SPARK SQL

Recap

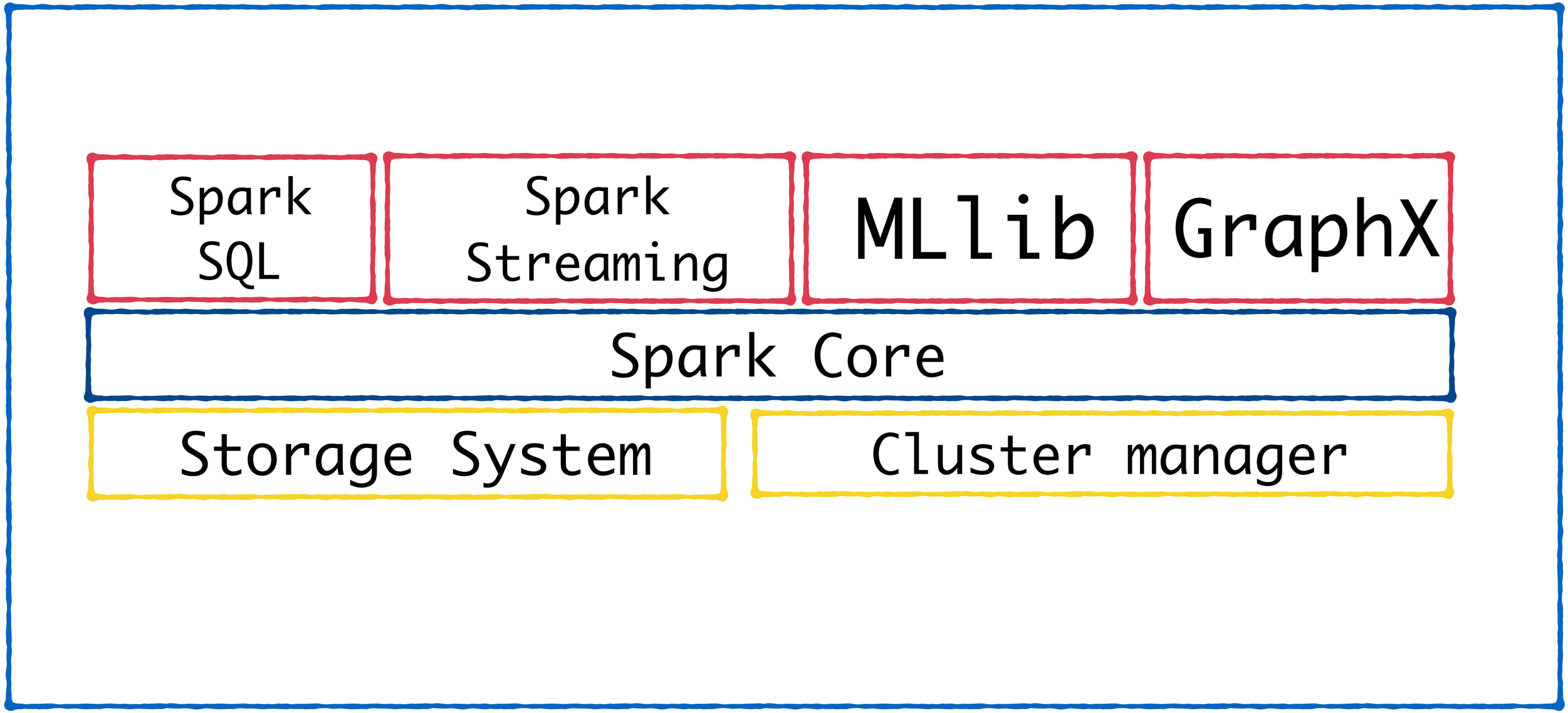
APACHE SPARK

Spark comes with some additional packages that make it truly general - purpose

Spark Core	
Storage System	Cluster manager

Recap

APACHE SPARK



Recap

APACHE SPARK

Spark SQL

Spark SQL provides an SQL interface for Spark

Recap

APACHE SPARK

Spark SQL

Lot's of folks are familiar with **SQL** and like to use to express **data manipulation**

Recap

APACHE SPARK

Spark SQL

With Spark SQL, folks can use SQL to work with large amounts of data stored in a cluster

Recap

APACHE SPARK

Spark SQL

You can use the familiar SQL and still get all the in-memory performance benefits of Spark

Spark SQL

Let's say you were performing some complex data manipulations in a program

Spark SQL

You could load data in table form into memory

using a special type of RDD called a DataFrame

DataFrame

Spark SQL

DataFrames are like in-memory database tables

You can use SQL statements to query and manipulate them

DataFrame

Spark SQL

DataFrames are also like RDDs

You can use all the usual transformations and actions

Spark SQL

Say you wanted to work with some Twitter data

Twitter data

Spark SQL

datahub has a dataset with tweets from the Presidential election in 2012

Twitter data

Spark SQL

datahub
The easy way to get, use and share data

Datasets Organizations About Blog Help Search

/ Organizations / Kingmolnar / **Twitter 2012 Presidential ...**

Twitter 2012 Presidential Election

Followers
1

Organization

Dataset Groups Activity Stream Related

Twitter 2012 Presidential Election

This data-set contains over 170,000,000 tweets collected during 3 months leading up to the 201
presidential elections.

