

PANORAMA BIGDATA, CONOCE EL PROBLEMA ANTES DE ELEGIR TU SOLUCIÓN

Javier Gómez Santos
Formador técnico



#OpenExpo2017

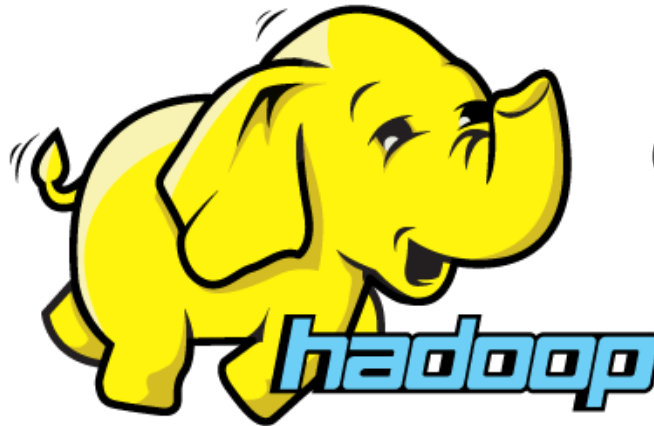


openexpo

Panorama Big Data



cassandra



mongoDB

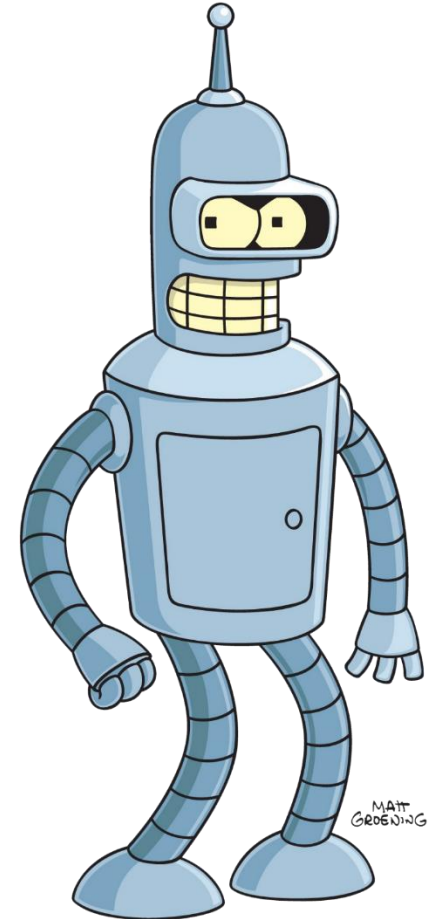


#OpenExpo2017



openexpo

Panorama Big Data



#OpenExpo2017

Hadoop, su historia

- **2002:** Doug Cutting crea Nutch
- **2003:** Google publica un White paper sobre su GFS
- **2004:** nace NDFS, precursor del HDFS
- **2004:** Google publica un White Paper con la metodología MapReduce
- **2005:** se implementa MapReduce sobre NDFS
- **2008:** Yahoo contrata a Doug Cutting para usar Hadoop a nivel web
- **Hadoop** era el nombre del peluche que tenía el hijo de Doug Cutting, un elefante amarillo



Hadoop, ¿Qué es?

- **Hadoop** es un **Data Lake**, un sistema que almacena datos tal y como son originados por una fuente.
- Estos datos, no son modificados, y serán explotados por distintos procesos para labores analíticas
- Los componentes principales de **Hadoop** son el **HDFS** y el **YARN**
- Está escrito en **Java**



Hadoop, HDFS

HDFS es un sistema de ficheros distribuido.

Almacena los ficheros partiéndolos en **bloques** de tamaño fijo, y distribuyéndolos a través de múltiples nodos.

Es muy fiable, ya que realiza múltiples copias de los datos (**réplicas**)

Tiene operaciones que se encargan de reparar los bloques defectuosos

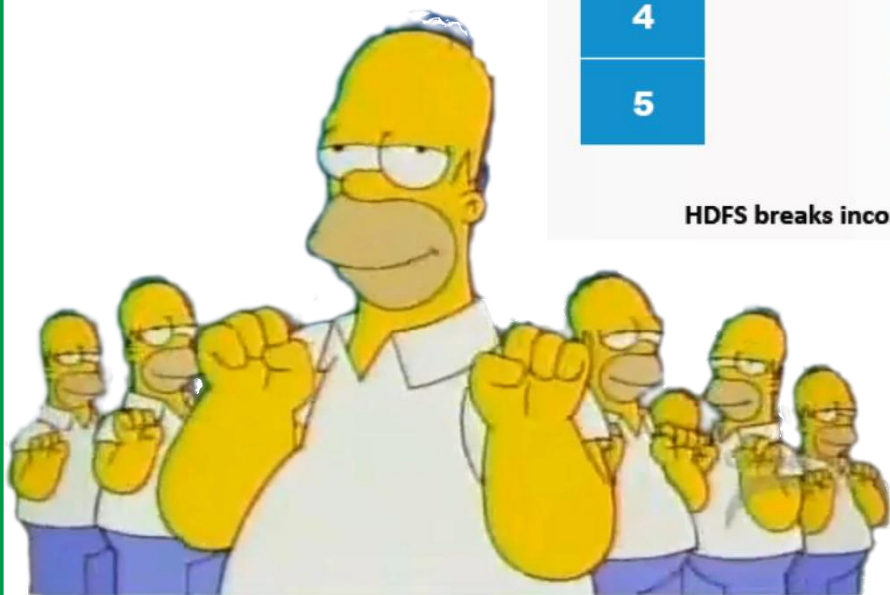


Hadoop, HDFS

Self-healing, high bandwidth **clustered storage.**



HDFS breaks incoming files into blocks and stores them redundantly across the cluster.



