Study of Storage Services at IOWN Global Forum

Tomohiro Inoue

Abstract

The IOWN Data Hub is a new data service that integrates database, storage, and other functions required in the Innovative Optical and Wireless Network (IOWN) era. The IOWN Data Hub is intended to be commonly used by other applications by leveraging the advanced IOWN infrastructure while maintaining compatibility with existing applications. This article introduces the status and the released document of the IOWN Data Hub discussed in the IOWN Global Forum.

Keywords: data hub, data-centric, storage

1. IOWN Data Hub

The IOWN Global Forum (IOWN GF) is a forum where participating companies collaborate to discuss common service-layer mechanisms that can leverage the new advanced communication and computing infrastructure created by the Innovative Optical and Wireless Network (IOWN). The IOWN Data Hub (IDH) is a data-hub service (integrated database, storage, and data-management services) for applications in the IOWN era. It is intended to be a data management and sharing infrastructure that enables fast and reliable data processing, usage, and exchange among multiple parties.

The IDH is an application functional node on top of the IOWN Data-Centric Infrastructure (DCI) architecture [1] and intended to be commonly used by other application functional nodes by maintaining compatibility with existing applications while leveraging the features of the IOWN Open All-Photonic Network (Open APN) [2] and DCI. IOWN GF defines a common infrastructure so that multiple service providers can develop and deploy IDH services.

In January 2022, the “Data Hub Functional Architecture” document (IDH document) [3] was released with the cooperation of forum members, including NTT, Oracle, and NEC. The document defines multiple models of the IDH. In the various use cases discussed in IOWN GF, there are numerous demand patterns expected of data hub services. IOWN GF typified these demands and organized them into several Service Types and Service Classes. The document also presents a reference implementation model for each Service Class using the IOWN infrastructure.

2. IDH Service Type and Service Class

2.1 Required data-hub services

IOWN GF has been investigating the system architecture and data flows for several use cases [4] towards full-stack optimization using the IOWN infrastructure, regardless of the layer structure of the existing system, which is a unique approach of IOWN GF that covers a wide range from communication layers to service layers. Nevertheless, data flows resulting from the analysis are conceptually close to those required by current Internet applications. It is assumed that the data flow used by applications in the early stages of IOWN deployment will not differ significantly from existing applications for compatibility reasons. This section introduces the Service Type and Service Class defined on the basis of these discussions.

2.2 Service Types

A Service Type is defined as a unit of functionality and behavior that will be implemented by one or

Table 1. IDH Service Types.

Service Type	Example of data flow	Supported APIs
Distributed Relational Database (DRD)	Structured data to be registered in RDBMS, labeled data of detected objects (output of AI inference), etc.	JDBC and ODBC that support SQL. REST API that accepts SQL-based data queries and responds in JSON format.
Key-Value Store (KVS)	Time-series data, sensor data, etc.	REST API (CRUD operations on single or multiple Key Values, support for simple queries)
Graph Store	User-related data, links to related event data, semantic-knowledge data, etc.	REST API that accepts queries based on SPARQL, PGQL, and other domain-specific languages (DSLs), and responds in JSON format
Message Broker	Notification messages, voice messages, etc.	General message broker API (e.g., Apache Kafka compatible API)
Object Storage	Video data, sensor data (e.g., point cloud), etc.	General Object Storage API (e.g., REST API that is compatible with AWS S3)

* Currently, only general requirements are listed; more detailed requirements will be discussed in the future.

AWS: Amazon Web Services
CRUD: Create, Read, Update and Delete
JDBC: Java Database Connectivity
JSON: JavaScript Object Notation
ODBC: Open Database Connectivity
PGQL: an SQL-based query language for the property graph data model.
RDBMS: relational database management system
REST: Representational State Transfer
SPARQL: Simple Protocol and RDF (Resource Description Framework) Query Language
SQL: Structured Query Language

several IDH services. Specifically, requirements such as supported application programming interfaces (APIs), queries, and data types are defined, and external applications can treat IDH services like databases or object storage by specifying a Service Type. Five Service Types are currently defined (**Table 1**).

