DATA ANALYTICS

SPARK OVERVIEW

SPARK ON AWS

AZURE DATABRICKS

SPARK ON GOOGLE

SPARK OVERVIEW

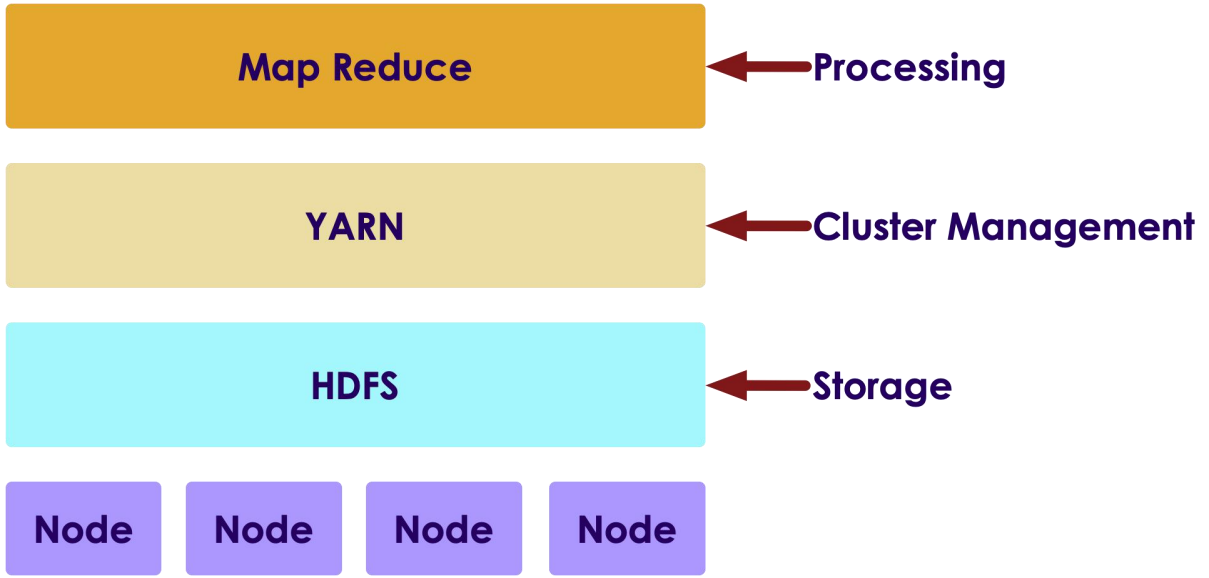
SPARK OVERVIEW
SPARK ON AWS
AZURE DATABRICKS
SPARK ON GOOGLE

BIG DATA V1: HADOOP

Hadoop was the first Big Data platform to be widely adopted

Hadoop has three main components

- Storage: **HDFS** - Store huge amount of data in a distributed fashion
- Operating System: **YARN** - manage the cluster
- Processing: **MapReduce Engine** - distributed computing



MAPREDUCE ENGINE

MapReduce was state of the art around 2008

It was written for a time when

- Data was on disk
- And most processing was batch

How ever MR had its limitations

- It had high overhead
- It didn't support 'in-memory' processing
- It couldn't do 'streaming / real time' work loads

SPARK

Spark is an open Source distributed computing engine 

- Very fast: On-disk ops are **10x** faster than MR
- In-memory ops **100x** faster than MR

General purpose: MR, SQL, streaming, machine learning, analytics

Hadoop compatible: Runs over Hadoop, Mesos, Yarn, or standalone

Plays nicely with Big Data ecosystem (S3, Cassandra, HBase)

Very easy to use API

"Spark is the First Big Data platform to integrate batch, streaming and interactive computations in a unified framework." - stratio.com

WHY IS SPARK POPULAR?

Ease of use

- Easy to get up and running
- Develop on laptop, deploy on cluster

Multiple language support

- Java, Scala, Python and R
- Developers (Java/Scala), Data Scientists (Python, R)

High performant

Plays nice with BigData eco system

Out of the box functionality

- Modern functional programming constructs
- Machine Learning / Streaming / Graph processing

