GlobalLogic®

A Hitachi Group Company

Highly Scalable Data Processing

Pavol Dudrík
Lead Software Engineer

9.5.2024

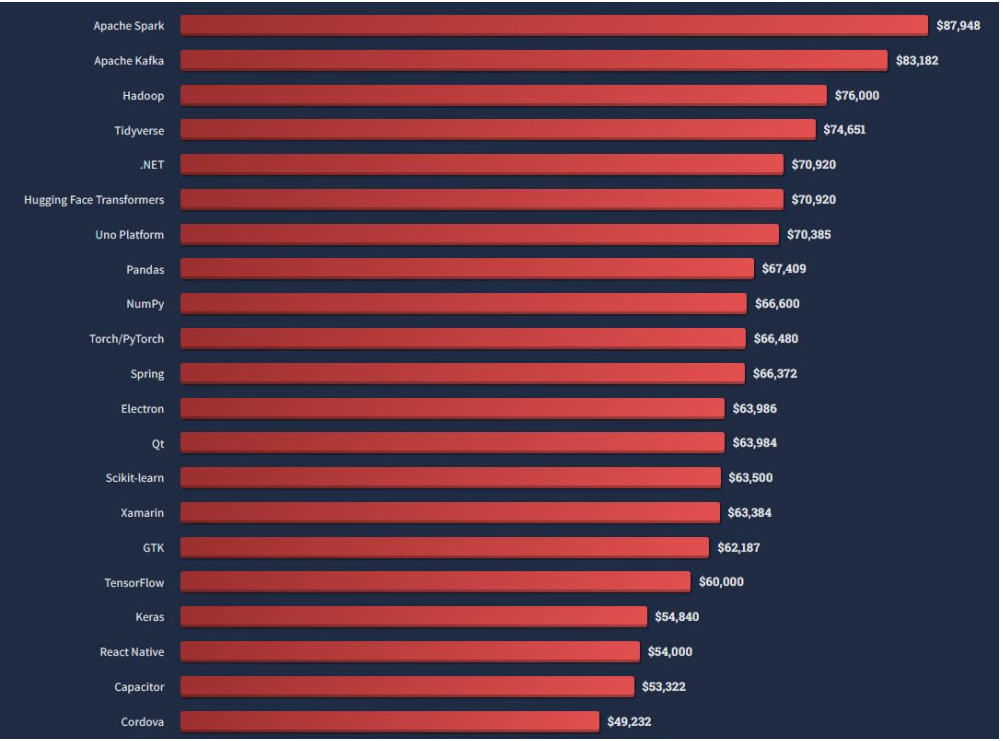
Key Takeaways From the Last Session

- Data integration in cloud environment
- Focusing on our expertise
- Outsourcing
- Azure Data Factory
- Data integrated in a single repository
- Need for processing of the data
- Apache Spark, Databricks

Agenda

1. Motivation
2. Brief History of Large-Scale Data Processing
3. Apache Spark
4. Databricks
5. Q&A

Motivation



- Apache Spark is used by thousands of companies
- 80% of the Fortune 500
- The most widely-used engine for scalable computing
- Databricks is used by more than 7,000 organizations worldwide
- Over 50% of the Fortune 500
- #1 in 2022 Stack Overflow top paying technologies survey in Other Frameworks category

Google's Challenges

1. How to store hundreds of TBs of data with high fault-tolerance?
2. How to process the stored data in distributed manner?

Selling Ball Raffle Tickets

1 ticket roll - 1 cashier

Selling Ball Raffle Tickets

A queue emerges

Selling Ball Raffle Tickets

1 ticket roll - 2 cashiers

Selling Ball Raffle Tickets

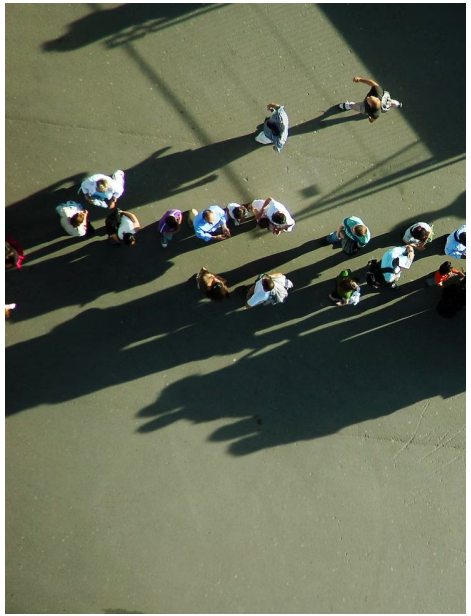
The queue intensifies

Selling Ball Raffle Tickets

2 ticket rolls - 2 cashiers

Selling Ball Raffle Tickets

The queue is halved



Cashiers - compute
Ticket rolls - data

Google's Focus

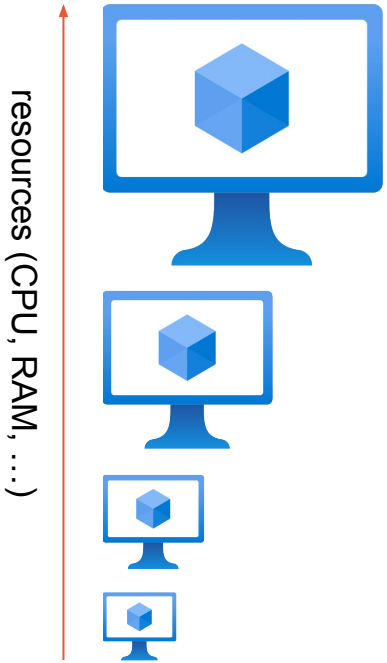
1. Distributed file system
2. Parallel execution model

Base requirement - high scalability

Scalability is the measure of a system's ability to increase or decrease in performance in response to changes in application and system processing demands.

