

# 大规模异构数据并行处理系统的设计、实现与实践

夏正勋, 罗圣美, 孙元浩, 唐剑飞, 张燕  
星环信息科技(上海)有限公司, 上海 200233

## 摘要

随着互联网和物联网应用的快速发展, 数据处理模式从结构化逐渐扩展到结构化、半结构化和非结构化混合的异构数据处理模式。设计了一种大规模异构数据并行处理系统, 在统一的平台功能视图基础上, 采用统一的资源管理框架, 实现对结构化、JSON/XML、图数据、文档数据等多种异构数据进行存储和查询, 采用统一的开发语言, 实现跨数据类型和数据存储引擎的并行计算, 满足多业务应用开发的需要, 并通过实际的商业部署, 验证了系统的可行性。

## 关键词

并行系统; 异构数据; 统一架构

中图分类号: TP311.13

文献标识码: A

doi: 10.11959/j.issn.2096-271.2020030

## *Design, implementation and practice of parallel processing system for a large-scale heterogeneous data*

XIA Zhengxun, LUO Shengmei, SUN Yuanhao, TANG Jianfei, ZHANG Yan

Transwarp Information Technology (Shanghai) Co., Ltd., Shanghai 200233, China

## *Abstract*

With the rapid development of Internet and IoT applications, data processing has gradually expanded from structured to structured, semi-structured and unstructured hybrid heterogeneous data processing mode. A large-scale heterogeneous data parallel processing system was designed. Based on the functional view of a unified platform, the unified resource management framework was adopted to store and query a variety of heterogeneous data, including structured, JSON/XML, graph data, document data, etc. Adopting a unified database language, the parallel computing across data types and database engines was realized, and the needs of multi business application development were met. The feasibility of the system is verified by standard evaluation environment and commercial deployment.

## *Key words*

parallel system, heterogeneous data, unified architecture

## 1 引言

随着互联网和物联网应用的快速发展,数据呈现爆发式增长趋势,一方面数据规模的膨胀导致传统的关系型数据库无法满足存储和查询的实时性要求,另一方面数据结构出现多样化,各种不同类型的数据库并行交互,数据流转和融合处理难度加大,成本上升。

为了满足企业数字化转型的需求,政府和大中型企业一方面需要在确保安全和隐私的前提下开放自身的业务数据,另一方面需要接入大量的互联网、物联网数据,提升数据的代表性和泛化能力,为此需要为数据共享和分析规划统一的大规模异构数据处理平台,实现资源的统一调度,兼容数据格式,实现存储和计算并行,统一用户开发接口,满足数据流转和融合处理需求,提高业务的实时性和安全性,快速实现数字化转型目标。

## 2 当前研究方向与现状

自1970年Edgar F. Codd发布划时代的论文《用于大型共享数据库的关系数据模型》以来,经过近50年的研究实践,产业界已能够通过关系型数据库很好地管理结构化数据。但随着互联网和物联网应用的快速发展,数据处理的种类及规模发生了翻天覆地的变化,图数据、文档数据、JSON/XML等非结构化数据的处理需求快速增长,企业对数据处理的需求从结构化数据处理逐渐扩展到结构化、半结构化和非结构化混合的数据处理。

近10年来出现了一种针对NoSQL数据的处理技术,其本质上是为解决某一特定

类型的异构数据问题而设计的数据处理及存储系统<sup>[1]</sup>。NoSQL数据处理系统主要分为三大类:文档数据处理系统、图数据处理系统和BigTable数据处理系统。国外开源技术在NoSQL领域比较活跃<sup>[2]</sup>,包括以MongoDB和Elastic为代表的文档数据处理系统,以Neo4J和JanusGraph为代表的图数据处理系统,以HBase和Cassandra为代表的BigTable数据处理系统。国内在结构化数据处理技术研发方面起步较晚,在2000年左右出现了武汉达梦数据库有限公司、北京人大金仓信息技术股份公司、天津南大通用数据技术有限公司等数据库代表厂商<sup>[3]</sup>;而在NoSQL数据处理技术方面,国内和国外几乎同一时间开始相关技术的研发。国内厂商在应用创新上表现较为突出,如阿里云结合大数据和云计算技术,提供一站式数据服务的非结构化数据管理平台,该平台可以提供高效的数据存储、数据挖掘、全文检索、视频转码、图像识别等服务;星环信息科技(上海)有限公司致力于基础软件平台的研发,实现了关系型数据、KV数据、图数据、文档数据等多种类型数据的存储、分析和挖掘。

与此同时,全球数据规模急剧增长,国际数据公司(international data corporation, IDC)预测2025年全球数据总量将达到175 ZB,预计中国的数据量在未来7年将平均每年增长30%以上<sup>[4]</sup>。数据处理的规模由TB级迅速提升至PB级,数据处理系统的架构也逐渐从纵向扩展(scale up)转向横向扩展(scale out),通过增加节点数实现处理能力的扩展。典型的数据处理系统架构包括对称多处理(symmetric multiprocessing, SMP)架构、大规模并行处理(massively parallel processing, MPP)架构以及MPP+Hadoop混合架构。SMP架构的典型代表有Oracle、DB2等, MPP架构的典型

代表有思爱普公司的HANA、GreenPlum等，MPP+Hadoop混合架构的典型代表有Databricks公司的Delta Lake、亚马逊公司的Lake Formation等。上述架构各具特色，在不同时期解决了企业面临的不同规模的数据处理需求。