HDFS, arquitectura

HDFS posee tres tipos de nodos:

- **Namenode**, un único nodo que conoce los ficheros y los bloques que lo forman
- **Datanode**, cada uno de los nodos susceptibles de recibir bloques
- **Secondary Namenode**, un único nodo que realiza labores de compactación para el Namenode



Hadoop, YARN

YARN gestiona los recursos de **procesador** y **memoria** de los nodos del clúster.

Cuando un **proceso** se planifica, se **distribuye** en varios nodos, en contendedores.

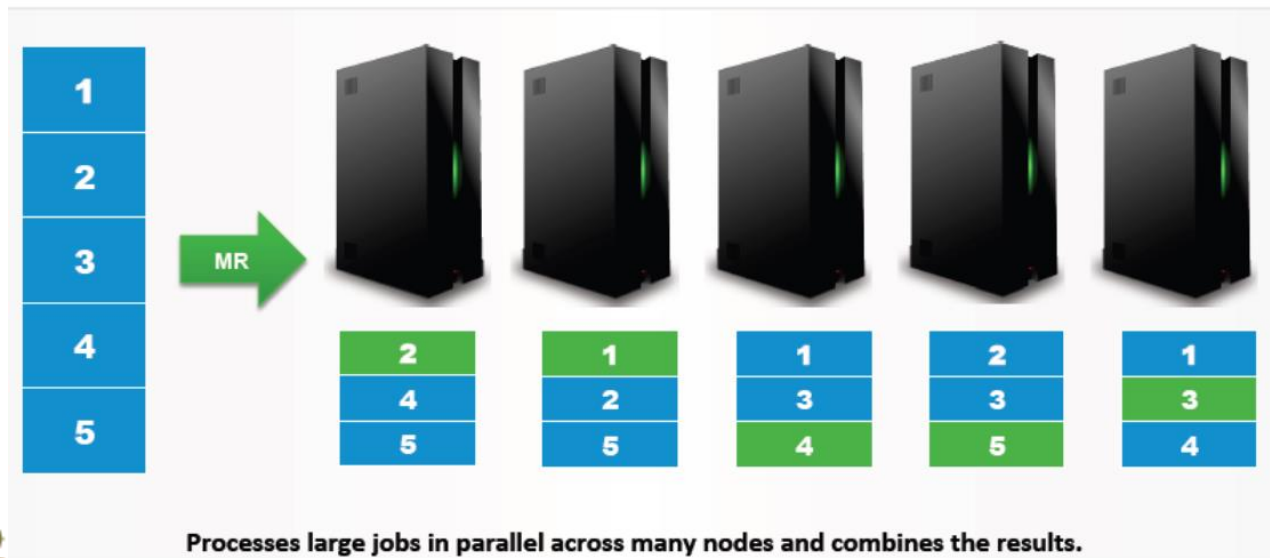
Cada **contenedor**, tiene una serie de recursos asignados.

Prioriza la cercanía, se mueve el proceso al dato y no al revés.



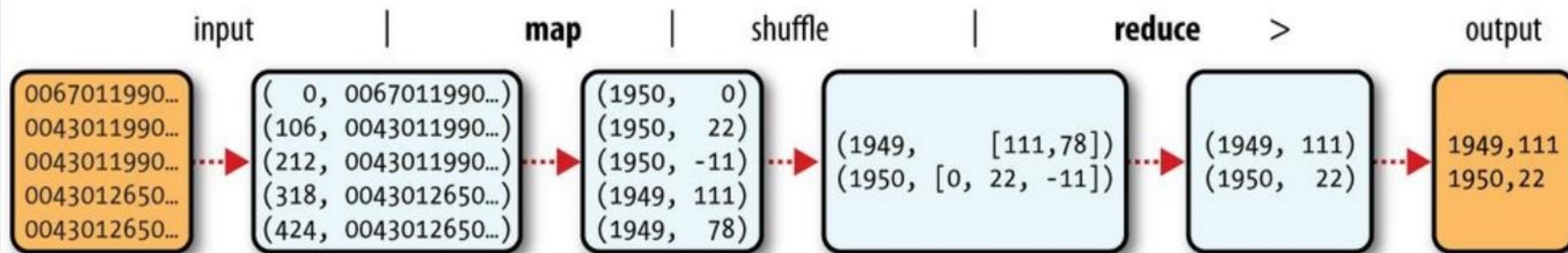
Hadoop, YARN

Distributed computing framework.

