Evolution of Virtualization Technology and IIJ's Initiatives

2.1 Introduction

The concept of the "cloud" has become such an ordinary part of our lives that we barely give it a second thought anymore. Today, companies all over the world offer cloud services, and the infrastructure underpinning those services continues to expand.

We covered the evolution of IIJ's cloud services platforms in a previous article[*1]. In this edition, we look at the ever evolving virtualization technology that has made the concept of cloud computing possible.

2.2 History of Virtualization Technology

2.2.1 Early Days of Virtualization

Virtualization is the abstraction of computer resources, and encompasses various technologies for providing an abstraction layer between software and physical hardware and managing those resources. Its history goes all the way back to the 1960s.

The word "virtualization" can be traced back to earlier stages of computer development. In the 1940s and 50s, different computer models commonly had different architectures, and so to mitigate the risks associated with new designs, computer designers began to use past models as a reference to create "compatible machines" that used instruction sets compatible with those of previous models and that shared the same logical design.

The late 1950s saw the advent of computer models that emulated the instruction sets of previous-generation models in microcode so as to provide backward compatibility, and the 1960s brought a rising trend toward standardizing computer architectures and maintaining compatibility. It was around this time that what were called "virtual machines" (the term was actually used earlier than this) came to be used to provide compatibility by emulating the instruction sets of different models of computer, and thus the word "virtualization" entered the vernacular.

It was then in 1964 that the first "hypervisor," capable of running multiple virtual operating systems on a single computer, appeared. As you may know, however, we would then have to wait until the 2000s before virtualization became further widespread.

Virtualization at the time was intended to allow multiple users to use a single large and expensive computer. The use cases revolved around corporate departments that used it to run thousands of routine tasks, such as batch payroll processing, at high speed.

Over the next few decades, a number of other approaches were taken to solve the problem of allowing multiple users to use a single device. One such approach was time sharing, which offered the solution of separating users within the one operating system (OS), rather than using hardware-based virtualization to logically separate users into different virtual machines. This meant the virtualization of the time was relegated to certain niche areas, and hardware-based virtualization thus faded from the forefront for a time. Time sharing is what created the impetus for the creation of UNIX and Linux, which we use in our systems today.

2.2.2 x86 Virtualization Sparks a Resurgence in Virtualization Technology

In the 1990s, many companies adopted vertically integrated systems (mainframes) comprising physical servers and software from a single vendor. Existing applications could not be run on hardware provided by other vendors.

Meanwhile, Intel Corporation, which rose to prominence with its x86 architecture being used in IBM's PCs in the 1980s, unveiled the Pentium CPU in 1993. It was also

*1 Internet Infrastructure Review (IRR) Vol. 60, Focused Research(2) "Evolution of the IIJ Cloud—Commemorating 30 Years" (https://www.iij.ad.jp/en/dev/iir/060.html).

around this time that the market for personal computers underwent a rapid expansion amid the spread of commercial Internet services and the release of Microsoft Windows. One after another, manufacturers who had previously developed computers based on their own proprietary architectures began developing compatible machines based on Intel CPUs.

Intel followed up in 2001 with the release of Xeon CPUs for servers. This prompted moves to replace corporate systems, which had until then been dominated by mainframes and UNIX, with low-cost, general-purpose IA (Intel Architecture) servers.

Intel's CPU architecture, however, was not designed for use with multiple operating systems and thus rendered existing forms of virtualization inapplicable. It was in this context that VMware Virtual Platform, released by VMware Inc. on February 8, 1999, made it possible to achieve virtualization in software.

The earliest form of software virtualization for the x86 architecture (x86 virtualization) worked by running a virtual layer on the host OS (Windows or Linux). The need to go through the host OS resulted in a large overhead, such that performance was not suitable for commercial services. This prompted the release of hypervisor products with their own custom kernels capable of providing a virtual layer without the need for a general-purpose host OS.

When operating without hardware virtualization support, x86 virtualization of this era always used a dynamic instruction conversion mechanism to capture the execution of specific instructions and dynamically replace them. This technique inherently came with some overhead in terms of performance compared with virtual machines on virtualization-friendly architectures.

Paravirtualization then arose as a means of solving this performance problem. Instead of emulating the hardware, paravirtualization involves modifying the guest OS to provide a special API. This was used in early forms of server virtualization. A key example of this approach was the open-source Xen (up to 2.x).

