Challenges and Solutions in Storage on Service Infrastructure

3.1 Introduction

IIJ has used storage array systems in its service infrastructure since the 2000 launch of its resource-on-demand service IBPS, the predecessor to the IIJ GIO cloud service. Both IIJ GIO and NHN (Next Host Network), a cloud system for IIJ's own services, currently use storage array systems in their infrastructure, and the capacity of those storage systems is constantly being increased.

Here, we start by describing storage in general to give the reader a deeper understanding of what it actually is. We then discuss what sort of storage and storage networks IIJ employs as it strives to provide services that customers can rely on.

3.2 A General Overview of Storage

3.2.1 What is Storage?

Storage is a general term for anything that stores data and programs. Familiar consumer storage devices include the hard disk drives (HDDs) and solid state drives (SSDs) used in personal computers, USB thumb drives used to carry data around, SD cards inserted into smartphones and digital cameras, media used to hold music and video such as CDs and Blu-ray discs, and network-attached storage (NAS) devices sometimes referred to as network-compatible HDDs. If you've been using PCs for long enough, you may at some point have experienced an HDD or SSD failure that stopped your computer from working.

The enterprise storage used in IIJ's service infrastructure is designed for high availability. It contains multiple HDDs, SSDs, and other components to ensure that, even if one component fails, that failure will not result in data being lost or access to storage being unavailable. These sorts of storage systems are known as storage array systems, disk array systems, or enterprise storage, but we usually refer to them simply as "storage".

This section focuses on HDDs and SSDs used as storage components, and explains how storage array systems work and what the network that connects a storage array system and servers looks like.

3.2.2 Difference between HDDs and SSDs

Both HDDs and SSDs are classified as block storage, and when present in a PC or server, they are generally formatted with a file system supported by the installed OS before being put into use.

Inside HDDs, a disk coated with a magnetic material, called the platter, is rotated at high speed, and a magnetic head reads and writes data from the platter. Because the gap between the platter and the magnetic head is narrower than a human hair, HDDs are vulnerable to impact and can break down if moved during operation. They also contain mechanically driven parts such as motors, and the malfunction of these drive parts can also cause drive failures.

HDD speed is determined by the rotational speed of the platter. As you know, PCs sold up until only a few years ago used HDDs for storage, so it was normal for users to have to wait a long time for the OS, such as Windows, to boot up after turning the machine on. As such, HDDs can pose a bottleneck in modern systems. More recently, SSDs are often used to speed things up.

SSDs use semiconductor elements called flash memory. As they have no mechanical parts, they are quieter than HDDs, more resistant to impact, generate relatively little heat, and, above all, offer faster read/write speeds. SSDs tend to fail less often than HDDs, but SSDs do have a lifespan dictated by how many write cycles the flash memory can endure.

The difference between consumer and enterprise HDDs and SSDs is that enterprise products offer superior component

© Internet Initiative Japan Inc.

precision, durability, and lifespan. Some enterprise products apparently also undergo a pre-aging process before being shipped to weed out initial defects.

3.2.3 RAID

While not limited to HDDs and SSDs, potential causes of failure exist in both cases, and there is the possibility of data loss due to failure. To prevent such data losses, storage systems generally use technology called RAID (Redundant Array of Independent Disks) to increase data availability. There are several types of RAID, with RAID 1, RAID 5, and RAID 6 currently being the most commonly used.

RAID 1 mirrors data across at least two drives and offers data protection in the event one of the drives fails. RAID 5 requires at least three drives, with parity information (error correction code) distributed across the drives. Data is protected if one of the drives fails, and data can be corrupted if two drives fail. RAID 6 uses at least four drives, with two parity blocks distributed across all drives. Data is protected in the case of up to two concurrent drive failures, and data can be corrupted if three drives fail.

As an aside, RAID 0 also exists and is designed not for data protection but for increasing performance. RAID 0 is usually not used by itself but in combination with RAID 1, RAID 5, or RAID 6.

Typically, RAID creates a logical unit (LU) or volume out of all or part of the RAID area created in the disk device group and provides that unit or volume to the server as a disk device.

