计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(10):46−53 [doi: 10.15888/j.cnki.csa.006599]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

融合大数据技术架构在大型商业银行海量数据分析的应用①

金磐石[1], 朱　志[2], 沈丽忠[2]

[1](中国建设银行股份有限公司, 北京 100025)
[2](中国建设银行股份有限公司 厦门开发中心, 厦门 361008)
通讯作者: 沈丽忠, E-mail: shenlizhong.zh@ccb.com

摘　要: 相对于软硬件一体的企业级数据仓库解决方案, 以 X86 开放式硬件平台为基础的并行处理数据库和 Hadoop/Spark 生态体系, 在扩展性和低成本方面有明显的优势, 随着其不断发展成熟, 已经成为企业级大数据平台不可或缺的一部分. 本文从如何应对“大数据”所带来的挑战出发, 提出一套适用于大型商业银行数据分析处理的融合大数据技术架构, 并已取得应用实践的成功.

关键词: 大数据; 融合技术架构; 数据分析; 数据仓库; 商业银行

引用格式: 金磐石,朱志,沈丽忠.融合大数据技术架构在大型商业银行海量数据分析的应用.计算机系统应用,2018,27(10):46–53. http://www.c-s-a.org.cn/1003-3254/6599.html

Application of Fusion Big Data Architecture in Large Commercial Bank’s Data Analysis Platform

JIN Pan-Shi[1], ZHU Zhi[2], SHEN Li-Zhong[2]

[1](China Construction Bank Copporation Ltd., Beijing 100025, China)
[2](Xiamen Development Center, China Construction Bank Copporation Ltd., Xiamen 361008, China)

Abstract: Compared with traditional Enterprise Data Warehouse (EDW) solution having vendor lock-in risk on both hardware and software, Massive Parallel Processing (MPP) database and Hadoop/Spark ecosystem based on open hardware platform, like X86, have advantages on both scalability and low cost. And with the mature of these technologies, they become integral parts of enterprise-class big data platform. Based on how to deal with the challenges posed by “big data”, this study proposes a fusion big data architecture suitable for data analysis and processing of large commercial banks, and has achieved the success of application practice.

Key words: big data; architecture; data analysis; data warehouse; commercial bank

1　现状与挑战

随着以社交媒体的兴起、智能设备的普及, 现代社会已经步入“大数据”时代, 它将对社会、商业和个人都产生巨大而深远的影响. 根据维基百科的定义[1], “大数据”是指无法在一定时间内用传统数据库软件工具对其内容进行抓取、管理和处理的数据集合; 而大数据技术指的是新一代的技术与架构, 在成本可控的前提下, 它可以通过快速的采集、发现和分析, 从海量的、多样的数据中提取价值. 这个定义描述了大数据的三个特征: Volume(海量)、Velocity(高速)、Variety(多样), 即俗称的“3V”. 第一个是 Volume(海量), 数据量越来越大, 已经超出传统技术和功能的存储和处理能力; 第二个是 Velocity(速度), 数据量增长越来越快, 需要处理的速度和响应的时间要求也越来越高; 第三个是 Variety(多样性), 指各种各样类型的数据出现, 过去的数据更多的是结构化的, 现在越来越多的数据是

① 收稿时间: 2018-03-06; 修改时间: 2018-04-18; 采用时间: 2018-04-20; csa 在线出版时间: 2018-09-28

半结构, 甚至是完全没有结构的数据, 如文本、邮件甚至于语音、视频等. “3V”是对大数据最基本特征的归纳, 得到业界的共识, 同时也是我行面临的主要挑战.

一方面, 在“大数据”时代背景下, 我行数据线架构面临“大数据”三个基本特征带来的冲击. 首先, 数据量持续增长, 我行拥有 6 亿客户和 20 亿账户, 由此产生的每日数据增量达 2~3TB,并以每年 10% 以上的速度增长, 然而传统软硬件一体的数据仓库解决方案成本高昂, 动则上亿元的升级成本, 令其无法满足我行在成本可控的前提下高效地存储、处理不断增长业务数据的需求; 其次, 原有的数据分析处理基本以 T+1 的批处理为主, 无法满足像反欺诈、运营指标实时分析/探索等需要实时响应的交互式数据分析、探索的业务需求; 最后, 传统的方案只能处理结构化的数据, 然而, 除了传统结构化的数据外, 越来越多的业务需求需要结合半结构化、非结构化的数据进行分析、决策, 如记录客户使用行为的网上银行系统日志、记录电话客服的语音和文本数据和第三方社交媒体数据等.