In contrast to this, full virtualization (native virtualization) worked by converting privileged CPU instructions in the virtualization layer to allow the guest OS on the virtual machine to run as is without any special modifications. This made it possible to run OSes like Windows that did not support paravirtualization at the time. Key products in this space included Windows Virtual Server, VMware Server, and VMware ESX.

2.2.3 Limitations of Software Virtualization and Emergence of Hardware-Assisted Virtualization

As discussed, the problem with past forms of virtualization was that operations performed by the CPU in hardware had to be executed in software, which created a non-trivial virtualization overhead. The I/O delays caused by the exchange of data between devices, OSes, and applications, in particular, were unsatisfactory. This is why the earliest forms of hypervisor-based virtualization were used somewhat sparingly for the purpose of running guest OSes migrated to run applications coming up against their limits due to equipment being too old, or for testing during application development. It did not have the power to replace the high-spec, high-speed mainframes and UNIX servers of the time.

IA servers were gaining market share in the late 2000s, and this is when Intel, struggling to increase the performance of its processors, switched to 64-bit multi-core CPU architecture. It was around the same time that Advanced Micro Devices (AMD) implemented hardware virtualization support, sparking rapid progress in the

field of virtualization. This support made it possible to run the virtual machine monitor (VMM), part of the functionality that had previously been run in software, on the CPU, and this greatly reduced the overhead.

To describe its role in simple terms, the VMM does for the guest OS what the OS does for user applications. The OS manages physical resources such as devices and memory, and isolates processes (user applications) from each other in memory. Similarly, the VMM emulates physical resources for each guest OS, coordinates requests from the guest OSes, and isolates the guest OSes from each other in memory.

Speeding up these VMM tasks and thus improving the performance of the guest OSes helped hypervisor-driver server virtualization gain traction.

2.2.4 Maturation of Server Virtualization and Advent of the Cloud

Hypervisors that greatly reduced the virtualization overhead thanks to the implementation of virtualization support functionality in CPUs thus made significant inroads into commercial use, and products that previously provided guest OSes via paravirtualization mechanisms (such as Xen 3.0 and later) also started providing full virtualization of guest OSes, and thus the development of virtualization products suddenly became increasingly more competitive.

It was around this time that Eric Emerson Schmidt, CEO of Google Inc., first used the term "cloud computing" in a speech at the Search Engine Strategies Conference[*2], and thus the term "cloud" came to be used in the sense we are familiar with today.

Following this, Microsoft implemented Hyper-V, which provides support for CPU virtualization support functionality, as a Windows Server feature. Similarly, in the open-source world, version 1.0 of the Kernel-based Virtual Machine (KVM) was released, integrated into the Linux kernel. And alongside the earlier products in this space, these offerings helped accelerate the development of the virtualization market.

In 2008, Google unveiled Google App Engine (GAE), a PaaS offering designed for cloud computing. And alongside Amazon Web Services EC2 (AWS EC2), a service Amazon had launched earlier for the purpose of renting out surplus resources present after replacing IA servers within its own systems, this really opened up the cloud market. IIJ also launched a commercial cloud service under the IIJ GIO brand in 2010, and we continue to develop this and make it available today.

Once the hypervisor performance issue was resolved, attention turned to the difficulty of operating and managing virtualization products. Similar to hardware servers, when it came to configuring system elements such as network and storage, different manufacturers and devices all adopted different architectures (unlike the case with IA servers, where this had gradually grown more consistent over time), and so specialized knowledge was required to operate the products. Virtual machines also had to be moved between physical resources to improve availability, efficiency, and stability. Then came products called orchestrators, which abstract and allow overarching control of system elements that had previously been furnished separately in the form of physical resources. Key products here include VMware vRealize Orchestrator, OpenStack, and Apache CloudStack.

2.2.5 Microservices Architecture and OS-Level Virtualization

Thus, various technical solutions were deployed to overcome the challenges associated with hypervisors, and the business environment, which benefited from these technologies, was changing at an ever-increasing pace.

*2 Google Press Center, "Search Engine Strategies Conference Conversation with Eric Schmidt hosted by Danny Sullivan," August 9, 2006 (https://www.google.com/press/podium/ses2006.html).

While mechanisms for rapidly deploying cloud resources had been developed, the great number of infrastructure elements that software developers had to deal with, beyond the required libraries, meant that prodigious effort was needed to keep up with the demands of business.

To solve this problem, developers devised a software development methodology called microservices, which involves building applications out of collections of small, independent pieces of software. As only a small amount of resources were required—the environment to run the application—container technology was adopted to virtualize some, not all, of the OS functionality, and this led to the creation of container management systems, a new type of product incorporating virtualization capabilities. Key products here include Kubernetes and a derivative of it called Red Hat OpenShift Virtualization.

