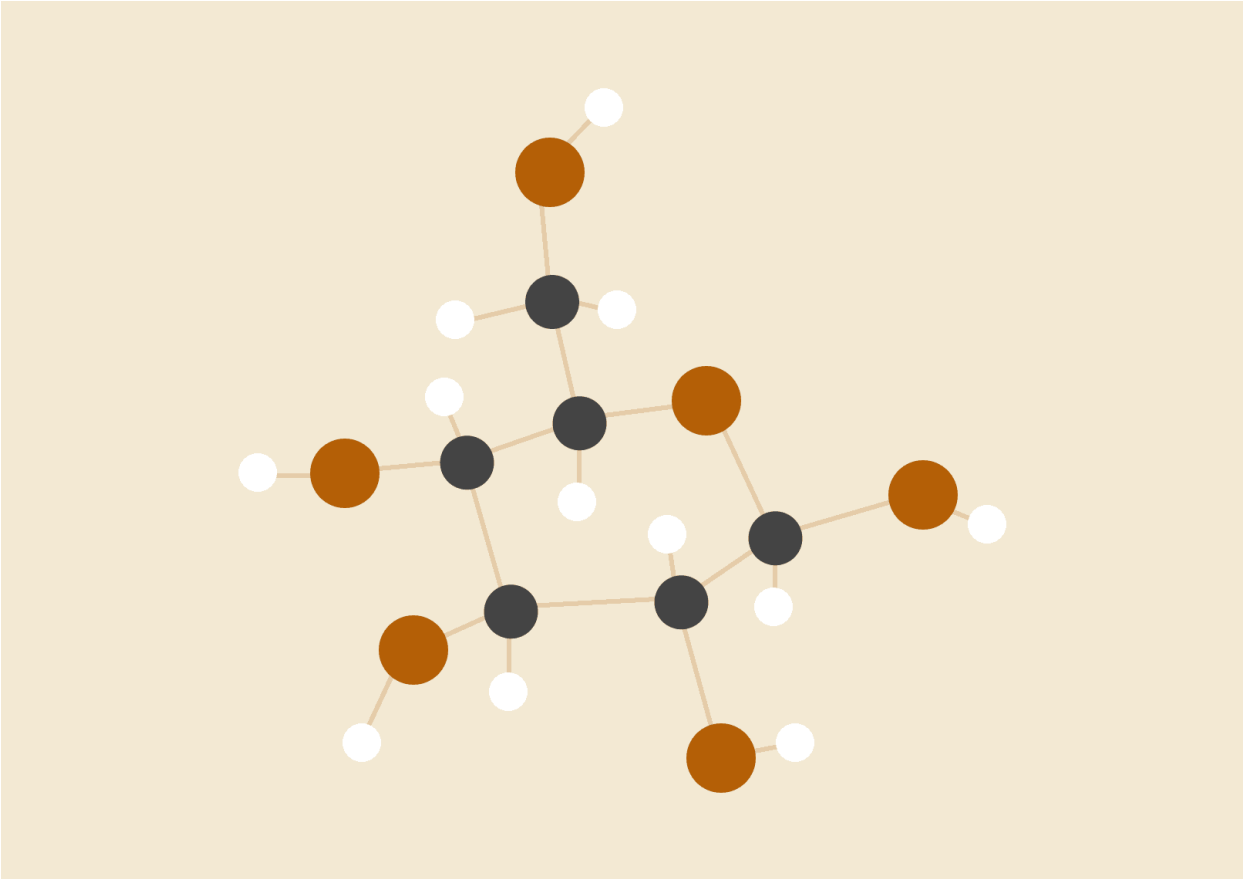
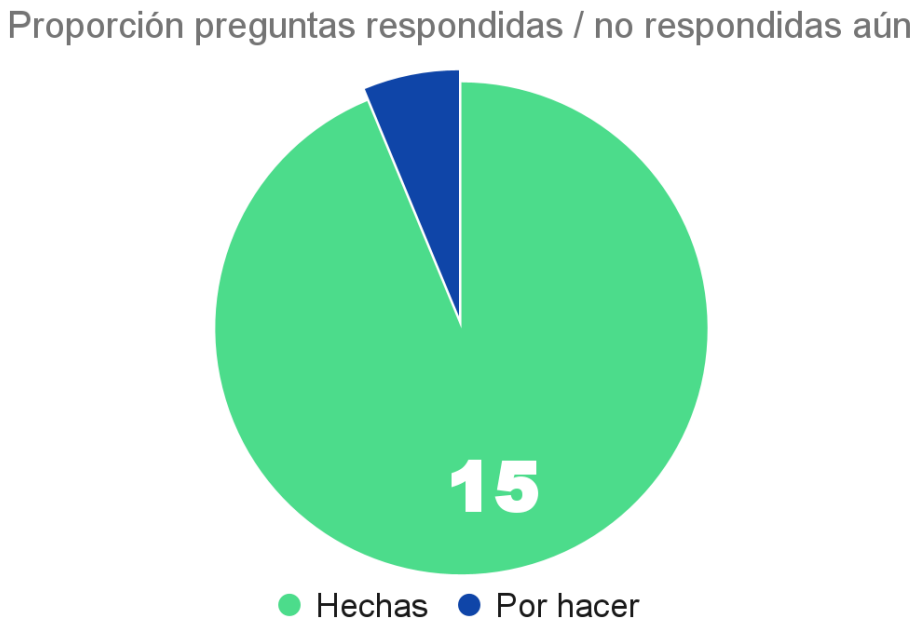
Preguntas de Clau Compuitin



