

基于算力路由的空间信息网络低时延在轨协同计算策略

马步云¹, 任智源¹, 郭 凯², 程文驰¹

(1 西安电子科技大学通信工程学院 西安 710071;

2 北京遥测技术研究所 北京 100076)

摘要: 针对当前空间信息网络(Spatial Information Network, SIN)地面数据处理模式中存在的传输时延问题, 提出了基于算力路由的低时延在轨协同计算策略。考虑到卫星网络的动态变化及卫星计算资源的异构性, 提出了要素时空扩展图模型, 以实现在屏蔽SIN动态性的同时对星上资源进行精确表征。在此基础上, 构建基于算力路由的多星在轨协同计算数学模型, 提出时延优化问题, 并利用改进的异构最早完成时间(Heterogeneous Earliest Finish Time, HEFT)算法进行求解。仿真结果表明: 卫星在轨协同计算可有效降低卫星数据的任务处理时延; 同时, 本文所提的改进HEFT算法以较小且可接受的收敛性能为代价, 换取了业务处理时延的大幅度降低。

关键词: 空间信息网络; 算力路由; 要素时空扩展图

中图分类号: TN927+.2; V474.2+1 文献标志码: A 文章编号: 2095-1000(2023)05-0008-08

DOI: 10.12347/j.ycyk.20230411001

引用格式: 马步云, 任智源, 郭凯, 等. 基于算力路由的空间信息网络低时延在轨协同计算策略[J]. 遥测遥控, 2023, 44(5): 8-15.

Computation dependent routing based on-orbit collaborative computing strategy for satellite information networks

MA Buyun¹, REN Zhiyuan¹, GUO Kai², CHENG Wenchi¹

(1. School of Telecommunications Engineering, Xidian University, Xi'an 710071, China;

2. Beijing Research Institute of Telemetry, Beijing 100076, China)

Abstract: In order to solve the problem of high transmission latency in the current data processing mode of Spatial Information Network (SIN), a low latency on-orbit collaborative computing strategy based on computation dependent routing is proposed. Considering the dynamic change of satellite network and the heterogeneity of satellite computing resources, an element time expanded graph model is proposed to realize the computing resources of satellite and shield the SIN dynamic characteristic. On this basis, a mathematical model of multi-satellite on-orbit collaborative computing based on computation dependent routing is constructed, and the latency optimization problem is proposed. To solve the above problem, an improved Heterogeneous Earliest Finish Time (HEFT) algorithm is designed. The simulation results show that the on-orbit collaborative computation can effectively reduce the task processing latency of satellite data, and the proposed improved HEFT algorithm reduce the task processing latency substantially with a little and acceptable convergence performance.

Keywords: Spatial information network; Computation dependent routing; Element time expanded graph

Citation: MA Buyun, REN Zhiyuan, GUO Kai, et al. Computation dependent routing based on-orbit collaborative computing strategy for satellite information networks[J]. Journal of Telemetry, Tracking and Command, 2023, 44(5): 8-15.

引 言

近年来, 空间信息网络发展迅速, 其以空间

平台为载体, 实时采集和传输数据, 可实现卫星遥感、卫星导航和卫星通信的一体化集成应用服务。同时, 卫星数据具备成像区域范围大, 场景

基金项目: 2021年航天科技集团青年拔尖人才支持工程

通信作者: 任智源 (zyren@xidian.edu.cn)

收稿日期: 2023-04-11 修回日期: 2023-05-19

中目标类型多等特点,因此,其通常需要被进一步处理才能直接应用。

传统的卫星数据处理模式为地面处理模式,即数据转发至地面中心进行集中式处理^[1],但这将导致传输时延过高,卫星数据得不到及时处理,对于应急救援等紧急任务响应不及时。为解决此问题,卫星在轨计算得到广泛关注。美国的EO-1(Earth Observation One Mission)卫星能够在轨完成高光谱图像区域变化检测、无效数据剔除。国内清华大学的“航天清华一号”微小卫星实现了在轨云检测,滤除云覆盖区域图像^[2]。虽然部分卫星具备一定的在轨数据处理能力,但由于卫星载荷受限,星上高速实时处理能力还需进一步提升,且不同的卫星通常具备不同的处理功能,单星无法独立完成所有数据的处理^[3]。因此,亟须研究空间动态组网协同计算技术,通过联合多颗卫星的异构资源,以实现海量空间数据的在轨观测与处理。

由于空间数据一方面需要端到端传输至用户,另一方面需要被进一步处理,因此,空间动态组网协同计算主要面临的挑战是路由和计算卸载联合优化。针对上述挑战,学者们提出了算力路由技术作为算力网络中的关键技术,可将网络中的通信资源与算力资源结合起来^[4],满足业务的传输和处理需求。但是,算力路由的相关研究大多数为地面静态场景中的性能优化^[5-8],面向卫星动态网络的研究很少。本课题组针对卫星动态网络中的协同计算问题展开初步研究^[9,10]。文献[9]提出了一种业务图驱动的星间协作计算方案,将协同计算问题转化为业务图到网络图的图映射问题,并采用离散二值粒子群算法(Discrete Binary Particle Swarm Optimization Algorithm, BPSO)求解该问题。文献[10]提出了一种基于多维梯度的分布式协同计算策略,通过综合考虑卫星的多维信息,选取参与任务处理的路径节点,逐条完成业务的计算与传输。然而,以上工作均假设卫星具备处理所有业务的能力,而在实际SIN中,星上计算资源具有异类的特点,如硬件包括FPGA、CPU、GPU等,软件操作系统有VxWorks、Integrity等,异构的星上资源导致卫星承载的业务类型不同。因此,现有的卫星在轨协同计算策略无法应用于实际的SIN中。

