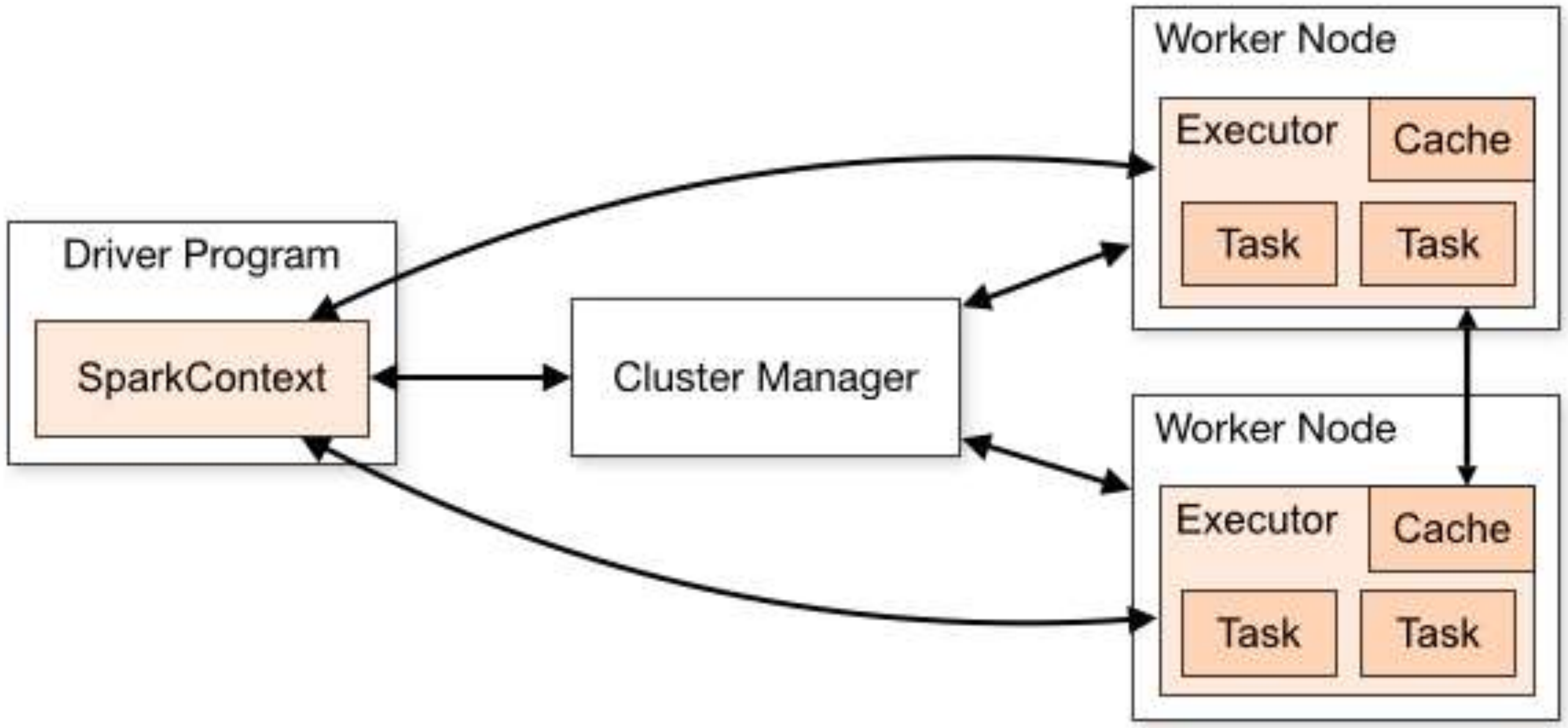
Introduction to Apache Spark

Spark - Intro

- A fast and general engine for large-scale data processing.
- Scalable architecuture
- Work with cluster manager (such as YARN)

