Data-centric Infrastructure for Supporting Data Processing in the IOWN Era and Its Proof of Concept

Ryosuke Kurebayashi, Teruaki Ishizaki, Sampath Priyankara, Christoph Schumacher, and Shintaro Mizuno

Abstract

"Data-centric infrastructure" is an information and communication technology infrastructure for highly efficient processing of data scattered over a wide area, and in a data-driven society, it is expected to be used as the foundation of large-scale cyber-physical systems (CPSs). The concept of a data-centric infrastructure, which effectively uses accelerators and the All-Photonics Network to provide a data-processing pipeline, is outlined in this article. A step-by-step demonstration of the concept is introduced by using video analysis as a use case concerning a CPS.

Keywords: data-centric infrastructure, disaggregated computing, accelerator

1. Towards creation of a data-driven society

Owing to recent improvements in sensing technology and advances in Internet of Things, digital transformation, and artificial-intelligence (AI) technology, the data-driven society is upon us. The data-driven society aims to address social issues and create new value by distributing and combining various types of data in physical space (i.e., the real world) and cyberspace (i.e., the online world of the Internet, computers, etc.) across a wide range of industries and fields. One of the core systems supporting this data-driven society is a cyber-physical system (CPS). A CPS analyzes vast amounts of data obtained from physical space in cyberspace and feeds the analysis results back to the real world for optimal control in the real world. CPSs have begun to be used in specific fields, such as constructing smart factories and traffic optimization. In the data-driven society, these various CPSs must be scaled up, applied to all fields, and interconnected.

While various technical issues need to be addressed before a large-scale CPS can be implemented, this article focuses on the information and communication technology (ICT) infrastructure responsible for data processing. It had been sufficient to process data in individual CPSs siloed by purpose and processing method. In contrast, a large-scale CPS will lead to two capabilities: (i) high-speed distribution of data among far-more-numerous and geographically dispersed entities (humans, systems, devices, etc.) that distribute more data than is currently possible and (ii) analysis of a much larger variety and volume of data. The ICT infrastructure to support such a large-scale CPS must provide a network connection with high quality of service (QoS) between geographically dispersed entities, sufficient computational resources responsible for analysis, and a highly efficient data-processing pipeline without bottlenecks to meet the enormous computing-resource demands.

