数据治理视角下的湖仓一体架构研究

陈氢[1,2] 宋仕伟[1]
(1. 湖北工业大学经济与管理学院，武汉 430068；2. 湖北循环经济发展研究中心，武汉 430068)

摘要：海量分布异构数据给企业数据治理带来严重挑战，加速数据仓库和数据湖向结合二者功能的湖仓一体转变。通过比较数据仓库、数据湖和湖仓一体之间的差异性，分析湖仓一体的优势及其面临的挑战，再通过划分业务领域并映射到数据视角来构建分布式湖仓一体架构，综合已有研究和相关技术构建湖仓一体功能模块，并阐述动态流批一体数据流转过程。分布式湖仓一体架构包括数据领域解耦、跨领域数据共享、联合数据治理等构建理念；湖仓一体功能模块主要包括数据源、湖仓一体核心功能区和用户；流批一体数据流转过程包括批量数据过程和实时数据过程。本研究可为湖仓一体融入有效数据治理过程，构建较为完善的湖仓一体架构体系，从而为相关研究或企业提供参考。

关键词：数据治理；数据湖；湖仓一体；数据共享
中图分类号：G203　　DOI：10.3772/j.issn.1673-2286.2023.04.003
引文格式：陈氢，宋仕伟. 数据治理视角下的湖仓一体架构研究[J]. 数字图书馆论坛，2023 (4)：19-28.

多源异构数据爆炸式增长带来数据沼泽、信息孤岛等问题，导致无用数据和陈旧数据产生，而数据湖凭借原始格式存储、数据存储类型多样和开放访问等优势解决了数据存入问题，但其缺乏事务管理支持能力、数据治理能力，从而限制了数据产出。因此，企业多以将数据提取/加载/转换（ELT）到数据湖后再提取/转换/加载（ETL）到数据仓库中的方式打通湖仓之间管道以同时获取二者优势，但这种二层架构存储成本高、数据一致性和可靠性不足、对高级分析的支持有限[1]。在此基础上，业界提出湖仓一体（lakehouse），在数据湖上添加高级管理层具化数据仓库功能，实现多元化数据存储、存储计算资源分离、事务管理支持、丰富场景分析应用等优势组合。

当前少数企业已面向业务需求实施湖仓一体解决方案，但仍存在功能不完善、架构不通用、治理不成熟、数据难共享、系统难扩展等挑战。为此，本文从数据治理的视角出发，基于数据仓库、数据湖、湖仓一体三代数据平台的差异性，分析湖仓一体面临的挑战，提出分布式湖仓一体架构，并综合当前研究构建静态湖仓一体功能模块及动态流批一体数据流转过程，形成具有一定可行性、通用性和可扩展性的湖仓一体架构体系，以支持企业决策。

1 相关研究

1.1 湖仓一体的内涵

Databricks公司于2020年率先提出湖仓一体概念，并将其描述为“一个新的、开放的数据管理架构，将数据湖的灵活性、成本效益和规模与数据仓库的数据管理和事务管理结合起来”[2]；Armbrust等[3]将其定义为“基于低成本和可直接访问的存储数据管理系统，同时具备传统分析型数据库管理系统的管理和性能特征，如事务管理、数据版本、审计、索引、缓存和查询优化”。目前并无对湖仓一体的统一且成熟的定义，相关研究多引用Armbrust等的说法。

国外学者分别定义了自下而上［数据源/数据湖/元数据、缓存和索引层/应用程序编程接口（API）层/消费

收稿日期：2023-02-28

用例]、自上而下（数据源/采集过程/原始数据开放格式存储并治理/开放API/消费用例）、自左到右（数据源/存储层/计算层/API层/消费用例）的湖仓一体架构[1,3]，虽表述存在差异，但提出的湖仓一体架构有相似的5个组成部分：数据源、数据存储、计算层、API层以及消费用例。Armbrust等[3]明确将数据存储在数据湖中以强调湖仓一体满足不同类型数据的开放格式存储需求。Microsoft Azure、Databricks等公司的湖仓一体架构则更为重视对存储对象的区域划分：青铜区（原始数据采集和保留）、白银区（过程数据过滤、清洗和增值）和黄金区（业务数据汇总）[1]。

1.2 数据仓库、数据湖和湖仓一体的差异

数据仓库、数据湖和湖仓一体之间的差异（见表1）主要体现在以下几点。

（1）数据类型：数据仓库内部高度结构化且多为关系型数据库，一般只支持在入仓前完成处理工作的结构化数据存储；数据湖可包容开放的数据类型，但其主要存储原始格式的数据，数据加工处理属于额外工作；湖仓一体存储所有类型的已处理和原格式数据[1]。

（2）采集过程：数据仓库的写时模式需在数据入仓前预先建模，并按照既定的ETL模式，以专属格式导入；数据湖的读时模式在数据入湖后按需定义架构，湖中数据以开放格式存在以适应多变的业务需求；湖仓一体同时支持预定义数据和开放数据导入以及需求导向的数据加工转换[4]。

（3）访问方式：数据仓库内的数据访问以SQL（Structured Query Language）为主，用户可以获取具有专属格式的数据；数据湖和湖仓一体配置大量开放API，可支持对数据的直接读取，读取方式包括SQL、R、Python等语言，湖仓一体同时支持原格式和处理后数据的访问[5]。

