



Apache Spark + GraphFrames + GraphX

FATEMA NAGORI 19635



TABLE OF CONTENT

Introduction

Design

Implementation

Test

Enhancement Ideas

Conclusion

References

INTRODUCTION



GraphFrame mainly provides the following built-in algorithms:

Triangle count

PageRank

Shortest Path

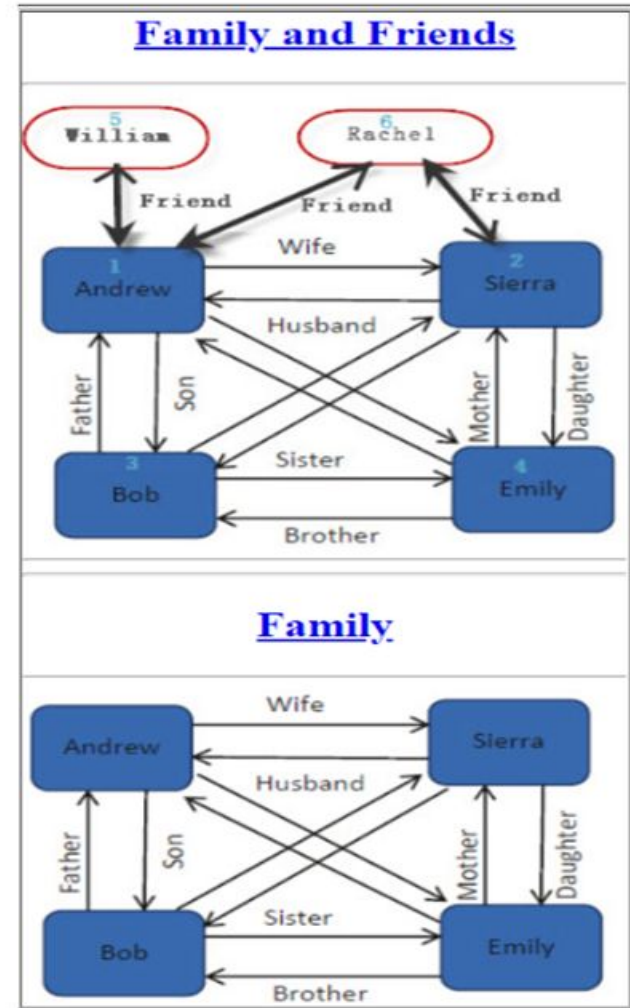
GraphFrames vs. GraphX

	GraphFrames	GraphX
Core APIs	Scala, Java, Python	Scala, Java, Python
Programming Abstraction	DataFrames	RDDs
Use Cases	Algorithms, Queries, Motif Finding	Algorithms
Vertex/edge attributes	Any number of DataFrame columns	Any type
Return Types	GraphFrames/DataFrames	Graph

DESIGN

Family and Friend information:

- Here **source** and **destination** are **user ids** to **relationship column** show the **relationship** between them.



IMPLEMENTATION

1. Create GCP project and compute engine vm instance

The image shows a Google Cloud Console interface for a project named 'New CS570'. The 'Compute Engine' section is active, displaying 'VM instances'. A table lists one instance named 'instance-1' in the 'us-central1-a' zone, with a status of 'Running' (indicated by a green checkmark). The 'Connect' column shows an 'SSH' button. To the right, an 'SSH-in-browser' terminal window is open, displaying the Ubuntu 20.04.5 LTS login banner and system information. The terminal output includes system load, memory usage, and a notification for a new Ubuntu release '22.04.1 LTS' available for upgrade.

Google Cloud | New CS570 | Search | Products, resources, docs (/)

Compute Engine | VM instances | CREATE INSTANCE | OPERATION

Virtual machines

- VM instances
- Instance templates
- Sole-tenant nodes
- Machine images
- TPUs
- Committed use discounts

INSTANCES | INSTANCE SCHEDULES

VM instances are highly configurable virtual machines for running workloads on Google infrastructure. [Learn more](#)

Filter Enter property name or value

Status	Name	Zone	Rev	Connect
Running	instance-1	us-central1-a		SSH

SSH-in-browser | UPLOAD FILE | DOWNLOAD FILE

Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-1025-gcp x86_64)

- * Documentation: <https://help.ubuntu.com>
- * Management: <https://landscape.canonical.com>
- * Support: <https://ubuntu.com/advantage>

System information as of Fri Dec 16 01:41:14 UTC 2022

System load: 0.0		Processes: 103
Usage of /:	37.9% of 9.51GB	Users logged in: 1
Memory usage: 7%		IPv4 address for ens4: 10.128.0.1
Swap usage: 0%		

0 updates can be applied immediately.