### 3 当前异构并行系统的问题与解决思路

2005年，Michael Stonebraker阐述了在数据处理领域不存在“one size fits all”的解决方案<sup>[5]</sup>，即不存在一种数据处理系统，能够适应所有的数据处理应用场景，数据处理领域的主流框架一直在不断地改进。基于SMP架构的关系型数据库起源于1970年<sup>[6]</sup>，至今仍在数据处理领域发挥着重要作用，但由于其原设计目标是处理结构化数据，所以对非结构化数据的处理能力较弱，无法单独解决当今日益复杂的异构数据分析问题。另外，基于SMP架构的数据处理系统性能提升一般采用纵向扩展的方式，严重依赖硬件的升级，性价比较低。

MPP架构的核心设计思想是无共享（share nothing），最早在Gamma<sup>[7]</sup>和Teradata中得到应用。MPP架构采用横向扩展的方式提升系统整体性能，解决了SMP架构性能扩展受限的问题。具体来说，MPP系统是由许多松耦合的处理实例组成的分布式并行数据处理系统，各实例之间不共享资源，每个实例负责存储和计算全局数据的一部分，但MPP系统依然不擅长处理异构数据，且在协作计算时，节点间需要进行全连接通信，容易产生网络风暴，这限制了MPP集群的规模。

Hadoop系统在2003年出现，现在已建立了庞大的生态系统，包含可以并行处理异构数据的MapReduce<sup>[8]</sup>、Impala<sup>[9]</sup>、

Spark<sup>[10]</sup>、Flink<sup>[11]</sup>等一系列开源计算框架。近几年来业界出现了将MPP系统与Hadoop系统相结合的“MPP+Hadoop”混合数据处理架构，此类数据处理架构既可以处理结构化数据，又可以处理半结构化数据、非结构化数据，数据需要存储在多个子系统中，造成资本性支出（capital expenditure, CAPEX）、运营成本（operating expense, OPEX）增加，同时数据的时效性及一致性难以保证。

图1简要总结了3种数据处理架构存在的问题。

本文基于结构分层、功能融合的设计思想，结合产业应用需求，提出了一种大规模异构数据并行处理系统，在架构上将系统分为统一的开发接口层、统一的数据计算引擎层、统一的分布式存储管理层、统一的资源调度管理层，该系统支持多种不同的SQL和NoSQL数据处理引擎，支持结构化数据、图数据、文档数据、大表、JSON等类型的数据<sup>[12]</sup>的存储、检索和分析，并能够通过统一的开发接口提供数据分析服务，为开发者提供统一的数据处理体验。

### 4 系统实现

本文提出的大规模异构数据并行处理系统架构如图2所示。

最上层是统一的开发接口层，其采用标准的SQL开发接口和Cypher语言接口，经过统一的语法解析，形成标准的计算原语，再通过统一的优化处理，生成高效的执行计划。这种统一开发接口层的设计避免了API开发方式存在的应用兼容性差、开发难度高等问题，提升了数据应用开发的效率。