另一方面, 在新一代实施前, 我行数据线已具备相当规模, 包括数据仓库、操作数据存储、分行操作数据存储、历史数据存储和管理分析类应用在内的数据线为全行提供经营、管理、决策所需的信息, 在客户关系管理、资产负债管理、风险管理等方面的应用在国内同业处于领先水平. 但在数据管理上, 缺乏统筹考虑, 部门级应用各自重复存储数据; 在数据应用上, 应用模式单一 (报表), 报表使用效率不高, 开发周期长. 这些问题, 导致数据在完整性、一致性、准确性和时效性上存在诸多质量隐患, 影响了数据线价值的实现.

2 方案架构

2.1 “新一代”整体架构

面对这些挑战, 我行“新一代”从业务架构及 IT 架构两大方向入手, 建立起统一、规范的企业级架构规划. 采用基于 SOA 的组件化设计方法[2], 将 IT 架构整合为 7 层 (细化为 12 个技术平台), 如图 1 所示, 各个区的具体定位和功能为[3]:

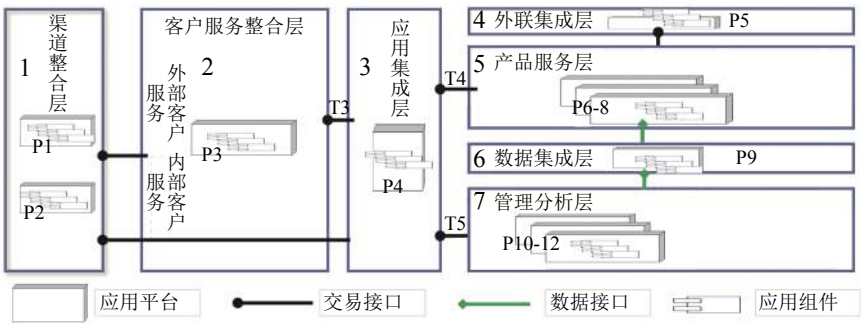

图 1 “新一代”整体架构

(1) 渠道整合层 (包含 P1 和 P2) 的作用是整合渠道接入, 统一渠道接口处理. 通过标准的方式接入、处理并封装请求数据和调用后台服务返回结果, 渠道整合层响应来自不同渠道的业务请求.

(2) 客户服务整合层 (包含 P3) 从客户视角出发, 通过整合端到端的客户服务流程, 为内外部客户提供具备良好体验的银行产品和集成服务.

(3) 应用集成层 (包含 P4) 通过标准化方式接受来自客户服务整合层及渠道整合层的服务请求, 屏蔽后台服务的技术特性, 以一致的形式返回结果. 主要功能包括服务目录管理、报文转换、交易路由、服务调用、资源分配等;

(4) 外联集成层 (包含 P5) 负责我行“新一代”与外部金融和非金融机构的系统交互, 包含如第三方支付系统、保险公司、人行征信系统等. 主要完成协议转换、报文转发和等功能, 为内部系统屏蔽各类网络接口的技术特性.

(5) 产品服务层 (包含 P6, P7, P8) 提供企业视角的产品服务, 支持银行集团核心业务, 涉及产品、客户、合约、核算等服务, 提供与渠道无关的产品、服务处理组件. 产品服务层以应用组件作为服务的载体, 将服务发布在应用集成层.

(6) 数据集成层 (包含 P9) 整合企业范围内的各类数据, 为产品服务层和管理分析层提供统一的数据服

务. 数据集成层可提供面向企业级数据的集成服务, 是企业级信息视图的必要条件, 实现了企业级统一的数据接入、数据存储、数据批处理、数据分析、数据归档、文件交换、数据访问、元数据管理和任务调度等能力.

