关系型与非关系型数据库融合技术在海量大数据存储中的应用

马亚楠
新疆天山职业技术大学　新疆　乌鲁木齐　830000

摘　要：随着互联网、物联网及人工智能技术的快速发展，企业面临的数据量呈指数级增长，单一的关系型数据库或非关系型数据库均难以兼顾海量数据存储的规模性、一致性与查询效率要求。通过将结构化核心数据存储于关系型数据库，非结构化及高并发写入数据存储于非关系型数据库，并设计双向数据同步与跨库查询引擎，实现存储成本与访问性能的优化平衡。该融合架构在典型业务场景中有效解决了单一数据库的性能瓶颈，为大数据应用提供了可落地的存储解决方案。

关键词：关系型数据库；非关系型数据库；数据融合；海量存储；混合架构

引言：在移动互联网、智能制造与智慧城市建设的推动下，企业数据采集来源日益多样化，数据结构从传统单一的行列表格扩展至键值、文档、时序、图结构等多种形态。关系型数据库以严格的事务与强一致性见长，但在海量高并发写入及横向扩展方面存在明显瓶颈；非关系型数据库虽具备高扩展性与灵活数据模型，却难以支撑复杂关联查询与强事务场景。本文聚焦于混合存储架构设计，提出一套融合存储方案，通过数据特征识别与智能路由机制，实现两类数据库的优势互补，为海量大数据存储提供可参考的技术路径。

1 数据库技术在海量存储场景下的能力边界分析

1.1 关系型数据库的架构特征与瓶颈

关系型数据库以结构化查询语言为核心接口，依托预定义的表结构与关系约束，通过事务日志、锁机制及多版本并发控制保障数据的强一致性与隔离性。在高精度交易、财务核算及企业管理系统等场景中，其事务特性具有不可替代的地位。然而，面对海量大数据写入与存储需求，关系型数据库暴露出明显的架构瓶颈。首先，垂直扩展受制于单机硬件上限，通过增加CPU核心数与内存容量虽然能够缓解压力，但成本呈非线性增长；其次，水平分库分表虽能分摊负载，却会破坏原有的外键与联接查询语义，迫使应用层接管分布式事务与聚合计算。当单表数据量突破千万级别时，多表关联查询的响应时间往往从毫秒级恶化至秒级，索引膨胀与写入锁竞争进一步加剧性能下降。此外，关系型数据库对半结构化和非结构化数据的支持能力有限，强行转换数据模型会带来额外的存储开销和复杂的数据清洗流程[1]。

1.2 非关系型数据库的优势领域与适用约束

非关系型数据库摒弃了固定的表结构，针对特定数据模型进行原生优化，天然支持分布式架构。键值型数据库以哈希结构实现O(1)复杂度的点查询，广泛应用于缓存与用户画像场景；文档数据库以JSON格式存储自描述数据，与前端对象模型天然对齐，减少了对象关系映射的转换损耗；列族数据库面向大规模分析负载优化，支持极高的压缩比与宽表扫描，适合海量数据的批量处理；时序数据库针对时间戳密集的物联网数据进行降采样与滚动聚合，有效管理时间序列数据。然而，非关系型数据库在丧失事务能力、外键约束与标准化查询语言的同时，对跨集合、跨分片的复杂关联查询支持较弱，通常需要在应用层进行多轮查询与内存归并，显著增加了开发复杂度与响应延迟。当业务需要多表联接或强数据一致性时，非关系型数据库的局限性便凸显出来。

1.3 单一架构面临的核心矛盾

在海量大数据存储实践中，单一的数据库技术路线面临着三个核心矛盾。第一，事务一致性与系统扩展性的矛盾。强事务要求分布式锁与两阶段提交，会严重阻滞数据节点的横向扩展能力；而追求高可用与最终一致性的系统，则无法承载金融级事务负载。第二，数据多样性与存储模型的矛盾。同一业务系统中的用户档案适合用关系表存储，用户行为日志更适合文档或列族模型，单一数据库无法同时高效处理这两种写入与查询模式，强制转换必然带来性能损耗。第三，存储成本与访问性能的矛盾。海量冷数据适合廉价存储介质与高压缩率格式，而热数据需要高速缓存与快速索引。单一数据库只能采用无差别存储策略，造成资源浪费。这三重矛盾驱动着融合存储架构的出现与发展。

