Programa de formación MLDS

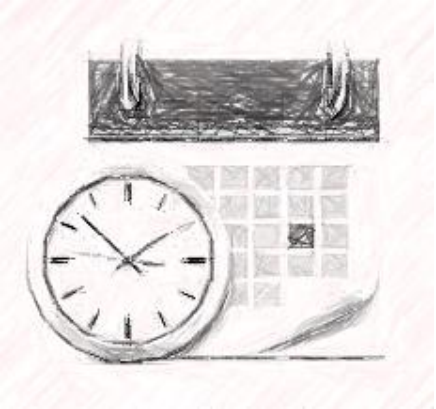
UNIVERSIDAD NACIONAL DE COLOMBIA

Ben Chams - Fotolia

Módulo
BIG DATA
Nube, Bases de Datos en la Nube

Por
Ing. Elizabeth León Guzmán, Ph.D.

Agenda



- Computación en la nube (cloud computing)
- Bases de Datos en la Nube
- ACID en Bases de Datos relacionales
- Bases de datos NoSQL
- Taller de Datastore de Google

Datos en la Web

- Crecimiento exponencial de la Web
- Incremento de fuentes de datos
- Problemas de almacenamiento y manipulación de los datos con modelos relacionales

Computación en la nube

Unir todos los servidores independientes y entre todos utilizar el poder de computo, que a la vez no dependa

de cada servidor.

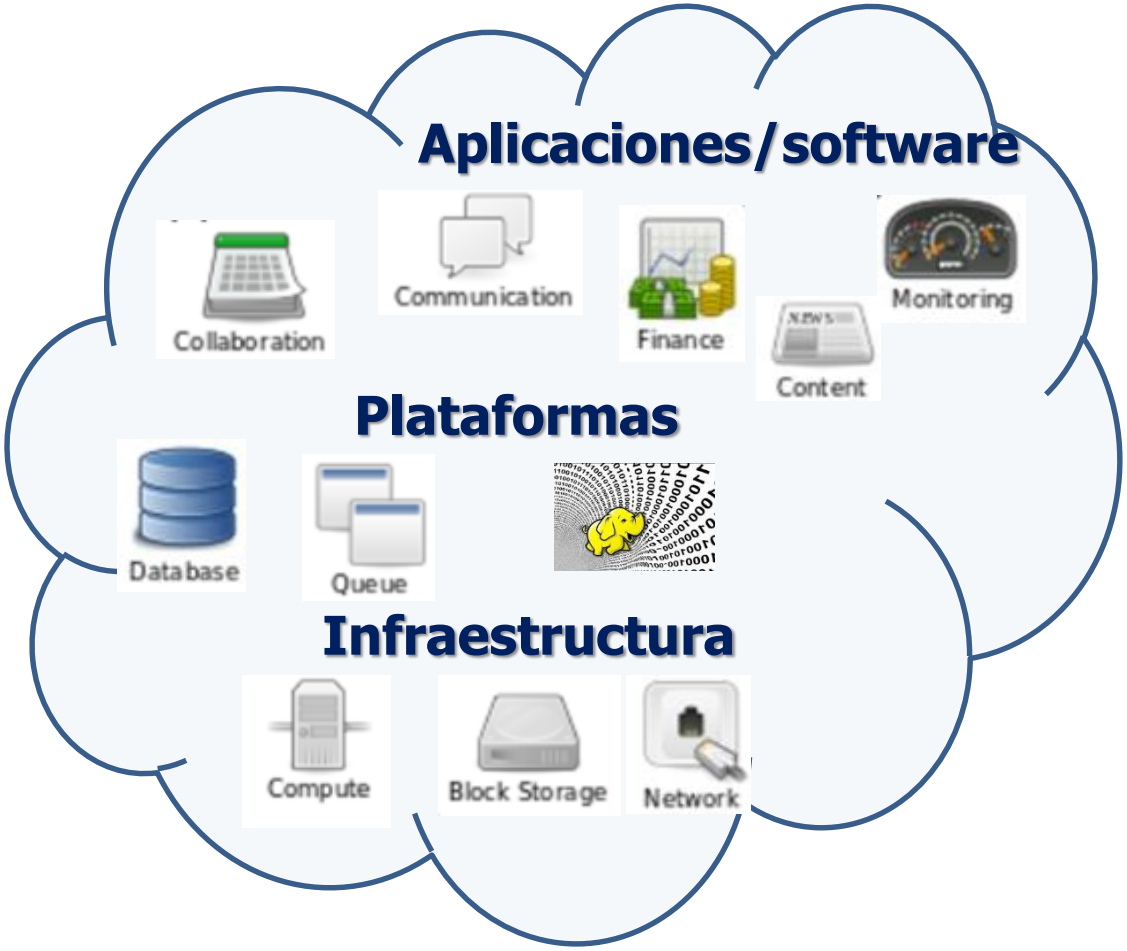
Computación en la nube

"Paradigma" que permite ofrecer servicios de computación a través de Internet:

- redes,
- servidores,
- almacenamiento,
- aplicaciones

Computación en la nube

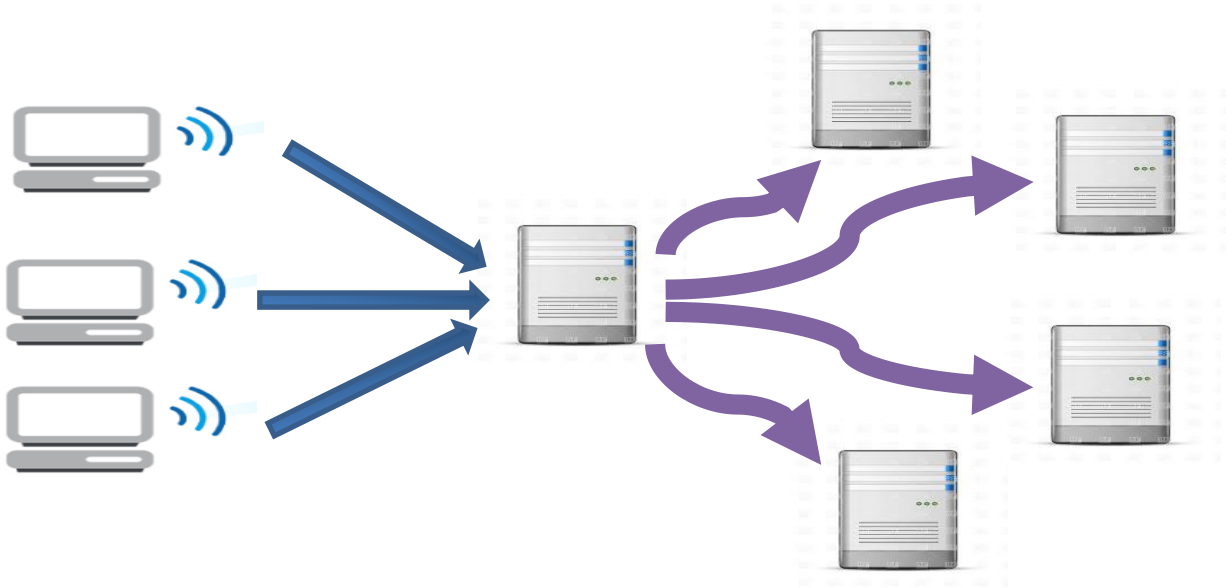
- Infraestructura como servicio (IaaS),
- plataforma como servicio (PaaS),
- software como servicio (SaaS).



Computación en la nube

Arquitectura

Las peticiones llegan a varios servidores, lo que produce el balanceo de carga.



Computación en la nube

Evolución natural de:

- adopción generalizada de la virtualización,
- arquitectura orientada a servicios

Usuarios finales ya no necesitan tener conocimientos o control sobre la infraestructura de tecnología "en la nube"

Computación en la nube

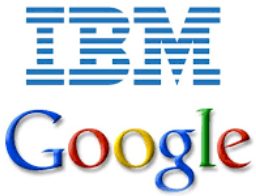
Salesforce.com, **1999** entrega de aplicaciones empresariales a través de una página web

2002

2006

Elastic Compute Cloud de Amazon (EC2)
Alquiler de equipos

2007



2006

Cloud computing
A la vanguardía

2007

2009

Windows Azure

2010

Proliferción de servicios

[Eucalyptus](#) en **2008**

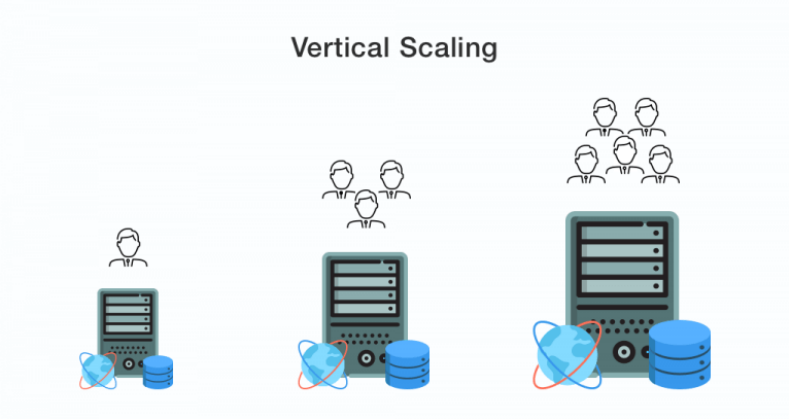
OpenNebula, Google APP

2011

Computación en la nube

Características:

- **Escalabilidad/elasticidad:** Capacidad para adaptarse al crecimiento (vertical y horizontal)



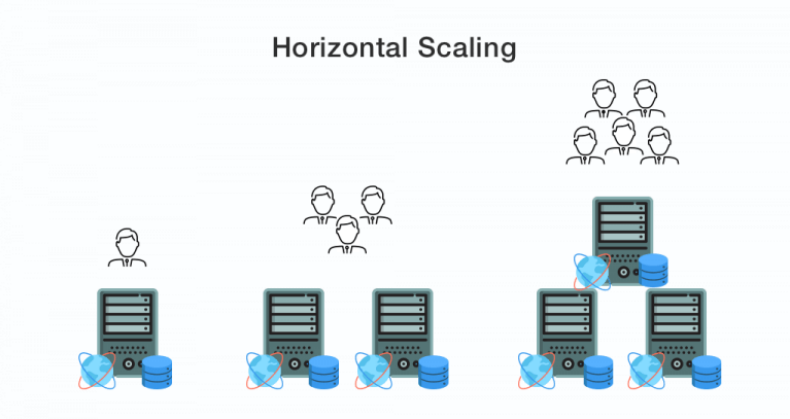


Figura tomada de Jignesh Solanki. "Vertical Scaling and Horizontal Scaling in AWS ". 2018

Computación en la nube

Características:

- **Tolerancia a fallos:** Funcionamiento del Sistema cuando hay fallas (replica de la información)