2. Data-centric infrastructure

Focusing on an ICT infrastructure that can meet the

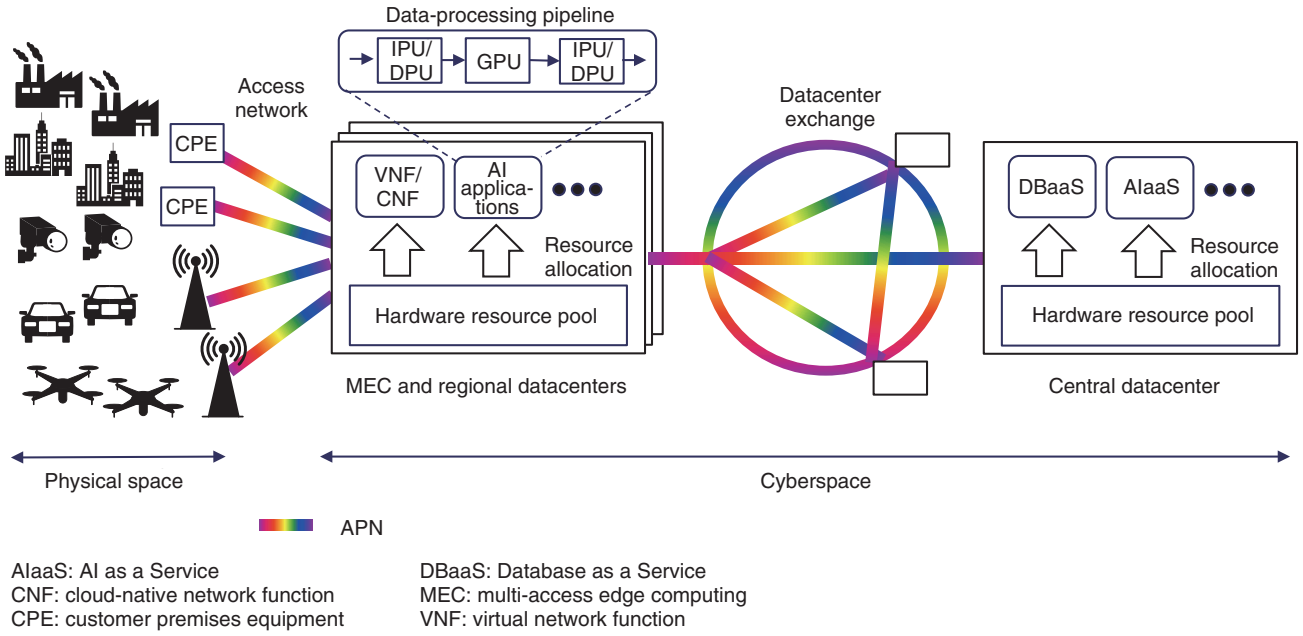

Fig. 1. Illustration of implementation of CPS using DCI.

requirements of a large-scale CPS, NTT is considering a data-centric infrastructure (DCI) subsystem that uses the Innovative Optical and Wireless Network (IOWN). The DCI is an ICT infrastructure that efficiently processes data scattered over a wide area in physical space and cyberspace by optimally combining widely distributed computing resources. The implementation configuration of a large-scale CPS based on the DCI is shown in **Fig. 1**. The main features of the DCI are listed below and explained hereafter.

- Highly efficient data-processing pipeline using accelerators
- Integration with the All-Photonics Network (APN)
- Formation of an open ecosystem

2.1 Highly efficient data-processing pipeline using accelerators

The first feature of the DCI is a highly efficient data-processing pipeline that uses accelerators. Conventional data processing is focused on software processing on a host central processing unit (CPU). Various accelerators have been used in areas where computational costs are high. Typical examples of accelerators are graphics-processing units (GPUs) for AI and media processing and smart network-interface cards (NICs), infrastructure-processing units (IPUs), and data-processing units (DPUs) for network processing. These accelerators provide highly efficient parallel processing in specific areas and use hardware such as application-specific integrated circuits (ASICs) and field-programmable gate arrays (FPGAs) to increase processing speed. The DCI actively uses these accelerators to increase the efficiency of the data-processing pipeline.

The challenge with using accelerators as an extension of conventional technologies is to reduce bottlenecks caused by the host CPU. As shown in **Fig. 2(a)**, a host CPU is required as an intermediary to manage data transfers even when an accelerator is used during data processing. This intermediary communication handling results in a bottleneck on the host CPU and limits the number of accelerators that can be effectively handled.

Accordingly, as shown in **Fig. 2(b)**, the DCI shifts from a data-processing pipeline involving the host CPU to a mechanism by which data are transferred between accelerators in a more-autonomous manner and further processed. To create this mechanism, we are investigating remote direct memory access (RDMA), which can arrange data on remote memory without intervention by the host CPU, communication-protocol-based processing by hardware using ASICs, FPGAs, etc., and direct data transfer between accelerators. We are also investigating methods to

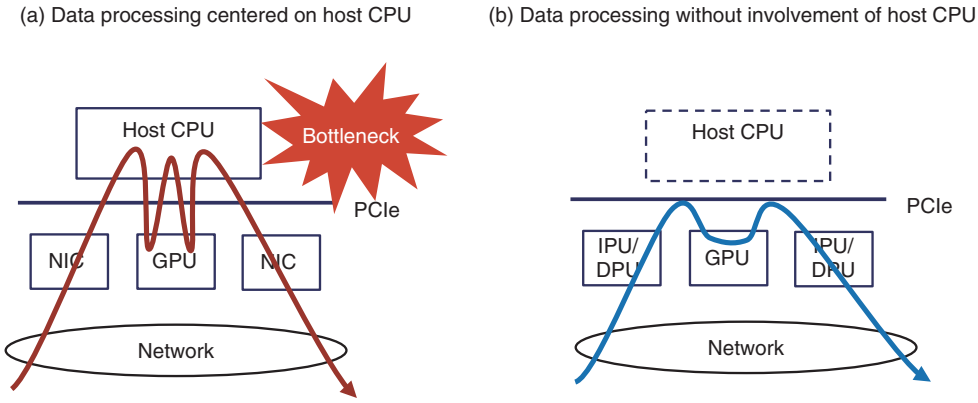

Fig. 2. Data processing without involvement of host CPU.

offload execution control, such as data-arrival detection and start-instructions processing, to the accelerators.

