

“十四五”数据库发展趋势与挑战

CCF 数据库专委会

2021 年 12 月

“十四五”数据库发展趋势与挑战

陈群、陈跃国、崔斌、范多锋、高云君、李国良、李战怀、毛睿、潘安群、
彭智勇、钱卫宁、童咏昕、屠要峰、王晓阳、杨晓春、姚斌、袁野、周立柱
(作者按姓名拼音字母顺序排列)

为了总结数据库研究进展、展望十四五期间发展趋势、形成数据库领域发展共识，数据库专委会于 2021 年 7 月 21-24 日在甘肃敦煌组织了首届 CCF 数据库启智会。启智会邀请了 18 位国内数据库专家参与研讨。与会专家逐一介绍了对数据库相关研究方向未来发展趋势的思考，之后所有专家进行了广泛深入的研讨。会后与会专家撰写了相关研究方向的未来发展趋势，最后形成了“十四五”数据库发展趋势与挑战报告。

目录

摘要	4
ABSTRACT	5
1 前言.....	6
1.1 新型数据库系统	6
1.2 新型数据库技术	6
1.3 数据科学	7
1.4 数据库生态建设	7
2. 数据库发展趋势与挑战.....	8
2.1 云原生数据库	8
2.2 分布式数据库	9
2.3 端边云数据库	13
2.4 AI 原生数据库.....	14
2.5 数据库新技术	18
2.5.1 新硬件驱动的数据管理	18
2.5.2 智能化数据管理	19
2.5.3 多模态数据管理	22
2.5.4 数据安全隐私计算	24
2.5.5 其他数据管理技术	31
3. 数据科学的发展趋势.....	32
3.1 数据科学基础	32
3.2 数据科学方法	34
4. 国产数据库生态建设.....	37
4.1 数据库评测基准建设	37
4.2 数据库开源生态建设	39
4.3 产学研用深度合作	40
4.4 小结	41
5. 总结.....	41
参考文献.....	42

摘要

在数据库六十年来发展过程中，数据库系统和技术得到了蓬勃发展。本文主要讨论数据库系统的发展趋势以及数据库技术的前沿热点。本文首先计算模式的改变和应用需求变化对数据库系统形态起到了至关重要的作用，也推动了数据库架构的迭代更新。(1) 线下数据库到云数据库的转变：数据库部署形态从传统的线下部署转变到云上部署，而为了提升云上数据库竞争力，基于计算存储分离的云原生数据库得到了广泛关注。(2) 集群数据库到分布式数据库的转变：随着大数据时代的到来，数据库从单机主备、集群模式，逐步发展到分布式数据库模型，而分布式查询优化、分布式事务和分布式一致性协议等技术也得到了快速发展。(3) 端边云协同数据库：随着移动互联网和数字孪生的普及，端边云协同的数据处理得到了广泛关注，而异构智能设备上的分布式数据协同与一致性管理需要深入研究与发展。(4) AI 原生数据库：随着智能时代到来，AI 和数据库的结合也成为了研究热点，一方面通过 AI 技术优化数据库的设计和管理（例如学习型索引、学习型代价估计、智能参数调优等），另外一方面通过数据库系统和技术降低 AI 使用门槛，普惠 AI。本文从云原生数据库、分布式数据库、端边云协同数据库、AI 原生数据库四个方面讨论数据库系统研究进展。其次，数据库技术近年来也得到了迅速发展，主要包括基于新型硬件的数据管理、智能数据管理、多模数据管理、数据安全和隐私保护，本文详细讨论这些新技术的未来发展趋势。第三，在大数据时代，数据科学也得到了迅猛发展，本文也详细讨论了数据科学的未来发展趋势和研究热点。最后本文还讨论了国产数据库生态建设和“产学研用”深入合作等问题。

Abstract

In the last six decades, database systems have been widely developed and deployed. This paper discusses the research challenges and trend of database systems and techniques. Firstly, the evolution of computing architecture and application requirements have played a vital role in database systems. (1) From on-premise databases to cloud databases: In order to improve the elasticity and competitiveness of cloud databases, the cloud native databases based on the disaggregation of computing and storage have received extensive attention. (2) From cluster databases to distributed databases: the database architecture has evolved from cluster databases to distribute databases to meet the requirement of big data era. (3) End-edge-cloud collaborative database: with the popularity of mobile Internet and digital twin, end-edge-cloud collaborative databases have attracted extensive attention to address the challenge of distributed data synchronization and consistency management. (4) AI native database: the combination of AI and database has also become a research hotspot to meet the requirement of AI era. On one hand, AI techniques can be integrated into databases to provide self-configuring, self-optimizing, self-monitoring, self-diagnosis, self-healing, self-assembling, and self-security capabilities. On the other hand, databases can be extended to provide AI capabilities using declarative languages in order to lower the barrier of using AI. This paper discusses the research progress of database systems from four aspects: cloud native database, distributed database, end-edge-cloud collaborative collaborative database and AI native database. Secondly, database techniques have also developed rapidly in recent years, mainly including data management techniques on new hardware (e.g., NUMA, GPU, NUMA, RDMA), learning-based data management, multi-mode data management, data security and privacy protection. This paper discusses the future development trend of these new techniques. Thirdly, in the era of big data, data science has also developed rapidly, and this paper also discusses the future development trend and research challenges of data science. Finally, this paper discusses how to construct database echosystems and how to promote the in-depth cooperation among industry, university, and research.

1 前言

数据库是计算机软件领域皇冠上的明珠，它可以向下发挥硬件算力，向上支撑各类应用，在软硬件栈中起到了承上启下的作用。自从 1964 年数据库概念被提出，经过 60 年来的蓬勃发展，数据库已经被广泛应用到各行各业，成为了 IT 领域不可或缺的基础软件。

1.1 新型数据库系统

在数据库六十年来发展过程中，计算模式的改变和应用需求变化对数据库系统形态起到了至关重要的作用，也推动了数据库架构的迭代更新。

- **新型计算模式：**新型计算模式，例如云计算架构，对数据库发展起到了重要作用。例如传统数据库为了实现高性能，采用的都是计算和存储紧耦合架构。然而云架构更希望计算存储解耦来发挥云计算弹性伸缩的优势，实现独立的计算弹性伸缩和存储的自动扩缩容，从而提升数据库的性价比。因此云原生数据库应用而生，旨在考虑云架构的特点设计云原生的数据库架构；随着物联网的发展和数字孪生的普及，端边云成为了未来发展的趋势，需要突破端边云数据处理技术来支持万物互联时代的数据管理，而目前还缺少端边云（协同）数据库。
- **应用需求变化：**数据库最初主要解决数据交易问题，因此 OLTP 数据库得到了广泛关注。随着 IT 系统的发展和业务创新需求，越来越多企业需要通过数据分析来支持商业决策，因此 OLAP 系统得到了蓬勃发展。随着智能时代的发展，越来越多应用需要融合数据库技术和 AI 技术来支持实时的智能决策，因此亟待研究 AI 原生的数据库系统来提升数据库的智能化水平。此外，随着大数据时代的到来，更多的数据管理系统更加注重系统的扩展性，因此 NewSQL 分布式数据库，例如 Spanner[1]，得到了大型应用的青睐，而分布式数据库和 AI 原生数据库还需要进一步完善和演进。

1.2 新型数据库技术

新型硬件、新型交叉学科技术、新型数据模型、安全隐私等因素对数据库技术的发展起到了至关重要的作用，产生了很多的新型数据处理技术。

- **新型硬件：**随着新型硬件的发展，例如内存、Flash、NVM（非易失存储），数据库技术

也得到了蓬勃发展。例如内存数据库对并发事务处理机制、索引结构、异常恢复机制都与传统数据库有很大的不同，NVM 数据库[2]的事务管理机制、页面设计机制也都与传统数据库大相径庭，基于 RDMA 的分布式查询优化[3]也提升了分布式处理性能。因此需要研究面向新硬件的数据处理技术来提升数据库的竞争力。

- **新型数据模型：**数据库经历了层次、网状、关系、对象、KV、文档、图等数据模型的发展，而当前多模数据管理得到了广泛关注，需要研究多模新型数据管理技术。
- **数据库系统的智能化：**大数据时代，大数据系统推动了数据库的发展，例如 Hadoop[4]和 HIVE[5]。AI 技术也推动了数据库智能化的演进和发展。
- **数据安全隐患计算：**数据安全隐患成为了当今关注的热点，如何保证数据的安全和隐私成为了数据管理系统的核心诉求。

本文将围绕这些对数据库发展和演进起到至关重要的因素展开分析和讨论，并给出未来发展的挑战和趋势。

1.3 数据科学

数据科学是利用科学方法、流程、算法和系统从数据中提取数据价值的跨学科领域。然而目前数据科学还处于探索阶段，还缺少数据科学基础理论、关键技术和系统的支撑。本文将探讨数据科学的挑战性问题 and 未来发展趋势。

1.4 数据库生态建设

由于数据库设计复杂和开发周期长，商业数据库通常被国外公司所垄断，因此数据库也成为了“卡脖子”的系统。近年来，为解决数据库卡脖子的问题，国产数据库应运而生。然而解决“卡脖子”问题，除了数据库核心技术，还需要构筑数据库生态，让数据库学习者、开发者、运维者、使用者加入到国产数据库建设当中，因此本文还将深入讨论如何加强数据库生态建设。

2. 数据库发展趋势与挑战

2.1 云原生数据库

云数据库现状：随着云基础设施的逐渐成熟以及企业用户的需求（软硬件维护成本降低）推动，云数据库近十年也得到了蓬勃发展。目前云数据库分为数据库云服务和云原生数据库两类。数据库云服务（**Database as a service**）主要是将传统数据库部署到云基础设施上（虚拟机），实现轻松部署、开箱即用、自动运维（备份恢复、软件升级、扩容、高可用部署、安全管理等由云服务商提供）。其优势就是数据库用户不用关心数据库的安装和维护，而云厂商提供了这些服务。而云原生数据库（**Cloud-native database**）则是为云架构而原生设计的数据库。目前的云原生数据库我们称之为云原生数据库 1.0，它通过计算存储分离、日志即数据、一写多读等技术实现的云数据库[6]。其优点（1）计算和存储的解耦和分离，实现计算节点无需保存数据库的状态（例如元数据），从而实现独立的计算节点弹性伸缩和存储节点弹性扩缩容，进而提升了数据库性价比；（2）日志即数据，为了支持快速的异常恢复，传统的数据库不仅写日志（redo log、undo log）还写数据，导致了读写 IO 放大。云原生数据库只写日志（redo log），通过存储层回放数据来避免 IO 放大，降低了云基础设施的网络压力（云基础设施节点度多，计算和存储之间网络跳数多，网络压力大）；（3）一写多读，存储层回放的数据能够保证多份数据的一致性（通过 Paxos[7]、Raft[8]、Quorum[9]协议），从而可以实现一写多读，并且支持读节点一致性读。

云原生数据库 1.0 存在的限制：然而云原生数据库 1.0 还存在一些限制。首先，只支持一写多读，不能实现多节点写，从而造成写扩展性受限，特别是不能支持写需求大的应用。其次，一写多读的可用性受限，当写节点故障时，需要切换到备节点，切换过程将导致数秒的业务恢复时间（**Recovery Time Objective --RTO**）。第三，一写多读的读节点读到的数据和写节点有时延，造成读节点可能读不到最新数据，从而使用场景受限。为了解决云原生数据库的写扩展性不足、可用性受限、读节点延迟读等局限性，急需研究具备多写能力的云原生数据库 2.0。

云原生数据库 2.0 的设计思路和发展趋势：实现云原生数据库多写的主要难点在于最新数据都在写节点的内存 buffer 中，备节点不论读取磁盘的数据页面还是日志回放数据页面，都得不到最新数据，从而造成了节点之间难以获取其他节点内存的最新数据。为了解决多写能力，

主要需要解决内存 **buffer** 数据的共享问题，即各计算节点对等，且可以访问到其他节点的 **buffer** 数据才能解决一份数据多写冲突问题。为了实现这一目的，需要研究分布式共享内存技术，将内存虚拟池化，节点间可以互相访问内存从而实现多节点写。随着网络技术的发展，**RDMA** 技术可以使得远程访问的时延降低到微秒级，从而可以实现高效的分布式共享内存。因此云原生数据库 2.0 将计算、内存、存储分层池化，实现三者的解耦，从而可以实现独立伸缩，进一步提升性价比。

云原生数据库 2.0 的技术挑战：然而云原生数据库 2.0 还存在着众多挑战性问题：

- 分布式共享内存技术：将分布式内存虚拟化，并提供近端（本地）、远端感知的内存访问能力，在保证高可用的同时降低访问时延；
- 计算、内存、存储分层解耦的事务处理架构：利用三层池化资源设计高效的事务处理技术实现分层解耦的数据库。
- 网络层和存储层算子下推：由于计算时跨网络、跨存储，节点间数据将产生多跳（网络间和存储层）传输，可能造成大量的冗余数据传输，因此需要研究存储层算子下推技术和网络层数据卸载技术，避免大批量数据传输。
- 细粒度弹性按需计算：设计细粒度弹性按需计算机制[10]（serverless），实现查询级、事务级、算子级的弹性按需计算，真正实现数据库的按需收费。此外还需要研究多租户之间的资源隔离和资源复用。
- HTAP[11]混合负载：云资源是按需收费的，因此如果预算充足，可以充分调用更多的资源，从而可以利用云资源来实现真正的 HTAP，例如通过自动添加内存节点，实现实时的行转列，从而利用列存来支持复杂分析查询（列存不持久化），从而可以支持高效的 HTAP。其研究难点在于如何实现高效的行转列以及如何智能选择哪些数据进行行转列。

2.2 分布式数据库

分布式数据库产生的背景：随着信息社会的不断发展，数据库承载的数据量和业务量不断的增加，数据库架构和部署模式也经历了单机工作站、集中式、分布式几个阶段的发展和演变。我国各行业的信息系统建设大多始于二十世纪八九十年代，大多数行业的信息系统初始采用烟囱式建设，大部分数据库系统是以工作站的形式存在。随着网络技术的发展，企业信息化进入了快速发展期，数据库的访问方式也逐渐从工作站模式向 Client/Server 模式转变，数据库承担的数据量和访问量不断增加。2000 年到 2010 年，随着各行业信息化、数字化转型

的加速，为了解决烟囱式建设导致数据孤岛、数据割裂的问题，企业信息系统由烟囱式向集中式演进，通过统一的数据库管理系统实现“数据大集中”，从而达到数据统一存储、统一管理、统一访问的目的。这一时期涌现一批如 DB2[12]、Oracle[13]、SQL Server[14]等比较成功的企业级数据库产品。同时，数据库系统对硬件的处理能力也提出了更高的要求，处理性能越来越强大的大机或 IOE 架构占据了主流，大型机占领了核心业务，小型机占领了非核心业务。

