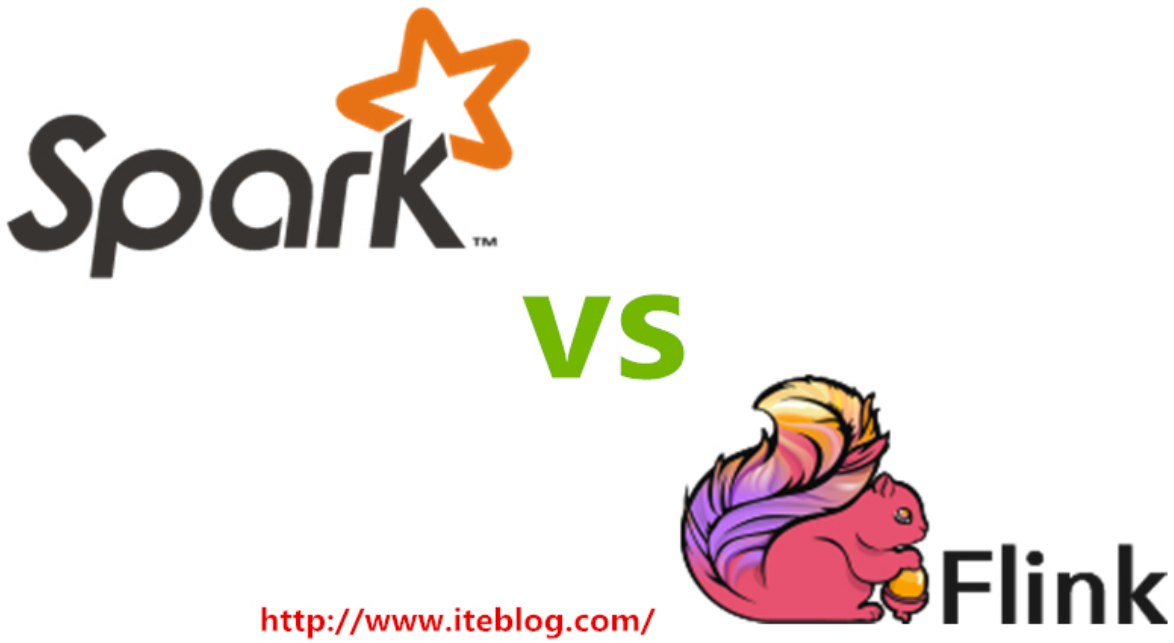
如何选择Apache Spark和Apache Flink

Spark Streaming和Flink都能提供恰好一次的保证，即每条记录都仅处理一次。与其他处理系统（比如Storm）相比，它们都能提供一个非常高的吞吐量。它们的容错开销也都非常低。之前，Spark提供了可配置的内存管理，而Flink提供了自动内存管理，但从1.6版本开始，Spark也提供了自动内存管理。这两个流处理引擎确实有许多相似之处，但它们也有着巨大的差异。近日，MapR Technologies产品经理Balaji Mohanam在公司内部的白板演示中比较了Apache Spark和Apache Flink的不同之处，用户可以参考这种比较做出选择。



为了方便说明，Mohanam首先对批处理、微批处理和连续流操作符等三种计算模式进行了解释。批处理基本上处理静态数据，一次读入大量数据进行处理并生成输出。微批处理结合了批处理和连续流操作符，将输入分成多个微批次进行处理。从根本上讲，微批处理是一个“收集然后处理”的计算模型。连续流操作符则在数据到达时进行处理，没有任何数据收集或处理延迟。

Apache Spark和Apache Flink的主要差别就在于计算模型不同。Spark采用了微批处理模型，而Flink采用了基于操作符的连续流模型。因此，对Apache Spark和Apache Flink的选择实际上变成了计算模型的选择，而这种选择需要在延迟、吞吐量和可靠性等多个方面进行权衡。

随着数据处理能力的提高，企业开始认识到，信息的价值在数据产生的时候最高。他们希望在数据产生时处理数据，这就是说需要一个实时处理系统。但也不是所有情况都需要实时系统。Mohanam分别例举了一些适合微批处理或实时流处理的场景。比如有两个广告科技行业的场景：一个是聚合来自不同IP地址的不同IP请求，将IP归入黑名单或白名单；另一个是设法阻止一个黑名单IP的特定请求。前者使用微批处理就可以，而后者就需要实时流处理。再比如，在电信行

业，统计特定用户使用的带宽，微批处理可能是一个更高效的方案，而网络异常检测就需要实时流处理了。也有一些场景，微批处理和实时流处理都适用，如在IoT行业查看特定工业设备的使用情况。

本文转自：http://www.infoq.com/cn/news/2016/03/Apache-Spark-Apache-Flink-choose