BİL401/BİL501
Distributed Data Processing and Analysis
«BigData»

Hadoop Architecture and HDFS

Spring 2015
Erdogan Dogdu
TOBB University of Economics and Technology
Department of Computer Engineering

Hadoop architecture

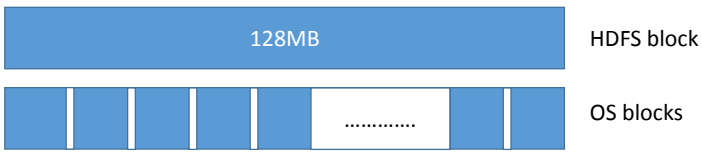
- Two main components
  - Disftributed file system (HDFS)
  - MapReduce engine

HDFS (Hadoop Distributed File System)

- Runs on top of existing file system
- Designed to handle very large files with streaming data access patterns
- Uses blocks to store a file or parts of a file

HDFS file blocks

- 64 MB (default), 128 MB (recommended) – compare to 4KB oin Unix
- Behind the scenes, 1 HDFS block is supported by multiple OS blocks



HDFS file blocks - Advantages

- Fixed size – easy to calculate how many fit on a disk
- A file can be larger than any single disk in the network
- If a file or a chunk of the file is smaller than the block size, only needed space is used. Eg. 420MB file is split as:
  - 128MB + 128MB + 128MB + 36MB
- Fits well with the replication to provide fault tolerance and availability

HDFS - Replication

- Blocks are replicated to multiple nodes
- Allows for node failure without data loss

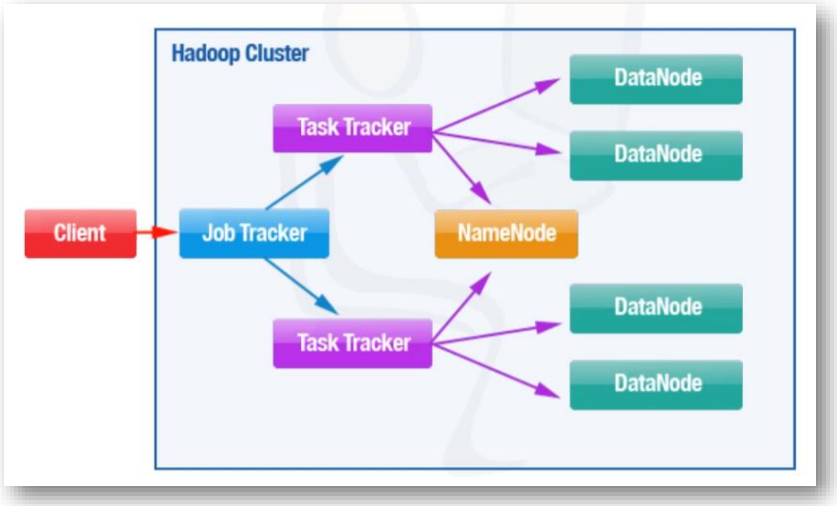
MapReduce engine

- Technology from Google
- MapReduce program: map and reduce functions
- MapReduce job: tasks that run in parallel

Hadoop nodes

- HDFS nodes
  - **NameNode (1)**
  - **DataNode (n)**
- MapReduce nodes
  - **JobTracker (1)**
  - **TaskTracker (n)**
- Other nodes as well (secondary name node, check point node, backup node)

Communication



NameNode

- Only one per Hadoop cluster
- Manages the file system namespace and metadata
- Single point of failure
  - But mitigated by writing the state to multiple filesystems
  - Don’t use inexpensive commodity hardware for this node
  - Large memory requirements

DataNode

- Many per Hadoop cluster
- Manages blocks with data and serves them to clients
- Periodically reports to name node the list of blocks it stores
- Use inexpensive commodity hardware for this node

JobTracker

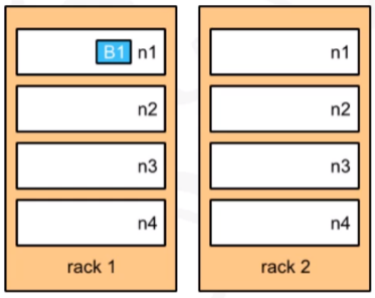
- One per Hadoop cluster
- Receives job requests submitted by client
- Schedules and monitors MapReduce jobs on task trackers