（4）可靠性和安全性：数据仓库发展较为成熟，基于其高度结构化的管理能力，可实现高质量和安全性的数据存储；数据湖内部数据具有多源异构性，尚未形成有效治理策略，易导致数据沼泽，这也是其当前面临的最大挑战；湖仓一体在湖存储机制上添加数据仓库管理功能和数据安全保障机制，可显著提高数据可靠性和安全性。

（5）适用场景：数据仓库适用于BI（Business Intelligence）、SQL应用和报告等；数据湖适用于数据科学和机器学习，二者仅支持有限应用场景；湖仓一体可同时满足SQL分析需求和数据科学、机器学习等高级分析需求，且支持直接在原始数据上应用各类分析工具，以及对流数据的持续处理和实时分析[1,5]。

表1 数据仓库、数据湖、湖仓一体的比较

	数据类型	采集过程	访问方式	可靠性和安全性	适用场景
数据仓库	结构化、已处理数据	写时模式	SQL为主，少量支持API	数据质量高、安全性高	BI、SQL应用和报告等
数据湖	结构化、半结构化、非结构化原始数据	读时模式	开放API	数据质量低、安全性低，易形成数据沼泽	数据科学、机器学习
湖仓一体	结构化、半结构化、非结构化数据	读时模式、写时模式	开放API	数据质量高、安全性高	丰富场景

1.3 湖仓一体存在挑战

尽管湖仓一体可以实现对数据湖和数据仓库优势的有机结合，但依然存在以下挑战。

（1）单体架构需扩展：湖仓一体仍是一种综合数据仓库、数据湖、Delta Lake、Spark等技术的单体架构，存在可扩展性不足、管理成本高等问题[6]。数据网格是Dehghani[7-8]于2019年提出的分散架构模式，融合了分布式领域驱动、自助平台设计、数据产品化以及联合计算治理思维。区别于微服务，数据网格专注于数字工程而非软件工程，将其分布式领域驱动概念抽象化，以支持湖仓一体扩展。

（2）架构体系需完善：相关企业湖仓一体架构多以特定业务为导向而不具有通用性，现有研究尚未形成完善的架构体系，存在缺乏完整数据生命周期过程、对存储对象定义不明确、数据治理能力需优化、支撑应用软件技术需补充等问题。

（3）领域数据需共享：当前企业数据多由中央数

据团队集中统一管理，各业务领域间缺乏高效数据共享机制，易形成数据孤岛。领域数据专业化管理并自主共享更符合海量数据环境下的数据治理愿景[8]。

为此，针对湖仓一体面临的挑战，在解耦海量数据资源并形成分布式湖仓一体架构基础上，分别构建静态湖仓一体功能模块以完善化、合理化湖仓一体系统中各功能组件构成，以及动态流批一体数据流转过程，这一过程贯穿数据采集、存储、处理、分析等数据生命周期。

2 分布式湖仓一体架构

分布式湖仓一体按业务领域划分海量数据资源，由熟悉业务的领域团队获取相关数据并履行管理义务，以降低数据集中化治理难度并提高治理效率，其优势如下：通过调整领域结构来应对不断变化的数据环境和多元化业务需求，避免企业“牵一发而动全身”；各领域按需部署湖仓一体功能模块以支撑数据采集、存储、处理、分析等数据生命周期过程，脱离中央数据团队而自主发布数据访问请求并有效共享；中央和领域团队协同发布联合数据治理策略并参与治理过程，既确保企业层面数据治理的可持续和有效性，又适应不同业务领域层面治理过程的独特性、灵活性，并最终实现共同战略目标下治理结果的一致性。

分布式湖仓一体架构（见图1）由中央发布网格目录、联合数据治理标准并提供相关基础设施，通过业务解耦定义分布式领域节点（领域1、领域2、…、领域*n*），对于不断扩展的数据源、用例或访问类型只需增加相应领域节点，每个领域独立治理和维护，通过端到端完备的数据服务确保数据可访问和调用[9]。各领域湖仓一体可拥有独立的数据目录，网格目录链接所有节点并获取完整元数据信息，是用于发现不同节点中可用数据元素的主目录，各节点通过数据共享服务和网格目录实现跨领域数据访问[9-10]。基于湖仓一体功能模块和流、批数据流转过程需要搭建基础设施，各节点通过本地实施的物理服务器、虚拟机和容器或云服务自动部署技术支撑组件，避免重复建设[10]。

