UTN.BA FACULTAD REGIONAL BUENOS AIRES
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

Centro de e-Learning

Diplomatura en Bases de Datos

Módulo 3 - Unidad 4

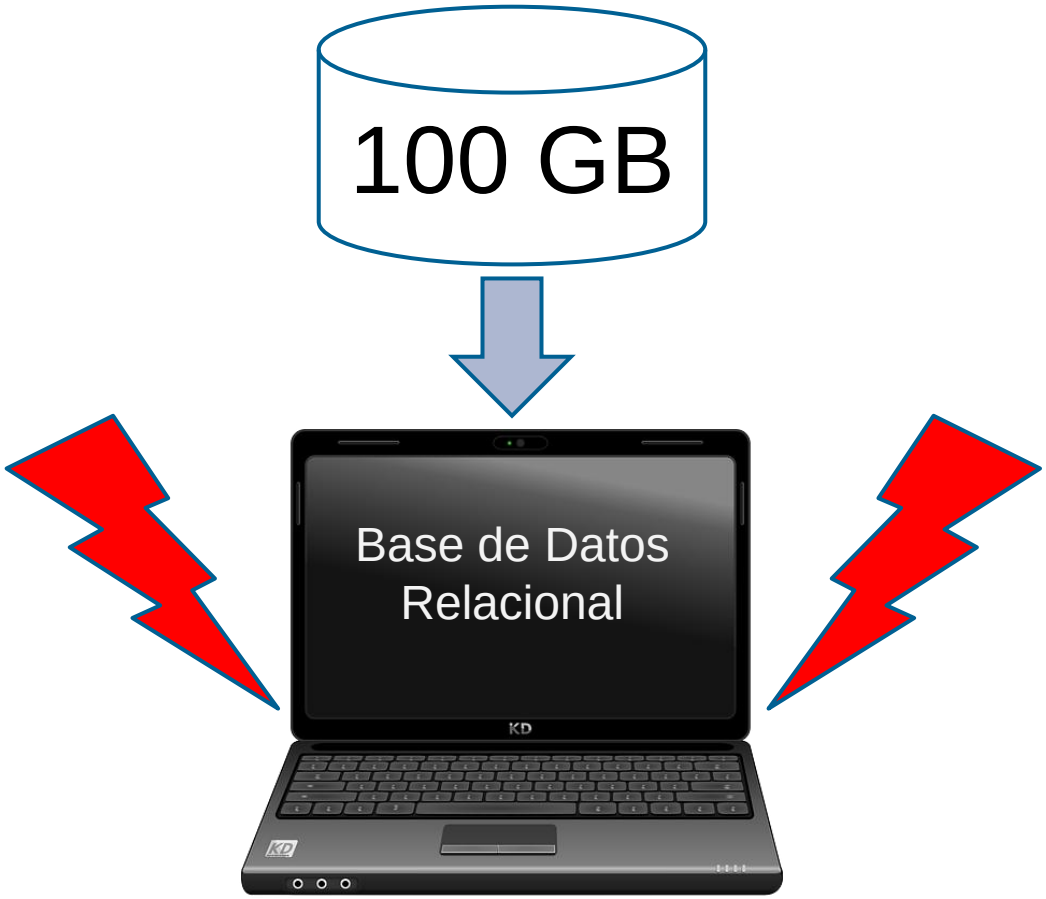
www.sceu.frba.utn.edu.ar/e-learning

Agenda

- *¿Qué es Big Data?*
- *¿Qué es Hadoop?*
- *¿Qué es HDFS?*
- *¿Qué es Map Reduce?*
- Herramientas de bases de datos para Big Data
  - Pig
  - Hive

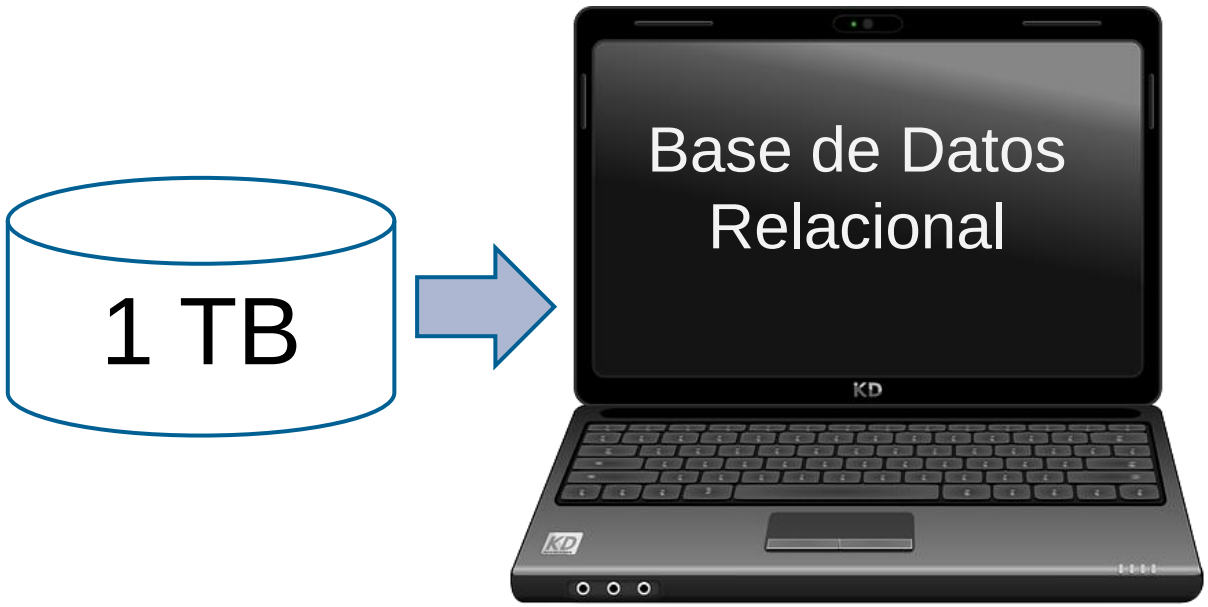
¿Qué es Big Data?

