计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202440416
61(11): 2835−2845, 2024

智能物联网技术赋能算网一体数据库的效能优化

朱茂盛[1] 王宝晗[1] 康曼聪[2] 于 巍[1] 杨利超[3]
[1](中移(苏州)软件技术有限公司 江苏苏州 215009)
[2](中国移动有限公司研究院 北京 100053)
[3](苏州大学苏州医学院公共卫生学院 江苏苏州 215000)
(zhumaosheng0712@bupt.cn)

Efficiency Optimization for Computer Network Integrated Database Empowered by Artificial Intelligence of Things Technology

Zhu Maosheng[1], Wang Baohan[1], Kang Mancong[2], Yu Wei[1], and Yang Lichao[3]
[1](*China Mobile* (*Suzhou*) *Software Technology Co.*, *Ltd.*, *Suzhou*, *Jiangsu* 215009)
[2](*China Mobile Communications Research Institute*, *Beijing* 100053)
[3](*School of Public Health*, *Suzhou Medical College of Soochow University*, *Suzhou*, *Jiangsu* 215000)

Abstract The collection, storage and sharing technology of massive perceptual data has promoted the rise of the Internet of things. Its large-scale application has put forward an urgent requirement for database to have strong data consistency and high resource efficiency. However, the existing database system architecture and management and control methods mostly reduce communication costs by reducing the amount of data transmitted and increasing node storage replicas, and lack of integrated awareness and optimization of system network resources, resulting in low database efficiency. For this reason, we construct an integrated computing network database system to drive the integrated perception representation of computing resources and networks, and enables the database with intelligent Internet of things technology (AIoT) to realize the joint intelligent scheduling of computing resources and networks, so as to reduce the overall cost and improve the efficiency of computing networks. First, we construct a distributed database network integrated with computing and network, and analyze its architecture characteristics. Then, in order to realize the unified representation of computing and communication optimization variables, an integrated cost perception model is constructed. On this basis, an intelligent one-stop resource optimization algorithm is proposed to maximize the overall computing efficiency of the database system. Finally, simulation experiments verify the superiority of the proposed architecture and algorithm in network performance, algorithm convergence, integration cost and resource efficiency.

Key words AIoT; computer network integration; distributed database; unified perception; efficiency optimization

摘 要 海量感知数据的采集、存储和共享技术推动了物联网的兴起,其大规模应用对数据库系统提出了数据强一致和资源高效能的迫切要求.然而,现有的数据库系统架构和管控方法在保证一致性的约束下,较多采用降低传输数据量和增加节点存储副本的方式降低通信成本,缺少对系统网络资源的一体化感知和优化,从而导致数据库效能低下.为此,构建算网一体数据库系统,驱动算力资源和网络的一体化感知

收稿日期:2024-05-31;**修回日期**:2024-07-31
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)
This work was supported by the National Natural Science Foundation of China Enterprise Innovation and Development Joint Fund Project (U22B2020).
通信作者:杨利超(2015111098@bupt.cn)

表征，并将智能物联网（artificial intelligence of things, AIoT）技术赋能数据库，实现算力资源和网络的联合智能调度，以降低综合成本和提升算网效能. 首先，构建算网一体的分布式数据库网络，并分析其架构特点. 然后，为实现计算和通信优化变量的统一表征，构建了一体化代价感知模型. 并在此基础上提出了智能化的一站式资源优化算法，满足一致性时延约束的前提下，最优化数据库系统的整体算网效能. 最后，仿真实验验证了所提架构和算法在算网性能、算法收敛、集成代价和资源效率上均具有优越性.

关键词 智能物联网；算网一体；分布式数据库；一体感知；效能优化

中图法分类号 TP393

随着经济社会数字化转型和智能化升级步伐的加快，物联网（Internet of things, IoT）[1]已经成为我国信息基础设施的重要组成部分. 特别是云计算、边缘计算和终端设备之间资源、数据和任务的动态协调，即云边端协同技术[2-3]的快速发展，IoT 得以优化系统性能、降低时延并提升数据处理效率，从而促进其商业化应用. IoT 主要采集物理世界中海量的感知数据，通过物与物之间的通信和管控，实现物物相连、万物互联[4]. 然而，在 IoT 中，数据的产生频率较高、规模量大且维度复杂，导致了数据无法被有效存储、处理和共享. 因此，亟需可拓展和强可靠性的数据库系统[5]，以实现数据的高效纳管. 分布式数据库系统（distributed database system, DDBS）[6]具有可拓展、高可用和架构灵活等特征[7]，能够将数据库的各个部分存储在不同的地点，通过网络连接相互协作，应用路由协议跨节点通信，从而实现应用程序的不断访问和数据节点的持续更新[8].

然而，DDBS 现有的网络架构和管控方法在节点数量增加时效能下降严重，制约了 IoT 的大范围部署与使用[9]. 具体而言，随着 6G 等新一代网络技术的发展，数据库的架构形态从集中式逐渐衍生到分布式，导致节点之间数据同步的通信资源使用增加和全局管控的计算资源占比上升[10]，使得系统算力资源和网络效能低下. 这将降低下游应用在大数据量接入时的每秒查询速率和每秒处理任务数[11]，使得业务的响应时延上升，甚至出现中断的情况. 为此，现有的研究多借助“log as database”[12]的理念，采用轻量级日志传递信息的方法，降低通信数据量. 这种方法需要节点划分出非业务算力资源解析和处理日志信息，并在节点存储数据副本以应对网络异常时的业务恢复和数据一致性保证[13]. 此外，节点之间的全局管控依赖于数据库管理系统[14]的整体纳管. 该系统需要在节点中划分非业务算力以感知信道状态和节点信息，完成全局事物的协调和局部数据转换等控制功能[15-16]. 面对 IoT 中数据感知海量、分布广泛、主体多元的特性[1,4]，DDBS 需通过网络大规模的部署. 但是，现有的数据通信方法和节点管控系统在大规模部署时易受网络波动、流量变化等影响，导致日志丢包率上升，从而造成副本冗余的急剧增加和非业务算力的大量消耗. 因此，亟需一种提升 DDBS 算力资源和网络综合效能的通信和管控方法以适应 IoT 广泛部署时的新需要.

目前，国内外研究人员对数据库中的数据通信和全局管控开展了广泛的研究. 其中，文献 [17] 提出“NF-log”系统，设计日志的写路径并优化写前日志（write ahead log, WAL）机制，以提升非易失性存储资源的效率. 文献 [18] 为了提升分布式日志的性能，提出“Declog”系统，设计并行日志记录方式，实现数据驱动的日志序列编号和松弛排序策略，以跟踪事务依赖并解决序列化问题. 这些工作通过对日志的设计与改造减少通信数据量，提升资源效率和系统性能，但缺少对系统的整体管控和通信资源的优化. 为此，文献 [19] 基于开关无关性和数据结构重塑面向 IoT 设备的数据库管理系统，以应对高频数据插入的挑战. 文献 [9] 聚焦于分布式数据库的能耗、网络和存储资源的管理，以探索云边协同框架下 DDBS 负载卸载的可行性，提升资源效率. 但是这些研究多将网络视为通信的“管道”，缺少对资源的统一表征和联合管理，增加了全局系统日志传输和存储副本等计算和通信资源的开销，限制了异构资源的联合优化效能.