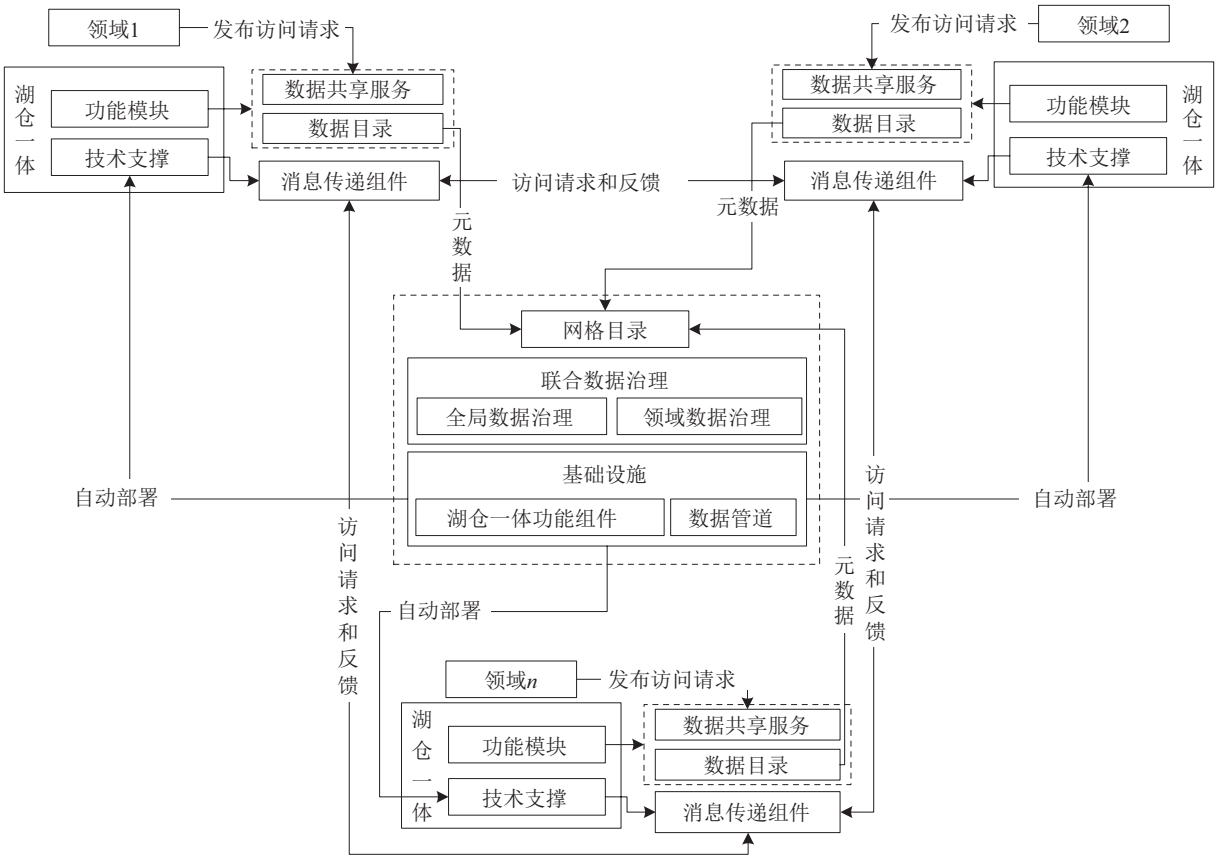

图1 分布式湖仓一体架构

2.1 数据领域解耦

应用分布式领域驱动概念，将企业业务域解耦并映射到数据视角，再将数据域解耦以消除冗余[11-12]。可根据企业实际业务和组织分类来定义领域，例如某个产品组、独立组织实体或特定业务流程都可以作为域[9]，更具系统性的方法是将数据域划分为3种类型：源数据域（以下简称“源域”）、聚合数据域和消费者/用户数据域[13-14]。

2.1.1 源域

源域与处于原始状态的数据源（例如业务系统、社交媒体等）一致，此类数据尚未用于拟合或建模而直接反映当前业务事实，多以业务域事件形式存在，其依据分布式日志或易于消费的历史片段（在密切反映业务域变化的间隔时间段内进行汇总）。源域内数据的建模和访问并非直接在数据源上进行，而是在湖仓一体中存储和结构化以便更好地理解和访问报告、机器学习训练和各类非交易性工作负载。此外，数据源变化与源域内数据变化的频率可能不一致，二者拥有不同的生命周期。

2.1.2 聚合数据域

企业范围内业务域与数据源系统可能并非一对一映射，通常许多源系统可为属于一个共享业务域的部分数据提供服务，因此可能有众多与源系统对应的数据，需要将其汇总到一个统一的聚合数据域中。例如，通过营销和销售数据生成用户画像，这个过程需要由多个源域内数据组成的长期聚合数据以形成对用户的整体看法。

2.1.3 消费者/用户数据域

与源域内数据相比，消费者/用户（例如数据科学家、数据分析师）数据域内的数据拥有不同的性质，通常通过处理、加工、转换等将源域中的事件转变为适合特定用例的结构和内容。例如，数据科学家从源域内数据中提取某些特征和附加信息，一旦这些特征（属性）被提取便可作为与用户数据相一致的领域数据被维护和共享，以训练情感分析模型或其他相邻领域模型。

2.2 跨领域数据共享

分布式领域驱动剥离中心节点，各领域湖仓一体提供数据共享服务，且任意领域间可以共享数据[9]，跨领域湖仓一体数据共享过程如下：各节点数据发布者发布元数据到数据目录中，管理员审查数据治理规范后录入，同时在网格目录中更新元数据信息。当某一节点产生数据需求，便可浏览网格目录查询感兴趣数据集所在节点，并向数据共享服务发送访问请求，访问请求通过消息传递组件被路由同步到相关数据发布者。当请求通过后，发布者则将感兴趣数据集通过数据共享服务与请求者所在领域节点共享，并不断监测数据使用模式以保证共享过程可控。

