

DBSphere: 面向云环境的多数据库统一管理框架

冯永设, 陶捷, 王宝晗, 陈敬, 吴奕飞
(中移(苏州)软件技术有限公司, 江苏苏州 215009)

摘要:随着云计算技术的快速发展和广泛应用,云服务提供商正逐渐成为数据库技术创新与维护的主导者。面对多数据库的统一管理的复杂性挑战,本文介绍了DBSphere,一款基于云原生技术的多数据库引擎管理的框架,能够高效地管理并优化数十种数据库引擎。本文深入剖析了云计算环境中异构数据库的管理的挑战,还探讨了云原生技术如何促进数据库管理向自动化与智能化的演进。此外,还提出了开放数据库服务模型(Open Database Service Model, ODSM),旨在实现云环境中不同数据库管理的标准化。DBSphere通过明晰的职责分配与高效的内部通信机制,将复杂的数据库管理任务拆分成一系列标准化作业单元,借助云原生技术显著提升了数据库生命周期管理和运维效率,为云环境中多数据库高效和统一管理提供了创新和实用的解决方案。

关键词:云原生技术;多数据库管理;开放数据库服务模型;云基础设施;数据库即服务

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2025)03-0069-07

doi:10.20165/j.cnki.ISSN1673-629X.2024.0343

DBSphere: A Unified Multi-database Management Framework for Cloud Environment

FENG Yong-she, TAO Jie, WANG Bao-han, CHEN Jing, WU Yan-fei
(China Mobile (Suzhou) Software Technology Co., Ltd., Suzhou 215009, China)

Abstract: With the rapid development and widespread application of cloud computing technology, cloud service providers are gradually becoming the leaders in the innovation and maintenance of database technology. Faced with the complexity of unified management of multiple databases, we introduce DBSphere, a cloud-native multi-database engine management framework that can efficiently manage and optimize dozens of database engines. We deeply analyze the challenges of managing heterogeneous databases in the cloud computing environment and explore how cloud-native technology promotes the evolution of database management towards automation and intelligence. In addition, we also propose the Open Database Service Model (ODSM), aiming to achieve standardization of different database management in the cloud environment. DBSphere, through clear responsibility assignment and efficient internal communication mechanisms, breaks down complex database management tasks into a series of standardized work units. Leveraging the powerful capabilities of cloud-native technology, DBSphere significantly enhances the lifecycle management and operational efficiency of databases, providing an innovative and practical solution for the efficient and unified management of multiple databases in the cloud environment.

Key words: cloud native; multi-database management; open database service model; cloud infrastructure; database as a service

0 引言

云计算技术作为现代企业IT基础设施的基石,在推动企业数字化转型中发挥着至关重要的作用^[1-2]。数据库作为数据管理的核心,在云计算技术的推动下正经历深刻的变革,云服务提供商在这个变革中扮演着日益重要的角色,不仅提供数据库的开发和维护服务,而且在引领数据库服务的创新和发展方法也发挥着关键作用。从最初传统数据库简单托管,到云厂商

自主研发的全面托管解决方案,再到云原生数据库架构的革新^[3-5],数据库服务化的发展正朝着智能化和多模态数据处理等更高级的领域迈进。

面对数据量的激增和数据类型的多样化,企业需要快速响应业务变化,部署和管理多种数据库类型,这不仅对性能和可靠性提出了更高要求,也对成本效益和资源的优化利用带来了挑战。因此,企业越来越多地采用云厂商提供的数据库产品,以减轻运维负担,提

收稿日期:2024-06-28

修回日期:2024-10-29

基金项目:国家自然科学基金企业创新发展联合基金项目(U22B2020)

作者简介:冯永设(1987-),男,硕士,通信作者,研究方向为数据库内核架构、大数据和机器学习。

高业务迭代速度。

针对云环境下多类型数据库的集成、大规模数据库实例的管理、异构数据库的运维复杂性以及成本控制,云服务商亟需一个统一、高效且灵活的数据库管理平台^[6-8],平台需要支持数据库引擎的快速集成、简化运维、动态资源调配和保证数据库的安全隔离,同时促进企业在应用开发、数据分析和数据集成方面的灵活性和创新力^[9-11]。

