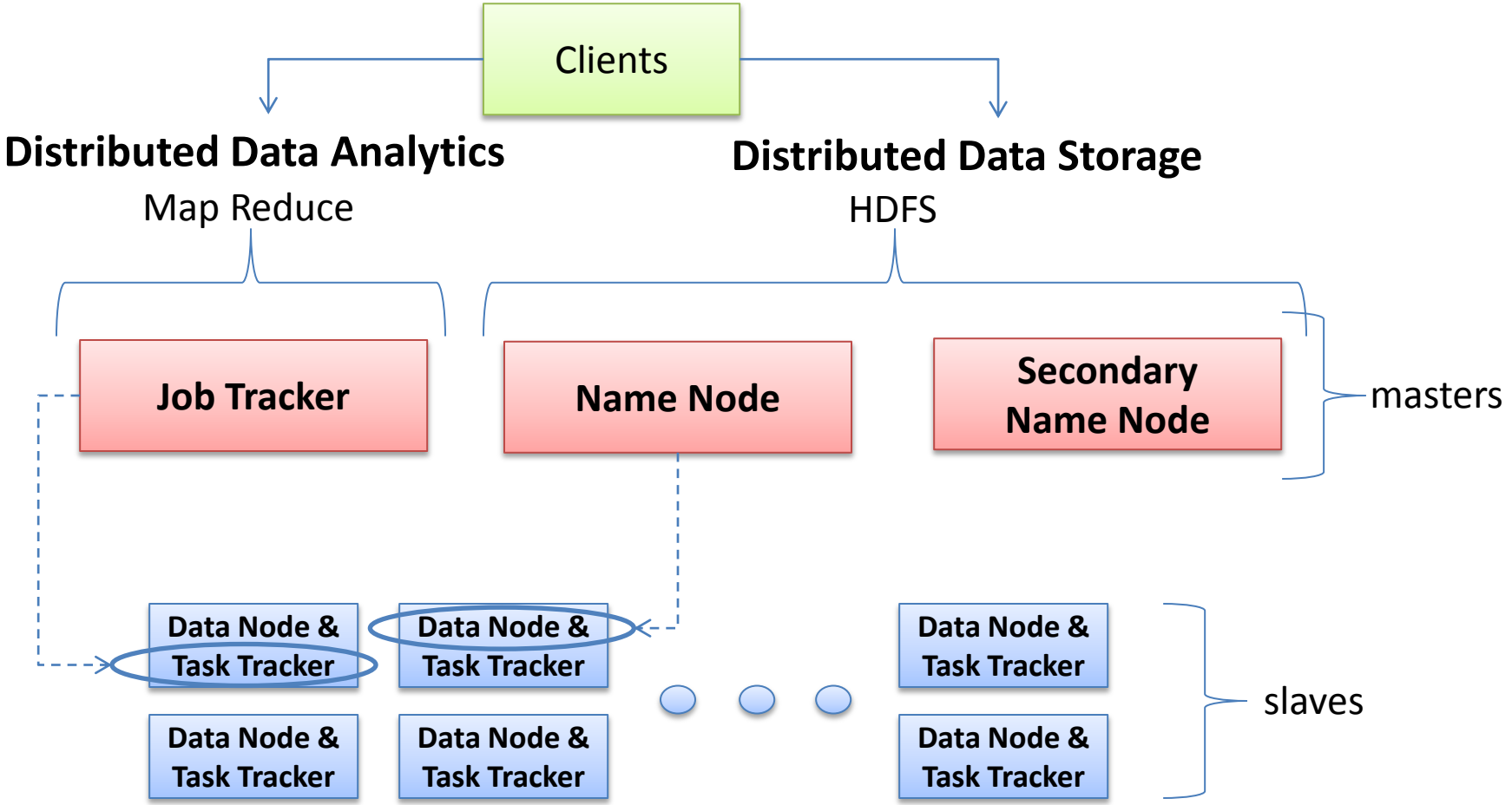
Understanding Hadoop Clusters and the Network

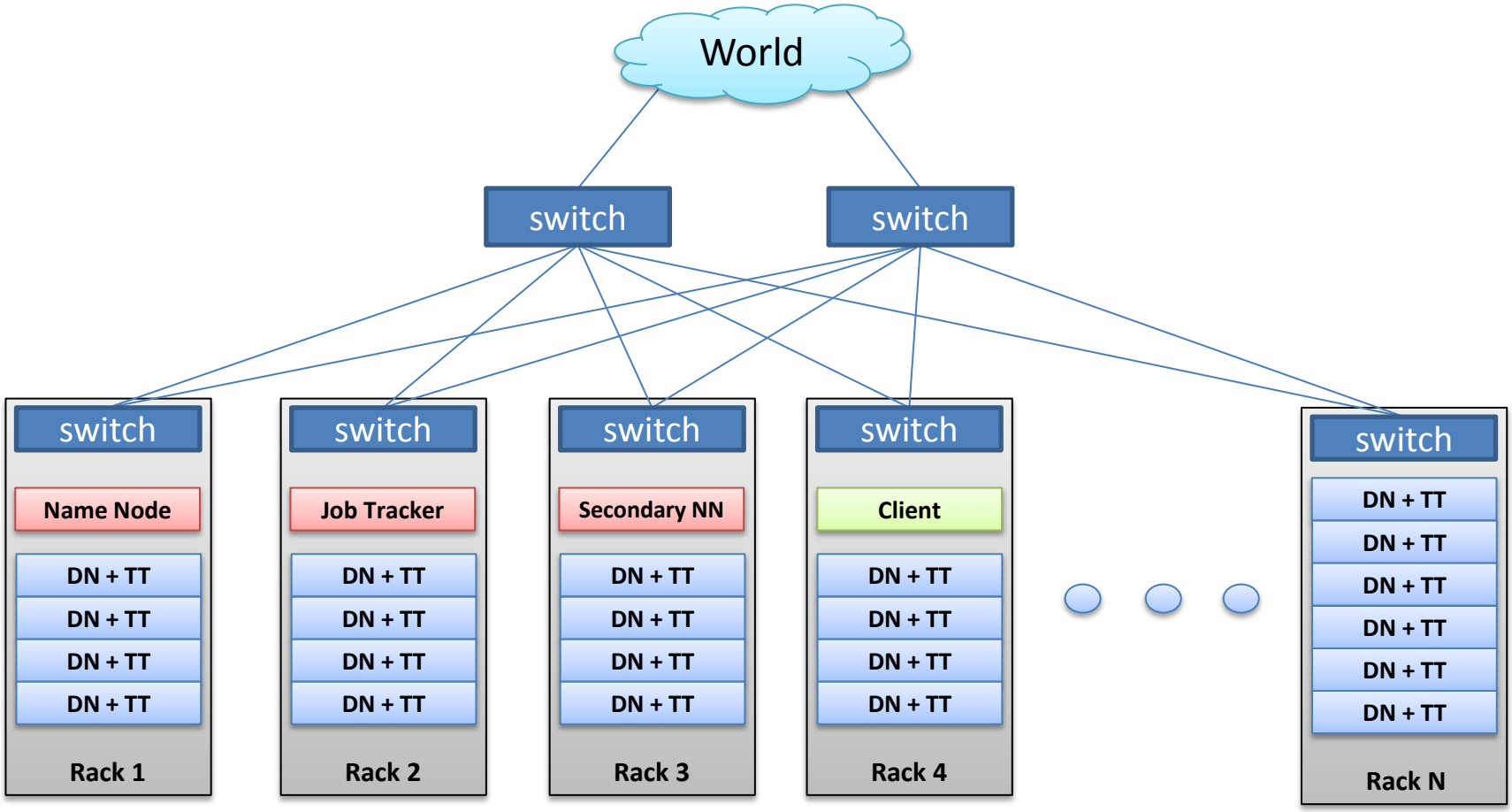
Part 1. Introduction and Overview

Brad Hedlund
http://bradhedlund.com
http://www.linkedin.com/in/bradhedlund
@bradhedlund

Hadoop Server Roles



Hadoop Cluster



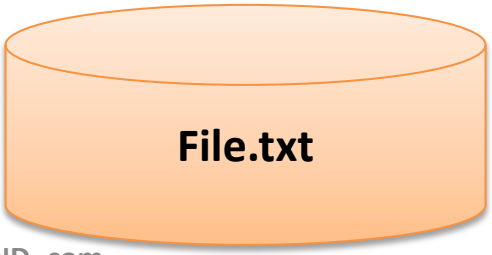
Typical Workflow

- Load data into the cluster (HDFS writes)
- Analyze the data (Map Reduce)
- Store results in the cluster (HDFS writes)
- Read the results from the cluster (HDFS reads)

