

面向车联网低时延需求的分布式路径计算方案

李瑞彪¹, 任继军¹, 任智源²

(1. 西安邮电大学通信与信息工程学院, 710121, 西安;

2. 西安电子科技大学综合业务网理论及关键技术国家重点实验室, 710071, 西安)

摘要: 针对传统云服务应用于车联网大数据处理时存在通信时间消耗大、服务质量无法保证的问题,提出一种低时延的分布式路径计算方案。利用云雾网络架构将数据业务就近上传到雾网络设备以及云服务平台;为了让雾网络设备协同处理车载业务,采用一种基于该架构的任务映射策略(也称为业务卸载策略),将以有向无环图表示的车载业务流模型映射到以无向图表示的雾网络拓扑中;为了解决映射中存在的时延波动问题,采用一种改进的离散二值粒子群优化(NBPSO)算法进行优化;将该算法与负载均衡技术相结合得到最优映射方案。仿真结果表明:所提出的分布式路径协同计算方案可有效解决云计算模式下数据处理的高延时问题;当数据量为 12 Mb 时,采用所提方案的时延比采用云计算的时延降低约 2.95 s。

关键词: 车联网;云雾网络;路径计算;任务映射

中图分类号: TN925+.1 **文献标志码:** A

DOI: 10.7652/xjtuxb202210009 **文章编号:** 0253-987X(2022)10-0091-10

Distributed Path Computing Scheme for Low-Latency Demand of Internet of Vehicles

LI Ruibiao¹, REN Jijun¹, REN Zhiyuan²

(1. School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications,

Xi'an 710121, China; 2. State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China)

Abstract: In this paper, a low-latency distributed path computing scheme is proposed to solve the problems of long communication time and unstable services during big data processing for the internet of vehicles using traditional cloud service. First, a cloud-fog network architecture is adopted to upload data tasks nearby to the fog network devices and the cloud service platform; second, a task mapping strategy based on this architecture, also known as task offloading strategy, is employed to map flow models of vehicle tasks represented by directed acyclic graph (DAG) to the fog network topology represented by undirected graph (UG), which enables fog network devices to collaboratively process in-vehicle tasks; then, a modified discrete binary particle swarm optimization (NBPSO) algorithm is employed to optimize the mapping in order to tackle delay fluctuation in the mapping; finally, a load balancing technique is used in combination with this algorithm to obtain the optimal mapping scheme. According to the simulation results, the distributed collaborative computing scheme proposed in this paper can effectively tackle the delay of data processing in the cloud computing mode; the latency under this scheme is reduced by about 2.95 s

收稿日期: 2022-03-06。 作者简介: 李瑞彪(1997—),男,硕士生;任继军(通信作者),男,博士,高级工程师,硕士生导师。基金项目: 陕西省重点研发计划资助项目(2021GY-100)。

网络出版时间: 2022-07-04

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1069.T.20220701.1109.002.html>

compared with cloud computing when the data size is 12 Mb.

Keywords: internet of vehicles; cloud-fog network; path computing; task mapping

近几年,移动互联网、物联网和传感器技术迅速发展,车联网(internet of vehicles, IoV)技术成为推动未来智慧交通发展的关键驱动力之一。车联网是传统车载自组织网络(vehicular ad hoc network, VANET)的演变,实现了人、车、路、环境的智能交互^[1],其组织结构由采集层、传输层和服务层共3部分组成^[2]。采集层完成信息获取,传输层传递数据,服务层主要指车载设备在与网络服务设备进行信息交换的基础上,充分利用其计算资源为车辆终端提供不同的功能服务。车联网的业务根据应用场景划分为交通控制、信息服务以及驾驶安全保障共3大类型,且这3个类型在网络带宽、连接性能、时延以及网络部署方面具有不同要求^[3]。车联网的传输层支持车对车(V2V)、车对人(V2P)、车对路边单元(V2I)通信,对于小数据的V2V及V2I间传输安全类业务,传输时延问题并不突出。然而,对于时延要求高的动态地图、AR导航等信息类服务,在处理周围庞大的交通数据时相应会面临严峻的传输时延问题。而且,随着5G通信服务的不断扩大,车载网络的可扩展和复杂性决定了大数据业务低时延处理的难度,因此对于处理一系列承载海量数据的新兴车联网服务的需求尤为迫切^[4]。早期的车载应用以云计算和无线传感器网络(WSN)深度融合的车联网架构^[5]为支撑,凭借其丰富的计算、存储资源摆脱车载设备计算能力弱的困扰,但云计算的缺点也很明显,其远离车辆终端用户,无法解决大数据传输带来的高时延问题,导致车载用户体验较差。

为克服云计算模式的缺陷,许多学者展开了研究。Vemireddy等提出了将车辆当作雾节点的车载雾计算卸载机制,利用移动的空闲计算资源迅速地协作完成应用服务^[6];刘可欣等提出了一种“车-边-云”协同网络架构,并基于改进的烟花算法完成最佳的任务调度,实现时延性能的最优化^[7];Vankadara等提出了一种移动云环境中动态任务卸载方案,专注于降低任务在本地和远程计算的能源消耗和处理时延^[8];Younis等提出了一个权衡任务分配的多边缘节点任务卸载方案,并基于线性编程方法对网络时延实现大幅度的性能提升^[9];Hou等提出了雾计算协助无人机群的理论架构,弥补了云计算在处理延迟敏感服务方面的不足^[10];Hussain等针对车联网中海量数据低时延通信问题,提出一种动态Q学

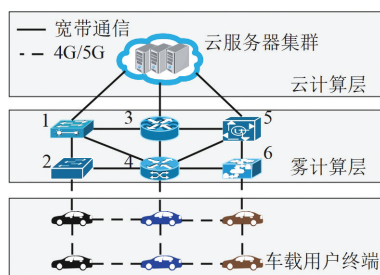
习算法来进行合理的网络选择,并通过对信道容量、车辆的性能考虑来选择最优的传输路径^[11]。但是,上述研究采用的任务结构和类型较为单一,且没有实现数据在传输过程中完成任务计算。