So as the technology has advanced through the years, user perceptions of the issues that needed to be addressed have changed, and virtualization technology has accordingly evolved and taken on new roles.

2.3 Technology Selection and Service Development

Technologies have solved various issues, and products incorporating those technologies have expanded the market, allowing us to diversify and differentiate our services, but as you know, not all products work to the desired level.

So just because a new product becomes available does not mean it will immediately be suited to the user environment. As discussed in the previous section, however, virtualization technology has been advancing rapidly in recent years, and given the lack of sufficient time to fully evaluate new technologies, we now face the need to start investigating such technologies at an earlier stage, rather than waiting for service requirements to be finalized, to identify elements that will determine the direction of development efforts. And even when it comes to products that have not currently been adopted for use in any services, there is always potential for the market to expand rapidly due to technological advances of one kind or another, so we continue to keep tabs on the markets around those products as well.

The actual product evaluation process consists of two steps: check the basic functionality, and conduct checks for each individual service. The basic functionality means the functions or features that the vendor claims have been implemented in the product, as well as their expected performance. It depends on the product being evaluated, but for the infrastructure layer, we look at the operation of physical equipment and the behavior of the catalog values or limit values, and for the virtualization layer, we look at the combination of equipment that makes up the product, the operation and performance of the hypervisor, and so on. The way we evaluate products is also not set in stone; instead, we incorporate knowledge and insight gained from operating services (mainly availability and completeness as it relates to quality) and gradually increase the number of common evaluation criteria. When it comes to conducting checks for each individual service, our basic approach is to evaluate the functional and non-functional aspects for fulfilling SLAs based on service requirements. This involves looking at the parts that, when assembled into a system, will be provided to users and the parts that we will be operating and managing (not just maintenance but also the mechanisms for distributing resources to users etc.). Our main focus is integration with relatively higher-level software and applications.

There is another important consideration when it comes to operating a service and maintaining quality. The emergence of new products determines the lifespan of existing or older-generation products, so you must consider how you

will handle changes to systems that use such products. All is well if successor products stick with established mechanisms and frameworks, but when they adopt a completely different architecture or a completely different license agreement, you have to determine whether you will be able to continue providing your service using the new product. To ensure you can deal swiftly with such situations, it is crucial to determine what level of compatibility exists between different products, in terms of the features a product shares in common with similar products and how they perform, rather than focusing on the unique characteristics of a given product. From these perspectives, identifying and understanding technologies early is crucial for the purpose of sorting out both the functional and non-functional requirements of competing products on the market.

Finally, when looking at products to provide as part of a service, we evaluate cost. Even if a product fulfills users expectations of functionality, we will not adopt it if the cost seems inappropriate for purpose. This applies both to third-party and IIJ products—if they are not suitable in this respect, we do not select them.

This is how we have evaluated products at IIJ over the years, adopting the technologies that had become established at the time and developing platforms suited to the application in question.

2.4 Virtualization in Action at IIJ

■ IIJ GIO Hosting Package Services

■ IIJ GIO Component Service Base Server V Series Linux Type

At IIJ around 2008, we reviewed the design of our internal service hosting infrastructure, consolidating uniformly configured resources into a pool so that they could be divided up and used according to demand. We adopted Xen as the virtual machine monitor for dividing up and using the resources of the physical servers. Drawing on our experience operating this system within IIJ, we also adopted Xen for the IIJ GIO Hosting Package Service and IIJ GIO Component Service V Series Linux Type released in 2010.

The Xen hypervisor fully isolates the privileged administrator domain, dom0, from each of the user domains, denoted domU, allocates resources to those domains, and provides memory protection. As the privileged domain, dom0 manages the hardware and operates the Xen hypervisor. The domU domains are virtual user machines—they run independently and are isolated from other domains by the Xen hypervisor. Each domain is thus fully isolated, ensuring security and stability.

While hardware-based performance support was still scarce at the time, we chose Xen because paravirtualization allowed for practically feasible performance when the guest OS was optimized for Xen, and the source code was open and it was thus being actively developed in a transparent manner.