目前主流云厂商及开源的多数据库管理采用“竖井式”架构,每种数据库都有独立的团队负责开发和运维,存在团队之间协作不足、管理复杂性高且成本高

等问题,已经不适应现代云环境的需求。DBSphere 正是在这样的需求下诞生,它是一个由云原生技术驱动的完全托管的多数据库管理框架,不仅能够实现多种数据库引擎的快速部署和配置,还能自动化地执行数据库备份、恢复、管理等任务,并支持数据库资源的动态调整,极大简化了多数据库环境的管理,提高了运维效率。作为一个通用的数据库管理平台,DBSphere 抽象了数据库实例的组件和运维操作,隐藏了云环境中的数据库的部署、运维和扩展的复杂性,使用户可以通过统一的界面轻松管理不同类型的数据库实例,实现资源的优化配置和加速业务创新,如图 1 所示。

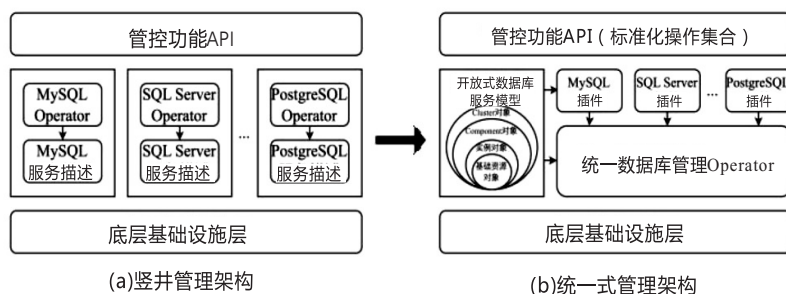


图 1 多数据库管理演进

该文深入探讨了 DBSphere 的构建理念,这一理念以云原生技术为核心,旨在云端环境中以经济高效的方式提供稳定且可靠的多数据库服务,并针对性地解决了以下难题。

(1) 数据库统一描述模型的欠缺。面对不同数据库引擎的集成与维护成本高的问题,通过引入开放数据库服务模型(ODSM),实现了数据库的描述和管理标准化,有效降低了成本。

(2) 操作指令不一致性。为了解决因操作指令不统一而造成的管理复杂性与系统的通用性问题,在框架中定义了一套标准化的操作集,以实现数据库的统一管理,简化运维流程并增强系统的通用性。

(3) 云基础设施的整合挑战。通过深度集成云原生技术,并采用云存储和网络等标准接口,实现了与云基础设施的无缝对接。这种整合不仅提升了数据库服务的弹性和自动化水平,而且优化了资源的使用和成本控制。

(4) 在智能化水平方面,利用最新的 AI 技术,如深度强化学习和大语言模型,构建了大模型驱动的多数据库智能运维平台。

基于大量的文献调研与丰富的大规模数据库的运维经验,全面阐述了如何使用云原生技术打造一个高效、统一的多数据库管理框架。该文的贡献包括深入介绍了 DBSphere 的系统架构和实现机制,展示了云原生技术在多数据库的统一高效管理中的应用范式;阐述了数据库服务的抽象概念,以及开放数据库服务模

型(ODSM)如何实现不同数据库引擎的集成与管理,简化运维流程;深入探讨了数据库管理技术中的关键难点,并提出相应的解决策略,为数据库管理技术的未来发展提供理论和实践指导。

1 总体架构设计

在本章中,将深入阐述云原生数据库管理框架 DBSphere 的设计目标和高层架构,致力于通过服务模块化、数据库引擎插件化、运维策略和能力灵活配置等设计构建一个高效、灵活且用户友好的数据库基础运行环境,其核心设计目标如下:

(1) 多数据库兼容性,确保系统能够无缝支持包括关系型、NoSQL、时序、图数据等在内的多种数据库类型,以满足不同业务场景的需求。

(2) 简化和标准化的运维管理,标准化和自动化数据库的运维管理任务,并利用集中管理界面和自动化工具进一步降低管理的复杂性。

(3) 动态的弹性资源管理,根据业务的负载的变化,实现数据库资源的即时分配与调整,优化资源利用率,提升成本效益。

(4) 增强高可用性和灾难恢复能力,抽象了数据库的高可用架构,确保数据服务的连续性和稳定性,同时提供有效的灾难恢复解决方案。

(5) 集成云基础资源的管理,整合云存储、监控、日志管理等云计算资源,构建统一的基础设施层,以支持数据库服务的高效运行。

(6)系统架构的开放性和扩展性,打造开放的系统架构,支持与第三方工具和服务的集成,便于系统的扩展和定制,满足未来业务发展的需求。

通过达成这些目标,DBSphere 将为云服务提供商提供一个功能强大的云原生数据库管理平台,显著提升在数据库服务领域的核心竞争力。在接下来的章节中,将深入详细探讨 DBSphere 的系统架构、关键功能及其实现细节,展现这些设计如何共同作用实现更高

效、更智能的数据库管理。

1.1 系统架构

DBSphere 为云计算环境提供一个先进的云原生多数据库管理框架,其系统架构从底层基础设施到用户交互精心构建,确保服务之间的高效协同,从下到上依次为云计算基础设施层、底层基础资源接口层、数据库服务管理面、数据库服务数据面、可观测性与管理工具等基础服务和数据库服务管理接口等,如图 2 所示。

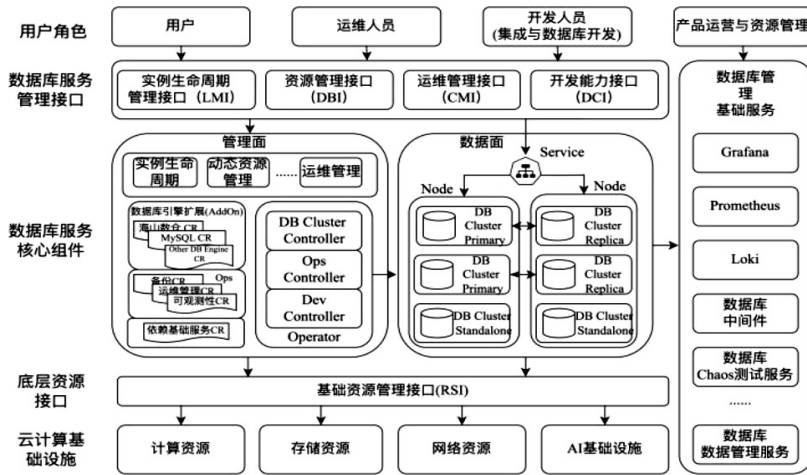


图 2 DBSphere 系统架构

云计算基础设施层作为支撑根基,集成云计算的计算、存储与网络等资源,利用云特性实现资源的弹性供给,并内置了可观测组件,并通过标准化接口简化上层应用对云计算的访问。数据库服务管理面和数据面共同协作完成数据库负责实例全周期、资源管理能力和运维支撑等功能。可观测性与管理供给等基础服务模块集成监控和日志分析等,提供系统可观测性并支持智能运维。

这一设计促进了层间紧密协作与信息流畅,将复杂管理活动标准化、自动化,极大提升了管理效率与灵活性,为用户带来简化高效的服务体验,同时优化成本并加速业务响应。

1.2 运行机制

DBSphere 是基于云原生技术的数据库管理框架,通过结合声明式 API 与 Operator 模式,提供数据库的高效和自动化管理与维护解决方案。它允许用户通过控制台来声明数据库的期望状态,涵盖集群实例的资源需求、配置参数和软件版本等关键信息等。

在这个过程中 Operator 作为核心组件,持续监听由用户通过声明式 API 创建的自定义资源变化,调用 Kubernetes 来执行相应的数据库声明周期管理任务,包括创建、更新、扩容或者删除等。数据库管理流程如图 3 所示。

