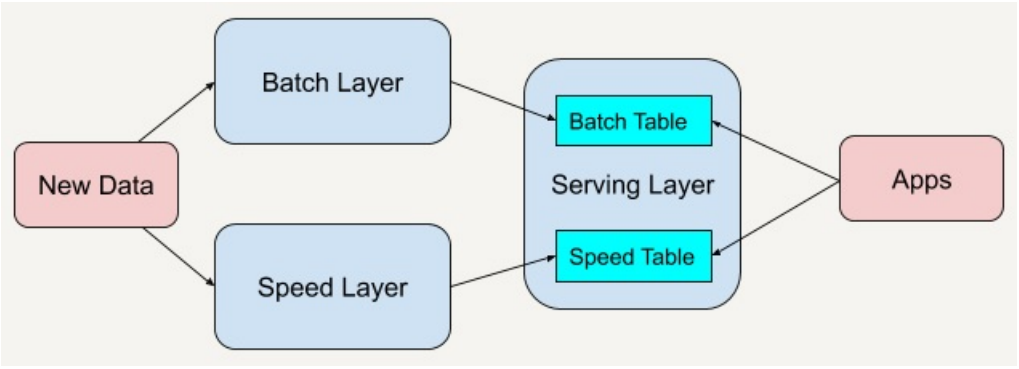
11-Kappa架构：利用Kafka锻造的屠龙刀

今天我要分享的主题是Kappa架构。同样身为大规模数据处理架构，Kappa架构这把利用Kafka锻造的“屠龙刀”与Lambda架构的不同之处在哪里呢？

上一讲中，我讲述了在处理大规模数据时所用到经典架构Lambda架构。我先来带你简要回顾一下。



Lambda架构结合了批处理和流处理的架构思想，将进入系统的大规模数据同时送入这两套架构层中，分别是批处理层（Batch Layer）和速度层（Speed Layer），同时产生两套数据结果并存入服务层。

批处理层有着很好的容错性，同时也因为保存着所有的历史记录，使产生的数据集具有很好的准确性。速度层可以及时地处理流入的数据，因此具有低延迟性。最终服务层将这两套数据结合，并生成一个完整的数据视图提供给用户。

Lambda架构也具有很好的灵活性，你可以将现有开源生态圈中不同的平台套入这个架构，具体请参照上一讲内容。

Lambda架构的不足

虽然Lambda架构使用起来十分灵活，并且可以适用于很多的应用场景，但在实际应用的时候，Lambda架构也存在着一些不足，主要表现在它的维护很复杂。

使用Lambda架构时，架构师需要维护两个复杂的分布式系统，并且保证他们逻辑上产生相同的结果输出到服务层中。

举个例子吧，我们在部署Lambda架构的时候，可以部署Apache Hadoop到批处理层上，同时部署Apache Flink到速度层上。

我们都知道，在分布式框架中进行编程其实是十分复杂的，尤其是我们还会针对不同的框架进行专门的优化。所以几乎每一个架构师都认同，Lambda架构在实战中维护起来具有一定的复杂性。

那要怎么解决这个问题呢？我们先来思考一下，造成这个架构维护起来如此复杂的根本原因是什么呢？

维护Lambda架构的复杂性在于我们要同时维护两套系统架构：批处理层和速度层。我们已经说过了，在架构中加入批处理层是因为从批处理层得到的结果具有高准确性，而加入速度层是因为它在处理大规模数据时具有低延时性。

那我们能不能改进其中某一层的架构，让它具有另外一层架构的特性呢？

例如，改进批处理层的系统让它具有更低的延时性，又或者是改进速度层的系统，让它产生的数据视图更具准确性和更加接近历史数据呢？

另外一种在大规模数据处理中常用的架构——Kappa架构（Kappa Architecture），便是在这样的思考下诞生的。

Kappa架构

Kappa架构是由LinkedIn的前首席工程师杰伊·克雷普斯（Jay Kreps）提出的一种架构思想。克雷普斯是几个著名开源项目（包括Apache Kafka和Apache Samza这样的流处理系统）的作者之一，也是现在Confluent大数据公司的CEO。

克雷普斯提出了一个改进Lambda架构的观点：

> 我们能不能改进Lambda架构中速度层的系统性能，使得它也可以处理好数据的完整性和准确性问题呢？我们能不能改进Lambda架构中的速度层，使它既能够进行实时数据处理，同时也有能力在业务逻辑更新的情况下重新处理以前处理过的历史数据呢？

