

2013年1月18日発行  
毎月1回18日発行  
通巻333号  
(発刊267号)  
1985年7月1日  
第三種郵便物認可  
ISSN 0916-6297  
特別定価  
**1,380円**

The logo for 'Software Design' features the word 'Software' in a bold, pink, sans-serif font. Below it, the word 'Design' is written in a bold, yellow, sans-serif font. The 'D' in 'Design' is significantly larger and more prominent than the other letters.

**[ソフトウェア デザイン]**  
OSとネットワーク、  
IT環境を支える  
エンジニアの総合誌

# いざというときに備える

### Special Feature 01

# システムノビックアップ

[illegible]

## Special Feature 02

## IT業界のキーパーソンに聞く

2013年に  
来そうな  
「技術」・  
「ビジョン」  
はこれだ!

### Extra Feature 01

# クロス開発環境 『Qt Quick』入門

### Extra Feature 02

インターネットを  
陰で見守る  
Akamai

## Special Appendix

サーバ・ネットワーク・  
システム安全祈願!  
[特別付録]  
法輪寺電電宮  
情報安全  
護符シール

情 報 安 全 護 符

Keep your data safe.

富士通

# Software Design

この個所は、雑誌発行時には広告が掲載されていました。編集の都合上、  
総集編では収録致しません。

# IT エンジニア必須の 最新用語解説

TEXT: (有)オングス 杉山 貴章 SUGIYAMA Takaaki  
takaaki@ongs.co.jp

## テーマ募集

本連載では、最近気になる用語など、今後取り上げてほしいテーマを募集しています。sd@gihyo.co.jp宛にお送りください。

## WinRT

### Windows 8 アプリ開発のための「WinRT」

Microsoft が新たにリリースした Windows 8 には、「Modern UI」または「Windows 8 スタイル UI」と呼ばれるまったく新しいユーザーインターフェースが採用されています。Metro UI という開発コードで呼ばれていたこの UI は、タッチデバイスのために最適化されている点が大きな特徴であり、タブレットなどのキーボードやマウスを持たないデバイスで優れた使い勝手を実現できます。

Windows 8 はこの Modern UI を使ったアプリケーション（通称、Windows 8 アプリケーション）と、従来のデスクトップアプリケーションの両方をサポートしており、用途に応じて使い分けことができます。一方、Windows 8 には ARM プロセッサ用の「Windows RT」もあります。こちらはタッチデバイスにターゲットを絞ったもので、MS Office などの一部のアプリケーション以外は Modern UI のみをサポートしています。

「WinRT (Windows Runtime)」は、この Windows 8 スタイルのアプリケーション開発のために提供されている新しいアプリケーション実行基盤および API セットです。従来の Windows では、アプリケーション開発のために Win32 API や .NET Framework といった API セットおよびフレームワークが提供されていました。WinRT はこれらの技術をベースにして再構築・

拡張されたものとなっており、Windows 8 アプリケーション開発ではこの WinRT が中心的な役割を担うことになります。

### 複数の開発言語をサポート

WinRT では、.NET と C++（ネイティブ）、JavaScript の 3 つの開発言語をサポートしています。そのうち、.NET と C++ で開発する場合は、WPF (Windows Presentation Foundation) や Silverlight と同様に XAML (eXtensible Application Markup Language) を利用して UI を記述します。そのため、WPF や Silverlight の経験がある開発者にとっては、馴染みのあるスタイルで WinRT を使ったアプリケーションを開発できます。

また、C++ 向けには新たに C++/CX や C++ AMP といった言語拡張が提供されています。C++/CX は C++ で WinRT を利用するための拡張であり、.NET を扱うために従来から提供されていた C++/CLI に似た文法で使うことができます。C++ AMP は GPU を利用して大規模データを高速処理するための拡張です。これによって、パフォーマンスが要求される用途では XAML と C++ の組み合わせという選択もできるようになっています。

JavaScript を使用する場合は、UI の記述には HTML5/CSS を利用します。JavaScript API は、標準の API に WinRT を参照する機能や Windows 8 スタイルの UI を作成するための専用のライブラリなど

を加えて拡張した形になっており、HTML5 と JavaScript の経験者であれば容易に使いこなすことができます。このことは、.NET や C++ の知識がない Web アプリケーション開発者や Web デザイナーにも、Windows 8 アプリケーション開発の門戸が広く開かれていることを意味しています。

### WinRT の狙い

WinRT が HTML5 や JavaScript といった標準技術をサポートしている背景には、より多彩な技術を持った開発者を Windows 8 アプリケーションの開発に巻き込みたいという狙いがあります。Windows 8 はデスクトップ PC だけでなく、タブレットをはじめとするモバイル端末もサポートする汎用的な Windows として開発されました。そのため、Web アプリケーションやモバイルアプリケーションのノウハウを持った開発者の参加が成功の鍵を握っているとも言えるわけです。

一方で表現力やパフォーマンスなどの側面から、従来のプラットフォームネイティブの技術やノウハウが必要とされるケースも多々あります。WinRT は標準技術とネイティブ技術を共通の基盤の上でシームレスに融合させるものであり、Microsoft の戦略にとって極めて重要な位置づけにあるプラットフォームだと言えます。SD

# Software Design

この個所は、雑誌発行時には広告が掲載されていました。編集の都合上、  
総集編では収録致しません。

いざというときに備える

# システム バックアップ

| 019



第1章	バックアップ今昔、昔テープ今クラウド	伊勢 幸一	020
第2章	UNIXバックアップ系コマンド復習	石田 つばさ	028
第3章	クラウドのデータはどこにいくの? [Nifty Cloud編]	大久保 智之	034
第4章	クラウドでのバックアップの基本 [AWS編]	柳瀬 任章、 宮澤 慶	042
第5章	知っているつもり? ハードディスクのイロハ	杉田 正	050
第6章	復旧業者に聞く データ復旧の実際	編集部	062
第7章	安定運用を目指すための Flashストレージ導入ポイント	長谷川 猛	070
コラム1	バックアップツール紹介Part① MASAMUNE誕生の背景と目標	沼田 理	027
コラム2	バックアップツール紹介Part② 仮想環境でも単一のコンソールで管理できるバックアップソフト	編集部	077

広告索引

AD INDEX

広告主名	ホームページ	掲載ページ
サイバーエージェント	<a href="http://www.cyberagent.co.jp/">http://www.cyberagent.co.jp/</a>	裏表紙
シーズ	<a href="http://www.seeds.ne.jp/">http://www.seeds.ne.jp/</a>	表紙の裏
システムワークス	<a href="http://www.systemworks.co.jp/">http://www.systemworks.co.jp/</a>	P.18
スカイアーチネットワークス	<a href="http://www.skyarch.net/">http://www.skyarch.net/</a>	第2目次対向
日本クエスト・ソフトウェア	<a href="http://www.bakbone.co.jp/">http://www.bakbone.co.jp/</a>	第1目次対向
日本コンピューティングシステム	<a href="http://www.jcsn.co.jp/">http://www.jcsn.co.jp/</a>	裏表紙の裏
ハイパーボックス	<a href="http://www.domain-keeper.net/">http://www.domain-keeper.net/</a>	第3目次対向

広告掲載企業への詳しい資料請求は、本誌Webサイトからご応募いただけます。 > <http://sd.gihyo.jp/>



# Software Design

この個所は、雑誌発行時には広告が掲載されていました。編集の都合上、  
総集編では収録致しません。

キーパーソンに聞く

# 2013年に来そうな 技術・ビジョンはこれだ!

| 079

第1章	<b>FreeBSD, Linux, Windows, OSはどう変わるのか?</b>	<b>080</b>
	2013年、FreeBSDで注目すべき7つのポイント 2013年はDebian7.0のリリースとさまざまな改善提案の行方に注目! 2013年、Ubuntuはアグレッシブに変化する! 2013年、Windows 8/Windows Serverはシステム構築にどんな影響を与えるか?	後藤 大地 やまねひでき あわしろいくや 横山 哲也
第2章	<b>ソフトウェア開発はどうなるのか</b>	<b>085</b>
	まだまだ注目技術に事欠かないWeb標準技術 ソフトウェア開発方法論としてのDDD、ユビキタス言語、Scala+Play2の可能性 Google Go言語はどうなった? プログラミングに役立つ数学 IT最前線シリコンバレー発、2013年注目のビッグデータ技術	白石 俊平 羽生田 栄一 千葉 滋 結城 浩 古橋 貞之
第3章	<b>情報技術の新しい流れ</b>	<b>090</b>
	RedHat社内でも注目、Linuxで利用できる最新メニューコア技術 そのメールのやりとり、開発の生産性を下げていませんか? クラウドの向こう側、AWSクラウドデザインパターン(CDP)! 海外から見て、世界標準で2013年以降、主流になりそうな情報技術は?	小島 克俊 藤村 能光 鈴木 宏康 谷本 真由美
第4章	<b>ネットワーク技術はどうなるのか</b>	<b>096</b>
	データセンターの未来とは、目指す新しいサービスとは? ネットワーク・インフラエンジニアは、どんな技術を身につけておくべきか? OpenFlowとファブリック、2013年にネットワーク技術はどうなる?	田中 邦裕 伊勢 幸一 小宮 崇博
第5章	<b>SIerは生き残るか?</b>	<b>101</b>
	ユーザ企業から見た、生き残るSIerの条件 SIerが今後生き残ることができるのか SIerに未来はあるのか? —— 現実を見よ 成功しない方法 35歳のITエンジニアが今後10年仕事を続けていくのに必要な技能は何か	谷口 有近 湯本 堅隆 神林 飛志 仙石 浩明 清水 亮

<b>一般記事</b>	<b>Article</b>
新年号特別付録!! 法輪寺電電宮情報安全護符シールの使い方	編集部 <b>017</b>
Win/Mac/LinuxアプリのUI記述が手軽にできる! クロス開発環境『Qt Quick』入門	折戸 孝行 <b>108</b>
増え続ける脅威に対抗せよ! インターネットを陰で見守るAkamai	渡邊 圭太、土屋 貴史、三宅 琢也、 倉持 尚員、中谷 喜久、新村 信 <b>120</b>

<b>巻頭Editorial PR</b>	<b>Editorial PR</b>
<b>【連載】Hosting Department [第81回]</b>	<b>H-1</b>

<b>アラカルト</b>	<b>A La Carte</b>
ITエンジニア必須の最新用語解説 [49] WinRT	杉山 貴章 <b>ED-1</b>
読者プレゼントのお知らせ	<b>016</b>
バックナンバーのお知らせ	<b>018</b>
SD BOOK FORUM	<b>078</b>
SD NEWS & PRODUCTS	<b>200</b>
Letters From Readers	<b>206</b>



# Software Design

この個所は、雑誌発行時には広告が掲載されていました。編集の都合上、  
総集編では収録致しません。

## Column

digital gadget[169]	質感豊かな、Skeuomorphismの世界とは?	安藤 幸央	001
小飼弾の コードなエッセイ[31]	The Round Lens in the Square Hole	小飼 弾	004
Google、Apple、Twitter... 深掘り裏読み 最新Webトレンド[33]	ビッグデータが米国の歴史を変えた? Amazonクラウドで 大成功したオバマ陣営/ハイヤーの革命児Uber	滑川 海彦、 高橋 信夫 (TechCrunch Japan翻訳者)	006
秋葉原発! はんだづけカフェなう[27]	World Maker Faire New York	坪井 義浩	010
ニートなphaの ぶらぶら日記 ギークハウスなう[33]	褒めればステマ、けなせば炎上、 とにかくネットは生きにくい	pha	014
Hack For Japan~ エンジニアだからこそできる 復興への一歩[13]	IT Bootcamp座談会:石巻の高校生との 学びの場で感じた支援のかたち	古山 隆幸、原 亮、佐々木 陽、 山本 直也、小野 哲生、 及川 卓也、高橋 憲一	182
温故知新 ITむかしばなし[19]	CPUのバグ	たけおかしょうぞう	186
Software Designer[44]	コンピュータサイエンス2.0[Part 3]:大学教育とオンライン教育 Daniel Jackson	Bart Eisenberg	188

## Development

Emacs 64bit化計画![5]	IMEの実装と環境整備	太田 博志	130
テキストデータなら お手のもの 開眼シェルスクリプト[13]	メールファイルを操作する ——grepでリストを作って一気に処理	上田 隆一	136
iPhone OSアプリ開発者の 知恵袋[33]	セルフブランディングのすすめ	リオ・リーバス	142
Androidエンジニア からの招待状[33]	マルチプラットフォーム開発環境を使ってみよう(2)	嶋崎 聡	150
ハイパーバイザの作り方[4]	I/O仮想化「割り込み編・その1」	浅田 拓也	156

## OS/Network

Ubuntu Monthly Report[33]	UbuntuとNexus 7の微妙なカンケイ	あわしろいくや	161
IPv6化の道も一歩から[2]	いよいよIPv6化を实践、 そのときに考えること、調べること	廣海 緑里、渡辺 露文、 新 善文、藤崎 智宏	166
レッドハット恵比寿通信[4]	パーソナルクラウドで自動化三昧	中井 悦司	172
Linuxカーネル 観光ガイド[10]	F2FSとは何か	青田 直大	174
Monthly News from jus[15]	インターネットの未来をかいま見たIC2012	今泉 貴史	180

## Inside View

OpenBlocks A ファミリによる サーバ実力診断最終回	XBee ZBによるZigBeeネットワークの構築	山縣 敦	192
インターネットサービスの 未来を創る人たち[19]	スマートフォンサービスを支える データベース環境(後編)	川添 貴生	198

### Part

Logo Design	ロゴデザイン > デザイン集合ゼブラ+坂井 哲也
Cover Design	表紙デザイン > Re:D
Cover Photo	表紙写真 > GYRO PHOTOGRAPHY/amanaimagesRF/Getty Images
Page Design	本文デザイン > 岩井 栄子、近藤しのぶ、SeaGrape、安達 恵美子 [トップスタジオデザイン室] 轟木 亜紀子、阿保 裕美、佐藤 みどり [BUCH+] 伊勢 歩、横山 慎昌 森井 一三、Re:D、[マップス] 石田 昌治



# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、  
総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、総集編では収録致しません。

# 小さくても、中身充実！

「あれ何だったっけ？」  
「こんなことできないかな？」  
というときに、すぐに調べられる機能引きリファレンス。  
軽くてハンディなボディに  
密度の濃い内容がギュウッと凝縮！  
関連項目への参照ページもあって、  
検索性もバツグン！



岡本隆史、武田健太郎、相良幸範 著  
四六判／272 ページ  
定価 2,604 円 (2,480 円 + 税)  
ISBN978-4-7741-5184-7



片瀬彼富 著、山田祥寛 監修  
四六判／448 ページ  
定価 2,709 円 (2,580 円 + 税)  
ISBN 978-4-7741-5067-3



山森文範 著  
四六判／304 ページ  
定価 2,499 円 (2,380 円 + 税)  
ISBN 978-4-7741-4396-5



石田つばさ 著  
四六判／448 ページ  
定価 2,289 円 (2,180 円 + 税)  
ISBN 978-4-7741-4836-6



峇名亮典、平山智恵 著  
四六判／576 ページ  
定価 2,394 円 (2,280 円 + 税)  
ISBN 978-4-7741-3816-9



細島一司 著  
四六判／464 ページ  
定価 2,919 円 (2,780 円 + 税)  
ISBN 978-4-7741-5135-9



土井敏、高江賢、飯島聡、高尾哲朗 著、山田祥寛 監修  
四六判／488 ページ  
定価 2,709 円 (2,580 円 + 税)  
ISBN 978-4-7741-4948-6



朝井淳 著  
四六判／640 ページ  
定価 2,079 円 (1,980 円 + 税)  
ISBN 978-4-7741-3835-0



山田祥寛 著  
四六判／528 ページ  
定価 2,919 円 (2,780 円 + 税)  
ISBN 978-4-7741-4980-6



森直彦 著  
四六判／392 ページ  
定価 2,709 円 (2,580 円 + 税)  
ISBN 978-4-7741-4982-0

この時代を**生き抜く自信**はありますか？

最新刊！

# 生きる技術！叢書



危機の時代に  
「生きる技術」  
を伝える  
新しい教養書



内田樹 著  
定価：本体 1,580 円 + 税  
ISBN 978-4-7741-4709-3



パオロ・マッツァリーノ 著  
定価：本体 1,480 円 + 税  
ISBN 978-4-7741-5385-8



藤井聡 著  
定価：本体 1,580 円 + 税  
ISBN 978-4-7741-5023-9



小田嶋隆 著  
定価：本体 1,480 円 + 税  
ISBN 978-4-7741-5316-2



小田嶋隆 著  
定価：本体 1,480 円 + 税  
ISBN 978-4-7741-4870-0



香山リカ 著  
定価：本体 1,380 円 + 税  
ISBN 978-4-7741-5066-6



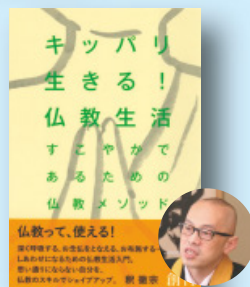
慎泰俊 著  
定価：本体 1,480 円 + 税  
ISBN 978-4-7741-5152-6



小林弘人 著  
定価：本体 1,480 円 + 税  
ISBN 978-4-7741-4935-6



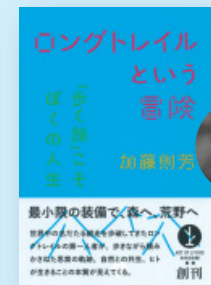
岩田健太郎 著  
定価：本体 1,580 円 + 税  
ISBN 978-4-7741-4837-3



釈徹宗 著  
定価：本体 1,580 円 + 税  
ISBN 978-4-7741-4708-6



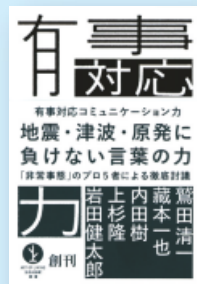
林志行 著  
定価：本体 1,580 円 + 税  
ISBN 978-4-7741-4764-2



加藤則芳 著  
定価：本体 1,580 円 + 税  
ISBN 978-4-7741-4768-0



小川仁志 著  
定価：本体 1,580 円 + 税  
ISBN 978-4-7741-4707-9



鷲田清一、内田樹、  
上杉隆、岩田健太郎、  
藏本一也 著  
定価：本体 1,280 円 + 税  
ISBN 978-4-7741-4871-7

いま、生きるための教養を！

# Software Design plus

Software Design plusシリーズは、OSとネットワーク、IT環境を支えるエンジニアの総合誌『Software Design』編集部が自信を持ってお届けする書籍シリーズです。

## サーバ/インフラエンジニア

### 養成読本 仮想化活用編

Software Design編集部 編  
定価 1,980円+税 ISBN 978-4-7741-5038-3

## サーバ/インフラエンジニア

### 養成読本 管理/監視編

Software Design編集部 編  
定価 1,980円+税 ISBN 978-4-7741-5037-6

## Webエンジニアのための

### データベース技術[実践]入門

松信 嘉範 著  
定価 2,580円+税 ISBN 978-4-7741-5020-8

## 2週間でできる!

### スクリプト言語の作り方

千葉 滋 著  
定価 2,580円+税 ISBN 978-4-7741-4974-5

## PCのウイルスを根こそぎ

### 削除する方法

本城 信輔 著  
定価 1,980円+税 ISBN 978-4-7741-4867-0

## Androidエンジニア養成読本

Software Design編集部 編  
定価 1,880円+税 ISBN 978-4-7741-4859-5

## Vyatta入門

### 実践ルーティングから仮想化まで

近藤 邦昭、松本 直人、浅間 正和、  
大久保 修一(日本Vyattaユーザー会) 著  
定価 3,200円+税 ISBN 978-4-7741-4711-6

## プロのためのLinuxシステム・

### ネットワーク管理技術

中井 悦司 著  
定価 2,880円+税 ISBN 978-4-7741-4675-1

## サーバ/インフラエンジニア

### 養成読本

Software Design編集部 編  
定価 1,880円+税 ISBN 978-4-7741-4600-3

## Linuxエンジニア養成読本

Software Design編集部 編  
定価 1,880円+税 ISBN 978-4-7741-4601-0

## Nagios統合監視

### [実践]リファレンス

(株)エクストランス 佐藤 省吾、  
Team-Nagios 著  
定価 3,200円+税 ISBN 978-4-7741-4582-2

## プロのためのLinuxシステム

### 構築・運用技術

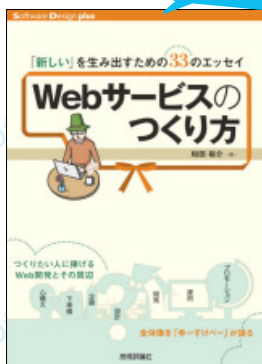
中井 悦司 著  
定価 2,880円+税 ISBN 978-4-7741-4501-3

## サーバ構築の実例がわかる

### Samba[実践]入門

高橋 基信 著  
定価 2,480円+税 ISBN 978-4-7741-4405-4

最新刊!



## 和田 裕介 著

A5判・208ページ  
定価 2,180円(本体)+税  
ISBN 978-4-7741-5407-7



## 三苦 健太 著

B5判・400ページ  
定価 3,200円(本体)+税  
ISBN 978-4-7741-5189-2

最新刊!



## 河村 嘉之、川尻 剛 著

B5変形判・480ページ  
定価 2,980円(本体)+税  
ISBN 978-4-7741-5438-1

最新刊!



(株)マビオン、山岸 靖典、  
谷内 栄樹、本城 博昭、  
長谷川 行雄、中村 和也、  
松浦 慎平、佐藤 亜矢子 著

B5変形判・256ページ  
定価 2,580円(本体)+税  
ISBN 978-4-7741-5325-4



## 中井 悦司 著

B5変形判・352ページ  
定価 3,400円(本体)+税  
ISBN 978-4-7741-5143-4



## 木村 明治 著

B5変形判・416ページ  
定価 3,180円(本体)+税  
ISBN 978-4-7741-5026-0



## 木本 裕紀 著

A5判・352ページ  
定価 2,780円(本体)+税  
ISBN 978-4-7741-5025-3



## 鶴長 鎮一 著

A5判・344ページ  
定価 2,980円(本体)+税  
ISBN 978-4-7741-5036-9

紙面版  
A4判・16頁  
オールカラー

# 電脳会議

一切  
無料

D E N N O U K A I G I

## 新規購読会員受付中!



『電脳会議』は情報の宝庫、世の中の動きに遅れるな!

『電脳会議』は、年6回の不定期刊行情報誌です。A4判・16頁オールカラーで、弊社発行の新刊・近刊書籍・雑誌を紹介しています。この『電脳会議』の特徴は、単なる本の紹介だけでなく、著者と編集者が協力し、その本の重点や狙いをわかりやすく説明していることです。平成17年に「通巻100号」を超え、現在200号に迫っている、出版界で評判の情報誌です。

今が旬の  
情報  
満載!



新規送付のお申し込みは…

Web検索が当社ホームページをご利用ください。  
Google、Yahoo!での検索は、

電脳会議事務局

検索

当社ホームページからの場合は、

<https://gihyo.jp/site/inquiry/dennou>

と入力してください。

一切無料!

●『電脳会議』紙面版の送付は送料含め費用は一切無料です。そのため、購読者と電脳会議事務局との間には、権利&義務関係は一切生じませんので、予めご了承下さい。

毎号、厳選ブックガイド  
も付いてくる!!



『電脳会議』とは別に、1テーマごとにセレクトした優良図書を紹介するブックカタログ (A4判・4頁オールカラー) が2点同封されます。扱われるテーマも、自然科学/ビジネス/起業/モバイル/素材集などなど、弊社書籍を購入する際に役立ちます。

# Software Design

この個所は、雑誌発行時には広告が掲載されていました。編集の都合上、  
総集編では収録致しません。

# DIGITAL GADGET

安藤 幸央 — Yukio Ando —  
EXA CORPORATION  
[Twitter] >> @yukio\_andoh  
[Web Site] >> http://www.andoh.org/

Volume  
169

## 質感豊かな、Skeuomorphismの世界とは？

### 高度な描写で 現実と仮想世界をつなぐ ユーザインターフェース手法

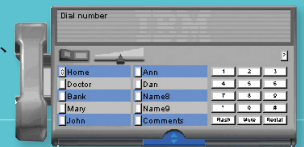
>>> Skeuomorphism (スクオモーフィズム) という言葉を聞いたことはあるでしょうか？

Skeuomorphismは世の中にあるモノの質感を真似て、デジタル世界のわかりやすいユーザインターフェース表現に生かす手法です。ファイルやフォルダ、デスクトップ、ゴミ箱など、現実世界の事象をデジタル世界に持ち込み、わかりやすく表現したものをメタファーと呼びますが、それとはまた違うアプローチです。

現実世界の事象をデジタル世界に持ち込み、わかりやすく表現したものをメタファーと呼びますが、それとはまた違うアプローチです。

現実にあるモノの質感と特徴を忠実に真似たユーザインターフェースなため、誰もが一目で使い方を把握することができる一方、装飾過剰であるともいわれます。また、現実にあるものを真似しすぎると、デジタルデバイスの世界ではかえって使いにくくなってしまいうケースがあるという問題も指摘されています。

古くはIBMの社内で使われるビジネス電話機を模した「RealPhone」(図1)をはじめとして、Apple、ことにiPhone/iPadアプリの中に数多くのSkeuomorphism手法が見られます。



▲図1 / RealPhone

- iOS 6の「Music」アプリのアルミヘアライン加工のスライダー(傾きに応じて輝きが変化する)
- Podcastアプリのオープンリールテープ(図2)
- iPhoneの電卓(初代はBRAUNの実在の電卓そっくりでした)
- iPhoneの録音アプリ(マイクをコンコンとたたくと、メータが動き、録音状態を確認できる。図3)
- iBooksの本棚やiBooksのページめくり
- iPadのキーボードのホームポジションの目印(図4)
- そのほか、革やノートの質感を持ったアプリが多数(図5)

一般的なスマートフォンアプリでも、Skeuomorphismテクニックを駆使したものも増えてきており、シンプル／ミニマルなデザインの対極に位置した手法は興味を持って受け入れられているのではないのでしょうか。



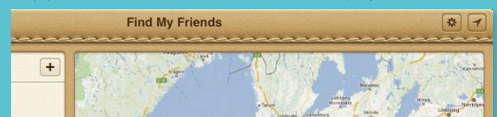
▲図2 / Podcastアプリ



▲図3 / 録音アプリ



▲図4 / ソフトキーボードのホームポジション目印



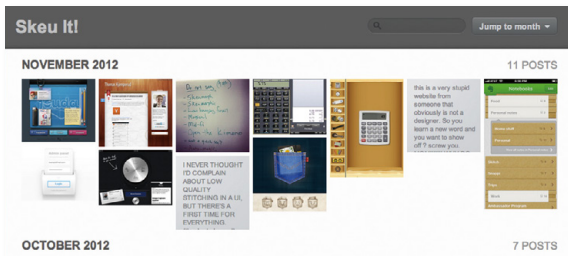
▲図5 / 革の質感を再現

### gadget 1 Skeu it!

<http://skeu.it/>

## Skeuomorphism事例サイトその1

Skeuomorphismに即した事例を集めたアーカイブサイト。検索や投稿、RSS配信なども便利に活用できます。アーカイブされた画像群を何となく見ているだけで「ああ、そういう活用方法もあったか!」と新しい発見ができることでしょう。



### gadget 2 MediaLoot

<http://medialoot.com/item/free-ui-toggle-switches/>

## Skeuomorphismアプローチが得意な素材サイト

質感豊かなボタンデザイン集のサイト。たいていはフリーですが、高品質なものについては有料の素材もあります。Skeuomorphismに限らず、数多くのデザイン素材が登録されていますが、MediaLootはとくにSkeuomorphismアプローチでの素材が多いのが特徴です。

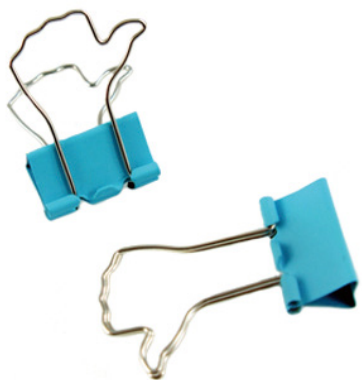


### gadget 3 LIKE CLIPS

[http://www.25togo.com/store/Product.aspx?pc\\_kind=Brands&pc\\_id=0000100055&prod\\_id=0001788](http://www.25togo.com/store/Product.aspx?pc_kind=Brands&pc_id=0000100055&prod_id=0001788)

## いいね!ボタンペーパークリップ

Facebookボタンの「いいね!」は、英語では「Like!」、と言語によって表現が異なりますが、親指を上に向けたサインは万国共通のようです。これからはデジタル世界で用いられている表現を現実世界のモノが模倣するという、逆Skeuomorphismが流行りはじめるかもしれません。

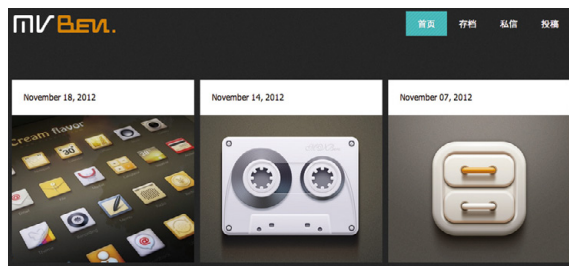


### gadget 4 MVBen

<http://www.mvben.com/>

## Skeuomorphism事例サイトその2

Skeuomorphismをテーマとしたアイコンデザインやユーザインターフェース部品を集めたサイトです。美しく立体表現されたデザイン画の数々と、独特のテイストに感化されることでしょう。

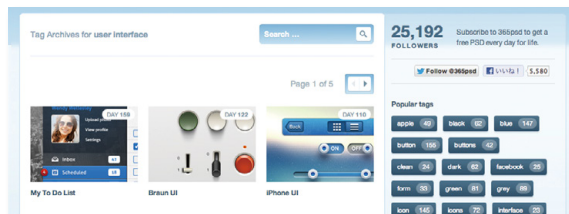


### gadget 5 365psd

<http://365psd.com/tag/user-interface/>

## Photoshop形式で提供される質感豊かな素材

365psdはPhotoshop素材であるPSDファイルが数多く登録されたサイトです。とくにuser-interfaceカテゴリは、質感豊かな素材群が登録されており、ユーザインターフェースのアイデアを練るときにも役立つサイトです。

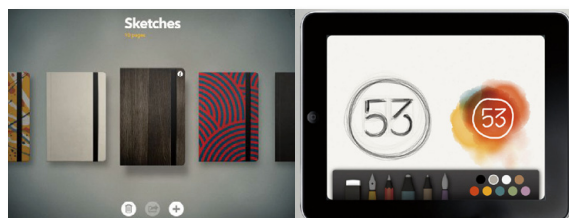


### gadget 6 Paper by FiftyThree

<http://www.fiftythree.com/paper>

## Skeuomorphism的なお絵描きアプリ

Paper by FiftyThreeは、数あるアプリの中でも突出したSkeuomorphism的なiOSアプリの1つです。Paperはお絵描きアプリの一種で、絵筆やペンを駆使して水墨画風のにじんだ絵や、ペンの質感で絵柄が描けるうえに、現実存在するノートブックのようにファイルを管理します。

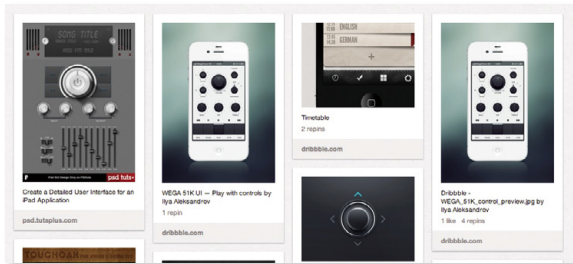


## gadget 7 PinterestのSkeuomorphismカテゴリ

<http://pinterest.com/marcosecchi/graphics-skeuomorphism/>

### Skeuomorphism事例サイトその3

画像共有サイトPinterestの中でも、Skeuomorphism三昧の画像収集ページがいくつかあります。その中でもmarco secchi氏のコレクションは一見の価値があります。すでに世の中にあるSkeuomorphism的アプリの画面やユーザインターフェース部品などが集められています。

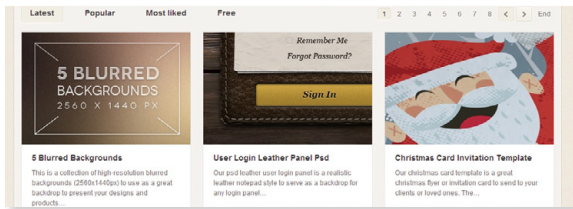


## gadget 8 Pixeden

<http://www.pixeden.com/>

### 素材感を重視した画像提供サイト

Skeuomorphism的素材を用意する一番便利な方法は、革や布、紙、金属などの実物をデジタルカメラで撮影し、その素材感たっぷりの画像を活用することです。PixedenではWebやアプリのデザインに用いることのできる質感豊かな素材が数多く登録されているサイトです。ほかにも同様のサイトとして、<http://opendept.net/>、<http://subtlepatterns.com/>、<http://pixelsdaily.com/>などがあります。

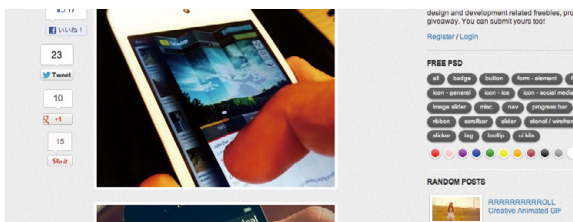


## gadget 9 PAPER FOLD EFFECT IN APP DESIGN

<http://favvuluous.com/post/1128/paper-fold-effect-in-app-design>

### 紙を折りたたむようなUIデザイン集

Skeuomorphism的デザイン手法の1つとして、紙をめくるようなアニメーションがよく使われます。PAPER FOLD EFFECT IN APP DESIGNサイトは、そういった紙をめくるような手法のなかでも“折りたたむ”効果を活用したデザインを数多く紹介しています。最近ではページめくりの効果も、jQueryなどのライブラリの応用で楽に実現できるようになりました。

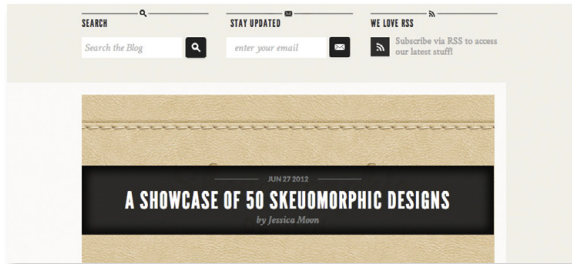


## gadget 10 Shocase of 50 Skeuomorphic Designs

<http://www.dtepathy.com/blog/inspiration/50-skeuomorphic-design>

### Skeuomorphism事例サイトその4

Skeuomorphism的デザインを50件紹介したサイトです。順々に見ていくと、アプリを見ているのか、現実のモノを見ているのかおぼつかなくなってしまうほどです。Skeuomorphism的アプリ作成時のヒントになるサイトです。



質感が豊かで、目をひくSkeuomorphismには弊害もあり、初めて見たときにも親しみやすいユーザインターフェースでありつつも、慣れると無駄に思える要素もあることです。デジタルデバイスの表現能力も高まっており、実物と見まがう表現ができたとしても、まだその表現方法は発展途上にあり、今ある何かを模倣している段階であるとも言えます。

Skeuomorphism手法をうまく活用するには、真似る要素が何なのかを注意深く考えることが大切です。短期的な親しみやすさ、取っ付きやすさとはまた別に、長期的な使いやすさも考慮しなければなりません。

初心者も使いやすく、習熟者が素早く使えるよう配慮し、単なる懐古主義ではなく、デジタルの世界で意味のある模倣をしてこそSkeuomorphism手法が生きてきます。

1944年にスイス国鉄のエンジニア、ハンス・フィルフィカー氏がデザインした「スイス レイルウェイ ステーションクロック」と、新しいiPadに搭載されている標準アプリの時計が酷似していると問題になりました。結局はライセンス料の支払い、クレジットの表記でことは収まったそうですが、膨大なライセンス料が支払われたとも噂されています。

針の動きで表現する時計は未来永劫使われると思われるかもしれませんが、現在もう主流ではない機器はどうでしょうか？ iOSのPodCastアプリに使われているオープンリールの表現はテープの動きで状態を把握しやすいかもしれませんが、すでにカセットテープも見なくなった現在、この機器が何を示すのか知らない世代も増えてくることでしょう。単に古いモノの見栄えや役割を引きずるのではなく、使いやすくなりやすい、次の段階の表現が求められているのかもしれないね。SD

# The Round Lens in the Square Hole



## 小飼弾の コードなエッセイ

TEXT= 小飼 弾 KOGAI Dan dankogai@dan.co.jp



### 先祖返りするデジカメ

もしかしてCASIO QV-10以来だろうか。携帯電話の一機能ではなく単体のデジタルカメラを自分用に手に入れたのは。機種はSonyのCybershot DSC-HX30Vというもの。「買った」ではなく「手に入れた」というのは、妻のデジカメがα NEX-6になった際のポイントが化けたものだから。スマートフォンでは望めない光学ズーム20倍もさることながら、かつてのデジカメには望めなかったGPSとWi-Fiの標準搭載がうれしい。これがミラーレス一眼レフの「おまけ」になってしまうのだから、世は買い手天国の売り手地獄である。



Sony Cybershot DSC-HX30V

しかしこれをいじっているうちに、ふと気づいた。

これって先祖返りじゃね？

まずはQV-10をご覧ください。読者の中には本品登場時点ではまだ生まれてなかった方もいらっしゃるかもしれないので。1995年に発売されたこれ

は、今年の9月には国立科学博物館が未来技術遺産に認定した逸品である。



Casio QV-10

次に今のステイルカメラの形を築いた、Leica M3をご覧ください。登場は1954年。もちろん私もまだ生まれていない。もちろんデジタルではない、銀塩フィルムに撮影するアナログカメラだ。カメラにとっての本機は、携帯電話にとってのiPhoneのようなものだったようである。



Leica M3

なぜQV-10の子孫であるはずの現代のデジカメはむしろLeica M3の子孫のように見えるのだろうか。直方体のボディの一番広い面の真中あたりに、

円筒形のレンズがぶら下がっている。コンデジことコンパクトデジタルカメラから、デジイチことデジタル一眼レフカメラに至るまで。私のDSC-HX30Vはおろか、妻のNEX-6もこの例外ではない。

Leica M3の形は、工学的な必然性でこうなった。ロール、つまり文字どおり巻物に収容されたフィルムは、撮影時に露光された後、カメラ上部の巻き取りレバーによってロールの反対側に巻き取られる。フィルムはボディの中で巻物を読んでいるような状態になっており、レンズが集めた光が当たるのは、その読んでいる部分。撮影時以外にフィルムに光を当てないようにするためには、巻物全体は箱で覆えばいい。かくして円筒形が生えた直方体は、カメラのアイコンとなった。

しかしフィルムを用いないデジタルカメラに、そんな工学的必然性はみじんもない。実際QV-10のレンズは、ボディの真横に回転式ピボットで付いている。QV-10の大きさは今のコンパクトデジタルカメラと変わらない。おかげでポケットに入れていたこれをイタリアですられたというのは苦い思い出だが、何が言いたいかというと、1995年当時にもカメラ部分はこの程度の大きさだったということだ。今のスマートフォンでは、5mm立方のサイコロにおさまってしまう。

「光学系は小さくできないから」というツッコミは、「ならカムコーダーはどうなのだ」というボケ返しにツッコミ返せない。実際現代のカムコーダー

は、液晶ディスプレイが光学系の横から生えている。そして現代のカムコーダーが、動画だけではなく静止画像も撮れるのと同様、およそ現代のデジカメに動画撮影ができないものはない。

なのにデジカメは、銀塩カメラそっくりの形をしている。現代のパソコンがタイプライターの子孫みたいな形をしていることを彷彿とさせる話ではないか。

「週刊ダイヤモンド 2012年9月22日号」の特集「カメラ激変!」によると、QV-10を生み出したカシオのデジカメ事業は、最盛期(2007年)の1/3以下にまで縮小しているそう。確かにカシオが得意とするコンデジはスマホに最も食われている分野であるが、スマホの脅威以上に、せっかくの革命的商品を他社と同じく先祖返りさせてしまったことにもその原因はあるのではないかな?

フィルムからの自由が本来もたらすべきだったのは、造型の自由ではなかったのか?

