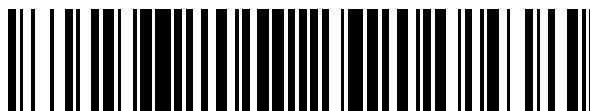


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 548 302**

51 Int. Cl.:

H04L 12/24

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **05.06.2003** **E 06026526 (1)**

97 Fecha y número de publicación de la concesión europea: **12.08.2015** **EP 1786142**

54 Título: **Método y sistema para simplificar la gestión de servidores distribuidos**

30 Prioridad:

12.06.2002 US 388112 P

10.03.2003 US 453308 P

16.04.2003 US 414959

16.04.2003 US 414958

16.04.2003 US 414887

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

15.10.2015

73 Titular/es:

BLADELOGIC, INC. (100.0%)

2101 CityWest Blvd.

Houston, TX 77042, US

72 Inventor/es:

KRAUS, THOMAS MARTIN;

MANWANI, VIJAY G.;

MUDDANA, SEKHAR;

SRINIVASA, BALAJI y

REDDY, RAVI

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 548 302 T3

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).

DESCRIPCIÓN

Método y sistema para simplificar la gestión de servidores distribuidos

Referencia cruzada a una aplicación relacionada

Esta solicitud reivindica la prioridad y el beneficio de la Solicitud de patente provisional de EE.UU. de N° de serie 60/388.112 presentada el 12 de junio de 2002, titulada METHOD AND SYSTEM FOR SYMPLIFYING SERVER MANAGEMENT; la Solicitud de patente provisional de US de N° de serie 60/453.308 presentada el 10 de marzo de 2003, titulada METHOD AND SYSTEM FOR SYMPLIFYING SERVER MANAGEMENT; Solicitud de patente provisional de EE.UU. de N° de serie 10/414.958 presentada el 16 de abril de 2003, titulada METHOD AND SYSTEM FOR EXECUTING AND UNDOING DISTRIBUTED SERVER MANAGEMENT; Solicitud de patente provisional de EE.UU. de N° de serie 10/414.959 presentada el 16 de abril de 2003, titulada METHOD AND SYSTEM FOR SIMPLYFYING DISTRIBUTED SERVER MANAGEMENT; y Solicitud de patente provisional de EE.UU. de N° de serie 10/414.887 presentada el 16 de abril de 2003, titulada METHOD AND SYSTEM FOR MODEL-BASED HETEROGENEOUS SERVER CONFIGURATION MANAGEMENT, cuyas descripciones completas se incorporan aquí como referencia.

Sector técnico

Esta invención se refiere al campo de la gestión de servidores y, más particularmente, a la gestión de servidores en un entorno informático heterogéneo.

Antecedentes

Los administradores de la Tecnología de la información (IT – Information Tecnology, en inglés) se enfrentan a nuevos retos debido a un importante incremento del número de servidores en la infraestructura de IT de una empresa y a la adopción de aplicaciones electrónicas distribuidas para empresas. Estos retos han resultado de: (1) una transición de arquitecturas de cliente – servidor a arquitecturas de Internet, lo que resulta en frecuentes interacciones entre diferentes tipos de servidores; y (2) el uso de servidores de aplicaciones para componentes, tales como J2EE (Plataforma Java 2, Edición para Empresas – Java 2 Platform, Enterprise Edition, en inglés) y .NET, para generar modelos de aplicación para componentes, para herramientas para sistema y complejos. Enfrentado a estos retos, un administrador de IT puede necesitar compaginar cientos de configuraciones de aplicaciones de software incompatibles y realizar un seguimiento a miles de componentes de servidor para los treinta o cuarenta servidores que gestiona.

Las herramientas de configuración actualmente disponibles resultan inadecuadas para la gestión de un gran número de componentes de configuración de aplicaciones de software y de servidor en múltiples servidores en un entorno informático heterogéneo. Para gestionar y configurar servidores heterogéneos, particularmente en la compleja infraestructura informática de una empresa, muchos administradores de IT utilizan productos de gestión de sistemas para empresas (ESM – Enterprise System Management, en inglés) que ofrecen herramientas de monitorización para automatizar la identificación de problemas en múltiples servidores. No obstante, estas herramientas de monitorización no proporcionan un sistema de gestión centralizado con una base de datos de configuración centralizada, que pueda realizar un seguimiento centralizado de los componentes de servidor actuales y de sus interdependencias en los diferentes servidores.

Además, estos productos de ESM proporcionan poca o ninguna ayuda para la corrección o la configuración de componentes de servidor en un entorno informático heterogéneo. Para servidores de sistema que operan en UNIX y Linux, a pesar de las herramientas de fuente abierta y las desarrolladas internamente y los programas para gestionar cambios de configuración simples a configuraciones de J2EE, ni las herramientas ni los programas pueden ser fácilmente extendidos para dirigirse a aplicaciones distribuidas complejas.

Los servidores de sistema operativo Microsoft Windows son incluso más difíciles de corregir y de configurar que los servidores de sistema operativo UNIX y Linux, debido a que un gran número de componentes de servidor tienen complejas interdependencias. Aunque existen herramientas de gestión de sistemas de Microsoft, han sido diseñadas solo para dirigirse a entornos informáticos de Windows homogéneos a pequeña escala, y no para los sistemas operativos múltiples que soportan entornos informáticos grandes y heterogéneos que los administradores de IT tienen que gestionar.

El documento EP 1091522A se refiere a un método para la configuración automática de un grupo de dispositivos, en los cuales se obtienen valores de configuración de otro dispositivo, y al menos algunos de los valores de configuración se aplican a otro dispositivo o grupo de dispositivos.

El documento US 6.349.306 se refiere a un método y aparato para la gestión de la configuración en redes de comunicación. El método y aparato implican monitorizar parámetros que gobiernan las características operacionales de un dispositivo de red, incluyendo el uso de plantillas para la generación de registros de configuración de los dispositivos de red de un tipo de modelo seleccionado.

El documento US 6.192.518 se refiere a un método para la distribución de software sobre enlaces de red mediante correo electrónico. El método implica distribuir mediante correo electrónico una aplicación de software residente en una red, de un ordenador de fuente acoplado a la red a un ordenador remoto.

El documento EP 0952521A se refiere a la gestión de un sistema informático. Más específicamente, los cambios de configuración son rastreados mediante monitorización histórica del estado de configuración de los dispositivos de una red informática, y los cambios de configuración rastreados se utilizan para la gestión y la resolución de problemas en el sistema informático.

5 El documento US 2002/0010910A se refiere a un sistema para la preparación y distribución de software. Más específicamente, el sistema comprende una preparación y sitio de pruebas que tiene un editor de paquetes de software, herramientas de transformación de paquetes de software y un módulo AutoPack para preparar y probar un software que va a ser distribuido a puntos finales.

10 El documento US2002/0069275 describe un método para configurar dispositivos de red. Para cada tipo de dispositivo y cada fabricante de dispositivos, esta librería de plantillas puede almacenar tanto los campos de atributos requeridos para la configuración de un dispositivo como el formato para la comunicación de esos campos de atributos.

Debido a las inadecuaciones en las herramientas de gestión actualmente disponibles, porciones importantes de algunas operaciones de cambio de configuración de servidor tienen que ser realizadas manualmente por el administrador de IT para cada servidor. De acuerdo con esto, pueden producirse errores humanos debido a estas operaciones manuales de cambio, y debido a la monitorización y al seguimiento manual de la configuración de cada servidor, lo que resulta en frecuentes configuraciones erróneas de los servidores y a caídas del sistema.

Compendio de la invención

20 Para aliviar esta situación, pueden utilizarse los sistemas y métodos de acuerdo con la invención para gestionar un gran número de servidores y sus componentes de servidor distribuidos en un entorno informático heterogéneo. El método, el sistema y el producto de programa informático de acuerdo con la presente invención se definen en esta memoria con referencia a las reivindicaciones 1 a 42.

En una realización, un usuario autenticado, tal como un administrador de sistema de IT, puede controlar y configurar de manera segura y simultáneamente múltiples servidores, que soportan diferentes sistemas operativos, implementando un servidor virtual del sistema de gestión del usuario. En una realización, el usuario es autenticado mediante un modelo de herencia de contexto de usuario de sistema operativo o protocolos de autenticación estándar, tales como un protocolo de clave pública, un protocolo Kerberos o un protocolo secreto compartido.

En algunas realizaciones, se utiliza un modelo de "servidor virtual". Un servidor virtual es un modelo abstracto que representa a una colección de servidores de objetivo reales. Para representar estos servidores físicos múltiples como un servidor virtual, se utilizan las llamadas abstractas de sistema que extienden la ejecución de llamadas de sistema específicas para un sistema operativo a múltiples servidores, independientemente de sus sistemas operativos soportados. Un servidor virtual es implementado por un cliente de servidor virtual y una colección de agentes de servidor virtual asociados con una colección de servidores reales. El cliente de servidor virtual puede ser implementado por una librería de código que detecta una red, tal como "libnc", que se implementa como una versión que detecta una red de la librería "libc". En otra realización, el cliente de servidor virtual es una librería, tal como "libnc".

El sistema de gestión de usuario contiene un sistema de aplicación de software, tal como un programa de control (denominado también una interfaz de línea de control) o un gestor de configuración, que genera llamadas abstractas de sistema para solicitar la realización de servicios en los servidores de objetivo. En una realización, el cliente de servidor virtual recibe las llamadas abstractas de sistema e instancia las llamadas abstractas de sistema en un modo de proceso seguro. La instanciación de proceso seguro asegura la ejecución simultánea de las llamadas de sistema en múltiples de servidores de objetivo, mientras comparte el cliente de servidor virtual único entre estos servidores de objetivo múltiples y sus agentes de servidor virtual asociados. En el proceso de instanciación, el cliente de servidor virtual identifica al servidor o a los servidores de objetivo y a su agente o agentes de servidor virtual asociado o asociados para recibir las llamadas abstractas de sistema. En una realización, el cliente de servidor virtual identifica al servidor o a los servidores de objetivo en respuesta a un identificador de servidor incluido en la llamada abstracta de sistema. Ejemplos del identificador de servidor incluyen un nombre de anfitrión especificado en una ruta y una dirección de red. El identificador de servidor puede obtenerse también a partir de un grupo de servidores a los cuales pertenece el servidor de objetivo.

50 Asimismo, en el proceso de instanciación, el cliente de servidor virtual transmite las llamadas abstractas de sistema al agente de servidor virtual identificado para su ejecución en el servidor de objetivo. Antes de la transmisión de la llamada abstracta de sistema, el cliente de servidor virtual puede codificar las llamadas abstractas de sistema utilizando protocolos de codificación estándar, tales como el protocolo SSL, el protocolo Kerberos o el protocolo de secreto compartido, para hacer segura la comunicación entre el cliente de servidor virtual y el agente de servidor virtual. Además, antes de la transmisión de la llamada abstracta de sistema, el cliente de servidor virtual puede especificar prioridad, uso de CPU y/o uso de memoria de la llamada abstracta de sistema en el servidor de objetivo identificado.

Después de que el agente de servidor virtual recibe las llamadas abstractas de sistema desde el cliente de servidor virtual, el agente de servidor virtual traduce la llamada abstracta de sistema en una llamada de sistema específica

para el sistema operativo, de manera que la llamada de sistema pueda ser ejecutada en el servidor de objetivo específico para el sistema operativo. Antes de traducir la llamada abstracta de sistema, en una realización, el agente de servidor virtual identifica al anfitrión fuente del sistema de gestión del usuario para determinar el protocolo de codificación utilizado en la llamada abstracta de sistema. El agente de servidor virtual descodifica la llamada abstracta de sistema tras averiguar el protocolo de codificación utilizado por el cliente de servidor virtual. A partir de la llamada abstracta de sistema descodificada, el agente de servidor virtual identifica al usuario autenticado. Además, el agente de servidor virtual contiene módulos de software que pueden mapear al usuario autenticado (usuario presentado) a otro usuario (usuario efectivo) y localizar una identidad de usuario local correspondiente en el servidor de objetivo para el usuario efectivo, y suplantar al usuario efectivo como un usuario local en el servidor de objetivo asociado con el agente de servidor virtual. En una realización, si el usuario efectivo no se ha identificado como un usuario local reconocido en el servidor de objetivo, el usuario se designa como un usuario invitado local en el servidor de objetivo. El agente de servidor virtual restringe además el acceso del usuario al servidor de objetivo mediante un módulo de software que limita al usuario a realizar acciones predeterminadas o a acceder a recursos predeterminados en el servidor de objetivo, sobre la base de un modelo de control de acceso basado en función y/o en las listas de control de acceso (ACL – Access Control Lists, en inglés).

Las llamadas de sistema traducidas son a continuación ejecutadas en el servidor de objetivo en un modo de proceso seguro, y los resultados de la ejecución son transportados desde el agente de servidor virtual al cliente de servidor virtual. En una realización, el agente de servidor virtual contiene un informe de auditoría para registrar los nombres de los usuarios y las llamadas abstractas de sistema ejecutadas en el servidor de objetivo.

En otra realización, el sistema de aplicación puede agregar múltiples llamadas abstractas de sistema en una única llamada abstracta de sistema de alto nivel, que a su vez es transportada al cliente de servidor virtual. Tras recibir la llamada abstracta de sistema de alto nivel, el cliente de servidor virtual desintegra la llamada abstracta de sistema de alto nivel en las múltiples llamadas abstractas de sistema originales e instancia estas llamadas abstractas de sistema originales individualmente. De acuerdo con esto, el agente de servidor virtual recibe las llamadas abstractas de sistema individuales para su ejecución en el servidor de objetivo asociado.

En otra realización más, tras la recepción de la llamada abstracta de sistema de alto nivel del programa de aplicación, el cliente de servidor virtual instancia la llamada abstracta de sistema de alto nivel como un todo. Así, el agente de servidor virtual identificado recibe la llamada abstracta de sistema de alto nivel, en lugar de las múltiples llamadas abstractas de sistema originales. El agente de servidor virtual a su vez traduce la llamada abstracta de sistema de alto nivel en las llamadas de sistema específicas para el sistema operativo individuales para ser ejecutadas en su servidor de objetivo asociado.

En otra realización, el servidor virtual modifica una aplicación no distribuida existente que soporta solo un sistema operativo específico para funcionar como una aplicación que detecta una red que es aplicable en servidores o dispositivos que soportan diferentes sistemas operativos sustituyendo una llamada de sistema que no detecta red con una llamada abstracta de sistema. En una realización de ejemplo, un programa de intérprete de órdenes de Unix no distribuido puede funcionar como un programa de aplicación que detecta red que es adaptable a múltiples servidores o dispositivos que soportan sistemas operativos distintos del Unix. En otra realización de ejemplo, lenguajes de programación no distribuida, tales como Perl y Python, pueden funcionar como programas de aplicación que detectan red que son aplicables en múltiples servidores y dispositivos que soportan diferentes sistemas operativos.

En otra realización, los componentes de configuración de software (denominados también objetos de servidor) que tienen interdependencias intrincadas con otros componentes de servidor pueden definirse y caracterizarse bajo un solo sistema unificado. Mediante este sistema unificado, las operaciones de cambio en aplicaciones de grano fino pueden ser implementadas uniforme y simultáneamente en servidores heterogéneos, en lugar de implementar diferentes operaciones de cambio de aplicación para cada uno de los servidores de manera individual.

En otra realización más, un sistema de gestión centralizado puede rastrear cambios, configurar y gestionar múltiples servidores de manera automática para cumplir con las políticas predefinidas mediante la incorporación de los métodos y sistemas descritos anteriormente.

Esta invención se refiere también a un método y sistema para ejecutar y deshacer operaciones de cambio en servidores distribuidos para una colección de objetos de servidor en múltiples servidores de objetivo en un modo de transacción segura. En esta memoria, transacción segura significa que todas las etapas requeridas de cada operación de cambio en servidor distribuido se completan antes de que la operación de cambio en servidor distribuido se consideren completadas y, si se produce un error cuando se llevan a cabo las etapas requeridas en los servidores de objetivo, cualquier cambio realizado en estas etapas se deshace.

Ejemplos de operaciones de cambio en servidores distribuidos para una colección de objetos de servidor puede ser instalación, copia, actualización o borrado de objetos de servidor. En una realización de ejemplo, una colección de objetos de servidor puede ser copiada desde una sola fuente a múltiples servidores de objetivo remotos. Asimismo, todos los cambios provocados por la copia de esta colección de objetos de servidor pueden ser revertidos en los múltiples servidores de objetivo remotos afectados.

En una realización, las operaciones de cambio en servidor para una colección de objetos de servidor, tal como archivos y entradas de archivos de configuración, se especifican en un paquete de transacción. En particular, las operaciones de cambio en servidor se especifican en un paquete de transacción para cambiar código y contenido (archivos, aplicaciones, componentes compuestos, etc.), configurar parámetros de múltiples servidores simultáneamente y deshacer los cambios en caso de fallo. Las operaciones de cambio en servidor en el paquete de transacción pueden estar especificadas para que se apliquen sobre objetos de servidor primitivo, objetos de servidor compuesto, objetos de servidor de configuración abstracta y objetos de servidor de componentes. Un objeto de servidor primitivo es un objeto de servidor elemental que sirve como base para todos los demás tipos de objetos de servidor. Un objeto de servidor compuesto es un objeto de servidor que contiene objetos de servidor primitivo y otros objetos de servidor compuesto relacionados. Un objeto de servidor de configuración abstracta es un tipo especial de un objeto de servidor primitivo que representa una entrada en un archivo de configuración cuando el archivo de configuración es mapeado a un formato de archivo de configuración abstracta común utilizando una gramática de configuración específica para un archivo. Un objeto de servidor de componentes es una colección secuenciada de objetos de servidor que contiene información de prerequisites y de herencia acerca de otros tipos de objetos de servidor.

En una realización, las operaciones de cambio en servidor en un paquete de transacción se especifican en un conjunto de instrucciones en formato XML. En otra realización, las operaciones de cambio en servidor se especifican en un conjunto de instrucciones de texto.

En una realización, el paquete de transacción incluye un contexto de transacción, un archivo de parámetros, acciones para manejo de errores, una instrucción de secuenciación para las operaciones de cambio y prerequisites de servidor de objetivo para la ejecución de las operaciones de cambio, además de las operaciones de cambio especificadas. El contexto de transacción es identificado mediante instrucciones de iniciar transacción y finalizar transacción que encapsulan las operaciones de cambio en el objeto de servidor. El archivo de parámetros especifica valores de parámetros para cada uno de los servidores de objetivo identificados. Estos valores de parámetros son comunicados a los servidores de objetivo identificados junto con el paquete de transacción. En una realización, el archivo de parámetros contiene parámetros que referencian valores de parámetros que son idénticos en los servidores de objetivo. En otra realización, el archivo de parámetros contiene parámetros que referencian valores de parámetros que son diferentes para cada uno de los servidores de objetivo. El paquete de transacción soporta varios tipos de errores, tales como errores leves y errores graves, en sus acciones de manejo de errores. La instrucción de secuenciación proporciona una secuencia de ejecución para las operaciones de cambio especificadas. Si esta instrucción no es proporcionada localmente dentro del paquete de transacción, se accede a un diagrama de dependencia externa para proporcionar una secuencia de ejecución para las operaciones de cambio especificadas. El paquete de transacción proporciona asimismo la información de prerequisites para que los servidores de objetivo ejecuten las operaciones de cambio especificadas.

En una realización, el usuario puede, opcionalmente, elegir continuar con un simulacro. El simulacro proporciona un conjunto adicional de pruebas para ver si las operaciones de cambio en el objeto de servidor pueden ser llevadas a cabo por servidores de objetivo receptores antes de hacer ningún cambio.

Después de que se ha comunicado el paquete de transacción a los servidores de objetivo, las operaciones de cambio especificadas son ejecutadas en cada uno de los servidores de objetivo identificados en un modo de transacción segura utilizando los valores del parámetro.

En una realización, las operaciones de cambio especificadas pueden ser revertidas cuando un usuario realiza una solicitud explícita o cuando se detecta un error en un informe de transacción guardado para el paquete de transacción, tras una ejecución parcial o completa de las operaciones de cambio. El informe de transacción hace un seguimiento de los detalles de todas las etapas ejecutadas, de manera que cada etapa ejecutada de una operación de cambio puede ser deshecha y revertida de los servidores de objetivo afectados.

En otra realización, pueden ensamblarse múltiples paquetes de transacción en un proyecto de transacción. Todas las operaciones de cambio especificadas en un proyecto de transacción pueden ser ejecutadas en un modo de transacción segura.

Esta invención se refiere también a un método y sistema para configurar servidores heterogéneos en una red, proporcionando módulos que pueden examinar, tomar instantáneas, rastrear cambios, rastrear cumplimiento, restaurar configuración previa, realizar actualizaciones en cada uno de los servidores y proveer nuevos servidores y aplicaciones.

Un objeto de servidor es uno o una colección de parámetros de configuración correspondientes y de equipos de servidor, tales como archivos, directorios, registros, parches, paquetes, servicios y aplicaciones. En una realización, existen cuatro tipos de objetos de servidor: un objeto de servidor primitivo, un objeto de servidor compuesto, un objeto de servidor de configuración abstracta y un objeto de servidor de componentes. Un objeto de servidor primitivo es un objeto de servidor elemental que sirve como base para todos los demás tipos de objetos de servidor. Un objeto de servidor compuesto es un objeto de servidor que contiene objetos de servidor primitivo y otros objetos de servidor compuesto afines. Un objeto de servidor de configuración abstracta es un tipo especial de un objeto de

servidor primitivo que representa una entrada en un archivo de configuración cuando el archivo de configuración está mapeado a un formato de archivo de configuración abstracta común que utiliza una gramática de configuración específica para un archivo. Un objeto de servidor de componentes es una colección secuenciada de objetos de servidor que contiene información de prerrequisitos y de herencia acerca de otros tipos de objetos de servidor.

