摘要

数据仓库和相关技术允许业务用户实时的做出以数据为依据的决策。它有效地处理和存储机构的用于数据分析的关键历史数据集。来自交易源系统的最新数据通过提取，转换和加载（ETL）流程存储到数据仓库中。传统上，ETL被安排为夜间批处理过程，以在非工作时间刷新数据。但是，随着科技的进步，24X7全天候业务运营模式，市场竞争以及引入现代数据源已迫使组织采取以下行动：更频繁地刷新其数据仓库，以提供近乎-实时环境。传统的ETL流程难以应对具有近实时环境的要求，因为它没有为此目的进行设计和优化。结果，更先进的ETL版本才能满足近乎实时环境，例如低延迟，高可用性和高扩展性。本文提供了实时ETL每个阶段的机会和挑战，它也描述了一些常用技术在行业中的实践，以实现接近实时的ETL。通过复盘实时ETL中有价值的研究确定实时ETL挑战和解决方案。

一. 介绍

根据Bill Inmon的定义，数据仓库是“一个主题面向，集成，时变和非易失性集合支持管理层决策的数据” 。换句话说，它是组织数据的集中存储库，专为查询和数据分析而设计，可帮助企业领导者做出以数据为依据的决策。它集成自各种源系统的数据，例如交易系统，客户关系管理（CRM）数据库，企业资源计划（ERP）系统等等。一组业务转换应用于源数据，使它更符合报告要求将其存储在数据仓库中。使用提取转换加载刷新数据仓库（ETL）流程。顾名思义，ETL具有三个主要方面阶段，每个阶段都照顾整体的特定部分数据刷新过程。提取阶段从异构数据系统读取数据并将其存储到中间存储中。转换阶段读取提取的数据，开展业务规则和基本数据操作，例如清理，数据类型转换等。加载阶段最终将转换后的数据加载到目标数据仓库。我们将在第二部分讨论ETL更详细的过程。通常，数据仓库是通过按预定义的时间执行ETL流程，每天晚上刷新。这样可以确保用户可以在工作时间和ETL流程中使用最新的数据，无需与用户查询竞争资源。少数其他因素，例如业务需求最近的数据，数据仓库的处理能力和源数据的性质也可能会产生影响关于数据刷新的频率。

如今，为了节省成本并在竞争中脱颖而出,全球各地的组织都在采取新举措使他们的业务流程更加完善。数据仓库在这些计划中起着关键作用因为它为分析和决策提供了必要的数据过程。但是，这些计划需要数据仓库更新频率更高，以获取最新数据进行分析。此外，还引入了现代数据源，例如传感器，闭路电视，Web反馈等。比传统的夜间批处理过程更频繁地读取更多的数据，因为数据量太大，无法在单个ETL运行中处理，有时源数据仅保留一小段时间，这意味着直到下一个批处理过程，它才能在源上提供，这使得ETL流程容易出现数据差异问题。同样，大多数大型组织的办公室遍布在世界上的多个国家/地区，这意味着用户正在积极地使用24X7的数据仓库。由于这种范例数据处理和分析要求的变化，组织将数据仓库更多地视为“活动”或“实时”数据仓库，而不是作为历史数据的系统分析，并期望更频繁地刷新数据，例如每小时或每隔几分钟。自然，传统的ETL还需要更改流程以处理较少的数据量快速接连以使其更接近实时数据仓库环境。

从近实时视角分析传统ETL流程视角已成为数仓研究中的热门话题，大多数研究专注于传统ETL中的特定问题，并提供了具体的替代方法以使其适用于近乎实时的环境。本文的重点是提供在每个阶段对挑战进行分类并提供可用的解决传统ETL的方案，并描述了一些用于科技行业实现近实时ETL的技术。

本文的其余部分安排如下。第二节详细的描述了ETL过程的提取，转换和加载阶段。第三部分介绍了近实时ETL，描述它的一些关键方面，并讨论它与传统ETL的不同之处。第四节介绍了与此相关的研究领域。第五节讨论了近实时环境中的ETL过程在每个阶段的挑战。第六节讨论针对确定的挑战的解决方案。第七节讨论近实时ETL在行业中的实践，以及第八节本文总结。

二. 传统 ETL

ETL流程是数据仓库环境的骨干因为它做了繁重的工作，将数据从源中移出到数据仓库架构中。在本节中，我们将详细讨论ETL流程的每个阶段。

A.提取

提取阶段的主要功能是查询数据源系统，提取数据存储在中间存储以供进一步处理。根据数据源系统的性质，提取阶段使用本机数据库连接器，文件传输协议（FTP），应用程序编程接口（API）请求等从源读取最新数据。有一个有效的提取策略是至关重要的，以便提取转换所需的数据集，作为不适当的数据集可能会对ETL流程的性能产生不利影响。传统上，以下两种提取策略用于数据仓库环境。

