Anatomy of MapReduce

Sundharakumar KB

Department of Computer Science and Engineering
School of Engineering

Shiv Nadar University Chennai

SHIV NADAR UNIVERSITY CHENNAI

Anatomy of a job run

- There are 4 entities:
- **Client:** this submits the Mapreduce Job

- **The jobtracker:** coordinates the job run

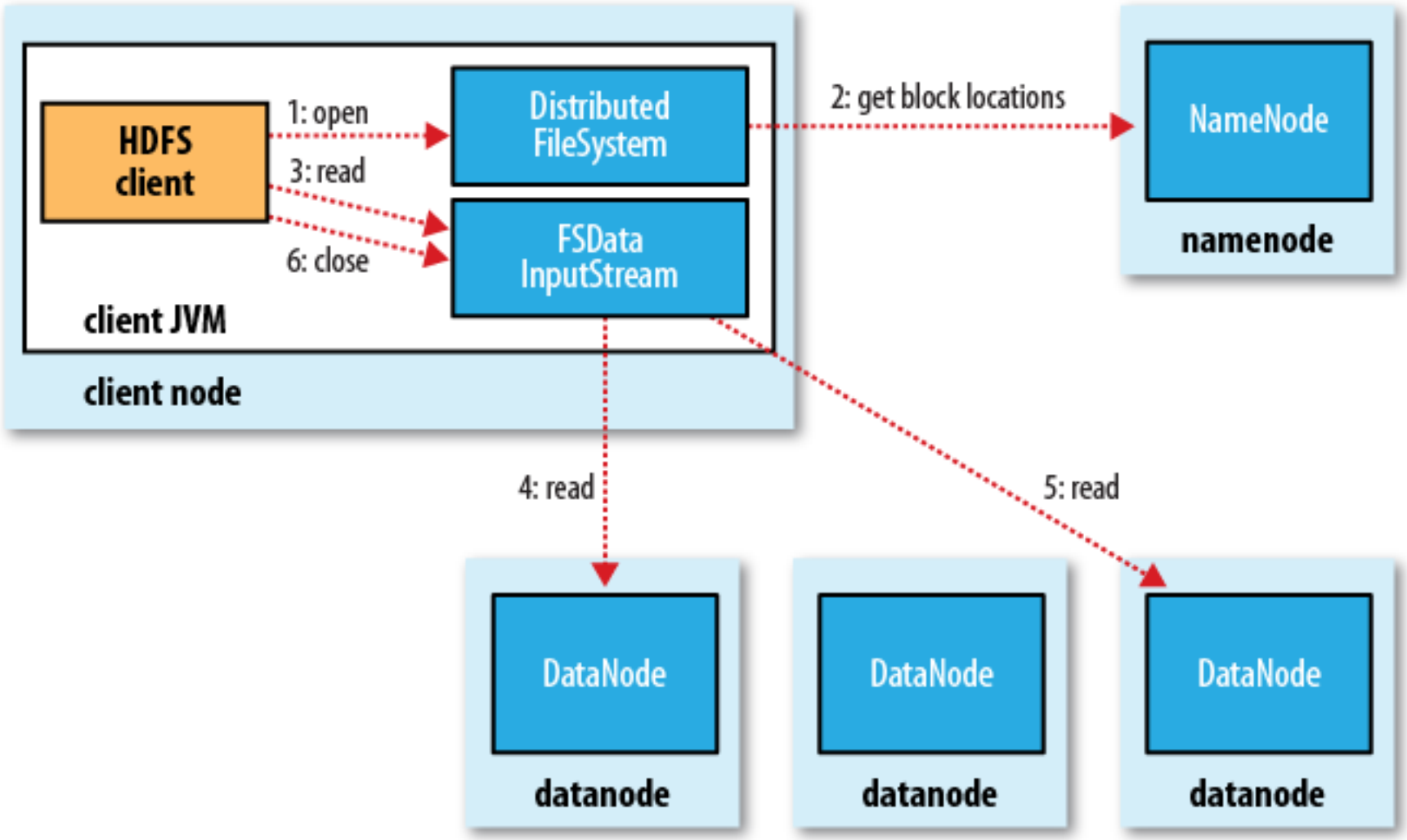
- **Tasktracker:** run the tasks that the job has been split into.

- **HDFS:** used for sharing the files between the entities.

