Graph Analysis with Flight Data!
Toronto Apache Spark Meetup

Matt McInnis

November 30th 2016

World of
Watson
2016

Big thanks to David Taieb (IBM)
for the original analysis!

IBM

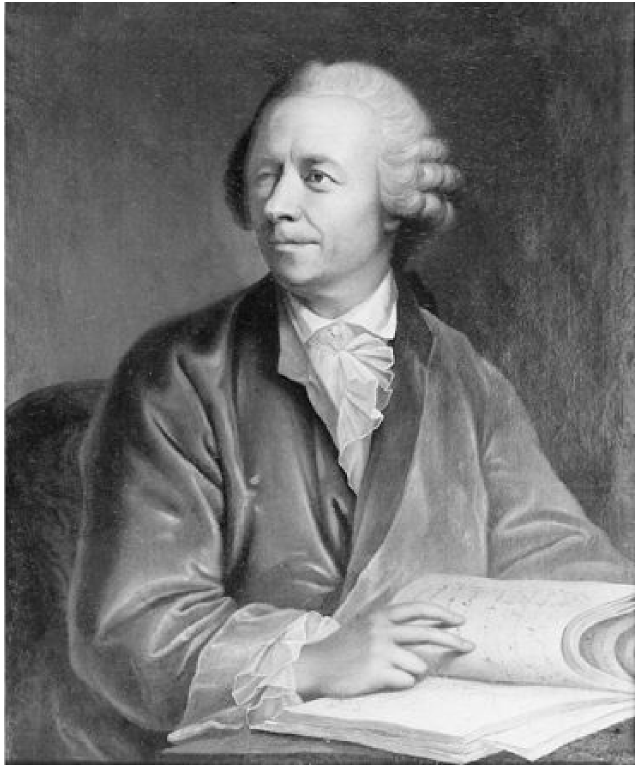
Objectives

By the end of this session, you should be able to:

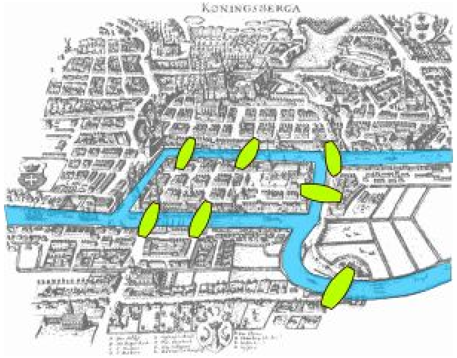
- Have basic knowledge of Graphs, Graph Databases and associated use cases
- Understand Graph processing with Apache Spark GraphX
- Use GraphFrames in a Python Notebook to perform basic Graph parallel computation on airports and flights data