- 完整提取：在这种方法中，在ETL运行期间整个源数据提取并转储到中间存储。这是最简单的策略，因为它没有需要任何机制来标识更改的记录。它适用于记录数量较少的来源。但是，建议不要将其用于高大量数据源，提取相同的记录集将数据加载到其中时会产生不必要的开销数据仓库。此外，源数据的增长超时影响提取过程生成的数据集的数量,最终影响整个ETL流程。

- 增量提取：在这种方法中，只有新记录，即自上次以来已更改的记录，提取并将其存储为ETL流程。它使用更改数据捕获（CDC）技术，例如基于时间戳，基于日志或基于事件的提取以识别更改的记录。这种方法的主要优点之一是生成用于加载的最佳数据集，这有助于加快进一步处理速度，避免不必要的处理加载期间的间接消耗。但是，CDC技术需要其他组件，例如日志表或一组参数，以跟踪上一次提取过程。这是最常用的提取方法，因为它在每个过程中生成相似的，针对特定来源的数据集，有助于优化整个ETL流程。

B.转换

ETL流程的核心功能是在此阶段实现的。转换阶段从中间存储中读取提取的数据，然后通过执行各种数据清理操作并对其应用指定的业务规则，将其转换为适合加载的格式。通常，所有转换都以顺序方式应用，并且每次转换的数据都存储到一个临时表中，该表用作下一个转换的源，该中间存储通常称为暂存区域。业务需求和不同源系统之间的差异决定了应用于提取数据的转换类型和数量。转换过程的主要目的是使源数据符合业务规则，并将其存储在暂存区中以进行加载。有时，很少有源记录在此过程中被拒绝，因为它们不满足一个或多个业务规则或转换。此类记录通常存储在单独的表中，业务用户会对其进行分析以识别任何源数据差异或修改转换规则以接受这些记录。另外，还定期检查关键转换任务（例如Join）的性能，以确保尽可能优化它们，以避免整个ETL过程中的任何延迟。

C.加载

ETL处理的最后阶段，即加载，将转换后的数据从暂存区移至目标数据仓库表。加载阶段使用数据操纵语言（DML）操作（例如截断，更新，插入和合并）将新记录添加到数据仓库表中。每个数据仓库环境都有一个加载策略，该策略确定如何将数据从上一个存储转移到目标表。这是受各种因素影响的策略，例如ETL流程的总体设计，数据量，加载计划和目标表的性质。由于加载阶段直接与数据仓库表交互，其他数据仓库流程和业务报告查询也使用该表，因此必须设计一种有效的加载策略，以最大程度地减少对数据仓库环境的影响并确保其不会延迟数据仓库的运行。

近实时ETL

A.动机

多年来，数据仓库环境已帮助业务用户制定数据驱动型决策，以解决业务问题并提高运营效率。业务用户查询数据仓库中的历史数据，该数据每天，每周或在某些情况下每月更新一次。但是，最近业务模式的变化（例如24X7操作）以及来自其他组织的激烈竞争，为了赢得市场份额，使企业更加频繁地做出这些决策，这需要比传统的夜间批处理负载更频繁地刷新数据仓库数据。例如，航空公司在其机票价格计算系统中采用了动态定价模型，该模型会根据特定旅行的座位数量来更改机票价格，这样的系统要求从交易系统中不断更新报告环境以进行必要的计算。另外，引入诸如人工智能和数据挖掘工具之类的先进技术来执行诸如贷款批准或背景验证之类的业务操作，使得在数据仓库中拥有最新数据至关重要，因为它为这些应用程序提供了所需

的数据。除此之外，诸如Web feed，CCTV流和传感器之类的现代数据源使下游过程必须定期提取其数据。这些系统连续生成数据，这些数据会在源上维护一会儿，然后再替换为新的数据集。

B.关键属性
近实时环境的主要重点是使源数据尽快在数据仓库环境中可用，并为用户提供所需的数据集以执行高级分析。为此，近实时环境应具有以下关键特征。

- 低延迟
- 高可用性
- 最小干扰
- 高可伸缩性

近实时ETL的挑战

A.提取

提取过程的主要功能之一是从多个源系统中识别和提取最新数据，并将其用于进一步处理。在传统的ETL中，源系统每天都会生成批处理文件/负载，并通过每晚刷新将其更新到数据仓库。但是，在接近实时ETL的情况下，一天中会提取多次，这需要使用有效的更改数据捕获技术来识别和提取自上次提取以来已更改的数据。同样，拥有多个源系统会使提取过程更加复杂，因为相同的变更数据捕获技术由于其异构性可能无法从所有数据中提取数据