RAID also has a spare drive feature. A spare drive is one that normally remains unused. If a drive failure occurs, data is automatically migrated to the spare, or data is recomputed from the remaining drives and the parity information and restored to the spare drive. RAID spare drive functionality has been in widespread use for a long time, but a drawback of this configuration is that I/O operations are concentrated on the spare drive when a drive fails, resulting in a performance hit. Some recent storage array systems on the market therefore provide a spare for all drives in the system as a means of reducing the performance degradation.

With RAID now explained, let's look at a simple example of each RAID level based on a setup with six 4TB HDDs (no spare drives). Table 1 summarizes capacity and availability for each level.

You can see that RAID 1 is the least capacity efficient, next comes RAID 6, and then RAID 5 is the most capacity efficient. Given the efficiency and availability characteristics, IIJ's storage systems use RAID 5 or RAID 6.

3.2.4 Storage Array Systems

Storage array systems consist of a controller that controls the storage functionality along with various types of drives and power supply units. They are designed to have redundancy such that the system can continue to execute read and write requests even if one of the devices fails. Normally, a storage array system must be connected to servers and the like via a network to be usable, and the protocols used differ depending on the way in which data needs to be exchanged. Specifically, there are protocols for file access, protocols for block access, and protocols for object access such as Amazon AWS S3. Here, we focus on file access and block access.

RAID level	Capacity	Availability
RAID0	24TB	×
RAID1	12TB	✓
RAID5	20TB	✓
RAID6	16TB	✓

Table 1: Capacity and Availability at Different RAID Levels

© Internet Initiative Japan Inc.

3.2.5 File Access Protocols

First, let's discuss file servers (NAS) and other types of commonly known storage. Protocols used on NAS include the Network File System (NFS) used on UNIX/Linux operating systems, and the Server Message Block (SMB) and Common Internet File System (CIFS) protocols used on Windows. When configuring file storage on an ordinary server, a volume carved out of a RAID group is formatted, and the area is mounted and used on other servers using NFS or SMB sharing. Applications for file storage include storing web server content, sharing data between applications, and sharing files in an office setting.

3.2.6 Block Access Protocols

Next, we look at methods for using HDD- and SSD-like block storage over a network. Protocol types include iSCSI and Fiber Channel (FC) as well as the recently released NVMe over Fabrics (NVMe-oF). A network that mainly connects block storage and servers is called a Storage Area Network (SAN), with storage networks that use FC generally being referred to as FC-SANs and Ethernet storage networks using iSCSI being referred to as IP-SANs.

As a quick introduction, products using FC began appearing on the market in 1993, and iSCSI was published as an RFC by the IETF in 2003, with products using it starting to ship thereafter.

iSCSI is a standard that uses the SCSI storage protocol over TCP/IP, an Ethernet protocol. Dedicated hardware is not required, and it can be configured using the types of NICs installed as standard in Ethernet switches and servers. If you know a thing or two about storage technology, perhaps you are using iSCSI on your home network.

FC is a standard that uses the SCSI storage protocol over FC networks designed exclusively for storage. Dedicated hardware is required, the switches need to be FC switches, and the servers need to have FC-HBAs. Specialized knowledge of these products is also required.

3.3 Very Large FC-SANs

IIJ GIO Infrastructure P2 and IIJ GIO Infrastructure P2 Gen.2 use FC-SANs to connect block storage and servers.

As explained in Section 2, FC requires dedicated hardware and specialized knowledge, so the bar to adoption is generally thought of as being high, but having been building and operating FC-SANs in-house since 2003, IIJ has developed an understanding of the characteristics of FC-SANs and IP-SANs and the ability to choose appropriately between both depending on the system environment.

A reason for implementing an FC-SAN, for example, is that FC-SAN switches are equipped as standard with functions that are difficult to recreate on an IP-SAN.

3.3.1 Multi-tenancy

When accommodating multiple systems in one network, a common approach to ensuring security between systems with IP-SAN is to use Ethernet VLANs to logically divide the network. This is not particularly difficult to configure when connecting dedicated storage to a single system as shown in Figure 1. When connecting a single storage system to multiple systems as shown in Figure 2, however, the interface on the storage side needs to support multiple VLANs (VLAN Tagging, IEEE 802.1Q).