他根据自身多年的架构经验发现，我们是可以做到这样的改进的。

在前面Publish–Subscribe模式那一讲中，我讲到过像Apache Kafka这样的流处理平台是具有永久保存数据日志的功能的。通过平台的这一特性，我们可以重新处理部署于速度层架构中的历史数据。

下面我就以Apache Kafka为例来讲述整个全新架构的过程。

第一步，部署Apache Kafka，并设置数据日志的保留期（Retention Period）。这里的保留期指的是你希望能够重新处理的历史数据的时间区间。

例如，如果你希望重新处理最多一年的历史数据，那就可以把Apache Kafka中的保留期设置为365天。如果你希望能够处理所有的历史数据，那就可以把Apache Kafka中的保留期设置为“永久（Forever）”。

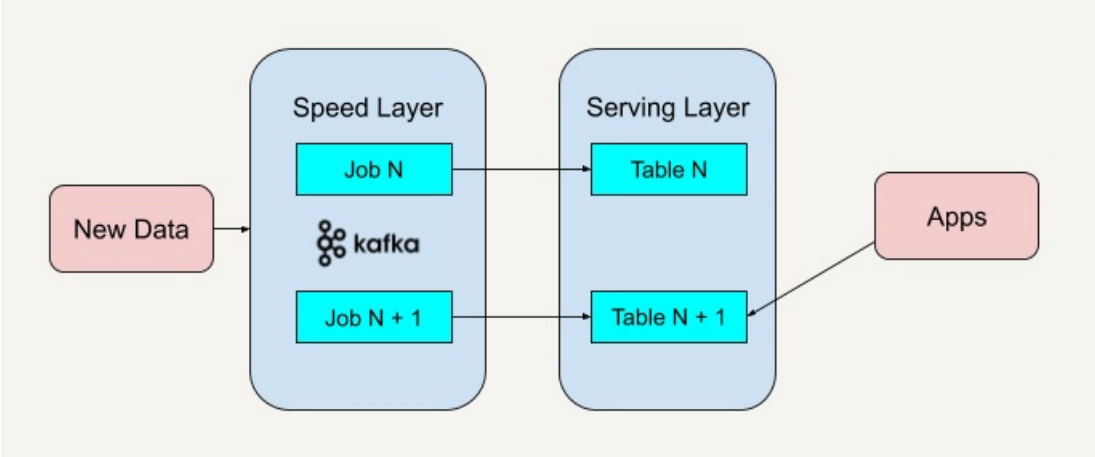
第二步，如果我们需要改进现有的逻辑算法，那就表示我们需要对历史数据进行重新处理。

我们需要做的就是重新启动一个Apache Kafka作业实例（Instance）。这个作业实例将重头开始，重新计算保留好的历史数据，并将结果输出到一个新的数据视图中。我们知道Apache Kafka的底层是使用Log Offset来判断现在已经处理到哪个数据块了，所以只需要将Log Offset设置为0，新的作业实例就会重头开始处理历史数据。

第三步，当这个新的数据视图处理过的数据进度赶上了旧的数据视图时，我们的应用便可以切换到从新的数据视图中读取。

第四步，停止旧版本的作业实例，并删除旧的数据视图。

这个架构就如同下图所示。



与Lambda架构不同的是，Kappa架构去掉了批处理层这一体系结构，而只保留了速度层。你只需要在业务逻辑改变又或者是代码更改的时候进行数据的重新处理。

当然了，你也可以在我上面讲到的步骤中做一些优化。

例如不执行第4步，也就是不删除旧的数据视图。这样的好处是当你发现代码逻辑出错时可以及时回滚（Roll Back）到上一个版本的数据视图中去。又或者是你想在服务层提供A/B测试，保留多个数据视图版本将有助于你进行A/B测试。

