
Map-Reduce

Problema

- Necesidad de procesar grandes volúmenes de datos.
-

Map-Reduce

- Procesamiento distribuido de datos utilizando un **cluster**.

Cluster

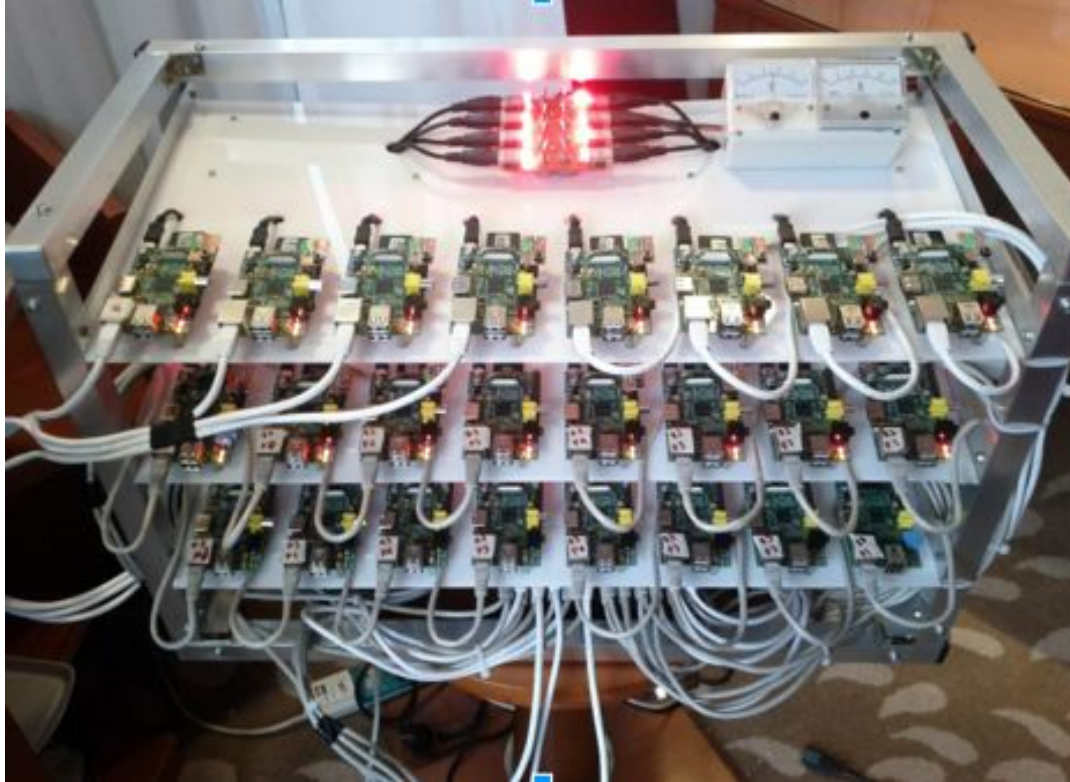
- Conjunto de computadoras que trabajan juntas y pueden ser vistas como un sistema único.
-

Cluster



Source:
https://en.wikipedia.org/wiki/Computer_cluster#/media/File:MEGWARE.CLIC.jpg

Cluster



Source:
<http://rachaeledwards.com/focus/raspberry-pi-cluster.html>

Cluster



Source: <https://cloud.google.com/tpu/>

Almacenamiento Distribuido

- FileSystem Distribuido
 - Encargado de gestionar cómo y dónde guardar la información en una computadora, y cómo poder consultarla.
-