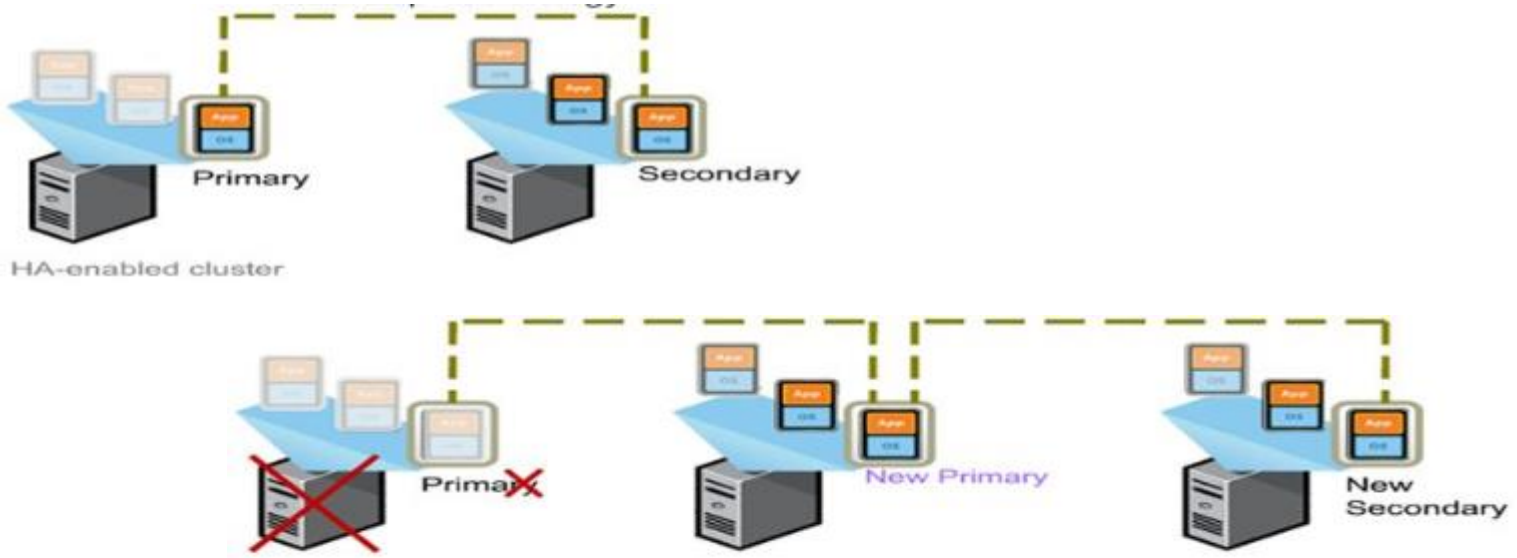
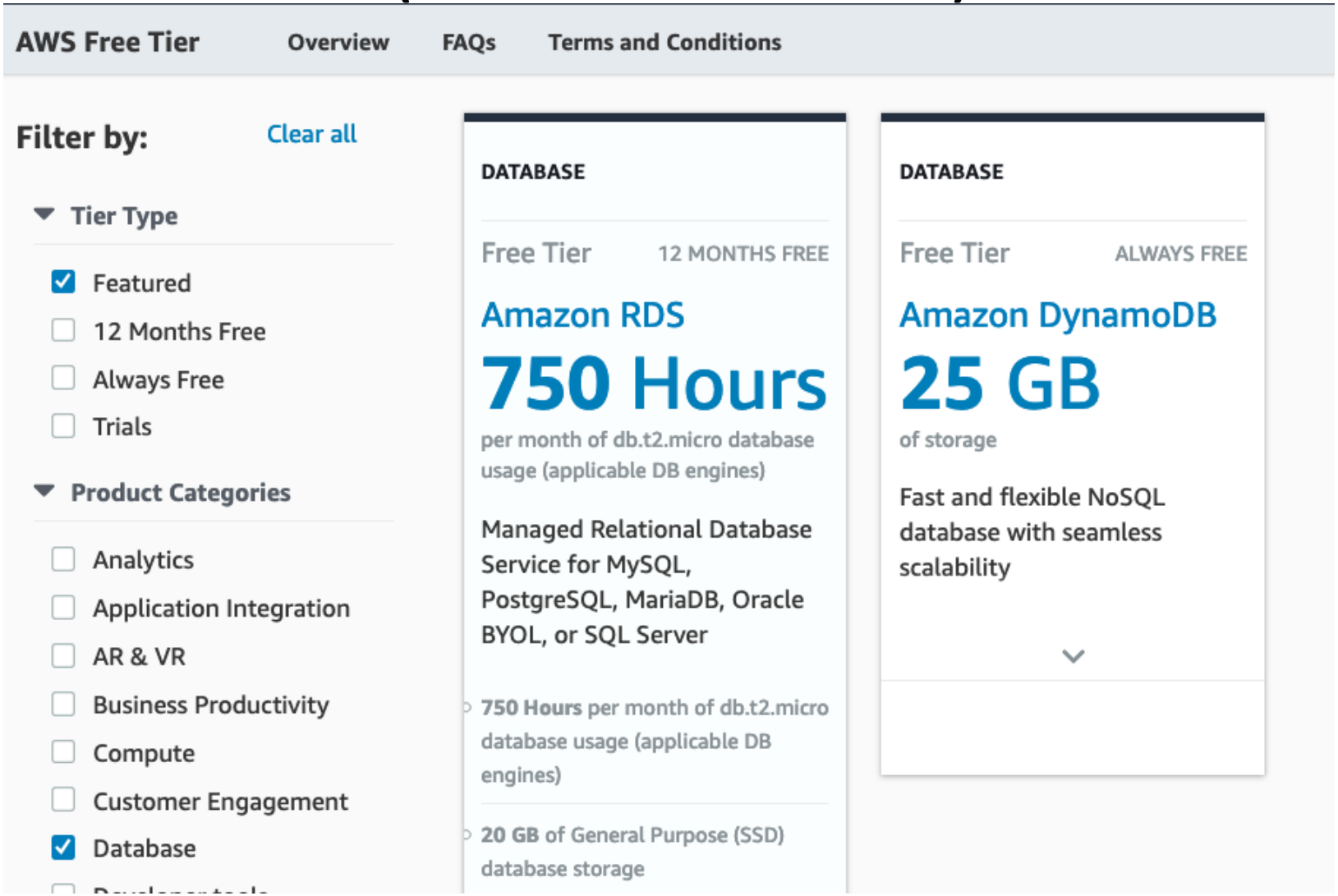


