DSC650: Data Technology and Future Emergence Lecture 7 : Big Data Technologies

Lecturer: Dr Khairul Anwar Sedek

Lecture 7: **Big Data Technologies**

Outlines:

- Introduction to Apache Hadoop
- Hadoop Ecosystem
- Query Language for Hadoop
- Hadoop and Amazon Cloud
- Migration to Other Big Data Platform

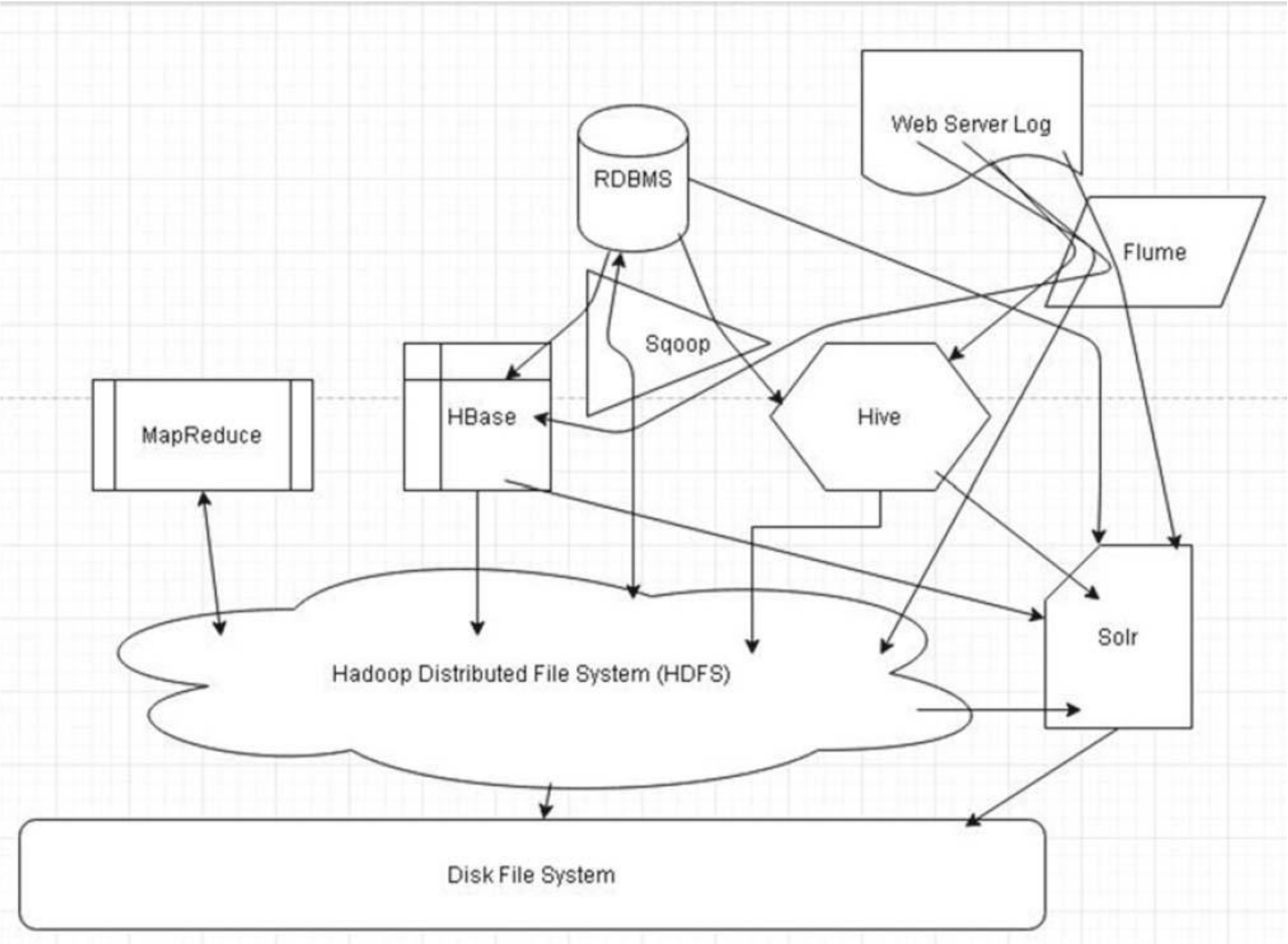
Learning Outcomes

- CLO1: Demonstrate an understanding on the basic **concepts and practices** of big data technology

