Q-Learning-Based Task Allocation for High-Speed Distributed Computing in Heterogeneous Clusters

Seung Geon Yang†, Girim Kwon‡ and Seung-Chan Lim†
†School of ICT, Robotics and Mechanical Engineering, Hankyong National University, Anseong, South Korea
‡School of Electrical and Computer Engineering, University of Seoul, Seoul, South Korea
{tmdrjs1543, sclim}@hknu.ac.kr†, gkwon@uos.ac.kr‡

***Abstract*—This paper proposes a Q-learning-based task allocation approach for wireless coded distributed computing systems with heterogeneous worker nodes. Task allocation in such systems is challenging due to the heterogeneity in computation and communication capabilities, leading to non-identically and independently distributed processing times across nodes. By modeling the task allocation problem as a Markov decision process and applying Q-learning, the master node learns to allocate tasks effectively, adapting to node heterogeneity and minimizing the average processing time. This approach highlights the potential of reinforcement learning to optimize distributed computing in heterogeneous environments.**

***Index Terms*—Q-learning, task allocation, wireless distributed computing.**

I. Introduction

The rapid growth of applications like machine learning and big data analytics has heightened demand for distributed computing systems, which must efficiently process large-scale data. However, system performance is often hindered by stragglers (i. e., nodes that experience delays in data processing), creating bottlenecks that increase overall data processing times. To address these issues, coded distributed computing has emerged as a powerful approach, introducing redundancy in task allocation by encoding tasks across multiple nodes [1].

Coded distributed computing was invented to mitigate computation stragglers [2], and the integration with wireless communication enhanced scalability [3]. Despite these advances, communication stragglers remain a persistent challenge in wireless environments because dynamic channels can introduce unpredictable communication delays. Although achievable data rate was incorporated into communication latency and emphasized the importance of jointly optimizing computation and communication [4], the study relies on a communication protocol in a time-division manner, which may be suboptimal to implement high-speed distributed computing frameworks.

To overcome the limitations of previous studies, we herein propose a novel Q-learning-based task allocation strategy to address the challenges of heterogeneous wireless coded distributed computing systems. We focus on both computation and communication stragglers by incorporating a frequency division multiple access communication protocol [5]. Within this framework, we model the task allocation problem as a Markov decision process (MDP) and employ Q-learning to enable the master node to learn effective task allocation policy for reducing the average processing time. Simulation results demonstrate the potential of reinforcement learning to handle system heterogeneity and optimize task allocation, facilitating high-speed distributed computing systems.

The remainder of the paper is organized as follows. In Section II, we introduce the system model for a heterogeneous wireless coded distributed computing framework and analyze the data processing time. Section III presents the proposed Q-learning-based task allocation method, aimed at minimizing average processing time. In Section IV, we evaluate the performance of the proposed method, demonstrating the ability to reduce processing time by adapting to node heterogeneity. Finally, we conclude the paper in Section V.

II. Preliminaries

A. System Model

We consider a distributed computing system based on a master-worker setup in a heterogeneous wireless cluster, with a master node and J worker nodes. For uplink communication, the total available bandwidth W is equally divided into orthogonal subbands, one for each worker node. The wireless channels between the j-th worker node and the master node are modeled as quasi-static fading channels, which remain constant during the completion of a distributed computing task. The channel fading coefficient from the j-th worker node to the master node is represented as h_j.

In the distributed computing setup, we consider a scenario where a matrix-vector multiplication $\mathbf{Ax}$ is computed, where $\mathbf{A}$ is a matrix of size $k \times c$, and $\mathbf{x}$ is a vector of length c. To mitigate the impact of stragglers, the maximum distance separable (MDS) coding is employed to encode $\mathbf{A}$ before distributing its rows among the worker nodes [2]. For $n > k$, a generator matrix $\mathbf{G}$ of size $n \times k$ generates the encoded matrix $\tilde{\mathbf{A}} = \mathbf{GA}$ of size $n \times c$. This encoding enables the system to tolerate up to $(n-k)$ stragglers since the master node can recover the final result using any k out of the n local results. After partitioning $\tilde{\mathbf{A}}$ into J subblocks, each is assigned to one of the J worker nodes. Specifically, the j-th worker node is allocated a local subblock $\tilde{\mathbf{A}}_j$ of size $n_j \times c$, consisting of n_j rows of $\tilde{\mathbf{A}}$.

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2022R1G1A1010641).

The coded distributed computing procedure involves several steps to efficiently perform distributed computation over a wireless environment. First, task allocation is conducted by pre-patching the encoded subblocks $\{\tilde{\mathbf{A}}_j\}$ at each worker node. Then, the master node broadcasts $\mathbf{x}$ to all worker nodes. Upon receiving $\mathbf{x}$, each worker node performs local computation by calculating n_j inner products based on $\tilde{\mathbf{A}}_j$, resulting in a local output $\tilde{\mathbf{x}}_j = \tilde{\mathbf{A}}_j\mathbf{x}$. These local results are then transmitted to the master node via uplink communication. Thanks to the MDS property, the master node can reconstruct the original computation by decoding as soon as it receives the first k inner products.