(1) Distributed Relational Database

Distributed Relational Database (DRD) is a Service Type for storing structured data intended for data-analysis use cases such as online transaction processing in general databases and online analytical processing in data warehouses. Using this Service Type, various applications will store and use data without executing any extra data-conversion processing outside the IDH. Specific data that some structural queries can identify should be handled by DRD as long as the processing speed permits.

(2) Key-Value Store

Key-Value Store (KVS) is a Service Type used when storing a vast amount of data, although individual records are not that large. For example, time-series data and sensor data fall under this category.

(3) Graph Store

This Service Type stores data that are modeled as a graph. The stored data may follow a pre-defined schema called an ontology. The ontology describes how the data are structured and enables application developers to form semantic queries. Semantic queries can retrieve information on the basis of associations or a range of contexts.

(4) Message Broker

This Service Type is intended for use in the following situations where data cannot simply be added to existing data storage.

- There are so many messages that they cannot be accommodated by other Service Types and must be stored in several queues.
- The number of providers and users of data is enormous and dynamically changing.
- The data hub handles data of mixed sizes, types, and formats and requires a complex pipeline for format conversion and aggregation.
- The data hub only needs to relay messages and does not require data persistence.

(5) Object Storage

This Service Type stores relatively large object data, such as semi-structured data, and unstructured data, such as videos, images, and logs, at low cost. It is expected to be used for the long-term storage of big data and provide data-analysis applications such as artificial intelligence (AI) and data lakes for future data utilization.

2.3 Service Class

To implement IOWN use cases, an IDH service implementing one Service Type is insufficient, and multiple Service Types may be required simultaneously. The IDH defines the Service Classes shown in **Table 2** as implementations of services with one or more Service Types.

The Basic Service Class inherits only one of each Service Type. The Applied Service Class provides a

Table 2. IDH Service Classes.

Basic Service Class: inherits only one corresponding Service Type		
	DRD / KVS / Graph Store / Message Broker / Object Storage	
Applied Service Class: inherits multiple service types and/or integrates multiple service classes		
	Converged Database	• DRD • KVS • Graph Store • Message Broker
	Context Broker	• Graph Store • Message Broker • DRD
	Virtual Data Lake	• Object Storage
	Virtual Data Lake House	• DRD • KVS • Graph Store • Message Broker • Object Storage

package of functionalities that more complex use cases collectively require from the IDH, and it inherits from and integrates multiple Service Types. The IDH document defines the following four Applied Service Classes.

(1) Converged Database

This Service Class is designed for cases in which different types of data are to be processed simultaneously. For example, let us consider a scenario in which we analyze people's behavior in a building and provide appropriate support. We need to manage and process interactions of people with people, and people with things all at once, and the data structure that represents this needs to be highly flexible. However, if different IDH services are used for each data structure and data conversion is carried out at the application level, the real-time performance required by IOWN cannot be achieved. Converged Database provides a platform for high-speed simultaneous processing of various data to meet these needs.

(2) Context Broker

This Service Class functions as a sophisticated broker that enables applications to request data by flexibly specifying the context. Context can be expressed as a thing (e.g., a specific building), type of thing, geographic area, temporal area, or combination of these, allowing the specification of information for particular conditions.

(3) Virtual Data Lake (Federated Object Storage)

This Service Class extends the current concept of data lakes to provide a unified data lake by virtually integrating not only the IOWN infrastructure but also geographically dispersed data sources such as clouds and on-premises. When considering the implementation of various use cases of IOWN, not all data are necessarily managed within the IOWN infrastructure. Still, cases in which data in multiple cloud, edge, and on-premise environments are analyzed and processed across them are also anticipated. For these cases, Virtual Data Lake provides a mechanism to bundle multiple geographically dispersed storage units and present them as a single object storage unit.

(4) Virtual Data Lake House

IOWN envisions a future in which data held by multiple companies and organizations are shared openly and securely to solve social problems. For example, in a society where renewable energy sources account for most of the energy supply, it is expected that the entire system will be able to be stably controlled by sharing real-time data on the power demand of each company and household, in addition to data on the amount of power supplied. For this purpose, Virtual Data Lake House provides secure and transparent access to various data sources belonging to different owners.