2.3 联合数据治理

传统单一集中化的数据治理模式低效且难以应对快速变化的数据视图，数据网格模式下各领域拥有数据主权并通过全局标准化实现互操作性，因此联合数据治理包含两种治理模式：全局数据治理和领域数据治理[11]。

2.3.1 全局数据治理

全局数据治理模式包括：定义数据所有权和各领域边界等信息；发布可适用于所有领域的通用治理方案，例如治理策略、流程、组织等；制定共同遵循的数据规范，例如共享API接口描述语言或元数据建模标准语言；统一业务词汇表建模，例如对客户识别号等概念的共同定义；确立最低数据质量和安全标准，例如在一组固定维度上评估所有领域质量以及根据中央身份管理来执行访问控制；通过网格目录对领域数据产品实施有效监控。

2.3.2 领域数据治理

在领域数据治理模式下，治理责任被分配给领域数据团队，并根据各领域数据产品的属性（例如数据类型、格式、访问方式等）制定元数据管理、数据质量管理、数据生命周期管理等治理计划，这种去中心化的自治模式对大数据治理更有效。

3 湖仓一体的功能模块

企业各领域按需自动部署湖仓一体，对功能模块要考虑可面向所有领域的通用性以及架构体系的完整性。功能上，确保结构化、半结构化、非结构化数据的采集、存储、共享、实时处理和分析，以及全生命周期数据治理；结构上，直接在数据湖上配置数据仓库功能，以免除数据迁库带来的成本和不一致等问题；服务上，支持复杂场景的数据分析应用。湖仓一体功能模块（见图2）包括3个部分：数据源、湖仓一体核心功能区和用户。立足数据生命周期维度，可将湖仓一体的核心功能区拆解为6层：数据采集层、数据湖层、计算层、数据服务层、数据分析层和数据治理层。