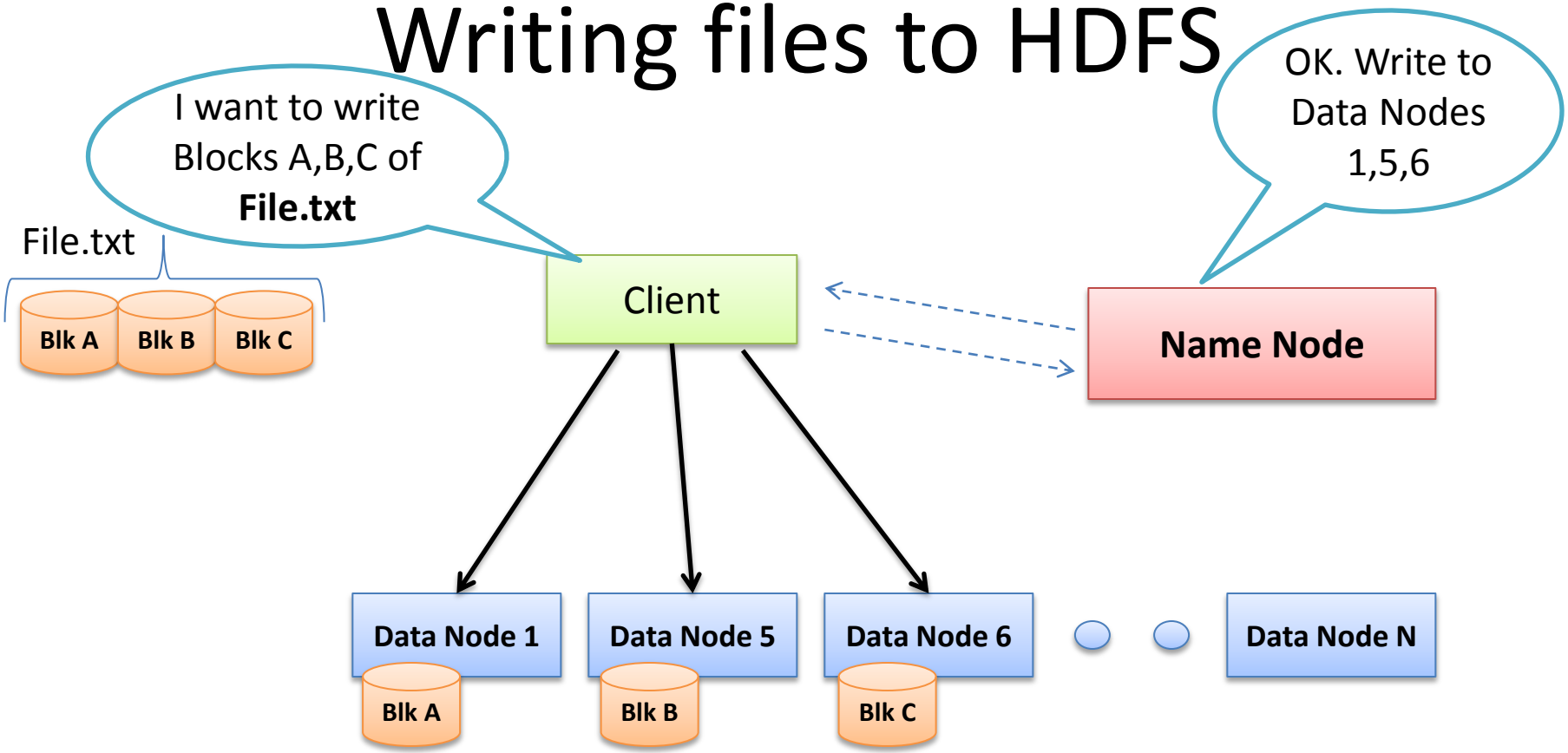
Sample Scenario:

How many times did our customers type the word "**Fraud**" into emails sent to customer service?

Huge file containing all emails sent to customer service

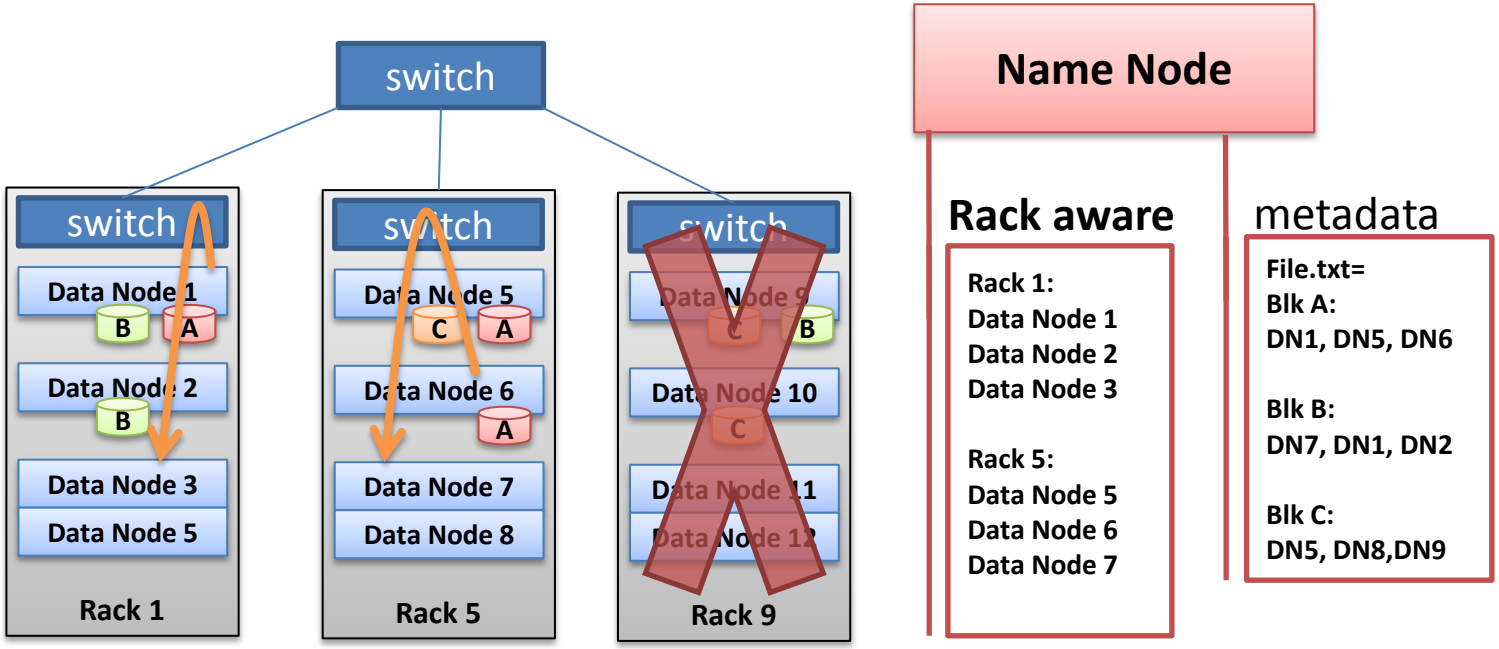


Writing files to HDFS



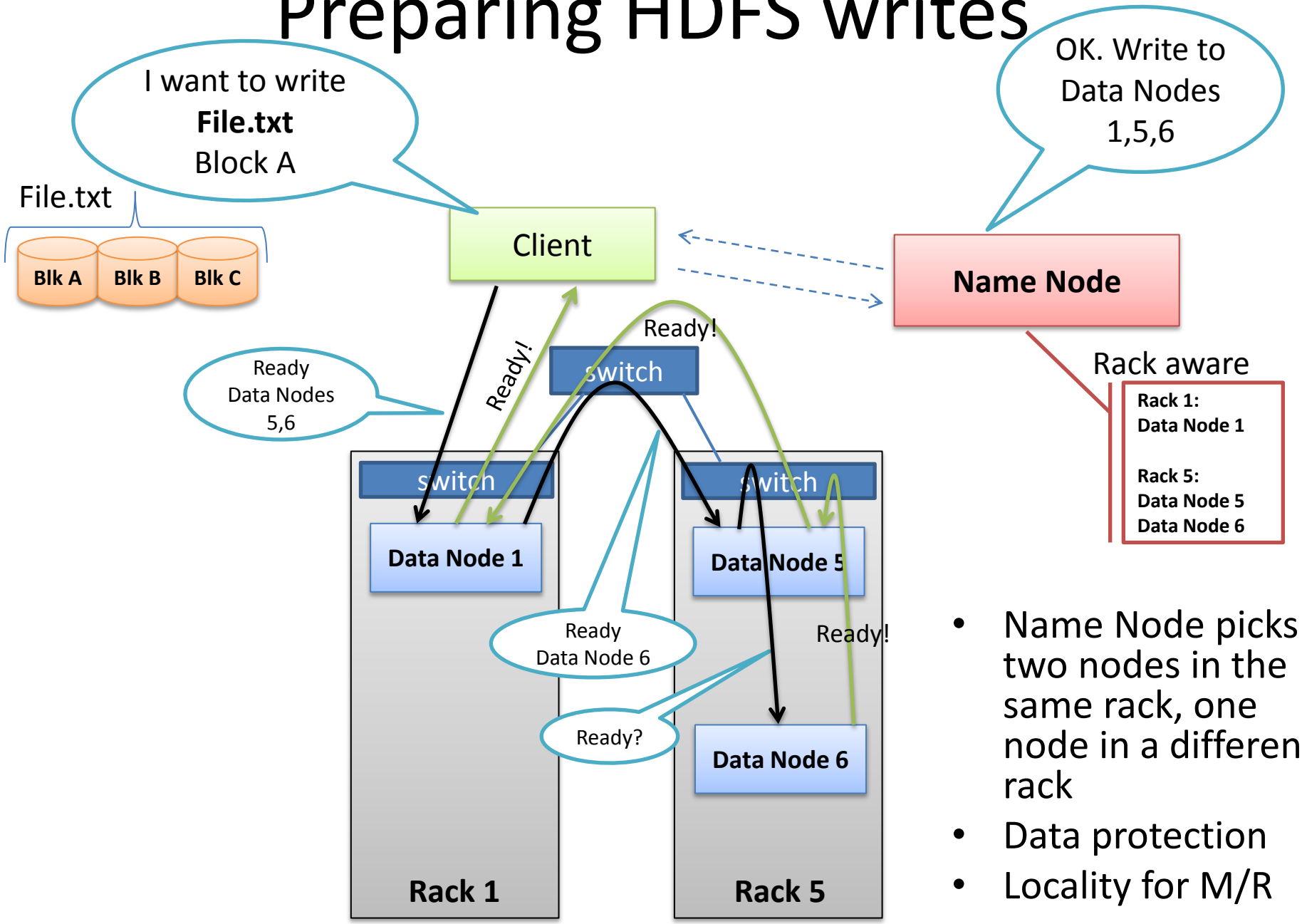
- Client consults Name Node
- Client writes block directly to one Data Node
- Data Nodes replicates block
- Cycle repeats for next block

Hadoop Rack Awareness – Why?