B. Processing Time

Let ζ_j be the computation time of an worker node j, which is modeled as a two-parameter shifted exponential random variable [6], [7]. The cumulative distribution function (CDF) of ζ_j is given by

$$P\{\zeta_j \leq t\} = 1 - \exp\left(-\tfrac{\mu_j}{n_j}\left(t - a_j n_j\right)\right), \tag{1}$$

for $t \geq a_j n_j$, where μ_j is the straggling parameter and a_j is the shift parameter. The computation time increases with the number of allocated tasks, as reflected by the dependence on n_j. Specifically, larger n_j leads to longer computation times because the worker needs to process more computing tasks. This model captures the heterogeneous nature of the worker nodes, where some nodes may experience straggling due to processing capabilities $\{\mu_j, a_j\}$.

The communication time ξ_j refers to the time required for worker node j to transmit the local computation results $\tilde{\mathbf{x}}_j$ to the master node via uplink, which is given as follows [4]:

$$\xi_j = \tfrac{n_j b}{R_j}. \tag{2}$$

Here, b is the number of bits required to express each local computation result (i.e., an inner product), and R_j is the transmission rate of worker node j, represented as

$$R_j = W_j \log_2\left(1 + \tfrac{|h_j|^2}{W_j \sigma^2}\right), \tag{3}$$

where W_j is the allocated bandwidth for worker j (i.e., $W_j = W/J$). Thus, the communication time ξ_j also depends on the number of allocated tasks n_j, and the channel fading coefficients $\{h_j\}$ reflects the heterogeneous nature of the wireless transmissions.

Based on the wireless coded distributed computing procedure, the processing time T represents the total time required for the master node to collect a sufficient number of local results for successful decoding, which is defined as

$$T = \nu^{\text{th}} \min_{j \in \{1,\ldots,J\}} \left(\zeta_j + \xi_j\right), \tag{4}$$

where ν is the minimum number of worker nodes from which the master node must receive results to enable successful decoding. Specifically, the MDS property guarantees successful decoding if the total number of $\tilde{\mathbf{x}}_j$ received from these ν nodes meets or exceeds k. Therefore, the processing time T is determined by the ν-th smallest sum of ζ_j and ξ_j, ensuring the master node obtains sufficient local computation results for decoding.

III. Q-Learning-based Task Allocation Method

The optimal task allocation strategy is designed to distribute the rows of the encoded matrix $\tilde{\mathbf{A}}$ among the J worker nodes to minimize the average processing time $\mathbb{E}[T]$. By assigning rows of $\tilde{\mathbf{A}}$ to each worker node based on both computation and communication capabilities, the system achieves a balanced load across the nodes, allowing for high-speed distributed computing. In this context, for the task allocation vector $\mathbf{n} = [n_1, n_2, \ldots, n_J]$, the optimal task allocation problem can be formulated as follows:

$$\mathbf{n}^* = \arg\min_{\mathbf{n}} \ \mathbb{E}[T] \tag{5a}$$

$$\text{subject to:} \ \sum_{j=1}^{J} n_j = n. \tag{5b}$$

The heterogeneity of the system makes the task allocation problem highly challenging. Since T is the ν-th order statistic of non-identically and independently distributed (non-i.i.d.) random variables, the objective function is difficult to express in closed form, adding difficulty to the optimization. To address these challenges, we propose a reinforcement learning-based task allocation algorithm. Reinforcement learning provides a promising approach by enabling the master node (i.e., agent) to dynamically learn an optimal allocation policy through interactions with the wireless coded distributed computing system (i.e., environment). This approach effectively handles system heterogeneity, thereby reducing the average processing time. To this end, we formulate the task allocation problem as a Markov decision process (MDP) as follows:

- **State (s_k):** At time step k, the state is defined by the current task allocation vector across the J worker nodes, i.e., $s_k = (n_1, n_2, \ldots, n_J)$, where n_j represents the number of tasks assigned to the j-th worker node.
- **Action (a_k):** The agent modifies the task allocation by increasing the task count for one worker node and decreasing it for another, ensuring that the total number of tasks remains constant.
- **Reward (r_k):** At time step k, the reward is defined as $r_k = \exp(-T_k)$, where T_k is the processing time observed after taking a_k. This reward design encourages minimizing processing time because a larger reward is given for shorter processing times while lower rewards are given if the processing time increases.