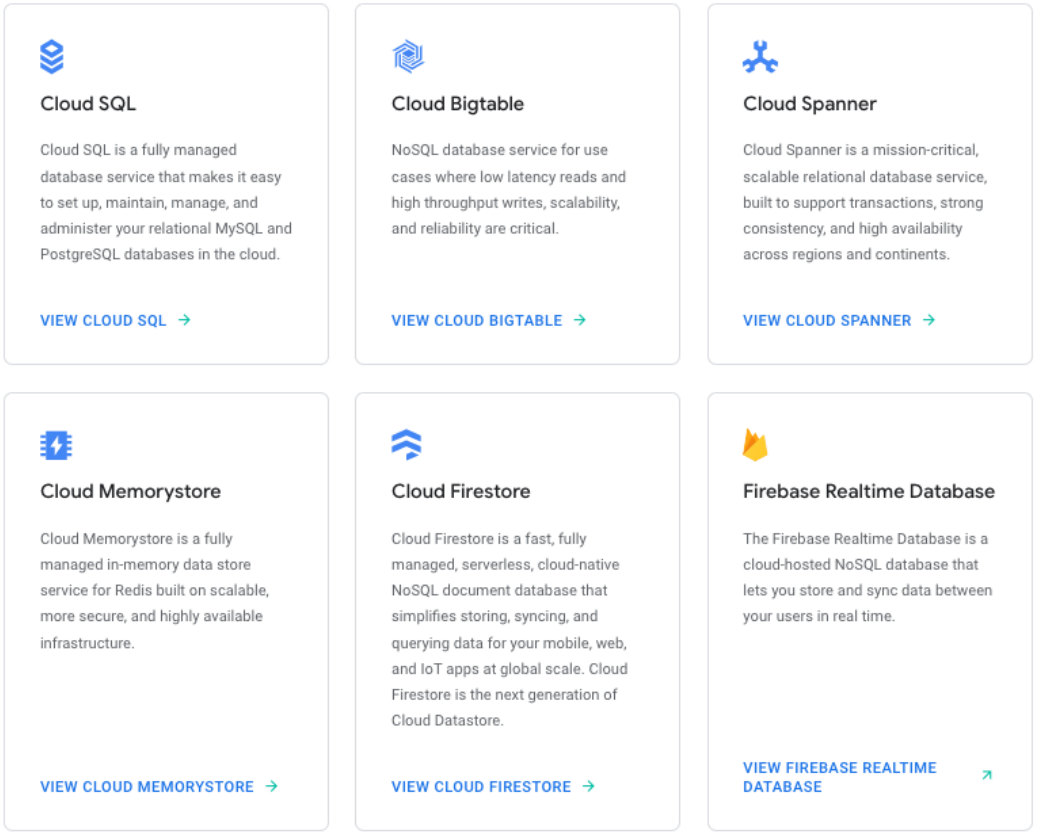
Figura tomada de Cloud Interoperability & Standards. Naomy Davidson

Amazon Web Services (Bases de Datos)



