CS 649 Big Data: Tools and Methods
Spring Semester, 2021
Doc 19 Running Spark
Apr 1, 2021

Virus infection Simulation

Assumptions

Infect 10% of people you meet

Contact 20 people per day

New Cases Each Day

0% Immune	50% Immune	80% Immune
1	1	1
2	1	0
6	2	1
18	4	1
54	8	1
162	16	2
486	32	2
1,458	64	3
4,374	128	4
13,122	256	6
39,366	512	8
118,098	1,024	12
354,294	2,048	16

Assumptions

Infect 10% of people you meet

0% Immune

New Cases Each Day

Contact 5 People	Contact 10 People	Contact 20 People	Contact 25 People
1	1	1	1
1	1	2	3
1	2	6	9
1	4	18	31
2	8	54	107
3	16	162	375
4	32	486	1,313
6	64	1,458	4,596
9	128	4,374	16,085
13	256	13,122	56,297
19	512	39,366	197,039
29	1,024	118,098	689,637
43	2,048	354,294	2,413,729

Why We Sleep

By Mathew Walker

Professor of neuroscience and psychology

Director of the Center for Human Sleep Science at the University of California, Berkeley

Effect of lack of sleep

Immune system

Learning

Installing PySpark

Instructions

https://spark.apache.org/docs/latest/api/python/getting_started/install.html

Short Version:

      conda install pyspark

Then We Can Run

```
In [1]:    1    from pyspark.sql import SparkSession
           2    spark = SparkSession.builder \
           3        .master("local") \
           4        .appName("Print") \
           5        .getOrCreate()
           6    print(spark.range(5000).selectExpr("sum(id)").collect())

[Row(sum(id)=12497500)]
```

```
from pyspark.sql import SparkSession
spark = SparkSession.builder \
    .master("local") \
    .appName("Print") \
    .getOrCreate()
print(spark.range(5000).selectExpr("sum(id)").collect())
```

5

Spark Properties

name

master

logging

memory

etc

https://spark.apache.org/docs/latest/configuration.html

name - displayed in Spark Master Web page

master

Master URL	Meaning
local	Run Spark locally with one worker thread.
local[K]	Run Spark locally with K worker threads
local[K,F]	Run Spark locally with K worker threads and F maxFailures
local[*]	Run Spark locally with as many worker threads as logical cores on your machine.
local[*,F]	Run Spark locally with as many worker threads as logical cores on your machine and F maxFailures.
spark://HOST:PORT	Connect to the given Spark standalone cluster master.
spark://HOST1:PORT1,HOST2:PORT2	Connect to the given Spark standalone cluster with standby masters with Zookeeper.
mesos://HOST:PORT	Connect to the given Mesos cluster.
yarn	Connect to a YARN cluster in client or cluster mode

Installing PySpark - Non-Notebook

Download Spark

https://spark.apache.org/downloads.html

Put SPARK/bin and SPARK/sbin on your path

printExample.py   **->spark-submit ./printExample.py**

```python
from __future__ import print_function
def print5000():
    from pyspark.sql import SparkSession
    spark = SparkSession.builder \
        .master("local") \
        .appName("Print") \
        .getOrCreate()
    print(spark.range(5000).selectExpr("sum(id)").collect())

if __name__ == "__main__":
    print5000()
```

Amazon Elastic Map-Reduce (EMR)

Hadoop, Hive, Spark, etc on Cluster

Predefined set of languages/tools available

Can create cluster of machines

https://aws.amazon.com
    Create new account
    Get 12 months free access

AWS Free Tier

12 months free

EC2 - compute instances
   740 hours per month
   Billed in hour increments
   Billed per instance

S3 - storage
   5 GB
   20,000 Get requests

RDS - MySQL, PostgresSQL, SQL Sever
   20 GB
   750 hours

EC2 Container - Docker images
   500 MB

AWS Educate

https://aws.amazon.com/education/awseducate/

SDSU is an institutional member

Students get $100 credit

EC2 Pricing

	Price Per Hour	
	On Demand	Spot
a1.medium	$0.0255	$0.0050
t3.nano	$0.0047	$0.0016
m5.large	$0.0960	$0.0202
c5.large	$0.0850	$0.0200
p3.2xlarge (GPU)	$3.0600	$0.9413

Basic Outline

Develop & test Spark locally

Upload program file & data to S3

Configure & launch cluster
   AWS Management Console
   AWS CLI
   SDKs

Monitor cluster

Make sure you terminate cluster when done

Simple Storage System - S3

Files are stored in buckets

Bucket names are global

Supports
    s3 - files divided in to block
    s3n

Accessing files
    S3 console
    Third party
    REST
    Java, C#, etc
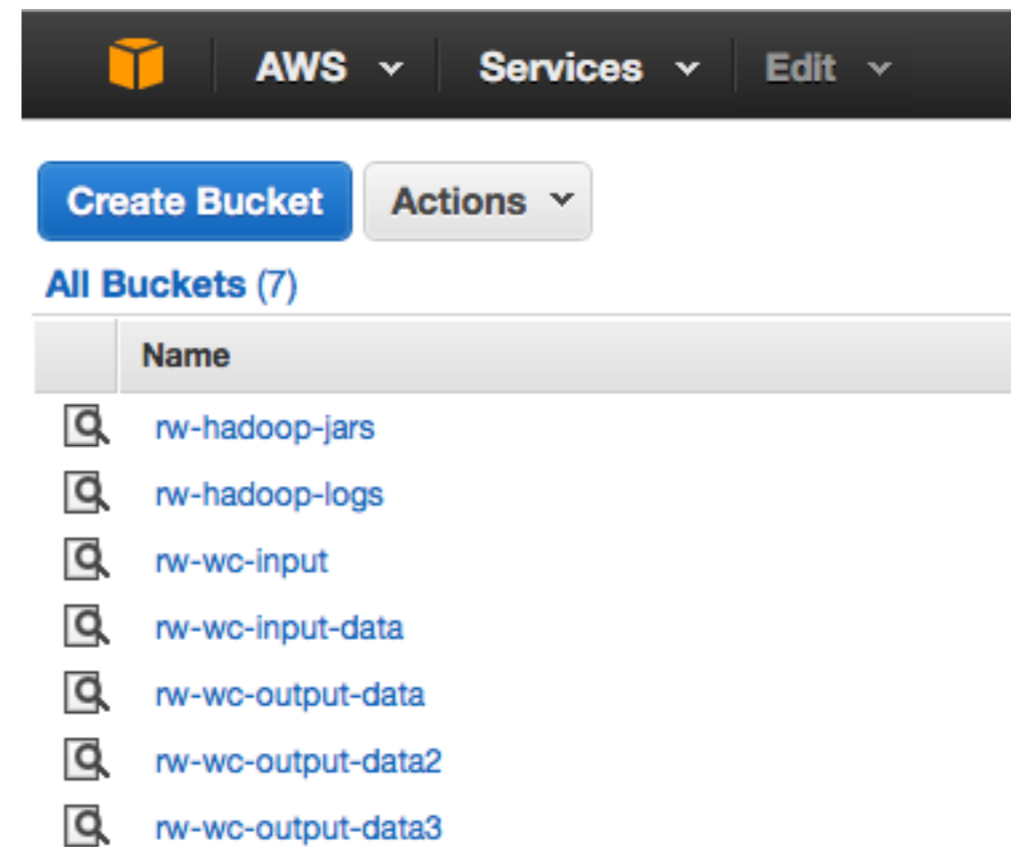
Amazon S3

