

Hadoop

*Máster en Business Intelligence e
Innovación Tecnológica*

Índice de contenidos

1. Introducción.
2. Arquitectura
3. Aplicaciones
4. Hadoop v2

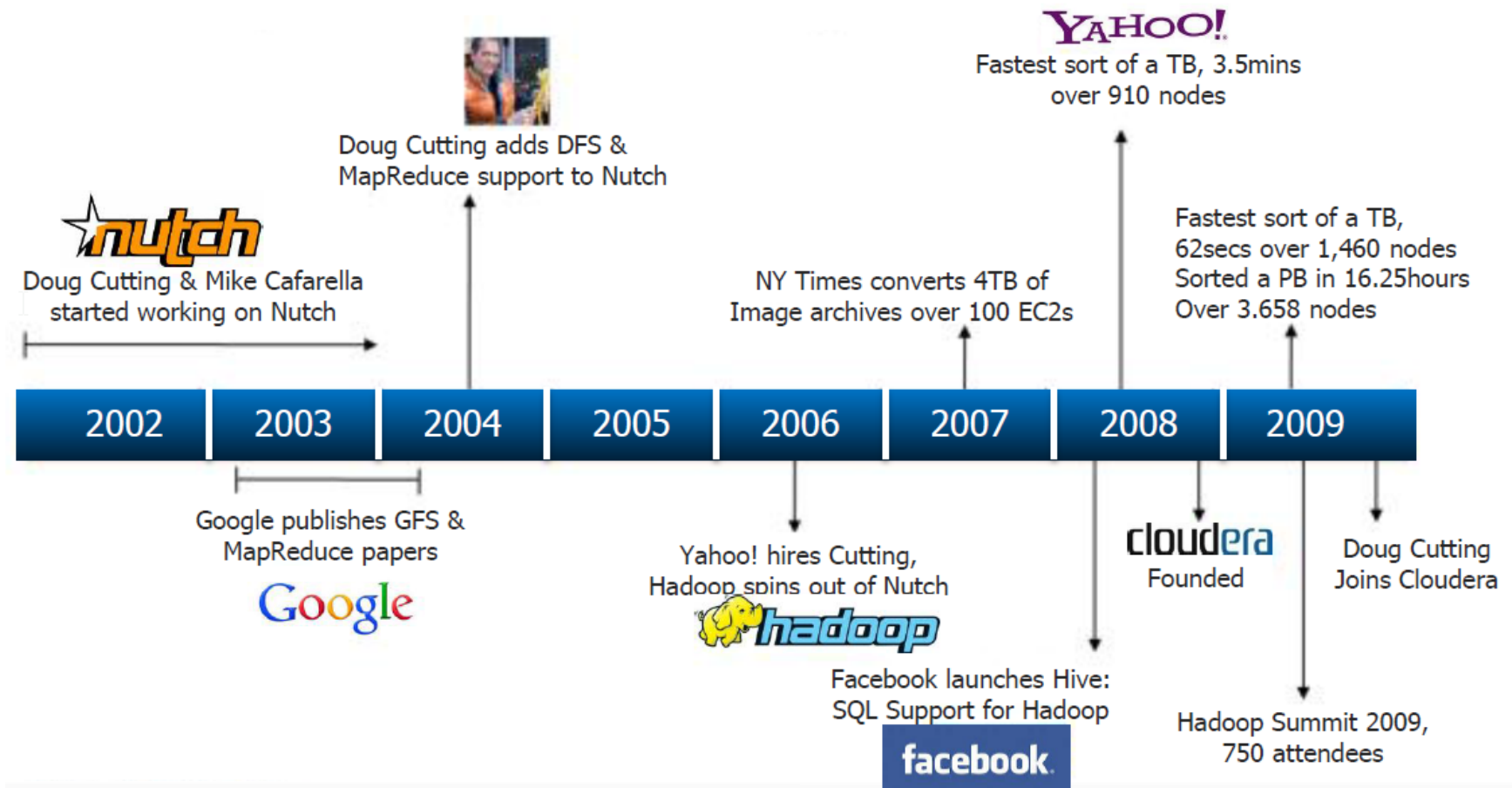
Introducción

- Hadoop es un framework para ejecución de aplicaciones en grandes clústeres integrados.
- Ofrece herramientas para el tratado de información que gestiona de forma transparente tanto para el movimiento de datos como fiabilidad.
- Implementa un paradigma computacional llamado Map Reduce dónde una petición se divide en tareas unitarias que pueden ser ejecutadas en cualquier nodo del clúster.
- Proporciona un sistema de archivos distribuido (HDFS) que almacena datos sobre los nodos de computación, proporcionando un gran ancho de banda en el clúster.
- Tanto Map Reduce como HDFS se han diseñado de forma que los errores en los nodos son tratados automáticamente por el framework.



[ver recurso](#)

Historia



Principales distribuciones

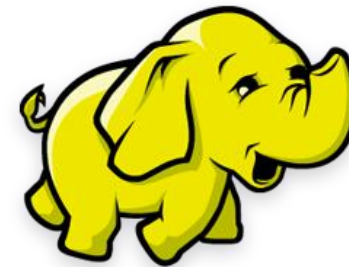


Introducción

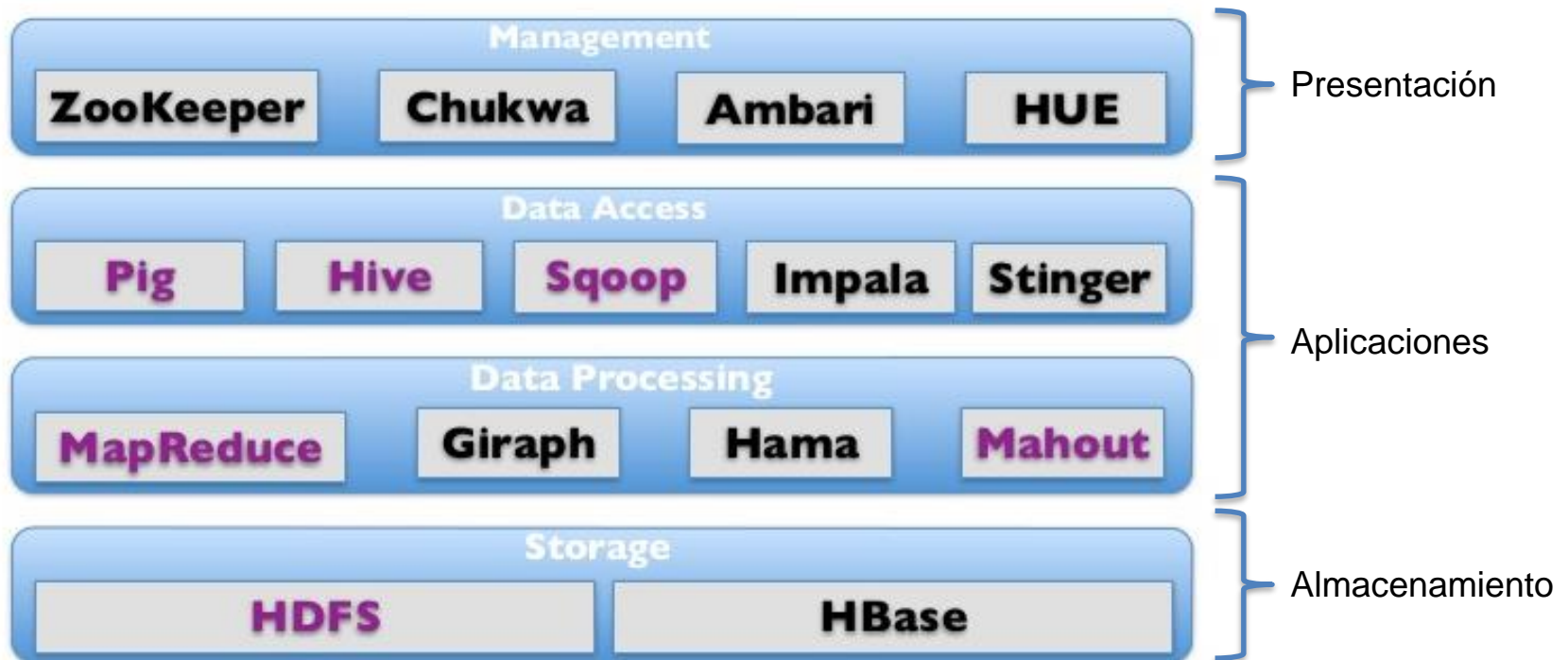
- Incluye:
 - Almacenamiento: HDFS
 - Gestor de recursos: YARN, (sólo MapReduce en Hadoop v1)
 - Modelo de programación: MapReduce
- Ventajas:
 - Bajo coste: clusters baratos o cloud
 - Facilidad de uso
 - Tolerancia a fallos
- Es bueno en:
 - Procesamiento paralelo
 - Sistema de ficheros distribuido
 - Heterogeneidad de fuentes
 - Tamaño de las fuentes
 - Dimensionamiento de las infraestructuras