For an accelerator-based data-processing pipeline, it is essential to select and combine the right accelerators for a certain workload. However, the type and number of accelerators required also depends on the workload. As a result, the conventional configuration of infrastructure—which is based on servers as the basic unit—results in many wasted accelerators. In other words, when servers with uniform configurations of accelerators are arranged, depending on the workload, some accelerators will be used and others will be unused. To accommodate all workloads, however, it is impractical to have a large number of servers with different patterns of accelerator configurations in advance.

In response to the above issues, as shown in **Fig. 3**, the DCI uses a hardware resource pool, which pools the multiple devices (including accelerators) that conventionally comprise a server by connecting them via a high-speed interconnect. The optimal number of devices for a given workload are then selected from the pool and allocated. Devices that are no longer needed can be returned to the pool and made available for reuse or turned off. Therefore, the use of a hardware resource pool allows accelerators to be flexibly selected and reused in a manner that dramatically increases device utilization compared with that in the case of the conventional infrastructure configured in units of servers.

A hardware resource pool can be configured at several different layers. On the first layer, a bare-metal server is formed at the hardware level (Fig. 3(1)). Composable disaggregated infrastructure (CDI) products that extend the Peripheral Component Interconnect Express (PCIe) standard as an interconnect and link arbitrary devices to the host CPU are beginning to be commercially offered [1]. As the next-generation interconnect standard Compute Express Link (CXL) [2] becomes more widely used, it is expected to unify standards and expand functions. On the second layer, under the assumption that bare-metal servers that are connected to a collection of devices required for specific applications or tenants, workloads and/or data flows on the devices are appropriately managed at the software level on a finer-grained basis (Fig. 3(2)). As data-processing pipelines that do not involve a host CPU become mainstream, the effective size of the accelerator that can be handled by a single host CPU will increase. Consequently, the importance of resource management at the software level, as shown in Fig. 3(2), is expected to increase. The DCI will be implemented by flexibly combining these layers in accordance with use cases and diffusion of the technology.

2.2 Integration with the APN

For the DCI, the high-speed, high-quality APN is used for the access network that connects physical and cyberspace and for the connections between datacenters (which form a datacenter exchange). This feature helps configure data-processing pipelines that span wide areas between devices and datacenters without forming bottlenecks. It is envisioned that optical networking technologies, such as the APN, will be applied to not only wide-area networks but also intra-datacenter networks and interconnects in hardware resource pools.

