Spark

Big Data

Working with Big Data

- Lack of Big Data Handling Skills
- Data Storage
- Querying & Analysis

Distributed Computing

- Get several computers to do a certain work at the same time

Distributed Computing

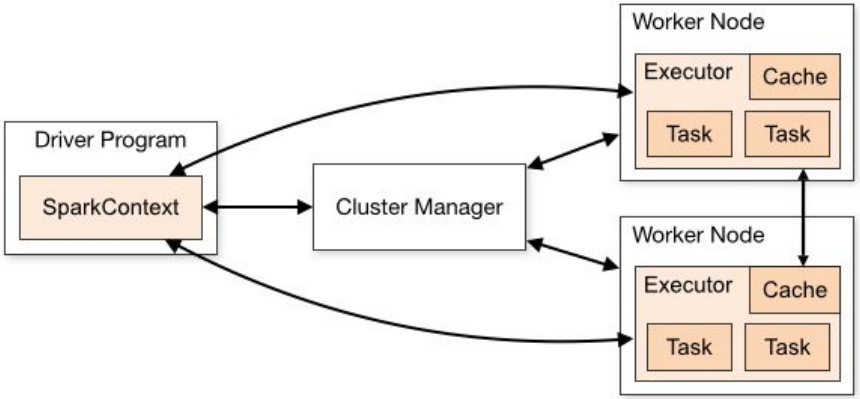
- Get several computers to do a certain work at the same time
- Systems for processing huge data in a distributed manner e.g Apache Hadoop and Apache Spark

Apache Hadoop

- Enables the distributed processing of large data sets across a cluster of computers
- Can scale from a single server to thousands of servers

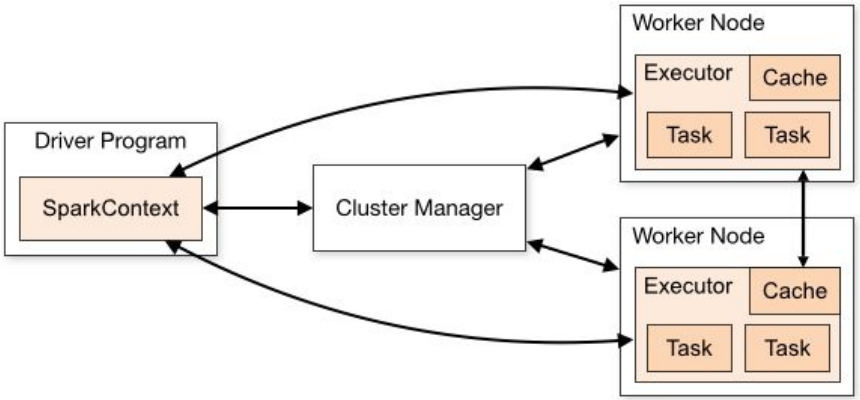
What is Apache Spark?

- Apache Spark is an open-source analytics engine for **large-scale data processing**
- Spark applications consists of a **driver program** that runs the **user's main function** and executes various **parallel** operations on a **cluster**



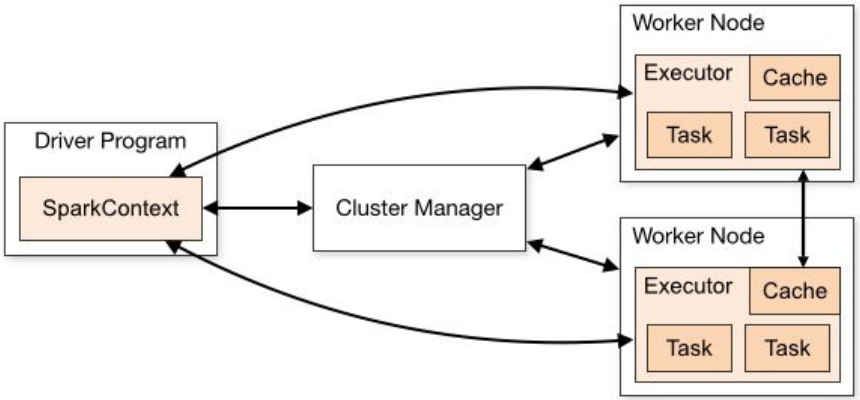
What is Apache Spark?

- A **cluster** is made up of many **nodes**. A node is a single machine or server.
- Spark applications are controlled by **SparkContext**. The SparkContext connects to the **cluster manager**.