Arquitectura

- Hadoop proporciona una arquitectura SMAQ (Store Map reduce And Query).
- Es un framework en desarrollo dónde todavía siguen apareciendo nuevos elementos en el stack.
- La arquitectura contempla 3 capas:
 - Almacenamiento
 - Aplicaciones
 - Presentación
- Cada fabricante desarrolla sus propias distribuciones. Éstas difieren sobretodo en capa de presentación y almacenamiento.
- Modos de funcionamiento:
 - Standalone: todo en un nodo, para pruebas
 - Pseudodistribuido: funciona como una instalación completa, pero en un solo nodo
 - Totalmente distribuido



Arquitectura

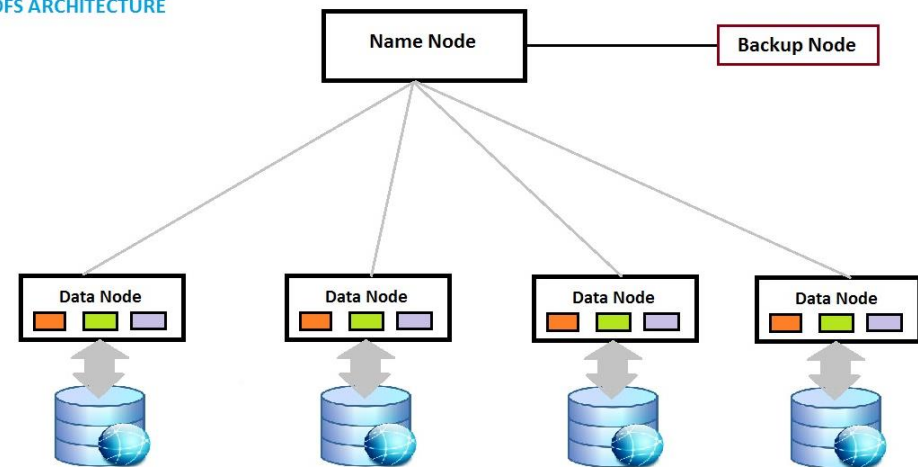


HDFS

- Hadoop Distributed File System, es una evolución de GFS (Google File System)
- Es la parte física de la arquitectura sobre la que se construyen las aplicaciones Hadoop.
- Es un sistema de ficheros de escritura única y lectura múltiple ejecutado en clúster.
- Permite gestionar la distribución de peticiones de información de forma muy rápida entre nodos.

- Namenode: Mantiene la información (metadatos) de los ficheros y bloques que residen en el HDFS
- Datanodes: Mantienen los datos
- Bloques Por defecto 128 MB (64 MB en v1) bloques pequeños aumentan el paralelismo (un bloque por Map) bloques más grandes ocultan la latencia de búsqueda y reducen la carga del NameNode

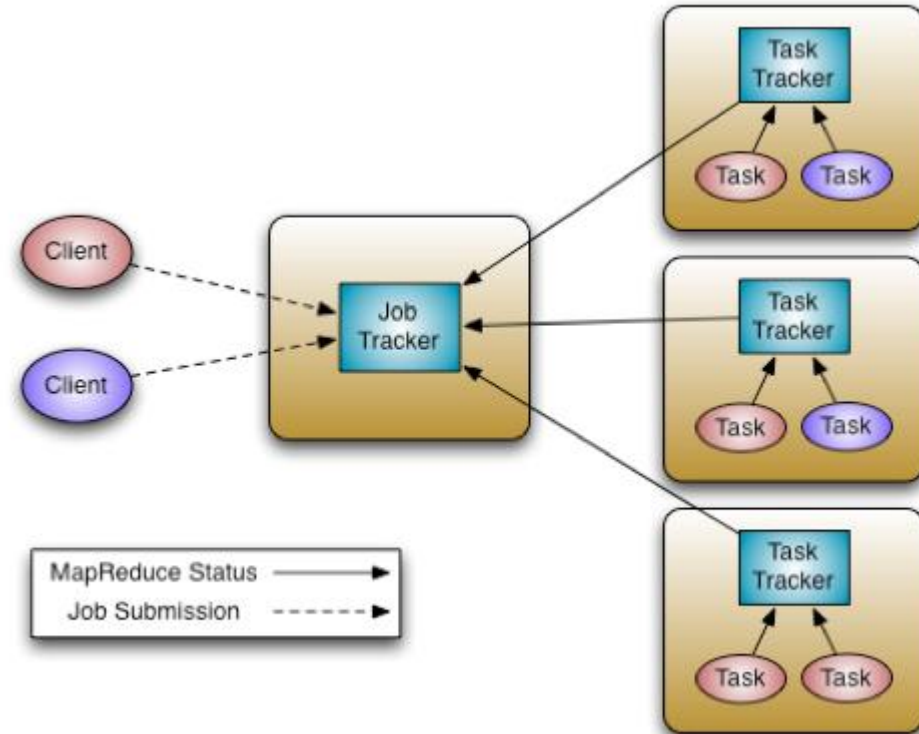
HDFS ARCHITECTURE



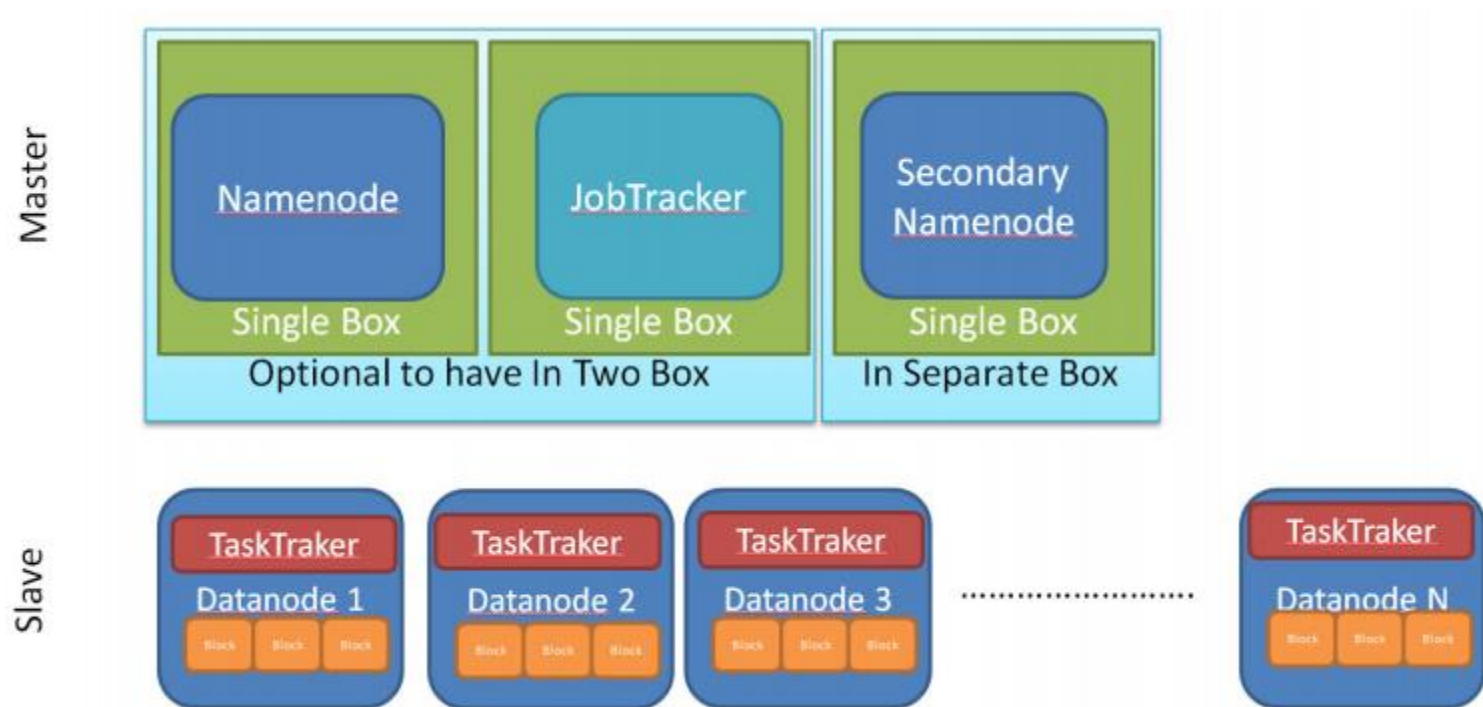
Otros nodos

- Partiendo de la ejecución de Jobs MapReduce, se introducen nuevos tipos de nodos:

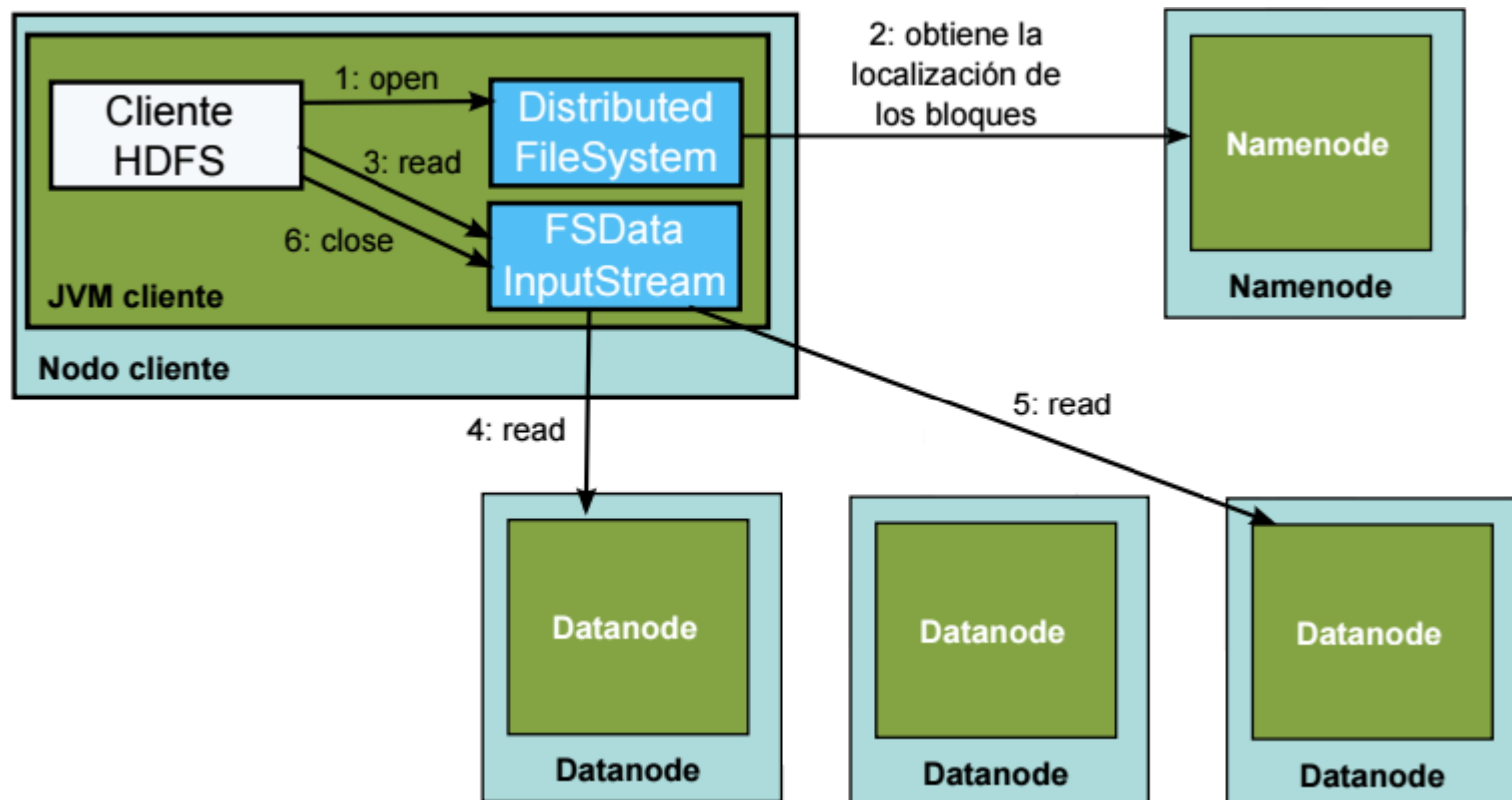
- Jobtracker:
 - Coordina todos los trabajos.
 - Envía tareas a los tasktrackers.
 - Mantiene información sobre el progreso de cada tarea.
- Tasktrackers:
 - Realizan las tareas.
 - Envían informes de progreso al Jobtracker.
- Splits
 - Trozos en los que se divide la entrada a una tarea MapReduce.
 - Normalmente del tamaño de 1 bloque (64 MB).