- 5 En una realización, los objetos de servidor en múltiples servidores pueden examinarse en tiempo real. Durante el examen, puede seleccionarse una colección de identificadores de objeto de servidor (sin valores) y pueden reunirse manualmente en una plantilla. En otra realización, la plantilla puede incluir una o más plantillas definidas previamente.

- 10 Los valores de los objetos de servidor identificados en la plantilla pueden ser registrados para un servidor específico (denominado también “servidor gold”) mediante un proceso denominado “instantánea”, que recoge los valores (denominados también “resultados instantáneos”) y los guarda en un modelo de referencia. En una realización, el modelo de referencia puede ser un modelo de referencia importado creado por un proveedor externo. En otra realización, el modelo de referencia puede incluir uno o más modelos de referencia previamente definidos. En una realización, el modelo de referencia puede ser utilizado para obtener a partir del mismo reglas de cumplimiento tales como valores de configuración de línea básica y rangos de cumplimiento, de los valores recogidos para otros servidores de la red. Comparando otros servidores en directo con el modelo de referencia, los sistemas y métodos llevan un registro del cumplimiento y los cambios en la configuración de los otros servidores de la red.

- 20 De manera alternativa, en lugar de guardar los resultados instantáneos en el modelo de referencia, un resultado instantáneo puede utilizarse para capturar la configuración de un servidor en un momento arbitrario del tiempo. La configuración puede incluir objetos de servidor que son seleccionados explícitamente de un servidor o que son seleccionados implícitamente por medio de la plantilla. En otra realización, los resultados instantáneos pueden recurrir también a instantáneas de un servidor tomadas a intervalos de tiempo programados. En esta realización, la primera instantánea sirve como línea básica para sucesivas instantáneas, de manera que para las siguientes instantáneas solo se capturen los cambios con respecto a la línea básica. Así, cualquier instantánea puede ser reconstruida para mostrar la configuración completa de un servidor en un momento específico de los intervalos de tiempo combinando la línea básica con los cambios incrementales guardados para los resultados instantáneos particulares. Además, estas instantáneas tomadas a lo largo de un periodo de tiempo pueden ser utilizadas por el usuario para analizar cambios en el servidor a lo largo del tiempo. Además, puede utilizarse una sola instantánea o instantáneas recurrentes para llevar un registro de un cambio en un momento arbitrario del tiempo o a lo largo de un periodo de tiempo programado.

- 30 Tras comparar los servidores en la red e identificar las discrepancias existentes en los servidores comparados se generan operaciones de cambio para corregir estas discrepancias. En una realización, estas operaciones de cambio en servidor pueden presentarse a los servidores como un paquete de transacción de manera que las operaciones de cambio puedan ser ejecutadas a través de múltiples servidores en un modo de transacción segura para sincronizar los servidores de objetivo con el modelo de referencia o con las instantáneas. De manera similar, para actualizar los servidores de objetivo, el usuario actualiza el modelo de referencia y empaqueta las actualizaciones en paquetes de una transacción para sincronizar los servidores de objetivo con el modelo de referencia.

En una realización, el modelo de referencia puede ser utilizado para proveer un nuevo servidor recién añadido a la red para asegurar la coherencia en la configuración de los servidores en la red.

- 40 En otra realización, los servidores de la red pueden restaurar su configuración previa a partir del modelo de referencia o de las instantáneas, de manera que en caso de fallo de servidor el servidor pueda ser restaurado para recuperar su configuración y contenidos existentes.

- 45 En otra realización más, un servidor en directo puede compararse con otro servidor en directo comparando los objetos de servidor identificados en la plantilla. En otra realización, el usuario puede seleccionar explícitamente los objetos de servidor que se comparten normalmente entre estos servidores en directo y compararlos de manera correspondiente.

- 50 En una realización, los sistemas y métodos de acuerdo con la invención gestionan parámetros de configuración relacionados en categorías a través de diferentes servidores modelizando los parámetros en plantillas. Los objetos de servidor (denominados también parámetros de configuración) son clasificados en categorías en una plantilla de categorías de tipo por servidor, tal como una categoría de servidor de aplicación, una categoría de servidor de base de datos y una categoría de servidor web. En la misma plantilla, los objetos de servidor son entonces clasificados en categorías mediante categorías de tipo de parámetros de configuración (por ejemplo, parámetros de red, parámetros de capacidad, parámetros de disponibilidad, parámetros de funcionamiento y parámetros de seguridad), subcategorías y claves asociadas sobre basadas en su función. Puede obtenerse una nueva plantilla a partir de la primera plantilla que combina los objetos de servidor relacionados en categorías en las categorías de servidor, que gestiona los parámetros de configuración como si perteneciesen a un solo servidor. Por ejemplo, los parámetros de configuración de un servidor web individual relativos a la seguridad pueden cambiar de acuerdo con los parámetros de seguridad de un servidor de aplicación y un servidor de base de datos.

Breve descripción de los dibujos

En los dibujos, caracteres de referencia iguales se refieren en general a las mismas partes en las diferentes vistas. Asimismo, los dibujos no están necesariamente a escala, poniéndose en general el énfasis, por el contrario, en ilustrar los principios de la invención.

- 5 La FIG. 1 es un diagrama de bloques que representa una realización de un sistema para gestionar múltiples servidores en un entorno informático heterogéneo.

La FIG. 2 es un diagrama de bloques que representa un cliente de servidor virtual de acuerdo con una realización de la invención.
- 10 La FIG. 3 es un diagrama de bloques que representa un agente de servidor virtual según una realización de la invención.

La FIG. 4 es un diagrama de flujo que representa una realización de un método para recibir y ejecutar una llamada de sistema desde un programa de aplicación.

La FIG. 5 es un diagrama de flujo que representa los detalles de la instanciación de una llamada abstracta de sistema en una realización del método de la FIG. 4.
- 15 La FIG. 6 es un pantallazo de una realización de un sistema que implementa el método de la FIG. 4.

La FIG. 7 es un diagrama de bloques que representa una realización de un sistema para ejecutar y deshacer operaciones de cambio en un servidor distribuido en un modo de transacción segura.

La FIG. 8 es un diagrama de flujo que representa una realización de un sistema para ejecutar y deshacer operaciones de cambio en un servidor distribuido en un modo de transacción segura.
- 20 La FIG. 9 es un diagrama de flujo que representa una realización de un método para ejecutar y deshacer operaciones de cambio en un servidor distribuido en un modo de transacción segura.

La FIG. 10 es un diagrama de bloques que representa una realización de un sistema para configurar múltiples servidores en un entorno informático heterogéneo.
- 25 La FIG. 11 es un diagrama de flujo que representa una realización de un método para configurar múltiples servidores en un entorno informático heterogéneo.

La FIG. 12 es un diagrama de bloques que representa una realización de un sistema para gestionar objetos de servidor tal como se describe en una realización de la invención.

La FIG. 13 es un diagrama de bloques que representa una realización de ejemplo del sistema de la FIG. 12.
- 30 La FIG. 14 es una pantalla de interfaz de usuario en una realización para un sistema que implementa el método de la FIG. 11.

Descripción detallada

- 35 En referencia a la FIG. 1, un usuario 10, tal como un administrador de sistema, gestiona varios servidores 15A, 15B, 15C, 15D, en general 15, que son ordenadores, cada uno de los cuales puede ser del mismo tipo o de un tipo diferente de los otros servidores 15. Los servidores 15 son típicamente ordenadores de propósito general de clase servidor, que proporcionan servicios (por ejemplo aplicaciones de software y/o datos) a otros ordenadores a través de una o más redes informáticas. Por ejemplo, los servidores pueden ser servidores de aplicación, encaminadores, cortafuegos, equilibradores de carga, controladores de almacenamiento o una combinación de estos o de otros ordenadores o dispositivos de red.
- 40 Ejemplos de servidores de aplicación son bases de datos, tales como la base de datos Oracle de Oracle Corporation de Redwood City, California, u otras aplicaciones de empresa. Los servidores de aplicación pueden incluir también servidores web, tales como el servidor web Apache de la Fundación Apache, y el Servidor de Información de Internet (IIS – Internet Information Server, en inglés) de Microsoft Corporation de Redmond, WA. Además de estos ejemplos, los servidores 15 pueden proporcionar otros programas. Resultará evidente que, tal como se utiliza en esta memoria, el término “servidor” no está limitado a ordenadores de clase servidor o a servidores de aplicación, sino que se refiere en general a ordenadores en los cuales operan las realizaciones de la invención, que pueden incluir otros tipos de ordenadores o de dispositivos de red.
- 45 Como se muestra, cada uno de los servidores 15 puede utilizar un sistema operativo diferente. Por ejemplo, el agente 15A utiliza MICROSOFT WINDOWS (por ejemplo, WINDOWS NT y WINDOWS 2000), de Microsoft Corporation de Redmond, WA; el servidor 15B utiliza SUN SOLARIS, de Sun Microsystems, Inc. de Santa Clara, CA; el servidor 15C utiliza RED HAT LINUX, de Red Hat, Inc. de Durham, N.C.; y el servidor 15D utiliza IBM AIX, de IBM
- 50

de Armonk, NY. Resultará evidente que este es solo un ejemplo de los sistemas operativos que pueden utilizarse en los servidores 15, y que pueden utilizarse otras combinaciones y sistemas operativos en los servidores 15 de acuerdo con las realizaciones de la invención. Uno de los beneficios del sistema es la capacidad de operar en un entorno que tiene servidores heterogéneos.

- 5 En una realización, el usuario 10 gestiona los servidores 15 por medio de un sistema de gestión 20. El sistema de gestión 20 es típicamente un ordenador de clase servidor que proporciona al usuario 10 la capacidad de gestionar servidores 15 de una manera coherente mediante el uso de programas de aplicación 25. El sistema de gestión 20 puede ser uno de los servidores 15 o cualquier ordenador de clase servidor que pueda comunicarse con los servidores 15 en una red. Cualquiera de los servidores 15 objetivo puede ser designado como sistema de gestión, siempre que el servidor designado incluya programas de aplicación y módulos de software apropiados para gestionar servidores situados remotamente.

- 10 Los programas de aplicación 25 en el sistema de gestión 20 pueden incluir uno o más de un programa 25A de intérprete de órdenes de línea de órdenes y programas afines para la ejecución de órdenes del intérprete de órdenes (por ejemplo, las órdenes del intérprete de órdenes UNIX tales como ls, mv, rm, etc.), un gestor de configuración 25B para gestionar la configuración del sistema y/u otras aplicaciones 25C. Los programas de aplicación 25, que en algunas implementaciones “detectan la red”, comunican las llamadas abstractas de sistema a un cliente de servidor virtual 30, que a su vez comunica las llamadas abstractas de sistema a los servidores 15 que son el objetivo o los objetivos para la ejecución de las operaciones solicitadas mediante las llamadas abstractas de sistema. Ventajosamente, mediante el uso de las llamadas abstractas de sistema, las operaciones “que detectan red” son capaces de solicitar servicios de servidores heterogéneos que soportan diferentes sistemas operativos sin tener que modificar su arquitectura para soportar cada uno de los diferentes sistemas operativos.

- 20 Por ejemplo, el usuario 10 introduce órdenes, tales como órdenes del intérprete de órdenes Unix, en el programa de intérprete de órdenes 25A a través de una interfaz de línea de órdenes. Las órdenes pueden ser introducidas, por ejemplo, para distribuir archivos, directorios, paquetes de software y parches a los servidores de objetivo 15. Pueden introducirse también órdenes para editar archivos de configuración de los servidores de objetivo 15. Además, pueden introducirse órdenes para reinicializar remotamente los servidores de objetivo 15, y detener e iniciar una operación de cambio en los servidores de objetivo 15.

- 25 Por ejemplo, en una implementación, la orden del intérprete de órdenes Unix “ls”, que solicita que un ordenador servidor enumere un directorio de archivos, puede modificarse para ser utilizada con el sistema de gestión de usuario 20 y el cliente de servidor virtual 30 para enumerar un directorio de archivos de cualquiera de los servidores de objetivo 15. Desde la perspectiva del usuario 10, la orden “ls” se utiliza de una manera normal, excepto por que el usuario 10 puede identificar un servidor de objetivo 15 para la orden en una ruta asociada con la orden. Por ejemplo, si el servidor de objetivo 15A se denomina “targetserver1”, el usuario 10 puede introducir la orden “ls//targetserver1/path/” para enumerar los archivos en la ruta especificada en el servidor de objetivo 15A.

- 30 Para implementar esta orden ls del programa de intérprete de órdenes 25A en el sistema de gestión del usuario 20, el programa de intérprete de órdenes 25A traduce las llamadas de sistema llamadas mediante la orden “ls” en una o más llamadas abstractas de sistema. Estas llamadas abstractas de sistema son enviadas al cliente de servidor virtual 30, el cual a su vez envía las llamadas abstractas de sistema a servidores de objetivo 15 apropiados, en este caso, el servidor de objetivo 15A. Tras la ejecución de la orden en los servidores de objetivo 15, los resultados son devueltos al usuario 10 mediante los programas de aplicación 25y el cliente de servidor virtual 30.

- 35 Otros programas pueden hacerse “de detección de red”. Por ejemplo, en algunas implementaciones, los intérpretes de programas, tal como los intérpretes para los lenguajes de programación Perl y Python pueden modificarse para que funcionen con el cliente de servidor virtual 30. En general, las llamadas abstractas de sistema realizadas mediante un programa de aplicación son traducidas en llamadas abstractas de sistema, que son comunicadas a través del cliente de servidor virtual 30 a los servidores 15.

- 40 Además de proporcionar órdenes del intérprete de órdenes y otros programas de aplicación, el sistema de gestión 20 puede incluir un gestor de configuración 25B. En una realización, el gestor de configuración 25B se utiliza para configurar uno o más de los servidores 15. El gestor de configuración es un programa de aplicación de software que implementa operaciones de cambio en un servidor que son a su vez traducidas en órdenes específicas para un sistema operativo correspondiente en los servidores de objetivo 15.

- 45 En una implementación, un programa de aplicación 25 dirige llamadas abstractas de sistema a servidores de objetivo 15 específicos. En otra implementación, el programa de aplicación 25 puede dirigir también llamadas abstractas de sistema a un grupo de servidores. Un grupo de servidores puede ser predefinido o definido dinámicamente sobre la base de atributos tales como sistemas operativos, capacidad, rangos de direcciones de IP y aplicaciones instaladas. Por ejemplo, el programa de aplicación 25 puede dirigir una llamada abstracta de sistema a un grupo de servidores, que consiste en un subconjunto de servidores 15 que ejecutan el sistema operativo Linux. El programa de aplicación 25 puede así desarrollar una orden en un servidor de este grupo sin especificar un servidor particular en el subconjunto. De esta manera, el programa de aplicación 25 no necesita llevar un registro de cada servidor, ni determinar qué servidores tienen suficiente capacidad o características para ejecutar el programa; por el

contrario, el programa de aplicación 25 puede desplegar órdenes (u operaciones de cambio) a un grupo predeterminado, y el cliente de servidor virtual 30 decide qué servidor específico debe ejecutar estas operaciones.

El cliente de servidor virtual 30, que puede estar incluido en el sistema de gestión 20, presenta los servidores 15 a los programas de aplicación 25 como un solo "servidor virtual" en el cual pueden ejecutarse operaciones de llamada de sistema. El cliente de servidor virtual 30 es implementado mediante una librería de software, que en una implementación es ligeramente análoga a la Librería C, libc. Los programas de aplicación 25 pueden estar estática o dinámicamente conectados a la librería de servidor virtual, que se denomina libnc. En una realización, ningún programa de aplicación 25 que detecta red es convertido en programas que detectan red remplazando llamadas en la librería libc con llamadas equivalentes a la librería libnc, que proporciona llamadas abstractas de sistema que detectan red.

En una realización alternativa, el cliente de servidor virtual 30 puede ser implementado como parte de un sistema operativo. Por ejemplo, el sistema operativo que ejecuta el sistema de gestión de 20 del usuario puede recibir llamadas abstractas de sistema y comunicarlas a los servidores de objetivo 15 remotos. De acuerdo con esto, con vistas a ejecutar una llamada abstracta de sistema a servidores de objetivo 15, la fuente de la llamada abstracta de sistema es inmaterial.

En algunas realizaciones, el cliente de servidor virtual 30 se comunica con los servidores 15 a través de los agentes de servidor virtual 35 asociados con los servidores 15, que se describirán con detalle en lo que sigue. El cliente de servidor virtual 30 se comunica con agentes de servidor virtual 35 para presentar los múltiples servidores de objetivo 15 físicos como un solo servidor virtual a los programas de aplicación 25. Como representación abstracta de una colección de los servidores 15 físicos, el servidor virtual intercepta las llamadas abstractas de sistema por medio del cliente de servidor virtual 30 y encamina las llamadas abstractas de sistema hacia los agentes de servidor virtual 35.

Cuando el cliente de servidor virtual 30 recibe una llamada abstracta de sistema desde un programa de aplicación 25, el cliente de servidor virtual 30 comprueba la llamada abstracta de sistema para determinar si esta llamada de sistema es una llamada local o una llamada remota. Si la llamada abstracta de sistema es una llamada local, entonces el sistema operativo que ejecuta el sistema de gestión 20 ejecuta la llamada de sistema localmente. Si se determina que la llamada abstracta de sistema es una llamada remota, el cliente de servidor virtual 30 envía la llamada abstracta de sistema a un agente de servidor virtual 35 asociado con un servidor de objetivo 15 por medio de un protocolo de mensajes. Por ejemplo, cuando una llamada abstracta de sistema "nuevamente abierta", que representa una orden abierta de archivo remoto, es recibida por el cliente de servidor virtual 30, los datos que representan la orden "nuevamente abierta" y los parámetros asociados con la orden "nuevamente abierta" son enviados a los agentes de servidor virtual 35 apropiados. Los servidores de objetivo 15 para una llamada de sistema son identificados por el usuario 10 o por los programas de aplicación 25. El cliente de servidor virtual 30 identifica los servidores de objetivo 15 a partir de sus agentes de servidor virtual 35, y determina adónde debe dirigirse el sistema.

Los agentes de servidor virtual 35 reciben llamadas abstractas de sistema desde el cliente de servidor virtual 30 y preparan las llamadas abstractas de sistema para sus servidores de objetivo 15 asociados. Cuando el cliente de servidor virtual 30 determina a qué agentes de servidor virtual debe dirigirse una llamada abstracta de sistema, cada uno de los agentes de servidor virtual 35 recibe la llamada abstracta de sistema. Como parte de la preparación de la llamada abstracta de sistema para los servidores de objetivo 15 asociados, los agentes de servidor virtual 35 proporcionan medidas de seguridad para asegurar que el usuario 10 está autorizado a acceder a los servidores de objetivo 15 y, que el agente de servidor virtual 35 controla el acceso del usuario, tal como se facilita mediante el servidor de objetivo 15 asociado. Una vez que el usuario 10 está autorizado, el agente de servidor virtual 35 traduce la llamada abstracta de sistema en una llamada específica para el sistema operativo dirigida a su servidor de objetivo 15 asociado. El servidor de objetivo 15 ejecuta la llamada abstracta de sistema y devuelve los resultados al agente de servidor virtual 35, que a su vez envía los resultados de nuevo a los programas de aplicación 25 apropiados a través del cliente de servidor virtual 30.

En una realización, los agentes de servidor virtual 35 (denominados también agentes de Demonio de llamada de sistema remoto o agentes RSCD (Remote System Call Daemon, en inglés)) son módulos de software conectados a sus correspondientes servidores de objetivo 15. En otra realización, los agentes de servidor virtual 35 son módulos de software que no están conectados a sus correspondientes servidores de objetivo 15, sino que están en comunicación con sus servidores de objetivo 15 situados remotamente asociados.

En algunas realizaciones, en lugar de actuar como un intercambiador de mensajes que envía una llamada abstracta de sistema a un servidor de objetivo 15 específico, uno de los agentes de servidor virtual 35 puede representar a un grupo de servidores físicos. Así, si es necesario ejecutar la misma orden en múltiples servidores, estos servidores pueden ser agregados en un grupo, representado por un único agente de servidor virtual 35, de manera que puedan realizarse llamadas de sistema apropiadas a un grupo de servidores simultáneamente a través de ese agente de servidor virtual 35.

Generalmente, las llamadas abstractas de sistema pueden incluir todos los tipos de llamadas de sistema, incluidas las llamadas de sistema de archivo, las llamadas de sistema operativo y otras similares. Una llamada abstracta de sistema se implementa típicamente como una modificación de una llamada específica para un sistema operativo

estándar análoga. Por ejemplo, la llamada abstracta de sistema “nuevamente abierta” es una llamada de sistema estándar “abierta”, que abre un archivo en un servidor.

Con pequeñas modificaciones a un código fuente de programa de aplicación, cualquier programa de aplicación puede realizar llamadas abstractas de sistema agnósticas del sistema operativo. Cambiando las llamadas de sistema a llamadas abstractas de sistema, puede realizarse cualquier programa de aplicación genérico en una aplicación que detecta red que puede operar de manera transparente en servidores que soportan diferentes sistemas operativos.

En una realización, solo las llamadas de sistema que son aplicables a todos los servidores de objetivo 15 pueden ser modelizadas como llamadas abstractas de sistema. Por ejemplo, si los servidores de objetivo 15 incluyen servidores de Unix, puede no ser posible modelizar una llamada de sistema para actualizar un registro como una llamada abstracta de sistema, puesto que un registro, que es un objeto específico de Windows, no existe y no tiene relevancia para plataformas de servidor de Unix.

