

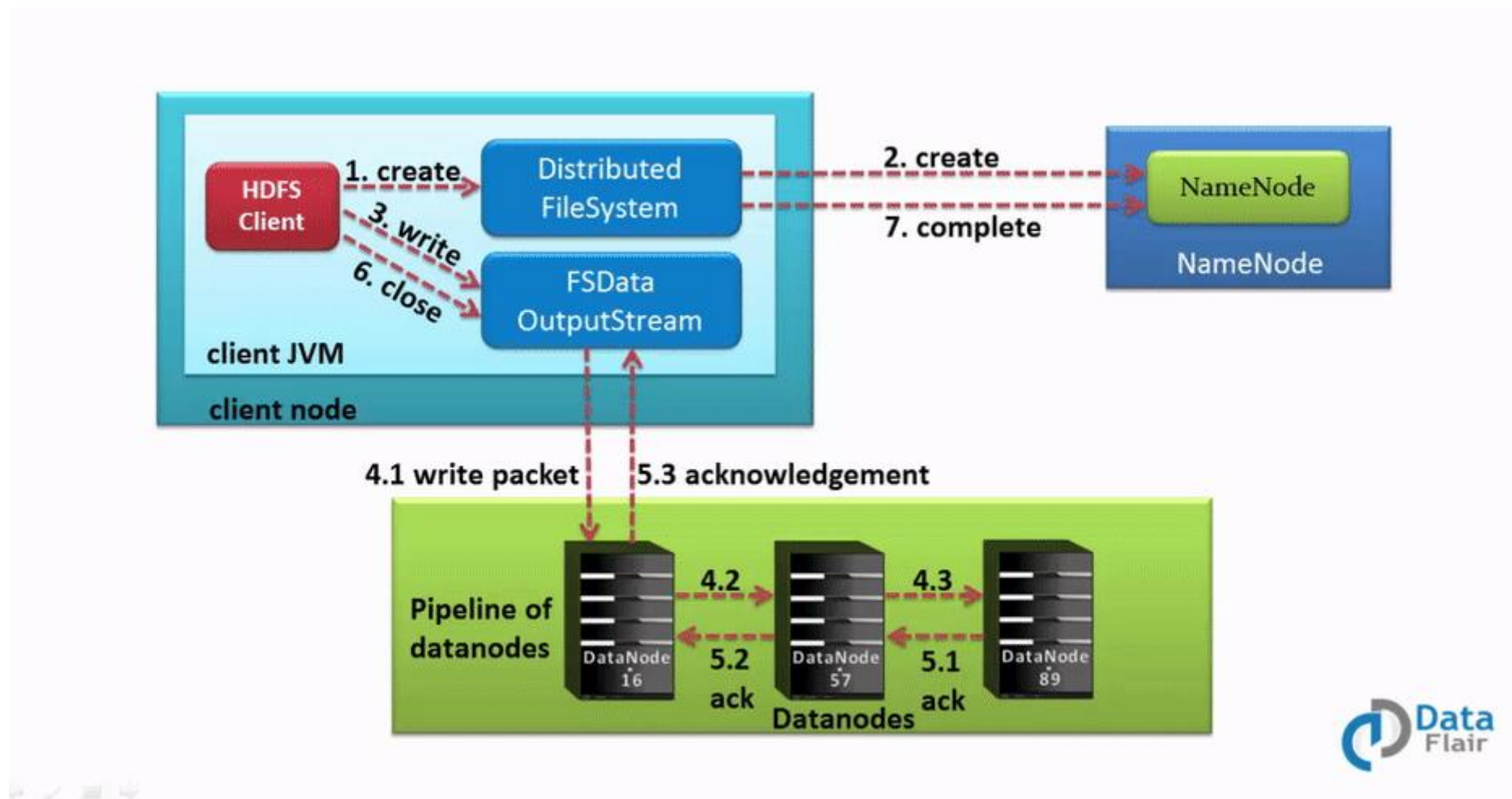
Odissee  
DE CO-HOGESCHOOL

# Big Data – HDFS operations



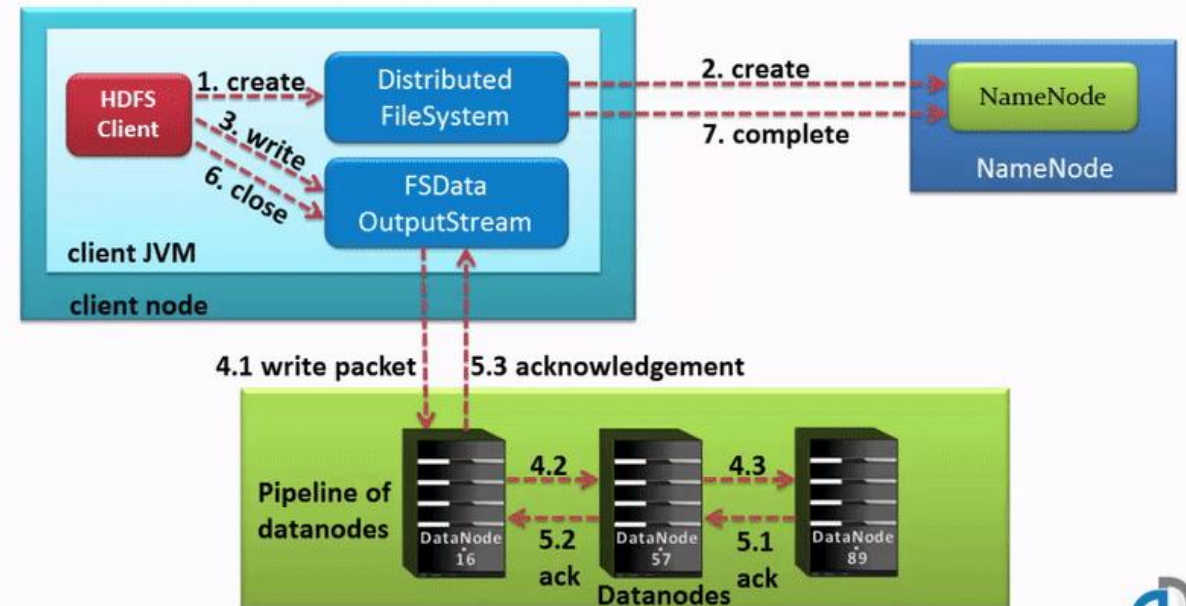
Jens Baetens

## Schrijven van data



## Wat met data consistentie?

- Data wordt achter elkaar geschreven van de ene datanode naar de andere
  - ▬ Lezen van data kan een willekeurige datanode kiezen waar de node op staat
  - ▬ Kan je dan nog de oude data lezen?
    - Ja
- Eventually consistent
  - ▬ Veel voorkomend concept
  - ▬ Ooit komt het in orde



## Hoe bewerken van een hdfs?

- ▣ Ofwel via commandline, communicatie met hdfs geconfigureerd in xml-files

- ▬ <https://www.informit.com/articles/article.aspx?p=2755708&seqNum=4>
- ▬ `hadoop/etc/hadoop/core-site.xml`
- ▬ `hadoop/etc/Hadoop/hdfs-site.xml`

- ▣ Maak gebruik van libraries om met een HDFS te communiceren

- |             |                      |
|-------------|----------------------|
| ▬ Pydoop    | Geavanceerdere taken |
| ▬ Snakebite | Eenvoudige taken     |
| ▬ Hdfs      | Eenvoudige taken     |
| ▬ Pyspark   | Data gebruiken       |

```
from snakebite.client import Client
```

```
client = Client('localhost', 9000)  
for x in client.ls(['/']):  
    print x
```

## HdfsCLI – python bindings

- ▣ API: <https://hdfscli.readthedocs.io/en/latest/>
- ▣ Bevat functies voor
  - te lezen
  - te schrijven
  - te verkennen
  - te manipuleren

## Aanmaken van bestanden en folders

### CLI

- ▣ Exists: hdfs dfs -test -d hdfs\_path
- ▣ Mkdir: hdfs dfs -mkdir -p /bigdata/HDFS
- ▣ hadoop fs -put /path/in/linux /hdfs/path *local cluster*
- ▣ hadoop fs -get /hdfs/path /path/in/linux

### Python

- ▣ status = client.status(path, strict=False)
- ▣ client.makedirs(path)
- ▣ client.upload(hdfs\_path, local\_path)
