IIR

Internet
Infrastructure
Review

Dec.2020
Vol.49

定期観測レポート

IIJインフラから見たインターネットの傾向～2020年

フォーカス・リサーチ(1)

耐量子計算機暗号の動向2020

フォーカス・リサーチ(2)

クエリサービス～柔軟なマネージドデータベースサービスへの挑戦

IIJ
Internet Initiative Japan

Internet Infrastructure Review

December 2020 Vol.49

エグゼクティブサマリ

本稿を執筆中の12月8日、2つのニュースが目に入りました。1つ目は、中国の武漢で最初に確認された新型コロナウイルスの感染者が発症したとされる日から、ちょうど1年になるというニュース。2つ目は、今日（12月8日）から英国で新型コロナウイルスのワクチン接種が始まったというニュースです。2020年は新型コロナウイルス一色の1年となりましたが、年末になってようやく沈静化に向かう期待が持てるニュースに接することができました。

この1年で人の考え方や価値観が大きく変わったと感じます。数年間で起きる変化が1年で起きたようにも感じています。急激に変わった新しい考え方のもとでも、従来の利便性をそのままに、不便を感じることなく享受できているのは、ネットワーク、セキュリティ、AIなど、情報通信技術が大きく貢献しているのは言うまでもありません。一方で、新型コロナウイルスによって、健康や生活に大きな支障が出た方が多くいらっしゃいます。また、今でも多数の医療関係者が、最前線で危険と向き合いながら職務を継続されています。そういった社会の痛みを和らげるうえでも情報通信技術が役に立てるよう、情報通信産業に従事する者として努めていきたいと思います。

「IIR」は、IIJで研究・開発している幅広い技術を紹介しており、日々のサービス運用から得られる各種データをまとめた「定期観測レポート」と、特定テーマを掘り下げた「フォーカス・リサーチ」から構成されています。

1章の「定期観測レポート」は、IIJインフラから見るインターネットの傾向の2020年版です。インターネット上のIPv4及びIPv6経路数、利用者に提供しているフルリゾルバから得られるDNSのクエリの解析、IPv6及びモバイルのトラフィック、RPKIを利用したBGP ROVの導入について、レポートを作成しました。新型コロナウイルスのインターネットトラフィックへの影響は本レポートのVol.47（https://www.iij.ad.jp/dev/report/iir/047.html）、Vol.48（https://www.iij.ad.jp/dev/report/iir/048.html）でも取り上げていますが、今回の分析では、過去の分析とは違った傾向も出ています。

2章の「フォーカス・リサーチ」では、耐量子計算機暗号の最新動向を紹介します。耐量子計算機暗号については、2016年6月に本レポートのVol.31（https://www.iij.ad.jp/dev/report/iir/031.html）の「フォーカス・リサーチ」でも取り上げており、今回の記事はそのアップデート版となります。2016年から始まっている米国NISTによる耐量子計算機暗号のコンペティションの状況に加えて、耐量子計算機暗号の安全性や量子暗号と耐量子計算機暗号の違いなどを解説しています。CRYPTRECの委員でもある筆者ならではの洞察やエピソードも含まれており、現代のインターネットを支える暗号技術に関心のある方には興味深く読んでいただけると思います。

3章の「フォーカス・リサーチ」は、筆者のアイデアによる「クエリサービスの開発」についてのレポートです。アプリケーションがコンテナ型仮想インフラを利用して、従来の課題を解決するなか、データベースに関しては、データの永続性・可用性・パフォーマンス面に課題があり、最適解といえる手法がありません。そこで、データベースについても、コンテナ型仮想インフラの特徴である柔軟性を実現すると共に、データの永続性・可用性をもたらすサービスを企画・実装したのが、本稿でご紹介する「クエリサービス」です。まだプロトタイプですが、今後のサービス化も検討していきたいと考えています。

IIJは、このような活動を通してインターネットの安定性を維持しながら、日々、改善・発展させていく努力を行っています。今後も企業活動のインフラとして最大限に活用いただけるよう、様々なサービスやソリューションを提供し続けてまいります。

島上 純一（しまがみ じゅんいち）

IIJ 取締役 CTO。インターネットに魅かれて、1996年9月にIIJ入社。IIJが主導したアジア域内ネットワークA-BoneやIIJのバックボーンネットワークの設計、構築に従事した後、IIJのネットワークサービスを統括。2015年よりCTOとしてネットワーク、クラウド、セキュリティなど技術全般を統括。2017年4月にテレコムサービス協会MVNO委員会の委員長に就任。

IIJインフラから見たインターネットの傾向 ～2020年

インターネットサービスを提供するIIJは、国内でも有数規模のネットワーク・サーバインフラを運用しています。ここでは、その運用によって得られた情報からこの1年間のインターネットの動向について報告します。今回は、BGP経路、DNSクエリ解析、IPv6、モバイルの各視点から変化の傾向を分析しました。またIIJバックボーンにおけるBGP ROV導入前の状況についても解説します。

Theme 01

BGP経路

最初に、IIJ網から他組織に広報している「IPv4フルルート」の情報（表-1）及び「IPv4フルルート」に含まれるunique IPv4アドレス数の情報（表-2）を確認します。なおこの1年の間にRIPE NCC及びLACNICのIPv4アドレス在庫が完全枯渇を迎えています。残るはAPNICとAfriNICですが、APNICは既に2021年初めの完全枯渇予測が出ており、またAfriNICでは2020年1月から割り振り/割り当てサイズに上限（/22）が設けられています。

経路の増加数には2018年をピークに減少傾向が見えますが、総数は80万を超えました。/22の経路数も10万超となりましたが/22-/24の3プレフィクスが経路総数に占める割合は80.9%と微増に留まっています。一方でunique IPv4アドレス数は、2011年以降で初の減少となった昨年（2019年）よりは増加したものの、まだ2018年及び2017年の値を下回っています。こちらも2018年がピークとなるのか、今後の推移に注目したいと思います。

次に「IPv6フルルート」の情報を確認します（表-3）。なお2019年11月にはARINに対して/12ブロックの最初の追加割り振りが行われています（同6月のRIPE NCCに続いて2番目）。

経路総数は昨年とほぼ同じ伸びを示し9万を超えました。来年10万超となっているのは間違いないと思えます。また総数の50.0%、増加数の58.1%は/48の経路であり、エンドサイトへのIPv6導入も順調に進んでいるものと推測されます。

表-1「IPv4 フルルート」に含まれるプレフィクス長ごとの経路数の推移

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

表-2「IPv4 フルルート」に含まれる unique IPv4アドレス総数の推移

年月	IPv4 アドレス数
2011年9月	2,470,856,448
2012年9月	2,588,775,936
2013年9月	2,638,256,384
2014年9月	2,705,751,040
2015年9月	2,791,345,920
2016年9月	2,824,538,880
2017年9月	2,852,547,328
2018年9月	2,855,087,616
2019年9月	2,834,175,488
2020年9月	2,850,284,544

最後に「IPv4/IPv6フルルート」広報元AS(Origin AS)数を確認します(表-4)。なおこの1年の間に、RIPE NCCに3072、LACNICに2048の32-bit only AS番号が追加割り振りされています。

16-bit AS番号Origin ASの減少及び32-bit only AS番号Origin ASの増加は共に昨年と同程度あり、Origin AS総数の4割を32-bit only ASが占めるに至っています。またIPv6経路を広報するAS ("IPv6-enabled")は全体の28.1%と順調に増加しています。来年は3割を大きく超えてくるか否か、こちらも注目したいと思います。

Theme 02

DNSクエリ解析

IIJでは利用者がDNSの名前解決を利用できるようフルリゾルバを提供しています。この項目では名前解決の状況を解説し、IIJで2020年9月30日に行ったフルリゾルバの1日分の観測データから、主にコンシューマサービス向けに提供しているサーバのデータに基づいて分析と考察を行います。

フルリゾルバはrootと呼ばれる最上位のゾーン情報を提供する権威ネームサーバのIPアドレスを手がかりとして問い合わせを行い、適宜権威ネームサーバをたどって必要なレコードを探します。フルリゾルバで毎回反復問い合わせを行っていると負荷や遅延が問題となるため、得られた情報はしばらくキャッシュしておいて再び同じ問い合わせを受けた場合にはそのキャッシュから応答しています。最近はこの他にもブロードバンドルータやファイアウォールなど、通信経路上の機器にDNS関連の機能が実装されており、DNS問い合わせの中継や制御ポリシーの適用に関わっている場合があります。また、Webブラウザなど一部のアプリケーションでは独自の名前解決機能を実装している場合があり、OSの設定に依存しない名前解決を行っている場合もあります。

表-3「IPv6 フルルート」に含まれるプレフィクス長ごとの経路数の推移

年月	/16-/28	/29	/30-/31	/32	/33-/39	/40	/41-/43	/44	/45-/47	/48	total
2011年9月	68	13	22	3530	406	248	45	87	95	2356	6870
2012年9月	102	45	34	4448	757	445	103	246	168	3706	10054
2013年9月	117	256	92	5249	1067	660	119	474	266	5442	13742
2014年9月	134	481	133	6025	1447	825	248	709	592	7949	18543
2015年9月	142	771	168	6846	1808	1150	386	990	648	10570	23479
2016年9月	153	1294	216	8110	3092	1445	371	1492	1006	14291	31470
2017年9月	158	1757	256	9089	3588	2117	580	1999	1983	18347	39874
2018年9月	168	2279	328	10897	4828	2940	906	4015	2270	24616	53247
2019年9月	192	2671	606	12664	6914	3870	1566	4590	4165	34224	71462
2020年9月	205	3164	641	14520	9063	4815	2663	5501	4562	45160	90294

表-4「IPv4/IPv6 フルルート」の広報元AS数の推移

AS番号	16-bit(1〜64495)					32-bit only(131072〜4199999999)				
広報経路	IPv4+IPv6	IPv4のみ	IPv6のみ	total	(IPv6 -enabled)	IPv4+IPv6	IPv4のみ	IPv6のみ	total	(IPv6 -enabled)
2011年9月	4258	32756	115	37129	(11.8%)	90	1278	13	1381	(7.5%)
2012年9月	5467	33434	125	39026	(14.3%)	264	2565	17	2846	(9.9%)
2013年9月	6579	34108	131	40818	(16.4%)	496	3390	28	3914	(13.4%)
2014年9月	7405	34555	128	42088	(17.9%)	868	4749	55	5672	(16.3%)
2015年9月	8228	34544	137	42909	(19.5%)	1424	6801	78	8303	(18.1%)
2016年9月	9116	33555	158	42829	(21.7%)	2406	9391	146	11943	(21.4%)
2017年9月	9603	32731	181	42515	(23.0%)	3214	12379	207	15800	(21.7%)
2018年9月	10199	31960	176	42335	(24.5%)	4379	14874	308	19561	(24.0%)
2019年9月	10642	31164	206	42012	(25.8%)	5790	17409	432	23631	(26.3%)
2020年9月	11107	30374	229	41710	(27.2%)	7653	19668	574	27895	(29.5%)

ISPは接続種別に応じてPPPやDHCP、RA、PCOなどの通知手段を利用してフルリゾルバのIPアドレスを利用者に伝え、利用者端末が名前解決用のフルリゾルバを自動設定できるようにしています。ISPは複数のフルリゾルバを利用者に伝えられるほか、利用者は自身でOSやWebブラウザなどの設定を変更して利用するフルリゾルバを指定、追加することもできます。端末に複数のフルリゾルバが設定されている場合、どれを利用するかは端末の実装やアプリケーションに依存するため、フルリゾルバ側では利用者が総量としてどの程度の問い合わせを行っているか分かりません。このため、フルリゾルバでは問い合わせ動向を注視しながら、常に処理能力に余裕を持たせて運用する必要があります。

IIJが提供するフルリゾルバの観測データを見てみると、利用者の利用傾向を示すように時間帯によって問い合わせ量が変動し、朝4時半頃に問い合わせ元のIPアドレス当たり最小の0.06query/sec、昼12時半頃にピークを迎えて0.24query/sec程度になっています。この値は昨年とほぼ同様ですが、早朝から日中体に掛けて+0.01ポイントと若干上昇しています。

問い合わせ傾向を通信に使われたIPv4とIPv6のIPプロトコル別に見てみると、昨年と比較してIPv4でのIPアドレス当たりの問い合わせが減っています。深夜帯に大きな変化はありませんが、日中帯や夕方夜間に渡って最大-0.03ポイント減少しています。一方でIPv6では、IPアドレス当たりの問い合わせは深夜帯も含めた全時間帯で+0.03ポイント前後増えています。これはIPv6対応した機器が家庭などに徐々に導入されたり、既存機器が置き換えられたりしていることを示唆していると考えています。また全体の問い合わせ数で見ると、IPv6による問い合わせが、問い合わせ元IP数、実際の問い合わせ数共にIPv4よりも多くなっています。IPv6による問い合わせ数は増加傾向にあり、昨年は全体の60%、今年は3ポイント増えて、全体の約63%がIPv6による問い合わせとなっています。

近年の特徴的な傾向として、朝方の毎正時などキリの良い時刻に一時的に問い合わせが増加しています。問い合わせ元数も同時に増えていますし、特に朝7時に顕著に傾向が見られるため、利用者の端末でタスクをスケジュールしたり、目覚まし機能などで端末が起動することに伴う機械的なアクセスが原因ではないかと推測しています。昨年は毎正時の14秒前と10秒前に問い合わせが増加していましたが、今年はそれに加えて毎正時の20秒前にも問い合わせが増加しています。毎正時では増加後、緩やかに問い合わせ量が減っていくのに比べて、毎正時前の増加では直ちにそれまでの問い合わせ量程度に戻っています。つまり多くの端末が綺麗に同期して問い合わせを行っていることから、何かすぐに完了する軽量なタスクが実行されているのではないかと推測しています。例えば接続確認や時刻同期など基本的なタスクを本格的なスリープ解除前に終わらせるような機構があり、これに利用されている問い合わせが影響していると考えられます。

問い合わせレコードタイプに注目すると、ホスト名に対応するIPv4アドレスを問い合わせるAレコードとIPv6アドレスを問い合わせるAAAAレコードがほとんどを占めています。AとAAAAの問い合わせ傾向は通信に利用されるIPプロトコルで違いが見られ、IPv6での問い合わせではより多くのAAAAレコード問い合わせが見られます。IPv4での問い合わせでは、全体の79%程度がAレコード問い合わせ、15%程度がAAAAレ

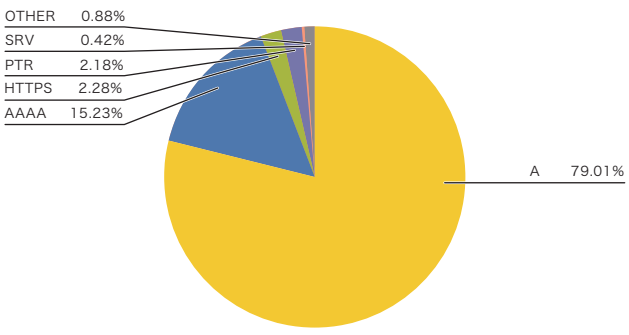

図-1 クライアントからのIPv4による問い合わせ

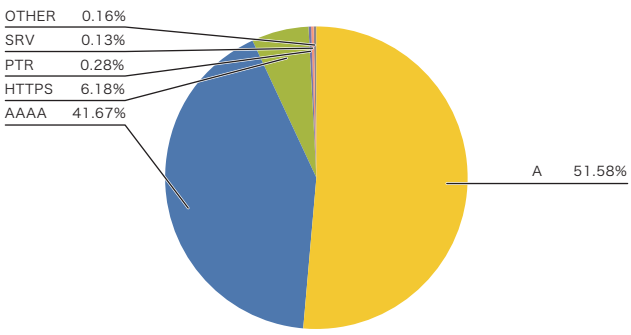

図-2 クライアントからのIPv6による問い合わせ

コード問い合わせです(図-1)。一方IPv6での問い合わせでは、全体の51%程度がAレコード問い合わせ、41%程度がAAAAレコード問い合わせとAAAAレコード問い合わせの比率が高まっています(図-2)。昨年と比べるとIPv4で5ポイント、IPv6で3ポイント程度Aレコードの問い合わせが減っています。今年は新しく実装されたばかりのHTTPSタイプのDNS問い合わせがIPv4で2%、IPv6で6%程度を占め、AとAAAAに次ぐ問い合わせ量になっています。この問い合わせは現状でApple社のiOS 14などがサポートしているようで、今後実装が広がるに連れて徐々に増加すると予想しています。

Theme 03

IPv6

ここではIIJバックボーンのIPv6トラフィックの流量、送信元AS、主なプロトコルについて見ていきます。

■ トラフィック

昨年同様、IIJのコアPOP(東京・大阪・名古屋)のバックボーンルータで計測した、IPv4トラフィックとIPv6トラフィックを図-3に示します。期間は2019年10月1日から2020年9月30日までの1年間です。