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Dec 16 00:11:37 2022 from 35.235.244.33
fnagori@instance-1:~\$

2.Install pyspark and java 11



```
fnagori@instance-1:~$ wget https://archive.apache.org/dist/spark/spark-3.1.3/spark-3.1.3-bin-hadoop2.7.tgz
--2022-12-16 00:14:39-- https://archive.apache.org/dist/spark/spark-3.1.3/spark-3.1.3-bin-hadoop2.7.tgz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 227452039 (217M) [application/x-gzip]
Saving to: 'spark-3.1.3-bin-hadoop2.7.tgz'

spark-3.1.3-bin-ha 100%[=====>] 216.92M  16.9MB/s   in 14s

2022-12-16 00:14:54 (15.5 MB/s) - 'spark-3.1.3-bin-hadoop2.7.tgz' saved [227452039/227452039]

fnagori@instance-1:~$ tar -xvf spark-3.1.3-bin-hadoop2.7.tgz
spark-3.1.3-bin-hadoop2.7/
spark-3.1.3-bin-hadoop2.7/bin/
spark-3.1.3-bin-hadoop2.7/bin/pyspark.cmd
spark-3.1.3-bin-hadoop2.7/bin/spark-submit
spark-3.1.3-bin-hadoop2.7/bin/spark-submit.cmd
spark-3.1.3-bin-hadoop2.7/bin/spark-class2.cmd
spark-3.1.3-bin-hadoop2.7/bin/spark-shell2.cmd
spark-3.1.3-bin-hadoop2.7/bin/pyspark2.cmd
spark-3.1.3-bin-hadoop2.7/bin/docker-image-tool.sh
```

- 
3. SET THE ENVIRONMENT VARIABLE IN .bashrc
 4. source .bashrc

```
elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
fi
export SPARK_HOME=/home/fnagori/spark
export PATH=$SPARK_HOME/bin:$PATH
export PATH=$SPARK_HOME/sbin:$PATH
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
~
~
~
```

5. Verify the pyspark

Copyright © 2004 John Wiley & Sons, Ltd.

```
fnagori@instance-1:~$ source .bashrc
```

```
fnagori@instance-1:~$ pyspark
```

```
Python 3.8.10 (default, Nov 14 2022, 12:59:47)
```

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

WARNING: An illegal reflective access operation has occurred

```
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/home/fnagori/spark-3.1.3-bin-hadoop2.7/jars/spark-unsafe_2.12-3.1.3.jar) to const
```

```

ructor java.nio.DirectByteBuffer(long, int)

```

WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform

WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations

WARNING: All illegal access operations will be denied in a future release

```
22/12/16 00:43:54 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
```

```
Setting default log level to "WARN"
```

To adjust logging level use `sc.setLogLevel(newLevel)`. For SparkR, use `setLogLevel(newLevel)`.

Welcome to



version 3.1.3

Using Python version 3.8.10 (default, Nov 14 2022 12:59:47)

Spark context Web UI available at <http://instance-1.us-central1-a.c.new-cs570.internal:4040>

Spark context available as 'sc' (master = local[*], app id = local-1671151437630).

```
SparkSession available as 'spark'.
```