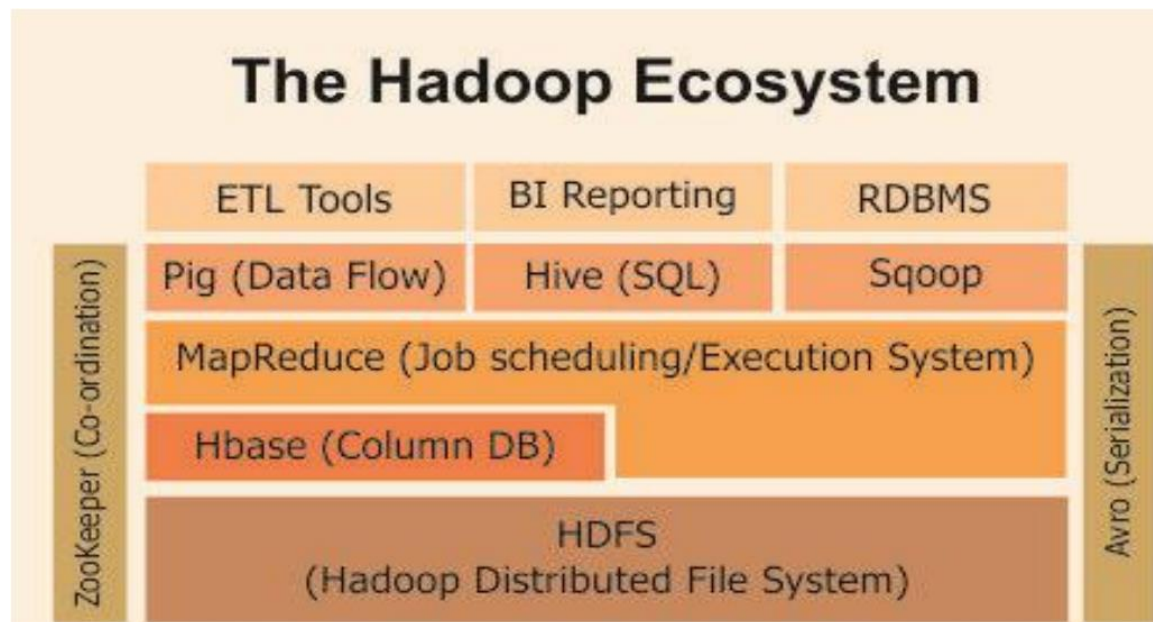


MapReduce

Es la metodología que se usa para explotar un clúster **Hadoop**. Permite distribuir la computación



Ecosistema Hadoop



Hadoop, ¿Cuándo?

Hadoop es de utilidad cuando:

- Tienes un conjunto de datos muy grande
- Datos muy diversos (desde ficheros de logs hasta xml...)
- Te ves obligado a deshacerte de información que podría ser útil
- Realizas labores analíticas sobre la información almacenada



Hadoop, ¿Cuándo no?

Hadoop NO debe usarse si:

- Buscas un sistema de consulta en tiempo real
- Vas a almacenar información sensible
- Vas a modificar la información almacenada
- Si tienes intención de sustituir un Warehouse (se complementan)



Mongo, su historia

- **2007:** 10gen comienza a desarrollar MongoDB
- **2009:** Cambian el modelo de negocio a **Open Source**, con soporte commercial
- **2013:** 10gen cambia su nombre a **MongoDB Inc.**
- **Mongo** viene de la palabra **humongous** (gigantesco)

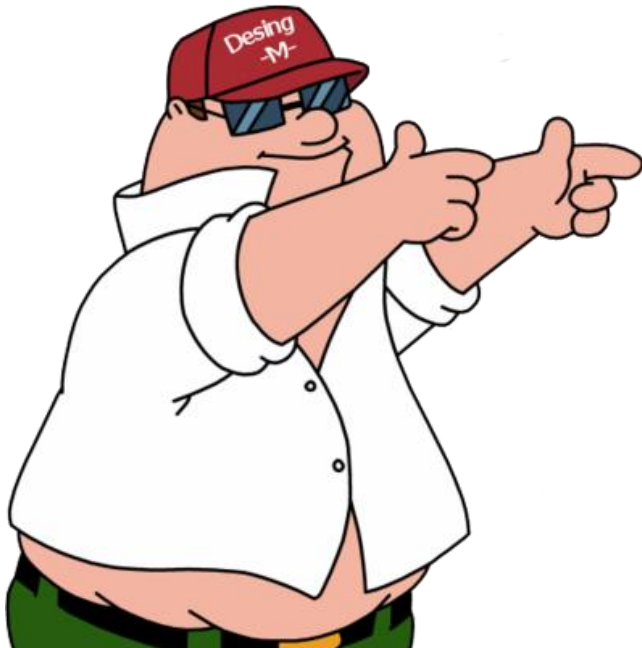


MongoDB, ¿qué es?

MongoDB es una base de datos **NoSQL**, de tipo documental, que permite un gran **escalado**

Los documentos se almacenan en colecciones, y no tienen por qué tener un patrón definido. Se define este patrón como **Esquema Dinámico**.

Permite modificar los datos insertados, crear índices y posee un framework de **agregación**



MongoDB, query language

Collection Query Criteria Modifier
`db.users.find({ age: { $gt: 18 } }).sort({age: 1 })`

{ age: 18, ...}
{ age: 28, ...}
{ age: 21, ...}
{ age: 38, ...}
{ age: 18, ...}
{ age: 38, ...}
{ age: 31, ...}