¿Qué es Big Data?



¿Qué es Big Data?

Incrementamos la capacidad

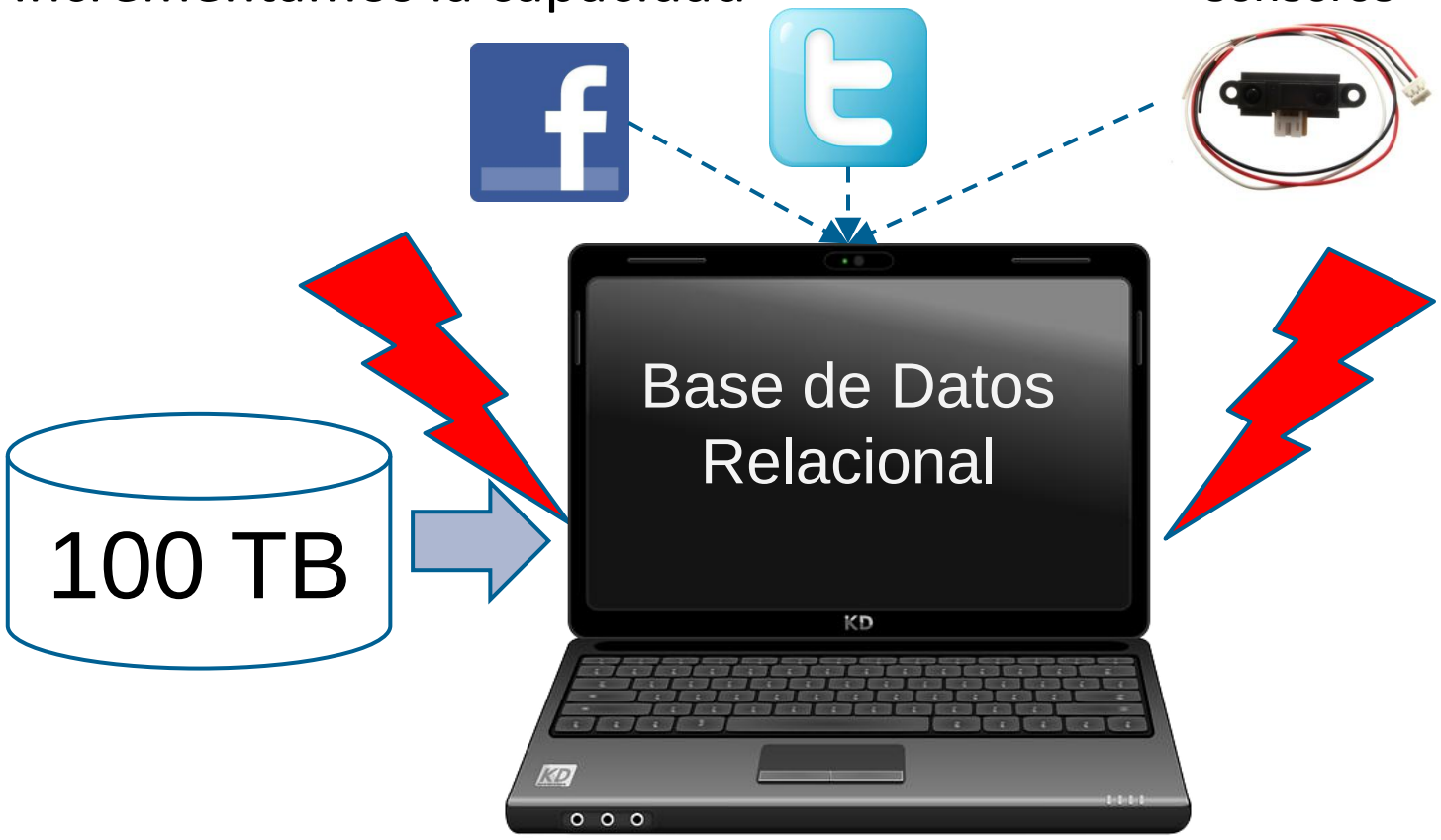


¿Qué es Big Data?

Incrementamos la capacidad

¿Qué es Big Data?

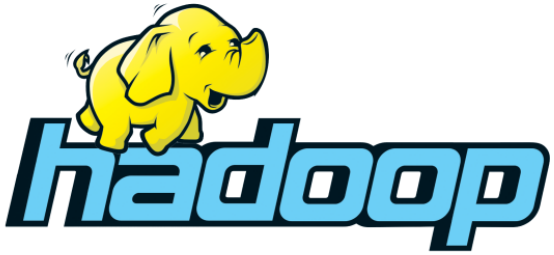
Incrementamos la capacidad



¿Qué es Big Data?

- *Frontera móvil*
- *Lo que no se puede manejar cómodamente con bases relacionales*
  - *Por tiempo*
  - *Por tamaño*
  - *Por complejidad*

¿Qué es Hadoop?



