(Hao Zhang, WustlKey: h.zhang633, ID: 452003), (Hanming Li, WustlKey: lihanming, ID: 451802)

Problem1:

(a) The commend line is: $ head -n 100 /home/training/training_materials/analyst/exercises/analyze_ads/ad_data1/part-m-00000 > test_ad_data.txt

(b) The PIG script:

```
-- TODO (A): Replace 'FIXME' to load the test_ad_data.txt file.

data = LOAD 'test_ad_data.txt' AS (campaign_id:chararray,
        date:chararray, time:chararray,
        keyword:chararray, display_site:chararray,
        placement:chararray, was_clicked:int, cpc:int);

-- TODO (B): Include only records where was_clicked has a value of 1
val = FILTER data BY (was_clicked == 1);

-- TODO (C): Group the data by the appropriate field
gro = GROUP val BY display_site;

/* TODO (D): Create a new relation which includes only the
 *         display site and the total cost of all clicks
 *         on that site
 */
gen = FOREACH gro GENERATE group, SUM(val.cpc) AS cost;

-- TODO (E): Sort that new relation by cost (ascending)
sor = ORDER gen BY cost;

-- TODO (F): Display just the first three records to the screen
t = LIMIT sor 4;
DUMP t;
```

The display information is like:

```
(diskcentral.example.com,68)
(megawave.example.com,96)
(megasource.example.com,100)
(salestiger.example.com,141)
[training@localhost analyze_ads]$
```

(c) The PIG script is:

```
-- TODO (A): Replace 'FIXME' to load the test_ad_data.txt file.

/*data = LOAD 'test_ad_data.txt' AS (campaign_id:chararray,
 *         date:chararray, time:chararray,
```

```
 *          keyword:chararray, display_site:chararray,
 *          placement:chararray, was_clicked:int, cpc:int);
 */
data1 = LOAD '/dualcore/ad_data1/part-m-00000' AS (campaign_id:chararray,
        date:chararray, time:chararray,
        keyword:chararray, display_site:chararray,
        placement:chararray, was_clicked:int, cpc:int);
data2 = LOAD '/dualcore/ad_data2/part-r-00000' AS (campaign_id:chararray,
        date:chararray, time:chararray,
        keyword:chararray, display_site:chararray,
        placement:chararray, was_clicked:int, cpc:int);
data = UNION data1, data2;
--STORE data INTO '/dualcore/low_cost_sites';
-- TODO (B): Include only records where was_clicked has a value of 1
f = FILTER data BY was_clicked == 1;

-- TODO (C): Group the data by the appropriate field
gro = GROUP f BY display_site;

/* TODO (D): Create a new relation which includes only the
 *         display site and the total cost of all clicks
 *         on that site
 */
gen = FOREACH gro GENERATE group, SUM(val.cpc) AS cost;

-- TODO (E): Sort that new relation by cost (ascending)
sor = ORDER gen BY cost;

-- TODO (F): Display just the first three records to the screen
t = LIMIT sor 4;
--DUMP t;
STORE t INTO '/dualcore/low_cost_sites';
```

The result is:

```
ne.util.MapRedUtil - Total input paths to process : 1
(bassoonenthusiast.example.com,1246)
(grillingtips.example.com,4800)
(footwear.example.com,4898)
(coffeenews.example.com,5106)
2017-04-18 17:22:23,518 [main] INFO  org.apache.hadoop.conf
ation - fs.default.name is deprecated. Instead, use fs.defa
```

(Hao Zhang, WustlKey: h.zhang633, ID: 452003), (Hanming Li, WustlKey: lihanming, ID: 451802)

Problem2:

(a) The PIG script is:

```
data1 = LOAD '/dualcore/ad_data1/part-m-00000' AS (campaign_id: chararray, date: chararray,
time: chararray,  keyword: chararray, display_site: chararray, placement: chararray, was_clicked:
int, cpc: int);
data2 = LOAD '/dualcore/ad_data2/part-r-00000' AS (campaign_id: chararray, date: chararray,
time: chararray,  keyword: chararray, display_site: chararray, placement: chararray, was_clicked:
int, cpc: int);
data = UNION data1, data2;

gro = GROUP data BY keyword;

cou = FOREACH gro GENERATE group, SUM(data.cpc) AS number;

dsor = ORDER cou BY number DESC;

DUMP dsor;
```

