Table of contents

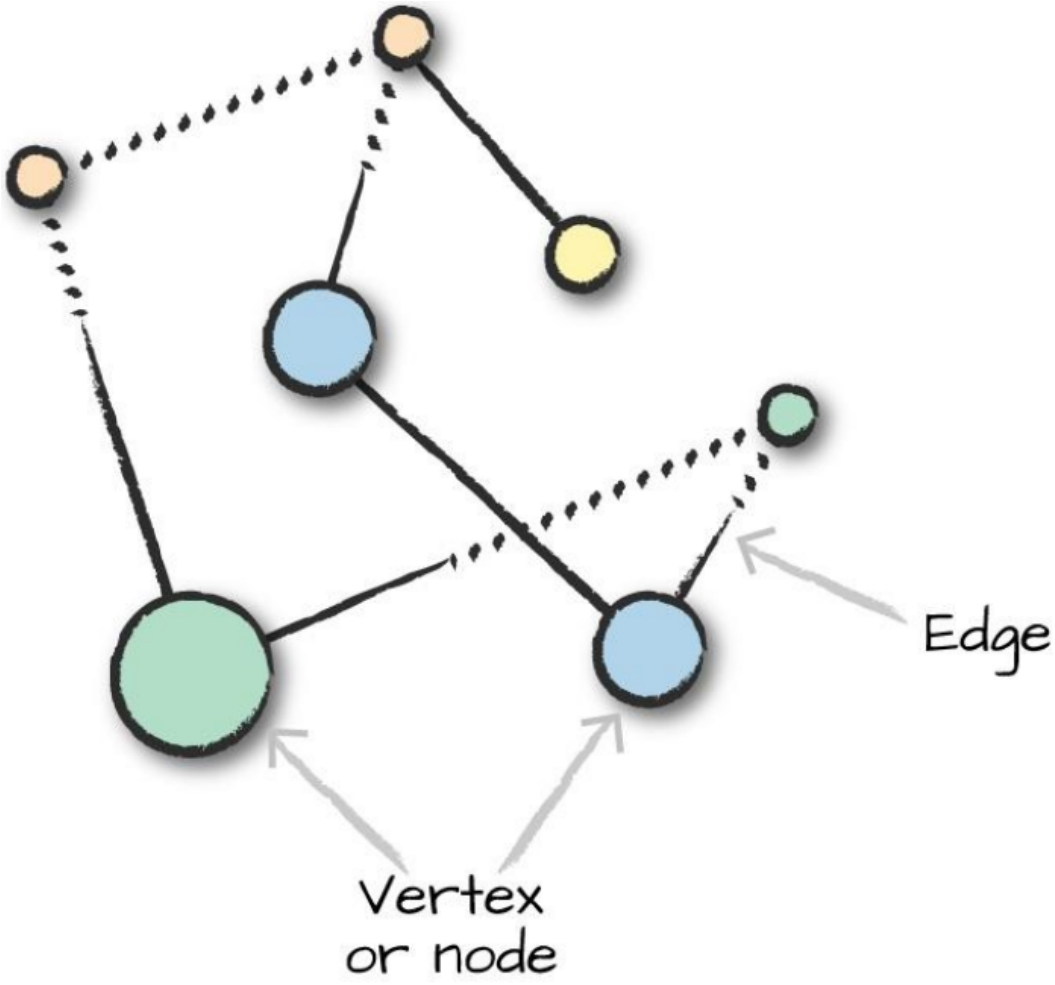
1 Introduction

Graphs are data structures composed of nodes and edges

- nodes/vertexes are denoted as $V = \{v_1, v_2, ..., v_n\}$
- edges are denoted as $E = \{e_1, e_2, ..., e_n\}$

Graph analytics is the process of analyzing relationships between vertexes and edges.

