Aim: To Perform Simple analytics using Map Reduce

```
Program :-
```

Java io, IOException; import

java util . Iterator; import

java. util. regex. Mathoher; import

java. util. regex. Pattern; import

org, apache, hadoop, fs. Path; import

import org. apache. hadoop, io. Intwritable;

import org. apache hadoop, io . Text;

import org. apache, hadoop, mapred. Job Couf;

import org. apache, hadoop, mapreduce, Job;

org, apache, hadoop, mapreduce, Mapper; import

org, apache, hadoof, mapreduce, Reducer; import

org, apache, hadoop, morreduce, lib, input. File import

Input Format;

org. apache. hadoop, magreduce, lib. output. import

FileOutputformat)

Import org. apache hadoor, util. Generic Options Parson;

Public class Web Log Message Size Aggregator

3

Public static final Pattern httplog Pattern = Pattern

compile ("([1115]+) -- 11[(+)11] 1"([1115]+)

(/[1/15]\*) HTTP/[1/15]+1"[1/15]+([0-9+)");

Public static class AMapper extends Mapper Cobject, Text, Text, Jutwritable>

9

```
Public void map Cobject Key, Text value, Context
                  context throws IDException, Interrupted
                          -Exception
     Matcher matcher-httplogfattern, matcher Calue.
                             to String();
      if (matcher, matches (1)
{ jut Size = Juteger, parseJut (matcher, group (5));
          context, write (new Text ("msg Size"), new
       Jutwritable ($30);
 Public Static class AReducer extends Reducer < Text,
          Int Writable, Test, Int Writable)
      Public void reduce (Test Key, Iterable < Juliantables
  -5
          Values, Context context) throws IDExceptions
                             Interrupted Exception
          double tot=9;
           int count=6;
           lut min = Integer, MAX_VALUE;
           jut max = 0;
           Iterator < Intluvitable > iterator = values iterator
         while (itexator has Next ())
              jut value = iterator, next(), get();
                 tot = tot + value;
```

```
Count +ts
     if (value < min)
           min = value;
      if (value > max)
         max=value;
   contextrurite (new Text ("Mean"), new Interitable
                     (ciut) tot (count));
    context, write (new Text ("Moz"), new Jutwritable (mi)
    context, write (new Text("Min"), new Intwritable (min))
Public static void main (String[] args) throws
                                 Exception
    JobCouf cout = new JobCouf();
    String[] other Avgs = new Generic Options Parson
               (couf, args). get Remaining Args );
     if (other Args, length != 2)
          Systemierri println ("Usage: <in> <out>");
           System, exit (2);
     Job job = new Job Cront, "Weblog Message Size Aggregator)
     job set Jan By Class (Weblog Mossage Size Agg regator while
      job set Mapper Class (AMarper, class);
```

job. Set Reducer Class (AReducer, class);

Job. Set Map Output Key Class (Text. class);

Job. Set Map Output Value Class (Int Writable. class);

File Tuput Format, add Tuput Path (Job, New Path (other Args [0]));

File Dutput Format. Set Output Path (Job, new Path (Other Args EII);

System. exit ( job. WaitFor Completion (true) ?0:1);

3

### steps to execute the Hadoop application:

- 1) gedit Weblog Message Size Aggregator, java
- (2) export HADOOP\_CLASSPATH = \$ \$ JAVA\_ HOME 3/116/ tools, jour
- 1) jar of weijar Weblog MessageSise
- 3 hadoop com. suntools, Javac. Main Weblog Message Sise Aggregator. Java.
- (4) Jan of wo jar weblog Message Size Aggregator to class
- 3 hadoop fs -mkdir P/wc
- @ hadoop is -copy From Local imput (wel
- hadoor jar we jar weblog Message Size Aggregator / we linear /word/out/6
- (8) hadoop fs -15 /word/out16
- @ hadoop fs -cat /word/out/6/part-r-00000

the state of the second Output: Mean 1150 I) him all the state of the state 6823936 Max (0) min MANUAL PROPERTY SERVICES STATES FROM THE PARTY OF THE

```
Experiment -17
1/12/21
Aimi- Performing Group-By using Map Reduce.
Program :
           java 10. IOException;
 Import
           java. util. Iterator;
  import
           java. util. reget. Matcher;
   import
            java. util. regex. Pattern;
   import
           org. apache, hadoop, fs. Path;
  import
           org. apache. hadoop, io. Jutwritable;
   import
            org. apache, hadoop, io, Text;
   import
           org. apache. hadoop, mapred. Job Conf;
   import
           org. apache, hadoop, mapreduce. Job;
   import
    import org, apache, hadoop, mapreduce, Mapper;
    import org. apache, hadoop, mapreduce, Reducer,
    import org. apache hadoop, mapreduce. 1:b, input, FileIn
                              -put Format)
    import org. opache, hadoop, morpreduce, lib. output.
                            FileOutput Format;
    import org. opache, hadoop, util, Generic OptionsPonser;
    Public class Weblog Hits By Link Processor
```