To ensure service stability, we made modifications to the Xen hypervisor as needed for IIJ services, and to ensure continuity of operations, we froze the version of Xen, backporting patches from later versions, and we defined our network security specifications and used our accumulated knowhow to build in measures against security threats, including virtual servers engaging in packet spoofing or attacks on other virtual servers. For the IIJ GIO platform, which provides IT resources to our customers, we streamlined operations by developing our own orchestrator, based on our operational knowhow with IIJ service hosts, to automate the resource delivery process, from the securing and deployment of resources through to their return to the pool. Building an automated control mechanism to centrally manage configuration information and rewrite settings in a coordinated manner made it possible to deploy, without error, a huge amount

of resources that it would have been impossible to deal with manually, and to calculate resource allocations in a fair and efficient manner (Figure 1).

Using simplified system configurations premised on the use of virtualization technology enabled, for instance, live migration, which let us perform maintenance without interrupting the equipment, and we were thus able to ensure sufficient quality even for services made up of thousands of servers. We also use the knowledge gained from this in running the successor service, IIJ GIO Infrastructure P2 Public Resources.

IIJ GIO Component Services Base Server V Series Windows Type

IIJ GIO Component Services are targeted at enterprise systems and were developed with the aim of providing the various components needed for system integration.

In many cases, users' internal IT environments use Windows Server components (Active Directory, file servers, WSUS, etc.), and so to migrate enterprise systems to the cloud, Windows Server needs to be made available in the cloud. We therefore built a separate virtualization platform using Hyper-V to enable us to operate and provide Windows Server with stability.

Microsoft released Windows Server 2008 R2 in 2009. Hyper-V 2.0, available on Windows Server 2008 R2, features Cluster Shared Volumes (CSV), which enables a single storage area (LUN) to hold multiple virtual machines and provide multiple virtualization hosts with simultaneous read/write access. It also enables live migration of virtual machines, among other features. It thus provides the minimum functionality required for a multi-tenant service platform.

The System Center products provided the functionality required for operating a service platform, such as virtual machine lifecycle management and component monitoring, but they are designed to be used through a GUI, so for automation and orchestration purposes, we had to develop tools using the Dynamic Datacenter Toolkit (DDTK) framework provided by Microsoft.

For the IIJ GIO Component Services Base Server V Series Windows Type platform, we used DDTK to implement the orchestration tool for IIJ's internal service operators and

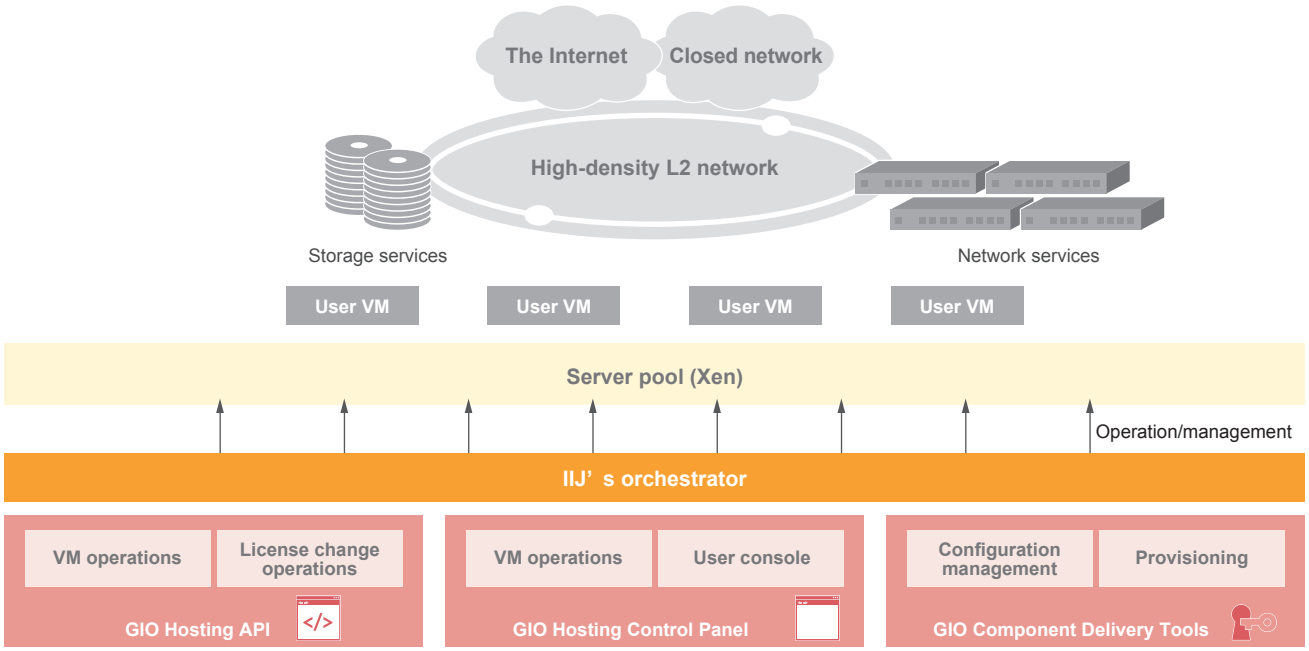

Figure 1: IIJ GIO Hosting Package Services / IIJ GIO Component Services Base Server V Series Linux Type

the control panel that lets users operate their virtual machines (Figure 2).