TaskTracker

- Many per Hadoop cluster
- Executes MapReduce operations

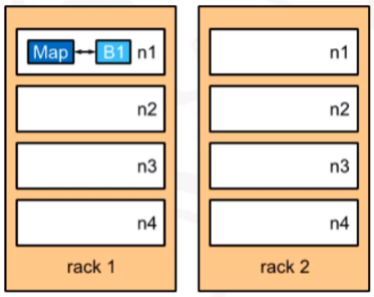
Topology awareness

- Where to process block B1?
- ***Data locality optimization***



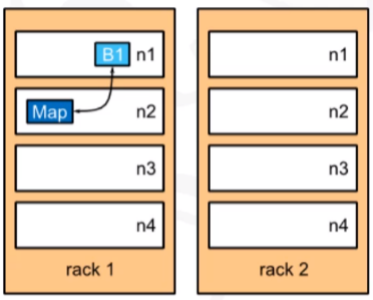
Topology awareness

- Where to process block B1?
- ***Data locality optimization***
- Best: On the same server



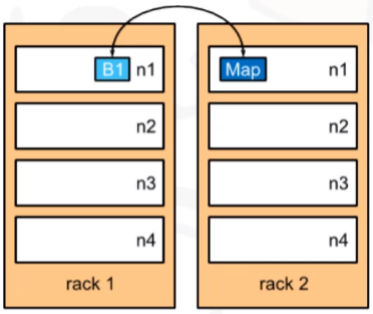
Topology awareness

- Where to process block B1?
- ***Data locality optimization***
- Best: On the same server
- Next to best: On the same rack



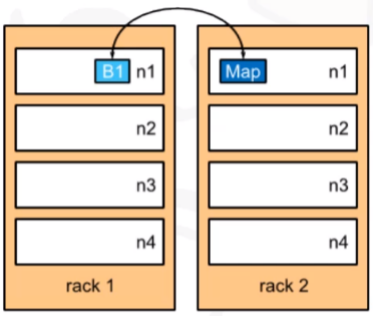
Topology awareness

- Where to process block B1?
- ***Data locality optimization***
- Best: On the same server
- Next to best: On the same rack
- Worst: On a different rack



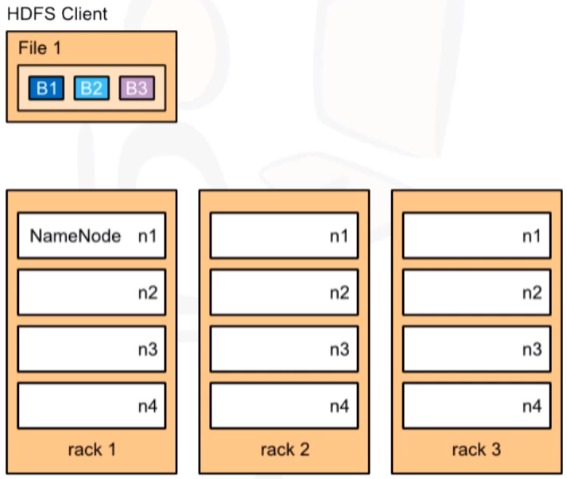
Topology awareness

- Where to process block B1?
- ***Data locality optimization***
- Best: On the same server
- Next to best: On the same rack
- Worst: On a different rack

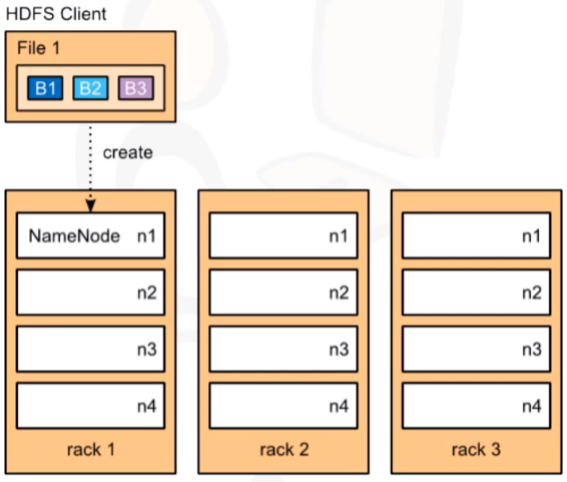


- *Bandwidth utilization decreases if data is on another server on the same rack or in another rack*

