Fundamentos de Almacenamiento y Procesamiento

Fundamentos de Almacenamiento

Dataging
Managing through data

Agenda

- **Introducción a HDFS**
- Línea de Comandos
- Direccionamiento de bloques y Red
- HDFS 2.0

Introducción

- Sistema de Ficheros para almacenar ficheros muy grandes
- Patrón de Escribe una vez, lee muchas veces
- Commodity Hardware
- Gran trasiego de datos
- Self-Healing High-Bandwidth Clustered Storage

Introducción

- HDFS no ofrece buen rendimiento para:
  - Accesos de baja latencia
  - Ficheros pequeños (a menos que se agrupen)
  - Múltiples “escritores”
  - Modificaciones arbitrarias de ficheros

Bloque HDFS

- Cantidad mínima de datos que puede ser leída o escrita.
- Bloques de Sistema de ficheros tienen habitualmente unos pocos kilobytes, los de disco son habitualmente de 512 bytes.
- El tamaño predeterminado de HDFS son 64 MB.
- Las tareas Map en MapReduce operan habitualmente con un bloque.
- Gran tamaño para optimizar búsquedas.

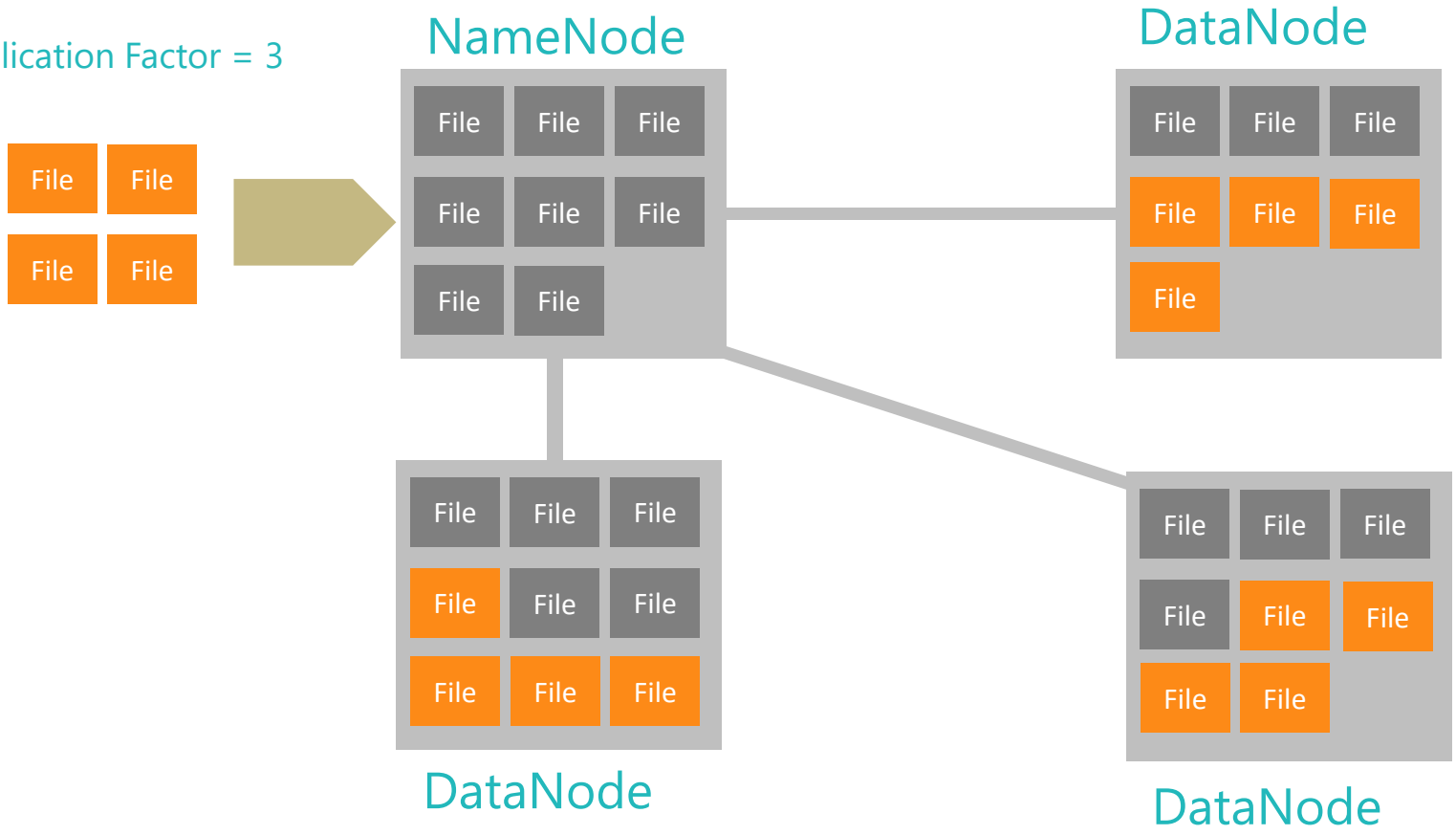
Namenodes y Datanodes

- 1 Namenode
  - El maestro
- Multiples Datanodes
  - Los “trabajadores”

Un cliente escribiendo datos en HDFS

Block Size = 64 Mb

Replication Factor = 3



Dataging
Managing through data

Namenode

- Solo uno.
- Gestiona el espacio del Sistema de ficheros
- Mantiene el árbol del Sistema de ficheros y los metadatos para todos los ficheros y directorios en el árbol.
- Es posible ejecutar un NameNode secundario

DataNodes

- Más de uno
- Almacena y lee bloques.
- Recuperado por Namenode o clientes
- Reportan al Namenode la lista de bloques que están almacenando.

Factor Replicación

- No para directorios
- Ficheros y Bloques
- Configuración para todo el cluster
- Se lista con ls
- Valor predeterminado es 3
  - Menor valor, más espacio, menos tolerancia datos en menos nodos

Formatos de fichero

- Texto / CSV
- JSON
- Avro
- Sequence – Formato nativo Hadoop
- Almacenamiento Columnar
  - RC
  - ORC (Horthonworks)
  - Parquet (Cloudera / Impala)

Agenda

- Introducción a HDFS
- **Línea de Comandos**
- Direccionamiento de Bloque y Red
- HDFS 2.0

Interface Línea de comandos

- HDFS por defecto puerto 8020. Hdfs//localhost/
- Interface POSIX

  hadoop fs -help

cmd hadoop fs

-ls	-lsr	-du	-dus
-count	-mv	-cp	-rm
-rmr	-expunge	-put	-copyFromLocal
-moveFromLocal	-get	-getmerge	-cat
-text	-copyToLocal	-moveToLocal	-mkdir
-setrep	-touch	-test	-stat
-tail	-chmod	-chown	-chgrp

Permisos de ficheros

- Cómo POSIX
  - (r)ead
  - (w)write
  - e(x)ecute
- Se pueden aplicar a ficheros o directorios -rw-r—r– or drwxr-xr-x
- Primer grupo → Owner
- Segundo grupo → Group
- Tercer grupo→ Mode

Patrones de Ficheros / Caracteres Globales

*	asterisk	matches zero o more characters
?	question mark	matches a single character
[ab]	character class	matches a singles character in the set {a,b}
[^ab]	negated character class	Matches a single character that is not in the set {a,b}
[a-b]	character range	matches single character in the (closed) range [a,b], where a is lexicographically less than or equal to b
[^a-b]	negated character range	matches single character that is not in the (closed) range [a,b], where a is lexicographically less than or equal to b
{a,b}	alternation	matches either expression a or b
\c	escaped character	maches character c when it is a metacharacter

cmd: hadoop

- namenode –format
  - Formatea el name node
- Secondarynamenode
  - Habilita un nodo secundario

cmd: hadoop

- dfsadmin
  - Comando de Administración para la configuración DFS.
- fsck
  - Chequeo de Sistema de Ficheros

cmd Hadoop distcp

- Copia paralela
- Para copiar gran cantidad de datos hacia o desde Hadoop
- Implementado como MapReduce
  - La copia hecha por los mappers no los reducers.

cmd hadoop archive

- Hadoop archive (HAR) utilizando la herramienta de archive, se crea a partir de una colección de pequeños ficheros
- Ficheros pequeños no usan el tamaño de bloque en los datanodes, pero no son óptimos para namenodes
- Ficheros HAR pueden ser entrada de MapReduce
- HAR crea una copia de los originales
- HARs son inmutables

Agenda

- Introducción a HDFS
- Línea de Comandos
- **Direccionamiento de Bloque & Red**
- HDFS 2.0

Topología de Ancho de Banda de Red

- Hadoop representa la red como un árbol.
- Distancia entre 2 nodos es la suma de sus distancias al antecesor más cercano en común.
- Hadoop necesita ayuda para configurar la topología de red
- Sirve bloques de los nodos más cercanos

Ubicación de Réplicas

- Fiabilidad
- Ancho de banda escritura
- Ancho de banda lectura
- Ejemplo para factor de replicación 3
  - Primera Replica: mismo nodo (sino aleatorio) como cliente
  - Segunda Replica: rack diferente al primero de forma aleatoria
  - Tercera Replica: mismo rack que el Segundo pero diferente nodo

Respuesta a fallos

- Si el NameNode no recibe heartbeat o informe de bloques de un DataNode
  - Marca como muerto
    - No envía nuevos IO
    - Los bloques que se queden por debajo del factor de replicación se re-replican en otros nodos
- Corrupción
  - Se calcula un checksum cuando se crea el fichero
  - Estos checksums se almacenan en el namespace
  - Se comprueba en las lecturas. Si no es correcto se lee de nuevo de otra réplica