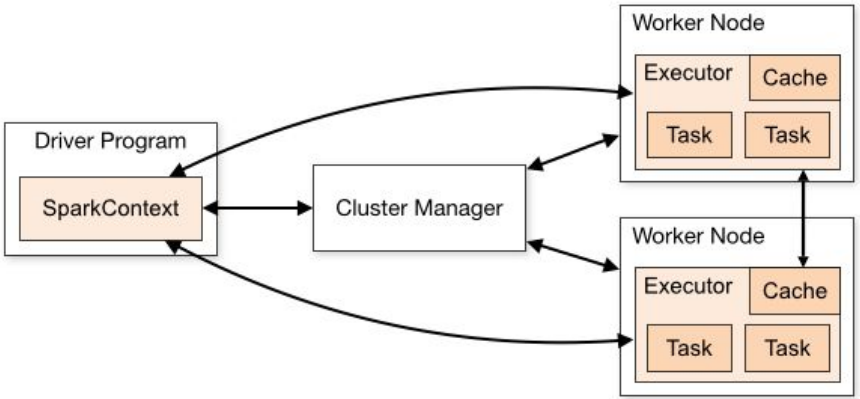
What is Apache Spark?

- There are several **cluster managers**, i.e Spark's own standalone cluster manager, Mesos, or YARN.
- The cluster managers allocate resources to various Spark applications.



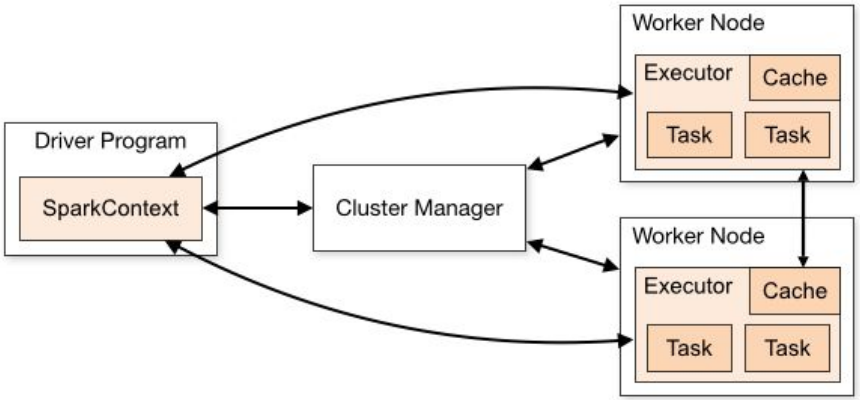
What is Apache Spark?

Executors on nodes are processes that run computations and store data for your application. So the Executor is a process that is initiated for an application on a worker node, it runs tasks and keeps data in memory or disk storage across them. Each application has its own executors. Tasks are sent to the executor by the SparkContext. Every application has its own executor program, so applications are isolated from each other.



What is Apache Spark?

- A **worker node** is any node that can run application code in a cluster.
- A **task** is a unit of work that will be sent to one executor.
- A parallel computation involving multiple tasks is known as a **job**.



Why Apache Spark?

- Speed because of in memory computation
- Ease of use
- Runs Everywhere
- A Unified Engine

Apache Spark Data Representations

- Resilient Distributed Datasets (RDDs)
- Dataframe
- Datasets

Resilient Distributed Datasets (RDDs)

- a fault-tolerant collection of elements that can be operated on in parallel
- RDDs automatically recover from node failures
- Used when:
    - low-level transformations on a dataset is needed
    - data is unstructured eg media or text streams

RDD Creation

- parallelizing an existing collection in your driver program
- referencing a dataset in an external storage system, such as a shared filesystem

```
my_list = [1, 2, 3, 4, 5]

my_list_distributed = sc.parallelize(my_list,4)

distributed_file= sc.textFile("file.txt")
```

RDD Persistence

- Achieved by persisting or caching a dataset in memory
- Kept in memory in the node the first time it is computed in an action
- The cache is fault tolerant
- If any RDD partition is lost, it will be re-computed using the transformations that created it

RDD Operations

- Transformations - creates a new dataset from an existing one e.g a map
- Actions return a value after running a computation on the dataset e.g a reduce

Spark Transformations

- Transformations in Spark are lazy, they do not compute their results right away
- Transformations are only computed when an action requires a result to be returned

Transformation Types

- map(func) - return a new **distributed dataset** resulting from passing each element through a function
- filter(func) - return a new dataset formed by selecting the items that return true on a certain condition
- union(otherDataset) - Return a new dataset that is the union of two datasets

Action Types

– – –

- collect() - return all the elements of the dataset as an array
- count() - return the number of items in a dataset
- take(n) - return the first n elements of the dataset
- first() - return the first item in the dataset

Spark DataFrames

- A Spark DataFrame is an immutable distributed collection of data
- Very similar to Pandas DataFrames
- Can be queried as if they were SQL Tables

Section Summary

- Distributed Computing
- Apache Spark
- Why Apache Spark
- Data Representation in Apache Spark
- Operations - Transformations & Actions