Towards Low-Latency Distributed Tasks Collaboration by Joint Optimization of Transmission, Computation and Storage Resources Allocation in Edge Computing

Jialong Li[1,2], Nan Hua[1,2], Chen Zhao[1,2], Yanhe Li[3], Xiaoping Zheng[1,2], and Bingkun Zhou[1,2]

[1]*Beijing National Research Center for Information Science and Technology (BNRist), Beijing 100084, China*
[2]*Department of Electronic Engineering, Tsinghua University, Beijing 100084, China*
[3]*Tsinghua Unigroup, Tsinghua University, Beijing 100084, China*

{*huan, xpzheng*}*@mail.tsinghua.edu.cn*

Abstract: We propose a collaboration strategy by jointly considering the transmission, computation, and storage resources to reduce the completion time of distributed tasks in edge computing. Results show that over 50% completion time reduction is achieved.

1. Introduction

Edge computing is proposed to tackle the data explosion and low-latency requirements by the emerging applications, such as virtual reality (VR), vehicle to everything (V2X), and online video streaming. Compared with cloud computing, edge nodes implemented with computation and storage resources process data at the edge of the networks and provide instant response to requests. Due to the room and power supply restriction, the computation and storage resources in a single edge node are usually limited [1], and some applications are required to be processed among distributed edge nodes. The authors in [2][3] proposed a low-latency network architecture, which is applied in edge computing scenarios based on fine-grained switching technology [4]. Subscribers could be served by other edge nodes or the cloud without excessive delay. Furthermore, this architecture could provide a large number of optical channels, which is essential for the interconnects among large-scale edge nodes. However, this work does not consider the coordination among distributed tasks when a single edge node could not support the whole applications. And the completion time of distributed tasks will influence the quality of services (QoS) significantly.

In this paper, we propose a collaboration strategy by jointly considering the transmission, computation, and storage resources (JTCS). Simulation results demonstrate that JTCS could realize lower completion time and fewer transmission resource requirements compared with the traditional greedy algorithm.

2. JTCS strategy and mathematical model

A directed acyclic graph (DAG) is used to model a job [5], which is depicted in Fig. 1(a) and (b). A job consists of several tasks, which are represented by the nodes in the DAG. The directed edge denotes the computational results transmission between forward and backward tasks (Task 1 and 2 in Fig. 1(a), for example). In our model, we also consider the storage resources factor. The task may require some raw data as well as the computational results from its forward tasks to complete its execution. Raw data transmission is necessary if data storage and the task are not accommodated by the same node. The traditional greedy method, depicted in Fig. 1(c), allocates tasks to the nearest node with sufficient computational resources, which will induce the mismatch between tasks scheduling and resources, resulting in higher completion time. As Fig. 1(c) shows, the forward task is buffered for 1 time slice before its execution, while the buffering time for the backward task is 2 time slices. The overall completion time achieved by the greedy method is 9 time slices, while the number for JTCS is 6, which is shown in Fig. 1(d). JTCS aims at joint optimization of transmission, computation, and storage resources allocation when scheduling tasks to different nodes, which realizes much lower completion time.

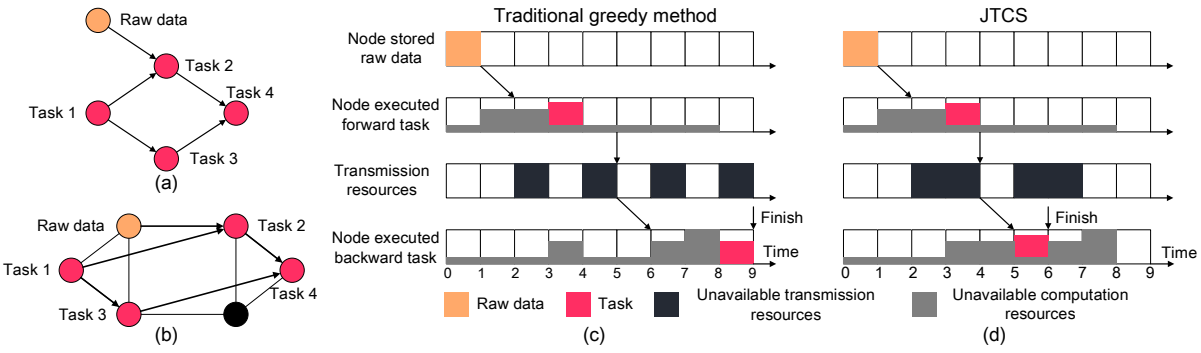


Fig. 1: (a) A job described by the DAG model. (b) Distributed tasks scheduling. (c) Greedy method. (d) JTCS.

