

# 基于多维梯度的 卫星集群高可靠协同计算方法

Multi-Dimensional Gradient Based High-Reliability  
Collaborative Computing Method for Satellite Network



马步云/MA Buyun, 任智源/REN Zhiyuan, 李赞/LI Zan

(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, 中国 西安 710071)  
(State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China)

**摘要:**在分布式低轨(LEO)卫星集群中进行端到端业务计算是当前的研究热点。但是,传统分布式LEO端到端业务计算采用的集中式调度方法,存在任务执行成功率难保障、中心节点难选取等问题。针对上述问题,提出了一种新型分布式端到端业务计算技术,研究了基于时空扩展图模型的LEO拓扑动态性屏蔽方法,并提出了基于可靠性梯度选择的端到端业务计算理论模型与分布式业务调度算法。该算法可通过综合考虑LEO节点的多维信息来选取参与任务处理的路径节点,并逐跳完成业务的计算与传输。仿真结果表明,与集中式业务调度相比,分布式业务调度以较小的时延增加为代价,有效提升了任务的执行成功率。

**关键词:**低轨卫星;分布式;端到端;协同计算

**Abstract:** It has become a hot research field to study end-to-end task computing in Low Earth Orbit (LEO) network. However, there exist difficulties to guarantee the task execution success rate and elect the management node caused by the centralized task scheduling. To solve the above problems, the distributed end-to-end information processing technology is proposed. At first, the dynamic of the LEO network is shielded based on the time expanded graph model; then, the end-to-end task computing model is constructed and the distributed task scheduling algorithm is proposed, which jointly considers the multi-dimensional information of nodes to select the processing nodes. The simulation results validate that the distributed task scheduling effectively improves the task execution success rate at the cost of little latency performance.

**Keywords:** Low Earth Orbit; distribution; end-to-end; collaborative computing

DOI: 10.12142/ZTETJ.202105008  
网络出版地址: <https://kns.cnki.net/kcms/detail/34.1228.TN.20210929.1604.002.html>

网络出版日期: 2021-09-29  
收稿日期: 2021-08-18

由于具有星地传输距离短、覆盖范围广等优势,基于低轨(LEO)卫星的通信系统<sup>[1]</sup>受到业界广泛关注。同时,大量数据在传输过程中仍需进一步处理才能被使用(例如,卫星采集的图像需要经过去噪、特征提取等后才可被使用)。然而,

受限于卫星的载荷能力和宇宙射线的影响<sup>[2]</sup>,单颗卫星的计算能力难以大幅提升,很难独自完成计算密集型任务。而将海量数据转发至地面云计算中心,利用云平台强大的计算资源处理数据<sup>[3]</sup>,虽然可有效降低计算时延,但是会带来过高的通信开销,仍难以有效满足业务需求。因此,研究端到端业务计算方法势在必行。通过协作可使卫星展现出强大的传

输与计算数据的能力。

目前,大多数研究者致力于单方面优化路由<sup>[4-6]</sup>或业务卸载策略<sup>[7-10]</sup>,将两者统一考虑的很少。而现有的端到端信息处理方案均为集中式业务调度<sup>[11-13]</sup>,其中,中心管理节点负责管理网络并制订合理的业务调度方案,LEO集群根据预先制订好的方案相互协作。然而,LEO卫星数目众多且计算资源有限,真实的卫星网络很难

基金项目:科技部重点研发计划资助项目(2019YFB1803305)

拥有一个强大的中心管理节点(该节点一旦发生故障,整个网络将瘫痪)。此外,由于卫星工作在复杂的宇宙环境中,极易受到干扰,如采用集中式调度模式处理业务,调度方案中的任何一颗卫星出现故障都将导致任务失败,很难满足业务的可靠性需求。基于此,针对单星计算能力弱、节点故障率高的分布式LEO集群,亟需一种分布式低时延高可靠的端到端业务计算方法,以满足业务需求。

本文面向分布式LEO集群,提出了一种去中心式端到端信息处理技术方法。该方法首先依托时空扩展图(TEG)来屏蔽LEO集群的高动态特性,随后对端到端业务调度进行理论建模并设计分布式业务调度算法。当任务到来时,每颗卫星基于其邻居节点信息,独自运行该算法来选择下一跳节点,并逐步完成数据的传输与计算。该算法提出了一种新的度量梯度指标(业务调度效率)作为选择下一跳节点的依据。该梯度指标综合考虑了节点的计算能力、链路传输速率、故障率、至目标卫星的跳数,可有效降低系统时延,提高系统可靠性。

### 1 系统模型

分布式LEO集群系统架构如图1所示。其中,为不失一般性,假设地面站定时向LEO集群广播全局拓扑信息,每颗卫星可计算自身到结果接收卫星的跳数。当任务到达时,每颗卫星根据自身相邻节点的信息逐步选择下一节点,并完成端到端业务计算。