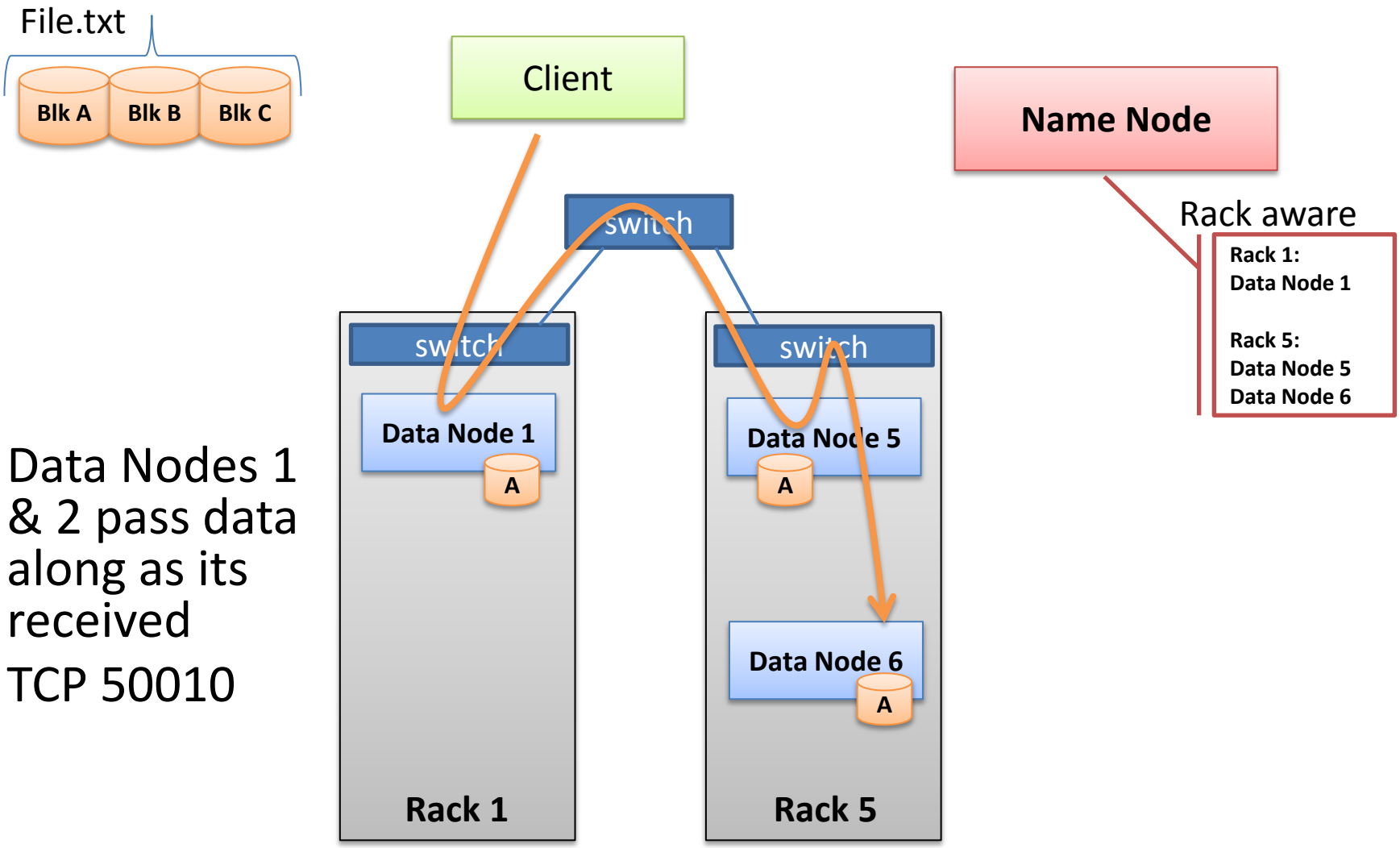
- Never loose all data if entire rack fails
- Keep bulky flows in-rack when possible
- Assumption that in-rack is higher bandwidth, lower latency

Preparing HDFS writes



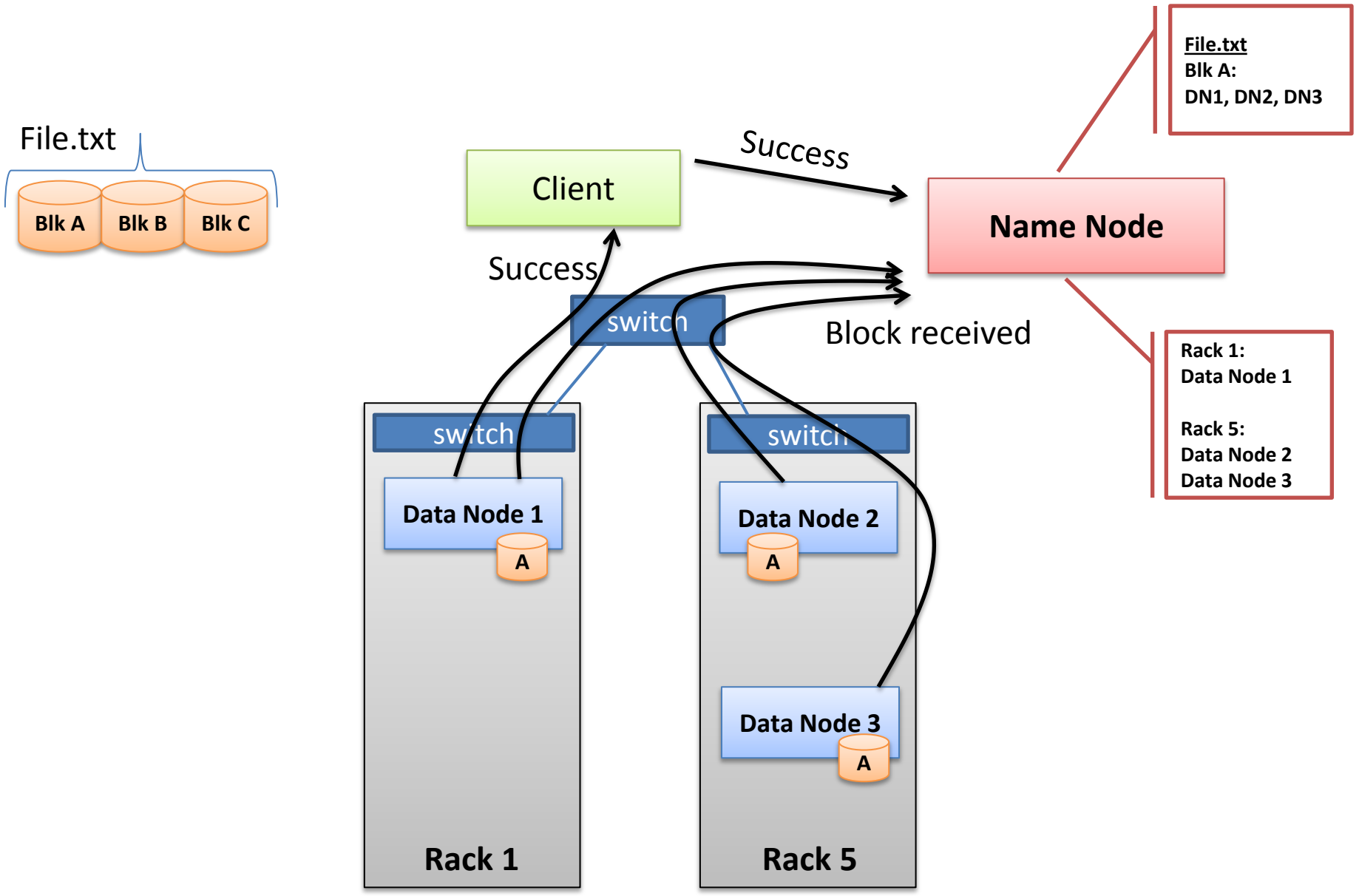
- Name Node picks two nodes in the same rack, one node in a different rack
- Data protection
- Locality for M/R