Graphs

A bit of History



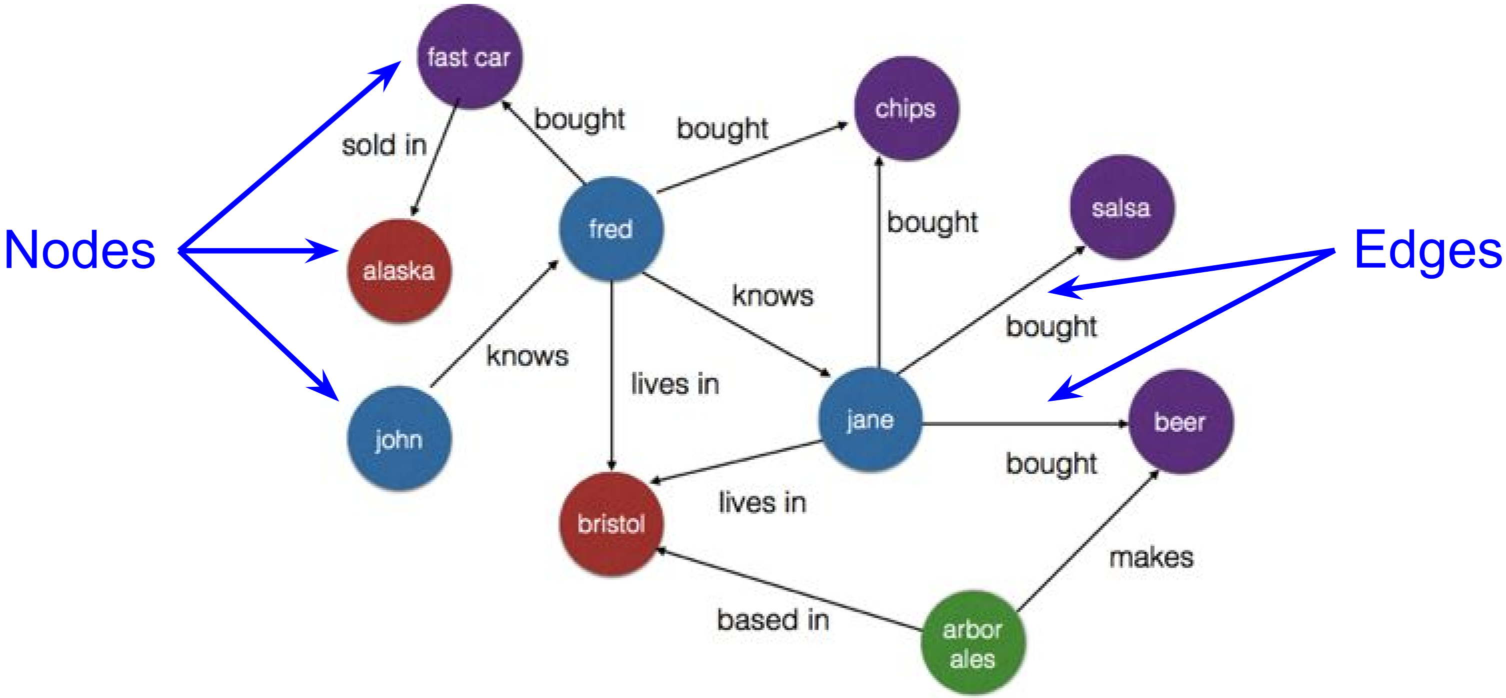
Graph Theory finds its roots from a paper written in 1736 by Leonard Euler on the Seven Bridges of Königsberg urban problem



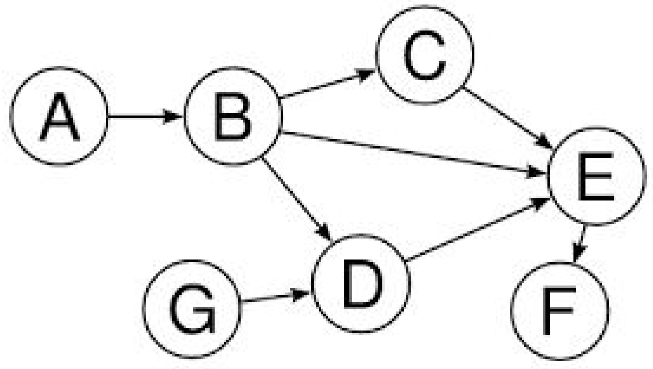
https://en.wikipedia.org/wiki/Graph_theory
https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

Graphs

A graph is a representation of a set of nodes (vertices) where some pairs of nodes are connected by edges.



Types of graphs



Directed Acyclic Graph

- **Directed graphs**

  A graph where the edges have a direction associated with them. An example of directed graph is a Twitter follower. User Bob can follow user Carol without implying that the reciprocal relationship is true

- **Regular graphs**

  Graphs where each vertex has the same number of edges (degree).