#### 1.1 LEO网络模型

为屏蔽LEO集群的动态性,本节依托LEO卫星运行轨道参数构建TEG模型。

令  $N = \{n_1, \dots, n_p, \dots, n_s\}$  表示LEO集群,以地心为坐标原点,以赤道平面为X轴、Y轴所在平面,Z轴通过地心并垂直于赤道平面指向北极,建立空间直角坐标系。则在任意时刻  $t$  时,  $n_p (n_p \in N)$  的位置坐标  $(x_p^t, y_p^t, z_p^t)$  可通过运行轨道计算得到。 $n_p$  与  $n_o (n_p, n_o \in N, p \neq o)$  之间的距离可通过式(1)来计算。

$$d_{po}^t = \sqrt{(x_p^t - x_o^t)^2 + (y_p^t - y_o^t)^2 + (z_p^t - z_o^t)^2} \quad (1)$$

定义  $t$  时刻  $n_p$  与  $n_o$  之间的链路状态为  $state_{po}^t$ ,并可表示为式(2):

$$state_{po}^t = \begin{cases} r^*, r_{po}^t \geq r^* \\ 0, r_{po}^t < r^* \end{cases} \quad (2)$$

其中,  $r^*$  为星间链路的设计速率,  $r_{po}^t$  为  $t$  时刻  $n_p$  与  $n_o$  的理论传输速率。 $state_{po}^t = r^*$  表示  $n_p$  与  $n_o$  连通且链路传输速率为  $r^*$ ,反之则表示  $n_p$  与  $n_o$  链路中断。可根据香农公式得出式(3):

$$r_{po}^t = B \log_2 \left( 1 + \frac{P_r^t}{\sigma^2 B} \right) \quad (3)$$

其中,  $B$  为星间链路带宽,  $\sigma^2$  为高斯白噪声方差,  $P_r^t$  为  $t$  时刻的信号接收功率。在星间链路中,信号传输损耗主要为自由空间传输损耗<sup>[4]</sup>。因此,  $P_r^t$  可由式(4)来表示:

$$P_r^t = \frac{P_t G_t G_r}{(4\pi d_{po}^t / \lambda)^2} \quad (4)$$

其中,  $G_r, P_t, G_t$  分别表示信号接收增益系数、信号发射功率和信号发射增益系数,  $\lambda$  为载波波长。则式(2)可进一步表示为:

$$state_{po}^t = \begin{cases} r^*, B \log_2 \left( 1 + \frac{P_t G_t G_r}{(4\pi d_{po}^t / \lambda)^2 \sigma^2 B} \right) \geq r^* \\ 0, B \log_2 \left( 1 + \frac{P_t G_t G_r}{(4\pi d_{po}^t / \lambda)^2 \sigma^2 B} \right) < r^* \end{cases} \quad (5)$$

基于式(5),通过遍历可获得LEO集群拓扑。此时,以LEO集群某一时刻状态为起点,将系统运行周期  $T$  等分为  $n$  个连续时隙,长度定义为  $\Delta = T/n$ 。假设时隙内拓扑稳定不变,



▲图1 低轨集群系统架构图

则LEO集群 $N$ 可表示为 $N = (N^T, E^T)$ , 其中 $N^T = \{N^1, \dots, N^n\}$ 为节点集合,  $E^T$ 为边集合, 如图2所示。

(1)时隙内边的权重。任意时隙 $\forall q \in T$ 内, 边 $(n_p^q, n_o^q) \in E^T$ 的权重为节点 $n_p^q$ 传输单位数据量到节点 $n_o^q$ 的时延, 如式(6)所示:

$$\pi_{po}^q = \begin{cases} \frac{1}{r^*}, state_{po}^t = r^*, t \in q \\ 0, state_{po}^t = 0, t \in q \end{cases} \quad (6)$$

则 $q$ 时隙内LEO集群可表示为式(7):

$$G^q = \begin{bmatrix} 0 & \pi_{12}^q & \dots & \pi_{1(d-1)}^q & \pi_{1d}^q \\ \pi_{21}^q & 0 & \dots & \pi_{2(d-1)}^q & \pi_{2d}^q \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \pi_{(d-1)1}^q & \pi_{(d-1)2}^q & \dots & 0 & \pi_{(d-1)d}^q \\ \pi_{d1}^q & \pi_{d2}^q & \dots & \pi_{d(d-1)}^q & 0 \end{bmatrix} \quad (7)$$

