Spark20190809韵方



Spark Core：包含Spark的基本功能；尤其是定义RDD的API、操作以及这两者上的动作。其他Spark的库都是构建在RDD和Spark Core之上的。

Spark SQL：提供通过Apache Hive的SQL变体Hive查询语言（HiveQL）与Spark进行交互的API。每个数据库表被当做一个RDD，Spark SQL查询被转换为Spark操作。Spark提供的sql形式的对接Hive、JDBC、HBase等各种数据渠道的API，用Java开发人员的思想来讲就是面向接口、解耦合，ORMapping、Spring Cloud Stream等都是类似的思想。

Spark Streaming：基于SparkCore实现的可扩展、高吞吐、高可靠性的实时数据流处理。支持从Kafka、Flume等数据源处理后存储到HDFS、DataBase、Dashboard中。对实时数据流进行处理和控制。Spark Streaming允许程序能够像普通RDD一样处理实时数据。

MLlib：一个常用机器学习算法库，算法被实现为对RDD的Spark操作。这个库包含可扩展的学习算法，比如分类、回归等需要对大量数据集进行迭代的操作。

GraphX：控制图、并行图操作和计算的一组算法和工具的集合。GraphX扩展了RDD API，包含控制图、创建子图、访问路径上所有顶点的操作



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负责分配资源，有点像YARN中ResourceManager那个角色，大管家握有所有的干活的资源，属于乙方的总包。

WorkerNode是可以干活的节点，听大管家ClusterManager差遣，是真正有资源干活的主。从节点，负责控制计算节点，启动Executor或者Driver。

Executor是在WorkerNode上起的一个进程，相当于一个包工头，负责准备Task环境和执行

Task，负责内存和磁盘的使用。Task是施工项目里的每一个具体的任务。

Driver是统管Task的产生与发送给Executor的，运行Application 的main()函数，是甲方的司令员。

SparkContext是与ClusterManager打交道的，负责给钱申请资源的，是甲方的接口人