IOWN Data Hub PoC Reference

Classification: APPROVED REFERENCE DOCUMENT

Confidentiality: PUBLIC

Version 1.0

October 21, 2022

[IDH PoC Reference]

Legal

Contents

List of Figures

List of Tables

1. Introduction

The IOWN Global Forum (hereinafter referred to as “IOWN GF”) was established to enable a fully digitalized world while ensuring the sustainability of society, and since then, many research and development activities are underway in various technical domains, including IOWN GF Data Hub (hereinafter referred to as “IDH”). The IDH is a data management, utilization, and sharing solution concept built on Open APN and disaggregated computing infrastructure, as described in the IOWN Global Forum Data Hub Functional Architecture document (hereinafter referred to as “IDH FA Document”) [IDH FA Doc].

This document describes various IDH-relevant proof-of-concept (hereinafter referred to as “IDH PoC”) activities, which are about to be conducted.

1.1. Purpose

Toward 2025 – 2030, various digital experience (hereinafter referred to as “DX”) services such as Smart Factory, Smart Grid, and Metaverse as described in Chapter 2 and Annexes A, B, and C are expected to be realized. However, this is not easy because of various severe requirements such as millions of simultaneous connections, dynamic scalability for millions of streams with various sizes and large fluctuations, real-time responses less than tens of milliseconds, joint-processing of PB-class data with dozens of servers for flexible and high-performance big data analysis, and instant data exchange between multiple parties must be met.

The purpose of the IDH PoC is to build the foundation for realizing DX services by verifying the IDH function architecture, which was designed to meet such severe requirements based on a hierarchically distributed architecture as described in the IDH FA Document. By quantifying IDH capabilities and identifying areas for improvement, developers will be able to design the foundation for real-time and economically rational DX services.

1.2. Objectives

To achieve the purpose described in Section 1.1, the following objectives are defined in the execution of the PoC:

- **Verification of the hierarchically distributed implementation model**

  The first objective of the PoC is, based on the hierarchically distributed architecture of the IDH solutions, to demonstrate that 1) very short end-to-end latency such as 10 milliseconds is achieved by leveraging regional edge centers and software components designed for edges, 2) data is smoothly moved from the resource-poor regional edge clouds to the resource-rich central cloud for long-term data storage by using technologies enabling efficient data transfer including Open APN, and 3) the feasibility of distributed data processing utilizing distributed data centers connected through Open APN, which is not easy because of the lack of solutions and cost of data transfer in today's implementation model. Please refer to Section 4-1 for the hierarchically distributed architecture of the IDH solutions, and Sections 4-2, 4-3, 4-4, 4-5, and 4-6 for the details of PoC scenarios.

- **Demonstration of the advantage and benefits of IDH**

  The second objective of the PoC is to verify the advantage and benefits of the IDH technologies over those existing technologies. With well-designed IDH functional architecture, read-and-write response time jitter of RDB, KVS, Message Broker, Graph Store, and Object Storage is expected to be reduced by one-tenth or less compared to today's implementation models, and similarly, the scalability limit will be greatly improved so that RDB and KVS will be able to handle more than 1 million per second read-and-write operations. The correctness of such a hypothesis shall be demonstrated by conducting the PoC. For details, please refer to sections 4-2, 4-3, 4-4, 4-5, and 4-6.

- **Gaining insights into tuning points of IDH**

The last objective of the PoC is to gain insights into such tuning points by performing tests while changing the test conditions. To achieve the best performance of the IDH solution, the knowledge around the optimal network parameter values such as TCP Window Size, the optimal batch size for data transfer processing, the effect of placing the gateway for block storage access, and the performance gains by compromising data persistence, shall be acquired by conducting PoC with changing test conditions. For details, please refer to variable conditions described in sections 4-2, 4-3, 4-4, 4-5, and 4-6.

1.3. Scope

Target DX Services

There are various eagerly awaited DX services in the world. IOWN GF plans to analyze requirements and develop required technologies. In the first step of this IDH PoC, three DX services described below are selected and how IDH can be technically used in realizing these services will be verified.

- Smart Factory
- Smart Grid
- Metaverse

Target Service Classes

In IDH FA Document, nine IDH service classes are described with implementation models defined. The IDH PoCs will evaluate these IDH service classes by a step-by-step approach. In the first step of IDH PoC, the following IDH service classes will be evaluated:

- Distributed RDB
- Key-Value-Store (KVS)
- Graph Store
- Object Storage
- Virtual Data Lake (Federated Object Storage)
- Message Broker

Test Models

To clarify the advantage and benefits of IDH, at least two models, today's implementation model, and the IDH-based implementation model, as well as related technologies used, will be compared.

- Today's implementation models and technologies to be tested are determined based on the typical reference architecture published by cloud vendors and/or system integrators.
- IDH-based implementation models and IDH service classes are determined by considering a high-speed network based on Open APN and/or DCI interconnect and the well-designed functionality of IDH service classes that reduce data movement and increase data processing density.

2. Target DX Services Scenarios and Requirements

As discussed in Chapter 1, the first step of the IDH PoC is to analyze the requirements of DX services. For that purpose, in this chapter, three DX services, Smart Factory, Smart Grid, and Metaverse, are selected and analyzed to derive data processing patterns and other requirements that IDH solutions fulfill.

2.1. Smart Factory

Scenarios

The Smart Factory is a solution concept that fully leverages sensors, robots, IoT, Big Data, and AI technologies to automate the production and maintenance processes, decrease production losses, increase the overall production efficiency, and save energy consumption.

As described in section 3.3.1 of the IOWN Global Forum Cyber-Physical System Use Case Document (hereinafter referred to as "IOWN GF CPS UC Document") [IOWN GF CPS UC], the Smart Factory services include the following scenarios:

- Real-time remote monitoring and control of manufacturing machines and robots
- Anomaly detection and failure prediction with various sensors and AI technology
- Manufacturing process optimization to increase the yield and reduce the energy consumption and CO2 emission

Requirements

To build Smart Factory services, it is necessary to install sensors on each manufacturing machine and robot and to collect and process data in real-time. Considering various expected Smart Factory services, the following requirements must be met[*1]:

- The feedback loop for remote machine/robot control must be done in cycles of 10 milliseconds or less for tens of thousands of manufacturing machines and robots. Considering the time required for data processing, the one-way communication latency would need to be sub-milliseconds.
- Train Machine learning and AI models with collected data, which can be 1 TB per machine/robot and tens of PB per factory if accumulated for 10 years or so, on variable resources like the cloud.
- Data must be securely stored, managed, and processed, as it contains manufacturing IP-relevant information.

*[1]Note: Please refer to Annex A for a detailed analysis of the requirements

The above requirements are not easy if compared with the requirements of today's factory systems. However, it is still worth the effort because it is directly linked to the competitiveness of the manufacturing industry. Therefore, many manufacturing companies want to introduce these services in their plants and have started research and development, although typical solutions are still insufficient because of the gaps described in section 3.

2.2. Smart Grid

Scenarios

As described in section 3.6 of the IOWN GF CPS UC Document, the increase in renewable energy destabilizes the power grid, which poses challenges for the further deployment and wide adoption of renewable energy. A new solution for building the electricity distribution network utilizing digital technologies, the so-called Smart Grid, is awaited to overcome this challenge.

Smart Grid solution provides a foundation to aggressively accommodate renewable energy through services described below:

- Real-time monitoring and accurate forecasting of the power supply and demand balance
- Charge and discharge controls of the distributed energy storage resources
- Intelligent electric power interchange across multiple aggregators (Smart Grid providers)

Requirements

In order to build Smart Grid services, it is necessary to collect data from various points in the power grid system, such as batteries of electric cars, as well as external parties, and process these data to determine the power supply and demand balance in real-time. In addition, to deal with the excess and deficiency of electricity supply, it is necessary to perform charge and discharge controls for distributed energy storage resources, and interchange electricity among Smart Grid providers in real-time while making precise near-future predictions. Considering these, the requirements of the typical U.S. County-level or local Japanese prefecture-level regional Smart Grid system will be as follows[*2]:

- Continuously collect power balance data from a large number of sources, such as hundreds of thousands of switchgear and/or transformer devices (more than once per second), millions of smart meters (more than once per minute), and millions of distributed energy storage resources, including electric vehicles (more than once per minute), mostly over the wireless network.
- Predict power balance statuses at tens of thousands of location points for the next 24-48 hours with a time resolution of at least 1-minute intervals or finer by considering not only internally collected data but also external data, e.g., weather conditions, operation plans of the large factories, etc.
- Control recharging and discharging of distributed energy storage resources every second or less, considering the power supply and demand balance statuses at tens of thousands of location points and the usage policies of distributed energy storage resources that their owners set.
- Coordinate electricity exchange with other Smart Grid providers connected at 1-minute intervals or less if electricity balance cannot be made within the Smart Grid provider's business.

*2Note: Please refer to Annex B for the detailed analysis of the requirements

The above requirements are severe as a lot of data needs to be collected over the wireless network, a complex structure, i.e., the network of the Smart Grid system needs to be considered, and every data has to be mapped to it. Therefore, it is very difficult to determine optimal control of the Smart Grid system (see B-4 for the detailed requirements of Smart Grid) with short latency.

2.3. Metaverse

Scenarios

The metaverse is a virtual space where people join together and interact with each other through their avatars. In such a virtual space, there are not only avatars of the participants but also buildings and vehicles, as in the real world. And people can enjoy various services there and get an experience close to the real world.

For example, there will be an interactive live music service as described in the IOWN Global Forum AI-Integrated Communications Use Case Document (hereinafter referred to as "IOWN GF AIC UC Document") [IOWN GF AIC UC]. In this service, people can attend the concerts of their favorite artists anytime, anywhere, without having to physically move to the actual concert hall.

Requirements

To build metaverse services in general, it is necessary to collect information about participants, represent them as avatars in virtual space, and let avatars interact together in the virtual space. For this purpose, the following requirements must be met in the production system of a high-quality Metaverse service[*3]:

- Collect motion data for up to millions of participants and reflect it as the movement of the avatars each of which represents the corresponding participant in a cycle of tens of milliseconds. Considering the time required for virtual space management and rendering, the one-way communication latency would need to be a single digit millisecond.
- Let avatars move and communicate with each other in a virtual space, which may also include many virtual buildings and other structural elements, based on the collected data.
- Render the scenery of the virtual space based on the viewpoint of the avatar with a motion-to-photon latency of tens of milliseconds or less.

*[3]Note: Please refer to Annex C for the detailed analysis of the requirements

The above requirements are more than severe as the metaverse service has to deal with a complex and detailed virtual space and granular movements of massive avatars in a very short cycle, e.g., 33 milliseconds for a 30 FPS service. In addition, the performance of selecting the data for rendering is also challenging. This means it is necessary to extract only the data about other avatars that can be viewed from the line of sight of the avatar for which the rendering process is working very quickly. This is a stringent requirement as there would be millions of avatars out there.

3. Gap Analysis

When trying to implement these DX services with today's cloud technologies with considering the requirements described in Chapter 2, DX service implementers will face many difficulties in achieving both real-time and cost-effectiveness at the same time.

The major reasons why it's so difficult in today's cloud are as described below.

3.1. Gap 1: Low performance and low scalability of Data Hub services

In today’s typical cloud-based implementation models, data hub services such as Message Broker, Distributed RDB, KVS, and Graph Store, are not fast, e.g., response time jitter tends to exceed 10 milliseconds, and the system size is not so scalable, e.g., the upper limit of data ingestion speed of the message broker tends to be much less than 1 GB per second. More detailed examples are shown below:

- No1. Distributed RDB has less overhead than KVS because it only deals with fixed columns. Therefore, both write and query operations in the Distributed RDB run faster than KVS when configured with the same number of shards and the same consistency level. Due to such characteristics, it will be used for specific purposes such as device location management. However, today’s cloud storage is slow, so read-and-write operations of Distributed RDB that rely on remote storage tend to have jitters as high as 10 milliseconds. Therefore, the real-time requirement for the remote device management in the Smart Factory which is described in Annex A is not met.
- No2. KVS can flexibly manage various data with different structures in one place. As such, it will be used to manage various types of devices and represent avatars performing various actions. However, the performance of KVS in today’s cloud is still not good enough. The jitter of KVS response time, especially for write operations, is large, and it becomes tens of milliseconds occasionally. Also, the scalability of KVS tends to be limited, for example, the read-and-write operations of 1KB data are limited to much less than 1 million times per second. These limitations are primarily caused by inter-node communications for data replication and cluster management over the slow network and its cluster structure which consists of many small nodes designed for scale-out in the cloud. With these limitations, the end-to-end latency requirement of 10 milliseconds or less for the remote device management in the Smart Factory service described in Annex A, and the throughput requirements of the hundreds of millions of read-and-write operations per second for the virtual space management in the Metaverse service described in Annex C will not be satisfied.
- No3. Graph Store can link various data together and analyze them as a whole. Therefore, it is expected to be a foundation for managing complex networks such as the electricity delivery network in the Smart Grid system. However, Graph Store’s response time for exploration queries such as multi-hop neighborhood queries will be more than a few seconds in today’s cloud. This slow response comes from repeated data accesses based on a slow storage system to explore the graph. Therefore, it is difficult to meet latency requirements for the power balance optimization in the Smart Grid which is described in Annex B.
- No4. Message broker accepts large-scale data ingestion with low latency in a stable manner and acts as a common buffered queue that feeds data to various subsequent processes. Therefore, it is often used when collecting large amounts of data. However, Message Broker has an internal delay of more than tens of milliseconds from data ingestion to data retrieval, which is caused by data replication among multiple servers for durability. Therefore, if Message Broker is used, it is not possible to meet the end-to-end response time requirements of 10-millisecond or so, which are required for the remote device management in the Smart Factory and the virtual space management in the Metaverse, as described in Annex A and C respectively.

