# IIR

Internet
Infrastructure
Review

Sep.2018
Vol.40

定期観測レポート

## ブロードバンドトラフィックレポート
## –ダウンロード量の増加率は2年連続で減少–

フォーカス・リサーチ(1)

## セキュリティ関連文書のレコメンデーション

フォーカス・リサーチ(2)

## クラウドとKubernetes

IIJ
Internet Initiative Japan

# Internet Infrastructure Review

September 2018 Vol.40

## エグゼクティブサマリ

我が国の電気通信事業を律する「電気通信事業法」は2015年に改正されました。施行から3年を経た時点で、改正後の規定の施行状況を検討し、必要に応じて措置を講ずるものと定められています。それを踏まえて、去る8月23日、情報通信審議会に対して「電気通信事業分野における競争ルールなどの包括的検証」が諮問され、これから2030年頃を見据えた本格的な検討が始まります。

内容は、通信ネットワーク全体に関するビジョン、通信基盤の整備などの在り方、ネットワーク中立性の在り方、プラットフォームサービスに関する課題への対応の在り方、モバイル市場の競争環境の確保の在り方、消費者保護ルールの在り方など、多岐にわたります。

インターネットは、ネットワークの仮想化やソフトウェア制御などの技術の発展もあり、多様なプレーヤーによる技術開発、サービス開発が進んでいます。今後、AI、IoT、5Gといった技術の本格的な利用が進むなか、技術開発のみならず、法制度などの整備についても注視し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ご紹介しており、我々が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、ブロードバンドトラフィックレポートです。これはIIJが運用しているブロードバンド接続サービスのトラフィックを分析した報告で、2009年から毎年お届けしているものです。総務省から公表される「我が国のインターネットにおけるトラヒックの集計・試算」はトラフィック全体の集計ですが、小誌では利用者の1日の使用量の分布やポート別使用量の分析を行っています。トラフィックの伸びは鈍化しているものの、継続的に増加しており、4年程前から利用が大きく拡大しているHTTPSも増加している、という結果が出ています。WebブラウザがHTTPを安全でないと表示したり、検索サイトにおいてHTTPのみのサイトの優先順位が下がる、などという動きも出ており、HTTPからHTTPSへの移行は、ますます進むと思われます。

2章では、フォーカスリサーチ(1)として、非定型情報を扱うための自然言語処理とトピックモデルの実験をご紹介します。セキュリティ対応においては、IPアドレスのブラックリストやSCAPなど、定型化されて機械的な処理が容易な情報は、支援システムで広く活用されていますが、画像や自然言語で書かれた文書など、機械的な処理がむずかしい情報の活用には、まだ課題が残されています。そこで、セキュリティ対応において、そのような非定型情報を活用するためのレコメンドシステムのプロトタイプを開発しました。満足のいく結果は得られなかったものの、一定の条件下では活用できるという手応えを掴むことができました。

3章では、フォーカスリサーチ(2)として、Kubernetesを取り上げます。クラウド関連の情報を収集するなかで、dockerやKubernetesといった言葉を聞く機会が増えていると思います。どちらも昨今のコンテナ技術の中核となるものです。ここでは、dockerとKubernetesの機能や役割、IaaSやハイブリッドクラウドにおいてKubernetesを利用する意義を説明したうえで、IIJが構築したIKE(IIJ Container Engine for Kubernetes)をご紹介します。実際にKubernetesを利用したコンテナクラスタ環境はどのようなもので、どのような狙いを持っているのかを解説していますので、これから取り組もうとされている皆様には参考になるのではないでしょうか。

IIJは、このような活動を通じて、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続けています。今後も、お客様の企業活動のインフラとして最大限にご活用ただけるよう、様々なサービス及びソリューションを提供してまいります。

**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。

# ブロードバンドトラフィックレポート –ダウンロード量の増加率は2年連続で減少–

## 1.1 概要

このレポートでは、毎年IIJが運用しているブロードバンド接続サービスのトラフィックを分析して、その結果を報告しています[*1*2*3*4*5*6*7*8*9]。今回も、利用者の1日のトラフィック量やポート別使用量などを基に、この1年間のトラフィック傾向の変化を報告します。

図-1は、IIJのブロードバンドサービス及びモバイルサービス全体について月平均トラフィック量の推移を示したグラフです。トラフィックのIN/OUTはISPから見た方向を表し、INは利用者からのアップロード、OUTは利用者へのダウンロードとなります。トラフィック量の数値は開示できないため、それぞれのOUTの最新値を1として正規化しています。ブロードバンドに関しては、今回からIPv6 IPoEのトラフィック量も含めて示しています。IPv6 IPoEを含まない分は、“broadband-IPoE”として細線で示します。IIJのブロードバンドにおけるIPv6は、IPoE方式とPPPoE方式がありますが[*10]、IPoEトラフィックはインターネットマルチフィード社のtransixサービスを利用していて直接IIJの網を通らないため、以降の解析の対象にはなっていません。2018年6月時点で、IPoEのブロードバンドトラフィック量の全体に占める割合は、INで12%、OUTで8%です。

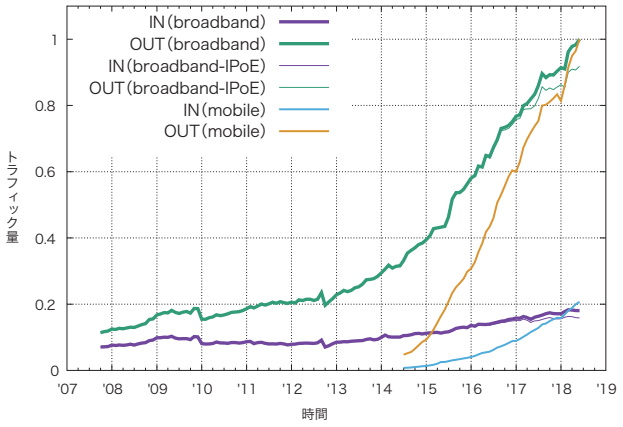


**図-1 ブロードバンド及びモバイルの月間トラフィック量の推移**

ブロードバンド、モバイル共に、昨年の後半に一時期伸びが鈍りましたが、今年に入って再度伸びてきて、元の成長曲線に戻ってきました。この1年のブロードバンドトラフィック量は、INは12%の増加、OUTは20%の増加となっています。1年前はそれぞれ10%と25%の増加、2年前は18%と47%の増加でしたので、ダウンロード量は2年連続で伸びが鈍ったことになります。モバイルに関しては、この4年の数字しかありませんが、この1年でINは69%の増加、OUTは36%の増加と、こちらも1年前の103%と70%に比べると伸びが鈍化しているものの、依然大きく伸びています。ただし、総量ではまだブロードバンドより1桁少ない状況が続いています。

## 1.2 データについて

今回も前回までと同様に、ブロードバンドに関しては、個人及び法人向けのブロードバンド接続サービスについて、ファイバーとDSLによるブロードバンド顧客を収容するルータで、Sampled NetFlowにより収集した調査データを利用しています。モバイルに関しては、個人及び法人向けのモバイルサービスについて、使用量についてはアクセスゲートウェイの課金用情報を、使用ポートについてはサービス収容ルータでのSampled NetFlowデータを利用しています。

トラフィックは平日と休日で傾向が異なるため、1週間分のトラフィックを解析しています。今回は、2018年5月28日から6月


*1 長健二朗. ブロードバンドトラフィックレポート: トラフィック増加はややペースダウン. Internet Infrastructure Review. vol.36. pp4-9. August 2017.
*2 長健二朗. ブロードバンドトラフィックレポート: 加速するトラフィック増加. Internet Infrastructure Review. vol.32. pp28-33. August 2016.
*3 長健二朗. ブロードバンドトラフィックレポート: ブロードバンドとモバイルのトラフィックを比較. Internet Infrastructure Review. vol.28. pp28-33. August 2015.
*4 長健二朗. ブロードバンドトラフィックレポート: この一年でトラフィック量は着実に増加、HTTPS の利用が拡大. Internet Infrastructure Review. vol.24. pp28-33. August 2014.
*5 長健二朗. ブロードバンドトラフィックレポート: 違法ダウンロード刑事罰化の影響は限定的. Internet Infrastructure Review. vol.20. pp32-37. August 2013.
*6 長健二朗. ブロードバンドトラフィックレポート: この1 年間のトラフィック傾向について. Internet Infrastructure Review. vol.16. pp33-37. August 2012.
*7 長健二朗. ブロードバンドトラフィックレポート: マクロレベルな視点で見た、震災によるトラフィックへの影響. Internet Infrastructure Review. vol.12. pp25-30. August 2011.
*8 長健二朗. ブロードバンドトラフィックレポート: P2P ファイル共有からWeb サービスへシフト傾向にあるトラフィック. Internet Infrastructure Review. vol.8. pp25-30. August 2010.
*9 長健二朗. ブロードバンドトラフィック: 増大する一般ユーザのトラフィック. Internet Infrastructure Review. vol.4. pp18-23. August 2009.
*10 小川晃通. プロフェッショナルIPv6. 付録A.3. IPv6 PPPoEとIPv6 IPoE. ラムダノート. July 2018.