(b) The PIG script is:

```
data1 = LOAD '/dualcore/ad_data1/part-m-00000' AS (campaign_id: chararray, date: chararray,
time: chararray,  keyword: chararray, display_site: chararray, placement: chararray, was_clicked:
int, cpc: int);
data2 = LOAD '/dualcore/ad_data2/part-r-00000' AS (campaign_id: chararray, date: chararray,
time: chararray,  keyword: chararray, display_site: chararray, placement: chararray, was_clicked:
int, cpc: int);
data = UNION data1, data2;

fil = FILTER data BY was_clicked == 1;

gro = GROUP fil BY keyword;

cou = FOREACH gro GENERATE group, SUM(data.cpc) AS number;

dsor = ORDER cou BY number DESC;

t = LIMIT dsor 3;

DUMP t;
```

The top-3 results are:

```
he.util.MapReduTil - Total
(PRESENT,165606)
(TABLET,106509)
(DUALCORE,95124)
2017-04-18 17:43:59,980 [m
```

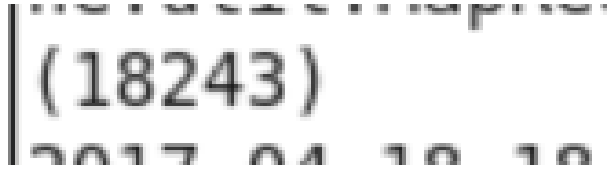
(Hao Zhang, WustlKey: h.zhang633, ID: 452003), (Hanming Li, WustlKey: lihanming, ID: 451802)

Problem3:

(a) The PIG script is:

```
-- Load only the ad_data1 and ad_data2 directories
data = LOAD '/dualcore/ad_data[12]' AS (campaign_id:chararray,
        date:chararray, time:chararray,
        keyword:chararray, display_site:chararray,
        placement:chararray, was_clicked:int, cpc:int);

-- Include only records where the ad was clicked
clicked = FILTER data BY was_clicked == 1;

-- A: Group everything so we can call the aggregate function
gro = GROUP clicked ALL;

-- B: Count the records
cou = FOREACH gro GENERATE COUNT(clicked.data);

-- C: Display the result to the screen
DUMP cou;
```

The total clicked received is:



(Hao Zhang, WustlKey: h.zhang633, ID: 452003), (Hanming Li, WustlKey: lihanming, ID: 451802)

Problem4:

pyspark

```
mydataaa = sc.textFile( "hdfs:/loudacre/weblogs").map(lambda line:line.split(' ')).map(lambda fields:(fields[0]+"/"+fields[2])).distinct()

mydataaa.collect()
```

```
[u'3.94.78.5/69827',
 u'3.94.78.5/69827',
 u'19.38.140.62/21475',
 u'19.38.140.62/21475',
 u'129.133.56.105/2489']
```

Problem 5:

(a):64978

(b):

Command to submit the job:

spark-submit --master local CountJPGs.py /loudacre/weblogs

The count is 64978.

The Driver program is executed on the client side. The process is happened locally. The result is stored locally.

(c):

Command to submit the job:

spark-submit --master yarn-client CountJPGs.py /loudacre/weblogs

The Driver program is executed on the client side. The process is happened in executors of data nodes. The result is stored in hdfs.

(Hao Zhang, WustlKey: h.zhang633, ID: 452003), (Hanming Li, WustlKey: lihanming, ID: 451802)

(d):

• Stages are operations that can run on the same data partitioning in parallel across executors/nodes

• Tasks within a stage are operations executed by one executor/node that are pipelined together

Only 1 stage, 311 tasks were executed in the job.

(e) What is pipelining: when possible, Spark will perform sequences of transformations by row so no data is stored.

map() and fiter() can be pipelined together.