随着人工智能等新兴技术与 IoT 的融合发展，智能物联网（artificial intelligence of things, AIoT）[4]相关技术应运而生. AIoT 应用云边端的高效协同和通感算的一体调度方法，完成智能化的感知与管控，实现更高形式的万物智联. 其中，网络除了作为节点之间通信的“管道”之外，也具有了灵活的资源编排能力[20-21]. AIoT 可以在网络控制平面编排能力的基础上，推进“算力”与“网络”的深度融合[22]，通过算力资源和网络的全面感知，实现算网一体的统一联合调度和按

需智能分配，以提升系统的资源效能. 于是，本文将AIoT技术嵌入到DDBS系统，利用传输工作台及时感知信道状态信息，应用云原生算力控制台打破“算-网”壁垒，一体调度系统计算和通信资源，实现算力资源和网络的全局管控，从而减少网络波动、流量变化时的数据副本和非业务算力开销.

为此，本文将AIoT技术引入算网一体数据库，在算力资源和网络统一感知表征的基础上，提出智能联合优化算法，打破“算-网”隔离，以提升系统算网综合效能. 首先构建算网一体赋能的分布式数据库网络，通过传输控制台和算力控制台实现资源统一感知. 接着建立一体化代价感知模型实现计算和通信优化变量的统一表征. 在此基础上设计智能化的一站式效能优化方法，在保证一致性时延约束下，最大化算网综合效能. 最后，与其他架构和算法相比，实验结果验证了本文工作在网络性能、算法收敛、集成代价和资源效率上的优越性.

1 网络架构

算网一体数据库的基本架构如图1所示. 其中，分布式数据库节点构成集合$D=\{D_1,D_2,\cdots,D_M\}$，节点之间通过网络传递信息. 对于任意第m个节点，其算力开销建模为二元组$f_m=\{\omega_m,\varpi_m\}$，其中，$\omega_m$和$\varpi_m$分别表示业务算力和管控算力. ω_m主要用来处理用户创建、查询数据等具体业务，ϖ_m主要用来满足收集节点状态、存储冗余数据等管控需要.

在数据库节点通过网络传递信息的过程中，节点的自身算力信息通过网关路由器传递到网络的控制平面. 同时，网关路由器还感知节点之间通信的信道状态，并传递到该平面. 控制平面在获取节点算力和信道状态信息后，统一管控和调度相关任务. 控制平面[21]主要包含算网管控和算网大脑2个部分，协同提供编排管理的相关功能，其详细架构如图2所示.

在算网管控模块中，传输工作台通过IF3接口感知网络信道状态，并通过IF1接口将状态信息传递到算网大脑. 类似地，云原生算力控制台通过IF4接口感知节点算力状态，并通过IF2接口将状态信息传递给算网大脑. 算网大脑模块通过网络域能力网关和算力域网关分别获得IF1和IF2接口信息后，对信道和算力资源统一建模与表征. 接着，在轻量预测网络和节点下一时刻负载的基础上，生成对信道和算力资源的统一智能调度与编排管理策略. 该策略分别通过IF1和IF2接口，将网络和算力管控信息传输到相应的工作台，并通过IF3和IF4接口实现信道和算力资源的调度与管理.

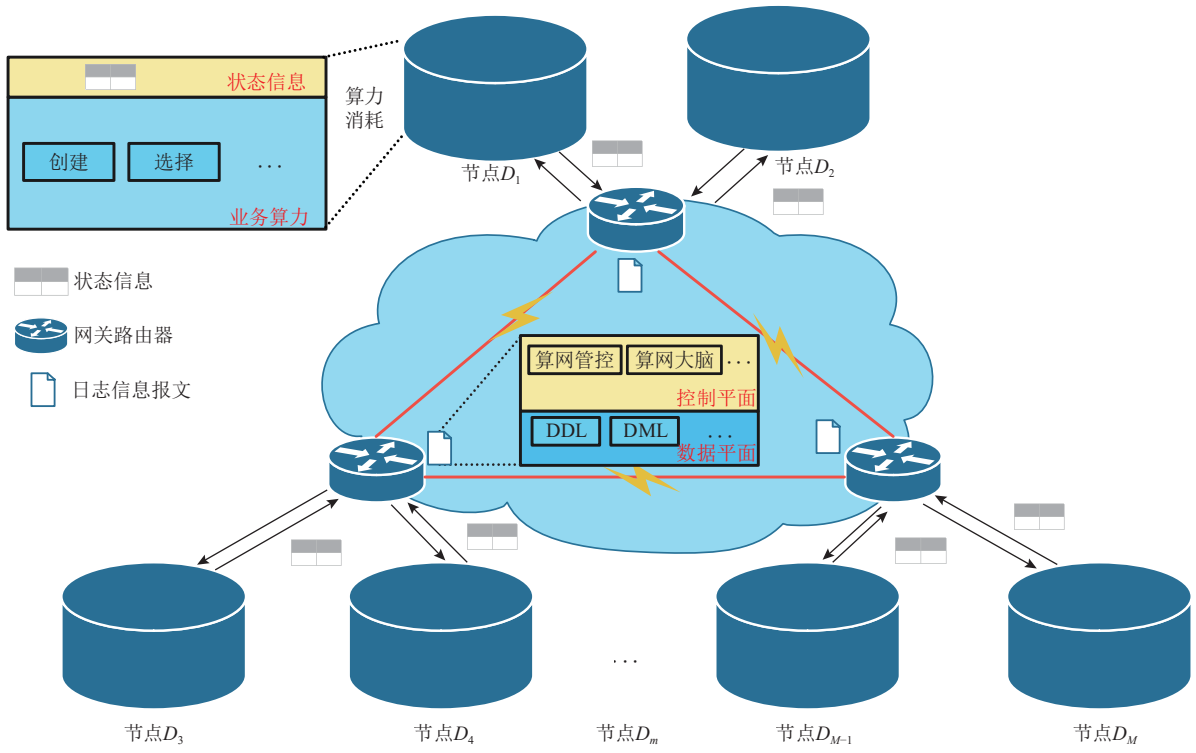


Fig. 1 Basic network diagram of computing and network integrated database

图1 算网一体数据库基本网络图

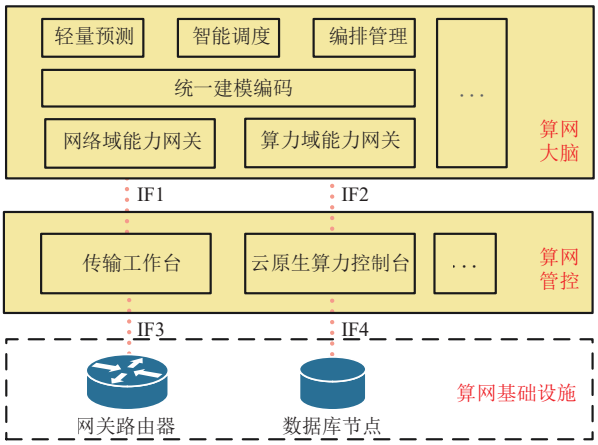


Fig. 2 Detailed architecture of network control plane

图 2 网络控制平面详细架构

与传统架构相比, 网络不仅承担着节点之间的通信, 还负责资源的统一管理与调度. 管控主体从数据库节点转换到网络, 带来了 2 点好处:

1)减轻节点非业务算力开销, 提升系统整体资源效率. 传统架构需要在数据库节点划分算力资源以实现网络信道状态感知、集群节点信息获取. 本文架构中节点只需少量算力维护自身状态信息即可, 降低节点的非业务算力开销, 感知从节点转换到网络. 通过节点状态的 1 次传输和信道状态的 1 次感知, 网络即可获取待调度的资源信息表, 减少非业务信息传输对信道资源的占用, 提升整体算网资源效能.

2)实现多维资源在网调度, 降低系统整体管控成本. 传统架构需要在数据库节点上划分算力资源和部署管理调度机制. 本文架构将调度机制嵌入到网络管控平面的算网大脑中, 实现节点算力和网络信道的统一在网调度. 通过对算力和网络的一体化管控, 减少两者单独管控时变量之间的解耦成本, 及时响应系统变化, 从而降低整理管控成本.

2 问题建模

本文将问题建模为一个优化问题, 旨在满足数据库数据一致性时延约束下, 最大化 DDBS 整体算网效能. 详细建模过程如下.

2.1 “算-网”模型

本文采用的算网一体数据库基本架构如图 1 所示, 一共有 M 个数据库节点. $\forall D_m \in D, 1 \leqslant m \leqslant M$, 其需在网络中传输的日志共有 N 条, 建模为集合 $L_m = \{l_{1,m}, l_{2,m}, \cdots, l_{N,m}\}$, 其中, $1 \leqslant m \leqslant M$. $\forall l_{n,m} \in L_m, 1 \leqslant n \leqslant N$. 其建模为三元组 $l_{m,n} = \{a_{n,m}, b_{n,m}, s_{n,m}\}$, 其中 $a_{n,m}$ 是日志处理时的指令数, $b_{n,m}$ 是数据库节点解析日志后需要执行的业务指令数, $s_{n,m}$ 是该日志的调度策略, $s_{n,m} \in \{s|1,2,\cdots,M\}$, 表示其在对应的数据库节点上处理和解析. 对于第 m 个数据库节点 D_m, 其算力以每秒浮点运算次数(floating-point operation per second, FLOPS)度量, 表示为

$$f_m = N_m \times h_m \times r_m, \tag{1}$$

其中, N_m 是中央处理器(central processing unit, CPU)核数, 表示其内部包含的独立运算单元的数量. h_m 是计算单元的主频, 表示 CPU 每秒可以执行的指令数. r_m 是单周期浮点计算值, 表示 CPU 在每个指令周期内可以执行的浮点运算数量.

此外, 任意 2 个数据库节点 D_j、D_k 之间的数据传输速率用香农公式计算为

$$R_{j,k} = W_{j,k} \mathrm{lb}\left(1 + \frac{p_{j,k} \times G_{j,k}}{\sigma^2}\right), \tag{2}$$

其中, $W_{j,k}$ 是 D_j 和 D_k 之间的信道带宽, $p_{j,k}$ 和 $G_{j,k}$ 分别表示传输功率和信道增益, 两者乘积 $p_{j,k} \times G_{j,k}$ 代表了信号功率. σ 表示加性信道噪声的功率谱密度, σ^2 代表了信道中的噪声功率. 令 W_k 表示数据库节点 D_k 中可以调度的总带宽, S_k 表示同一时刻传递到该节点的日志集合, 即 $S_k = \{l_{n,m}|s_{n,m} = k\}$, 则有如下的带宽约束:

$$\sum_{l_{n,m} \in S_k} W_{j,k} \leqslant W_k. \tag{3}$$

2.2 一致性时延模型

对数据库系统而言, 其数据一致性与恢复时间目标(recovery time object, RTO)紧密相关[23]. 在本文中, RTO 建模为基于概率的日志解析完成时间约束, 包含了传输、执行和等待时延 3 个部分. 其中, 日志 $l_{n,m}$ 传输时延计算为

$$t_{\mathrm{tra}}(l_{n,m}) = \frac{a_{n,m}}{R_{j,k}}, \tag{4}$$

其中, 对于在本地处理的日志而言, 传输时间设置为 $t_{\mathrm{tra}}(l_{n,m}) = 0$. 此外, 对于 $s_{n,m} = i$ 的日志, 即传输到第 i 个数据库节点的日志, 其执行时间计算为

$$t_{\mathrm{pro}}(l_{n,m}) = \frac{a_{n,m} + b_{n,m}}{f_i}. \tag{5}$$

基于排队理论[24], 本文采用 $M/M/1/\beta/\infty$ 模型计算日志在数据库节点上的等待时间, 其中 $\beta = N_m$ 表示第 m 个数据库节点能同时处理的日志总数. 假设日志到达数据库节点的速率符合参数为 ι 的泊松分布, 数据库节点响应日志的服务时间服从参数为 κ 的指数分布. 则平均等待时延计算为

$$t_{\mathrm{wai}}(l_{n,m})=\frac{\Phi}{\kappa(1-\Psi)}, \tag{6}$$

其中，

$$\Phi=\frac{p}{1-p}=\frac{(\beta+1)p^{\beta+1}}{1-p^{\beta+1}}, \tag{7}$$

$$\Psi=\frac{1-p}{1-p^{\beta+1}},\ p=\frac{\iota}{\kappa}, \tag{8}$$

其中Ψ表示该数据库节点中没有日志正在执行时的概率.

综上，第 m 个数据库节点的日志解析完成时间表示为

$$T_m=\sum_{n=1}^{N}t_{\mathrm{tra}}(l_{n,m})+t_{\mathrm{pro}}(l_{n,m})+t_{\mathrm{wai}}(l_{n,m}). \tag{9}$$

令T_m^{r}表示该节点处理日志信息的最大 RTO 时间限，其一致性时延约束建模为：

$$Pr(T_m^{\mathrm{r}}\leqslant T_m)\leqslant\upsilon. \tag{10}$$

这表示日志完成时间必须以极大的概率，即强一致地满足 RTO 的时间约束. 其中，υ是可以容忍的噪声，表示超过 RTO 时间的误差以松弛该约束.