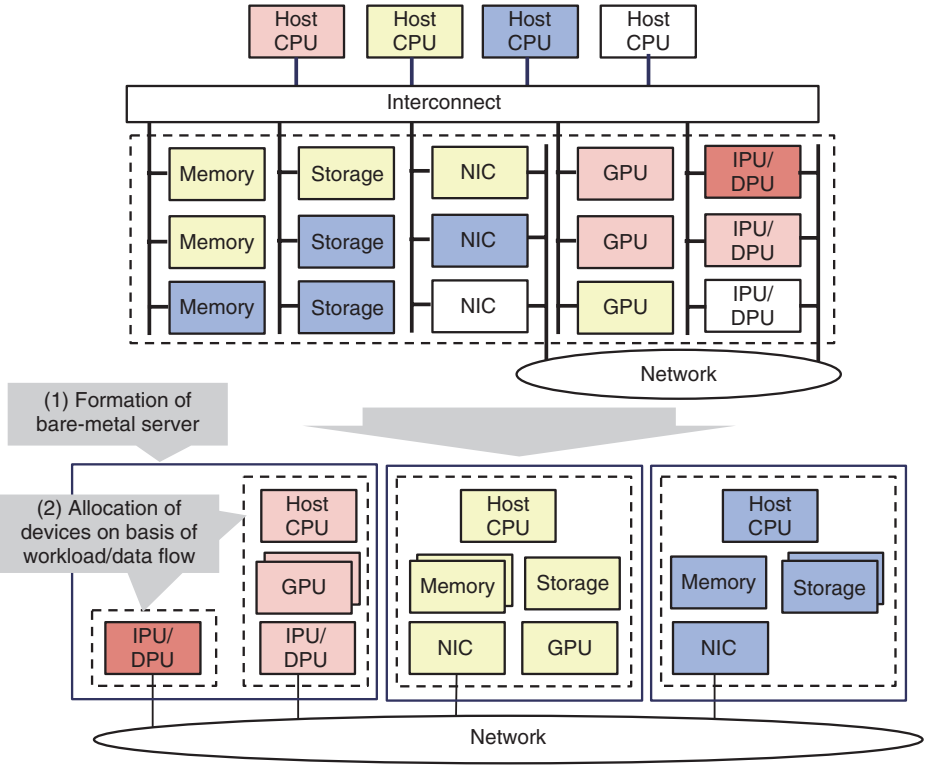

Fig. 3. Concept of hardware resource pool.

2.3 Formation of an open ecosystem

The implementation of the DCI requires a restructuring of technology that spans a wide range of disciplines—from networking to computing and from hardware to software. It is therefore important to establish an open ecosystem in which many companies can participate and pool their knowledge. For that reason, the DCI is being discussed within the global community, at bodies such as the IOWN Global Forum (IOWN GF), and its embodiment is being promoted through steady consensus building. Discussions of the DCI at IOWN GF have focused mainly on efficiency and management of hardware resources. In particular, IOWN GF is considering the formation of logical service nodes (typically corresponding to bare-metal servers in Fig. 3(1)), which are a set of devices allocated for specific applications, and the allocation of network resources among logical service nodes in consideration of QoS. The application of RDMA technology over the open APN to speed up data transfer between logical service nodes is also being discussed. The second edition of the document that summarizes these results, DCI Functional Architecture, has been published [3].

3. Proofs of concept

NTT Software Innovation Center is researching and developing disaggregated computing as one of the technologies to implement a highly efficient data-processing pipeline using accelerators, which is a feature of the DCI. To increase the scale of a CPS, we are progressing step by step with proofs of concept (PoCs) for disaggregated computing. The following two PoCs concerning real-time video analysis with a CPS are introduced hereafter (**Fig. 4**).

- PoC-1: Video-analysis pipeline using various accelerators
- PoC-2: Disaggregated computer controller to implement the “as a service” model

3.1 PoC-1: Video-analysis pipeline using various accelerators

In this PoC, we demonstrate the effectiveness of a data-processing pipeline that combines various accelerators. Specifically, the video-analysis unit shown in Fig. 4, in which eight 4K cameras are used for detecting people, is the focus. The power consumption is then reduced by optimally configuring the data-processing

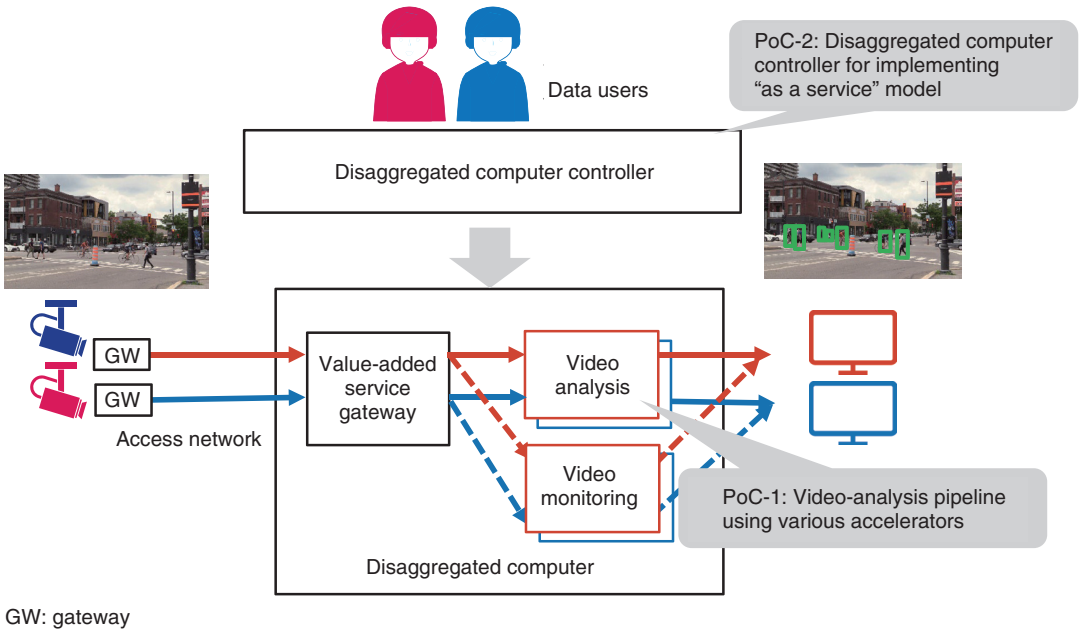

Fig. 4. Proof of concept using disaggregated computer.