- Proyecto open source
- Framework escrito en Java
- Optimizado para manejar:
  - cantidades masivas de datos (procesamiento en paralelo)
  - variedad de datos (estructurados y no estructurados)
  - hardware barato
- operación por lotes (batch) - respuesta no inmediata
- fiabilidad proporcionada a través de la replicación

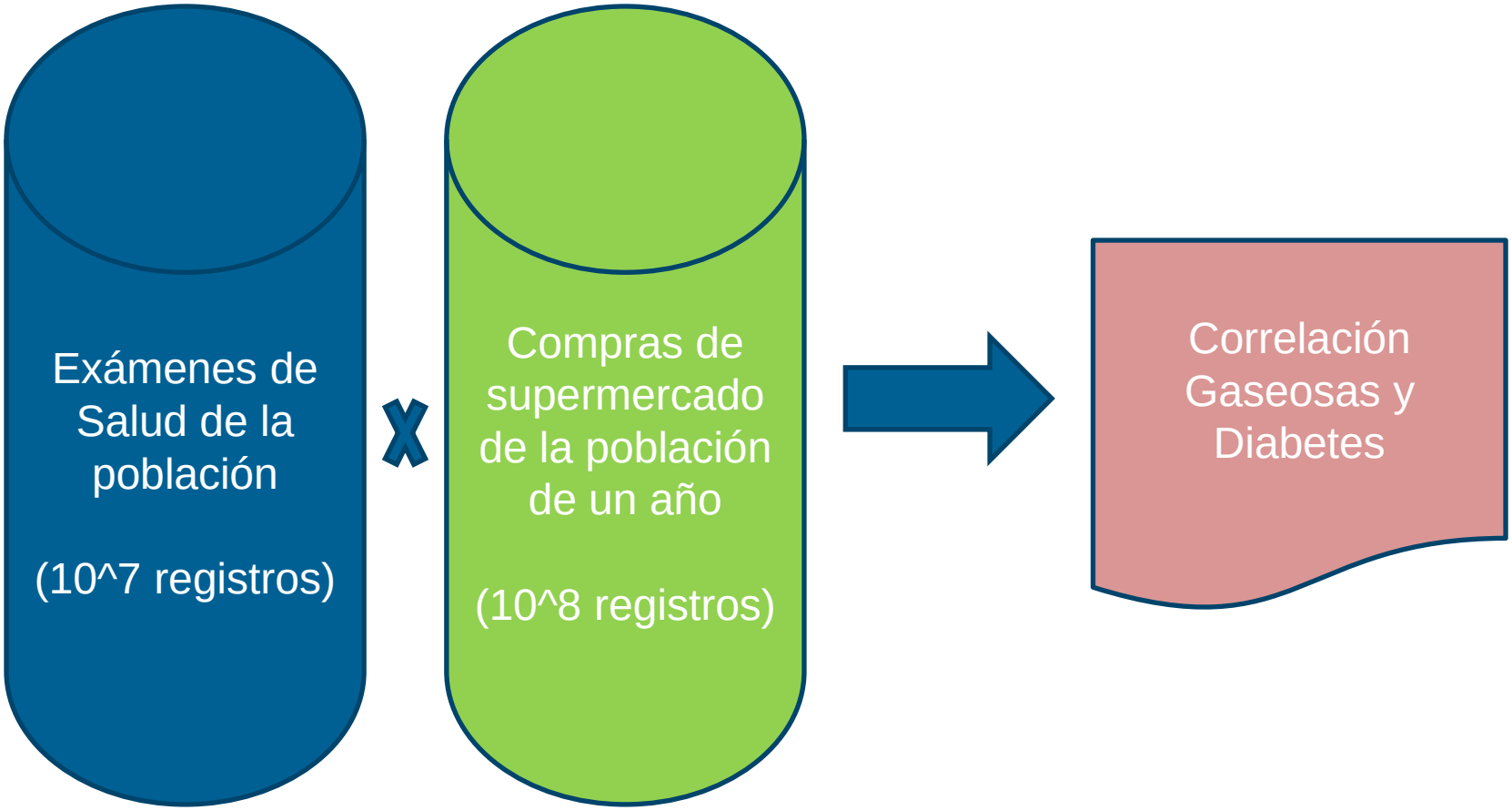
¿Qué es HDFS?

- *Hadoop Data File System*
- *Todos los datos se graban en varios servers*
- *El usuario no lleva control directo de que está donde.*
- *Orientado a que se procese en paralelo*

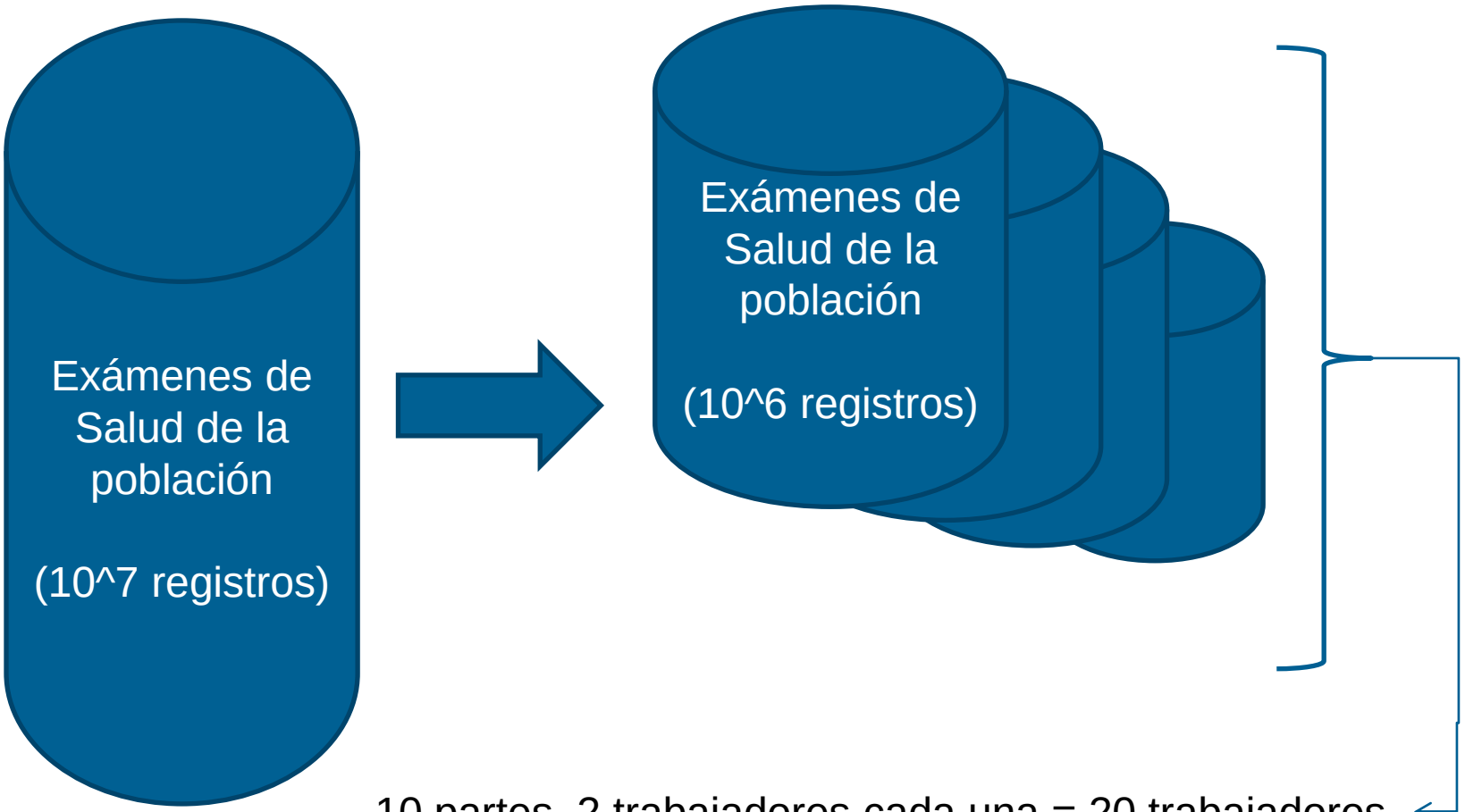
¿Qué es Map Reduce?