A lot of useful features for operating services have since been added with each new generation of Windows Server. One such feature is Storage Live Migration, which lets you relocate a virtual machine's storage even while the virtual machine is running.

As this was not available on the initial platform, we implemented our own migration tool (a proprietary tool that required us to stop the virtual machines but also supported rollback after migration work had begun). But Storage Live Migration became available on the service platform starting with Windows Server 2012, and this made it possible to reduce the impact on users when carrying out service maintenance and troubleshooting.

We continued to use System Center as our management tool, and using Service Provider Foundation (SPF), which provides a RESTful API for service providers, made it possible to implement the orchestration functionality that IIJ needed while also keeping development man-hours down.

The Hyper-V base platform was originally developed as a service platform for Windows Server, but from a support and licensing standpoint, it came to be used in a variety of ways over time, such as to provide other OSes on the Hyper-V base platform.

We have continued to make fine-grained updates to ensure we can operate hundreds of Hyper-V virtualization hosts and provide increased service stability in step with the evolution of Windows Server and Hyper-V, but with the

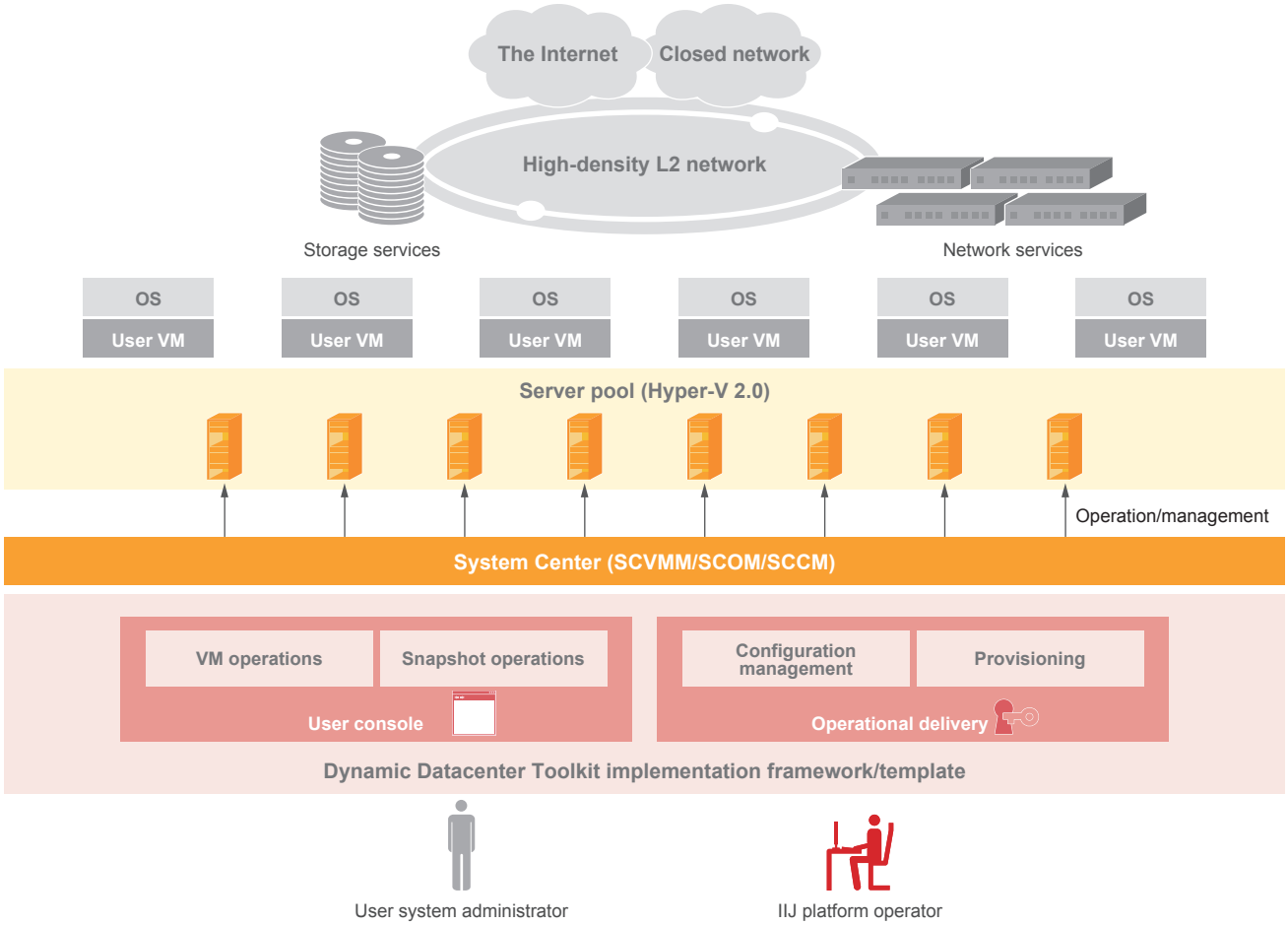

Figure 2: IIJ GIO Component Services Base Server V Series Windows Type

release of the successor services, IIJ GIO P2, we discontinued IIJ GIO Component Services in September 2023.

IIJ GIO Infrastructure P2 Public Resources