2.3 统一代价模型

本文设计归一化代价模型耦合计算资源和通信资源优化变量，以统一表征. 与前序研究工作[25]类似，每个在 DDBS 内部通信的日志成本与第 m 个节点的单位计算成本α_m及传输信道的单位通信成本β_m有关. 其中，单位计算成本α_m和单位通信成本β_m与该节点的计算资源和通信资源的使用率线性相关. 计算资源的使用率Γ_m表示为已使用的计算资源与节点总计算资源的比值，计算为

$$\Gamma_m=\frac{\sum_{l_{n,m}\in S_m}(a_{n,m}+b_{n,m})}{f_m}, \tag{11}$$

其中，S_m表示调度到第 m 个节点上处理和解析的日志集合. 类似地，通信资源的使用率Λ_m表示为已使用的带宽与节点总带宽之间的比值，计算为

$$\Lambda_m=\frac{\sum_{l_{n,m}\in S_m}W_{n,m}}{W_n}. \tag{12}$$

因此，单位计算资源和通信资源的使用成本分别为：

$$\alpha_m=\lambda\Gamma_m+\gamma_m \tag{13}$$

和

$$\beta_m=\mu\Lambda_m+\varsigma_m, \tag{14}$$

其中，λ和μ分别表示相对于计算资源和通信资源利用率的单位百分比成本转换系数，γ_m和ς_m分别是第 m 个数据库节点的计算资源和通信资源的基础成本. 从而，系统中需要通信的所有日志的总成本建模为

$$C_m^{tl}=\sum_{n=1}^{N}[\alpha_m(a_{n,m}+b_{n,m})+\beta_m a_{n,m}]. \tag{15}$$

综上，为了最大化数据库系统中算力资源和网络的综合效能，本文的优化问题建模为在满足 RTO 的一致性时延约束下，最小化 DDBS 中所有节点的计算资源和通信资源代价，公式表示为

$$\min_{s_{n,m},W_{n,m}}\sum_{m=1}^{M}C_m^{tl},\ \forall D_m\in D, \tag{16}$$

s. t.

$$s_{n,m}\in\{s|1,2,\cdots,M\}, \tag{16a}$$

$$\sum_{l_{n,m}\in S_k}W_{j,k}\leqslant W_k, \tag{16b}$$

$$Pr(T_n^r\leqslant T_m)\leqslant\upsilon, \tag{16c}$$

式(16a)表示日志的调度策略必须在区间 $[1, M]$ 中取整数；式(16b)表示每个节点中可调度带宽不能超过其信道容量；式(16c)表示数据的一致性时延保证，即日志完成时间必须以极大概率满足 RTO 的时间约束.

3 代价感知的智能调度算法

问题(16)是一个 NP-hard 的非线性混合整数规划问题，包含计算资源的整数优化变量$s_{n,m}$和通信资源的连续优化变量$W_{n,m}$. 为此，本文设计了代价感知的智能调度算法(cost- aware intelligent scheduling method, CAIS)，包含代价最小化资源调度算法和一致性时延校验算法 2 部分，分别介绍如下.

3.1 代价最小化资源调度算法

为了使问题(16)能在多项式时间复杂度内可解，本文基于第三代非支配排序遗传算法(non- dominated sorting genetic algorithm-III, NSGA-III)提出以代价最小化为目标的效能优化算法. 作为一种多目标遗传优化算法，NSGA-III[26]适合于低复杂度的多个节点的目标优化，并规避局部最优的情况. 本文优化后的算法步骤如下：

步骤 1: 编码及配置初始化

本文将所有的日志对应到一条染色体上，其中每条染色体需要传输的日志$l_{n,m}$编码为染色体上的一个基因，共有 $N\times M$ 个基因. 其中每个基因的值设置为日志的调度策略$s_{n,m}$，每 N 个基因绑定通信时的带宽$W_{n,m}$. 多条染色体共同组成了一个种群 P. 接着，定义种群规模为P_{o}的种群 A，在 $[1, M]$ 的整数范围内中随

机初始化该种群个体中所有基因的值. 然后将遗传算子应用于种群 A, 包含选择、突变和交叉, 形成一个新的种群 B. 再将这 2 个种群 A, B 结合形成一个新的种群 C, 大小为 P_o.

步骤 2: 快速非支配排序

首先, 定义一个集合 S_p 包含所有被个体 p 支配的其他个体. 然后, 选择那些被个体 p 支配的个体数 n_p 等于 0 的个体形成第一前列面 F_1, 并为这些个体设置 rank 值 $r=1$. 接下来, 对于第 i 个前列面 F_i 中的个体, 找出由这些个体支配的其他个体所组成的集合 S_p, 并将 n_p 减 1. 如果 n_p 降为 0, 则这些个体将构成下一个前列面 F_{i+1}. 通过这种非支配排序方法, 依次选出 rank 值 r 为 1, 2, 3,⋯的个体组成新的种群 D, 直到该种群的规模达到 P_o 时为止.

步骤 3: 种群自适应归一化

首先, 计算所有维度上的优化目标的最小值, 即 C_m^{tl}. 将第 m 个维度, 即第 m 个节点上的最小值记作 z_m. 则其归一化过程可表示为

$$f'_m(c)=f_m(c)-z_m, \tag{17}$$

其中, $f_m(c)$ 表示上一轮循环中 z_m 初始化的取值. 接着, 通过拓展实现函数(achievement scalarizing function, ASF)获取最大值, 计算为

$$ASF(C,H)=\max_{m=1:M}\frac{f'_m(c)}{H_m}, \tag{18}$$

其中, $H=\{H_1,H_2,\cdots,H_M\}$, $\forall H_m\in H$, 初始化为 $H_m=10^{-6}$. 最后通过遍历搜索到 ASF 值最小的染色体, 即极值点. 然后根据这些点的函数值计算出它们在相应坐标轴上的截距, 记为 a_m. 自适应归一化的结果记作 f_n, 其计算为

$$f_n=\frac{f_m(c)}{c_n}. \tag{19}$$

步骤 4: 关联参考点

在前序步骤归一化的基础上, 得到了若干可行解, 记为参考点. 这些参考点按升序逐级下降, 且运用递归方法计算后不需要每代求值. 连接这些参考点即构成了归一化的超级连接平面. 连接参考点与原点构成参考向量. 遍历每个种群的所有参考向量, 以找到每一个种群中与每个个体最接近的参考点.