- *Registros del tipo clave - valor*
- *Ejemplo de la corporación cuenta palabras*

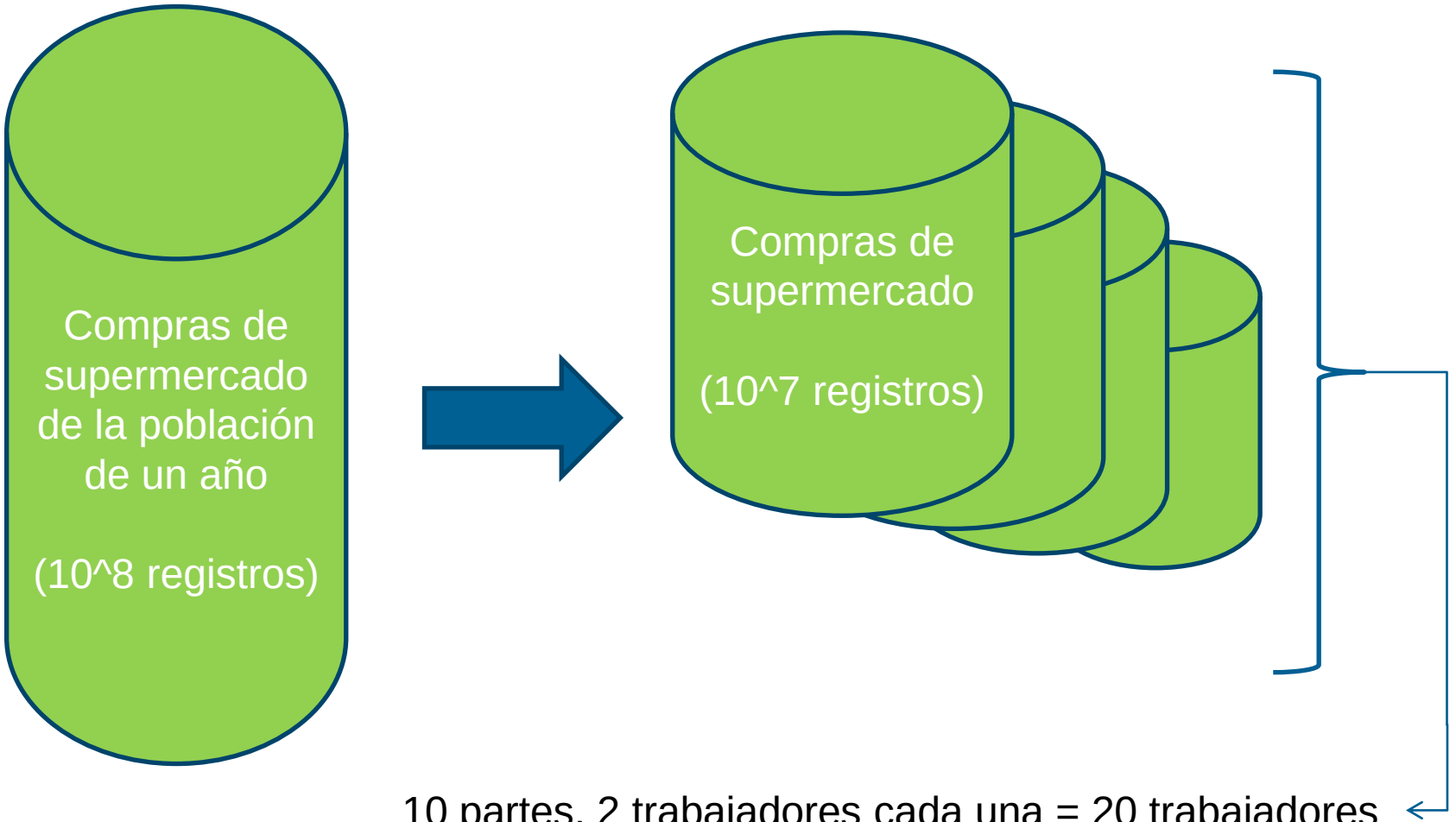
HDFS: Map – Reduce
Un ejemplo por favor!