2010 年后，随着“互联网+”不断深入，电子商务的迅速发展引发全民购物狂欢浪潮，互联网金融让日常支付更加便捷，用户形成了从现金付款到数字货币结算的消费习惯，电子政务也让人民的生活越来越便捷。互联网业务的发展为数据库的带来了诸多挑战：

- **海量数据存储：**传统数据库容量大多在百 GB 至 TB 级别，伴随着互联网业务量的增加，数据库所需存储的数据容量也在急剧增长且数量呈现指数级增长，从 TB 级别增加到 PB 级别，未来很快就会增加至 EB 级别。
- **海量并发访问：**数据库的业务访问量也发生了量到质的变化，传统的业务模式业务访问仅局限于企业内部，通常数百至数千并发即可满足。而在互联网模式下，数据库的访问服务于海量的互联网终端用户，需要万级至百万级的并发支持能力。
- **敏捷的弹性伸缩能力：**互联网业务快速增长变化的特点，以及行业新的应用场景的产生，例如双 11、618 等全民狂欢购物节、在线支付移动支付交易业务、在线教育等，要求数据库具备弹性伸缩能力，以应对敏捷的业务发展与变化。
- **基础设施的分布式：**为了提升数据库的高可用，各行业在加速信息化基础设施的建设，从传统的两地三中心向分布式多地多中心变化，要求数据库从传统的可用区（AZ）内部署向跨 AZ、跨地域（region）的分布式部署架构演进。

综上，传统集中式数据库架构在新型互联网业务模式下面临诸多挑战，传统集中式数据库难以满足业务需求。解决这些问题本质是要解决数据库扩展性问题，如何让数据库部署运行在多台服务器上，新型的分布式数据库架构应运而生。

分布式数据库发展现状与挑战：早期研究人员对分布式数据库的探索，推动了 NoSQL 数据库于 2005 年以来的大发展。2012—2013 年，业界在谷歌(Google)发表的 Spanner 和 F1 系统的论文中看到了关系模型和 NoSQL 的扩展性在一个大规模生产系统上融合的可能性，开启了分布式关系型数据库发展的崭新时代。伴随着互联网业务、行业数字化转型的加速，近年来分布式数据库如雨后春笋般得到快速发展，当前分布式数据库的发现路线主要包括如下几种：

- **一主多备、读写分离方案:** 在单机数据库的基础上, 扩展到一主多备架构, 主机提供读写服务, 备机提供只读服务, 实现读写分离、一写多读, 该架构可以实现一定程度的读扩展, 但无法实现写负载的扩展。同时, 如何减少主备机之间的读时延也是需要解决的难点之一。
- **分库分表中间件方案:** 这种架构大多起源于互联网行业, 在单机数据库上层搭建分库分表中间件, 实现整个系统的高扩展。数据按照业务模块或者某个维度进行垂直切分, 中间件层负责存储分库分表元数据信息以及 SQL 接入路由转发。互联网场景数据模型相对简单, 通常可以按照模块或用户维度完美 sharding 拆分 (即没有跨 shard 之间的事务), 并且模块或用户之间没有数据耦合关系, 使得分库分表中间件方案迅速的推广和发展。分库分表中间件方案在跨节点分布式事务、分布式复杂查询方面能力不足, 使得在政企客户、尤其是难以完美 sharding 场景下推广难度较大。
- **分布式数据库方案:** 为继承集中式数据库 ACID 特性以及丰富的企业级 SQL 能力, 同时在扩展性方面实现突破, 分布式数据库应运而生。原生分布式数据库的架构设计与中间件方案有本质区别。原生分布式数据库是基于完整数据库理论架构设计和 SQL 引擎、执行引擎、存储引擎分层架构设计开发。可实现分布式 ACID、分布式事务与分布式查询的支持。分布式数据库对上层应用屏蔽底层实现细节, 可较容易实现传统集中式数据库向分布式数据库的迁移与改造。在海量数据存储、大并发支持能力、弹性伸缩等方面已实现比集中式数据库更大的突破。

分布式数据库要解决的核心问题: 随着分布式数据库应用与研究的深入, 分布式数据库的发展与演进也面临一些技术挑战, 分布式数据库的大规模商用, 还需解决如下核心问题:

(1) 分布式事务处理与查询优化: 在数据模型能够完美 sharding 的情况下, 不会产生分布式事务, 此时分布式数据库性能、扩展性具备较高的水平。在数据模型无法完美 sharding 的情况下, 会产生跨节点事务处理, 性能会急剧下降。提升分布式事务处理性能需要解决如下关键技术:

- **分布式事务处理:** 为保证分布式事务的强一致性, 分布式数据库需要全局单调递增的序列, 以确保并发事务能够在全局有序的前提下进行。目前业界分布式数据库全局单调递增序列的实现机制主要有两种: 集中式授时和分布式授时。
 - **集中式授时:** 单点生成单调递增的序列号, 用以标记事务的先后顺序, 保障事务并发场景下的隔离性。集中式授时方案实现简洁, 全局时钟严格一致, 但授时节点容易成为中心瓶颈点, 系统的扩展性差。另外, 集中式授时方案在分布式数据库跨可

用区部署场景下，存在跨可用区访问授时节点时延过高的问题。

■ **分布式授时：**为解决集中授时存在的单点瓶颈和跨可用区高时延问题，业界积极研究构建分布式授时方案，如谷歌 Spanner 即采用“本地原子钟+GPS”的分布式授时方案，支撑分布式数据库在跨可用区、跨地域部署场景下的数据一致性方案。

- **分布式优化器：**复杂查询在分布式数据库使用过程之中不可避免，大多数中间件分库分表中间件产品不能支持跨节点的复杂查询。原生分布式数据库要求具备强大的分布式优化器能力，能够处理复杂查询的解析、并生成最优分布式执行计划。分布式数据库的复杂查询优化的核心方法是减少 I/O，包括磁盘 I/O 和网络 I/O，减少 I/O 同时也减少内存开销。当前主流技术是基于代价的优化器（CBO），基于 AI 的智能代价评估与计划选择的智能优化器是未来发展趋势。

(2) 智能数据分布技术：分布式数据库由于需要将数据按照一定规则存储到多个节点上，通常在创建表时需要指定分布键，分布键的选择合理与否直接影响到数据库的性能指标。因此如何高效选择分布键是分布式数据库是否能做到应用透明的关键技术之一。目前业界主要有两种方法：

- **基于 AI 的智能分布键推荐技术：**利用 AI 算法对数据模型和数据分布特性进行学习，帮助用户更加高效的选择最合理的分布键，从而将分布式数据库性能发挥到最优水平。
- **透明分布键技术：**在数据库建表时，自动创建默认的隐藏分布键，整个过程用户无感知。此类技术需要解决应用场景的通用性问题。

(3) 分布式高可用技术：分布式数据库要解决在通用硬件基础平台上，利用数据冗余技术，结合分布式数据库一致性协议算法等数据冗余相关技术，实现分布式数据副本间的一致性，同时也要保证分布式事务一致性。分布式数据库高可用应具备多层次的技术要求，以满足不同级别的故障高可用方案，当数据库系统发生节点级故障时，RTO 要求在秒级，宜采用自动切换方式；跨可用区、跨地域高可用需要满足相关行业标准要求。而研究难点在于如何保证高性能的基础上能够支持跨地域多活（全球多活）。

(4) 智能运维调优技术：相对传统集中数据库分布式数据库部署规模、架构复杂度更大，随着分布式数据库的应用与推广，运维管理面临更大的挑战，因此如何利用 AI 技术实现分布式数据库智能运维、智能调优、智能诊断、智能索引/视图推荐是分布式数据库需要解决的主要问题之一。

总结与展望：近年来，随着金融、政企行业客户大规模推进数字化转型与分布式改造，原生分布式数据库架构产品得到快速的发展，主流 TOP 行业客户已经开始探索基于分布式数据

库实现集中式数据库的迁移改造。学术界有大量的学者在分布式数据库理论算法与架构方面进行深入研究，产业界也在加大对分布式数据库的研发投入。同时，异构处理器、新型存储介质、高性能网络等新型硬件技术的出现，也会加速分布式数据库的发展。相信，随着分布式数据库的技术发展与突破，分布式数据库会是未来政企客户应对互联网业务发展、企业数字化转型的必然选择。

2.3 端边云数据库

端边云数据库背景和现状：随着物联网的发展和数字孪生技术的普及，端边云协同计算成为了未来发展的趋势。例如大型商场安装很多端侧的监控设备，例如摄像头、水电管道监控、烟雾传感器等设备，这些设备实时采集多源异构信息，我们需要依赖端边云协同技术来使用这些信息实现智能的实时决策。

首先，端侧（例如传感器、摄像头）实时采集信息，需要时序处理技术来获取、存储和分析各个端侧数据，实现数据采集和简单数据端侧实时分析。

其次，边侧（大楼的机房和数据管理服务）需要实时汇聚、关联分析这些数据，实现近数据处理。对于计算力要求低的需求，可以通过边缘侧算力来实现实时决策。而对于计算力要求高的需求，可以通过将数据传输到云端，通过云端进行计算分析来实现复杂数据分析。

第三，云侧主要实现数据的存储、复杂分析和决策支持。

端边云协同计算中的一个重要问题是为什么不直接用云计算，而引入端侧和边侧的意义是什么？其主要原因包括：

- 端侧、边侧实时决策：将数据传输到云端再做决策的时延很多场景不能满足实时要求，例如烟雾传感器数据的分析和预警，因此需要实现端边云协同智能分析。
- 网络传输优化：直接将端侧数据上传到云端可能导致带宽不足，占用大量网络带宽，因此端侧数据不需要全部传到云端，而通过两种方式可以消除网络拥塞：（1）数据聚集分析过滤：将分析聚合的数据、或者关键的数据传输到云端，例如传感器异常的数据；（2）数据的压缩（例如视频）：通过数据的压缩和编码变换来降低传输开销。
- 能耗优化：端侧和边侧往往电池容量受限、计算资源受限、存储资源受限，因此需要优化端侧和边侧的能耗，保证端侧设备可用时间。
- 数据隐私安全：很多数据具有隐私性，不能将全部数据传到云端，例如进入商场人员的摄像头数据，因此不能传输原始数据，而只传输抽取的特征数据。

目前还没有端边云协同计算的数据库来支持端边云计算的场景。因此需要研究端边云协同的数据库系统。

端边云数据库研究挑战：

- **端边云协同数据库系统：**综合考虑端侧能耗、端边云之间带宽、应用响应时延需求、端侧边侧计算资源配置，需要重新设计端边云协同的数据管理，解决端边云协同数据分析的 where-when-what 问题，即数据应该存在哪一侧、什么时机传输和分析数据、采用什么数据模型和分析手段？与传统数据库基于代价的分析不同，端边云协同数据库需要考虑能耗、带宽、时延等多个因素，设计新的代价优化模型和新的数据存储模型来实现端边云协同数据处理。
- **端边云异构硬件协同优化：**端边云之间的硬件各异，计算能力、存储能力、能耗都各不相同，传统的优化模型难以适应异构硬件，因此需要设计适配异构硬件的新型协同优化模型，考虑不同操作在不同设备上的计算代价，设计适应异构硬件的算子模型、算子实现机制和复杂查询优化理论，实现高效的端边云协同异构计算。
- **端边云协同数据编码压缩：**为了节省带宽需要在端侧和边侧进行数据编码压缩，降低数据传输。然而由于端侧数据类型复杂、数据源多样，需要研究端边云协同数据编码压缩技术，智能自适应地压缩不同类型数据，例如字典压缩、增量压缩等，一方面减少数据量，另外一方面支持快速分析处理（例如列存分析、聚集过滤）。
- **端边云数据近数据计算：**在端边云中实现近数据计算技术，通过移动模型（而非移动数据）实现实时数据分析。首先由于数据可能存储端、边、云不同位置，需要设计数据提前聚集和计算卸载技术实现分布式融合计算，提升计算速度；其次需要将算子下推到数据所在设备，减少冗余计算。
- **端边云数据的隐私保护：**数据在端边云协同计算时需要保证的数据安全和隐私，因此需要研究保护安全隐私的计算方法。

2.4 AI 原生数据库

AI 原生数据库研究背景。近年来，随着大数据、机器学习算法、新型硬件技术的发展，几乎所有行业对 AI 的需求都在快速增长。在很多行业（如互联网和医疗）中，相比于结构化数据，非结构化数据和半结构化数据不仅数量更多，而且其商业价值甚至可能超越传统的关系数据。比如，通过分析用户商品评价的文本信息，电子商务网站可以进行高效的客户关系管

理，并且可以预测客户的未来购买行为。综合利用各种异构数据进行分析预测通常需要借助复杂的机器学习算法来实现。但是 AI 算法门槛较高，依赖 AI 算法设计师和模型训练师来落地算法应用，从而造成 AI 落地难。因此急需利用数据库系统技术来降低 AI 使用门槛。另一方面，随着场景的日趋复杂化和管理的日趋精细化，DB 和 AI 的功能界限变得模糊，单纯的 DB 或 AI 的系统已经不能满足很多业务的需求，它们需要的是兼具 DB 和 AI 的混合分析功能。然而，目前的关系数据库无法提供完善的 DB 和 AI 混合功能，AI 原生数据库将是解决这个需求的关键技术。AI 原生数据库一方面扩展 SQL 的算子来支持 AI 算子，实现库内的训练和推理，另外一方面通过数据库内置 AI 算法来提升数据库的智能优化、智能运维。

AI 原生数据库研究现状。AI 原生数据库在数据模型、数据操作和系统部署三个层面都带来新的挑战。从数据层面来说，其需要综合处理大量异构多模态数据，如结构化、半结构化和非结构化数据，以及静态批数据和动态流数据等。从数据操作层面来说，AI 模型比传统的 SQL 模型要复杂得多比如深度学习模型通常都包含上千万甚至上亿的参数，而且随着训练数据的变化，它们需要被渐进地维护，因此如何管理 AI 模型和优化它们的操作是巨大的挑战。从系统部署层面来说，AI 运算通常需要新型硬件（如 GPU，NVM）进行加速，而数据库系统如何利用这些新型硬件仍在探索阶段；另外，AI 原生数据库需要可伸缩性的部署，可以根据数据量和业务需求增长进行轻松扩展。