Spark Context

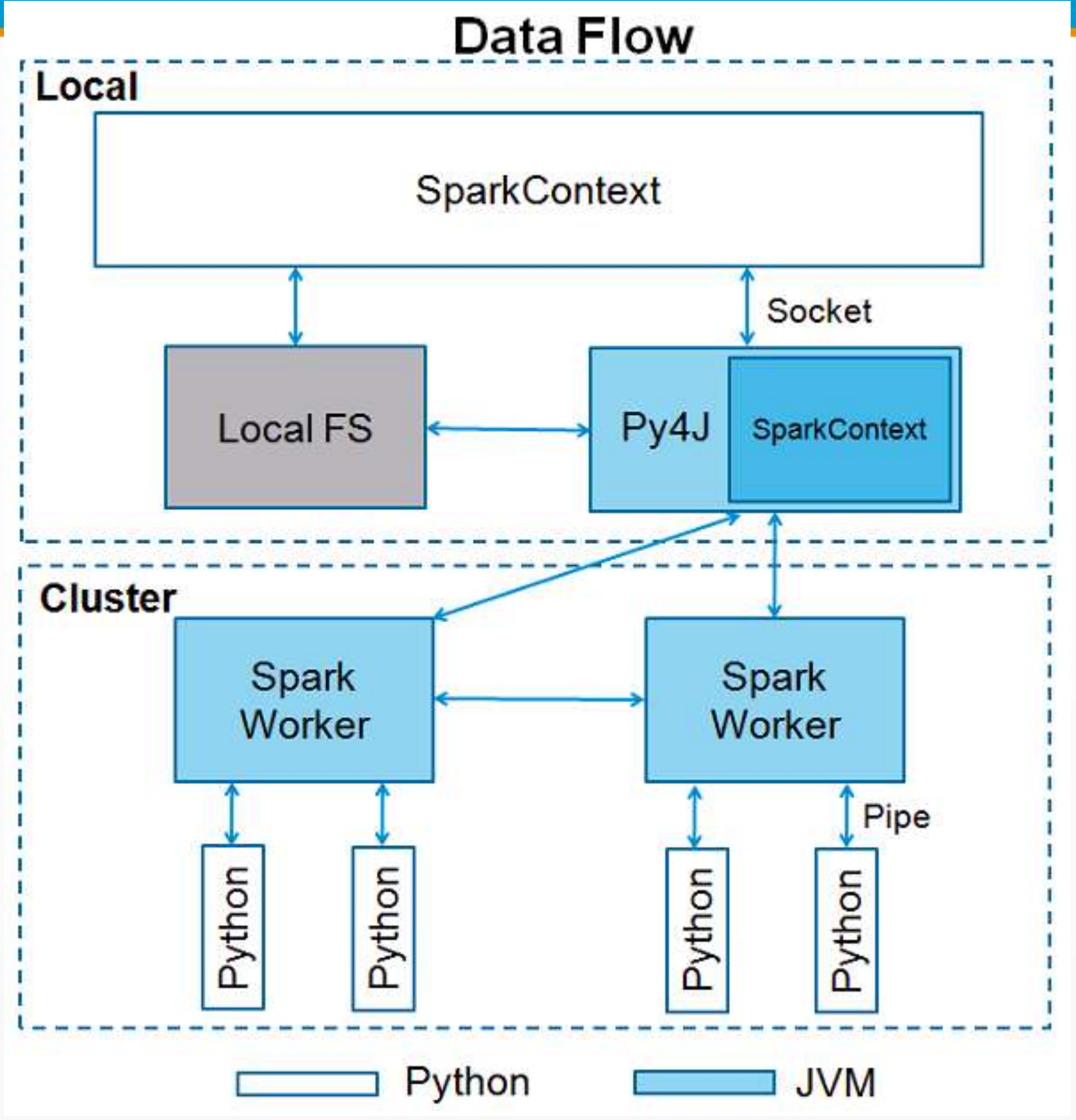


Spark Context

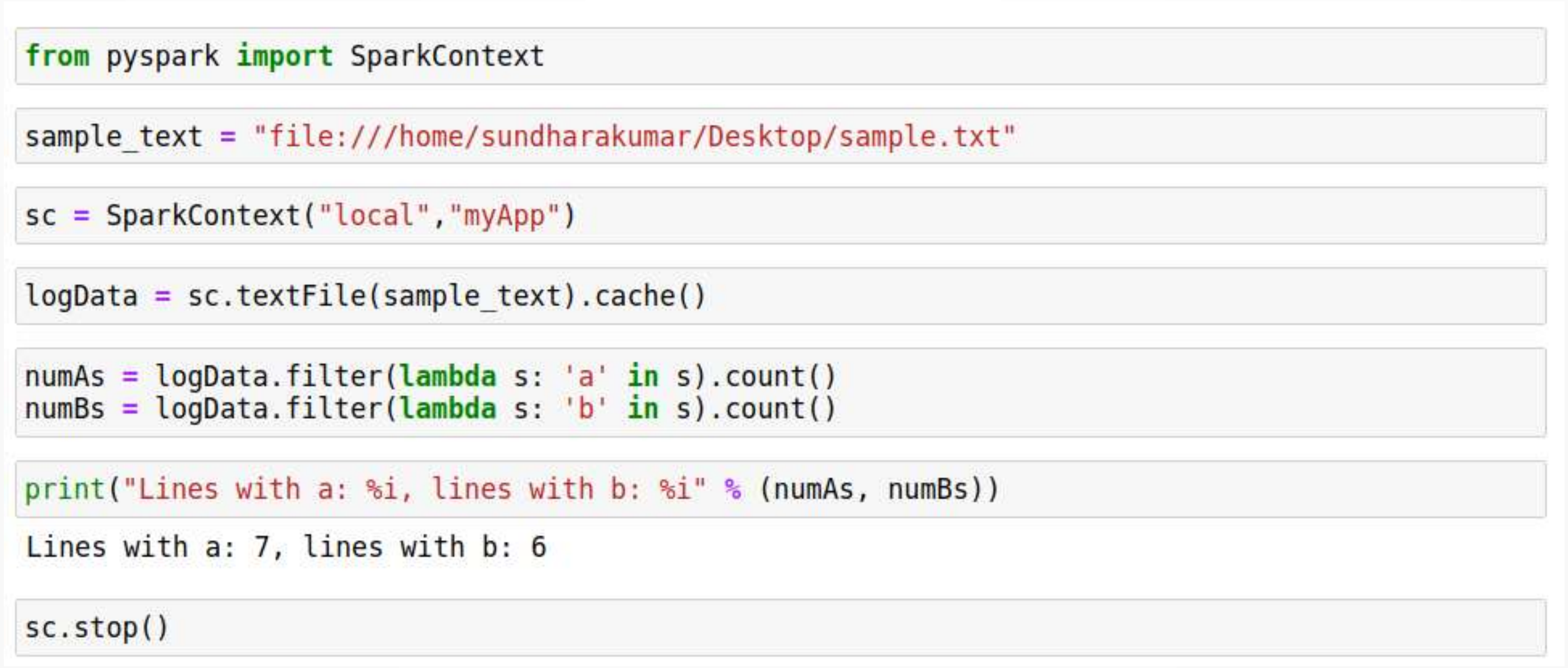
- SparkContext is the entry point to any spark functionality.
- As soon as you run a spark application, a driver program starts, which has the main function and the sparkcontext gets initiated.
- The driver program runs the operations inside the executors on the worker nodes.

Spark Context

- SparkContext uses Py4J to launch a JVM and creates a JavaSparkContext.
- Spark supports Scala, Java and Python. PySpark is the library to be installed to use python code snippets.
- PySpark has a default SparkContext library. This helps to read a local file from the system and process it using Spark.


Data Flow
Local
SparkContext
Socket
Local FS
Py4J
SparkContext
Cluster
Spark Worker
Spark Worker
Pipe
Python
Python
Python
Python
Python
JVM

Sample program



```
from pyspark import SparkContext
```

```
sample_text = "file:///home/sundharakumar/Desktop/sample.txt"
```

```
sc = SparkContext("local","myApp")
```

```
logData = sc.textFile(sample_text).cache()
```

```
numAs = logData.filter(lambda s: 'a' in s).count()
numBs = logData.filter(lambda s: 'b' in s).count()
```

```
print("Lines with a: %i, lines with b: %i" % (numAs, numBs))
```