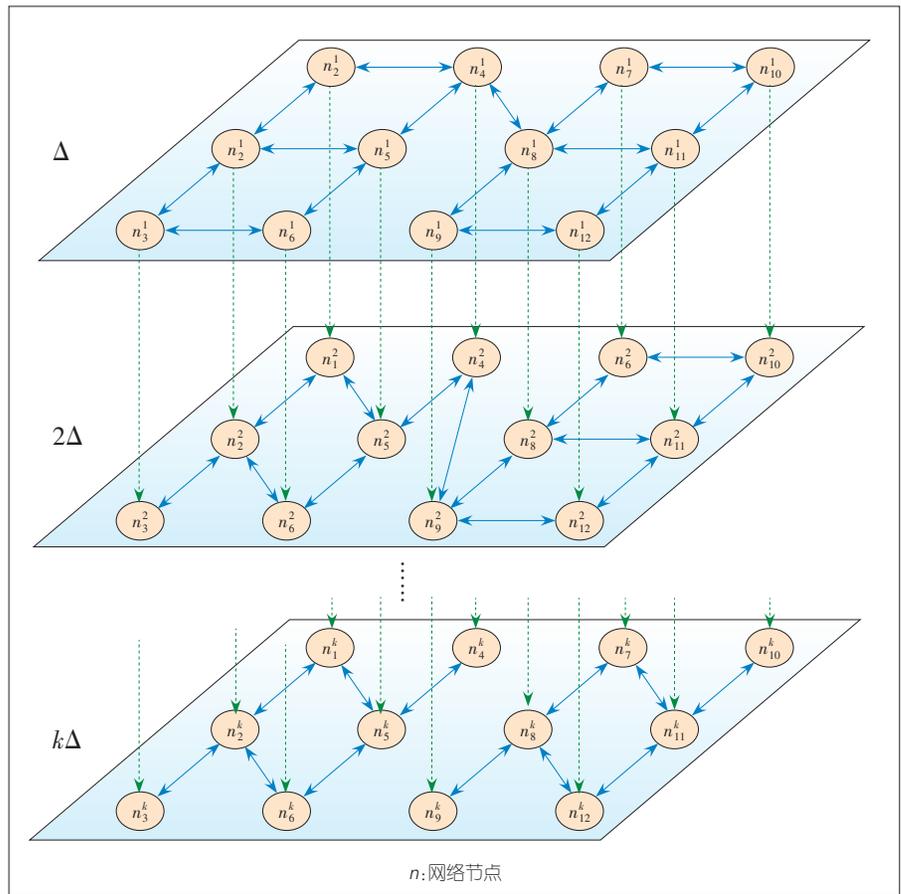
(2)时隙间边的权重。数据在传输过程中可能存在由链路中断所导致传输失败的情况, 因此, 需要定义时隙间边的权重 $\pi_p^{q(q+1)}$ , 即数据到达卫星节点 $v_p^q$ 时, 当前时隙的剩余时间, 如式(8)所示:

$$\pi_p^{q(q+1)} = \Delta t - t_p^q \quad (8)$$

则相邻时隙 $q, q+1 \in T$ 间LEO集群可表示为式(9):

$$G^{q(q+1)} = \begin{bmatrix} \pi_1^{q(q+1)} & \dots & \infty & \dots & \infty \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \infty & \dots & \pi_p^{q(q+1)} & \dots & \infty \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \infty & \dots & \infty & \dots & \pi_d^{q(q+1)} \end{bmatrix} \quad (9)$$

此时, LEO集群的TEG模型可表示为式(10):



▲图2 低轨集群时空扩展图模型

$$TEG = \begin{bmatrix} G^1 & G^{12} & \infty & \dots & \infty & \infty \\ \infty & G^2 & G^{23} & \dots & \infty & \infty \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \infty & \infty & \infty & \dots & G^{(n-1)} & G^{(n-1)n} \\ \infty & \infty & \infty & \dots & \infty & G^n \end{bmatrix} \quad (10)$$

对于任意连通节点对 $n_p^q, n_o^w \in N^T$ , 基于TEG可通过最短路径算法(如Dijkstra)求得其最短时延 $g_{po}^{qw}$ 和最短路由 $Path_{po}^{qw}$ 。