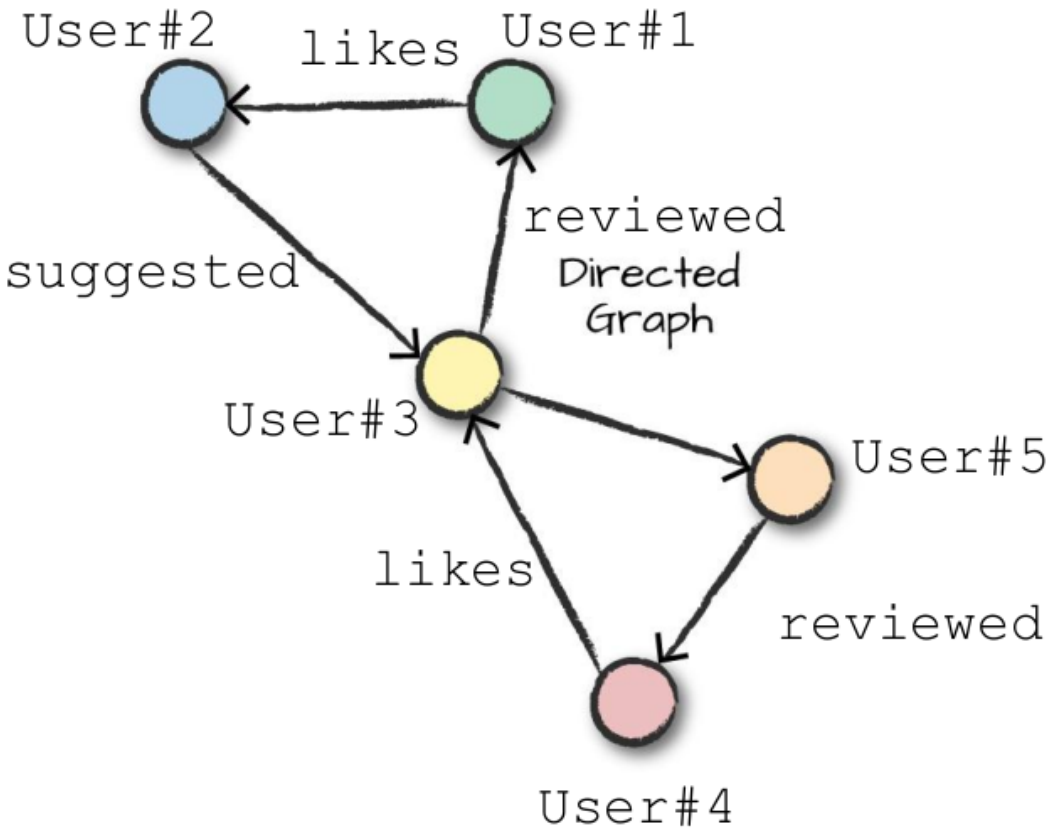
Figure 1: Example of graph



Graphs are called **undirected** if edges do not have a direction, otherwise they are called **directed** graphs. Vertexes and edges can have data associated with them

- weights are associated to edges (e.g., they may represent the strength of the relationship);
- labels are associated to vertexes (e.g., they may be the string associated with the name of the vertex).

