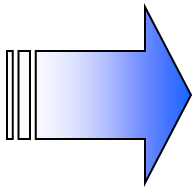
Hadoop architecture

IBM Information Management
Cloud Computing Center of Competence
IBM Canada Labs

Agenda

- Terminology review
- HDFS
- MapReduce
- Type of nodes
- Topology awareness

Agenda



- Terminology review
- HDFS
- MapReduce
- Type of nodes
- Topology awareness

Terminology review

Node 1

Terminology review

Node 1

Node 2

Terminology review

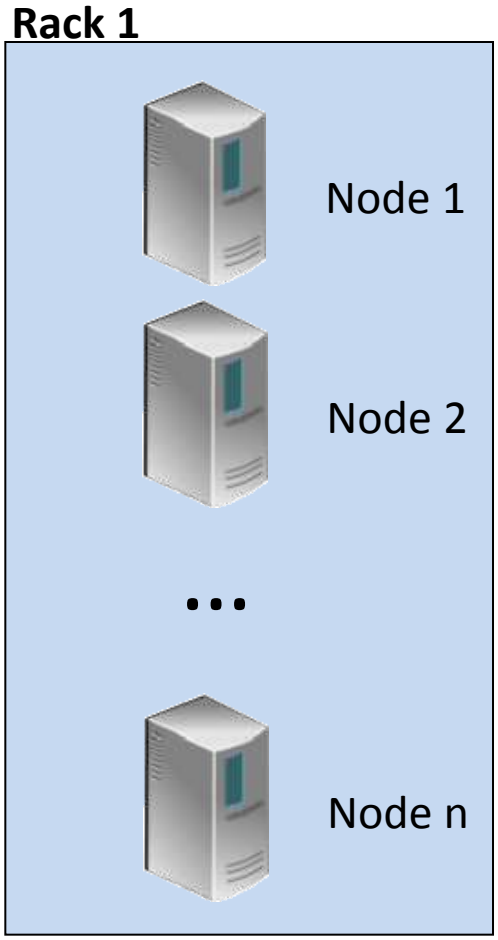
Node 1

Node 2

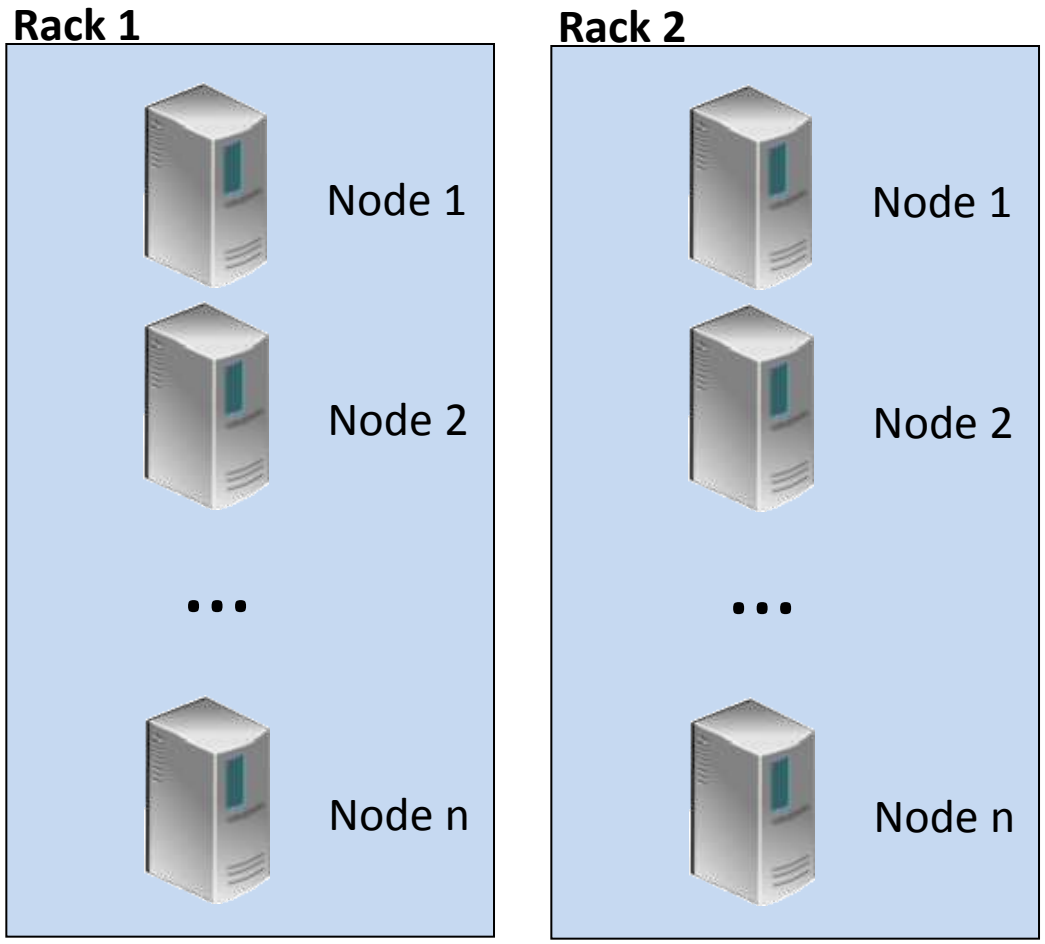
...

Node n

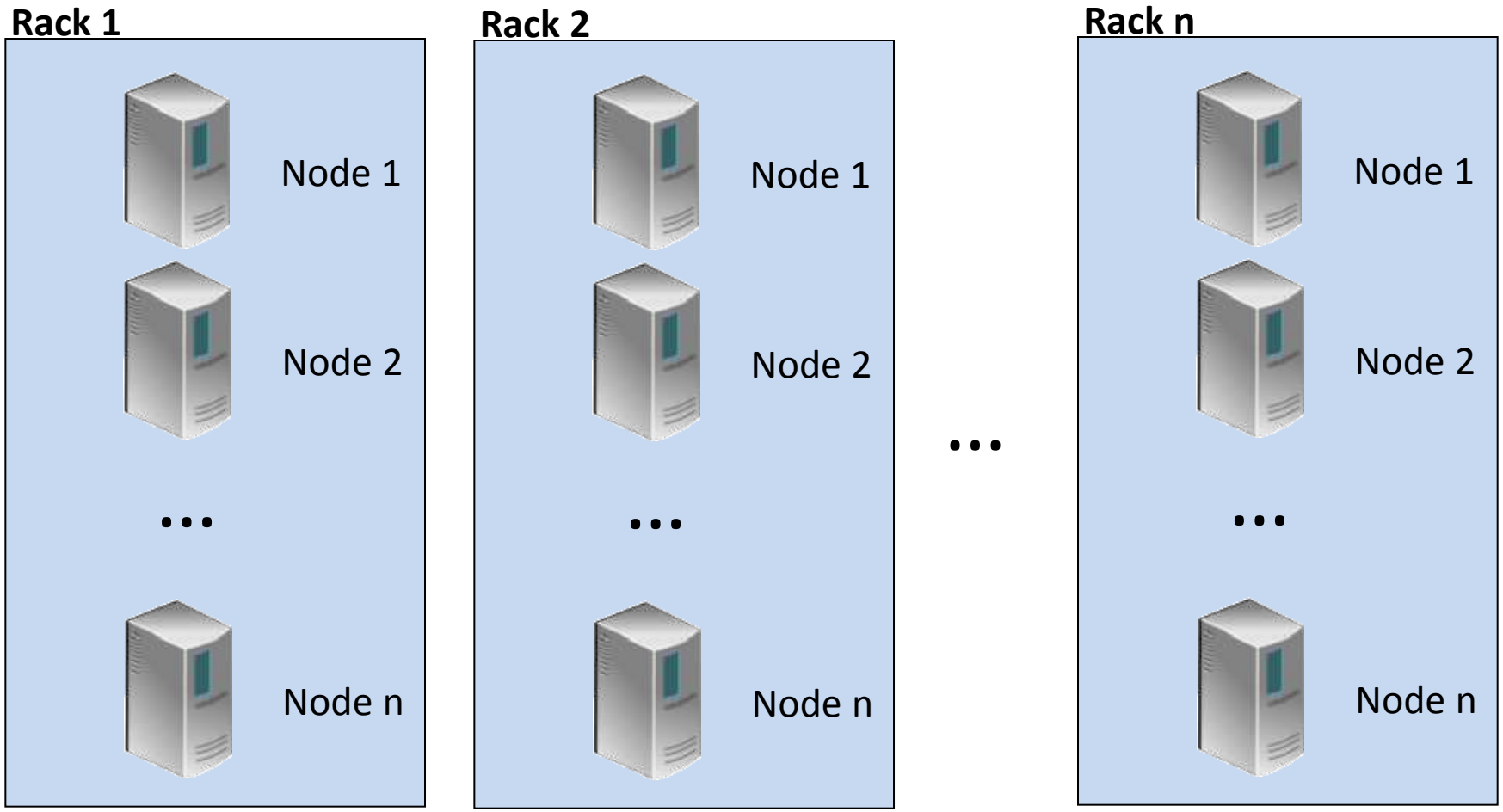
Terminology review



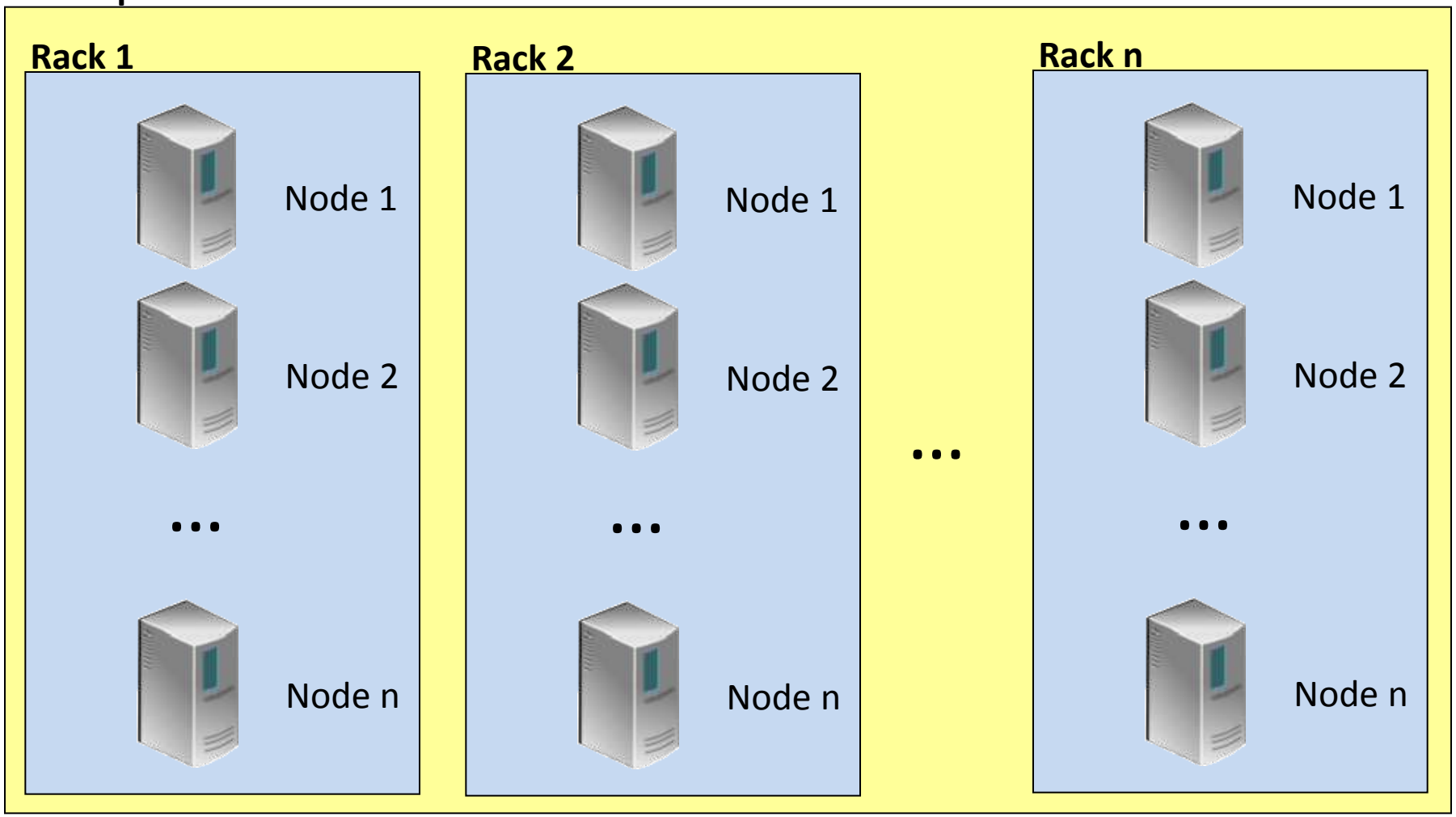
Terminology review



Terminology review



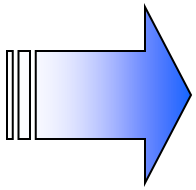
Terminology review



Hadoop architecture

- Two main components:
  - Hadoop Distributed File System (HDFS)
  - MapReduce Engine

