(Com)^{2}Net: A Novel Communication and Computation Integrated Network Architecture

Weiting Zhang, *Member, IEEE*, Dong Yang, *Member, IEEE*, Chuan Zhang, *Member, IEEE*, Qiang Ye, *Senior Member, IEEE*, Hongke Zhang, *Fellow, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—With the wide deployment of computing capabilities and artificial intelligence (AI) techniques, Internet of intelligence is envisioned as a natural tendency in future networks. This technological revolution will foster abundant computing services with new requirements, such as distributed large AI model training and on-the-fly image rendering, which require dynamic collaboration of multi-dimensional resources from both the communication and computation perspectives. In this article, we introduce a communication and computation integrated network architecture, named (Com)^{2}Net. Enormous computing traffic can be scheduled across space-air-ground domain, end-edge-cloud domain, and multi-data center domain, which facilitates ubiquitous connectivity and collaborative computation, thus supporting diverse advanced computing services. Furthermore, an intelligent resource adaption scheme is proposed to dynamically orchestrate multi-dimensional resources for massive concurrent computing tasks, in which quantum genetic algorithms are utilized to make the best joint decision for inter-domain traffic scheduling and intra-domain resource allocation. Finally, we present a case study, and discuss open research issues that are fundamental for efficient collaborative computation in (Com)^{2}Net.**

I. Introduction

With the advent of the Intelligence of everything (IoE) era, a wide variety of new network scenarios are emerging, such as Metaverse, ultrahigh-speed railway (USR), and intelligent industrial Internet-of-Things (IoT) [1]. Numerous end terminals are connected to the Internet, and tremendous data is generated at the network edge, which needs to be transmitted and processed efficiently [2]. As reported by Internet Data Center, the annual data generated worldwide will grow to 175 Zettabytes by 2025 [3]. Leveraging the largely increased data volume, various advanced services with different quality of service (QoS) and quality of experience (QoE) requirements can be provided to improve operation and production efficiency. For instance, in USR scenarios, with the captured audio and video data, the operation safety of USR vehicles can be guaranteed effectively via artificial intelligence (AI) algorithms. In industrial IoT, with the collected multi-sensory data, the potential failures of robotic arms can be detected instantly via mathematical optimization methods. These services have common features, namely, computation-intensive and time-sensitive. To this end, how to facilitate efficient computation is especially critical to support abundant advanced services in IoE scenarios.

Recently, numerous computing paradigms are proposed to achieve this goal. Fueled by powerful computing capabilities, cloud computing significantly promotes the development of AI techniques. Yet, long-distance data transmission suffers from a prohibitive response latency. To provide low-latency computing services, multi-access edge computing (MEC) arises as the computing resources sink from the cloud to the edge, such as base stations, industrial gateways, or road side units. Moreover, taking into account data privacy laws and regulations, distributed computing has attracted considerable attention from both academia and industry since the dispersed computing resources can be collaboratively utilized, e.g., federated learning (FL) [4]. However, when the computing traffic tends to be diversified, customized, and intelligentize, single computing paradigm is difficult to satisfy the increasingly complex QoS and QoE requirements. In such case, it is of paramount importance to converge multiple computing paradigms into a unified computing platform, where massive concurrent computing traffic can be globally orchestrated in a more efficient manner and the advanced service requirements can be guaranteed effectively.

Scheduling massive concurrent computing traffic over a unified computing platform encounters many challenges. *First*, computing task processing requires dynamic collaboration of multi-dimensional resources from both the communication and computation perspectives. Moreover, the unified computing platform needs to manage massive concurrent computing tasks generated from a wide variety of advanced services, such as large AI model training and image rendering, which further complicates computing traffic scheduling. Hence, it is necessary to design a novel architecture to support integrated management for communication and computing resources, thus facilitating efficient ubiquitous computation. *Second*, diverse computing services are fostering numerous new QoS and QoE requirements, such as deterministic latency and jitter, training accuracy, inference time and energy consumption, and data quality and privacy preserving [5]. Hence, it is essential to develop an intelligent resource adaption algorithm to dynamically allocate multi-dimensional resources over the unified computing platform. To accelerate computation efficiency, how

Weiting Zhang, Dong Yang, and Hongke Zhang are with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China. E-mail: {wtzhang, dyang, hkzhang}@bjtu.edu.cn.

Chuan Zhang is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China. E-mail: chuanz@bit.edu.cn.

Qiang Ye is with the Department of Electrical and Software Engineering, Schulich School of Engineering, University of Calgary, 2500 University Drive NW, Calgary, AB T2N 1N4. Email: qiang.ye@ucalgary.ca.

Xuemin (Sherman) Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo N2L 0B5, Canada. E-mail: sshen@uwaterloo.ca.

Corresponding author: Dong Yang (dyang@bjtu.edu.cn).

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3355922

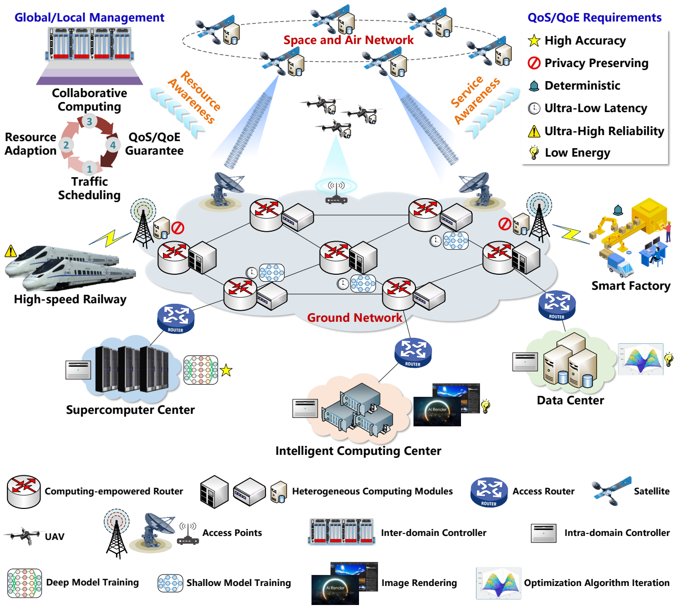


Fig. 1. An illustration of (Com)^{2}Net.

can we schedule massive concurrent computing traffic while satisfying the differentiated and stringent requirements needs further investigation.