Figure 2: Graph with labels and weights

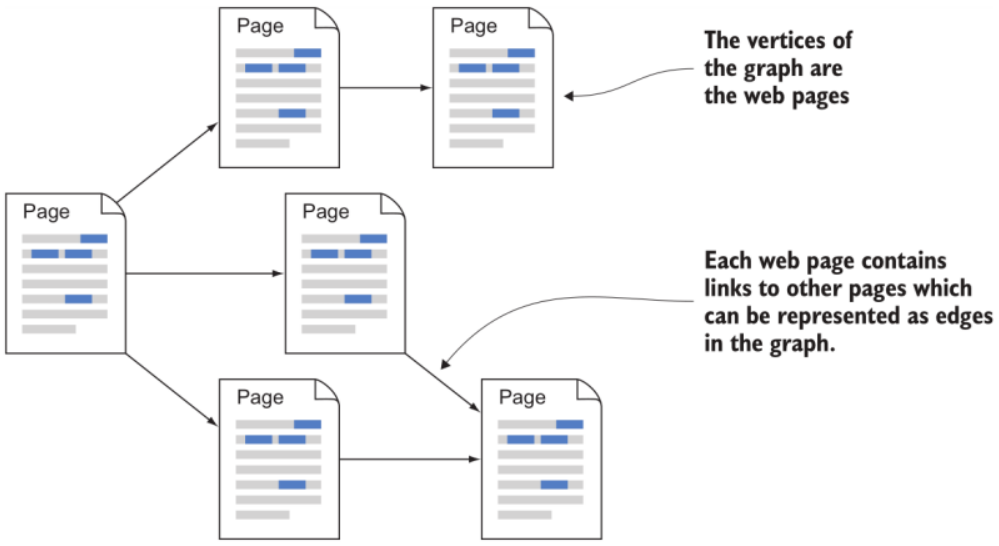


Why graph analytics?

Graphs are natural way of describing relationships. Some practical example of analytics over graphs