Buckets (13)

Buckets are containers for data stored in S3. **Learn more** ⬏

| | Copy ARN | Empty | Delete | **Create bucket** |

🔍 Find buckets by name   ⟨ **1** ⟩ | ⚙

	Name ▲	AWS Region ▽	Access ▽	Creation date ▽
○	aws-emr-resources-834365227482-us-west-2	US West (Oregon) us-west-2	Objects can be public	March 26, 2019, 14:47:39 (UTC-07:00)
○	aws-logs-834365227482-us-west-2	US West (Oregon) us-west-2	Objects can be public	December 8, 2016, 15:27:27 (UTC-08:00)
○	aws-sam-cli-managed-default-samclisourcebucket-busdmbpxv8ww	US West (Oregon) us-west-2	Objects can be public	February 25, 2021, 16:19:47 (UTC-08:00)
○	elasticbeanstalk-us-west-2-834365227482	US West (Oregon) us-west-2	Objects can be public	January 27, 2021, 21:50:54 (UTC-08:00)
○	rew-648-test	US West (Oregon) us-west-2	Objects can be public	February 22, 2021, 20:31:11 (UTC-08:00)
○	rw-648-data	US West (Oregon) us-west-2	Objects can be public	February 7, 2021, 14:47:53 (UTC-08:00)
○	rw-696-flight	US West (Oregon) us-west-2	Objects can be public	March 26, 2019, 09:56:35 (UTC-07:00)
○	rw-cs696-data	US West (Oregon) us-west-2	Objects can be public	November 6, 2017, 14:47:10 (UTC-08:00)

15

S3 Creating a Bucket

Create bucket

Buckets are containers for data stored in S3. Learn more 🔗

General configuration

Bucket name

> myawsbucket

Bucket name must be unique and must not contain spaces or uppercase letters. **See rules for bucket naming** 🔗

AWS Region

> US West (Oregon) us-west-2 ▼

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

> Choose bucket

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public

S3 Costs

AWS Free Usage Tier

New AWS customers receive each month for one year
5 GB of Amazon S3 storage in the Standard Storage class,
20,000 Get Requests,
2,000 Put Requests, and
15 GB of data transfer out

	Standard Storage	Standard - Infrequent Access Storage	Glacier Storage
First 50 TB / month	$0.023 per GB	$0.0125 per GB	$0.004 per GB
Next 450 TB / month	$0.022 per GB	$0.0125 per GB	$0.004 per GB
Over 500 TB / month	$0.021 per GB	$0.0125 per GB	$0.004 per GB

S3 Objects

Objects contain
  Object data
  Metadata

Size
  1 byte to 5 gigabytes per object

Object data
  Just bytes
  No meaning associated with bytes

Metadata
  Name-value pairs to describe the object
  Some http headers used
    Content-Type

S3 Buckets

Namespace for objects

No limitation on number of object per bucket

Only 100 buckets per account

Each bucket has a name
   Up to 255 bytes long
   Cannot be same as existing bucket name by any S3 user

Bucket Names

Bucket names must

Contain lowercase letters, numbers, periods (.), underscores (_), and dashes (-)

Start with a number or letter

Be between 3 and 255 characters long

Not be in an IP address style (e.g., "192.168.5.4")

To conform with DNS requirements, Amazon recommends

Bucket names should not contain underscores (_)

Bucket names should be between 3 and 63 characters long

Bucket names should not end with a dash

Bucket names cannot contain dashes next to periods (e.g.,

"my-.bucket.com" and "my.-bucket" are invalid

Key

Unique identifier for an object within a bucket

Object Url

   http://buckerName.s3.amazonaws.com/Key

http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl

   Bucket = doc
   Key = 2006-03-01/AmazonS3.wsdl

Access Control Lists (ACL)

Each Bucket has an ACL

Determines who has read/write access

Each Object can have an ACL

Determines who has read/write access

ACL consists of a list of grants

Grant contains

One grantee

One permission

S3 Data Consistency Model

Updates to a single object at a key in a bucket are atomic

But a read after a write may return the old value
   Changes may take time to progate

No object locking
   If two writes to same object occur at the same time
   The one with later timestamp wins