3日の1週間分のデータを使っていて、前回解析した2017年5月29日から6月4日の1週間分と比較します。

ブロードバンドの集計は契約ごとに行い、一方モバイルでは複数電話番号の契約があるので電話番号ごとの集計となっています。ブロードバンド各利用者の使用量は、利用者に割り当てられたIPアドレスと、観測されたIPアドレスを照合して求めています。また、NetFlowではパケットをサンプリングして統計情報を取得しています。サンプリングレートは、ルータの性能や負荷を考慮して、1/8192 ～1/16382に設定されています。観測された使用量に、サンプリングレートの逆数を掛けることで全体の使用量を推定しています。

IIJの提供するブロードバンドサービスにはファイバー接続とDSL接続がありますが、今ではファイバー接続の利用がほとんどとなっています。2018年には観測されたユーザ数の97%はファイバー利用者で、ブロードバンドトラフィック量全体の99%を占めています。

## 1.3 利用者の1日の使用量

まずは、ブロードバンド及びモバイル利用者の1日の利用量をいくつかの切り口から見ていきます。ここでの1日の利用量は各利用者の1週間分のデータの1日平均です。

図-2及び図-3は、ブロードバンドとモバイル利用者の1日の平均利用量の分布（確率密度関数）を示します。アップロード（IN）とダウンロード（OUT）に分け、利用者のトラフィック量をX軸に、その出現確率をY軸に示していて、2017年と2018年を比較しています。X軸はログスケールで、10KB（$10^4$）から100GB（$10^{11}$）の範囲を示しています。一部の利用者はグラフの範囲外にありますが、概ね100GB（$10^{11}$）までの範囲に分布しています。

ブロードバンドのINとOUTの各分布は、片対数グラフ上で正規分布となる、対数正規分布に近い形をしています。これはリニアなグラフで見ると、左端近くにピークがあり右へなだらかに減少するいわゆるロングテールな分布です。OUTの分布はINの分布より右にずれていて、ダウンロード量がアップロード量より、1桁以上大きくなっています。2017年と2018年で比較すると、INとOUT共に分布の山が僅かながら右に移動しており、利用者全体のトラフィック量が増えていることが分かります。

右側のOUTの分布を見ると、分布のピークはここ数年間で着実に右に移動していますが、右端のヘビーユーザの使用量はあまり増えておらず、分布の対称性が崩れてきています。一方で、左側のINの分布は左右対称で、より対数正規分布に近い形です。

図-3のモバイルの場合、ブロードバンドに比べて利用量は大幅に少ないことが分かります。また、使用量に制限があるため、分布右側のヘビーユーザの割合が少なく、左右非対称な形になります。極端なヘビーユーザも存在しません。外出時のみの利用や、使用量の制限のため、各利用者の日ごとの利用量のばらつ

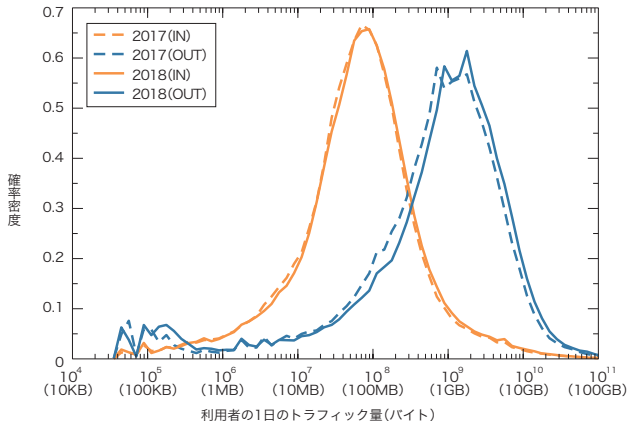


図-2 ブロードバンド利用者の1日のトラフィック量分布
2017年と2018年の比較

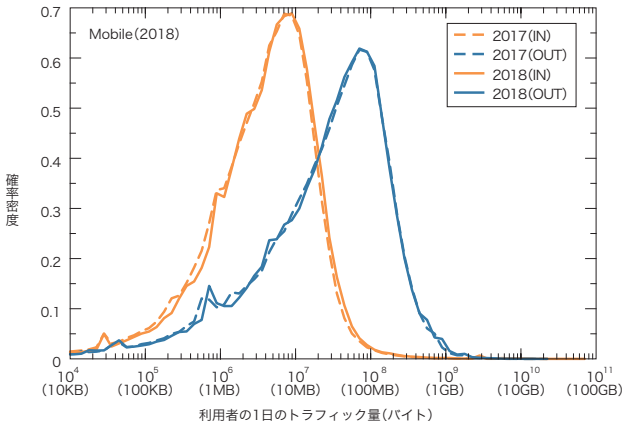


図-3 モバイル利用者の1日のトラフィック量分布
2017年と2018年の比較

きはブロードバンドより大きくなります。そのため、1週間分のデータから1日平均を求めると、1日単位で見た場合より利用者間のばらつきは小さくなります。1日単位で同様の分布を描くと、分布の山が少し低くなり、その分両側の裾が持ち上がりますが、基本的な分布の形や最頻出値はほとんど変わりません。

表-1は、ブロードバンド利用者の1日のトラフィック量の平均値と中間値、分布の山の頂点にある最頻出値の推移を示します。分布の山に対して頂点が少しずれているので、最頻出値は分布の山の中央に来るように補正しています。

| | IN (MB/day) | | | OUT (MB/day) | | |
|---|---|---|---|---|---|---|
| 年 | 平均値 | 中間値 | 最頻出値 | 平均値 | 中間値 | 最頻出値 |
| 2005 | 430 | 3 | 3.5 | 447 | 30 | 32 |
| 2007 | 433 | 5 | 4 | 712 | 58 | 66 |
| 2008 | 483 | 6 | 5 | 797 | 73 | 94 |
| 2009 | 556 | 7 | 6 | 971 | 88 | 114 |
| 2010 | 469 | 8 | 7 | 910 | 108 | 145 |
| 2011 | 432 | 9 | 8.5 | 1,001 | 142 | 223 |
| 2012 | 410 | 12 | 14 | 1,026 | 173 | 282 |
| 2013 | 397 | 14 | 18 | 1,038 | 203 | 355 |
| 2014 | 437 | 22 | 28 | 1,287 | 301 | 447 |
| 2015 | 467 | 33 | 40 | 1,621 | 430 | 708 |
| 2016 | 475 | 48 | 56 | 2,081 | 697 | 1,000 |
| 2017 | 520 | 63 | 79 | 2,624 | 835 | 1,260 |
| 2018 | 582 | 67 | 79 | 3,139 | 1021 | 1,413 |

**表-1 ブロードバンド利用者の1日のトラフィック量の平均値と最頻出値の推移**

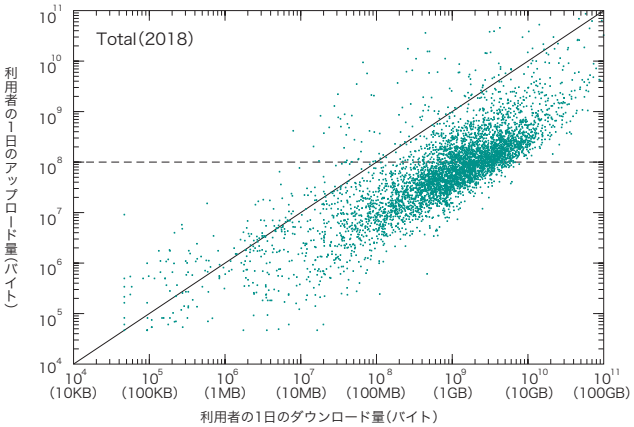


**図-4 ブロードバンド利用者ごとのIN/OUT使用量**

分布の最頻出値を2017年と2018年で比較すると、INでは79MBと前年から変わらず、OUTでは1260MBから1413MBに増えており、伸び率で見ると、INとOUTそれぞれで1倍と1.1倍になっています。一方、平均値はグラフ右側のヘビーユーザの使用量に左右されるため、2018年には、INの平均は582MB、OUTの平均は3139MBと、最頻出値よりかなり大きな値になりました。2017年には、それぞれ520MBと2624MBでした。モバイルでは、表-2に示すように、ヘビーユーザが少ないため平均と最頻出値が近い値になります。2018年の最頻出値は、INで7MB、OUTで79MBで、平均値は、INで17.0MB、OUTで81.9MBです。最頻出値は、INもOUTも昨年と同じ値となっています。中間値と最頻出値はほとんど変わっていないのに、平均値が増加していて、ヘビーユーザが特にIN側で増えていることが窺えます。

図-4及び図-5では、利用者5,000人をランダムに抽出し、利用者ごとのIN/OUT使用量をプロットしています。X軸はOUT(ダウンロード量)、Y軸はIN(アップロード量)で、共にログス

| | IN (MB/day) | | | OUT (MB/day) | | |
|---|---|---|---|---|---|---|
| 年 | 平均値 | 中間値 | 最頻出値 | 平均値 | 中間値 | 最頻出値 |
| 2015 | 6.0 | 2.7 | 5.5 | 46.6 | 19 | 40 |
| 2016 | 7.8 | 3.6 | 7 | 63.0 | 27 | 63 |
| 2017 | 12.0 | 4.3 | 7 | 77.4 | 35 | 79 |
| 2018 | 17.0 | 4.7 | 7 | 81.9 | 36 | 79 |

**表-2 モバイル利用者の1日のトラフィック量の平均値と最頻出値**

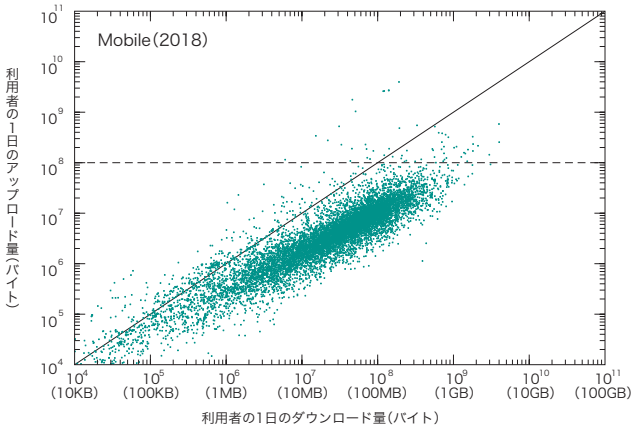


**図-5 モバイル利用者ごとのIN/OUT使用量**

ケールです。利用者のIN/OUTが同量であれば対角線上にプロットされます。

対角線の下側に対角線に沿って広がるクラスタは、ダウンロード量が1桁多い一般的なユーザです。ブロードバンドでは、以前は右上の対角線上あたりを中心に薄く広がるヘビーユーザのクラスタがはっきり分かりましたが、今では識別ができなくなっています。また、各利用者の使用量やIN/OUT比率にも大きなばらつきがあり、多様な利用形態が存在することが窺えます。ここでは、2017年と比較しても、違いはほとんど確認できません。

モバイルでも、OUTが1桁多い傾向は同じですが、ブロードバンドに比べて利用量は少なく、IN/OUTのばらつきも小さくなっています。また、クラスタの傾きは対角線より小さくなっており、使用量の多いユーザほどダウンロード比率が高くなっていることが分かります。

図-6及び図-7は、利用者の1日のトラフィック量を相補累積度分布にしたものです。これは、使用量がX軸の値より多い利用者の、全体に対する割合をY軸に、ログ・ログスケールで示したもので、ヘビーユーザの分布を見るのに有効です。グラフの右側が直線的に下がっていて、べき分布に近いロングテールな分布であることが分かります。ヘビーユーザは統計的に分布しており、決して一部の特殊な利用者ではないと言えます。モバイルでも、OUT側ではヘビーユーザはべき分布していますが、IN側では昨年よりも直線的な傾きが崩れていて、大量にアップロードするユーザの割合が大きくなっています。

