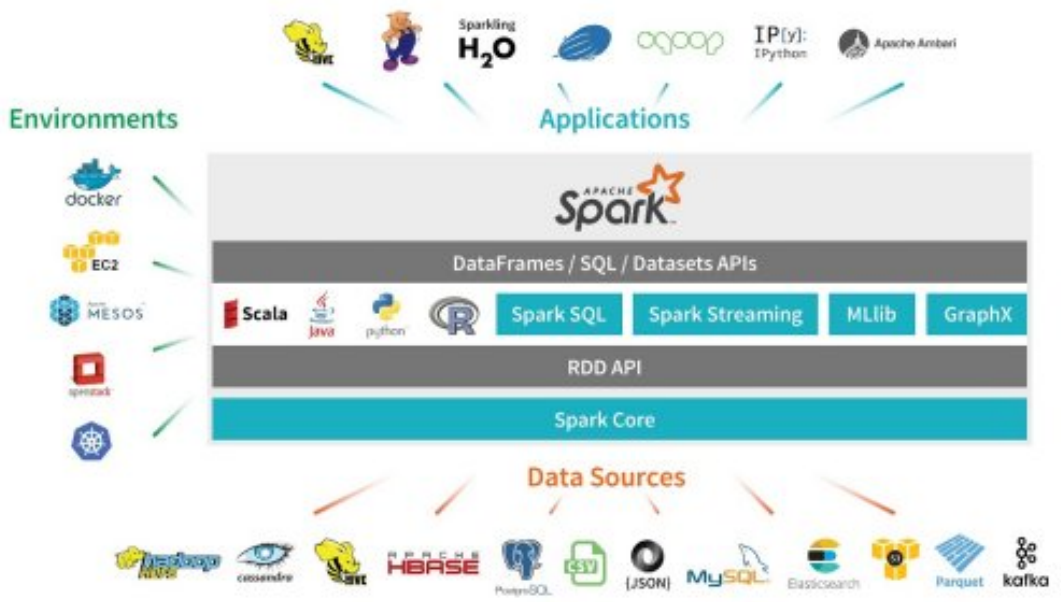
Why Spark!

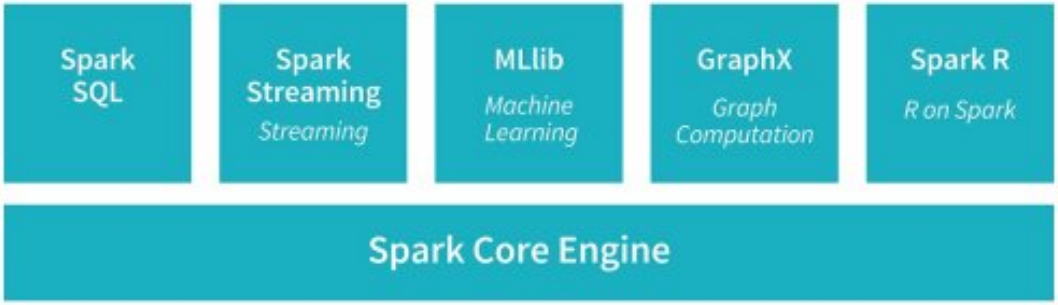
首先、Spark是易于使用的快速处理且提供高级分析功能的开源数据处理引擎，开源社区非常活跃!

第二、作为一个大规模分布式数据处理通用计算引擎，Spark通过一个统一API，支持流行的编程语言，包括Scala、Java、Python和R!

最后、它可以部署在多种环境中，支持多种数据源读取，与各种应用程序交互!



统一的核心计算引擎使得在一个程序中就可以同时完成如：ETL、Spark SQL、Machine Learning、GraphX/ GraphFrames和Spark Streaming



在随后的讲解中，部分组件我们会重点介绍，首先让我们介绍一下它关键概念和术语。

Apache Spark的概念、关键术语和关键词

2016年6月，KDnuggete发表了《Apache Spark的关键术语解释》（http://www.kdnuggets.com/2016...），这是一个非常不错的介绍。下面补充一些Spark的术语词汇表，它们都将经常在本文中出现。

Spark Cluster

在云端或者安装Spark的数据中心预置的一组机器或者节点。那些机器就是Spark workers、Spark Master（在一个独立的模式下的集群管理器）和至少一个Spark Driver。

