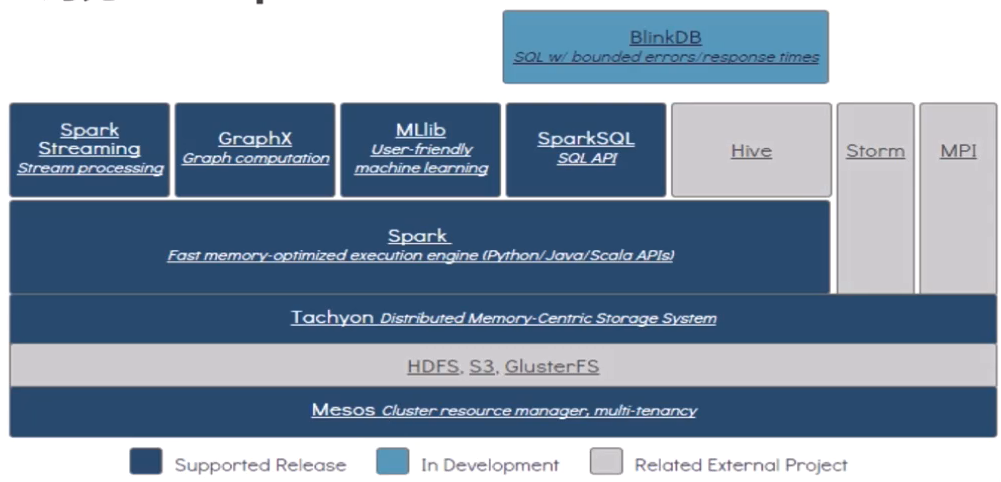
分布式计算框架Spark特点：计算快，通用，开发容易，可以访问任何数据库以及HDFS分布式文件系统

Spark生态系统BDAS（Berkeley Data Analysis Stacks）：



Mesos：集群资源调度管理框架类似YARN，支持多租户

Tachyon：基于内存的分布式文件系统，HDFS是基于磁盘的分布式文件系统

Spark：基于内存优化的快速执行引擎

Spark Streaming：基于内存优化的快速执行流式处理的引擎框架

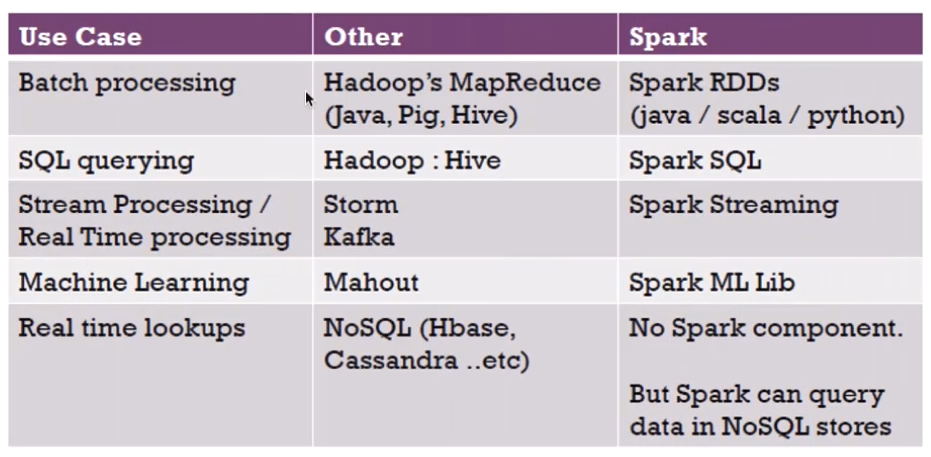
GraphX：图形计算框架

MLlib：机器学习框架

SparkSQL：SQL语言API

BlinkDB：在海量式数据上进行交互式快速查询引擎，以牺牲数据精准度提升查询响应时间的引擎

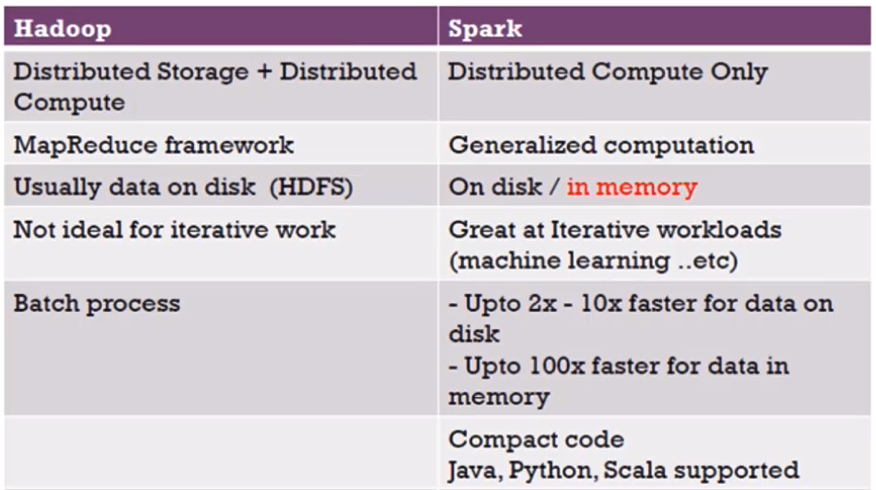
Hadoop生态圈与Spark BDAS对比



|  |  |  |
| --- | --- | --- |
| 场景 | Hadoop | Spark |
| 批处理 | Hadoop’s MapReduce(Java,Pig,Hive) | Spark RDDs(Java/Scala/Python) |
| SQL查询 | Hadoop:Hive | Spark SQL |
| 流处理/实时处理 | Storm Kafka | Spark Streaming |
| 机器学习 | Mahout | Spark ML Lib |
| 实时数据查找 | NoSQL(Hbase,Cassandra…etc) | Spark是计算引擎不关注数据存储在哪里，所有没有相应的Spark组件与实时数据查找对应，但Spark可以在非关系型数据库中查询数据 |

Hadoop与Spark对比

|  |  |
| --- | --- |
| Hadoop | Spark |
| 分布式存储系统，分布式计算框架，资源调度框架 | 只关注分布式系统的计算 |
| 使用MapReduce框架 | 通用计算框架 |
| 数据存储在磁盘（HDFS） | 磁盘/内存（内存中效率更高） |
| 不适合迭代次数多的工作，每次完成的数据需要落地即存到HDFS或者其他存储位置上 | 擅长迭代次数多的工作，因为基于内存计算所以数据不需要落地，例如机器学习 |
| 批处理框架 | 可以批处理，在磁盘上比Hadoop快2到10倍，在内存上快100倍以上 |
|  | 支持多种语言Java,Python,Scala |



Spark开发语言支持Python，Scala，Java，R语言，Spark大部分代码是由Scala语言开发

Spark运行模式：YARN，Local，Standalone是Spark内置的运行模式，Mesos是Spark最初开始支持的运行模式

spark启动：./spark-shell --master local[2] 以本地模式2个线程启动Spark

spark实现WordCount：

val file = sc.textFile("file:///home/文件路径")读取文件

val a = file.flatMap(line => line.split(" "))将行按照空格拆分

val b = a.map(word => (word,1))为每个单词添加数字1

val c = b.reduceByKey(\_ + \_)按照Key相同的统计

sc.textFile("file:///home/hadoop/data/hello.txt").flatMap(line => line.split(" ")).map(word => (word,1)).reduceByKey(\_ + \_).collect Spark实现WordCount的代码合并

Spark默认webUI监控界面http://主机IP:4040