

Chen ZN, Chen K, Jiang JL et al. Evolución del sistema operativo en la nube: de la tecnología al ecosistema. REVISTA DE CIENCIAS Y TECNOLOGÍA INFORMÁTICA 32(2): 224–241 marzo de 2017. DOI 10.1007/s11390-017-1717-z

Evolución del sistema operativo en la nube: de la tecnología al ecosistema

Zuo Ning Chen¹, Miembro, CCF, Kang Chen¹, Jin Lei Jiang¹, Miembro, CCF, ACM, IEEE, Lu-Fei Zhang²
 canción wu³, Miembro, CCF, IEEE, Zheng-Wei Qi⁴, Miembro, CCF, ACM, IEEE
 Chun Ming Hu⁵, Miembro, CCF, IEEE, Yong-Wei Wu¹, Miembro sénior, CCF, IEEE
 Sol Yu-Zhong⁶, Miembro, CCF, IEEE, Hong Tang⁷, Ao-Bing sol⁸, y Zi Lu Kang⁹

¹Departamento de Informática y Tecnología, Universidad de Tsinghua, Beijing 100084, China

²Instituto de Tecnología Informática de Jiangnan, Wuxi 214083, China

³Escuela de Informática y Tecnología, Universidad de Ciencia y Tecnología de Huazhong, Wuhan 430074, China

⁴Escuela de Software, Universidad Jiao Tong de Shanghai, Shanghai 200240, China

⁵Escuela de Informática e Ingeniería, Universidad de Beihang, Beijing 100191, China

⁶Instituto de Tecnología Informática, Academia China de Ciencias, Beijing 100190, China 7Alibaba Cloud Computing Inc., Hangzhou 310024, China 8G-Cloud Technology Inc., Dongguan 523808, China

⁹Instituto de Tecnología de Internet de las Cosas, Academia de Ciencias de la Información de China Grupo de Tecnología Electrónica Corporación, Pekín 100081, China

Correo electrónico: chenzuoning@vip.163.com; {chenkang, jilei}@tsinghua.edu.cn; zhanglf04@126.com
 wusong@hust.edu.cn; qizhwei@sjtu.edu.cn; hucm@buaa.edu.cn; wuyw@tsinghua.edu.cn
 yuzhongsun@ict.ac.cn; hongtang@alibaba-inc.com; sunab@g-cloud.com.cn; zlkang@189.cn

Recibido el 14 de noviembre de 2016; revisado el 16 de diciembre de 2016.

Resumen El sistema operativo en la nube (SO en la nube) se utiliza para administrar los recursos de la nube de manera que puedan usarse de manera efectiva y eficiente. Y también es deber del sistema operativo en la nube proporcionar una interfaz conveniente para los usuarios y las aplicaciones. Sin embargo, estos dos objetivos a menudo entran en conflicto porque la abstracción conveniente generalmente necesita más recursos informáticos. Por lo tanto, el sistema operativo en la nube tiene sus propias características de gestión de recursos y programación de tareas para admitir varios tipos de aplicaciones en la nube. De hecho, la evolución del sistema operativo en la nube está impulsada por estos dos objetivos, a menudo en conflicto, y encontrar el equilibrio adecuado entre ellos hace que cada fase de la evolución suceda. En este documento, hemos investigado las formas de evolución del sistema operativo en la nube desde tres aspectos diferentes: la evolución de la tecnología habilitadora, la evolución de la arquitectura del sistema operativo y la evolución del ecosistema en la nube. Mostramos que encontrar las API (interfaces de programación de aplicaciones) adecuadas es fundamental para la siguiente fase de la evolución del sistema operativo en la nube. Es necesario proporcionar interfaces convenientes sin sacrificar la eficiencia cuando se eligen las API. Presentamos una práctica de SO en la nube impulsada por API, que muestra la gran capacidad de las API para desarrollar un mejor SO en la nube y ayudar a construir y ejecutar el ecosistema de la nube de manera saludable.

Palabras clave computación en la nube, sistema operativo, evolución de la arquitectura, virtualización, ecosistema en la nube

1. Introducción

1.1 Sistema operativo en la nube

En los últimos años, los sistemas de computación en la nube se han vuelto cada vez más frecuentes tanto en el extranjero•1 •3

y a nivel nacional•4 . Muchas aplicaciones tradicionales se han migrado a los sistemas en la nube. Para los desarrolladores y proveedores de servicios, los servicios y las aplicaciones se alojan sin preocuparse por la infraestructura. construcción, implementación de aplicaciones, actualización de hardware

Sección

especial de la encuesta sobre MOST Cloud y Big

Data El trabajo cuenta con el apoyo del Programa Nacional de Investigación y Desarrollo Clave de China bajo la Subvención No.

2016YFB1000500. •1 <https://aws.amazon.com/cn/documentation/>, noviembre de 2016. •2 <https://www.azure.cn/documentation/>, noviembre de

2016. •3 <https://cloud.google.com/>, noviembre de 2016. •4 <https://develop.aliyun.com/>, noviembre de 2016. ©2017 Springer Science + Business

Media, LLC & Science Press, China

o mantenimiento del centro de datos. Ahora, los componentes de la plataforma que se ejecutan dentro de los proveedores de la nube se consideran sistemas operativos (SO) en la nube. El sistema operativo en la nube[1], similar al sistema operativo tradicional que administra el hardware sin sistema operativo[2-6] y los recursos de software, atraviesa una serie de evoluciones y ahora ingresa a la siguiente etapa de evolución. Comenzamos nuestra discusión con las funciones que puede proporcionar el sistema operativo en la nube[1]. Y hacia el final de este documento, esperamos proporcionar una guía para la próxima etapa de evolución.

Los sistemas de computación en la nube [1,7-9] a menudo se crean a partir del sistema operativo de una sola máquina existente, generalmente Linux. Ahora, los otros sistemas operativos como Windows se han utilizado en la infraestructura del centro de datos. La mayoría de Los componentes del sistema de computación en la nube se construyen como aplicaciones de usuario[2-3] desde el punto de vista del sistema operativo tradicional. Sin embargo, la plataforma para ejecutar aplicaciones en la nube aún puede denominarse sistema operativo, ya que la plataforma cumple los mismos dos objetivos fundamentales que un sistema operativo tradicional. Por un lado, el sistema operativo en la nube se utiliza para administrar los recursos informáticos distribuidos a gran escala, de manera similar al sistema operativo tradicional que administra el hardware en una sola máquina.

Por otro lado, el sistema operativo en la nube proporciona abstracción para ejecutar aplicaciones de usuario. Se proporcionan API (interfaces de programación de aplicaciones) para que los programadores desarrollen aplicaciones en la nube. Esto es similar al caso en que un sistema operativo local proporciona llamadas al sistema para dar servicio a las aplicaciones. Por ejemplo, los sistemas de archivos se utilizan para proporcionar API de almacenamiento en lugar de exponer la operación de bloqueo directamente desde los discos en el sistema operativo local. Cloud OS proporciona API similares a un sistema de archivos distribuido. Además de las API de programación, se proporcionarán algunas otras facilidades para ejecutar el sistema sin problemas en la nube. El programador de trabajos es un servicio de este tipo para programar trabajos[10-12]. El sistema operativo local programa los procesos, mientras que el sistema operativo en la nube programa los trabajos distribuidos. La gestión de datos también es un componente muy importante en el sistema operativo en la nube[13-18]. Los usuarios y las aplicaciones necesitan realizar un seguimiento del flujo de datos dentro de sus aplicaciones que se ejecutan en el entorno de la nube. Las aplicaciones de procesamiento de datos necesitan especialmente una comprensión profunda de su flujo de datos entre los diferentes componentes del sistema operativo en la nube. La figura 1 muestra los niveles de la API del sistema operativo en la nube, así como varios servicios compatibles. Las API centrales son relativamente estables y se utilizan para administrar la infraestructura subyacente y ocultar la heterogeneidad y distribución de los recursos. Además de las API principales, hay otras API que brindan más funcionalidades, lo que alivia la carga de crear niveles más altos de servicios y aplicaciones en la nube. De esta manera, el ecosistema de la nube se puede construir y los servicios y aplicaciones en la nube pueden proliferar.

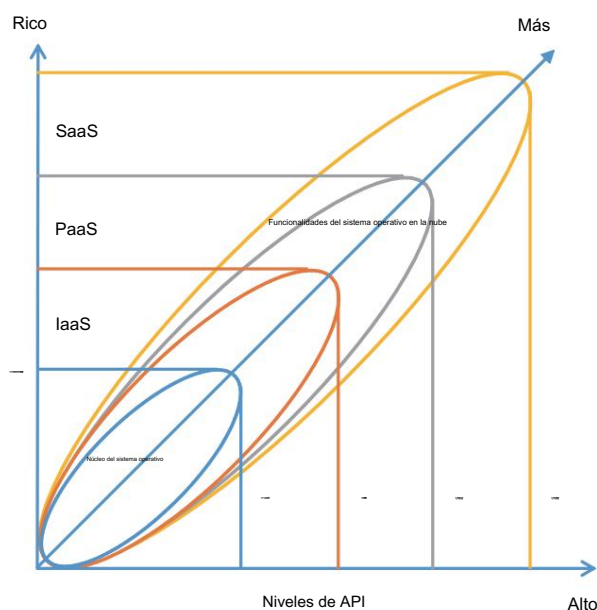


Figura 1. Los niveles de la API de Cloud OS y los servicios del ecosistema.

1.2 Requisitos del sistema operativo en la nube

En una sola computadora, está bastante claro que se necesita un sistema operativo que abstraiga el hardware básico para facilitar que las aplicaciones y los usuarios utilicen los recursos informáticos de manera conveniente. Sin embargo, la mayoría de los componentes del sistema operativo en la nube se implementan como programas a nivel de usuario. La necesidad fundamental de construir un sistema operativo en la nube necesita más discusión, ya que las aplicaciones en la nube siempre se pueden construir directamente en los sistemas operativos actuales sin preocuparse por la administración del hardware en el modo kernel. La verdadera necesidad se basa en las API de programación y el soporte de tiempo de ejecución en el entorno de la nube[13,18-21]. Los sistemas operativos locales se centran principalmente en una sola máquina. Por lo general, no consideran lógicamente ninguna aplicación en la nube como una aplicación unificada completa que abarca una gran cantidad de máquinas. Por lo tanto, los sistemas operativos tradicionales solo brindan las instalaciones básicas para las comunicaciones. Además, las aplicaciones en la nube tienen diferentes formas, como microservicios para crear aplicaciones web, procesamiento por lotes para análisis de big data[19-21], análisis de datos basados en consultas [18,22-23] y procesamiento en tiempo real de transmisión de datos . [24-26]. Todas las aplicaciones tienen sus propias organizaciones lógicas internas de múltiples componentes que se ejecutan en múltiples o incluso miles de máquinas. Exponer interfaces de comunicación muy simples no es suficiente. Los desarrolladores necesitan mayores niveles de abstracción para construir sus aplicaciones sin tener que lidiar con los detalles relacionados con los complejos patrones de comunicación y la arquitectura de la máquina. Exponer las API que se pueden ejecutar en

la parte superior de varias máquinas será muy útil. Este es un caso muy común para los profesionales actuales del sistema operativo en la nube, como Google GAE⁵ Amazon AWS⁶ Microsoft Azure⁷ y Alibaba Cloud⁸. Los desarrolladores pueden obtener las interfaces de programación sin tener en cuenta los recursos físicos. Por ejemplo, los desarrolladores pueden almacenar objetos en la nube sin conocer los discos finales que almacenan los datos[14]. La programación siempre se trata de abstracciones y necesitamos el sistema operativo en la nube para proporcionar la abstracción de programación en la nube para crear aplicaciones en la nube. La otra razón por la que es necesario un sistema operativo en la nube es que ejecutar aplicaciones en la nube es diferente de ejecutar aplicaciones en una sola máquina. Para cada sistema operativo de una sola máquina, se utilizará un programador de tareas para administrar los procesos creados en el sistema. Sin embargo, las características de tiempo de ejecución son bastante diferentes para ejecutar diferentes tipos de aplicaciones en la nube, como el procesamiento por lotes, el procesamiento de vapor o el procesamiento de gráficos. Los sistemas operativos en la nube deben construirse para administrar los recursos informáticos relacionados con miles o incluso decenas de miles de máquinas[27].

Y deberían proporcionar diferentes esquemas de gestión para diferentes aplicaciones. La coordinación[28-30] y la interferencia de diferentes aplicaciones deben tenerse en cuenta al crear un programador de trabajos de este tipo en el entorno de la nube. Por lo tanto, los administradores no necesitan administrar cada máquina individual. Según la observación anterior, el sistema operativo en la nube es bastante necesario e importante.