在介绍完Kappa架构的概念后，我想通过一个实战例子，来和你进一步学习Kappa架构是如何应用在现实场景中的。

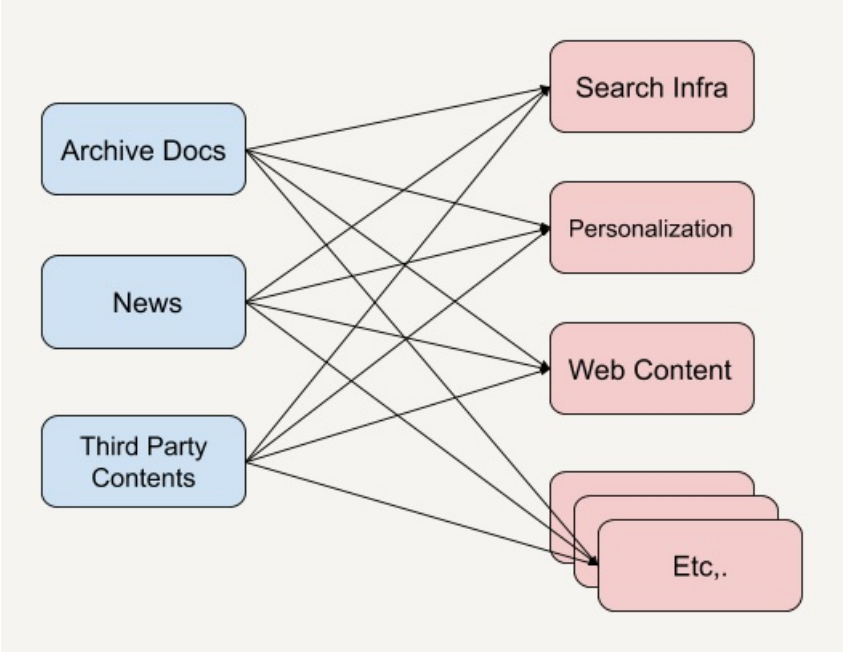
《纽约时报》内容管理系统架构实例

《纽约时报》是一个在美国纽约出版，在整个美国乃至全世界都具有相当影响力的日报。

《纽约时报》的内容管理系统收集、保存着各种各样来源的文档。这些文档有从第三方收集来的资料，也有自己报社编辑部所撰写的故事。当你访问《纽约时报》网站主页时，甚至能够查到162年前的新闻报道。