たとえば銀塩カメラをデジタル化するのではなく、今までカメラが付いていなかった光学機器をカメラ化するというのはどうだろう? QV-10より先に発売されたApple QuickTake 100は双眼鏡のような形をしていたし、「撮った後の画像のどこでも後から焦点を合わせ直せる」画期的なデジカメ、Lytroは万華鏡のような形をしている。メガネのようにかけられるデジカメはいつできるのだろうか? 私の老眼が始まる前にぜひ文字どおりお目にかかるたいものなのだが……。SD



Apple QuickTake 100



Lytro

協力：TechCrunch Japan

<http://jp.techcrunch.com/>

## ▶ ビッグデータが米国の歴史を変えた？ Amazon クラウドで大成功した オバマ陣営

### ▶ ハイヤーの革命児 Uber

滑川 海彦 NAMEKAWA Umihiko ● [techtrans1999@gmail.com](mailto:techtrans1999@gmail.com) (TechCrunch Japan 翻訳者)高橋 信夫 TAKAHASHI Nobuo ● [nobuo.takahashi@nifty.com](mailto:nobuo.takahashi@nifty.com) (TechCrunch Japan 翻訳者)

本連載では、Webメディア「TechCrunch Japan」(<http://jp.techcrunch.com/>)の記事を翻訳している2人が、毎回、同サイトで取り上げている最新Webサービスや企業・ビジネスに関する膨大なエントリを訳出する中で集積している、米国を中心とした動向、さらにその背景を解き明かしていきます。

## ビッグデータが米国の歴史を変えた？ Amazon クラウドで大成功したオバマ陣営

滑川 海彦

### Amazon クラウド (オバマ) 対 Orca (ロムニー)

解散総選挙が決まり騒然とする年末の日本ですが、相変わらずインターネットによる選挙運動は禁止されたまま。さらに選挙運動期間中の世論調査による選挙動向の予測さえ禁止されています。このような有権者を愚民扱いする固陋な制度がなぜ改革できないのか不可解です。民意の正しい反映にあたって、インターネット利用と世論調査公表の解禁は定数削減以上に緊急の課題ではないでしょうか？

さて11月の米国大統領選挙は大接戦と予想されていましたが、蓋を開けてみれば大統領選挙人獲得数は332対206と現職オバマ大統領の圧勝に終わりました。米国の大統領選挙は州ごとに大統領選挙人を選ぶ間接選挙であり、ほとんどの州が「勝者総取り」方式を採用しているため、直接投票率の差がわずかでも選挙人数では大差になるという事情があります。スイングステートと呼ばれる支持率伯仲の州をどちらが制するかで最終結果が大きく動きます。今回も直接投票での得票差はわずか2.85%に過ぎませんでした。

選挙戦の終盤でものをいうのが投票率です。世論調査の支持率で上回っていても支持者が実際に投票所に足を運んでくれないのでは意味がありません。

運動員が最後の瞬間に投票所に足を運ぶよう支持者に電話で訴えることが決定的な重要性を持ちます。

今回オバマ陣営は選挙キャンペーンシステムにすべてAmazon Web Services (AWS) クラウドを利用しました。Amazonの公式ブログ<sup>注1</sup>によると、オバマ選対はAWS上で200以上のアプリケーションを開発し、そのユーザは数百万人に上りました。ビッグデータの処理にはHPのデータ解析システムVerticaとAWS版のHadoop、Elastic MapReduceを全面的に利用しました。電話キャンペーンシステムは支持者の氏名、電話番号、居住地、職業、性別、支持の程度などの情報に加えて最新世論調査の結果なども総合し、最も効果的な電話リストを作成して運動員のスマートフォンに表示させる必要があります。オバマ陣営の電話キャンペーンツールは同時に7,000人の運動員をサポートし、最後の4日間に200万通話を発信したといっています。

これに対してロムニー陣営が同様の目的で開発した電話ツール、Orca (シャチ) は前代未聞の大失敗に終わりました。これはデータセンターのサーバでホストされる伝統的なWebアプリだったようですが、

注1) 「AWS in Action - Behind the Scenes of a Presidential Campaign」  
URL <http://aws.typepad.com/aws/2012/11/aws-in-action-behind-the-scenes-of-a-presidential-campaign.html>

アクセスが急増するとクラッシュを繰り返してほとんど満足に稼働せず、3万人の運動員が決定的な最後の数時間を手をつかねて過ごすという結果になりました。Orcaはロムニー選対のボランティアとコンサルタントが急ごしらえしたもののようです。ロムニー陣営はITに4,000万ドルもつぎ込んだにもかかわらず、なぜこの重要なシステムを最高のプロに任せなかったのか不思議です。オバマ陣営とロムニー陣営のITに対する理解の度合いがそのまま明暗を分けたということでしょうか。

### 選挙結果予測を唯一全的中したシステムの秘密

一方、選挙後に一番メディアで話題になったのは、ニューヨークタイムズ紙の専属選挙アナリストのネイト・シルバーがコンピュータによる分析で50州すべてで選挙結果予測を的中させたことでした(図1)。TechCrunchではさっそく「政治専門家などいない」という記事を掲載しました<sup>注2</sup>。なにしろクリントン大統領の元補佐官ディック・モリスなどはロムニーが325票獲得すると予測して100票も外しています。

シルバーのシステムが収集する情報量は膨大であり、処理も高度なものだと言われていますが、その本質は各種世論調査の結果を規模、質、時期などによって重み付けし、過去の同種の選挙結果と照合してトレンドを割り出すというオーソドックスなものです。同種の大規模なコンピュータ分析は他にも多数あります。ではなぜシルバーの予測だけが的中したのでしょうか？

その秘密はネイト・シルバーの経歴にありそうです。シルバーはもともと政治や選挙の専門家ではありませんでした。シカゴ大学経済学部を卒業した後、2002年にはKPMG会計事務所に勤務中にメジャーリーグ野球選手の統計的評価システム、PECOTAを開発しました。これは映画「マネーボール」で日本でも知られるようになったセイバーメトリクスという選手評価システムをオンライン化したも

ので、2007年にはPECOTAをBaseball Prospectus社に売却して、選挙予測の分野に転身したということです。そして早くも2008年にはオバマ対マケインの大統領選で50州中49州の勝敗を的中させ、一躍注目されるようになりました。

いわゆる政治の専門家が作ったモデルが予測を外した理由はそのほとんどすべてが重み付けの失敗によるもののようです。あるモデルでは「民主党支持者の投票率は低い」と見て世論調査を重み付けしたところ、実際の投票率はずっと高く、結果は大外れとなりました(前述のAWSを利用した投票勧誘システムの成果かもしれません)。つまり「専門家としての知識」が結果として誤った先入観だったわけです。政治専門家は、もともと政治に興味がある人間なので、それぞれが強固な政治的立場から物事を考えます。シルバーの当たりの秘密は予断を廃してデータの意味するところだけを分析したことにありそうです。プログラミングの世界ではGIGO (Garbage In, Garbage Out、ゴミを入れるとゴミが出てくる) と言いますが、コンピュータモデルの世界でも言えるでしょう。

ちなみに、ネイト・シルバーが開拓したメジャーリーグ野球のコンピュータ分析ですが、同じくTechCrunchの記事によると、現在複数の球団がHadoopによるビッグデータ処理による戦術分析をテストしているそうです。メジャーリーグの球場ではすべてのヒットの飛跡がビデオカメラで記録されるなど試合データがマルチメディア化し、今までのデータベースでは管理しきれなくなるからだと言われています。将来、ここからさらに強力な選挙分析も生まれるかもしれません。

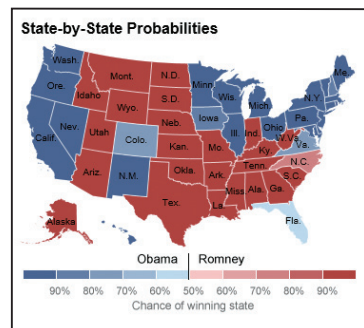


図1 ネイト・シルバーは事前にこの結果を完全に予測した

注2) 「大統領選でニューヨークタイムズのネイト・シルバーの数理モデル予測が全50州で的中—政治専門家はもはや不要？」  
URL <http://jp.techcrunch.com/archives/20121107pundit-forecasts-all-wrong-silver-perfectly-right-is-punditry-dead/>

## ハイヤーの革命児 Uber

高橋 信夫

ウーバー

Uberをご存じですか？米国で急速に成長している「スマートフォンで車を呼べるサービス」です。具体的な利用場面を想像してみましょう。路上にいる利用者がスマートフォンの専用アプリを立ち上げます。地図には近くを走っている空車が複数表示されます(図2)。1台を選ぶと迎えに来てくれます。運転手もiPhoneアプリを使っていて、客の現在位置を知っているのです。あらかじめクレジットカード情報を登録してあるので、目的地に着いたらそのまま降りるだけで支払いもおしまい。現金やカードはもちろんスマートフォンを取り出す必要さえありません。

Uberの共同創業者、トラビス・カラニック氏が11月に来日してTechCrunch Tokyoで講演したことでもあり、今回はどうしてこんなサービスが可能になり、そのためにどんな苦労があったのかをご紹介します。

### Uberを作った男とは

Uberの話に入る前に、創業者のトラビス・カラニック自身の波瀾万丈の物語を少し書きます。カラニックは高校生のとき、仲間と一緒にScourというファイル共有サービスを始めました。Napsterに似たサービスと言えば想像がつくでしょうか。そしてNapsterと同じようにメディア会社たちににらまれ、計29社からの訴訟で総額2,500億ドルの賠償請求をされます。結局示談になり100万ドルを払ってサービスを閉鎖しました。彼

いわく、「100万は2,500億よりずっと少ない」。次にスタートしたRed Swooshというサービスは、前と同じテクノロジーを使って、自分を訴訟した会社を顧客にしようという「リベンジビジネス」でした。結局29社中23社のメディアが彼のシステムの中を流れたそうで

す。その会社を売却して充電中だったある日、彼と共同創業者の2人は、サンフランシスコでタクシーを捕まえるのが異常にたいへんだということに気づきました。そして2010年にサンフランシスコでUberのサービスを開始すると、翌年にはニューヨーク、シアトル、シカゴ、ワシントンDC、パリなどの都市へ次々と進出していきました。資金も計3,500万ドルを調達済みで、出資者の中にはAmazonのCEOジェフ・ベゾスもいます。

### 車も運転手も持たない

毎日Uberを通じて大量の車が利用されていますが、Uber自体は車も所有せず運転手も雇っていません。小さなハイヤーやリムジンの会社、あるいは個人の運転手と契約を結んでいるだけです。このためUberは「自分たちはタクシー会社ではない」という立場で、タクシーの運行に必要な免許を取得することもなく、規則にも縛られないようにしています。実はサービス開始当初の名前はUberCabだったのですが、Cabがタクシーを連想するので社名から外したという経緯があります。

### 規制との戦い

サンフランシスコでスタートしたUberは、他の都市に進出するたびに、あらゆる規制や抵抗勢力と戦ってきました。現在も進行中です。もともとタクシー／ハイヤー業界は規制が厳しく、先行者の既得権も大きいので、それを破壊するUberに風当たりが強いのも当然です。都市ごとにさまざまな攻撃に曝されています。

たとえば2012年11月、シカゴで提出された法案は「ハイヤー車内で電子機器を使って計測することを禁止する」ものでした。UberではiPhoneのGPS機能を使って走行距離を測り、所要時間と合わせて料金を計算していますが、これを禁止するというのです。「時間と距離の両方で料金を請求する」ことに対しては他の都市でも反発があるのですが、シカゴ



図2 スマホで車を呼ぶサービスUber

の場合、「問題なのは、ここには以前から同様の料金体系があり、その場合は運転手が腕時計と車の走行距離計を使っていること」だとカラニックは言っています。つまり法案は、正確に測定することはまかりならん、と言っていることになります。もちろんこれは、なんとかしてUberを追い出すための方便である可能性が高いのですが、Uberは署名を募って規制の取り下げを当局に提出しましたが、結果はまだわかっていません。

## 料金はタクシーの1.5倍

伝統的業界を破壊する新サービスと聞くと、価格も破壊的に安いことを想像してしまいがちですが、実はUberの標準的な料金は一般タクシーの1.5倍です。それでもあの「捕まらないタクシー」よりいいと言って使う人がたくさんいるのです。また、使う車もタクシーの代名詞であるイエローキャブではなく、「ブラックカー」と呼ばれる高級感のある車種で、日本のハイヤーに相当します。さらに驚くのが「変動価格システム」を採用していることです。需要が供給を上回ったときには、「通常料金の何倍」という価格がアプリに提示されます。それでもいいと思う人はどうぞ、というわけです。しかしワシントンDCでは2011年の大晦日に料金が高騰してTwitterで大騒ぎになったそうです。

Uberの料金体系はちょっと複雑で、まず車種がいくつかあります。都市によって違いますが、サンフランシスコでは先ほどのブラックカーの他にもっと大きいSUV（スポーツ用多目的車）、価格の安いハイブリッドまたは中型車のUBERxがあります。車種ごとに基本料金があり、これに距離ごと、時間ごとの料金が加算されます。さらに一番安い選択肢としてUBERタクシーがあります。これは普通のイエローキャブをUberを通じて呼ぶもので料金体系が異なり、通常のタクシー料金の他に呼び出し料金1ドルと20%のチップが自動的に加算された金額がUberアカウントに請求されます。

Uberの車や運転手はUber専門ではなく、他でも客を乗せている車が、空いているときにはUberから紹介された客を乗せるので、運転手やハイヤー会社してみれば、客のいない時間を有効活用できて

万々歳です。こうして乗客だけでなく乗せる側も満足させているのがUberの成功の大きな理由です。

## 競争相手

最大のライバルは既存勢力かもしれませんが、Uberと同じような新規ビジネスもいくつか登場しています。英国ロンドン拠点のHailoとはニューヨーク、シカゴなどいくつかの都市で争っています。他にもCabulousやTaxi Magicなど、スマートフォンでタクシーを呼ぶサービスはいくつかありますが、シリコンバレーを賑わしている点ではUberが一番です。Uberでは博士号を持つ人を何人も雇っていて、それは、「需要と供給のバランスを取り、使いたい人の近くに車がいるように配置する」ために高度な技術が必要だからだそうです。ハイヤー呼び出しサービスの競争はテクノロジーの戦いでもあります。

ライバルではなくジャンルも異なりますが、コンセプトの似たサービスにAirbnbエア・ビー・エス・ビーがあります。こちらは宿泊場所を提供する人と泊まる場所を探している人を結び付けます。宿泊業者ではない個人でも空いた部屋があれば貸すことができます。カラニック氏はAirbnbのCEOとも懇意で、いろいろ参考にしているそうです。

## 日本への進出は？

TechCrunch Tokyoの講演中、カラニック氏は、東京で「20人程度の非常に有能な人材」を採用し、「ウーバー東京」を発足させるつもりだと言いました<sup>注3</sup>。日本のタクシーは呼べば比較的正確な時間にくるし、規制の壁もあるので参入は容易ではないでしょうが、便利でおもしろいサービスが日本でも見られるようになるのは楽しみです。またUberのライバルHailoのWebページ<sup>注4</sup>を見ると、サービス提供都市の中に「東京」があって、「東京でのサービス開始は2013年上旬予定(原文ママ)」というメッセージが掲載されています。上陸はこちらが先になるかもしれません。どちらも楽しみです。**SD**

注3) 「TechCrunch Tokyo 2012：スマートフォンでタクシー・ハイヤーが呼べるUberのファウンダー、東京進出を発表」

URL <http://jp.techcrunch.com/archives/20121115tokyo-might-be-ubers-first-asian-launch/>

注4) URL <https://hailocab.com/tokyo/>

# はんだづけカフェなう

## World Maker Faire New York

text : 坪井 義浩 TSUBOI Yoshihiro ytsuboi@gmail.com @ytsuboi

協力 : (株)スイッチサイエンス <http://www.switch-science.com/>

### Maker Faire NY

東京で開催されている“Make: Tokyo Meeting”ですが、より規模の大きなFlagshipと呼ばれるMaker Faireがアメリカでは開催されています。Flagshipとされているのは、カリフォルニアのSan Mateoというサンフランシスコ近郊のものと、ニューヨークで行われているものの2つです。

“Make: Tokyo Meeting”から名前が変わり、今年の12月1、2日に開催された“Maker Faire Tokyo 2012”も比較的規模の大きなイベントですが、上記2つのイベントはさらに規模が大きいです。今回の“Maker Faire New York (写真1)”の数日前には“Open Hardware Summit”というイベントもニューヨークで開催されていたため、ぜひ本場のMaker Faireを見たいということで行ってきました。

海外のMakerたちがMakeするものの中に

### ▼写真1 Maker Faire会場



は、あまり日本のMakerが手を出していない、まさにHardware(金物)と呼ぶべき作品も見受けられます。

たとえば自転車のペダルを漕ぐと、蝶の羽根が動く乗り物を出展している“Bike Zoo(写真2)”、Metrocardと呼ばれるニューヨークの地下鉄のプリペイドカードを貼り付けたロボット“Metrocard Man and Robot(写真3)”、サーバ

### ▼写真2 Bike Zoo



### ▼写真3 Metrocard Man and Robot



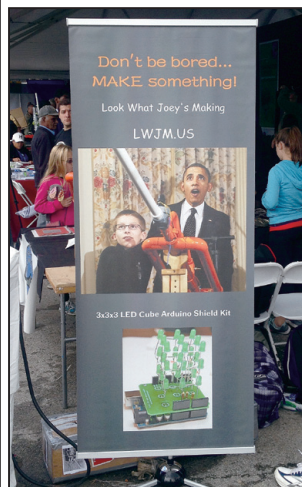
### ▼写真4 Node.JS Tank Robot



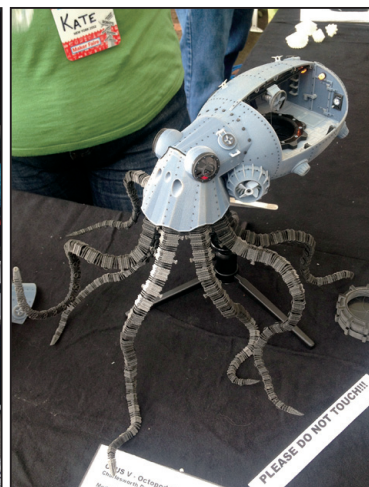
▼写真5 Truncated Icosahedron



▼写真6 Extreme Marshmallow Cannon



▼写真7 Octopod 3D Printed Model



サイドJavaScriptのNode.jsを用いている“Wi-Fiでコントロールできる大きなアルミ製戦車(写真4)”といった具合に比較的大きな出展が目立ちます。

また、Maker Faire New Yorkでは家族連れも多く見られ、子供向けのワークショップだけでなく、Young Makersという“Make:”している子供たちが出展していたりもします。たとえば、Truncated Icosahedronという巨大な切頭二十面体(サッカーボール状の構造)の展示に子供が入って遊んでいた(写真5)、オバマ大統領にマシュマロキャノンを披露した少年がArduinoのシールドのキットを出展(写真6)していました。

## 3Dプリンタ

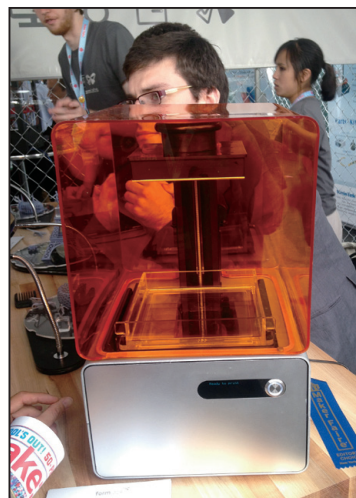
World Maker Faire New York 2012では、今日、日本でも少しずつ注目を集めはじめている3Dプリンタ関連の出展も多くありました。3Dプリンタパビリオンというコーナーが設けられているだけでなく、写真7のような、3Dプリンタで作った作品を展示している人たちも多く見受けられます。また、従来の手頃な価格帯の3Dプリンタはフィラメントと呼ばれる樹脂を融かして積み重ねていく“積層型”と呼ばれる

タイプなのですが、来年には登場する予定の“FORM1”という液体の樹脂をレーザーで凝固させる“光造形”のプリンタ(写真8)で作ったサンプル(写真9)も展示されていました。

以前、オープンソースハードウェアの3Dプリンタとして、MakerBot Industries(以降Makerbot)の製品(第16回と第21回)を紹介しましたが、ほかにもオープンソースの3Dプリンタは存在します。RepRapプロジェクト<sup>注1</sup>は

注1) <http://reprap.org/>

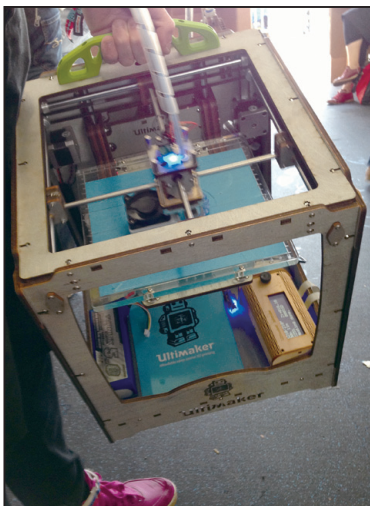
▼写真8 FORM1



▼写真9 FORM1の出力サンプル



▼写真10 ポータブルUltimaker



Makerbotの古巣であり、いくつものモデルの3Dプリンタを公開しています。このRepRapの3Dプリンタも多く見受けられましたし、最新のPrusa i3を開発したJosef Prusaの出展もありました。また、オランダのFabLab生まれのUltimakerというオープンソースの3Dプリンタをバッテリーで動かし、持ち歩いてデモをしているのも見かけました(写真10)。日本のMaker Faireでもこのようにさまざまな3Dプリンタを見るようになる日は近いのでしょうか。

## Raspberry Pi

そういえばこの連載ではまだRaspberry Pi(ラズベリーパイ)を紹介していませんでした。数ヵ月前から話題になっているのですでご存じの方も多いでしょうが、Raspberry Piは、ARMベースのプロセッサを搭載し、Linuxも動く名刺くらいのサイズのマイコンボードです。なんと2,950円で販売されており、アールエスコンポーネンツで購入するのが手っ取り早いでしょう<sup>注2</sup>。

このボードに使われているプロセッサは、H.264のエンコード／デコード機能も搭載しており、HDMI端子も備えています。ですからLinuxを使ってマルチメディア用途に使うこともできるのですが、GPIOというマイコンの汎用入出力端子やSPIといったインターフェースも備えており、USBやEthernetも搭載していますので、相当にリッチなマイコンボードとして利用できます。

ArduinoではC言語ライクな言語での開発を行います。Raspberry Piはボードの上でhttpdを動かしたり、Pythonなどのスクリプトで先述の入出力端子をコントロールしたりできます。もちろん価格が価格ですので、日頃使い慣れたPCと比べると非常に遅いことは覚悟しておいてください。

Maker Faireでは、Fedora ProjectがTuxと一緒に写真を撮ることができるフォトブースをRaspberry Piとデジタル一眼で作って展示していました(写真11)。Raspberry PiでもFreeBSDを動かせないか<sup>注3</sup>と思っているくらいBSD党員の筆者ですが、ついついTuxと一緒に写真を撮ってもらってしまいました。

注2) <http://jp.rs-online.com/web/generalDisplay.html?id=raspberrypi>

注3) 実は動きます。 <http://kernelnomicon.org/?p=164>

▼写真11 Fedoraのフォトブース



▼写真12 Arduinoブースでの筆者



## Arduino

筆者の今回のMaker Faire参加の目的の1つにArduino Teamによる展示のボランティアというものがありました。Arduinoの開発者コミュニティでボランティアスタッフが募集されており、それに応募をしていました。ボランティアはArduinoのブースで“Arduinoとは何か”を説明したり、来場者の質問に答えたりといった作業で、筆者は半日ほどシフトに入って手伝いをしてきました(写真12)。

今回は、Arduino TeamがArduino Dueという新しいボードを発表したため、このボードに関する質問が多いのではないかと身構えていましたが、質問のほとんどはArduinoの始め方や、

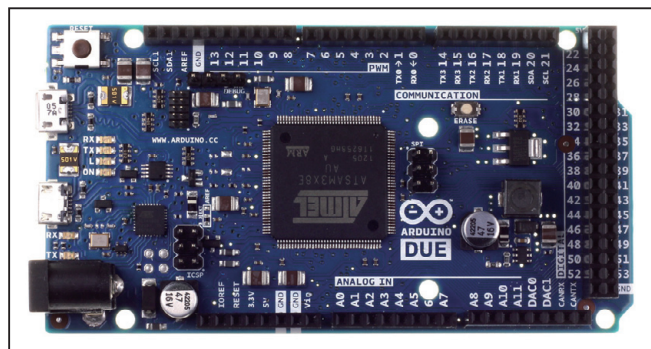
同時に発表されたArduino Starter Kitに関するものでした。筆者は、日本で初対面の人にArduinoの始め方を聞かれた経験がないのですが、このボランティアの間に多くの人からArduinoの始め方を質問されました。始めたいけどどうすればいいかわからない人がこんなに多いということは、日本でもまだまだ広がる余地はあるんじゃないかなと感じました。

## Arduino Due

Arduino Due(写真13)はMaker Faire New Yorkの会場で発表された、新しいArduinoです。従来のArduinoは8bitのAVRプロセッサを搭載していましたが、Arduino Dueは32bitのARM Cortex-M3のプロセッサを搭載し、84MHzのクロックで動作するという製品です。かなりmbedに近いハードウェア性能を実現するようになりました。

プロセッサのアーキテクチャが変わったので、Arduino IDEに内蔵しているコンパイラもavr-gccだけでなく、gcc-arm-none-eabiが追加されています。IDEのバージョンも一気に1.5となりました。SD

▼写真13 Arduino Due





# ニートな pha のぶらぶら日記 ゴシックハウスなう

TEXT=pha pha.japan@gmail.com

第  
33  
回

## 褒めればステマ、 けなせば炎上、 とかくネットは生きにくい

### ステルスマーケティング

ステルスマーケティング(英: stealth marketing)とは、消費者に宣伝と気づかれないように宣伝行為をすること<sup>[1]</sup>。アンダーカバー・マーケティング(英: Undercover marketing)とも呼ばれる<sup>[2]</sup>。

(Wikipedia より)

### エゴサーチの 弊害とか

僕はもうかれこれ10年くらいネットでだらだらとブログを書いていて、最近はすっかり本を出版したりしてしまっただけ、そんなふうに自分の作ったものを不特定多数の人に公表するということをやっていると、他人にどう評価されているかという評判がやっぱり気になったりする。この高度に発達したインターネット社会では自分の名前やブログのURL、書籍の名前で検索すればいくらでも誰かが書いた感想を見つけることができるし、わざわざ検索しなくても直接ブログのコメント欄やTwitterを通じて感想をもらうことも多い。

でもそうすると当然、好意的な感想だけではなく批判的な意見や攻撃

的な書き込みを目にすることもよくある。人によってはこうした批判をもらうと落ち込んでしまうという理由でブログのコメント欄を閉じたりトラフィックを拒否したりすることもある。「(自分に自信がなくなって作品が作れなくなるから)2ちゃんねるの自分のスレは見ない」と言うクリエイターの話もよく聞く。これは誰もが感想を手軽に作者の目の届くところに発表できてしまう時代の弊害なのかもしれない。

僕自身は(「刺しに行きます」みたいな身体に直接的な危害が加えられそうなものでなければ)否定的な意見を目にしてもそれほど気にならなくて、「褒めてる意見ばかりでも気持ち悪いから批判的なものもあつたほうがバランスが取れていいだろう」と思うほうだ。それは多分、僕

が他人の気持ちとか感情とかに鈍感なせいなんだろうけど、これはネットに向いてる性格なのかもしれないと思う。インターネットはよく知らない人の意見を適度にスルーする力が求められる空間なのだ。

### 公正なレビュー空間 の課題

「インターネットによって誰でも手軽に自分の意見を発表できるようになることで、多くの人の意見が集まった集合知というもの生まれ、集合知の力でいろんなものが今までより正しく評価できるようになる」みたいな考えは結構昔からあって、僕も大筋ではそれに賛成なんだけど、でも多くの人が自由に意見を発表できる空間は何も調整せずに放っておいても無条件にうまく機能するものでもないということを最近よく感じる。

たとえばネットで何かを褒めるときだっていろいろと問題が起きたりする。代表的なのが「やらせ」や「ステマ<sup>注1)</sup>」の問題だ。一般ユーザの口コミや集合知による評価が力を持つとやっぱりそれを操作して利益を得ようという動きは出てきてしまうもので、「2ちゃんねるまとめブログで特定の商品やサイトがやたら褒められているけどステマじゃないか」とか、「飲食店評価サイトでやらせの高

2ちゃんねる

掲示板 | imode | 携帯orz | ビューアp2

スレッドタイトル 検索  
本文検索や検索設定はログイン時詳細

ニュース | ヘッドライン | テレビ欄 | 過去ログ倉庫



2ちゃんねるには毀誉褒貶さまざまな情報が集まり続ける。

注1) ステルスマーケティングの略。広告と明示しない広告のこと。

評価レビューがついていた事例」とか、「書籍やアプリのレビューで関係者っぽい不自然なべた褒めのものが並んでいる」みたいな事件をちょくちょく目にするようになった。

僕自身も、ブログで何かの商品とかWebサービスを褒める内容の記事を書いたときに、「これを書いたことでいくらもらってるんだろう」というようなコメントをもらうことがときどきある。単にそれがいいと思ったから褒めただけでお金なんてもらってないのに……(くれるならもらってもいいけど)。でも確かに昔より広告とわかりにくい広告が増えてきているので、そういう疑問が出やすくなるのはしかたないのかもしれない。

褒めるのと逆のけなすほうになるとさらにトラブルが多くなる。良くないものに対して良くないという批判がちゃんと出るのは健全なことなんだけど、そうした常識的な範囲を超えて、一部の粘着質のモンスターユーザみたいな人が執拗に攻撃を繰り返すみたいなことになってしまうと、そういう人に限って精力的に多数のレビューを投稿するので、少数の力で極端に評価が偏ってしまうことがある。そうした場合は攻撃を受ける側の心理的ダメージも大きくなってしまいます。また、批判が盛り上がって2ちゃんねるなどの匿名コミュニティに火がついてしまうとさらにたいへんで、多数の匿名の人が無差別にその対象の関係者に攻撃を加え始めて、大規模な炎上事件になってしまい、現実生活にも大きな被害を及ぼしたりする。

### インターネットという集合意識

ただまあ、ステマとか炎上とかそうした問題は残しつつも、大筋としては今のネット上の評価システム

はそこそこうまく回っているのではないと思う。何かを選ぶときにネットでの評価を参考にすることは日常的になったし、明らかに質の良くないものが不自然に持ち上げられていたらそれを正当に批判したり、無名だけどクオリティの高い作品を拾い上げて流行らせたりするという機能も、ネットは今でも果たしている。やらせやステマみたいな公正な評価を狂わせるノイズは、ネットが登場する前の雑誌やテレビの時代から存在したことだ。匿名集団の炎上による被害はもうちょっととなかなかならないものかと思うけれど。

どんなコンテンツでも、100%褒める人ばかりとか100%けなす人ばかりということはあまりなくて、どちらもほどほどに存在するものだ。だから、全体としてプラス評価とマイナス評価のバランスが取れていれば、個々の意見はあまり気にしないでもいいと思う。しかし、それでもどこかの誰かが1人でも自分に対して否定的で攻撃的な意見を投げけると、気にして落ち込んでしまうのは人間の習性なのかもしれない。例えば今まで1万年くらい、人間はどこかの見知らぬ人が自分に対して悪口を言っているのが簡単に目に入ることのない世界に生きていたわけだから、そんなにすぐに習性は変わらないのだろう。生まれたときからネットが高度に発達している子どもたちが大人になるころには変化するのだろうか。

僕は個人的には、ネット上で褒める感想や批判する意見をもらったとしても、継続的なものではなく単発で知らない人からコメントをもらっただ

けなら、あまり個々の人間として意識しないようにしている。そうした場合にいちいち相手のバックグラウンドを慮って一対一の人間として相手をしていると、自分がキャパオーバーで壊れてしまう。

イメージとしては、インターネットは個々の人間で構成されているのではなく、ネット上には人間の意識が集まって形成された、不定形だがよぶよぶしてもややとした巨大な集合意識の塊みたいなものが存在しているように想像している。毎日毎日地球上の無数の人がネット上に気持ちや感想をつぶやき続けることで集合意識は維持されている。それで僕が何かブログをネット上に発表すると、ある内容だとその集合意識の右上のほう喜んでように青く光って脈動を始め、ある内容だと集合意識の左下のほうが怒ったように赤く膨張してこちらに棘のようなものを飛ばしてきて、それが批判コメントやAmazonの辛辣レビューや2ちゃんねるの匿名書き込みとして具現化する。ネットでマス(大量の人間の集団)を相手にするときは、見知らぬ他人のことをそれくらいに考えておいたほうが生きやすいように思う。

SD

#### カスタマーレビュー



pha著『ニートの歩き方』(技術評論社)のAmazonでのレビュー状況。

See what's happening **right now**

Type in operators for advanced search.

Trends  
#TUTUYAZHONZ #ベビークン Promoted 光緒の日記 #なガリタンの誕生日だし思い出されよ 三連珠 ベストヒット歌謡集  
東方神起の日 地上波 ジャンプフェス agnema

Twitterの検索画面。1日5回くらい自分の名前などで検索しよう。

# PRESENT

## 読者プレゼントのお知らせ

『Software Design』をご愛読いただきありがとうございます。本誌サイト <http://sd.gihyo.jp/> の「読者アンケートと資料請求」からアクセスし、アンケートにご協力ください。ご希望のプレゼント番号をご入力いただいた方には抽選でプレゼントを差し上げます。締め切りは **2013 年 1 月 17 日** です。プレゼントの発送まで日数がかかる場合がありますが、ご了承ください。

ご記入いただいた個人情報、プレゼントの抽選および発送以外の目的で使用することはありません。アンケートのご回答については誌面作りのために集計いたしますが、それ以外の目的ではいっさい使用いたしません。ご入力いただいた個人情報は、作業終了後に当社が責任を持って破棄いたします。なお掲載したプレゼントはモニター製品として提供になる場合があります。当選者には簡単なレポートをお願いしております（詳細は当選者に直接ご連絡いたします）。

# 01

1 名

## ScanSnap iX500 Deluxe

ワンタッチでスピーディーに書類を PDF 化できるパーソナルドキュメントスキャナ「ScanSnap」シリーズの新製品。Wi-Fi 接続によるスマートフォンやタブレットへの直接転送や、25 枚/分の高速読み取りを実現しました。

提供元 PFU

URL <http://scansnap.fujitsu.com/jp>



# 02

3 名

## 弥生会計 13 スタンダード



使いやすさを追求した誰にでもすぐに使える会計ソフト。シンプルな操作であらゆる業種の個人事業、小規模法人の会計業務を効率化します。法人決算、個人決算（青色申告／白色申告）にも対応。

提供元 弥生

URL <http://www.yayoi-kk.co.jp>

# 03

1 名

## USB 紙コップ ウォーマー



紙コップの飲み物を保温してくれるカップホルダー型のヒーター。付属の USB ケーブルを PC などに接続してスイッチ ON！ いつでも温かい飲み物を飲むことができます（製品に紙コップは付属しません）。

提供元 サンコー

URL <http://www.thanko.jp>

# 04

## ソーシャル もうええねん

村上 福之 著／  
四六判、192 ページ／  
ISBN = 978-4-904899-33-5



総アクセス 2200 万 PV をたたき出したエンジニアブロガー村上福之の総裁による、ソーシャルメディアの波に立ち向かうための思考術。この 1 冊でいかに「ソーシャル」が危ういものかわかります。

提供元 ナナ・コボレート・コミュニケーション URL <http://www.nana-cc.com>

# 05

## C++ のための API デザイン

マーティン・レディ 著、ホジソンますみ 訳、  
宅 陽一郎 監修／B5 変形判、528 ページ／  
ISBN = 978-4-7973-6915-1



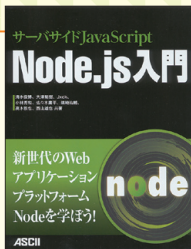
拡張性がない、効率が悪い、安全でないなど、正しく動くプログラムにも設計センスの良し悪しは明確に存在します。長期の運用に堪える「良い」設計について、具体的事例とともに詳説します。

提供元 ソフトバンククリエイティブ URL <http://www.sbcr.jp>

# 06

## サーバサイド JavaScript Node.js 入門

清水 俊博、大津 繁樹、Jxck、小林 秀和、  
佐々木 庸平、篠崎 祐輔、高木 敦也、西山 雄也 著／  
B5 変形判、584 ページ／  
ISBN = 978-4-04-870367-3



今最も注目される技術の 1 つ「Node.js」についての解説書。「Node とは何か？」という基本的なことから「Node の内部がどうなっているのか知りたい」という専門的な内容まで幅広く取り上げます。

提供元 アスキー・メディアワークス URL <http://asciimw.jp>

# 07

## サーバ・インフラ 構築・運用完全ガイド

笠野 英松 著／  
A5 判、488 ページ／  
ISBN = 978-4-7741-5408-4



異種複数のプラットフォームによるネットワークシステムを高信頼で堅牢に構築し、さまざまな削減をはかるためのインフラ技術について解説。実装から運用管理まで 1 冊でわかる完全ガイドです。

提供元 技術評論社 URL <http://gihyo.jp>

新年号特別付録!!

# 法輪寺電電宮 情報安全護符シールの使い方

新年号の特別付録として、京都府嵐山にある虚空蔵法輪寺の協力のもと、ITエンジニアにとって命の次に大事とも言える情報(データ)を守る護符シールを制作しました。法輪寺ご住職自ら開眼供養していただいたお寺のお墨つき。サーバ、PC、スマートフォンなどいろいろなものに貼ってお使いください!