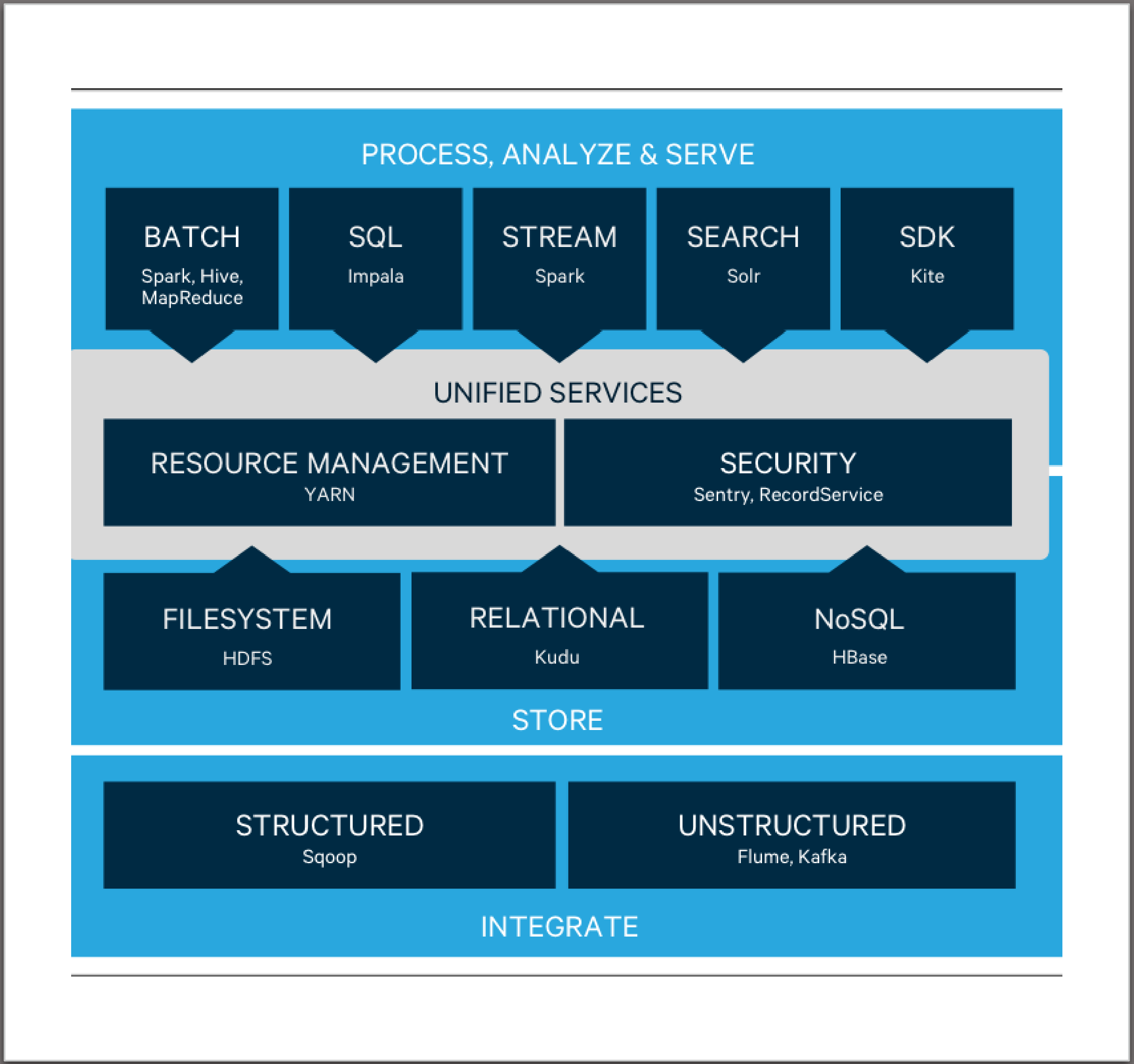
Introduction to Apache Hadoop

- (Apache) Hadoop is the **de facto framework** for **processing and storing large quantities of data**, what is often referred to as "big data“
- Hadoop is a **large-scale distributed processing system** designed for a cluster consisting of hundreds or thousands of nodes, each with multi-processor CPU cores.
- Hadoop is designed to distribute and process large quantities of data **across the nodes in the cluster**.
- Hadoop **does not require any special hardware** and is designed for commodity hardware.
- Hadoop may be used with **any type of data, structured or unstructured**.
- Hadoop is **not a database**, nor does Hadoop replace traditional database systems.
- The (Apache) Hadoop ecosystem consists of dozens of projects providing functionality ranging from storing, querying, indexing, transferring, streaming, and messaging.

Hadoop Ecosystem



Hadoop Ecosystem



PROCESS, ANALYZE & SERVE

BATCH
Spark, Hive, MapReduce

SQL
Impala

STREAM
Spark

SEARCH
Solr

SDK
Kite

UNIFIED SERVICES

RESOURCE MANAGEMENT
YARN

SECURITY
Sentry, RecordService

FILESYSTEM
HDFS

RELATIONAL
Kudu

NoSQL
HBase

STORE

STRUCTURED
Sqoop

UNSTRUCTURED
Flume, Kafka

INTEGRATE

Two main components of Hadoop:

The Hadoop Distributed Filesystem (HDFS)

- The HDFS is a **data storage and data processing file system**.

MapReduce

- MapReduce is a **distributed data processing framework** for processing large quantities of data, distributed across a cluster of nodes, in parallel.

HDFS

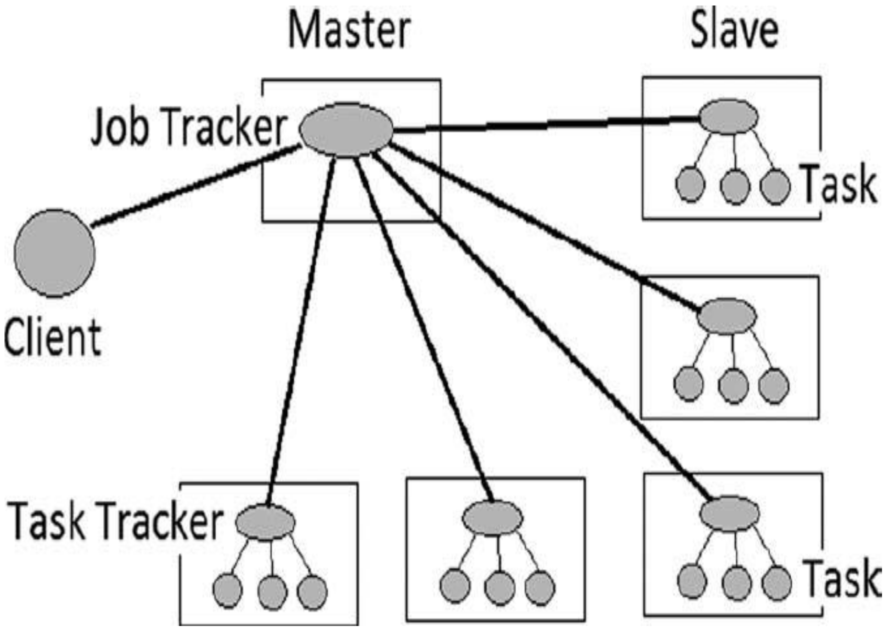
HDFS is designed to store and provide **parallel, streaming access to large quantities of data**

HDFS storage is **spread across a cluster of nodes**; a single large file could be stored across multiple nodes in the cluster.

