

## 10-Lambda架构：Twitter亿级实时数据分析架构背后的倚天剑

今天我要分享的主题是Lambda架构。

通过这一讲，你可以了解什么是Lambda架构，以及它为什么能够成为Twitter亿级实时数据分析架构背后的“倚天剑”。

在学习了架构师的必备技能后，你是否已经摩拳擦掌，跃跃欲试地想要上手一个实际项目了呢？没问题，我们一起来看一个我的架构经历里的真实项目。

情况是这样的，我们正运行着广告精准投放业务，并且拥有海量的用户网站访问行为。我们需要进行用户行为分析来建立一个模型，然后根据这个模型来投放用户喜好的广告。

你可能想到了批处理架构。没错，这个时候数据批处理架构无疑是一种很好的选择。

可是我们不要忘了，之前讲过批处理架构有着高延时性的不足，而互联网用户行为的数据往往可以达到Pb或Eb，甚至是Zb的级别。做这种分析挖掘用户行为的任务，往往能耗时好几个小时甚至是几天。这样的话，我们根据模型精准投放给特定用户的广告就会有一定延时了。

那我们只用流处理架构行不行呢？

在广告精准投放的业务需求下，只用流处理架构会造成忽略了用户的历史网站访问行为，一些异常行为可能会让我们的服务投放错误的广告。

例如，用户A的电脑暂时借给用户B使用了一下，而用户B浏览了一些新的网站类型（与用户A不同）。这种情况下，我们无法判断用户A实际上是否对这类型的广告感兴趣，所以不能根据这些新的浏览记录给用户A推送广告。

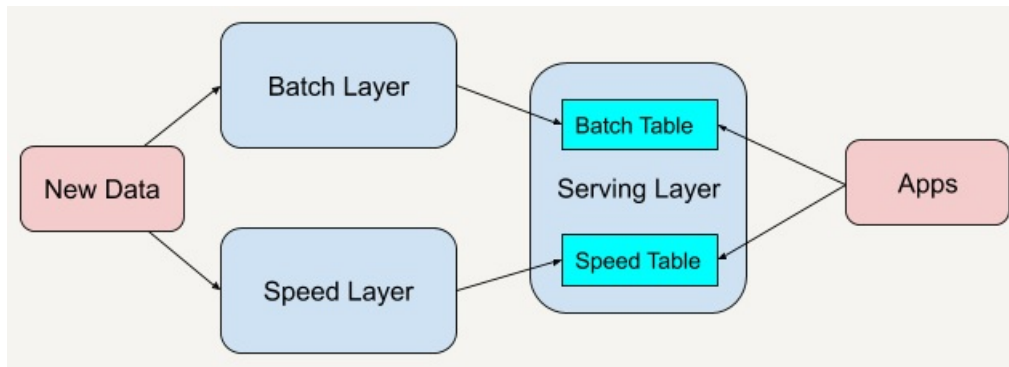
这个时候应该怎么优化我们的架构呢？我们先把问题放一放，在介绍完了Lambda架构之后或许会对你有所启发。

### Lambda架构

Lambda架构（Lambda Architecture）是由Twitter工程师南森·马茨（Nathan Marz）提出的大数据处理架构。这一架构的提出基于马茨在BackType和Twitter上的分布式数据处理系统的经验。

Lambda架构使开发人员能够构建大规模分布式数据处理系统。它具有很好的灵活性和可扩展性，也对硬件故障和人为失误有很好的容错性。

Lambda架构总共由三层系统组成：**批处理层**（Batch Layer），**速度处理层**（Speed Layer），以及用于响应查询的**服务层**（Serving Layer）。



在Lambda架构中，每层都有自己所肩负的任务。

批处理层存储管理主数据集（不可变的数据集）和预先批处理计算好的视图。

批处理层使用可处理大量数据的分布式处理系统预先计算结果。它通过处理所有的已有历史数据来实现数据的准确性。这意味着它是基于完整的数据集来重新计算的，能够修复任何错误，然后更新现有的数据视图。输出通常存储在只读数据库中，更新则完全取代现有的预先计算好的视图。

速度处理层会实时处理新来的大数据。

速度层通过提供最新数据的实时视图来最小化延迟。速度层所生成的数据视图可能不如批处理层最终生成的视图那样准确或完整，但它们几乎在收到数据后立即可用。而当同样的数据在批处理层处理完成后，在速度层的数据就可以被替代掉了。