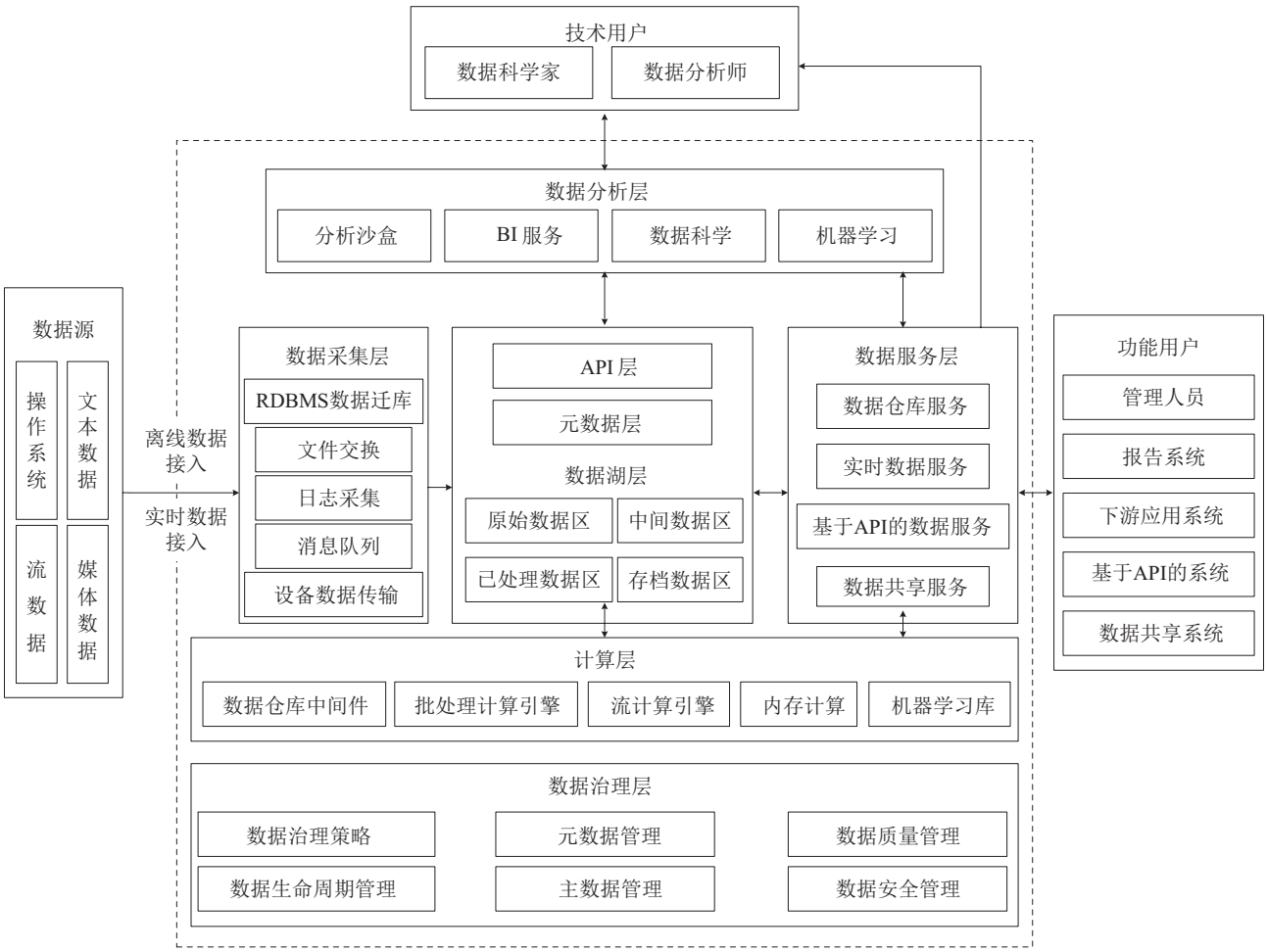

图2 湖仓一体功能模块

3.1 数据源

任何生成数据的操作系统都可能成为潜在数据源，通常联机事务处理过程（Online Transaction Processing，OLTP）产生事务数据，这些事务数据以结构化数据为主，并以高度结构化形式存储于关系数据库中。此外，非结构化数据存储于NoSQL数据库中，包括键值对、图形和JSON（JavaScript Object Notation）数据等。

除操作系统外的其他数据源以非结构化数据为主，其中：文本数据是非结构化数据的主要类型，包括文件和纯文本等，通过自然语言处理可以从文本数据中洞察信息；流数据是固定时间点某一系统不断传输的数据，包括从物联网设备发出的遥测数据、来自社交媒体平台的持续反馈以及来自地理信息系统的位置信息等，流数据为实时分析提供支持，可以满足情感分析、关键词检测等用例需求；媒体数据以图像、语音和

视频为主，可以完成对图像识别、语音识别、语音翻译文本等高级用例的支持。

3.2 湖仓一体核心功能区

3.2.1 数据采集层

数据采集方式以批量数据采集和实时数据采集为主。批量数据采集是指定期将数据提取到数据湖层，采集周期受数据源生产、推送能力和允许拉取数据能力等影响，实施批量采集操作需考虑源系统可否用于数据获取以及可获取批量数据规模的大小。实时数据采集是指在数据产生于源系统时将其拉入数据湖层，多基于Kafka队列服务实现，该服务将实时数据流分组并临时存储为用于采集的队列，还可捕获数据库中的数据变更。实时数据为持续数据流，对系统吞吐量和延迟等方面有更高要求。

数据采集层的功能是支持各主流关系数据库管理系统（Relational Database Management System，RDBMS）数据迁库，可进行跨平台实时文件交换、各类应用系统实时日志采集、基于业务数据库日志的增量同步和各类消息队列服务等。

3.2.2 数据湖层

湖仓一体将存储对象可视化为数据湖，以确保继承数据湖的多元混合存储机制，同时按数据格式和存储状态将数据湖划分为原始数据区、中间数据区、已处理数据区和存档数据区[1,9]。数据以离线和实时两种形式接入原始数据区长期存储，并保留原格式（如CSV、Apache Parquet和JSON等）。中间数据区是原始数据完成处理任务的中间阶段，例如数据清洗、过滤、聚合、附加等，保留中间数据的好处在于避免处理过程重启造成的数据丢失。已处理数据区是数据湖最高层，经过加工转换后的数据将被应用于数据分析任务（例如BI、报告和机器学习等），以及响应消费需求，服务于各类系统（如报告系统、下游应用系统、数据共享系统）等。存档数据区为应对长期存储目标而存在，以冷数据存储为主，这类数据一般没有快速检索的需求，因此多采用低价存储技术。

湖内数据以Apache Parquet等标准文件格式存储在对象存储系统中，并附加元数据层和API层。元数据层是数据湖存储重要组件，以往数据湖存储系统仅支持低级别的对象存储或文件系统接口，而丰富有效的元数据可实现湖内事务管理和其他数据管理功能，如Apache Hive工具允许以事务方式更新表，Delta lake和Apache Iceberg工具支持Apache Parquet、ORC格式文件，在实现与原始数据湖相似存储机制的基础上增强可操作性。API层提供丰富的SQL API以支持BI和报告等，并开发大量声明性DataFrame API以支持数据科学和机器学习等高级分析，外部应用可查询元数据层并通过适当接口实现对湖内数据的访问。

3.2.3 计算层

湖仓一体功能模块解耦存储和计算资源，使其分别使用独立集群，可以实现存储计算的独立扩展以支持具有大数据工作负载的并发用户。计算层内置Hive、Spark、Flink、Impala等计算引擎，为湖内数据的集成、处理、分析等提供丰富的计算环境，主要包括可支持读写分离的数据仓库中间件、数据ETL/ELT过程所需的批处理计算引擎、实时分析所需的流计算引擎、内存计算以及可支持高级分析的机器学习库。

3.2.4 数据服务层

已处理数据通过数据服务层满足下游用户消费需求，主要包括基于SQL技术的数据仓库服务、基于NoSQL技术的数据接口（API）和实时数据服务以及基于数据共享技术的共享数据服务。

（1）SQL技术：根据系统的查询性能和成本优化，主要可采用两种架构类型的SQL数据库作为服务层。其一是对称多处理（SMP）架构，包括SQL Server、Oracle、MySQL等，此类架构优势在于高速度、低延迟、少故障及事务管理，但其高耦合结构只适用于数据量较少的场景，当数据以PB级别扩张，则需应用大规模并行处理（MPP）架构，包括Azure Synapse、Amazon Redshift等。该架构将数据按区块划分且并行独立处理，同时支持系统的垂直和水平扩展，适用于大数据场景。