Running Program on AWS EMR

Make sure program runs locally

Create program file containing code

Create s3 bucket(s) for
  program file file
  logs
  input
  output

Upload program & data files to s3

Spark Components
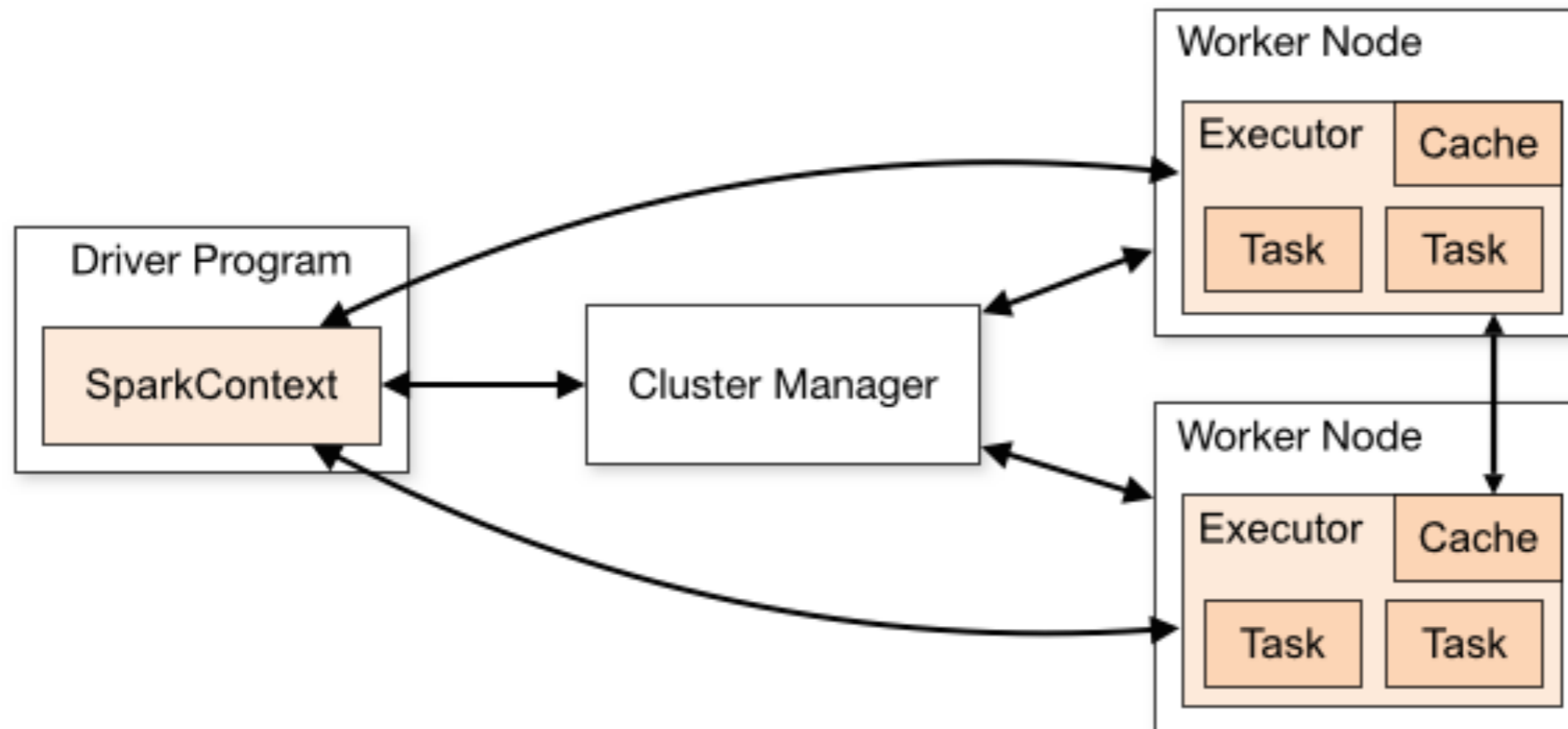
Terms

Application

    User program built on Spark

    Driver program + executors

Driver program

    The process running the main() function of the application and creating the SparkContext
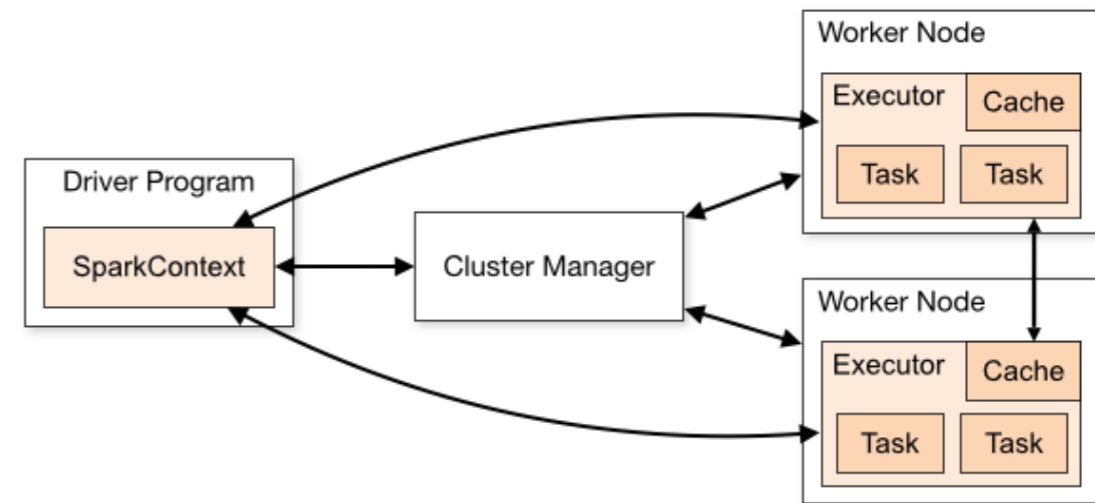
Cluster manager

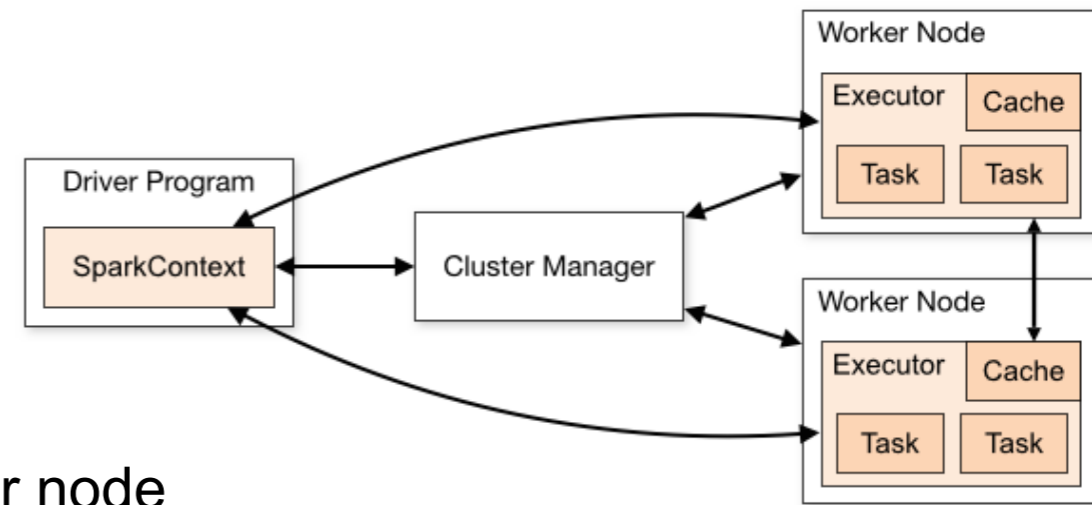   External service for acquiring resources on the cluster

Deploy mode

    Where the driver process runs

    "cluster" - the driver inside of the cluster

    "client" - the driver outside of the cluster

Terms

Executor

    A process launched for an application on a worker node

    Runs tasks and keeps data in memory or disk storage across them.