2 融合存储架构设计

2.1 基于数据特性感知的混合存储模型

为了解决单一数据库架构的核心矛盾，本文设计了一套基于数据特性感知的混合存储模型。该模型的核心思想是根据数据的结构特征、访问频率与事务要求，对入库数据进行自动归类并路由至最合适的存储引擎。模型定义三类数据通路：通路一定位为强事务性结构化数据，如订单记录、账户余额、交易流水等，将其路由至关系型数据库集群，保障ACID特性；通路二定位为高并发写入的无结构或半结构化数据，如应用日志、埋点事件、设备上报数据，将其写入非关系型数据库，利用其海量写入能力与弹性扩展优势；通路三定位为需要跨库关联查询的数据视图，通过物化视图或数据管道技术将部分关系型数据定期同步至非关系型数据库，加速分析场景[2]。该模型的核心优势在于，业务应用无需感知底层存储细节，只需按照统一接口写入数据，系统自动完成分类与存储，大幅降低了应用改造的复杂性。

2.2 融合架构总体框架

融合存储架构总体框架由数据接入层、特征识别引擎、路由控制器、双存储引擎层及统一查询接口五部分构成。数据接入层对上游业务屏蔽底层存储细节，接收标准SQL或API写入请求，提供统一的访问入口。特征识别引擎解析请求中的元数据，结合预定义策略表，判断当前数据应当归属的存储类别。策略表支持动态配置，允许数据管理员根据业务变化调整分类阈值，如调整事务优先级、冷热数据判断标准等。路由控制器依据识别结果，将写入请求转发至对应的关系型数据库或非关系型数据库集群，同时记录数据的位置元数据至目录服务模块。双存储引擎层由关系型数据库集群与非关系型数据库集群并行组成，彼此之间通过异步数据同步管道保持关键数据的一致性。统一查询接口接受跨库查询请求，通过解析查询计划，将子查询分发至不同引擎执行，并在内存中进行结果集的归并与排序，最终返回给业务应用，实现了对上层业务的透明化数据访问（如表1所示）。

表1：数据特征识别与路由策略示例表

数据类别	数据结构	事务要求	访问模式	并发写入量	路由目标
订单主表	强结构化	强ACID	低频点查	低	关系型数据库
用户行为日志	半结构化	最终一致	批量分析	极高	非关系型数据库
设备实时状态	时序型	最终一致	范围扫描	极高	非关系型数据库
热数据缓存	键值型	弱	高频随机	高	非关系型数据库
跨库物化视图	宽表型	弱	分析查询	中	非关系型数据库

表1展示了五类典型业务数据在融合架构中的路由策略，清晰反映了数据结构、事务需求与存储目标之间的映射关系，为实际应用中的策略配置提供了参考依据。

2.3 数据同步与一致性保障机制

融合架构中关系型数据库与非关系型数据库之间的数据同步与一致性是保证系统可靠运行的核心环节。本方案采用基于日志捕获的异步复制策略，关系型数据库的事务日志或归档日志被变更数据捕获组件实时监听，数据变更事件以消息形式写入分布式消息队列，确保变更事件不丢失、不重复。消费端适配器从消息队列拉取事件，按照预定义的映射规则将关系型数据表的行记录转换为非关系型数据库的文档格式或键值对格式，并执行写入操作。对于从非关系型数据库向关系型数据库的反向同步，主要依赖于定时批处理任务，因为非关系型数据库通常不支持细粒度的事务日志。在存在跨库数据冗余的场景下，通过版本向量与时间戳机制检测数据不一致，并由巡检服务定期发起一致性校验与修复任务，确保两类存储之间的核心数据最终一致。同步延迟控制在秒级以内，能够满足绝大多数业务场景的实时性要求[3]。

3 融合架构的应用模式与典型场景

3.1 电商交易系统的融合存储方案

电商交易系统是关系型与非关系型数据库融合存储的典型应用场景。在该场景中，订单主表、商品信息表、用户账户表等核心业务数据具有强事务要求，需要在关系型数据库中存储，以保障交易数据的一致性和完整性。用户浏览轨迹、点击流日志、购物车临时数据、商品评价等数据并发写入量高且结构多变，适合存放于非关系型数据库。在具体实践中，当用户提交订单时，订单核心信息写入关系型数据库，同时将用户本次浏览路径、推荐点击等行为数据异步写入非关系型数据库。后续的订单查询请求可能需要同时获取订单状态与用户行为数据，统一查询接口将分别从两类数据库中获取数据并在应用层完成组装。这种方案既保障了交易数据的ACID特性，又避免了高并发行为日志对关系型数据库的性能冲击。在实际运营中，非关系型数据库承载了超过80%的写入请求，关系型数据库专注于事务处理，系统整