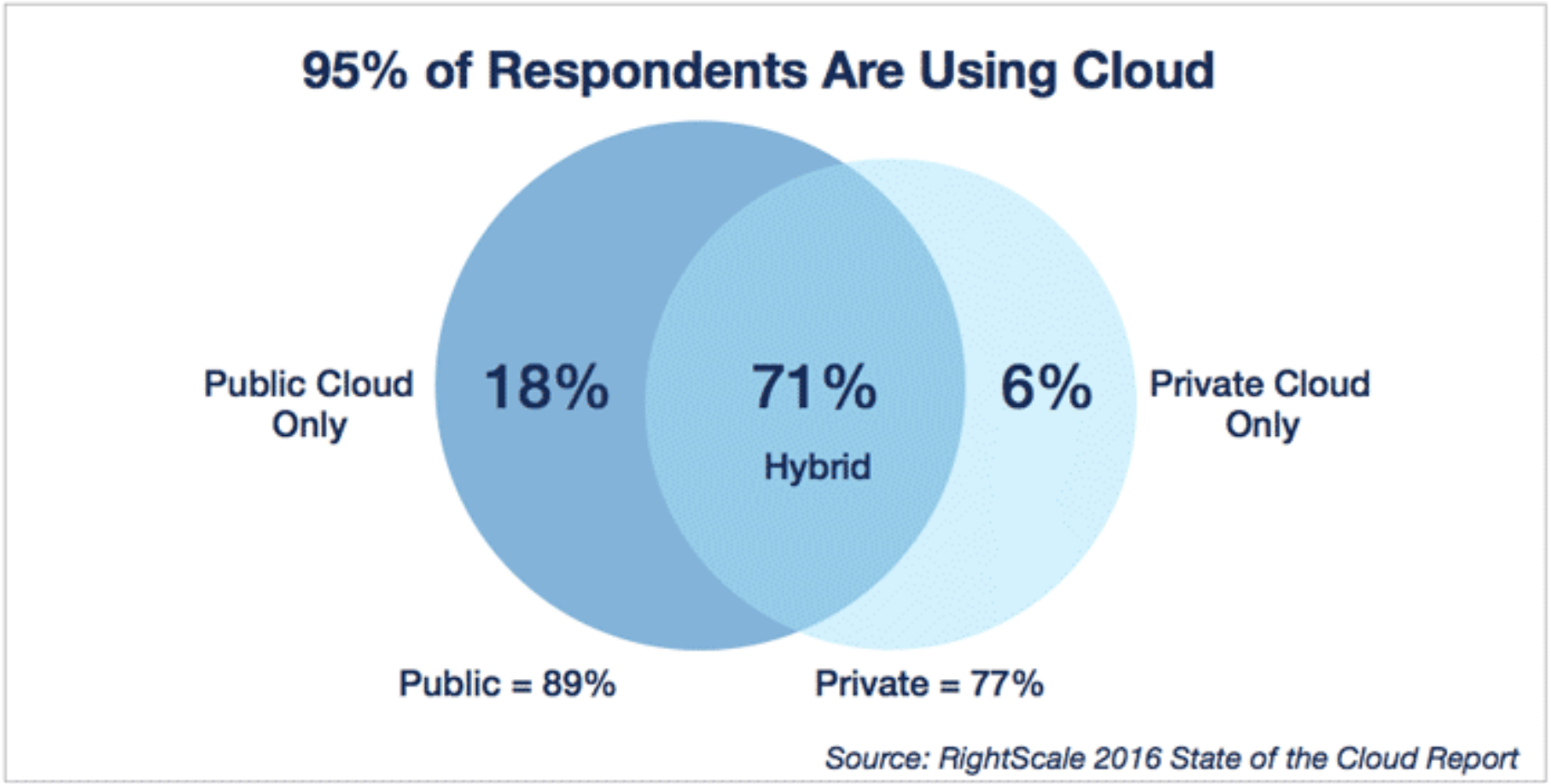
Google Cloud Platform

https://cloud.google.com/products/databases/



Computación en la nube

Cloud Survey



Computación en la nube

- Responsabilidad del almacenamiento de datos y su control en el proveedor. ¿Volver al pasado? … ¿sistemas centralizados de los años 50, 60?
- Privacidad

  "computación en nube pone en peligro las libertades de los usuarios, porque éstos dejan su privacidad y datos personales en manos de terceros"
- Seguridad

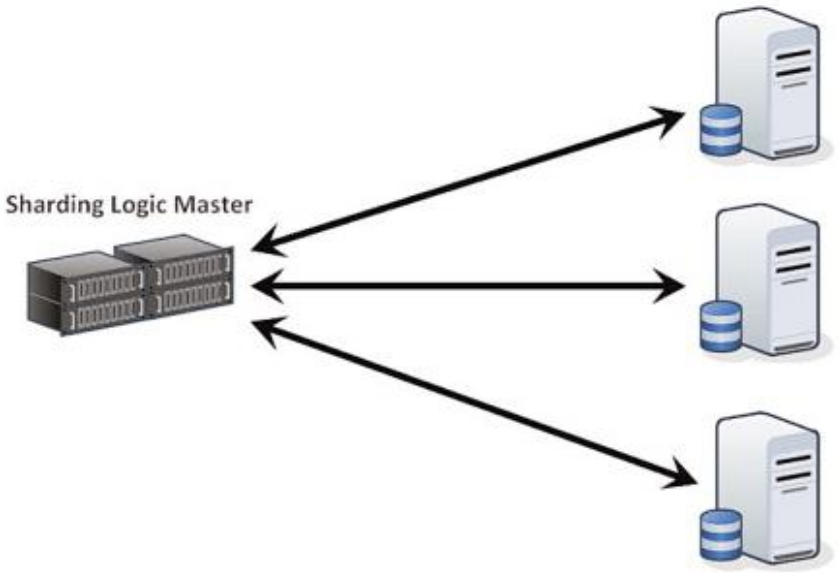
Bases de datos en la nube

- Concepto de computación en la nube: se ejecutan desde la nube.
- Almacenamiento de datos no estructurados