1.3 Prácticas de OS en la nube de última generación

Actualmente, existen múltiples proveedores de nube. Los famosos incluyen Google, Amazon, Microsoft y Alibaba Cloud. Los servicios que brindan se denominan nubes públicas porque ayudan a crear aplicaciones en la nube disponibles públicamente para los usuarios de Internet. Al enfrentarse a la gran cantidad de usuarios, estos proveedores tienen que construir sus infraestructuras para que sean escalables con un alto rendimiento. Las tecnologías utilizadas detrás de la infraestructura son tan importantes que las comunidades de código abierto las han adoptado y ayudan a construir las versiones de código abierto de la infraestructura en la nube, como OpenStack y Hadoop. Podemos identificar las infraestructuras y sus encarnaciones de código abierto como sistemas operativos en la nube.

propio sistema operativo en la nube bajo diferentes consideraciones. Amazonas comenzó con instancias de máquinas virtuales (VM) para que los desarrolladores y administradores comenzaran a construir sus propios sistemas, liberándolos del mantenimiento del hardware subyacente. Al mismo tiempo, Google inició su sistema operativo en la nube considerando una gran cantidad de procesamiento de datos. Estos son los dos principales pioneros en la construcción del sistema operativo en la nube. Los servicios prestados por estas dos empresas son bastante diferentes. Los seguidores, como Microsoft, comenzaron con servicios similares con diferentes intereses.

rostros expuestos. Por razones comerciales, las interfaces son propietarias, lo que lleva a versiones de código abierto para proporcionar interfaces similares pero ligeramente diferentes. Por lo tanto, los desarrolladores de aplicaciones en la nube tienen que enfrentar el problema del bloqueo del proveedor, lo que significa que las aplicaciones creadas para un proveedor específico no se ejecutarán en las otras plataformas. Incluso para crear entornos de nube privada utilizando software de código abierto, los desarrolladores deben confiar en la implementación específica de código abierto de una versión específica. Esto impedirá la construcción de un ecosistema en la nube o dificultará la adopción de tecnologías de computación en la nube. Para la próxima fase de la evolución del sistema operativo en la nube, las API bien definidas serán fundamentales.

1.4 Impulso de la evolución del sistema operativo en la nube

Al igual que el sistema operativo local tradicional, el sistema operativo en la nube no es estático. Va evolucionando de generación en generación. El ímpetu de la evolución del sistema operativo en la nube se debe principalmente a dos objetivos considerados en conflicto. Uno es mejorar la eficiencia y el otro es mejorar las experiencias de los usuarios (y desarrolladores). El primero incluye la mejora del rendimiento y la eficiencia de los recursos informáticos utilizados. Este objetivo es bastante sencillo: no desperdiciar nunca la potencia informática proporcionada. Lo último está relacionado con mejores interfaces de programación, instalaciones de administración más convenientes e interfaces fáciles de usar. Sin embargo, estos dos objetivos a menudo se consideran contradictorios. La razón es que mejores experiencias de usuario requieren interfaces más convenientes que a menudo necesitan capas adicionales de abstracciones. Se introducirán más cálculos generales para tales abstracciones. La exposición de interfaces de nivel inferior es más amigable con el rendimiento, pero implica más trabajo para los desarrolladores, lo que daña el ecosistema de la nube. Por lo tanto, la evolución del sistema operativo en la nube es principalmente para encontrar una mejor compensación entre estos dos conflictos.

Sin embargo, cada practicante de SO en la nube comenzó su

•5 <https://cloud.google.com/>, noviembre de 2016. •6 <https://aws.amazon.com/cn/documentation/>, noviembre de 2016. •7 <https://www.azure.cn/> documentación/, noviembre de 2016. •8 <https://develop.aliyun.com/>, noviembre de 2016.

objetivos. La compensación impulsa la evolución de la tecnología, la evolución de la arquitectura y la evolución del ecosistema. Discutiremos cada uno de ellos en detalle en la Sección 2, Sección 3 y Sección 4 respectivamente.

1.5 Importancia de las API de Cloud OS

Para admitir más aplicaciones en la nube, se agregan más interfaces de programación de aplicaciones al conjunto API de sistemas operativos en la nube. Sin embargo, algunas interfaces de programación fundamentales (llamadas API centrales) se están volviendo cada vez más estables. Estas API centrales son bastante importantes y casi todos los proveedores de la nube las ponen a disposición de los desarrolladores y usuarios, aunque sus implementaciones pueden ser diferentes. Esto es clave para construir el ecosistema de aplicaciones en la nube. Las aplicaciones creadas con el mismo conjunto de API pueden beneficiarse de

la misma plantilla de construcción. Esto puede hacer que prolifere la aplicación que se ejecuta con una tecnología similar.

Además, con las API centrales, las aplicaciones pueden comunicarse y facilitar la creación de aplicaciones en la nube más grandes y complejas.

Al igual que en los casos de los sistemas operativos tradicionales, la elección de las API adecuadas es muy importante para los usuarios de sistemas operativos en la nube. Para una mayor evolución, debemos aclarar las definiciones de API y sus niveles y categorías. Esto trata de hacer la evolución en la dirección correcta. Para tal objetivo, este documento analiza la evolución actual del sistema operativo en la nube desde diferentes ángulos, incluidas las tendencias tecnológicas, la mejora de la arquitectura del sistema operativo y el ecosistema del sistema operativo en la nube.

2 Habilitación de la evolución de la tecnología

Dado que la definición de API está estrechamente relacionada con el desarrollo de tecnologías, primero estudiaremos la evolución de las tecnologías habilitadoras para la computación en la nube. Dado que la evolución se basa en el ímpetu de la compensación adecuada entre los dos objetivos, a saber, la mayor eficiencia y las experiencias de usuario más convenientes, el análisis también se centrará en las tecnologías que se ocupan de estos dos aspectos. Las evoluciones de la virtualización y la interfaz de programación están bastante relacionadas con brindar una mejor experiencia de usuario del sistema operativo en la nube. El hardware y la informática de gran venta están relacionados con la mejora de la eficiencia informática.

2.1 Evolución de la tecnología inducida por máquinas virtuales

La primera evolución tecnológica está bastante relacionada con la introducción de las máquinas virtuales[31-32]. A principios

etapa de la computación en la nube, VM se considera incluso como sinónimo de computación en la nube. Se podría decir que si no hubiera máquinas virtuales, no habría computación en la nube. Las máquinas virtuales son como máquinas físicas, pero ejecutan instrucciones utilizando hardware simulado en lugar de hardware físico. Esto es principalmente conveniente para los usuarios que desean implementar sus sistemas en Internet sin comprar, implementar y mantener hardware. Amazon es la primera empresa en brindar servicios de VM al público. Esto es especialmente útil para las empresas emergentes que brindan servicios de Internet.

Las instancias de VM se pueden alquilar en función del usuario actual escala y se puede ampliar bajo demanda. Este es un nuevo modelo de aprovisionamiento de infraestructura que generalmente se considera difícil y requiere mucho tiempo[33]. Sin embargo, la tecnología de VM introduce una gran cantidad de sobrecarga de rendimiento. Se han realizado muchos esfuerzos para mejorar el rendimiento [34-38]. A pesar de la mejora del hardware discutida en la Sección 3, se han desarrollado muchas tecnologías ligeras para proporcionar una capacidad similar pero con un rendimiento mejorado.

Una es usar contenedores livianos para reemplazar las máquinas virtuales pesadas [39-41]. Los contenedores son la virtualización dentro del kernel del sistema operativo. Proporciona un mayor nivel de virtualización en lugar de la virtualización a nivel de instrucciones que utilizan las máquinas virtuales. Por lo tanto, se puede eliminar una gran cantidad de sobrecarga de simulación. Es una tendencia típica que se mejora la experiencia del usuario con la introducción de gastos generales. La compensación requiere mejorar el rendimiento de la plataforma de próxima generación. Aquí, no significa que todas las máquinas virtuales serán reemplazadas por contenedores. Las máquinas virtuales tienen su propia virtud y se utilizan para la

aplicaciones que necesitan más aislamiento en lugar de un mayor rendimiento. Algunas aplicaciones pueden beneficiarse de los contenedores para un mayor rendimiento.

A pesar de la mejora del rendimiento de los contenedores sobre las máquinas virtuales, la computación en la nube quiere mejorar aún más el rendimiento. Con este objetivo, en los últimos años han surgido otras formas de modelos de computación para llevar la computación al usuario final. Edge computing[42-43] se encuentra entre las tecnologías más prometedoras en esta tendencia. Mientras que la CDN (red de entrega de contenido) se utiliza para acercar los datos al usuario final, la computación perimetral intenta acercar la computación, en lugar de los datos, al usuario final. Esta es otra forma de mejorar el rendimiento de la infraestructura virtualizada además del uso de contenedores. En la informática perimetral, los desarrolladores tienen que dividir su computación en diferentes partes y descargar parte de la computación a los servidores perimetrales en lugar de a los centralizados.

2.2 Evolución de la mejora del hardware

El rendimiento es siempre la principal preocupación en cualquier evolución informática. Cómo mejorar el rendimiento del sistema operativo en la nube no es una excepción. Existen principalmente dos formas de mejorar el rendimiento de las aplicaciones en la nube ambiente.

Una forma es utilizar el hardware recién desarrollado y la otra es a través de la mejora del software. La evolución del hardware siempre va acompañada de algún software correspondiente. Durante la evolución del sistema en la nube, hay muchos avances de hardware principalmente para mejorar el rendimiento. Como se mencionó anteriormente, VM introduce grandes cantidades de sobrecarga informática debido al nivel adicional de direccionamiento indirecto. Intel ha realizado grandes esfuerzos para mejorar el rendimiento de la virtualización de hardware, por ejemplo, proporcionando VT-x para virtualización de CPU y virtualización de memoria, y VT-d para virtualización de dispositivos. Con la ayuda de la virtualización de hardware, la sobrecarga de rendimiento de la VM actual es insignificante, por lo general menos del 5 %. Además de la mejora de la virtualización del sistema, hay muchas otras mejoras que utilizan nuevas funciones de hardware. RDMA (acceso directo a memoria remota) se introdujo hace una década para mejorar el rendimiento de la red y ahora se vuelve predominante en los centros de datos de computación en la nube [44-46]. Además, la topología de la red de un centro de datos es muy importante para el rendimiento de la computación en la nube[47-48]. Para el rendimiento de una sola máquina, se introdujeron FPGA[49-53] y GPGPU[54] para mejorar el rendimiento de aplicaciones específicas. Ahora se utilizan ampliamente en el entorno de la nube para aplicaciones de nube especializadas. Los dispositivos SSD (unidad de estado sólido) y NVM (memoria no volátil) serán cada vez más importantes para mejorar el rendimiento del almacenamiento.

Por el lado del software, el software del sistema tiene que adaptarse a las nuevas características del hardware. Los sistemas operativos heredados se diseñaron para máquinas de un solo núcleo y una sola CPU. Ahora, las máquinas multinúcleo son omnipresentes. El sistema debe adaptarse al cambio para proporcionar una capacidad real de tareas múltiples [55-57] en lugar de tareas múltiples de tiempo compartido. Los sistemas operativos han proporcionado una arquitectura multiproceso para ejecutar tareas simultáneas[58-59] . Los subprocesos se utilizarán como tareas múltiples dentro de un solo proceso. Hoy en día, los nuevos componentes operativos y la biblioteca de programación tienen que soportar nuevas características como instrucciones SIMD[60-63], procesamiento paralelo GPGPU y FPGA para acelerar el rendimiento[64]. Estas son todas las adaptaciones del sistema para adoptar la evolución del hardware subyacente. Y también, será bastante claro.

que el futuro sistema operativo en la nube admitirá el nuevo desarrollo de hardware. Por ejemplo, han surgido máquinas virtuales a gran escala para admitir aplicaciones en la nube, como los juegos de Internet, que requieren grandes cantidades de memoria y una gran cantidad de núcleos de CPU. Por lo tanto, para la próxima generación de desarrollo de computación en la nube, obviamente se debe considerar seriamente la evolución del hardware. En

por otro lado, las aplicaciones en la nube y el sistema operativo en la nube también pueden impulsar la mejora del hardware.

2.3 Evolución de la escalabilidad del software

A pesar del desarrollo de hardware y sus módulos de soporte de software, la plataforma de software tiene su propia adaptación para soportar aplicaciones en la nube de mayor rendimiento. Existen principalmente dos formas de mejora.