Pipelined Write

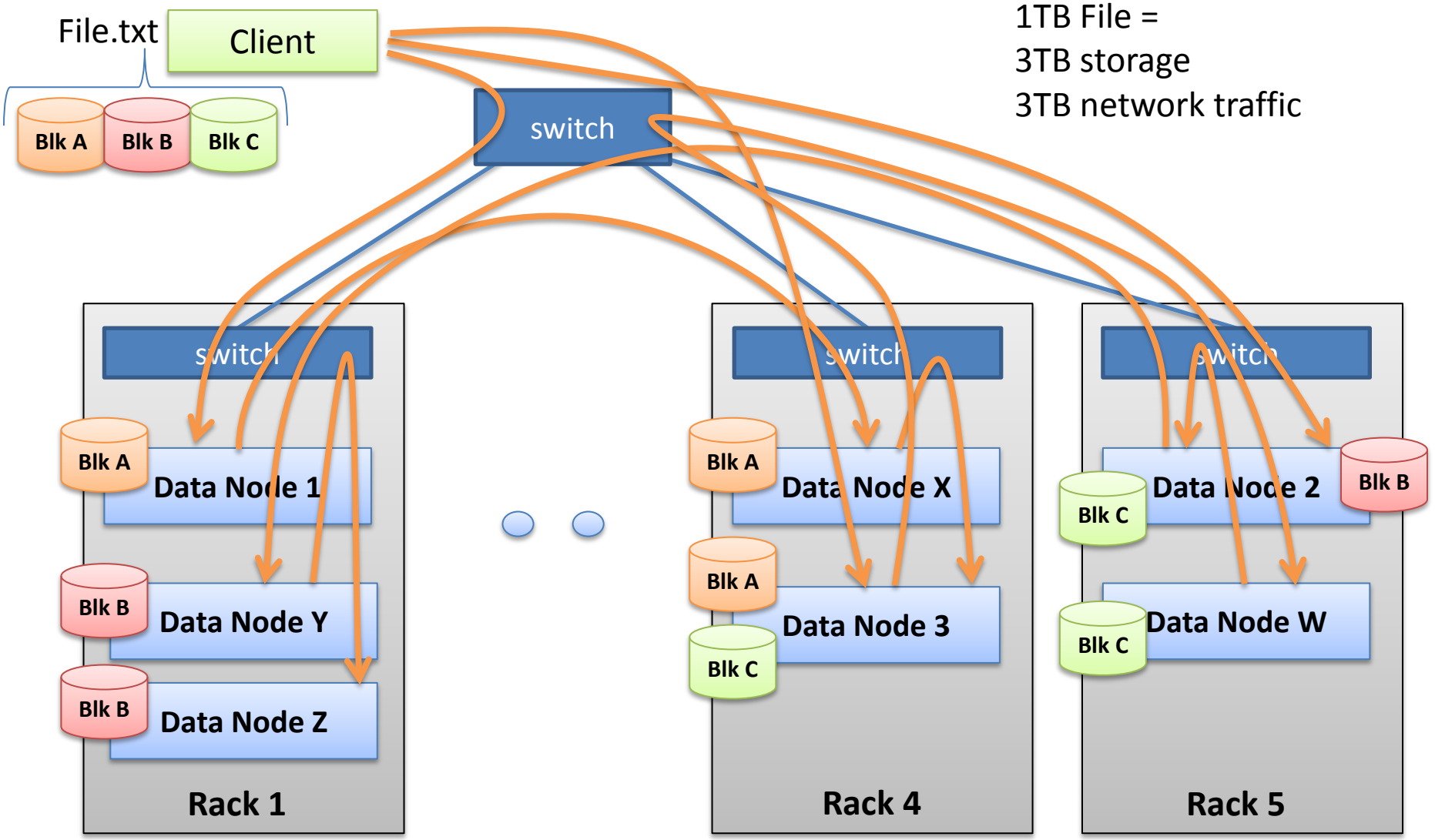


- Data Nodes 1 & 2 pass data along as its received
- TCP 50010

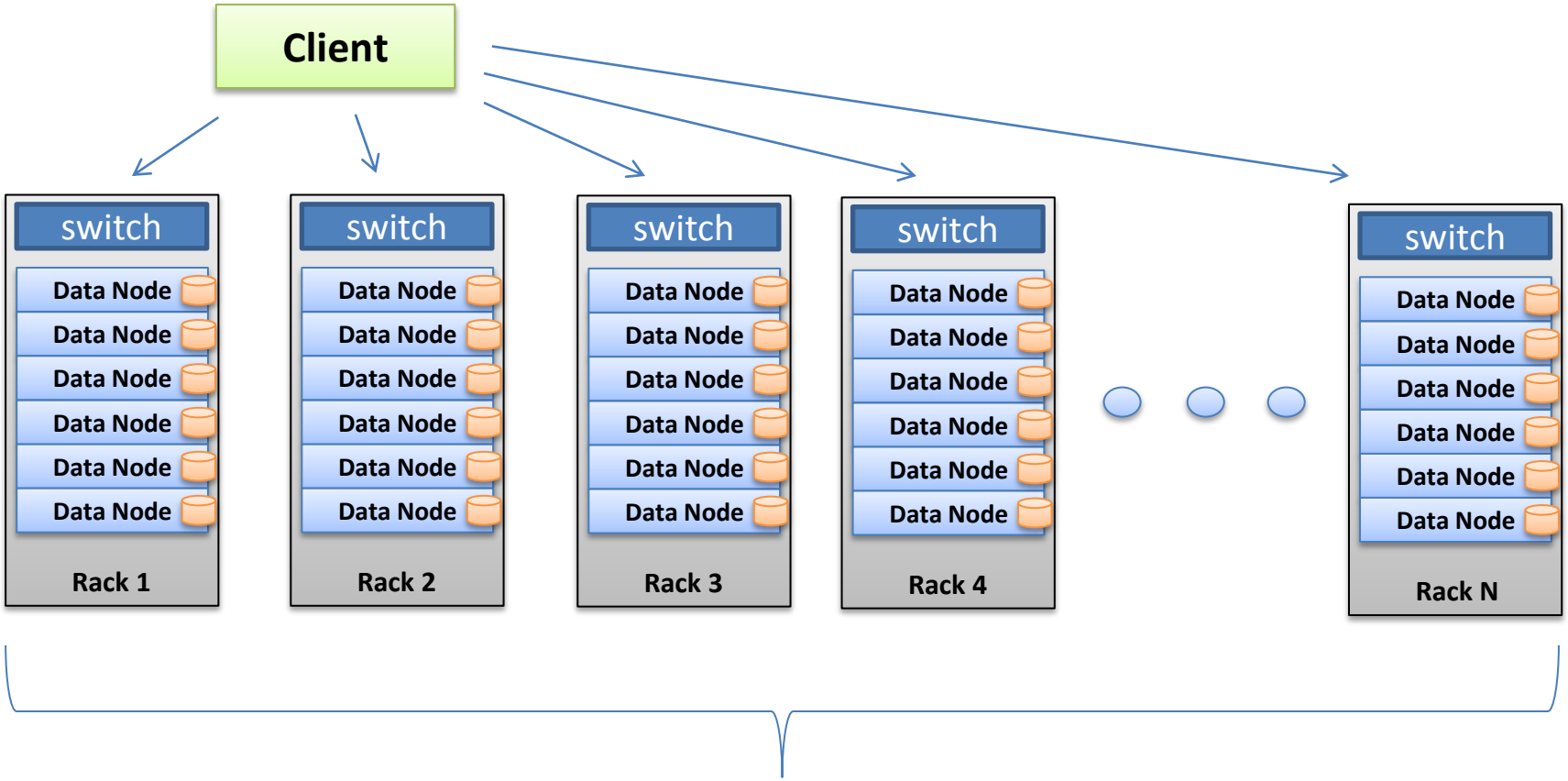
Pipelined Write



Multi-block Replication Pipeline



Client writes Span the HDFS Cluster



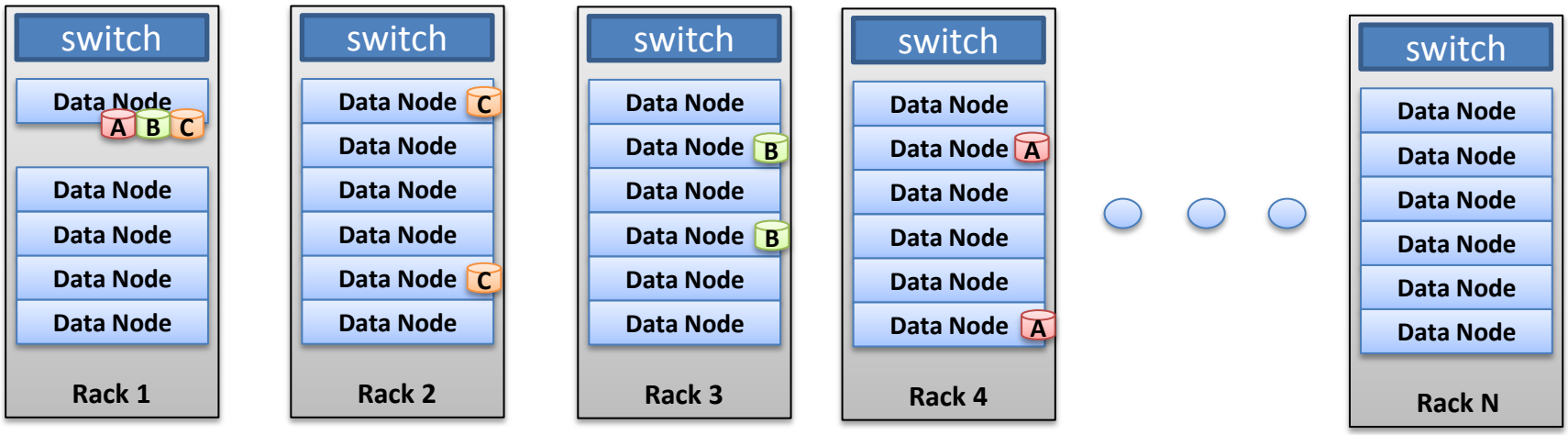
Factors:

