capacidad de CPU alternativa en el SYSIB.

Capacidad de CPU Secundaria:

- 45 La palabra 7 contiene un entero binario sin signo de 32 bits que, cuando no es cero, especifica una capacidad secundaria que puede ser aplicada a ciertos tipos de CPU en la configuración. No hay descripción formal del algoritmo usado para

generar este entero, excepto que es el mismo que el algoritmo usado para generar la capacidad de CPU. El entero es usado como una indicación de la capacidad de una CPU con relación a la capacidad de otros modelos de CPU, y también con relación a la capacidad de otros tipos de CPU dentro de un modelo. El valor de capacidad se aplica a cada una de las CPU de uno o más tipos aplicables de CPU en la configuración. Es decir, todas las CPU en la configuración de un tipo o tipos aplicables tienen la misma capacidad. Cuando el valor es cero, todas las CPU de cualquier tipo de CPU en la configuración tienen la misma capacidad, como se ha especificado por la capacidad de CPU. La capacidad de CPU secundaria puede o no ser el mismo valor que las CPU en la configuración. Es decir, todas las CPU en la configuración tienen la misma capacidad alternativa.

Capacidad de CPU:

Si los bits 0-8 de la palabra 8 son cero, la palabra contiene un entero (I) binario sin signo de 32 bits en el intervalo $0 < I < 2^{23}$ que especifica la capacidad de una de las CPU en la configuración. Si los bits 0-8 de la palabra 8 no son cero, la palabra contiene un número de formato corto de coma flotante binario de 32 bits en lugar de un entero binario sin signo de 32 bits. Independientemente de la codificación, un valor inferior indica una capacidad de CPU proporcionalmente mayor. Más allá de eso, no hay una descripción formal del algoritmo usado para generar este valor. El valor es usado como una indicación de la capacidad de la CPU con relación a la capacidad de otros modelos de CPU. El valor de capacidad se aplica a cada una de las CPU no secundarias en la configuración. Es decir, todas las CPU no secundarias en la configuración tienen la misma capacidad.

Cómputo total de CPU:

Los bytes 0 y 1 de la palabra 9 contienen un entero binario sin signo de 16 bits que especifica el número total de CPU en la configuración. Este número incluye todas las CPU en el estado configurado, el estado estacionario, o el estado reservado.

Cómputo de CPU Configurada:

Los bytes 2 y 3 de la palabra 9 contienen un entero binario sin signo de 16 bits que especifica el número de CPU que están en el estado configurado. Una CPU está en el estado configurado cuando está en la configuración y disponible para ser usada para ejecutar programas.

Cómputo de CPU Estacionaria:

Los bytes 0 y 1 de la palabra 10 contienen un entero binario sin signo de 16 bits que especifica el número de CPU que están en el estado estacionario. Una CPU está en el estado estacionario cuando está en la configuración, no está disponible para ser usada para ejecutar programas, y puede ser hecha disponible emitiendo instrucciones para colocarla en el estado configurado.

Cómputo de CPU Reservada:

Los bytes 2 y 3 de la palabra 10 contienen un entero binario sin signo de 16 bits que especifica el número de CPU que están en el estado reservado. Una CPU está en el estado reservado cuando está en la configuración, no está disponible para ser usada para ejecutar programas y no puede ser hecha disponible emitiendo instrucciones para colocarla en el estado configurado. (Puede ser posible colocar una CPU reservada en el estado estacionario o configurado por medio de acciones manuales).

Factores de Ajuste de la Capacidad de CPU de Multitratamiento:

Comenzando con los bytes 0-1 de la palabra 11, el SYSIB contiene una serie de campos, cada uno de los cuales contiene un entero binario sin signo de 16 bits usado para formar un factor de ajuste (fracción) para el valor contenido en el campo de capacidad de CPU. Tal fracción es desarrollada usando el valor (V) del primer campo de dos bytes de acuerdo con uno de los siguientes métodos:

- Si V es del orden de $0 < V \leq 100$, un denominador de 100 es indicado que produce una fracción de $V/100$
- Si V es del orden de $101 \leq V < 255$, un denominador de 255 es indicado que produce una fracción de $V/255$
- Si V es del orden de $255 < V < 65.536$, un denominador de 65.536 es indicado que produce una fracción de $V/65.536$

Así, la fracción representada por cada campo de dos bytes es a continuación desarrollada dividiendo el contenido de un campo de dos bytes por el denominador indicado. El número de campos de factor de ajuste es una unidad menor que el número de CPU especificadas en el campo de cómputo total de CPU. Los campos de factor de ajuste corresponden a

configuraciones con números crecientes de CPU en el estado configurado. El primer campo de factor de ajuste corresponde a una configuración con dos CPU en el estado configurado. Cada campo de factor de ajuste sucesivo corresponde a una configuración con un número de CPU en el estado configurado que es uno más que el del campo precedente.

5 Capacidad de CPU Alternativa:

Cuando el campo de formato tiene un valor de uno, si los bits 0-8 de la palabra N son cero, la palabra contiene un entero (I) binario sin signo de 32 bits del orden de 2^{31} que especifica la capacidad anunciada de una de las CPU en configuración. Si los bits 0-8 de la palabra N no son cero, la palabra contiene un número de formato corto de coma flotante binario de 32 bits en vez de un entero binario sin signo de 32 bits. Independientemente de la codificación, un valor inferior indica una capacidad de CPU proporcionalmente mayor. Más allá de eso, no hay descripción formal del algoritmo usado para generar este valor. El valor es usado como una indicación de la capacidad anunciada de la CPU con relación a la capacidad anunciada de otros modelos de CPU. El valor de capacidad alternativa se aplica a cada una de las CPU en la configuración. Es decir, todas las CPU en la configuración tienen la misma capacidad alternativa.

Factores de Ajuste de la Capacidad de CPU de Multitratamiento Alternativo.

15 Comenzando con los bytes 0 y 1 de la palabra N+1, el SYSIB contiene una serie de campos de dos bytes contiguos, cada uno de los cuales contiene un entero binario sin signo de 16 bits usado para formar un factor de ajuste (fracción) para el valor contenido en el campo de capacidad de CPU alternativo. Tal fracción es desarrollada usando el valor (V) del primer campo de 2 bytes de acuerdo con uno de los siguientes métodos:

- Si V es del orden de $0 < V \leq 100$, un denominador de 100 es indicado que produce una fracción de $V/100$
- 20 • Si V es del orden de $101 \leq V < 255$, un denominador de 255 es indicado que produce una fracción de $V/255$
- Si V es del orden de $255 < V < 65.536$, un denominador de 65.536 es indicado que produce una fracción de $V/65.536$

Así, la fracción representada por cada campo de dos bytes es a continuación desarrollada dividiendo el contenido de un campo de dos bytes por el denominador indicado. El número de campos de factor de ajuste alternativo es una unidad menor que el número de CPU especificado en el campo de cómputo total de CPU. Los campos de factor de ajuste alternativo corresponden a configuraciones con números crecientes de CPU en el estado configurado. El primer campo de factor de ajuste alternativo corresponde a una configuración con dos CPU en el estado configurado. Cada campo de factor de ajuste alternativo sucesivo corresponde a una configuración con un número de CPU en el estado configurado que es uno más que el del campo precedente.

30 SYSIB 15.1.2 (Topología de Configuración)

SYSIB 15.1.2 tiene el formato mostrado en la fig. 10. Los campos tienen los significados siguientes:

Reservado: el contenido de los bytes 0-1 de la palabra 0, byte 2 de la palabra 2 y palabra 3 están reservados y son almacenados como ceros. El contenido de las palabras N-1023 está reservado y puede ser almacenado como ceros o puede permanecer sin cambios.

35 Longitud: Los bytes 2-3 de la palabra 0 contienen un entero binario sin signo de 16 bits cuyo valor es el cómputo de bytes del SYSIB 15.1.2 completo. La longitud sólo de la lista de topología es determinada sustrayendo dieciséis del valor de longitud en los bytes 2-3 de la palabra 0. N en la fig. 10 es determinado evaluando la fórmula $N = \text{Longitud}/4$.

40 Mag 1-6: La palabra y bytes 0-1 de palabra constituyen seis campos de un byte donde el contenido de cada byte indica el número máximo de entradas de lista de topología (TLE) de tipo de contenedor o TLE de tipo CPU en el nivel de anidado correspondiente. Las TLE de tipo CPU son siempre encontradas solamente en el nivel de Mag1. Adicionalmente, el valor de Mag1 también especifica el número máximo de CPU que pueden ser representadas por una TLE de tipo contenedor del nivel Mag2. Cuando el valor del nivel de anidado es mayor que uno, niveles que contienen anidado superiores al nivel de Mag1 son ocupados solamente por TLE de tipo contenedor. Un cambio dinámico en la topología puede alterar el número de TLE y el número de CPU en el nivel 45 Mag1, pero los límites representados por los valores de los campos Mag1-6 no cambian dentro de una familia de máquinas de modelo.

La topología en una lista estructurada de entradas en las que una entrada define una o más CPU u otras está implicada con la estructura de anidado. Lo siguiente ilustra el significado de los campos de magnitud:

Cuando todas las CPU de la máquina son peers y no existe organización de contención, que no sea la de la totalidad del propio complejo de tratamiento central, el valor del nivel de anidado es 1, Mag1 es el único campo de magnitud distinta de cero, y el número de TLE de tipo CPU almacenados no excede del valor de Mag1.

- 5 Cuando todas las CPU de la máquina están subdivididas en grupos peers de tal modo que existe un nivel de contención, el valor del nivel de anidado es 2, Mag1 y Mag2 son solamente campos de magnitud distintos de cero, el número de TLE de tipo contenedor almacenado no excede del valor de Mag2, y el número de TLE de tipo CPU dentro de cada contenedor no excede del valor de Mag1.

Los bytes de Mag3-6 resultan usados similarmente (prosiguiendo en una dirección de derecha a izquierda) cuando el valor de nivel de anidado cae en el intervalo 3-6.

- 10 MNest: El byte 3 de la palabra 2 especifica el nivel de anidado de la topología a la que puede ser extendida la configuración mientras se continúa permitiendo que sobreviva el programa invitado. El máximo valor MNest es 6; el mínimo es uno. Si MNest es uno, no hay estructura de anidado de TLE real, Mag1 es el único campo no-cero en el intervalo Mag1-6, y solamente están representadas las TLE de tipo CPU en la lista de topología. El valor de MNest indica el número de valores de magnitud distinta de cero que comienzan con el campo de magnitud en el byte de la palabra 2
- 15 (Mag1), prosiguiendo a la izquierda cuando MNest es mayor que uno, y con los campos de magnitud restantes almacenados como ceros.

El valor de MNest es el anidado máximo posible. Ningún cambio de configuración dinámica excede de este límite.

- 20 Lista de topología: Las palabras de la fig. 10 en el intervalo de 4 a N-1 especifican una lista de una o más entradas de lista de topología (TLE). Cada TLE es un campo de ocho bytes o dieciséis bytes; así N es un número par de palabras, y un corolario es que una TLE siempre comienza con un límite de doble palabra.

Entradas de lista de topología: La primera TLE en la lista de topología comienza en un nivel de anidado igual a MNest-1. La lista de topología completa representa la configuración del emisor de la instrucción STSI que especifica SYSIB 15.1.2; no se usa una entrada de TLE de contenedor más exterior ya que sería redundante con la lista completa, y la configuración completa. Por ello, el mayor nivel de anidado puede tener más de un único contenedor peer.