利用者間のトラフィック使用量の偏りを見ると、使用量には大きな偏りがあり、結果として全体は一部利用者のトラフィックで占められています。例えば、ブロードバンド上位10%の利用者がOUTの60%、INの86%を占めています。更に、上位1%の利用者がOUTの25%、INの59%を占めています。ここ数年のヘビーユーザ割合の減少に伴い、僅かながら偏りは減ってきています。モバイルでは、上位10%の利用者がOUTの50%、INの70%を、上位1%の利用者がOUTの15%、INの52%を占めています。ここ数年でヘビーユーザの割合が着実に増えています。

## 1.4 ポート別使用量

次に、トラフィックの内訳をポート別の使用量から見ていきます。最近では、ポート番号からアプリケーションを特定することは困難です。P2P系アプリケーションには、双方が動的ポートを使うものが多く、また、多くのクライアント・サーバ型アプリケーションが、ファイアウォールを回避するため、HTTPが使う80番ポートを利用します。大まかに分けると、双方が1024番以上の動的ポートを使っていればP2P系のアプリケーションの可能性が高く、片方が1024番未満のいわゆるウェルノウンポートを使っていれば、クライアント・サーバ型のアプリケーションの可能性が高いと言えます。そこで、TCPとUDPで、ソースとデスティネーションのポート番号の小さい方を取り、ポート番号別の使用量を見てみます。

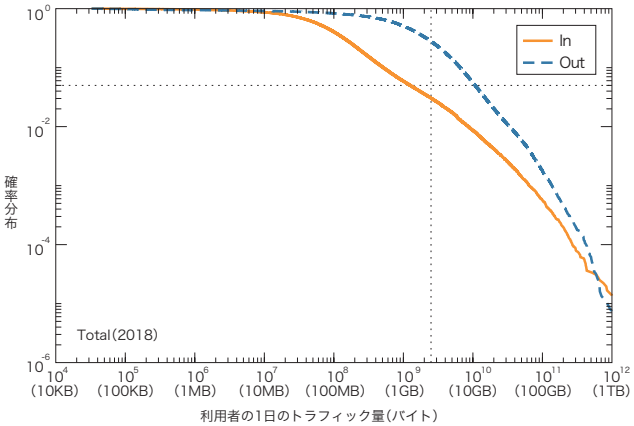


図-6 ブロードバンド利用者の1日のトラフィック量の相補累積度分布

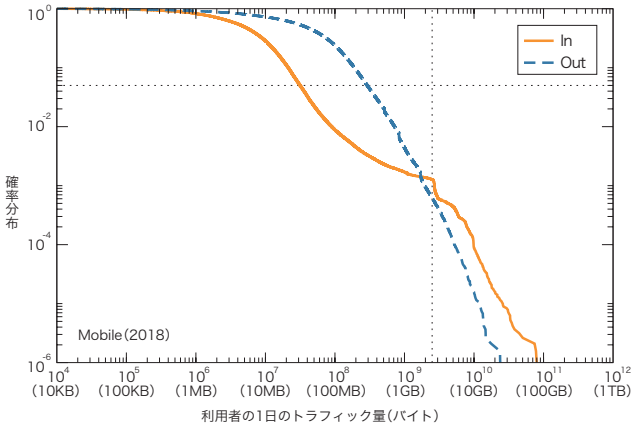


図-7 モバイル利用者の1日のトラフィック量の相補累積度分布

表-3はブロードバンド利用者のポート使用割合の過去4年間の推移を示します。2018年の全体トラフィックの79%はTCPです。前回まで増加を続けていたHTTPSの443番ポートの割合は、2017年の43%から41%に減っています。HTTPの80番ポートの割合も、2017年の28%から27%に減っており、GoogleのQUICプロトコルで使われるUDPの443番ポートが、11%から16%に増えています。このことから、以前から続いているHTTPからHTTPSへの移行が、更にQUICへと移ってきたことが分かります。減少傾向のTCPの動的ポートは、2017年の11%から2018年には10%にまで減りました。動的ポートでの個別のポート番号の割合は僅かで、Flash Playerが利用する1935番が最大で総量の約0.7%ありますが、後は0.3%未満となっています。TCP以外のトラフィックでは、UDPの443番ポート以外は、ほとんどVPN関連です。

表-4はモバイル利用者のポート使用割合です。ブロードバンドと比べると、HTTPSの割合が多くなっていますが、全体的にはブロードバンドの数字に近い値となっており、モバイル利用者もブロードバンドと同様のアプリケーションの使い方をしていることが窺えます。

図-8は、ブロードバンド全体トラフィックにおける主要ポート利用の週間推移を、2017年と2018年で比較したものです。TCPポートの80番、443番、1024番以上の動的ポート、UDPポート443番の4つに分けてそれぞれの推移を示しています。今回から、利用量の少なくなったウェルノウンポートに代わって、UDPポート443番を加えています。グラフでは、ピーク時の総トラフィック量を1として正規化して表しています。全体のピークは19:00から23:00頃で、443番ポートのピークは80番ポートの

| year | 2015 | 2016 | 2017 | 2018 |
|---|---|---|---|---|
| protocol port | (%) | (%) | (%) | (%) |
| **TCP** | **80.8** | **82.8** | **83.9** | **78.5** |
| (< 1024) | 63.3 | 69.1 | 72.9 | 68.5 |
| 443(https) | 23.3 | 30.5 | 43.3 | 40.7 |
| 80(http) | 37.9 | 37.1 | 28.4 | 26.5 |
| 182 | 0.4 | 0.3 | 0.3 | 0.3 |
| 993(imaps) | 0.1 | 0.1 | 0.2 | 0.2 |
| 22(ssh) | 0.2 | 0.2 | 0.1 | 0.1 |
| (>= 1024) | 17.5 | 13.7 | 11.0 | 10.0 |
| 1935(rtmp) | 1.8 | 1.5 | 1.1 | 0.7 |
| 8080 | 0.3 | 0.2 | 0.3 | 0.3 |
| **UDP** | **11.4** | **11.1** | **10.5** | **16.4** |
| 443(https) | 0.9 | 2.4 | 3.8 | 10.0 |
| 4500(nat-t) | 0.2 | 0.2 | 0.2 | 0.2 |
| **ESP** | **7.4** | **5.8** | **5.1** | **4.8** |
| **IP-ENCAP** | **0.2** | **0.2** | **0.3** | **0.2** |
| **GRE** | **0.2** | **0.1** | **0.1** | **0.1** |
| **ICMP** | **0.0** | **0.0** | **0.0** | **0.0** |

**表-3 ブロードバンド利用者のポート別使用量**

| year | 2015 | 2016 | 2017 | 2018 |
|---|---|---|---|---|
| protocol port | (%) | (%) | (%) | (%) |
| **TCP** | **93.8** | **94.4** | **84.4** | **76.6** |
| 443(https) | 37.4 | 43.7 | 53.0 | 52.8 |
| 80(http) | 52.5 | 46.8 | 27.0 | 16.7 |
| 31000 | 0.0 | 0.2 | 1.8 | 2.9 |
| 993(imaps) | 0.5 | 0.5 | 0.4 | 0.3 |
| 1935(rtmp) | 0.5 | 0.3 | 0.2 | 0.1 |
| **UDP** | **5.2** | **5.0** | **11.4** | **19.4** |
| 443(https) | 1.0 | 1.5 | 7.5 | 10.6 |
| 4500(nat-t) | 0.3 | 0.2 | 0.2 | 4.5 |
| 12222 | 0.0 | 0.1 | 0.1 | 2.3 |
| 53(dns) | 0.1 | 0.2 | 0.1 | 0.1 |
| **ESP** | **0.7** | **0.4** | **0.4** | **3.9** |
| **GRE** | **0.3** | **0.1** | **0.1** | **0.1** |
| **ICMP** | **0.0** | **0.0** | **0.0** | **0.0** |

**表-4 モバイル利用者のポート別使用量**

ピークより若干早くなっています。土日には昼間のトラフィックが増加しており、家庭での利用時間を反映しています。

図-9のモバイルでは、トラフィックの大半を占めるTCP80番ポートと443番ポート、UDP443番ポートについて推移を示します。ブロードバンドに比べると、朝から夜中までトラフィックの高い状態が続きます。平日には、朝の通勤時間、昼休み、夕方17:00頃から22:00頃にかけての3つのピークがあり、ブロードバンドとは利用時間の違いがあることが分かります。

## 1.5 まとめ

この1年間のブロードバンドトラフィックの傾向として、昨年後半にややブレーキがかかったトラフィック増加が、今年に入って再度増加傾向にあることが挙げられます。この1年間で見るとダウンロード量は20%、アップロード量は12%増加していますが、ダウンロードの伸び率は2年連続して低下しています。

モバイルトラフィックについては、増加率は少し下がって来たものの、この4年間で大きく伸びてきています。ブロードバンドと比較すると、ヘビーユーザの割合が少なく、利用時間では平日の通勤時間帯や昼休みの利用が目立つなどの違いがあります。

また、4年ほど前からHTTPSの利用が大きく拡大していて、TCPとUDPの443番を合わせるとブロードバンドの51%、モバイルの63%になります。最近では、WebブラウザがHTTPを安全でないと表示したり、HTTPのみのサイトは検索サイトで優先順位が下がったりと、HTTPSへの移行の圧力が強まっているので、今後もHTTPの減少が続くと予想されます。

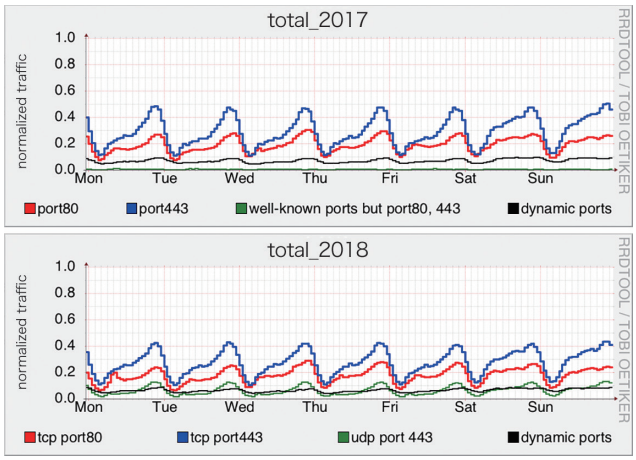


図-8 ブロードバンド利用者のTCPポート利用の週間推移
2017年(上)と2018年(下)