根据数据库管理系统（DBMS）和 AI 融合程度，目前的解决方案可以分为以下三类：

- (1) DB 和 AI 的 Shared-nothing 方案，即 DBMS 系统和 AI 系统无共享机制，具体例子包括 SQLFlow[15]、SQL4ML[16]和 MLog[17]等。该方案使用 DBMS 存储数据，分别使用 SQL 引擎和 AI 引擎进行 SQL 和 AI 操作；AI 操作所需数据需要从 DBMS 中导出。为了方便用户使用，有些系统（如 SQLFlow）使用声明型语言（如扩展的 SQL）统一定义关系操作和 AI 操作，并自动地在 DBMS 和 AI 系统间实现数据传输。该类方案目前是产业界最经常采用的方案，可以直接利用已有的 DBMS 系统和 AI 系统（如 Pytorch[18]、Tensorflow[19]和河图[20]等），容易启动；但缺点也很明显，DB 和 AI 无融合，几乎没有优化，需要用户进行大量 AI 编程和手动的性能微调；
- (2) DB 和 AI 的 Shared-data 方案，很多文献里也被称为 In-database ML，具体例子包括 MADlib[21]、Vertica[22]、SQL Server[14]等。该方案通过用户自定义函数在 DBMS 内部实现 AI 操作，不需要从 DBMS 导出数据，但是 DB 和 AI 的操作仍是由不同的引擎分开处理。该类方案目前被很多主流的 DBMS 提供商所采用，优势是可以使用自定义函数灵活实现 AI，无需在 DB 和 AI 系统间传输数据，比 Shared-nothing 方案更加高

效；但其缺点是把 AI 功能仅作为数据库一个二等公民的插件，DB 和 AI 的功能融合程度低，优化少，AI 操作效率低；

- (3) DB 和 AI 的 Shared-everything 方案，即统一的 DB & AI 融合方案，具体例子包括 openGauss[23], RelationAI[24], SciDB[25]等。该方案使用统一的数据模型、数据操作模型与操作优化引擎来实现 DB & AI 的混合功能。该方案处于初始研究的阶段，学术界提出了很多原型系统，但产业界目前仍非常谨慎。主要的障碍在于，目前的方案是在关系型的 DBMS 之上加以简单扩展实现的，利用 SQL 引擎支持 AI 功能，能高效支持的 AI 功能很有限；虽然 SQL 语言理论上能支持几乎所有的 AI 操作，但是 SQL 引擎是针对传统的 SQL 查询设计优化的，并不能有效支撑复杂的 AI 操作。另外，目前的方案对非结构化和弱结构化数据的支持也非常有限。

AI 原生数据库核心挑战。现有的的解决方案都是从 DB 的角度来研究如何支持 AI 功能。然而，目前主流的 DBMS 都是为传统 SQL 操作设计和优化的，并不能高效地支持复杂的 AI 操作。因此，未来的 AI 原生数据库需要从 AI 需求的角度来重新设计和实现 DBMS，以友好的方式高效地支持 DB&AI 的混合处理（Hybrid DB & AI Processing），其核心技术包括：

- **如何设计和实现统一的数据模型。**混合的 DB & AI 处理需综合处理结构化、弱结构化和非结构化等不同类型的数据，它们的统一存储与表示是 AI 原生数据库面临的挑战之一。另一方面，AI 操作经常涉及标量、向量、矩阵以及高维的张量等数据，这些数据的存储与表示也直接关系到 AI 原生数据库的性能。因此，如何设计统一的数据模型来表示异构多模态数据，并实现相应的存储方法，使之能无缝地支持关系代数操作（如选择、投影、交）、线性代数操作（如标量、向量、张量操作）以及其他基于更复杂 AI 模型（如神经网络）的操作，是需要解决的核心挑战之一。
- **如何设计和实现统一的操作算子。**目前 DBMS 的核心算子基于关系代数，是为传统简单的 SQL 操作设计和优化。然而，AI 操作通常涉及基于线性代数以及更复杂 AI 模型（如神经网络）的运算。虽然理论上绝大部分的 AI 操作都可以通过目前的 SQL 算子实现，但需要非常多轮的迭代运算，效率非常低下。因此，如何定义和实现一套 AI 原生的数据操作算子，友好高效地支持 DB & AI 的混合运算，是 AI 原生数据库需要解决的另一个核心挑战。考虑到现代数据库大都部署到分布式的新硬件环境中，如何在这些环境中实现高可扩展的 DB & AI 混合操作尤其面临着严峻的挑战。
- **如何设计和实现统一的优化引擎。**在统一的数据模型和操作模型之上，AI 原生数据库需要一个统一的执行引擎，优化 DB & AI 的混合操作，包括混合查询代价估计、混合查

询优化以及 AI 模型的训练、更新和维护等。跟传统的 DBMS 类似，如何基于代价模型优化执行计划是需要解决的核心挑战之一。如何引入新的数据结构，如模型物化视图和模型索引等，以加速执行过程，也是值得研究的关键问题。另外，AI 操作经常涉及非常复杂的 AI 模型，这些模型需要渐进地训练和维护，AI 模型的管理以及其跟执行优化的耦合也是执行引擎的核心挑战之一。

- **如何利用 CPU+GPU 异构硬件实现训练和推理加速。**DB 和 AI 通常需要不同的计算能力和硬件，需要充分利用多样化的计算能力。对于数据库，传统的优化器使用 CPU 处理查询。而人工智能技术则需要新的 AI 芯片来支持并行处理(如 GPU, NPU)和自调度。现在许多应用程序都需要同时使用 DB 和 AI 技术，尤其是在大型数据分析场景中。因此，需要支持多个模型，如关系模型、图模型、流模型、矩阵模型、向量模型和张量模型。可以自动选择应该使用哪些模型。还需要能够根据数据模型切换计算能力。例如，对于优化器中的调优模块，在训练调优模型时，我们使用 AI 芯片将训练数据提取到内存中，然后使用 NPU 进行反向传播(训练神经网络)。对于连接和过滤操作，仍然可以使用传统的硬件。还需要研究一个模型(如关系模型)是否可以转化为其他模型(如张量模型)。最终目标是充分利用 x86、ARM、GPU、NPU、加速器等多种计算能力。

AI 原生数据库未来发展趋势。关系型的数据库管理系统在过去几十年中取得了巨大的成功。过去提出的一些替代模型，如面向对象和 XML 等，虽然从来没有取得类似的成功，但它们的一些理念和技术最终被融入到关系型 DBMS 的设计和实现中，推动了关系型 DBMS 的发展。近些年来，人工智能的崛起深刻改变了各个行业对数据管理的需求，体现在以下三个方面：(1) 对人工智能的需求不断增长，而且其增长速度已经大大超过传统 DB 数据分析的需求增长速度；(2) 非结构化数据和半结构化数据大量增长，不仅数量已经远超关系数据，而且其商业价值也将可能超过传统的关系数据；(3) 然而，目前的 DBMS 系统是为传统简单的 SQL 操作设计和实现的，并不能友好高效地支持更加复杂的 AI 操作。因此，我们需要从 AI 需求的角度重新思考和定义 DBMS，在数据模型、数据操作模型和执行优化引擎等层面全面推动数据管理的理论和实践创新。AI 原生的数据库作为满足人工智能时代数据管理需求的核心技术，将推动数据库管理系统进入 3.0 时代（注：1.0 时代指的是传统关系数据库，2.0 时代指的是云原生数据库）。

2.5 数据库新技术

2.5.1 新硬件驱动的数据管理

新硬件驱动的数据管理研究动机。数据库在基础硬件和上层软件之间起到了“承上启下”的作用，向下发挥硬件算力，向上支撑上层应用。底层硬件技术决定了数据存取、并发处理等处理性能的物理极限，而上层软件系统也需要通过优化数据库架构和算法的设计，以提高软硬件契合度、最大化硬件利用效率。传统数据管理系统基于经典硬件设计，其作为各类复杂应用的基础，在经济社会各领域起重要作用。近年来，各类高性能处理器、非易失存储器、支持 RDMA（Remote Direct Memory Access）的高速网络等新硬件技术取得了重大突破，给数据管理系统带来新的机遇与挑战。针对新硬件特点，数据库需要在架构模式与关键技术层面取得创新，以充分利用新硬件能力，满足新型应用在高性能、高可用、可扩展性等方面的需求。

新硬件驱动的数据管理研究现状。目前，新硬件驱动的数据管理与分析技术已成为学术界和工业界共同关注的热点。

- 在计算层面，众核高性能处理器和各类硬件加速器（例如 GPU、FPGA、AI 芯片等）广泛用于数据处理的加速。主要包括 CPU 指令级优化、多核并行优化、事务并发控制等[26,27]，协处理器查询加速、存储层计算卸载、数据压缩加速等[28,29]，硬件间的工作负载迁移[30]、硬件级并行性优化[31]、硬件级查询算子[32]等。数据管理与分析系统正在从单 CPU 架构向异构、混合处理架构逐步演变。
- 在存储层面，非易失、按字节存取的 NVM 的出现，使得内存和外存之间的界限变得模糊，对数据管理分析系统的存储层次结构和索引设计产生深刻影响。NVM 可以在不改变数据管理分析系统存储层次结构的情况下替换传统存储介质[33,34,35]或与其混合[36,37]，也可以作为 RAM 和磁盘之间的缓存层[38,39]丰富存储层次结构。同时，NVM 和磁盘有显著差异，存在字节级访问粒度、读写性能不对称、读写功耗不对称、写耐受性差等特征，针对传统块存储设计的索引在 NVM 存储环境中性能会受到一定影响，因此，出现了针对于 NVM 设计或优化的索引[40,41]。
- 在传输层面，支持 RDMA 的高性能网络极大改善了服务器间跨节点的数据访问性能，克服了分布式系统固有的网络瓶颈。一些数据管理系统[42, 43, 44]已经在其内部引入了 RDMA 网络，用于加速分布式查询与事务处理。目前针对 RDMA 相关数据管理技术

的研究多是对已有系统的改进,而研究 [45] 表明,为了充分利用 RDMA 高性能网络技术,需要对分布式数据管理系统的架构进行根本性的重新设计。

新硬件驱动的数据管理发展趋势。从数据管理软件的发展历史来看,近半个世纪以来,从存储介质层面,数据库经历了磁盘数据库、闪存数据库、内存数据库的发展轨迹,这反映了数据管理软件随底层存储介质变化的趋势。Shore-MT[46]、MapD[47]等基于众核处理器设计的系统的出现,也反映出数据管理技术随处理器共同演化的趋势。以高性能处理器和硬件加速器、NVM、RDMA 高性能网络为代表的新硬件技术,将改变传统的数据管理系统的底层载体支撑,数据管理系统将向异构计算架构、混合存储环境和高性能互连网络逐步演进。此类新硬件驱动的数据管理,已成为研究热点且取得一定研究进展,但仍存在以下未完全解决的问题:

- **基于 NVM 的新型存储和索引管理:** NVM 的非易失、字节级访问等特点为数据库架构带来了新的挑战和机会,传统的基于页面的存储机制可能不再适应新型 NVM 设备,在混合存储 (RAM、NVM、Disk) 下的存储和索引管理,是未来重要的研究方向。
- **CPU 和 GPU 混合异构计算:** 查询处理是数据管理中的核心操作,异构计算架构 (CPU+GPU) 提供的高度并行性和可定制能力使得现有的查询处理和优化机制难以应用。研究异构环境下的协同查询处理技术、查询优化技术、混合查询执行计划生成技术等具有重要意义,特别需要研究数据处理操作在 CPU 和 GPU 之间的自动调度以发挥芯片最大算力。
- **基于 RDMA 的分布式优化和调度:** RDMA 可以通过单边读写卸载 CPU 压力,需要研究基于 RDMA 的分布式优化技术,包括 CPU 的卸载、可编程网卡的 IO 卸载,以及综合考虑网络 IO 和磁盘 IO 的分布式优化模型。
- **面向数据管理的专用硬件:** 数据处理专用芯片能够进一步提升数据处理速度,降低数据处理能耗,由于高成本和将新硬件集成到软件基础设施中的长周期,数据处理专用芯片尚未得到广泛应用。然而,在过去十年中,计算领域已经发生重大变化,构建专用硬件架构的成本急剧下降,以深度学习专用硬件为代表的新硬件不断涌现,这也为数据处理专用硬件的发展和普及创造了一定的条件。

2.5.2 智能化数据管理

智能数据库系统是指数据库系统具有智能化,借鉴人工智能技术,可以实现包括数据库

的自优化、自管理、自监控、自诊断、自恢复等在内的多维度等高度自治功能。从功能角度看，智能数据库系统也称为自治数据库系统；从技术角度看，智能数据库系统也可被称为 **AI** 赋能的数据库系统。

智能化数据管理研究动机。随着云时代的到来，云数据库得到了蓬勃发展。但是云数据库带来的一大挑战就是如何在云平台上支持百万级的数据库实例、千万级的数据库用户？传统的数据库通过数据库管理员来实现数据库的调优和运维管理，这种方法需要大量的人力，显然不适合具有海量用户和数据库实例的云数据库。近年来，随着 **AI**（人工智能）技术的发展，学术界和工业界开始探索利用人工智能方法解决数据库管理、优化、运维等问题，因此 **AI** 赋能的智能数据库系统应运而生。

智能化数据管理研究现状。目前学术界和工业共识的研究重点是把机器学习与数据管理在功能上融合统一，利用机器学习增强系统设计开发。一方面工业界以甲骨文公司为代表的自治数据库系统（**Autonomous Database System**）和华为的 **AI** 原生数据库 **openGauss** 开发，另外一方面学术界各个研究团体在自优化、自监控、自诊断、自恢复等方面取得了一定研究成果。

- **自优化：**包括数据库内核的智能优化（包括智能查询优化器、智能代价估计、智能计划选择）以及数据库外置的根据用户提交的查询来自动实现索引、视图、参数推荐。内核智能优化包括基于强化学习的连接顺序选择器、基于深度神经网络的代价估计器，以及智能查询重写器等。
- **自监控：**根据数据库运行过程中的各项指标（例如 **CPU** 使用率、响应时间、运行日志），监控数据库运行状态，并可以通过强化学习等人工智能技术，实时检测系统异常，并可以调整系统参数，提升数据库运行性能。
- **自诊断：**主要根据数据库性能表现以及运行指标，抽取特征，利用人工智能技术发现数据库系统异常的根本原因（例如锁冲突、网络异常等）以及慢 **SQL** 诊断（例如缺失索引），并提供解决数据库故障的决策手段。
- **自恢复：**主要是利用历史诊断手段来自动避免错误和问题，自动恢复数据库，保证数据库的高可用性。

