Introduction to HDFS

**Prasanth Kothuri**, CERN

What's HDFS

- HDFS is a distributed file system that is fault tolerant, scalable and extremely easy to expand.

- HDFS is the primary distributed storage for Hadoop applications.

- HDFS provides interfaces for applications to move themselves closer to data.

- HDFS is designed to 'just work', however a working knowledge helps in diagnostics and improvements.

Components of HDFS

There are two (*and a half*) types of machines in a HDFS cluster

- NameNode :– is the heart of an HDFS filesystem, it maintains and manages the file system metadata. E.g; what blocks make up a file, and on which datanodes those blocks are stored.

- DataNode :- where HDFS stores the actual data, there are usually quite a few of these.

HDFS Architecture
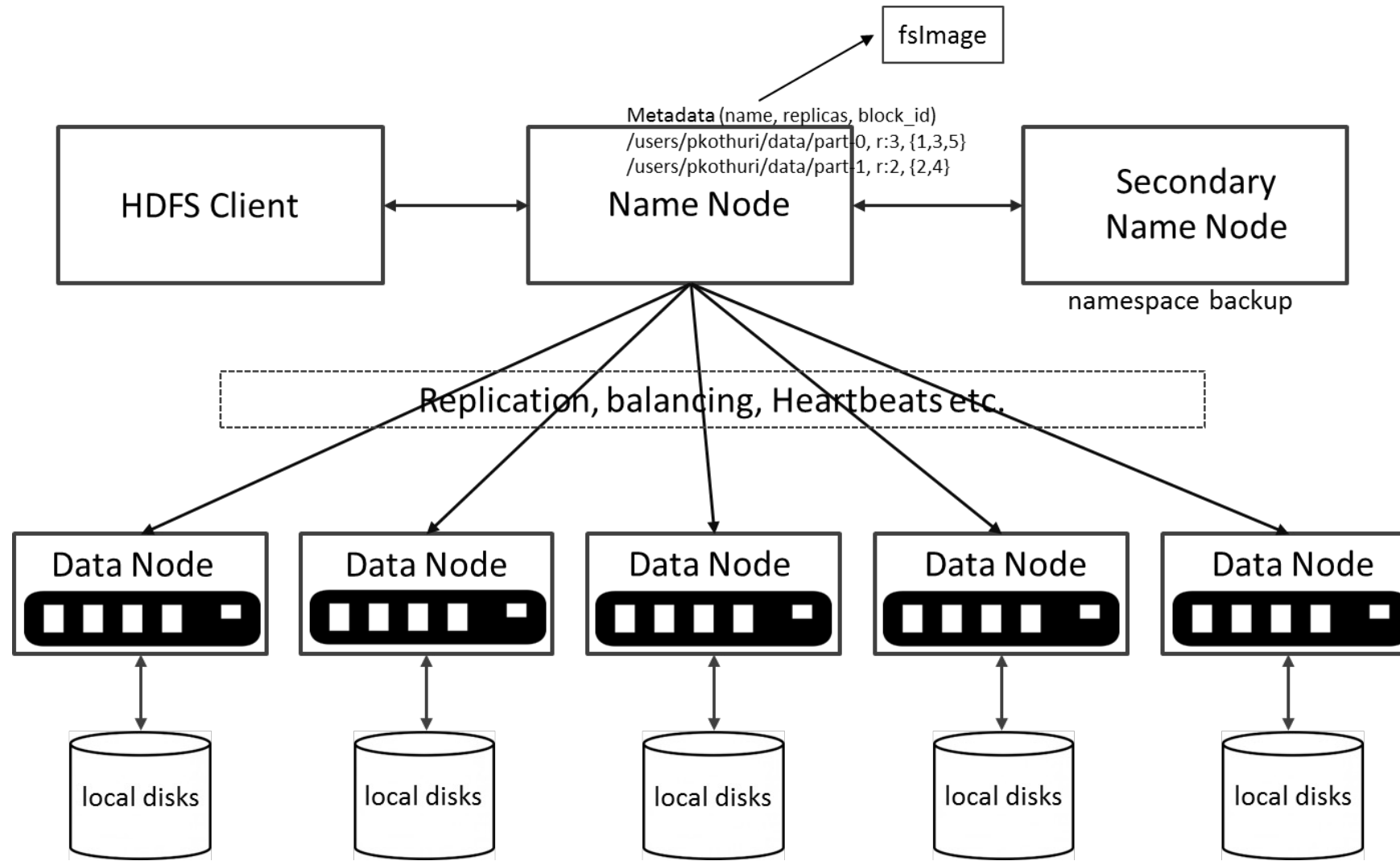
Unique features of HDFS