    Each application has its own executors

Task

    A unit of work that will be sent to one executor

Job

    A parallel computation consisting of multiple tasks

    Gets spawned in response to a Spark action (e.g. save, collect)

Stage

    Job divided into smaller tasks called stages

    Depend on each other

Test Program 1 - Pi

Designed to have no
  Command line dependancies
  No input or output files

```
from random import random
from operator import add

from pyspark import SparkContext

if __name__ == "__main__":
    sc = SparkContext(appName="PythonPi")
    partitions =  3
    n = 100000 * partitions

    def f(_):
        x = random() * 2 - 1
        y = random() * 2 - 1
        return 1 if x ** 2 + y ** 2 < 1 else 0

    count = sc.parallelize(range(1, n + 1), partitions).map(f).reduce(add)
    print("Pi is roughly %f" % (4.0 * count / n))

    sc.stop()
```

My S3 Buckets

Buckets (9)

🔍 Find bucket by name

	Name ▽	Region
○	aws-emr-resources-834365227482-us-west-2	US West (Oregon) us-west-2
○	aws-logs-834365227482-us-west-2	US West (Oregon) us-west-2
●	rw-696-flight	US West (Oregon) us-west-2
○	rw-cs696-data	US West (Oregon) us-west-2
○	rw-hadoop-jars	US West (Oregon) us-west-2
○	rw-hadoop-logs	US West (Oregon) us-west-2
○	rw-output-data	US West (Oregon) us-west-2
○	rw-ws-input-data	US West (Oregon) us-west-2

My S3 Buckets

Overview	Properties	Permissions	Management	Access points

🔍 Type a prefix and press Enter to search. Press ESC to clear.

⬆ Upload **＋ Create folder** Download Actions ⌄

☐ Name ▾

☐ flight.py

☐ pi.py

My S3 Buckets

pi.py   Latest version ▼

Overview	Properties	Permissions	Select from

Open   Download   Download as   Make public   Copy path

Owner
rwhitney

Last modified
Mar 27, 2019 8:58:15 PM GMT-0700

Etag
2a37d19cc32e1a51c0472473e26f72d1

Storage class
Standard

Server-side encryption
None

Size
498.0 B

Key
pi.py

Object URL
https://rw-696-flight.s3-us-west-2.amazonaws.com/pi.py

Spark on AWS - EMR Console

Using Quick Options

General Configuration

Cluster name | My cluster

☑ Logging ⓘ

S3 folder | s3://aws-logs-834365227482-us-west-2/elastici 📁

Launch mode ● Cluster ⓘ ○ Step execution ⓘ

Software configuration

Release | emr-6.2.0 | ⓘ ◄────────────────

Applications
- ○ Core Hadoop: Hadoop 3.2.1 with Hive 3.1.2, Hue 4.8.0, Pig 0.17.0 and Tez 0.9.2
- ○ HBase: HBase 2.2.6 with Hadoop 3.2.1, Hive 3.1.2, Hue 4.8.0, Phoenix 5.0.0, and ZooKeeper 3.4.14
- ○ Presto: Presto 0.238.3 with Hadoop 3.2.1 HDFS and Hive 3.1.2 Metastore
- ○ PrestoSQL: PrestoSQL 343 with Hadoop 3.2.1 HDFS and Hive 3.1.2 Metastore
- ● Spark: Spark 3.0.1 on Hadoop 3.2.1 YARN with and Zeppelin 0.9.0-preview1 ◄────────────

☐ Use AWS Glue Data Catalog for table metadata ⓘ

Hardware configuration

Instance type | m3.xlarge | ◄────────────

Number of instances | 3 | (1 master and 2 core nodes)

Cluster scaling ☐ scale cluster nodes based on workload

Security and access

EC2 key pair | Choose an option | ⓘ Learn how to create an EC2 key pair.

Permissions ● Default ○ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role EMR_DefaultRole 🔗 ⓘ

EC2 instance profile EMR_EC2_DefaultRole 🔗 ⓘ

33

Cancel | **Create cluster** ◄────────────

Use Advanced Options

Create Cluster - Quick Options   Go to advanced options

General Configuration

Cluster name   My cluster

☑ Logging ⓘ

S3 folder   s3://aws-logs-834365227482-us-west-2/elasticmapreduce/ 📁

Launch mode   ⦿ Cluster ⓘ   ○ Step execution ⓘ

Software configuration

Release   emr-5.22.0 ⓘ

Applications
⦿ Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.4, Hue 4.3.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.1

○ HBase: HBase 1.4.9 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.4, Hue 4.3.0, Phoenix 4.14.1, and ZooKeeper 3.4.13

○ Presto: Presto 0.214 with Hadoop 2.8.5 HDFS and Hive 2.3.4 Metastore

○ Spark: Spark 2.4.0 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.1

☐ Use AWS Glue Data Catalog for table metadata ⓘ

Advanced Options

Software Configuration

Release `emr-6.2.0` ⬍ ⓘ

- ☑ Hadoop 3.2.1
- ☐ JupyterHub 1.1.0
- ☐ Ganglia 3.7.2
- ☐ Hive 3.1.2
- ☐ ZooKeeper 3.4.14
- ☐ Sqoop 1.4.7
- ☐ Oozie 5.2.0
- ☐ TensorFlow 2.3.1

- ☐ Zeppelin 0.9.0
- ☐ Tez 0.9.2
- ☐ HBase 2.2.6-amzn-0
- ☐ Presto 0.238.3
- ☐ JupyterEnterpriseGateway 2.1.0
- ☐ Hue 4.8.0
- ☑ Spark 3.0.1

- ☐ Livy 0.7.0
- ☐ Flink 1.11.2
- ☐ Pig 0.17.0
- ☐ PrestoSQL 343
- ☐ MXNet 1.7.0
- ☐ Phoenix 5.0.0
- ☐ HCatalog 3.1.2