针对以上问题,本文提出一种基于算力路由

的空间信息网络在轨协同计算策略。具体包括:构建SIN网络模型,通过将卫星具有的服务功能抽象为元素,构建卫星要素时空扩展图(Element Time-Expanded Graph, ETEG),以实现在屏蔽SIN动态性的同时对星上资源进行表征。在此基础上,构建基于算力路由的多星在轨协同计算模型,将任务表征为有向无环图(Directed Acyclic Graph, DAG),并进一步将在轨协同计算问题转化为业务图至网络图的图映射问题,并通过引入跨时隙机制及节点功能约束,对传统的DAG调度算法-异构最早完成时间算法进行改进,用于求解上述问题。

1 系统模型

SIN系统架构如图1所示,主要包括探测卫星、低轨卫星等。在此架构中,探测卫星作为数据源卫星,不仅负责探测数据采集,同时通过部署改进HEFT算法,制订多星协同策略,所需的全局拓扑信息可通过运行OSPF(Open Shortest Path First, 开放式最短路径优先)获得;其他卫星根据协同策略参与任务计算。

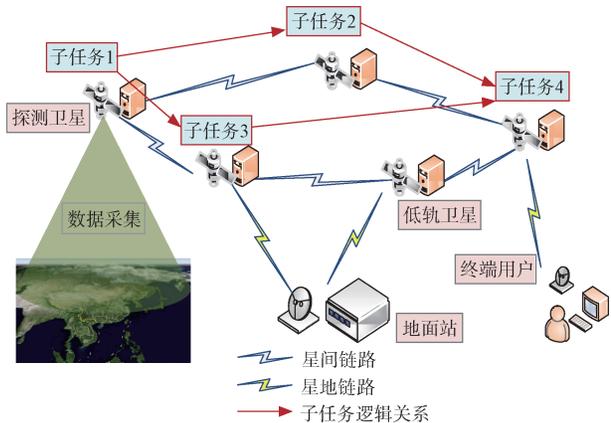


图1 SIN系统架构图

Fig. 1 The SIN system architecture

1.1 SIN网络模型

为了屏蔽SIN的动态性,同时有效管理卫星的资源,本节构建SIN网络模型,为后续任务映射提供基础。

首先,以地心为原点构建三维坐标系,卫星 $V = \{v_1, v_2, \dots, v_m, \dots, v_M\}$ 的坐标 $(x_{v_i}, y_{v_i}, z_{v_i})$ 可根据开普勒六大参数 $P_i = \{a_i, e_i, \delta_i, \Omega_i, \omega_i, f_i(t)\}$ 来确定,其中 a_i 为半长轴, e_i 为偏心率, δ_i 为倾角, Ω_i 为升交点赤经, ω_i 为近地点幅角, $f_i(t)$ 为真近地点角。

根据自由空间损耗以及香农公式, (v_i, v_j) 之间的星间链路容量可表示为

$$C_{ij}(t) = B \log_2(1 + en(t)), \quad (1)$$

其中, B 为星间链路带宽, $en(t)$ 为动态信噪比。设卫星间通信的最低容量为 C_0 , 当 $C_{ij} \geq C_0$ 时, (v_i, v_j) 之间的星间链路连通, 且链路容量为 C_0 , 否则链路断开, 其中 C_0 为星间链路连通的容量门限。

其次, 依托链路模型, 构建 ETEG, 以实现在卫星网络稳态化的同时对异构星上算力资源的管理。

为了解决 SIN 拓扑频繁变化引起的网络不稳定问题, 本文采用时空扩展图(Time-Expanded Graph, TEG)模型^[11,12]对 SIN 的动态拓扑进行稳态化处理。

定义 SIN 拓扑变化周期为 T , 将周期 T 划分为 k 个时隙, 每个时隙的长度为 $\Delta t = T/k$ 。假设每个时隙内卫星的拓扑固定不变。假设每个时隙内卫星的拓扑固定不变, 并引入时隙间的虚拟链路, 以表示数据在相同节点的缓存过程, 从而形成时空扩展图。

具体来讲, 在第 k 个时隙中, 卫星网络的加权邻接矩阵可表示为式(2), 其中, 单位数据量的传输时延可由 $\pi_{ij}^k = 1/C_{ij}$ 计算, 若 $\pi_{ij}^k = \infty$, 则代表卫星 $v_i, v_j (i \neq j)$ 在第 k 个时隙不连通。

$$\mathbf{G}_k = \begin{bmatrix} \infty & \pi_{12}^k & \cdots & \pi_{1(M-1)}^k & \pi_{1M}^k \\ \pi_{21}^k & \infty & \cdots & \pi_{2(M-1)}^k & \pi_{2M}^k \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \pi_{(M-1)1}^k & \pi_{(M-1)2}^k & \cdots & \infty & \pi_{(M-1)M}^k \\ \pi_{M1}^k & \pi_{M2}^k & \cdots & \pi_{M(M-1)}^k & \infty \end{bmatrix}. \quad (2)$$

