



Universidad de Alcalá

Escuela Politécnica Superior

Universidad de Alcalá

Ampliación de Programación Avanzada

Tema 3 – Cloud computing

Grado en Ingeniería Informática – Curso 2018/2019

APA – TEMA 3 – UNIDAD 1

CLOUD COMPUTING

INTRODUCCIÓN

Cloud computing está basado en la computación en Internet, donde los recursos, el software y la información son proporcionados a computadores y otros dispositivos bajo demanda.

Permite, por tanto, una computación a gran escala con acceso a cantidad prácticamente ilimitada de recursos.

Previamente se intentó con el ubiquitous computing, grid computing, etc.

PRECEDENTES

- Crecimiento exponencial en las aplicaciones
- Utilizado en generación de contenido escalable
- Crecimiento exponencial en capacidades de cómputo
- Ciclos de obsolescencia muy cortos
- Nuevas arquitecturas
- Dificultad de manejo de situaciones complejas con la infraestructura tradicional

CARACTERÍSTICAS

El cloud computing es una forma para las empresas de ahorrar dinero en infraestructuras y mano de obra. Las soluciones cloud:

- Suelen facturarse en base al consumo
- Pueden ser públicas o privadas
- Tienen una API basada en servicios

VENTAJAS GENERALES

- Te despreocupas del HW
- Se suele pagar por el consumo realizado
- Es escalable
- Independencia de localidad
- Seguridad física en el centro de datos

MODELOS TÍPICOS

SAAS (SOFTWARE AS A SERVICE)

Los consumidores acceden a aplicaciones software a través de la nube. Esencialmente se usa para alquilar software en lugar de comprarlo.

Ejemplos: P.ej. Google drive, Gmail, etc).

Ventajas:

- No tiene requerimientos hardware
- Es escalable
- Se paga lo justo y necesario
- Accesible y personalizable

PAAS (PLATFORM AS A SERVICE)

Plataforma para desarrolladores de cara a crear aplicaciones y servicios que funcionen a través de internet. Se acceden a través del navegador.

Ejemplos: Google App Engine, Microsoft Azure, Amazon Web service...

Ventajas:

- Todas las de SaaS.
- Gestor de BBDD
- Sistema operativo
- Aplicaciones online por defecto
- Soporte técnico
- Colaboración entre varios equipos en lugares separados
- Seguridad

IAAS (INFRAESTRUCTURE AS SERVICE)

Recursos informáticos en la nube. Mientras que en PaaS sólo debes preocuparte por el desarrollo de tu aplicación y el resto lo hace la empresa, con IaaS es un self-service: te dan la plataforma en la nube y el resto lo debe hacer la empresa cliente: Almacenamiento, sistemas operativos, etc.

Ejemplos: AWS, Azure...

Ventajas:

- Similares a las anteriores: escalabilidad, independencia de localidad, seguridad, sin necesidad de invertir en HW...

TIPOS DE NUBES SEGÚN PRIVACIDAD

Se entiende por nube el ecosistema e infraestructura computacional que se contrata.

Existen:

- Nubes privadas: el servicio y uso exclusivo es **por parte de la empresa**. Tiene unos costes de inversión más altos, y muchas medidas de seguridad.
- Nubes públicas: el servicio pertenece a **terceros** en lugar de la empresa. Es más barato o incluso gratuito, y posee una infraestructura multiuso (varios usuarios).
- Nubes híbridas: combinan ambas.

Las privadas tienen la ventaja de poder usar la misma infraestructura para todo lo que haga la empresa, lo cual abarata costes. Como desventaja, requieren de más seguridad y de altos requerimientos computacionales.

PRINCIPALES PROVEEDORES

AZURE

Nube de Microsoft. Características:

- Se ajusta a la demanda
- Ciclos y almacenamiento bajo
- Se usa la API de Azure.
- Web role, worker role, blob storage (fichero de datos inmutables), table y drive-storage.