## ●電電宮とは

13歳になった少年少女が虚空蔵菩薩に智恵を授かりに行く「十三まいり」で有名な虚空蔵法輪寺。その境内に、寺の鎮守のために建立された神社が「電電宮」です。ここには、大空(宇宙)の自然現象を広く包含する本尊虚空蔵大菩薩に関連した明星天子、雨宝童子、電電明神などが祀られています。中でも電電明神は雷、稲妻の神様で、今ふうに解釈すれば電気祖神、電波祖神と言えます。このことから長年、電力/電機/電波/電子各業界の事業者、事業の成功や、電気/電波業界の発展を祈る場として参詣されています。

虚空蔵 法輪寺  
京都市西京区嵐山虚空蔵山町  
<http://www.2ocn.ne.jp/horinji>



電電宮本殿



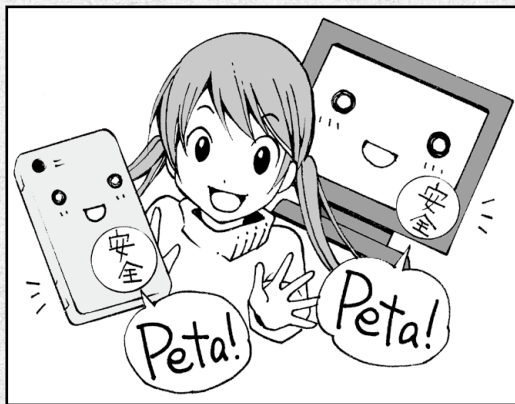
シールのデザインデータが入ったCD-ROMを開眼供養し、そのデータで制作したちゃんとしたお守りです。

## ■法輪寺住職 藤本高全氏よりひとこと

法輪寺の本尊虚空蔵菩薩、並びに鎮守社電電宮は千年以上の歴史があり、京都をはじめ全国の方々にご信仰して頂いております。このたび、虚空蔵菩薩の御宝前において厳肅に祈願させていただきました。どうか人々を幸せにするようなすばらしいプログラムを設計していただき、またバグなど発生せず、大切なデータが不慮の事故で消失しないよう御加護をお受けください。

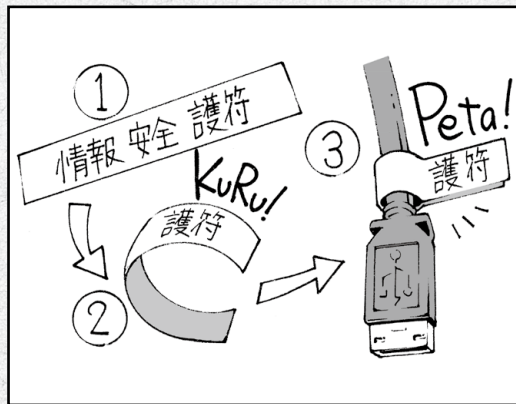
## シールの 使用例

**丸** 型のステッカーはサーバ、PC、スマートフォンに貼ってお使いください。剥がしても跡が残りませんので、会社から支給されたPCにも安心して貼れます。



イラスト：マツダアヤコ

**長** 方形のステッカーはケーブル用です。バックアップサーバの電源ケーブルやファイルサーバのLANケーブルなど大事なケーブルに貼れば、目印としても有効です。



# BACK NUMBER バックナンバーのお知らせ

## バックナンバー好評発売中！常備取り扱い店もしくはWebより購入いただけます

本誌バックナンバー（紙版）はお近くの書店に注文されるか、下記のバックナンバー常備取り扱い店にてお求めいただけます。また、「Fujisan.co.jp」(<http://www.fujisan.co.jp>) (<http://www.fujisan.co.jp/sd>) や、e-hon (<http://www.e-hon.ne.jp>) にて、Webから注文し、ご自宅やお近くの書店へお届けするサービスもごさいます。



2012年12月号

**第1特集**  
判断をおく／経緯を説明する／手順の理解を得る  
文章を書くためのアタマの整理術  
**なぜエンジニアは文章が下手なのか？**

**第2特集**  
高速・高性能 HTTP サーバ  
**Nginx 構築・設定マニュアル**

**一般記事**  
・エリデープログラマの発想と実践

1,280円



2012年11月号

**第1特集**  
**もし、OpenFlow でやれと言われたら？**  
SDN、仮想化でネットワークはどうなる

**第2特集**  
サーバの運用支援に  
グラフィカルなリソース監視ツールを！  
**Munin が手放せない理由**

**一般記事**  
・SpeedSilverBullet とは？

1,280円



2012年10月号

**第1特集**  
サーバ管理自動化の恩恵とリスクを見直しませんか？  
**Chef 入門**

**第2特集**  
Ipr コマンドが動く裏側の仕組みがわかる！  
**Linux プリント環境の教科書**

**一般記事**  
・SSH 力をつけよう！  
・JSX 入門 [前編]

1,280円



2012年9月号

**第1特集**  
理解の壁を乗り越える Final Answer！  
**C言語のポイントは必要ですか？**

**第2特集**  
セキュリティ向上をあきらめない管理者になる！  
**SELinux を無効にしない理由**

**一般記事**  
・機械学習ライブラリ「Mahout」入門 [後編]

1,280円



2012年8月号

**第1特集**  
いま読んでおくべき本はどれ？  
**エンジニアのパワーアップ読書**

**第2特集**  
いま改めてお勧めする OS はこれだ！  
FreeBSD、Debian、Ubuntu、CentOS、Gentoo

**一般記事**  
・機械学習ライブラリ「Mahout」入門 [前編]

1,280円



2012年7月号

**第1特集**  
エンジニアはなぜエディタにこだわるのか？  
**Vim と Emacs**

**第2特集**  
ソフトウェア開発はセンバ・ファルシムス！  
**もしも、新卒女子SEが「アジャイル」をマスターしたら**

**一般記事**  
・データセンターの作り方  
・パケットキャプチャの作り方 [後編]

1,280円

Software Design    バックナンバー常備取り扱い店								
北海道	札幌市中央区	MARUZEN& ジュンク堂書店 札幌店	札幌店	011-223-1911	神奈川県	川崎市高津区	文教堂書店 満の口本店	044-812-0063
	札幌市中央区	紀伊國屋書店 札幌本店		011-231-2131	静岡県	静岡市葵区	戸田書店 静岡本店	054-205-6111
東京都	豊島区	ジュンク堂書店 池袋本店		03-5956-6111	愛知県	名古屋市中区	三洋堂書店 上前津店	052-251-8334
	新宿区	紀伊國屋書店 新宿本店		03-3354-0131	大阪府	大阪市北区	ジュンク堂書店 大阪本店	06-4799-1090
	渋谷区	紀伊國屋書店 新宿南店		03-5361-3315	兵庫県	神戸市中央区	ジュンク堂書店 三宮店	078-392-1001
	千代田区	書泉ブックタワー		03-5296-0051	広島市南区	ジュンク堂書店 広島駅前店		082-568-3000
	千代田区	丸善 丸の内本店		03-5288-8881	広島県	広島市中区	丸善 広島店	082-504-6210
	中央区	八重洲ブックセンター本店		03-3281-1811	福岡県	福岡市中央区	ジュンク堂書店 福岡店	092-738-3322
	渋谷区	MARUZEN & ジュンク堂書店 渋谷店		03-5456-2111				

※店舗によってバックナンバー取り扱い期間などが異なります。在庫などは各書店にご確認ください。

# DIGITAL

## デジタル版のお知らせ

## デジタル版 Software Design 好評発売中！紙版で在庫切れのバックナンバーも購入できます

本誌デジタル版は「Fujisan.co.jp」(<http://www.fujisan.co.jp/sd>) と、「雑誌オンライン.com」(<http://www.zasshi-online.com/>) で購入できます。最新号、バックナンバーとも1冊のみの購入はもちろん、定期購読も対応。1年間定期購読なら5%割引になります。デジタル版はPCのほかiPad / iPhoneにも対応し、購入するとどちらでも追加料金を払うことなく、読むことができます。

Webで  
購入！



家でも  
外出先でも

## 第1特集

# いざというときに備える システムバックアップ

ある日、PCの電源を入ると、ディスクから異音がして起動しない……。いろいろな思いが交錯します。あの大事なメール、住所録、銀行のオンラインパスワード、再インストール、……。個人用でも修復が大変です。24時間365日稼働しているサーバだったら、と思うと日頃のデータのバックアップ、データ復旧の重要性が身にしみます。まして今やクラウド環境の利用が当たり前になり、手元にないインスタンスをどうやって管理するのか。本特集は、データのバックアップの基礎を復習しながら、クラウド環境でのバックアップの現状を解説します。そもそもハードディスクがデータをどのように保存しているのか、故障の原因をしくみから振り返り、極限まで破壊されたハードディスクの専門家による修復をレポートします。サーバで使われることが多くなってきたFlashストレージの場合の導入の注意点も言及します。コンピュータはデータが命！

第1章	バックアップ今昔、昔テープ今クラウド	20
	伊勢 幸一 (ISE Kouichi)	
第2章	UNIXバックアップ系コマンド復習	28
	石田 つばさ (ISHIDA Tsubasa)	
第3章	クラウドのデータはどこにいくの?	34
	[Nifty Cloud編] 大久保 智之 (OKUBO Tomoyuki)	
第4章	クラウドでのバックアップの基本 [AWS編]	42
	柳瀬 秀明 (YANASE Hideaki) / 宮澤 慶 (MIYAZAWA Kei)	
第5章	知っているつもり? ハードディスクのイロハ	50
	杉田 正 (SUGITA Tadashi)	
第6章	復旧業者に聞く データ復旧の実際	62
	協力: (株)ワイ・イー・データ オントラックデータリカバリーサービス	
第7章	安定運用を目指すための Flashストレージ導入ポイント	70
	長谷川 猛 (HASEGAWA Takeshi)	

## 第1章

# バックアップ今昔、 昔テープ今クラウド

伊勢 幸一(ISE Kouichi) データホテルCTA TwitterID: @ibuchou  
イラスト: 高野 涼香

## 予期せぬ システムダウン

いつものようにオフィスで端末に向かい、Excelとパワポを開いて来期の予算策定をしていたところ、デスクの向こうで「うっ」という嘔み殺したような呻きとも嗚咽とも思えるような声がし、そちらに視線を向けるとエンジニアスタッフのN(エヌ)君が青白い顔をこちらに向け、震える声で呟いた。

「Wind7が全落ちしてます。ユーザデータがすべて吹っ飛んだっばい……」

Wind7とは会社の他部門が1年前にサービスインしたばかりのパブリッククラウドサービスである。私S(エス)は直接関係していないが、N君は設計の段階で部分的にかかわっていたはずだ。

「N君、ストレージ側の障害か？ ストレージならミラーがあるだろ」

「直接担当していなかったので詳細はわかりかねますが、ディスク単体の故障ではなくファイルシステムの論理情報自体がおかしくなったようで、当然ミラー側も一緒にやられています」

「ほかにバックアップは？」

「顧客管理関連のバックアップにはテープライブラリを使ったシステムがありますが、ユーザデータは対象外です。今クラウドサービスのストレージは数十TBに達しようとしています、こんな破壊的なサイズのバックアップってどうすればいいんでしょう」

申し遅れたが、私Sはベンチャー系データセンタ事業者に20年以上勤めるサラリーマンである。もともとはエンジニアであったが10年ぐらい前から管理職に就き、すでに現場を離れて幾久しい。

通常ストレージのバックアップはHDD(ハードディスクドライブ)やストレージサブシステムの機械的故障によるデータ消失を前提としているが、実際にはハードウェア障害よりオペレーションミスやアプリケーションの不具合によって能動的にデータ破壊を引き起こす場合が多い。すると原因の想定がまったく異なるため、それに起因するデータ消失にバックアップは無力なのである。

N君に知り得る限りの事情を聞いてみると、Wind7は分散型ブロックデバイスストレージを共有ファイルシステムとして全クライアントからネットワークマウントして利用する形式のようだった。ストレージはネットワークを介したRAID-1(ミラーリング)のような構成をしており、一応セカンダリのみラーメンバがバックアップ



ブの役割を果たすというアーキテクチャなのだ  
そうだ。

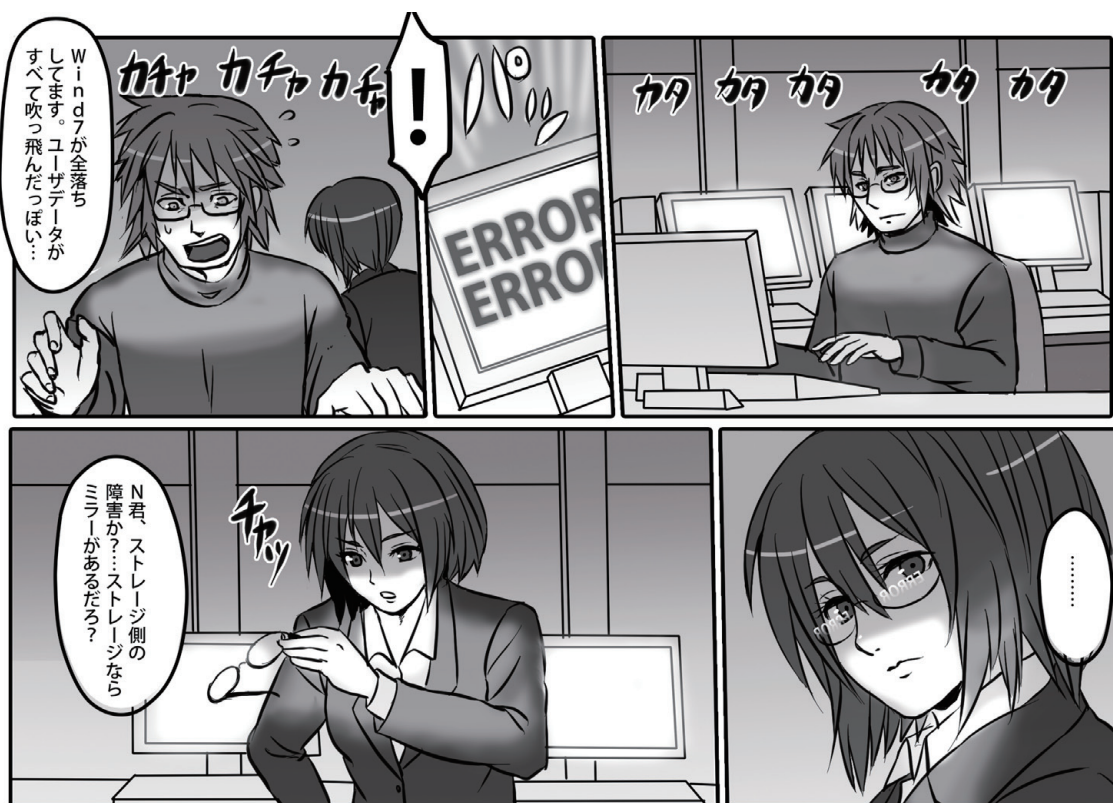
通常、ディスクドライブ単体の障害であれば、  
メンバ内のRAID機能が有効となり、プライマ  
リメンバ全体に障害があってもセカンダリメン  
バによって障害時とほぼ同じ状態を維持するこ  
とはできる。しかし、今回はハードウェア障害  
ではなくそのファイルシステムをマウントして  
いるクライアント側のファイルロック機能が誤  
動作をし、それが<sup>ふくそう</sup>輻輳してファイルシステム情  
報領域に致命的な不整合が発生したようだ。デー  
タの復旧はファイルシステム情報を基にデータ  
ブロックを連結してリカバリするしかないが、  
それは気の遠くなるような手間である。時間を  
かけて復旧したとしても元どおりに復元された  
かどうかの保証もなく、復元したデータがユー  
ザにとって不必要な物になってしまっている可  
能性は非常に高い。

——今日はこのWind7の大規模障害で周囲は騒  
然とし、メールが飛び交い電話が鳴り響いてま  
ともに予算策定などできる気配ではない。そこ  
で私SはN君にバックアップについて長い長い  
話をしてみることにした。



## そもそもバックアッ プとは何か？

データのバックアップというのはコンピュー  
タにHDDが装備されてから発生した作業だ。私  
Sの学生時代、計算機センターの大型汎用機に  
は当然記憶デバイスとしてHDDが実装されて  
いたが、所属する研究室のミニコンやパソコン  
にはまだディスクが搭載されていなかった。当  
時のコンピュータとはCPUとメモリからなる本  
体、キーボードとCRT(またはテレタイプ)、そ  
してテープやフロッピーディスク(FD)などのド  
ライブとデータが記録されているメディアとい  
う外部ストレージで構成されている。PCの場

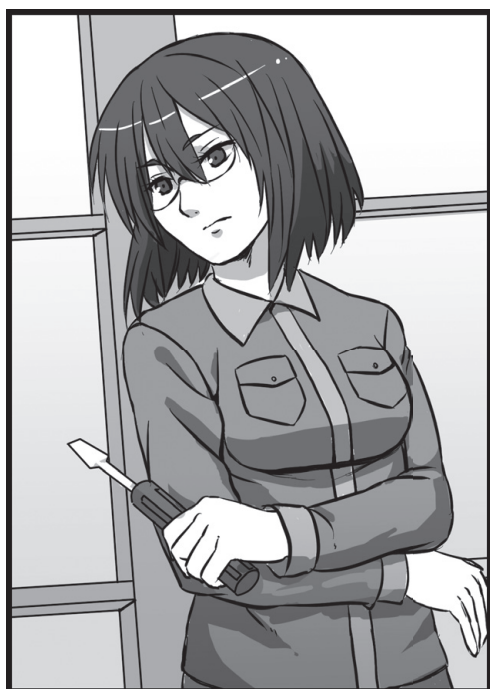


※コミック部分は右から左へ読んでください。

合、起動時にFDからシステムをロードし、その状態でデータやプログラムを入力した後、システムFDを別のデータ用FDに差し替え、それにデータをセーブするという七面倒くさい作業をしていたものである。つまりメモリ上で作成されたデータのセーブ先がバックアップそのものだった。

汎用機データは計算機センターの職員がバックアップしていたのでユーザである私達はバックアップ作業をする必要がなかった。しかしミニコンやパソコン、そしてUNIXワークステーションという部門別に設置されるコンピュータにもHDDが搭載されてからユーザ側でデータのバックアップをする必要性が生まれてしまった。

その頃、PCのディスクは数MBであり、8インチや5インチのFDの容量は1MBに足りない程度でディスクのバックアップというよりは個人ファイルのバックアップに過ぎなかった。UNIXワークステーションは数十MBのディスク容量がありFDなどでは全然足りず、数十M



から百数十Mの容量を持つ1/2インチオープンリールテープか、1/4インチカートリッジテープにバックアップを取っていた。



## UNIXのバックアップコマンド

UNIXにはバックアップ用のコマンドが標準実装され、ファイルシステム単位でバックアップするdumpと、ディレクトリ単位でアーカイブするtar、あらかじめファイルリストを作成し、そのリストを基にファイル単位でバックアップするcpioがあった。tarやcpioは利用者が個人的にアーカイブするための用途であり、システム管理者側はどのファイルがどのユーザにとって必要なかわからないため、ファイルシステムごとバックアップするdumpを使っていた。しかし、dumpはファイルシステム情報を基にファイルを構成するデータブロックをシリアルライズするため、バックアップ中にデータを更新されたり作成されたりするとファイル情報とデータブロックとの間に不整合が生じる。したがってシステムをシングルユーザモードにし、ユーザが利用できない状態で実施する必要がある。

そのあと、ディスクの容量が指数関数的に増加し、1本のテープで1つのファイルシステムが収まりきらなくなりはじめた。もともとdumpやtarはマルチボリュームをサポートしているため複数のテープに1つのファイルシステムをバックアップすることは可能だが、テープ交換するまで端末の前で立ち会わなければならず、またそれが何十本にもなるとどのテープがどのファイルシステムなのかが管理しきれなくなってきた。



## テープバックアップメディアの歴史

バックアップ中、端末に張り付いているのは退屈な時間であり、また1つのファイルシステムを復元するのに関連する何本ものテープを探し当てて集めるのもやりきれない。そんな中、



ディスクの容量を超えるテープメディアが出現した。ストレートトラックではなくオーディオ用のヘリカルトラックテープだ。もともと音声や映像といったアナログデータを記録するためのテープであるが、それをデジタルデータ記録用に流用したものである。テープには8mm、4mmDAT、そしてBetaCamテープというバリエーションがあった。記憶容量も数GBと当時のディスク容量よりも大きかったのでディスクドライブがまるごと1本のテープに収まることでテープ交換作業が不要になり、システム管理者は嬉々として8mmテープドライブやDATドライブを導入した。

しかし、この頃利用されていたヘリカルトラックテープには致命的な欠点があった。これらのテープを長期間保存しておくとしるに巻き付いているテープ間の磁気干渉により、データが読めなくなってしまうのだ。磁気干渉自体はストレートトラックタイプのテープにも生じる。しかし、容量を上げるためヘリカルスキャン型のトラック密度はストレートトラックに比べて非常に高く、磁気干渉による影響をより多く受けてしまうのだ。オーディオやビデオなどのアナログ用途であれば多少のノイズとなって現れるだけだが、デジタルデータ用途の場合、干渉によるノイズはデータ破壊を意味する。

後に比較的強靱なテープメディアとしてDLTが利用されるようになる。DLTはもともとDEC社のミニコンOS用ディストリビューションに利用されていたものだが、薄膜フィルムの8mmや1/4カートリッジテープに比べ、テープ強度が高かったため繰り返しの読み書きにおけるテープ劣化が小さかった。もともとほかのメディアに比べて容量的に劣るためバックアップメディアとしてはあまり利用されていなかったが、1990年代後半から数十GBから百数十GBまで容量がアップしたことでバックアップメディアとして利用可能になってきたのである。

バックアップテープの容量が驚異的に増えたことは別の問題を引き起こした。大容量テープドラ

イブは書き込み速度もある程度高速化されていたが、それでもデータをまるごとバックアップするには数時間の時間を要したため、ユーザがコンピュータを利用できない時間がさらに長くなってしまったのである。そこでユーザが利用中でもバックアップのできるオンラインバックアップアプリケーションやテープライブラリサブシステムが商品化され販売されるようになる。



## テープバックアップの落とし穴

これでバックアップ作業、復旧するためのテープの探索収集作業からシステム管理者は解放されるはずだった。しかし、それは儚い<sup>はかな</sup>幻想でしかなかった。

これらの自動バックアップシステムはバックアップしたログのデータベースが巨大化し、その巨大化したデータベースに不整合が発生するとまったくバックアップが役に立たないのだった。そのため、常にバックアップの終了時にデータベースもテープに落とすのだが、データベー



ストレージに障害が発生したとき、どのテープにデータベースが格納され、どのテープを集めれば最終的にデータベースを復旧できるのかはどこにも情報がない。したがって、すべてのテープを順番に読み出してバックアップデータベースを復旧しなければならない。テープライブラリ内に100本以上のカートリッジが格納されていると、データベースを復旧するだけで数日かかる。その間、ユーザは一切前のデータにアクセスできないのだ。ユーザとしてもデータが戻るまで何もせずに遊んでいるわけにはいかない。復旧作業中でもユーザは作業を進め、プログラムを最初から書き直したり、データを再製作したりして失われたデータを自力で作直し、データがバックアップテープから復旧された時点では新しく再生成した別のデータがアクティブとなっているわけである。ディスク容量が巨大化したこのころ、テープに取ったバックアップはバックアップの意味を成さなくなった。



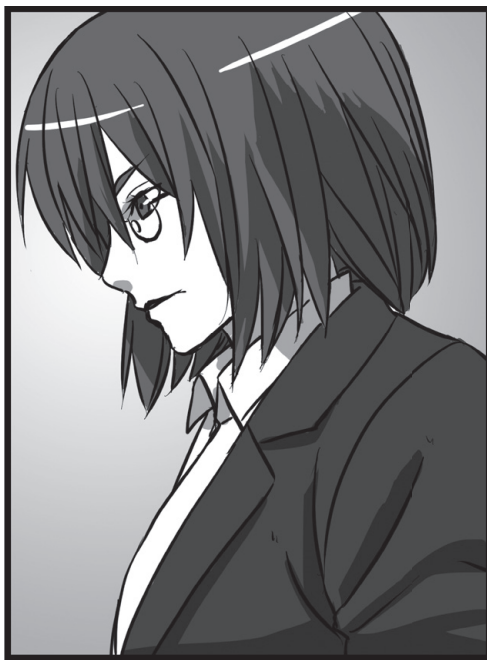
## HDDバックアップはインターネット普及のせい

2000年以降、インターネット上でWebサービス旋風が巻き起こった。従来は企業の広報的ホームページやわずかなe-Commerceサイトだけがインターネット上でのコンテンツだったが、ソーシャルメディアの台頭によるWeb2.0系サービスサイトが乱立し、インターネットの様相が一変した。基本的にコンシューマ向けのこのサービスは24時間365日無停止のサービスであることが大前提だ。当然ながら、従来の企業内業務システムのように週単位で定期的なメンテナンス時間を設けて実施するバックアップでは話にならない。オンラインバックアップであってもディスクのデータを読み出し、シリアルライズしてテープメディアにセーブするにはディスクに対して一定以上の負荷をかける。その間、データアクセスの性能は著しく低下し、24時間365日常に同じレスポンスが期待され

るサービスでは通用しない。

万が一アクティブ側ディスクに障害がありバックアップからデータを復旧するとしても、何時間も何日もかかっている間は、その間にユーザは別のサイトに移ってしまい、サイトが復旧したときにはすでにユーザは離れてしまい、再びサイトへ戻ってくることは期待できない。長時間の復旧作業はコンシューマサービスにおいては致命的な欠陥なのだ。

そこでディスクのバックアップはテープではなくディスクに行くようになった。それもファイルシステムやディレクトリツリーをシリアルライズして保存するのではなく、アプリケーションからアクセス可能な状態のまま複製を作るのである。テープバックアップのように障害時に何時間もかけてデータを復旧するような運用はすでに許されない。そこでバックアップ側も即時アクティブストレージとして利用可能な状態を保ち、障害時にマウントポイントに差し替えるだけで最低限のサービス再開が可能となるようにした。





## スナップショットの 誕生と課題

しかし、複製とはいえ、アクティブ側のストレージをオンラインに保ったまま複製することに問題がある。なぜなら、複製している途中でそのファイルに更新が入ると、複製側のデータに不整合が発生するからだ。それほど頻繁に起こり得ることはないが、逆にいつ発生するかは管理者にとって予測できないため、顧客情報や課金決裁ログなど非常に重要なファイルの複製に不整合が発生する可能性もある。そのため、複製する前に一度データブロックをロックし、ロックしたブロックへの追記や更新には別のブロックを新規で割り当て、ロック時点でのデータに不整合が発生しないようにするスナップショットという機能が現れた。これを使うと非常に短時間で既存のファイル属性ブロックとデータブロックが読み出し専用ブロックとしてロックされ、その状態で複製を行えば、少なくともロックした時点での不整合は発生しない。

だが、これにも問題はあった。スナップショッ

トを取る処理はファイルシステムインデックス部(UFSでいうiノードリスト)だけへのアクセスなので比較的短時間で実行されるが、スナップショットしたデータブロックを複製するには、すべて、もしくは差分のブロックを読み出し、複製側へコピーしなければならない。これは従来型のオンラインバックアップと同じくアクティブ側ストレージに負荷を与えてしまうこととなり、その間のアクセスレスポンスは著しく低下する。24時間365日サービスの場合、一番アクセス頻度の低い時間帯を選んで実行することになる。しかし、サービスをワールドワイド展開することを考えると、各国の時差が入り混じり、サービスの閑散時間というものが存在しなくなり、どの時間帯を利用するとユーザにもっとも不快感をあたえないのかを精査することが難しくなってくる。そこで、複製側へのデータ転送量をシェーディングすることでディスクへの負荷を軽減する方策が取られたが、それは同時に複製時間を長時間化してしまう結果となり、複製が完了した時点と、現状の状態との乖離が大きくなることを意味する。そのためスナップショット用に別のコントローラに接続された別のストレージを利用したり、ネットワーク分散型のファイルシステム上で別のホストに接続されているストレージを使うなどしてスナップショットの複製負荷がアクティブストレージに影響しない方策が考えられていた。

ただし、これだけでも十分とは言えない。これからのバックアップは、データが作成される時、もしくは更新された直後に複製されるようなプラットフォームを開発する必要があるかもしれない。SQLレプリケーションのように、更新されたデータを複製するのではなく、そのデータを作成するSQLクエリをコピーして同じ処理を別のストレージに適用して複製を作成するのも有効だろう。SQLだけではなくアップロードされたデータをそのままディスクに反映するのではなく、キューシステムにいったん格納し、キューイングをトリガーにして別のワーカーが



複数のストレージに書き込むようなエージェントを実装することも有効だ。いずれにしても従来どおりの標準コマンドやOS、サブシステムが提供する機能だけではなく、システム管理者がプラットフォームやプログラムレベルで改善、効率化していくことが必要となってくるだろう。

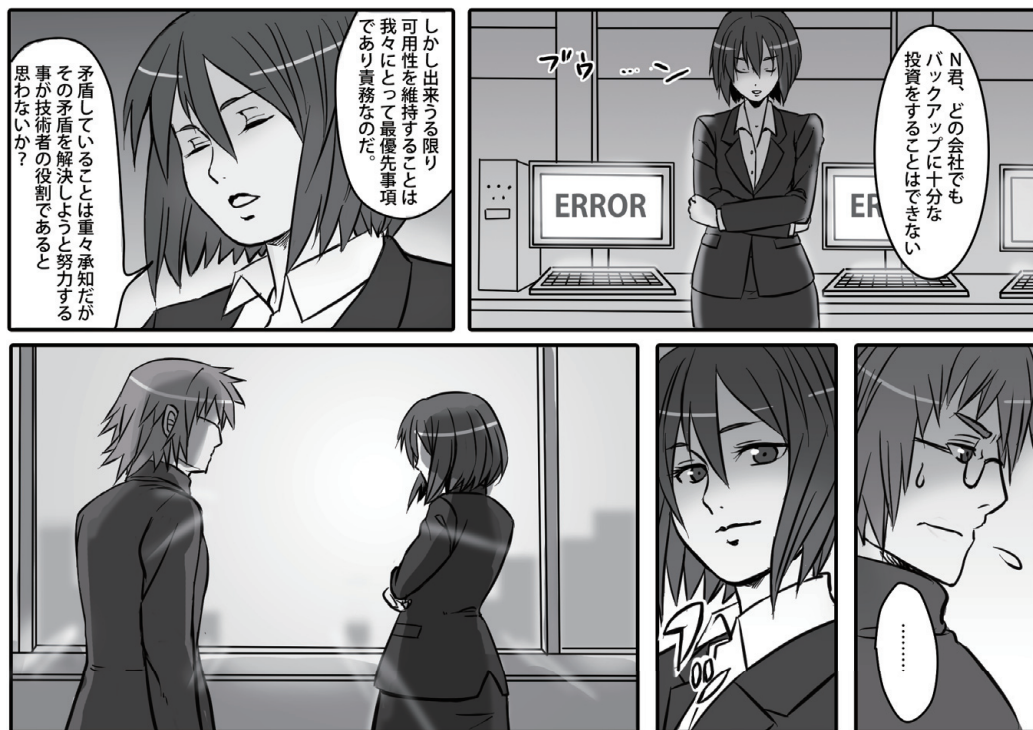


## クラウドバックアップの課題とは

個人PCのローカルディスクであればクラウドストレージサービスにrsyncなどでバックアップを取っておくことも可能であろう。しかし、クラウドサービス事業者向けのクラウドサービスというものは存在せず、あったとしても自社サービスのユーザデータを他社のクラウドサービスでバックアップするというのは有り得ない。データの保全性が他社サービスのクオリティに

依存することとなり自社サービスの品質を自分達の努力で保証できなくなる。

このように安全かつ負担の少ないバックアップ運用というのは簡単なものではない。さらに、バックアップは障害が発生しない限り無用の長物であり作業者もそれほど意識を高く持つこともできぬだろう。経営サイドとしても経営者自身が過去に致命的なディスククラッシュなどの障害を経験していない限り十分な投資を行う判断はできないものだ。しかし、障害があったとき、どんなに後悔してもどうにもならない。今回のWind7の件がそうである。おそらくデータは復旧しない。すべてのユーザのデータは消えてなくなってしまった。ユーザへの賠償責任だけではなくほかの顧客の信頼や社会的信用も失い、今後の事業展開への悪影響まで考慮すると、投資以上の損害を受けてしまうのは間違いない。



私Nが入社した時点で上司Sは管理職だった。意外にもその前はバリバリの運用技術者だったのだ。「矛盾を解決するのが技術者だ」などと、現場を知らず見聞きしたことだけでもものを言う管

理職の言葉であれば聞く耳など持たないが、これだけ運用の現場を知り尽くしている上司に言われると、私Nは素直に頷くしかほかはなかった。**SD**



### ▶ データ復旧の現場から

データ復旧を依頼されたハードディスクドライブ(HDD)で、最も多い障害をご存じでしょうか。それはリードエラーです。そして、それは通常使用の環境下では再現性が高いので、バッドセクタとして扱われるのが一般的です。しかし、技術力の高いデータ復旧業者が作業すれば、分解も部品の交換もせずにデータを取り出せます。つまり正しくはバッドセクタではないのです。

### ▶ そもそもリードエラーとは

リードエラーは物理障害の中で約5割を占めます。発生要因は種々多くありますが、HDD内部には自動的に解消を試みる「リードリトライ機能」が備えてあります(それはメーカー特許で保護されている)。たとえば次のようなものが挙げられます。

- ①ヘッドが正規のトラックをトレースしていないことを想定し、ヘッドを一度原点に戻し、再度規定のトラックに送りなおす
- ②データが理想的なトラック位置(中心)からずれて書き込まれたことを想定して、読取ヘッドを中心からオフセットする
- ③ヘッドの書き込み／読取性能の劣化を想定して、読取ヘッドのアナログ出力信号を、A/D変換するためのフィルタ(微分積分回路)の定数を変更する

これらの方法がありますが、前述のように障害の半数をリードエラーが占めているという現実、それだけでは解消し切れない例があることを証明しています。

### ▶ HDDの現実の姿

最新のHDDは、ヘッドの浮上量は2nm、データの記録されているトラックの幅は55nmしかありません。この超精密な世界でのヘッド位置の正確な制御に影響を与える大きな要因は、軸受けです。軸受けには、耐久性を維持するためにオイルが必須で、これには避けることのできない摩擦抵抗が存在し、ごく少量とは言えども静止摩擦／動摩擦といった、非線形な特性があります。そして、ヘッ

ドをプラッタの半径方向に高速に移動させるために、過／減速を効率よく行い、アクセスタイムを最小化するための、特殊な制御パターンが設定されています。さらにヘッドアームには、慣性質量やVCMのコイルによる、電氣的／機械的時定数があり、その制御による実際の動作も非線形です。これらの要因によって、ヘッドの位置は、たとえ同じトラックやセクタであろうとも、送り方によって、微妙な位置の違いが発生し、トラッキングサーボやリトライ機能では解消できず、結果としてリードエラーの原因になります。

### ▶ クローンの作成

データ復旧業者は論理障害に分類されるものでも、顧客から預かったHDD全体のクローン、またはイメージファイルの作成から作業を始めます。これは、障害が悪化するリスクを最小限に押ええることが目的です。また、最近耳にするデジタルフォレンジックの証拠保全のためにも、リードエラーのあるHDDであってもクローン作成が必須作業です。さらに、そのリードエラーをより解消することが、その最終的な成果の決め手となります。修復の「あの手、この手」を持つのがデータ復旧業者であり、その詳細は企業秘密です(重度の物理的な障害の場合は、部品交換が必要になることもあります。この場合でもエラーを解消する能力の高い「あの手、この手」は、部品の互換性範囲を広げることができるので、復旧データの最終的な品質に影響します)。

### ▶ 日本産クローン作成ツール「MASAMUNE」

クローン作成のツールはDD、DD Rescue、DCFLDD、ハードウェアとしては、YEC社のDemi、ICS社のSoloなどが知られています。データ復旧業界では、総合的なデータ復旧装置であるAceLab社のPC-3000などがあります。本製品はこれらのツールに対して、同等以上の性能を目指して、当社の10年以上の経験から見つけ出した、独自の特殊なアルゴリズムを使った日本製のクローンツールです。SD

## 第2章

# UNIXバックアップ系 コマンド復習

石田 つばさ (ISHIDA Tsubasa) ITライター



## データバックアップ の基本

20数年前、私がUNIXに出会ったころは、データのバックアップ方法というと、ワークステーションに接続されたテープデバイスにコマンドでダンプをとる、というくらいしかなく、そのテープデバイスも、まだオープンリールテープやQICと呼ばれるカートリッジ式のテープでした。それも今は昔、現在ではバックアップに利用できるデバイスもメディアも多種多様で、バックアップのためのソフトウェア、ソリューション、サービスもより便利で高機能で高いデータ保全性をうたうものが数多く存在しています。昔のような「これしかやり方がないから、このやり方でのバックアップだけは確実にやっておく」というような環境から、数多くある手法からデータ保全を図ろうとしているデータの性質や運用環境に合わせて、いかに最適なソリューションを選択し運用していくか、ということを考えられる環境に変わってきています。

しかし、もし理想的な環境が用意してもらえなかったからといって、データが失われても許されるということはありません。幸いにも UNIX 環境ではいまでもバックアップに利用できるコマンドがサポートされており、これらを利用することで最低限のデータ保全を図ることができます。まずは、復習ということで、データのバックアップに利用できるコマンドとその利用方法を簡単に紹介しておきます。



## dump、restore コマンド

### ● 基本中の基本を押さえる

データのバックアップ用のコマンドといえば、まず真っ先に出てくるのはdump、restoreです。dump、restoreは、ペアで利用し、そのコマンド名のとおりにdumpがデータのダンプ(保存)、restoreがデータのリストア(復元)を行います。

基本的に、ディスクパーティション単位(ファイルシステム単位)でデータをダンプするので、あるパーティションの状態をまとめて一気にバックアップするのに向いています。逆に、このディレクトリ以下のデータのみをバックアップしたいとか、ある少数の重要なファイルだけをバックアップしたいというような細かな単位でのバックアップには向いていません。

このため、ディレクトリツリー上のデータがある程度うまく分類して必要に応じてバックアップをとろうとすると、インストール時にディスクのパーティションの分割をする段階で、dumpによるバックアップを意識してうまくパーティションの分割の仕方を設計しておかなくてはならないことになります。この点、最近のLinuxディストリビューションのインストールでパーティション分割をインストーラのデフォルトに任せると、/パーティションだけか、あるいは/と/bootパーティションだけというような分割の仕方になったりするので、これでは/から全



部のデータをバックアップするしかできないことになってしまいます(逆に、何も考えなくても全部のバックアップがとれるので、それはそれでいいのかもしれませんが)。

## ● dumpは自分でインストール

dump、restoreは、UNIXで標準的にサポートされているコマンドとはいえ、最近のLinux ディストリビューションでは、デフォルトインストールではインストールされず、別途パッケージを追加しなければならない場合があります。たとえば、Ubuntu 12.04LTSでデフォルトインス

トールした場合などは、dumpは追加インストールする必要があります(図1)。

dumpで保存先として通常ファイルを指定すると、その名前でファイルを作成しdump形式でバックアップデータを書き込みます。UNIXではデバイスもファイルの一種ですから、保存先のファイルとして、バックアップデバイスのデバイスファイルを指定すると、そのデバイスにdump形式でバックアップしたデータが保存されます。たとえば、外付けのUSB DATドライブやHDDドライブにもバックアップをとることができます(図2)。

▼図1 dumpのインストール(Ubuntu 12.04LTSの例)

```
genesis@AD0:~$ dump
プログラム 'dump' はまだインストールされていません。 次のように入力することでインストールできます:
sudo apt-get install dump
genesis@AD0:~$ sudo apt-get install dump
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下のパッケージが新たにインストールされます:
  dump
.../dump_0.4b44-1ubuntu1_i386.deb から) dump を展開しています...
man-db のトリガを処理しています ...
dump (0.4b44-1ubuntu1) を設定しています ...
update-alternatives: /usr/sbin/rmt (rmt) を提供するために 自動モード で /usr/sbin/rmt-dump を使います。
```

▼図2 dumpでバックアップ

```
genesis@AD0:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1        111263528 3344700 102266912   4% / ←sda1が/パーティションとしてマウントされている
udev             2047492      4    2047488   1% /dev
none             2054772     72    2054700   1% /tmp
tmpfs            821912     816     821096   1% /run
.....[省略].....
genesis@AD0:~$ dmesg
.....[省略].....
[ 702.639950] usb 1-6: new high-speed USB device number 5 using ehci_hcd
[ 702.777283] Initializing USB Mass Storage driver..
.....[省略].....
[ 703.777740] sd 4:0:0:0: Attached scsi generic sg1 type 0
[ 703.778189] sd 4:0:0:0: [sdb] 625142448 512-byte logical blocks: (320 GB/298 GiB)
.....[省略].....
[ 703.778954] sd 4:0:0:0: [sdb] Write Protect is off
.....[省略].....
[ 704.386185] sd 4:0:0:0: [sdb] Attached SCSI disk
genesis@AD0:~$ sudo dump -0 -a -f /dev/sdb1 /←/パーティション(sda1)をsdb1にダンプレベル0(フルバックアップ)でダンプする
DUMP: Date of this level 0 dump: Tue Nov 13 14:04:39 2012
.....[省略].....
DUMP: Average transfer rate: 25034 kB/s
DUMP: DUMP IS DONE
```

dumpでとったバックアップデータは、restoreで復元します。restoreは、指定したdump形式のファイル(あるいはdump形式のデータが書き込まれているデバイスが接続されているデバイスファイル)を読み込み、その内容をディスク上に展開します。バックアップされたデータを一気にまとめて復元することもできますし、指定した一部のファイルだけを復元することもできます。また、restoreは「対話モード」を持っていて、バックアップデータの内容を対話的に確認しながら、データを選択して復元していくこともできます(図3)。

dump(あるいは後述するtar)では、DATテープなどのテープデバイスにバックアップをとる場合、元のデータよりもテープの容量が大きければ、1本のテープに対して、複数のバックアップデータを容量が許す限り追記していくという使い方もできます。たとえば、/、/boot、/varの3つのパーティションに分割して利用している場合に、1本のデータに各パーティションのデータをまとめてバックアップすることもできます(リスト1)。

上記のようにしてダンプをとったテープからの復元は次のようになります。順番にすべてを復元するには、

```
restore -f /dev/nst0
```

と実行するたびに/boot、/、/varの内容がそれぞれ復元されます。

テープの頭出しをしていづれかのパーティション

### ▼図3 restoreで「対話モード」を使う

```
genesis@AD0:/tmp$ ls      ←/tmpディレクトリにはetcディレクトリはない
at-spi2                  pulse-2L9K88eMlGn7  pulse-j8u8YhoSzMAf  unity_support_test.1
keyring-F9R70v          pulse-PKdhtXMmr18n  ssh-k0pwbTaF1487
genesis@AD0:/tmp$ sudo restore -i -f /dev/sdb1
                                ←ディスクsdb1にあるダンプデータをrestoreの対話モード(iオプション)で読み込み復元する

restore > ls                  ←対話モードの内部コマンドlsでダンプされている内容を確認する
..
bin/                          home/                        media/                      run/                         tmp/
boot/                         initrd.img@                mnt/                       sbin/                       usr/
cdrom/                       initrd.img.old@            opt/                       selinux/                    var/
dev/                          lib/                       proc/                      srv/                        vmlinuz@
etc/                          lost+found/                root/                      sys/ ←etcディレクトリがダンプされている
restore > add etc              ←内部コマンドaddでetcディレクトリを復元対象として指定する
restore > ls                  ←再度lsで確認してみる
..
bin/                          initrd.img@                opt/                       srv/
boot/                         initrd.img.old@            proc/                     sys/
cdrom/                       lib/                       root/                     tmp/
dev/                          lost+found/                run/                      usr/
*etc/                        media/                     sbin/                     var/
↑ 復元対象として選択されたetcに選択済みの*がついている
home/                        mnt/                       selinux/                  vmlinuz@
restore > extract              ←内部コマンドextractで指定されたデータ(この場合はetcディレクトリ)の復元を実行する
```

