Apache Spark：承诺和面临的挑战

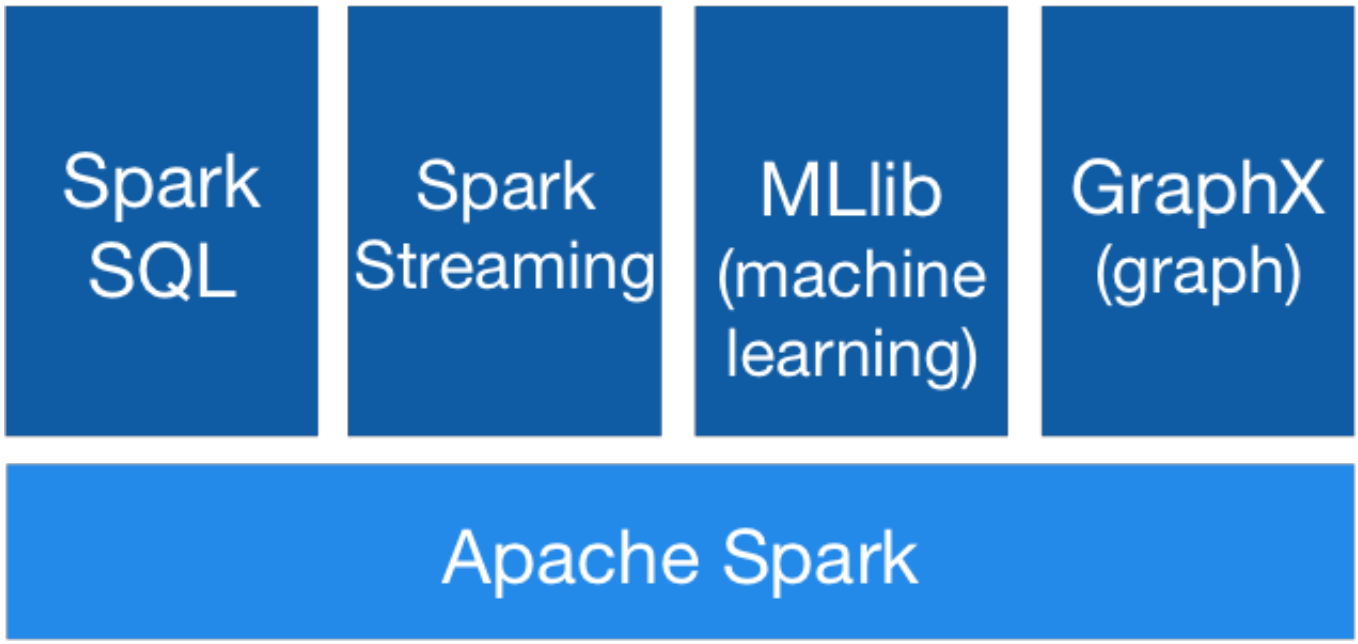
如果你要寻求一种处理海量数据的解决方案，就会有很多可选项。选择哪一种取决于具体的用例和要对数据进行何种操作，可以从很多种数据处理框架中进行遴选。例如Apache的Samza、Storm和Spark等等。本文将重点介绍Spark的功能，Spark不但非常适合用来对数据进行批处理，也非常适合对时实的流数据进行处理。

Spark目前已经非常成熟，数据处理工具包可以对大体量数据集进行处理，不必担心底层架构。工具包可以进行数据采集、查询、处理，还可以进行机器学习，进而构建出分布式系统的数据抽象模型。

处理速度也是Spark的亮点，MapReduce在处理过程中将数据放到内存中，而不放在磁盘上进行持久化，这种改进使得Spark的处理速度获得了提升。Spark提供了三种语言环境下的类库，即Scala、Java和Python语言。

除了上述这些优点之外，Spark自身也存在一些问题。例如，部署过程过于复杂，可扩展性差。本文对此也会进行论述。

Spark体系结构



如果想及时了
解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

上图显示了Spark所包含的不同功能模块。虽然这些模块的主要功能是处理流式数据，但还包括一些帮助执行各种数据操作的组件。

Spark
SQL
：Spark自带SQL接口，也就是说，可以使用SQL语句进行数据查询。查询操作会被Spark的执行引擎执行。
Spark
Streaming
：该模块提供了一组API，用来在编写应用程序的时候调用，执行对时实数据流的处理操作。该模块将进入的数据流拆分成微型批处理流，让应用程序进行处理。
MLib：该模块提供了在海量数据集上运行机器学习算法的一组API。
GraphX
：当处理由多个节点组成的图类型数据时，GraphX模块就派上用场了，主要的突出之处在于图形计算的内置算法。