可以看出，设计一款原生基于机器学习实现自优化、自监控、自诊断、自恢复的自治数据库系统已经成为可能。

智能化数据管理发展趋势。虽然有越来越多的学习型数据库算法被提出并应用到数据库系统中，但仍然存在一些问题有待进一步攻克：

- **端到端的自治数据库系统：**决定智能数据管理性能的主要因素来自三个方面:其一是数据库管理对象的数据特点与分布;其二是数据库系统的任务需求;其三是数据库管理系统本身的优化策略。现有的方法仍然停留在针对系统各个环节的局部优化层面，缺乏对数据库系统的整体感知的动机和手段，缺乏统筹考虑，无法达到系统整体优化的目标。而未来的智能数据库系统需要提供整体优化能力，将数据库系统作为一个运行整体，智能监控系统内部数据和感知系统外部数据(往往是一个开放域,传统的方法不一定适用)，并能快速筛选数据，完成系统的整体优化功能。
- **鲁棒性的智能优化方法：**系统负载稳定性是自治数据管理性能的保证，目前的机器学习赋能方法只追求系统的最优指标，例如最高 **TPS** 和 **QPS**、最低时延，但没有考虑系统的鲁棒性，例如是否可以持续保持稳定的性能、是否可以支持多变场景。而实际的智能数据库系统需要系统保持运行的稳定性，即构建的索引、查询优化器、负载预测效果能在一段时间间隔内确保系统持续保证较高的性能并避免较大的波动。
- **轻量级的机器学习算法：**查询性能是数据管理的核心，机器/深度学习很难保证这种映射的高效性，尤其是很多深度学习网络往往具有较深的网络层次(例如几十甚至上百层的结构)，网络参数的规模庞大(几千到上亿)，无论是训练还是适用这样的模型都无法满足数据库系统查询处理的要求。而且，数据库中存储的数据因用户的更新操作而实时变化，希望学习的映射模型可以根据数据库状态完成增量学习。这些因素都导致大部分机器/深度学习无法媲美传统的基于统计和启发式的策略。如何提供时间和空间上小巧而轻量的学习模型是 **AI** 赋能的查询优化技术的难点和关键问题之一。
- **智能优化的一致性保证：**一致性是查询处理正确性的保证，例如，对于关系 R 上的两个查询 $\sigma_{a \leq l}(R)$ 和 $\sigma_{a \leq u}(R)$ ，如果 $l \leq u$ ，则相应的映射函数 $f(\cdot)$ 需要满足 $f(l) \leq f(u)$ 。数据库系统在很多场景下要求严格保证映射的一致性，而机器/深度学习是基于训练样例得到的模型，大部分深度学习模型不能保证映射的一致性。类似的，查询优化器中如何设计机器学习模型满足基数估算的一致性，智能索引构建中如何在理论上保证学习型索引的准确性都要求学习模型能满足映射一致性约束。需要相应的学习算法中理论上可以保证系统的一致性。
- **可迁移的智能优化方法：**数据库数据和查询都是动态变化的，而传统的机器学习优化方法一般都是假设数据模式 (**schema**) 不变、数据不变、查询不变，显然不适用数据库中频繁变化的场景,例如一个训练好的系统如何迁移到一个新的数据库业务而保持较好的性能。

- **自组装数据库。**每个数据库组件都有多个实现方式，例如，优化器包括基于代价的模型（cost-based）、基于规则的模型（rule-based）和基于学习的模型（learning-based）。可以根据用户的需求来选择最好的组件。其中，相同组件的不同变体应该采用相同的标准接口，以便可以组装组件。对于不同的场景，可以在每个服务层中动态地选择适当的组件，并组合适当的执行路径。因此需要研究针对不同场景的最优的组装路径，从而可以做到场景自适应的数据库。

2.5.3 多模态数据管理

多模态数据管理研究动机。多模态数据管理即是针对多种模态的数据进行统一管理。多模态数据主要包括结构化数据、半结构化数据和非结构化数据，例如关系，键值，图，XML/JSON 文档等。传统的数据库都是针对单个特定模态数据管理设计与实现的。目前很多领域都呈现多模态数据统一管理的重大需求，特别是制造业领域。制造业产生了越来越多的数据，根据麦肯锡全球研究院发布的研究报告表明，美国制造领域拥有的数据规模为各领域之首。大数据已成为制造业智能化的核心要素，而针对制造业大数据管理的主要挑战在于多模态数据管理。目前，多模态数据管理已成为世界前沿科技热点。

多模态数据管理研究现状。首先，各个模态的数据都有其特定的建模方法，需要找到一个模型能够对多模态数据统一建模。目前的主要做法是扩展某个特定模态数据的模型使之能够定义其他模态的数据。其次，多模态数据存储目前的主要功能在于如何协调不同模态数据源之间的数据重复性和互补性特征，以利用这些特征帮助用户构建完整视图，提高数据读取效率。第三，目前在多模态数据查询领域中，最具有代表性的查询功能是增大查询(Augmentation Query)功能。多模态数据查询优化目前比较有代表性的方案包括基于索引的优化、基于查询代价的优化和基于机器学习的优化方法。目前的多模态数据管理系统很少考虑到对事务处理问题的研究。其原因在于目前的多模态数据管理系统主要功能在于为用户提供一个面向底层引擎的查询接口，而并没有向用户提供对底层数据的修改功能。针对不同行业应用领域，目前已有许多多模态数据管理系统，在各行各业有着广泛的应用。

多模态数据管理研究挑战。多模数据管理在统一建模、统一存储、查询优化、并发控制等方面还存在着众多挑战性问题。

- **多模态数据统一建模：**基于现有的关系、图、键值或 JSON 文档模型进行扩展，难以将现有这些模态数据都融合起来，各种模态数据之间彼此模式映射会引起组合爆炸，模式

演化也变得更加复杂。

- **多模态数据存储管理：**集中存储难以支持海量多模态数据，也缺乏扩展性；分布存储带来数据读取效率低下，而现有的聚簇机制和缓存机制都没有充分考虑多模态数据的关联性和粒度性。
- **多模态数据查询处理：**现有语言只是针对某几个特定模型设计，缺乏多模态数据查询的“世界语”，关联查询、溯源查询和预测查询能力有限，查询优化没有充分利用多模态数据的丰富语义关系。
- **多模态数据并发控制：**由于各种模态数据之间语义关联复杂，针对单一模态设计的分布式事务机制难以适用，需要增强各种模态数据的更新感知、基于语义关系的更新迁移和可保证的更新一致。
- **多模态数据质量保证：**由于不同模式的数据实例所表示的信息可能存在差异或冗余，在集成和转换多模态数据导致低质量数据，针对上层任务进行高质量数据清洗和集成也面临着挑战。

多模态数据管理发展趋势。多模态数据建模将由目前的以关系为中心逐步转为以对象为中心，更多地关注非结构化数据的语义建模，强调数据与操作的可封装性、多重分类和动态分类。特别是多模态数据之间复杂的语义关联，需要有相应的元数据管理技术，将其抽象成数据库的模式。然后，通过模式所表达的多模态数据内在联系进行数据组织和管理。多模态数据动态管理要求模式可动态演化，未来不是模式限定数据变化，而是数据驱动模式演化。

多模态数据管理将由目前的各种模态数据存储引擎协同工作转换为单一功能强大的存储引擎集中高效管理。多模态数据的分散存储将制约多模态数据管理性能的提升。将多模态数据按照其统一模式和语义关联进行集中存取能够避免模式转换和网络通讯所带来的额外开销，可以将多模态数据管理有效地应用到未来自动驾驶等实时场景。

多模态数据查询将由目前的 SQL 语言表达的集合运算发展到自然语言处理的关联计算。单一模态数据查询主要基于关系代数进行集合运算获得结果，多模态数据查询更多基于语义关联进行关联计算推导知识。SQL 语言表达能力不足以表达复杂的多模态数据关联查询需求，使用自然语言将成为必然发展趋势。自然语言处理技术和基于知识图谱推导技术将使得未来多模态数据查询更加智能。

2.5.4 数据安全隐私计算

近年来随着大众对保护个人信息意识的提升,以及各国家地区纷纷出台数据安全相关的政策法规,例如中国的《中华人民共和国数据安全法》与欧盟的《通用数据保护条例》(General Data Protection Regulation, GDPR)等。如何保护数据安全隐私的问题受到重大的关注,同时也是工业界与学术界研究的热点问题。

本节将从数据管理流程中的数据存储、数据计算、数据维护、数据采集与发布这四个步骤出发,简要介绍各步骤中涉及的数据安全隐私计算的研究方向,分别为:全密态数据处理、数据安全多方计算、防篡改数据处理、数据隐私计算四个研究方向。

2.5.4.1 全密态数据处理

全密态数据库研究动机。全密态数据处理通常是指对密态数据库(Encrypted Database)中的数据进行的操作,其重点关注如何对数据进行加密存储以便在加密后的数据上进行多种类型的查询。追溯历史,该研究主题源于一种新的计算范式“数据库即服务”(Database as a Service)[48]的提出。在这种计算范式中,用户通过向服务提供商购买数据管理服务来减少存储与开发的成本并提高数据访问的灵活性。然而将数据外包至服务提供商的云服务器后,会导致数据不再完全受控于数据拥有者,带来数据隐私泄露的风险[49]。为了规避隐私泄露风险,加密外包数据是保障数据安全的有效手段。但加密后的数据失去了原数据的一些特征如数值大小、格式等信息,导致数据提供商所能提供的云服务受到限制。为了解决该问题,密态数据处理利用全同态加密[50]等技术对数据进行加密的存储以实现在保证数据安全的情况下尽可能地提高云服务处理加密数据的能力,使尽可能多的操作由服务提供商完成。

研究现状。密态数据处理的相关工作在数据维度上从简单的一维数据发展到多维数据[51,52],在数据类型上从关系型数据发展到丰富的时空数据[53]、半结构化数据[54]、图数据[55]等多种数据类型并支持对应的查询操作,进一步形成密态数据处理系统。密态数据处理系统根据加密方式不同主要分为基于软件加密和基于硬件加密两大类。基于软件加密的系统通过特定加密算法来实现数据的加密存储主要包括 CryptDB[56]、MONOMI[57]等系统。CryptDB 针对不同查询类型使用了保序加密[58]、半同态加密[59]、全同态加密[50]等算法对数据进行加密存储,而 MONOMI 则在 CryptDB 的基础上能够对更大规模的数据集进行加密存储,并支持更加复杂的分析操作。相较于基于软件的加密方式,基于硬件的加密将部分

操作转移至可信硬件处理单元，获得更好的效率和通用性。比如 TrustedDB[60]和 Cipherbase[61]利用安全协处理单元(SCPU)[62]等硬件模块来减少查询开销，StealthDB[63]则利用 Intel SGX 技术[64]对加密数据进行存储管理以支持更多类型的 SQL 查询。

全密态数据库研究挑战。尽管密态数据处理在加密技术等方面已经有了较为全面的研究，但在实际使用中仍然存在一些挑战**包括执行效率的挑战和安全性的挑战**，具体简介如下：

- **基于存储和计算开销的效率挑战：**首先，在实际应用中人们对处理效率的需求越来越高，比如对于关键词检索等数据分析需要能够处理快速更新的数据[65]。然而当前常见的密态数据处理系统如 CryptDB 使用的保序加密、同态加密等加密算法会带来巨大的计算开销，比如同态加密会使计算所占用的存储空间和运算时间增长分别达到 10 和 12 个数量级[66]。因此如何在保证数据安全的情况下提升密态数据处理系统的效率成为一个新的挑战。

- **基于数据操作过程的安全性挑战：**目前的加密算法大多是对数据本身的保护，然而除了数据的明文信息之外，对数据的访问操作模式也可能属于敏感信息。比如对数据进行操作的类型、操作数据的频次以及触发操作的原因等数据访问模式都有可能泄露数据隐私信息[67]。所以最大程度的保护数据隐私也成为了密态数据处理系统的另一个挑战。

全密态数据库发展趋势。近年来随着 5G、物联网等技术的发展，数据量将呈现爆发式增长的趋势。而另一方面，随着我国的《网络安全法》、《个人信息安全规范》等法律条规的出台，数据使用时的隐私泄露问题日益被人们所关注。因此能够高效且安全地进行大规模数据的加密存储将会是密态数据处理在未来的发展趋势。具体的发展方向可以包括查询精度与查询效率的平衡、动态数据的安全存储以及软件加密和硬件加密的结合。

- **软硬结合的密态数据处理系统。**综合现有的密态数据处理系统，基于软件的加密方式实现成本较低但计算开销大，而基于硬件的加密方式虽然有较高的效率但对处理器的要求高，实现成本较大。因此，未来的密态数据处理系统可以利用可信硬件（TEE）实现软件和硬件结合的密态数据处理技术，发挥软件加密的灵活性和硬件加密的效率两大优势，使其能够在尽可能少的硬件资源下充分提高系统效率。
- **支持范围查找（排序查询、like 查询）的密态索引。**传统的 B 树索引可以支持高效的范围查找，但是加密后的数据不能支持保序（例如 $X < Y$ ，加密后的 $e(X)$ 和 $e(Y)$ 不一定满足 $e(X) < e(Y)$ ），而传统索引和查找技术强依赖与保序特性。因此需要研究高效的支持范围查找的索引结果以支持数据库快速查找。
- **实现动态数据的安全存储。**在很多的应用场景中，数据库中的数据需要动态地更新，比

如出租车公司的运营数据。但现有的密态数据处理系统大多只考虑了静态数据，将其用于动态数据的存储可能会存在着数据的更新模式泄露等安全问题。因此，如何实现动态更新场景下的密态数据处理是未来重要的研究方向。

2.5.4.2 数据安全多方计算

安全多方计算研究动机。数据安全多方计算是在多方场景下进行数据计算的关键问题，指多个数据拥有方在保证不泄露各自数据隐私的前提下共同完成计算。随着多个国家地区相继出台数据安全相关法律法规，限制各大公司直接共享数据进行分析计算，使得数据孤岛现象日趋严重。如何在保护各公司数据安全的同时对多方数据进行联合分析，成为打破数据孤岛充分挖掘数据价值的关键。安全多方计算技术的基本思想由图灵奖得主姚期智院士在 1982 年提出[68]。其主要目标是让多个互相独立且彼此不信任的数据拥有方在不依赖可信第三方的条件下，仅凭借各自的秘密输入联合完成某预定函数的计算。不同于上一节的密态数据处理技术将数据加密后交由第三方保存和计算，安全多方计算的每个参与方都是数据的实际拥有者，均参与目标函数的计算过程。由于安全多方计算的特点恰好能够应对数据孤岛带来的挑战，近年出现了一批将该技术应用于数据处理的研究工作。

安全多方计算研究现状。近年来数据安全多方计算的研究工作主要聚焦于构建安全多方的数据查询系统。最早结合安全多方计算技术构建数据查询系统的研究工作是 SMCQL[69]，其借助安全多方计算中的混淆电路技术，能够联合两个参与方的关系型数据库执行复杂的 SQL 查询，且不泄露除查询结果之外的任何其他数据。随后 Conclave[70]系统将该框架应用于大数据处理引擎之上，结合安全多方计算中的秘密共享技术，能够联合三个参与方各自的大数据引擎执行复杂分析。但受限于安全多方计算协议，上述系统的执行效率还存在较大的提升空间。

为提升数据安全多方计算的执行效率，后续工作从结果精度与特定操作两个角度着手。在精度与效率的平衡上，一些工作如[72,73]将差分隐私技术与安全多方计算技术相结合，以降低结果精度为代价提升计算的执行效率。在特定操作上，一些工作[71,74]在数据库的经典操作如连接操作等进行针对性优化，还有工作聚焦于数据库的函数依赖问题进行探索。然而这些工作的性能表现较真实应用的需求还有一定差距。