Public Static final Pattern httplogfattern = Pattern · compile ("([1115]+)--11[(+)11] \"([1115]+)(/[1 115]\*)HTTP/[115]+\"[115]+([0-9]+)");

Public static class AMapper extends Mapper <object, Text, Text, IntWritable> 5

```
Private final static Intluvitable one = new
                                 IntWritable(1);
    Private Text word = new Text();
  Public void map (Object key, Text value, Contest contest)
            throws IDException, Interrupted Exception
    Matcher matcher = httplog fattern, matcher (value, to
                                String();
      if (matcher. matches())
          String link Url = matcher, group (4);
           word set (link Url);
          condext write (word, one);
     3
   3
        static class A Reducer extends Reducer < Text,
Public
       Inturitable, Text, Inturitable>
     Private IntWritable result = new IntWritable();
 f
     Public void reduce (Text key, Iterable < Irt Writable)
                values, (outest context) throws IO Exception
                   Interrupted Exception
           int sum = 0;
           for (Int writable val : values)
                sum + = val. get ();
            result. Set (Sum);
          contents write (Key, result);
```

```
Public static wid main (String[] args) throws
                             Exception
JobCout cont = new JobConf();
String[] other Args = new Generic Options Parser (cont)
                 args). get Remaining Args ();
if (other Args leugth != 2)
     System. exr. println ("Usage : Lin > cout>");
 2
      System, exit(z);
 2
Job Job = new Job (Conf, "weblog Hits By Link Processor").
job. set Jor By Class ( web by Hits By Link Processor, class);
job. setMapper Class (AMapper. class);
  job. set Reducer Class (A Reducer, class);
  Job. Set Mop Output Key Class (Text. class);
  job set Map Output Value class (Interitable class);
  File Input Format, add Input Path (job; new Path (other
                         -Args[0])),
  FileOutput Format, set Output Path Ciob, new Path
                       (other Args [1]);
  System, exit (job, wait tor Completion (true) ? 0:1);
```

Steps to execute the Hadoop application:
1) gedit webbg Hits By Link Processor, Java - HOME 3/(16)
1001s, jar tools, jawac, Main
Weblog Hits By Link Processor, java.  Weblog Hits By Link Processor to Java  Weblog Hits By Link Processor to Java
6 hadoop fs -mkdir -p /wc 6 hadoop fs -copy From Local input /wc
(a) hadoop jar uc. jar weblog Hits By Link Processor  (we lingut /word/out 17  (a) hadoop fs - 1s /webin / word/out 17
@ hadoor to -cat /word/out17/part-r-00000
Indowns/www/summary.html 1
/~downs/www/query. html 3 /~downs/www/index. html 2

output:	1
/~downs/www/query. html	3
Indowns/www/index. html	2
Indowns / www	3
Indowns/ www/	41
Indowns/launchitep, gif	37

Indowns/harvest-1.2/brokers/www/query.html 8

```
Experiment -18
8/12/21
```

webloghits using Map Reduce. Aim: Calculating

```
Program:
```

java.io. IO Exception; import

java. util. regex. Matcher; import

jour util regex lattern; import

org. apache, hadoop. fs, Poth; import

org. apache, hadoop, io, Jut Writable; import

org, apache, hadoop, io, Text; import

org, apache, hadoop, morred. Job Cout; import

import org. apache, hadoop, mapreduce, Job;

org, apache. hadoop, mapreduce. Mapper,

import org. apache. hadoop, mapreduce. Reducer; import

org. apache, hadoop, mapreduce. lib. input. import

File Input Format;

org. apache. hadoop. mapreduce. lib. output. import

File Output For mat;

import org, apache, hadoop, util, Generic Options Parson

Public class webloghits

Public static final Pattern httplog Pattern = Pattern · compile ("([NIS]+)-- 11[(·+)11]1"([NIS]+)