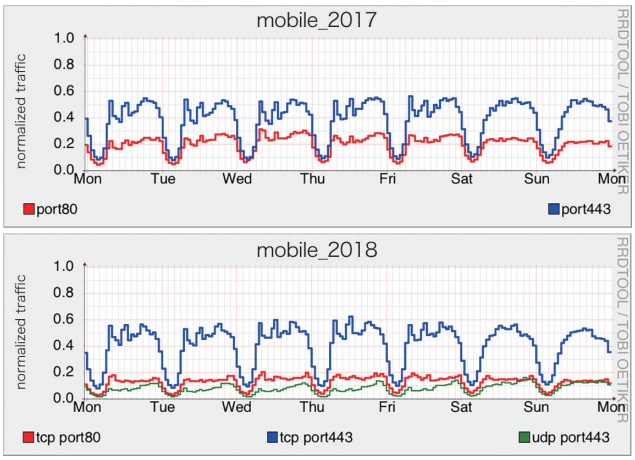


図-9 モバイル利用者のTCPポート利用の週間推移
2017年(上)と2018年(下)

執筆者:
**長 健二朗**(ちょう けんじろう)
株式会社IIJ イノベーションインスティテュート 技術研究所所長。

# セキュリティ関連文書のレコメンデーション

## 2.1 セキュリティ対応で取り扱う情報

SOCやCSIRTなどのセキュリティ対応を行う組織を運用していると、いやおうなく日々多くの情報と向き合うことになります。「情報」と一言で書きましたが、そこには性質も扱い方も様々なものが含まれています。多くの組織で、セキュリティ対応を行っている各人が、それぞれの仕事に必要な情報を収集、編集、作成していることでしょう。

システム開発の視点から見ると、これらの情報は大きく2つに分けられます。機械的な処理がしやすいように定型化されたものと、非定型なものです。

セキュリティ対応組織で使用することがある定型化された情報の例としては、IPアドレスのブラックリストやTCP/IPのポートデータベース、セキュリティ対策を自動化するために策定されたSCAPなどが挙げられます。意味付けをして定型化された情報は機械的な処理がしやすいため、セキュリティ対応を支援するようなシステムでは、これらの情報が広く活用されています。

一方、セキュリティ対応で同じく必要となる定型化されていない情報、例えば自然言語で書かれた文書や画像などについては機械的な処理がしづらく、対応支援システムでも参考情報などで文書として表示する以外の扱いが難しいことがありました。非定型の文書や画像は、そのときに作成したかったレポートなどに記述することで単に蓄積するだけになるなど、その後の活用が人的努力に任されていることが多いのではないでしょうか。

では、非定型の情報を必要なときに取り出す、例えばセキュリティ対応中に関連情報として参照するにはどうするのが良いのでしょうか。

## 2.2 非定型な情報の取り扱い

このような問題を解決しようとしている技術はいくつかあり、そのうち最初に思い付くのが全文検索システムです。作成した文書を集めて全文検索システムに保管して、必要なときにキーワード検索して探し出すのがとても便利であることは誰もが実感しているでしょう。

一方で、ユーザが何かを探していなくても重要と思われるものを提示するシステムがあります。ショッピングサイトでオススメされる商品や、ニュースサイトなどで表示される関連記事がその一例で、このようなものをレコメンドシステムと呼びます。

便利で利用実績も多いこのような技術を手元にある自然言語で記述された文書に対しても使いたいと考えたとき、全文検索システムの導入は容易でしたが、レコメンドシステムなどのいわゆる集合知を活用するシステムの導入は困難でした。社内にあるセキュリティ対応に必要な情報には関係者外秘のものが多く、それを使用するのも組織内の限られたメンバーのみであるため、そもそものユーザ数が少なく集合知を活用できる程のデータが蓄積できません。

このような前提を踏まえて、ユーザの行動履歴を使わずに非定型の文書そのものの情報だけをもとにお勧めを決める手法を試してみたいと思います。この手法の利点として、ユーザの行動や文書そのものを外部に出すことなく扱えることも挙げられます。ユーザがある文書を選んだとき、それと関連のある文書をお勧めすることを考えてみましょう。

## 2.3 自然言語処理とトピックモデル

自然言語で書かれた非定型の文書を扱うための技術はどのようなものでしょうか。このような技術は自然言語処理と呼ばれる分野で長く研究されてきており、様々な要素技術が存在しています。その1つに、機械学習の技術で文書データを解析するトピックモデルという手法があります(図-1)。

### 2.3.1 トピックモデル

文書には様々な種類のものがあります。普段私たちが読む文書に限っても、技術文書とニュース記事は違う種類の文書だと思えます。ニュース記事にも国際情勢を報じたもの、スポーツの結果を報じたものなど、いくつもの種類があります。そして文書の種類によって、使われる言葉の種類や頻度に違いがあると考えられます。また、1つの文書が属する種類は1つであるとは限りません。例えば国際紛争に起因して発生したインターネット上の攻撃のことを書いた文書は、国際情勢と情報セキュリティ、どちらにも属することになるでしょう。

トピックモデルでは文書の属する種類のことをトピックと呼び、文書は次のようにしてできあがると仮定しています。

まず、ある確率分布に従って文書のトピックの混ざり具合が決まります。そこからトピックごとの言葉の出現確率を使って文書が言葉で埋まっていき、最終的な文書ができあがります。実際の文書データから、これらトピックや単語に関する確率分布を求めておけば、今注目されているトピックが何か調べたり、トピックに基づいて文書を分類したりといったことに応用できます。

このような考え方に基づく手法を総称してトピックモデルと呼んでいます。潜在的ディリクレ配分法(Latent Dirichlet Allocation、LDA)という手法を代表として、様々なバリエーションが考え出され研究されています。

関連度の高い文書同士ではトピックの分布も近くなると期待できるので、これを利用して、ユーザに選ばれた文書とトピック分布の近い文書をお勧めとして提示する方法を取ることにします。本稿ではトピック分布の計算にはLDAを使います。

### 2.3.2 自然言語処理技術による前処理

LDAのアルゴリズムは文書のリストを渡すだけで処理してくれるわけではなく、テキスト内の単語とその出現回数を入力するものです。つまり、これに合うよう文書に様々な前処理を施す必要があります。同時に、LDAのモデル生成の際の精度向上や必要なデータ量の低減を狙い、不要と考えられる情報を削減します。

必要とされる前処理は以下のようなものです。

1. 文書データから本文を抽出
   文書データから本文以外の部分を取り除きます。
2. 単語に分割
   英語の文章であれば、基本的にはスペースや改行で分割すれば概ね実現できます。ただし、例えば一行の文字数制限でハイフネーション後に改行された場合など、いくつか対応が必要なポイントがあります。活用形を同じ単語として扱いたければ[*1]、stemmingやlemmatizingといった自然言語処理の技術を使います。日本語の文章

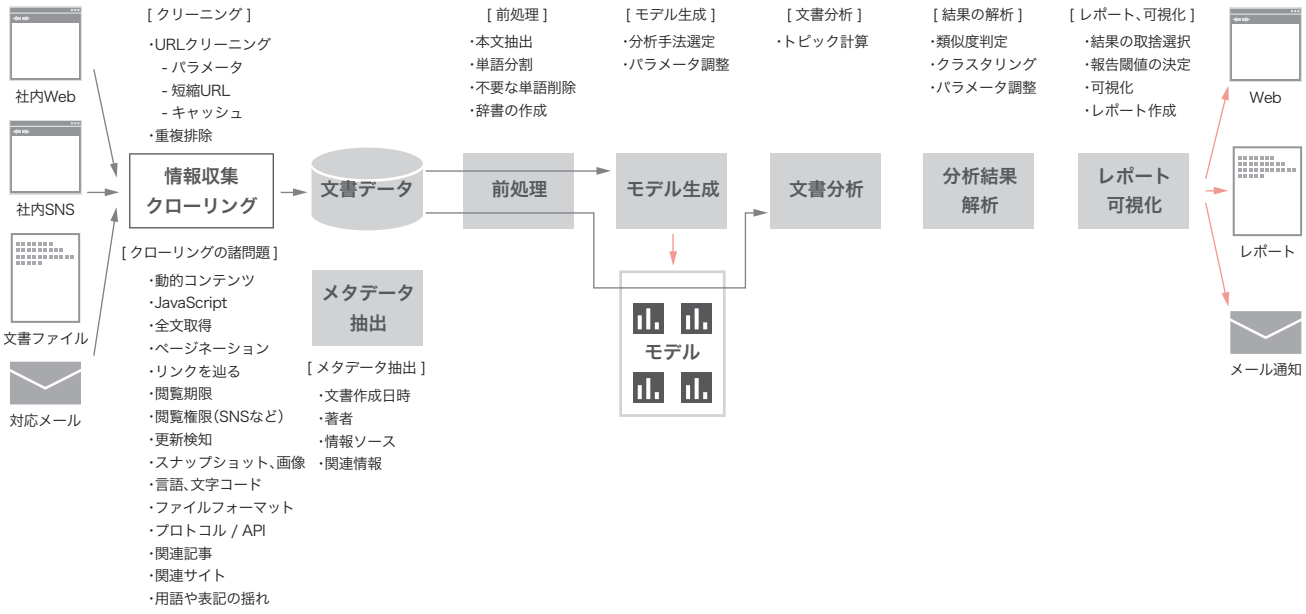


図-1 トピックモデルを利用した自然言語処理の全体像

*1 例:word - wordsやwrite - wrote - written。

では単語を分ける明確な区切りがないので、形態素解析と呼ばれる処理を使うことになるでしょう。

3. 「不要」な単語の削除
   例えばどの文書にも頻繁に登場する「です／ます」などは文章の内容にはあまり関係ないと考えられます。こうした、目的に対して役立ちそうにもない単語でデータが大きく膨らんだところで分析精度が向上するとは考えにくいので、前処理の段階で削除しておくのが常套手段になっています。
   削除対象とする単語をあらかじめ決めておくストップワードと呼ばれる方式のほか、入力テキスト中で登場回数の多い単語を削除する方式、ごく稀にしか使われない単語も削除する方式などが使われています。どのような単語を削除すると効果的なのかは、先例にならいながら目的に応じて試行錯誤することになります。
4. 文書ごとの単語別頻度表を作成
   文書内の単語の登場回数を数え上げます。このような単語の出現頻度表をBag of Wordsと呼び、それぞれの文書に1つずつ頻度表を作成します。LDAに実際に渡すのはこれらの頻度表のリストです。

この手順でお気づきのとおり、Bag of Wordsに単語の登場順序は反映されないので、前後関係や文脈といった情報はLDAでは考慮されないことが分かります。

## 2.4 プロトタイプ作成

これらの技術を検証するために、実際にプロトタイプを作成してみました。本稿では米国の非営利団体MITRE社[*2]が公開している脆弱性情報データベースCVE[*3]のsummaryを使用することにします。英文の自然言語処理に関する技術は多くがライブラリ化されているので、それを使用するだけで多くのことができます。

CVE-2018-5383

Bluetooth firmware or operating system software drivers in macOS versions before 10.13, High Sierra and iOS versions before 11.4, and Android versions before the 2018-06-05 patch may not sufficiently validate elliptic curve parameters used to generate public keys during a Diffie-Hellman key exchange, which may allow a remote attacker to obtain the encryption key used by the device.