users

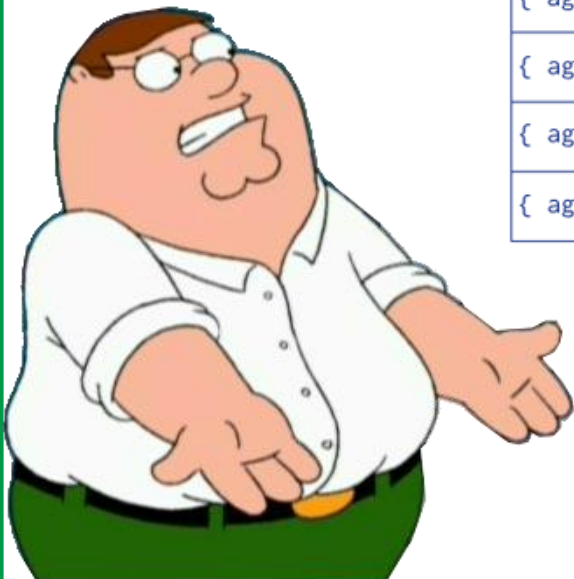
Query Criteria

{ age: 28, ...}
{ age: 21, ...}
{ age: 38, ...}
{ age: 38, ...}
{ age: 31, ...}

Modifier

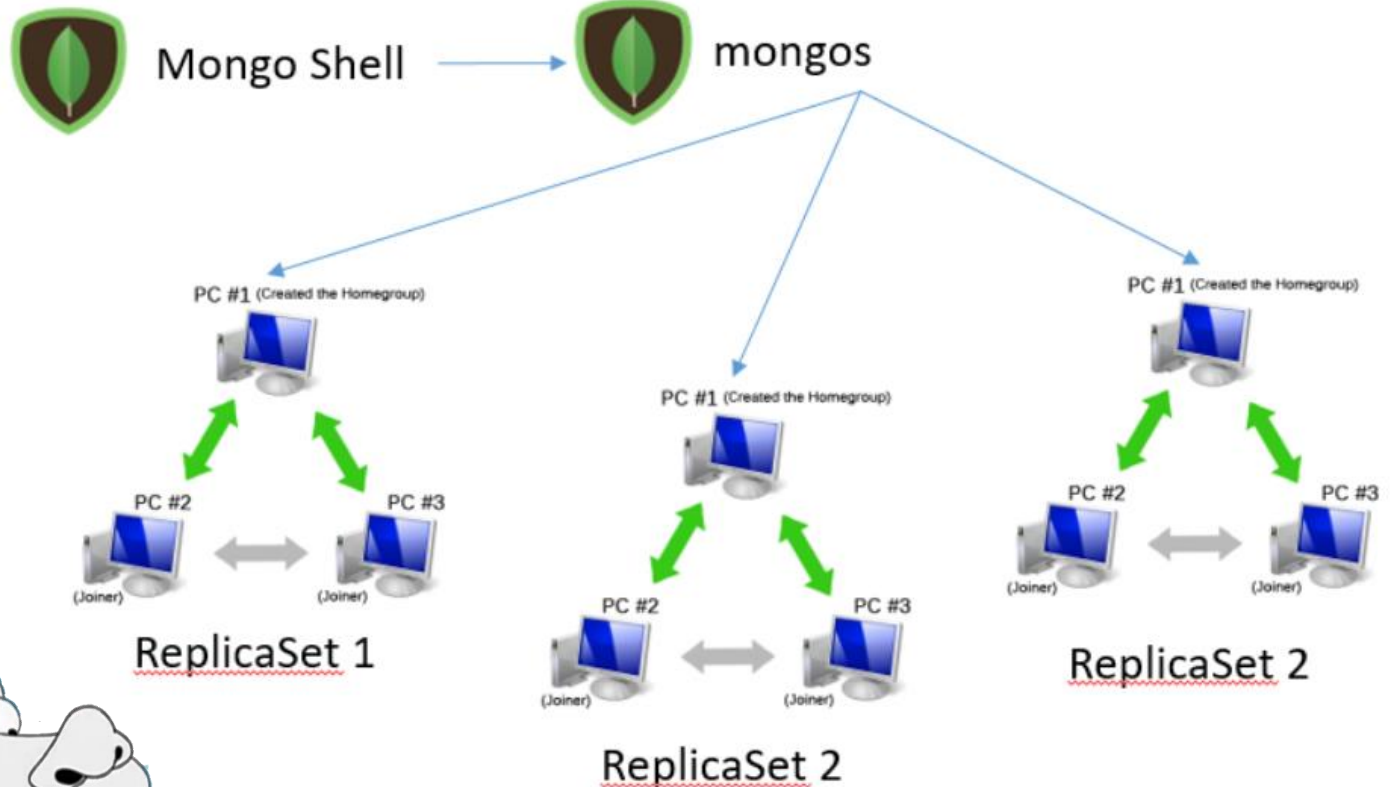
{ age: 21, ...}
{ age: 28, ...}
{ age: 31, ...}
{ age: 38, ...}
{ age: 38, ...}

Results



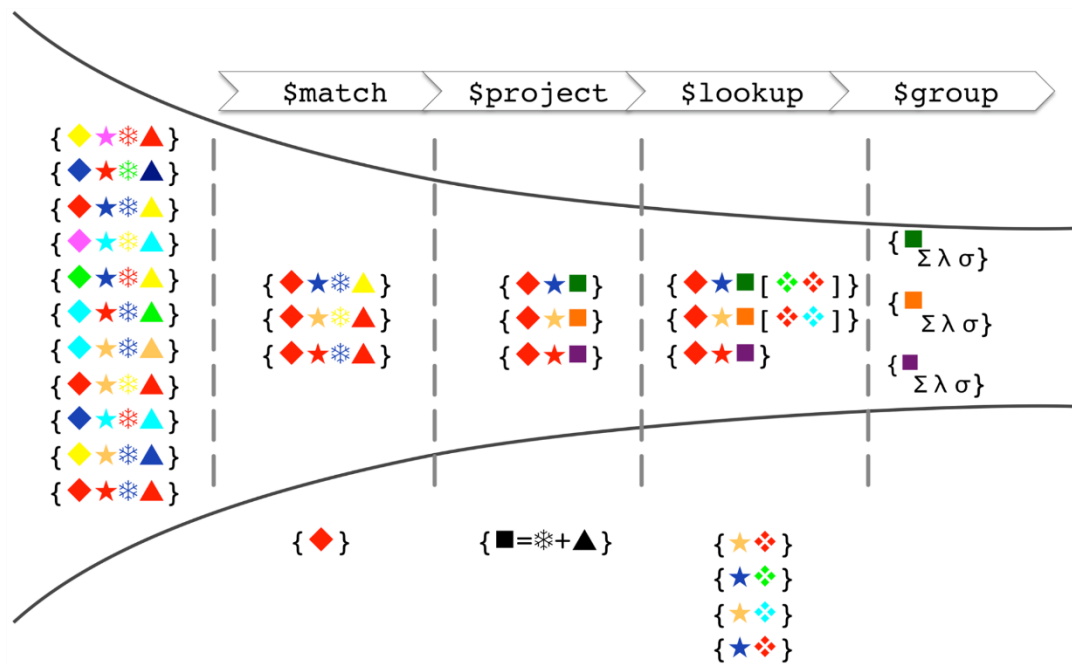
MongoDB, Arquitectura

MongoDB permite el escalado usando Shards, formados por Replica Set



MongoDB, Aggregation

MongoDB permite realizar labores analíticas, mediante su framework de agregación



MongoDB, ¿Cuándo?