Format

Twitter data

Spark SQL

The records are stored in JSON form

{ "created_at": "Fri Dec 21 22:53:46 +0000 2012", "id": 282257583836889090, "id_str": "282257583836889090", "text": "Obama meets Reid on fiscal cliff, plans remarks http://t.co/Nuec5Wh6", "source": "twitterfeed", "truncated": false, "in_reply_to_status_id": null, "in_reply_to_status_id_str": null, "in_reply_to_user_id": null, "in_reply_to_user_id_str": null, "in_reply_to_screen_name": null, "user": { "id": 27703934, "id_str": "27703934", "name": "Washington Examiner", "screen_name": "washexaminer", "location": "Washington DC", "url": "http://www.washingtonexaminer.com", "description": "", "protected": false, "followers_count": 1995, "friends_count": 67, "listed_count": 94, "created_at": "Mon Mar 30 18:56:55 +0000 2009", "favourites_count": 0, "utc_offset": -18000, "time_zone": "Eastern Time (US & Canada)", "geo_enabled": true, "verified": false, "statuses_count": 42208, "lang": "en", "contributors_enabled": false, "is_translator": false, "profile_background_color": "FFFFFF", "profile_background_image_url": "http://a0.twimg.com/profile_background_images/553522199/twitter-design-Sections1.jpg", "profile_background_image_url_https": "https://si0.twimg.com/profile_background_images/553522199/twitter-design-Sections1.jpg", "profile_background_tile": false, "profile_image_url": "http://a0.twimg.com/profile_images/2223644256/logo-social_normal.gif", "profile_image_url_https": "https://si0.twimg.com/profile_images/2223644256/logo-social_normal.gif", "profile_link_color": "CA0613", "profile_sidebar_border_color": "C0DEED", "profile_sidebar_fill_color": "DDEEF6", "profile_text_color": "333333", "profile_use_background_image": true, "default_profile": false, "default_profile_image": false, "following": null, "follow_request_sent": null, "notifications": null }, "geo": null, "coordinates": null, "place": null, "contributors": null, "retweet_count": 0, "entities": { "hashtags": [], "urls": [{ "url": "http://t.co/Nuec5Wh6", "expanded_url": "http://bit.ly/UWIjMN", "display_url": "bit.ly/UWIjMN", "indices": [48, 68] }], "user_mentions": [] }, "favorited": false, "retweeted": false, "possibly_sensitive": false, "lang": "en" }

Twitter data

Spark SQL

We can treat the JSON file as any other text file

```
twitterData=sc.textFile(twitterPath)
```

Twitter data

Spark SQL

```
import json
twitterData=sc.textFile(twitterPath).map(lambda x:json.loads(x))
```

We can then use Python's JSON parser to parse the data

Twitter data

Spark SQL

```
import json
twitterData=sc.textFile(twitterPath).map(lambda x:json.loads(x))
```

twitterData is an RDD where each record is a **dictionary**

```
twitterData.take(10)
```

```
[{u'contributors': None,
  u'coordinates': None,
  u'created_at': u'Sun Sep 09 21:17:55 +0000 2012',
  u'entities': {u'hashtags': [{u'indices': [87, 95], u'
   u'urls': [{u'display_url': u'STLtoday.com',
     u'expanded_url': u'http://STLtoday.com',
     u'indices': [45, 65],
     u'url': u'http://t.co/OcISvreb'},
    {u'display_url': u'bit.ly/Tzhmly',
     u'expanded_url': u'http://bit.ly/Tzhmly',
     u'indices': [66, 86],
     u'url': u'http://t.co/FsJ7xgGW'}],
   u'user_mentions': []},
  u'favorited': False,
```

Twitter data

Spark SQL

```
import json
twitterData=sc.textFile(twitterPath).map(lambda x:json.loads(x))
```

Each record has all details of a single tweet

```
twitterData.take(10)
```

```
[{u'contributors': None,
  u'coordinates': None,
  u'created_at': u'Sun Sep 09 21:17:55 +0000 2012',
  u'entities': {u'hashtags': [{u'indices': [87, 95], u'
   u'urls': [{u'display_url': u'STLtoday.com',
     u'expanded_url': u'http://STLtoday.com',
     u'indices': [45, 65],
     u'url': u'http://t.co/OcISvreb'},
    {u'display_url': u'bit.ly/Tzhmly',
     u'expanded_url': u'http://bit.ly/Tzhmly',
     u'indices': [66, 86],
     u'url': u'http://t.co/FsJ7xgGW'}],
   u'user_mentions': []},
  u'favorited': False,
```