2020年は、年初からのCOVID-19(新型コロナウイルス)の影響もあり、昨年までとは異なったトラフィック傾向が観測されています。2020年2月くらいまでは大きな変化は見られなかったのですが、3月以降、COVID-19による休校や緊急事態宣言に伴う外出自粛が始まると、IPv4トラフィックが大きく伸びる結果となりました。本レポートのVol.48(https://www.iij.ad.jp/dev/report/iir/048.html)で触れられているように、モバイル系トラフィックは自粛期間中は減少し、固定系ブロードバンドサービスや法人VPNサービスが増加することとなり、それに伴いIPv4トラフィックが伸びたと考えられます。

今回は計測期間中の増減を相対的に把握するため、計測開始日(2019年10月1日)のIPv4、IPv6それぞれのトラフィックを1として正規化したグラフを作成(図-4)してみました。グラフ中程の4月から5月あたりが緊急事態宣言による外出自粛期間ですが、この期間はIPv4トラフィックが約8%増え、IPv6トラフィックは減少しているように見えます。緊急事態宣言終了後

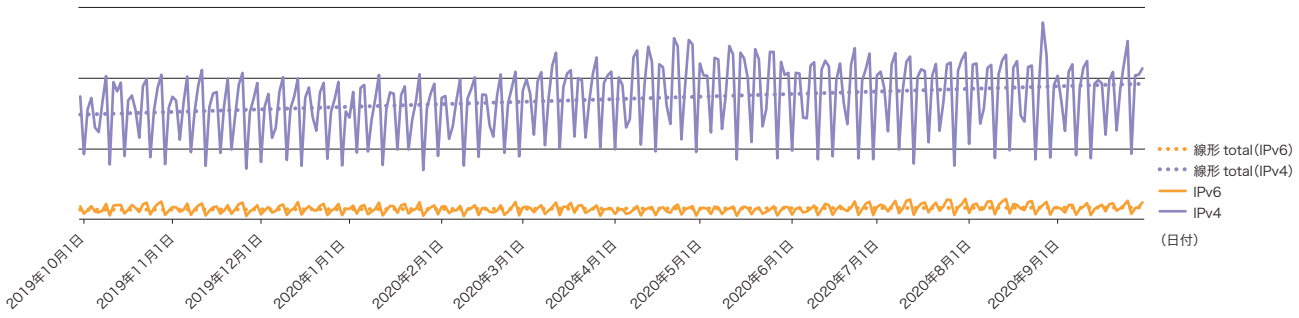

図-3 IPv4トラフィックとIPv6トラフィック(2019年10月～2020年9月)

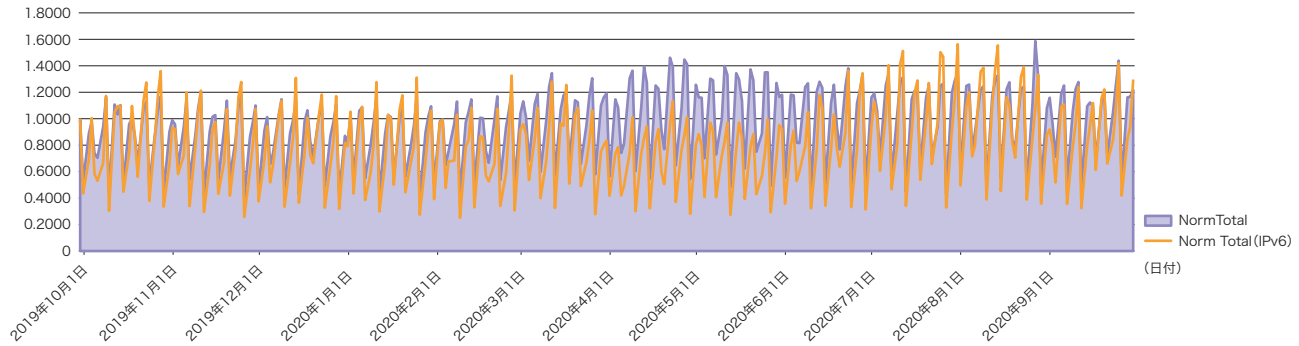

図-4 期初を1とした場合のIPv4トラフィックとIPv6トラフィックの増減

はIPv4トラフィックの増加は落ち着き、計測開始時と比べると微増、IPv6トラフィックも最終的には微増となっています。

IPv6が全体に占める比率は、図-5のとおり、緊急事態宣言中は低下しているが、最終的に期初と同じくらいの比率に戻る結果となりました。

■ 送信元組織(BGP AS)

次に、2019年10月から2020年9月までの1年間の、IPv6とIPv4の平均トラフィック送信元組織(BGP AS番号)の上位を図-6と図-7に示します。

やはり最上位はA社ですが、占有率は前回比で5ポイント下がりました。大きく伸びているはIIJのAS番号を送信元に持つトラフィックで、計測点による特殊事情もあるかもしれませんが、昨年同様JOCDNプラットフォームの動画配信トラフィックでIPv6が伸びていることが主要因と考えられます。その他は特に大きな動きは観測されず、目立った傾向はありませんでした。

■ 利用プロトコル

IPv6トラフィックのProtocol番号(Next-Header)と送信元ポート番号で解析したグラフを図-8に、IPv4トラフィックのProtocol番号と送信元ポート番号のグラフを図-9に示します。期間は2020年10月5日(月)から1週間です。

IPv6の1位はTCP443(HTTPS)、2位はUDP443(QUIC)となっており、HTTP系暗号化プロトコルで8割を超えるほどです。今回特徴的なのはTCP80 (暗号化なしHTTP)が4位に落ち、ESP (IPSec暗号化)が3位に入ってきていることでしょうか。ESPは平日日中帯により多く観測されており、土日は少ないことから、企業ネットワークにおいて、IPv6 VPNの利用が増加している

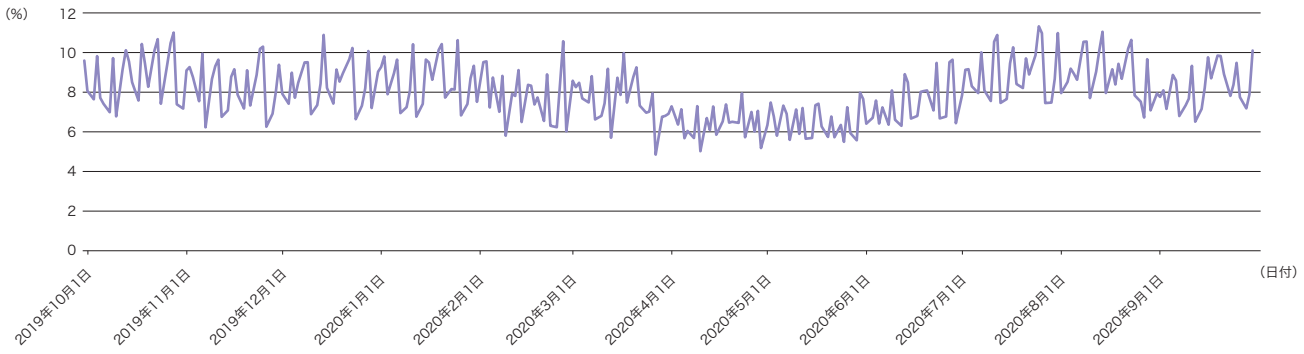

図-5 トラフィック全体に占めるIPv6の比率

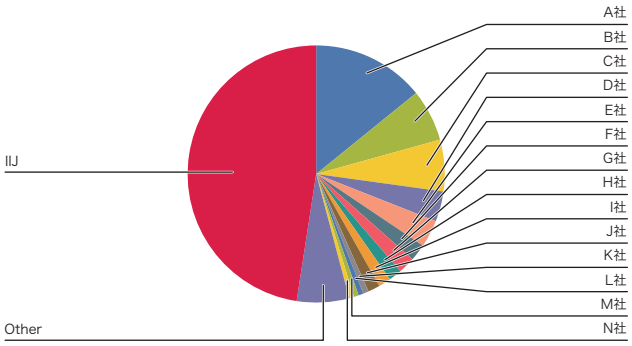

図-6 IPv6の平均トラフィック送信元組織(BGP AS番号)別占有率

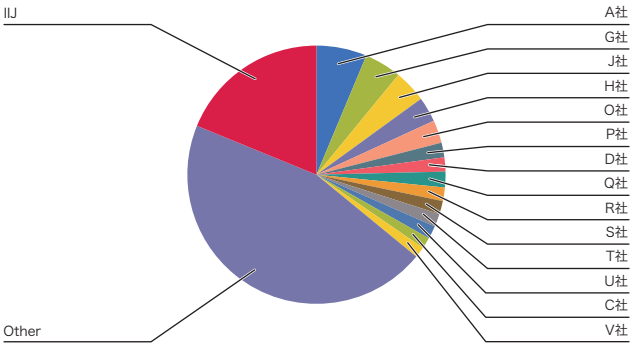

図-7 IPv4の平均トラフィック送信元組織(BGP AS番号)別占有率

ものと思われます。平日昼の最も多い時間帯に全体の19%程を占めており、日中はQUICを逆転する程です。なお、グラフで見て取れるのは5位までであり、6位以下は量が少なく、グラフで確認することが困難なほどの量しかありません。

IPv4については利用プロトコルに大きな変動はないようですが、夜のピークトラフィックが若干減り、日中のトラフィックが全体的に増加しているように見えます。リモートワーク（在宅勤務）によって日中自宅で仕事をする人が増え、日中トラフィックが増えた、もしくは、転送量のピークが早い時間にシフトしているように思えます。また、平日のグラフの形（1 ～5番目の山）と土日の形（6、7番目の山）が非常に似た形となっており、平日と休日のトラフィック傾向に差が少なくなってきているようです。

■ まとめ

IPv6のトラフィック量、送信元AS、利用プロトコルについて見てきました。今回はCOVID-19の影響もあり、これまでとは違った傾向が見られました。IPv6トラフィックの割合については、最終的には前回と大きく変わらない利用率でしたが、途中外出自粛の影響が見られました。IPv6の利用プロトコルやIPv4のトラフィックの山など、リモートワーク（在宅勤務）の影響と思える新たな動きも見られました。

COVID-19の流行については、まだまだ収束の気配を見せていません。ウィズコロナやアフターコロナと言われる世の中において、IPv6が、またインターネットの利用形態がどのように変化していくのか、引き続き見ていきたいと思います。

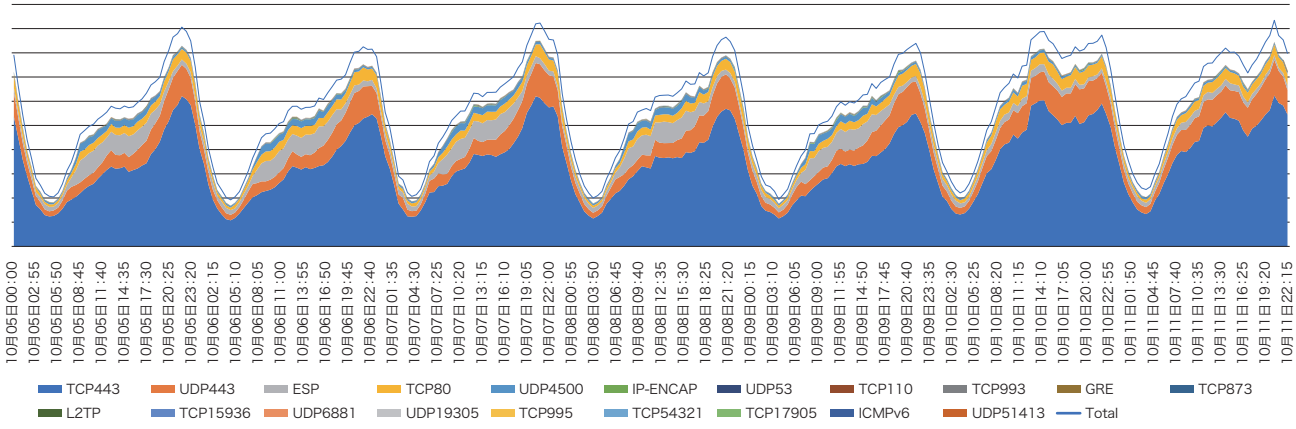

図-8 IPv6トラフィックのProtocol番号・送信元ポート番号別推移

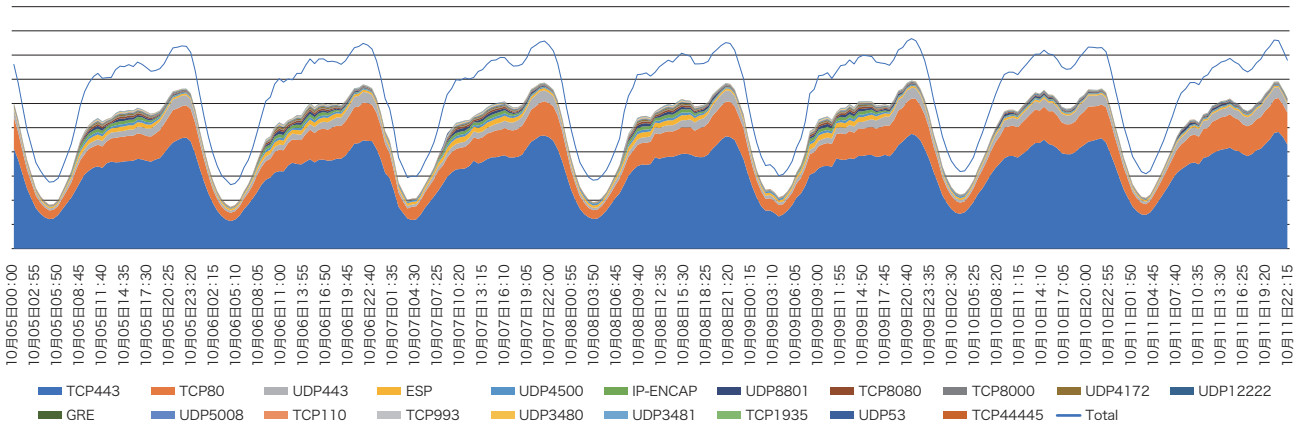

図-9 IPv4トラフィックのProtocol番号・送信元ポート番号別推移

Theme 04

モバイル 3G、LTEの状況

今年はモバイルのトラフィック傾向に関しては、本レポートのVol.48（https://www.iij.ad.jp/dev/report/iir/048.html） で触れているとおり例年とは異なり、新型コロナウイルスの影響による外出自粛期間中に大きく落ち込みました。また、モバイルの世界では“5G”という言葉が世の中に広く知られる状況になり、MNO各社が5Gのサービスを開始し、IIJでも法人向けに提供しているIIJモバイルサービスにおいてau 5Gに対応したサービスを10月30日から提供開始しました。新しい規格の仕組みが世の中に浸透していく一方で、古い規格は終息に向かっていく現状もあります。2019年10月にNTTドコモが3G通信サービス「FOMA」を2026年3月末で終了すると発表しました。

今回はIIJで提供しているモバイルサービスにおける3Gトラフィックの傾向について解説します。対象期間は2019年10月1日から2020年9月30日です。

全体トラフィックにおける3Gの割合は図-10のとおりです。

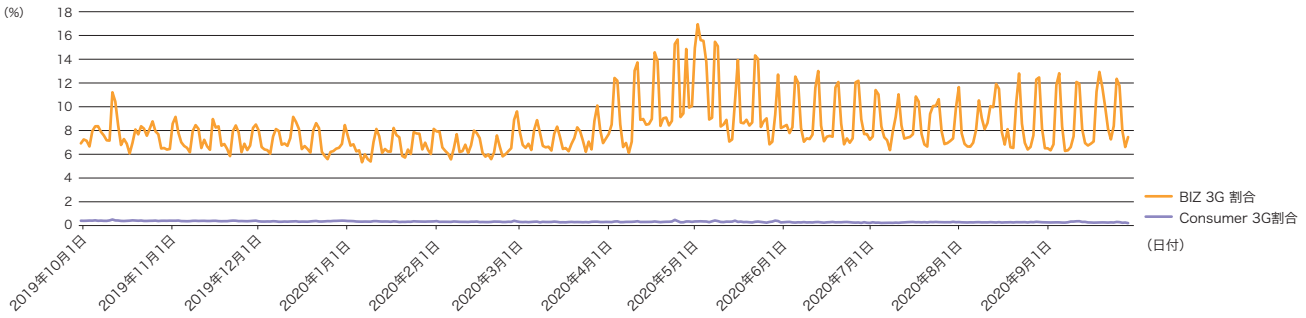

図-10 全体トラフィックにおける3Gの状況

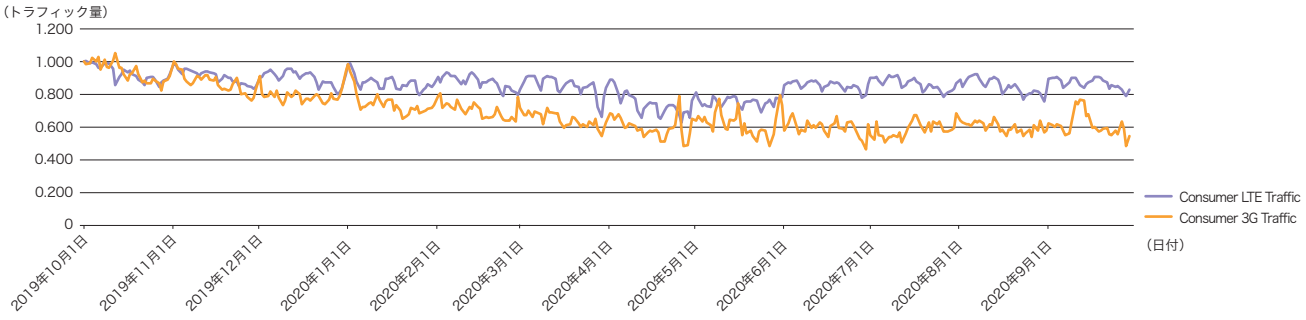

図-11 コンシューマ向けトラフィック量の傾向

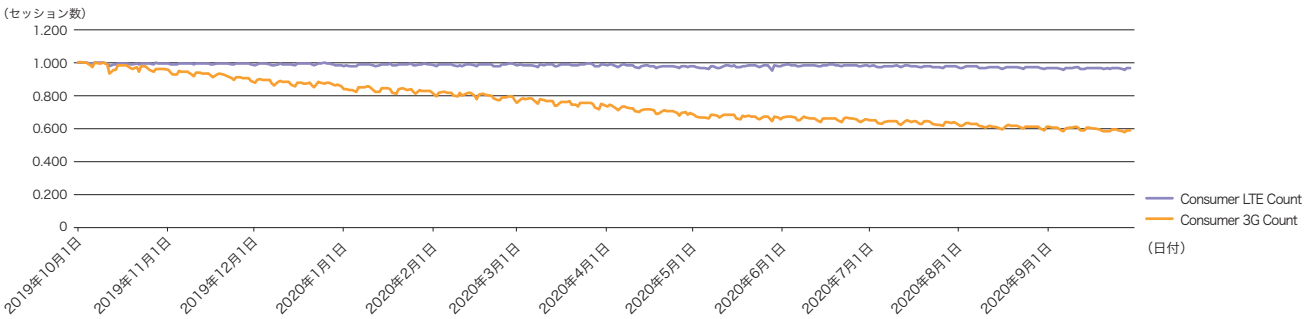

図-12 コンシューマ向けセッション数の傾向

コンシューマ向けサービスでは0.5%以下の状況になっていますので3Gはほぼ利用されていないと言えます。それに対して、法人向けサービスでは平均で8%程度のトラフィックが3Gとなっており、法人ユーザ様では3Gが根強く利用されていることが分かります。