体吞吐量显著提升。

3.2 物联网监控平台的数据分层存储

物联网监控平台需要处理海量设备上报数据，是融合存储架构的另一重要应用场景。设备元数据如设备编号、型号、所属网关、配置参数等数据量小但修改频繁，宜采用关系型数据库存储，便于维护设备间的关联关系与配置版本管理。设备上报的时序采样点数据如温度、湿度、电压等指标数据量极大且极少更新，每秒可能接收数万条写入请求，适配非关系型数据库（尤其是时序数据库）存储，并按天分表管理以便数据老化与清理。在数据查询层面，实时监控需要查询当前最新数据，可直接从非关系型数据库的高速缓存层获取；历史趋势分析则需要扫描大量历史数据，通过非关系型数据库的列式存储与压缩技术，可在秒级内完成百万级数据点的聚合计算。告警规则引擎所需的设备关联信息从关系型数据库加载，告警事件本身则写入非关系型数据库。

3.3 金融风控系统的混合存储实践

金融风控系统对数据一致性和查询性能均有极高要求，融合存储架构在平衡这两方面需求方面展现出独特优势。交易流水数据必须满足强事务与审计追溯要求，归入关系型数据库存储，每一笔交易记录都需要完整的事务日志支持。行为轨迹、设备指纹、IP信誉库等辅助特征数据体量大且结构多变，体量可达交易数据的数十倍，适配非关系型数据库存储。在风控决策流程中，当一笔交易到达时，系统首先从关系型数据库中查询该账户的基本信息和历史交易记录，同时从非关系型数据库中并行获取该账户近期的行为特征、关联设备信息等。两类数据在风控引擎中完成融合计算，输出风险评估结论。这种并行查询模式将风控决策的端到端延迟控制在200毫秒以内，远优于将所有数据存储在单一数据库中时的性能表现。跨库数据的异步关联避免了阻塞主交易链路，即使非关系型数据库出现短暂延迟，核心交易仍可基于关系型数据库中的核心数据完成基础风控判断[4]。

3.4 融合架构的工程落地要点

融合存储架构从理论设计走向工程实践，需要重点关注以下几个方面。第一，数据分片策略的合理规划。关系型数据库的分库分表键应选择业务主键，非关系型数据库的分片键应选择写入分布均匀的字段，避免产生数据倾斜。第二，跨库查询的性能优化。对于高频的跨库关联查询，应设计物化视图或宽表进行预计算，避免每次查询都进行跨库数据拉取。第三，数据同步管道的稳定性保障。变更数据捕获组件需要具备断点续传与故障恢复能力，确保在组件重启后能够从断点继续同步，消息队列的持久化配置能够防止数据丢失。第四，存储成本的持续管控。非关系型数据库中的数据应配置合理的数据生命周期策略，定期清理过期数据，并对冷数据启用压缩存储。通过上述措施，融合架构能够在保障业务功能的前提下，实现性能、成本与运维复杂度的最佳平衡。

结束语

海量大数据存储场景对数据库系统提出了多维度的苛刻要求，单一关系型或非关系型数据库均无法同时满足事务性、扩展性与性能诉求。本文设计了一套基于数据特征感知的关系型与非关系型数据库融合存储架构，通过智能路由引擎、双存储引擎协同与跨库统一查询，有效整合两类数据库的优势。该架构在电商交易、物联网监控及金融风控等典型场景中展现出良好的适用性，通过数据分层存储与差异化访问策略，显著改善了系统性能与资源利用率，为复杂业务环境下的海量数据存储提供了可落地的技术路径。

参考文献

[1]王彦婕.多源异构数据融合技术的研究[J].山西电子技术,2022(3):71-73.

[2]武朝尉.利用NewSQL融合数据库构建数据资源库的探讨[J].信息系统工程,2022(4):71-76.

[3]孙惠芬.基于云计算的海量大数据存储系统设计和实现[J].信息与电脑，2022，34(23):147-149.

[4]曹德建,赵鹏飞,刘硕,等.基于海量数据分层统计、存储的方法研究[J].自动化与仪表,2025,40(5):140-143.