En referencia a la FIG. 2, en una realización, el cliente de servidor virtual 30 incluye varios módulos de software que implementan su funcionalidad. Estos módulos incluyen un receptor 40 que recibe una llamada abstracta de sistema realizada mediante un programa de aplicación 25, y envía la llamada abstracta de sistema a un instanciador 42. El receptor 40 es un módulo de software que actúa como un intercambiador de mensajes entre los programas de aplicación 25 y el instanciador 42. En una realización, el receptor 40 recibe la llamada abstracta de sistema desde uno de los programas de aplicación 25 de software utilizados por el usuario 10. El receptor 40, a continuación, envía la llamada abstracta de sistema directamente al instanciador 42. En otra realización, el receptor 40 puede recibir llamadas de sistema específicas para un sistema operativo estándar desde un programa de aplicación 25. El receptor envía tales llamadas abstractas de sistema al instanciador 42 para que el instanciador 42 decida adónde deben dirigirse las llamadas de sistema.

El instanciador 42 instancia llamadas abstractas de sistema en un modo de proceso seguro. La instanciación de proceso seguro comparte un único recurso entre múltiples operaciones sin requerir cambios a la arquitectura de los programas de aplicación que solicitan las operaciones. Típicamente, la instanciación de proceso seguro comparte el mismo cliente de servidor virtual 30 entre múltiples ejecuciones simultáneas de llamadas de sistema. El uso del recurso compartido, tal como el cliente de servidor virtual 30, está coordinado, de manera que la ejecución de una operación no impacta la ejecución de otras operaciones. En una realización de la instanciación de proceso seguro, los programas de aplicación 25 pueden instanciar múltiples órdenes (u operaciones) por medio del instanciador 42. Por ejemplo, los programas de aplicación 25 pueden invocar múltiples llamadas de sistema “reabiertas” que son dirigidas a uno o más servidores de objetivo 15. La llamada de sistema “reabierta” es recibida por el instanciador 42 en el cliente de servidor virtual 30. El instanciador 42 distribuye entonces la llamada abstracta de sistema “reabierta” a cada uno de los agentes de servidor virtual asociados con los servidores de objetivo, de manera que los servidores de objetivo 15 pueden ejecutar simultáneamente múltiples llamadas “reabiertas”.

En una realización, el instanciador 42 se implementa como una librería de software que proporciona rutinas que representan las llamadas abstractas de sistema. Una implementación particular de la librería de software se denomina “libnc”. Libnc es una librería “que detecta red” que es análoga a la librería C estándar. La librería Libnc soporta los programas de aplicación 25 que detectan red mediante la instanciación de las llamadas abstractas de sistema generadas mediante los programas de aplicación 25.

En una realización, el instanciador 42 determina a qué agentes de servidor virtual 35 debe dirigirse una llamada abstracta de sistema. El instanciador 42 identifica servidores de objetivo 15 encontrando los identificadores de servidor de objetivo especificados en la llamada abstracta de sistema. El identificador de servidor de objetivo puede incluir un nombre de ruta, que a su vez puede incluir un nombre de anfitrión o una dirección de red (por ejemplo, una dirección de IP) para el servidor. El servidor de objetivo 15 puede asimismo ser identificado mediante nombres de servidor indicados explícitamente en un archivo que debe ser ejecutado sobre servidores de nombres específicos. Si no, la identidad de servidor puede obtenerse a partir de un subconjunto de servidores o de un grupo de servidores (por ejemplo, un grupo de servidores de Linux) al cual pertenece el servidor de objetivo 15.

Antes de transmitir la llamada abstracta de sistema a los agentes de servidor virtual 35, el instanciador 42 puede especificar también la prioridad, uso de CPU, y el uso de memoria de la llamada de sistema para los servidores de objetivo 15, de manera que las plataformas del servidor de objetivo 15 identificado pueden realizar los servicios solicitados tal como el cliente de servidor virtual 30 especifica. Una vez que la llamada abstracta de sistema ha sido instanciada, se envía a un codificador 44 para su posterior procesamiento.

El codificador 44 codifica la llamada abstracta de sistema antes de enviarlo a un transmisor 46 para su transmisión a los agentes de servidor virtual 35. El codificador 44 utiliza protocolos de codificación y algoritmos estándar para hacer la comunicación segura entre el cliente de servidor virtual 30 y los agentes de servidor virtual 35. Ejemplos de protocolos de codificación estándar incluyen, pero no están limitados a, protocolos de SSL (Capa de conexión segura – Secure Sockets Layer, en inglés), Kerberos y de secreto compartido. SSL utiliza una clave pública para codificar datos. Kerberos asigna una única clave a cada usuario autorizado. El algoritmo de codificación estándar

incluye, pero no está limitado a, DES (Estándar de codificación de datos – Data Encryption Standard, en inglés), 3DES (Triple DES), Blowfish y AES (Estándar de codificación avanzado – Advanced Encryption Standard, en inglés).

El protocolo y el algoritmo de codificación utilizados por el instanciador 44 deben estar soportados por cada agente de servidor virtual 35 con el cual se va a comunicar el cliente de servidor virtual 30. Por ejemplo, si el cliente de servidor virtual 30 soporta SSL, el agente de servidor virtual 35 debe ser capaz de soportar SSL para ese protocolo que se va a utilizar. Si el cliente de servidor virtual 30 soporta Kerberos, el agente de servidor virtual 35 debe asimismo ser capaz de soportar Kerberos para ese protocolo que se va a utilizar.

El transmisor 46 utiliza un protocolo de interfaz de red, tal como TCP / IP o Ethernet, para enviar la llamada abstracta de sistema sobre una red a los agentes de servidor virtual 35. El transmisor transmite la misma llamada abstracta de sistema a cada agente de servidor virtual objetivo. En una realización, el transmisor 46 utiliza una dirección de IP para determinar a qué servidores de objetivo 15 debe enviarse una llamada abstracta de sistema. Una dirección de IP puede estar incluida directamente en la llamada abstracta de sistema o puede obtenerse a partir del identificador de servidor incluido en la llamada abstracta de sistema. El agente de servidor virtual 35 acepta la llamada abstracta de sistema que contiene la dirección de IP del servidor de objetivo 15 asociado con ese agente de servidor virtual 35. Una vez que el agente de servidor virtual 35 recibe la llamada abstracta de sistema, el agente de servidor virtual 35 procesa la llamada abstracta de sistema para su ejecución en el servidor de objetivo 15.

En referencia a la FIG. 3, cada agente de servidor virtual 35 incluye módulos de software que implementan su funcionalidad. Estos módulos incluyen un receptor 50, que recibe llamadas abstractas de sistema del cliente de servidor virtual 30, y transfiere las llamadas abstractas de sistema a un módulo descodificador 52.

Antes de que el usuario 10 pueda acceder al sistema de gestión de usuario 20, el usuario 10 es autenticado para asegurar que el usuario 10 es en realidad la persona que dice ser. El usuario 10 puede ser autenticado de muchas maneras. En una realización, el usuario 10 es autenticado por el sistema operativo del sistema de gestión 20, y los servidores de objetivo 15, a continuación, heredan la identidad del usuario 10. En otra realización, se utilizan certificados de SRP (Clave remota secreta – Secret Remote Password, en inglés) o de PKI (Certificados X.509) para autenticar al usuario 10. En otra realización más, puede utilizarse el sistema Kerberos 5 para autenticar al usuario 10 mediante la asignación de una clave privada única al usuario 10.

El módulo identificador de fuente 52 identifica la máquina de fuente, por ejemplo, el sistema de gestión 20 del usuario. El módulo identificador de fuente 52 determina en primer lugar la máquina de fuente a través de la dirección de red (por ejemplo, dirección de IP) que fue presentada al agente de servidor virtual 35 desde el cliente de servidor virtual 30 con la llamada abstracta de sistema y comprueba si el anfitrión de fuente está autorizado.

Identificando la máquina de fuente, el módulo de fuente 52 determina los protocolos de seguridad que van a ser utilizados por el agente de servidor virtual 35 para la codificación y la descodificación. En una realización, el agente de servidor virtual 35 puede soportar diferentes protocolos de seguridad. Por ejemplo, el agente de servidor virtual 35 puede soportar de manera flexible bien SSL o bien Kerberos sobre la base del protocolo de seguridad de los datos entrantes desde el cliente de servidor virtual 30. A continuación, la llamada abstracta de sistema es enviada a un descodificador 54, que descodifica la llamada abstracta de sistema. A partir de la llamada abstracta de sistema descodificada, el módulo identificador de usuario 55 identifica al usuario 10 invocando los programas de aplicación 25 de la máquina de fuente y verifica que el usuario 10 está autorizado a acceder a la máquina de fuente.

Después de que el usuario es identificado por el identificador de usuario 55, un mapeador de identidad 56 y un imitador 58 proporcionan medidas de seguridad adicionales cuando el usuario 10 intenta acceder a los servidores de objetivo 15 remotos desde el sistema de gestión 20 del usuario. El mapeador de identidad 56 opcionalmente mapea al usuario autenticado (usuario presentado) a otro usuario (usuario efectivo) y sitúa una identidad de usuario local en el servidor de objetivo 15 que corresponde a la identidad autenticada del usuario efectivo. A través del imitador 58, el usuario 10 es imitado en un servidor de objetivo 15 remoto, de manera que si el usuario efectivo es identificado y existe como usuario local en el servidor de objetivo 15 remoto, el usuario 10 toma la identidad local del usuario efectivo y los permisos proporcionados por la identidad en el servidor de objetivo 15 remoto. Así, el acceso del usuario 10 al servidor de objetivo 15 remoto está además restringido a los niveles apropiados proporcionados por los permisos concedidos a la identidad local de usuario efectiva en el servidor de objetivo 15 remoto. Por ejemplo, si el usuario 10 es autenticado como “Joe” en el sistema de gestión 20 y mapeado a un usuario efectivo “Jane”, los permisos locales de “Jane” estarán disponibles para el usuario 10 en el servidor de objetivo 15 remoto. Si “Jane” no existe en el servidor de objetivo 15 remoto, entonces al usuario 10 se le proporcionará una cuenta de invitado. En una realización, la combinación del usuario presentado y la función, que está definida por el Control de acceso basado en función (RBAC – Role Based Access Control, en inglés), es mapeada a un usuario efectivo. Por ejemplo, el usuario “Joe” que tiene la función de un administrador junior puede ser mapeado a un usuario efectivo denominado “administrador junior”. Otro usuario “Bob” que también tiene la función de un administrador junior puede ser mapeado al mismo usuario efectivo llamado “administrador junior”.

El acceso del usuario efectivo para el usuario 10 presentado está además restringido por un autorizador 60, que permite al usuario 10 realizar acciones predeterminadas o acceder a recursos predeterminados en un servidor de objetivo 15 particular. Esto se consigue utilizando Listas de control de Acceso (ACL) para gestionar el acceso del

usuario efectivo a recursos en los servidores de objetivo 15 remotos. La ACL informa a los sistemas operativos de los servidores de objetivo 15 remotos de los derechos de acceso del usuario efectivo en recursos específicos de servidor, tales como archivos o directorios. Por ejemplo, si el usuario 10 es mapeado al usuario efectivo "administrador junior", entonces el usuario 10 solo está autorizado a poner en práctica órdenes de solo lectura en ciertos directorios o archivos de un grupo de servidores de objetivo 15 remotos y no puede efectuar ningún cambio en los servidores de objetivo 15.

Después de que el usuario está autorizado, un traductor 62 traduce la llamada abstracta de sistema en una llamada estándar del sistema operativo que puede entender y ejecutar el servidor de objetivo 15. El traductor 62 examina la llamada abstracta de sistema e identifica una llamada de sistema específica para un sistema operativo estándar que es análoga a la llamada abstracta de sistema y que está soportada por el sistema operativo que ejecuta el servidor de objetivo 15 asociado. Una vez que la llamada estándar de sistema análoga es identificada, el traductor cambia la llamada abstracta de sistema en la llamada estándar de sistema. Esta llamada estándar de sistema operativo es enviada a un ejecutor 66 para su ejecución en el servidor de objetivo 15.

Una vez que el ejecutor 66 recibe una llamada estándar del sistema operativo, el ejecutor 66 realiza los servicios que son solicitados por la llamada estándar del sistema. En una realización, el ejecutor 66 es el sistema operativo que ejecuta el servidor de objetivo 15. El sistema operativo examina las llamadas de sistema y lleva a cabo las operaciones solicitadas por la llamada de sistema, por ejemplo, comunicándose con otras aplicaciones que se ejecutan en el servidor de objetivo 15.

Cada agente de servidor virtual 35 guarda un informe de auditoría 64 para hacer un seguimiento de los nombres de los usuarios y de todas las actividades realizadas por cada usuario, y para resolver los problemas de cambios en el servidor y los errores de configuración. Por ejemplo, el informe de auditoría 64 guarda información acerca de las actividades solicitadas y realizadas por usuarios autorizados, información acerca de los datos, tales como las llamadas de sistema y los resultados de las llamadas de sistema, que fueron recibidas y enviadas entre el cliente de servidor virtual 30 y el agente de servidor virtual 35, así como todos los parámetros asociados con la llamada abstracta de sistema. El contenido del informe de auditoría 64 es a continuación transmitido a un informe agregado centralizado mantenido para todos los agentes de servidor virtual 35.

A continuación se muestra un primer ejemplo de las medidas de seguridad incorporadas en una realización de la implementación de servidor de objetivo. En primer lugar, el usuario 10 inicia una sesión en el sistema de gestión y es autenticado como "Joe" durante el proceso de inicio de sesión. Este proceso de autenticación puede alcanzarse utilizando un servidor de autenticación de red conocido, tal como NTLM, K5, AD, APM, NIS, etc., dependiendo del sistema operativo que se ejecuta en el sistema de gestión 20. Después de que el usuario "Joe" está autenticado en el sistema de gestión 20, el usuario "Joe" es autenticado para los servidores de objetivo 15 heredando la identidad "Joe" del usuario a través del sistema de gestión 20.

A continuación, el usuario 10 introduce una orden "ls", solicitando una enumeración de los archivos en el servidor de objetivo 15A remoto, mediante el programa de órdenes del intérprete de órdenes 25A en el sistema de gestión 20. El programa de órdenes del intérprete de órdenes 25A genera una llamada abstracta de sistema en respuesta a la orden y envía la llamada abstracta de sistema al cliente de servidor virtual 30 para continuar con la solicitud del usuario 10. El cliente de servidor virtual 30 examina la configuración de seguridad de la llamada abstracta de sistema y codifica la llamada de sistema utilizando un esquema de clave secreta compartida con un algoritmo de codificación, tal como DES, 3DES o Blowfish. Una vez que la llamada abstracta de sistema está codificada, la llamada de sistema es comunicada a través de una red al agente de servidor virtual 35A del servidor de objetivo 15A.

Cuando el agente de servidor virtual 35A recibe la llamada abstracta de sistema, el agente de servidor de objetivo 15A 35A intenta descodificar el mensaje utilizando la clave secreta compartida con el cliente de servidor virtual 30. El agente de servidor virtual 35A mira si el usuario "Joe" está reconocido como un usuario local en el servidor de objetivo 15A mediante un usuario efectivo. Si el usuario "Joe" está reconocido como un usuario local, entonces el agente de servidor virtual examina la lista de control de acceso para determinar si la combinación de usuario "Joe" 10, servidor de objetivo 15A y la llamada abstracta de sistema está permitida. Si la combinación está permitida, entonces la lista de control de acceso se utiliza para determinar si existe alguna otra restricción aplicable al acceso del usuario 10 al servidor de objetivo 15A. El agente de servidor virtual 35A ejecuta la llamada de sistema de acuerdo con cualquier restricción de seguridad y codifica los resultados utilizando la misma clave secreta compartida. Los resultados de la orden "ls" son devueltos al cliente de servidor virtual 30, donde son descodificados y mostrados al usuario.

En un segundo ejemplo de medidas de seguridad incorporadas en una realización de servidor virtual, el usuario 10 es autenticado utilizando certificados de SRP o de PKI. Una vez que el usuario 10 está autenticado el usuario 10 introduce una orden "ls", solicitando una enumeración de los archivos del servidor de objetivo 15A, mediante el programa de órdenes de intérprete de órdenes 25A en el sistema de gestión 20. El programa de órdenes del intérprete de órdenes 25A genera una llamada abstracta de sistema en respuesta a la orden y envía la llamada abstracta de sistema al cliente de servidor virtual 30. El cliente de servidor virtual 30 examina la configuración de seguridad de la llamada abstracta de sistema y codifica la llamada abstracta de sistema utilizando codificación de

clave pública. Pueden utilizarse algoritmos de codificación estándar, tales como DES, 3DES o Blowfish para cambiar la clave de la sesión entre el cliente de servidor virtual 30 y el agente de servidor virtual 35A para establecer una sesión de comunicación entre ellos.

Tras descodificar la llamada abstracta de sistema recibida por el agente de servidor virtual 35A, el agente de servidor virtual 35A mira si el usuario "Joe" está reconocido como un usuario local en el servidor de objetivo 15A mediante un usuario efectivo. Si el usuario "Joe" está reconocido como un usuario local, entonces el agente de servidor virtual 35A examina la ACL para determinar si la combinación de usuario 10, servidor de objetivo 15A y la llamada abstracta de sistema está permitida. Si la combinación está permitida, entonces la lista de control de acceso se utiliza para determinar si existe alguna otra restricción aplicable al acceso del usuario 10 al servidor de objetivo 15A. El agente de servidor virtual 35A ejecuta la llamada de sistema de acuerdo con cualquier restricción de seguridad, y codifica los resultados utilizando la clave de sesión establecida. Los resultados de la orden "ls" son devueltos a continuación al cliente de servidor virtual 30, donde son descodificados y mostrados al usuario.

A continuación, se muestra un tercer ejemplo de las medidas de seguridad incorporadas en una realización de servidor virtual. Si el sistema de gestión 20 dispone de una infraestructura de Kerberos 5 (K5), el usuario 10 puede ser autenticado introduciendo una clave de Kerberos en el sistema de gestión 20. Una vez que el usuario 10 ha iniciado una sesión como el usuario "Joe" autenticado, el usuario 10 introduce la orden "ls", solicitando una enumeración de los archivos del servidor de objetivo 15A remoto, mediante el programa de órdenes de intérprete de órdenes 25A en el sistema de gestión 20. El programa de órdenes 25A del intérprete de órdenes genera una llamada abstracta de sistema en respuesta a la orden y envía la llamada abstracta de sistema al cliente de servidor virtual 30 para continuar con la solicitud del usuario 10. El cliente de servidor virtual 30 a continuación envía la llamada abstracta de sistema y un tique de Kerberos, que es obtenido del controlador de dominio Kerberos (KDC – Kerberos Domain Controller, en inglés) al agente de servidor virtual 35A.

Después de que el agente de servidor virtual 35A recibe la llamada abstracta de sistema y el tique, el agente de servidor virtual 35A valida la llamada abstracta de sistema verificando el tique por medio del KDC. Una vez validado, el agente de servidor virtual 35A mira si el usuario "Joe" está reconocido como un usuario local en el servidor de objetivo 15A mediante un usuario efectivo. Si el usuario "Joe" está reconocido como un usuario local, entonces el agente de servidor virtual examina la ACL para determinar si la combinación del usuario "Joe" 10, el servidor de objetivo 15A y la llamada abstracta de sistema está permitida. Si la combinación está permitida, entonces la lista de control de acceso se utiliza para determinar si alguna otra restricción aplica al acceso del usuario 10 al servidor de objetivo 15A. El agente de servidor virtual 35A ejecuta la llamada de sistema de acuerdo con todas las restricciones de seguridad y codifica los resultados utilizando una clave de Kerberos. Los resultados de la orden "ls" son devueltos al cliente de servidor virtual 30, donde son descodificados y mostrados al usuario.

En referencia ahora a la FIG. 4, se describe un método para gestionar múltiples servidores como un solo servidor virtual. En primer lugar, en la etapa 400, el sistema representa múltiples servidores como un solo servidor virtual. A continuación, en la etapa 410, sobre la base de una solicitud del usuario para realizar operaciones en los servidores de objetivo, el cliente de servidor virtual 30 recibe una llamada abstracta de sistema desde un programa de aplicación 25. Finalmente, en la etapa 420, el cliente de servidor virtual instancia las llamadas abstractas de sistema y envía la llamada abstracta de sistema a los agentes de servidor virtual 35 para su ejecución.

La FIG. 5 muestra las etapas implicadas en la instanciación de una llamada abstracta de sistema. Primero, en la etapa 422, el cliente de servidor virtual 30 identifica a los servidores de objetivo 15 mediante los identificadores de servidor de objetivo proporcionados en la llamada abstracta de sistema. Una vez que los servidores de objetivo están identificados, en la etapa 424, la llamada abstracta de sistema es transmitida a los agentes de servidor virtual asociados con los servidores de objetivo identificados. Los agentes de servidor virtual 35 preparan la llamada abstracta de sistema para los servidores de objetivo 15, de manera que la llamada abstracta de sistema pueda ser ejecutada en los servidores de objetivo 15. Por ejemplo, para el servidor de objetivo 15A, las llamadas abstractas de sistema son traducidas a llamadas específicas de sistema de Windows NT/W2K estándar, que son ejecutables mediante el sistema operativo que se ejecuta en el servidor de objetivo 15A. Cuando se completa la ejecución de la llamada de sistema, en la etapa 426, el cliente de servidor virtual 30 recibe los resultados de la ejecución desde los agentes de servidor virtual 35.