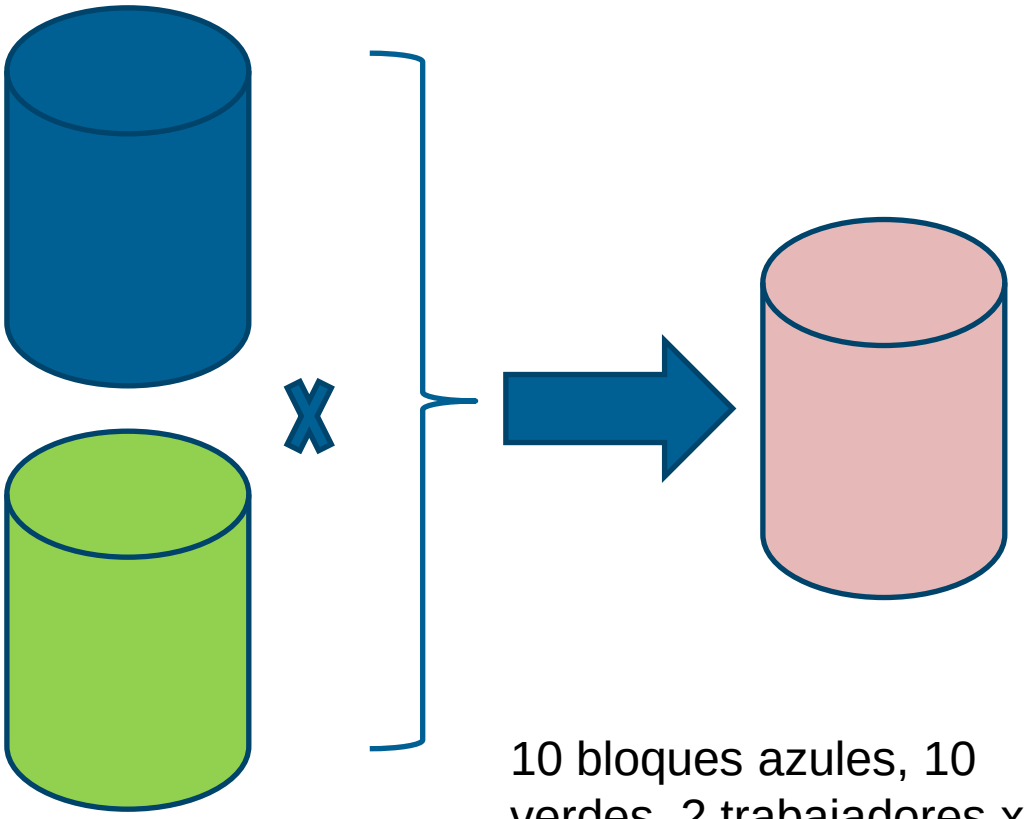
HDFS: Map – Reduce
Un ejemplo por favor! Paso 1: Map



HDFS: Map – Reduce
Un ejemplo por favor! Paso 2: Map



HDFS: Map – Reduce
Un ejemplo por favor! Paso 3 Reduce



Para las personas encontradas se anotan sus exámenes y su compras de gaseosas

10 bloques azules, 10 verdes, 2 trabajadores x combinación

200 Trabajadores

¿Siempre conviene usar Hadoop?

- Ventaja por procesar en paralelo
- Ventaja por la resiliencia
- Desventaja por la multiplicación
- Desventaja por la complejidad

¿Siempre conviene usar Hadoop?

- Ventaja por procesar en paralelo
- Ventaja por la resiliencia
- Desventaja por la multiplicación
- Desventaja por la complejidad

Siempre habrá una decisión por tomar

Ley de Amdhal

$$A = \frac{1}{(1 - F_m) + \frac{F_m}{A_m}}$$

- A: aceleración total
- Am: aceleración de la fracción paralelizable
- Fm: fracción paralelizable

Small data contra-ataca

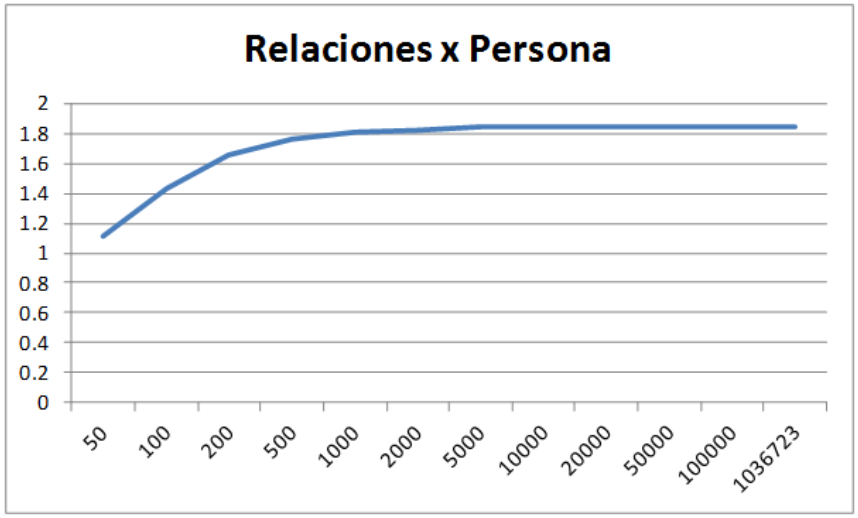
- ¿Por que no muestrear?
- ¿Qué es una muestra aleatoria?
- ¿Qué pasa si una muestra es no aleatoria?
- ¿Qué pasa si los datos no están todos disponibles al momento del análisis?
- ¿Cómo construir una muestra aleatoria?

