IIR

Internet
Infrastructure
Review

Feb.2021
Vol.49

Periodic Observation Report

Internet Trends as Seen from IIJ Infrastructure—2020

Focused Research (1)

Trends in Post-QuantumCryptography— 2020

Focused Research (2)

Query Service—The Challenge of Developing a Flexible Managed Database Service

IIJ
Internet Initiative Japan

Internet Infrastructure Review

February 2021 Vol.49

Executive Summary

Two news stories caught my eye while I was writing this summary on December 8, 2020. The first noted that a year has now passed since the first COVID-19 case was detected in Wuhan, China. The second was that COVID-19 vaccinations were to commence in the UK from today (December 8). The year 2020 has been synonymous with COVID-19, and finally at this late date, we have some news inspiring hope that the situation may eventually ease off.

Peoples' views and values seem to have changed significantly over this past year. It also feels like several years' worth of changes have happened in the space of a single year. Amid the new, radically different way of thinking, our ability to enjoy the conveniences we have always known without too much trouble obviously owes much to information and communications technology (ICT) including networking, security, and AI. On the other hand, COVID-19 has seriously affected the lives and health of many people. And many healthcare professionals continue to do their jobs on the front line in the face of the dangers. As members of the ICT industry, we hope to continue doing our part to bring ICT to bear in alleviating some of this hardship.

The IIR introduces the wide range of technology that IIJ researches and develops, comprising periodic observation reports that provide an outline of various data IIJ obtains through the daily operation of services, as well as focused research examining specific areas of technology.

Our periodic observation report in Chapter 1 is the 2020 edition of our look at Internet trends as seen from IIJ's infrastructure. The report covers IPv4 and IPv6 routes on the Internet, an analysis of DNS queries obtained from the full resolver provided by IIJ, IPv6 and mobile traffic, and the deployment of BGP ROV, which uses RPKI. We also discussed the impact of COVID-19 on Internet traffic in IIR Vol. 47 (https://www.iij.ad.jp/en/dev/iir/047.html) and Vol. 48 (https://www.iij.ad.jp/en/dev/iir/048.html), but our analysis here reveals some trends that differ from what we previously observed.

The focused research report in Chapter 2 looks at the latest developments in post-quantum cryptography. We previously covered post-quantum cryptography in our focused research report in IIR Vol. 31 (https://www.iij.ad.jp/en/dev/iir/031.html), and this issue provides an update. The report discusses the post-quantum cryptography competition that US-based organization NIST has been running since 2016, looks at the security of post-quantum cryptography, and explains the difference between quantum cryptography and post-quantum cryptography. As a CRYPTREC member, the report's author also relates key insights and episodes from the cryptography scene. This report should be an engaging read for anyone interested in the cryptographic technology that underpins the modern Internet.

In the focused research report in Chapter 3, the author describes his journey developing a query service that originated from his own ideas. While applications are able to use virtual infrastructure in containers to solve long-standing problems, this is not really an optimal solution for databases given the issues of data persistence, availability, and performance. To address this, the author designed and implemented a query service that provides the characteristic flexibility of containerized infrastructure while also ensuring data persistence and availability. Although it is still in the prototype phase, we are looking at turning it into a commercial service.

Through activities such as these, IIJ strives to improve and develop its services on a daily basis while maintaining the stability of the Internet. We will continue to provide a variety of services and solutions that our customers can take full advantage of as infrastructure for their corporate activities.

Junichi Shimagami

Mr. Shimagami is a Senior Executive Officer and the CTO of IIJ. His interest in the Internet led to him joining IIJ in September 1996. After engaging in the design and construction of the A-Bone Asia region network spearheaded by IIJ, as well as IIJ's backbone network, he was put in charge of IIJ network services. Since 2015, he has been responsible for network, cloud, and security technology across the board as CTO. In April 2017, he became chairman of the Telecom Services Association of Japan MVNO Council.

Internet Trends as Seen from IIJ Infrastructure —2020

Internet services provider IIJ operates some of the largest network and server infrastructure in Japan. Here, we report on Internet trends over the past year based on information obtained through the operation of this infrastructure. We analyze changes in trends from the perspective of BGP routes, DNS query analysis, IPv6, and mobile. We also discuss conditions observed before the deployment of BGP ROV on the IIJ backbone.

Topic 1

BGP Routes

We start by looking at IPv4 full-route information advertised by our network to other organizations (Table 1) and the number of unique IPv4 addresses contained in the IPv4 full-route information (Table 2). During the past year, RIPE NCC and LACNIC completely exhausted their IPv4 address pools. Only APNIC and AfriNIC remain, but APNIC has already projected that it will run out at the beginning of 2021, and in January 2020 AfriNIC placed a size limit (/22) on allocations/assignments.

The increase in the number of routes per year has been trending downward since peaking in 2018, but the total number now exceeds 800,000. The /22 prefix now accounts for over 100,000 routes, but the proportion of routes accounted for by the /22, /23, and /24 prefixes rose only slightly to 80.9% of the total.Meanwhile, the number of unique IPv4 addresses, although increasing vs. 2019, when it fell for the first time since 2011, remains below its 2018 and 2017 levels. Whether 2018 will turn out to have been the peak here as well will bear watching ahead.

Next we take a look at IPv6 full-route data (Table 3). In November 2019, ARIN received its first additional /12 block allocation (the second such allocation since RIPE NCC received one in June 2019).

The total number of routes increased by about the same amount as in the previous year and now exceeds 90,000. We expect it to surpass 100,000 in 2021. Also, 50.0% of all routes and 58.1% of those constituting the increase are

Table 1: Number of Routes by Prefix Length for Full IPv4 Routes

Date	/8	/9	/10	/11	/12	/13	/14	/15	/16	/17	/18	/19	/20	/21	/22	/23	/24	Total
Sep. 2011	19	12	27	81	233	457	794	1407	11909	5907	9885	19515	26476	26588	35515	34061	190276	363162
Sep. 2012	19	14	29	84	236	471	838	1526	12334	6349	10710	20927	30049	31793	42007	39517	219343	416246
Sep. 2013	16	11	30	93	250	480	903	1613	12748	6652	10971	22588	32202	34900	48915	42440	244822	459634
Sep. 2014	16	12	30	90	261	500	983	1702	13009	7013	11659	24527	35175	37560	54065	47372	268660	502634
Sep. 2015	18	13	36	96	261	500	999	1731	12863	7190	12317	25485	35904	38572	60900	52904	301381	551170
Sep. 2016	16	13	36	101	267	515	1050	1767	13106	7782	12917	25229	38459	40066	67270	58965	335884	603443
Sep. 2017	15	13	36	104	284	552	1047	1861	13391	7619	13385	24672	38704	41630	78779	64549	367474	654115
Sep. 2018	14	11	36	99	292	567	1094	1891	13325	7906	13771	25307	39408	45578	88476	72030	400488	710293
Sep. 2019	10	11	37	98	288	573	1142	1914	13243	7999	13730	25531	40128	47248	95983	77581	438926	764442
Sep. 2020	9	11	39	100	286	576	1172	1932	13438	8251	14003	25800	40821	49108	101799	84773	473899	816017

Table 2: Total Number of Unique IPv4 Addresses in Full IPv4 Routes

Date	No. of IPv4 addresses
Sep. 2011	2,470,856,448
Sep. 2012	2,588,775,936
Sep. 2013	2,638,256,384
Sep. 2014	2,705,751,040
Sep. 2015	2,791,345,920
Sep. 2016	2,824,538,880
Sep. 2017	2,852,547,328
Sep. 2018	2,855,087,616
Sep. 2019	2,834,175,488
Sep. 2020	2,850,284,544

/48 routes, from which we infer that IPv6 is also rolling out smoothly on end sites.

Lastly, let's also take a look at IPv4/IPv6 full-route Origin AS figures (Table 4). In the past year, an additional 3072 32-bit-only AS numbers were allocated to RIPE NCC and 2048 to LACNIC.

Both the decrease in 16-bit Origin Autonomous System Numbers (ASNs) and the increase in 32-bit-only Origin ASNs were around the same as those in the previous year, and 32-bit-only ASNs now account for 40% of all Origin ASNs. IPv6-enabled ASNs, which advertise IPv6 routes, continue to rise and now account for 28.1% of the total. Whether this will far exceed 30% or not in 2021 will be a point to watch.

Topic 2

DNS Query Analysis

IIJ provides a full resolver to enable DNS name resolution for its users. In this section, we discuss the state of name resolution, and analyze and reflect upon data from servers provided mainly for consumer services, based on a day's worth of full resolver observational data obtained on September 30, 2020.

The full resolver starts by looking at the IP address of an authoritative name server for the root zone (the highest level zone), which it queries, and then goes through other authoritative nameservers to find the records it needs. Queries repeatedly sent to the full resolver can result in increased load and delays, so the information obtained is cached, and when the same query is received again, the response is sent from the cache. Recently, DNS-related functions are implemented on devices that lie on route paths, such as broadband routers and firewalls, and these devices are sometimes involved in relaying DNS queries and applying control policies. Some applications, such as Web browsers, also have their own implementations of name resolver functionality and in some cases resolve names without relying on OS settings.

Table 3: Number of Routes by Prefix Length for Full IPv6 Routes

Date	/16-/28	/29	/30-/31	/32	/33-/39	/40	/41-/43	/44	/45-/47	/48	Total
Sep. 2011	68	13	22	3530	406	248	45	87	95	2356	6870
Sep. 2012	102	45	34	4448	757	445	103	246	168	3706	10054
Sep. 2013	117	256	92	5249	1067	660	119	474	266	5442	13742
Sep. 2014	134	481	133	6025	1447	825	248	709	592	7949	18543
Sep. 2015	142	771	168	6846	1808	1150	386	990	648	10570	23479
Sep. 2016	153	1294	216	8110	3092	1445	371	1492	1006	14291	31470
Sep. 2017	158	1757	256	9089	3588	2117	580	1999	1983	18347	39874
Sep. 2018	168	2279	328	10897	4828	2940	906	4015	2270	24616	53247
Sep. 2019	192	2671	606	12664	6914	3870	1566	4590	4165	34224	71462
Sep. 2020	205	3164	641	14520	9063	4815	2663	5501	4562	45160	90294

Table 4: IPv4/IPv6 Full-Route Origin AS Numbers

ASN	16-bit (1–64495)					32-bit only (131072–4199999999)				
Advertised route	IPv4+IPv6	IPv4 only	IPv6 only	Total	(IPv6 -enabled)	IPv4+IPv6	IPv4 only	IPv6 only	Total	(IPv6 -enabled)
Sep. 2011	4258	32756	115	37129	(11.8%)	90	1278	13	1381	(7.5%)
Sep. 2012	5467	33434	125	39026	(14.3%)	264	2565	17	2846	(9.9%)
Sep. 2013	6579	34108	131	40818	(16.4%)	496	3390	28	3914	(13.4%)
Sep. 2014	7405	34555	128	42088	(17.9%)	868	4749	55	5672	(16.3%)
Sep. 2015	8228	34544	137	42909	(19.5%)	1424	6801	78	8303	(18.1%)
Sep. 2016	9116	33555	158	42829	(21.7%)	2406	9391	146	11943	(21.4%)
Sep. 2017	9603	32731	181	42515	(23.0%)	3214	12379	207	15800	(21.7%)
Sep. 2018	10199	31960	176	42335	(24.5%)	4379	14874	308	19561	(24.0%)
Sep. 2019	10642	31164	206	42012	(25.8%)	5790	17409	432	23631	(26.3%)
Sep. 2020	11107	30374	229	41710	(27.2%)	7653	19668	574	27895	(29.5%)

ISPs notify users of the IP address of full resolvers via various protocols, including PPP, DHCP, RA, and PCO, depending on the connection type, and they enable automatic configuration of which full resolver to use for name resolution on user devices. ISPs can notify users of multiple full resolvers, and users can specify which full resolver to use, and add full resolvers, by altering settings in their OS, browser, or elsewhere. When more than one full resolver is configured on a device, which one ends up being used depends on the device's implementation or the application, so any given full resolver is not aware of how many queries a user is sending in total. When running full resolvers, therefore, this means that you need to keep track of query trends and always keep some processing power in reserve.

Observational data on the full resolver provided by IIJ show fluctuations in user query volume throughout the day, with volume hitting a daily trough of about 0.06 queries/sec per source IP address at around 4:30 a.m., and a peak of about 0.24 queries/sec per source IP address at around 12:30 p.m. These values are almost the same as last year, with a slight increase of 0.01 points from early morning into the daytime.

Broken down by protocol (IPv4 and IPv6), IPv4 queries per IP address fell vs. the previous year. While there is no major change in the middle of the night, we observe up to a 0.03-point drop in the daytime and from evening through nighttime. Meanwhile, IPv6 queries per IP address are up by around 0.03 points across all times of day, including late night. This suggests that IPv6-capable devices are gradually making their way into the home and that existing devices are being replaced. And looking at total query count, both the number of source IPs and the number of actual queries are higher for IPv6 than for IPv4. The number of IPv6-based queries is on the rise, accounting for around 63% of the total, up by 3 points from 60% in the previous year.

Recent years have seen a tendency for queries to rise briefly at certain round-number times, such as on the hour marks in the morning. The number of query sources also increases, with a particularly noticeable pattern around 7 a.m., which is possibly due to tasks scheduled on user devices and increases in automated network access that occur when devices are activated by, for example, an alarm clock function. In the previous year, we noted increases in queries 14 seconds and 10 seconds before every hour mark, and the 2020 results also show another increase 20 seconds before every hour. The increase in queries that occurs on the hour tapers off gradually, but with the spikes that occur before the hour mark, query volume quickly returns to roughly where it had been. Hence, because a large number of devices are sending queries in almost perfect sync, we surmise that lightweight, quickly completed tasks of some sort are being executed.

For example, there are mechanisms for completing basic tasks, such as connectivity tests or time synchronization, before bringing a device fully out of sleep mode, and we posit that the queries used for these tasks are behind the spikes.

Looking at the query record types, most are A records that query the IPv4 address corresponding to the host name and AAAA records that query IPv6 addresses. The trends in A and AAAA queries differ by IP protocol, with more AAAA record queries being seen for IPv6-based queries. Of IPv4-based queries, around 79% are A record queries and 15% AAAA record queries (Figure 1). With IPv6-based queries,

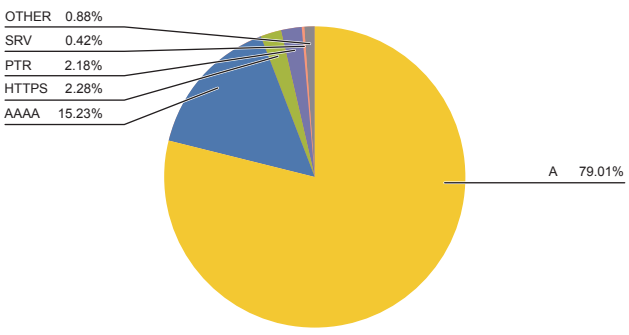

Figure 1: IPv4-based Queries from Clients

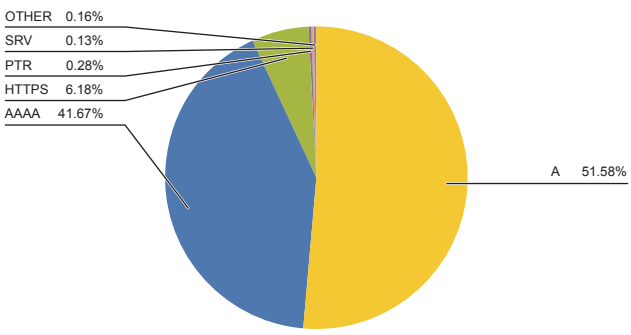

Figure 2: IPv6-based Queries from Clients

meanwhile, AAAA record queries account for a higher share of the total, with around 51% being A record and 41% being AAAA record queries (Figure 2). Compared with the previous year, we observe drops in A record queries of 5 percentage points for IPv4 and 3 percentage points for IPv6. HTTPS-type records, newly implemented in 2020, accounted for some 2% of IPv4 and 6% of IPv6 queries, coming in behind the A and AAAA query volumes. Systems currently supporting DNS over HTTPS include Apple's iOS 14, and we expect these queries to rise gradually as implementations spread.