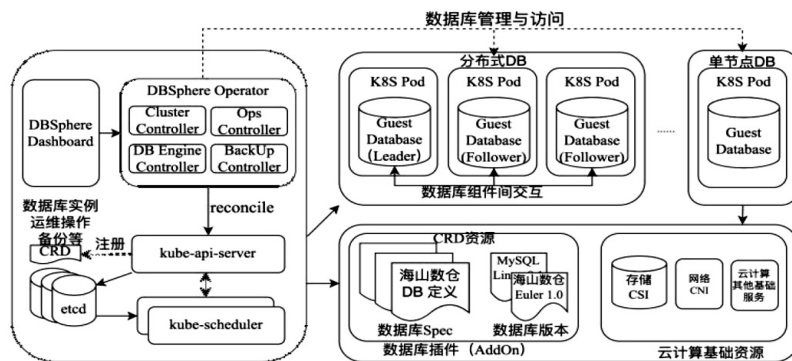


图 3 数据库管理流程

为了支持多数据的管理,DBSphere 实现了灵活的插件机制。不同的数据库引擎被定义为遵循统一规范

的 YAML 文件,这些文件包含数据库引擎的详细模型、配置信息和版本等,使得平台能够以统一和标准化

的方式进行管理。

1.3 数据库引擎的集成

目前 DBSphere 中已经支持了包括 MySQL、

Redis、MongoDB、海山 PG、海山数仓等主流数据库产品,得益于 ODSM 模型的规范定义,集成新的数据库引擎的过程简洁高效,集成过程如图 4 所示。

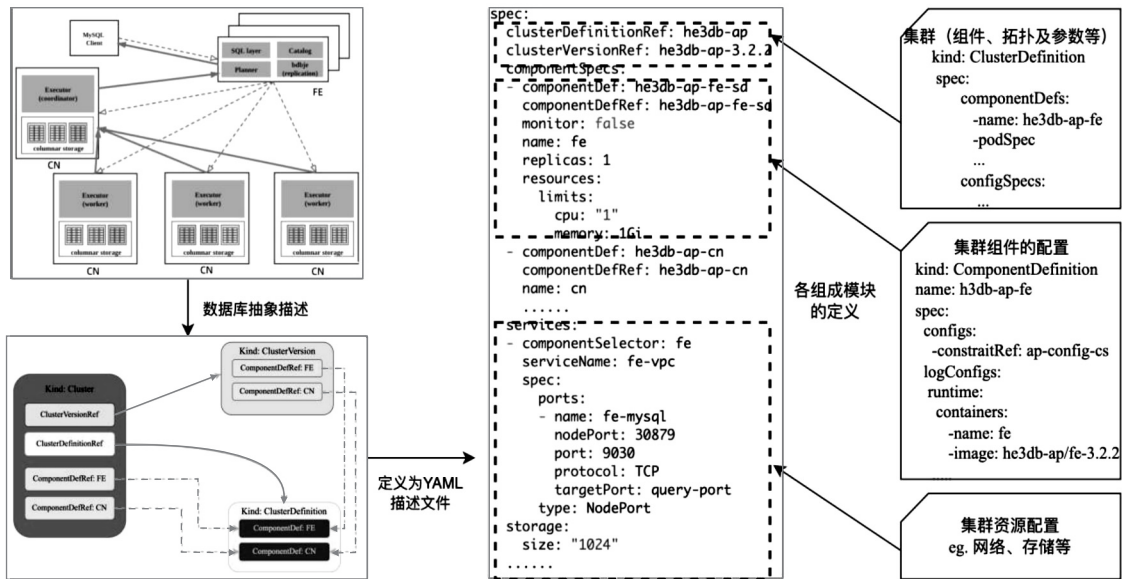


图 4 移动云海山数仓引擎集成示意图

其中海山数仓是一款自研国产数据库,由前端 (FE) 和后端 (BE) 构成,分别负责元数据管理、SQL 请求接收与执行以及数据存储与 SQL 执行。海山数仓通过定义集群信息、版本信息和资源信息来集成,确保数据库系统的部署、运行和性能优化。

在 Kubernetes 中 Operator 会根据海山数仓的资源定义创建相应的工作负载来启动 FE 和 BE 组件,每个组件实例对应一个 Pod。

海山数仓实例启动后与物理机上部署的数仓实例进行性能对比,在测试中使用相同的物理资源,数据集使用 SSB 基准测试,通过 dop 参数来设置运行任务中执行 Pipeline 的并行数量,测试结果如图 5 所示。

