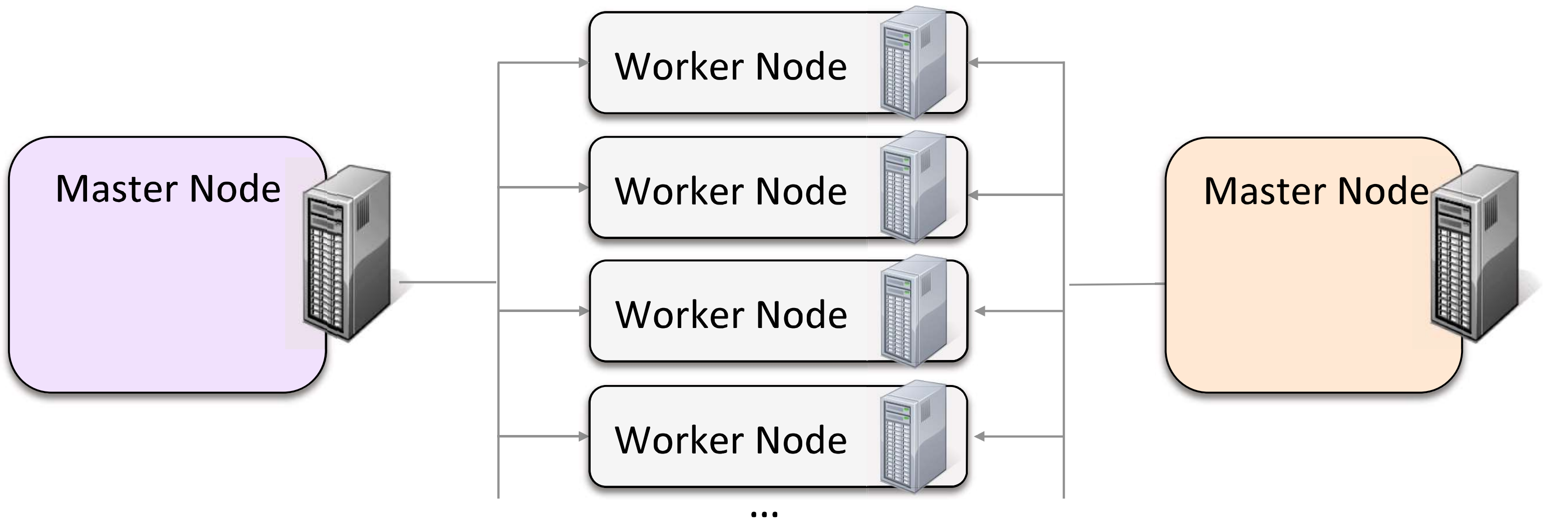
Hadoop Architecture and HDFS

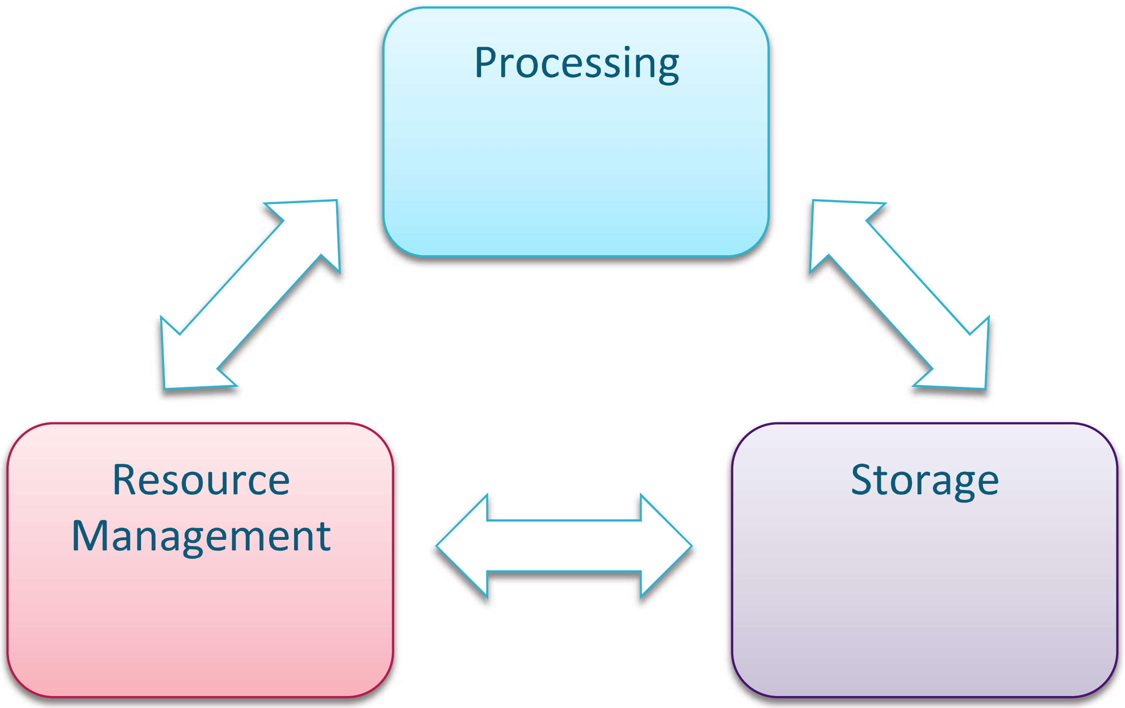
Hadoop Cluster Terminology

- **A *cluster* is a group of computers working together**
  - Provides data storage, data processing, and resource management
- **A *node* is an individual computer in the cluster**
  - *Master* nodes manage distribution of work and data to *worker* nodes
- **A *daemon* is a program running on a node**
  - Each Hadoop daemon performs a specific function in the cluster



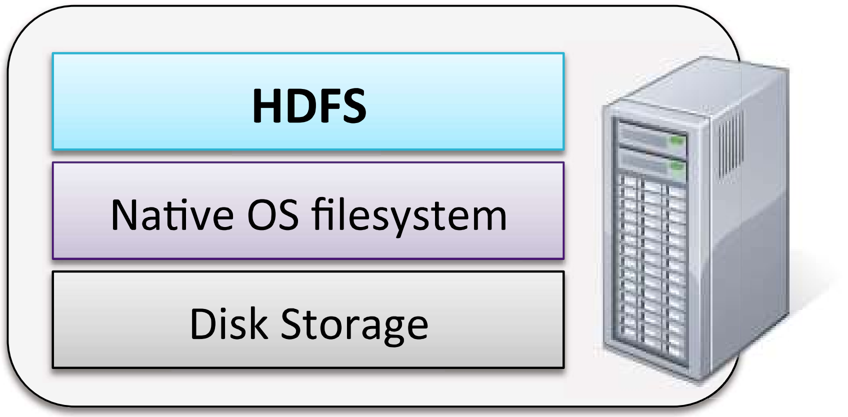
Cluster Components

- **Three main components of a cluster**
- **Work together to provide distributed data processing**
- **We will start with the Storage component**
  - HDFS



HDFS Basic Concepts (1)

- **HDFS is a filesystem written in Java**
  - Based on Google's GFS
- **Sits on top of a native filesystem**
  - Such as ext3, ext4, or xfs
- **Provides redundant storage for massive amounts of data**
  - Using readily-available, industry-standard computers

