什么是实时流计算

所谓实时流计算，就是近几年由于数据得到广泛应用之后，在数据持久性建模不满足现状的情况下，急需数据流的瞬时建模或者计算处理。在这种实时计算的应用实例有金融服务、网络监控、电信数据管理、Web应用、生产制造、传感检测，等等。在这种数据流模型中，单独的数据单元可能是相关的元组（Tuple），如网络测量、呼叫记录、网页访问等产生的数据。但是，这些数据以大量、快速、时变（可能是不可预知）的数据流持续到达，由此产生了一些基础性的新的研究问题 —— 实时计算。实时计算的一个重要方向就是实时流计算。

1. 实时流计算背景

数据的价值随着时间的流逝而降低，所以事件出现后必须尽快对它们进行处理，最好事件出现时便立刻对其进行处理，发生一个事件进行一次处理，而不是缓存起来成一批处理。例如商用搜索引擎，像Google、Bing和Yahoo!等，通常在用户查询响应中提供结构化的Web结果，同时也插入基于流量的点击付费模式的文本广告。为了在页面上的最佳位置展现最相关的广告，通过一些算法来动态估算给定上下文中一个广告被点击的可能性。上下文可能包括用户偏好、地理位置、历史查询、历史点击等信息。一个主搜索引擎可能每秒钟处理成千上万次查询，每个页面都可能会包含多个广告。为了几十处理用户反馈，需要一个低延迟、可扩展、高可靠的处理引擎。

对于这些实时性要求很高的应用，若把持续到达的数据简单地放到传统数据库管理系统DBMS中，并在其中进行操作，是不切实际的。传统的DBMS并不是为快速连续地存放单个的数据单元而设计的，而且也不支持“持续处理”，而“持续处理”是数据流应用的典型特征。另外，现在人们都认识到，“近似性”和“自适应性”是对数据流进行快速查询和其它处理（如数据分析和数据采集）的关键要素，而传统DBMS的主要目标恰恰与之相反：通过稳定的查询设计，得到精确的答案。

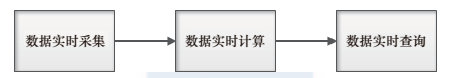
另外一些方案是采用MapReduce来处理实时数据流。但是，尽管MapReduce做了实时性改进，也很难稳定地满足应用需求。这是因为Hadoop MapReduce框架为批处理做了高度优化，典型的是通过调度批量任务来操作静态数据，任务不是常驻服务，数据也不是实时流入；而数据流计算的典型范式之一是不确定数据速率的事件流流入系统，系统处理能力必须与事件流量匹配。

1. 实时计算应用场景

互联网领域的实时流计算一般都是针对海量数据进行的，除了非实时计算的需求（如计算结果准确）以外，实时计算最重要的一个需求是能够实时响应计算结果，一般要求为秒级。个人理解，互联网行业的实时计算可以分为以下两种应用场景。

1. 数据源是实时的、不间断的，要求对用户的响应时间也是实时的。主要用于互联网流式数据处理。所谓流式数据，是指将数据看作数据流的形式来处理。数据流则是在时间分布和数量上无限的一系列数据记录的集合体；数据记录是数据流的最小组成单元。例如，对于大型网站，活跃的流式数据非常常见，这些数据包括网站的访问PV/UV、用户访问的内容、搜索的内容等。实时的数据计算和分析可以动态实时地刷新用户访问数据，展示网站实时流量的变化情况，分析每天各小时的流量和用户分布情况，这对于大型网站来说具有重要的实际意义。
2. 数据量大且无法或没必要预算，但要求对用户的响应时间是实时的。主要用于特定场合下的数据分析处理。当数据量很大，同时发现无法穷举所有可能条件的查询组合或者大量穷举出来的条件组合无用时，实时计算就可以发挥作用，将计算过程推迟到查询阶段进行，但需要为用户提供实时响应。
3. 实时计算处理流程

互联网上海量数据（一般为日志流）的实时计算过程可以划分为3个阶段：数据的产生与收集阶段、传输与分析处理阶段、存储对外提供服务阶段，下面分别进行简单介绍。



1. 数据实时采集

需求：功能上保证可以完整地收集到所有日志数据，为实时应用提供实时数据；响应时间上要保证实时性、低延迟（在1s左右）；配置简单，部署容易；系统稳定可靠等。

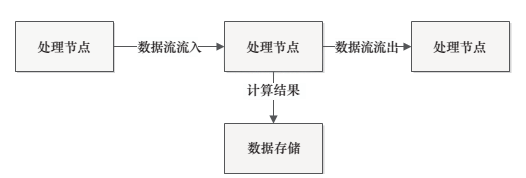
目前，互联网企业的海量数据采集工具有Facebook开源的Scribe、LinkedIn开源的Kafka、Cloudera开源的Flume，淘宝开源的TimeTunnel、Hadoop的Chukwa等，它们均可以满足每秒数百MB的日志数据采集和传输需求。

1. 数据实时计算

传统的数据操作，首先是将数据采集并存储在DBMS中，然后通过查询和DBMS进行交互，得到用户想要的答案。在整个过程中，用户是主动的，而DBMS系统是被动的，过程操作如图所示。



但是，对于现在大量存在的实时数据，如股票交易的数据，这类数据实时性强，数据量大，没有止境，传统的架构并不合适。流计算就是专门针对这种数据类型准备的。在流数据不断变化的运动过程中实时地进行分析，捕捉到可能对用户有用的信息，并把结果发送出去。在整个过程中，数据分析处理系统是主动的，而用户却处于被动接收的状态，处理流程如图所示。



需求：适应流式数据、不间断查询；系统稳定可靠、可扩展性好、可维护性好等。有关计算的一些注意点：分布式计算、并行计算（节点间的并行、节点内的并行）、热点数据的缓存策略、服务端计算。

1. 实时查询服务

全内存：直接提供数据读取服务，定期转存到磁盘或数据库进行持久化。

半内存：使用Redis、Memcache、MongoDB、BerkeleyDB等内存数据库提供数据实时查询服务，由这些系统进行持久化操作。

全磁盘：使用HBase等分布式文件系统（HDFS）为基础的NoSQL数据库，对于Key-Value内存引擎，关键是设计好Key的分布。

1. 实时计算框架

最近这几年随着实时计算的流行，相继出现了以下实时计算框架。

1. IBM的StreamBase
2. Yahoo的S42
3. Twitter的Storm
4. Twitter的Rainbird
5. Facebool的Puma
6. 阿里的JStorm
7. 其他实时计算系统
   * 1. HStreaming
     2. Esper
     3. Borealis

框架对比