- No5. Object storage is configured to allow multiple tenants to share storage resources and scale massively in the public cloud. Considering the utilization of open data by many companies, object storage shall be used, because, the block storage is expensive and cannot be shared many servers. However, there is a side effect in using object storage, which makes it difficult to guarantee consistent performance. For example, it is very difficult to guarantee the response time for a system to receive the first byte of requested data and the throughput/bandwidth for continuously extracting large amounts of data. As a result, Object Storage is not currently used for near real-time processing, and when using object storage as a data lake infrastructure for big data, moving data to server clusters for analysis takes time. On the other hand, for use cases like the Smart Grid described in Annex B, the time available for data analysis utilizing open data is very limited. Therefore, it is necessary to devise ways to ensure that data analysis applications can stably retrieve data from object storage-based data lakes at remote sites within the expected required time.
- No6. Compute-and-Storage decoupling is halfway in today's cloud. There is no good solution to build the dynamically scalable shared storage system that provides block access with latency comparable to local storage. For example, access jitter to block storage tends to exceed 10 milliseconds and a single volume cannot be shared by 20 or more nodes. Therefore, when running data-access intensive jobs, many implementation models in today's cloud give up using the remote block storage and rely on the local storage, which impacts availability design and prevents dynamic use of computing power. Also, when running heavy data processing that requires more than dozens of servers against large common data, it is required to split the data and load each data split on each server in advance which results in a lot of inter-node communication for complex query and/or shuffling jobs. Alternatively, it is required to transfer data from shared but not fast storage such as object storage on demand, which slows down the data processing overall. Therefore, real-time analysis against common data is not possible, and also the efficiency of the analytical process is not high in today's cloud. This hinders the realization of intelligent machine failure prediction and intelligent manufacturing control as described in Annex A. Because dozens or hundreds of data processing servers cannot quickly retrieve necessary relevant data from all the data generated throughout the factory and process it in real-time.

Considering the business requirements of DX services described in chapter 2, such low performance is a critical issue. The major reason for limiting the performance of these services is that data service servers and storage servers that make up these services are connected through a slow network as described in sections 4.1, 4.2, 4.3, and 4.4 of the IDH FA Document. In today's cloud, network packets are being forwarded by hopping multiple network devices. Due to the common oversubscription of network links, at each hop, packets may need to wait in packet queues, which causes jitter and latency. Additionally, introducing tag-based priority classes for packet forwarding would further increase the overall latency. As a result, the network becomes slow, e.g., even within the same Availability Zone[*4], the one-way latency becomes over 100 microseconds, and the jitter becomes one millisecond [Cloud Network]. Table 3-1 below summarizes the performance and scalability limits that can be found in today's cloud.

[*4]Note: Availability zones correspond to isolated computing infrastructures up to 100 km apart within a region of cloud service.

Table 3-1: Performance and Scalability Limits in today's cloud

No	Domain	PERFORMANCE AND SCALABILITY LIMITS IN TODAY'S CLOUD	USE CASE GAPS (REQUIREMENTS)
1	Distributed RDB	● Response time jitter:>10 milliseconds	● Smart Factory - Device Management (less than 1 millisecond guaranteed response time for both read and write operations) Note: To manage fixed columns, such as location, etc.
2	KVS	● Response time jitter: > a few tens of milliseconds ● Throughput: < 1 million read-and-write operations per second	● Smart Factory - Device Management (less than 1 millisecond latency and jitter for both read and write operations) ● Metaverse - Avatar Management (guaranteed response time of less than a few milliseconds to update the data, and less than 10 milliseconds for queries) Note: To manage different/arbitrary attributes for each entity.
3	Graph Store	● multi-ho analytical query response time:> a few seconds	● Smart Grid - Electricity Delivery Network Management (less than a few hundred milliseconds response time for multi-hop analytical queries)
4	Message Broker	● Internal delay: > a few tens of milliseconds	● Data collection for Smart Factory, Smart Grid and Metaverse (far below than 10 milliseconds guaranteed delay)
5	Object Storage	● Single Thread GET Bandwidth: 45 - 90 MB/s ● First-byte response time of GET access: 5 milliseconds - up to 1 second Which result in ● Response Time for 1 GB File GET Access: 11 - 23 seconds	● Smart Grid - Power Balance Prediction (Response Time for 1 GB File GET Access: less than 9 seconds)
6	Compute-and-storage decoupling for the Block Storage	● Maximum size of the shared storage: up to 1 PB ● Volume sharing: up to 16 servers	● Smart Factory - Intelligent machine failure prediction and intelligent manufacturing control (5.6 - 56 PB of storage dynamically shared by dozens of servers)

3.2. Gap 2: Lack of realtime-ness and large overhead in Data Hub services

In today's typical cloud-based implementation models, data persistence services such as Distributed RDB, KVS, and Graph Store are not fast and stable enough to fulfill the requirements. This is because, as mentioned above, the cluster is built on the slow network, and also there is an additional variable workload per data item to determine in which partition to store the data, sort the data within the partition, and/or generate indexes. Therefore, as noted in section 4.4 of the IOWN GF IDH FA Document, a simple message broker that focuses on stable data ingestion with relatively lower latency by managing data locally in each server is placed in front of data persistence services, i.e., other data hub services such as KVS. However, the real-time requirements set by IOWN GF are still not met due to the following characteristics of the Message Broker:

- Today's major implementation models of the Message Broker designed for scalability do not support pub-sub and push-delivery features and cause delays due to pull-based data retrieval.
- The current Message Broker supports only sequential reads. No other function such as condition-based record extraction is supported. Therefore, if the consumer application differs depending on the content of the data, the common consumer must retrieve all the data once and pass the data to adequate subsequent processes.
- Many Message Brokers can group ingested data together into partitions by a partition key specified by the data publisher and store the data in the order of ingestion within each partition. With this mechanism, one server, or a server set for data replication, can manage multiple streams. However, as most Message Broker is configured to write data to storage and control the data replication per partition, the Message broker slows down when managing a lot of partitions. Therefore, the number of partitions should be designed to be appropriate, such as 5 - 10 per CPU core depending upon the system configuration and message size, to get an optimal throughput. This means that it is not possible to efficiently manage a large number of low-rate streams. This constraint becomes a problem, if there are many continuous messages to be classified in detail such as management of autonomous manufacturing robots by location for collision avoidance, or if there are a wide variety of ad-hoc messages with different priorities such as exception messages published by the distributed energy storage resources of the Smart Grid.
- Most of today's cloud-based Message Brokers replicate data across multiple servers that span multiple availability zones to guarantee data persistence. Therefore, there is an internal delay of more than tens of milliseconds from data ingestion to data retrieval as described in Section 3-2.

Therefore, the implementation model of the Message Broker has to be improved to support real-time processing with fewer system resources. It is also required to make the persistence level configurable depending on the system requirements, for instance, just to keep the ingested data in memory, to reduce the overhead.

3.3. Gap 3: Centralized only, but not decentralized/federated Data Hub services

In today's cloud, it is assumed that all relevant data is collected in one logical data center consisting of several availability zones, and geo-distributed environments are not well-supported, as described in section 5.2.8 of the IDH FA Document. With such an implementation model, the following issues arise easily:

- End-to-end response time and energy consumption of the DX service are increased because large amounts of data need to be transferred over long-distance networks.
- Many companies are reluctant to upload data related to intellectual property into the cloud as there is the risk of data leakage impacting competitive advantages. To mitigate such concerns and risks, new solutions are required that enable data processing to obtain the desired results while avoiding uploading confidential information, confidential information to be stored temporarily in the cloud only for the minimum

time necessary for processing, and/or a strong encryption mechanism that can be completely controlled only by the user company but cannot be touched by the cloud provider.

- Efforts utilizing distributed data centers cannot progress. In the era of big data, collecting and processing data in one place is nonsense. This is especially noticeable in global organizations. In addition, efforts to utilize data centers in cold regions have begun to reduce power consumption. This initiative has to be accelerated.

To realize various DX Services such as Smart Factory, Smart Grid, and Metaverse, it is required to build the distributed data management so that the real-time data processing is executed in an environment near to its data source and the wide-area data integration is enabled on top of the distributed environments which are connected through a high-speed network.

3.4. Remarks on On-premise Solution

Building and maintaining these DX services on-premises is a challenging task because of its cost and operational burden due to the complexity of technologies. In addition, it is difficult to achieve a good ROI due to spare capacity to take care of a lot of load fluctuations. Therefore, those DX services discussed cannot be supported on-premise, too.

4. IDH PoC Scenarios

In this chapter, a series of IDH PoCs is described that would be the first effort to address the group of gaps described in Chapter 3. PoC Teams may select any scenarios to demonstrate the benefits of the IDH technologies.

4.1. Overview of IDH PoC

As described in section 5.1 of the IDH FA Document, each IDH service assumes the layered architecture consisting of clients, frontend servers, data service servers, and storage as a common concept.

The IDH clients represent the devices in the physical space, such as factory machines, switchgears in the power grid system, head mount displays of metaverse users, or the local systems that manage these devices, and consume the services provided by the IDH solutions. The IDH frontend servers provide not only connection management and load balancing but also intelligent things such as data caching, data preprocessing, preliminary judgment, etc[*5]. The IDH data service servers guarantee data persistence, perform comprehensive data processing, and cooperate with remote IDH data service servers to support global data usage. The IDH storage servers store data for the IDH data service servers and provide additional features such as encryption and data replication.

[*5]Note: In the IDH FA Document, the IDH frontend was described as providing “load balancing”, “query routing”, “service protocol translation”, “message aggregation”, and ”caching”. In this document, the assumption of the IDH frontend is expanded as doing more sophisticated data processing.

Figure 4-1-1 shows an image of the PoC testing environment considering the geo-distributed deployment of IDH services, with five IDH PoC scenarios designed in this document. As described in the figure, it is assumed that a subgroup of frontend servers of an IDH system is placed in the regional edge center to provide short latency services for connected things, and data service servers and storage servers of the same IDH system are placed in the on-premise DC or Central Cloud so that long term data persistence and data utilization are supported.

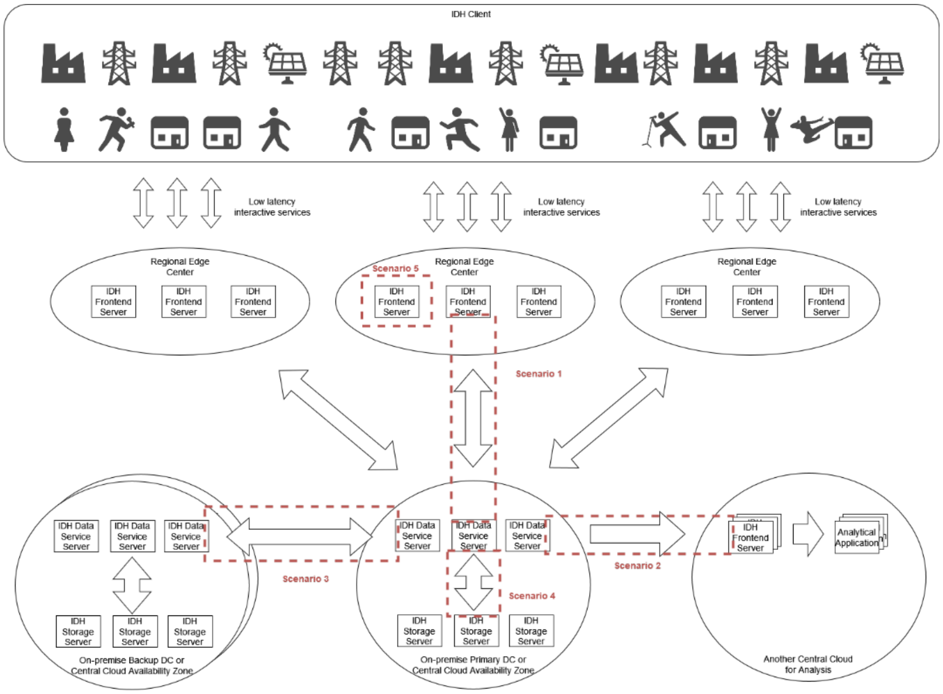

Figure 4-1-1: Geo-distributed IDH adoption model and IDH PoC scenario mapping

With such an assumption, a series of PoC scenarios were determined to ascertain the value of the IOWN solutions against the gaps described in chapter 3. The points of each IDH PoC scenario are described below:

- Scenario 1: Frontend-to-Data Service PUT Communication Acceleration with Open APN (dataflow from the outside into the database)
  This scenario will address Gap 3 by testing how PUT communications from the Frontend servers placed in the regional edge to the Data Service servers placed in on-premise DC or Central Cloud can be accelerated with Open APN.
- Scenario 2: Data Service-to-Frontend GET Communication Acceleration with Open APN (dataflow from the database to the outside)
  This scenario will also address Gap 3 by testing how utilization of data by remote applications can be accelerated through intelligent GET communications from the data service servers to the frontend servers near the analytical applications over Open APN.
- Scenario 3: Data Service-to-Data Service Communication Acceleration with Open APN
  This scenario will address Gap 1 by testing how communication between Data Service servers placed in different DC or Availability Zone of Cloud can be accelerated with Open APN.
- Scenario 4: Elastic High-speed Shareable Storage with DCI
  This scenario will also address Gap 1 by testing how an elastic high-speed shareable storage system can be established with DCI.
- Scenario 5: New frontend implementation supporting low-latency responses and efficient geo-distributed processing
  This scenario is to test a new enhanced IDH frontend that provides various data processing capabilities on in-memory queues, which leads to real-time responses and geo-distributed data processing that reduces data transfer load.