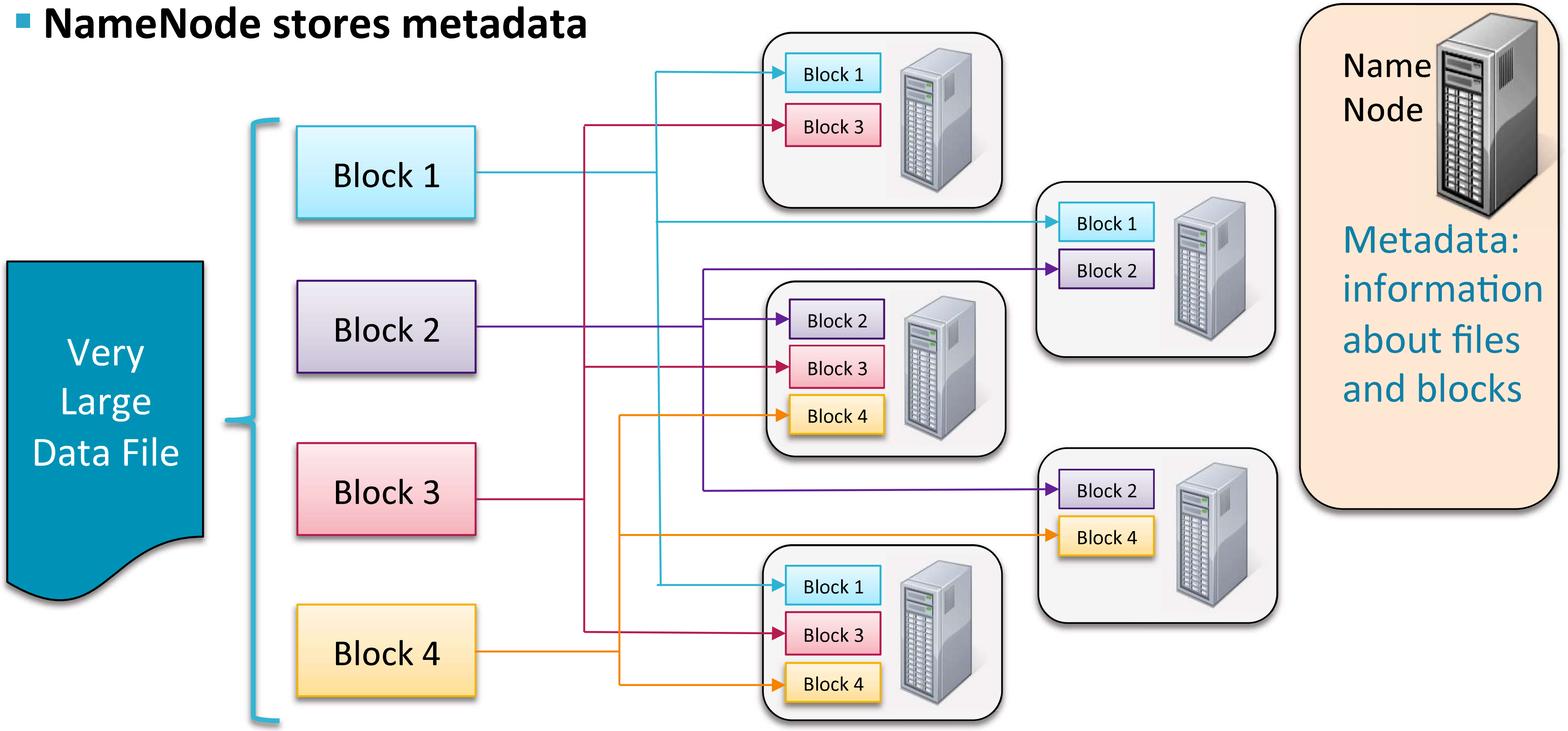


HDFS Basic Concepts (2)

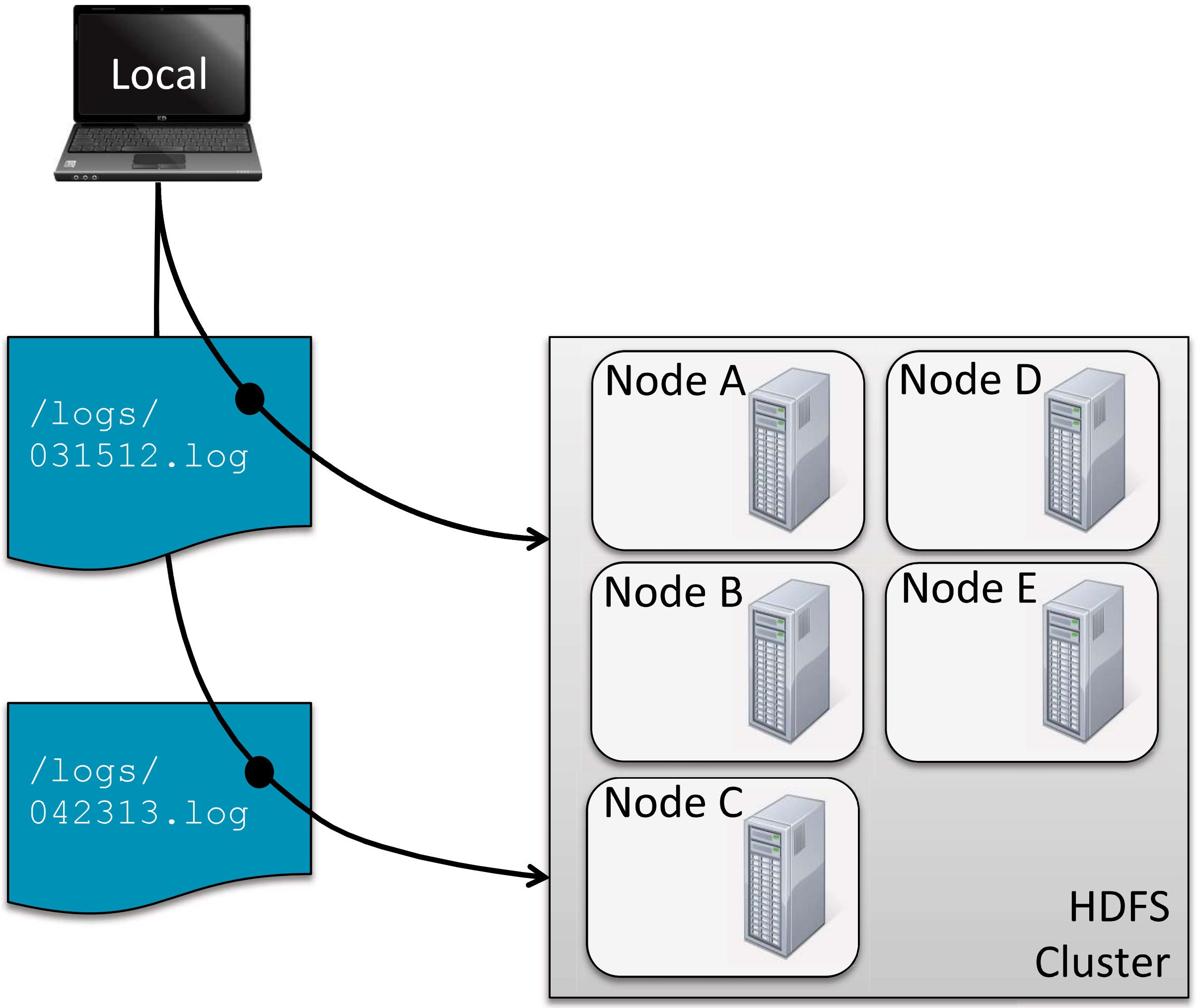
- **HDFS performs best with a ‘modest’ number of large files**
  - Millions, rather than billions, of files
  - Each file typically 100MB or more
- **Files in HDFS are ‘write once’**
  - No random writes to files are allowed
- **HDFS is optimized for large, streaming reads of files**
  - Rather than random reads