### 1.2 端到端业务计算理论模型

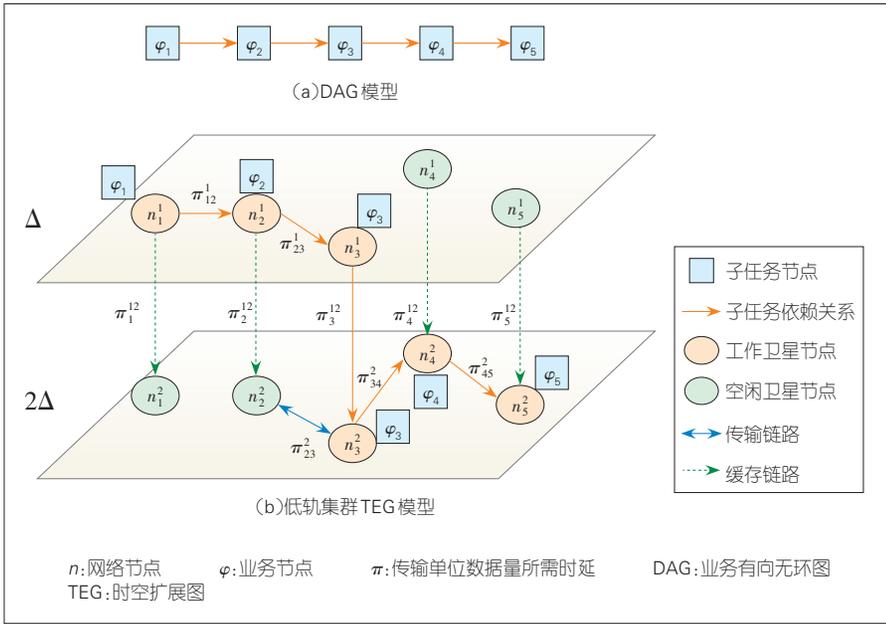
基于TEG, 本节提出端到端业务计算理论模型。为不失一般性, 本节按照子业务间的依赖关系建立业务有向无环图(DAG)模型。同时, 根据文献[15], 任何结构的DAG均可解析为串行DAG, 因此, 本文仅考虑串行

DAG。

定义DAG为 $\Omega = (\Psi, s)$ 。其中,  $\Psi = \{\varphi_1, \dots, \varphi_l\}$ 为节点集合, 表示子业务集群,  $\varphi_i$ 为业务起点,  $\varphi_i$ 为业务终点;  $s$ 为边集合, 表示子业务间的依赖关系。此外,  $\varphi_i \in \Psi$ 由元组 $\{D_i, \eta_i, \varepsilon_i\}$ 表征, 其中 $D_i$ 为输入数据量,  $\eta_i$ 为数据压缩系数,  $\varepsilon_i$ 为计算复杂度系数。同时, 定义 $\Theta_1(\varphi_j) = \{\varphi_i | (\varphi_j, \varphi_i) \in s\}$ 为子任务 $\varphi_j$ 的先驱节点集合。此时, 业务 $\Omega$ 在LEO集群中的调度可转化为DAG至TEG的映射规则, 如图3所示。

#### (1) 节点映射规则

我们首先定义 $\tau: \Psi \rightarrow N^T$ 表示子业务节点 $\Psi$ 至卫星节点 $N^T$ 的映射。具体地, 如式(11)所示, 业务起



▲图3 DAG至TEG的映射示例

点映射至业务发起卫星,业务终点映射至结果接收卫星,中间业务节点映射至任意卫星。为不失一般性,假设子业务不可再分,所有子业务均在单颗卫星上计算,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, $\rho_i$ 为跨时隙数目。

$$\tau(\varphi_i) = \begin{cases} \{n_1^1, \dots, n_1^{1+\rho_i}\}, \varphi_i = \varphi_1 \\ \{n_d^q, \dots, n_d^{q+\rho_i}\}, \varphi_i = \varphi_i \\ \{n_p^q, \dots, n_p^{q+\rho_i}\}, \text{otherwise} \end{cases} \quad (11)$$

(2) 边映射规则

$s \rightarrow E^T$  表示 DAG 有向边  $s$  至 TEG 无向边  $E^T$  的映射,以反映子业务间的依赖关系。具体地,如式(12)所示,将 DAG 的有向边  $\forall(\varphi_i, \varphi_j) \in s$  映射为图  $N$  中  $\tau(\varphi_i)$  至  $\tau(\varphi_j)$  之间的最短路由  $Path_{\tau(\varphi_i)\tau(\varphi_j)}$

$$Z((\varphi_i, \varphi_j)) = Path_{\tau(\varphi_i)\tau(\varphi_j)} \quad (12)$$

1.3 分布式端到端业务调度算法

为了实现在分布式 LEO 集群中数据的“边传输边计算”,本节提出分布式端到端业务调度算法,如算法 1 所示。该算法主要由 3 个步骤构成:(1)任务到来时,通过广播发现邻居节点,并获取其必要的状态信息以用于计算任务调度效率(TSE);(2)计算邻居节点的 TSE,并根据 TSE 选择下一跳节点;(3)判断当前时隙剩余时间是否充足,若充足则将数据发给已确定好的下一跳节点,否则返回步骤 2,并基于下时隙信息重新选择下一跳节点。

基于上述端到端业务计算理论模型分析,算法需统一考虑节点的计算能力和链路状态以实现端到端业务计算,而由于缺乏中心节点的统一调度,仅考虑计算能力和链路状态可能会导致数据的反向传输。因此,需要引入目标节点位置信息以实现数据的定向传输,同时为了保证数据传输的可靠性,节点故障率也需要被考虑进算法中。基于以上分析,本节定义 TSE 梯度指标,综合考虑了节点的

