ABSTRACT

大规模实时分析应用程序（实时库存/定价，移动应用程序推荐，欺诈检测，风险分析，物联网等）的普及率不断提高。这些应用程序需要分布式数据管理系统，可以处理快速并发事务（OLTP）和最近数据的分析。其中一些甚至需要运行分析查询（OLAP）作为事务的一部分。然而，在构建数据管理系统时，对各个事务和分析请求的有效处理会导致不同的优化和架构决策。

对于需要分析和交易的数据处理类型，Gartner最近创造了术语 混合交易/分析处理（HTAP）。许多HTAP解决方案都来自行业以及针对这些新应用的学术界。虽然其中一些是单一系统解决方案，但其他一些是OLTP数据库或NoSQL系统与分析大数据平台（如Spark）的松散耦合。本教程的目标是1-）快速回顾OLTP和OLAP系统的历史进展，2）讨论HTAP的驱动因素，最后3）提供对现有和新兴HTAP解决方案的深入技术分析，详细说明关键的建筑差异和贸易。

1.INRODUCTINO

在本教程中，我们计划调查现有和新兴的HTAP（混合交易和分析处理）解决方案。HTAP是由Gartner创建的术语，用于描述可在单个事务中同时支持OLTP（联机事务处理）和OLAP（联机分析处理）的系统。 但是，HTAP这一术语目前使用得更为广泛，即使对于支持插入（不一定是ACID事务）以及OLAP查询的解决方案也是如此。 其中一些系统能够对最近的数据运行分析查询，而其他系统在查询查看最新数据之前需要一些延迟。

为了理解HTAP，我们首先需要研究OLTP和OLAP系统以及它们多年来的发展历程。 关系数据库已用于事务处理和分析。 但是，OLTP和OLAP系统具有非常不同的特性。 OLTP系统由其各自的记录插入/删除/更新语句以及从索引中获益的点查询来识别。 没有索引支持，人们无法想到OLTP系统。 另一方面，OLAP系统分批更新，通常需要扫描表。 批量插入OLAP系统是ETL（提取转换负载）系统的工件，它将事务数据从OLTP系统整合并转换为OLAP环境以进行分析。

在Stonebraker [34]争论多个专业系统的开创性论文之后，数据库领域出现了一系列专门的面向列的OLAP系统，如BLU [30]，Vertica [23]，ParAccel，GreenPlumDB，Vectorvise等。 尽可能多的内存中OLTP系统，包括VoltDB [35]，Hekaton [13]，MemSQL [24]等。 数据库引擎中这种重新应对的主要驱动力是现代硬件的进步。 第二代OLAP和OLTP系统更好地利用了多核，各种级别的内存缓存和大型内存。

与此同时，在过去十年中，新一代应用推动了许多大数据技术的爆炸式增长。 NoSQL或键值存储，例如Voldemort [32]，Cassandra [8]，RocksDB [31]，快速插入和查找，以及非常高的扩展，但缺乏查询功能，并且只有松散的事务保证（ 参见Mohan的教程[25]）。

还有许多SQL-on-Hadoop [10]，包括Hive [36]，Big SQL [15]，Impala [20]和Spark SQL [3]，它们提供大数据集的分析功能，重点关注 仅OLAP查询，缺少事务支持。 虽然所有这些系统都支持对文本和CSV文件的查询，但它们的重点是柱状存储格式，如ORCFile [27]和Parquet [1]。

近年来已经需要更多的实时分析。 此外，移动和物联网已经产生了新一代的应用程序，其特征在于大量的摄取率，即它们在短时间内产生大量数据，以及它们需要更多的实时分析。 企业正在推动对其数据进行更实时的分析以提高竞争优势，因此他们需要能够尽快对其运营数据进行分析。

随着这些发展，现在有很多兴趣，研究重点是在大数据集上提供HTAP解决方案。 在本教程中，我们计划提供快速历史视角，了解不同技术的进展，并讨论当前的HTAP解决方案。我们将研究当前解决方案的不同架构方面，确定它们的优点和缺点。 我们将按照许多技术方面对现有系统进行分类，并为一些代表性系统提供深度潜水。 最后，我们将讨论现有的研究挑战，以实现真正的HTAP，其中单个事务可以包含插入/更新/删除语句，以及复杂的OLAP查询。

2. HTAP SOLUTIONS:DESIGN OPTIONS

HTAP解决方案如今遵循各种设计实践。 本教程的这一部分重点介绍了他们的主要trade-offs，同时举例说明了工业领域和学术解决方案。 HTAP系统必须做出的主要设计决策之一是，是否对OLTP和OLAP请求使用相同的引擎。