Spark Master

顾名思义，Spark Master JVM在一个独立的部署模式下作为集群的管理器，Spark Works注册它们自己作为集群的一部分。根据部署模式，它作为一个资源管理器，决定在集群的哪台机器发布多少个执行器。

Spark Worker

Spark Worker JVM，在接到来自Spark Master的指令后，代表Spark driver发布执行器。Spark的应用程序，分解为任务单元，被每个Worker的执行器执行。简而言之，Worker的工作是代表Master发布一个执行器。

Spark Executor

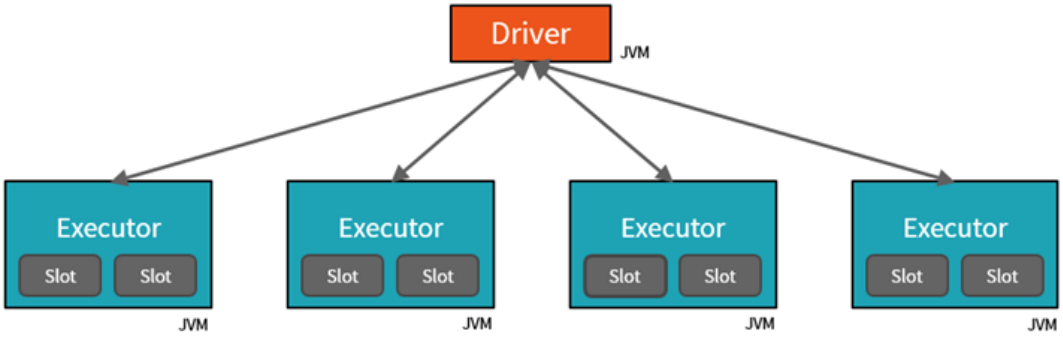
它是一个分配好处理器和内存数量的JVM容器， Spark在其上运行它的任务。每个Worker节点通过一个可配置的核心(或线程) 发布自己的Spark执行器。除了执行Spark任务，每个执行器还在内存中存储和缓存数据分区。

Spark Driver

一旦它从Spark Master得到集群中所有的Worker的信息，驱动程序

就为每个Worker的执行器分配Spark的任务。Drive也从每个执行器的任务中获得计算结果。

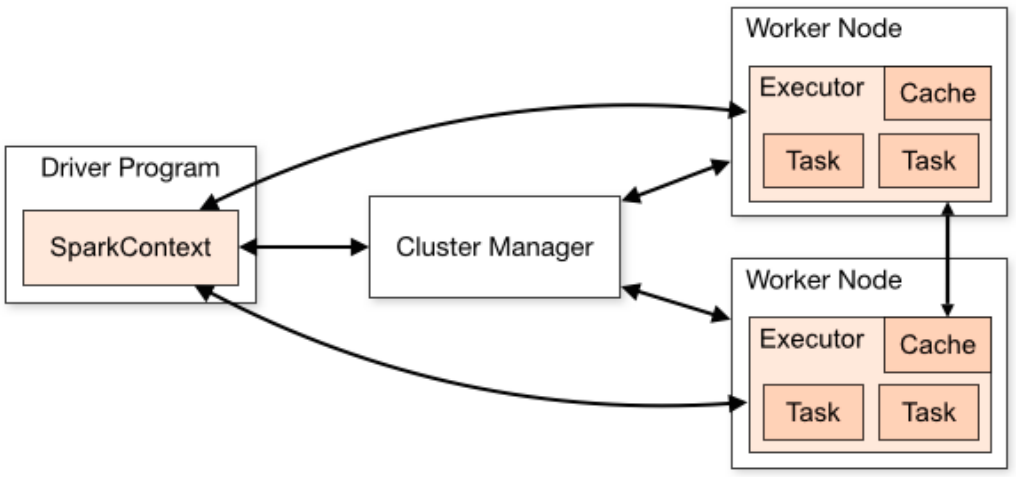
Spark Physical Cluster



SparkSession和SparkContext

如图表所示,SparkContext是访问所有Spark功能的渠道;在每个JVM只有一个SparkContext。Spark驱动程序使用它连接到集群管理器进行通信并提交Spark工作。它允许您配置Spark参数。通过SparkContext，驱动可以实例化其他contexts，例如SQLContext、HiveContext、StreamingContext。

使用Apache Spark 2.0,SparkSession可以通过一个统一的入口点访问所有提到的Spark的功能，同时可以更简单地访问Spark功能，以及贯穿底层context来操作数据。



