

L8: Spark GraphX 1

IT5427
Tích hợp và xử lý dữ liệu lớn

IT542701/2023
Thanh-Chung Dao Ph.D.

1

GraphX

- Apache Spark's API for graphs and graph-parallel computation
- GraphX unifies ETL (Extract, Transform & Load) process
- Exploratory analysis and iterative graph computation within a single system

2

Use cases

- Facebook's friends, LinkedIn's connections
- Internet's routers
- Relationships between galaxies and stars in astrophysics and Google's Maps
- Disaster detection, banking, stock market

3

RDD on GraphX

- GraphX extends the Spark RDD with a Resilient Distributed Property Graph
- The property graph is a directed multigraph which can have multiple edges in parallel
- The parallel edges allow multiple relationships between the same vertices

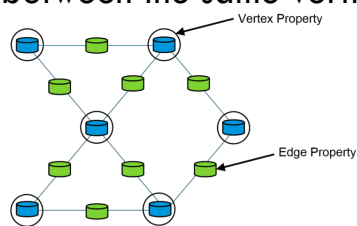


Figure: Property Graph

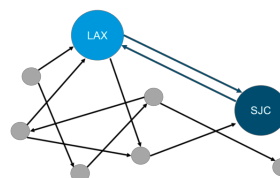


Figure: An example of property graph

4

Spark GraphX Features

Flexibility

- Spark GraphX works with both graphs and computations
- GraphX unifies ETL (Extract, Transform & Load), exploratory analysis and iterative graph computation

Speed

- The fastest specialized graph processing systems

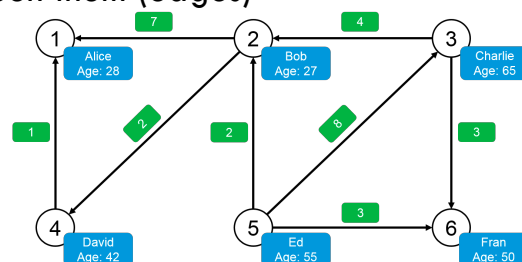
Growing Algorithm Library

- Page rank, connected components, label propagation, SVD++, strongly connected components and triangle count

5

GraphX with Examples

- The graph here represents the Twitter users and whom they follow on Twitter. For e.g. Bob follows Davide and Alice on Twitter
- Looking at the graph, we can extract information about the people (vertices) and the relations between them (edges)



6

Source code

```

1 //Importing the necessary classes
2 import org.apache.spark._
3 import org.apache.spark.rdd.RDD
4 import org.apache.spark.util.IntParam
5 import org.apache.spark.graphx._
6 import org.apache.spark.graphx.util.GraphGenerators

```

Displaying Vertices: Further, we will now display all the names and ages of the users (vertices).

```

1 val vertexRDD: RDD[(Long, (String, Int))] = sc.parallelize(vertexArray)
2 val edgeRDD: RDD[Edge[Int]] = sc.parallelize(edgeArray)
3 val graph: Graph[(String, Int), Int] = Graph(vertexRDD, edgeRDD)
4 graph.vertices.filter { case (id, (name, age)) => age > 30 }
   .collect.foreach { case (id, (name, age)) => println(s"$name is $age") }

```

The output for the above code is as below:

```

David is 42
Fran is 50
Ed is 55
Charlie is 65

```

7

More source code

Displaying Edges: Let us look at which person likes whom on Twitter.

```

1 for (triplet <- graph.triplets.collect)
2 {
3   println(s"${triplet.srcAttr._1} likes ${triplet.dstAttr._1}")
4 }

```

The output for the above code is as below:

```

Bob likes Alice
Bob likes David
Charlie likes Bob
Charlie likes Fran
David likes Alice
Ed likes Bob
Ed likes Charlie
Ed likes Fran

```

8

Other example in PySpark

```

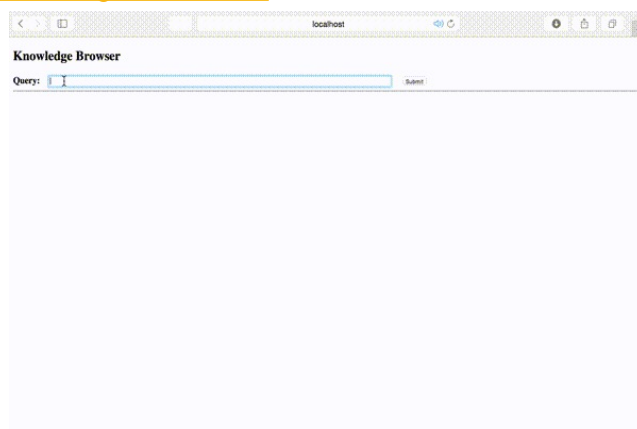
2  ## pyspark --packages graphframes:graphframes:0.6.0-spark2.2-s_2.11
3  from graphframes import *
4  from pyspark import *
5  from pyspark.sql import *
6  spark = SparkSession.builder.appName('fun').getOrCreate()
7  vertices = spark.createDataFrame([('1', 'Carter', 'Derrick', 50),
8                                   ('2', 'May', 'Derrick', 26),
9                                   ('3', 'Mills', 'Jeff', 80),
10                                  ('4', 'Hood', 'Robert', 65),
11                                  ('5', 'Banks', 'Mike', 93),
12                                  ('98', 'Berg', 'Tim', 28),
13                                  ('99', 'Page', 'Allan', 16)],
14                                  ['id', 'name', 'firstname', 'age'])
15  edges = spark.createDataFrame([('1', '2', 'friend'),
16                                  ('2', '1', 'friend'),
17                                  ('3', '1', 'friend'),
18                                  ('1', '3', 'friend'),
19                                  ('2', '3', 'follows'),
20                                  ('3', '4', 'friend'),
21                                  ('4', '3', 'friend'),
22                                  ('5', '3', 'friend'),
23                                  ('3', '5', 'friend'),
24                                  ('4', '5', 'follows'),
25                                  ('98', '99', 'friend'),
26                                  ('99', '98', 'friend')],
27                                  ['src', 'dst', 'type'])
28  g = GraphFrame(vertices, edges)
29  ## Take a look at the DataFrames
30  g.vertices.show()
31  g.edges.show()
32  ## Check the number of edges of each vertex
33  g.degrees.show()

```

9

Spark Knowledge Graph

- Example: <https://github.com/spoddutur/graph-knowledge-browser>



10

Acknowledgement and References

Books:

Slides:

- <https://www.edureka.co/blog/spark-graphx/>