Spark

May 22, 2018

1 Starting point for Spark on Google Cloud

In [1]:
```
from pyspark.sql.types import *
import pyspark.sql.functions as F
```

1.1 Step 2.3

In [2]:
```
# TODO: read files, load graph_sdf, etc.
answers_sdf = spark.read.format("com.databricks.spark.csv").option("delimiter", ' ').loa
answers_sdf=answers_sdf.select(answers_sdf['_c0'].alias('from_node').cast("integer"),ans

comments_answers_sdf = spark.read.format("com.databricks.spark.csv").option("delimiter",
comments_answers_sdf = comments_answers_sdf.select(comments_answers_sdf['_c0'].alias('fr

comments_questions_sdf = spark.read.format("com.databricks.spark.csv").option("delimiter
comments_questions_sdf = comments_questions_sdf.select(comments_questions_sdf['_c0'].ali
```

In [3]:
```
# Add as many cells as you like
graph_sdf=answers_sdf.unionAll(comments_questions_sdf).unionAll(comments_answers_sdf)
#graph_sdf=answers_sdf
graph_sdf=graph_sdf.dropDuplicates()

graph_sdf.show()
```

```
+---------+-------+
|from_node|to_node|
+---------+-------+
|   892256|1527217|
|   620444|2557834|
|   788770|2038761|
|   439667| 863502|
|  1889925|1889925|
|   415448|1177969|
|  1610271|2558200|
|   131226|1388484|
|  1338158|2472144|
|  2767755| 557527|
```

```
|   390913|2402616|
|  1458983|2558703|
|   181965|2554489|
|  2498746|1149981|
|  1831602|1382653|
|  1265817|1659854|
|   841632|2521606|
|  2538402|2518523|
|  1291499| 387194|
|   689579|1379347|
+---------+-------+
only showing top 20 rows
```

```
In [4]: def sdf_is_empty(sdf):

            try:

                sdf.take(1)

                return False

            except:

                return True
```

```
In [5]:
        def transitive_closure(G, origins, max_depth):
            ##Your logic goes here
            frontier = origins
            visited = frontier
            for i in range(max_depth):
                if i == 0:
                    return_sdf = frontier
                    return_sdf = return_sdf.withColumn('new_col', F.lit(i)).alias('depth')
                else:
                    d1 = frontier.alias('f').join(G.alias('g'), F.col('f.node') == F.col('g.from
                    d2 = d1.select(d1['to_node'].alias("node"))
                    visited = visited.unionAll(frontier)
                    frontier = d2
                    frontier = frontier.join(visited , frontier.node == visited.node , 'leftanti
                    G = G.join(visited, G.to_node == visited.node, 'leftanti')
                    temp_df = frontier
                    temp_df = temp_df.withColumn('new_col', F.lit(i)).alias('depth')
                    return_sdf = return_sdf.unionAll(temp_df)
                    return_sdf=return_sdf.dropDuplicates()

            return return_sdf
```

```
In [6]: # Compute nodes_sdf
        nodes_sdf=graph_sdf[(graph_sdf.from_node) <8]
        nodes_sdf=nodes_sdf.select(nodes_sdf['from_node'].alias('node'))
        nodes_sdf=nodes_sdf.dropDuplicates()

In [7]: reachable_sdf = transitive_closure(graph_sdf, nodes_sdf, 3)
```

1.2 Step 2.3 Results

```
In [8]: reachable_sdf.count()

Out[8]: 677359

In [9]: reachable_sdf.show()

+------+-------+
|  node|new_col|
+------+-------+
|     1|      0|
|     3|      0|
|     5|      0|
|     4|      0|
|     2|      0|
| 17389|      1|
|179115|      1|
|408870|      1|
|   392|      1|
|  4219|      1|
| 30183|      1|
| 36706|      1|
| 42348|      1|
|  3488|      1|
|269578|      1|
|738811|      1|
| 17712|      1|
| 33690|      1|
| 42754|      1|
|113570|      1|
+------+-------+
only showing top 20 rows
```