Uno se llama escalamiento vertical, que se enfoca en el rendimiento de una sola máquina, y el otro es escalamiento horizontal, que se enfoca en el rendimiento de varias máquinas. El sistema operativo tradicional se centró principalmente en la ampliación del rendimiento. Múltiples tareas y múltiples subprocesos son las tecnologías comunes en el sistema operativo para mejorar el rendimiento. Y las bibliotecas de programación se utilizan para aprovechar el poder de SIMD, GPGPU y FPGA como se menciona en Subsección 2.2.

Sin embargo, a pesar de los grandes esfuerzos para mejorar el rendimiento de una sola máquina, el rendimiento de una sola máquina nunca será suficiente para muchas aplicaciones en la nube que requieren una gran cantidad de procesamiento como en los sistemas operativos distribuidos[65]. Con el desarrollo de aplicaciones de big data, este será un caso inevitable en muchas áreas. Cloud OS ha hecho un gran trabajo para mejorar el rendimiento y la programabilidad de la plataforma. Esto generalmente se denomina esquema de escalamiento horizontal para tratar con más máquinas. Por ejemplo, los sistemas de archivos distribuidos se construyen para almacenar una gran cantidad de datos no estructurados, mientras que las bases de datos distribuidas se construyen

para almacenar datos estructurados. Se han creado marcos de computación como MapReduce y Spark para la construcción de aplicaciones en la nube. Otra mejora que no puede ocurrir en una sola máquina es que la gestión y programación de tareas en el entorno de la nube es bastante diferente a la de una sola máquina.

El sistema operativo en la nube tiene que administrar miles de máquinas de un clúster, o administrar múltiples clústeres en un centro de datos, o incluso administrar múltiples centros de datos en todo el mundo. La infraestructura de gestión es sin duda bastante completa y compleja. El sistema operativo en la nube tiene que proporcionar la función de gestión tolerante a fallos y escalable para hacer frente a una gran cantidad de nodos y las situaciones graves como fallos de la máquina, fallos de la red e incluso

falla del centro de datos. Además de la escala de la nube computación, el sistema operativo en la nube tiene que considerar varios tipos de aplicaciones. El programador de tareas en el entorno de la nube tiene que enfrentarse a diferentes granularidades de tareas[66]. Por ejemplo, las máquinas virtuales y los contenedores se pueden usar como unidades para la programación[67]. Esto es bastante diferente de la programación de tareas en el sistema operativo tradicional [68-69]. La migración en vivo de VM puede ayudar a migrar la carga de trabajo de una máquina a otra [70-71], mejorando así la interoperabilidad de los diferentes componentes. Tal caso nunca ocurrirá en una sola máquina.

En resumen, la evolución del software relacionada con la mejora del rendimiento trata de mejorar la eficiencia mediante la ampliación y la ampliación. Para el desarrollo futuro, la organización del sistema operativo debe adaptarse a las nuevas aplicaciones en la nube. La granularidad única y la interfaz uniformada no serán suficientes para construir el ecosistema de la nube. Los desarrolladores tienen que acceder a varias granularidades de abstracción de infraestructura que debe proporcionar el sistema operativo en la nube.

2.4 Evolución de la programabilidad

Además de la evolución de la máquina virtual, la mejora del rendimiento del hardware y la plataforma, la otra evolución bastante importante es la evolución de la interfaz de programación. En los primeros días de la computación en la nube, había una cantidad bastante pequeña de interfaces de programación. La mayoría de las interfaces estaban relacionadas con la infraestructura subyacente. Por ejemplo, AWS primero proporcionó interfaces de programación para manipular instancias de VM y para la comunicación entre instancias de VM. Estos se consideran API de bajo nivel y son adecuados para la construcción.

ción de la infraestructura virtual. Estas API no son para cualquier aplicación como la lógica empresarial o el análisis científico. Aunque pueden beneficiar a los desarrolladores o administradores del sistema, los desarrolladores de aplicaciones no pueden obtener mucha ayuda de estas interfaces de programación a nivel de infraestructura. Los desarrolladores de aplicaciones tienen que recurrir a las API del sistema operativo tradicionales, especialmente las API de redes y/o el middleware de clúster para crear sus aplicaciones en la nube.

El uso de middleware o API de sistema operativo existentes no es suficiente para crear aplicaciones en la nube por varios motivos. Los diseñadores del entorno existente no tuvieron una consideración adecuada de las características de los sistemas de nube actuales. Nunca han visto un sistema de ultraescala como la computación en la nube. Las API de computación en la nube necesitan rediseñarse y tienen un mejor soporte para las aplicaciones que se ejecutan en el entorno de la nube. Es por eso que la evolución de la interfaz de programación comenzó a admitir la nube.

aplicaciones Y las primeras interfaces de programación para aplicaciones intentaron imitar las interfaces tradicionales del sistema operativo tradicional, pero se ajustaron al entorno de la nube. Un sistema de archivos distribuido es un ejemplo típico que amplía un sistema de archivos local. Las aplicaciones pueden usar interfaces de programación similares para almacenar objetos de datos en los sistemas de archivos distribuidos. Suceden cosas similares en el sistema operativo en la nube con SQL (lenguaje de consulta estructurado) API[24].

Con la mejora adicional y el florecimiento de las aplicaciones, los ecosistemas de la nube han extraído interfaces de programación más comunes para crear nuevas aplicaciones en la nube. A diferencia de las API generales anteriores, estos nuevos conjuntos de API tienen sus propios propósitos a nivel de aplicación.

Por ejemplo, las API de mensajería se utilizan para enviar y recibir mensajes. Las API de autenticación distribuida se utilizan para la identificación y verificación de usuarios. Dis

Las API de registro tributadas ayudan a los desarrolladores a realizar el registro de programación, que es muy importante para el desarrollo y la supervisión de aplicaciones. Estas API son bastante necesarias en el entorno de la nube, pero no tienen equivalente en los sistemas operativos tradicionales.

Es obvio que diferentes conjuntos de API son y se construirá para facilitar el desarrollo de diferentes aplicaciones en la nube. Por ejemplo, hemos visto las API de computación en la nube que admiten el procesamiento por lotes, el procesamiento de transmisión [24-25] y el procesamiento de gráficos [72-79]. Las API en la nube para el aprendizaje profundo y la inteligencia artificial[80] se encuentran ahora en un rápido desarrollo. Los patrones de cómputo han cambiado del cómputo de datos estructurados a la combinación de cómputo de datos estructurados y no estructurados para admitir más aplicaciones de big data. Por lo tanto, el futuro sistema operativo en la nube tendrá más API de programación compatibles con nuevas aplicaciones en la nube. La plataforma y las API expuestas deben admitir varias aplicaciones con una lógica compleja que maneja una gran cantidad de datos.

En resumen, hemos investigado la evolución de la virtualización de sistemas, la mejora del hardware y las interfaces de rendimiento y programación. No hay duda de que el sistema operativo en la nube proporcionará más y más API de programación en el futuro. Algunas de las API no son puramente funcionales y se utilizarán para detectar las características de tiempo de ejecución a nivel de aplicación para mejorar la eficiencia. Además, las API principales serán estables porque se requieren API estables para crear y ejecutar ecosistemas en la nube. En función de las API principales, se puede definir un nivel más alto de API y se puede construir todo el ecosistema de la nube.

3 Evolución de la arquitectura del sistema operativo en la nube

Con el desarrollo de la infraestructura informática, proliferan nuevas aplicaciones de red en la nube. El sistema operativo en la nube y el sistema operativo de un solo nodo tienen la evolución de sus arquitecturas internas para adaptarse a esa tendencia de desarrollo. Las aplicaciones en la nube han puesto a disposición de los programadores más API, enriqueciendo así el ecosistema del sistema operativo en la nube.

3.1 Evolución de la arquitectura del sistema operativo de un solo nodo

En una etapa inicial, los recursos informáticos, como los mainframes, eran bastante caros y el sistema operativo de tiempo compartido se utilizaba como elemento fundamental para gestionar el hardware subyacente. Esta fue la primera vez que se creó el sistema operativo. El sistema operativo abstrajo la interacción entre los programas y las máquinas e hizo posible compartir recursos informáticos entre múltiples usuarios. Según los recursos informáticos disponibles en ese momento, el sistema operativo se diseñó para tener un kernel monolítico (como UNIX y Solaris) con un espacio de nombres único y un sistema binario a gran escala que cubriera la gestión de procesos, la gestión de memoria y el sistema de archivos. Dicha arquitectura puede tener un alto rendimiento pero con una estructura interna compleja. Más tarde, surgió el micronúcleo como Mach y QNX. Dichos núcleos solo contienen componentes necesarios y críticos, como el aislamiento y la comunicación entre procedimientos. Otros módulos y funciones del sistema se ejecutan como componentes independientes sobre el kernel. Esto puede simplificar el desarrollo de sistemas operativos y facilitar la estabilización de todo el sistema. Al usar el micro-núcleo, se puede agregar un nuevo servicio usando un nuevo proceso sin modificar el propio núcleo. El micronúcleo tiene mejor escalabilidad pero menos rendimiento. Es el rendimiento lo que impide la preferencia de micro-kernel. Aunque el debate entre estas dos arquitecturas continuará, existen API estándar de facto. POSIX (interfaz de sistema operativo portátil para UNIX) es un estándar de programación muy importante y ahora es compatible con la mayoría de los sistemas operativos disponibles. Esto hace posible ejecutar aplicaciones en diferentes plataformas.

Más tarde, los recursos informáticos se volvieron más potentes, la red conecta todas las instalaciones informáticas del mundo y la intención del sistema operativo se ha ampliado. El modelo de interacción entre máquinas y humanos también se cambia y las interfaces gráficas de usuario (GUI) se hacen imprescindibles. Mientras tanto, el sistema operativo Windows se hizo omnipresente, proporcionando a los desarrolladores

capacidades de creación de ventanas, dibujar en ventanas y servicios para usar varios tipos de dispositivos gráficos.

Recientemente, se ha puesto más poder de cómputo en la computación del lado del servidor, y las CPU se están volviendo multinúcleo y heterogéneas. Los sistemas operativos tradicionales carecen del soporte de arquitecturas heterogéneas y las aplicaciones pueden interferir en el rendimiento. Además, el modelo de memoria compartida y el protocolo de coherencia de caché basado en sincronización y exclusión mutua han limitado la escalabilidad del sistema[81]. La tecnología de virtualización se puede utilizar para lidiar con la heterogeneidad del procesador y para realizar el aislamiento. Además de la mayor utilización de recursos, la virtualización puede aumentar la movilidad de las aplicaciones, reducir los costos de administración y permitir la recuperación ante desastres. En el mantenimiento de sistemas a gran escala, las aplicaciones con el sistema operativo subyacente se empaquetan en una sola imagen virtual. Esta es una nueva forma de entrega de software llamada dispositivo virtual, que puede desacoplar la implementación de la aplicación de la plataforma de hardware subyacente. Se proponen muchas interfaces de programación como libvirt[82] para manipular

máquinas virtuales.

3.2 Evolución de la arquitectura del sistema operativo en la nube

La computación en la nube se puede considerar como un paradigma de computación distribuida completamente comercializado. Los sistemas operativos distribuidos tradicionales como Amoeba[83] intentan que el entorno distribuido sea transparente para la capa del sistema operativo y la capa de la aplicación. En la capa del sistema operativo, la memoria compartida distribuida y la migración de procesos pueden lograr el objetivo. En la capa de aplicación, la llamada a procedimiento remoto y la comunicación de mensajes entre procesos se pueden utilizar para lograr el mismo objetivo. La transparencia del entorno distribuido es válida solo en la situación de latencia de comunicación lo suficientemente baja. Ahora los recursos informáticos son más flexibles, ubicuos y heterogéneos. Junto con las aplicaciones en la nube recientemente desarrolladas como MapReduce, todos los factores mejoran el desarrollo del software del sistema en la nube. MapReduce[13], propuesto primero por Google, proporciona un marco débilmente acoplado que se ocupa de un gran número de nodos heterogéneos y poco fiables y se utiliza para crear aplicaciones de procesamiento de datos.

La arquitectura actual del sistema operativo en la nube se basa en el software de la plataforma. Los componentes de software simplemente se apilan juntos para proporcionar las funciones necesarias. La arquitectura es simplemente el sistema operativo combinado con el middleware de red. Los componentes de la red suelen ser la base de datos distribuida y las

almacenamiento y middleware para la comunicación de mensajes. Se utilizan para ocultar la distribución de los recursos informáticos subyacentes. A veces, la infraestructura puede realizar la optimización en función de diferentes tipos de aplicaciones. La plataforma subyacente virtualizada hace que el sistema operativo en la nube sea transparente mediante el uso de la migración en vivo de VM. Sin embargo, las máquinas virtuales pueden generar una gran sobrecarga, lo que lleva a una menor eficiencia. El soporte de la transmisión de datos en red tampoco es suficiente[31].

