Paul 6 - examples …

```
%pyspark                                                                    FINISHED

# Zeppelin notebook to extract host and in/out-link examples for each of the PLDs in th
# Complements summaries produced in 'Paul 5', and gets combined with these in 'Paul 7'.
# Recomended config for complete run: 3xr4.8xlarge, and set spark.driver.maxResultSize
# PJ - 30 October 2017

import boto
from pyspark.sql.types import *

# Load the saved files from Paul 5.
loadURI="s3://billsdata.net/CommonCrawl/hyperlinkgraph/cc-main-2017-may-jun-jul/domaing
pld_df_tmp=spark.read.load(loadURI)
pld_df=pld_df_tmp.select(pld_df_tmp.ID.cast("long"),pld_df_tmp.PLD) # Cast IDs from Str
pld_df.show(3)
pld_df.cache()
#print(pld_df.count()) # Should have 91M domains
```

```
+---+-------+
| ID|    PLD|
+---+-------+
|  0|  aaa.a|
|  1| aaa.aa|
|  2|aaa.aaa|
+---+-------+
only showing top 3 rows
DataFrame[ID: bigint, PLD: string]
```

```
%pyspark                                                                    FINISHED

# Next import the PLD edges as a DataFrame - i.e. in/out links
loadURI="s3://billsdata.net/CommonCrawl/hyperlinkgraph/cc-main-2017-may-jun-jul/domaing
pld_edges_df=spark.read.load(loadURI).limit(10000000).repartition(8) # TODO: Remove tem
pld_edges_df.show(3)
pld_edges_df.cache()
```

```
+---+--------+
|src|     dst|
+---+--------+
| 21|46356172|
| 27|      33|
| 27|      54|
+---+--------+
only showing top 3 rows
DataFrame[src: bigint, dst: bigint]
```

```
%pyspark

# Load the host-level graph vertices in the same way
saveURI="s3://billsdata.net/CommonCrawl/hyperlinkgraph/cc-main-2017-may-jun-jul/hostgra
host_df=spark.read.load(saveURI) #.repartition(64)
host_df.show(3)
host_df.cache()
#print(host_df.count()) # Should have 1.3B hosts
```

FINISHED

```
+------+-------+
|hostid|   host|
+------+-------+
|     0|  aaa.a|
|     1| aaa.aa|
|     2|aaa.aaa|
+------+-------+
only showing top 3 rows
DataFrame[hostid: string, host: string]
```

```
%pyspark

# Debug partitioning of our 3 big dataframes
print(pld_df.rdd.getNumPartitions())
print(pld_edges_df.rdd.getNumPartitions())
print(host_df.rdd.getNumPartitions())
```

FINISHED

```
128
8
128
```

```
%pyspark

# Create a dictionary of PLDs (for ID to PLD mapping of in/out links)
pld_dict=pld_df.rdd.collectAsMap()

# Distribute and test
pld_dict_distrib=sc.broadcast(pld_dict)
print(pld_dict_distrib.value[2]) # Should be aaa.aaa
```

FINISHED

```
aaa.aaa
```

```
%pyspark

# TODO: Save the map to disk for faster load next time
#pld_dict_distrib.dump(pld_dict_distrib.value, "s3://billsdata.net/CommonCrawl/domain_t
#help(pld_dict_distrib)
```

ERROR

```
Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-7601759374216390521.py", line 349, in <module>
    [code.body[-(nhooks + 1)]])
IndexError: list index out of range
```

```
%pyspark
```