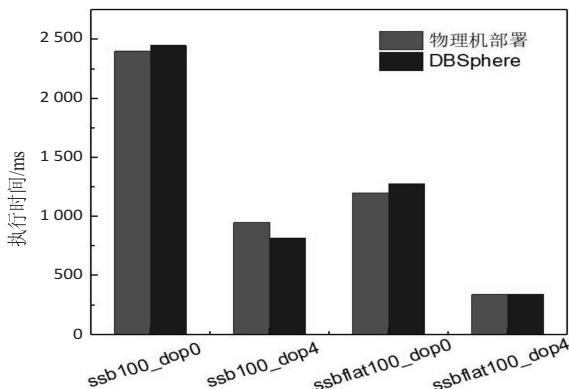


图 5 性能评估

从测试结果可以看出,使用 DBSphere 部署的海山数仓实例在整体性能上与直接在物理机上部署的实例并无显著差距,二者表现出近乎相同的查询效率,表明 DBSphere 部署数仓实例没有明显的性能损失。

2 云原生模块设计

数据库技术的发展历经从单一节点架构演变为分布式系统,从传统的关系型数据库发展到云原生数据库服务,同时数据模型也从单一类型扩展到包括文档、图形、键值存储等在内的多元化新型模型,构建了多样的技术生态。但无论数据库技术如何变革,核心功能与特性始终保持共性,包括数据模型、查询语言、事务一致性、数据访问控制与安全、备份与恢复、性能监测与调优等。这些功能和特性的趋同性为数据库即服务的平台构建提供了可能性,可以通过对这些共性能力进行抽象化处理,以实现统一、高效且兼容多种数据库引擎的管理服务。该文提出了开放数据库服务模型 (Open Database Service Model, ODSM),支持异构数据库无缝接入,标准的操作集合实现数据库的稳定、高效及安全的运行,同时屏蔽掉基础设施的复杂性和差异性。ODSM 模型构成如图 6 所示。

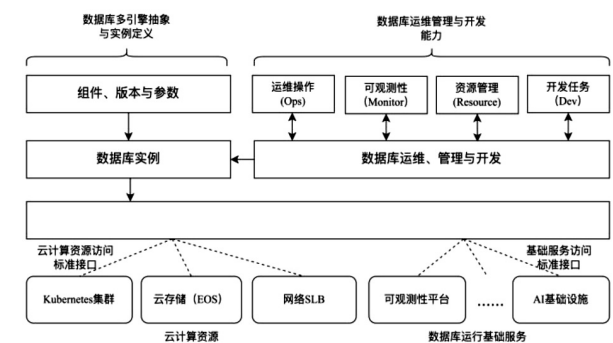


图 6 ODSM 模型构成

ODSM 模型是构建在云基础设施之上,支持从多数据库实例生命周期管理、运维治理、资源管理到开发等过程统一抽象,屏蔽了底层相关的资源定义,降低运维和开发人员的成本,可以将这些功能的抽象围绕数据库的开发者、运维人员、资源管理者、基础设施管理

员等角色职责进行组合和拆分。

2.1 数据库引擎抽象

采用插件机制来扩展不同类型的数据库,通过抽象后形成了如图 7 所示的定义文件。

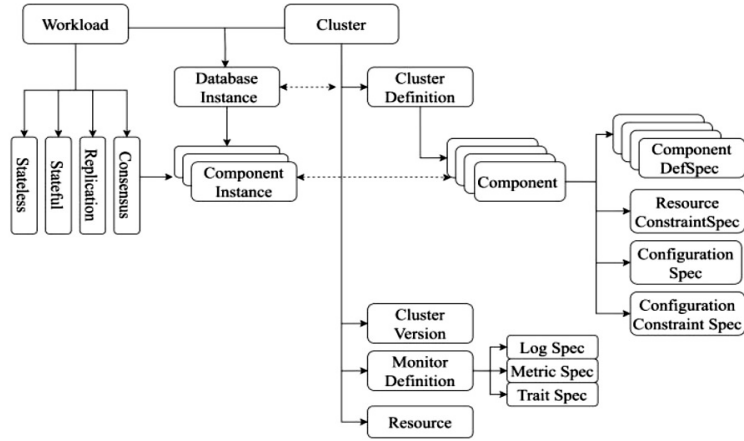


图 7 ODSM 规范示意图

数据库实例由 Cluster 来描述,每个集群的定义包括组件、参数、版本等核心信息。由于数据库是分布式系统中较为复杂的部分,其拓扑结构可以是单节点、主从复制或分布式集群等多种形式,在这些系统中,组件之间需要进行交互,如数据的主从复制、分片及一致性维护等过程,这交互逻辑在 Workload 中进行详细定义和控制。

