主流大数据架构调研报告

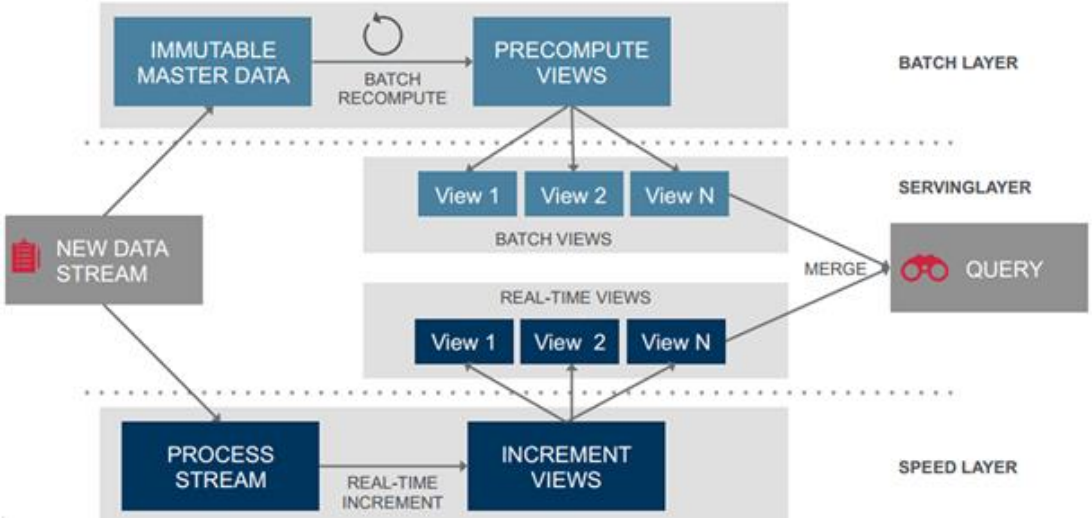
一、前言

随着时代的发展变化，我们对于数据的使用方式也在发生改变。数据仓库中的数据量从 TB 级增长到 PB 级，并且还在爆炸式增长，对于不同的需求，我们需要使用不同的方式去处理数据，应用不同的数据存储方式，支持不同类型的数据查询，以满足各种各样的用户需求。随着 Hadoop 的诞生，一系列大数据架构应运而生，并且在不断发展进步，本文通过对当前主流的大数据框架进行调研，分析其优缺点以及应用场景，并结合时代背景，对未来的大数据架构进行展望。

二、Lambda 架构

Lambda 架构可以实时处理海量高并发数据，其首先将大数据系统架构分为批处理层（batch layer）、实时处理层（speed layer）和服务层（serving layer)3 个层次，然后分别对这 3 个层次进行操作，以便缓解数据库的读写压力和降低实时数据处理的复杂程度。该架构整合离线计算和实时计算，融合不可变性、读写分离和复杂性隔离等一系列架构原则[1]，可集成 Hadoop,Kafka,Storm,Spark,Hbase 等各类大数据组件。

Lambda 通过三层架构解决了在总体数据集上进行实时查询和计算的问题。在 batch layer 中，进行批处理，对全体数据进行运算和处理，为下游更快的实时查询提供基础。此外，由于这一层的批处理是基于全体离线历史数据的，因此，运算的准确性可以得到保证。



在 Speed Layer 层，进行实时增量数据的处理，在这一层中，系统只需要处理当前时间点附近产生的数据，数据量相比于全体数据来说很小，所以这一层的重点在于低延迟，当系统在进行批处理运算，无法及时产生结果时，实时计算的结果可以反馈给用户，以确保用户的查询总有结果返回。

在 Serving Layer 层，系统将批处理层和实时处理层的数据进行合并，形成最终的结果。

Lambda 的优点在于，首先，它把实时数据和历史数据在不同的层次分开处理，计算实时数据不会使用到历史数据，因此确保了 Speed Layer 层的效率；同时，分层处理不同的数据也提高了整个系统的可靠性；另外，实时处理的数据之后都会写入到历史数据当中，成为计算批处理的数据的一部分，即使在某一次的实时数据中存在问题，当其混入到庞大的总体数据库中，再参与计算时，其影响可以忽略不计；

Lambda 也有一些缺点，首当其冲的就是维护两个不同计算系统的复杂性，在之前的分析中我们知道，Lambda 架构需要进行全体数据批处理计算和实时数据的流式计算，这两种计算方式的不同决定了我们在实现 Lambda 框架的时候就要同时维护两套代码，这样一来出现代码漏洞的可能性会大大增加，维护系统的成本也会随之提高；另外，随着信息时代的发展，我们数据仓库中的数据越来越多，批处理计算需要的时间越来越长，批处理结果和实时结果之间的延迟越来越长，甚至无法在一天之内完成一次批处理计算。