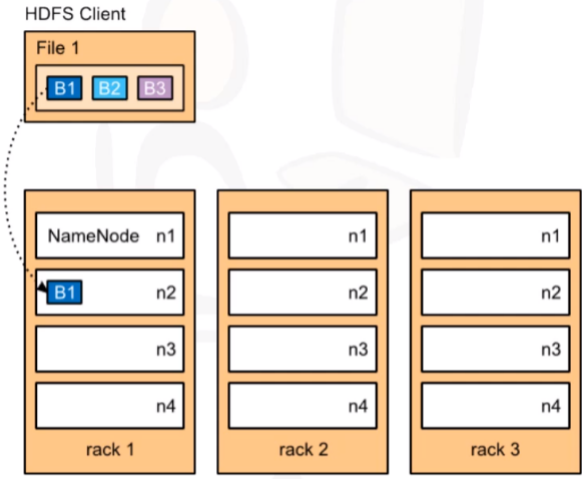
Writing a file to HDFS



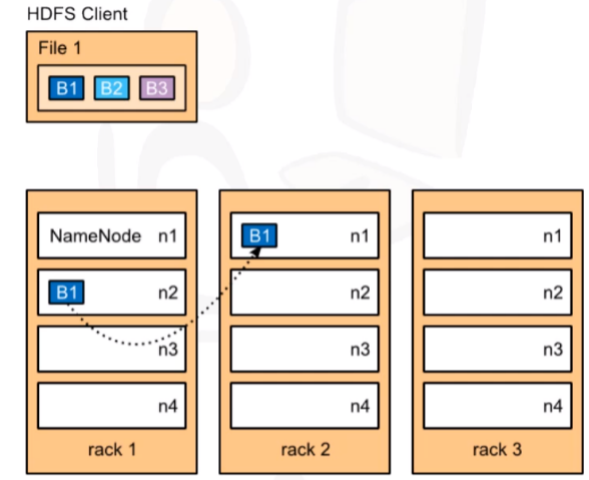
Writing a file to HDFS



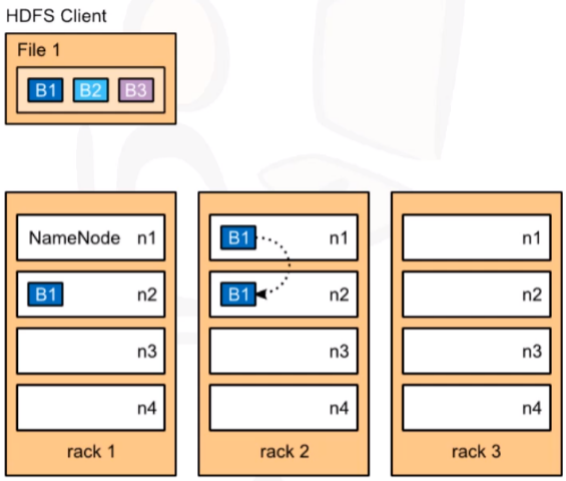
Writing a file to HDFS



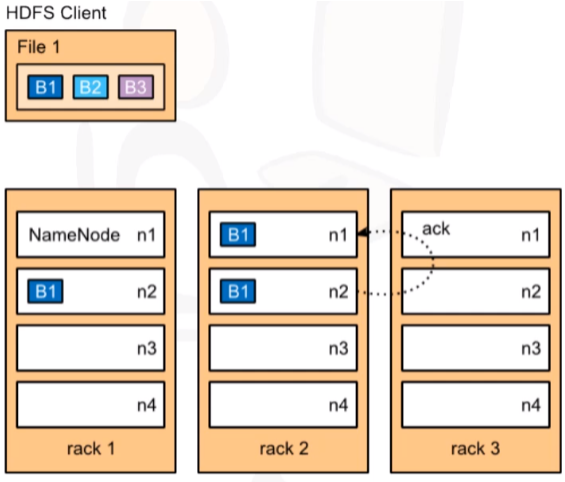
Writing a file to HDFS



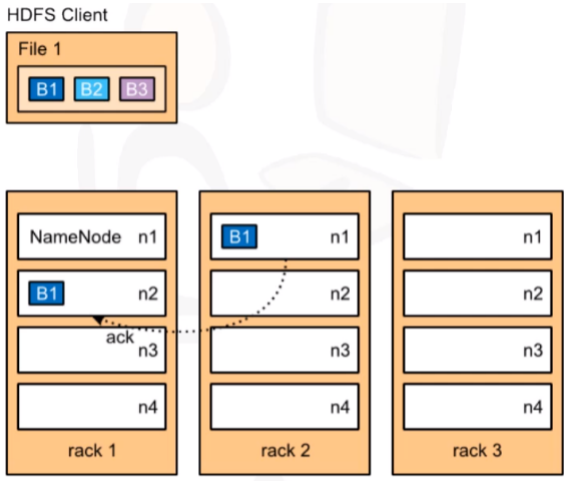
Writing a file to HDFS



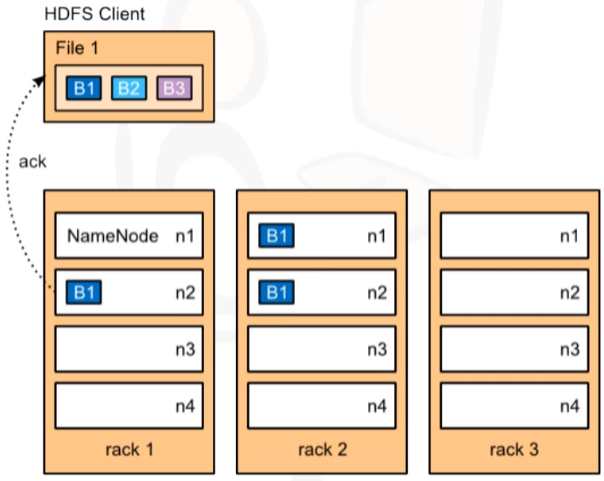
Writing a file to HDFS



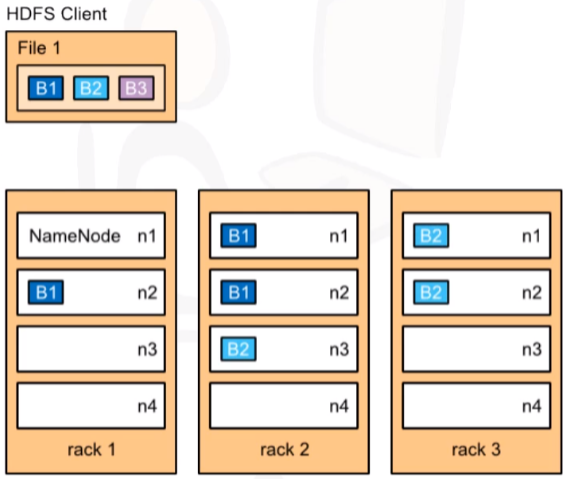
Writing a file to HDFS



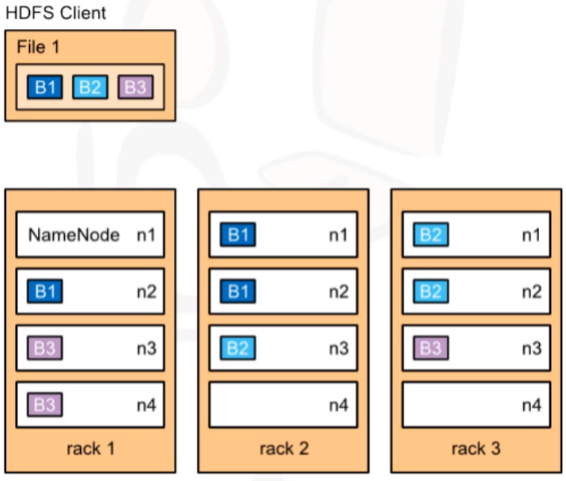
Writing a file to HDFS



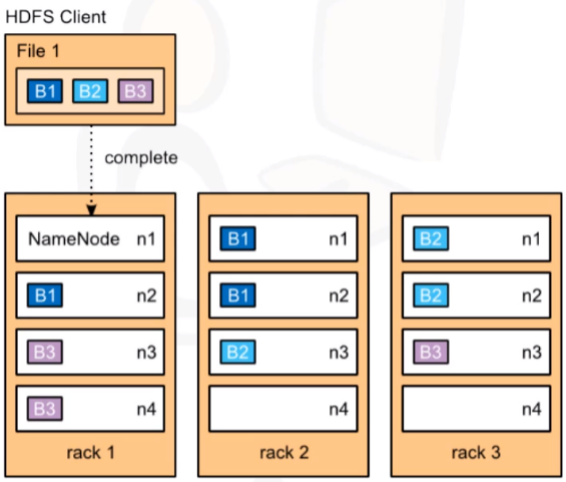
Writing a file to HDFS



Writing a file to HDFS



Writing a file to HDFS



Hadoop command line

```
> hadoop
Usage: hadoop [--config confdir] COMMAND
where COMMAND is one of:
  namenode -format     format the DFS filesystem
  secondarynamenode    run the DFS secondary namenode
  namenode             run the DFS namenode
  datanode             run a DFS datanode
  dfsadmin             run a DFS admin client
  mradmin              run a Map-Reduce admin client
  fsck                 run a DFS filesystem checking utility
  fs                   run a generic filesystem user client
  balancer             run a cluster balancing utility
  fetchdt              fetch a delegation token from the NameNode
  jobtracker           run the MapReduce job Tracker node
  pipes                run a Pipes job
  tasktracker          run a MapReduce task Tracker node
  historyserver        run job history servers as
                       a standalone daemon