**図-2 CVE IDごとにサマリ文書を保存したテキストファイルの例**

まずは元となるデータを作成しましょう。今回は2018年にリリースされたCVE 7,692件を処理対象にします。米NISTが提供しているNVD Data Feeds[*4]のページから、CVEのデータをダウンロードします。JSON形式のものもベータ版として提供されていましたが、今回は以前に扱ったことがあるXML形式のデータを取得しました。このデータにはCVE IDや公開日時、最終更新日時、関連情報の参照先などが、計算機でも扱いやすい形に構造化されて含まれていますが、今回はこの中からあえて非定型の自然言語記述（vuln:summary）だけを取り出して、CVE IDごとに別のファイルに保存します（図-2）。これで元となるテキストファイルが準備できました。

### 2.4.1 文章の前処理

LDAモデル作成に必要な一連の処理ではPythonのgensim[*5]やnltk[*6]というライブラリを使用します。

まずは元となる文書にLDAモデルを作成するために必要な前処理を施します。今回ターゲットとしたCVEのデータには含まれていませんでしたが、例えばWebサイトに掲載された英文に対する定番の処理として、

- HTMLタグの除去やHTML特殊文字の処理
- 行末でハイフネーションされた単語の接続

などがあります。

次に単語分割（tokenize）、lemmatize、ストップワード除去と呼ばれる一連の処理をして文書から単語データを生成します。lemmatizeは関数に文字列を渡すだけですが、内部では名詞、動詞、形容詞、副詞だけを抽出し活用形を原型に揃えるなど、非常に多くの複雑な処理をしています。

＊2　MITRE社（https://www.mitre.org/）。
＊3　CVE（https://cve.mitre.org/）。
＊4　NVD Data Feeds（https://nvd.nist.gov/vuln/data-feeds）。
＊5　gensim（https://radimrehurek.com/gensim/）。
＊6　nltk（http://www.nltk.org/）。

ここまでの処理を対象とするそれぞれの文書に適用して、取り出した単語データのリストを作成します。

```
def normalize(txt):
    # De-hyphenation of words across a line-break(行末のハイフンを除去)
    txt = re.sub(r'-\n', '', txt)
    # Concatenate lines(改行を除去)
    txt = re.sub(r'\n', ' ', txt)
    # Tokenization and lemmatization (単語分割と lemmatize)
    tokens = [ re.sub(r'/[A-Z]+$', '', x.decode('utf-8'))
                for x in gensim.utils.lemmatize(txt) ]
    # Remove stop-words(ストップワードを除去)
    stopwords = nltk.corpus.stopwords.words('english')
    tokens = [ token for token in tokens if token not in stopwords ]
    return tokens

docs = []
for path in files:
    with open(path, encoding='utf-8') as f:
        txt = ''.join(f.readlines())
    tokens = normalize(txt)
    docs.append(tokens)
```

normalize関数中で使用しているgensim.utils.lemmatize()関数は、単語の後ろに品詞の情報をつけて返すという仕様のため、正規表現でそれを取り除いています。

```
[b'cve/VB',
 b'high/JJ',
 b'rate/NN',
 b'vlan/NN',
 ...
```

### 2.4.2 辞書とBag of Wordsを作成

今回使用したgensimライブラリでは、単語にIDを振って扱います。作成した単語データのリストを元に単語にIDを割り振り、それぞれの文書に含まれている単語を数え上げます。このデータ構造のことを辞書と呼びます。

その後、その辞書の内容を解析して整理します。具体的には、

- 共通して出現することが多い単語
  (例:20%以上の文書に含まれている単語)
- 滅多に出現しない単語
  (例:1つの文書にしか出現しない単語)

などを取り除くのが定式となっているようです。

この処理を省いたりパラメータを変更する実験をしてみたところ、共通して出現する単語を除去しないで作ったモデルで、分類の精度が落ちる傾向が見てとれました。滅多に出現しない単語を取り除く処理に関しては、実験した範囲では目に見える程の効果は確認できませんでした。

整理した辞書を使用して、文書ごとにBag of Words(BoW)ベクトルと呼ばれるものを作成します。これは辞書に含まれる単語が文書中に出現する回数をベクトルで表したものです。

```
dic = gensim.corpora.Dictionary(docs)
dic.filter_extremes(no_above=0.2, no_below=1)
bow = [ dic.doc2bow(doc) for doc in docs ]
```

### 2.4.3 LDAモデルを生成

辞書とBag of WordsからLDAモデルを生成して、モデルとここまで作った関連データをファイルに保存します。

LDAモデルの生成時にトピック数を指定する必要があります。このトピック数ですが、元となる文書や出現する単語の量、偏り方などによって適正な値が変わるようで、その決め方にもいろいろな流儀があるようです。対象とする文書や数を変更して実験を繰り返してみたところ、手元では30～50の値を指定したときに比較的良い結果が見えるモデルが生成できたことが多かったため、ここでは50を指定しました。

```
lda = gensim.models.ldamodel.LdaModel(bow, id2word=dic, num_topics=50)

lda.save(filename_model)
dic.save(filename_dic)
gensim.corpora.MmCorpus.serialize(filename_corpus, bow)
```

## 2.5 生成したモデルを使用して文書を分析

このモデルを使って、さっそく対象となる文書を分析してみます。それぞれの文書がどのようなトピックを含んでいるか、結果を得ることができました。

```
results = []
for doc in docs:
    bow = lda.id2word.doc2bow(doc)
    doc_topics = lda.get_document_topics(bow)
    results.append(doc_topics)
```

### 2.5.1 類似した文書をピックアップする

文書ごとに持つトピックの分析結果を使って、ベクトル間のコサイン類似度を計算することで、すべての文書の中からトピック成分の近い文書を選び出すことができます。そこで、最近

ニュースなどで話題になったCVEをいくつか選んで、それと類似したものを選び出すことができるか検証してみました。

例えば8月のMicrosoft月例パッチで修正されたCVE-2018-8373（図-3）を調べると、同様の「Scripting Engine Memory Corruption Vulnerability」に関するCVE（CVE-2018-0955、CVE-2018-0996、CVE-2018-1001、CVE-2018-8267など）がずらっとリストアップされました。

```
[('CVE-2018-0955', 1.0),
 ('CVE-2018-0988', 1.0),
 ('CVE-2018-1001', 1.0),
 ('CVE-2018-8267', 1.0),
 ('CVE-2018-8353', 1.0),
 ('CVE-2018-8371', 1.0),
 ('CVE-2018-8373', 1.0),
 ('CVE-2018-8389', 1.0),
 ('CVE-2018-0996', 0.9999999),
 ('CVE-2018-8242', 0.9999997),
 ('CVE-2018-0839', 0.9968462),
 ...
 ('CVE-2018-8385', 0.9579928),
 ...
 ('CVE-2018-8372', 0.8273082),
 ('CVE-2018-8355', 0.8272891),
 ...
 ('CVE-2018-8359', 0.7355896),
```

類似判定された文書をピックアップして読んでみると、確かにほぼ定型文と言っていいほどよく似ているものが多く見られます。

今回のCVE-2018-8373サマリの中に出現している関連CVEがどの程度類似しているかを調べると、0.73 〜1.0の範囲に出現していました。しかし、この範囲のコサイン類似度を持つCVEは178件あります。類似した文書をピックアップすること自体はできているとも言えるかもしれませんが、ユーザが見たいものを提示するためにはこの類似度をどう扱うべきか、もうひと工夫必要そうです。

続いて、Foreshadow-NGなどとして知られるCVE-2018-3620（図-4）「CPUの投機的実行機能に対するサイドチャネル攻撃」と類似したCVEを計算してみると、関連するCVE-2018-3646、CVE-2018-3615などを含むリストを作成できました。しかし、それらと似たようなコサイン類似度でPHP製のチャットプログラムやWebアプリケーションフレームワークの脆弱性もリストに入っています。トピックのコサイン類似度だけを閾値として判断するとあまりよい結果が得られない可能性があります。

```
[('CVE-2018-3620', 0.9999924),
 ('CVE-2018-3646', 0.9803927),
 ('CVE-2018-3640', 0.88989854),
 ('CVE-2018-3693', 0.8448397),
 ('CVE-2018-3615', 0.8389072),
 ('CVE-2018-5954', 0.8318751),
 ('CVE-2018-1000181', 0.80068177),
 ...
```

このようにして、トピックモデルとコサイン類似度を用いてCVEのsummaryからそれと類似するCVEを選び出した場合、似た傾向のものをある程度ピックアップできることが確かめられました。

一方で、あまり似ていなさそうな内容の文書までまとめて選び出してしまうケースが多く見られました。また、期待する文書が選び出されていないケースもありました。

CVE-2018-5390（図-5）はLinux kernelのTCP実装にDoS脆弱性があったものですが、上位に類似判定された文書にはそのような内容の文書は選び出されていませんでした。例えば、Linux kernelの脆弱性は対象としたCVEに105件含まれていましたが、それらの類似度は高くありませんでした。

```
[('CVE-2018-5390', 0.99944544),
 ('CVE-2018-1237', 0.81856203),
 ('CVE-2018-1240', 0.8044424),
 ('CVE-2018-1217', 0.80138516),
 ...
```

A remote code execution vulnerability exists in the way that the scripting engine handles objects in memory in Internet Explorer, aka "Scripting Engine Memory Corruption Vulnerability." This affects Internet Explorer 9, Internet Explorer 11, Internet Explorer 10. This CVE ID is unique from CVE-2018-8353, CVE-2018-8355, CVE-2018-8359, CVE-2018-8371, CVE-2018-8372, CVE-2018-8385, CVE-2018-8389, CVE-2018-8390.

**図-3 CVE-2018-8373 のサマリ**

Systems with microprocessors utilizing speculative execution and address translations may allow unauthorized disclosure of information residing in the L1 data cache to an attacker with local user access via a terminal page fault and a side-channel analysis.