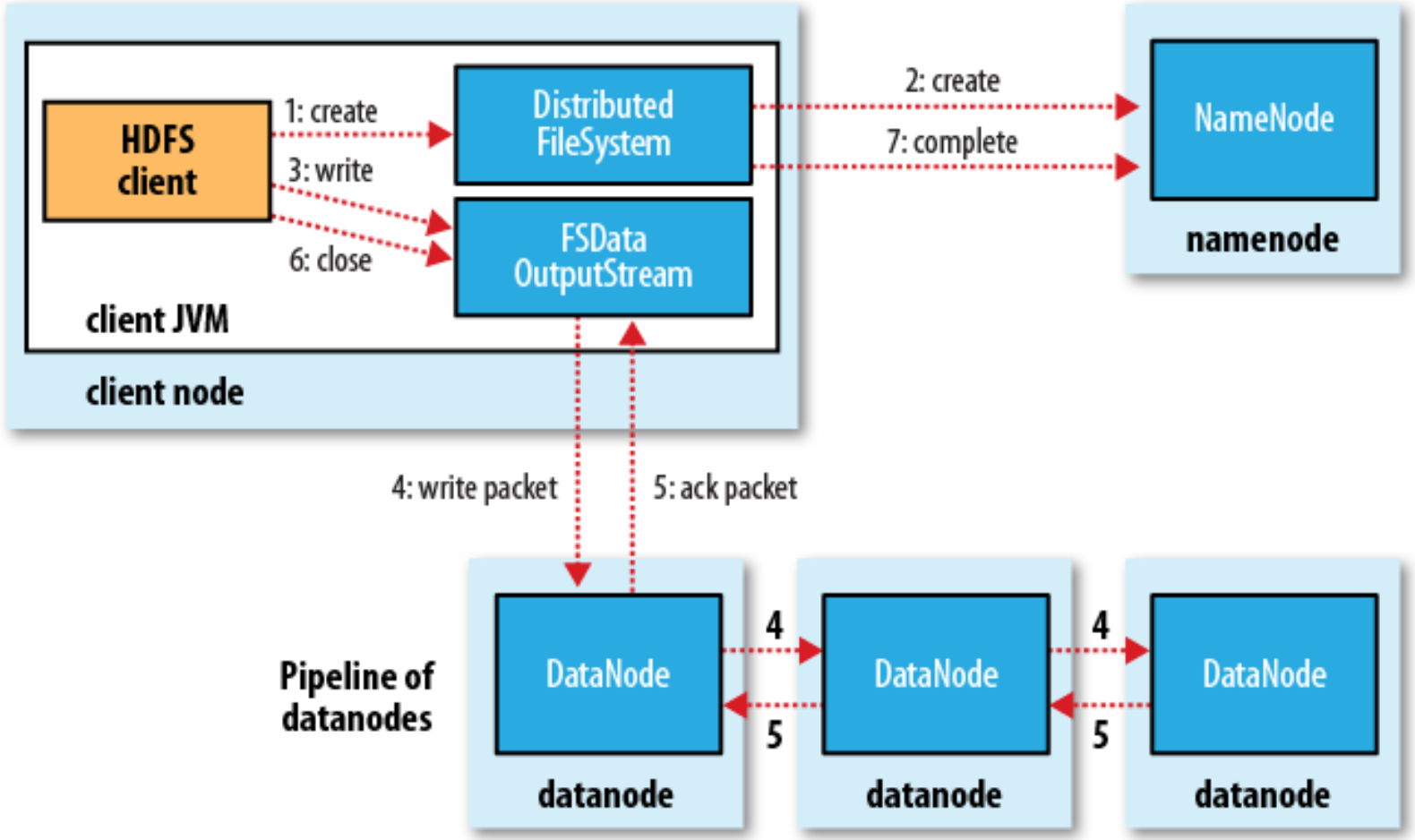
File read

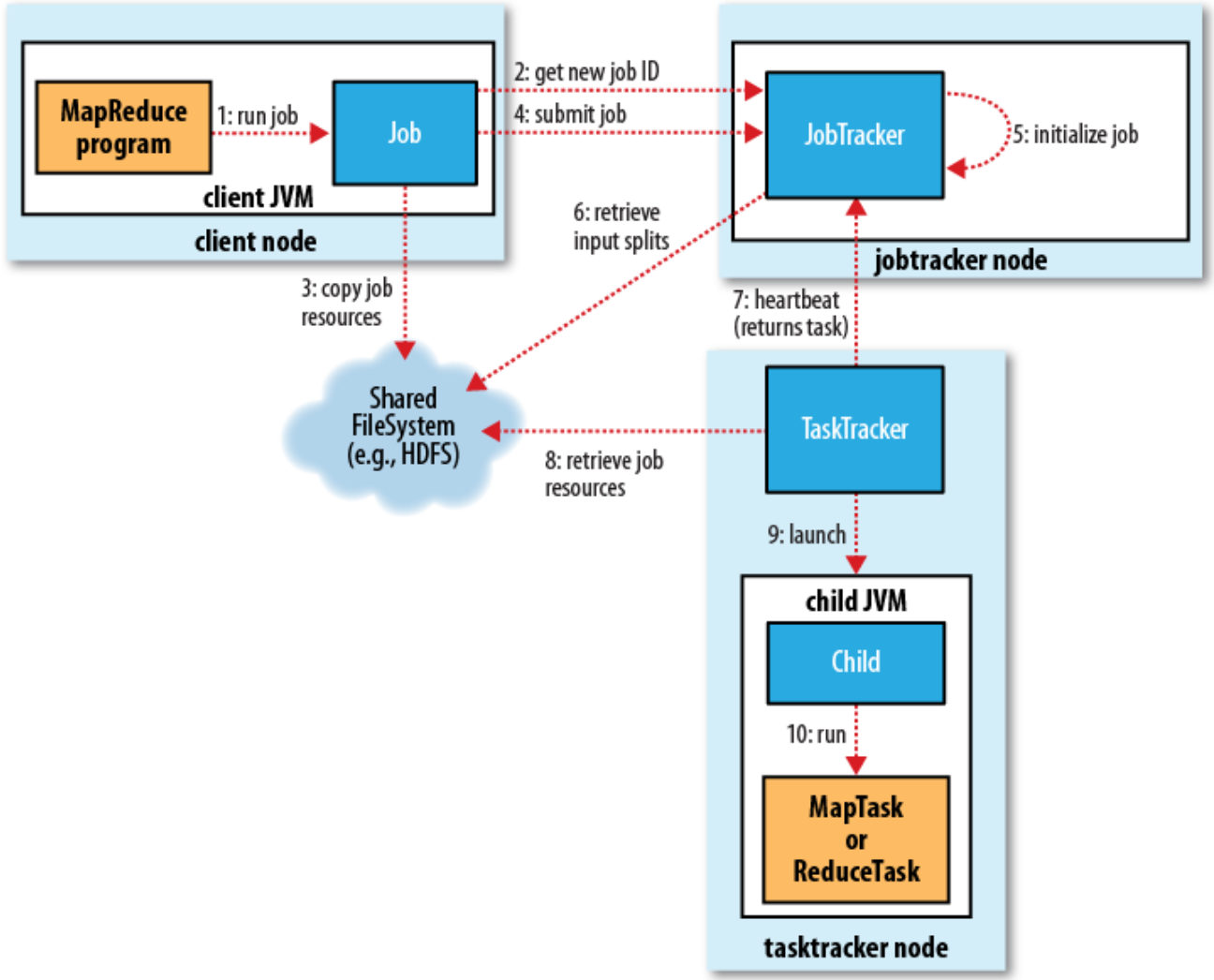
- To read a file from HDFS, the client needs to interact with the Namenode.
- Namenode provides the address of the datanodes (slaves) where the files are stored.
- Client will then interact with the datanodes to read the file.
- Authentication mechanism: namenode will provide a token to the client as well as datanodes.