本质上，速度层弥补了批处理层所导致的数据视图滞后。比如说，批处理层的每个任务都需要1个小时才能完成，而在这1个小时里，我们是无法获取批处理层中最新任务给出的数据视图的。而速度层因为能够实时处理数据给出结果，就弥补了这1个小时的滞后。

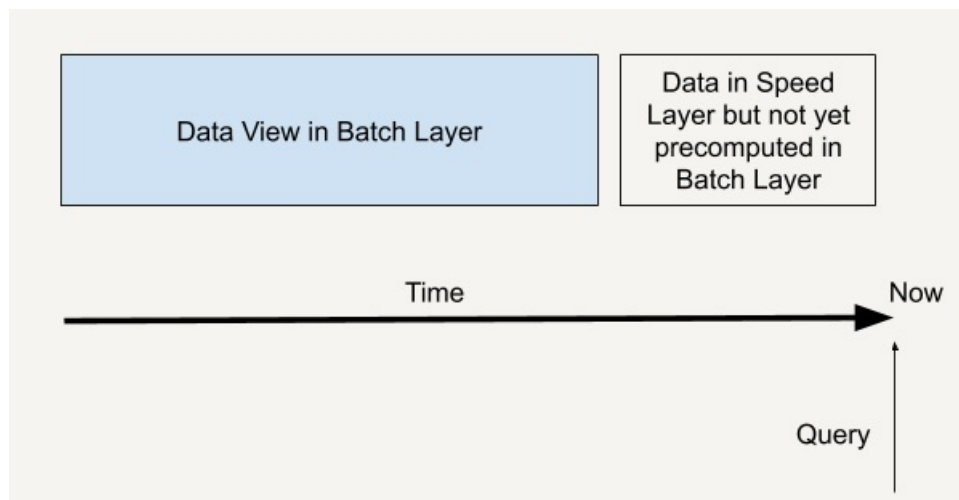
所有在批处理层和速度层处理完的结果都输出存储在服务层中，服务层通过返回预先计算的数据视图或从速度层处理构建好数据视图来响应查询。

好了，我们回到刚刚的问题中。我们如何做到既能实时分析用户新的网站浏览行为又能兼顾到用户的网站浏览行为历史呢？没错，就是利用Lambda架构。

所有的新用户行为数据都可以同时流入批处理层和速度层。批处理层会永久保存数据并且对数据进行预处理，得到我们想要的用户行为模型并写入服务层。而速度层也同时对新用户行为数据进行处理，得到实时的用户行为模型。

而当“应该对用户投放什么样的广告”作为一个查询（Query）来到时，我们从服务层既查询服务层中保存好的批处理输出模型，也对速度层中处理的实时行为进行查询，这样我们就可以得到一个完整的用户行为历史了。

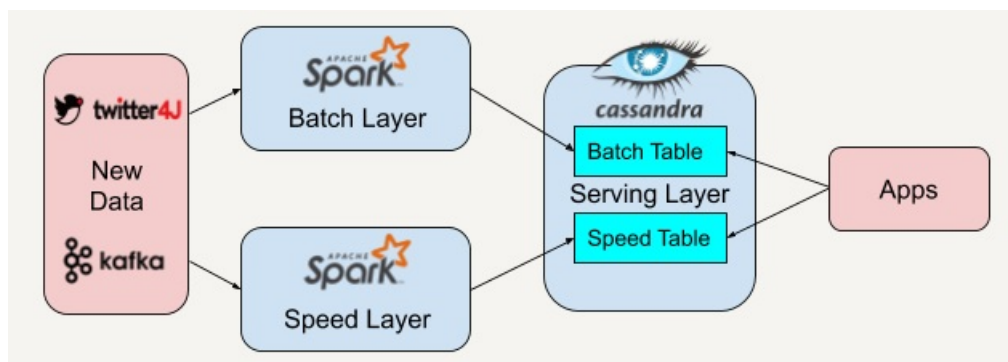
一个查询就如下图所示，既通过批处理层兼顾了数据的完整性，也可以通过速度层弥补批处理层的高延迟性，让整个查询具有实时性。



