## 课程的Spark版本

spark-2.2.0-bin-hadoop2.6.tgz，cdh最高版本目前集成的Spark 为v1.6。

Spark 2在性能上大大增强，编程模型更加统一和简单。

<http://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/5.9.0/>

**Spark2** 比spark1 在性能上很大提升，编程模型更加简洁。

## Spark是什么？

Spark 是 Apache 顶级项目里面最火的大数据处理的内存计算引擎，包括离线计算或交互式查询、数据挖掘算法、流式计算以及图计算等。

与 Mapreduce 相比，Spark 具备 DAG 执行引擎以及基于内存的多轮迭代计算等优势，在SQL 层面上，比Hive引入了更多关系数据库的特性，以及内存管理技术。

Spark 基于**磁盘**的性能是 MR 的 10 倍，基于内存的性能是 MR 的 100 倍。

Spark 提供 SQL、机器学习库 MLlib、流计算 Streaming 和图计算 Graphx，同时也支持 Scala、Java、Python 和 R 语言开发的基于 API 的应用程序。

MR：纯硬盘计算引擎

Spark ：内存计算引擎

## Spark应用场景

·替换Mapreduce，离线计算，统计分析类。

·Hive on spark，用spark置换Hive的Mapreduce计算引擎，离线计算，统计分析类。

·替换Storm，进行实时流计算场景。

对比Storm，Spark开发简单得多。但是当你需要1s或毫秒级计算时，Spark不是好选择，此时用Storm为宜。

·图计算 Graphx，关系数据的计算。

这里的图不是图片，是关系数据。

查询：我的朋友的朋友的朋友的朋友，从mysql里查询是一个多表关联（单表多层自关联），性能极差。

图数据库：基于图数据库的话，该场景只是一个普通查询，存储和提供查询关系数据，Neo4j。

图计算、Sql加工转换——》关系数据——》图数据库

·替换Mahout（基于MR计算引擎）数据挖掘算法包。Spark的MLlib（机器学习lib）里有丰富的数据挖掘算法，比Mahout的更丰富。值得注意的是Mahout项目不更新了。

## Spark运行模式

Spark 的执行模式有 local、Yarn、Standalone、Mesos 四类。

Yarn模式最广泛。

开发和测试用 local 模式，其实就是用多线程模似分布式执行。

如果业务部门较少且不需要对部门或组之间的资源做划分和优先级调度的话，可以使用 Standalone 模式来部署。

当如果有多个部门或组，且希望每个组织可以限制固定运行的最大资源，另外组或者任务需要有优先级执行的话，可以选择 Yarn 或 Mesos。

**Standalone**模式，即独立模式，master/slave(worker)，自带完整的服务，可单独部署到一个集群中，无需依赖任何其他资源管理系统。需要启动Master和Worker守护进程，即服务端进程，就好比Mapreduce的JobTracker和TaskTracker。

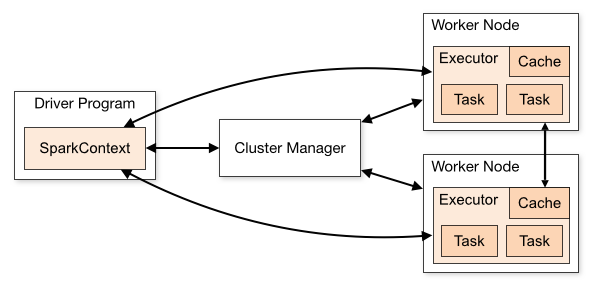
Spark on Yarn: 把Spark作业的调度和资源分配交给Yarn，Yarn相当于Spark集群的Master，

Spark无自己的守护进程，仅仅作为客户端存在。

MR（Hive）、Storm、Tez、Spark 期望这些作业有统一的调度和资源分配的角色，Yarn（MR2）.