AMAZON EC2

Proporciona una API para ejecutar instancias de los SSOO que soporta.

Características:

- Es un servicio web. (AWS?)
- Facilita la computación vía Amazon Machine Image para varios modelos.
- S3, Cloud Management Console, MapReduce Cloud, AML.
- Excelente distribución, balance de carga y herramientas de monitorización cloud.

GOOGLE APP ENGINE

Diseño y desarrollo de Java, Go y Python. Algunos otros de la JVM. Características:

- Servicio PaaS.
- Escala irrelevante (¿?)
- Interface basada en la programación.
- Templates, excelente monitorización y gestión desde la consola.

APA – TEMA 3 – UNIDAD 2

GOOGLE APP ENGINE

INTRODUCCIÓN

GAE es una herramienta para el alojamiento de aplicaciones web escalables sobre la infraestructura de Google.

Su misión es permitir al desarrollador web crear fácilmente aplicaciones web escalables si tener que ser un experto en sistemas.

Admite aplicaciones escritas en Java, Python, Ruby y Go.

Ofrece un servicio de almacenamiento de datos con motor de búsqueda y procesador de transacciones.

Escalable y gratuito.

GAE aporta recursos de forma dinámica. Si nuestro portal crece, le asigna más recursos de procesamiento y BBDD.

GAE permite una interacción sencilla e integrada con otros servicios de Google como Gmail, Google pay...

CARACTERÍSTICAS

- Alojamiento y escalado automático de apps usando:
 - Aplicaciones Python y Java
 - Bases de datos BigTable
 - Sistema de archivos global file system.
- Sólo hay que subir el código compilado a Google y monitorizarlo.
- **No permite** el acceso a ficheros, llamadas al SO ni configurar la red para facilitar la escalabilidad.

GLOBAL FILE SYSTEM

- El fallo de un archivo no le afecta.
- Almacena archivos muy grandes
- El trabajo está repartido entre grandes y pequeñas lecturas aleatorias.
- Diseñado para ofrecer concurrencia.
- Gran ancho de banda y poca latencia.

LIMITACIONES DE GAE

- Gratis durante el periodo de prueba y con limitaciones
- Lenguajes limitados
- Acceso de solo lectura al Google File System.

GAE PARA JAVA

App Engine utiliza el estándar Java Servlet para aplicaciones web. Permite crear aplicaciones web a través de tecnologías estándar de Java y las ejecuta en la infraestructura escalable de Google.

JVM se ejecuta en un entorno **aislado y seguro** para aislar la aplicación por seguridad. Una aplicación en GAE solo puede realizar acciones que no interfieran con el rendimiento y escalabilidad de otras aplicaciones.

FUNCIONALIDAD

App Engine proporciona servicios escalables para:

- Almacenar datos persistentes (JDO y JPA)
- Acceder a recursos en la red (Fetch API)
- Cachear información (JCache)
- Enviar email (JavaMail)
- Procesar imágenes
- Gestionar usuarios
- Lanzar tareas planificadas o en background.

CREACIÓN

Se tiene la clase servlet y los ficheros xml.

Para probar la app se va con el navegador a localhost:8080/nombre.

Cuando esté listo se sube a appspot.com.

Pasos:

1. Crear proyecto: crear la estructura del directorio.
2. Crear clase servlet que extienda a Http Servlet.
3. Fichero de despliegue XML
4. Fichero de config. De GAE en XML
5. Probar en local
6. Subir a Google.

SERVLET

Un servlet HTTP es una clase de aplicación que puede procesar y responder solicitudes web.

JDO: proporciona librerías para gestión de usuarios, login y almacenamiento de datos.

JDO permite guardar objetos en Java.

Los datos a guardar se marcan con @persistent antes de la línea del atributo.

JSP

Como el código HTML es difícil de mantener, es conveniente utilizar un sistema de plantillas donde la interfaz de usuario esté diseñada en varios archivos independientes. Al cargar un JSP por primera vez, el servidor de desarrollo lo convierte en código fuente Java que más tarde será compilado a bytecode.