Multiple master nodes (optional)

- ☐ Use multiple master nodes to improve cluster availability. Learn more ↗

AWS Glue Data Catalog settings (optional)

- ☐ Use for Spark table metadata ⓘ

Edit software settings ⓘ

- ◉ Enter configuration ○ Load JSON from S3

```
classification=config-file-name,properties=[myKey1=myValue1,myKey2=myValue2]
```

Steps (optional)

A step is a unit of work you submit to the cluster. For instance, a step might contain one or more Hadoop or Spark jobs. You can also submit additional steps to a cluster after it is running. Learn more ↗

Concurrency: ☐ Run multiple steps at the same time to improve cluster utilization

After last step completes: ◉ Clusters enters waiting state
○ Cluster auto-terminates

Step type `Spark application` ⬍ [Add step]

Cancel   **Next**

35

Advanced Options

Hardware Configuration ⓘ

If you need more than 20 EC2 instances, see this topic ↗.

Instance group configuration	🔵 **Uniform instance groups** *Specify a single instance type and purchasing option for each node type.* ⚪ **Instance fleets** *Specify target capacity and how Amazon EMR fulfills it for each node type. Mix instance types and purchasing options.* Learn more ↗

Launch the cluster into a VPC with a public, private, or shared subnet. Subnets may be associated with an AWS Outpost or AWS Local Zone.

Network `vpc-9a87d8fe (172.31.0.0/16) (default)` ⬍   Create a VPC ↗ ⓘ

EC2 Subnet `subnet-0f55196b | Default in us-west-2a` ⬍

Root device EBS volume size `10` GiB ⓘ

Choose the instance type, number of instances, and a purchasing option. You can choose to use On-Demand Instances, Spot Instances, or both. The instance type and purchasing option apply to all EC2 instances in each instance group, and you can only specify these options for an instance group when you create it. Learn more about instance purchasing options ↗

Node type	Instance type	Instance count	Purchasing option
Master Master - 1 ✏️	**m5.xlarge** ✏️ 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB ⓘ ✏️ Add configuration settings ✏️	1 Instances	🔵 **On-demand** ⓘ ⚪ **Spot** ⓘ `Use on-demand as max price` ⬍
Core Core - 2 ✏️	**m5.xlarge** ✏️ 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB ⓘ ✏️ Add configuration settings ✏️	`2` Instances	🔵 **On-demand** ⓘ ⚪ **Spot** ⓘ `Use on-demand as max price` ⬍
Task ✖️ Task - 3 ✏️	**m5.xlarge** ✏️ 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB ⓘ ✏️ Add configuration settings ✏️	`0` Instances	🔵 **On-demand** ⓘ ⚪ **Spot** ⓘ `Use on-demand as max price` ⬍

+ Add task instance group

36

Advanced Options

General Options

Cluster name | CourseSlidesExample

☑ Logging ⓘ

    S3 folder | s3://aws-logs-834365227482-us-west-2/elastic ⟵

☑ Debugging ⓘ

☑ Termination protection ⓘ

Tags ⓘ

Key	Value (optional)	
Add a key to create a tag		

Additional Options

☐ EMRFS consistent view ⓘ

Custom AMI ID | None ⓘ

▸ Bootstrap Actions

Cancel   Previous   Next

Advanced Options

Security Options

EC2 key pair | Proceed without an EC2 key pair ⬍ | ℹ ⟵

☑ Cluster visible to all IAM users in account ℹ

Permissions ℹ

⦿ Default ◯ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role EMR_DefaultRole ↗ ℹ

EC2 instance profile EMR_EC2_DefaultRole ↗ ℹ

Auto Scaling role EMR_AutoScaling_DefaultRole ↗ ℹ

▶ Security Configuration

▶ EC2 security groups

ℹ No EC2 key pair has been selected, so you will not be able to SSH to this cluster. Learn how to create an EC2 Key Pair.

Cancel   Previous   **Create cluster**

Cluster Created - Either Quick or Advanced

Clone	Terminate	AWS CLI export

Cluster: CourseSlidesExample   Starting

Summary	Application history	Monitoring	Hardware	Configurations	Events	Steps	Bootstrap actions

Connections: --

Master public DNS: --

History service: --

Tags: -- View All / Edit

Summary

ID: j-173L2X4KKU7HX

Creation date: 2020-03-24 12:05 (UTC-7)

Elapsed time: 30 seconds

After last step completes: Cluster waits

Termination protection: On Change

Configuration details

Release label: emr-5.29.0

Hadoop distribution: Amazon 2.8.5

Applications: Spark 2.4.4

Log URI: s3://aws-logs-834365227482-us-west-2/elasticmapreduce/

EMRFS consistent view: Disabled

Custom AMI ID: --

Network and hardware

Availability zone: us-west-2a

Subnet ID: subnet-0f55196b ↗

Master: Provisioning 1 m5.xlarge

Core: Provisioning 2 m5.xlarge

Task: --

39

Adding a Step

Clone   Terminate   AWS CLI export

Cluster: CourseSlidesExample   Starting

| Summary | Application history | Monitoring | Hardware | Configurations | Events | Steps | Bootstrap actions |

Concurrency: 1 Change

After last step completes: Cluster waits

Add step   Clone step   Cancel step

Filter: All steps   Filter steps ...   1 step (all loaded) ↻

	ID	Name	Status
▶ ○	s-18YQWR7FSE5XT	Setup hadoop debugging	Pending

Add step ✕

Step type	Spark application ⇕
Name	Spark application
Deploy mode	Client ⇕ Run your driver on a slave node (cluster mode) or on the master node as an external client (client mode).
Spark-submit options	Specify other options for spark-submit.
Application location*	s3://rw-696-flight/pi.py Path to a JAR with your application and dependencies (client deploy mode only supports a local path).
Arguments	Specify optional arguments for your application.
Action on failure	Continue ⇕ What to do if the step fails.