Demo

- HDFS

Laboratorio 02

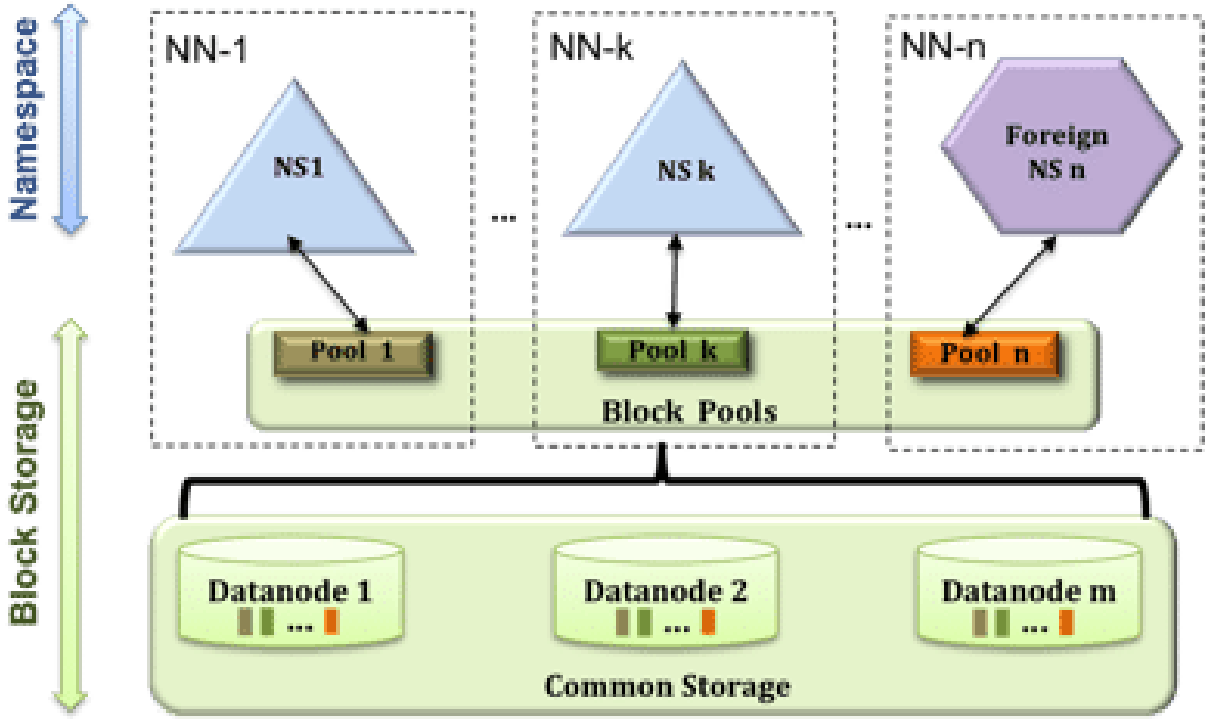
Hadoop Distribution File System

Agenda

- Introducción a HDFS
- Línea de Comandos
- Direccionamiento de bloques y Red
- **HDFS 2.0**

HDFS 2.0 – Federación

- Escalabilidad de Namespaces



Dataging
Managing through data

Agenda

- Introducción a HDFS
- Linea de Comandos
- Direccionamiento de bloques y Red
- HDFS 2.0

Procesamiento

Agenda

- Framework de Hadoop 2.0
- YARN
- Ficheros de Configuración

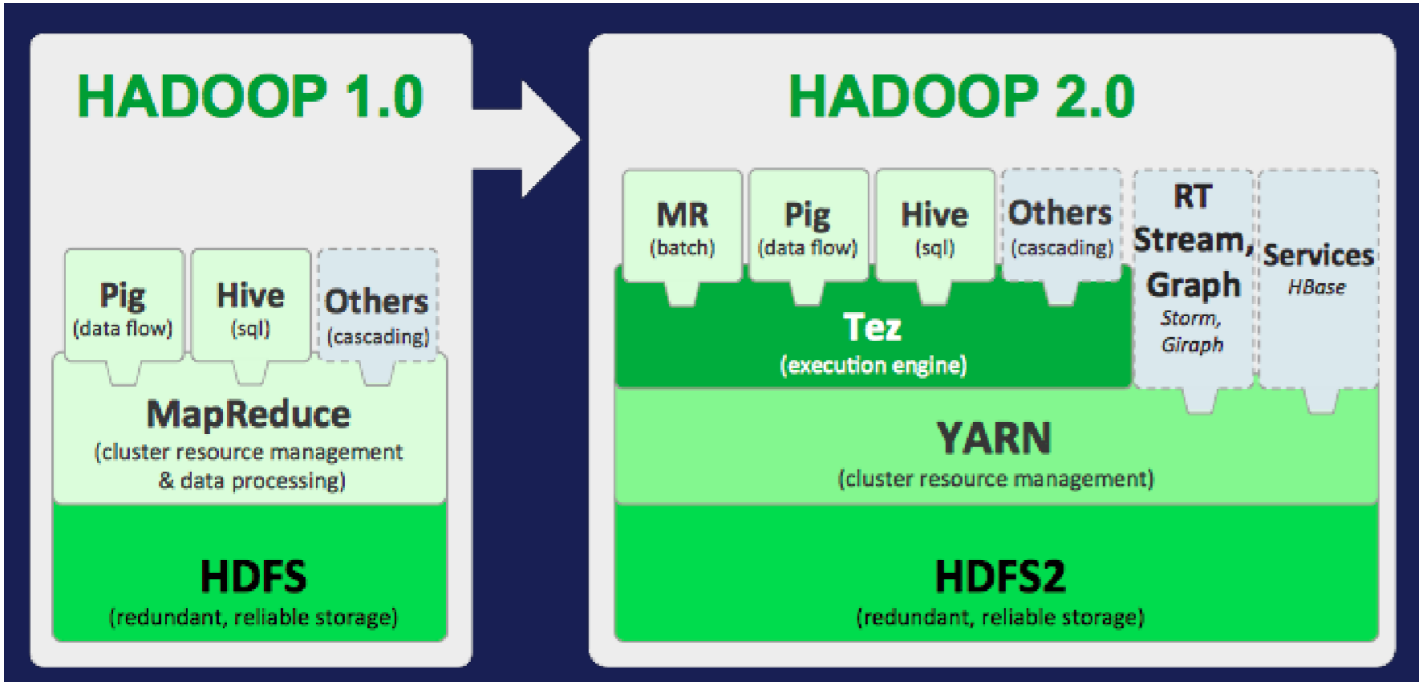
Hadoop 2.0

Single Use System

Batch Apps

Multi Purpose Platform

Batch, Interactive, Online, Streaming, …



Dataging
Managing through data

Servicios Hadoop

- Servicios HDFS
  - Gestiona el almacenamiento
  - Las unidades son NameNodes y DataNodes
  - NameNode – mantiene metadatos, en memoria, sobre la estructura hdfs y nombres
  - DataNodes – nodos que comunican el NameNode cambios en hdfs o actualizaciones durante las computaciones locales
- Servicios YARN
  - ResourceManager – el servicio “maestro” para el cluster que se ejecuta en uno de los headnodes
    - Responsable de direccionar los recursos del cluster y la planificación de trabajos en los nodos de trabajo
  - ApplicationMaster – Un servicio maestro único por aplicación.
    - Coordinada la ejecución de una aplicación en el cluster y negocia con el ResourceManager los recursos para la aplicación

Introducción a Map Reduce

- Basado en el framework de Google Map Reduce y en el Sistema de ficheros de Google
- Procesamiento de datos distribuidos a gran escala
- Pensado para hardware “commodity”
- Auto recuperable
- Escrito en Java

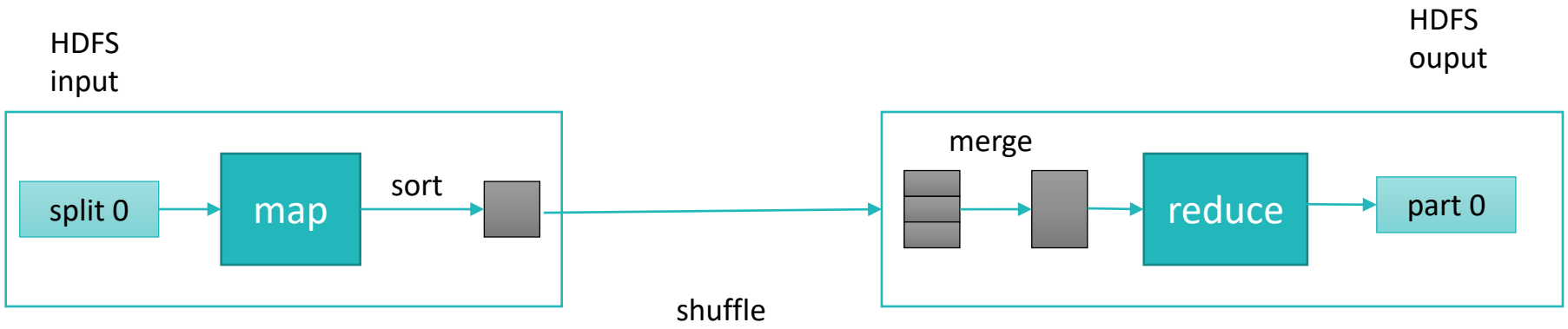
Introducción

- Map Reduce forma parte del Core de Hadoop junto con HDFS
- Nos proporciona un modelo de programación paralelo
- Divide una tarea entre procesadores “cerca” de los datos y ensambla los resultados
- Se encarga de programar y tolerancia a fallos
- Monitorización y reporte de Estado

¿Por qué MapReduce?