  job                  manipulate MapReduce jobs
  queue                get information regarding JobQueues
  version              print the version
  jar <jar>            run a jar file
  distcp <srcurl> <desturl> copy file or directories
recursively
  archive -archiveName NAME -p <parent path> <src>* <dest>
create a hadoop archive
  classpath            prints the class path needed to get
the
                       Hadoop jar and the required
libraries
  daemonlog            get/set the log level for each
daemon
 or
  CLASSNAME            run the class named CLASSNAME
Most commands print help when invoked w/o parameters.
```

HDFS system shell (fs)

```
> hadoop fs
Usage: java FsShell
        [-ls <path>]
        [-lsr <path>]
        [-du <path>]
        [-dus <path>]
        [-count[-q] <path>]
        [-mv <src> <dst>]
        [-cp <src> <dst>]
        [-rm [-skipTrash] <path>]
        [-rmr [-skipTrash] <path>]
        [-expunge]
        [-put <localsrc> ... <dst>]
        [-copyFromLocal <localsrc> ... <dst>]
        [-moveFromLocal <localsrc> ... <dst>]
        [-get [-ignoreCrc] [-crc] <src> <localdst>]
        [-getmerge <src> <localdst> [addnl]]
        [-cat <src>]
        [-text <src>]
        [-copyToLocal [-ignoreCrc] [-crc] <src> <localdst>]
        [-moveToLocal [-crc] <src> <localdst>]
        [-mkdir <path>]
        [-setrep [-R] [-w] <rep> <path/file>]
        [-touchz <path>]
        [-test -[ezd] <path>]
        [-stat [format] <path>]
