分布式数据库面临大数据挑战时的演进

——以MPP体系和Hadoop框架为例

dz1933026 王国畅

一、背景

分布式数据处理的一个主要场景是对海量数据的存储和管理。随着互联网的发展，当今社会的企业和互联网用户持续产生大量的数据，截至2012年，单一数据集的大小从数太字节(TB)至数十兆亿字节(PB)不等，而截至2020，已有Amazon Redshift Spectrum等服务致力于将可处理的数据量扩展至艾字节(EB)。

非分布式的数据库软件难以支撑如此大规模数据的存储、管理和分析。因此，致力于对大规模数据进行获取、存储、管理、分析的大数据架构逐渐出现。其中两类典型的分布式数据处理方案是MPP体系的数据库方案和Hadoop体系的一系列大数据处理方案。两种框架的分布式数据处理系统都在数据处理的实时性和高吞吐挑战下呈现出旺盛的生命力。

二、MPP架构的演化

MPP(Massively Parallel Processing)，即大规模并行处理，其核心原则是Shared Nothing。在数据库非共享集群中，每个节点都有独立的磁盘存储系统和内存系统，业务根据不同节点的特点进行子业务划分和调度。一个典型的MPP架构数据库图示如下：

图片包含 白色, 游戏机

描述已自动生成

传统数据库架构通常为SMP(Symmetric Multi-Processing)，即对称多处理架构。在SMP架构下，服务器中多个 CPU 对称工作，无主次或从属关系。各 CPU 共享相同的物理内存，每个 CPU 访问内存中的任何地址所需时间是相同的，因此SMP也被称为一致存储器访问结构(UMA：Uniform Memory Access)。对 SMP 服务器进行扩展的方式包括增加内存、使用更快的 CPU 、增加 CPU 、扩充 I/O(槽口数与总线数) 以及添加更多的外部设备 (通常是磁盘存储) 。

由于CPU，I/O，内存都是共享的，SMP架构的扩展能力十分有限，随着CPU数量增加，共享资源的访问冲突会日益严重，例如不同CPU对内存的访问冲突。这些冲突会导致内存等资源变成性能瓶颈。

MPP架构下的分布式数据库解决方案对比SMP架构有如下优势：

* 并发性：通过高度的并行处理，MPP架构下的数据库的查询完成速度通常比SMP架构下的数据库大幅加快，同时由于可以并发读取和写入，系统的吞吐量得到大幅增长。
* 可扩展性：基于MPP架构的数据库可以增量的增加计算和存储资源，从而增长数据库的存储、处理能力，因此能够支撑一定数据级别的大数据架构。

MPP架构的典型数据仓库方案如Greenplum，Asterdata，Nettezza等。

尽管MPP架构的出现给予了数据库进行分布式并发处理的手段，也足以实现批量数据处理场景下较高的吞吐和较低的时延。但随着数据处理的实时性要求不断增强（如天气预报等），传统数据库技术难以支撑其需求。因此MPP架构的分布式数据处理需要面向OLTP进行了全面的技术优化。以Greenplum为例，其版本更新中对OLTP场景进行如下优化：

* 全局死锁检测
* 锁优化
* 事务优化
* 复制表
* 多模存储
* 灵活索引
* OLTP友好的优化器
* 内核升级：PostgreSQL 9.4

经过面向OLTP的优化， MPP架构下的数据处理延迟大幅降低，足以达到PB级别数据即时处理的性能需求。在Greenplum6.2.1 JDBC和MySQL 5.6的对比试验中，两者呈现了近似的insert性能：

手机屏幕截图

描述已自动生成

三、Hadoop体系的演化

Hadoop体系中最初的三个关键数据处理组件是HDFS，MapReduce和Hive。

HDFS是一个高容错性的分布式文件系统，可以被广泛部署在廉价的PC上。它以流式访问模式访问应用程序的数据，从而大大提高了整个系统的吞吐量。HDFS架构如图所示：

图片包含 游戏机

描述已自动生成

HDFS采用主从架构。一个典型的HDFS集群包含一个NameNode节点和多个DataNode节点。NameNode节点负责整个HDFS文件系统中的元数据保管和管理，主要负责管理文件系统的命名空间，集群配置，存储块复制等；DataNode节点通过心跳机制与NameNode节点进行定时通信，负责保存文件块在本地，并周期性的向NameNode发送报告。

除了HDFS的分布式文件系统支撑，Hadoop还通过MapReduce算法实现分布式数据计算机制，Map(映射)，Reduce(化简)，采用分而治之的思想，先把任务分发到集群多个节点上，并行计算再把计算结果合并，从而得到最终计算结果。其中所涉及的任务调度，负载均衡，容错处理等都由MapReduce框架提供。

在上述基础上，Hadoop通过Hive组件支撑SQL语句进行分布式数据查询。HDFS、MapReduce加上Hive的分布式数据处理方案一度成为潮流，它们的组合可以轻易处理PB级别以上的数据存储和管理问题并达到相对高效的分布式计算和检索性能。