步骤 5: 筛选下一代种群

经过如上步骤后, 假设从组成的 F_1 第一个前列面累计, 恰好第 g 个前列面 F_g 的染色体总数首次超过种群预定规模 P_o. 首先, 定义 S_g 为包含 F_g 中所有个体的一个集合. 由于 S_g 中染色体的加入使得种群规模超过了预先设定数目, 因此需要进行相应的筛选. 筛选的第一步是遍历每个参考点, 计算其不包含 F_g 的 S_g 引用的次数. 紧接着找出引用个数最少的参考点, 即与最少个体数量相关联的参考点, 将其引用数量记录为 p_j.

如果 $p_j==0$ 为真, 并且在前列面 F_g 中有与这个参考向量相关的个体. 寻找到距离最小的点, 从 F_g 中提取添加到下一代中, 并令 $p_j=p_j+1$; 如果没有个体与前列面 F_g 中的参考点相关联, 则删除该参考向量; 如果 $p_j>0$, 则选出其最近的参考点.

上述 5 个步骤可用图 3 表示.

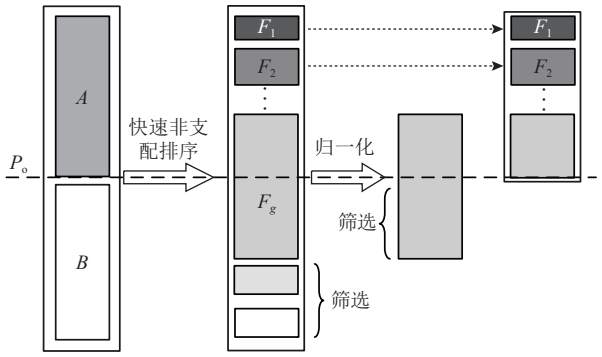


Fig. 3 Schematic diagram of algorithm 1 steps

图 3 算法 1 步骤示意图

在初始化规模为 P_o 的种群 A 之后, 利用交叉、变异等遗传算子生成相同规模的种群 B. 接着, 利用快速非支配排序生成若干个前列面 $F_1,F_2,\cdots,F_g,\cdots$, 对于刚好超过了种群规模 P_o 的前列面 F_g, 计算个体的参考点大小后筛选掉多余的部分, 最后形成了下一代种群.

不断循环步骤 1~5 直到达到预先设置的迭代次数 I_0. 其算法的伪代码如算法 1 所示.

算法 1. 代价最小化资源调度算法.

输入: 节点集合 D、日志集合 L、种群规模 P_o、迭代次数 I_0;

输出: 调度策略 $s_{n,m}$、带宽分配 $W_{n,m}$、统一代价 C_m^{tl}.

① 初始化

② A=*Encoding*(L_m), *SizeOf*(A)=P_o;

③ B=*Cross-Mutation*(A);

④ $C=A\cup B$;

⑤ for $i\in[1,I_0]$ do

⑥ $(F_1,F_2,\cdots)$=*Non-dominated Sorting*(C_i);

⑦ $S_i=\varnothing, p=1$;

⑧ do

⑨ $S_i=S_i\cup F_p$;

⑩ $p=p+1$;
⑪ while $|S_i| \geqslant P_o$
⑫ $F_l = F_p$;
⑬ if $S_i == P_o$ then
⑭ $A_{i+l} = S_i$;
⑮ break;
⑯ else
⑰ $A_{i+l} = \cup_{j=1}^{l-1} F_j$;
⑱ $K = P_o - |A_{i+l}|$;
⑲ $f'_m(c) = f_m(c) - z_m$;
⑳ $Association\ (A_i, z_n)$;
㉑ $p_j = \sum_{S \in A_i \setminus F_l} (\pi(s) == j ? 1 : 0)$;
㉒ 生成下一代种群A_{i+1};
㉓ end if
㉔ end while
㉕ end for
㉖ 返回$s_{n,m}$, $W_{n,m}$, C_m^{tl}.

3.2 一致性时延校验算法

数据库系统的数据一致性与RTO时延有密切的关系，本文将其建模为基于概率的时延约束. 为此，本节设计了一致性时延约束校验算法嵌入到CAIS中，实现可行解的筛选，其实现细节详见如下算法2所示.

算法 2. 一致性时延校验算法.

输入：第i代种群A_i，截止时间T_m^r，可容忍噪声υ;

输出：调度策略$s_{n,m}$，带宽分配$W_{n,m}$，统一代价C_m^{tl}.

① 初始化：$S = \varnothing, \eta = 0$;
② foreach 染色体$\in A_i$ do
③ $T_m = \sum_n t_{tra}(l_{n,m}) + t_{pro}(l_{n,m}) + t_{wai}(l_{n,m})$;
④ if $T_m > T_m^r$ then
⑤ $S=S+\{l_{n,m}\}$;
⑥ end if
⑦ end for
⑧ $\eta = 1 - SizeOf(S)/(N \times P_o)$;
⑨ if $\eta \geqslant \upsilon$ then
⑩ $A = A_i$;
⑪ 返回算法1(行③);
⑫ else
⑬ 返回$s_{n,m}$, $W_{n,m}$, C_m^{tl};
⑭ end if

算法2的主要作用是筛选算法1生成的第i代种群的可行解. 具体而言，算法2定义集合S储存那些日志处理和解析时间超过RTO时间限的个体，并计算这些个体占种群总规模的比例，记为η. 如果$\eta \geqslant \upsilon$，则需调整该种群多样性，返回算法1的行③；否则，该种群多样性被认为是满足需求的，继续返回调度策略、带宽分配和代价结果.

4 实验结果和性能分析

本节通过仿真实验的方式，详细分析本文所提网络架构和调度算法的性能优越性. 具体介绍如下.

4.1 初始设置

本文实验主要参数和默认取值详见表1.

Table 1 Description of Experiment Parameter Setting

表1 实验参数设置说明

系统参数	默认取值
数据库节点数量M	4
数据库节点算力f_m/MHz	2 000~5 000
节点日志数量N	10
节点信道带宽$W_{n,m}$/MHz	20~30
日志处理指令数$a_{n,m}$/kb	8~12
节点解析指令数$b_{n,m}$/kb	200~500
RTO截止时间T_m^r/s	10
可容忍噪声υ	2%
加性高斯白噪声功率σ^2/mW	3.98×10^{-12}
等待时延模型中的参数	$\beta = 30$, $p = 0.75$
计算资源基准代价γ_m	3×10^{-3}~7×10^{-3}
通信资源基准代价ι_m	3×10^{-3}~7×10^{-3}
代价转换系数	$\lambda = 100$, $\mu = 100$

为了更好地分析本文所提方法CAIS的性能优越性，将以下3种算法作为对比：

1）交替方向乘子法(alternating direction method of multipliers, ADMM). 通过分解协调过程交替优化多个子变量，将大的全局问题分解为多个较小、较容易求解的局部子问题.

2）Benchmark1. 此算法在满足基本算力和通信资源约束的前提下，将所有的优化变量均匀取值，即所有日志采用轮询的方式调度到节点、节点传输的所有日志平分信道带宽，作为对比的另一基准算法.