Alumnos



¿Qué es el cloud computing?

> Términos clave: Escalabilidad elástica, catálogo de servicios estandarizados, pago por uso (las dos ideas más importantes según este señor, min 19:25 de la clase del 2 de marzo).

Respuesta: El cloud computing es un **paradigma computacional** y **modelo de prestación de servicios** que fundamentan el uso de tecnologías para permitir el acceso remoto a softwares, almacenamiento de archivos, y procesamiento de datos por medio de una vía de comunicación, **Internet**. Las empresas que se dedican a ofrecer servicios de cloud computing contienen un catálogo de servicios estandarizados a los clientes y son capaces de responder a las necesidades de los negocios clientes de forma flexible y adaptativa, es decir, ofrece una escalabilidad elástica. En caso de demandas no previsibles o de picos de trabajo, el coste del consumo de estos servicios es únicamente por el consumo efectuado, y no por un contrato fijo de servicio.

Cuestiones adicionales:

- Se puede integrar con mucha velocidad y facilidad con las aplicaciones empresariales que no estaban pensadas para este paradigma.
- El coste necesario para desarrollar aplicaciones basadas en el uso de soluciones Cloud Computing es mucho menor al desarrollo convencional y mucho más simple, ya que no requiere tener el personal, conocimiento experto, ni hardware especializado para nuestros servicios a prestar.
- Las infraestructuras de Cloud Computing proporcionan mayor capacidad de adaptación ante la recuperación de desastres como terremotos o inundaciones, o incluso guerras, a la vez de que ofrece la capacidad de replicación multiregión, para minimizar las latencias de acceso a esos servicios.

Fuente: Manual de Retos y Oportunidades del Ministerio de Industria, Energía y Turismo:https://www.ontsi.red.es/sites/ontsi/files/1-_estudio_cloud_computing_retos_y_oportunidades_vdef.pdf

Autor: Pepe.

¿Qué es la digitalización o Transformación digital?

Términos clave: transformación digital, economía material o analógica, economía digital,

Respuesta. La Transformación digital es el **cambio** de una **economía** esencialmente material o **analógica** a una **economía digital** que depende del intercambio de datos. Este cambio implica una revolución de las organizaciones tal y como las conocemos ya que deben de adaptar sus procesos y modelos de negocios para conseguir la transformación de los procesos operativos de forma holística. El cloud computing es un pilar fundamental para conseguir llevar a cabo este cambio debido a que nos permite solventar los problemas de la **infraestructura tecnológica limitada, el incumplimiento de los niveles de servicio** (SLA), o la **falta de conocimiento técnico-experto** en las propias organizaciones.

El Cloud Computing permite implementaciones ágiles de plataformas de alto rendimiento así como una gran capacidad de almacenamiento y de acceso distribuido, rápido, y seguro de la información, donde la redundancia, disponibilidad y escalabilidad están garantizadas. Además el Cloud Computing permite realizar esta transformación económica al estar basado en una economía de escala, es decir, los costes se facturan según el consumo por uso, y no por inversiones contractuales fijas, lo que facilita la transformación digital de las pequeñas y medianas empresas además de las grandes.

Fuente:
https://revistaempresarial.com/tecnologia/tendencias/la-importancia-del-cloud-computing-en-la-transformacion-digital/

Autor: Pepe.

Esta fuente también está muy bien y viene con ejemplos:
https://www.netguru.com/blog/digitization-and-digitalization

¿Qué queremos decir con la expresión: Democratización de la inteligencia artificial?

El uso de la Inteligencia Artificial se encuentra en auge en la actualidad y su uso es ya cotidiano por millones de personas a través de un dispositivo inteligente como un móvil o un ordenador. Esta IA interactúa en procesos cada vez más cotidianos de nuestro día a día y mucha gente aún no comprende el impacto que produce esta, ni como se llega a desarrollar.