Lambda架构在硅谷一线大公司的应用已经十分广泛，我来带你一起看看一些实际的应用场景。

## Twitter的数据分析案例

Twitter在欧美十分受欢迎，而Twitter中人们所发Tweet里面的Hashtag也常常能引爆一些热搜词汇，也就是Most Popular Hashtags。下面我来给你讲述一下如何利用Lambda架构来实时分析这些Hashtags。



在这个实际案例里，我们先用twitter4j的流处理API抓取实时的Twitter推文，同时利用Apache Kafka将抓取到的数据保存并实时推送给批处理层和速度层。

因为Apache Spark平台中既有批处理架构也兼容了流处理架构，所以我们选择在批处理层和速度层都采用Apache Spark来读取来自Apache Kafka的数据。

批处理层和速度层在分析处理好数据后会将数据视图输出存储在服务层中，我们将使用Apache Cassandra平台来存储他们的数据视图。Apache Cassandra将批处理层的视图数据和速度层的实时视图数据结合起来，就可以得到一系列有趣的数据。

例如，我们根据每一条Tweet中元数据（Metadata）里的location field，可以得知发推文的人的所在地。而服务层中的逻辑可以根据这个地址信息进行分组，然后统计在不同地区的人所关心的Hashtag是什么。

时间长达几周或者的几个月的数据，我们可以结合批处理层和速度层的数据视图来得出，而快至几个小时的数据我们又可以根据速度层的数据视图来获知，怎么样？这个架构是不是十分灵活？

看到这里，你可能会问，我在上面所讲的例子都是来自些科技巨头公司，如果我在开发中面对的数据场景没有这么巨大，又或者说我的公司还在创业起步阶段，我是否可以用到Lambda架构呢？

答案是肯定的！我下面将和你一起分享一个在硅谷旧金山创业公司的App后台架构。

## Smart Parking案例分析

在硅谷旧金山地区上班生活的小伙伴肯定都知道，找停车位是一大难题。这里地少车多，每次出行，特别是周末，找停车位都要绕个好几十分钟才能找得到。

智能停车App就是在这样的背景下诞生的。这个App可以根据大规模数据所构建的视图推荐最近的车位给用户。

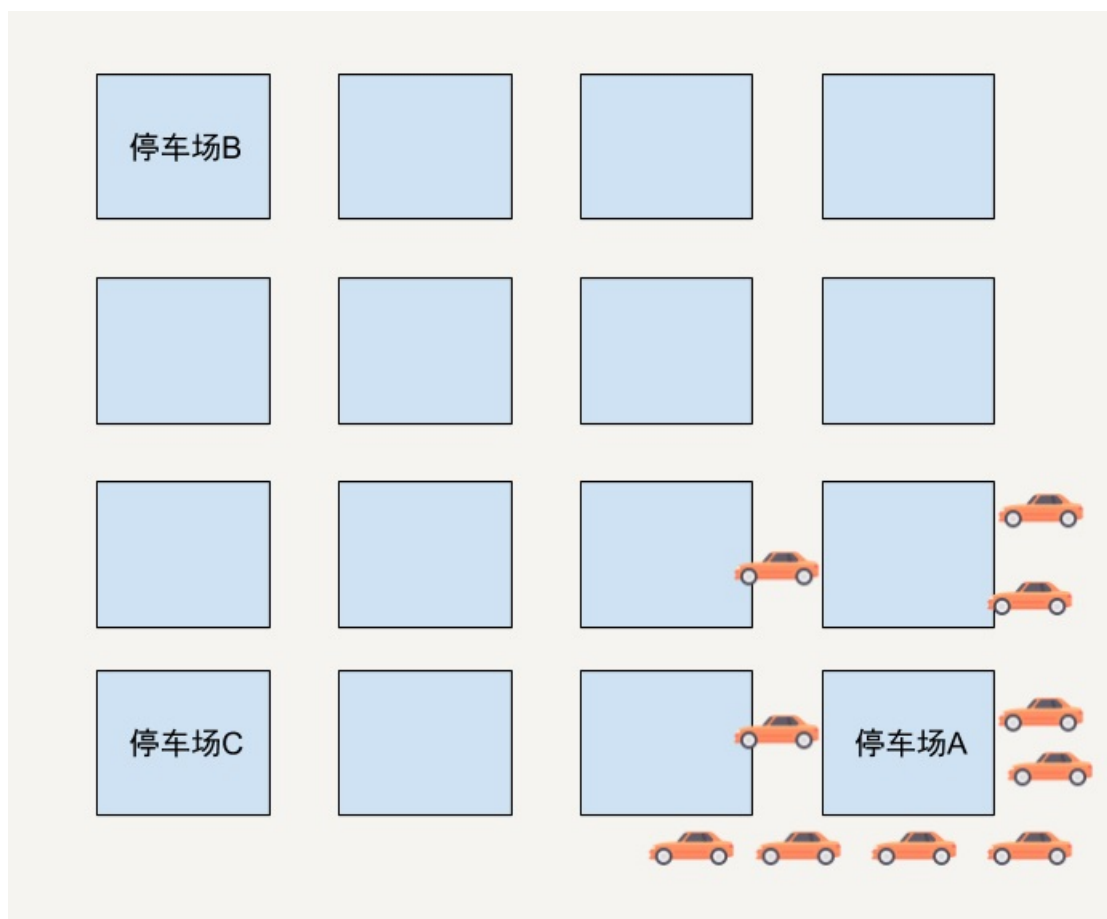
看到这里，我想先请你结合之前所讲到的广告精准投放案例，思考一下Lambda架构是如何应用在这个App里的，然后再听我娓娓道来。