Topic 3

IPv6

In this section, we report on the volume of IPv6 traffic on the IIJ backbone, source ASNs, and the main protocols used.

Traffic

As before, we again present IPv4 and IPv6 traffic measured using IIJ backbone routers at core POPs (points of presence—Tokyo, Osaka, Nagoya), shown in Figure 3. The data span the year from October 1, 2019 to September 30, 2020.

Traffic trends in 2020 differed from what we observed up till the previous year, with COVID-19 being a factor from the year's outset. Although no major changes were apparent until about February, IPv4 traffic increased substantially from March as COVID-19 prompted school closures and Japan's state of emergency declaration, resulting in people staying home. As discussed in Vol. 48 (https://www.iij.ad.jp/en/dev/iir/048.html), mobile traffic fell during the stay-at-home period, and fixed broadband services and corporate VPN services saw increases, which is likely behind the increase in IPv4 traffic.

To see the relative increases and decreases during the observation period, we graphed normalized IPv4 and IPv6 traffic with the values for the first day (October 1, 2019) indexed to 1 (Figure 4). The middle of the graph around April and May corresponds to Japan's state of emergency, when people were staying home. IPv4 traffic was up about 8% during the period, while IPv6 traffic looks to have fallen. Once the state of emergency ended, the increase in IPv4 traffic settled down to a level representing a slight increase vs. the start of the observation period. IPv6 traffic also ultimately saw a mild increase.

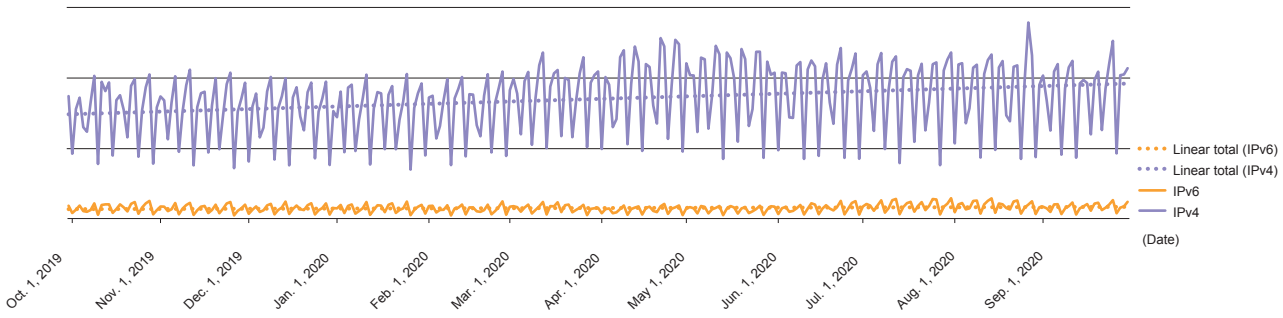

Figure 3: IPv4 Traffic and IPv6 Traffic (Oct. 2019 – Sep. 2020)

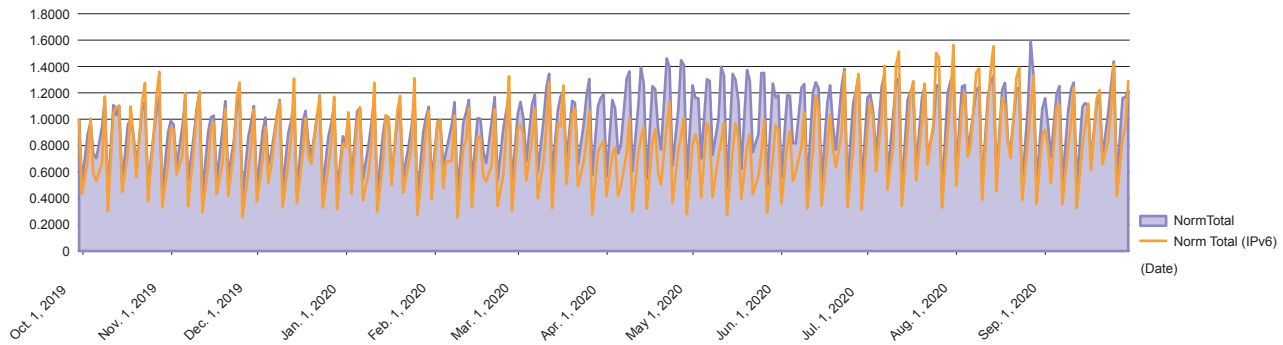

Figure 4: IPv4 Traffic and IPv6 Traffic, Indexed to Initial Value of 1

As Figure 5 shows, IPv6 declined as a percentage of the total during the state of emergency, but it eventually returned to about its initial level.

Traffic Source Organization (BGP AS)

Next, Figures 6 and 7 show the top annual average IPv6 and IPv4 traffic source organizations (BGP AS Number) for the year from October 2019 through September 2020.

Company A retains the top spot, but its share of traffic is down 5 percentage points since last time we reported. Traffic with IIJ's ASN as the source has grown substantially, and while this could be due to peculiarities of the observation point, the main factor is probably the growth in IPv6 video streaming traffic on JOCDN's platform, like last year. We observed no other major shifts or noticeable trends.

Protocols Used

Figure 8 plots IPv6 traffic according to protocol number (Next Header) and source port number, and Figure 9 plots IPv4 traffic according to protocol number and source port number (for the week starting Monday, October 5, 2020).

In the IPv6 space, TCP 443 (HTTPS) came in at No. 1 and UDP 43 (QUIC) at No. 2, so over 80% was attributable to HTTP encryption protocols. Of particular note this time around, TCP 80 (unencrypted HTTP) fell to No. 4 and ESP (IPSec encryption) came in at No. 3. ESP is observed more during the daytime on weekdays and is scarce on weekends, which probably indicates an increase in IPv6 VPN usage on corporate networks. It accounts for some 19% of traffic during its most prominent period around noon on weekdays, and even eclipses QUIC during the daytime.

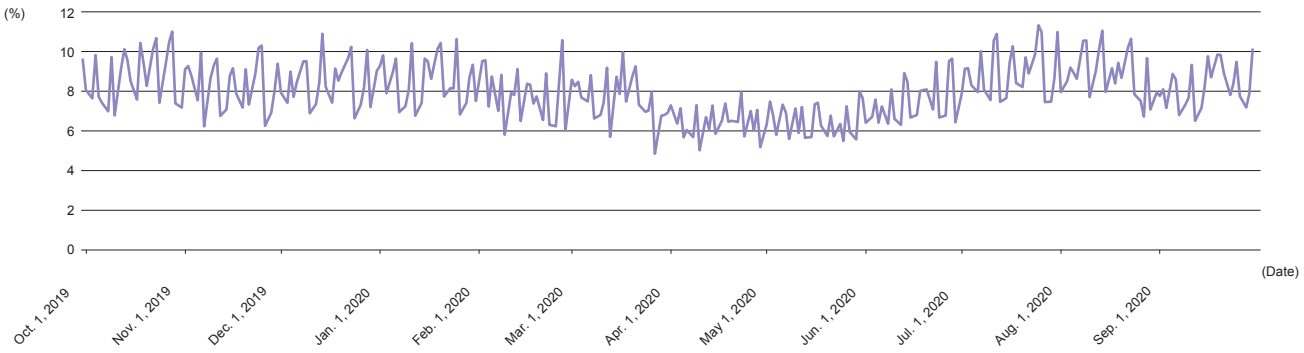

Figure 5: IPv6 as a Proportion of Total Traffic

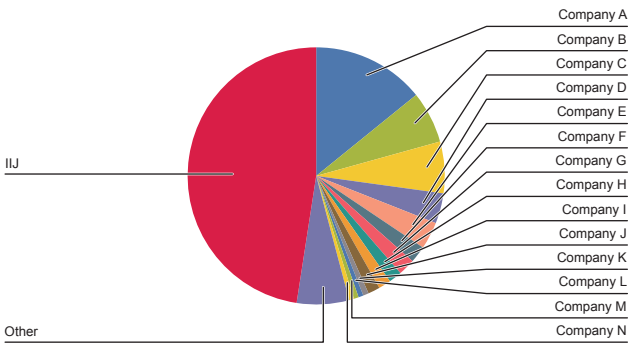

Figure 6: Top Annual Average IPv6 Traffic Source Organizations (BGP AS Number)

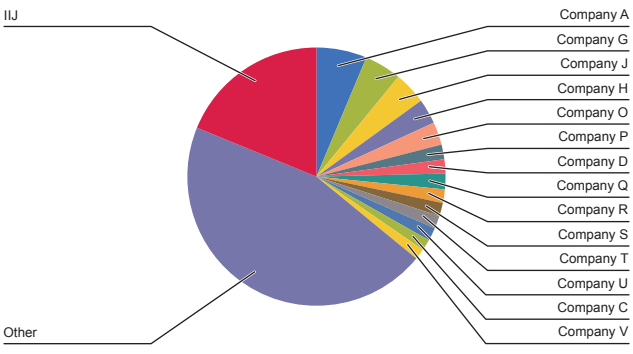

Figure 7: Top Annual Average IPv4 Traffic Source Organizations (BGP AS Number)

Only the top 5 are readily discernible on the graphs. Traffic is scarcer for No. 6 on down and does not show up readily on the graphs.

In the IPv4 space, protocol usage does not appear to have changed much, but nighttime peak traffic has fallen slightly, and daytime traffic appears to have increased overall. With the increase in people remoting into work from home during the day, it looks like daytime traffic has increased or that the data transfer peak has shifted to an earlier time. The shape of the weekday (peaks 1–5) and the weekend (peaks 6 and 7) sections of the graph are very similar, so the differences between weekday and weekend traffic trends appear to have diminished.

Summary

In this issue, we examined IPv6 traffic volume, source ASNs, and protocols used. With the impact of COVID-19, we observed different trends than in the past. IPv6 traffic as a percentage of total usage ultimately did not change much vs. last time, but the impact of people staying at home in the middle of the year was apparent. We observed new trends likely due to remote working in the IPv6 protocol usage figures, the IPv4 traffic peaks, and so on.

The COVID-19 pandemic is yet to show signs of winding down fully. As we move through what are being called the with-COVID and post-COVID eras, we will continue to monitor changes in IPv6 and Internet usage patterns.

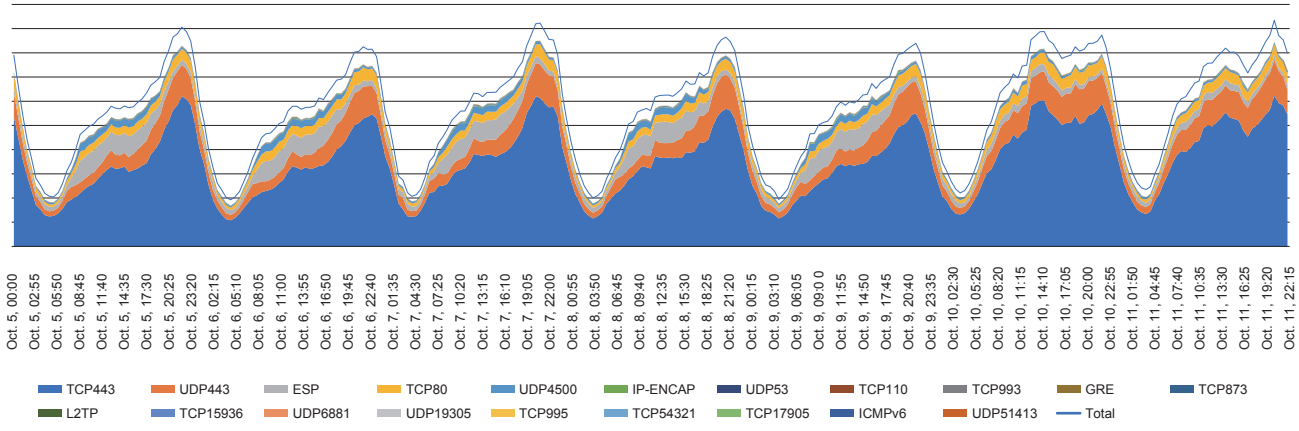

Figure 8: Breakdown of IPv6 Traffic by Protocol Number (Next Header) and Source Port Number

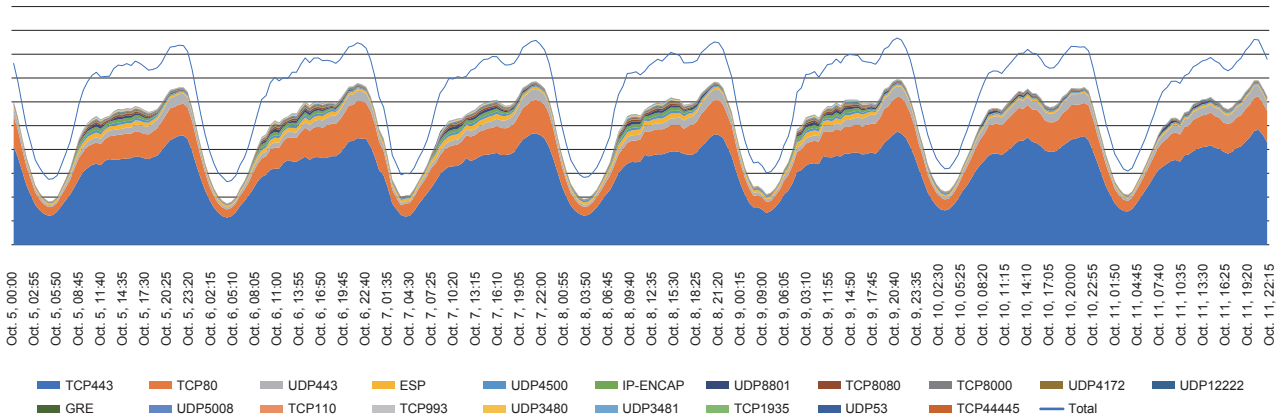

Figure 9: Breakdown of IPv4 Traffic by Protocol Number and Source Port Number

Topic 4

Mobile 3G and LTE

The trends in mobile traffic in 2020, as discussed in Vol. 48 (https://www.iij.ad.jp/en/dev/iir/048.html), were different from usual, with traffic down substantially during periods when people were staying home amid the COVID-19 situation. The term 5G has also gone mainstream in the mobile space, with MNOs launching 5G services, and on October 30 IIJ also began offering a service that supports au 5G as part of the IIJ mobile services that it provides to business customers. While new standards frameworks are gaining traction, old standards are heading for end of life. In October 2019, NTT Docomo announced that its 3G FOMA service will close down at end-March 2026.

In this edition, we look at 3G traffic on IIJ's mobile services based on observations for the period from October 1, 2019 to September 30, 2020.

Figure 10 shows 3G as a proportion of total traffic.