统一的数据计算引擎层可以支持在

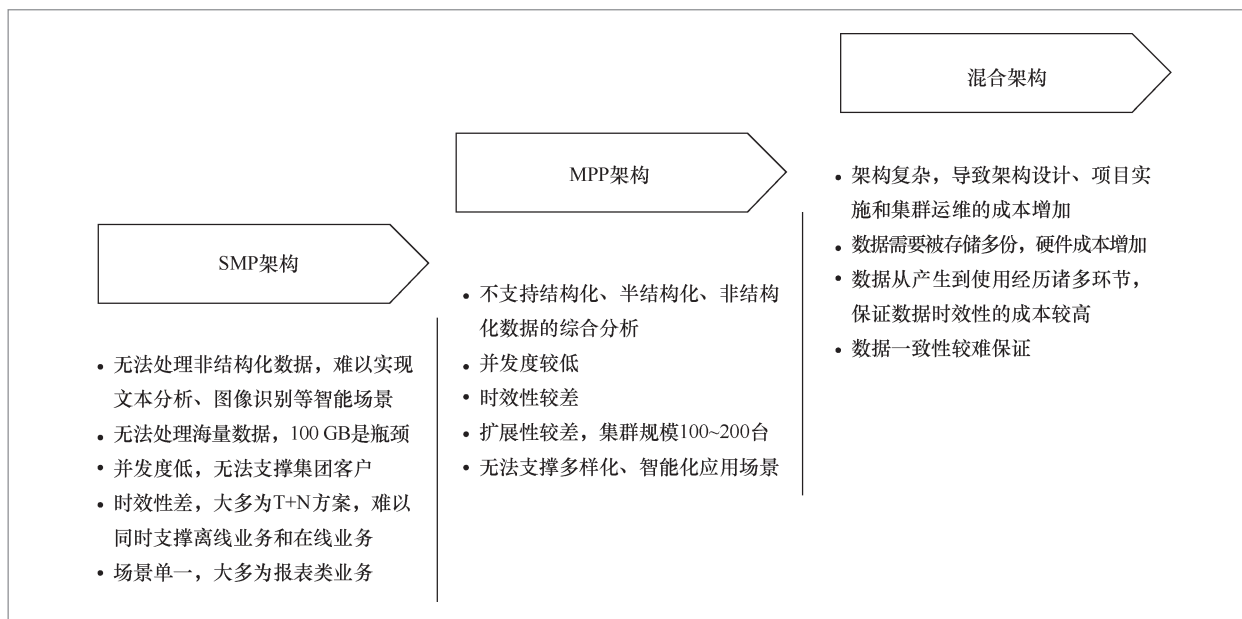


图1 3种数据处理架构存在的问题

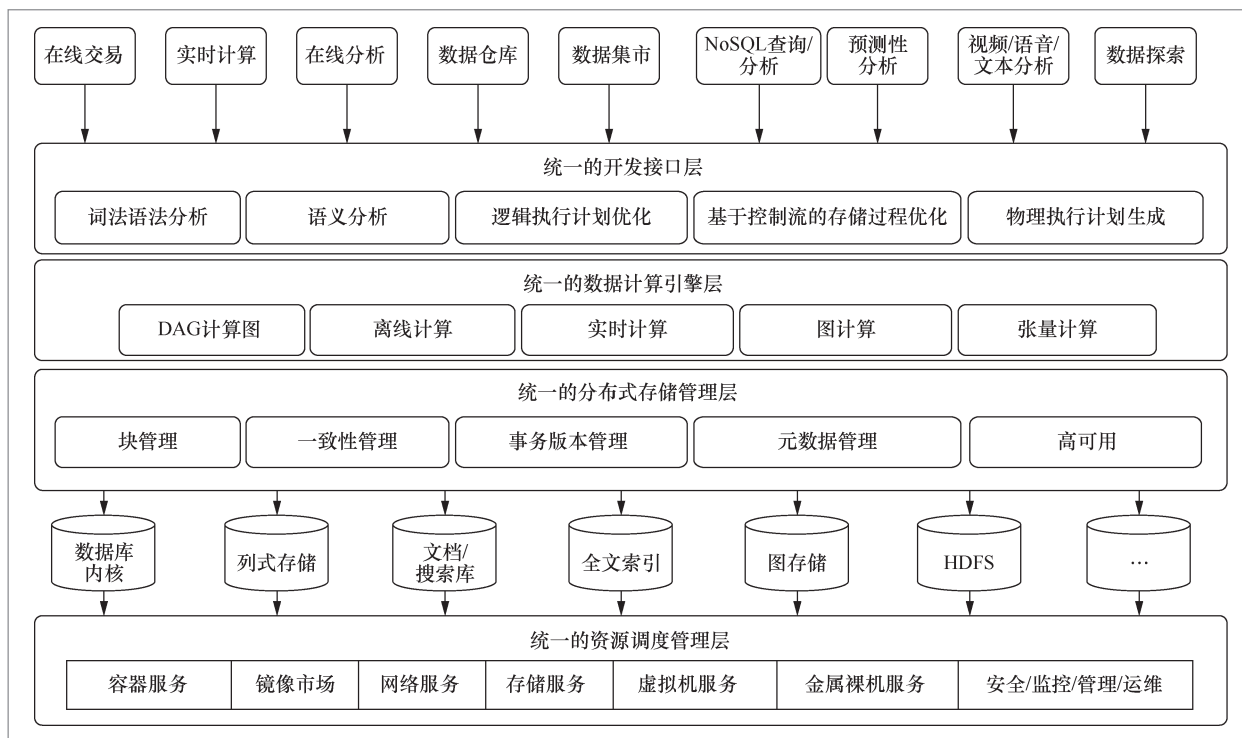


图2 大规模异构数据并行处理系统架构

线或离线计算，采用有向无环图 (directed acyclic graph, DAG) 计算模式支持各种数据处理，负责数据的计算、变换、关联、聚合

等操作，能够将标准的SQL操作转换为不同能力引擎的数据访问操作。相对于MPP架构，DAG计算模式更适合大规模集群计算

场景,基于DAG模式的计算有很好的扩展性,能够满足包括图计算、深度学习在内的多迭代的计算特性,并能够避免集群间的网络风暴,在实践中已经被广泛应用。

统一的分布式存储管理层支持不同的数据存储与分析需求,对不同类型的数据存储引擎的存储、检索、分析的能力进行抽象,实现对存储引擎能力的统一管理,这些引擎包括用于分析型数据库的Hadoop分布式文件系统(Hadoop distributed file system, HDFS)、用于BigTable存储的HBase、用于全文索引的Lucene、用于图计算的图存储引擎等。

系统底层是基于Kubernetes的统一资源调度管理层,负责管理和调度各种计算任务。通过对Kubernetes和数据处理系统底层的融合创新,统一的资源调度管理层可以支撑各种计算任务,包括MapReduce、Spark、深度学习,甚至是消息传递接口(message passing interface, MPI)类的高性能计算任务。

下面对各层涉及的技术进行详细说明。

#### 4.1 统一的开发接口层

统一的开发接口层的核心是语法编译器和执行优化器,语法编译器可以对SQL 2003、OraclePL/SQL、MySQL方言、Cypher图计算语言等声明式开发语言进行词法语法解析、语义分析、执行计划生成、基于控制流的存储过程优化等处理,并将其转换成标准的计算原语;执行优化器可以对逻辑执行计划和物理执行计划进行统一的优化处理,提高执行性能。

面向异构数据并行处理的统一编译及优化过程如图3所示。

##### (1) 词法语法分析模块

词法分析是编译的第一个阶段,负责将SQL/Cypher/PL语句转换为单词

(token)序列,词法分析器一般以函数的形式存在,供语法分析器调用。语法分析器借助元数据信息和多存储抽象信息对SQL/Cypher/PL语句进行语法检查,并构建由输入的单词组成的抽象语法树。

##### (2) 语义分析模块

语义分析是编译过程中的一个逻辑阶段,审查源程序有无语义错误,为代码生成阶段收集类型信息。语义分析将业务逻辑转化为若干可以在分布式系统中执行的执行块,基于执行块的依赖关系,创建栈式函数执行框架。