Agenda



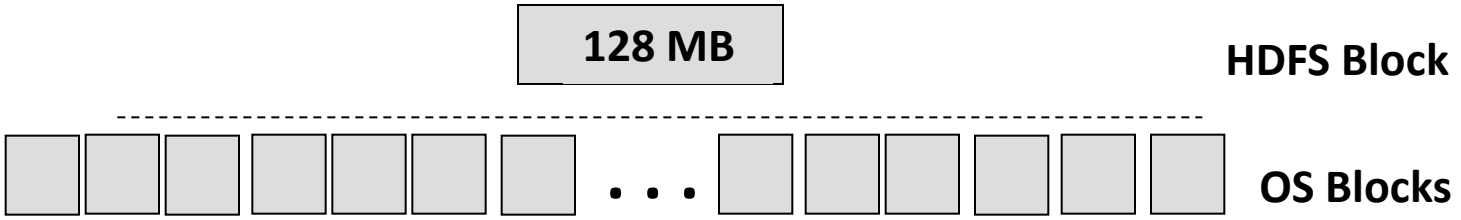
- Terminology review
- HDFS
- MapReduce
- Type of nodes
- Topology awareness

Hadoop distributed file system (HDFS)

- Hadoop file system that 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 - Blocks

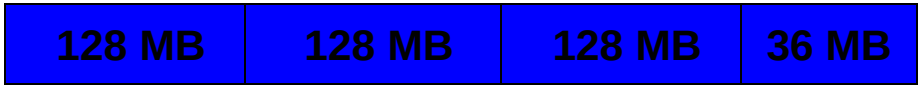
- File Blocks
  - 64MB (default), 128MB (recommended) – compare to 4KB in UNIX
  - Behind the scenes, 1 HDFS block is supported by multiple operating system (OS) blocks



HDFS - Blocks

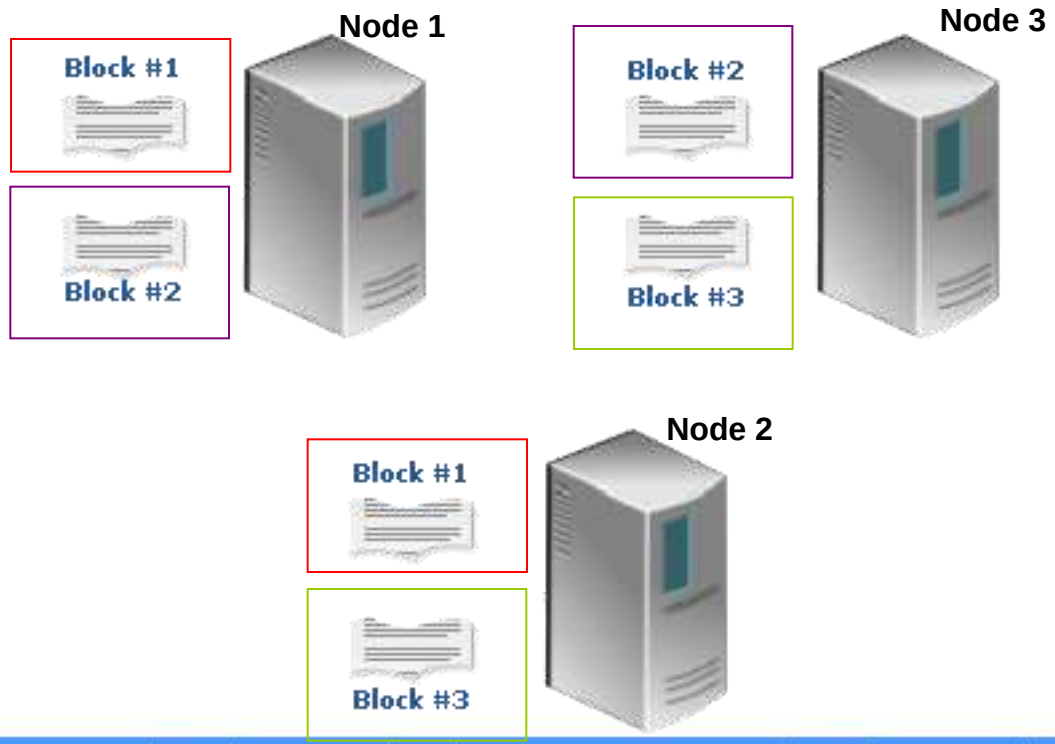
- **Fits well with replication to provide fault tolerance and availability**

- **Advantages of blocks:**
  - 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:

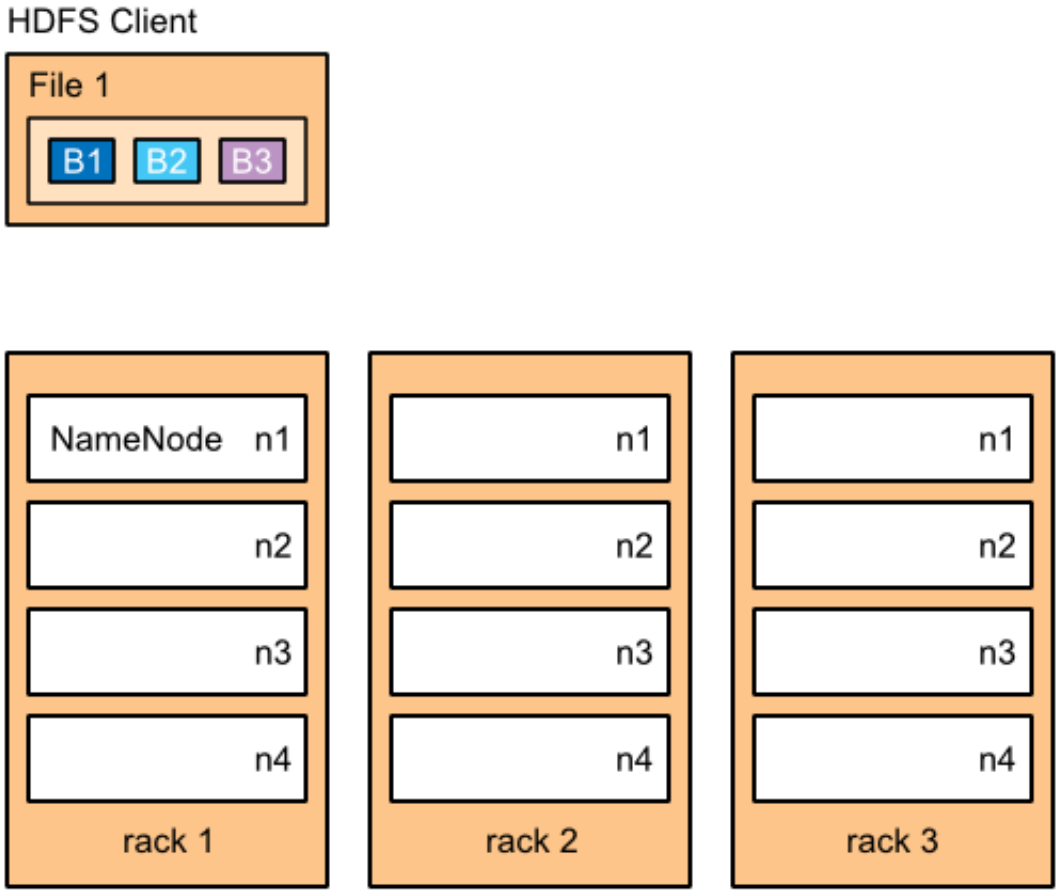


HDFS - Replication

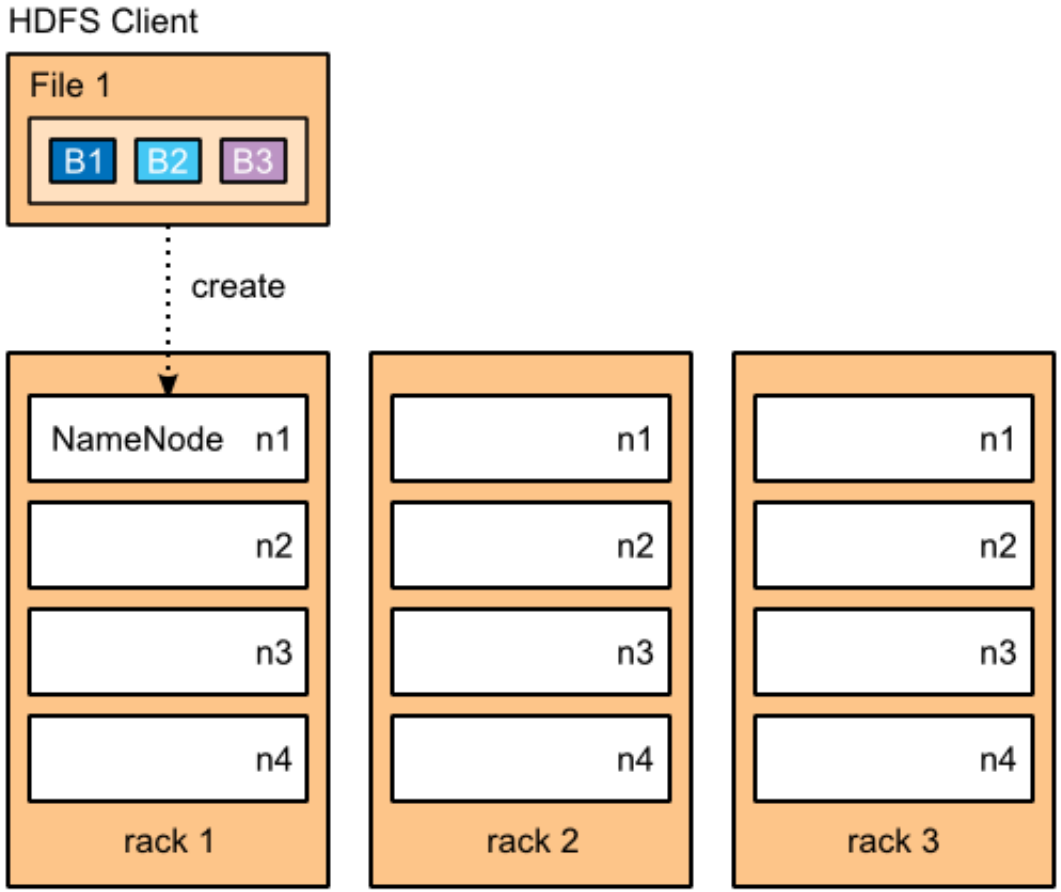
- Blocks with data are replicated to multiple nodes
- Allows for node failure without data loss