Spark部署模式

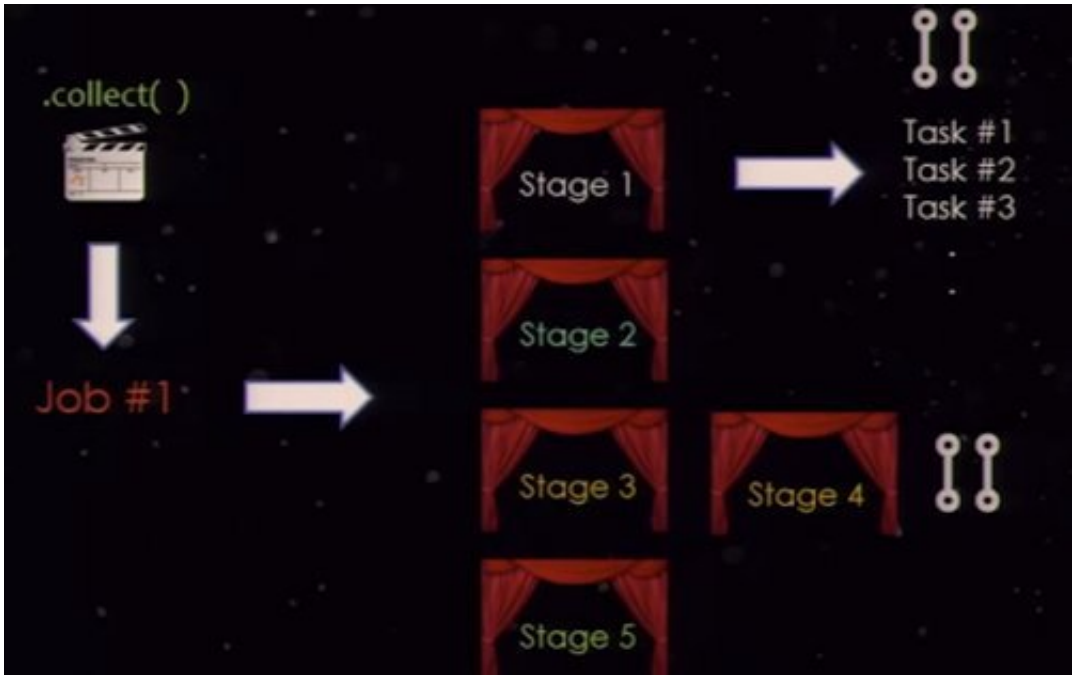
Spark支持四个集群部署模式,对应运行在Spark集群里的Spark的组件，每个都有自己的特点。所有模式中，本地模式是在一个单独的主机上运行,是目前为止最简单的。

作为初级或中级开发人员是不需要知道这个复杂表格的，在这里供您参考。此外,本文的第五步会深入介绍Spark体系结构的各个方面。

Mode	Driver	Worker	Executor	Master
本地	跑在单一的JVM上	跑在和Driver同一个JVM上	跑在和Driver同一个JVM上	跑在单一主机上
单机	跑在集群的任意节点上	跑在每个节点自己的JVM上	每个worker会发布它自己的JVM	可以被任意分配到master开的地方
YARN（客户端）	在客户端上，并不是集群的一部分	YARN节点管理器	YARN节点管理器的容器	YARN的资源管理器通过ApplicationMaster来为执行器分配节点管理器上的容器
YARN（集群）	跑在YARN的Application Master内	同客户端模式	同客户端模式	同客户端模式
Mesos（客户端）	在客户端机器上，并不是Mesos集群的一部分	跑在Mesos Slave上	Mesos Slave上的容器	Mesos的master
Mesos（集群）	跑在Mesos的一个master内	同客户端模式	同客户端模式	同客户端模式

Spark的Apps, Jobs, Stages and Tasks

一个Spark应用通常包括了数个Spark的操作，可以分解为数据集上的transformation或者action，来使用Spark的RDD、数据框或者数据集。举例来说，在Spark应用，如果你调用一个action，这个action会产生一个job。一个job会分解成单一或者多个stage；stage会进一步切分成单独的task；task是执行单元，Spark driver的调度会将其运送到Spark worker节点上的Spark执行器进行执行。通常多task会并行跑在同一个执行器上，在内存的分区数据集上分别进行单元进程。


.collect( )
Job #1
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Task #1
Task #2
Task #3