You might ask whether VLAN Tagging is well supported by storage array systems currently on the market. Many NAS products offer support, but only a handful of products for IP-SANs that use iSCSI and the like provide support, and even then there are restrictions on the number of VLANs, and it is often difficult to accommodate a large number of systems.

Let's turn to FC. Only protocols designed specifically for storage, not Ethernet protocols, are used on FC-SANs, and security between servers is maintained at the FC-SAN level. To enhance FC security, we block access to storage used by other systems using a feature called zoning, which links storage (target) and server (initiator) as shown in Figure 3.

© Internet Initiative Japan Inc.

And unlike with Ethernet, sharing a single storage device with multiple systems only requires you to configure zoning for each system, as shown in Figure 4, so there is no need to agonize over array system selection. Another point would be that using an FC-SAN eliminates the need for IP address design between the servers and the storage.

Also, in these examples, there is one switch for both FC-SAN and IP-SAN, but when there are multiple switches between the storage and the servers, VLAN needs to be configured for all of the switches in the case of IP-SAN, whereas FC-SAN provides a mechanism for the zoning information to be shared by all of the switches, so only one needs to be configured. So the configuration workload is lighter in the case of FC-SAN.

3.3.2 Lossless Networks

FC provides high performance, low latency, and lossless transmission as standard. If packet losses occur on an Ethernet network, data integrity is maintained using TCP's retransmission control scheme. In environments where storage packet losses occur frequently, storage performance can degrade and applications can experience fatal errors. So when setting up an IP-SAN with Ethernet, network design and other such factors require careful consideration.

Setting up a lossless Ethernet similar to FC can be done by selecting switches that make it possible to use Data Center Bridging (DCB) and other such functionality, but you also need to do additional configuration of the server NICs and storage DCB.