##### (3) 逻辑执行计划优化模块

逻辑执行计划优化是优化器利用语法结构优化语法树结构及执行顺序的过程。本文设计的优化器可以支持上百种优化规则,如公共子表达式消除、列裁剪、分区裁剪、谓词下推等,减少了数据的计算量,使得逻辑执行计划的整体效率更高。

##### (4) 基于控制流的存储过程优化模块

与传统的SQL优化原理不同,基于控制流的存储过程优化针对存储过程具有复杂函数调用逻辑的特点,采用计算级别的优化方法,包括死代码删除、函数内敛、冗余删除、循环不变量外提、参数传递优化、循环优化、并行化等,计算级别的优化可以进一步减少存储及计算的开销,提升查询效率。

##### (5) 物理执行计划生成模块

物理执行计划是基于逻辑执行计划生成的可以在分布式物理环境下运行的执行计划。为了实现最佳的分布式计算整体效率,物理计划生成通常既要考虑数据特性,也要考虑计算特性,如依据数据的物理分布特点,按就近原则生成数据源读取执行计划,避免跨节点读取数据;同时,在计算过程中,对中间计算结果进行合理的层次聚合,降低集群间的通信数据总量。

在逻辑执行计划优化和物理执行计

划生成优化方面,本文还针对图、键值、文档、关系型数据等异构数据的特点执行多层次的优化。针对简单查询,可以采用条件过滤、全表扫描规避、查询结果快速索引等优化处理;针对多数据源关联分析,可以根据数据规模采用普通Join、MapJoin、空表Join等优化处理;针对图数据处理,可以通过语法语义分析,首先利用资源描述框架(resource description framework, RDF)的模式信息对SPARQL图模式进行简化处理,在此基础上利用连接代价和动态规划方法寻找每一次迭代的最优连接顺序,最终得到最优查询计划。上述具有针对性的优化可以保证异构数据处理逻辑在不需要手工改动的情况下发挥最大的性能。

### 4.2 统一的数据计算引擎层

目前业界主流的数据并行计算架构有两种,一种是MPP架构,另一种是DAG架构。从整体上来看,基于MPP的计算引擎采用多实例来加速分析,实例之间不共享数据,在小数据量情况下数据分析性能较好。DAG天生适合流水线(pipeline)计算模式,在数据处理领域,可以通过DAG完整地描述数据处理流水线的全过程,DAG解析后可生成若干并行的子任务,能够充分发挥大规模数据处理集群的计算性能。

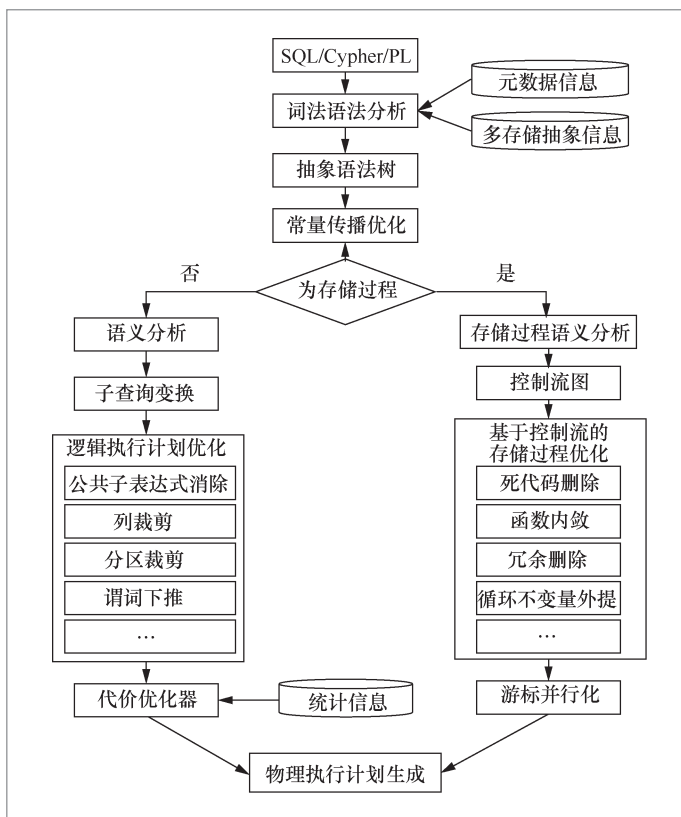


图3 面向异构数据并行处理的统一编译及优化过程

本文设计的基于DAG的统一的数据计算引擎层可以有效地减少通信的性能损耗,适合大规模集群之间的各种数据计算任务。DAG与MPP架构的具体比较见表1。

在DAG的计算模式基础上,统一的数据计算引擎层支持同时调用批处理算子和

表1 MPP与DAG架构对比

技术点	基于MPP的计算引擎	基于DAG的统的数据计算引擎
计算优化	单机执行速度快,侧重节点域存储与计算优化,全局计算任务优化困难	有全局的元数据信息,可以更好地协调执行器之间的数据通信、任务启停,支持集群域全局优化
容错性	依赖各个节点完成任务切分	任务设计简单、有幂等性
数据通信性能	依赖数据分布减少数据通信的性能损耗,不灵活	依赖全局的数据元信息减少通信的性能损耗,更加灵活
核心优势	优化器成熟,本地执行性能更好	灵活性、容错性更高,能够更好地减少数据通信消耗
劣势	总体性能依赖业务特性和数据分布	开发接口标准语法支持程度、优化器等仍需持续提升和改进

流处理算子,使用事件驱动和批处理统一的模型,通过轮询数据变化来驱动实时计算,使一个计算引擎既能支持实时计算又能支持离线计算,实现流批一体,满足更多样化的数据计算需求。