**図-4 CVE-2018-3620 のサマリ**

Linux kernel versions 4.9+ can be forced to make very expensive calls to tcp_collapse_ofo_queue() and tcp_prune_ofo_queue() for every incoming packet which can lead to a denial of service.

**図-5 CVE-2018-5390 のサマリ**

これらの結果に影響を与える要素としては、コサイン類似度の閾値をどれくらいに置くかということ以外に、モデル生成時に各段階で与えることができるパラメータがあります。本稿で例示した以外にも実験を繰り返した結果、これらのさじ加減によって結果が大きく変わることがありました。それぞれのパラメータ調整をすることが、結果をつぶさに見る前には難しいという、トピックモデルの扱いづらさも垣間見ることができました。

### 2.5.2 周辺情報の活用などによる精度向上

トピックモデルは、非定型の自然言語で記述された文書群から類似したものを手早く選び出す手法として有力なものの1つです。一方で、読み手が期待する文書を適切にピックアップできるかで出力結果を評価するならば、その精度を上げていくためには用途に応じた工夫が必要です。なぜなら、読み手が期待する「類似文書」は場面によって異なると考えられるからです。

例えば、日々のニュースにおいて類似の記事をまとめたい場合は直近の情報以外は邪魔になると考えられます。しかし、発生の稀な問題に対応するため関係する文書を調べているときは、読み手は古い情報まで含めて類似の文書を探し出したいのではないでしょうか。

トピックモデルの学術研究ではこうしたニーズの多様性に応えようと、モデルの構造を工夫して、特定用途での精度向上を狙った研究も進められています。例えばトピックの時系列変化を分析に採り入れたモデルや著者名を分析に採り入れたモデルなどがあります。

今回は、CVEやセキュリティ関連の文書をターゲットに、単純に周辺情報を活用したフィルタリングを実験してみました。例えば、社内の内部向けメモのようなものは、作成日時の前後に作られたものを優先することで、ユーザが読みたいものを採り上げる精度を上げることができました。社内の文書をターゲットにした場合、プロジェクト名や文書の置かれているパス名に含まれているキーワードでの絞り込みも有効です。

もっとシチュエーションに依存しない絞込み方法で精度向上できないかを試してみたところ、DBSCANなどのPythonのscikit-learn[*7]ライブラリに含まれるクラスタリングアルゴリズムを使い、トピックのクラスタリングを行った結果をもとにリストをフィルタリングすることで良好な振り分けができる場合があることが確認できました。しかし、この手法もクラスタリングを行う際のパラメータ設定で大きくクラスタの様子が変わるためチューニングが必須で、シチュエーションに依存せず安定的に使うことは難しそうです。

今回、非定型の情報をうまく扱う手段を探して、自然言語処理とトピックモデルを中心にいろいろと実験してみました。1つの手法だけで満足のいく結果を得られるものはありませんでしたが、目的に応じて複数の手段を組み合わせることで、望みの出力に近づける工夫はできそうなことが分かりました。これらの知見から、一定の条件下で非定型の文書を取り扱う場面への適用を考えていきます。

執筆者:
**永尾 禎啓**（ながお ただあき）
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。
1998年4月入社。セキュリティサービス開発やSDN開発などを経て、現在は理論的視点から情報セキュリティ全般の調査活動に従事。
IIJグループの緊急対応チーム、IIJ-SECT メンバー。

執筆者:
**桃井 康成**（ももい やすなり）
IIJ セキュリティ本部 セキュリティ情報統括室 リードエンジニア。
1999年1月入社。セキュリティサービスや無線ICタグ関連システムなど各種サービスの研究開発を経て、現在は情報セキュリティ全般に関わる調査研究活動に従事。IIJ-SECTメンバーとして、日本セキュリティオペレーション事業者協議会(ISOG-J)、ICT-ISACなどの活動や運営に参加。

*7　scikit-learn(http://scikit-learn.org/)。

# クラウドとKubernetes

## 3.1 はじめに

クラウド関連の情報をウォッチしていると、毎日のようにコンテナ技術にまつわるニュースが流れていることにお気付きでしょうか。キーワードとしてdockerやKubernetes(クバネテスと発音されることが多い)[*1]、CNCFなどを含むニュースであれば、それに類する情報であると考えて間違いありません。更に、今ではdocker、Kubernetesを基盤とするさまざまなプロダクトが生み出され、エコシステムが拡大を続けているため、一見してそれと気付かなくても実はコンテナ系技術の話題であることも珍しくありません。

今後しばらくはメガクラウドベンダを中心にクラウド業界のルールが整えられていくであろうことはある程度否定できませんが、コンテナ技術の興隆を見るにつけ、思ったよりも早く次のトレンドが業界のルールを塗り替えていくのではと思わずにはいられません。もしかしたら、IaaSの登場よりももっと大きなインパクトをIT業界へ与えるイベントが進行しているのかもしれません。

IIJにおいてもこの新しい技術を、ビジネスの迅速な展開や大規模システムの効率的で高品質な運用、ポータビリティの高いソフトウェアの開発、インフラの効率的な利用によるコスト最適化など、幅広く生かすべく活用を始めています。社内で利用されているIKE(IIJ Container Engine for Kubernetes)についてはまた後で触れますが、まずはなぜこんなにも急激にコンテナ技術が注目を集めるようになったのか、そしてこの技術がクラウドへどのような影響を与えると考えられるのかを解説します。

## 3.2 dockerとKubernetes

コンテナ技術を取り巻くサービスやプロダクトは雨後の竹の子のごとく登場していますが、その中心にあるのはたった2つのプロダクト、dockerとKubernetesです。日進月歩のコンテナ界隈ですが、この2つのプロダクトの動向をキャッチアップすることで、主要な流れを見極めることができるでしょう。

両者の関係は少々複雑ですが、なるべく単純化して表現するとdockerはプログラムをコンテナにくるんで起動するコンテナエンジン、Kubernetesは複数のコンテナエンジンを束ねて制御するコンテナオーケストレータです。一般的にオーケストレータとは連携する複数システムを協調させ、全体を1つの系として統合させるコントローラを示す言葉ですが、Kubernetesのようなコンテナオーケストレータとはコンテナエンジン(≒docker)が稼働する多数のホストノードを束ねて、大きなひとつのリソースプールとして効率的かつ自律的に制御することを意味しています。製品としてのdockerはKubernetesと競合するところもあるのですが、ひとまずコンテナエンジンとコンテナオーケストレータの関係と理解した方が混乱はないでしょう(図-1)。

とはいえ、常に両者がペアで使われるわけではありません。docker単体での利便性は既に広く認知され大いに活用されているものの、Kubernetesがそこまで実環境で利用されているとは言えません。それは、Kubernetesがそれなりの規模で構成されたコンテナクラスタを制御するプラットフォームであるため、試すにしても一定のハードルがあるのに対して、dockerは手元の作業環境を便利にしてくれるユーティリティ

---

*1 dockerに代表されるコンテナエンジンを制御し、多数のノードから構成されるコンテナクラスタを管理するコンテナオーケストレータの1つ。googleによって生み出され、現在はCNCF(Cloud Native Computing Foundation)にホストされるオープンソースソフトウェアである。googleの社内システムであるborgをベースに開発されたとされる。コンテナ化されたアプリケーションのためのランタイム環境であり、インフラに依存しないポータブルなアプリケーションのパッケージング、プロビジョニング、オペレーションが可能になる。クラウド時代のOSと称されることもあり、マルチクラウド、ハイブリッドクラウドのための統一されたオペレーションインタフェースとしても期待される。

Kubernetesを利用するにはインフラに合わせたネットワークドライバやストレージドライバ、トラフィックマネージャなどが必要になるが、Kubernetesにそのような実装は基本的に含まれていない。またKubernetesを環境として整えるには、アプリケーション管理やアカウント管理を行うポータルやモニタリングツールなども事実上必須となるが、そのようなツール類も別途そろえる必要がある。そのため、Kubernetesに周辺の環境を加え、インストーラなどを整備したパッケージであるKubernetesディストリビューションが登場しつつある。本稿で触れているIKE(IIJ Container Engine for Kubernetes)もそうしたKubernetesディストリビューションの1つである。

でもあるので、ちょっとしたユースケースでも分かりやすく便利だからです。おそらく多くのエンジニアがdockerをテスト環境を整えるツールとして、またソフトウェアの配布手段として、便利に利用していることでしょう。dockerは既にエンジニア必携のツールとなりつつあるのです。

ですが、今IT業界を賑わしているのは、どちらかと言えばKubernetesの方でしょう。それは、コンテナ技術がもたらす効果は便利なユーティリティにとどまるものではなく、Kubernetesを活用することでサーバサイドシステムの在り方を大きく変えることが期待されているからです。

まだ成熟しているとは言い難いKubernetesがそこまで注目を浴びているのは、Kubernetesのオリジナルが10年以上に渡ってgoogleのシステムを支え続けているborgだということでしょう。googleの社内システムが詳細に語られることはほとんどありませんが、borgがどのように利用されているのかはSRE(Site Reliability Engineer)Bookを通して一部が明らかにされ、その驚くべき実態が垣間見えるようになりました。これがきっかけとなりコンテナ技術に注目するようになった人も少なくないでしょう。googleのシステムに仮想マシンはなく、基本的にすべてのプロセスがコンテナとして起動されている話は、特にクラウドビジネスに携わるエンジニアに大きなインパクトを与えました。KubernetesはborgのOSS版であるとされていますが、実際にどれほどの共通性があるのかは分かりません。しかし、まだ登場して日が浅く、実績に乏しいと感じられることもあるKubernetesも、そのデザインにはgoogleにおいて長期間利用され、熟成されてきたベストプラクティスが詰め込まれているであろうことは想像できます。

## 3.3 IaaSを使いこなすための ベストプラクティス

それでは、Kubernetesが変えるサーバサイドシステムの在り方とは何でしょう。曰く、ポータビリティが高く、インフラに依存しないデプロイメントが可能になる。曰く、大規模なクラスタの管理能力を備え、コンピューティングリソースをダイナミックに活用するスケーラビリティに優れる。様々な言葉で表現されますが、曖昧で具体性に欠けるように感じられないでしょうか。それはある意味仕方のないことで、Kubernetesの役割はコンピュータを管理するOSのようなものなのです。OSが何かご存じない方にOSとは何かを説明すれば、やはりとりとめがなく断片的な話になるか、極めてテクニカルなディテールの話になるかどちらかでしょう。