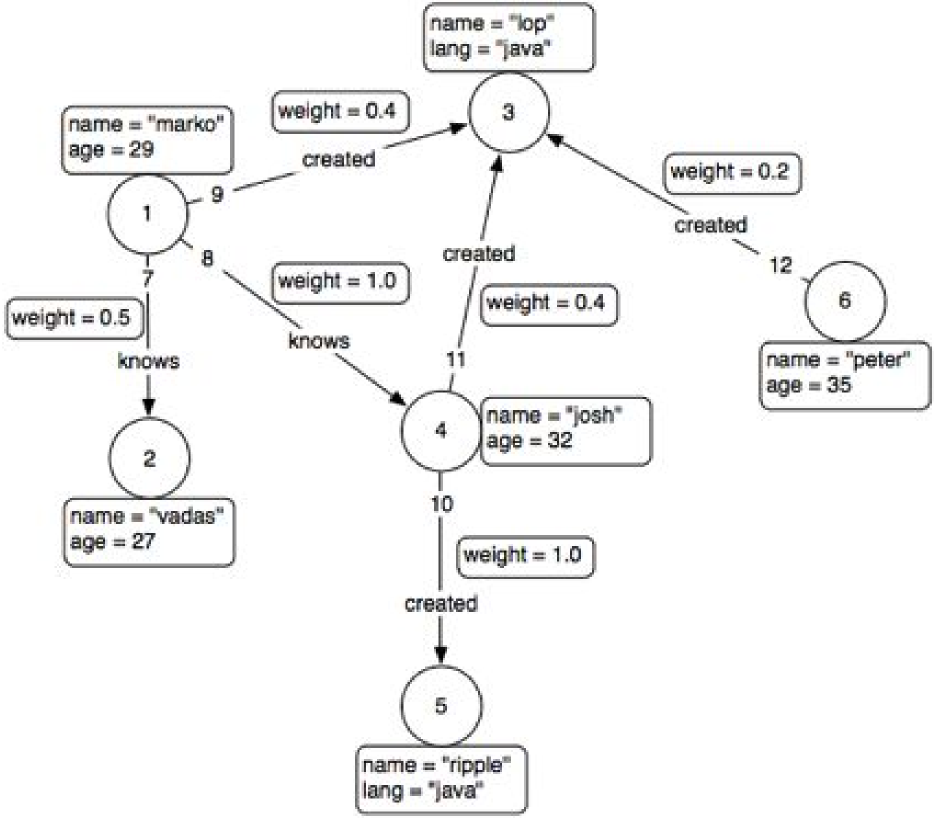
- **Multigraph**

  Graph where multiple edges with same end nodes are permitted

- **Acyclic graphs**

  Graphs which have no cycle

Property Graph



Nodes and edges have properties that can be included in Queries.

Why Graphs?

- Graphs are Central to Analytics
  - Data is not just getting bigger, it’s getting more connected
  - In many use cases, the relationship between data points provides as much value or more than the data points themselves
- Discovering data relationships and interdependencies is critical to many applications
  - fraud detection
  - better understanding customer relationships
  - ranking web pages or people in social networks
- Graph analytics is a powerful tool for understanding and exploiting the connections in data
- Graph applications are everywhere today

Graph Use cases

- Recommendation engines
- Anomaly/Fraud Detection
- Network Analysis/Route planning
- Social Networks
- Identity/Access Management
- Graph-based search
- Master Data Management

Graph is the ‘Natural’ way to represent and query highly connected data

Graph Databases

Titan

IBM Graph

- Provide OLTP (Online Transaction Processing) capabilities.
- Focus on optimizing storage and querying of graph data: vertices, edges, and associated metadata
- Alternative when complexity of graph data makes classic RDBMS inadequate
- Typically work with small sections of the graph
- Example Query engines
  - Gremlin (Tinkerpop)
  - Motif Findings (Neo4J)

