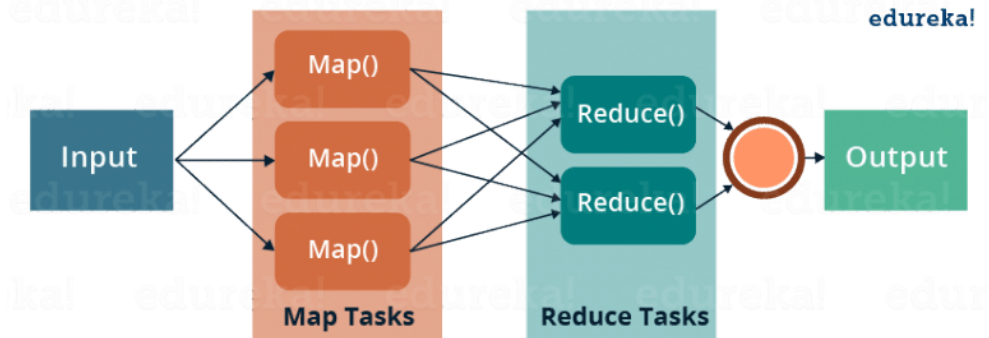


## Module -2 Answers

### 1) Explain with neat diagram the MapReduce execution steps?



MapReduce is a programming framework that allows us to perform distributed and parallel processing on large data sets in a distributed environment.

MapReduce programming enables job scheduling and task execution.

MapReduce consists of two distinct tasks — Map and Reduce.

As the name MapReduce suggests, reducer phase takes place after the mapper phase has been completed. MapReduce program can be written in any language including JAVA, C++ PIPEs or Python

#### Steps for map reduce:

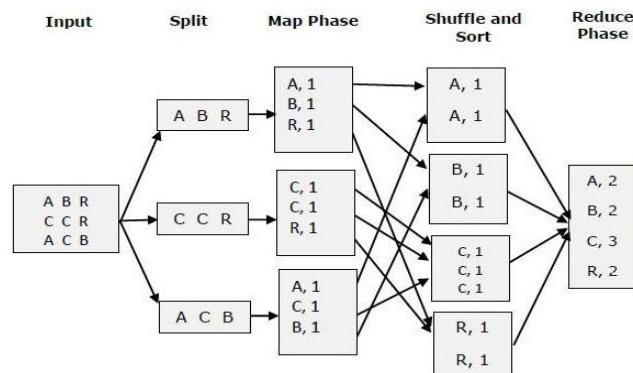
So, the first is the map job, where a block of data is read and processed to produce key-value pairs as intermediate outputs.

The output of a Mapper or map job (key-value pairs) is input to the Reducer.

The reducer receives the key-value pair from multiple map jobs.

Then, the reducer aggregates those intermediate data tuples (intermediate key-value pair) into a smaller set of tuples or key-value pairs which is the final output.

Mapper role :



**Mapper** is a function which process the input data.

The mapper processes the data and creates several small chunks of data.

The input to the mapper function is in the form of ([key, value](#)) pairs,

Each Mapper deals with a single input split.

**Reduce role:**

Reducer is a phase in hadoop which comes after Mapper phase

Phases of Reducer:

There are 3 phases of Reducer in Hadoop MapReduce.

1) Shuffling

2) Sorting

2) Reduce

The shuffle and sort phases occur concurrently. In Shuffling phase, the sorted output from the mapper is the input to the Reducer after shuffling and, sorting, reduce task aggregates the key value pairs.

## 2) Explain Apache Flume architecture with multi agent flow and consolidation network?

a) Apache Flume is for feeding streaming data from various data sources to the Hadoop HDFS or Hive. Apache Flume has a simple architecture that is based on streaming data flows. The design goal of Flume Architecture is,

1. Reliability

2. Scalability

3. Manageability

4. Extensibility

B) Apache Flume provides a distributed, reliable, and available service.

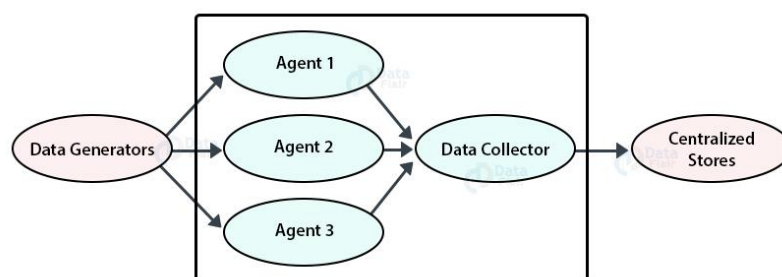
c) Flume efficiently collects, aggregates and transfers a large amount of streaming data into HDFS.

d) Flume enables upload of large files into Hadoop clusters.

e) The features of flume include robustness and fault tolerance. Flume provides data transfer which is reliable and provides for recovery in case of failure.

f) Flume is useful for transferring a large amount of data in applications related to logs of network traffic, sensor data, geo-location data, e-mails and social-media messages.