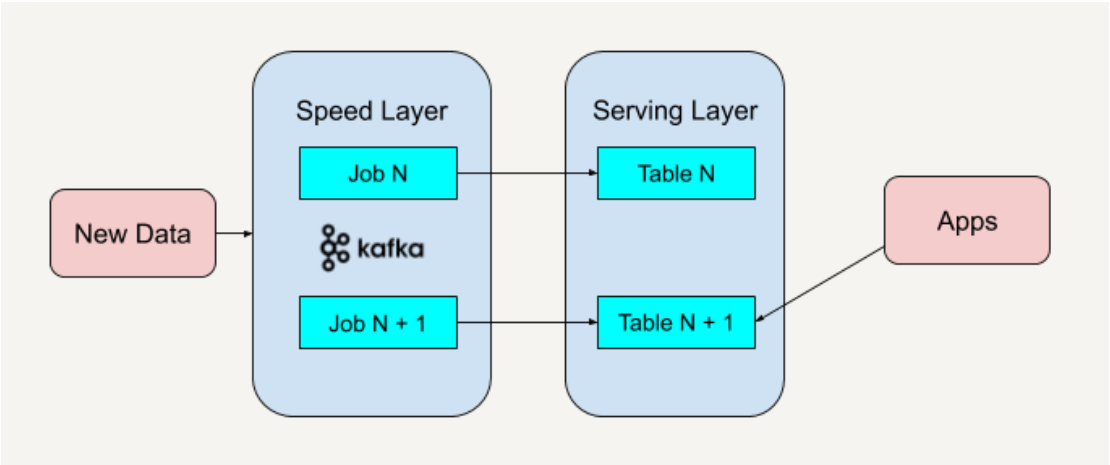
三、Kappa 架构

Kappa 结构通过使用纯流式计算的方式，解决了 Lambda 架构中存在的需要维护两套程序的问题。Kappa 架构的核心思想是通过改进流计算系统来解决数据全量处理的问题，使得实时处理过程和全量处理过程都使用流式计算。

Kappa 架构在 Lambda 的基础上进行优化，删除了 batch layer 层，将数据通道以消息队列进行替代，将数据存储在数据湖中，当需要进行离线分析或再次计算时，把数据湖中的数据经过消息队列重播一次即可。

在 Speed Layer，提供接收和存储流数据的消息队列，数据可以在某个限度内全量存储，并在需要时从头开始重新读取计算。

在 Serving Layer，提供流计算引擎，进行流分布式实时计算，并确保快速响应。当需要全量计算时，只需要重新起一个流计算实例，从头开始读取数据进行计算，输出到一个新的结果存储中即可，当新的实例做完后，停止老的流计算实例，并删除一些历史结果。

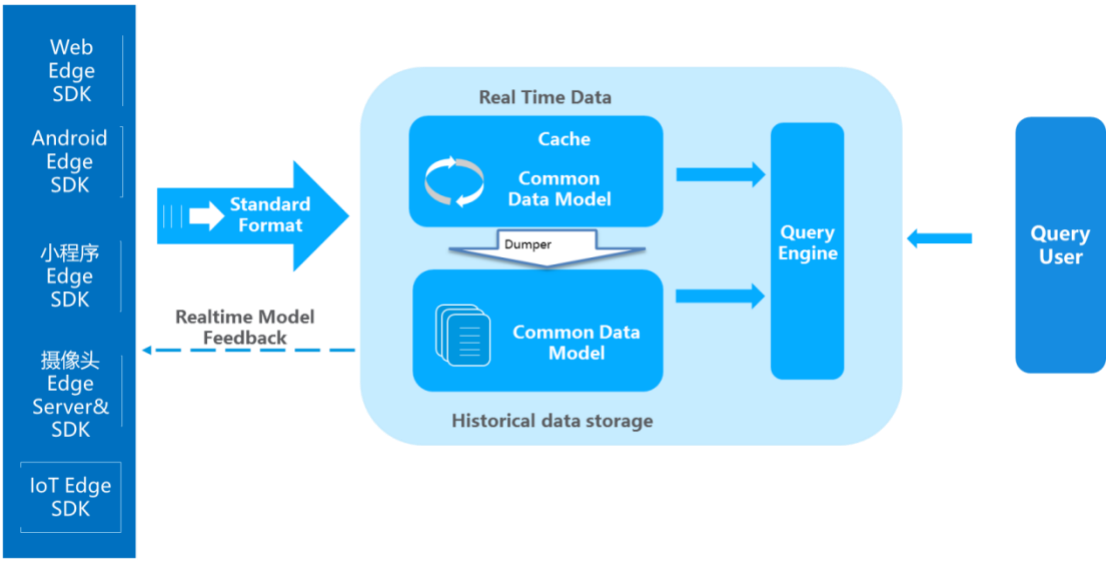


Kappa 架构的优点在于同意了全量计算和实时计算的代码，方便维护。

Kappa 架构的缺点在于，所有数据都通过流式计算，对于高吞吐量的历史数据，计算时间长，很难满足查询的即时性要求；在实时数据处理时，遇到大量不同的实时数据相关联时，高度依赖实时计算系统的能力，很可能因为数据流的先后问题，导致数据丢失；Kappa 架构的原理需要依赖于高性能的存储器，但是把全量数据都存储到高性能存储器中，会导致成本上的浪费。