En los últimos años, la aparición y proliferación de contenedores ha mejorado el desarrollo de los sistemas operativos en la nube. Los contenedores utilizan la capacidad de aislamiento del sistema operativo existente. Esto puede reducir el número de abstracciones, mejorando así la eficiencia. Mediante el uso de contenedores como bloques de construcción básicos para el sistema operativo en la nube, el marco de gestión unificado puede admitir sistemas informáticos y sistemas de almacenamiento a una escala extremadamente grande. Los trabajos admitidos pueden ser una mezcla de varios tipos, como servicio y procesamiento por lotes. El lenguaje de descripción de recursos complejo y completo se puede utilizar para la asignación de tareas, la programación y la tolerancia a fallas [10-12,84-86].

En la pila de software del sistema operativo en la nube, los contenedores de aplicaciones son adecuados para hospedar microservicios. Ahora, la herramienta de gestión como Docker[87] se vuelve muy popular en la práctica. Con su archivo de compilación estándar y las API RESTful flexibles, la herramienta de administración puede ayudar en gran medida a mejorar la automatización del empaquetado, las pruebas y la implementación de software. Sin embargo, la implementación actual de contenedores carece de aislamiento de rendimiento y aislamiento de seguridad[88]. Por lo tanto, los contenedores generalmente se ejecutan en máquinas virtuales secundarias para el aislamiento del rendimiento. Esta redundancia obvia brinda nuevas oportunidades para llevar los contenedores un nivel más bajo en el sistema operativo mediante el uso del micro kernel o kernel personalizado para eliminar el problema del aislamiento del rendimiento [89-91].

La arquitectura del sistema operativo tradicional en la nube débilmente acoplado hace que su uso sea complejo e introduce una gran cantidad de trabajo redundante. Carece de percepción de la carga de trabajo de la aplicación y la política no puede notificar al componente subyacente de manera efectiva. Por lo tanto, la arquitectura debe considerar la integración vertical de SO para reducir los niveles de abstracción. Se pueden poner más puntos de percepción en el sistema operativo que pueden considerar las políticas de programación en el entorno distribuido y en la nube. Esto puede ayudar a construir un marco de gestión unificado y plano para la plataforma de computación en la nube. Dicha arquitectura se beneficia de la convergencia de funciones de cada componente, la optimización basada en la percepción de la carga de trabajo y la evolución ordenada. No es fácil desvincular la política del sistema operativo de un solo nodo monolítico tradicional. SO de nodo único

necesita la mejora de la estructura interna para mejorar la capacidad de percepción de la carga de trabajo de la aplicación. El mecanismo se puede implementar en el núcleo pequeño y proporcionar a los usuarios interfaces de políticas más personalizables. Y la investigación más avanzada incluye la distribución de datos.

ción, distribución de funciones y reutilización del espacio para reemplazar la reutilización del tiempo en el sistema operativo. Esto puede hacer que el sistema operativo de un solo nodo sea más escalable y flexible[92-95].

Como conclusión, el sistema operativo en la nube debe integrar verticalmente el sistema operativo de un solo nodo y el middleware de red para la percepción de la carga de trabajo de la aplicación. En la capa inferior, la tecnología de virtualización y los contenedores se pueden utilizar para admitir procesadores multinúcleo y heterogéneos. Además, el desacoplamiento del desarrollo de aplicaciones y el hardware debe admitirse como una caja de arena de alta eficiencia. Las tecnologías de micronúcleo se pueden utilizar para mejorar el aislamiento de los contenedores y la flexibilidad del sistema operativo de un solo nodo. En la capa superior, el marco de gestión de recursos unificado puede lograr los objetivos de implementación de trabajos mixtos, optimización de la programación global y percepción de la carga de trabajo de la aplicación[96-99]. El mashup de servicios se puede implementar mediante la administración de paquetes de aplicaciones. Para cada etapa de la evolución del sistema operativo en la nube, se introducen nuevas API. Con las nuevas API y su estandarización, las nuevas aplicaciones pueden impulsar la siguiente etapa de la evolución de la arquitectura del sistema operativo, ampliando la adopción de tecnologías en la nube.

4 Evolución del ecosistema de la nube

Para el desarrollo futuro del sistema operativo en la nube, el objetivo principal es respaldar el ecosistema de la nube y crear más aplicaciones. Esta sección estudiará la evolución del ecosistema de nubes. La evolución de los sistemas en la nube está estrechamente relacionada con la evolución de la tecnología [99-101]. La evolución del sistema operativo, como se mencionó anteriormente, este segundo capítulo mostrará cómo las tecnologías y la arquitectura del sistema operativo encajan en la evolución del sistema operativo en la nube. Con base en la observación de diferentes proveedores de nube y sus contrapartes de código abierto, la evolución del ecosistema de la nube se puede estudiar desde dos perspectivas diferentes. La primera se denomina evolución de las capas y está bastante relacionada con el desarrollo de la tecnología y está impulsada por las aplicaciones que emergen gradualmente en cada fase de la evolución del sistema operativo en la nube. El segundo está relacionado con la adopción de tecnologías en la nube.

Antes de sumergirnos en la discusión detallada, primero esbozamos brevemente la importancia de las API y su papel en la mejora del ecosistema. La tabla 1 muestra varios aspectos de cómo las API pueden mejorar el sistema operativo en la nube subyacente.

y las aplicaciones que se ejecutan sobre ellos. Las API pueden definir claramente el límite entre la plataforma del sistema y las aplicaciones. Eso es fundamental para crear aplicaciones sin preocuparse por la rápida evolución de las tecnologías subyacentes. El ecosistema de la nube y el sistema operativo no se pueden construir sin la introducción y evolución de las API.

Tabla 1. Función de las API para el ecosistema de la nube

Aspecto	Descripción
Productividad mejorada	Los diferentes componentes del sistema operativo en la nube y las aplicaciones pueden ser desarrollados de forma independiente por diferentes personas
Usabilidad mejorada	Los detalles de implementación están ocultos y los usuarios solo necesitan comprender la información expuesta
Funcionalidad reutilizar	Se pueden desarrollar funciones más avanzadas utilizando las API existentes; de esta manera también mejora la productividad
Mejor interoperabilidad	Aplicaciones desarrolladas con estándares Las API pueden ejecutarse en cualquier sistema que admita estas API
Crecimiento saludable	Se puede formar un ecosistema a mayor escala utilizando las API existentes, desarrollando y entregando nuevas API

4.1 Evolución del ecosistema de nubes en capas

Los ecosistemas de nubes parecen bastante proliferados últimamente. Y está más claro que las API principales ahora son relativamente estables. Sin embargo, hasta hace poco, los ecosistemas de nubes no estaban muy claros. Diferentes proveedores de nube tienen su propia definición de ecosistemas de nube. La evolución del ecosistema de la nube está bastante impulsada por las aplicaciones en su parte superior. Al principio, los proveedores de la nube solo querían brindar servicios a través de Internet, liberando a los usuarios y administradores de la implementación y el mantenimiento. El sistema operativo en la nube en ese momento tenía una apariencia muy vaga en el ecosistema. Por lo tanto, muchos de los practicantes pioneros de la nube hicieron todo lo posible para mantener los servicios de los desarrolladores para los usuarios finales. Por ejemplo, Google AppEngine ayuda a los desarrolladores a ejecutar sus servicios de atención al cliente dentro de la infraestructura de la nube de Google. Por lo tanto, las empresas emergentes no necesitan preocuparse por toda la infraestructura subyacente. Solo necesitan preocuparse por la lógica de la aplicación que puede atraer a los usuarios finales. Amazon hizo lo mismo con un nivel mucho más bajo de interfaces de programación. Sin embargo, en esta etapa de la evolución del sistema operativo en la nube, no existe una metodología clara para definir las API centrales y las API de nivel de aplicación. Los esfuerzos anteriores del sistema operativo en la nube solo proporcionaron los servicios que ya estaban dentro de los proveedores. De todos modos, esta etapa al menos reveló algo sobre la gran evolución posterior del sistema operativo en la nube.

Con el desarrollo del ecosistema de la nube, y especialmente debido a las aplicaciones que se ejecutan sobre la infraestructura de la nube, los servicios fundamentales subyacentes se volvieron muy importantes. Todos los profesionales ahora tienen una imagen más clara de cómo debería ser el ecosistema. La gente se da cuenta de que hay al menos tres niveles para muchos componentes en cualquier ecosistema de nube. Cada nivel sirve para diferentes propósitos. En el primer nivel, los recursos informáticos subyacentes se pueden virtualizar. La razón es bastante simple: tenemos que usar algunas formas de direccionamiento indirecto para ocultar los recursos físicos. Y luego los recursos físicos pueden evolucionar por sí mismos sin perturbar las pilas de software que se ejecutan en la parte superior. En este nivel de ecosistema, podemos ver componentes de hardware virtualizados que incluyen máquinas virtuales, redes virtuales[100] y almacenamiento virtualizado. Este nivel también tiene su propia evolución al introducir contenedores y servidores de computación perimetral. Separar este nivel de los niveles superiores es muy importante para la evolución del ecosistema, ya que los desarrolladores ahora no luchan con detalles de implementación y administración de hardware que consumen mucho tiempo y son propensos a errores. El primer nivel de virtualización crea principalmente una interfaz muy conveniente para la administración del sistema y la implementación de aplicaciones. Llamamos a esta capa la capa de virtualización.

Además de la capa de virtualización, el sistema operativo en la nube debe proporcionar soporte para los desarrolladores. De hecho, esta capa tiene su propio historial de desarrollo independiente de la capa de virtualización. De hecho, algunas de las aplicaciones en la nube no necesitan ejecutarse en el entorno virtualizado. Por lo tanto, el procesamiento de datos por lotes o el procesamiento de datos continuos necesita algunos marcos de trabajo de cómputo para que los programas se construyan de manera más productiva. Y también, para alojar servicios web en el entorno de la nube sin tener en cuenta los recursos informáticos virtuales o físicos, el sistema operativo de la nube debe exponer las interfaces necesarias sin revelar los servidores virtuales o físicos. Para admitir tales aplicaciones en la nube, los proveedores de la nube han creado el sistema de archivos distribuidos y varios marcos informáticos para desarrolladores. Las interfaces de programación deben ser abstractas para que los programas no necesiten notar la ubicación del alojamiento. Con la evolución del sistema operativo en la nube, ahora se considera que los servicios de almacenamiento se encuentran en la capa de virtualización. Y los marcos de programación se pueden considerar en un nivel superior de ecosistema. Esta capa se puede llamar capa de soporte de servicios. Hay otro componente muy importante en esta capa, a saber, la gestión de tareas. En cuanto a la compatibilidad con la ejecución de aplicaciones en la nube, la gestión de tareas se utilizará en la capa de virtualización.

ing varios tipos de cargas de trabajo. Las cargas de trabajo incluyen servicios web, procesamiento por lotes, monitoreo en tiempo real y otros tipos de aplicaciones. Dado que la escala del sistema de computación en la nube suele ser bastante grande, la gestión de tareas es muy importante para garantizar el uso eficiente de los recursos informáticos y mejorar el rendimiento general del sistema. La capa de soporte de servicios se considera como el servicio de aplicación fundamental en el ecosistema de la nube.

artículo

La capa de servicio de la aplicación es la capa final que se encuentra sobre la capa de soporte del servicio. La capa de aplicación proporciona servicios a los usuarios finales. Por lo general, las aplicaciones en el sistema operativo tradicional no exponen ninguna interfaz de programación para los programadores. Sin embargo, la mayoría de las aplicaciones en la nube lo hacen porque, de esta manera, los desarrolladores de aplicaciones pueden mezclar varios servicios para brindar un nuevo servicio a los usuarios finales. Es bastante común en el entorno de la nube que los servicios se juntan para construir aplicaciones más complejas y de mayor nivel. Y algunos de los servicios son tan comunes e importantes que pueden formar otro nivel de API centrales. Por ejemplo, los servicios de mensajería y autenticación de usuarios se pueden considerar como API de capa de aplicación para casi todas las aplicaciones.