Among these, Scenario 5 is a particularly important idea to address Gap 2, an additional delay caused by a Message Broker being placed in front of other data services such as KVS. So, the new IDH frontend will be designed to eliminate this delay and provide a faster response by combining required features together. Also, running the IDH frontend at the regional edge center eliminates network latency issues. Therefore, the new IDH frontend is an essential element for real-time services. The new IDH frontend also contributes to fulfilling Gap 3. Data transfer volume can be reduced by pre-processing data, such as compression and filtering, at IDH frontends located at regional edge centers before sending the data to the central cloud. To build such a frontend, the following additional considerations are required:

- In-memory-queue data processing
  In order to achieve real-time processing in less than a few milliseconds, it will be necessary to process data directly in in-memory-based queues. Considering typical data processing such as time-series aggregation and complex event processing, it means that features like queries in the database need to be implemented on top of the in-memory-based queue.
- Configurable data persistence
  The data persistence options shall be configurable in the IDH frontend server. It means that 1) the IDH frontend can be configured to tolerate data loss, i.e., one copy of data is just kept in memory, 2) data can be replicated to the other IDH frontend servers in a synchronous or asynchronous manner, 3) data can be preserved in the persistent memory of the IDH frontend server, and/or 4) log data can be generated and written to storage synchronously.
- Dynamic and detailed resource and process controls
  Various processes such as data ingestion, queue management, query, data export, etc. are running in the IDH frontend in parallel. Therefore, dynamic and detailed resource management is required for stable system operation. Process scheduling is also important, such as deferring low-priority processes when the workload is high.
- Scaling mechanism
  The IDH frontend must be able to scale dynamically in case of load fluctuations. Either or both scale-up/down and/or scale-out/in operations must be supported. To scale up/down, the allocation of resources such as CPU and memory must be dynamically changed online. To scale out/in, not only adding/removing frontend servers in the IDH cluster but also moving or sharing the data among the frontend servers may be required.

Figure 4-1-2 shows such a conceptual activity, the new implementation model, and the mapping of IDH PoC scenarios on top of it.

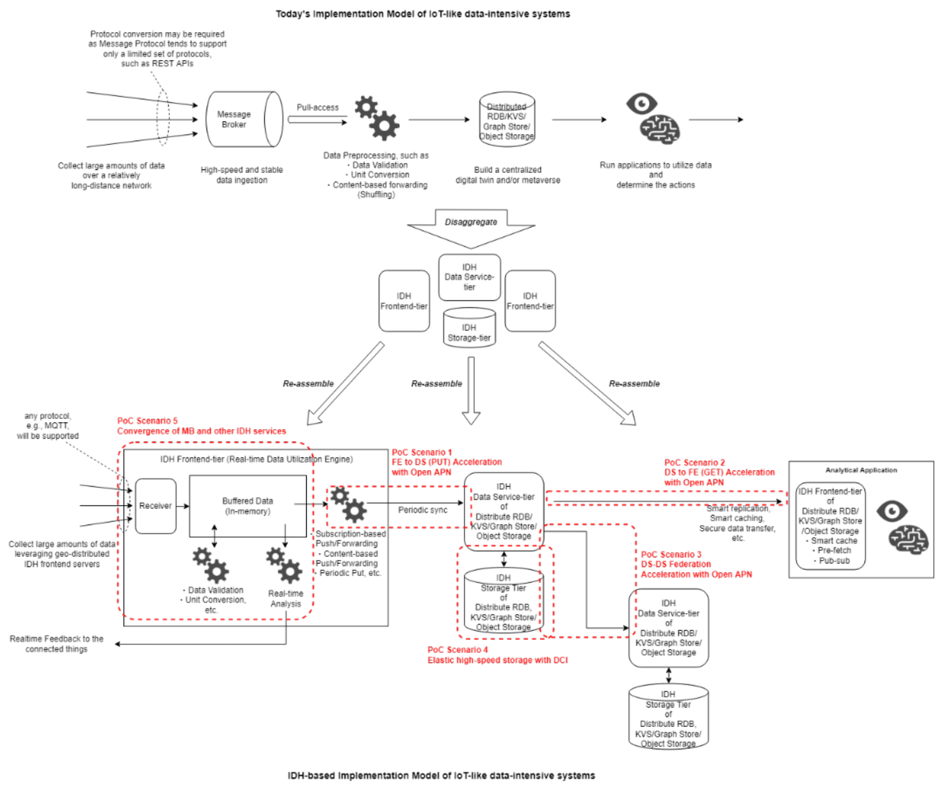

Figure 4-1-2: Devising a new implementation model and PoC scenario mapping

4.2. Scenario 1: Frontend-to-Data Service PUT Communication Acceleration with Open APN

As described in Chapter 3, centralized data processing by one data center hinders the realization of various DX services. In order to overcome this, the Data Hub services have to support geo-distributed deployment and accelerate geo-distributed data processing/usage/management. In this PoC, how the PUT communications from the Data Hub frontend server in the regional edge center to the Data Hub data service server in the central cloud can be accelerated with Open APN will be tested.

Description

As described, the PoC environment assumes that Data Hub frontend servers are placed in the regional edge center to provide real-time services and preprocess the data before it is transferred to the IDH data service servers, and Data Hub data service servers are placed in the central cloud or on-premises data center to guarantee data persistence and support the comprehensive data usage. Based on such assumptions, this PoC scenario tests how PUT Communication from the IDH frontend server to the IDH data service server can be accelerated by Open APN.

Regarding a test data set and a workload profile to be used, it is a choice of the PoC Team. For instance, some MQTT or JSON message generators can be used.

To evaluate the benefit of IDH, at least two test environments shall be created, one represents the IDH-based implementation model, and the other represents today's implementation model, and be benchmarked by running the same workloads in both environments.

Figure 4-2-1 shows the configuration of PoC scenario 1.

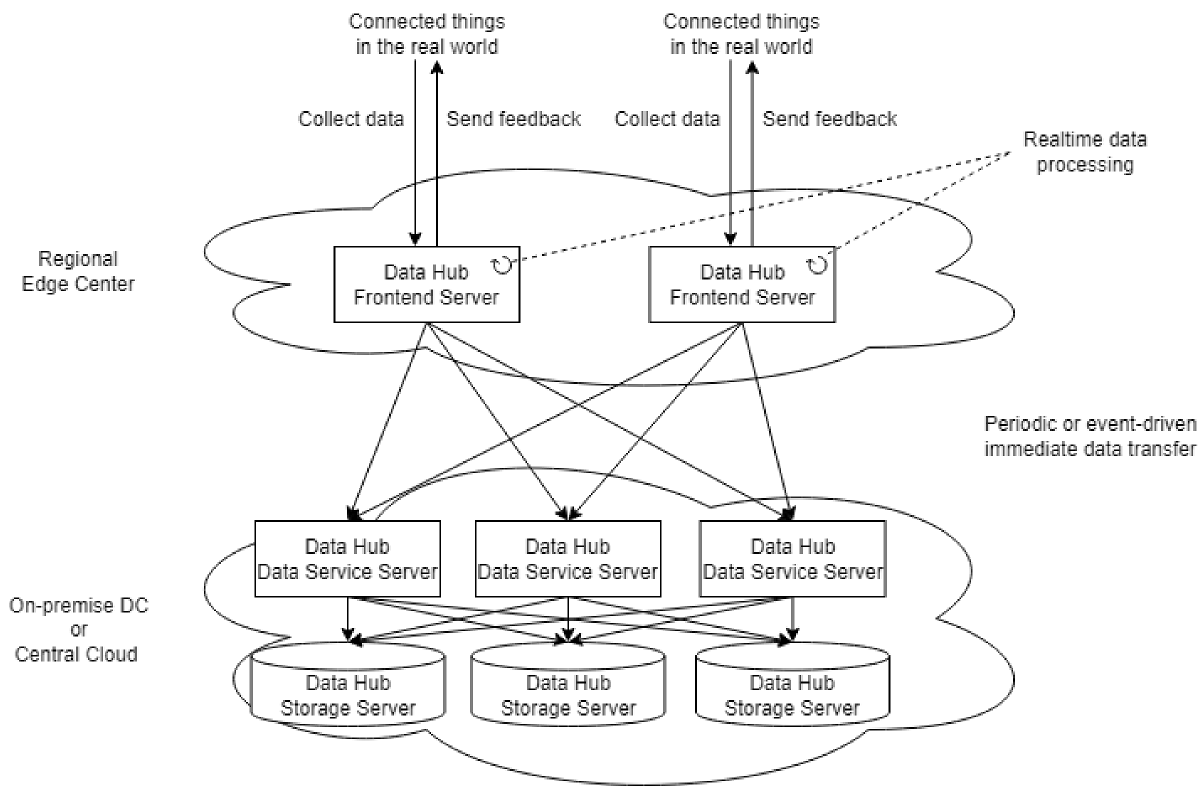

Figure 4-2-3: Configuration of PoC Scenario 1

Selected Features

In this sub-section, the features that shall be implemented and tested during PoC, are described. Depending upon the PoC context, some features will be only implemented in the IDH-based implementation model, but not in today's implementation model as noted in the bracket.

- PUT Communications over Open APN (IDH-based test environment only)
  PUT communications from the Data Hub frontend server placed in the regional edge center to the Data Hub data service server placed in the central cloud or on-premise data center shall be conducted over Open APN.
- Data preprocessing before data transfer (IDH-based test environment only)
  When data is transferred from the IDH frontend servers to the IDH data service servers, compression, or if allowed and adequate, aggregation, and filtering, shall be applied to reduce the network traffic.

Optional Features

This sub-section describes the features that can be implemented and tested to further investigate the feasibility, better adoption, and/or benefit of IOWN GF technology.

- Selection of the Data Hub data service server
  When pushing the data to Data Hub data service servers, each Data Hub frontend server can consider those assigned scopes of data management and/or workload status, to determine which server to push the data to.
- Usage of advanced protocols such as RDMA
  To further accelerate PUT communications, advanced protocols such as RDMA can be used.
- SmartNIC, DPU, or IPU usages

PUT communications can be accelerated by SmartNIC, DPU, or IPU, for example, by compressing the data by those accelerators.

Variable Conditions

In this sub-section, the conditions that must be varied to understand the characteristics of tested features and to find an optimal configuration, are described.

- Effect of the network type, latency, jitter, bandwidth, packet loss, and packet order change
  For example, to simulate today's network communication between the regional edge center 250km apart from the central cloud, it would be adequate to set the average one-way latency to 2 milliseconds, jitter to 1 - 1.3 milliseconds, and effective bandwidth variation to 40 - 80 Gbps even assuming 100 Gbps dedicated connection is provisioned, etc. In addition, the packet loss rate and presence/absence of packet order change shall also be variable, as these affect communication performance and hinder the adoption of RDMA protocol.
- Network Protocol parameters
  Depending upon the protocol used, such as TCP, UDP, and RDMA, relevant parameters, such as frame size, window size, etc. in the case of TCP, shall be varied.
- Data object size and presence/absence of data compression
  The size of data objects to be transferred also affects performance. Considering various DX service scenarios, various data object sizes, such as 1 KB, 100 KB, 10 MB, and 1 GB shall be tested. Data can be compressed before transfer. If so, use the size after compression as a variable parameter. This is because the impact of network performance can be evaluated more accurately with that parameter. However, it should also be noted that when compressing small data, the compression ratio is not high.

Expected Benchmark

During benchmarking, the PoC Team shall set up two environments, one based on today's implementation model, and the other based on the IDH implementation model as explained, implement selected/desired features, vary the conditions, and measure the following KPIs:

- Throughput per unit system resource
  Measure the throughput, i.e., how much data can be transferred from the frontend server in the regional edge center to the data service server in the central cloud. The unit system resource is a PoC Team's choice, which could be something like a 16-vCPU server equipped with 100 Gbps Open APN connectivity.
- Energy consumption per unit system resource
  Measure the energy consumed by the unit system resource selected by the PoC Team during a steady-state operation.

4.3. Scenario 2: Data Service-to-Frontend GET Communication Acceleration with Open APN

Similar to scenario 1, this PoC is conducted on top of the geo-distributed deployment of the Data Hub based on the configuration shown in Figure 4-3-1. With such an assumption, it is tested that how the GET communications requested by the Data Hub frontend server in another central cloud for analysis to retrieve data from the Data Hub data service server in the on-premise DC or in the central cloud can be accelerated with Open APN and advanced data transfer mechanisms.