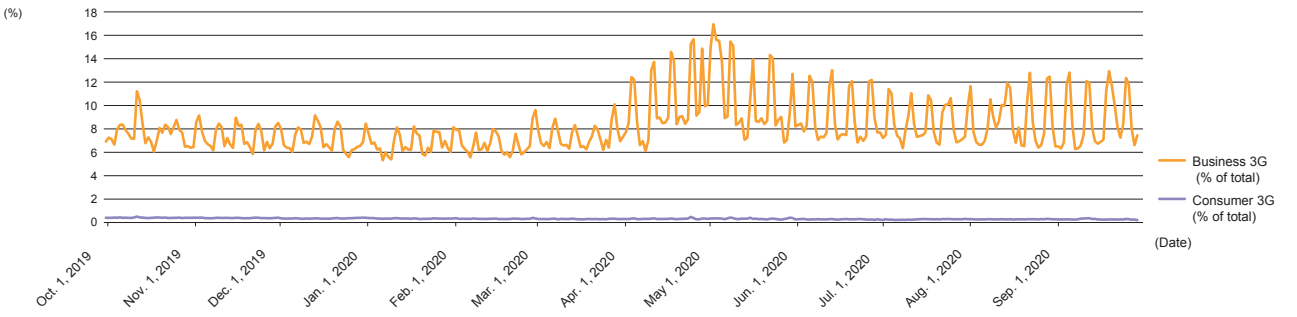

Figure 10: 3G as a Proportion of Total Traffic

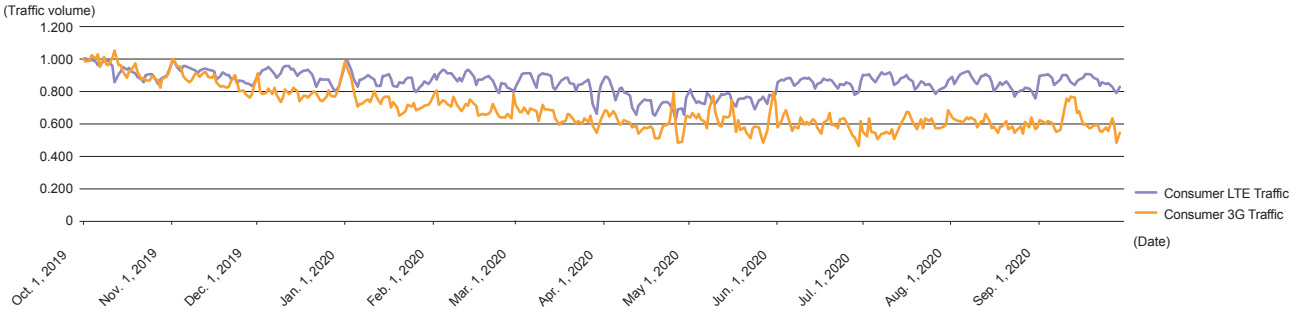

Figure 11: Consumer Service Traffic Volume

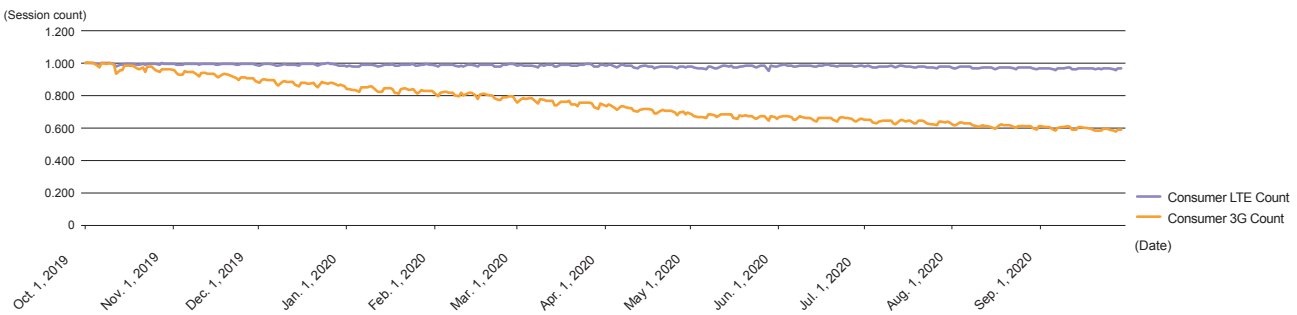

Figure 12: Consumer Service Session Count

On consumer services, 3G is barely used, accounting for under 0.5% of traffic. On business services, meanwhile, it averages around 8% of traffic, indicating that 3G remains well embedded in the corporate space.

Next, we graph trends in consumer service traffic volume (Figure 11) and session count (Figure 12) indexed to October 1, 2019. The figures for traffic volume and session count for 3G on consumer services continue to decline, with both having fallen by around 40% over the past year. One can think of various reasons for this, but given that almost all devices on consumer services are now smartphones, a reasonable explanation is that with the improvements in LTE connectivity at large, connections rarely drop to 3G anymore. We will be keeping tabs on how this unfolds ahead.

Turning to LTE traffic on consumer services, the session count remained largely flat, while traffic volume experienced a lull March through May 2020, when it was down 30%, probably because of people staying home amid the COVID-19 situation.

Next, we look at trends in business service traffic volume (Figure 13) and session count (Figure 14), also indexed to October 1, 2019.

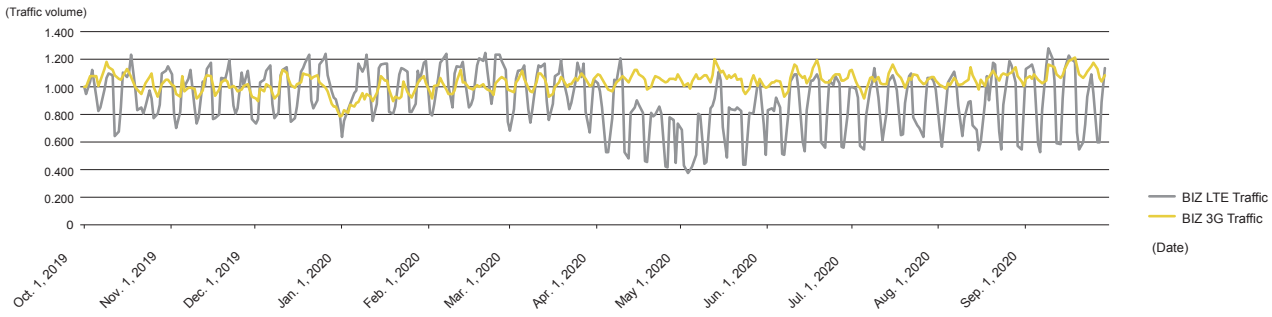

Figure 13: Business Service Traffic Volume

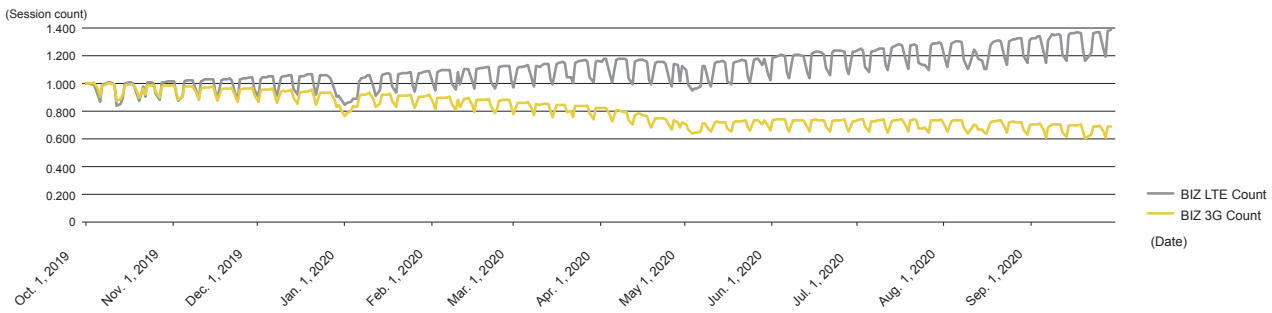

Figure 14: Business Service Session Count

Session count for 3G traffic on business services is in an intermittent downtrend. One likely reason for this is the ongoing migration from 3G to LTE in anticipation of the end of 3G services. Also, the slope of the decline in May 2020 onward is more moderate than from October 2019 up to April 2020, and one possible reason for this is that the speed with which companies are migrating has eased due to the effects of COVID-19. Meanwhile, 3G traffic volume is in a mild uptrend, unaffected by COVID-19. This phenomenon is dependent on the usage of business users, so we will be watching developments closely ahead.

Looking at LTE traffic trends, we see a drop in session counts during the stay-at-home period as with 3G, but the overall trend is of an intermittent rise. Traffic volume troughed in April–May 2020, when people were staying home, and is gradually coming back.

Finally, we look at 5G. As mentioned, IIJ released a business service that uses au 5G on October 30. So while we are not yet able to analyze traffic trends and the like, we investigated the extent to which IIJ users are using 5G-capable devices.

Figure 15 shows the rate of growth in devices likely to be 5G capable (Android devices with 5G in the device name and the iPhone 12 series) relative to October 1, 2019. Over the year before the iPhone 12's release, the figure increased about 40 fold, but in the year after release, it increased around 400 fold (around 10 fold vs. just before release). Traffic trends related to 5G services will also bear close watching ahead.

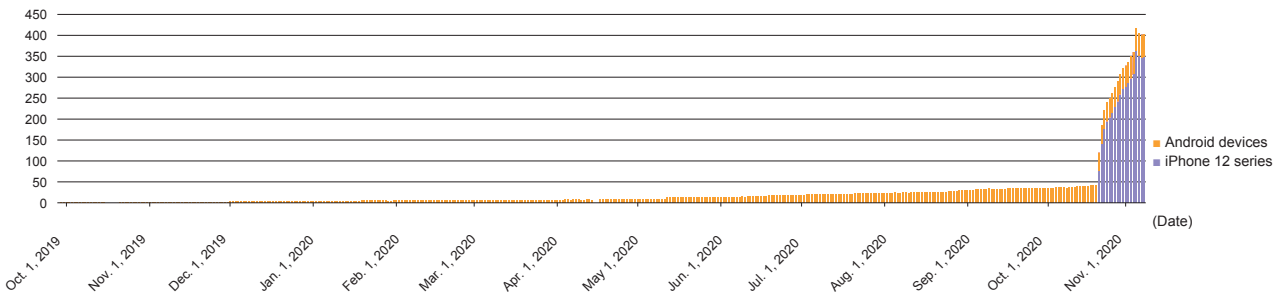

Figure 15: 5G-capable Device Connections

Topic 5

Deploying BGP ROV on the IIJ Backbone

Since November 2020, we have been progressively rolling out BGP ROV (Route Origin Validation), which uses RPKI, on IIJ's Internet backbone.

Note that we also discussed the deployment and workings of RPKI etc. on the engineers blog ahead of the deployment. The number of RPKI ROAs issued, etc., can be looked up via RIPE, NIST, and so on. RPKI itself has been used by RIRs since around 2008, and JPNIC also started taking registrations in 2015. RIPE is a fair way ahead in terms of number of ROAs, but other RIRs' ROA counts have increased greatly in the past few years, so usage appears to be progressing steadily.

Here, we extract information from a specific day in October 2020 before IIJ adopted BGP ROV. Table 5 was created from VRP (Validated ROA Payloads) records from IIJ's ROA cache server for that day in October 2020. You register ROAs with a set of information: a prefix that your organization advertises, an origin AS number that identifies your organization, and a max length, being the maximum prefix length that is acceptable. The number of addresses registered in ROAs as a proportion of BGP routes represents registered ROA prefixes as a proportion of the number of unique BGP route addresses on the specific day for IIJ.

Next, Figures 16 and 17 show the distributions of prefix length and max length in ROA registrations.

Table 5: VRP Data from IIJ's ROA Cache

	IPv4	IPv6	Total count
Unique prefixes	144,785	25,085	169,870
Unique ASNs	16,479	8,769	17,670
Unique prefixes+ASNs	158,099	27,024	185,123
AS0 prefixes registered	184	100	284
No. of ROA-registered addresses as % of BGP routes	27.9%	32.8%	-

Trust Anchors: RIPE NCC, ARIN, APNIC, AfriNIC, LACNIC

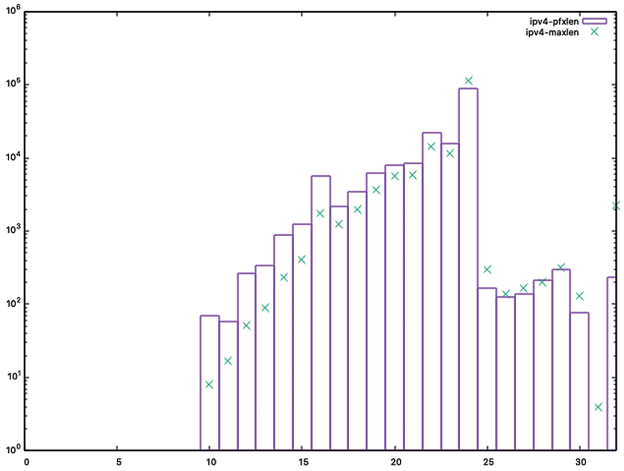

Figure 16: Distribution of Registered Prefix Lengths and Max Lengths (IPv4)

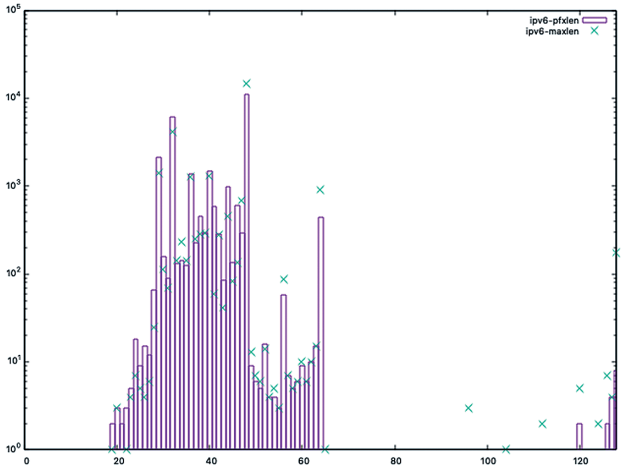

Figure 17: Distribution of Registered Prefix Lengths and Max Lengths (IPv6)

Prefix length is on the horizontal axis and registration count is on the vertical axis. The bars indicate prefix length in the ROA registrations, and the crosses indicate max length, which is the maximum acceptable prefix length. The registered prefix length and max length have the same values in 81.6% of cases for IPv4 and 78.7% of cases for IPv6. Meanwhile, max prefix tends to be longer for the /24 prefix on IPv4 and for the /48 prefix on IPv6, which looks to be the same sort of trend as when organizations exchange BGP route information on the Internet.

Figures 18 and 19 show the extent to which invalid routes are found when validating BGP routes for IIJ's region based on the aforementioned VRP data.