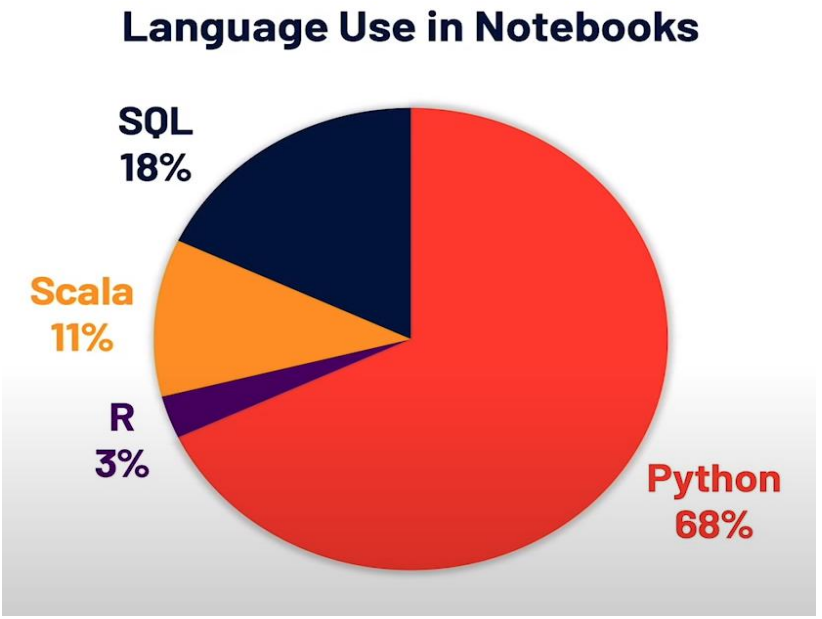
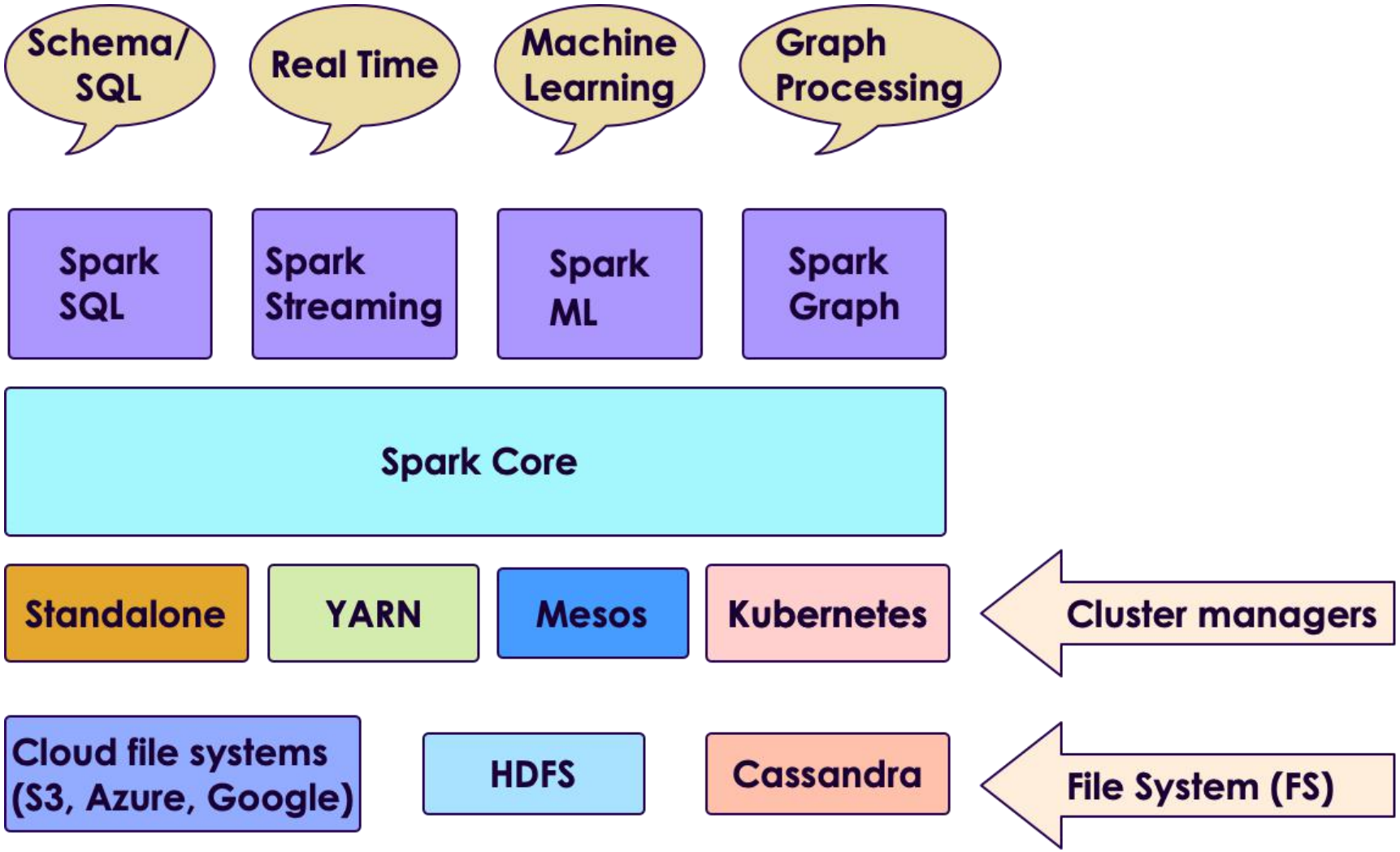


Image source and reference

SPARK COMPONENTS



SPARK USE CASES

Netflix

- Recommendations using Spark + Cassandra
- Analyzes streaming events (450 billion events per day)
- Personalization through recommendations
- Sources: 1 , 2

Starbucks

- 30,000+ stores generate Petabyte scale data
- 1000+ data pipelines in Spark
- Large scale machine learning using Spark
- Stack: Azure cloud + Spark + Delta Lake
- Source

More case studies @ BigDataUseCases.Info

SPARK USE CASES

TERALYTICS

Teralytics

- Processing cell phone events
- 180 billion events per day
- Spark + HDFS
- Estimating usage patterns to enhance coverage (sporting events, commuting, etc.)
- Source: 1 , 2

YAHOO!

Yahoo

- News personalization
- 120 line Scala program with ML lib replaced 15,000 lines of C++
- Spark took 30 minutes to run on 100 million samples
- Source

MORE SPARK USE CASES

CERN

Genomics

Time series

Checkout customer success strories at Databricks

SPARK AT LARGE SCALE

Tencent (Social network in China)

- 8000 nodes
- 400 TB+ data

Alibaba (largest e-commerce site in China)

- 1 PB scale processing
- Large scale image processing

Streaming @ Jenelia Farm

- 1 TB per hour
- Analyze medical images

SPARK AND HADOOP TIMELINE

Hadoop	Year	Spark
Created	2006	
	2009	Starts at AMP lab
	2010	Open sourced
Version 1	2011	
Version 2	2013	
	2014	Version 1, Apache top level project
	2016	Version 2
Version 3	2019	
	2020	Version 3

SPARK VS. MAPREDUCE

Map Reduce

```
@Override
public int run(String[] args) throws Exception {
    System.out.println("Running WordCount");
    Job job = new Job(getConf());
    job.setJarByClass(WordCount.class);
    job.setJobName("WordCount");

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    job.setMapperClass(Map.class);
    job.setCombinerClass(Reduce.class);
    job.setReducerClass(Reduce.class);

    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    System.out.println("Input path: " + args[0]);
    System.out.println("Output path: " + args[1]);
    FileInputFormat.setInputPaths(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    boolean success = job.waitForCompletion(true);
    return success ? 0 : 1;
}

public static void main(String[] args) throws Exception {
    int ret = ToolRunner.run(new WordCount(), args);
    if (ret != 0) {
        System.exit(ret);
    }
}
```

```
public static class Map
        extends Mapper<LongWritable, Text, Text, IntWritable> {

    private static IntWritable ONE = new IntWritable(1);

    @Override
    public void map(LongWritable key, Text value, Context context)
            throws IOException, InterruptedException {
        String line = value.toString();
        String[] words = line.split("\\W");
        for (String word : words) {
            if (word.trim().length() > 0) {
                Text text = new Text();
                text.set(word);
                context.write(text, ONE);
            }
        }
    }
}
public static class Reduce
        extends Reducer<Text, IntWritable, Text, IntWritable> {

    @Override
    public void reduce(
            Text key, Iterable<IntWritable> values, Context cont
            throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        context.write(key, new IntWritable(sum));
    }
}
```

Spark

```
val wordcount = r.flatMap(lines =>
lines.split(" ")).map(word => (word,
1)).reduceByKey(_+_)
```

SPARK VS. MAPREDUCE

Spark is easier to use than MapReduce

Friendlier development environment

- Interactive shells allow faster development
- Web based UI notebooks allow easier development

Multiple language support: Java, Python, Scala, R

Spark is high performant than MR

SPARK VS. MAPREDUCE BENCHMARK

Daytona Grey Benchmark: Sort 100TB of data

References:

- Databricks blog
- http://sortbenchmark.org/

	Hadoop MR Record	Spark Record	Spark 1 PB
Data Size	102.5 TB	100 TB	1000 TB
Elapsed Time	72 mins	23 mins	234 mins
# Nodes	2100	206	190
# Cores	50400 physical	6592 virtualized	6080 virtualized
Cluster disk throughput	3150 GB/s (est.)	618 GB/s	570 GB/s
Sort Benchmark Daytona Rules	Yes	Yes	No
Network	dedicated data center, 10Gbps	virtualized (EC2) 10Gbps network	virtualized (EC2) 10Gbps network
Sort rate	**1.42 TB/min**	**4.27 TB/min**	**4.27 TB/min**
Sort rate/node	**0.67 GB/min**	**20.7 GB/min**	**22.5 GB/min**

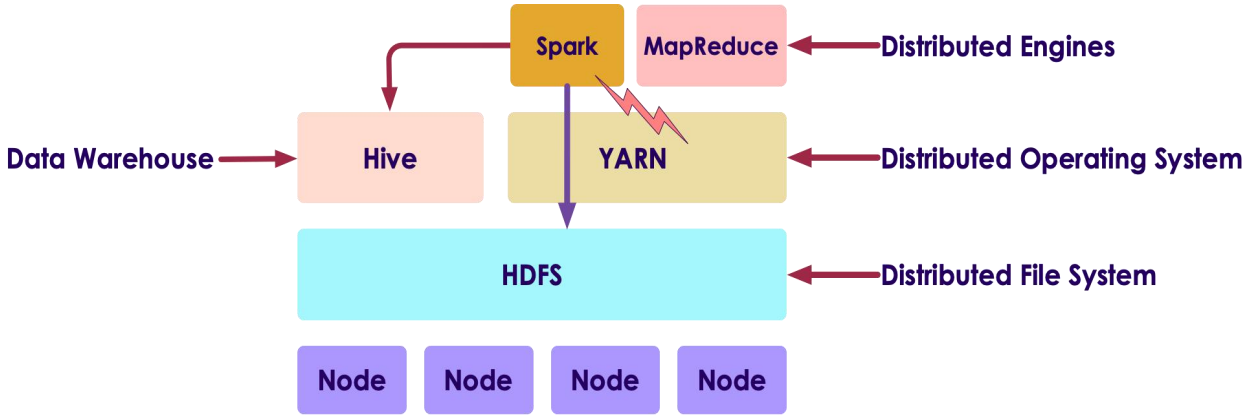
SPARK AND HADOOP

Hadoop is a Data Platform comprised of:

- HDFS: File system
- YARN: Cluster manager
- Hive: Data warehouse
- Engines: MapReduce, Spark

Spark and Hadoop work well together

- Spark can utilize HDFS distributed data



SPARK RUNTIMES

On-Premise

- Spark is part of most modern Hadoop distributions
- Spark can also be downloaded and installed as a standalone system

Hosted solutions

- Databricks cloud - hosted Spark platform
- Cloud vendors: Amazon, Azure, Google

DATABRICKS

Founded by Spark's founders

Develops majority of Spark platform and offers commercial support

Also provides hosted Spark platform (Databricks Cloud)

Databricks is recognized as a leading provider for Data Analytics and Machine Learning platform (Source: Gartner report)

DATABRICKS CLOUD

A hosted platform of Spark

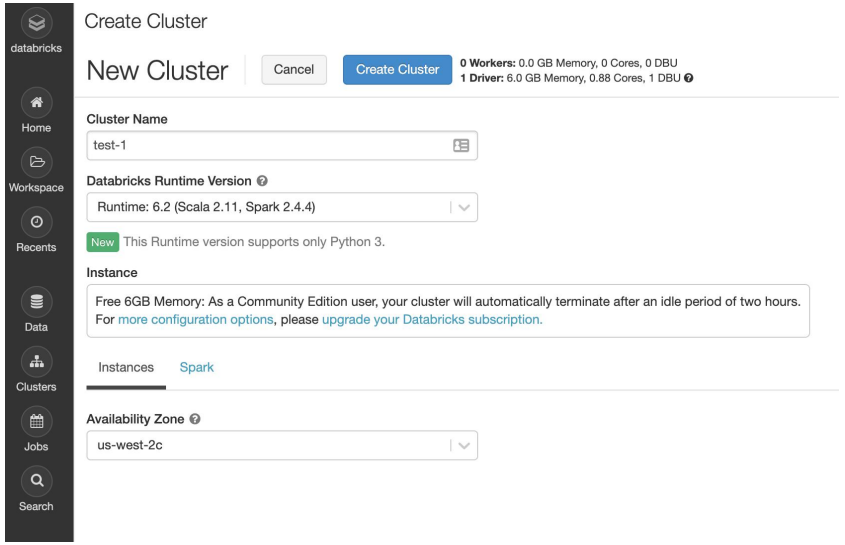
Zero maintenance

Auto scale based on work loads

Community edition is free

- A single node with 6GB memory
- Notebook environment

https://community.cloud.databricks.com/



SPARK IN THE CLOUD

Spark is pretty well supported on all major cloud platforms

Basic idea:

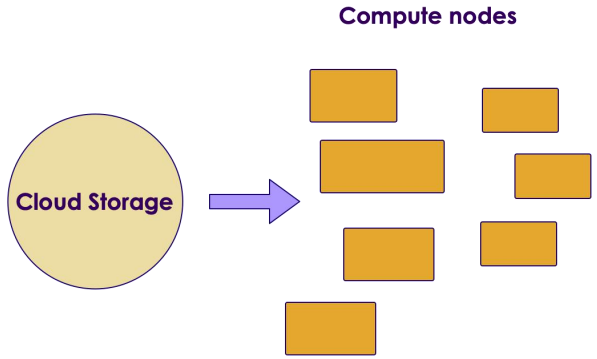
- Upload data into Cloud storage
- Spin up on-demand Spark cluster to process your data
- Shutdown when done
- Pay for use of compute and storage

Amazon offers Elastic Map Reduce (EMR) that includes Spark

Google has DataProc that provisions Spark clusters

Azure has HDInsight * that includes Spark

SPARK SCALING ON THE CLOUD



In Cloud architecture, storage and compute are separate!

Compute nodes stream data from storage (called buckets)

For this to work, compute nodes and storage must have ultra high speed network

Google built the next gen network for their data centers using custom hardware, software, network switches (source)

It can deliver more than 1 Petabit/sec of total bisection bandwidth.

