Seminario de Estadística I
Introducción a la Ciencia de Datos y Machine Learning

HDFS

Universidad Nacional Autónoma de México
Facultad de Ciencias

Contenido

Sobre Hadoop

- Hadoop es framework que permite procesamiento distribuido de grandes cantidades de datos entre clusteres de computadoras.
- Esta diseñado para escalarse de un único servidor a miles de máquinas.
- Esta diseñado para detectar y manejar fallos, ofreciendo alta disponibilidad.

Módulos

Módulos de Hadoop

- Hadoop Common
- Hadoop Distributed File System (HDFS)
- Hadoop MapReduce:
- Hadoop Ozone:
- Hadoop Submarine

Modos

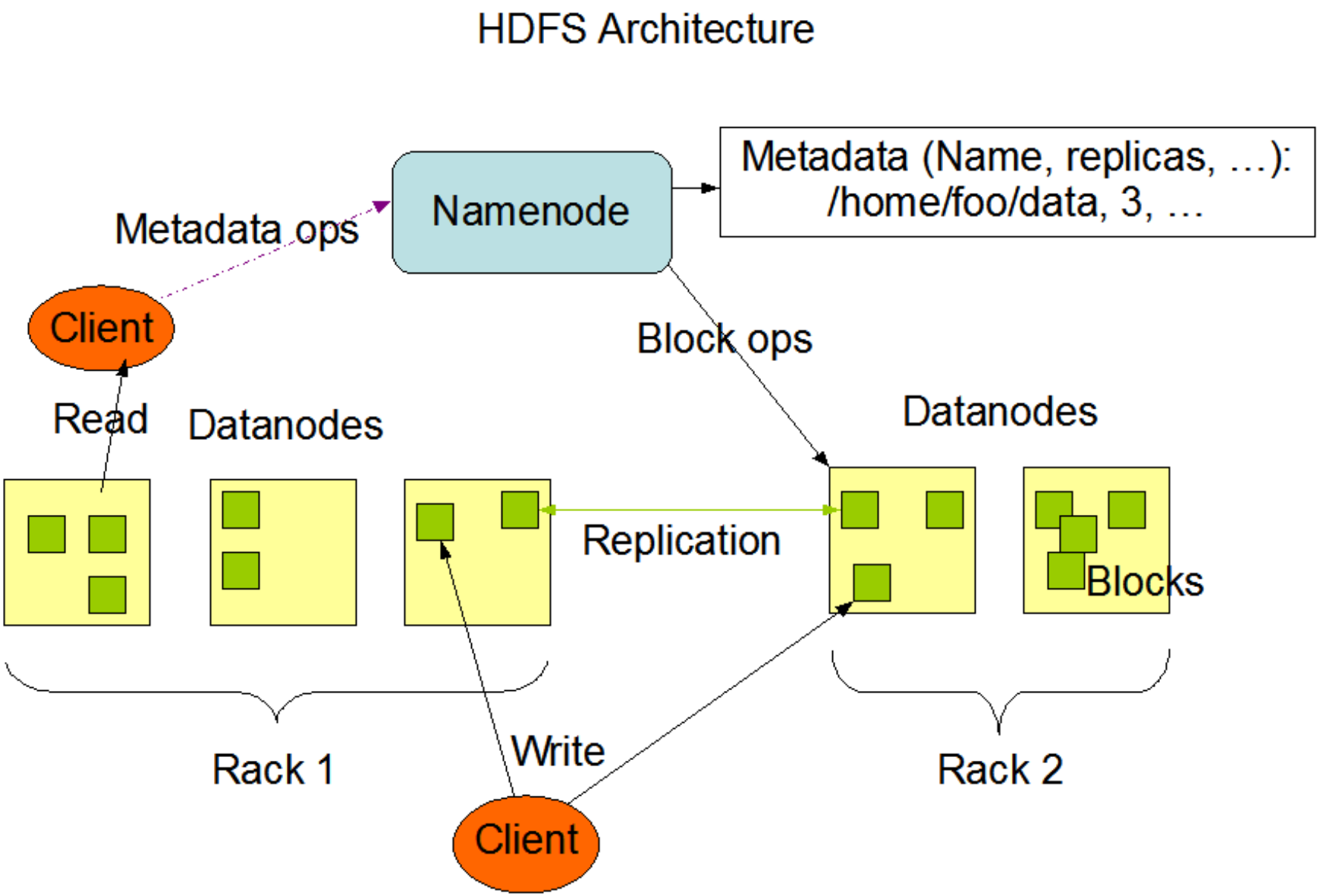
Modo Standalone, Pseudo-Distribuidos)