FINISHED

```
# Function to lookup and unreverse PLDs
from pyspark.sql.functions import udf
def reverse_domain_from_ID(id):
    domain=pld_dict_distrib.value[id]
    return '.'.join(reversed(domain.split('.')))
print(reverse_domain_from_ID(2002))
udf_reverse_domain_from_ID = udf(reverse_domain_from_ID, StringType())

#  First, create a new edges dataframe consisting of unreversed PLDs
pld_edges_df2=pld_edges_df.withColumn("src2",udf_reverse_domain_from_ID("src")).drop("s
pld_edges_df.unpersist()
pld_edges_df2.show(5)
```

```
londonmet.ac
+--------+--------------------+
|    src2|                dst2|
+--------+--------------------+
|kxcr.net|    tsunamiwave.info|
|kxcr.net|         archive.org|
|kxcr.net|firstvoicesindige...|
|kxcr.net|            kpft.org|
|kxcr.net|         onbeing.org|
+--------+--------------------+
only showing top 5 rows
```

```
%pyspark
```

FINISHED

```
# Next use reduceByKey to aggregate and ensure no more than 10 per PLD - note we create
out_degree_examples=pld_edges_df2.rdd.map(lambda x:(x['src2'],[x['dst2']])).reduceByKey(
in_degree_examples=pld_edges_df2.rdd.map(lambda x:(x['dst2'],[x['src2']])).reduceByKey(

# Convert back to dataframes
out_schema = StructType([StructField('PLDout', StringType(), False),StructField('outLin
out_degree_examples_df=out_degree_examples.toDF(out_schema)
in_schema = StructType([StructField('PLDin', StringType(), False),StructField('inLinkPL
in_degree_examples_df=in_degree_examples.toDF(in_schema)

# TODO: Investigate slave lost and SparkContext shut down errors with LIMIT>=10M edges
# Note that the below also works but not sure how to restrict to only 10 IDs per PLD:
#from pyspark.sql.functions import collect_list
#out_degree_examples=pld_edges_df.groupBy("src").agg(collect_list("dst"))

pld_edges_df2.unpersist()
out_degree_examples_df.show(10)
```

```
|rafaelfelipesanto...|[instagram.com, c...|
+--------------------+--------------------+
only showing top 10 rows
+--------------------+--------------------+
|               PLDin|          inLinkPLDs|
+--------------------+--------------------+
|forensicsciencete...|   [blogspot.com.br]|
|  fernandooly.com.br|[blogspot.com.br,...|
|      sushi.training|    [foods.business]|
|         ixuepin.com|   [blogspot.com.br]|
|aalborgnetwork.co...|[agenciaondaforte...|
|         afmuseet.no|[aeol.com.br, han...|
|        ericcarr.com|[blogspot.com.br,...|
|     satellit.net.ua|[botecodoilgo.com...|
|       sil.sp.gov.br|[acsincor.com.br,...|
|   higherpages.co.uk|[linguagemclipper...|
+--------------------+--------------------+
only showing top 10 rows
```

```
%pyspark
```

FINISHED

```
# Join the In/Out-Link examples together
pld_df_joined=out_degree_examples_df.join(in_degree_examples_df, out_degree_examples_df
out_degree_examples_df.unpersist()
in_degree_examples_df.unpersist()
pld_df_joined.show(5)
pld_df_joined.cache()
pld_df_joined.count() # Should still be 91M
```

```
+----------------+--------------------+--------------------+
|     outLinkPLDs|               PLDin|          inLinkPLDs|
+----------------+--------------------+--------------------+
|            null|           022af.com|      [mynew.com.br]|
|[irradie.com.br]|100acoesparacapta...|[padrinhonota10.c...|
|            null|       100kursov.com|   [blogspot.com.by]|
|            null|         100news.net|     [fc-arsenal.by]|
|            null|100porcentojeansu...|     [websim.com.br]|
+----------------+--------------------+--------------------+
only showing top 5 rows
2799227
```

```
%pyspark
```