### ▼リスト1 dumpのまとめバックアップスクリプト例

```
#!/bin/sh
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

mt -f /dev/st0 rewind      ←mtでテープを巻き戻す
dump -0auf /dev/nst0 /boot
                                ←/bootの内容をフルバックアップする。st0ではなくnst0を指定しているので、dump実行後にテープは巻き戻されない
dump -0auf /dev/nst0 /
                                ←/bootの後に/の内容をフルバックアップする
dump -0auf /dev/nst0 /var
                                ←/の後に/varの内容をフルバックアップする
mt -f /dev/st0 offline     ←mtでテープを巻き戻してオフラインにする(イジェクトする)
```

ンのデータのみを復元するには、restoreを実行する前に次のようにmtを実行します。

```
mt -f /dev/nst0 fsf n
```

n=1の場合、/bootの部分を「1つ」読み飛ばして2番目の/のデータの頭出しができ、そこでrestoreを実行すると/のデータが復元できます。n=2の場合、「2つ」読み飛ばして3番目の/varのデータの頭出しができます。



## ddコマンド

### イメージのコピー

ddは、一説にはDisk Dumpの略と言われるよ

うに、ディスクのイメージをそのままファイルやデバイスにコピーすることができます。ディスクをまるごとそっくりコピーしてくれるので、コピー元のドライブのパーティションの切り方からファイルシステムのフォーマットなどもそのまま保持された状態でコピーされます。同じディスクを用意してddでコピーすれば、それこそまったくの複製を作ることができます。

ddでバックアップしたデータはddで復元できます。ddではifオプションで入力(コピー元のファイル、デバイス)を、ofオプションで出力(コピー先のファイル、デバイス)を指定しますが、復元するときにはこの入力と出力を逆に指定することになります。

単純に復元するとddではすべてのバックアップ

▼図4 ddコマンドの使用例(※図2の例で使用したディスクを使用。dumpでデータを保存した状態なので、通常のファイルシステムとしてはマウントできない)

```
genesis@AD0:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1      111263528 3288816 102322796 4% /
udev            2047488      4    2047484 1% /dev
tmpfs           821912      800    821112 1% /run
none            5120         0      5120 0% /run/lock
none           2054772     124    2054648 1% /run/shm

genesis@AD0:~$ sudo mount /dev/sdb1 /mnt
mount: ファイルシステムタイプを指定する必要があります
genesis@AD0:~$ sudo mount -t ext4 /dev/sdb1 /mnt
mount: 間違ったファイルシステムタイプ、不正なオプション、
/dev/sdb1 のスーパーブロックが不正、コードページまたは
ヘルパープログラムの未指定、或いは他のエラー
In some cases useful info is found in syslog - try
dmesg | tail or so

genesis@AD0:~$ sudo dd if=/dev/sda1 of=/dev/sdb1
226076672+0 レコード入力
226076672+0 レコード出力
115751256064 バイト (116 GB) コピーされました、4429.93 秒、26.1 MB/秒

genesis@AD0:~$ sudo mount /dev/sdb1 /mnt

[sudo] password for genesis:
genesis@AD0:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1      111263528 3296204 102315408 4% /
udev            2047488      4    2047484 1% /dev
tmpfs           821912      800    821112 1% /run
none            5120         0      5120 0% /run/lock
none           2054772     124    2054648 1% /run/shm
/dev/sdb1       111263528 3272336 102339276 4% /mnt

genesis@AD0:~$ ls /mnt/
bin  cdrom  etc  initrd.img  lib  media  opt  root  sbin  srv  tmp  var
boot dev  home  initrd.img.old  lost+found  mnt  proc  run  selinux  sys  usr  vmlinuz
```

←sda1は3GB程しか使われていない

←sdb1のディスクをマウントできない(※)

←ddでsda1のディスクイメージをsdb1にコピーする

←sda1のディスクイメージ約116GBがコピーされた(使用されている3GB分がコピーされるのではないことに注意)

←ファイルシステムの情報も含めてコピーされているので、マウント可能になっている

←/mntにマウントできている

←sda1の/からの内容が確かにコピーされている



データをまとめて復元することになりますが、元のフォーマットも保持された状態でコピーされるので、ext4などマウント可能なディスクフォーマットであれば、mountで/mntなどにマウントして、必要な一部のファイルのみをcpなどで復元していくということもできます(図4)。もちろん、元のディスクの完全な複製を作ったのであれば、ディスクをつなぎ替えてしまうということも考えられます。

ddでバックアップを取るときに注意しなければならないのは、コピー元と同容量以上のディスクをコピー先として用意しなければならない、ということです。たとえば、500GBのディスク(パーティション)のうち実際にはデータは200GB分しかないマシンがあったとします。dumpやtarでバックアップをとる場合は、200GB以上のディスクを用意すればバックアップがとれるわけですが、ddでは500GBよりも大きなディスクを用意しなければいけないということになります。これは、ddがディスクイメージをそのままコピーするため、ファイルシステム上のファイルとして有効な状態になっている部分を選択してコピー、記録するのではないということから生じる制限です。

図4の例では、実際にddでコピー先としたのは図2でも使用した320GBの外付けUSB HDD

ですが、ddでコピーした後にマウントしてみると、コピー元と同じ120GBのディスクとしてしか見えていないことがわかります。

また、このようにならずディスク容量全体のコピーを行うということは、その分だけバックアップのための時間もかかるので、頻繁にddでバックアップをとることは難しいでしょう。



## tarコマンド

tarは、Tape ARchivierというコマンド名の由来のとおり、もともとはテープにアーカイブをとるコマンドでしたが、もちろんテープデバイス以外のデバイスにアーカイブしたり、通常のファイルにアーカイブイメージを保存したりすることができます(図5)。

tarは、個別に指定したファイルのみアーカイブしたり、指定したディレクトリ以下のディレクトリツリーをアーカイブしたりできるので、ファイルシステム(パーティション)単位でのバックアップになってしまうdumpよりも、より対象となるデータを絞り込んでバックアップできます。データを絞り込めれば、当然バックアップする容量も小さくなり、それにかかる時間も短くできます。たとえば、この原稿用のデータディレクトリ/home/genesis/gihyoのバック

▼図5 tarコマンドの使用例

```
genesis@AD0:~$ ls
dd.txt  examples.desktop  restore.txt~ テンプレート ドキュメント ピクチャ 公開
dd.txt~ restore.txt  ダウンロード デスクトップ ビデオ ミュージック
genesis@AD0:~$ cd /usr/src
genesis@AD0:/usr/src$ ls
linux-headers-3.2.0-23          linux-headers-3.2.0-32
linux-headers-3.2.0-23-generic-pae  linux-headers-3.2.0-32-generic-pae
genesis@AD0:/usr/src$ tar cfz ~/src.tgz ./*
genesis@AD0:/usr/src$ cd
genesis@AD0:~$ ls
dd.txt  examples.desktop  restore.txt~ ダウンロード デスクトップ ビデオ ミュージック
dd.txt~ restore.txt  src.tgz      テンプレート ドキュメント ピクチャ 公開
genesis@AD0:~$ tar tvfz src.tgz
drwxr-xr-x root/root      0 2012-04-23 20:37 ./linux-headers-3.2.0-23/
-rw-r--r-- root/root    252 2012-01-05 08:55 ./linux-headers-3.2.0-23/Kconfig
drwxr-xr-x root/root      0 2012-04-23 20:37 ./linux-headers-3.2.0-23/fs/
..... [省略] .....
```

←ホームディレクトリにはsrc.tgzファイルはない

←/usr/srcディレクトリに移動する

←usr/src以下のディレクトリツリーをホームディレクトリのsrc.tgzにアーカイブする

←先ほどなかったsrc.tgzファイルが作成されている

←tオプションでsrc.tgzの内容を表示してみると、確かに/usr/src以下のデータがアーカイブされている



アップをとろうとしたときに、dumpでは/homeパーティション全体(ということは、マルチユーザで利用しているサーバだと他のユーザのホームディレクトリまで)をバックアップすることになりますが、tarならばまさにgihyoディレクトリ以下だけをバックアップすることができます。

また、あるディレクトリツリーが複数のパーティションをまたがっているような場合、そのディレクトリツリーをバックアップするには、dumpだとすべてのパーティションのバックアップを実行しなければなりません、tarであれば1回にまとめてバックアップすることができます。たとえば、/usr以下を/usrパーティションと/usr/localパーティションと2つのパーティションに分けていた場合、dumpだと/usrと/usr/localと2つのパーティションのバックアップを実行しないといけません。tarであれば/usr以下をアーカイブするように指定すれば/usr/local以下もまとめてアーカイブできます。

このように、主要なコマンドで少しずつ機能が異なるので、それぞれの特徴をうまく利用してバックアップのとり方を考えるとよいでしょう。



## rsyncコマンド

以上、バックアップ用途向けと考えられるUNIXコマンドを簡単に紹介しましたが、これらのコマンドを使うことがバックアップではありません。重要なのは、必要なデータが失われたときにそのデータが復元できるように適切な複製をどこかにとっておくこと(さらに、しっかりとその複製から復元でき、必要なサービスが再開できるようになっていること)です。その複製をとれるのであれば、単なるcpだろうがGUI上でのドラッグ&ドロップでのコピーだろうが、それでも十分とも考えられます。

データをコピーするということで考えると、rsyncも便利なバックアップツールとして使えるでしょう。rsyncは、単にコピーするだけでな

く、タイムスタンプを比較して更新されたデータのみをコピーしたり(この機能はcpにも取り込まれていたりしますが)、コピー元で削除されたデータをコピー先からも削除するかしらないか指定できたりと、便利な機能を備えています。rsyncは、rsyncdが動作しているリモートホストとの間や、あるいはSSH経由で、リモートでデータの同期をとることもできます。バックアップ用のディスクをマウントしておいて、そのマウント先のディレクトリにコピーしたり、あるいはリモートのホストのディスクをバックアップ先として利用したりできるわけです。

完全に同じ状態のディレクトリの複製としたければコピー元で削除されたデータはコピー先でも削除するようにすればよいし、誤って削除した場合にコピー先から復元できるようにしたければ、コピー元で削除されたファイルはコピー先には反映しないようにする、といった使い方ができます。



## まとめ

駆け足でいくつかのコマンドを紹介しましたが、コマンドベースだと、スクリプトにして手順を簡略化、自動化したり、さらにcronから深夜に自動実行させてバックアップ漏れを防いだりという工夫もできて、意外とそこそこのバックアップニーズに応えることができます。Sambaなどと組み合わせるとWindowsサーバにデータをコピーしてバックアップをWindows側でとってもらったり、Windowsの共有ディスクをUNIX側でマウントしてtarでバックアップをとったりと、いろいろな応用が考えられるので、工夫してみるとおもしろいでしょう。SD

## 第3章

# クラウドのデータはどこにいくの？

## [Nifty Cloud 編]

大久保 智之(OKUBO Tomoyuki)  
フォースクーナ(株)



### はじめに

昨今、パブリッククラウドの利用がより身近なものになってきました。パブリッククラウドを利用するメリットとして、オンプレミス環境ではこれまですべて自分達で行ってきた運用管理から一部解放されるという点が挙げられます。しかしながら、パブリッククラウド利用者が引き続き意識しなくてはいけない運用管理項目もあります。その1つがデータのバックアップです。パブリッククラウドを利用するにあたり、パブリッククラウド事業者とパブリッククラウド利用者の責任分解点を定義すると、パブリッククラウド利用者側で何をしなくてはいけないのかが見えてきます。

今回は筆者が所属する当社が提供しているパブリッククラウド運用管理サービス「Easy Cloud Operation(以降、E.C.O)」で採用しているバックアップツールを中心に、Nifty Cloudにおけるデータバックアップの環境構築例についてご紹介します。



### E.C.O.におけるバックアップ運用の検討

#### ● バックアップ対象の検討

弊社サービスE.C.O.におけるバックアップ運用の検討にあたり、はじめにパブリッククラウ

ド事業者とその利用者の責任分解点を検討しました。両者の責任分解点を意識することで、パブリッククラウド利用者が行うべきことが明確になるためです。具体的にはクラウド基盤の管理はパブリッククラウド事業者、クラウド基盤の上で動く仮想サーバ／ユーザデータの管理はパブリッククラウド利用者という分解点を設けました。バックアップの対象とする範囲を定義した後、今度はバックアップ手法について検討しました(図1)。仮想サーバのバックアップ方針はパブリッククラウドの環境に依存するため、パブリッククラウドの機能を利用したバックアップ取得としました。

次にユーザデータ保護を目的としたバックアップを検討するために、ユーザデータを分類しました。最終的にユーザデータをユーザデータ、データベースのデータに分類し、それぞれツールの選定を行いました。

#### ● バックアップツールの選定

バックアップツールの選定は、次の条件を満たすものにしました。

- ・オープンソース
- ・機能(エージェント／サーバ)の同居／分離
- ・ツールに依存せずに復旧できること
- ・公開ポート数

#### ▶ オープンソース

後述する条件を満たしていたこともあり、バックアップツールはAmandaとZmanda Recovery



Manager for MySQLを選定しました。エンタープライズの観点で、必要に応じて商用サポートが受けられるかという点も考慮しました。

#### ▶機能の同居／分離

稼働中のサーバにバックアップ機能を同居できることを条件としました。サービスインしているサーバに運用サービスを組み込むこともあるため、専用のバックアップサーバを用意できることが望ましいですが、既存環境に対して、アドオンの組み込むめることを要件としました。

#### ▶ツールに依存せずに復旧できること

バックアップしたデータのデータ形式が特殊なものであると、取得したバックアップデータを戻せない可能性があります。このリスクを避けるため、OSやデータベースソフトウェアで一般的に使用されている手段を使っていることを選定基準としました。

AmandaはdumpおよびGNU tar形式でのデータバックアップ、Zmanda Recovery Manager for MySQLはmysqldumpやmysqldump コマンドを使ったバックアップを行っているため、ツールなしでのデータ復旧が可能です。この点が選定理由の1つとなりました。

#### ▶公開ポート

パブリッククラウド事業者が提供するファイアウォールや仮想サーバ上でパケットフィルタリングを行うケースもあるため、ポートが多いと構成が複雑になることが考えられました。そのため、公開ポートの少なさを条件としました。



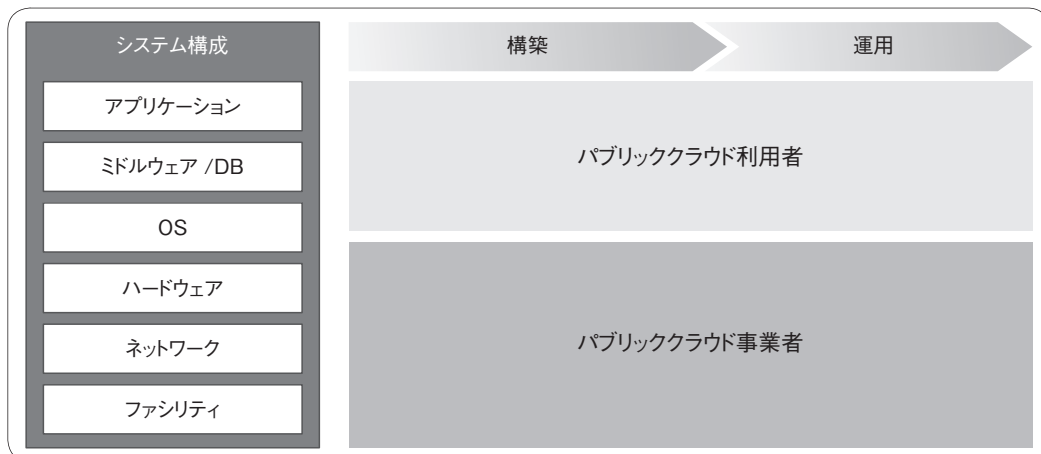
### Nifty Cloudでのバックアップ運用例

それではE.C.OにおけるNifty Cloudでの例を紹介します。仮想サーバのバックアップはカスタマイズイメージ、ユーザデータ領域のファイルバックアップにはAmanda、MySQLのバックアップにはZmanda Recovery Manager for MySQLを利用する構成になっています。以降はバックアップツールの使い方ということで、Amanda、Zmanda Recovery Manager for MySQLのセットアップ方法と使い方を中心に説明します。

#### ●カスタマイズイメージ

仮想サーバのバックアップを行う場合、弊社ではNifty Cloudで提供されているカスタマイズイメージ機能を利用しています。これは物理環境におけるOSのシステムバックアップに相当します。イメージ化したいサーバが稼働して

▼図1 パブリッククラウド事業者／利用者の責任分解点イメージ



いる場合は停止しなくてはなりませんが、「サーバのイメージ化」を実行することで、Nifty Cloud上にその時点でのカスタマイズイメージを取得／保存します。図2はNifty Cloudで取得したカスタマイズイメージの例です。



## ユーザデータファイルのバックアップ(Amandaの利用例)

Amandaはユーザデータ領域のファイルをバックアップする目的で利用しています。今回はセットアップ手順とバックアップ・リストアの方法について説明します(図3)。

### ● セットアップ

AmandaはRPMパッケージが用意されているため、パッケージを使ってインストールします。サーバ／クライアント別にそれぞれ例示します。次はサーバ／クライアント共通で行う設定です。

#### ① hosts ファイルへのIPアドレス登録

Amandaはhostsを使って名前解決するため、hosts ファイルにAmanda Server/ClientのIPアドレスを設定します。

```
# vi /etc/hosts
192.168.0.100 amanda_client
192.168.0.101 amanda_server
```

#### ② パッケージのインストールと xinetd の起動

事前に依存するパッケージをインストールし、xinetdを起動させます。

```
# yum -y install xinetd.x86_64 gnupg.x86_64 sharutils.x86_64
# /etc/init.d/xinetd start
# chkconfig xinetd on
```

#### ③ Amandaの通信許可設定

ファイアウォールやiptablesを使っている場合は、10080番ポートの通信を許可します。

#### ▶ クライアントのセットアップ

①から⑤はクライアントで行う作業です。

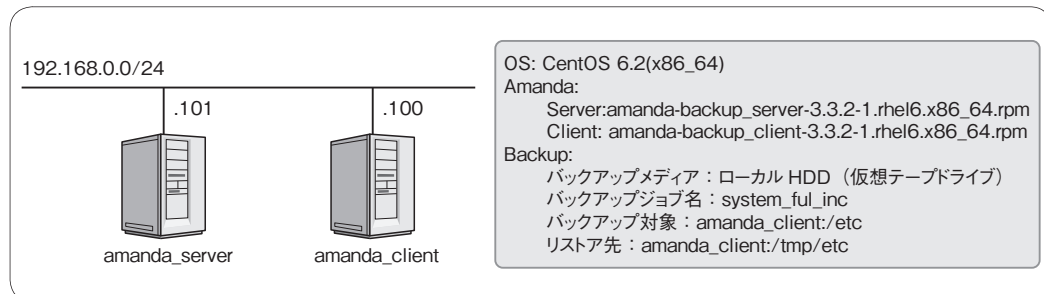
#### ① Amanda Clientのインストール

```
# wget http://www.zmanda.com/downloads/
community/Amanda/3.3.2/Redhat_Enterprise_6.0/amanda-backup_client-3.3.2-1.rhel6.x86_64.rpm
# rpm -ivh amanda-backup_client-3.3.2-1.rhel6.x86_64.rpm
```

▼図2 カスタマイズイメージ



▼図3 Amanda環境構成





## ② Server/Client アクセスファイル (amandahosts) の作成、パーミッション設定

```
# vi ~amandabackup/.amandahosts
amanda_server amandabackup amdump
# chmod 700 ~amandabackup/.amandahosts
```

## ③ Amanda Client の設定ファイルの作成。バックアップジョブ名は「system\_ful\_inc」

```
# mkdir /etc/amanda/system_ful_inc
# vi /etc/amanda/system_ful_inc/amanda-☐
client.conf
conf "system_ful_inc"
index_server "amanda_server"
tape_server "amanda_server"
tapedev "file://var/lib/amanda/☐
vtapes"
auth "bsdtcp"
ssh_keys "/var/lib/amanda/.ssh/id_☐
rsa_amrecover"
```

## ④ amandates ファイルの作成

```
# touch /var/amanda/amandates
```

## ⑤ 関連するディレクトリの権限の変更

```
# chown -fR amandabackup:disk /var/☐
amanda /etc/amanda
```

### ▶ サーバのセットアップ

①から⑪はサーバで行う作業です。

## ① Amanda Server のインストール

```
# wget http://www.zmanda.com/downloads/☐
community/Amanda/3.3.2/Redhat-☐
Enterprise_6.0/amanda-backup_server-☐
3.3.2-1.rhel6.x86_64.rpm
# rpm -ivh amanda-backup_server-3.3.2-1.☐
rhel6.x86_64.rpm
```

## ② Server/Client アクセスファイル (amandahosts) の作成とパーミッションの設定

```
# vi ~amandabackup/.amandahosts
amanda_server root amindexd amidxtaped
amanda_server amandabackup amdump
amanda_client root amandabackup ☐
amindexd amidxtaped amdump
# chmod 700 ~amandabackup/.amandahosts
```

## ③ コンフィグファイルの作成 (内容はリスト1 参照)

```
# vi /etc/amanda/system_ful_inc/amanda.conf
```

## ④ ディレクトリの作成

```
# mkdir -p /var/lib/amanda/holding /etc/☐
```

```
amanda/system_ful_inc/state/☐
{curinfo,log,index}
```

## ⑤ 関連ディレクトリの権限を変更

```
# chown -fR amandabackup:disk /var/☐
amanda /etc/amanda
```

## ⑥ 仮想テープドライブの作成

```
# su - amandabackup
$ for slot_num in `seq 1 25` ; do
  mkdir -p /var/lib/amanda/vtapes/☐
slot${slot_num}
done
```

## ⑦ 仮想テープドライブのセット

```
$ ln -s /var/lib/amanda/vtapes/slot1 ☐
var/lib/amanda/vtapes/data
```

## ⑧ 仮想テープのセットアップテスト

```
$ amtape system_ful_inc show
```

## ⑨ テープチェンジャのスロットを戻す

```
$ amtape system_ful_inc reset
```

## ⑩ バックアップ対象ファイル (disklist) リストの作成 (ここでは amanda\_client の /etc 以下をバックアップする設定)

```
$ vi /etc/amanda/system_ful_inc/disklist
amanda_client /etc comp-user-tar
```

## ⑪ 構文チェックの実行 (エラーが出力されなければ OK)

```
$ amcheck system_ful_inc
```

### ▶ バックアップ

バックアップは Amanda Server で①と②のコマンドを実行します。

## ① バックアップの実行

```
$ amdump system_ful_inc
```

## ② バックアップ結果の確認

```
$ amadmin system_ful_inc find
date host ☐
disk lv tape or file file part status
2012-11-04 23:41:58 amanda_client ☐
/etc 0 system_ful_inc-01 1 1/1 OK
```



## ▶ リストア

リストアはAmanda ClientからAmanda Serverに接続して①から⑥の手順で行います。今回はワーキングディレクトリにリストアしましたので、その後必要な場所にファイルを再配置して復旧作業を行います。

### ① Amanda Backup サーバへ接続

```
# amrecover -s amanda_server -t amanda_
server -C system_ful_inc
AMRECOVER Version 3.3.2. Contacting
server on amanda_server ...
220 amanda AMANDA index server (3.3.2)
ready.
Setting restore date to today (2012-11-10)
200 Working date set to 2012-11-10.
200 Config set to system_ful_inc.
200 Dump host set to amanda_client
Use the setdisk command to choose dump
disk to recover
```

### ② バックアップ対象一覧を表示

```
amrecover> listdisk
200- List of disk for host amanda_client
201- /etc
200 List of disk for host amanda_client
```

### ③ リストア対象ディレクトリを指定

```
amrecover> setdisk /etc
200 Disk set to /etc.
```

### ④ リストア先のワーキングディレクトリを確認

```
amrecover> lcd /tmp/etc
amrecover> lpwd
/tmp/etc
```

### ⑤ リストア対象とするファイルを追加

```
amrecover> add *
Added dir /yum/ at date 2012-11-12-04-
13-29
Added dir /yum.repos.d/ at date 2012-11-
12-04-13-29
Added file /yum.conf
...
```

## ▼ リスト1 amanda.confの例

```
# vi /etc/amanda/system_ful_inc/amanda.conf
org "system_ful_inc"
send-amreport-on all
dumpuser "amandabackup"
inparallel 4
dumporder "ssss"
taperalgo first
displayunit "k"
netusage 100000 Kbps
dumpcycle 1 weeks
runspcycle 7
tapecycle 16 tapes
bumpsizes 20 Mb
bumppercent 20
bumpdays 1
bumpmult 4
ctimeout 120
etimeout 1800
dtimeout 300
connect-tries 3
req-tries 5
device_output_buffer_size 1280k
usetimestamps yes
flush-threshold-dumped 0
flush-threshold-scheduled 0
taperflush 0
autoflush no
runtapes 1
maxdumpsizes -1
labelstr "^system_ful_inc-[0-9][0-9]*$"
amrecover_changer "changer"
holdingdisk hd1 {
    comment "main holding disk"
    directory "/var/lib/amanda/holding"
    use 3 Gb
    chunksize 1Gb
}
infofile "/etc/amanda/system_ful_inc/state/curinfo"
logdir "/etc/amanda/system_ful_inc/state/log"
```

```
indexdir "/etc/amanda/system_ful_inc/state/index"
tpchanger "chg-disk"
tapedev "file:/var/lib/amanda/vtapes"
tapetype HARDDISK
define tapetype global {
    part_size 36
    part_cache_type none
}
define tapetype HARDDISK {
    length 3072 mbytes
}
define dumptype global {
    comment "Global definitions"
    index yes
    auth "bsdtcp"
}
define dumptype root-tar {
    global
    program "GNUTAR"
    comment "root partitions dumped with tar"
    compress none
    index
    priority low
}
define dumptype user-tar {
    root-tar
    comment "user partitions dumped with tar"
    priority medium
}
define dumptype comp-user-tar {
    user-tar
    compress client fast
    estimate calcsizes
}
define taperscan taper_lexical {
    comment "lexical"
    plugin "lexical"
}
taperscan "taper_lexical"
```



## ⑥ リストアの実行

```
amrecover> extract
./NetworkManager/
./NetworkManager/dispatcher.d/
./X11/
./X11/applnk/
./X11/fontpath.d/
...
```



## MySQLのバックアップ

MySQL のデータバックアップは Zmanda Recovery Manager for MySQL を使用しています(図4)。こちらも RPM パッケージが用意されているため、インストールはパッケージで行います。

## ● インストール

①から④の手順でインストールを行います。

### ①事前にパッケージのインストール

```
# yum -y install perl-DBI perl-XML-Parser
```

### ② Zmanda Recovery for MySQL のインストール (サーバとクライアントを共存させる構成)

```
# wget http://www.zmanda.com/downloads/
community/ZRM-MySQL/2.2/RPM/MySQL-zrm-
2.2.0-1.noarch.rpm
# rpm -ivh MySQL-zrm-2.2.0-1.noarch.rpm
# wget http://www.zmanda.com/downloads/
community/ZRM-MySQL/2.2/RPM/MySQL-zrm-
client-2.2.0-1.noarch.rpm
# rpm -ivh MySQL-zrm-client-2.2.0-1.
noarch.rpm
```

### ③バックアップ用のユーザ"backup"の作成

```
mysql> GRANT SELECT, INSERT, UPDATE,
```

```
CREATE, DROP, RELOAD, SHUTDOWN, ALTER,
SUPER, LOCK TABLES, REPLICATION CLIENT
ON *.* TO 'backup'@'localhost'
IDENTIFIED BY '*****';
mysql> FLUSH PRIVILEGES;
```

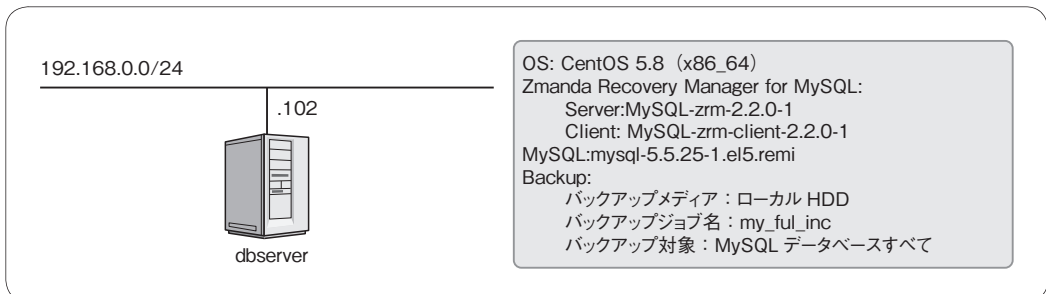
### ④ディレクトリ／コンフィグレーションの作成(弊社で使用している例)

```
# mkdir -p /etc/mysql-zrm/my_ful_inc
# vi /etc/mysql-zrm/my_ful_inc/mysql-
zrm.conf
backup-mode=logical
destination=/var/lib/mysql-zrm
retention-policy=2W
compress=1
compress-plugin=/usr/bin/gzip
encrypt=1
encrypt-plugin="/usr/share/mysql-zrm/
plugins/encrypt.pl"
decrypt-option="-d"
all-databases=1
user="backup"
password="*****"
port=3306
socket=/var/lib/mysql/mysql.sock
ssl-options="--ssl --ssl-ca=file1 --ssl-
cert=file2 --ssl-key=file3"
mysql-binlog-path="/var/lib/mysql"
tmpdir=/tmp
verbose=1
mail-policy=always
passfile="/etc/mysql-zrm/my_ful_inc/.
passphrase"
```

### ⑤コンフィグレーションのチェック

```
# mysql-zrm --action check --backup-set
my_ful_inc
check:INFO: ZRM for MySQL - version
built from source
my_ful_inc:check:INFO: The quick backup-
type is supported only for snapshot
backups. Setting backup-type to 'regular'
my_ful_inc:check:INFO: ZRM Temporary
configuration file = /etc/mysql-zrm/
my_ful_inc/tmpKczNu.conf
my_ful_inc:check:INFO: {
my_ful_inc:check:INFO: verbose=1
```

▼図4 Zmanda Recovery Manager for MySQL 環境構成



```

my_ful_inc:check:INFO: encrypt-plugin=
usr/share/mysql-zrm/plugins/encrypt.pl
my_ful_inc:check:INFO: retention=
policy=2W
my_ful_inc:check:INFO: tmpdir=/tmp
my_ful_inc:check:INFO: all-databases=1
my_ful_inc:check:INFO: destination=/var
/lib/mysql-zrm
my_ful_inc:check:INFO: decrypt-option=-d
my_ful_inc:check:INFO: password=*****
my_ful_inc:check:INFO: backup-
mode=logical
my_ful_inc:check:INFO: ssl-options=
--ssl --ssl-ca=file1 --ssl-cert=file2
--ssl-key=file3
my_ful_inc:check:INFO: backup-
type=regular
my_ful_inc:check:INFO: compress-plugin=
/usr/bin/gzip
my_ful_inc:check:INFO: user=backup
my_ful_inc:check:INFO: backup-level=0
my_ful_inc:check:INFO: encrypt=usr/
share/mysql-zrm/plugins/encrypt.pl
my_ful_inc:check:INFO: port=3306
my_ful_inc:check:INFO: socket=/var/lib/
mysql/mysql.sock
my_ful_inc:check:INFO: mail-policy=always
my_ful_inc:check:INFO: passfile=/etc/
mysql-zrm/my_ful_inc/.passphrase
my_ful_inc:check:INFO: compress=/usr
/bin/gzip
my_ful_inc:check:INFO: mysql-binlog-
path=/var/lib/mysql
my_ful_inc:check:INFO: }
my_ful_inc:check:INFO: Getting mysql
variables
my_ful_inc:check:INFO: mysqladmin
--user="backup" --password="*****"
--port="3306" --socket="/var/lib/mysql/
mysql.sock" --ssl --ssl-ca=file1 --ssl-
cert=file2 --ssl-key=file3 variables
my_ful_inc:check:INFO: datadir is /var/
lib/mysql/
my_ful_inc:check:INFO: mysql_version is
5.5.25-log
my_ful_inc:check:INFO: InnoDB data file
are /var/lib/mysql/ibdata1
my_ful_inc:check:INFO: InnoDB log dir is
/var/lib/mysql

```

## ▼図5 Amandaのバックアップ成功

Current Status:	OK (for 41d 3h 20m 31s)
Status Information:	Backup Success - Summary/etc=OK /var/www/html=OK /opt=OK - Verbose 2012-11-11 01:00:02 /etc 0 system, 5 1/1 OK 2012-11-11 01:00:02 /opt 0 system, ful_inc-04 9 1/1 OK 2012-11-11 01:00:02 /var/www/html 0
Performance Data:	etc=99960 /var/www/html=10 /opt=312500
Current Attempt:	1/3 (HARD state)
Last Check Time:	11-11-2012 13:16:56
Check Type:	ACTIVE
Check Latency / Duration:	0.054 / 0.407 seconds
Next Scheduled Check:	11-12-2012 13:16:56
Last Date Change:	10-01-2012 12:10:50
Last Notification:	N/A (notification 0)
Is This Service Flapping?	NO (0.00% state change)
Is Scheduled Downtime?	NO
Last Update:	11-11-2012 16:37:22 ( 0d 0h 6m 6s ago)
Modified Attributes:	None
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obssessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

```

my_ful_inc:check:INFO: Configuration
check successful

```

## ▶バックアップ

バックアップを取得する場合はmysql-zrm-scheduler コマンドを使います。次はバックアップセットにmy\_ful\_incを指定し、フルバックアップを取得する場合の例です。

```

# mysql-zrm-scheduler --now --backup-
level 0 --backup-set my_ful_inc

```

実行結果の確認はmysql-zrm-reporter コマンドで確認します。

```

# mysql-zrm-reporter --show backup-
status-info --where backup-set=my_ful_inc
reporter:INFO: Reporter configuration
file not found, assigning default values
..... (途中省略) .....

```

バックアップの整合性確認を行うには次のようになります。

```

# mysql-zrm --action verify-backup
--backup-set my_ful_inc
verify-backup:INFO: ZRM for MySQL -
version built from source
my_ful_inc:verify-backup:INFO:
Verification successful

```

## ▶リストア

Zmanda Recovery Manager for MySQL を使ったリストアを行います。mysql-zrm-reporter コマンドを使って、バックアップの状態を確認します。

```

# mysql-zrm-reporter --show restore-
info --where backup-set=my_ful_inc
reporter:INFO: Reporter configuration
file not found, assigning default values
.....途中省略.....