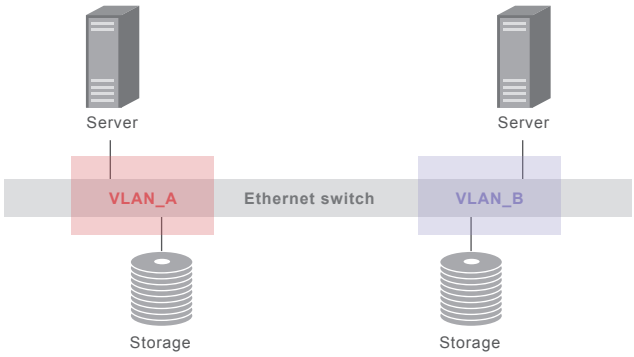

Figure 1: iSCSI Storage Dedicated Configuration

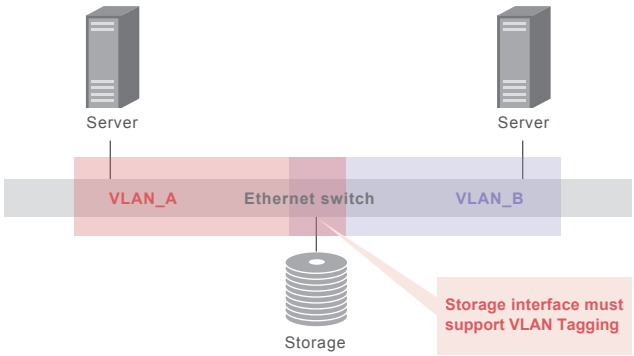

Figure 2: iSCSI Storage Sharing Configuration

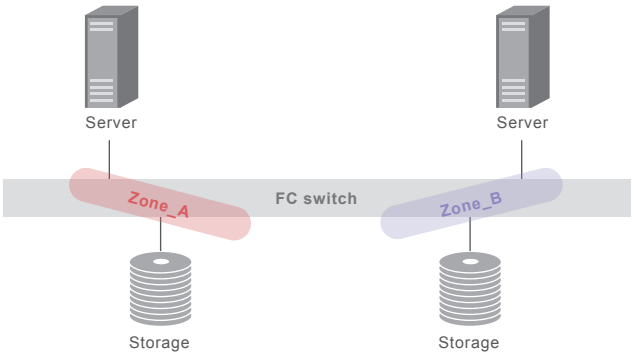

Figure 3: FC Storage Dedicated Configuration

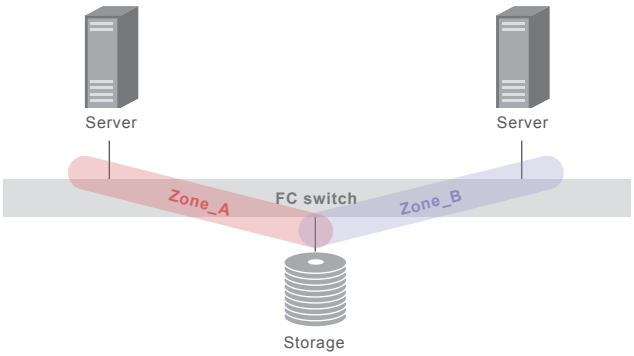

Figure 4: FC Storage Sharing Configuration

© Internet Initiative Japan Inc.

3.3.3 FC Fabric

With FC Fabric, the following types of fabric topologies can be created. In Figure 5, the servers and storage are connected by a single switch, but in reality you would use two sets of switches to ensure there are two routes between the servers and storage, meaning that a single fault will not cause the system to stop working.

First, let's consider the single-switch topology. IIJ initially used this setup because it allows you to start small. The drawback of this topology has to do with scalability; as more servers are added and storage is increased, you are forced into doing extensive maintenance if you try to scale up.

Figure 6 shows the Core-Edge, Edge-Core-Edge, and Full Mesh topologies.

The Core-Edge topology is effective when storage and servers are densely packed on a single floor of a datacenter. We use it for GIO P2 Gen.2 and some parts of GIO P2. Figure 6 makes it look like the FC switches are connected by a single cable, but they are actually connected by two or more to mitigate the impact if any one of them is disconnected. This uses functionality provided by FC switches called Inter-Switch Link (ISL). This is similar to Ethernet switch port channels, but unlike Ethernet switches, if an FC switch has an ISL Trunk license, the connections will