具体来说,统一的数据计算引擎层将批处理算子或流处理算子按编程接口形成DAG计算图,再根据DAG计算图中子任务的依赖关系将计算划分成任务集,并结合流水线并行规划,将不同的任务集提交到多个计算节点,进行灵活的调度。

另外,本文还使用向量化计算技术对批量数据处理做进一步的优化,向量化计算可以对批量数据进行一次性计算,而不需要逐个计算数据,可以极大地提升计算效率。

### 4.3 统一的分布式存储管理层

随着大规模异构数据并行计算应用的

普及,越来越多的专用存储引擎被整合到数据处理系统中,协同完成计算任务。因此,大规模异构数据并行系统需要解决各存储引擎的兼容问题,保证各存储引擎数据的一致性,实现与事务操作相关的接口。

本文使用Raft协议来保证各分布式存储管理层的数据一致性,并在此基础上完成与事务操作相关的接口,当客户端需要进行数据操作时,发送指令到统一的分布式存储管理层中,由统一的分布式存储管理层查询元数据和路由表,找到数据所在的Raft分组并进行交互,完成对数据的具体操作,统一的分布式存储管理层逻辑示意图如图4所示。

分布式存储管理层基于Raft协议可以实现以下功能:

- 各单机存储副本之间的状态同步,从而实现各数据节点的数据一致性;
- 可信的选主过程,确保相关状态的

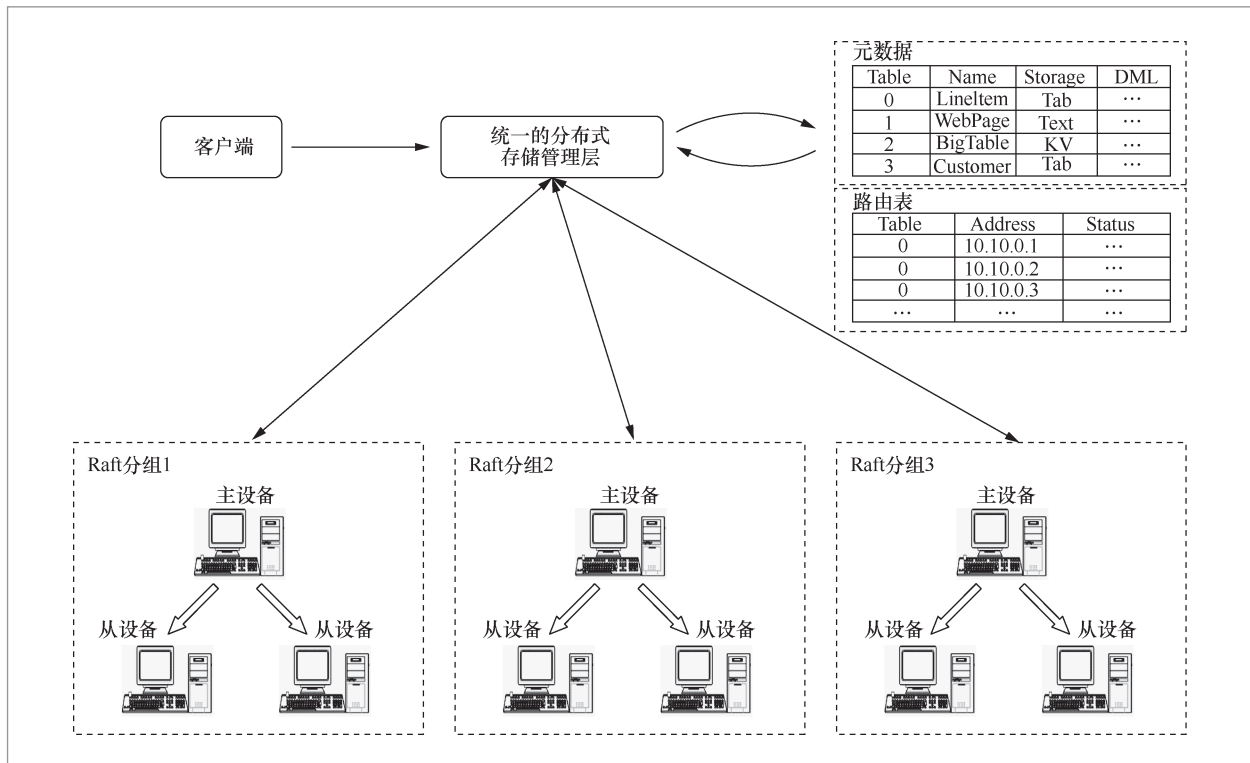


图4 统一的分布式存储管理层逻辑示意图

同步;

- 事务协同组的选主和相关状态同步, 以实现与事务操作相关的接口;
- 具有存储服务的恢复服务能力, 以实现容错与灾备。

统一的分布式存储管理层将分布式存储的通用功能接口抽象出来, 如一致性接口、事务操作接口、多版本并发控制 (multi-version concurrency control, MVCC) 接口、分区策略接口等功能接口, 如图5所示。存储引擎只需要实现其单机版本, 在此基础上实现特定驱动, 与本层进行接口适配就可以实现分布式处理的功能。

在具体实现过程中, 可以通过以下步骤将单机存储系统接入统一的分布式存储管理层。

- 在各个单机服务中定义状态机, 用来进行各服务之间的协同。
- 在状态机中实现实际处理逻辑, 完成特定状态下的具体操作。例如, 可以实现OnExecute方法, 完成具体的执行逻辑, 也可以实现OnFailed方法, 完成执行失败之后的处理逻辑。