（2）NoSQL技术：包括文本数据库（如MangoDB），具有高灵活性和可扩展性且支持实时数据访问、键值存储，简化查询，用数据缓存方式降低访问延迟、提高吞吐量并实现类似于关系数据库的宽列存储，通过表内可灵活调整的列结构和数据库写入模式优化查询

速度。

（3）数据共享技术：规定数据发布过程的相关条款和使用条件、共享过程的访问管理和请求审批，确保数据以受控和结构化方式在企业内部各领域或外部实体间流转。

按照湖仓一体内数据的来源、数据类型以及数据消费群体等，可主要划分4类数据服务：①数据仓库服务不仅可存储在线和历史数据，还可以支持BI和报告，为业务分析需求提供数据查询平台，同时可作为下游数据集市的数据源；②实时数据服务对接下游应用系统，例如在线移动系统、客户关系管理（CRM）系统、网站推荐引擎等；③基于API的服务通过各类API与外部服务进行交互，通常以JSON文件类型为主，此类服务可扩展性最高；④数据共享服务提供多源异构数据系统间数据共享所需的控制方式和策略，支持数据以结构化方式共享，该服务通常基于API进行。

3.2.5 数据分析层

数据分析是将数据转化为洞察力的过程。湖仓一体中可能存在的数据分析类型包括按需查询、商业报告、自助BI、数据湖探索等在内的描述性分析，以及数据科学、机器学习等高级分析。

（1）描述性分析：按需查询以业务需求为导向，理解潜在数据模式并创建SQL数据表以查询获取数据、执行分析；商业报告根据具体要求定期生成，并以移动APP等渠道分发给相关用户，报告通过挖掘服务层中的数据和预先定义的标准化报告格式来创建，提供了对多业务领域历史和当前的看法以及跨功能维度的分析汇总；自助BI为功能用户摆脱技术依赖进行分析提供可能，不同用户可依托自助BI对分析报告进行切片以创建不同数据视图，满足个性化分析需求；基于SQL的工具无法支撑数据湖探索工作，应由技术型用户探索数据特征生成特征集，并建立针对性数据模型，例如Jupyter Notebook为典型的开源数据湖探索工具[9]。

（2）高级分析：数据科学、机器学习等高级分析以算法为支撑，用历史数据集来获得预测结果或从数据中提取复杂数学关系来产生洞察力。

数据分析层通过分析沙盒、BI服务、数据科学和机器学习等组件，支撑多种类型的分析活动。

（1）分析沙盒：根据执行分析的类型创建按需计算的集群，例如SQL集群和Spark集群等，并通过如Jupyter Notebook等交互式工具与数据湖或数据仓库交互，用其存储的大量数据进行特别分析或假设推演分析，沙盒可支持按需查询和数据湖探索[9]。

（2）BI服务：具有查看、分析和理解数据等能力，并辅助关键决策制定，为企业提供强有力的分析平台，可支持商业报告和自助BI。商业报告以标准化格式为主，例如预制报告和可视化仪表盘等；自助BI可实现在线分析处理（OLAP）功能，以测量和维度的形式创建用户友好型数据描述[3]。

（3）数据科学和机器学习系统：基本支持湖格式数据的直接读取以进行有效访问，同时声明性DataFrame API可对机器学习工作负载中的数据访问进行查询优化。类似分析沙盒，机器学习系统由计算层机器学习库获取计算资源并创建按需计算的实例，以执行相关探索性数据分析、特征工程、模型训练和测试、可视化模型开发等[3]。

3.2.6 数据治理层

数据治理为相关数据管理活动提供可靠遵循规范，避免系统沦为数据沼泽。数据治理层以保证湖仓一体数据的可靠性、可访问性和高质量为目标，从内容和过程控制的角度可划分为数据治理策略、数据生命周期管理、元数据管理、主数据管理、数据质量管理和数据安全管理[15]。

（1）数据治理策略：是确保企业数据和信息资产得到一致管理和正确使用的章程，定义数据的可用性、完整性和一致性等参数，在数据发布方式和消费方式上保持一致，同时建立数据架构、标准定义和版本控制等原则，这些原则随企业成熟度、愿景变化而变化。为企业制定数据治理度量评估体系，依托评估工具，根据相关度量维度和规则为治理结果打分并提出改进计划[9]。

（2）数据生命周期管理：持续监管数据由产生到消亡的价值属性，并实时反馈，使企业掌握当前数据状态。此工作需依据基于完整的元数据信息获取的数据特征和关联信息来完成，并创建索引来跟踪特定数据及其关联数据。

（3）元数据管理：可为湖内数据提供完整性描述以保持其可操作性，以元数据为基础形成数据目录是避免数据沼泽化的关键。按蕴含信息和功能属性可将元数据划分为技术元数据、操作元数据和业务元数据[4]，需