Nueva generación de DBMS

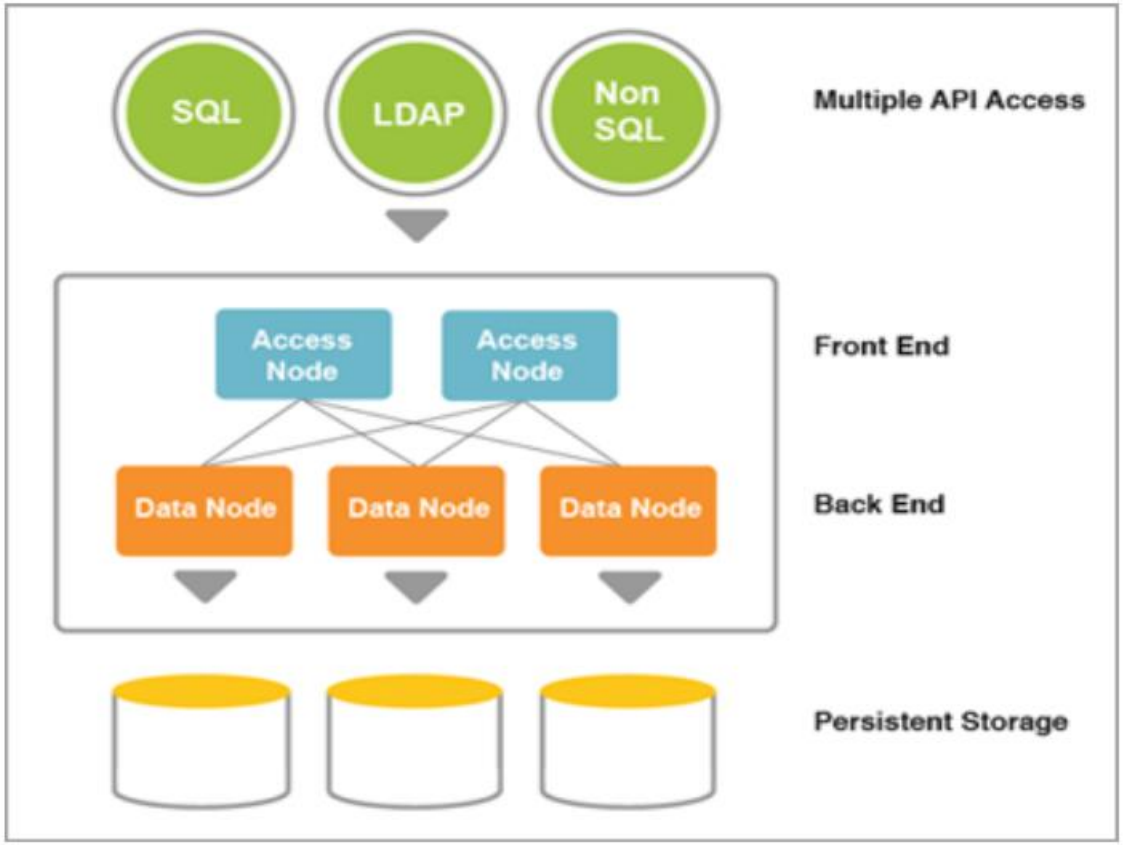
Bases de datos en la nube

- Escalamiento horizontal basado en el particionamiento de los datos.
- Se ejecutan desde infraestructuras que garantizan una alta disponibilidad, escalabilidad y balanceo de carga.



Arquitectura

- Consolas web o APIs para la administración de la plataforma.
- Datos guardados en datacenters.
- Proveedor del servicio responsable de administrar la plataforma
- Las arquitecturas dependen del proveedor



Fuente: http://xeround.com/blog/2011/10/in-network-cloud-database-architecture-benefits-performance-other-considerations

Modelo de Datos Relacional (SQL)

- Datos estructurados cuidadosamente (esquema)

  Estudiante (código, nombre, apellido, teléfono, dirección)

- Basado en tablas que representan conjunto de relaciones
- Restricciones como llaves primarias y foráneas

Modelo Relacional
Relación
Estructura Básica

Relación → Tabla

Columna
Campo
Atributo

Estudiante

Codigo	Nombre	Apellido	Edad
256678	Pepito	Perez	23
279909	Maria	Suarez	22

fila
registro
tupla

Modelo Relacional
Relación

ESTUDIANTE

Codigo	Nombre	Apellido	Edad
256678	Pepito	Perez	23
279909	Maria	Suarez	22

Conjunto de todos los códigos → D_1 D_2 D_3 D_4

Cada fila consisten en una tupla *(v1,v2,v3,v4)* donde *v1* esta en el dominio D_1
v2 esta en el dominio D_2 ...

Por lo tanto:

$$ESTUDIANTE(codigo, nombre, apellido, edad) \subset D_1 \text{ X } D_2 \text{ X } D_3 \text{ X } D_4$$

Modelo Relacional

- Algebra Relacional -> SQL (Structured Query Language)

LIBRO

lib_id	lib_nombre	lib_año
100051	Cien años de soledad	1967
200032	La vorágine	1924
300033	María	1867
401156	Cóndores no se entierran todos los días	1971

Modelo Relacional

- Algebra Relacional -> SQL (Structured Query Language)

LIBRO

lib_id	lib_nombre	lib_año
100051	Cien años de soledad	1967
200032	La vorágine	1924
300033	María	1867
401156	Cóndores no se entierran todos los días	1971

Proyección

$$\pi_{lib\_nombre}\,(LIBRO) = \left\{ \begin{array}{c} \langle Cien\ años\ de\ soledad \rangle, \\ \langle La\ vorágine \rangle, \\ \langle María \rangle, \\ \langle Cóndores\ no\ entierran\ todos\ los\ días \rangle \end{array} \right\}$$

Modelo Relacional

- Algebra Relacional -> SQL (Structured Query Language)

LIBRO

lib_id	lib_nombre	lib_año
100051	Cien años de soledad	1967
200032	La vorágine	1924
300033	María	1867
401156	Cóndores no se entierran todos los días	1971

Proyección en SQL

$$\pi_{lib\_nombre}\ (LIBRO)$$

SELECT lib_nombre FROM LIBRO

Modelo Relacional

- Algebra Relacional -> SQL (Structured Query Language)

LIBRO

lib_id	lib_nombre	lib_año
100051	Cien años de soledad	1967
200032	La vorágine	1924
300033	María	1867
401156	Cóndores no se entierran todos los días	1971

