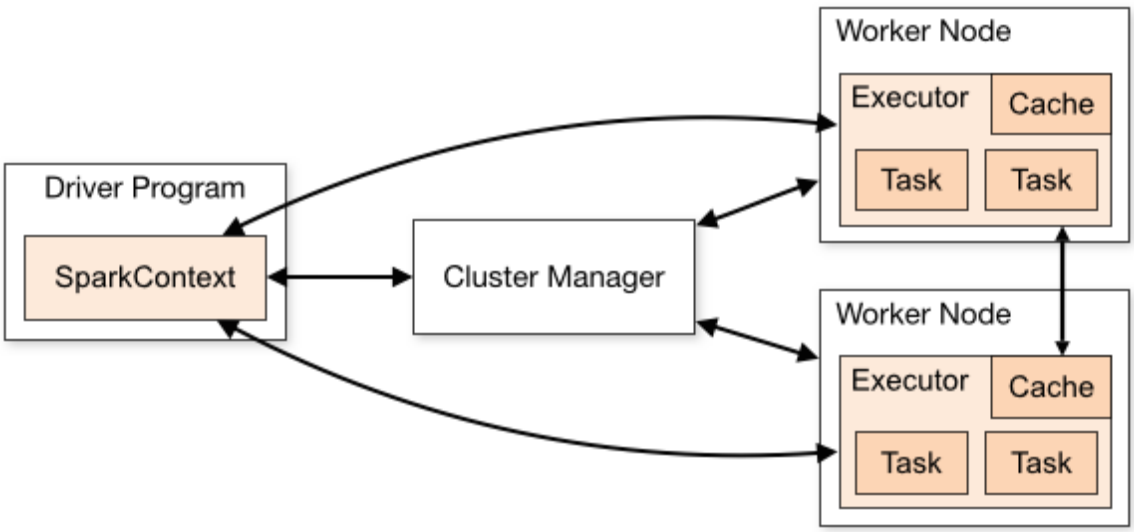
Spark

概述

Apache Spark 是专为大规模数据处理而设计的快速通用的计算引擎。

和**Hadoop**一样，**Spark**也是基于分布式文件系统**HDFS**的批处理计算框架。但是Spark是基于内存的计算框架，速度比Hadoop快很多。

结构



- Spark应用跑在**Driver Program**上，由**SparkContext**进行管理；
- SparkContext可以连接到多种**Cluster Manager**上，比如**YARN**，**Mesos**，**K8s**等等。
- SparkContext通过Cluster Manager从集群中获取**executors**， 把应用代码（Python或者jar包）发送到executors，并分配任务给它们执行。

注意：

- Driver Program对于worker nodes而言必须**network addressable**
- Driver Program必须靠近worker，最好位于同一块网络区域，以加快数据传输的速度。

配置

RDD

概述

RDD（Resilient Distributed Dataset）叫做**弹性分布式数据集**，**是Spark中最基本的数据抽象**，它代表一个不可变、可分区、里面的元素可并行计算的集合。RDD具有数据流模型的特点：自动容错、位置感知性调度和可伸缩性。RDD允许用户在执行多个查询时显式地将工作集缓存在内存中，后续的查询能够重用工作集，这极大地提升了查询速度。

数据源

RDD的数据源可以是本地文件也可以是分布式文件系统。

操作

RDD支持两类操作：

- **transformations**：在原有的数据集的基础上新建一个数据集，两者之间存在依赖关系。
- **actions**：返回运算结果到Driver Program。是一个汇总的动作。

以常见的map和reduce为例，前者就是一个transformation，在本地执行；而后者就是一个action，将运算结果返还给driver program进行reduce。

RDD transformations

- 所有的transformations都是**lazy**，即不会立即计算结果，而是需要结果（执行action）的时候才会进行运算。
- 默认情况下，每个RDD在每次执行action的时候都会重新计算，但是也可以指定缓存到内存里（当然这并不一定能够保证内容一定存储在内存中，也可能会通过压缩保存到磁盘上）。

For RDD {1, 2, 3, 3}

Transformations	Meaning	Example	Result
map(func)	Return a new distributed dataset formed by passing each element of the source through a function func.	rdd.map(x => x + 1)	{2, 3, 4, 4}
flatMap(func)	Similar to map, but each input item can be mapped to 0 or more output items (so func should return a Seq rather than a single item).	rdd.flatMap(x => x.to(3))	{2, 3, 4, 4}
filter(func)	Return a new dataset formed by selecting those elements of the source on which func returns true.	rdd.filter(x => x != 1)	{2, 3, 3}
distinct()	Return a new dataset that contains the distinct elements of the source dataset.	rdd.distinct()	{1, 2, 3}

For RDDs {1, 2, 3} and {3, 4, 5}

Transformations	Meaning	Example	Result
union(otherDataset)	Return a new dataset that contains the union of the elements in the source dataset and the argument.	rdd.union(other)	{1, 2, 3, 3, 4, 5}
intersection (otherDataset)	Return a new RDD that contains the intersection of elements in the source dataset and the argument.	rdd.intersection(other)	{3}
cartesian (otherDataset)	When called on datasets of types T and U, returns a dataset of (T, U) pairs (all pairs of elements).	rdd.cartesian(other)	{(1, 3), (1, 4), ..., (3,5)}

RDD actions

For RDD {1, 2, 3, 3}

Transformations	Meaning	Example	Result
collect()	Return all the elements of the dataset as an array at the driver program. This is usually useful after a filter or other operation that returns a sufficiently small subset of the data.	rdd.collect()	{1, 2, 3, 3}
count()	Return the number of elements in the dataset.	rdd.count()	4
reduce(func)	Aggregate the elements of the dataset using a function func (which takes two arguments and returns one). The function should be commutative and associative so that it can be computed correctly in parallel.	rdd.reduce((x,y) => x+y)	9