- Ranking web pages (Google PageRank)

Figure 3: Pages in the web



- Detecting group of friends

Figure 4: Social networks

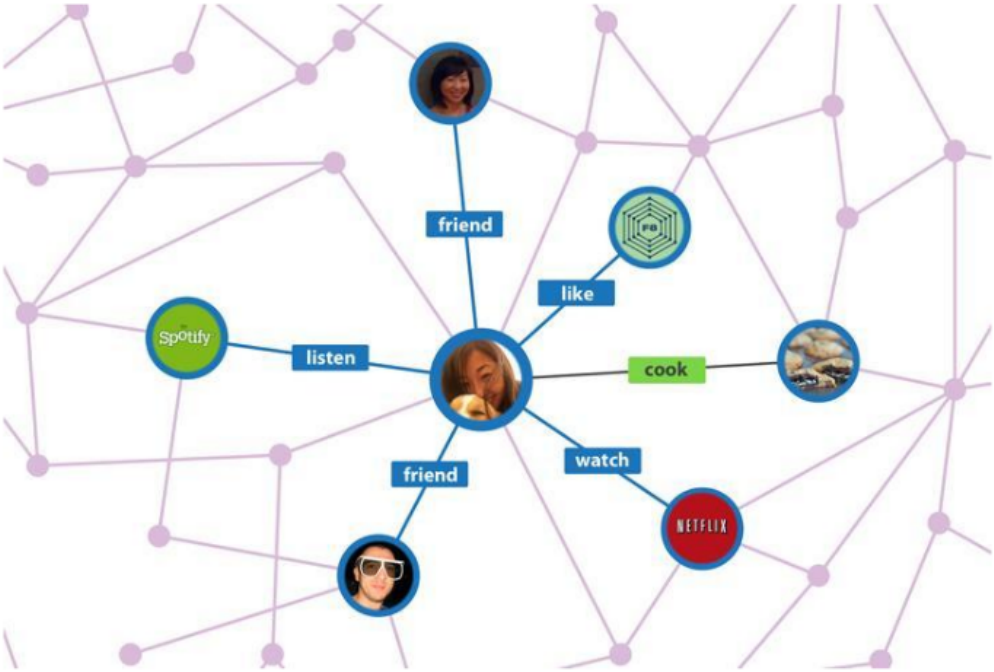
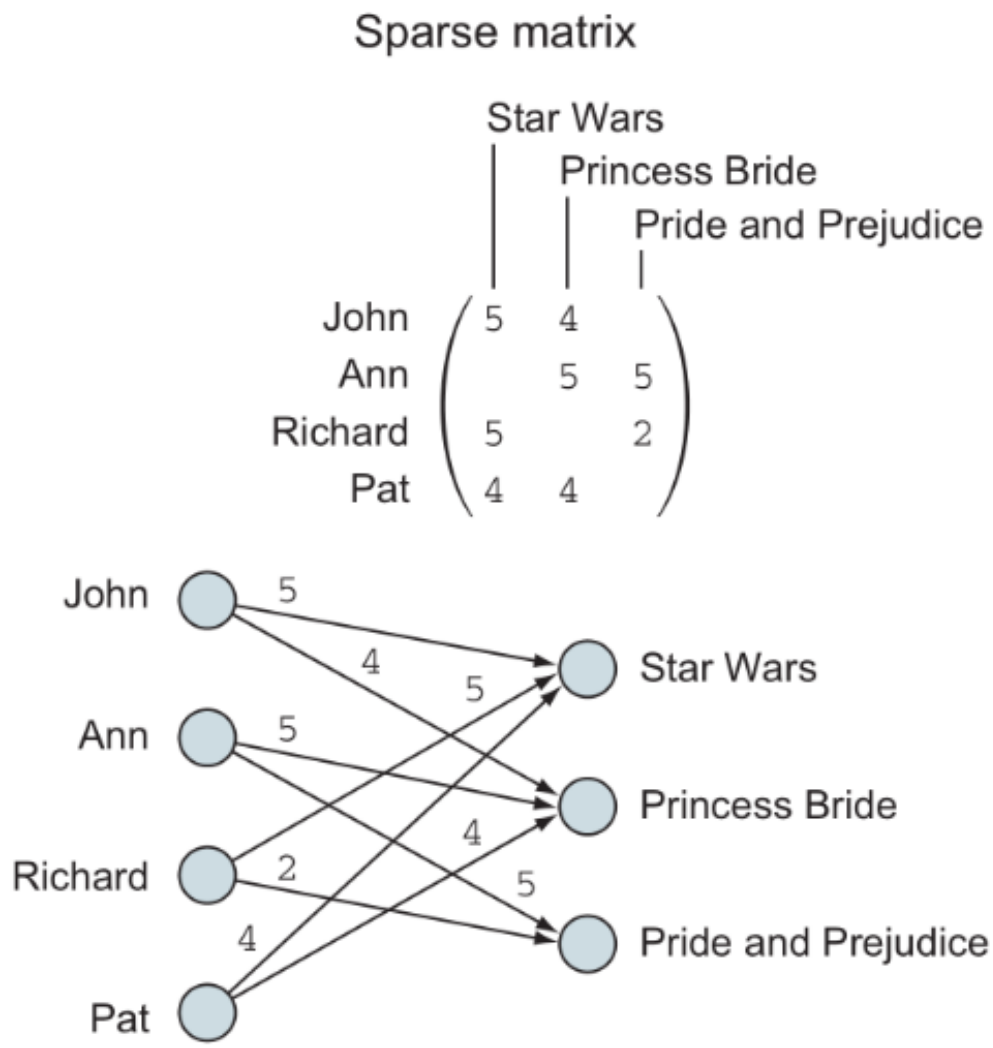