要元数据捕获引擎、图谱和搜索引擎等组件支持编制数据目录，分别实现元数据的定期自动扫描捕获、自动编目形成可视化关系图以及提供访问窗口[9]。

（4）主数据管理：针对关键业务实体（如员工、客户、产品、供应商等）建立统一视图以在相关数据存储中获得唯一实体识别，并对已识别主数据按数据规范要求进行治理和互联网技术（IT）改造，其目标在于提供准确、及时、完整的主数据来源，保证在企业范围内重要业务实体数据的一致性（定义和实际物理数据一致），以支持企业内部决策分析和业务流程再造、企业业务流和工具链的打通和串联。

（5）数据质量管理：数据治理层的最重要环节，数据质量可定义为数据适合实际使用的程度，高质量的数据是企业做出正确决策的基础。质量管理的前提是定义同时适用于数据湖层和数据服务层的质量规则，依据规则划分质量等级（以数据成本和价值效益为参考），按规则实时监测数据的完整性、时效性、唯一性、一致性、有效性和精确性等并进行打分，以评估企业当前数据质量水平，打分结果为提出质量改进计划和重新定义质量规则提供可靠参考。

（6）数据安全管理：通过身份和访问管理（IAM）对访问用户进行授权和认证，以确保湖仓数据按需访问的安全性，可以阻止恶意访问并通过基于风险的访问控制、身份保护和认证工具来保证证书的完整性，同时不会干扰正常的访问过程。通过数据加密将信息编码，包括对称加密和非对称加密，加密过程以算法（如高级加密标准等）和各类加密工具为基础，以保护存储在云服务器中的数据并防止多种网络攻击。通过数据屏蔽实现数据加密但保持其可读性（仅适用于特定的敏感数据元素），并在需要时将屏蔽解除，以静态数据屏蔽（SDM）和动态数据屏蔽（DDM）方法为主。除身份和访问控制之外，湖仓一体外部网络和内部各层级之间数据流动的安全性可通过虚拟网络、防火墙、虚拟专用网络（VPN）等技术得到保障[4,9]。

3.3 用户

依据数据分析和服务的主题可将用户分为技术用户和功能用户：前者是在特定领域具有技术能力的利益相关者，包括数据分析师、数据科学家等；后者是拥有特定领域专业知识的利益相关者，例如管理人员和各类信息系统等。湖仓一体确保所有类型数据原始格式和转换格式的可访问性以满足各类用户需求。

3.3.1 技术型用户

数据科学家专注于所有数据，在特定平台创建和测试数据模型；数据分析师受业务驱动，专注于清洗和处理后的数据，通过查询、汇总和切片数据来完成分析任务。

3.3.2 功能型用户

企业管理人员通过BI工具，由报告系统获取业务报告用于业务决策。报告系统定期获取数据并生成报告，为订阅者提供预定、临时和自助式服务。下游应用系统可能是OLTP系统、其他系统中的数据仓库或数据湖，这些系统定期提取处理后的信息或采用一定机制将信息推送至最终用户。湖仓一体通过大量API向各类内、外部系统提供数据服务，基于API的数据消费可满足特定系统的个性化交付需求，具有较高的可扩展性。数据共享系统为用户提供开放可控的数据市场。

4 流批一体数据流转过程

各业务领域湖仓一体功能模块各层级间依据一定规则实现数据流入流出。由于数据仓库的ETL或数据湖的ELT过程均难以满足湖仓一体内部数据流转需要，采用一种提取/加载/转换/加载（ELTL）方法实现流批一体的数据采集、存储和处理[9]，如图3所示。

4.1 批量数据过程

原格式数据通过批量采集引擎以推/拉方式加载到数据湖原始数据区并长期保存，采集服务按需提供，在数据源可执行采集操作时被激活，数据推送方法包括文件传输协议（FTP）和编程语言等，数据拉取方法包括开放数据库链接或JAVA数据库链接等。原始数据会加载到批处理引擎，由于湖内数据的多源异构属性，以分布式批处理方式为主：制定分布策略，将数据集划分为多块，包括散列分布和循环分布等，并为每一块数据子集配置独立计算单元以实施转换过程。分布式批处理支持计算单位的水平和垂直扩展来加速处理，在数据处理过程中，中间数据被保留，已处理数据被重新