MongoDB es de utilidad cuando:

- Necesitamos realizar lecturas/escrituras en tiempo real.
- Vamos a modificar la información almacenada.
- Los requisitos nos obligan a tener un esquema altamente cambiante.
- Buscamos un sistema que escale con facilidad



MongoDB, ¿Cuándo no?

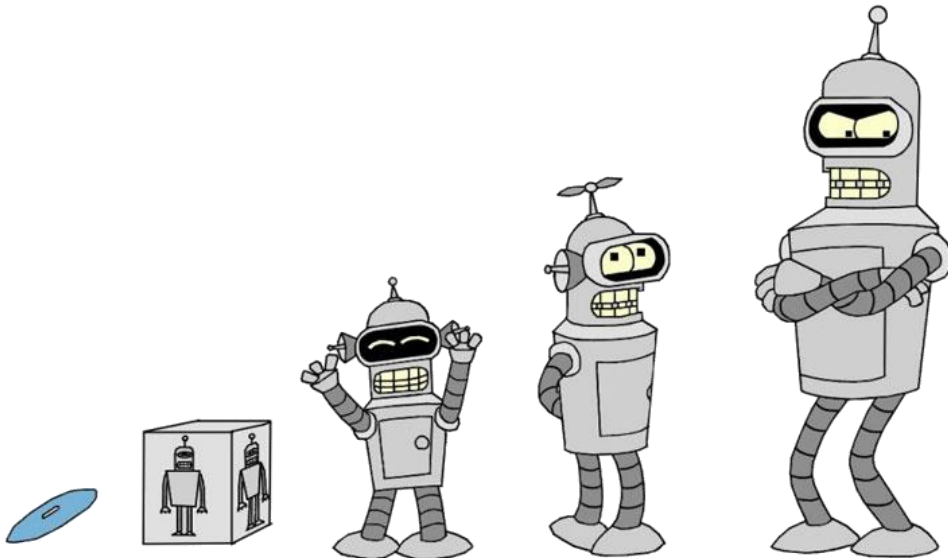
MongoDB NO debe usarse si:

- Necesitamos realizar labores analíticas pesadas
- Los datos no son transformados desde la fuente, y necesitamos guardarlos tal como se generan
- Tenemos miedo de que los desarrolladores no sigan unas reglas (MongoDB no les fuerza)



Cassandra, su historia

- **2004:** Comienza a desarrollarse **BigTable**
- **2007:** Amazon presenta **Dynamo**
- **2008:** Facebook libera **Cassandra**, como proyecto Open Source (basado en BigTable y Dynamo)
- **2011:** Se presenta **CQL**, lenguaje de acceso a datos basado en **SQL**



Cassandra, ¿qué es?

Una base de datos **NoSQL** basada en **Column Families**.

Posee un **esquema definido**, pero flexible, que permite su alteración.

Usa un modelo **totalmente distribuido**, sin nodos maestros.