C/[^115]\*) HTTP/[^115]+1"[^115]+

(EO-9]+)");

Public static das AMapper extends Mapper 20bject, Text, Inturitable, Int writable>

```
Private final static Interitable one =
                 new Intwritable(1);
Public void may (Object Key, Text value, Context
        context) throws IDException, Interrupted
                   Exception
  Matcher matcher = httplog Pattern, matcher (value.
                        toString());
  if (mot cher, matches())
       int Size = Integer, Parse Int (matcher, group(5));
   context, write (new Intwritable (5230 /1024),
 3 and a series of the series of the series
Public Static class A Reducer extends Reducer
 < Jutwritable, Jutwritable, Jutwritable, Jutwritable
 5
   Public void reduce (Int Writable Key, Iterable
     < Interitable > values, Content context) throws
     IOException, Interrupted Exception
        int sum = 0;
        for ( Int writable val : values)
       { Sum += val.get();
        Context, write (key, new Intwitable (sum);
      3
```

```
Public static void main (String[] args) throws
Exception
  JobCout conf = new JobConf ();
  String [7 other Args = new Generic Options Parsex
            (cont, args) get Remaining Args ();
   if (other Args, length 1=2)
   { System. err, Println ("Usage: <in> <out>"}
       System. Brit (2);
  Job job = new Job (cout, "weblog Message size us
                - Hits Processor");
   job. set Ex By Class (Web loghits, class);
    job. Set Mapper Class (AMapper . class);
    Job , Set Reducer Class (A Reducer Class);
    Job . set Map Out put Key Class (Int writable . class);
    Job: Set Map Output Value Class (Int Writable, class);
    File Input Format add Input Path ( Job, new Path
                   Cother Args [0]) L'
   FileOutput Format, setOutput Path (job, new fath
                   (otherArgs[i]);
  System. exit ( isb. wait For (ampletion (true) ?0:1);
     accessing, txt - which contains website
             licks and Haddreges.
```

	2 30
Execution	
1 gedit	Wable

oghits, joura

@ export HADOOP\_CLASSPATH = \$ & JAVA - HOME & / lib)

tools, jar.

3 hadoop com. sunitools, javac Main Webbyhitsipu

4) Jar of wo.jar webloghits \* . class

3 hadoor fs -mkdir -Plwc

@ hadoop fs -copy From Local access log /wc

Thadoop jar we, jar webloghits /welaccess bg /word/out19

(8) hadoop fs -ls (word/out19

9 hadoop to -cat /word/out19/Part-r-00000

#### output:

12

7	30266
13	10164
14	7243
15	7739
16	3381
17	
18	5608
19	5844
20	7465
21	1555
22	1459
2.3	1622
24	9 527
29	1284
26	2691
27	6008
28	3106
29	6948
	3621 and Soon
U-às j	
890 J 7 31 3	
Sall Ell	
R P Fine.	
The same of the sa	

## 8 lieles Experiment -19

Aim it plotting the Hadoor regults using GNU plot

Download the results of the last recipe to a

local computer by running the following command

The hadoop fs -get /word/output/7/part - Y-00000

-) gruplot (Kado godit http-fregdist. plot

Set terminal png

set output " Frequency fog"

Set title "Frequency Distribution of Hits by LRL"

Set ylabel "Number of Hits"

Set xlabel "urs"

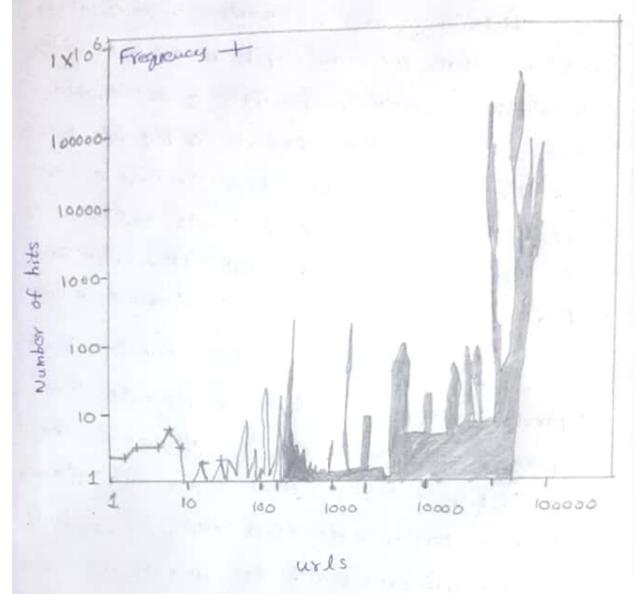
Set key left top

Set log y

Set log x

plot "2.data" using 2 title "Frequency" with lines Points.