安全多方计算研究挑战。目前数据安全多方计算的研究工作均难以直接应用至工业生产场景，因此构建可用的数据安全多方计算系统始终是本领域重要的导向目标，其中挑战主要为

效率与参与方数量两点，具体如下：

- **基于通讯和计算开销的效率挑战。**经过数十年的发展，安全多方计算协议的执行效率已经有大约五个数量级的提升[76]，并出现了一些易于开发使用的安全多方计算工具库，但其执行效率受限于安全多方计算协议的高通讯开销，相比直接明文计算仍有较大差距。例如联合两个参与方执行相同的 SQL 查询，使用安全多方计算库的耗时较明文计算相差约四到六个数量级，部分复杂查询运行时间可达数十个小时[71]，难以应用至工业生产环境。
- **关于参与方数量的可扩展性挑战。**现有的安全多方计算技术难以适应多个参与方的场景。例如常用的基于混淆电路的安全多方计算库[77,78]仅支持两个参与方，而支持更多参与方的安全多方计算库在参与方数量稍大时运行效率就会显著下降[79]。因此如何适应实际应用中多个数据拥有方参与的场景，也是本领域的重要挑战。

安全多方计算发展趋势。实现数据的安全多方计算是打破数据孤岛的有效途径，因此仍将是未来重要的研究方向。为研究切实可行的安全多方数据计算系统，有如下发展趋势：

- **平衡应用中的安全性与执行效率。**针对具体的数据处理应用场景，结合安全多方计算中协议以优化数据处理任务，目前仅针对极少部分的数据库操作进行了优化，因此还有许多操作与计算任务在未来具有很大的优化空间。对于不同应用场景下不同的安全性需求，可以合理降低协议安全性以提升计算效率，从而设计出高效且满足具体场景安全性需求的协议。
- **结合可信硬件加速安全多方计算。**充分利用 Trust Zone、Intel SGX 等可信硬件提供的隔离执行环境及专用指令，降低传统安全多方计算协议的通讯和计算开销，设计和实现软硬件结合的高效安全多方计算协议，将安全操作的开销控制在真实场景可接受的范围之内。
- **构建易用的安全多方数据共享系统。**结合安全多方计算技术设计实现能够支持各种查询分析操作的实用系统。现有的系统级工作仅仅将数据库系统与安全多方计算做了简单整合，支持的操作少且只能支持有限个数的参与方。并且没有考虑传统数据库领域中常见的索引、事务、权限管理等经典问题，距实际应用有较大差距，如何将安全多方计算与经典的数据库领域问题相结合并设计出可在真实场景中落地应用的系统始终是数据安全多方计算的发展目标。

2.5.4.3 防篡改数据处理

防篡改数据库研究动机。如何保证数据不受恶意篡改是数据维护中关乎数据安全的基础性问题。一般可通过哈希技术与签名技术验证数据的完整性，再结合身份验证等加密方法构建数据防篡改机制。然而随着大数据时代的飞速发展，数据规模发生爆炸性增长，云服务也逐渐普及。以往的防篡改机制难以适应在复杂环境下对大规模数据的保护要求。因此如何构建高效的数据防篡改机制以及如何在不可信环境下保护数据安全成为亟待解决的研究问题。与此同时伴随加密货币兴起的区块链技术[85]受到了极大的关注。由于区块链具有不可篡改性这一特点能够保护数据不被篡改，同时由于区块链的去中心化与可追溯性等特性，保证了区块链能在不可信环境下构建可信的计算环境。因此结合区块链技术进行数据管理一直是近年来热门的研究问题。

防篡改数据库研究现状。结合区块链技术实现数据管理这一研究问题的相关工作中，主要分为数据存储与事务处理两大类。在数据存储方向可按数据存储类型分为键值对数据存储[86,87]与关系型数据存储[88,89]两种。此外也有研究工作聚焦于数据的版本控制问题，使用区块链技术构建支持版本查询的数据存储管理系统[90]。在事务处理方向，研究工作聚焦于在区块链上完成数据库的事务并发控制[91]、数据的访问控制授权[92]与查询处理优化[93,94,95]等传统数据库的经典问题。

除此两类方向之外还有应用驱动的相关研究工作。由于区块链分为公有链、许可链与私有链三种，三种区块链基于不同的信任程度各自具有不同加入退出机制与共识算法，适用于不同类型的数据管理场景。由此产生了基于不同类型区块链的数据管理研究工作[96][97]。

防篡改数据库研究挑战。尽管构建基于区块链技术的数据管理系统已经成为工业界应用与学术界研究的热门方向，然而目前区块链管理系统难以应用至生产中，在构建可用的区块链管理系统上还存在如下效率与隐私性关键挑战：

- **基于共识算法的执行效率挑战。**目前基于区块链的数据管理系统的主要瓶颈即为系统的吞吐量问题。现有广为使用的比特币其吞吐流量为平均支持每秒 7 笔交易，而大型信用卡公司每秒需执行 10000 笔左右的交易。因此现有区块链系统的吞吐量与完全无法媲美传统的分布式数据库系统，难以满足实际应用场景中的吞吐需求[98]。
- **多方参与的数据隐私性挑战。**由于区块链技术具有不可篡改、可追溯与去中心化的特性，因此区块链数据系统也常被认为是解决多方数据共享问题的有效途径[99]。然而该场景下，由于区块链技术基于对等网络的架构，导致难以保障链上各方的数据隐私；当共享

场景涉及多条区块链时，存在跨链场景下数据难协同的问题。

防篡改数据库发展趋势。由于区块链技术能在不可信环境下提供可信计算环境，可以适用于复杂的存储计算环境，因此继续研究基于区块链技术的数据管理方法仍将是数据领域未来重要的发展趋势。在该问题上未来有以下三个发展方向。

- **平衡系统可信性与吞吐量。**尽管目前已有工作研究提高区块链数据系统的吞吐量[100][101]，然而要想将其真正应用于实际工业生产环境中，则现存的系统在性能上仍有巨大的提升空间。在该问题上可从存储方式与共识算法两个角度着手。从存储方式上：可采用链上链下混合式数据存储方式提升系统对大规模数据的可扩展性；可利用分片、多链等技术提升系统吞吐量[102]。在共识算法上可从许可链等不同类型的链分别讨论，根据不同应用场景引入监管中心或半监管中心等提升共识算法效率。
- **基于链上链下混合存储的防篡改机制。**目前为提升区块链处理数据规模，链上链下混合存储已成为区块链系统发展趋势。基于混合存储方式如何实现媲美传统数据库的事务处理效果是区块链数据系统易用性的关键之一。具体问题例如：如何使用智能合约构建灵活动态、细粒度的访问控制机制；如何基于混合存储方式构建安全高效的数据索引；如何设计可验证的高效查询计算方法等。
- **面向跨链场景的数据协同处理系统。**面向多方参与场景的跨链数据协同处理系统存在重大应用价值，此类系统主要包含多链系统架构、跨链机制与数据互访互操作机制等方面。如何在保护数据隐私前提下完成协同共享，其中可能存在如下具体问题：在多链系统架构层面是否结合不同参与方的数据量与可信性确定整体系统架构；在跨链机制上已有的公证人等跨链交易机制如何适应数据协同计算应用场景；在数据互访互操作机制上如何设计跨链的数据访问控制确权机制。

2.5.4.4 数据隐私计算

数据隐私计算研究动机。数据隐私计算常用于保护数据在采集与发布阶段中涉及的数据隐私，通过增强所采集发布数据的不可区分性来达到隐私保护目的。在大数据时代下，越来越多的服务平台例如医疗机构、银行、电商和社交媒体等均需通过采集大量的用户数据，对用户数据进行分析建模，从而提升平台的服务质量。然而，各个平台需采集的数据往往涉及到敏感的私人数据。随着人们的隐私保护意识逐渐增强，人们往往不愿将涉及个人隐私的数据直接提供给这些平台。近些年，各国也纷纷通过立法的形式规范在数据的采集和发布过程中

应遵循的隐私保护政策。这些相关法律法规的出台也在不同程度上增加了对隐私数据进行采集与发布的难度。因此，如何在保证数据隐私不被泄露的前提下进行数据的采集与发布成为了近些年的一大研究热点。

数据隐私计算研究现状。数据隐私计算领域的研究工作在早期主要基于数据泛化方法，近年来基于添加噪音的差分隐私(Differential Privacy)方法为该领域开辟了新的重要发展方向。基于数据泛化方法的研究多在 21 世纪初期，例如 k 匿名(k-Anonymity) [103]，l-多样性(l-Diversity) [104]，t-保密(t-Closeness) [105]等隐私保护机制。然而，采用数据泛化的方法仍然存在隐私泄露的风险。

为进一步提高隐私保护力度，Dwork 等人提出了差分隐私(Differential Privacy)[106]的概念。该方法通过对数据添加噪音的方式保证添加或删除任意一条数据不会对查询结果造成很大的影响，从而保证个人的数据隐私，添加噪音的大小与给定的参数 ϵ 以及数据本身的灵敏度有关。这一方法在数据隐私计算领域的应用也越来越普遍，例如利用差分隐私进行数据发布[107]、数据定价[108]、数据查询[109]等。近年来，考虑到不同的用户对隐私保护的要求可能并不相同，本地化差分隐私(Local Differential Privacy) [110]作为一种无需可信第三方的隐私保护机制也逐渐兴起，在频率估计[111]、位置采集[112]、主题模型[113]等场景中的应用已初见端倪。

数据隐私计算研究挑战。尽管差分隐私技术在数据隐私计算领域的应用越来越广泛，这一技术的发展仍存在着**精确性**和**动态性**两方面的挑战。

- **查询结果精确性挑战。**相较于其他安全计算技术，差分隐私技术通过对数据添加噪音的方式保护数据隐私，其优势在于其计算成本低，无需复杂的加密计算。然而，这会导致查询结果的不精确。因此，如何在保证隐私保护力度的前提下，尽量减少添加噪声带来的误差一直是将差分隐私技术应用到数据隐私计算领域的一大挑战。
- **应用场景动态性挑战。**在真实应用场景中，数据往往会动态实时变化且同一数据可能被多次查询，此时存在隐私泄露风险。例如对同一数据多次进行查询时，差分隐私机制每次添加的噪音虽不相同，但攻击者可以通过对其取平均值的方式估算出真实数据。此时，直接采用传统的差分隐私机制仍有可能造成隐私泄露。如何设计更符合数据库领域中实时、动态特点的隐私保护机制，也是将差分隐私技术应用到数据隐私计算领域的挑战之一[114]。

数据隐私计算发展趋势。差分隐私机制为用户隐私数据的采集和发布提供了理论保证。未来几年，继续研究基于差分隐私机制的数据采集和发布方法仍将是数据隐私计算领域的发展趋势。

势。具体而言，这一领域也将主要围绕差分隐私机制中的精确性、动态性以及近似查询处理相结合三方面展开。

- **平衡隐私保护力度与精确性。**差分隐私机制中往往涉及到隐私力度和精确程度的权衡。噪音添加得过大就会导致查询结果不具有参考价值，噪音添加得过小又会影响隐私保护的力度，增加隐私泄露的风险。而本地化差分隐私机制对隐私保护的力度更强，其添加的噪音相较于传统差分隐私也更多。因此，如何利用置乱模型(Shuffle Model)[115]、多方差分隐私(Multiparty Differential Privacy)[116]等方式优化本地化差分隐私机制，使其在不损失隐私保护力度的前提下进一步减少查询误差，提升查询结果的准确性是该领域的研究方向之一。
- **设计动态场景下隐私保护机制。**目前大多数的差分隐私和本地化差分隐私机制都仅针对单轮次的查询进行设计。然而，在真实场景中，往往需要对动态数据进行多轮次的查询。此时，直接采用传统的差分隐私机制仍有可能造成隐私泄露。因此，如何针对动态数据、连续的查询设计隐私保护机制[117]，使其能够真正在各个服务平台的真实应用场景中落地应用也是未来的研究方向之一。
- **结合数据隐私计算与近似查询处理。**近似查询处理通过采样等方式可在保证查询误差不超过一定范围的前提下快速得到查询结果。由采样等方式带来的查询误差也在一定程度上掩盖了个人的隐私信息。因此，将近似查询处理技术与数据隐私计算领域的差分隐私技术进一步结合，在保证较小查询误差的同时提供隐私保障，也是数据隐私计算领域未来的发展趋势之一。

2.5.5 其他数据管理技术

此外，还有很多其他数据库技术需要深入研究，例如嵌入式数据库（资源受限的数据管理）、视频数据库（视频数据的抽取、管理和查询处理）、面向量子时代的数据管理、基于编译优化的数据处理技术、仓湖一体数据库（数据仓库数据湖融合数据管理）等。

3. 数据科学的发展趋势

3.1 数据科学基础

数据科学基础研究动机。图灵奖得主 Jim Gray 曾将数据科学认为是科学的“第四范式”，其核心是以数据为驱动。当前各行各业掀起的数字化转型浪潮，已经从早期的以信息化建设为基础的 1.0 版升级到当前“以数据为中心”的 2.0 版。毋庸置疑，在当前“以数据为中心”的数字化转型浪潮中，以数据为驱动的“数据科学”必将起到非常关键的推动作用，也将为转型提供必要的基石。

数据科学是一门多学科交叉的学科，一般将其看作是数据库管理、统计和机器学习、分布式和并行系统三个专业方向的结合。它实质上是围绕结构化和非结构化数据，通过使用科学方法、流程、算法和系统从数据中提取知识和洞察，并将这些知识和洞察应用到广泛的应用领域，从而提升社会的运转效率和人们的认知。

虽然数据科学是一门应用性极强的学科，但它的产生和发展均依赖一些基础。我们可以把数据科学的基础概括为数学基础（理论基础）、数据基础、计算基础、人才基础等多个方面。随着数字化的深入，数据科学在得到长足发展的同时也面临着一系列新的挑战和问题，需要学术界和工业界共同努力去寻找答案。

数据科学基础研究现状。数据科学发源于计算机科学、统计学、信息论等多个学科和方向，其诞生和发展有着坚实的数学理论基础，包括支撑早期数据库发展的关系代数、作为深度学习计算基础的线性代数、机器学习中的概率统计等。数据科学一方面得益于这些基础理论，另一方面这些专业方向的发展也将持续为数据科学的发展提供养料。在数据基础方面，数据科学所处理的数据对象已经从传统的结构化关系型数据发展到结构化和非结构化并存的多源异构数据，从表层的感知数据发展到深层的认知知识。有研究表明，未来产生的 80% 数据都是非结构化数据。这为数据科学的发展带来极大的“数据红利”，但同时也带来了各种挑战。在计算基础方面，围绕大数据的分析处理出现了两种计算范式。一种是所谓的 **Scale-up** 范式，即通过增强处理能力的方式来满足大规模数据的快速分析和处理需求，例如现在非常流行的各种分布式计算技术和框架。另一种是所谓的 **Scale-down** 范式，即通过从原始数据中提取各种数据概要（**Data Synopsis**）来缩小数据处理规模，以便提升分析效率，例如基于采样的各种近似查询处理方法。从数据的全生命周期角度，在数据的获取、准备、管理、