The mathematical model for JTCS is described below. For simplicity, the execution and transmission time is measured by time slices. The propagation delay is also considered in our model. The computation resources required by tasks in this paper refer to the central processing unit (CPU) and random-access memory (RAM).

Given:

- $G(N,E)$: network topology, where N and E represent the set of nodes and fiber links, respectively.
- A: set of DAGs.
- a: a DAG, and $a \in A$.
- W: set of wavelengths in each fiber link.
- c: the speed of light in the fiber.
- (a_p,a_q): computational results transmission between tasks a_p and a_q, where a_p is the forward task and a_q is the backward task. $a_p,a_q \in a$.
- $T^{(a_p,a_q)}$: number of time slices required by tranmission (a_p,a_q).
- (b,a_p): raw data transmission for task a_p, where raw data are stored in node b initially. $b \in N$.
- $T^{(b,a_p)}$: number of time slices required by tranmission (b,a_p).
- C^{a_p}: number of time slices required by execution a_p.
- t: the length of a time slice.
- S: set of time slices in one cycle.
- $l^{(a_p,a_q)}/l^{(b,a_p)}$: lightpath used by computational results transmission (a_p,a_q) and raw data transmission (b,a_p), respectively.
- D^l/D^e: physical length of lightpath l and edge e, respectively. $e \in l$.
- CPU/RAM: CPU/RAM capacity of each node.
- cpu^{a_p}/ram^{a_p}: CPU/RAM resource requirements for task a_p.

Variables:

- $x_n^{a_p}$: binary decision variable, which is one if the task a_p is executed in node n.
- $y_{(e,w,i)}^{(a_p,a_q)}$: binary decision variable, which is one if results transmission (a_p,a_q) uses the i_{th} time slice on wavelength w in edge e.
- $z_{(e,w,i)}^{(b,a_p)}$: binary decision variable, which is one if data transmission (b,a_p) uses the i_{th} time slice on wavelength w in edge e.
- $\pi_{(e,w)}^{(a_p,a_q)}$: binary decision variable, which is one if results transmission (a_p,a_q) uses wavelength w in edge e.
- $\theta_{(n,i)}^{a_p}$: binary decision variable, which is one if the task a_p uses computation resources in the i_{th} time slice in node n.

Optimize:

$$Minimize \quad F = \sum_{a \in A} (\max(i), \quad \theta_{(n,i)}^{a_p} = 1, a_p \in a, n \in N, i \in S)/A \tag{1}$$

Constraints:

$$x_n^{a_p} + x_n^{a_q} \leqslant 1, \quad \forall n \in N, a_p \neq a_q, a_p, a_q \in a, a \in A \tag{2}$$

$$\sum_{n \in N} x_n^{a_p} = 1, \quad \forall a_p \in a, a \in A \tag{3}$$

$$T^{(b,a_p)} = \begin{cases} T^{(b,a_p)}, & x_n^{a_p} = 1, b \neq n \\ 0, & x_n^{a_p} = 1, b = n \end{cases} \quad a_p \in a, a \in A, n \in N \tag{4}$$

$$\sum_{i \in S} \theta_{(n,i)}^{a_p} = \begin{cases} C^{a_p}, & x_n^{a_p} = 1 \\ 0, & x_n^{a_p} = 0 \end{cases} \quad a_p \in a, a \in A, n \in N \tag{5}$$

$$\sum_{a \in A} \sum_{a_p \in a} x_n^{a_p} \cdot \theta_{(n,i)}^{a_p} \cdot cpu^{a_p} \leqslant CPU, \quad \forall n \in N, \forall i \in S \tag{6}$$

$$\sum_{a \in A} \sum_{a_p \in a} x_n^{a_p} \cdot \theta_{(n,i)}^{a_p} \cdot ram^{a_p} \leqslant RAM, \quad \forall n \in N, \forall i \in S \tag{7}$$

$$\pi_{(e_1,w)}^{(a_p,a_q)} = \pi_{(e_2,w)}^{(a_p,a_q)}, \quad \forall w \in W, e_1, e_2 \in l^{(a_p,a_q)}, a_p, a_q \in a, a \in A \tag{8}$$