Graph processing

APACHE GIRAPH

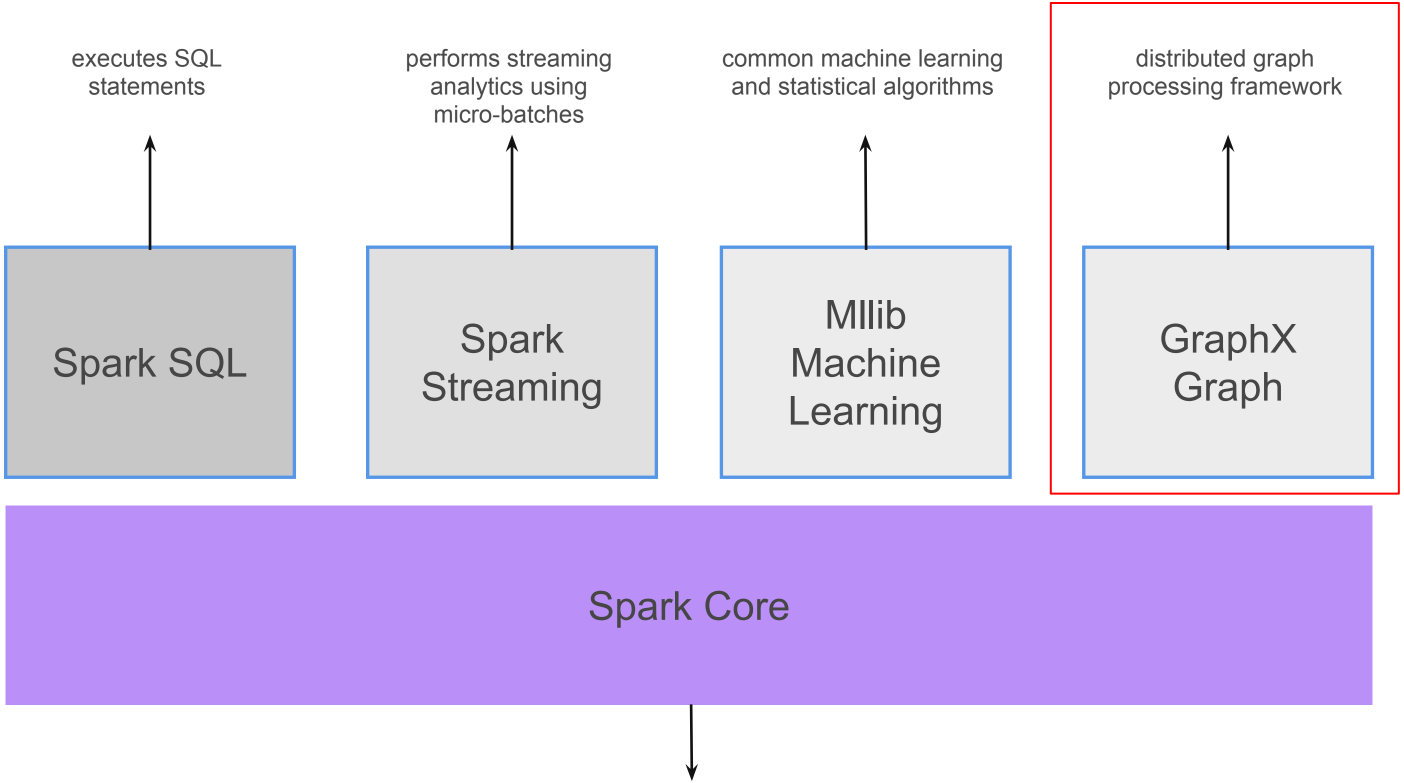
GraphLab

Spark GraphX

- Focuses on data analytics (OLAP or Online Analytical processing) on Graphs
- Suited when relational databases are inadequate because of the data high dimensionality
- Scalability: Distributed Graph-parallel support e.g. BSP (Bulk synchronous processing)
- Example building block operations:
  - Subgraph extractions
  - Neighborhood aggregation
  - …
- Example algorithms:
  - BFS: Breadth First Search
  - PageRank
  - Collaborative Filtering