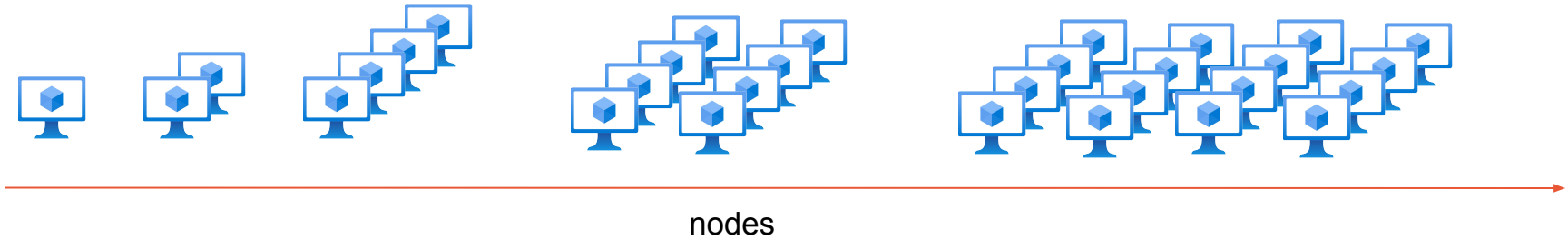
Vertical Scaling

Scaling up



Horizontal Scaling

Scaling out



Scaling - Ball Raffle Ticket Cashier

Horizontal - adding more cashiers

Vertical - motivating cashier to better performance by increasing the salary

salary increase

What Google Came up With

3 Main Components

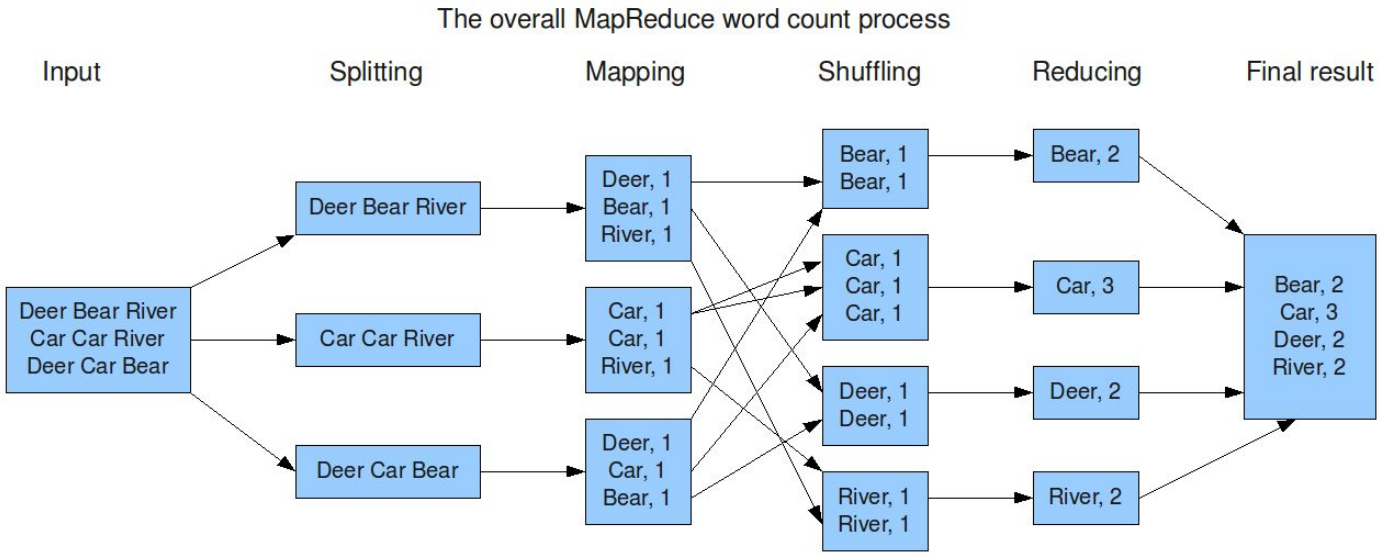
- Google published 3 papers between 2003 and 2006
- Google File System (2003) - distributed, fault-tolerant, scalable file system
- MapReduce (2004) - parallel programming paradigm based on functional programming
- BigTable (2006) - NoSQL database
- Data locality - sending code to where data resides to save I/O cost
- Proprietary work - not shared with public as open-source

Google’s Concepts Inspired the World

Namely Doug Cutting - The Father of Hadoop

- Cutting working at Yahoo on a web crawler (Apache Nutch)
- Desire to use the concepts proposed by Google in his project
- 2005 - Created an open-source version of Google’s components - Hadoop
- Hadoop File System (HDFS)
- Hadoop MapReduce
- Yahoo handed over the projects to The Apache Software Foundation

MapReduce


The overall MapReduce word count process
Input
Splitting
Mapping
Shuffling
Reducing
Final result
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, 1
Bear, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2

MapReduce - Sample Code

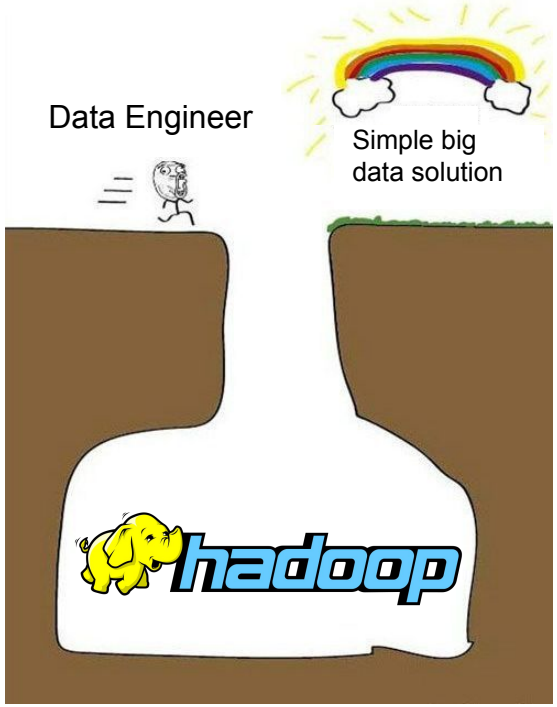
```
public static class TokenizerMapper
     extends Mapper<Object, Text, Text, IntWritable>{

  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(Object key, Text value, Context context
                 ) throws IOException, InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      context.write(word, one);
    }
  }
}

public static class IntSumReducer
     extends Reducer<Text,IntWritable,Text,IntWritable> {
  private IntWritable result = new IntWritable();

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