可想而知，要处理这么大规模的内容，并将这些内容提供于在线搜索、订阅的个性化推荐以及前端应用程序等等的服务，是一个非常棘手的任务。

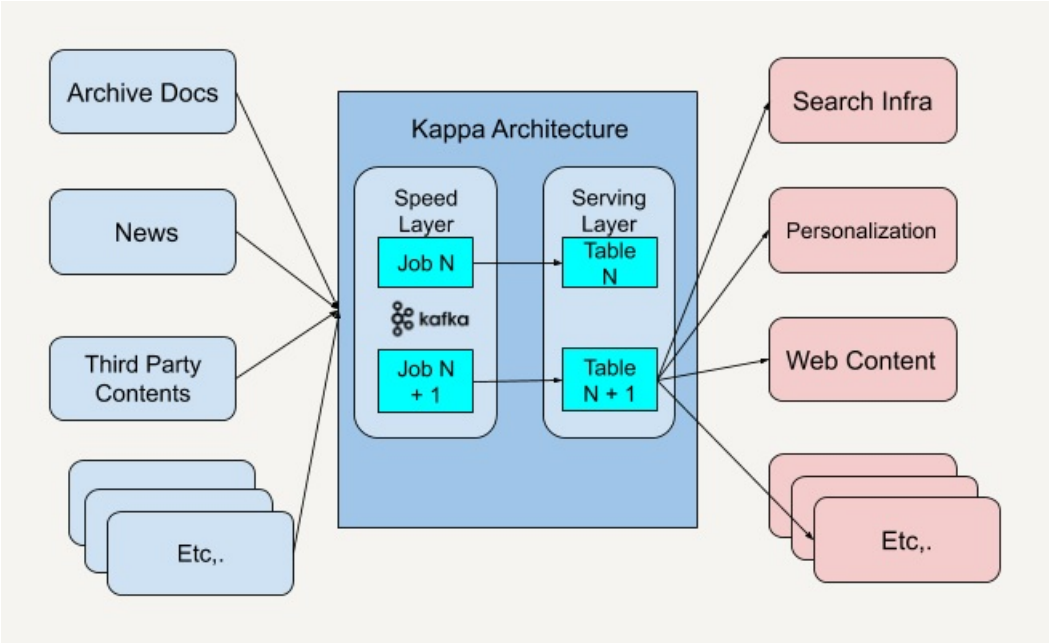
我们先来看看他们曾经使用过的一个老式系统架构。



我们可以看到，这种系统架构是一种相当典型的基于API的架构，无论是在系统调度上还是使用场景上都存在着自身的不足。我来给你举一些例子。

- 不同的内容API可能由不同的团队开发，从而造成API有不同的语义，也有可能需要不同的参数。
- 调用不同API所得到的内容结果可能有不同的格式，在应用端需要重新进行规范化（Standardization）。
- 如果客户端上会实时推送一些新的热点新闻或者突发新闻（Breaking News），那么在上述基于API的架构中，想要实时获知新闻的话，就需要让客户端不停地做轮询操作（Polling）。轮询操作在这里指的是客户端定期地重复调用系统API来查看是否有新的新闻内容，这无疑增加了系统的复杂性。
- 客户端很难访问以前发布过的内容。即便我们知道这些已发布过的新闻列表需要从哪里获取，进行API调用去检索每个单独的新闻列表还是需要花很长的时间。而过多的API调用又会给服务器产生很大的负荷。

好，那现在你再来看看当《纽约时报》采取了Kappa架构之后，新的系统架构是什么样的。



首先，Kappa架构在系统调度这个层面上统一了开发接口。

你可以看到，中间的Kappa架构系统规范好了输入数据和输出数据的格式之后，任何需要传送到应用端的数

据都必须按照这个接口输入给Kappa架构系统。而所有的应用端客户都只需要按照Kappa架构系统定义好的输出格式接收传输过来的数据。这样就解决了API规范化的问题。

我们再来看看增加了中间一层Kappa架构之后数据传输速度上的变化。

因为Apache Kafka是可以实时推送消息数据的，这样一来，任何传输进中间Kappa架构的数据都会被实时推送到接收消息的客户端中。这样就避免了在应用层面上做定期轮询，从而减少了延时。而对于重新访问或者处理发布过的新闻内容这一问题，还记得我之前和你讲述过的Kafka特性吗？只需要设置Log Offset为0就可以重新读取所有内容了。

好了，在讲述完Kappa架构和它的应用实例之后，我想强调一下，Kappa架构也是有着它自身的不足的。

因为Kappa架构只保留了速度层而缺少批处理层，在速度层上处理大规模数据可能会有数据更新出错的情况发生，这就需要我们花费更多的时间在处理这些错误异常上面。

还有一点，Kappa架构的批处理和流处理都放在了速度层上，这导致了这种架构是使用同一套代码来处理算法逻辑的。所以Kappa架构并不适用于批处理和流处理代码逻辑不一致的场景。

小结

在最近两讲中，我们学习到了Lambda架构和Kappa架构这两种大规模数据处理架构，它们都各自有着自身的优缺点。我们需要按照实际情况来权衡利弊，看看我们在业务中到底需要使用到哪种架构。

如果你所面对的业务逻辑是设计一种稳健的机器学习模型来预测即将发生的事情，那么你应该优先考虑使用Lambda架构，因为它拥有批处理层和速度层来确保更少的错误。

如果你所面对的业务逻辑是希望实时性比较高，而且客户端又是根据运行时发生的实时事件来做出回应的，那么你就应该优先考虑使用Kappa架构。

思考题

在学习完Lambda架构和Kappa架构之后，你能说出Kappa架构相对Lambda架构的优势吗？

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：

- :) 2019-05-10 02:22:26
  1.kappa架构使用更少的技术栈，实时和历史部分都是同一套技术栈。lambda架构为了解决历史部分和实时部分可能会使用不同的技术栈。

  2.kappa架构使用了统一的处理逻辑。而lambda架构分别为历史和实时部分使用了两套逻辑。一旦需求变更，两套逻辑都要同时变更。

  3.kappa架构具有流式处理的特点和优点。比如可以具有多个订阅者，比如具有更高的吞吐量。

  [3赞]

  作者回复2019-05-10 08:28:40
  谢谢你的留言！不错的总结！

- leben krieg 2019-05-10 09:43:51
  如果批处理比较多的话，每次都从kafka的earlist offset消费的话，第一会耗费很长很长时间，而且消费者如果资源不够多，会导致任务堆积的吧。所以kappa不适合批处理多的架构。

  Kappa架构因为整合了批处理层和速度层，优势就是:
  1. 实时性比较高，适合对实时性要求高的场景
  2. 业务逻辑可以使用统一的API来编写，那么对于之后的业务需求变更和代码维护都比较友好 [1赞]