### Apache Flume Architecture



### 1. Data Generators

Data generators generate real-time streaming data. The data generated by data generators are collected by individual Flume agents that are running on them. The common data generators are Facebook, Twitter, etc.

### 2. Flume Agent

The agent is a JVM process in Flume. It receives events from the clients or other agents and transfers it to the destination or other agents. It is a JVM process that consists of three components that are a source, channel, and sink through which data flow occurs in Flume.

## Flume Agent



Flume agent is composed of three components:

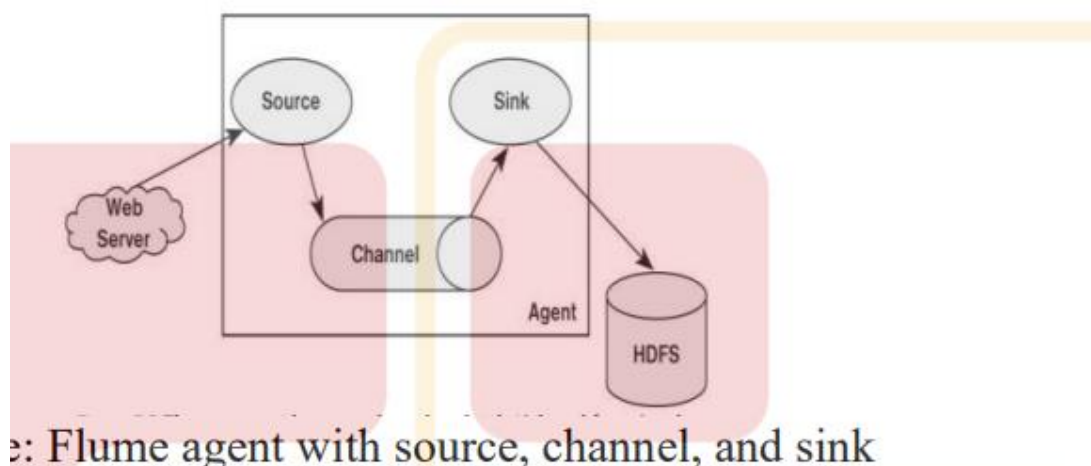
- **Source.** The source component receives data and sends it to a channel. It can send the data to more than one channel. The input data can be from a real-time source (e.g., weblog) or another Flume agent.
- **Channel.** A channel is a data queue that forwards the source data to the sink destination. It can be thought of as a buffer that manages input (source) and output (sink) flow rates.
- **Sink.** The sink delivers data to destination such as HDFS, a local file, or another Flume agent.

### 3. Data collector

The data collector collects the data from individual agents and aggregates them. It pushes the collected data to a centralized store.

### 4. Centralized store

Centralized stores are Hadoop HDFS, HBase, etc.



e: Flume agent with source, channel, and sink

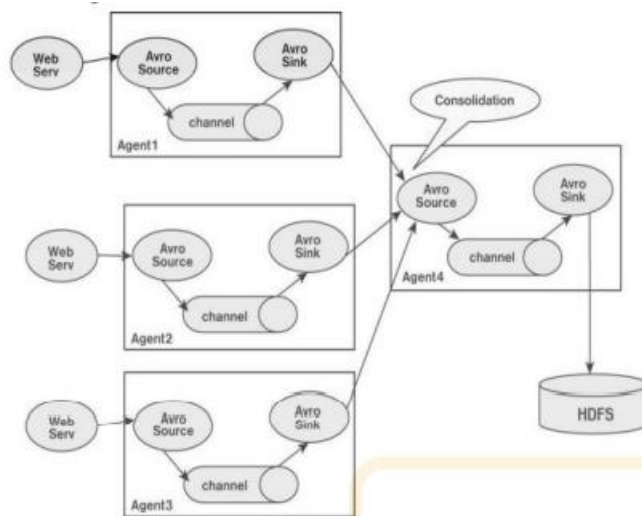


Figure: A Flume consolidation network

3) Explain with a neat diagram Apache Oozie workflow for Hadoop architecture.

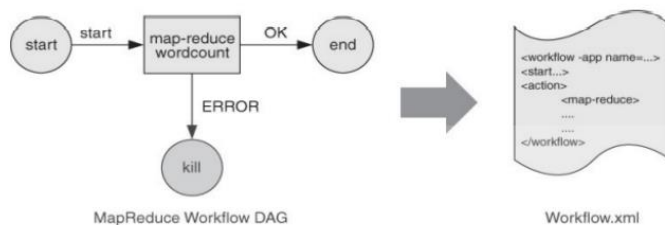


Figure: A simple Oozie DAG workflow

1) An Oozie Workflow is a collection of actions arranged in a Directed Acyclic Graph (DAG)  
 2) Apache Oozie is a workflow scheduler for Hadoop. It is a system which runs the workflow of dependent jobs.

3) Oozie is not a substitute for the YARN scheduler. Oozie provides a way to connect and control Hadoop jobs on the cluster

It consists of two parts:

Workflow engine: Responsibility of a workflow engine is to store and run workflows composed of Hadoop jobs e.g., MapReduce, Pig, Hive.