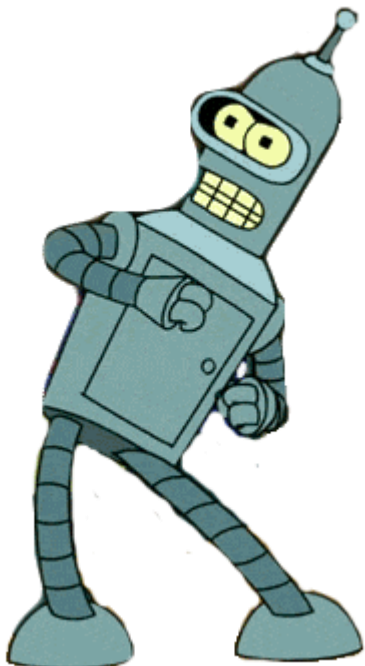
Cassandra, CQL

Un lenguaje de consultas similar a SQL

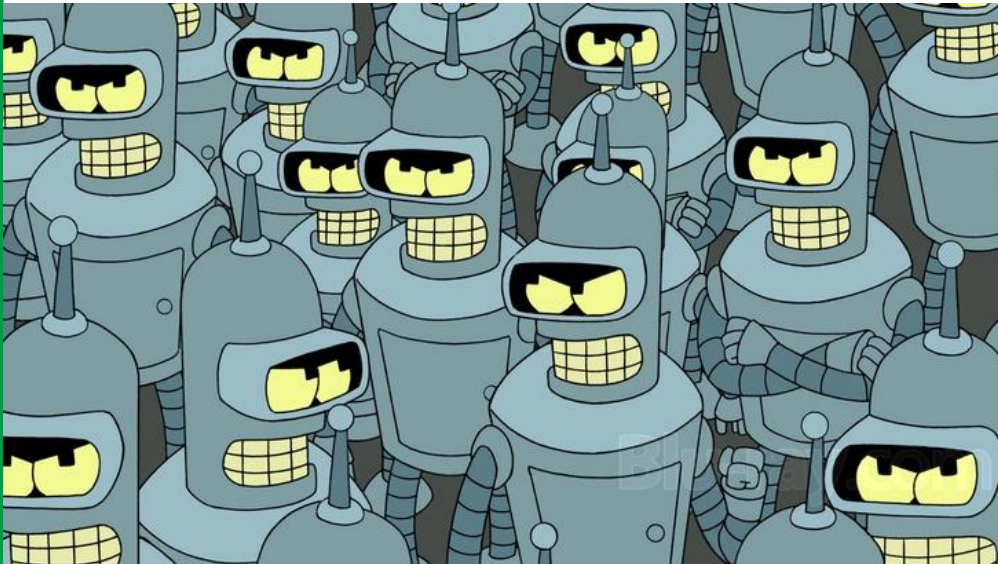
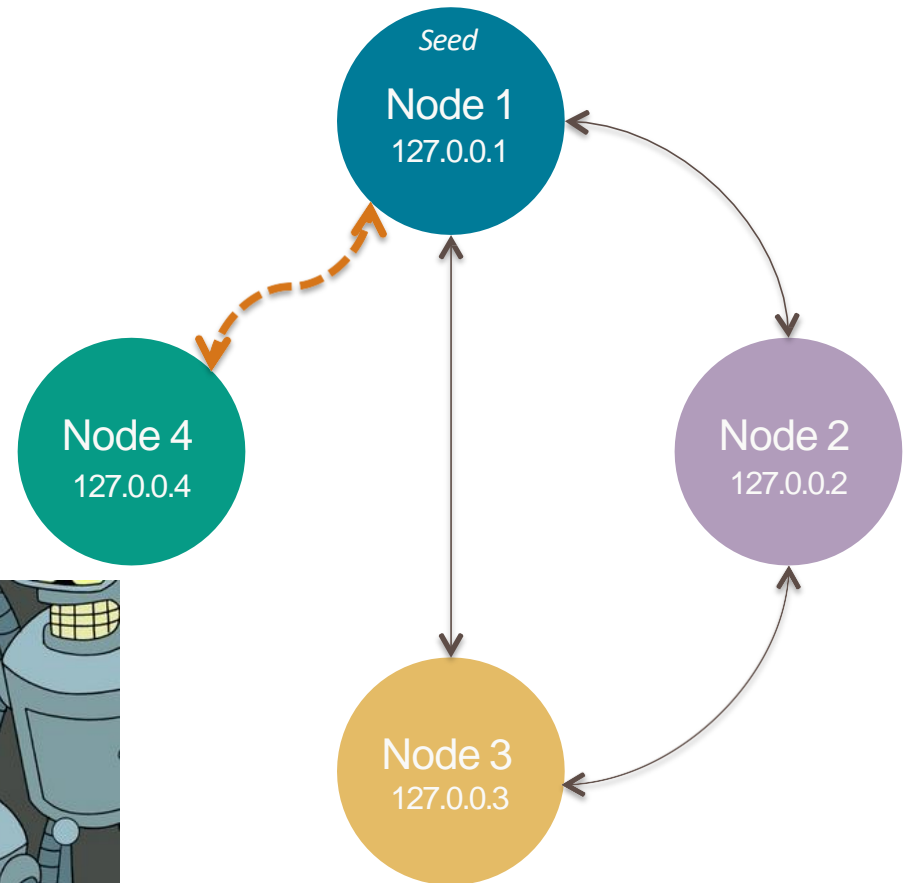
```
-- Inserts or updates  
INSERT INTO Standard1 (KEY, col0, col1)  