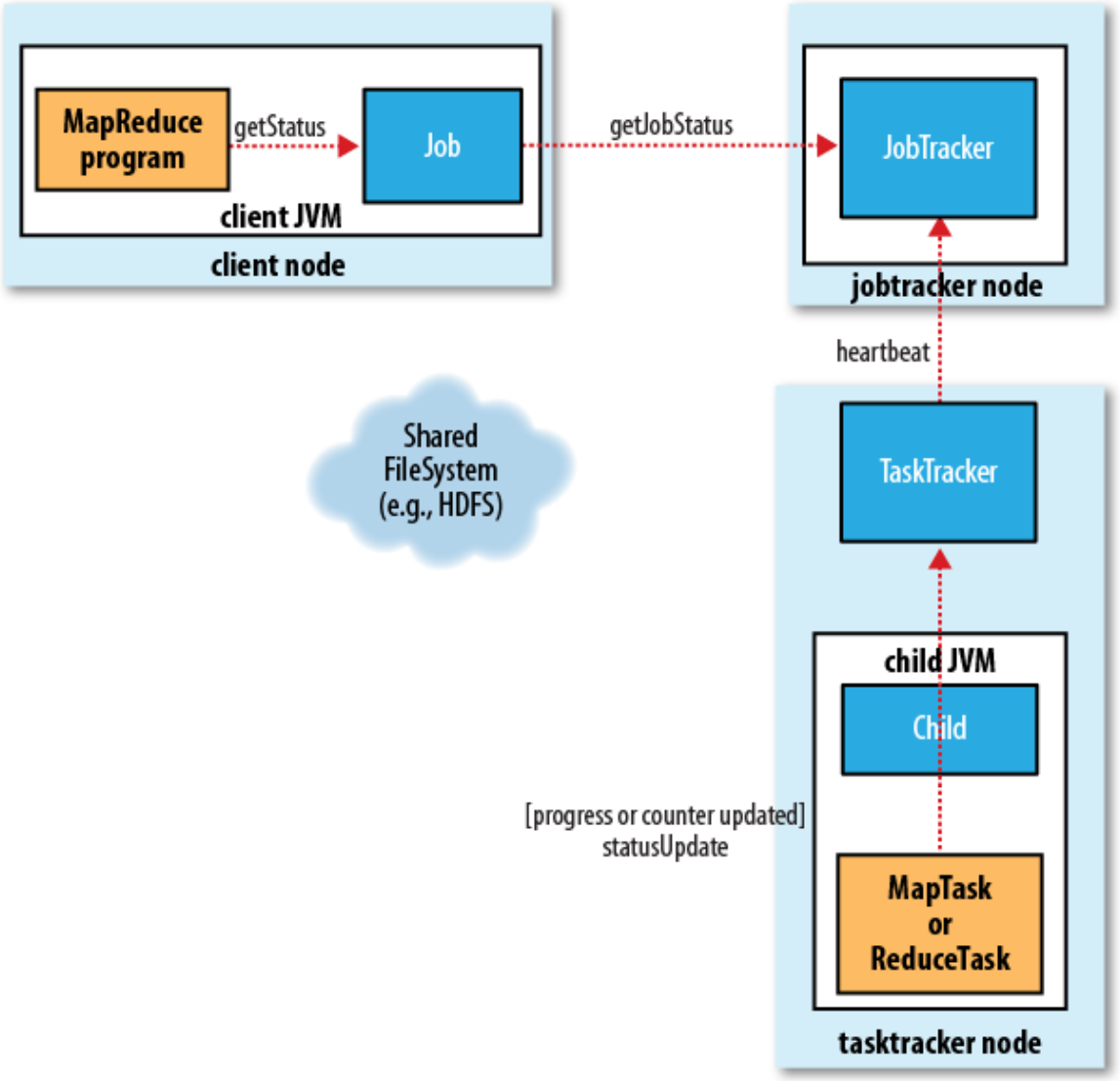
File read


HDFS client
1: open
3: read
6: close
Distributed FileSystem
FSData InputStream
client JVM
client node
2: get block locations
NameNode
namenode
4: read
5: read
DataNode
datanode
DataNode
datanode
DataNode
datanode

File write




MapReduce program
1: run job
Job
client JVM
client node
2: get new job ID
4: submit job
JobTracker
5: initialize job
jobtracker node
6: retrieve input splits
3: copy job resources
7: heartbeat (returns task)
Shared FileSystem (e.g., HDFS)
8: retrieve job resources
TaskTracker
9: launch
child JVM
Child
10: run
MapTask or ReduceTask
tasktracker node


MapReduce program
getStatus
Job
client JVM
client node
getJobStatus
JobTracker
jobtracker node
heartbeat
Shared FileSystem (e.g., HDFS)
TaskTracker
child JVM
Child
[progress or counter updated]
statusUpdate
MapTask or ReduceTask
tasktracker node