为此,本文引入一种基于云雾网络架构的分布式路径节点计算技术^[12-16],它与网络功能虚拟化(network functions virtualization, NFV)技术^[17]相结合,将不同结构的应用服务灵活地嵌入到附近广泛分布的边缘雾网络设备中,在保障大数据业务的传输性能上有独特的优势。设备之间构建的网络链接支持任务在传输过程中完成计算的能力,但路径计算技术对传输时延问题不突出的安全类业务的提升空间不大。在当前负载均衡技术深入分布式计算的应用当中,考虑到单个网络执行节点的数据负载过高或过低都会造成不必要的时延消耗,因此节点设备间要合理分配任务资源^[18]。又考虑到每个雾节点设备的运算能力有限,无法独立完成复杂的计算业务,所以一个完整业务可以分割成若干个独立的子业务。本文采用一种任务映射策略,将分割的子业务映射到附近固定的雾计算单元完成路径计算,在保障通信质量良好的同时有效降低了时延^[19]。由于映射的不唯一,必然存在不合理的任务映射方案,因此基于改进的离散二值粒子群优化(NBPSO)算法得到最优的任务映射。本文还研究了不同类型业务、不同结构业务以及不同雾设备负载在本文方案下的性能。需要说明的是,在追求时延最小化时会导致网络设备的能耗增加,本文为简化研究的复杂性并未加入能耗约束^[20]。

1 云雾网络架构

为了将车载服务处理周期变短,保障用户服务质量(quality of service, QoS),本节选取一个从上到下依次为云计算层、雾计算层以及车载用户终端构成的云雾网络架构,如图1所示。

云计算层由高性能服务器集群以分布式方式构建,具备计算速度快、规模大、可用性高等优点。雾计算层由部署在不同地点的交换机、路由器、网关等固定的边缘设备组成,实现应用服务的全网覆盖。这些设备具有一定的计算、存储和转发能力,便于在设备集群内进行数据传输。车载用户终端指车载智能终端,包括车载电脑、智能导航系统等智能设备。



1,2—交换机;3,4—路由器;5,6—网关。

图 1 云雾网络架构

Fig. 1 Cloud and fog network architecture

3 层网络架构之间用无线通信技术进行信息交换。其中,车载用户终端既是服务请求的发起者,也是数据的最终接收者,可以与附近或远端的网络边缘设备以及云服务器进行无线通信连接。一般而言,云计算层中服务器的计算能力远远超过单个雾节点的计算能力,但又因其部署远离终端应用,使得数据传输时延在整个业务周期中的比重较大。因此,在处理简单的车载业务时,将车载用户终端的服务请求上传到附近的网络雾节点,利用多个雾节点的计算资源协同处理数据。但是,如果当前的车载业务过于复杂,将任务交给云端可以更好地保证服务质量。此外,云计算层负责雾设备的实时监控和管理,同时根据雾节点的算力状态、转发能力以及业务流模型等相关信息,提前拟定好车辆计算业务卸载到网络边缘设备的业务卸载策略,并将业务之间明确的调度关系从云端缓存到指定雾网络中的执行节点,利用泛在网络节点的算力资源实现在网协同计算,降低了业务消费的时间成本。

2 基于 NBPSO 算法的业务卸载策略

2.1 有向无环图的业务流模型

有向无环图(DAG)描述的业务流定义如下: DAG 表征为无环的有向图,有向无环图是对具有优先关系或依赖关系的事物进行抽象描述。将有向无环图应用于车联网业务场景中,一个复杂的车载业务往往可以由若干个相对简单且独立的车载子业务组成,它们在执行时间上存在同步或异步的依赖约束。有向无环图的各节点作为子业务节点,有向边代表子业务之间的依赖关系,用边上权值表示前后子业务的数据传输时延。本文将同一业务的 DAG 业务流模型简单分为串行业务流、并行业务流以及串并业务流共 3 种结构。

在业务调度过程中,从业务起点计算到达业务

终点的有向路径的数量是由 DAG 业务流模型的分支条数所决定的,且每条路径上的边权值总和存在差异,故导致不同分支路径耗费总时间可能不同,只有各分支路径中的全部子业务均处理完毕,整个业务才能结束。因此,完整业务的结束时间为所有任意分支路径中起点到汇点的最大累积时延。

为了实现车联网应用服务从中心转移到网络边缘,从集中处理数据转变为分散处理数据,本节将有向无环图表示的业务流模型映射到无向图(UG)表示的雾网络拓扑结构中,映射示例如图 2 所示。

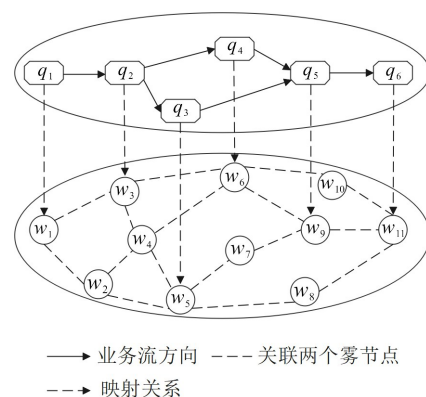
 $q_1 \sim q_6$ —子业务节点; $w_1 \sim w_{11}$ —雾网络节点。

图 2 有向无环图至无向图的映射示例

Fig. 2 Example of a mapping from a directed acyclic graph to an undirected graph

2.2 DAG 至 UG 映射方案

本文将车载应用服务分散化,用 $M=(Q,K)$ 表示 DAG 形式的车载业务流模型,然后定义 $Q=\{q_1, q_2, \dots, q_s, q_{s+1}, \dots, q_{n-1}, q_n | s \geq 1, n > s+1\}$ 为模型 M 的车载子业务集合。 $q_1, q_2, \dots, q_s$ 为 s 个车载子业务起点, $q_{s+1}, \dots, q_{n-1}$ 为中间车载业务集合, q_n 为车载业务终点。 K 为 M 的有向边的集合,表示车载子业务间的依赖性。子业务 q_j 的前向子业务集合用 $r(q_j)=\{q_j | (q_i, q_j)\}$ 定义。