$$\sum_{i \in S} y_{(e,w,i)}^{(a_p,a_q)} = \begin{cases} T^{(a_p,a_q)}, & \pi_{(e,w)}^{(a_p,a_q)} = 1 \\ 0, & \pi_{(e,w)}^{(a_p,a_q)} = 0 \end{cases} \quad \forall e \in l^{(a_p,a_q)}, a_p, a_q \in a, a \in A, w \in W \tag{9}$$

$$\max(i) - \min(i) = T^{(a_p,a_q)} - 1, \quad y_{(e,w,i)}^{(a_p,a_q)} = 1, \forall e \in l^{(a_p,a_q)}, a_p, a_q \in a, a \in A, w \in W, i \in S \tag{10}$$

$$\sum_{e_1,e_2 \in l^{(a_p,a_q)}} (\min(j) - \min(i)) = \lceil D^{l^{(a_p,a_q)}}/c/t \rceil, \quad y_{(e_1,w,i)}^{(a_p,a_q)} = 1, y_{(e_2,w,j)}^{(a_p,a_q)} = 1, e_1 \neq e_2, a_p, a_q \in a, a \in A, w \in W, i, j \in S \tag{11}$$

$$z_{(e,w,i)}^{(b,a_p)} - \theta_{(n,i)}^{a_p} = \begin{cases} 1, & z_{(e,w,i)}^{(b,a_p)} = 1 \\ -1, & \theta_{(n,i)}^{a_p} = 1 \\ 0, & else \end{cases} \quad \forall e \in l^{(b,a_p)}, a_p \in a, a \in A, n \in N, w \in W, i \in S \tag{12}$$

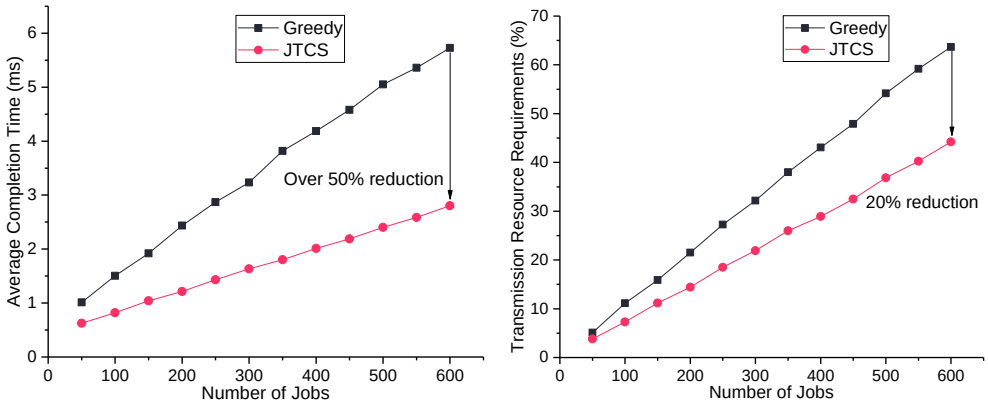
$$\min(j) - \min(i) \geqslant \lceil D^{l^{(b,a_p)}}/c/t \rceil + T^{(b,a_p)}, \quad z_{(e,w,i)}^{(b,a_p)} = 1, \theta_{(n,j)}^{a_p} = 1, e \in l^{(b,aP)}, a_p \in a, a \in A, n \in N, w \in W, i, j \in S \tag{13}$$

$$\theta_{(n,i)}^{a_p} - y_{(e,w,i)}^{(a_p,a_q)} = \begin{cases} 1, & \theta_{(n,i)}^{a_p} = 1 \\ -1, & y_{(e,w,i)}^{(a_p,a_q)} = 1 \\ 0, & else \end{cases} \quad \forall e \in l^{(a_p,a_q)}, a_p, a_q \in a, a \in A, n \in N, w \in W, i \in S \tag{14}$$

$$\min(j) - \min(i) \geqslant \lceil D^{l^{(a_p,a_q)}}/c/t \rceil + C^{a_p}, \quad \theta_{(n,i)}^{a_p} = 1, y_{(e,w,j)}^{(a_p,a_q)} = 1, e \in l^{(a_p,a_q)}, a_p, a_q \in a, a \in A, n \in N, w \in W, i, j \in S \tag{15}$$