## Spark集群架构

Spark也是**master/slave**架构，官方提供的架构图如下：



有中心就会存在单点故障，比如HDFS中Namenode挂掉的话，HDFS就不能用了。解决办法HA。

现在越来越多的新技术都讲究无中心（Master）设计，Zookeeper、Kafka。

角色解释如下：

Driver程序：称作驱动器程序，即包含SparkContext的，含义main，是整个应用程序的入口。

执行用户程序，转为Task任务发给Worker节点。相当于功能强大的Client。

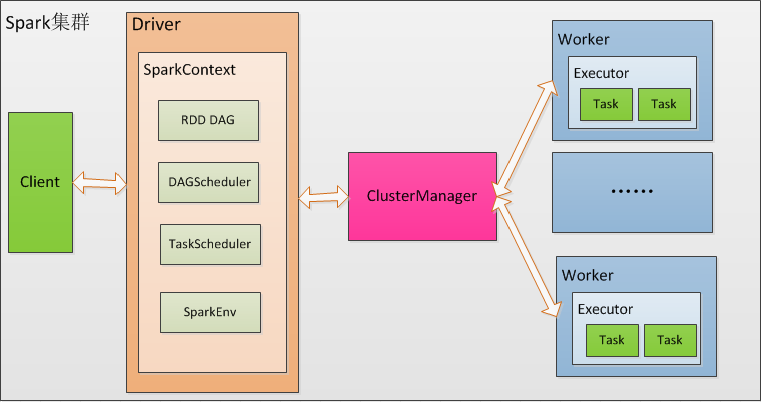
Cluster Manager：集群管理器，相当于master节点，负责资源管理和调度工作，如Yarn。

Standalone是自带的集群管理器。

Worker节点：工作节点，相当于slave节点。

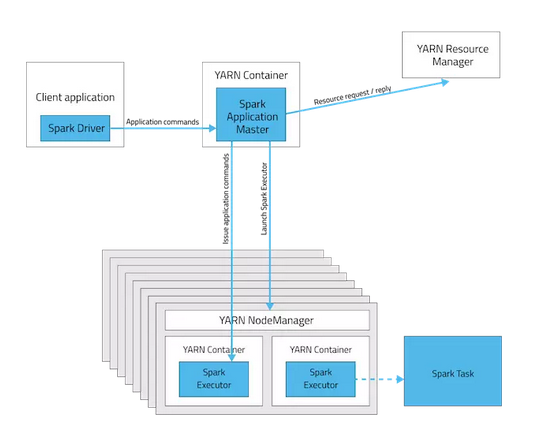
**Standalone架构图：**

该模式当前企业运用的比较少。



通过bin/start-all.sh 和 stop-all.sh 启动和停止集群的Master和Worker守护进程，即服务端进程，就好比Mapreduce的JobTracker和TaskTracker。

**Spark on Yarn架构图如下：**



当前最有广泛使用的模式。

yarn模式是不需要启动服务端守护进程的！

## 组件介绍：

Cluster Manager：在Standalone模式中即为Master（主节点），控制整个集群，监控Worker。

在YARN模式中为资源管理器。

Worker：从节点，负责控制计算节点，启动Executor或Driver。在YARN模式中为NodeManager，负责计算节点的控制。

Driver：运行Application的main()函数并创建SparkContext。

Executor：执行器，在worker node上执行任务的组件、用于启动线程池运行任务。

SparkContext：整个应用的上下文，控制应用的生命周期，提交作业的入口。

RDD：Spark的基本计算单元，一组RDD可形成执行的有向无环图RDD Graph。

DAG Scheduler：实现将Spark作业分解成一到多个Stage，每个Stage根据RDD的Partition个数决定Task的个数，然后生成相应的Task set放到TaskScheduler（NodeManager中）中。

TaskScheduler：将任务（Task）分发给Executor执行。

Stage：一个Spark作业一般包含一到多个Stage。

Task：一个Stage包含一到多个Task，通过多个Task实现并行运行的功能。