En una realización, múltiples órdenes generan múltiples llamadas de sistema, que pueden ser agregadas en una sola llamada abstracta de sistema de alto nivel, mediante el programa de aplicación 25. Por ejemplo, si dos órdenes, tales como las órdenes de permiso para copiar y cambiar, deben dársele a un servidor de objetivo 15A, las llamadas abstractas de sistema que llevan a cabo estas órdenes, tales como llamadas de sistema reabrir, releer, reescribir y rchmod, pueden ser agregadas en una sola llamada abstracta de sistema de alto nivel. Cuando se recibe por parte del cliente de servidor virtual 30, el cliente de servidor virtual 30 puede desintegrar la llamada abstracta de sistema de alto nivel en las llamadas abstractas de sistema originales, y transmitir las llamadas abstractas de sistema separadamente al agente de servidor virtual 35. En otra realización, en lugar de desintegrar la llamada de sistema de alto nivel en las llamadas abstractas de sistema originales en el cliente de servidor virtual 30, la llamada abstracta de sistema de alto nivel es recibida por un agente de servidor virtual 35, que a su vez traduce la llamada abstracta de sistema de alto nivel en llamadas de sistema específicas para el sistema operativo separadas para ser ejecutadas en el servidor de objetivo 15.

La FIG. 6 es una instantánea de pantalla que muestra una orden cuando se está emitiendo a múltiples servidores mediante el sistema de gestión 20. Como se muestra en esta memoria, los nombres de servidor utilizados como parámetros para órdenes están precedidos de dos barras inclinadas, para distinguirlos de un nombre de ruta, el cual está generalmente separado por una barra inclinada. Por ejemplo, “//redhatbiz1/etc” especifica la ruta /etc en el servidor de nombre “redhatbiz1”. Así, como se ve en la instantánea de pantalla, para comparar el archivo “/etc/hosts” en dos servidores diferentes, uno de nombre “redhatbiz1” y otro de nombre “redhatbiz2”, el usuario 10 introduce la orden “diff//redhatbiz1/etc/hosts //redhatbiz2/etc/hosts”.

En referencia de nuevo a la FIG. 1, en una realización alternativa, el usuario 10 gestiona los servidores de objetivo 15 ejecutando y deshaciendo operaciones de cambio en un servidor distribuido en los servidores de objetivo 15 en una transacción en modo seguro, utilizando la implementación de servidor virtual descrita anteriormente. Las operaciones de cambio en un servidor distribuido solicitan a los sistemas operativos de los servidores de objetivo 15 que actualicen borren, instalen y/o copien los equipos de servidor y/o las entradas de archivo de configuración de los servidores de objetivo 15. Las operaciones de cambio en servidor en modo de transacción segura aseguran que todas las etapas requeridas de cada operación de cambio en servidor se completan antes de que las operaciones de cambio en servidor distribuido se consideren completadas. Además, si se produce un error mientras se llevan a cabo las etapas requeridas en los servidores de objetivo 15, cualquier cambio realizado desde estas etapas se deshace, y los valores de los equipos de los servidores de objetivo 15 y/o las entradas de configuración vuelven a los valores que tenían antes de la ejecución de las operaciones de cambio en el servidor. En una realización, los programas de aplicación 25 pueden generar un paquete de transacción que agrupa un conjunto de instrucciones y los contenidos de servidor necesarios para que el sistema operativo de cada uno de los servidores de objetivo 15 realice las operaciones de cambio en el servidor.

En referencia a la FIG. 7, en una realización, el gestor de configuración 25B genera un paquete de transacción 700 que incluye archivos o entradas de archivos de configuración 705 (denominadas en conjunto objetos de servidor), un archivo de parámetros 710 y un conjunto de instrucciones 715 para llevar a cabo las operaciones de cambio en servidor en uno o más de los servidores de objetivo 15 que están especificados mediante un archivo externo, como solicitó el gestor de configuración 25B.

En una realización, el conjunto de instrucciones 715 incluye una secuencia de ejecución de las operaciones de cambio en servidor proporcionadas para los sistemas operativos de los servidores de objetivo 15 que llevan a cabo las operaciones de cambio en servidor. Si esta información no se proporciona en el conjunto de instrucciones 715 en el paquete de transacción 700, se accede a un diagrama de dependencia externa 720 para proporcionar una secuencia de ejecución de las operaciones de cambio en servidor. Por ejemplo, el diagrama de dependencia externa 720 puede proporcionar información acerca de relaciones direccionales entre objetos del servidor. En particular, si el programa A basado en NT es un prerrequisito para otro programa B basado en NT, para ejecutar correctamente los programas A y B, el programa A debe iniciarse antes que el programa B y el programa B debe finalizar antes que el programa A. Aunque la información de secuencia se utiliza para ordenar la secuencia de operaciones de cambio para los objetos de servidor que están especificados en el paquete de transacción, la información de la secuencia se utiliza también para añadir operaciones de cambio en un objeto de servidor implicado para objetos de servidor correspondientes, tal como los objetos de servidor que dependen de y/o que dependen de estos objetos de servidor especificados, que no se especifican en el paquete de transacción. En particular, continuando desde el ejemplo anterior, si la única instrucción de cambio proporcionada en un paquete de transacción es detener el programa A, la información de la secuencia añade la instrucción requerida para detener el programa B y a continuación detener el programa A sobre la base de la relación direccional entre los programas A y B. Así, la información de secuencia del diagrama de dependencia determina las secuencias de operaciones de cambio en servidor que deben llevarse a cabo no solo en los objetos de servidor especificados, sino también en sus objetos de servidor correspondientes. Si se produce un error mientras se realizan operaciones de cambio en servidor, la información de la secuencia provoca también que las operaciones de cambio en el servidor se detengan y que se reviertan no solo en los servidores especificados, sino también en los objetos de servidor correspondientes.

Como se ha descrito anteriormente, si el conjunto de instrucciones 715 proporciona la información de secuencia para las operaciones de cambio en un servidor, el conjunto de instrucciones 715 sobrescribe la información de secuencia proporcionada por el diagrama de dependencia 720. De manera similar a la información de secuencia proporcionada por el diagrama de dependencia 720, el conjunto de instrucciones 715 proporciona la información relativa al orden en el cual deben llevarse a cabo las operaciones de cambio en el servidor. Se proporcionan los objetos de servidor correspondientes de los objetos de servidor especificados, de manera que las operaciones de cambio en servidor pueden efectuar cambios en los objetos de servidor correspondientes, así como en los objetos de servidor especificados. El conjunto de instrucciones 715 proporciona asimismo información de dependencia entre tipos de servidores. Por ejemplo, si un servidor de aplicación depende de un servidor de base de datos, la información de secuencia proporcionada en el conjunto de instrucciones 715 dará instrucciones de ejecutar las operaciones de cambio en el servidor de base de datos antes de ejecutar las operaciones de cambio en el servidor de aplicaciones.

En una realización, el conjunto de instrucciones 715 especifica que las operaciones de cambio en un servidor se realicen en cualquiera de los cuatro tipos de objetos de servidor 705: objetos de servidor primitivo, objetos de servidor compuesto, objetos de servidor de configuración abstracta y objetos de servidor de componentes. Un objeto

de servidor primitivo es un objeto de servidor elemental que sirve como base para todos los demás tipos de objetos de servidor. Por ejemplo, para servidores de Linux, los objetos de servidor primitivo incluyen, pero no están limitados a, archivos, directorios, archivos de gestor de paquetes Redhat y entradas de archivo de configuración para archivos de configuración de texto, tales como el archivo "inetd.conf". Para servidores de Solaris, los objetos de servidor primitivo incluyen, pero no están limitados a, archivos, directorios, paquetes, parches y entradas de archivo de configuración para archivos de configuración, tales como el archivo "inetd.conf". Para servidores de MS NT o de W2K, los objetos de servidor primitivo incluyen, pero no están limitados a, archivos, ACL de archivos, directorios, ACL de directorios, programas de aplicación, parches de resolución de problemas (hot fixes, en inglés), las entradas de registro, ACL de entradas de registro, entradas de catálogo COM/COM+ (modelo de objeto de componente), entradas de Metabase, usuarios, cuentas y entradas de archivo de configuración para todos los archivos de configuración, tales como archivos ".ini".

Un objeto de servidor compuesto es un objeto que contiene objetos de servidor primitivo y otros objetos de servidor compuesto relacionados. Por ejemplo, un objeto modelo de objeto de componente extendido (COM+ - Component Object Model +, en inglés), un objeto de servidor compuesto de NT o de W2K, contiene objetos de servidor primitivo, tales como una entrada de catálogo COM+, entradas de registro de NT y archivos DLL. En otro ejemplo más, un objeto JavaBeans de empresa (EJB - Enterprise JavaBeans, en inglés), un objeto de servidor compuesto, contiene objetos de servidor primitivo que incluyen un archivo Java (JAR - Java ARchive, en inglés) y múltiples entradas de archivo de configuración. En otro ejemplo, un proceso de servidor es un objeto de servidor compuesto, que contiene objetos de servidor primitivo, tales como entradas de archivo de configuración (por ejemplo, una entrada de permiso, una entrada de prioridad, una entrada de señal de control), archivos y ejecutables.

Un objeto de servidor de configuración abstracta es un tipo especial de un objeto de servidor primitivo que representa una entrada en un archivo de configuración por medio de una entrada correspondiente en un archivo de configuración abstracta, donde el mapeo de un archivo de configuración a un formato de configuración abstracta común es proporcionado mediante una gramática específica para un archivo de configuración. Por ejemplo, en el entorno MS NT / W2K, las entradas de archivo de configuración se almacenan en archivos ".ini" o en archivos de configuración XML. En el entorno UNIX, las entradas de archivo de configuración se almacenan en archivos de texto tales como archivos "inetd.conf" o "httpd.conf" o archivos de configuración XML.

Para conciliar la diferencia entre los formatos de entrada de archivo de configuración en los diferentes servidores, se proporciona un formato de configuración abstracta común mediante la normalización de las entradas del archivo de configuración mediante una gramática específica para un archivo de configuración soportado. Modelizando cada entrada de archivo de configuración abstracta como una entrada de archivo de configuración abstracta mediante este proceso de normalización, las operaciones de cambio en servidor pueden realizarse basándose en las entradas del archivo de configuración abstracta normalizadas. Las operaciones de cambio solicitadas mediante las entradas de archivo de configuración abstracta son realizadas, y los cambios se comunican a continuación a las entradas de archivo de configuración real. Así, en esta realización, las entradas del archivo de configuración pueden ser gestionadas individualmente mediante el uso de entradas de archivo de configuración abstracta, sin tener que cambiar todo el archivo de configuración cada vez que una operación de cambio en servidor cambia una entrada individual. Pueden existir gramáticas específicas para un archivo de configuración para numerosos sistemas, incluidos Solaris, Linux, NT4 / W2K, Apache, Web Logic y Web Sphere.

Un objeto de servidor de componentes es una colección secuenciada de objetos de servidor. Por ejemplo, un Paquete de servicios de NT es una colección secuenciada de parches para resolución de problemas de NT para ser aplicados en un orden predefinido. Por consiguiente, una colección de operaciones de cambio correspondientes predefinidas puede ser efectuada en orden mediante un servidor de componentes.

Además de las circunscripciones del conjunto de instrucciones 715 descritas anteriormente, el conjunto de instrucciones 715 especifica las operaciones de cambio en servidor para ser realizadas en servidores de objetivo 15 de una colección de objetos de servidor predeterminados mediante comunicación con los objetos de servidor (por ejemplo, archivos o entradas de archivo de configuración 705), el diagrama de dependencia 720 y el archivo de parámetros 710. Las operaciones de cambio en servidor pueden utilizarse para desplegar o copiar archivos, directorios y paquetes de software en servidores de objetivo 15. Asimismo, pueden utilizarse operaciones para editar entradas de archivo de configuración 705 sin tener que abrir una sesión en el servidor de objetivo 15. En una realización, el conjunto de instrucciones 715 proporciona la información que necesitan los servidores de objetivo 15 y sus agentes de servidor virtual 35 asociados para realizar las operaciones de cambio en un servidor. En una realización, el conjunto de instrucciones 715 proporciona un contexto de transacción que se identifica como indicaciones de iniciar transacción y de finalizar transacción que encapsulan las operaciones de cambio en un objeto de servidor. Después de que ha realizado la indicación de iniciar transacción, el conjunto de instrucciones proporciona la información necesaria para realizar las operaciones de cambio solicitadas por los programas de aplicación 25.

El conjunto de instrucciones 715 proporciona también instrucciones de manejo de errores para los servidores de objetivo y sus agentes de servidor virtual asociados. En una realización, existen varios tipos de errores. Existen errores leves para alertar a los servidores de objetivo y a sus agentes de servidor virtual de una probabilidad de ocurrencia de un error durante operaciones de cambio en un servidor. Debido a que no se ha producido ningún error

real, el usuario 10 puede ignorar los errores leves y continuar con la ejecución de las operaciones de cambio en el servidor. De manera alternativa, el usuario 10 puede dar instrucciones a los agentes de servidor virtual para deshacer explícitamente los cambios realizados desde la ejecución de las operaciones de cambio en el servidor tras recibir la información de error devuelta por los errores leves.

- 5 Existen errores graves para notificar a los agentes de servidor virtual de una ocurrencia de un error durante la realización de operaciones de cambio en un servidor en los servidores de objetivo. En una realización, los errores graves pueden estar programados para desencadenar automáticamente operaciones de deshacer para deshacer cualquiera de los cambios realizados durante la ejecución de las operaciones de cambio en un servidor. En otra realización, los errores graves pueden estar programados para abortar la ejecución de lo que resta de operaciones de cambio de paquetes de transacción. Los errores graves son desencadenados por condiciones de error indicadas en el conjunto de instrucciones 715. Estas condiciones de error especifican que si se dan ciertas condiciones, los errores graves deben ser enviados a los servidores de objetivo y a sus agentes de servidor virtual asociados.

- 15 El conjunto de instrucciones 715 incluye asimismo información de prerequisites para las instrucciones. Un ejemplo de esta información de prerequisites puede incluir, pero no está limitada a, el conjunto mínimo de instrucciones de operación de cambio que deben ser especificadas en un paquete de transacción para su correcta ejecución. Por ejemplo, para añadir correctamente un componente COM+ en los servidores de objetivo, se deben especificar instrucciones para añadir la entrada COM+ en el catálogo, la entrada de registro correspondiente y el archivo DLL en el paquete de transacción. Otro ejemplo de la información de prerequisites puede incluir tipos de permisos necesarios para llevar a cabo las operaciones de cambio, el espacio en disco mínimo requerido por los servidores de objetivo 15 y el tipo de sistema operativo requerido. Además, la información de prerequisites puede incluir también instrucciones implícitas para objetos de servidor jerárquicos. Por ejemplo, para añadir un archivo en los servidores de objetivo, el directorio parental para el archivo debe existir en los servidores de objetivo, de manera que el archivo pueda ser creado bajo el directorio parental especificado en estos servidores.

- 25 En una realización, el conjunto de instrucciones 715 define los cambios que deben ser realizados en los objetos de servidor utilizando parámetros designados, y posteriormente remplazando en los parámetros los valores reales obtenidos de un archivo de parámetros 710. Los agentes de servidor virtual 35 reciben el paquete de transacción 700 en nombre de sus servidores de objetivo 15 asociados, y rempazan los parámetros designados con valores obtenidos del archivo de parámetros 710. Estos parámetros designados resultan particularmente útiles cuando se llevan a cabo operaciones de cambio en servidor en objetos de servidor que están dirigidos a múltiples servidores de objetivo 15, porque el parámetro designado que representa la identidad de cada servidor de objetivo puede ser remplazado con los identificadores de servidor reales por el agente de servidor virtual 35. Por ejemplo, parámetros designados de una instrucción pueden referenciar a un nombre de ruta para un servidor de objetivo 15 que incluye un nombre de anfitrión o una dirección de IP de servidor de objetivo 15. Estos parámetros son remplazados con identificadores de servidor reales para cada servidor de objetivo 15, según se proporciona en el archivo o los archivos de parámetros 710.

- 40 En una realización, el archivo de parámetros 710 puede ser un archivo de parámetros global o un archivo de parámetros específico para un anfitrión. Un archivo de parámetros global contiene parámetros que están configurados por el usuario 10, así el archivo de parámetros global idéntico se pasa a todos los servidores de objetivo 15. Un archivo de parámetros específico para un anfitrión contiene parámetros que son específicos para cada uno de los servidores de objetivo 15, así el archivo de parámetros específico para un anfitrión es diferente para cada uno de los servidores de objetivo 15. Los valores de parámetro contenidos en el archivo de parámetros global resultan útiles cuando se copia el mismo objeto de servidor de objetivo al mismo destino en múltiples servidores de objetivo 15. Ejemplos de este tipo de parámetro son el nombre y la clave del usuario. Para valores de parámetro contenidos en el archivo de parámetros específico para un anfitrión, los valores de parámetros son resueltos por cada uno de los servidores de objetivo 15. Ejemplos de estos parámetros son nombres de anfitrión y nombres de ruta de servidores de objetivo 15. Además, existen parámetros intrínsecos que son resueltos mediante variables de entorno de anfitrión en el servidor de objetivo. En una realización, uno o más archivos de parámetros 710 están asociados con uno o más servidores de objetivo. Por ejemplo, para un servidor de objetivo basado en Windows, "windir" y dirección de IP son ejemplos de variables de entorno de anfitrión que pueden ser utilizadas para resolver parámetros intrínsecos asociados con uno o más servidores de objetivo y pasados a través del paquete de transacción 700.

- 55 En referencia a las FIGS. 1 y 7, en una realización, en lugar de utilizar llamadas abstractas de sistema para llevar a cabo operaciones de cambio en un servidor generadas por los programas de aplicación 25, se puede utilizar un paquete de transacción 700 para llevar a cabo estas operaciones de cambio utilizando un conjunto de instrucciones 715 basado en XML. Para acomodar tanto las órdenes de nivel de llamada de sistema como los conjuntos de instrucciones de XML, cada agente de servidor virtual 35 es dividido en dos partes. Una parte del agente de servidor virtual 35 es un API de XML que puede interpretar el conjunto de instrucciones 715 de XML contenido en el paquete de transacción 700, y la otra parte del agente de servidor virtual 35 es un API de llamada de sistema que puede interpretar llamadas abstractas de sistema. Así, cuando un agente de servidor virtual 35 recibe un paquete de transacción 700 de XML a través del cliente de servidor virtual 30, el conjunto de instrucciones 715 de XML en el paquete de transacción 700 puede ser interpretado mediante el API de XML. En una realización alternativa, el paquete de transacción 700 puede ser implementado con un conjunto de instrucciones 715 de texto. Las órdenes del

conjunto de instrucciones 715 de texto son traducidas en llamadas abstractas de sistema que a su vez son interpretadas por el API de XML del sistema.

5 A continuación se muestra un ejemplo de un paquete de transacción de XML, denominado "Package_1.XML", que especifica un prerequisite, un contexto de transacción, un objeto de servidor compuesto, una secuencia e información de manejo de error utilizando un conjunto de instrucciones 715 de XML.

Package_1.XML

```
<blpackage schema-version="2.0" created-date="02/12/03" modified-
date="02/22/02" revision="23">
  <name>
    name of the blpackage
  </name>

  <description>
    description of the package
  </description>

  <source type="host">web-demol</source>

  <!-- default parameters -->
  <param name="$APP_PATH"> c:\program files\app </param>
  <param-file>foo.params</param-file>

  <applies-to>
    <condition>
      <os>"$(os) = Windows"</os>
      <os-version>$(os-version) > 5</os-version>
      <service-pack>2</service-pack>
    </condition>
  </applies-to>

  <!-- requires the following items before we deploy this package -->
  <depends>
    <condition>
      <application>SQL server</application>
      <version>$(version) = 8.0 </version>
    </condition>
  </depends>

  <!-- failure conditions if the following exit on target -->
  <FailIf>
    <ErrorLevel <4
```

```

/>
</FailIf >

<transaction id="0">
  <command id = "1005" undo="net start w3svc">net stop w3svc</command>
  <service action="add" refid="1003" key="RSCDsvc">
    <depends>
      <file refid="1002"/>
    </depends>
  </service>
  <command id = "1006" undo="net stop w3svc">net start w3svc </command>
  <file action="add" key="%WINDIR%ado.dll" refid="1001"/>
  <file action="add" key="%WINDIR%/System32/svchost.exe" refid="1002" />

  <assets>
    <file id="1001">
      <name>ado.dll</name>
      <source>0</source>
      <attributes>2</attributes>
      <created-date>02/12/03</created-date>
      <modified-date>02/22/03</modified-date>
      <owner></owner>
      <group>0</group>

      <acl key="%WINDIR%ado.dll" owner="BUILTIN\Administrators">
        <ace action="add" id="1313">web admins</ace>
        <acemode>0</acemode>
        <aceflags>3</aceflags>
        <acemask>1179817</acemask>

        <ace action="add" id="1314">dbas</ace>
        <acemode>1</acemode>
        <aceflags>3</aceflags>
        <acemask>2032127</acemask>

      </acl>
    </file>

    <file id="1002">
      <name>svchost.exe</name>
      <source>0</source>

```

```

<attributes>2</attributes>
<created-date>02/12/03</created-date>
<modified-date>02/22/03</modified-date>
<owner></owner>
<group>0</group>

<acl key="%WINDIR%ado.dll" owner="BUILTIN\Administrators">
  <ace action="add" id="1313">web admins</ace>
  <acemode>0</acemode>
  <aceflags>3</aceflags>
  <acemask>1179817</acemask>

  <ace action="add" id="1314">dbas</ace>
  <acemode>1</acemode>
  <aceflags>3</aceflags>
  <acemask>2032127</acemask>

</acl>
</file>

<service id="1003" name="RSCDsvc">
  <binary_path>%WINDIR%/System32/svchost.exe</binary_path>
  <name>RSCDsvc</name>
  <description></description>
  <state>Stopped</state>
  <runas>
    <userid>$Token1</userid>
    <pwd>$Token2</pwd>
  </runas>
</service>

</assets>
</transaction>
</blpackage>

```

El archivo de parámetros *foo.params* contiene

\$TOKEN1 como parámetro que corresponde al id de usuario - "R2D2\web-admins"

\$TOKEN2 como parámetro para la clave para R2D2\web-admins - "c3-po"

- 5 En este ejemplo, la etiqueta <blpackage schema> denota el inicio del conjunto de instrucciones 715. Las etiquetas <name>, <description> y <source type> respectivamente proporcionan el nombre de paquete, descripción y servidor de fuente, en este ejemplo el servidor "web-demo1", a partir del cual fue creado el paquete. La etiqueta <param> se utiliza para especificar ubicación, en este ejemplo "c:\program files\app", de parámetros que tienen el nombre de

“\$APP_PATH” en el paquete 700, mientras que la etiqueta <param file> se utiliza para especificar un archivo de parámetros externos 710 denominado “foo.params”. En la sección de prerequisites, que se introduce con la etiqueta <applies-to>, el sistema operativo MS Windows, versión posterior a 5 y con pack 2 de servicio, se especifica como un prerequisite para llevar a cabo este conjunto de instrucciones. También en la sección de prerequisites, la etiqueta <depends> indica que el servidor SQL, versión 8, es un prerequisite para el paquete. La información de manejo de error, que se introduce con la etiqueta <FailIf>, especifica que las operaciones de servidor deben fallar si el nivel de error cae por debajo de 4.