- Aplicaciones de procesado de gran cantidad de datos
- Divide los datos y procesa en varios nodos
- Cada aplicación maneja
  - Comunicación entre los nodos
  - División y programación del trabajo
  - Tolerancia a Fallos
  - Monitorización y reporting

Map Reduce – Single Reducer



Map Reduce

- Basado en lenguajes funcionales
- Ejecuta una función cerca de la partición de datos
- Uno o más mappers por host
- Múltiples hosts ejecutando mappers
- Tiene como salida un par Clave / Valor
- **Map** f listas: aplica una función f a cada elemento de una lista y devuelve una nueva lista
  - Map square [1 2 3 4 5]=[1 4 9 16 25]

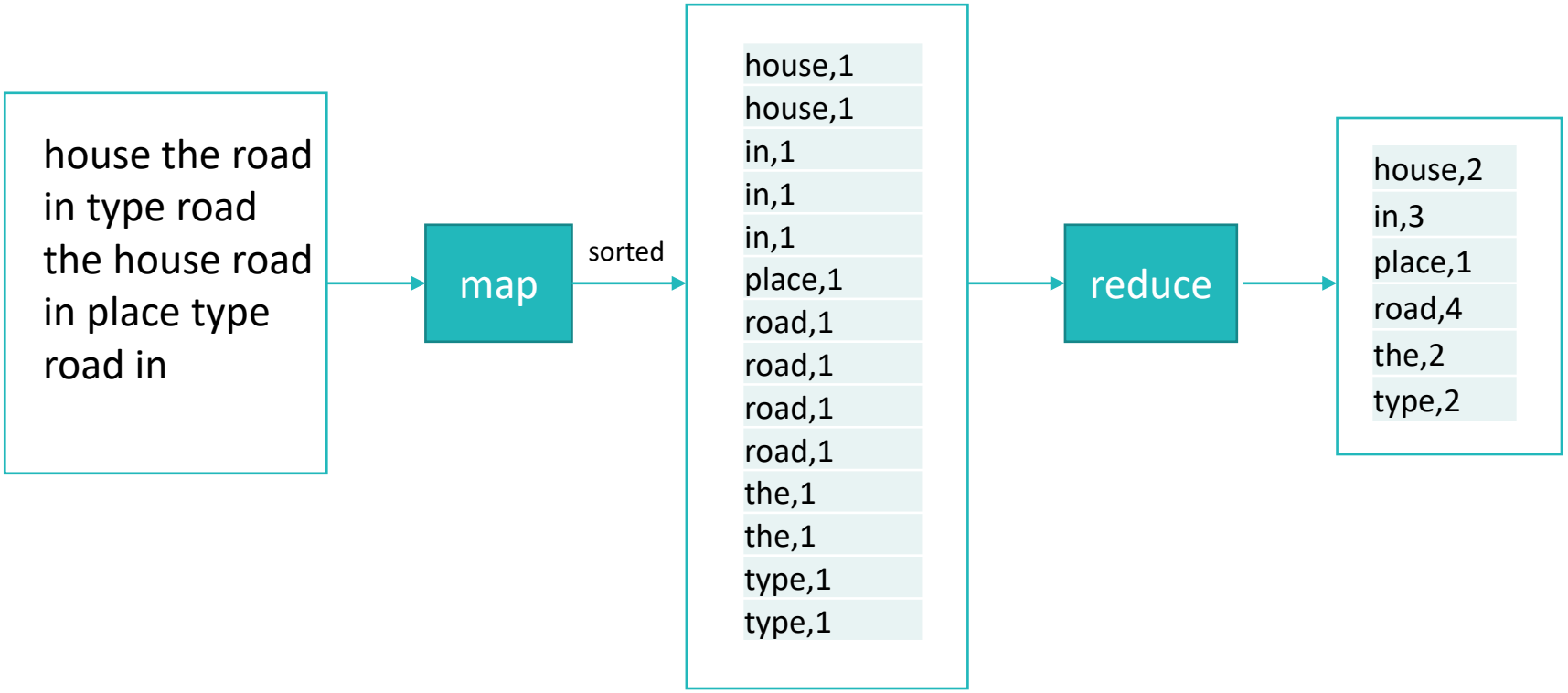
Map Reduce

- Basado en lenguajes funcionales
- Se ejecuta como una fase posterior después de la fase mapper
- Uno o más reducers por host
- Múltiples hosts ejecutando reducers
- **Reduce** g lista: combina elementos de una lista utilizando la función g para generar un nuevo valor
  - Reduce sum[1 2 3 4 5]=[15]

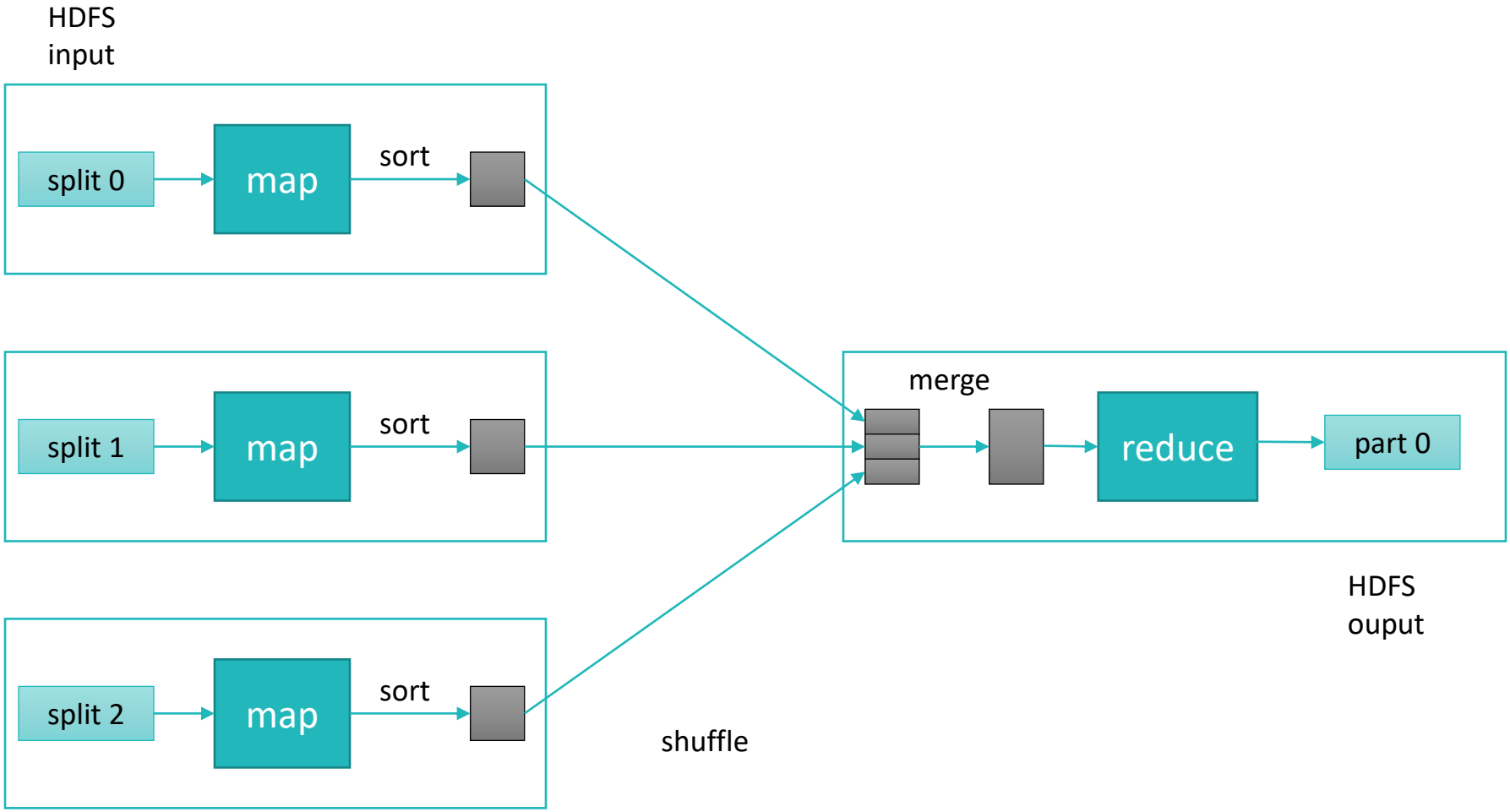
Jobtracker y Tasktracker

- JobTracker -- Maestro
  - Divide las tareas según la ubicación de los datos
  - Programa y monitoriza varias tareas map reduce
- Task Tracker -- Esclavos
  - Ejecuta tareas map y reduce