2.1 Single System for OLTP and OLAP

传统的关系数据库（例如，DB2，Oracle，Microsoft SQL Server）能够使用单一类型的数据组织（主要是行存储）在一个引擎中支持OLTP和OLAP。 但是，它们对这些工作负载中的任何一个都不是非常有效。

因此，遵循一个规模并不是所有规则[34]，过去十年中，OLTP和OLAP专用引擎的兴起正在利用现代硬件的进步（更大的主存储器，多核等）。 各种供应商和学术团体已经建立了内存优化的行存储（例如，VoltDB [35]，Hekaton [13]，MemSQL [24]，Silo [37]，...）和列存储（例如，MonetDB [7]，Vertica [23]，BLU [30]，SAP HANA [14]，...）分别专门用于事务和分析处理。 这些系统偏离了传统的关系数据库代码库，并从头开始构建更精简的引擎，以避免传统引擎的大量指令占用。

但是，许多系统针对一种类型的处理进行了优化，后来又开始添加对其他类型的支持以支持HTAP。 这些系统主要基于他们用于交易和分析请求的数据组织。

2.1.1 Using Separate Data Organization for OLTP and OLAP

SAP HANA [14]或Oracle的TimesTen [22]的引擎主要针对内存中的列式处理进行了优化，这对OLAP工作负载更为有利。 这些系统还支持ACID事务。 但是，他们使用不同的数据

组织进行数据提取（逐行）和分析（列式）。

相反，MemSQL [24]的引擎主要是为可扩展的内存中OLTP而设计的，但今天它也支持快速分析查询。 它以行格式摄取数据，并以行格式保存数据的内存部分。 将数据写入磁盘时，会将其转换为列式格式以便更快地进行分析。 同样，IBM dashDB [11]是传统行存储向HTAP系统的演进，分别用于OLTP和OLAP工作负载的混合行和列数据组织。

另一方面，HyPer [19]从一开始就旨在使用一个引擎支持快速交易和分析。 尽管最初它使用OLTP和OLAP的数据的逐行处理，但今天它还提供了选择柱状格式以便能够更有效地运行分析请求的选项。

最后，Pelaton最近的学术项目[28]旨在构建一个自主的内存HTAP系统。 它提供自适应数据组织[4]，它根据请求的类型在运行时更改数据格式。

在运行时根据请求的类型。 所有这些系统都需要在行和列式格式之间转换数据以进行事务和分析。 由于这些转换，最新提交的数据可能无法立即用于这些类型的系统的分析查询。

2.1.2 Same Data Organization for both OLTP and OLAP

H2TAP [2]是一个学术项目，旨在构建一个HTAP系统，主要关注在异构硬件上运行时单个节点的硬件利用率。 它属于此类别，因为系统设计为行存储。

在SQL-on-Hadoop系统中，还有HTAP解决方案可以扩展现有的OLAP系统，并能够更新数据。 从版本0.13开始，Hive在ORCFile的行级[18]引入了事务支持（插入，更新和删除），这是他们的列式数据格式。 但是，主要用例是更新维度表和流数据摄取。 Impala [20]与存储管理器Kudu [21]的集成也允许SQL-on-Hadoop引擎处理更新和删除。 相同的Kudu存储也用于运行分析查询。

由于这些系统不需要从一个数据组织转换到另一个数据组织以执行事务和分析请求，因此OLAP查询可以读取最新提交的数据。 但是，它们可能面临传统关系引擎所面临的相同缺点。 他们没有最适合这两种处理类型的数据组织。 因此，由于非行式格式处理数据的开销，它们可能依赖于快速事务请求的批处理，或者由于非柱状格式而对次分析执行次优。

2.2 Separate OLTP and OLAPS ystems

此类别下的系统可以进一步区分它们处理底层存储的方式，即它们是否为OLTP和OLAP使用相同的存储。

2.2.1 Decoupling the Storage for OLTP and OLAP

许多应用程序将OLTP和OLAP系统松散地耦合在一起用于HTAP。 应用程序可以维护混合架构。 OLTP系统中的操作数据使用标准ETL过程老化到OLAP系统。 实际上，这在大数据世界中非常普遍，其中应用程序使用像Cassandra这样的快速键值存储来处理事务性工作负载，并且操作数据在HDFS上用于SQL-on-Hadoop系统上的Parquet或ORC文件。因此，OLAP系统可以查询的数据与OLTP系统看到的数据之间存在滞后。