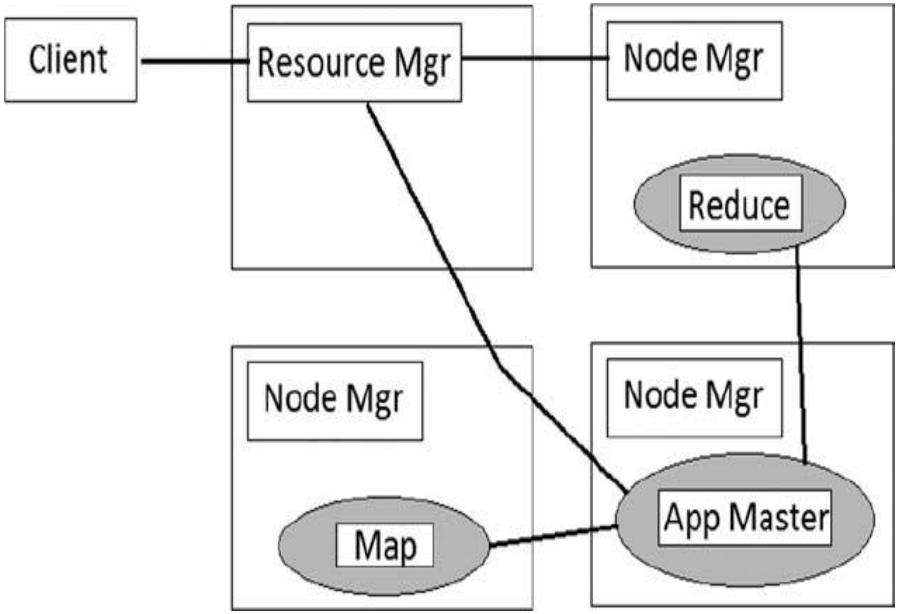
A **file is broken into blocks**, which is an abstraction over the underlying filesystem, with default size of 64MB.

HDFS is designed to store **large files and lots of them**.

Hadoop V1 vs V2



In the V1 architecture, a master Job Tracker is used to **manage Task Trackers on slave nodes**. Hadoop's **data node and Task Trackers co-exist on the same slave nodes.**



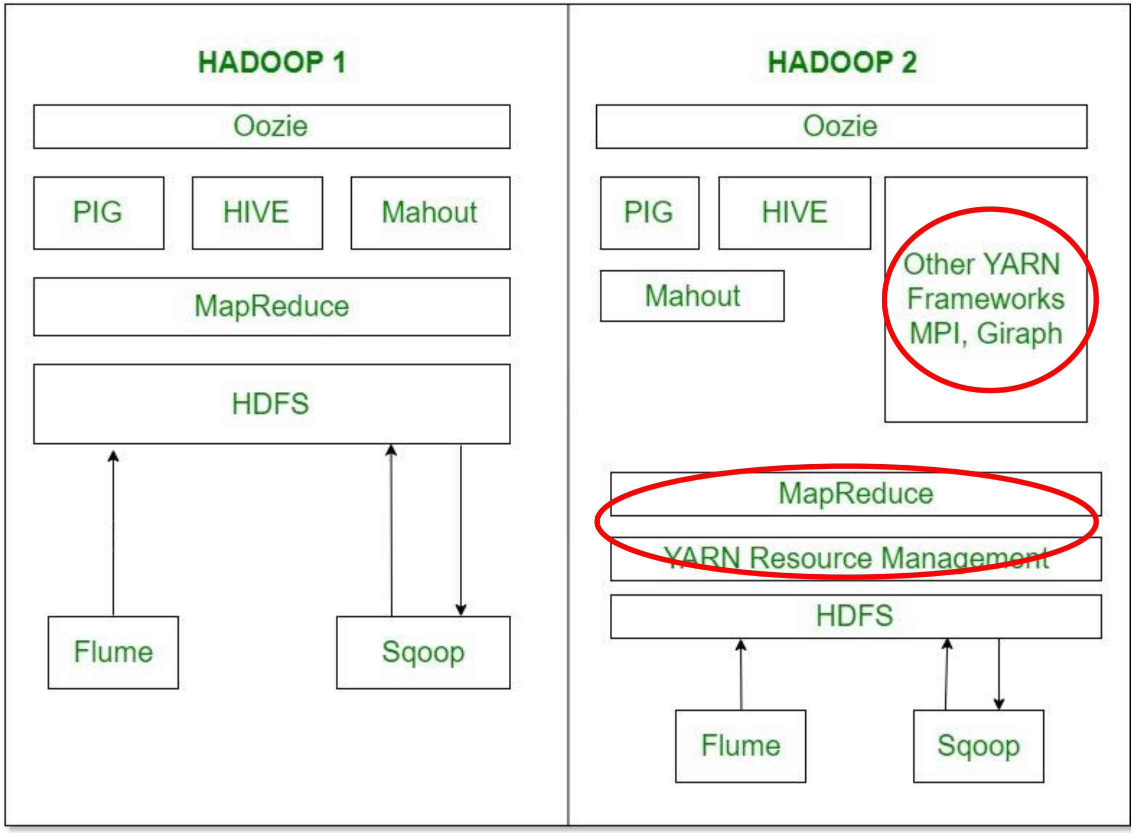
Hadoop V2 offers the **ability to use non-Map Reduce processing**, like Apache Graph for graph processing, or Impala for data query.
Client task requests being sent to the **global Resource Manager** and the **slave-based Node Managers** launching containers, which have the actual tasks.
The Application Master requests containers from the scheduler and receives status updates from the container-based Map Reduce tasks.