好，我们来梳理一下各种可以利用到的大数据。

首先是可以拿到各类停车场的数据。这类数据的实时性虽然不一定高，但是数据的准确性高。那我们能不能只通过这类大数据来推荐停车位呢？

我来给你举个极端的例子。假设在一个区域有三个停车场，停车场A现在只剩下1个停车位了。

停车场B和C还有非常多的空位。而在这时候距离停车场比A较近的位置有10位车主在使用这个App寻求推荐停车位。如果只通过车主和停车场的距离和停车场剩余停车位来判断的话，App很有可能会将这个只剩下一个停车位的停车场A同时推荐给这10位用户。



结果可想而知，只有一位幸运儿能找到停车位，剩下的9位车主需要重新寻找停车位。

如果附近又出现了只有一个停车位的停车场呢？同理，这个App又会推荐这个停车场给剩下的9位用户。这时又只能有一位幸运儿找到停车位。

如此反复循环，用户体验会非常差，甚至会导致用户放弃这个App。

那我们有没有办法可以改进推荐的准确度呢？

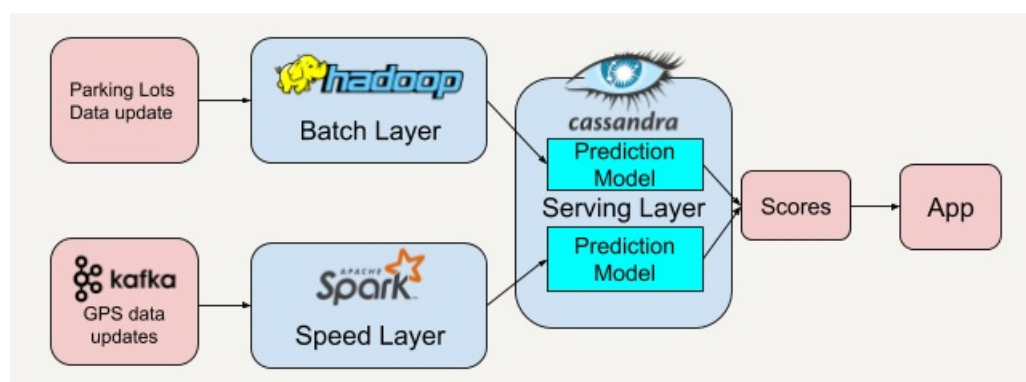
你可能会想到我们可以利用这些停车场的历史数据，建立一个人工智能的预测模型，在推荐停车位的时候，不单单考虑到附近停车场的剩余停车位和用户与停车场的相邻距离，还能将预测模型应用在推荐里，看看未来的一段时间内这个停车场是否有可能被停满了。