In 2015, IIJ released IIJ GIO Infrastructure P2 Public Resources, which can be scaled using software-defined networking (SDN) technology we developed in-house. We selected KVM as the virtual machine monitor and QEMU as the virtual machine emulator. The five years following the initial launch of IIJ GIO in 2010 were also a time during which the various technologies around server virtualization were refined and became increasingly widespread. In keeping with the times, we also switched to KVM+QEMU on the service host platform that provides virtual servers within IIJ.

KVM is tightly integrated with the Linux kernel, allowing direct use of Linux features and security updates. It can use hardware-assisted virtualization features such as Intel VT-x to enable the efficient use of I/O and CPU resources. The virtual machines run as independent QEMU processes, which ensures robust isolation from other virtual machines.

Combining KVM and QEMU entails a high degree of difficulty in terms of configuration and management, necessitating a high level of expertise, particularly in large-scale environments where the intention is to provide resources to customers. Our decision to select KVM+QEMU rested on the fact that IIJ had built a track record and the knowhow to operate virtualization environments using KVM+QEMU, and the fact that advances in hardware virtualization support meant that KVM+QEMU provided practically feasible performance even with full virtualization, so it would be possible to provide SEIL/x86[*3] and Windows without the need for paravirtualization support in the guest OS.

When it came to providing services, given KVM's tight integration with the Linux Kernel, IIJ decided to avoid actively making any unique modifications. We added the functionality we needed, which included making our internally developed orchestrator compatible with our own SDN technology, utilizing development assets imbued with our pioneering services knowhow (Figure 3).