```
public static void main(String[] args) throws Exception {
  Configuration conf = new Configuration();
  Job job = Job.getInstance(conf, "word count");
  job.setJarByClass(WordCount.class);
  job.setMapperClass(TokenizerMapper.class);
  job.setCombinerClass(IntSumReducer.class);
  job.setReducerClass(IntSumReducer.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(IntWritable.class);
  FileInputFormat.addInputPath(job, new Path(args[0]));
  FileOutputFormat.setOutputPath(job, new Path(args[1]));
  System.exit(job.waitForCompletion(true) ? 0 : 1);
```

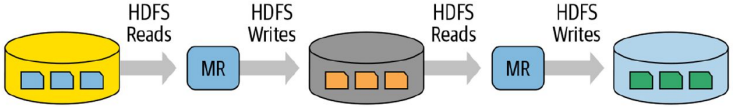
Hadoop - The Origin of the Name

JOKE TIME

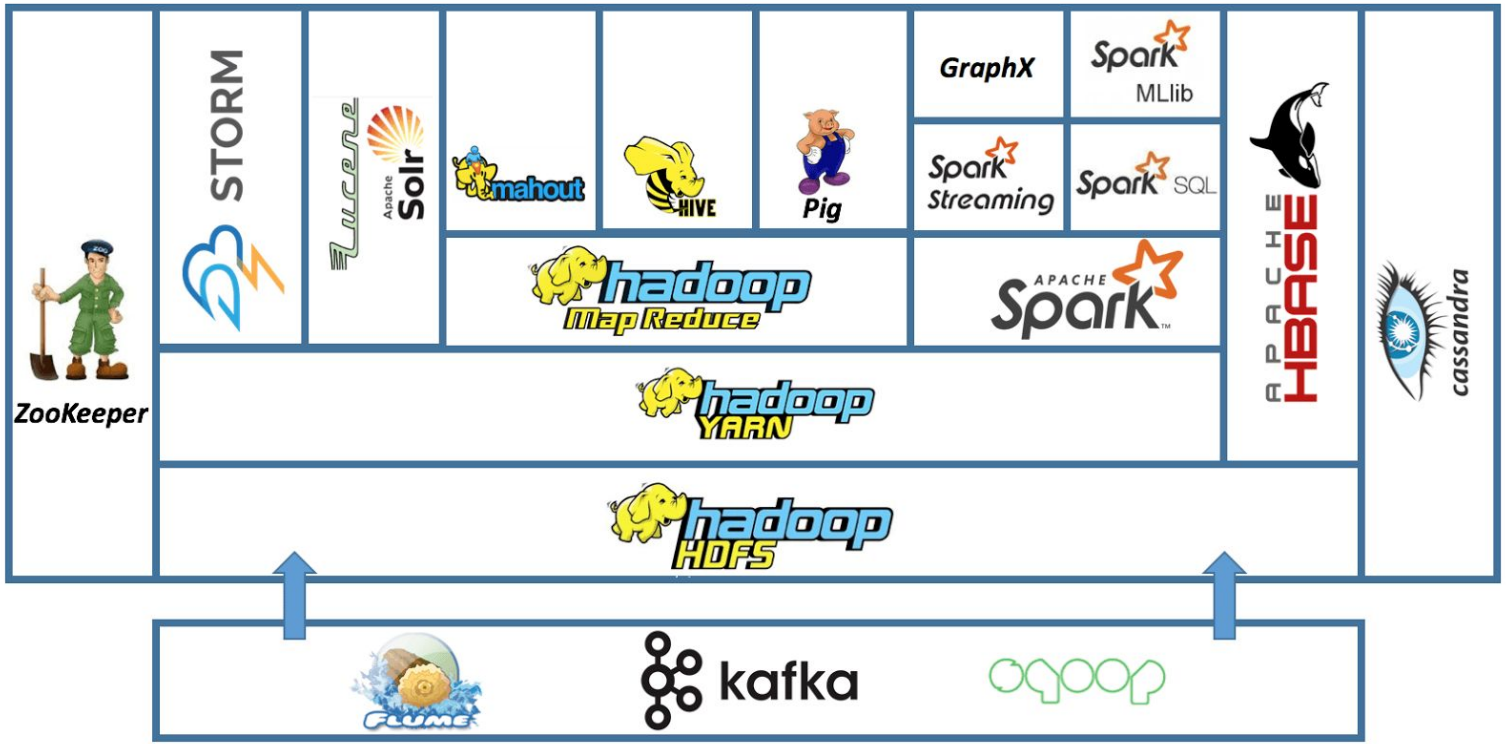
Hadoop - Pitfalls



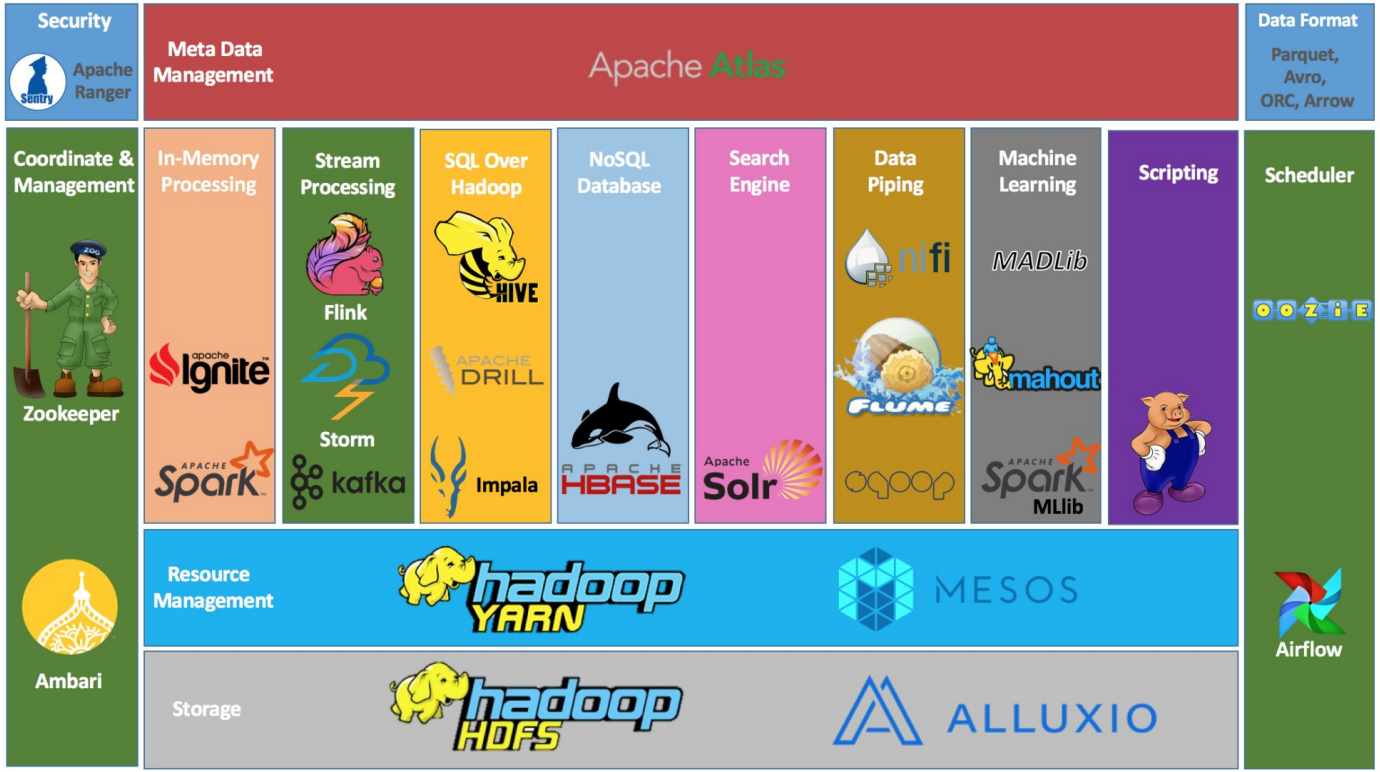
- Difficult to configure and maintain
- Verbose APIs
- I/O overhead stemming from results getting written to disk between jobs
- Machine learning - each run a separate job - performance issues
- Great for batch jobs, but falls short for other workloads (ML, Streaming, interactive queries, …)
- Issues usually solved by adding more components - increasing complexity of the whole system



Hadoop Ecosystem



Hadoop Ecosystem



File Formats