这时候我们的停车位推荐系统就变成了一个基于分数（Score）来推荐停车位的系统了。

好了，这个时候的系统架构是否已经达到最优了呢？你有想到应用Lambda架构吗？

没错，这些停车场的历史数据或者每隔半小时拿到的停车位数据，我们可以把它作为批处理层的数据。

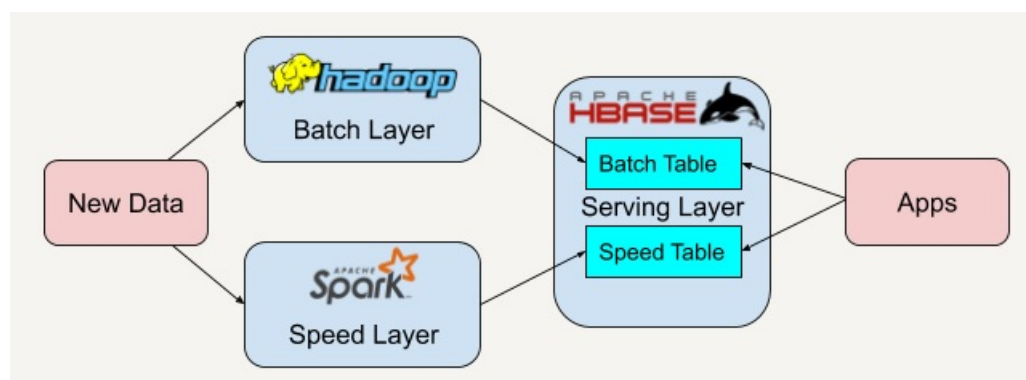
那速度层的数据呢？我们可以将所有用户的GPS数据聚集起来，这些需要每秒收集的GPS数据刚好又是速度层所擅长的实时流处理数据。从这些用户的实时GPS数据中，我们可以再建立一套预测模型来预测附近停车场位置的拥挤程度。



服务层将从批处理层和速度层得到的分数结合后将得到最高分数的停车场推荐给用户。这样利用了历史数据（停车场数据）和实时数据（用户GPS数据）能大大提升推荐的准确率。

## 小结

在了解Lambda架构后，我们知道Lambda架构具有很好的灵活性和可扩展性。我们可以很方便地将现有的开源平台套用入这个架构中，如下图所示。





当开发者需要迁移平台时，整体的架构不需要改变，只需要将逻辑迁移到新平台中。

例如，可以将Apache Spark替换成Apache Storm。而因为我们有批处理层这一概念，又有了很好的容错性。

假如某天开发者发现逻辑出现了错误，只需要调整算法对永久保存好的数据重新进行处理写入服务层，经过多次迭代后整体的逻辑便可以被纠正过来。

从我的开发经验来看，现在有很多的开发项目可能已经有了比较成熟的架构或者算法了。

但是如果我们平时能多思考一下现有架构的瓶颈，又或者想一想现在的架构能不能改善得更好，有了这样的思考，在学习到这些经典优秀架构之后，说不定真的能让现有的架构变得更好。

也就是说，作为一名优秀的架构师，“现有的架构能不能做得更好？”应该是一个需要经常思考的问题。

## 思考题

你所做的项目开发能否利用Lambda架构呢？在生活中有没有哪些大数据处理场景可以利用Lambda架构呢？

欢迎在评论中留言，与我和其他同学一起讨论。

如果你觉得今天有些不一样的收获，也欢迎你把这篇文章分享给你的好友。



# 大规模数据处理实战

Google 一线工程师的大数据架构实战经验

蔡元楠  
Google Brain 资深工程师



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

## 精选留言：

- :) 2019-05-08 00:54:32  
我们公司做的实时数仓 就满足Lambda 架构。1.批处理部分。定时拉取业务库的数据，并在hive做批处理计算。  
2.速度部分。通过订阅mysql数据库的binlog，实时获取数据库的增删改等的操作，通过kafka和flink，生

成相关结果。 [7赞]

作者回复2019-05-08 07:07:50

谢谢你的经验分享！

- JohnT3e 2019-05-08 09:01:09

lambda架构是不是可能会导致相似的处理逻辑在batch层和speed层都要开发一遍？ [5赞]