Selección

$$\sigma_{lib\ año>1970}\ (LIBRO) = \{\langle 401156, Cóndores\ no\ entierran\ todos\ los\ días, 1971\rangle\}$$

Modelo Relacional

- Algebra Relacional -> SQL (Structured Query Language)

LIBRO

lib_id	lib_nombre	lib_año
100051	Cien años de soledad	1967
200032	La vorágine	1924
300033	María	1867
401156	Cóndores no se entierran todos los días	1971

Selección en SQL

$$\sigma_{lib\ año>1970}\ (LIBRO) = \{\langle 401156, Cóndores\ no\ entierran\ todos\ los\ días, 1971\rangle\}$$

SELECT * FROM *LIBRO* WHERE *lib_año>1970*

Modelos tradicionales

Atomicidad (Atomicity)

Consistencia (Consistency)

Aislamiento (**I**solation)

Durabilidad (Durability)

Modelos tradicionales

Atomicidad (Atomicity)

Las transacciones de deben realizar completamente. Todas las tareas o ninguna.

Modelos tradicionales

Consistencia (Consistency)

Se refiere a la integridad de los datos. Garantiza que los datos son exactos, que no existen inconsistencias. Los datos disponibles para consulta son los que deben ser.

Modelos tradicionales

Aislamiento (**I**solation)

Asegura que una operación sobre los datos (insert, update, delete) no debe afectar otras operaciones. Cada operación sobre los datos debe ser independiente y no generar errores.

Modelos tradicionales

Durabilidad (**D**urability)

Asegura que las operaciones sobre los datos persisten. La operación no se puede deshacer.

Nuevos retos para las BD

- Proliferación de sistemas de almacenamiento.
- El incremento de datos y usuarios hacen que los datos se almacenen en servidores distribuidos, lo que dificulta mantener las relaciones entre los datos.
- Propiedades tradicionales (ACID) no son apropiadas para todas las aplicaciones (web),

Eventualmente consistente vs ACID

Requerimientos
Alto desempeño

- Aplicaciones web ejecutan miles de transacciones por segundo:
  - Publicidad (Ad Serving)
  - Servicio de email (Email Services)
  - Carritos de compra (Shopping Carts)
  - Operaciones financieras (Financial Trades)
- Bases de datos tradicionales no capacitadas para el manejo de esas “tasas” de generación de datos.

Requerimientos
Altamente Disponible
Soluciones distribuidas

- Para aplicaciones web la disponibilidad es criticamente importante
- Se puede sacrificar la calidad de los datos a cambio de disponibilidad (búscadores web, consistencia vs disponibilidad).
- Replicación en múltiples nodos (en el caso de una caída de un nodo)

Requerimientos
Nuevos modelos de programación

- Datos no relacionales (Json: *JavaScript Object Notation*)
- SQL puede ser lento y muy complejo
- SQL requiere esquema, ahora definir esquema de los datos puede ser díficil, y algunas veces los datos no conforman un esquema.

A JSON Document

```
{ sku: "00e8da9d",
  type: "Film",
  ...,
  asin: "B000P0J0AQ",

  shipping: { ... },

  pricing: { ... },

  details: {
    title: "The Matrix",
    director: [ "Andy Wachowski", "Larry Wachowski" ],
    writer: [ "Andy Wachowski", "Larry Wachowski" ],
    ...,
    aspect_ratio: "1.66:1"
  },
}
```

Source: http://docs.mongodb.org/ecosystem/use-cases/product-catalog/

Modelo de Datos

Not Only SQL

- No solo SQL, pueden usar lenguajes parecidos al SQL
- Habilidad de distribuir los datos sobre nodos garantizando disponibilidad, mientras mantiene escalabilidad en niveles aceptables, e ignorando la Consistencia de los datos (integridad)
- Modelos: Clave-valor, documentos, grafos

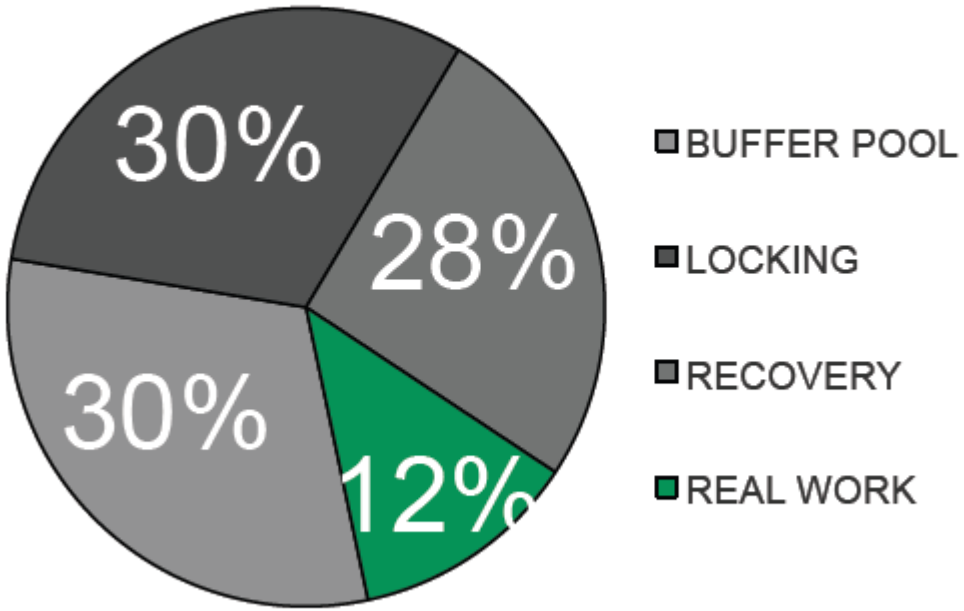
NoSQL
Ventajas

- **Ejecución en máquinas con pocos recursos:** poca computación
- **Escalabilidad horizontal:** para mejorar el rendimiento basta con añadir más nodos
- **Manejan gran cantidad de datos:** estructura distribuida
- **No generan cuellos de botella:** no realiza transcripción de las sentencias