VALUES (key, value0, value1)
```

vs.

```
-- Inserts or updates  
UPDATE Standard1  
SET col0=value0, col1=value1 WHERE KEY=key
```

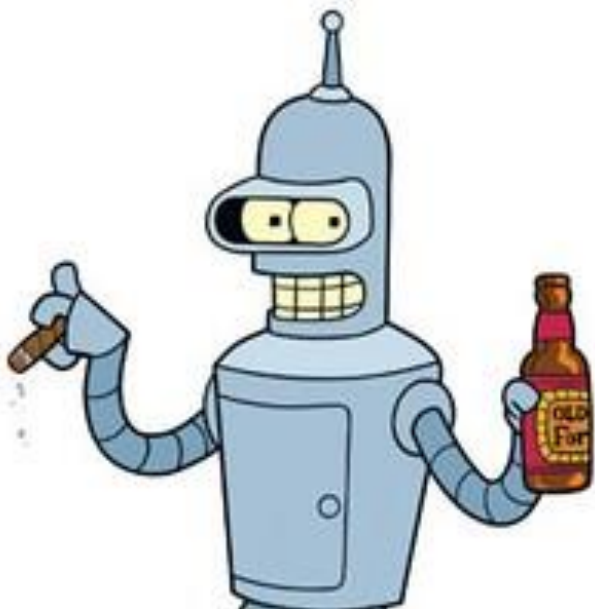
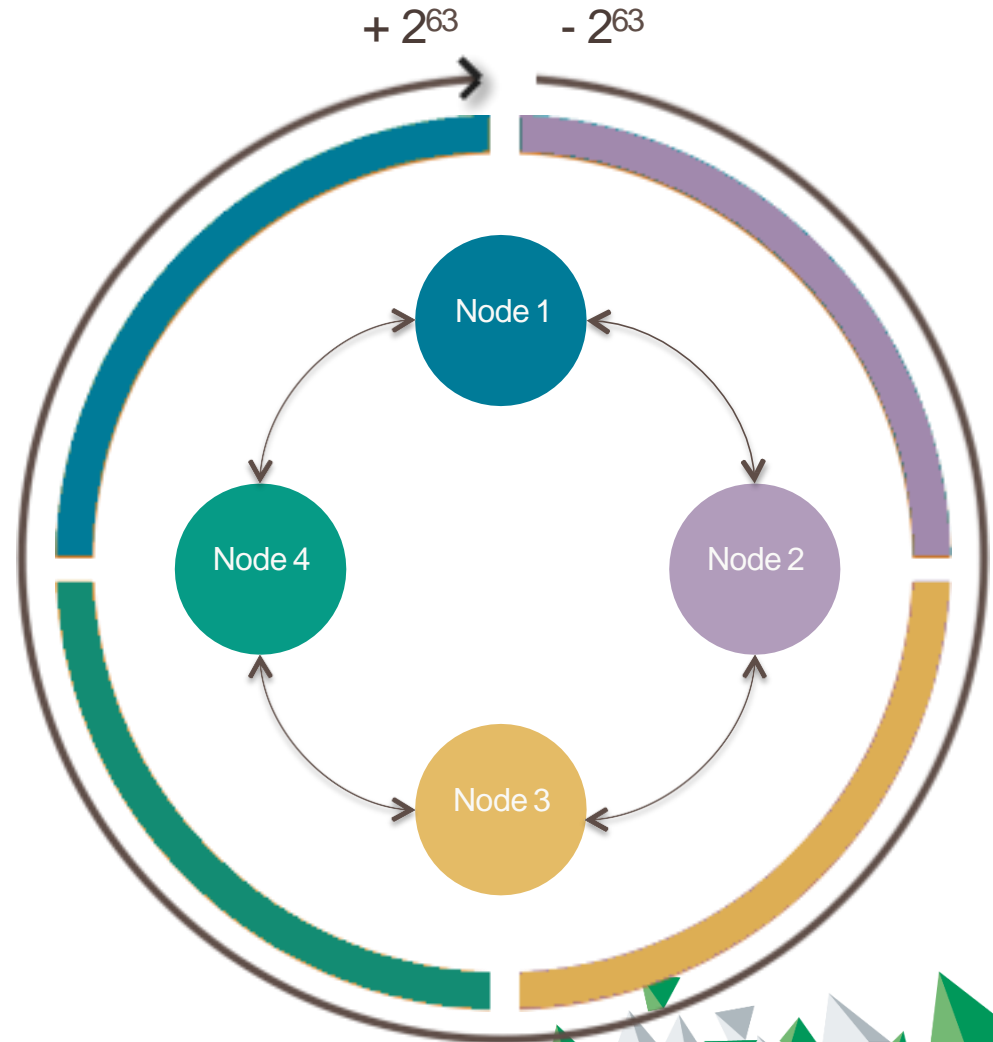