How Files Are Stored

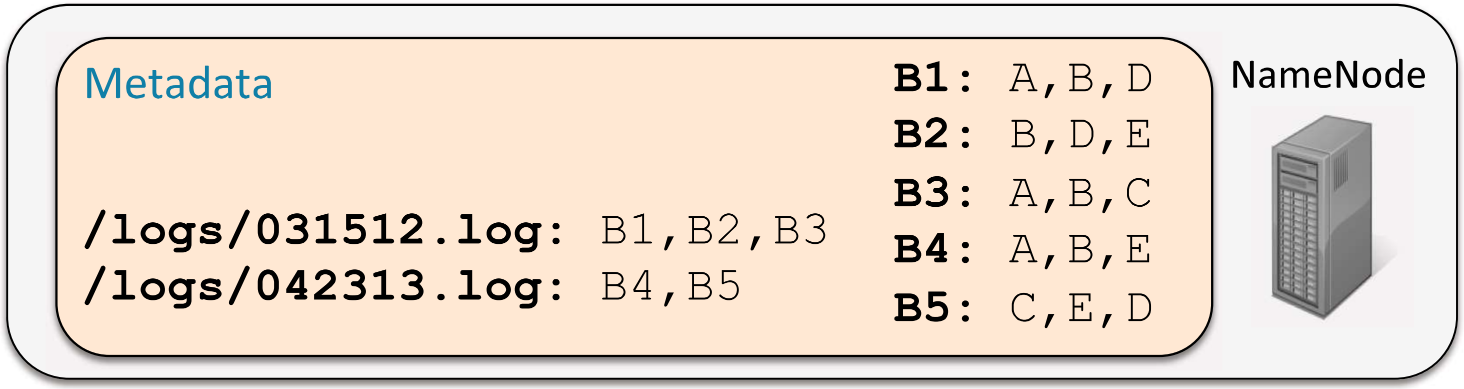
- **Data files are split into 128MB blocks which are distributed at load time**
- **Each block is replicated on multiple data nodes (default 3x)**
- **NameNode stores metadata**



Example: Storing and Retrieving Files (1)

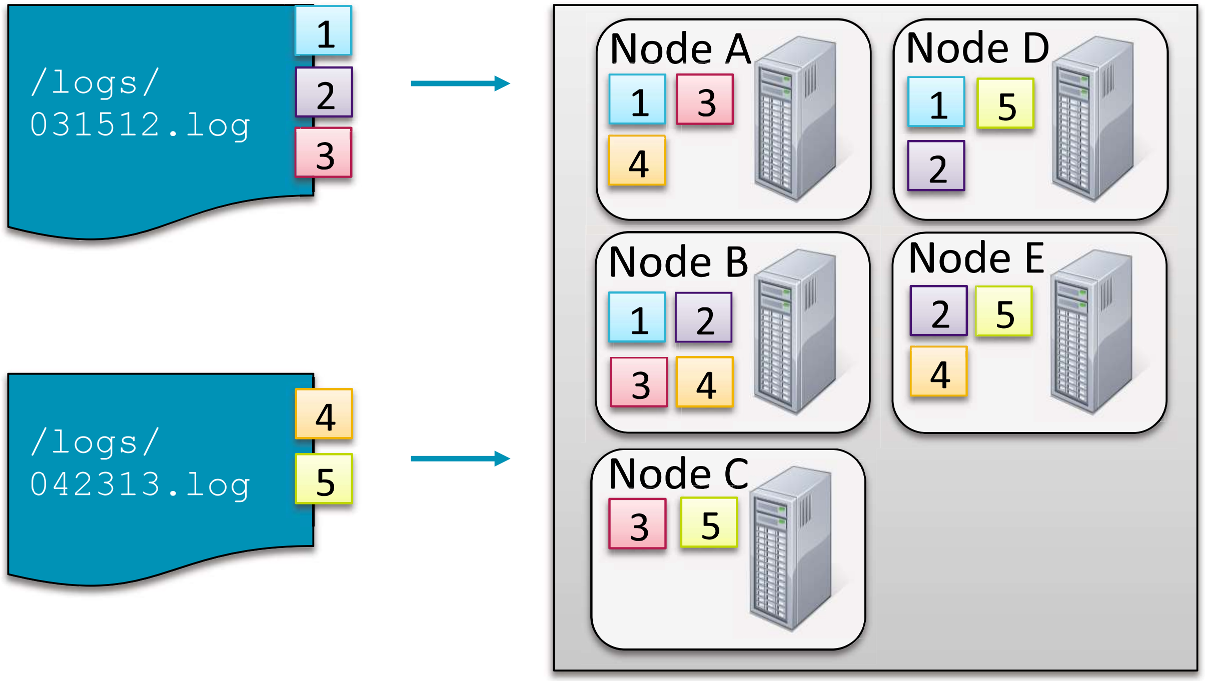


Example: Storing and Retrieving Files (2)