Hadoop V1 vs V2

Aspect of	V1	V2
Components	HDFS Map Reduce	HDFS YARN / MRv2
Daemons	Namenode Datanode Secondary Namenode Job Tracker Task Tracker	Namenode Datanode Secondary Namenode Resource Manager Node Manager
Working	In *Hadoop 1*, there is HDFS which is used for storage and top of it, **Map Reduce which works as Resource Managemen**t as well as Data Processing. Due to this workload on Map Reduce, it will affect the performance.	In *Hadoop 2, there* HDFS which is again used for storage and on the top of HDFS, there is **YARN which works as Resource Management**. It basically allocates the resources and keeps all the things going on.
Limitations	Consists of a single master and multiple slaves.	consists of multiple masters (i.e active namenodes and standby namenodes) and multiple slaves.

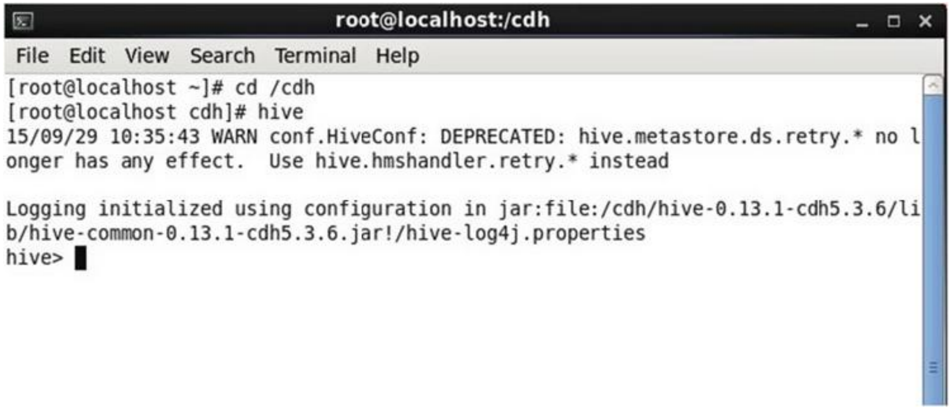
https://www.geeksforgeeks.org/difference-between-hadoop-1-and-hadoop-2/

Hadoop Ecosystem V1 vs V2



Hive

- (Apache) Hive is a **data warehouse framework** **for querying and managing large datasets** stored in Hadoop distributed filesystems (HDFS).
- Hive also provides a **SQL-like query** language called **HiveQL**. The HiveQL queries may be run in the Hive CLI shell.



```
[root@localhost ~]# cd /cdh
[root@localhost cdh]# hive
15/09/29 10:35:43 WARN conf.HiveConf: DEPRECATED: hive.metastore.ds.retry.* no l
onger has any effect.  Use hive.hmshandler.retry.* instead

Logging initialized using configuration in jar:file:/cdh/hive-0.13.1-cdh5.3.6/li
b/hive-common-0.13.1-cdh5.3.6.jar!/hive-log4j.properties
hive> 
```

Starting the Hive command line interface (CLI)

Type of Hive Tables

Managed/internal tables

managed by the Hive framework

When a Hive-managed table is deleted, the **table's data and the metadata are deleted**.

```
CREATE TABLE ratings_managed
( userid int,
movieid int ,
rating int ,
tstamp string
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

external tables.

Hive **manages only the metadata** and the **data is stored in some external location** not managed by Hive.

When a Hive external table is deleted, **only the metadata is deleted**; the data is retained.

That location is external to the Hive framework but not external to HDFS, as data for a Hive external table is **also stored in HDFS**.