- 定义分布式服务中负责增删节点和数据重分布的状态机。

添加故障恢复的处理方法, 以完成容错与灾备功能。在单机存储系统中向外提供与事务相关的接口。

#### 4.4 统一的资源调度管理层

大规模异构数据并行系统需要管理和调度各种不同类型的计算任务, 如MapReduce、Spark、GraphX、深度学习等, 目前尚没有一个开源的资源管理框架可以支持所有计算任务。一个可行的方案是结合Kubernetes技术, 通过对大数据底层技术的创新, 实现云原生的、统一的资源调度管理层, 支撑各种计算任务。

统一的资源调度管理层是大规模异构数据并行系统能够有效运行的重要保证, 包括资源管理模块、服务模块和云调度模块, 如图6所示。

服务模块包括以下5个部分。

- 配置中心: 实时地收集和管理云平台内运行的服务的配置参数。
- 物理资源池: 放置各种资源池化后

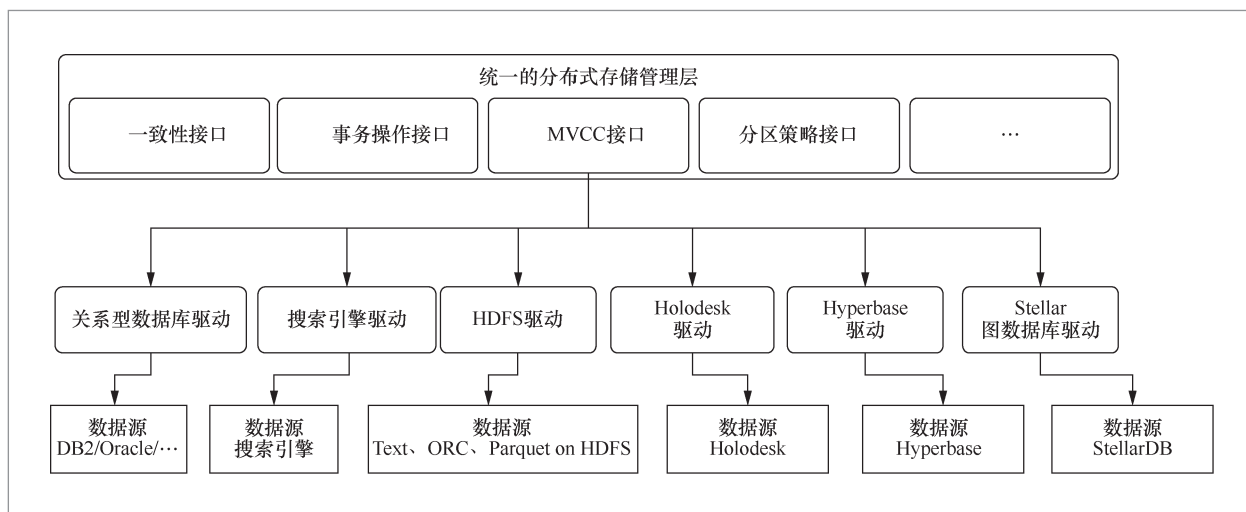


图5 统一的分布式存储管理层结构

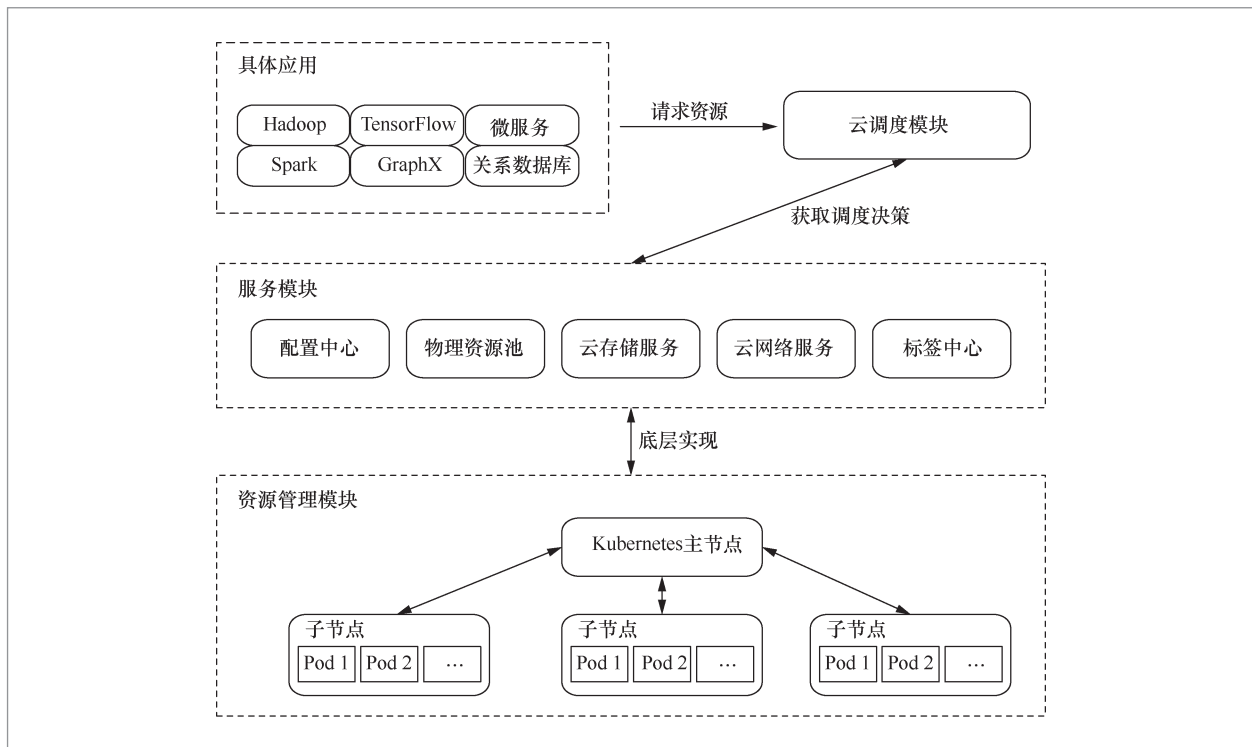


图6 统一的资源调度管理层

的逻辑资源。

- 云存储服务：基于本地存储开发的分布式存储服务，对有状态服务的数据进行持久化。

- 云网络服务：给租户提供类似于虚拟私有云的网络服务。

- 标签中心：对容器、资源的标签进行管理。

云调度模块负责与具体应用交互，接收应用的输入，然后从服务模块获取实时的状态数据，形成调度决策，分配资源供具体应用使用。

资源管理模块采用Kubernetes提供基础的技术支撑。

#### 4.5 总结

2019年咨询公司Gartner在《2019年数据管理技术成熟度曲线》分析报告中预

测“在一个系统中处理多种异构数据是未来数据处理的必然趋势”<sup>[12]</sup>，本文针对大规模异构数据处理设计了一种大规模异构数据并行处理系统，并完成了产品研制和应用部署。与传统数据处理系统相比，基于统一架构的数据处理系统有以下特点。