>>>

- Prepare input data files

```
$ mkdir in
```

```
$ cd in
```

```
$ vi person.csv
```

```
$ vi relationship.csv
```

```
$ cat person.csv
```

```
$ cat relationship.csv
```

person.csv			relation.csv		
id	Name	Age	src	dst	relation
1	Andrew	45	1	2	Husband
2	Sierra	43	1	3	Father
3	Bob	12	1	4	Father
4	Emily	10	1	5	Friend
5	William	35	1	6	Friend
6	Rachel	32	2	1	Wife
			2	3	Mother
			2	4	Mother
			2	6	Friend
			3	1	Son
			3	2	Son
			4	1	Daughter
			4	2	Daughter
			5	1	Friend
			6	1	Friend
			6	2	Friend

```
fnagori@instance-1:~$ mkdir in
fnagori@instance-1:~$ cd in
fnagori@instance-1:~/in$ vi person.csv
fnagori@instance-1:~/in$ vi relationship.csv
fnagori@instance-1:~/in$ vi pyspark_graphX.py
fnagori@instance-1:~/in$
```

6. Prepare script file- pyspark-graphX.py

```
# Import PySpark
import pyspark
from pyspark.sql import SparkSession

#Create SparkSession
spark =
SparkSession.builder.master("local[1]").appName("pysparkGraphX").getOrCreate()

from graphframes import *

# Recipe 9-1. Create GraphFrames
# person dataframe : id, Name, age
personsDf = spark.read.csv('in/person.csv',header=True, inferSchema=True)

# Create a "persons" SQL table from personsDf DataFrame
personsDf.createOrReplaceTempView("persons")
spark.sql("select * from persons").show()

# relationship dataframe : src, dst, relation
relationshipDf = spark.read.csv('in/relationship.csv',header=True, inferSchema=True)
relationshipDf.createOrReplaceTempView("relationship")
spark.sql("select * from relationship").show()

# - Create a GraphFrame from both person and relationship dataframes
# >>> graph
# GraphFrame(v:[id: int, Name: string ... 1 more field], e:[src:
# int, dst: int ... 1 more field])
# - A GraphFrame that contains v and e.
# + The v represents vertices and e represents edges.
graph = GraphFrame(personsDf, relationshipDf)

# - Degrees represent the number of edges that are connected to a vertex.
# + GraphFrame supports inDegrees and outDegrees.
# - inDegrees give you the number of incoming links to a vertex.
# - outDegrees give the number of outgoing edges from a node.
```

```
# - Find all the edges connected to Andrew.
graph.degrees.filter("id = 1").show()

# Find the number of incoming links to Andrew
graph.inDegrees.filter("id = 1").show()

# Find the number of links coming out from Andrew using the outDegrees
graph.outDegrees.filter("id = 1").show()

# Recipe 9-2. Apply Triangle Counting in a GraphFrame
# - Find how many triangle relationships the vertex is participating in
personsTriangleCountDf = graph.triangleCount()
personsTriangleCountDf.show()

# Create a "personsTriangleCount" SQL table from the
# personsTriangleCountDf DataFrame
personsTriangleCountDf.createOrReplaceTempView("personsTriangleCount")

# Create a "personsMaxTriangleCount" SQL table from the
# maxCountDf DataFrame
maxCountDf = spark.sql("select max(count) as max_count from personsTriangleCount")
maxCountDf.createOrReplaceTempView("personsMaxTriangleCount")

spark.sql("select * from personsTriangleCount P JOIN (select * from
personsMaxTriangleCount) M ON (M.max_count = P.count)").show()

# Recipe 9-3. Apply a PageRank Algorithm
pageRank = graph.pageRank(resetProbability=0.20, maxIter=10)
pageRank.vertices.printSchema()

pageRank.vertices.orderBy("pagerank",ascending=False).show()

pageRank.edges.orderBy("weight",ascending=False).show()

# Recipe 9-4. Apply the Breadth First Algorithm
graph.bfs(fromExpr = "Name='Bob'",toExpr = "Name='William'").show()

graph.bfs(fromExpr = "age < 20", toExpr = "name = 'Rachel'").show()
graph.bfs(fromExpr = "age < 20", toExpr = "name = 'Rachel'", edgeFilter = "relation !=
'Son'").show()
```



Pip3 install numpy

```
fnagori@instance-1:~/in$ pip3 install numpy
Collecting numpy
  Downloading numpy-1.23.5-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.1 MB)
    |████████████████████| 17.1 MB 4.8 MB/s
Installing collected packages: numpy
  WARNING: The scripts f2py, f2py3 and f2py3.8 are installed in '/home/fnagori/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-1.23.5
fnagori@instance-1:~/in$
```