Logical table representation

a	b	c
a1	b1	c1
a2	b2	c2
a3	b3	c3
a4	b4	c4
a5	b5	c5

Row Layout

a1	b1	c1	a2	b2	c2	a3	b3	c3	a4	b4	c4	a5	b5	c5

Column Layout

a1	a2	a3	a4	a5	b1	b2	b3	b4	b5	c1	c2	c3	c4	c5

encoding

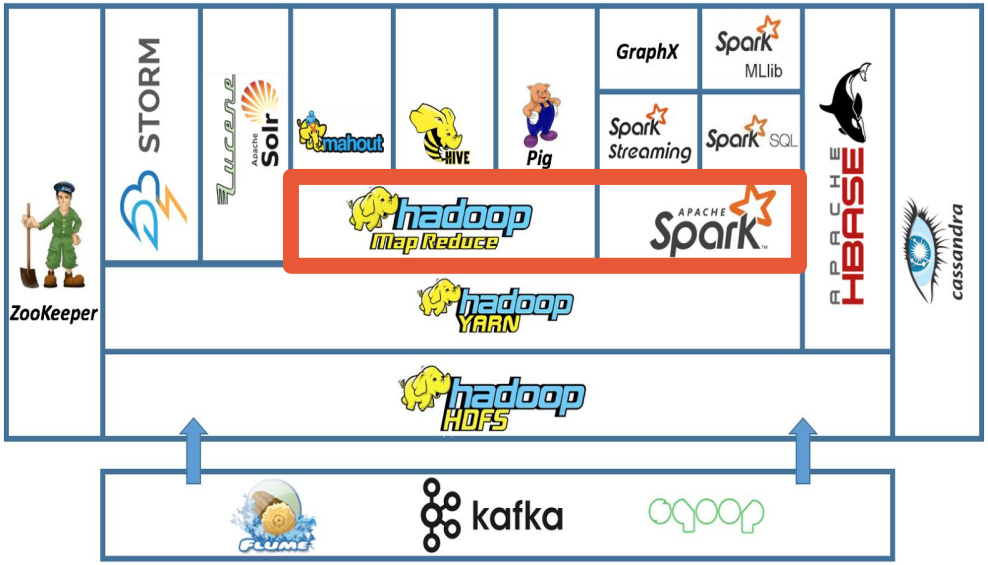
encoded chunk	encoded chunk	encoded chunk

Can Hadoop be improved and simplified?

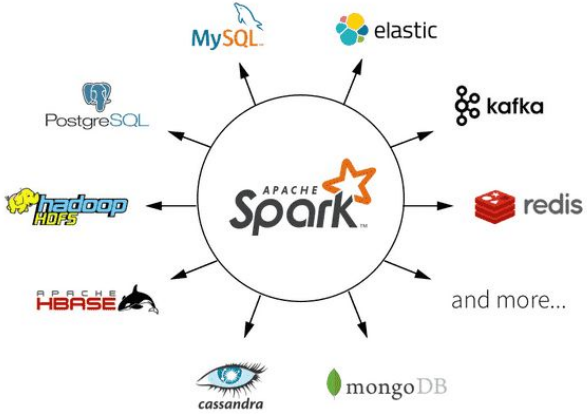
Apache Spark

- Matei Zaharia created (2009) an alternative to MapReduce - Apache Spark
- 10-100x faster than MapReduce in many use cases (in-memory caching)
- Support for ML, Streaming, interactive queries using unified abstraction
- Apache Spark™ is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters
- The most widely-used engine for scalable computing
- Scala, Java, Python, SQL, R
- Wide support by 2013
- v1.0 released in 2014

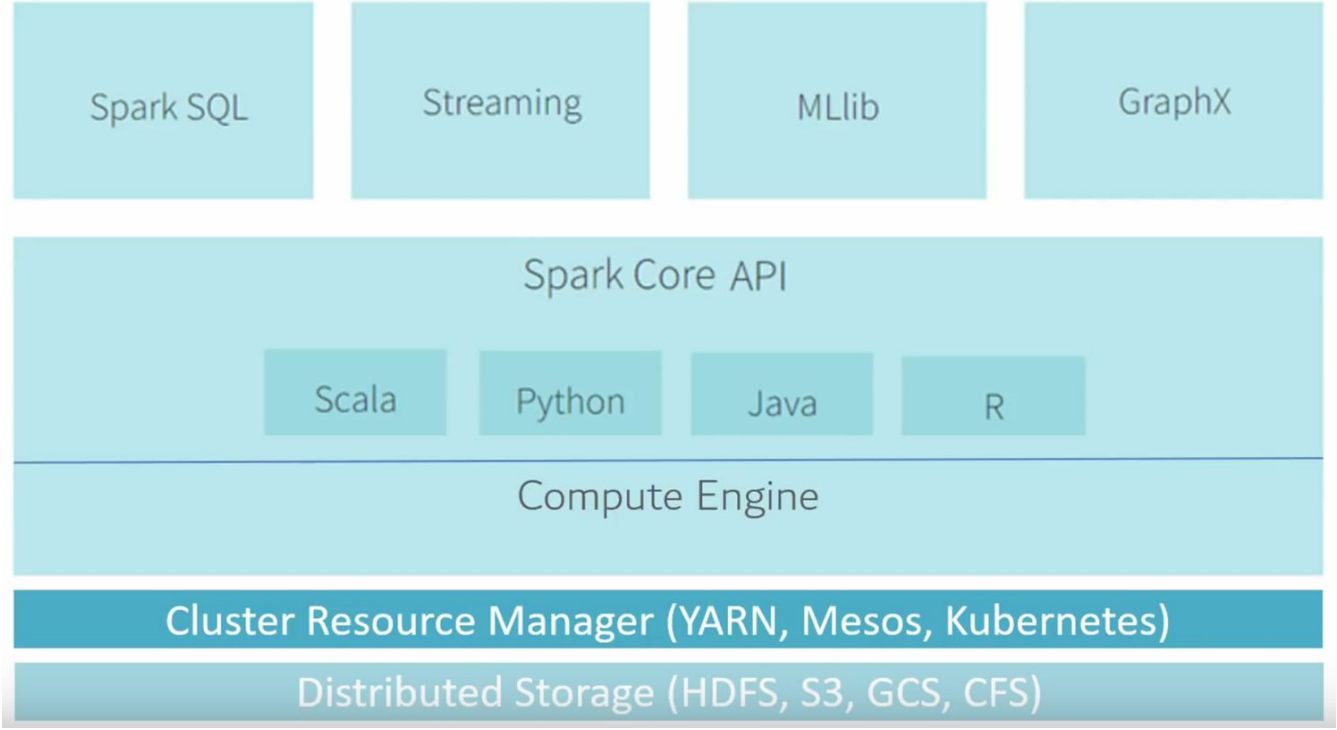
Spark's Role



- Spark is a compute engine - need for a cluster manager and a storage solution
- Lots of connectors out-of-the-box with possibility to create custom connectors
- Possibility to run the same code locally and on a cluster without major changes in behavior



Spark - Architecture

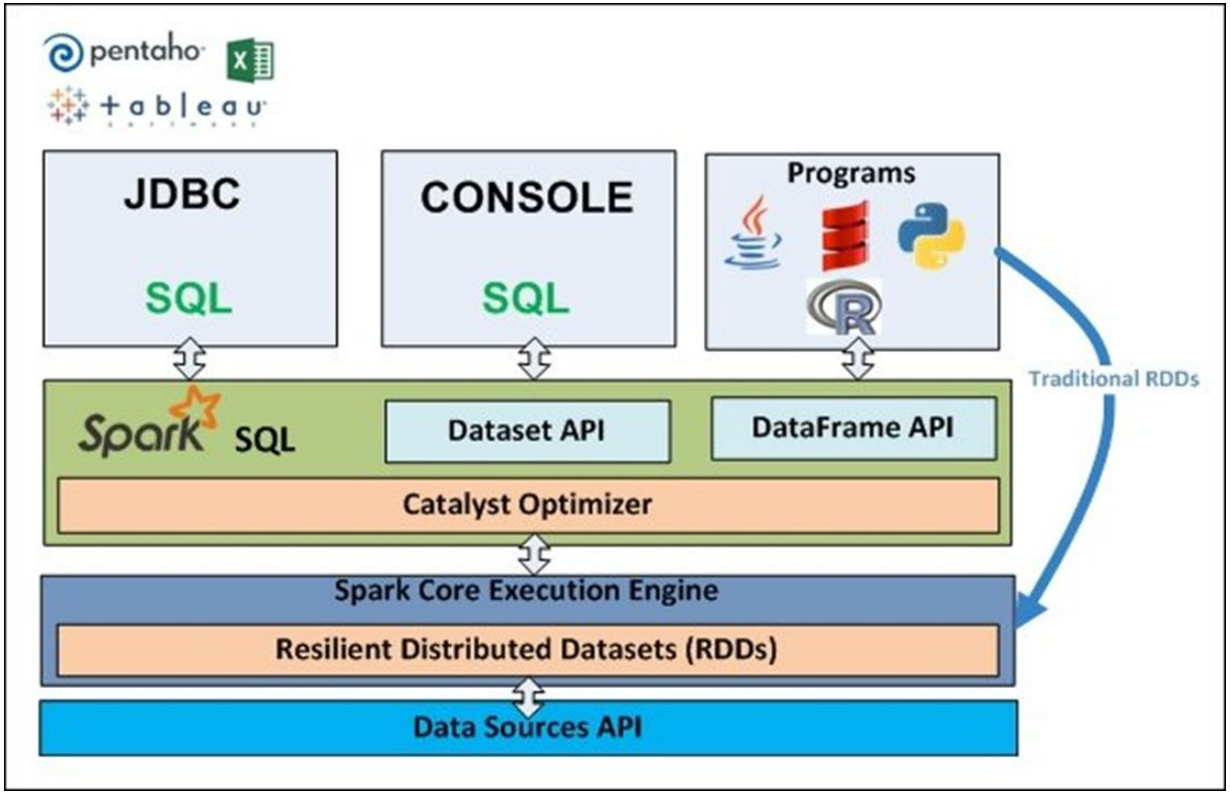


First Steps

