HDFS Introduction

2009-01-05
Wang Di

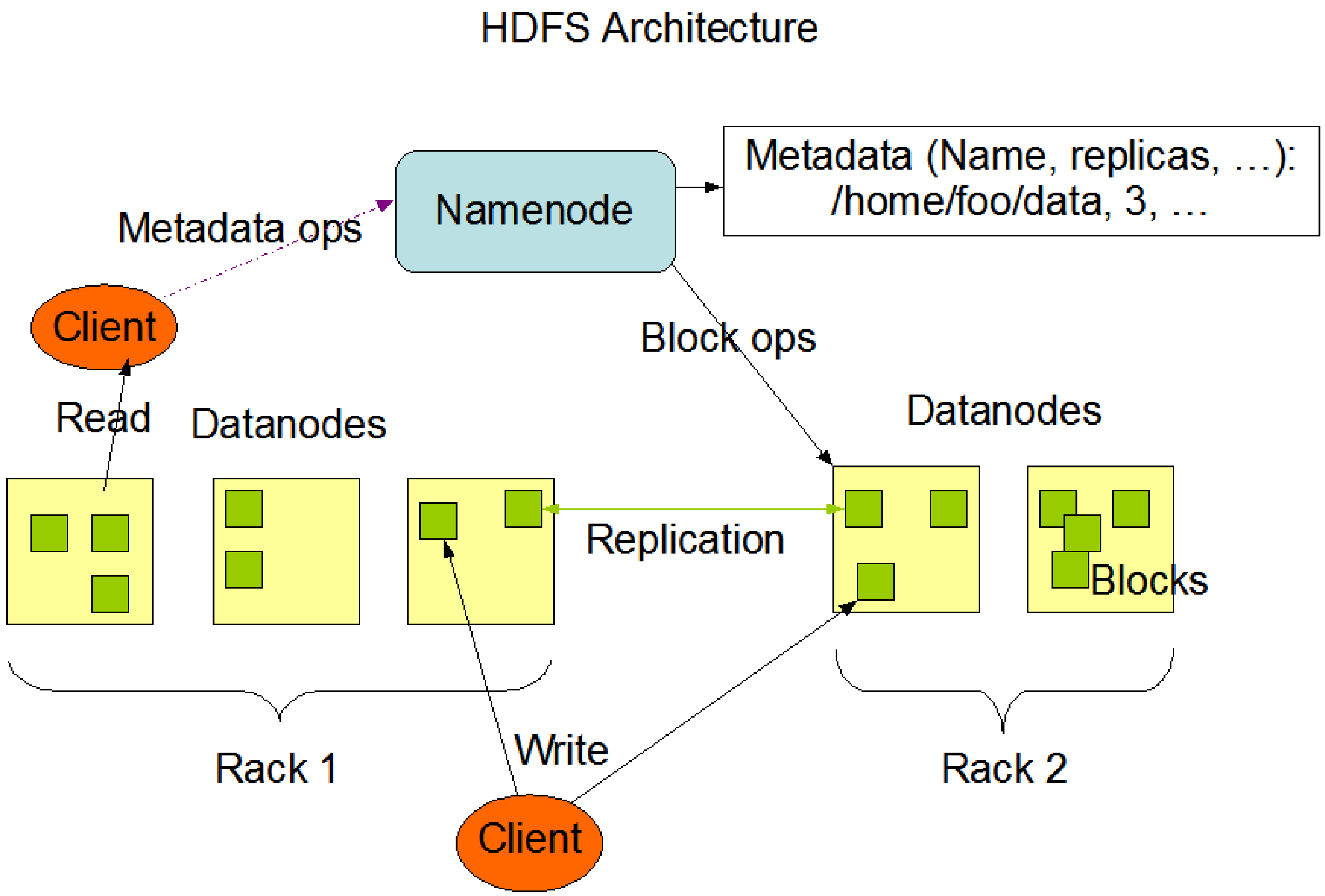
Introduction

- What is HDFS
  - > Hadoop and HDFS
  - > HDFS architecture
  - > Several operations for HDFS
  - > HDFS vs Lustre
- Lustre (Hadoop style) replication

Hadoop and HDFS

- Hadoop
  - Hadoop is composed of Map/reducer frame + HDFS, and it is part of search engine project (nutch).
  - When it works, Map/reduce frame will allocate the job to the node near the file jobs needs, according to the information HDFS provided.

HDFS Architecture



HDFS Write

- Write process
  - HDFS client caches the file data into a temporary local file.
  - When the local file accumulates data worth over one HDFS block size (64M), the client will contact to the namenode.
  - Namenode inserts the file name into the file system hierarchy and allocates a data block for it and reply to the client.
  - The client flushes the block of data from the local temporary file to the specified DataNode.
  - When a file is closed, the remaining un-flushed data in the temporary local file is transferred to the DataNode.

HDFS replication

- Replication process
  - Suppose the HDFS file has a replication factor of three. When the local file accumulates a full block of user data, the client retrieves a list of DataNodes from the NameNode.
  - This list contains the DataNodes that will host a replica of that block. The client then flushes the data block to the first DataNode.
  - The first DataNode starts receiving the data in small portions (4 KB), writes to its local repository and also transfers to the second DataNode in the list.
  - The second DataNode works in the similar way, write to its repository and send that to the third DataNode.
  - Finally, the third DataNode writes the data to its local repository.

HDFS Failure recovery

- Failures
  - Datanode failures
    - Each DataNode sends a Heartbeat message to the NameNode periodically.
    - The NameNode marks DataNodes without recent Heartbeats as dead and does not forward any new IO requests to them.
    - DataNode death may cause the replication factor of some blocks to fall below their specified value. And the namenode will initiate the replication process then.
  - Metanode failures
    - Single node failure. If the NameNode machine fails, manual intervention is necessary.

HDFS

- HDFS Features
  - Write_once_and_read_many. No POSIX compatible.
  - Big Files (64M block_size)
  - Data replication.
  - Hadoop has been demonstrated on clusters with 2000 nodes. The current design target is 10,000 node clusters.
  - Implemented by Java.
  - Storage node == compute node

HDFS vs LUSTRE

- Lustre
  - POSIX compatible.
  - performance and scalability.
  - Storage node != compute node.

Replication plan

- Hadoop style Replication on Lustre
  - > Replication
    - – The data will be replicated between pools.
    - – The user could set replication factors on the stripe.
    - – MDS will control the replication based on the stripe information (replication factors + change logs).
  - > Choose storage by location
    - – MDS choose storage for client
    - – Client choose itself
  - > Maintain the replication factor

Current Status

- Three interns from SCUT
- Investigate HDFS architecture and code.
- Tried to wrap liblustre with Java interface.
- Comparing performance between HDFS and Lustre.

Q&A !