Figure 5: Movies watched by users

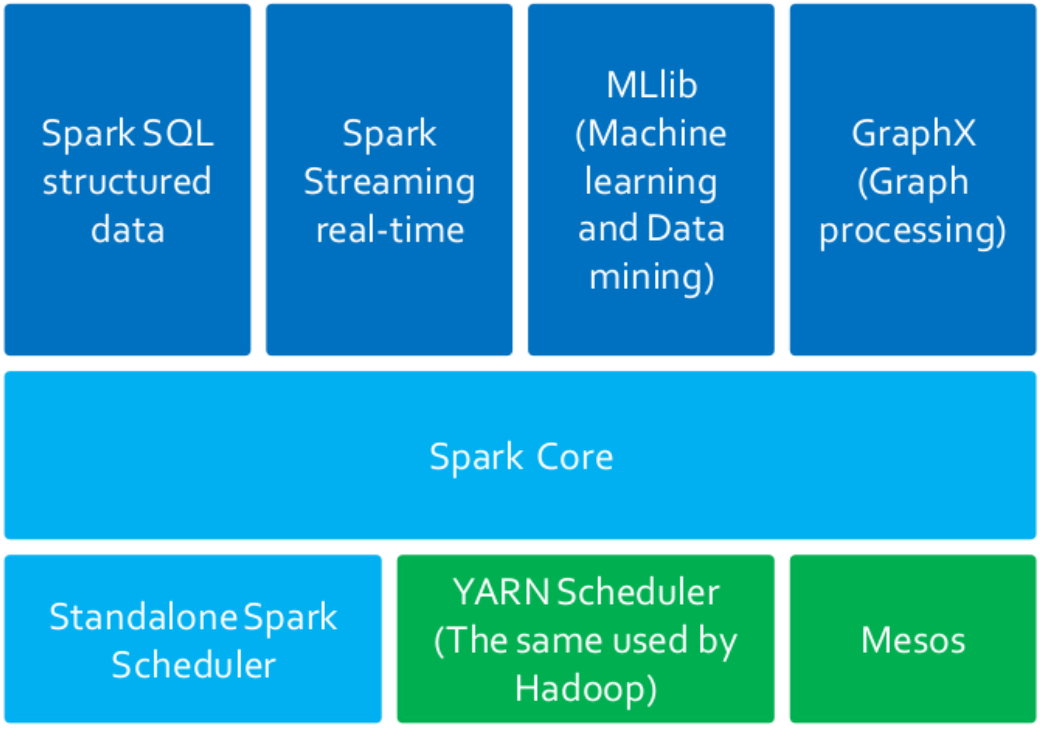


- Determine importance of infrastructure in electrical networks
- ...

2 Spark GraphX and GraphFrames

GraphX is the Spark RDD-based library for performing graph processing. It is a core part of Spark.

Figure 6: Spark core libraries

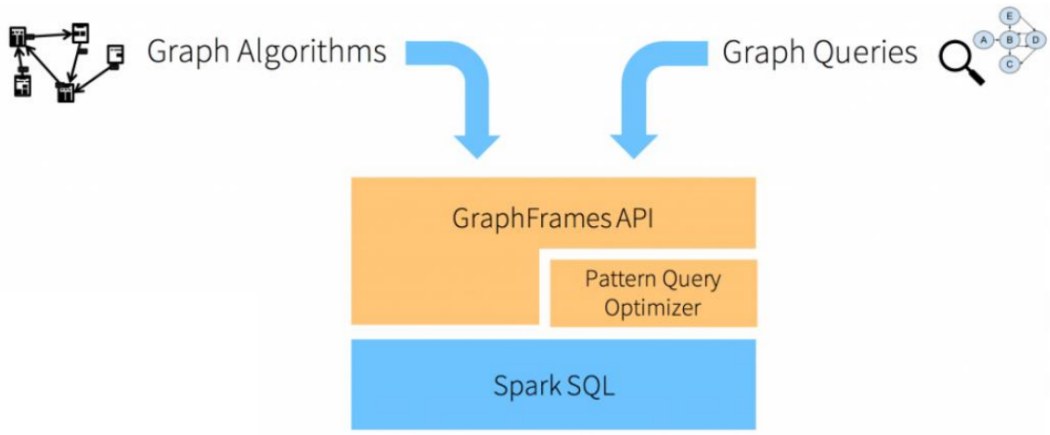


GraphX

- is low level interface with RDD
- is very powerful: many application and libraries built on top of it
- is not easy to use or optimize
- has no Python version of the APIs

GraphFrames is a library DataFrame-based for performing graph processing. It is a Spark external package built on top of GraphX.

Figure 7: GraphFrame structure



3 Building and querying graphs with GraphFrames

3.1 Building a Graph

Define vertexes and edges of the graph: vertexes and edges are represented by means of records inside DataFrames with specifically named columns