Description

This PoC tests how GET communication to transfer the required data from the remote IDH data service server to the IDH frontend server can be accelerated by Open APN. It is valuable to accelerate GET communication since a lot of computing resources are needed only when doing analytical processing.

However, required analysis data may contain a lot of confidential information. For instance, in the Smart Factory case, the manufacturing control parameters contained in the machine log are considered confidential data. Therefore, it may not be acceptable to transfer such confidential data to the cloud as it is for many manufacturing companies. In addition, log data may contain many irrelevant data, so it will not be cost-effective to transfer all data to the cloud, run a full scan against the data and train the models with filtered data, especially when only a small portion of the transferred data is used.

This means that a new solution that can quickly extract only the necessary data from the big data store, e.g., Object Storage, convert the extracted data so that all confidential information is hidden, and transfer it to the cloud securely at high speed, needs to be developed.

In addition, such a mechanism has a positive effect on the reduction of energy consumption. This is because it will be possible to more actively utilize data centers in cold regions where power consumption can be reduced.

Therefore, in this PoC, not only GET communication acceleration by Open APN, but also advanced data transferring mechanisms will be tested.

Regarding a test data set and a workload profile to be used, it is a choice of the PoC Team. For instance, the Parquet files that contain some log data can be used.

To evaluate the benefit of IDH, at least two test environments shall be created, one represents the IDH-based implementation model, and the other represents today's implementation model, and be benchmarked by running the same workloads in both environments.

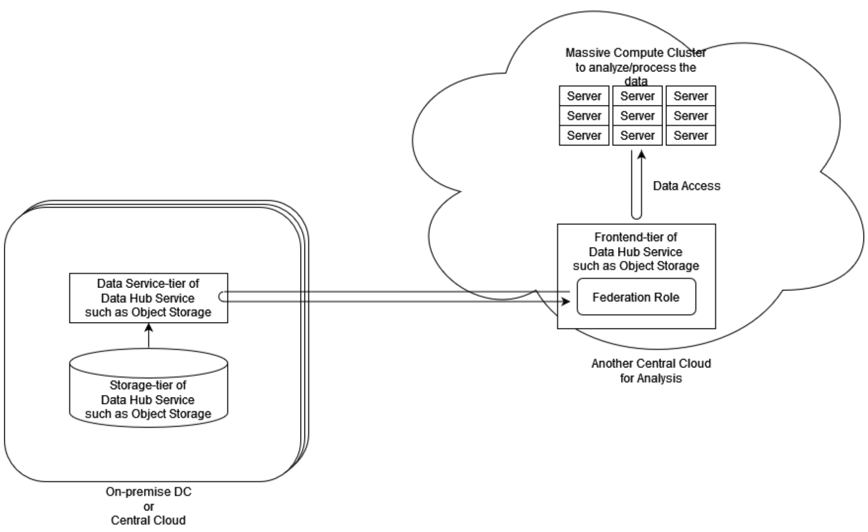

Figure 4-3-4: Configuration of PoC Scenario 2

Selected Features

In this sub-section, the features that shall be implemented and tested during PoC, are described. Depending upon the PoC context, some features will be only implemented in the IDH-based implementation model, but not in today's implementation model as noted in the bracket.

- Federation function in Cloud DC (IDH-based test environment only)
  To support the geo-distributed data usages, a federation function, which works with the IDH Data Service servers in the on-premise DC to retrieve the data, shall be implemented in the IDH frontend servers in the cloud DC.
- Get Communications over Open APN (IDH-based test environment only)

Get communications from the Data Hub frontend server placed in the remote cloud to the other Data Hub frontend placed in the central cloud or on-premise data center shall be conducted over Open APN.

- Data filtering and preprocessing in the on-premise DC (IDH-based test environment only)
  It makes no sense to transfer irrelevant data through a federation. Instead, it is desirable to narrow down the data and recompress it on the sender side. Such a mechanism can be implemented.

Optional Features

This sub-section describes the features that can be implemented and tested to further investigate the feasibility, better adoption, and/or benefit of IOWN GF technology.

- Intelligent prefetch
  The analysis process can be speeded up by prefetching the data that is likely to be used in advance. Such a mechanism can be implemented.
- Data conversion to guarantee the confidentiality of data
  By removing confidential information before sending data, you will be able to use the cloud with greater peace of mind. Such a mechanism can be implemented.
- Clustered deployment of the IDH system over IOWN GF DCI
  As described in chapter 5 of the IDH FA Document, any of IDH services itself is a distributed system. Therefore the test IDH system can be built internally on top of IOWN GF DCI. Such an implementation model can be tested.
- SmartNIC, DPU, or IPU usages
  Data filtering and preprocessing described in the Selected Features sub-section can be accelerated by SmartNIC, DPU, or IPU.

Variable Conditions

In this sub-section, the conditions that must be varied to understand the characteristics of tested features and to find an optimal configuration, are described.

- Effect of the network type, latency, jitter, and bandwidth
  If the PoC Team cannot prepare a test network equivalent to the production one, then it is required to simulate the production network by varying latency, jitter, and bandwidth. For example, to simulate today's network communication between Cloud DC and on-premise DC, it would be adequate to set the average one-way latency to 3 - 10 milliseconds, jitter to 5 - 10 milliseconds, and effective bandwidth variation to 40 - 80 Gbps even when a 100 Gbps dedicated connection is provisioned, etc.
- Network Protocol parameters
  Depending upon the protocol used, such as TCP, UDP, and RDMA, relevant parameters, such as frame size, window size, etc. in the case of TCP, shall be varied.
- Data object size
  The size of data objects to be transferred also affects performance. Considering various DX service scenarios, various data object sizes, such as 10 MB, 100 MB, 1 GB, and 10 GB, shall be tested. It should be noted that, in the case of the Object Storage service, a typical cloud vendor recommends making the object a few GB in size for performance, as described in their whitepaper [Cloud Analysis and Object Storage].
- Ratio of relevant data
  Each data object may contain a lot of irrelevant data for analysis, and only some portion of data will need to be analyzed. This implies that it will be a good idea to apply some filters on the data source side, although there is a tradeoff between filtering accuracy and required computing resources. Therefore, to evaluate the impact on end-to-end latency, these conditions shall be changed during PoC so that the percentage of data actually used out of transmitted data varies between a few percent to 100 percent.

Expected Benchmark

During benchmarking, the PoC Team shall set up two environments, one based on today's implementation model, and the other based on the IDH implementation model as explained, implement selected/desired features, vary the conditions, and measure the following KPIs:

- Time for required data transfer
  Measure the time consumed for transferring the required data.

4.4. Scenario 3: Data Service-to-Data Service Communication Acceleration with Open APN

As described in Chapter 3, today's Data Hub services are not fast in today's cloud, because data is replicated across multiple data service servers through a slow network. This IDH PoC is to see how a high-speed network improves the performance of these services.

Description

This PoC tests how communication between IDH data service servers can be accelerated by Open APN, and how it improves IDH data service server performance.

In today's cloud, the data service servers of these Data Hub services tend to be distributed across multiple availability zones that are separated by a certain distance, which could be a few km or up to 100 km, and interconnected through a legacy network, which could generate a large jitter of 1 millisecond or more and large bandwidth instability as the same connection of 25 or 100 Gbps is shared by multiple servers. In such a configuration, data is replicated in a synchronous manner across multiple Data Hub servers, to guarantee data persistence and distribute the data access workloads. Thus, the data hub services cannot run fast and their performance becomes unstable. In this PoC, the inter-server network is replaced with Open APN, so that the latency connecting servers becomes much shorter, the jitter becomes almost zero, and the bandwidth becomes larger, e.g., 100 Gbps or so, and two configurations, one based on Ethernet and the other based on Open APN will be compared.

Regarding a test data set and a workload profile to be used, it is a choice of the PoC Team. However, common benchmark tools, such as the TPC benchmark series, Swingbench, and/or YCSB, can be used. The important thing is to measure how Open APN improves performance, cost, and power consumption.

Figure 4-4-1 shows the configuration of PoC scenario 3.

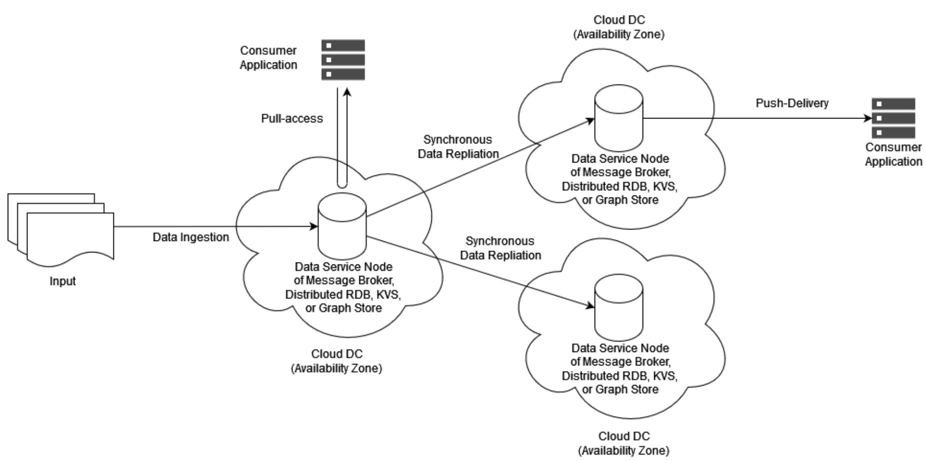

Figure 4-4-5: Configuration of PoC Scenario 3

Selected Features

In this sub-section, the features that shall be implemented and tested during PoC, are described. Depending upon the PoC context, some features will be only implemented in the IDH-based implementation model, but not in today's implementation model as noted in the bracket.

- Inter-server communications over Open APN (IDH-based test environment only)
  Data service servers of selected IDH Service, i.e., Message Broker, Distributed RDB, or KVS, shall be connected through Open APN. Regarding the configuration of Open APN, at least, switches correspond to a DCI gateway that ensures the flexibility of communication path establishment shall be placed, and communication between them shall be performed by wavelength-based paths. However, communication between those switches and the Data Hub Servers may be based on Ethernet.
- Continuous data ingestion and data usage (Both test environments)
  Continuously ingestion and access of data shall be supported.
- Pull-access of data (Both test environments)
  Query-based pull-data access shall be supported.
- Push-delivery of data (IDH-based test environment only)
  Push-data delivery based on predefined conditions shall be supported.
- Synchronous data replication across multiple data service servers of Message Broker, Distributed RDB, and KVS (Both test environments)
  Data shall be synchronously replicated across multiple data service servers.
- Scalability
  Considering the Smart Factory service that collects and processes data from 10,000 machines/robots in a 10-millisecond cycle, the distributed RDB or KVS must be able to support data updates and reads at a speed of more than 2 million records per second. Considering Smart Grid, the Graph Store must have the scalability to support data updates at the speed of more than 10,000 records per second and run complex network queries more than every second. Considering Metaverse service where one million people interact together at a speed of 30 FPS, the message broker needs to be scaled up to handle up to 33 million message ingestions per second and corresponding continuous reads.

Optional Features

This sub-section describes the features that can be implemented and tested to further investigate the feasibility, better adoption, and/or benefit of IOWN GF technology. (Please note that these desired features need to be implemented in both test environments for equitable comparison of IDH PoC and today's cloud solution.)

- Usage of advanced protocols such as RDMA
  To further accelerate inter-data service server communication, advanced protocols such as RDMA can be used.
- SmartNIC offloading
  To further accelerate the inter-data service server communication, processes for TCP connection management, encryption, etc., can be offloaded to SmartNIC.

Variable Conditions

In this sub-section, the conditions that must be varied to understand the characteristics of tested features and to find an optimal configuration, are described.

- Effect of the network type, latency, jitter, and bandwidth
  During PoC, it is required to simulate the physical implementation of the network by varying latency, jitter, and bandwidth. For example, to simulate today's inter-AZ communication in an IDH cluster, it would be adequate to set the average one-way latency to 0.5 millisecond, jitter to 0.5 millisecond, and bandwidth variation to 15 - 20 Gbps, etc. There are various papers for more detailed performance information on today's cloud networks, such as the University of Michigan's one [Cloud Network]. Please refer to such papers to determine the right parameters.

- Effect of the transaction size
  Multiple data records can be updated together to increase efficiency. This is called the transaction size. During the PoC, various transaction sizes, such as 1, 10, 100, and 1,000, shall be tested.
- Effect of the data record size
  Data record size also affects performance. Considering various DX service scenarios, various data record sizes, such as 100 B, 1 KB, 10 KB, 100 KB, and 1 MB, shall be tested.

Expected Benchmark

During benchmarking, the PoC Team shall set up two environments, one based on today's implementation model, and the other based on the IDH implementation model as explained, implement selected/desired features, vary the conditions, and measure the following KPIs:

- Data Hub response time and jitter
  Measure the response time of the Data Hub service and its jitter. It should be noted that if these measurements change depending upon the system scale, please report it, too.
- Throughput per unit system resource
  Measure the throughput, e.g., read and write transactions per second, achieved with a unit system resource. The unit system resource is a PoC Team's choice, which could be something like a 16-vCPU server equipped with 100 Gbps Open APN connectivity.
- Energy consumption per unit system resource
  Measure the energy consumed by the unit system resource selected by the PoC Team during a steady-state operation.
- Scaling Factor
  Measure how linearly the system scales and the rate of performance improvement when adding the system resources.