此外, 引入虚拟链路 $\pi_i^{k(k+1)} = \Delta t - t_i^k$ 表示相邻时隙间同一节点中的数据缓存延迟, 其中 t_i^k 是第 k 个时隙中第 i 个节点的消耗时间。时隙间的加权矩阵可表示为

$$\mathbf{G}_{k,k+1} = \begin{bmatrix} \pi_1^{k(k+1)} & \cdots & \infty & \cdots & \infty \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \infty & \cdots & \pi_j^{k(k+1)} & \cdots & \infty \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \infty & \cdots & \infty & \cdots & \pi_M^{k(k+1)} \end{bmatrix} \quad (3)$$

根据当前时刻拓扑与未来时隙的预测拓扑, 卫星网络的 TEG 可表示为

$$\mathbf{G}_{\text{Graph}} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_{1,2} & \infty & \cdots & \infty & \infty \\ \infty & \mathbf{G}_2 & \mathbf{G}_{2,3} & \cdots & \infty & \infty \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \infty & \infty & \infty & \cdots & \mathbf{G}_{N-1} & \mathbf{G}_{N-1,N} \\ \mathbf{G}_{N,1} & \infty & \infty & \cdots & \infty & \mathbf{G}_N \end{bmatrix} \quad (4)$$

依托 TEG, 可实现卫星网络的稳态化。然而, 在实际 SIN 系统中, 不同的卫星上配备了不同的功能模块, 在制定多星协同策略时, 必须充分考虑卫星功能异构这一特性。因此, 本文在 TEG 的基础上, 对卫星上配备的功能模块用要素进行表征, 将 TEG 进一步升级为 ETEG, 在稳态化的基础上实现对星上异构算力资源的管理, 为后续多星在轨协同计算提供基础。

ETEG 中包含了各种卫星资源, 例如星间连通关系、星上的功能模块等, 如图 2 所示。

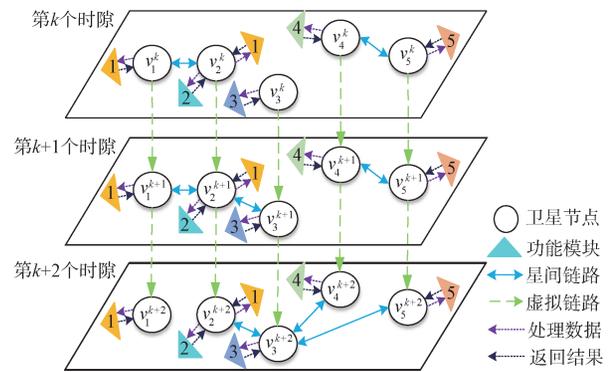


图 2 多时隙的卫星要素图示意图

Fig. 2 The multi-slot element time expanded graph

为了直观起见, 功能模块被抽象为直接连接到卫星的节点^[13]。设卫星网络可提供 N 种功能模块, 所有第 $1 \leq x \leq N$ 种的功能模块类型可以表示成数组 $f^x = \{f_1^x, f_2^x, \dots, f_M^x\}$, 其中 $f_j^x (1 \leq j \leq M)$ 代表卫星 v_j 装备的第 x 类型的功能模块。则卫星网络中的所有功能模块节点均可用 $F = \{f^1, f^2, \dots, f^N\}$ 表示, 则第 k 个时隙内的要素卫星网络邻接矩阵可表示为

$$\mathbf{GM}_k = \begin{bmatrix} \mathbf{G}_k & \mathbf{VF}^1 & \mathbf{VF}^2 & \cdots & \mathbf{VF}^N \\ \mathbf{FV}^1 & \infty & \infty & \cdots & \infty \\ \mathbf{FV}^2 & \infty & \infty & \cdots & \infty \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{FV}^N & \infty & \infty & \cdots & \infty \end{bmatrix} \quad (5)$$

其中, \mathbf{VF}^x 表示卫星执行功能 f^x 的时延, 其可进一步表示为式(6), 矩阵中的第 i 行代表节点 V_i , 第 j 列代表功能 f_j^x , 位于对角线的元素值表示为 ct_{ix} , 其余元素均为无穷大; \mathbf{FV}^x 表示卫星调用功能 f^x 处理结果的时延, 其可进一步表示为式(7), 由于要素直接部署在卫星上, 所以该时延为 0。

$$VF^x = \begin{bmatrix} ct_1 & \infty & \cdots & \infty \\ \infty & ct_2 & \cdots & \infty \\ \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \cdots & ct_M \end{bmatrix} \quad (6)$$

$$FV^x = \begin{bmatrix} 0 & \infty & \cdots & \infty \\ \infty & 0 & \cdots & \infty \\ \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \cdots & 0 \end{bmatrix} \quad (7)$$

相邻时隙 $\{k, k+1\}$ 增加要素的卫星网络拓扑结构的带权邻接矩阵可表示为

$$GM_{k,k+1} = \begin{bmatrix} G_{k,k+1} & \infty & \infty & \cdots & \infty \\ \infty & 0 & \vdots & \ddots & \vdots \\ \infty & \cdots & 0 & \cdots & \infty \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \infty & \cdots & \infty & \cdots & 0 \end{bmatrix} \quad (8)$$