Cassandra, arquitectura



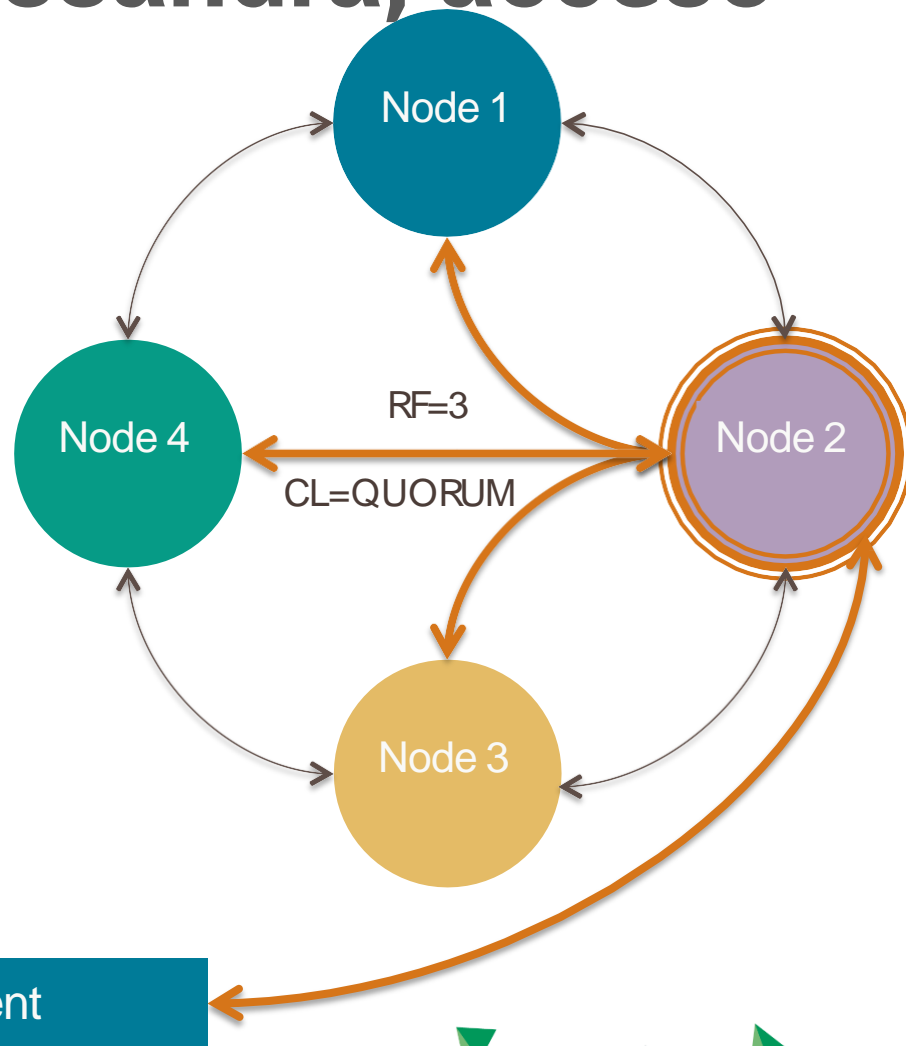
openexpo

Cassandra, distribución



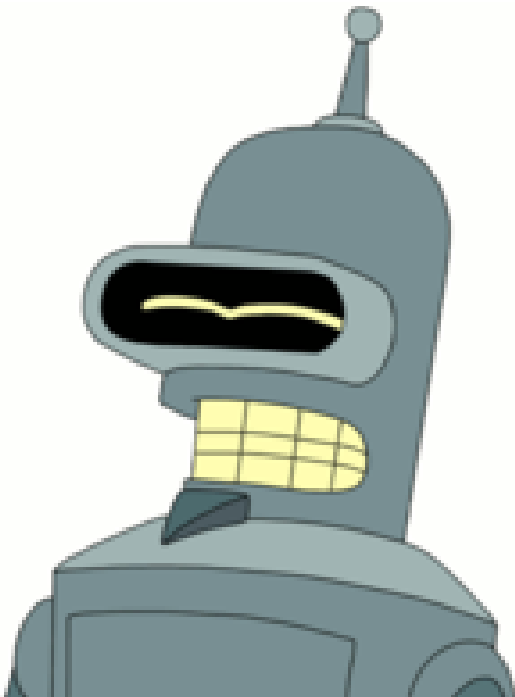
#OpenExpo2017

Cassandra, accesso



Cassandra, analíticas

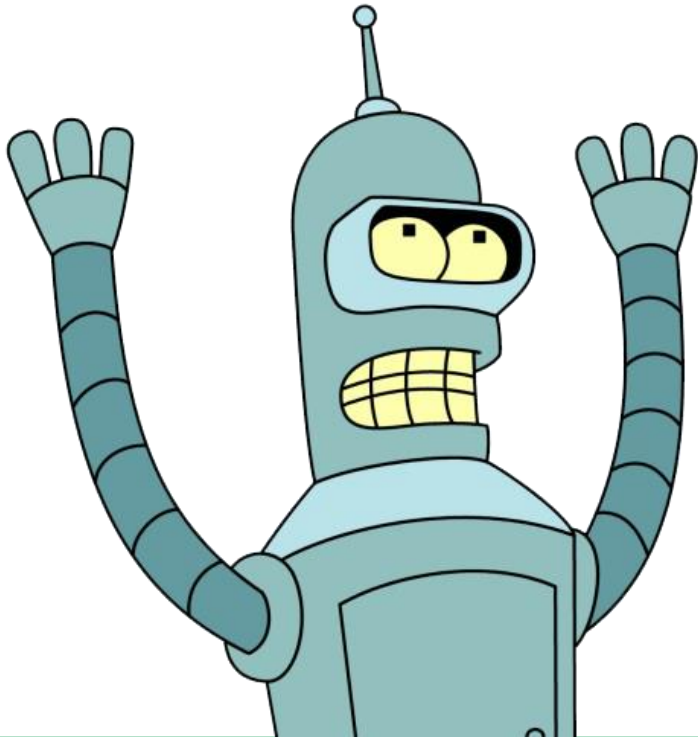
Cassandra no permite realizar operaciones analíticas directamente, pero es compatible con **Spark**, con lo que podemos delegar en él para dicha tarea



Cassandra, ¿cuándo?

Cassandra es de utilidad cuando:

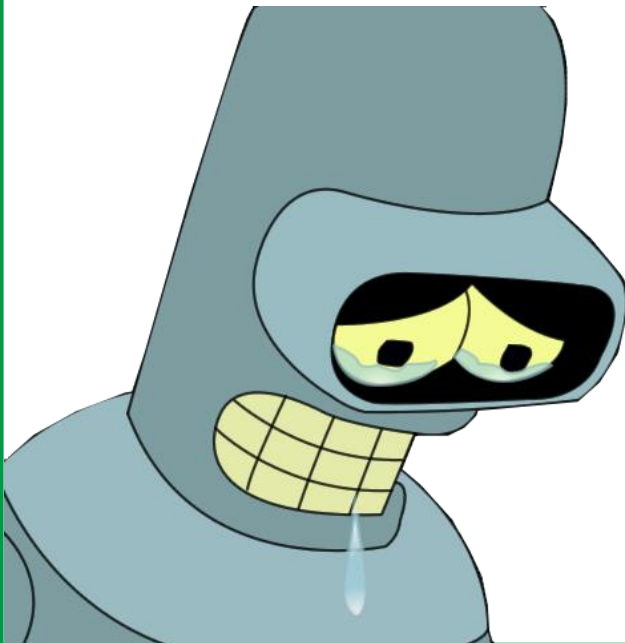
- Necesitamos un acceso en tiempo real de escritura o lectura.
- Conocemos de antemano cómo vamos a atacar a nuestro modelo de datos
- Buscamos un modelo que ofrezca el mejor rendimiento, escalado y disponibilidad



Cassandra, ¿cuándo no?

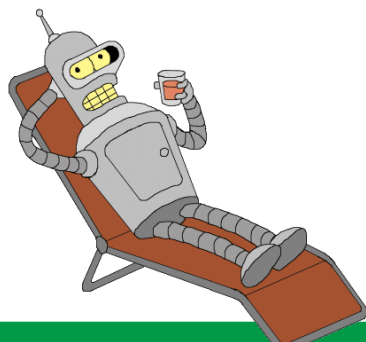
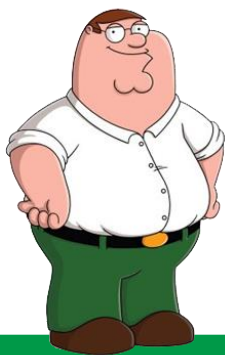
Cassandra NO debe usarse cuando:

- Cuando queramos guardar los datos tal como se generan en el origen.
- No tengamos claro cómo vamos a realizar las consultas
- Necesitemos realizar analíticas y no podamos implantar Spark
- Nuestro modelo de datos cambia de manera drástica en poco tiempo



En definitiva:

	Hadoop	MongoDB	Cassandra
Analíticas	Si	Limitadas	Si (Spark)
Acceso TR	No	Si	Si
Modificar	No	Si	Si
Modelo de datos	Puede con todo	Muy flexible	Flexible
Alta disponibilidad	Soluciones complejas	Caídas de hasta 10s	Si
Definir workflow previamente	No es necesario	No es necesario	Es obligatorio para modelar

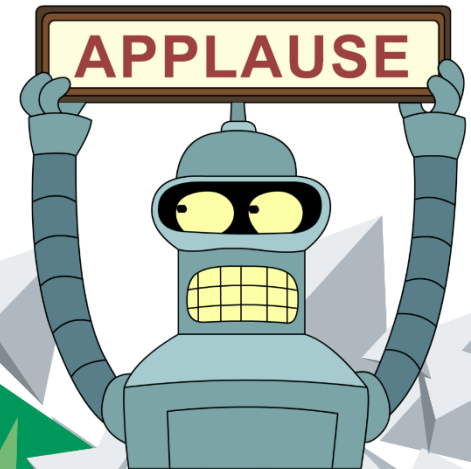




openexpo



¡MUCHAS GRACIAS!



#OpenExpo2017