pipeline for detecting people in accordance with changes in the flow of people during the day and night. This PoC is officially recognized by IOWN GF as a PoC conducted in accordance with the PoC Reference for the DCI [4].

The configuration of the video-analysis unit considered in this PoC is shown in **Fig. 5**. Camera images from the distribution servers are first input into FPGAs and decoded. For pre-processing by FPGAs, the images are then filtered and resized, and the resized images are subjected to video inference on GPUs to detect the presence of people. For the video inference, it is assumed that advanced inference and lightweight inference are used as follows.

- Advanced inference: For cameras that have captured people in previous frames, the video images are subjected to highly accurate person detection, which prioritizes detection accuracy and infers a person's presence from high-quality images at high frame rates.
- Lightweight inference: For cameras that have not captured people in previous frames, the video images are subjected to low-power-consumption, rough person detection, which prioritizes power efficiency and infers a person's presence from low-quality images at low frame rates.

During the daytime, more people are present in the scenes, so more cameras execute advanced inference; in contrast, at nighttime, fewer people are present, so more cameras execute lightweight inference. Thus, the daytime and nighttime workloads vary, so the data-processing pipeline is configured in accordance with the processing and computational load required for each scene (daytime or nighttime). The upper and lower parts of Fig. 5 show the data-processing pipelines for daytime and nighttime scenes, respectively. To demonstrate the concept of using a variety of accelerators, the data-processing pipeline of the video-analysis unit uses FPGAs and GPUs. To execute data processing without the aforementioned CPU intermediary, the FPGAs are equipped with NIC functions and proprietary circuits specialized for data transfer. That is, the FPGAs are used for decoding and pre-processing, and the GPUs are applied for inference processing. A high-performance GPU (NVIDIA A100) is used for advanced inference, and a power-saving GPU (NVIDIA T4) is used for lightweight inference. As shown in Fig. 5, the data-processing pipeline for daytime scenes is allocated more FPGAs (four) for advanced inference at higher frame rates, while the data-processing pipeline for nighttime scenes is allocated fewer FPGAs (two) for lightweight inference at lower frame rates.

The scheme used for communication between accelerators is described next. The video-coding stream from the cameras is input to the decoding