作者回复2019-05-08 10:20:50

谢谢你的提问！没错，这也是Lambda架构的一个缺点，开发者必须要把同样的逻辑在两个地方维护，特别是当技术栈不一样的时候会很头疼。

- leeon 2019-05-08 09:36:54

目前是通过Kafka将行为数据收集到hdfs，然后spark批处理t+1计算长期数据，生成固定格式的特征同步到kv上线，同时实时收集服务也从Kafka中消费最新的行为，两层输出特征格式统一，供画像服务使用 [4赞]

作者回复2019-05-08 10:21:59

谢谢你的经验分享！

- 刘鹤东[机智] 2019-05-08 10:47:02

今天的课让我想到了昨天评审的一个需求。这个需求是通过收集用户的及时数据能够知道用户常走的路线，后续用户在使用产品进行导航时，如果没有开启路线引导，但是却正在行驶在常走的路线上，一样给用户准确预报常走路况的路况，这个其实就可以实用了lambda架构。历史数据实用批处理，实时数据实用流处理也就是速度层，这样可以根据历史数据和用户实时上传的数据进行准确的路况播报了。

当然，包括网约车，外卖订单和老师讲的停车场的应用场景也相近，这节课受益匪浅。 [3赞]

作者回复2019-05-08 12:45:07

谢谢你的留言！能有收获就好。

- 越甲非甲 2019-05-08 08:10:49

老师您好！lambda架构的思路中，感觉最终决定结果的还是批处理结果。而流处理结果更多的是满足实时性的需求。不同的业务场景下，综合两种处理结果获得对外服务的结果模型，其模型和算法应该都不相同。这种处理结果的合并方面，是否有某些原则范式或者思路呢？谢谢老师！ [3赞]

作者回复2019-05-08 08:57:36

您好，谢谢提问！

其实Lambda架构的应用场景最终还是会去服务同一种业务，毕竟流处理结果是对批处理结果延时的一种补偿。即使用到的算法不尽相同，但是合并的时候，最后存储的模型或者是存储数据的Schema都还是要一致的。

- yiwu 2019-05-08 07:49:57

实时报表，实时数仓，实时用户画像标签，实时反欺诈，实时风控，以前很多需要批量计算的数据都可以变成batch+speed或speed替代 [3赞]

作者回复2019-05-08 08:57:54

谢谢你的经验之谈！

- 大牛凯 2019-05-08 03:35:40

老师您好，请问lambda架构可以看成一种在线学习的实现方式么？ [3赞]

作者回复2019-05-08 07:05:00

您好，谢谢提问！可以的，Lambda架构不仅仅可以应用在在线学习上，有非常多应用场景都可以应用上。

● 明翼 2019-05-09 08:36:00

简单又实用的lambda架构，如果实时和批量不能同时满足那就分开吧，用的时候综合下，让我想到现在开源的数据湖，delta data lake，如果批量和实时矛盾就分开吧，读写分，采用不同行式或列式存储，实时和历史分开，实时数据再定期变成历史，这个架构最大难点是如何合并speed和batch [2赞]

作者回复2019-05-09 11:01:59

谢谢你的经验之谈！

● Codelife 2019-05-08 17:34:50

原来这就是Lambda架构，其实，我们现在就用的Lambda 架构，kafka+storm+MR+Hbase来实现，现在的问题是：

1.storm是用java开发，MR是用python开发，导致同一逻辑需要两种实现

2.storm窗口期数据一般在5-10分钟，由于我们的数据有时间和空间属性的时序数据，前后关联性比较大，中间可能有噪点数据，所以很容易出现实时和历史分析结果不一致的问题，虽然最终用历史覆盖了实时，保持了最终一致性。 [2赞]

作者回复2019-05-09 01:57:26

谢谢你的经验分享！

是的啊，有时候不知不觉就会发现自己原来就已经在使用了某个特定技术。这让我想起设计模式，早期阶段在代码中自己已经在实现某个设计模式了，但是因为那时候还没有系统地去学习设计模式，没察觉到而已。

● 4 2019-05-09 21:49:46

关于为什么叫lamda架构有一个猜想。lamda的希腊字母是λ，这正好表示batch 和 speed两种最后汇聚到一起。不知道猜想对不对，但是感觉通过希腊字母，象形的代表架构模式还是挺有意思的。 [1赞]