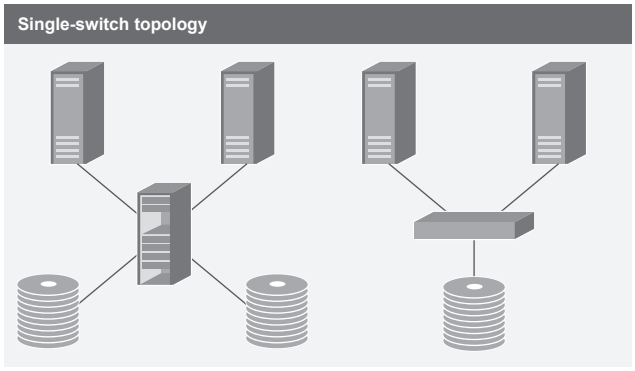

Figure 5: FC Fabric Topology with a Single Switch

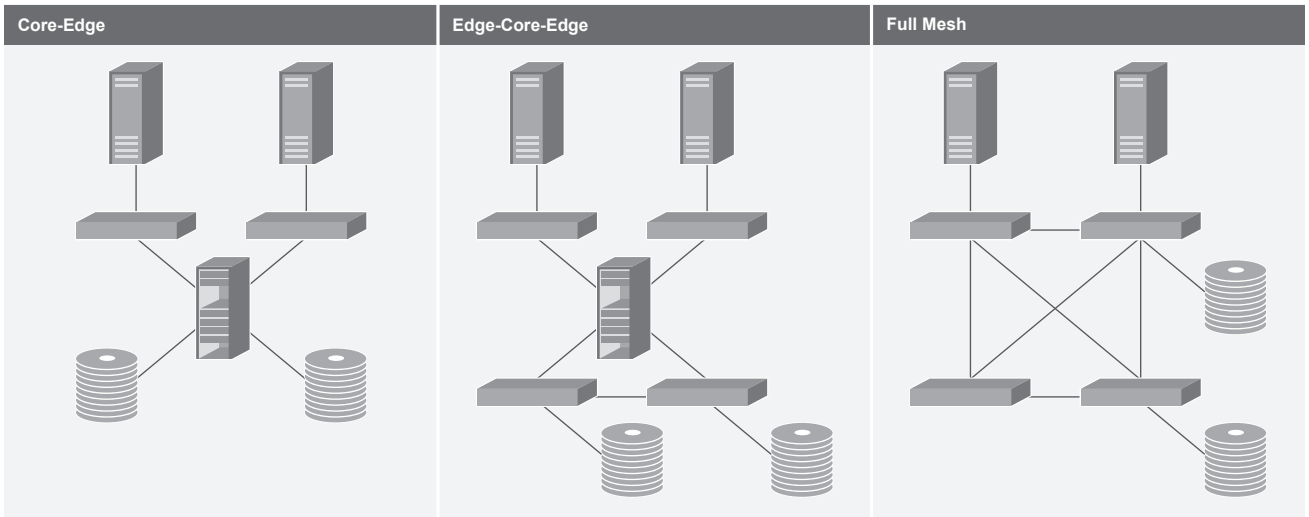

Figure 6: FC Fabric Topology with Multiple Switches

© Internet Initiative Japan Inc.

automatically be configured as a single trunk link. For example, if four 32Gbps FC switches are connected with ISL Trunk, the bandwidth between the switches is 32 x 4 = 128Gbps.

Edge-Core-Edge is used when the storage and the servers are installed separately on multiple floors of a datacenter. This can be, for example, when storage cannot be installed near the core, so an edge switch for storage needs to be set up to make the connection with the core. At large scales, an Edge-Core-Core-Edge topology can also be used for cores installed on different floors, and because a lot of traffic can be expected to travel between the cores in this case, we make an effort to design in a large number of ISL Trunks.

We use Full Mesh when datacenter racks contain a mix of storage devices and servers in a small-scale setup. It is worth noting that you often need to configure advanced settings when using Full Mesh with an ordinary Ethernet switch. With FC switches, the ability to automatically configure storage–server routes in an appropriate manner means that you do not need to do any special configuration, so the setup is very easy in this case.

3.3.4 FC Fabric Services

FC switches provide a fabric service to manage all devices connected to an FC-SAN and the information needed to connect servers and storage. To give just one example of a fabric service, if zoning between the storage and the servers is set up properly, the servers will automatically be able to connect to the storage without any special settings having to be configured on the server side. In the IP-SAN space, a similar feature called the Internet Storage Name Service (iSNS) exists, but implementations are limited and there are not many system environments in which it is used. The approach with IP-SAN, therefore, is for the servers to keep a record of the storage devices they connect to. Figure 7 summarizes the differences.

If you only have a few servers, for instance, there is not much difference between FC-SAN and IP-SAN in terms of the amount of work required to connect, but with dozens of servers or as many as 100 or so servers, the engineers managing the servers have a lot of work on their hands in the case of an IP-SAN because all of the servers need to have a record of the storage devices they connect to.