(7) 管理分析层 (包含 P10, P12) 为银行内部管理和决策提供支持. P10 主要的功能是搭建企业级数据应用平台基础设施, 为各类管理分析类应用提供报表、自助维度分析、即席查询和数据挖掘等服务; P12 主要为在线交易运营决策服务和信息访问服务提供开发和运行支持, 具备灵活的规则定制和高效的数据访问能力.

2.2 数据线逻辑分区

在“新一代”整体架构内, 数据线主要包括数据集成层 (P9) 和管理分析层 (P10 和 P12), 其细化的逻辑分析如图 2 所示, 包含如下几个数据区:

(1) 数据缓存区, 它支持数据集散存放, 作为源系统数据的暂存区. 数据模型与组件数据格式相近, 以文本文件方式存储. 支持做一部分简单的数据处理: 如基本的数据的转码、标准化、技术清洗; 完成基于增量数据的快速/简单加工等. 数据保留的周期较短, 一般为 15 天.

(2) 数据仓库区, 主要完成企业级数据整合, 支持全行公共计算和应用计算, 提供数据给全行使用. 数据仓库区又细分为五个区: 贴源区, 基础主题区, 公共计算区, 应用计算区, 公共访问区, 他们的功能和定位分别为:

① 贴源区. 顾名思义, 它的数据模型主要为贴源结构 (贴近源系统/组件), 同时采用拉链表技术降低数据冗余量. 提供前日快照数据, 供日常加工计算使用; 提供 30 天以内的截面全量; 支持快速、跨业务领域的应用计算; 设有存放半结构化数据的子分区, 提供半结构化数据转换为结构化数据、半结构化数据预处理等功能. 数据保存周期一般为 30 天.

② 基础主题数据区. 主要完成企业级的基础数据整合计算, 包括按基础主题模型重新组织数据、合并不同组件数据到同一实体中. 数据以 3NF 方式完成数据合并, 形成企业级全口径, 确保模型稳定.

③ 公共计算区. 主要做公共指标和衍生数据的计算加工, 具体还包含完成账户、客户和流水粒度的衍生加工; 完成通用的交叉维度衍生计算; 完成分行共用指标的加工.

④ 应用计算区. 主要做与特定应用相关的指标和衍生数据加工.

⑤ 公共访问区. 涵盖基础数据、衍生数据等全部数据; 支持非固定、高并发的用户对全部数据的访问; 支持 SAS, 支持少数用户对全部数据的分析和挖掘.

(3) 实验数据区. 主要用于数据深度探索、模型研发、和需求验证等目的. 模型训练结果是模型及参数, 由应用计算区负责对应用模型训练结果进行加工; 支持管理分析应用需求分析人员确认口径; 一般按月更新样本数据, 同时也支持按需同步数据; 支持分行以实验项目方式使用实验数据区进行模型训练.

(4) 历史数据归档区. 主要用于源系统数据的归档和数据仓库历史数据归档, 同时也支持对归档数据的联机访问.

(5) 非结构化数据区. 顾名思义, 主要提供非结构化数据存储、访问和分析.

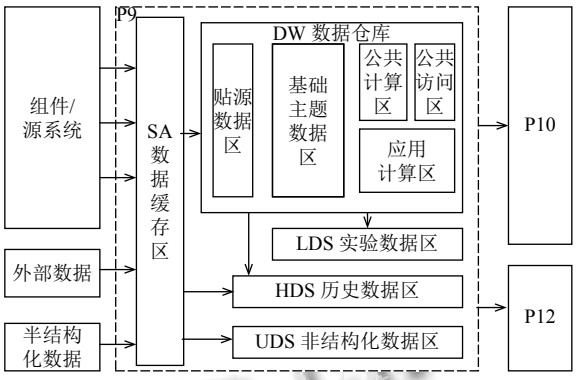

图 2 数据线逻辑分区

3 关键技术方案和策略

3.1 海量数据的存储和处理

新一代前我行的数据仓库主要采用软硬件一体的数据库和 SAN 存储存放和处理数据, 存在成本居高不下、扩展能力有限、支持的数据类型有限 (不支持半结构化和非结构化) 等问题. 随着业务数据的快速增加和业务需求的涌现, 这些问题日渐突出.