Modelo de datos	Máquina Virtual	Base de datos como servicio
SQL	Oracle Database IBM DB2 Ingres PostgreSQL MySQL NuoDB GaianDB	Amazon Relational Database Service (MySQL) Microsoft SQL Azure (MS SQL) Heroku PostgreSQL como servicio (compartido o dedicado) Clustrix Base de datos como servicio Xeround Cloud Database - MySQL front-end EnterpriseDB Postgres Plus Cloud Database GaianDB
NoSQL	*CouchDB *Apache Cassandra *-Neo4J *-MongoDB *Disponibles en (Amazon EC2 * y Microsoft Azure -)*	Amazon DynamoDB, Amazon SimpleDB Cloudant Data Layer (CouchDB) Database.com por SalesForce Google App Engine MongoDB

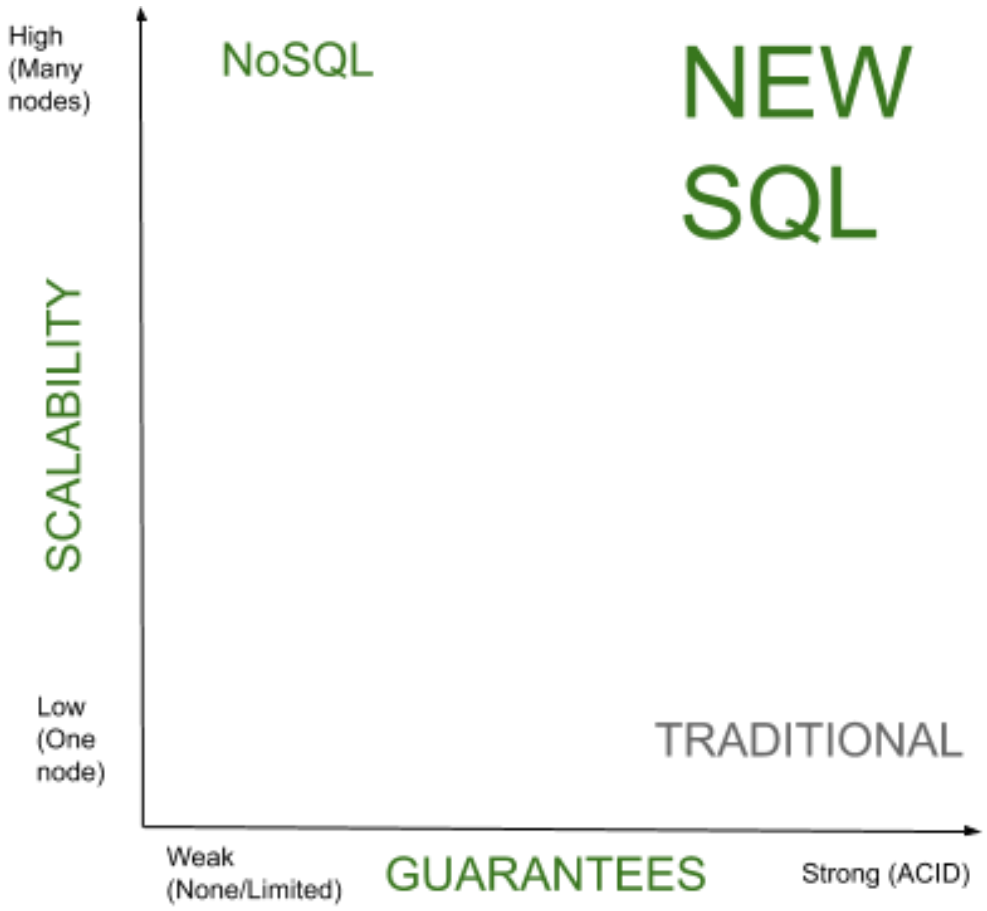
TRADITIONAL DBMS

Measured CPU Cycles



OLTP THROUGH THE LOOKING GLASS, AND WHAT WE FOUND THERE
SIGMOD, pp. 981-992, 2008.

NewSQL


High
(Many
nodes)
NoSQL
NEW
SQL
SCALABILITY
Low
(One
node)
TRADITIONAL
Weak
(None/Limited)
GUARANTEES
Strong (ACID)

NewSQL

- DBMS moderno que pretende proveer la escalabilidad de los modelos NoSQL (lectura/escritura) manteniendo ACID
- Diseñadas para operar en clusters distribuidos
- H-Store (MIT)
- Google spanner, clustrix, voltDB, MemSQL

Referencias

[1] Samuel Madden.Massachusetts Institute of Technology

Tackling the Challenges of Big Data. 2014

[2] Ashle Matthew. "How Will The Database Incumbents Respond To NoSQL And NewSQL?“2011

[3] Stavros Harizopoulos y otros. OLTP Through the Looking Glass, and What We Found There. 2008

[4] Marc Seeger. "Key-Value Stores: a practical overview" 2009