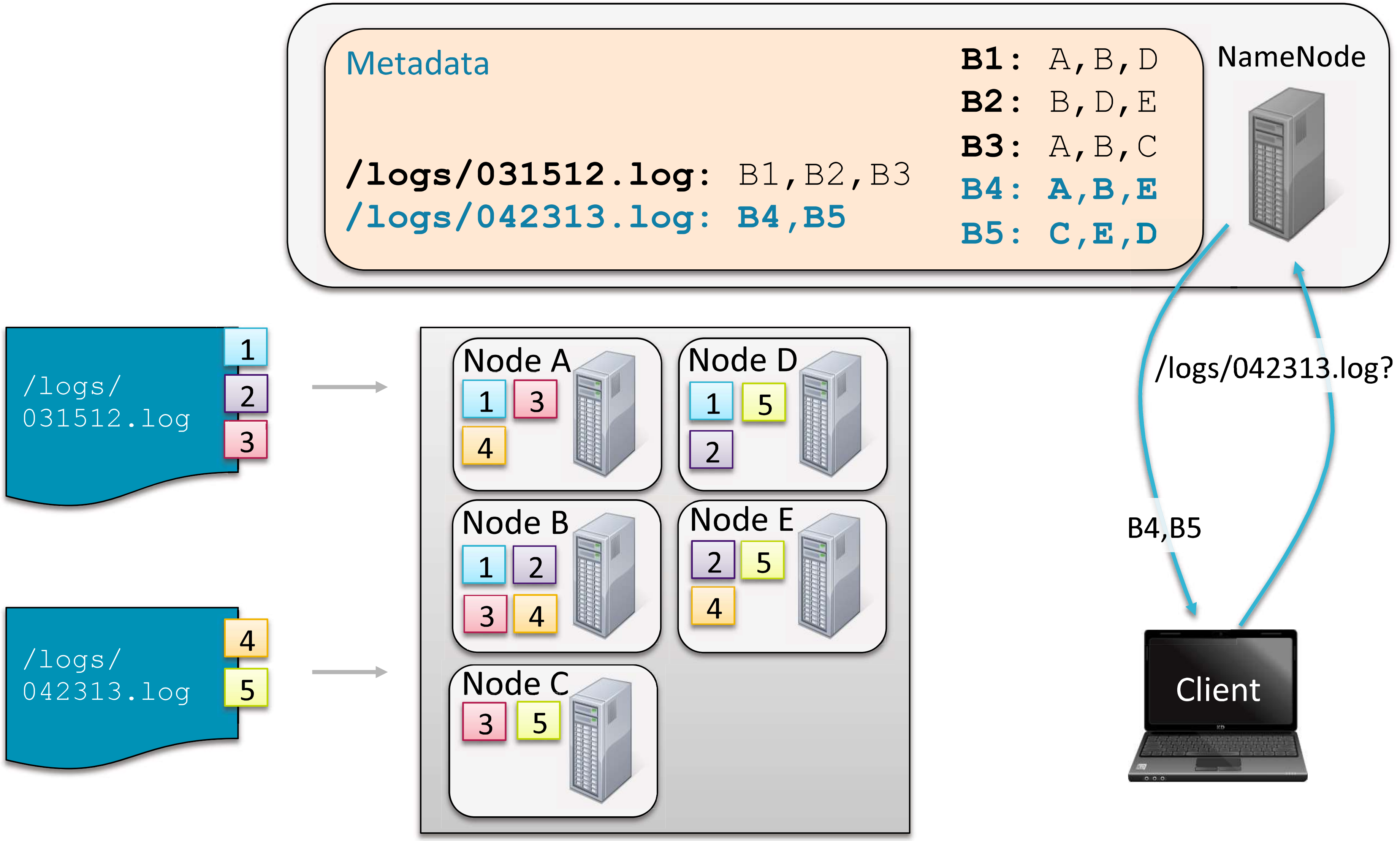


```
/logs/031512.log: B1,B2,B3
/logs/042313.log: B4,B5
```