- 基于编译器及优化器的统一的开发接口层统一了异构数据处理的开发接口，避免了数据处理开发者同时使用多种开发语言，降低了数据应用开发门槛和开发者的学习成本，提升了数据业务开发效率及客户需求响应效率。

- 相对于传统数据处理集群，基于DAG模式重新设计的统一的数据计算引擎层的计算优化策略的范围从节点域扩展到集群域，实现了全局的计算调度及通信路由优化，减少了传统数据处理集群的通信开销，提升了数据业务的计算效率。

- 基于存储驱动设计的统一的分布式

存储管理层将数据存储的统一处理接口抽象出来,能够兼容支持各类数据的专有存储引擎,简化了异构数据的存储处理流程,提升了数据业务的存储效率。

## 5 应用案例

以某世界500强集团公司数据处理系统升级为例,原来基于SMP架构和MPP架构构建的数据处理平台存在以下问题。

- 数据规模逐年增大,受架构限制,数据处理平台规模难扩展。
- 非结构数据处理能力不足。
- 不同部门存在数据壁垒,数据共享消耗大量资源,难保证数据的一致性。

基于本文的设计思想,实现了基于统一架构的大规模异构数据处理平台解决方

案,如图7所示。

在基础设施层面,利用资源调度管理模块对计算资源、存储资源、网络资源进行集中调度管理,提升基础资源的利用效率;在平台及服务层面,采用统一的计算引擎和存储管理,支持多种类型数据的存储及分析,并提供批量处理、流式计算、图计算、BI/AI分析等多种数据分析处理能力,从而突破了原系统架构对计算规模及数据处理能力的限制。基于统一的开发接口层为应用层提供了开发测试、数据共享、多租户隔离等丰富的能力开发接口。

本文提出的处理平台已被应用于实际系统中,截至2020年3月,使用该平台进行系统组网的规模达到了近300台设备,分为数据湖、数仓、分析实验室、省分、开发测试五大集群,平台CPU核心近5 000个,内存近100 TB,可存储空间超过3 PB,有

