1 Spark Streaming动手实战演示

1. 闪电般理解Spark Streaming 原理
2. 应用程序的数据流，2.监控设备 3.工业设备 4.电商（时刻监控用户的行为，在线的实时推荐） 5.社交网络 6.搜索引擎 7.金融领域（合法的交易)
3. 所有都不经过流处理的数据是没有价值的数据

storm目前已经趋于稳定了，没有太大的发展了

去年就是流处理的一年

Spark Streaming与storm之间的选择

1. 如果毫秒级实时的处理（例如火灾报警时）选择storm
2. 如果不介意1分钟的响应，优先选择spark Streaming

Spark Streaming的优势:1.高层API非常适合编程，高度抽象的API，非常适合业务逻辑（机器学习，图计算）来自于微博上面的数据，电商方面的数据

2.杀手级的优势，数据流进来，可以很方便的调用spark sql,也可以调用spark core的内容

Storm是一条一条的处理，Spark Streaming是基于时间单位的，他跟内容无关，计算的时间是1s钟，可以用机器模型进行计算

当数据丢失，稳定，处理速度都不是问题的话

storm是一条一条的处理的，而streaming是基于时间单位进行处理的，跟内容无关，只跟时间有关，即使没有数据，也会启动一个Job

Spark Streaming也可以基于一条一条的数据，一般和kafka来配合处理，让kafka来控制数据

Dstream的程序本身不会产生Job,因为他不是RDD，框架定时会产生一个Job,要产生Job就必须有RDD,action操作

你写业务逻辑的时候，你想的是这1s钟该怎么处理数据，而且你处理数据是按照对RDD处理的角度对其进行处理的，但是我们的Spark Streaming了写的业务逻辑代码又不是基于RDD,而实际上由于时间间隔是1秒钟的话，那每隔一秒钟，基于RDD触发一个job,也就是说每一秒钟，都基于Spark Streaming写的代码产生的job的实例，而job的实例之所以产生，是因为有RDD的产生，所以你写的代码是job的模板，而这个模板在固定时间间隔内产生一个实例，时间一运动，就开始实例化

1. Spark Stream抓到代码先分析再运行，Spark Stream的框架不是RDD，因为RDD有个特征是数据不变性，Spark框架本身运行的时候，只认RDD.框架定时执行，生成一个job，job本身会使用RDD,要产生作业必须有RDD的操作。我们写好业务逻辑的计算。其实我们写的代码是RDD,job的模版。
2. RDD的模版是DStream,RDD之间是有依赖关系，Dstream之间也是有依赖关系的

也就是说你写的代码是Job的模板，在特定的时间内会产生实例，就是类和ojbect的关系

RDD的模板就是DStream,Dstream都会产生RDD,RDD之间有依赖关系，那么Dstream也有依赖关系，就构成了Dstream Grouph,这个Dstream Grouph的有向无环图就是job的有向无环图。只是根据流式处理的要求，在RDD上面做了一层封装而已，Dstream是一个没有状态的流，上一次和下一次没有关系

框架之所以能够产生作业是由时间来决定的，你写的代码无法产生job,只有框架才能产生job,框架产生Job的方式是通过时间来驱动的

补充:

流(Streaming),在大数据时代为数据流处理，就像水流一样，是数据流；既然是数据流处理，就会想到数据的流入，加工，数据的流出。

日常工作，生活中数据来源很多不同的地方。例如：工业时代的汽车制造，监控设备，工业设备会产生很多源数据；信息时代的电商网站，日志服务器，社交网络，金融交易系统，黑客攻击，垃圾邮件，交通监控等；通信时代的手机，平板，智能设备，物联网等会产生很多实时书u，数据流无处不在。

在大数据时代Spark Streaming能做什么？

平时用户都有网上购物的经历，用户在网站上进行的各种操作，通过Spark Streaming流处理技术可以被监控，用户的购买爱好，关注度，交易等可以进行行为分析。在金融领域，通过Spark Streaming流处理技术可以对交易量很大的账号进行监控，防止罪犯洗钱，财产转移，防欺诈等。在网络安全方面，黑客攻击时有发生，通过Spark Streaming流处理技术可以将某类可疑IP进行监控并结合机器学习训练模型匹配出当前请求是否属于黑客攻击。其他方面如：垃圾邮件监控过滤，交通监控，网络监控，工业设备监控的背后都是Spark streaming发挥强大流处理的地方。

大数据时代，数据价值一般怎么定义?

所有没经过流处理的数据都是无效数据或没有价值的数据；数据产生之后立即产生的价值是最大的，数据放置越久或滞后其使用价值越低。以前绝大多数电商网站盈利走的是网络流量（即用户的访问量），如今：电商网站不仅仅需要关注流量，交易量，更重要的是要通过数据流技术让电商网站的各种数据流动气啦，通过实时流动的数据及时分析，挖掘出各种有价值的数据；比如：对不同交易量的用户指定用户画像，从而提供不同服务质量；准对用户访问电商网站板块爱好及时推荐相关信息。

Spark Streaming vs Hadoop MR:

SparkStreaming是一个准实时的流处理框架，而Hadoop MR是一个离线，批处理框架，很显然，在数据的价值性角度，Spark Streaming完胜于Hadoop MR.