4.5. Scenario 4: Elastic high-speed shareable storage with DCI

As described in Chapter 3, the scalability of data hub services is limited in today's typical cloud due to the lack of an elastic high-speed shareable storage system. This PoC is to test how such an elastic high-speed shareable storage can be established.

Description

In today's cloud, there is no block storage that can simultaneously provide block access with sub-millisecond latency and jitter, a total capacity of more than 1 PB, and a multi-attach capability to let it be shared by dozens of servers. This is due to a lack of flexible control mechanisms, good network bandwidth and latency, and sufficient storage-side accelerators. In order to realize DX services, it is necessary to realize an elastic shareable remote storage system with performance close to local storage.

In this PoC, it is tested that how the DCI architecture and technologies can be used to build such a storage solution and evaluate its impact on Data Hub services. For example, it is tested that how the storage system can be configured dynamically with DCI, and how the storage performance can be accelerated by running the iSCSI protocol (or other protocols if applicable) over Open APN provisioned by DCI. It shall be noted that FCoE provides better performance than iSCSI but requires a lossless network. Open APN technology could support FCoE, because of its high quality.

Data replication should also be considered to guarantee date persistence, as is done in the cloud. This data replication process might be controlled by the data service server or by the storage server. In addition, if it is required to build a pool with more storage resources for the data service servers, a gateway will be used in front of the storage servers. By comparing these different topologies, their impact on latency and scalability can be understood.

In summary, this PoC will seek a way in building the required storage system for future DX services. with utilizing IOWN GF DCI technologies. There might be more than option one depending on the requirements of Data Hub Services.

Regarding a test data set and a workload profile to be used, it is a choice of the PoC Team. However, common benchmark tools, such as the SPC (Storage Performance Council) benchmark series, can be used.

Figure 4-5-1 shows the possible configuration options to be tested in PoC scenario 4. It should be noted that the number of connections from one IDH Data Service Server (or Storage Gateway Server) to IDH Storage Servers shall be generally up to 10 - 15 if connection overhead is considered. On the other hand, it may be desirable to share PB-class data across dozens of IDH Data Service Servers. Because, with that configuration, PB-class data can be processed and analyzed more rapidly. An ideal storage service should be designed with these requirements in mind.

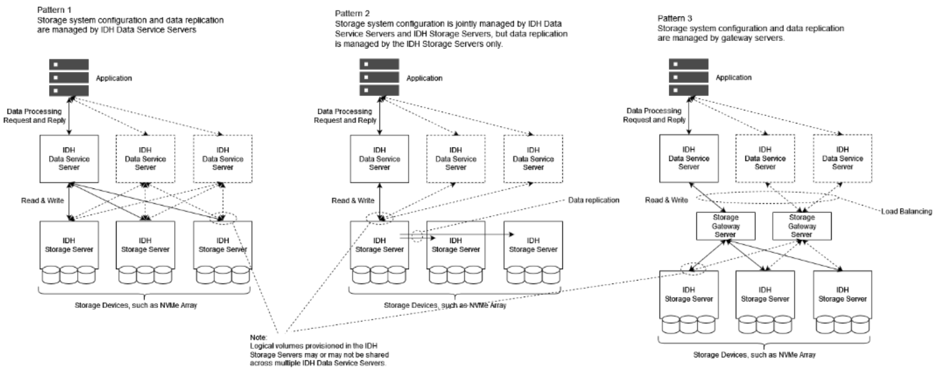

Figure 4-5-6: Configurations of PoC Scenario 4

Selected Features

In this sub-section, the features that shall be implemented and tested during PoC, are described. Depending upon the PoC context, some features will be only implemented in the IDH-based implementation model, but not in today's implementation model as noted in the bracket.

- Establishment of communication paths between data service servers and storage servers, which may go through the gateway server, made by DCI (IDH-based test environment only).
  A prototype of DCI technology for the remote storage system shall be prepared and communication paths between data service servers and storage servers have to be provisioned on top of it. To guarantee the scalability of the system, a storage resource pool shall be built on top of multiple DCI clusters, and a data service server that has access to the storage service provisioned from the storage resource pool shall be put on a different DCI cluster. However, data replication may or may not be performed within the same DCI cluster, as it is a design choice of the IDH solution provider.
- Usage of standard protocols such as iSCSI (Both test environments)
  To enable the remote storage to be attached to any hosts, well-known protocols such as iSCSI shall be implemented.
- Continuous and simultaneous writing and reading of data (Both test environments)
  Continuous and simultaneous writing and reading of the data shall be supported.
- Shared storage
  A capability to support multi-attach of a single storage volume exposed at the data storage server to multiple data service servers shall be supported.
- Scalability

The storage system of each IDH service cluster has to scale well to support PB-class shareable storage system.

Optional Features

This sub-section describes the features that can be implemented and tested to further investigate the feasibility, better adoption, and/or benefit of IOWN GF technology.

- Connection over Open APN
  Open APN can be used to connect the data service servers and the storage servers
- Attachment type control
  When attaching the storage to the data service server, an attachment type, e.g., read-write, read-only, etc., can be designed and/or updated online through the control plane functions.
- Usage of advanced protocols such as iSCSI over RDMA
  To further accelerate remote storage access, advanced protocols such as iSCSI over RDMA can be used.
- Sequential read optimization
  Considering the data warehouse use case, any considerable techniques that can optimize the sequential read can be used.

Variable Conditions

In this sub-section, the conditions that must be varied to understand the characteristics of tested features and to find an optimal configuration, are described.

- Effect of the network type, latency, jitter, and bandwidth
  During PoC, it is required to simulate today's cloud DC network by varying latency, network, and bandwidth. For example, it would be adequate to set these to match the actual performance of today's cloud, for instance, set the average one-way latency to 150 microseconds, jitter to 1 millisecond, and bandwidth variation to 15 - 20 Gbps, etc. For more detailed performance information, please refer to the University of Michigan's paper [Cloud Network].
- Effect of the data block size
  Data block size also affects performance. Considering various DX service scenarios, various data record sizes, such as 4 KiB, and 16 KiB, shall be tested.
- Random/Sequential access
  Data access type, i.e., random or sequential, also affects performance. Both access types shall be tested.
- Ratio of Read/Write operations
  Ratio of read-operations and write-operations also affects performance. Various Read/Write ratios such as 100:0, 80:20, 50:50, etc. shall be tested.
- Storage system topology
  As discussed, different topologies shall be taken depending on response speed and scalability requirements. During PoC, three different topologies described in Figure 4-2 have to be tested to know these characteristics.

Expected Benchmark

During benchmarking, the PoC Team shall set up up to 6 storage systems with different topologies as described in Figure 4-6, and with different network types, i.e., today's Ethernet and Open APN, and implement the selected/desired features there. Then the team shall vary the conditions and measure the following KPIs:

- Storage System Response Time
  Run one or several storage benchmark tools, such as the SPC (Storage Performance Council) benchmark series, and measure the response time of the storage access and its jitter. It should be noted that if these measurements change depending upon the system scale, please report it, too.

- Throughput per unit system resource
  Measure the throughput, i.e., IOPS and bandwidth, achieved with a unit system resource. The unit system resource is a PoC Team's choice, which could be something like a 16-vCPU server equipped with 100 Gbps Open APN connectivity.
- Energy consumption per unit system resource
  Measure the energy consumed by the unit system resource selected by the PoC Team during a steady-state operation.
- Scaling Factor
  Measure how linearly the system scales and the rate of performance improvement when adding the system resources.
- Data Hub performance
  Measure how performance, such as response time, jitter, and throughput, of the Data Hub service will change depending upon the differences in storage system configuration.

4.6. Scenario 5: New frontend implementation supporting low-latency responses and efficient geo-distributed processing

As described in Chapter 3, in today's IoT systems, a Message Broker is placed in front of Distributed RDB, KVS, and Graph Stores, which increases processing latency and worsens system cost and energy consumption. That is not good. To achieve real-time services with low cost and low energy consumption, a new technology that supports massive data ingestion and in-place data processing for real-time responses and efficient data transfer must be developed. This PoC is to see how such a new implementation can be made, and how the system performance can be improved with it.

Description

In this PoC, the new approach to building an IoT system that leverages the geo-distributed regional edge centers will be evaluated so that real-time services assumed by IOWN GF can be realized.

Traditionally, in order to receive and process a large amount of data in a stable and scalable manner, the Message Broker is placed in front of the other Data Hub service such as KVS, data is first ingested to the Message Broker, then pulled and preprocessed by some application running on a different server to restore the data in the target Data Hub service. Then, data is finally analyzed, processed, and used there.

However, real-time processing was not possible with such a legacy architecture. This is because essentially the same data is repeatedly transferred and stored. Furthermore, it requires a pull-access that runs every second or so. Therefore, real-time processing of sub-seconds or less could not be realized at all. In addition, it required higher system costs and higher energy consumption due to repetitive data storage and transfer.

In this PoC, a new implementation model that is designed to reduce the unnecessary transfer and storage of data, while keeping the system operating in a stable and scalable manner, will be tested.

Regarding a test data set and a workload profile to be used, it is a choice of the PoC Team. Of course, IoT-specific benchmark tools can be used.

To evaluate the benefit of IDH, at least two test environments shall be created, one represents the IDH-based implementation model, and the other represents today's implementation model, and be benchmarked by running the same workloads in both environments.

Figure 4-6-1 shows the configurations of PoC scenario 5.

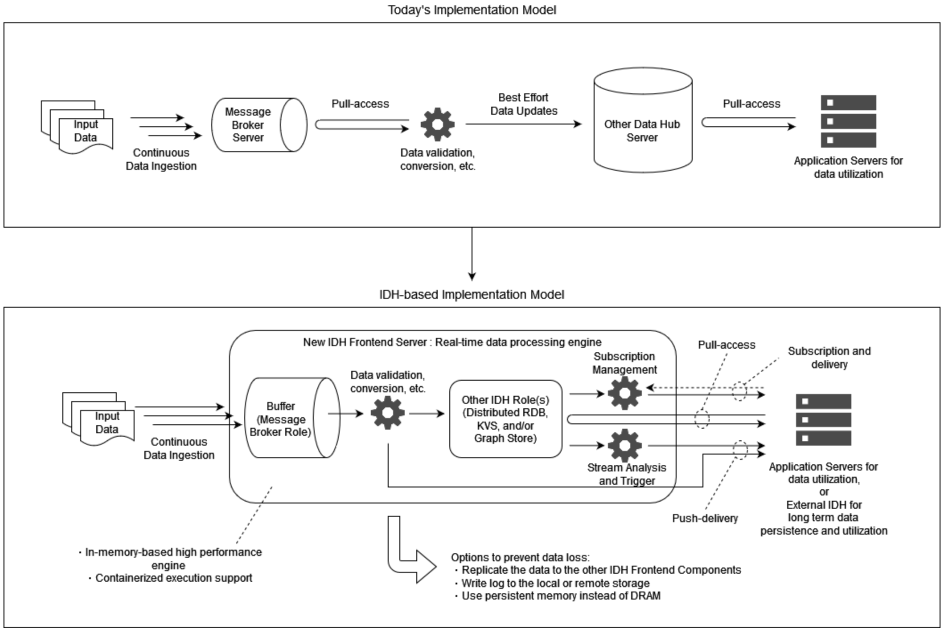

Figure 4-6-7: Configuration of PoC Scenario 5

Selected Features

In this sub-section, the features that shall be implemented and tested during PoC, are described. Depending upon the PoC context, some features will be only implemented in the IDH-based implementation model, but not in today's implementation model as noted in the bracket.

- Data Ingestion through Open APN (IDH-based test environment only)
  Data ingestion to this new IDH frontend server shall be conducted through Open APN.
- Convergence of Message Broker and other Data Hub services (Both test environments)
  To avoid unnecessary movement and storage of data, Message Broker and other Data Hub services shall co-exist in the same server, as described in section 5.2.6 of IDH FA Document.
- In-memory-based data processing
  Within the new IDH frontend server, the Message Broker service has to temporarily buffer large amounts of data for high-speed data ingestion, data validation, and data preprocessing. However, to write such buffered data to the storage generates a large overhead. Therefore, the new IDH frontend server must be able to be configured to store buffered data only in DRAM.
- Memory space sharing across Message Broker (buffering) role and other Data Hub roles (Both test environments)
  As the Message Broker and other Data Hub services co-exist in the same server, the communication between the service processes of these services shall be accelerated through memory space sharing.
- Continuous data ingestion and subsequent data processing (Both test environments)
  Continuous data ingestion and subsequent data processing shall be supported. Subsequent data processing includes data conversion, data judgment, data delivery to subscribers, etc.
- Resource controls (Both test environments)
  As the Message Broker and other Data Hub services co-exist in the same server, strict resource controls shall be implemented, for the stability of the system.
- Scalability

Considering Metaverse service, where one million people interact together and send 30 FPS animation data for their avatars at a bit rate of about 15 Mbps each, the system has to be able to scale up well to support massive data ingestion at a bit rate of about 15 Tbps and at a speed of 33 million messages per second. It should be noted that tens of geo-distributed regional edge centers can be used to distribute the data and the workload.