计算、呈现等各个环节均出现了非常多的技术和工具，也逐步形成了蓬勃发展的数据科学生态。可喜的是我们看到越来越多的中国企业和学术机构在为这个生态的健康发展贡献着力量。在人才基础方面，国家高度重视数据科学人才的培养。截至到 2020 年我国已有 600 余所本科院校开设了数据科学与大数据技术相关专业，为我国数据科学的发展持续输出高质量的人才。

数据科学基础研究挑战。随着数字化进程的不断深入，数据科学的基础也面临着各种新问题和新的挑战。

- **数据建模。**随着非结构化数据的不断增多，越来越多的数据被表达成向量/张量。这对数据的管理、检索、利用都带来了新的挑战。此外，我们知道深度学习依赖于“大”数据，然而很多行业/企业没有大数据，但有高质量的“小”数据。因此，如何破解“小数据难题”也是摆在我们面前的新问题。此外如何对数据建模，定义数据的价值和复杂度也是亟待解决的难题。
- **计算模型。**我们通常在数据准备阶段采用关系型数据库（以关系代数为基础）来管理和处理数据，而在计算/预测阶段我们却采用向量来表达数据，并采用线性代数的方法进行计算。显而易见，同一套数据在两套代数（关系代数和线性代数）系统之间进行传递需要进行（等价）转换，这会带来极大的转换开销，并为跨系统的优化带来极大的障碍。因此，这里就亟需我们进行创新来克服这个问题。
- **计算范式。**Scale-up 和 scale-down 两个计算范式都面临着不同的发展问题，例如超算作为国之重器，如何通过软件栈改造升级实现平民化，以便更广泛地服务于国民经济发展。再如传统的概率采样方法如何适应如时空数据这样的新型数据类型。此外，数据共享中的安全和隐私保护也面临着挑战。需要研究新的计算范式。

数据科学基础发展趋势。从需求发展的角度将数据科学的发展趋势概括为以下几个方面：

- **数据科学全流程分析框架。**数据分析流程呈现复杂化趋势，现代的数据科学不仅仅是解决简单的统计计算问题，更多的是解决复杂的决策分析和预测问题。问题的复杂化随之带来的是数据分析流程的复杂化。一个复杂的数据分析流程往往会涉及各种技术和系统，是一个异构软硬件系统的综合利用过程。在这个过程中为了提升分析处理效率，有很多开放问题值得研究。例如在面对成千上万的算法模型时，如何自动选择合适算法模型？如何选择合适的参数配置？在涉及多个系统的分析流程中，如何实现跨系统的整体性能优化？其中还涉及一些基础理论问题，如跨代数系统的数据统一表达和转换机制等。
- **端到端全流程数据科学系统。**数据科学呈现平民化趋势，数据科学少不了人的参与，其

过程是一个典型的“Human-in-the-loop”的过程，但是目前缺少端到端的数据库科学系统。随着越来越多的人对数据分析有需求，数据科学的发展呈现一种平民化的趋势。为了适应这种发展趋势，需要在更多的方面做出创新。例如，如何利用人工智能技术发展智能数据分析工具，使得数据工具更易用、更亲民。再如，如何发展可自然语言交互的可视化探索分析技术和工具，来进一步降低数据科学的门槛，使得人人都能成为数据科学家，从而有效解决数据科学人才短缺的问题。

3.2 数据科学方法

数据科学方法研究动机。在各行各业数字化转型浪潮的推动下，近些年，大数据技术和产业都得到了快速发展。在这样的背景下，围绕大数据的数据科学研究，也取得了前所未有的进展。从数据库研究的视角，数据科学更关注于数据整个生命期中的管理和分析问题，包括数据探索 and 发现、数据抽取、数据清洗、数据集成、数据分析、数据可视化等众多方面，它一直以来是数据库研究的一个重要方向。近些年来，为应对大数据的一些挑战，数据科学与传统数据库研究又有了一些新的不同，主要体现在：1) 数据已不单纯地存放在数据库或者数据仓库中，它可以是独立的数据集，也可以存在于更混杂的数据湖中；2) 计算不再是关系数据库中和系统紧密绑定的一些关系运算，而是更为宽泛的数据分析处理模型和算法；3) 计算结果不再是关系运算得到的准确结果，数据科学的分析结果会受所选用的模型和算法影响，在准确度上有所不同；4) 在人工智能高速发展的时代，很多数据科学方法越来越多的采用了深度学习技术，相应的开发和运行环境也以交互性更强的 Python 环境和一些硬件加速环境为主。在这样的发展趋势下，数据科学的研究成果如何更好地融入到数据库系统的研发中？数据库研究者能为数据科学体系做出什么样的贡献？都是值得我们去深入思考的问题。

数据科学方法研究现状。以数据准备、数据管护、数据集成等为代表的数据库科学研究近些年来得到了很多数据库学者的关注。数据库领域的两大旗帜会议 SIGMOD 和 VLDB 分别设立了 Data Science and Engineering 和 Scalable Data Science 两个 Track，吸引了很多学者的关注。近些年来，围绕数据生命期中的各种数据管理与数据分析任务，研究人员在模型和算法层面做出了很多创新和贡献，发表了很多高水平的数据科学论文。然而，这方面的研究成果却很少能够产生较大的影响力。究其原因，一方面，数据科学研究范围广，研究点碎片化、具体化。每个研究工作可能解决具体的一个小问题，不同工作之间关联性差，使得研究

成果缺少整体性，显示度不高；另一方面，研究成果缺少工具化、系统化的研究导向，难以在实际的系统或应用中进行有效的验证，这与传统数据库研究所崇尚的以系统实现为导向的理念有所偏离。这进一步造成了我们在数据科学的很多方向上做了很多重复性和增量性的研究工作，不断重新发明轮子，浪费了很多人力和物力。

我们也看到，数据科学越来越重视开放科学的理念。以 **PaperWithCode** 为例，研究人员在发表论文的同时还开放代码和实验数据，注重研究工作的可重复性。在这样的背景下，越来越多的数据科学研究工作走开源的路线。然而，仅仅开源并不能很好地解决数据科学工作产生的影响力和对学科贡献不足的问题。早在 2016 年的中国数据库学术会议（NDBC）上，美国威斯康星大学 Anhai Doan 教授在其主题报告中就明确建议到，和数据库相关的数据科学研究要重视围绕具体问题构建 Python 生态社区，形成良性的发展模式。今天我们在数据科学研究所面临的影响力不足问题，恰恰暴露了我们在生态建设方面的不足。中国计算机学会数据库专委会作为学术机构，有责任发挥组织的力量，动员相关数据科学工作者积极参与到数据科学生态社区的构建中，做有组织的科研。

数据科学方法研究挑战。在数据科学高速发展的今天，研究人员根据各自的研究领域和兴趣点，做出了大量的研究成果，我们面临众多的研究方向和研究点。在这样背景下，想要梳理出脉络，构建良好的数据科学生态体系，要面临很多挑战：

首先，面对如此多的数据科学领域，选择那些研究点去构建社区是很有挑战的难题。范围太大，会使得涵盖面广，研究点关联不紧密；范围太小可能又聚集不起来人气，产生不了影响力。选择合适的社区关注粒度，以及这个粒度下的研究点如何与数据库系统的研究更好的结合起来，对我们的社区建设是至关重要的。

其次，有了社区的关注点后，如何汇聚人气，使得相关方向的研究人员愿意把自己已有的和正在研究的工作汇聚到社区中，愿意更多地参与和贡献自己的智慧，是社区建设最大的挑战。作为专委会，我们如何设计机制来吸引同行的积极参与，如何让每位参与者能够得到归属感和荣誉感、能够通过社区建设提升参与者的科研效率和影响力（反哺参与者），都是社区建设面临的挑战性问题。

再者，社区要持续良性的发展，需要核心人员持续的投入，需要企业的参与和贡献，需要有从成果到产品的转换，才能产生更大的影响力。如何借助社区，吸引企业的参与和投入，形成一系列基准和标准，进一步指导相关产业的健康发展；如何借助社区，吸引教师的参与，形成一系列和应用紧密关联的教学案例。这些都是社区建设的要点和所面临的巨大挑战。

数据科学方法发展趋势。虽然国内外学术界和工业界都非常关注数据库科学的发展，但是目

前还处于探索阶段，还需要研究其中关键共性技术。

- **打通数据科学全链条关键技术：**数据科学包含了数据探索与发现、数据抽取、数据清洗、数据融合、数据分析、数据可视化等多个关键步骤，目前还没有系统和平台来打通数据科学全链条平台。目前主要有两大挑战，一是缺少统一的标准和接口，难以打通不同模块；而是缺少可复制的技术能够迁移到不同的应用。目前都需要人工参与，所以很难搭建可复制的数据科学平台。因此需要通过统一标准和可迁移技术等方面深入研究数据科学的共性关键技术。
- **以数据库系统的数据科学工具集构建为社区建设驱动力：**在数据科学平民化需求的驱动下，更多的易用性强的数据科学工具是数据科学平民化的重要支撑，也是数据库系统延伸到数据科学领域必须具备的基础功能。这为我们选择合适的数据科学研究点，以某类数据库系统为核心构建相应的数据科学工具集，以此来推动社区的建设提供了重要参考。比如，以时间序列数据库为例，结合某个开源时序数据库，配套数据修复、补全、可视化、预测、根因分析等工具集，就可以构建一个应用领域相对集中的数据科学社区。
- **抽象数据科学任务，以基准研发促进社区良性建设：**围绕某类数据库，我们需要抽象出一些具体的数据科学任务，对应到一些具体的数据科学工具上，每个工具背后的核心是相应的模型和算法。在这样抽象基础上，可以结合一些典型应用领域，利用已有的数据集或开发一些新的数据集，做成相应任务或研究点的评测基准。社区为每个任务提供标准化的评测数据集和评测接口，研究人员只需关注于核心算法和模型的改进，无需重复运行已有的对比实验，努力提升在相应数据集上的评测得分，形成类似传统数据库评测基准的榜单，进而促进数据科学在具体问题上的突破。使得研究成果具有更好的公开性和公平性，也能更好地和工具集生态建设融合起来。
- **以社区建设促进行业标准、教学案例的建设，提升影响力：**为了更好地产生影响力，社区建设产生的成果需要在企业和教学中找到落地点。数据科学领域普遍面临缺少标准支撑，自家产品都说自家好，野蛮生长的局面。由学会或者专委会推动建设的社区，可以作为第三方，吸引更多企业的参与。在建设过程中，以应用驱动研发的原则，逐渐摸索出哪些工具是必须的，工具接口应如何设计，逐渐形成标准，这将对引导数据科学相关产品和产业的健康发展发挥重要作用。与此同时，好的工具还可以作为教学案例，在数据科学类课程中发挥作用，形成更紧密的产学研良性发展途径。

4. 国产数据库生态建设

数据库已经有了 60 年的发展历史。2020 年之前数据库系统都由被美国垄断，例如商业数据库 Oracle、DB2、SQL server，和开源数据库 PostgreSQL 和 MySQL。近年来国产数据库也开始加大了投入，开始逐渐进入到国家核心系统，例如金融和证券。虽然国内数据库厂商的技术相比 Oracle 有差距，但是目前来看最重要的卡脖子点不是技术本身，而是数据库生态的建设。

数据库生态指由基于数据库管理系统（DBMS）软件产品和服务的组织与个人，通过他们之间的交流、合作、供求满足等相互作用而形成的环境。一个成熟的数据库生态不仅包括功能完善、性能优秀、运行稳定的系统，还包括完整的文档与知识库、丰富多样的开发和运行维护工具链、典型的应用案例和评测报告，以及一批熟悉系统、经验丰富的开发者和用户。典型的数据库生态包括以 Oracle、IBM DB2、Microsoft SQL Server 为代表的商业数据库生态，以 MySQL 和 PostgreSQL 为代表的开源数据库生态，以及以 TPC 为代表的事务处理性能基准评测生态等。数据库生态是 DBMS 软件产品保持稳定持续发展，取得商业成功的关键，也是当前国产数据库系统最欠缺的方面。

以下三个方面在我国数据库生态建设中具有紧迫性。

4.1 数据库评测基准建设

评测基准是评价数据库系统最重要的标准，是应用进行数据库系统选型的重要依据，是促进数据库系统研发的主要驱动力之一。因此，可以说评测基准是支撑数据库产业健康发展 40 多年的最重要支柱之一。例如，TPC 系列评测基准的出现和发展使诸多关系数据库管理系统厂商能在一个公平、公开的环境中良性竞争。

以 TPC 系列为代表的现有评测基准在数据库系统普遍适配云计算平台，采用分布式架构，并提供越来越多的新数据类型支持和新数据管理和数据处理服务的场景下，并不能满足当前以及未来系统比较、系统选型、指导研发、规范产业的需要。同时，作为企业和行业标准的基础和前提，形成能够反映我国应用需求和系统厂商优势的评测基准也是规范我国数据库产业，培育国产数据库系统的先决条件。

- **分布式数据库系统的正确性测试：**新型数据管理系统普遍采用分布式架构实现可扩展性、高性能以及高可用。分布式的架构也同时带来了支持分布式事务的复杂并发控制协

议以及支持高可用和快速恢复的分布式共识协议。协议的正确性以及工程实现的正确性都会影响分布式数据库系统的正确运行。一套面向分布式数据库的完整的事务正确性测试方法，以及公开、易用的测试工具是促进分布式数据库系统研发，提升新型数据库系统在关键核心任务应用（mission-critical applications）形成替代的必需品。现有的评测工具无法实现这一目的。

