Adrien Xie xie.z 451852
Guangda Ouyang 467428

Problem 1

(a) Two ways of generating sample dataset

1. We can use the first or the last 100 entry of the whole data set as the sample dataset. We have done this many times before in lab using UNIX commend “head –n 100” for the first 100 entries or “tail –n 100” for the last 100 entries.
2. In pig we can use the keyword LIMIT to limit the number of tuples from a relation. So we can load the whole dataset first and use limit to use only a small number of entries and use STORE to save the subset of the dataset.

The reason why running pig using local subset of the dataset is faster is because essentially PIG will generate a MapReduce job and running the pig script on the cluster with full dataset is equivalent to execute a complete MapReduce job on the entire dataset, which will takes a really long time. Therefore running pig using local subset of the dataset is faster.

(b) Command:

$hadoop fs –cat /dualcore/ad_data1/part* | head -100 > test_ad_data.txt

This command will generate a local subset of the ad data

(c) Four least costly sites on test data

(diskcentral.example.com,68)

(megawave.example.com,96)

(megasource.example.com,100)

(salestiger.example.com,141)

(d) Four least costly sites on whole data set

(bassoonenthusiast.example.com,1246)

(grillingtips.example.com,4800)

(footwear.example.com,4898)

(coffeenews.example.com,5106)

Adrien Xie xie.z 451852
Guangda Ouyang 467428

Problem 2

(a) Please refer to the files in the SVN repository.
(b) Top 3 results:

(TABLET,3193033)

(DUALCORE,2888747)

(DEAL,2717098)

Adrien Xie xie.z 451852
Guangda Ouyang 467428

Problem 3

(a) Please refer to the files in the SVN repository
(b) The lowest click through rates are:

(bassoonenthusiast.example.com,10007)

(grillingtips.example.com,17343)

(butterworld.example.com,19003)

To obtain five significant digits in integer calculation, total number of clicks is first multiplied by one million. Therefore the actual click through rate can be obtained by dividing the number by the multiplication factor, so the corresponding click through rate is 1.0007%, 1.7343% and 1.9003%.

Adrien Xie xie.z 451852
Guangda Ouyang 467428

Problem 4

(a) Create an RDD from the activation dataset

files="/loudacre/activations"

activationFiles = sc.wholeTextFiles(files)

wholeTextFiles read a directory of text files from HDFS, a local file system (available on all nodes), or any Hadoop-supported file system URI. Each file is read as a single record and returned in a key-value pair, where the key is the path of each file, the value is the content of each file.

(copied from spark documentation: https://spark.apache.org/docs/2.2.0/api/java/org/apache/spark/SparkContext.html#wholeTextFiles-java.lang.String-int-)

Therefore the first value are the keys which is the path to each file and the second value is the content of each file.

(b) SPARK operation to create separate RDD elements:

// use flatMap()

activationRecords = activationFiles.flatMap(lambda (filename,xmlstring): getactivations(xmlstring))

(c) Extract and save to HDFS:

// extrac the account number and device model

models = activationRecords.map(lambda record: getaccount(record) + ":" + getmodel(record))

// Save to a file

models.saveAsTextFile("/loudacre/account-models")

Adrien Xie xie.z 451852
Guangda Ouyang 467428

Problem 5

(a) Pipelining: When possible, Spark will perform sequences of transformations by row so no data is stored.
Benefit: Intermediate records/RDDs do not have to be stored.

(b) An example of two operations that can be pipelined together:

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

(c) An example of two operations that cannot be pipelined together:

map() and groupByKey() because groupByKey() transformation usually needs to shuffle data over the network so map() and groupByKey() cannot be pipelined together.