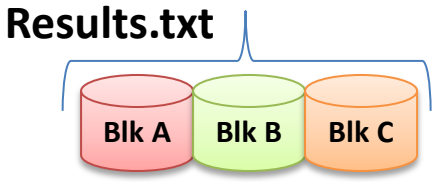
- Block size
- File Size

File.txt

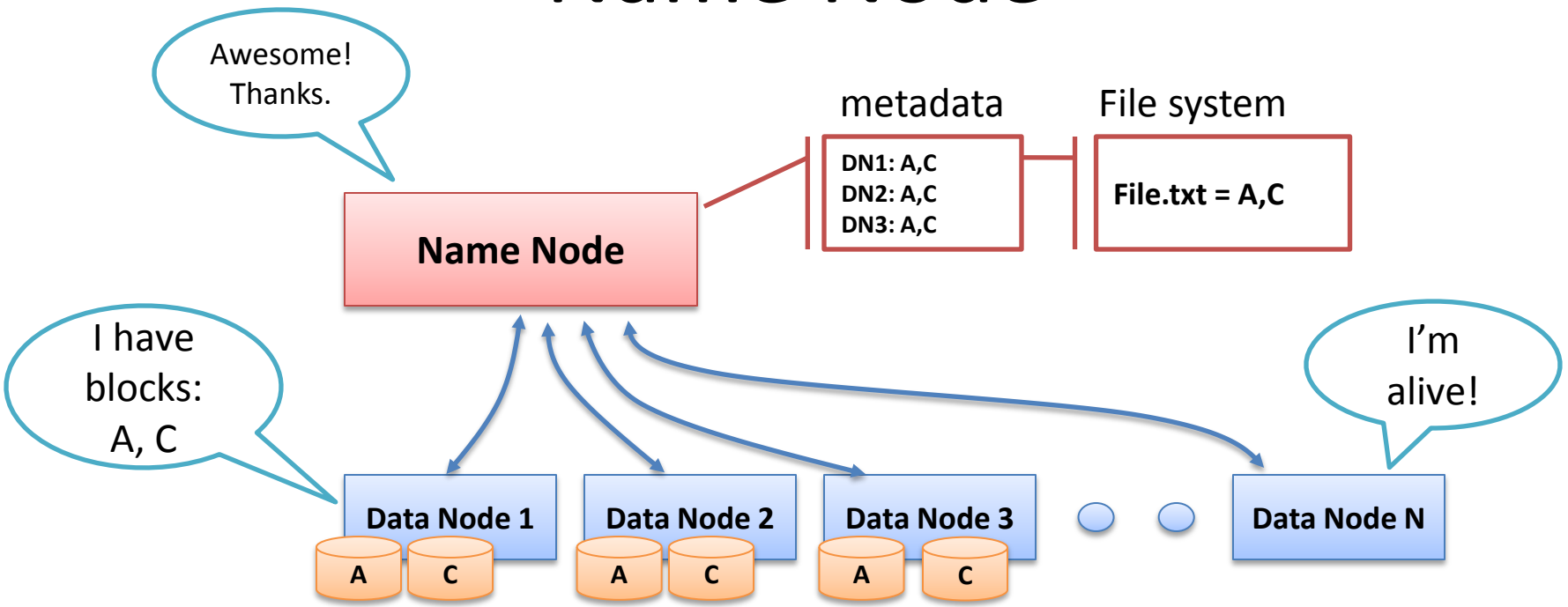
More blocks = Wider spread

Data Node writes span itself, and other racks



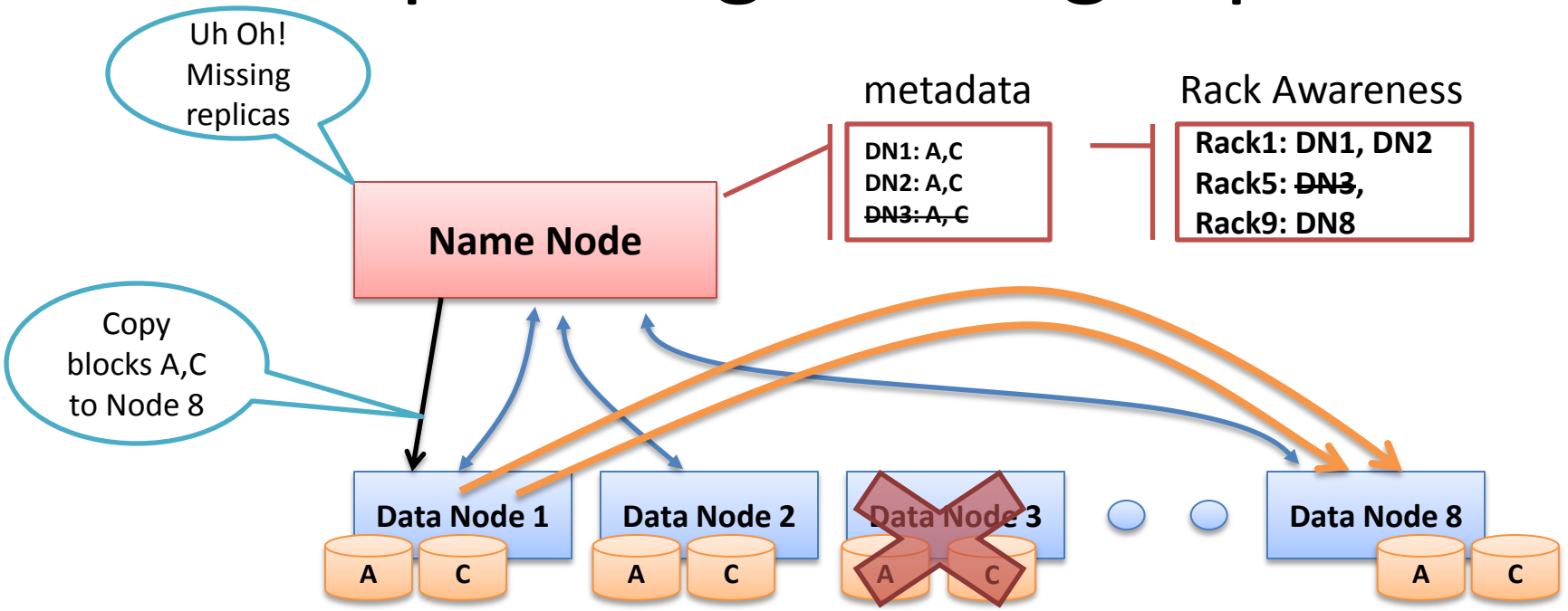


Name Node



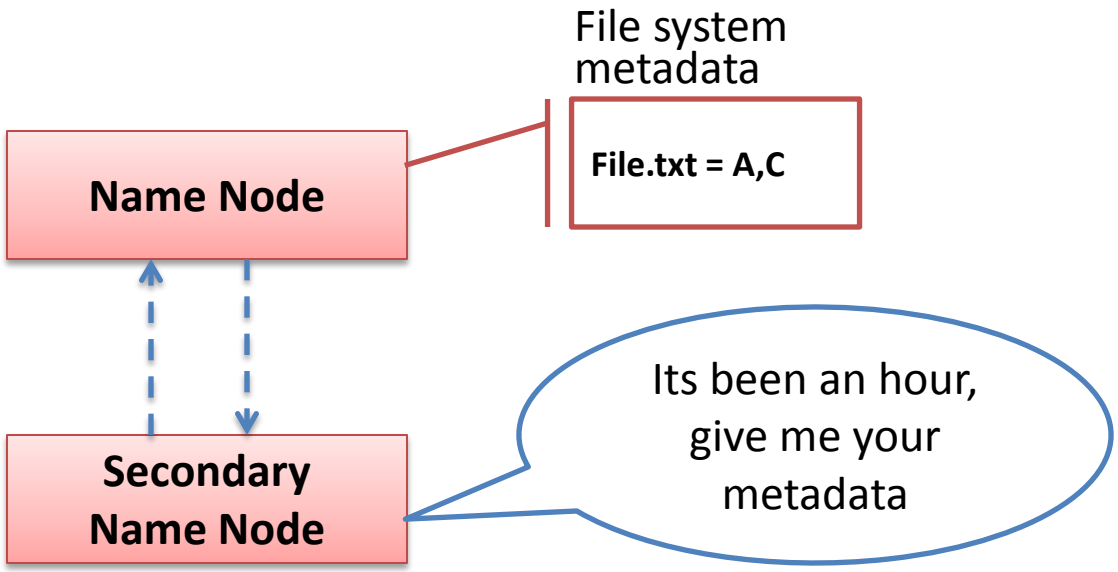
- Data Node sends Heartbeats
- Every 10th heartbeat is a Block report
- Name Node builds metadata from Block reports
- TCP – every 3 seconds
- If Name Node is down, HDFS is down

