06 | 如何区分批处理还是流处理?

2019-04-29 蔡元楠 来自北京

《大规模数据处理实战》

你好，我是蔡元楠。

今天，我将会带领你一起学习在进行大规模数据处理时，无论如何也绕不开的两个处理模式：批处理（Batching Processing）和流处理（Streaming Processing）。

在我看来，大规模的视频流系统、大规模物联网（IoT）数据监控系统等各种现代大规模数据系统的出现，已经成为了一种必然的历史潮流。

无论你是在从事哪一种开发方向，都不可避免地要与这些海量数据打交道。如何能既满足实际应用场景的需求，又高效地处理好大规模数据，在整个项目开发架构中都是非常重要的一个环节。

在开始讲解批处理和流处理之前，我想先介绍一下几个必要的背景知识。

无边界数据和有边界数据

这个世界上的数据可以抽象成为两种，分别是无边界数据（Unbounded Data）和有边界数据（Bounded Data）。

顾名思义，**无边界数据**是一种不断增长，可以说是无限的数据集。

这种类型的数据，我们无法判定它们到底什么时候会停止发送。

例如，从手机或者从传感器发送出来的信号数据，又比如我们所熟知的移动支付领域中的交易数据。因为每时每刻都会有交易产生，所以我们不能判定在某一刻这类数据就会停止发送了。

在国外的一些技术文章上，有时候我们会看到“流数据（Streaming Data）”这一说法，其实它和无边界数据表达的是同一个概念。

与此相反，**有边界数据**是一种有限的数据集。

这种数据更常见于已经保存好了的数据中。例如，数据库中的数据，或者是我们常见的 CSV 格式文件中的数据。

当然了，你可能会问，那我们把无边界数据按照时间窗口提取一小份出来，那这样的数据是什么数据呢？

拿我们之前提到过的移动支付中的交易数据来说吧。移动支付中的交易数据可以看作是无边界数据。那我们按 2019 年 4 月 29 日这个时间窗口提取出来的数据呢？这个当日的交易数据就变成了有边界数据了。

所以，有边界数据其实可以看作是无边界数据的一个子集。

事件时间和处理时间

在处理大规模数据的时候，我们通常还会关心**时域**（Time Domain）的问题。

我们要处理的任意数据都会有两种时域，分别是事件时间（Event Time）和处理时间（Precessing Time）。

事件时间指的是一个数据实际产生的时间点，而**处理时间**指的是处理数据的系统架构实际接收到这个数据的时间点。

下面我来用一个实际的例子进一步说明这两个时间概念。

现在假设，你正在去往地下停车场的路上，并且打算用手机点一份外卖。选好了外卖后，你就用在线支付功能付款了，这个时候是 12 点 05 分。恰好这时，你走进了地下停车库，而这里并没有手机信号。因此外卖的在线支付并没有立刻成功，而支付系统一直在重试（Retry）“支付”这个操作。

当你找到自己的车并且开出地下停车场的时候，已经是 12 点 15 分了。这个时候手机重新有了信号，手机上的支付数据成功发到了外卖在线支付系统，支付完成。

在上面这个场景中你可以看到，支付数据的事件时间是 12 点 05 分，而支付数据的处理时间是 12 点 15 分。事件时间和处理时间的概念，你明白了吗？

在了解完上面的 4 个基本概念后，我将开始为你揭开批处理和流处理模式的面纱。

批处理

数据的批处理，可以理解为一系列相关联的任务按顺序（或并行）一个接一个地执行。批处理的输入是在一段时间内已经收集保存好的数据。每次批处理所产生的输出也可以作为下一次批处理的输入。

绝大部分情况下，批处理的输入数据都是**有边界数据**，同样的，输出结果也一样是**有边界数据**。所以在批处理中，我们所关心的更多会是数据的**事件时间**。

举个例子，你在每年年初所看到的“支付宝年账单”就是一个数据批处理的典型例子。

支付宝会将我们在过去一年中的消费数据存储起来，并作为批处理输入，提取出过去一年中产生交易的事件时间，然后经过一系列业务逻辑处理，得到各种有趣的信息作为输出。

在许多情况下，批处理任务会被安排，并以预先定义好的时间间隔来运行，例如一天，一个月或者是一年这样的特定时间。

在银行系统中，银行信用卡消费账单和最低还款额度也都是由批处理系统以预先定义好的一个月的时间间隔运行，所产生出来的。

批处理架构通常会被设计在以下这些应用场景中：

- 日志分析：日志系统是在一定时间段（日，周或年）内收集的，而日志的数据处理分析是在不同的时间内执行，以得出有关系统的一些关键性能指标。
- 计费应用程序：计费应用程序会计算出一段时间内一项服务的使用程度，并生成计费信息，例如银行在每个月末生成的信用卡还款单。
- 数据仓库：数据仓库的主要目标是根据收集好的数据事件时间，将数据信息合并为静态快照（static snapshot），并将它们聚合为每周、每月、每季度的报告等。