Around 24% of IPv4 and 29% of IPv6 routes are deemed valid, with around 0.32% of IPv4 and 0.49% of IPv6 routes deemed invalid. NotFound, indicating that a prefix matching the ROA was not found, accounted for 70% overall. As ROA registrations increase ahead, the NotFound slice of the pie can be expected to shrink. Even for invalid routes,

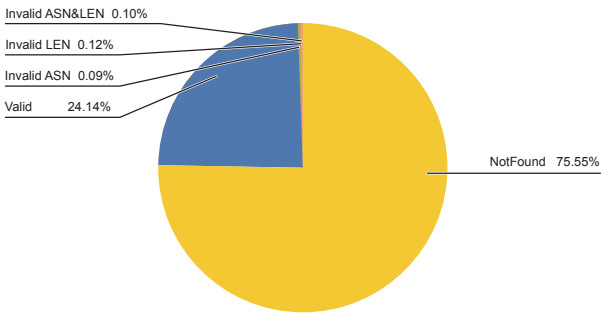

Figure 18: BGP Route Validation Results for a Specific Router (IPv4)

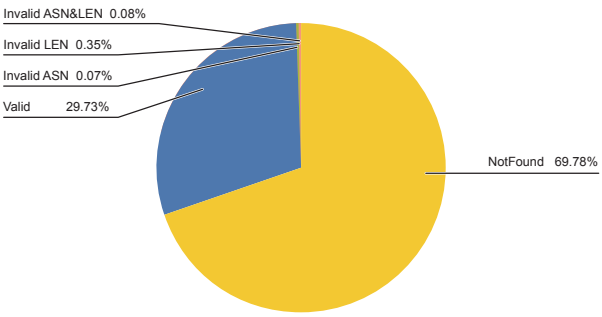

Figure 19: BGP Route Validation Results for a Specific Router (IPv6)

a larger IP space is determined to be valid or NotFound is some cases, so reachability is not necessarily lost if a route is deemed invalid, but there are also routes for which this is not the case, and these account for around 0.028% of IPv4 and 0.02% of IPv6 routes.

The Internet is constantly changing, so we are constantly exposed to the threat of a range of problems with routes due to misconfigurations and malfunctions. Determining whether these sorts of routes are valid or not can be problematic, so it has so far been very difficult to prevent problems from occurring. But with these recent initiatives along with the uptake of RPKI, the possibility of preventing some threats ahead of time is rising.

IIJ will continue to pursue initiatives geared toward providing resilient infrastructure to support a pleasant and convenient Internet experience for all.

1.BGP Routes
Tomohiko Kurahashi
Technology Development Section, Operation Technology Department, Infrastructure Engineering Division, IIJ

2.DNS Query Analysis
Yoshinobu Matsuzaki
Technology Development Section, Operation Technology Department, Infrastructure Engineering Division, IIJ

3.IPv6
Taisuke Sasaki
Deputy General Manager, Network Technology Department, Infrastructure Engineering Division, IIJ

4.Mobile 3G and LTE
Tsuyoshi Saito
Mobile Technology Manager, Network Technology Department, Infrastructure Engineering Division, IIJ

5.Deploying BGP ROV on the IIJ Backbone
Fumiaki Tsutsuji
Network Planning Manager, Network Technology Department, Infrastructure Engineering Division, IIJ

Trends in Post-Quantum Cryptography— 2020

This report provides an update as of November 2020 on the section titled "1.4.3 Trends in Post-Quantum Cryptography"[*1] in our focused research report in IIR Vol. 31. In the five years since the last report, post-quantum cryptography (PQC) has become so widely known that PQC textbooks[*2] have been published.

2.1 NIST Competition Overview

Last time we reported, we looked at the following four categories of promising algorithms with mathematical backgrounds (IIR Vol. 31, Table 2: Post-Quantum Cryptography Classifications).

- Lattice-based cryptography
- Code-based cryptography
- Multivariate cryptography
- Hash-based signatures

In addition to the above four categories, a cryptosystem known as isogeny-based cryptography also appears as a classification in the latest NIST competition[*3] report. Considerable time was devoted to discussing isogeny-based cryptography at the ECC2018 workshop[*4] held at Osaka University in November 2018, and the elegant figures presented by Chloe Martindale were a pleasure to simply gaze at.

To set up the algorithm in elliptic curve cryptography protocols like ECDH, we create a group structure by defining an additive operation for points on an elliptic curve determined as one of the public parameters and use the characteristics of the elliptic curve discrete logarithm problem. We define a point Q as $Q = kP$, meaning point P added k times. Security in this case relies on the difficulty of finding k given P and $Q = kP$. An isogeny is a type of mapping from one elliptic curve to another, and a key exchange method with a mathematical structure similar to the Diffie-Hellman key exchange algorithm has been proposed on the basis that it is difficult to find a mapping φ given the elliptic curves E and $E' = \varphi(E)$.

Dustin Moody gave a NIST announcement during his invited talk at PQCrypto2016 held in Fukuoka in February 2016, revealing plans for a post-quantum cryptography competition[*5]. The criteria for submissions were finalized at end-2016, and 82 algorithms were submitted by the November 2017 deadline. The submissions were screened, and 69 were selected as first-round candidates[*6]. Following intensive discussion at NIST's First PQC Standardization Conference in April 2018, in January 2019 NIST released NISTIR 8240[*7] and announced that 26 algorithms had advanced to the second round.

In August 2019, NIST held its Second PQC Standardization Conference, co-located with CRYPTO2019, and on July 22, 2020, it announced that seven finalists and eight alternates were advancing to the third round. A detailed status report on the selection process appeared in NISTIR 8309[*8]. Dustin

*1 Internet Infrastructure Review Vol. 31, "1.4.3 Trends in Post-Quantum Cryptography" (https://www.iij.ad.jp/en/dev/iir/031.html).
*2 An example of a post-quantum cryptography textbook in Japanese (https://www.morikita.co.jp/books/book/3503).
*3 NIST Post-Quantum Cryptography (https://csrc.nist.gov/Projects/Post-Quantum-Cryptography).
*4 ECC2018 (https://cy2sec.comm.eng.osaka-u.ac.jp/ecc2018/program.html). Chloe Martindale's presentation slides on CSIDH, CSIDH: An Efficient Post-Quantum Commutative Group Action (https://cy2sec.comm.eng.osaka-u.ac.jp/ecc2018/slide/slide_program/1121-3%20CSIDH%20Martindale.pdf).
*5 Dustin Moody, Post Quantum Cryptography Standardization: Announcement and outline of NIST's Call for Submissions, PQCrypto2016 (https://csrc.nist.gov/presentations/2016/announcement-and-outline-of-nist-s-call-for-submis), (https://www.youtube.com/watch?v=nfLAVybabMs).
*6 Dustin Moody, The ship has sailed: The NIST Post-Quantum Cryptography "Competition", Asiacrypt2017 invited talk (https://csrc.nist.gov/presentations/2017/the-ship-has-sailed-the-nist-post-quantum-cryptog), (https://www.youtube.com/watch?v=3doS6joRYTE).
*7 NISTIR 8240, Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process (https://csrc.nist.gov/publications/detail/nistir/8240/final).
*8 NISTIR 8309, Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process (https://csrc.nist.gov/publications/detail/nistir/8309/final).

himself published a blog post in December 2020 giving some background on these developments[*9].

Table 1 is a list of finalists and alternates created from Dustin's latest presentation materials[*10][*11] (alternates are shown in parentheses in red).

Of the seven finalists, three are digital signatures (two of those are lattice cryptography, one is multivariate public key cryptography) and four are key encapsulation mechanisms, or KEMs (three of those are lattice cryptography, one is code-based cryptography). Tweaks to each of the algorithms were allowed at the start of round three, and information on the algorithms and links to their websites appear on the Round 3 page[*12]. PDF documents listing the updates are also available[*13].

The timeline for the competition from here out is as follows. Round 3 (already commenced) is slated to last 12–18 months, at the end of which, NIST will select at most one of the finalists categorized as lattice schemes from the digital signature candidates and, likewise, at most one of the lattice schemes from the KEM candidates. NIST plans to hold a third conference in spring/summer 2021, and it tentatively expects draft standards to be available in 2022–23, and a standard to be published in 2024.

A project called PQCRYPTO was funded under the Horizon 2020 budget[*14]. It is evident from the "D5.2 Standardization: Final report" that PQCRYPTO's contribution to the NIST competition is sizeable. For example, of the 15 algorithms that advanced to Round 3, 11 were PQCRYPTO project submissions.

2.2 Cryptographic Algorithms Published by NIST and Their Impact

NIST has created a range of specifications and guidelines strongly linked to US government procurement requirements. NIST covers an extraordinarily wide range of technological fields, but from what we mainly see, its various guidelines related to information security receive significant attention. The documents on passwords specified in SP 800-63, for example, are read by many engineers and have triggered discussion on how we think about passwords. Fruitful discussion has taken place in Japan, too, regarding the pros and cons of periodically changing passwords and the pros

Table 1: List of Finalists and Alternates

Category/method	Digital signatures	KEM
Lattices	CRYSTALS-DILITHIUM, FALCON	CRYSTALS-KYBER, NTRU, SABER (FrodoKEM, NTRU Prime)
Code-based	None	Classic McEliece (BIKE, HQC)
Multivariate public key	Rainbow (GeMSS)	None
Hash-based signature	(Picnic, SPHINCS+)	N/A
Isogeny	None	(SIKE)

*9 Dustin Moody, The Future Is Now: Spreading the Word About Post-Quantum Cryptography, December 2, 2020 (https://www.nist.gov/blogs/taking-measure/future-now-spreading-word-about-post-quantum-cryptography).
*10 NIST PQC Standardization Update - Round 2 and Beyond, September 23, 2020 (https://csrc.nist.gov/Presentations/2020/pqc-update-round-2-and-beyond).
*11 NIST PQC Standardization Update - Round 2 and Beyond (https://www.nccoe.nist.gov/file/3-pqc-nccoepdf).
*12 Post-Quantum Cryptography Round 3 Submissions (https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions).
*13 History of PQC Standardization Round 3 Updates (https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/history-pqc-round-3-updates.pdf).
*14 PQCRYPTO project (https://pqcrypto.eu.org/), (https://cordis.europa.eu/project/id/645622). D5.2 Standardization: Final report (https://pqcrypto.eu.org/deliverables/d5.2-final.pdf).

and cons of using SMS messages in two-factor or multifactor authentication.

Also, many engineers who use, or who are involved in the use of, cryptographic algorithms know that the NIST publication series FIPS and SP continue to lead the world. While NIST also develops SHA-2, a set of widely used hash functions, suspicions that backdoors might exist given the US government's involvement have been around since DES was developed and published in the 1970s. As such, it is true that similar suspicions regarding the public parameters used in the elliptic curve cryptography method known as NIST curves have recently spurred a preference among some people for cryptographic methods standardized by the IETF as part of a grassroots effort. Yet many of the hash function algorithms implemented in cryptoassets based on blockchain technologies were laid out by NIST. There are cases in which the design and implementation of systems by engineers not well versed in cryptography has led to problems with the survival of the cryptoassets themselves, and it does seem like some people think of SHA-2 as being a secure, well-established technology that has been adequately scrutinized.

2.3 Views on Security in Post-Quantum Cryptography

The security of RSA relies on the difficulty of factoring certain numbers, but anyone can easily factor numbers up to, say, 100 or so (you just need to check divisibility by 2, 3, 5, 7, 11, and 13 for instance). The RSA cryptosystem is based on computational security, so it requires sufficiently large prime numbers to be used securely. The current recommendation is to use a composite number N of at least 1024 bits x 2 = 2048 bits, but consensus on this key length has continued to change with the times. What this tells us is that parameter settings, which rely on the security of the cryptographic algorithm, are of utmost importance. RSA is currently still recognized as secure, but this is because a sufficient key length is maintained. It can only be used securely if correctly implemented on that basis.

The same sort of issues with parameter settings apply to post-quantum cryptography. Even if a cryptosystem itself is thought to be secure, you still need to consider what sort of data should be used in terms of the analog of key length, for example, to ensure security. In this context, key length is an important consideration for cryptographic algorithms, especially the sort of public key cryptographic methods that achieve security through computational complexity currently in use. Similarly, an important issue for post-quantum cryptographic algorithms will be how the various parameters should be set to be secure. So for some categories, cryptanalysis competitions are being held to determine whether the methods are suited to current computing environments, and it is clear in some cases that the sharing of the latest attack methods is something that excites and stimulates the research community.

Competitions dealing with post-quantum cryptosystems classified as multivariate cryptography have been held since 2015. Owing to rapid advances in the research, cryptographic algorithms for which we had assumed some set of parameters was adequate have turned out not to be as secure as we thought in many cases. A presentation by Jintai Ding[*15] at the Second PQC Standardization Conference[*16] co-located with CRYPTO2019, for instance, necessitated a major review of parameters.

*15 Jintai Ding, New Attacks on Lifted Unbalanced Oil Vinegar (https://csrc.nist.gov/Presentations/2019/new-attacks-on-lifted-unbalanced-oil-vinegar).
*16 Second PQC Standardization Conference (https://csrc.nist.gov/events/2019/second-pqc-standardization-conference).

2.4 Bit Security

The cryptographic algorithms widely used today are called classical algorithms, as opposed to post-quantum cryptography. The common way of thinking about security with classical algorithms is that you select parameters to achieve a certain level of bit security. This concept of bit security is easy to understand in the context of symmetric key cryptography and hash functions, and past IIR reports have discussed the compromise of cryptographic algorithms and equivalent security[*17].

For example, the widely used symmetric key cryptographic algorithm AES-128 uses a 128-bit key, and 2128 operations are required to identify the decryption key, so it is said to have 128-bit security. The notion of bit security can also be applied to public key cryptosystems, letting us compare the degree of security offered by cryptographic algorithms based on certain key parameters. NIST publication SP 800-131A[*18] is a well-known source of tables comparing key lengths that is often cited, but what is interesting is that even for the same RSA key length, there is a little variation among different stakeholders' assessments of the level of bit security (see, for example, Table 1 in Section 1.4.1 of IIR Vol. 8).

Past reports on post-quantum cryptography have also contained similar cases in which the strength of an algorithm changes depending on what view the group or organization has formulated. Grover's algorithm, which I discussed in my previous report on post-quantum cryptography, has been shown to reduce the bit security of a symmetric key cryptosystem with n bits of security by half. But there are also reports indicating that some stakeholders have determined that the security of all symmetric key cryptosystems will drop to zero bits. In the case of symmetric key cryptography, the view that the exponent n in 2^n (which indicates how many operations are required for decryption) will fall by half is now widely accepted. Because NIST is running a competition on asymmetric key cryptosystems such as KEMs and digital signatures, symmetric key cryptography does not receive much attention as a post-quantum encryption scheme, but a number of independent research papers with intriguing findings have been published. One, for example, looks at how much of a threat quantum computing poses to the widely used AES[*19]. Based on their analysis, the paper's authors assert that there is a wide security margin in both the classical and quantum computing worlds, but we leave the assessment of this analysis and its prospects up to the reader.

2.5 Quantum Cryptography and Post-Quantum Cryptography

The two terms quantum cryptography and post-quantum cryptography have different meanings and backgrounds, but judging by some mainstream media articles, this seems to be the source of some confusion among the general public. An example in the case of the former is quantum key distribution, or QKD, which is a different concept from post-quantum cryptography. The technical subjects we cover in this report are part of the field known as post-quantum cryptography. We do not discuss technical topics that deal directly with quantum mechanics, such as quantum communication.

*17 For a discussion on the compromise of cryptographic algorithms, bit security, and equivalent security, see IIR Vol. 8 (https://www.iij.ad.jp/en/dev/iir/008.html), "1.4.1 Trends in the Year 2010 Issues on Cryptographic Algorithms".

*18 NIST Special Publication 800-131A Revision 2, Transitioning the Use of Cryptographic Algorithms and Key Lengths, March 2019. (https://doi.org/10.6028/NIST.SP.800-131Ar2).

*19 Xavier Bonnetain et.al, Quantum Security Analysis of AES (https://tosc.iacr.org/index.php/ToSC/article/view/8314). Presented at FSE2020 (https://fse.iacr.org/2020/program.php).