则 ETEG 的加权邻接矩阵可表示为

$$G_{\text{Graph}} M = \begin{bmatrix} GM_1 & GM_{1,2} & \infty & \cdots & \infty & \infty \\ \infty & GM_2 & GM_{2,3} & \cdots & \infty & \infty \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \infty & \infty & \infty & \cdots & GM_{N-1} & GM_{N-1,N} \\ GM_{N,1} & \infty & \infty & \cdots & \infty & GM_N \end{bmatrix} \quad (9)$$

1.2 基于算力路由的多星在轨协同计算

基于 ETEG, 本小节构建多星在轨协同计算策略。为不失一般性, 将业务表示为 DAG 模型^[14-16], 则多星在轨协同计算问题可被转化为 DAG 至 ETEG 的图映射问题。

定义 $\Phi = (\Psi, Y)$ 为业务的 DAG 模型, 其中 $\Psi = \{\phi_1, \dots, \phi_j\}$, 表示 DAG 中的子任务节点, Y 表示子任务间的逻辑关系。定义 $\Theta_{\uparrow}(\phi_j) = \{\phi_i | (\phi_i, \phi_j) \in Y\}$, 为 ϕ_j 的前向节点, 则 ϕ_j 的任务量为 D_j 可表示为

$$D_j = \sum_{\phi_i \in \Theta_{\uparrow}(\phi_j)} D_i \eta_i, \quad (10)$$

其中, η_i 是数据缩放因子, 表示进一步处理后数据量的变化情况。

定义 $B: \Psi \rightarrow F$, 表示 Ψ 到 $\{V, G_{\text{Graph}} M\}$ 的映射, 如式(11)所示, ϕ_i 映射到任务发起卫星, ϕ_i 映射到结果接收卫星, 其他任务映射到传输路径的任意节点。此外, 考虑到传输过程中链路可能断开, 此时数据需在卫星上缓存, 经过虚拟链路至下一时隙, ρ_i 为跨时隙数目。

$$B(\phi_i) = \begin{cases} \{v_1^1, \dots, v_1^{1+\rho_i}\}, \phi_i = \phi_1; \\ \{v_d^q, \dots, v_d^{q+\rho_i}\}, \phi_i = \phi_i; \\ \{v_p^q, \dots, v_p^{q+\rho_i}\}, \text{otherwise.} \end{cases} \quad (11)$$

定义 $Z: Y \rightarrow E$, 表示 Y 至 E 的映射, 以反映子业务间的依赖关系, 如式(12)所示, 将 DAG 的有向边 $\forall (\phi_i, \phi_j) \in Y$ 映射为图中 $B(\phi_i)$ 至 $B(\phi_j)$ 之间的最短路由 $p_{B(\phi_i)B(\phi_j)}^{\text{path}}$

$$Z((\phi_i, \phi_j)) = p_{B(\phi_i)B(\phi_j)}^{\text{path}} \quad (12)$$

为了最小化任务处理时延, 基于上述映射规则, 任务 Φ 的执行时延如式(13)。其中, $f_{B(\omega_i)}$ 为节点 $B(\omega_i)$ 的计算能力, $d_{B(\omega_i)B(\omega_i)}$ 为沿最短路由 $p_{B(\omega_i)B(\omega_i)}^{\text{path}}$, 由 $\forall \omega_j \in \Phi_{\uparrow}(\omega_i)$ 传输单位比特至 ω_i 的时延。 α_i 为子任务的计算复杂度, 单位为 cycles/bit, 表示处理单位比特所需的 CPU 周期数。

$$\begin{aligned} T(\Phi) &= T(\omega_i) = T_{\text{comp}}(\omega_i) + T_{\text{accu}}(\omega_i) \\ &= \max_{\omega_i \in \Phi, (\omega_i)} [T(\omega_i) + d_{\text{end}(B(\omega_i))\text{start}(B(\omega_i))} D_i \eta_i]^+ \\ &\quad \frac{D_i \eta_i \alpha_i}{f_{B(\omega_i)}}. \end{aligned} \quad (13)$$

对于任务 Φ 来说, 存在多个映射结果同时满足映射规则, 进而导致不同的任务处理时延。因此, 为了方便表征任务的映射结果, 进而依托映射结果寻找时延最低的任务调度策略, 本文进一步定义了决策矩阵 X

$$X = \begin{bmatrix} x(\omega_1, v_1^1) & \cdots & x(\omega_1, v_M^1) & \cdots & x(\omega_1, v_M^k) \\ x(\omega_2, v_1^1) & \cdots & x(\omega_2, v_M^1) & \cdots & x(\omega_2, v_M^k) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x(\omega_l, v_1^1) & \cdots & x(\omega_l, v_M^1) & \cdots & x(\omega_l, v_M^k) \end{bmatrix} \quad (14)$$

其中, 元素 $x(\omega_i, v_p^q) \in X$ 表示节点 ω_i 是否被映射到节点 v_p^q , 当 $x(\omega_i, v_p^q) = 1$ 时, 表示子任务 ω_i 被映射至节点 v_p^q ; 反之, 子任务 ω_i 不被映射至 v_p^q 。