Failure cases

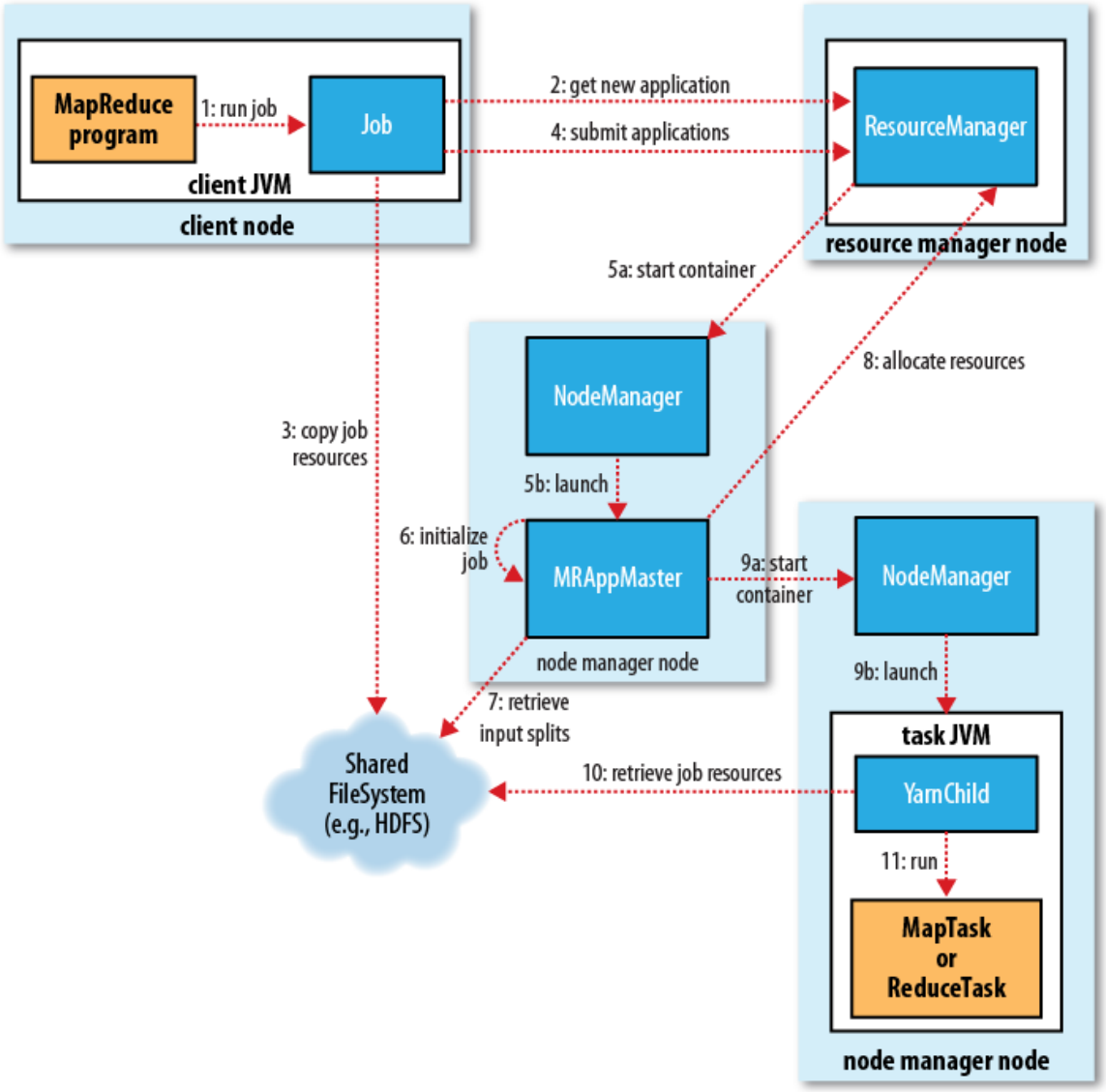
- Task failure:

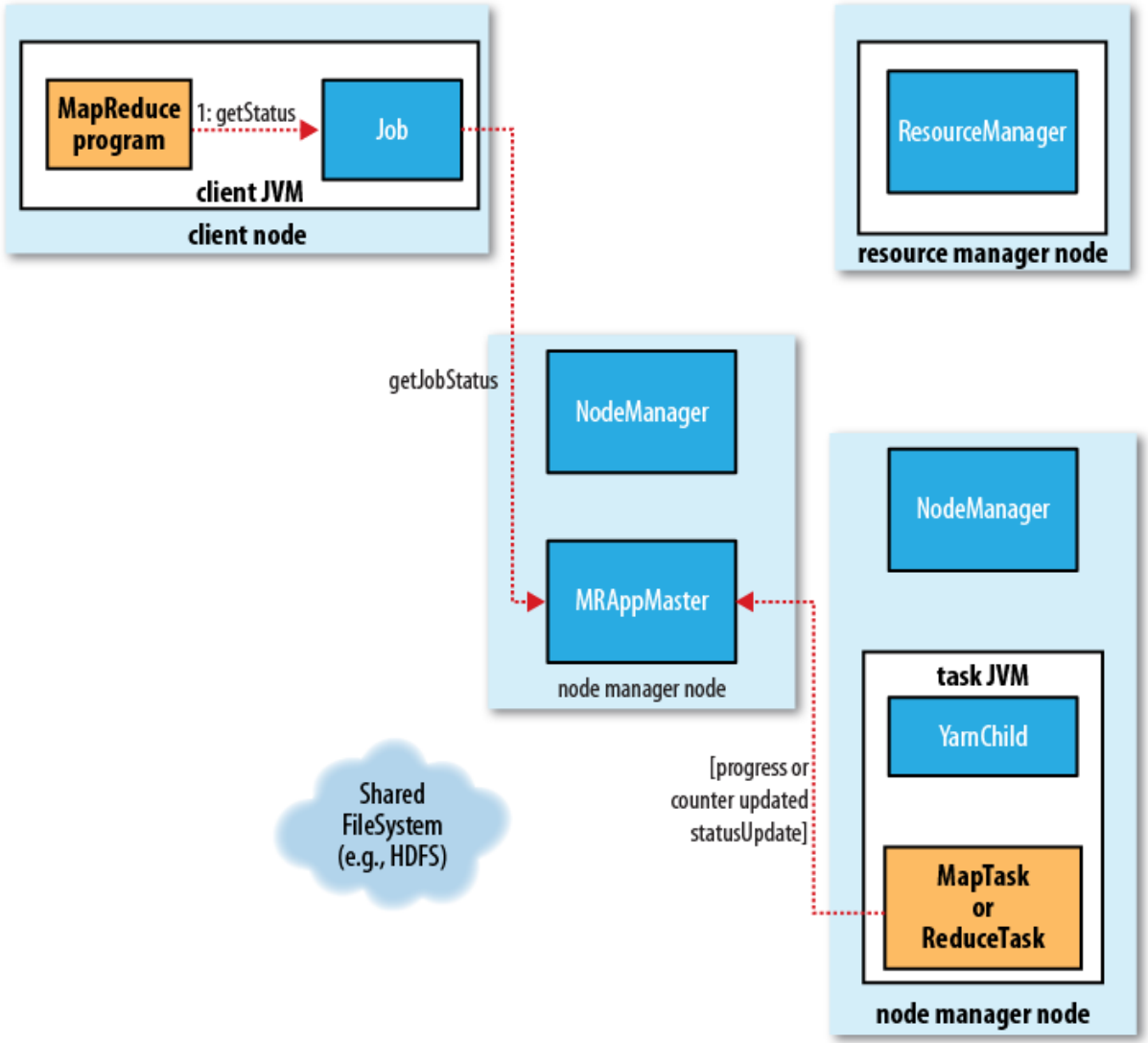
- Code caused issues:
  - Infinite loop/hung cases
  - Runtime errors (reported to tasktrackers)
  - JVM errors

- Mapred.map.max.attempts (4 by default)
- Mapred.reduce.max.attempts (4 by default)

Failure cases

- Tasktracker failure:
- Reschedules the task to another tasktracker.
- Re-run the incompleted tasks.
- Blacklisting tasktrackers – mapred.max.tracker.failures
- **Job tracker failure - SPOF**


MapReduce program
1: run job
Job
client JVM
client node
2: get new application
4: submit applications
ResourceManager
resource manager node
5a: start container
8: allocate resources
NodeManager
3: copy job resources
5b: launch
6: initialize job
MRAppMaster
node manager node
9a: start container
NodeManager
9b: launch
task JVM
YarnChild
11: run
MapTask or ReduceTask
node manager node
7: retrieve input splits
Shared FileSystem (e.g., HDFS)
10: retrieve job resources


MapReduce program
1: getStatus
Job
client JVM
client node
ResourceManager
resource manager node
getJobStatus
NodeManager
MRAppMaster
node manager node
NodeManager
task JVM
YarnChild
MapTask or ReduceTask
node manager node
[progress or counter updated statusUpdate]
Shared FileSystem (e.g., HDFS)