- 25 Las figs. 11 y 12 ilustran los tipos de TLE, en los que los campos tienen las siguientes definiciones:

Nivel de Anidado (NL): Byte 0 de palabra 0 especifica el nivel de anidado de TLE.

NL	Significado
0	La TLE es una TLE de tipo CPU
1-5	La TLE es una TLE de tipo contenedor
06-FF	Reservado

- 30 1-5 La TLE es una TLE de tipo contenedor

La primera TLE de tipo contenedor almacenada en una lista de topología o en un recipiente contenedor padre tiene una ID de contenedor del orden de 1-255. Si existen contenedores hermanos dentro del mismo padre, prosiguen en orden ascendente de ID del contenedor, que pueden o no ser consecutivas, a un valor máximo de 255.

- 35 06-FF Reservado

Las TLE hermanas tienen el mismo valor de nivel de anidado que es equivalente o bien al valor del nivel de anidado menos uno de la TLE madre inmediata, o bien al valor de MNest menos uno, debido a que el padre inmediato es la lista de topología en vez de una TLE.

- 40 Reservado, 0: Para una TLE de tipo contenedor, los bytes 1-3 de la palabra 0 y los bytes 0-2 de la palabra 1 son reservados y almacenados como ceros. Para una TLE de tipo CPU, los bytes 1-3 de la palabra 0 y los bytes 0-4 de la palabra 1 son reservados y almacenados como ceros.

ID de contenedor:

El byte de la palabra 1 de una TLE de tipo contenedor especifica un entero binario no-cero sin signo de 8 bits cuyo valor es el identificador del contenedor. La ID del contenedor para una TLE es única dentro del mismo contenedor padre.

- 45 Dedicado (D):

El bit 5 de la palabra 1 de una TLE de tipo CPU, cuando es uno, indica que la una o más CPU representadas por la TLE están dedicadas. Cuando D es cero, la una o más CPU de la TLE no están dedicadas.

Polarización (PP):

- 5 Los bits 6-7 de la palabra 1 de una TLE de tipo CPU especifican el valor de polarización y, cuando la polarización es vertical, el grado de polarización vertical también llamado autorización (alta, media, baja) de las CPU correspondientes representada por la TLE. Se usan los siguientes valores:

Significado de PP:

- 0 La una más CPU representadas por la TLE están polarizadas horizontalmente.
- 1 La una más CPU representadas por la TLE están polarizadas verticalmente. La autorización es baja.
- 10 2 La una más CPU representadas por la TLE están polarizadas verticalmente. La autorización es media.
- 3 La una o más CPU representadas por la TLE están polarizadas verticalmente. La autorización es alta.

- 15 La polarización es sólo significativa en una configuración de multitratamiento lógica y virtual que usa procesadores anfitriones compartidos y accede a cómo es aplicado el recurso asignado a una configuración a través de las CPU de la configuración. Cuando la polarización horizontal está en efecto, cada CPU de una configuración tiene garantizada aproximadamente la misma cantidad de recursos. Cuando la polarización vertical está en efecto, las CPU de una configuración son clasificadas en tres niveles de autorización de recursos: alta, media y baja.

- 20 Tanto la reposición del subsistema como la ejecución satisfactoria del orden de arquitectura de conjunto de SIGP que especifican que el modo SA/390 tuvo en una configuración y la totalidad de sus CPU en polarización horizontal. Las CPU afectadas inmediatamente son aquellas que están en el estado configurado. Cuando una CPU en el estado estacionario está configurada, adquiere la polarización actual de la configuración y provoca un cambio de topología de esa configuración que ha de ser reconocido.

Una CPU dedicada está polarizada bien en horizontal o bien verticalmente. Cuando una CPU dedicada está polarizada verticalmente, la autorización es siempre alta. Así, cuando D es uno, PP es o bien un 00 binario o bien un 11 binario.

Tipo CPU:

- 25 El byte 1 de la palabra 1 de una TLE de tipo CPU especifica un entero binario sin signo de 8 bits cuyo valor es el tipo de CPU de una o más CPU representadas por la TLE. El valor de tipo CPU especifica o bien un tipo de CPU primaria o cualquiera de los tipos posibles de CPU secundaria.

Origen de Dirección de CPU:

- 30 Los bytes 2-3 de la palabra 1 de una TLE de tipo CPU especifican un entero binario sin signo de 16 bits cuyo valor es la dirección de CPU de la primera CPU en el intervalo de las CPU representado por la máscara de CPU, y cuya presencia está representada por el valor de la posición 0 de bit en la máscara de CPU. Un origen de dirección de CPU es divisible de modo uniforme por 64. El valor de un origen de dirección de CPU es el mismo que el almacenado por la instrucción STORE CPU ADDRESS (ALMACENAR DIRECCIÓN DE CPU) (STAP) cuando es ejecutada en la CPU representada por la posición 0 de bit en la máscara de CPU.

- 35 Máscara de CPU:

- 40 Las palabras 2-3 de una TLE de tipo CPU especifican una máscara de 64 bits donde cada posición de bit representa una CPU. El valor del campo de origen de la dirección de CPU más una posición de bit en la máscara de CPU es igual a la dirección de CPU para la CPU correspondiente. Cuando un bit de una máscara de CPU es cero, la CPU correspondiente no está representada por la TLE. La CPU o bien no está en la configuración o de otro modo debe estar representada por otra TLE de tipo CPU.

Cuando una máscara de CPU es uno, la CPU correspondiente tiene los valores de atributo modificador especificados por la TLE, está en la topología de la configuración, y no está presente en ninguna otra TLE de la topología.

- 45 Así, por ejemplo, si el origen de dirección de la CPU es un valor de 64, y la posición 15 de bit de la máscara de CPU es uno, la CPU 79 está en la configuración y tiene el tipo de CPU, polarización, autorización, y dedicación según se ha especificado por la TLE.

Ordenación de TLE:

Los atributos modificadores que se aplican a una TLE de tipo CPU son el tipo de CPU, polarización, autorización y dedicación. La polarización y la autorización (para polarización vertical) son tomadas como un único atributo, pero con cuatro valores posibles (horizontal, vertical alta, vertical media y vertical baja).

- 5 Una única TLE de CPU es suficiente para representar tantas como 64 CPU que todas tienen los mismos valores de atributo modificador.

- 10 Cuando existen más de 64 CPU, o el intervalo completo de direcciones de CPU no está cubierto por un único origen de dirección de CPU, y los atributos modificadores son constantes, una TLE de CPU hermana separada es almacenada para cada origen de dirección de CPU, cuando sea necesario, en orden ascendente y origen de dirección de CPU. Cada una de tales TLE almacenada tiene una CPU representada. La colección de una más de tales TLE de CPU es llamada conjunto de TLE de CPU.

Cuando existen múltiples tipos de CPU, un conjunto de TLE de CPU es almacenado para cada uno, en orden ascendente de tipo de CPU.

- 15 Cuando existen múltiples valores de polarización y autorización, un conjunto de TLE de CPU separado es almacenado para cada uno, en orden descendente de valor de polarización y grado (vertical alta, media, baja, luego horizontal). Cuando están presentes, todos los conjuntos de TLE de CPU de polarización de un tipo de CPU dado son almacenados antes del siguiente conjunto de TLE de CPU del siguiente tipo.

- 20 Cuando existen tanto CPU dedicadas como no dedicadas, un conjunto de TLE de CPU separado es almacenado para cada una, apareciendo dedicadas antes que no dedicada. Todas las TLE son ordenadas suponiendo una primera profundidad transversal en la que el orden de clasificación de mayor a menor es el siguiente:

1. Tipo de CPU

- a. Valor de tipo CPU más bajo
- b. Valor de tipo CPU más alto

2. Polarización-Autorización

- 25
- a. Vertical alta
 - b. Vertical media
 - c. Vertical baja
 - d. Horizontal

3. Dedicación

- 30
- a. Dedicada
 - b. No dedicada

La ordenación por origen de dirección de CPU y atributos modificadores de TLE de CPU hermana dentro de un contenedor padre es hecha de acuerdo a la siguiente lista, que va desde más alta a más baja.

- 35
- 1. Conjunto de CPU-TLE de menor valor de tipo CPU, vertical alta, dedicada
 - 2. Conjunto de CPU-TLE de menor valor de tipo CPU, vertical alta, no dedicada
 - 3. Conjunto de CPU-TLE de menor valor de tipo CPU, vertical media, no dedicada
 - 4. Conjunto de CPU-TLE de menor valor de tipo CPU, vertical baja, no dedicada
 - 5. Conjunto de CPU-TLE de menor valor de tipo CPU, horizontal, dedicada
 - 6. Conjunto de CPU-TLE de menor valor de tipo CPU, horizontal, no dedicada
 - 40 7. Conjunto de CPU-TLE de mayor valor de tipo CPU, vertical alta, dedicada

8. Conjunto de CPU-TLE de mayor valor de tipo CPU, vertical alta, no dedicada
9. Conjunto de CPU-TLE de mayor valor de tipo CPU, vertical media, no dedicada
10. Conjunto de CPU-TLE de mayor valor de tipo CPU, vertical baja, no dedicada
11. Conjunto de CPU-TLE de mayor valor de tipo CPU, horizontal, dedicada
- 5 12. Conjunto de CPU-TLE de mayor valor de tipo CPU, horizontal, no dedicada

Otras Reglas de TLE:

Una TLE de tipo contenedor está situada en niveles de anidado del orden de 1-5.

Una TLE de tipo CPU está situada en el nivel 0 de anidado.

10 El número de TLE de tipo contenedor hermano en una lista de topología o un contenedor padre dado no excede del valor del byte de magnitud (Mag2-6) del nivel de anidado correspondiente a los hermanos.

El número de CPU representado por una o más TLE de tipo CPU del contenedor padre no excede del valor del byte de magnitud Mag1.

El contenido de una TLE es definido como sigue:

15 Si una TLE es una TLE de tipo contenedor, el contenido es una lista que sigue inmediatamente a la TLE padre, comprendido de una o más TLE hijas, y cada TLE hija tiene un nivel de anidado menor en una unidad que el nivel de anidado de la TLE padre o final de lista de topología. Si una TLE es una TLE de tipo CPU, el contenido es una o más CPU, como identificado por los otros campos de una TLE de CPU.

Cuando la primera TLE en un nivel de anidado es una entrada de CPU, el nivel 0 de anidado máximo ha sido alcanzado.

Nota de Programación:

20 Un proceso de examen posible de una lista de topología está descrito. Antes de que haya comenzado un examen de una lista de topología, el indicador de TLE actual es inicializado para referenciar la lista de TLE superior en la lista de topología, el indicador de TLE anterior es inicializado a nulo, y a continuación las TLE son examinadas en un orden descendente.

25 Cuando un examen de lista de topología prosigue, el indicador de TLE actual es hecho avanzar incrementando el indicador de TLE actual por el tamaño de la TLE actual al que indica o apunta. Una TLE de tipo contenedor es hecha avanzar añadiendo ocho al indicador de la TLE actual. Una TLE de tipo CPU es hecha avanzar añadiendo dieciséis al indicador de la TLE actual. El proceso de hacer avanzar el indicador de TLE actual incluye ahorrar su valor cuando el indicador de TLE anterior justo anterior es aumentado. El examen de TLE no es realizado si la lista de topología no tiene TLE.

El proceso de examen está bosquejado en las siguientes operaciones:

30 1. Si el nivel de anidado de TLE actual es cero, y el nivel de anidado de la TLE anterior es nulo o uno, la TLE actual representa la primera TLE de tipo CPU de un grupo de una o más TLE de tipo CPU. El programa debería realizar cualquier acción que sea apropiada para cuando un nuevo grupo de una o más CPU es observado en primer lugar. Ir a la operación 5.