```

データのリストアはmysql-zrm コマンドで行います。次は2012年11月12日のバックアップデータをリストアする例です。

```

# mysql-zrm --action restore --backup-

```

注1) Amanda/Zmanda Recovery Manager for MySQLにも同様の機能が備わっています。



```
set my_ful_inc --source-directory /var/□
lib/mysql-zrm/my_ful_inc/20121112040001/
```

MySQL については、Zmanda Recovery Manager for MySQLを使って、MySQLのデータベースデータをローカルに複数世代保持するようにしています。

### ▶バックアップ運用と監視

運用フェイズではバックアップの取得に加えて、「日々のバックアップ取得が成功しているかどうか」を確認することが必須になります。一般

▼図6 MySQLのバックアップ失敗

Service State Information	
Current Status:	CRITICAL (for 0d 7h 23m 35s)
Status Information:	Backup Failure - Summary: backup-status=Backup failed
Performance Data:	
Current Attempt:	3/3 (HARD state)
Last Check Time:	11-11-2012 14:34:34
Check Type:	ACTIVE
Check Latency / Duration:	0.212 / 0.195 seconds
Next Scheduled Check:	11-12-2012 14:34:34
Last State Change:	11-11-2012 09:14:52
Last Notification:	11-11-2012 09:35:02 (notification 1)
Is This Service Flapping?	NO (5.06% state change)
In Scheduled Downtime?	NO
Last Update:	11-11-2012 16:38:22 (0d 0h 0m 5s ago)
Modified Attributes:	None
Active Checks:	ENABLED
Passive Checks:	ENABLED
Obsessing:	ENABLED
Notifications:	ENABLED
Event Handler:	ENABLED
Flap Detection:	ENABLED

的に実行結果をメールで通知することは可能<sup>注1</sup>ですが、通知されたメールの内容でバックアップ処理の結果を判断する必要があるため、管理対象とするノード数に比例して運用負荷が増えます。

弊社ではAmanda/Zmanda Recovery Manager for MySQLのバックアップ実行結果についてNagios/Icingaのプラグインを用いて、バックアップの実行結果を監視しています。バックアップ処理が異常終了したときに監視システムからアラートを通知する運用としています(図5、図6)。



## おわりに

駆け足になりましたが、今回は弊社サービスE.C.Oで使用しているバックアップツールの紹介を中心にNifty Cloudでのバックアップ運用の一例をご紹介させていただきました。Amandaについては弊社のブログでも技術情報を公開しています<sup>注2</sup>。お役に立ていただければ幸いです。

### SD

注2) <http://www.forschooner.net/blog-unkai/entry/000074.html>  
<http://www.forschooner.net/blog-unkai/entry/000075.html>



## Column

## そのデータは必要ですか?

データのバックアップというと、なんとなく必要になりそうなものをすべて取り、いつまでも残そうとする事が多いのではないのでしょうか。すべてのデータを毎回取り、ずっと残しておくことが理想かもしれませんが、実際のところそういうわけにもいきません……。一般的にバックアップを取る対象を決めずにバックアップを取得すると、バックアップの対象データが膨大になるため、バックアップに長時間要する事態が予想されます。これは違う言い方をするならデータリストアが必要な障害対応に時間がかかるということです。

また、バックアップデータの保持期間や世代を決めず、いつまでもデータを残しておくことは、

ローカルストレージやストレージサービスのストレージ領域を圧迫します。

バックアップデータの保持期間や世代を決めないと、最初から大容量のローカルディスクが必要になったり、後からディスクを増設する必要が発生したりするなど、本来であれば必要のないリソースを無駄に使うことになってしまいます。ストレージサービスを利用している場合、データ量の増大=ランニングコストの増加につながります。上記の理由からバックアップに関してはできる限りバックアップ対象データを特定し、必要最小限のバックアップを心がけたいところです。



## 第4章

# クラウドでのバックアップの基本[AWS編]

柳瀬 秀明 (YANASE Hideaki) TwitterID:@oko\_chang /  
宮澤 慶 (MIYAZAWA Kei) TwitterID:@alan\_kei (株)サーバーワークス



## バックアップ先の大本命! Amazon S3

AWSをバックアップ先として使う場合、本命のサービスはやはり Amazon S3(以下S3)です。筆者がそう考える理由は次の理由によります。

- ・強い耐久性を誇る設計と実績
- ・従量課金
- ・簡単なアクセスと高速性

AWSで実装するバックアップの基本としてS3の知識はとても大事な要素です。まずは概要を説明します。

### ● 強い耐久性を誇る設計と実績

S3は99.9999999999%のデータ耐久性を誇ります。保存ファイルが1兆個を超えた現在でも「いまだファイルを1つも損失したことがない」という実績があります。この高いデータ耐久性を実現するために、S3では次のようにユーザがアップロードしたファイルが地理的に離れたデータセンターに同期されるように設計されています。仮にその内の1つのデータセンターでファイルが消失しても、自動的にS3はそれを検知し、ほかのデータセンターからファイルを復元します。そのため万が一、2つのデータセンターで同時にファイルが消失したとしてもデータそのものが損失されないしくみになっています(図1)。

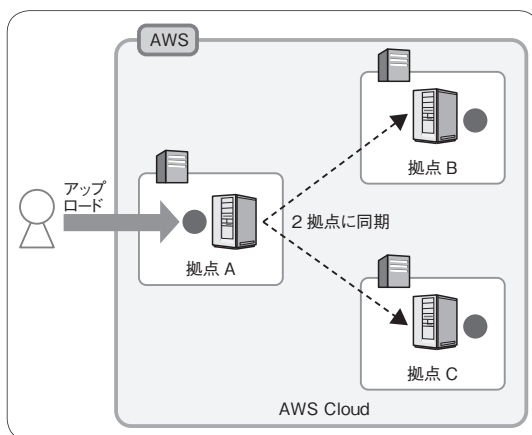
### ● 従量課金

システムが稼働する時間が長くなれば、次第にバックアップすべき対象のファイルが増える傾向にあります。しかし、システム稼働開始当初はそれほどバックアップすべき対象が少ないので、将来必要だからといって大きめのストレージを購入するのは高いリスクになります。その点でS3はストレージの使用量に応じた従量課金ですので、バックアップデータの容量に合わせて使用でき、コスト効率の良いサービスといえます。

### ● 簡単なアクセス

S3に対する操作はAWS SDKを使用できますし、さまざまなサードパーティのツールを利用できます(とても多くのツールがリリースされ

▼図1 データファイル損失から守る耐久性のしくみ





ています)。FTPのように使えるものから、CLI (Command Line Interface)で使うツールまで、いろいろなものが提供されているので、自分に合ったものを選びます。



## スナップショットによる バックアップの実際 (Linux、Windows)

AWS上でEC2インスタンスを使ってシステムを運用している場合に、一番手軽なバックアップはEBS(Elastic Block Store)スナップショットを取得する方法です。EBSはEC2インスタンスに接続させて使用できる仮想ディスクですが、ある時点の状態をスナップショットとして保存できます。さらにEC2は現在、EBSにシステム領域がインストールされているタイプのものが主流となっており、このシステム領域がインストールされたEBSもスナップショットを作成できます。作成したスナップショットはKernel IDやRAM Disk ID、名前などを指定することでAMI(Amazon Machine Images)として登録できます。AMIにするとそこからEC2インスタンスを起動できるようになりますので、Linux系のインスタンスであればスナップショットがお手軽なバックアップとなります。

しかしこれはあくまでLinuxの場合で、Windowsの場合は少し手順が異なります。WindowsインスタンスもEBSのスナップショットを作成できますが、システム領域となっているEBSボリュームのスナップショットからAMIを作成できません(スナップショットからインスタンスを復元する方法がないわけではないのですが、手順が煩雑で現実的ではありません)。

それではEC2上のWindowsインスタンスのシステム領域はバックアップする方法がないのでしょうか？ WindowsのEC2インスタンスの場合は、インスタンスからスナップショットを経由することなく直接AMIを作成すれば、バックアップができます<sup>注1)</sup>(こちらの手順はWindows

固有ではなく、Linuxのインスタンスでも同様の運用ができます)。

AMIを直接作成すると言っても実際にはスナップショットが取得されるわけですが、デフォルトでこのような動作となっているのは、EC2インスタンスを再起動することでファイルシステムのキャッシュやメモリの内容をディスクに反映し、さらにディスクアクセスがない状態でAMIを取得するためのようです。そのため再起動ができない運用中のインスタンスの場合は、no-rebootオプションを有効にしてバックアップすることになります<sup>注2)</sup>。



## スナップショットの世 代管理/Cloudworks

EC2インスタンスは、スナップショットやAMIを作成することで、手軽にバックアップができますが、毎日続ければスナップショットの個数が多くなります。スナップショットのしくみとして、EBSを複数保存した場合は差分が保存/課金の対象となります。保存世代分だけ料金が倍増するといったことはありませんが、スナップショットの数が多いといくら差分とはいえコストもかさみますし、管理の手間も増えます。スナップショットやAMIには任意の文字列などを使用できるので、実際の運用ではバックアップの日付を付与するなどして古いスナップショットの削除と世代管理をする必要があります。

こうした面倒を省き運用の手間を軽減するため、当社ではCloudworksというAWSの管理サービスにスケジュールジョブと呼ばれる機能を実装して提供しています(図2)。

Cloudworksのスケジュールジョブを使うと、EBSボリュームのバックアップやRDS(Relational Database Service)などのスナップショットを時間や曜日を指定して定期的に取得できますので、日々のバックアップ運用がとても楽になります。当社でも、当初はAPIを使い

注1) EC2インスタンスから直接AMIを取得する場合、Management Consoleから取得するとデフォルトではEC2インスタンスを再起動しますので注意が必要です。

注2) no-rebootでバックアップする場合もディスクアクセスが少ない時間帯での取得をお勧めします。



世代管理を行うスクリプトを自作してバックアップを行っていましたが、今ではCloudworksで快適かつ確実なバックアップを実現しています。また、執筆時点では開発中というステータスですが、Windowsのインスタンスに有効なEC2インスタンスを直接指定してAMIを作成する機能も2012年内を目標に鋭意開発中です。みなさんが本稿をご覧の時点で利用いただけたと思います。



## オンプレミスからAWSへ! Storage Gatewayによるバックアップ

### ● AWS Storage Gatewayとは?

AWS上のデータをバックアップするためには前章のとおり Amazon S3が最適ですが、AWSには「オンプレミス上のデータをクラウドにバックアップする」ためのサービスも用意されています。

### ● サービス内容、ユースケース

AWS Storage Gatewayは既存のオンプレミ

ス環境のデータを Amazon S3にバックアップすることを可能にする仮想アプライアンスです。

この仮想アプライアンスはiSCSI対応のストレージサーバとして利用でき、ストレージに書き込まれたデータをS3上に自動的に暗号化して転送を行うことができます。AWSからVMwareに対応した仮想マシンイメージとして提供されており、VMware上で動作しますので、簡単に稼働させることができます。Storage Gatewayの主なユースケースとして次の3点が挙げられます。

#### ▶ ① オンプレミスからクラウドへのバックアップ

既存のオンプレミス環境をバックアップする場合、一般的にはバックアップするデータ容量以上のストレージを用意する必要があります。Storage Gatewayの場合(必要なスペックを満たしていれば)一般的なサーバで環境構築ができて、さらにS3を使用するので、巨大なバックアップ領域を確保することができます。また、前述したようにS3はデータの耐久性が非常に高いため、バックアップ先のデータ保存基盤として非常に適しています。

### ▼ 図2 Cloudworks (<http://cloudworks.jp/>) スケジュールジョブ画面

スケジュールジョブの一覧				
			アクセスキー <input type="text" value="swx-dev-wild"/>	<input type="button" value="追加"/>
10件ずつ表示	絞り込み			
ジョブ名	アクション	スケジュール	次回実行日時	
① 日次バックアップジョブ	スナップショットの作成	月、火、水、木、金、土、日曜日の0時	2012-11-22 00:00:00	<input type="button" value="詳細"/> <input type="button" value="削除"/>
② 東京リージョン停止ジョブ	インスタンスの停止	月、火、水、木、金、土、日曜日の3時	2012-11-22 03:00:00	<input type="button" value="詳細"/> <input type="button" value="削除"/>
全 2 件中 1 件目から 2 件目を表示				
最初   前へ   1   次へ   最後				
Copyright © 2012 Serverworks Co., Ltd. All Rights Reserved. Icons by Mark James.				

### スケジュールジョブの詳細

削除

ジョブ名	日次バックアップジョブ
アクション	スナップショットの作成
リージョン	ap-northeast-1
オプション	<div>対象となるボリューム</div> <div>ボリュームIDが vol-39fba51a のもの</div> <div>作成するスナップショットに設定する説明</div> <div>90Ykm9id-daily-backup</div> <div>スナップショットを作成する世代数</div> <div>5</div> <div>取得したスナップショットにつけるタグ</div> <div>Key = Name , Value = test-backup</div>
スケジュール	月、火、水、木、金、土、日曜日の0時
次回実行日時	2012-11-22 00:00:00
<div>◀ スケジュールジョブ一覧へ戻る</div>	



### ▶②ディザスタリカバリー(BCP)

Storage Gateway を利用してバックアップを行う場合、バックアップ先は Amazon S3 となります。Amazon S3 を利用する場合、世界中の 8 カ所(2012 年 11 月時点)のリージョンから任意に選択して保存先を選ぶことができるため、日本以外のリージョンにバックアップを取るといった構成を簡単に構築できます。

これにより、ディザスタリカバリ用のデータセンターなどを用意する必要がなくなり、ハードウェア投資やメンテナンスコストを大幅に削減することが可能となります。S3 に保存されるデータは EBS のスナップショット形式で保存されるので、クラウド上でデータを復元することもできますし、オンプレミスの環境に復元できます。

### ▶③クラウドへのデータ移行

オフィスやデータセンターなどに存在する既存のデータを Storage Gateway を通してクラウド上に移行できるため、社内システムなどをクラウドへ踏み出す第一歩として利用できます。また、クラウドをバックアップ先として使用することに不安を感じるユーザも少なくないようですが、Storage Gateway ではセキュリティに関しても配慮された設計となっています。具体的には、データセンターと AWS との通信は SSL で暗号化されており、経路の安全性を確保。さらにデータ保存基盤である S3 側では保存されたデータは AES-256 を用いて暗号化されて保存されています。

もちろんバックアップデータへのアクセスは AWS で使用されている認証が必要なため、ほかのユーザがアクセスすることもできませんし、データが盗聴されたりする恐れ也没有ありません。Storage 用の iSCSI イニシエータでは CHAP (Challenge- Handshake Authentication protocol) を用いて暗号化認証を行うこともできます。



## Storage Gateway のセットアップ

### ● ハードウェアの用意

ここからは Storage Gateway (以降 SG) をセットアップする手順について解説をしていきたいと思います。まず、SG を動作させる物理サーバは、次のスペック条件を満たす必要があります。

- ・バーチャルプロセッサ 4 個
- ・最低 1 つの物理 NIC (iSCSI 接続と AWS との通信に使用)
- ・75GB 以上の SG 本体インストール用領域

その他、次のような環境が必要となります。

- ・SG に iSCSI 接続するサーバ (物理・仮想問わず)
- ・インターネット回線 (AWS へ接続可能な SSL 通信の経路)
- ・VMware ESXi を管理するために VMware vSphere クライアントを実行可能な環境

### ● VMware ESXi

前述のスペックを満たしているサーバには SG を動作させるハイパーバイザとして VMware ESXi をインストールします。Storage Gateway を動作させる VMware は ESX または ESXi のバージョン 4.1 以上である必要があります、今回は最新版を使用しています。VMware ESXi は VMware のホームページからダウンロードしてインストールを実行してください<sup>注3</sup>。ここまですをまとめるとネットワーク構成は図 2 のようになります。

### ● SG のインストールとセットアップ

次の手順で SG をインストールします。

注3) 本稿では VMware ESXi のインストール説明は割愛します。

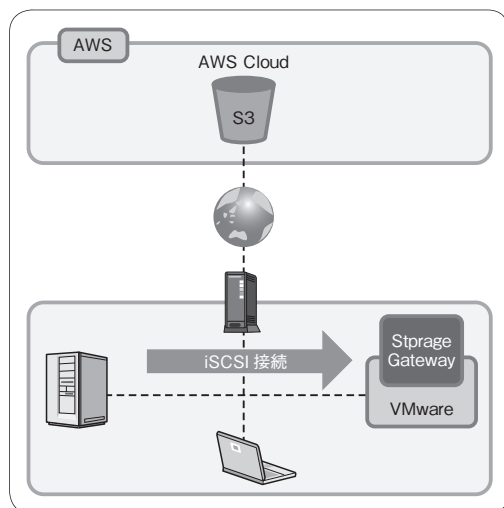


- ① AWSのマネジメントコンソールにログインし、SG管理画面に移動
- ② NavigationのDeploy a new Gatewayをクリックし、ウィザードを開始。SG仮想マシンテンプレートをダウンロードする
- ③ ZIPファイルを解凍し、VMwareのテンプレートであるファイル(本稿執筆時点ではAWS-Storage-Gateway-12-Oct-2012.ova)を使用
- ④ ウィザードによりSGのセットアップ手順が解説される
- ⑤ 手元のvSphereクライアントから、ESXiに設定したIPアドレスを指定してログイン
- ⑥ ツールバーからOVFテンプレートのデプロイを実行し、ウィザードによりSGのデプロイを実行

## NTPの設定

SGは、ホストOSと時刻を同期する必要があります。デプロイされたSGとなる仮想マシンのプロパティを開き、VMware Toolsのオプションからホストとゲスト時間を同期する設定をオンにします。また、ESXiのサーバは別途NTPサーバと時刻同期するような設定を行い時刻がずれないようにしておきます。

▼図3 Storage Gatewayのネットワーク構成



## 仮想ハードディスクの追加

SGをセットアップするうえで、次に必要なものは仮想ハードディスクの取り付けです。1つはiSCSI接続するサーバのボリュームとなる仮想ディスクと、後述するUpload Buffer用の仮想ディスクです。仮想マシンのプロパティを開き仮想ディスクの追加を行います。仮想ディスクは合計で2つ追加します。この時、SCSIコントローラが準仮想化になっているかご確認します。

## IPアドレスの設定

続いてSGにIPアドレスを設定します。vSphereクライアントのコンソールを表示するとユーザ名とパスワードが表示されていますので、こちらを使ってSGにログインしてください。ログインが完了すると、次のように表示されますので、2を選択します。

```
1: SOCKS Proxy Configuration
2: Static IP Address Configuration
3: Test Network Connectivity
4: Exit
```

すると次のように表示されるので、2を選択し、適切なIPアドレスを指定します。

```
1: View Network Configuration
2: Configure Static IP
3: View DNS Configuration
4: Reset to DHCP
5: Set Default Route Adapter
6: View Routes
7: Exit
```

IPアドレスの設定が完了したら、7を選択し1つ前の画面に戻ります。今回の構成ではプロキシサーバはありませんが、外部に接続する場合にProxyサーバを経由する場合はSOCKS Proxy Configurationを選択してProxyサーバの情報(IPアドレス、またはDNS名)を入力します。



## ● SGのアクティベーション

次に作成したSGをAWSに登録し、アクティブにする設定をします。ウィザードに従い、Gateway-Cached Volumes と Gateway-Stored Volumeの選択をします。Gateway-Cachedは最近追加された機能で、プライマリのデータはS3に保存しつつ頻繁にアクセスされるデータに関してはキャッシュとしてローカルに保存します。こちらの機能を使用することで、プライマリのストレージを大幅に削減しながらも使用可能なストレージ領域を多く確保できます。

Gateway-Storedはデータをすべてローカルに保存するようになり、データへの低レイテンシなアクセスが必要な場合に選択します。データのスナップショットを非同期にS3にバックアップし、災害復旧時にはローカルもしくはAWS上に復元できる高い耐久性を確保した構成となります。

今回はGateway-Storedの構成で設定してみます。引き続きウィザードを進めていくと、IPアドレスを入力する項目がでてきますので、こちらに先ほど設定したSGのIPアドレスを入力します。最後に次の項目を入力し、SGのアクティベーションが完了します。

## ● ストレージの設定

### ▶ Upload Bufferの作成

アクティベーションが完了するとManagement Consoleから作成したStorage Gatewayが選択できるようになります。GatewayタブのConfigure Upload Bufferから先ほどVMware側で作成した仮想ディスクの1つを選択して設定します(Upload Bufferの動作に関しては後述)。

### ▶ データ格納用のボリュームの作成

次にiSCSI接続される対象となる仮想ディスクを作成します。先ほどと同様にManagement ConsoleでStorage Gatewayを選択し、ボリュームタブからCreate Volumeをクリックし作成し

ます。Upload Bufferで使用したものとは異なる仮想ディスクを選択し、iSCSI ターゲット名を指定します。

以上でStorage Gatewayのセットアップは完了となります。実際にデータを送り込むには、別途用意したサーバからボリュームをiSCSI接続して、データを書き込むことになります。



## Storage Gateway でできること

### ● スケジュールによる自動転送

SGはボリュームのスナップショットをS3に転送しますが、こちらの転送間隔をManagement Consoleからスケジューリングして設定できます。デフォルトでは24時間となっていますが、こちらを1,2,4,8,12,24時間間隔で実行できます。最短で1時間設定、最長は24時間設定で自動的にスナップショットの転送を設定できます。実際の設定はデータの更新頻度など環境によって変わりますので、最適な時間を設定しましょう。

### ● 帯域制御

こちらもManagement Consoleから設定を行います。転送されるべき容量や回線の速度に応じて適切に設定します。これはデータ転送中でも、即時に適用されます。デフォルトの設定ではとくに帯域制限をかけていない状態となります。

### ● SGサーバの自動メンテナンス

SGにアップデートがある場合に、自動適用する設定です。自動アップデートのスケジューリングは曜日と時間で指定できます。なお、アップデート時は再起動が発生するので、再起動完了までしばらくの間使用できなくなります。そのため、アクセスが少ない時間を選んで、Management Consoleから都合がよい時間帯に変更します。



## Management Console の用途

上記のように Management Console で SG セットアップや設定作業ができます。また、スナップショットは Management Console から手動で取得できますし、スナップショットの一覧も閲覧できます。しかし、本稿執筆時点で「取得した SG のスナップショットを世代管理する」という機能はありませんので、世代管理を必要とする場合は別途実装する必要があります(図4)。



### 細かい仕様の説明

#### 必要な仮想ボリューム (boot 用、Upload Buffer、データボリューム用)

前述のように SG を動作させるまでには、いくつかの仮想ディスクを使用することになりますので、こちらの役割について解説します。

##### ▶ boot 用

SG 自身がインストールされている仮想マシン本体のボリュームとなります。OVF テンプレートをインポートして起動した段階で約 75GB 以上の領域が必要となります。

##### ▶ Upload Buffer 用

これはデータの差分を管理するために使用されるボリュームで、Storage Gateway を動作させるうえで非常に重要な役割を担います。スナップショットは差分で転送されることは前述しましたが、この差分領域が Upload Buffer に書きこまれ、スナップショットを取得すると差分が Upload Buffer から削除されます(図5)。

この Upload Buffer 用の仮想ディスクが溢れると、異常ステータスとなりスナップショットが取得取れない状態になります(PASS THROUGH)。この状況ではローカルディスクの読み書きには影響は出ないものの、S3 へのアップロードが停止していますので、サイジングやスナップショットのスケジューリングは注意してください。このボリューム容量は、スナップショットを取得する間隔に更新される容量に余裕を持たせて割り当てます<sup>注4</sup>。

##### ▶ iSCSI 接続用

iSCSI を経由して実際のデータが書き込まれ、AWS にバックアップされるボリューム領域です。このボリュームは 1 ボリュームにつき 1TiB (テビバイト) までという制限がありますが、これは AWS 上でリストアする場合に EBS で作成できるボリュームの上限が 1TiB までということに関係しています。SG 作成時に Gateway-Cached を選択した方は、1TiB 以上のボリュームを作成できますが、クラウド上にリストアする場合は同じように 1TiB という制限が発生します。また、作成後は簡単に拡張できないため、作成時には余裕を持って作成することをお勧めします。

注4) 不足した場合は Upload Buffer の増強か、ボリュームの削除、再作成して復旧できます。Upload buffer に割り当てるべき容量は次のような計算式で求められますが、実際に運用する場合は CloudWatch を使用して監視し、事前に検知できるように設定しておきます。

(アプリケーションスループット→AWS へのネットワークスループット×圧縮率)×書き込み時間=Upload Buffer  
例) ((40MB/sec) - (12MB/sec × 2)) × (12hours × 3,600sec/hour) = 691,200MB

▼図4 Storage Gateway の画面

Stored	
Volumes	Gateway
Snapshot Schedules	
Shut Down	Delete Gateway
Configure Local Storage	
Name:	Stored (edit name)
Gateway ID:	sgw-BDA247D4
IP Addresses:	192.168.43.109
Time Zone:	(GMT +9:00) Tokyo, Seoul, Osaka, Sapporo, Yakutsk (edit time zone)
Rate Limit on Upload to AWS:	No Limit (edit limit)
Rate Limit on Download from AWS:	No Limit (edit limit)
Upload Buffer Used:	0.00 KiB of 1.00 GiB (0%)
Maintenance Start Time:	Wednesdays 17:38 GMT+9:00 (edit time)



## リストアの方法とは

### バックアップからリストアまでの流れ

#### ▶ オンプレミス環境にリストア

オンプレミス環境にデータを戻したり、あるいはSGサーバが壊れてしまった場合は、別のサーバにSGをインストールして、それまで取得していたスナップショットから手元にデータをリストアできます。そのためには、構築したSGに同じサイズの仮想ディスクを作成し、SG管理画面からボリュームを作成する際に、リストアしたいスナップショット(たいていの場合は最新のスナップショット)を選択するだけです。

#### ▶ クラウド上でリストアする

SGで取得したバックアップは、AWS上でリストアできます(AWSでEC2を使用しているユーザにとって非常に簡単な操作です)。Management Consoleにログインして、EC2の管理画面から、SGで取得したバックアップの

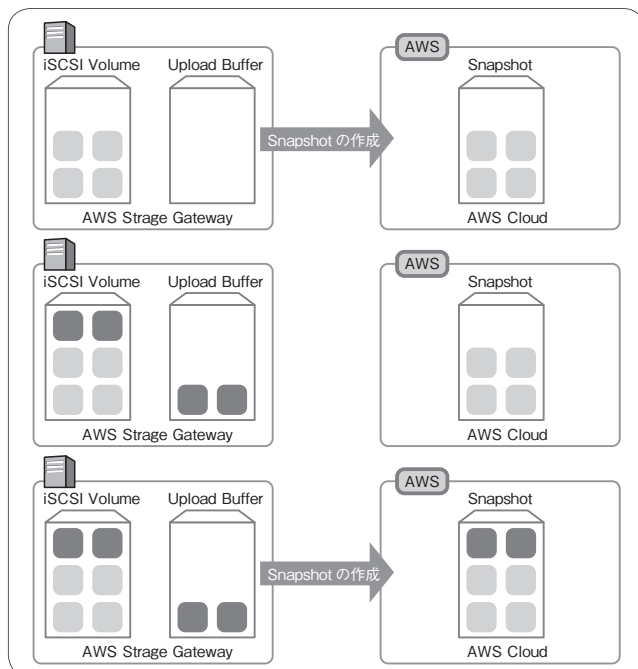
スナップショットにアクセスができます。リストアしたいスナップショットを選択して、こちらからEBSボリュームを作成すれば、オンプレミスで運用していたボリュームがそのままクラウド上展開されます。そして、任意のEC2インスタンスを起動してEBSボリュームをマウントすれば、今までのデータにアクセスできるようになります。このように簡単にデータをAWS上に展開することができますので、Storage Gatewayはクラウド移行に向けたファーストステップとして利用できます。



## まとめ

SGを利用することで、コストを抑えながらさまざまなシナリオに対応できるバックアップ環境を容易に構築できます。AWSというと、クラウド上で完結するイメージがありますが、SGのような「オンプレミスでも便利に使える」サービスを提供していることはあまり知られていません。業務でAWSを使えない方でも、こうしたサービスであれば利用できる可能性が出てくるのではないかと思います。本稿が、みなさんをAWSクラウドへ導くきっかけになれば幸いです。SD

### ▼図5 Upload Bufferのしくみ



## 第5章

知っているつもり?

# ハードディスクのイロハ

LXスタイル 杉田 正(SUGITA Tadashi) TwitterID: @sugipooh



## はじめに

パソコンのメモリは、電源をオフにすると記憶したデータが消えてしまうので外部にデータを保存／記憶させるストレージが必要です。そしてスピード、容量そしてコストに優れたハードディスク(以降HDD)が大昔からストレージの代表装置として使われているのは皆さんご存じの通りです。HDDはたくさんのデータを安価に保存できるのが最大の長特です。しかしHDDは突然「壊れる!」「遅い!」など思わぬ動作に戸惑うことが多いストレージでもあります。

HDDは非常に精密な装置ですが、使用方法のイロハを守らず寿命を短くしてしまうことも多く、壊れたときに、ユーザの責任なのか? 製造側、PC実装(PCメーカー)の問題なのか? はたまた寿命なのか? わからないことも多い装置です。HDDに寿命があることを知らない方も多くいらっしゃるでしょう。

本章ではHDD構造を理解して、故障発生を抑え、寿命を延ばすためのイロハを説明します。

### ● フォーマットはなぜ必要か

HDDを追加したり乗せ換えたりするときには、OSを使ってそのOS用にフォーマットします。

DVDレコーダなどにHDDが採用される時代になり、HDDの「品質」が画期的に良かったか

らでしょうが、最近のWindowsなどのフォーマットは、クイックフォーマットオプションが初期設定になっていて、1分ぐらいで終了します(図1)。

クイックフォーマットは、HDDの一部にOSの管理情報を書き込むだけです。HDD全体を検査せずに使って大丈夫なのでしょう? インターネットに接続してニュースを見たり、SNSにアクセスするだけならHDDの稼働時間は短いので寿命に達せず故障しないだけかもしれません。最近では4TBもの大容量HDDも販売されていますが、用途がビデオデータの保存なら、再生する回数も少なく故障しないだけかもしれません。

しかしサーバなどで使うならばクイックフォーマットでなく、物理フォーマットしましょう。物理フォーマットをすると、端から端まで特定のパターン(C3H、33Hなど)で書き込み検査がされます。HDDの容量やマシンの速さにもよりますが、おおよそ1時間から2時間の時間がかかります。物理フォーマットすることにより「書きにくい」「読み出しにくい」部分がないかを検査するのです。



## HDDをとりまく弱点たち

### ● HDDはエラーを出しながら動く

HDDは、平たい円板(プラッタ)に磁石の粉を



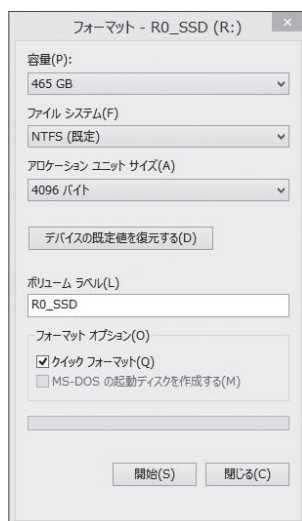
まぶした磁性板を超高速回転させ、超小型で強力な電磁石(ヘッド)が接近してN極S極を反転させながら書き込み、またその状態を読み出すストレージです。データはアナログな信号で記録されます。プラッタ上いっぱいの超小型磁石は、予期せぬ障害が発生し、反発してN極S極状態が変化してしまったりするとデータのエラーとなります(N極とS極の磁石は反発するって小学生の理科で習いましたよね)。ですから、工場で生産され検査されて出荷されても、搬送中に磁性が変化してエラーがないかフォーマットしてチェックしたほうが良いのです。

磁石の一部分が反転すればデータ内容が変わります。磁力の強さも超高速で読み出すため一定ではありません。たいへんな数の超小型磁石がHDDには収納されているわけで、磁石の状態が少し変化することはないとは言いきれません。しかし、少し変化しただけでエラーになってしまうと一般に売れる製品にはできません。

HDDは、超精密なアナログ技術で読み取られ

た信号を、高度なデジタル技術で補正して、アナログ的に発生するエラーがあっても読み書きするデータには影響が出ないしくみを持っています。HDDはエラーを出しながら動作しても、内部で処理して復旧したり、複数回書き込んだりしながら、エラーにならないようになっているのです。

#### ▼図1 クイックフォーマットオプション



#### Column

### データの保存に適したメディアは何か

テープストリーマ(バックアップ用のテープ)に記録されたデータは、数年経つと読めないリスクが高いです。ただし、テープストリーマを使うようなユーザは、重複して記録しているので問題が表面化しません。最近では扱うデータが多くなり、復旧時間のかかるテープは敬遠されつつありますが、HDDよりビット単価が安いので、まだまだ使われています。

CD-ROMよりCD-Rのほうがデータ保持年数が長いことをご存じですか。CD-ROMは保護に使われているアクリル材と浸透する湿気によってアルミ蒸着反射面が錆びて、20年ぐらいで読めなくなります。その点CD-Rは化学変化なので50年は読めると言われています。

個人用ストレージとしてUSBメモリが多く使われていますが、静電気障害で簡単に読めなくなるので長期保存には向きません。10年も長期保存すると読めないものが出てくるはず。思い出の写真をSDカードで保管してる人などは、将来読めなくなってしまう困る場合があります。

最終的には、重要なデータは重複して保存することになりますが、HDDは読み書きが高速なのでそういう用途には最適です。ただし、HDDは動作時の電力が大きいので、省電力なSSDなどとの組み合わせた階層ストレージをデータセンタで使うのが増えてくると思います。将来的に個人データはデータセンタに預ける時代になるでしょう。



## ● ゴツン(衝突音)

HDDのヘッド浮上技術はすごい精密技術です。髪の毛の直径が50,000nm( $10^{-9}$ m)ですが浮上隙間はなんと10nm。ウィルスのサイズと比べても1/10の隙間しかありません<sup>注1</sup>。

昔のHDDは、稼働中に手などで「ゴツン」と衝撃を与えるとディスクエラーが出ました。起動しないHDDなどは、電源を入れ、ヘッドが出てくるタイミングで「ゴツン」とヘッドが動く方向に衝撃を与えると復活した時代もありました。

しかし、最近のHDDはヘッドを支えるサスペンションが進化して、浮上高をフィードバック制御しています。昔のようにフリーで浮かせていないのです。構造的には、ヘッドの先には電熱器があり、温めるとヘッドが大きくなり浮きます。浮き過ぎるとその電熱器をオフするといった感じで制御しています。「ゴツン」の程度や方向にもよりますが耐G(耐加速度)は格段に向上しています。

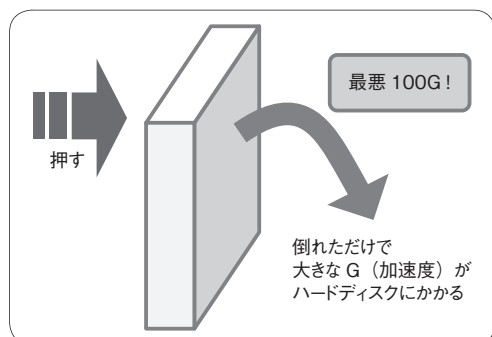
## ● 耐G

HDDはどれぐらいのG(加速度)に耐えられるのでしょうか? Seagate社のHDD「Baracuda」の仕様書には動作時80Gに耐えると記載されています<sup>注2</sup>。しかし図2のように机の上でパタン

注1) 参考資料: CQ出版社、『改訂 ハード・ディスク装置の構造と応用—記録/再生の原理とメカニズム&インターフェース』ISBN978-4789845441

注2) <http://www.seagate.co.jp/jp/ja/internal-hard-drives/desktop-hard-drives/barracuda/>

▼図2 大きなG(加速度)がかかる



と倒すと瞬間的に100Gを超える場合があります。

最近のHDDは非動作時の耐Gも大幅に上がっています。これは、通電していないときの取り扱いがHDDのエラー発生や寿命に影響するとわかり、強化されたからです。Baracudaの非動作時の耐Gは300Gとなっています。それでも1mの高さから落下させると300Gを超える場合があるので、外見は壊れていなくてもHDDは壊れている場合があるのです。

サーバに使うなら、1回でも「ゴツン」と鳴らしたHDDは使わないようにしましょう。そうすると数ヵ月後のハードディスクエラーが減っていくのです。

## ● プチプチ

HDDは精密機械とご理解いただけたなら、使っていないときにもプチプチに包んで保管してください。そうしておけばちょっとのことでは、カタログに記載されている耐Gを越えなくなります。ただし、白いプチプチは静電気対策がされていないので使用厳禁です。ピンクもしくは青色のプチプチを使ってください(写真1)。

白いプチプチに包んで、ひょうたんと言われる発泡スチロールを詰めた段ボールでHDDを納品してきた業者がありました。筆者は全品返却しました。

▼写真1 静電気対策されたプチプチと静電気防止袋、ケース





## ● 静電気ストレス

最近のHDDに搭載されているICも静電気に強くなりました。モータードライバなども混載されたICも見かけます。しかし、端子を手で触るなど静電気ショックを与えると、静電気ストレスが蓄積され、故障の原因になります(写真2)。

静電気対策されていない床材なども多く、柔らかくてもけっしてモケットタイルカーペットの上などに直接HDDを置いてはいけません。冬の乾燥した時期などはとくに危険です。

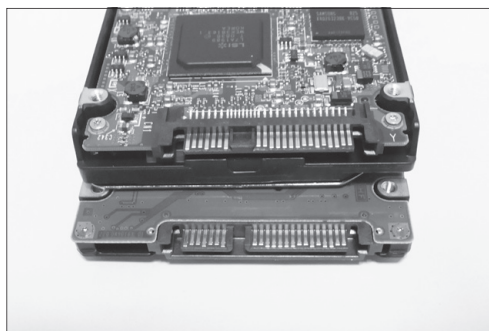


## HDDに潜む危険

### ● 危険な最初の1ヵ月

HDDの1年以内の故障率が1%から3%以下とすると、100台に1〜3台が1年間で故障することになります。

#### ▼写真2 端子を触るな



とになります。

しかしこの故障する時期は、最初の1ヵ月間に半分、残り11ヵ月間に半分くらいの割合です。最初の1ヵ月はとくに故障が多いのです。新しいパソコンを使い始めて、やっと設定なども終わった時期に壊れるのですからユーザは困った状況に陥ることがあり、大きなクレームになる場合も多々あります。販売店やメーカーも失くしたい故障でもあります。この1ヵ月以内に故障する初期着荷不良品を、工場から出荷する前に排除できれば良いのですが、まだまだ検査を抜けてくる製品があるようです。

HDDがDVDレコーダなど家電製品に内蔵されるようになってから、品質は格段によくなっています。製品保証期間が、3年間と長期保証になっている製品も多くなっています。

### ● ロット不良

HDDはその動作原理から「もうこれ以上性能を上げられない」と言われ続けながら40年も長くストレージの主流として使われてきています。多いときには1週間に日本だけで数十万台も販売される大量生産品です。

それゆえにメーカー間競争もきつく、市場からの容量アップやアクセス速度向上など性能アップ要求があるにもかかわらず、コストダウン要求も強く、時々コストダウンのために採用した部品の耐久性が弱く、大量のリコールを出したりします。

ファームウェアをアップデートすればバグを



## Column

### 免震と耐震

日本は地震の多い国ですから、データセンターは「耐震強度」をもった強固な建物が多いです。しかし、免震は必要なのでしょうか？ ラック据え付け、サーバ取り付けにて耐震性を持たせておけば、地震時にサーバなどが飛んできて人身事故が

発生することもないでしょう。最近のハードディスクは動作時の耐Gも大きくなり、大きな加速度で壊れない機構を持つので、HDDエラー対策の免震は不要です。大事なのは耐震強度です。

修正できることもありますが、パソコンのソフトウェアのようにインターネット経由で書き換えができるわけでもありませんし、アップデートはユーザーデータを失くすリスクもあります。ですから、いったん出荷すると簡単にアップデートできません。



## HDDのうんちく

### ● 石鹸でいつもツルツル

HDD内部のメディア(プラッタ)には、錆びないように界面活性剤(石鹸と同様な液体)が薄く塗られています(写真3)。なんせヘッドは10nmの低空を超高速で飛行しているので、この界面活性剤が凸凹しているとヘッドが衝突して故障になります。最近の機種には、さらにもう1層ヘッドが落下してきたら“はじく”界面活性剤を塗ってある機種もあるそうです。

ここに化学薬品を使っているのがHDDの寿命に関係してきます。高温になって蒸発したり、低温で固まってヘッドが衝突したりするのです。

軸受けはボールベアリングから流体軸受けに進化しているので、ボールベアリングの油切れの心配はなくなりましたが、低温での動作には弱くなっているようです。

### ● 暑いのも寒いのもキライ

HDD内部は密閉されていません。内部に水滴

が溜まり結露などを起こさないようにホコリフィルタを通して呼吸しています(写真4)。

連続したシーク動作などが続くで発熱量も多くなり、冷却方法は寿命に大きく影響します。

2006年ごろ発表されたGoogle社の論文で、10万台運用において40℃程度が安定していると発表がありました。当初の翻訳が「熱いほうがHDDに良い」と誤訳され、未だに信じている方とたまにお会いします。本当の答えは「温度を安定させて運用すれば故障率が下がる。温度を一定にするには40℃付近が都合良い」です。

界面活性剤の性能も上がり、動作保証温度は50℃を超えています。Seagate社2.5インチエンタープライズ用HDDは5～55℃が動作温度範囲です<sup>注3</sup>。

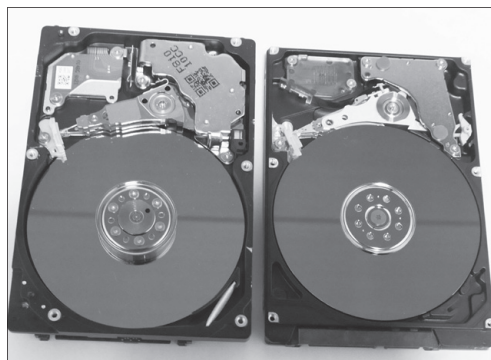
内部メディアはピカピカですから、水分が入り込むとヘッドが貼り付く心配もありますので、急激な温度変化は危険です。HDDは汗かきで寒がり、ツルツル同士がピタッと貼り付く湿気がキライなのです。

### ● 冷却

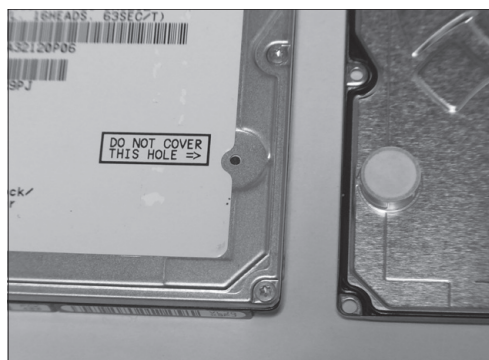
最近のHDDは消費電力が下がり、界面活性剤の性能も上がったので冷却条件は緩くなっています。2.5インチであれば、密閉ケースに入れても使用できます。

3.5インチでもベアードで使う装置が販売されて

▼写真3 HDD内部(2.5インチ)



▼写真4 この穴を塞いではいけない



注3) <http://www.seagate.co.jp/jp/ja/internal-hard-drives/enterprise-hard-drives/2-5/sawio-15k/>



います。いずれもファンを必要としないとされています。

写真5はHDDを「生」で扱うUSBアダプタです。冷却ファンなしで使うのですからユーザー自身のリスクで使うアダプタです。便利ですので筆者も使っています(笑)。

しかしHDDの種類によってはモータ回転数が高く、発熱したり、シークが多いアプリケーション(データベースやメールサーバなど)を動作させると過熱します。過熱する場合ファンにて適正な温度になるように風を当てて冷却します。

## NAS

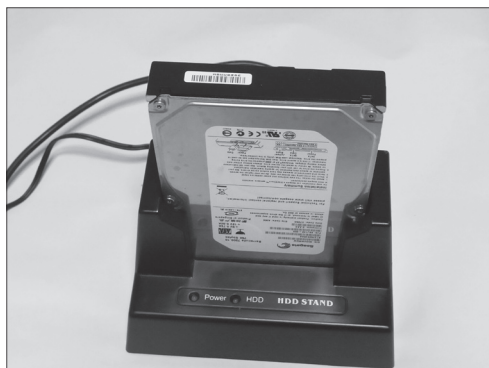
ネットワーク越しにHDDへアクセスできる

NAS(Network Attached Storage)が一般化しています。ファイルサーバとしての利用すると、ストレージの共有化が簡単に行え、高機能を発揮します。Wi-Fi機能を内蔵して、スマートフォンからアクセスできる製品もあります。NASは24時間連続で利用する場合が多く、冷却は必須です。容量の大きさから消費電力の多い3.5インチHDDを使う場合も多く、電源が発生する熱処理も行うのでそれなりの風量、静圧を持つファンが使われています(写真6)。

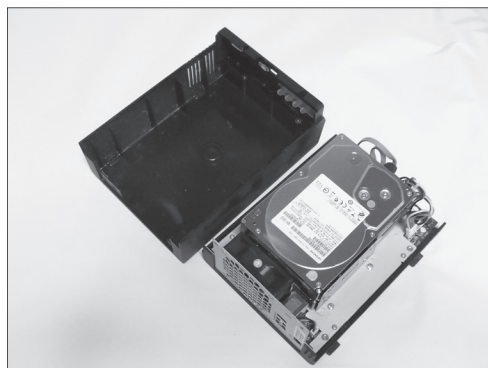
小さなファンで風量を確保すると騒音が気になります。大きなファンでゆっくり回転させると、静かに省電力にHDDを冷却できます。