Ejemplo Cuenta Palabras

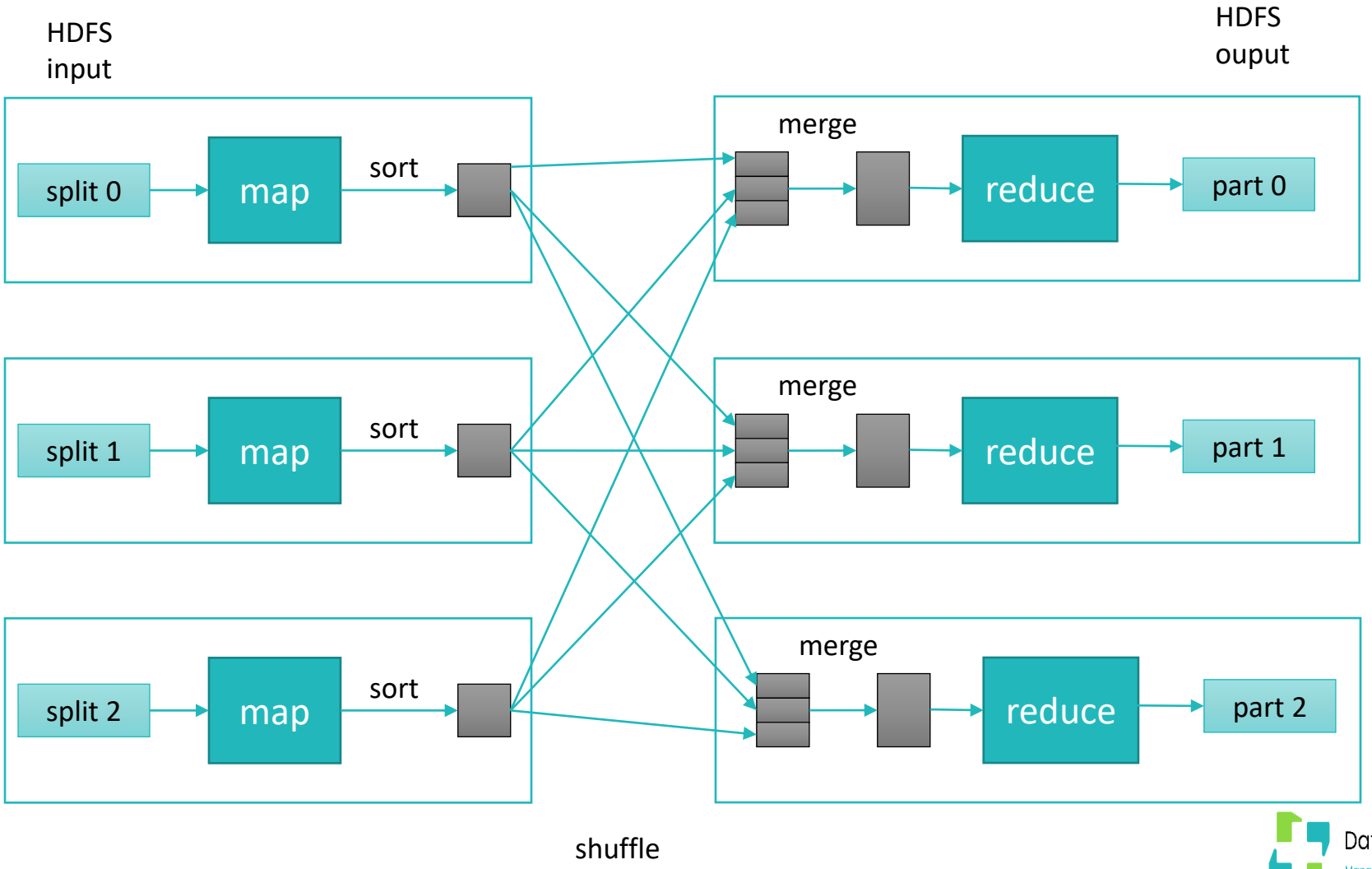


Map Reduce



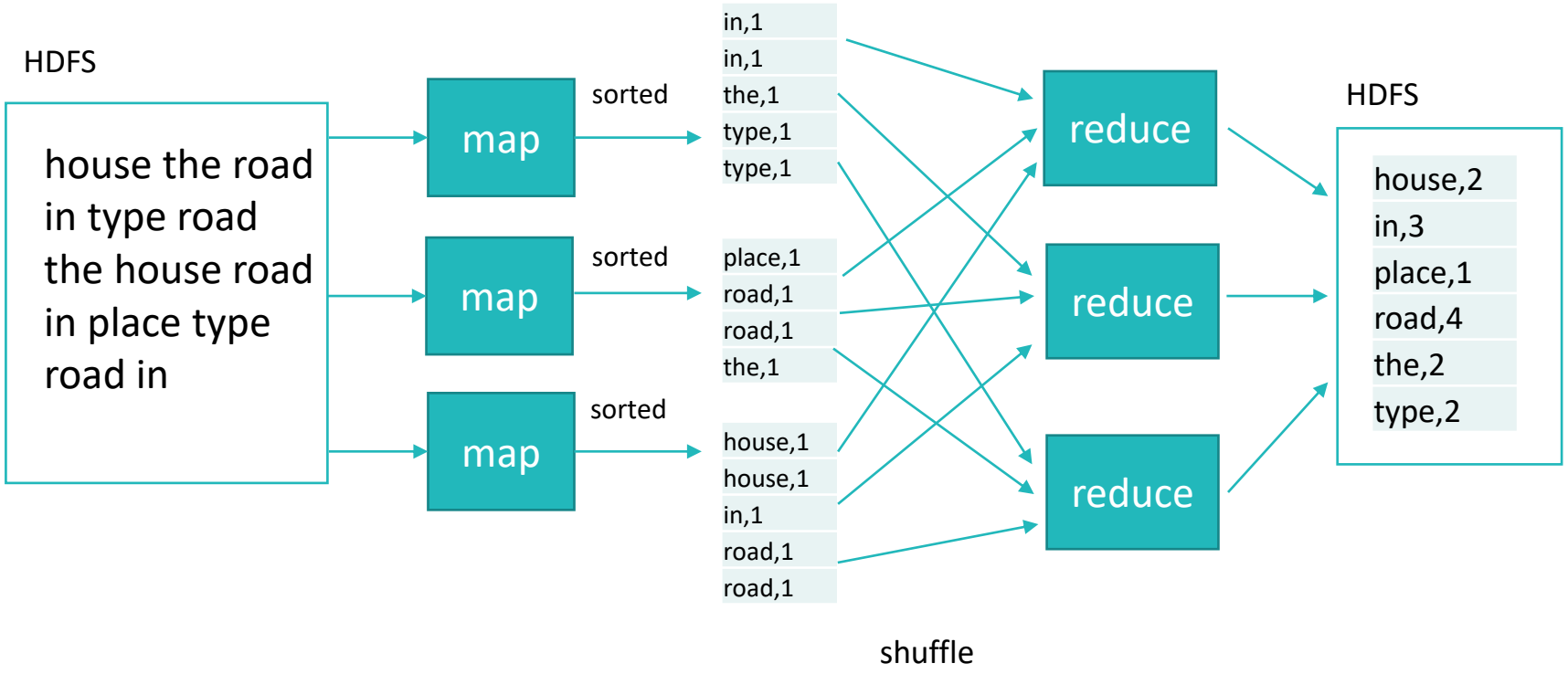
Dataging
Managing through data

Map Reduce



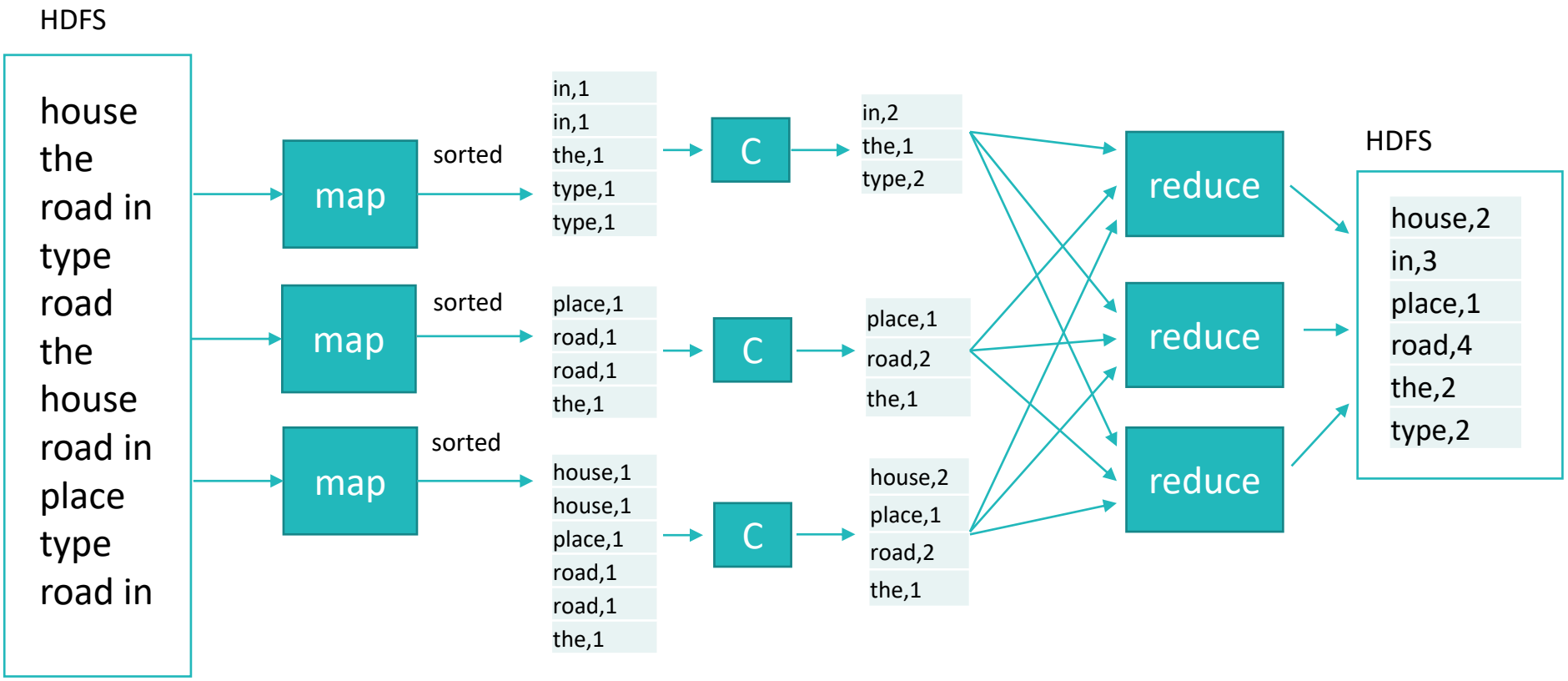
Dataging
Managing through data

Ejemplo cuenta palabras



Dataging
Managing through data

Combiner



Dataging
Managing through data

Combiner

- La función Combiner después y sobre la salida de la función map
- Reduce ancho de banda de red
- La función debe de ser **acumulativa** y **asociativa**
  - No se puede utilizar para todo tipo de cálculos, como medias o significados
- Funciona con Cuenta palabras, Máximos, etc

En Resumen….