La etiqueta <transaction id="0"> introduce el conjunto de operaciones de cambio solicitadas, y cualquier información de dependencia para las operaciones de cambio en servidor especificado. La información de secuencia de ejecución para las operaciones de cambio en servidor se proporciona bajo la etiqueta <depends>. En este ejemplo, el orden de las operaciones, -detener w3svc, añadir servicio RSCDsvc, iniciar w3svc, añadir archivo ado.dll y añadir archivo svchost.exe, se produciría en el siguiente orden: detener w3svc, añadir archivo svchost.exe, añadir servicio RSCDsvc, iniciar w3svc y añadir archivo ado.dll.

Los equipos de servidor que están afectados por las operaciones de cambio en servidor se especifican bajo la etiqueta <assets>. Este ejemplo tiene tres equipos –dos archivos, id = 1001 e id = 1002, y un servicio, id = 1003. Cada archivo tiene una ACL de archivos anidada correspondiente que tiene las etiquetas <acl key>.

El archivo de parámetros 710, “foo.params” tiene dos parámetros que son utilizados en el paquete de transacción 700, denominado “\$TOKEN1” y “\$TOKEN2”. En lugar de pasar valores físicos dirigidos hacia cada servidor de objetivo, los parámetros designados son enviados, y son resueltos por el archivo de parámetro 710 cuando el archivo de parámetros 710 sustituye los valores reales que están específicos para cada servidor de objetivo 15 para los parámetros designados. Como se muestra en este ejemplo, estos valores pueden ser una ruta para una conexión de objetos de servidor (por ejemplo, archivos), un nombre de usuario o una clave. En este ejemplo, el primer parámetro, \$TOKEN1, corresponde al nombre de usuario “R2D2\web-admins”, y el archivo \$TOKEN2 corresponde a la clave “c3-po”.

En una realización, pueden agregarse múltiples paquetes de transacción en un proyecto de transacción 725. El proyecto de transacción 725 coordina los paquetes de transacción 700 y sus operaciones de cambio en servidor, de manera que cada operación de cambio en servidor pueda ser ejecutada en un modo de transacción segura. A continuación se muestra un ejemplo de un proyecto de transacción 725 de XML que contiene un paquete de transacción designado “BLPkg_web.XML”, dirigido a seis servidores web, un paquete de transacción designado “BLPkg_app.XML”, dirigido a dos servidores de aplicación y un paquete de transacción designado “BLPkg_db.XML”, dirigido a dos servidores de base de datos:

```
<PROJECT>
  <BLPkg>
    <Name>BLPkg_web.XML</Name>
    <Hosts>Web Server1</Hosts>
    <Hosts>Web Server2</Hosts>
    <Hosts>Web Server3</Hosts>
    <Hosts>Web Server4</Hosts>
    <Hosts>Web Server5</Hosts>
    <Hosts>Web Server6</Hosts>
  </BLPkg>

  <BLPkg>
    <Name>BLPkg_app.XML</Name>
    <Hosts>App Server1</Hosts>
    <Hosts>App Server2</Hosts>
  </BLPkg>

  <BLPkg>
    <Name>BLPkg_db.XML</Name>
    <Hosts>Db Server1</Hosts>
    <Hosts>Db Server2</Hosts>
  </BLPkg>
</PROJECT>
```

En este ejemplo, primero, el paquete “BLPkg_web.XML” va a ser ejecutado en seis servidores web designados Web Server1 a Web Server6, el paquete “BLPkg_app.XML” va a ser ejecutado en dos servidores de aplicación y el paquete “BLPkg_db.XML” va a ser ejecutado en dos servidores de base de datos.

5 El gestor de configuración 25B, o cualquiera de los programas de aplicación 25, prepara el paquete de transacción 700 y da instrucciones al cliente de servidor virtual 30 para pasar el paquete 700 a los agentes de servidor virtual 35 asociados con los servidores de objetivo. Tras recibir el paquete de transacción 700, los agentes de servidor virtual 35 desempaquetan el paquete 700 y ejecutan las operaciones en sus servidores de objetivo 15 asociados. En la FIG. 8 se muestra un método para conseguir esto.

10 En la etapa 800, el gestor de configuración 25B comprueba la información de prerequisites de las operaciones de cambio solicitadas. Ejemplos de la información de prerequisites incluyen comprobaciones relativas a la integridad y a la finalización de paquete tal como solicitar nombre de usuario y clave si es necesario, asegurándose que las dependencias simples son resueltas, y asegurándose de que los archivos correspondientes se encuentran en el paquete.

15 Después de que los prerequisites se han comprobado en la etapa 800, en la etapa 810, el gestor de configuración 25B comprueba la información de secuencia que indica el orden de ejecución de las operaciones de cambio solicitadas en el conjunto de instrucciones 715 del paquete. Si la información de secuencia no se encuentra en el conjunto de instrucciones 715, el gestor de configuración 25B accede al diagrama de dependencia externa 720 para obtener la información de secuencia. Tras la finalización de la etapa 810, en la etapa 815, el gestor de configuración 25B transfiere el paquete 700 y los archivos asociados y los archivos de parámetros a los agentes de servidor virtual 35 a través del cliente de servidor virtual 30.

25 En una realización, el agente de servidor virtual 35 recibe el paquete de transacción 700 completado a través del cliente de servidor virtual 30. En el agente de servidor virtual 35, en la etapa 820, los parámetros designados son sustituidos con valores reales. El agente de servidor virtual 35 ejecuta a continuación las operaciones de cambio en servidor especificadas en el paquete de transacción para su servidor de objetivo 15 asociado. En otra realización, en lugar de transportar el paquete de transacción 700 completado, el cliente de servidor virtual 30 puede transportar solo el archivo de parámetros 710 y el conjunto de instrucciones 715, sin los archivos reales o ninguno de los objetos de servidor, al agente de servidor virtual 35, en caso de que el usuario 10 opcionalmente elija continuar con un simulacro. El simulacro proporciona un conjunto adicional de pruebas para ver si el conjunto de instrucciones 715 puede ser llevado a cabo por el agente de servidor virtual 35 receptor antes de realizar ningún cambio en el servidor de objetivo 15. Después de que el agente de servidor virtual 35 recibe un paquete de transacción 700 parcial desde el cliente de servidor virtual 30 en la etapa 820, los parámetros son sustituidos con valores reales según se proporcionan en el archivo de parámetros 710. Después de la finalización del simulacro, el gestor de configuración 25B puede transferir todo el paquete 700 a los agentes de servidor virtual 35 a través del cliente de servidor virtual 30, para su ejecución real.

35 Antes de ejecutar las operaciones en cada servidor de objetivo 15, en la etapa 835, el agente actualiza un informe de deshacer. El informe de deshacer, que está guardado para cada servidor de objetivo, registra las operaciones ejecutadas, y hace un seguimiento a los cambios realizados mediante estas operaciones, de manera que si se produce un error mientras se están ejecutando las operaciones de cambio en servidores, las operaciones pueden ser deshechas durante las operaciones de cambio en servidor utilizando los registros del informe de deshacer. En una realización, el informe de deshacer es idéntico en estructura al paquete de transacción, pero con los archivos de parámetro dispuestos en orden inverso y las operaciones de cambio grabadas en orden inverso. Finalmente en la etapa 840, las operaciones de cambio en servidor son ejecutadas en los servidores de objetivo 15.

45 En referencia ahora a la FIG. 9, se describe un método para ejecutar y deshacer una operación de cambio en servidor en un modo de transacción segura. En la etapa 900, uno o más programas de aplicación 25 generan y especifican operaciones de cambio utilizando un paquete de transacción 700. Diferentes tipos de objetos de servidor y correspondientes servidores de objetivo 15 están soportados mediante el conjunto de instrucciones proporcionado en el paquete de transacción 700. A continuación, en la etapa 910, el programa de aplicación especifica al servidor o a los servidores de objetivo a cuál están dirigidas las operaciones de cambio en servidor. En la etapa 920, el programa de aplicación especifica el archivo de parámetros que proporciona parámetros y sus correspondientes valores definidos para cada uno de los servidores de objetivo, y pone esta información en el paquete de transacción 700. En la etapa 930, el cliente de servidor virtual 30 envía la operación de cambio en servidor desde el programa de aplicación 25 a los agentes de servidor virtual 35 en los servidores de objetivo 15. En la etapa 940, los servidores de objetivo 15 ejecutan las operaciones de cambio en servidor en un modo de transacción segura.

Gestor de configuración

55 En referencia ahora a la FIG. 10, el gestor de configuración 25B es un programa de aplicación 25 de ejemplo que hace un seguimiento de cambios y cumplimiento y configura servidores de objetivo generando y desplegando un paquete de transacción 700. El gestor de configuración 25B proporciona un método y sistema para configurar diferentes servidores utilizando una variedad de módulos de software, tales como un examinador 1000, una plantilla 1010, un grabador 1020, un modelo de referencia 1030, un comparador 1040 y un corrector 1050.

El examinador 1000 examina objetos de servidor en diferentes servidores en tiempo real, para examinar la configuración actual de los objetos de servidor contenidos en los servidores 15. Primero, el usuario selecciona un servidor que desea examinar. Mediante el examen, una colección de identificadores de objeto de servidor que identifican cada objeto de servidor son seleccionados e introducidos en la plantilla 1010. De manera alternativa, en lugar de construir una plantilla 1010 a partir del examen, la plantilla 1010 puede ser importada de un proveedor externo. La plantilla 1010 puede ser asimismo creada incluyendo una o más plantillas previamente definidas. En una realización, la plantilla 1010 es una plantilla abstracta que identifica objetos de servidor contenidos en un servidor. Por ejemplo, si un servidor Apache contiene archivos, y entradas en el archivo de configuración, una plantilla 1010 de servidor Apache contiene identificadores que son suficientes para identificar los archivos y las entradas de los archivos de configuración del servidor Apache. Tras identificar los objetos de servidor en la plantilla 1010, los valores de estos objetos de servidor identificados son grabados para configurar servidores en la red.

En una realización, el grabador 1020 toma una instantánea de valores (por ejemplo, atributos) asociados con una colección de objetos de servidor. En otra realización, el grabador 1020 toma una instantánea de valores de los objetos de servidor identificados en la plantilla 1010. Los valores pueden proceder de cualquier servidor examinado por el examinador. De manera alternativa, los valores pueden proceder de un servidor seleccionado, denominado también servidor gold. Ejemplos de los valores (o atributos) de los archivos grabados en las instantáneas incluyen, pero no están limitados a, nombres de archivo, tamaños, permisos, propietarios, fechas de creación, fechas de modificación y versiones. Ejemplos de atributos (o valores) de directorio grabados en instantáneas son ubicaciones de directorios, permisos, fechas de creación y fechas de modificación. Ejemplos de atributos de entradas de registro grabados en instantáneas son nombres de archivo y valores correspondientes.

En una realización, los valores grabados o los resultados de las instantáneas de servidor gold son utilizados para obtener valores básicos, y rangos de cumplimiento a partir de los mismos en el modelo de referencia 1030. En otra realización, en lugar de crear el modelo de referencia, los resultados de la instantánea pueden ser utilizados directamente para hacer un seguimiento a los cambios, configurar servidores existentes y proveer nuevos servidores en la red. Los resultados de la instantánea registran una configuración de un servidor en un momento dado, y así no pueden modificarse. No obstante, el modelo de referencia 1030 puede ser editado para representar la implementación de referencia para cumplimiento o aprovisionamiento.

Por ejemplo, cuando las instantáneas del servidor gold son tomadas por el grabador 1020, los valores recogidos en las instantáneas se guardan en el modelo de referencia 1030. Basándose en los valores del servidor gold, el modelo de referencia 1030 puede proporcionar información, tal como valores de línea básica y rangos de cumplimiento, para su uso por otros servidores en la red para identificar su desviación en comparación con el servidor gold. Los valores de línea básica proporcionan una base para la configuración de otros servidores. Los rangos de cumplimiento son rangos de valores de configuración aceptables que son aceptables para otros servidores para que estos servidores cumplan. Como alternativa a crear un modelo de referencia 1030, el modelo de referencia 1030 puede ser un modelo de referencia importado que fue creado por un proveedor externo. También, el modelo de referencia 1030 puede incluir uno o más modelos de referencia predefinidos. A continuación, el comparador 1040 compara un servidor con el modelo de referencia 1030 para hacer un seguimiento de los cambios y hacer un seguimiento del cumplimiento en el servidor.

En otro ejemplo, una instantánea de una configuración actual de un servidor capturada en un momento arbitrario puede ser comparada con una versión en directo del servidor capturado para rastrear los cambios en el servidor capturado. La configuración de un servidor puede incluir objetos de servidor seleccionados explícitamente que están en el servidor u objetos de servidor seleccionados implícitamente que están en el servidor u objetos de servidor seleccionados implícitamente a través de la plantilla 1010.

En otro ejemplo más, los resultados de la instantánea de instantáneas periódica de un servidor tomadas a intervalos de tiempo programados (por ejemplo, diariamente, semanalmente, etc.) pueden ser utilizados para hacer un seguimiento de los cambios en el servidor capturado. En este ejemplo, la primera instantánea de servidor sirve como base, de manera que para siguientes instantáneas, solo se guardan los cambios con respecto a la base en los resultados de la instantánea. Así, todos los resultados de instantáneas tomadas durante estos intervalos de tiempo pueden ser reconstruidos para ver su configuración y contenido completos combinando la base con los cambios incrementales guardados en el resultado de la instantánea. Además, los cambios incrementales muestran cambios ocurridos en la configuración de servidor en un periodo de tiempo para que el usuario analice los cambios de este servidor particular.

A continuación, el comparador 1040 compara la versión en directo de servidor con la instantánea de la referencia para hacer un seguimiento y guardar solo cambios en el servidor.

En una realización, dos servidores en directo pueden ser comparados entre sí sin las instantáneas o el modelo de referencia 1030, de una manera específica. En esta realización, el usuario 10 puede seleccionar explícitamente los objetos de servidor que están compartidos comúnmente entre los dos servidores en directo, de manera que el comparador 1040 pueda comparar los valores de los objetos de servidor entre estos servidores. En otro ejemplo de esta realización, el comparador 1040 compara los valores de los objetos de servidor que son proporcionados implícitamente mediante la plantilla 1010.

Tras comparar los servidores e identificar las discrepancias presentes en los servidores comparados, el corrector 1050 corrige las discrepancias en cada servidor de objetivo. El corrector 1050 examina las discrepancias y genera operaciones de cambio en servidor que solicitan servicios de los sistemas operativos que se ejecutan en los servidores de objetivo para corregir estas discrepancias. Como se ha descrito previamente, las operaciones de cambio en servidor pueden ser presentadas a los servidores como un paquete de transacción 700 para eliminar las discrepancias y sincronizar los servidores de objetivo con el modelo de referencia 1030 en un modo de transacción segura. De manera similar, en una realización, las actualizaciones de configuración a los servidores de objetivo pueden ser realizadas mediante el paquete de transacción 700. En particular, el gestor de configuración 25B realiza primero todas las actualizaciones al modelo de referencia 1030, que a continuación empaqueta las discrepancias (introducidas en el modelo de referencia) como configuraciones en el paquete de transacción 700. El paquete de transacción 700 es propagado a los servidores de objetivo para sincronizarlos con el modelo de referencia 1030 actualizado.

El modelo de referencia 1030 puede ser también utilizado para proveer un nuevo servidor para asegurar la coherencia en la configuración de los servidores en la red cuando se añade un nuevo servidor. Por ejemplo, un modelo de referencia 1030 Apache puede ser utilizado para proveer un nuevo servidor Apache de manera que la configuración de todos los servidores Apache de la red sea coherente entre sí.

Además, tanto el modelo de referencia 1030 como las instantáneas pueden ser utilizados para restaurar una configuración previa de un servidor en caso de recuperación de un desastre. En particular, en caso de fallo de un servidor, este servidor puede recuperar su configuración más reciente y su contenido reconstruyendo la configuración del servidor a partir de las instantáneas tomadas en un periodo de tiempo. Con el modelo de referencia 1030, en caso de fallo de un servidor, el servidor puede mirar los valores básicos del servidor gold en el modelo de referencia 1030 y sincronizarse a esta configuración para cumplir de nuevo.

La FIG. 11 muestra un método de ejemplo de seguimiento de cambios y de cumplimiento, y componentes de corrección, así como cambios a nivel de parámetro en múltiples servidores. En la etapa 1100, el gestor de configuración 25B examina los servidores de la red para obtener equipos de servidor e información de estado de configuración (denominados en conjunto objetos de servidor) para cada servidor. En la etapa de examen 1100, los objetos de servidor seleccionados y sus objetos de servidor dependientes son examinados en tiempo real. En una realización, los servidores en directo en la red y sus objetos de servidores almacenados pueden ser examinados por medio de la Interfaz gráfica de usuario (GUI – Graphic User Interface, en inglés) que presenta el servidor y los objetos de servidor jerárquicamente.

A continuación, en la etapa 1105, el gestor de configuración 25B selecciona identificadores de los objetos de servidor examinados para que estén en la plantilla 1010. Los identificadores pueden incluir cualquier información acerca del objeto de servidor que sea suficiente para identificar el objeto de servidor. A continuación en la etapa 1110, el gestor de configuración selecciona un servidor gold, para proporcionar una configuración de línea básica y rangos de configuración para otros servidores de la red. En la etapa 1115, instantáneas de los valores de los objetos de servidor identificados en la plantilla que existen en el servidor gold son grabadas en el modelo de referencia 1030. Sobre la base de los valores grabados en el modelo de referencia 1030, en la etapa 1115, el modelo de referencia establece las reglas de cumplimiento, tal como la configuración básica y los rangos de cumplimiento. De manera alternativa, las instantáneas de los valores no son grabados en el modelo de referencia. Por el contrario, los resultados de la instantánea de un servidor pueden ser utilizados para comparar directamente con una versión en directo de este servidor para hacer un seguimiento de los cambios.

En la etapa 1120, el gestor de configuración 25B selecciona servidores y sus respectivos parámetros de configuración (denominados también objetos de servidor) para comparar contra el modelo de referencia 1030. Estos servidores pueden ser seleccionados de cualquier servidor en directo de la red. De manera alternativa, estos servidores de la versión en directo pueden ser también comparados directamente con sus propias instantáneas, tomadas en un momento arbitrario, o tomadas en un periodo específico, sin el modelo de referencia 1030, para hacer un seguimiento del cumplimiento y los cambios en estos servidores. Los resultados de la etapa de comparación 1125 pueden ser visualizados elemento a elemento, mostrando qué software (u objetos de servidor) están instalados o no instalados, o anfitrión a anfitrión, mostrando cada servidor y los objetos de servidor presentes en el servidor.

Finalmente, basándose en las discrepancias obtenidas durante la etapa de comparación 1120, una etapa de corrección 1130 repara los servidores para que cumplan, sincronizando la configuración de estos servidores con el modelo de referencia 1030 o las instantáneas. Además, un servidor recién añadido puede ser aprovisionado para que sea coherente con otros servidores sincronizando este nuevo servidor con el modelo de referencia 1030.

En referencia a la FIG. 12, en una realización, el gestor de configuración 25B puede gestionar el mismo tipo de parámetros de configuración (denominados también objetos de servidor) en diferentes servidores especificando una o más categorías para los parámetros en plantillas. La plantilla 1200 primero especifica la categoría “tipo de servidor” (por ejemplo, categoría de servidor de aplicaciones 1210, categoría de servidor web 1215 y categoría de servidor de base de datos 1220) para especificar a qué tipo de servidor pertenece cada objeto de servidor de la red, y a continuación especifica la categoría “tipo de parámetro” (por ejemplo, parámetros de red, parámetros de capacidad,

parámetros de disponibilidad, parámetros de funcionamiento, parámetros de seguridad) para especificar el tipo de parámetro al cual pertenece cada objeto de servidor. Cada objeto de servidor de la plantilla 1200 puede estar clasificado en una o más categorías, subcategorías y claves. En un ejemplo, para los parámetros de seguridad, las subcategorías pueden incluir tipo de codificación y tipo de autenticación, y las claves pueden incluir "solo lectura" y constante.