- **面向应用的评测基准构建方法：**随着数据管理系统在信息系统中所起作用的提升，其应用范围和应用的多样性和诞生之初相比都有了质的飞跃。2014 年图灵奖获得者 Michael Stonebraker 提出了经典的结论 “One size does not fit all!”，而 Michael J. Carey 则认为数据库系统的发展将呈现出 “One size fits a bunch” 的趋势，即具有类似特性或需求的每一类应用都需要不同的数据管理系统。近年来国内外雨后春笋般涌现的新的数据库系统验证了这一趋势的到来。“One size fits a bunch” 不仅应体现在新系统的出现，也应体现在对应的评测基准中。TPC 系列评测基准的本质是用 “一体通用（One size fits all）” 式的模型去评测所有系统。但是，类似于云环境、嵌入式环境、“现象级” 事务处理、多模态数据管理、数据管理与机器学习/AI 相结合的应用等场景，普遍缺乏公开和权威的评测基准。制定一成不变同时涵盖所有应用需求的评测基准一方面代价高昂，同时也难以适应快速变化和发展的应用场景。构建典型应用的评测基准，同时研发面向应用的评测基准构建方法和理论，实现便于定制的评测基准构建工具，是形成适应于新型数据管理系统和应用发展的评测基准，进而为我国数据库产业发展保驾护航的前提。
- **面向新型数据库系统的评测基准：**新型数据库系统（云数据库、分布式数据库、AI 原生数据库等）的形态、提供服务的方式、依赖的硬件环境都发生了巨大的变化。传统的以响应延迟、系统吞吐率为代表的性能评价指标不再能准确和全面反映新系统的特性。例如，对于能够以近线性扩展的系统，系统的最高吞吐率或者单机平均吞吐率都不足以反映系统的扩展能力。应用与数据库厂商都迫切需要能够评测云服务能力、评测系统可扩展性（scalability）/弹性（elasticity）/可用性（availability）/能耗以及新硬件利用率新的评测基准。新的评测指标和评测方法的制定需要应用单位确认其对于系统比较和选型的实用意义，也需要数据库厂商深度参与以体现诸多系统的特点和优势。这一工作对于指导系统研发方向，同时争夺数据库系统发展方向的话语权至关重要。

4.2 数据库开源生态建设

开源在数据库系统的发展过程中发挥了重要的作用：最早的关系数据库系统之一 Ingres 诞生之初即以 BSD 协议开源；具有广泛用户的 MySQL 和 PostgreSQL 都是开源系统；安装量最大的数据库系统 SQLite 也是开源系统。不仅如此，即使对于像 Oracle、IBM DB2 这样的商用闭源数据库系统，它们也依赖包含诸多开发和运维工具的庞大开源生态。

数据库开源生态建设在以下 5 方面具有重要意义：

- **技术公开与安全可靠保障：**关键核心任务应用对系统的安全可靠具有极高的要求。系统开源有助于系统的安全审计、公开评测、Bug 的及时发现与修补。
- **用户反馈收集与经验积累：**优秀的系统是“用”出来的，需要基于大量应用的使用进行优化和提升。开源有利于应用推广，有利于问题反馈和收集，也为用户间的经验积累和交流提供了媒介。换言之，开源社区将以众包的形式构建围绕数据库系统的知识库。对于新兴的数据库系统产品和厂商，这将极大节省由厂商自身构建知识库的成本。
- **开发和运维工具链构建：**数据库系统只是应用的软件栈中一个环节。应用设计与开发、中间件适配、应用运行维护都需要专用的工具。由于数据管理系统的复杂性，适配一个系统的工具通常不适用于其他系统；由于应用环境的复杂性，在不同的应用中，一个数据管理系统可能需要适配不同的操作系统、中间件、开发环境或者其他工具。一个数据库系统厂商，特别是对于新系统，在短时间内构建这样的工具链成本巨大。开源是积聚社区力量，减少重复工作，构建工具链的有效方式。
- **用户与产业链培育：**数据库系统，尤其是新系统，需要吸引用户，同时吸引应用开发者和服务提供商。开源是消除系统神秘感，宣传系统，培育潜在用户，构建开发、集成、运行维护产业链的有效手段。
- **研发人员教育与技术探索：**数据库系统核心研发需要一大批具有系统性训练的研发人员。研发人员不仅需要丰富的理论知识积累，也需要有大量工程实践锻炼。开源系统是阅读优秀代码、实现理论验证和工程实践训练的最佳载体，也是探索新方法新技术，进而形成研究成果验证和转化的最佳载体。

以上 5 个方面的需要包括数据库厂商、应用单位、研发机构以及研究和工程技术人员个体，从核心技术研发，到问答和文档建设，乃至社区文化建设等多个层面长期投入，形成围绕数据库系统产品的开源社区。

4.3 产学研用深度合作

数据库系统涉及的技术链条长，内部各模块耦合度高，与软件栈其他部分以及硬件平台关联程度高。无论是数据库系统核心研发，还是评测基准构建，或者开源生态建设，都需要产学研用各方密切合作。

近年来，华为 GaussDB、PingCAP TiDB、奥星贝斯 Oceanbase、阿里 PolarDB 等国内数据库系统已经通过开源开始构建多方参与的社区生态。同时，数据库厂商也广泛地通过项目形式与高校和研究所合作，代表性的合作计划包括 CCF-华为数据库创新研究计划、腾讯犀牛鸟项目、阿里 Air 项目等。这些项目使得厂商和研究机构间开始形成了较为稳定和密切的合作关系。

数据库系统的发展越来越呈现出“应用驱动创新”的特点。系统的发展受到应用需求的刺激和驱动。当前数据库系统研发和生态建设中，数据库系统厂商与高校和研究所等研究机构的合作较多，同时系统厂商和应用方也有密切的合作。但是，产学研用共同参与的合作仍不多。与美国、欧洲等先进的高校和研究所相比，我国研究机构面向应用研发系统原型，进而通过开源或者企业合作等形式孵化或者转化，形成具有重大影响力的数据管理系统产品的研发道路仍有待探索。这既需要在多方合作模式上的创新，也需要我国科研机构在成果和科研绩效评价机制上的创新。

- “学”：学校要承担起人才培养和引领性研究的使命。（1）在本科生培养方面，不仅要教给同学们“什么是数据库”、“怎么用数据库”，更要教给“如何设计数据库”、“为什么这么设计数据库”，让本科生知其然又知其所以然。此外，高校也应推广国产数据库走进课堂，让学生们更多了解和体验国产数据库，如何利用国产数据库系统，形成循序渐进的系列实验，在理论教学的同时开展工程实践教育和研究性教学仍有待深入研究。虽然国产开源数据库进入计算机等专业的教学体系仍在探索阶段，但是目前正在通过教育部产学研合作协同育人项目等形式积极推进。（2）在研究生培养方面，老师也要深入了解企业界需求，凝练有价值问题，引领学生们做那些“真、难、重”的重要挑战性问题，即数据库系统中重要、真实、挑战性、共性的基础性问题。学校还要承担起引领性科研的使命，突破“0 到 1”的顶天问题，引领产业界未来发展。
- “研”：科研机构主要负责攻克当前卡脖子的基础性、共性的关键问题，例如分布式事务和 Paxos 的融合、全球高可用等技术，解决当前遇到的挑战性问题。学校和科研机构也一定要深度合作，推动引领性技术的研究和落地。

- “产”：数据库研发企业要承担研制国产数据库的使命，一方面解决数据库卡脖子问题，另外一方面要推动技术的革新与进步，为数据库用户带来新的价值。当前交易型数据库（OLTP）还存在着稳定性、生态、兼容性等卡脖子问题，企业界要坚持不懈解决这些核心问题。此外还有很多新型数据库，例如图、时序、流、多模等，也需要企业界百花齐放、百家争鸣。
- “用”：数据库用户不应该只是简单的要求数据库提供商“apple2apple”的替换，而是通过数据库升级迭代来引领产品和应用的发展，以未来的眼光来看待数据库的更新换代。数据库用户也应该思考为“产、学、研”提供有价值、有前景的应用“真”需求，推动数据库领域的跨越式发展。

4.4 小结

数据库系统生态建设是促进国产数据库系统和产业健康发展的关键，也是促进国产数据库系统助力行业领域数字化转型，在关键核心任务中实现对国外商用数据库系统形成替代的关键。

数据库系统生态建设需要研发针对新应用场景的评测方法和工具，需要构建多方积极参与、密切互动、持续成长的社区，需要培育包括用户、产业链厂商、研发梯队的社区队伍。对于发展历史较短的国产数据库系统厂商，开源是构建这样的生态的重要方式。

围绕数据库系统开展产学研用多方紧密联系的合作仍刚起步。在现有的基于开源和基于项目的合作基础上，引入应用方，同时将合作向基础人才梯队培养延伸，探索新的国产数据库系统研发和生态建设道路，还有待大量的工作。

5. 总结

本文主要讨论数据库系统的发展趋势以及数据库技术的前沿热点。首先计算模式的改变和应用需求变化对数据库系统形态起到了至关重要的作用，也推动了数据库架构的迭代更新。本文讨论了云原生数据库、分布式数据库、端边云数据库、AI 原生数据库的研究挑战和发展趋势。其次，数据库技术近年来也得到了迅速发展，本文总结了基于新型硬件的数据管理、智能数据管理、多模数据管理、数据安全和隐私保护的未来发展趋势。本文也详细讨论了数据科学的未来发展趋势和研究热点。最后本文总结了国产数据库生态建设的思考。

参考文献

- [1] Corbett, James C., Jeffrey Dean, Michael Epstein et al. Spanner: Google's globally distributed database. In TOCS, 2012.
- [2] Arulraj, Joy, Andrew Pavlo. How to build a non-volatile memory database management system. In SIGMOD, 2017.
- [3] Dragojević A, Narayanan D, Nightingale EB et al. No compromises: Distributed transactions with consistency, availability, and performance. In SOSP, 2015.
- [4] The Apache Software Foundation. Apache Hadoop. <https://hadoop.apache.org/>
- [5] Thusoo, Ashish, Joydeep Sen Sarma, Namit Jain et al. Hive: a warehousing solution over a map-reduce framework. In VLDB, 2009.
- [6] Verbitski, Alexandre, Anurag Gupta, Debanjan Saha et al. Amazon aurora: Design considerations for high throughput cloud-native relational databases. In SIGMOD, 2017.
- [7] Lamport, Leslie. Paxos made simple. ACM Sigact News, 2001.
- [8] Ongaro, Diego, John Ousterhout. In search of an understandable consensus algorithm. In USENIX ATC, 2017.
- [9] Liu, M. L., D. Agrawal, A. El Abbadi. On the implementation of the quorum consensus protocol. In Proc. Parallel and Distributed Computing Systems, 1995.
- [10] McGrath, Garrett, Paul R. Brenner. Serverless computing: Design, implementation, and performance. In ICDCSW, 2017.
- [11] Yang, Jiacheng, Ian Rae, Jun Xu et al. F1 Lightning: HTAP as a Service. In VLDB, 2020.
- [12] IBM corporation. DB2 Database. <https://www.ibm.com/analytics/db2>.
- [13] Oracle Corporation. Oracle Database. <https://www.oracle.com/index.html>.
- [14] Microsoft. SQL Server. <https://www.microsoft.com/en-us/sql-server/sql-server-2019>.
- [15] Wang Y. Yang Y. Zhu WG. SQLflow: a bridge between SQL and machine learning. arXiv preprint arXiv:2001.06846, 2020.
- [16] Makrynioti N, Ley-Wild R, Vassalos V. sql4ml A declarative end-to-end workflow for machine learning. arXiv preprint arXiv:1907.12415, 2019.
- [17] Li XP, Cui B, Chen YR. Mlog: Towards declarative in-database machine learning. In VLDB, 2017.
- [18] Paszke A, Gross S, Massa F et al. Pytorch: An imperative style, high-performance deep learning library. In NIPS, 2019.
- [19] Abadi M, Barham P, Chen JM. Tensorflow: A system for large-scale machine learning. In OSDI, 2016.
- [20] Miao XP, Zhang HL, Shi YN et al. HET: Scaling out Huge Embedding Model Training via Cache-enabled Distributed Framework. In VLDB, 2022.
- [21] The Apache Software Foundation. Apache MADlib. <https://madlib.apache.org/>.
- [22] Vertica System. Vertica Analytics Platform. <https://www.vertica.com/>.
- [23] Li G, Zhou XH, Sun J. opengauss: An autonomous database system. In VLDB, 2021.
- [24] Relational AI, Inc. relationalAI. <https://www.relational.ai/>.
- [25] Stonebraker M, Brown P, Zhang DH. SciDB: A database management system for applications with complex analytics. Computing in Science & Engineering, 2013.

- [26] Orestis Polychroniou, Arun Raghavan, Kenneth A. Ross. Rethinking SIMD vectorization for in-memory databases. In SIGMOD, 2015.
- [27] Saurabh Jha, Bingsheng He, Mian Lu, Xuntao Cheng, Huynh Phung Huynh. Improving main memory hash joins on Intel Xeon Phi processors: an experimental approach. In VLDB, 2015.
- [28] Bharat Sukhwani, Mathew Thoennes, Hong Min, Parijat Dube, Bernard Brezzo, Sameh W. Asaad, Donna Dillenberger. A hardware/software approach for database query acceleration with fpgas. Int J Parallel Program, 2015.
- [29] Andreas Meister, Sebastian Breß, Gunter Saake. Toward GPU-accelerated database optimization. Datenbank Spektrum, 2015.
- [30] Louis Woods, Zsolt István, Gustavo Alonso. Ibex: an intelligent storage engine with support for advanced SQL offloading. In VLDB, 2014.
- [31] Louis Woods, Gustavo Alonso, Jens Teubner. Parallelizing Data Processing on FPGAs with Shifter Lists. ACM Trans. Reconfigurable Technol, 2015.
- [32] Bingsheng He, Ke Yang, Rui Fang, Mian Lu, Naga K. Govindaraju, Qiong Luo, Pedro V. Sander. Relational joins on graphics processors. In SIGMOD, 2008.
- [33] Rajendra Bishnoi, Fabian Oboril, Mojtaba Ebrahimi, Mehdi Baradaran Tahoori. Avoiding unnecessary write operations in STT-MRAM for low power implementation. In ISQED, 2014.
- [34] Jianhui Yue, Yifeng Zhu. Accelerating write by exploiting PCM asymmetries. In HPCA, 2013.
- [35] Jisoo Yang, Dave B. Minton, Frank Hady. When poll is better than interrupt. In FAST, 2012.
- [36] Moinuddin K. Qureshi, Vijayalakshmi Srinivasan, Jude A. Rivers. Scalable high performance main memory system using phase-change memory technology. In ISCA, 2009.
- [37] Gaurav Dhiman, Raid Ayoub, Tajana Rosing. PDRAM: a hybrid PRAM and DRAM main memory system. In DAC, 2009.
- [38] Hyunchul Seok, Youngwoo Park, Kyu Ho Park. Migration based page caching algorithm for a hybrid main memory of DRAM and PRAM. In SAC, 2011.
- [39] Qingsong Wei, Chundong Wang, Cheng Chen, Yechao Yang, Jun Yang, Mingdi Xue. Transactional NVM cache with high performance and crash consistency. In SC, 2017.
- [40] Fei Xia, Dejun Jiang, Jin Xiong, Ninghui Sun. HiKV: A hybrid index key-value store for DRAM-NVM memory systems. In USENIX ATC, 2017.
- [41] Jihang Liu, Shimin Chen, and Lujun Wang. Lb+ trees: optimizing persistent index performance on 3dpoint memory. In VLDB, 2020.
- [42] Christian Tinnefeld, Donald Kossmann, Martin Grund, Joos-Hendrik Boese, Frank Renkes, Vishal Sikka, Hasso Plattner. Elastic online analytical processing on RAMCloud. In EDBT, 2013.
- [43] Feng Li, Sudipto Das, Manoj Syamala, Vivek R. Narasayya. Accelerating relational databases by leveraging remote memory and RDMA. In SIGMOD, 2016.
- [44] Wei Cao, Yingqiang Zhang, Xinjun Yang, Feifei Li, Sheng Wang, Qingda Hu, Xuntao Cheng, Zongzhi Chen, Zhenjun Liu, Jing Fang, Bo Wang, Yuhui Wang, Haiqing Sun, Ze Yang, Zhushi Cheng, Sen Chen, Jian Wu, Wei Hu, Jianwei Zhao, Yusong Gao, Songlu Cai, Yunyang

Zhang, Jiawang Tong. PolarDB Serverless: A Cloud Native Database for Disaggregated Data Centers. In SIGMOD, 2021.