Re-replicating missing replicas



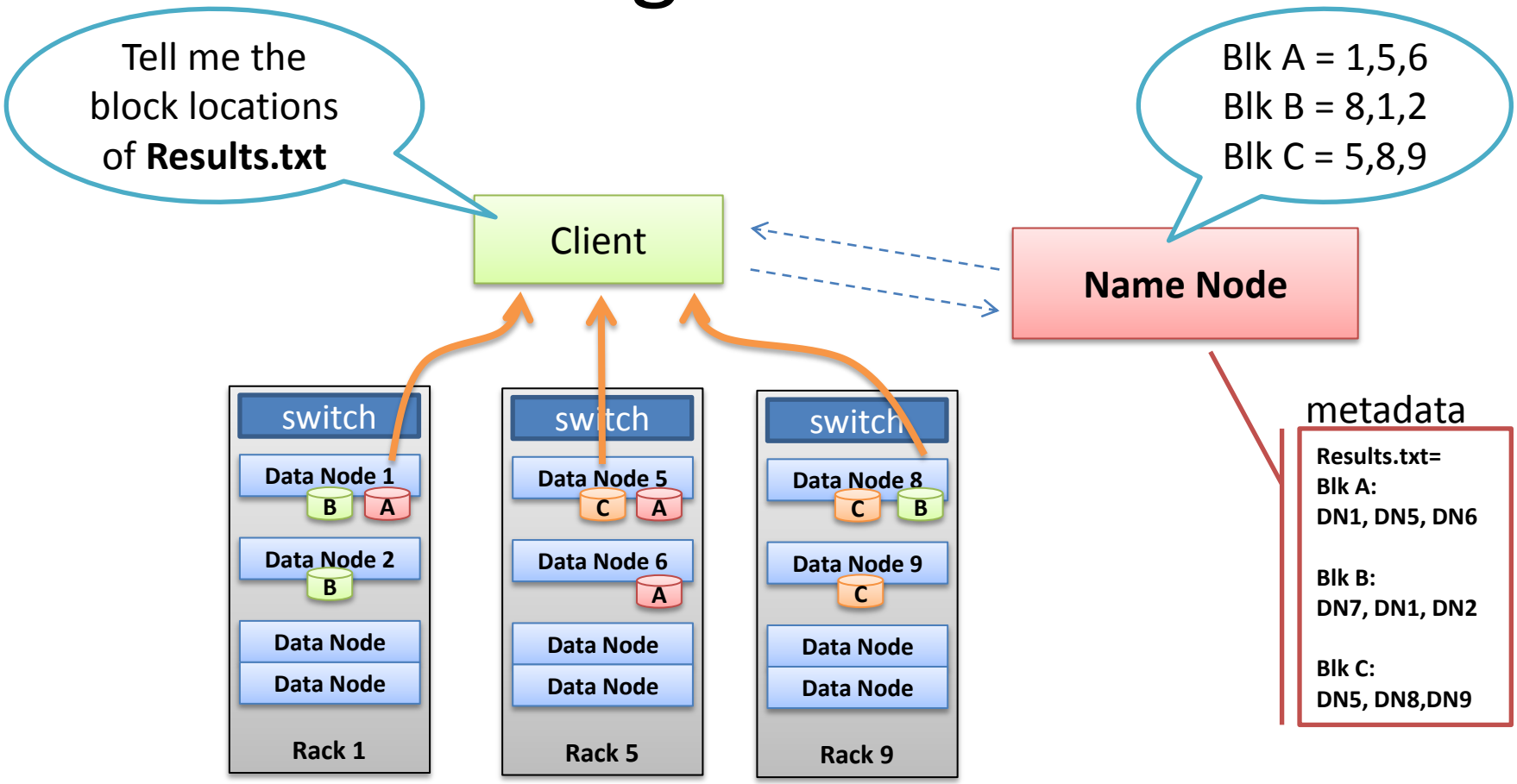
- Missing Heartbeats signify lost Nodes
- Name Node consults metadata, finds affected data
- Name Node consults Rack Awareness script
- Name Node tells a Data Node to re-replicate

Secondary Name Node



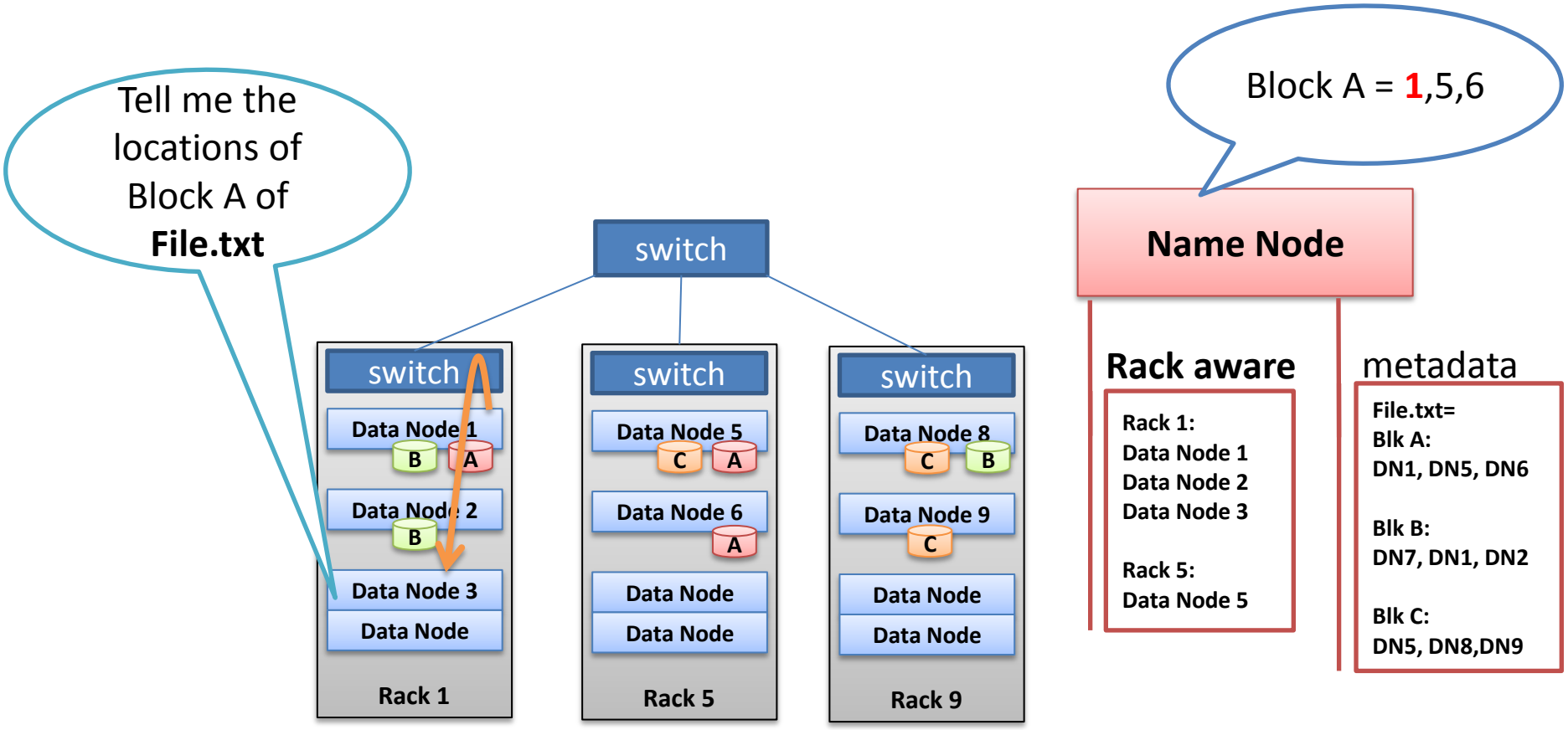
- Not a hot standby for the Name Node
- Connects to Name Node every hour*
- Housekeeping, backup of Name Node metadata
- Saved metadata can rebuild a failed Name Node