```
C:\Users\Pali>spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/04/16 22:03:05 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform...
Spark context Web UI available at http://DESKTOP-OK3CS3U:4040
Spark context available as 'sc' (master = local[*], app id = local-1681675386660).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.3.2
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_362)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

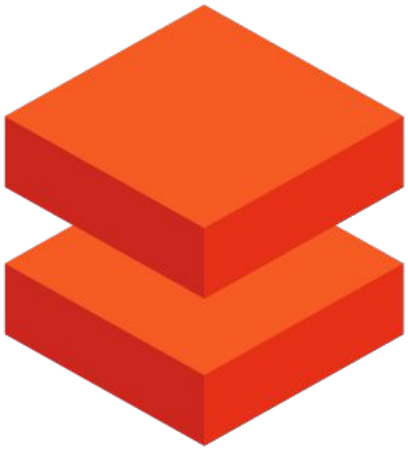
- Install Apache Spark - very simple installation for both Windows and UNIX systems
- Needs JDK 8 installed
- Set up environment variables
- Various options for working with Spark
- spark-shell (pre-created SparkSession)
- Working in an IDE (maven, sbt, …)
- spark-submit

Spark - Architecture



How can we get rid of the need to manage storage and a cluster manager?

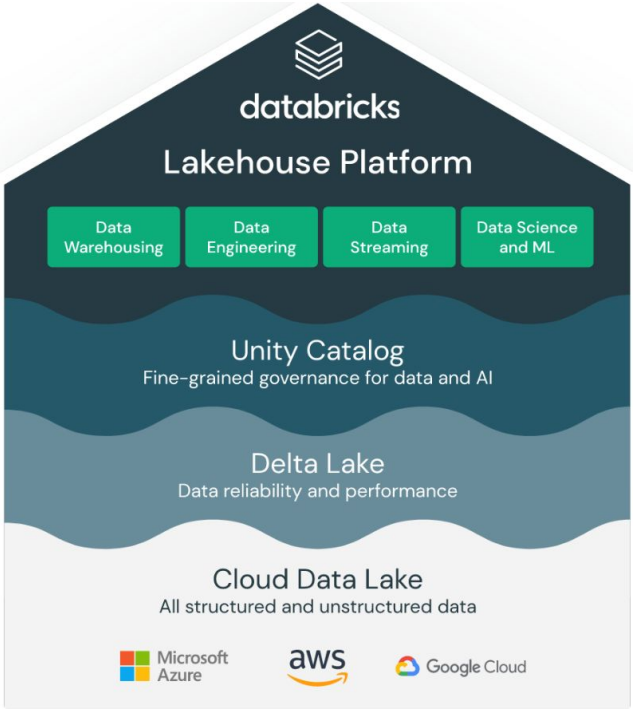
Databricks



Spark-based Platform in Cloud

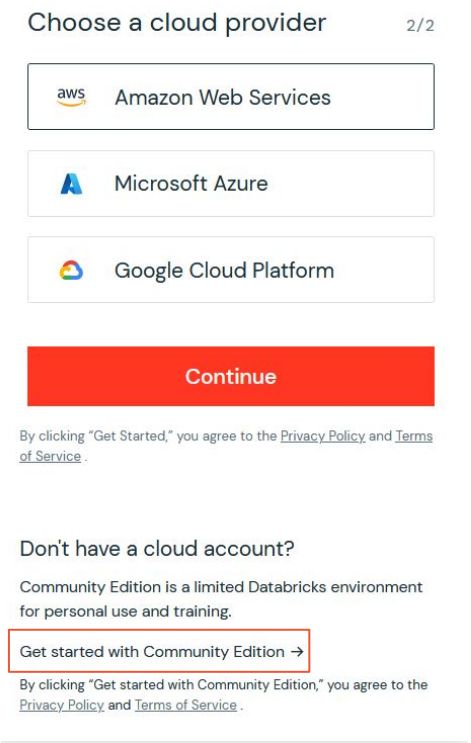
- Company founded in 2013
- No need to handle infrastructure and cluster management
- Focused on notebook-based development
- Auto-scaling, ad-hoc interactive queries
- New file format - Delta (Parquet + ACID)
- Databricks File System (DBFS) - abstraction for various storage solutions
- Lots of features that make life easier (Unity Catalog, Delta Live Tables, Autoloader, Workflows, etc.)