Discussion around post-quantum cryptography focuses on what sort of impact the implementation and widespread availability of quantum computers, assuming this happens, would have on the cryptographic algorithms currently in use. Some readers may therefore be very surprised to learn that post-quantum encryption is already implemented in Web browsers and such[*20]. Reports in the press saying that quantum computers are already in commercial circulation may lead to the misunderstanding that cryptographic algorithms than run on those quantum computers have been implemented. The models we deal with, however, should be understood as modelling attacks on the basis that only attackers with quantum computers have access to the enormous amount of computing power that quantum computing provides, while the vast majority of people are using classical computers.

2.6 What Does Post-Quantum Cryptography Mean?

The definition of post-quantum cryptography is not clear, and it is very difficult to delineate a proposed algorithm as being post-quantum or not, but a decent way to think about it is that post-quantum algorithms are those for which security relies on factors other than the computational security employed by the widespread cryptographic algorithms of the past. That is, we can think of them as replacements for algorithms that are implementable on classical computers.

So even some algorithms proposed way back in the 1970s, for example, are being revisited and featured as post-quantum cryptographic methods.

Meanwhile, research on yet other algorithms has advanced rapidly in the past few years, post-quantum cryptography is becoming a major topic in the cryptographic research community.

Two triggers that drive cryptographic research are the discovery of attacks that make today's commonly used algorithms unusable (i.e., the compromise of cryptographic algorithms), and the prospect of attacks with a large enough impact to render algorithms unusable in the future. An example relevant to the latter is the formulation of a new hash function called SHA-3. NIST selected a proposed algorithm that is internally different from the mathematical structure used in the design of SHA-1 and SHA-2 to become the standard (published as FIPS documents). But almost no progress has been made migrating to SHA-3 and it is believed that SHA-2 can still be safely used. Post-quantum cryptography is also seen as relevant to the latter and is more of an effort to prepare for the future rather than a reaction to algorithms being compromised.

The concept of agility in cryptographic algorithms highlights the importance of having "another card up your sleeve" in terms of cryptographic algorithms designed based on differing ideas and backgrounds, and indeed a whole host of algorithms with various backgrounds are featured in the current efforts to develop post-quantum cryptography. Of them, the lattice-based cryptographic algorithms that have been the subject of research since the 2000s are strong contenders and account for many of the remaining finalists.

*20 qTESLA (https://qtesla.org/) and NewHope (https://www.imperialviolet.org/2018/04/11/pqconftls.html) were not selected for Round 3, whereas SIDH (https://blog.cloudflare.com/introducing-circl/) did make it to Round 3.

2.7 Impact of Post-Quantum Cryptography on Symmetric Key Cryptography and Hash Functions

A look solely at algorithms in the NIST competition seems to indicate that post-quantum cryptography is focused only on public key cryptosystems, but this is not the case in actual practice. Hybrid systems that use, for example, symmetric key encryption along with public key encryption are in use. The digital signature methods use two kinds of algorithms to sign messages with a cryptographic hash function and a public key cryptosystem. The balance between the two algorithms is crucial in these hybrid methods, and you need to consider whether each of the algorithms has n-bit security. As such, we also need to consider the impact of the advent of quantum computers on symmetric key cryptography and cryptographic hash functions. In light of Grover's algorithm, it is known that a symmetric key algorithm with an n-length key has only n/2 bits of security. In specific terms, once quantum computers eventually arrive, using a cryptographic algorithm with 256-bit security only provides the same strength as a classical cryptographic algorithm that uses a 128-bit key[*21].

Next, how should we approach hash functions? Cryptographic hash functions need to have two cryptographic properties. One is collision resistance, and the other is preimage resistance. It is known that on classical computers, hash functions with an output size of n bits have n/2-bit collision resistance and n-bit preimage resistance. Grover's algorithm is the most optimal for the latter, and it is known that the number of operations required for a preimage attack on a hash function with n-bit output falls to $2^{n/2}$.

The number of operations needed to find a collision using a quantum computer using an efficient algorithm called BHT is $2^{n/3}$, but this attack requires $2^{n/3}$ of quantum memory, which is a huge amount that makes the attack unrealistic[*22].

In CRYPTREC Report 2019 (a group of documents summarizing the results of CRYPTREC activity in FY2019)[*23], Akinori Hosoyamada's commentary says that the CNS algorithm[*24] is the one most realistically likely to have an impact and reports that this algorithm can find collisions in $2^{2n/5}$ operations. For example, the SHA-256 algorithm has 128 bits of security (in terms of collision resistance), so even attacking it using the CNS algorithm would require over 2^{100} operations, and Hosoyamada therefore concludes that it probably does not pose a realistic threat. Hence, it is thought that the advent of quantum computers should have less of an impact on symmetric key cryptography and hash functions than it does on public key cryptography. In the sense that one can make ready by using algorithms already published and put into widespread use, there is not much to do in the way of preparations at this point.

*21 Internet Infrastructure Review (IIR) Vol. 31, 1.4.3 Trends in Post-Quantum Cryptography (https://www.iij.ad.jp/en/dev/iir/031.html).
*22 Gilles Brassard et al., Quantum cryptanalysis of hash and claw-free functions. SIGACT News 28 (2): 14–19, 1997.
*23 Akinori Hosoyamada, Investigation and Assessment of the Impact of Quantum Computers on Symmetric Key Cryptography, CRYPTREC EX-2901-2019, Jan. 2020. (https://www.cryptrec.go.jp/exreport/cryptrec-ex-2901-2019.pdf, in Japanese).
*24 Andre Chailloux et al., An efficient quantum collision search algorithm and implications on symmetric cryptography, LNCS10625, pp.211–240, Asiacrypt2017, 2017.

2.8 CRYPTREC's View on the Threat of Quantum Computers

In February 2020, the CRYPTREC Cryptographic Technology Evaluation Committee issued an alert[*25] that offered CRYPTREC's response from a technical perspective to a paper that had, at the time, recently appeared in Nature claiming to have realized quantum supremacy on a quantum computer[*26]. The alert reported that although the paper had raised concerns that the security of widely used public key cryptographic methods could be greatly diminished, the likelihood of the cryptographic algorithms in CRYPTREC's cipher list being compromised is low. The basis for this assertion was that the paper in question assumes an ideal environment with zero quantum errors, and that another paper[*27] claiming that RSA integers can be factored in 8 hours estimates that 20 million qubits would be needed, a situation far removed from current progress in the implementation of quantum computers.

CRYPTREC's rationale for issuing the alert was that it needed to "Release accurate, highly trustworthy information as a means of preventing overreactions" under item B in its communications workflow that applies when information on vulnerabilities in cryptographic algorithms is detected. Refer to Chapter 1 of CRYPTREC Report 2019 for background information and details of the communications workflow.

2.9 Dialog with People at NIST

At a workshop[*28] co-located with EUROCRYPT2016, I had the pleasure of talking with NIST's Lily Chen about policies on cryptography including the post-quantum variety. I pointed out that NIST has two different cryptography policy directions: post-quantum cryptography and lightweight cryptography. At the time, I only envisioned the post-quantum response for symmetric key cryptography would entail extending the life of the technology by, for example, doubling key lengths. I asked whether NIST would be looking at developing new algorithms, like AES-512 for instance, or turning to, say, Triple AES (using a three-key bundle like Triple DES). Her answer was that we already have AES-256, which was secure at the time and will provide 128-bit security with key lengths available in 2030 and beyond, so even AES-256 will provide sufficient resistance to quantum computers.

I also hadn't imagined the introduction of symmetric key cryptographic methods with new backgrounds, like in public key cryptography, but at FSE2020, there was actually a presentation about Saturnin, a post-quantum symmetric key cryptosystem[*29]. Saturnin is both a lightweight and post-quantum cryptographic suite of algorithms. A NIST competition aimed at standardizing lightweight cryptography is also underway. Such algorithms are called lightweight as they are aimed at devices with little computing power,

*25 CRYPTREC Cryptographic Technology Evaluation Committee, The impact of current quantum computers on the security of cryptographic technologies, Feb. 17, 2020, CRYPTREC ER-0001-2019 (https://www.cryptrec.go.jp/topics/cryptrec-er-0001-2019.html, in Japanese).

*26 Frank Arute et al., Quantum supremacy using a programmable superconducting processor, Nature 574, pp. 505–510, 23 October 2019. (https://doi.org/10.1038/s41586-019-1666-5).

*27 Craig Gidney et. al, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits (https://arxiv.org/abs/1905.09749).

*28 9th International View of the State-of-the-Art of Cryptography and Security and its Use in Practice (https://www.iacr.org/conferences/eurocrypt2016/Vienna-May13-2016.pdf).

*29 Anne Canteaut et.al., Saturnin: a suite of lightweight symmetric algorithms for post-quantum security, FSE2020 (https://iacr.org/cryptodb/data/paper.php?pubkey=30514). Presented at FSE2020 (https://fse.iacr.org/2020/program.php). Saturnin project (https://project.inria.fr/saturnin/).

such as IoT devices. Key lengths of around 80 bits are envisioned, and security is weaker than with commonly used algorithms. CRYPTREC also ran a lightweight cryptography working group from FY2013 through FY2016, and in June 2017 it published the CRYPTREC Cryptographic Technology Guideline - Lightweight Cryptography*30 with the objective of supporting the appropriate use of lightweight cryptography. Holding separate competitions for lightweight cryptography, like those for lattice-based methods and such, provide an opportunity to test whether your own method is secure, and this is one area of research that I am personally very excited about seeing future developments in.

Cryptographic algorithms are being standardized for various use scenarios. As discussed above, the advent of quantum computers will not immediately have an impact, but you can keep up with the latest trends via the CRYPTREC website, so I encourage you to take a look.

Yuji Suga
Senior Engineer, Office of Emergency Response and Clearinghouse for Security Information, Advanced Security Division, IIJ.
Dr. Suga has been in his current position since July 2008. He is engaged in investigation and research activities related to cryptography and information security as a whole. He is a member of the CRYPTREC Cryptographic Technology Promotion Committee. He is also secretariat of the Cryptographic protocol Evaluation toward Long-Lived Outstanding Security Consortium (CELLOS); secretary of the Information Processing Society of Japan's Computer Security Group (CSEC); IWSEC2021 General co-chair; AsiaCCS'22 General co-chair; Cryptoassets Governance Task Force (CGTF) Security Working Group member; APSIPA Multimedia Security and Forensics Technical Committee member; and BGIN (Blockchain Governance Initiative Network) co-initial contributor.

*30 CRYPTREC Lightweight Cryptography Working Group, CRYPTREC Cryptographic Technology Guideline - Lightweight Cryptography (https://www.cryptrec.go.jp/report/cryptrec-gl-2003-2016en.pdf). CRYPTREC Symposium 2017, Introduction to the Lightweight Cryptography Guideline (https://www.cryptrec.go.jp/symposium/20171218_cryptrec-lw.pdf, in Japanese).

Query Service—The Challenge of Developing a Flexible Managed Database Service

3.1 Introduction

IIJ began a Tech Challenge scheme in fiscal 2019 to provide opportunities for engineers to breathe life into new ideas for services and technologies they ponder on a daily basis. My project was selected for the inaugural Tech Challenge, and over the course of a year through September 2020, I worked on my own to design the service specifications and develop a prototype of the service.

The theme of my project was the development of a query service. While deploying applications in Kubernetes containers is common practice these days, we still do not have an optimal solution for using Kubernetes containers when it comes to database persistence, availability, and performance. My aim was to address these issues by developing a query service (a managed database service) in the form of an external service that runs alongside Kubernetes and provides the same level of flexibility as when using containers, as well as data persistence and availability.

3.2 Key Features Developed

So I had a big development theme set for the Tech Challenge—that of developing a query service—and while I could see what issues I needed to solve, once I began working on the project, I found I had to rethink specifically what features I would be developing. I spoke to administrators who develop and operate IIJ's services to learn about the issues surrounding databases. Those conversations made a lot of sense to me, and I identified some overlap with the issues in my existing development role. Several potential features that would be highly useful for developers/operators came out of this, and I set to work laying out the requirements and designing and developing the query service.

Databases are a crucial component when developing and operating services, but building and operating a database is a heavy burden for the small teams that focus on developing and running service applications. Developers and operators do not want a database server. They want database functionality that allows them to connect via a client to a data store and use CRUD operations. Moreover, when it comes to non-functional requirements, they want these to be met, but they do not want to deal with them. So I saw the need for a service that completely hides (from the user) the deployment of a database before it is used and the design of instances in a way that takes into consideration high database availability, backups, security configuration, and performance. Yet these really are features you would expect from a database service, and I didn't feel the query service would be offering anything novel in that respect.

Yet there was something that really made sense to me in what the administrators who develop and run IIJ's services were saying. When a service application's execution speed drops off, they wanted some way of just powering through it. For developers, the standard responses when database performance fails to improve include tuning SQL queries and adding indexes. But when an application suddenly slows down, a feature that sort of "magically" increases database power without having to stop the application would be wonderful. This is something that I certainly would have benefited from when I was developing services, so I decided to implement this sort of magic feature into my query service. This is something that not even Kubernetes database operators offer, so the technical challenge it posed made it feel all the more worthwhile and motivating for me.

One other issue came up quite often in my conversations, which can be summarized as follows: "Data outlives systems, so we need long-term access to the database, but we have to migrate the data every time the system is replaced. The database itself also needs to be updated to new versions, but no one in our team is very familiar with the process, so we continue to use the old version as is." I spent many years in our system integration team providing technical support on a lot of database migrations and version updates due to customer system replacements, and I know firsthand how exhausting such projects can be.

I knew the query service should naturally allow users to use the same database, and that I should aim for the hardware and software on which the database runs to be constantly upgraded, but I also understood that it was crucial for this not to put a burden on the users. This is the sort of thing Kubernetes' rolling upgrades aim to achieve, and I think this would be a similar feature, but I developed a completely original implementation for the query service.

Although I had a year, the reality is that I was actually developing it entirely on my own. There was a chance it would end up half baked if I tried building all sorts of stuff into it. But I knew it had to have the bare minimum for modern-day databases in terms of the non-functional requirements of high availability and backups. In terms of what makes the query service distinct, I wanted it to solve the issues that I and the engineers I knew faced and, in line with the initial concept, to provide the same or greater level of functionality as databases in a Kubernetes container. So I moved forward with the following as my development priorities.

(1) Develop features that let users control the performance of the database without having to stop the database
(2) Develop an interface to allow (1) above to be used freely and easily from within the application
(3) Develop flexible billing functionality that makes (1) above easy to use
(4) Functionality that lets the system control (1) above on behalf of the user and always provide optimal performance
(5) A service update feature to facilitate ongoing use of the query service

Figure 1 is a conceptual overview of the features developed. At its core, the service uses Oracle Database (we are looking at supporting other databases too).

Starting in the next section, I describe the query service's core features, with (1) and (2) above covered in "Online Resource Reallocation Feature", (3) in "Per-second Billing Feature", (4) in "Autoscaling Feature", and (5) in "Service Update Feature".