Hadoop divide el fichero de entrada y asigna cada trozo a un mapper diferente.

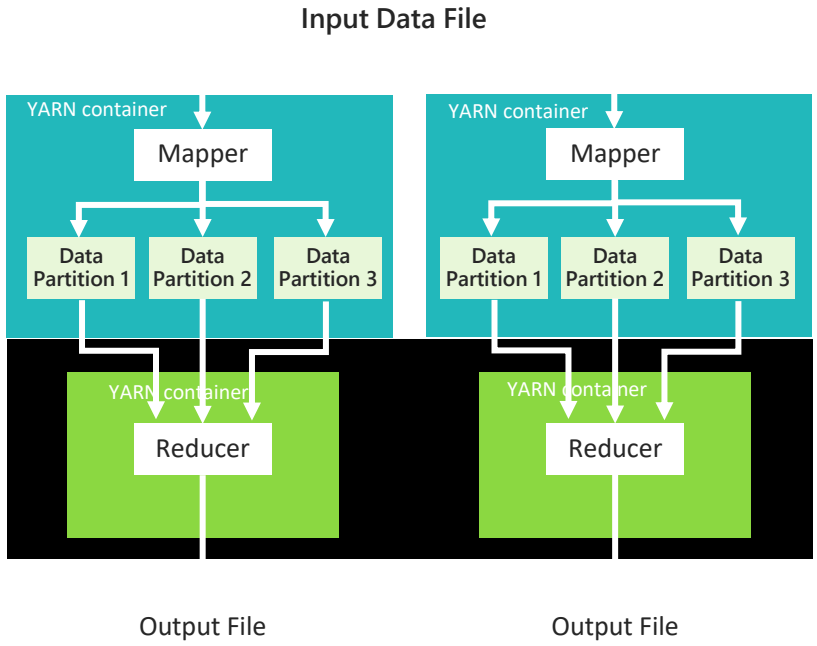
Hadoop lee localmente el trozo línea por línea y llama a map() para cada línea, pasándolo como parámetros clave / valor

El mapper genera otro par de clave / valor intermedio

Todos los valores intermedios para una clave intermedia se combinan juntos en una lista.

La lista se le da a uno o varios Reducers

- Todos los valores asociados con una clave intermedia van al mismo Reducer
- Se envían ordenadas por clave '`shuffle` and `sort`'
- Hadoop llama `reduce()` por cada línea de entrada
- El Reducer genera los pares clave / valor finales que se escriben en HDFS