冷却すればHDDは過熱せず寿命が短くならず

▼写真5 3.5インチ用USBアダプタ



▼写真6 3.5インチ2台構成NAS



## Column

## CAV、CLV、ZCAVとセクターサイズ

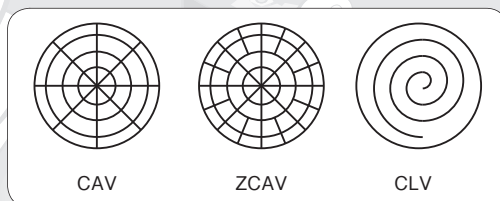
今でもLinuxなどで物理フォーマットしようとすると、セクター数などの入力を要求されます。しかし、同心円状のCAVは、すでにHDDでは採用されておらず、外周で密度が低くならないZCAVが主流です。しかしZCAVのHDDでも、HDDコントローラがCAVとしてふるまってくれます。親切に入力するデータをHDDに貼ってある機種もあります。

セクターサイズも、HDDは長い期間512B /Sectorでしたが4TBを超えるHDDで4096B /Sectorの利用が始まっています。SSDは1024B

/Sectorですのでハードディスク用ドライバは基本的に使えません。

CD-ROM、DVDなどは渦巻きのCLVです。

▼図3 CAV、ZCAV、CLV



良いのですが、家庭などで使うと、今度はホコリが問題になります。フィルタを付けると新しいときは良いのですが、使っていくと詰まり冷却できなくなる不具合のほうが多くなると考えられます。写真6のNASは筐体を工夫してフィルタなし構造になっています。



## RAID

### ● RAIDにすると故障が増える?

たくさんのアクセスがあるファイルサーバに使われるHDDは故障確率は高くなり、故障からデータ損失を防ぐためにRAIDによる故障保護機能を使われている方々も多いと思います。

RAID機能付きNAS(写真7)や、サーバにRAIDコントローラを内蔵した製品もありますが、それらは冷却性能もよく、ホコリ対策もなされていて手軽に大容量ストレージをネットワークで共用できます。

しかしながら、故障しない保証はなく、数年利用して突然アクセスできなくなり困った経験を持つ方々も多いでしょう。

▼写真7 RAID機能付きNAS  
(BUFFALO LS-QV1.0TL/1D)



RAIDはHDD単体で使うよりコントローラなどパーツが増えるので「故障確率が増える」製品です。とくにRAID 5モードで使い、1台目のエラーを放置して2台目がエラーを出すと、RAID全体にアクセスできなくなるので被害が大きくなります。RAIDでもバックアップは必須なのです。

安価なRAID装置でもバックアップを容易に行える機能が付いています。必ずバックアップしましょう。

### ● RAIDはミラーモードがお勧め

コントローラ性能が上がりRAID 6を使える製品も増えてきました。HDD 8台ぐらいでRAIDを構成するならRAID 6が使い、2台が連続してエラーを出してもアクセスできます。2台がエラーを出したものを放置して3台目がエラーになってしまうとアクセスできなくなります。当たり前のようですが、よく起きる事件でもあります。RAIDには、HDDのエラーが出たらメールなどで報告する機能やブザーで知らせる機能が付いています。そんな場合は警報を無視せず、ただちにメンテナンスしないと“騒ぎ”になります。

復旧短時間を要求されるレンタルサーバなどではRAID 5、RAID 6は禁止です。長時間サーバサービスを停止させるリスクが高く、1台のエラーを修復するリカバリ動作のときにコントローラパフォーマンスが低下して、データベースなどが極端に遅くなります。

RAID 1(ミラー)の場合、最悪コントローラが壊れてもサーバなどに直接接続すれば読み書きできる場合が多く、サーバサービスを短時間で継続できます。大容量が必要なときには、RAID 6をミラー化して使うRAID 61モードなどを使いましょう。

いずれにしても、リビルド(縮退モードとも言います)時のコントローラ低パフォーマンスでの性能評価は必須です。



## ● インターフェース

最近のHDDインターフェースはSATA (Serial Advanced Technology Attachment) が主流です(写真8)。6Gbpsの新規格も普及してきました。しかし、この3Gbpsとか6Gbpsと言うスピードはとても速く、30年前は導波管<sup>注6</sup>を使っていたスピードです。

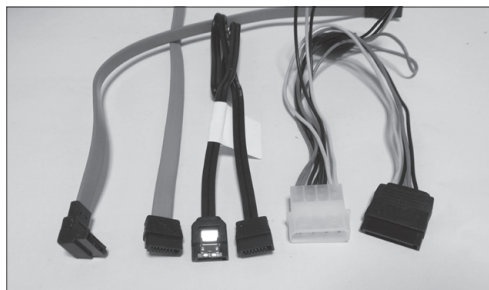
とても速いインターフェースが非常に安価に使えます。L型とかロックできるとか、外部へ引き出すeSATA(external Serial ATA)などの拡張された規格があります。

SATAは1対1で接続するインターフェースですが、連結して使えるSAS(Serial Attached SCSI)などもサーバでは使われています(図4上)。

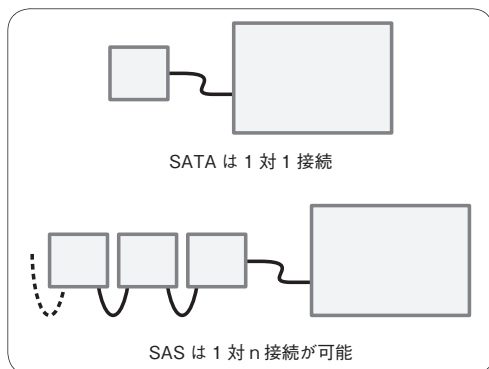
SASホストアダプタでは、SASドライブを連結(ディジーチェーン)して大量のSASドライブを接続できます(図4下)。またSATAドライ

注6) 光や電磁波を伝えるための管。その1例が光ファイバ。たとえば古過ぎるか……(V\_V)

▼写真8 SATAケーブルと電源ケーブル



▼図4 SATAとSASのHBA接続台数



ブを使用できる上位互換性を持っています。6Gbpsで100台以上連結できるのですからすごいです。

サーバでのHDD稼働時間は、個人所有ノートPCでの稼働時間と比べて3倍から5倍以上なので、故障する(=寿命)も1/3から1/5の時間になります。メールサーバのように小さいファイルを多量に扱うのはHDDにとっては過酷なことです。そんなときはサーバ用に設計されたSASドライブの出番です。

## ● 電源

HDDの動作には電源が必要です。電源が不安定になると、HDDに悪影響が出て正常に動作できなくなりますので注意が必要です。

通常の回路では電源リップル(直流がデコボコな波形になる)が少ないことを前提に組んであるので、寒いと「不安定」という難儀な状態になります。たとえばパソコンを冷蔵庫に入れておいて取り出したりすると、結露によって電源が通電し、壊れたり劣化したりします。

夏の炎天下であなたのパソコンは起動しますか？ 暑くても日本では気温40℃ぐらいですので大丈夫です。これら気象条件に影響されるのは「コンデンサ」です。交流をダイオードで整流してデコボコな波形を平滑化するのも「コンデンサ」の役目です。「コンデンサ」が性能を発揮しないと、電源リップルを含むようになったり、故障したりします。

サーバやルータなどは、24時間連続稼働なので通常のパソコンより長時間稼働するので短期間で寿命がきます。しかしそれでは都合が悪いので、二重化(冗長化)電源などを搭載できるようになっています。

しかし、よく考えてみると二重化(冗長化)電源は一般的にはあまり見かけません。……そうです、電源設計が悪いのです。設計でコンデンサ環境温度を間違っている場合が多いのです。55℃が動作保証最高温度だったコンデンサは、5℃高い60℃で使うと寿命が半分になると仕様

書に書いてありました。しかし、カーナビのような過酷な高温条件に耐えるコンデンサが開発され、最近のコンデンサは65℃以上に耐えるのが標準になっています。



## HDDをとりまく 最近の技術

### ● クラスタ

Windows Serverなどを冗長化する場合、ク

ラスタがよく採用されていました。SCSIなどの接続口を2つ以上持つRAID装置に、2台のサーバを接続してハートビートケーブルでつなぎ、マスタサーバが停止するとスレーブサーバが代わりに稼働するシステムです(図5)。

このクラスタシステムは実績のあるシステムですが、RAIDコントローラがSOPF(単一障害ポイント)になるので、コントローラが二重化されたFC RAID(FibreChannel RAID)を使うなどRAIDが高コストでした。FC RAIDを共用す



### Column

## サーバ機で使われるトレイ

RAID機能付きNASやサーバ機では、HDDが故障した時に容易に交換できる(写真9)ようにHDDをトレイに取り付けて実装されています(写真10)。

このトレイには相反する問題を解決する工夫が必要です。

#### ①容易に取り外せること

→動作時しっかり固定

#### ②冷却風がたくさん通る

→VCCI<sup>※4</sup>対策で穴を大きくできない

#### ③実装を高密度にする

→熱が溜まる

サーバには高性能なストレージが要求されるの

で、HDDも高回転で熱をたくさん出す高性能な機種が使われることが多く、寿命に影響する冷却は大事な管理事項です。高密度実装を実現しようとすると、容易に取り外しでき、振動に強い固定機構は難しい構造問題です。同じ水晶クロックを使うため“共振”などにも注意が必要です。

VCCI対策は電波漏れを規定していますが、電波障害を“受けない”規格でもあります。

サーバ専門会社にも関わらず、HDDトレイのレーンに樹脂成型品を使っていた会社がありました。高回転なハードディスクを冷却できず短寿命となり、数年後大量のHDDエラーを出して業績が悪化しました。

▼写真9 HDDを交換しているところ(BUFFALO LS-QV1.0TL/1D)



▼写真10 HDDのトレイ



注4) Voluntary Control Council for Interference by Information Technology Equipment



るためのスイッチも高価でした。

FCインターフェースは、SCSIを光ファイバ用に進化させたインターフェースです。SCSIと同じくチャンネル通信であり高速です。しかし、レーザーダイオードには発光寿命があって数年でエラーを出します。クラスタで使わず単独で使用するFC RAIDは、エラー情報が収集できるメーカーならともかく、一般Sierには取扱いが難しく一般化しませんでした。

FCはSATA、SASへと進化しています。

## ● iSCSI

Ethernetが高速になっても、Ethernet上のHDDをデータベースなどから使用するにはたいへん遅く実用的ではありませんでした。FCを使い直接ネットワークする製品もありますが、高価でした。

Ethernetを使いSCSIのようにストレージを使う規格がiSCSIです。Ethernetは待ち時間が長いとパケットを捨てるので、チャンネルネットワークと相性が悪いものでした。iSCSIの登場は15年以上前で、専用アクセラレータTOE(TCP/IP Over Engine)なども開発されましたが、クラスタなどでも満足な速度が出ず普及しませんでした。

10GbEの登場により、やっと実用的な速度で

使える時代となり、インターネットに関する技術標準を定める団体であるIETF(Internet Engineering Task Force)が正式に発行する文書(RFC: Request for Comments)に採用されストレージネットワーク標準になってきました。

10GbE NIC(Network Interface Card)やHUBも値下がりしてきて使いやすい時代になってきました。

TCP/IPですから、通信経路である光ファイバ折れやSFP+(Small Form-Factor Pluggable Plus)と呼ばれるメディアインターフェースのレーザーが不調になっても、スイッチとバルク(二重化)で接続しておけば安全です。

## ● オブジェクトストレージ

ネットワーク経由でストレージを扱う場合、HDDだけでなくファイルシステムも持って、必要ときにHDDを追加すると容量が増加する機能や、並列化してアクセススピードを向上させるなどの機能を提供する製品があります。このような機能をオープンソフトで構成できます。時代はどんどん変わってきています。

## ● バックアップ

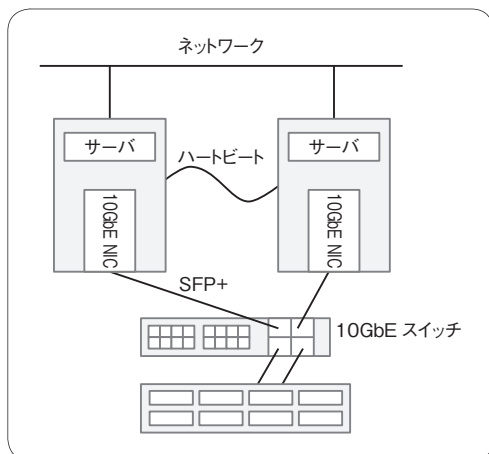
USBメモリから、HDDを高機能に使いこなすオブジェクトストレージまで、ストレージと呼ばれる製品は多種多様にあります。しかし、世界で唯一のデータを1ヵ所に置き、そのストレージが盗難や水没してしまうとデータは戻って来ません。

バックアップが重要だとわかっているても、安定して使えているストレージに対して追加の投資は難しいです。

ストレージ自体を、ネットワークを経由してcopy、dd、rsyncなどOS標準コマンドで簡単に複製できるので、大事なデータは離れた場所に保管するのがバックアップの基本です。

筆者のお勧めは激安NASの三重化分散配置。当然NASはRAID 1(ミラー)設定です。

▼図5 iSCSIを使うクラスタ



## DRDB

DRDB ( Distributed Replicated Block Device)はHDDのセクター単位(ブロック単位)でバックアップするソフトウェアでrsyncと使い分けるとHDD全体のバックアップなどに有用です。

## 重複排除

比較的新しいストレージ技術に「重複排除」があります。たとえば社内サーバには各個人データが収納されていても、もともと同じデータを収納している場合があります。このようなデータを探してインデックスだけ残し、重複したデータを消去します。

これを実行すると、一般的に容量は1/3以下になるようで高価なオブジェクトストレージの使用容量を増やすことができます。

インターネット越えのバックアップなどでも使える場合があり、バックアップ時間を大幅に短縮できます。



## HDDのテストは難しい

HDDは内部でエラーが出て、それを外部に出さないしくみを持っているので故障を再現できない場合が多いです。1年に1回とか出るエラーが再現できないので、その対策をとるのがたいへん難しいのです。

長期間サンプルテストして合格しても、本番製造での品質バラツキが多いHDDの事前検証を簡単にできると思うのがおかしいのです。

エージング(安定動作するまで仮動作させる)をまじめにやっていると人的コストがかかってしかたありません。そういったまじめなエンジニアを、仕事が遅いとか低能と営業担当は誤解します。まじめにエージングして故障しなければ「当たり前」なのです。インフラ管理は「何も起きない」ように作業するのが仕事ですから。これがわからない人たちが多過ぎます。安から

う悪からうが良いのでしょうか？

ネットワークではRFCという文書があり、それに従えばちゃんと動きます。しかしストレージの世界は競争が厳しいので自社技術を公開しない場合が多いです。SCSI-2の規格通りに作ればRAIDは動かなかったでしょう。ストレージのテストは難しいのです。



## 最後に

儲かるのはアプリケーションの近くで、間違いないです。だから組織のトップは偉いわけです。でもその儲けを支えるのはインフラです。

ほとんどの人はHDDやサーバが壊れることを知っていれば良いのです。故障を予測できるS.M.A.R.T.(Self-Monitoring Analysis and Reporting Technology)コマンドなどがあることを知っておけば、なお良いでしょう。利用するサーバ提供会社を選ぶときの「目」を持つには必要なことでしょう。

でもそれを管理することを仕事にしている人には必須な知識です。「サービスの品質」を維持するには対策を手順化して多数の仲間意識を同調して対応しなければいけません。

サーバ管理の極意は「寿命のある部品を管理する」ことです。自動車業界じゃ当たり前ですよ。

最後に……EthernetとACラインを25mm以上離してください。結束バンドで縛ったりしたらダメですよ。ちゃんと規格化されています。

それから案外と知らない人多いようですが、“ACコードを丸く巻いてはいけません”。折り返して束ねるのが正解です。電磁誘導でよからぬ「過熱」が起きてしまいます。気をつけましょう。SD



## Column

## リスク管理

バックアップは大惨事になってから予算が付くようになる場合が多いです。

### ▶ 懲りないと改善は難しい

週末フルバックアップ、毎晩差分をとって、フルバックアップしたものを平日日中にテープに書くようにしておけば、HDDが全部お亡くなりになっても週初の状態には戻れます。しかし、多分どこもそうですが、一度懲りないと予算が付かないのが現実です。

米国の医療データを預かる会社は、大陸間バックアップを行っています。超大手企業が導入しているデータセンタミラーは、ボタン1つで10万台のサーバが切り替わるそうです。これらにコストを掛けるのはちゃんとバックアップなどの投資をリスク管理として投資しているからです。

大手レンタルサーバ会社でデータ消去事故が起きましたが、本当の教訓は「バックアップは大事だよ」ではなく「リスク管理は大事だよ」です。

データ保全に限らず、さまざまなバターンの障害を想定したテストと復旧手段の確認、障害時の代替手段の用意、サーバの切り替えテスト。せめてこのくらいまではやらないとサービス維持は難しいのです。運用事故は、このような事前に人的コストを負担しないセキュリティ管理の問題と考えます。

どんなにコストをかけても最悪な場合は、データロス事故がゼロにならないことをご理解いただけると「どれだけのコストをかけて守るか?」になります。

### ▶ 経営責任者の判断

セキュリティに絶対はないので、ISMS(ISO 27001)では、対策に係るコストを表にして、経営責任者にどんなリスクが存在して、「どれだけコストをかけるとリスクをどれだけ減らせるか?」の判

断をさせます。「価値のあるデータと価値が低いデータ」「必須なのに設備していない設備」「世界で1つしかないデータとみんなが持つデータ」「バックアップを分散させると管理がたいへん」。C.I.A.(Confidentiality、Integrity、Availability)理論そのものです。

管理レベルを上げるにはPDCA(Plan-Do-Check-Act)で上げる。製造する製品品質を上げるためによく工場で採用されているISO9001管理と同じです。たとえば発電機はあれば良いですが、高価な買い物です。手持ち資本が減るリスクと発電機がなくて予想される発生被害額とのバランスを社長が判断します。このような判断は社長しかできません。

### ▶ ISMSで業務の見直しを

運用などのサービス事故は、結局組織責任者の問題です。ツールが悪いのもエンジニアの責任ではありません。ちゃんとチェックさせない経営の問題です。情報流出とかウィルス感染といった問題が起きたときに担当責任者が居ないとかも組織責任者の問題なのです。

ISMSはPDCAで業務を改善するしくみです。マーク取ったら終わりにしてしまうから、事故が起きます。

大手レンタルサーバ会社の事故は重大なISMS違反です。継続審査に問題があったかもしれません。

みなさんインテルサーバというパソコンをサーバに使っていることを忘れています。予備品が安価に購入できるので、メンテナンスに係る工数人件費を下げる工夫ができます。しかし何も起きないからといって予備品を購入しなかったりしたら、故障を速やかに復旧できなくなります。現場のモチベーションが下がり、それがさらに事故を呼んでしまうのです。SD



## 第6章

# 復旧業者に聞く データ復旧の実際

聞き手：Software Design編集部

協力：(株)ワイ・イー・データ <http://www.yedata.co.jp/>  
オントラックデータリカバリーサービス <http://www.ontrack-japan.com/>



## 物理障害

データが読めなくなって自分では復旧もままならなくなったら業者に依頼することになりますが、実際にどのように復旧しているのでしょうか。そこで本章では世界最大級のデータ復旧会社の米国オントラック社<sup>注1</sup>と提携してオントラックデータリカバリー(写真1)を提供している(株)ワイ・イー・データ(写真2)に復旧作業の実際の作業についてお聞きます。

SoftwareDesign編集部(以降S) さっそくですが、こちらに来る依頼はHDDがほとんどでしょうか？ どんな案件が多いのですか？

ワイ・イー・データ(以降Y) こちらに来るのは8割くらいがHDDの物理障害のものです。大別して、HDDの障害には物理障害と論理障害があ

注1) <http://www.kroll.com/krollontrack/>

### ▼写真1 オントラックデータリカバリーサービス



ります。物理障害とは電氣的、機械的に破損しているものです。論理障害はファイルの誤削除や誤フォーマットなどHDDのアクセスには問題ないものです。

S 御社のHP上でも、障害をある程度区分けして分類するような話がありますが、具体的にHDDの種類による復旧の差はあるものなのですか？

Y あります。やはりSCSI(Small Computer System Interface)やSAS(Serial Attached SCSI)、FC(Fibre Channel)は部品単体の精度が高いので故障率自体も低いですし、故障したとしても最初にリードエラーが起きて止まるケースが多いんです。その状態を過ぎてしまうとないがクラッシュとなりデータが復旧できないケースが多いです。ヘッド(写真3)がプラッタ(磁性体の円盤；写真4)に触って磁性膜が削

### ▼写真2 (株)ワイ・イー・データ本社





れてしまった状態になってしまうとさすがにデータ復旧を行うことはほとんどできません。

電気的な故障はIDE(Integrated Drive Electronics)やSATA(Serial Advanced Technology Attachment)が多いです。IDEとかSATAは部品の精度が1桁低いのでいろんな故障があります。基板が焦げたりヘッドの回路がおかしくなったり、ありとあらゆる故障がすべて起きるのがエンタープライズモデルとの違いです。基板に表面実装されている部品を見てもSCSIやSAS、FCはコストをかけていますし、IDEやSATAはローコストモデルなのでどうしてもそういうところで障害が発生しやすいです。

**Q** 物理的な障害がわかるのは、やはり音とかですか？

**A** 異音が発生している場合はほとんどが物理障害なので、すぐに専門業者に出した方が良いでしょう。音がしなくなったりした場合は、比較的基板のほうに障害が起きていることが多いです。個人のホームページでは復旧のために基板を載せ替えたという成功事例が載っていたりする場合がありますが、基板を替えたまま送られてくる例もありますが、近年のHDDは基板交換で動作するケースはあまりないので交換しない方が良いと思います。ほかの業者の戻し忘れなど、オリジナル以外の部品になっている場合は作業自体が複雑になります。

▼写真3 ヘッド



**Q** ほかの業者でダメだったものが持ち込まれるケースって多いんですか？

**A** けっこうあります。2~3社またいでいるようなケースも希にあります。技術的に高い評価をいただいているので、こちらに持ち込まれるのは最後の砦的な感じです。

**Q** HPに、HDDを開けてしまったら復旧率が下がるといったことが書いてありますが、かなり影響されますか。

**A** HDDの設計自体がシビアですので禁則事項ですが、開けてしまったから持ち込まれても、復旧できたケースもあります。HDDを開けただけならまだ良いのですが、ディスクのプラッタ面とかヘッドを指で触られてしまうと厳しいです。他業者を経て来たものにそういうことが多々あります。HDDを開けるのは専用工具が必要ですから個人で開けてしまうケースは少ないです。以前に、プラッタやヘッドなどが全部がパーツで送られてきたことがあるんですが、これはちょっと無理だろうという感じでした。1枚もののプラッタならばまだ可能性はあるのですが、2枚以上になるとそれぞれの角度とかもあるので、ずれてしまうと読み取れなくなってしまいます。持ち込まれるHDDの中には火災とか水没といったものもあるので、そういったものは開けたとかいうレベル以上に汚染されていますが、

▼写真4 プラッタ



そういうのもかなりの率で復旧した実績があります(後述)。



## 復旧時の読み出し テクニック

**Q** 物理障害のHDDは大まかにどのような扱いになるのですか？

**A** FAT(File Allocation Table)や MFT(Master File Table)などが読めれば、情報のある部分だけのイメージをとってサーバ(後述)に上げてそれを処理するという形をとっています。HDDの容量は年々増加傾向にありますが、作業にかかる時間は容量増加ほど倍々にはなっていません。ただ2TBのHDDで全容量復旧作業するような場合はそれなりに時間がかかります。

**Q** 大きい容量のHDDの場合にはデータも連続して入っている場合も多いんじゃないですか？

**A** 多いです。また障害を抱えているHDDをフルスピードで読むと破壊を助長してしまう可能性もあります。こちらに送られてくるのは具合の悪いHDDですので最大能力で動かしてしまうと、それで息の根を止めてしまう可能性もあります。あえて昔のパソコンを使ってゆっくりデータを取り出さなければならないようなケースもあるんです。

**Q** 復旧の容易さとプラッタの数は関係ありますか？

**A** ありますね。同じような障害であれば、プラッタの数は少なければ少ないほどいいです。ただクラッシュになると、プラッタが1枚で1ヘッドの場合は全滅もありえます。プラッタが2枚でヘッドが複数あり、そのうち1ヘッドが故障の場合は、飛ばして読むということもできるのでそのほうがデータを取れる場合もあります。

**Q** HDDの中でデータ書き込み時のプラッタはどのような使われ方をしているのですか？ たとえばプラッタが3枚あったとして、シーケンシャルデータを書き込んだ場合、1→2→3とすべてを使うのか、ある程度溜まったら2枚目に行くのですか。

**A** それはどちらの場合もあります。昔は1枚目を使い切ったら2枚目を使うというような機種もありましたが、現在はほぼすべてがヘッドスイッチして1→2→3、1→2→3……という感じで容量を詰めていくという形になっています。ですからあまり大きくないデータだと、ヘッドが1つ死んでいても比較的データを取れますが、やはりケースバイケースです。



## サーバを使った復旧

**Q** そういえば、先ほどイメージをサーバに上げるという行程の説明がありました。

**A** まず物理障害があるHDDのデータを吸い上げて、その展開イメージをサーバに上げるようにします。その後回収作業を行う技術者がサーバ上のイメージデータをマウントして復旧用に利用します。最後にリペアしたデータを別のサーバにファイルとして上げるといった感じです。

HDDのイメージをサーバに上げるメリットは、何度でもリトライできるということです。オリジナルのHDDのデータが1回読めればそれが壊れてもオリジナルの状態を保持できます。イメージ化しないで作業する業者の場合は一度失敗したら終わりだと思います。

**Q** データの吸い上げと修復を別に行うわけですね。

**A** 重度の物理障害だと、一度HDDの電気を切ってしまうと2度と動かないケースが多いのです。HDDの半分まで読んでいる最中に止まってしま



うとか。HDDから直接ファイルを取り出そうとすると、構造情報を読み込んでからセクタを読み込むのでランダムアクセスとなり、HDDに対して負荷が非常に高くなります。ですから負荷が少ないように(ヘッドの動きが激しくないように)読み出すわけです。その時点でまず差があります。そのデータをもとに復旧をするので、最良の結果が出るまで何度でも試行錯誤できます。

通常のリカバリーソフトだと必ずランダムリードが入りますから、それだけで壊れるリスクが大きくなります。イメージ読み出しにすれば、危なそうなエリアとかを避けて読んだりすることもできるので、先頭から読み始めてリードエラーが集中しているような部分を避けて別のところから読ませるとかそういうやり方ができます。そういった地道な作業を行い極力イメージがとれるように努力します。



## 火災時のHDD

**Q** HPに火事とかで焼けてしまったHDDの復旧例がありますが、実際にはどんな状況になっているんですか？

### ▼写真5 焼け焦げたノートパソコン



**Y** 磁性体は高温になると磁気が抜けてしまうので、脱磁している状態になってしまうんです。そうになってしまうとデータとしては存在しないのでどうすることもできません。比較的復旧できるのはノートパソコンの場合が多いです(写真5)。内部が密集しているため、外から見ても消し炭のようになっている場合には、パールとかでまわりを壊してHDDを取り外すんですが(写真6)、型式の文字がかすかに読めるくらいまで燃えていても何件か助かったケースはあります。ただ100%読み取れたケースは少ないです。

**Q** そんな状態になっても読めるものなんですか？

**Y** そういう状態だとHDDを普通に開けられるレベルではありません。基板も焼けてしまっているとICなどもボロボロとれてしまいます。コネクタの電線の被覆が溶けて中のワイヤは見えていて形式がかすかにしか読めなくても、経験的にだいたいどんな機種のものか見ればわかるので、アタリを付けて基板を外して、ストックしてある同種のHDDの基板に載せ替えてなんとか動かしたりします。火災の場合、熱と水が関係するので条件としては一番悪いんです。

**Q** 水も関係あるんですか？

**Y** 火事の場合、燃えさかって温度が上がって

### ▼写真6 取り出されたHDD



るところに消火のために水をかけるので、冷えたときに水がピューって入るんですよ。また、熱せられたところに水がかかるので酸化が早いんです。だから火事の後すぐに持ってこられたものでも「何ヵ月外に置いといたらこうなるの?」という感じで、すぐにボロボロになってしまいうんです。

写真5がHPに載せた建設現場の火事のときに送られてきたノートパソコンですが、この中からHDDを抜き出して復旧したんですが、全部で14台くらい来て、写真のものはまだ形が残っていますが、まったく形になっていないようなものまで来て4台だけ復旧できました。100%は無理でしたがある程度は復旧できました。

**S** そこまでして必要なデータだったんですね。

**Y** だと思います。データ復旧の場合には我々はお中を見ません。見てしまうといろんな不都合が生じてきますから見ないのが大原則です。必要性というのはお客さんしだいですからね。

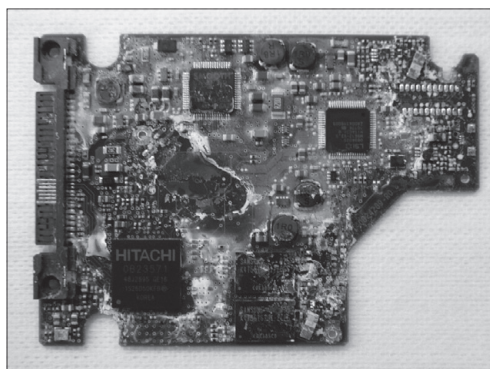


## 津波で被災したHDD

**Y** あとは海水の場合ですね。去年の東日本大震災のときに被災したHDDが何百台も来たんですよ(写真7)。

**S** これは塩ですか?

▼写真7 津波による塩水に浸かった基板



**Y** そうです。これはHDDのコントロール基板部分です。それを薬剤処理してこういった状態(写真8)にまで回復しました。元のままの状態ではとても電気を入れられるような状態ではないので、薬品で洗浄してここまで回復しました。HDD復旧のためには、基板なども極力オリジナルの環境にできたほうがいいのです。

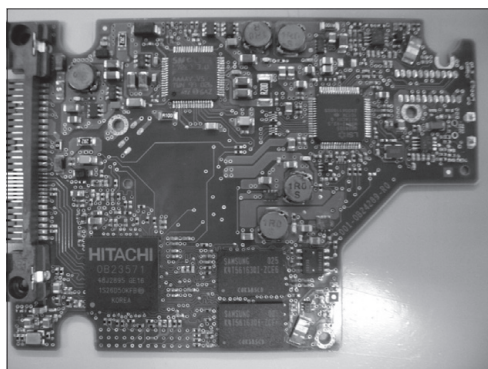
**S** これで使えるようになったんですか?

**Y** はい。これはSASの基板ですがハンダに鉛を使ってるんですよ。SATAは環境問題(ROHS規制)のために鉛フリーを使っているものが多いのですが、塩水に浸かるとほとんどが短時間で取れてしまいます。しかしSASは有鉛ハンダを使用しているものが多いため、部品が残っているケースが多いです。メーカーもまさかHDDが海水に浸かるということは想定外だと思います。

**S** まわりがこんな状況だと、中もひどい状況ですか?

**Y** 津波はパワーがすごいので、全部HDDの中に水が入ってますね。開けたら満水のときとかもありました。(火災や津波などで)事故に遭ったHDDは、すごいにおいのするものがあるんですよ。そういった作業には、顔を覆ってタオルとかで何重にも鼻を覆って目だけ出して作業するわけです。焼けたにおいと、肉が腐ったよ

▼写真8 薬品処理して使えるようになった基板





うなにおいとか、いろいろあります。処理中に薬品を使って化学反応を利用し清掃する場合もあるのでガスマスク、防護服を着用し、チャンバー内で作業することもあります。薬品についても、同じような事例がないわけで、試行錯誤しているんな薬品を試してみたりもします。サンプリングした中で一番結果がよかったものを使うわけです。

**S** 絵画の修復並ですね。

**Y** 復旧にはいろいろ広範囲な知識が必要です。特殊な薬品を使うようなこともあり、水没や火災品などは開かないのでボール盤なども使います。特殊工具や治具の作成のため、旋盤やフライス盤などの工作機器も所有しています。幅広いいろんなことを知らないと、「こういうものが来たら、こうすればいい」というのは思いつかないと思います。同じ壊れ方するなんてほとんどないので。



## SSDの場合

**S** SSDに関しても対応されているということですが、SSDはどんな故障があって、どんなふうには持ち込まれるのですか。普通のHDDと同じではないですよ？

**Y** プラッタなどの回転する部分がないので、物理的な故障というと素子が焦げたりユーザが誤って踏んでヒビが入ったとかです。ただ認識しないとか、リードエラーといった軽度なものもあります。ですが、HDDと違ってできることが限られるので難しいです。たとえば読むということであれば、素子を取り出して読めばいいのですが、単独で読めたとしても、それを組み立てられない。どうしても難しいところがあります。

最近チップの中でもRAIDを組んでいるものがあったり、暗号化されているものがあって復旧がたいへんです。以前から弊社ではUSBメ

モリなどの復旧などもしてきていますが、SSDはそういったものと比較して複雑かつ容量が大きいので復旧も難しいです。



## SDカードやUSBメモリ

**S** SDカードやUSBメモリとかの修復依頼も多いですか？

**Y** SDカードとか折れてしまったものなどは、パターンが切れてしまった程度であれば復旧できる可能性はあります。携帯からSDカードを抜く際にペンチで抜いて割ってしまったケースがあったのですが、その場合はパターンの部分の切断だったので、顕微鏡で見ながらパターンを削り出し髪の毛より細い線で回路を修復した結果、読めたというケースもあります。今日も犬に咬まれたっていうSDカードが来てました。

**S** やはり犬が多いんですか？

**Y** けっこう多いですね(笑)。

旅行先でデジタルカメラを落としたとか、車に踏まれたといったものもありました。シリコンメディア系は水には強いので、よく洗ってちゃんと乾かせば読めるようになることが多いです。USBメモリとかの場合は積層基板を使っていたりしていますから、自分で無理にハンダ付けなどを行って壊すより業者に任せるのが良いです。



## 記録フォーマット

**S** 最近のNAS(Network Attached Storage)とかですとWindowsフォーマットではないものが使われていたりしますが、そういったものは持ち込まれる場合はありますか？

**Y** しょっちゅうあります。昔のNASは数百万クラスが当たり前でそれなりの設計もされていますしそれなりのコストもかかっているののでい



いものだったんですよ。今は低価格帯のNASで10万円も出せば買えてしまう。NASが4TBとかいうと、中に5~6千円のHDDが4台入ってたりしていますので、価格相応の性能ということになるのでしょう。開けると電源回路からして、値段相応の作りになっています。

**Q** ファイルフォーマットによる復旧しやすさはあるのですか？

**A** NASでいうと、最近LinuxでXFSでフォーマットされているものが多いです。次にEXT3とかですけど、NASとか関係なしに言えば、NTFSが一番楽です。ファイル構造的にも削除されたファイルの復旧はやりやすいです。XFSやEXTは削除されたものの復旧は難しいです。消された時点で、どこに入っているかというポインタが消えてしまうので。

Btrfsは新しいフォーマットなので、復旧ツールがそろっていませんが、それ以外のフォーマットは全部ありますので、削除したとか上書きしたとかフォーマットしたとかそういうものの復旧はできます。特定のトラックやセクターがクラッシュした場合、そこがポインタの場合はセクタが連続しているものはなんとかなるかもしれませんが、セクタが飛んでいるものは難しいです。

**Q** 最近の大容量のHDDの場合は、1つのクラスタサイズが大きくなってありますが、それによる復旧への影響はありますか？

**A** 1つのクラスタにファイルが収まっているものについては、助かりやすいということはありません。

**Q** 一番持ち込まれるのはFAT32、NTFSのものでしょうか？

**A** NTFSが一番多く、ちょっと古いシステムだとFAT、NASだとXFS、それ以外にMac

(HFS)もあります。



## 論理障害

**Q** 論理障害についてはいかがですか？ 市販ツールと業者の技術力についてはいかがですか？

**A** 簡単な論理障害の場合は、たとえば区画を削除してしまったとかファイルを削除してしまったような場合は市販ツールを使ってもほとんど変わらないですよ。業者の中にも市販ツールを使っているところがあります。

お客さんから依頼を受けた場合、削除したファイルを復活したいとか簡単と思われるものは、弊社で販売しているソフトウェアを案内しています。それだと1万~3万4千円くらいなので。それでどうしてもダメだったらご連絡くださいというような対応をしています。



## 保存メディアの信頼性

**Q** 話は復旧とは違いますが、個人が大事なデータを保存しておくことを想定した場合、どんなメディアに保存するのが一番安全だと思われますか？

**A** 意外とMOなんです。あれは大丈夫なんです。光で非接触じゃないですか。リードエラーが起きても何度でも読めるんですよ。HDDのようにクラッシュが起きることもなく好きなだけトライできるので。

**Q** BDやDVDとかはいかがですか？

**A** まず信頼できる日本製かどうか(ただ日本製は今現在見かけなくなってきましたが)。1ヵ月で読み込み不良を起こすような台湾製のものがあるという噂も聞きますし、直射日光を記録面で当ててしまうとすぐに劣化してしまうということもあるのでお勧めしません。触ってしまったり傷付けるとか、折り曲げにも弱いんです。それ



をちゃんと管理できるならばDVDも有効と思いますが、一番手間がかからずに生存率が高いのはMOだと思います。BDはまだやめておいたほうが賢明ですね。

HDDも外付けの縦置きタイプっていうのは意外と生存率が低いんですよ。人がひっかけて倒す率が高いからです。それだけでHDDが壊れますから。倒し方によっては中のデータは全損です。平置き型の方が転倒に対する安全性は高いです。できればパソコンの中でRAID1とか5を構築し、定期的に本当に重要なものだけ外付けのポータブルHDDにとっておくとか、DVDに焼くとか。お勧めはテープですが、個人じゃ買えませんからね。

HDDも回転部品が多いので、使わないときでもたまには通電してやらなければいけないですね。



## 復旧に関する価格

**S** 料金の話ですが、たとえば最近ではHDDの容量もどんどん上がってきています。そうなると金額が復旧容量に関係すると、かなりの高額になってしまいうんじゃないかと思うのですが？

**Y** 今のところ、シングルのHDDの場合はHDDの容量で値段を決めています。TB級のHDDの場合は、そのまま容量で金額が決められないので、そこは抑えるようにはしています。お客さんにとっても納得できる金額はあると思いますので。

**S** コピーするだけでも時間がかかりますからね。

**Y** 実際に復旧にかかる時間も容量増加で長くなるようになってしまいますし、今はファイル構造を把握して、実際にファイルのある部分だけを復旧するようにしますが、それができないケースもありますので。



## 復旧率と良い業者とは

**S** 復旧業者に出すというのは、かなり敷居が高い(お値段も)という気がするのですが、御社が考える良い復旧業者の条件とは何ですか？

**Y** よくお客さんに「御社の復旧率は何のくらいですか？」と聞かれるのですが、その復旧率が件数で言うのか、トータルのデータ量について何%復旧できるということなのかそれによっても違うし、日々違います。復旧率何%とうたっている業者は気を付けたほうがいいと思います。たとえばすべての案件について、1セクターでも復旧できたら復旧できたということで数字を出すのはナンセンスですから。

まずは依頼する前にどんなことをしてくれるのかを電話して聞いたほうがいいです。調査とか初期診断とかいう場合がありますが、そこで何をやって何がわかるのか、どういうレポートを出してくれるのかということを納得いくまで聞くことが重要だと思います。

物を送ってしまうと、人質をとられたのと同じことですから、あとから見積金額が上がったり、それを拒否すると返してもらえなくなるとかのトラブルも聞きますからね。

お客さんからの質問ではかの業者で復旧依頼したらできないと言われたが、御社ではどうですかというのがあります。その際、その業者さんの報告書には何と書かれていますかと聞くと、電話で「できません」と言われたとの答えがあります。少なくとも復旧不能な場合でも報告を書面でもらえる業者を選びましょう。

**S** 今回はいろいろ貴重なお話をありがとうございました。**SD**



# 第7章

## 安定運用を目指すための Flashストレージ 導入ポイント

長谷川 猛 (HASEGAWA Takeshi) フュージョンアイオー (株) TwitterID: @hasegaw



### はじめに

皆さんが使っているスマートフォンにタブレット端末、音楽プレーヤーはもちろん、手許のクライアントPCなども2次記憶としてFlashメモリを採用したものが主流になってきています。サーバに接続されるノードがFlashメモリに染まっていく中、サーバ内のHDDもFlashメモリで代替されるのが「当たり前」となる日も近いでしょう。ある意味コンピュータアーキテクチャが大きく変化しつつある局面にあるとも言えます。

しかし、Flashメモリベースの製品の代表例であるSSDがHDDの延長線として語られている状況もあり、従来のHDDとFlashストレージは異なった特性を持っていることについてはなかなか理解されていないように感じます。本稿を通じて、HDDとFlashストレージを再度区別し、Flashメモリの特性を引き出すためのアイデアを思いついていただければと思います<sup>注1</sup>。