Optional Features

This sub-section describes the features that can be implemented and tested to further investigate the feasibility, better adoption, and/or benefit of IOWN GF technology.

- Message Broker and the other IDH service convergence on top of disaggregated computing architecture
  As the Message Broker and the other Data Hub services co-exist in the same server, the resource allocation becomes very severe. The IOWN GF disaggregated computing architecture can be used to provision a server flexibly with the required resources.
- Retention period of the buffered data
  The buffered data needs to be managed just temporarily. So once it is processed, or after a certain period of time, it can be deleted. Such a mechanism can be implemented and tested.
- Persistent Memory adoption
  Buffered data is kept in DRAM within the new IDH frontend server in the standard configuration. However, to better protect data and speed up the reboot process, data can be stored in persistent memory.
- Configurable data persistence level
  To protect data from data loss, several optional features can be added. For instance, data can be synchronized across multiple IDH frontend servers, change logs can be written to the local or remote storage, and/or data can be replicated to the other IDH service in the central cloud periodically.
- Clustered convergence over Open APN
  These new IDH frontend servers can be clustered for data persistence and multi-server data processing. This means that the IDH frontend server will be similar to the IDH Data Service Server. In addition, the communication between these servers can be accelerated by RDMA over Open APN.

Variable Conditions

In this sub-section, the conditions that must be varied to understand the characteristics of tested features and to find an optimal configuration, are described.

- Effect of the network type, latency, jitter, and bandwidth
  During PoC, it is required to simulate the physical implementation of the network by varying latency, jitter, and bandwidth. For example, to simulate today's network communication between the Message Broker servers and the other Data Hub servers, it would be adequate to set the average one-way latency to 150 microseconds, jitter to 0.5 millisecond, and bandwidth variation to 15 - 20 Gbps, etc. There are various papers for more detailed performance information on today's cloud networks, such as the University of Michigan's one [Cloud Network]. Please refer to such papers to determine the right parameters.
- Effect of the transaction size
  Multiple data records can be updated together to increase efficiency. This is called the transaction size. During the PoC, various transaction sizes, such as 1, 10, 100, and 1,000, shall be tested.
- Effect of the data record size
  Data record size also affects performance. Considering various DX service scenarios, various data record sizes, such as 100 B, 1 KB, 10 KB, 100 KB, and 1 MB, shall be tested.

Expected Benchmark

During benchmarking, the PoC Team shall set up two environments, one based on today's implementation model, and the other based on the IDH implementation model as explained, implement selected/desired features, vary the conditions, and measure the following KPIs:

- End-to-end response time and jitter
  Measures the latency and its jitter from the time of data ingestion into Message Broker (role) to the time of data acquisition in the application server.
- Throughput per unit system resource
  Measure the throughput, i.e., message ingestion and acquisition speed, achieved with a unit system resource. The unit system resource is a PoC Team's choice, which could be something like a 16-vCPU server equipped with 100 Gbps Open APN connectivity.
- Energy consumption per unit system resource
  Measure the energy consumed by the unit system resource selected by the PoC Team during a steady-state operation.

5. Other Considerations

The PoC Reports should include considerations regarding the following items:

- Qualitative and quantitative analysis of IOWN technologies by comparing IOWN technologies with existing technologies.
- Insights on the feasibility of the three DX services.

6. Summary

This paper describes the initial set of IDH PoC scenarios, which are determined by considering Smart Factory, Smart Grid, and Metaverse services, identifying the technology gaps, and devising IDH solutions to fill those gaps. It is hoped that the realization of future DX services will be accelerated by executing these PoCs. Lastly, we would like to ask any interested companies to be involved in this initiative and accelerate technological development jointly.

References

[IOWN GF CPS UC]: IOWN Global Forum Cyber-Physical System Use Case Document; https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-CPS_Use_Case_1.0.pdf

[IOWN GF AIC UC]: IOWN Global Forum AI-Integrated Communications Use Case Document; https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-AIC_Use_Case_1.0.pdf

[IDH FA Doc]: IOWN Global Forum Data Hub Functional Architecture; https://iowngf.org/wp-content/uploads/formidable

[APN FA Doc]: IOWN Global Forum Open All-Photonic Network Functional Architecture; https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-Open-APN-Functional-Architecture-1.0-1.pdf

[DCI FA Doc]: IOWN Global Forum Data-Centric Infrastructure Functional Architecture; https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-DCI-Functional-Architecture-1.0-1.pdf

[Cloud Network]: Characterizing the Availability and Latency in AWS Network From the Perspective of Tenants; https://ieeexplore.ieee.org/abstract/document/9718592

[Cloud Analysis and Object Storage]: Best Practices for Amazon EMR; https://d0.awsstatic.com/whitepapers/aws-amazon-emr-best-practices.pdf

Annex A: Smart Factory Services in detail

In this annex, three examples of the smart factory services and relevant data management requirements are described.

A.1. Remote monitoring and control of manufacturing machines and robots

Service Scenario

In the Smart Factory, more and more machines and robots will be connected and remotely controlled to flexibly and efficiently manufacture various products. Machines and robots will be able to manufacture more autonomously because the system can consider the factory structure and interference between machines and robots, and so on.

However, the latency of such controls will be an issue. For example, in some manufacturing processes, including chemical and/or thermal reactions, it is necessary to monitor the state of work-in-process with various sensors and perform feedback control within 10 milliseconds or less. The robot's motion must be snappy for better production efficiency. Therefore, the end-to-end process, which consists of sensor data collection, device state recognition, and feedback controls, will have to be done in cycles of tens of milliseconds or less, too.

Figure A-1 below shows an image of such device state management and remote control system.

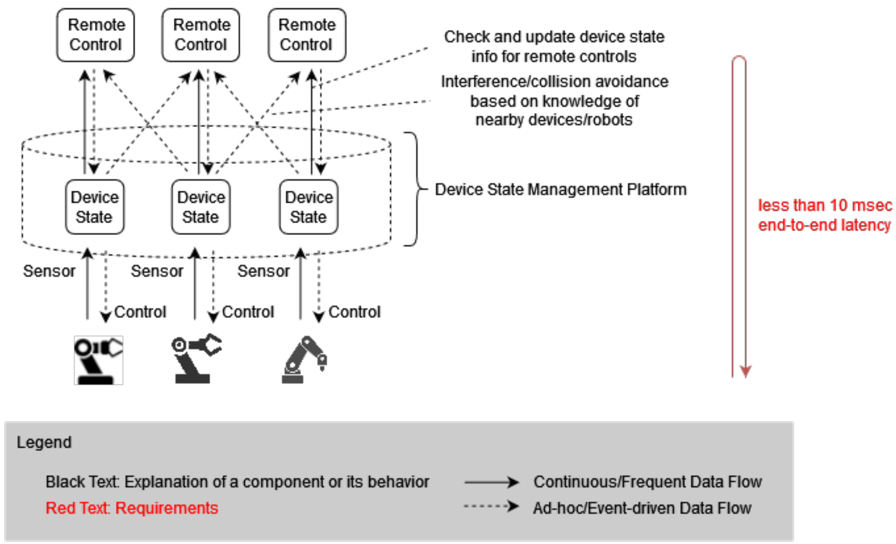

Figure A-8: Device State Management and Remote Control System

Challenges

Obviously, end-to-end latency is the most difficult challenge. In today's cloud, even in the process of just reflecting the device state in KVS, there is a large jitter of more than a 10-millisecond. Alternatively, if it is configured not to use KVS and manage it in cache memory in the control application for latency purpose, then the system will easily fail. It is noted that the application environment, such as VM, containers, etc., will be easily reset. Therefore, it is quite difficult to achieve the necessary real-time control. Device state management should ideally be able to update and query data on the order of sub-millisecond.

Another challenge is data processing efficiency. When such a system is built in today's cloud with a focus on real-time-ness, it is required to run remote control workloads on a resource that is almost entirely dedicated to each device, which is very costly. This system must be built so that one server can manage the data of at least dozens of machines and robots.

A.2. Intelligent Machine Failure Prediction

Service Scenario

Machine failure prediction is another expected service in the Smart Factory. Once a machine failure happens, it can lead not only to the disposal of work-in-process material but also to the stoppage of the entire production line, resulting in significant costs. Therefore, it is desired to collect big data, train a machine learning model to predict machine failure for each machine, monitor the status of manufacturing machines in real-time, predict failures and stop the operation before it actually happens.

Figure A-2 below shows an image of such an intelligent machine failure prediction system.

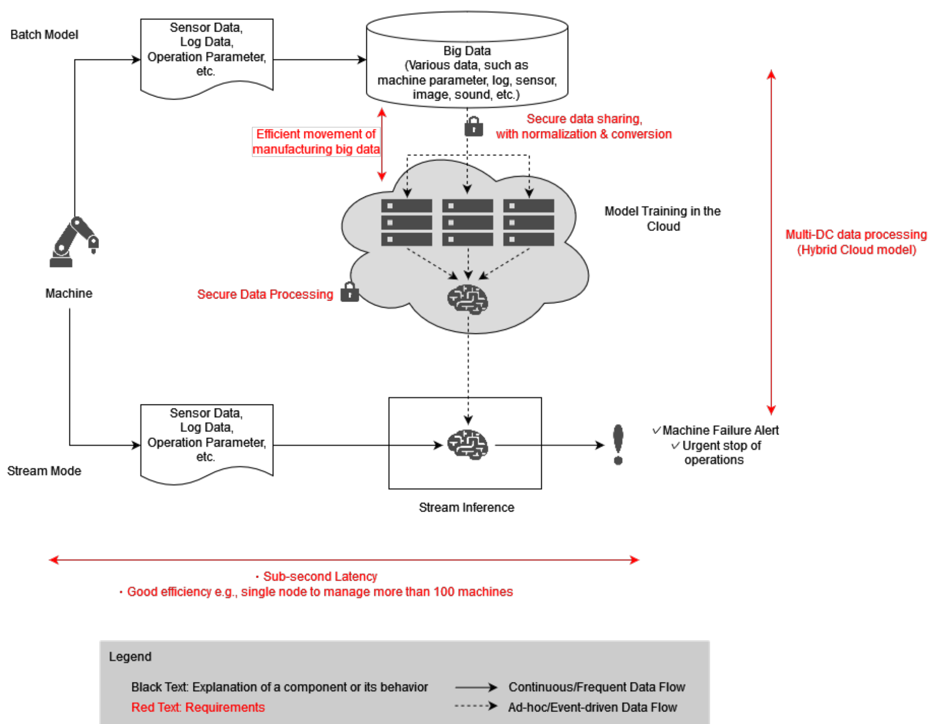

Figure A-9: Intelligent Machine Failure Prediction System

Challenges

To realize the concept described above, big data must be preserved, and processing in machine learning model training processes in an effective and secure manner, which are executed on top of a large amount of computational resources.

Such computing resources are generally not available in the factory. This makes it necessary to use the cloud, but manufacturers are generally reluctant to bring log data out of their facilities. This is because log data contains a large amount of information that can be considered manufacturing know-how and intellectual property, and once this information is leaked outside the company, the company loses its competitive advantage. Therefore, such a system is not yet widely adopted today.

Another reason hindering this adoption is the system performance in the cloud. Because to build such a prediction system, Spark, Database, Object Storage, and other technologies are needed, and as these technologies are essentially distributed computing technologies, they are largely affected by network performance. This means that they can be really slow in today's cloud.

The last reason is the system operation and management burden. It is difficult to build and operate various technologies such as Spark, Database, and Object Storage in-house. This is why the adoption of this technique is so slow.

Therefore, there is a need for new technology that can process data more securely and faster, and that can be set up easily. For example, it is conceivable to normalize the log data and send it to the cloud so that it can be processed safely. The conversion parameters used for normalization never leave on-premises. It will also be possible to let distributed computing run many times faster. That would be achieved by introducing a high-speed network and also by processing the required data only in a pinpoint manner. And lastly, it is desired to have a mechanism that enables people to use such solutions easily as a service.

A.3. Intelligent Manufacturing Control

Service Scenario

The last example in the Smart Factory is a so-called intelligent manufacturing control. This is a very important concept that determines the competitiveness of manufacturing processes where yield ratio is important. Examples of such manufacturing processes are the semiconductor factory wafer processing and the steel mill rolling process.

To prevent the occurrence of manufacturing defects, it is necessary to correctly grasp the condition of work-in-process (WIP) materials and adjust the manufacturing conditions based on them continuously.

The direct response to this is to include an inspection process to check the condition of the work-in-process materials directly. However, such inspection processes are very expensive. It also greatly increases the manufacturing lead time. Therefore, it is required to infer the status of the work-in-process materials by collecting and analyzing various sensor data instead of directly inspecting them. Such analysis should be based on data about raw materials and data from all previous processes. This technique is called ”Virtual Metrology”.

In addition, the condition of each machine will change over time. Therefore, it is also necessary to adjust the manufacturing parameters by taking such changes into consideration. This technique is called ”Run-to-Run control”.

Figure A-3 below shows an image of such an intelligent machine failure prediction system.