次に、2019年10月1日を基準日としたときのコンシューマ向けサービスのトラフィック量(図-11)とセッション数(図-12)に関する傾向をグラフに示します。コンシューマ向けサービスの3Gトラフィックに関してはトラフィック量、セッション数共に右肩下がりとなっており、ここ1年間でトラフィック、セッション数共に40%程度減っています。要因はいろいろと考えることができますが、コンシューマサービスの接続端末はほぼスマートフォンであることを考えると、世間でのLTEの接続環境が向上しており、3Gに落ちることが少なくなっていることが考えられます。今後どのような推移を見せていくかは確認していきます。

コンシューマ向けサービスのLTEのトラフィック量については、セッション数ではほぼ横ばい状態でしたが、トラフィック量に関しては2020年3月から5月にかけて一時的に3割減という状態になりました。こちらはコロナ禍での自粛期間中の落ち込みと考えられます。

次に、同じ2019年10月1日を基準日としたときの法人向けサービスのトラフィック量(図-13)とセッション数(図-14)に関する傾向を見てみます。

法人向けサービスの3Gトラフィックに関してですが、セッション数は断続的に減少傾向となっています。これは3Gサー

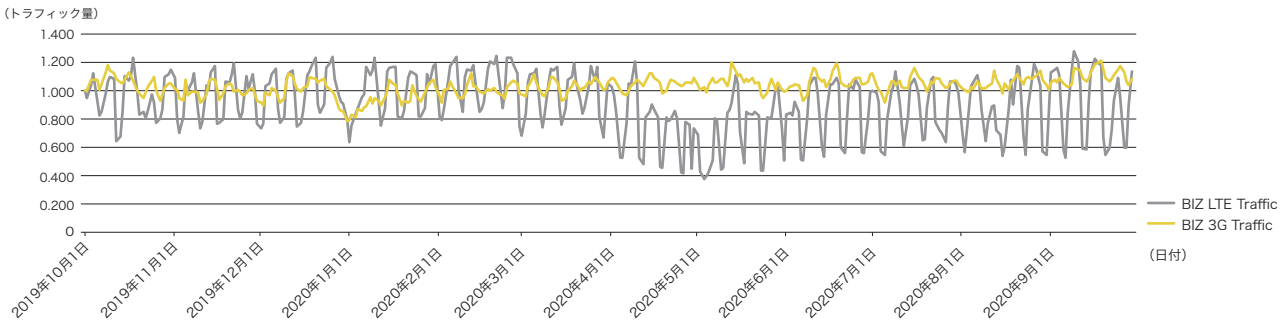

図-13 法人向けトラフィック量の傾向

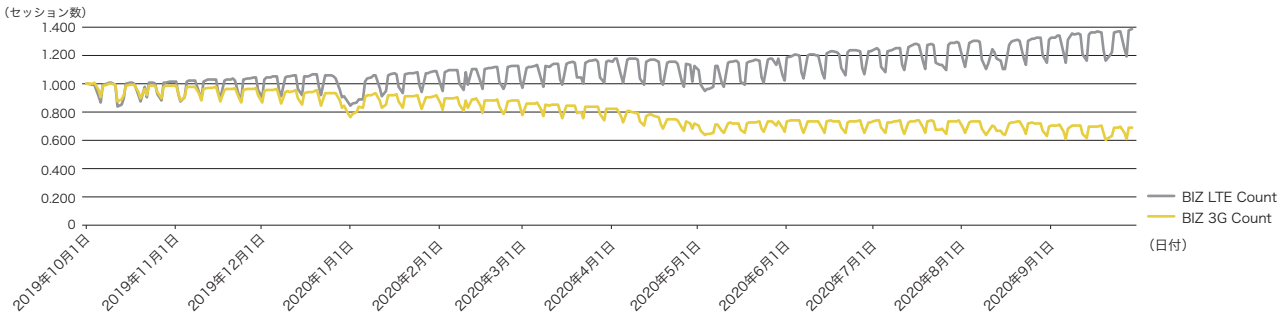

図-14 法人向けセッション数の傾向

ビス終了を見越して、3GからLTEへの移行などが進められていることが要因の1つと考えられます。また、2019年10月から“2020年4月まで”と“2020年5月以降”では減少の傾きが緩やかになっていますが、これはコロナ禍の影響によって移行などのスピードが緩やかになったことが1つの要因と考えられます。それに対して3Gのトラフィック量はコロナ禍の影響もなく緩やかな増加傾向が見られます。この点については法人ユーザ様の利用状況に依存することがあるため、今後の状況を注視していこうと思います。

また、LTEのトラフィック傾向ですが、セッション数では3Gと同様に自粛期間中の落ち込みはあるものの、断続的な増加傾向にあります。またトラフィック量に関しても自粛期間である2020年4月～5月を底として、いまは徐々に戻ってきている状況にあります。

最後に、5Gの状況に関してです。最初に申し上げたとおりIIJでは10月30日からau 5Gを利用した法人サービスをリリースしました。そのため、トラフィック傾向などはまだ分析できる状況ではないのですが、IIJユーザ様がどの程度5Gに対応した端末を利用しているか調べてみました。

図-15のグラフは、「端末名に5Gと入っているAndroid端末」と「iPhone12シリーズ」という5G対応と思われる端末が2019年10月1日を基準日としたときの増加率を示したものです。iPhone12がリリースされる前までは1年間で40倍程度の増加でしたが、リリース後には1年前の約400倍（リリース直前に比べ約10倍）に増加しています。今後は5Gサービスに関するトラフィック傾向にも注視していきたいと思います。

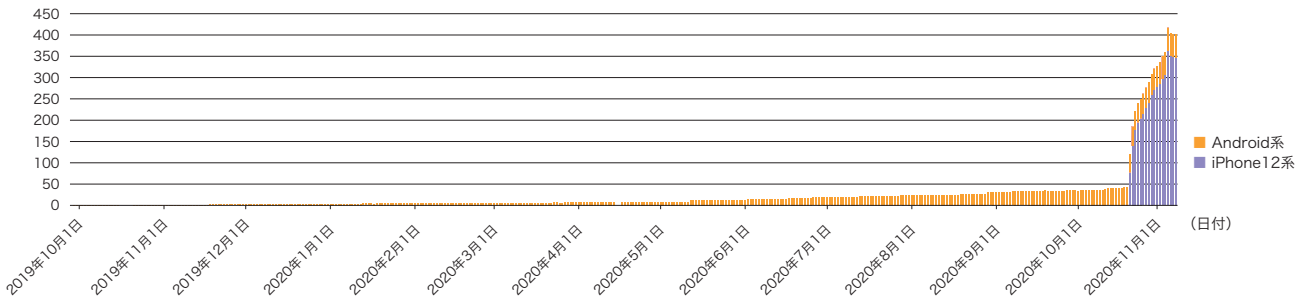

図-15 5G対応端末の接続状況

Theme 05

IIJバックボーンにおけるBGP ROV導入

IIJでは2020年の11月以降、インターネットバックボーンにおけるRPKIを利用したBGP ROV（Route Origin Validation）を順次導入していきます。

RPKIなどの導入や仕組みについては、導入前にエンジニアブログでも紹介していますので、併せて参照ください。RPKIで利用するROAの発行数などの推移はRIPEやNISTなどで確認できます。RPKI自体は2008年頃からRIRで利用されており、JPNICでも2015年には登録が開始できるようになっています。ROA数はRIPEが随分と先行していますが、ここ数年は他のRIRのROAも非常に増加しており、活用が着々と進んでいる状況にあるようです。

今回は、IIJでBGP ROVを適用する前の2020年10月の特定の日における情報を抽出しています。表-5は、2020年10月のIIJのROAキャッシュサーバから特定の日のVRP（Validated ROA Payloads）をエクスポートしたデータから作成しています。ROAには、自組織が広告するprefixと、自組織を示すoriginとなるAS、そして該当のprefixを最大どこまでのprefix長で受け入れるのかを示すmax lengthがセットで登録されています。BGP経路に対するROA登録アドレス数の割合は、IIJの特定の日時のBGP経路数のUniqueアドレス数に対するROAの登録prefixの割合を示しています。

次に、ROAに登録されているprefix長とmax lengthの分布を図-16、図-17に示します。

横軸は、prefix長で、縦軸は登録数を示しています。棒グラフがROAに登録されているprefix長で、ポイントで表示してあるのが最大で受け入れるprefix長であるmax lengthとなり

表-5 IIJ ROAキャッシュのVRPの情報

	IPv4	IPv6	合計数
Unique prefix数	144,785	25,085	169,870
Unique AS数	16,479	8,769	17,670
Unique Prefix+AS数	158,099	27,024	185,123
AS0のprefix登録数	184	100	284
BGP経路に対するROA登録アドレス数の割合	27.9%	32.8%	-

トランスアンカーはRIPE NCC, ARIN, APNIC, AfriNIC, LACNIC

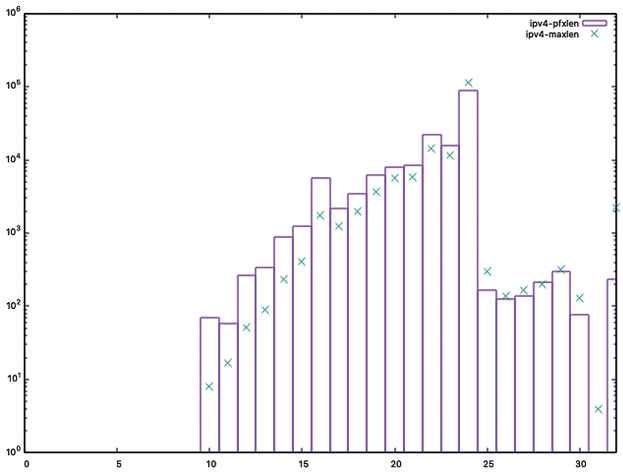

図-16 登録されているprefix長とmax lengthの分布（IPv4）

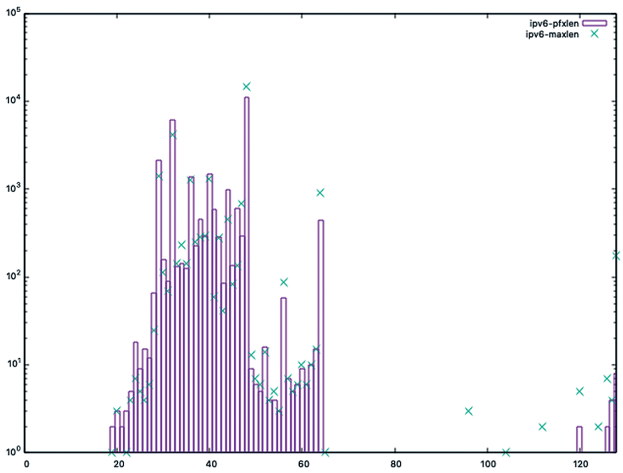

図-17 登録されているprefix長とmax lengthの分布（IPv6）

ます。登録prefix長とmax lengthの値が同じに設定してあるものは、IPv4で81.6%、IPv6で78.7%程度です。一方でmax prefixの方が長く設定してあるものは、IPv4は/24、IPv6は/48が多い傾向となっており、インターネットにおいて組織間でBGP経路を交換する際のトレンドと同じような状況であると推察します。

図-18、図-19は、上記で利用したVRPのデータを元にIIJの特定地域のBGP経路をValidationするとどの程度invalidな経路が存在するかを調査したものです。

Validとして判定されるのは、IPv4では24%、IPv6では29%程度であり、Invalidとして判定されるものは、IPv4では0.32%, IPv6では0.49%程度となりました。ROAに合致するprefixがないものはNotFoundとして示されており、全体の7割がそれにあたります。今後ROAの登録がすすむにつれて、NotFoundの割合は少なくなっていくと期待しています。また、Invalidな経路であっても、より大きい空間がValidやNotFoundとして判定されるケースがあるため、Invalidな経路として判定されたとしても、必ずしも到達性がなくなるわけではありませんが、そうでない経路も存在しており、IPv4の0.028%、IPv6の0.02%程度がそれにあたります。

インターネットは常に変化しているため、設定ミスや不具合に

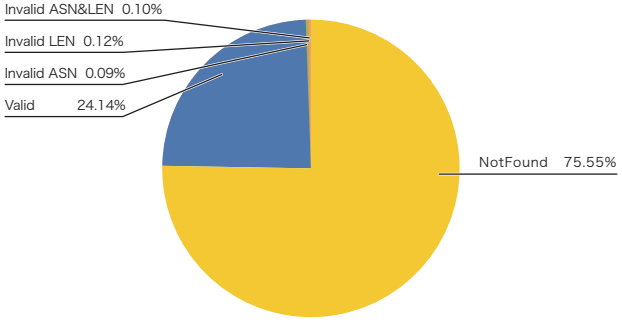

図-18 特定ルータのBGP経路のValidation結果の推察(IPv4)

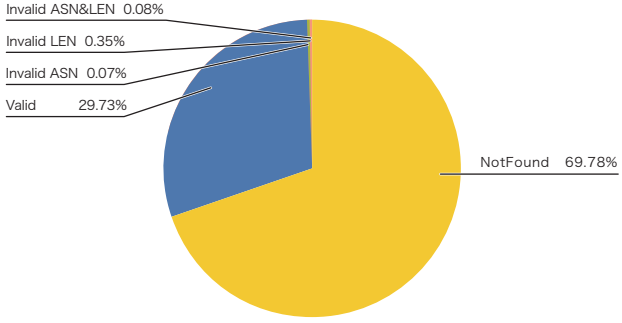

図-19 特定ルータのBGP経路のValidation結果の推察(IPv6)

よる様々な経路の問題の脅威に常にさらされている状況です。こうした経路が正当であるかどうかも判断に困るものが多数であるため、未然に防ぐことは非常に難しい状況でしたが、それが今回の取り組みを行うと同時にRPKIが浸透していくことによって、未然に一部の脅威を防ぐ可能性が高くなっていくことにつながります。

IIJでは今後もより快適なインターネット社会を支えるため堅牢なインフラの提供に向けた取り組みを実施していきます。

執筆者:

1.BGP経路
倉橋 智彦（くらはし ともひこ）
IIJ 基盤エンジニアリング本部 運用技術部 技術開発課

2.DNSクエリ解析
松崎 吉伸（まつざき よしのぶ）
IIJ 基盤エンジニアリング本部 運用技術部 技術開発課

3.IPv6
佐々木 泰介（ささき たいすけ）
IIJ 基盤エンジニアリング本部 ネットワーク技術部 副部長

4.モバイル 3G、LTEの状況
齋藤 毅（さいとう つよし）
IIJ 基盤エンジニアリング本部 ネットワーク技術部 モバイル技術課長

5.IIJバックボーンにおけるBGP ROV導入
津辻 文亮（つつじ ふみあき）
IIJ 基盤エンジニアリング本部 ネットワーク技術部 ネットワーク企画課長

耐量子計算機暗号の動向2020

本稿は2016年6月にIIR Vol.31フォーカスリサーチで取り上げた「1.4.3 耐量子暗号の動向」[*1]について2020年11月の時点でのアップデートに関して取り上げます。前回の報告では日本語訳として「耐量子暗号」という用語をあてて紹介しましたが、技術用語としてより正確性を期するために「耐量子計算機暗号」と呼ばれることが一般的となっており[*2]、本報告でもこの用語を使用することとします。