```
CREATE EXTERNAL TABLE ratings (
userid int,
movieid int ,
rating int ,
tstamp string
)ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/cloudera/360dt/data/ratings' ;
```

Apache HBase

- Apache Hbase is a **distributed, scalable database** designed for Apache Hadoop.
- HBase is a flexible format NoSQL database.
- HBase has three main components: *HMaster, ZooKeeper*, and *RegionServers*.
  - The HMaster handles the DDL (create and delete) operations.
  - The ZooKeeper is a distributed coordination service for an HBase cluster.
  - RegionServers manage HBase table data and serve client requests.
- HMaster manages region assignment.
- Regions are stored in RegionServers, which serve PUT/GET requests from a client.
- Each RegionServer is collocated with a DataNode on HDFS.
- HBase table data is stored in the HDFS.
- The metadata for the Region ➤ RegionServer mapping is kept in a metatable, which is stored on the ZooKeeper.
- A client request is first sent to the ZooKeeper, which provides the RegionServer the locations for the requested data.
- Subsequently, the client GETs/PUTs data directly on a RegionServer.

HBase: ‘The Hadoop Database’

HBase is a column family-store database layered on top of HDFS

- Based on Google’s Big Table
- Provides interactive access to data

Can store massive amounts of data

- Multiple Terabytes, up to Petabytes of data

High Write Throughput

- Scales up to millions of writes per second

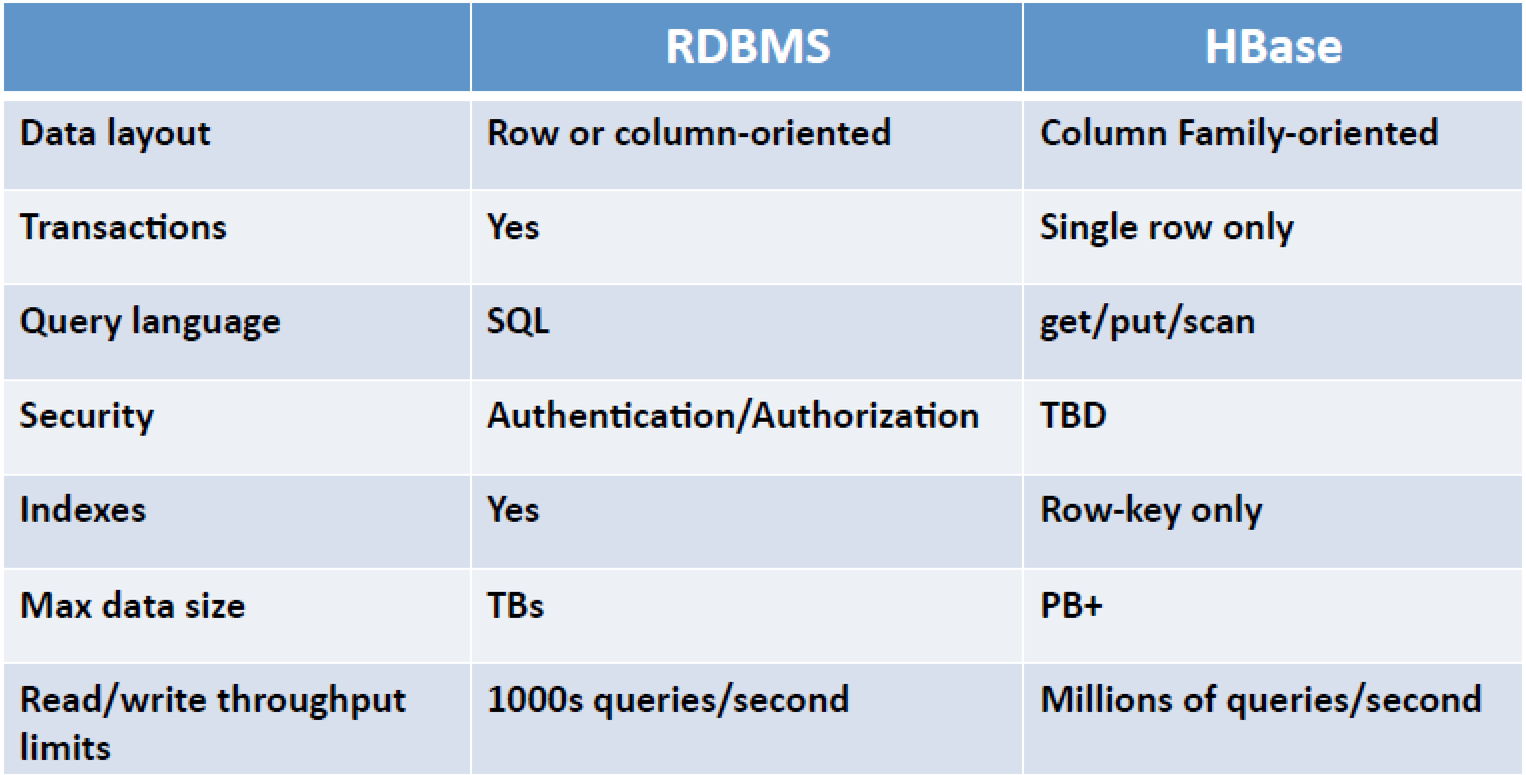
Copes well with sparse data

- Tables can have many thousands of columns
- Even if a given row only has data in a few of the columns

Has a constrained access model

- Limited to lookup of a row by a single key
- No transactions
- Single row operations only

HBase vs A Traditional RDBMS



	RDBMS	HBase
Data layout	Row or column-oriented	Column Family-oriented
Transactions	Yes	Single row only
Query language	SQL	get/put/scan
Security	Authentication/Authorization	TBD
Indexes	Yes	Row-key only
Max data size	TBs	PB+
Read/write throughput limits	1000s queries/second	Millions of queries/second

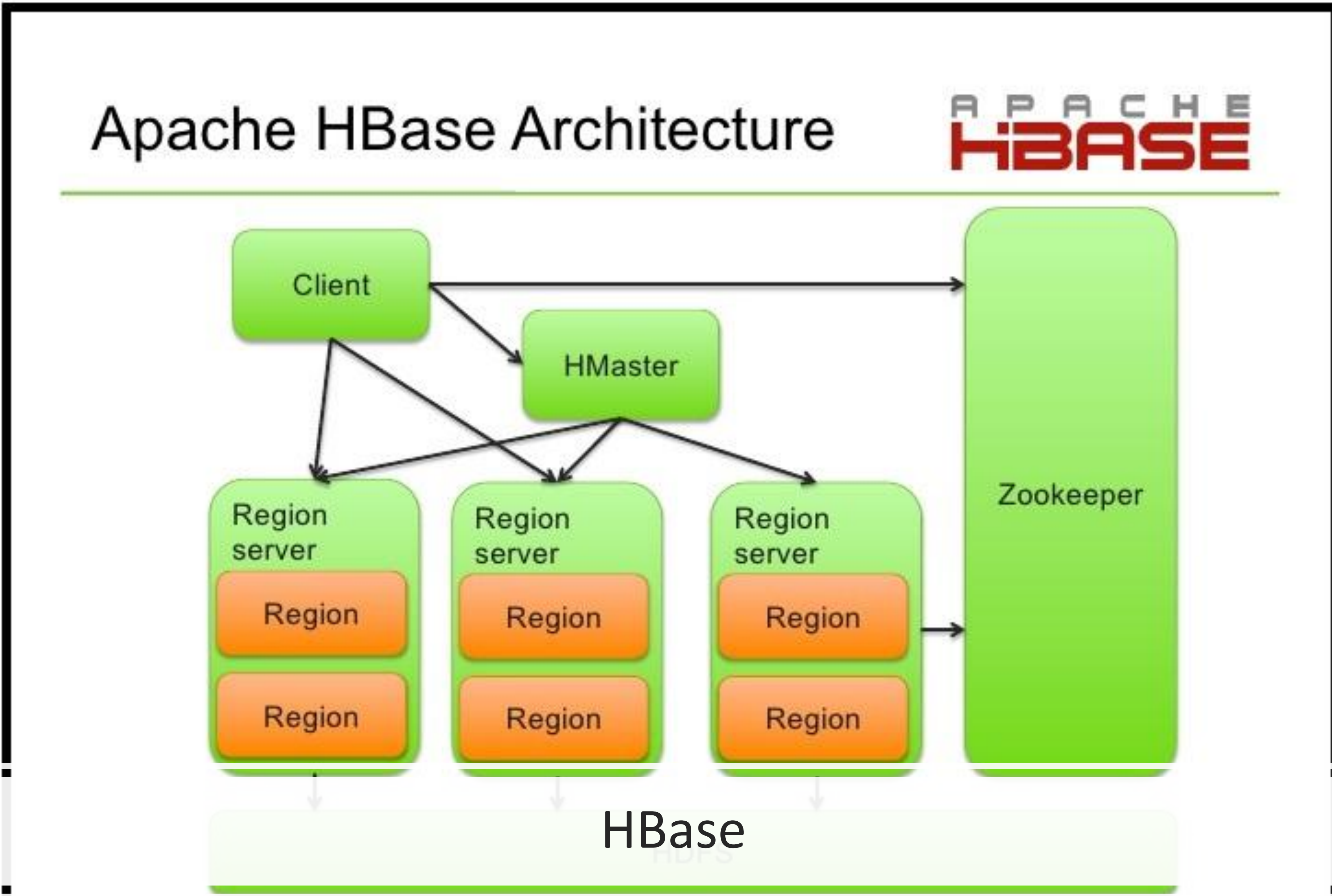
When To Use HBase

Use HBase if…

- You need to have **random, real-time read/write access** to Big Data.
- You need to do **many thousands of operations per second** on multiple TB of data
- Your **access patterns are well-known** and **simple**