依托式(13)~(14), 某个映射结果下的任务处理时延可表示为

$$\begin{aligned} T(X) &= \max_{\omega_i \in \Phi, (\omega_i)} [T(\omega_i) + \\ &\quad \sum_{v_p^{q+\rho_i}, v_d^q \in V^T} d_{v_p^{q+\rho_i}, v_d^q} D_i \eta_i x(\omega_i, v_p^{q+\rho_i}) x(\omega_i, v_d^q)]^+ \\ &\quad \sum_{v_d^q \in V^T} \frac{D_i \eta_i \alpha_i}{f_{v_d^q}} x(\omega_i, v_d^q) \end{aligned} \quad (15)$$

其中, $x(\omega_i, v_d^q)$ 表示 ω_i 映射的最后一个节点。

基于式(15), 可计算任意映射方案 X 对应的任务处理时延, 不同的 X 导致了不同的任务处理时延。则寻找最优多星在轨协同计算方案问题可被抽象为寻找任务处理时延最低的 X , 这大大简化了问题的求解。本小节构建了以 X 为自变量的时延优

化问题, 如式 (16) 所示

$$\begin{aligned} \min_{\mathbf{X}} T(\mathbf{X}) \\ \text{s.t. C1: } x(\omega_i, v_p^q) \in \{0, 1\} \quad \forall \omega_i \in \Omega, \forall v_p^q \in V^T \\ \text{C2: } \sum_{v_p^q \in V^T} x(\omega_i, v_p^q) = \rho_i + 1 \quad \forall \omega_i \in \Omega, \end{aligned} \quad (16)$$

其中, 约束 C1 表示, 对于 $x(\omega_i, v_p^q) \in \mathbf{X}$, 值只能为 0 或 1, 当 $x(\omega_i, v_p^q) = 1$ 时, 表示子任务 ω_i 被映射至节点 v_p^q ; 反之, 子任务 ω_i 会被映射至除节点 v_p^q 外的其他单个节点; 约束 C2 表示, 对于子任务 ω_i , 其计算过程可能经过 ρ_i 个时隙, 则其被映射到 ETEG 中的 $\rho_i + 1$ 个不同时隙的相同节点上。

1.3 改进 HEFT 算法

上节构建的二值优化问题本质上是求解 DAG 调度方案。由于复杂度低、性能好等优势, HEFT 算法^[7]被广泛应用于求解此问题。然而, 由于传统 HEFT 算法并未考虑网络拓扑的动态变化, 且依赖于任意网络节点均包含所有服务功能的假设, 因此, 无法应用于具有动态异构特性的实际 SIN 网络中。为此, 本小节对 HEFT 算法做出改进: ① 增加了跨时隙机制, 以配合 ETEG 克服网络的动态性; ② 引入节点功能约束, 以适用于节点功能受限的 SIN 环境。

改进 HEFT 算法的流程图如图 3 所示, 同传统 HEFT 算法类似, 改进 HEFT 算法主要分为两个阶段: ① 子业务排序阶段: 将所有任务按照规则定义优先级, 形成业务子步骤调度队列; ② 节点选择阶段: 基于业务子步骤调度队列, 将子业务调度到完成时间最早的节点上, 本文对 HEFT 作出的改进均位于节点选择阶段。

在子业务排序阶段, 与传统 HEFT 算法相同, 基于 upward rank 方式对子业务进行排序

$$\text{rank}_u(n_i) = \bar{w}_i + \max_{n_j \in \text{succ}(n_i)} (\bar{c}_{ij} + \text{rank}_u(n_j)) \quad (17)$$

其中, $\text{succ}(n_i)$ 是子业务 n_i 的后继子业务, \bar{c}_{ij} 是子任务 n_i 和子任务 n_j 之间的通信开销, \bar{w}_i 是任务 n_i 的平均计算开销。

在节点选择阶段, 为了适应实际 SIN 网络的动态性与异构性, 本文对传统 HEFT 算法作出如下改进。

首先, 为了适应 SIN 网络拓扑的动态变化, 基于上述 ETEG 模型, 本文增加了跨时隙机制。具体来说, 需要进一步判断完成子任务 i 的最短时间是否小于当前时隙的剩余时间。若小于, 则将子任

务 i 在当前时隙进行调度; 否则, 需进入下一时隙, 依托下一时隙的网络拓扑完成对子任务 i 的调度。其次, 针对 SIN 网络中节点功能的异构性, 本文在传统 HEFT 算法的基础上, 引入了节点的功能约束。具体来说, 构建具备子任务 i 功能的节点集合, 并在该集合内, 依托式(15)计算子任务 i 的最短完成时间, 从而为子任务 i 的调度提供基础。

基于上述两点改进策略, 可实现在动态异构的 SIN 网络中的任务调度。

在节点选择阶段, 具体步骤如下:

① 从排好序的任务队列中取出第一个任务, 构建具备该任务功能的节点集合, 并在集合内, 根据式(15)计算对应的子任务的最短完成时间。

② 判断该最短完成时间是否小于当前时隙的剩余时间; 若小于, 则分配子任务到对应节点; 否则, 数据在前向任务节点上缓存, 进入下一时隙。

③ 重复上述过程, 直到所有队列中的子业务分配完毕。

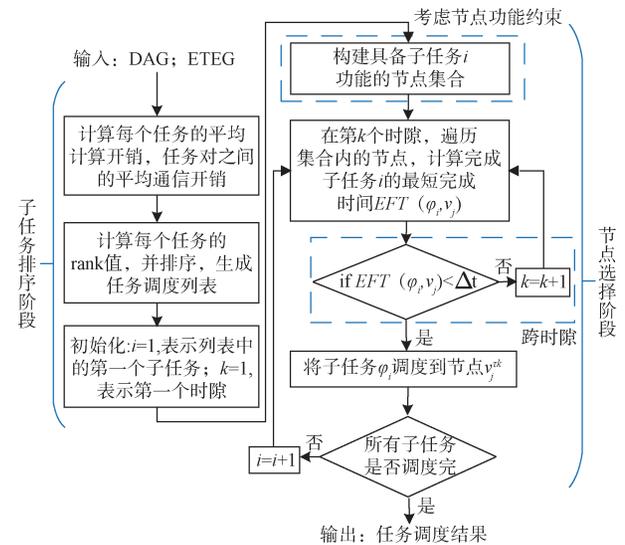


图 3 改进 HEFT 算法流程图

Fig. 3 The flowchart of the improved HEFT algorithm

2 仿真与评估

为验证本文提出的在轨协同计算的有效性, 基于改进 HEFT 算法, 本节比较了不同计算模式的时延性能。同时, 对比分析了改进 HEFT 算法与加权轮转法(WRR)、随机动态算法(Pick-KX)、贪婪

负载均衡算法(Greedy-LB)的时延及收敛性能。仿真中,为不失一般性,所有实验均基于1000次蒙特卡洛实验。

仿真平台采用Python 3.7,所处理的任务为卫星遥感图像处理任务^[18],其DAG如图4所示。仿真采用卫星为Starlink星座的15颗卫星,其轨道参数由文献[19]获得,仿真时隙长度设置为5 min。参考文献[20,21],仿真参数如表1所示。

表1 基本参数

Table 1 The system parameters

参数	值
C_0	[150,500] Mbps
η_i	0.8
α_i	1 900/8 cycles/bit
f_s	[0.5,1] GHz
f_c	10 GHz

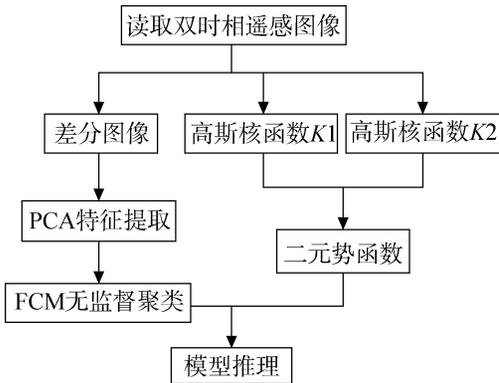


图4 高分辨率遥感图像处理任务DAG模型

Fig. 4 The DAG model for high resolution remote sensing image processing task

2.1 卫星协同计算与地面云计算、本地计算时延性能对比

图5比较了不同任务量下不同计算范式的时延性能(即在轨协同计算、云计算与本地计算)。如图5所示,随着任务量的增加,所有计算范式的时延都会增加,这是因为更多的任务量会带来更高的计算时延和通信时延。同时,在不同任务量下,在轨协同计算的任务处理时延低于云计算和本地计算。这是因为在云计算模式中,需要将原始数据发送至距离较远的地面云计算平台,导致其传输时延远高于其他方案。在本地计算模式中,虽可避免较高的通信开销,但单颗卫星的计算能力有限,导致其计算时延高于在轨协同计算。而在

轨协同计算将业务卸载到传输路径的多颗卫星上,对路由与任务卸载进行联合优化,一方面避免了云计算模式中长距离传输导致的高传输时延,另一方面联合多颗卫星的计算资源,降低了业务的计算时延。因此,在处理不同任务量时,在轨协同计算的任务处理时延均低于云计算和本地计算。

此外,在轨协同计算模式下,任务量为350 Mb时,业务处理时延的中位值由241 s跳跃式增长至307 s。这是因为随着任务量增大,任务不能在当前时隙完成,需缓存到下一时隙,此缓存过程导致传输时延增加,最终,业务处理时延出现跳跃式增长。

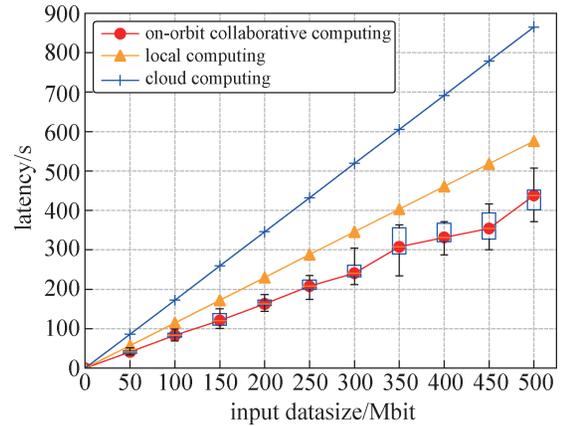


图5 卫星协同计算与云计算、本地计算时延性能对比

Fig. 5 Latency performance comparison of satellite collaborative computing versus cloud computing and local computing