Spark Streaming VS Storm

Spark Streaming是一个准实时流处理框架，处理相应时间一般以分钟为单位，也就是说处理实时数据的延迟时间是秒级别的；Storm是一个实时流处理框架，处理响应式毫秒级的。所以在流框架选型方面要看具体业务场景。需要澄清的是现在很多人认为Spark Streaming流处理运行不稳定，数据丢失，事务性 支持不好等等，那是因为很多人不会驾驭Spark Streaming及Spark本身。在Spark Streaming流处理的延迟时间方面。

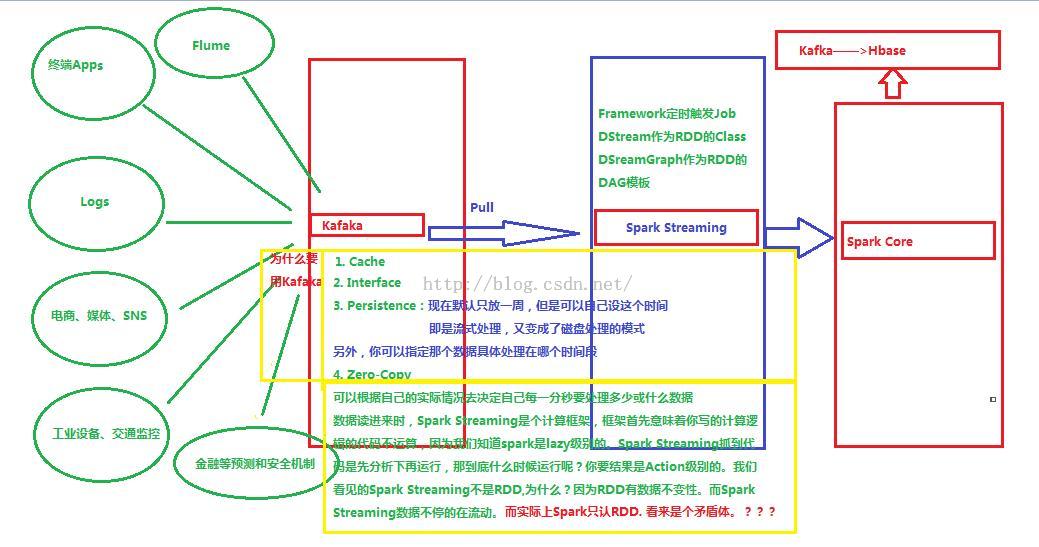
Spark Streaming特点：

1. 提供了丰富的API,企业中能快速实现各种复杂的业务逻辑。
2. 流入Spark Streaming的数据流通过和机器学习算法结合，完成机器模拟和图计算
3. Spark Streaming基于Spark 优秀的血统

SparkStreaming能不能像Storm一样，一条一条处理数据？

Storm处理数据的方式是以条为单位来一条一条处理的，而Spark Streaming基于单位时间处理数据的，Spark Streaming能不能像Storm一样了？答案是：可以的。

业界一般的做法是Spark Streaming和Kafaka搭档即可以达到这种效果



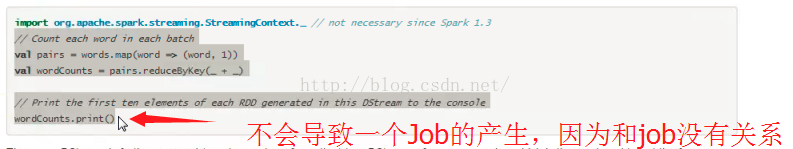
Kafka业界认同最主流的分布式消息框架，此框架即符合消息广播模式又符合消息队列模式。

kafka内部使用的技术:

1. Cache
2. Interface
3. Persistence:默认最大持久化一周
4. Zero-copy技术让Kafka每秒吞吐量几百兆，而且数据只需要加载一次到内核提供其他应用程序使用

外部各种数据源推进（push)Kafka，然后通过Spark Streaming抓取(pull)数据，抓取的数据量可以根据自己的实际情况确定每一秒钟要处理多少数据

通过Spark Streaming动手实战



但是实际上，Job的执行时Spark Streaming框架帮我们产生和开发者自己写的Spark代码业务逻辑没有关系，而且Spark Streaming框架的执行时间间隔可以手动设置，如：每隔一秒钟就会产生一次Job的调用。所以在开发者编写好的Spark代码时（如：flatMap,map,collect)，不会导致job的运行（如:flatMap,map,collect)，不会导致job的运行，job运行是Spark Streaming框架产生的，可以配置成每隔一秒钟都会产生一次job调用。

Spark Streaming流进来的数据是Dstream,但是Spark core框架只认RDD，这就产生矛盾了？

Spark Streaming框架中，作业实例的产生都是基于rdd实例来产生，你写的代码是作业的模板，模板一运行rdd就会被执行，此时action必须处理数据。RDD的模板就是Dstream离散流，RDD之间存在依赖关系，Dstream就有了依赖关系，也就构成了Dstream有向无环图。这个DAG图，是模板。Spark Streaming只不过是附在RDD上面的一层薄薄的封装而已，你写的代码不能产生Job,只有框架才能产生Job,如果一秒内计算不完数据，就只能调优了。

总结:

使用Spark Streaming可以处理各种数据来源类型，如：数据库，HDFS,服务器log日志，网络流，其强大超越了你想象不到的场景，只是很多时候大家不会用，其真正原因是对Spark,Spark streaming本身不了解。

Spark Streaming的劣势:不稳定，耗时，数据丢失，事物丢失