2.1 NISTコンペティション概要

前回報告では、有力なアルゴリズムとして以下の4つの数学的背景を持つカテゴリを紹介しました(IIR31 表-2 耐量子暗号の分類)。

- 格子暗号
- 符号ベース暗号
- 多変数公開鍵暗号
- ハッシュベース署名

これらのうちNISTコンペティション[*3]の最新報告でも分類で使われているカテゴリは、上記4つに加え、現在はIsogeny (同種写像)と呼ばれる暗号方式も追加されています。Isogenyに関しては2018年11月に大阪大学で開催されたECC2018[*4]でも多くの時間を割いて議論されており、Chloe Martindaleによる発表資料は美しく観ているだけでも楽しくなる図面が並んでいます。

ECDHなどの楕円曲線暗号では、公開パラメータとしてある固定された楕円曲線上の点に加法演算をうまく定義することで群構造を作り、楕円曲線上の離散対数問題という特性を用いてアルゴリズムを構成しています。ここでの安全性は点PをK回加算した点QをQ=kPとしたときPとQ=kPからkを求めることが難しいことに依拠しています。Isogenyは楕円曲線から楕円曲線への写像の一種であり、楕円曲線EとE'=Φ(E)から写像 Φ を求めることが困難であることを用いてDiffie-Hellman鍵共有法と同様の数学的構造を持つ鍵共有方法が提案されています。

2016年2月に福岡で開催された国際会議PQCrypt2016の招待講演で、Dustin MoodyによってNISTからのアナウンスがあり、耐量子計算機暗号のコンペションが開催されることになりました[*5]。2016年末には応募要項が固まり、2017年11月の期限までに82のアルゴリズムが投稿されました。書類審査を経て69のアルゴリズムが第1ラウンドの候補として審査が開始されました[*6]。更に2018年4月には第1回NIST PQC Standardization workshopが開催され集中議論が進められた後、2019年1月にはNISTIR8240[*7]と共に26のアルゴリズムが第2ラウンドに進んだことが公表されました。

2019年8月にはCRYPTO2019と併設して第2回のNIST PQC Standardization workshopが開催され、2020年7月22日に7つのファイナリストと8つの代替候補(alternates)が第3ラ

＊1 Internet Infrastructure Review vol.31「1.4.3耐量子暗号の動向」(https://www.iij.ad.jp/dev/report/iir/031/01_04.html)。
＊2 森北出版より同名の書籍が発刊されている(https://www.morikita.co.jp/books/book/3503)。
＊3 NIST Post-Quantum Cryptography(https://csrc.nist.gov/Projects/Post-Quantum-Cryptography)。
＊4 ECC18(https://cy2sec.comm.eng.osaka-u.ac.jp/ecc2018/program.html)。Chloe MartindaleによるCSIDHに関する発表スライド、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)。

ウンドに進んだことがアナウンスされました。詳しい選定状況はNISTIR8309にてレポートされています[*8]。この辺りの背景はDustin本人による回顧録として2020年12月に公開されています[*9]。

表-1はDustinによる最新の発表資料[*10][*11]などから作成したファイナリストと代替候補のリストとなります(括弧つきで赤文字で示されたものは代替候補)。

ファイナリスト7つのうち署名は3方式(格子暗号2、多変数公開鍵暗号1)、KEM (鍵カプセル化メカニズム)は4方式(格子暗号3、符号ベース暗号1)となっています。それぞれのアルゴリズムは第3ラウンドに入った段階で若干の修正が認められており、第3ラウンドのページ[*12]に仕様やWebサイトに関する情報が記載されています。またUpdateについてもPDFが提供されています[*13]。

本コンペティションの今後の予定は以下となっています。既に開始されている第3ラウンドは12～18ヵ月の期間で完了することが予定されており、第3ラウンド終了時には、現在格子暗号にカテゴライズされている複数のファイナリストは署名・KEMでそれぞれ1つに絞るよう選定が行われます。第3回ワークショップは2021年春または夏に開催、標準化文書は2022～2023年にドラフト、2024年には標準化文書が発行される、というスケジュールで進められます。

Horizon 2020の予算の枠組みでプロジェクト化されたPQCRYPTO projectがあります[*14]。D5.2 Standardization: Final report によると同プロジェクトからNISTコンペティションへの貢献が大きいことが分かります。PQCRYPTO projectから応募されたアルゴリズムは、第3ラウンドに進んだ15のアルゴリズムのうち11方式を占めるなど大きな存在感を持っています。

2.2 NIST策定の暗号アルゴリズムとその影響

NISTは米国政府における調達要件にも大きく関わるような各種仕様及びガイドラインを作成しています。NISTが扱う技術領域は非常に広範囲ですが、主に我々が目にするところでは情報セキュリティに関するあらゆるガイドラインが多く取り上げられています。例えばSP800-63で規定されている、パスワードに関連するドキュメント類は多くの技術者の目に触れるところとなり、パスワードに関する考え方に関して議論のト

カテゴリ／方式	署名	KEM
格子暗号	CRYSTALS-DILITHIUM、FALCON	CRYSTALS-KYBER、NTRU、SABER(FrodoKEM、NTRU Prime)
符号ベース暗号	なし	Classic McEliece(BIKE、HQC)
多変数公開鍵暗号	Rainbow(GeMSS)	なし
ハッシュベース署名	(Picnic、SPHINCS+)	N/A
Isogeny	なし	(SIKE)

表-1 ファイナリストと代替候補のリスト

*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)。
*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)。

リガーとなりました。定期的にパスワード変更することの是非や2要素認証もしくは多要素認証におけるSMSメッセージを使った認証方式の是非に関して、有益なディスカッションが日本でも行われました。

また、暗号アルゴリズムの利用そのものやその利用に携わる技術者においては、NISTから発行されるFIPSやSPが世界的にもリードし続けていることを知る人も多いと思います。現在広く利用されているハッシュ関数の仕様群SHA-2もNISTによる策定が行われている一方で、米国政府の息がかかっていることからバックドアが存在するのではないかという懸念は1970年代に策定されたDES発行の頃から指摘されています。そのため最近ではNIST Curveと呼ばれる楕円曲線暗号方式に使われる公開パラメータにおける同様の懸念から、草の根活動としてのIETFで標準化された暗号方式を好む人たちが一定量いることも事実です。ただし暗号資産で利用されているブロックチェーンなどで実装されているハッシュ関数アルゴリズムはNISTに策定されたものも多く含まれています。暗号技術に精通していないエンジニアが独自に設計・実装したことで、暗号資産そのものの存続に関わる問題に発展した事例もあり、SHA-2は十分に精査された枯れた技術で安全であると認識されているようにも見受けられます。

2.3 耐量子計算機暗号における安全性の考え方

RSA暗号の安全性は素因数分解の困難性に依存していますが、例えば100程度までの合成数であれば誰でも簡単に素因数分解できます（例えば2、3、5、7、11、13くらいで割れるかを確認すれば良い）。RSA暗号は計算量的安全性に基づくため、ある程度大きな素数を用いなければ安全に利用することはできません。現在、1024ビット×2=2048ビット以上の合成数Nを用いることが推奨されていますが、この鍵長に関するコンセンサスは時代と共に変化してきました。この事例のように、暗号アルゴリズムそのものの安全性に依拠するパラメータ設定はとても重要となります。RSA暗号は現在も安全と認識されていますが、それは十分鍵長を保っているからであり、この前提のもと正しく実装されることで初めて安全に利用することができます。

耐量子計算機暗号においても同じようにパラメータ設定の課題があります。暗号方式そのものが安全と認識されていても、例えば鍵長に類する情報などがどのようなデータを利用すれば安全になるかという検討が必要です。こうした背景から、暗号アルゴリズム、特に現在利用されている計算量安全性に基づく公開鍵暗号方式は、鍵長が重要な事項となります。これと同様に、今後策定される耐量子計算機暗号アルゴリズムにおいても各種パラメータをどのように設定して安全性を確保するかが重要となります。そのため、一部のカテゴリに関しては現在の計算機環境において利用に適しているかどうかを判断するために解読コンペティションが開かれており、最新の攻撃手法を共有することで研究コミュニティを盛り上げる・維持するケースも見受けられます。

耐量子計算機暗号のうち多変数公開鍵暗号としてカテゴライズされる分野においては2015年からコンペティションが開催されており、このくらいのパラメータで十分と思われていた暗号アルゴリズムが急激な研究の進展によって我々が考えていたよりも安全でなくなった事例も多く見られます。例えばCRYPTO2019に併催された第2回PQC Standardization

Conference[*15]においてJintai Ding[*16]による講演で大きくパラメータ変更見直しが必要となりました。

2.4 ビットセキュリティ

耐量子計算機暗号とは対照的に、現在多く利用されている暗号アルゴリズムは古典的アルゴリズムと呼ばれています。古典的アルゴリズムにおいては、各種パラメータを選択することにより、あるビットセキュリティを確保することで安全性を担保するという考え方が一般的です。このビットセキュリティという考え方は共通鍵暗号やハッシュ関数などにおいては理解しやすい概念となっており、過去のIIRレポートにおいて暗号危殆化や等価安全性に関する解説を行っています[*17]。

例えば広く使われているAES-128と呼ばれる共通鍵暗号アルゴリズムでは128ビット長の鍵が使われており、復号のための鍵を同定するには2^{128}の施行を必要とするため、128ビット安全もしくは128ビットセキュリティを確保しているといわれています。ビットセキュリティの考え方は公開鍵暗号アルゴリズムにも適用することができ、ある鍵パラメータを用いた各種暗号アルゴリズムが、どの程度の安全性を確保できているかを比較対照できるようになっています。このような鍵長に関する対応表で有名なものとしてNISTによるSP 800-131A[*18]がよく取り上げられますが、同じRSA鍵長を用いてもステークホルダーによってどのくらいビットセキュリティと見なせるか少々ブレがあるのが面白いところです(例えば、本レポートのVol.8 1.4.1 表-1を参照)。

このように策定された団体や組織の思惑に依存して強度が変化する同じようなケースが、過去にも耐量子計算機暗号に関するレポートで見られました。前回の耐量子計算機暗号に関するレポートを執筆した際に紹介したGroverのアルゴリズムは、nビットセキュリティを確保する共通鍵暗号方式においてその鍵長の半分の強さしか持たないことが示されています。しかしステークホルダーによっては共通鍵暗号方式はすべてゼロビットセキュリティに低下するという判断を下しているレポートも見受けられました。現在は共通鍵暗号において、解読に必要な計算量2^nの肩に乗ったnの値が半減するという考え方が広く受け入れられています。NISTによりKEMやデジタル署名などの公開鍵暗号方式のコンペティションが行われているため、耐量子計算機暗号としての共通鍵暗号方式がフォーカスされることは少ないですが、いくつかの独立したペーパーが発行されており興味深い結果が得られています。例えば現在一般的に使われているAES暗号に対して量子計算機がどのくらい脅威となるかに関する文献があります[*19]。古典計算機でも量子計算機でも大きなセキュリティマージンを持っているとの解析結果を主張していますが、その見通しや判断は読者に任せることとします。

2.5 量子暗号と耐量子計算機暗号について

量子暗号と耐量子計算機暗号という2つのキーワードはその意味や背景が異なりますが、一般の方から見ると混乱を引き起こしているような記事が見受けられます。前者としては例えばQKDとして知られる量子通信における鍵配送などがありますが、耐量子計算機暗号とは異なる概念です。本稿で扱う技術対象は後者の耐量子計算機暗号と呼ばれる分野に関する話題であり、量子通信など量子力学を直接的に扱う技術的話題は行いません。

*15 Second PQC Standardization Conference(https://csrc.nist.gov/events/2019/second-pqc-standardization-conference)。
*16 Jintai Ding, New Attacks on Lifted Unbalanced Oil Vinegar(https://csrc.nist.gov/Presentations/2019/new-attacks-on-lifted-unbalanced-oil-vinegar)。
*17 暗号危殆化の事例や、ビット安全性、等価安全性に関する解説は本レポートのVol.8(https://www.iij.ad.jp/dev/report/iir/pdf/iir_vol08.pdf)の「1.4.1 暗号アルゴリズムの2010年問題」にて紹介している。
*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)。FSE2020にて発表(https://fse.iacr.org/2020/program.php)。

耐量子計算機暗号を議論する際に用いられる仮定は、量子計算機が今後実装され普及した際に現在使われている暗号アルゴリズムがどのような影響を及ぼすかという点が焦点となっています。そのため読者の中には耐量子計算機暗号が既にブラウザなどで実装されていることに非常に大きな驚きをもって捉えられるでしょう[*20]。報道などにより、既に量子計算機が商用で流通していると伝えられると、そのような量子計算機上で動作する暗号アルゴリズムが実装されていると勘違いされるかもしれません。しかし我々が扱うモデルは、量子計算機を持つ攻撃者のみが量子計算機を使い莫大な計算量を持っている、一方で大多数の方々が古典的計算機で使って利用しているという前提で攻撃が行われると理解していただければと思います。

2.6 耐量子計算機暗号の意味するところ

耐量子計算機暗号の定義は明瞭ではなく、ある提案アルゴリズムが耐量子計算機暗号かどうかの線引きは非常に難しいところではありますが、過去の一般的な暗号アルゴリズムで使われているような計算量的安全性の仮定ではないところに安全性を置いているアルゴリズムと考えれば良いということになります。つまり、古典的な計算機上でも実装可能なアルゴリズムが新たに取って替わると考えることができます。

そのためかなり昔に遡り、例えば1970年代に提案されている暗号アルゴリズムであっても現在それらが再評価され、耐量子計算機暗号としてフィーチャーされているものもあります。一方、ここ数年で急速に研究が進んだものもあり、暗号研究業界において1つの大きなトピックとしても認識されています。

暗号研究が進むトリガーとして、今まで普通に使われていたアルゴリズムが使えなくなるような攻撃が見つかる（これを暗号アルゴリズムの危殆化と呼ぶ）、あるいは今後使えなくなるような大きなインパクトのある攻撃が見込まれるといった2つの要因があります。後者の一例としては、例えばSHA-3と呼ばれる新しいハッシュ関数の策定があります。SHA-1やSHA-2ハッシュ関数の設計で見られるような数学的構造とは異なる設計方針が採択された提案アルゴリズムがNISTからFIPS仕様として標準化されています。しかしSHA-2はまだまだ安全に利用できると信じられており、SHA-3への移行はほとんど進んでいません。耐量子計算機暗号も後者に該当するものとして認識されており、危殆化が進んでいるというよりも将来に備えての準備を行っているという意味合いが強いです。

暗号アルゴリズムにおけるAgilityと呼ばれる考え方では、異なるアイデアやバックグラウンドに基づいた設計を持つ暗号アルゴリズムの「別の引き出しを持っておく」ことが重要とされており、今回のように耐量子計算機暗号の策定においても様々なバックグラウンドを持つアルゴリズムが勢揃いしていると考えることができます。その中でも2000年代から研究されてきた格子ベースの暗号アルゴリズムが有力であり、ファイナリストとして残留しているアルゴリズムの多くを占めていることが分かります。

*20 qTESLA（https://qtesla.org/）とNewHope（https://www.imperialviolet.org/2018/04/11/pqconftls.html）はRound3に組み込まれなかった。一方でSIDH（https://blog.cloudflare.com/introducing-circl/）がRound3に残留している。

2.7 量子計算機に伴う共通鍵暗号方式とハッシュ関数への影響

耐量子計算機暗号は、NISTのコンペティションで対象となるアルゴリズムを見る限り、公開鍵暗号方式のみに注力されているように見受けられますが、実際の利用場面を考えるとそればかりではありません。例えば、共通鍵暗号方式による暗号化と共に公開鍵暗号方式による暗号化が行われるなど、ハイブリッドに利用されています。デジタル署名と呼ばれる技術においても、暗号学的ハッシュ関数と公開鍵暗号方式による署名の2つのアルゴリズムが併用されています。このようなハイブリッドな利用方法においては、2つのアルゴリズムのバランスが重要で、それぞれのアルゴリズムに対してnビット安全性を有しているかどうか考える必要があります。そのため量子計算機の登場による共通鍵暗号方式と暗号学的ハッシュ関数の影響も考慮する必要があります。Groverのアルゴリズムにより、nビットの鍵を持つ共通鍵暗号方式ではその半分のビット長の強度しか持たないことが知られています。具体的にいえば、256ビットセキュリティを持つ暗号アルゴリズムを利用することにより、いずれ量子計算機が登場しても128ビット長の鍵を用いる古典的な暗号アルゴリズムと同じ強度を持つことになります[*21]。

次にハッシュ関数についてはどう考えるべきでしょうか。暗号学的ハッシュ関数には2つの暗号学的な機能を持つことが求められています。1つは耐衝突性(コリジョンレジスタンス)を持つこと、もう1つが原像計算困難性を持つことです。古典計算機においてはnビット長の出力を持つハッシュ関数は、耐衝突性としてn/2ビット安全性、原像計算困難性としてnビット安全性を持つことが知られています。後者についてはそのままGroverのアルゴリズムが最適なアルゴリズムであり、nビット出力のハッシュ関数における現像攻撃に必要な計算量は$2^{n/2}$まで落ちることが知られています。