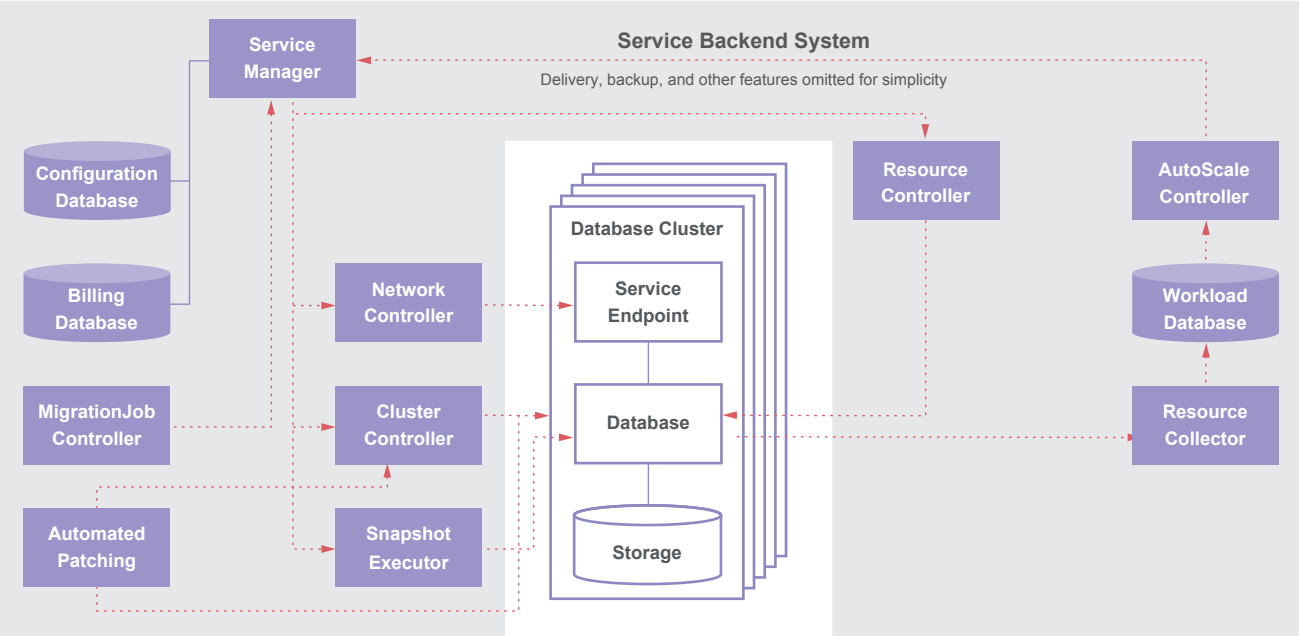

Figure 1: The Orchestration System Developed

3.2.1 Online Resource Reallocation Feature

To address the first development priority—functionality allowing the user to freely control database performance without stopping the database—I developed an online resource feature.

Setting databases up to easily allow for a distributed configuration to be adopted as loads increase, like with Web and application servers, is difficult. Replication, sharding, and configurations that sacrifice integrity are often used to facilitate scalability. The effects of such strategies are limited to read operations, however, and they come with restrictions on transaction processing and so forth. Particularly when performance issues arise with a single, large, database-specific process, scalability through load balancing can be ineffective.

A database's multi-workload capabilities are another aspect of its performance, and characteristically the amount of resources needed by a database can vary greatly depending on how it is used. Database processing varies greatly mainly depending on what combination of the following is in play: (1) SQL syntax, (2) amount of data to be processed, and (3) number of concurrent processes.

What combination of these factors is in play changes substantially throughout the day (Table 1). For example, many users will use a database concurrently during the day, but each operation is small. At night, however, daily jobs are executed, and while only a few of these run concurrently, it is not uncommon for any single operation to be large. Adding to this are seasonal fluctuations and spontaneous events, making workload even more complicated.

Table 1: Characteristics of Database Operations

	Day	Night
SQL syntax	Simple	Complicated
Data volume per operation	Low	High
Concurrent operations	Many	Few
Desired performance characteristic	Response	Throughput
Most important resource	CPU/memory	I/O

In traditional on-premises computing environments, databases have always had the most resources allocated within a system given that they combine different characteristics using inelastic resources.

In the last decade, it has become commonplace to use cloud services for computing resources. The days of people being skeptical about running databases in the cloud are long gone, and databases now run on cloud services seemingly as a matter of course (this is reminiscent of today's efforts to run databases on Kubernetes).

The multitude of computing resources that cloud services provide has greatly expanded users' resource options, but databases in the cloud do have their issues. If you build and run a database on IaaS, for instance, just because the location of virtual machines shifts into the cloud does not mean the fundamental problems you face when working with an on-premises system are solved. And a lot of the database services that cloud vendors provide on a PaaS basis need to be restarted when CPU resources, in particular, are reallocated. Stopping the database means stopping the entire service, so resources reallocations cannot be executed on a whim.

Scaling out a database with Kubernetes and Kubernetes operators is also effective and has gained a lot of attention of late (with replication, increasing the system's overall processing capacity is effective in the case of high-volume load balancing). But as discussed above, scaling up is probably more effective than scaling out when a single operation is slow (NewSQL is also an option, but I personally don't think it's quite mainstream yet). "Well then don't write inefficient

SQL," I hear you cry, but I think others working in this area will understand that this actually happens a lot in reality. I do not mean to criticize the scale-out strategy. In fact, I like scaling out, and while I do not mention it in this report, my query service also supports scaling out.

But I digress. Turning back to the query service, in an effort to increase processing performance without stopping the database, I developed an online resource reallocation feature that abstracts the computing resources. It provides the following functionality.

(1) CPU and I/O data throughput affects database performance, and this feature makes available as much of this as is needed when it is needed, without having to stop the database.
(2) It provides an interface in SQL so that resources can be easily reallocated from within the application that is the source of the resource request.

Table 2 shows (1) expressed as a service specification.

The CPU core and I/O data throughput resources can be increased or decreased separately, from the basic level up to the maximum spec. The change in resources happens within a few seconds, as I will discuss, and this is immediately reflected in the billing information.

One characteristic of (2) is that the online resources feature can be used via SQL as well as an API. The online resources feature is easy to understand even when controlled manually, and I developed it so that it would be easy to reallocate resources from the application that is the source of the resource request, so it is easy to embed into programs. This makes it possible to use resources in a way not previously possible, since you can, for example, change the number of CPUs before and after executing a large processing operation, and it is easy for developers to implement this (Figure 2).

Table 2: Query Service Specifications

	Fixed	Variable	
Resource	Basic	Options	
		Maximum spec	Increment
Database	1	—	—
CPU score	1	6	1
Data throughput (MB/s)	100	2000	100
No. of concurrent connections	50	300	Linked to CPU score
Data area (GB)	50	8000	Automatically increased

```
        if maxpom <= 2000 and maxgon >= 100 then
            dbms_output.put_line('Current max power ==>'||rc1.ag_power);
            vSQL := 'exec cpumod(6)';
            execute immediate vSQL;
            insert into testtab as select * from testman;
            commit;
            vSQL := 'exec cpumod(2)';
            execute immediate vSQL;
            select  testcol, to_char(modified_datetime,'YYYY-MM-DD HH24:MI:SS') as monday into testaa
            dbms_output.put_line('New maxmbps count ==>'||testaa);
            dbms_output.put_line('Resource was modified at '||moddatechar);
        else
            dbms_output.put_line('ERR-xx1 : Invalid argument [ '||aaaaa||' ]');
            dbms_output.put_line('ERR-xx2 : Valid argument is between 100 and 2000 ');
        end if;
    end loop;
end;
/
```

Change no. of CPUs to 6

Parallel processing to increase performance

Change no. of CPUs to 2

Figure 2: CPU Parallel Processing

Next, I explain how the online resources feature works. If we're going to provide a paid service, then allowing users to directly change the number of CPU cores and I/O throughput would quickly put us out of commission. Hence, although the service provides an interface via SQL procedures, the system changes are not made by executing a simple program that wraps the SQL command for making the change. Instead, the resource reallocation request is sent to the backend system's service manager via an external program (Figure 3).

So the procedure used to execute a request via SQL or the API is a simple program that takes the user request, determines on the user database whether the values are valid, and passes the request to an external program.

The service manager, having received the request via the external program, changes the information about the user database in the backend system's configuration database and changes the user database's billing criteria in the billing system. It then runs the database resource manager via the resource controller to change the user database's CPU core and I/O throughput configuration. Once this is done, it returns a message saying that the configuration is complete to the connected user session via the procedure on the user database (Table 3).

I don't have enough space to explain the program's implementation in detail, but these processes complete within a few seconds, so the user is able to adjust resources almost in

Table 3: Query Service Billing

	Daytime	Billing increment	Charges
Basic	IIJ Query Service, basic fee	1	Fixed monthly
Options	Additional CPU cores	1 core	
	Additional data throughput	100MB/s	Per-second billing
	Data capacity of 51GB or more	1GB	

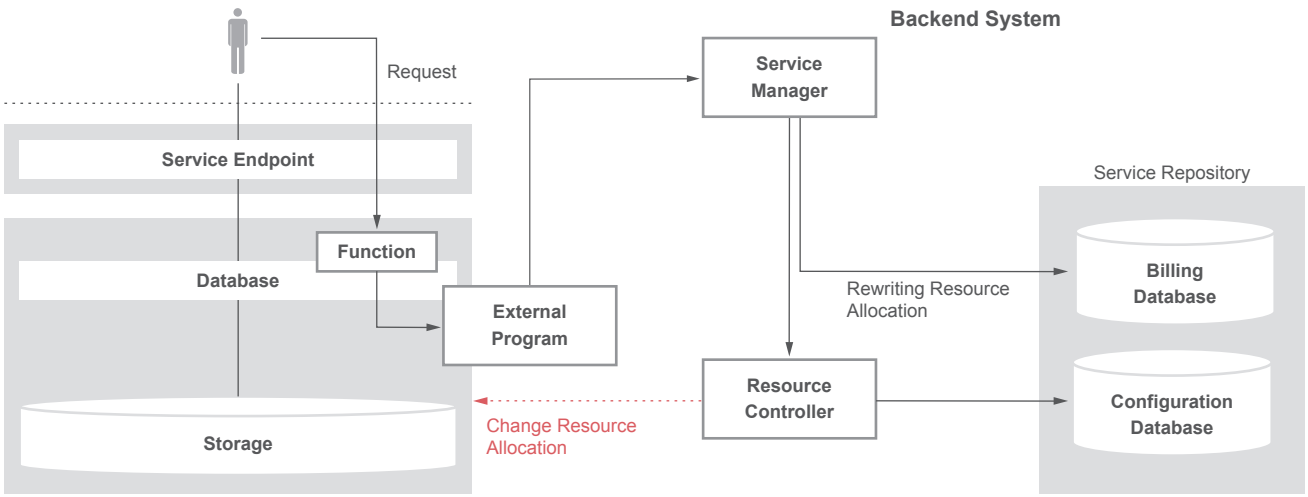

Figure 3: Resource Reallocation Backend

real time. Although possibly an extreme example, embedding the online resource reallocation feature into a program opens the door to an entirely new way of using database services as it means an application can dynamically measure the amount of data to process and dynamically change the resource allocation for every operation based on how much data there is (Figure 4).

3.2.2 Per-second Billing Feature

The online resources feature makes it possible to expand and shrink the resource allocation at any time, but it wouldn't be convenient if the resources billing were overly rigid. Aiming for a flexible billing structure for the query service, I also developed per-second resources billing for when the basic specs (CPU cores, data throughput, etc.) are exceeded.

Figure 5 illustrates how the query service pricing is implemented. CPU core counts, data throughput, and data area that exceed the basic spec are each billed for separately. This means users are billed only for the resources that they use when they need them.

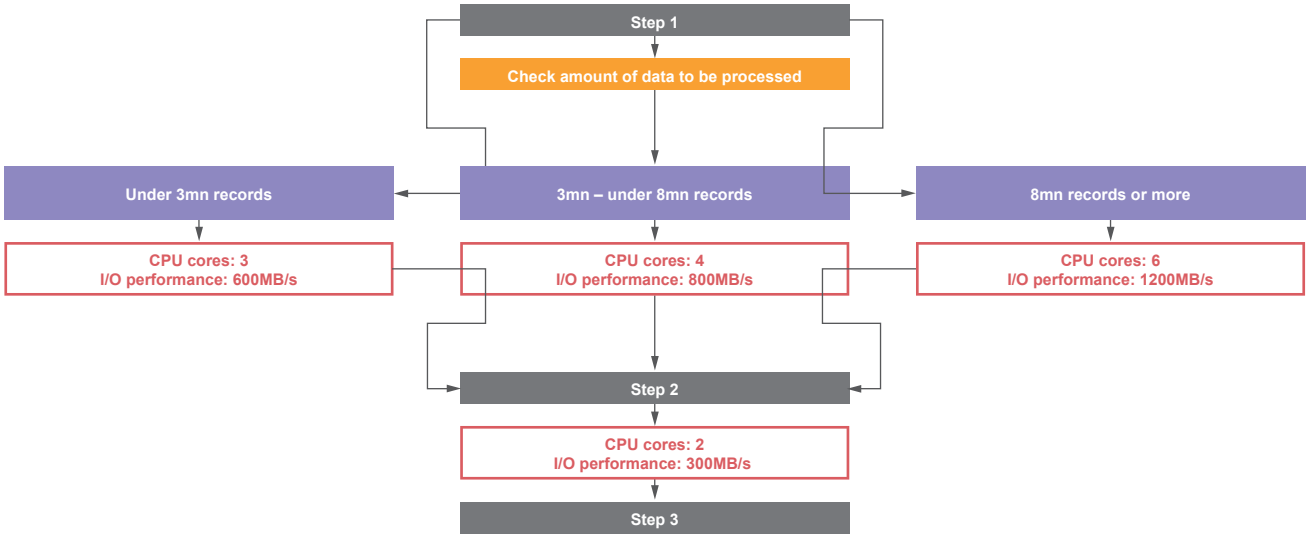

Figure 4: Programmable Resource Control

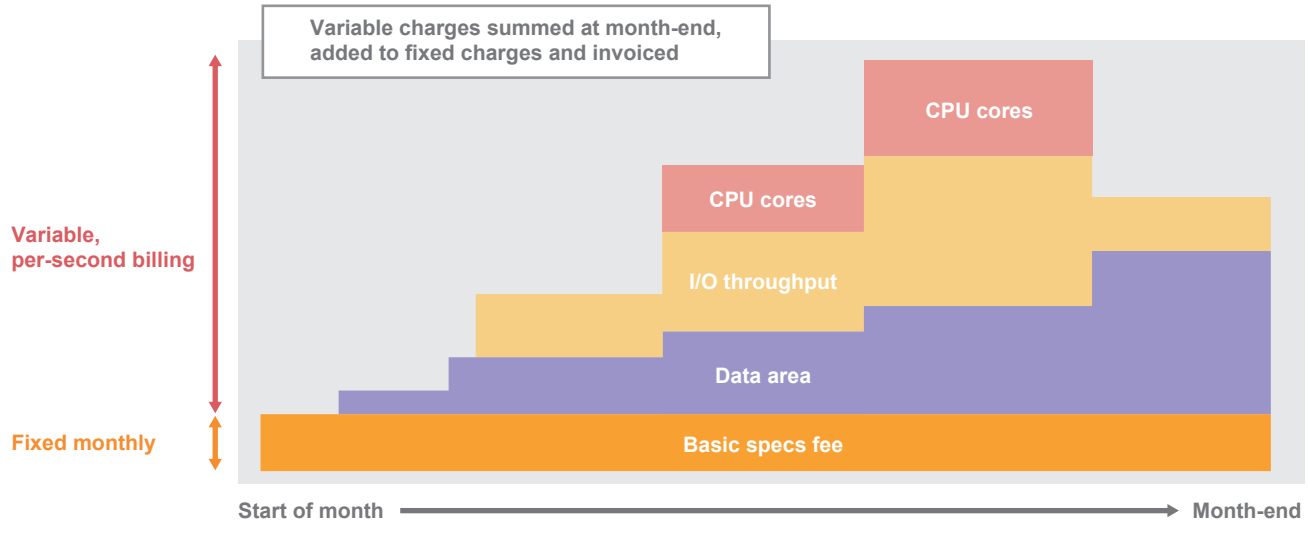

Figure 5: How Query Service Billing Works

If this can be combined with parallel database processing to balance processing speed and increases in resources, then we can keep the impact on overall charges small even as the per-second charge rises (Figure 6). I believe that more flexible billing functionality is what will encourage people to use the online resources feature.