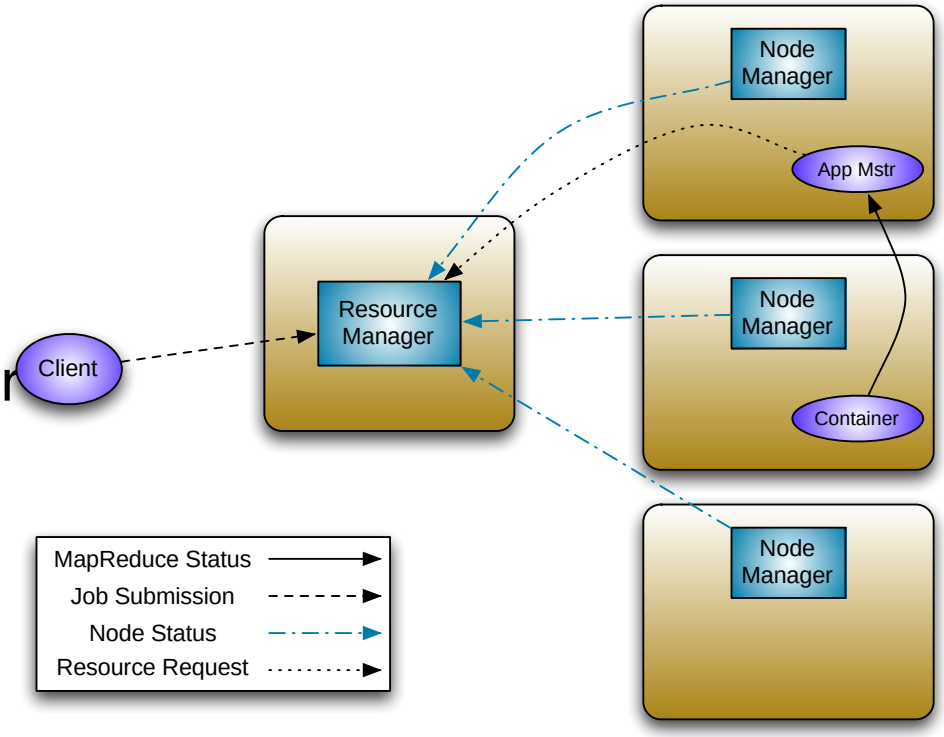


Agenda

- Framework de Hadoop 2.0
- **YARN**
- Ficheros de Configuración

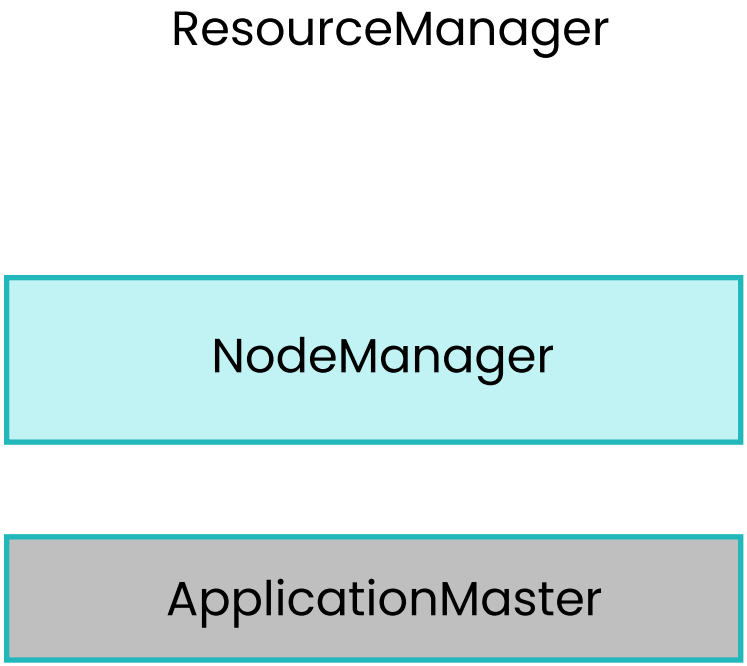
Arquitectura YARN

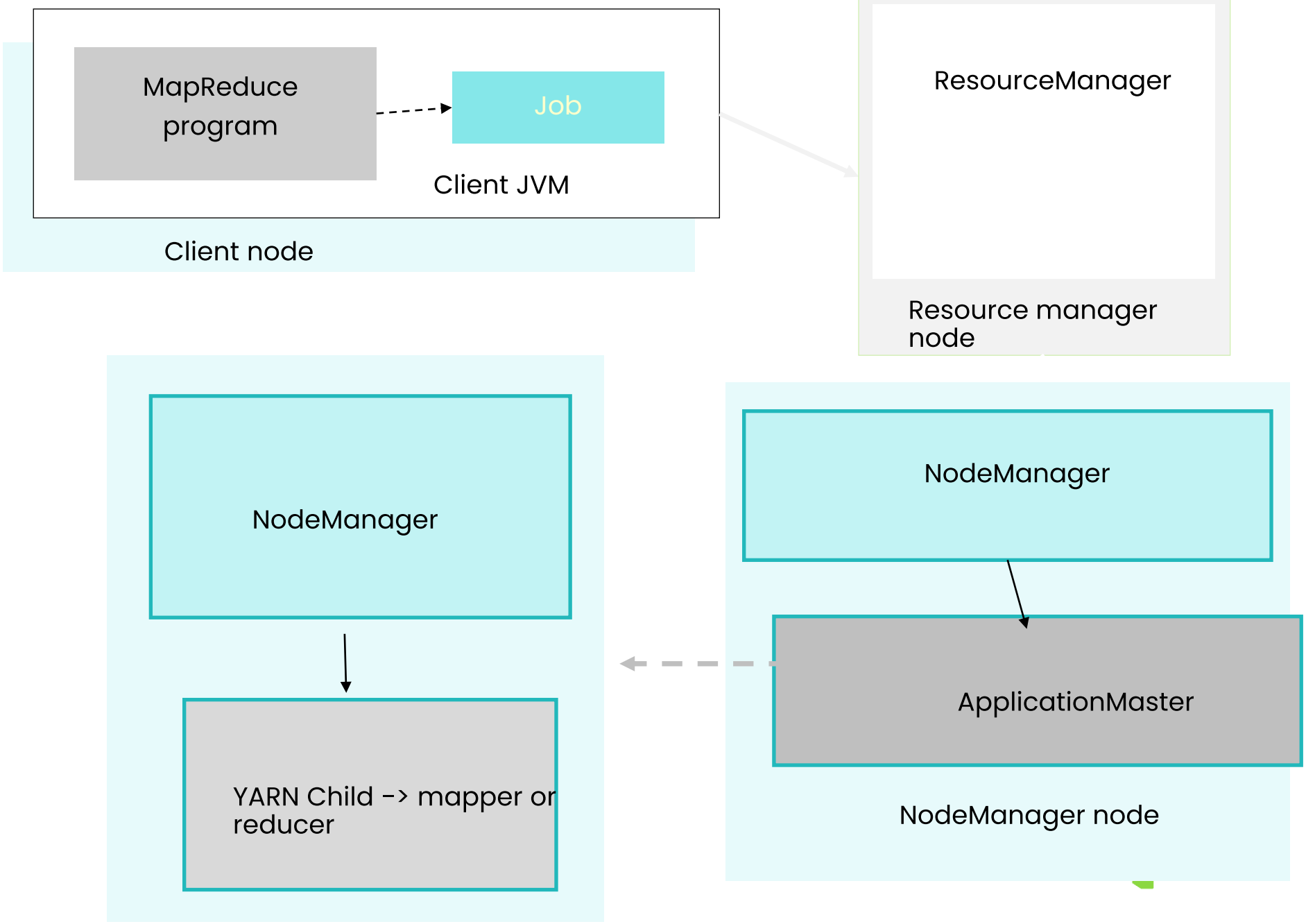
- Resource Manager
- Node Manager
- Application Manager



YARN Daemons

- ResourceManager (RM)
  - Se ejecuta en el nodo maestro
  - Planificador de recursos Global
  - Arbitra los recursos del Sistema entre aplicaciones
  - Tiene un planificador externo para soportar diferentes algoritmos
- NodeManager (NM)
  - Se ejecuta en un nodo de trabajo (en un contenedor)
  - Se comunica con RM para asegurar que tiene recursos y que está vivo
- Application Master (AM)
  - Uno por aplicación
  - Se ejecuta en un contenedor

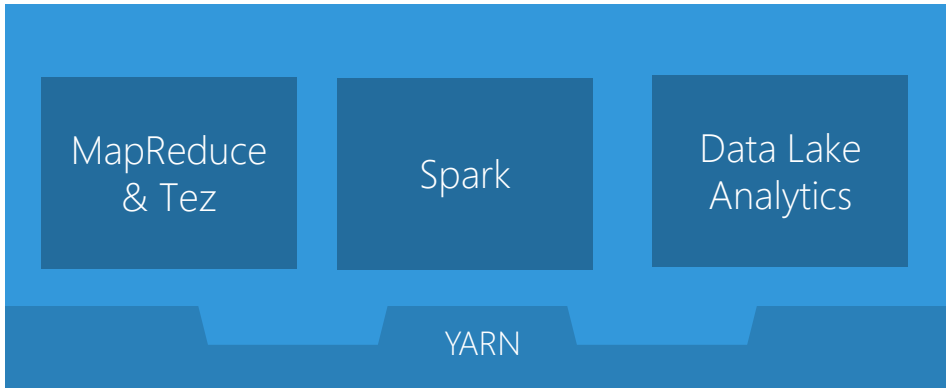



MapReduce program
Job
Client JVM
Client node
ResourceManager
Resource manager node
NodeManager
YARN Child -> mapper or reducer
NodeManager
ApplicationMaster
NodeManager node

Mejoras clave con YARN

- Framework que soporta múltiples aplicaciones
- Utilización del cluster
- Escalabilidad
- Agilidad
- Servicios Compartidos

Cargas de Trabajo en YARN

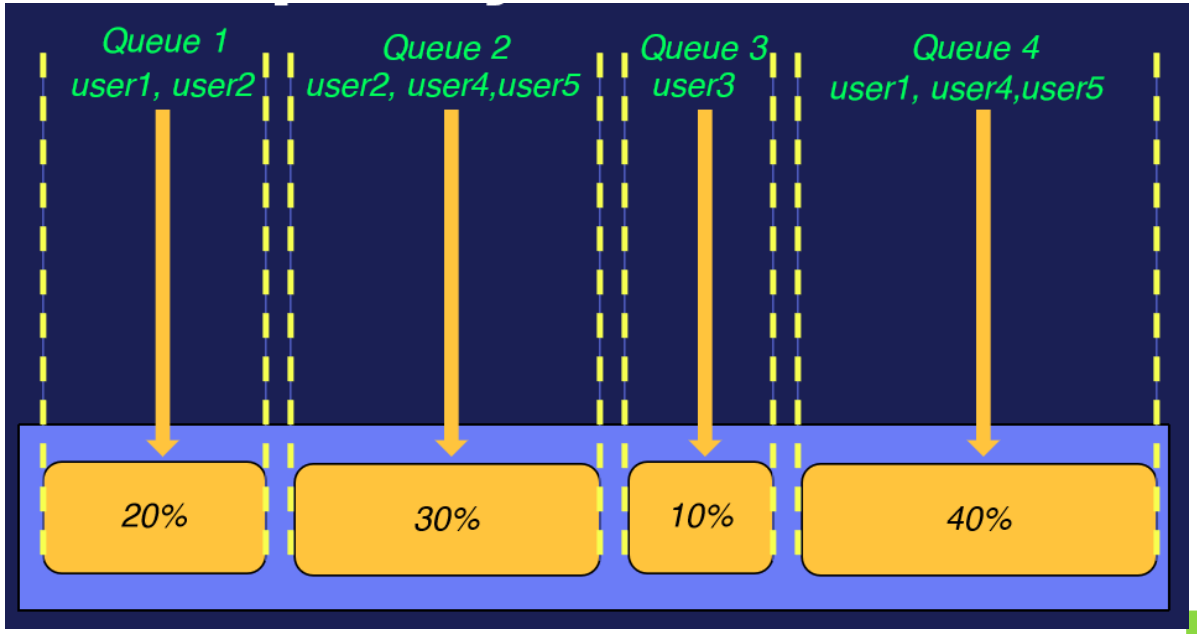


Gestión de recursos (scheduling)

- FIFO → predeterminado
- Por capacidad
- Fairshare

Por capacidad

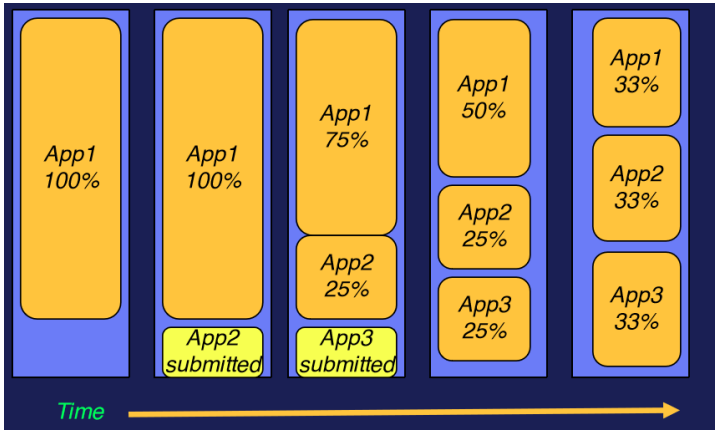
- Definición de colas en YARN
- Asignación basada en recursos
- ACL para seguridad



Dataging
Managing through data

Fairshare

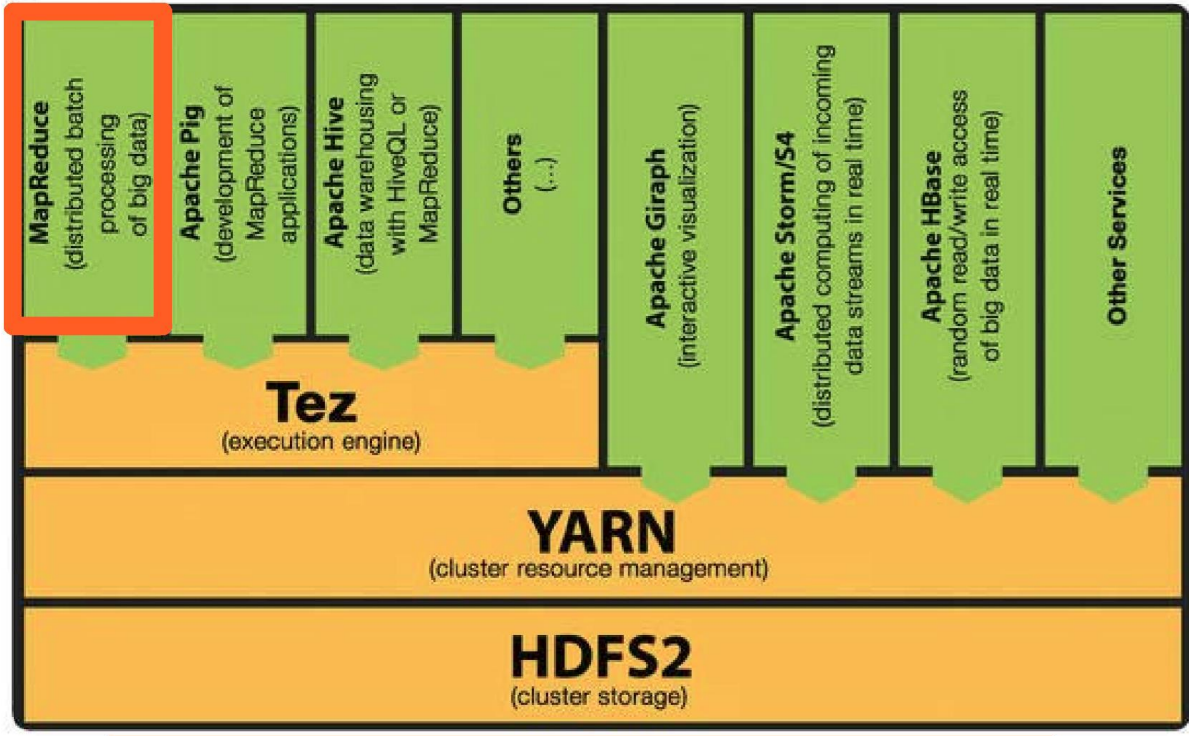
- Balancea recursos entre las aplicaciones a lo largo del tiempo
- Se pueden aplicar pesos a las aplicaciones
- Límites por usuario / aplicación