Tratando este punto entramos en lo que se conoce como **Democratización de la inteligencia artificial**, si bien no es un término como tal, si es una idea o planteamiento sobre el desarrollo y aceptación de la IA en el futuro de la sociedad.

Esto requiere un aprendizaje por parte de los diferentes usuarios y en un futuro requerirá que la gente aprenda a convivir y trabajar con ella de forma natural debido al obvio avance tecnológico. Pero por otro lado surge un contrapunto importante: ¿Es realmente la IA comprensible?

Entre otras afirmaciones, se destaca esta sobre la democratización: “La IA **no** puede ser una **caja negra** y si una solución **no puede explicar** cómo y con qué criterios elabora sus recomendaciones, **no debería utilizarse**. Solo la IA **razonada**, **transparente** y **justa** sobrevivirá”.

Bajo este enfoque, la democratización consiste en el procedimiento llevado a cabo por la sociedad de aceptación y uso de la Inteligencia Artificial para el ámbito que sea, la cual establece una serie de criterios como conocer el funcionamiento, procedimiento y razonamiento de la misma para poder confiar en ella, ya que en caso contrario la desconfianza conlleva al no uso de la misma y por lo tanto a su abandono.

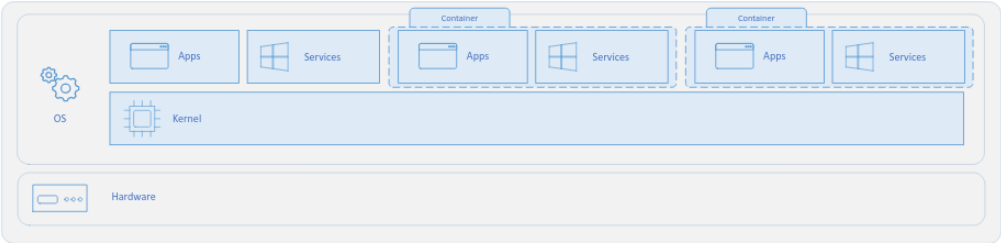
Podemos afirmar que la IA se encontrará **democratizada** ya que el uso de esta dependerá directamente de su **comprensión** por parte de los usuarios, y por ende, de la **sociedad**.

Fuente: https://elpais.com/retina/2019/01/24/tendencias/1548329836_548262.html

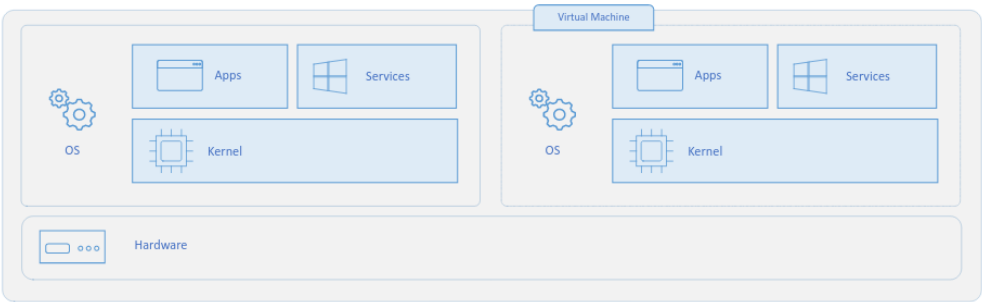
Autor: Carlos.

¿Puedes hacer más tareas en un contenedor que en una máquina virtual?

Contenedor



Máquina virtual



No, en ambas igual, realmente el límite lo pone el hardware subyacente en ambos casos, si bien suele ser más común realizar más tareas con un contenedor debido a que no soporta la carga del sistema operativo ni la configuración de aspectos como la seguridad, por lo que suele ser la opción más usada.

No, la capacidad de procesamiento es la misma. Sólo se pueden tener en ejecución tantas hebras como hebras soporte el procesador (cores), es decir, 8 cores en una máquina virtual va a procesar lo mismo que 8 cores en un contenedor. Si un ordenador tiene 8 cores, NO vamos a tener 4 máquinas virtuales con 8 cores cada una, sino que cada una tendrá 2 cores.

En términos de posibilidad, ambos igual, aunque exista una tendencia, y no tenga demasiado sentido dedicar una máquina virtual completa a la realización de cada tarea, sino la virtualización mediante contenedores en una misma máquina. La conclusión final es **no**, depende principalmente del **hardware**.

Autor: Carlos.

¿Cómo lleva a cabo el reparto de recursos un hipervisor en una máquina virtual?