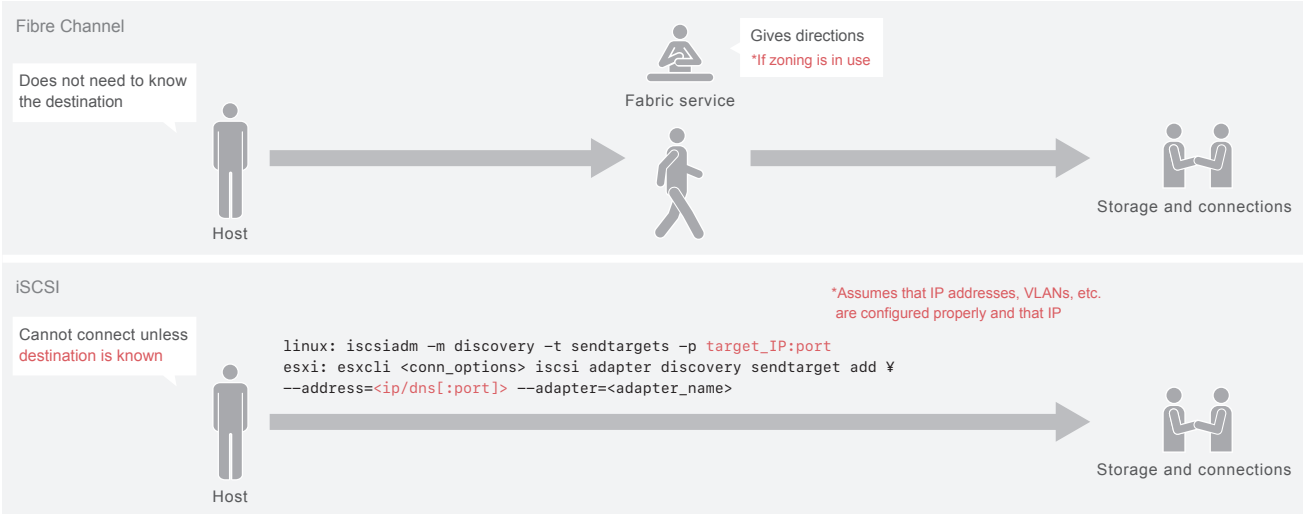

Figure 7: Difference between FC-SAN and IP-SAN Connections

© Internet Initiative Japan Inc.

3.3.5 Criteria for Selecting FC-SAN

On the operations side, FC switches provide the ability to upgrade firmware versions without stopping FC traffic as well as features to forcibly take ports offline if the error count on a port exceeds a certain level due to SPF+ module faults or the deterioration of cables, thereby preventing any further adverse impact on the FC Fabric.

In view of this, IIJ's decision criteria is as follows. Either FC-SAN or IP-SAN are fine if there is no requirement that a storage array system be used by multiple systems, and FC-SAN is preferable if there is a need for the array to be shared by multiple systems.

That said, this decision criteria could change in future. If a greater range of IP-SAN storage array system products that offer more flexible support for VLAN and the like were to become available, then one would expect the better choice to be IP-SAN over Ethernet, which offers transmission speeds of 100Gbps or more.

3.4 Operations Technology

IIJ takes a number of steps to ensure the stable operation of storage systems.

3.4.1 Automating Service Delivery

Up until around 2010, customer requests to use storage on IIJ GIO were accommodated by having an engineer whose main role is operating storage systems go in and manually change settings on the storage system and FC switches. Back then, this only had to be done once a week or so at most, which is fairly infrequent, so the engineers were quite capable of getting everything done in time manually.

After 2010, however, once a week turned into several times a week, and even several times a day. Anticipating that it would no longer be feasible to do everything manually, we designed and built systems that fully automate storage system configuration changes.

Automating the configuration of storage is easy with commercially available applications, but commercially available applications commonly offer more than just storage automation functionality (and they are expensive), so we decided to build the storage control part in-house.

Storage control is done in languages such as Python and Ruby. Normally when changing a storage configuration using such a language, it is preferable to use the Storage Management Initiative Specification (SMI-S) or a special API. But the storage products and FC switches we were using at the time did not by themselves support SMI-S or a suitable API. We would have needed separate (expensive) applications to do this, so our standard was to use the command line to manage storage.

The command line is great for when a human wants to provide input and see the output, but the output generally comes in a format that makes it very difficult to get at the appropriate values in the results using a programming language, so the programs we created had to employ fairly cumbersome text processing routines. Moreover, some of the equipment we were using would return a 0 exit status (command executed successfully) even if there were errors in the command or parameters, so we also had to add in our own error handling.

Firmware upgrades on the storage array systems and FC switches present the most trouble. Some products give slightly different command line output before and after a version upgrade, so we needed to run tests in a development environment before performing a version upgrade in the production system.

More recent storage and FC switch offerings are a lot easier to program than the products available back then. With increasing support for configuration management software such as Ansible, device API implementations are moving forward, and storage vendors are now providing modules that can be used to control storage using Python and the like.

© Internet Initiative Japan Inc.

3.4.2 Storage Infrastructure Monitoring

Storage system monitoring methods include using ping, syslog, SNMP, and the like to monitor the systems, as well as remote monitoring by the manufacturer using the manufacturer's own proprietary monitoring tools. Initially, we used the monitoring service provided by IIJ. Because the number of servers connected to any one storage system was small back then, it did not take long for the storage operations team to be made aware of any faults after they occurred.

As the scale of our operations increased, however, the number of servers connected to any one storage system also increased. As a consequence, IIJ's monitoring service would detect hundreds of alerts for any single fault occurring on those storage systems, and this meant that it was taking longer for the storage operations team to recognize faults.