基于 X86 开放平台的新技术在大数据时代以其相对合理的成本和高可扩展性逐渐受到业界的关注和应用, 新一代实施需要适时对新技术进行探索和使用.

3.1.1 MPP 数据库技术

MPP (Massive Parallel Processing) 数据库, 即大规模并行处理数据库系统, 由多个独立的节点组成, 每个节点只访问自己的本地资源 (如内存和存储等), 节点之间通过网络连接通讯, 是一种完全无共享架构.

MPP 架构数据库具备如下特点:

① 拥有较强的海量计算能力: 所有数据分布到所有节点, 每个节点都计算自己的部分数据, 并行处理无需人工干预, 系统自动通过增加通用硬件来扩充新的计算结点; ② 具有较强的容错能力: 对于普通的计算节点, 一份数据分布在多个节点上,避免出现由于单个节点出现故障而导致的数据丢失和系统不可用. 对于管理节点, 一般也采用高可用设计, 避免了单点故障; ③ 移动计算比移动数据更方便: 由于数据分布的特性,可利用普通计算节点的处理能力, 处理本机器上的数据, 从而尽量避免数据在不同节点之间的大量数据移动; ④ 较好的扩展性: 可线性扩展到上百个节点, 每增加一个节点, 查询、加载性能都成线性增长; ⑤ 数据管理相对简单: 无需复杂的调优需求, 只需要加载数据和查询, DBA 工作量较少, 无需复杂的调优工作和维护工作; ⑥ BI 工具支持比较成熟, 一般能够广泛地支持各个 BI 和 ETL 软件工具. 除此之外, 不同产品在事务支持、数据存储格式、存储分布优化等方面都有不同程度的增强.

3.1.2 Hadoop/Spark 技术

Apache Hadoop[4]是 GOOGLE 大数据技术存储、计算框架的开源实现, 具备良好的可扩展性、低廉的硬件成本、高度容错、较强的灵活性等特点. 它包含以下几个核心组件:

(1) HDFS(Hadoop Distribute File System) 基于 Google 发布的 GFS 论文[5]设计开发, 其除具备其他分布式文件系统相同特性外, 还有自己特有的特性: 高容错, 能很好地应对通用硬件的单点故障; 高吞吐量, 为有大量数据访问的应用提供高吞吐量支持; 大文件存储, 支持存储 TB-PB 级别的数据. HDFS 特别适合大文件一次写入, 多次读取的情景, 而不适合存储大量小文件、随机写入和低延迟读取的情景

(2) MapReduce 是基于 Google 发布的 MapReduce 论文[6]开源实现一个并行计算框架, 该框架把复杂的并行计算抽象为 Map 和 Reduce 两个阶段, 用户开发应用时无需关注一系列复杂的分布式计算问题, 如任务调度、资源分配、容错和数据分片等, 只需重点关注如何把要解决的问题转换为一系列 Map 和 Reduce 操作.

(3) YARN (Yet Another Resource Negotiator)[7]是一个通用资源管理系统 (主要包含集群的 CPU、内存和 IO 计算资源), 可为上层应用提供统一的资源管理和调度服务, 能为集群的利用率、资源统一管理和数据共享等方面带来了大幅提升. 在它之上, 可以运行不同的并行计算框架, 如 MapReduce, Spark, Tez 等.

在 Hadoop 核心组件的基础之上, 又陆续发展出多种适用于不用使用场景的技术产品, 如图 3 所示, 适用于海量数据批处理的 Hive, Pig 等; 适用于交互式分析的 Impala、Kylin 等; 适用于数据挖掘的 Mahout 等; 适用于实时计算的 Storm 和适用于高并发在线访问的 HBase 等 NoSQL 产品. 此外还包含用于集群部署安装的 Ambari; 用于元数据管理的 HCatalog; 用于底层分布式协同的 Zookeeper; 用于作业编排的 Oozie; 用于提供友好可视化访问、分析界面的 Hue 和 Zepplin 等产品.