El hipervisor impone las restricciones del hardware en las distintas máquinas virtuales y garantiza que únicamente vea los componentes especificados. También se llama monitor de máquina virtual. (Clase 16 marzo 2021 - 1:08:00) Esto se produce porque hay una capa más entre el hardware y el SO que permite que se pueda gestionar mejor los recursos a consumir de cada Máquina Virtual. Hay 3 conceptos clave:

- **Particionamiento:** Permite definir el tiempo que se le comparte un recurso a una MV, es decir que se puede compartir 0,5 cores en el tiempo que se ejecuta un programa (1 hora por ejemplo) y luego se puede configurar para que tenga otro valor.
- **Individual:** Cada Máquina Virtua tiene que ser independiente del host físico, para que no fallen, JAMÁS se puede permitir que el fallo de una Máquina Virtua afecte a otra
- **Encapsulamiento:** Una Máquina Virtua se gestiona como un único fichero.

(Clase 4, 16 marzo 2021 - 1:10:00)

Autor: Guillermo.

¿Dónde se guarda la información sobre las máquinas virtuales?

La información sobre las máquinas virtuales se guarda en un directorio de la máquina host o anfitriona (la que tiene acceso al hardware sin la capa de virtualización y da soporte a la capa de virtualización). Esta ubicación es normalmente establecida por defecto por el sistema hipervisor que usemos, aunque es normalmente intercambiable.

Para VirtualBox, las ubicaciones más típicas suelen ser:

> Ubuntu : /home/usuario/VirtualBox VMS
>
> OS X : /Users/usuario/VirtualBox VMS
>
> Windows : C:\Users\usuario\VirtualBox VMS

A su vez la configuración de estas máquinas virtuales suele estar englobada en un archivo etiquetado (estilo xml) en el directorio raíz de cada máquina virtual.

La información sobre las máquinas virtuales se guarda en un único archivo que ocupa mucho espacio (llegando a pesar GB), pero es un único archivo.

Autor: Pepe.

Ventajas del cloud computing.

Aspectos clave: **Servicio bajo demanda, acceso ubicuo de la red, Servicio medido, Recursos centralizados, Elasticidad rápida.**

Escalabilidad elástica: Adaptación de los recursos usados en función de las demandas cambiantes. Con aprovisionamiento frente a solicitudes de demandas dinámicas (recursos ilimitados), limitando a los usuarios los recursos que quiere (Clase 2, 2 de marzo 45:00)

Multitenant (inquilino): Los recursos son virtualizados, cada recurso real es utilizado concurrentemente por varios usuarios. Seguridad, privacidad y protección de datos es una prioridad. (Clase 2, 2 de marzo 48:00)

Aspectos clave:

- **Autoservicio:** los recursos cloud se demandan por el usuario en cualquier momento. El usuario accede a los servicios cloud a través de un panel de control. Puede escalar la infraestructura como quiera.
- **Escalabilidad elástica:** adaptación de los recursos usados frente a demandas cambiantes. Aprovisionamiento frente a solicitudes de demandas dinámicas: recursos ilimitados.
- **Multitenant:** un servicio cloud debe servir a múltiples usuarios a la vez usando el **modelo multitenant**, con recursos físicos y virtuales que se asignan dinámicamente y se reasignan de acuerdo a la demanda del usuario. Los usuarios no tienen ni idea de dónde está el recurso que buscan, pero pueden especificar una zona en concreto (un país o estado). Cada cliente es un **tenant**. Los recursos son virtualizados y cada recurso real se utiliza concurrentemente por varios tenant.
- **Red de acceso:** se debe acceder a los recursos a través de toda la red a cualquier tipo de cliente (portátil, ordenador, móvil, tablet, etc)
- **Servicios medidos:** el uso de recursos se miden independientemente a cada usuario, con lo cual es un **pay-as-you-use**.

Ventajas:

- **Costo:** se reduce la infraestructura, ordenadores de usuario, licencias software, energía y personal. Por ejemplo: en condiciones normales, cada empleado tiene su propio ordenador y cada ordenador almacena el software que tiene que utilizar. Si un ordenador tiene 1TB y sólo tiene ocupado 300GB, se están desperdiciando 700GB.
- **Gestión centralizada:** cuando la gestión se centraliza, se facilitan las cosas:
    - **Menos incidentes**

    - **Actualizaciones de software instantáneas**
- **Mejora de prestaciones:** adaptación elástica a la demanda dinámica de recursos:
    - Equipos más potentes, más cores, más almacenamiento, más ancho de banda…
    - Adaptación elástica a la demanda dinámica de recursos
    - Mayor seguridad en los datos
    - Disponibilidad 24/7 casi siempre
    - Accesible desde cualquier dispositivo
- **Universabilidad:** acceso desde cualquier dispositivo. Colaboración en grupo más sencilla y acceso universal a los documentos.

Desventajas

- Requiere conexión a internet todo el tiempo
- No funciona bien con anchos de banda pequeños
- Puede ser lento
- Ofrece menos características que aplicaciones de escritorio
- Cuestiones de privacidad y seguridad

Autor: Guille.

