2. Cloud Computing Technology

The Evolution of IaaS on Cloud Native Infrastructure

Despite skepticism of cloud services in Japan, they began to slowly catch on here four or five years ago. In the short space of time since then, terms such as Cloud First have entered the vernacular and spread like wildfire, and it could be said that IaaS is close to reaching a certain level of perfection. In the coming years the main battlefield for the cloud business is likely to gradually shift to a higher level based on IaaS.

Against this backdrop, we released the new "IIJ GIO Infrastructure P2" IaaS on November 30, 2015 (hereinafter "P2"). While providing IaaS with high reliability and security, this will serve as a robust infrastructure for future service development. The new service was released after determining the need for a forward-thinking IaaS, as cloud computing now progresses to the next stage.

In this report, we will explain the technology used in this new IaaS infrastructure, approaching it from the concept of service.

■ Selectable Cloud Computing

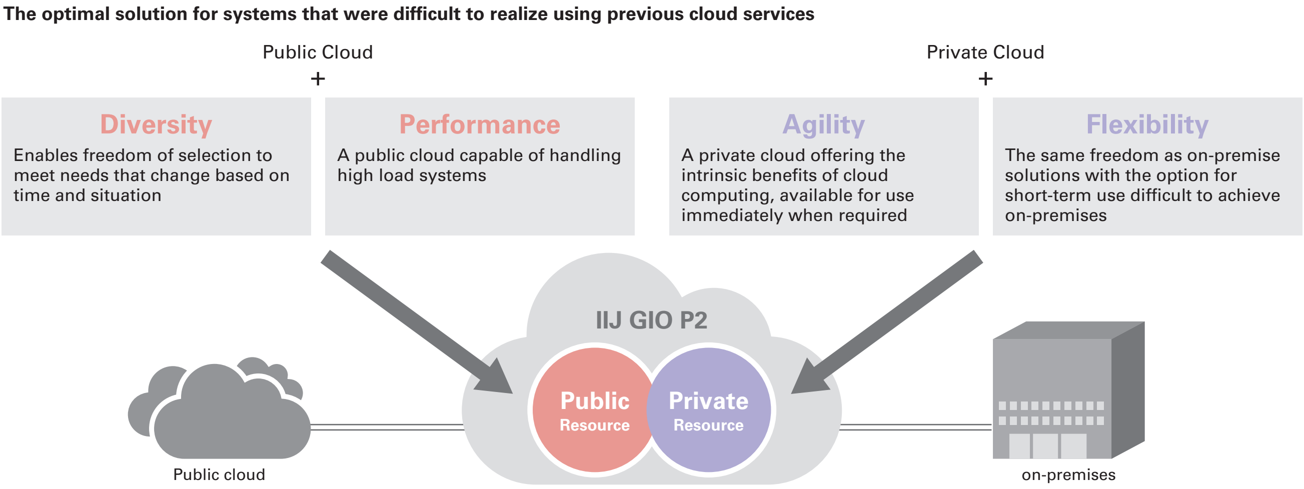
During the process of implementing cloud services, and in particular IaaS, many people have to consider at some point whether to use a public cloud, or construct a private cloud within their company.

Considering convenience and initial investment, there is no doubt that public cloud utilization will grow at an increasingly fast pace from now on. On the other hand, given issues with product license terms and compliance, and requirements for an extremely high level of stability and performance, a virtualized public cloud isn't always the best option. This is likely to remain true for some time, no matter how refined cloud computing becomes.

To resolve these concerns, the P2 service menu has two service lineups. The first is called public resources, which are made up of large-scale virtualized infrastructure. The second is called private resources, which consists of bare metal servers and storage so that tenants can have exclusive access to devices. Depending on the intended purpose, customers may only use one of these resources, but it is also possible to build a single system that combines both as you see fit. P2 is a single cloud service that integrates public resources and private resources (Figure 1).

■ Multiple Resource Types to Satisfy Both Cost and Efficiency

We provide an additional two resource types within the public resources category: performance-guarantee and best-effort types (Figure 2). The concept of providing these two resource types is incorporated into the service design consistently with regard to servers, storage, and networks.