除了用来对数据进行处理的类库之外，Spark还带有一个web图形用户接口。当运行Spark的应用时，通过4040端口会启动一个web界面，用来显示任务执行情况的统计数据和详细信息。我们还可以察看一个阶段任务执行的时间。如果想要获得最佳的性能，这样的信息是非常有帮助的。

用例

数据分析
——对进入的数据流作实时分析是Spark很在行的事情。Spark能够高效处理来自各式各样数据源的大量数据，支持HDFS、Kafka、Flume、Twitter和ZeroMQ，也能对自定义的数据源进行处理。

趋势数据
——Spark能够用来对进入的事件流进行处理，用于计算趋势数据。找到某个时间窗口的趋势，对于Spark来说变得异常简单。

物联网
——物联网系统产生了大量的数据。这些数据会被推往后台进行处理。Spark能够构建出数据管线，在特定的时间间隔（分钟、小时、周、月等等）内进行转换。还可以基于一组事件触发一系列动作。

机器学习
——由于Spark能够对线下数据进行批量处理，并且提供了机器学习类库（MLib），因而我们能够对数据集轻松地使用机器学习算法。另外，我们还可以在海量数据集中尝试各种不同的机器学习算法。把MLib与Streaming这两个库联合起来使用，就可以构建起机器学习系统。

Spark存在的问题

尽管Spark在较短的一段时间内就流行了起来，但是其自身也存在着一些问题。

复杂的部署过程

应用程序开发完毕后需要部署，对吗？这个时候有可能会出现难以适从的情况。虽然部署应用有多个可选项，但是最简单和直接的方式就是进行单独部署。Spark支持Mesos和Yarn，但如果

对这两者任何一个不熟悉的话，部署过程就会变得异常艰难。在绑定依赖关系的时候，也可能会遇到一些前期的坑坎儿。如果不能正确处理的话，Spark虽然会单独运行，但在cluster模式下，会遇到抛出Classpath异常的情况。

内存问题

由于Spark被用来处理海量数据，对内存的使用情况进行监控和度量就非常关键。在常见的使用范围内Spark完全没有问题，但针对不同的用例，要做非常多的配置工作。我们时常会受到所做的配置与用例不相配这样的限制。使用默认配置运行Spark应用并不是最佳选择，所以我们强烈建议你去查看相应的配置文档，对Spark内存相关的设置进行调整。

频繁的版本更新导致API发生变化

Spark以三个月为周期就要进行一次副版本（1.x.x）发布；每隔三到四个月，就要进行一次主版本（2.x.x）发布。尽管频繁的版本发布意味着较为迅速地推出了更多的功能特性，但这也意味着这些版本更迭的背后，某些情况下要求API也要发生变化。如果我们没有意识到新版本所带来的变化的话，就会由此陷入困境。而确保Spark应用不受这些变化影响，也会带来额外的开销。

对Python的支持不甚完善

Spark支持Scala、Java和Python语言。支持自己喜欢的语言当然是再好不过的事情了。但是Spark的最新版本中，对Python语言API的支持不像对Java和Scala语言的支持那样完善。Python类库需要一定时间完善功能，向最新版本的功能特性及API靠拢。如果打算使用Spark最新版本的话，可能需要用Scala或Java语言来实现，至少需要检查是否已经有Python版本功能或API的实现。

匮乏的文档

文档和指南，还有代码样例对新手成长来说至关重要。然而Spark的情况是，尽管在文档中有一些代码样例，但质量和深度都有待提高。文档中的样例都过于基础，无法给予程序员有效指导，完全发挥Spark应起的作用。

结语

Spark在构建数据处理应用方面可谓是了不起的框架。需要搞清楚的是在使用场景和数据规模方面不会出现“杀鸡焉用牛刀”的局面。如果你要处理小规模的数据，也许会有更简单的解决方案。对于Apache基金会的所有产品来说，了解其数据处理框架的所有细节和要点都是必需的，这样才能物尽其用。

本文转载自：http://geek.csdn.net/news/detail/136186
英文原文：Apache Spark: Promises and Challenges