We therefore decided to build our own monitoring system for the storage operations team that would provide storage system alerts directly to the team. This shortened failure detection times, which were in the tens of minutes, down to the level of a few minutes.

3.4.3 Data Migration

HDDs and SSDs installed in storage array systems start to develop faults more frequently after operating continuously for about five years. IIJ looks to replace each piece of storage array system equipment once it has been running for five years, and data must of course be migrated at such times.

In a vSphere environment, you can migrate data with little impact using Storage vMotion and the like, but in other environments, you need to migrate data using some other method. One way is to mount both the source and destination storage on the server or other device and migrate each file individually. The process is shown in Figure 8. In Windows, you can use drag-and-drop or robocopy, for instance, while the options on Linux include cp, tar, and rsync.

This is a very simple way to do it and something we did a lot at IIJ in the past, but in system environments where write operations are executed once only, a single migration run can result in inconsistencies at the level of the entire file repository. So when copying files, our approach is to perform the migration multiple times across several days.

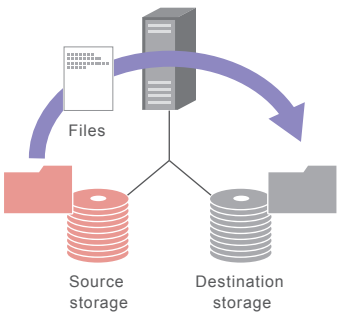

Figure 8: File-by-File Migration

© Internet Initiative Japan Inc.

In specific terms, we first do a copy of everything. Next, every business day, we do a copy of anything that has changed. We then single out directories that are frequently written to in parallel and, finally, we stop applications to prevent write operations and then copy only the files that have been written to.

Next, if we are using a logical volume manager on a physical server, we use the logical volume manager's functionality to perform the migration. The process is shown in Figure 9.

You need to have knowledge of the logical volume manager in this case, but data can be migrated with less of an impact than with the file-based migration method described above. And while there is a slight performance degradation for applications running on the storage system, the impact on performance is not huge. Specific examples here include the