2.2 多种算法时延及收敛性能比较

本小节比较了改进HEFT算法与贪婪负载均衡算法(Greedy-LB)、加权轮转法(WRR)和随机动态算法(Pick-KX)的时延及收敛性能。其中, Greedy-LB算法根据当前节点负载情况进行节点映射,将子任务分配到当前负载最少的节点上; WRR算法根据节点的处理能力大小为其赋予权值,按权值的高低和轮转方式将子任务分配到网络节点上; Pick-KX算法在收到任务请求后,首先随机选择若干个节点,再依托其负载情况,将子任务分配到所选节点集合中的某个节点上。在完成节点映射后,根据式(12)定义的边映射规则进行边映射,并最终形成任务调度方案。

图6比较了改进HEFT算法与不同算法的时延

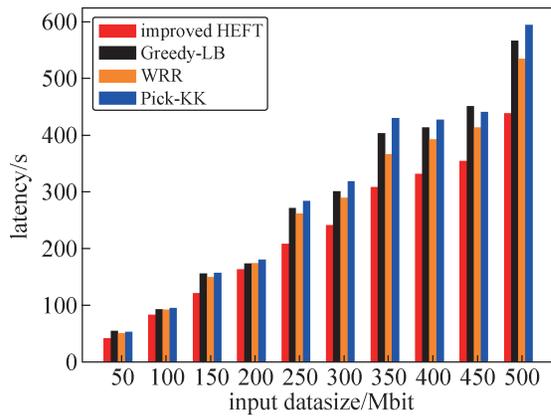


图 6 多种算法时延性能比较

Fig. 6 Latency performance comparison of different algorithms

性能。由图 6 可知, 随着数据量的增加, 所提算法的时延明显低于其他算法。例如, 当数据量为 500 Mb 时, 改进 HEFT、Greedy-LB、WRR、Pick-KX 算法的时延分别为 438.58 s、566.50 s、534.62 s、594.73 s, 改进 HEFT 算法相比于其他算法, 时延分别降低了 127.92 s、96.04 s、156.15 s。这是因为 Greedy-LB 和 WRR 仅考虑了卫星的计算能力与负载, 未考虑网络的传输能力, 而 Pick-KX 算法随机选择卫星节点, 未考虑其计算能力。改进 HEFT 算法综合考虑了节点的计算能力和网络的传输能力, 可有效降低业务处理时延。

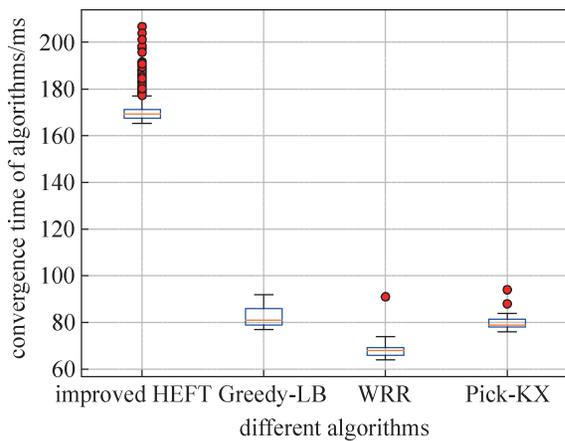


图 7 多种算法收敛性能比较

Fig. 7 Convergence performance comparison of different algorithms

图 7 比较了改进 HEFT 算法与不同算法的收敛性能, 其中任务量为 500 Mb。由图 7 可知, 改进 HEFT 算法的收敛时间略高于其他算法。例如, 改

进 HEFT、Greedy-LB、WRR、Pick-KX 算法在 1 000 次实验中的算法运行时间中位值分别为: 169 ms、81 ms、68 ms、79 ms, 相比其他算法, 改进 HEFT 算法运行时间分别增加了 88 ms、101 ms、90 ms。这是因为, 改进 HEFT 算法在进行映射时, 需要根据当前节点计算能力与网络传输能力计算子任务的完成时延, 并依托该时延对当前子任务进行分配, 而其他算法仅仅依托当前节点负载/计算能力进行子任务节点映射, 省去了复杂的时延计算过程。

但应注意到, 虽然改进 HEFT 算法的收敛时间略高于其他算法, 但其收敛时间的增长幅度是毫秒级的, 仍能以较快的速度生成多星在轨协同计算策略, 适用于实际 SIN 系统。同时, 所提算法的任务处理时延明显低于其他算法, 其以较小且可接受的数十毫秒收敛时间为代价, 换取了业务处理时延数百秒的大幅度降低。

3 结束语

本文面向高动态、资源异构 SIN 环境, 提出了基于算力路由的低时延在轨协同计算策略。首先, 构建 ETEG, 在屏蔽卫星网络动态性的同时实现对卫星计算资源的管理; 其次, 通过将任务的处理步骤表示为 DAG 模型, 进而将多星协同转化为 DAG 至 ETEG 的映射; 最后, 对 HEFT 算法进行改进, 用于求解上述问题。仿真结果表明: 在轨协同计算的任务处理时延明显低于云计算和本地计算。

参考文献

[1] 郝玉龙, 孙阳, 李冰. 基于云计算的卫星地面应用系统设计[J]. 计算机应用与软件, 2012, 29(4): 216-219.
HAO Yulong, SUN Yang, LI Bing. Cloud computing based satellite ground application system design[J]. Computer Applications and Software, 2012, 29(4): 216-219.