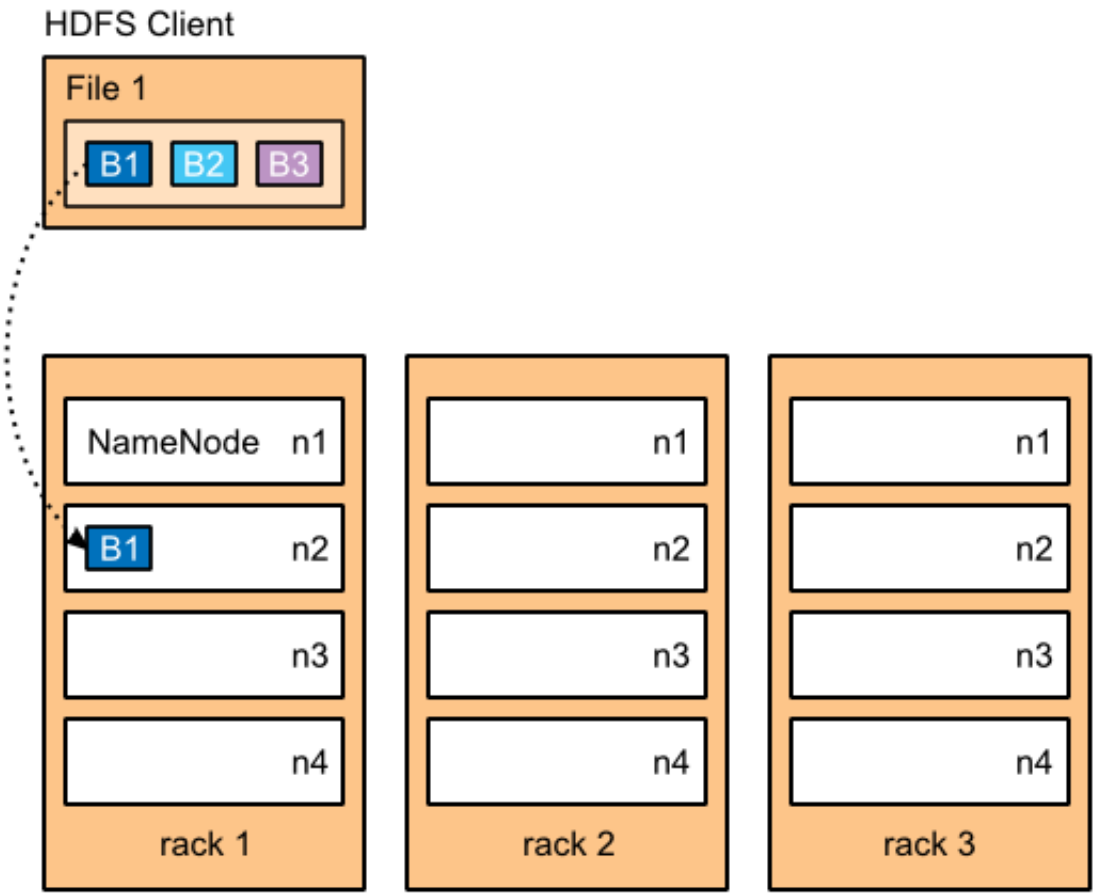
Writing a file to HDFS



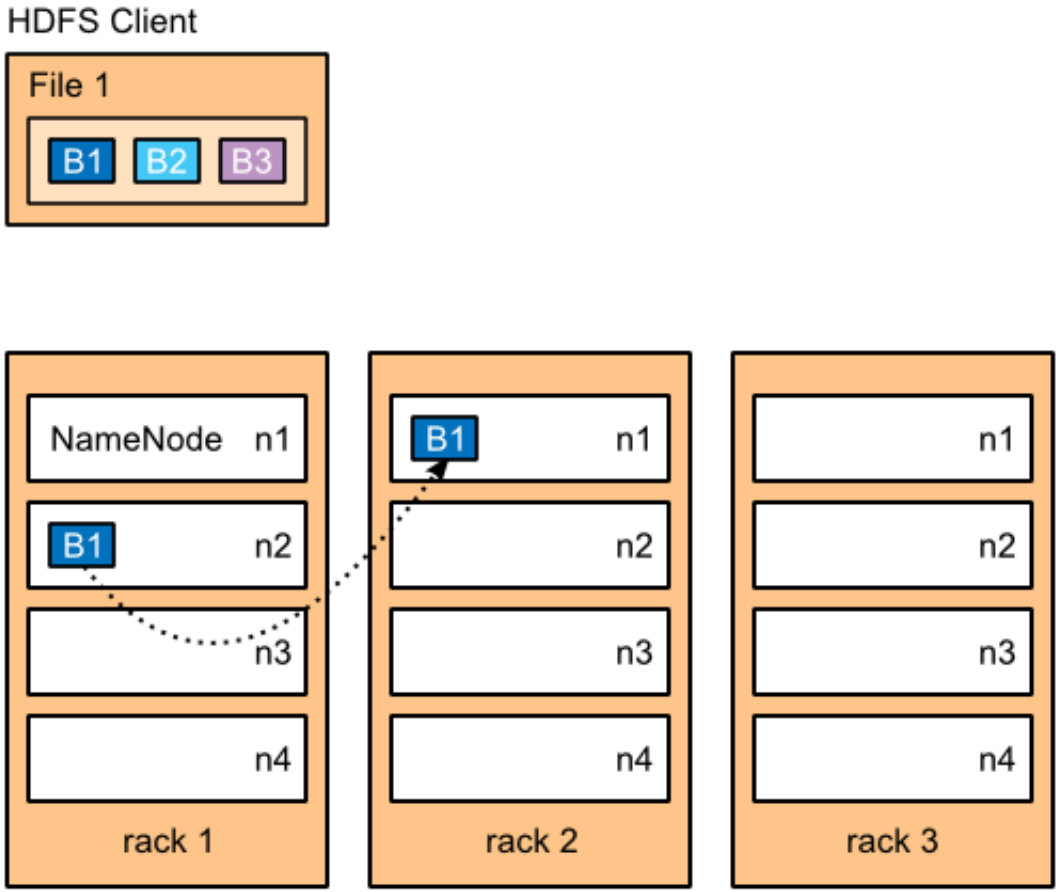
Writing a file to HDFS



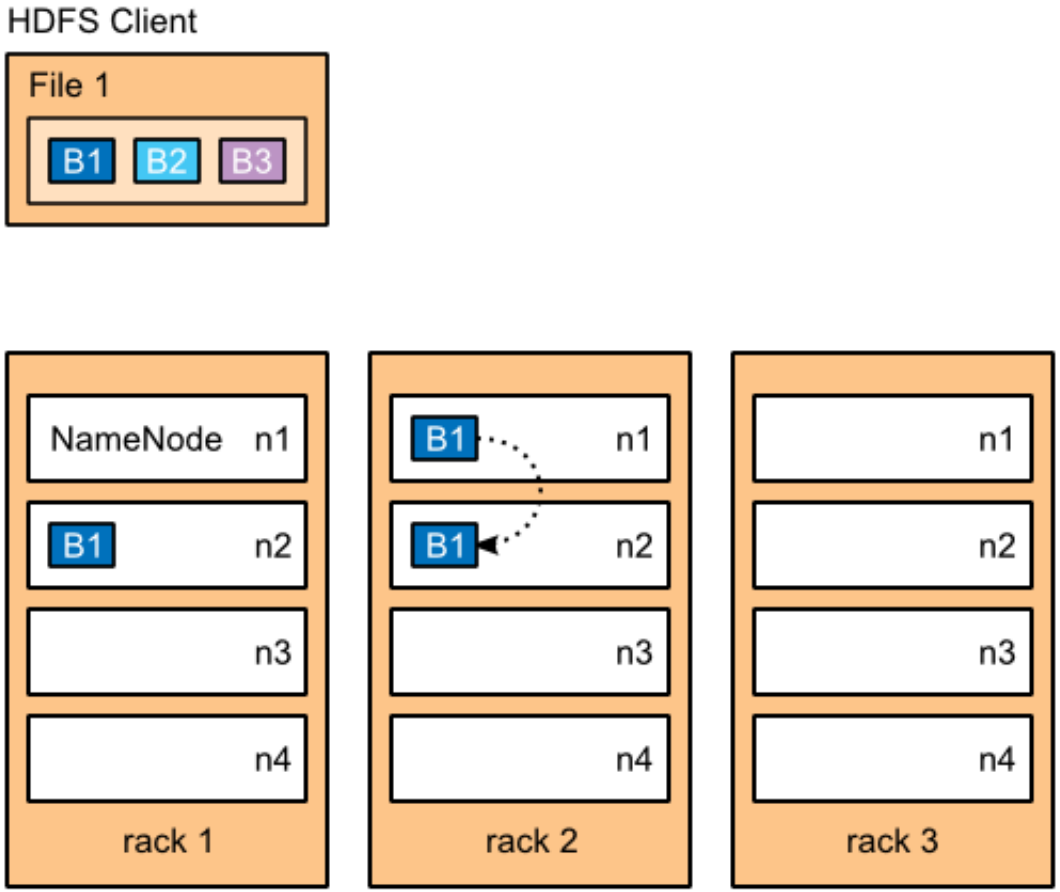
Writing a file to HDFS



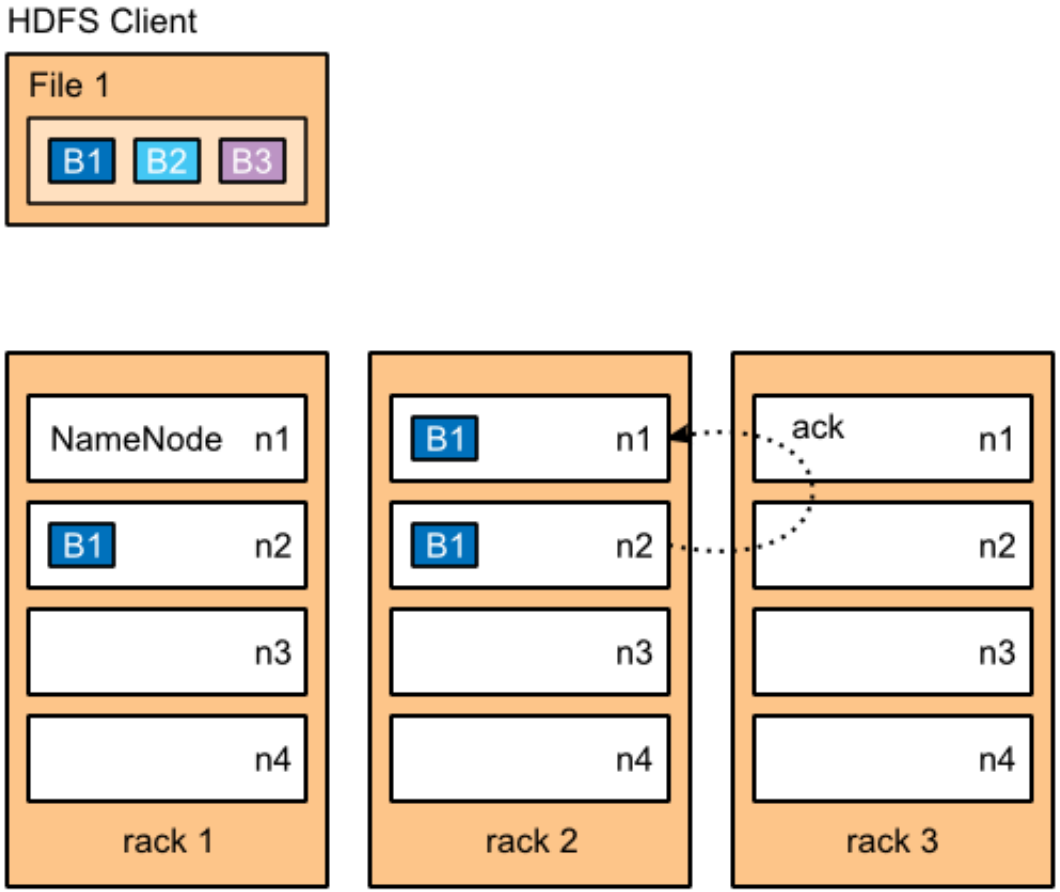
Writing a file to HDFS



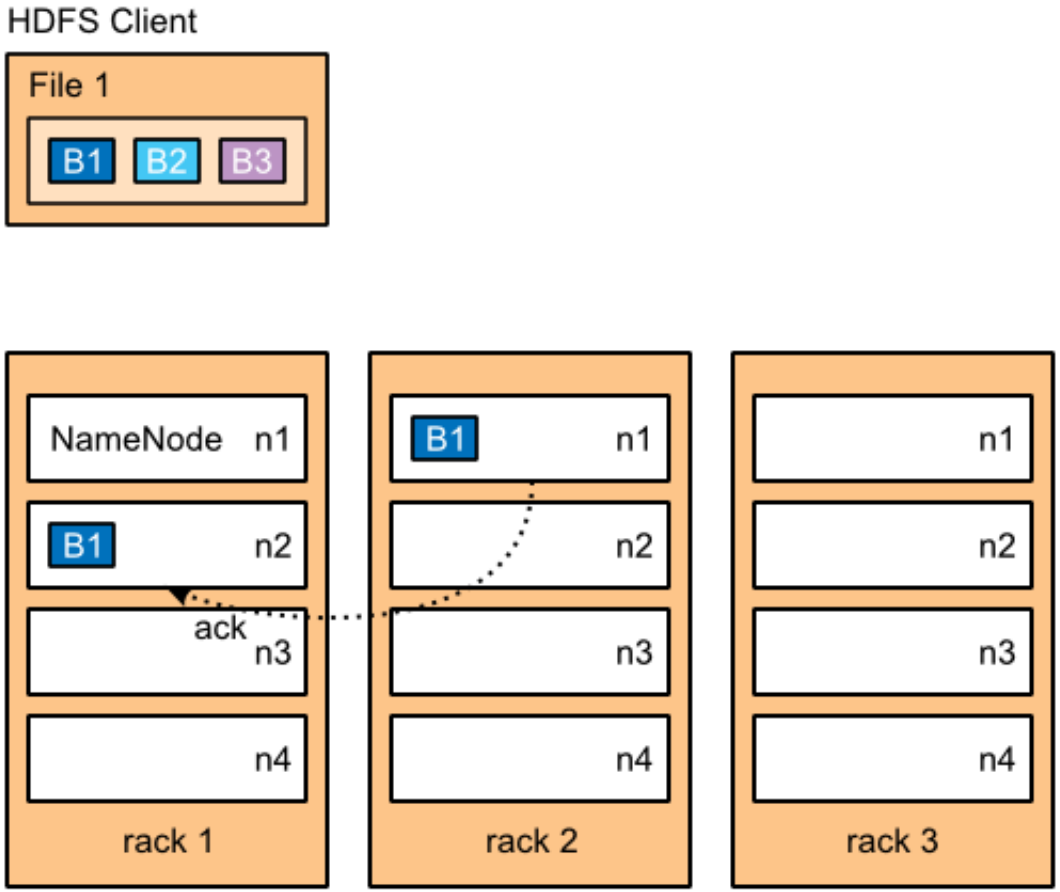
Writing a file to HDFS



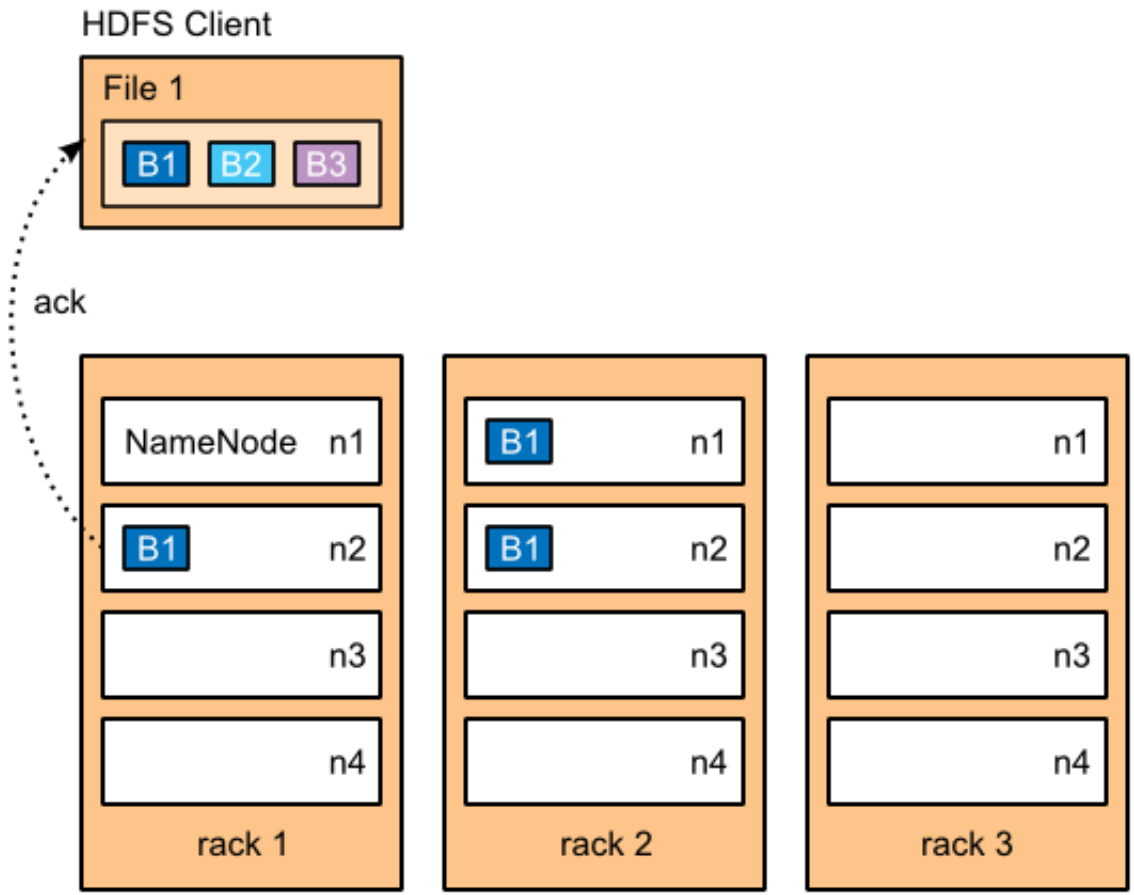
Writing a file to HDFS



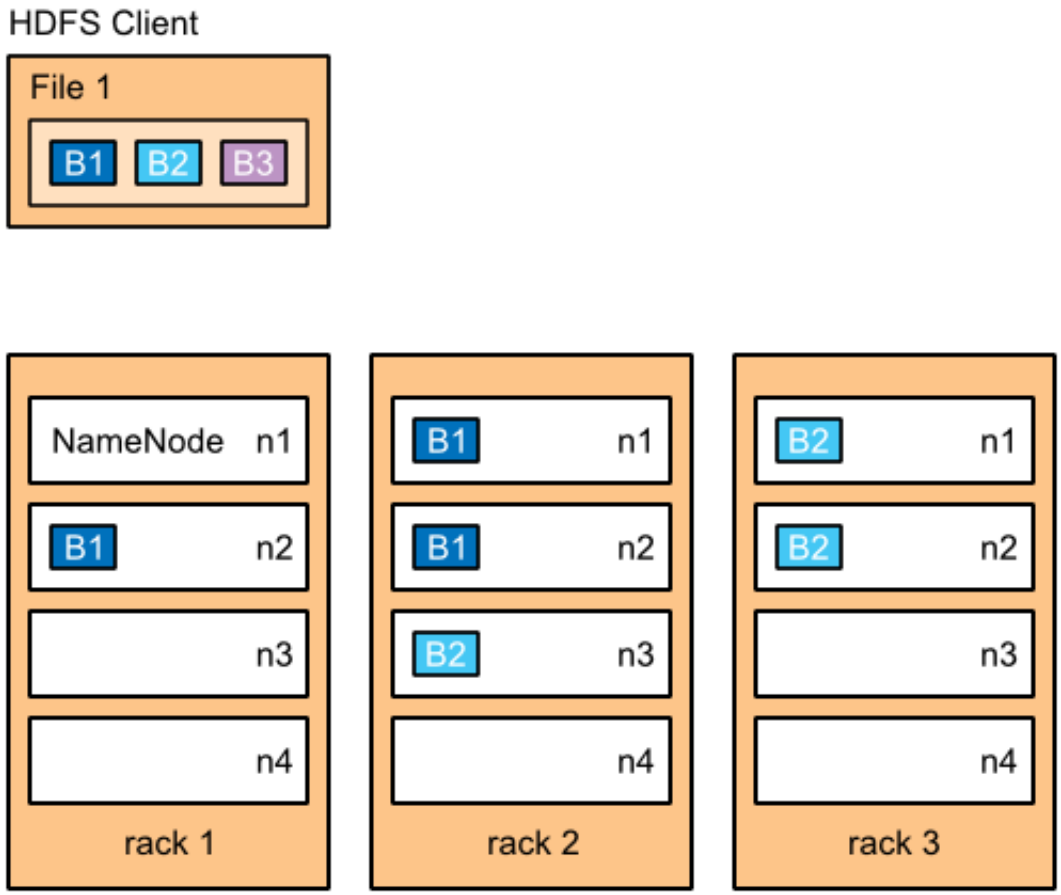
Writing a file to HDFS



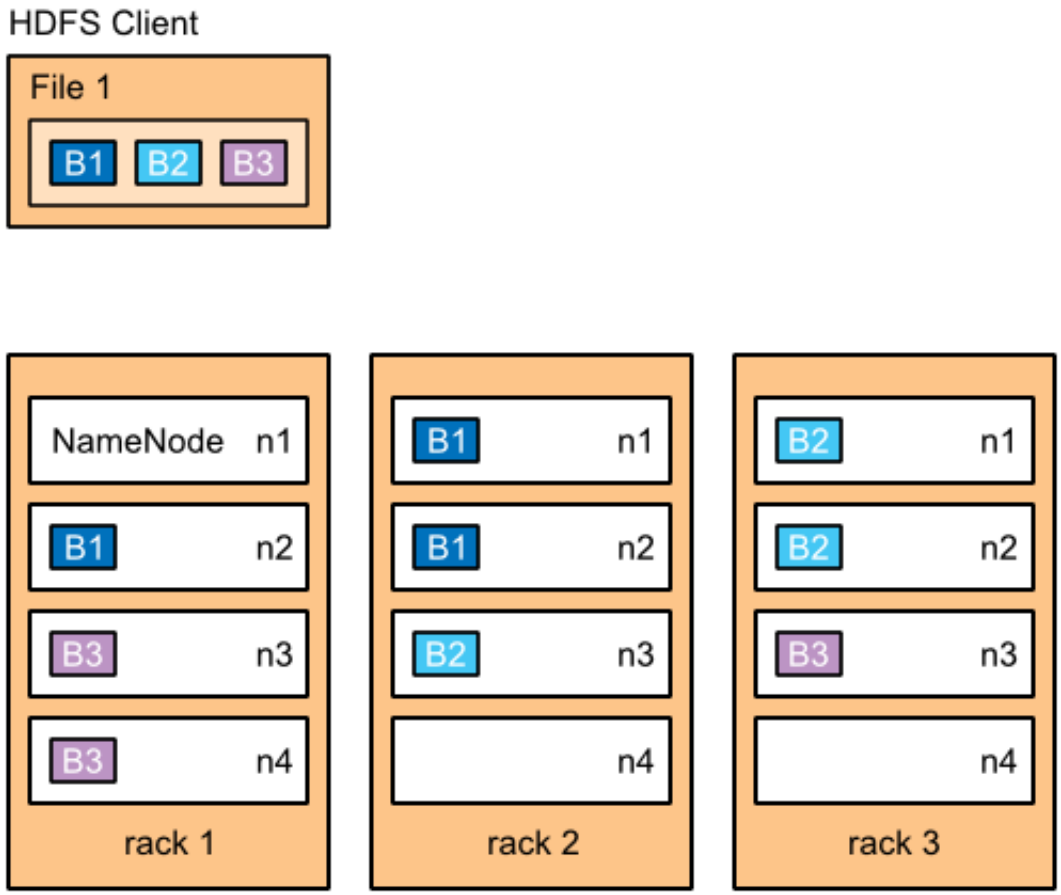
Writing a file to HDFS



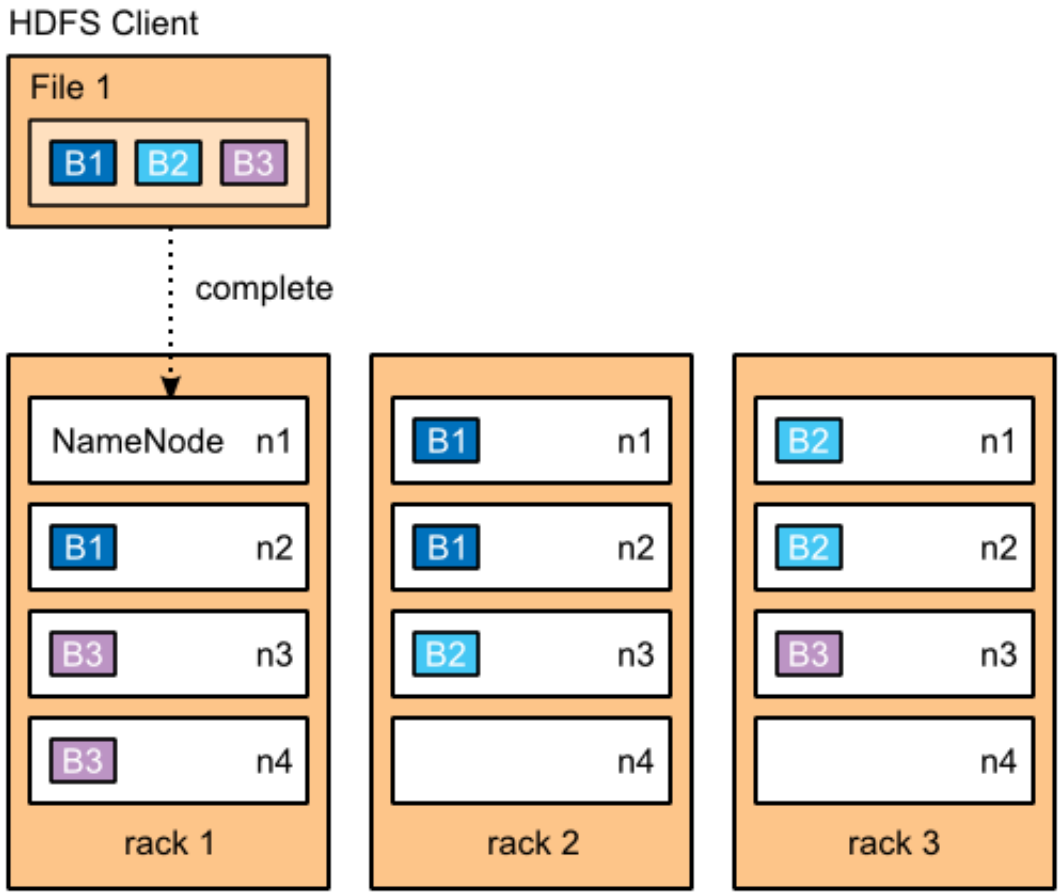
Writing a file to HDFS



Writing a file to HDFS



Writing a file to HDFS



HDFS Command line interface

- **File System Shell (fs)**
  - Invoked as follows:

```
hadoop fs <args>
```

- **Example:**
  - Listing the current directory in hdfs

```
hadoop fs –ls .
```

HDFS Command line interface

- **FS shell commands take paths URIs as argument**
  - URI format: scheme://authority/path