To put this in perspective,

- enough for 100,000 servers to exchange information at 10Gb/s each
- enough to read the entire scanned contents of the Library of Congress in less than 1/10th of a second

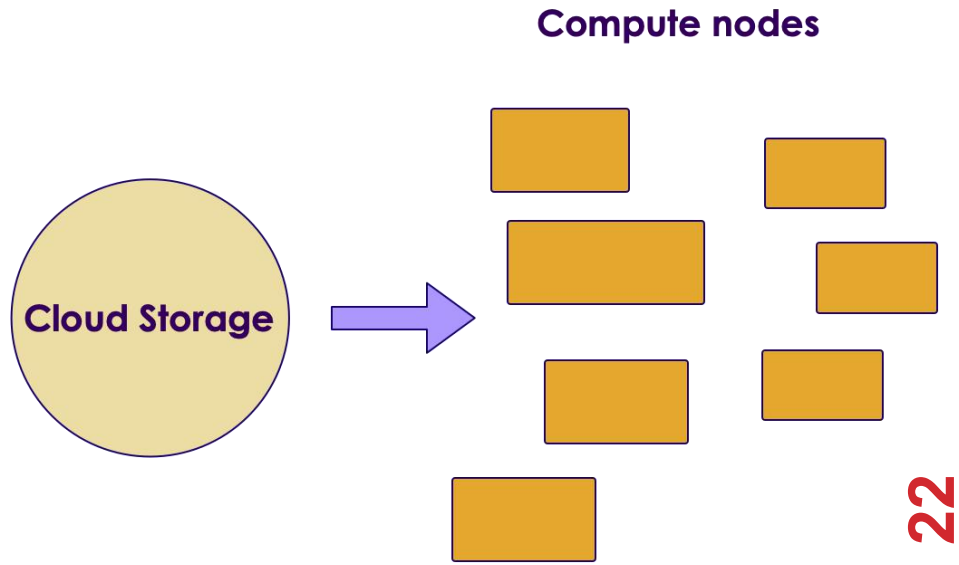
SPARK SCALING ON THE CLOUD

Pros:

- Gives a lot of flexibility on scaling and scheduling computes
- Can dynamically scale compute capacity up/down
- Leverages massive infrastructure the cloud vendors have
- Implemented by cloud vendors / hosted platforms

Cons:

- Not easily implemented on-prem/in-house
- Need to be on a cloud environment
- Costs can add up for storage and compute



SPARK 3

Spark 3 is a big release; 2020 Q3

Performance focused

Over 3400+ patches (alost half of them for Spark SQL

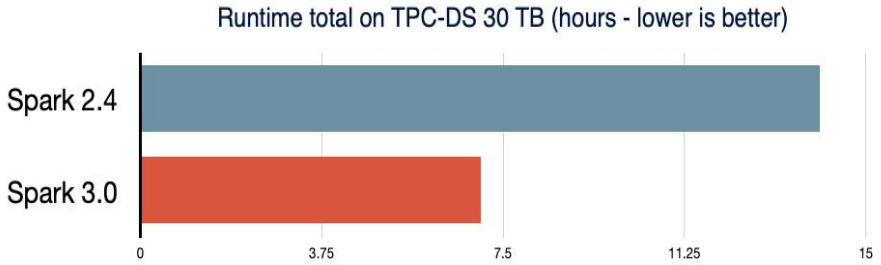
Easy to switch from 2.x

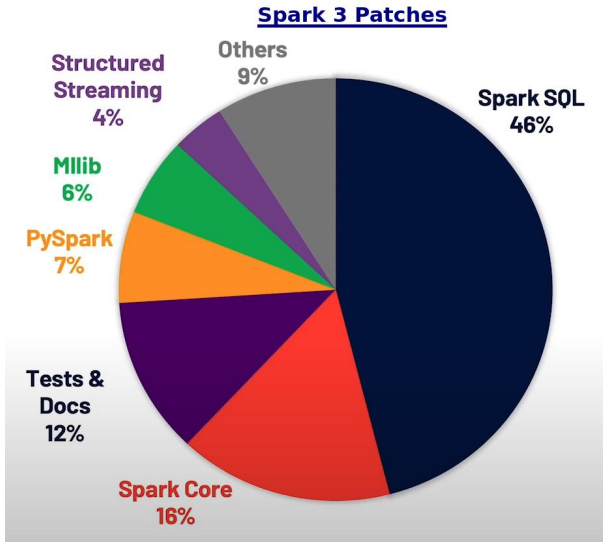
Spark 3 features:

- Delta Lake
- Spark SQL improvements (adaptive query execution)
- Better python performance
- Better Structured Streaming + metrics
- More on these in the next slides

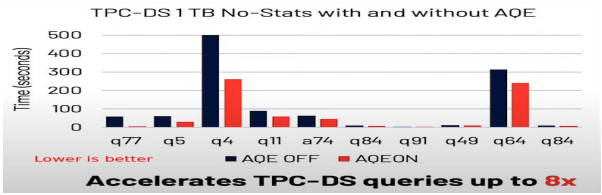
References

- Spark Summit 2020 Keynote - Spark 3
- Introducing Apache Spark 3.0 - blog
- Spark 3.0 Features with Examples





SPARK 3 SQL IMPROVEMENTS



Spark SQL is very widely used

Spark has one of the best SQL engines around

ANSI SQL support improved

Adaptive Query Execution (AQE) :

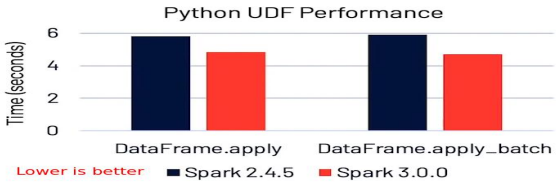
- Can adjust execution plan at runtime (change number of reduces ..etc)
- Can even observe **data skew** and make changes (This is a big deal, as it happens a lot in real life workloads)
- Can do effective joins automatically

Source

SPARK 3 PYTHON IMPROVEMENTS

New APIs for Pandas function

Faster Apache Arrow based calls to Py

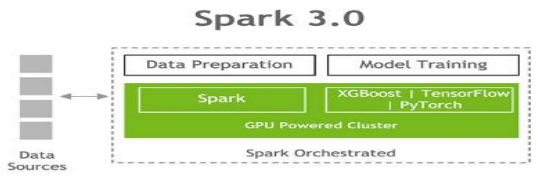


- Apache Arrow is a language-independent columnar memory format, for efficient operations on modern hardware like CPUs and GPUs.
- Also supports zero-copy reads for lightning-fast data access without serialization overhead.