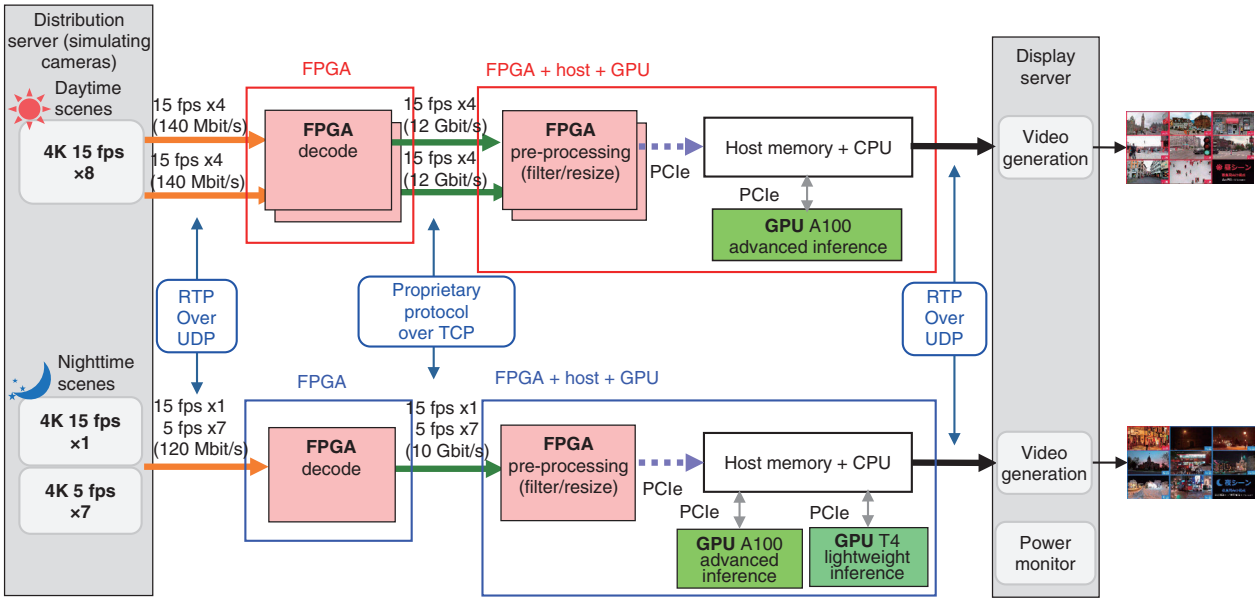

Fig. 5. Configuration diagram of video-analysis unit of disaggregated computer.

processor using the Real-Time Transport Protocol (RTP) over the User Datagram Protocol (UDP). The decoded video is then transferred to the pre-processing section at a later stage using a proprietary protocol over Transmission Control Protocol/Internet Protocol (TCP/IP). The protocol processing is terminated by the FPGAs, and the payloads are directly deployed on the memories for the user circuits of the FPGAs. This scheme achieves the effect of hardware offloading of protocol processing. In particular, the FPGA that executes the decoding process completes a series of processes from video reception to decoding and transmission within the FPGA; in other words, data processing becomes autonomous without the need for a host CPU. Data transfer from the FPGAs, which execute pre-processing, to the GPUs, which execute inference processing, is also executed on a proprietary-DMA basis with little overhead on the host CPUs, although the data are transferred through the host memory.

The power consumption of the video-analysis unit was measured. A conventional configuration (a typical configuration in 2020 was assumed) and the configuration of the video-analysis unit in Fig. 5, as well as their respective power consumptions, are compared in **Fig. 6**. Comparing the power consumptions of the video-analysis unit of the disaggregated computer and the conventionally configured computer for daytime scenes reveals that the power consumption of the former decreased from 2626.8 to 990.6 W (approximately 62% lower). This reduced power consumption is due to the fact that the video-analysis unit is highly efficient in terms of selecting the optimal accelerator, including hardware evolution (from NVIDIA T4 to A100), using accelerators in a wider range, including decoding and pre-processing, and transferring data efficiently between accelerators. Furthermore, comparing the power consumptions in the daytime and nighttime scenes confirms that the power consumption decreased from 990.6 W (daytime scene) to 696.7 W (nighttime scene). In other words, flexibly reconfiguring the pipeline in accordance with the scene can reduce power consumption from 2626.8 to 696.7 W (approximately 73% lower) compared with that of the conventional configuration.

Note that the video-analysis unit is intended for PoC and is subject to the following restrictions.

- The original circuit on the FPGA is a prototype implementation, and its power consumption can be further reduced by improving the implementation.
- Daytime and nighttime scenes are switched statically offline. In practical use, it will be necessary to upgrade the system for dynamic switching.