FINISHED

```
# Debugging
#help(collect_list("dst"))
#help(host_df.rdd.reduceByKey(lambda x,y: x+y))
print("Debug")
```

```
Debug
```

```
%pyspark
```
FINISHED

```
# Next, we'll construct a local dictionary from of all the PLDS (key is the PLD, value
# This is our truth-table of known PLDs that we'll use when extracting host examples

# Create a bloom filter using a pure python package (might be a little slow)
from pybloom import BloomFilter
pld_bf = BloomFilter(capacity=91000000, error_rate=0.005)

for row in pld_df.rdd.collect(): #.take(10000): # limit(10000000) # TODO: Still bad (an
    pld_bf.add(row['PLD'])

#print(pld_df.rdd.take(3))
#print(pld_df.rdd.take(3)[2]['PLD'])
print("aaa.aaa" in pld_bf) # Should be True

import sys
print(sys.getsizeof(pld_bf))
print(len(pld_bf)) # Should match number of items entered

# Broadcast the bloom filter so it's available on all the slave nodes - we don't need t
# it any more so it's fine being immutable.
pld_bf_distrib=sc.broadcast(pld_bf)

print("aaa.aaa" in pld_bf) # Should be true
print("aaa.aaa.bla" in pld_bf) # Should be false
print("aaa.aaa" in pld_bf_distrib.value) # Should be true
print("aaa.aaa.bla" in pld_bf_distrib.value) # Should be false
```

```
True
64
90751305
True
False
True
False
```

```
%pyspark
```
FINISHED

```
# Returns a Boolean to say whether PLD is a hostname in itself
def is_a_pld(hostname):
    #if hostname in pld_lookup_table:
    #if pld_lookup_table.filter(lambda a: a == hostname).count()>0:
    if hostname in pld_bf_distrib.value:
        return True
    else:
        return False

# Function to do the hostname->pld conversion, if the reversed pld exists in our dictio
def convert_hostname(hostname):
    # Return hostname as-is, if this is already a PLD
    #if hostname in pld_lookup_table:
    #if pld_lookup_table.filter(lambda a: a == hostname).count()>0:
    if hostname in pld_bf_distrib.value:
```

```
        return hostname
    # Otherwise we're going to have to split it up and test the parts
    try:
        parts=hostname.split('.')
        if (len(parts)>4 and is_a_pld('.'.join(parts[0:4]))):
            return '.'.join(parts[0:4])
        if (len(parts)>3 and is_a_pld('.'.join(parts[0:3]))):
            return '.'.join(parts[0:3])
        if (len(parts)>2 and is_a_pld('.'.join(parts[0:2]))):
            return '.'.join(parts[0:2])
        if (len(parts)>1 and is_a_pld('.'.join(parts[0:1]))):
            return '.'.join(parts[0:1])
        return "ERROR" # Couldn't find a corresponding PLD - this should never happen!
    except:
        return "ERROR"

# Test
print(convert_hostname("aaa.aaa"))
print(is_a_pld("aaa.aaa")) # Should be true
```

```
aaa.aaa
True
```

```
%pyspark                                                                    FINISHED

# Generate 10 host examples per PLD.

# Firstly, define a reverse domain function
def reverse_domain(domain):
    return '.'.join(reversed(domain.split('.')))
print(reverse_domain("com.facebook"))
#udf_reverse_domain = udf(reverse_domain, StringType())

# Now reverse all host names after conversion to PLDs (including lookup) but prior to s
#host_example_rdd=unrev_host_df.rdd.map(lambda x: (convert_hostname(x['host']),[x['host
host_example_rdd=host_df.rdd.map(lambda x: (reverse_domain(convert_hostname(x['host']))
print(host_example_rdd.take(20))

#print(host_example_rdd.count())
#host_df.unpersist()
```

```
facebook.com
[(u'savourea.be', [u'savourea.be']), (u'mywpm.com', [u'zdunex25.mywpm.com']), (u'autospo
rtevents.com', [u'autosportevents.com']), (u'alsident.co.uk', [u'alsident.co.uk']), (u'a
gent-fashion.com', [u'agent-fashion.com']), (u'thepsychologist.com.ua', [u'thepsychologi
st.com.ua']), (u'modnihouse.co.kr', [u'modnihouse.co.kr']), (u'monclerjacketssales2012.c
om', [u'monclerjacketssales2012.com']), (u'business-co.ru', [u'business-co.ru']), (u'diy
workouts.com', [u'diyworkouts.com']), (u'dovira.kiev.ua', [u'dovira.kiev.ua']), (u'coldi
lamodigiovannaneri.com', [u'coldilamodigiovannaneri.com']), (u'virtualcycles.com', [u'vi
rtualcycles.com']), (u'austinstarroofing.com', [u'austinstarroofing.com']), (u'labtechni
ka.sk', [u'labtechnika.sk']), (u'agapelive.co.za', [u'agapelive.co.za']), (u'mvin0smhny.
com', [u'mvin0smhny.com']), (u'automotivesalesconsultantsofamerica.com', [u'automotivesa
lesconsultantsofamerica.com']), (u'micrograde s.net', [u'micrograde s.net']), (u'blackhatt
ersguide.com', [u'blackhattersguide.com'])]
```

```
%pyspark

#print(host_example_rdd.take(100))

# Convert host examples back to a dataframe
out_schema = StructType([StructField('PLD', StringType(), False),StructField('hostExamp
host_examples_df=host_example_rdd.toDF(out_schema)
host_examples_df.show(100)
```