由 Google MapReduce 衍生出来的开源项目 Apache Hadoop 或者是 Apache Spark 等开源架构都是支持这种大数据批处理架构的。

由于完成批处理任务具有高延迟性，一般可以需要花费几小时，几天甚至是几周的时间。要是在开发业务中有快速响应用户的时间需求，我们则需要考虑使用流处理 / 实时处理来处理大数据。

流处理

数据的流处理可以理解为系统需要接收并处理一系列连续不断变化的数据。例如，旅行预订系统，处理社交媒体更新信息的有关系统等等。

流处理的输入数据基本上都是**无边界数据**。而流处理系统中是关心数据的事件时间还是处理时间，将视具体的应用场景而定。

例如，像网页监控系统这样的流处理系统要计算网站的 QPS，它所关心的更多是**处理时间**，也就是网页请求数据被监控系统接收到的时间，从而计算 QPS。

而在一些医疗护理监控系统的流处理系统中，他们则更关心数据的**事件时间**，这种系统不会因为接收到的数据有网络延时，而忽略数据本来产生的时间。

流处理的特点应该是要足够快、低延时，以便能够处理来自各种数据源的大规模数据。流处理所需的响应时间更应该以毫秒（或微秒）来进行计算。像我们平时用到的搜索引擎，系统必须在用户输入关键字后以毫秒级的延时返回搜索结果给用户。

流处理速度如此之快的根本原因是因为它在数据到达磁盘之前就对其进行了分析。

当流处理架构拥有在一定时间间隔（毫秒）内产生逻辑上正确的结果时，这种架构可以被定义为**实时处理**（Real-time Processing）。

而如果一个系统架构可以接受以分钟为单位的数据处理时间延时，我们也可以把它定义为**准实时处理**（Near real-time Processing）。

还记得我们在介绍批处理架构中所说到的不足吗？没错，是高延迟。而流处理架构则恰恰拥有高吞度量和低延迟等特点。

流处理架构通常都会被设计在以下这些应用场景中：

实时监控：捕获和分析各种来源发布的数据，如传感器，新闻源，点击网页等。

实时商业智能：智能汽车，智能家居，智能病人护理等。

销售终端（POS）系统：像是股票价格的更新，允许用户实时完成付款的系统等。

在如今的开源架构生态圈中，如 Apache Kafka、Apache Flink、Apache Storm、Apache Samza 等，都是流行的流处理架构平台。

在介绍完这两种处理模式后，你会发现，无论是批处理模式还是流处理模式，在现实生活中都有着很广泛的应用。你应该根据自己所面临的实际场景来决定到底采用哪种数据处理模式。

小结

批处理模式在不需要实时分析结果的情况下是一种很好的选择。尤其当业务逻辑需要处理大量的数据以挖掘更为深层次数据信息的时候。

而在应用需求需要对数据进行实时分析处理时，或者说当有些数据是永无止境的事件流时（例如传感器发送回来的数据时），我们就可以选择用流处理模式。

思考题

相信在学习完这一讲后，你会对批处理模式和流处理模式有着清晰的认识。今天的思考题是，在你的日常开发中，所面临的数据处理模式又是哪一种模式呢？

欢迎你把答案写在留言区，与我和其他同学一起讨论。如果你觉得有所收获，也欢迎把文章分享给你的朋友。

精选留言 (40)

孙稚昊
2019-04-29

我们的用户画像本质还是批处理，还不能做到实时更新每个人的 profile，但对用户的每次电机有一个实时的劣化推荐版本，就是根据session中点的几个item的click，找到它们的simiiar item，这个是通过cache 和API实现的，并不是实时数据处理