Twitter data

Spark SQL

```
twitterData
```

You can manipulate this using the usual transformations and actions

map, reduce, aggregate etc

Twitter data

Spark SQL

```
twitterData.\
    filter(lambda x:x['user']['screen_name']=="realDonaldTrump").\
    map(lambda x:x['text'])\
    .take(10)
```

If you wanted to look at a sample of tweets from Donald Trump during this period

Twitter data

Spark SQL

```
twitterData.\
    filter(lambda x:x['user']['screen_name']=="realDonaldTrump").\
    map(lambda x:x['text'])\
    .take(10)
```

First, filter tweets by Donald Trump

Twitter data

Spark SQL

```
twitterData.\
    filter(lambda x:x['user']['screen_name']=="realDonaldTrump").\
    map(lambda x:x['text'])\
    .take(10)
```

Extract the text from those tweets

Twitter data

Spark SQL

```
twitterData.\
    filter(lambda x:x['user']['screen_name']=="realDonaldTrump").\
    map(lambda x:x['text'])\
    .take(10)
```

A sample of tweets

Twitter data

Spark SQL

```
twitterData.\
    filter(lambda x:x['user']['screen_name']=="realDonaldTrump").\
    map(lambda x:x['text'])\
    .take(10)
```

Many folks are comfortable with querying data using SQL

Twitter data

Spark SQL

```
twitterData.\
    filter(lambda x:x['user']['screen_name']=="realDonaldTrump").\
    map(lambda x:x['text'])\
    .take(10)
```

If twitterData were a database table, this same operation could be expressed as

```
SELECT text FROM twitterData
WHERE screen_name='realDonaldTrump'
LIMIT 10
```

Twitter data

Spark SQL

```
twitterData.\
    filter(lambda x:x['user']['screen_name']=="realDonaldTrump").\
    map(lambda x:x['text'])\
    .take(10)
```

With Spark SQL and DataFrames, you can do exactly this!

```
SELECT text FROM twitterData
WHERE screen_name='realDonaldTrump'
LIMIT 10
```

Twitter data

Spark SQL

We normally load data into RDDs using a SparkContext

```
twitterData=sc.textFile(twitterPath)
```

To load data into a DataFrame we need an SQLContext

Twitter data

SQLContext

Spark SQL

```
from pyspark.sql import SQLContext, Row
```

Import SQLContext from pyspark.sql

Twitter data

SQLContext

Spark SQL

```
from pyspark.sql import SQLContext, Row
```

```
sqlC=SQLContext(sc)
```

Use the SparkContext to set up the SQLContext

Twitter data

SQLContext

Spark SQL

```
from pyspark.sql import SQLContext, Row
sqlC=SQLContext(sc)
```

```
twitterTable=sqlC.read.json(twitterPath)
```

The SQLContext has a method to directly read JSON files

Twitter data

SQLContext

Spark SQL

```
twitterTable
```

While the data is loaded, SQL Context will infer the schema of the table

Twitter data

SQLContext

Spark SQL

```
twitterTable
```

The data is loaded into an in-memory table i.e. DataFrame

Twitter data

SQLContext

Spark SQL

```
twitterTable
```

You can use SQL statements to query this DataFrame

Twitter data

SQLContext

Spark SQL

```
twitterTable
```

The SQL statements will be executed by the SQLContext

Twitter data

SQLContext

Spark SQL

```
twitterTable.registerTempTable("twitterTable")
```

First you need to register this DataFrame as a table with the SQLContext

Twitter data

SQLContext

Spark SQL

```
twitterTable.registerTempTable("twitterTable")
```

This is just a bit of setup before you can use Spark SQL with twitterTable

Twitter data

SQLContext

Spark SQL

```
twitterTable.registerTempTable("twitterTable")
```

This is the alias for the Dataframe in Spark SQL

Twitter data

Spark SQL

SQLContext

```
twitterTable.registerTempTable("twitterTable")
```

You can give any name here

Just use the same name in your SQL statements

Twitter data

SQLContext

Spark SQL

```
twitterTable.registerTempTable("twitterTable")
sqlC.sql("Select text, user.screen_name from" +\
         "twitterTable where user.screen_name='realDonaldTrump'"+\
         "limit 10").collect()
```

The SQLContext has **an SQL method** to which you'll give your SQL query

Twitter data

SQLContext

Spark SQL

```
twitterTable.registerTempTable("twitterTable")
sqlC.sql("Select text, user.screen_name from" +\
         "twitterTable where user.screen_name='realDonaldTrump'"+\
         "limit 10").collect()
```