量子計算機を用いた衝突探索に必要な計算量はBHTと呼ばれる効率的なアルゴリズムを用いると$2^{n/3}$になりますが、この攻撃には$2^{n/3}$というサイズの非常に多くの量子メモリの利用が前提となるため現実的ではありません[*22]。

CRYPTREC Report 2019(CRYPTRECにおいて2019年度の成果を集約したドキュメント群)[*23]によると、細山田氏の解説においてCNSのアルゴリズム[*24]が最も現実的に影響を及ぼすと考えられるという記載があり、CNSのアルゴリズムは$2^{2n/5}$の計算量で衝突発見が行われると報告されています。これは例えばSHA-256が古典的計算機では(耐衝突性として)128ビット安全性を持つアルゴリズムですが、このときCNSアルゴリズムで攻撃したとしても2^{100}以上の計算量が必要となるため現実的な脅威となるとは考えづらいという結論で締めくくられています。このように、共通鍵暗号方式及びハッシュ関数においては、量子計算機の登場による影響は公開鍵暗号方式に比べれば少ないと考えられています。既に策定・普及しているアルゴリズムを用いることで対策できるという意味では、現時点では備えるべきことは少ないといえます。

*21 Internet Infrastructure Review(IIR)Vol.31, 1.4.3 耐量子暗号の動向(https://www.iij.ad.jp/dev/report/iir/031/01_04.html)。
*22 Gilles Brassard et al., Quantum cryptanalysis of hash and claw-free functions. SIGACT News, 28(2):14–19, 1997.
*23 細山田 光倫, 量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価, CRYPTREC EX-2901-2019, 2020年1月. (https://www.cryptrec.go.jp/exreport/cryptrec-ex-2901-2019.pdf)。
*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による量子計算機の脅威に関する考え方

2020年2月にCRYPTREC暗号技術評価委員会から注意喚起に関する文書が発行されました[*25]。これは当時ネイチャー誌に掲載された、量子コンピュータが量子超越を実現したと主張する論文[*26]に対し、技術的側面からCRYPTRECの見解を述べたものです。文書によれば、発表された論文により現在広く利用されている公開鍵暗号方式の安全性が大きく低下することが懸念されるようになったが、近い将来にCRYPTREC暗号リスト記載の暗号アルゴリズムが危殆化する可能性は低いと報告されています。その根拠として、取り上げられた論文では量子誤りが一切ない理想的な環境下を想定しており、2048ビットRSA暗号が8時間で解けると主張している論文[*27]においても2000万量子ビットが必要であるという見積もりもあり、現在の量子計算機の実装進捗とは大きく乖離している点が挙げられています。

この注意喚起レポートは、暗号アルゴリズムの脆弱性情報を検知した際の情報発信フローの情報分類B「正確で信頼性の高い情報を発信することによる過剰反応防止」を目的に発出されました。この背景や情報発信フローに関する詳細については前述のCRYPTREC Report 2019の第1章にて紹介されています。

2.9 NISTとの個人的な対話を通して

EUROCRYPT2016の併設ワークショップ[*28]で、NISTのLily Chenと耐量子計算機暗号を含む暗号政策に関して議論する機会に恵まれました。その際にNISTが暗号政策に関して耐量子計算機暗号と軽量暗号という2つの相異なる方向性を持っていることを指摘しています。当時の私は共通鍵暗号方式のポストクォンタム対応は鍵長を倍にするなどの延命技術による対策しか考えておらず、例えばAES-512などの新しいアルゴリズムを開発する、またはTriple AES（TripleDESのように3つの鍵で繋ぐ方法）などの対応を行うといった対策を、今後検討するのか尋ねてみました。その際の回答としては、既にAES-256があり、当時でも安全に利用可能であるし2030年以降も利用可能な鍵長は128ビットセキュリティを持つ暗号方式であるので、AES-256でも十分に量子暗号計算機耐性を持つと認識しているとのことでした。

また、共通鍵暗号方式での対策において公開鍵暗号方式と同じように新しいバックグラウンドに基づく暗号方式の導入という考え方が私には想像できませんでしたが、実際FSE2020では耐量子計算機対応共通鍵暗号方式「Saturnin」に関する発表が行われました[*29]。Saturninは軽量暗号でもあり耐量子暗号でもあるという2つの側面を持つ暗号方式です。軽量暗号はNIST

*25 CRYPTREC 暗号技術評価委員会, 現在の量子コンピュータによる暗号技術の安全性への影響, 2020年2月17日, CYPTREC ER-0001-2019.（https://www.cryptrec.go.jp/topics/cryptrec-er-0001-2019.html）。

*26 Frank Arute et al., Quantum supremacy using a programmable superconducting processor, Nature volume 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/ViennaMay13-2016.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）。FSE2020（https://fse.iacr.org/2020/program.php）にて発表。Saturninプロジェクト（https://project.inria.fr/saturnin/）。

で標準化のためのコンペティションが同様に開催されており、IoT機器での利用など非力なデバイスを想定した暗号アルゴリズムの総称として知られています。鍵長としては80ビット程度が想定されており安全性は一般的に用いられるアルゴリズムよりも弱い方式が採用されています。CRYPTRECでも2013年度から2016年度にかけて軽量暗号WGが設置され、軽量暗号の適切な利用を支援することを目的としてCRYPTREC暗号技術ガイドライン(軽量暗号)が2017年6月に発行されています[*30]。軽量暗号においても格子暗号などで見られるように独自にコンペティションを開催することで、自らの方式が安全かを検証する試みがなされており、今後の進展が個人的にも大変楽しみな研究分野の1つです。

このように、様々な利用場面において暗号アルゴリズムの標準化が行われています。既に述べたように量子計算機の登場が即座に影響を及ぼすことはありませんが、最新動向についてはCRYPTRECのサイトでキャッチアップできますので、活用していただければと思います。

執筆者:
須賀 祐治(すが ゆうじ)
IIJ セキュリティ本部 セキュリティ情報統括室 シニアエンジニア。2008年7月より現職。暗号と情報セキュリティ全般に関わる調査・研究活動に従事。CRYPTREC暗号技術活用委員会 委員。暗号プロトコル評価技術コンソーシアム 幹事。情報処理学会 CSEC研究会 幹事。IWSEC2021 General co-chair。AsiaCCS'22 General co-chair。Cryptoassets Governance Task Force(CGTF) Security WG member。APSIPA Multimedia Security and Forensics Technical Committee member。BGIN(Blockchain Governance Initiative Network) co-initial contributor。

*30 CRYPTREC 軽量暗号WG, CRYPTREC 暗号技術ガイドライン(軽量暗号)(https://www.cryptrec.go.jp/report/cryptrec-gl-2003-2016jp.pdf)。
CRYPTRECシンポジウム2017 軽量暗号ガイドライン紹介(https://www.cryptrec.go.jp/symposium/20171218_cryptrec-lw.pdf)。

クエリサービス
～柔軟なマネージドデータベースサービスへの挑戦

3.1 はじめに

IIJでは、2019年度よりエンジニアが日頃考えている技術や新サービスのアイデアを具現化する機会として「テックチャレンジ」という制度が始まりました。この度、第1回テックチャレンジに筆者の企画が採択され、去る2020年9月末日までの1年間にわたり、サービスの仕様検討・設計、サービスのプロトタイプ開発を１人で行ってきました。

採択された開発テーマは「クエリサービスの開発」です。昨今、アプリケーションがKubernetes上のコンテナへデプロイされることが当たり前の時代となっているものの、データベースに関してはデータの永続性、可用性、パフォーマンスの面でKubernetesのコンテナ利用にまだ最適解といえる手法がないのが実状です。そこで、コンテナを利用した際と同等の柔軟性に加えデータの永続性、可用性を持つマネージドデータベースサービスであるクエリサービスを、Kubernetesに隣接する外部サービスとして開発することで、これらの課題を解決しようというのが今回の狙いです。

3.2 重点開発した機能

テックチャレンジとして「クエリサービスの開発」という大きな開発テーマが設定され、解決するべき課題は見えていましたが、具体的に開発する機能は着任後に改めて考える必要がありました。機能の開発に当たっては、IIJのサービスを開発・運用している管理者の方々からデータベースにまつわる課題をヒアリングしました。話を聞くと、それらは筆者が開発担当者であった当時抱えていた課題と重なるものがあり、大変共感を覚えました。開発者・運用者の視点で「こんな機能があれば助かる」という機能を複数挙げ、要件の整理を行い、クエリサービスの設計・開発へ進みました。

データベースはサービスを開発・運用する上で必須の構成要素であるものの、サービスのアプリケーション開発や運用が主務の部門が少人数のチームでデータベースの構築や運用を担当するのは重荷となります。サービス開発・運用者にとって使いたいのは、クライアントから接続してデータを格納し、CRUD操作ができる「データベースの機能」であって「データベースサーバ」ではないのです。更に言えば、その非機能要件になると「あってほしいけれど関わりたくない」という存在になっています。従って、データベースを使い始めるまでのデプロイ、データベースの高可用性構成やバックアップ、セキュリティ構成、パフォーマンスを考慮したインスタンスの設計は利用者からは完全に隠蔽されるサービスであることが必要だと考えました。ただ、これはデータベースのサービスとしては言わば当たり前の機能であり、クエリサービスならではという新規性は感じませんでした。

一方、IIJのサービスを開発・運用している管理者の話を聞くうちに筆者も大きく共感したことがありました。それは、サービスのアプリケーションの実行速度が遅いときは「力業」で何とか乗り切りたいということです。開発者にとってデータベースのパフォーマンスが上がらないときはSQLのチューニングや索引を追加するなどの手法が正攻法ですが、アプリケーションが急にスローダウンしたときには、まずはアプリケーションを止めずにデータベースのパワーが上がる「魔法」のような機能があればとても助かるというのです。筆者自身が開発担当だった頃でもあれば助かる機能でしたので、クエリサービスにはこの魔法のような機能を実装しようと考えました。これはKubernetesのデータベースOperatorでも提供されていない機能で、技術的にチャレンジし甲斐があるというのもモチベーションとなりました。

課題としてよく話に挙がってきたことがもう1つあります。「データはシステムの寿命より長いのでデータベースは長く使いたいが、システムのリプレースの度にデータ移行を行わなければならない。更に、データベース自体もバージョンアップしないといけないのに詳しい人が部門にいないので塩漬けされた状態で使い続けている」という課題でした。筆者もシステムインテグレーション部門でお客様のシステムのリプレースに伴うデータベースの移行やバージョンアップを長年にわたり多くの案件で技術支援し、その対応に疲弊するプロジェクトの現場を経験してきました。

クエリサービスでは同じデータベースを使い続けられるのはもちろんのこと、そのデータベースが稼働するハードウェアもソフトウェアも常にアップグレードしていくものを目指すべきと考えましたが、これらが利用者の負担にならないことが非常に重要だと理解しました。このようなことを目指しているのがKubernetesのローリングアップグレードであり、似た機能になると思うのですが、ここはクエリサービス完全オリジナルの実装を目指して開発を行いました。

1年間という時間はあれど、現実的な話として実際に開発を行うのは筆者1人です。いろいろ手を出しても中途半端になる可能性はありました。しかし、やはり現代のデータベースサービスにおいて最低ラインとなる高可用性構成やバックアップといった非機能要件の機能開発はもちろん、クエリサービスの独自性として身近なエンジニアや自分自身が抱いていた課題解決、また当初のコンセプトのとおりKubernetes上のコンテナのデータベース同等またそれ以上の機能を提供するべきと考えました。そこで、以下を重点開発項目に掲げて開発を行いました。

① データベースを止めることなくユーザが自由にデータベースのパフォーマンスをコントロールできる機能を開発
② ①の機能をアプリケーションから簡単かつ自由に使えるインタフェースの開発
③ ①の機能を使いやすくする柔軟な課金機能の開発
④ ①の機能をユーザに代わりシステムが自律制御し、常に最適なパフォーマンスを提供する機能
⑤ クエリサービスを継続利用してもらうためサービスをアップデートする機能

開発した機能群を概念図で示したのが図-1です。なお、今回はコアとなるデータベースにはOracle Databaseを採用しています(他のデータベースへの対応も検討中です)。

次項から、①②については「オンラインリソース機能」、③は「秒課金機能」、④は「オートスケーリング機能」、⑤は「サービス更新機能」と分けて、クエリサービスのコア機能を紹介します。

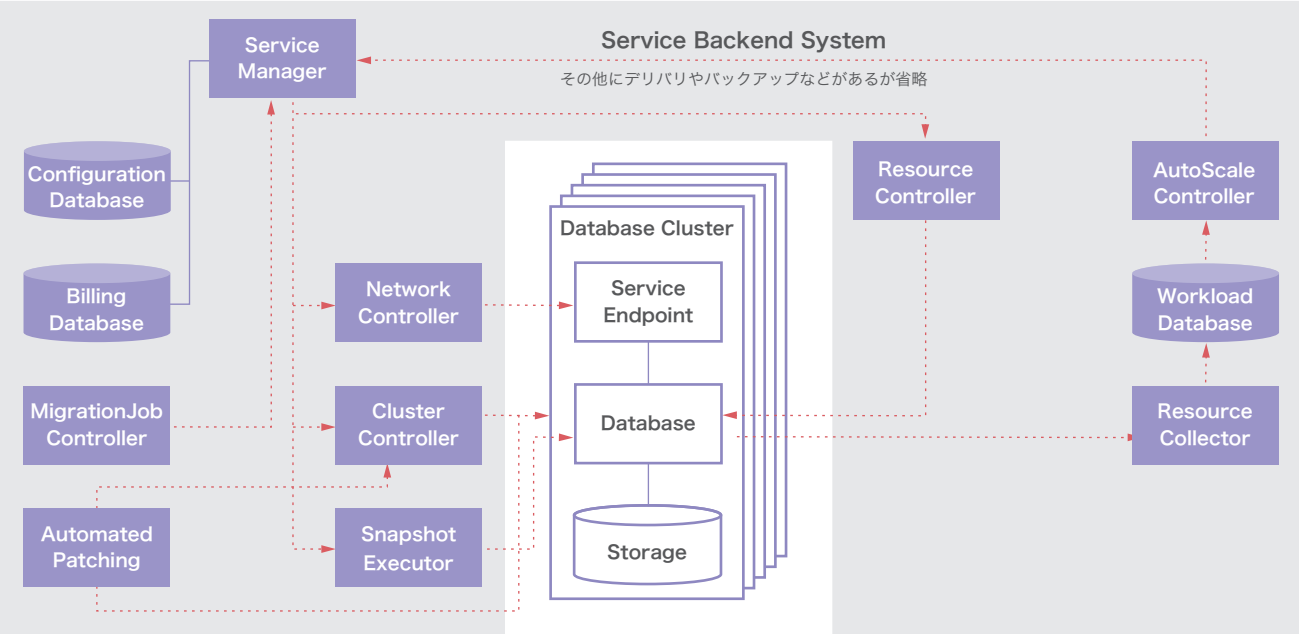

図-1 開発したオーケストレーションシステム

3.2.1 オンラインリソース変更機能

重点開発項目の1つ目に、データベースを止めることなくユーザが自由にデータベースのパフォーマンスをコントロールできる機能として「オンラインリソース機能」を開発しました。

データベースは、Web/アプリケーションサーバのように負荷の上昇に併せた分散構成を容易に構成することが難しいとされており、スケーラビリティ構成としてレプリケーションやシャーディング、または整合性を犠牲にした構成が用いられることが多くあります。しかしながら、それらは読み込み処理に限定された効果である、またはトランザクション処理に制限が生じるなどの制約が付きまといます。特にデータベース特有の大きな1つの処理にパフォーマンスの問題が生じた場合は負荷分散でのスケーラビリティでは効果が得られないと言えます。

また、データベースのパフォーマンスの特性としてマルチワークロードと言われており、同じデータベースに対しても使い方により必要となるリソースが大きく変わってくることも特徴として挙げられます。データベースの処理は主に①SQL構文②処理対象のデータ量③処理の同時実行数の組み合わせにより大きく異なります。

これら要素の組み合わせは1日の中でも大きく変化します(表-1)。例えば、日中帯は多くのユーザにより同時に利用されるのですが、1つの処理が小さい。一方で夜間は日次ジョブが実行され、同時に実行される数は少ないが、1つの処理が大きいことが珍しくありません。これに季節変動や突発的なイベントの発生などが重なると更にワークロードが複雑化していきます。

	日中	夜間
SQL構造	単純	複雑
1処理あたりのデータ量	小	大
同時実行数	多	少
求められる性能特性	レスポンス	スループット
より重視するリソース	CPU/メモリ	I/O

表-1 データベースの処理特性

従来のオンプレミスと呼ばれるコンピューティング環境では、異なる特性の論理積を弾力性のないリソースで実現するため、データベースには常にシステムの中で最も大きなリソースを用意してきました。

この10年でコンピューティングリソースはクラウドサービスを利用することが当たり前の時代になりました。かつてクラウド上でデータベースを動かすことに懐疑的であった時代が嘘のような話となり、当たり前のようにクラウドサービス上でデータベースが動いています(今日ではKubernetes上でデータベースを動かす取り組みが似ているように見えます)。