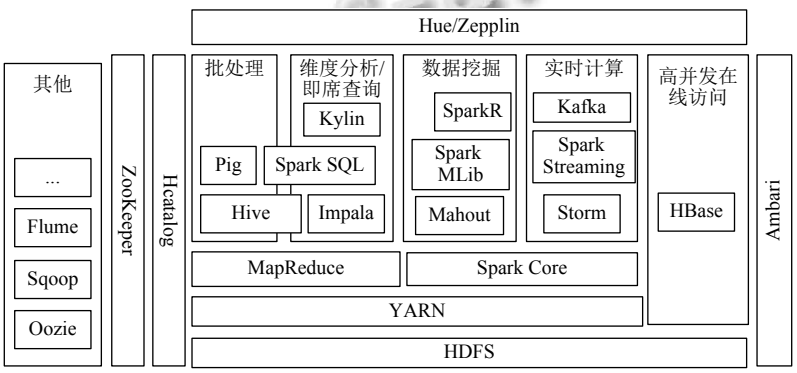

图 3 Hadoop/Spark 生态体系

Apache Spark[8]是为大规模数据运算而设计的一个开源通用计算框架, 最初是由加州大学柏克莱分校 AMP Lab 所开发. 相对于 Hadoop 的 MapReduce 重度依赖磁盘缓存中间计算结果不同, Spark 尽可能地基于缓存在内存中的数据进行运算, 大幅度提高并行计算速度, 特别适合需要多次迭代的复杂运算场景, 如数据

挖掘. 和 Hadoop MapReduce 相比, 根据不同的场景, 运算速度能提升 10~100 倍. 除了性能上的提升, Spark 具备另外一个独特的优势: 依赖基于 Spark Core 之上的 Spark SQL, Spark MLib 和 Spark Streaming 等库, Spark 可以支持批处理、即席分析、数据挖掘和实时计算多种使用场景,大幅减低了系统部署和运维的复杂度.

3.1.3 MPP 数据库技术与 Hadoop/Spark 技术对比

MPP 数据库技术与 Hadoop/Spark 技术各有特点, 其主要的区别如表 1 所示, 整体而言, Hadoop/Spark 技术具备更好的扩展能力、更低的软硬件成本和多样化数据处理能力, 而 MPP 数据库技术在对 SQL 支持、事务支持、BI 工具支持等方面较 Hadoop/Spark 技术成熟, 而且对人员技能要求相对较低, 后期运维投入成本也会比较低. 简而言之, MPP 数据库技术和 Hadoop/Spark 技术各有优劣, 各有各适用的场景, 需要综合多种因素进行选择.

表 1 MPP 技术与 Hadoop/Spark 技术对比

	MPP 数据库	Hadoop/Spark
SQL 支持	成熟	一般
响应时间	秒~分钟	秒~分钟
软硬件成本	较高	较低
事务支持	好	一般
扩展性	100~200	>1000
运维成本	低	高
对人员技能要求	低	高
处理的数据类型	格式化数据	格式化和非格式化数据
BI 工具支持	好	一般

3.1.4 我行的技术引入策略

我行数据线各个逻辑分区的应用特点如表 2 所示.

表 2 数据线各个逻辑分区的应用特点

	数据量	数据格式	使用方式	并发度
贴源数据区	TB	结构化	ETL	中
基础主体数据区	TB	结构化	ETL/复杂分析	中
公共计算区	PB	结构化	ETL	中
公共访问区	PB	结构化	OLAP 查询	高
应用计算区	TB	结构化	ETL	中
实验数据区	TB	结构化	深度挖掘	中
历史数据归档区	PB	结构化/非结构化	归档/ETL	低
历史数据在线访问区	PB	结构化/非结构化	固定模式访问	超高
非结构化数据区	PB	非结构化	ETL/分析	中

根据 MPP 数据库、Hadoop/Spark 技术各自的特点和数据线各个逻辑分区的应用特点, 在综合考虑到快速见成效、人员技能、开发模式等因素, 我行引入基于 X86 开放式硬件平台的 MPP 数据库技术产品, 应用于贴源数据区、应用计算区和实验数据区, 降低 Teradata 平台的处理压力, 为 Teradata 平台减负, 另一方面, 综合考虑到产品的成熟度、特性、业界最佳实践和应用迁移成本等因素, Teradata 依然承担基础主题数据区和公共计算区的负载. 考虑到 Hadoop/Spark 技术良好的可扩展性、较低的软硬件成本和多样化数据处理能力, 将其运用于历史数据归档区存储源系统和仓库海量备份数据; 运用于历史数据在线访问区提供高并发且访问模式相对固定的历史数据在线访问; 运用于非结构化数据区对非结构化数据的存储、加工和分析.