- 邱从贤※klion26 2019-05-10 09:57:29
  kappa 最好的地方在于代码写一份就好了，维护成本大大降低，曾经核对过 mr 和 storm 产生的结果，真的很麻烦（还只对过一次），如果逻辑变更比较频繁的话，改一次需要对一次

  作者回复2019-05-10 10:30:18
  谢谢你的经验分享！是的，而且很多时候批处理层和速度层所产生的结果只是逻辑上一致，格式上可能还不一定能够放在一起核对。

- 锦 2019-05-10 09:32:37
  Kappa是解决lombda架构维护两套处理逻辑的复杂性痛点而诞生的，它使用kafka配合处理流处理业务，具有简单，处理数据快，支持发布订阅模式的优点？。但同时也带来了数据错误和异常处理的成本。

- 青石 2019-05-10 09:29:11
  kappa架构感觉很适合接入新的消费者时，用来同步kafka中保存的底量数据的场景？或者通过调整log offset后实现前后翻页？

  kappa架构只保留速度层，是否更适合非特大数据集，毕竟实时性和kafka的数据存储也是要考虑的问题。

  另外，如果经常调整log offset也会给kafka带来更大的压力呀，部署场景在消费者端加缓存？

- cricket1981 2019-05-10 09:28:24
  kappa架构和lambda架构都有各自的优点和缺点和各自的使用场景。关键是要不要维护两套不同的技术栈，现在像spark和flink都在往批流统一的方向努力，目的就是统一技术栈，减少开发和运维的成本。

- miwucc 2019-05-10 09:18:20
  文中说的kappa的两个缺点，不是很理解。

  第一点，说实时数据需要处理数据异常，按理说说lambda结构的速度层也会面对这个问题啊。还是是说kappa的历史数据计算层在计算的时候同时需要清洗数据，而lambda的批处理层是直接读取的清洗好结构化好的数据？

  弟二点：说如果历史计算和实时计算需要不同逻辑时候不适用。就在该框架下写两套逻辑不就行了么？为啥非要用lambda架构再写两套逻辑，还面对技术栈切换

- 朱晋君 2019-05-10 09:18:06
  1.如果批处理和流处理的逻辑一致，那可以选择用kappa架构，否则lambda架构更好

  如果选择kappa架构，kafka offset置为0来计算批数据，会不会非常耗时呢？可以用批结果加流结果作为下次计算的批结果吗

- 小度 2019-05-10 09:05:41
  有没有完美的架构？Kappa 架构Lambda 架构都有优缺点能否融合使用？

  作者回复2019-05-10 09:57:50
  谢谢你的提问！现在还没有什么架构是完美的，具体使用哪一个架构肯定还是要根据自身需求和应用场景来做取舍。Kappa架构的提出就是想改善Lambda架构里需要维护两套系统而提出的，所以并不能融合使用。

- 明翼 2019-05-10 08:38:08
  kappa架构优点：
  1.流处理和批处理同一套算法逻辑，更简单。
  2.数据格式统一，给后端处理带来方便。
  3.批量处理的速度更快。

  作者回复2019-05-10 09:06:32
  谢谢你的总结！

- jon 2019-05-10 08:34:00
  kappa优点：不用分别编写批、流两套处理逻辑，采用发布-订阅模式，接受多个订阅者，可重复消费历史数据。
  kappa缺点：在速度层进行大规模实时流处理容易处理数据更新错误，可靠性不如批处理层。

  作者回复2019-05-10 09:20:40
  谢谢你的留言！很不错的总结，把优缺点都罗列出来了！

- hua168 2019-05-10 07:28:35
  如果kafka处理全部数据，数据量非常大会不用卡死？要使用群集吗？

  作者回复2019-05-10 09:55:41
  谢谢你的提问！Kafka的优点之一就是拥有高吞吐量的，而Kafka系统的设计里肯定是有集群存在的。

- Rainbow 2019-05-10 06:59:50
  spark算kappa架构吗？批处理 流处理一套

  作者回复2019-05-10 10:14:51
  谢谢你的提问！Spark不算Kappa架构，Spark是有能力可以处理批处理和流处理，我们可以利用Spark搭建Kappa架构出来，但是它本身不具备这种架构的思想在里面。这就好比我们可以借助编程语言实现一些算法，但是我们不会说这种编程语言就属于这种算法一样。