# Frequency Distribution of Hits by URL



Experiment -20

Aim: Calculating histograms using Map Reduce.

Histograms are a common vigualization technique that gives an empirical extimate of the Probability density function (Pdf) of a variable. Histograms are well-suited to a big data emin -nment, because they can reduce the Size of raw input data to a vector of counts. Each countis the number of observations that falls within each of a Set of countiguous, numeric jutervals or birs.

Separately on multiple chunks of the data. Then mapreduce sums the counts from all chunks. The map function and reduce function are both extremely simple in this example. Nevertheless, you can built flexible visualizations with the Summary Information that they collect.

### Vigualize Results:

range of the data in gruplet by using fallowing commands having input from previous program output 1.c.,

-> hadoop fs -get /word/out/9/Part-Y-00000 hist data

set term png medium

Set output "histogram. png"

Set xlabel "number of bits"

Set ylabel "urss"

Set style histogram clustered gap 2

Set style histogram columnstacked

Set style histogram columnstacked

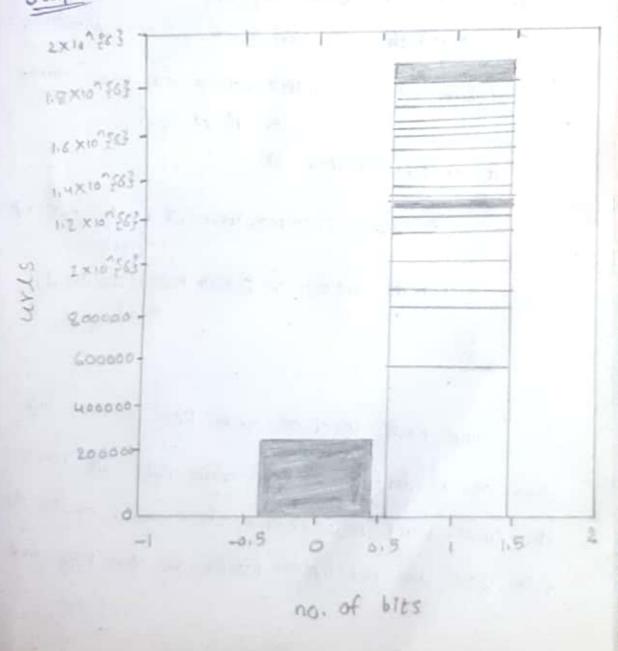
Set style data histograms

Set style data histograms

Set style fill solid 1.0 border-1

Plot "histogram" using 1, "histograms 2





15/12/21

Experiment -21

Aim: - Calculating scatter plats using map Reduce

Another useful tool while analyzing data

15 a Scatter plot: scatter plot is used to find the

relationship between two measurements (dimension).

It plots the two dimensions against each other.

The following code segment shows the code for the mapper.

Public void map (Object Ked, Text value, Context

Context) throws InException, Interrupted Exception

E matcher matcher = httplogPattern. matcher (value.

to String(1);

if (matcher, matches (1)

int size = Integer, parse Int (matcher, group

(5));

context. write (new IntWritable (size /1024))

one);

3

9

Map task receives each line in the log files as a different key-value Pair. It porse the lines using regular expressions and emits the file size as 1024-bytes blocks as the key and

one as the values. Then, Hadoop collects the Key-value pairs sorts them, and then invokes the reducer once for each key. Each reducer walks through the values and calculates the count of page acresses for each file size.

Public void reduce(IntWritable Key, Iterable values,

Context context) throws IO Exception, Interrupted

- Exception

for (Int writable val: values)

{ Sum + = val. get();

}

context. write (Key, new Int Writable (Sum));

The following commands are used for plotter graph between the Size of the web pages and the number of hits received by the web page in gruplet.

Set output "Hitsby Message"

Set title "Hits by Size of the Message"

Set ylabel "no of Hits"

Set xlabel "Size of the message"

Set key left top

Set log y

Set log X

Plot "hist, data" using 1:2 title "2 Node" with

Points

Output:

Hits by Size of the message