Lakehouse Platform

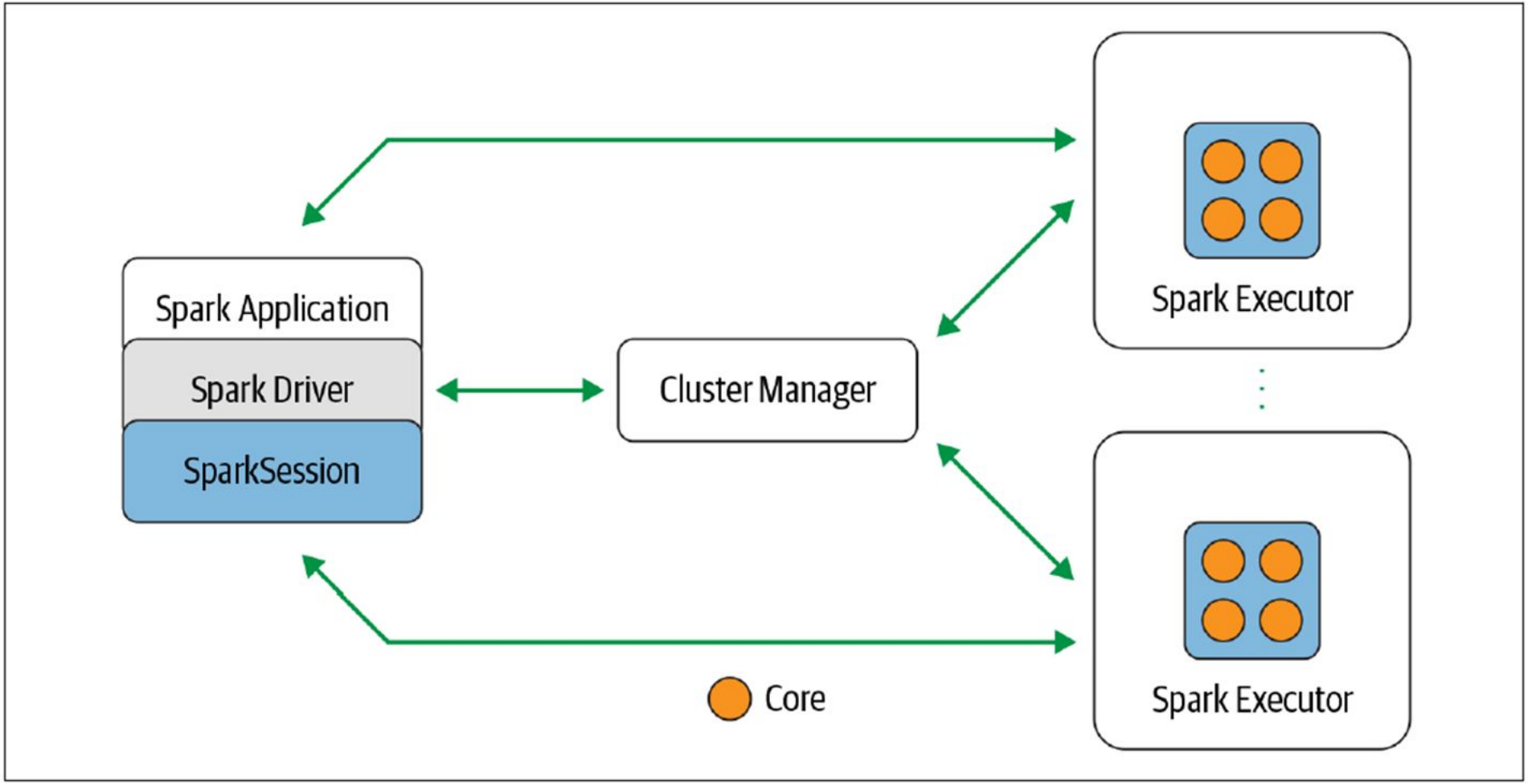


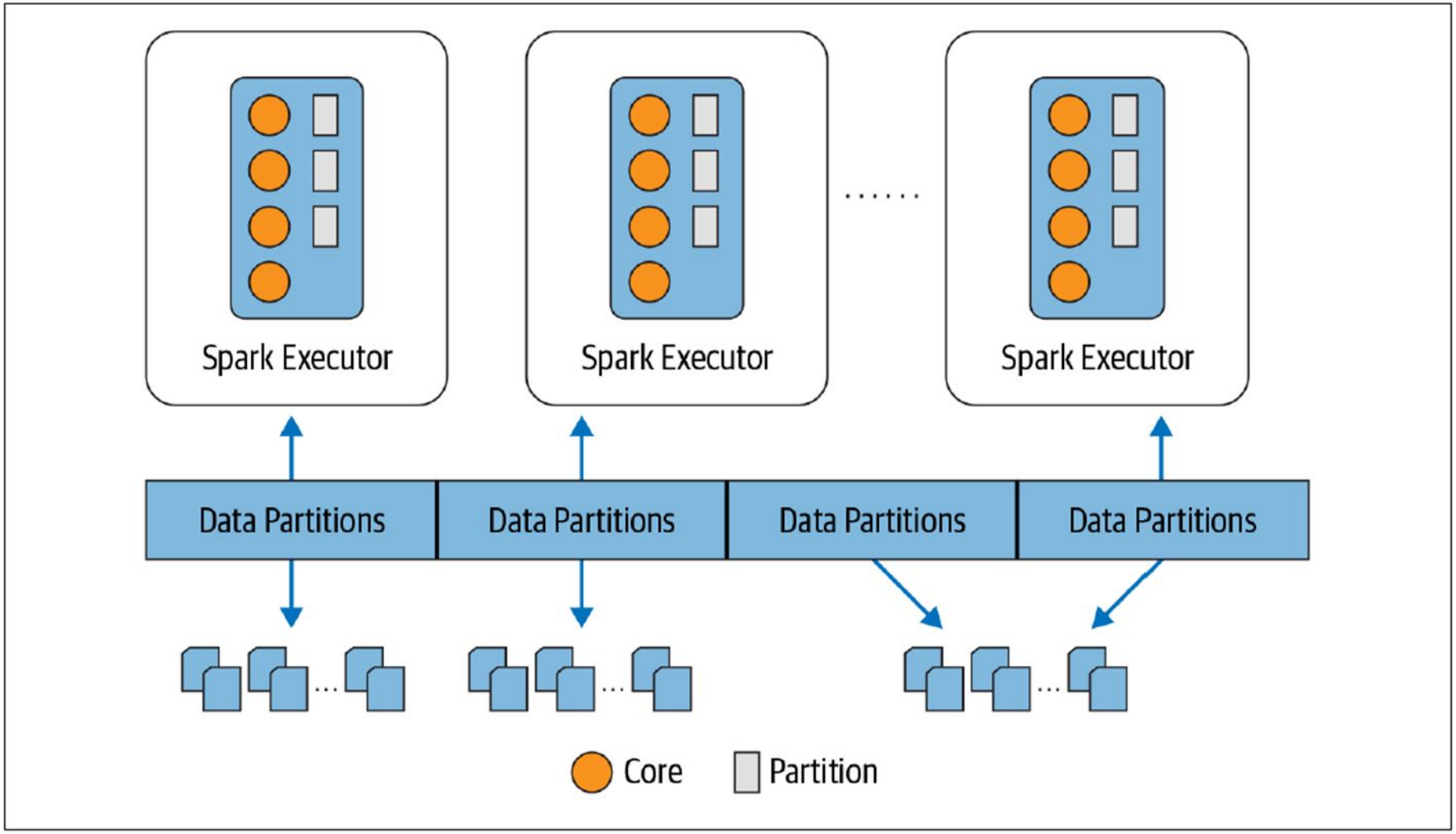
- Alternative to traditional Data Warehouses and Data Lakes
- Use of Delta Lake
- Data stored in cloud storage
- Limited vendor lock-in
- Unification for all roles
- Dedicated clusters or serverless SQL (DWH-like experience)