3. Advantages of the IDH

As mentioned above, the IDH supports many of the features that future applications will require in databases and storage services. The IDH also takes advantage of the features of IOWN to achieve functions and performance that are insufficient for conventional databases due to infrastructure limitations. This section describes the current limitations and how they are expected to be solved in the IDH.

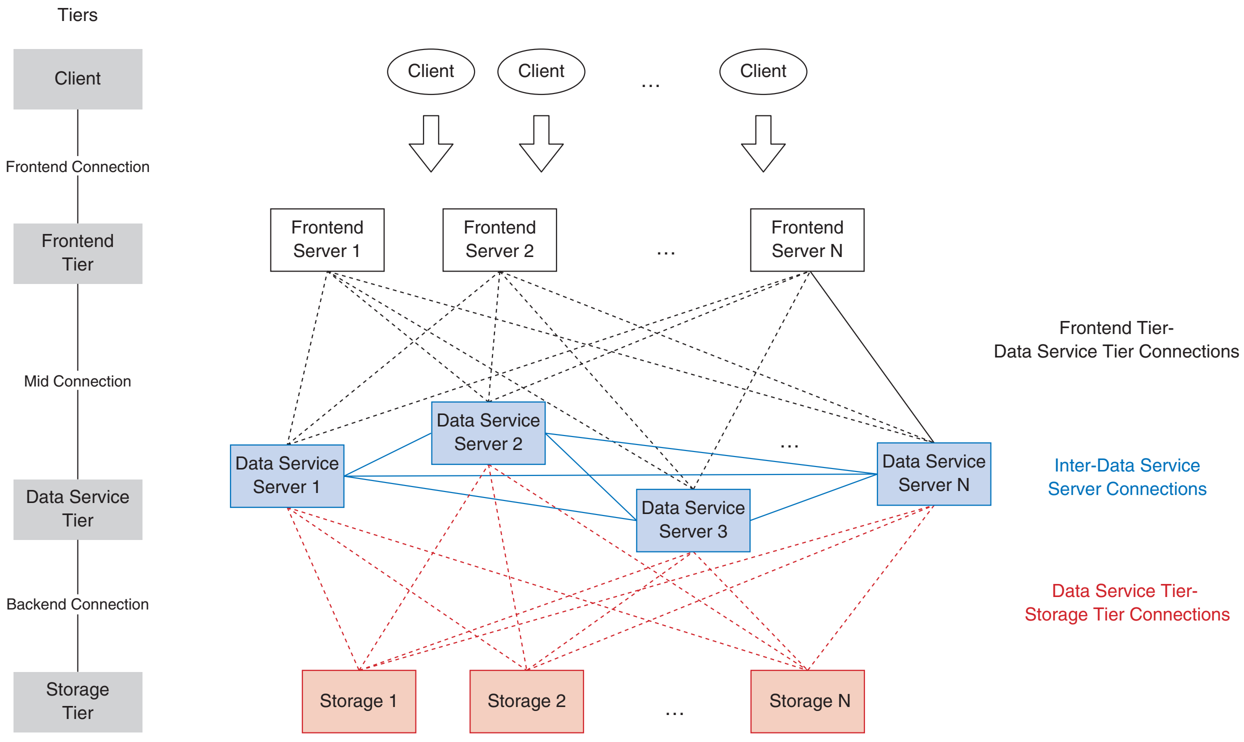

Fig. 1. IDH reference implementation model.

3.1 Limitations of the current situation

For example, suppose we use the Object Storage service implemented in the current cloud model to upload or download 10-GB-class data. The multiple copies of data between the client and storage where the data are stored will become a bottleneck, and stable performance will not be guaranteed. In addition, with the current technology, complex queries that require DRD to join petabytes of data across multiple servers are very slow due to the bottleneck of network connections between database servers. Similarly, for full-scan queries that read several exabytes of data from storage, data transfer from storage becomes a bottleneck. The IDH document also points out the limitations of current cloud-based implementations for other Service Classes.