Client reading files from HDFS



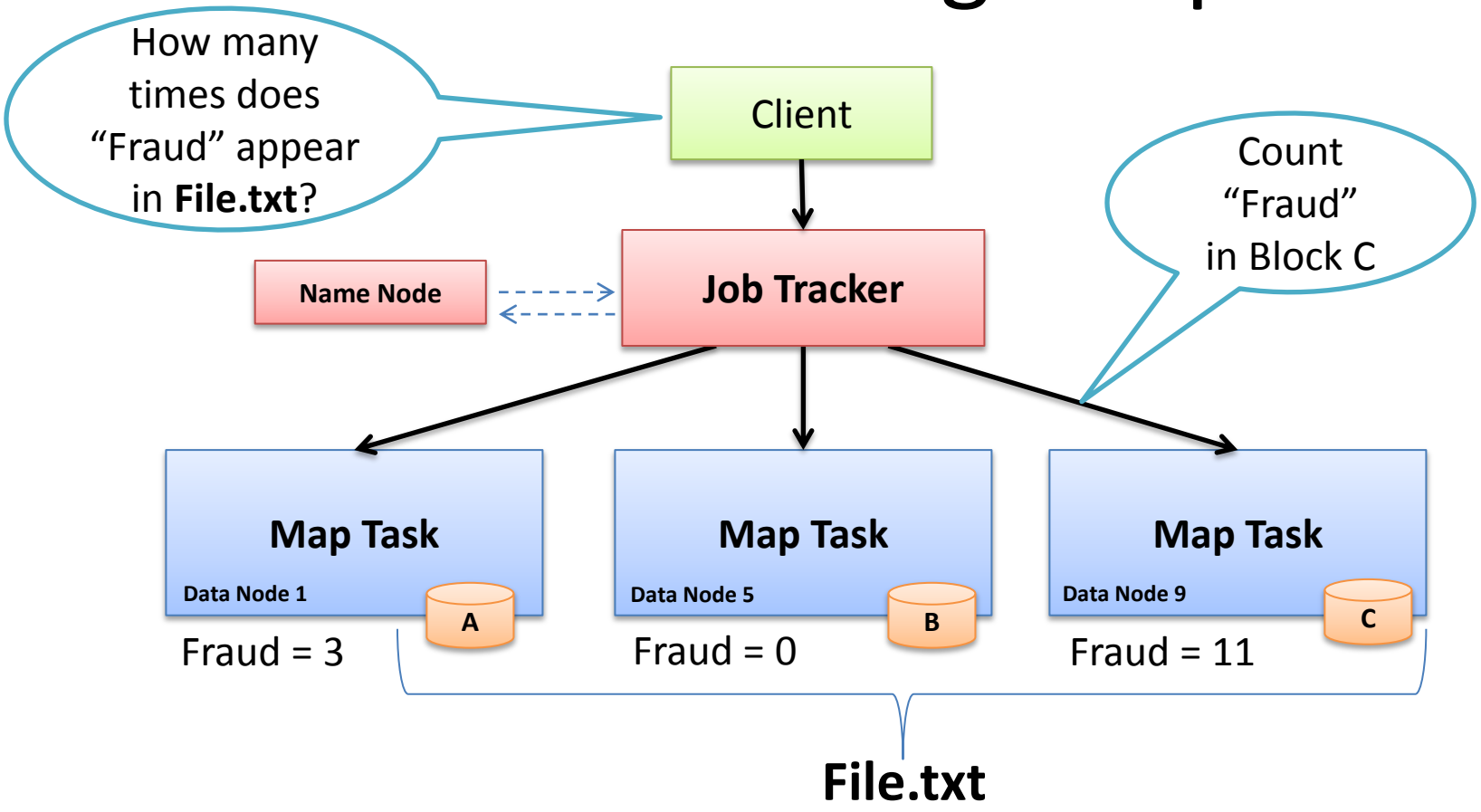
- Client receives Data Node list for each block
- Client picks first Data Node for each block
- Client reads blocks sequentially

Data Node reading files from HDFS



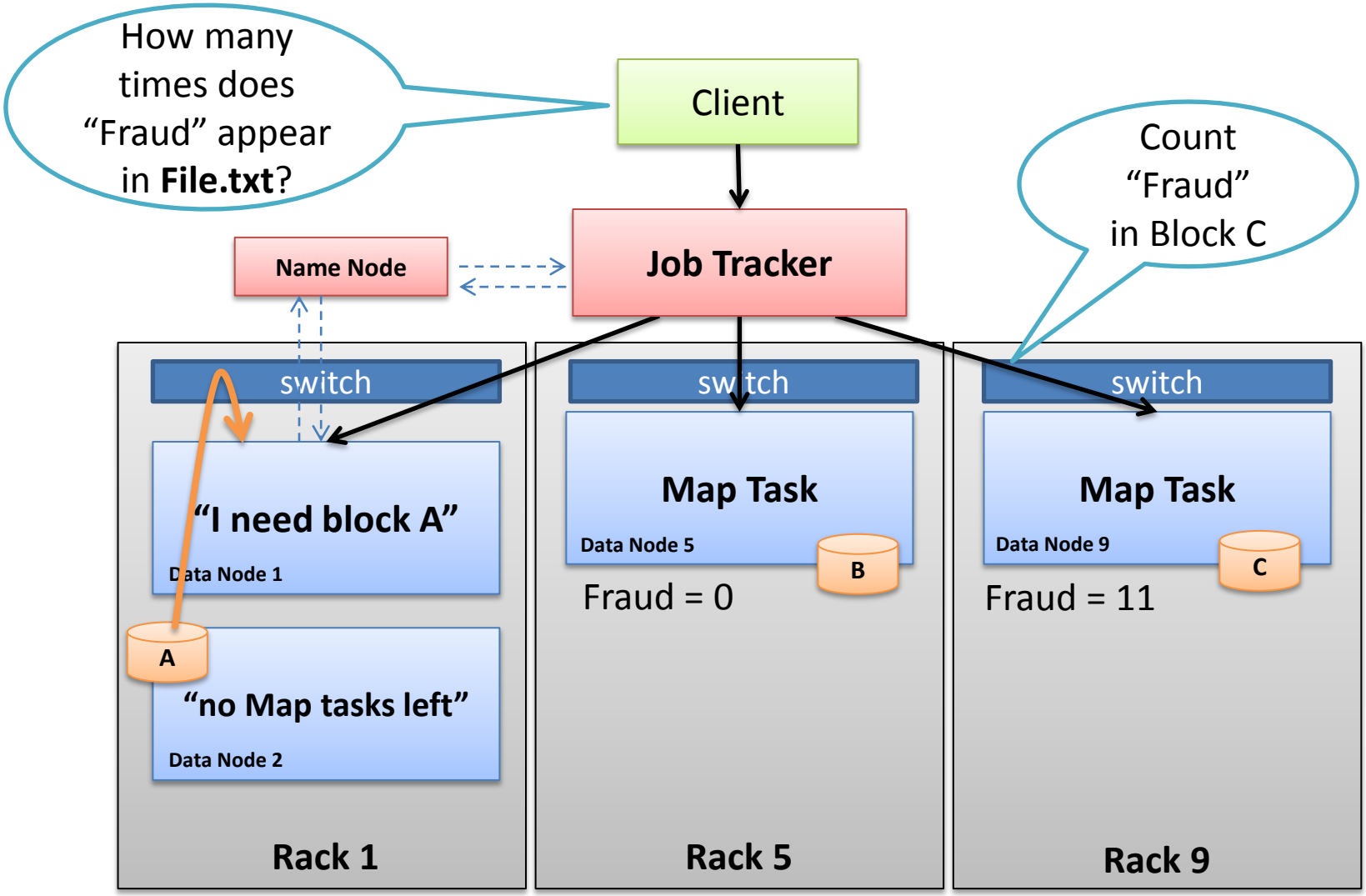
- Name Node provides rack local Nodes first
- Leverage in-rack bandwidth, single hop

Data Processing: Map



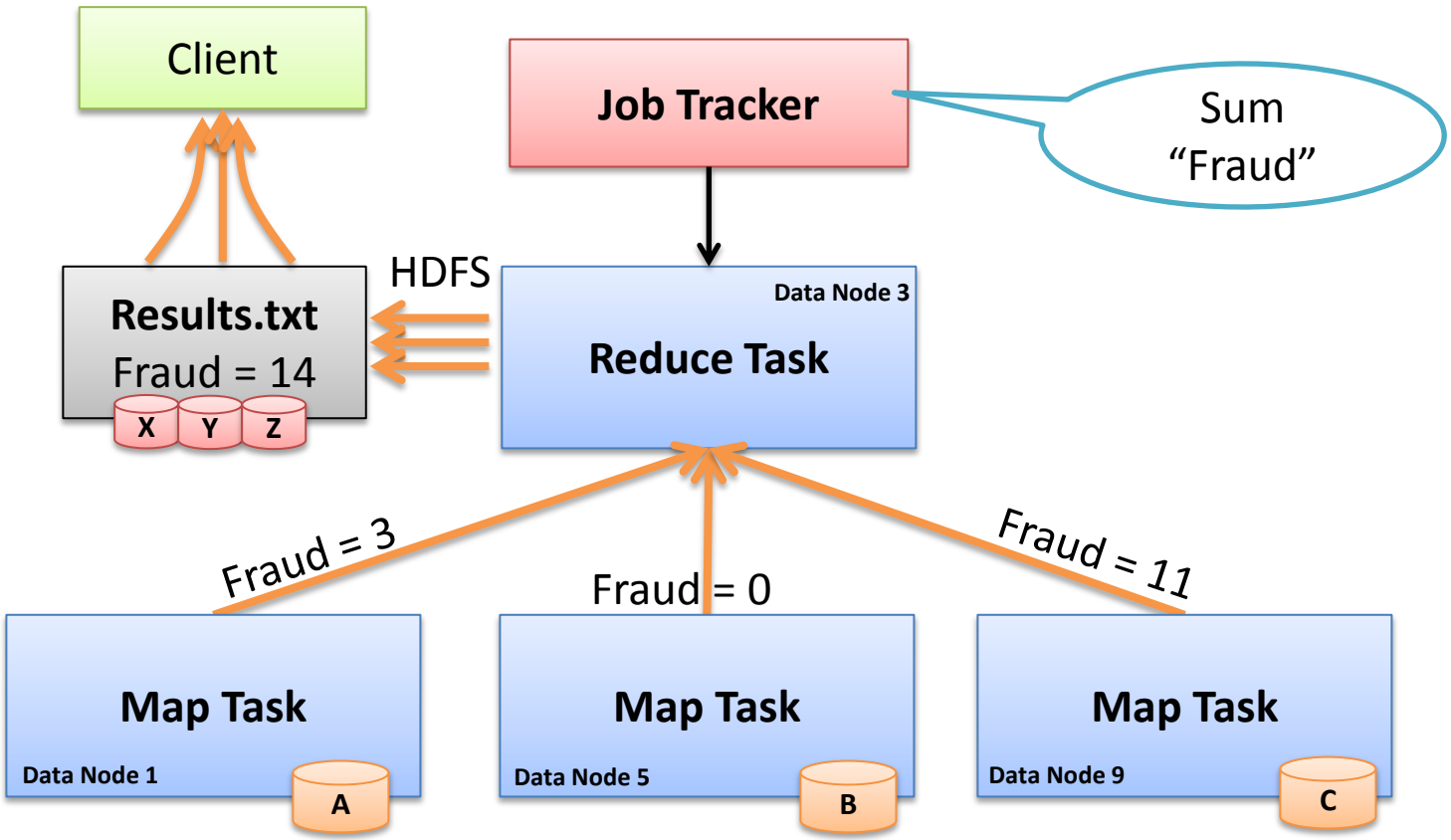
- **Map:** "Run this computation on your local data"
- Job Tracker delivers Java code to Nodes with local data