UDFs (User Defined Functions) are easier to write and perform better

Source

SPARK 3 AND GPU



Spark 3 recognizes GPUs as a first-clas CPU and system memory

So Spark can place GPU-accelerated workloads directly onto servers containing the necessary GPU resources

Operations on Dataframes, Spark SQL and Spark ML can utilize GPU

NVIDIA Rapids library enables GPU acceleration for Spark

References:

- NVIDIA page on Spark + GPU
- Get free ebook on Spark + GPU

SPARK ECOSYSTEM PROJECTS

Koalas : Pandas API over Spark

Delta Lake - Reliable, transactional table storage for Big Data

Scikit Learn on Spark Run ML algorithms from Scikit Learn library on Spark

Spark Rapids - GPU acceleration

Data-fu Spark - A good collection of UDFs for Spark

MLFLow - Manage machine learning lifecycle

More

Koalas

DELTA LAKE

Delta Lake is an implementation of modern Data Lake

Features:

- Fully atomic operations
- Transactions are supported
- Scalable to massive amount of data

For more details see Delta-Lake section

APACHE SPARK ON AMAZON EMR

"Amazon EMR is the best place to run Apache Spark."

You can quickly and easily create managed Spark clusters from the

- AWS Management Console
- AWS CLI, or the
- Amazon EMR API
- **Let's do a demo!**

FEATURES AND BENEFITS

Fast performance

Apache Spark natively supports Java, Scala, SQL, and Python

Create diverse workflows

INTEGRATION WITH AMAZON EMR FEATURE SET

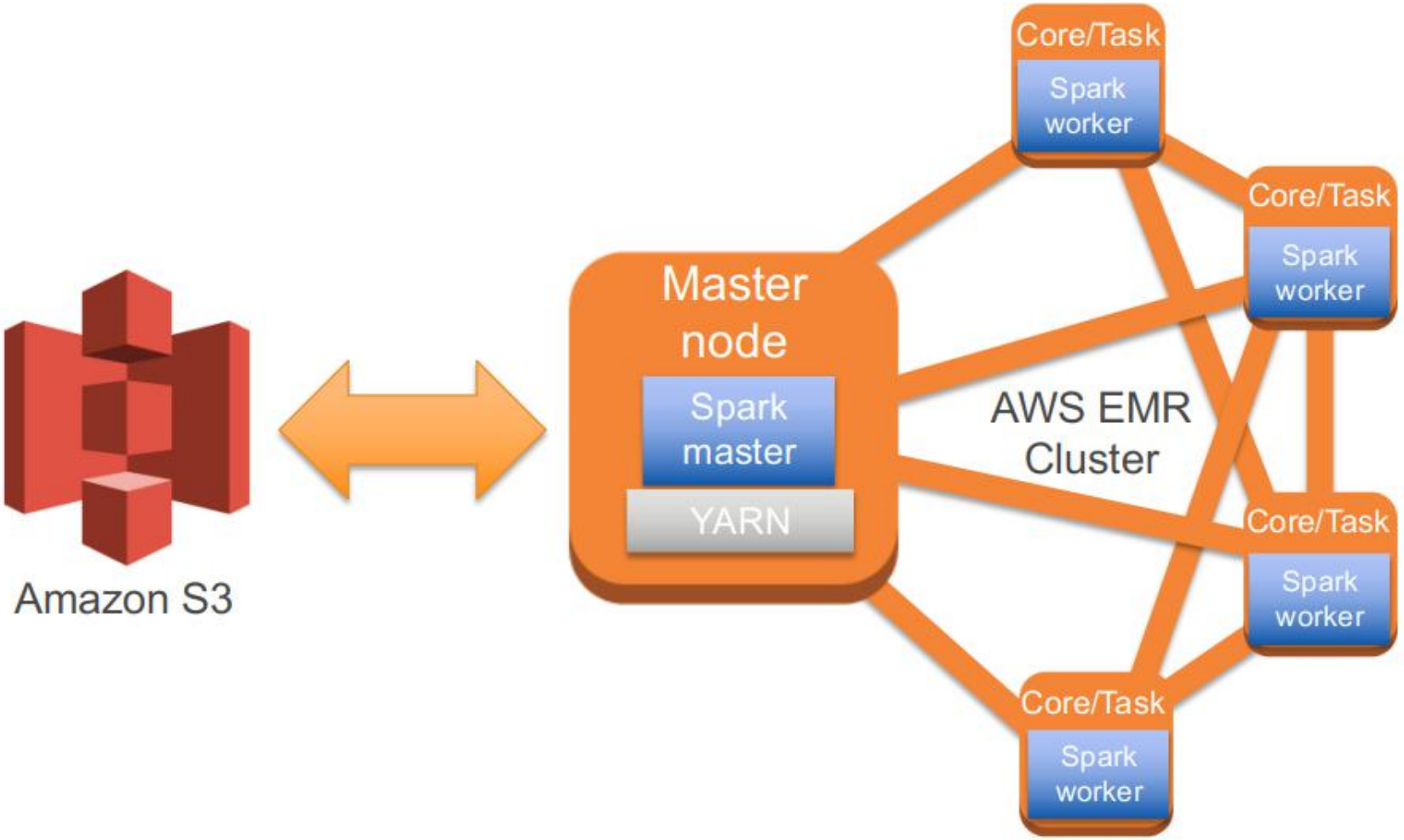
Submit Apache Spark jobs with the EMR Step API,

use Spark with EMRFS to directly access data in S3,

save costs using EC2 Spot capacity,

use EMR Managed Scaling to dynamically add and remove capacity

SPARK ON AMAZON EMR



BENEFITS OF SPARK AND SPARK SQL ON EMR

Ease of use

- Spark can be installed at launch with Amazon EMR AMI v3.8+, and Amazon EMR Release 4.0+
- Deploy small and large Spark clusters in minutes

Amazon EMR container management

- Node recovery in case of failures
- Automatic log collection in Amazon S3 for analysis, debugging

SPARK ON AMAZON EMR COMPARED ON ON-PREM

Low cost

- Run clusters inexpensively
- Increase memory, CPU capacity cheaply by adding task nodes using spot instances
- Run Spark on Amazon EMR at no additional charge

AWS service integration

- Create RDDs and DataFrames directly from and save them directly to Amazon S3
- Use CloudWatch, Ganglia to monitor cluster

Amazon EMRFS integration

- Directly access data in and push logs to Amazon S3

SPARK METRICS AND CLOUDWATCH

Monitor Spark metrics with CloudWatch

Setup CloudWatch alarms and get notified if CPU, memory metrics reached your threshold

- Receive notification via email, SNS, HTTP API call

Examples:

- Monitor memory usage with JvmHeapUsed metric
- Monitor load using Amazon EMR TotalLoad metric

Take manual or automated actions

- Manually add task nodes to increase capacity

QUIZ

When you deploy a cluster with Spark and Spark SQL, the Spark framework replaces the MapReduce framework.

- A. True
- B. False

QUIZ

Name two benefits of running Spark and Spark SQL on Amazon EMR.