実際のところ、Kubernetesはクラウド時代のOSに例えられることがあります。インフラの差異をドライバで吸収し、ネットワークやストレージのコンフィグレーションを仮想化し、Kubernetes上にデプロイされるシステムへ統一されたインタフェースを提供します。Kubernetesを利用することでIaaS固有のインタフェースに頼らず、統一されたシステム設計、構

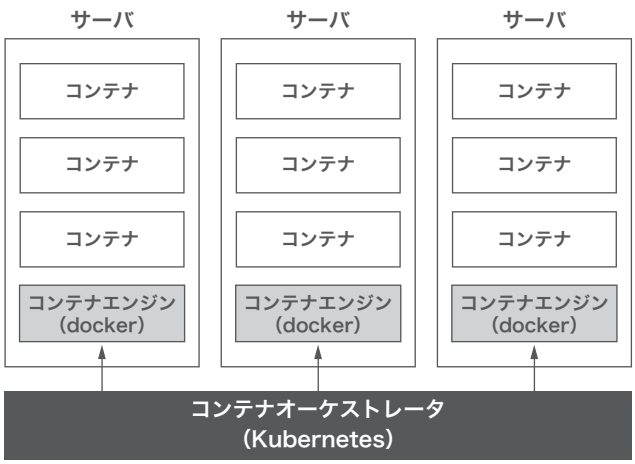


図-1 コンテナエンジンとコンテナオーケストレータ

築、運用が可能になるのです。ただ、KubernetesはOSではないので、アプリケーションインタフェースを提供するわけではありません。Kubernetes上で動くのはあくまでも通常のLinuxやWindowsのアプリケーションです。

似て非なる存在であるOSとKubernetesですが、大きく違うのは前者が物理的な箱に収まった1台のコンピュータを管理するのに対して、後者はネットワークで接続された複数のコンピュータを大きな1つのリソースプールとみなして管理することです。OSによって管理されるプロセスは箱の外へ出ることができませんが、Kubernetesクラスタで動くプロセス(≒コンテナ)はクラスタを構成するどこかのノードで動いてさえいれば問題ありません。だから、リソースが不足すればノードを増強してコンテナを収容替えするだけで済みますし(これは自動的に行われます)、あるノードに障害が発生してコンテナが停止しても、他のノードでコンテナを再起動するだけで復旧します(これも自動で行われます)。

多くの場合、クラウドサービスは止まらない安定性に優れたシステムとみなされています。しかし、実際にはクラウドサービスも様々で、特にIaaSのコンピューティングリソースは冗長化されているわけではありませんから、障害があれば停止しますし、メンテナンスのために計画的に停止されることもあります。IaaSが普及したことでシステムリソースの調達と構築、それにハードウェア障害からの復旧は劇的に短時間で可能になりましたが、多くの運用に関してはさほど変化がありませんでした。仮想化されていようがいまいが、扱う対象が結局はサーバ、ストレージ、ネットワークであれば、オペレーションは大きくは変わりません。IaaSを利用するのは難しくありませんが、そのメリットとされるユーティリティコンピューティングの特性を生かすには、多くの労力を必要とするのが現実です。

そこでKubernetesの出番です。必要なときに必要なだけリソースを確保し、使っただけのコストが発生するIaaSの特性はKubernetesと親和性が高く、動的にリソースを管理することでリソースを無駄なく使ったり、スケーラビリティを生かしたりすることが容易です。また、複数ノードを組み合わせて可用性を保つだけでなく、障害を起こしたノードの復旧をKubernetesへ任せることで、システム全体として影響がない範囲の障害であれば人手を介さずに自動で復旧させることが可能です(図-2)。

コンテナを利用すると従来利用されていたハイパーバイザや仮想マシンといった分厚い管理層が無くなり、単一のOS上にコンテナ化されたプロセスが直接乗るため管理層が薄く、軽くなると説明されることが多いのですが、ただそれだけではコンテナ技術の効果は極めて限定的です。仮想マシンよりもコンテナの方が多くの場合効率的なのは事実ですが、それは手段の話でしかありません。多数のサーバを束ねて大きなリソー

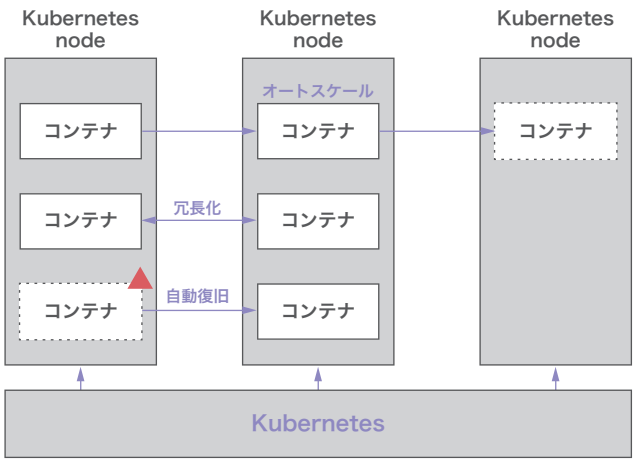


図-2 IaaSを生かすKubernetes

スプールとして構成し、構成情報の管理をKubernetesに任せて自動化することで、初めて本来の効果が発揮されるのです。実際のところ、現時点ではおそらく多くのコンテナが仮想マシンとして実装されているIaaS上で動いていることでしょう。Kubernetesとは、IaaSをより良く使いこなすためのベストプラクティスがつまったパッケージと考えて良いでしょう。

## 3.4 ハイブリッドクラウドを実現するKubernetes

KubernetesがIaaSの利点をより良く使いこなすためのプラットフォームであるのは事実ですが、IaaS上でしか使えないわけではありません。それどころか、今後を考えるむしろオンプレミスな環境が重視されるワークロードでこそKubernetesに注目すべきです。これまで10年近くに渡りお客様の声を聴き、クラウド関連のレポートを見てきましたが、クラウドが広く普及し、100%オンプレは現実的ではないと考えられるようになっても、100%クラウド化もやはり難しいと回答される方が大多数を占めています。多くのエンジニアがIaaSを手足のように使いこなすようになった今このような意見が多数を占めるということは、今後の在り方が見えてきたのかもしれません。

となると、IaaSとオンプレ環境を適切に使い分ける、ハイブリッドクラウドを本気で考える必要があるということですが、言うまでもなくそれは簡単なことではありません。IaaSはクローズドなシステムですから、オンプレ環境で使うことはできません。仮に使うことができても複雑なIaaSシステムの運用をオンプレ環境で行っては、もはやなんのためにIaaSを利用しているのか分からず、本末転倒な事態になりかねません。プライベートな環境にIaaSと同等の環境を持つにふさわしいワークロードももちろん存在しますが、多くのトレードオフを考慮する必要はあるでしょう。

ですが、KubernetesでIaaSとオンプレ環境の両方をラップすることで、それが現実的な解となるかもしれません。ハイブリッドクラウドを実現するには大きく2つの課題があります。IaaSとオンプレの統一的な管理運用システムとアプリケーションやデータのポータビリティです(図-3)。

コンプライアンス上の理由からIaaSには出せないワークロードやデータ、それにリソースの増減なく長期的に一定の大規模リソースを利用するワークロードはオンプレ環境に置きたいものです。そして、それ以外のワークロードはIaaSに置きたいと考えるかもしれません。ある程度最初からどちらが適しているかはっきりしていれば問題ないのですが、ビジネスの立ち上げ時期はIaaSに、安定時期にはオンプレにといった具合に、時期によって適した環境が変わっていくことも珍しくはありません。前述した2つの課題をKubernetesが効果的に解決することができれば、一気にこの市場が広がる可能性を秘めています。

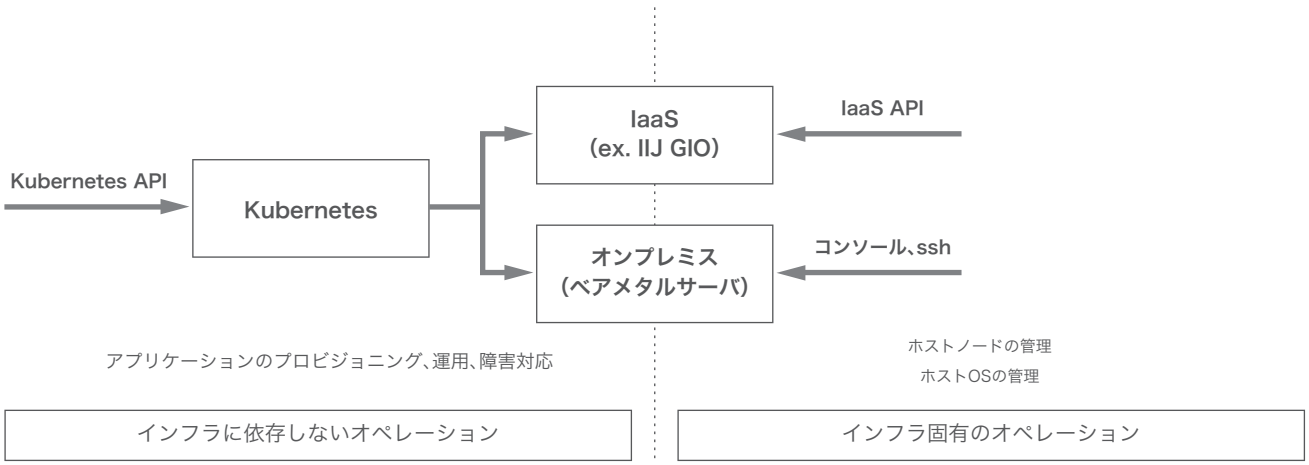


図-3 ハイブリッドクラウドを実現するKubernetse

ただし、現時点ではKubernetesを利用したハイブリッドクラウドはもちろん、オンプレでKubernetesを活用する環境は、IaaS上のそれに比べてまだ成熟しているとは言い難い状態です。あらゆるリソースをAPIを通じてソフトウェアで制御できるIaaSに比べて、オンプレ環境は統一的なソフトウェア制御ができませんから、環境を整えていくにはKubernetesとデバイス双方の歩み寄りと協力が必要でしょう。また、SDN(Software Defined Network)、SDS(Software Defined Storage)といった技術がオンプレ環境で使いやすくなると、オンプレ環境におけるKubernetesの普及に弾みがつきそうです。

ハイブリッドクラウドを実現するには、IaaSとオンプレを管理する共通システムが必要です。Kubernetesがその有力候補であることは間違いありません。

## 3.5 IKE(IIJ Container Engine for Kubernetes)

このように今後のサーバサイドシステムの設計や運用、アプリケーションの流通とプロビジョニングを根底から変え、劇的に効率を向上させる可能性を秘めたKubernetesですが、IIJにおいてもコンテナクラスタシステムが開発され、その活用は始まっています。IKE(IIJ Container Engine for Kubernetes)と名付けられたこのシステムは、サービスの共通基盤として、また社内システムの運用環境として生み出されたものです。