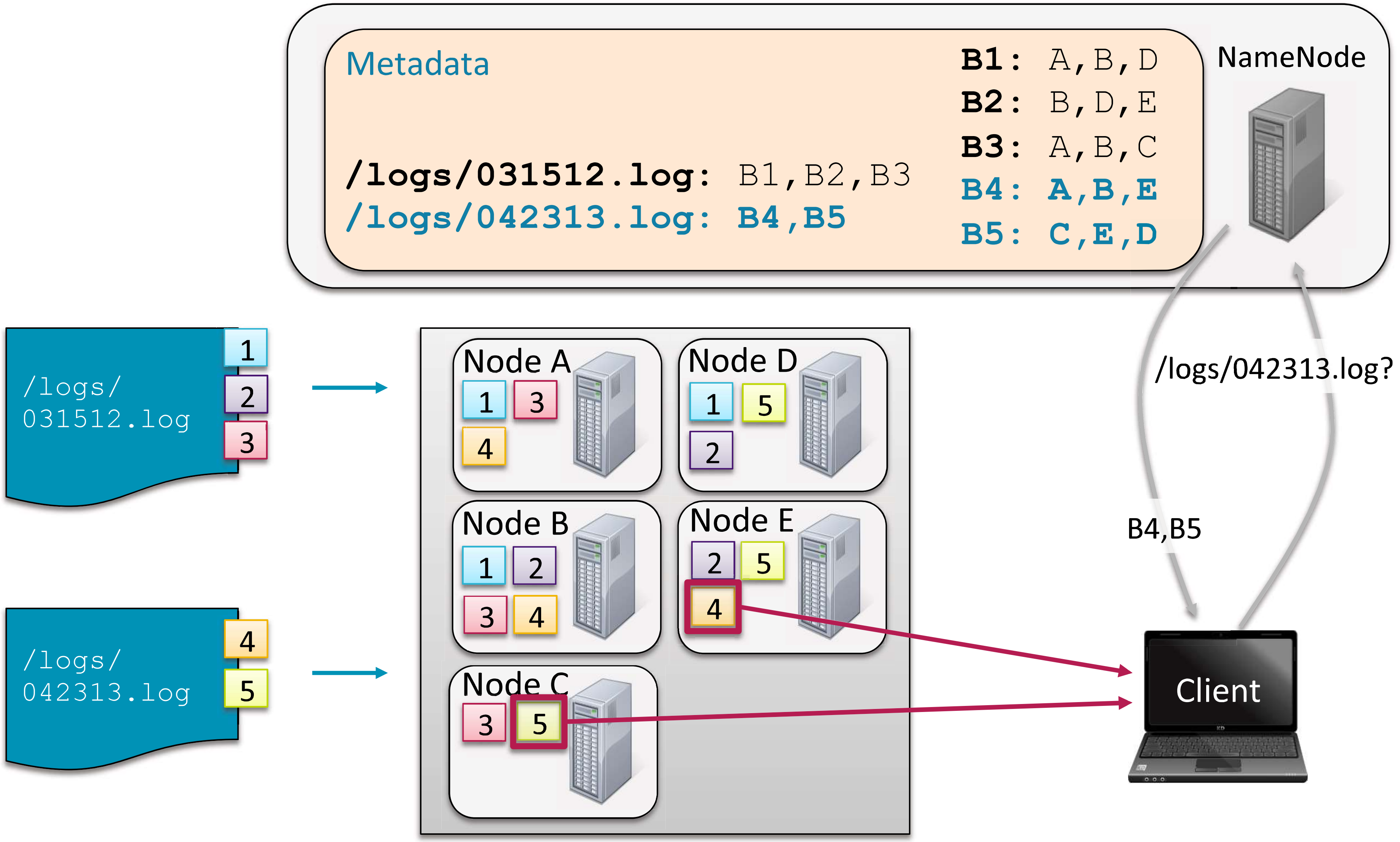
```
B1: A,B,D
B2: B,D,E
B3: A,B,C
B4: A,B,E
B5: C,E,D
```



Example: Storing and Retrieving Files (3)

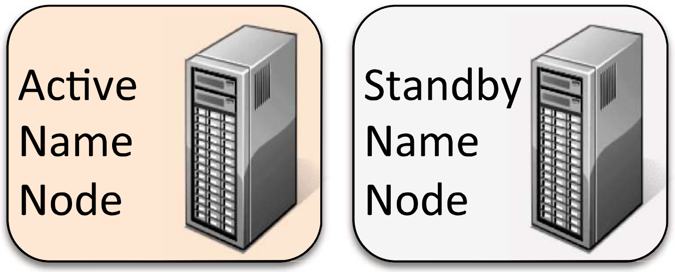


Example: Storing and Retrieving Files (4)



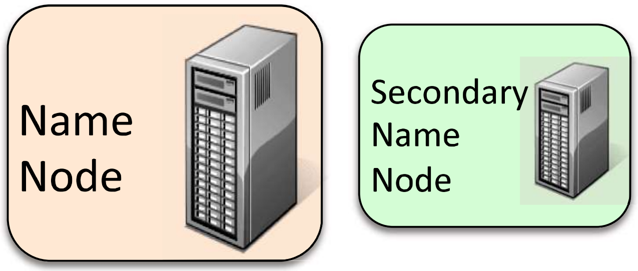
HDFS NameNode Availability

- **The NameNode daemon must be running at all times**
  - If the NameNode stops, the cluster becomes inaccessible
- **HDFS is typically set up for High Availability**
  - Two NameNodes: Active and Standby