Del análisis anterior, se puede llegar a la conclusión de que la arquitectura del sistema operativo en la nube debe ser and now es bastante similar al del sistema operativo local. El sistema debe dividirse en diferentes capas y cada capa tiene su propio propósito. El sistema operativo local tiene la capa de hardware, el sistema operativo y la capa de tiempo de ejecución, y la capa de aplicación de usuario. Cloud OS tiene cada contraparte. Para el desarrollo futuro del sistema operativo en la nube, el servicio en capas es inevitable. El trabajo crítico que debe hacerse aquí es separar cada capa claramente y definir la interfaz entre las capas con precisión. Esta es una práctica común del sistema y puede reducir toda la complejidad del sistema. Además, cada capa puede tener su propia evolución sin perturbar demasiado a las otras capas. Por ejemplo, los contenedores y la informática perimetral se pueden adoptar en la capa de virtualización. Debido al sistema claramente estratificado, la introducción de nuevas tecnologías puede adaptarse rápidamente a todo el marco. Además, dicha introducción podría tener algunos otros beneficios. Por ejemplo, con la introducción de contenedores, el sistema operativo en la nube puede enmascarar los recursos informáticos de hardware heterogéneos sin demasiada sobrecarga como las máquinas virtuales. Por lo tanto, las API de administración de servicios y las API de administración de recursos se pueden desacoplar mediante el uso de contenedores.

Las API de administración de recursos pueden manejar la heterogeneidad de diferentes recursos informáticos y exponer la

recursos informáticos como contenedores. Las API de administración de servicios pueden administrar y programar estos contenedores. Los contenedores se pueden programar para los recursos informáticos apropiados para lograr un alto rendimiento y eficiencia.

Aunque la separación de estas tres capas es bastante clara hoy en día, no existe un límite claro entre ellas. Y a menudo, no es la capa inferior la que ayuda a mejorar las capas superiores. Por el contrario, la capa de aplicaciones a menudo desempeña el papel de impulsor del ecosistema. Con más y más aplicaciones en la nube prevaleciendo en los servicios para usuarios finales, la infraestructura y la plataforma tienen que adaptarse a tales tendencias y optimizar para ejecutar la aplicación de manera más eficiente.

En resumen, el ecosistema de la nube actual funciona de manera similar a los sistemas operativos tradicionales. Al exponer múltiples capas de interfaces, los desarrolladores de aplicaciones y de sistemas pueden hacer contribuciones al ecosistema. El ímpetu sigue siendo la compensación entre la mejora de la experiencia del usuario en el nivel de la aplicación y la mejora de la eficiencia en el nivel del sistema. Por lo tanto, para la evolución futura del ecosistema en la nube, el modelo en capas del sistema operativo en la nube sigue siendo válido.

La interfaz proliferará y se agregarán al ecosistema más interfaces de administración y programación específicas de aplicaciones. Las API principales relacionadas con la infraestructura y la gestión de recursos serán mucho más estables en el ecosistema. Con el modelo en capas que desacopla las diferentes funciones en el sistema operativo en la nube, la evolución de cada capa puede ser relativamente independiente.

4.2 Evolución de la adopción de tecnología en la nube

A pesar de la nube pública, las tecnologías utilizadas en el sistema de nube han llamado mucho la atención de la industria tradicional. La facilidad de uso, la capacidad de procesamiento de datos a gran escala y la característica de tolerancia a fallas pueden beneficiar la infraestructura de TI actual de muchos institutos. De hecho, la adopción de tecnologías en la nube fue lenta en las primeras etapas. Esto se debe principalmente a que el ecosistema, incluso para la nube pública, no estaba maduro. Como la implementación de la tecnología habilitadora es bastante difícil, no había una implementación de referencia pública en ese momento. Además de las dificultades, una de las principales razones del problema de disponibilidad de la implementación pública es la falta de interfaces estandarizadas, es decir, API.

Más tarde, con el desarrollo del ecosistema de nube pública, las interfaces de programación para el sistema operativo en la nube se vuelven cada vez más maduras. Todos los proveedores de nube pública exponen interfaces de programación similares que cubren la gestión de infraestructura y la gestión de tareas.

y algunas interfaces de programación críticas a nivel de aplicación. Y se han realizado muchos esfuerzos para implementar las API centrales relativamente maduras y estables. Dos proyectos de código abierto muy importantes, Hadoop y Open Stack, comenzaron a tener una gran influencia en la adopción de tecnologías en la nube. Ambos pueden considerarse como la encarnación de los sistemas operativos de nube pública. OpenStack, que es bastante similar a la virtualización de infraestructura en la nube de Amazon, se puede utilizar para crear la infraestructura virtualizada, incluida la virtualización informática, la virtualización de redes y la virtualización de almacenamiento. Hadoop ha implementado una gran cantidad de infraestructura de procesamiento de datos para mejorar el rendimiento, la programabilidad y la tolerancia a fallas de los clústeres a gran escala. Estos dos proyectos de código abierto tienen una gran cantidad de componentes, lo que permite a los usuarios adoptar la tecnología de la nube con mayor facilidad. Está bastante claro que, con el tiempo, las interfaces de programación estables han hecho posible exponer la infraestructura de la nube utilizada internamente para que esté disponible públicamente.

Sin embargo, la adopción de la tecnología actual aún se limita a las tecnologías de nube generales. Algunas aplicaciones en la nube pueden necesitar consideraciones especiales. Por ejemplo, muchas de las aplicaciones informáticas de alto rendimiento actuales no se benefician de la flexibilidad de la infraestructura de la nube. Por lo tanto, las tecnologías generales actualmente implementadas podrían no ser apropiadas. En la evolución futura, la tecnología de la nube debe integrarse en las aplicaciones. La infraestructura debe percibir las aplicaciones y hacer la adaptación adecuada para ejecutar las aplicaciones de manera más eficiente. Hemos visto que la tecnología de la nube mejora el servicio de TI para muchas aplicaciones existentes. El sistema operativo en la nube hará más abstracciones en capas de los servicios subyacentes. Con las interfaces de programación mejoradas y su implementación, las tecnologías de nube serán más convenientes para construir servicios de nube especializados. Los desarrolladores de aplicaciones y el especialista en TI pueden usar la pila de software del sistema operativo en la nube disponible públicamente para construir una nube interna.

mientras que el servicio de nube pública no puede satisfacer sus demandas.

5 Metodología de abstracción de la API de Cloud OS

Según el estudio anterior, las API son fundamentales para impulsar la evolución del futuro sistema operativo en la nube. La abstracción de API será el primer paso para implementar cualquier sistema operativo en la nube y también es la base para crear aplicaciones en la nube. Hay algunos principios para hacer abstracción de API en la nube.

1) La abstracción de las API debe ser compatible con la práctica actual de la nube. Este principio muestra el respeto por los esfuerzos realizados actualmente y puede ajustarse a las expectativas de los usuarios. Muchas aplicaciones han demostrado la validez del actual sistema operativo en la nube.

2) Las API principales deben ser estables. Hemos visto muchas aplicaciones en la nube que requieren diferentes soportes subyacentes. Sin embargo, las API principales deben ser estables. Esto ayudará a construir el ecosistema de la nube. Los desarrolladores tienen una base relativamente sólida para construir sus aplicaciones. De lo contrario, si las API cambian rápidamente, el conocimiento aprendido pronto quedará obsoleto y dañará las aplicaciones en la nube existentes.

3) Las API deben estar en capas y cubrir suficientes demandas. Como se analizó anteriormente, el ecosistema necesita abstracción de API en capas. Cada capa necesita servir para diferentes propósitos. De hecho, cada capa puede tener su propio ecosistema y puede evolucionar por sí misma. De esta manera, las API se pueden hacer prolíficas para cubrir suficientes demandas sin demasiada sobrecarga.

4) Si bien las API, especialmente las API centrales, deben mantenerse relativamente estables, pueden y deben tener la evolución necesaria para adoptar la mejora de la tecnología, así como los nuevos requisitos planteados por las aplicaciones en la nube. Se introducirá nuevo hardware en el ecosistema de la nube, y el sistema operativo de la nube debe admitir dicho hardware para ejecutar aplicaciones de manera más eficiente.

Con base en los principios anteriores, ahora está claro cómo podemos definir las API del sistema operativo en la nube. Para compatibilidad con los sistemas operativos de nube actuales, las API se pueden obtener de los profesionales actuales del sistema de nube, incluidas las nubes públicas y sus contrapartes de código abierto. Por lo general, las API definidas por estas dos comunidades son similares porque las versiones de código abierto se derivan de los servicios públicos. Las versiones de código abierto a menudo evolucionan más rápidamente y las nuevas funciones se pueden fusionar rápidamente con la implementación de código abierto. Los proveedores de la nube con otras consideraciones no pueden hacer el cambio tan rápido. Todas estas API son la fuente de la que puede aprender un nuevo sistema operativo en la nube. Las API se dividirán en diferentes categorías en función de sus funciones. Además, las categorías deben colocarse en varias capas en función de sus distancias a los recursos informáticos subyacentes.

Después del trabajo de categorización y estratificación, se deben tomar consideraciones muy cuidadosas de la estabilidad relativa de todas las API, así como de sus funciones. El

Las API principales no deben estar vinculadas a ninguna aplicación específica y deben ser estables. Por lo tanto, siempre se deben considerar dos categorías de API como API centrales. Uno

está relacionado con la virtualización de infraestructura y re administración de fuentes como máquinas virtuales, contenedores, almacenamiento virtual y redes virtuales. El otro está relacionado con la gestión de tareas, como la supervisión de la carga de trabajo, la granularidad diferente de la gestión de trabajos, el programador de tareas basado en diferentes cargas de trabajo, etc. También se deben tener en cuenta algunos marcos de programación, como el procesamiento por lotes, el procesamiento de transmisión, el análisis de gráficos[101-102] y el procesamiento de consultas distribuidas, porque se usan tan ampliamente que se encuentran entre las aplicaciones más importantes, incluido el marco de aplicación que puede ayudar a construir el ecosistema de la nube. Deben incluirse las API compatibles con el hardware más reciente, ya que las aplicaciones las requieren y pueden beneficiar la adopción de tecnologías en la nube mientras se crea una infraestructura de nube interna especializada.

En resumen, se pueden definir las API principales del sistema operativo en la nube, así como las API necesarias para crear aplicaciones a partir de las nubes públicas existentes y sus equivalentes de código abierto. Luego, el ecosistema se puede construir a partir de estas API. Las API deben estandarizarse para que las aplicaciones se puedan desarrollar más rápidamente sin preocuparse demasiado por la implementación de la API.

En la siguiente sección, brindaremos algunas prácticas ejemplares en la nube que primero definen las API y luego construyen un ecosistema a su alrededor.

6 Práctica de SO en la nube impulsada por API

Por API-driven, queremos decir que las API se definen primero y luego se desarrolla un sistema operativo en la nube que implementa estas API. Esa forma es posible porque ha pasado casi una década desde la llegada de la computación en la nube y la gente ha adquirido suficiente experiencia sobre el concepto y las aplicaciones del mundo real. Además, de esta manera se pueden producir API de la mayor comodidad y sistemas que puedan respaldarlas de manera eficiente.

6.1 Definición de las API principales

En la Sección 5, hemos señalado algunos principios para definir las API del sistema operativo en la nube. De acuerdo con estos criterios y la práctica de los servicios de nube pública y los proyectos de nube de código abierto, definimos cinco categorías de API de la siguiente manera.

- Las API relacionadas con contenedores incluyen Create, Start, Kill, Delete según lo define la iniciativa de contenedor abierto y otras que se definen automáticamente, como List (para listar todos los contenedores de un determinado usuario), Watch (para obtener el estado de un determinado contenedor), Migrar y Detener. Estos

Las API hacen posible que los usuarios creen y manipulen aplicaciones basadas en contenedores. • Se proporcionan API relacionadas con máquinas virtuales para que los usuarios creen y controlen máquinas virtuales. Virtual

Las máquinas proporcionan una forma de utilizar los recursos de manera más flexible y eficiente. • Se proporcionan API relacionadas con la programación para que los usuarios envíen trabajos y supervisen el proceso de ejecución. Es deber de la programación garantizar la calidad deseada en la ejecución del trabajo. • Las API relacionadas con el almacenamiento se utilizan para que los usuarios guarden su contenido. Los tipos de almacenamiento admitidos incluyen almacenamiento de objetos, almacenamiento de archivos y almacenamiento de bloques.

- Las API relacionadas con la gestión de operaciones ofrecen funcionalidades como la implementación y gestión de recursos, la gestión de la configuración, la supervisión del sistema, la auditoría del sistema y la gestión de la seguridad. Estas funcionalidades son esenciales para el correcto funcionamiento del sistema.

6.2 Arquitectura del sistema operativo en la nube

Como referencia, diseñamos un sistema operativo en la nube que implementa las API mencionadas anteriormente bajo el patrocinio del Programa Nacional de Investigación y Desarrollo Clave de China. La arquitectura de nuestro sistema operativo en la nube se muestra en

Figura 2.