本稿ではFlashメモリを用いたストレージ全般について考えます。特筆しないかぎり、多くのFlashストレージ製品全般に共通しますので、これらの傾向について理解を深めていただければと思います。



### HDDとFlash ストレージの違い

HDDでは、プラッタと呼ばれる円盤に、磁気でデータを記録します。プラッタの上に、データを読み書きするためのヘッドをあてて、ディスクの回転およびアクチュエータによるヘッド移動を組み合わせて目的のデータにアクセスするという特徴を持っています。これに対し、現在コンピュータでよく使われるFlashストレージでは、記憶メディアとしてNAND Flashメモリを使用します。このように、HDDとFlashストレージでは、データ記録のメカニズムがまったく違います。

HDDは記録メディアの回転とヘッド移動を伴うため、アクセスには数ミリ秒の時間を要し、また時間あたりのI/O回数(IOPS)が伸びづらいという制限があります。NAND Flashはこのような物理動作がないため故障率が低い、また高速と言われますが、部品レベルでは読み込みは速いものの、書き込みや書き換えには時間を要しています。このギャップをコントローラでいかに隠蔽するかが、Flashストレージにおける技術的なチャレンジとなっています。

### Flashメモリの基本操作

NAND Flashメモリは、セルと呼ばれる記憶素子に電荷を閉じ込めて情報を保存するしくみになっており、プログラム(書き込み)、リード(読み込み)、イレース(消去)の3つの操作が可

注1) 筆者はエンタープライズ向けのネイティブFlashメモリ製品を提供するFusion-io (<http://www.fusionio.jp/>) に所属しており、今回の内容の検討ではかなり悩んだのですが、本稿を通して確信をもってioDriveシリーズを選定いただけるようになればと思っています:-)



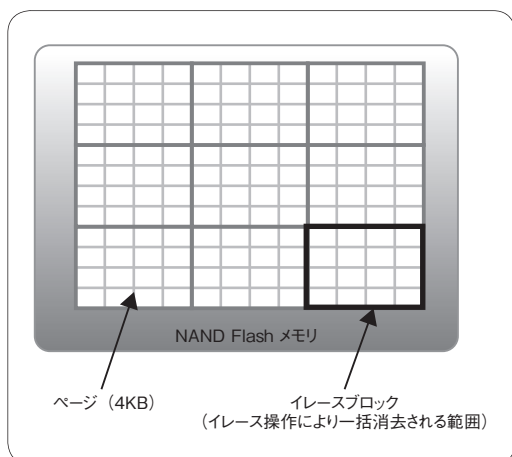
能になっています。NAND Flash メモリは、一度プログラムしてしまうと、イレースしなければ再度プログラムできません。

NAND Flash メモリのプログラム単位は基本的にページ単位、たとえば4KB単位となっています。このため、Flash メモリからデータをリードしたりプログラムしたりする場合には必ずこの倍数のデータサイズになります。しかし、現在、多くのOSはHDDのセクタサイズが512Bである前提で設計されています。このため、Flash メモリでHDDを代替するには、4KB単位の記

憶メディアで512B単位のアクセスができるようにエミュレーションする必要があります(図1)。

HDDとは決定的に異なるのが、ブロックイレースの存在です。Flash メモリはイレースを操作をまとまった単位で行う必要があります。たとえばOSが512Bのセクタだけを更新しようとしても、ブロックイレースの単位が256KBであれば、この単位で一括消去されてしまいます<sup>注2</sup>。このように、HDDと同等のランダムアクセスは、Flash メモリ自体では実現できません。HDDの代替として使いたければ、やはり何らかの形でエミュレーションする必要があります<sup>注3</sup>。

▼図1 ページとイレースブロックの関係



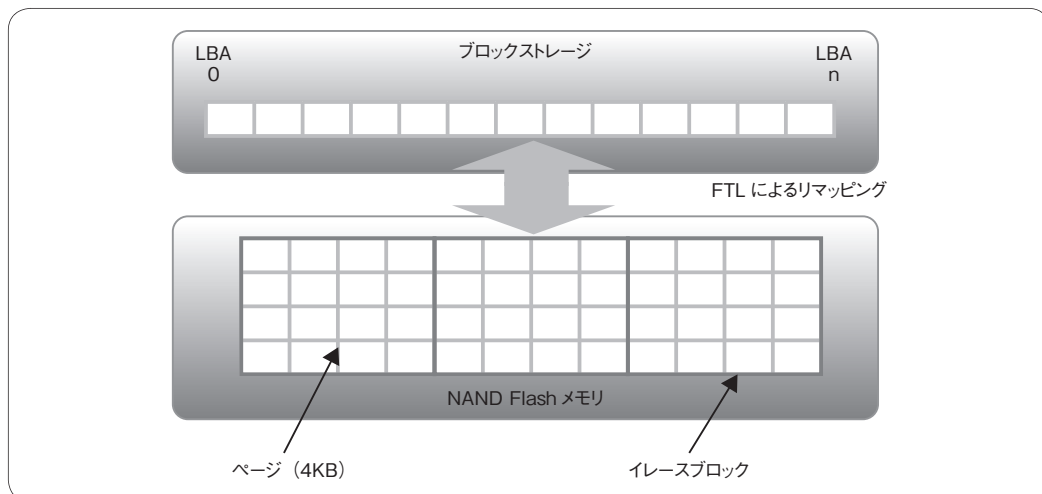
## Flash Translation Layer

HDDとは異なる読み書きのプリミティブを持ったFlash メモリを、HDD同様にブロックデバイスとして利用可能にするのが、FTL(Flash Translation Layer)と呼ばれるしくみです(図2)。FTLは、一般的に、Flash ストレージの内

注2) ページ単位のリード/プログラム、またブロック単位のイレースという条件を付けることにより、NAND Flash メモリの構造を単純化できた結果、容量やバイト単価に優位性あるメモリとなっています。

注3) Flash メモリの操作に興味がある方は、Flash メモリ LSI の型番をキーワードに、Googleなどでデータシートなどを探してみてください。

▼図2 Flash Translation Layer



部にログ構造のファイルシステムを持っています。これはいわゆるExt4、XFSやNTFSといったものとは違い、読み書きの単位や書き換え回数に制約があるメディアであるFlashメモリをHDDと等価に扱うためのしくみを提供します。たとえば、メディアが4Kバイト単位のプログラム／リード、また256Kバイト単位のブロックイレースとなる前提で、512バイトセクタの記憶域をエミュレーションします。

### ▶ハウスキーピング

Flashメモリのセルは、特定のセルに偏ってプログラムを繰り返すと、その領域がウェアアウト(Wear-out)してしまい、記憶デバイスとしての機能を果たさなくなるため、プログラムできる回数に限界があります。この問題を解決するために、プログラム頻度を均一化するための配置換えが適宜行われます。この処理をウェアレベリング(Wear-leveling)と呼びます。

また、FTLの多くはログ構造のファイルシステムになっていることがほとんどです。ログ構造ファイルシステムはデータの更新を繰り返すと、やがて有効なデータと不要になったデータ

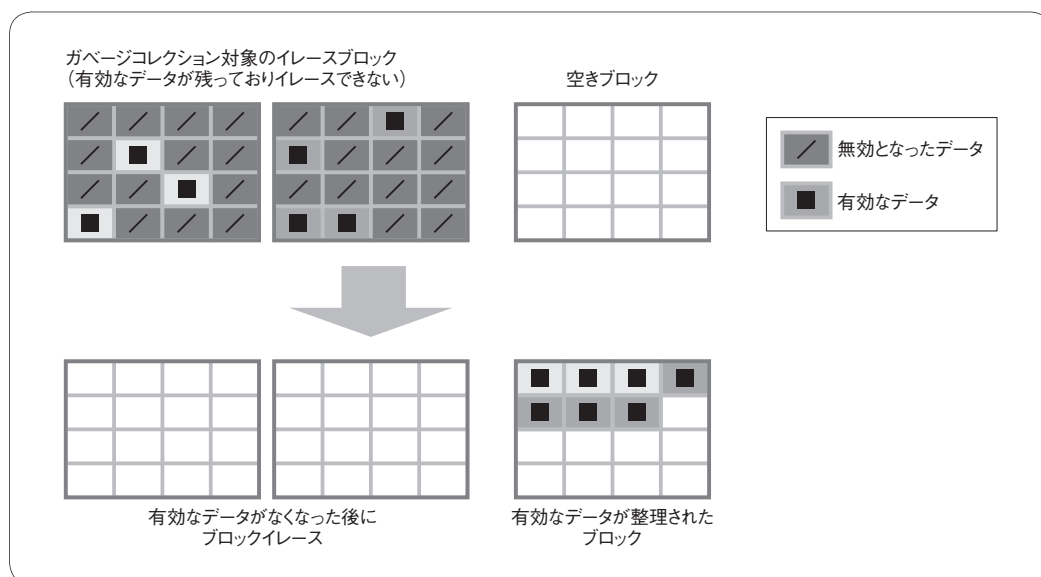
がブロック内に入り混じるようになります。このために、必要なデータをほかの領域にプログラムしなおして、空いた領域をブロックイレースする、ガベージコレクションと呼ばれる処理がFTLの内部で行われています(図3)。

このように、FTLの実装はフラッシュストレージを開発する各社の個性が出る部分であり、性能だけでなく、デバイス自体の信頼性をも決める重要な心臓部です。フラッシュ製品のFTL実装は製品次第であるため、信頼がおける製品を選定することが大事です。

### ▶レイテンシーとIOPS

HDDの特性では、I/Oあたりの所要時間がコンピュータの性能から見て明らかにボトルネックであったため「アプリケーションが何回のI/O操作が発行できるか」、すなわちIOPSを1つの性能指標としてきました。HDDの場合はメカ部分の制約により待ち時間の影響が大きかったのですが、Flashストレージの場合は優秀なもののほどデータ転送時間が占めることになり、傾向が異なっています(図4)。

### ▼図3 ガベージコレクション





## ● Flashストレージの性能に裏切られないためのポイント

Flashストレージの選定の際によく見落としがちなポイントをいくつか紹介します。

### ▶ ストライピングとレイテンシー

Flashメモリは、単体では決して速くありません。Flashストレージは、内部的に、複数のチャンネルに接続されたFlashメモリを大量にストライピングすることにより帯域幅を確保しています。

システムレベルで複数のディスクをストライピングすると、帯域幅が増すのはご存じかと思いますが、レイテンシー性能も悪化しにくくなりますが、しかし、とくにI/O単位が細かい場合、単体のデバイスよりもレイテンシーが短くなることはほとんどありません。多くの場合、ストライプ幅は128～256KBなどの単位であり、ストライプしてもシーケンシャルかつ細かい書き込み負荷、たとえばOLTP(OnLine Transaction Processing)のログ書き込みのような負荷にはス

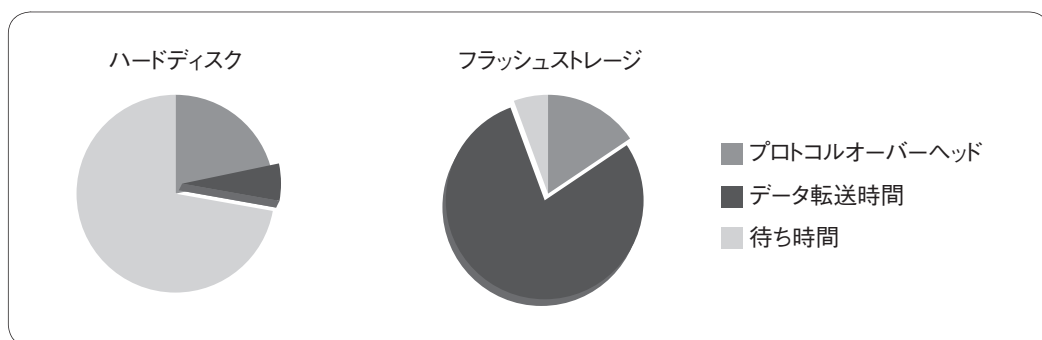
トライプの効果が得られにくいのです。このようなワークロードの場合は、内部チャンネル数が少ないデバイスを複数ストライプしても、性能が得られないどころか障害のリスクも高まるため、低レイテンシーなエンタープライズ向け製品の選定が賢明でしょう<sup>注4</sup>。

### ▶ QD(Queue Depth)が低い状態の性能に注目する

FlashストレージのIOPSスペックは、QD=32など極端にI/Oが飽和した状態で測定した結果であることがほとんどです。しかし、実際のアプリケーションは、このような高い並列度でI/Oしません。たとえばRDBMSであればWALを同期書き込みするため並列度は期待できないため、ベンチマークでQD=1～4程度における実効的なレイテンシーを測定し参考にすると良いでしょう(リスト1)。

注4) ぜひioDriveシリーズを検討してください:-)

## ▼図4 HDDとFlashストレージにおけるレイテンシー原因の違い(イメージ)



## ▼リスト1 fio(flexible I/O tester)のコマンドライン例

```
# fio --filename=/dev/○○ --direct=1 --rw=write --bs=512 --size=5G --numjobs=4 --runtime=10  
--group_reporting --name=foo  
# fio --filename=/dev/○○ --direct=1 --rw=write --bs=512 --size=5G --numjobs=1 --runtime=10  
--group_reporting --name=foo
```





## 実際の導入デザインに 向けて

### ● すべてのホットデータをFlash上へ

Flashストレージの導入によって普段の応答速度が高速になっても、時々HDDアクセスが残ってしまうと一気に性能が低下してしまいます。Flashストレージを導入する場合には、原則としてすべてのホットデータをFlashに載せたほうがよいでしょう。

### ● Flashストレージの冗長化?

Flashストレージを冗長化する場合のRAIDレベルは何がよいのか、よく話題になるかと思えます。RAIDレベルで有名なものはRAID-1(ミラーリング)とRAID-5(分散パリティ)もしくはRAID-6などがHDDではよく使われます。RAID-1はパリティ計算が不要なため性能が出やすいのですが、実効容量が半分に目減りしますのでバイト単価が高くなります。RAID-5/6はパリティ計算が発生しオーバーヘッドとなるほか、またHDDを前提としたパリティデータの更新書き込みが多発するため、Flashストレージの寿命を大幅に縮めることになります。

HDDの場合は単体でほとんど耐障害性がないため、ミラーリングやパリティ分散による冗長化(RAID)によってシステムの可用性を担保する手法が一般的です。これに対してFlashストレージの場合は、SSDのように、単純にHDDの代替として使うことを想定したもののほか、エンタープライズ向け製品の場合ではデバイスの内部で分散パリティに相当する保護が実装されているものまでさまざまです<sup>注5)</sup>。信頼性や可用性が求められるようなシステムでは、デバイスレベルで冗長性が確保されているタイプのエンタープライズ向けのFlashストレージ製品を選定するほうが良いでしょう。

注5) たとえばioDriveシリーズの場合は、内部のFlashメモリの故障に対する自己治癒能力をもっています。

### ● バックアップを怠らない

FlashストレージはHDDと違って稼働部分がないため信頼性が高いと言われることがありますが、形あるデバイスであり、故障の可能性がある点は同じです。また書き込み許容量による制約を伴うFlashストレージの場合、RAIDで冗長化したとしても、複数のFlashストレージへの書き込み量はほぼ均等になるため、HDDと比べると各デバイスに障害が発生する時期は重なるリスクが高くなります。データのバックアップを怠らないようにしましょう。

十分な性能を持ったFlashストレージであれば、システムが稼働中であつたとしてもオンラインバックアップを並行動作させることが可能になります。Flashストレージは性能の向上だけでなく、システムの運用改善に向けての一手を打つための手段となるでしょう。

### ● レプリケーション

サーバにFlashストレージを搭載した場合によく心配されるのが、サーバ自体が単一障害点(SPOF: Single Point of Failure)となってしまう、という指摘です。FlashストレージはHDDと比較し、故障が少ないうえ、内部で冗長性を持った製品を選定していれば可能性はかなり低いはずですが、とはいえ、性能が高いがゆえに、障害が起きた際の影響範囲が大きく、無視できないのも事実でしょう。

そこでお勧めしたいのがレプリケーションによる、サーバをまたいだ冗長化です。ミドルウェアによってはレプリケーションを標準でサポートしていますし、またレプリケーション非対応のミドルウェアであってもブロックレベルのレプリケーションソフトウェアを組み合わせることで実現できます。また、共有ディスクを使わないためSPOFが生じづらい、コストが下げられるといった利点もあります。

レプリケーションには、おもに同期レプリケーションと準同期レプリケーションの2種類があ



ります。前者は、マスタ側に書き込まれたデータがスレーブ側にも書き込まれることが保証されてから処理が完了したとみなします。すべてのデータがスレーブに必ずレプリケーションされているため金銭のやりとりなど確実性が必要になる場合に使われますが、同期処理となるため性能が伸びにくくなります。

非同期レプリケーションの場合は、マスタが自分の中でデータの書き込みを完了した時点で処理が完了したとみなし、バックグラウンドでスレーブにデータを送信します。マスタとスレーブを同期させないため高速ですが、データの送信が完了しないうちにマスタとスレーブが正常に通信できなくなると、すべての書き込みがスレーブに伝わらず、直近のデータがいくらか失われる可能性を伴います。ある程度のデータ喪失が許容できる用途や長距離の場合に非同期レプリケーションが使われます。

レプリケーションの構成では、サーバ間を接続するインターコネクトの性能に注意します。とくに同期レプリケーションを使いたい場合には、インターコネクトのレイテンシーが重要な

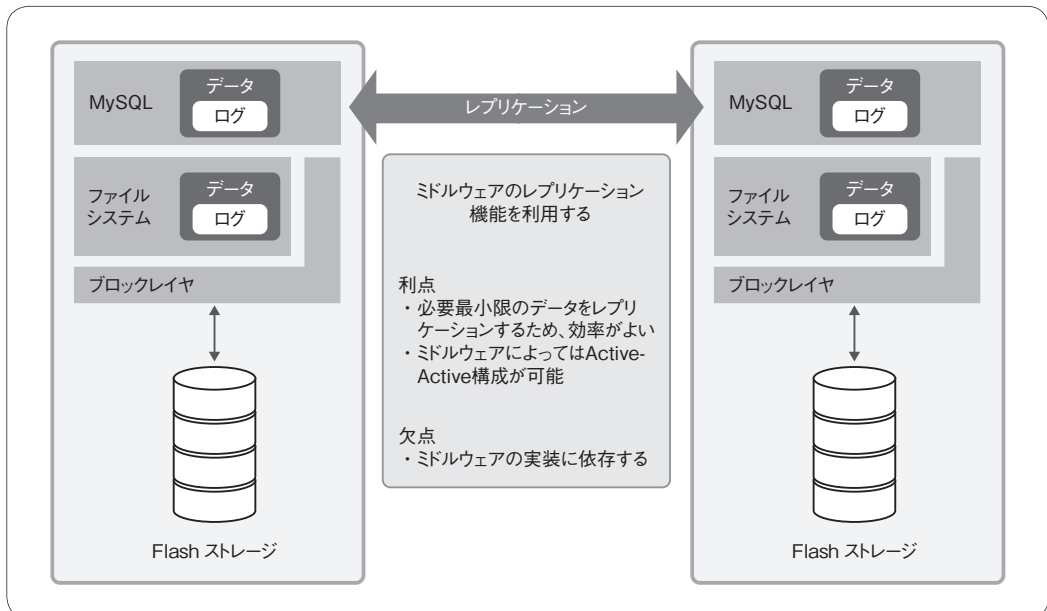
要素となります。可能であれば、低レイテンシーな10Gb EthernetもしくはInfiniBandなどを検討すると良いでしょう。

### ▶ミドルウェアレベルのレプリケーション

本誌をお読みの方であればMySQLなどのユーザは多いかと思いますが、MySQLのレプリケーションを使って、Flashストレージを搭載したサーバ間にてレプリケーションを採る方法は王道の構成ともいえます。また、最近のMySQLでは準同期レプリケーションという選択肢が増えています。準同期レプリケーションとは、MySQLのログファイルがスレーブに転送されてから完了とする方法です。このため過去のトランザクションが失われることはありません。

ミドルウェアによるレプリケーションは、ミドルウェアが必要とするデータのみ転送される点がメリットです。フェイルオーバーなどの操作についてはミドルウェアごとに流儀がありますので、その状況ごとに構成を設計する必要があります(図5)。

▼図5 ミドルウェアレベルレプリケーション



### ▶ブロックレベルのレプリケーション

ミドルウェアがレプリケーション機能を持っていない場合、もしくはミドルウェアのレプリケーション機能を使わなくてもレプリケーションの構成がとれる方法として、ブロックレベルのレプリケーションが考えられます(図6)。低レイテンシーなFlashストレージの一般化により、ブロックレベルの同期レプリケーションが現実的な選択肢となってきています。

Linuxの場合はDRBDと呼ばれるオープンソースソフトウェア、もしくは商用ソフトウェアであればLifeKeeper<sup>注6</sup>やCLUSTER PRO<sup>注7</sup>などがあります。

ブロックレベルでレプリケーションする場合、スレーブ側ではファイルシステムをマウントできないため、待機系として、稼働系に問題が生じた場合にのみミドルウェアが起動しサービスを引き継ぐ設計となります。その代わり、これまでのアプリケーションやミドルウェアをその

まま冗長化できる点、ファイルシステムのマウント、サービスIPの付け替え、スプリットブレイン対策がHAソフトウェア側にて担保される点がメリットと言えるでしょう。



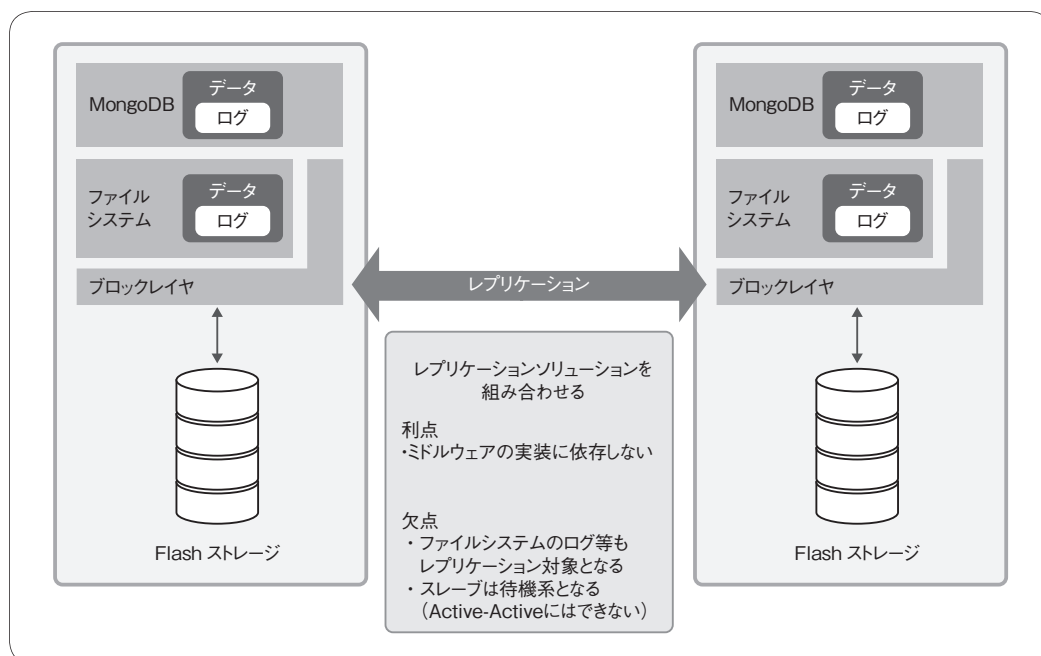
## おわりに

HDDとFlashストレージは動作原理だけでなく、特性までもが異なり、これまでのノウハウが適用できない場合もあるでしょう。Flashストレージを使いこなすには、システム自体を新しいデザインに見直すことさえあるでしょう。Flashストレージは間違いなく昨今のトレンドとなっており、当面の間、コンピュータの性能を最大限に引き出すためのカギとなることは間違いありません。「なんとなく速いし大丈夫」ではなく、自信をもってFlashストレージを使いこなしてください。SD

注6) <http://www.sios.com/products/bcp/lkdk/product/lifekeeper.html>

注7) <http://www.nec.co.jp/pfsoft/clusterpro/clp/index.html>

### ▼図6 ブロックレベルレプリケーション





## 仮想環境でも単一のコンソールで管理できる バックアップソフト

Software Design編集部

### ▶ NetVault Backup

日本クエスト・ソフトウェアの「NetVault Backup (NVBU)」は、企業のシステムの拡張に合わせてサーバ1台のバックアップからエンタープライズレベルのバックアップまで、必要なときに必要なだけ拡張できるソフトウェアです。モジュラーアーキテクチャを採用した設計により新しい製品や機能にもプラグインで対応できるため、余計なライセンスコストを省き容易にシステム変更が可能なことが特徴です。

NVBUのオプションである「BMR(Bare Metal Recovery)」は、万一の壊滅的なデータ障害の際にも、システム全体をDR(Disaster Recovery)イメージから素早く復旧することができるシステムリカバリ・オプションです。大きく分けてネットワーク型とローカル型の2種類が用意されています。

### ▶ BMRの威力とは

ネットワーク型BMR(VaultDR)では、NVBU ServerがVaultDR Serverとなり、ネットワーク経由でDR対象システムのシステムイメージのバックアップを取得します。これを実現しているのが、システムリカバリ専用のOS「VaultOS」です。これは、システムリカバリ対象となるマシンをブートしてバックアップ／リストアを行います。マシンごとにブートディスクを作成する必要がないこと、通常のバックアップと同じテープやディスクに、ディザスタリカバリバックアップも同居して保存しておけることが特徴です(図1)。

NVBUは動作が非常に軽快で、他社製品から乗り換えたというケースもあります。操作方法や運用方法は、わかりやすいGUIによって初めての管理者でも容易に把握することができ、乗り換えの際でも安心なことも特徴となっています。

NVBUのBMRは、x86系のマシンならFreeBSD、Linux、Solaris(Intel)、WindowsなどOSに関係なくディザスタバックアップ／リカバリの対象とすることが可能です。また、WindowsかLinuxであれば、システム・イメージのバックアップをオンラインで行えるオプションが用意されています。

### ▶ VMwareのバックアップとは

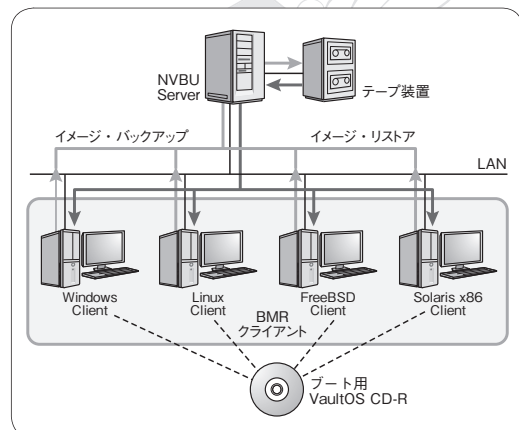
BMRはVMwareの仮想マシンにも対応していま

すが、「VMware Plug-in」というNVBUの別のオプションを使ってイメージレベルのバックアップをすることで、システムリカバリに備えることも可能です。本オプションはVADP(vStorage APIs for Data Protection)を使ったバックアップに対応しており、ユーザが複雑なスクリプトを作成することなく、NetVault BackupのGUIからVMware ESX環境のデータ保護を簡単にできます。

仮想マシンのイメージレベルのバックアップから、イメージレベルのリストアだけでなく、ファイルレベルのリストアもできるので、必要なファイルを迅速にリストアすることが可能です。これでダウンタイムを最小限に抑えられることも大きなメリットです。このほか、VSSによる静止スナップショットのサポートや、vMotionを使用して別のVMware ESX上に仮想マシンを移動しても、その仮想マシンを追跡することができ、NetVault側のオペレーションを変更することなく移動後のVMware ESXからデータをバックアップできること、vSphere HA対応など、現場の管理を容易にする機能を搭載しています。

たとえば、一度設定しておけば、その後システムに追加した仮想マシンもすべて、NVBUの管理下に入ります。システム上にある仮想マシンをすべて、自動でバックアップすることが可能になるのです。SD

▼図1 ネットワーク型BMR(VaultDR)のバックアップ／リカバリイメージ



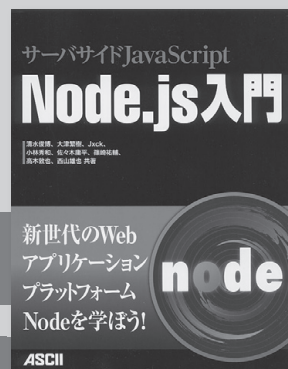
BOOK  
no.1サーバサイド JavaScript  
Node.js 入門

清水 俊博、大津 繁樹、Jxck、小林 秀和、佐々木 庸平、篠崎 祐輔、高木 敦也、西山 雄也【著】  
B5変形判、584ページ／価格＝3,990円（税込）／発行＝アスキー・メディアワークス  
ISBN＝978-4-04-870367-3

本書は現時点で国内唯一のサーバサイドJavaScript「Node.js」に特化した解説書。Node.jsの習得には、非同期プログラミングやイベントループといった概念を理解することが必須だ。これらについてはWeb上にも多くの解説はあるが、抽象的でわかりにくい。ここが初心者には最初の壁だ。本書では豊富なサンプルコードや図と

ともに説明されている。まずは動かして試してみる、それを繰り返すうちに感覚的に前述の概念が理解できる、そんな工夫がなされている。

後半は具体的なWebアプリの作り方、Node.jsの内部構造、非同期プログラミングの改善方法、など実践的な内容が盛り込まれている。入門者から上級者までを満足させる1冊だ。

BOOK  
no.2

## C++のためのAPIデザイン

マーティン・レディ【著】／ホジソンますみ【訳】／三宅 陽一郎【監修】  
B5変形判、528ページ／価格＝3,990円（税込）／発行＝ソフトバンク クリエイティブ  
ISBN＝978-4-7973-6915-1

本書は米ピクサーに在籍していたマーティン・レディ博士が、映画制作でも使われるソフトウェアの開発で培ったAPIデザインのノウハウをまとめたもの。大多数のプログラマはAPIを使う立場だが、本書はAPIを作る側から見たノウハウ本。APIを作るには、ユーザがどのような場面で利用するのか、どの程度の汎用性を

持たせるか、インターフェースはどうするか、など数多くの知識を総動員する。APIを作るにあたって念頭に置く事柄から始まり、「パターン」「デザイン」「スタイル」に関しての設計、パフォーマンスやテストなどの方法を解説する。C++以外の言語でも利用可能な考え方が詰まっているので、他言語ユーザにもお勧めだ。

BOOK  
no.3

## ソーシャルもうええねん

村上 福之【著】  
四六判、192ページ／価格＝980円（税込）／ナナ・コーポレート・コミュニケーション  
ISBN＝978-4-904899-33-5

今、流行のSNSや携帯ゲームがどうやって利益を得ているか、どんな人が利用しているかを、冷静かつ皮肉たっぷりに分析し、そのどちらの裏側を垣間見せてくれる1冊。SNSがだんだんと金と欲望だらけの世界に見えてくる。辛口の批評が多いが、同感できることばかり。私たちはもう少しSNSと冷静に付き合う

必要があるのかもしれない、と思わせられる。だが、本書の魅力はそんな高尚なものではないだろう。単純に著者の歯に衣着せぬ意見は、読者が感じていたSNSに対する胡散臭さを代弁してくれていて気持ちがよい。みんなが熱をあげているところを1人冷静に突っ込みを入れて、そんな構図が読んでいておかしい。

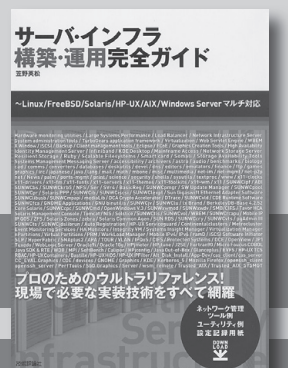
BOOK  
no.4サーバ・インフラ構築・運用完全ガイド  
～Linux/FreeBSD/Solaris/HP-UX/AIX/Windows Server マルチ対応

笠野 英松【著】  
A5判、488ページ／価格＝3,675円（税込）／発行＝技術評論社  
ISBN＝978-4-7741-5408-4

本書は幅広い分野への精通が求められる最近のサーバ・インフラエンジニアに向けた、マルチプラットフォームを対象とした内容が特徴。

各種サーバの構築、クラスタリングおよび仮想化の手法、セキュリティ設定、BYODの導入、そしてリモート管理やネットワーク監視の運用と盛りだくさんの内容ながら、A5版で厚さ

25mm程度というコンパクトさ。その代わりと言っては何だが、内容は極めてシンプルな「手順書」という形でまとめられている。初学者がこれだけを見て設定するのは難しいだろうが、中・上級のエンジニアにとっては、自身の手順を整理・確認するうえですっきりと見やすいリファレンス的な構成となっている。



# キーパーソンに聞く 2013年に来そうな 技術・ビジョンは これだ!

2012年、IT業界は次々に発表されるガジェットやWebサービスなどのおかげで話題に事欠かないにぎやかな1年でした。

一方で、SIビジネスを中心に暗い話も跡を絶ちません。いったい2013年はどうなるのでしょうか？そこでIT業界を牽引するオピニオンリーダーたちに2013年にITエンジニアが注力すべき技術、能力、行動指針などについて聞きました。新年を迎える前に、本特集を参考にして自分が学習する対象や、目標とする理想の働き方を見つけてください！

## 第1章 FreeBSD、Linux、Windows、OSはどう変わるのか？

後藤 大地 ▶ p.80      やまねひでき ▶ p.81  
あわしろいくや ▶ p.82      横山 哲也 ▶ p.84

## 第2章 ソフトウェア開発はどうなるのか

白石 俊平 ▶ p.85      羽生田 栄一 ▶ p.86  
千葉 滋 ▶ p.87      結城 浩 ▶ p.88  
古橋 貞之 ▶ p.89

イラスト 中野裕子

## 第3章 情報技術の新しい流れ

小島 克俊 ▶ p.90      藤村 能光 ▶ p.91  
鈴木 宏康 ▶ p.92      谷本 真由美 ▶ p.94

## 第4章 ネットワーク技術はどうなるのか

田中 邦裕 ▶ p.96      伊勢 幸一 ▶ p.98  
小宮 崇博 ▶ p.99

## 第5章 Slerは生き残るか？

谷口 有近 ▶ p.101      湯本 堅隆 ▶ p.102  
神林 飛志 ▶ p.103      仙石 浩明 ▶ p.104  
清水 亮 ▶ p.106

# FreeBSD、Linux、Windows、OSはどう変わるのか?

## 2013年、FreeBSDで注目すべき7つのポイント

BSD コンサルティング(株) 取締役/最高技術責任者

後藤 大地 GOTO Daichi ● daichi@bsdconsulting.co.jp ● Twitter: @daichigoto, @BSDc\_tweet

### ◎ 次世代パッケージ管理システム pkg(8)

2013年にリリースされるFreeBSDのパッケージは従来のpkg\_install(8)系のパッケージから次世代のパッケージ管理システムpkg(8)へ変わります。pkg(8)はPorts Collectionの利点を活用しつつ、yum(8)やapt-get(8)と同様の操作を提供するシステムです。パッケージのビルドクラスタおよび分散インフラを整え、最終的には週に1回のペースで最新のパッケージが提供される予定です。

pkg(8)はFreeBSDにおけるソフトウェアのインストールと管理に大きな変更をもたらします。2013年はFreeBSDのシステム管理という点においてエポックメイキングな年になります。

### ◎ 最新ツールチェーンLLVM/bmake

2012年末、amd64/i386 10-CURRENTのデフォルトのコンパイラはgccからLLVM Clangへ変わりました。gccのライセンスはGPLv3ですが、FreeBSDプロジェクトはGPLv3を採用したソフトウェアをベースコードにマージしない方針を発表しています。LLVMはBSDライセンスでFreeBSDと相性がよく、LLVM開発チームとの関係も良好であることからデフォルトのコンパイラ化へ向けた取り組みが数年続いてきました。

FreeBSDビルドシステムの基幹技術であるmakeはbmakeへ変わります。カーネルモジュールを併用してこれまでよりも細かいところまで依存関係のトラッキングを実施し、優れたビルドを実現します。LLVMデバッガやLLVM Clang++、i386/amd64以外への対応も進められます。2013年はツールチェーンが最新技術実装へ置き換わる

年でもあります。

### ◎ 最新ARM/MIPSコア対応/組込み機器

この2、3年でARMおよびMIPSの対応が一気に進みました。2013年はARM/MIPSともに最新コアに対応する年になります。とくにARMの対応が積極的で、2013年には従来よりも多くのARM搭載機でFreeBSDが利用できるようになる見通しです。すでにFreeBSDを搭載した組込み機器の販売が始まっていますが、機種がさらに増えることになります。

FreeBSDはもともとi386機で高速に動作するサーバとして重点的に開発が進められてきましたが、とくにこの2、3年で組込み向けOSとしての開発が進められました。今後、組込みはFreeBSDプロジェクトにとって重要な分野です。

### ◎ ベンダ/FreeBSD アプライアンス

2013年はFreeBSDプロジェクトとFreeBSDを活用したプロダクトやアプライアンスを製造販売している企業の交流がより積極的に進む年になるでしょう。多くのベンダがFreeBSDファウンデーションと連携を進めています<sup>注1</sup>。

ベンダはFreeBSDを活用した高機能/高性能アプライアンスの開発製造および販売に取り組んでいます。製品の差別化をはかる点においてBSDライセンスを採用したFreeBSDは扱いやすく、俄

注1) 日本にFreeBSDファウンデーションに相当する組織はまだありませんが、同等の組織ができるまでは企業からFreeBSDへの窓口としてBSDコンサルティング(株)をご利用ください。BSDコンサルティングはそういった目的もあって設立された企業です。



然注目されています。

## ◎ 仮想化技術BHyVe/KVM/Xen/Hyper-V

Microsoft Research が Hyper-V 向けの FreeBSD ドライバを発表するなど、仮想化における FreeBSD 活用が進んでいます。KVMでのFreeBSD利用、XenでのFreeBSD利用、Amazon EC2でのFreeBSD利用が進められています。

アプリケーションやサーバでの活用が期待される FreeBSD ホストの仮想化技術が BHyVe です。KVMに相当する仮想化技術で、プロセッサの仮想化支援技術を利用することを前提にシンプルな設計になっており、2013年はBHyVeの活用場が広がるものとみられます。

## ◎ マルチコアスケール++

プロセススケジューラ、I/Oスケジューラ、仮想マシン内部のロック機構、キャッシュヒット率などの改良が進められています。2013年はマルチコアにおける性能のスケールが進む年になるでしょう。24スレッドや32スレッドのマシンで従来よりも性能が発揮しやすくなるものとみられます。

## ◎ Capsicum 普及

FreeBSD 9に導入された新しいセキュリティ/コンパートメント技術Capsicum。2013年はこの技術がさまざまなコマンドやソフトウェア、サブシステムに取り込まれる年になりそうです。デフォルトの設定でよりセキュアなOS、それがFreeBSDです。SD



# 2013年はDebian7.0のリリースとさまざまな改善提案の行方に注目！

Debian Developer  
やまねひでき YAMANE Hideki ● henrich@debian.org

2013年はDebianにとって、どんな年になるでしょう？ 短い期間で新規バージョンが公開される他ディストリビューションと比較して、2年周期リリースのDebianはメディアでは目立ちにくいのですが、いくつかのトピックが考えられます。

## ◎ お待ちかね「Debian7.0」リリースへ

現在、フリーズ期間に入っているDebian7.0「Wheezy」ですが、暖かくなるころにはリリースされるでしょう<sup>注1</sup>。Linuxカーネル3.2<sup>注2</sup>を皮切りにGNOME3など<sup>注3</sup>の新しいパッケージが投入されています。

Debian6.0「Squeeze」のサポートは、7.0がリリースされてからきっかり1年で終了します。6.0のユーザの方は2013年中に7.0へのアップグレードの準備を整えておくとい良いでしょう。

また、7.0のリリースと共にリリースマネジメント上の理由からexperimentalに押し込められていたパッケージが次々とunstableに押し寄せるこ

ととなり、新しもの好きのユーザはうれしい悲鳴をあげることとなります。お楽しみに。

## ◎ 「クラウド」の積極的な受容へ

private メーリングリストでの話し合い<sup>注4</sup>から、公開の場でクラウド関連の議論をしようということでdebian-cloudメーリングリスト<sup>注5</sup>が開設され

注1) RCバグが0になったらリリースですので、<http://bugs.debian.org/release-critical>のグラフをチェックしてください。緑の線が次のリリースに対するRCバグ数を表します。