定义 $U=(W,E)$ 无向连通图表征静态雾网络节点拓扑图,图 U 的雾网络节点集合 $W=\{w_1, w_2, \dots, w_s, w_{s+1}, \dots, w_{l-1}, w_l | s \geq 1, l > s+1\}$ 。 $w_1, w_2, \dots, w_s$ 为数据的发起节点, w_l 为目标节点, $w_{s+1}, \dots, w_{l-1}$ 为中间节点,发起节点和目标节点都是车辆终端设备的直连节点。 E 为图 U 中雾节点通过无线连通的边集合,相邻节点间的边支持双向传输,关联两个雾计算节点。

定义 1 子业务集合卸载规则。定义卸载节点集合 ξ 为 $Q \rightarrow W$, 满足

$$\xi(q_i) = \begin{cases} w_i, i \in \{1, 2, \dots, s\} \\ w_i, i = n \\ w_j, w_j \in W / \{w_1, w_2, \dots, w_s, w_l\} \end{cases} \quad (1)$$

由式(1)可知,车载子业务集合 Q 中的车载业务起点 $q_1, q_2, \dots, q_s$ 、车载中继业务 $q_{s+1}, \dots, q_{n-1}$ 、车载终点业务 q_n 分别依次被卸载到雾计算节点集合 W 的数据发起节点 $w_1, w_2, \dots, w_s$ 、中间节点 $w_{s+1}, \dots, w_{l-1}$ 以及目标节点 w_l 。卸载关系由 ξ 进行统一描述,且源末节点业务卸载唯一,中间节点业务卸载具有随机性,故同一车辆卸载业务卸载至固定的雾网络拓扑时存在多种卸载方案。

定义 2 有向边集合映射规则。定义映射 ψ 为 $K \rightarrow P$, 其中, $P = \{P_{\text{ath}, w_i, w_j} \mid w_i, w_j \in W\}$, P_{ath, w_i, w_j} 为节点 w_i 到 w_j 的数据传输最短路径, P 为所有 P_{ath, w_i, w_j} 的集合, d_{w_i, w_j} 为在此路径下两节点的单位数据量传输最短时延。对于雾网络中所有直接或间接关联的两个雾节点,参照网络边权重和节点连通状态,利用最短路径(Dijkstra)算法得到连通节点间的最短时延 d_{w_i, w_j} 和最短路径 P_{ath, w_i, w_j} 。

将 DAG 业务流模型的有向边映射为节点 $\xi(q_i)$ 至节点 $\xi(q_j)$ 的数据传输最短路径 P_{ath, w_i, w_j} , 并用 ψ 表示映射的结果集,即

$$\psi(q_i, q_j) = P_{\text{ath}, \xi(q_i), \xi(q_j)} \quad (2)$$

2.3 DAG 至 UG 的映射时延优化

同一车辆卸载业务映射至固定静态雾网络拓扑时,存在多种映射方案,不同的路径方案产生的时延不同。因此,合理分配和有效利用雾节点的运算资源可以改善业务处理的时延性能。为降低不同雾节点之间因数据多跳带来的传输时延开销,本文构建了一种时延性能优化方法。

在某次业务映射关系中,子业务 q_i 完成所耗费的时延 $T(q_i)$ 为子业务 q_i 的累积时延 $T_a(q_i)$ 和此业务下的计算时延 $T_c(q_i)$ 之和,即

$$T(q_i) = T_a(q_i) + T_c(q_i) \quad (3)$$

式中 q_i 的累积时延 $T_a(q_i)$ 是 q_i 前向子业务的时延之和,包括计算时延和传输时延,计算式为

$$T_a(q_i) = \max_{q_j \in r(q_i)} [T(q_j) + d_{\xi(q_j), \xi(q_i)} D_{q_j q_i}] \quad (4)$$

其中, $D_{q_j q_i}$ 表示节点 $\xi(q_j)$ 到达节点 $\xi(q_i)$ 最短通路上处理的数据量。 q_i 的计算时延 $T_c(q_i)$ 与其映射节点的数据量 D_{q_i} 、计算复杂度系数 η 以及雾节点计算能力 $p_{\xi(q_i)}$ 的变化有关,即

$$T_c(q_i) = \frac{D_{q_i} \eta}{p_{\xi(q_i)}} \quad (5)$$

由上述得出,业务终点 q_n 的处理时延与模型 M

的业务处理时延等价,即

$$T(M) = T(q_n) \quad (6)$$

以 DAG 业务流形式的车辆卸载应用映射至指定雾网络节点时存在多维性,因此业务处理的平均延迟不同。业务与雾节点的多维映射关系都可以由关系矩阵 $\mathbf{X}$ 来表征。关系矩阵 $\mathbf{X}$ 为

$$\mathbf{X} = \begin{bmatrix} x_{q_1 w_1} & \cdots & x_{q_1 w_s} & \cdots & x_{q_1 w_l} \\ x_{q_2 w_1} & \cdots & x_{q_2 w_s} & \cdots & x_{q_2 w_l} \\ \vdots & & \vdots & & \vdots \\ x_{q_n w_1} & \cdots & x_{q_n w_s} & \cdots & x_{q_n w_l} \end{bmatrix} \quad (7)$$

式中 $x_{q_p w_q} \in \mathbf{X}$ 表征指定车载子业务到指定雾节点的映射情况。若 $x_{q_p w_q} = 0$, 则表示子业务 q_p 到雾节点 w_q 无映射关系;若 $x_{q_p w_q} = 1$, 则表示子业务 q_p 到雾节点 w_q 有唯一映射关系。映射关系也可以表示为

$$x_{q_p w_q} \in \{0, 1\}, \forall q_p \in Q, \forall w_q \in W \quad (8)$$

根据式(8)的映射关系,式(3)更新为

$$T(q_i) = \max_{q_j \in r(q_i)} \{T(q_j) + \sum_{w_p, w_m \in W} (d_{w_p w_m} D_{q_j q_i} \cdot x_{q_j w_m} x_{q_i w_p})\} + \sum_{w_p \in W} \frac{D_{q_i} \eta}{p_{w_p}} x_{q_i w_p} \quad (9)$$