El sistema operativo consta de varias capas, a saber, recursos físicos, núcleo del sistema operativo, servicios del sistema, soporte del ecosistema y aplicaciones. La capa de recursos físicos proporciona varios tipos de recursos, incluidos CPU, memoria, almacenamiento, red, etc. El kernel del sistema operativo, que se ubica entre la capa de recursos físicos y la capa de servicios del sistema, cumple la tarea de abstracción de recursos y provisión de servicios. La capa de servicios del sistema proporciona dichos servicios y las API correspondientes, como servicios de red, computación en memoria a gran escala, almacenamiento de datos, computación elástica, etc., en función de las API enviadas por el kernel del sistema operativo. Además de los servicios del sistema, hay otras instalaciones (p. ej., administración de cuentas, autorización de usuarios, facturación, servicios de contenedores, orquestación de recursos y depósito de VM/contenedores) que son necesarias para construir un ecosistema. Estas instalaciones también proporcionan sus propias API. La capa más superior es la capa de aplicaciones que ofrece capacidades tales como procesamiento de big data, computación científica, computación gráfica, aprendizaje profundo, etc. Tenga en cuenta que la aplicación aquí se llama desde la perspectiva del sistema operativo y es posible que no involucre la lógica comercial.

En nuestro sistema, nos enfocamos principalmente en las funciones clave

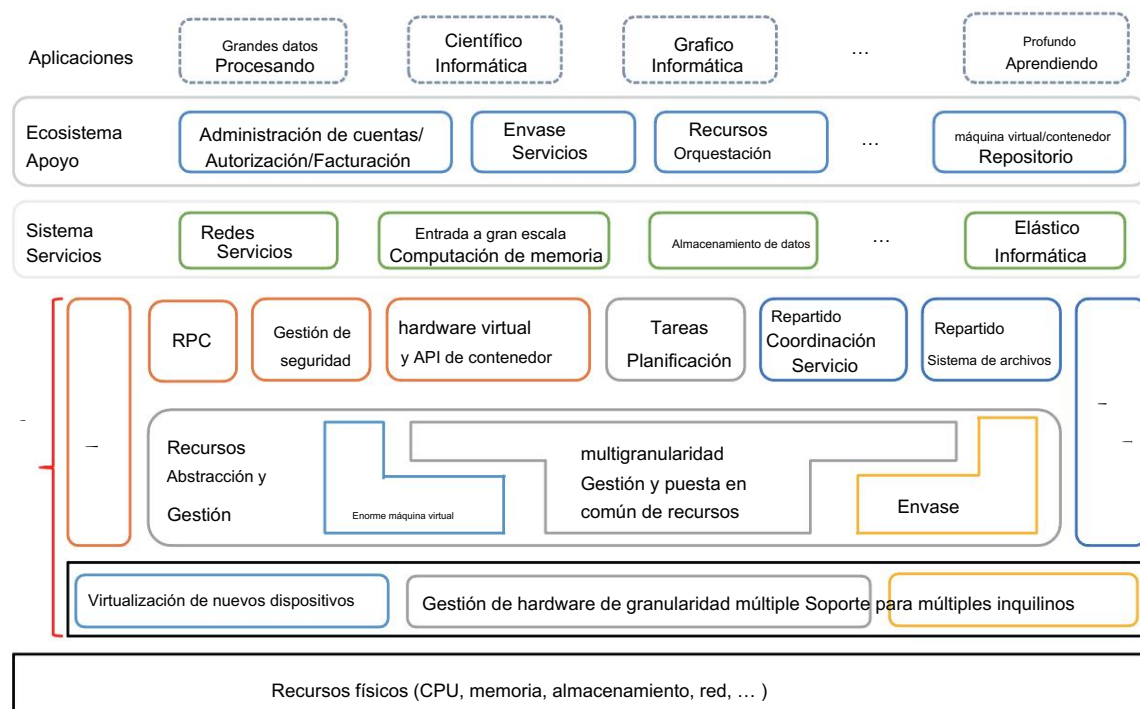


Figura 2. Se desarrolló la arquitectura general del sistema operativo en la nube. Mgmt significa gestión.

en el núcleo del sistema operativo. El detalle de los mismos se explicará en el siguiente subapartado.

6.3 Tecnologías clave

Para lograr el propósito de alta eficiencia y facilidad de uso, la implementación de nuestro sistema operativo en la nube presta atención a las siguientes tecnologías.

Tecnología de contenedores. Los contenedores brindan una forma liviana para que los usuarios desarrollen y ejecuten aplicaciones distribuidas y se consideran una dirección futura. Aunque hay muchos contenedores disponibles (por ejemplo, Docker y LXC), están lejos de ser perfectos. Nuestro sistema operativo investigará formas de mejorar el aislamiento de contenedores y realizar una migración en vivo sin degradación del rendimiento.

Además, dado que muchos inquilinos consumirán los servicios de contenedores en entornos de nube y que la computación ha ampliado su alcance desde el centro de datos central hasta el perímetro, también estudiaremos cómo admitir mejor la función multiinquilino y la computación perimetral por parte de los contenedores.

Enorme tecnología VM. La máquina virtual tradicional encarna la idea de la partición de recursos, lo que permite a los usuarios manipular los recursos de una manera más detallada. Además de este paradigma, es necesario considerar escenarios donde se combinan grandes cantidades de recursos y se presentan como una poderosa máquina virtual (llamada VM enorme) junto con la llegada de nuevos dispositivos (por ejemplo, GPU, RDMA y

FPGA) y aplicaciones (por ejemplo, computación científica, procesamiento de big data). Los temas a estudiar incluyen cómo realizar la virtualización de nuevos dispositivos a fin de sentar una buena base para la construcción de máquinas virtuales enormes, cómo construir una máquina virtual enorme que pueda soportar más de 500 núcleos de computación virtual y hasta 2 TB de memoria, y cómo utilizar los recursos programación para mejorar el rendimiento.

Tecnología de programación de tareas. La programación de tareas se ejecuta sobre la agrupación y gestión de recursos de granularidad múltiple. Es deber de la programación de tareas garantizar tanto la utilización de los recursos como la calidad de los servicios. Dado que se trata de recursos a gran escala y las tareas que se ejecutan son de diferentes características, no es una tarea fácil realizar la programación de tareas. Aquí nos enfocamos en la programación de cargas de trabajo híbridas, la escalabilidad y la tolerancia a fallas del propio programador, y la adaptación de marcos informáticos típicos con entornos de computación en la nube.

Tecnología de sistema de archivos distribuidos. Al igual que el sistema de archivos para el sistema operativo tradicional, los sistemas de archivos distribuidos desempeñan un papel clave en la computación en la nube, ya que simplifican el acceso al almacenamiento. Dado que hay muchos sistemas de archivos distribuidos (p. ej., GFS/HDFS, Gluster, Ceph) disponibles, cambiamos nuestro enfoque de la escalabilidad a la confiabilidad, con el objetivo de lograr una mejor confiabilidad y eficiencia de almacenamiento. Utilizamos el código de borrado a gran escala para lograr el propósito. Para garantizar el rendimiento de lectura/escritura, se introduce la computación paralela para la codificación y decodificación de datos.

Tecnología de Despliegue de Recursos y Gestión de Centros de Datos. Para la implementación de recursos, presentamos un método para identificar las dependencias entre múltiples aplicaciones y diseñar una herramienta de implementación basada en Puppet y habilitada para caché, con la cual los recursos se pueden implementar de manera rápida y eficiente en energía. Para la administración del centro de datos, monitoreamos los estados de tiempo de ejecución del sistema y diseñamos un algoritmo basado en aprendizaje profundo para identificar y localizar fallas y realizar una evaluación de seguridad. De esta forma, la gestión del centro de datos se vuelve inteligente.

Además de las tecnologías clave mencionadas anteriormente, también investigamos el problema de las pruebas del sistema, con el fin de brindar una evaluación completa de todo el sistema en términos de cumplimiento de API y rendimiento de la interfaz. El resultado es una prueba automática basada en contrato semántico de interfaz.

esquema de ing que puede proteger la diferencia en los formatos de interfaz y reducir el costo de prueba.

En resumen, la implementación de nuestro sistema operativo muestra la posibilidad de construir un ecosistema alrededor de algunas API. En esta implementación, solo nos enfocamos en las funciones principales del kernel del sistema operativo y las API correspondientes. Todavía se puede mejorar, por ejemplo, introduciendo funciones y/o servicios más avanzados sobre el kernel del sistema operativo para aumentar la eficiencia del desarrollo de aplicaciones.

7. Conclusiones

En este documento, brindamos un breve estudio de la evolución de las tecnologías y los ecosistemas relacionados con el sistema operativo en la nube. Está muy claro que las API juegan un papel central en la evolución del sistema operativo en la nube. Las API son la abstracción de la ONU infraestructura informática avanzada e implementar los requisitos de las aplicaciones en la nube. Con la gestión interna de trabajos y recursos, el sistema operativo en la nube intenta ejecutar aplicaciones de manera eficiente. La creación de ecosistemas en la nube basados en las API del sistema operativo cuidadosamente seleccionadas es fundamental para que el sistema sea productivo y saludable.

Referencias

- [1] Armbrust M, Fox A, Griffith R et al. Una vista de la computación en la nube. *Comunicaciones de la ACM*, 2010, 53(4): 50-58.
- [2] Tanenbaum AS, Woodhull A S. *Diseño e implementación de sistemas operativos* (3.ª edición). Pearson, 2006.
- [3] Auslander MA, Larkin DC, Scherr A L. La evolución del sistema operativo MVS. *Revista IBM de Investigación y Desarrollo*, 1981, 25(5): 471-482.
- [4] Deitel HM, Deitel PJ, Choffnes D. *Sistemas operativos*. Pearson/Prentice Hall, 2004.
- [5] Bic LF, Shaw A C. *Principios de sistemas operativos*. Aprendiz Sala, 2003.
- [6] Silberschatz A, Galvin PB, Gagne G. *Sistema operativo Conceptos*. John Wiley & Sons Ltd., 2008.
- [7] Hu T H. Una prehistoria de la nube. *Prensa del MIT*, 2016.
- [8] Mell P, Grance T. SP800-145. La definición del NIST de computación en la nube. *Comunicaciones de la ACM*, 2010, 53(6): 50.
- [9] Zheng W. Introducción a la nube de Tsinghua. *Science China Information Sciences*, 2010, 53(7): 1481-1486.
- [10] Hindman B, Konwinski A, Zaharia M et al. Mesos: una plataforma para compartir recursos de forma detallada en el centro de datos. en *profesional Conferencia USENIX sobre diseño e implementación de sistemas en red*, 31 de marzo al 1 de abril de 2013, pp.429-483.
- [11] Schwarzkopf M, Konwinski A, Abd-El-Malek M et al. Omega: programadores escalables y flexibles para grandes clústeres de cómputo. En *Proc. ACM European Conference on Computer Systems*, abril de 2013, pp.351-364.
- [12] Verma A, Pedrosa L, Korupolu M et al. Gestión de clústeres a gran escala en Google con Borg. En *Proc. la 10ª Conferencia Europea sobre Sistemas Informáticos*, abril de 2015, pp.18:1- 18:17.
- [13] Dean J, Ghemawat S. MapReduce: procesamiento de datos simplificado en clústeres grandes. En *Proc. el 6º Simposio sobre Diseño e Implementación de Sistemas Operativos*, diciembre de 2004, pp.137-150.
- [14] Ghemawat S, Gobioff H, Leung S T. El sistema de archivos de Google. *Revisión de sistemas operativos ACM SIGOPS*, 2003, 37(5): 29-43.
- [15] Chang F, Dean J, Ghemawat S et al. Bigtable: Un sistema de almacenamiento distribuido para datos estructurados. *ACM Transactions on Computer Systems (TOCS)*, 2008, 26(2): 205-218.
- [16] Baker J, Bond C, Corbett J et al. Megastore: proporciona almacenamiento escalable y de alta disponibilidad para servicios interactivos. En *Proc. la 5ª Conferencia Biental sobre Investigación de Sistemas de Datos Innovadores*, enero de 2011, pp.223-234.
- [17] Corbett JC, Dean J, Epstein M et al. Spanner: la base de datos distribuida globalmente de Google. *ACM Transactions on Computer Systems (TOCS)*, 2013, 31(3): 8:1-8:22.
- [18] Yu Y, Isard M, Fetterly D et al. DryadLINQ: un sistema para computación paralela de datos distribuidos de propósito general que utiliza un lenguaje de alto nivel. En *Proc. el 8º Simposio USENIX sobre Diseño e Implementación de Sistemas Operativos*, diciembre de 2008, pp.1-14.
- [19] Isard M, Budiu M, Yu Y et al. Dryad: programas paralelos de datos distribuidos a partir de bloques de construcción secuenciales. *Revisión de sistemas operativos ACM SIGOPS*, 2007, 41(3): 59-72.
- [20] Zaharia M, Chowdhury M, Das T et al. Conjuntos de datos distribuidos resilientes: una abstracción tolerante a fallas para la computación en clúster en memoria. En *Proc. la 9ª Conferencia USENIX sobre diseño e implementación de sistemas en red*, abril de 2012, pp.141-146.
- [21] Power R, Li J. Piccolo: Creación de programas rápidos y distribuidos con tablas particionadas. En *Proc. el 9º Simposio USENIX sobre Diseño e Implementación de Sistemas Operativos*, octubre de 2010, pp.293-306.
- [22] Melnik S, Gubarev A, Long JJ et al. Dremel: análisis interactivo de conjuntos de datos a escala web. *Comunicaciones de la ACM*, 2011, 54(6): 114-123.
- [23] Peng D, Dabek F. Procesamiento incremental a gran escala utilizando transacciones y notificaciones distribuidas. En *Proc. el 9º Simposio USENIX sobre Diseño e Implementación de Sistemas Operativos*, octubre de 2010, pp.251-264.
- [24] Neumeyer L, Robbins B, Nair A et al. S4: plataforma informática de flujo distribuido. En *Proc. la 10ª Conferencia Internacional IEEE sobre Talleres de Minería de Datos*, diciembre de 2010, pp.170-177.