计算能力、链路状态、故障率、距目标节点跳数多维梯度信息,如式(13)所示:

$$TSE(\tau(\varphi_i)) = H_{\tau(\varphi_i)\tau(\varphi_j)} \cdot \chi_{\tau(\varphi_i)} \cdot \left( \frac{1}{f_{\tau(\varphi_i)}} + \frac{1}{e_{\tau(\varphi_i)\tau(\varphi_j)}} \right) \quad (13)$$

其中, $H_{\tau(\varphi_i)\tau(\varphi_j)}$  为映射节点  $\tau(\varphi_i)$  至结果接收卫星  $\tau(\varphi_j)$  沿最短路径所需跳数, $\chi_{\tau(\varphi_i)}$  为节点  $\tau(\varphi_i)$  的故障率, $f_{\tau(\varphi_i)}$  为节点  $\tau(\varphi_i)$  的计算能力, $e_{\tau(\varphi_i)\tau(\varphi_j)}$  为子任务  $\varphi_i$  的前向节点  $\varphi_j$  的映射节点  $\tau(\varphi_j)$  沿最短路径至  $\tau(\varphi_i)$  的传输速率。由式(13)可知,距目标节点越近,节点计算能力越强,故障率越低、链路传输速率越快,TSE 就越小,该节点的调度效率也就越高。

算法 1 分布式端到端业务调度算法

输入: DAG 模型, TEG  
 步骤 1: 任务到来时,通过广播发现邻居节点并获取其多维状态信息,包括计算能力、链路状态、故障率、距目标节点跳数;  
 步骤 2: 根据式(13)计算各邻居节点的 TSE 指标,并选取 TSE 最小的节点为下一跳节点;  
 步骤 3: 判断此时将数据传输至下一跳节点的时延是否小于当前剩余时隙,若小于则传输;否则就缓存数据,返回步骤 2,并根据 TEG 预测下时隙的 TSE 指标,重新选择下一跳节点。  
 输出: 下一跳节点

2 仿真与评估

为验证本文提出的分布式业务调度方案的有效性,本节将该方案同集中式方案进行比较。在比较过程中,所有实验均基于相同假设。在集中式业务调度模式下,中心节点运行

集中式业务调度算法以获取传输路径上的关键计算节点。集中式业务调度算法采用经典的 DAG 调度算法—异态最早结束时间 (HEFT) 算法<sup>[15]</sup>。值得注意的是,由于集中式业务调度算法依赖较多的计算资源,卫星节点虽具备一定计算能力,但很难运行集中式业务调度算法。本节同时将基于 TSE 指标选择下一跳节点的分布式业务调度算法(记为 Proposed)同随机式(记为 Random)和贪婪式(记为 Greedy)两种常用业务调度算法进行比较,并对仿真结果进行分析与讨论。

### 2.1 仿真场景及参数设置

本文考虑由 15 颗低轨卫星构成的卫星集群。其中,低轨卫星均取自铱星星座(轨道高度 780 km)。本文中,我们利用卫星工具包(STK)获取网络真实连通情况。仿真时间段为 2021 年 4 月 26 日 00:00—00:30。本文仿真平台为 Python 3.7,采用的业务图为图 1 中的 DAG。参照文献[11]和[16],仿真参数如表 1 所示。此外,为不失一般性,本文所有仿真结果均基于 3 000 次蒙特卡洛实验。

为了分析与评估性能,我们考虑端到端业务处理时延和任务成功率两个指标。

#### (1) 端到端业务处理时延

基于 1.2 节的 DAG 至 TEG 的调度规则,端到端业务处理时延可建模如下。

进行到子任务  $\varphi_i$  时的处理时延如式(14)所示:

$$T(\varphi_i) = T_{\text{comp}}(\varphi_i) + T_{\text{accu}}(\varphi_i) = \frac{D_i \eta_i \varepsilon_i}{f_{\tau(\varphi_i)}} + \max_{\varphi_j \in \phi_{\tau(\varphi_i)}} [T(\varphi_j) + g_{\tau(\varphi_j)\tau(\varphi_i)} D_j \eta_j], \quad (14)$$

其中,  $T_{\text{comp}}(\varphi_i)$  表示  $\varphi_i$  的计算时延,

▼表1 基本参数

具体参数	数值	具体参数	数值
$G_r$	37.5 dB	$G_t$	52.5 dB
$P_i$	5 kW	$\sigma^2$	$5 \times 10^{-15}$
$\varepsilon_i$	237 cycles/bit	$\eta_i$	0.8
$B_{sc}$	10 Mbit/s	$r^*$	30 Mbit/s
$f_c$	10 GHz	$f_s$	[3,5.5] GHz