En referencia brevemente a la FIG. 13, se muestra un ejemplo del sistema descrito con referencia a la FIG. 12. En este ejemplo, Internet 1300 e intranet 1305 están disponibles para diferentes categorías de servidores 1215, 1210, 1220 a través de cortafuegos 1310. La categoría de servidor web 1215 incluye un servidor IIS 1215A para servicios de intranet y servidores Apache 1215B, 1215C para servicios de internet de HTTP / FTP e inalámbricos / de video, respectivamente. La categoría de servidor de aplicación 1210 incluye servidores que ejecutan aplicaciones de venta 1210A, aplicaciones de corretaje en línea 1210B y aplicación de servicio al cliente 1210C. La categoría de servidor de base de datos 1220 incluye bases de datos de ventas, comercio y cuentas 1220A, 1220B y 1220C.

En referencia de nuevo a la FIG. 12, cada objeto de servidor de la plantilla 1200 está situado en una categoría de parámetro basada en su función y en el tipo de servidor. Por ejemplo, los objetos de servidor pueden estar agrupados en parámetros de red 1330, parámetros de capacidad 1335, parámetros de disponibilidad 1340, parámetros de funcionamiento 1345 y parámetros de seguridad 1350. El gestor de configuración 25B selecciona objetos de servidor relacionados por categorías de cada categoría de servidores, y los almacena en la plantilla 1200. Por ejemplo, todos los parámetros de seguridad en la categoría de servidor de aplicación 1210 y todos los parámetros de red en la categoría de servidor de aplicación 1210 están almacenados en la plantilla 1200.

En referencia de nuevo a la FIG. 13, para la categoría de servidor web 1215, los parámetros de configuración de servidor web a, b, c, d, e están respectivamente clasificados en categorías como parámetros de red 1330, parámetros de capacidad 1335, parámetros de disponibilidad 1340, parámetros de funcionamiento 1345 y parámetros de seguridad 1350. Para la categoría de servidor de aplicación 1210, los parámetros de configuración de servidor de aplicación i, ii, iii, iv, v están respectivamente clasificados en categorías como parámetros de red 1330, parámetros de capacidad 1335, parámetros de disponibilidad 1340, parámetros de funcionamiento 1345 y parámetros de seguridad 1350. De manera similar, para la categoría de servidor de base de datos 1220, los parámetros de configuración de servidor de base de datos I, II, III, IV, V están respectivamente clasificados en categorías como parámetros de red 1330, parámetros de capacidad 1335, parámetros de disponibilidad 1340, parámetros de funcionamiento 1345 y parámetros de seguridad 1350.

Tras clasificar en categorías todos los objetos de servidor en la plantilla 1200 mediante las categorías de tipo de servidor y las categorías de tipo de parámetro, puede obtenerse una nueva plantilla a partir de la plantilla 1200 para aislar los objetos de servidor relacionados en categorías en las categorías de servidor y gestionar los parámetros de configuración como si perteneciesen a un solo servidor. Por ejemplo, los parámetros de configuración de seguridad de un servidor web individual pueden ser cambiados de acuerdo con otros parámetros de configuración de seguridad para otros servidores web, así como para servidores de aplicación y servidores de base de datos. En el ejemplo mostrado en la FIG. 13, por ejemplo, el parámetro a de red de servidor web puede ser cambiado de acuerdo con los parámetros de red i de la categoría de servidor de aplicación 1210 y el parámetro I de la categoría de servidor de base de datos 1220. De manera similar, el parámetro b de capacidad de servidor web puede ser cambiado de acuerdo con otros parámetros de capacidad ii de la categoría de servidor de aplicación 1210 y II de la categoría de servidor de base de datos 1220. Asimismo, pueden realizarse cambios sucesivos de parámetros para los parámetros de disponibilidad 1346, los parámetros de funcionamiento 1345 y los parámetros de seguridad 1350.

En referencia a la FIG. 14, una instantánea de ejemplo de un gestor de configuración 25B de GUI incluye un módulo denominado examinador de equipos 1400, que permite a un usuario 10 examinar servidores de objetivo 15 remotos en directo, y gestionar y almacenar equipos de servidor frecuentemente utilizados (denominados también objetos de servidor). El examinador 1400 de equipos se divide en dos paneles. El panel izquierdo 1410 funciona como cualquiera de un panel de servidores o un panel de almacenes, dependiendo de una etiqueta 1420 seleccionada por el usuario 10. El panel de contenidos 1430 en el lado derecho muestra el contenido de un elemento seleccionado en el panel de servidores o de almacenes.

En la FIG. 14, el panel izquierdo 1420 muestra el panel de servidores que muestra una representación jerarquizada de los servidores que el usuario 10 gestiona. Por ejemplo, el usuario 10 puede disponer los servidores en grupos basados en la ubicación geográfica y/o en el sistema operativo. Los grupos de servidores están divididos en las divisiones este y oeste de una empresa, y dentro de esos grupos, otro nivel de jerarquía para servidores de Windows, UNIX y Linux. Más específicamente en la FIG. 14, en los servidores de la división este 1440, se selecciona el objeto de parches 1460 en el servidor sun 2 1450. El panel de contenido 1430 muestra el contenido del objeto de parches 1460.

El panel de almacenes (no mostrado) puede mostrar almacenes centrales de objetos de servidor a los que se accede de manera común (por ejemplo, todos los archivos, software para ser desplegados, y apuntadores al contenido de los archivos y al software residente en otros servidores de la red). Además, el panel de almacenes almacena tareas programadas para ser llevadas a cabo, instantáneas de objetos de servidor, programaciones de intérprete de órdenes y paquetes de transacción 700.

Ejemplo

En un ejemplo global de operación del gestor de configuración, el gestor de configuración examina servidores en directo en una red, rastrea cambios y cumplimiento en los servidores comparando sus objetos de servidor con un modelo de referencia o una instantánea e identificando todas las discrepancias con respecto al modelo de referencia o la instantánea. Realizando registros de los valores del objeto de servidor de servidor gold mediante una instantánea y guardando los resultados como un modelo de referencia, el modelo de referencia puede ser utilizado para auditar a otros servidores, para determinar cómo pueden cambiar las configuraciones de los otros servidores a partir del modelo de referencia. De manera alternativa, una instantánea del propio servidor puede ser tomada arbitrariamente, o en un periodo de tiempo específico para rastrear cambios en el servidor, sin utilizar el modelo de referencia. En un ejemplo, los objetos de servidor que se están comparando en el proceso de auditoría se proporcionan automáticamente mediante el gestor de configuración por medio de plantillas. En otro ejemplo, el usuario puede seleccionar manualmente los objetos de servidor para comparar. Adicionalmente, el proceso de auditoría puede ser programado para hacer un seguimiento del cumplimiento en el tiempo.

Tras identificar las discrepancias de configuración de servidor presentes en los servidores, el gestor de configuración 25B corrige las discrepancias generando un paquete de transacción 700, que contiene operaciones de cambio en servidor para ser llevadas a cabo en los servidores 15. El paquete de transacción 700 agrupa operaciones de cambio en servidor e instrucciones correspondientes para ser desplegadas en servidores de objetivo 15 remotos para corregir todas las discrepancias que existen en los objetos de servidor contenidos en esos servidores 15. Con el paquete de transacción 700, el gestor de configuración 25B puede instalar cualquier tipo de objeto de servidor desde una sola fuente a múltiples ubicaciones. De manera similar, el gestor de configuración 25B puede desinstalar software, y deshacer despliegues de objeto de servidor en los servidores de objetivo 15 remotos. Como se ha explicado previamente, ciertos valores del paquete de transacción 700 pueden ser parametrizados y sucesivamente remplazados con valores reales durante el despliegue del paquete de transacción 700 en los servidores de objetivo 15, sin cambiar el contenido del paquete de transacción 700 para cada servidor de objetivo 15.

En un ejemplo particular, el gestor de configuración 25B puede ser utilizado para mover una base de datos de servidor MS SQL de trabajo desde un servidor gold a múltiples servidores de objetivo 15, para duplicar los cambios realizados en esta base de datos a múltiples servidores. Para conseguir esta duplicación, el usuario 10 copia los cambios realizados en la base de datos de servidor SQL al modelo de referencia, de manera que el gestor de configuración 25B puede posteriormente agrupar estos cambios a otras instancias de la misma base de datos de servidor SQL en los servidores de objetivo 15 remotos. El modelo de referencia y los servidores de objetivo 15 remotos tienen la misma instalación inicial de la base de datos del servidor SQL. El gestor de configuración toma una instantánea del servidor gold para crear un modelo de referencia que es utilizado como base para comparar las bases de datos de servidor SQL entre el servidor gold y los servidores de objetivo 15. Los cambios de base de datos necesarios son realizados primero en el servidor gold. A continuación, el gestor de configuración 25B crea un paquete de transacción 700 para agrupar estos cambios para ser desplegados en los agentes de servidor virtual 35 asociados con los servidores de objetivo 15 para solicitar que estos cambios sean realizados en sus bases de datos de servidor SQL.

En algunas realizaciones, la funcionalidad de los sistemas y métodos descritos anteriormente puede ser implementada como software en uno o más ordenadores de propósito general. En tal realización, el software puede estar escrito en cualquiera de un número de lenguajes de alto nivel, tales como FORTRAN, PASCAL, C, C++, LISP, JAVA o BASIC. Además, el software puede estar escrito en una escritura, macro o funcionalidad insertada en software disponible comercialmente, tal como EXCEL o VISUAL BASIC. Adicionalmente, el software podría estar implementado en un lenguaje de ensamblador dirigido a un microprocesador residente en un ordenador. Por ejemplo, el software podría estar implementado en lenguaje de ensamblador 80 x 86 de Intel si estuviese configurado para ser ejecutado en un PC de IMB o en un PC clon. El software puede estar incorporado en un artículo de fabricación que incluye, pero no está limitado a, un "medio legible por ordenador" tal como un disco flexible, un disco grave, un disco óptico, una cinta magnética, una PROM, una EPROM o un CD-RM.

Variaciones, modificaciones y otras implementaciones de lo que se ha descrito en esta memoria se les ocurrirán a personas no expertas en la materia, sin separarse del espíritu y el alcance de la invención según se reivindica. De acuerdo con esto, la invención debe estar definida no por la descripción ilustrativa precedente, sino por el contrario, por el espíritu y alcance de las siguientes reivindicaciones.

Lo que se reivindica es:

REIVINDICACIONES

1. Un método para configurar una pluralidad de servidores heterogéneos en una red, estando el método caracterizado por:
 - 5 (a) examinar (1000, 1100), en respuesta a una entrada del usuario, objetos de servidor en uno o más de una pluralidad de servidores en una red para examinar la configuración actual de los objetos de servidor en los servidores, donde los objetos de servidor comprenden uno o más equipos de servidor, parámetros de configuración o ambos;
 - (b) crear una plantilla (1010) seleccionando identificadores (1105) de uno o más de los objetos de servidor examinados;
 - 10 (c) crear un modelo de referencia (1030):
 - (d) seleccionando un servidor de referencia (1110) de la pluralidad de servidores;
 - (e) grabar (1020) valores de los objetos de servidor en el servidor de referencia (1110) seleccionado para los objetos de servidor especificados por los identificadores de la plantilla (1010);
 - 15 (f) proporcionar valores para objetos de servidor de base para comparación con otros de la pluralidad de servidores, incluyendo el citado modelo de referencia (1030) valores de línea básica y rangos de cumplimiento para proporcionar una base para la configuración de otros servidores;
 - (g) comparar (1040, 1120) los respectivos parámetros de configuración de un segundo servidor de la pluralidad de servidores con el modelo de referencia (1030);
 - 20 (h) identificar discrepancias entre los objetos de servidor del segundo servidor y los objetos de servidor de la línea básica del modelo de referencia, en el que todas las discrepancias identificadas para el al menos un objeto de servidor básico están fuera de los rangos de cumplimiento;
 - (i) corregir (1050, 1125) las discrepancias identificadas de los objetos de servidor del segundo servidor con respecto al modelo de referencia (1030) generando operaciones de cambio en servidor.
- 25 2. El método de la reivindicación 1, en el que los objetos de servidor comprenden al menos uno de un archivo y de una entrada de archivo de configuración.
3. El método de la reivindicación 2, en el que los objetos de servidor comprenden al menos uno de un objeto de servidor primitivo, un objeto de servidor compuesto, un objeto de servidor de configuración abstracta y un objeto de servidor de componentes.
- 30 4. El método de la reivindicación 3, en el que cada objeto de servidor primitivo es un objeto de servidor elemental seleccionado de un conjunto de objetos de servidor elementales que comprende: archivos, directorios, listas de control de acceso (ACL), programas de aplicación, parches de resolución de problemas, paquetes, archivos de gestor de paquetes, parches, entradas de registro, entradas de catálogo de modelo de objeto de componentes (COM / COM+), entradas de metabases, usuarios, cuentas y entradas de archivo de configuración.
- 35 5. El método de la reivindicación 3, en el que el objeto de servidor compuesto comprende una pluralidad de objetos de servidor, cada uno de la pluralidad de objetos de servidor en el objeto de servidor compuesto seleccionado del conjunto de objetos de servidor primitivo, objetos de servidor de configuración abstracta, objetos de servidor de componentes y objetos de servidor compuesto.
- 40 6. El método de la reivindicación 3, en el que el objeto de servidor de configuración abstracta representa una entrada en un archivo de configuración cuando el archivo de configuración es mapeado a un formato de archivo de configuración abstracta común.
7. El método de la reivindicación 3, en el que el objeto de servidor de componentes comprende una colección secuenciada de objetos de servidor.
8. El método de la reivindicación 1, en el que la plantilla se crea manualmente.
9. El método de la reivindicación 1, en el que la plantilla se crea externamente y es importada.
- 45 10. El método de la reivindicación 1, en el que la plantilla comprende al menos una plantilla definida previamente.
- 50 11. El método de la reivindicación 1, que comprende además repetir la etapa de grabación (d), tomando con ello una pluralidad de instantáneas de los valores para los objetos de servidor básicos para el servidor de referencia seleccionado a intervalos de tiempo predeterminados y capturando una versión más reciente de una línea básica con cada instantánea.

12. El método de la reivindicación 11, que comprende además repetir la etapa de comparación (e), auditando con ello un servidor en directo en diferentes momentos para hacer un seguimiento del cumplimiento con respecto a la línea básica en el tiempo.
- 5 13. El método de la reivindicación 1, en el que el servidor de referencia seleccionado es un servidor en directo, y el modelo de referencia es un modelo de referencia en directo que comprende valores obtenidos de servidor en directo para los objetos de servidor especificados en la plantilla.
14. El método de la reivindicación 1, en el que el segundo servidor no tiene inicialmente ningún objeto de servidor identificado en él, y el segundo servidor está aprovisionado mediante la corrección de las discrepancias con respecto al modelo de referencia.
- 10 15. El método de la reivindicación 1, en el que la etapa de corrección (1050, 1125) (f) comprende además:
recoger las discrepancias identificadas y generar un conjunto de instrucciones (715) para ejecutar una pluralidad de operaciones de cambio de servidor en el segundo servidor; y
sincronizar el segundo servidor con el modelo de referencia (1030).
- 15 16. Un sistema para configurar una pluralidad de servidores heterogéneos en una red, estando el sistema caracterizado por:
(a) un examinador (1000) para examinar, en respuesta a una entrada del usuario, objetos de servidor en uno o más de una pluralidad de servidores en una red para examinar la configuración actual de los objetos de servidor en los servidores, donde los objetos de servidor comprenden uno o más equipos de servidor, parámetros de configuración o ambos;
- 20 (b) una plantilla (1010) que comprende identificadores de uno o más de los objetos de servidor examinados;
(c) un grabador (1020) para grabar valores de los objetos de servidor examinados especificados por los identificadores en la plantilla de un servidor de referencia seleccionado de una pluralidad de servidores, siendo el servidor de referencia seleccionado proporcionando objetos de servidor de línea básica para su comparación con otros de la pluralidad de servidores;
- 25 (d) un servidor de referencia (1030) que comprende los valores grabados para proporcionar valores para objetos de servidor de línea básica y rangos de cumplimiento para proporcionar una base para la configuración de otros servidores;
- 30 (e) un comparador (1040), en comunicación con el modelo de referencia, para comparar los parámetros de configuración respectivos de un segundo servidor de una pluralidad de servidores con el modelo de referencia que tiene los valores grabados e identificar todas las discrepancias entre los objetos de servidor del segundo servidor y los objetos de servidor de la línea básica del modelo de referencia, en el que todas las discrepancias identificadas para el al menos un objeto de servidor básico están fuera de los rangos de cumplimiento; y
- 35 (f) un corrector (1050), en comunicación con el comparador, para generar operaciones de cambio en servidor para corregir las discrepancias de los objetos de servidor del segundo servidor con respecto a los objetos de servidor de la línea básica del modelo de referencia.
17. El sistema de la reivindicación 16, en el que los objetos de servidor comprenden al menos uno de un archivo y una entrada de archivo de configuración.
- 40 18. El sistema de la reivindicación 17, en el que los objetos de servidor son cada uno un objeto de servidor seleccionado del conjunto de objetos de servidor que comprende un objeto de servidor primitivo, un objeto de servidor compuesto, un objeto de servidor de configuración abstracta y un objeto de servidor de componentes.
- 45 19. El sistema de la reivindicación 18, en el que cada objeto de servidor primitivo es un objeto de servidor elemental seleccionado de un conjunto de objetos de servidor elementales que comprende: archivos, directorios, listas de control de acceso (ACL), algoritmos de aplicación, parches de resolución de problemas, paquetes, archivos de gestor de paquetes, parches, entradas de registro, entradas de catálogo de modelo de objeto de componentes (COM / COM+), entradas de metabase, usuarios, cuentas y entradas de archivo de configuración.
20. El sistema de la reivindicación 18, en el que el objeto de servidor compuesto comprende una pluralidad de objetos de servidor, estando cada uno de la pluralidad de objetos de servidor del parámetros de servidor compuesto seleccionado de un conjunto de objetos de servidor primitivo, objetos de servidor de configuración abstracta, objetos de servidor de componentes y objetos de servidor compuesto.
- 50 21. El sistema de la reivindicación 18, en el que el objeto de servidor de configuración abstracta representa una entrada en un archivo de configuración cuando el archivo de configuración es mapeado a un formato de archivo de configuración abstracta común.

22. El sistema de la reivindicación 18, en el que el objeto de servidor de componentes comprende una colección secuenciada de objetos de servidor.
23. El sistema de la reivindicación 16, en el que la plantilla (1010) comprende una plantilla creada manualmente.
- 5 24. El sistema de la reivindicación 16, en el que la plantilla (1010) comprende una plantilla importada externamente.
25. El sistema de la reivindicación 16, en el que la plantilla (1010) comprende al menos una plantilla definida previamente.
- 10 26. El sistema de la reivindicación 16, en el que el grabador (1020) toma repetidamente una pluralidad de instantáneas (1115) de los valores para objetos de servidor de línea básica para el servidor de referencia seleccionado a intervalos de tiempo predeterminados, capturando con ello una versión más reciente de una línea básica con cada instantánea.
27. El sistema de la reivindicación 26, en el que el comparador (1040) toma repetidamente una pluralidad de auditorías de un servidor en directo en diferentes momentos para hacer un seguimiento del cumplimiento frente a la línea básica a lo largo del tiempo.
- 15 28. El sistema de la reivindicación 16, en el que el servidor de referencia seleccionado es un servidor en directo y el modelo de referencia es un modelo de referencia en directo que comprende valores obtenidos del servidor en directo para los objetos de servidor especificados en la plantilla.
29. El sistema de la reivindicación 16, en el que el segundo servidor no tiene inicialmente ningún objeto de servidor identificado en el modelo de referencia y el segundo servidor es por ello aprovisionado mediante la corrección de discrepancias con respecto al modelo de referencia.
- 20 30. El sistema de la reivindicación 16, en el que el corrector utiliza las discrepancias identificadas para generar un conjunto de instrucciones (715) para ejecutar una pluralidad de operaciones de cambio en servidor en el segundo servidor y sincroniza el segundo servidor con el modelo de referencia.
- 25 31. Un producto de programa informático que comprende un medio utilizable por ordenador que incluye lógica de control almacenada en el mismo, permitiendo la lógica de control la configuración de una pluralidad de servidores heterogéneos en una red, comprendiendo el producto de programa de ordenador:
 - (a) medios de examen (1000) para examinar, en respuesta a una entrada de usuario, objetos de servidor en uno o más de una pluralidad de servidores en una red para examinar la configuración actual de los objetos de servidor en los servidores, comprendiendo los objetos de servidor uno o más equipos de servidor, parámetros de configuración, o ambos;
 - 30 (b) medios de creación para crear una plantilla (1010) seleccionando identificadores (1105) de uno o más de los objetos de servidor examinados;
 - (c) medios de selección para seleccionar un servidor de referencia (1110) de una pluralidad de servidores, seleccionándose el servidor de referencia seleccionado para proporcionar objetos de servidor de la línea básica para su comparación con otros de la pluralidad de servidores;
 - 35 (d) medios de grabación (1020) para grabar valores para los objetos de servidor especificados por los identificadores en la plantilla (1010) de los objetos de servidor en el servidor de referencia seleccionado (1110) para crear un modelo de referencia (1030);
 - 40 (e) medios de comparación (1040) para comparar los parámetros de configuración respectivos de un segundo servidor de la pluralidad de servidores con el modelo de referencia que tiene valores grabados del servidor de referencia seleccionado e identificar todas las discrepancias entre los objetos de servidor del segundo servidor y los objetos de servidor de la línea básica del modelo de referencia, en el que todas las discrepancias para el al menos un objeto de servidor de la línea básica están fuera de los rangos de cumplimiento; y
 - 45 (f) medios (1050) de corrección para generar operaciones de cambio en servidor para corregir las discrepancias de los objetos de servidor del segundo servidor con respecto a los objetos de servidor de la línea básica del modelo de referencia.
32. El producto de programa informático de la reivindicación 31, en el que uno o más de los objetos de servidor son un archivo o una entrada de archivo de configuración.
- 50 33. El producto de programa informático de la reivindicación 31, en el que cada uno de los objetos de servidor son uno de: un objeto de servidor primitivo, un objeto de servidor compuesto, un objeto de servidor de configuración abstracta o un objeto de servidor de componentes.