- **Scheme:**
  - For the local filesystem, the scheme is *file*
  - For HDFS, the scheme is *hdfs*

```
hadoop fs –copyFromLocal
        file://myfile.txt
        hdfs://localhost/user/keith/myfile.txt
```

- **Scheme and authority are optional**
  - Defaults are taken from configuration file core-site.xml

HDFS Command line interface

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

HDFS – Specific commands

- **copyFromLocal / put**
  - Copy files from the local file system into fs

```
hadoop fs -copyFromLocal <localsrc> .. <dst>
```

Or

```
hadoop fs -put <localsrc> .. <dst>
```

HDFS – Specific commands

- **copyToLocal / get**
  - Copy files from fs into the local file system

```
hadoop fs -copyToLocal [-ignorecrc] [-crc]
                                    <src> <localdst>
```

Or

```
hadoop fs -get [-ignorecrc] [-crc]
                                    <src> <localdst>
```

HDFS – Specific commands

- **getMerge**
  - Get all the files in the directories that match the source file pattern
  - Merge and sort them to only one file on local fs
  - <src> is kept

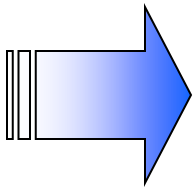
hadoop fs -getmerge <src> <localdst>

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 achieved.

```
hadoop fs -setrep [-R] [-w] <rep> <path/file>
```

Agenda

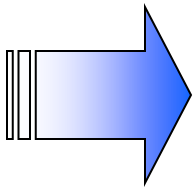


- Terminology review
- HDFS
- MapReduce
- Type of nodes
- Topology awareness

MapReduce engine

- Technology from Google
- A MapReduce program consists of map and reduce functions
- A MapReduce job is broken into tasks that run in parallel

Agenda

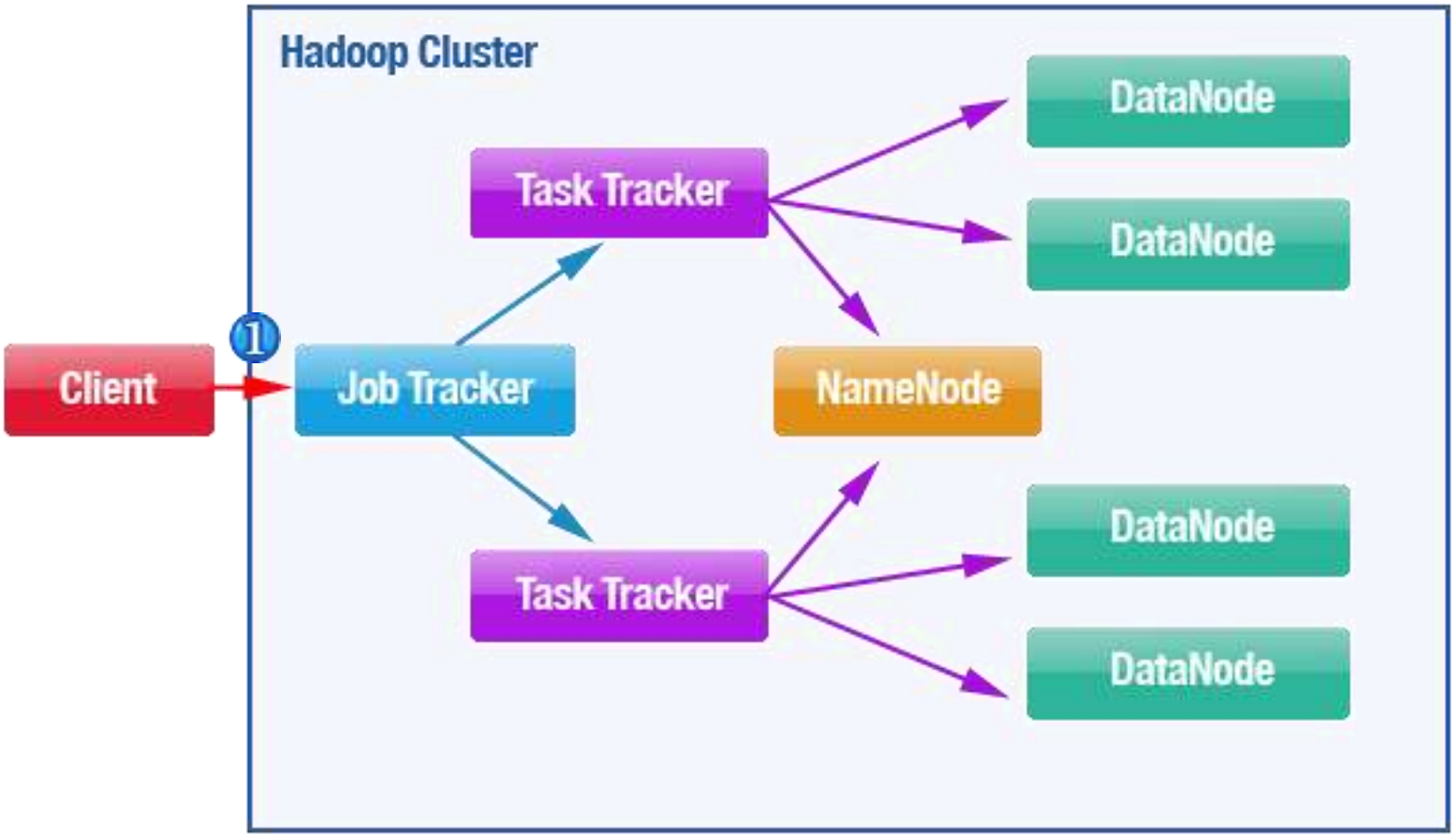


- Terminology review
- HDFS
- MapReduce
- Type of nodes
- Topology awareness

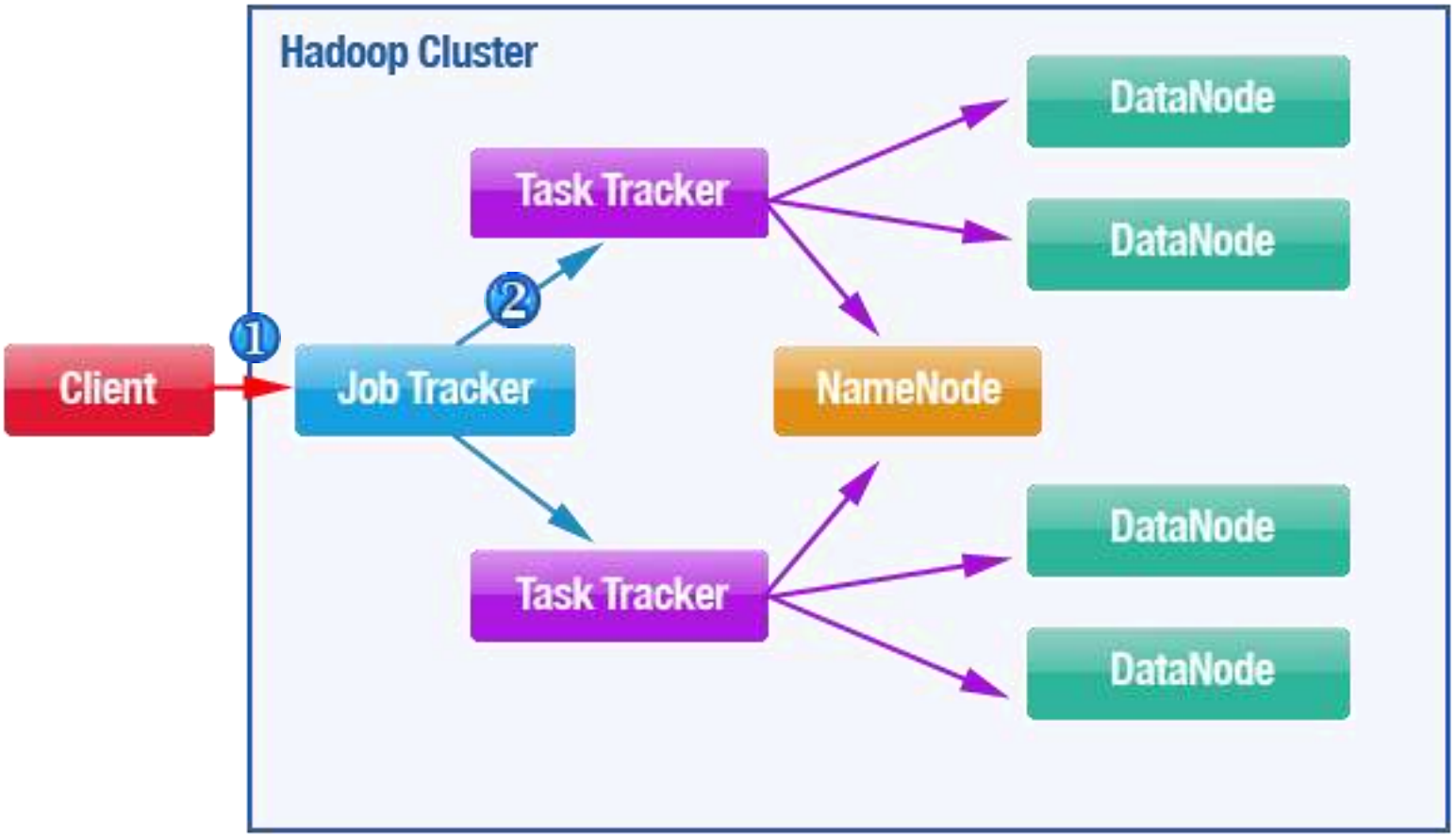
Types of nodes - Overview

- **HDFS nodes**
  - NameNode
  - DataNode
- **MapReduce nodes**
  - JobTracker
  - TaskTracker
- **There are other nodes not discussed in this course**

