

Material de apoyo sobre

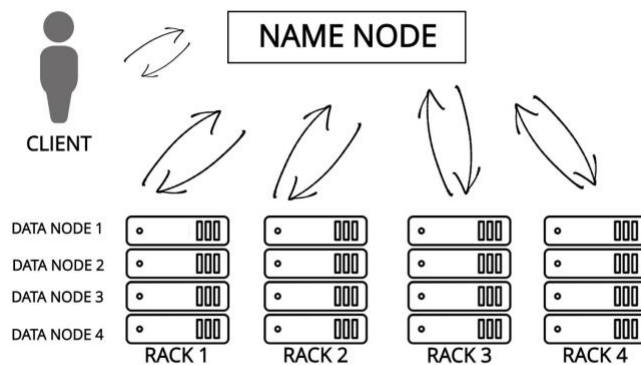
# Sistema de archivos distribuidos Hadoop

De Anne Pernille Wulff Wold y Mari Elida Tuhus



## El plan para la presentación:

1. Introducción: ¿qué es el sistema de ficheros distribuido de Hadoop?
2. Descripción de la arquitectura y los diversos componentes
3. Replicación de datos
4. Ejemplos
5. Tolerancia a fallos
6. Cuándo usar HDFS y cuándo no usar el sistema



## Breve descripción sobre los diversos componentes del sistema de archivos:

### 1. Namenode:

Actúa como el máster o el administrador en el clúster de HDFS. Su tarea principal es la distribución de archivos, para lo que realiza acciones relacionadas al equilibrio de carga y el seguimiento de la localización de estos archivos en el clúster.

### 2. Datanode:

En la arquitectura de HDFS, los nodos de datos almacenan los datos reales, por lo que cada nodo se conoce también como nodo esclavo. Los nodos de datos son responsables de atender todas las solicitudes de lectura y escritura para los clientes.

### 3. Rack:

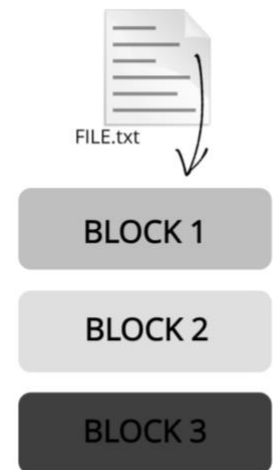
Un rack es una colección de 30 o 40 nodos que se almacenan físicamente muy cerca y están todos conectados con el mismo interruptor de red.

### 4. Client:

El cliente en el sistema de archivos Hadoop se refiere a la interfaz utilizada para comunicarse con el sistema de archivos. Hay diferentes tipos de clientes disponibles para realizar diferentes tareas.

### 5. Block:

Un archivo se divide en uno o más bloques, los que suelen ser muy grandes y, generalmente, tienen un tamaño predeterminado de 64 MB o 128 MB.



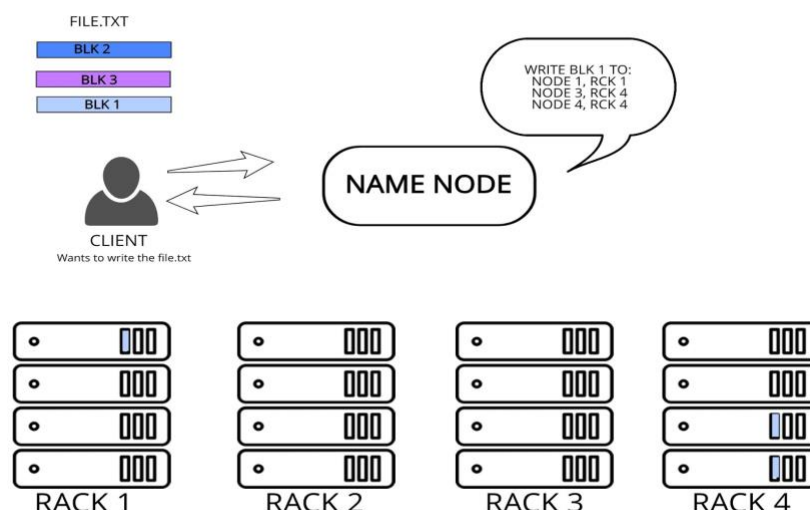
### Replication de datos:

Los datos escritos por un cliente a HDFS se dividen en uno o más bloques dependiendo del tamaño del archivo. Estos bloques se replican más tarde y se almacenan en diferentes nodos de datos para proporcionar tolerancia a fallas.

El factor de replicación predeterminado es 3, pero este factor es nuevamente configurable. La ubicación de las réplicas es fundamental para garantizar una alta fiabilidad y rendimiento de HDFS.

### Tolerancia a fallos

El objetivo principal de HDFS es almacenar datos de manera confiable incluso en presencia de fallas. Dado que los datos se almacenan en hardware básico, habrá fallas. Los tres tipos fallas de comunes son las fallas de los nodos de datos, las fallas en las particiones de red y las fallas de Name Node.



**1. Fallas de los nodos de datos:**

Un nodo de datos envía un latido de corazón frecuente a el Name Node para decirle que el nodo de datos está activo y funcionando. El Namenode interpreta la ausencia de latidos como una muerte del Data Node. Esto significa que no se asignará operaciones IO a ese nodo, y el contenido del Datanode está programado para ser copiado usando una de las otras réplicas a un otro nodo de datos.

**2. Partición de red**

Una partición de red puede hacer que un subconjunto de nodos pierda la conexión con el namenode y prohíbe que su señal de latido llegue al namenode.

**3. Fallas del Name Node**

El Name Node es el más importante, porque es una falla de punto único. Este nodo puede ejecutar un nodo de CheckPoint o Backup. Los nodos de CheckPoint y Backup frecuentemente fusionan el *Namespace Image* y el *Edit log* antes de devolverlo como un Checkpoint al Name Node.

En Hadoop 2.0 lanzaron el HDFS HA, *alta disponibilidad*. En esta versión hay dos Name Nodes, un activo y otro en modo de espera. El nodo activo actúa como un Name Node, y el otro siempre será actualizado. De esta manera, el nodo en modo de espera siempre está listo si hay problemas.

**Apuntes:**