クラウドサービスが提供する多種多様なコンピューティングリソースにより、ユーザのリソース選択肢は大幅に拡大していますが、そんなクラウドサービス上のデータベースにも課題はあります。例えば、IaaS上にデータベースを構築・運用している場合は、仮想マシンが起動するロケーションがクラウドに変わっただけでオンプレミスとの運用における根本的な課題は解決されていないと言えます。また、クラウドベンダーがPaaSとして提供するデータベースのサービスの多くは、特にCPUのリソースを変更する際に再起動が必要となります。データベースの停止はサービス全体の停止となるため、気軽にリソースの変更が実行できない状況です。

昨今、注目度が急上昇のkubernetesとそのOperatorによるデータベースのスケールアウトも有効ではあります(レプリケーションを使用する場合、負荷分散は実行数が多い場合にはシステム全体の処理量を上げることが有効な手段であると思いますが)。しかし、前述のとおり1つの処理が遅い場合にはスケールアウトよりスケールアップの方が有効な手法であると言えます(NewSQLという選択肢もありますが、まだまだ一般的であるとは言えないと個人的には思っています)。「そんな遅いSQLを書いていてはダメだ」と指摘したくなるかもしれませんが、実際によく発生する事象ではあるという点は同業者として共感いただけるのではと思います。なお、スケールアウトを批判しているわけではなく、寧ろ筆者はスケールアウトが好きで、本稿では触れませんが、クエリサービスではスケールアウト“にも”対応しています。

話題が少し逸れてしまいましたが、クエリサービスではデータベースを止めることなく処理性能を上げることに取り組み、コンピューティングリソースを抽象化した“オンラインリソース変更機能”を開発し、以下の機能を実現しています。

① データベースのパフォーマンスに影響を与えるCPUとI/Oのデータスループットを、データベースを止めることなく必要なときに必要なだけ利用可能
② リソースをリクエストする発生源であるアプリケーションから簡単にリソースの変更ができるようSQLでのインタフェースを提供

①はクエリサービスの仕様として表現すると表-2のようになります。

CPUコアとI/Oのデータスループットのリソースはそれぞれ個別にリソースを基本スペックから最大スペックまで増加減させることができます。リソースの変更に要する時間は数秒以内で後述しますが課金対象も即時反映されます。

②ですが、オンラインリソース機能はAPIの他にSQLから実行できることが特徴の１つとなります。オンラインリソース機能は手動で制御する場合も分かりやすく、何といってもリソースをリクエストする発生源であるアプリケーションから簡単にリソースの変更ができるように開発されていますので、プログラム内に埋め込むことが容易にできます。例えば、大きな処理を実行する前後にCPU数を変更するなど、従来はできなかったリソースの使い方が可能となり、開発者なら簡単に実装することができます(図-2)。

	固定	可変	
リソース	基本	オプション	
		最大スペック	追加単位
データベース	1	ー	ー
CPUスコア	1	6	1
データスループット(MB/s)	100	2000	100
同時接続数	50	300	CPUスコア数連動
データ領域(GB)	50	8000	自動拡張

表-2 クエリサービスのスペック

```
        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;
/
```

CPU数を6へ変更
CPU数を2へ変更
並列処理による性能向上

図-2 CPUの並列処理

次にオンラインリソース機能の動きを解説します。有償のサービス提供を想定している以上、ユーザが直接CPUコア数やI/Oスループット性能を変更させてしまってはサービスとして破綻してしまいます。そのためユーザには通常のSQLのプロシージャとしてインタフェースを提供しているものの、システム変更のSQLコマンドがラップされた単純なプログラムを実行して変更しているわけではなく、外部プログラムを経由しバックエンドシステムのService Managerへリソース変更依頼を投げる方式を採用しています(図-3)。

従って、ユーザがSQLやAPIで実行に用いられるプロシージャはユーザリクエストを受け付け、設定可能な値であることをユーザデータベース上で判断させ、外部プログラムへ渡すだけの簡素なプログラムとなっています。

外部プログラム経由でリクエストを受け取ったサービスマネージャは、バックエンドシステムの構成管理データベースの対象ユーザデータベースの情報を変更し、課金システムで対象ユーザデータベースに関する課金条件を変更した後に、リソースコントローラを経由して、データベースのリソースマネージャを実行することで対象ユーザデータベースのCPUコア数やI/Oのスループットの設定を変更します。これらの処理が完了した後に、ユーザデータベース上のプロシージャを介して設定が完了したことを接続中のユーザセッションへ戻しています(表-3)。

詳細なプログラムの実装を説明するとページが全く足りないのですが、これらの処理は数秒以内に完了するため、ユーザはほぼリアルタイムでリソースの増加減を実行ができるよう

	日中	課金単位	使用料
基本	IIJクエリサービス基本	1	月額固定
オプション	CPUコア追加分	1コア	
	データスループット追加分	100MB/s	秒課金
	データ容量51GB以上の拡張分	1GB	

表-3 クエリサービスの課金

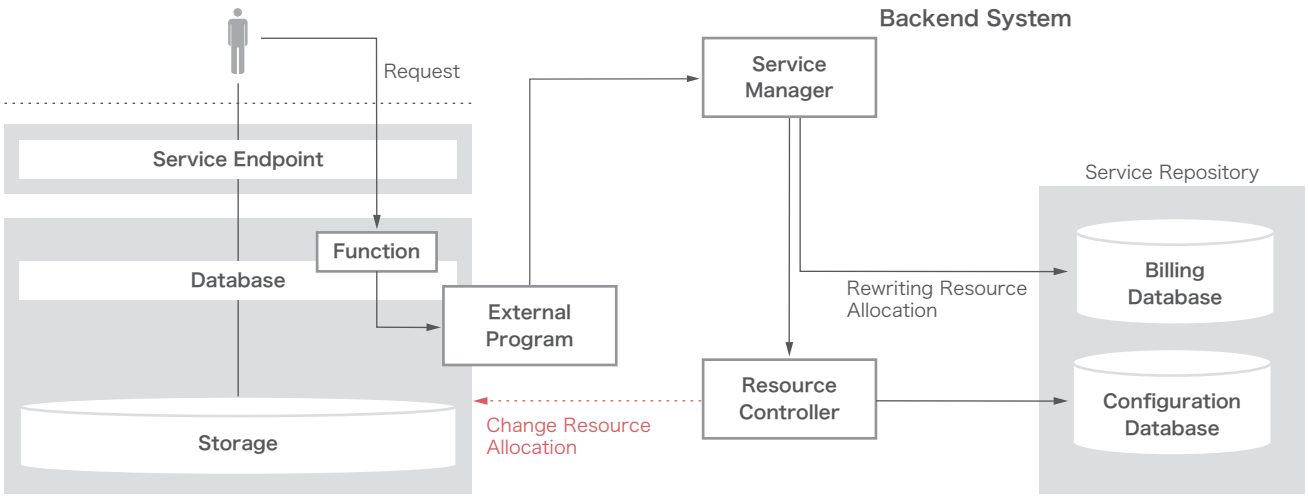

図-3 リソース変更処理の裏側

になります。これは極端な例かもしれませんが、オンラインリソース変更機能をプログラム内に組み込むことで、アプリケーションが動的に処理対象のデータ量を計測することによってデータ量に合わせたリソースを処理の都度、動的に変更するといった全く新しいデータベースサービスの使い方が可能になります(図-4)。

3.2.2 秒課金機能

オンラインリソース機能により好きなタイミングでリソースの拡張・縮小が可能になりましたが、リソースの利用料金が硬直的であっては使い勝手が悪いと言えます。クエリサービスでは柔軟な課金体系を目指し、基本スペックを超えるCPUコアやデータスループットなどのリソースについては秒課金とする機能も併せて開発しています。

図-5はクエリサービスで実装されている使用料金のイメージです。クエリサービスでは基本スペックを超えるCPUコアやデータスループット、データ領域がそれぞれ独立して秒課金されます。そのため必要になったときに使ったリソースに対し「使った分だけ課金」が可能になっています。

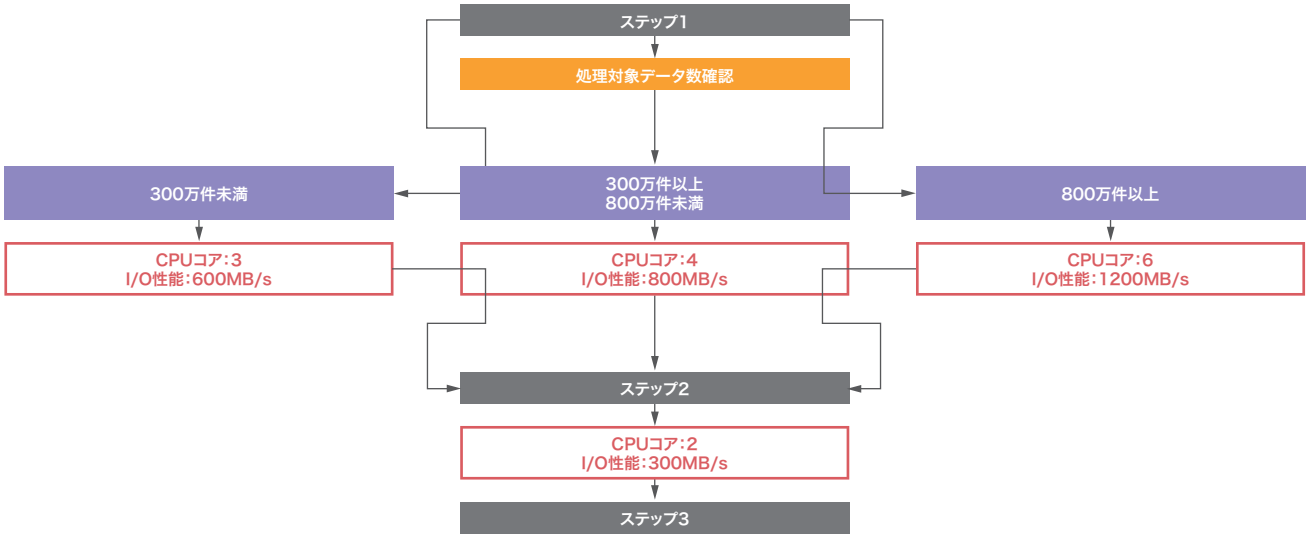

図-4 プログラマブルなリソース制御

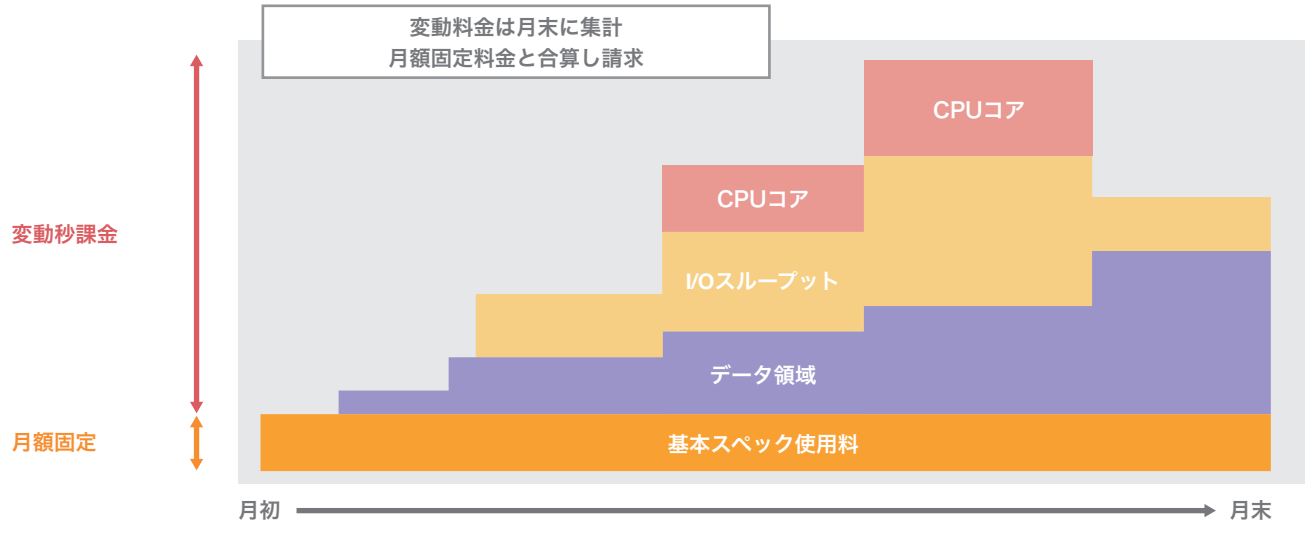

図-5 本サービスにおける課金の考え方

データベースの並列処理を組み合わせることでリソースの増強と処理速度のバランスが取れるようになれば、1秒当たりの単価が上がっても総利用料金へのインパクトは小さく抑えることができると言えるのです(図-6)。より柔軟な課金機能があってこそオンラインリソース機能が活用されると考えています。

3.2.3 オートスケーリング機能

オートスケーリング機能は、オンラインリソース変更機能と連動し、ユーザに代わってシステムが自動制御し、常に最適なパフォーマンスを提供する機能です。オンラインリソース変更機能は便利な機能ではありますが、その実行に際し人が都度判

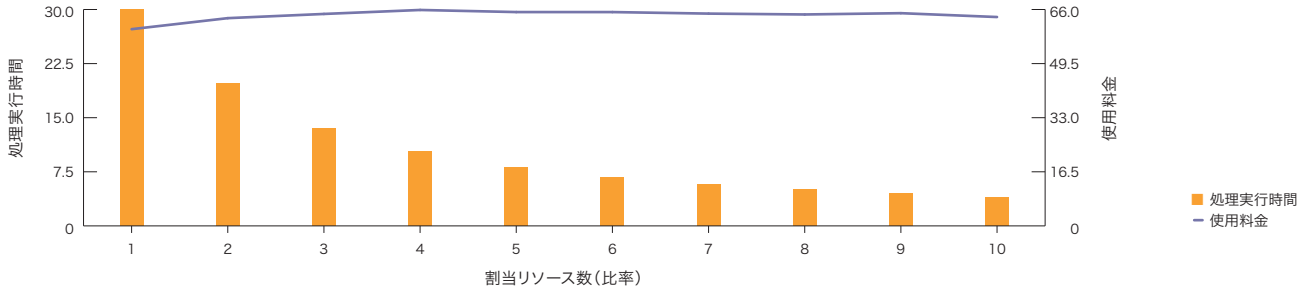

図-6 オンラインリソース変更機能と秒課金の考え方

リソース	最小値	最大値	増減値	拡張条件	縮小条件	評価間隔
CPU	1コア	6コア	1コア	割当済み全CPUコア 直近5分間の平均使用率を評価		1分間隔
				70%以上	65%以下	
I/Oスループット	100MB/s	2000MB/s	100MB/s	割当済みI/Oスループット 直近5分間の平均使用率を評価		1分間隔
				80%以上	75%以下	

表-4 自動スケーリング機能

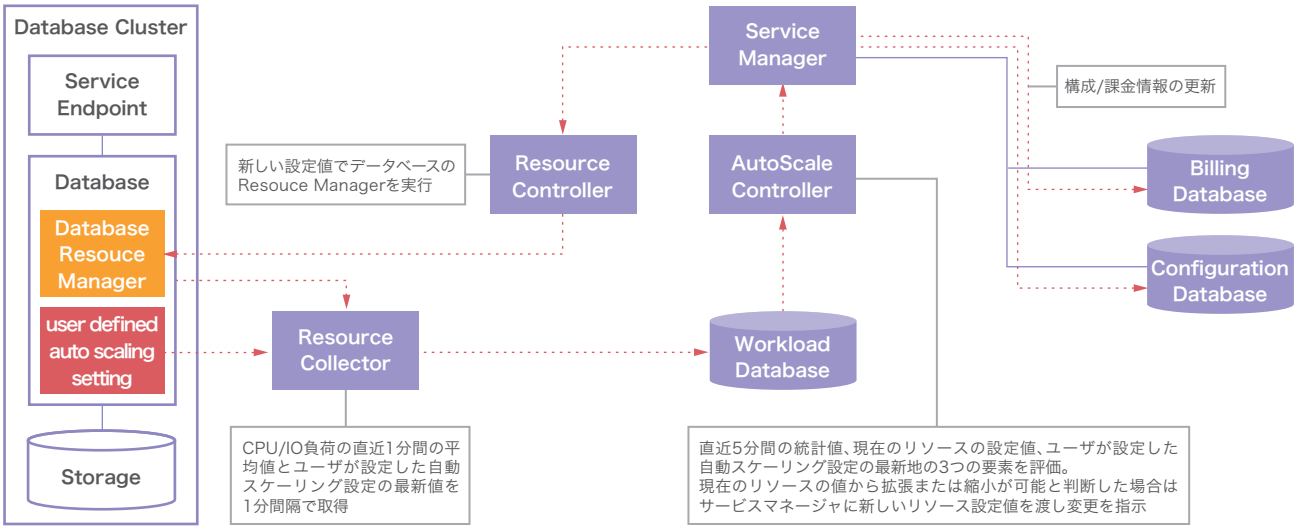

図-7 自動スケーリング機能の内部動作

断するような手動実行や外部の監視システムとの連携などを組み合わせた複雑な構成が必要になってしまうと、オンラインである利点や機能としての魅力が半減すると考えられます。クエリサービスの便利な機能をシステム運用の自動化でより活用してもらうために、オンラインリソース変更機能を自動制御するオートスケーリング機能も開発しました(表-4)。