AZURE DATABRICKS

SPARK OVERVIEW
SPARK ON AWS
AZURE DATABRICKS
SPARK ON GOOGLE

AZURE DATABRICKS

Fully-managed,

cloud-based Big Data and Machine Learning platform

Databricks, an end-to-end, managed Apache Spark platform

optimized for the cloud

OPTIMIZED ENVIRONMENT

- High-speed connectors to Azure storage services, such as Azure Blob Store and Azure Data Lake
- Auto-scaling and auto-termination of Spark clusters to minimize costs
- Caching
- Indexing
- Advanced query optimization

WHO IS DATABRICKS WITH MS?

Databricks was founded by the creators of Apache Spark, Delta Lake, and MLflow.

Over 2000 global companies use the Databricks platform across big data & machine learning lifecycle.

Databricks Vision: Accelerate innovation by unifying data science, data engineering and business.

Databricks Solution: Big Data Analytics Platform

DATABRICKS PARTS NOT OPEN-SOURCE?

Databricks Workspace - Interactive Data Science & Collaboration

Databricks Workflows - Production Jobs & Workflow Automation

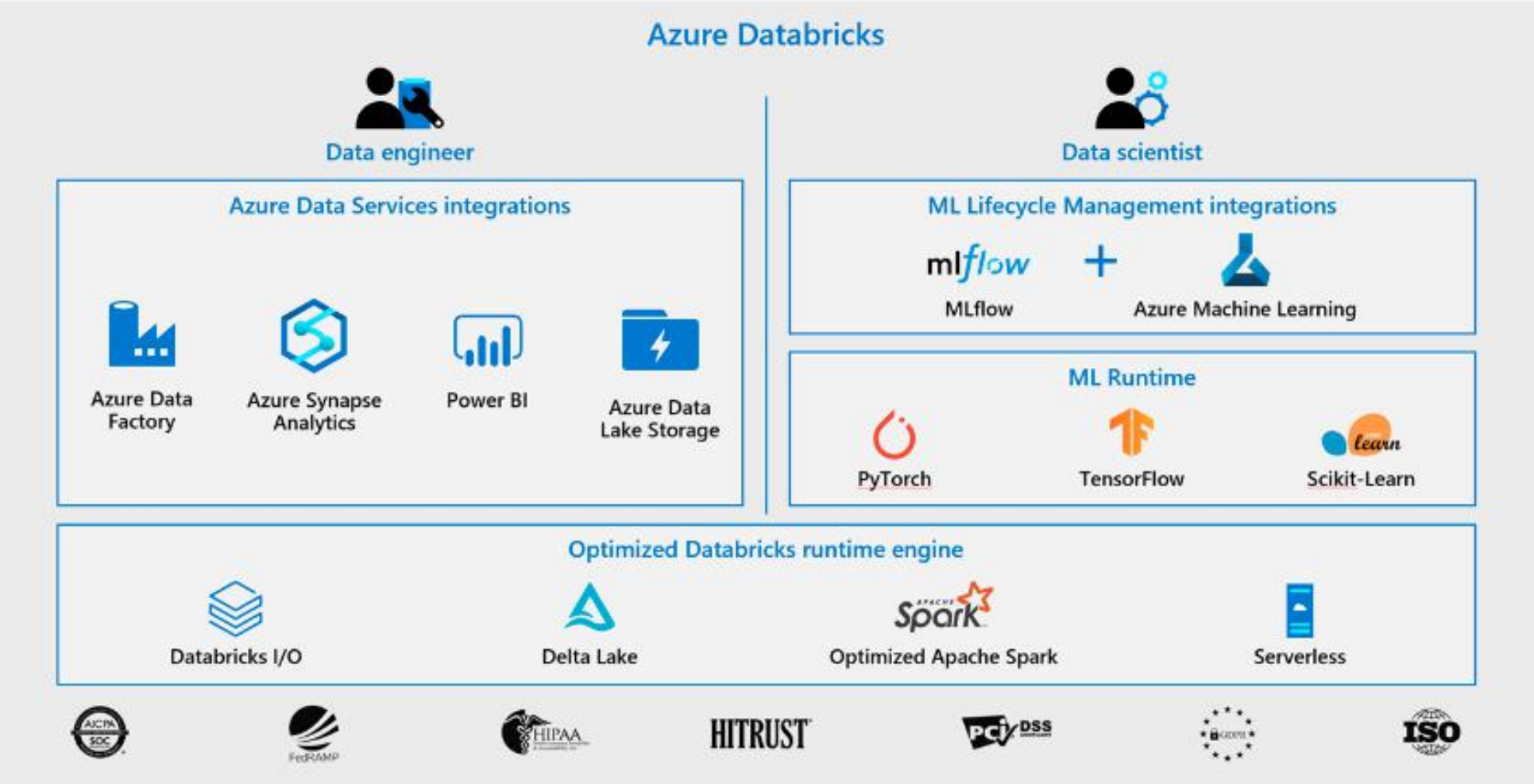
Databricks Runtime

Databricks I/O (DBIO) - Optimized Data Access Layer

Databricks Serverless - Fully Managed Auto-Tuning Platform

Databricks Enterprise Security (DBES) - End-To-End Security & Compliance

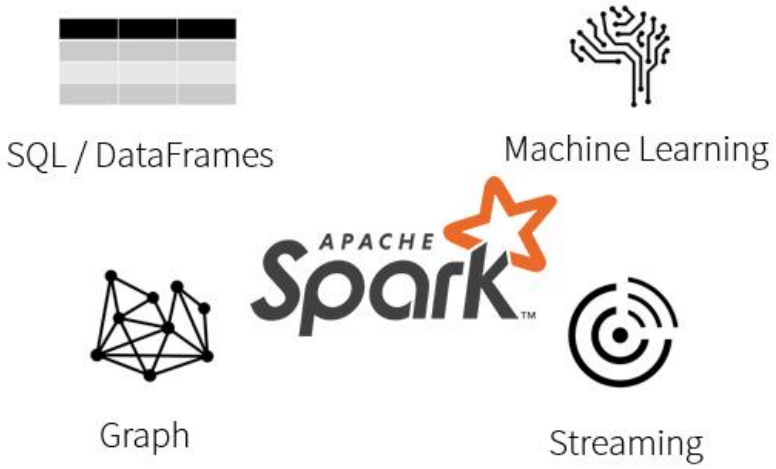
AZURE DATABRICKS



SPARK ON AZURE

Azure Databricks

- Fully-managed version of the open-source Apache Spark analytics and data processing engine. Azure Databricks is an enterprise-grade and secure cloud-based big data and machine learning platform.
- Databricks provides a notebook-oriented Apache Spark as-a-service workspace environment, making it easy to manage clusters and explore data interactively



AZURE SPARK OPTIMIZATIONS

- High-speed connectors to Azure storage services, such as Azure Blob Store and Azure Data Lake
- Auto-scaling and auto-termination of Spark clusters to minimize costs
- Caching
- Indexing
- Advanced query optimization

QUIZ

How many drivers does a Cluster have?