$T_{\text{accu}}(\varphi_i)$  表示  $\varphi_i$  前向节点的累积时延。 $f_{\tau(\varphi_i)}$  为节点  $\tau(\varphi_i)$  的计算能力,表示卫星节点  $\tau(\varphi_i)$  中央处理器(CPU)每秒运行的周期数。

因此,  $\Omega$  的业务处理时延为最后一个子任务  $\varphi_l$  的处理时延,如式(15)所示。

$$T(\Phi) = T(\varphi_l) = \max_{\varphi_i \in \phi_{\tau(\varphi_l)}} [T(\varphi_i) + g_{\tau(\varphi_i)\tau(\varphi_l)} D_i \eta_i \varepsilon_i] + \frac{D_l \eta_l \varepsilon_l}{\rho_{\tau(\varphi_l)}}. \quad (15)$$

#### (2) 任务成功率 $\alpha$

任务成功率  $\alpha$  是成功完成的任务数与总试验次数的比值,如式(6)所示。

$$\alpha = \frac{N_{\text{succ}}}{N_{\text{total}}} \times 100\%, \quad (16)$$

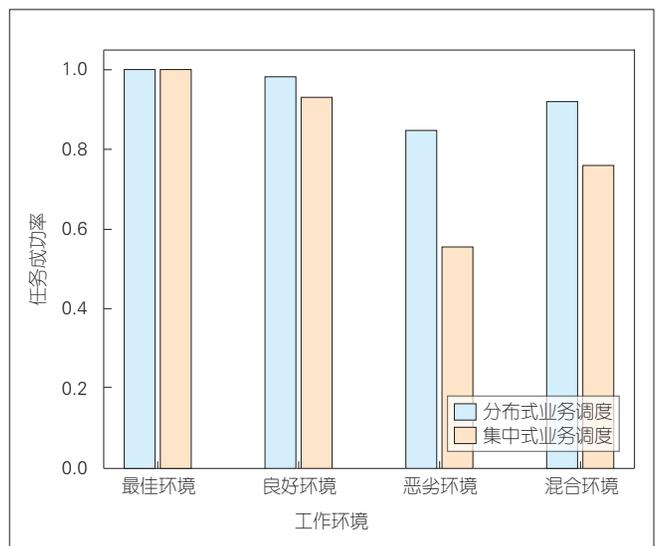
其中,  $N_{\text{succ}}$  为成功完成的任务数,  $N_{\text{total}}$  为总实验次数。

### 2.2 仿真结果与分析

#### 2.2.1 可靠性性能

图 4 比较了不同业务调度模式在不同环境下的可靠性性能。其中,任务量大小为 100 Mbit。值得注意的是,卫星的故障概率包括卫星器件故障概率和卫星受到环境干

扰(如发生“0-1 翻转”等)导致任务失败的故障概率。因此,为不失一般性,本节设置了 4 种不同环境:最佳环境、良好环境、恶劣环境、混合环境。在最佳环境中,卫星的故障概率设置为 0,即  $\chi_i = 0$ ;在良好环境中,假定卫星的故障概率均匀分布,即  $\chi_i \sim U([0,0.5\%])$ ;在恶劣环境中,  $\chi_i \sim U([1\%,3\%])$ ;在混合环境中,某些卫星的故障概率为  $\chi_i \sim U([0,0.5\%])$ ,另外一些卫星的故障概率为  $\chi_i \sim U([1\%,3\%])$ 。由图 4 可知,在最佳环境下,分布式业务调度和集中式业务调度的任务成功率均为 100%。这是因为在理想环境中,不会出现卫星故障,任务能 100% 完成。然而,由于理想情况根本不存在,因此本文研究了 3 种现实环境下的可靠性性能。由图 5 可知,集中式业务调度模式的可靠性性能在各种环境下均比较低。



▲图 4 不同环境下不同业务调度模式的可靠性性能比较

恶劣环境中,集中式业务调度模式的任务成功率仅为55.0%。相比之下,分布式业务调度的任务成功率为84.4%。这是因为,分布式业务调度仅须保障当前执行业务节点在执行业务期间不会发生故障,而集中式业务调度模式须保障业务调度方案中所有节点在执行任务之前均不会发生故障。

### 2.2.2 时延性能

图5比较了不同计算范式的时延性能,即云计算、本地计算和协同计算。其中,协同计算可进一步分为集中式业务调度和分布式业务调度,并且工作环境为混合环境。由图5可知,当任务量较小时,3种计算范式均表现出良好的时延性能。但随着任务量的增加,云计算的时延也迅速增加。这是因为云计算中心距卫星较远,导致传输时延较高。而本地计算虽可避免较高的通信开销,但由于单星计算能力有限,计算时延也较高。对于协同计算,由于卫星集群具备强大的计算能力,且卫星之间距离较近,因此,随着数据量的增加,其时延仍在可接受范围之内。