HDFS also has a bunch of unique features that make it ideal for distributed systems:

- <u>Failure tolerant</u> - data is duplicated across multiple DataNodes to protect against machine failures. The default is a replication factor of 3 (every block is stored on three machines).
- <u>Scalability</u> - data transfers happen directly with the DataNodes so your read/write capacity scales fairly well with the number of DataNodes
- <u>Space</u> - need more disk space? Just add more DataNodes and re-balance
- <u>Industry standard</u> - Other distributed applications are built on top of HDFS (HBase, Map-Reduce)

HDFS is designed to process large data sets with write-once-read-many semantics, it is not for low latency access

HDFS – Data Organization

- Each file written into HDFS is split into data blocks

- Each block is stored on one or more nodes

- Each copy of the block is called replica

- Block placement policy

  - First replica is placed on the local node

  - Second replica is placed in a different rack

  - Third replica is placed in the same rack as the second replica

Read Operation in HDFS
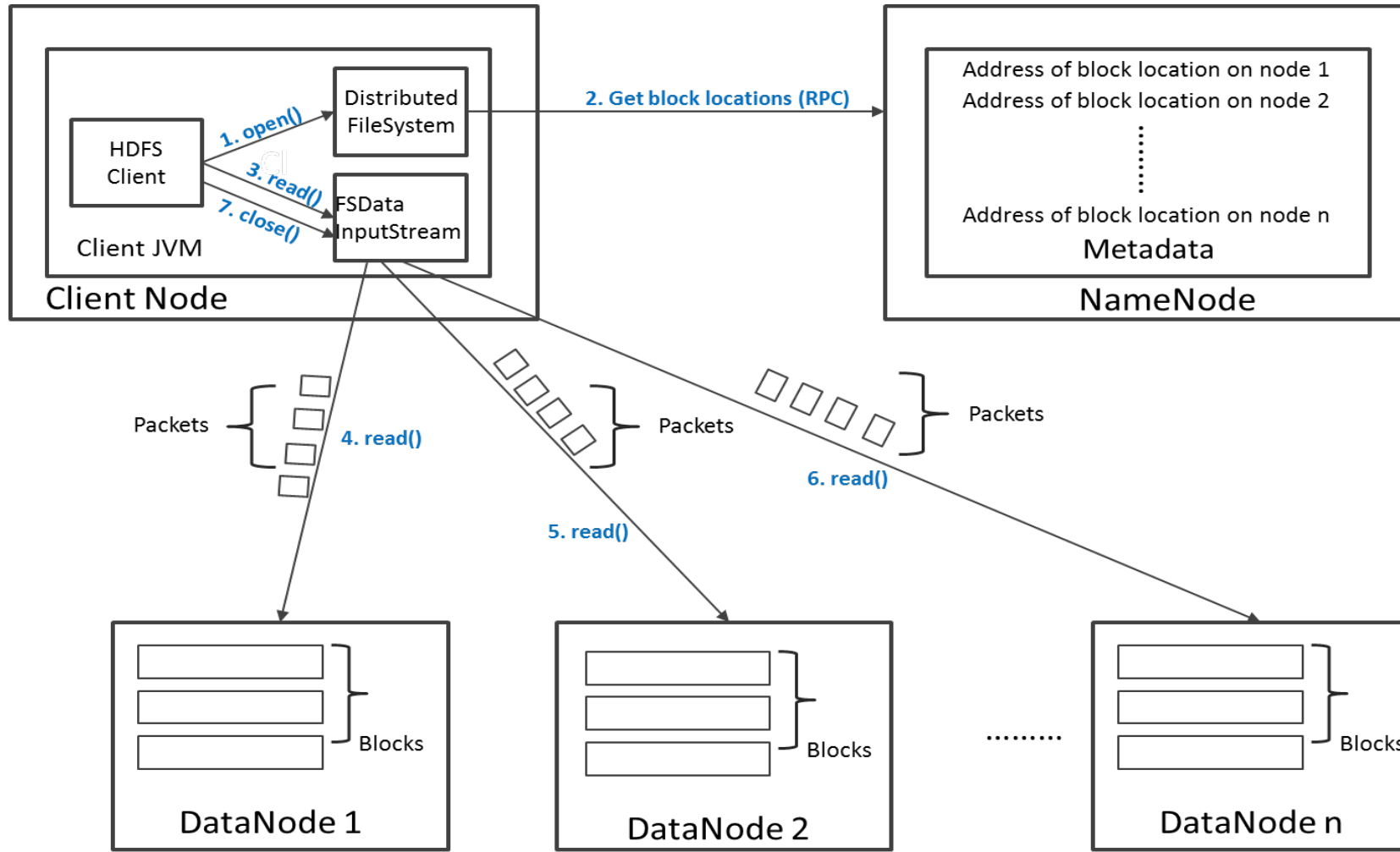
Write Operation in HDFS
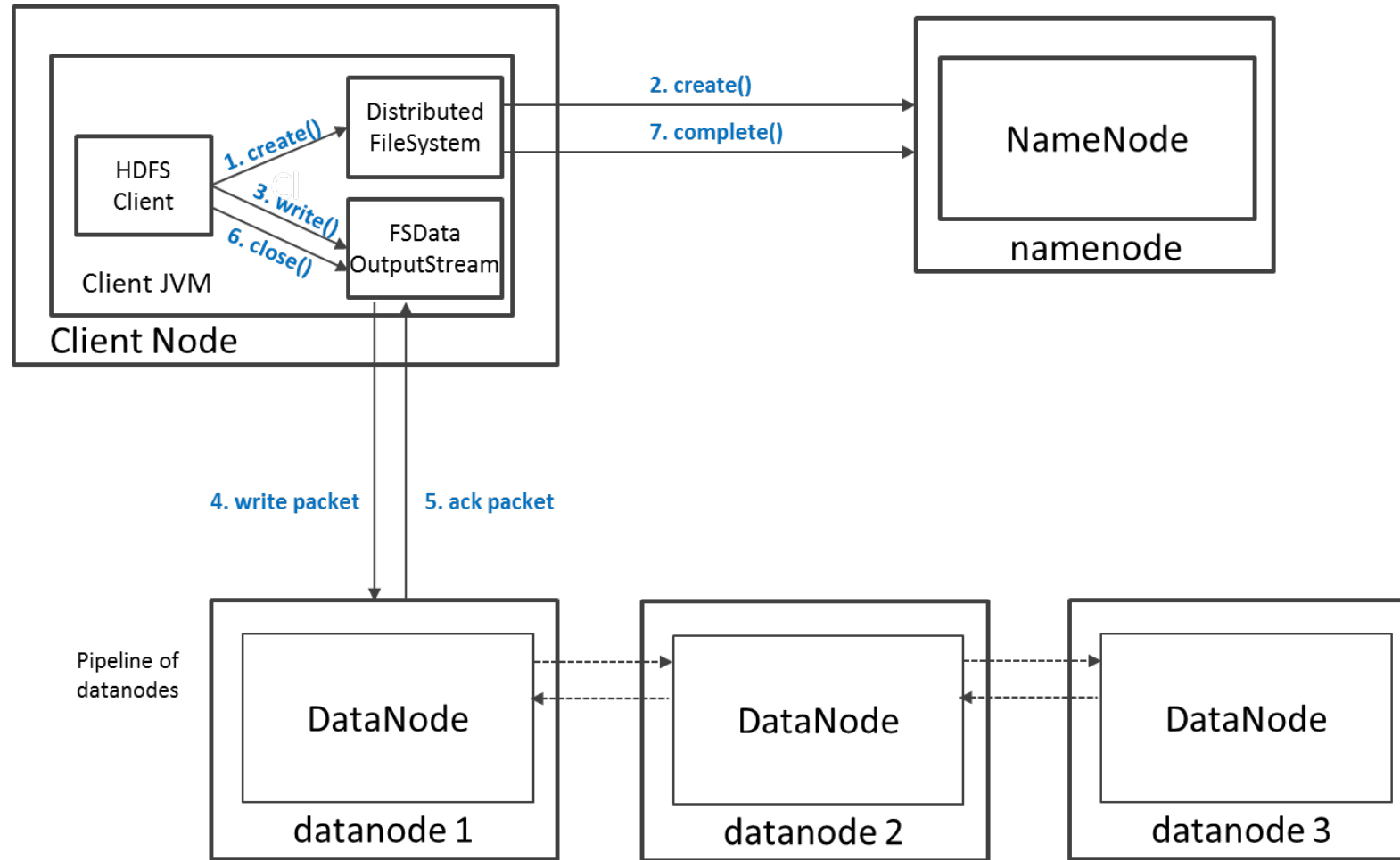
HDFS Security

- Authentication to Hadoop
  - Simple – insecure way of using OS username to determine hadoop identity
  - Kerberos – authentication using kerberos ticket
  - Set by `hadoop.security.authentication=simple|kerberos`