High availability

- **Small clusters may use 'Classic mode'**
  - One NameNode
  - One "helper" node called the Secondary NameNode
    - Bookkeeping, not backup



Classical mode:Not highly available; namenode is

Options for Accessing HDFS

terminal com

- **From the command line**
  - FsShell:

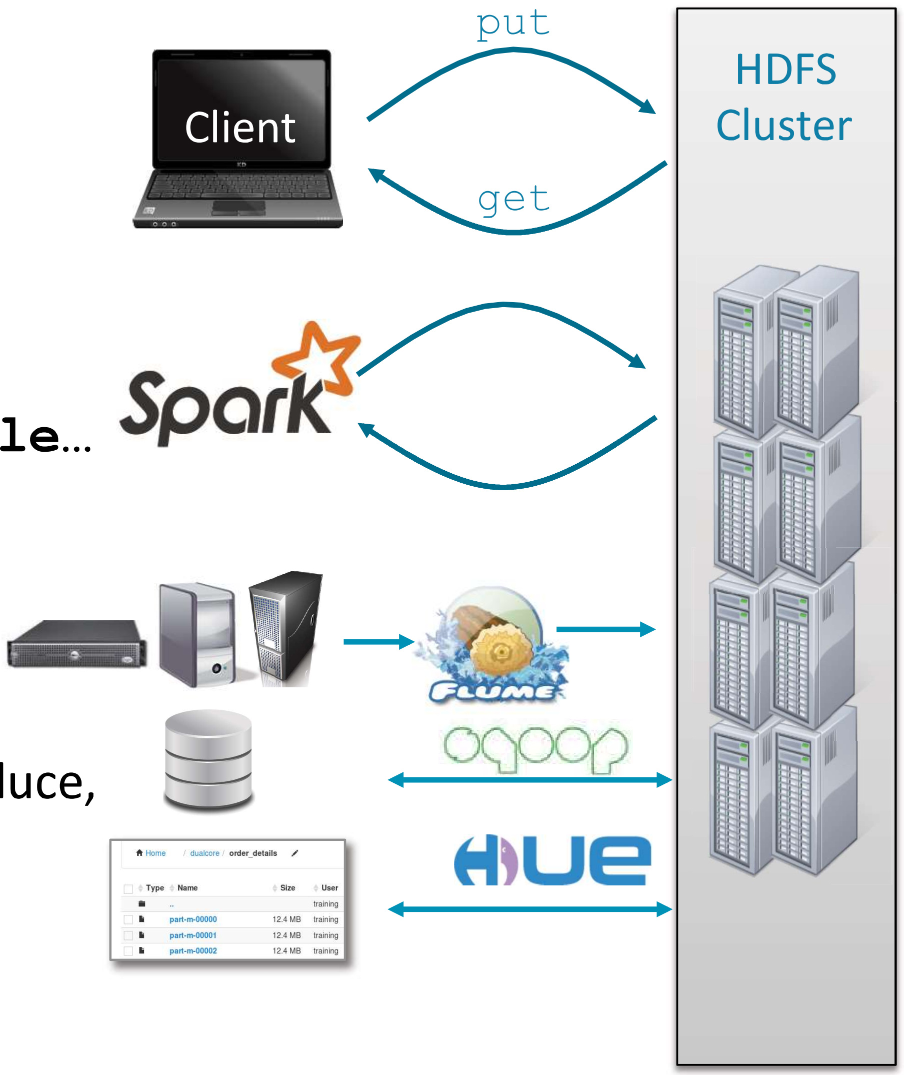
```
$ hdfs dfs
```

- **In Spark**
  - By URI, e.g.

```
hdfs://nnhost:port/file…
```

Hadoop API

- **Other programs**
  - Java API
    - Used by Hadoop MapReduce, Impala, Hue, Sqoop, Flume, etc.
  - RESTful interface



Home / dualcore / order_details

	Type	Name	Size	User
		..		training
		part-m-00000	12.4 MB	training
		part-m-00001	12.4 MB	training
		part-m-00002	12.4 MB	training

HDFS Command Line Examples (1)

- **Copy file `foo.txt` from local disk to the user's directory in HDFS**

```
$ hdfs dfs -put foo.txt foo.txt
```

  - This will copy the file to `/user/`*`username`*`/foo.txt`

- **Get a directory listing of the user's home directory in HDFS**

```
$ hdfs dfs -ls
```

- **Get a directory listing of the HDFS root directory**

```
$ hdfs dfs -ls /
```

HDFS Command Line Examples (2)

- **Display the contents of the HDFS file `/user/fred/bar.txt`**

```
$ hdfs dfs -cat /user/fred/bar.txt
```