此外, 对于公共访问区, 单一的产品无法很好地满足其相对比较有挑战的功能和非功能需求, 为此我们为其提出如下解决方案.

3.2 联机在线分析处理的快速响应

联机分析处理 (OnLine Analytical Processing, 简称 OLAP) 的概念最早由关系数据库之父 Codd EF 于 1993 年提出[9]. 不同于面向基本、日常事务处理的联机事务处理 (OnLine Transaction Processing, 简称 OLTP), 例如银行交易, 它支持复杂的分析操作 (如多维度分析), 让用户能够根据自身需求, 多维度、多角度地快速洞悉原始数据隐含的价值, 并为其商业决策提供强有力的支持. 它是商业智能和数据库仓库的基础. 随着数据平民化趋势的不断发展, 联机分析类应用逐渐对更多的管理人员和业务人员开放, 这对公共访问区的联机分析处理的访问性能和并发度提出更高的要求. 然而, 现有的产品, 无论是 MPP 数据库还是单一的 Hadoop/Spark 技术产品都无法很好地同时满足应用在功能和非功能方面的要求.

目前可用于 OLAP(联机分析) 的产品主要分为两

大类: 多维联机分析处理 (Multidimensional OnLine Analytical Processing, 简称 MOLAP) 和关系型联机分析处理 (Relational OnLine Analytical Processing, 简称 ROLAP). MOLAP 以多维数据组织方式为核心, 使用多维数组存储数据, 把数据组织成"立方块 (Cube)"的结构, 然后对"立方块"进行"旋转"、"切块"、"切片"等操作来形成多维数据报表需要的数据. 它的特点是将聚合后的数据保存在 Cube 中, 以空间换效率, 查询时效率高, 但生成 Cube 时需要大量的时间和空间.

在大数据领域, Apache Kylin[10]是 MOLAP 技术路线的典型代表. 单从公开的测试数据来看[11], 它的查询性能相比 ROLAP 要好很多. 当然, Kylin 的亚秒级响应也是要付出一定的代价, 即预计算的代价, 包含两部分: 其一是预计算所需计算资源和时间, 其二是保存预计算结果 (即 Cube) 所需的存储空间. 如果不对预计算的维度组合进行合理的选择, 很容易引起 Cube 作业失败或维度爆炸等问题. 对此, Kylin 提供了部分 Cube 优化的功能, 帮助用户控制数据膨胀问题. 总体而言, 要在查询性能和预计算代价之间做出合理的权衡, 需要用户对其工作原理和业务查询模式都有较深的理解. 除此之外, Apache Kylin 也不是能够很好地满足所有的场景, 比如: 明细数据的查询; 高基维的指标查询和即席查询等.

ROLAP 以关系数据库为核心, 以关系型结构进行多维数据的表示和存储, 星型模型为其数据模型的典型代表: 包含一张事实表和零个或多个维度表, 其中事实表存储指标明细数据 (如销售额等), 维度表存储维度信息 (如销售的商品、地区等), 事实表和维度表之间通过主键外键关联, 维表之间没有关联. ROLAP 的特点是数据组织相对比较灵活, 不需要进行预计算, 但是它的查询性能相对 MOLAP 来说比较差.

在大数据领域, Hive, SparkSQL, Presto, Impala 等 SQL on Hadoop 产品本质上是采用大数据技术来构建 SQL 引擎实现或部分实现数据库的能力, 所以他们不仅可以用来支撑 ROLAP 应用, 即对以维度模型方式组织的数据进行查询分析, 也支持对以贴源方式组织的数据进行查询分析. 相对 MOLAP 产品来说, 他们具备更高的灵活度和适应性.

MOLAP 和 ROLAP 各有优缺点, 由此在传统的 OLAP 技术领域还出现了 HOLAP (Hybrid OnLine Analytical Processing, 简称 HOLAP) 技术, 试图吸取 MOLAP 和 ROLAP 各自的长处来满足 OLAP 分析. 但是在大数据领域, 根据我们的调研了解, 目前还没有一个成熟的开源 HOLAP 产品能满足我行联机分析应用的需求.