2.2 操作集空间

通过对多种数据库运维管理进行深度分析和抽

象,可以为通用的运维功能构建操作集合,见表 1。

数据库备份及恢复、版本升级、动态扩缩容、故障切换及配置调整等核心运维功能均可以被规范化为具体的 OpsRequest 对象,在这个对象中会定义相关参数,例如根据预设的规则自动扩容数据库实例,自动备份功能可以定期备份数据库,以防止数据丢失,自动监控功能可以实时监测数据库的运行状态,及时发现并解决潜在的问题等。

表 1 数据库操作集

运维操作	描述
Upgrade	对数据库实例进行版本升级或补丁应用,以获取新功能或安全更新
HorizontalScaling	增加或减少数据库节点以扩展或缩减集群的水平容量
VolumeExpansion	调整数据库存储卷的大小以适应数据增长
VerticalScaling	调整单个数据库节点的资源配置,如 CPU 或内存,以提升性能
Backup	创建数据库的备份副本,以便于数据恢复和灾难恢复
Restore	从备份中恢复数据库,用于数据恢复或迁移
Reconfiguring	重新配置数据库实例的设置,以适应变化的业务需求或优化性能
Start	启动数据库实例
Stop	停止数据库实例
Restart	重启数据库实例,可以用于应用更改或重启服务
Custom	允许用户定义特定的运维操作或脚本以满足个性化需求

2.3 可观测性定义

日志、度量和分布式追踪作为系统健康运行、快速定位问题、优化性能和提升服务质量的核心,在模型中定义了可观测性基础服务的集成与管理规范,在数据库实例运行时通过声明式的定义方法在定义数据库实例运行过程中应收集哪些信息、如何收集这些信息以

及如何与可观测性基础设施进行交互。

(1)信息类型:明确列出要收集的日志条目、度量指标和追踪上下文,确保关键数据不遗漏。

(2)收集方式:定义数据采集的频率、采样策略、数据源接入点(如端口、API 接口、文件路径等),以及可能需要的认证和授权信息。

(3)基础设施对接:指定数据应发送至可观测平台,例如 Prometheus、Loki 集群、Jaeger 代理或 ELK Stack 的相应组件的连接信息,包括必要的传输协议、地址、端口等连接参数。

这种标准化、声明式的配置方法极大地简化了运维工作,增强了系统的可扩展性和一致性,使得整个组织内的可观测性实践能够保持高效且统一。

2.4 云基础资源的管理

DBSphere 通过云基础资源的统一管理和维护,为数据库提供一个高效、集成的运行环境,主要的资源包括以下几种:

(1)云存储的集成,云存储为数据库提供了可扩展性和持久性的数据存储方案,支持计算与存储分离架构的云原生数据库的数据存储、自动数据备份任务中定期备份的数据和快照等,目前支持 S3 标准接口的存储访问。

(2)网络服务,提供全面的网络配置和管理,包括 VPN、负载均衡和网络安全组等保障了数据的安全性和灾难恢复能力。

(3)弹性计算资源,可以根据数据库工作负载的动态变化和用户的需求自动调整数据库的计算资源,优化了系统的性能和成本效率。

在系统中这些资源以标准版接口的方式提供给数据库,同时管理方式也仅可能规范化,降低了开发和运维的复杂性。

除了以上的基本资源,还提供开发工具如混沌测试、数据库实例之间的互操作、AI 能力等,这些能力可以提升数据库开发效率、简化多数据库实例的运维与集成、增强应用的灵活性和可扩展性,以及实现智能化运维与决策支持,从而降低成本,提升数据管理效率和数据价值。

3 大模型驱动的智能运维

传统数据库的运维面临着扩展性的局限性、手动维护运维规则、难以充分利用专家经验和领域知识、人机交互不友好等问题。针对这些问题,DBSphere 构建了大模型驱动的智能运维机制,涵盖了问题溯源、根因分析、优化方案推荐等端到端异常诊断,提高了数据库运维自动化和智能化。智能运维的整体实现如图 8 所示。