35 2. Si el nivel de anidado de la TLE actual es cero, y el nivel de anidado de la TLE anterior es cero, la TLE actual representa una TLE de tipo CPU subsiguiente de un grupo de una o más TLE de tipo CPU que representan hermanas de las CPU previamente observadas en las operaciones 1 ó 2. El programa debería realizar cualquier acción que sea apropiada para cuando el tamaño de un grupo hermano existente de una o más CPU es incrementado. Ir a la operación 5.

40 3. Si el nivel de anidado de la TLE actual no es cero, y el nivel de anidado de la TLE anterior es cero, la TLE anterior representa una último TLE de tipo CPU o solo una de un grupo de una o más TLE de tipo CPU. El programa debería realizar cualquier acción que sea apropiada para cuando un grupo existente de una o más CPU es completado. Ir a la operación 5.

4. Ir a la operación 5.

Por eliminación, éste sería el caso cuando el nivel de anidado de la TLE actual no es cero, y el nivel de anidado de la TLE anterior no es cero. Si el nivel de anidado de la TLE anterior es menor que el nivel de anidado de la

TLE anterior, la dirección de la lista de topología transversal es hacia una TLE de tipo CPU. Si el nivel de anidado de la TLE actual es mayor que el nivel de anidado de la TLE anterior, la dirección de la lista de topología transversal se aleja de una TLE de tipo CPU. Las TLE de tipo CPU están siendo atravesadas conduciendo, o bien (1) a otro grupo de TLE de tipo CPU que son un grupo separado en la topología total, o bien (2) al final de la lista de topología. En cualquier caso, no se requiere un tratamiento particular más allá de hacer avanzar a la siguiente TLE.

5. Avanzar a la siguiente posición de TLE basado en el tipo de TLE actual. Si el indicador de TLE actual avanzado es equivalente al final de la lista de topología:

a. No existen más TLE de ningún tipo

b. Si el nivel de anidado de la TLE anterior es cero, el programa debería realizar cualquier acción que sea apropiada para cuando se ha completado un grupo existente de una o más CPU.

c. El examen está completo.

De otro modo ir a la operación 1.

En una puesta en práctica ejemplar, fig. 16, un sistema de ordenador comprende dos cuadros físicos (Cuadro 1 y Cuadro 2). Cada cuadro contiene dos placas lógicas (Placa 1 y Placa 2), almacenamiento principal (Memoria), y adaptadores de I/O (Canales 1 de I/O y Canales 2 de I/O) para comunicar con dispositivos periféricos y redes. Cada Placa contiene dos módulos de múltiples chips (Mod 1, Mod 2, Mod 3, y Mod 4). Cada módulo de múltiples chips contiene dos procesadores (CPU1, CPU2, CPU3, CPU4, CPU5, CPU6, CPU7 y CPU8). Cada módulo contiene también una Caché de nivel 2 (Caché 1, Caché 2, Caché 3 y Caché 4). Cada procesador (Unidad de Tratamiento Central o CPU), incluye una Caché de nivel 1 y una Memoria Tampón de Traslación de Búsqueda a un Lado (TLB). Las memorias TLB acceden a la información de traslación durante la traslación de dirección dinámica.

De acuerdo con el invento, figs. 17, 18 y 19, los componentes del sistema de ordenador están asignados a "contenedores" de acuerdo con la proximidad. Cada módulo está asignado a los contenedores más interiores 6, 7, 8 y 9 cuando las CPU de cada módulo están en proximidad más cercana entre sí. Como los módulos están empaquetados en placas, los módulos de las placas respectivas son asignados a los contenedores 4 y 5 del siguiente nivel. El siguiente agrupamiento de mayor proximidad es el Cuadro. Las placas de un cuadro son asignadas a un contenedor que representa el cuadro (contenedores 2 y 3). El sistema completo está representado por el contenedor 1.

Puede verse que dos CPU del mismo módulo tales como CPU1 y CPU2 están en relación o distancia topológica más próxima entre sí y que ambas residen en un contenedor (contenedor 6) y no es atravesado ningún límite de contenedor cuando sólo están implicadas esas CPU. Sin embargo, si CPU1 y CPU8 están implicadas, hay cuatro límites de contenedor atravesados. CPU1 en el contenedor 6 está separada por el contenedor 4, 5 y 9 de la CPU8. Por ello, conociendo la estructura del contenedor, un usuario puede dar un vista de la topología del sistema.

Desde luego, en un sistema dividido en particiones lógicamente, las CPU pueden ser compartidas entre sistemas operativos como se ha descrito previamente. Por ello, si una partición lógica tenía asignadas 3 CPU lógicas, cada CPU lógica asignada al 20% de cada una de las tres CPU reales, la partición ejecutaría mejor si las 3 CPU reales estuvieran en proximidad más estrecha entre sí ya que la comunicación entre las CPU y los recursos de CPU (caché y memoria por ejemplo) se realizaría mejor. En este ejemplo, la CPU1 y la CPU2 en una partición experimentarían menor descarte de líneas caché en cada Caché 1 que si las dos CPU fueran CPU1 y CPU8,

En el ejemplo es creada una partición que incluye CPU1, CPU2 y CPU3. Un programa que funciona en la partición emite una instrucción STSI y una tabla de SYSIB 15.1.2 (fig. 10) es devuelta al programa. En este caso la partición comprende el contenedor 6 y el contenedor 7 y por ello hay dos niveles de anidado. Los valores de la tabla SYSIB son:

El campo MNest está configurado a 4 que indica que la topología incluye cuatro niveles de anidado, que es un anidado máximo absoluto para el modelo, y puede o no ser totalmente explotado sobre cualquier lista de topología arbitraria, dependiendo de la asignación de recursos a la configuración de invitado que sale de STSI, y cómo son asignados los recursos a otras configuraciones de invitado del sistema.

El campo Mag 1 es configurado a 2 que indica que 2 CPU están disponibles en el primer nivel más primitivo.

El campo Mag 2 es configurado a 2 que indica que 2 contenedores de segundo nivel (Módulo) están disponibles.

El campo Mag 3 es configurado a 2 que indica que 2 de tercer nivel (placas) están disponibles.

El campo Mag 4 es configurado a 2 que indica que 2 de cuarto nivel (cuadros) están disponibles.

En un ejemplo, las tres CPU están contenidas en dos módulos de la misma placa, por ello son proporcionadas las siguientes cuatro entradas de TLE:

1. NL = 1; CtnrID = 1 (contenedor 6)
2. NL = 0; CPU tipo = 1 (el tipo de cpu); CPU Addr Origin = 0; CPU Mask 0.....11 (PCU1 y CPU2 de la CPU accedida)
3. NL = 1; CtnrID = 2 (contenedor 7)
4. NL = 0; CPU tipo = 1 (el tipo de cpu); CPU Addr Origin = 4; CPU Mask 0.....10 (CPU3 de la CPU accedida)

Así, el programa tiene una representación de la topología física basada en el contenedor y en las TLE de CPU devueltos.

PERFORM TOPOLOGY FUNCTION (PTF)

- 10 Con referencia a la fig. 13, una instrucción de PERFORM TOPOLOGY FUNCTION (PTF) comprende preferiblemente un código de operación y un campo de registro R1 que identifica un primer operando. El contenido del registro general identificado por R1 especifica un código de función en posiciones 56-63 de bit, como se ha ilustrado en la fig. 14.

Los códigos de función definidos son como sigue:

FC	Significado
0	Solicitud de polarización horizontal
1	Solicitud de polarización vertical
3	Comprobación del estado de cambio de topología

Los códigos de función indefinidos en el intervalo 0-255 están reservados para extensiones futuras.

- 20 Al terminar, si el código de condición 2 está configurado, un código de razón es almacenado en las posiciones 48-55 de bit del registro general R1. Los bits 16-23 y 28-31 de la instrucción son ignorados.

Operación de Código 0 de Función:

La ejecución se termina con el código de estado para cualquiera de las siguientes razones y (códigos de razón):

Sin razón especificada (0).

- 25 La configuración de solicitud está ya polarizada horizontalmente (1).

Un cambio de topología está ya en proceso (2).

De otro modo, un proceso es iniciado para colocar todas las CPU en la configuración a polarización horizontal.

La terminación del proceso es asíncrona con respecto a la ejecución de la instrucción y puede ser completada cuando se completa la ejecución de la instrucción.

- 30 Código de Estado Resultante:

0 Cambio de topología iniciado

1 –

2 Solicitud rechazada

3 –

- 35 Operación de Código 1 de Función

La ejecución se termina con el código de estado 2 para cualquiera de las siguientes razones (códigos de razón):

Sin razón especificada (0).

La configuración de solicitud está ya polarizada verticalmente (1).

Un cambio de topología está ya en proceso (2).

De otro modo, un proceso es iniciado para colocar todas las CPU en la configuración a polarización vertical. La terminación del proceso es asíncrona con respecto a la ejecución de la instrucción y puede o no ser completada cuando se completa la ejecución de la instrucción.

5

Código de Estado Resultante:

0 Cambio de topología iniciado

1 –

2 Solicitud rechazada

10 3 –

Operación de Código 2 de Función: El estado de relación de cambio de topología pendiente es comprobado, y la instrucción se completa con la configuración del código de estado.

Código de Estado Resultante:

0 Topología de relación de cambio de topología no pendiente

15 1 relación de cambio pendiente

2 –

3 –

Un cambio de topología es cualquiera alteración tal que el contenido de una SYSIB 15.1.2 sería diferente del contenido de la SYSIB 15.1.2 anterior al cambio de topología.

20 Un estado de relación de cambio de topología pendiente es creado cuando se completa un proceso de cambio de topología. Un estado de relación de cambio de topología pendiente es borrado para la configuración cuando se realiza cualquiera de las siguientes cosas:

La ejecución de PERFORM TOPOLOGY FUNCTION especifica el código 2 de función que se completa con el código 1 de estado.

25 STORE SYSTEM INFORMATION para SYSIB 15.1.2 es ejecutado satisfactoriamente por cualquier CPU en la configuración.

La reposición del subsistema es realizada.

Condiciones especiales:

30 Si las posiciones 0-55 de bit del registro general R1 no son ceros, se reconoce una excepción de la especificación. Si se ha especificado un código de función indefinida, se reconoce una excepción de la especificación.

Excepciones del Programa:

Operación (equipo de configuración de topología no está instalado)

Operación privilegiada

Especificación

35 UN ENTORNO EJEMPLAR DE COMPUTADORA CENTRAL

Cuando las arquitecturas altas de servidor final aumentan el número de procesadores físicos y las velocidades del procesador continúan mejorando, el “nido” de procesadores necesita construir grandes máquinas continúa para que se hagan bloques de construcción menores que el arco más modal en la naturaleza. Por ejemplo, mientras la caché L2 de una máquina z990 o z9 es una caché totalmente coherente, los modelos completamente popularizados tienen realmente cuatro (4) L2 separadas que están conectadas por un tejido para presentar la apariencia de una sola caché L2. La

40

penalización para salirse del nodo para resolver una caché errónea continua aumentando. Por ejemplo, resolver un error de L1 en una L2 remota es más caro que resolverlo en la L2 local. Equivocarse en una CP privada, usualmente en chip, la caché L1 es cara de poner en marcha y tener que ir todo el camino fuera a la memoria puede parecer una eternidad. El aumento de velocidad de memoria y las conexiones a ella no se conservan en paz con aumentos de la velocidad del procesador. Mientras se podría querer intentar empaquetar cada cosa más junto a “en el chip” o similar, el consumo de potencia y los problemas de refrigeración actúan contra esto.