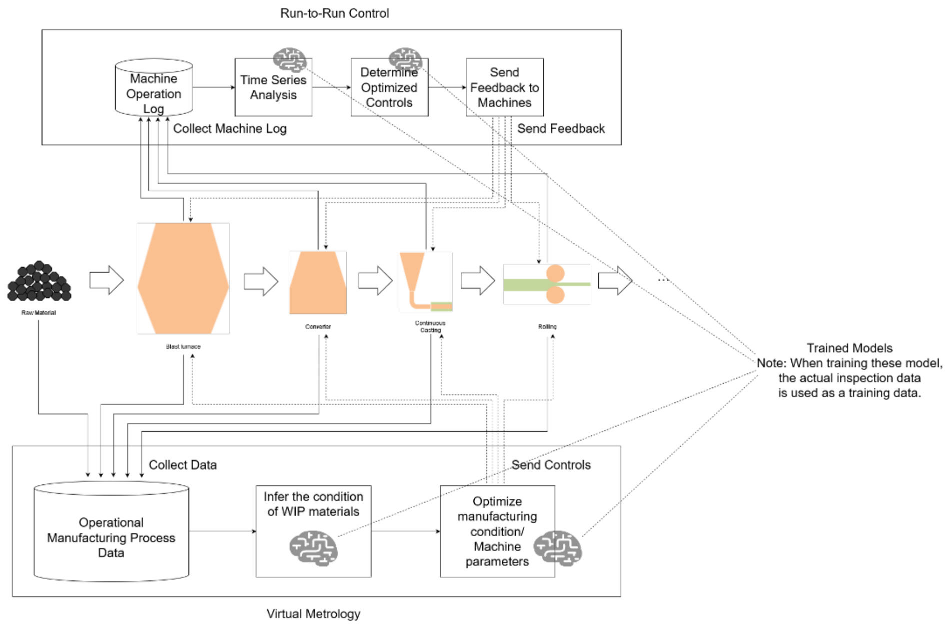

Figure A-10: Intelligent Manufacturing Process Controls

Challenges

The techniques described above need to be run in a near real-time manner to react immediately to minor anomalies and improve yield and product quality. And if it's acceptable in terms of confidential data protection, manufacturing companies want to implement it in the cloud.

But again, today's cloud infrastructure is not enough. This is because the end-to-end manufacturing process is complex and contains many sub-processes, therefore, it is necessary to continuously collect data across all sub-processes and curate it based on a serial or batch number of concerns and the manufacturing processes.

However, the Data Hub for this, namely the Graph Store, which is responsible for such data processing and analysis, is slow, costly, and not scalable in today's cloud. Therefore, the only way to implement such a solution is to develop an expensive on-premises system. The emergence of a new IDH technology that can realize 10x or 100x faster processing is awaited.

A.4. Requirements Analysis

In this section, the detailed requirements of the Smart Factory services are described considering the above service scenarios:

- Input Data requirements
  - The number of machines and robots in the factory: 10,000
  - The average number of sensors installed at each machine or robot: 10
  - The average data generation velocity at each sensor: 4 bytes at a 10-millisecond cycle, 3.2 Kbps in total
  - The data generation velocity at each machine or robot: 32 Kbps
  - Data ingestion velocity at peak hour: 32 Kbps × 10,000 = 320 Mbps
  - The average operating rate of machines and robots: 60%
  - Data retention period: 10 years
  - The total amount of preserved log data per machine or robot: 10 years × 32 Kbps × 0.60 ≒ 5.6 TB
  - The total amount of preserved log data for 10 years in total: 56 PB

- Data Processing requirements
  - State data management for remote monitoring and control of manufacturing machines and robots:
    - Read-Write ratio: 25%:75%
    - Read-Write transactions per second per machine or robot: 1 second ÷ 10 millisecond × 4 = 400
    - Read-Write transactions per second in total: 400 × 10,000 = 4,000,000
    - Read-Write response time: less than two milliseconds, ideally less than one millisecond to achieve 10-millisecond end-to-end control cycle
    - Read-Write response jitter: less than one millisecond to guarantee 10-millisecond end-to-end control cycle
  - Anomaly detection and failure prediction
    - Algorithms: CEP rules or Time Window-based rules, MSET (multivariate sampling estimation technique), MSET2, etc.
    - Algorithm execution cycle: 1-min to 1-sec cycle
    - Algorithm execution time: less than the cycle time
  - Manufacturing process optimization
    - The average number of preceding manufacturing processes to consider: 10
    - The average number of features extracted per preceding manufacturing process: 100
    - The average number of relevant raw materials: 10
    - The average number of features to manage per raw material: 100
    - Total number of features to determine the WIP material status: 2,000
    - Feature extraction time per manufacturing process, and manufacturing condition optimization: Less than a few sec

      Note: Considering a highly-automated continuous manufacturing process with an inter-process time of 15 seconds, etc.
- Compute-and-storage decoupling requirements
  - Shared storage
    - Ideal shared storage size considering 10 year retention: 56 PB
    - Modest shared storage size considering last 1 year hot data: 5.6 PB
  - Scalability of analysis server clusters
    - Dynamicity: Scale-out and scale-in operations in minutes
    - Scaling range: 0 - dozens of servers, to rapidly process PB-class data and free them when not needed

Annex B: Smart Grid Implementation Models

In this annex, three examples of the Smart Grid services and relevant data management requirements are described..

B.1. Real-time monitoring of the power supply and demand balance

Service Scenario

The first step in building a Smart Grid system is to monitor the supply and demand balance of electricity in real-time at each point in the Smart Grid system.

Regarding the devices to be monitored and their numbers per grid are as follows:

- Smart meters at prosumer households, millions
- Smart meters at large consumer facilities such as manufacturing factories, thousands
- Distributed energy storage resources, such as EV cars, millions
- Switchgear devices and/or transformers facilities, hundreds of thousands
- Power plants such as solar panels and wind turbines, thousands

Figure B-1 below shows an image of real-time monitoring of power balance in the Smart Grid system.

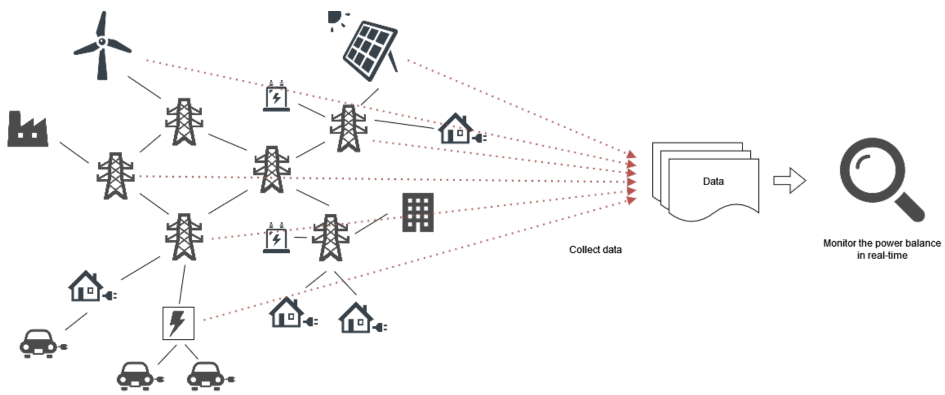

Figure B-11: Real-time Monitoring of Power Balance in the Smart Grid system

Challenges

To monitor the Smart Grid system, it is required to collect data from millions of devices and map them on top of the network model of the Smart Grid system. This process should occur at a frequency of every second or more, with a latency of sub-seconds to respond to sudden changes in the power supply and demand balance. However, this is very challenging in today's cloud. The reasons why it is so difficult are described below:

- Today's cloud network service is not fast. This adds latency to data collection.
- Data Hub services in today's cloud are not fast, too. Data write and read operations have significant jitter, and their response time could be more than hundreds of milliseconds.
- The default method for large-scale data collection in today's cloud is to use a message broker. But this adds an even bigger delay.

Due to these limitations, it is very difficult to build a system that monitors the situation with a delay of less than 1 second.

B.2. Charge and discharge controls of the distributed energy storage resources

Service Scenario

For the purpose of stable operation of the power grid, forward-looking adjustments will also be needed. This means that it is also necessary to periodically forecast the supply and demand based on data collected from the Smart Grid system as well as external data such as weather forecast for the next several tens of hours, and determine the right actions, i.e., charge and discharge controls for all distributed energy storage resources.

To forecast the power supply and demand at each household, EV car, manufacturing factory, solar power plant, etc., not only supply and demand history data but also other data such as weather conditions and plant operation data need to be analyzed together. It should be noted that external data access needs to be done quickly so that the less than 1-minute prediction cycle is guaranteed and also securely so that any confidential information is not leaked.

Once the forecast is made, the Smart Grid system determines the optimal controls to balance the power supply and demand. This means that it determines from which distributed energy storage resources to supply power if there is a shortage of power and determines to which distributed energy storage resources to store the power if there is the extra power in an optimal way. And for that purpose, the Smart Grid system will consider the energy flow limits of devices such as switchgear. This is an essential element for stable bi-directional power supply control. The Smart Grid system will also consider the propagation of power supply through the network, because there is transmission loss, which must be taken into account in order to optimize controls, i.e., to determine from which distributed energy storage resource to supply the power.

In addition to flow limits and transmission loss, there are other conditions to consider. For example, an EV car cannot run when the battery is exhausted. Therefore, it will be necessary to keep a certain amount according to the policy defined by each owner participating in the Smart Grid. Another example is the case where power shortage cannot be avoided. In other words, it is the case that the amount of electricity stored in the distributed energy storage resources run out considering the power supply and demand forecast. In such cases, coordination will be made with other Smart Grid systems to get power supply from them, power saving requests will be made to reduce power consumption, and/or dimming and air conditioning will be adjusted within a range that does not affect people's lives and businesses.

It should be noted that some of the distributed energy storage resources may be disconnected or broken suddenly. Such a situation must be detected immediately, and appropriate alternative measures must be taken immediately if the distributed energy storage resource of concern is used to balance supply and demand.

With these mechanisms, the Smart Grid system can accommodate millions of prosumer households, buildings, facilities, and a similar number of distributed energy storage resources such as EV cars, and supply power to them in a stable manner while maximizing the usage of renewable energy.

Figure B-2 below shows expected mechanisms for the charge and discharge controls of the distributed energy storage resources based on the precise power balance forecasts.

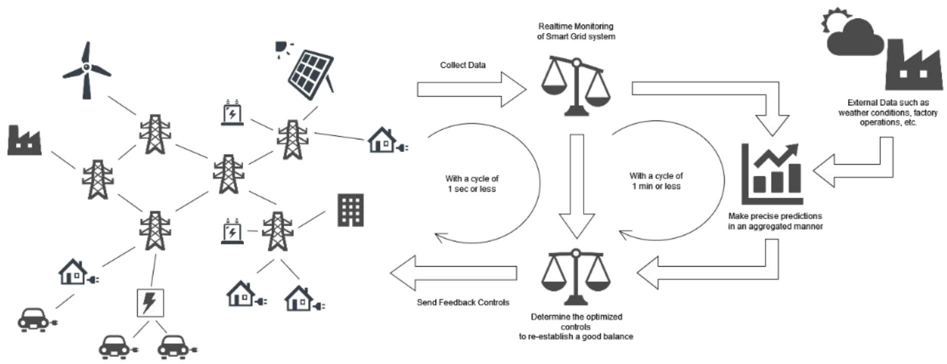

Figure B-12: Charge and discharge controls of the distributed energy storage resources based on the precise power balance forecasts

Challenges

In order to update the power supply and demand forecast data in less than a minute, it is necessary to ensure that both data collection and data analysis processing are completed within a certain amount of time. Collecting external data, in particular, can be problematic because it requires access to many external data sources and because data sharing relies on slow storage services such as object storage. These problems can be summarized as:

- Bandwidth/throughput of data transfer from object storage is not stable and not fast.
- First-byte response time can vary largely such as from a few milliseconds to 1 second.
- With a large number of data sources, the slowest response determines the overall processing time significantly.

In addition, power balance optimization will be technically determined by using a Graph Store, because it can link and associate various data through the electricity delivery network model of the Smart Grid system. However, the Graph Store is not fast enough in today's cloud for this purpose. For instance;

- It will take hundreds of milliseconds to update the data, and
- It will take more than several hundred milliseconds to run an analytical query to find optimal distributed energy storage resources to supply the power or charge if multiple servers are used to manage and analyze the network data.

Therefore, to realize the Smart Grid system, it may be required to use a Graph Store that can accept more than hundreds of thousands of data updates to the network data consisting of millions of nodes and edges, and analyze the propagation and aggregation of location point data throughout the electricity delivery network to make a precise prediction of the supply and demand balance statuses in the Smart Grid system, in sub-second latency. This could mean that latency for write operations to the Graph store shall be tens of milliseconds, and multi-hop analytical queries shall be a few hundred milliseconds or less.

B.3. Intelligent power interchange among multiple Smart Grids

Service Scenario

Considering the power consumption and power generation forecasts, if the power is inevitably insufficient or is expected to exceed the power storage amount, power exchange with other Smart Grid systems shall be coordinated. Such adjustments can be nationwide. That's because there may be the case only a far remote Smart Grid can supply the power. In such cases, power is exchanged between multiple Smart Grids that link the requester and the supplier.