In this article, we present a communication and computation integrated network architecture for massive concurrent computing traffic, named (Com)^{2}Net. Enormous computing traffic can be scheduled across space-air-ground domain, end-edge-cloud domain, and multi-data center domain. Then, we propose a quantum genetic algorithm (QGA)-assisted scheme to support inter-domain traffic scheduling and intra-domain resource adaption, thereby facilitating efficient multi-dimensional resource management. A detailed procedure of signaling interaction among end terminals, an inter-domain controller, intra-domain controllers, and computing platforms is presented. Finally, potential research directions such as computing-oriented routing protocol and blockchain-enabled computing transaction are introduced.

The remainder of this article is organized as follows. First, a communication and computation integrated network architecture is presented in Section II. Then, a quantum-assisted intelligent resource adaption scheme is proposed in Section III, followed by a case study in Section IV. Finally, we discuss some open research issues and draw concluding remarks in Section V.

II. (Com)^{2}Net: Communication and Computation Integrated Network Architecture

A. System Architecture

As previously mentioned, how to satisfy diverse QoS and QoE requirements for advanced computing services remains a challenging issue [6]. Therefore, a multi-dimensional integrated system architecture for (Com)^{2}Net is introduced to efficiently support traffic scheduling and resource adaption for massive concurrent computing tasks. The architecture, as depicted in Fig. 1, aims to integrate ubiquitous connectivity and collaborative computation, while supporting a variety of computing services with distinct requirements.

Compared to conventional forwarding-dedicated networks, the (Com)^{2}Net is endowed with several brand-new features. For one thing, all network components are equipped with a computing unit to process computing tasks. Here, the network components refer to end devices, MEC servers, cloud servers, router devices, unmanned aerial vehicle (UAV), and satellites. Moreover, multiple computing centers, including data center, intelligent computing center, and supercomputer center, are embraced into the introduced architecture [7]. As such, enormous computing traffic can be scheduled across space-air-ground domain, end-edge-cloud domain, and multi-center domain, which facilitates ubiquitous connectivity and collaborative computation thus supporting diverse advanced computing services. For another thing, a two-level control mechanism is considered to jointly perform global and local management, respectively. In specific, to process the computing tasks, two optimal decisions should be dynamically made, i.e., *traffic scheduling* and *resource adaption*. First, the traffic scheduling decision is to reasonably schedule computing tasks across the multiple domains. Second, the resource adaption decision is to dynamically allocate multi-dimensional resources for transmitting and processing the corresponding computing tasks.

As shown in Fig. 2, the functional architecture of (Com)^{2}Net consists of three layers: *computing and networking component layer*, *computing service layer*, and *resource convergence adaption layer*. In the following, we illustrate the detailed descriptions of each layer in the (Com)^{2}Net functional architecture.

B. Computing and Networking Component Layer

This layer plays a fundamental role for the entire (Com)^{2}Net architecture. Three types of components are considered to provide computing and networking services for computing tasks, i.e., network devices, computing devices, and integrated devices. Specifically, the network devices, such as routers, switches, and gateways, construct the core networks and are mainly responsible for hop-by-hop task forwarding. The computing devices consist of one or more processing modules such as central processing unit (CPU), graphics processing unit (GPU), and field programmable gate array (FPGA), and are in charge of instant task processing [8]. Compared to the network devices, the computing devices are generally distributed at the network edge, including multiple computing centers. That is to say, the computing tasks can be transmitted to computing devices by those network devices. In addition, the integrated devices converge the above two types of devices into one, thus providing forwarding and processing services for the computing tasks. For example, through introducing computing capabilities, the router devices can possess forwarding and computing functions, simultaneously. When a computing task is traversed through a computing-enhanced router device, it can either process the task locally or forward the task to other devices.

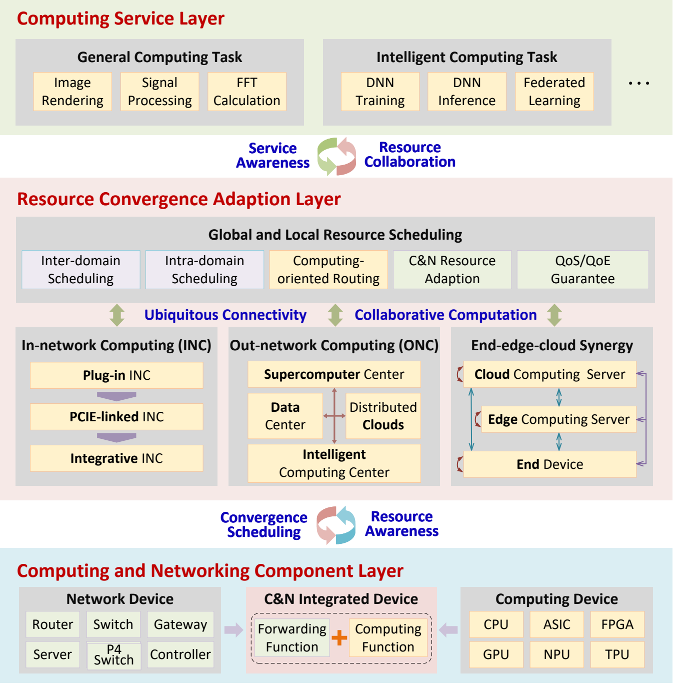


Fig. 2. (Com)^{2}Net Functional Architecture.