- [25] Viglas S, Naughton J F. Optimización de consultas basada en tasas para fuentes de información de transmisión. En Proc. ACM SIGMOD International Conference on Management of Data, junio de 2002, pp.37-48.
- [26] Shen H, Zhang Y. Detección aproximada mejorada de duplicados para flujos de datos sobre ventanas deslizantes. Revista de informática y tecnología, 2008, 23(6): 973-987.
- [27] Li Y, Chen FH, Sun X et al. Gestión de recursos autoadaptable para clústeres compartidos a gran escala. Revista de informática y tecnología, 2010, 25(5): 945-957.
- [28] Hunt P, Konar M, Junqueira FP et al. ZooKeeper: Coordinación sin esperas para sistemas a escala de Internet. En Proc. Conferencia técnica anual de USENIX, junio de 2010.
- [29] Ongaro D, Ousterhout J. En busca de un algoritmo de consenso comprensible. En Proc. Conferencia técnica anual de USENIX, junio de 2014, pp.305-319.
- [30] Lamport L. Paxos simplificado. ACM SIGACT News, 2001, 32(4): 18-25.
- [31] Barham P, Dragovic B, Fraser K et al. Xen y el arte de la virtualización. Revisión de sistemas operativos ACM SIGOPS, 2003, 37(5): 164-177.
- [32] Ben-Yehuda M, Day MD, Dubitzky Z et al. El proyecto tur tles: Diseño e implementación de virtualización anidada. En Proc. la 9ª Conferencia USENIX sobre Diseño e Implementación de Sistemas Operativos, octubre de 2010, pp.423-436.
- [33] Xiao Z, Song W, Chen Q. Asignación dinámica de recursos utilizando máquinas virtuales para entornos de computación en la nube. Transacciones IEEE en sistemas paralelos y distribuidos, 2013, 24(6): 1107-1117.
- [34] Kivity A, Laor D, Costa G et al. OSv: optimización del sistema operativo para máquinas virtuales. En Proc. Conferencia técnica anual de USENIX, junio de 2014, pp.61-72.
- [35] Ren S, Tan L, Li C et al. Samsara: reproducción determinística eficiente en entornos multiprocesador con extensiones de virtualización de hardware. En Proc. Tecnología anual de USENIX Conferencia técnica, junio de 2016, pp.551-564.
- [36] Chen H, Wang X, Wang Z et al. DMM: un modelo de asignación de memoria dinámica para máquinas virtuales. Science China Information Sciences, 2010, 53(6): 1097-1108.
- [37] Zhao X, Yin J, Chen Z et al. vSpec: especialización en sistema operativo adaptable a la carga de trabajo para máquinas virtuales en computación en la nube. Science China Information Sciences, 2016, 59(9): 92-105.
- [38] Wang X, Sun Y, Luo Y et al. Paravirtualización de memoria dinámica transparente para el sistema operativo huésped. Science China Information Sciences, 2010, 53(1): 77-88.
- [39] Lu L, Zhang Y, Do T et al. Desenredo físico en un sistema de archivos basado en contenedores. En Proc. el 11º Simposio USENIX sobre diseño e implementación de sistemas operativos, octubre de 2014, págs. 81-96.
- [40] Arnavot S, Trach B, Gregor F et al. SCONe: Contenedores Linux seguros con Intel SGX. En Proc. Simposio USENIX sobre diseño e implementación de sistemas operativos, noviembre de 2016, pp.689-704.
- [41] Banga G, Druschel P, Mogul J C. Contenedores de recursos: una nueva instalación para la gestión de recursos en sistemas de servidores. En Proc. Simposio USENIX sobre diseño e implementación de sistemas operativos, febrero de 1999, pp.45-58.
- [42] Pedro GL, Alberto M, Dick E et al. Computación centrada en el borde: visión y desafíos. ACM SIGCOMM Computer Communication Review, 2015, 45 (5): 37-42.
- [43] Shi W, Cao J, Zhang Q et al. Computación perimetral: visión y desafíos. IEEE Internet of Things Journal, 2016, 3(5): 637-646.
- [44] Dragojević A, Narayanan D, Castro M et al. FaRM: memoria remota rápida. En Proc. Simposio USENIX sobre diseño e implementación de sistemas en red, abril de 2014, pp.401-414.
- [45] Mitchell C, Geng Y, Li J. Uso de lecturas RDMA unilaterales para crear un almacén de valor clave rápido y eficiente en la CPU. En Proc. Conferencia técnica anual de USENIX, junio de 2013, pp.103-114.
- [46] José J, Subramoni H, Luo M et al. Diseño de Memcached en interconexiones compatibles con RDMA de alto rendimiento. En Proc. Conferencia Internacional sobre Procesamiento Paralelo, septiembre de 2011, pp.743-752.
- [47] Greenberg A, Hamilton JR, Jain N et al. VL2: una red de centro de datos escalable y flexible. ACM SIGCOMM Computer Communication Review, 2009, 39(6): 51-62.
- [48] Paraíso F, Haderer N, Merle P et al. Una infraestructura PaaS multinube federada. En Proc. la 5ª Conferencia Internacional IEEE sobre Computación en la Nube, junio de 2012, pp.392-399.
- [49] Eguro K, Venkatesan R. FPGA para computación en la nube confiable. En Proc. la 22.ª Conferencia internacional sobre aplicaciones y lógica programable en campo, agosto de 2012, págs. 63-70.
- [50] Hutchings BL, Franklin R, Carver D. Asistencia en la detección de intrusos en la red con hardware reconfigurable. En Proc. el 10º Simposio IEEE sobre máquinas informáticas personalizadas programables en campo, abril de 2002, pp.111-120.
- [51] Chalamalasetti SR, Lim K, Wright M et al. Un dispositivo FPGA Memcached. En Proc. Simposio internacional ACM/SIGDA sobre matrices de puertas programables en campo, febrero de 2013, págs. 245-254.
- [52] Huang M, Wu D, Yu CH et al. Programación y soporte de tiempo de ejecución para la implementación del acelerador FPGA blaze a escala del centro de datos. En Proc. Simposio de ACM sobre computación en la nube, octubre de 2016, pp.456-469.
- [53] Wang XM, Thota S. Una arquitectura de comunicación eficiente en recursos para multiprocesadores de chip en FPGA. Revista de informática y tecnología, 2011, 26(3): 434-447.
- [54] Dong Y, Xue M, Zheng X et al. Aumento del rendimiento de la virtualización de la GPU con tablas de páginas paralelas híbridas. En Proc. Conferencia técnica anual de USENIX, julio de 2015, pp.517-528.
- [55] Zhang K, Chen R, Chen H. Análisis estructurado de gráficos compatible con NUMA. En Proc. el 20º Simposio ACM SIGPLAN sobre Principios y Prácticas de la Programación Paralela, febrero de 2005, pp.183-193.
- [56] Mao Y, Kohler E, Morris R T. Astucia de caché para un almacenamiento rápido de clave-valor multinúcleo. En Proc. Conferencia europea ACM sobre sistemas informáticos, abril de 2012, pp.183-196.
- [57] Tu S, Zheng W, Kohler E et al. Transacciones rápidas en bases de datos en memoria multinúcleo. En Proc. Simposio de ACM sobre principios de sistemas operativos, noviembre de 2013, pp.18-32.
- [58] Zhang G, Horn W, Sanchez D. Explotación de la conmutatividad para reducir el costo de las actualizaciones de datos compartidos en sistemas coherentes con caché. En Proc. Simposio internacional IEEE/ACM sobre microarquitectura, diciembre de 2015, pp.13-25.
- [59] Wang Z, Qian H, Li J et al. Uso de memoria transaccional restringida para crear una base de datos en memoria escalable. En Proc. la 9ª Conferencia Europea sobre Sistemas Informáticos, abril 2014, Artículo N° 26.
- [60] Russell RM. El sistema informático CRAY-1. Communications of the ACM, 1978, 21(1): 63-72.