The optimal solution for systems that were difficult to realize using previous cloud services

Enables freedom of selection to meet needs that change based on time and situation

A public cloud capable of handling high load systems

A private cloud offering the intrinsic benefits of cloud computing, available for use immediately when required

The same freedom as on-premise solutions with the option for short-term use difficult to achieve on-premises

Figure 1: Public Resources and Private Resources

Guaranteeing performance for a cloud service means securing the required amount of resources in a fixed manner, whether or not they will be used. For example, when allocating 100 Mbps of bandwidth to each tenant while guaranteeing network performance, ten tenants could be accommodated on a 1 Gbps network. Although guaranteeing bandwidth provides peace of mind due to the stable performance provided, it is known that in many cases that level of performance is not actually necessary, and tends to end up costing an excessive amount. When not using a cloud solution, both line and server performance should be fixed at the level determined at the design stage, so it could be said this type of solution guarantees performance whether wanted or not.

Meanwhile, the best-effort type is based on an entirely opposite concept. A 1 Gbps network is prepared in advance, then tenants numbers are controlled based on an estimate of how many can be housed while still maintaining acceptable performance. When doing this, it is expected that there will be a certain amount of variance in the network usage of each tenant. This maintains a smooth experience without allocating an equal amount of resources to all tenants. However, as only the maximum value is indicated in the specifications, the performance provided is not known until you actually use the service, and some people would feel uneasy about this. In fact, when sharing resources on a very small scale, even slightly disproportionate variance inevitably leads to performance degradation. That said, in an extremely large-scale resource pool like that of a cloud service, it is easier to anticipate resources will be allocated efficiently to make up for variance, because a huge number of tenants cohabit the space. Of course, to achieve this it is essential to have a system for reallocating resources dynamically, as this is not self-regulated.

So which service is better? It would be very appealing to have a cloud service that guaranteed performance, and was also available at a reasonable fixed price, but to achieve this would likely have to involve some kind of trick. As long as there is a limit to system resources, a trade-off must be made between cost, performance, and reliability. In that case, wouldn't being able to choose your own solutions according to application be best?

For example, let us consider implementing a cloud solution for an internal system. It should be rare for the system to have extreme differences between the performance required regularly and the performance required at peak times. A performance-guarantee type solution that offers stable performance and makes cost control easy is best suited to systems like this.

Meanwhile, for the cloud deployment of systems with extreme differences in system load between regular and peak times, such as Internet services, sizing based on peak periods is usually problematic from a cost perspective. In this case, it would be optimal to utilize a best-effort type solution set with a high performance cap in relation to the price, keeping costs down during regular operation, and increasing costs according to the performance exhibited at peak times.

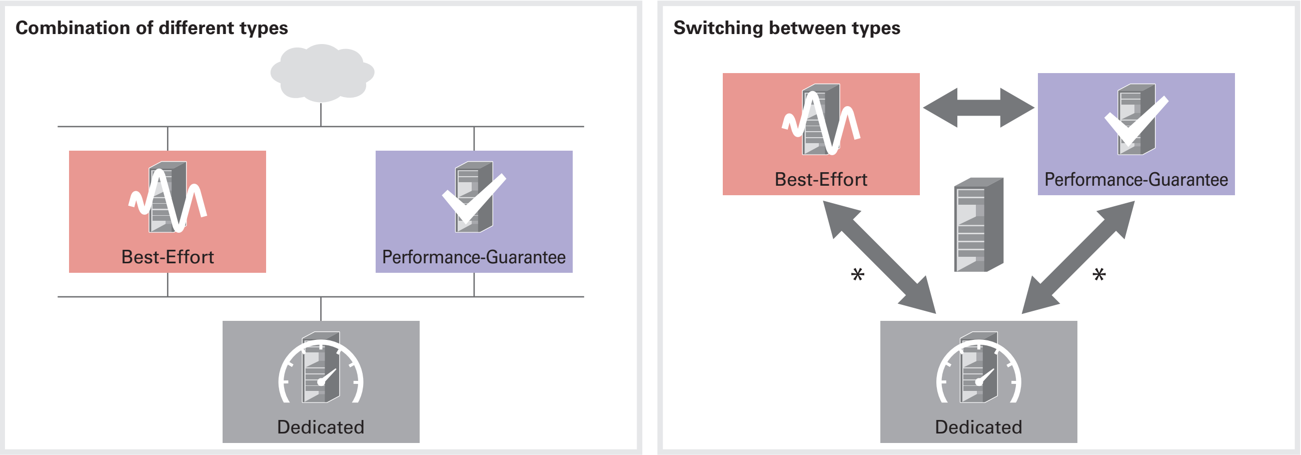


Figure 2: Performance-Guarantee and Best-Effort

■ A Cloud Solution Offering Overall Optimization Through Use of Resource Types According to Need

Although the term “utility computing” is not heard very often today, back in the early days of the cloud it was said to be one of the essential elements of cloud computing. This concept is based on using computing resources like utilities such as water, gas, and power.