3.2 Structure and improvements of the reference implementation model

The reference implementation model for the IDH is shown in **Fig. 1**. Although the details differ for each Service Class, this model is the common internal structure for all Service Classes. This structure consists of a Client that requests data from the IDH service, a Frontend Tier that provides load balancing, query routing, protocol conversion, etc., a Data Service Tier that provides data from the local cache or connected storage layer in response to data-access requests, and a Storage Tier that stores data. Each layer consists of multiple components (e.g., servers). The IDH uses IOWN infrastructure technology to accelerate inter-layer and inter-component connections and the operation of each component as follows.

(1) Optimization of communication between the Client, Frontend Tier, and Data Service Tier

The Client exchanges data from the Data Service Tier via the Frontend Tier. To achieve near real-time processing, the latency for these communications needs to be short, and the bandwidth needs to be high, especially when large amounts of data must be transferred repeatedly. In the IDH, Open APN provides direct optical communication between two endpoints to ensure low-latency high-bandwidth connection as needed.

(2) Optimization of communication within the Data Service Tier

When data processing occurs frequently and a large amount of data is exchanged between servers in the Data Service Tier, communication between servers becomes a bottleneck in the data service. Current implementations of inter-server communication using blocking protocols, such as TCP (Transmission Control Protocol), tend to waste server CPU (central processing unit) cycles due to input/output latency. In the IDH, the DCI mechanism enables a dynamic combination of multiple resources to build high-performance clusters. Furthermore, using protocol optimization in DCI (e.g., fast non-blocking communication using a relational database management system (RDMA)), the IDH is designed to process and manage vast amounts of data in near real time.

(3) Optimization of communication between the Data Service Tier and Storage Tier

In IOWN use cases, the total amount of data transferred in a single request can be several hundred terabytes. If there is not enough network bandwidth to communicate between the Data Service Tier and Storage Tier, the service will not respond for a long time. Increasing the network bandwidth with Open APN and optimizing the data transfer with DCI (RDMA) can also help to eliminate this bottleneck.

(4) Making each component smart

In addition to network acceleration, the IDH will actively offload data pre-processing to each component using smart network interface cards. For example, the Storage Tier component filters and aggregates data, reducing the amount of data transferred on the network to less than 1/100th. Also, by distributing to the Client and Frontend Tiers where the data are located in the Data Service Tier, data access requests can be sent directly to the most appropriate component, reducing excess data transfers between members and improving latency and bandwidth.

With these mechanisms, the IOWN reference implementation model is expected to run much faster than the current cloud-based implementation model. The IDH document describes more specific improvements in the implementation model of each Service Class.

4. The Future of the IDH

The IDH document summarizes the technical position of the IDH in the IOWN infrastructure and provides reference implementation models to solve existing cloud-based data processing bottlenecks. However, this should be called the Minimum Viable Version of the IDH features. Future discussions by IOWN GF will lead to more detailed APIs for each Service Type and a better implementation model. In addition, the advanced data processing flow required by future IOWN applications will require more advanced functionalities. For example, in use cases such as smart cities, data owned by multiple organizations will need to be handled across multiple data users. This will require the development of authentication and authorization mechanisms, as well as means to control and audit data usage policies. A feature such as secure computation that executes processing while keeping data secret should also be considered as a function of the IDH. The IOWN GF community will continue to discuss the ideal data-hub services required in the IOWN era.

References

[1] IOWN GF, “Data-Centric Infrastructure Functional Architecture,” https://iowngf.org/technology/#Data-Centric-Infrastructure

[2] IOWN GF, “Open All-Photonic Network Functional Architecture,” https://iowngf.org/technology/#Open-All-Photonic-Network

[3] IOWN GF, “Data Hub Functional Architecture,” https://iowngf.org/technology/#Data-Hub

[4] IOWN GF, “Reference Implementation Model (RIM) for the Area Management Security Use Case,” https://iowngf.org/technology/#Reference-Implementation-for-CPS-AM

Tomohiro Inoue

Senior Research Engineer, Supervisor, Social Information Sharing Research Project, NTT Social Informatics Laboratories.

He received a B.E. and M.E. from the University of Tokyo in 1999 and 2001. He joined NTT in 2001 and is currently engaged in the research and development of data hubs, trusted data-sharing infrastructure, and global data-space interconnection as the leader of Information Sharing Architecture Group.