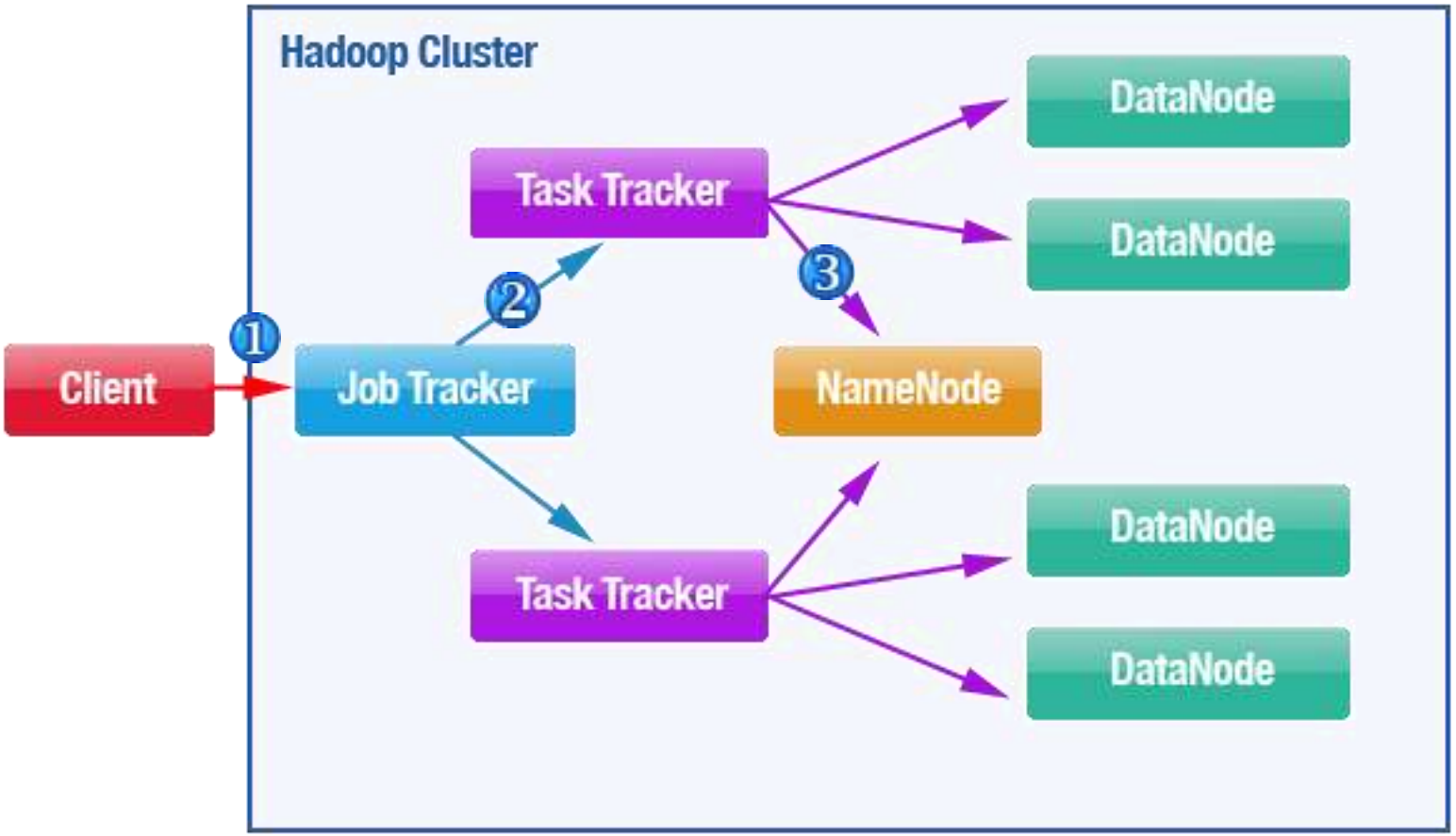
Types of nodes - Overview



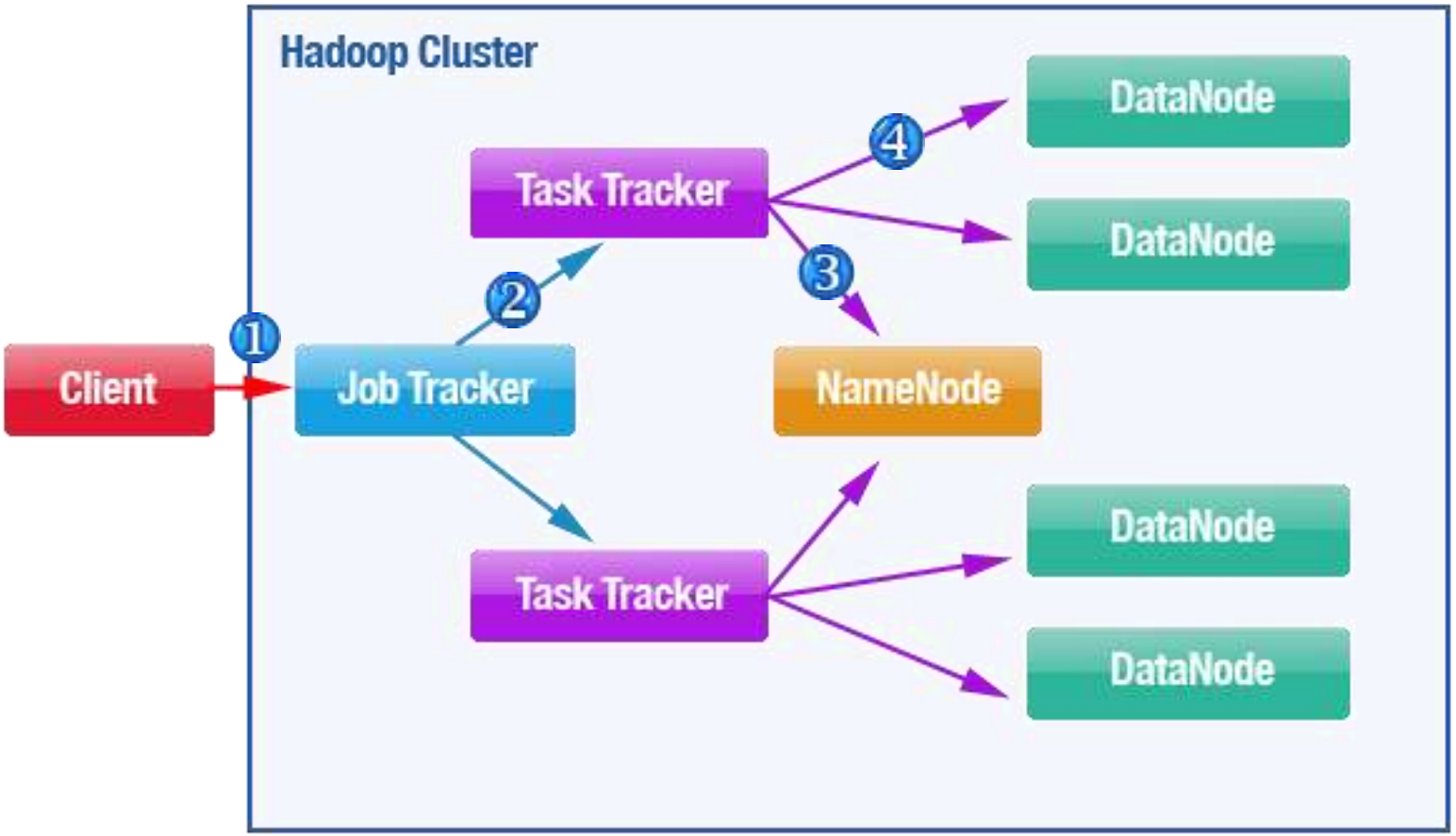
Types of nodes - Overview



Types of nodes - Overview

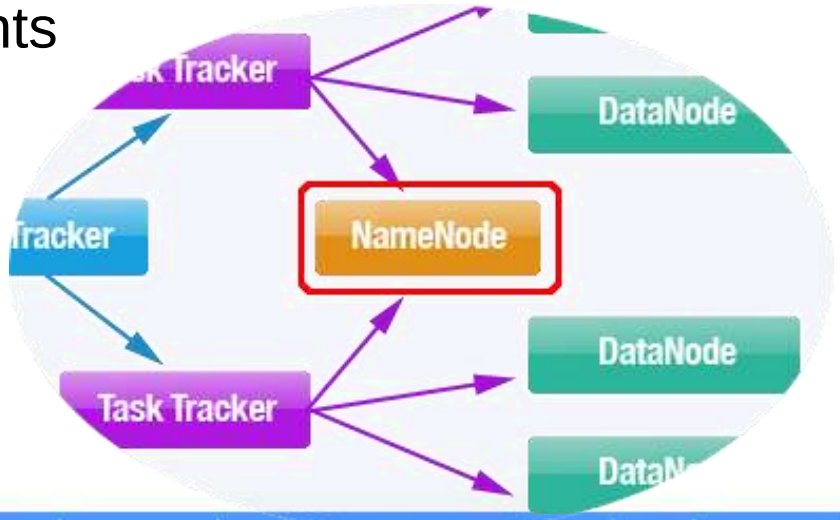


Types of nodes - Overview



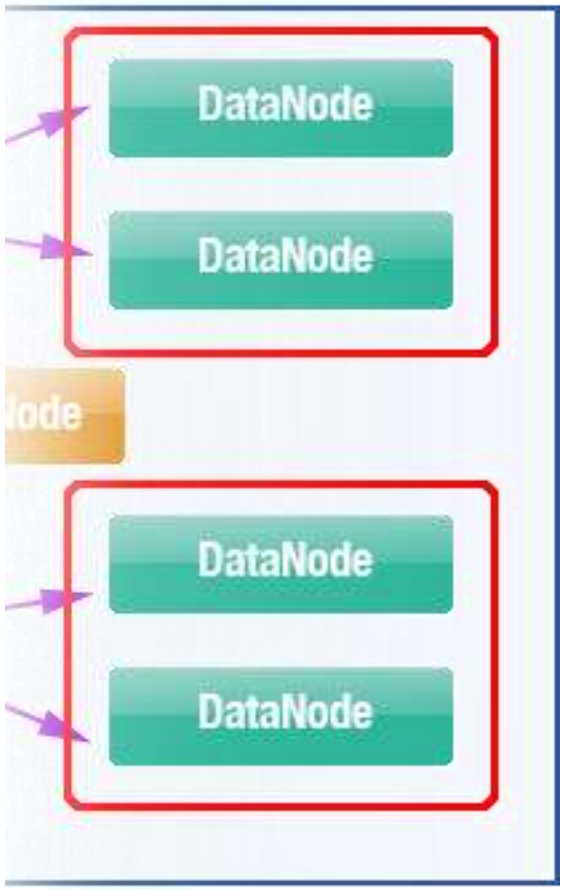
Types of nodes - NameNode

- **NameNode**
  - Only one per Hadoop cluster
  - Manages the filesystem namespace and metadata
  - Single point of failure, but mitigated by writing state to multiple filesystems
  - Single point of failure: Don't use inexpensive commodity hardware for this node, large memory requirements