FINISHED

```
|         sextilm.com|       [sextilm.com]|
|         iran330.com|       [iran330.com]|
|bagombergandassoc...|[bagombergandasso...|
|  travelchinanow.com|[travelchinanow.com]|
| responsiveninja.com|[responsiveninja....|
|agence-immobilier...|[agence-immobilie...|
|g725t2mm9sly10i01...|[g725t2mm9sly10i0...|
|    gocreditshop.com|  [gocreditshop.com]|
|beekeepingsuperst...|[beekeepingsupers...|
|        rynoland.com|      [rynoland.com]|
|   guide-site-web.fr| [guide-site-web.fr]|
|    arenda-mebeli.su|  [arenda-mebeli.su]|
|          shsbwyg.cn|        [shsbwyg.cn]|
|    transdialogue.eu|  [transdialogue.eu]|
|             sktd.it|[sktd.it, files.s...|
| happytobenatural.nl|[happytobenatural...|
+--------------------+--------------------+
only showing top 100 rows
```

```
%pyspark

# Join in/out-link summaries with host examples dataframe
example_df=pld_df_joined.join(host_examples_df, pld_df_joined.PLDin==host_examples_df.PL
example_df.show(10)
example_df.cache()
example_df.count() # Should still be 91M!
```

FINISHED

```
+--------------------+--------------------+----------+-----------+
|                 PLD|        hostExamples|inLinkPLDs|outLinkPLDs|
+--------------------+--------------------+----------+-----------+
|0----------------...|[0---------------...|      null|       null|
|           0-0la.com|         [0-0la.com]|      null|       null|
|           0-3-6.com|         [0-3-6.com]|      null|       null|
|           0-3ani.ro|         [0-3ani.ro]|      null|       null|
|           0-5-1.com|         [0-5-1.com]|      null|       null|
|       0-60times.net|     [0-60times.net]|      null|       null|
|            0-744.cn|          [0-744.cn]|      null|       null|
|0-ads-free-web-pa...|[0-ads-free-web-p...|      null|       null|
|       0-artlove.net|     [0-artlove.net]|      null|       null|
|  0-clubpenguin-0.tk|[0-clubpenguin-0.tk]|      null|       null|
+--------------------+--------------------+----------+-----------+
only showing top 10 rows
90948651
```

```
%pyspark                                                                    FINISHED

# Save final table to S3 in parquet format, broken into smaller files (for fast reading
outputURI="s3://billsdata.net/CommonCrawl/domain_examples3/"
codec="org.apache.hadoop.io.compress.GzipCodec"
example_df.coalesce(1).write.format('com.databricks.spark.csv').options(header='true', c
#example_df.write.save(outputURI)
```

```
%pyspark                                                                    FINISHED
```