JDO – ALMACÉN DE DATOS

Utilizaremos la interfaz JDO para la recuperación y publicación de los mensajes de los usuarios en el almacén de datos para App Engine.

Se deben utilizar anotaciones Java para describir cómo se deben guardar las instancias en el almacén de datos.

JDO también permite almacenar objetos Java utilizando Access Platform y serializando los objetos instanciados.

Se debe utilizar `@Persistent` con los atributos para indicar que son propiedades, y `@PrimaryKey` para PKs en BBDD.

ARCHIVOS ESTÁTICOS

AppEngine considera todos los archivos del directorio WAR como archivos estáticos.

Para mostrarlos eficientemente, App Engine muestra los archivos estáticos desde servidores independientes.

Para almacenar un objeto en el almacén de datos, se ejecuta el método `makePersistent()` de `PersistenceManager`. Puede después eliminarse con `deletePersistent()`.

Para realizar queries sobre JDOQL se puede llamar con `pm.newQuery` que retorna un objeto de tipo query.

PLANIFICADOR DE TAREAS CON CRON

El App Engine Cron Service permite planificar tareas que se ejecutarán en un momento determinado. Sintaxis:

```
("every"|ordinal) (days) ["of" (monthspec)] (time)
```

MEMCACHÉ JAVA API

Se utiliza una caché distribuida con la interfaz JCache. Proporciona una interfaz en forma de mapa que permite almacenar y recuperar valores, inspeccionar contenido y obtener estadísticas, utilizar funciones de escucha, etc.

OTROS

- **URL Fetch Api:** GAE permite realizar conexiones HTTP y HTTPS con URLConnection. Permite un acceso sencillo a los contenidos con Streams de input y output.
- **Mail Java api:** Se pueden enviar mails a nombre de la aplicación de App Engine desarrollada.
- **Task queue:** Se puede realizar una cola de tareas que utiliza un servlet para ejecutarlas.
- **Importar y exportar datos:** Se puede importar y exportar datos en ficheros CSV.

GOOGLE WEB TOOLKIT

Esencialmente es un framework para crear aplicaciones Ajax.

Permite crear interfaces gráficas estilo swing para funcionar por http.

Se actualiza asíncronamente, no hay que hacer refresh y tiene buena escalabilidad.

Sin embargo, tiene problemas de compatibilidad entre navegadores y es lento debido a JS.

Solución: el código java es compilado a JavaScript. Está muy optimizado y se descarga solo el JS que vaya a correr el navegador.

RAID

Forma de distribuir o replicar datos entre unidades de almacenamiento. El raid es transparente al SSOO, éste solo ve un disco duro.

Niveles de raid:

- **Raid 0:** se distribuyen en partes iguales los datos entre dos o más discos duros. No hay redundancia. Sólo mejora la velocidad de escritura.
- **Raid 1:** Son discos espejo. Todos los discos duros tienen toda la información. No es más rápido pero tiene copias de seguridad.
- **Raid 5:** 3 discos mínimo. Se reparten los datos entre los tres, y cada uno de ellos guarda info. Sobre la paridad de los bloques almacenados en los otros discos duros. Gran velocidad y escritura y posibilidad de recuperar datos por los bits de paridad.

APA – TEMA 3 – UNIDAD 3

PREGUNTAS DE EXAMEN

1. APP Engine for Java permite el uso de estándar/es para el almacén de datos...
El API JDO (Objetos de datos en Java) y JPA (API de persistencia Java).
2. Señala cuál es la afirmación falsa respect a AJAX:
Permite la compatibilidad entre todos los navegadores.
3. Explicar y describir SaaS, PaaS, IaaS y un ejemplo de cada uno.
4. Características que aporta GAE.
5. Justifique y comente los problemas existentes en la contratación de desarrollo a medida, y como el cloud computing puede ayudar a solucionar dichos problemas.