Con la introducción de z990, LPAR resultó conocedor de la topología de la máquina y comenzó a optimizar la asignación de la partición lógica de CP y los recursos de almacenamiento a los recursos físicos. Las mejoras en las capacidades para volver a optimizar dinámicamente las asignaciones de recursos a particiones lógicas fueron introducidas con z9 GA-1 fundamentalmente en el soporte de reparación del libro concurrente.

El nuevo soporte descrito aquí accede al verdadero comienzo de tener software de OS zSeries que resulta conocedor de la topología de la máquina, presentada como una topología de partición lógica, para proporcionar entonces afinidad de distribución con respecto a la colocación de la CPU en la estructura del libro del CEC.

Se puede pensar que la LPAR de zSeries gestiona particiones lógicas compartidas hoy como si estuvieran polarizadas horizontalmente. Es decir, el peso del tratamiento para la partición lógica está dividido de igual modo entre todas las CP lógicas en línea en la partición lógica. Este soporte introduce una nueva forma opcional de polarización para gestionar las CP lógicas compartidas de una partición lógica llamada polarización vertical.

Cuando una partición lógica elige ejecutarse en modo vertical y el software emite una nueva instrucción para informar al hipervisor de zSeries de esto y el hipervisor cambiará cómo distribuye la partición lógica.

Dependiendo de la configuración de la partición lógica vertical, los procesadores lógicos tendrían una polaridad alta, media o baja. La polaridad describe la cantidad de procesadores lógicos verticales que están autorizados a compartir procesador físico. Los clientes definen pesos para las particiones lógicas que definen efectivamente la cantidad de ciclos de procesador físico a las que la partición lógica de una máquina está autorizada.

La polaridad es medida por la relación de un peso actual de partición lógica al número de procesadores lógicos configurados a la partición lógica. Los procesadores de alta polaridad están próximos a compartir el 100% de la CPU. Los procesadores de Polaridad Media comparten desde > 0 a 99% y los procesadores de polaridad baja comparten 0% (o están muy próximos a ello). Los CP de lógica de polaridad alta serán asignados a un procesador físico para ejecutarse sobre CP muy similares a CP dedicados pero el CP de polaridad alta compartida puede aún dar el recurso físico y permitir que otros CP compartidos usen sus ciclos de exceso. La clave aquí resulta entonces que el software vea la topología lógica e intente explotar los CP lógicos muy polarizados para sus colas de trabajo.

Por ejemplo, un cliente configura un procesador de tres vías con 2 particiones lógicas, cada una con 2 procesadores lógicos y cada una con un peso de 50. Si la primera partición lógica fuera definida en sí misma como vertical, tendría 1 CP lógico de polaridad alta y 1 CP lógico de polaridad media.

Obsérvese que cuando una partición lógica elige ejecutarse en modo vertical, la partición lógica completa se ejecuta en modo vertical. Esto incluye la totalidad de sus procesadores secundarios tales como los zAAP (IFA) y/o los zIIP. Es responsabilidad del cliente definir pesos para todos los tipos de procesadores para estas particiones lógicas que conseguirán el nivel deseado de tratamiento vertical para cada tipo.

Soporte de Topología de Partición Lógica

- Establecer infraestructura para una topología nodal de partición lógica.
- Hacer cualesquiera cambios necesarios para algoritmos de asignación nodal de LPA para particiones horizontales existentes necesarias para proporcionar una topología válida.
- Proporcionar simulación de instrucción para el nuevo bloque de información de topología de configuración para la instrucción de STORE SYSTEM INFORMATION (STSI) para proporcionar topología nodal de partición lógica a la partición lógica.
- Examinar cambios en la configuración física o lógica para determinar si es necesario cambiar una topología. Esto puede ocurrir cuando:
 - Un CP físico es añadido o retirado de la configuración
 - Un CP lógico es añadido o retirado de la configuración

- Una partición lógica es activada
 - Una partición lógica es desactivada
 - Los pesos de la partición lógica son cambiados desde el HMC/SE
 - La iniciación del software cambia un peso de partición lógica
- 5
- Una partición lógica es repuesta (cambiada a horizontal).
 - Una partición lógica cambia a modo ESA/390 (cambio a horizontal).

ALGORITMOS DEL ENTORNO

10 Una topología debe ser asignada a una por partición lógica cuando es activada en primer lugar y a continuación cualesquiera cambios en la topología nodal asignada a una partición lógica deben dar como resultado que sean notificados a la partición lógica. El resultado de la topología nodal debe ser conservado en una nueva estructura de datos conveniente para permitir preguntas más fáciles por el nuevo tratamiento de STSI así como para limitar el tratamiento del mejor modo posible cuando se han hecho cambios en la configuración. Esta nueva estructura también permite completar el tratamiento de cambios de topología en múltiples operaciones con la formación en serie requerida para cada operación sin introducir vistas inconsistentes de la topología a la partición lógica.

15 Asignación de Topología

De qué modo es elegida la topología lógica, no es importante para esta exposición. Baste decir que debe hacerse una determinación de cuantos de cada tipo de procesadores lógicos son necesarios y qué nodos o libros necesitan que le sean asignados. Para una partición en modo vertical, esto significa el cómputo de procesadores vertical alto, vertical medio, y vertical bajo para cada tipo de procesador.

20 Asignar valores de Polarización a CP lógicos

Una vez que se han determinado los cálculos anteriores, se hacen las asignaciones de polarización desde la dirección de CP lógico en línea más bajo del tipo cp al más alto del orden de (1) todos verticales altos, (2) todos verticales medios, y (3) todos verticales bajos. El orden en el que se hace esto es arbitrario y son posibles otros órdenes de selección.

Correspondencias de la instrucción "STORE SYSTEM INFORMATION"

25 Añadir 3 estructuras para hacer corresponder la respuesta de 15.1.2:

1. Correspondencia para STSI 15.1.2, el bloque de respuesta que corresponde a la topología

Dcl	1	syibk 1512	char(4096)	based(*),
	3	*	char(2),	
	3	syibk 1512_legth	fixed(16),	length
30	3	syibk 1512_mag6	fixed(8),	6° level nest
	3	syibk 1512_mag5	fixed(8),	5°
	3	syibk 1512_mag4	fixed(8),	4°
	3	syibk 1512_mag3	fixed(8),	3°
	3	syibk 1512_mag2	fixed(8),	2°, nodes
35	3	syibk 1512_mag1	fixed(8),	1°, cpus
	3	*	char(1),	
	3	syibk 1512_mnest	fixed(8),	nesting level
	3	*	char(4),	
	3	syibk 1512_topology_list	char(0),	topology list

2 Correspondencia para un TLE de tipo contenedor para STSI 15.1.2

Dcl	1	syibk_vcm_container	char(8)	based(*),	
		3	syibknl	fixed(8),	nesting level
5		3	*	char(3),	
		3	*	char(1),	
		3	*	char(2),	3 syibknl
		3	syibk_container_id	fixed(8),	node id

10	Dcl	1	syibk_vcm_cpu	char(16)	based(*),
		3	syibkn12	fixed(8),	nesting level
		3	*	char(3),	
		3	syibk_ded_polarization	bit(8),	vcm byte
		3	*	bit(5),	
15		3	syibk_dedicated	bit(1),	dedicated bit
		3	syibk_polarization	bit(2),	polarization bits
		3	syibk_cputype	fixed(8),	cpu type
		3	syibk_cpuaddrorg	fixed(16),	address origin
		3	syibk_cpumask	bit(64);	cpu mask entry

20 TOPBK – Topología de Partición

Un resumen de una topología de partición lógica es mantenido actual en este bloque por las rutinas de asignación nodales. Los datos en este bloque están ordenados de tal modo que el tratamiento STSI puede hacer un paso de la estructura completa para crear la respuesta de topología de partición lógica al programa, preservando el orden y separación de entradas de CPU cuando sea requerido por la arquitectura.

25 Consiste de una matriz tridimensional (nodo, tipo cp, clasificación de polarización) con una máscara de CPU de 64 bits por entrada.

Un segundo área de trabajo, TOP_WORKING, está incluida para usar en la actualización de la topología.

DECLARE

1 TOPBK BASED BDY(DWORD)

30 3 TOP_CURRENT,

5 TOPCPUMASK(1:MAXNODE, /*Cada uno de 1-4 nodos

0:CPUTMAX, /* índice de tipo CP

*/

35 0:3 /* 4 clasificaciones posibles de topología cuando la partición lógica es vertical. Hay solo 2 clasificaciones cuando la partición es horizontal.

*/

BIT(64), /* Máscara de las CPU lógicas que caen en esta clasificación.

*/

3 TOP_WORKING CHAR(LENGTH(TOP_CURRENT));/* Área de trabajo para construir nueva topología

*/

5 TO3PO – Traslación de Topología de Partición

Estas tablas de traslación “constante” son usadas por las rutinas de asignación nodal que construyen la topología (hechas corresponder por TOPBK) y tratamiento STSI que lee la topología.

/******

/* Matrices de traslación STSI: */

10 /******

DECLARE

1 TOPVERT(0:3)BIT(8) /* Para particiones verticales, traslada el índice de clasificación por encima de los valores de arquitectura D (dedicación) y PP (polarización) que han de ser devueltos para este grupo de CP en los datos de STSI.

*/

15 STATIC INIT('00000'b || '1'b || PPVH, /* Clasificación 0: Dedicada Vertical alta */

'00000'b || '0'b || PPVH, /* Clasificación 1: Compartida Vertical Alta */

'00000'b || '1'b || PPVM, /* Clasificación 2: Compartida Vertical Media */

'00000'b || '1'b || PPVL, /* Clasificación 3: Compartida Vertical Baja */

3* BIT(5), /* Sin usar, necesita alineación de byte para matrices de bits*/

20 3 TOPDPP BIT(3), /* Valores D, PP reales para usar */

1 TOPHOR(0:1)BIT(8) /* Para particiones horizontales, traslada el índice de clasificación a los valores de arquitectura D (dedicación) y PP (polarización) que han de ser devueltos para este grupo de CP en los datos de STSI. Obsérvese que sólo las dos primeras clasificaciones pueden ser usadas para una partición horizontal.

*/

25 STATIC INIT('00000'b || '1'b || PPH, /* Clasificación 0: Dedicado Horizontal */

'00000'b || '0'b || PPH, /* Clasificación 1: Compartido Vertical Alto */

3* BIT(5), /* Sin usar, necesita alineación de byte para matrices de bits*/

3 TOPDPP BIT(3), /* Valores D, PP reales para usar */

/******

30 /* Matriz de traslación NDI */

/******

1 TOPDPP2CLASS(0:7) FIXED /* Usada por las rutinas de asignación nodal para crear la información de topología. LPDPP es usado como un índice en esta matriz para determinar qué índice de clasificación debería usar el CP lógico. Esta matriz es usada tanto para particiones horizontales como verticales */

35 STATIC INIT(1, /* Compartida, Horizontal */

3, /* Compartida, vertical baja */

2, /* Compartida, vertical media */

1, /* Compartida, vertical alta */

0, /* Dedicada, horizontal */

0, /* No aplicable */

0, /* No aplicable */

5 0), /* Dedicada, vertical alta */

3* CHAR(4); /* La fuerza ha de ser un artículo no simple */

Bloque Procesador Lógico:

Una codificación de 2 bits de polarización de partición puede ser seguida para cada procesador lógico para reflejar su organización. Agrupar ésta con una indicación de 1 bit de dedicación permite una imagen de polaridad completa para un procesador lógico en 3 bits:

10 '''

3 LPDPP BIT(3), /* Polarización, incluyendo dedicación */

4 LPDED BIT(1), /* = 1, el CP lógico es dedicado */

4 LPPP BIT(2), /* Polarización del Procesador */

15 ...