Dataging
Managing through data

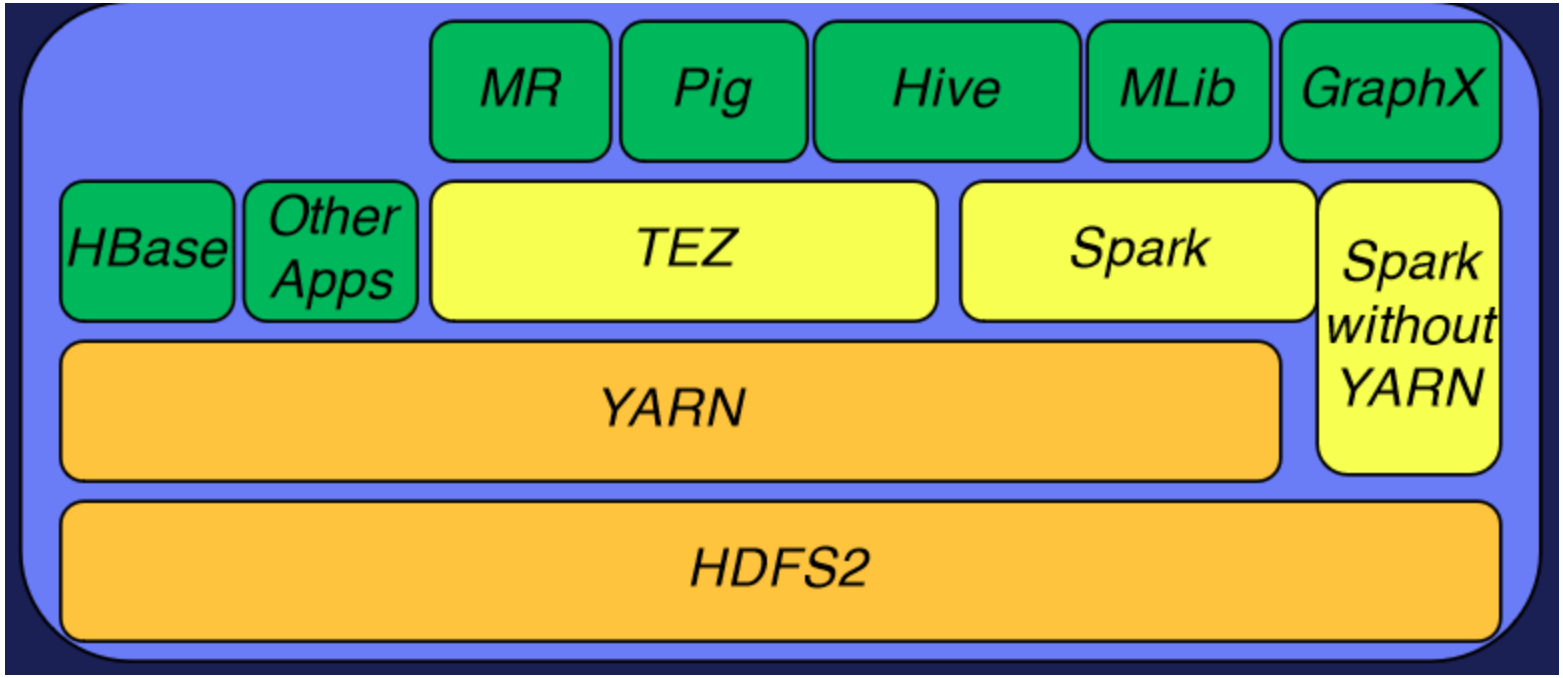
Recuerda...

- MapReduce sigue existiendo como un framework basado en YARN para procesamiento paralelo de datos



Dataging
Managing through data

Los nuevos Frameworks de ejecución



Dataging
Managing through data

TEZ

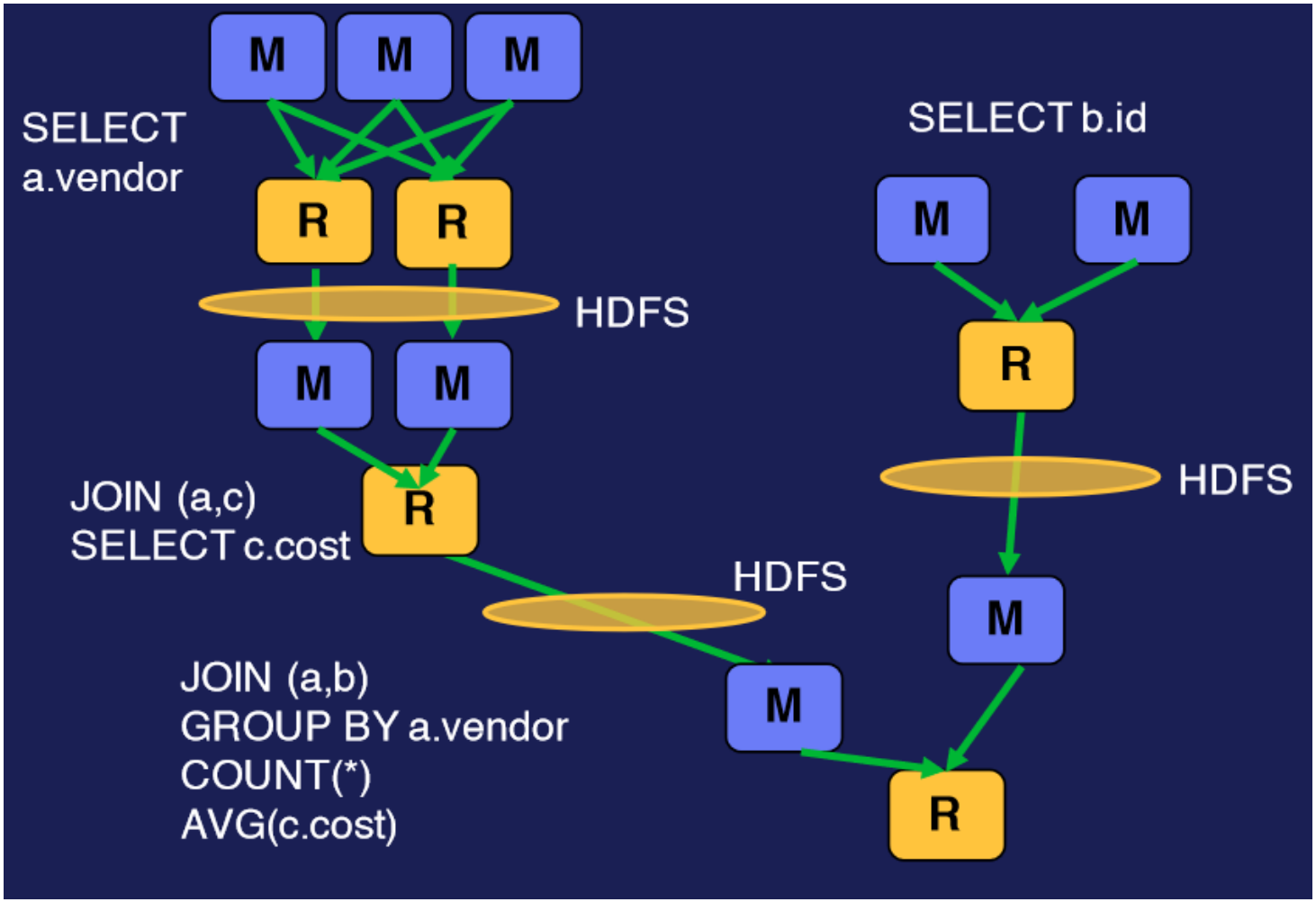
- DAG
  - Grafos de flujos de datos acíclicos (no iterativos)
- Puede ejecutar DAG complejos
- Soporte cambios dinámicos
- Mejor gestión de recursos
  - “Algo” de caché

Imagina esta consulta.....

```
SELECT a.vendor, COUNT(*), AVG(c.cost) FROM a
JOIN b ON (a.id = b.id)
JOIN c ON (a.itemid = c.itemid)
GROUP BY a.vendor
```