3）Benchmark2. 此算法在满足基本算力和通信资源约束的前提下，将所有的优化变量完全随机取值，作为对比的基准算法之一.

4.2 性能分析

针对4.1节的3种算法，本文采用Java语言在

Eclipse 工具上进行数值实验. 实验设备为具有 4 核 2 GHz 英特尔酷睿 i5 处理器、16 GB 3 733 MHz 内存的 MacBook. 所有实验在结果收敛的前提下运行超过 100 次, 取其均值作为结果, 其置信区间在 95% 以上.

首先, 本文分析了算网一体架构的驱动效果, 定量分析非业务算力的消耗, 实验结果见图 4. 为了评估数据库在运行时的非业务算力消耗, 本文采用 CAIS 作为基准算法, 将调度算法执行收敛前的迭代次数与算力消耗线性相关, 传输日志执行完成前会留下对应的冗余算力以在波动时实现数据一致性的保证.

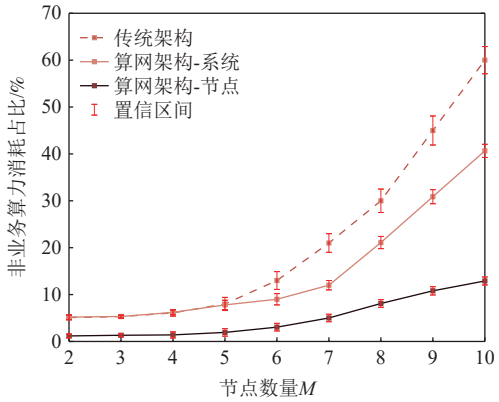


Fig. 4 Comparison of non-business computing comsumption

图 4 非业务算力消耗对比

随着节点数量的增加, 管控成本也在增加, 使得非业务算力的消耗占比提升. 其中, 传统架构中的占比近似指数型快速增加, 而本文算网架构中的节点消耗占比则近似线性缓慢增加. 这是因为在引入智能物联网技术后, 节点只需收集状态信息, 调度算法和冗余算力中的大部分在网络的管控层面消耗了. 同时, 算网架构中系统消耗的占比也小于传统架构. 这是因为新的架构通过在网调度的方式, 减少需要存储的数据副本, 降低了系统的非业务资源消耗. 数据库作为 AIoT 中的算力基础设施, 通常 1 个节点会存放多个物联网设备的数据信息[5,23]. 在占用相同非业务算力比例时, 本文的架构中 DDBS 节点数量拓展 1 倍以上, 增加了大规模部署的能力.

然后, 本文从节点规模和日志规模两方面详细评估各调度算法的成本优化性能, 其实验结果详见图 5 和图 6.

在评估节点规模时, 本文设定系统中需要通信调度的日志总数恒定, 即$|L|$=50, 每个节点中的日志均匀分布. 随着数据库节点数的不断增加, 系统的资源使用代价不断下降. 这是因为节点算力和网络信道资源逐渐丰富, 调度算法通过将日志调度到资源使用率更低的信道和节点上, 显著减少资源的使用总代价. 对比 4 个算法, CAIS 和 ADMM 的优化效果要显著优于 2 个基准算法, CAIS 的优化效果最好. 这是因为 CAIS 将算力和信道两者优化变量融合统一建模为了代价模型, 而 ADMM 是以分别优化 2 个优化变量的方式寻找最优解, 这一过程不可避免地导致优化效率下降. 因此, CAIS 算法避免了 2 个变量分别交替优化时的性能损失, 实现了更好的系统算网效能优化. 在 M=2 时, CAIS 的优化性能似乎弱于 ADMM. 这主要是因为 CAIS 中嵌入了 RTO 时延的约束校验算法. 该算法将日志调度到性能更好但代价可能更高的节点上更快速地执行.

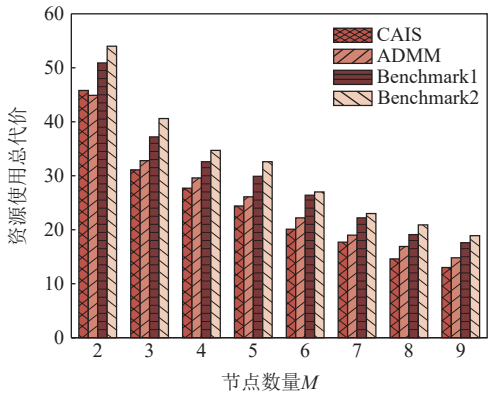


Fig. 5 Total cost varing with number of database nodes

图 5 总代价随数据库节点数量的变化

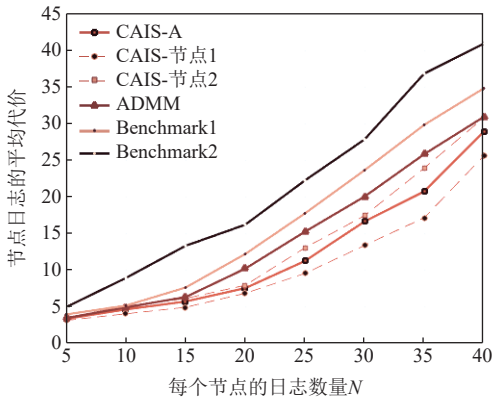


Fig. 6 Average cost varing with quantity of node logs

图 6 节点日志平均代价随节点日志数量的变化

在评估日志规模时, 本文设定系统中的数据库节点总数恒定, 通过调整每个节点的日志数量分析算法性能. 此外, 本文从每个节点的角度分析了算法的优化性能, 分别记作 CAIS-节点 1 和 CAIS-节点 2, 设置其日志参数为下限$a_{n,1}=8$、$b_{n,1}=200$和上限 $a_{n,2}=12$、$b_{n,2}=500$. 同时, 将所有节点的代价平均值记为 CAIS-A. 观察图 6 可知, 本文所提 CAIS 的优化性能

也显著优于其他算法. 特别是从单节点来看, CAIS 算法对需要解析和处理的日志权重有着良好的个体适应性, 展现了良好的局部优化性能.

接着, 本文分析嵌入的 RTO 时延约束校验算法的作用, 其实验结果详见图 7. 该实验设置数据库节点总数恒定, 以系统中的日志数量为自变量, 设计了 CAIS-w 和 CAIS-o 两个算法, 分别表示嵌入了和没有嵌入约束校验算法的情况. 同时, 引入 ADMM 算法作为对比. 实验结果显示, 在节点日志数量超过 25 个时, CAIS-w 和 CAIS-o 的性能差异开始显著. 这主要是因为面对T_m^r=10 s 的 RTO 时延约束时, 约束校验算法作用显现, 过滤掉时延较大的可行解, 最大限度地保证了系统一致性时延地约束. 同时适当松弛了该约束, 以满足流量极具变化等特殊情况下的系统响应.