C. Computing Service Layer

In the computing service layer, the attribute features of various computing services can be abstracted according to their QoS and QoE requirements, and useful information can be obtained, including whether the task needs to be processed on multiple computing devices and how many computing devices need to be scheduled to complete the task. Such key information can provide service-level support for computing and networking convergence scheduling [9]. Furthermore, the computing tasks can be classified into different types, such as general computing tasks and intelligent computing tasks, which is helpful to conduct global task management. Typically, the computation process for general computing tasks cannot be split and needs to be carried out on a separate computing device. Taking fast fourier transform (FFT) as an example, the calculation can be performed on an MEC server that has installed the required mathematical tool. On the contrary, the computation process for intelligent computing tasks can be divided and coordinated among several computing devices. For instance, federated training requires deep neural network (DNN) model frequently interacting among a centralized server and dispersed end devices. During the interaction process, multi-dimensional resources including networking, computing, and storage should be flexibly allocated to support efficient federated training.

D. Resource Convergence Adaption Layer

The goal of the resource convergence adaption layer is to efficiently manage and schedule the multi-dimensional resources. Three computing paradigms can be provided to support collaborative computation services, that is, in-network computing (INC), out-network computing (ONC), and end-edge-cloud synergy (EECS) computing [10]. For lightweight computing tasks, the INC paradigm can provide on-the-road computing services via the integrated devices deployed along the task transmission path. Due to the powerful computing capabilities, the ONC can support heavyweight computing tasks by scheduling them to the geographically dispersed data centers, intelligent computing centers, and supercomputer centers. In addition, a entire task can also be divided into several portions and completed collaboratively. The EECS is capable of orchestrating different portions among the end, edge, and cloud. For example, to improve DNN inference efficiency, an inference task can be split into two blocks, where the shallow layers' inference operation is performed on the end devices and the remaining layers' inference can be offloaded to the edge or cloud. In a word, the key for the above computing paradigms is to build relationship for the service, cluster, and component.

To this end, the dynamic mapping mechanisms of "service-to-cluster" and "cluster-to-component" are developed. By precisely perceiving differentiated task requirements and multi-dimensional resource attributes, functional modules including inter-domain scheduling, intra-domain scheduling, computing-oriented routing, computing and networking convergence adaptation, and QoS and QoE guarantee are built to realize dynamic mapping between services and components. Specifically, the "service-to-cluster" mapping mechanism is responsible for detecting and translating the requirements of massive concurrent computing tasks, and establishing a service cluster that facilitates cooperative processing for each computing task at the logical level. The "cluster-to-component" mapping mechanism is in charge of bringing the established service cluster into the physical computing and networking space and building the component cluster that performs the actual computing operations.

III. Quantum-Enhanced Intelligent Resource Adaption for (Com)^{2}Net

In this section, we present a QGA-assisted resource adaption algorithm and then investigate inter-domain and intra-domain scheduling schemes. In addition, we discuss the detailed procedure of signaling interaction in (Com)^{2}Net.

A. QGA-assisted Resource Adaption Algorithm

As illustrated in Fig. 1, two types of controllers are deployed in the (Com)^{2}Net architecture. One is the inter-domain controller, which is in charge of performing traffic scheduling across multiple domains from the global perspective. The other is the intra-domain controller, which is to allocate resources for computing tasks scheduled to the corresponding domains. To make the best decisions of traffic scheduling and resource allocation, each controller is endowed with a QGA-assisted resource adaption algorithm [11]. To approach the optimal solution, the algorithm requires multiple iterations, including chromosome construction, initialization, selection, crossover, and mutation, and the iteration process is as follows [12].

1) Quantum Chromosome Construction and Population Initialization: In the QGA-assisted algorithm, quantum chromosome i is utilized to represent individuals in the population,

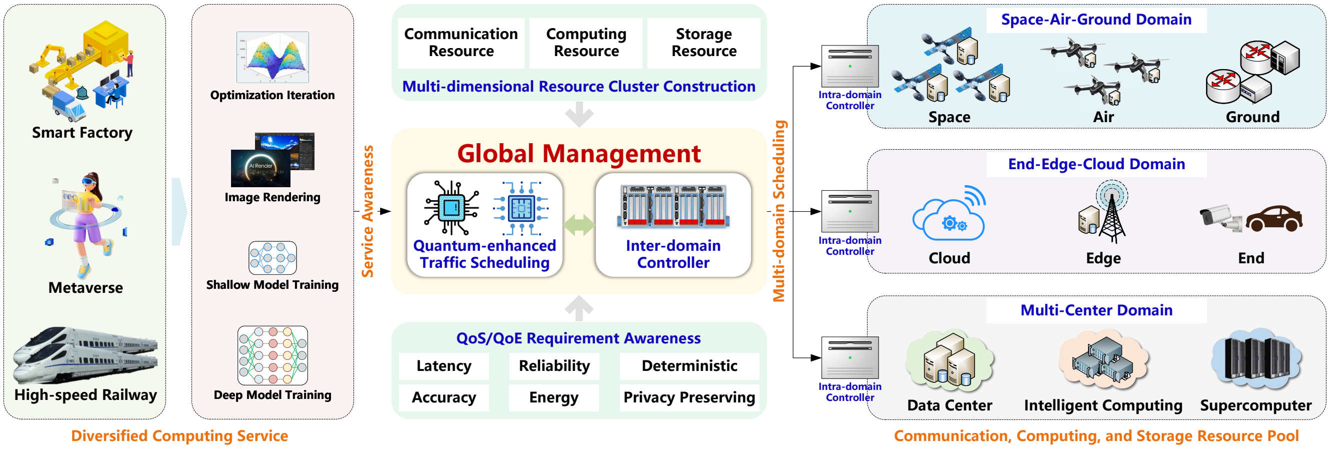


Fig. 3. Illustrate of inter-domain global scheduling in (Com)^{2}Net.