IKEのようにKubernetesを中心としてコンテナクラスタの環境を整えるパッケージをKubernetesディストリビューションと呼びます。先にKubernetesをOSに例えましたが、KubernetesはOSのカーネルのような存在です。OSがカーネルだけでは何もできず、各種ツールやデバイスドライバを整えたディストリビューション(例:RedHat、Ubuntu)があって初めて役に立つように、Kubernetesをただインストールしても何もできません。インフラに応じたネットワークドライバとストレージドライバが最低限必要ですし、より快適なコンテナクラスタを実現するにはKubernetesを制御するマネージメントツールと、Kubernetes上にデプロイされたアプリケーションをモニタリングしたり、アラートを通知したり、ログを収集したり、運用を支援する環境の整備が欠かせません。Kubernetesクラスタを構成するエコシステムを整え、そうした環境を指定されたインフラに合わせてインストールする機能を備えたパッケージがKubernetesディストリビューションであり、IKEはその1つというわけです。

IKEはお客様へのサービス提供を目的としたものではないため、ある程度動作環境を限定して設計されていますが、自社クラウドサービスであるIIJ GIOはもちろんのこと、オンプレ環境へのインストールが可能です。また、今後は他社IaaS上にもインストールできるようにして、どのようなインフラであっても共通の環境を提供できるようにする計画です。

IIJがKubernetesディストリビューションを実装した理由はいくつかありますが、IaaSの活用だけが目的ではありません。一番の目的はビジネスのスピードアップ、二番目が運用の専門性を高めて高度化、複雑化するシステムに対応することです。こうしてまとめてしまうと抽象的で目的が曖昧に思えるでしょうが、実現したいのは例えばサービスを開発する部

門は開発に、運用部門は運用に専念できる環境を整えることです。これを実現できれば前述の目的はおのずと達成されていくことでしょう(図-4)。

それにしても、仮想マシンのごとく1つのコンテナにサーバ環境をすべて詰め込むのではなく、プロセスを1つずつコンテナに包むようにしただけで、インフラに依存しないサーバサイドシステムのディストリビューションと汎用的な運用システムが実現してしまい、IT業界全体へ影響を及ぼすようなプロダクトが誕生するのですから、この世界は本当に面白いものです。おそらくこれはまだ始まりにすぎず、これからも思いもよらぬアイディアが登場するのだろうと思うとワクワクしますね。

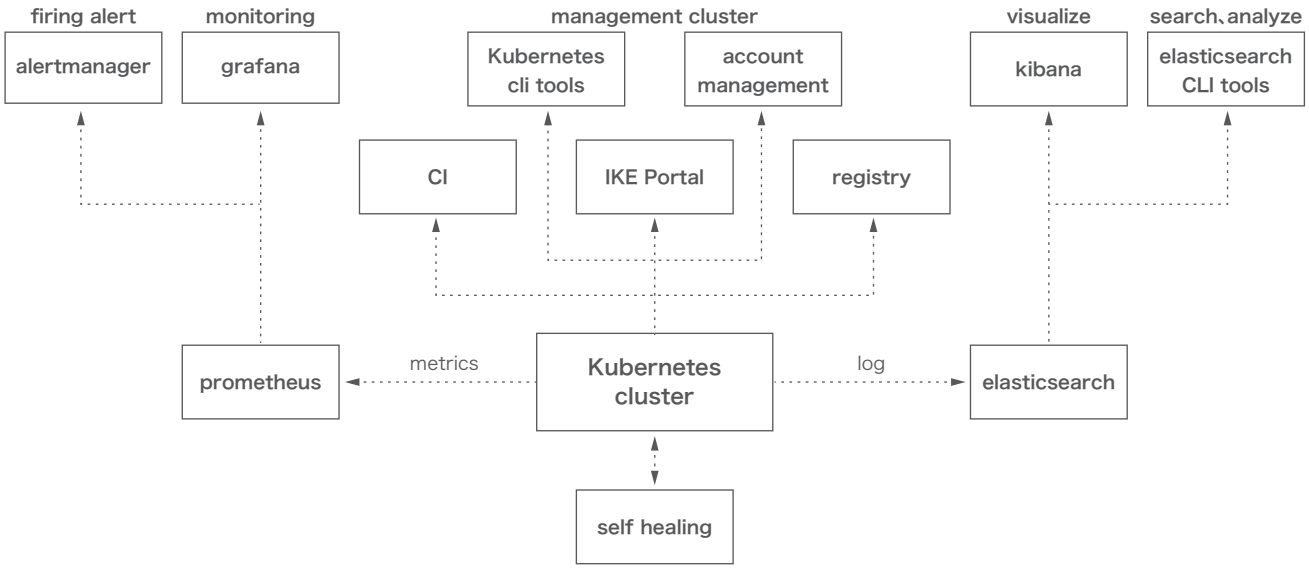


図-4 IKE

執筆者:
**田口 景介**(たぐち けいすけ)
IIJ サービス統括本部 技術戦略室。
社会人生活の半分をフリーランス、半分をIIJで過ごすエンジニア。元々はアプリケーション屋だったはずが、クラウドと出会ったばかりに半身をインフラ屋に売り渡す羽目に。現在はコンテナ技術に傾倒中だが語りだすと長いので割愛。タグをつけるならコンテナ、クラウド、ロードバイク、うどん。

Information

# IIJ 技術情報発信コンテンツの紹介

**【ポータル】**

**■ IIJの技術 https://www.iij.ad.jp/dev/**

IIJのエンジニアが発信する様々な技術情報のポータルです。新着情報を随時ご紹介します。


**【ブログ】**

**■ wizSafe Security Signal（https://wizsafe.iij.ad.jp/）**

35号まで本誌で定期的に掲載しておりましたインターネットセキュリティの観測レポートをはじめ、新たな攻撃手法や脆弱性などにより被害が懸念される緊急度の高い情報をSOCからタイムリーに発信しています。

[ 最近の掲載記事 ]

2018.08.31「wizSafe Security Signal 2018年7月 観測レポート」
2018.07.31「wizSafe Security Signal 2018年6月 観測レポート」
2018.06.29「wizSafe Security Signal 2018年5月 観測レポート」
2018.05.31「wizSafe Security Signal 2018年4月 観測レポート」


**■ IIJ Engineers Blog（http://eng-blog.iij.ad.jp/）**

開発・運用の現場エンジニアが執筆する公式ブログです。2016年に開設して以来、これまでに30名以上のエンジニアが、インターネットに関して業務で取り組んだことから個人的趣味で実践していることまで、興味のある分野を様々な角度から自由に執筆しています。

[ 最近の掲載記事 ]

2018.09.10「素人がトピックモデルを試してみた（第2回）」
2018.08.28「Black Hat USA でトレーニング講師になってみた」
2018.08.02「【JANOG42】今回のJANOGはモバイルで！
　　【初参戦レポート】」
2018.07.25「Let's EncryptとACME」

**■ スマートメーターBルートブログ(http://route-b.iij.ad.jp/)**
IIJのスマートメーターBルートの公式ブログです。2016年に開設し、IoTや電力メーターのBルートに関連する技術解説・業界動向・サービスやイベントの情報など、様々な情報をお伝えしています。

[ 最近の掲載記事 ]
2018.09.10「卒FITに向けてLEAF導入を検討してみる(1)」
2018.08.28「新製品紹介 NextDrive Beep 外観編」
2018.08.02「電柱でIoT ！」
2018.07.25「ネコリコオフィス紹介！」


**■ てくろぐ(http://techlog.iij.ad.jp/)**
技術広報担当の堂前が2010年から執筆しているIIJの元祖エンジニアブログです。インターネット技術の話題はもちろん、個人向けサービス「IIJmio」で提供しているSIMの使い方やスマホの動作確認情報まで、みんなが知りたい・気になっている情報をわかりやすく紹介しています。

[ 最近の掲載記事 ]
2018.08.24「インターネット・トリビア：プログラム中の名前の付け方」
2018.07.14「インターネットと通信の秘密(IIJmio meeting 20資料公開)」
2018.06.15「20回記念！IIJmio meeting 参加者募集中」
2018.06.05「インターネット・トリビア：コンピュータでの数字の並べ方」


**■ IIJ-SECT Security Diary(https://sect.iij.ad.jp/)**
IIJの緊急対応チームIIJ-SECT(IIJ group Security Coordination Team)によるブログです。国内外で発生したセキュリティインシデントについての調査結果・考察を掲載しています。

**■ GIOろぐ(https://giolog.iij.ad.jp/)**
IIJのクラウドサービス「IIJ GIO」に関するサービスやイベントのお知らせ、クラウドの技術的な背景などを掲載しています。

**各種ブログを更新した際には、以下のTwitter/Facebookアカウントにてお知らせを投稿しています。気になる情報がありましたら、ぜひこちらもフォローしてみてください。**

**【Twitter】**

**■ @IIJSECT**
IIJのセキュリティコーディネーションチームによる調査報告

**■ @IIJ_doumae**
IIJエンジニアの堂前が興味のある技術ネタについてつぶやきます

**■ @SEIL_SMF**
IIJが開発するルータ「SEIL(ザイル)」の情報やファームウェアの更新のお知らせなど

**■ @IIJGIO**
IIJ GIOの最新情報やイベント・セミナーの情報

**■ @IIJ_PR**
プレスリリース、イベント情報

**【Facebook】**

**■ IIJ公式ファンページ(http://www.facebook.com/IIJPR)**
IIJの公式ファンページです。プレスリリースやお知らせ、技術・開発情報、イベント・セミナー情報など、IIJに関する様々な情報をお届けします。

**■ IIJ GIO公式ファンページ(http://www.facebook.com/IIJGIO)**
IIJのクラウドサービス「IIJ GIO(ジオ)」の公式ファンページです。IIJ GIOの最新情報をお届けすると共に、ファン限定のコンテンツやイベントも予定しています。

**■ SEIL公式ファンページ(http://www.facebook.com/SEIL.jp)**
IIJ独自開発ルータ「SEIL (ザイル)」の公式ファンページです。SEILに関連する最新情報や活用方法、便利な設定方法、開発秘話、などをお届けします。

September 2018 Vol.40

IIJ
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。



本冊子の情報は2018年9月時点のものです。


IIJ-MKTG019-0040

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