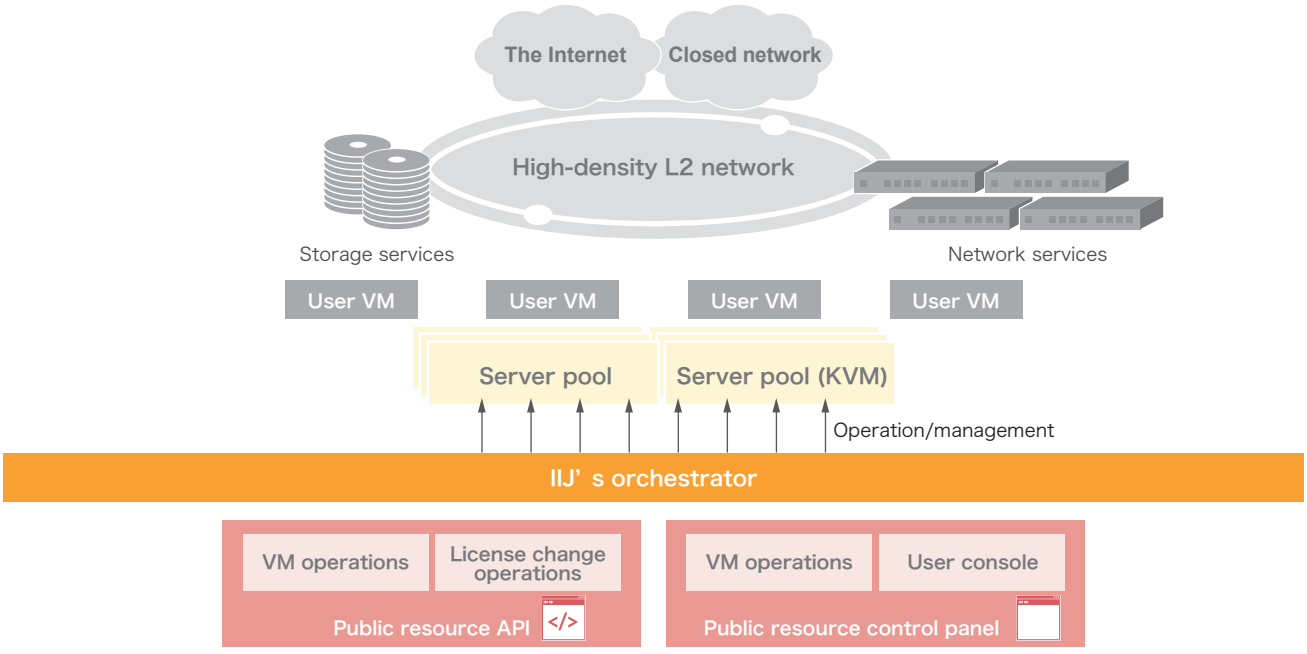

Figure 3: IIJ GIO Infrastructure P2 Public Resources

*3 SEIL/x86 is a high-performance software router developed by IIJ that runs on an x86 architecture-based platform.

While maintaining our established operational approach, making active use of live migration and so forth, we continue to operate our services so as to maintain consistently high quality even at larger scales.

■ **IIJ GIO Virtualization Platform VW Series**
■ **IIJ GIO Infrastructure P2 Private Resources**
□ **IIJ GIO Infrastructure P2 Gen.2 Dedicated Server Resources**

Since 2012, IIJ has provided hosted private cloud services to enterprise customers that use VMware vSphere as the hypervisor. When it comes to virtual machine specs on the sorts of cloud services generally available, user systems need to be configured based on the instance models (vCPU count, memory capacity, OS type, etc.) specified by the cloud service provider, and the difficulty of selecting the right menu options had been a problem for users unfamiliar with designing systems for peak resource usage. With this service from IIJ, the user has direct control over a dedicated hypervisor, allowing them to freely design their virtual machines and select which OS to use, making it possible to configure systems based on the resource allocations that the user has in mind (Figure 4).

Of course vSphere was in use by many users, but its applicability to a wide variety of hardware also made it an easy choice for us to provide as a service operator. On the technical side, as well, commonly implemented functionality was sufficient to provide isolation between users—by using VLAN for networking and functionality for carving out logical disk volumes for data stores, for instance—so development could be tailored to scale without the need for any specialized development efforts (development of special protocols etc.).

As a result, the service has now evolved through three generations and been in operation for over a decade, providing a platform that lets users who had built their own on-premises private clouds using vSphere migrate smoothly to the cloud without any major changes.