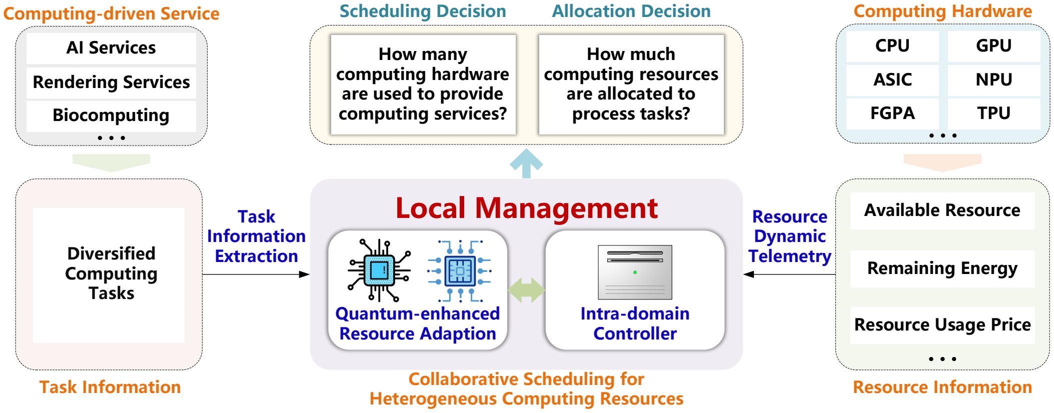

Computing-driven Service
AI Services
Rendering Services
Biocomputing
• • •
Scheduling Decision
Allocation Decision
How many computing hardware are used to provide computing services?
How much computing resources are allocated to process tasks?
Computing Hardware
CPU
GPU
ASIC
NPU
FGPA
TPU
• • •
Diversified Computing Tasks
Task Information Extraction
Local Management
Quantum-enhanced Resource Adaption
Intra-domain Controller
Resource Dynamic Telemetry
Available Resource
Remaining Energy
Resource Usage Price
• • •
Task Information
Collaborative Scheduling for Heterogeneous Computing Resources
Resource Information

Fig. 4. Illustrate of intra-domain local scheduling in (Com)^{2}Net.

i.e., $C_i = \{c_i^1, c_i^2, \cdots, c_i^D\}$. Here, D is the length of chromosomes, and $c_i^k \in [1, m], 1 \leq k \leq D$ denotes the traffic scheduling and resource allocation decisions (i.e., s_k and a_k) for k-th computing task. Note that the alternative scheduling decisions have m cases. The chromosome is constructed using quantum bit (i.e., qubit) coding methods that differ from traditional genetic algorithms in the binary coding and integer coding methods. The qubit can exist in a superposition of states, thus enabling the chromosome simultaneously represent multiple solutions. Then, a set of population $P(0) = \{C_1^0, C_2^0, \cdots, C_p^0\}$ is initialized which corresponds to the potential solution for the joint traffic scheduling and resource allocation optimization problem, where p is the population size. In addition, a Hadamard's gate is utilized to carry out the transformation for each gene of the chromosomes, such that the chromosomes can be represented by a uniform superposition state.

2) Fitness Function Definition: Each individual of the population is measured by calculating the corresponding value of fitness function Fit_i, which is used as the index for selecting the optimal individual. As such, individuals with low fitness will be excluded. If $M{>}f(s_k, a_k)$, $Fit_i = M - f(s_k, a_k)$; otherwise 0. Here, M is a constant value that ensures the fitness is non-negative, and $f(s_k, a_k)$ is the objective function of the algorithm. In addition, livability is utilized to evaluate the solution quality for genetic iteration process, i.e., $livability_i = Fit_i / \sum_{i=1}^{k} Fit_i$. Obviously, larger livability means a larger fitness value, namely a better individual. Furthermore, individual with a larger livability has more opportunities to regenerate in the next generation, otherwise is the opposite. Thus, it is important to formulate an appropriate fitness function that guarantees the feasibility of the candidate individuals.

3) Quantum Chromosome Selection, Crossover, and Mutation: Taking the optimal individual of the current iteration as the evolutionary goal, the next generation population can be formed through a quantum gate operation that modifies the chromosome's qubit coding. Specifically, three processes are performed to approach the optimal solution: In quantum chromosome selection process, the tournament approach is applied to select parent groups for the next generation. First, $p/2$ individuals are randomly selected from the population, and then two individuals with the highest survivability are selected as parent group members. The process is repeated until the number of parental members reaches p. In quantum chromosome crossover process, quantum crossover operators are utilized to combine the quantum chromosomes of parent individuals, which enables the algorithm to explore a larger solution space by utilizing quantum principles (e.g., entanglement and superposition). In quantum chromosome mutation process, quantum gates (e.g., Pauli-X gate) are used to avoid local optima by inverting the amplitudes of quantum chromosomes. With the above processes, the next generation population of quantum chromosomes can be formed accordingly.