一些现代数据源，例如传感器，CCTV，网络馈送等，不断生成大量数据，通常可在短时间内提取这些数据，然后再将其替换为新数据。传统的提取过程无法应对此类数据源，因为它必须等待下游转换和加载阶段完成才能再次运行提取过程，这对于某些数据源而言可能为时已晚，并可能导致数据丢失。同样，在每次ETL运行期间处理大量记录可能会降低ETL处理的性能，并影响整个数据仓库环境。

B.转换

为了执行复杂的转换，将整个转换任务分为多个子任务，每个子任务对源数据执行专门的操作，并将其传递给下一个子任务以进行进一步的操作。这种方法简化了转换过程，并使得在出现任何数据问题或故障时易于分析。但是，此策略需要一个中间区域来存储每个子任务的输出，因为下游子任务会将其用作输入。在数据仓库术语中，此中间存储区域称为暂存区域，用于在所有必需的转换完成并将最终的转换数据加载到表示区域以进行报告之前临时存储转换的数据。在传统的ETL环境中，由于源数据量很大，专用数据库或某些情况下，专用服务器用作暂存区域，以在转换过程的各个阶段存储源数据的副本，并确保转换过程不会影响报表数据库的性能。对近实时ETL使用类似的暂存区策略会产生维护开销，因为对近实时ETL的数据存储要求与传统ETL不同，后者在每次ETL运行期间仅处理有限数量的记录。另外，不能将暂存区从ETL流程中完全删除，因为需要执行所需的转换，这给在近实时环境中有效利用暂存区提出了挑战。

C.加载

加载阶段的主要功能是将转换后的数据从暂存区移到数据仓库报告表中。根据ETL流程的设计，可以使用各种数据加载策略（例如update-insert，delete-insert或truncate-load）将数据加载到目标表中。同样，其他依赖的数据库对象（如实例化视图和索引）也将刷新，以确保它们具有表中的最新数据。所有这些策略都需要对目标表的独占访问权，这意味着没有其他进程可以更新，并且在某些情况下，可以在更新表时从表中读取数据。在传统的数据仓库环境中，ETL流程计划在非高峰时间运行，因此不会发生用户启动的活动，例如报告查询，以及来自其他数据仓库过程的最少活动，这为数据加载过程提供了足够的窗口来更新所需的表而又不影响其他过程。但是，在接近实时的环境中，ETL过程全天运行并定期更新数据仓库，这导致了 资源争用问题，因为加载过程必须与用户启动的过程竞争才能获得对报告表的独占访问权限，这可能会延迟ETL过程并最终影响数据仓库和用户报告查询的性能。

近实时ETL的机遇

A. 提取

1.数据流控制器

在近实时的ETL环境中，提取阶段面临的主要挑战是在不中断源系统且不影响下游ETL流程性能的情况下，定期处理事务数据。多项研究引起了人们的关注，并提出了各种解决方案。所提出的解决方案可以分为以下两种方法。

- 数据缓冲
- 有条件提取

2.流处理

处理来自现代数据源（例如传感器，CCTV，Web Feed等）的数据的能力是近实时数据仓库环境的最基本要求之一，因为这些是此类环境中常用的源系统。各种研究已经分析了此问题，并建议使用数据流处理作为解决该问题的可能解决方案。流处理是一种计算模型，它允许以更有效的方式对数据执行转换。在这样的体系结构中，源系统生成连续的数据流，该数据流由下游应用程序“即时”使用和处理，这意味着数据在接收到后就立即进行了转换，而没有实际存储。在数据仓库环境中，数据流处理是通过使用诸如Apache Kafka之类的应用程序来实现的，其中源系统将数据发布到数据队列中，该队列在预定的时间段内保持数据的副本，然后将数据从队列中删除。

B. 转换

- 缓存加入 cache
- 动态暂存区
- 数据存储库

C. 加载

1.为近实时数据分离ETL

提高传统ETL过程的效率以实现近实时ETL一直是该领域许多研究的重点，作者提出了各种减少ETL过程运行时间的技术，以增加ETL过程的运行时间。但是，一项研究提出了一种解决方案，用于将数据仓库中的传统/历史数据和近实时数据分开，并建议为它们设置专用的ETL过程。在这种方法中，传统的ETL过程继续以批处理模式运行，而近实时ETL利用CDC技术以更频繁的时间间隔识别和加载最新数据。使用不同的表集来存储历史和近实时数据，用户可以根据其报告要求对其进行访问。