Diferencias entre SaaS y el método tradicional.

Tradicionalmente se hacían demos para vender el software, mediante soportes físicos. Para poner en marcha había que hacer instalaciones en cada ordenador y se vendían licencias de uso. (Clase 6, 6 de abril 41:00)

Ya no hay un modelo físico de ello.

Características:

- Software alojado y ejecutándose en equipos remotos. Por tanto se ejecuta la versión más actualizada, sin preocupación de la versión
- Pago por uso.
- **Escalable**.
- Los recursos hardware se adecúan a demanda
- Infraestructura local mínima
- Siempre actualizado

(Clase 6, 6 de abril 51:00)

Las aplicaciones son multiplataforma. Algunas aplicaciones que no se pueden hacer, software de control de presa, de una central nuclear, de dispositivos embebidos (controles de un avión, por ejemplo):

- Cuando el tiempo de respuesta sea crítico.
- Cuando la legalidad vigente no permita que se alojen datos en el extranjero.
- Cuando las soluciones locales cumplan los requisitos.

(Clase 6, 6 de abril 1:00:00)

Antes para que el cliente pudiera probar la aplicación realizada, se metía en un USB y se le daba al cliente. En SaaS se despliega la aplicación en la nube y el cliente puede acceder a ella desde cualquier parte.

Autor: Guille.

Diferencias entre el proceso de ciclo de vida de una app en el método tradicional y en el cloud.

Una app tradicional a menudo incluye tres capas, el nivel de presentación, el nivel de lógica de aplicación y el nivel de acceso a los datos (DAO). Cada uno de estos niveles requiere de un servidor dedicado, cuenta con una infraestructura estática y es en su mayoría un sistema rígido, lo que significa que la escalabilidad normalmente es limitada y desafiante a la vez que costosa. Estas aplicaciones siguen un ciclo de vida lento y su entrega al cliente final suele ser pactada en el tiempo y de forma física o presencial.

Una app cloud native, se caracteriza por la automatización. Mientras que las aplicaciones tradicionales usan bases de datos rígidas, las arquitecturas de aplicaciones en la nube ofrecen opciones más flexibles y con capacidad de dividir los datos en función de los requisitos de uso (bases noSQL). Estas aplicaciones cloud native son un conjunto de servicios pequeños, comúnmente denominados microservicios, independientes y de bajo acoplamiento, suelen ofrecer una API, es decir una interfaz de programación para acceso a los datos y funciones, suelen estar empaquetadas en contenedores para crear un entorno aislado, reproducible y seguro, y suelen seguir la filosofía o cultura DevOps.

- **Análisis de requisitos:** no hay diferencia. El análisis de requisitos es de lo que quieres que haga el sistema, las propiedades que debe tener en el que queremos desplegar la app.
- **Análisis de requisitos software:** requisitos funcionales y no funcionales. No hay diferencia porque estamos pensando en qué se quiere resolver, no en cómo se va a implementar. Hay una excepción: si queremos garantizar respuestas en tiempo real sí habría diferencia.
- **Diseño preliminar:** a partir de aquí sí hay diferencia. Se debe hacer el diseño arquitectónico, descomposición de la solución. Se tienen unas entradas y unas salidas. Estamos haciendo el cómo lo vamos a diseñar. Vamos a pasar de un diseño monolítico a descomponer en varios servicios. En vez de tener bibliotecas, tenemos servicios.
- **Diseño detallado:** el diseño preliminar arrastra los cambios a esta etapa.
- **Codificación y pruebas:** también cambia.
- **Explotación y mantenimiento:** capacidad de aprovisionamiento, distintos recursos, etc. Desde el punto de vista del programador no hay que pensar en la autenticación, bd, etc. Sólo hay que utilizar una API.

Autor: pepe

Diferencia entre servidor web y servidor de aplicaciones.

Un servidor web es un programa informático que acepta solicitudes de información y envía los documentos requeridos mientras que un servidor de aplicaciones es un marco software que se dedica a la ejecución de programas, scripts y rutinas. Un servidor de aplicaciones puede contener servidores web en su interior, por lo que un servidor de aplicaciones es superior o más potente que un servidor web.

Un ejemplo de servidor web puede ser Nginx o Apache , mientras que un ejemplo de servidor de aplicaciones puede ser Apache Tomcat, Glassfish, JBoss (actualmente Wildfly), o Internet Server para aplicaciones .NET.