4) Optimal Solution Search and Iteration: Repeating the above steps until reaching the iteration termination condition, and then the optimal solution for traffic scheduling and resource allocation can be obtained.

The difference between the traditional genetic algorithm and the QGA-assisted algorithm is mainly in population coding and evolutionary strategies. The fundamental idea behind the QGA's population coding method is to encode chromosomes using qubits and quantum superposition states, which allows each chromosome to simultaneously represent information from multiple states. In addition, the population is updated using a quantum rotating gate, and the evolution is directed by the knowledge of the optimal individual in the current iteration. During the iteration process, each qubit's superposition state gradually tends to a deterministic state and reaches convergence, and thus accomplishing the optimization goal. Due to the unique coding and updating techniques, the QGA-assisted algorithm outperforms classical genetic algorithms in terms of population diversity, convergence speed, and convergence accuracy.

B. Inter-domain Global Traffic Scheduling

Deploying the QGA-assisted algorithm in the inter-domain controller, the traffic scheduling decision can be centrally made to support massive concurrent computing task scheduling among multiple computing domains, such as space-air-ground domain, cloud-edge-end domain, and multi-center domain. Notably, two or more domains can be simultaneously selected to provide computing services for the tasks that have extremely strict QoS or QoE requirements. Moreover, two additional strategies can be generated after the traffic scheduling decision is determined. On the one hand, the QGA-assisted algorithm will generate the computing-oriented routing strategies for each task with differentiated computing resource demands. Implementing the routing strategies to the corresponding network component clusters, the massive tasks can be forwarded to their destination computing node. On the other hand, reasonable resource allocation strategies for network bandwidth will be generated to facilitate efficient task transmission. With these decisions and strategies, the established system can achieve efficient collaborative computing.

C. Intra-domain Local Resource Allocation

With the QGA-assisted algorithm, the intra-domain controller can locally make resource allocation decisions for the computing tasks scheduled to the corresponding domain. Firstly, the resource information of the computing domain is dynamically telemetered, including the available computing and storage resources, remaining energy provision, and resource usage price. Secondly, heterogeneous computing and storage resources are collaboratively scheduled to support task processing. Two important decisions should be dynamically made, i.e., how many computing hardware units are selected to provide computing services? and how much computing resources for each hardware unit are allocated to process the current task? After sufficient iterations, the QGA-assisted algorithm is capable of extracting the dynamic task information and generating accurate resource allocation decisions in real time. As such, the QoS and QoE requirements of the computing task can be effectively guaranteed.

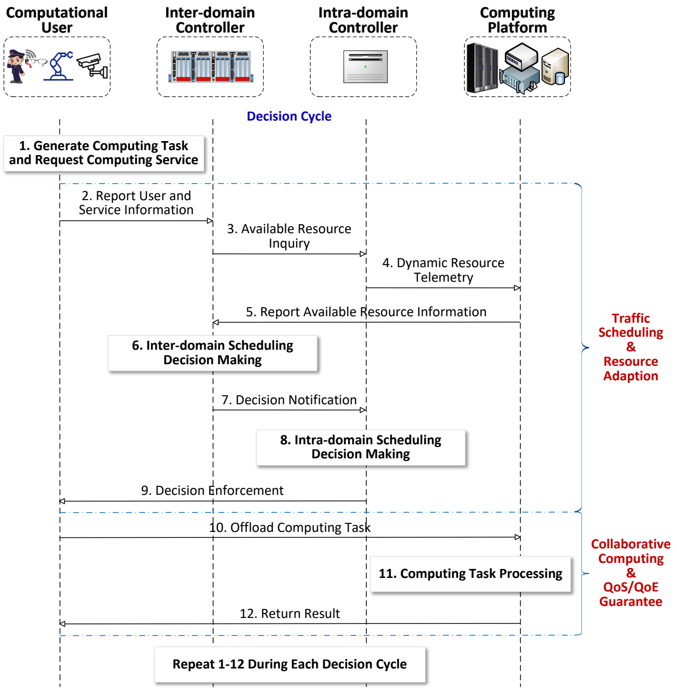

Computational User
Inter-domain Controller
Intra-domain Controller
Computing Platform
Decision Cycle
1. Generate Computing Task and Request Computing Service
2. Report User and Service Information
3. Available Resource Inquiry
4. Dynamic Resource Telemetry
5. Report Available Resource Information
6. Inter-domain Scheduling Decision Making
7. Decision Notification
8. Intra-domain Scheduling Decision Making
9. Decision Enforcement
10. Offload Computing Task
11. Computing Task Processing
12. Return Result
Traffic Scheduling & Resource Adaption
Collaborative Computing & QoS/QoE Guarantee
Repeat 1-12 During Each Decision Cycle

Fig. 5. Procedure of signaling interaction in (Com)^{2}Net.

D. Procedure of Signaling Interaction

The QGA-assisted algorithm can perform inter-domain traffic scheduling and intra-domain resource allocation by offline optimization and online decision. The procedure of the QGA-assisted algorithm in (Com)^{2}Net involves the signaling interaction among computational users, an inter-domain controller, intra-domain controllers, and computing platforms. As illustrated in Fig. 5, the detailed procedure is as the following steps:

1) Computational users generate diverse computing tasks with differentiated requirements, and request moderate computing resources to complete the tasks.
2) The user information and service information are reported to the inter-domain controller to perform global management, such as traffic scheduling.
3) The inter-domain controller inquiries available resource information from the selected intra-domain controllers.
4) The intra-domain controllers collect the required available resource information from heterogeneous computing platforms via dynamic resource telemetry, including the available computing and storage resources, remaining energy provision, and resource usage price.