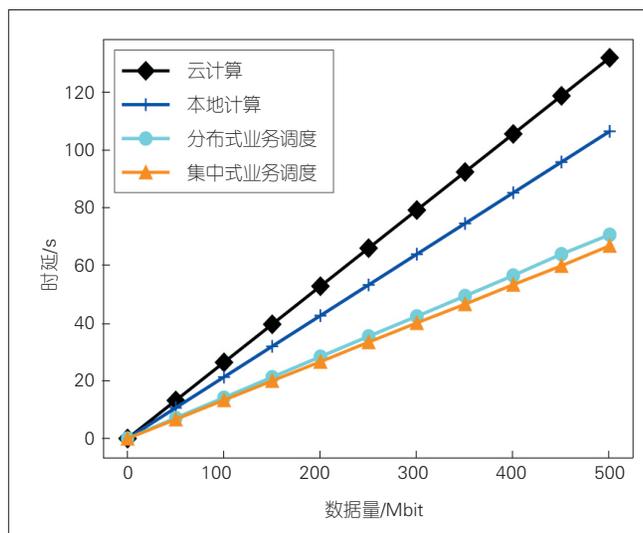
由图5可知,分布式业务调度的时延略高于集中式业务调度。但应注意到,混合环境下,在处理100 Mbit的数据时,分布式业务调度的任务成功率比集中式业务调度提升了21.3%,而时延仅增加了6.21%,即以较小且可接受的时延为代价换取了可靠性性能的大幅度提升。

### 2.2.3 多种算法可靠性及时延性能分析

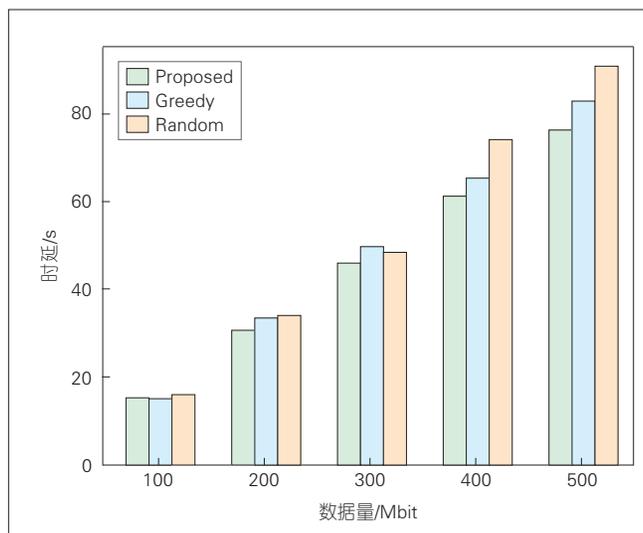
本节比较了基于TSE指标的算法(记为Proposed)同随机式(记为Random)和贪婪式(记为Greedy)算法的时延性能和可靠性性能,分别如图6、图7所示。

图6比较了不同算法的时延性能。其中,工作环境为混合环境。由图6可知,当任务量较小时,3种算法时延差别不明显。而随着任务量的增加,所提算法的时延明显低于其他两种算法。例如,当任务量为500 Mbit时,Proposed、Greedy、Random的时延具体分别为76.14 s、83.08 s、90.94 s,所提算法比其他两种算法的时延分别低了8.35%、16.27%。这是因为,Random算法随机选取下一跳节点,并未考虑其计算能力,同时Greedy算法选取计算能力最强的节点作为下一跳节点,并未考虑边的传输能力和传输方向,因此时延性能均不如Proposed算法。

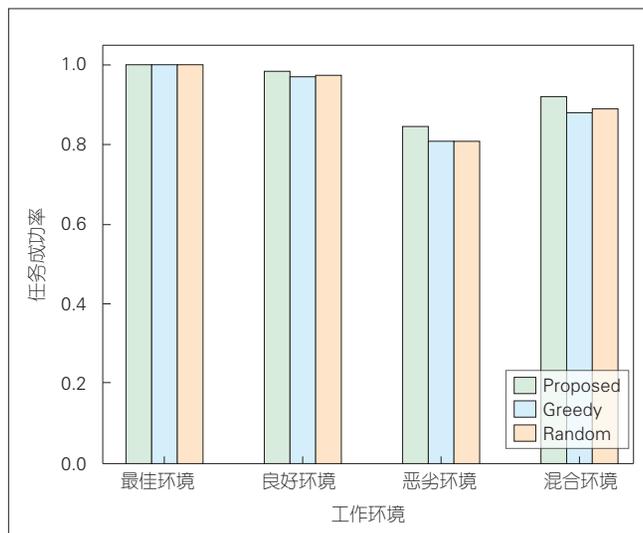
图7比较了不同算法的可靠性性能,其中,任务量为100 Mbit。可以看出,除最佳环境外,在其他环境下所提算法的任务成功率均高于其他两种算法。



