Apache oozie:-

* Oozie is a job sheduling system integrated with Hadoop.

* used to execute multiple jobs in parallel.

* Integrated spark, Hive, scala

* It is a workflow schedular for Hadoop.

* It is a system which runs the workflow of dependent jobs

* user permitted to create DAG (Directed Acyclic Graph of workflow which can run in parallel and sequential

3 types of jobs

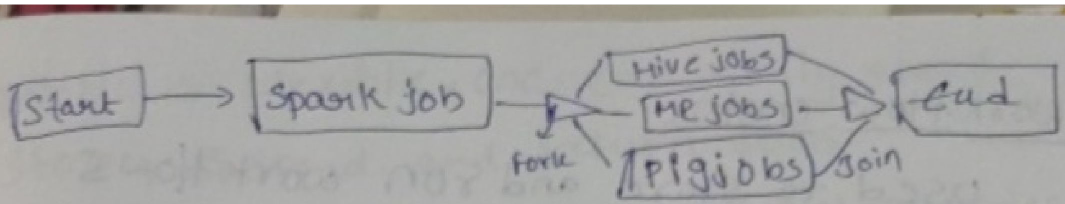
1. Oozie work flow jobs
   - DAG is used to define jobs
     edges specifies action.

2. oozie coordinator jobs
   - work flow jobs triggered by time and data availability.

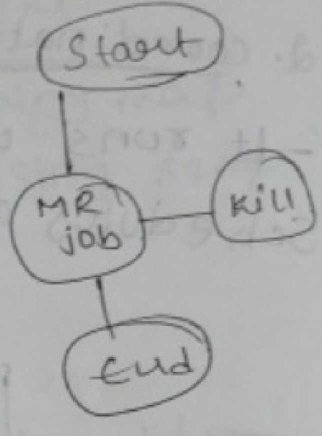
3. oozie bundle.
   - Multiple coordinator.

How oozie works:-

(1) Oozie work flow jobs



- output of the previous action is the input of the present.
- output of the present is the input of the future.



31/3/23

features of oozie:-

*Oozie has client API and command line enterface which can be used to travel, control and monitor jobs from Java application.

*using webservices API can control the jobs from anywhere.

*provision to execute jobs which are schedule to run periodically.

*provision to send email notification upon completion of jobs.

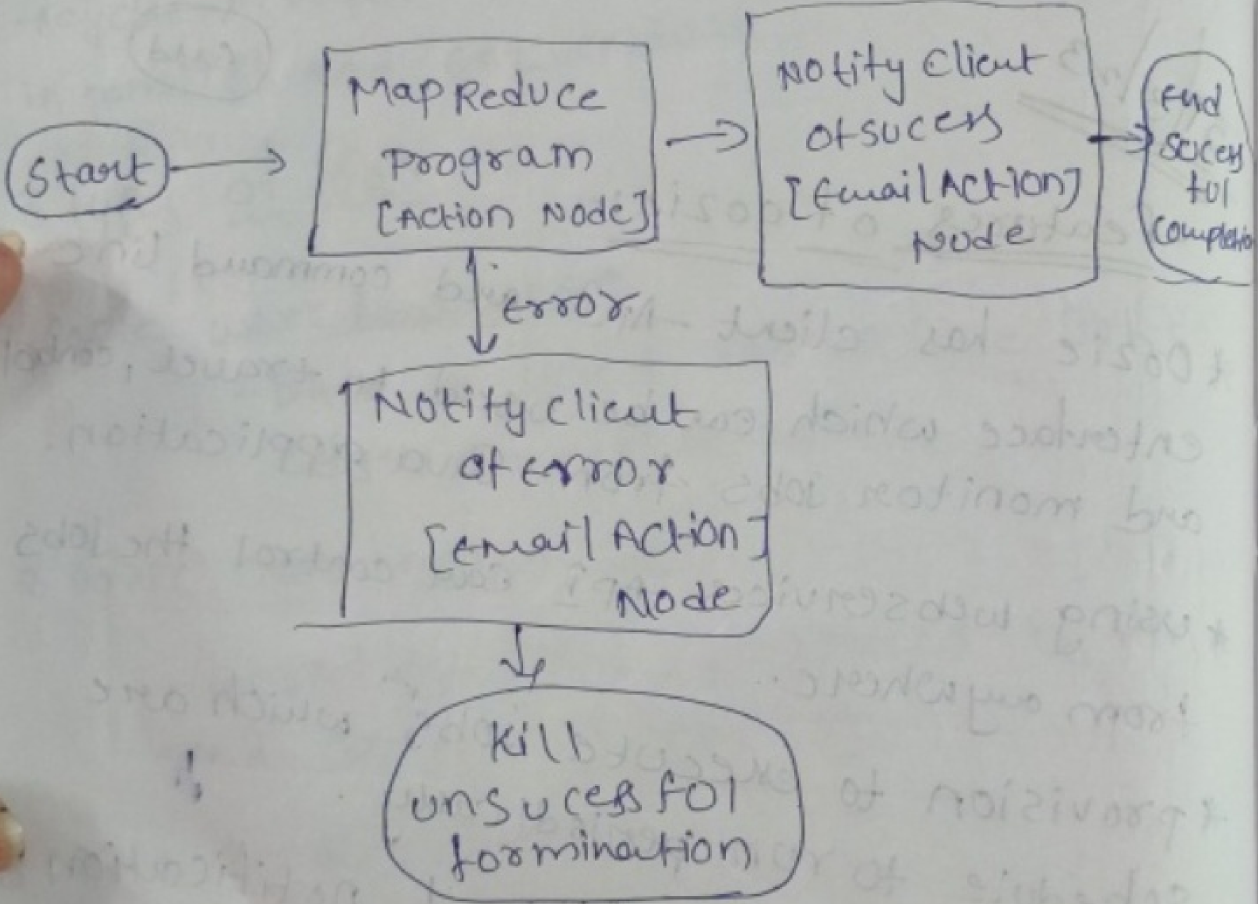
1. work flow engine:

used to store and run work flows composed of hadoop jobs.

-Eg: map reduce, Pig, Hive.

2. coordinator engines:-

- It runs work flow jobs based on predefine schedules and availability of data.



Start
Map Reduce Program [Action Node]
Notify client of sucess [Email Action] Node
End sucess ful completion
error
Notify client of error [email Action] Node
Kill unsucessful termination

*Oozie is scalable and can manage the finally exeevition of thousands of work flow (each consist of dozens of Jobs) in a hadoop cluster.

*Oozie is flexible, one can easily start. stop, suspend and return jobs.

* oozie workflow consist of Action nodes and control flow nodes.

Action node:-

- Represents work flows tasks.

(eg) moving files to HDFS, running map reduce jobs import data using sqoop or running shell script.

*Control Node:-

control work flow execution b/w actions

→ allowing conditional logic where different branches may be followed depend upon the result of the earlier node.

→ Start node:- it used to start the work flow job.

→ end node: it signals the end of the job

→ error node:- it designated to the occurence of the error and notify the error to ~~notify~~ print