- A. A Cluster has one and only one driver.
- B. Two, running in parallel
- C. Configurable between one and eight

QUIZ

Spark is a distributed computing environment. Therefore, work is parallelized across executors. At which two levels does this parallelization occur?

- A. The executor and the slot
- B. The Driver and the Executor
- C. The slot and the task

QUIZ

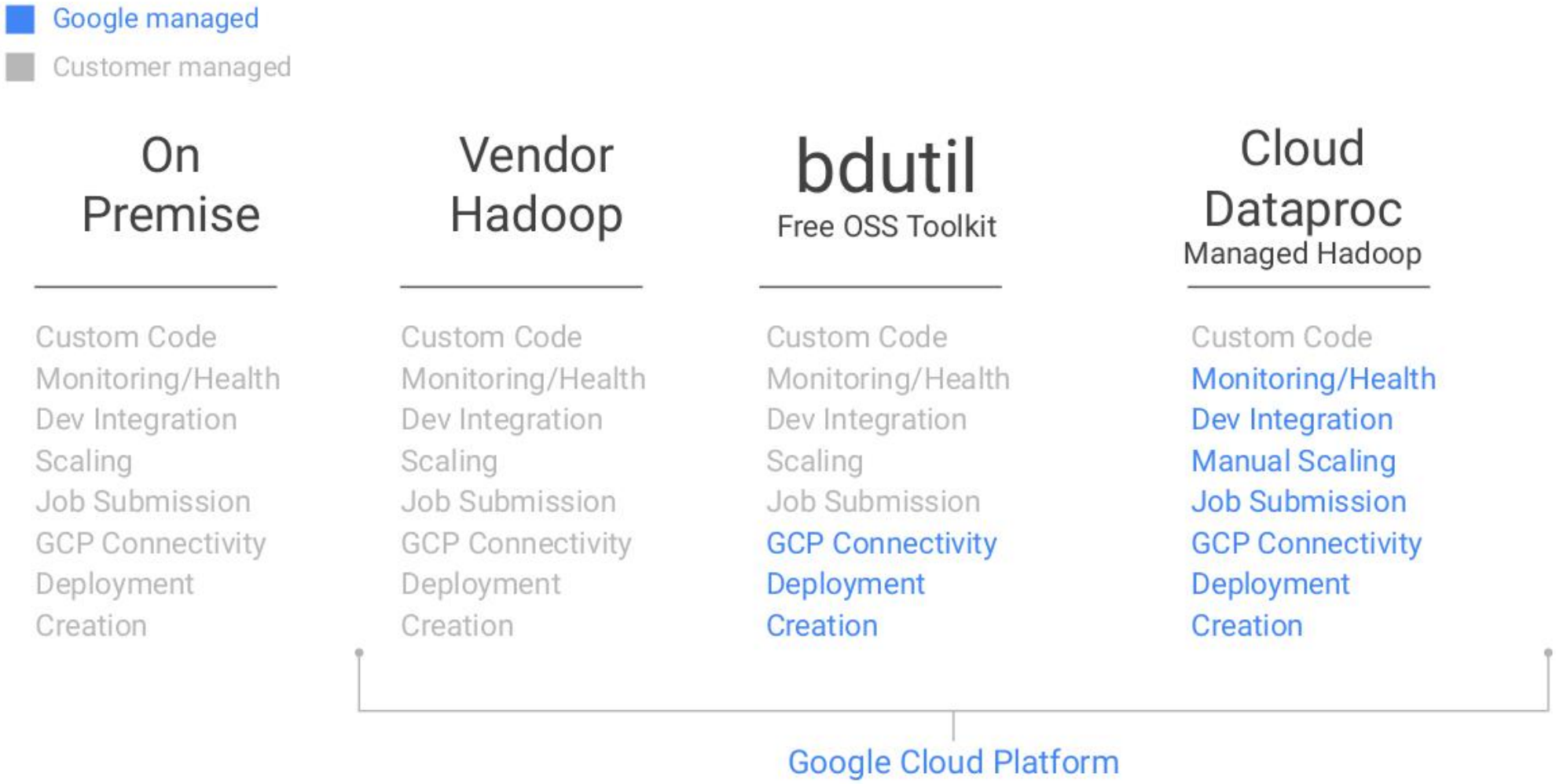
What type of process are the driver and the executors?

- A. Java processes
- B. Python processes
- C. C++ processes

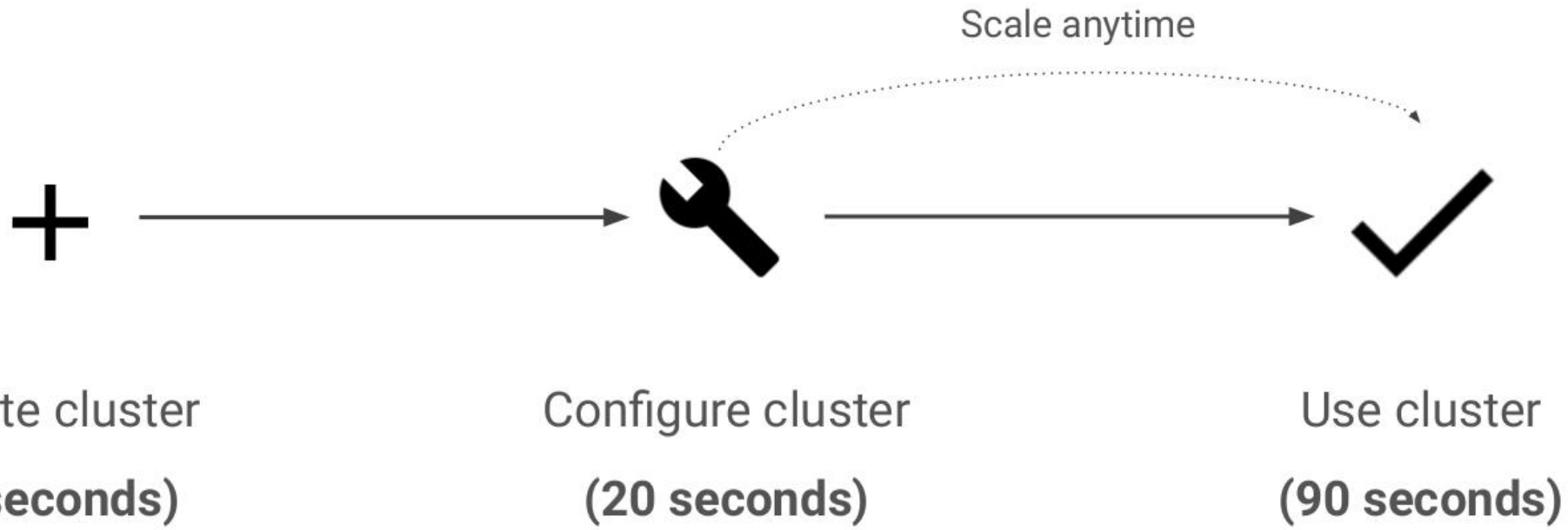
SPARK ON GOOGLE

SPARK OVERVIEW
SPARK ON AWS
AZURE DATABRICKS
SPARK ON GOOGLE

GCP DATAPROC EASES HADOOP MANAGEMENT



TYPICAL DATAPROC DEPLOYMENTS INVOLVE...