▲图5 不同计算范式的时延性能比较



▲图6 不同算法时延性能比较



▲图7 不同算法可靠性性能比较

这是因为 Random 和 Greedy 算法在选择下一跳节点时,均未考虑节点的故障概率,因此可靠性性能不如所提算法。

### 3 结束语

本文面向分布式 LEO 集群,提出了分布式端到端信息处理技术。首先我们构建 TEG 将 LEO 集群动态拓扑稳定化,随后构建端到端信息处理理论模型并提出分布式业务调度算法。该算法通过综合考虑计算资源、通信资源、至目标节点跳数、节点故障率多维信息来选取下一跳节点,并逐步完成数据的传输与计算。仿真结果表明,所提分布式业务调度技术以牺牲较小时延为代价,有效地提升了业务的执行成功率。

### 致谢

本研究得到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程文驰老师的帮助,谨致谢意!

### 参考文献

- [1] 王艳峰,谷林海,刘鸿鹏.低轨卫星移动通信现状与未来发展[J].通信技术,2020,53(10):2447-2453. DOI: 10.3969/j. issn. 1002-0802.2020.10.014
- [2] BRAVHAR K, MARTINS V, SANTOS L, et al. BRAVE NG-MEDIUM FPGA reconfiguration

- through SpaceWire: example use case and performance analysis [C]//2018 NASA/ESA conference on adaptive hardware and systems (AHS). Edinburgh, UK: IEEE, 2018: 135-141. DOI: 10.1109/AHS.2018.8541461
- [3] 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. DOI: 10.3969/j. issn. 1000-386X.2012.04.061
- [4] 李红艳,张焱,张靖乾,等.基于时变图的天地一体化网络时间确定性路由算法与协议[J].通信学报,2020,41(10):116-129
- [5] ZHANG T, LI H Y, ZHANG S, et al. STAG-based QoS support routing strategy for multiple missions over the satellite networks [J]. IEEE transactions on communications, 2019, 67(10): 6912-6924. DOI: 10.1109/TCOMM.2019.2929757
- [6] LV Y, XING C, XU N, et al. Research of adaptive routing scheme for LEO network [C]//2019 IEEE 5th International Conference on Computer and Communications (ICCC). Chengdu, China: IEEE, 2019: 987-992. DOI: 10.1109/ICCC47050.2019.9064312
- [7] TANG Q Q, FEI Z S, LI B, et al. Computation offloading in LEO satellite networks with hybrid cloud and edge computing [J]. IEEE Internet of Things journal, 2021, 8(11): 9164-9176. DOI:10.1109/JIOT.2021.3056569
- [8] SONG Z Y, HAO Y Y, LIU Y W, et al. Energy-efficient multi-access edge computing for terrestrial-satellite Internet of Things [J]. IEEE Internet of Things journal, 2021, 8(18): 14202-14218. DOI:10.1109/JIOT.2021.3068141
- [9] 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),2018,52:1474-1481
- [10] 王元君.星地混合网络中的计算资源分配和负载均衡[D].北京:北京邮电大学,2020
- [11] 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
- [12] 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. DOI: 10.7652/xjtub201610011
- [13] 马步云,马新策,黄松,等.WSN低功耗低时延路径式协同计算方法[J].无线电通信技术,2021,47(2):168-177
- [14] XUE R, YU H, CHENG Q L. Adaptive coded modulation based on continuous phase modulation for inter-satellite links of global navigation satellite systems [J]. IEEE access, 2018, 6: 20652-20662. DOI: 10.1109/ACCESS.2018.2825255
- [15] ARABNEJAD H, BARBOSA J G. List scheduling algorithm for heterogeneous systems by an optimistic cost table [J]. IEEE transac-

tions on parallel and distributed systems, 2014, 25(3): 682-694. DOI: 10.1109/TP-DS.2013.57

- [16] WANG C, REN Z Y, CHENG W C, et al. Time-expanded graph-based dispersed computing policy for LEO space satellite computing [C]//2021 IEEE Wireless Communications and Networking Conference (WCNC). Nanjing, China: IEEE, 2021: 1-6. DOI: 10.1109/WCNC49053.2021.9417247

### 作者简介



马步云,西安电子科技大学在读硕士研究生;主要研究领域为面向卫星网络、无人机网络、车载网络等高动态场景的多目标协同下的动态网络分布式计算。



任智源,西安电子科技大学副教授、博士生导师;主要研究领域为云计算、边缘计算、物联网、分散智能;已发表论文20余篇,申请国家发明专利40余项,获授权专利6项。



李赞,西安电子科技大学教授、博士生导师,教育部“长江学者特聘教授”,“中国青年科技奖”获得者;主要研究领域为突发通信、数字信号处理、无线通信系统。