2.基于意义的加载

在近实时环境中，确保仅将所需数据作为ETL流程的一部分进行加载非常重要，因为加载不必要数据的开销可能会延迟整个数据刷新周期。基于重要性的数据加载通常用于解决此问题的技术。在这种方法中，数据的重要性（即重要性）是在将其加载到数据仓库之前确定的，并且仅加载具有高于特定阈值的重要性的数据。

3.数据聚合技术

数据量直接影响数据加载过程的性能，因为大量的记录导致更多的写I / O操作。近实时环境有关的多项研究对此问题进行了分析，并提出了一些解决方案，例如基于重要性的加载以及在每次ETL运行中仅加载固定数量的记录，以减少要加载到数据仓库中的数据量。解决此问题的另一种解决方案是利用数据聚合技术，通过将一个或多个维/业务键的记录聚合在一起，数据聚合可以显着减少记录的数量。例如，可以加载一天或业务单位级别的数据，而不是加载可能数以百万计的单个销售交易，这可以将其减少到仅几千条记录。

业界的实践

在本节中，我们将讨论技术行业中用于实现近实时ETL的一些实践，还将提供一些有关实现这些实践的常用工具和技术的见解。

A.微批次ETL

微批次ETL是传统ETL过程的演进版本，更适合于近实时环境。这种方法侧重于以规则的间隔处理小块数据，而不是在夜间批处理模式下处理整个源数据集。它使用CDC技术从源中提取最新数据，然后每小时或有时每隔几分钟将其加载到数据仓库中。因此ETL流程不会影响数据仓库和OLAP查询性能。消息队列应用程序（例如Apache Kafka或RabbitMQ）用于存储来自源系统的数据，该数据由微批量ETL定期读取，Apache Kafka是流数据集的首选选项。但是，微批处理ETL不能提供纯实时ETL，因为源数据仓库和数据仓库数据之间仍然存在几分钟或几小时的延迟。

B.采用云技术

云计算技术的最新进展使其成为许多组织将其数据仓库系统从本地迁移到云的可行选择，从而以可承受的成本获得更高的性能。云环境提供了可配置资源池，例如可以通过Internet访问的存储，网络和计算能力。基本的基础设施是由提供的云服务来设置和管理的，而客户只需为使用的商品付费。它提供了多种服务模型，例如平台即服务（PaaS），基础架构即服务（IaaS）和软件即服务（SaaS），并且可以部署在不同的配置中，例如公共云，私有云和混合云。在近实时环境中使用DWaaS具有两个主要优势；首先，这些环境具有高度可扩展性，可以轻松适应近实时环境的数据量要求；其次，作为云产品，它们始终与现代数据源（例如流数据集）所需的必要连接器和驱动程序保持同步。使ETL流程的设置比本地环境容易得多。Amazon Redshift，Google BigQuery，Microsoft Azure SQL服务器和Snowflake DB是一些最受欢迎的DWaaS平台。

C.大数据平台

大数据平台（例如Apache Hadoop）提供了一个分布式框架，用于在高度可扩展的商品硬件集群上存储和处理大量数据。它使用MapReduce编程模型将作业拆分为多个任务，这些任务以并行方式在集群上执行。此方法不同于基于大规模并行处理（MPP）体系结构的传统数据库管理系统（例如Teradata或Vertica）。重要的是要注意，Hadoop不一定是现有MPP系统的替代品，因为在典型DW查询的查询执行时间上，大多数MPP系统的性能均优于Hadoop。但是，它可以用作引擎以分布式方式处理大量数据。一些组织（例如Facebook）拥有Hadoop集群来处理PB级数据。

总结

数据仓库环境在组织的决策支持系统中起着重要的作用，因为它为业务用户提供了必要的数据和平台，以使他们能够进行数据驱动的决策。ETL流程执行必需的任务，以从事务源系统中提取，转换和加载最新数据，并将其加载到目标数据仓库环境。传统上，ETL按计划按夜间批处理流程运行，这使它可以执行所需的数据刷新操作，而不会中断事务源系统，也不会影响数据仓库用户的服务质量。业务用户使用这些环境中的数据进行历史数据分析，例如暂存区和对主数据的管理被确定为主要挑战，而将高速缓存与联接运算符，动态暂存区和主数据存储库一起使用被确定为解决这些问题的解决方案。影响数据仓库环境的性能并在OLAP查询中引入不一致是加载阶段的主要挑战，同时将单独的ETL用于近实时，基于重要性的加载，数据聚合技术以及为OLAP复制数据仓库模式被确定为：解决这些问题的解决方案。带有Apache Kafka等流处理器的微批次ETL，使用供应商提供的工具采用ELT方法，使用领先的云服务提供商（例如Amazon）提供的数据仓库即服务（DWaaS）迁移到云。