**Launch pig in mapreduce mode**

pig -x mapreduce

-- create a pig relation with schema

nyse = LOAD '/user/hive/warehouse/nyse\_raghu.db/nysedaily/nysedaily\_2016.csv'

using PigStorage(',')

AS

(stock\_date:chararray,

stock\_symbol:chararray,

stock\_price\_open:double,

stock\_price\_close:double,

stock\_price\_low:double,

stock\_price\_high:double,

stock\_volume:double);

describe nyse;

-- Generate columns and limit the rows

nyse\_cols = FOREACH nyse GENERATE stock\_symbol, stock\_date, stock\_price\_close;

nyse\_cols\_limit = LIMIT nyse\_cols 10;

DUMP nyse\_cols\_limit;

-- create a pig relation without schema

nyse1 = LOAD '/user/hive/warehouse/nyse\_raghu.db/nysedaily/nysedaily\_2016.csv'

using PigStorage(',');

describe nyse1;

-- Generate columns and limit the rows for the pig relation without schema

nyse\_cols1 = FOREACH nyse1 GENERATE $1 AS stock\_symbol:chararray, $3 AS stock\_price\_close:double;

nyse\_cols\_limit1 = LIMIT nyse\_cols1 10;

DUMP nyse\_cols\_limit1;

Note : All built in functions in apache pig are case sensitive

--Filter the records

nyse\_filtered = FILTER nyse by stock\_symbol == 'GE';

nyse\_limit = LIMIT nyse\_filtered 10;

dump nyse\_limit;

--Group the records

nyse\_group = GROUP nyse BY stock\_symbol;

When the pig relation is grouped it results in 2 columns:

1. The column on which we are grouping named as **group**
2. The name of the bag is same as the name of the relation which we are grouping - **nyse**

--Count the grouped records

nyse\_cnt = FOREACH nyse\_group GENERATE group, COUNT(nyse);

dump nyse\_cnt;

Here we are displaying the group name i.e stock\_symbol and counting the number of tuples inside the bag nyse for every stock\_symbol

--get the max stock\_price\_open for the grouped records

nyse\_max = FOREACH nyse\_group GENERATE group, MAX(nyse.stock\_price\_open);

nyse\_limit = LIMIT nyse\_max 10;

--store the max stock\_price\_open for the grouped records in a file

store nyse\_max into '/user/raghavendra/nyse\_pig\_output';

in fs shell check the output file contents:

hadoop fs -ls /user/raghavendra/nyse\_pig\_output

hadoop fs -cat /user/raghavendra/nyse\_pig\_output/part\* | head -n 10

--Sorting in pig use desc for descending order sort

nyse\_sort = ORDER nyse BY stock\_date;

Task :

Get the stock closing and opening price for GE stock symbol on 5th may 2016.

Sort the data in the increasing order of opening price

nyse\_filtered = FILTER nyse by stock\_date == '2016-05-05' and stock\_symbol == 'GE';

nyse\_cols = FOREACH nyse\_filtered GENERATE stock\_price\_close, stock\_price\_open;

nyse\_ordered = order nyse\_cols by stock\_price\_open;

dump nyse\_ordered;

nyse\_cols = FOREACH nyse GENERATE stock\_price\_close, stock\_price\_open

nyse\_filtered = FILTER nyse\_cols by stock\_date == '2016-05-05' and stock\_symbol == 'GE'

nyse\_ordered = order nyse\_filtered by stock\_price\_open

dump nyse\_ordered

**How grouping works in pig**

nyse sample data

(04/01/16,AAL,41.290001,40.91,40.299999,41.349998,12037200)

(22/01/16,AAL,40,40.16,39.57,40.720001,7247000)

(25/01/16,AAL,40.02,38.880001,38.82,40.16,10113200)

(04/01/16,AAP,149.25,152.240005,147.509995,154.479996,1935300)

(05/01/16,AAP,152.350006,151.199997,149.889999,153.789993,1127700)

When grouped as below

nyse\_group = GROUP nyse BY stock\_symbol;

below will be output of the nyse\_group:

(AAL,{ (04/01/16,AAL,41.290001,40.91,40.299999,41.349998,12037200), (22/01/16,AAL,40,40.16,39.57,40.720001,7247000) (25/01/16,AAL,40.02,38.880001,38.82,40.16,10113200)})

(AAP,{ (04/01/16,AAP,149.25,152.240005,147.509995,154.479996,1935300), (05/01/16,AAP,152.350006,151.199997,149.889999,153.789993,1127700)})

When COUNT is appled on nyse\_group

nyse\_cnt = FOREACH nyse\_group GENERATE group, COUNT(nyse);

(AAL,3)

(AAP,2)

Counting all records in pig

nyse\_group = GROUP nyse BY null;

nyse\_count = FOREACH nyse\_group GENERATE COUNT(nyse);

dump nyse\_count

pig in local mode

pig -x local

nyse = LOAD '/home/manipalproshare/batch7/nysedaily\_2016.csv'

using PigStorage(',')

AS

(stock\_date:chararray,

stock\_symbol:chararray,

stock\_price\_open:double,

stock\_price\_close:double,

stock\_price\_low:double,

stock\_price\_high:double,

stock\_volume:double);

nyse\_cols = FOREACH nyse GENERATE stock\_symbol, stock\_date, stock\_price\_close;

nyse\_cols\_limit = LIMIT nyse\_cols 10;

DUMP nyse\_cols\_limit;

Word count in pig

text = LOAD '/user/raghavendra/modi\_speech' AS(line:Chararray);

words\_text = FOREACH text GENERATE FLATTEN(TOKENIZE(line,' ')) AS word;

grouped\_words = GROUP words\_text BY word;

wordcount = FOREACH grouped\_words GENERATE group, COUNT(words\_text.word);

dump wordcount;

Additional demos for self study

removing the header

launch pig in local mode on the VM

petrol = LOAD '/home/cloudera/Desktop/Labs/petrol.txt' using PigStorage(',');

--to find out the names of the header

petrol\_limit = LIMIT petrol 1;

-- remove the header row using filter and store it in a file

petrol\_filtered = FILTER petrol by $0 != 'District.ID';

store petrol\_filtered into '/home/cloudera/Desktop/Labs/petrol\_cleaned';

Below query results in an error

nyse\_cols = FOREACH nyse GENERATE stock\_symbol, stock\_volume;

nyse\_group = GROUP nyse\_cols BY stock\_symbol, SUM(stock\_volume);

nyse\_filtered = FILTER nyse\_group by stock\_volume >500;

DUMP nyse\_filtered;

It can be resolved using the below, where the aggregate is further filtered using the filter operator

nyse\_cols = FOREACH nyse GENERATE stock\_symbol, stock\_volume;

nyse\_group = GROUP nyse\_cols BY stock\_symbol;

sum\_stockvol = foreach nyse\_group generate group as stock\_symbol, SUM(nyse\_cols.stock\_volume) as total\_volume;

nyse\_filtered = FILTER sum\_stockvol by total\_volume >5000;

The above can be tried without using a pig schema

nyse1 = LOAD '/user/hive/warehouse/nyse\_raghu.db/nysedaily/nysedaily\_2016.csv'

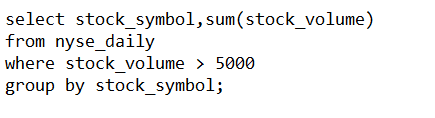
using PigStorage(',');

nyse\_cols = FOREACH nyse1 GENERATE $1 as stock\_symbol, $6 as stock\_volume;

nyse\_group = GROUP nyse\_cols BY stock\_symbol;

sum\_stockvol = foreach nyse\_group generate group as stock\_symbol, SUM(nyse\_cols.stock\_volume) as total\_volume;

nyse\_filtered = FILTER sum\_stockvol by total\_volume >5000;



nyse\_daily = load ….

nyse\_cols = FOREACH nyse\_daily GENERATE stock\_symbol, stock\_volume;

nyse\_fil = FILTER nyse\_cols BY stock\_volume > 5000;

nyse\_grp = GROUP nyse\_fil by stock\_symbol;

nyse\_sum = FOREACH nyse\_grp group as stock\_symbol, sum(nyse\_fil.stock\_volume);

dump nyse\_sum;