根据以上介绍的节点映射规则,车载业务 $T(M)$ 的处理时延可表示为 $\mathbf{X}$ 的函数,即

$$T(M) = F(\mathbf{X}) \quad (10)$$

将上述时延优化问题转化为二值优化问题,模型 M 至图 U 的最佳映射方案建模如下

$$\begin{cases} \mathbf{X} = \arg \min(F(\mathbf{X})) \\ \left\{ \begin{array}{l} x_{q_p w_q} \in \{0, 1\}, \forall q_p \in Q, \forall w_q \in W \\ \sum_{w_q \in W} x_{q_p w_q} = 1, \forall q_p \in Q \\ x_{q_p w_q} = 1, p = q, \forall p \in [1, s] \\ x_{q_p w_q} = 1, q_p = q_n, w_q = w_l \\ x_{q_p w_q} = 0, p \in [1, s], \forall q \in [s+1, l-1] \\ x_{q_p w_q} = 0, w_q = w_l, \forall p \in [s+1, n-1] \\ T(q_p) = 0, \forall p \in [1, s] \end{array} \right. \end{cases} \quad \text{s. t} \quad (11)$$

2.4 改进的离散二值粒子群优化算法

改进的离散二值粒子群优化(NBPSO)算法是一个全局随机寻优算法,主要用来优化离散空间的约束问题,是粒子群算法(BPSO)的一种改进算法^[21-22]。该算法不仅促进了粒子收敛于群体最优,而且改善了后期的个体寻优能力,提高了BPSO算法的收敛性能,方便求式(11)的时延优化问题。

假设粒子群由 M 个粒子构成,寻优空间为 m

维,最大迭代次数为 $N_{\max}$,则第 i 个粒子在有限次迭代过程中的位置 $\mathbf{X}_i$ 和速度 $\mathbf{V}_i$ 可表示为

$$\begin{cases} \mathbf{X}_i = (x_{i1}, x_{i2}, \dots, x_{io}) \\ \mathbf{V}_i = (v_{i1}, v_{i2}, \dots, v_{io}) \end{cases} \quad 1 \leq i \leq m \quad (12)$$

式中 o 表示寻优空间总维数,即子业务与每个雾节点可能存在映射的所有情况。

NBPSO 算法的每个粒子是由二进制串组成,其运动的局部轨迹均存在一个最优值点,且众多粒子中必然存在一个粒子的局部最优位置达到群体最优,定义为 P_{best} 。第 k 次迭代,粒子的局部最优加速度 a_{io}^k 为

$$a_{io}^k = c_i \zeta_i^k (p_{io}^k - x_{io}^k) \quad (13)$$

式中: p_{io}^k 是第 i 个粒子在局部范围内的最优取值; x_{io}^k 为第 i 个粒子在 o 维的位置信息; c_i 是加速因子(也称为学习因子); ζ_i^k 是一个从区间 $[0, 1]$ 统一分布中产生的随机值。

粒子的动态速度值与迭代次数呈正相关,定义粒子的自适应速度 e_{io} 为

$$e_{io} = c_i \zeta_i^k I_{io}^k (\bar{x}_{io}^k - x_{io}^k) (1 - S(A_i)) \quad (14)$$

$$S(A_i) = \frac{1}{1 + \exp(-A_i)} \quad (15)$$

式中: $\bar{x}_{io}^k$ 是 o 维定义下的 ζ_i^k ; $\bar{x}_{io}^k$ 是 x_{io}^k 的二进制取反; I_{io}^k 指第 i 粒子的未更新迭代次数; A_i 为第 i 粒子的自适应值; $S(A_i)$ 为第 i 粒子速度值转化为二进制位取 1 的概率。

第 i 个粒子经过 k 次迭代后的速度变化公式为

$$v_{io}^{k+1} = \omega v_{io}^k + \sum_{i=1}^m a_{io}^k + e_{io} \quad (16)$$

式中 ω 为惯性权重。第 i 个粒子经过 k 次迭代后的位置变化公式为

$$x_{io}^{k+1} = x_{io}^k + v_{io}^{k+1} \quad (17)$$

定义自适应度函数为

$$f(\mathbf{X}) = F(M) = F(\mathbf{X}) \quad (18)$$

此外,用 $f_{\min}(p_{io}^k)$ 表示从局部最优值中筛选出的全局最优值。

NBPSO 算法求解时延优化模型的具体步骤如下。

- 1 参数的设置: $M, N_{\max}, c_i, \omega$
- 2 for $i \leq M$ do
- 3 随机初始化粒子的位置和速度
- 4 由式(18)自适应度函数 $f(\mathbf{X})$ 评估所有粒子
- 5 初始化粒子的个体最优 p_{io}^k
- 6 end for
- 7 for 1 to $N_{\max}$ do
- 8 for $i \leq M$ do
- 9 根据式(13)~(16)计算粒子迭代的速度 $\mathbf{V}_i$

- 10 根据式(17)计算粒子迭代后的位置 $\mathbf{X}_i$
- 11 根据式(18)重新评估粒子的自适应值
- 12 if $f(x_{io}^{k+1}) < f(p_{io}^k)$ then
- 13 $p_{io}^k = x_{io}^{k+1}$
- 14 end if
- 15 end for
- 16 end for
- 17 $p_{\text{best}} = f_{\min}(p_{io}^k)$
- 18 return p_{best}

3 性能仿真和评估

为了测试分布式路径计算方案应用于车联网场景中高通信效率的优点,本文首先对比中心云服务计算与分布式路径计算的时延,然后分别从车联网的业务类型、DAG 业务流模型、雾节点负载以及不同算法等多个维度去分析该计算方案下的时延性能。同时,考虑到仿真数据的真实性,本次仿真得到的最终数据是通过 30 次实验的平均值。

假设将一个完整的车载业务划分为 6 个相对独立的子业务,本文针对同一业务仿真采取了 3 种简单结构的 DAG 业务流模型,依次为串行业务流、并行业务流、串并业务流,如图 3 所示。