- **Coordinator engine:** It runs workflow jobs based on predefined schedules and availability of data.

Oozie provides a CLI and a web UI for monitoring jobs.

Oozie runs a basic MapReduce operation. If the application was successful, the job ends; if an error occurred, the job is killed

Oozie is very much flexible, as well. One can easily start, stop, suspend and rerun jobs. Oozie makes it very easy to rerun failed workflows. One can easily understand how difficult it can be to catch up missed or failed jobs due to downtime or failure. It is even possible to skip a specific failed node.

Oozie workflow definitions are written in hPDL. Such workflows contain several types of nodes:

- Control flow nodes define the beginning and the end of a workflow. They include start, end, and optional fail nodes.

An **action node** represents a workflow task, e.g., moving files into HDFS

Action nodes can also include HDFS commands.

**Start Node**, **End Node**, and **Error Node** fall under this category of nodes.

**Start Node**, designates the start of the workflow job.

**End Node**, signals end of the job.

**Error Node** designates the occurrence of an error and corresponding error message to be printed.

Fork/join nodes enable parallel execution of tasks in the workflow. The fork node enables two or more tasks to run at the same time.

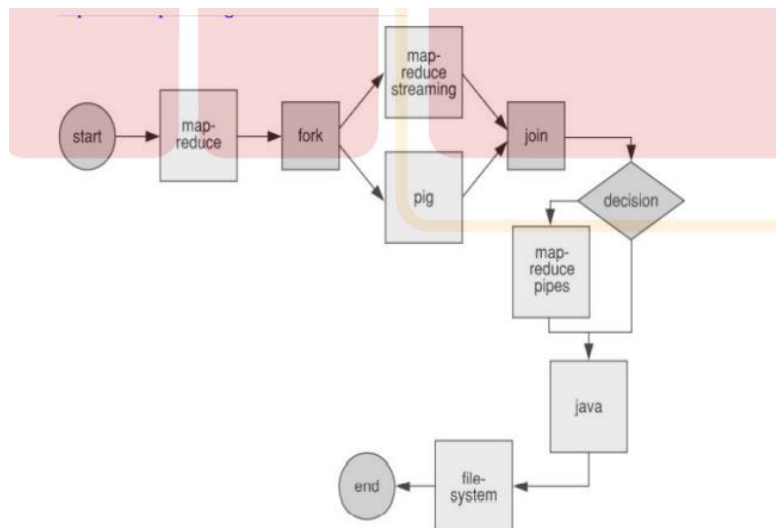


Figure: A more complex Oozie DAG workflow


#### 4) Explain apache hbase?

##### Using Apache HBase

Apache HBase is an open source, distributed, versioned, nonrelational database modeled after Google's Bigtable.

Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.

Important features are:

- Linear and modular scalability
  - Strictly consistent reads and writes
  - Automatic and configurable sharding of tables
  - Automatic failover support between RegionServers
  - Convenient base classes for backing Hadoop MapReduce jobs with Apache HBase tables
  - Easy-to-use Java API for client access
- 

5) explain apache sqoop import and export method with a neat diagram?

19

**Apache Sqoop:**  
Sqoop is a tool designed to transfer data between Hadoop and relational databases.

Sqoop is used to

- import data from a relational database management system (RDBMS) into the Hadoop Distributed File System (HDFS),
- transform the data in Hadoop and
- export the data back into an RDBMS.

**Sqoop Import method:**

The diagram illustrates the Sqoop Import process. It starts with a box labeled 'Sqoop Import' on the left. A dashed arrow labeled '(1) Gather Metadata' points from 'Sqoop Import' to a cylinder labeled 'RDBMS'. From the 'RDBMS', four arrows point to a dashed box labeled 'Hadoop Cluster'. Inside this cluster, there is a 'Sqoop job' section with four 'Map' tasks. Each 'Map' task has an arrow pointing to a document icon in the 'HDFS Storage' section. A dashed arrow labeled '(2) Submit Map-Only Job' points from 'Sqoop Import' to the 'Sqoop job' section. The entire dashed box is labeled 'Hadoop Cluster' at the bottom.

**Sqoop import**  
The data import is done in two steps:

- 1) Sqoop examines the database to gather the necessary metadata for the data to be imported.
- 2) Map-only Hadoop job : Transfers the actual data using the metadata.

1) The imported data are saved in an HDFS directory.

---

prithvi@sdmit.in, dhawalj@sdmit.in Page 19



□ Sqoop will use the database name for the directory, or the user can specify any alternative directory where the files should be populated. By default, these files contain comma delimited fields, with new lines separating different records.

**Sqoop Export method :-**

Data export from the cluster works in a similar fashion. The export is done in two steps :-

1) examine the database for metadata.

2) Map-only Hadoop job to write the data to the database.

Sqoop divides the input data set into splits, then uses individual map tasks to push the splits to the database.