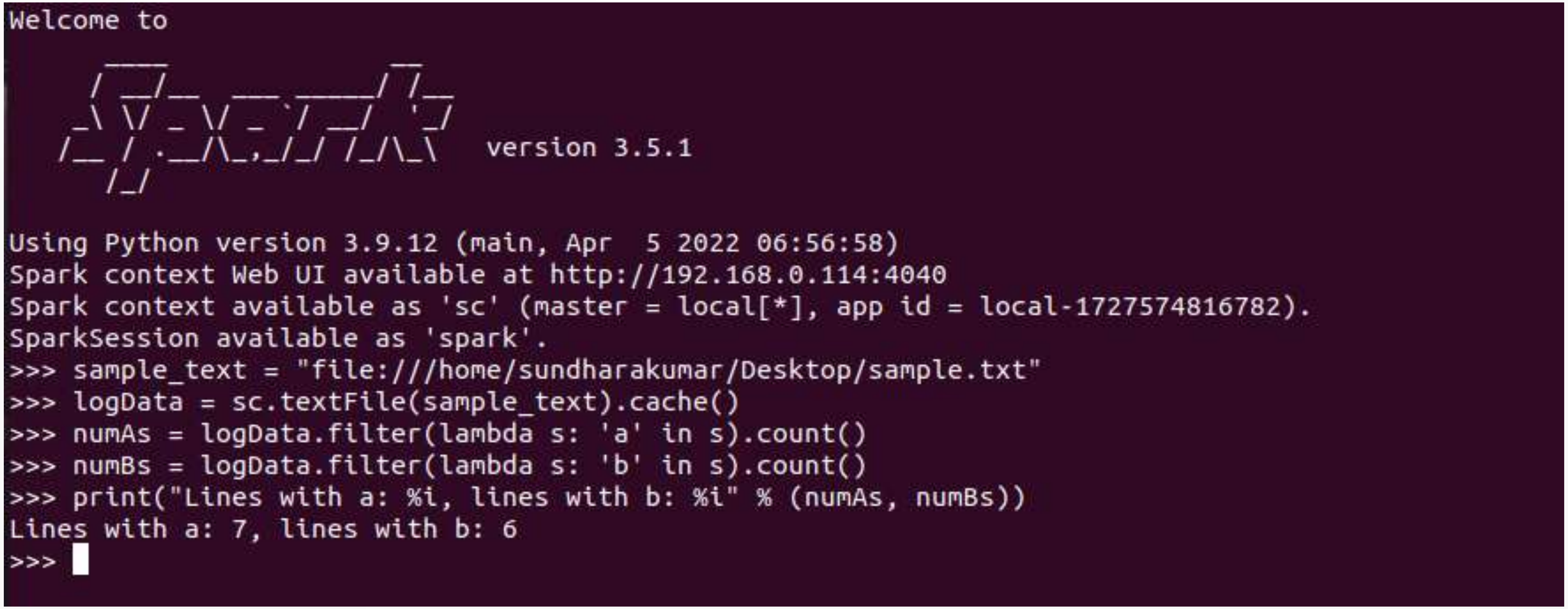
```
Lines with a: 7, lines with b: 6
```

```
sc.stop()
```

SparkShell

- Simple interactive REPL (Read-Eval-Print-Loop).
- Provides a simple way to connect and analyze data interactively.
- Can be started using pyspark or spark-shell command in terminal. The former supports python based programs and the latter supports scala based programs.