Cancel   **Add**

Modes - client or cluster

Either works

client mode gives access to standard out

41

Clone | Terminate | AWS CLI export

Cluster: CourseSlidesExample   Starting Configuring cluster software

| Summary | Application history | Monitoring | Hardware | Configurations | Events | Steps | Bootstrap actions |

Concurrency: 1 Change

After last step completes: Cluster waits

Add step | Clone step | Cancel step

Filter: [All steps ⇕] [Filter steps ...]   2 steps (all loaded) ↻

		ID	Name	Status	Start time (UTC-7) ▾	Elapsed time	Log files ⬈
○	▸ ○	s-3E10OBDAULFO	Spark application	Pending		--	View logs
○	▸ ○	s-18YQWR7FSE5XT	Setup hadoop debugging	Pending		--	View logs

42

Cluster: CourseSlidesExample   Waiting Cluster ready after last step completed.

| Summary | Application history | Monitoring | Hardware | Configurations | Events | Steps | Bootstrap actions |

Concurrency: 1 Change

After last step completes: Cluster waits

[Add step] [Clone step] [Cancel step]

Filter: [All steps ⌄] [Filter steps ...]   2 steps (all loaded) ↻

		ID	Name	Status	Start time (UTC-7) ⌄	Elapsed time	Log files ↗
○	▶	s-3E10OBDAULFO	Spark application	Completed	2020-03-24 12:10 (UTC-7)	24 seconds	View logs
○	▶	s-18YQWR7FSE5XT	Setup hadoop debugging	Completed	2020-03-24 12:09 (UTC-7)	2 seconds	View logs

Concurrency: 1 Change

After last step completes: Cluster waits

[Add step] [Clone step] [Cancel step]

View Jobs in the

Filter: [All steps ⌄] [Filter steps ...]   2 steps (all loaded) ↻

		ID	Name	Status	Start time (UTC-7) ⌄	Elapsed time	Log files ↗
○	▶	s-3E10OBDAULFO	Spark application	Completed	2020-03-24 12:10 (UTC-7)	24 seconds	controller	syslog*	stderr	stdout ↻
○	▶	s-18YQWR7FSE5XT	Setup hadoop debugging	Completed	2020-03-24 12:09 (UTC-7)	2 seconds	View logs

Pi is roughly 3.138120

43

Example 2

```python
def flight(input, output):
    import pyspark.sql.functions as F
    from pyspark.sql import SparkSession
    spark = SparkSession.builder \
        .appName("Fight") \
        .getOrCreate()

    flight_df = spark.read.json(input)

    grouped_df = flight_df.groupBy('DEST_COUNTRY_NAME').agg(F.sum('count'))
    grouped_df.write.format('csv').save(output)

def files_from_args():
    import argparse
    parser = argparse.ArgumentParser()
    parser.add_argument('-i', '--input', default='input')
    parser.add_argument('-o', '--output',default='output')
    args = parser.parse_args()
    return (args.input, args.output)

if __name__ == "__main__":
    inputfile, outputfile = files_from_args()
44 flight(inputfile, outputfile)
```

S3 Buckets

▶ 🗑 aws-emr-resources-834365227482-us-west-2	•	
▶ 🗑 aws-logs-834365227482-us-west-2	•	
▼ 🗑 rw-696-flight	•	
📄 2015-summary.json	21 KB	3/26/19
📄 flight.py	2 KB	3/26/19
📄 pi.py	49...tes	3/27/19
▶ 🗑 rw-cs696-data	•	
▶ 🗑 rw-hadoop-jars	•	

Adding a Step

Add step   ✕

Step type	Spark application ⇅
Name	Spark application
Deploy mode	Client ⇅ Run your driver on a slave node (cluster mode) or on the master node as an external client (client mode).
Spark-submit options	Specify other options for spark-submit.
Application location*	s3://rw-696-flight/flight.py Path to a JAR with your application and dependencies (client deploy mode only supports a local path).
Arguments	-o s3://rw-output-data/run2 -i s3://rw-696-flight/2015-summary.json Specify optional arguments for your application.
Action on failure	Continue ⇅ What to do if the step fails.

Cancel   **Add**

S3 output

- ▼ 🗑 rw-output-data   •
  - ▼ 📁 run2   •
    - _SUCCESS   Zero KB   3/28/19
    - part-00000-d6b0488c-30c4-472a-9a53-0a...   Zero KB   3/28/19
    - part-00002-d6b0488c-30c4-472a-9a53-0a...   35...tes   3/28/19
    - part-00008-d6b0488c-30c4-472a-9a53-0a...   11...tes   3/28/19
    - part-00009-d6b0488c-30c4-472a-9a53-0a...   11...tes   3/28/19
    - part-00010-d6b0488c-30c4-472a-9a53-0a...   12...tes   3/28/19
    - part-00012-d6b0488c-30c4-472a-9a53-0a...   10...tes   3/28/19
    - part-00013-d6b0488c-30c4-472a-9a53-0abd52629b65-c000.csv
    - part-00014-d6b0488c-30c4-472a-9a53-0a...   11...tes   3/28/19
    - part-00015-d6b0488c-30c4-472a-9a53-0a...   23...tes   3/28/19
    - part-00016-d6b0488c-30c4-472a-9a53-0a...   8 bytes   3/28/19
    - part-00017-d6b0488c-30c4-472a-9a53-0a...   11...tes   3/28/19
    - part-00021-d6b0488c-30c4-472a-9a53-0a...   7 bytes   3/28/19
    - part-00022-d6b0488c-30c4-472a-9a53-0a...   13...tes   3/28/19
    - part-00026-d6b0488c-30c4-472a-9a53-0a...   10...tes   3/28/19
    - part-00029-d6b0488c-30c4-472a-9a53-0a...   9 bytes   3/28/19
    - part-00030-d6b0488c-30c4-472a-9a53-0a...   40...tes   3/28/19
    - part-00031-d6b0488c-30c4-472a-9a53-0a...   10...tes   3/28/19
    - part-00032-d6b0488c-30c4-472a-9a53-0a...   9 bytes   3/28/19
    - part-00038-d6b0488c-30c4-472a-9a53-0a...   50...tes   3/28/19
    - part-00040-d6b0488c-30c4-472a-9a53-0a...   10...tes   3/28/19
    - part-00044-d6b0488c-30c4-472a-9a53-0a...   14...tes   3/28/19
    - part-00045-d6b0488c-30c4-472a-9a53-0a...   22...tes   3/28/19

Warning on AWS

It can take 5-10 minutes to start cluster

Logs do not show your logging statements

When you configure Steps incorrectly they fail
   Error messages are not very helpful

SSH to your Master Node

Create Amazon EC2 Key pair

Instructions

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#having-ec2-create-your-key-pair