5) Reporting the required available resource information to the intra-domain controller, and further to the inter-domain controller.
6) The inter-domain controller runs the quantum-assisted traffic scheduling algorithms to make decisions with the obtained real-time information.
7) Deploying the determined traffic scheduling decisions (e.g., distributing computing tasks to corresponding domains) to the intra-domain controller.
8) The intra-domain controllers run the quantum-assisted resource adaption algorithms to allocate computing and storage resources for the corresponding computing tasks.
9) Computational users enforce the received traffic scheduling and resource allocation decisions.
10) Computing service requests are supported using the allocated networking and computing resources. For instance, DNN inference tasks can be offloaded to MEC servers using spectrum resources.
11) Computing tasks are processed using the allocated computing resources. For example, federated learning tasks can be accelerated using GPU and FPGA resources.
12) Computing results are sent back to the initiated computational users for further utilization. For instance, an inference accuracy value of fault diagnostic applications can be sent back to an industrial controller, which can be utilized to determine whether to take maintenance measures for industrial equipment.

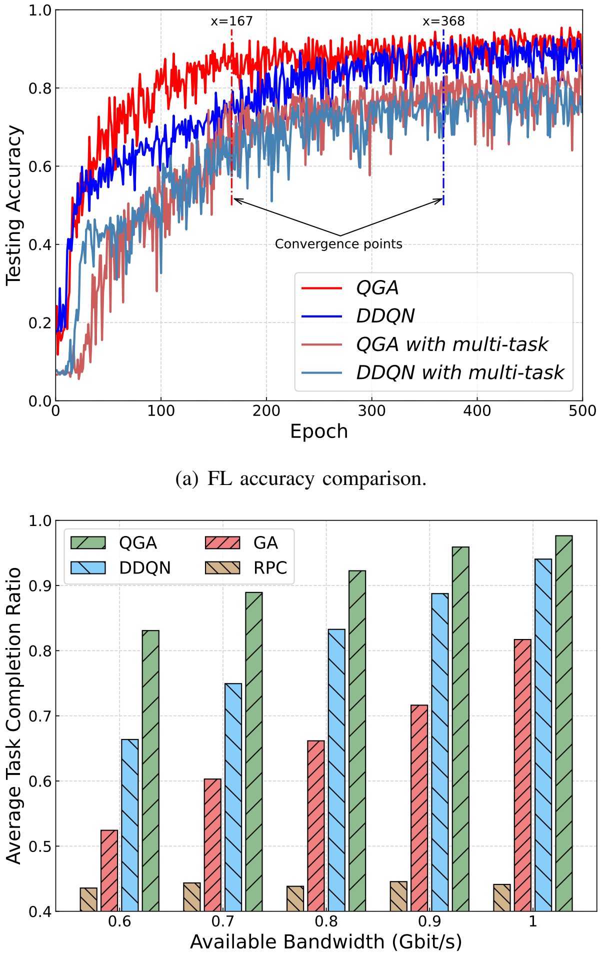
During the learning stage, steps 1–12 are repeated until the QGA-assisted algorithm reaches convergence. For each online decision cycle, steps 1–12 are disposable and have a quick response time [12]. Specifically, in offline optimization stage, the algorithm requires multiple iterations to approach the optimal solution, including chromosome construction, initialization, selection, crossover, and mutation, which is generally time consuming. In online decision stage, the obtained scheduling decisions can be deployed in the controllers to support the inter-domain traffic scheduling and intra-domain resource allocation which can be performed with a low latency [11].

IV. Case Study

In this section, simulations are carried out to evaluate the proposed QGA-based resource adaption algorithm for optimizing hierarchical FL in the (Com)^{2}Net.

A. Considered Scenario

We consider a multi-center integrated network for providing federated training services to intelligent industrial applications. Specifically, three data centers, two intelligent centers, and one supercomputer center are geographically distributed at different cities which construct a federated training-oriented (Com)^{2}Net. Each center involves a set of computing devices, e.g., CPU, GPU, and FPGA, which are connected into the (Com)^{2}Net via an access router. When a smart factory connects the (Com)^{2}Net, enormous computing tasks have to be processed. We consider an accuracy-sensitive fault detection service, whose accuracy requirement is 90% for production safety. To facilitate efficient federated training, we adopt a three-layer learning architecture, in which a self-attention convolutional-based fault detection model is distributively trained on the centers' computing devices, and the updated model parameters are aggregated at the corresponding access routers every epoch or at a centralized factory server every a few epochs. In addition, we utilize a vibration signal-based dataset (i.e., *AWE dataset*[1]) to support the federated training for the model.



(a) FL accuracy comparison.

(b) Average task completion ratio comparison.

Fig. 6. Performance comparison between the proposed QGA-assisted scheme and DDQN benchmark.

To accelerate FL convergence, we formulate the federated training as a joint bandwidth and computing resource optimization problem. Aiming at optimizing task completion ratio and testing accuracy, we utilize the proposed QGA-assisted algorithm to learn the resource adaption policy. For performance comparison, we adopt a policy-gradient algorithm, named *Double Deep Q-Network (DDQN)*, in which an online network and a target network are used to make resource allocation decisions. In this scheme, both online and target networks are fully-connected neural networks with two hidden layers, and the neuron numbers of the hidden layers are $[|S_{dim}|$, 1000, 500, $|A_{dim}|]$. In the simulation, the dimensions of network

[1] Online available. https://github.com/Intelligent-AWE/DeepHealth.

state $|S_{dim}|$ and resource allocation decision $|A_{dim}|$ are 688 and 4, respectively. In addition, the learning rate of this scheme is set to 1×10^{-4}. In addition, the genetic algorithm (GA) and random probabilistic configuration (RPC) are adopted as benchmark schemes.