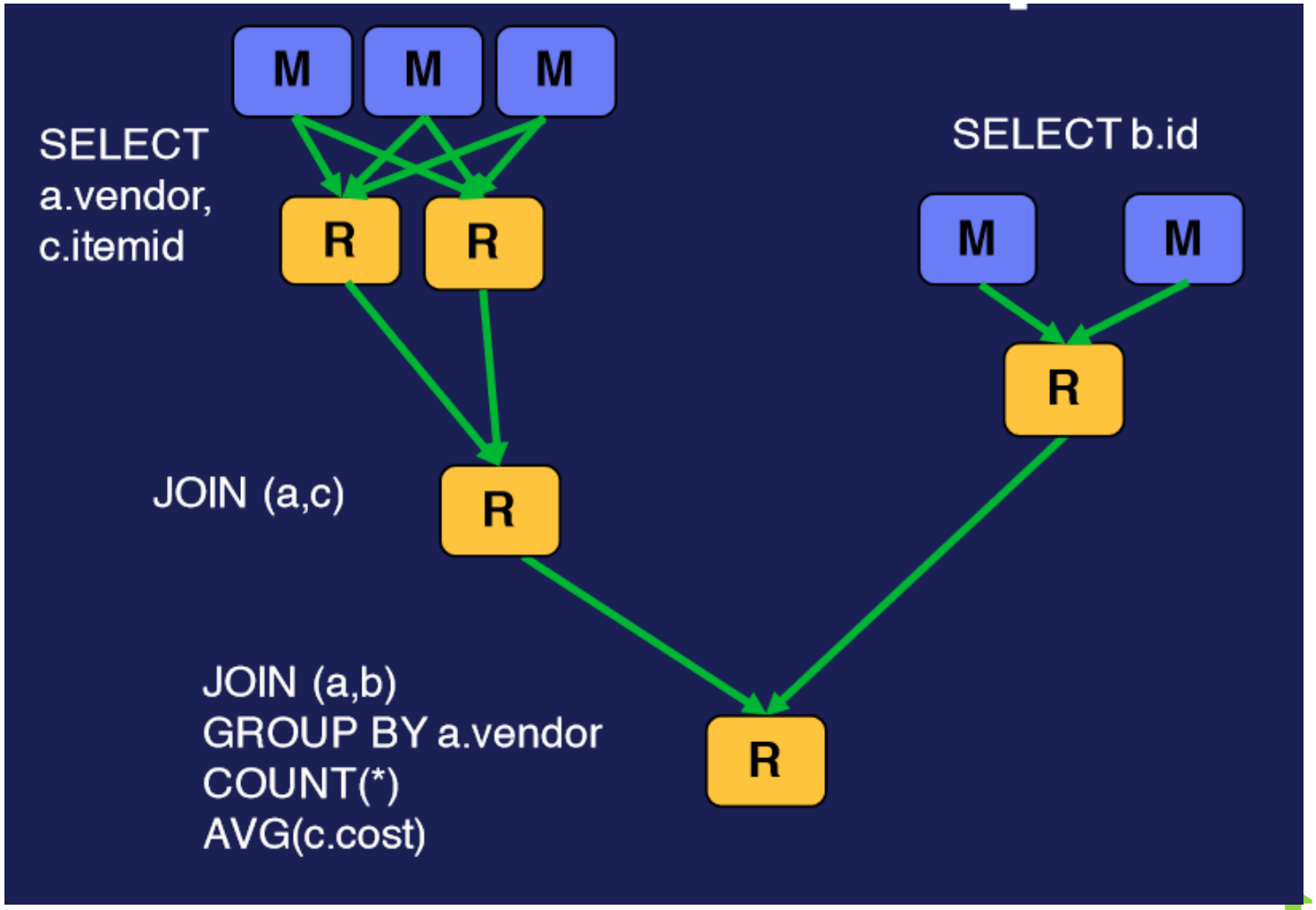
Dataging
Managing through data

Con MapReduce....



Dataging
Managing through data

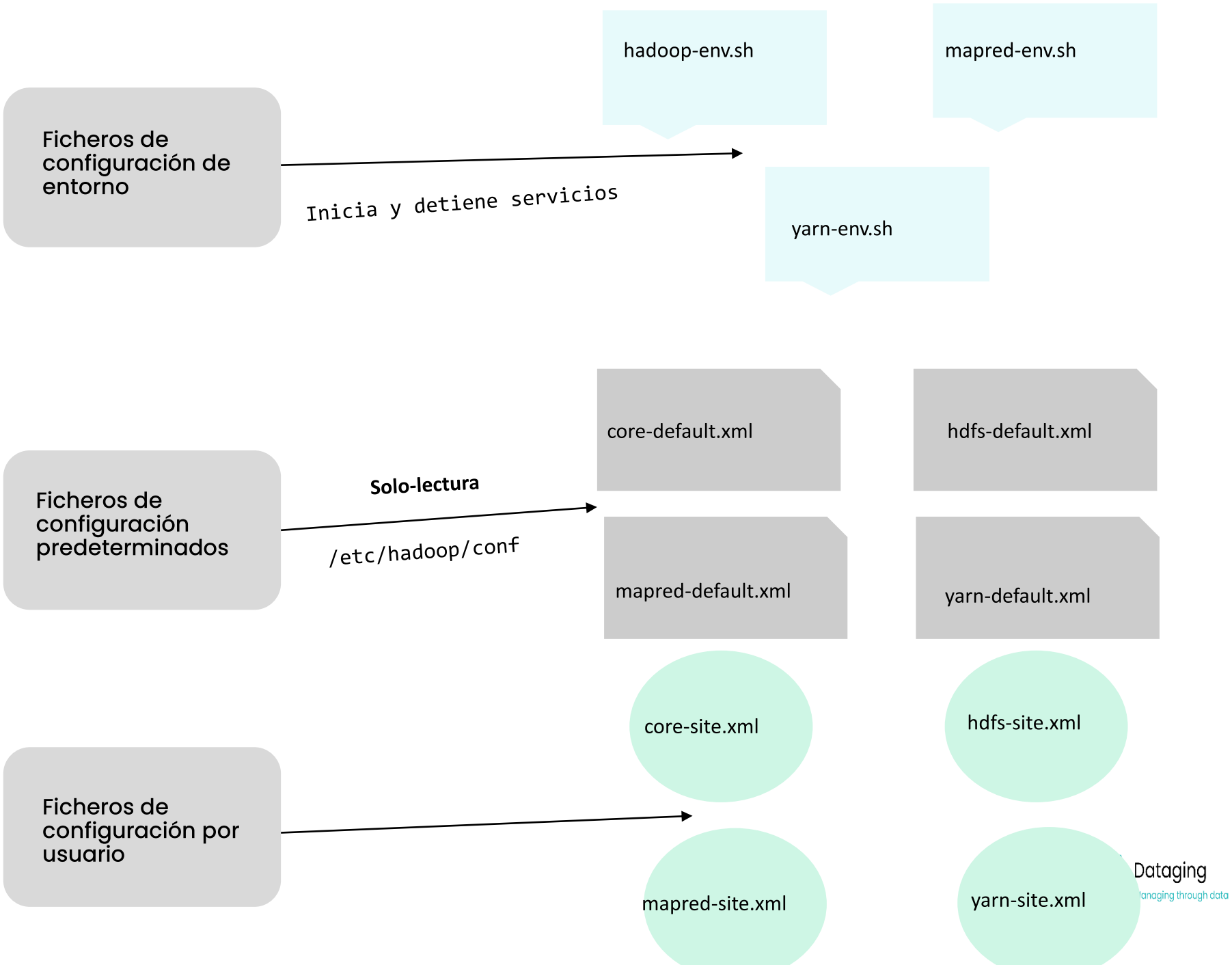
Con TEZ...



Dataging
Managing through data

Agenda

- Framework de Hadoop 2.0
- YARN
- **Ficheros de Configuración**


Ficheros de configuración de entorno
Inicia y detiene servicios
hadoop-env.sh
mapred-env.sh
yarn-env.sh
Ficheros de configuración predeterminados
Solo-lectura
/etc/hadoop/conf
core-default.xml
hdfs-default.xml
mapred-default.xml
yarn-default.xml
Ficheros de configuración por usuario
core-site.xml
hdfs-site.xml
mapred-site.xml
yarn-site.xml
Dataging

Ficheros de configuración

Nombre Fichero	Formato	Propósito
core-site.xml	Hadoop configuration XML	Configuraciones base de Hadoop, HDFS, YARN, MapReduce
hdfs-site.xml	Hadoop configuration XML	Configuraciones específicas HDFS (NameNode y DataNode)
yarn-site.xml	Hadoop configuration XML	Configuraciones YARN
Mapred-site.xml	Hadoop configuration XML	MapReduce
Hadoop-evn.sh	Bash script	Variables de entorno
log4j.properties	Java properties	Configuraciones del log de sistema
Hadoop-metrics2.properties	Java properties	Configuración de métricas

Precedencia de Configuración

La configuración actual para cualquier trabajo en ejecución en un cluster se deriva de una combinación de orígenes, incluyendo configuración predeterminada, configuración del cluster o del nodo y la configuración del trabajo

Configuración predeterminada

hadoop-common.jar

hadoop-hdfs.jar

Hadoop-mapreduce-client-core.jar

Hadoop-yarn-common.jar

JAR files contain (example)

Core-default.xml

Hdfs-default.xml

Mapred-default.xml

Yarn-default.xml

Hereda, extiende, sobrescribe

Configuración cluster

Core-site.xml

Hdfs-site.xml

Mapred-site.xml

Yarn-site.xml

Hereda, extiende sobrescribe

Conf. trabajo

#yarn jar –D prop=value …

Propiedades **final**

Dataging
Managing through data