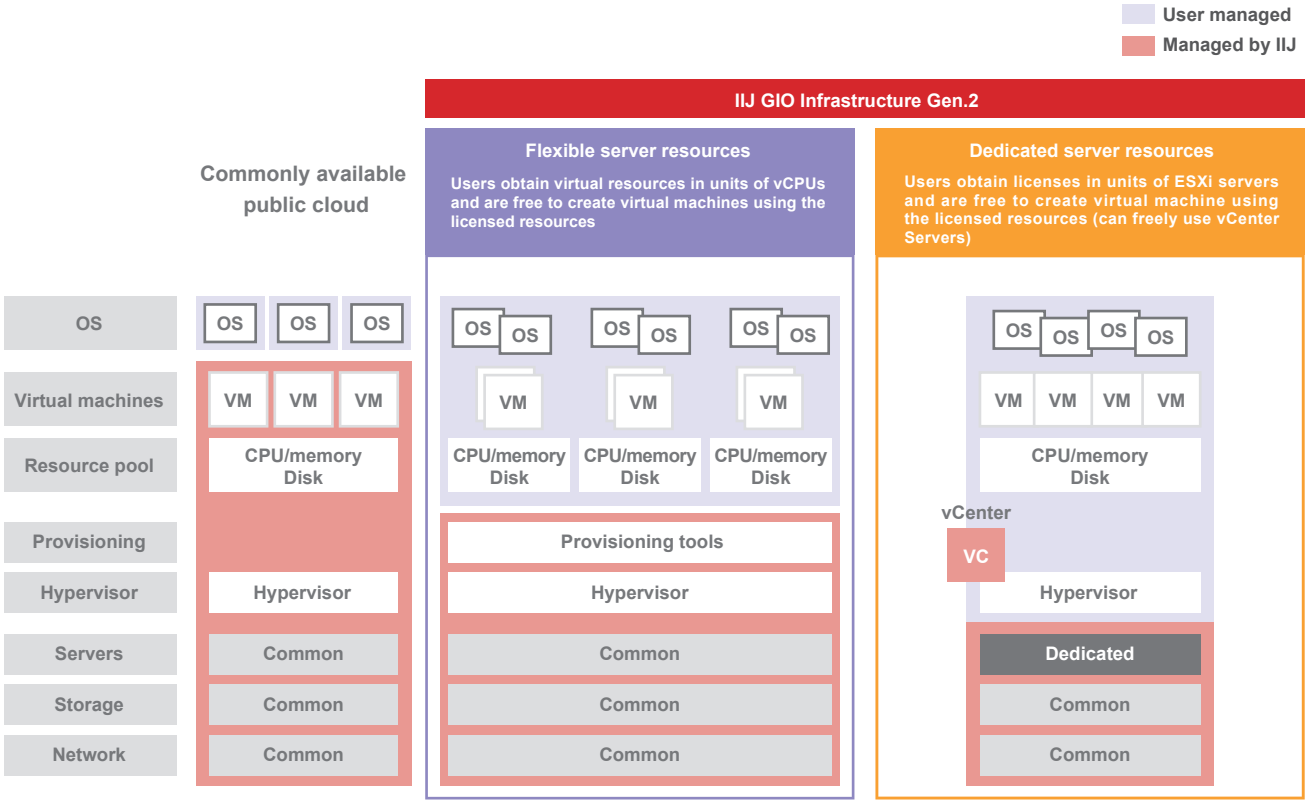

Figure 4: Service Demarcation of Responsibility

☐ IIJ GIO Infrastructure P2 Gen.2 Flexible Server Resources

IIJ released IIJ GIO Infrastructure P2 Gen.2 Flexible Server Resources (FSR), its third-generation public resource service, in 2021. With the previously mentioned service on which users directly operated a dedicated hypervisor, they were able to operate vSphere directly, giving them considerable freedom in building their own systems, but on the flip side, users faced operational challenges, such as the need to upgrade and perform maintenance on the virtualization platform themselves.

For FSR, we selected VMware Cloud Director (VCD), making it possible to abstract individual resources, such as CPUs and memory, and present them to users as a resource pool, and provide a mechanism for dividing up resources and building systems in the same way as with vSphere. The hypervisor layer is hidden from users, but users have the authority to control resources just like with vSphere, while IIJ handles lifecycle management for the hypervisor and hardware, a setup that resolves the issues mentioned above.

This platform design not only benefits users but also completely separates the user environment (mainly software) from the service provider's operating environment (hardware), allowing IIJ to handle maintenance and other tasks that, while important for maintaining the platform, had in the past been a challenge due to user considerations, and it thus provides a more stable service platform from IIJ's perspective as well. The platform also lends itself to continuous development and modifications to meet the contemporary demands of business environments.

2.5 Conclusion

We have taken a trip through the history and evolution of virtualization technology, along with some examples of it in action at IIJ. In our current times, we are all being compelled to make choices about virtualization technology, and I hope this article provides some assistance to you in selecting and using the most suitable technology. IIJ has been proactive about adopting and operating virtualization technology, and this has improved business efficiency and reduced costs. Another consideration that we must pay attention to is that of potential threats to supply chains, with those in recent memory including the global chip shortage and open source vulnerabilities as highlighted in the media. As a service provider, it is incumbent upon us to prepare a whole range of options that we can turn to in order to continue to provide stable services to our users even when corporate acquisitions or other such events mean that products we use are no longer available in the same way as before, and we will thus continue to keep up with advances in virtualization technology and the products in this space.

Looking ahead, we will continue to monitor the latest trends in technology and adapt as virtualization technology evolves so that we are able to deliver even greater value to our users.

2.1 Introduction / 2.2 History of Virtualization Technology / 2.3 Technology Selection and Service Development
Kaoru Tanaka
Technical Manager, Technology Development, Cloud Division, IIJ

2.4 Virtualization in Action at IIJ
Takehiro Yamamoto
General Manager, UAG Development Office, Cloud Division, IIJ

Yasuhide Takahashi
Deputy Head of Cloud Services Division 2, Cloud Division, IIJ

Kazuki Takai
Technical Manager, Technology Development, Cloud Division, IIJ

2.5 Conclusion
Shinri Kimura
Deputy Head of Cloud Division, IIJ