Don’t use HBase if…

- You only append to your dataset, and tend to read the whole thing
- You primarily do ad-hoc analytics (ill-defined access patterns)
- Your data easily fits on one beefy node



Apache HBase Architecture

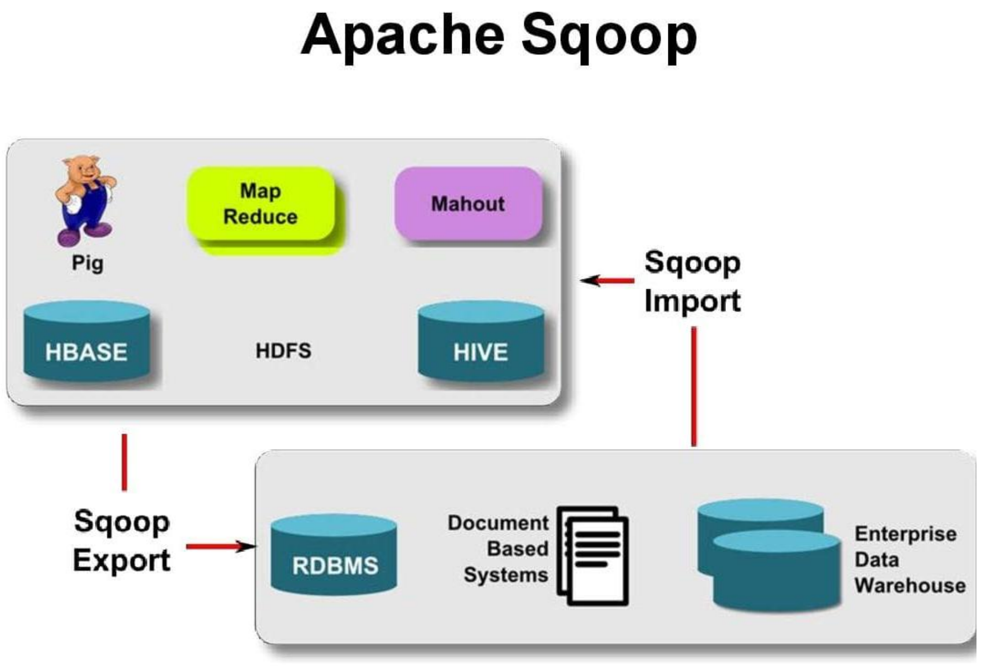
HBase

https://www.tutorialandexample.com/hbase-architecture/

Hbase Basic Commands

- General Commands
  - general commands are categorized into following commands: **Status, Version, Table_help (scan, drop, get, put, disable, etc.), and Whoami**
- Data Definition Commands
  - to create tables and table schemas with rows and column families
  - **Create Table**
    - **create <tablename>, <columnfamilyname>**
    - **create ‘student' ,’profile’,‘program’, ‘course’**
- Data Manipulation Commands
  - These commands will work on the table related to data manipulations such as putting data into a table, retrieving data from a table and deleting schema, etc.
  - Count, Put, Get, Delete, Delete all, Truncate, Scan
- Other HBase Shell Commands
  - **Cluster Replication Commands**
    - For adding and removing peers to cluster and to start and stop replication these commands are used in general.

Apache Sqoop

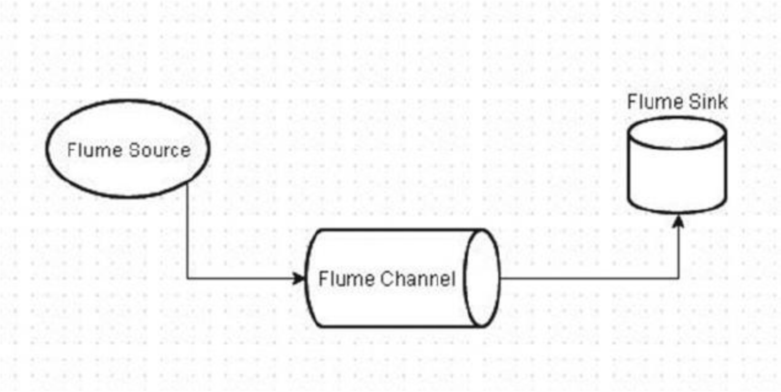


- Sqoop is a tool **for transferring large quantities of data between a relational database, such as MySQL and Oracle database, and the Hadoop ecosystem**, which includes the Hadoop Distributed File System (HDFS), Apache Hive, and Apache HBase.
- While Sqoop supports transfer <u>between a relational database and HDFS</u> **bi-directionally**,
- Sqoop only supports transfer <u>**from a relational database to Apache Hive**</u> and Apache HBase **uni-directionally.**

Flume

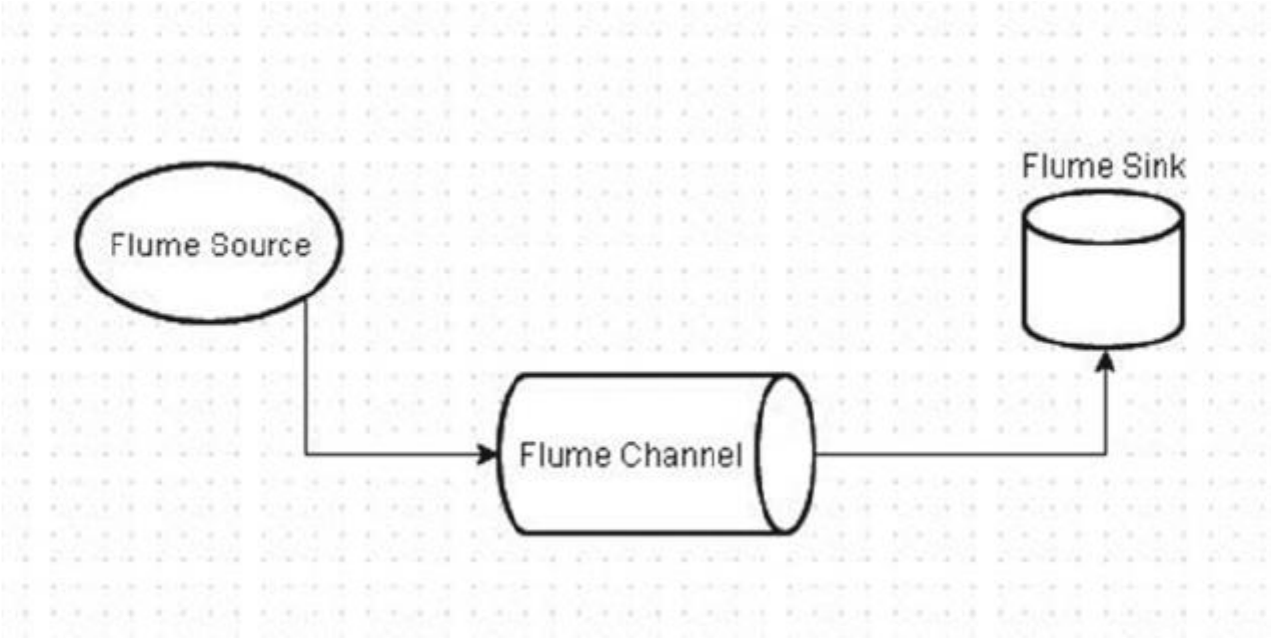
- (Apache) Flume is a **framework based on streaming data flows** for **collecting, aggregating, and transferring large quantities of data**.
- Flume is an efficient and reliable **distributed service**.
- A unit of data flow in Flume is called an ***event***.

Flume Main Components



- *Flume source*
  - A Flume source consumes events from an external source such as a log file or a web server.
- *Flume channel*
  - A Flume source stores the events it receives in a passive data store called a Flume channel
  - Examples of Flume channel types are a JDBC channel, a file channel, and a memory channel.
- *Flume sink*