Apache Spark GraphX

Spark Core Libraries



executes SQL statements

performs streaming analytics using micro-batches

common machine learning and statistical algorithms

distributed graph processing framework

Spark SQL

Spark Streaming

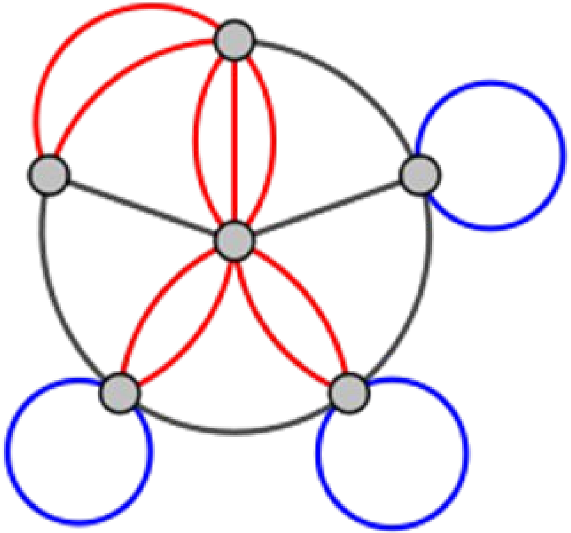
Mllib Machine Learning

GraphX Graph

Spark Core

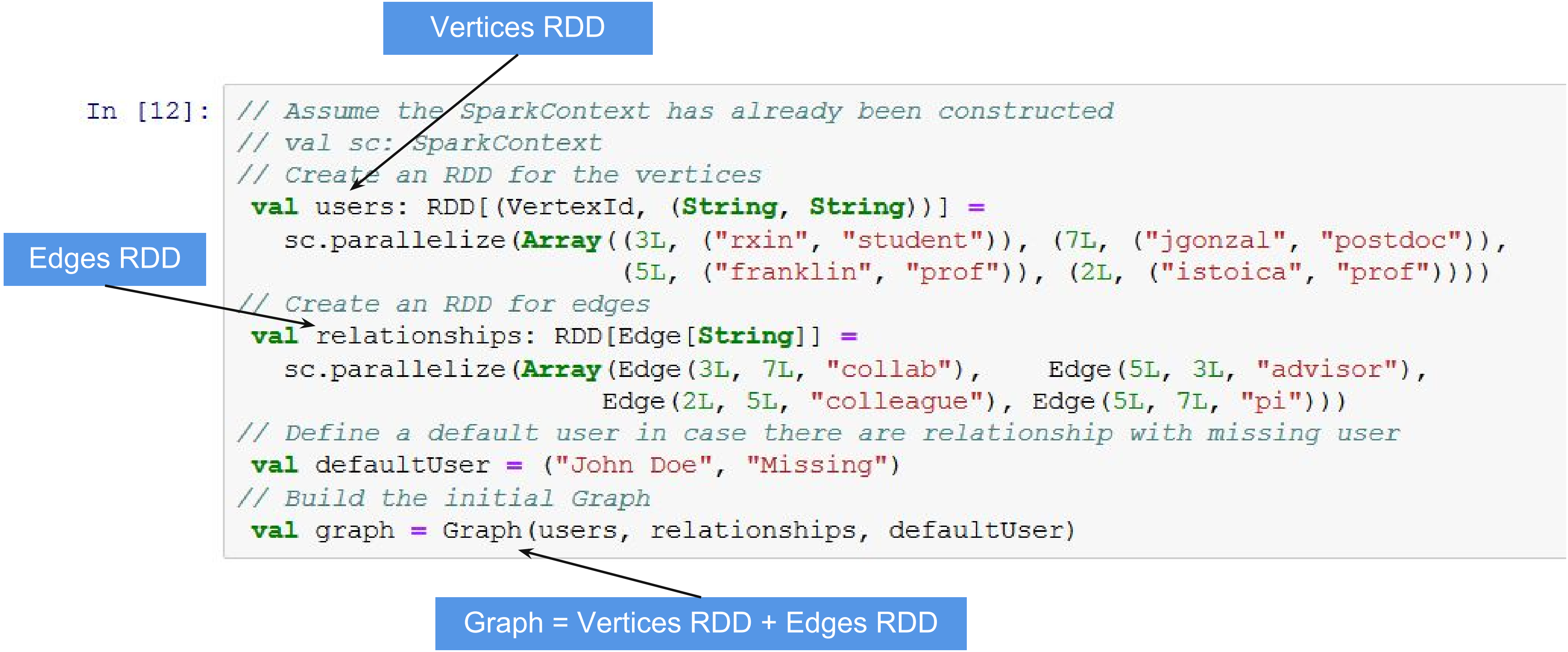
general compute engine, handles distributed task dispatching, scheduling and basic I/O functions