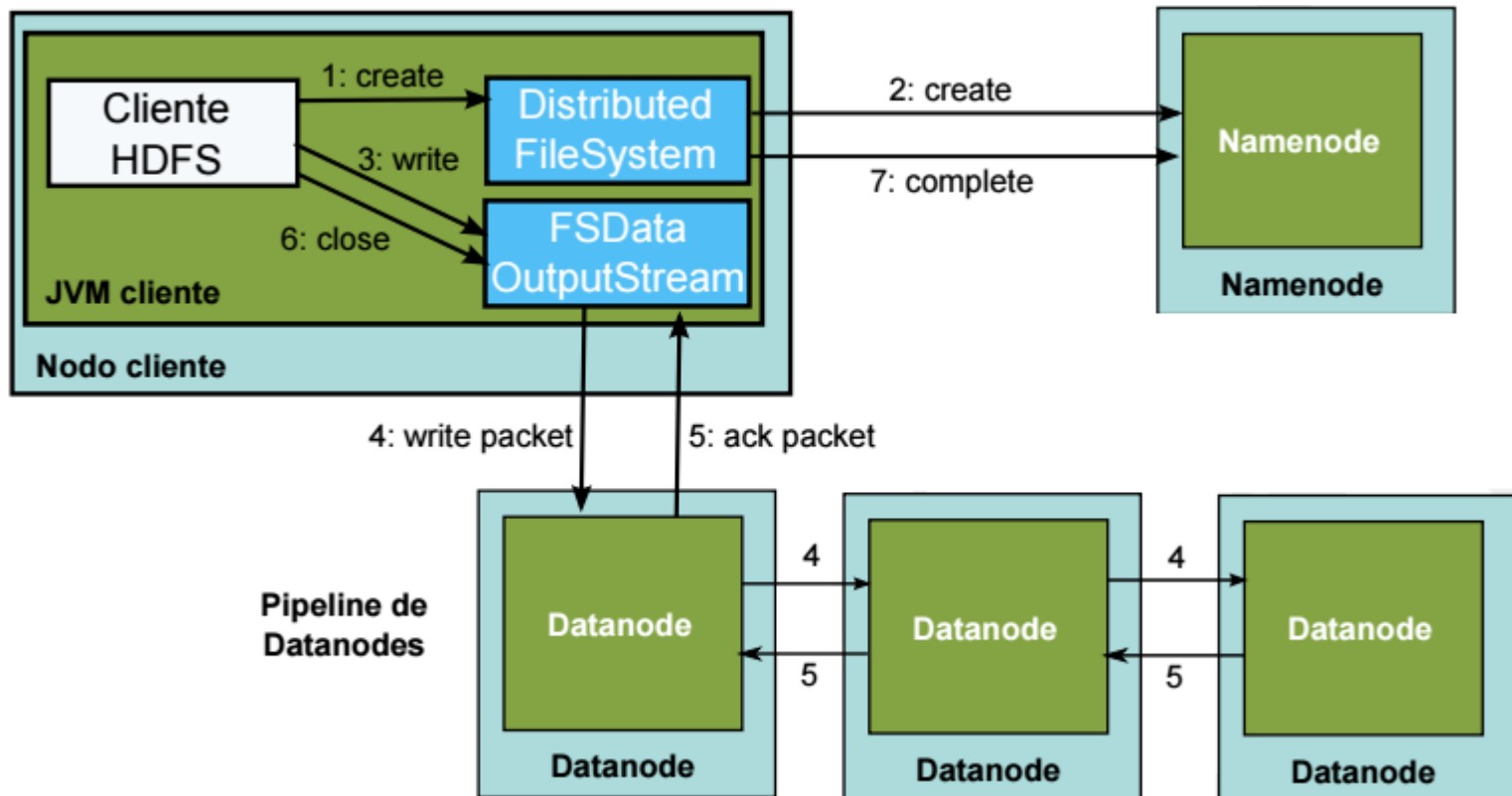
Arquitectura



Lectura en HDFS

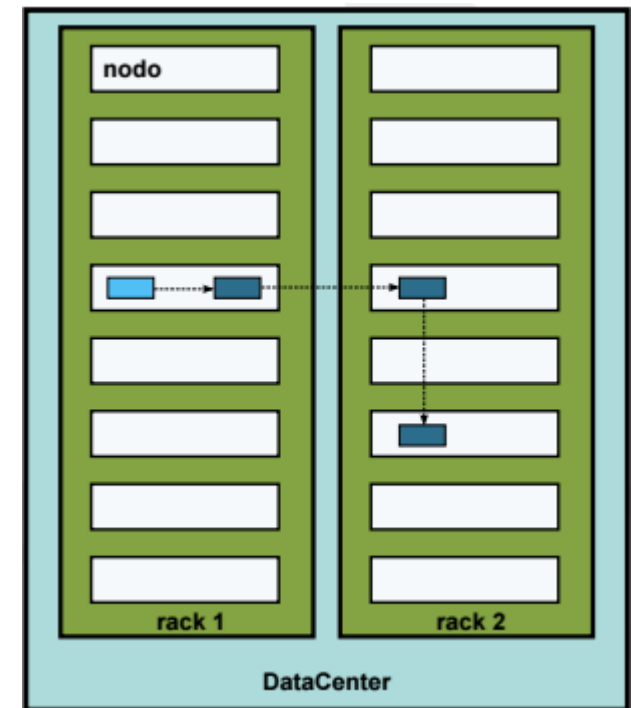


Escritura en HDFS



Replicación

- HDFS realiza un mínimo de tres réplicas de cada elemento para poder asegurar su tolerancia a fallos.
- Política por defecto:
 - 1ª réplica: en el nodo del cliente o en un nodo al azar
 - 2ª réplica: en un rack diferente de la primera (elegido al azar)
 - 3ª réplica: en el mismo rack que la 2ª, pero en otro nodo
 - Otras réplicas: al azar (se intenta evitar colocar demasiadas réplicas en el mismo rack)



HBASE

- HDSF es un sistema de ficheros orientado al añadido, HBASE nace de la necesidad de añadir accesos de lectura/escritura sobre los datos.
- Es una Base de Datos NoSQL distribuida (manteniendo coherencia con su sistema de ficheros, HDSF) y orientada a columnas.
- Su arquitectura está basada en BigTable.
- Características:
 - Consistencia de datos contra accesibilidad.
 - Se integra en cualquier arquitectura basada en Hadoop, con alta eficiencia en cargas masivas y MapReduce.
 - Particiones de Rango ordenadas, sin algoritmos hash.
 - Fragmenta y escala la información de forma ordenada.
- Está modelado para el uso de un nodo maestro HBase que gestiona un grupo de otros nodos esclavos (*regionserver*).

HBASE

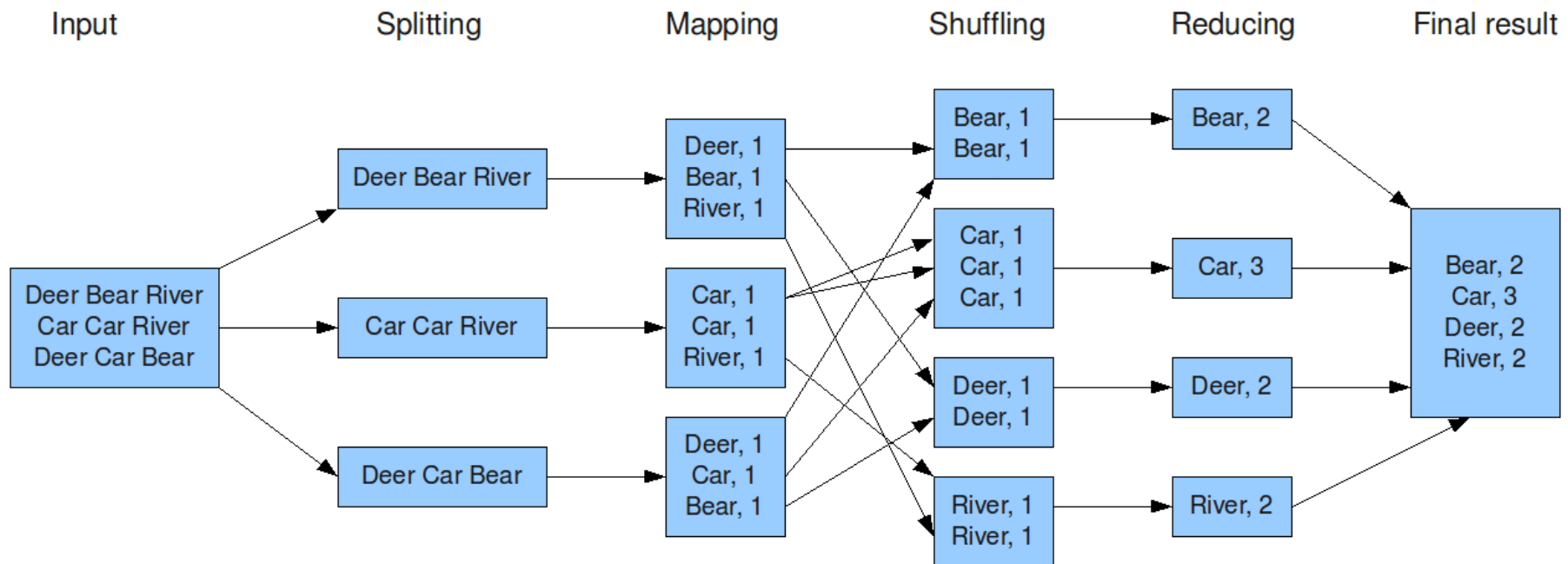


MapReduce

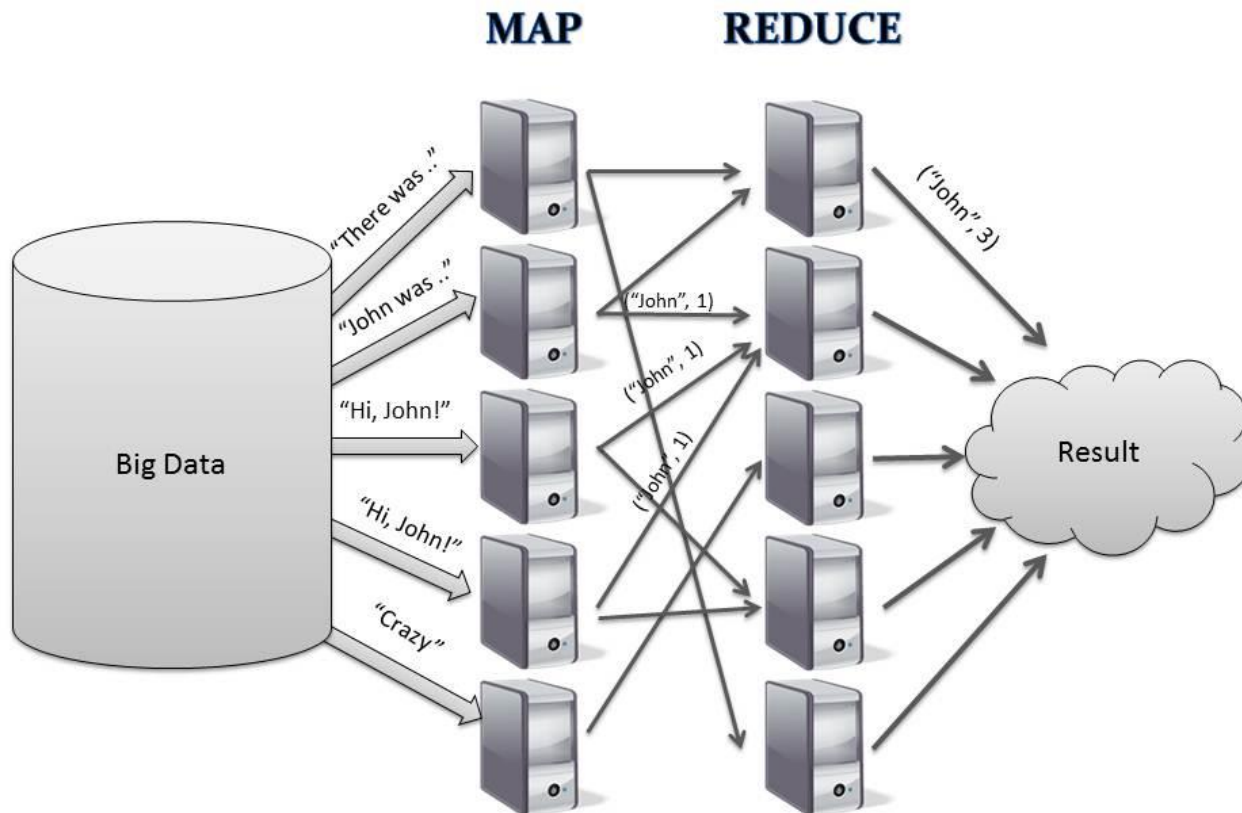
- Representa una capa de cómputo para la información almacenada distribuida.
- Realiza dos funciones:
 - **Mapeo** (Split and Map), divide las consultas realizadas en distintas partes y las procesa a nivel de nodo.
 - **Reducir** (Shuffle and Reduce), agrega los resultados de la función Map y determina el resultado de la consulta.
- Gracias a dos tipos de nodos:
 - **Jobtrackers**: coordinan los jobs del sistema programando tareas para ejecutarse en los *tasktracers*.
 - **Tasktracers**: ejecutan las tareas definidas por el *jobtracker* a quién devuelven la información generada, lo que le permite mantener un registro de la evolución del proceso. Si una tarea falla, el *jobtracker* puede reprogramar la tarea ejecutada en el *tasktracer*.
- Gracias a este punto, se consigue un buen nivel de tolerancia a fallos, pudiendo gestionar las excepciones automáticamente para rehacer los cálculos efectuados sobre la petición del cliente.