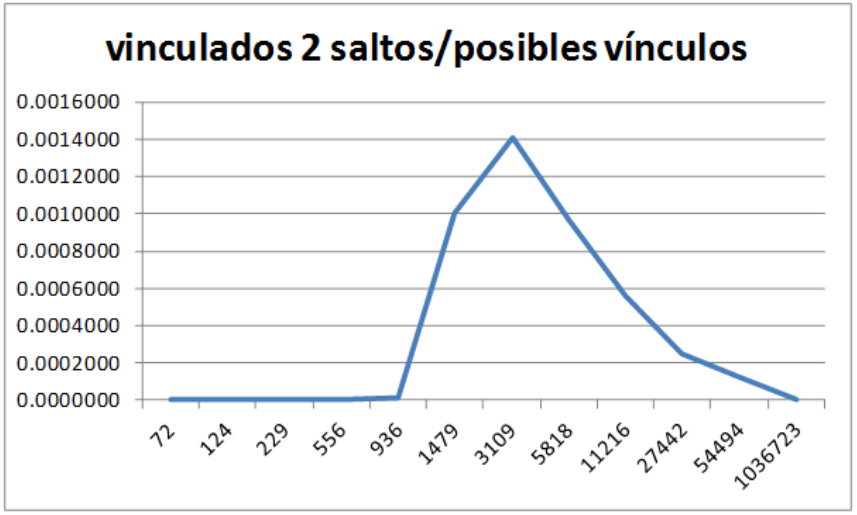
¿Todo problema es muestreable?

- Ejemplo del problema de los grados de separación

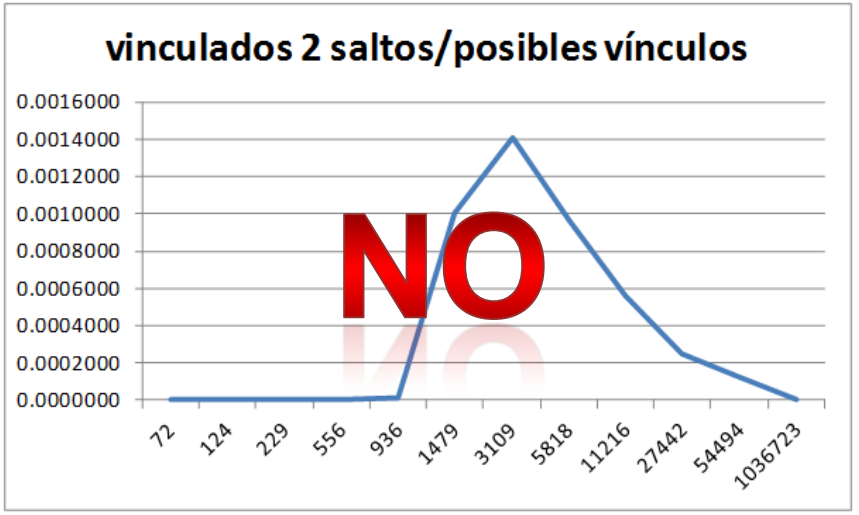
-

Ejemplo: Linked In





Ejemplo: Linked In



Proyectos relacionados con Hadoop

Más software para Big Data

- PIG
  - Principales usos
  - Principales comandos
  - Ejemplo
- Hive
  - Principales usos
  - Principales comandos
  - Ejemplo

PIG: Historia

- Lo concibieron en 2006 en Yahoo
- En 2007 Pig Latin en Apache Incubator
- Para hacer un join o un group by se repite mucho código Map-Reduce
- El propósito de PIG Latin es automatizar esa repetición ofreciendo una interface orientada al flujo de datos

PIG: Usos

- Manipulación de datos crudos
- Acciones básicas repetitivas
- Desde la consola
- Embebido en otros lenguajes

PIG: Tipos de datos

Tipos de Datos	Descripción	Ejemplo
Escalares		
int	Entero con signo de 32 bits	10
long	Entero con signo de 64 bits	Data: 10L or 10l Display: 10L
float	Punto flotante con 32 bits	Data: 10.5F or 10.5f or 10.5e2f or 10.5E2F Display: 10.5F or 1050.0F
double	Punto flotante con 64 bits	Data: 10.5 or 10.5e2 or 10.5E2 Display: 10.5 or 1050.0
Conjuntos		
chararray	Conjunto de caracteres en formato Unicode UTF-8	hello world
bytearray	Conjunto de bytes (BLOB)	
Tipos de datos complejos		
tuple	Un conjunto ordenado de campos (registro)	(19,2)
bag	Una colección de tuple (dataframe)	{(19,2), (18,1)}
map	Conjunto de pares clave, valor	[open#apache]

PIG: Principales Comandos

- Load: Carga datos de un archivo o directorio
- Filter: Filtra los datos de una tabla
- Join: Realiza el producto cartesiano
- Group: Agrupa registros por una clave
- Foreach: Calcula una función para los agrupados
- Order: Ordena los registros
- Store: Guarda el resultado a un archivo

PIG: Ejemplo

```
# Users = load 'users' as (name, age);
# Fltrd = filter Users by age >= 18 and age <= 25;
# Pages = load 'pages' as (user, url);
# Jnd = join Fltrd by name, Pages by user;
# Grpd = group Jnd by url;
# Smmd = foreach Grpd generate group, COUNT(Jnd) as clicks;
# Srtd = order Smmd by clicks desc;
# Top5 = limit Srtd 5;
# store Top5 into 'top5sites';
```

Hive: Principales Usos

- Permite correr consultas sobre datos distribuidos.
- Prepara el código map-reduce que necesita para realizar las consultas.
- Es compatible con realizar operaciones tipo Map-Reduce independientes.

Hive: Principales Comandos

- SELECTS and FILTERS
- GROUP BY
- JOIN

Hive: Principales Comandos

- SELECTS and FILTERS:

```
SELECT a.foo
FROM invites a
WHERE a.ds='2008-08-15';
```

Hive: Principales Comandos

- GROUP BY:

```
INSERT OVERWRITE TABLE events
SELECT a.bar, count(*)
FROM invites a
WHERE a.foo > 0
GROUP BY a.bar;
```

Hive: Principales Comandos

- JOIN:

```
FROM pokes t1
JOIN invites t2 ON (t1.bar = t2.bar)
INSERT OVERWRITE TABLE events
SELECT t1.bar, t1.foo, t2.foo;
```

Hive: Ejemplo

```
# CREATE TABLE docs (line STRING);
# LOAD DATA INPATH 'docs' OVERWRITE INTO TABLE docs;
# CREATE TABLE word_counts AS
# SELECT word, count(1) AS count FROM
# (SELECT explode(split(line, '\s')) AS word FROM docs) w
# GROUP BY word
# ORDER BY word;
```

- HBase
  - ¿Para que sirve?
  - Principales características
  - Usos comunes
  - Instalación
  - Ejemplo

HBase

¿Para que sirve?