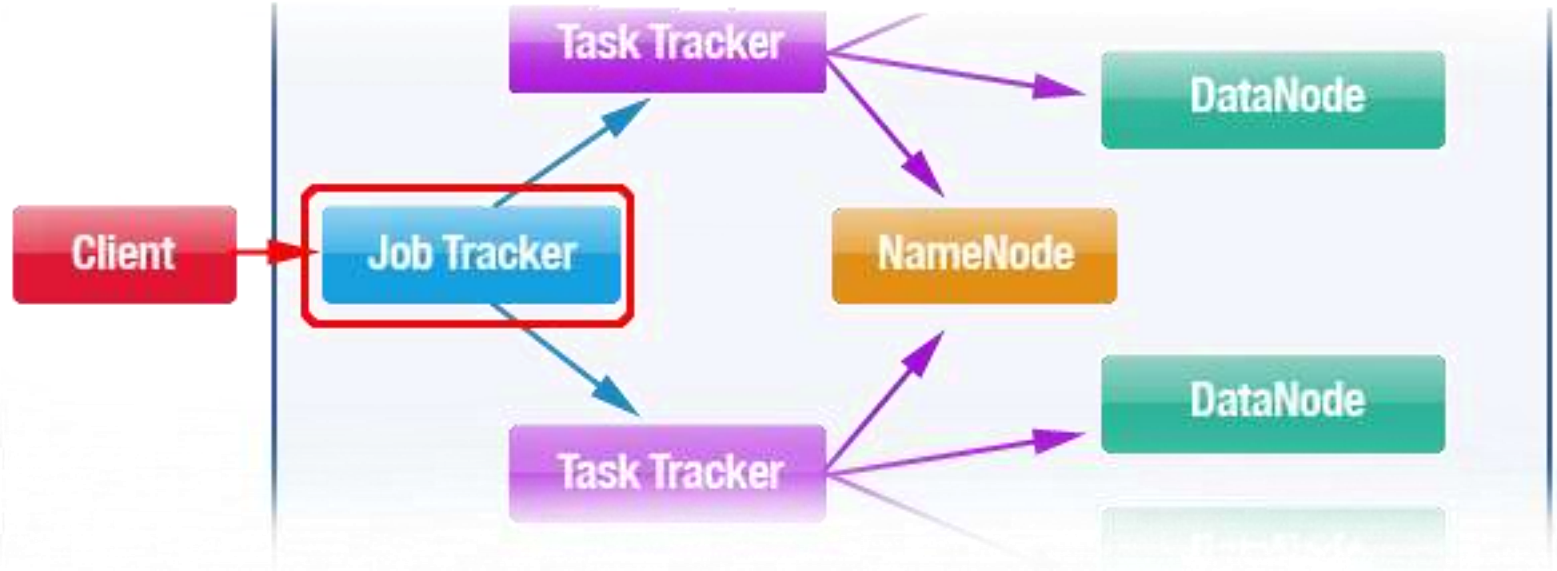
Types of nodes - DataNode

- **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



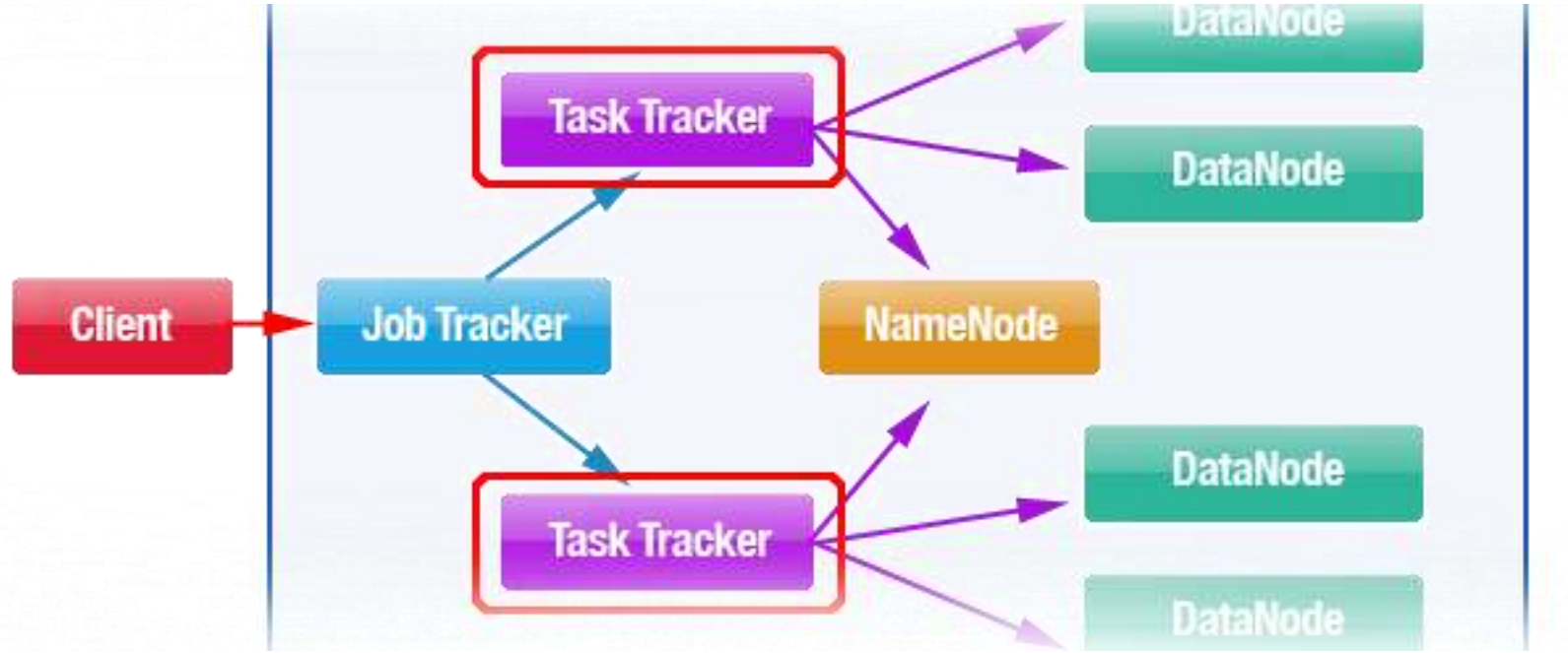
Types of nodes - JobTracker

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



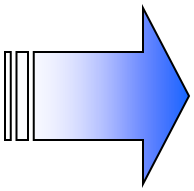
Types of nodes - TaskTracker

- **TaskTracker node**
  - Many per Hadoop cluster
  - Executes MapReduce operations
  - Reads blocks from DataNodes



Agenda

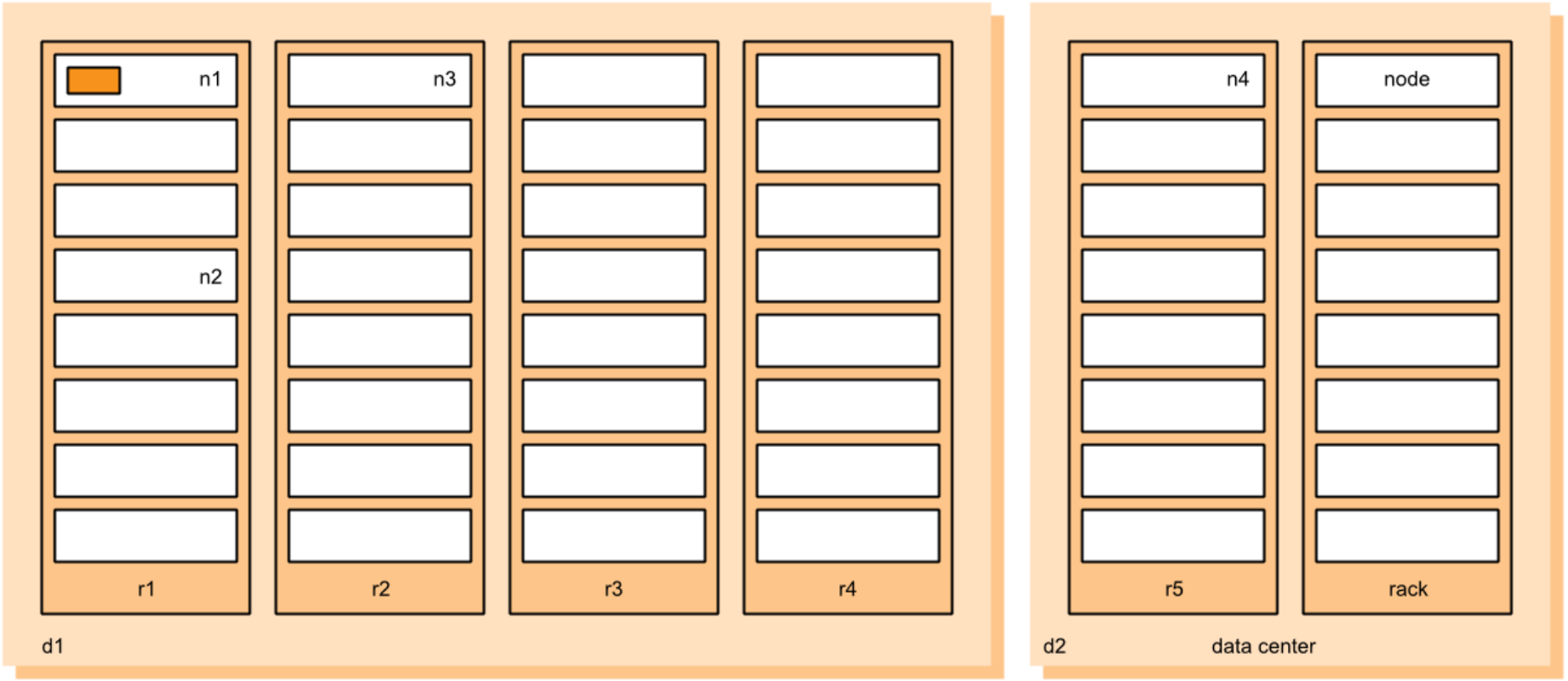
- Terminology review
- HDFS
- MapReduce
- Type of nodes



- Topology awareness

Topology awareness (or Rack awareness)

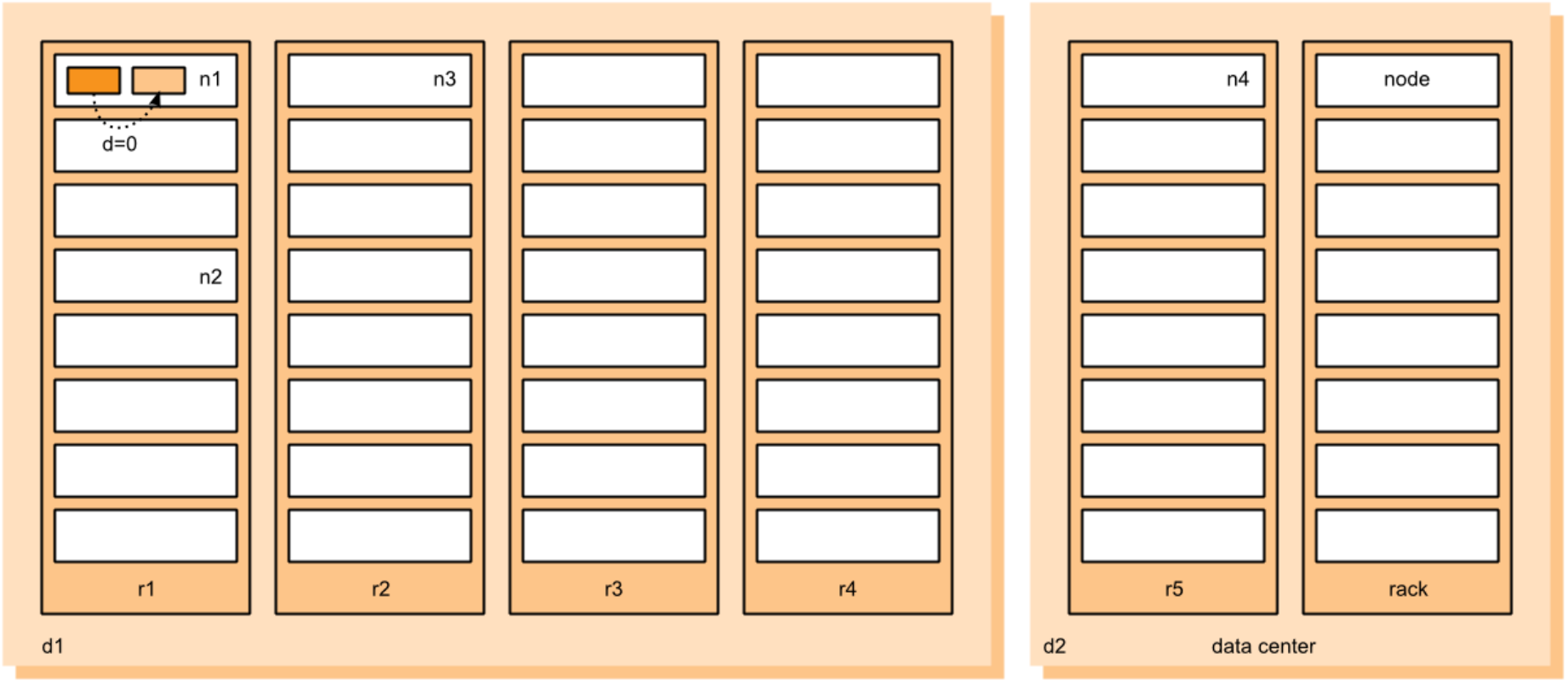
Bandwidth becomes progressively smaller in the following scenarios:



Topology awareness

Bandwidth becomes progressively smaller in the following scenarios:

1.Process on the same node.