MapReduce

The overall MapReduce word count process



MapReduce



Map(): separa los registros en nodos que los procesan de forma separada.

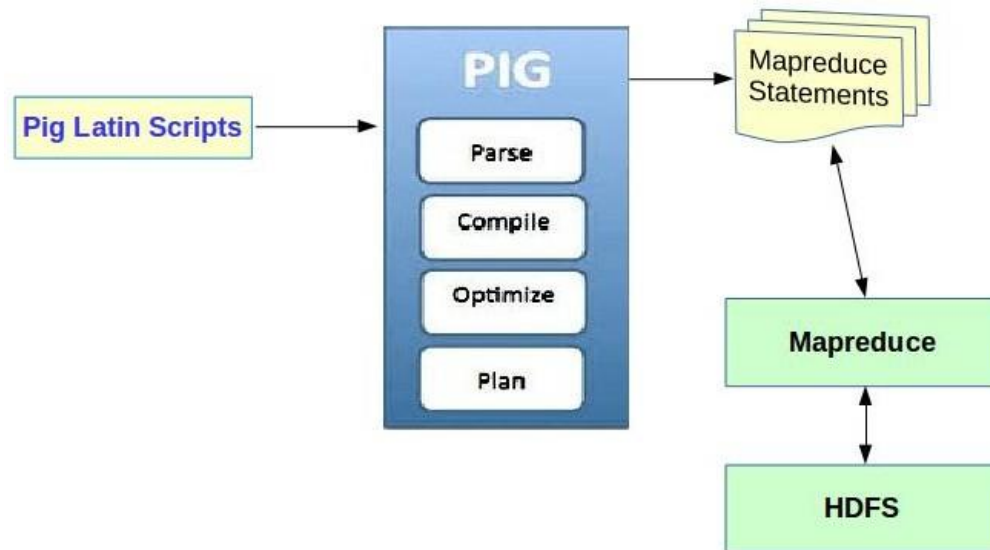
Reduce(): agrupa los valores con la misma clave

MapReduce



Pig

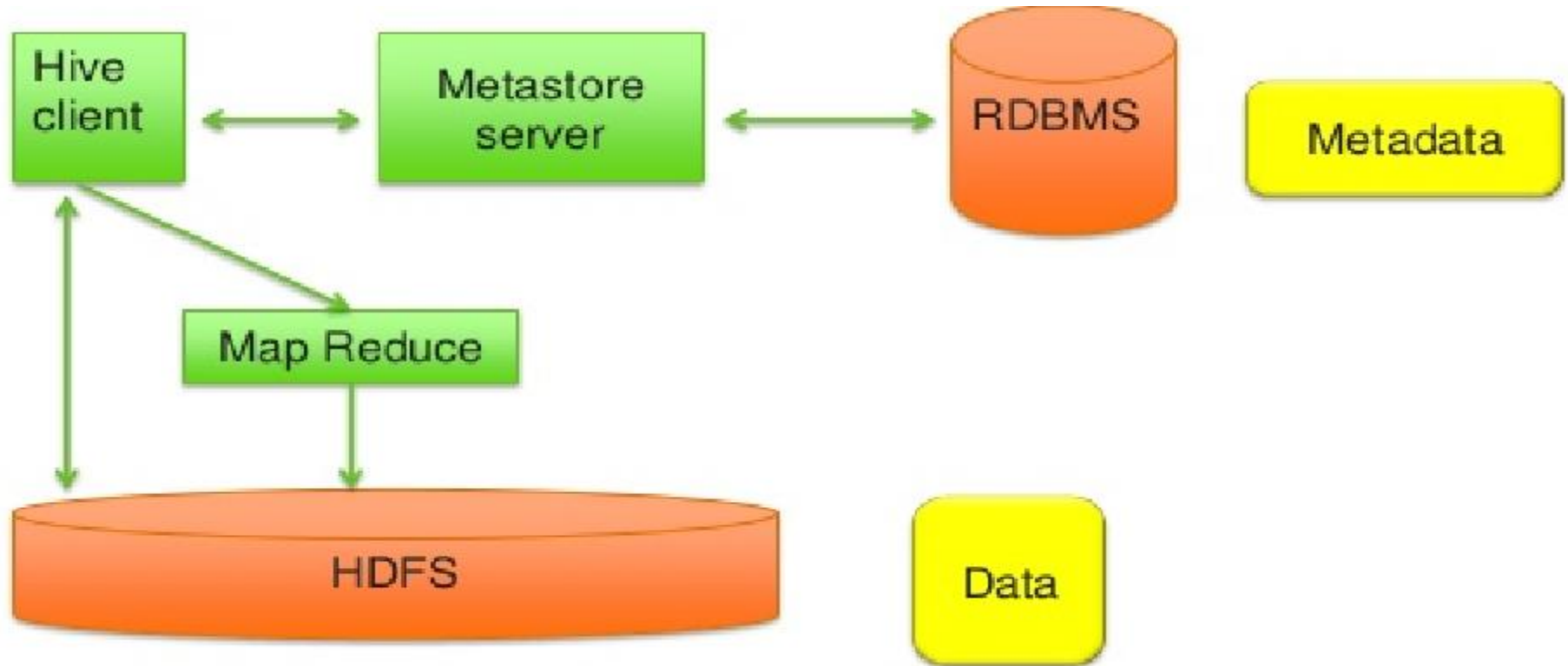
- Es un lenguaje de programación de alto nivel para generar programas batch MapReduce.
- Su nombre viene de Pig Latin, un juego de lengua inglesa.
- Muy similar a SQL.
- Permite la creación de funciones que pueden implementarse con lenguajes externos a Hadoop (Java, Python, Javascript) y utilizarlas desde el mismo.
- Acaba ejecutando tareas MapReduce.