There will be two possible ways to make such wide-area adjustments. One is based on direct coordination among the Smart Grid systems. In this case, the Smart Grid system that will run out of power will query other Smart Grid systems, including those far away, for potential power interchange. Then, receiving and considering replied answers, the final power interchange will be ordered to the Smart Grid systems involved in the power interchange and receive the power. The other way is to build another system to orchestrate the power interchange under the consensus of multiple Smart Grid systems. This system monitors wide-area power balance statuses, determines the adequate power interchange automatically as needed, and orders it to the participating Smart Grid systems.

Beyond that, even frequent coordination will also need to be supported. This is because there will be a case where it is adequate to supply power from the adjacent Smart Grid rather than from the distributed energy storage resources in remote locations within the same Smart Grid, especially when a power shortage occurs near the boundaries of these two Smart Grid systems. If such a scenario is in use, more frequent coordination controls will be made between the smart grids.

Figure B-3 below shows the federation-based power interchange mechanism among multiple Smart Grid systems.

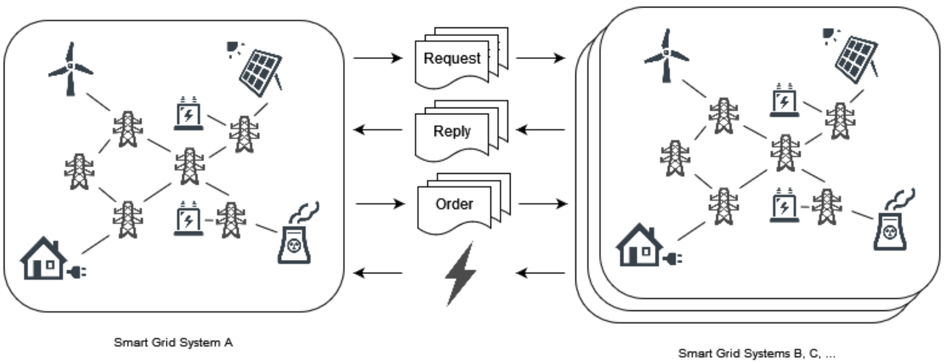

Figure B-13: Direct coordination-based power interchange mechanism among multiple Smart Grids

Challenges

The power interchange mechanism among multiple Smart Grids will require efficient integration of Smart Grid systems, which may be placed in different data centers. This is because it is expected that the controls to rebalance the power supply and demand need to be done within one second after a power shortage has occurred if frequent coordination between two adjacent Smart Grid described above is assumed.

Such a mechanism is difficult to achieve with today's cloud-based solutions. The reasons are as follows:

- The electricity distribution network models managed on Graph Stores in different data centers need to be interconnected, and cross-sectional analysis needs to be executed on top of it. However, Graph Stores with different semantics and with different security contexts belonging to different entities cannot be integrated easily and rapidly today.
- In addition, due to delays in communication between data centers and overhead to access the Graph Stores, such coordination only works at low speeds, and thus, the required latency of less than one second cannot be satisfied.
- The additional processing workload associated with this coordination between Smart Grids can affect the overall service performance. Disruption of local power control due to wide-area coordination of power interchange is a mess. Graph store implementations in today's cloud do not have sufficient resource controls for this.

A new cloud-based graph store that solves the above problems will be needed for fine coordination of nationwide power interchanges.

B.4. Requirements Analysis

In this section, the detailed requirements of the Smart Grid services are described considering the above service scenarios:

- Input Data requirements
  - The number of data sources per data source category
    - Smart meters at prosumer households: 2,000,000
    - Smart meters at large consumer facilities such as manufacturing factories: 5,000
    - Distributed energy storage resources, such as EV cars: 1,000,000
    - Switchgear devices and/or transformers facilities: 50,000
    - Power plants such as solar panels and wind turbines: 1,000
    - Other sources:
      - Weather forecast: 1-3
      - Plant operation plan: 100
  - The average data collection frequency per data source category
    - Smart meters at prosumer households: 1 min
    - Smart meters at large consumer facility: 1 sec
    - Distributed energy storage resources, such as EV cars: 1 min
    - Switchgear devices and/or transformers facilities: 1 sec
    - Power plants such as solar panels and wind turbines: 1 sec
    - Other sources:
      - Weather forecast: Hourly
      - Plant operation plan: Hourly
  - The average data generation/collection velocity per data source category
    - Smart meters at prosumer households: 4 byte/sec
      Note: Assuming that the power consumption and power generation amount per second are managed as 2-byte data respectively.
    - Smart meters at large consumer facility: 40 byte/sec
      Note: Assuming power consumption per 100 milliseconds is managed as 4 bytes of data each.
    - Distributed energy storage resources, such as EV cars: 28 byte/min
      Note: Assuming that data such as the battery capacity (2 bytes), the remaining battery capacity (2 bytes), the allowable lower limit of remaining battery capacity (2 bytes), charger/discharger connection status, and connection points (18 bytes), charge & discharge amount for the last 1 minute (4 bytes), are collected, continuously.
    - Switchgear devices and/or transformers facilities: 1600 byte/sec
      Note: Assuming there are 2 bus bars, and 18 branch bus conductors, and AC voltage and current data of 4 bytes each are collected from there every 100 milliseconds.
    - Power plants such as solar panels and wind turbines: 40 byte/sec
      Note: Assuming power generation amount per 100 milliseconds is managed as 4 bytes of data.
    - Other sources:
      - Weather forecast: 4,000 × 100 x 60 × 48 = 1 GB/min
        Note: Assuming that the Smart Grid system of concern manages a 1,000 Km2 area, the weather forecasts are created at 1-minute intervals over the next 48 hours based on 0.25 Km2 mesh, and each data record corresponds to 100 bytes of data including such as weather condition, temperature, humidity, illuminance, wind, etc.
      - Plant operation plan: 1 KB × 60 × 48 ≒ 2.8 KB/min
        Note: Assuming that power consumption forecasts represented as 1 KB of data are created at 1-minute intervals over the next 48 hours.

  - The open data sharing/access
    - Number of external data sources: more than 1,000

      Note: It should be noted that there are various external data sources such as weather forecast information, factory systems, large-scale commercial facility systems, charging stations for EV cars, tram operation systems, etc.
    - Data size: a few KB to a few GB

      Note: The size of the data to be published is determined by the convenience of the publishing company.
    - Data retrieval response time: less than 9 sec

      Note: It is assumed that 15% of 1 minute prediction cycle time can be used for data collection.
- Data Processing requirements
  - Real-time monitoring of the power supply and demand balance:
    - The collected data must be reflected in the supply-demand balance with a latency of much less than the collection cycle. This shall be within several tens of milliseconds.
    - With the same latency as above, the data of the child nodes within the electricity delivery network must be aggregated for parent nodes, such as switchgear.
  - Charge and discharge controls of the distributed energy storage resources
    - If the supply and demand balance collapses beyond the permissible range in any of the parent nodes such as the switchgear, the system has to select appropriate distributed energy storage resources and instruct them to charge and discharge within 1 second.
    - When selecting appropriate distributed energy storage resources, the transmission loss, the allowable lower limit of remaining battery capacity, etc. must be considered.
    - To ensure the required response time, evaluate the score of the solutions found within the permissible time and select the most appropriate one. This shall be within a few hundred milliseconds.
  - Intelligent power interchange among multiple Smart Grids
    - A power interchange shall be coordinated within 1 second, considering flexible power interchange near the smart grid boundary.
    - When dealing with excess or deficiency based on the power supply and demand forecasts, inquire about the possibility of power interchange with multiple Smart Grid systems, evaluate the results, and select the optimal plan.

Annex C: Metaverse Implementation Models

C.1. Interactive Live Music

Interactive live music provides new means to enjoy music entertainment, by collecting data from artists and audiences, such as volumetric capturing of the artists, and creating a virtual space with these input data. In this virtual space, artists and audiences can intimately interact, such as the audience cheering on the artists and waving together, and the artists calling out to the audience as if they are all together in the real live performance.

More than tens of thousands, even millions of audiences can participate from their homes or karaoke boxes and experience an immersive experience. The collected audience data will be analyzed to provide the artists with appropriate feedback on the behavior of the audiences.

Assuming a case with a very large number of viewers, the video image provided to the viewers will be rendered on the data center. Because, it's more efficient than providing the data needed for rendering, and it also makes it easier for viewers to join using cheap consumer devices.

Figure C-1 shows an image of such interactive live music services.

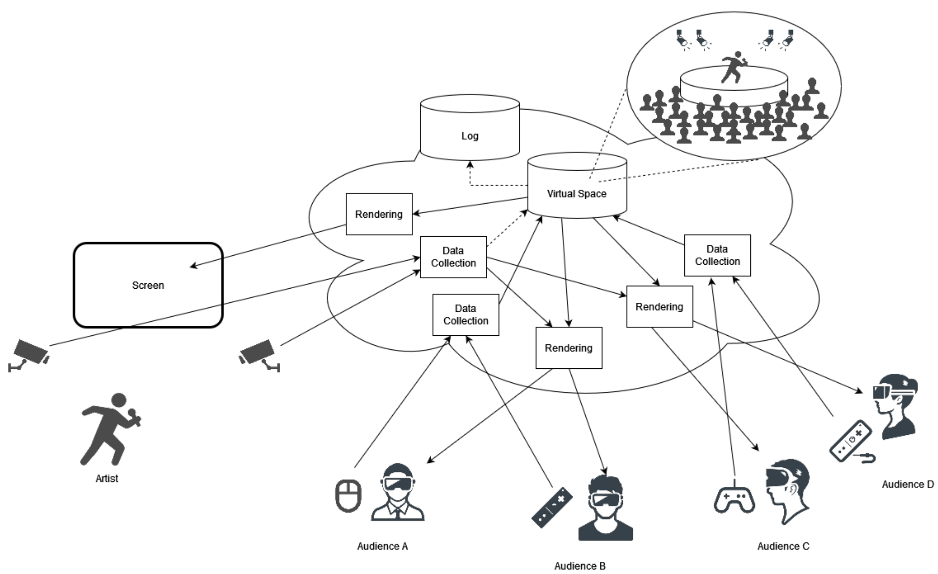

Figure C-14: Interactive Live Music Service

Challenges

To create a sense of reality, input data from the artist and audiences must be projected on each video device within a 70 milliseconds delay. To achieve that, it is necessary to collect data in a cycle of 10 to 20 milliseconds, considering the rendering and other delays.

This implies the creation of the virtual space with millions of audiences will be very difficult. This is because tens of millions of data records in the virtual space server(s) have to be updated every second, and some percentage of the virtual space data (partitioned data) needs to be fed to the rendering servers so that this overall latency becomes less than 10ms or so.

This seems to be very difficult to realize in today's cloud. Even with KVS, which focuses on the primary key-based record accesses, the update alone can cause a delay of 20 milliseconds or more because there is a large jitter in its response time. Therefore, it is required to develop a new faster data hub technology.

Another issue is the rendering time. If millions of audiences participate together, it is necessary to make the video from the different viewpoints of millions of audiences. Assuming a 120 FPS service, virtual space data need to be referenced more than 100 million times per second or so. It's very difficult. Perhaps it is necessary to divide the rendering server by a certain visualization range and synchronize the partitioned data required for rendering in real-time from the virtual space server(s) to the rendering servers. For that, high-speed data replication technology will also need to be developed.

C.2. Requirements Analysis

In this section, the detailed requirements of the Metaverse services are described considering the above service scenarios:

- Input Data requirements
  - Artist Input
    - Volumetric data, e.g., 60 FPS point cloud data: ~120 Gbps
    - 32ch 3D Audio data: ~25 Mbps
  - Audience Input
    - Number of audiences: 1 million
    - Motion data, e.g., 30 FPS depth sensor data, per audience: 0.25 Gbps
    - Motion data in total: ~ 0.25 Pbps
    - Audio data per audience: ~1.5 Mbps
    - Audio data per audience: ~1.5 Tbps
- Data Processing requirements
  - Virtual space creation:
    - Avatar motion updates per second: 33 million avatar object updates per sec.
    - Avatar motion update response time: within a few milliseconds.
    - Collective avatar query response time: less than 10 milliseconds.

      Note: Assuming that all input data collected is reflected by the next two frames and the server-side delay becomes less than 66.7 milliseconds.
  - Rendering
    - Video image to be created and delivered to audiences: 120 FPS, 4K.

      Note: Assuming that motion-to-photon latency shall become less than 10 milliseconds, although the collection frequency of each input data is larger than it.
    - Video image to be created and delivered to the artist(s): 120 FPS, 4K x 2 Displays.

Annex D: IDH PoC roadmap

As mentioned in Chapter 1, this document describes an initial PoC scenario only. However, to address various DX service demands, and to develop various IDH service classes, multiple PoC scenarios shall be conducted. The below table shows a planned roadmap for performing such PoC:

Table D-2: IDH PoC Roadmap

	STEP 1 (2022 H2 - 2023 H1)	STEP 2 (2023 H2 - 2024 H1)	STEP 3 (2024 H2 – 2025 H1)
DX Services	● Smart Factory ● Smart Grid ● Metaverse	● Smart City ● Green Twin ● Financial Twin	● Human Twin, etc.
IDH Service Classes	● Distributed RDB ● Key-Value-Store (KVS) ● Graph Store ● Virtual Data Lake (Federated Object Storage) ● Message Broker	TBD	TBD
IDH Common Functionality		● Data Security ● Data Rights Management	TBD

…

History

Revision	Release Date	Summary of Changes
1.0	October 21, 2022	Initial Release