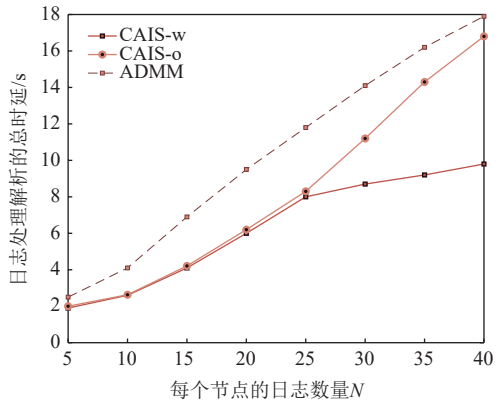


Fig. 7 Log procession resolution latency varing with number of node logs

图 7 日志处理解析时延随节点日志数量的变化

最后, 本文详细分析了节点算力和网络通信优化一体感知表征的意义, 其实验结果详见图 8. 本实验设置数据库节点总数恒定, 即 M=4, 采用 ADMM 作为对比. 实验设计了 2 个算法, 分别记作 CAIS-c 和 CAIS-d. 其中, CAIS-c 是本文采用的算法, 通过统一建模表征为代价的方式实现单目标的优化; CAIS-d 则将算力和信道资源双目标优化, 最后耦合为代价表示. 图 8 显示, CAIS-c 的优化性能显著好于其他算法, 这主要是因为多目标优化时增加的 1 个维度使得迭代过程中 2 个方向的优化解不断权衡, 共同实现多目标的最优化. 但这却不能等价于最优化资源使用的总成本, 造成了一定的性能损失.

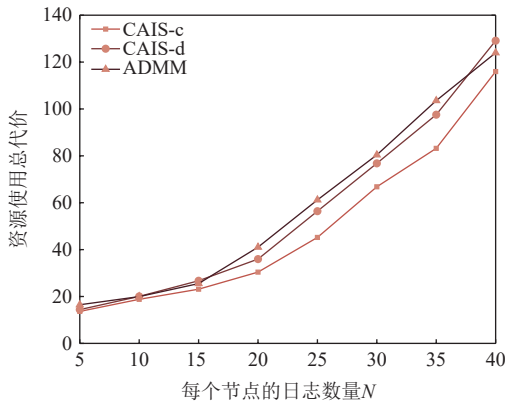


Fig. 8 Total cost of resource usage and number of node logs

图 8 资源使用总代价与节点日志数

从图 8 可以看出, CAIS-d 的性能较为接近 ADMM, 特别是在 N=40 时性能似乎更差了. 这主要是因为 CAIS-d 中的一致性时延校验算法对可行解的筛选所致, 它需要在资源紧张时将其日志调度到性能更高但成本可能不是最优的节点上解析和处理. 这再次验证了本文提出的统一建模能更好地降低系统集成成本, 提升资源效率.

同时, 本文也分析了统一代价建模表征与算法迭代次数的关系, 其实验结果详见表 2. 本实验设置数据库节点数 M=4, 节点日志数量 N=10, 以资源使用总代价为优化目标.

Table 2 Experimental Results under Different Iterations

表 2 不同迭代次数下的实验结果

迭代次数	CAIS-c 的总代价	CAIS-d 的总代价
100	50.2	61.1
200	40.3	53.2
300	30.8	40.1
400	21.2	32.3
500	19.1	26.4
600	18.8	25.3
700	18.8	22.4
800	18.8	20.3
900	18.8	20.1
1 000	18.8	20.1
1 100	18.8	20.1

表 2 显示, CAIS-c 大约在迭代到 600 次时就能收敛, 而 CAIS-d 大约到 900 次迭代才基本收敛, 揭示了本文的统一代价模型能显著地提升算法收敛性. 这是因为统一变量后的单目标优化过程减少了求解时的搜索维度, 大大提升了算法的收敛性.

5 总 结

本文针对物联网技术的广泛应用对高性能数据

库的迫切需要,将 AIoT 技术引入分布式数据库系统,构建算网一体数据库. 基于算力资源和网络的统一感知建模和智能联合调度,打破“算-网”隔离,实现了成本下降和效率提升. 具体而言,本文首先构建了 AIoT 技术赋能的分布式数据库网络架构,并分析其架构在降低集成成本和提升算网效能上的优势;接着,为了实现多变量的融合,构建统一代价感知模型实现计算和通信优化变量的统一表征;然后在此基础上提出智能化的一站式资源调度算法;最后,与其他架构和算法相比,实验结果验证了本文工作在网络性能、算法收敛、资源效率上的优越性.

未来的工作将探索现有算法和架构的实际部署性能,并开发出在线的方式实现数据库系统的实时分析与调度. 此外,数据库作为 AIoT 中的关键基础设施,未来也将联系具体的下游应用,定量分析其每秒查询速率和每秒处理任务数的效能优化效果.

作者贡献声明:朱茂盛完成实验并撰写论文;王宝晗提供论文思路;康曼聪提供仿真实验方案;于巍提供算法设计;杨利超提供指导意见并修改全文.

参 考 文 献

[1] Yin Huanpu, Yu Shuhui, Zhang Yingshuo, et al. Let IoT know you better: User identification and emotion recognition through millimeter-wave sensing[J]. IEEE Internet of Things Journal, 2022, 10(2): 1149−1161

[2] Xiao Han, Zhuang Yirong, Xu Changqiao, et al. Transcoding-enabled cloud-edge-terminal collaborative video caching in heterogeneous IoT networks: An online learning approach with time-varying information[J]. IEEE Internet of Things Journal, 2024, 11(1): 296−310

[3] Zhang Tao, Xu Changqiao, Lian Yibo, et al. When moving target defense meets attack prediction in digital twins: A convolutional and hierarchical reinforcement learning approach[J]. IEEE Journal on Selected Areas in Communications, 2023, 41(10): 3293−3305

[4] Xiao Han, Xu Changqiao, Ma Yunxiao, et al. Edge intelligence: A computational task offloading scheme for dependent IoT application[J]. IEEE Transactions on Wireless Communications, 2022, 21(9): 7222−7237

[5] Liu Shuai, Qiao Ying, Luo Xiongfei, et al. Key techniques of time series databases: A survey[J]. Journal of Computer Research and Development, 2024, 61(3): 614−638 (in Chinese)
(刘帅, 乔颖, 罗雄飞, 等. 时序数据库关键技术综述[J]. 计算机研究与发展, 2024, 61(3): 614−638)

[6] Rosenkrantz D J, Stearns R E, Lewis P M. System level concurrency control for distributed database systems[J]. ACM Transactions on Database System, 1978, 3((2):): 178−198

[7] Gardarin G, Goodman N, Lindsay B G, et al. Distributed database systems[C]//Proc of the 6th Int Conf on Very Large Data Bases (VLDB ’80). New York: ACM, 1980: 201–202