Apache Spark GraphX



- Graph Processing System, not a database
- Directed multigraph with properties attached to each vertex and edges
- Exposes a set of fundamental operators that support graph computation:
  - Subgraph, joinVertices, aggregateMessages, etc…
- Provides algorithms to simplify analytics tasks
  - PageRank, BFS, Triangle Counting, etc…
- Massively parallel: Built on top of Spark RDD

Constructing a property graph with GraphX



Vertices RDD

Edges RDD

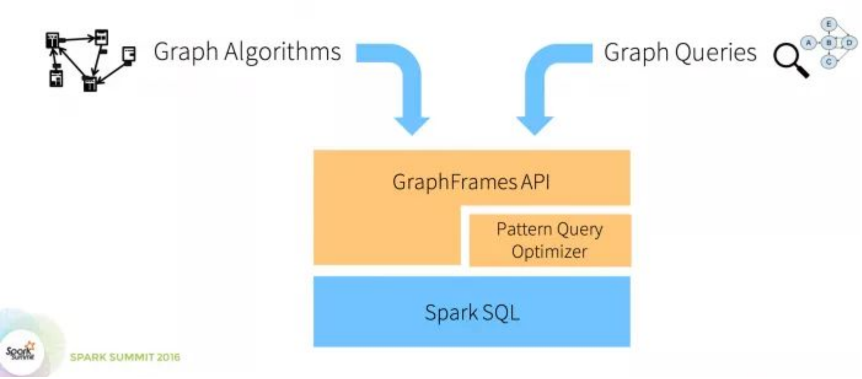
In [12]:

```
// Assume the SparkContext has already been constructed
// val sc: SparkContext
// Create an RDD for the vertices
 val users: RDD[(VertexId, (String, String))] =
   sc.parallelize(Array((3L, ("rxin", "student")), (7L, ("jgonzal", "postdoc")),
                        (5L, ("franklin", "prof")), (2L, ("istoica", "prof"))))
// Create an RDD for edges
 val relationships: RDD[Edge[String]] =
   sc.parallelize(Array(Edge(3L, 7L, "collab"),    Edge(5L, 3L, "advisor"),
                       Edge(2L, 5L, "colleague"), Edge(5L, 7L, "pi")))
// Define a default user in case there are relationship with missing user
 val defaultUser = ("John Doe", "Missing")
// Build the initial Graph
 val graph = Graph(users, relationships, defaultUser)
```

Graph = Vertices RDD + Edges RDD

GraphFrames

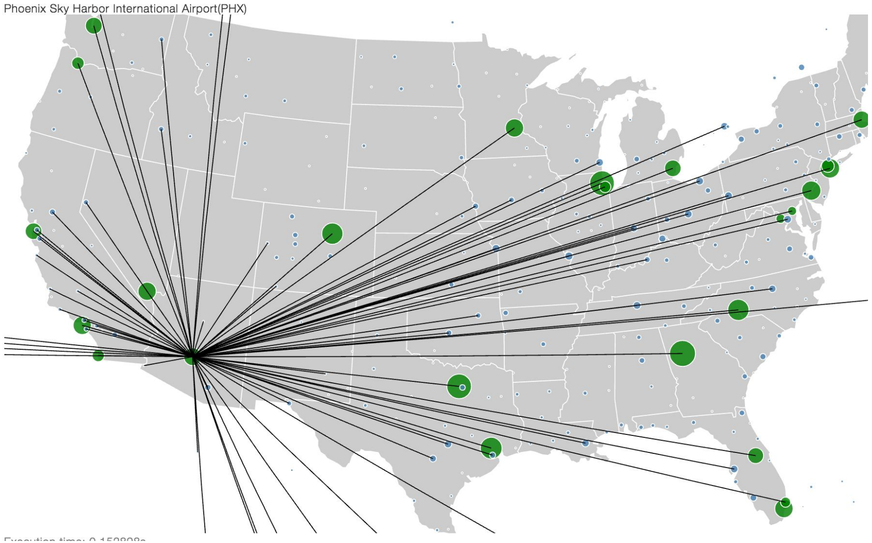
GraphFrames Overview



- Add-on component built on top of GraphX: available on spark-packages.org
- Addresses main limitations of GraphX:
  - Python APIs
  - Uses DataFrames instead of lower-level RDDs: therefore can leverage Catalyst query optimizations
- GraphX-GraphFrame conversion
- Added Feature:
  - Motif finding