- ▣ client.download(hdfs\_path, local\_path)

## Bekijken van het filesystem

- ▣ `hdfs dfs -ls` command

- ▣ `hdfs dfs -usage`

- ▣ `hdfs dfs -cat <hdfs_filename> | head -n 5`

- ▣ `client.list(path, status=True)`

- ▣ `client.status(path, strict=False)`

```
with client.read(path) as reader:
    for i, line in enumerate(reader):
        if i >= num_lines:
            break
        print(line.decode('utf-8').strip())
```



## Aanpassen van het filesystem

- ▣ `hadoop fs -mv oldname newname`
- ▣ `hdfs dfs -rm path` (-r voor folders)
- ▣ `hdfs dfs -setrep -w 3 /tmp/logs/file.txt`
- ▣ `client.rename(old_path, new_path)`
- ▣ `client.delete(path, recursive=recursive)`
- ▣ `client.set_replication(path, replication)`

# Welke processen zijn er allemaal nodig voor een HDFS?

## ▣ Namenode

- ▬ Beheren van de file system namespace and reguleren van toegang tot bestanden
- ▬ `http://localhost:9870`

## ▣ SecondaryNamenode

- ▬ Backup van de primaire namenode
- ▬ `http://localhost:9868`

# Welke processen zijn er allemaal nodig voor een HDFS?

## ▣ Datanode

- ▬ normaal 1 per node in de cluster
- ▬ manage storage attached to the nodes that they run on
- ▬ <http://localhost:9864>

# Welke processen zijn er allemaal nodig voor een HDFS?

## ▣ Nodemanager

- ▬ Verantwoordelijk om containers te starten en te beheren op een node. Containers voeren taken uit gegeven door de AppMaster.
- ▬ <http://localhost:8042>

## ▣ Resourcemanager

- ▬ Verantwoordelijk voor het beheren van de beschikbare resources en plannen van applicaties
- ▬ <http://localhost:8088>

- ▣ Lijst van alle gebruikte poorten: <https://kontext.tech/article/265/default-ports-used-by-hadoop-services-hdfs-mapreduce-yarn>



<https://youtu.be/4GfIOWuONMY>