        [-tail [-f] <file>]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-chgrp [-R] GROUP PATH...]
        [-help [cmd]]
```


HDFS system shell (fs)

- **hadoop fs <args>**

- List the current directory in hdfs
  - **hadoop fs –ls .**

HDFS system shell (fs)

- fs shell commands take path URIs as args
  - scheme://authority/path
  - scheme://authority is optional
- Scheme
  - file: local file system
  - hdfs: HDFS file
- Örnek:
  - hadoop fs –copyFromLocal file://myfile.txt hdfs://localhost/hw1/myfile.txt
  - hadoop fs –copyFromLocal myfile.txt hw1/myfile.txt

HDFS system shell (fs)

- Many posix-like commands
  - cat, chgrp, chmod, cp, du, ls, mkdir, mv, rm, stat, tail
- HDFS specific commands
  - copyFromLocal, copyToLocal, get, getmerge, put, setrep

HDFS specific commands

- copyFromLocal / put
  - hadoop fs –copyFromLocal <localsrc> .. <dst>

  or
  - hadoop fs –put <localsrc> .. <dst>

- copyToLocal / get
  - hadoop fs –copyToLocal [-ignorecrc] [-crc] <dst> <localsrc>
  - hadoop fs –get [-ignorecrc] [-crc] <dst> <localsrc>

HDFS specific commands

- getmerge
  - Get all the files in the directories that match the source file pattern
  - Merge and sort them only one file on local fs
  - <src> is kept
- hadoop fs –getmerge <src> <localdst>

HDFS specific commands

- setrep
  - Set the replication level of a file
  - The –R flag requests a recursive change of replication level for an entire tree
  - If –w is specified, waits until new replication level is reached
- hadoop fs –setrep [-R] [-w] <rep> <path/file>