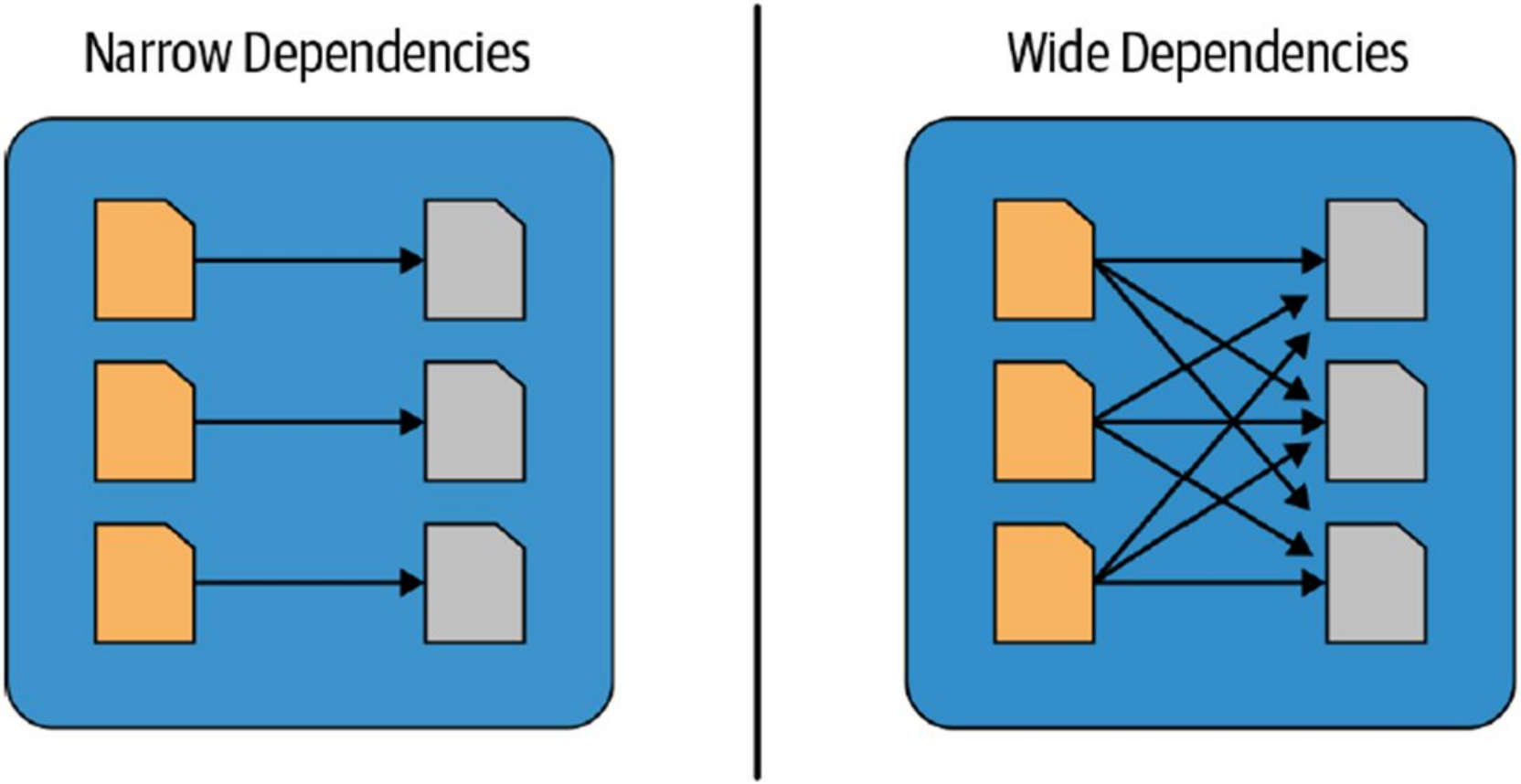
Databricks Community Edition

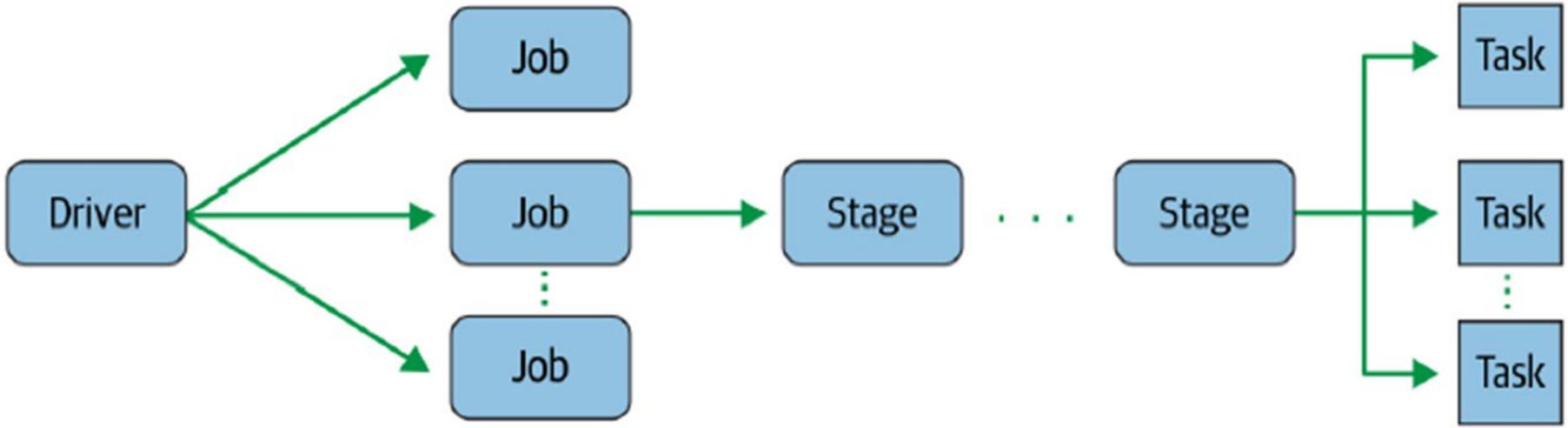


- For free
- Unlimited cluster time (limited cluster size)
- https://community.cloud.databricks.com