This is just like a Transformation

Twitter data

SQLContext

Spark SQL

```
twitterTable.registerTempTable("twitterTable")
sqlC.sql("Select text, user.screen_name from" +\
         "twitterTable where user.screen_name='realDonaldTrump'"+\
         "limit 10").collect()
```

The output is another DataFrame

Twitter data

SQLContext

Spark SQL

```
twitterTable.registerTempTable("twitterTable")
```

```
sqlC.sql("Select text, user.screen_name from" +\
         "twitterTable where user.screen_name='realDonaldTrump'"+\
         "limit 10").collect()
```

Just like with RDDs you need to use an action to fetch the results

Twitter data

SQLContext

Spark SQL

```
twitterTable.registerTempTable("twitterTable")
```

```
sqlC.sql("Select text, user.screen_name from" +\
         "twitterTable where user.screen_name='realDonaldTrump'"+\
         "limit 10").collect()
```

A DataFrame is like a special kind of RDD

Twitter data

SQLContext

Spark SQL

```
twitterTable.registerTempTable("twitterTable")
```

```
sqlC.sql("Select text, user.screen_name from" +\
         "twitterTable where user.screen_name='realDonaldTrump'"+\
         "limit 10").collect()
```

You can still apply all your regular transformations and actions

Twitter data

Spark SQL

DataFrames are made up of Row objects

```
[Row(text=u'Obama asked a 7 yr old for his birth certi
e ball. (cont) http://t.co/FufZD79U', screen_name=u're
 Row(text=u'Obama is taunting the Republicans on the b
 facts. He (cont) http://t.co/NWmVp06e', screen_name=u
 Row(text=u'"President Obama is the greatest hoax ever
_name=u'realDonaldTrump')]
```

Twitter data

Spark SQL

DataFrames are made up of **Row** objects

Row objects are like structs

You can use the field names to extract the data from the Row

Twitter data

Spark SQL

DataFrames are made up of Row objects

```
twitterTable.map(lambda x:x.text).take(10)
```

```
[u'Obama vies for health care edge in Florida - http://t.co/OcISvreb http://t.co/FsJ7xgGW #florida',
 u'Just crossed the border back to Canada! Had a great time on the State side this weekend #USA #cottage #awesome \ue
50c',
 u"Obama Blasts Romney's Medicare Plan in Appeal to Florida Seniors - ABC News (blog) http://t.co/CyktT3cx #florida",
 u"Suddenly the President was German Suplex'd into a grandma. http://t.co/2ZGGZgYs",
 u'Obama maintains post-convention lead over Romney - http://t.co/XLdRbGPx via http://t.co/sjkJdXlc',
 u'Bear hugs and Medicare for Obama in Florida - Orlando Sentinel http://t.co/lHsa9SKL #florida',
 u'Ryan Blames Obama for Higher Pump Prices - http://t.co/Z5jtBJco http://t.co/4tNoPwk9',
 u'More Bad News for Obama ... http://t.co/8ANM08Vj',
 u'The Daily What http://t.co/S57uzv3a via @TheDailyWhat',
 u'RT @fvdlfvdl: Tomorrow: a shared piece by Rubin and Morris entitled: "Romney\'s brilliant strategy of letting Obam
a take the lead".']
```

Twitter data

Spark SQL

DataFrames are made up of Row objects

```
twitterTable.map(lambda x:x.text).take(10)
```

```
[u'Obama vies for health care edge in Florida - http://t.co/OcISvreb http://t.co/FsJ7xgGW #florida',
 u'Just crossed the border back to Canada! Had a great time on the State side this weekend #USA #cottage #awesome \ue
50c',
 u"Obama Blasts Romney's Medicare Plan in Appeal to Florida Seniors - ABC News (blog) http://t.co/CyktT3cx #florida",
 u"Suddenly the President was German Suplex'd into a grandma. http://t.co/2ZGGZgYs",
 u'Obama maintains post-convention lead over Romney - http://t.co/XLdRbGPx via http://t.co/sjkJdXlc',
 u'Bear hugs and Medicare for Obama in Florida - Orlando Sentinel http://t.co/lHsa9SKL #florida',
 u'Ryan Blames Obama for Higher Pump Prices - http://t.co/Z5jtBJco http://t.co/4tNoPwk9',
 u'More Bad News for Obama ... http://t.co/8ANM08Vj',
 u'The Daily What http://t.co/S57uzv3a via @TheDailyWhat',
 u'RT @fvdlfvdl: Tomorrow: a shared piece by Rubin and Morris entitled: "Romney\'s brilliant strategy of letting Obam
a take the lead".']
```

Twitter data

Spark SQL

As you can see, DataFrames are very versatile!

Use SQL manipulations and Python functions on the very same dataset

Twitter data

Spark SQL

With DataFrames you can

Use SQL manipulations

Python functions

RDD Transformations and Actions

On the same dataset

In the same program