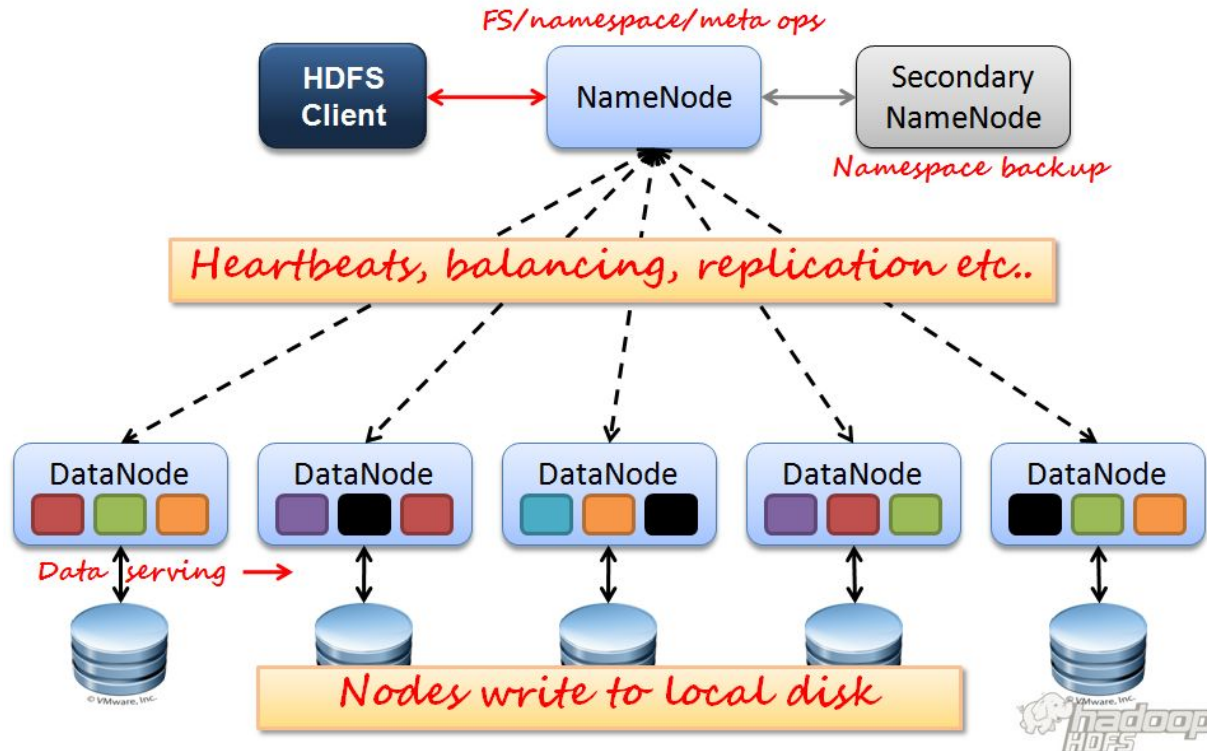
Almacenamiento Distribuido

- Almacenar grandes volúmenes de datos en múltiples equipos.
 - Replicación de datos.
 - Tolerante a fallos.
 - Alta disponibilidad.
 - Relativo bajo costo.
-