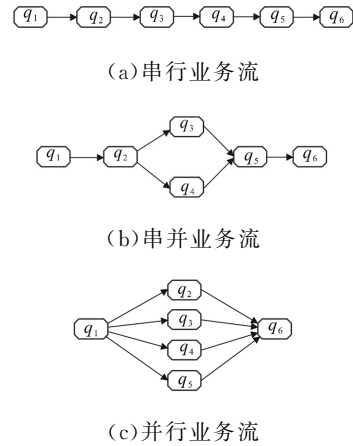


图 3 DAG 形式的业务流模型

Fig. 3 Business flow model in the form of DAG

图 3 中,3 种结构图的首尾分别表示业务始点和业务终点,其余均为中间业务,根据业务卸载策略,将 6 个子业务卸载到下文采用的雾网络拓扑图中。

为不失一般性,本文利用 Matlab 仿真平台随机产生由 12 个雾节点组成的雾网络拓扑图,并设置边缘权重表征连通节点间的数据传输速率,如图 4 所示。其中, w_1 为业务处理起始节点, w_{12} 为业务处理终止节点, $w_2 \sim w_{11}$ 为业务处理中间节点。

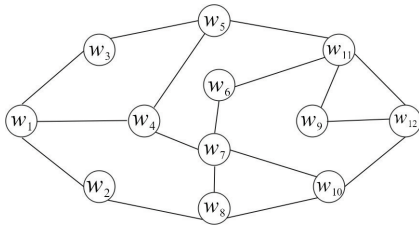


图4 雾网络拓扑

Fig. 4 Fog network topology

3.1 仿真参数设置

本文采用的仿真软件为 Matlab, 实验涉及的参数借鉴文献[23-25]的数据, 具体设置见表1。其中, 平均业务计算复杂度 η 表示平均分配到单个雾映射节点的业务计算复杂度。

表1 云雾网络和 NBPSO 算法参数

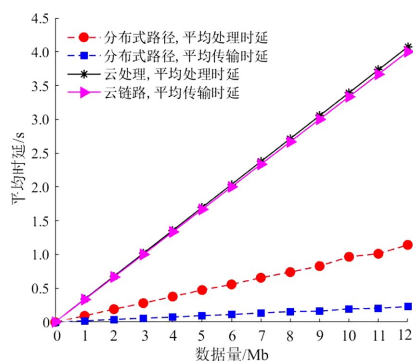
Table 1 Parameters of cloud-fog network and NBPSO algorithm

参数	数值
中心云计算能力 f_c/GHz	20
云雾链路通信传输速 $v_c/(\text{Mb} \cdot \text{s}^{-1})$	4
单个雾节点的计算能力 $f_{\text{fog}}/\text{GHz}$	[1.5, 4.5]
雾链路通信传输速率 $v_{\text{fog}}/(\text{Mb} \cdot \text{s}^{-1})$	[25, 45]
平均业务计算复杂度 η	1 900/8
种群数量 M	100
最大迭代次数 N_{max}	500
加速因子 c_i	1
惯性权重 ω	1.5

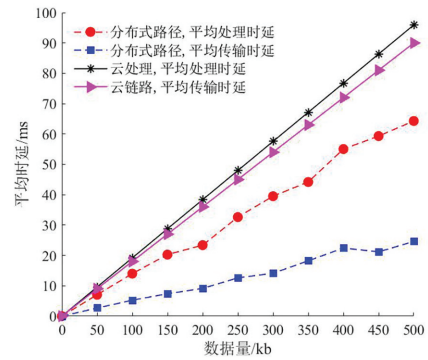
3.2 仿真结果分析

3.2.1 云计算和分布式路径计算平均时延对比

为验证分布式路径计算的优越性, 本小节在雾节点计算能力相同的前提下, 从数据量、业务计算复杂度两个角度来比较云计算和分布式路径计算的时延性能。每个雾映射节点处理业务的数据量和复杂度相同, DAG 业务流模型仅采用图3中的并行结构。云计算与路径计算基于不同数据量的时延对比如图5和图6所示。



(a) 大数据量下云计算和路径计算的时延对比



(b) 小数据量下云计算和路径计算的时延对比

图5 云计算与路径计算基于不同数据量的时延对比

Fig. 5 Latency comparison between cloud computing and path computing based on different data volumes

从图5可知, 随着数据量的不断增加, 分布式路径计算和中心云计算的平均时延(包括平均传输时延和平均处理时延)均呈现上升趋势。此外, 不难看出, 在0~500 kb和0~12 Mb下, 中心云计算的平均时延均要高于分布式路径计算的平均时延, 这是由于中心云远距离接收和转发业务数据导致的结果。当业务数据量为300 kb和12 Mb时, 本文所提方案的时延分别为39.51 ms和1.12 s, 较云计算的时延分别降低了18.73 ms和2.95 s。显然, 该方案对于传输安全类业务和用户数据类业务都具有改善意义。特别地, 对于大数据用户业务的时延改善程度更为显著。这是因为当处理小数据安全类业务时, 云强大的计算能力在一定程度上可以缓解小数据传输过程中的延时问题。但是, 当处理大数据业务时, 云端的计算能力发挥的效果有所下降, 其传输上的消耗变大。相反, 与云中心的超负荷通信相比, 泛在连接的网络边缘设备靠近终端用户, 服务响应及时。基于云雾网络架构实现的分布式路径协同计算方案可以显著降低大数据业务请求时延。

图6比较了分布式路径协作计算和云计算在不同业务计算复杂度下的时延变化趋势, 此时仿真的业务总数据量固定为12 Mb, 平均业务计算复杂度根据表1设置为 $0.6\eta \sim 1.6\eta$ 。可以看出, 当业务复杂度由 η 逐渐增加至 1.6η 时, 路径协作计算的平均处理时延与云处理平均时延也相应增长, 但云计算增长迟缓。例如, 复杂度由 η 增加至 1.6η 时, 时延仅从3.21 s增加到3.43 s。这是因为数据量一定时, 云计算模式的传输时延不会随业务复杂度的变化而改变, 路径协作计算的传输时延也变化微小, 所以业务的平均等待时延仅存在计算上的时延消耗。考虑