To use water resources as an example, conventional system design is similar to digging a well in your back yard. The initial investment is large, and it takes time before it is ready to use, but afterwards water is available by simply drawing it from the well, and no additional costs are involved. On the other hand, the water may dry up in a drought, and it is necessary to implement your own measures to prevent water pollution. In contrast, cloud computing is like tap water. It offers stable and secure water resources unaffected by the weather, while also requiring no initial investment, and you only pay for what you use. It goes without saying that tap water is the superior option for the majority of users, but a portion of them may make a different decision if they have plentiful water sources of their own, or run a business that hinges upon good quality ground water.

The advent of cloud computing enabled on-demand, rapid changes to systems. However, systems with a conventional style also have an advantage in certain situations. Appropriate use of public and private solutions, as well as cloud-based resource types, is the key to overall system efficiency.

■ Cloud Native Infrastructure Creates Value from Scale

Here I would like to discuss the details of the P2 infrastructure that realizes this concept.

In 2009, we launched the IIJ GIO Service (hereinafter “GIO”), and went on to develop our cloud services. However, it cannot be said that this was a purely cloud-based infrastructure initially. Up until now, GIO was a combination of cloud services built upon traditionally-designed physical infrastructure. This functioned well, was stable, and gave customers a cloud service that offered convenience and peace of mind at the expense of fully-fledged cloud characteristics.

I believe this was a good configuration at the dawn of the cloud era, but it is somewhat lacking as an infrastructure for services offering the benefits of cloud computing that are required by the market today. In light of this, we utilized the knowledge we gained through our experience providing cloud solutions on a scale of tens of thousands of devices, and built the overhauled P2 cloud service on cloud native infrastructure.

■ SDN on Cloud Infrastructure

So what is cloud native infrastructure? One example of its characteristics is the software implementation of features conventionally carried out by hardware. In other words, the implementation of technologies such as Software-Defined Networking (SDN) and Software-Defined Data Centers (SDDC).

For a period of time, SDN created quite a stir alongside keywords such as network virtualization and OpenFlow, but I’d hazard a guess that there are people who sense its popularity has waned recently. It may not be evident when compared to server virtualization, which provides widely accepted benefits and is an obvious technology choice these days, but this drop in appeal is not because SDN had no value, but rather because this technology doesn’t provide broad universal advantages. It shows its true value when used in an environment that meets certain conditions, and cloud infrastructure falls under this category.

P2 networks are not actually all that unique when you look at their structure. If anything, you might say they are implemented using only technology that is as simple and universal as possible. However, they are distinguished by their scale and applications.

For example, in the past a system comprised of several hundred servers would have been considered large scale, but this is quite diminutive in cloud infrastructure terms. There are very few environments other than cloud solutions where servers, storage, and networks are tied together on such a monumental scale, and treated as a massive resource pool. This enormous scale means it has the potential to exceed a variety of upper limits that are normally assumed. These environments easily eclipse the scale expected for single physical devices, affecting everything from the maximum number of VLANs, to the maximum number of learnable MAC addresses, as well as bandwidth caps, which means it is necessary to implement SDN and control them using software.

The virtualized multi-tenant system also makes cloud infrastructure special. No matter how large a resource pool is, it is nothing but a mish-mash of small-scale single tenant systems if it is cut down into physical device units for allocation to tenants, falling short of the scalability and flexible system configuration that cloud computing enables. In contrast, cloud infrastructure creates massive multi-tenant systems that can accommodate any kind of tenant, regardless of the servers or storage on the network that is involved. That is why cloud infrastructure makes it possible to allocate the necessary resources on-demand, enabling swift deployment.

It was necessary to adopt SDN as we sought an economy of scale only achievable above a certain size (Figure 3).