[2] 何友, 姚力波, 李刚, 等. 多源卫星信息在轨融合处理分析与展望[J]. 宇航学报, 2021, 42(1): 1-10.
HE You, YAO Libo, LI Gang, et al. Summary and future development of on-board information fusion for multi-satellite collaborative observation[J]. Journal of Astronautics, 2021, 42(1): 1-10.

[3] 何友, 姚力波, 江政杰. 基于空间信息网络的海洋目标监视分析与展望[J]. 通信学报, 2019, 40(4): 1-9.
HE You, YAO Libo, JIANG Zhengjie. Summary and future development of marine target surveillance based on spatial information network[J]. Journal on Communica-

- tions, 2019, 40(4): 1-9.
- [4] TANG Xiongyan, CAO Chang, WANG Youxiang, et al. Computing power network: The architecture of convergence of computing and networking towards 6G requirement[J]. China Communications, 2021, 18(2): 175-185.
- [5] 黄光平, 史伟强, 谭斌. 基于SRv6的算力网络资源和服务编排调度[J]. 中兴通讯技术, 2021, 27(3): 23-28.
HUANG Guangping, SHI Weiqiang, TAN Bin. Computing power network resources based on SRv6 and its service arrangement and scheduling[J]. ZTE Technology Journal, 2021, 27(3): 23-28.
- [6] PANG R, LI H, JI Y, et al. Energy-saving mechanism based on tidal characteristic in computing power network [C]// International Conference on Networking and Network Applications (NaNA), 2021.
- [7] SHU C, ZHAO Z, HAN Y, et al. Multi-user offloading for edge computing networks: A dependency-aware and latency-optimal approach[J]. IEEE Internet of Things Journal, 2020, 7(3): 1678-1689.
- [8] CHEN J, YANG Y, WANG C, et al. Multitask offloading strategy optimization based on directed acyclic graphs for edge computing[J]. IEEE Internet of Things Journal, 2022, 9(12): 9367-9378.
- [9] 国晓博, 任智源, 程文驰, 等. 低轨卫星网络中业务图驱动的星间协作计算方案[J]. 天地一体化信息网络, 2021, 2(2): 35-44.
GUO Xiaobo, REN Zhiyuan, CHENG Wenchi, et al. Inter-satellite cooperative computing scheme driven by business graph in LEO satellite network[J]. Space-Integrated-Ground Information Networks, 2021, 2(2): 35-44.
- [10] 马步云, 任智源, 李赞. 基于多维梯度的卫星集群高可靠协同计算方法[J]. 中兴通讯技术, 2021, 27(5): 36-42.
MA Buyun, REN Zhiyuan, LI Zan. Multi-dimensional gradient based high-reliability collaborative computing method for satellite network[J]. ZTE Technology Journal, 2021, 27(5): 36-42.
- [11] WANG P, ZHANG X S, ZHANG S, et al. Time-expanded graph-based resource allocation over the satellite networks[J]. IEEE Wireless Communications Letters, 2019, 8(2): 360-363.
- [12] WANG C, REN Z, CHENG W, et al. Time-expanded graph-based dispersed computing policy for LEO space satellite computing[C]// IEEE Wireless Communications and Networking Conference (WCNC), 2021.
- [13] ZHENG S, REN Z, CHENG W, et al. Minimizing the latency of embedding dependence-aware SFCs into MEC network via graph theory[C]// IEEE Global Communications Conference (GLOBECOM), 2021.
- [14] SHAH V, DEY B K, MANJUNATH D. Network flows for function computation[J]. IEEE Journal on Selected Areas in Communications, 2013, 31(4): 714-730.
- [15] ROST M and SCHMID S. Virtual network embedding approximations: Leveraging randomized rounding[J]. IEEE/ACM Transactions on Networking, 2019, 27(5): 2071-2084.
- [16] MICHAEL M, LLORCA J, TULINO A. Approximation algorithms for the optimal distribution of real-time stream-processing services[C]// IEEE International Conference on Communications (ICC), 2019.
- [17] TOPCUOGLU H, HARIRI S, WU M. Performance-effective and low-complexity task scheduling for heterogeneous computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(3): 260-274.
- [18] 周天天. 基于云计算的遥感图像变化检测高效处理方法[D]. 南京: 南京理工大学, 2019.
- [19] KELSO T S. NORAD general perturbations (GP) element sets current data[EB/OL]. (2022-05-19)[2022-05-19]. <https://celestrak.org/NORAD/elements/gp.php?GROUP=starlink&FORMAT=t1e>.
- [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] SpaceX. Starlink business[EB/OL]. [2022-05-19].<https://www.starlink.com/specifications?spec=3>.

[作者简介]

- 马步云 1998年生, 硕士研究生。
任智源 1983年生, 教授, 博士生导师。
郭凯 1982年生, 博士, 研究员。
程文驰 1986年生, 教授, 博士生导师。

(本文编辑: 傅杰)