What if data isn't local?



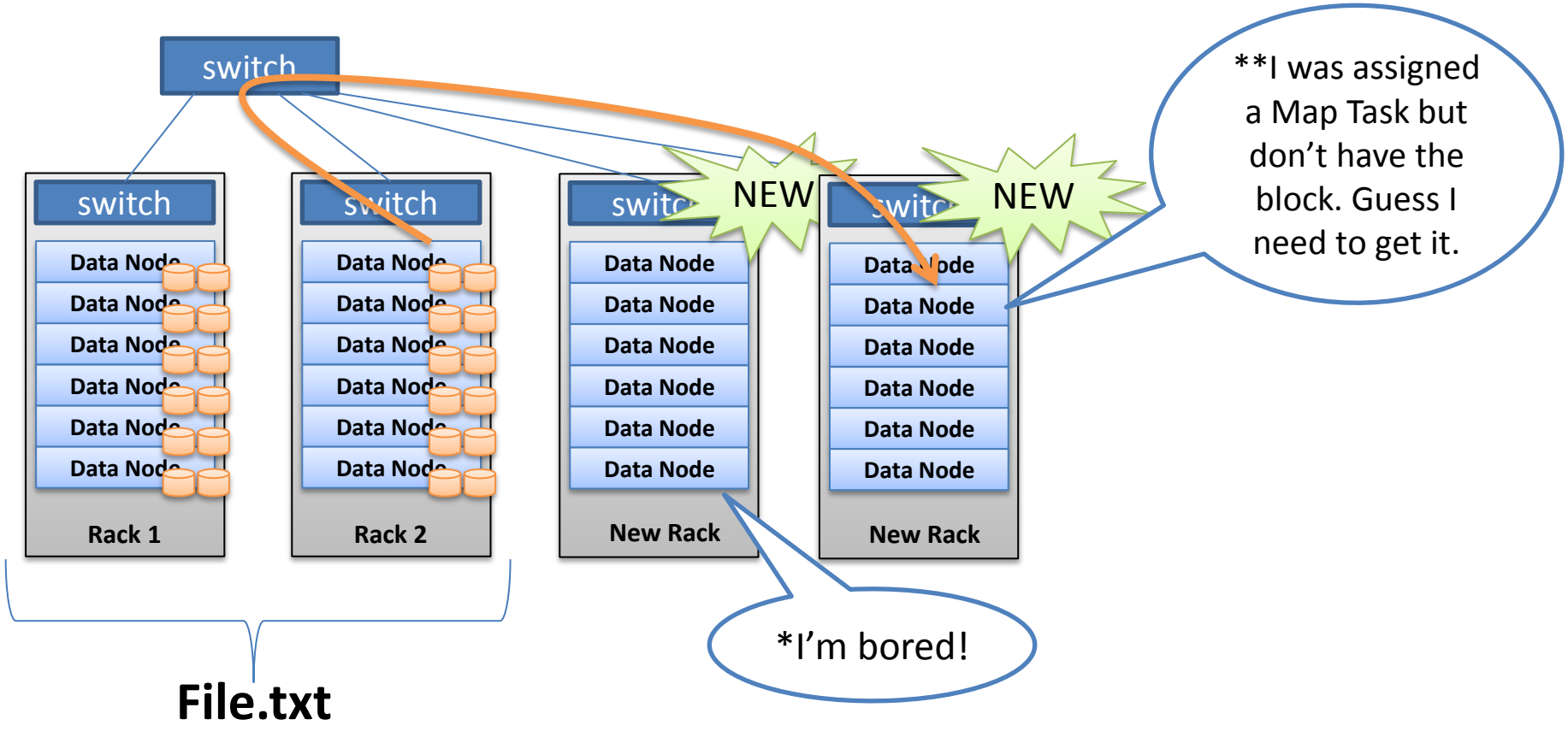
- Job Tracker tries to select Node in same rack as data
- Name Node rack awareness

Data Processing: Reduce



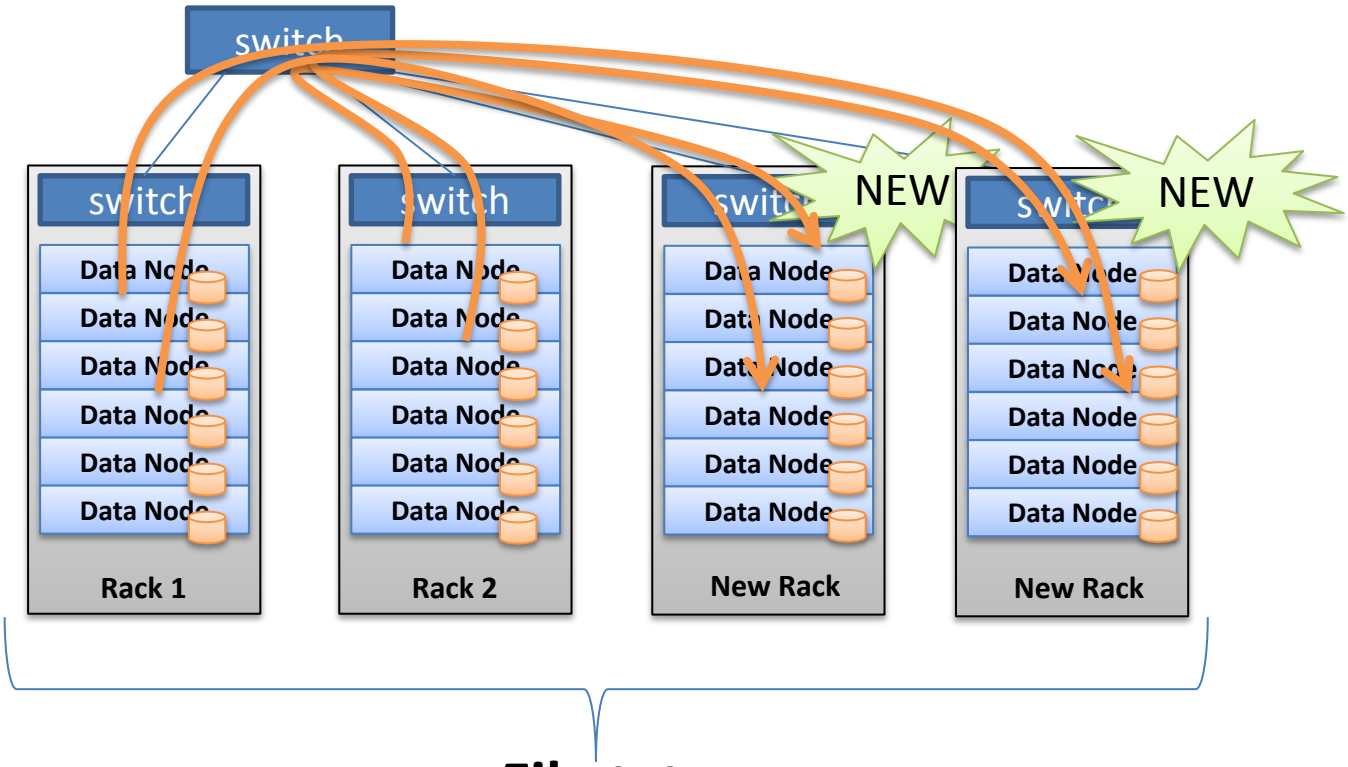
- **Reduce:** "Run this computation across Map results"
- Map Tasks deliver output data over the network
- Reduce Task data output written to and read from HDFS

Unbalanced Cluster



- Hadoop prefers local processing <u>if possible</u>
- New servers underutilized for Map Reduce, HDFS*
- Might see more network bandwidth, slower job times**

Cluster Balancing




```
brad@cloudera-1:~$hadoop balancer
```

- Balancer utility (if used) runs in the background
- Does not interfere with Map Reduce or HDFS
- Default speed limit 1 MB/s

Thanks!

Narrated at:

http://bradhedlund.com/?p=3108