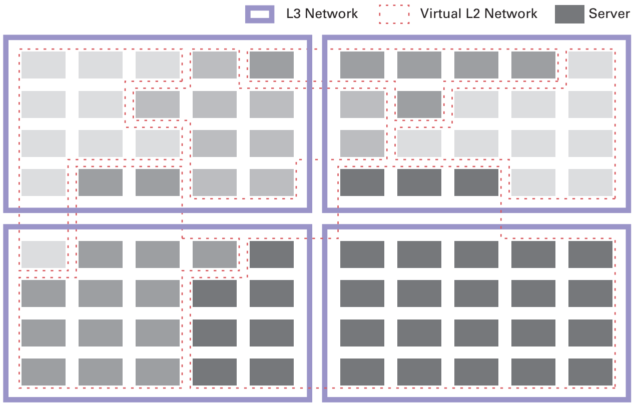
Lengthening Life Cycles

P2 primarily consists of an original software stack, although a large amount of OSS is utilized for middleware. The SDN control system that forms part of the platform is also implemented through coordination between an original system called SSP and the P2 orchestrator.

Cloud vendors sometimes implement proprietary infrastructure software like this because it has features not found elsewhere, but another reason is the fact that it frees you from the life cycles of the devices and software that a cloud solution is comprised of, making it possible to lengthen the life cycle of cloud infrastructure. This contributes greatly to the long-term stability of a cloud service.

In the IT industry, there seems to be an almost tacit understanding that systems are renewed in a cycle of every four or five years. However, these days the pace of device performance improvements is slowing compared to the past. On the other hand, they are constantly becoming more and more sophisticated, and the cost of replacement continues to rise, so in an increasing number of cases the aim is to extend the life cycle of a system as long as possible.

With regard to servers, the spread of virtualization technology is opening the door to system operation independent of physical environments. That said, network virtualization is not standard, and normally the control of networks depends on a specific



SDN technology allows free configuration of L2 networks within zones that expand across L3, enabling large-scale cloud infrastructure

Figure 3: SDN

product. In a single tenant system, there are few disadvantages as long as the life cycle for hardware, OS, and applications match up, but the same cannot be said for multi-tenant systems like cloud infrastructure. Because a variety of tenants each operate systems with their own life cycles, they will never match the life cycle of the cloud infrastructure. There is no doubt that continuing to offer services on a long-term basis will be an increasingly appealing factor for cloud services in the coming years.

The Technology Supporting Clouds With No Downtime

I mentioned that the P2 cloud infrastructure requires SDN because of its large scale, but I'd also like to touch upon the live migration technology that supports the operation of cloud infrastructure. Like SDN, live migration is not a technology peculiar to cloud infrastructure. In fact, almost all hypervisors feature this function, and many people use it on a daily basis. Of course, depending on the purpose, there may be many cases where live migration is not necessary, as the process can be carried out while the system is suspended.

On the other hand, live migration is an essential function for the infrastructure of cloud services. It is used to perform a range of maintenance to keep the cloud healthy, while making every effort to avoid impact to users.

One example of this is when a vulnerability is discovered in cloud infrastructure. When there is a serious vulnerability in multi-tenant infrastructure, it could have a wide-ranging effect, so it must be dealt with swiftly. In this case, the virtual machines are moved out using live migration, and the empty hypervisors are then updated safely.

Let us consider an example where traffic is concentrated on a portion of the virtual servers. Not all best-effort type virtual servers are equal, but even so, in this kind of extreme case the corresponding servers are isolated, and an effort made to limit the extent of impact on other tenants housed on the same equipment.

The Live Migration Mechanism

Live migration is a general term for technology that transfers virtual machines to other hypervisors without suspending them (Figure 4). However, it is deceptive to say they are not suspended at all, as in some cases a certain amount of downtime occurs. Allow me to explain the logic behind this briefly.

First, when live migration is executed, the following process is carried out.