- **Copy that file to the local disk, named as `baz.txt`**

```
$ hdfs dfs -get /user/fred/bar.txt baz.txt
```

- **Create a directory called `input` under the user's home directory**

```
$ hdfs dfs -mkdir input
```

Note: `copyFromLocal` is a synonym for `put`; `copyToLocal` is a synonym for `get`

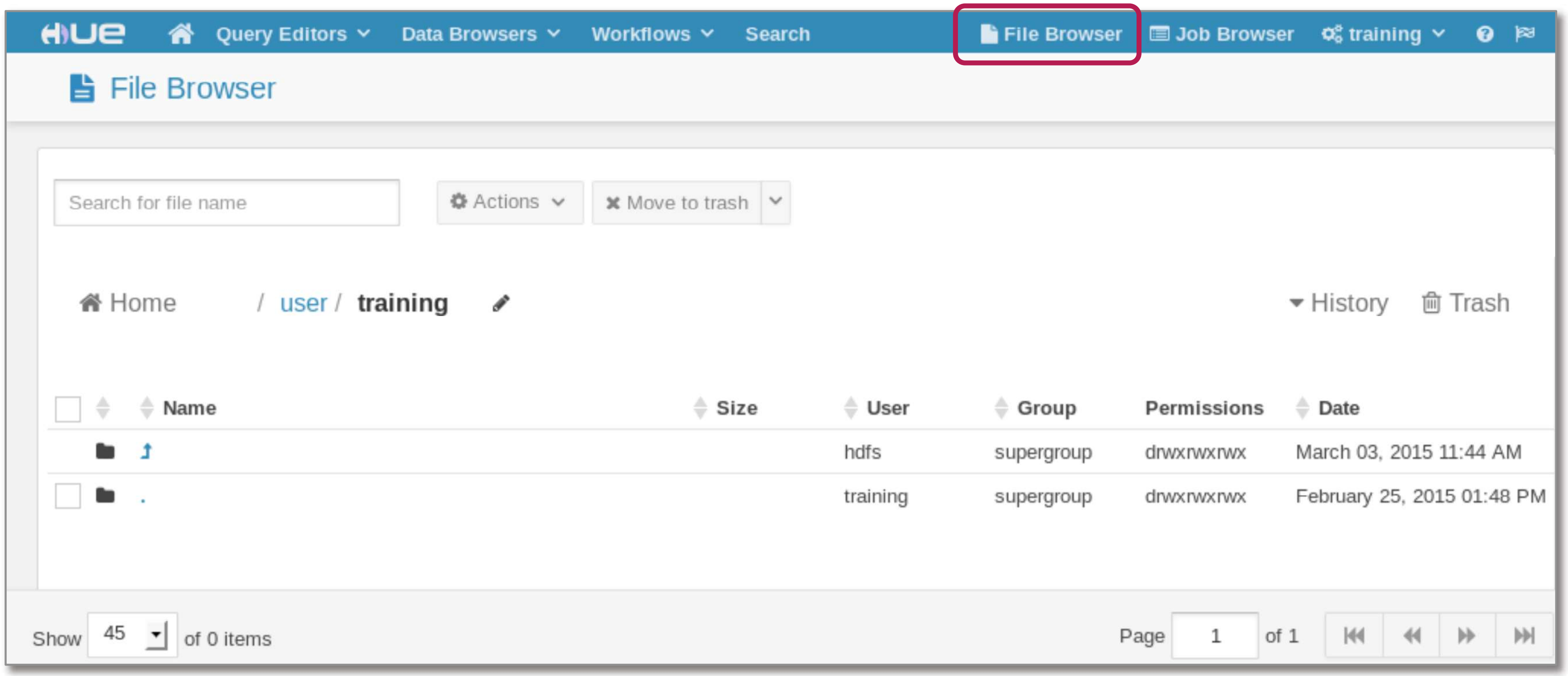
HDFS Command Line Examples (3)

- **Delete the directory `input_old` and all its contents**

```
$ hdfs dfs -rm -r input_old
```

The Hue HDFS File Browser

- **The File Browser in Hue lets you view and manage your HDFS directories and files**
  - Create, move, rename, modify, upload, download and delete directories and files
  - View file contents



	Name	Size	User	Group	Permissions	Date
	↥		hdfs	supergroup	drwxrwxrwx	March 03, 2015 11:44 AM
	.		training	supergroup	drwxrwxrwx	February 25, 2015 01:48 PM

HDFS Recommendations

- **HDFS is a repository for all your data**
  - Structure and organize carefully!
- **Best practices include**
  - Define a standard directory structure
  - Include separate locations for staging data
- **Example organization**
  - **`/user/`**… – data and configuration belonging only to a single user
  - **`/etl`** – Work in progress in Extract/Transform/Load stage
  - **`/tmp`** – Temporary generated data shared between users
  - **`/data`** – Data sets that are processed and available across the organization for analysis
  - **`/app`** – Non-data files such as configuration, JAR files, SQL files, etc.