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

Demo: Create a GraphFrame from airport and flight data

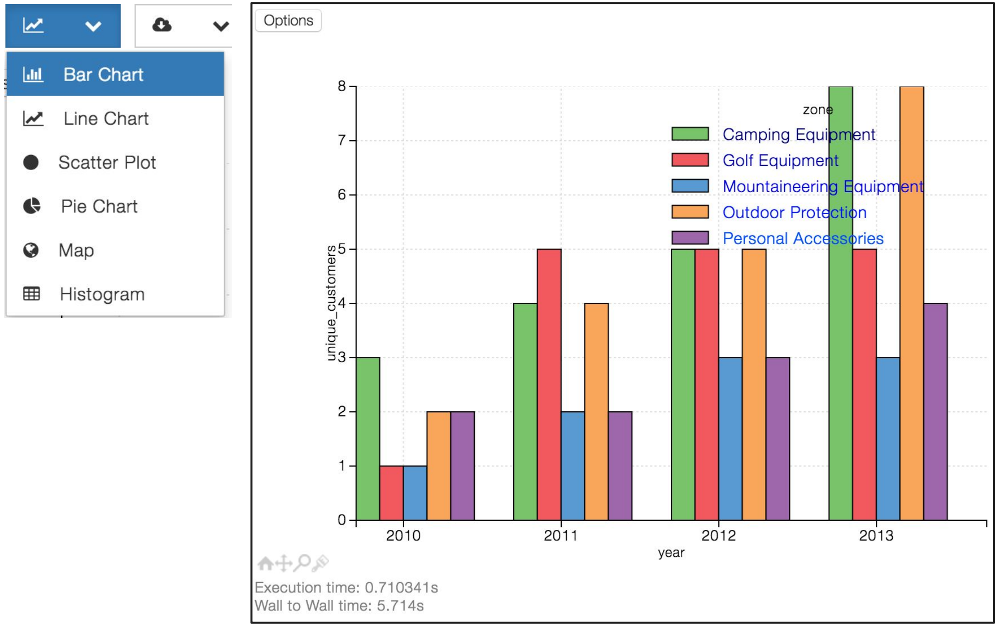

Phoenix Sky Harbor International Airport(PHX)
Execution time: 0.152808s

Demo Steps:

1. Load airport and flight data from Cloudant database
2. Build the Vertex and Edge DataFrame
3. Build the GraphFrame
4. Visualize the graph using PixieDust
5. Graph computation using the Python APIs
6. More Graph Computation using the Scala AggregateMessages API and PixieDust Scala bridge

A word about PixieDust

An Open Source Library that simplifies and improves Jupyter Python Notebooks



```
Execution time: 0.710341s
Wall to Wall time: 5.714s
```

Jupyter + Pixiedust =

1. PackageManager
2. Visualizations
3. Cloud Integration
4. Scala Bridge
5. Extensibility
6. Embedded Apps

https://github.com/ibm-cds-labs/pixiedust

Do code now!

Resources

- Big thanks to David Taieb for the great work building most of this notebook!!
- http://spark.apache.org
- http://www.ibm.com/analytics/us/en/technology/cloud-data-services/spark-as-a-service
- http://datascience.ibm.com
- https://developer.ibm.com/clouddataservices/2016/07/15/intro-to-apache-spark-graphframes/
- https://developer.ibm.com/clouddataservices/2016/10/11/pixiedust-magic-for-python-notebook/
- https://developer.ibm.com/clouddataservices/2016/01/15/real-time-sentiment-analysis-of-twitter-hashtags-with-spark/
- https://developer.ibm.com/clouddataservices/2016/08/04/predict-flight-delays-with-apache-spark-mllib-flightstats-and-weather-data/
- https://github.com/ibm-cds-labs/spark.samples
- https://github.com/ibm-cds-labs/pixiedust

Learn more

Notices and disclaimers

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and disclaimers continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services ®, Global Technology Services ®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Thank You!

World of Watson 2016