- Los servidores web son adecuados para contenido estático, mientras que los servidores de aplicaciones son apropiados para el contenido dinámico.
- Los servidores de aplicaciones suelen tener mayor capacidad de cargar que los servidores web.
- Los servidores web no admiten el soporte de ejecución multihilo, mientras que los servidores de aplicaciones sí.
- Los servidores web admiten servir contenido con lenguajes como Perl, PHP, ASP, JSP, mientras que los servidores de aplicaciones permiten usar lenguajes como Java (en su versión empresarial)

En resumen, la tarea principal de un servidor web es la de responder y aceptar todas las solicitudes de los usuarios para acceder a contenido estático que se pueda visualizar en el navegador web, mientras que los servidores de aplicaciones alojan y exponen la lógica empresarial en las aplicaciones y procesos que dan funcionalidad y operativa.

Un **servidor de aplicaciones** es lo que **almacena código**. Son aplicaciones ejecutables y gestionan la interacción entre ellas. Por ejemplo, Tomcat es un servidor de aplicaciones de java. Tenemos distintas aplicaciones en java ya compiladas y se comunican entre ellas. Por ejemplo, una API en java lo que hace por dentro es ejecutar código en java, procesar la información y devolverla en xml, json, u otro.

Un **servidor web es** una **máquina física** sobre las que se despliega un servidor web. Un programa que entiende el protocolo HTTP y atiende peticiones HTTP. Atiende peticiones de navegadores y de aplicaciones. Rastreator es un servidor web y el usuario hace una petición al servidor.

Autor: Pepe.

Diferencias entre el Hosting Tradicional y el Cloud hosting.

La arquitectura tradicional tiene 3 capas: presentación, intermedia y de datos. El desarrollo es estático y se asumen servidores y red totalmente fiables. Normalmente no tiene capacidad de escalado, cuando el sistema llega a su límite de capacidades se plantea trasladar la aplicación a otro sistema con más hardware. No hay posibilidad de crecer infinitamente.

Sin embargo, el Cloud Computing proporciona la escalabilidad, auto-servicio y variedad en los soportes de computación y almacenamiento que faltan en la arquitectura tradicional. En lugar de bibliotecas, en Cloud Computing se tienen servicios que son código vivo y que están constantemente funcionando, y se accede a ellos a través de una API. En la arquitectura cloud se necesitan cookies para establecer una relación entre un navegador y su estado para recabar la información del usuario y su comportamiento.

¿Es posible llevar una arquitectura tradicional al cloud? Sí. La forma más simple es tener las 3 capas pero añadiendo un balanceador de carga. Obviamente esto no funcionará bien ya que la arquitectura no está diseñada para soportar balanceos.

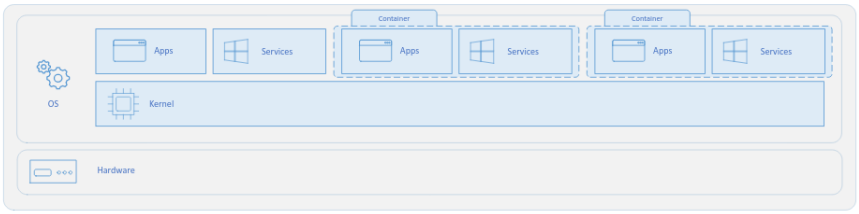
Autor: Juanma

Pepe anotación: Creo que esta pregunta puede enfocar de la forma que ha comentado Juanma, pero creo qeu se refiere a: qué diferencia habría entre contratar hosting (como Arsys, Hertzner, 1&1,etc) frente al Cloud Computing actual (AWS, Azure, Alibaba, IBM).

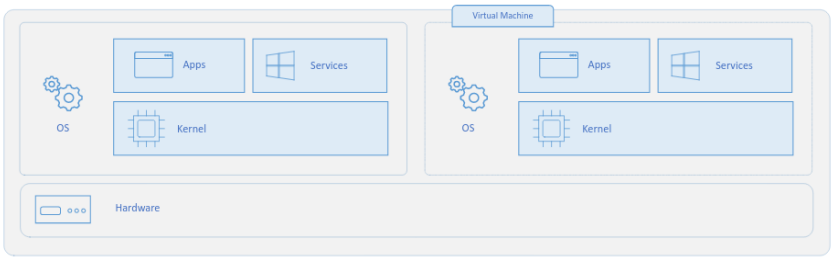
Si es así, el hosting tradicional se clasifica en tres categorías, el gratuito, el compartid y el VPS (que es el más similar a alquilar una instancia de computación AWS Ec2 por así decirlo), y la principal diferencia con el cloud computing es que el pago de los servicios se hace bajo contratos mensuales o anuales y no mediante pago por uso (demanda elástica), y que la escalabilidad dinámica no está garantizada, mientras que en el cloud si, porque si por ejemplo yo pido un VPS de Hertzner con 1 core y 2 de RAM, escalar significa pagar por la antigua instancia y por la nueva en ese mismo mes, mientras que en aws, se paga las horas de la instancia antigua, y las horas de cuando ya la has convertido a la más potente. Además, Hertzner no deja por ejemplo escalar mucho, sólo cambios pequeños, como añadir 1 o 2 cpu, 2 o 3 mas de ram, mientras que AWS o Azure tienen un catálogo de instancias muy grande y muy potentes.