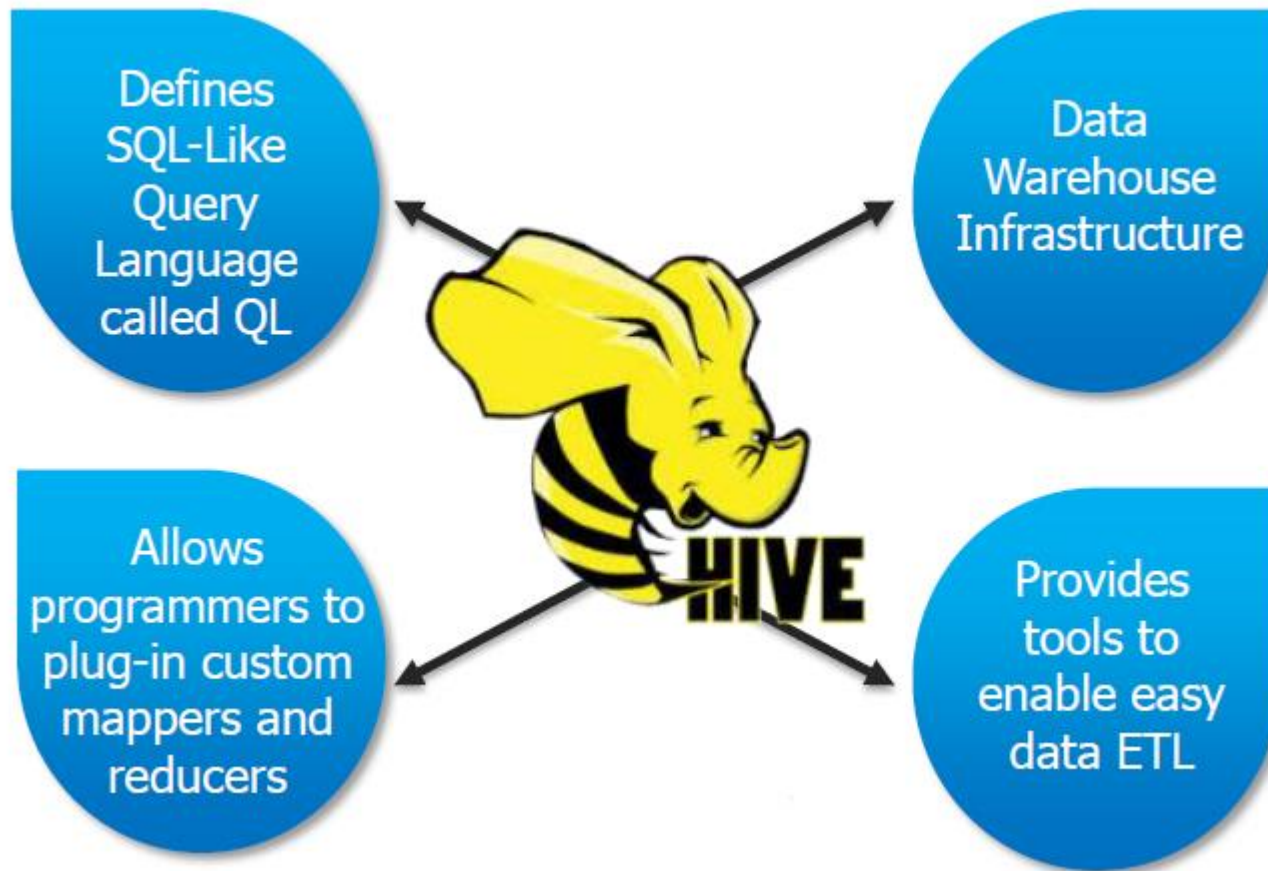
Hive

- Desarrollado para la explotación de la información de una empresa llamada Facebook.
- Permite tratar grandes cantidades de información en formato analítico al igual que se haría en un DWH.
- Realiza consultas mediante el lenguaje HiveQL que se convierte a MapReduce.
- Gracias a ello, desarrolladores sin conocimiento de MapReduce pueden explotar los datos del DWH.
- Permite una integración fácil con herramientas de BI y visualización de datos como *Microstrategy, Tableau,...*

Hive



Hive



Pig & Hive

Pig Latin

```
countries = load '/user/gharriso/PIG_COUNTRIES' AS
(country_id, country_name , country_subregion , region);

customers= load '/user/gharriso/PIG_CUSTOMERS' AS
(cust_id,first_name, last_name, gender, yob, marital, postcode,city,country_id);

asianCountryys = filter countries by region matches 'Asia';

joined = join customers by country_id, asianCountryys by country_id;

grouped = group joined by country_name;

aggeg = foreach grouped generate group, COUNT(joined.customers::cust_id);

morethan500cust = filter aggeg by $1 > 500;

ordered =order morethan500cust by $1 desc;

dump ordered;
```

SQL or Hive QL

```
SELECT country_name,COUNT(cust_id) AS cust_count
```

```
FROM countries co
```

```
JOIN customers cu
ON (co.country_id=cu.country_id)
```

```
WHERE country_region='Asia'
```

```
GROUP BY country_name
```

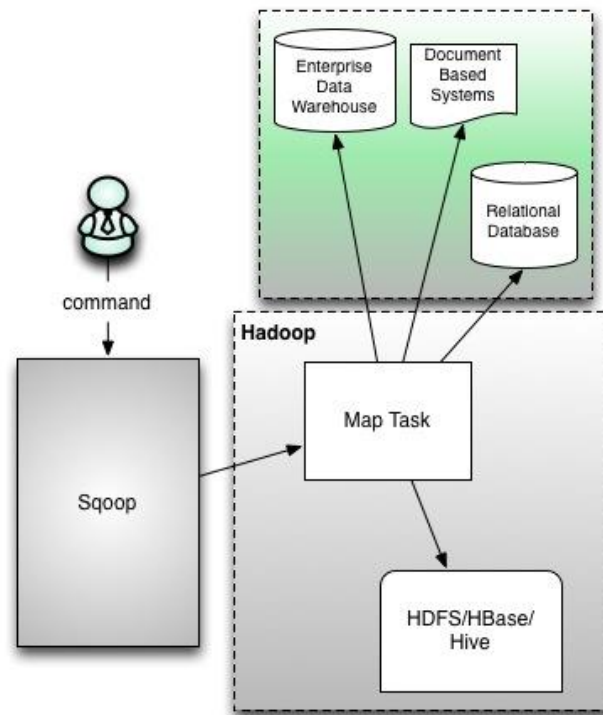
```
HAVING COUNT(cust_id)>500
```

```
ORDER BY cust_count DESC
```

<http://guyharrison.net>

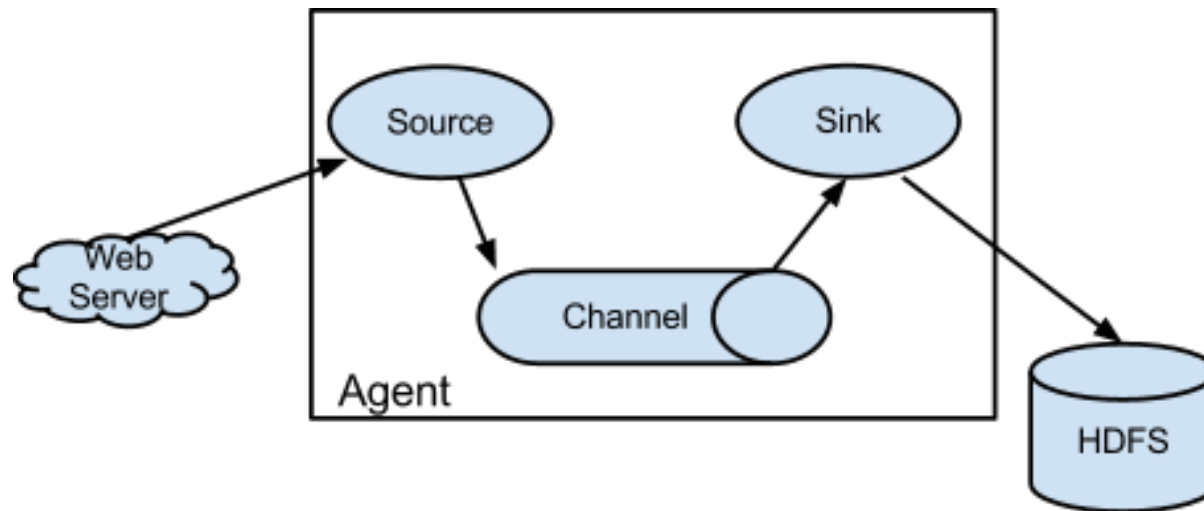
Sqoop

- Permite importar y exportar datos de una forma estructurada desde SGBDR, DWH y sistemas NoSQL.
- Los datasets importados se dividen en particiones y una tarea MapReduce de tipo Map los incorpora en los nodos correspondientes.



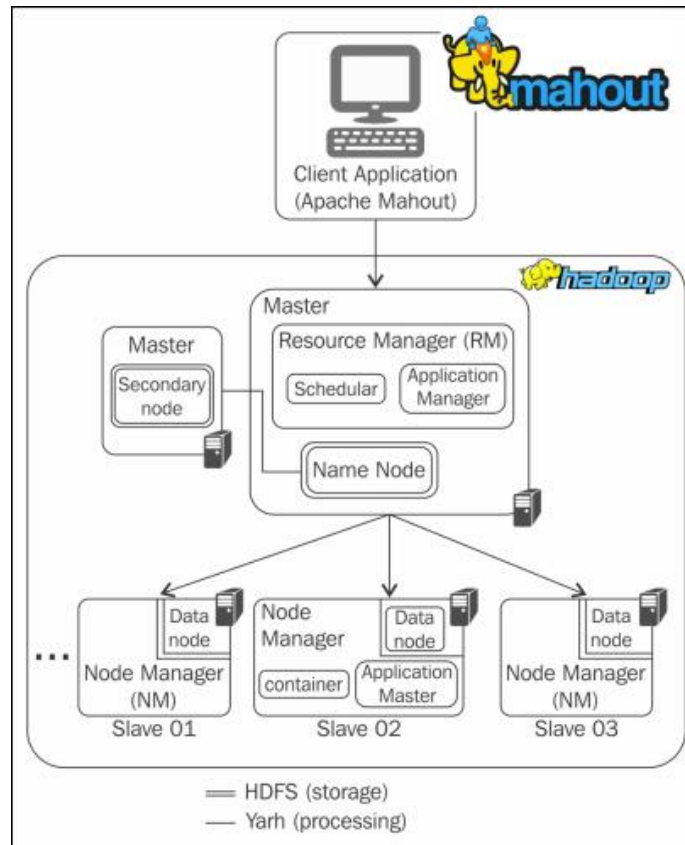
Flume

- Incluye datos externos, provenientes de aplicaciones web, servidores o aplicaciones móviles.



Mahout

- Proporciona una interfaz de procesamiento de algoritmos escalables de machine learning y minería de datos sobre Hadoop.