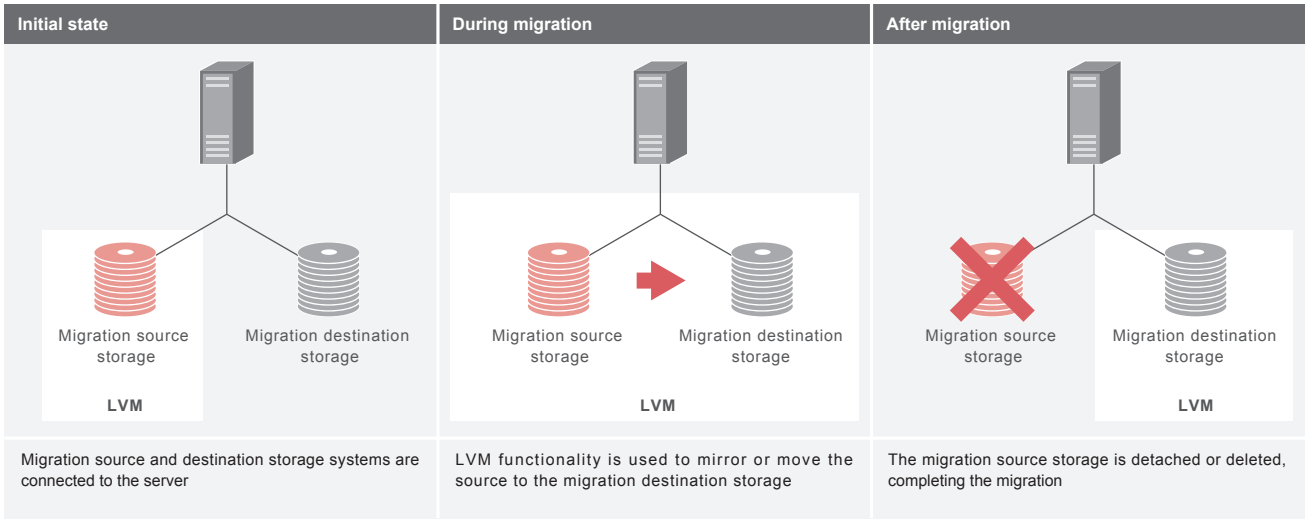

Figure 9: Migration Using Logical Volume Functionality

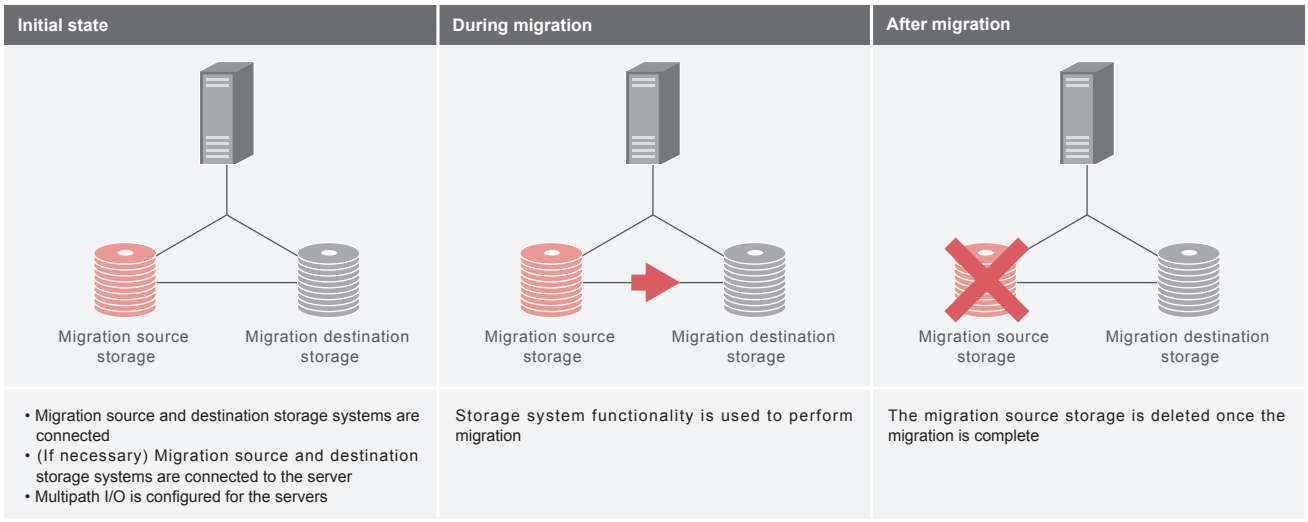

Figure 10: Migration Using Storage System Functionality

© Internet Initiative Japan Inc.

use of Logical Volume Manager (LVM) on Linux or HP-UX to temporarily mirror the migration source and destination storage device and, subsequently, disconnect the source once everything has been synchronized, and to limit the discussion to Linux, the use the LVM pvmove command to perform migration.

In environments that do not use a logical volume manager, however, these methods cannot be used, so the only choices are to migrate file by file, as mentioned above, or to use storage migration functionality as described below.

Finally, let's look at the use of storage migration functionality. Figure 10 shows how this works.

Some of the midrange-class storage systems and above these days have functionality allowing data to be migrated between storage systems of the same model or series of products, and some products allow one-way data migration to other vendors' products. Using this method makes it possible to accomplish all of your data migration regardless of the system environment in which the storage is being used. A point to note here is that this is premised on you using the same protocol on both the migration source and destination. That is, you cannot change the protocol used between the server and storage if, for example, you want to use FC on the migration source and iSCSI on the migration destination. Also, depending on the environment in which the storage is being used, you may need to stop the server around the time the migration is performed.

3.5 Conclusion

This report has introduced the reader to the way IIJ handles storage by looking at common storage functionality, going into a technical discussion of storage array and storage network adoption, and discussing storage operation technologies. IIJ will continue striving to ensure customers can always feel confident in relying on the storage they use.

Takahiro Kikuchi
Senior Engineer, Technology Development, Cloud Division, IIJ
After joining IIJ, Mr. Kikuchi's role involved designing, building, and operating storage infrastructure for IIJ cloud services. He is currently involved in surveys and research activities in the area of storage, the development of engineers both within and external to IIJ, and SINA Japan Forum working groups.

© Internet Initiative Japan Inc.