APACHE PIG

codes

To get the list of files we use ls

hadoop fs - ls

To create a directory in hadoop mkdir is used

hadoop fs -mkdir samad

To copy the file from local system to HDFS we use this code

hadoop fs - copyFromLocal ' /home/cloudera/Desktop/emp.csv' samad

To get into the pig enviroment

pig

To get the present Working directory

pwd

To change the Working directory

cd samad

TO load the data with Schema

A= load 'emp.csv' using PigStorage(',') as (eid: int, ename: chararray, epos: chararray, esal:int, ecom: int, edpno:int);

To run and display

dump A;

TO load the data without Schema

A2 = load '/user/root/pig/emp.csv’;

To get the structure of the file

describe A2;

To get the rows with edpno =to 20

B = filter A by edpno==20;

To get the rows with edpno =to 20 and in position of MANAGER

B2 = filter A by edpno==20 and epos=='MANAGER’,

To get only 3 rows

C = limit B 3;

To get the data in descending order

D = order C by esal desc;

To Store Data

store D into '/pig/pigout1’ using PigStorage(',’);

Select existing column

E = foreach A generate eid;

Create new column

F = foreach A generate \*, ecom\*2 as Bonus,esal\*5 as Incentive;

Transform columns

G = foreach A generate SUBSTRING(ename,0,4);

Advanced codes

H = foreach A generate $0,$1;

I = group A by edpno;

J = foreach I generate group as edpno, COUNT($1) as count;

K = foreach A generate MAX(A.esal) as maxsal,MIN(A.esal) as minsal, SUM(A.esal) as sumsal, COUNT($1) as count;

L = group A by (edpno, epos)

SPLIT A into B if edpno==10, C if edpno==20, D if epos=='MANAGER';

Joins

A = load '/emp.csv' using PigStorage(',') as (eid:int,ename:chararray,epos:chararray,esal:int,ecom:int,edpno:int);

B = load '/dept.csv' using PigStorage(',') as (edpno:int,epos:chararray,ecity:chararray);

C = JOIN A by edpno,B by edpno;

D = foreach C generate A::eid,B::epos;

E = JOIN A by edpno RIGHT OUTER, B by edpno;

Word Count

lines = load '/plaintext.txt' as (line:chararray);

token = foreach lines generate TOKENIZE(line);

flats = foreach token generate FLATTEN($0);

group\_words = group flats by $0;

count\_word = foreach group\_words generate group as word, COUNT($1) as word\_occurence;