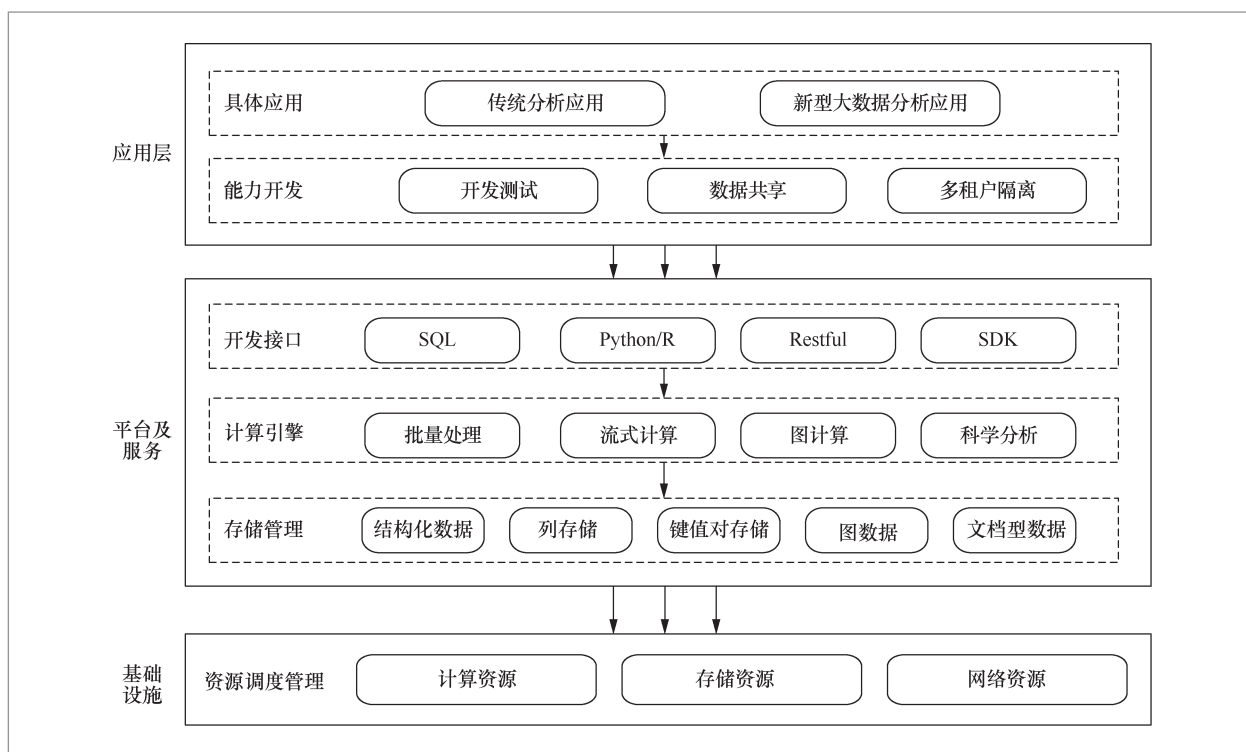


图7 某基于统一架构的大规模异构数据处理平台

效支撑了各部门的数据处理和分析工作。在系统部署过程中,业务需求经过了3次迭代才逐步厘清。第一次迭代:圈定范围,搭建基础系统平台;第二次迭代:细化平台内各项主要应用,逐步完善系统异构数据的处理能力;第三次迭代:提升数据分析能力,强化数据管理流程。3次迭代充分验证了大规模异构数据并行处理系统的可行性,也通过实践进一步完善了数据处理细节,使系统能够更好地适应多样化的产业应用场景。

## 6 结束语

目前,大数据已经进入了3.0时代,大数据3.0时代要求在同一平台中可以满足不同层次的大数据需求<sup>[13]</sup>。然而,分别以Hadoop和Spark为代表的大数据1.0与大数据2.0技术在数据存储、计算和数据业务打通方面存在很多缺陷<sup>[14]</sup>。为了解决这些问题,本文提出了一种大规模异构数据并行处理系统,经过理论分析和商业应用实践,验证了系统的可行性,为异构数据并行化处理技术和应用提供了参考,有助于推动大数据、人工智能、云计算应用一体化的实现。

## 参考文献:

- [1] DAVOUDIAN A, LIU C, LIU M C. A survey on NoSQL stores[J]. ACM Computing Surveys, 2018, 51(2): 1-43.
- [2] SHARMA S, SHANDILYA R, PATNAIK S, et al. Leading NoSQL models for handling big data: a brief review[J]. International Journal of Business Information Systems, 2016, 22(1): 1-25.
- [3] 陈民峰. 国产数据库管理系统应用的研究与探讨[J]. 计算机应用, 2003, 23(z2): 39-42.  
CHEN M F. Research and discussion on the application of domestic database management system[J]. Journal of Computer Applications, 2003, 23(z2): 39-42.
- [4] IDC. Data age 2025, the digitization of the world from edge to core[R]. 2018.
- [5] STONEBRAKER M, CETINTEMEL U. One size fits all: an idea whose time has come and gone[C]//The 21st International Conference on Data Engineering. Piscataway: IEEE Press, 2005.
- [6] CODD E F. A relational model of data for large shared data banks[J]. Communications of the ACM, 1970, 13(6): 377-387.
- [7] DEWITT D J, GHANDEHARIZADEH S, SCHNEIDER D A, et al. The Gamma database machine project[J]. IEEE Transactions on Knowledge & Data Engineering, 1990, 2(1): 44-62.
- [8] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.
- [9] KORNACKER M, BEHM A, BITTORF V, et al. Impala: a modern, open-source SQL engine for Hadoop[C]//Conference on Innovative Data Systems Research. [S. l. : s. n. ], 2015.
- [10] ZAHARIA M, CHOWDHURY M, DAS T, et al. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing[C]//The 9th USENIX Conference on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2012.
- [11] ALEXANDROV A, BERGMANN R, EWEN S, et al. The stratosphere platform for big data analytics[J]. VLDB Journal, 2014, 23(6): 939-964.
- [12] DONALD F, ADAM R. Hype cycle for data management[Z]. 2019.
- [13] 孙元浩. 大数据进入3.0时代[J]. 信息技术与标准化, 2019(5): 94-104.  
SUN Y H. Big data enters 3.0 era[J]. Information Technology & Standardization, 2019(5): 94-104.
- [14] 刘汪根, 孙元浩. 大数据3.0——后Hadoop时

代大数据的核心技术[J]. 数据与计算发展前沿, 2019, 1(1): 94-104.

LIU W G, SUN Y H. Big data 3. 0, the key

technologies of big data in post-Hadoop era[J]. Frontiers of Data and Computing, 2019, 1(1): 94-104.

#### 作者简介



**夏正勋** (1979- ), 男, 星环信息科技(上海)有限公司高级研究员, 主要研究方向为大数据、数据库、人工智能、流媒体处理技术等。



**罗圣美** (1971- ), 男, 博士, 星环信息科技(上海)有限公司大数据研究院院长, 主要研究方向为大数据、并行计算、云存储、人工智能等。



**孙元浩** (1976- ), 男, 星环信息科技(上海)有限公司创始人, 从2009年开始研究大数据技术, 2013年创立公司, 并开始新一代大数据技术的自主研发。



**唐剑飞** (1986- ), 男, 星环信息科技(上海)有限公司大数据技术标准研究员, 主要研究方向为大数据、数据库、图计算等。



**张燕** (1985- ), 女, 星环信息科技(上海)有限公司大数据技术研究员, 主要研究方向为大数据、人工智能等。

收稿日期: 2020-05-09