- [61] Barik R, Zhao J, Sarkar V. Selección eficiente de instrucciones vectoriales mediante programación dinámica. En Proc. Simposio internacional IEEE/ACM sobre microarquitectura, diciembre de 2010, pp.201-212.
- [62] Klimovitski A. Uso de SSE y SSE2: Conceptos erróneos y realidad. Intel Developer Update Magazine, marzo de 2001. <http://saluc.engr.uconn.edu/refs/process/in tel/sse sse2.pdf>, febrero de 2017.
- [63] Intel I. Referencia de programación Intel SSE4, D91561-2007. <http://software.intel.com/sites/default/fi 103, les/m/8/6/8/D9156103.pdf>, febrero de 2017.
- [64] Tian C, Zhou H, He Y et al. Un programador Mapreduce dinámico para cargas de trabajo heterogéneas. En Proc. Conferencia Internacional sobre Grid y Computación Cooperativa, agosto de 2009, pp.218-224.
- [65] Sun N, Liu W, Liu H et al. Sistema operativo distribuido Dawning-1000 PROOS. Journal of Computer Science and Technology, 1997, 12(2): 160-166
- [66] Zhang L, Litton J, Cangialosi F et al. Picocenter: admite aplicaciones de larga duración, en su mayoría inactivas, en entornos de nube. En Proc. la 11.^a Conferencia Europea sobre Sistemas Informáticos, abril de 2016, pp.37:1-37:16.
- [67] Canali C, Lancellotti R. Mejora de la escalabilidad del monitoreo en la nube a través de la agrupación en clústeres de máquinas virtuales basada en PCA. Revista de informática y tecnología, 2014, 29(1): 38-52.
- [68] Le K, Bianchini R, Zhang J et al. Reducción del costo de la electricidad a través de la colocación de máquinas virtuales en nubes informáticas de alto rendimiento. En Proc. Conferencia internacional de informática, redes, almacenamiento y análisis de alto rendimiento, noviembre de 2011.
- [69] Chun BG, Ihm S, Maniatis P et al. CloneCloud: Ejecución elástica entre dispositivo móvil y nube. En Proc. la 6^a Conferencia Europea sobre Sistemas Informáticos, abril de 2011, pp.301-314.
- [70] Jin H, Deng L, Wu S et al. Migración de máquinas virtuales en vivo con compresión de memoria adaptativa. En Proc. Conferencia internacional IEEE sobre informática en clúster y talleres, agosto de 2009.
- [71] Ye K, Jiang X, Huang D et al. Migración en vivo de múltiples máquinas virtuales con reserva de recursos en entornos de computación en la nube. En Proc. Conferencia internacional IEEE sobre computación en la nube, julio de 2011, pp.267-274.
- [72] Malewicz G, Austern MH, Bik AJ et al. Pregel: un sistema para el procesamiento de gráficos a gran escala. En Proc. Conferencia internacional ACM SIG MOD sobre gestión de datos, junio de 2010, pp.135-146.
- [73] Kyrola A, Blelloch G, Guestrin C. GraphChi: cálculo de gráficos a gran escala en solo una PC. En Proc. Simposio USENIX sobre diseño e implementación de sistemas operativos, octubre de 2012, págs. 31-46.
- [74] Girod L, Mei Y, Newton R et al. XStream: un sistema de gestión de flujo de datos orientado a señales. En Proc. la 24^a Conferencia Internacional IEEE sobre Ingeniería de Datos, abril de 2008, pp.1180-1189.
- [75] Low Y, Bickson D, González J et al. GraphLab distribuido: un marco para el aprendizaje automático y la minería de datos en la nube. Actas de la Fundación VLDB, 2012, 5(8): 716-727.
- [76] Chen R, Shi J, Chen Y et al. PowerLyra: cálculo de gráficos diferenciados y creación de particiones en gráficos sesgados. En Proc. Conferencia Europea sobre Sistemas Informáticos, abril de 2015.
- [77] Zhang M, Wu Y, Chen K et al. Explorando la dimensión oculta en el procesamiento de gráficos. En Proc. el 12^o Simposio USENIX sobre Diseño e Implementación de Sistemas Operativos, noviembre de 2016, pp.285-300.
- [78] Zhu X, Chen W, Zheng W et al. Gemini: un sistema de procesamiento de gráficos distribuidos centrado en la computación. En Proc. Simposio USENIX sobre diseño e implementación de sistemas operativos, noviembre de 2016, págs. 301-316.
- [79] González JE, Xin RS, Dave A et al. GraphX: Procesamiento de gráficos en un marco de flujo de datos distribuido. En Proc. Simposio USENIX sobre diseño e implementación de sistemas operativos, octubre de 2014, pp.599-613.
- [80] Abadi M, Barham P, Chen J et al. TensorFlow: un sistema para el aprendizaje automático a gran escala. En Proc. el 12^o Simposio USENIX sobre Diseño e Implementación de Sistemas Operativos, noviembre de 2016, pp.265-283.
- [81] Nesbit KJ, Moreto M, Cazorla FJ et al. Gestión de recursos multinúcleo. IEEE Micro, 2008, 28(3): 6-16.
- [82] Bolte M, Sievers M, Birkenheuer G et al. Gestión de virtualización no intrusiva mediante libvirt. En Proc. Conferencia de la Asociación Europea de Diseño y Automatización sobre Diseño, Automatización y Pruebas en Europa, marzo de 2010, pp.574-579.
- [83] Tanenbaum AS, Kaashoek MF, van Renesse R et al. El sistema operativo distribuido Amoeba: un informe de estado. Computer Communications, 1991, 14(6): 324-335 [84] Vavilapalli VK, Murthy AC, Douglas C et al. Apache Hadoop YARN: Otro negociador de recursos. En Proc. Simposio de ACM sobre computación en la nube, octubre de 2013, pp.5:1-5:16.
- [85] Burns B, Grant B, Oppenheimer D et al. Borg, Omega y Kubernetes. ACM Queue, 2016, 14(1): 70-93 [86] Zhang Z, Li C, Tao Y et al. Fuxi: un sistema de programación de trabajos y administración de recursos tolerante a fallas a escala de Internet. Actas de la Fundación VLDB, 2014, 7(13): 1393- 1404
- [87] Harter T, Salmon B, Liu R et al. Slacker: Distribución rápida con contenedores acoplables perezosos. En Proc. Conferencia USENIX sobre tecnologías de archivo y almacenamiento, febrero de 2016.
- [88] Singh B, Srinivasan V. Contenedores: Desafíos con el controlador de recursos de memoria y su rendimiento. En Proc. Simposio Linux de Ottawa, junio de 2007.
- [89] Nikolaev R, Back G. VirtuOS: un sistema operativo con virtualización del kernel. En Proc. Simposio de ACM sobre principios de sistemas operativos, noviembre de 2013, págs. 116-132.
- [90] Soltesz S, P'otzl H, Ficzyński ME et al. Virtualización de sistemas operativos basada en contenedores: una alternativa escalable y de alto rendimiento a los hipervisores. Revisión de sistemas operativos ACM SIGOPS, 2007, 41(3): 275-287.
- [91] Steinberg U, Kauer B. NOVA: Una arquitectura de virtualización segura basada en un microhipervisor. En Proc. Conferencia Europea sobre Sistemas Informáticos, abril de 2010, pp.209-222.
- [92] Boyd-Wickizer S, Clements AT, Mao Y et al. Un análisis de la escalabilidad de Linux a muchos núcleos. En Proc. Simposio USENIX sobre diseño e implementación de sistemas operativos, octubre de 2010, pp.86-93.
- [93] Colmenares JA, Bird S, Eads G et al. Sistema operativo de teselación: construcción de un sistema operativo de cliente de alto rendimiento, receptivo y en tiempo real para arquitecturas de muchos núcleos. En Proc. Simposio IEEE Hot Chips, agosto de 2011.
- [94] Baumann A, Peter S, Sch'upbach A et al. Su computadora ya es un sistema distribuido. ¿Por qué no es su sistema operativo? En Proc. la 12^a Conferencia sobre temas candentes en sistemas operativos, mayo de 2009.

- [95] Wentzlaff D, Agarwal A. Factored operating systems (FOS): The case for a scalable operating system for multicores. *Revisión de sistemas operativos ACM SIGOPS*, 2009, 43(2): 76-85.
- [96] Grandl R, Chowdhury M, Akella A et al. Programación altruista en clústeres de múltiples recursos. En *Proc. Simposio USENIX sobre diseño e implementación de sistemas operativos*, noviembre de 2016, pp.65-80.
- [97] Grandl R, Kandula S, Rao S et al. GRAPHENE: Empaquetado y programación consciente de dependencias para clústeres de datos paralelos. En *Proc. Simposio USENIX sobre diseño e implementación de sistemas operativos*, noviembre de 2016, pp.81-98.
- [98] Gog I, Schwarzkopf M, Gleave A et al. Firmamento: programación de clústeres rápida y centralizada a escala. En *Proc. Simposio USENIX sobre diseño e implementación de sistemas operativos*, noviembre de 2016, págs. 99-115.
- [99] Jyothi SA, Curino C, Menache I et al. Morpheus: Hacia SLO automatizados para clústeres empresariales. En *Proc. Simposio USENIX sobre diseño e implementación de sistemas operativos*, noviembre de 2016, pp.117-134.
- [100] Zhou FF, Ma RH, Li J et al. Optimizaciones para virtualización de redes de alto rendimiento. *Revista de informática y tecnología*, 2016, 31(1): 107-116.
- [101] Tang H, Mu S, Huang J et al. Zip: un algoritmo basado en el árbol de perdedores para buscar contactos comunes en gráficos grandes. *Revista de informática y tecnología*, 2015, 30(4): 799-809.

- [102] Ma C, Yan D, Wang Y et al. Modelo gráfico avanzado para el seguimiento de variables contaminadas. *Science China Information Sciences*, 2013, 56(11): 1-12.



Zuo-Ning Chen recibió su maestría en tecnología de aplicaciones informáticas de la Universidad de Zhejiang, Hangzhou, en 1999. Es profesora adjunta de informática y tecnología en la Universidad de Tsinghua, Beijing, y académica de la Academia China de Ingeniería. Sus intereses de investigación actuales incluyen computación de big data,

computación en la nube y computación de alto rendimiento. Ha realizado importantes contribuciones en el campo del software informático y las computadoras de alta gama y recibió los Premios Especial y Primero del Premio Nacional al Progreso de la Ciencia y la Tecnología de China.



Kang Chen recibió su Ph.D. Licenciado en informática y tecnología de la Universidad de Tsinghua, Beijing, en 2004. Actualmente, es profesor asociado de informática y tecnología en la Universidad de Tsinghua, Beijing. Sus intereses de investigación incluyen computación paralela, procesamiento distribuido y

computación en la nube.



Jin-Lei Jiang recibió su Ph.D. Licenciado en informática y tecnología de la Universidad de Tsinghua, Beijing, en 2004, con honores de excelente disertación. Actualmente es profesor asociado de informática y tecnología en la Universidad de Tsinghua, Beijing. Sus intereses de investigación incluyen computación y sistemas distribuidos,

computación en la nube, big data y virtualización.



Lu-Fei Zhang recibió su licenciatura de la Universidad de Tsinghua, Beijing, en 2008. Actualmente es ingeniero en el Instituto de Tecnología Informática de Jiangnan, Wuxi. Sus intereses de investigación incluyen computación en la nube, big data, etc.



Song Wu es profesor de informática en la Universidad de Ciencia y Tecnología de Huazhong (HUST), Wuhan. Recibió su Ph.D. en ciencias de la computación de HUST en 2003. Ahora se desempeña como director

del Instituto de Cómputo Paralelo y Distribuido, y el vicepresidente del Laboratorio de Sistemas y Tecnología de Cómputo del Servicio (SCTS) del HUST. Ha publicado más de cien artículos y obtenido más de treinta patentes en el área de computación paralela y distribuida. Sus intereses de investigación actuales incluyen computación en la nube y virtualización.



Zheng-Wei Qi es profesor en la Escuela de Software de la Universidad Jiao Tong de Shanghai (SJTU), Shanghai, y miembro del Laboratorio Clave de Computación y Sistemas Escalables de Shanghai. Recibió su Ph.D. licenciatura en ciencias de la computación de SJTU con una tesis sobre el método formal en dis

sistemas tributados. Sus intereses de investigación se centran en sistemas informáticos confiables: máquinas virtuales, análisis de programas, computación en la nube, computación móvil y virtualizaciones de GPU. Recibió el Segundo Premio de el Premio Nacional al Progreso de la Ciencia y la Tecnología de China en 2014.



Chun-Ming Hu recibió su Ph.D. Licenciado en Ciencias de la Computación por la Universidad de Bei Hang, Beijing, en 2006. Es profesor asociado en la Facultad de Ciencias e Ingeniería de la Computación de la Universidad de Beihang, Beijing. Sus intereses de investigación actuales incluyen

sistemas distribuidos, virtualización de sistemas, gestión y programación de recursos de centros de datos y sistemas de procesamiento de datos a gran escala.



Yong-Wei Wu recibió su Ph.D. Licenciado en Matemáticas Aplicadas por la Academia de Ciencias de China, Beijing, en 2002.

Actualmente es profesor de informática y tecnología en la Universidad de Tsinghua, Beijing. Sus intereses de investigación incluyen procesamiento paralelo y distribuido,

sistemas móviles y distribuidos, computación en la nube y almacenamiento. Ha publicado más de 80 publicaciones de investigación y recibió dos premios Best Paper Awards.

Actualmente forma parte de los consejos editoriales de IEEE Transactions on Cloud Computing, Journal of Grid Computing, IEEE Cloud Computing y International Journal of Networked and Distributed Computing.

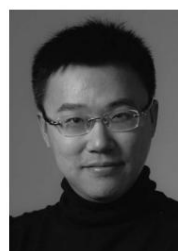


Yun-Zhong Sun recibió su Ph.D. Licenciado en Ingeniería Informática del Instituto de Tecnología Informática (ICT), Academia de Ciencias de China, Beijing. Es profesor en el Laboratorio Estatal Clave de Arquitectura de Computadores del ICT. Sus intereses de investigación se centran

en software de sistemas distribuidos y modelos de computación/programación. Es

autor y coautor de más de 50 publicaciones, y ha participado en varias conferencias académicas y

diarios Es miembro de CCF e IEEE, y de IEEE Computer Society.



Hong Tang recibió su licenciatura en informática de la Universidad de Zhejiang, Hangzhou, en 1997, y su Ph.D. Licenciado en informática de la Universidad de California (UC), Santa Bárbara, en 2003. Se unió a Alibaba Cloud Computing en 2010 y actualmente es el arquitecto jefe de Apsara Cloud Platform. Antes de Alibaba, fue director

de Infraestructura de sistemas de búsqueda en Ask.com. En 2008, se unió al equipo de computación en la nube de Yahoo (EE. UU.) como ingeniero principal sénior, impulsando la investigación y el desarrollo de Hadoop. Los intereses de investigación del Dr. Tang incluyen computación de alto rendimiento y sistemas paralelos, computación distribuida y sistemas de almacenamiento, servicios de Internet a gran escala y computación en la nube. Fue elegido como Talento Innovador en el Programa Nacional de Reclutamiento de Expertos Globales en 2013.



Ao-Bing Sun recibió su Ph.D. Licenciado en Ciencias de la Computación de la Escuela de Ciencias y Tecnología de la Computación de la Universidad de Ciencia y Tecnología de Huazhong, Wuhan, en 2008, y trabajó como investigador postdoctoral en el Instituto de Tecnología de la Computación (ICT), Academia de Ciencias de China, Beijing,

durante 2010–2012. Ahora es vicepresidente de G-Cloud Technology Inc., Dongguan.

Sus intereses de investigación incluyen computación en la nube, computación en cuadrícula, procesamiento de imágenes, etc.



Zi-Lu Kang es el director del Instituto de Tecnología de Internet de las Cosas, Academia de Ciencias de la Información de China Electronics Technology Group Corporation, Beijing. Ha dirigido o participado en múltiples proyectos de ingeniería y desarrollo de productos relacionados con la

ciudad inteligente y el Internet de las Cosas. También ha participado en el desarrollo de estándares de Internet de las Cosas de la Unión Internacional de Telecomunicaciones (UIT).