但与MPP架构相同，随着大数据场景的复杂化，实时性需求推动着Hadoop继续演化。

在实时大数据处理场景下，分布式数据处理任务通常包括对系统的高频率的少量数据插入和对这些数据的实时处理。这时，HDFS、MapReduce加上Hive的方案则呈现延迟高，响应缓慢的特点。

其关键瓶颈在于MapReduce。Hadoop中MapReduce可处理的情形仅限于待处理的数据可以分解成若干小数据集或每一个小数据集可以完全并行的进行处理。

为了解决MapReduce在实时数据处理方面的短板，Spark和Storm应运而生。

Spark是一个围绕速度，易用性和复杂分析构建的处理框架。Spark本质是一个分布式计算模块，用于解决MapReduce无法处理的实时性要求高的流处理任务。其核心概念为RDD(Resilient Distributed Dataset) ，即分布式数据集。它代表一个不可变，可分区，内部元素可计算的集合。RDD具有数据流模型的特点：自动容错、位置感知性调度和可伸缩性。RDD允许用户在执行多个查询时显式地将工作集缓存在内存中，后续的查询可以重用工作集，从而极大的提升查询速度。

RDD可以看做是对各种数据计算模型的统一抽象，Spark的计算过程主要是RDD的迭代计算过程。RDD的迭代计算过程非常类似于管道。分区数量取决于partition数量的设定，每个分区的数据只会在一个Task中计算。所有分区可以在多个机器节点的Executor上并行执行。示意图如下：

图片包含 游戏机, 截图

描述已自动生成

Spark的RDD设计使得其数据处理逻辑得到了简化，且通过在内存中运算大大降低了时延，同时由于RDD本身大小具有弹性，可以对不同粒度的batch数据进行分布式处理。这样的设计使得Spark小规模高频率的流式处理任务上性能远超MapReduce，从而令HDFS、Spark加SparkSQL的分布式数据库方案足以支持延时在秒级的流式处理任务。

Storm则是更加大胆的尝试。对比Spark，在批处理和流处理方面，Spark的核心思想是通过RDD使得批处理的批(batch)可以伸缩，从而既可以应对庞大数据的批处理，也可以通过缩小batch应对流式处理，在运行时，Spark总是将数据收集，直到成为一个RDD再对RDD进行处理。而Storm则着眼处理的实时性，对数据队列中的每条数据逐条处理。数据不再有批的概念，而是呈流状被处理：

图片包含 游戏机, 项链

描述已自动生成

Storm完全实时处理的设计使得Storm能够达到低于1秒的延迟，因此以Storm为流处理链路的数据仓库足以支撑比Spark方案实时性要求更高的场景。

四、MPP体系与Hadoop体系的对比

MPP体系与Hadoop体系都足以支撑大数据场景，也都可以应对不同的需求场景实现高吞吐和低延时特性。但两者在可处理的数据量上有显著区别。

MPP架构可以提供多种大数据架构下PB级别的分布式数据库解决方案，但MPP架构作为分布式数据处理系统仍然有其局限，即其集群扩展能力有限：多种异构数据库的任务细分和调度在过大的数据量和过多节点的场景下会成为集群的性能瓶颈。

Hadoop体系通过高容错性的分布式文件系统HDFS实现了比MPP体系更高的可扩展性。对比MPP的Shared nothing原则，Hadoop的核心文件系统HDFS使得Hadoop技术体系得以处理更大的数据量。

在性能优化方面，MPP体系更依赖于传统数据库技术的纵向性能优化，而Hadoop体系则依赖于横向的不同分布式数据处理组件的加入，在效果上，MPP体系往往能在同级别的数据量上取得领先，由于技术成熟，MPP体系的分布式数据库系统在PB及以下级别往往也比Hadoop体系更受欢迎。

五、总结

从结果上看，MPP体系虽然限于仅能处理PB级别的数据，却在PB级别的数据处理方面比Hadoop体系更受欢迎。一方面，MPP架构的数据仓库方案底层基于Shared Nothing，可以看作传统数据库方案的一种水平延伸，另一方面，MPP架构持续对传统数据库概念中的OLTP和OLAP场景进行性能优化，而不是通过添加新的处理引擎来适应新的场景，这使得MPP架构整体上学习成本更低，更易维护。

而Hadoop体系的技术则由于开源社区的高热度持续发展出新的技术方案来应对新的挑战，彰显出了蓬勃的生命力，且几乎占领了PB以上级别的大数据处理高地，在数据量更大的情况下，基于Hadoop体系的技术（Spark，Storm，Flink均属于分布式处理引擎且均通常与HDFS等技术协作使用）几乎是开源方案中构建分布式数据库的最佳选择。

而无论是MPP体系还是Hadoop体系，可以预见的是在新的分布式数据处理挑战来临之时，核心技术的创新才是进一步演进的关键。