[8] Cai Hui, Qian Zhenghao, Jiang Jiang. Application of association rule algorithm in distributed new SQL database design[C]//Proc of the 2023 IEEE Int Conf on Integrated Circuits and Communication Systems (ICICACS). Piscataway, NJ: IEEE, 2023: 1−5

[9] Mansouri Y, Prokhorenko V, Ullah F, et al. Resource utilization of distributed databases in edge–cloud environment[J]. IEEE Internet of Things Journal, 2023, 10(11): 9423−9437

[10] Prasetio N, Ferdinand D, Wijaya B C, et al. Performance analysis of distributed database system in cloud computing environment[C]//Proc of 9th Int Conf on Computing, Engineering and Design (ICCED). Piscataway, NJ: IEEE, 2023: 1−6

[11] Su Yanzhi, Chen Guang, Jiang Yuewei. Overview of development of distributed database[J]. Digital Communication World, 2023(10): 172−174 (in Chinese)
(苏彦志, 陈广, 蒋越维. 分布式数据库发展综述[J]. 数字通信世界, 2023(10): 172−174)

[12] Peterson R J, Strickland J. P. Log write-ahead protocols and IMS/VS logging[C]//Proc of the 2nd ACM SIGACT-SIGMOD Symp on Principles of Database Systems(PODS ’83). New York: ACM, 1983: 216–2431

[13] Jhingran A, Khedkar P. Analysis of recovery in a database system using a write-ahead log protocol[C]//Proc of the 1992 ACM SIGMOD Int Conf on Management of Data (SIGMOD’92). New York: ACM, 1992: 175–184

[14] Sidell J. The Mariposa distributed database management system[C]//Proc of the 1996 ACM SIGMOD Int Conf on Management of Data (SIGMOD’96). New York: ACM, 1996: 76–80

[15] Wang Rusheng. Analysis of the application of distributed database in information management system[J]. Computer Knowledge and Technology, 2023, 19(35): 83−85 (in Chinese)
(王儒升. 分布式数据库在信息化管理系统中的应用探析[J]. 电脑知识与技术, 2023, 19(35): 83−85)

[16] Shen Hongxue, Guo Yanpu. New trends in distributed database research[J]. Software, 2021, 42(10): 31−34 (in Chinese)
(申红雪, 郭延哺. 分布式数据库研究新趋势[J]. 软件, 2021, 42(10): 31−34)

[17] Zguem S, Chen Qichen, Yeom H Y. NF-Log: Revisiting log writes in relational database for efficient persistent memory utilization[C]//Proc of the 38th ACM/SIGAPP Symp on Applied Computing (SAC’23). New York: ACM, 2023: 305–312

[18] Zheng Bolong, Gao Yongyong, Wan Jingyi, et al. DecLog: Decentralized logging in non-volatile memory for time series database systems[J]. Proceedings of the VLDB Endowment, 2024, 17(1): 1−14

[19] Wang Jinjin, Du Yizhou, Wang Xiangyu, et al. TEE-assisted time-scale database management system on IoT devices[C]//Proc of the 13th Int Conf on Internet of Things(IoT ’23). New York: ACM, 2023: 253–259

[20] Gong Xiaomin, Bai Chenchen, Ren Shuangyin, et al. A survey of compute first networking[C]//Proc of the 23rd Int Conf on Communication Technology (ICCT). Piscataway, NJ: IEEE, 2023: 688−695

[21] China Mobile Communications Group Co., Ltd.. Computing Power Network White Paper [R]. Beijing: China Mobile Communications Group Co., Ltd., 2021
(中国移动通信集团有限公司. 算力网络白皮书[R]. 北京: 中国移动通信集团有限公司, 2021)

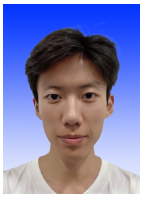
[22] Guo Fengxian, Sun Yaohua, Peng Mugen. Computing force networks in 6G: Architecture and key technologies[J]. Radio Communications Technology, 2023, 49(1): 21−30 (in Chinese)
(郭凤仙, 孙耀华, 彭木根. 6G 算力网络: 体系架构与关键技术[J]. 无线电通信技术, 2023, 49(1): 21−30)

[23] Yang Zhenkun, Yang Chuanhui, Han Fusheng, et al. Architecture and technology of OceanBase distributed relational database[J]. Journal of Computer Research and Development, 2024, 61(3): 540−554 (in Chinese)
(阳振坤, 杨传辉, 韩富晟, 等. OceanBase 分布式关系数据库架构与技术[J]. 计算机研究与发展, 2024, 61(3): 540−554)

[24] Vilaplana J, Solsona F, Teixido I, et al. A queuing theory model for cloud computing[J]. The Journal of Supercomputing, 2014, 69(1): 492−507

[25] Zhu Maosheng, Li Xi, Ji Hong, et al. Cost-aware resource allocation with probabilistic latency guarantee in B5G industrial private networks[C]//Proc of the 2022 IEEE Int Conf on Communications Workshops (ICC Workshops). Piscataway, NJ: IEEE, 2022: 532−537

[26] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(4): 577−601



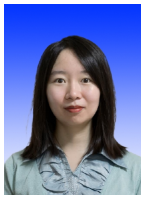
Zhu Maosheng, born in 1998. Master. His main research interests include computing force network, database technology, and edge computing.

朱茂盛, 1998 年生. 硕士. 主要研究方向为算力网络、数据库技术、边缘计算.



Wang Baohan, born in 1986. Senior system architecture expert, vice general manager of Ecloud's Database Product Department (hosted), expert of Artificial Intelligence Department of China Mobile Technology Association. His main research interests include computing force network, database theory, data mining, and intelligent data processing.

王宝晗, 1986 年生. 资深系统架构专家, 移动云数据库产品部副总经理(主持), 中移科协人工智能学部专家. 主要研究方向为算力网络、数据库理论、数据挖掘、智能数据处理.



Kang Mancong, born in 1995. PhD. Her main research interests include digital twin networks, machine learning, and haptic-visual communication.

康曼聪, 1995 年生. 博士. 主要研究方向为数字孪生网络、机器学习、多维感官通信.



Yu Wei, born in 1978. Advanced system architecture expert, director of Ecloud's Database Technology Committee. His main research interests include database kernel architecture, storage engine, and database security.

于　巍, 1978 年生. 高级系统架构专家, 移动云数据库技术委员会主任. 主要研究方向为数据库内核架构、存储引擎、数据库安全.



Yang Lichao, born in 1992. PhD, assistant professor. Her main research interests focus on machine learning, intelligent public health management, and medical databases technology.

杨利超, 1992 年生. 博士, 讲师. 主要研究方向为机器学习、智慧公共卫生管理、医学数据库技术.