HUE

- Hadoop User Experience, capa web front-end.
- Integra, bajo una misma interfaz:
- Hadoop
 - File Browser para HDFS
 - Job Browser para MapReduce/YARN
 - Apache HBase Browser
 - Query editors para Apache Hive, Apache Pig, Impala
 - Editor para Apache Sqoop2
 - Editor Apache Oozie
 - Constructor de dashboards Apache Solr
 - Apache ZooKeeper Browser
- Generic
 - Buscador visual para Apache Solr
 - Spark Editor
 - SQL Editor para BDD tradicionales

HUE

Navigator

Settings

DATABASE

default

Table name...

page_view
 tweets
 business
 city (string)
 review_count (int)
 name (string)
 neighborhoods (string)
 type (string)
 business_id (string)
 full_address (string)
 state (string)
 longitude (float)
 stars (float)
 latitude (float)
 open (boolean)
 categories (string)
 top_cool4_hbase
 top_reviews
 review
 top_cool
 top_cool_hbase
 timestamp_invalid_data
 test_partitions
 counties
 hanks

Sample: Salary growth

Salary growth (sorted) from 2007-08

```

1 SELECT s07.description, s07.salary, s08.salary,
2       s08.salary - s07.salary
3 FROM
4       sample_07 s07 JOIN sample_08 s08
5       ON ( s07.code = s08.code)
6 WHERE
7       s07.salary < s08.salary
8 ORDER BY s08.salary-s07.salary DESC
9 LIMIT 20

```

Execute

Save

Save as...

Explain

or create a

New query

...

Recent queries

Query

Log

Columns

Results

Chart

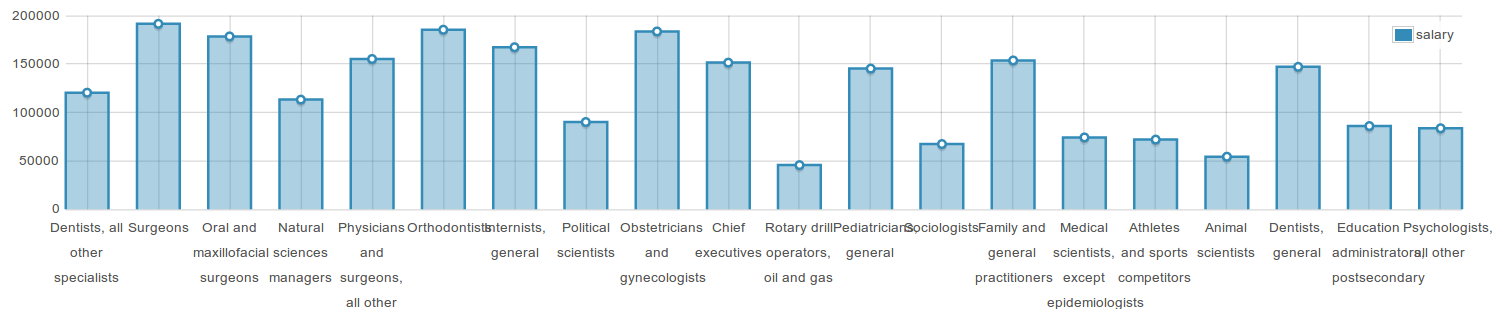


Chart type



X-Axis description

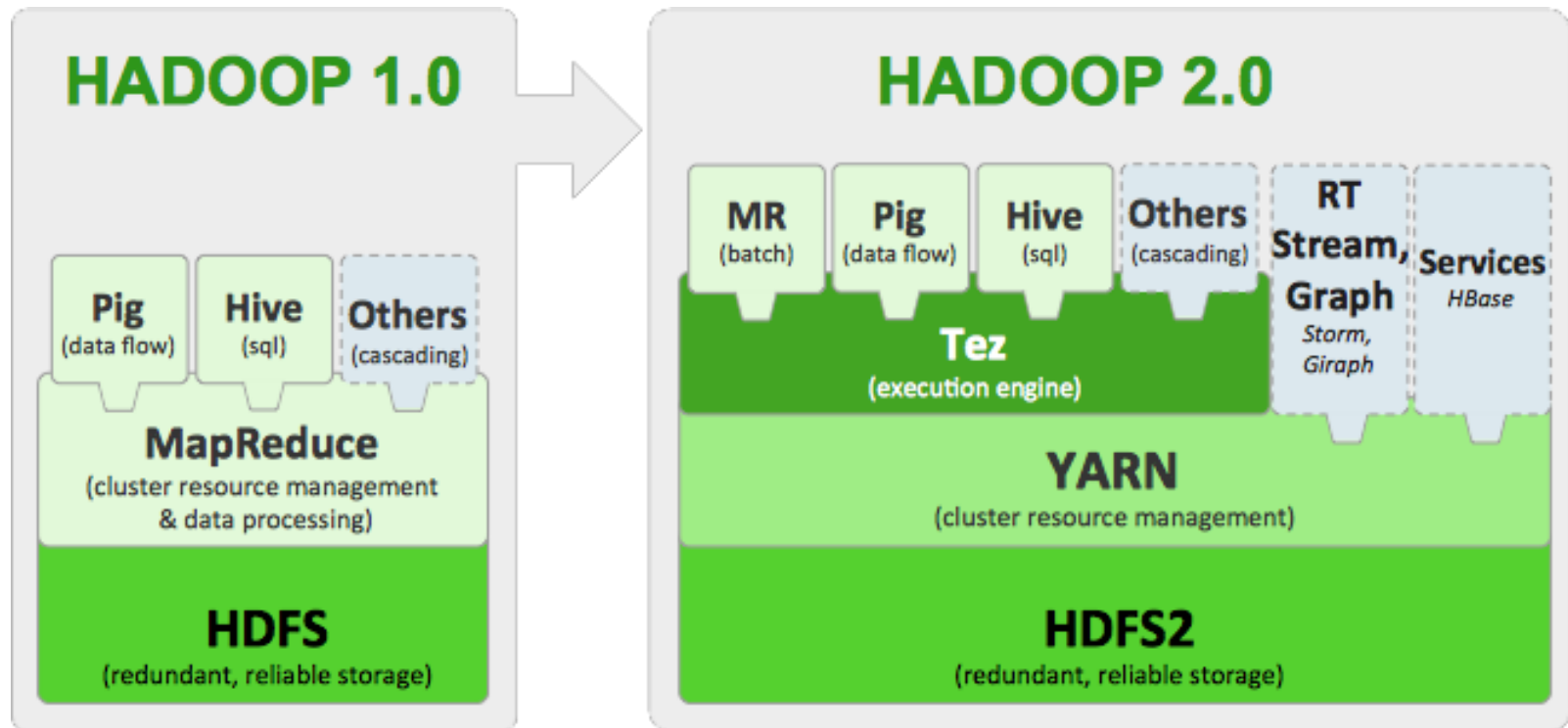
Y-Axis salary



Introducción

- Una vez Hadoop se estabiliza en el mercado, se identifican una serie de deficiencias en él principalmente derivadas de su dependencia con MapReduce:
 - Muy orientado a batch
 - Limitaciones en tamaño de clústers
 - Mal rendimiento en datasets pequeños
 - Problemas en alta disponibilidad, especialmente las dependencias hacia los NameNodes.
 - Otros
- En 2008 se inicia una línea de trabajo que duraría 4 años, resultando en un cambio radical en la arquitectura concebida.
- Se diseña un nuevo FS (HDFS2) y una nueva versión de MapReduce (MapReduce2), pero sobretodo introduce un nuevo gestor de recursos: YARN.