  - The Flume sink component removes the events from the Flume channel and puts them in an external storage such as HDFS.
  - A Flume sink can also forward events to another Flume source to be processed by another Flume agent.

Flume



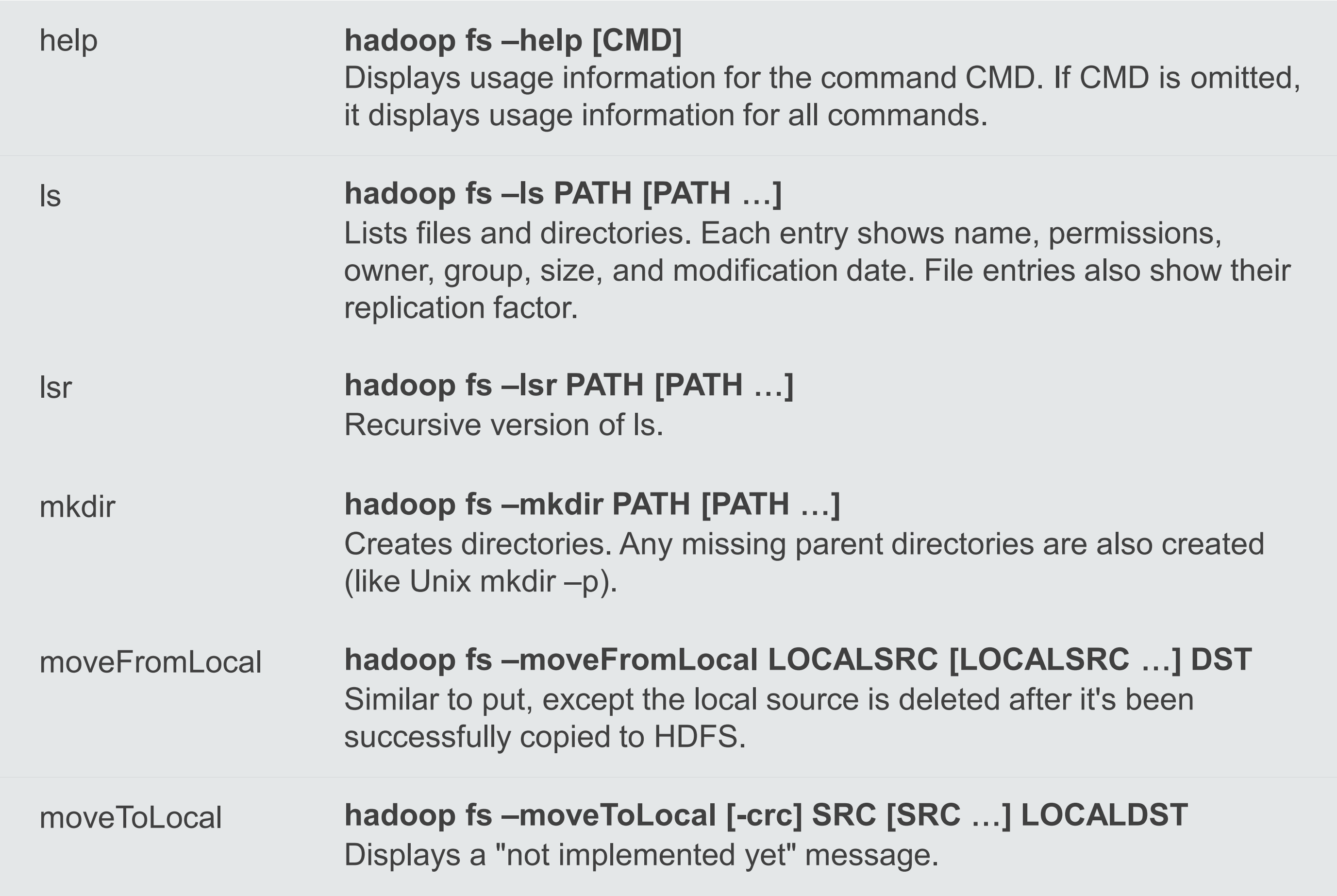
Different types of sources, channels, and sinks are supported by the Apache Flume framework.

Some of the supported source types are Avro Source, Thrift Source, Exec Source, and HTTP Source.

Some of the supported channel types are Memory Channel, JDBC Channel, and File Channel.

Some of the supported Sink types are HDFS Sink, MorphlineSolr Sink, and HBase Sink.

Hadoop Commands



help	**hadoop fs –help [CMD]** Displays usage information for the command CMD. If CMD is omitted, it displays usage information for all commands.
ls	**hadoop fs –ls PATH [PATH …]** Lists files and directories. Each entry shows name, permissions, owner, group, size, and modification date. File entries also show their replication factor.
lsr	**hadoop fs –lsr PATH [PATH …]** Recursive version of ls.
mkdir	**hadoop fs –mkdir PATH [PATH …]** Creates directories. Any missing parent directories are also created (like Unix mkdir –p).
moveFromLocal	**hadoop fs –moveFromLocal LOCALSRC [LOCALSRC …] DST** Similar to put, except the local source is deleted after it's been successfully copied to HDFS.
moveToLocal	**hadoop fs –moveToLocal [-crc] SRC [SRC …] LOCALDST** Displays a "not implemented yet" message.

Hadoop Commands

mv	**hadoop fs –mv SRC [SRC …] DST** Moves files from source(s) to destination. If multiple source files are specified, destination has to be a directory. Moving across filesystems is not permitted.
put	**hadoop fs –put LOCALSRC [LOCALSRC …] DST** Copies files or directories from local system to destination filesystem. If LOCALSRC is set to -, input is set to stdin and DST must be a file.
rm	**hadoop fs –rm PATH [PATH …]** Deletes files and empty directories.
copyFromLocal	**hadoop fs –copyFromLocal LOCALSRC [LOCALSRC …] DST** Identical to put (copy files from the local file system).
copyToLocal	**hadoop fs –copyToLocal [-ignorecrc] [-crc] SRC [SRC…] LOCALDST** Identical to get (copy files to the local file system).

Hadoop & Amazon Cloud

Amazon – pioneers in the big data cloud services space, Amazon offers three relevant services, all of which fall under the Amazon Web Services (http://aws.amazon.com/) umbrella.