オートスケーリング機能の内部動作は図-7のようになります。オートスケーリング機能はデータベースの稼働状況を収集し、負荷状況に応じたCPUコアとI/Oデータスループットといったデータベースの性能に関わるリソースを、オンラインリソース変更機能を介して自動で割り当てます。各リソースの負荷状況が拡張及び縮小の条件内で保つことを目標にして常時動作しています。プログラムとしてはこれら一連のジョブを制御するループ処理が実装されています。

オートスケーリング機能は非常に便利な機能でユースケースはいろいろありますが、こんなときにとても有効な仕組みだと思える例が“性能障害”が発生した際の対応ではないでしょうか。システムを運用していると、データベースの処理が突如スローダウンする現象に遭遇することが年に数回あります。その原因は様々で、よく言われる「何もしてないのに急に遅くなった」(多くの場合はデータベースの実行計画が突然変わった)というものもあれば、新しいプログラムがリリースされた直後や、データが大量に増えた、セールなどのイベントを実施した等々、いつもと何か違うことを契機に発生することもあります。このような“性能障害”といえる事象が発生すると、その対応は“多くの矛盾”との戦いとなっていきます。

まず、アプリケーションのスローダウンはインフラ視点でのリソース監視では検知しづらい場合があり、運用部門での検知が遅れることがあります。ユーザからのクレームが契機となる場合が多く、最初からハードルの高い対応となってしまいます。一方で障害の原因となっている実行中の処理は、プロセスレベルで止められる場合もあれば、夜間バッチのように翌日の営業開始時間まで完了しないと業務インパクトが大きいので止められないというものがあります。止められないがすぐに対応しなければならないという難しい局面です。これらのスローダウンの根本原因は、データの急増や不適切な実行計画に基づきクエリが実行されている、索引がない項目に対して検索が実行されてしまったなど、アプリケーション側に起因する原因がほとんどなのですが、仮に特定されたとしても、止めずに対応することは不可能または高難易度なオペレーションとなります。しかも何故かこのような障害が発生するのは人がいない休日か深夜の場合が多いのが厄介なところです。

上記のような矛盾に満ちた性能障害を、根本解決ではないものの、サービスが解決してくれたら、ユーザも運用者も開発者も幸せになれるだろうという思いからオートスケーリング機能を開発しています。

■ オートスケーリング機能の効果測定

オートスケーリング機能の効果を確認するために性能障害を起こすのは難しいのですが、代わりに使用中のスペックでは捌くことができないような大きな処理がデータベースに実行されるとクエリサービスのオートスケーリング機能がどのような動きを見せるのか、実際に処理を実行して計測してみました。

今回の試験では、1億件超のデータが格納された受注テーブルと商品マスタ、顧客マスタなど複数のテーブルを結合してクエリを実行しています。なお、今回はオートスケーリングの効果を分かりやすくする検証であるため、クエリの実行が完了するたびに共有メモリのバッファ領域をクリアしています。

クエリが実行されると受注テーブル内の全レコードがシーケンシャルにアクセスされます。シーケンシャルアクセスかつ、共有メモリは都度フラッシュしているクエリが実行されると、ストレージに配置されているブロックが大量に読み出されます。そのためクエリ実行に多くのI/Oリソースが要求されます。

初期値となっているクエリサービスの最小構成となる基本スペックのI/Oスループット性能は100MB/sと小さいため、億を

超えるレコード件数へのシーケンシャルスキャンが実行されると、開始直後からI/Oスループットリソースの使用率は急上昇します(図-8)。オートスケーリング機能は常時データベースの稼働統計情報を収集して診断を繰り返し、リソースの拡張または縮小を指示します。なお今回の検証で流したクエリはI/Oバウンドな処理であるため、オートスケーリング機能はI/Oスループット性能のみ900MB/sまで拡張を続けます。

面白いのは、I/Oスループットが900MB/sになるとCPUがI/Oの待機する時間を縮小するため、処理の特性がCPUバウンドへ変化を見せることです。そうなると今度はCPUコア数を増やして並列度を上げようと試みますが、CPUの使用率がそこまで上がるクエリではないので、オートスケーリング機能がCPUコアを再度手放す動きを見せました。いわゆる「オンラインスケールアップ」というべき動きになるのですが、Kubernetes上のコンテナではこの実装が今のところまだ安定版には至っていないようです。見ていると面白いのですが、CPUコアのキャッチ&リリースするような動きは性能にブレを生じさせる原因でもあるので改良すべき余地があります。もっとも、それほど大きな問題とは考えていません。このサンプルデータベースではオートスケーリング機能によりCPUコアは最大6コア、I/Oスループットは最大1000MB/sまで拡張可能なのですが、CPUコア数は1または2コア、I/Oスループットは900MB/sで十分と判断しています。実際、1000MB/s以

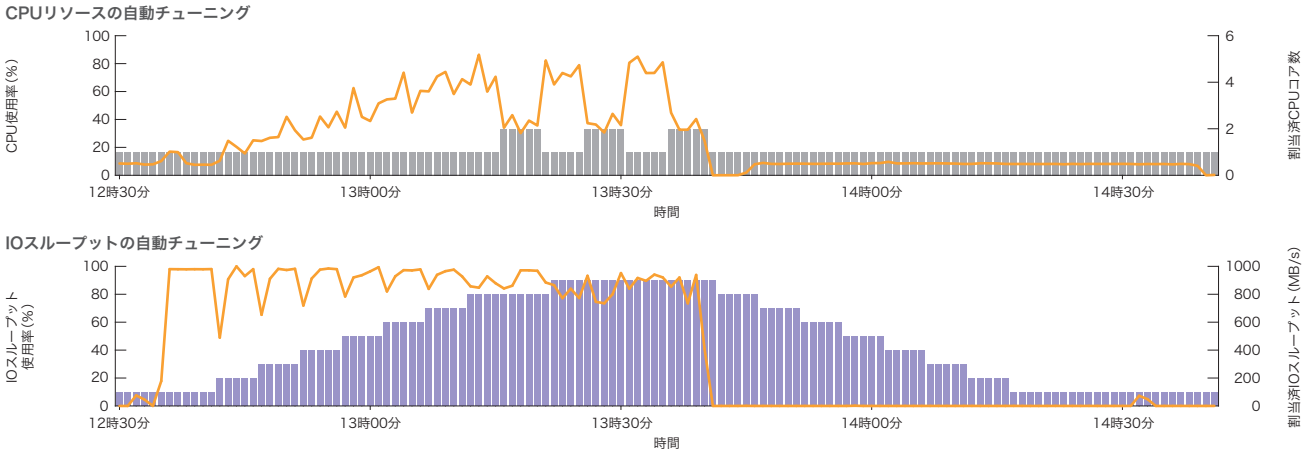

図-8 自動スケーリング機能により制御されるリソースの遷移例

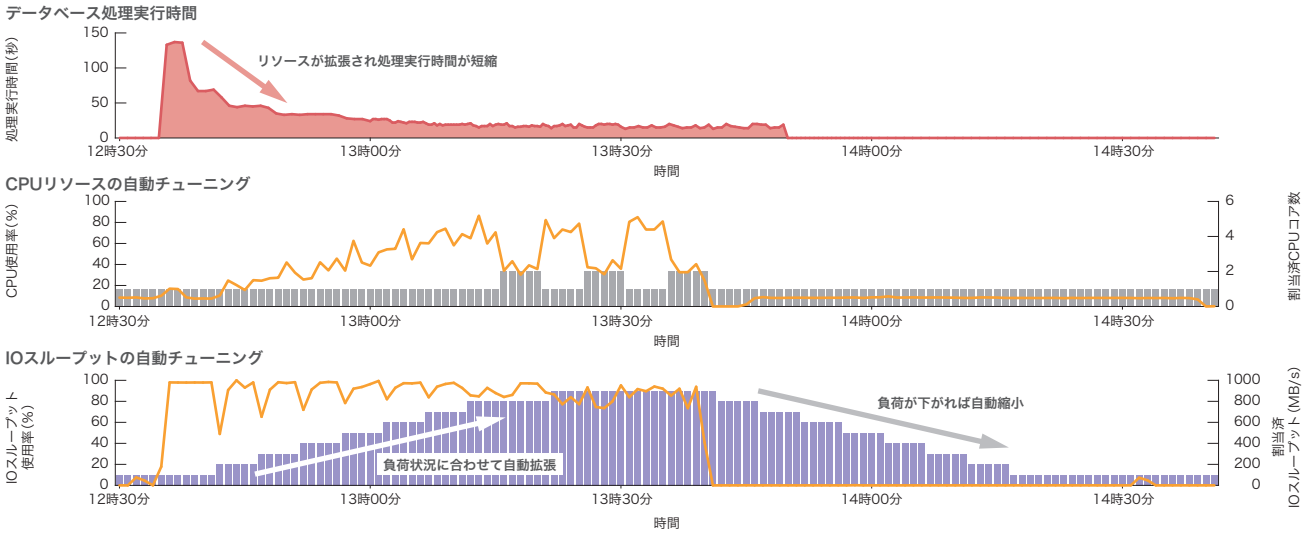

図-9 自動スケーリングによるデータベース処理時間の変化例

上やCPUコア数の設定を増やしても確かに性能は上がるのですが、目を見張るような大きな変化は見られませんでした。

実際の処理性能に対するインパクトは図-9のとおりです。初期のI/Oスループット性能の拡張の効果が非常に大きいのですが、処理時間は確実に短縮されていきます。

ところで、オートスケーリング機能が有効の状態でユーザが手動によりI/Oスループット性能の値を2000MB/sに設定した場合はどうなるのでしょうか。設定値自体は手動設定が優先されるためユーザのデータベースのI/Oスループット性能は2000MB/sに設定されます。オートスケーリングが無効の場合は2000MB/sが維持されますが、有効の場合はシステムが稼働状況を判断し、やはり900MB/sを目指してリソースが縮小していきます。

オートスケーリング機能で毎回100MB/sのような最小構成まで縮小してから拡張していては期待する性能を得るまでに時間がかかって困るという場合は、オートスケーリングの対象範囲をユーザが自由に設定できるようになっています。例えばCPUコアは2-4コア、I/Oスループット性能は500MB/sから1500MB/sの間で自動調整とすれば500MB/sを下回ることはありません。また、CPUコアはオートスケーリングさせずI/Oスループットだけという設定も可能です。設定の変更はオンラインででき、1分後(バックエンドシステムの都合)にシステムに反映されます。

■ オートスケーリング機能の課題

リソースの適正利用を図るため、リソースの拡張・縮小を判断する間隔や閾値、増加減値はバックエンドシステムの管理下でユーザが変更することはできないのですが、これもユーザが変更できる実装に変えていこうと考えています。

また、便利そうに見えるオートスケーリング機能ですが、まだまだ課題は多く存在しています。特にリソース割り当ての精度に関しては、開発している立場から見てもまだまだ難ありと言えます。

スケーリングの動作の根拠は前述のとおりデータベースの稼働統計値となるのですが、ここで得られる数値は過去に発生した事象に基づくものであり、オートスケーリング機能のコア機能であるオートスケールコントローラは「この傾向が今後もしばらく継続するであろう」という極めて単純な思考に基づき動作しているため、未来の負荷状況を先回り予測して動作ができるような格好の良いものではありません。また、この単純明快な思考は時系列における変化点の発生直後に期待値から外れることが多いのも問題です。具体的に言うと、負荷が下がっているにもかかわらずリソースを拡張することがあります。MLなどで更に賢いオートスケーリング機能を目指したいと考えています。

課題が残るオートスケーリング機能ではありますが、突発的に発生する性能障害のような事象に対しては有効な対応策の1つであろうかと思います。何といってもすべてが自動制御されているため、事象検知の遅延やクレームになって発覚するといった間にもデータベースのリソースが確実に拡張し性能を維持しようとします。事象が収まればリソースが縮小するので、場合によっては気がつかないかもしれません(使用料金には反映されますが)。自動化は構成管理やアプリケーションのデプロイだけではなく、システムの性能を自動的に維持するところまでやっていくと、運用はより楽になると思われます。

3.2.4 サービス更新機能

クエリサービスはサーバレスを標榜しているものの、実行環境は通常のデータベースと変わらないためサービス基盤は旧くなります。まずサービス基盤のハードウェアが旧くなり経年劣化に伴う機械的故障の発生率が高まります。OSやデータベースのソフトウェアも、サポート切れやパッチの提供が終わることでセキュリティホールやバグが修正されなくなります。このようにサービス基盤の更改は安定したサービスの継続提供には必要なイベントです。

新しいクエリサービスの基盤を用意することは簡単なのですが、利用中のユーザデータベースを新しい基盤へ移行することが必要です。データベースだけをバージョンアップするだけな

らインプレースアップグレードが最も簡単な方法ですが、バージョンアップに伴う時間が長くサービスの長時間の停止が避けられません。また、OSやハードウェアの更改が不十分、またはその更改を別途実施する必要が生じる可能性が出てくるため、効率の良い方法とは思えません。一方でバックアップから別のインスタンスを作成する方法もありますが、クライアントの接続先変更を伴う場合があり、作業範囲がクエリサービスの範囲内に収まらなくなってしまう可能性が出てきます。

クエリサービスでは、新しいサーバやストレージへの切替、OSのバージョンアップ、データベースソフトウェアのバージョンアップとデータベース自体のアップグレードをすべて1つの処理で完結させます。Kubernetesのローリングアップグレードに近いと思いますが、クエリサービスではレプリケーションのような技術を用いず、より高速かつユーザに透過的にサービス全体を更新する仕組みを実装しています。その特徴は以下のとおりです。

① データ移行が不要
② データの整合性を維持したデータベースの切替やバージョンアップなどすべての処理を完全自動化
③ ユーザが1クリックするだけですべてが完了
④ 最大15分で完了
⑤ 何度でも再実行可能
⑥ 移行後もクエリサービスの接続先は不変

クエリサービスには上記のようなサービスを更新し、継続的にデータベースを利用していただく機能が基盤に備わっています。サービス更新機能では、データ移行を不要としています。不要と言うと語弊がありますが、ユーザは全く意識することなく、利用しているデータベースが新しいサービス基盤へ複製されます。データの整合性もトランザクションログの転送で自動解決されます。

サービス更新機能は大きく3つのフェーズに分かれています。フェーズ1として、まずユーザのデータベースのスナップショットが新サービス基盤へオンラインで作成されます(図-10)。スナップショットの作成はユーザが手動実行する必要はなく、スナップショットが取得可能な状態をモニタリングして自動生成されます。データベースが複数個存在していても処理は個々のデータベースごとに完全に独立して処理されます。

フェーズ2以降はデータベースが新しいサービス基盤へ切り替わるため、処理の開始はユーザにトリガーが移動します。ユーザは任意のタイミングで処理を開始することができます。フェーズ2が実行されると、対象のユーザデータベース用のサービスエンドポイントがクローズされます(図-11)。これにより、切替を切り戻す際の静止点ポイントが確定されます。静止点ポイントが確定したら、新しいサービス基盤上のスナップショットとの最終的な同期処理を実行します。最終のデータベース間の同期処理が完了した後に現行のサービス基盤上の

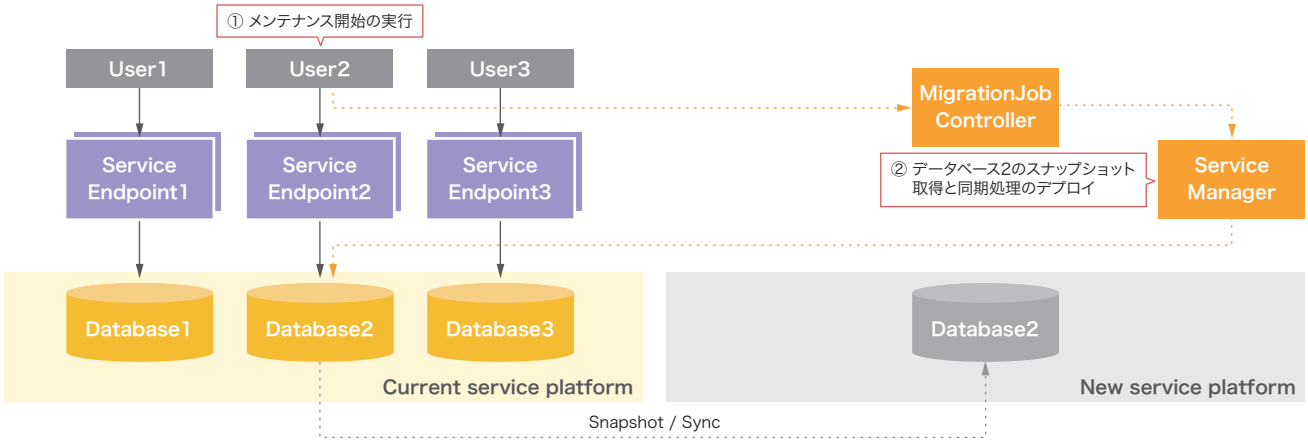

図-10 クエリサービス更新機能の内部動作1

データベースのステータスをinactiveへ変更し、オフライン状態へ遷移させます。オフラインとなってもデータベースの物理的な削除は行いません。これはサービス更新処理の実質的なバックアップという意味合いがありますが、ユーザが切戻しを実行した際にサービス更新処理をスナップショットから復元する時間を要しません。ステータスをactiveへ切り替えるだけで、極めて短時間で切戻しが完了できるからです。