- Base de datos distribuida
- Implementa compresión
- Fuertemente consistente
- Código abierto
- No-SQL
- Tolerante a fallas
- Busca rápida de datos infrecuentes en > 10^9
- No es un reemplazo completo de SQL
- En términos del teorema CAP es CP

HBase

Aplicaciones relevantes de su ecosistema:

- Apache Phoenix le agrega una capa de consulta SQL y un conector JDBC.

- Apache Trafodion proporciona un motor de consultas SQL y conectores JDBC y protección de transacciones ACID.

HBase

Principales características

- Escalabilidad linear y modular.
- Lecturas y escrituras consistentes.
- Segmentación automática y configurable de tablas
- Soporte a la tolerancia a fallas entre servidores regionales.
- Clases para usar map reduce desde tablas HBase
- APIs java para acceso desde clientes

HBase

Usos comunes

- Bloomberg para almacenar series temporales
- Facebook entre 2010 y 2018 para la plataforma de mensajería
- Flipkart para el índice de búsqueda y la información de usuarios
- Airbnb para el cálculo de flujo ent iempo real AirStream
- Spotify para los trabajos de aprendizaje automático
- Tuenti para la plataforma de mensajería
- Imgur para su sistema de notificaciones
- Grupo Alibaba, Salesforce, Sears…

HBase

Instalación

- Descargar Hbase:
  cd /home/hadoop/Descargas wget
  http://apache.uvigo.es/hbase/2.1.1/hbase-2.1.1-bin.tar.gz

- Descomprimir
  tar xvf apache-hbase-2.1.1-bin.tar.gz

- Mover a la carpeta de los programas Hadoop
  su –
  mv /home/hadoop/Descargas/hbase-2.1.1 /opt/hbase

HBase

Instalación II

- Modificar variables de entorno:

  gedit /home/hadoop/.bashrc export HBASE_HOME=/opt/hbase
  export PATH=$PATH:$HBASE_HOME/bin:$HBASE_HOME/conf
  cd /home/hadoop
  . ./.bashrc

HBase

Ejemplos

Crear tabla

```
create 'cliente', 'personal', 'profesional'
```

Descripción de tabla

```
desc 'clientes'
```

Insertar datos

```
put 'cliente',1,'personal:Nombre','Paco'
put 'cliente',1,'personal:Apellido','Martin'
put 'cliente',1,'personal:Telefono','612345678'
put 'cliente',1,'profesional:Telefono','690123456'
```

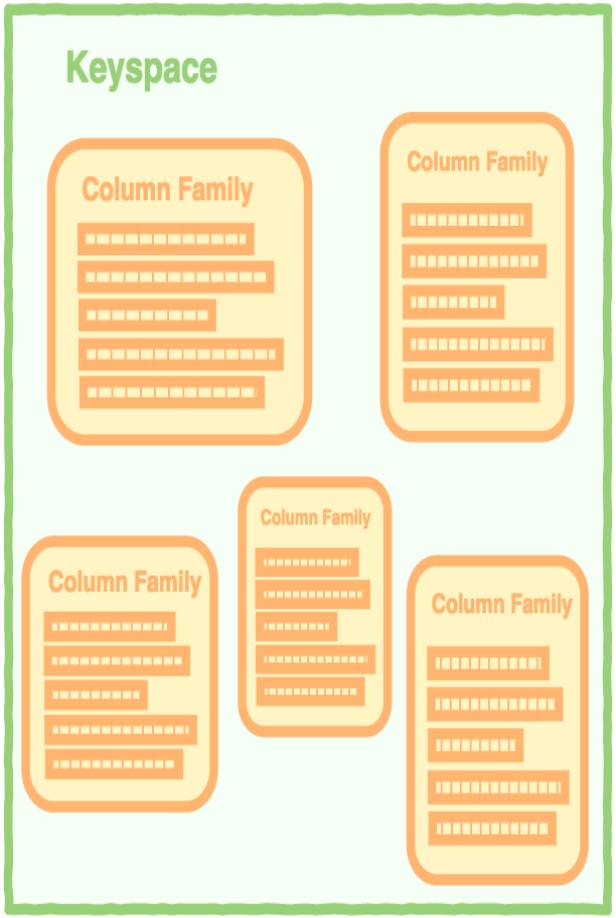
Visualizar datos

```
scan 'cliente'
```

Agenda

- Cassandra
  - Conceptos tabulares
  - Instalación
  - Ejecución
  - Principales comandos
  - Ejemplo