为此, 基于以上对不同 SQL on Hadoop 引擎的调研和比较, 并结合我行的实际业务场景需求, 在此提出一套 HOLAP 技术方案, 如图 4 所示: 其核心思想是利用 MOLAP 的预计算技术预先聚合、统计出常用的 OLAP 分析结果, 满足相当大一部分 OLAP 查询分析请求, 对于其他不适合用预计算来满足的请求 (比如明细查询或基数太大的维度), 下推到底层的 ROLAP 引擎, 利用其高效的 SQL 引擎查询出结果返回给客户端, 对于计算代价较高的查询请求, 可以进一步缓存在分布式缓存中, 减少后端 OLAP 引擎的压力.

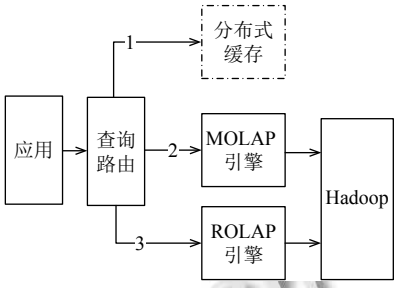

图 4 混合技术架构图

此方案的先决条件就是要选择合适的 MOLAP 引擎和 ROLAP 引擎. Apache Kylin 和 Apache Druid 都属于大数据领域的 MOLAP 开源产品. 但是 Apache Druid 的查询接口不是 SQL, 且有不支持 join 等缺陷, 最后我们根据对 SQL 支持程度、适用场景、业界参考案例等方面的对比, 选择 Apache Kylin 作为我们方案中的 MOLAP 引擎. 图 5 为在 Apache Kylin 上对 SSB 测试数据集进行测试的结果[11], 其中 Kylin SF 10, Kylin SF 20 和 Kylin SF 40 分别代表在 SSB 基准测试数据上翻 10 倍, 20 倍和 40 倍的测试结果. 从测试结果可以看出, 经过预计算, Apache 的查询结果基本都在 600 毫秒以内, 而且查询时间不随着数据量的增长而增加. 不过预计算的时间和计算结果所占用的空间会随着数据量的增长而线性增加的.

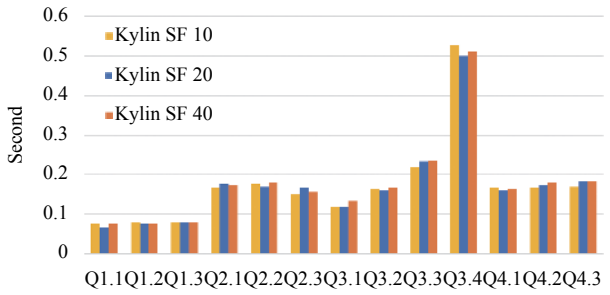

图 5 MOLAP (Apache Kylin) 运行 SSB 大数据测试的结果[11]

另一方面, Hive, SparkSQL, Presto, Impala 等都具备 ROLAP 引擎的能力, 根据公开的测试结果 (图 6), 这些引擎各有所长, 综合考虑 ROLAP 引擎性能、Hadoop 发行版兼容性、运维成本和团队技能等因素, 我们选择 SparkSQL 作为此方案的 ROLAP 引擎. 由测试结果可以知道, ROLAP 引擎的响应时间一般是在秒到分钟之间, 具体取决于数据量、查询语句复和计算资源等因素.

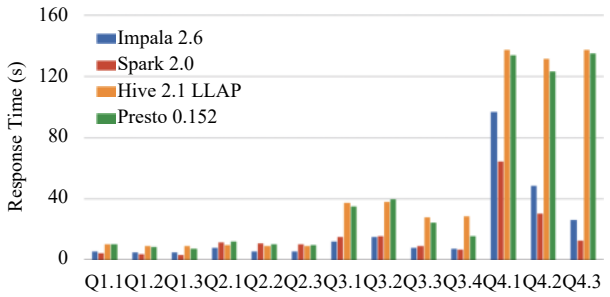

图 6 ROLAP 运行 SSB 大数据测试的结果[12]

此方案既吸取 MOLAP 引擎查询速度快的优势, 又汲取 ROLAP 引擎灵活和适应性强的优点, 实现了在有效控制成本的前提下, 大大提高整体的 OLAP 查询处理性能的目标.