[45] Erfan Zamanian, Carsten Binnig, Tim Kraska, Tim Harris. The end of a myth: distributed transactions can scale. In VLDB, 2017.

[46] Ryan Johnson, Ippokratis Pandis, Nikos Hardavellas, Anastasia Ailamaki, Babak Falsafi. Shore-MT: a scalable storage manager for the multicore era. In EDBT 2009.

[47] Christopher Root, Todd Mostak. MapD: a GPU-powered big data analytics and visualization platform. In SIGGRAPH Talks, 2016.

[48] Hacigumus H, Iyer B, Mehrotra S. Providing database as a service[C]//Proceedings of the 2002 International Conference on Data Engineering. 2002: 29-38.

[49] Sun Y, Wang S, Li H, et al. Building enclave-native storage engines for practical encrypted databases[J]. Proceedings of the VLDB Endowment, 2021, 14(6): 1019-1032.

[50] 赵志远, 王建华, 徐开勇, 等. 面向云存储的支持完全外包属性基加密方案[J]. 计算机研究与发展, 2019, 56(2): 442-452.

[51] Hacigümüş H, Iyer B, Li C, et al. Executing SQL over encrypted data in the database-service-provider model[C]//Proceedings of the 2002 ACM international conference on Management of data. 2002: 216-227.

[52] Savvides S, Khandelwal D, Eugster P. Efficient confidentiality-preserving data analytics over symmetrically encrypted datasets[J]. Proceedings of the VLDB Endowment, 2020, 13(8): 1290-1303.

[53] Shi E, Bethencourt J, Chan T H H, et al. Multi-dimensional range query over encrypted data[C]//Proceedings of the 2007 Symposium on Security and Privacy. 2007: 350-364.

[54] Wong W K, Cheung D W, Kao B, et al. Secure knn computation on encrypted databases[C]//Proceedings of the 2009 ACM International Conference on Management of data. 2009: 139-152.

[55] Wang H, Lakshmanan L V S. Efficient secure query evaluation over encrypted XML databases[C]//Proceedings of the 2006 International Conference on VLDB. 2006: 127-138.

[56] Cao N, Yang Z, Wang C, et al. Privacy-preserving query over encrypted graph-structured data in cloud computing[C]// Proceedings of the 2011 International Conference on Distributed Computing Systems. 2011: 393-402.

[57] Popa R A, Redfield C M S, Zeldovich N, et al. CryptDB: Protecting confidentiality with encrypted query processing[C]//Proceedings of the 2011 Symposium on Operating Systems Principles. 2011: 85-100.

[58] Tu S, Kaashoek M F, Madden S, et al. Processing analytical queries over encrypted data [J]. Proceedings of the VLDB Endowment, 2013, 6(5): 289-300.

[59] Agrawal R, Kiernan J, Srikant R, et al. Order preserving encryption for numeric data[C]//Proceedings of the 2004 ACM International Conference on Management of Data. 2004: 563-574.

[60] Bajaj S, Sion R. TrustedDB: A trusted hardware-based database with privacy and data confidentiality[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 26(3): 752-765.

[61] Arasu A, Blanas S, Eguro K, et al. Secure database-as-a-service with cipherbase[C]//Proceedings of the 2013 ACM International Conference on Management of Data. 2013: 1033-1036.

- [62] Vinayagamurthy D, Gribov A, Gorbunov S. StealthDB: a Scalable Encrypted Database with Full SQL Query Support[J]. Proceedings on Privacy Enhancing Technologies, 2019, 2019(3): 370-388.
- [63] Shan Z, Ren K, Blanton M, et al. Practical secure computation outsourcing: A survey[J]. ACM Computing Surveys, 2018, 51(2): 1-40.
- [64] Ryan M D. Cloud computing security: The scientific challenge, and a survey of solutions[J]. Journal of Systems and Software, 2013, 86(9): 2263-2268.
- [65] Gampala V, Malempati S. A study on privacy preserving searching approaches on encrypted data and open challenging issues in cloud computing[J]. International Journal of Computer Science and Information Security, 2016, 14(12): 294.
- [66] Orr L, Balazinska M, Suci D. EntropyDB: a probabilistic approach to approximate query processing[J]. The VLDB Journal, 2020, 29(1): 539-567.
- [67] Wang C, Bater J, Nayak K, et al. DP-Sync: Hiding Update Patterns in Secure Outsourced Databases with Differential Privacy[C]//Proceedings of the 2021 International Conference on Management of Data. 2021: 1892-1905.
- [68] Yao A C . Protocols for secure computations[C]// Proceedings of the 1982 Annual IEEE Symposium on Foundations of Computer Science. 1982: 160-164.
- [69] Bater J, Elliott G, Eggen C, et al. SMCQL: secure query processing for private data networks[J]. Proceedings of VLDB Endowment, 2017, 10(6):673-684.
- [70] Volgushev N, Schwarzkopf M, Getchell B, et al. Conclave: secure multi-party computation on big data[C]//Proceedings of the 2019 EuroSys Conference. 2019: 3:1-3:18.
- [71] Bater J , He X , Ehrich W , et al. Shrinkwrap: efficient SQL query processing in differentially private data federations[J]. Proceedings of the VLDB Endowment, 2018, 12(3):307-320.
- [72] Bater J, Park Y, He X, et al. SAQE: Practical Privacy-Preserving Approximate Query Processing for Data Federations[J]. Proceedings of the VLDB Endowment, 2020, 13(12):2691-2705.
- [73] Roy Chowdhury A, Wang C, He X, et al. Cryptc: Crypto-assisted differential privacy on untrusted servers[C]// Proceedings of 2020 ACM International Conference on Management of Data. 2020: 603-619.
- [74] Wang Y, Yi K. Secure Yannakakis: Join-Aggregate Queries over Private Data[C]//Proceedings of the 2021 ACM International Conference on Management of Data. 2021: 1969-1981.
- [75] Ge C, Ilyas I F, Kerschbaum F. Secure multi-party functional dependency discovery[J]. Proceedings of the VLDB Endowment, 2019, 13(2): 184-196.
- [76] Evans D, Kolesnikov V, Rosulek M. A pragmatic introduction to secure multi-party computation[J]. Foundations and Trends® in Privacy and Security, 2017, 2(2-3).
- [77] Liu C, Wang X S, Nayak K, et al. Oblivm: A programming framework for secure computation[C]//Proceedings of the 2015 Symposium on Security and Privacy. 2015: 359-376.
- [78] Zahur S, Evans D. Obliv-C: A Language for Extensible Data-Oblivious Computation[J]. IACR Cryptol. ePrint Arch., 2015, 2015: 1153.
- [79] Keller M. MP-SPDZ: A versatile framework for multi-party computation[C]//Proceedings of the 2020 Conference on Computer and Communications Security. 2020: 1575-1590.

- [80] 蒋瀚, 徐秋亮. 基于云计算服务的安全多方计算[J]. 计算机研究与发展, 2016, 53(10): 2152-2162.
- [81] Cao Y, Fan W, Wang Y, et al. Querying shared data with security heterogeneity[C]//Proceedings of the 2020 ACM International Conference on Management of Data. 2020: 575-585.
- [82] Alves T. Trustzone: Integrated hardware and software security[J]. White paper, 2004.
- [83] Costan V, Devadas S. Intel SGX explained[J]. IACR Cryptol. ePrint Arch., 2016, 2016(86): 1-118.
- [84] He X, Rogers J, Bater J, et al. Practical Security and Privacy for Database Systems[C]//Proceedings of the 2021 ACM International Conference on Management of Data. 2021: 2839-2845.
- [85] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. Decentralized Business Review, 2008: 21260.
- [86] 焦通, 申德荣, 聂铁铮, 等. 区块链数据库: 一种可查询且防篡改的数据库[J]. 软件学报, 2019, 30(9): 2671-2685.
- [87] Dinh A, Wang J, Wang S, et al. UStore: a distributed storage with rich semantics[J]. arXiv preprint arXiv:1702.02799, 2017.
- [88] Nathan S, Govindarajan C, Saraf A, et al. Blockchain meets database: design and implementation of a blockchain relational database[J]. Proceedings of the VLDB Endowment, 2019, 12(11): 1539-1552.
- [89] El-Hindi M, Heyden M, Binnig C, et al. Blockchaindb-towards a shared database on blockchains[C]//Proceedings of the 2019 ACM International Conference on Management of Data. 2019: 1905-1908.
- [90] Wang S, Dinh T T A, Lin Q, et al. Forkbase: an efficient storage engine for blockchain and forkable applications[J]. Proceedings of the VLDB Endowment, 2018, 11(10): 1137-1150.
- [91] Sharma A, Schuhknecht F M, Agrawal D, et al. Blurring the lines between blockchains and database systems: the case of hyperledger fabric[C]//Proceedings of the 2019 ACM International Conference on Management of Data. 2019: 105-122.
- [92] Rouhani S, Deters R. Blockchain based access control systems: State of the art and challenges[C]// Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence. 2019: 423-428.
- [93] Ruan P, Chen G, Dinh T T A, et al. Fine-grained, secure and efficient data provenance on blockchain systems[J]. Proceedings of the VLDB Endowment, 2019, 12(9): 975-988.
- [94] Ruan P, Dinh T T A, Lin Q, et al. LineageChain: a fine-grained, secure and efficient data provenance system for blockchains[J]. The VLDB Journal, 2021, 30(1): 3-24.
- [95] Wang H, Xu C, Zhang C, et al. vChain: A Blockchain System Ensuring Query Integrity[C]//Proceedings of the 2020 ACM International Conference on Management of Data. 2020: 2693-2696.
- [96] Amiri M J, Agrawal D, Abbadi A E. Caper: a cross-application permissioned blockchain[J]. Proceedings of the VLDB Endowment, 2019, 12(11): 1385-1398.
- [97] Dubovitskaya A, Novotny P, Thiebes S, et al. Intelligent Health Care Data Management Using Blockchain: Current Limitation and Future Research Agenda[C]// Proceedings of the 2019 VLDB Workshops. 2019: 277-288.

- [98] 张志威, 王国仁, 徐建良, 等. 区块链的数据管理技术综述[J]. 软件学报, 2020, 31(09):2903-2925.
- [99] Peng Y, Du M, Li F, et al. FalconDB: Blockchain-based collaborative database[C]//Proceedings of the 2020 ACM International Conference on Management of Data. 2020: 637-652.
- [100] Buchnik Y, Friedman R. FireLedger: a high throughput blockchain consensus protocol[J]. Proceedings of the VLDB Endowment, 2020, 13(9): 1525-1539.
- [101] Gupta S, Hellings J, Rahnema S, et al. Building high throughput permissioned blockchain fabrics: challenges and opportunities[J]. Proceedings of the VLDB Endowment, 2020, 13(12): 3441-3444.
- [102] Amiri M J, Agrawal D, El Abbadi A. Sharper: Sharding permissioned blockchains over network clusters[C]//Proceedings of the 2021 ACM International Conference on Management of Data. 2021: 76-88.
- [103] Sweeney L. k-anonymity: A model for protecting privacy[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(05): 557-570..
- [104] Machanavajjhala A, Kifer D, Gehrke J, et al. l-diversity: Privacy beyond k-anonymity[J]. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1): 44-95.
- [105] Li N, Li T, Venkatasubramanian S. t-closeness: Privacy beyond k-anonymity and l-diversity[C]// Proceedings of the 2020 IEEE International Conference on Data Engineering. 2007: 106-115.
- [106] Dwork C. Differential privacy[C]//Proceedings of the 2006 International Colloquium on Automata, Languages, and Programming. 2006: 1-12.
- [107] Wang S, Qian Y, Du J, et al. Set-valued data publication with local privacy: tight error bounds and efficient mechanisms[J]. Proceedings of the VLDB Endowment, 2020, 13(8): 1234-1247.
- [108] Liu J, Lou J, Liu J, et al. Dealer: an end-to-end model marketplace with differential privacy[J]. Proceedings of the VLDB Endowment, 2021, 14(6): 957-969.
- [109] Xiao X, Wang G, Gehrke J. Differential privacy via wavelet transforms[J]. IEEE Transactions on knowledge and data engineering, 2010, 23(8): 1200-1214.
- [110] 叶青青, 孟小峰, 朱敏杰, 等. 本地化差分隐私研究综述[J]. 软件学报, 2018, 29(07):1984-2005.
- [111] Li Z, Wang T, Lopuhaä-Zwakenberg M, et al. Estimating numerical distributions under local differential privacy[C]//Proceedings of the 2020 ACM International Conference on Management of Data. 2020: 621-635.
- [112] Chen R, Li H, Qin A K, et al. Private spatial data aggregation in the local setting[C]//Proceedings of the 2016 IEEE International Conference on Data Engineering. 2016: 289-300.
- [113] Wang Y, Tong Y, D. Federated latent Dirichlet allocation: A local differential privacy based framework[C]//Proceedings of the 2020 AAAI Conference on Artificial Intelligence. 2020, 34(04): 6283-6290.
- [114] Cormode G, Jha S, Kulkarni T, et al. Privacy at scale: Local differential privacy in practice[C]//Proceedings of the 2018 International Conference on Management of Data. 2018: 1655-1658.

- [115] Ghazi B, Golowich N, Kumar R, et al. On the Power of Multiple Anonymous Messages: Frequency Estimation and Selection in the Shuffle Model of Differential Privacy[C]//Proceedings of the 2021 Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2021: 463-488.
- [116] Xu M, Ding B, Wang T, et al. Collecting and analyzing data jointly from multiple services under local differential privacy[J]. Proceedings of the VLDB Endowment, 2020, 13(12): 2760-2772.
- [117] Ye Q, Hu H, Li N, et al. Beyond Value Perturbation: Local Differential Privacy in the Temporal Setting[C]//Proceedings of the 2021 IEEE Conference on Computer Communications. 2021: 1-10.