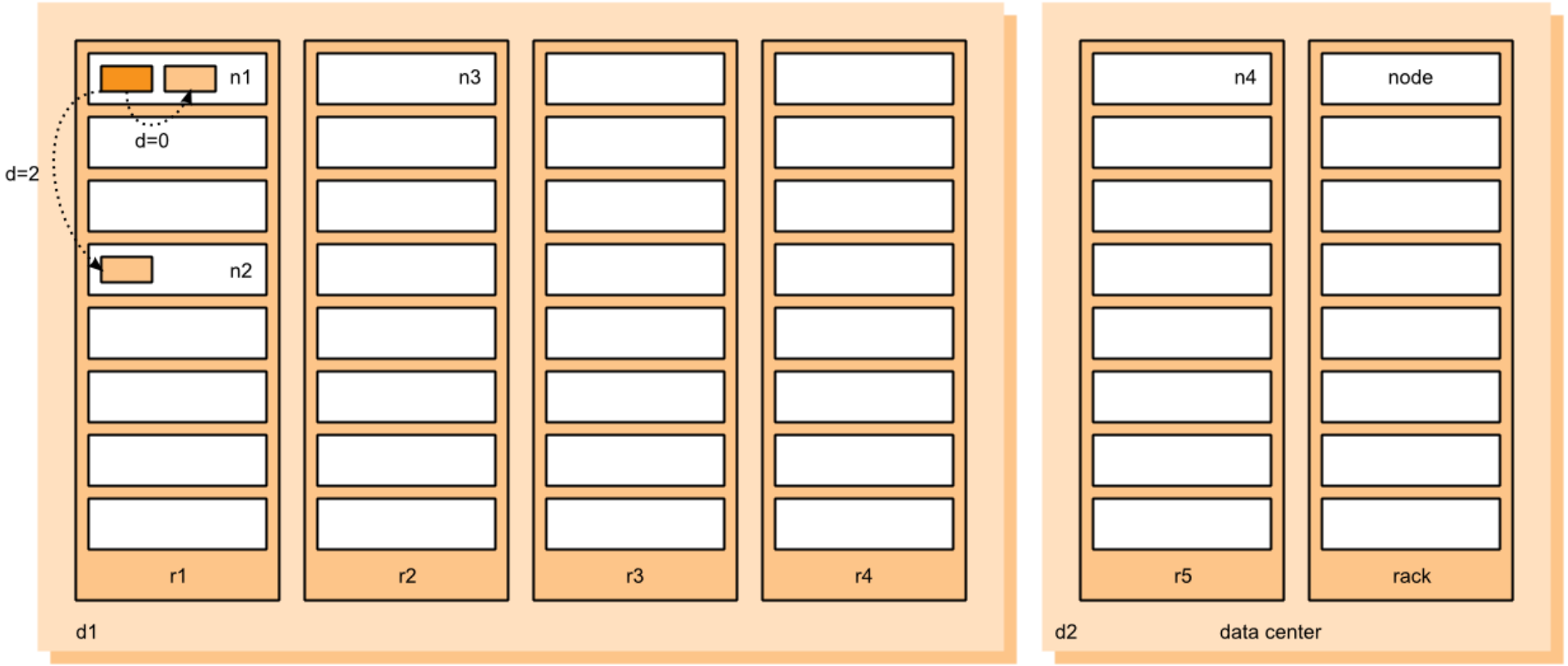


Topology awareness

Bandwidth becomes progressively smaller in the following scenarios:

1.Process on the same node

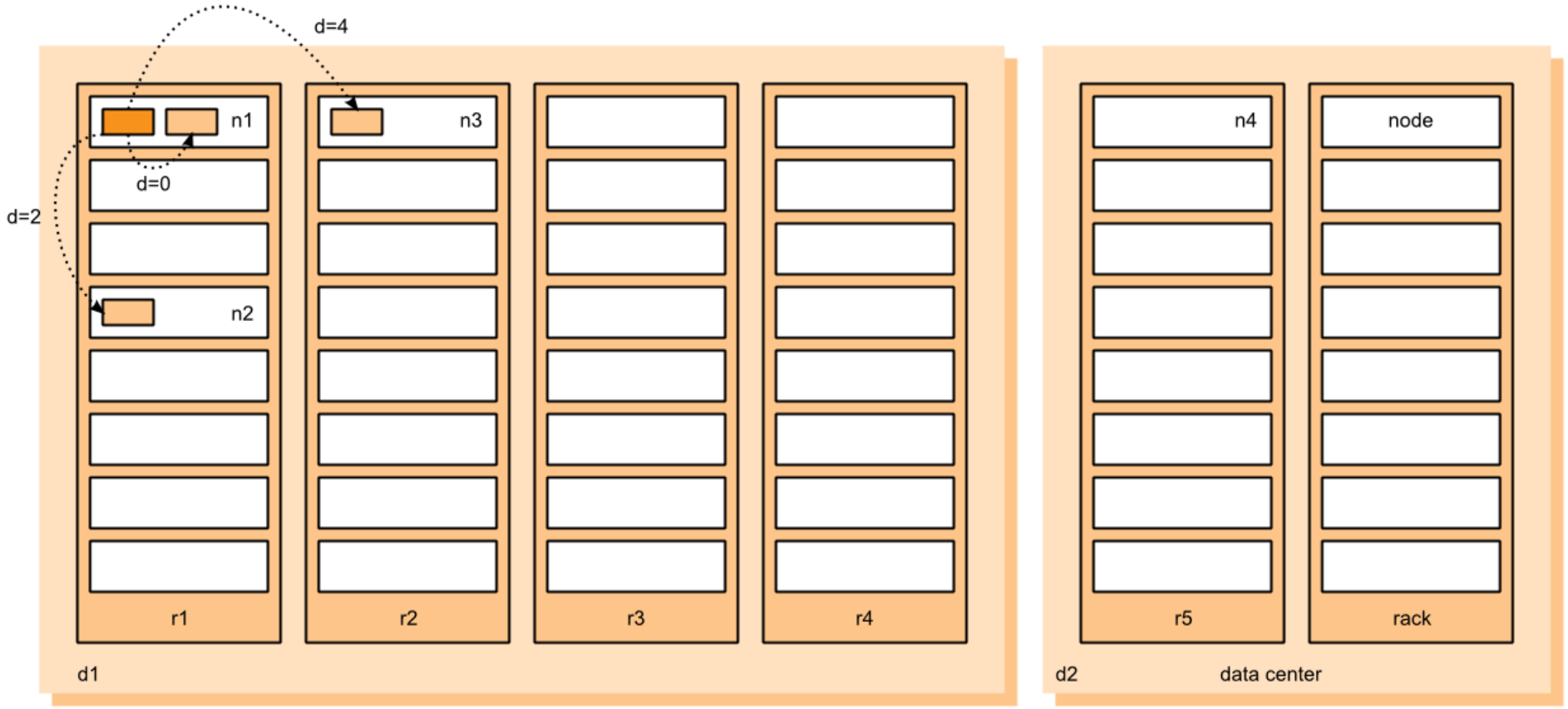
2.Different nodes on the same rack



Topology awareness

Bandwidth becomes progressively smaller in the following scenarios:

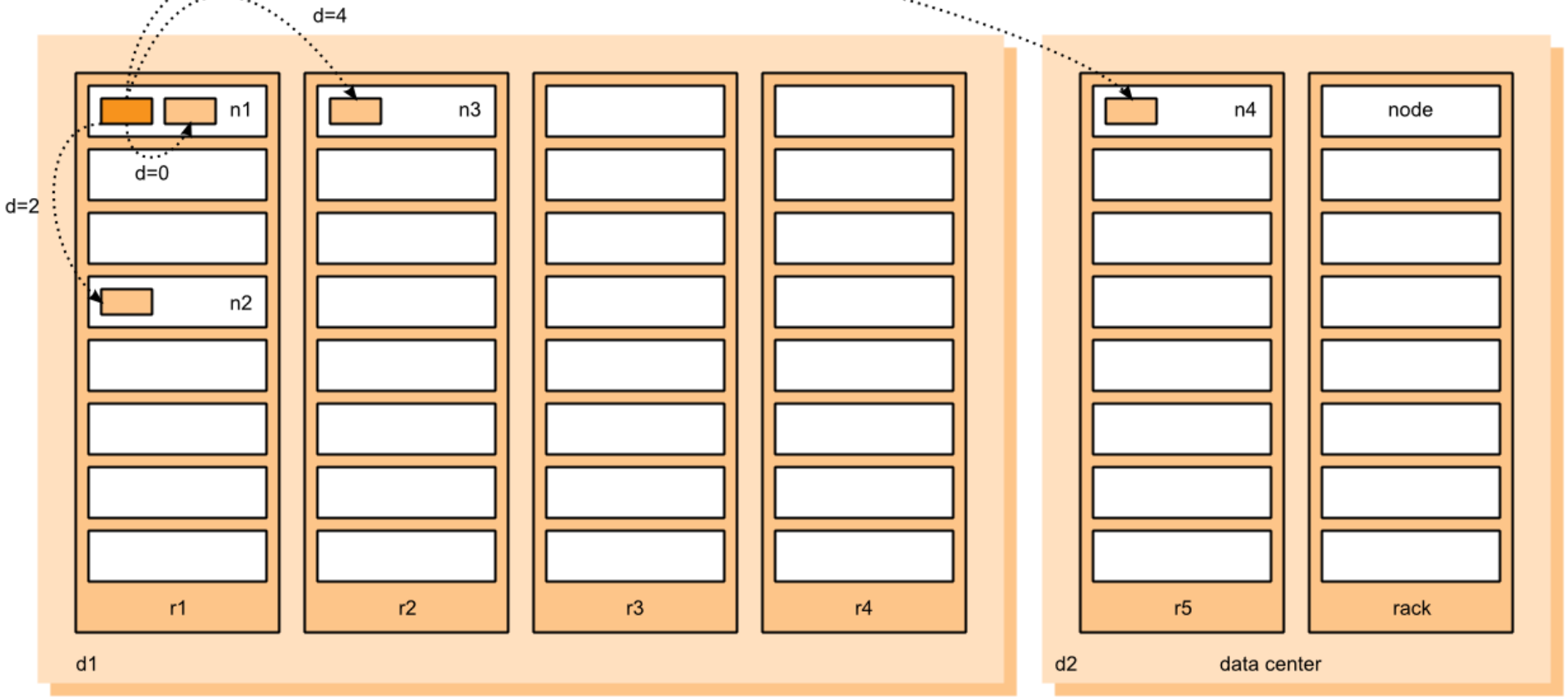
1. Process on the same node
2. Different nodes on the same rack
3. Nodes on different racks in the same data center



Topology awareness

Bandwidth becomes progressively smaller in the following scenarios:

1. Process on the same node
2. Different nodes on the same rack
3. Nodes on different racks in the same data center
4. Nodes in different data centers



Thank you!