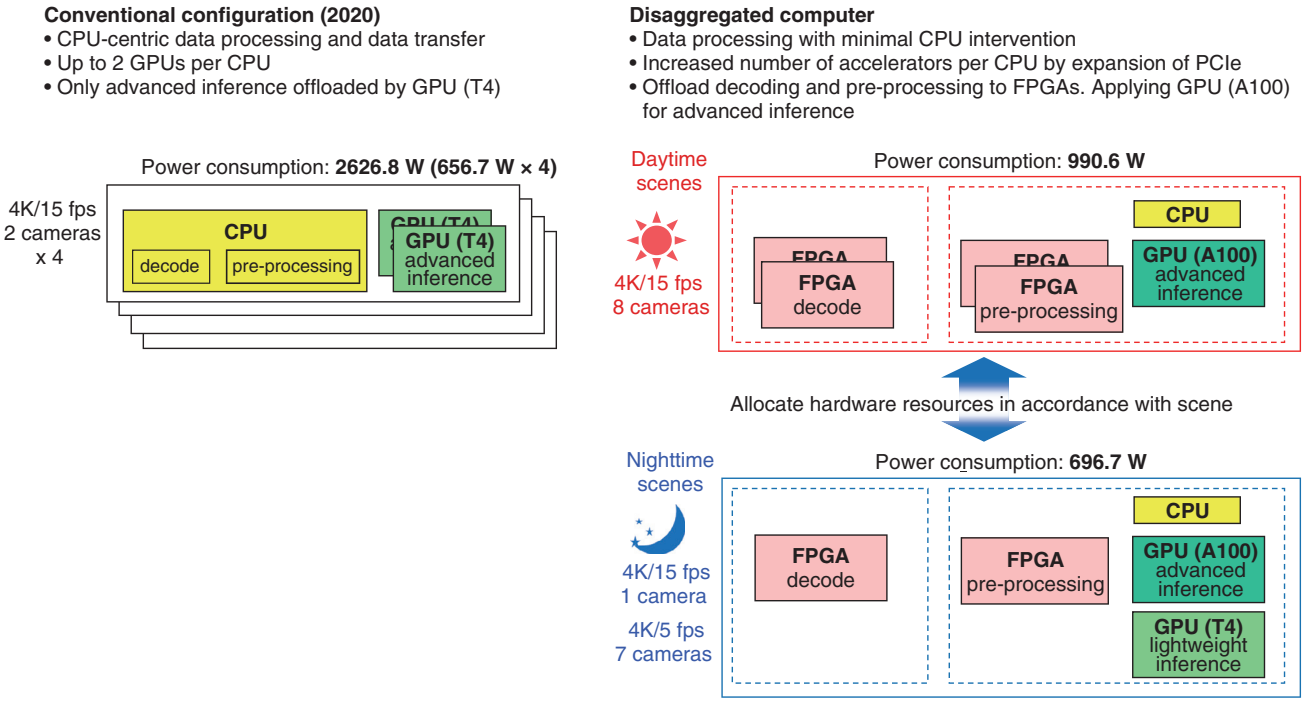

Fig. 6. Comparison of power in cases of conventional computer configuration and disaggregated-computer configuration.

3.2 PoC-2: Disaggregated computer controller that implements the "as a service" model

In PoC-2, we demonstrate that by using a disaggregated computer controller (DCC), data users without advanced expertise can easily configure data-processing pipelines that take advantage of accelerators. This PoC was exhibited at NTT R&D Forum 2023.

For this PoC, the data-processing pipeline shown in Fig. 4 is generated or reconfigured in response to requests from data users. The daytime scenes analyzed in PoC-1 are used as the video-analysis function in this PoC. Moreover, the video-monitoring unit forwards the video images appropriately for the observer. In this PoC, it is assumed there are multiple tenants. The value-added service gateway [5] aggregates and distributes the video images captured with the cameras, and a secure connection between the cameras and video-analysis and -monitoring units is provided for each tenant. In other words, generic routing encapsulation (GRE) tunnels are created and used to connect the cameras so that images can be separated for each tenant. The video data are then appropriately duplicated and distributed to the following processing units, namely, video analysis and video monitoring, in response to requests from the data users of each tenant. For this distribution, the GRE tunnels are converted to virtual local area networks.

The DCC plays an important role in this PoC. When the DCC receives a request from a data user to generate a data-processing pipeline or change the configuration of an existing pipeline, it automatically builds a data-processing pipeline for the video-analysis and video-monitoring units and changes the relevant setting, including that of the value-added service gateway unit. At that time, the data user can describe a blueprint of the requested data-processing pipeline in a YAML file that abstracts the details of the accelerator. The DCC is also responsible for software-level resource management (as shown in Fig. 3(2)). That is, it is assumed that many accelerators are already connected to the bare-metal server. According to the data-user request, the necessary accelerators are then allocated to the data-processing pipeline. The DCC is implemented by extending Kubernetes, namely, the de facto container orchestrator. Kubernetes custom resources are then used to manage the accelerators on the worker nodes (i.e., the bare-metal server mentioned above) and the connections between them.

Conventionally, the actual construction and operation of a data-processing pipeline using accelerators

requires a high level of expertise and a great deal of time. The DCC appropriately hides this work and allows it to be provided as a service. This service allows data users to enjoy the benefits of accelerator-based data-processing pipelines while remaining focused solely on the functional aspects of data analysis.