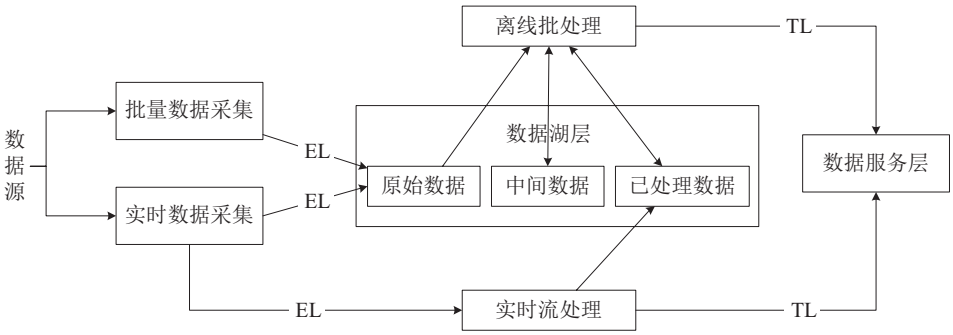

图3 流批一体数据流转过程

合并为统一数据集保存在已处理数据区并按需加载到数据服务层，中间数据区和已处理数据区可与批处理引擎进行多次交互。

4.2 实时数据过程

实时数据来自社交媒体平台、物联网等流数据源，主要价值在于时效性，实时分析多在数据上直接进行，因此无须长期存储。实时数据通过事件发布/订阅服务等消息中间件采集到原始数据区用作备份，并按需加载到流处理引擎。流数据的转换过程可分为微批处理和执行阶段，微批处理负责从流数据中提取较小数据集并转换，这个阶段基于时间和事件。完成微批处理创建后需在其上执行处理任务，包括分组任务（如过滤、聚合、排序、连接等）以及某一特定事件的事件导向型任务，经处理的数据将被第一时间推送至数据服务层并载入已处理数据区[9]。

5 结论

数据湖向湖仓一体的过渡代表数据平台向管理结构化、系统化方向转变。分布式湖仓一体意在通过业务数据解耦使单体数据架构向可扩展架构转变，这种转变更是海量分布数据环境下数据治理思维方式的升级。通过比较三代数据平台差异并分析当前湖仓一体面临的挑战，提出分布式湖仓一体架构，构建完整数据生命周期下的湖仓一体功能模块和数据流转过程。下一步，将依托相关技术研究该架构落地方法以供相关研究或企业参考。

参考文献

[1] JANSSEN N E. The evolution of data storage architectures: examining the value of the data Lakehouse [D]. Enschede: University of Twente, 2022.

[2] Databricks. Data Lakehouse [EB/OL]. [2022-11-03]. https://www.databricks. com/glossary/data-lakehouse.

[3] ARMBRUST M, GHODSI A, XIN R, et al. Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics [EB/OL]. [2022-11-08]. https://cs.stanford.edu/people/matei/papers/2021/cidr_lakehouse.pdf.

[4] 陈氢，张治. 融合多源异构数据治理的数据湖架构研究 [J]. 情报杂志，2022，41（5）：139-145.

[5] OREŠČANIN D, HLUPIĆ T. Data Lakehouse-a novel step in analytics architecture [C] //2021 44th International Convention on Information, Communication and Electronic Technology（MIPRO）, 2021: 1242-1246.

[6] SHIYAL B. Beginning Azure Synapse Analytics [M]. Berkeley: Apress, 2021: 21-48.

[7] DEHGHANI Z. How to move beyond a monolithic data lake to a distributed data mesh [EB/OL]. [2022-11-03]. https://martinfowler.com/artic-les/data-monolith-to-mesh.html.

[8] DEHGHANI Z. Data mesh principles and logical architecture [EB/OL]. [2022-11-03]. https://martinfowler.com/articles/data-mesh-principles.html.

[9] MENON P. Data Lakehouse in Action [M]. Birmingham: Packt Publishing, Incorporated, 2022: 171-176.

[10] MACHADO I, COSTA C, SANTOS M Y. Data-driven information systems: the data mesh paradigm shift [C] //29th International Conference on Information Systems Development（ISD2021 Valencia, Spain）, 2021: 67-73.

[11] GOEDEGEBUURE A A. Data mesh：systematic gray literature study，reference architecture，and cloud-based instantiation at ASML [D]. Tilburg：Tilburg University，2022.

[12] HOKKANEN S. Utilization of data mesh framework as a part of organization’s data management [D]. Eastern Finland：University of Eastern Finland，2021.

[13] DEHGHANI Z. Data Mesh [M]. California：O’Reilly Media，Incorporated，2022：31-60.

[14] MACHADO I A，COSTA C，SANTOS M Y. Data mesh：concepts and principles of a paradigm shift in data architectures [J]. Procedia Computer Science，2022，196：263-271.

[15] 张宁，袁勤俭. 数据治理研究述评 [J]. 情报杂志，2017，36（5）：129-134，163.

作者简介

陈氢，女，1968年生，博士，教授，研究方向：信息资源管理、数据治理，E-mail：cq29cn@126.com。
宋仕伟，男，1997年生，硕士研究生，研究方向：数据治理。

Lakehouse Architecture Under Perspective of Data Governance

CHEN Qing SONG ShiWei
(School of Economics and Management, Hubei University of Technology, Wuhan 430068, P. R. China)

Abstract: The huge amount of distributed heterogeneous data brings serious challenges to enterprise data governance and accelerates the transformation of data warehouse and data lake to lakehouse which combines the functions of both. We compare the differences between data warehouse, data lake, and lakehouse, analyze the advantages of lakehouse and its challenges, build a distributed lakehouse architecture by dividing business areas and mapping them to data perspective, synthesize the existing research and related technologies to build a functional module of the lakehouse, and explain the flow batch all-in-one data flow process. The distributed lakehouse architecture includes the concepts of data domain decoupling, cross-domain data sharing, and federated data governance. The lakehouse functional modules mainly include data sources, lakehouse core functional areas, and users. The flow batch all-in-one data flow process includes batch data process and real-time data process. This study integrates effective data governance process for lakehouse integration and builds a more complete architecture system for lakehouse integration to provide references for related research or enterprises.

Keywords: Data Governance; Data Lake; Lakehouse; Data Sharing

（责任编辑：王玮）