- **Modo Standalone** Modo no distribuido en el que hadoop corre como un único proceso Java.
- **Modo Pseudo-Distribuido** Hadoop corre en un único nodo donde cada demonio Hadoop corre en un proceso Java separado.

Modos

Introducción a HDFS

- El Hadoop Distributed File System (HDFS), es un sistema de ficheros distribuido altamente tolerante a fallos y esta diseñado para implementarse en software de bajo costo.
- Proporciona acceso de alto rendimiento a datos de la aplicación y adecuado para grandes volúmenes de datos.


HDFS Architecture
Namenode
Metadata (Name, replicas, …):
/home/foo/data, 3, …
Metadata ops
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Rack 2
Client

DataNode y NameNode

DataNode y NameNode

HDFS tiene una arquitectura maestro-esclavo. Un clúster HDFS consta de un único NameNode, un servidor maestro que administra y regula el acceso a los archivos por parte de los clientes. Además, hay varios DataNodes, generalmente uno por nodo, que administran el almacenamiento. Internamente, un archivo se divide en uno o más bloques (128MB) y estos bloques se almacenan en un conjunto de DataNodes, cabe mencionar que los bloques de información no solamente se dividen también se replican por default 3 veces. El NameNode ejecuta operaciones como abrir, cerrar y renombrar archivos y directorios. También determina la asignación de bloques a DataNodes. Los DataNodes son responsables de atender las solicitudes de lectura y escritura de los clientes del sistema de archivos.

Comunicación

DataNode y NameNode

Todos los protocolos de comunicación de HDFS están construidos sobre TCP/IP. Un cliente establece comunicación a un puerto TCP configurable del NameNode y este se encarga de establecer comunicación con los DataNodes.

Fallo en disco, latidos y Re-Replicación

Cada DataNode envía un mensaje latido (Heartbeat) al namenode periódicamente. El namenode en caso de no detectar latidos por un periodo de 10min por default declara muerto al datanode e inicia un proceso de replicación para evitar que el numero de replicas caiga por debajo de su valor especificado. La necesidad de volver a replicarse puede surgir debido a muchas razones: un DataNode puede dejar de estar disponible, una réplica corrupta, un disco duro en un DataNode puede fallar.

HDFS web-ui

Para interactuar con HDFS lo primero es "invocar" a los demonios NameNode, DataNode y el SecundaryNamenode con `start-dfs.sh` Verificamos que estén corriendo escribiendo jps en la terminal o bien con la interfaz web en nuestro navegador con la dirección `http://localhost:9870`, recordemos que 9870 es el puerto por default en hadoop 3 para ver el NameNode. Una vez lanzado hdfs podemos utilizar linea de comando o un cliente hdfs en python para realizar acciones sobre el sistema de ficheros. A continuación se mostraran los comandos básicos de interacción.

Algunos comandos básicos

Comandos HDFS

- **hdfs dfs -mkdir path**
  Crea un directorio en la ruta indicada
- **hdfs dfs -ls path**
  Lista el contenido dentro del directorio indicado en la ruta
- **hdfs dfs -du path**
  Muestra el uso en disco
- **hdfs dfs -mv pathorigen pathdestino**
  Mueve un directorio indicado por el path origen al pathdestino en hdfs.
- **hdfs dfs -rmr path**
  Elimina de manera recursiva, es decir elimina directorios y sus subcarpetas.
- **hdfs dfs -put pathlocal pathenhdfs**
  Subir archivos del path local a hdfs
- **hdfs dfs -get pathhdfs destino**
  Obtener archivos de hdfs a local