- 5 34. El producto de programa informático de la reivindicación 33, en el que cada objeto de servidor primitivo comprende uno de los siguientes objetos de servidor elementales: archivos, directorios, listas de control de acceso (ACL), programas de aplicación, parches para resolución de problemas, paquetes, archivos de gestión de paquetes, parches, entradas de registro, entradas de catálogo de modelo de objeto de componente (COM / COM+), entradas de metabase, usuarios, cuentas y entradas de archivo de configuración.
35. El método de la reivindicación 12, en el que cada una de las auditorías son con respecto a una versión específica de la línea básica.
36. El método de la reivindicación 12, en el que cada una de las auditorías son con respecto a la línea básica entonces más reciente.
- 10 37. El sistema de la reivindicación 27, en el que cada una de las auditorías son con respecto a la misma versión de la línea básica.
38. El sistema de la reivindicación 27, en el que cada una de las auditorías son con respecto a la línea básica entonces más reciente.

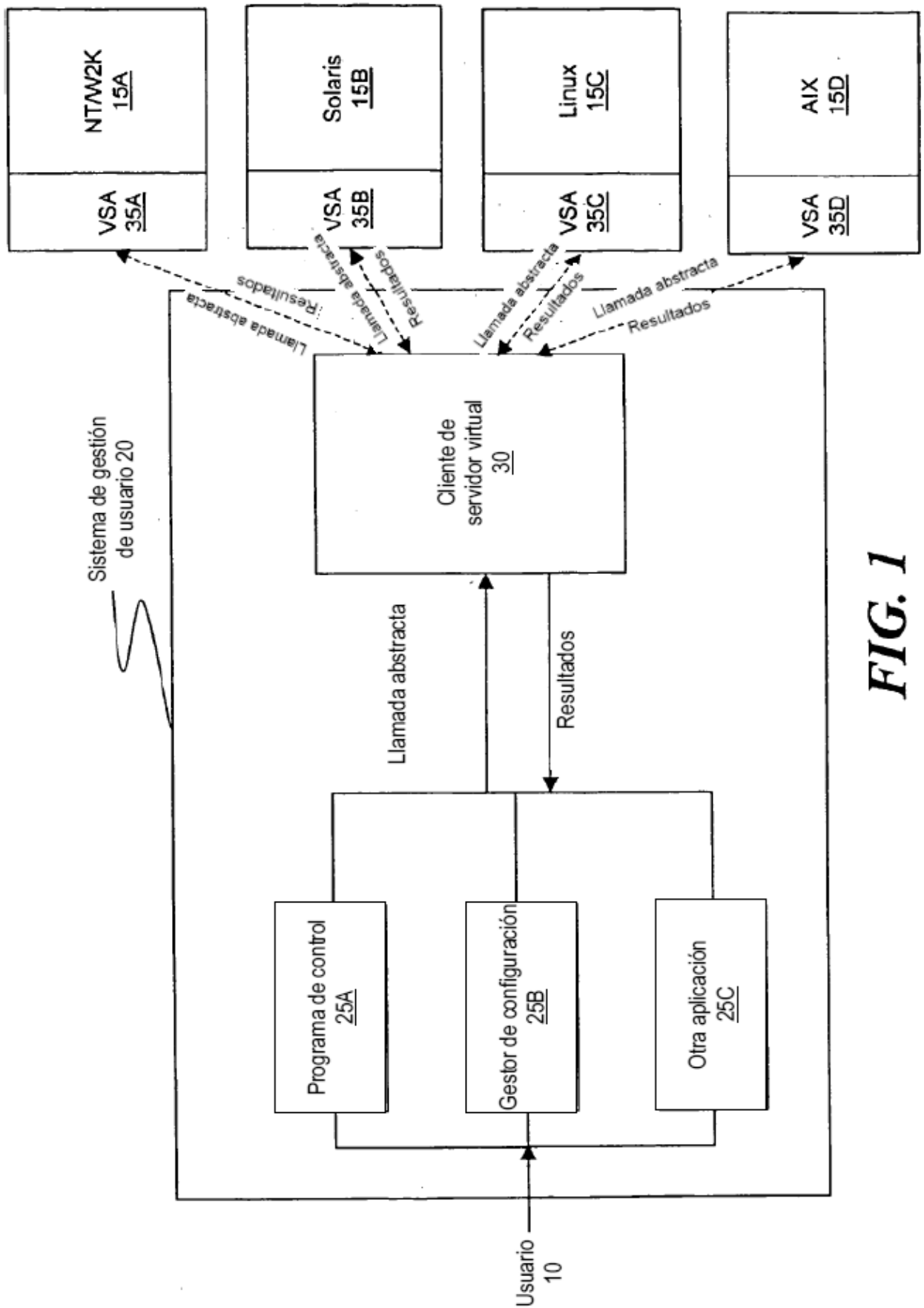


FIG. 1

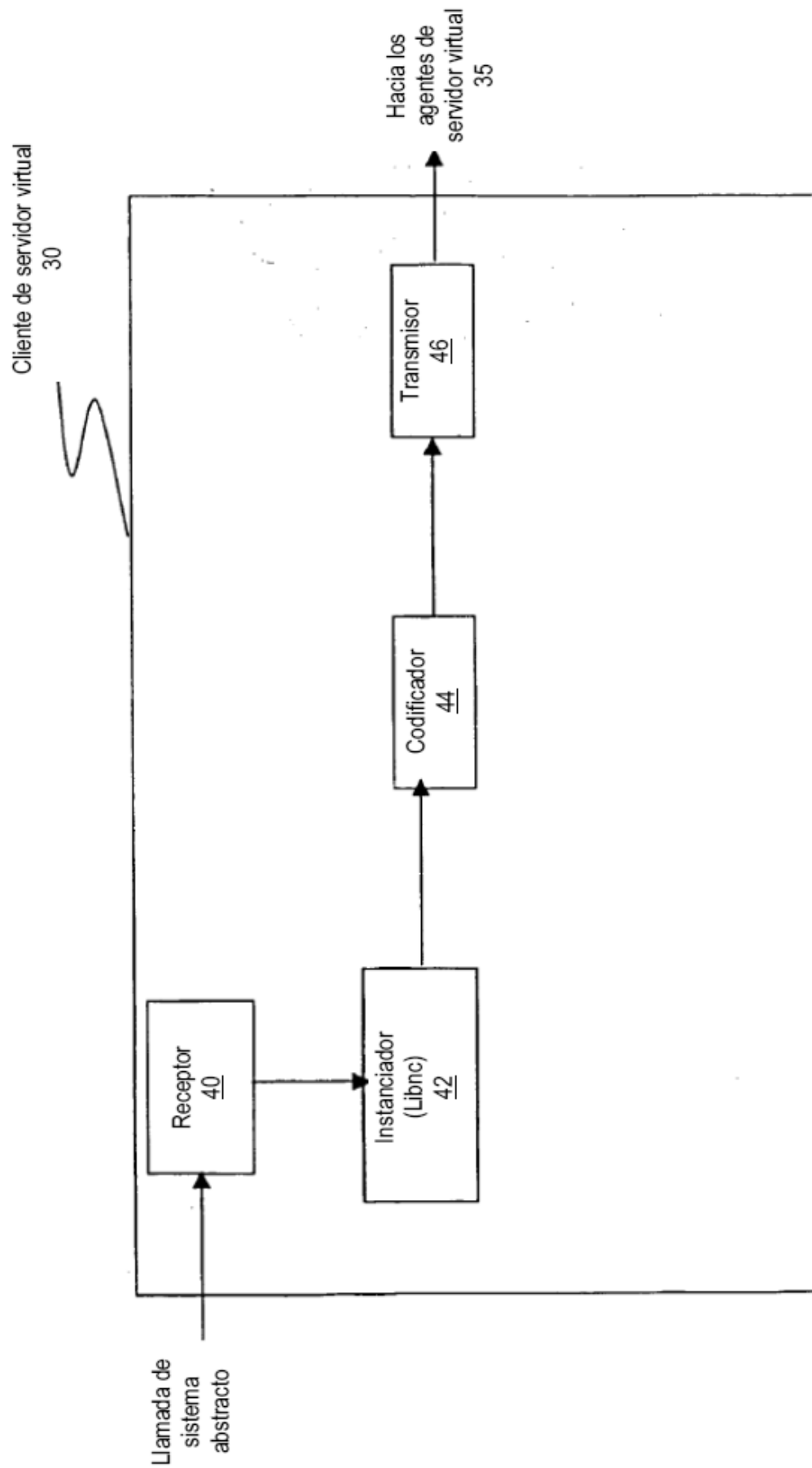


FIG. 2

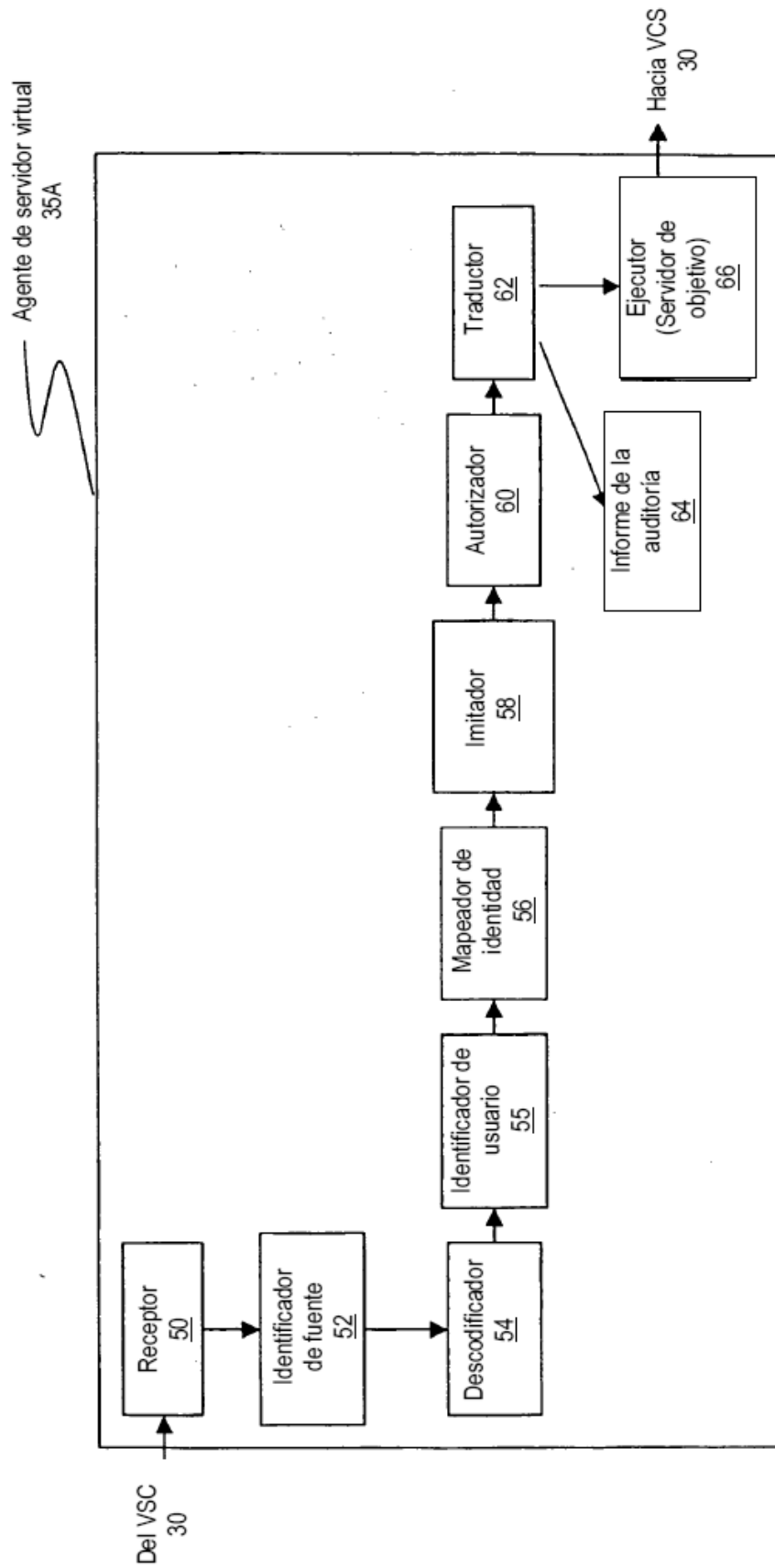


FIG. 3

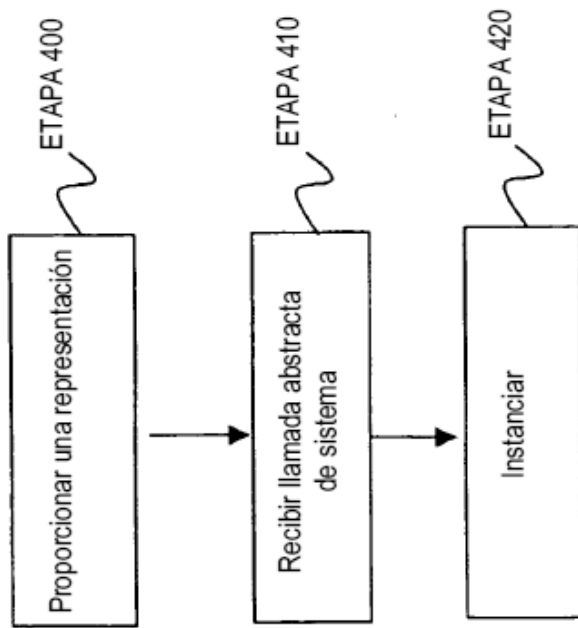


FIG. 4

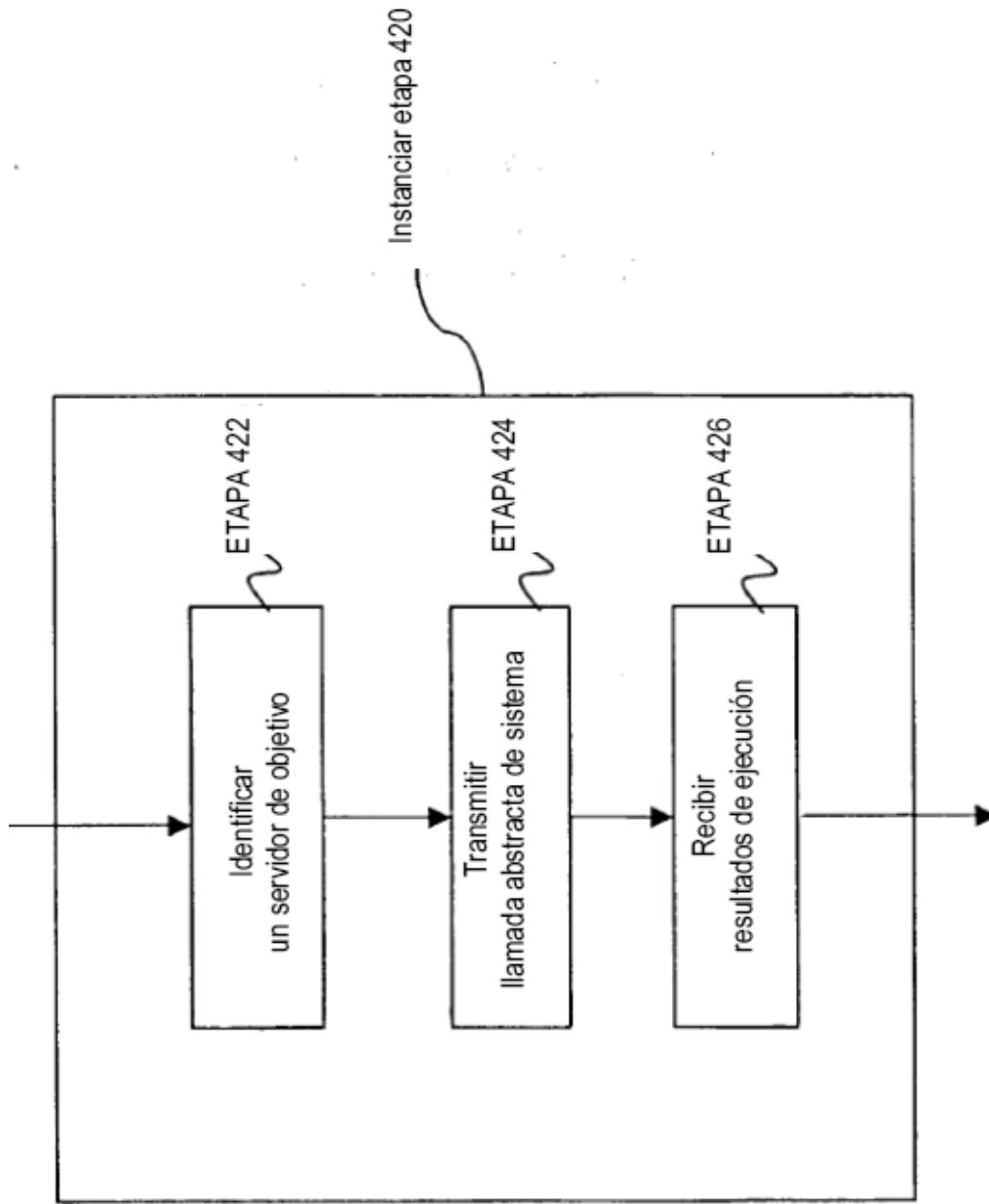
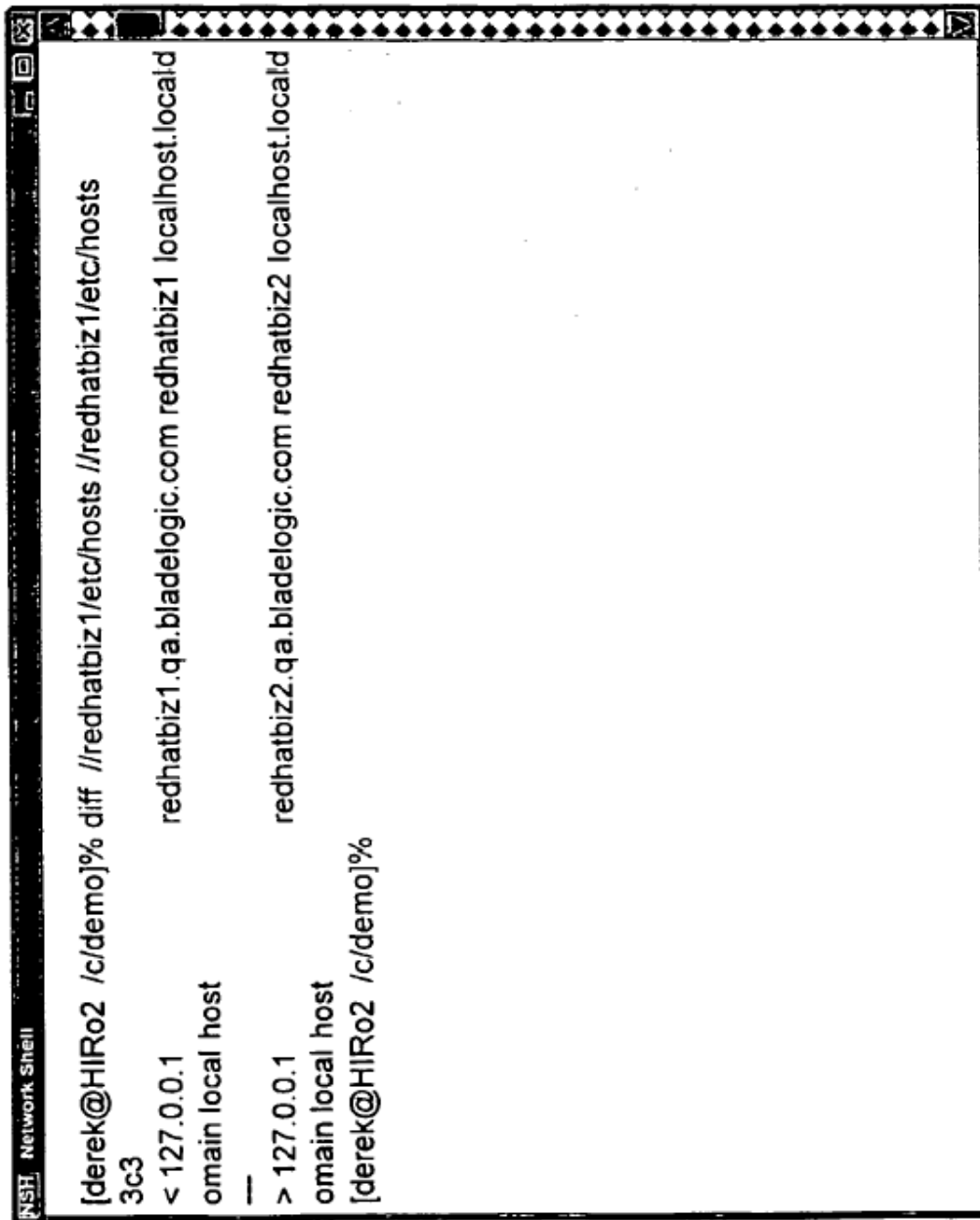