DATAPROC RUNS OPEN SOURCE TOOLS

- Stateless clusters in <90 seconds
- Supports Hadoop, Spark, Pig, Hive, etc.
- High-level APIs for job submission
- Connectors to Bigtable, BigQuery, Cloud Storage

CREATE A CLUSTER FROM THE CONSOLE

GIVE THE CLUSTER A UNIQUE NAME

CHOOSE SOMETHING YOU WILL REMEMBER

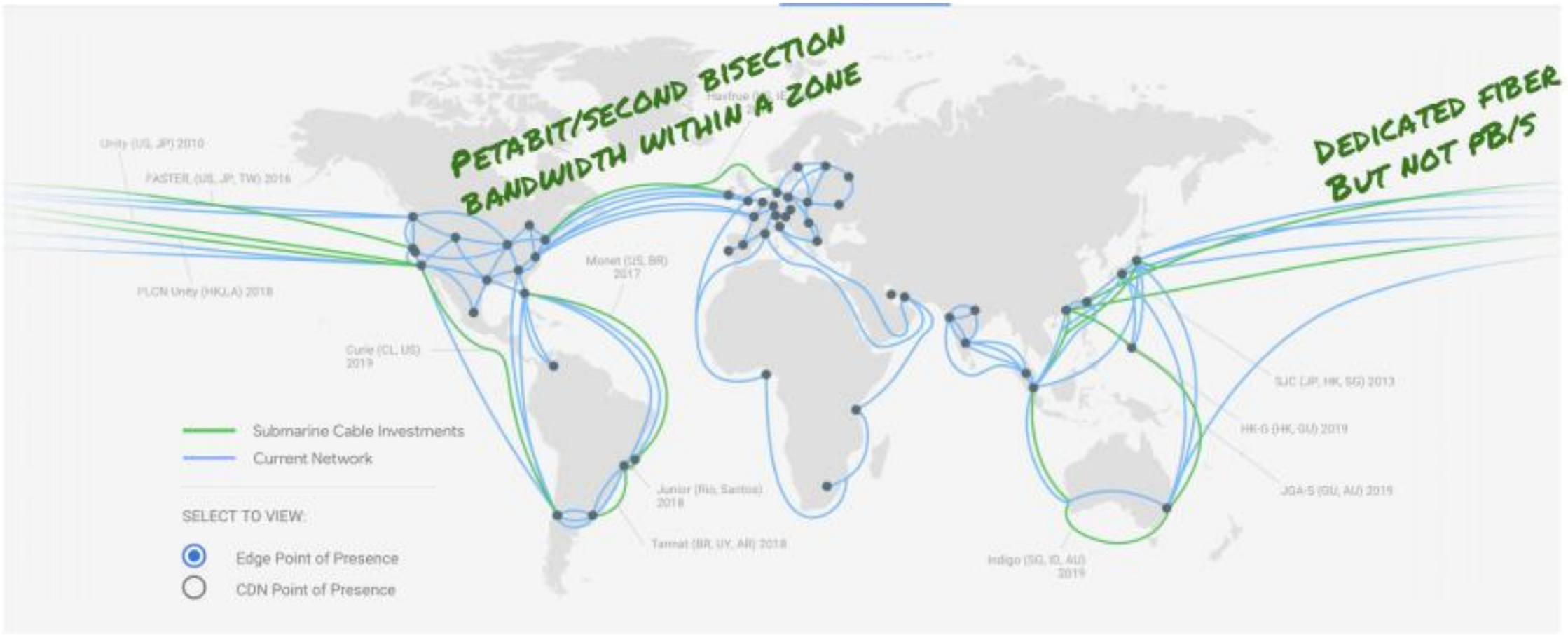
ONE CLUSTER PER JOB

CHOOSE SOMETHING YOU WILL REMEMBER, SUCH AS WHAT YOU ARE GOING TO USE THE CLUSTER FOR

THE ZONE IS VERY, VERY IMPORTANT

AZ IS WHERE THE COMPUTE NODES WILL LIVE

MATCH DATA WITH COMPUTE (SAME REGION)



THREE CLUSTER CONFIGURATIONS POSSIBLE

THE MASTER NODE MANAGES THE CLUSTER
CHOOSE BETWEEN:

1. SINGLE NODE (FOR EXPERIMENTATION)
2. STANDARD (1 MASTER ONLY)
3. HIGH AVAILABILITY (3 MASTERS)

HDFS IS AVAILABLE, BUT DON'T USE IT

MACHINE TYPE, NUMBER OF WORKERS

DISK PERFORMANCE SCALES WITH SIZE!!!
DON'T USE HDFS TO STORE INPUT/OUTPUT DATA

YOU CAN CUSTOMIZE THE DATAPROC CLUSTER

CAN SET UP FIREWALL RULES ETC.

CAN ALSO INSTALL CUSTOM SOFTWARE ON THE DATAPROC WORKERS AND MASTER

MOST THINGS YOU CAN DO FROM WEB CONSOLE

WEB CONSOLE

GCLOUD SDK
COMMAND LINE

```
gcloud dataproc clusters
        --master-machine
        --num-workers 2
        --worker-boot-di
```

REST API call

Google Cloud Platform

CUSTOM
SOFTWARE ...

CREATING A CLUSTER USING GCLOUD SDK

```
gcloud dataproc clusters create my-second-cluster --zone us-central1-a \
        --master-machine-type n1-standard-1 --master-boot-disk-size 50 \
        --num-workers 2 --worker-machine-type n1-standard-1 \
        --worker-boot-disk-size 50
```

CONTEXT-SPECIFIC HELP

```
gcloud dataproc --help
gcloud dataproc clusters --help
gcloud dataproc clusters create --help
```

CONGRATS ON COMPLETION