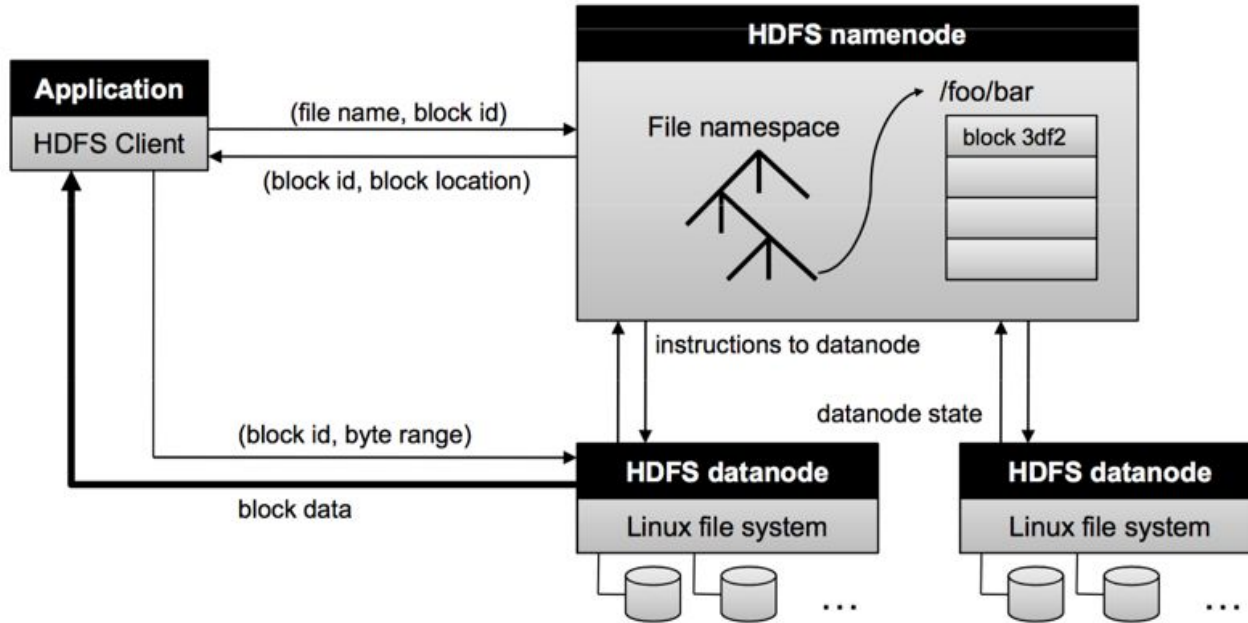
Almacenamiento Distribuido

- Implementaciones:
 - GFS (Google)
 - HDFS (Apache Hadoop)
 - CEPH
 - S3 (Amazon)
-

Almacenamiento Distribuido (HDFS)



Almacenamiento Distribuido (HDFS)



Map-Reduce

[2004] Jeffrey Dean and Sanjay Ghemawat, Google Inc.

- Modelo de programación para procesar grandes conjuntos de datos.
 - Necesidad de procesar grandes volúmenes de datos **de forma escalable.**
-

Map-Reduce

- El usuario especifica una función **map** que procesa un par clave/valor para generar un conjunto intermedio de pares clave/valor.
 - Se debe especificar también una función **reduce** que combina todos los valores asociados a la misma clave.
-

Map

- Transforma nuestros datos.
 - Debe ser aplicada a cada dato de nuestro set.
 - Puede ser paralelizada y distribuirse entre las distintas máquinas de un cluster.
-

Map

- Algunas diferencias dependientes de la implementación:
 - **Hadoop:** $\text{Map}(k,v) \rightarrow [k_2,v_2]$
 - **Spark:** $\text{Map}(r) \rightarrow [r']$
-

Reduce

- Combina los resultados del map.
 - Es necesario procesar los datos de todas las máquinas del cluster.
 - Reduce locales en paralelo y reduce entre máquinas mediante etapa de shuffle & sort.
-

Reduce

- Algunas diferencias funcionales en base a la implementación.
-

Reduce (Hadoop)

`ReduceByKey((k,v),f) -> [(k,v)]`

- El sistema agrupa todos los registros para los cuales la clave es la misma.
 - Requiere que todos los registros de igual clave estén en el mismo equipo que ejecute el reduce: Shuffle & Sort.
-

Reduce (Spark)

- ReduceByKey: combina los elementos para una misma clave.
 - Reduce: da un único resultado para todo el set de datos.
-

Reduce (Spark)

- La función reduce toma dos valores para dar como resultado la combinación de ambos.
 - El resultado de un reduce entre dos registros es el input del siguiente reduce.
-

Reduce (Spark)

(fecha, cliente, importe)

- Map(a):
 - a[importe]
- Reduce(a,b):
 - $a + b$

Reduce (Spark)

- Operaciones **conmutativas y asociativas** de modo de poder ejecutarse distribuidas.
-

Shuffle & Sort

- Mueve la salida de un proceso map a un cierto equipo de tal forma que un reducer pueda procesar sus registros.
 - Fase más costosa del proceso Map-Reduce.
 - Optimización para reducir la cantidad de datos a transmitir por la red.
-

Framework Map-Reduce

- Abstracción para el procesamiento distribuido.
 - API de distintos niveles para ejecutar Map-Reduce.
 - Manejo del framework de los trabajos a ejecutar en cada máquina del cluster.
-

Framework Map-Reduce

- División de los trabajos entre los nodos del cluster.
 - Manejo de errores en los trabajos.
-

Framework Map-Reduce

- El framework nos soluciona varios de estos inconvenientes.
 - Nos da opciones de cómo manejarlos.
-