5. Submit the job

```
$ spark-submit --packages graphframes:graphframes:0.8.2-spark3.1-s_2.12 pyspark_graphX.py
```

Note: graphframes versions available at: <https://spark-packages.org/package/graphframes/graphframes>

```
fnagori@instance-1:~$ spark-submit --packages graphframes:graphframes:0.8.2-spark3.1-s_2.12 pyspark_graphX.py
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/home/fnagori/spark-3.1.3-bin-hadoop2.7/jars/spark-unsafe_2.12-3.1.3.jar) to const
ructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
:: loading settings :: url = jar:file:/home/fnagori/spark-3.1.3-bin-hadoop2.7/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
Ivy Default Cache set to: /home/fnagori/.ivy2/cache
The jars for the packages stored in: /home/fnagori/.ivy2/jars
graphframes#graphframes added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-460dfc75-7f5c-4919-83f8-e4813f157c2c;1.0
  confs: [default]
  found graphframes#graphframes;0.8.2-spark3.1-s_2.12 in spark-packages
  found org.slf4j#slf4j-api;1.7.16 in central
:: resolution report :: resolve 330ms :: artifacts dl 8ms
  :: modules in use:
  graphframes#graphframes;0.8.2-spark3.1-s_2.12 from spark-packages in [default]
  org.slf4j#slf4j-api;1.7.16 from central in [default]
-----
|               |          modules          || artifacts |
|   conf        | number| search|dwnlded|evicted|| number|dwnlded|
-----+-----+-----+-----+-----+-----+-----+-----+
| default       |     2 |     0 |     0 |     0 ||     2 |     0 |
-----+-----+-----+-----+-----+-----+
:: retrieving :: org.apache.spark#spark-submit-parent-460dfc75-7f5c-4919-83f8-e4813f157c2c
  confs: [default]
  0 artifacts copied, 2 already retrieved (0kB/8ms)
22/12/16 01:15:55 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
22/12/16 01:15:56 INFO SparkContext: Running Spark version 3.1.3
22/12/16 01:15:56 INFO ResourceUtils: =====
22/12/16 01:15:56 INFO ResourceUtils: No custom resources configured for spark.driver.
22/12/16 01:15:56 INFO ResourceUtils: =====
22/12/16 01:15:56 INFO SparkContext: Submitted application: pysparkGraphX
22/12/16 01:15:56 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory ->
name: memory, amount: 1024, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
```

Result:

GraphFrame:

id	Name	Age
1	Andrew	45
2	Sierra	43
3	Bob	12
4	Emily	10
5	William	35
6	Rachel	32

src	dst	relation
1	2	Husband
1	3	Father
1	4	Father
1	5	Friend
1	6	Friend
2	1	Wife
2	3	Mother
2	4	Mother
2	6	Friend
3	1	Son
3	2	Son
4	1	Daughter
4	2	Daughter
5	1	Friend
6	1	Friend
6	2	Friend

TriangleCount:

id	degree
1	10

id	inDegree
1	5

id	outDegree
1	5

count	id	Name	Age
3	1	Andrew	45
1	6	Rachel	32
1	3	Bob	12
0	5	William	35
1	4	Emily	10
3	2	Sierra	43

count	id	Name	Age	max_count
3	1	Andrew	45	3
3	2	Sierra	43	3

PageRank:

```
root
|-- id: integer (nullable = true)
|-- Name: string (nullable = true)
|-- Age: integer (nullable = true)
|-- pagerank: double (nullable = true)
```

id	Name	Age	pagerank
1	Andrew	45	1.787923121897472
2	Sierra	43	1.406016795082752
6	Rachel	32	0.7723665979473922
4	Emily	10	0.7723665979473922
3	Bob	12	0.7723665979473922
5	William	35	0.4889602891776001

src	dst	relation	weight
5	1	Friend	1.0
3	1	Son	0.5
4	1	Daughter	0.5
4	2	Daughter	0.5
6	1	Friend	0.5
3	2	Son	0.5
6	2	Friend	0.5
2	3	Mother	0.25
2	4	Mother	0.25
2	1	Wife	0.25
2	6	Friend	0.25
1	2	Husband	0.2
1	6	Friend	0.2
1	3	Father	0.2
1	4	Father	0.2
1	5	Friend	0.2