Open EC2 Dashboard - Select Key Pairs

SSH to your Master Node

In Create Cluster - Quick Options

Hive 2.3.0 Metastore

◉ Spark: Spark 2.2.0 on Hadoop 2.7.3 YARN with Ganglia 3.7.2 and Zeppelin 0.7.2

☐ Use AWS Glue Data Catalog for table metadata ⓘ

Hardware configuration

Instance type	m3.xlarge	⇕
Number of instances	3	(1 master and 2 core nodes)

Security and access

EC2 key pair	Choose an option ⇕	ⓘ Learn how to create an EC2 key pair.
Permissions	◉ Default ○ Custom	

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role EMR_DefaultRole ⓘ

EC2 instance profile EMR_EC2_DefaultRole ⓘ

Cancel   **Create cluster**

SSH to your Master Node

Clone   Terminate   AWS CLI export

Cluster: LargeRun   Waiting   Cluster ready after last step completed.

| Summary | Application history | Monitoring | Hardware | Events | Steps | Configurations | Bootstrap actions |

Connections: Enable Web Connection – Zeppelin, Spark History Server, Ganglia, Resource Manager ... (View All)

Master public DNS: ec2-34-211-49-97.us-west-2.compute.amazonaws.com   SSH

Tags: --   View All / Edit

Click for Instructions

Command-line Tools

Flintrock

   Open-source command-line tool for launching Apache Spark clusters

   https://github.com/nchammas/flintrock

aws cli

   Amazon's command line tool

   https://aws.amazon.com/cli/

Generating the Command Line

53

AWS CLI export   ✕

```
aws emr create-cluster --termination-protected --applications Name=Hadoop Name=Spark --ec2-attributes
'{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetId":"subnet-0f55196b","EmrManagedSlaveSecurityGroup":"sg-
65bffa1c","EmrManagedMasterSecurityGroup":"sg-62bffa1b"}' --release-label emr-5.22.0 --log-uri 's3n://aws-
logs-834365227482-us-west-2/elasticmapreduce/' --steps '[{"Args":["spark-submit","--deploy-
mode","client","s3://rw-696-flight/pi.py"],"Type":"CUSTOM_JAR","ActionOnFailure":"CONTINUE","Jar":"command-
runner.jar","Properties":"","Name":"Spark application"}]' --instance-groups
'[{"InstanceCount":1,"InstanceGroupType":"MASTER","InstanceType":"m3.xlarge","Name":"Master - 1"},
{"InstanceCount":2,"InstanceGroupType":"CORE","InstanceType":"m3.xlarge","Name":"Core - 2"}]' --auto-scaling-
role EMR_AutoScaling_DefaultRole --ebs-root-volume-size 10 --service-role EMR_DefaultRole --enable-
debugging --name 'CourseSlides' --scale-down-behavior TERMINATE_AT_TASK_COMPLETION --region us-west-2
```

54

Hadoop Ecosystem

Hadoop
  HDFS
  MapReduce
  YARN
Tez
Pig
Hive
Hbase
Sqoop
Oozie
Falcon
Spark
ZooKeeper
Mahout
Phoenix
BigTop
+ others

Apache Pig

Programming Map-Reduce can be low level

Apache Pig - high-level platform for creating programs for Hadoop

Pig Latin

```
input_lines = LOAD '/tmp/my-copy-of-all-pages-on-internet' AS (line:chararray);
words = FOREACH input_lines GENERATE FLATTEN(TOKENIZE(line)) AS word;
filtered_words = FILTER words BY word MATCHES '\\w+';
word_groups = GROUP filtered_words BY word;
word_count = FOREACH word_groups GENERATE COUNT(filtered_words) AS
                    count, group AS word;

ordered_word_count = ORDER word_count BY count DESC;
STORE ordered_word_count INTO '/tmp/number-of-words-on-internet';
```

Apache Hive

SQL is common way to interact with data

Hive provides SQL like query language for HDFS, Amazon S3 data

HiveQL - converted into MapReduce

```
DROP TABLE IF EXISTS docs;
CREATE TABLE docs (line STRING);
LOAD DATA INPATH 'input_file' OVERWRITE INTO TABLE docs;
CREATE TABLE word_counts AS
SELECT word, count(1) AS count FROM
 (SELECT explode(split(line, '\s')) AS word FROM docs) temp
GROUP BY word
ORDER BY word;
```

Apache HBase

BigTable for Hadoop

Non-relational distributed database

Fault-tolerant way of storing large quantites of sparse data

58

Apache Sqoop

People have data in non-hadoop databases

Sqoop
   Transferring data between relational databases & Hadoop

Apache Phoenix

But SQL is common

Phoenix
   Massively parallel relational database for Hadoop
   Uses HBase to store data

Apache Spark

Hadoop has latency issues - reads data from disk

MapReduce is not conducive to solving all problems

Spark

Uses distributed shared memory: Resilient distributed dataset (RDD)

Iterative algorithms

Implemented in Scala

Spark Core

Spark SQL

Dataframes & SQL

Spark Streaming

Spark MLlib

Machine learning

Apache Mahout

Hadoop does not have machine learning libraries

Mahout
  Environment for quickly creating scalable machine learning applications
  Samsara - R-line syntax & environment

Apache Flink, Apache Storm
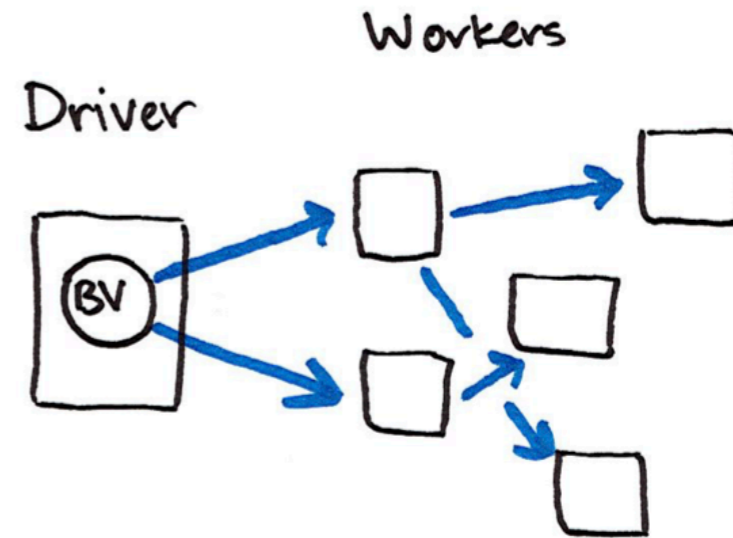
Hadoop does batch jobs