3.2.3 Autoscaling Feature

The autoscaling feature works in conjunction with the online resource reallocation feature and constantly provides optimum performance by automatically controlling the system on behalf of the user. The online resource reallocation feature is useful, but if someone has to make a decision to

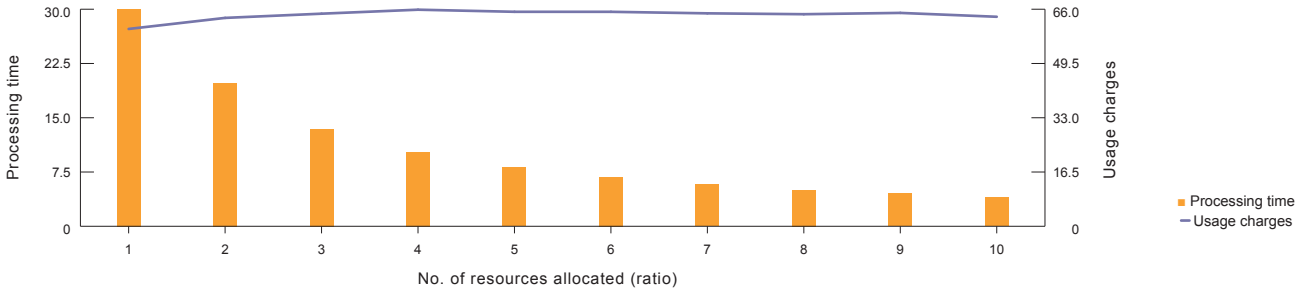

Figure 6: Online Resource Reallocation Feature and Per-second Billing

Table 4: Autoscaling Feature

Resource	Minimum	Maximum	Increment/decrement	Criteria for increase	Criteria for decrease	Decision interval
CPU	1 core	6 cores	1 core	Average usage of all allocated CPU cores over the last 5 minutes		1 minute
				70% or higher	65% or lower	
I/O throughput	100MB/s	2000MB/s	100MB/s	Average usage of allocated I/O throughput over the last 5 minutes		1 minute
				80% or higher	75% or lower	

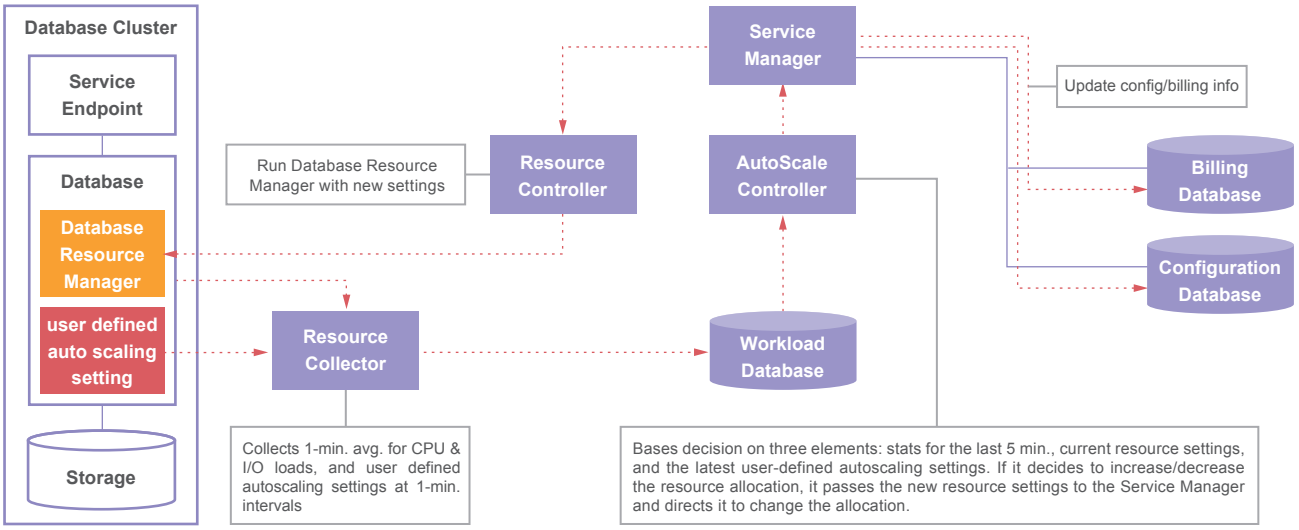

Figure 7: Internal Workings of the Autoscaling Feature

run it manually each time, and if it requires a complicated setup to combine it with links to an external monitoring system and so forth, this would greatly reduce the benefits of having it online and diminish the appeal of the feature. To automate system operations and facilitate the use of the query service's convenient features, I also developed an autoscaling feature to automatically control the online resource reallocation feature (Table 4).

Figure 7 shows the internal workings of the autoscaling feature. The autoscaling feature collects database operating status information, and in response to load levels, it automatically allocates, via the online resource reallocation feature, resources that affect database performance, namely CPU cores and I/O data throughput. It works constantly to keep resource loads within the criteria for either increasing or shrinking the resource allocation.

Autoscaling is a very useful feature with various use cases. One situation in which it is probably highly effective is that of dealing with performance degradation. If you run systems, you will come up against sudden slowdowns in database performance several times a year. This can have a variety of causes. Sometimes it suddenly slows down even though you haven't done anything (often, the database execution plan has suddenly changed), and otherwise, the trigger is always something different: a new program has just been resourced, or the volume of data has increased substantially, or you've run an event like a bargain sale, for example. When performance degradations visit, you face a battle against many conflicting forces.

FIrstly, it can be difficult to detect an application slowdown via resource monitoring on the infrastructure side, meaning that the operations team can be slow to detect what is happening. In many cases, user complaints are the trigger, and you face a high bar in dealing with the issue from the outset. Also, in some cases the operation causing the problem can be stopped at the process level, but stopping is not an option in other cases, such as nighttime batch jobs that have a significant business impact if not completed by the start of business hours. This is a tough situation: you cannot stop the operation, but you have to deal with it immediately. The root cause of these slowdowns is almost always on the application side and can include data spikes, queries running based on improper execution plans, and searches on items with no index. But even if you can identify the cause, dealing with it without stopping the system is either impossible or a highly difficult operation. And what's worse is that these sorts of disruptions often occur on holidays or late at night when no one is around.

While we might not be able to fundamentally resolve performance degradations riddled with conflicting issues like this, if the service could provide a solution, this would make users, operators, and developers all happy, and this is what I had in mind when developing the autoscaling feature.

■ Measuring the Effects of Autoscaling

Deliberately causing a performance degradation to see what effect the autoscaling feature has is difficult. Instead, I executed large operations on the database that were beyond the capabilities of the current specs to see how the autoscaling feature would react.

In this test, I joined several tables together—including an orders table with over 100 million records, a product master, and a customer master—and executed a query. Since this test makes it easy to tell what effect autoscaling has, I cleared the shared memory buffer each time a query finished executing.

When the query is executed, all records in the orders table are accessed sequentially. Running queries that cause sequential access and flush shared memory every time result in a large number of blocks located in storage being read out. So the queries require a lot of I/O resources.

When initialized, the query service has the basic specs (it's minimum configuration), which means a modest 100MB/s

of I/O throughput performance, so executing a sequential scan of over 100 million records drives the I/O throughput resource usage up sharply as soon as it is started (Figure 8). The autoscaling feature constantly collects and evaluates database usage statistics, and directs the system to increase or decrease the resource allocation. The queries used in the test were I/O-bound, so the autoscaling feature continued to increase I/O throughput performance only, up to 900MB/s.

What's interesting is that when I/O throughput reaches 900MB/s, the CPU reduces the I/O wait time, so the process becomes CPU-bound. When this happens, the service tries to increase the degree of parallelism by increasing the CPU core count, but since the queries were not the sort that raise CPU usage all that much, the autoscaling feature began to let go of the CPU cores again. This can be called online scaling up, but with containers on Kubernetes, it looks like this implementation has not yet reached a stable version. While interesting to watch, this sort of CPU core catch-and-release behavior will cause performance fluctuations, so there is certainly room to improve. Yet I don't think it's that big of a problem. With this sample database, the autoscaling feature is able to increase resources up to at most six CPU cores and 1000MB/s, but it determines that one or two CPU cores and I/O throughput of 900MB/s is adequate. Bumping it up to 1000MB/s or more and increasing the CPU core count does indeed increase performance, but the change is not remarkably large.

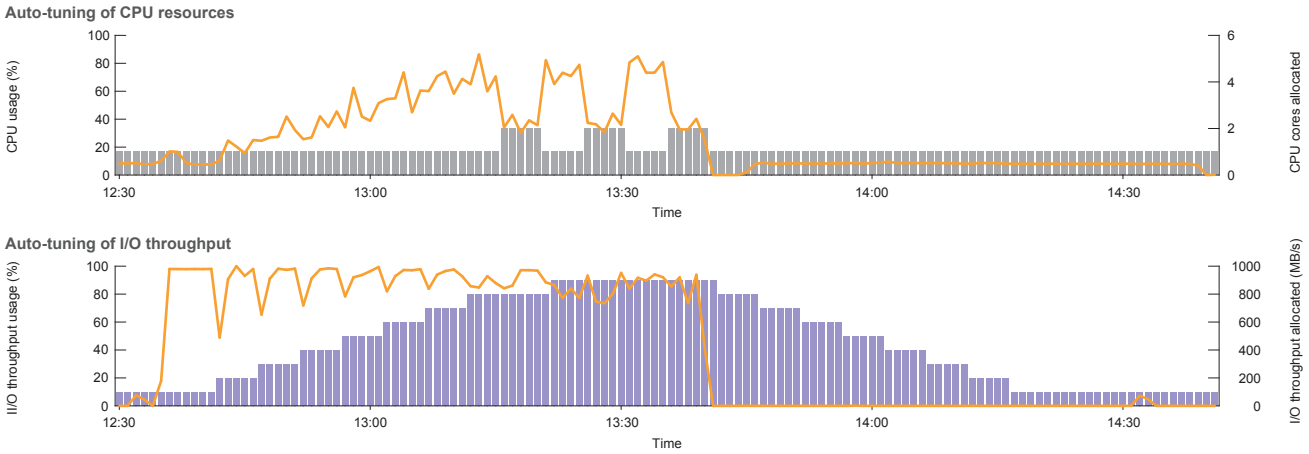

Figure 8: Example of Changes in Resources Controlled by Autoscaling Feature

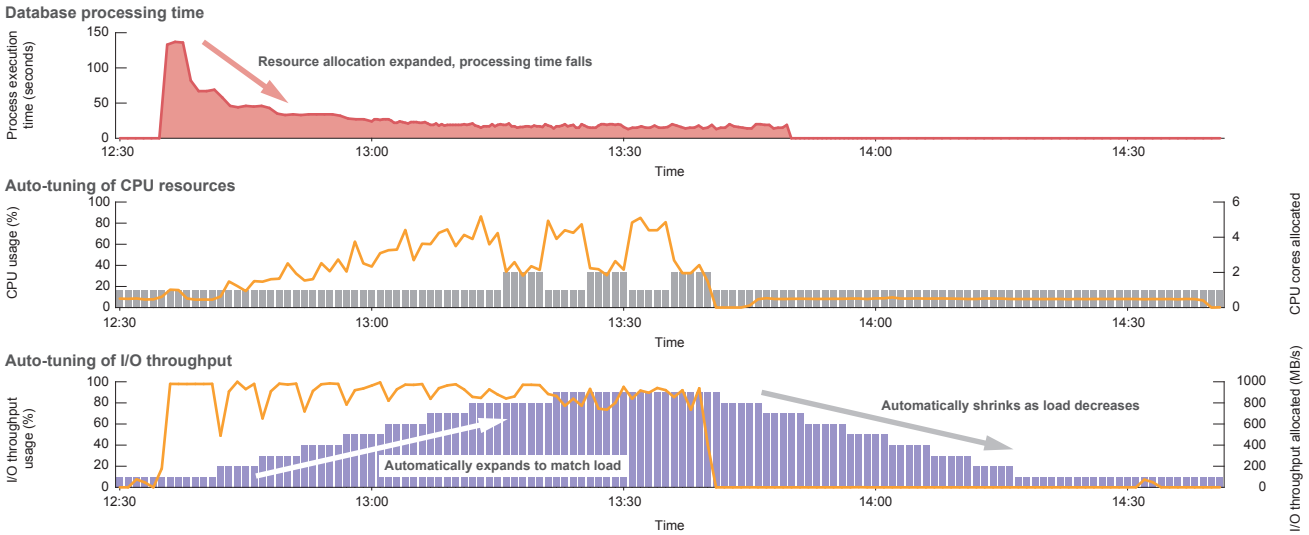

Figure 9: Example of Changes in Database Processing Time due to Autoscaling

Figure 9 shows the impact on actual processing performance. The initial increase in I/O throughput has a huge effect, and processing time certainly does drop.

Now, what happens if the user manually sets the I/O throughput performance value to 2000MB/s while the autoscaling feature is enabled? Manual settings take priority, so the database I/O throughput performance will be set to 2000MB/s. If autoscaling is disabled, the value will stay at 2000MB/s, but with it enabled, the system will assess operating conditions and shrink the resource allocation down toward 900MB/s.

The autoscaling feature could get in the way if it always reduces throughput to the minimum 100MB/s before then increasing it, resulting in delays in achieving the desired performance, so users are able to set a parameter range for the autoscaling feature to work within. For example, if it is configured to automatically adjust in the range of 2–4 CPU cores and 500–1500MB/s, it will not drop below 500MB/s. Users can also set autoscaling to adjust I/O throughput only and leave CPU core count unchanged. Changes to the settings can be made online, and these are reflected in the system after a delay of one minute (for reasons to do with the backend system).

■ Issues with Autoscaling

To ensure resources are used properly, the user cannot change the resource decision interval, thresholds, or increment/decrement values in the backend system, but I am looking at changing to an implementation that does allow the user to change these values as well.

And while autoscaling may seem useful, many issues remain. Especially in terms of resource allocation accuracy, I still face many difficulties from a developer standpoint.

As discussed, autoscaling operates on the basis of database usage statistics. Yet these figures are based on past occurrences, and the core autoscaling functionality, the autoscale controller, operates on the very simple assumption that past trends will persist for a while into the future, so it is not some sophisticated system capable of predicting future load levels ahead of time. Also problematic is that this clear and simple approach often deviates from desirable values immediately after a time-series regime shift. In specific terms, it sometimes increases resources despite loads being in decline. I'm hoping to make the autoscaling features even smarter via machine learning and so forth.

So while autoscaling has some remaining issues, I think it is one effective means of dealing with sudden performance degradations. In the end, since everything is controlled automatically, even if you are slow to detect an issue or it is discovered because of customer complaints, the autoscaling feature will be working in the meantime to expand the resource allocation in an attempt to maintain performance. And the resources allocation is shrunk once the issue is resolved, so you may not even notice it in some cases (although it will appear in your service charges). Configuration management and application deployments can be automated, and I think automating system performance maintenance as well will make system operations even easier.