/* Codificaciones para Polarización del Procesador */

PPH BIT(2) CONSTANT('00'B), /* Polarizado horizontalmente */

PPVL BIT(2) CONSTANT('01'B), /* Polarizado verticalmente – Bajo */

PPVM BIT(2) CONSTANT('10'B), /* Polarizado verticalmente – Medio */

20 PPVH BIT(2) CONSTANT('11'B), /* Polarizado verticalmente – Alto */

Actualizar Bloque de Topología

Borrar copia local de máscara de topología de CPU

HACERLO para cada CP lógico en la partición lógica objetivo.

SI el CP lógico está ENTONCES en línea

25 HACERLO

Índice de polaridad = índice de polaridad apropiado para el valor de polarización de CP lógico de acuerdo con el valor de polarización trasladado a matriz de índice de polaridad.

Copia local de máscara de topología de CPU (cpu_address, nodo, cptype, índice de polaridad) = ON

FIN

30 FIN

SI nuevo bloque de topología NO = bloque de topología actual para partición ENTONCES

Ajustar el bit de relación de cambio de topología y copiar la nueva topología a la actual.

Simulación de instrucción para STSI 15.1.2

35 Dentro de las correspondencias de syibk para el bloque de respuesta de STSI 15.1.2, un TLE de tipo contenedor, y un TLE de tipo CPU han sido añadidos. Esencialmente, los datos deben ser devueltos al contenedor o contenedores con la entrada al nivel más bajo siendo un TLE de tipo CPU. Se puede pensar de esto como una matriz de matrices basada en cómo han sido subdivididos o asignados los recursos de la partición lógica. Para la realización preferida, cada contenedor

es esencialmente un nodo con nivel 1 de anidado e incluye TLE de tipo CPU que tienen cada uno un nivel 0 de anidado. Los TLE de CPU están ordenados por tipo de CPU seguido por su clasificación. Las particiones verticales tienen cuatro clasificaciones (dedicada vertical, compartida vertical alta, compartida vertical media, compartida vertical baja) y las particiones horizontales tienen dos clasificaciones (dedicada y compartida).

- 5 Las siguientes operaciones ilustran un caso de uso para cómo es manejado un STSI 15.1.2 después de que todas las comprobaciones frontales han validado la entrada de instrucción.

Para la realización actual se ha asumido un máximo de 4 nodos y 64 procesadores.

- 10 Se está actualmente en iniciar la exploración de topbk, y en una variable local denominada `current_node_value` mantener el valor del índice de nodo. La razón que se necesita es debido a que si todas las máscaras de 64 bits dentro de un nodo son cero, no se necesita crear un TLE de tipo contenedor para ese nodo.

- 15 Una vez que es encontrada la primera entrada distinta de cero dentro de un nodo, se crea en primer lugar una entrada TLE de tipo contenedor para ese nodo. Dentro de la entrada de TLE contenedor, el valor de anidado es 1, seguido por 48 bits reservados. Los últimos bits son la ID del nodo que es el índice en topbk del nodo actual que se está tratando. Después de crear el TLE de tipo contenedor, se crea un TLE de tipo CPU para la entrada con una máscara de bit distinta de cero. Dentro de esta entrada, el nivel de anidado es 0, seguido por 24 bits reservados. Los siguientes 8 bits incluyen el bit dedicado y el bit de polarización. Si la partición es actualmente vertical, llénese el valor de polarización y el bit dedicado como sigue:

Clasificación 0 en topbk es dedicada vertical, almacenar un 11 en el PP y un 1 en D.

Clasificación 1 en topbk es compartida vertical alta, almacenar un 11 en el PP y un 0 en D.

- 20 Clasificación 2 en topbk es compartida vertical media, almacenar un 10 en el PP y un 0 en D.

Clasificación 3 en topbk es compartida vertical baja, almacenar un 01 en el PP y un 0 en D.

Para particiones horizontales, sólo la clasificación 0 y 1 son válidas actualmente. Llénese el bit dedicado y en el valor de polarización como sigue:

Clasificación 0 en topbk es dedicada horizontal, almacenar un 00 en el PP y un 1 en D.

- 25 Clasificación 1 en topbk es compartida horizontal, almacenar un 00 en el PP y un 0 en D.

El Tipo CPU, el siguiente valor que ha de ser llenado en el TLE-CPU es justo el índice de la segunda matriz dentro del topcpumask en topbk. (0-GP, 2-IFA, 3-IF, 4-ICF, 1 no es usado actualmente).

1. El siguiente valor es el origen de dirección de la CPU. Este valor es almacenado explícitamente como 0 cuando el 64 es el número máximo de CPU disponibles en la realización actual.

- 30 2. El último valor en syibk_vcm_cpu es la máscara de CPU, la máscara de 64 bits distintos de cero almacenados en la matriz anidada de matrices topcpumask.

3. Para cada máscara distinta de cero que sigue a la primera máscara de bit no-cero dentro de un nodo, crear una entrada de TLE de tipo CPU separada e iterar a través de este proceso para los 4 nodos.

- 35 La Relación de Cambio de Topología de Múltiples Procesadores Pendiente (MTCR) si el equipo de configuración de topología está instalado y habilitado (ECB.7 es uno), y la dirección de SCA para la CPU invitada es distinta de cero, la posición 0 de bit es usada para recordar el estado de relación de cambio de topología pendiente para la configuración. El estado repuesto de la relación de cambio de topología pendiente es cero.

- 40 El invento descrito aquí no está limitado a la topología de procesadores. Puede apreciarse que el componente básico del invento podría aplicarse ventajosamente a componentes distintos de CPU, incluyendo, pero no estando limitado a coprocesadores, caché, TLB, trayectos de datos internos, memorias tampón de trayecto de datos, memoria distribuida y adaptadores de comunicaciones de I/O por ejemplo.

REIVINDICACIONES

1. Para un sistema de ordenador anfitrión dividido en particiones lógicamente que comprende procesadores anfitriones, un método para descubrir una topología de uno o más procesadores invitados de una configuración invitada, caracterizado el método por comprender:

5 un procesador invitado de la búsqueda (2001) de configuración una instrucción de STORE SYSTEM INFORMATION (ALMACENAR INFORMACIÓN DEL SISTEMA) para su ejecución, la instrucción de STORE SYSTEM INFORMATION definida por una arquitectura de ordenador y especificando la instrucción de STORE SYSTEM INFORMATION una posición en la memoria de una tabla de topología de configuración;

ejecutar la instrucción de STORE SYSTEM INFORMATION que comprende:

10 basado en una solicitud de información de topología de la instrucción de STORE SYSTEM INFORMATION, obtener (2004) información de topología de la configuración invitada, comprendiendo la información de topología información acerca del agrupamiento lógico de procesadores de la configuración invitada a contenedores lógicos de acuerdo con su proximidad;

almacenar (2006) la información de topología en la tabla de topología de configuración;

15 en la que la tabla de topología de configuración incluye una entrada de procesador de lista de topología para un primer nivel de un agrupamiento jerárquico de procesadores que tienen atributos similares;

en el que la entrada del procesador de la lista de topología comprende además un indicador que indican cómo son dedicados los procesadores del grupo de procesadores a la configuración invitada de partición lógica; y

20 en el que la agrupación jerárquica consiste de uno o más niveles, y la tabla de topología de configuración incluye además una entrada de contenedor de lista de topología de cada nivel mayor que el primer nivel, siendo usada tal entrada de contenedor de lista de topología cuando los procesadores anfitriones del sistema están subdivididos en grupos peer.

25 2. El método según la reivindicación 1, en el que la instrucción de STORE SYSTEM INFORMATION comprende un campo de código de operación, un campo de registro base, un campo de desplazamiento con signo, en el que la instrucción de descubrimiento de topología comprende además un primer registro general implicado que contiene un campo de código de función y un campo de selector-1 y un segundo registro general implicado que contiene un campo selector-2, especificando el campo de función de código la solicitud de información de topología, identificando el campo de registro base y el campo de desplazamiento con signo una posición en memoria de un bloque de información del sistema que contiene la tabla de topología de configuración en la que valores del campo selector-1 y del campo selector-2, en combinación, determinan la solicitud de información de topología que ha de ser realizada.

3. El método según la reivindicación 1, en el que la instrucción de STORE SYSTEM INFORMATION definida por la arquitectura del ordenador es buscada y ejecutada por una unidad de tratamiento central de una arquitectura de ordenador alternativa,

35 en el que el método comprende además interpretar la instrucción de STORE SYSTEM INFORMATION para identificar una rutina de software predeterminada para emular la operación de la instrucción de STORE SYSTEM INFORMATION; y

en el que ejecutar la instrucción de STORE SYSTEM INFORMATION comprende ejecutar la rutina de software predeterminada para realizar operaciones del método para ejecutar la instrucción de STORE SYSTEM INFORMATION.

40 4. Un sistema que comprende medios destinados a llevar a cabo todas las operaciones del método de acuerdo con cualquier reivindicación de método precedente.

5. Un programa de ordenador que comprende instrucciones para llevar a la práctica todas las operaciones del método de acuerdo con cualquier reivindicación de método precedente, cuando dicho programa de ordenador es ejecutado en un sistema de ordenador.