作者回复: 谢谢你的分享！

13

hua168
2019-04-30

老师，上面说流数据是在没到达磁盘之前就处理了，所以速度很快，但是如果那处软件挂了，那部分流数据不是丢失了吗？是不是不能处理重要的数据？

如果我的数据很重要，但是又想像流那样处理的快速怎办？像redis那样，使用持久化，边处理写延迟写及磁盘这种处理思想吗？还是其它？

作者回复: 谢谢你的提问！数据如果没有保存到磁盘的话，确实整个软件挂了所有数据就丢失了。不过流处理一样可以处理重要数据的。一般即使数据存在内存中，有的软件会定时将数据的snapshot保存到磁盘中，以防软件全部挂掉。而很多软件都会有data replica，而且会有N+1或者N+2的policy，以此来保证如果有其中一台机器上的软件挂了，另外一台机器可以顶替它。

一般全部机器都挂的情况非常少见，这就如同存在磁盘上的数据被人运行"rm -fR /"一样，所以在采用流处理的时候不必过于担心。

9

xzy
2019-04-30

既有批处理也有流处理，生产环境利用elasticsearch来存储监控数据、日志数据等。为了降低成本和查询速度，会按照小时、天粒度对历史数据做预聚合，这应该属于批处理。其次，es作为搜索引擎，用户也有实时查询的需求，这块应该属于流处理。 谢谢

作者回复: 谢谢你的分享！

共 3 条评论>

9

yangs
2019-04-29

老师您好，之前看到网上说flink实现的流处理和spark streaming不一样，是因为spark使用了微批处理模拟流处理，可是我觉得flink实现的原理也像是用批处理模拟流处理，将一段一段数据包裹在时间窗口里来实现，这个时间窗口的数据处理，可不可以也理解成为是批处理?

作者回复: 可以

共 3 条评论>

8

mini希
2019-04-29

数仓有没有准实时的解决方案呢?

共 1 条评论>

7

邱从贤※klion26
2019-04-29

有限流是无限流的一个特例，所以一直在想是不是未来不再需要批处理，所有的都可以流处理，从而达到真正的流批一体。

从现在的情况看，批处理主要用于分析，用 sql 较多，且会对多个表进行处理，是不是意味着流上的 sql 也是刚需。

线下批处理能够不停重算的特性，应该可以让流处理不停做 checkpoint 来支持，这样是不是就和 db 的 backup 就有点像了，那是不是最后流处理，批处理，数据库也会统一起来呢?

作者回复: 谢谢你的留言！我很认同你的观点，关于流处理和批处理未来应该会统一起来。数据库作为存储系统的话还是会单独存在的吧。

共 3 条评论 >

4

Fiery
2020-03-08

"流处理架构则恰恰拥有高吞度（吐）量和低延迟等特点"，关于这一句有点疑惑请老师解答，我之前一直把Throughput和Latency当做互斥的一组指标，一般来讲高吞吐的系统都会选择牺牲响应速率（即低延迟），而如果专注提供低延迟响应，那一般吞吐量都到达不了系统的peak capacity。比如同样的一组集群，同样的数据量，如果不考虑其它影响处理效率的问题，那么集群进行批处理作业时的吞吐量应该是一定会超过做流处理作业时的吞吐量的，不是吗？所以我觉得这句话难道不应该是“流处理更专注于低延迟的数据处理，而批处理更专注于高吞吐的数据处理”吗?

3

涵
2019-04-29

在实际工作中数据仓库的数据处理使用的是批处理，根据需要大多数据是日处理，个别数据是一天处理几次，但都是批处理。在做核心业务系统时使用的是流数据处理，通常用消息中间件来传递事件，接收到事件时即开始处理。一直想尝试的是通过日志信息抽取业务信息，实现对业务信息的实时分析，例如当日的实时交易笔数，交易额等，无需侵入核心业务系统，通过日志即可以流数据的形式实时传递给数据平台。了解过splunk,elasticsearch都可以做，但是不清楚哪个更好，或者有其他更好的选择。

作者回复: 谢谢你的经验分享！赞一个！

3

JohnT3e
2019-04-29

一般业务中都会涉及到实时处理和批处理的需求，现在采取的类似于Kappa的架构。

Kappa Architecture: http://milinda.pathirage.org/kappa-architecture.com/
Samba Architecture: http://lambda-architecture.net/