到网络边缘单个汇聚节点的计算能力有限,所以路径式协作计算处理复杂业务的平均时延上涨的幅度较大。相反,中心云的计算能力较强,处理复杂业务时的时延性能优于路径协作计算。由此可推出,当业务复杂度不断增加甚至超越某阈值后,路径式协作计算的处理时延将会超过云计算的处理时延。因此,复杂度较低时,分布式路径协作计算的时效性能好,复杂度较高时,中心云处理的时效性能好。

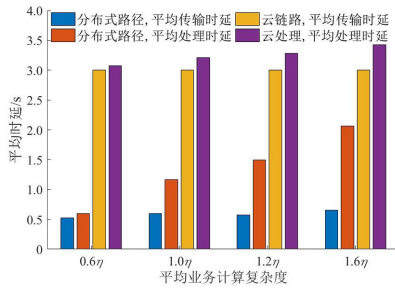


图 6 云计算和路径计算基于计算复杂度的时延对比

Fig. 6 Latency comparison between cloud computing and path computing based on computational complexities

综上所述可知,与云计算远距离服务相比,分步骤处理业务的路径协作计算方案极大地提升了业务请求的响应速度,更易体现低时延的特性。

3.2.2 不同业务类型的时延性能分析

车联网不同业务场景对于时延的容忍度不同,本小节旨在研究不同类型业务在分布式路径计算方案下的时延性能。根据业务承载数据量和复杂度的不同,本文设计了驾驶安全类(A1~A3)、交通控制类(B1~B3)、信息服务类(C1~C3)共 3 种业务。其中,1、2、3 分代表任务的复杂度,1 代表复杂度最小,3 代表复杂度最大。不同业务类型的云计算和路径计算的时延比较如图 7。

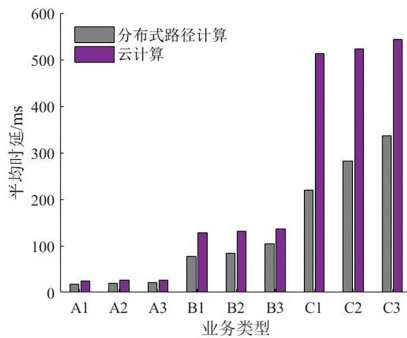


图 7 不同业务类型的云计算和路径计算的时延比较

Fig. 7 Latency comparison between cloud and path computing for different business types

从图 7 可以看出,分布式路径计算处理不同类型业务的平均时延均低于云计算的。其中,驾驶安全类、交通控制类、信息服务类业务的时延降低范围分别为 19.4%~29.3%、24.2%~40.4% 以及 38.5%~57.0%。信息服务类业务在路径计算方案下的改善效果最好。这是因为承载小数据的驾驶安全业务虽然业务的计算复杂度要低很多,但是在传输性能上的改善程度远不如承载大数据的信息服务类业务明显。因此,分布式路径计算在降低大数据和低复杂度类型任务的时延效果最好。

3.2.3 异构 DAG 业务流模型的时延性能分析

针对同一业务存在多种 DAG 业务流形式的内在结构,不同结构对分布式路径计算方案的时延性能会产生不同的影响,本小节旨在分析图 3 中所提 3 种异构的业务流模型映射到雾网络拓扑中的时延性能差别。设置各雾映射节点的业务计算复杂度一致,不同数据量下 3 种 DAG 结构的时延比例如图 8 所示。

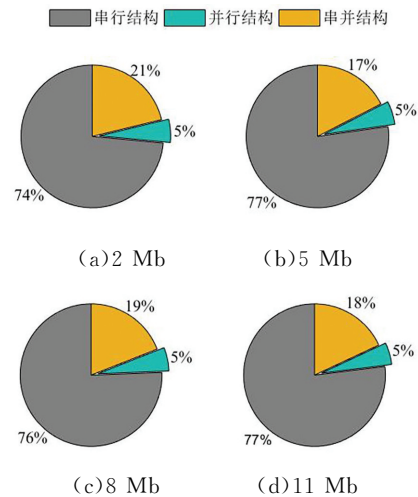


图 8 不同数据量下 3 种 DAG 结构的时延比例

Fig. 8 Latency proportion of three DAG structures with different data volumes

从图 8 可以看出:不同数据量下的端到端时延情况基本一致;3 种结构中并行结构的时延占比最小,约为 5.0%;串行结构的时延占比最大,约为 76.0%;串并结构的时延占比居中,约为 18.7%。这是因为不同结构的 DAG 业务流模型卸载到固定的网络边缘节点上分步骤处理业务数据,业务完成所花费的时间是雾映射起始节点到雾映射终止节点的最大累积时延。从图 3 可知,本文采用的串行业务流、并行业务流以及串并业务流的最大累积时延由任意分支路径下的最大业务数目确定,分别为 6、5、3。由于同一时间段处理的业务数目越多,时延累

积权重越大,故对于同一业务的 3 种异构 DAG 模型,并行结构下用户体验的最好,串行结构欠佳。

3.2.4 业务处理时延与雾节点负载方案的相关性

为避免出现部分雾映射节点因过载故障或过低负载导致的资源浪费问题,本小节研究数据量和计算复杂度负载分配关系对业务处理总时延的影响,DAG 业务流模型同样仅采用图 3 的并行结构,将此 DAG 模型中的子业务集合划分为前端业务、中端业务以及后端业务。前端业务有 q_1, q_2, q_3 ,中端业务有 q_4, q_5 ,后端业务有 q_6, q_7, q_8 。4 种前中后端业务处理数据量和复杂度的负载分配关系为 1:2:3,2:2:2,3:2:1,1.5:3:1.5,分别用 A、B、C、D 表示,并设置总业务处理数据量为 15 Mb,总计算复杂度为 1 900。

数据量和计算复杂度分配比例对总时延的影响如图 9 所示。可以看出:当节点数据量负载分别固定为 A、B、D 方案时,复杂度分配中 A、B、C、D 方案的时延依次增大;当节点数据量的负载分配比例为 C 时,复杂度分配方案中 A 的时延性能优于其他分配比例,C 的时延性能弱于其他比例关系,且 A 至 C 的时延增长为 2.06 s。此外,节点复杂度分配方案固定时,不同数据量分配下 A 方案时延最小。综合考虑可知,数据量和计算复杂度的分配关系都为 A 时,时延性能达到最优,约 1.44 s,两者都为 C 时的时延性能最差,约 3.78 s,所以基于 4 种比例关系中最大时延差距约 2.34 s。依此结果推断,前端业务节点分配的数据量负载少或复杂度较低时,业务处理时延有明显改善。因此,本文任务映射策略在业务前简单、后复杂和数据量前缩后放方案下有好的负载均衡性能。