Amazon Elastic Compute Cloud (EC2)

- The Amazon Elastic Compute Cloud (http://aws.amazon.com/ec2/) is web-based service that provides resizable compute capacity in the cloud.
- Amazon offers preconfigured Windows and Linux templates that include a variety of preinstalled application systems.
- Instances from small or micro to large or high memory/processor capacity can be created.
- You can easily resize your instances as your computing needs change.

Amazon Elastic MapReduce (EMR)

- Built on top of the EC2 service, the Elastic MapReduce service (http://aws.amazon.com/elasticmapreduce/) offers businesses a resizable, hosted Hadoop platform for processing large quantities of data.
- The EMR service supports a number of processing jobs, including MapReduce, MapReduce Streaming (via Java, Ruby, Perl, Python, PHP, R, and C++), Hive, and Pig.

Amazon Simple Storage Service (S3)

- The Simple Storage Service (S3) service (http://aws.amazon.com/s3/) provides inexpensive, highly scalable, reliable, secure, and fast data storage in the cloud.
- Data can easily be backed up or even globally distributed and is accessed and managed via a REST API.

Hadoop & Microsoft Azure

Microsoft

- Microsoft (although arriving later) has quickly and dynamically expanded their footprint through innovative and proven solutions in their Windows Azure (http://www.windowsazure.com/) offering.
- Within the context of big data, the services of interest include HDInsight and Azure Blob Storage.

HDInsight

- HDInsight (http://www.windowsazure.com/en-us/services/hdinsight/) is Microsoft's 100% Apache Hadoop implementation available as a cloud service.
- You can quickly and easily provision an elastic Hadoop cluster ranging from 4 to 32 nodes that allows for seamless scaling.

Azure (ASV) Blob Storage

- Like Amazon's S3 service, the Microsoft Azure (ASV) Blob Storage (http://www.windowsazure.com/en-us/services/data-management/) offers a high-performance, reliable, scalable, and secure cloud storage solution that supports backup and global distribution via a content delivery network.
- In addition to traditional blob storage, the ASV service also supports table-based and queue storage.
- Together, both of these service providers offer a first-class storage service and robust, stable processing platform capable of hosting your big data solution.