$$y_{(e,w,i)}^{(a_p,a_q)} - \theta_{(n,i)}^{a_q} = \begin{cases} 1, & y_{(e,w,i)}^{(a_p,a_q)} = 1 \\ -1, & \theta_{(n,i)}^{a_q} = 1 \\ 0, & else \end{cases} \quad \forall e \in l^{(a_p,a_q)}, a_p, a_q \in a, a \in A, n \in N, w \in W, i \in S \tag{16}$$

$$\min(j) - \min(i) \geqslant \lceil D^{l^{(a_p,a_q)}}/c/t \rceil + T^{(a_p,a_q)}, \quad y_{(e,w,i)}^{(a_p,a_q)} = 1, \theta_{(n,j)}^{a_p} = 1, e \in l^{(a_p,a_q)}, a_p, a_q \in a, a \in A, n \in N, w \in W, i, j \in S \tag{17}$$



(a) (b)

Fig. 2: Simulation results. (a) Average completion time. (b) Transmission resource requirements.

Optimization F tries to minimize the average completion time of all jobs. Constraint (2) demonstrates that two tasks could not be supported by the same node. Constraint (3) indicates that a task is executed by one and only one node. Constraint (4) indicates data transmission is avoided when data storage and task execution are served in the same node. Constraint (5) denotes the computation resources allocation. Constraints (6)-(7) imply the limitation of computation resources. Constraint (8) ensures the wavelength continuity. Constraints (9)-(10) guarantee the time slices continuity and constraint (11) shows the propagation latency of time slices. Constraints (12)-(13) indicate that a task could not be executed before the data transmission. Constraints (14)-(15) ensure computational results transmission starts after the task execution is finished. Constraints (16)-(17) infer that a task will not be executed until all computational results from its forward tasks are transmitted.

3. Performance evaluation

We adopt a 6-node mesh topology depicted in Fig. 1(b) to evaluate the performance. The distance between each node pair is set to 20 *km*. Four wavelengths are implemented in each fiber link with bi-directional transmission. The length of one cycle is set to 10 *ms* and each cycle is divided into 100 time slices, which means the length of a time slice is 100 μs. The CPU and RAM capacities of each node are both set to 100, and the CPU and RAM requirements of each task range from 5 to 50 uniformly. The execution of each task lasts for 1 time slice. The computational results and raw data transmission demand 1 and 3 time slices, respectively. We generate 50 to 600 jobs (the job model is shown in Fig. 1(a)) in our simulation.

Fig. 2(a) and (b) show the average completion time and transmission resource requirements under different number of jobs, respectively. From the figures, we could learn that the average completion time and transmission resource requirements increase with the increment of the number of jobs. More tasks and computational results are waiting for computation and transmission resources, leading to higher completion time. JTCS achieves over 50% completion time reduction compared with the traditional greedy method when 600 jobs are generated. Furthermore, JTCS realizes 20% transmission resource requirements reduction. It is because JTCS aims at minimizing the completion time and tends to assign the task to the node where the required raw data are stored. Under such circumstances, raw data transmission is avoided and transmission resources are saved.

4. Conclusions

Distributed tasks scheduling is a key challenge in edge computing. We propose a collaboration method by joint optimization of transmission, computation, and storage resources allocation. A mathematical model aiming at minimizing the completion time is also formulated in this paper. Over 50% completion time reduction and 20% transmission resource requirements decrement are realized according to the simulation results.

5. Acknowledgement

This work was supported in part by the National Key R&D Program (No.2018YFB1801702), and the NSFC (No.61871448).

6. References

[1] E. Ahmed, et al., "Bringing computation closer toward the user network: Is edge computing the solution?" IEEE Commun. Mag., 55.11 (2017)

[2] J. Li, et al., "Flexible low-latency metro-access converged network architecture based on optical time slice switching." IEEE/OSA J. Opt. Commun. Netw., vol. 11, no. 12, pp. 624-635, December 2019.

[3] J. Li, et al., "A flexible low-latency metro-access converged network approach based on time-synchronized TWDM-PON." Proc. OFC, 2018.

[4] N. Hua, et al., "Optical time slice switching (OTSS): An all-optical sub-wavelength solution based on time synchronization." Proc. ACP, 2013.

[5] W. Guo, et al., "Demonstration of joint resource scheduling in an optical network integrated computing environment [Topics in Optical Communications]." IEEE Commun. Mag., vol. 48, no. 5, pp. 76-83, May 2010.