45

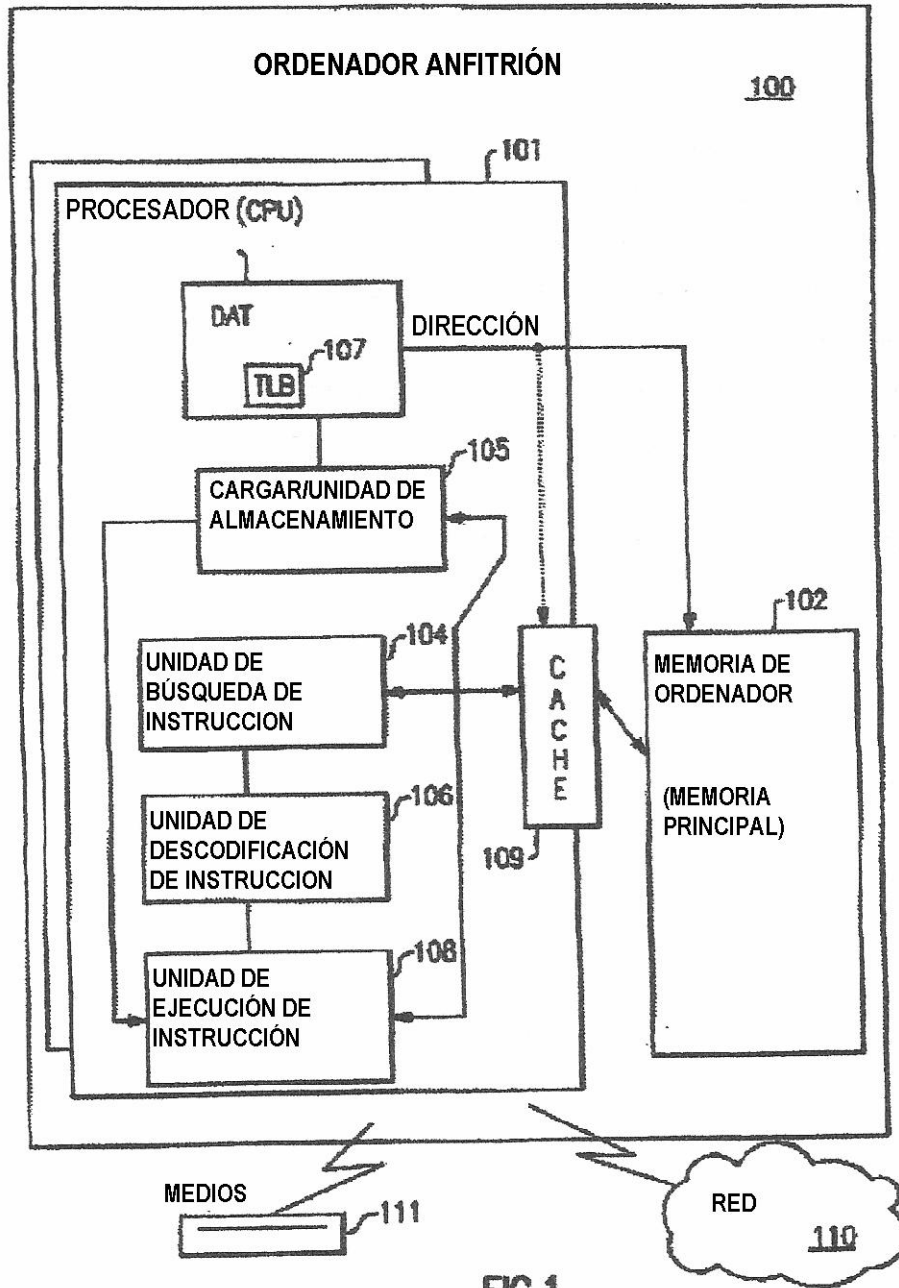


FIG.1
Técnica Anterior

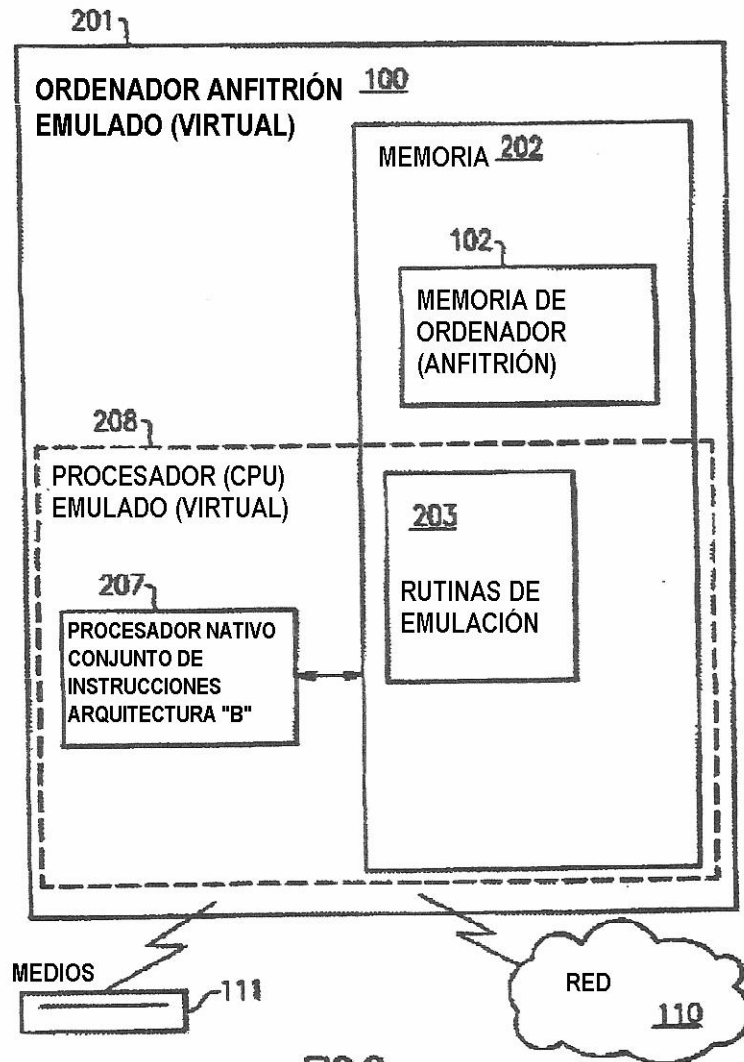


FIG.2
Técnica Anterior

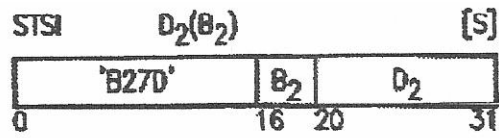
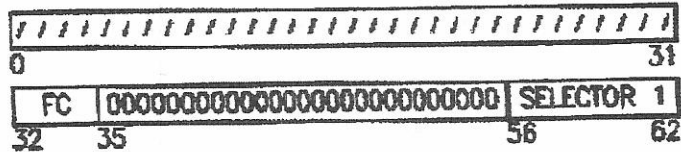


FIG.3

Técnica Anterior

REGISTRO GENERAL 0



REGISTRO GENERAL 1

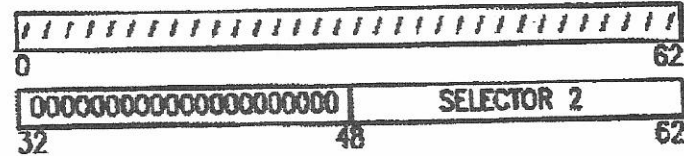


FIG.4

FIG.5

CÓDIGO DE FUNCIÓN	SEL 1	SEL 2	INFORMACIÓN SOLICITADA
0	—	—	NÚMERO DE NIVEL DE CONFIGURACIÓN ACTUAL
1	1	1	CONFIGURACIÓN DE MÁQUINA BÁSICA
1	2	1	CPU DE MÁQUINA BÁSICA
1	2	2	CPU DE MÁQUINA BÁSICA
2	2	1	CPU PARTICIÓN LÓGICA
2	2	2	CPU PARTICIÓN LÓGICA
3	2	2	CPU MÁQUINA VIRTUAL
15	1	2	INFORMACIÓN DE TOPOLOGÍA DE CONFIGURACIÓN ACTUAL