2.2.2 Using the Same Storage for OLTP and OLAP

一些数据库供应商提供了替代的HTAP，它们将传统产品与Spark生态系统相结合，以实现大规模的HTAP。 SAP HANA Vora [16]就是这样一个例子。 在HANA Vora中，事务处理通过HANA执行，而分析请求由Spark SQL处理，子查询下推到数据库。

最近开发的几个数据管理引擎也采用了类似的方法。 例如，SnappyData [26]使用事务引擎GemFire for OLTP并利用Spark生态系统进行OLAP。

诸如HBase [17]和Cassandra [8]之类的键值存储被许多现代应用程序选为快速更新的在线操作数据存储。 为了引入缺少的OLAP功能，键值存储通常与SQL-on-Hadoop引擎一起使用。 在一种方法中，两个系统都看到存储在键值存储中的完全相同的数据。 这要求SQL-on-Hadoop系统直接查询键值存储中的数据。 已经开发了许多对现有SQL-on-Hadoop系统的扩展来实现这种集成。 使用类似于上述系统的Spark SQL中的Data Source API，Spark HBase连接器[38]和Spark Cassandra连接器[12]允许Spark SQL分别直接查询HBase和Cassandra数据。 这种方法的主要问题是性能缓慢。 运行扫描大量数据的查询（通常是分析查询）通过这些连接器非常慢。

在另一种方法中，许多SQL-on-Hadoop系统，如HIVE [36]，Impala [20]，IBM Big SQL [15]和Actian VectorH [9]，使用HBase作为可更新的存储引擎来存储需要的表 经常更新。 用户可以将操作和分析请求发送到SQL-on-Hadoop系统中的同一接口。 在下面，对HBase表的请求是通过HBase的处理引擎执行的。

像Splice Machine [33]和Phoenix [29]这样的系统在HBase之上提供了一个SQL层。 它们还允许存储在HBase表中的数据的更新和事务。 因此，他们依靠HBase进行更新。 Splice Machine甚至支持ACID事务。

直接访问HBase表的SQL-on-Hadoop系统，以及Splice Machine和Phoenix都会缓慢运行分析查询，因为对HBase表的扫描效率不高。 HBase专为快速插入和单记录查找而设计。 因此，这些系统都不能提供快速的OLAP功能。

Wild fi [5]是IBM Research最近的一个项目，该项目构建了一个HTAP引擎，其中分析和事务请求都通过同一个柱状数据组织，即Parquet [1]。 通过使用单一数据组织进行数据提取和分析，Wild fi re可以立即对最新提交的数据进行分析。 Wild fi还利用Spark生态系统来实现大规模分布式分析。 对Wild的请求通过Spark SQL进入，并尽可能地下推到Wild fi引擎。

由于OLTP和OLAP组件为此类别中的系统共享相同的底层存储，因此最新提交的事务可立即查询以进行分析。

3. ROAD TO TRUE HTAP

在结束我们的教程之前，我们计划强调HTAP系统构建者以及HTAP用户今天仍然面临的挑战。

即使有许多系统标记为HTAP解决方案（如第2节所述），但它们都不支持真正的HTAP。 现有的解决方案确实提供了一个合适的平台，用于在分别发送到系统时支持事务和分析请求。 但是，现有的解决方案都不支持在同一事务中高效处理事务和分析请求。 为了完全支持HTAP，系统不仅应该在提交或更新数据的事务已提交之后，而且还应作为同一请求的一部分，对最近的数据进行分析。

此外，如今大多数HTAP解决方案都使用多个组件来提供所有所需的功能。 这些不同的组件通常由不同的人群维护。 因此，保持这些组件兼容并为最终用户提供单个系统的错觉是一项具有挑战性的任务。

最后，索引分发的数据并进行大规模访问以实现高效的点查找并不是直截了当的。 此外，大多数这些系统都部署在公共云或私有云上，这些云使用对象存储以及HDFS等共享文件系统。 需要细粒度索引才能实现高效的点查找和更丰富的OLTP。 快速OLTP引擎将索引保留在内存中，但是对于大规模数据和HTAP，这些索引不能仅保留在内存中。 可以为最频繁访问的数据缓存索引的一部分，并降低索引条目的访问成本。 但是，大规模分布式OLAP系统使用主要针对扫描进行优化的共享文件系统和数据组织，这些组织不能快速访问单个记录或列。 对这些共享文件系统和对象存储的更快点访问仍然是一个悬而未决的问题。