B. Simulation Results

We evaluate the performance of the proposed QGA-assisted resource adaption scheme based on real-world AWE dateset. Fig. 6(a) shows the convergence performance of the proposed scheme with respect to FL epochs. Three important observations can be obtained from the simulation results. First, when the available bandwidth is the same, the proposed scheme operated in a single-task scenario can obtain a higher accuracy than that in a multi-task scenario within 500 test epochs. Second, the proposed scheme can reduce the number of training epochs until FL convergence by 54.62% and 12.5% in both scenarios compared with the DDQN scheme. The reason is that the quantum-based scheme can allocate network resources for FL traffic more reasonably via utilizing quantum superposition states, which allows each chromosome to simultaneously represent information from multiple states and thus obtaining the optimal resource allocation decision. Second, the proposed scheme achieves higher testing accuracy at the end of the FL training process in both single-task and multi-task scenarios. This indicates that the QGA-assisted scheme can support efficient transmission services for FL parameter aggregation and distribution within the (Com)^{2}Net.

As shown in Fig. 6(b), we evaluate the average task completion ratio with respect to different available bandwidth resources. It can be seen that the proposed QGA-assisted scheme is capable of improving average task completion ratio compared with the benchmark schemes, which indicates that networking and computing resource adaption are optimized. In addition, with the increase of the available bandwidth resources, the average task completion ratio achieved by the QGA-assisted scheme, DDQN, and GA increases. Particularly, when the amount of available bandwidth is 1 $Gbit/s$, the proposed scheme improves the average task completion ratio by 3.7%, 19.5%, and 121.6% compared with the DDQN, GA, and RPC benchmarks. The underlying reason is that more available bandwidth resources can be utilized to transmit FL model parameters over the (Com)^{2}Net, and the quantum superposition characteristic can enhance the decision-making capability of the proposed scheme for resource adaption and thus improving the task completion ratio.

V. Open Research Issues for (Com)^{2}Net

In this section, we present several open research directions for (Com)^{2}Net.

A. Computing-oriented Routing Protocol

The essential of traffic scheduling in (Com)^{2}Net is to forward a computing task to the most appropriate device and process it in an efficient way. This requires comprehensively take network states and available computing resource information into consideration, such that the optimal forwarding decision can be made to support traffic scheduling. Moreover, the availability of dispersed computing resources may dynamically change as the network topology reconstruction, which further complicates the forwarding decision making [13]. How to design computing-oriented routing protocols is an urgent issue to achieve ubiquitous connectivity and collaborative computation in (Com)^{2}Net.

B. Deterministic Scheduling for Massive Computing Traffic

New services, such as autonomous driving and metaverse, not only require ready-to-use computing resources, but also require deterministic guarantee of delay, jitter, and packet loss to satisfy their strict requirements [14]. For instance, image rendering computation of VR-based applications needs to be performed on the remote cloud, and users will suffer from vertigo and a significantly reduced QoE if the delay is too large. To effectively support advanced computing services, developing a deterministic scheduling mechanism for massive computing traffic is an important research issue.

C. Blockchain-enabled Computing Resource Transaction

With the development of blockchain technologies, dispersive and heterogeneous computing resources can be securely trade on a public platform [15]. Users can publish their requests for computing resources on the platform, and suppliers can sell or rent their available resources, thus realizing the schedulable, exchangeable, and on-demand resource supply. However, when misleading information is published, it will result in an imbalance between supply and demand, which is not conducive to efficient platform management. How to construct a trustworthy and transparent computing transaction platform is an interesting topic.

VI. Conclusion

In this article, we have presented the communication and computation integrated network architecture, i.e., (Com)^{2}Net, for the efficient management of diverse advanced computing services and multi-dimensional resources. Furthermore, we have proposed a QGA-assisted resource adaption scheme to facilitate inter-domain traffic scheduling and intra-domain resource allocation. By integrating space-air-ground domain, end-edge-cloud domain, and multi-data center domain, the (Com)^{2}Net can efficiently support collaborative computation for massive concurrent computing tasks. For the future work, we will study a quantum reinforcement learning based intent awareness scheme to enable a behavior-sensitive (Com)^{2}Net for metaverse-oriented computing service support.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 62201029 and 62394321, in part by the China Postdoctoral Science Foundation under Grants 2022M710007 and BX20220029, and in part by the National Key Research and Development Program of China under Grant 2022YFB2901302.

References

[1] D. Yang, W. Zhang, Q. Ye, C. Zhang, N. Zhang, C. Huang, H. Zhang, and X. Shen, "DetFed: Dynamic resource scheduling for deterministic federated learning over time-sensitive networks," *IEEE Transactions on Mobile Computing*, Early access, 2023, DOI: 10.1109/TMC.2023.3303017.

[2] Q. Tang, F. R. Yu, R. Xie, A. Boukerche, T. Huang, and Y. Liu, "Internet of intelligence: A survey on the enabling technologies, applications, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1394–1434, May 2022.

[3] S. Duan, D. Wang, J. Ren, F. Lyu, Y. Zhang, H. Wu, and X. Shen, "Distributed artificial intelligence empowered by end-edge-cloud computing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 591–624, Nov. 2022.

[4] W. Zhang, D. Yang, W. Wu, H. Peng, N. Zhang, H. Zhang, and X. Shen, "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3688–3703, Dec. 2021.