- File and Directory permissions are same like in POSIX
  - read (r), write (w), and execute (x) permissions
  - also has an owner, group and mode
  - enabled by default `(dfs.permissions.enabled=true)`

- ACLs are used for implemention permissions that differ from natural hierarchy of users and groups
  - enabled by `dfs.namenode.acls.enabled=true`

HDFS Configuration

HDFS Defaults

- Block Size – 64 MB
- Replication Factor – 3
- Web UI Port – 50070

HDFS conf file - `/etc/hadoop/conf/hdfs-site.xml`

```xml
<property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///data1/cloudera/dfs/nn,file:///data2/cloudera/dfs/nn</value>
</property>

<property>
    <name>dfs.blocksize</name>
    <value>268435456</value>
</property>

<property>
    <name>dfs.replication</name>
    <value>3</value>
</property>

<property>
    <name>dfs.namenode.http-address</name>
    <value>itracXXX.cern.ch:50070</value>
</property>
```

Interfaces to HDFS

- Java API (`DistributedFileSystem`)
- C wrapper (`libhdfs`)
- HTTP protocol
- WebDAV protocol
- Shell Commands

However the command line is one of the simplest and most familiar

HDFS – Shell Commands

There are two types of shell commands

User Commands

`hdfs dfs` – runs filesystem commands on the HDFS

`hdfs fsck` – runs a HDFS filesystem checking command

Administration Commands

`hdfs dfsadmin` – runs HDFS administration commands

HDFS – User Commands (dfs)

List directory contents

```
hdfs dfs –ls
hdfs dfs -ls /
hdfs dfs -ls -R /var
```

Display the disk space used by files

```
hdfs dfs -du -h /
hdfs dfs -du /hbase/data/hbase/namespace/
hdfs dfs -du -h /hbase/data/hbase/namespace/
hdfs dfs -du -s /hbase/data/hbase/namespace/
```

HDFS – User Commands (dfs)

Copy data to HDFS

```
hdfs dfs -mkdir tdata
hdfs dfs -ls
hdfs dfs -copyFromLocal tutorials/data/geneva.csv tdata
hdfs dfs -ls –R
```

Copy the file back to local filesystem

```
cd tutorials/data/
hdfs dfs –copyToLocal tdata/geneva.csv geneva.csv.hdfs
md5sum geneva.csv geneva.csv.hdfs
```

HDFS – User Commands (acls)

List acl for a file

```
hdfs dfs -getfacl tdata/geneva.csv
```

List the file statistics – (%r – replication factor)

```
hdfs dfs -stat "%r" tdata/geneva.csv
```

Write to hdfs reading from stdin

```
echo "blah blah blah" | hdfs dfs -put - tdataset/tfile.txt
hdfs dfs -ls -R
hdfs dfs -cat tdataset/tfile.txt
```

HDFS – User Commands (fsck)

Removing a file

```
hdfs dfs -rm tdataset/tfile.txt
hdfs dfs -ls -R
```

List the blocks of a file and their locations

```
hdfs fsck /user/cloudera/tdata/geneva.csv -
files -blocks -locations
```

Print missing blocks and the files they belong to

```
hdfs fsck / -list-corruptfileblocks
```

HDFS – Adminstration Commands

Comprehensive status report of HDFS cluster

```
hdfs dfsadmin -report
```

Prints a tree of racks and their nodes

```
hdfs dfsadmin -printTopology
```

Get the information for a given datanode (like ping)

```
hdfs dfsadmin -getDatanodeInfo
localhost:50020
```

HDFS – Advanced Commands

Get a list of namenodes in the Hadoop cluster

```
hdfs getconf –namenodes
```

Dump the NameNode fsimage to XML file

```
cd /var/lib/hadoop-hdfs/cache/hdfs/dfs/name/current
hdfs oiv -i fsimage_0000000000000003388 -o
/tmp/fsimage.xml -p XML
```

The general command line syntax is

hdfs command [genericOptions] [commandOptions]

Other Interfaces to HDFS

HTTP Interface

```
http://quickstart.cloudera:50070
```

MountableHDFS – FUSE

```
mkdir /home/cloudera/hdfs
sudo hadoop-fuse-dfs dfs://quickstart.cloudera:8020
/home/cloudera/hdfs
```

Once mounted all operations on HDFS can be performed using standard Unix utilities such as 'ls', 'cd', 'cp', 'mkdir', 'find', 'grep',

Q & A

E-mail: Prasanth.Kothuri@cern.ch

Blog: http://prasanthkothuri.wordpress.com