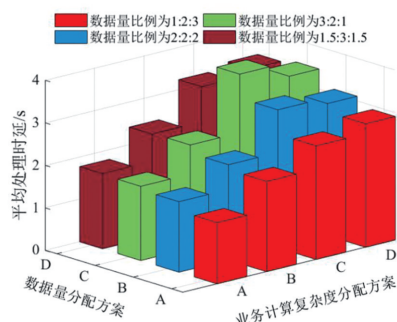


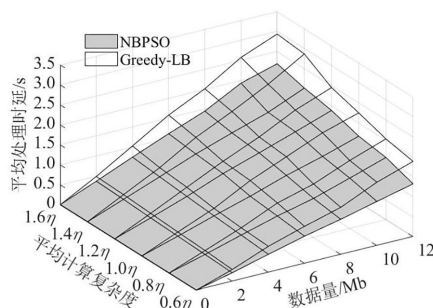
图 9 数据量和计算复杂度分配比例对总时延的影响

Fig. 9 Impact of data volume and computational complexity allocation ratio on total latency

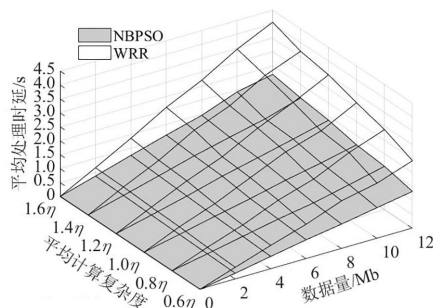
3.2.5 NBPSO 算法与其他算法的时延性能对比

本小节将 NBPSO 算法同随机动态算法(Pick-KX)、加权轮转法(WRR)以及贪婪算法(Greedy-

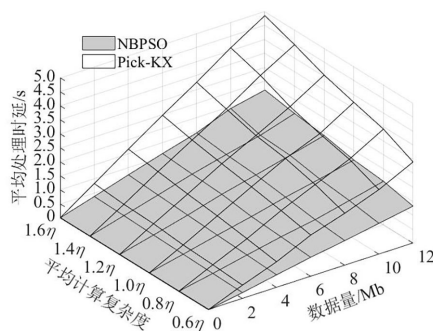
LB)的时延容忍度比较,从而验证该算法在分布式路径计算方案中良好的时效性。采用并行的 DAG 业务流模型,NBPSO 算法与其他算法的时延性能对比如图 10 所示。



(a) NBPSO 与 Greedy-LB 的时延比较



(b) NBPSO 与 WRR 的时延比较



(c) NBPSO 与 Pick-KX 的时延比较

图 10 NBPSO 算法与其他算法的时延性能对比

Fig. 10 Latency performance of NBPSO algorithm compared with other algorithms

从图 10 可以看出,NBPSO 算法的平均时延容忍度均低于其他 3 种算法的。显然,这 4 种算法在数据量和计算复杂度影响下的整体变化趋势一致,在任意平均计算复杂度下,NBPSO 算法与其他 3 种算法的时延差距随着数据量的增长而增大。同样地,不同数据量下,当平均计算复杂度从 1.6η 逐渐减少至 0.6η 时,两者之间的时延宽度也相应变窄,

由此表明 NBPSO 算法的时延性能优于其他 3 种算法的。分析图 10 可知,在业务量为 12 Mb、复杂度为 0.67 时,NBPSO 算法的时延为 1.34 s,较 Greedy-LB、WRR、Pick-KX 算法的时延分别降低了 25.2%、46.3%、53.4%。

研究分析可知,Pick-KX 算法随机选择若干个边缘网络节点进行业务映射,并没有将节点的计算能力考虑在内;WRR 算法和 Greedy-LB 算法虽然都参照了节点的计算能力,且 Greedy-LB 算法优先选择当前局部负载最小的节点,但两者均未考虑网络边数据传输速率低带来的高通信消耗问题,依然达不到时效性能最好。本文采取的 NBPSO 算法全面地研究了节点的计算能力和网络边的数据传输速率。

4 结束语

本文提出的基于云雾网络架构的分布式任务映射策略具备优越的传输性能,有效改善了车联网新兴大数据任务的低时延问题。但是,本文提出的计算方案只能在静态网络边缘设备中应用,没有切实考虑映射节点故障的服务迁移问题。而且,车辆应用服务的多样性决定了 DAG 业务流模型并不固定,若不同服务同时映射到相同雾网络拓扑处理时,则需考虑多业务在相同或不同时间下的执行优先级问题。因此,未来的研究方向应该从静态节点计算转移到移动车辆协同计算、故障转移提高可靠性能、多业务并发或按时间顺序卸载等方面。

参考文献:

- [1] PAN Ying. Research on the development of internet of vehicles technology [J]. Internet of Things and Cloud Computing, 2019, 7(1): 12-18.
- [2] 岳胜. 车联网技术研究 [J]. 数字通信世界, 2020 (11): 121-122.
YUE Sheng. Research on internet of vehicles technology [J]. Digital Communication World, 2020 (11): 121-122.
- [3] 章文卿. 例析 5G 车联网业务的类型及应用 [J]. 信息通信, 2019(7): 235-236.
ZHANG Wenqing. Analysis of the types and application of 5G internet of vehicles services [J]. Information & Communications, 2019(7): 235-236.
- [4] ZHOU Haibo, XU Wenchao, CHEN Jiacheng, et al. Evolutionary V2X technologies toward the internet of vehicles: challenges and opportunities [J]. Proceedings of the IEEE, 2020, 108(2): 308-323.
- [5] 唐小淋, 林培群, 徐建闽. 基于云计算和 WSN 的车联网体系架构及关键技术研究 [J]. 交通信息与安全, 2011, 29(5): 106-111.
TANG Xiaolin, LIN Peiqun, XU Jianmin. Framework and the key technologies for car networking system based on cloud computing and WSN [J]. Journal of Transport Information and Safety, 2011, 29(5): 106-111.
- [6] VEMIREDDY S, ROUT R R. Fuzzy reinforcement learning for energy efficient task offloading in vehicular fog computing [J]. Computer Networks, 2021, 199: 108463.
- [7] 刘可欣, 陈桂芬, 张文萍. 基于移动边缘计算的车联网卸载策略研究 [J]. 长春理工大学学报(自然科学版), 2021, 44(2): 67-73.
LIU Kexin, CHEN Guifen, ZHANG Wenping. Research on unloading strategy of internet of vehicles based on mobile edge computing [J]. Journal of Changchun University of Science and Technology (Natural Science Edition), 2021, 44(2): 67-73.
- [8] VANKADARA S, DASARI N. Energy-aware dynamic task offloading and collective task execution in mobile cloud computing [J]. International Journal of Communication Systems, 2020, 33(13): e3914.
- [9] YOUNIS A, QIU B, POMPILI D. Latency and quality-aware task offloading in multi-node next generation RANs [J]. Computer Communications, 2022, 184: 107-117.
- [10] HOU Xiangwang, REN Zhiyuan, WANG Jingjing, et al. Distributed fog computing for latency and reliability guaranteed swarm of drones [J]. IEEE Access, 2020, 8: 7117-7130.
- [11] HUSSAIN S M, YUSOF K M, HUSSAIN S A. Artificial intelligence-based network selection and optimized routing in internet of vehicles [J]. Transport and Telecommunication Journal, 2021, 22(4): 392-406.
- [12] AHMED M, MUMTAZ R, ZAIDI S M H, et al. Distributed fog computing for internet of things (IoT) based ambient data processing and analysis [J]. Electronics, 2020, 9(11): 1756.
- [13] SHEN Jing, LI Yongjie, ZHANG Yong, et al. A survey on task offloading in edge computing for smart grid [C]//Proceedings of the 11th International Conference on Computer Engineering and Networks. Singapore: Springer Singapore, 2022: 13-20.
- [14] ALAM I, KUMAR S. Functionality, privacy, security and rewarding based on fog assisted cloud compu-

- ting techniques in internet of vehicles [J]. Journal of Discrete Mathematical Sciences and Cryptography, 2021, 24(3): 763-775.
- [15] CARVALHO G, CABRAL B, PEREIRA V, et al. Edge computing: current trends, research challenges and future directions [J]. Computing, 2021, 103(5): 993-1023.
- [16] 丛犁, 李锐, 王华, 等. 面向工业物联网的分布式数据存储分配方案 [J]. 电力信息与通信技术, 2020, 18(7): 52-57.
CONG Li, LI Rui, WANG Hua, et al. Distributed data storage allocation scheme for the industrial internet of things [J]. Electric Power Information and Communication Technology, 2020, 18(7): 52-57.
- [17] COTRONEO D, DE SIMONE L, IANNILLO A K, et al. Network function virtualization: challenges and directions for reliability assurance [C]//2014 IEEE International Symposium on Software Reliability Engineering Workshops. Piscataway, NJ, USA: IEEE, 2014: 37-42.
- [18] XU Xiaolong, GU Renhao, DAI Fei, et al. Multi-objective computation offloading for Internet of Vehicles in cloud-edge computing [J]. Wireless Networks, 2020, 26(3): 1611-1629.
- [19] 惠宁, 伍杰, 周一青, 等. 未来车辆雾计算网络 [J]. 电信科学, 2020, 36(6): 14-27.
HUI Ning, WU Jie, ZHOU Yiqing, et al. Future vehicular fog computing networks [J]. Telecommunications Science, 2020, 36(6): 14-27.
- [20] 任智源, 侯向往, 郭凯, 等. 分布式卫星云雾网络及时延与能耗策略 [J]. 浙江大学学报(工学版), 2018, 52(8): 1474-1481.
REN Zhiyuan, HOU Xiangwang, GUO Kai, et al. Distributed satellite cloud-fog network and strategy of latency and power consumption [J]. Journal of Zhejiang University (Engineering Science), 2018, 52(8): 1474-1481.
- [21] 刘建华, 杨荣华, 孙水华. 离散二进制粒子群算法分析 [J]. 南京大学学报(自然科学版), 2011, 47(5): 504-514.
LIU Jianhua, YANG Ronghua, SUN Shuihua. The analysis of binary particle swarm optimization [J]. Journal of Nanjing University (Natural Sciences), 2011, 47(5): 504-514.
- [22] 沈佳杰, 江红, 王肃. 基于多点速度向量和自适应速度值的离散二进制粒子群算法改进 [J]. 计算机科学, 2013, 40(S2): 125-130.
SHEN Jiajie, JIANG Hong, WANG Su. Improved binary particle swarm optimization algorithm based on multi velocity vector and adaptive speed value [J]. Computer Science, 2013, 40(S2): 125-130.
- [23] HASSAN M A, XIAO Mengbai, WEI Qi, et al. Help your mobile applications with fog computing [C]//2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking-Workshops (SECON Workshops). Piscataway, NJ, USA: IEEE, 2015: 1-6.
- [24] 何秀丽, 任智源, 史晨华, 等. 面向医疗大数据的云雾网络及其分布式计算方案 [J]. 西安交通大学学报, 2016, 50(10): 71-77.
HE Xiuli, REN Zhiyuan, SHI Chenhua, et al. A cloud and fog network architecture for medical big data and its distributed computing scheme [J]. Journal of Xi'an Jiaotong University, 2016, 50(10): 71-77.
- [25] 张夏童, 任智源, 胡锦涛, 等. 面向医疗大数据任务低时延需求的路径计算方案 [J]. 西安交通大学学报, 2020, 54(2): 119-126.
ZHANG Xiatong, REN Zhiyuan, HU Jintao, et al. A path computing scheme for low-latency requirement of medical big data task [J]. Journal of Xi'an Jiaotong University, 2020, 54(2): 119-126.

(编辑 陶晴 刘杨)