四、IOTA 架构

在 IOT 大潮下，智能手机、PC、智能硬件设备的计算能力越来越强，而业务需求要求数据实时响应需求能力也越来越强，过去传统的中心化、非实时化数据处理的思路已经不适应现在的大数据分析需求，提出新一代的大数据 IOTA 架构来解决上述问题，整体思路是设定标准数据模型，通过边缘计算技术把所有的计算过程分散在数据产生、计算和查询过程当中，以统一的数据模型贯穿始终，从而提高整体的预算效率。



IOTA 框架的核心模型是 Common Data Model，它保持了整个系统中数据的一致性。

Edge SDK & Server 是数据的采集端，同时，应用边缘计算技术，在数据采集端就可以进行一些数据的处理；Real Time Data 是实时数据缓存区，实时计算的数据从这里获得，实时数据在这里暂存之后也会存入到历史数据存储区；Historical Data 是历史数据存储区；Dumper 把相关的数据从 Realtime Data 区写入 Historical Data 区；Query Engine 是查询引擎，把 Realtime Data 和 Historical Data 合并到一起查询；

IOTA 的优点在于，使用了边缘计算技术，把计算过程分散到系统的各个部分，由 Common Data Model 确保数据的统一性，并且只专注于某一个具体领域的数据计算，实现去 ETL 化，从采集端就能够开始数据的处理，提高了数据分析的效率。

IOTA 的缺点在于，目前 IOTA 架构的应用范围还不够广，具体的应用实例和相关经验也很少，还需要进行发展。

五、未来趋势

近年来,世界产生的数据就超过了可用存储量。已经从思考“存储什么”转向了“清除什么”[2]。

在传统的大数据架构中，使用到数据仓库的结构，用 ETL 对采集到的数据进行清洗，分类，之后按主题进行存储，这样对于固定主题的数据处理起来方便快捷，也便于使用，但是现实使用的过程中，我们有时候并不能事先确定好研究的主题，而是需要在数据中分析研究后才能确定研究方向；此外，对于同样的数据仓库，往往有不同的用户，而这些用户需求的数据也不是相同主题的；并且数据仓库架构在处理海量异构数据和时效性数据需求时缺陷明显[3]。所以,“数据湖”的概念应运而生,数据湖的概念指出，数据无需加工整合，可直接堆积在平台上,由最终使用者按照自己的需要进行数据处理。

无论数据架构如何设计，目的都是为了满足最终用户的需求[4]。企业内部的不同条线，不同部门的用户之间，对于数据应用的深度和广度是有差异的。这种差异带来了使用模式、数据基础和时效性要求等方面的不同要求。所以，数据湖的概念更加符合不同用户对于大数据的使用方式，在企业内部的大数据框架中，使用数据湖来代替数据仓库，这样就不必为每一个部分设置专门的数据仓库，只需要设计不同的数据加工方式即可。在未来，很多的大数据架构中可能会使用数据湖代替数据仓库的功能，给用户更多的自由度。

大数据架构的计算效率是非常重要的，因此，上文中提到了 Lambda 架构和 Kappa 架构在不久的将来将会因为数据量过于庞大而不再适用,IOTA 架构将成为时代的主流。另外，机器学习结合大数据也是当今的一大趋势，越来越多的大数据框架中会添加关于机器学习的部分,使得机器学习和大数据能够更好的协同工作，提高效率。

本文通过分析当今主流大数据架构的优缺点，提出了对于未来大数据框架发展趋势的一些见解。无论未来变化如何，我们都应该积极面对，顺应时代潮流，把握发展机遇。

参考文献

[1] 胡卫民.Lambda 架构在轨道交通车辆智能运维系统中的应用[J].控制与信息技术,2021,(02):67-71.
[2] 叶惠仙,贾如春.大数据架构关系分析及应用[J].计算机时代,2016,(12):42-45
[3] 谭景信,刘玉龙,李慧娟.虚拟化模型驱动的分布式数据湖构建方法研究[J].计算机科学与探索,2019,13(09):1493-1503
[4] 张新宇.大数据时代的数据架构设计[J].中国金融电脑,2015,(08):32-35