新しいデータベースは現行のデータベースがinactiveになるまで処理を待機します。これは双方のステータスがactiveになることでスプリットブレインが発生する原因を排除するためです。そのため独立して処理は行われません。現行データベースが正常にinactiveとなった後に、新しいデータベースでサービスを継続するための処理が再開されます。データベースのバージョンをアップする場合はこのタイミングで実行します。また、新しいデータベースを高可用性構成へ構成変更します。フェーズ2の最終段階で、ユーザデータベース用サービスエンドポイントに登録されている経路情報を新サービス基盤側へ更新します。

フェーズ3では新しいサービス基盤上のユーザデータベースのステータスがactiveとなった時点でエンドポイントをオープンします(図-12)。フェーズ3での特徴は、サービスエンドポイントの経路情報が変更になっただけで、ユーザ側の接続ポイントが不変であることです。そのためユーザは処理完了の通知後に再接続を実行すれば、何も変更することなく新しいサービス基盤上のユーザデータベースへ接続されます。

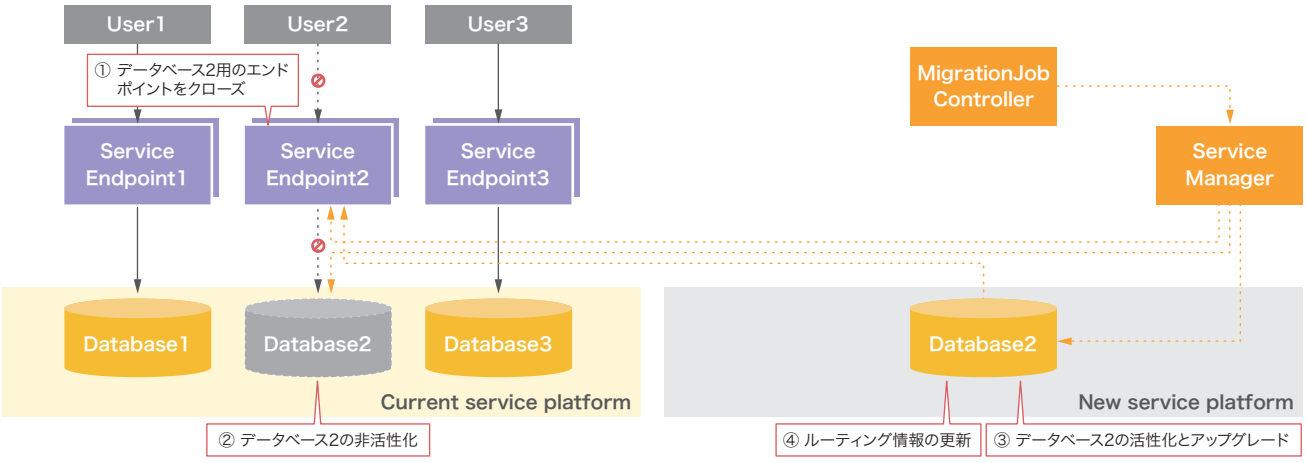

図-11 クエリサービス更新機能の内部動作2

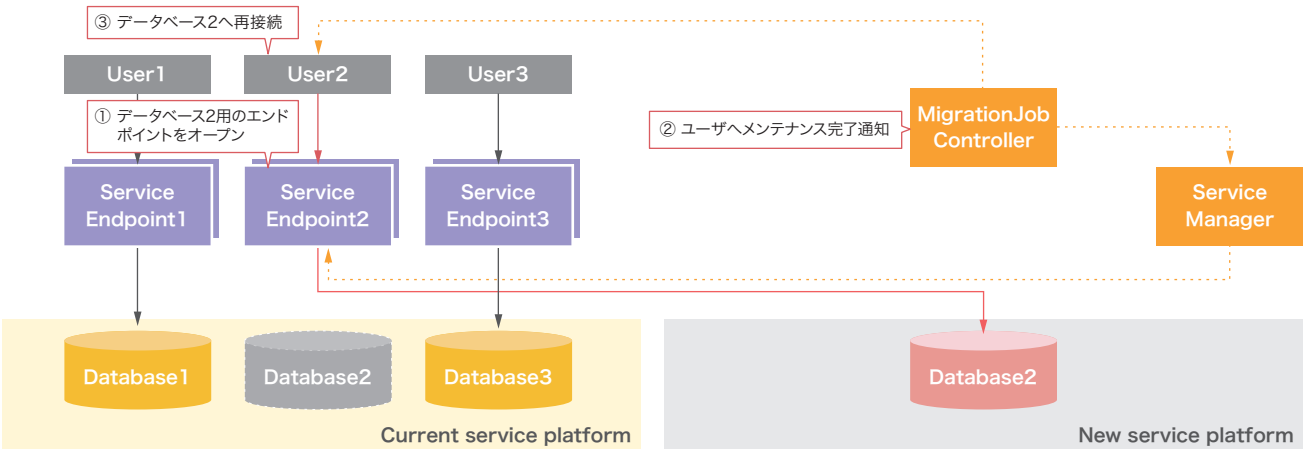

図-12 クエリサービス更新機能の内部動作3

切替が実行されると、まずクライアントとデータベース間に存在するプロキシであるサービスエンドポイント（以下、エンドポイント）を閉塞します。エンドポイントの閉塞によりクライアントとデータベース間の経路が消えるため、ユーザからのセッションは完全に切断されます。従って、もしトランザクション処理を実行中のセッションがある場合はデータベースにてロールバックされます。そのためトランザクションが実行されていない状態で実行されることが望ましいと言えます。エンドポイント閉塞完了後の現行データベースの整合性が確立された状態で、新データベースとの最終差分同期を実行します。

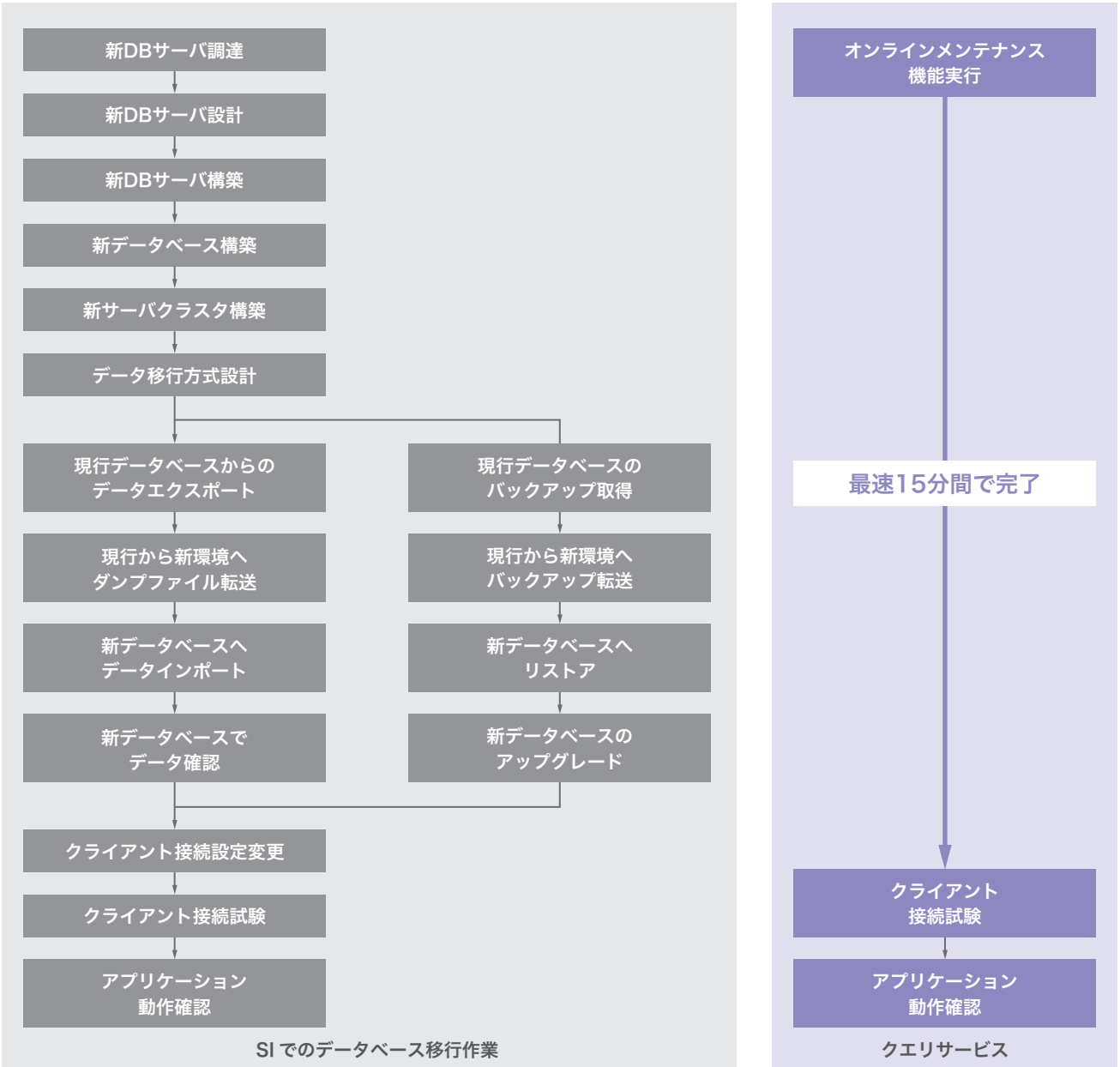

図-13 サービス更新機能による設備更改ステップの大幅な圧縮効果

サービス更新機能の処理は、従来はSIのプロジェクト作業として実施されてきました。これを手動で実行するとしたら図-13のようになります。なかなか骨の折れる手順だと思います。クエリサービスであれば裏側で自動処理されているので、サービスをご利用になる方はコントロールパネルから1クリックするだけで良いのです。

3.3 まとめ

クエリサービスはデータベースを作ったわけではなく、エンジニアが如何にデータベースと楽に付き合うかを求めたオーケストレーションシステムで、その目的はプロトタイプとはいえ達成できたように思います。また、本稿を読んだ方の中には、筆者がKubernetesに対抗心を持っているように感じ取られた方がいるかもしれませんが、実はKubernetes自体はとても好きで、機会があればKubernetesを使ってクエリサービスを開発してみたいものです。

テックチャレンジは、ユーザ要件がある従来の開発とは異なり、自分のアイデアを形にして良いという、エンジニアにとってこの上なく面白い時間で、社会人になってからすっかり忘れていた純粋なコンピュータの楽しさを味わった1年でした。

執筆者:
二ノ宮 務（にのみや つとむ）
IIJ システムクラウド本部 サービス企画室テクニカルマネージャー。
クラウドサービスの企画・開発やプロジェクトの技術支援に従事。元はDWH/BIシステムの開発者でSQLと並列処理を好む。

Information

IIJ Engineers Blog 最新トピック

開発・運用の現場エンジニアが執筆するIIJ公式ブログ「IIJ Engineers Blog」では、今年で4回目となる「IIJ 2020 TECHアドベントカレンダー」を開催。最新のコアな技術情報からゆるく柔らかい社内ネタまで、様々な技術記事を12月1日から24日のクリスマスイブまで毎日1記事ずつ公開してきました。

ここでは開催から1週間分の1日から7日までの記事をダイジェストでご紹介します。

▼COVID-19のラストマイル遅延への影響：
https://eng-blog.iij.ad.jp/archives/7834
ブロードバンド接続の要となるユーザ宅と最寄りの局舎を繋ぐラストマイル回線。この部分の通信がユーザの体感品質に直結します。本記事ではラストマイル回線の混雑状況、コロナ禍でどのような影響があったかを紹介します。

▼アクセス網はどれぐらい輻輳しているか？：
https://eng-blog.iij.ad.jp/archives/7740
コロナ禍でインターネット利用が増えてネットが混んで遅くなったという話をあちこちで耳にしました。しかし、実際に何処がどの程度混んでいるのかを把握することは大変難しい課題です。この記事では、なぜ輻輳の把握が難しいのかを説明するとともに、研究のアプローチを紹介します。

▼Teams / Webex 対応！ Linux でバーチャル背景を使う：
https://eng-blog.iij.ad.jp/archives/7345
Linux版のTeamsではバーチャル背景機能は公式には使えないと言われています。しかしエンジニアは諦めません。クライアント側で対応されないのなら、Webカメラ側に細工をすれば良い、ということで、その実現方法をご紹介します。

▼マンホールをたずねて三千里：
https://eng-blog.iij.ad.jp/archives/7186
マンホールの中には通信ケーブルが通っています。バックボーンを構築する上で欠かせない存在。その蓋には敷設した会社のロゴやマークが描かれています。様々な通信系のマンホール蓋を、マンホール探しが趣味の筆者が紹介します。

▼コロナ禍のIT勉強会、リアルからオンラインへの切り替えで考えた2つのコト：
https://eng-blog.iij.ad.jp/archives/7141
IIJのIT勉強会は、講師と参加者、参加者同士の「リアルな交流」を大切にしてきました。コロナ禍でオンライン開催を余儀なくされたなかで、リアルな交流をどう実現するか、オンラインだからこそできるチャレンジ、など、完全オンライン開催に舵をきるにあたって考えたことをご紹介します。

▼配信構成・設備、動画編集、本番 〜IT勉強会をオンライン配信する傍らでやってたコトその1：
https://eng-blog.iij.ad.jp/archives/7044
IT勉強会のオンライン開催にあたって、配信の裏側を、実際に配信を担当したエンジニアが技術的視点から紹介します。

全記事一覧＞＞
https://eng-blog.iij.ad.jp/adventcalendar2020

■ IIJ Technical WEEK 2020開催報告

去る12月14日(月)～17日(木)の4日間、ITエンジニアを対象とした技術イベント「IIJ Technical WEEK 2020」を開催しました。IIJ Technical WEEKは、日頃IIJのエンジニアが携わっているサービス開発・運用に関する技術をはじめ、ココでしか聞けないノウハウ、業界ネタなどをご紹介する技術イベント。今年は1テーマずつ、夕方1時間のオンラインセッションを4日間行いました。

テーマは「セキュリティ」「インターネット・バックボーン」「アプリケーション」「データセンター」。講演や座談会、トークコーナーを各日1時間に凝縮してお届けしました。

▼日時/プログラム

12月14日(月)16:00～17:00
『2020年のセキュリティ振り返り・お客様担当セキュリティエンジニア座談会』

セキュリティ本部長 齋藤衛による「セキュリティ脅威動向2020」に加えて、現場のセキュリティエンジニアが現場の視点で、いま感じていること、やらなきゃいけないことを語りつくす座談会。

詳細:https://iij.connpass.com/event/196656/

12月15日(火)16:00～17:00
『世界のインターネットの今・あらゆるトラフィックをさばくIIJバックボーン』

「インターネット・バックボーン」をテーマに、インターネットの世界で起こっていること、バックボーン運用のリアルを紹介。講演後には講演者がざっくばらんにお話するトークコーナーも。

詳細:https://iij.connpass.com/event/196663/

12月16日(水)16:00～17:00
『開発研修をオンライン化するツール開発解説WebAssembly・確定間近！QUIC(HTTP/3)』

「アプリケーション」をテーマに、コロナ禍のオンライン社内研修のために有志エンジニアが短期間でWebアプリを開発した話と、技術トレンドとして「WebAssembly」「QUIC(HTTP/3)」についての解説。

詳細:https://iij.connpass.com/event/196667/

12月17日(木)16:00～17:00
『データセンターを作る:
コロナ禍の運用体制・蓄電池で逼迫する夏の電力消費を抑える』

「データセンター」をテーマに、コロナ禍の運用体制や、IIJならではの蓄電池を活用した省エネの取り組みを紹介。

詳細:https://iij.connpass.com/event/196669/

なお、当日の資料は以下のブログ記事で紹介しています。
https://eng-blog.iij.ad.jp/archives/8429

December 2020 Vol.49

Internet Initiative Japan

株式会社インターネットイニシアティブ(IIJ)について

IIJは、1992年、インターネットの研究開発活動に関わっていた技術者が中心となり、日本でインターネットを本格的に普及させようという構想を持って設立されました。
現在は、国内最大級のインターネットバックボーンを運用し、インターネットの基盤を担うと共に、官公庁や金融機関をはじめとしたハイエンドのビジネスユーザに、インターネット接続やシステムインテグレーション、アウトソーシングサービスなど、高品質なシステム環境をトータルに提供しています。
また、サービス開発やインターネットバックボーンの運用を通して蓄積した知見を積極的に発信し、社会基盤としてのインターネットの発展に尽力しています。

株式会社インターネットイニシアティブ
〒102−0071 東京都千代田区富士見2-10-2　飯田橋グラン・ブルーム
E-mail : info@iij.ad.jp　URL : https://www.iij.ad.jp