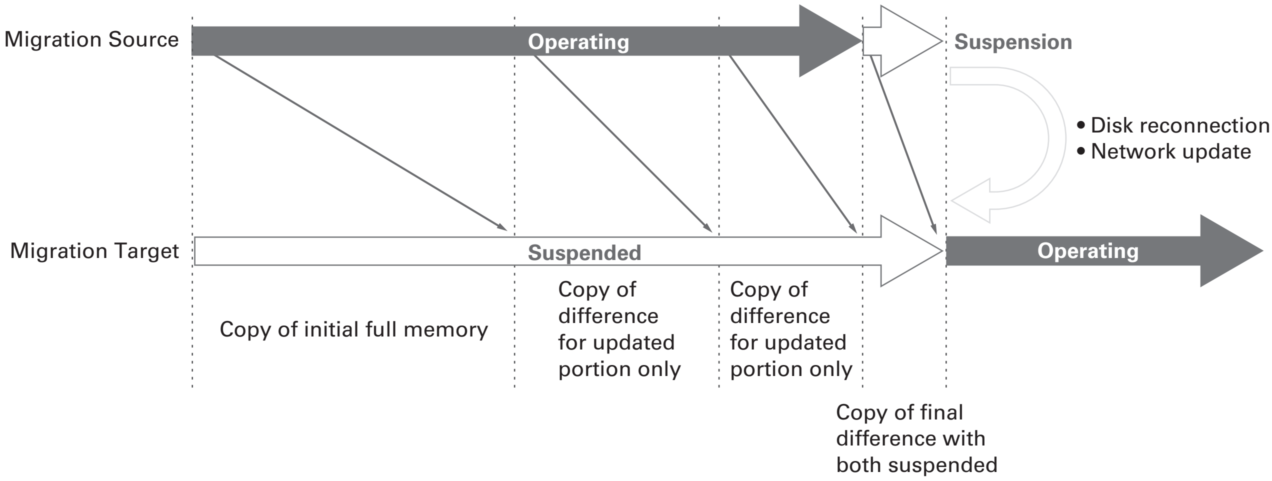

Migration Source
Operating
Suspension
• Disk reconnection
• Network update
Migration Target
Suspended
Operating
Copy of initial full memory
Copy of difference for updated portion only
Copy of difference for updated portion only
Copy of final difference with both suspended

Figure 4: The Live Migration Mechanism

1. Prepare the target virtual machine for migration
2. Transfer the virtual machine's entire memory
3. Suspend the virtual machine
4. Reconnect storage to the target hypervisor for migration
5. Transfer various virtual machine attributes such as CPU state
6. Resume operation of the target virtual machine for migration
7. Relearn the MAC address on the network switch

Basically all information on a virtual machine must be transferred, and this large volume of data is located in storage and memory. However, because the volume of storage data is so large, in virtual environments where live migration is used, all data is typically placed on remote storage connected via iSCSI or NFS. Conversely, on cloud services that offer servers with local disks installed, cloud vendors may opt to suspend the virtual server, as they receive no benefit from live migration.

Accordingly, it is almost always memory that determines the suspension time during live migration. For P2, virtual servers equipped with 48 GB of memory are available, and when transferring all this memory over a network, it takes almost a minute even under the best of conditions. If a virtual machine is suspended during this transfer time, it does not count as live migration. Consequently, the transfer of memory is first carried out with the virtual machine still running. In this case, the memory will of course continue to be updated during the transfer, so the difference between these states is sent next. This process is carried out a number of times as necessary, and when the difference is judged to be small enough, the virtual machine is suspended, and the final transfer is carried out. The volume of this final transfer determines the suspension time during live migration. In other words, live migration is influenced by the amount of memory that is updated, rather than simply how much memory the virtual machine has installed. This live migration mechanism enables the transfer of hypervisors with next to no suspension time at all.

That said, for systems where the memory is updated extremely frequently, there is also a chance that the transfer amount will not decrease no matter how many times the difference is sent, ultimately causing a certain amount of downtime to occur. From experience, database servers that regularly bear a heavy load are one of the systems that are incompatible with live migration, as their memory is updated frequently. In light of this, for P2 we sometimes ask for virtual servers to be suspended before carrying out scheduled maintenance. After suspending virtual servers, they will be housed in an updated hypervisor after restarting. That means live migration does not apply, and it is possible to avoid unexpected suspension.

Cloud infrastructure is a complex system, and I cannot exhaust the topic on the pages of this report, so here I have focused on networks and operation. If you are interested in the infrastructure of P2, I touched upon this subject a little in a presentation[*1] I gave at the "IIJ Technical WEEK 2015" IIJ technology event held last year. Please take a look if you would like more information.

Author:
Keisuke Taguchi
Mr. Taguchi is Manager of Platform Service Department 1, Platform Division, IIJ.
He joined the IIJ GIO Service cloud project from its planning stages in 2008.
He works as the public resources service manager for IIJ GIO Infrastructure P2.

[*1] IIJ Technical WEEK 2015, "Implementing SDN on the Overhauled IIJ GIO Cloud Infrastructure" 11 November, 2015, presentation material (http://www.iij.ad.jp/company/development/tech/techweek/pdf/151111_1.pdf) (in Japanese).