Keyspace


Keyspace
Column Family
Column Family
Column Family
Column Family
Column Family

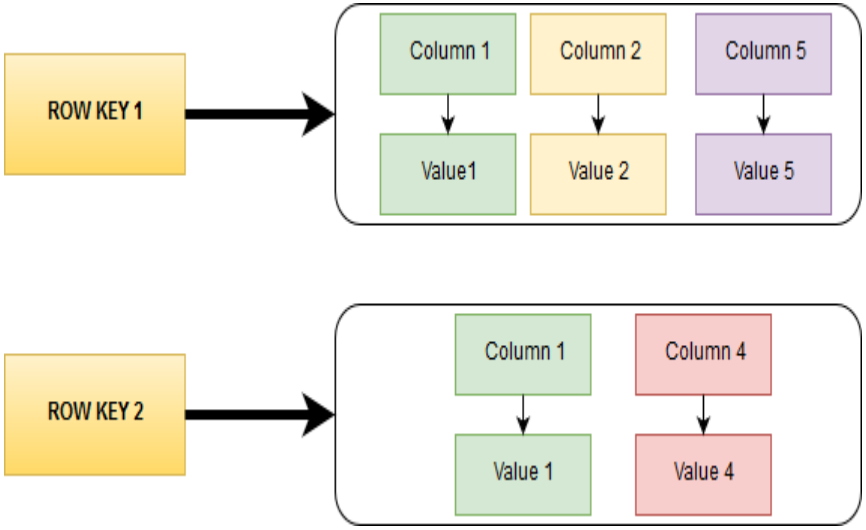
Keyspace

- Un Keyspace es el contenedor más externo para almacenar datos en este tipo de Bases de Datos. Esta estructura contiene a las estructuras llamadas Column Family(familia de columnas).
- Los atributos que definen un Keyspace son:
  - **Factor de repicacion: es el número de nodos en el clúster que van a almacenar copias de cada dato**.
  - **Replica placement strategy** : define la estrategia para distribuir las copias dentro del anillo (organización del clúster).
  - **Column families** : el keyspace es un contenedor de una lista determinada de column families.

Column Family

- Una Column Family es un contenedor de una colección ordenada de filas, cada fila es una colección ordenada de columnas.
- Cada fila cuenta con un Row Key (clave de fila) que identifica unívocamente a cada una y permite ubicar el almacenamiento de su respectiva fila.
- Cada fila tiene una colección de pares de Column Key y el valor de cada columna.
- Cada column key referencia unívocamente al valor de una columna dentro de una fila.
- La cantidad de columnas no está definida en la estructura, sino que cada fila tiene una cantidad independiente de columnas.

Column Family



Column Family

La column family está definida por los siguientes atributos:

- **keys_cached**: cantidad de row keys almacenadas en cache.
- **rows_cached:** cantidad de filas que se mantendrán en cache completes.
- **preload_row_cache:** especifica si se quiere pre-cargar a cache de filas o hacerlo a demanda

Columna

- La columna es la estructura de datos básica.
- Además del nombre de la columna y el valor, también se almacena un Instante o Timestamp con el momento en que fue insertado.

COLUMNA		
Nombre: byte[]	Valor: byte[]	Instante: clock[]

Comparación con RDBMS

RDBMS	Aproximación columnar
• **Datos estructurados.**	• Datos no estructurados.
• **Estructura fijada a priori.**	• Estructura flexible.
• **Una tabla es un arreglo de arreglos.**	• Una “table” es una lista de pares clave-valor anidados.
• **Database.**	• Keyspace.
• **Tablas.**	• Column families.
• **Relaciones con foreign keys.**	• Relaciones representadas por colecciones.

cassandra

Introducción

- Cassandra es una Base de Datos dentro del grupo NoSQL, distribuida y masivamente escalable, que implementa como estrategia de almacenamiento Collumn Family.
- Por su capacidad de escalar fácilmente resulta particularmente interesante en el ámbito de las tecnologías Big Data.
- El origen de Cassandra se remonta a Facebook, que lo diseño para potenciar las búsquedas inbox.
- Inspirado en el proyecto BigTable de Google.
- Desde 2008 facebook lo libero como proyecto opensource y actualmente se sitúa entre los proyectos top-level de la fundación Apache.

Introducción: características destacables

- **Distribuida**: la información se distribuye a lo largo de los nodos del clúster. Esto también permite alta disponibilidad, de manera que si alguno de los nodos se cae el servicio no perderá disponibilidad.
- **Escalamiento horizontal:** se puede escalar el sistema agregando nodos.
- **Requerimientos de Hardware escasos**: el hardware requerido para montar nodos es de bajo costo y fácilmente accesible.
- **Escalamiento lineal**, el rendimiento crece

TEOREMA CAP

El teorema CAP enuncia que es imposible para un sistema de cómputo distribuido garantizar simultáneamente:

- La consistencia (**C**onsistency), es decir, que todos los nodos vean la misma información al mismo tiempo.
- La disponibilidad (**A**vailability), es decir, la garantía de que cada petición a un nodo reciba una confirmación de si ha sido o no resuelta satisfactoriamente.
- La tolerancia al particionado (**P**artition Tolerance), es decir, el sistema sigue funcionado incluso si algunos nodos fallan

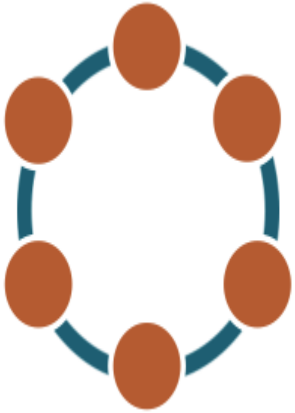
Características y arquitectura

- De acuerdo a la clasificación del teorema CAP Cassandra garantiza las últimas dos características a cambio de ser eventualmente inconsistente, aunque el nivel de consistencia puede ser configurado.

- Como ya se mencionó anteriormente, la arquitectura de Casandra está orientada

Características y arquitectura

- El clúster se dispone en forma de anillo, donde cualquier nodo puede atender las peticiones de los clientes. Para determinar el nodo de los datos, utiliza algoritmo ring.



Nodo coordinador

- A la hora de realizar una petición al clúster el cliente selecciona un nodo para actuar como coordinador de la misma, pudiendo elegir entre cualquier nodo del clúster. El nodo coordinador es quien define el factor de replicación y el nivel de consistencia requerido para la operación.

UTN.BA FACULTAD REGIONAL BUENOS AIRES
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

Centro de e-Learning

¿Alguna pregunta en el tintero?

UTN.BA FACULTAD REGIONAL BUENOS AIRES
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

Centro de e-Learning

Muchas Gracias

ignacio.urteaga@gmail.com