FIG. 5



```
NSH, Network Shell
[derek@HIRo2 /c/demo]% diff //redhatbiz1/etc/hosts //redhatbiz1/etc/hosts
3c3
< 127.0.0.1          redhatbiz1.qa.bladelogic.com redhatbiz1 localhost.locald
o main local host
—
> 127.0.0.1          redhatbiz2.qa.bladelogic.com redhatbiz2 localhost.locald
o main local host
[derek@HIRo2 /c/demo]%
```

FIG. 6

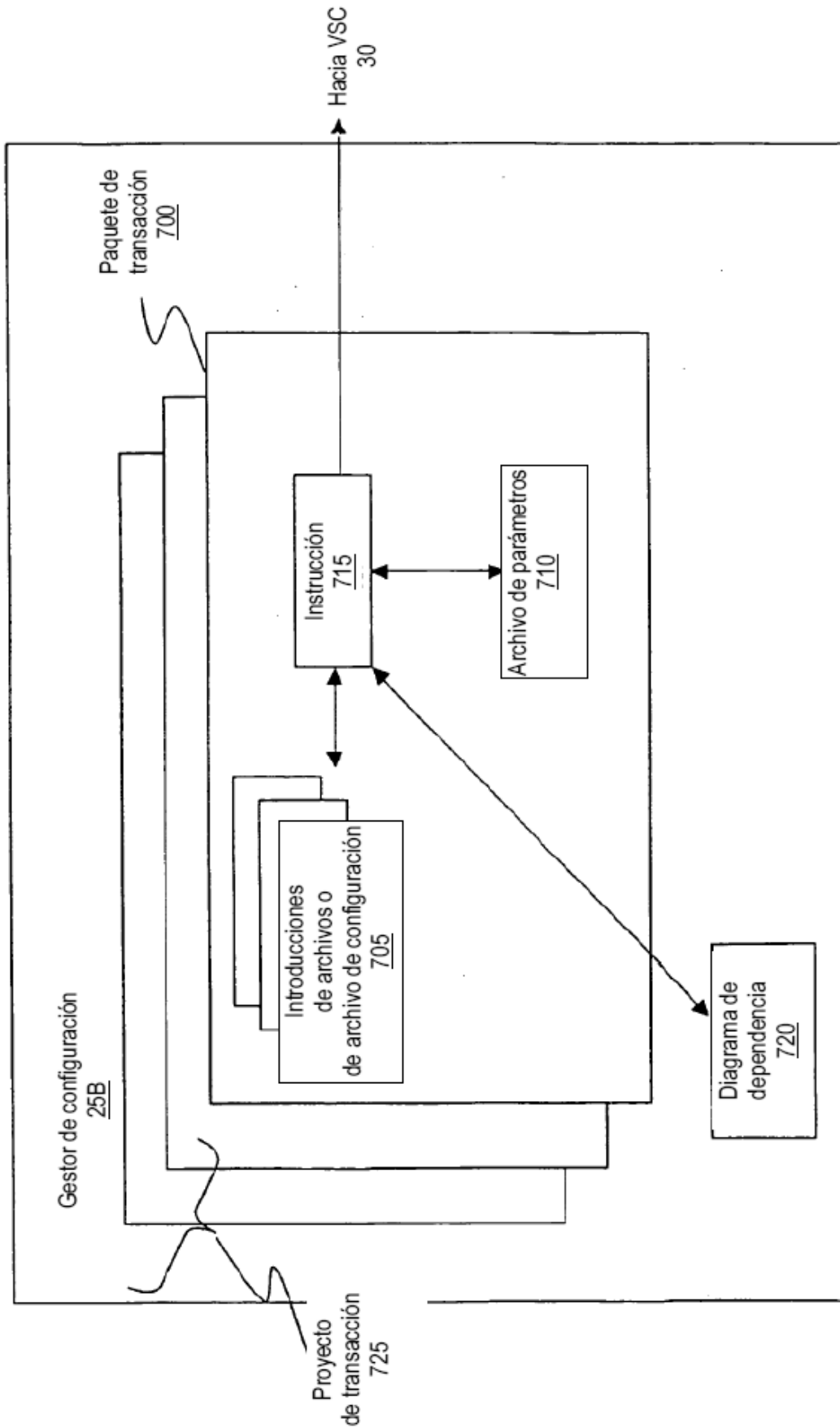


FIG. 7

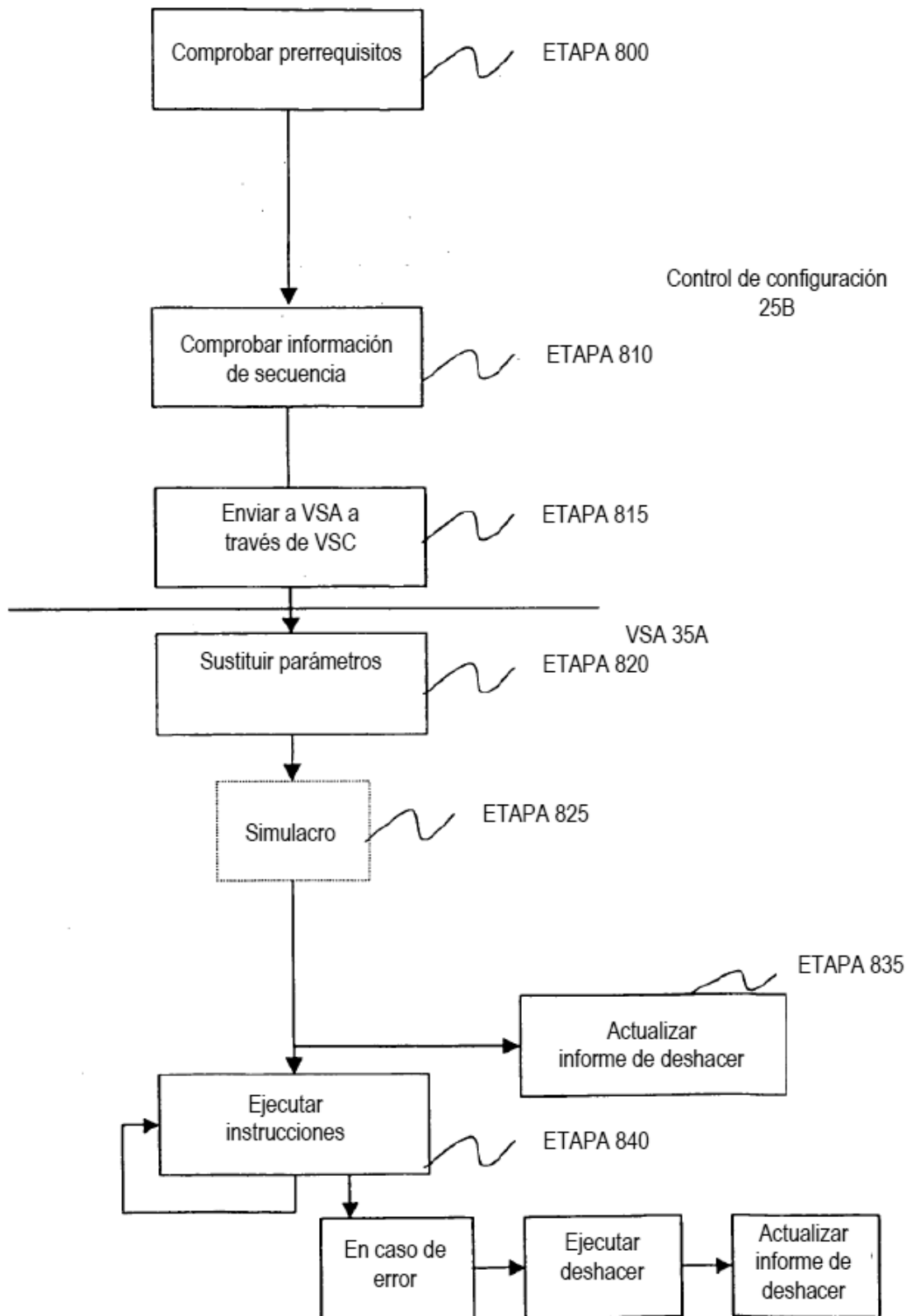


FIG. 8

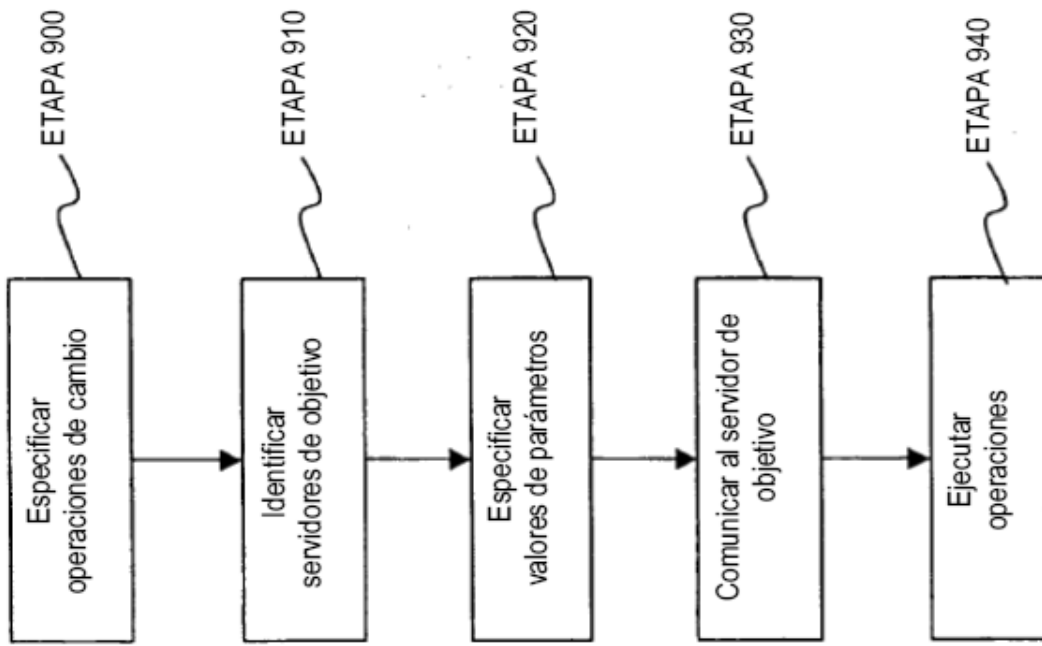


FIG. 9

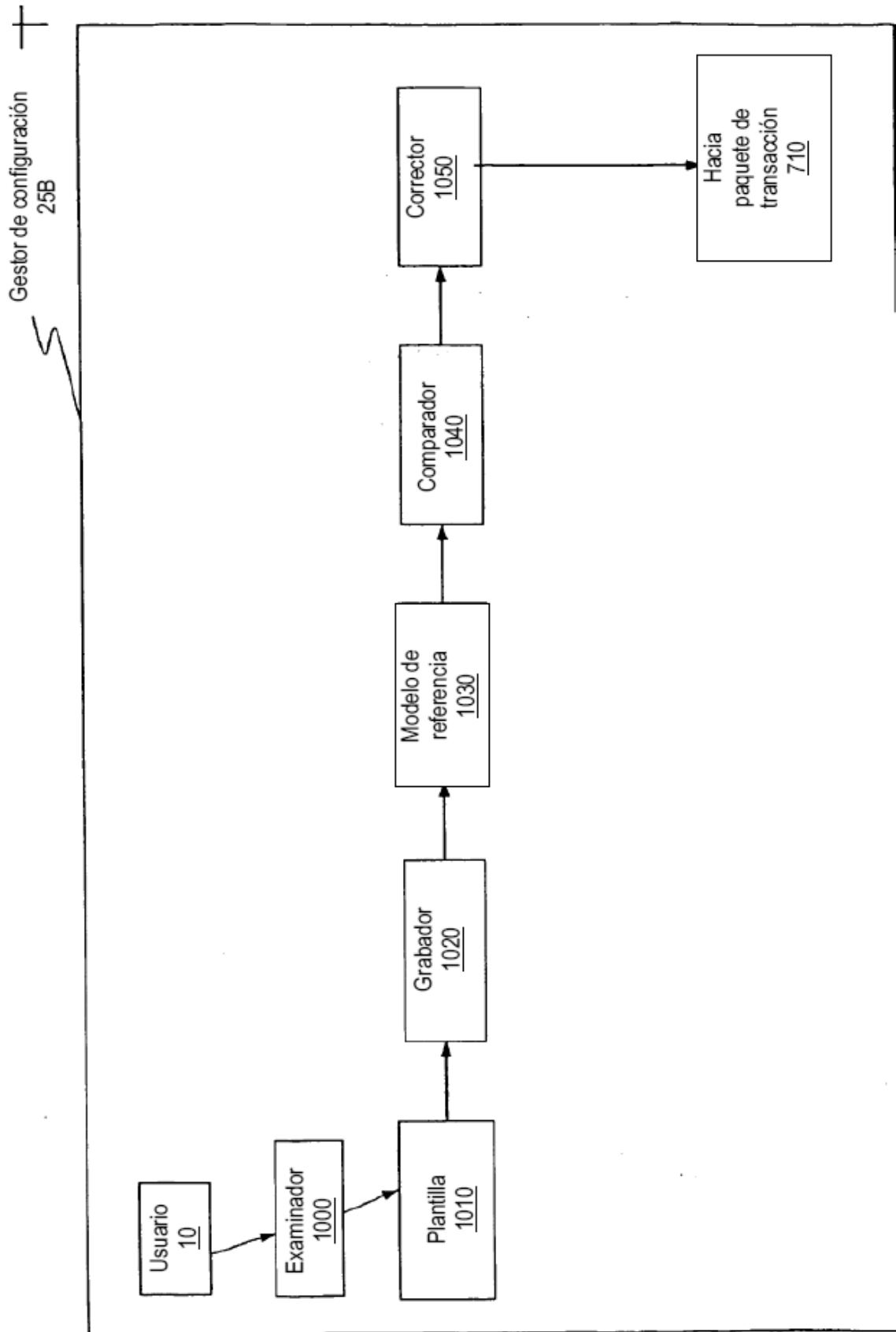


FIG. 10

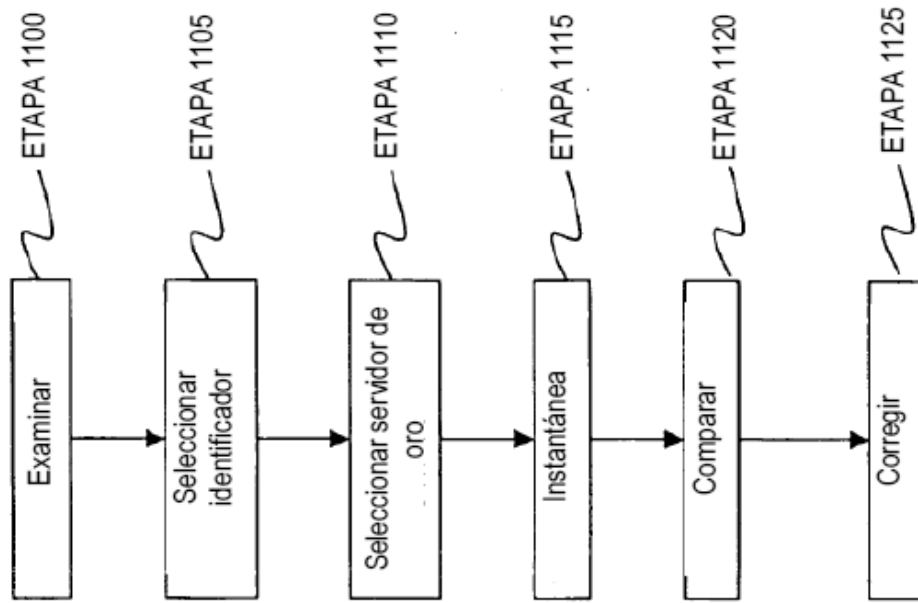


FIG. 11

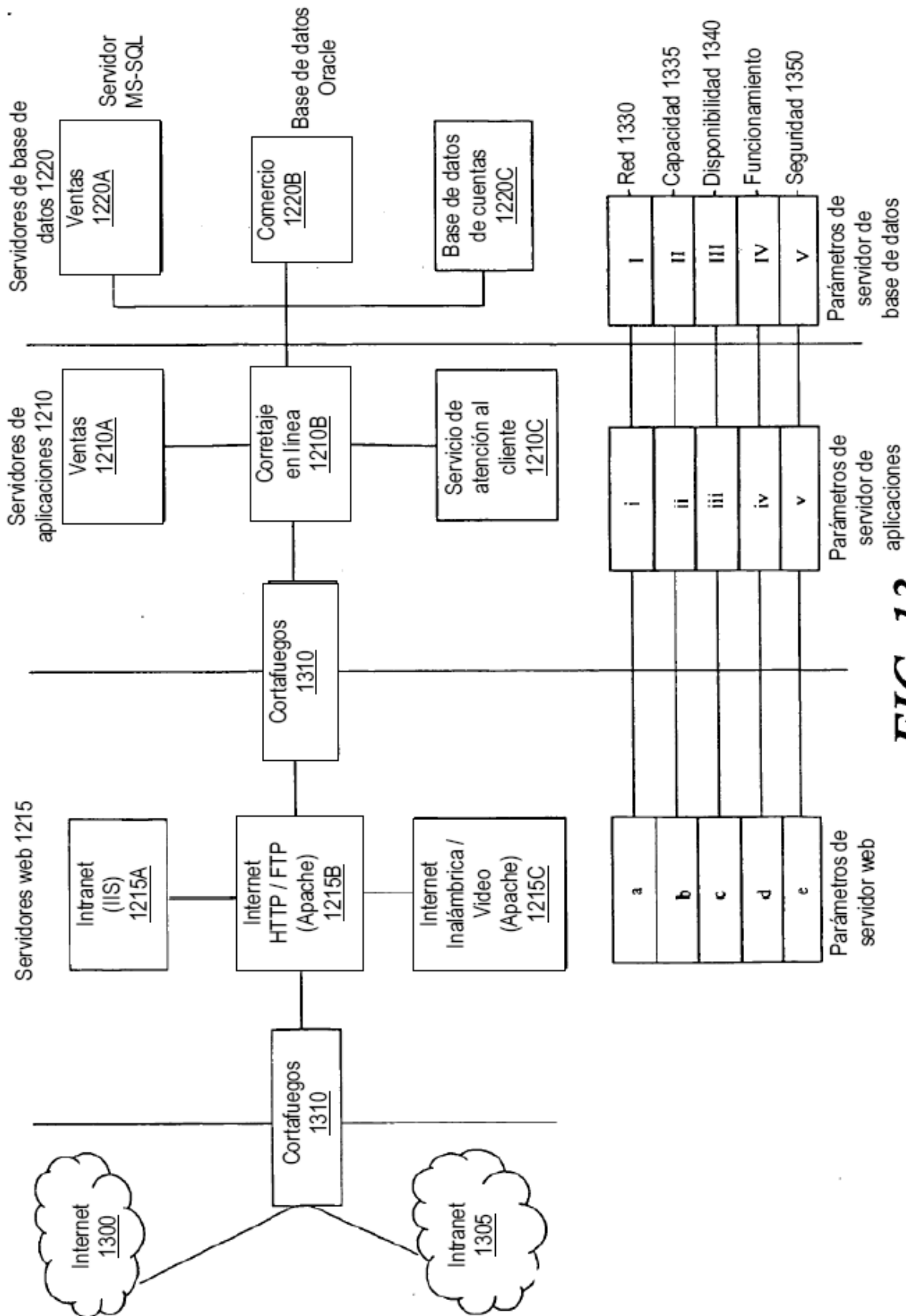


FIG. 13

1400

Bladetronic Configuration Manager

File Edit View Actions Tools Help

1420 1440 1450 1460 1410

1430

Name /	Description
SUNWmpsr	SUNWmpsr
SUNWm64, SUNWm64w, SUNWm64x	SUNWm64, SUNWm64w, SUNWm64x
SUNWdiath	SUNWdiath
SUNWwml, SUNWwml, SUNWwml, SUNWwml, SUNWwml	SUNWwml, SUNWwml, SUNWwml, SUNWwml, SUNWwml
SUNWdbas, SUNWdbas	SUNWdbas, SUNWdbas
SUNWcsr, SUNWcar	SUNWcsr, SUNWcar
SUNWcsu, SUNWcsr, SUNWcar, SUNWcsu, SUNWwml	SUNWcsu, SUNWcsr, SUNWcar, SUNWcsu, SUNWwml
SUNWcsr, SUNWcar, SUNWwml	SUNWcsr, SUNWcar, SUNWwml
SUNWdcd, SUNWdcd, SUNWdcd	SUNWdcd, SUNWdcd, SUNWdcd
SUNWman	SUNWman
SUNWcsu, SUNWcsr	SUNWcsu, SUNWcsr
SUNWcsu	SUNWcsu
SUNWcsu	SUNWcsu
SUNWcsu, SUNWcsr	SUNWcsu, SUNWcsr
SUNWdcdst	SUNWdcdst
SUNWmbl, SUNWwml, SUNWwml, SUNWwml, SUNWwml	SUNWmbl, SUNWwml, SUNWwml, SUNWwml, SUNWwml
SUNWcar, SUNWcar	SUNWcar, SUNWcar
SUNWcsu, SUNWcsr, SUNWcsu, SUNWcsr, SUNWwml	SUNWcsu, SUNWcsr, SUNWcsu, SUNWcsr, SUNWwml
SUNWwml	SUNWwml

Contents of Patches (Number of entities: 188)

FIG. 14