Explica las ventajas y desventajas de una máquina virtual frente a un contenedor.

Contenedor



Máquina virtual



Ventajas

- Todos los recursos del sistema operativo están disponibles para las aplicaciones.
- Herramientas de gestión y seguridad establecidas.
- Controles de seguridad más sencillos y controlables.

Desventajas

- Recursos de gestión de TI más amplios.
- Más lento en la gestión de aplicaciones.
- Actualizaciones de seguridad más amplias y desarrolladas.
- Necesidad de mantenimiento de la seguridad y configuración de la máquina para todas las aplicaciones.
- Mayor peso -> Más carga en memoria y mayor esfuerzo en migraciones.

Fuente:
https://customprofessionalhosting.com/noticias/contenedores-de-servicios-vs-maquinas-virtuales/

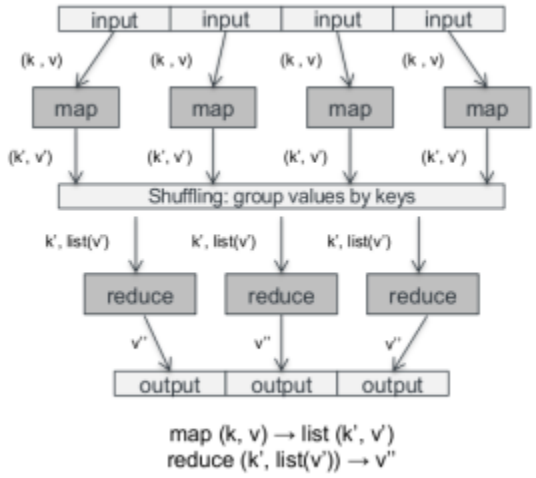
Autor: Carlos.

Explica *brevemente* en qué consiste el paradigma MapReduce.

Es un paradigma de programación que aplica la computación paralela y distribuida a grandes conjuntos de datos sobre clusters de ordenadores. Está caracterizado por dividirse en dos fases o pasos diferenciados: **Map y Reduce**. Estos subprocesos asociados a la tarea se ejecutan de manera distribuida, en diferentes nodos de procesamiento o esclavos. Para controlar y gestionar su ejecución, existe un proceso Master o *Job Tracker*.

Es sencillo y no aplicable a todos los problemas, ya que solo se aplicará en problemas descomponibles en funciones **Map** y **Reduce** y con datos que se puedan representar mediante pares de **clave-valor**. Sigue la filosofía de **divide** y **vencerás**.

1. Cada trozo es procesado por **Map** el cual devuelve un valor -> {clave, valor}
2. Al finalizar los procesos se organiza la salida por las claves mediante **Shuffling** y se unen los valores.
3. Todos los grupos se mandan a sus correspondientes procesos **Reduce** el cual obtiene la lista y genera un valor final.



Autor: Carlos.

¿Qué ventajas tiene el despliegue de una aplicación en una plataforma de *Cloud Computing*?

La respuesta puede ser igual que la de Ventajas del cloud computing

¿Cuál es la relación entre *Cloud Computing* y los sistemas gestores de bases de datos No-SQL?

Las bases de datos NoSQL se crearon en la era del Cloud Computing e hicieron posible implementar aún más fácilmente la escalabilidad horizontal. La escalabilidad horizontal se consigue aumentando el almacenamiento de datos y el trabajo para procesar los datos a través de clusters. Para incrementar la capacidad, se añaden más ordenadores al cluster, y esto es muy sencillo a través del Cloud Computing debido a una de sus propiedades más importantes que es la escalabilidad elástica.

También las NoSQL se crearon para los desafíos que presentan las aplicaciones en el Cloud: datos a escala web, procesamiento masivo de datos, alta frecuencia de lecturas y escrituras, y aplicaciones sociales (no bancarias). Permiten almacenar los datos de maneras mucho más fáciles de entender y muy parecidas a las que se utilizan en la propia aplicación. En SQL hay que recuperar las sentencias a través del lenguaje SQL y adaptarlas a los objetos de la aplicación. En NoSQL se pueden recuperar los datos y te los puede devolver en un JSON, que es una estructura muy sencilla de manejar.