  2

高景洋
2021-09-22

在我们的业务中，流处理 和 批处理都有使用。

流处理
1、我们将数据，按特定频次调度到kafka中，比如说 2小时一次、6小时一次、1天一次...
2、有一个分发程序，将数据按特定频率,从kafka消费出来 （比如说，一共有120条数据，2小时要处理一次，那平均到1分钟的粒度，就是每分钟一条。实际业务中，每2小时要处理的数据量可能到2000W级）
3、第二步消费出来的数据，会推到redis 的一个队列中，进行后续的业务逻辑处理（为什么不用kafka，而用redis做队列组件呢？kakfa受分区数限制，而我们的业务逻辑程序，要求高并发1W+线程，kafka的分区数，限制了业务的线程数）
4、业务处理后的数据，又会重新推回新的 kafka,会有数据处理程序，对新的kafka 进行消费入库操作
5、形成闭环，这是业务中的 一个流处理的流程~

批处理
1、批处理我们用在了数据的汇总统计上
2、我们要对库中（hbase）,每天数据的新增量、更新量、各渠道的来源量做汇总统计，形成报表
3、我们会在每天凌晨，将hbase中的数据，导入到hive，由hive对各个维度的数据，进行汇总group by统计
4、统计结果入mysql，对外生成报表输出
5、这是一个批处理的流程

  1

柳年思水
2019-07-09

我个人也是比较赞同 DataFlow 模型的思想的，认为批是流的一个特例，未来的计算不会再明显区分到底是流还是批，但不能排除除一些特殊情况（毕竟当前的批计算引擎针对批的场景做

了大量的优化，通用系统的性能肯定是赶不上专用系统的），但计算不仅仅是批和流两种形态，还有复杂计算场景，比如现在的 TensorFlow（AI 框架的本质也是计算）、RAY 等，计算引擎最后会不会完全融合到一起呢？或者换个思路，一个引擎可以兼容所有的引擎（有点类似 Beam），在一个计算框架里，可以跑多个 runner（这个 runner 可以是不同的引擎），未来会不会是这样的呢？

1

小凡

2019-05-18

请问spring-batch和hadoop这类批处理框架有什么不同吗？还有spring data flow

共 1 条评论 >

1

slowforce

2019-05-09

我们接收现场设备发回来的数据，数据以email或者sftp的形式上传 或者以自定义的格式通过socket直接传。对于前一种情况，我们采取批处理的方式 定时去处理，而对于第二种情况 我认为就是流处理

作者回复: 谢谢你的分享!

1

越甲非甲

2019-05-07

目前我们做的流处理场景下的解决方案，都是控制较小时间窗口的批处理，通过累加类似的方案来实现近似流处理的效果。请问老师，流处理的更一般性的解决思路是什么样子的呢？是这种微批处理的路子吗？谢谢老师!

1

CoderLean

2019-05-04

Flink的本质就是流处理，而里面的批处理api底层是将时间或者个数设定在某个区域里面，可以认为在这个架构中批处理是流处理的一个特例，我看有的评论说反了。说明还没好好掌握flink

作者回复: 赞一个大牛的留言啊!

1

每天晒白牙
2019-04-29

产生特定格式和维度的报表数据一般是批处理，但实时报表是流处理，需要低延迟

作者回复: 谢谢你的分享！

1

peter
2019-04-29

老师在谈流处理框架时没有说spark，难道spark不是流处理框架吗？(spark streaming也是流处理呀)

作者回复: 谢谢你的留言！没错，Spark也是支持流处理的。现在数据处理的Framework太多了，可能没有面面俱到描述到。

1

徐李
2022-11-02 来自浙江

这一节学了 醍醐灌顶，之前对流数据的的概念理解的不是这么深刻，实际上就是无边界的数据，一直会产生的数据，而且还需要实时响应。
我们的业务中，有采集电表水表的数据，都是通过http直接处理的，这样在大数据量的时候就容易崩。
本来准备决定放到消息队列rabbitmq中，这样来看，大数据量的话，还是采用apache kafka 会比较好。

杨大伟
2022-10-04 来自江苏

我做的国家电网相关的项目，在计算电费，客户支付电费订单时的数据处理使用流处理，而每个月，每个季度的电费统计，使用了批处理

哇哈哈
2022-03-11

event time & process time是流式处理的重要概念，推荐大家一本《streaming system》，虽然没有中文版，但是写的也算比较好理解了