作者回复2019-05-10 03:00:40

谢谢你的分享！Bingo，我觉得是完全正确的！以前在读技术文章的时候就看到过一种说法是：完整的数据集 = λ (实时数据) \* λ (历史数据)。

● json 2019-05-09 13:22:34

最近看到一个技术名词，叫做Reactive Design Patterns，国内的翻译是反应式设计模式，对应的编程方式叫做Reactive Programming，不知道这种设计模式跟专栏中提及的各种架构思想的关系是什么？ [1赞]

● 利利 2019-05-08 23:36:44

lambda架构，批处理层和流处理层（速度层），对于某些相同的业务往往需要开发两套代码，这个很不友好；对于催生的Apache Flink、Apache Beam这种在做流批统一的，感觉会是未来的主流 [1赞]

作者回复2019-05-09 15:53:11

谢谢你的留言！哈哈，这个观点我也很赞同！

● Mr.Mouse 2019-05-08 17:15:07

lambda架构提供给服务层的结果是不是 speed层实时结果+batch层上次存储的历史结果 整合在一起的结果？



[1赞]

作者回复2019-05-09 01:59:18

谢谢你的提问！是的呢，你的理解是正确的。Batch层因为拥有历史数据，所以Batch层的结果可以不断校对Speed层的误差。

• 风翱 2019-05-08 16:29:48

公司目前的一个系统可以引入lambda架构。原来的系统中，分为了两部分：一部分的数据收集处理，采用每天汇总的方式，数据截止到前一天；另一部分的数据为实时变化，采用的是RabbitMQ的消息处理方式。刚好可以利用lambda架构，来达到两部分数据都实时的情况。

目前节假日出现，一些旅游景点，出现了人满为患的情况，可以利用lambda架构，做到错峰出行。 [1赞]

作者回复2019-05-09 02:02:16

谢谢你的经验分享！

• dylan 2019-05-10 09:56:57

把批处理层和速度处理层称为离线处理和实时处理（近实时）更好理解吧

作者回复2019-05-10 10:33:24

谢谢你的建议！在翻译这一块我也没有找到一个标准，所以就直译了，哈哈。

• hua168 2019-05-10 00:22:14

老师，开车的现在在国内基本上用的都是高德地图，比其它地图还准确，比如：

- 1.开到哪个位置了，基本上误差都很小，像环道，到第几环道开出
- 2.如果在高速公路开着车打开高德地图，后面有车来都会提示
- 3.哪个高速公路塞车，什么时候没塞车

这些也是Lambda吗？

• 汶恬萝 2019-05-09 19:41:42

其实impala mix with hdfs and kudu就是一种较好的lambda实现方案。应用层统一使用sql，底层存储兼容基于hdfs的批量任务(impala,hive,spark等)和kudu的近实时存储。

使用场景在并发要求较高的olap或者吞吐量不大的olap均可。

如果又要求吞吐量高，又要求数据量大的实时场景(感觉这种场景相当少)，可能才会考虑两种不同技术架构分别跑批和处理实时数据

作者回复2019-05-10 03:01:19

谢谢你的技术分享，学习了！

• 夜阳羽 2019-05-09 14:55:57

为什么叫lambda架构呢，lambda在此处的含义是什么？

作者回复2019-05-10 03:05:12

谢谢你的提问！以前在读技术文章的时候看到过一种说法是因为：完整的数据集 =  $\lambda$  (实时数据) \*  $\lambda$  (历史数据)。所以就有了Lambda Architecture的叫法。

• hua168 2019-05-09 13:20:07

老师打扰一下：

Hadoop MapReduce, Apache Spark, Apache Storm, Apache Flink, Apache Apex, Apache Beam

如果现在才开始学大数据，除了hadoop过时了，其它的Apache Spark, Apache Storm, Apache Flink, Apache Apex 还需要学吗？直接Apache Beam学起？

- CoderLean 2019-05-09 08:32:15

我想问一下，作为服务层套个hbase虽然能实现数据存储，但是hbase的时效性不高啊？作为app层还要用什么工具来提高对外查询服务？

作者回复2019-05-09 16:23:55

谢谢你的留言提问！App层也还可以用Cassandra或者Mysql等等。另外我不是很理解你说“HBase的时效性不高”的意思，能否展开说明或者举例说明，我们也可以就这个问题继续讨论，欢迎你继续留言。