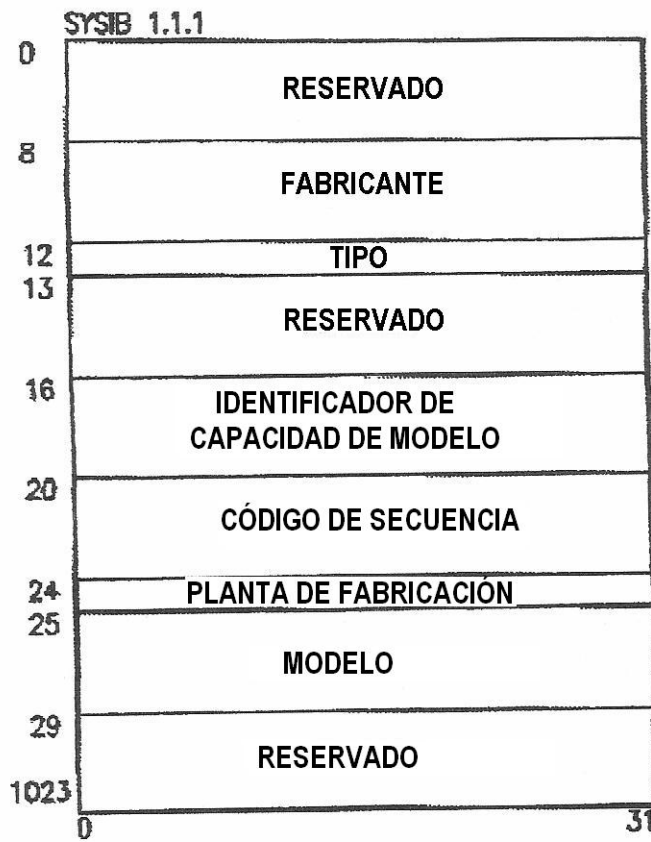


FIG.6
Técnica Anterior

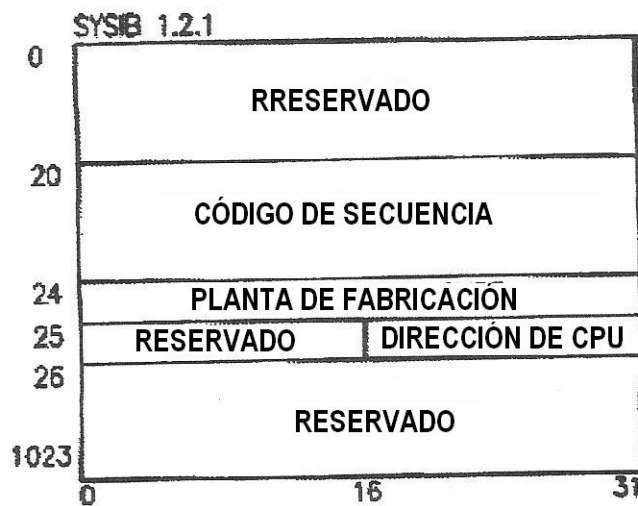


FIG.7
Técnica Anterior

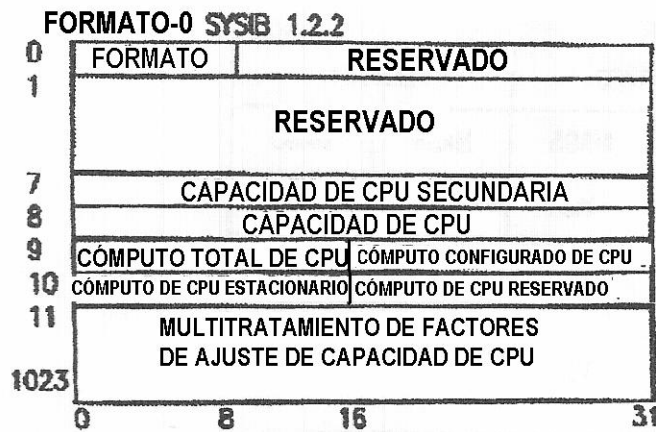


FIG.8

Técnica Anterior

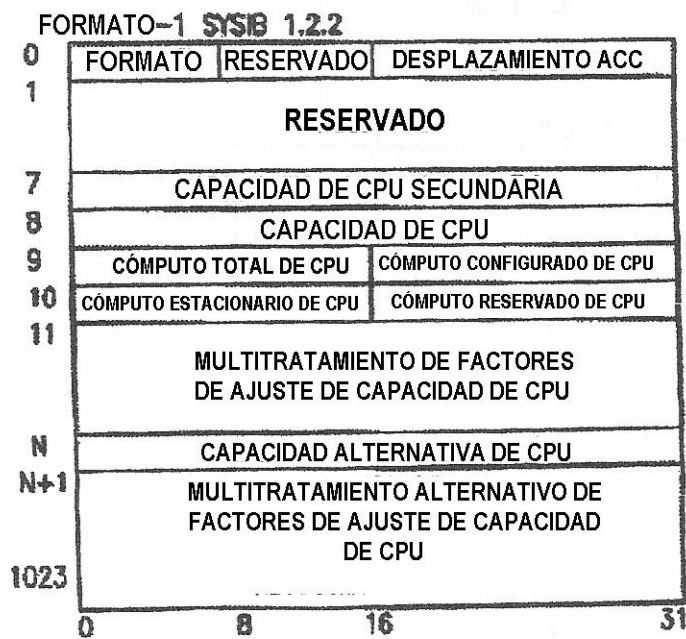


FIG.9

Técnica Anterior

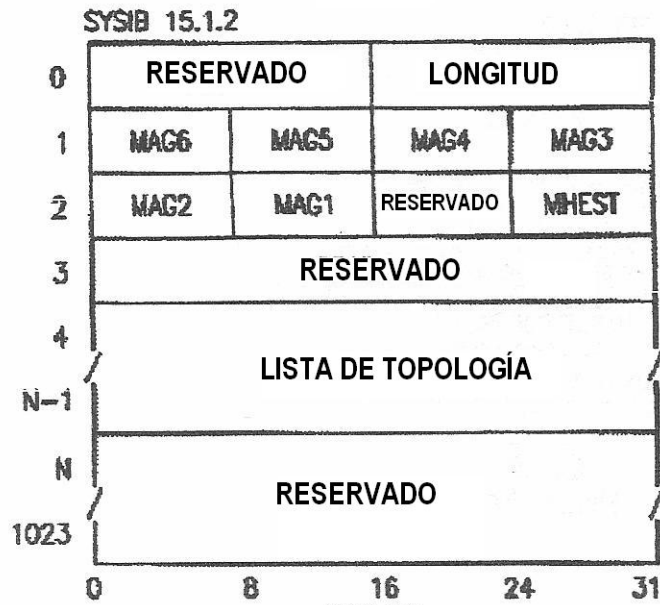


FIG.10

TLE TIPO CONTENEDOR

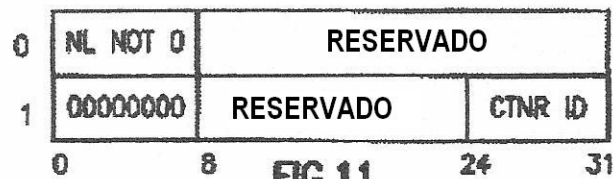


FIG.11

TLE TIPO CPU

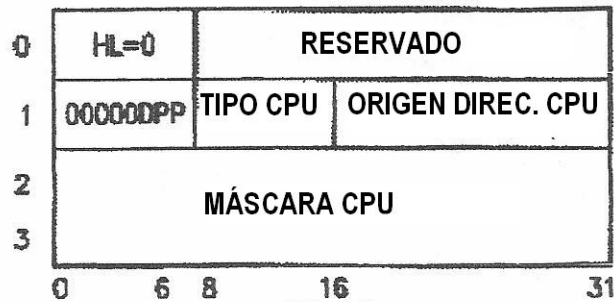
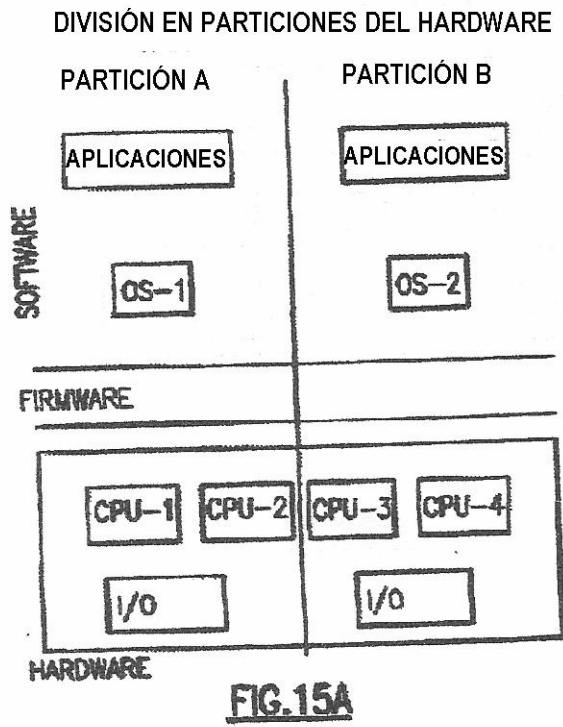
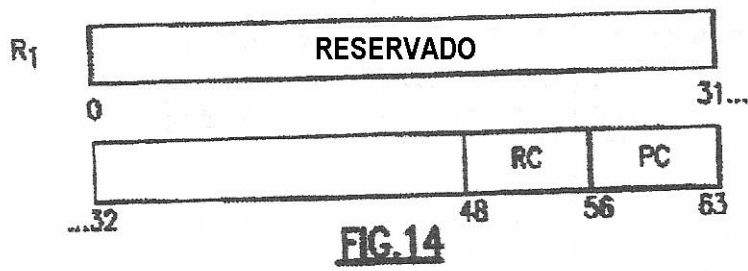
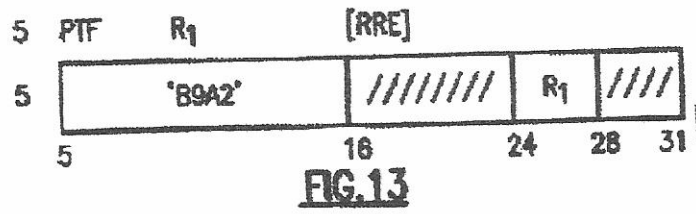