To learn the optimal policy $\pi^*(s_k)$ that minimizes the processing time, we apply the Q-learning algorithm. Q-learning maintains a Q-value function $Q(s_k, a_k)$, which estimates the expected cumulative reward for taking action a_k in state s_k. At each time step, the master node observes the current state, selects an action using an exploration-exploitation strategy, receives a reward based on the processing time, observes the next state, and updates the Q-value function. The Q-value update is computed as follows:

$$Q(s_k, a_k) \leftarrow Q(s_k, a_k) + \alpha \left[r_k + \gamma \max_{a_{k+1}} Q(s_{k+1}, a_{k+1}) - Q(s_k, a_k) \right], \tag{6}$$

where α is the learning rate and γ is the discount factor. This process repeats until a set number of episodes is reached, allowing the agent to iteratively improve the task allocation policy.

IV. Simulation Results

To validate the effectiveness of the proposed Q-learning-based task allocation method, we evaluate the average processing time in a wireless coded distributed computing system with one master node and four heterogeneous worker nodes. The computation parameters $\{\mu_j, a_j\}$ for each worker node are uniformly distributed between 0.5 and 10, and the wireless channels are modeled as static Rayleigh fading channels with a 20 dB SNR, introducing heterogeneity in both computation and communication times. Applying the MDC coding, four computing tasks are encoded into 8 tasks, with each local result represented in 16 bits, and the total 1 MHz bandwidth is equally allocated across the worker nodes.

The proposed Q-learning-based task allocation method was trained over 1000 episodes, each consisting of 200 timesteps, with a learning rate of 0.1 and a discount factor of 0.9. To enhance exploration, an epsilon-decay strategy was implemented, gradually reducing the epsilon value to balance exploration and exploitation as training progressed. These parameters allowed the master node to efficiently learn the optimal task allocation policy by interacting with the environment and progressively minimizing average processing time.

Fig. 1 compares the average processing time of the proposed Q-learning-based task allocation method with that of the uniform task allocation strategy, which evenly distributes tasks across all worker nodes. While the uniform allocation shows significant fluctuations and processing times exceeding 2.3 seconds, the proposed method converges within 250 episodes, stabilizing at approximately 1.9 seconds and consistently achieving lower and more stable performance. This result demonstrates the effectiveness of the proposed approach in dynamically adjusting task allocation, effectively handling the heterogeneity and significantly reducing processing times.

V. Conclusion

In this work, we addressed the challenge of efficient task allocation in wireless coded distributed computing systems by introducing a Q-learning-based strategy. The heterogeneity in computation and communication capabilities results in non-i.i.d. processing times across worker nodes, complicating the optimization of task allocation. By formulating the problem as an MDP and applying Q-learning, the master node learns a task allocation policy that minimizes average processing time. The proposed Q-learning-based method significantly outperforms the uniform task allocation strategy, achieving a reduction in processing time by dynamically handling node heterogeneity and minimizing delays. These results reveal the potential of reinforcement learning in optimizing task allocation to enable high-speed wireless distributed computing systems.

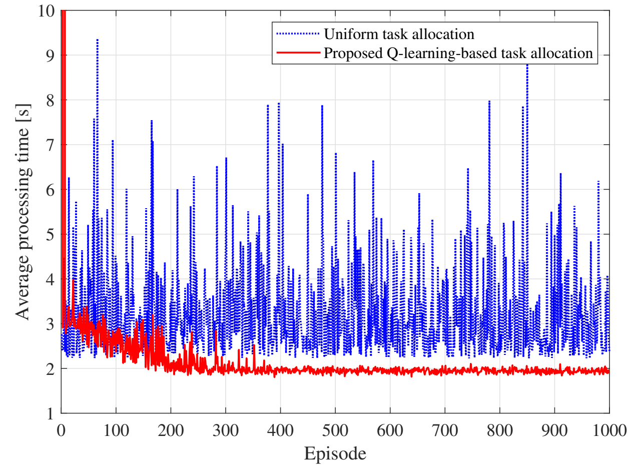

Fig. 1. Comparison of average processing time between the proposed Q-learning-based task allocation method and the uniform task allocation strategy over 1000 episodes.

References

[1] J. S. Ng, W. Y. B. Lim, N. C. Luong, Z. Xiong, A. Asheralieva, D. Niyato, C. Leung, and C. Miao, "A comprehensive survey on coded distributed computing: Fundamentals, challenges, and networking applications," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 3, pp. 1800–1837, 2021.

[2] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inform. Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.

[3] S. Li, Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "A scalable framework for wireless distributed computing," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2643–2654, 2017.

[4] F. Wu and L. Chen, "Latency optimization for coded computation straggled by wireless transmission," *IEEE Wireless Commun. Lett.*, vol. 9, no. 3, pp. 388–391, 2020.

[5] D.-J. Han, J.-Y. Sohn, and J. Moon, "Coded wireless distributed computing with packet losses and retransmissions," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5295–5310, 2021.

[6] G. Liang and U. C. Kozat, "TOFEC: Achieving optimal throughput-delay trade-off of cloud storage using erasure codes," in *IEEE INFOCOM*, 2014.

[7] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, "Coded computation over heterogeneous clusters," *IEEE Trans. Inform. Theory*, 2019.