En Cloud Computing el tamaño del esquema crece desproporcionadamente, hay muchos datos temporales y las transacciones locales no son muy durables. Las bases de datos NoSQL permiten modificar el esquema de forma muy sencilla sin tener que modificar absolutamente toda la estructura como sí pasa en SQL (hay que modificar, realizar paso a tablas de nuevo, etc). En las NoSQL la disponibilidad es más importante que la integridad. Las aplicaciones sociales no necesitan que sus datos cumplan las reglas de integridad (no es deseable, pero no es lo que más preocupa), sino que necesitan que sus datos estén disponibles y sean fáciles de recuperar.

Autor: Juanma

¿En qué aspectos Spark mejora a Hadoop?

Cuando hablamos de procesamiento de datos en Big Data existen dos grandes framework para ello, Apache Hadoop y Apache Spark.

Apache Hadoop es un proyecto de código abierto de la fundación Apache (nacido en 2011) que permite procesar grandes volúmenes de datos cuando estos se encuentran distribuidos.

Apache Spark también es un proyecto de código abierto (nacido en 2012) como mejora del paradigma Map Reduce que propuso Hadoop. Tiene abstracciones de programación de alto nivel, y nos permite trabajar con lenguaje SQL. Entre sus APIs encontramos Spark Stream y Spark Structured Streaming (para el procesado de datos de tiempo real), Spark MLib (para el procesamiento de trabajos de machine learning) y Spark GraphX (para el procesamiento de grafos).

Cuando hablamos de diferencias entre Spark y Hadoop realmente queremos hablar de las diferencias entre Spark con Hadoop Map Reduce, ya que son los que realizan las mismas funciones y no podemos comparar peras con manzanas.

1. El rendimiento de Spark es hasta 100 veces más rápido que Hadoop Map Reduce porque trabaja en memoria RAM. Spark trabaja con un planificador de tareas llamado DAG que optimiza el reparto de tareas y el flujo de ejecución.
2. Hadoop Map Reduce se programa principalmente en Java, y la metodología de desarrollo Map Reduce es costosa en tiempo. Spark abstrae este esfuerzo, que aunque por dentro use Map Reduce, lo abstrae y es mucho más rápido y sencillo trabajar con él.
3. Hadoop Map Reduce requiere de un cluster que cuente con mayor cantidad de almacenamiento y que sean rápidos para el procesamiento, mientras que Apache Spark requiere de que los nodos del clúster poseen la mayor cantidad de memoria RAM posible.

A forma de resumen, Apache Spark puede parecer el candidato perfecto para la computación Big Data, pero éste no posee un sistema de ficheros distribuido, por lo que le tendríamos que conectar a cada nodo Spark el volumen o volúmenes donde se encuentren los datos que queremos procesar, mientras que con Hadoop Map Reduce, que se basa en el HDFS visto en las clases de práctica, éste problema desaparece de un plumazo al ser un acceso distribuido. Además cabe destacar, que en la jerarquía de memoria, el coste de los discos de almacenamiento son varias magnitudes de veces más baratos que la memoria principal (RAM), por lo que Spark sería más caro de escalar.

Autor: pepe

¿Qué diferencia hay entre un modelo paralelo y un modelo distribuido? ¿Qué tipo de modelo es Hadoop?

En el modelo distribuido la comunicación entre los nodos se hace a través de una red. Esta red está formada por switches, routers, etc. Como hay comunicación de nodo a nodo, hay más latencia.

En un modelo paralelo la comunicación se hace a través de un bus. Este modelo va mucho más rápido porque la comunicación va a través de la memoria principal.

El problema de los modelos paralelos es que son muy costosos, su programación es muy complicada y no tienen escalabilidad, es decir, una vez que se llega al máximo de su capacidad, ya no se puede superar más.

Hadoop es un modelo distribuido.

Autor: Juanma

Dudas que le han surgido a Ángel P. y que ha resuelto

Definición de contenedor

Definición de Ángel: Un contenedor representa un espacio asignado en memoria que contiene un conjunto de aplicaciones y servicios que se comunican entre sí y/o con el exterior para ofrecer funcionalidades (generalmente servicios).

Definición de las diapositivas: Unidad estándar de software que empaqueta código y todas sus dependencias de forma que la aplicación pueda ejecutarse rápida y fiablemente desde un entorno computacional a otro. Una imagen de contenedor es un paquete de software ejecutable ligero y autónomo que incluye todo lo que necesita para ejecutar la aplicación.

Diferencia entre contenedor y máquina virtual

Las máquinas virtuales realizan sus operaciones sobre un hardware virtualizado a través de una capa intermedia (el hipervisor). Los contenedores se ejecutan en el espacio del usuario, sobre el propio S.O. de la máquina anfitriona en la que se encuentran instalados.

Por esta diferencia, los contenedores son menos flexibles, ya que pueden existir dependencias de S.O. que no pueden solventar al estar obligados a usar software compatible con el S.O. anfitrión.