4 结论

依托新一代构建的 7 层 12P 中的数据集成层 (P9) 和管理分析层 (P10, P12) 的能力, 我行在企业运营管理、风控管理和产品与服务等方面的应用都取得较大的进步和提升.

以运营管理为例, 为我行数据管理和日常运营打造的移动端数据可视化项目--“慧视”, 拥有“慧决策”、“慧管理”、“慧算账”、“鹰眼”等系列产品, 着眼于移动优先、数据可视化、便捷高效、体验第一的项目目标, 满足总行领导、分行行长、支行行长、网点负责人、各管理条线负责人等全行各级管理者的数据需求. 依托企业级统一数据视图, 自 2017 年上线以来, 已经涵盖 16 个岗位角色、实现 600+指标数据的移动端可视化. 目前, “慧视”已在全行范围内推广, 得到各级管理者的广泛好评.

在产品服务方面, 基于企业级数据分析平台,对客户定活期资产、贷款、代发工资、公积金、征信等数据进行深度挖掘, 对客户进行差额化授信, 灵活设置多种服务模式, 在有效控制风险的前提下提升客户体验和服务水平,使快贷业务规范化、规模化发展. 截至 2017 年 9 月底, 个人快贷累计服务客户达 500 万户, 额度授信金额 3000 亿元, 贷款余额 1377 亿元. 另一方面, 建设银行以小微企业金融服务为履行社会责任的重点, 不断拓宽小微企业金融服务覆盖面. 从 2016 年 6 月投产至 2017 年 8 月底, 已成功授信 113 451 笔, 授信金额 709.9 亿元, 客户累计支用 450.2 亿元 , 授信客户数从 1.02 万户增长到 8 万多户, 增长幅度高达 684%.

在风险管控方面, 随着企业级反欺诈联合防控方案落地, 业务功能全部释放, 实现“五个首次”——首次建立信用卡、借记卡的首笔实时侦测能力, 首次将借记卡、善融商务、电话支付业务纳入反欺诈监测范围, 首次将监测对象扩大到所有对私客户和收单商户, 首次建立统一的欺诈告警和案件管理机制, 首次建立全面的客户行为档案.

实践证明, 商业银行通过引入基于开放硬件平台的技术和产品, 结合自身的业务和应用特点自主研发融合多种大数据技术的企业级分析处理平台是可行的, 而且比单一的、传统的软硬件一体方案具有明显的优势.

参考文献

1 Big data. https://en.wikipedia.org/wiki/Big_data. [2018-08-18].

2 金磐石. “产、用”战略联动 推进信息化自主可控进程. 金融电子化, 2014, (8): 28–29.

3 金磐石, 朱玉红, 胡宪忠, 等. 基于 SOA 的银行集团“新一代”系统架构. 计算机系统应用, 2016, 25(12): 42–52. [doi: 10.15888/j.cnki.csa.005542]

4 What is Apache Hadoop? https://hadoop.apache.org/. [2017-08-09].

5 Ghemawat S, Gobioff H, Leung ST. The Google file system. The 19th ACM Symposium on Operating Systems

Principles. New York, NY, USA. 2003. 29–43.

6 Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation. San Francisco, CA, USA. 2004. 10.

7 Vavilapalli VK, Murthy AC, Douglas C, *et al*. Apache hadoop YARN: Yet another resource negotiator. Proceedings of the 4th annual Symposium on Cloud Computing. Santa Clara, CA, USA. 2013. Article No.5 [doi: 10.1145/2523616.2523633]

8 http://spark.apache.org/.

9 Codd EF, Codd SB, Salley CT. Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Codd and Date Inc., 1993.

10 Apache Kylin™. Apache Kylin™ overview. http://kylin.apache.org/.

11 https://github.com/Kyligence/ssb-kylin.

12 Performance benchmark: Business intelligence on big data Q4'16. http://info.atscale.com/atscale-business-intelligence-on-hadoop-benchmark.