该机制依托了中国移动自研九天大模型,该模型是首个通过国家双备案的央企自研大模型,为构建多专家模型协同诊断机制提供了有力支撑。

面对单个大模型在处理多因素复杂异常时的局限性,DBSphere 实现了基于大模型的 Multi-Agent 协同诊断机制,主管专家根据异常问题与多个子领域专家

进行桥接,子领域专家之间通过编排调度引擎进行互相对话和调用来获取该问题的相关上下文信息,通过这种方式实现了多个大模型之间协同诊断、交叉评审的迭代细化分析能力,以此来提高诊断的准确性和性能。

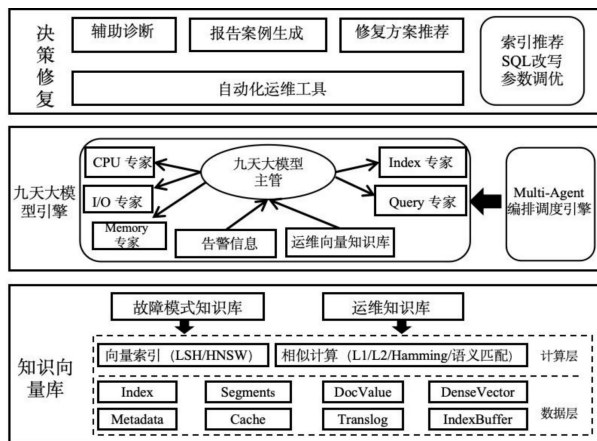


图 8 大模型驱动的智能运维架构

同时 DBSphere 提供了异常诊断机制运行所需要的基础设施的轻松启动和管理,例如可观测性服务、向量数据库、LLM 运行环境等;最后 DBSphere 提供了根据异常诊断进行自动化问题修复需要的标准化操作集,例如参数调优、索引设置等,可以适应不断变化的运维需求。

目前智能运维系统覆盖了移动云 28 个资源池及逾万数据库实例,实现实时自动巡检、服务调度及稳定性监控等能力,确保了异常的快速响应能力,同时还加强数据库的安全监控,提升数据的安全性,为运维管理提供全面的保障。

4 相关工作

数据库技术的演进已从传统孤立的本地部署系统向高度集成的云端服务的转变,这一变革催生了新的数据库产品和服务,同时也让多元数据库的需求和管理技术成为学术界和产业实践的焦点话题^[12]。

云服务商如 AWS、Microsoft Azure、Google Cloud Platform^[13-14]以及国内的阿里云、腾讯云和华为云等都提供多样化的数据库服务,利用云平台的弹性和可扩展性增强服务能力和市场竞争力。然而这些服务往往以单一或者单类型的数据库为中心形成了竖井式架构,限制了集成和能力复用。

开源社区在这个领域也发挥重要的作用,以 OpenStack Trove 作为代表性的开源 DBaaS 解决方案^[15],支持众多主流的关系型与非关系型数据库,为用户提供了一站式的数据库生命周期管理体验,然而 Trove 底层基于虚拟机的架构限制了资源的动态配置和系统的弹性,且每种数据库都需要定制 GuestAgent,

增加了集成难度与运维工作量。

Kubernetes 生态系统中的 Operator 模式为数据库管理引入了自动化^[16],通过将特定数据库的运维知识和组件抽象为自定义资源,并通过控制器实现了自动化管理。例如 Crunchy Data 的 PostgreSQL Operator 和 Percona Kubernetes Operator for MySQL 展示了这种模式的强大功能,相对成熟的产品如 QFusion、Nutanix Era 和 KubeDB 等,利用云原生技术实现了资源的动态调整,提高了服务响应速度和故障恢复能力。但现有的产品与云厂商提供的服务都面临相同的问题,数据库引擎和管理服务缺乏统一的实现标准,这导致必须为每种数据库产品单独进行定制开发,同时也限制了数据库之间的互操作性和集成。

为了应对这一挑战,研究的方向需要聚焦于如何打破现有的竖井式架构,推动资源共享和服务集成向更高效、更统一的方向发展。这不仅涉及到技术层面的创新,还需要制定和推广统一的数据库服务管理标准,以简化跨数据库服务的管理和开发流程。