4. Conclusion and future directions

We validated the concept of the DCI through a PoC using disaggregated computing. This PoC used video analysis via a CPS to demonstrate the effectiveness of data-processing pipelines using accelerators and the usefulness of the DCI as a service. In the future, we will promote practical application of the DCI by expanding its various operational functions while promoting its cooperation with the APN, which is another feature of the DCI. We will also expand the DCI to other use cases that use video and use cases other than video processing.

These results utilize technology under study in a joint research and development project with Fujitsu Limited [6].

References

[1] Y. Mizuno, K. Hatakeyama, H. Fukuda, K. Matsui, and T. Matsuda, "A Study of Using Composable Disaggregated Infrastructures in Telecommunications Carriers," IEICE Society Conference, BI-9-4, Sept. 2022.

[2] D. D. Sharma and I. Agarwal, "Compute Express Link 3.0," https://www.computeexpresslink.org/_files/ugd/0c1418_a8713008916044ae9604405d10a7773b.pdf

[3] IOWN GF, "Data-Centric Infrastructure Functional Architecture," Ver. 2.0, Mar. 2023. https://iowngf.org/wp-content/uploads/2023/04/IOWN-GF-RD-DCI_Functional_Architecture-2.0.pdf

[4] IOWN GF, "Data-centric-infrastructure-as-a-service PoC Reference," Ver. 1.0, July 2022. https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-DCIaaS_PoC_Reference_1.0.pdf

[5] T. Yokoi, K. Mochizuki, T. Nakamura, S. Nishiyama, K. Takahashi, and T. Osaka, "Service Node Architecture Technology for Disaggregated Network Service Functions and Fixed-mobile Convergence Networks," NTT Technical Review, Vol. 21, No. 1, pp. 18–23, Jan. 2023. https://doi.org/10.53829/ntr202301fa2

[6] Press release issued by NTT and Fujitsu, "NTT and Fujitsu Embark on Strategic Alliance to Drive 'Realization of Sustainable Digital Society,'" Apr. 2021. https://www.fujitsu.com/global/about/resources/news/press-releases/2021/0426-01.html

Ryosuke Kurebayashi
Director, System Software Project, NTT Software Innovation Center.
He received a B.S., M.S., and Ph.D. from University of Tsukuba, Ibaraki, in 1998, 2000, and 2003. He joined NTT in 2003, and his research interests include computing and networking technologies for Internet of Things and AI. He is a member of the Information Processing Society of Japan (IPSJ) and the Institute of Electronics, Information and Communication Engineers (IEICE).

Teruaki Ishizaki
Senior Manager, System Software Project, NTT Software Innovation Center.
He received a B.E. and M.E. in mechanical and environmental informatics from Tokyo Institute of Technology in 2002 and 2004. He joined NTT Cyber Space Laboratory in 2004 and studied the Linux Kernel and virtual machine monitor. From 2010 to 2013, he joined the cloud service division at NTT Communications and developed and maintained cloud and distributed storage services. He is currently studying a persistent memory programming model, RDMA programming model, cloud-native computing, and memory-centric computing.

Sampath Priyankara
Senior Manager, System Software Project, NTT Software Innovation Center.
He received a B.E., M.E., and Ph.D. in information science from Osaka University in 2008, 2010, and 2013. He joined NTT in 2013, and his research interests include disaggregated computing architecture, data-centric computing, and cloud-native computing.

Christoph Schumacher
Senior Manager, System Software Project, NTT Software Innovation Center.
He received a doctoral degree in engineering from RWTH Aachen University, Germany, in 2015 and joined NTT in 2020. His interests include future large-scale distributed applications and the accompanying new requirements toward computing and communication infrastructures.

Shintaro Mizuno
Project Manager, System Software Project, NTT Software Innovation Center.
He received a B.E. and M.E. in mechanical and environmental informatics from Tokyo Institute of Technology in 1995 and 1997. He joined NTT Software Laboratory in 1997 and studied distributed computing and authentication systems. Since 2011, he has been engaged in the R&D of computing systems, especially on cloud computing technologies. He is currently a manager of an R&D project to develop the next-generation computing architecture for IOWN.