BFS:

```
+-----+-----+-----+-----+
|      from|      e0|      v1|      e1|      to|
+-----+-----+-----+-----+
|{3, Bob, 12}|{3, 1, Son}|{1, Andrew, 45}|{1, 5, Friend}|{5, William, 35}|
+-----+-----+-----+-----+

+-----+-----+-----+-----+
|      from|      e0|      v1|      e1|      to|
+-----+-----+-----+-----+
|{4, Emily, 10}|{4, 1, Daughter}|{1, Andrew, 45}|{1, 6, Friend}|{6, Rachel, 32}|
|{3, Bob, 12}|{3, 1, Son}|{1, Andrew, 45}|{1, 6, Friend}|{6, Rachel, 32}|
|{4, Emily, 10}|{4, 2, Daughter}|{2, Sierra, 43}|{2, 6, Friend}|{6, Rachel, 32}|
|{3, Bob, 12}|{3, 2, Son}|{2, Sierra, 43}|{2, 6, Friend}|{6, Rachel, 32}|
+-----+-----+-----+-----+

+-----+-----+-----+-----+
|      from|      e0|      v1|      e1|      to|
+-----+-----+-----+-----+
|{4, Emily, 10}|{4, 1, Daughter}|{1, Andrew, 45}|{1, 6, Friend}|{6, Rachel, 32}|
|{4, Emily, 10}|{4, 2, Daughter}|{2, Sierra, 43}|{2, 6, Friend}|{6, Rachel, 32}|
+-----+-----+-----+-----+
```

DELETE THE INSTANCE:

The screenshot shows the Google Cloud console interface. On the left, the 'Compute Engine' sidebar is visible with 'VM instances' selected. The main area displays 'VM instances' with a list of instances. A modal dialog titled 'Delete instance-1?' is centered on the screen, asking for confirmation to delete the instance and its boot disk. The dialog has 'CANCEL' and 'DELETE' buttons. In the background, the 'instance-1' details page is visible, showing tabs for 'PERMISSIONS', 'LABELS', and 'MONITORING'. The 'PERMISSIONS' tab is active, showing a list of principals with their inheritance status.

Google Cloud New CS570 Search Products, resources, docs (/)

Compute Engine VM instances CREATE INSTANCE OPERATIONS HELP ASSISTANT HIDE INFO PANEL LEARN

Virtual machines VM instances Instance templates Sole-tenant nodes Machine images TPUs Committed use discounts Migrate to Virtual Machin...

INSTANCES INSTANCE SCHEDULES

VM instances are highly configurable virtual machines for running workloads on Google in

Delete instance-1?

Are you sure you want to delete instance "instance-1"? (This will also delete boot disk "instance-1")

CANCEL DELETE

instance-1 PERMISSIONS LABELS MONITORING

delete permissions below, or "instance-1" to grant new ADD PRINCIPAL

show inherited permissions

Enter property name or value

principal ↑ Inheritance

Compute Engine Service Agent (1)

Dataprof Service Agent (1)

Editor (2)

Kubernetes Engine Service Agent (1)

Owner (2)

Related actions

Explore Backup and DR NEW Back up your VMs and set up disaster recovery

Monitor VMs View outlier VMs across metrics like CPU



Conclusion:

The Spark GraphFrame is a powerful abstraction for processing large graphs using distributed computing. It provides a plethora of common graph algorithms including label propagation and PageRank. Further, it provides the foundations for implementing complex graph algorithms, including a robust implementation of the Pregel paradigm for graph processing.



REFERENCE:

<https://spark-packages.org/package/graphframes/graphframes>

<https://towardsdatascience.com/graphframes-in-jupyter-a-practical-guide-9b3b346cebc5#:~:text=The%20functionality%20of%20GraphFrames%20and,browsing%20through%20the%20API%20documentation.>

https://hc.labnet.sfbu.edu/~henry/sfbu/course/pyspark_sql_recipes/graphframes/slide/graphx.html