Arquitectura



YARN

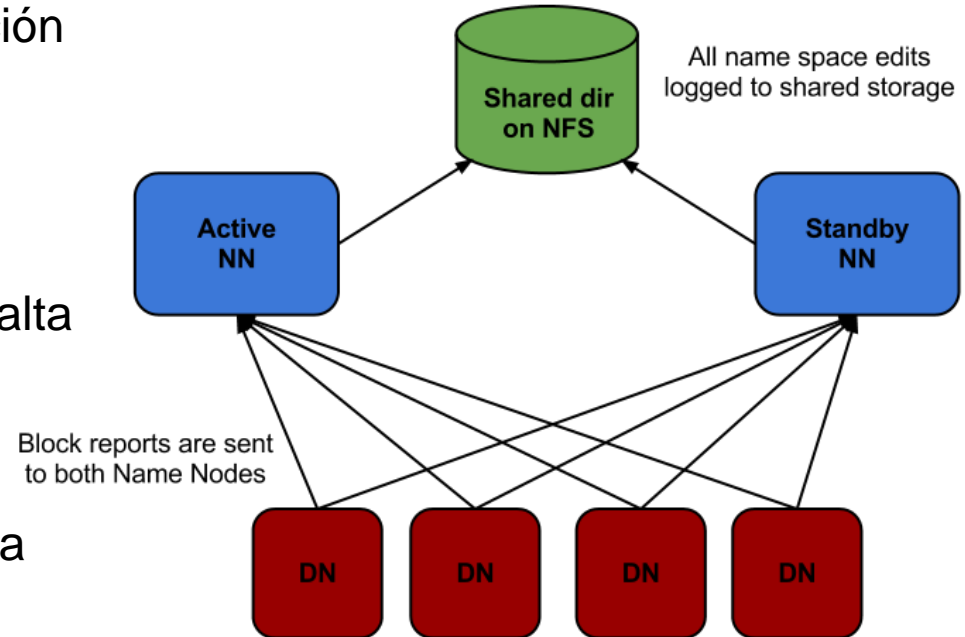
- Un único Jobtracker es un cuello de botella en grandes clusters (más de 4000 nodos)
- YARN (Yet Another Resource Negotiator): aparece una nueva arquitectura.
- Característica principal: separa las dos funcionalidades del Jobtracker (gestión de recursos y job-scheduling/monitorización) en demonios separados:
 - resource manager
 - node managers
 - application masters
- MapReduce es uno de los tipos de aplicación que pueden correr sobre YARN

Demonios YARN

- Resource manager:
 - arbitra los recursos entre las aplicaciones en el sistema
 - demonio global, obtiene datos del estado del cluster de los node managers
 - dos componentes:
 - Scheduler: planifica aplicaciones en base a sus requerimientos de recursos (se basa en el concepto de Container de recursos) .
 - Applications Manager: acepta trabajos, negocia containers y gestiona fallos de los Application Masters
- Node managers uno por nodo monitorizan los recursos del cluster (containers)
- Application masters:
 - uno por aplicación, gestiona el ciclo de vida de la aplicación.
 - solicita recursos (containers) al Resource manager y ejecuta la aplicación en esos containers
 - trabaja con los Node managers para ejecutar y monitorizar las tareas
 - permiten que diferentes tipos de aplicaciones (no solo MapReduce) se ejecuten en el cluster

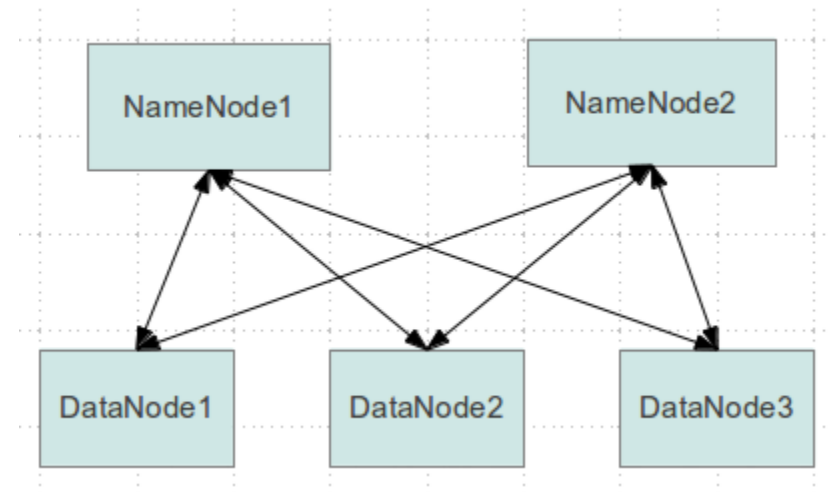
HDFS High-Availability

- Un par de Namenodes en configuración activo-standby: si falla el Namenode activo, el otro ocupa su lugar.
- Consideraciones:
 - Los Namenodes deben usar un almacenamiento compartido de alta disponibilidad
 - Los Datanodes deben enviar informes de bloques a los dos Namenodes (el block mapping va en memoria, no en disco)
 - Los Clientes deben manejar el fallo del Namenode de forma transparente

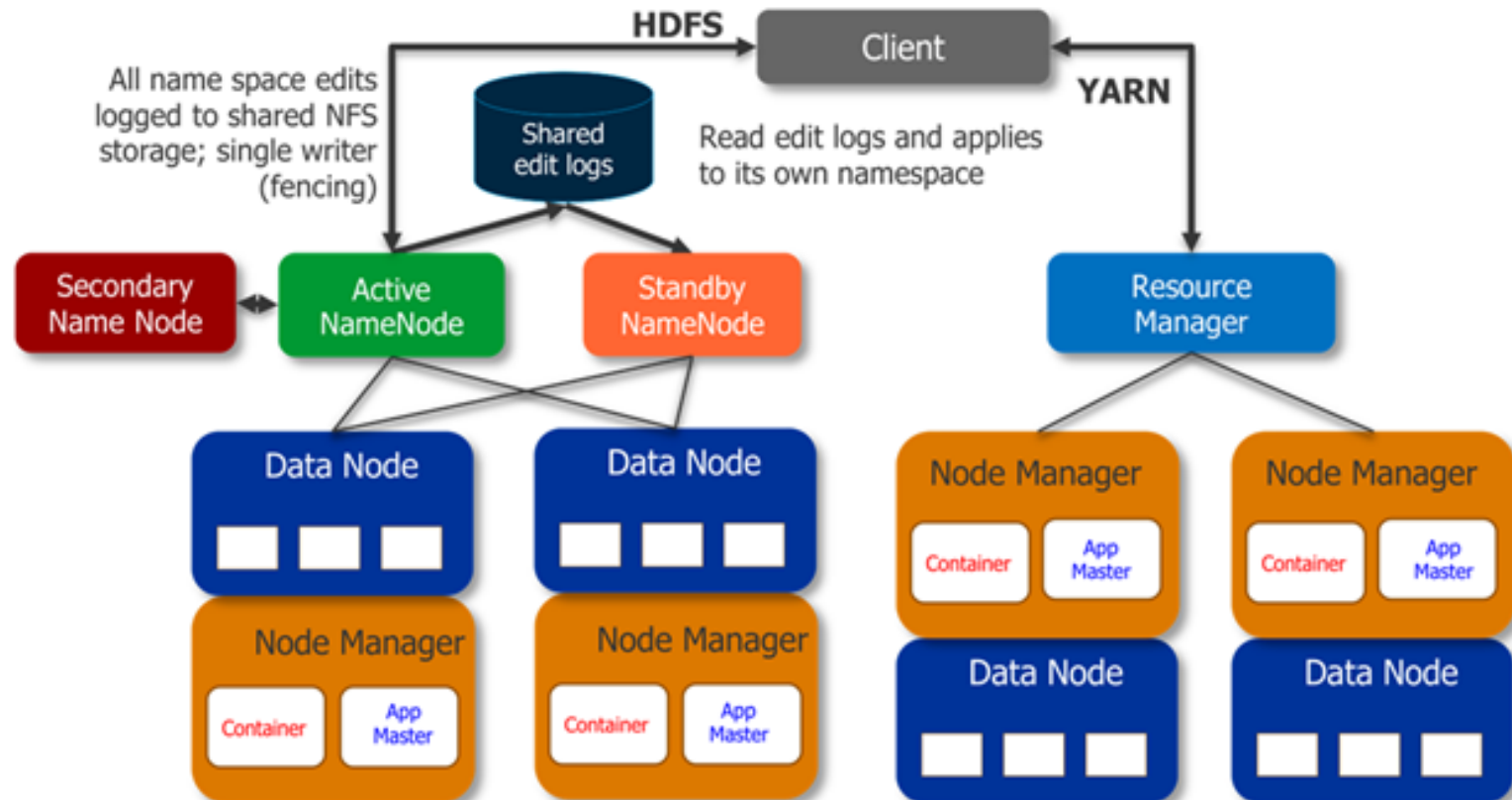


HDFS Federation

- El Namenode mantiene, en memoria, referencias a cada fichero y bloque en el filesystem:
 - problemas de escalabilidad
- HDF Federation, introducida en la versión 2.0.
- Permite usar varios Namenodes
- Cada uno gestiona una porción del espacio de nombres del filesystem.
- Los Namenodes no se coordinan entre sí.
- Cada Datanodes se registra con todos los Namenodes.



Resumen



Hadoop 1 vs Hadoop 2

Hadoop 1.X	Hadoop 2.X
Limitado a 4000 nodos por clúster	Hasta 10000 nodos por clúster
'0' número de trabajos en un clúster	'0' (tamaño de clúster)
Cuello de botella en Jobtracker	Utilización del clúster más eficiente (YARN)
Sólo tiene un namespace para manejar HDFS	Soporta múltiples namespaces para manejar HDFS
Los slots de Map y Reduce son estáticos	No restringido a Java
Sólo tiene un job para correr MapReduce	Cualquier aplicación se puede integrar con Hadoop