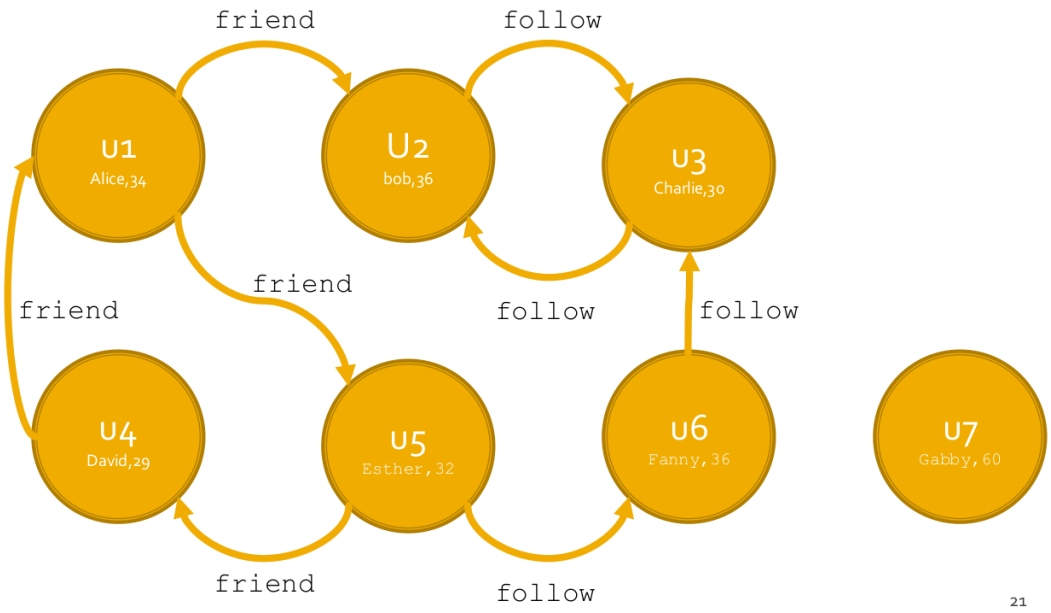
- One DataFrame for the definition of the vertexes of the graph. The DataFrames that are used to represent nodes/vertexes
  - Contain one record per vertex
  - Must contain a column named "id" that stores unique vertex IDs
  - Can contain other columns that are used to characterize vertexes
- One DataFrame for the definition of the edges of the graph. The DataFrames that are used to represent edges
  - Contain one record per edge
  - Must contain two columns "src" and "dst" storing source vertex IDs and destination vertex IDs of edges
  - Can contain other columns that are used to characterize edges

Create a graph of type `graphframes.graphframe.GraphFrame` by invoking the constructor `GraphFrame(v,e)`

- `v`: the DataFrame containing the definition of the vertexes
- `e`: the DataFrame containing the definition of the edges

Graphs in graphframes are directed graphs.

Figure 8: Building a graph example



Example

Given this Vertex DataFrame

id	name	age
u1	Alice	34
u2	Bob	36
u3	Charlie	30
u4	David	29
u5	Esther	32
u6	Fanny	36
u7	Gabby	60

And this Edge DataFrame

src	dst	relationship
u1	u2	friend
u2	u3	follow
u3	u2	follow
u6	u3	follow
u5	u6	follow
u5	u4	friend
u4	u1	friend
u1	u5	friend

```python
from graphframes import GraphFrame

# Vertex DataFrame
v = spark.createDataFrame(
    [
        ("u1", "Alice", 34),
        ("u2", "Bob", 36),
        ("u3", "Charlie", 30),
        ("u4", "David", 29),
        ("u5", "Esther", 32),
        ("u6", "Fanny", 36),
        ("u7", "Gabby", 60)
    ],
    ["id", "name", "age"]
)

# Edge DataFrame
e = spark.createDataFrame(
    [
        ("u1", "u2", "friend"),
        ("u2", "u3", "follow"),
        ("u3", "u2", "follow"),
        ("u6", "u3", "follow"),
        ("u5", "u6", "follow"),
        ("u5", "u4", "friend"),
        ("u4", "u1", "friend"),
        ("u1", "u5", "friend")
    ],
    ["src", "dst", "relationship"]
)

# Create the graph
g = GraphFrame(v, e)
```

3.2 Directed vs undirected edges

In undirected graphs the edges indicate a two-way relationship (each edge can be traversed in both directions). In GraphX it is possible to use `to_undirected()` to create an undirected copy of the Graph. Unfortunately GraphFrames does not support it: it is possible to convert a graph by applying a flatMap function over the edges of the directed graph that creates symmetric edges and then create a new GraphFrame.