[5] Q. Ye, W. Shi, K. Qu, H. He, W. Zhuang, and X. Shen, "Joint RAN slicing and computation offloading for autonomous vehicular networks: A learning-assisted hierarchical approach," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 272–288, June 2021.

[6] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues," *IEEE Network Magazine*, vol. 34, no. 3, pp. 224–231, June. 2020.

[7] L. Zeng, C. Yang, P. Huang, Z. Zhou, S. Yu, and X. Chen, "GNN at the edge: Cost-efficient graph neural network processing over distributed edge servers," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 3, pp. 720–739, Dec. 2022.

[8] S. Sundar, J. P. Champati, and B. Liang, "Multi-user task offloading to heterogeneous processors with communication delay and budget constraints," *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 1958–1974, Sept. 2020.

[9] Q. Qi, X. Chen, A. Khalili, C. Zhong, Z. Zhang, and D. W. K. Ng, "Integrating sensing, computing, and communication in 6G wireless networks: Design and optimization," *IEEE Transactions on Communications*, vol. 70, no. 9, pp. 6212–6227, Sept. 2022.

[10] T. Mai, H. Yao, S. Guo, and Y. Liu, "In-network computing powered mobile edge: Toward high performance industrial IoT," *IEEE Network Magazine*, vol. 35, no. 1, pp. 289–295, Feb. 2020.

[11] M. Kim, S. Kasi, P. A. Lott, D. Venturelli, J. Kaewell, and K. Jamieson, "Heuristic quantum optimization for 6G wireless communications," *IEEE Network Magazine*, vol. 35, no. 4, pp. 8–15, Aug. 2021.

[12] T. Q. Duong, L. D. Nguyen, B. Narottama, J. A. Ansere, D. Van Huynh, and H. Shin, "Quantum-inspired real-time optimisation for 6G networks: Opportunities, challenges, and the road ahead," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1347–1359, Aug. 2022.

[13] J. Zhang, A. Sinha, J. Llorca, A. M. Tulino, and E. Modiano, "Optimal control of distributed computing networks with mixed-cast traffic flows," *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1760–1773, Aug. 2021.

[14] D. Yang, K. Gong, J. Ren, W. Zhang, W. Wu, and H. Zhang, "TC-Flow: Chain flow scheduling for advanced industrial applications in time-sensitive networks," *IEEE Network Magazine*, vol. 36, no. 2, pp. 16–24, Apr. 2022.

[15] C. Zhang, M. Zhao, L. Zhu, W. Zhang, T. Wu, and J. Ni, "FRUIT: A blockchain-based efficient and privacy-preserving quality-aware incentive scheme," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3343–3357, Dec. 2022.

Weiting Zhang [S'20, M'21] (wtzhang@bjtu.edu.cn) earned the Ph.D. degree in Communication and Information Systems with the Beijing Jiaotong University, Beijing, China, in 2021. From Nov. 2019 to Nov. 2020, he was a visiting Ph.D student with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Starting from Dec. 2021, he works as an associate professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include industrial Internet of Things, deterministic networks, edge intelligence, and machine learning for network optimization.

Dong Yang [M'11] (dyang@bjtu.edu.cn) received his B.S. degree from Central South University, Hunan, China, in 2003 and Ph.D. degree in Communications and Information Science from Beijing Jiaotong University, Beijing, China, 2009. From March 2009 to June 2010, he was a Post-Doctoral Research Associate with Jonkoping University, Jonkoping, Sweden. In August 2010, he joined the School of Electronic and Information Engineering, Beijing Jiaotong University. Since 2017, he is a Full Professor of Communication Engineering in Beijing Jiaotong University. His research interests include network architecture, wireless sensor networks, industrial network and Internet of Things.

Chuan Zhang [S'19, M'21] (chuanz@bit.edu.cn) received his Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2021. From Sept. 2019 to Sept. 2020, he worked as a visiting Ph.D. student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently an assistant professor at School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include secure data services in cloud computing, applied cryptography, machine learning, and blockchain.

Qiang (John) Ye [S'16, M'17, SM'21] (qiang.ye@ucalgary.ca) received the PhD degree in Electrical and Computer Engineering from the University of Waterloo, ON, Canada, in 2016. Since Sept. 2023, he has been an Assistant Professor with the Department of Electrical and Software Engineering, Schulich School of Engineering, University of Calgary, AB, Canada. Before joining UCalgary, he worked as an Assistant Professor with the Department of Computer Science, Memorial University of Newfoundland, NL, Canada from Sept. 2021 to Aug. 2023 and with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, USA, from Sept. 2019 to Aug. 2021, respectively. He was with the Department of Electrical and Computer Engineering, University of Waterloo as a Postdoctoral Fellow and then a Research Associate from Dec. 2016 to Sept. 2019. He is a Senior Member of IEEE.

Hongke Zhang [M'13, SM'16, F'20] (hkzhang@bjtu.edu.cn) received the M.S. and Ph.D. degrees in Electrical and Communication Systems from the University of Electronic Science and Technology of China, Chengdu, China, in 1988 and 1992, respectively. From 1992 to 1994, he was a Postdoctoral Researcher with Beijing Jiaotong University, Beijing, China, where he is currently a Professor with the School of Electronic and Information Engineering and the Director of the National Engineering Research Center on Advanced Network Technologies. His research has resulted in many papers, books, patents, systems, and equipment in the areas of communications and computer networks.

Xuemin (Sherman) Shen [M'97, SM'02, F'09] (sshen@uwaterloo.ca) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.