SparkShell



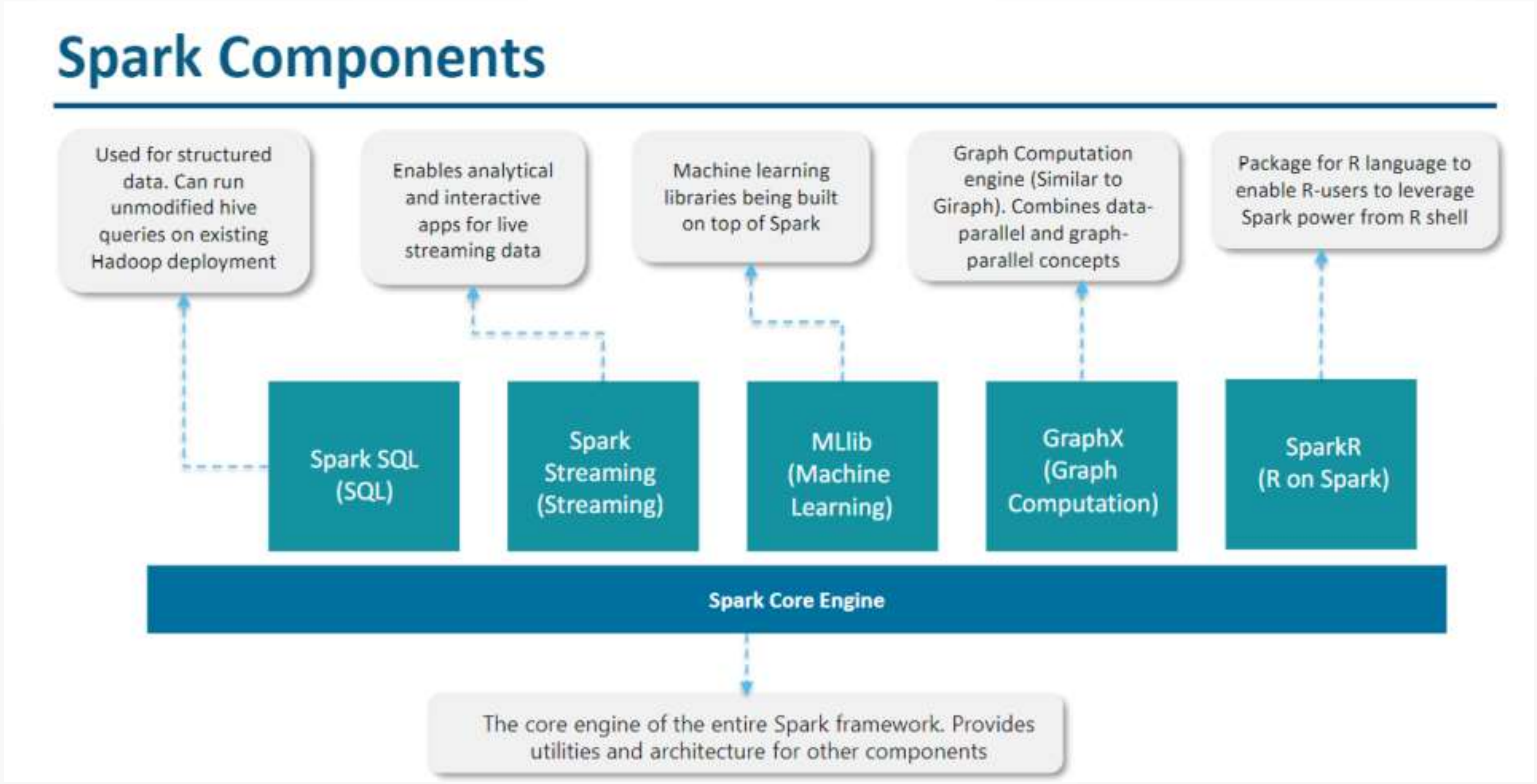
```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.5.1
      /_/

Using Python version 3.9.12 (main, Apr  5 2022 06:56:58)
Spark context Web UI available at http://192.168.0.114:4040
Spark context available as 'sc' (master = local[*], app id = local-1727574816782).
SparkSession available as 'spark'.
>>> sample_text = "file:///home/sundharakumar/Desktop/sample.txt"
>>> logData = sc.textFile(sample_text).cache()
>>> numAs = logData.filter(lambda s: 'a' in s).count()
>>> numBs = logData.filter(lambda s: 'b' in s).count()
>>> print("Lines with a: %i, lines with b: %i" % (numAs, numBs))
Lines with a: 7, lines with b: 6
>>> 
```

Features

- Runs programs up to 100x faster than Hadoop Mapreduce in memory or 10x faster in disk.
- DAG engine – a directed acyclic graph is created that optimzes workflows.
- Lot of big players like amazon, eBay, NASA Deep space network, etc. use Spark.
- Built around one main concept: Resilient Distributed Dataset (RDD).

Components of spark


Spark Components
Used for structured data. Can run unmodified hive queries on existing Hadoop deployment
Enables analytical and interactive apps for live streaming data
Machine learning libraries being built on top of Spark
Graph Computation engine (Similar to Giraph). Combines data-parallel and graph-parallel concepts
Package for R language to enable R-users to leverage Spark power from R shell
Spark SQL (SQL)
Spark Streaming (Streaming)
MLlib (Machine Learning)
GraphX (Graph Computation)
SparkR (R on Spark)
Spark Core Engine
The core engine of the entire Spark framework. Provides utilities and architecture for other components

RDD – Resilient Distributed Datasets

- This is the core object on which the spark revolves including SparkSQL, MLLib, etc.
- Similar to pandas dataframes.
- RDD can run on standalone systems or a cluster.
- It is created by the sparkcontext object.

Creating RDDs

- Nums = sc.parallelize([1,2,3,4])
- sc.textFile("file:///users/....txt")
  - Or from s3n:// or hdfs://
- HiveCtx = HiveContext(sc)
- Can also be created from
  - JDBC, HBase, JSON, CSV, etc.

Operations on RDDs

- Map
- Filter
- Distinct
- Sample
- Union, intersection, subtract, cartesian

RDD actions

- collect
- count
- countByValue
- reduce
- Etc...
- Nothing actually happens in the driver program until an action is called.! - **Lazy Evaluation**