3.2.4 Service Update Feature

While the query service espouses a serverless setup, it runs in the same system environment as a normal database, so the service platform will get old. Firstly, as the service platform's hardware ages, the incidence of mechanical failures rises. Security holes and bugs in the OS and database software stop being fixed because they are no longer supported or patches are no longer being released. So service platform renewals are crucial to continuously provide stable services.

Preparing a new query service platform is easy, but the current user database needs to be migrated to the new platform. In-place upgrades are the easiest way of dealing with simple version updates to the database alone, but version updates take considerable time, so lengthy service outages are unavoidable. There is also the possibility of OS

and hardware being inadequately renewed or of separate OS/hardware renewals being required, so this is not an efficient way of doing things. Another way is to create a separate instance from a backup, but this can involve changing the client's connection endpoint, which raises the possibility of the work required going beyond the scope of the query service.

The query service completes all of the following in a single process: switching over to new servers and storage, OS version updates, database software version updates, and upgrades to the database itself. This is similar to Kubernetes' rolling upgrades, but the query service does not use anything like replication. The mechanism implemented upgrades the entire service in a way that is faster and more transparent to the user.

(1) No data migration required
(2) All steps fully automated, including version updates and database switching that preserves data integrity
(3) A single click by the user will complete the entire process
(4) Takes at most 15 minutes to complete
(5) Can be run as many times as you like
(6) The query service connection endpoint does not change after the migration

The query service is equipped with functionality to perform the above service updates and allow continued use of the database. The service update feature does not require data to be migrated. Well, to be more precise, the in-use database is replicated on the new service platform under the hood, and the user does not need to think about it at all. Data integrity is automatically handled by transferring the transaction log.

The service update can be broadly divided into three phases. In Phase 1, a snapshot of the user's database is created on the new service platform online (Figure 10). The user does not need to create the snapshot manually. The service keeps tabs on whether it is possible to take a snapshot and creates one automatically when it is. Even if the user has multiple databases, the process is performed completely independently for each database.

From Phase 2, the database is switched to the new service platform, so the user needs to trigger the process, which can be done at a time of the user's choosing. Once Phase 2 is started, the service endpoint for the relevant user database is closed (Figure 11). This determines the quiescent point for reverting the switch. Once the quiescent point is

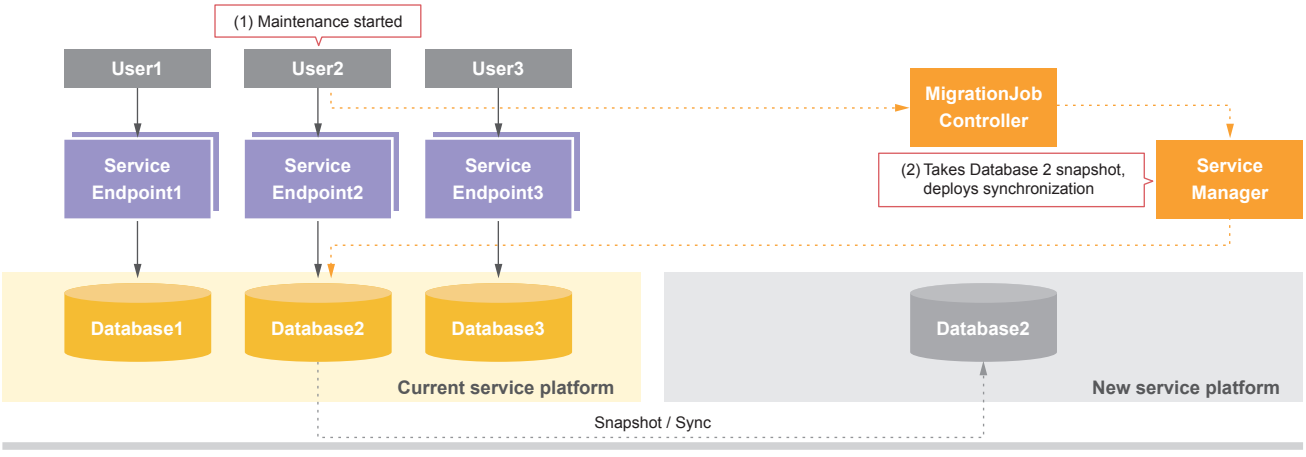

Figure 10: Internal Workings of Query Service Update Feature 1

determined, the final synchronization with the snapshot on the new service platform is done. Once the final synchronization process between databases is completed, the status of the database on the current service platform becomes inactive and the database goes offline, but it is not physically deleted. This means it is effectively a backup for the service update process. Plus, it avoids the time needed to restore from snapshot if the user rolls back the update. All that needs to be done is to change the status to active, so reverts happen very quickly.

The new database stands by on any processing until the current database becomes inactive. This is to avoid a split-brain scenario caused by both databases being active at the same time. So it doesn't process anything independently. Once the current database is properly inactive, processing resumes so that the service can continue on the new database. If the database version is to be updated, this is when it happens. The new database is also reconfigured for high availability. In the final stage of Phase 2, route information on the user database service endpoint is refreshed to reflect the new service platform.

In Phase 3, the endpoint is opened once the user database on the new service platform goes to active status (Figure 12). A notable aspect of Phase 3 is that the users connection endpoint does not change; only the route information on the service endpoint is changed. So once the user receives notification that everything is complete and reconnects, they are connected to the user database on the new service platform without having to change anything.

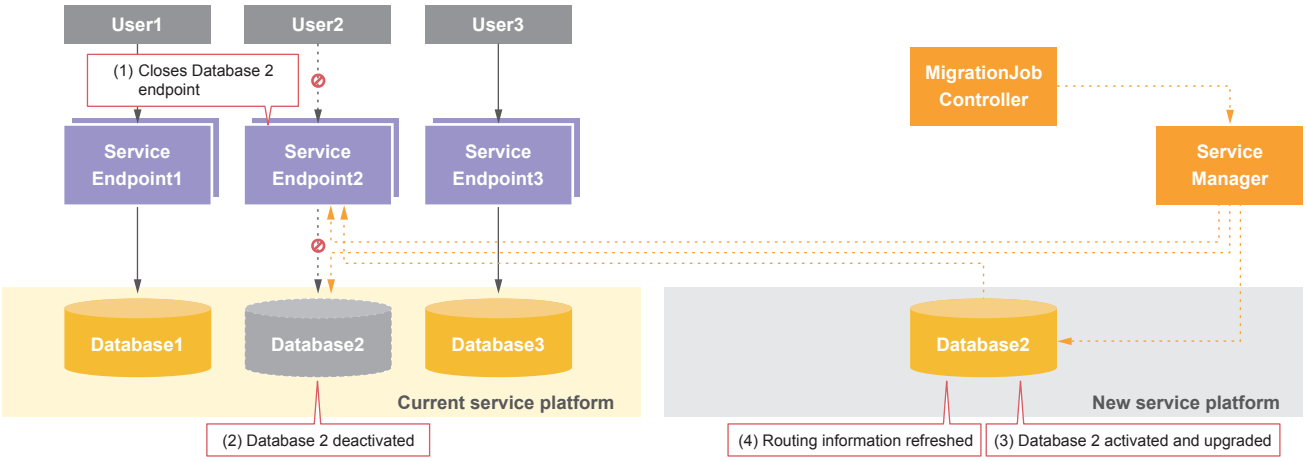

Figure 11: Internal Workings of Query Service Update Feature 2

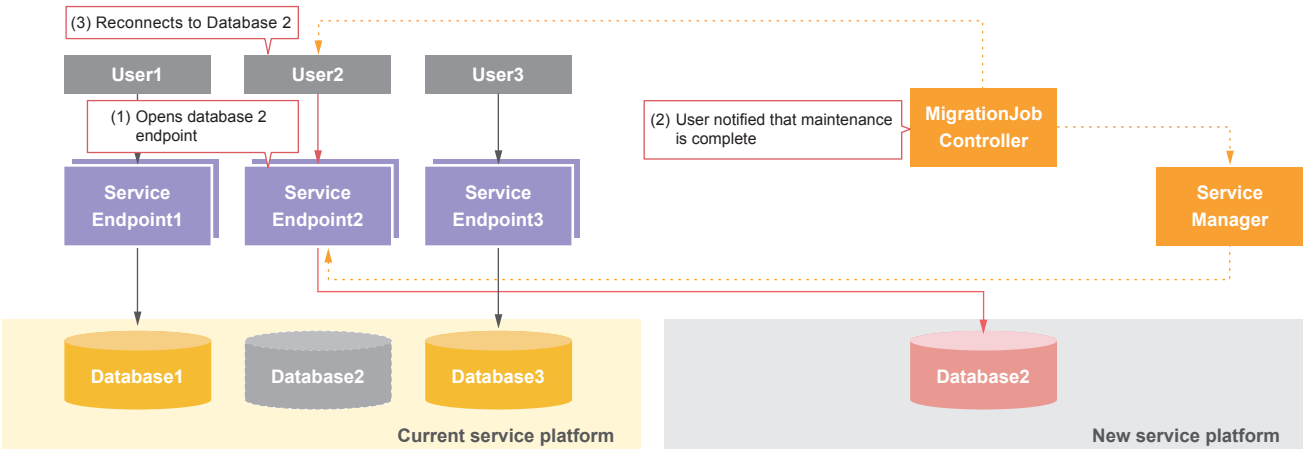

Figure 12: Internal Workings of Query Service Update Feature 3

When a switchover is executed, the service endpoint, which is a proxy between the client and the database, is blocked. Blocking the service endpoint eliminates the route between the client and the database, so user sessions are completely disconnected. Therefore, if any transactions were being processed in any sessions, these are rolled back on the database. This means it is best to initiate a switchover when no transactions are being processed. Once current database integrity is established with the service endpoint block in place, the final delta synchronization with the new database takes place.

The work done by the service update feature has traditionally been performed as a system integration project. Figure 13 shows what it takes to do this manually. Quite the laborious task. The query service automates all of this under the hood, so service users simply need to make a single click on the control panel.

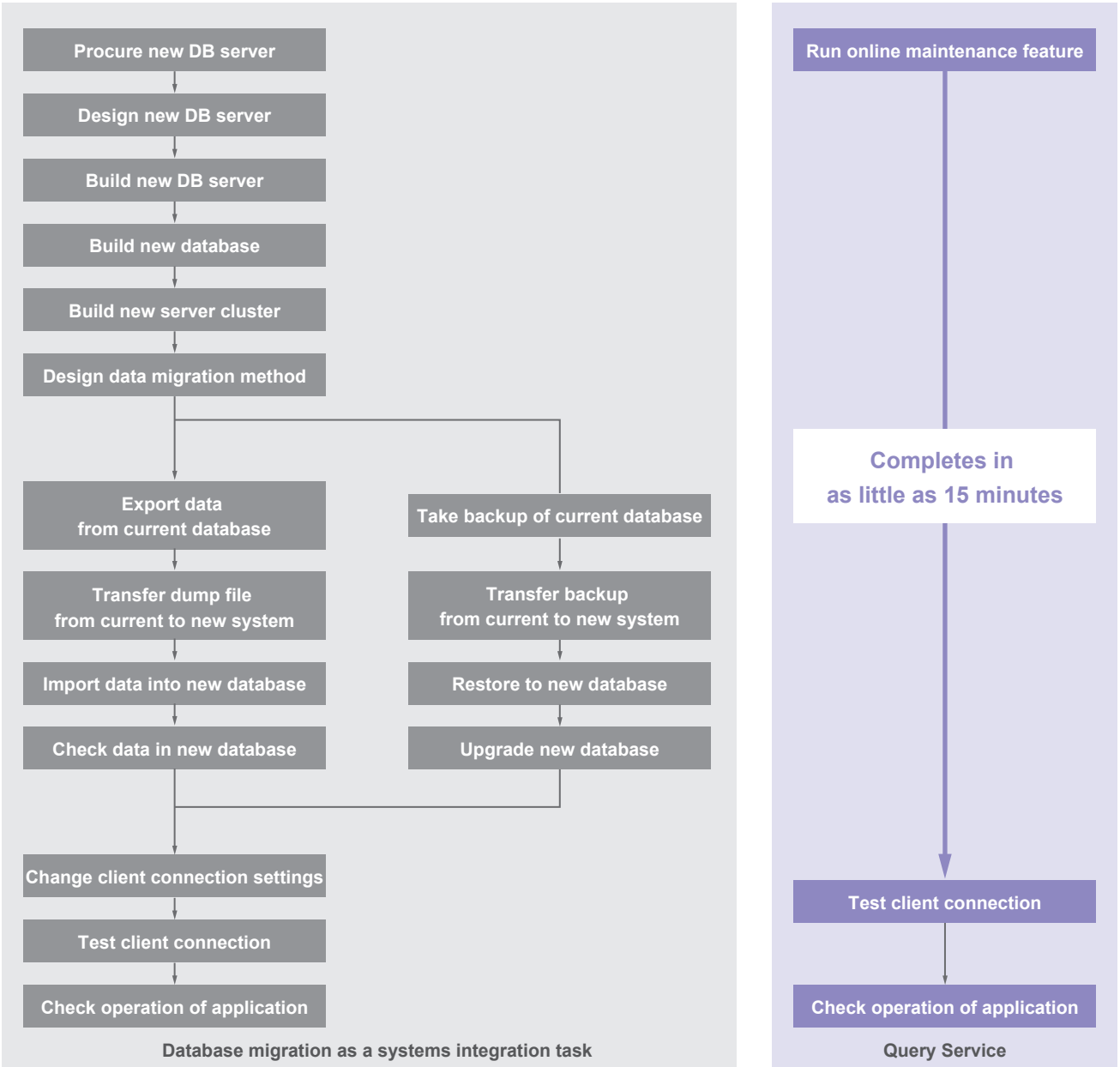

Figure 13: The Service Update Feature Greatly Streamlines the Platform Renewal Process

3.3 Conclusion

With the query service, my aim was not to develop a database itself, but to develop an orchestration system designed to make it easy for engineers to interact with databases, and while it is a prototype, I believe it has achieved that aim. Some readers may sense that I have a bone to pick with Kubernetes, but I actually rather like Kubernetes, and I would like to develop a query service using Kubernetes if I get the chance.

The Tech Challenge was different from the normal development routine guided by user requirements. I was able to bring my own ideas to life, and it was a most engaging and stimulating time for me as an engineer. It was a year that brought back all the simple joys of working with computers I had long forgotten since becoming a serious, working-age adult.

Tsutomu Ninomiya
Technical Manager, Service Planning Office, IIJ System Cloud Division.
Mr. Ninomiya is engaged in the planning and development of cloud services as well as technology support for projects in this area. He was originally a DWH/BI developer, and he likes SQL and parallel processing.

February 2021 Vol.49

Internet Initiative Japan

About Internet Initiative Japan Inc. (IIJ)

IIJ was established in 1992, mainly by a group of engineers who had been involved in research and development activities related to the Internet, under the concept of promoting the widespread use of the Internet in Japan.
IIJ currently operates one of the largest Internet backbones in Japan, manages Internet infrastructures, and provides comprehensive high-quality system environments (including Internet access, systems integration, and outsourcing services, etc.) to high-end business users including the government and other public offices and financial institutions.
In addition, IIJ actively shares knowledge accumulated through service development and Internet backbone operation, and is making efforts to expand the Internet used as a social infrastructure.

IIJ-MKTG020-0047

Internet Initiative Japan Inc.

Address: Iidabashi Grand Bloom, 2-10-2 Fujimi, Chiyoda-ku,
Tokyo 102-0071, Japan
Email: info@iij.ad.jp URL: https://www.iij.ad.jp/en/