FIG.12



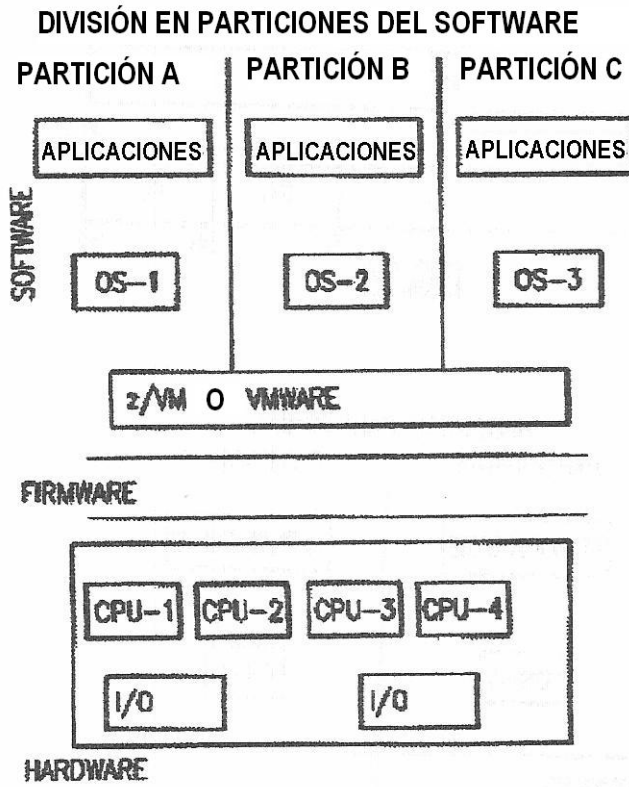


FIG.15B

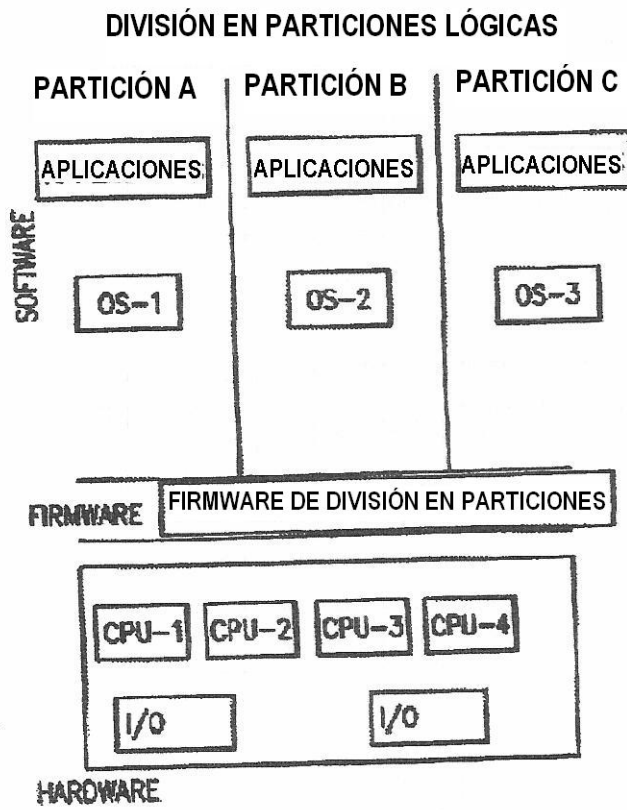


FIG.15C

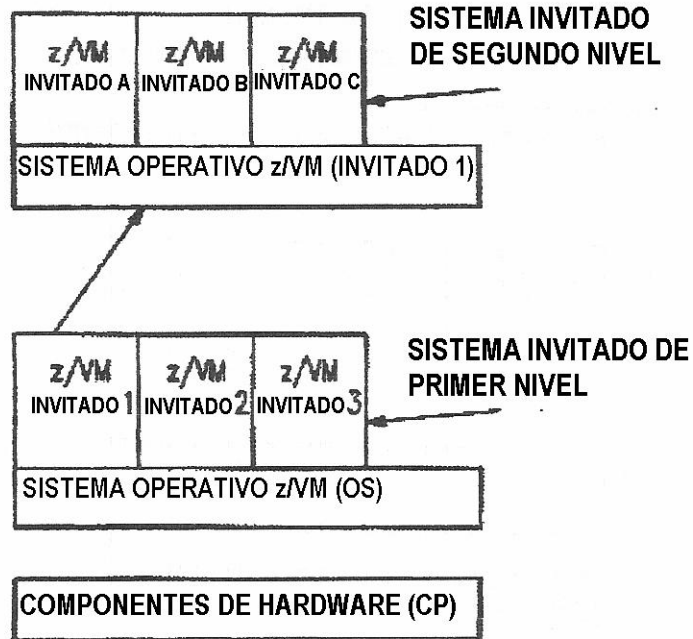


FIG.15D

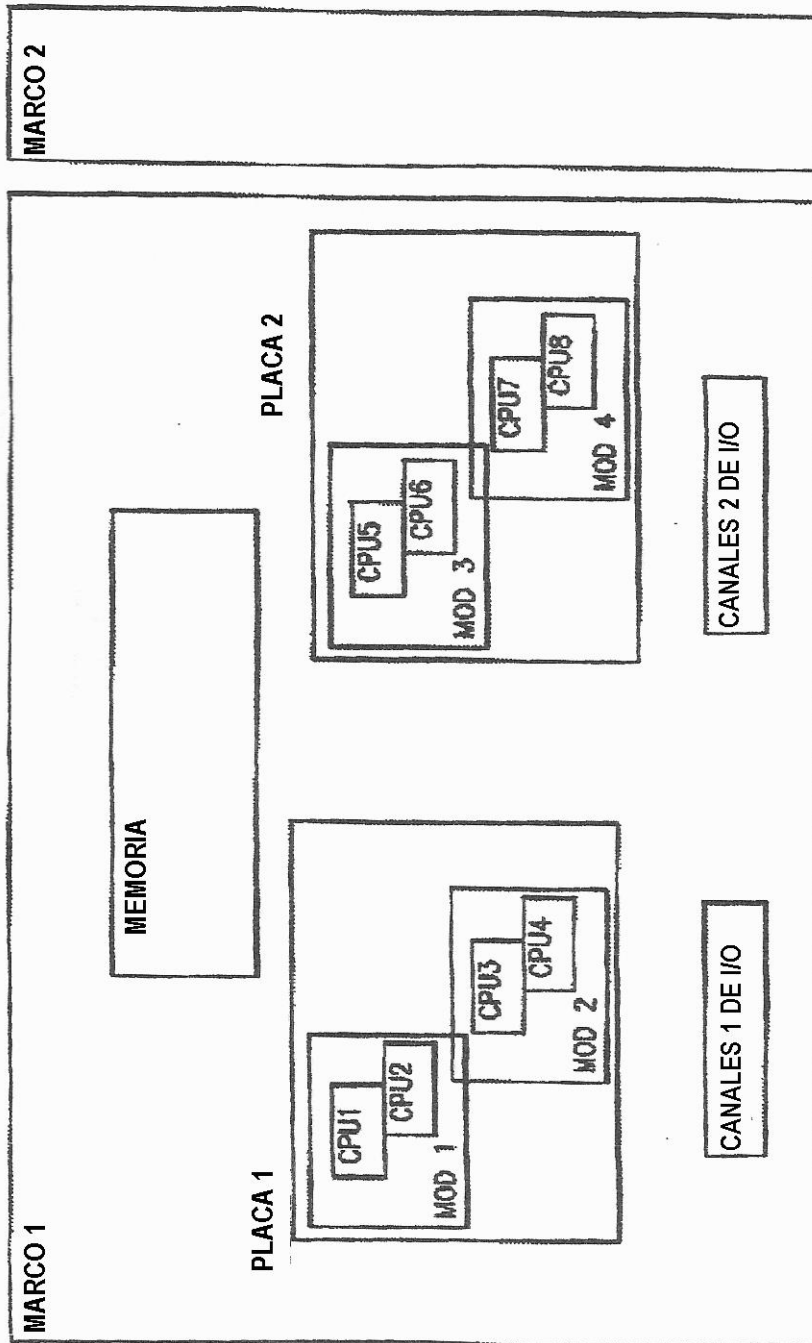


FIG.16
Técnica Anterior

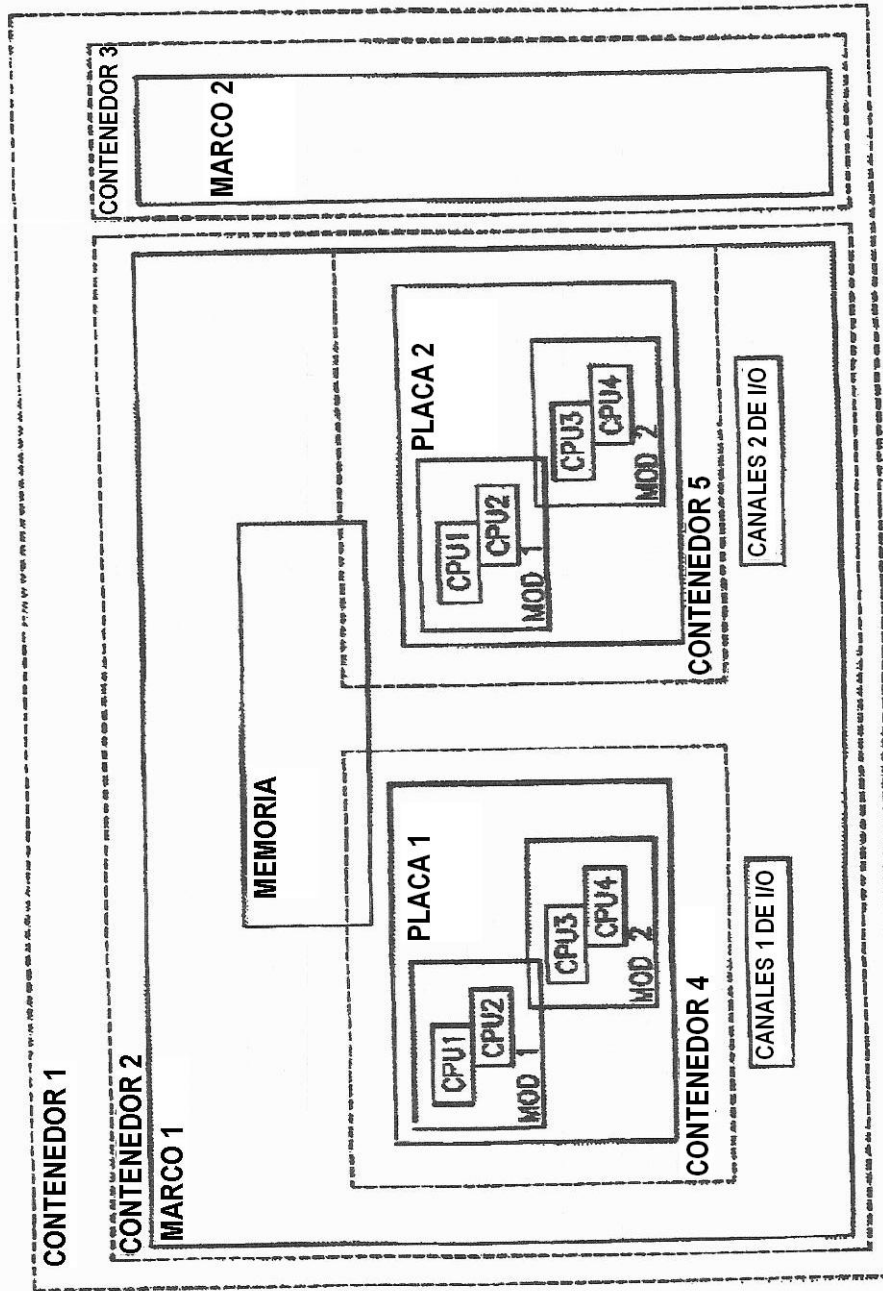


FIG.17

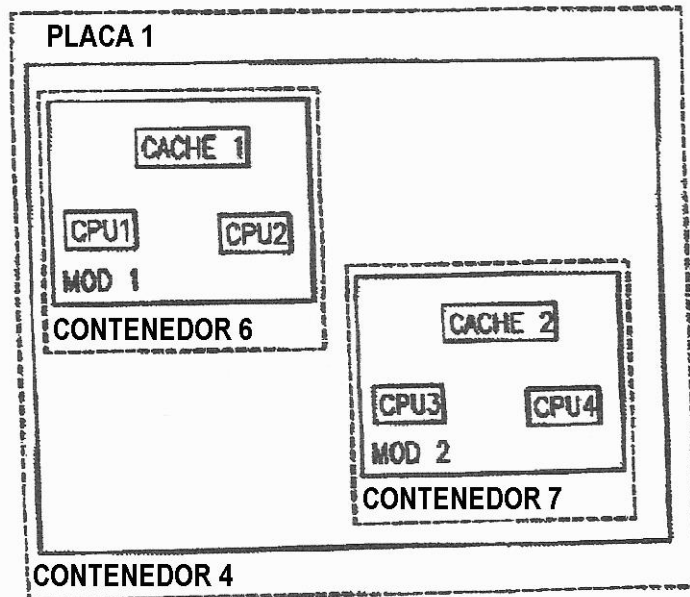


FIG.18

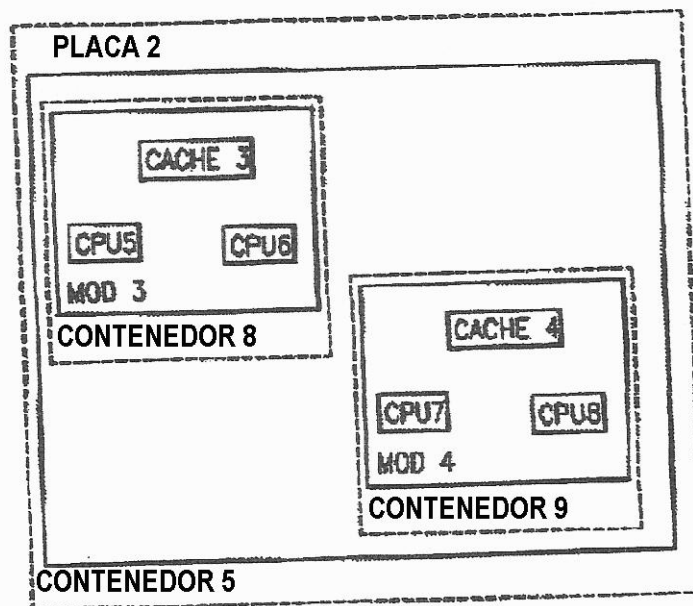


FIG.19

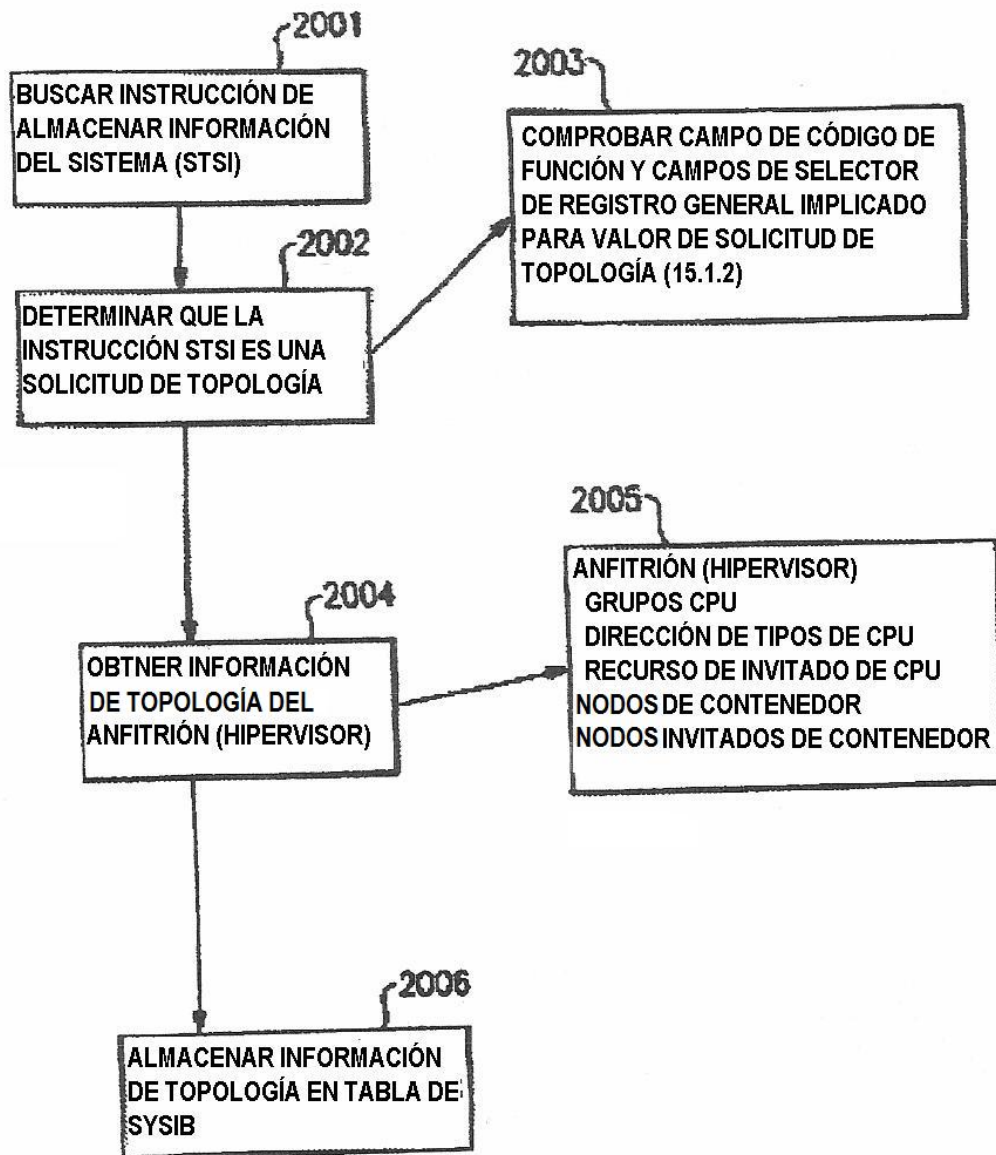


FIG. 20