Spark Application
Spark Driver
SparkSession
Cluster Manager
Spark Executor
Spark Executor
Core


Spark Executor
Spark Executor
Spark Executor
Data Partitions
Data Partitions
Data Partitions
Data Partitions
Core
Partition


Narrow Dependencies
Wide Dependencies


Driver
Job
Job
Job
Stage
Stage
Task
Task
Task

Takeaways

It’s worth looking into Apache Spark and Databricks as they are considered the industry standard

It’s very easy to start working with Spark either locally or via Databricks

If you know SQL, you are almost there

GlobalLogic®

A Hitachi Group Company

Q&A

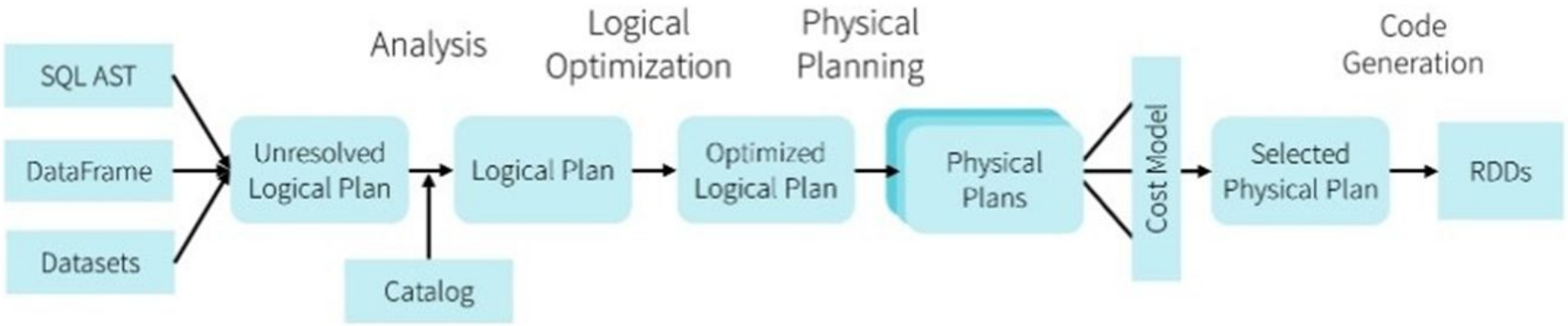
bit.ly/spark_2024

pavol.dudrik@globallogic.com

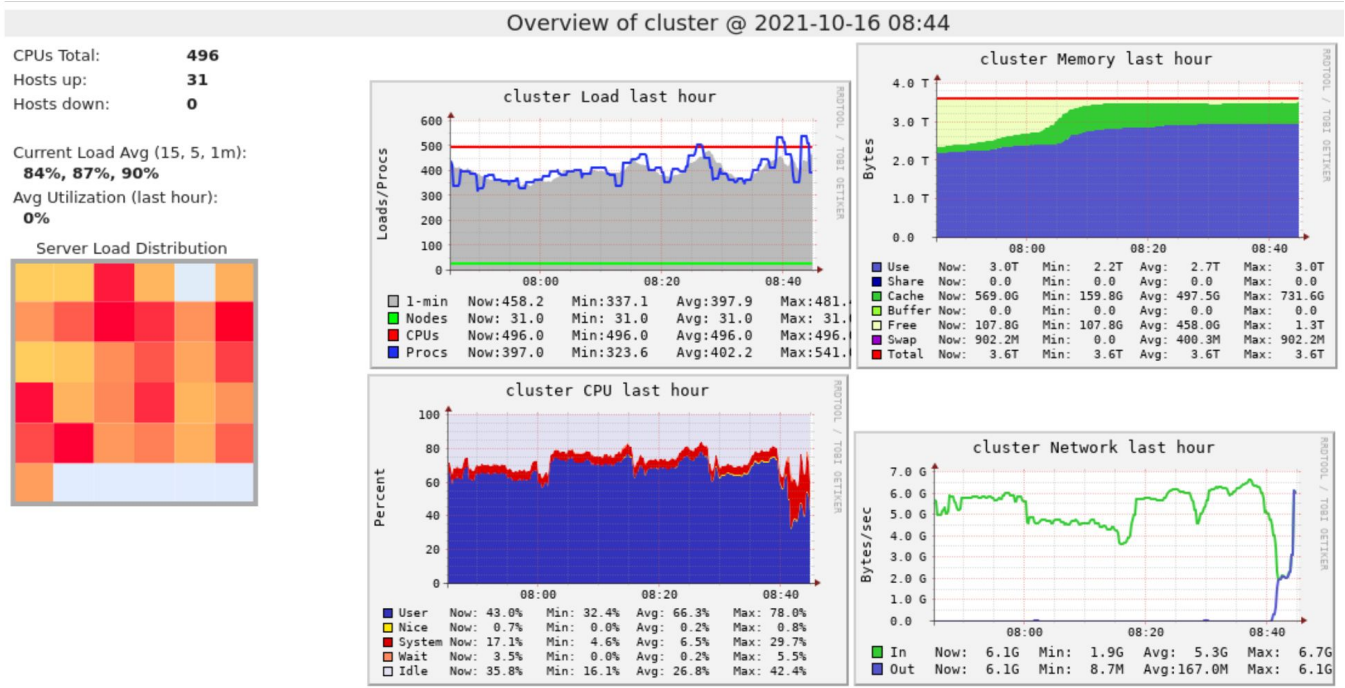
www.linkedin.com/in/pavoldudrik/

GlobalLogic®

A Hitachi Group Company


Analysis
Logical Optimization
Physical Planning
Code Generation
SQL AST
DataFrame
Datasets
Unresolved Logical Plan
Catalog
Logical Plan
Optimized Logical Plan
Physical Plans
Cost Model
Selected Physical Plan
RDDs

Apache Spark



Databricks

```
%sql
SELECT * FROM pdudrik_family
```

(2) Spark Jobs

	family_id	family_members
1	180388626432	["Jane", "Jim", "John"]
2	944892805120	["Adam", "Molly"]
3	601295421440	["Gary"]

Showing all 3 rows.

Databricks Runtime Version

8.1 (includes Apache Spark 3.1.1, Scala 2.12)

Autopilot Options

☑ Enable autoscaling

☑ Terminate after 60 minutes of inactivity

Worker Type		Min Workers	Max Workers	Current
Standard_DS5_v2	56 GB Memory, 16 Cores	2	8	2

Driver Type

Standard_DS5_v2 56 GB Memory, 16 Cores