5 结束语

该文着重探讨了云原生架构在多数据库统一管理与优化中的核心价值,云原生架构通过容器化、Serverless、可观测性与自动运维等关键技术,赋予数据库服务前所未有的弹性、自动化与智能化水平。这一变革不仅从根本上革新了数据库的设计理念与运维模式,还为云厂商在云环境中应对多数据库引擎管理的挑战,提升了数据库管理效率、降低了成本,并在保障数据安全与一致性方面发挥了关键作用。

以云环境为背景,该文围绕采用云原生技术构建数据库服务基础管理设施展开论述,提出了开发数据库服务模型(ODSM)。基于此模型构建了移动云云原生数据库管理框架 DBSphere,在文章中介绍了系统设计、实现细节与实际应用效果,为构建高效、统一且适应云环境的数据库管理平台提供了理论依据与实践指导。同时将数据库服务管理范畴延伸到了数据库可观测性基础设施、多数据库管理工具等方面。

然而在数据库安全性、数据库与 AI 协同、多数据库互操作与集成服务等领域仍存在诸多挑战,有待进一步研究与实践。这些问题的解决将有助于提升数据库基础设施的安全性、智能化水平和互操作性,更好地服务于多元化的数据管理需求。

总之,通过深入剖析云原生架构赋能数据库服务统一管理的核心价值,探讨技术趋势与未来研究方向,为云原生数据库基础设施的研究与实践指明了发展路线与方向,为数据库管理的智能化、统一化与高效化提

供了重要指导。

参考文献:

- [1] XUE C T S, XIN F T W. Benefits and challenges of the adoption of cloud computing in business [J]. *International Journal on Cloud Computing: Services and Architecture*, 2016, 6(6): 1-15.
- [2] 中国信息通信研究院. 云计算发展白皮书[EB/OL]. 2024-07. <http://www.caict.ac.cn/kxyj/qwfb/bps/202407/P020240723334151523502.pdf>.
- [3] DB-Engines Ranking[EB/OL]. 2024-06. <https://db-engines.com/en/ranking>.
- [4] LI Guoliang, DONG Haowen, ZHANG Chao. Cloud databases: new techniques, challenges, and opportunities [J]. *Proceeding of the VLDB Endowment*, 2022, 15(12): 3758-3761.
- [5] 叶志伟, 蔡敦浓, 钱岭. 计算存储分离云原生关系型数据库技术发展[J]. *软件导刊*, 2021, 20(8): 236-240.
- [6] AGRAWAL D, ABBADI A E, EMEKCI F, et al. Database management as a service: challenges and opportunities [C]// *IEEE 25th international conference on data engineering*. Shanghai: IEEE, 2009: 1709-1716.
- [7] DEKA G C. A survey of cloud database systems [J]. *IT Professional*, 2014, 16(2): 50-57.
- [8] ZHENG Xi. Database as a service - current issues and its future [J]. *arXiv:1804.00425*, 2018.
- [9] TAN J, GHANEM T, PERRON M, et al. Choosing a cloud DBMS: architectures and tradeoffs [J]. *Proceeding of the VLDB Endow*, 2019, 12(12): 2170-2182.
- [10] Research and Markets, Database-as-a-service (DBaaS): Global strategic [EB/OL]. 2024-09. <http://www.researchandmarkets.com/reports/databaseas-a-service-dbaas-global-strategic>.
- [11] ABADI D, AILAMAKI A, ANDERSNE D, et al. The seattle report on database research [J]. *Communications of the ACM*, 2022, 65(8): 72-79.
- [12] IRINA A, KOSCHEL A, EICKEMEYER C, et al. Evaluating DBaaS Offerings [J]. *International Journal for Information Security Research*, 2017, 17(1): 714-725.
- [13] VIKIRU A, MUIRURI M. An overview on cloud distributed databases for business environments [J]. *arXiv:2301.10673*, 2023.
- [14] Amazon Web Services. Amazon RDS 功能-云关系型数据 [EB/OL]. 2024-06. <https://aws.amazon.com/rds/features/>.
- [15] SHELLEY K A. An introduction to database as a service [C]// *OpenStack Trove*. Berkeley: [s. n.], 2015.
- [16] PERERA H, SILVA T, WASALA W. Database scaling on Kubernetes [C]// *International conference on advancements in computing (ICAC)*. Colombo: [s. n.], 2021: 258-263.