Spark streaming has delays

Fling & Storm

    Each calin to have high throughput and low latency streaming
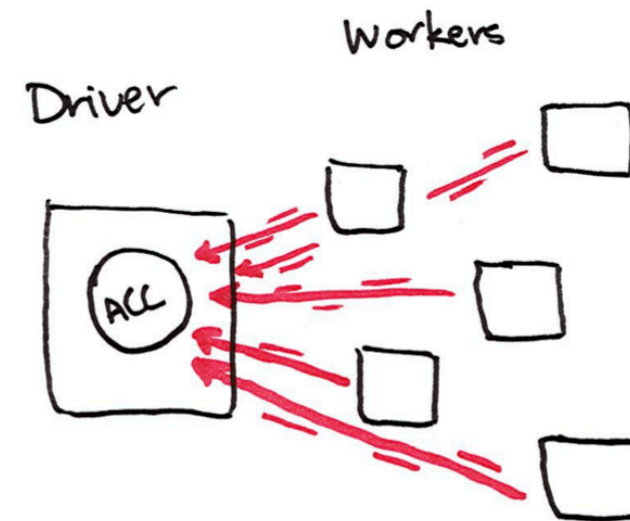
Distributed Variables

Broadcast

    Read-only data shared among workers

Accumulator

    Write only by workers

    Read only on master

Broadcast Example

```
from pyspark.sql import SparkSession

spark = SparkSession\
    .builder\
    .appName("variables")\
    .getOrCreate()

courseSize = 45
courseSizeBroadcast = spark.sparkContext.broadcast(courseSize)

courseSizeBroadcast.value

data = spark.sparkContext.parallelize((1,2,3,4,5,6,7,8), 2)
data.map(lambda x: x + courseSizeBroadcast.value).collect()
```

Using ComplexType

sampleMap = {'a': 10,'bat':1 }

sampleBroadCast = spark.sparkContext.broadcast(sampleMap)

sampleBroadCast.value

```scala
import org.apache.spark.sql.SparkSession
val blockSize =  "4096"
val spark = SparkSession.builder().
    appName("Broadcast Test").
    config("spark.broadcast.blockSize", blockSize).
    getOrCreate()

val sc = spark.sparkContext
val slices =  2
val num =  10000000

val arr1 = (0 until num).toArray

for (i <- 0 until 3) {
    println("Iteration " + i)
    println("===========")
    val startTime = System.nanoTime
    val barr1 = sc.broadcast(arr1)
    val observedSizes = sc.parallelize(1 to 10, slices).map(_ => barr1.value.length)
  observedSizes.collect().foreach(i => println(i))
    println("Iteration %d took %.0f milliseconds".format(i, (System.nanoTime - startTime) / 1E6))
}
```

Accumulator Example

```python
from pyspark.sql import SparkSession

spark = SparkSession\
    .builder\
    .appName("variables")\
    .getOrCreate()


counter = spark.sparkContext.accumulator(0)
def count(item):
    global counter
    print("item: ", item.id)
    counter.add(1)


df = spark.range(16)
smaller = df.coalesce(4)
smaller.foreach(count)
counter.value
```

Output

16

Accumulator
  add()
  value

Numbers only

Can create custom accumulators

Machine Learning in Spark

MLlib

  RDD-based
    org.apache.spark.mllib
    Maintenance mode

  DataFrame based (Spark ML)
    org.apache.spark.ml
    Pipelines
      Inspired by Python scikit-learn

Classification

Regression

Clustering

Collaborative Filtering

Dimension reduction

Linear Algebra

Statistics

http://spark.apache.org/docs/latest/ml-guide.html

SPark 2.4.0   Overview   Programming Guides ▾   API Docs ▾   Deploying ▾   More ▾

MLlib: Main Guide

- Basic statistics
- Data sources
- Pipelines
- Extracting, transforming and selecting features
- Classification and Regression
- Clustering
- Collaborative filtering
- Frequent Pattern Mining
- Model selection and tuning
- Advanced topics

«

Machine Learning Library (MLlib) Guide

MLlib is Spark's machine learning (ML) library. Its goal is to make practical machine learning scalable provides tools such as:

- ML Algorithms: common learning algorithms such as classification, regression, clustering, and co
- Featurization: feature extraction, transformation, dimensionality reduction, and selection
- Pipelines: tools for constructing, evaluating, and tuning ML Pipelines
- Persistence: saving and load algorithms, models, and Pipelines
- Utilities: linear algebra, statistics, data handling, etc.

Announcement: DataFrame-based API is prima

The MLlib RDD-based API is now in maintenance mode.

As of Spark 2.0, the RDD-based APIs in the `spark.mllib` package have entered maintenance mode.

70

Python Examples

$SPARK_INSTALL_DIR/examples/src/main/python/ml

aft_survival_regression.py

als_example.py

binarizer_example.py

bisecting_k_means_example.py

bucketed_random_projection_lsh_example.py

bucketizer_example.py

chi_square_test_example.py

chisq_selector_example.py

correlation_example.py

count_vectorizer_example.py

cross_validator.py

dataframe_example.py

dct_example.py

decision_tree_classification_example.py

decision_tree_regression_example.py

elementwise_product_example.py

estimator_transformer_param_example.py

feature_hasher_example.py

fpgrowth_example.py

gaussian_mixture_example.py

generalized_linear_regression_example.py

gradient_boosted_tree_classifier_example.py

gradient_boosted_tree_regressor_example.py

imputer_example.py

index_to_string_example.py

isotonic_regression_example.py

kmeans_example.py

lda_example.py

linear_regression_with_elastic_net.py

linearsvc.py

logistic_regression_summary_example.py

logistic_regression_with_elastic_net.py

max_abs_scaler_example.py

min_hash_lsh_example.py

min_max_scaler_example.py

multiclass_logistic_regression_with_elastic_

multilayer_perceptron_classification.py

n_gram_example.py

naive_bayes_example.py

normalizer_example.py

one_vs_rest_example.py

onehot_encoder_estimator_example.py

pca_example.py

pipeline_example.py

polynomial_expansion_example.py

prefixspan_example.py

quantile_discretizer_example.py

random_forest_classifier_example.py