注2) 「もっと新しいバージョンは？」と思われる方もいらっしゃるでしょう。「3.4あたりを採用しないの？」と筆者もLinuxパッケージメンテナBen Hutchings(Linux kernel3.2シリーズのリリースマネージャーでもあります)に聞いたのですが「これが一番安定したバージョンとしてリリース/メンテナンスできる」との返答がきました。

注3) デフォルトのデスクトップ環境はGNOME3であろうと思われる。ただ、これを選択するtaskselパッケージのgitリポジトリではXfce4がデフォルトになっています……ややこしいですね。

注4) Windows Azureについて、Microsoftの方から「コミュニティの公式窓口がないのでDebianはサポートしていない」という話を聞いて、筆者が「やったらーじゃねーの」とプロジェクトリーダーを巻き込んで公式サポートへ向けてプッシュしたところ、議論が活発に行われたという背景があります。Azureについては今後にご期待ください。

注5) <http://lists.debian.org/debian-cloud>

ました。これをきっかけにAWSマーケットプレイス<sup>注6</sup>でのDebianプロジェクト公式イメージが提供されるなど、すでに目に見える形で動きが出てきていますので、興味のある方はこのメーリングリストをチェックしてください。

### 🎯「自動処理」とその整備

Debianパッケージには、そのパッケージのソースについてのcopyrightファイルがありますが、最近までその書式は定義されていませんでした。そのため、パッケージ(A)とパッケージ(B)を結合するときにライセンスに衝突が生じないかなどはいちいち開発者が確認をしないといけないという手間(とブレ)がありました。

これを改善すべく、機械的に処理可能なメタデータとして記述しようというプロポーザルが挙げられ、ポリシーに「Machine-readable copyright information」<sup>注7</sup>として記載されました。また、パッチについても同様に「Patch Tagging Guidelines」<sup>注8</sup>として書式が決まっていますので、このあたりの整備が進むと思われます。

また、逆説的ではありますが、今までざっくりとくくってライセンスを判別していたものについて、ファイル単位で確認をすることで新たな「ライセンス問題」が認識されて対処が必要な件が増えるでしょう(面倒ではありますが、「ディストリビューター」が担う責務ですので逃げられません)。

そのほか、「autopkgtest」パッケージとして実装されている自動テストへの各パッケージの対応や、Amazon Web Serviceを使って行われるようになった「全パッケージリビルド」<sup>注9</sup>テストの定期

的な実施も、今後の継続的インテグレーションでの品質向上の一環として期待したいところです。

### 🎯ユーザとの「距離」の改善？ ソーシャルメディアの利用

Christian Perrierによる「バグ投稿数のスピードが鈍っている。これはユーザが離れているという悪い兆候じゃないか？」というブログへのポスト<sup>注10</sup>をきっかけにしたさまざまな議論がありました。確かに、Debian BTSはほかに例を見ない「メールベース」のBTSですので、ユーザに対して敷居が高いという向きもあるようです。これに限らず、Debianは良くも悪くも老舗のために「昔ながら」のメーリングリスト／IRCベースでのコミュニケーションが主な手段となっています。しかし、みなさんもご存じのとおり、現在、ユーザサイドではWebベースのツールの利用が盛んです。7.0のリリース以降にはなりますが、ユーザとのギャップを縮める試みが盛んに行われるようになるでしょう。

なお、現プロジェクトリーダーのStefano Zacchiroli(@zacchiro)は月次レポートの発表のみならず、twitter/identi.caでの発言を活発に行っています。日本でも「軟式」アカウント@debianjpでDebian情報をユルめに流していますので、ぜひフォローください。SD

注6) <https://aws.amazon.com/marketplace/>

注7) <http://www.debian.org/doc/debian-policy/ch-docs.html#s-copyrightfile>

注8) <http://dep.debian.net/deps/dep3/>

注9) <http://www.lucas-nussbaum.net/blog/?p=718>

注10) <http://www.perrier.eu.org/weblog/2012/10/09#690000>



## 2013年、Ubuntuは アグレッシブに変化する！

Ubuntu Japanese Team

あわしろいくや AWASHIRO Ikuya ● ikuya@fruitsbasket.info

### 🎯2013年はどんな年？

Ubuntuからみて2013年はどんな年なのでしょう。今一度リリースに関して思い出してみると、

偶数年の4月に5年間サポートが継続するLTS (Long Term Support／長期サポート) がリリースされ、それ以外は1.5年サポートです。というこ

とは、2013年は奇数年ですのでLTSがリリースされないということです。これが何を意味するかというと、Ubuntuがアグレッシブに変化する1年ということです。

## 🎯 これまでに登場したもの

これまでにデモなどが登場したものの、実際に製品としては(筆者の知る限り)存在しないものがあります。具体的にはUbuntu for Android<sup>注1</sup>やUbuntu TV<sup>注2</sup>です。前者はAndroidの中でUbuntuを動作させるという意欲的なものであり、後者はUnityを拡張してTVに合ったインターフェースに変更したものです。とくに後者はもともとUnity 2Dをベースに開発していましたが、12.10で削除されたのでUnityで動作するように作業が進んでおり<sup>注3</sup>、13.04のころにはリリースされるのではないかと思います。HDMI端子に接続するAndroidスティックがいくつか販売されていますが、これの中身がUbuntu TV、というものが販売されたらおもしろいのではないのでしょうか。

また、Ubuntu Web Apps<sup>注4</sup>も本格的に使用されるようになるでしょう。

## 🎯 13.04は

Ubuntuの創始者でCanonicalの創業者兼COOのMark Shuttleworth氏は、CanonicalがUbuntuに実装する新機能は強い批判や非難を浴びることが多く<sup>注5</sup>、これに辟易としたのか、13.04では直前<sup>注6</sup>までいくつかの新機能を発表しないと公言しています<sup>注7</sup>。

とはいえ従来どおりさまざまな計画が練られており、そのうち注目すべきものがいくつかあります。筆頭はSteam<sup>注8</sup>が対応したことによるゲームプラットフォームとしてのUbuntuの最適化です。SteamはValve Softwareのゲームプラットフォームで、2012年7月16日にUbuntu用のクライアントをリリースすることを発表しました<sup>注9</sup>。11月末現在はベータテスト中です。Ubuntuはゲームのことは考えてこられなかったので、グラフィックや

サウンドなどを遅延なく、かつ高速に動作させる機会があまりありませんでした。しかし、これを機にゲームプラットフォームとしての開発が進み、ゲームがない、または少ないというLinuxの弱点がまた1つ克服されることになりそうです。

## 🎯 Ubuntu Serverは

Ubuntu Serverのキーワードは引き続きOpen Stack(言わずもがなのオープンなクラウドプラットフォーム)、Juju(Ubuntu Serverのデプロイツール)、Charm(Juju用のレシピ)だろうと思われます。しかし、LTSでないUbuntu Serverを実運用するのはなかなか厳しいものがあり、12.04を使用しつつ14.04に備えて新機能を勉強しておく、ということになるのではないのでしょうか。本誌でもOpenStackの特集を期待したいところです。

## 🎯 さらに普及が進むARM

サーバ、モバイル(ひょっとしたらデスクトップ)ではますますARMの普及が進み、デバイスも多数発売されることでしょう。そして、Ubuntuはそれらのデバイスで動作するように開発が進むはずで、現段階でもNexus 7用のUbuntuイメージが(当然プレビュー版ですが)すでに存在します<sup>注10</sup>。もともとUnityはあらゆるデバイスで使用しやすいインターフェースを目指しており、タブレット端末との相性もいいはずで、ARMサーバの普及が進むのは2014年以降かもしれません。Linaro<sup>注11</sup>の活動が活発になっていることから、ARMへの最適化が進むのは確実です。SD

注1) <http://www.ubuntu.com/devices/android/>

注2) <http://www.ubuntu.com/devices/tv/>

注3) このあたりの事情に関しては本誌2012年11月号「Ubuntu 12.10の新機能と変更点」をご覧ください。

注4) <http://developer.ubuntu.com/resources/technologies/webapps/>

注5) 直近だとUbuntu 12.10のUnity Shopping Lens(Unityで検索するとAmazonの商品紹介が表示される)です。

注6) それでなくても充分直前に発表されているのですが……。

注7) <http://www.markshuttleworth.com/archives/1200>

注8) <http://store.steampowered.com/about/>

注9) <http://blogs.valvesoftware.com/linux/>

注10) <https://wiki.ubuntu.com/Nexus7/Installation>

注11) <http://www.linaro.org/>



## 2013年、Windows 8/Windows Serverはシステム構築にどんな影響を与えるか？

グローバルナレッジネットワーク ラーニングサービス本部

横山 哲也 YOKOYAMA Tetsuya

### ◎ 仮想マシンの拡大

近年のITシステムのキーワードは間違いなく仮想化とクラウドです。Windows Server 2012でもサーバ仮想化システム「Hyper-V」関連機能やクラウドとの関係が強化されました。また、記憶域プールにより、物理ディスクも仮想化できます。その結果、サーバ構成が大きく変化し、エンジニアに必要なスキルも変化します。

Hyper-Vの機能強化には、サポートするCPUコア数(64から320へ拡大)や、仮想ディスク容量の拡大(最大2TBから64TBへ増加)のほか、クラスターを使わないライブマイグレーション、ディザスタリカバリ用としての仮想マシン複製機能(レプリケーション)などがあります。これにより大規模なサーバを、柔軟に構成できるようになりました。2013年の物理サーバは「仮想マシンの入れ物」としてのみ意味を持つでしょう。

### ◎ 仮想スイッチの強化

あまり目立たないのですが、Hyper-Vで本当に重要な強化点は仮想スイッチ(仮想ネットワーク)の機能拡張です。Windows Server 2012の仮想スイッチは従来のVLANに加え、大規模環境向けにプライベートVLANやNVGRE(Network Virtualization using Generic Routing Encapsulation)が利用できます。いずれもPowerShellのコマンドレットでのみサポートされる機能ですが、System Center Virtual Machine Manager 2012(SCVMM 2012)SP1を使えば比較的楽に設定できます。

もちろん、従来のVLAN機能も使えるので、Hyper-V内の仮想スイッチは、市販されている企業向けのスイッチ製品と比べても遜色ないものとなりました。しかもHyper-V仮想スイッチには機能拡張用のAPIがあり、サードパーティによる拡張が可能です。すでにシスコシステムズやNECから対応製品が出ています。

### ◎ クラウドの本格利用

Windows Server 2012は「クラウドOS」をうたっています。「Windows Azure Online Backup サービス」を契約すれば(2012年11月現在ベータテスト中)、バックアップ先としてクラウドが選べます。また、Windows Server 2012の標準機能であるActive Directory Federation Services(AD FS)を構成すれば社内のActive DirectoryとOffice 365の統合認証が実現できます。さらにWindows AzureのVMロールとVPNを使えばクラウド上のWindows Server 2012を社内システムと同等に利用することもできるのです。

こうなると、本当に社内にサーバが必要なのか、という議論が当然出てくるでしょう。自社のサーバを全廃するケースは少ないにしても、規模を縮小する可能性は十分にあります。

### ◎ IT技術者はどこに行くか

多くの組織では、サーバ管理者がOSを管理し、ネットワーク管理者がL2/L3スイッチの管理をしています。ではHyper-Vに含まれる仮想スイッチの管理は誰がするべきでしょう。Hyper-Vの管理にはWindows Serverの知識が不可欠ですが、サーバ管理者はネットワークが苦手で、ネットワーク管理者はOSが苦手です。

一部の組織では、ネットワークエンジニアがサーバを担当できるようにスキルシフトが始まっています。一方、サーバエンジニアはアプリケーション方向にシフトします。2013年はこうした流れが加速するでしょう。

クラウドを使えばOSのインストールは不要ですし、構成も最小限で済みます。しかし、どのサーバを社内に残し、どのサーバをクラウドに配置するかを判断するのは人間の仕事です。仮想化とクラウドの利用により、2013年は社内データセンター構成の見直しと、エンジニアのスキルシフトの年になるでしょう。SD

# ソフトウェア開発はどうなるのか

## まだまだ注目技術に事欠かない Web標準技術

(株)オープンウェブ・テクノロジー 代表取締役

白石 俊平 SHIRAISHI Shumpei ● Twitter: @Shumpei

筆者はHTML5を中心としたWeb標準技術に注目していますが、現在凄まじい速度で進化を続けている分野です。注目に値する技術は多岐に渡るため広く浅い紹介となりますが、ご容赦ください。

### Web制作／デザイン系での注目技術

2012年にWeb制作の世界を席卷した「レスポンシブWebデザイン」は、単なるWebデザインの枠にとどまらず、見積もりから開発のワークフローに至るまで影響を及ぼす奥の深い技術です。2013年も引き続き注目を集めるでしょう。

また、Web制作の世界では、JavaScriptを使用した高度な「アプリケーション化」が進んでいます。2012年は「視差スクロール」というエフェクトが流行していましたが、今後も表現の革新は続くものと考えられます。筆者は個人的には、パノラマ画像を効果的に使用したWebサイトが流行するのではないかと睨んでいます。

### Webアプリ開発における注目技術

Webアプリケーション開発においては、まず「単一ページアプリケーション」の開発手法が注目を浴びるのは間違いないでしょう。単一ページアプリケーションとは、従来のように複数のページを遷移するのではなく、1つのHTML(DOM)上で構築されるWebアプリケーションです。画面が遷移する煩わしさからユーザを解放できるだけでなく、アプリケーションの状態が持続するので、開発者にとってもGUIを構築しやすいという利点があります。この分野では、Backbone.jsやAngular.jsなどのMVCフレームワーク、jQuery Mobileや

Sencha TouchといったUIフレームワークが中心的な役割をはたすでしょう。

また、バックエンドを含めたWebアプリケーション開発を促進するという点で、Meteor<sup>注1</sup>やWakanda<sup>注2</sup>といったフルスタックのJavaScriptフレームワークや、BaaS(Backend as a Service)といった動向にも注目したいところです。

もうひとつ注目すべきは、Webサイトのオフライン化です。オフライン技術はIE10でもサポートされたため、採用の裾野が広がるものと考えられます。Webサイトのパフォーマンス向上技術として採用が広まりつつ、完全なオフライン動作を提供するWebアプリケーションも少しずつ数を増やしていくでしょう。

### 開発ツール、プラットフォーム、仕様など

HTML5関連の開発ツールとしては、筆者はAdobe Edge Animateに注目しています。このツールを中心として、かつてのFlashクリエイターのような「Web(HTML)クリエイター」という役割がその萌芽を現すのでは、と考えています。

また、2013年にはFirefox OSやTizenといった、Web技術を中心としたモバイルOSも少しずつ存在感を増すでしょう。そして、こうしたプラットフォームの細分化によってマルチプラットフォーム開発に対する要求がさらに高まり、PhoneGapやTitanium Mobileといったハイブリッドソリューションも勢いづくと思われます。

そして、先進的なWeb標準仕様についても、仕

注1) <http://meteor.com>

注2) <http://www.wakanda.org/>

様の成熟と実装が進むものと思われます。字数の都合上、ここではキーワードの紹介に留めますが、

Web Intents、WebRTC、Web Components、WebSocket といった仕様の動向は要注目です。SD



## ソフトウェア開発方法論としてのDDD、ユビキタス言語、Scala+Play2の可能性

株豆蔵

羽生田 栄一

HANYUDA Eiji

### 🎯 オブジェクト指向開発から ドメイン駆動開発へ

現在、注目している開発方法論は、やはりDDD(ドメイン駆動開発)ということになります。DDD自体はオブジェクト指向開発の延長線上にあります。Javaなどのオブジェクト指向プログラミング言語での開発から始まり、カプセル化を中心思想としたオブジェクト指向設計が次に続き、ここまではある程度開発の常識化してきたと言ってもいいでしょう。しかし、その後、モデリングが開発に定着したか、あるいはシステム化の対象となっている問題領域(ドメイン)そのものをモデル化し、それを開発の中核に据えるという発想自体が定着したかと言うと、残念ながらそのような状況にはありません。

これからの数年がDDDの正念場になるのではないかという予感には、2つ理由があります。1つめは、UMLの限界が明らかになりUMLはしょせんは単なる表記法に過ぎず、単純にUMLでユースケース図やクラス図を描いたからと言って必ずしもオブジェクト指向デザインにはならないということにみんなが気づいた点。言い換えるとオブジェクト指向デザインの指針/方法論が必要だと認識しました。いくらアジャイルで開発するといっても、オブジェクト設計に関する共通理解がなければ、ペアプロもリファクタリングもうまく機能しません。そこに大規模なシステム開発へのオブジェクト指向デザインの適用パターン言語としてのDDDの存在理由があります。

2つめとして、オブジェクト指向言語の代表格であるJavaが枯れてきてそのポストJavaとし

てさまざまな言語が取りざたされる中で、安全な強い型づけ言語でありながら型推論による軽量スクリプト言語並みのコンパクトなコーディングを可能とするScalaの現実的な可能性がだいたい見えてきました。実務的なそここの規模の開発にScalaが使えるかどうか現在世界中のプロジェクトで検証が行われているところです。しかも、Scalaでのクラス定義は非常にコンパクトでUMLのクラスアイコン相当の情報量を実行可能な形で持つことになります。さらにScalaの柔軟な拡張言語定義能力をもってすればDSL(ドメイン特化言語)の延長線上にユビキタス言語とモデリング言語の融合された世界が垣間見れるというものです。

### 🎯 DDDのさらに先には……

DDD以外に視野を広げると、Haskellを筆頭にScalaを含む関数型言語やSPIN、Alloyなどの軽量形式手法ツールの普及があります。これらは、DDDの先を見据えて、世界のソフトウェアエンジニアと渡り合っていくためには、今後必要な武器になるものと思われます。

さらに視野を広げると、製造業やエンジニアリング分野でのシステムズエンジニアリングおよびその記述言語SysMLの浸透、ビジネス領域でのBABOKの浸透がかなり進むのではないかと期待しています。いずれも現在のビジネスを見直しグローバルに自分たちのやっていることを再度定義しなおす際の良いツールになると思います。

最後にScalaとあわせてRailsライクなPlay2フレームワークの普及にも期待しています。SD



# Google Go言語はどうなった？

東京大学 情報理工学系研究科  
千葉 滋 CHIBA Shigeru



## Google 社内で導入が進む Go 言語

先日、ACM SPLASHという国際会議(いわゆる学会です)に参加したのですが、そこでRob Pike氏の講演を聞きました。主題はGoogleにおけるGo言語についてです。会場の外にはマスコットであるgopherの人形が山と積まれ、聴衆が自由にもらえるようになっていました。講演では、いかに#includeが非効率的で、Google社内のソフトウェアをビルドするたびに同じヘッダファイルが何千回!と読まれているか、それによってビルドにどんなに時間がかかっているか、Goはそれをいかに解消しているか、云々が語られました。すでにGoogle社内ではあちこちでGo言語が使われていて、生産性向上に貢献しているそうです。



## 言語は簡単に作れる？

とはいえ、2009年に発表され大きな注目を集めたGo言語ですが、このところあまり話題に上りません。発表当初は私も周囲の人から「Go言語についてどう思います? 勉強したほうがいいですよね?」とよく聞かれました。私も「そうですね」と適当に答えたのを覚えています。正直に告白すると、私がGoの概要について聞いたのは、このRob Pike氏の講演が初めてでした。大学の先生というものは、得てして世相にうといようで、私の周囲の人達も皆、初めてGoの中身を知った、などと言っていました。

Googleをもっとしてもこんなのですから、新しいプログラミング言語を開発し第二のMatz氏を目指す、という野望はなかなか大変そうです。とはいえ新しいプログラミング言語を開発する技術的なハードルは昔に比べればだいぶ下がりました。言語処理系(コンパイラとインタプリタを合わせてこう総称します)は大きく前半の構文解析と後

半のコード生成あるいは実行に分かれます。前半の構文解析は、既存のパースジェネレータ(構文解析器生成器)やパースコンビネータを使えば、比較的手軽に実装できますし、後半もLLVMやJava仮想機械など既存のものを活用できます。あとはセンスです。



## 新しい言語の作り方

最近ではもっと簡単なアプローチもあります。Embedded DSL(組込み型ドメイン専用言語)です。アイデア自体は昔からありますが、Scalaなどのおかげで最近よく利用されるようになりました。要は、一見、新言語で書いたプログラムのように見えるけれども、実はライブラリを使って書いた普通のScalaプログラム、のように新言語を実現する方式です。Scalaのimplicit conversionや、古くはLispのマクロ、C++の演算子オーバーロード、といった機能を駆使して新言語を実現します。実装はトリッキーになりがちですが、うまくいけば簡単に新言語を実現できてしまいます。

誰もが新言語を簡単に作れるようになって、アプリケーション分野ごとにたくさんのオレオレ言語が登場してくると面白いことになりそうです。これまではフレームワークの開発と活用が重要なテーマでした。Rubyの普及もRailsあってこそ、でしょう。しかしフレームワークによる生産性の向上には限界があり、そこから先は、フレームワークでなくそのアプリケーション分野専用のプログラミング言語を開発して活用した方がよくなります。実際、GPGPUやXeon Phiを使ったビッグデータ処理などと言い始めると、CUDAやOpenCLのような専用言語が必要になってきます。さて、オレオレ言語、百花繚乱となりますか。SD



## プログラミングに役立つ数学

結城 浩 YUKI Hiroshi ● <http://www.hyuki.com/> ● Twitter: @hyuki

こんにちは。数学物語『数学ガール』を書いている結城浩と申します。今日はプログラミングに関係がある数学として「指数的爆発」のお話をします。

### 🎯 動作テスト

プログラムの動作テストをするとき、そのプログラムにオン／オフできるオプションが1個あったとしましょう。そのとき、動作テストはオプションがオンのときと、オプションがオフのときの2回必要になります。

オプションがAとBの2個あったなら、テストは4回必要になります。これを繰り返していくと、オプションがn個になると、テストが2のn乗、つまり $2^n$ 回必要になることがわかるでしょう。「 $2^n$ になる」という式を見た時点で「これはまずい!」と気づかなくてははいけません。なぜなら、指数的爆発を起こして、オプションの個数に対してテストの回数が急激に増加するからです。

「1個くらいオプションが増えてもいいや」と思っているはいけません。オプションが1個増えるたびに、テストの回数は2倍になるのですから、オプションが10個になったら、テストは $2^{10}$ 、つまり1,024回必要になってしまいます。オプションが20個になったら、テストは $2^{20}$ で、なんと1,048,576回必要です。100万回以上です!

nに対して $2^n$ になるのは指数的爆発の例です。

### 🎯 二分探索

このように、指数的爆発は敵にまわすとやっかいです。逆に使って味方につければ心強いものです。指数的爆発を効果的に使ったアルゴリズムが二分探索です。バイナリサーチとも呼ばれています。たくさんのが順序よく並んでいる中から何かを探したいときに、まずちょうど真ん中のものを取り上げ、それよりも前にあるか後にあるかを調べます。前半分にあることがわかったら、今度はその真ん中を調べます。このような「半分

にする」という操作を繰り返して探するのが二分探索です。

半分にする操作を繰り返せば、1,024個のものであっても、たった10回の比較で見つけることができます。20回比較すれば、なんと1,048,576個のもの、つまり100万個のものの中から目的のものを見つけることができます。

二分探索は、バージョン管理システムgitでバグの発生した変更箇所を見つけるためのコマンドbisectでも使われていますね。

指数の逆は対数です。二分探索を使うと、X個のものから $\log_2 X$ 回の比較で目的のものが見つかります。対数を表す数式 $\log_2 X$ は難しそうですが簡単です。たとえば、 $\log_2 2 = 1$ で、 $\log_2 4 = 2$ で、 $\log_2 8 = 3$ です。2、4、8を2進数で表すと、それぞれ10、100、1,000ですね。つまり、Xが2の累乗のとき、 $\log_2 X$ は「Xを2進数で表したときに0が何個並ぶか」を表しているのです。

### 🎯 プログラミングの目的

プログラムは大量のデータを扱います。普通なら何万年もかかる処理を、工夫したアルゴリズムにより数秒で済ませることがよくあります。

大量のデータを扱うときには、指数的爆発に陥らないようにすることと、対数を使って管理可能なレベルまで引き下げることが大切です。アルゴリズムの複雑さや、データ量を評価して表現するときには数式がしばしば現れます。 $2^n$ が出てきたら「指数的爆発だ」と注意し、 $\log_2 X$ が出てきたら「対数で押さえられてる」と思えます。

日々のプログラミングで高度な数学を使う人は少ないかもしれませんが、指数的爆発に陥らないように注意を払うこと、そして押さえるということはよくあります。数式に対する苦手意識をなくすだけでも、日々のプログラミングに役立つのではないのでしょうか。SD



# IT最前線シリコンバレー発、 2013年注目のビッグデータ技術

Treasure Data, Inc. Founder & Software Architect  
古橋 貞之 FURUHASHI Sadayuki

シリコンバレーに移住して早1年。ここでは著名なOSSの開発者を含むトップレベルのエンジニアが集うイベントが常にか開かれており、とても活気があります。その中でも最近ホットな話題は、やはり「ビッグデータ」でしょう。ここではビッグデータを扱うシリコンバレーエンジニアの1人として、自分の経験を述べてみます。

## ◎ 古くて新しいビッグデータ解析

データ解析は、古くから数多くの手法が確立された分野です。たとえば、多次元的なデータをさまざまな面から分析したり絞り込んだりするためには、OLAP キューブと呼ばれるデータ構造が使われます。データの可視化を行う BI ツールは、この OLAP キューブに対して解析を行うように設計されていることが多く、OLAP のコンセプトを理解しているとスムーズにデータ解析を行うことができます。このほかにも、相関分析に代表されるさまざまな統計的な手法が存在します。

ただし、これらの手法を適用するデータは変化しています。近年実装が成熟してきた分散処理技術を用いることで、従来ではデータ解析には適さなかったと思われてきたデータに対しても解析を行えるようになりました。たとえば、初めのうちこそ少量だが規模が拡大するほどに単調増加していくユーザの行動記録や、すべての検索履歴、複数のサービスを横断した全購買履歴の解析などです。

ソーシャルメディアの台頭によって、情報を Web 上に投稿することに誰も違和感を持たなくなった結果、メモリや CPU の性能向上よりも速くデータ量が増大していくのが 2012 年の情勢でした。これらのデータを JSON 形式で収集し、解析可能な形で保存しておくというところまでは、現在、当然のように行われています。しかし、こうして大量に溜まっていくデータをどのように捉え、どのような面から見れば有益な情報が得られるのかを試行錯誤していくことが、現在の課題で

す。このような課題に対して、旧来から改善され続けている解析手法を応用していく技術は、2013 年に必要とされる技術だと言えるでしょう。

## ◎ データ解析の技術

現在具体的に行われている方法の1つは、Hadoop を使ってデータを集計して OLAP キューブに変換し、その OLAP キューブを可視化ツールを使って解析するという手法です。OLAP キューブの保存には PostgreSQL や SQL Server などの RDB が使われ、Excel を含む BI ツールで扱うことができます。しかしこの方法の問題は、Hadoop では直近のデータを扱うことが難しい点と、OLAP キューブが巨大化すると RDB では扱えなくなってしまう点です。そこで現在注目を浴びている技術が、分散ストリーム処理と分散集約です。

## ◎ ストリーム処理と分散集約

分散ストリーム処理は流れてくるデータを短期間だけメモリ上に溜め、そこで集計処理を行うものです。数日や数ヶ月といった単位の長期間に渡るデータの解析はできないものの、数分間程度の単位で行う解析であれば非常に効率的に処理できます。

分散集約は Google が発表した論文から端を発する技術です。処理ノードをツリー状に連結し、データセットをツリーの葉から根に向かって集約しながら転送していくことにより、大規模な分散処理を可能にします。処理の結果としてデータ量が減る種類の処理 (SUM など) を低遅延で実行できる一方で、データが減らない処理 (JOIN など) が苦手であるという特徴があります。

分散ストリーム処理も分散集約も、高い耐障害性を特徴とする MapReduce とは異なる技術であり、相互に補完し合うものです。どちらも実装は発展途上で、2013 年にかけてじっくりと議論していくことになりそうです。SD

# 情報技術の新しい流れ

## RedHat社内でも注目、Linuxで利用できる最新メニーコア技術

レッドハット(株) 小島 克俊 KOJIMA Katsutoshi

### 2013年注目の技術

ソフトウェアは土台であるハードウェアの進化に伴って革新されます。2013年はHPC(High Performance Computing、高速演算)向けの「メニーコア」製品が増えることに注目しています。

### CPU性能の課題

近年のCPUの周波数は4GHz程度で頭打ちです。CPUの周波数を高めるため微細化をすすめると回路を流れる電気の多くがリーク電流として消えることになり、計算能力が高まらないからです。むしろ周波数を抑えたコアの数を増やし、キャッシュのサイズを大きくし、命令セットを増やすことでその性能は強化されています。

### MIC(マイク)

そのような状況でインテルはPCIボードの形で新たなアーキテクチャ、MIC(メニー・インテグレートッド・コア、略してマイク)を製造し始めました。製品は開発コードネーム Knights Cornerとして知られていましたが、2013年上半期から正式にIntel Xeon Phi(ファイ)コプロセッサとしてXeon搭載サーバのオプション扱いで複数のメーカーから出荷されます。Xeon PhiはXeonの行列演算を補うコプロセッサという位置づけです。最初の製品5110Pは60コア/240スレッドです。

このMIC、1枚で1TFLOPS、つまり1秒間に $10^{12}$ 回浮動小数点の演算能力があります。といってもよくわからないので一般的な2CPUの1Uサーバと比較すると、おおむねMIC1枚で3倍、2枚で6倍の計算ができることになります。しかもMIC

の電力消費は225WとCPU2個分くらいですので、計算能力に対して消費電力は抑えられます。

### MICの並列演算応用例、資源探査

MICによる技術革新が大きな影響を与える1つの応用例として資源探査があります。海底などの広範囲な地質調査を行う場合、微妙な地質の違いを音が反射する具合から推定します。このとき、同じデータを何度も再計算し、徐々に地層境界や断層の輪郭をはっきりさせていきます。最終的にその結果をみて掘削場所を決定するのです。

このときの計算はデータを小さく分割して行います。その結果を最後に集めればよいので複数の計算機をネットワークでつなぎ、クラスタを組むこともあります。

### 計算機的能力

このような並列計算において1台の計算機的能力はベクトル演算能力に比例します。2011年ごろに販売されていたインテルのCPU、Xeonの場合、ベクトル演算(SIMD)は128bit幅でした。2012年にはAVX命令により256bit幅になりました。そしてMICの命令セットでは512bit幅です。クロックあたりで処理できる量が倍増し続けたのです。

### MICとオープンソース

たいへんうれしいことにMICが利用できる最初の環境としてLinuxが選択されています。検証対象はRHEL6.2などです。CPU命令にかかわるLinuxの機能拡張はもちろんオープンソースですし、MICを動作させるために必要なソフトウェア

もオープンソースとして提供されています。

とでメニーコア技術が発展してほしいものです。

## ソフトウェア開発

MICの登場でシステム構築とソフトウェア開発は次の3段階で工夫できるようになりました。

システム構成	ソフトウェア開発
CPU	マルチコア
MIC	メニーコア
サーバとネットワーク	クラスタ

これまでもメニーコア技術としてGPGPUを利用する方法がありました。新たに登場したMICはGPGPUでのプログラミングの課題を解消できる可能性があります。ぜひそれぞれの良さを競争するこ

## これからの展望

MICによりこれまで数年かかった事業がもっと速く進む分野が出てきます。とくに石油やガスなどの資源探査、製薬などの分野では経済的な効果が非常に高く、私たちが想像しているよりずっと大きなお金が動きます。それらの発展に関連しているメニーコア技術がLinuxとともに進化することは間違いないでしょう。SD

### <並列計算全般の参考文献>

Michael McCool ほか 著『Structured Parallel Programming: Patterns for Efficient Computation』Morgan Kaufmann Publishers  
資源探査については10章 Forward Seismic Simulation (P.265)



## そのメールのやりとり、開発の生産性を下げていませんか？

サイボウズ株式会社 藤村 能光 FUJIMURA Yoshimitsu

### メールがなくなると、仕事はどう変わるのか？

「サイボウズでは社内でのメールのやりとりはゼロなんです」。こう言う多くの人に驚かれますが、事実、サイボウズでは社内の情報共有にメールを使いません。社外の人と共同でプロジェクトを進める場合も、メールを使わずにコミュニケーションをしています。本稿では「メールを使わない情報共有の実態とその効果」についてお伝えします。

### メールがエンジニアの生産性を下げる理由

エンジニアにとってメールが使いにくい理由は、「やりとりが多くなって、開発のスピードが落ちる」「経緯を追いにくい」といった点でしょう。要件設計から計画、開発、試験に至る各フェーズで情報共有は必須ですが、メールでのやりとりは適していません。

メールは1対1のやりとりをする「手紙」を電子化したもので、チームのコミュニケーションには不

向きです。返信が積み重なって議論の経緯がわからなくなったり、重要な情報が整理できず埋もれたりしがちです。受信メールが膨大になると頻繁な返信がなくなり、プロジェクトの温度感が下がる結果、開発がスムーズに進まなくなります。

開発を成功させるには、エンジニアや企画担当者が簡単に情報を共有できる環境が必須です。ではメールを使わない開発チームの情報共有とは、どのようなものなのでしょう。

### 社内でメールを使わず、オンラインでタスク管理

サイボウズの業務アプリケーション構築向けクラウドサービス「kintone」<sup>注1</sup>の開発チームを例に挙げます。同チームでは、開発の担当分野が異なるメンバーが、コードのレビューや試験といったタスクをkintoneに登録しています。

各タスクには担当者や優先度、詳細などの入力項目があらかじめ設定されており、メンバー全員

注1) <https://kintone.cybozu.com/jp/>

が同レベルの報告を過不足なくできるようになっています。各タスクに対してコメントも付与できるため、メールやWord、Excelを使った堅い報告は必要なくなり、簡単に進捗を共有できます。

Webベースのタスク管理はメールに比べて入力操作简单で、開発の合間に無理なくコメントができます。結果として、メンバーのこまめな進捗情報の共有が習慣化します。タスクごとに情報が整理され、開発の経緯が集約されるようになるため、メールにありがちな「1つのスレッド内に複数のやりとりが同時並行して行われ、経緯がわからなくなる」といったミスをなくせます。

### 🎯 社外の人とも、1つの情報共有スペースでやりとりを

社外の人と開発を進める場合は、企業間で使える無料グループウェア「サイボウズLive」<sup>注2</sup>を使っています。

(株)レレレが提供するWebサービス「Coffee Meeting」は、別の組織に所属する企画担当者、エンジニア、デザイナーが集まって開発されました。開発チームだけが参加できる情報共有スペース(グループ)を作り、「β版へのフィードバック」や「機能の改善点」といった掲示板を立て、議論を進めていきました。各トピックにはコメントや意思表示を示す「いいね！」が付けられます。

メールでの情報共有には、文面の作成や受信メールの確認に時間をとられがちです。「了解です」といった確認メールをわざわざ送るのも、骨が折れるでしょう。サイボウズLiveの「いいね！」機能は既読管理に使われることが多く、「了解しました」といったコメントを投稿せずに、このボタンだけでやりとりを完結しています。さらに、サイボウズLiveでのやりとりに慣れてくると、「〇〇様」といった宛名や「お世話になっております」といった枕詞も使わずに、やりとりができます。

このようにメールではなく、開発メンバーだけがアクセスできるディスカッションの場を使うことで、開発に必要な情報だけを手軽にやりとりでき、開発のスピードも増していきます。

オンラインでのタスク管理やディスカッションは、開発チームの情報共有を円滑にします。メールのわずらわしさをなくし、生産性を高める情報共有をみなさんの開発チームでも取り入れてみませんか。SD

注2) <https://live.cybozu.co.jp/>

注2) <https://live.cybozu.co.jp/>



## クラウドの向こう側、AWSクラウドデザインパターン(CDP)！

cloudpack(iret(株)) CTO 鈴木 宏康 SUZUKI Hiroyasu ● Twitter: @suz\_lab

Amazon EC2の導入設計／運用／保守のサポートサービス「cloudpack」に携わっている鈴木(suz-lab)と申します。黒いブログ<sup>注1</sup>でおもにAWSの利用に関するノウハウを発信しています。

2011年、AWSの東京リージョンが開設され、日本でもクラウド(AWS)の利用が検討され始めました。そして2012年、筆者の周りではとくにWeb系のシステムでクラウドを利用することが非常に多く、クラウドがインフラの選択肢の1つとして当たり前に認知されてきたのでは、と思っています。

さらに2013年、もうクラウドの利用は当たり

前となり、その上のクラウドの使い方、つまり、「クラウドデザインパターン(CDP)」がITエンジニアの重要なスキルになると筆者は考えています。

### 🎯 AWSクラウドデザインパターン<sup>注2</sup>

AWSクラウドデザインパターン(以下CDP)は、AWSを使ったシステムアーキテクチャ設計を行う上で発生する典型的な問題とそれに対する解決策／設計方法を、わかりやすく分類してノウハウ

注1) <http://blog.suz-lab.com/>

注2) <http://aws.cloudpattern.org/>

として利用できるように整理したものです。

しかしそれらの手法そのものよりも、CDPではクラウドを用いてシステムをアーキテクティングする際の心構え、言い替えれば「クラウド脳」とさえ言える「クラウドアーキテクティング原則」を重視しています。

## 🎯 クラウドアーキテクティング原則

「クラウドアーキテクティング原則」は次の6カ条となります。

- ・できるだけサービスを利用
- ・机上実験よりも実証実験
- ・スモールスタートからスケールアウト
- ・変化に対し全レイヤで対処
- ・故障のための設計 (Design For Failure)
- ・最初だけでなく周期的なカイゼン

とくに「できるだけサービスを利用」と「机上実験よりも実証実験」、そして「スモールスタートからスケールアウト」に関しては、AWSを利用したシステム構築を行う上で、今までの考え方を意識して変えていかなければならない部分、と考えています。

## 🎯 できるだけサービスを利用

AWSにはストレージサービスのS3、キューイングサービスのSQSなど、ITシステムでよく利用されるコンポーネントがすでにいくつか用意さ

れており、自分で実装するよりも、可用性／耐久性の高いサービスを手軽に低コストで利用できるようになっています(図1)。これらのサービス／コンポーネントを適材適所で利用できるスキルが、これからのクラウド時代でITエンジニアにとって必須のものになっていくと考えています。

## 🎯 机上実験よりも実証実験

従来、陥りがちであった、机上実験に時間をかけ、結局、過剰な見積もりでシステム構築してしまうという現実、クラウドの瞬時に安く調達できる良さをまったく活かしきれていません。

手軽なクラウドでとにかく実環境を構築し、できることから試してみて、走りながら実際の本番環境に近づけていくイメージが大切だと思います。サーバのサイジングなども、本番環境が一通り固まるまで行わず、その後、負荷テストを行いながらプロジェクト終盤で決定するほうが無駄がありません。今後、この方法が主流になっていくと考えています。

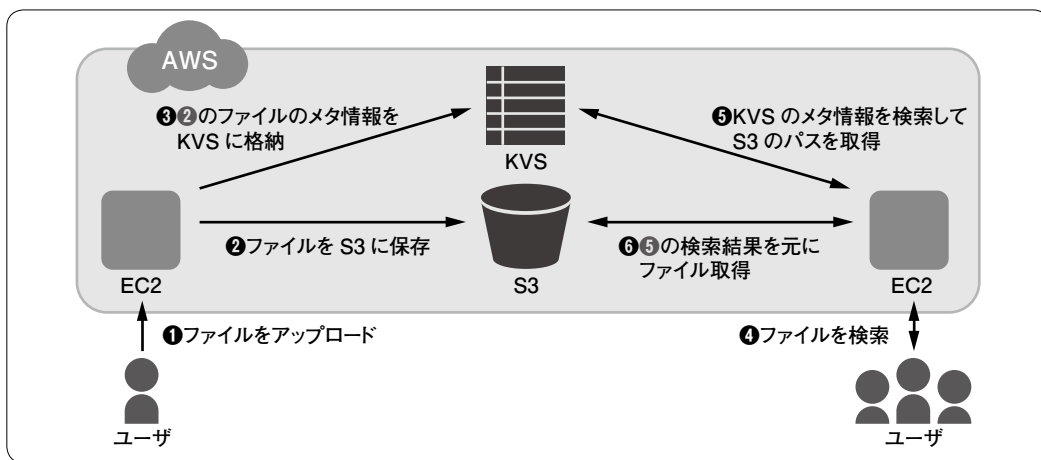
そうすると、気軽にできる負荷テストツールの必要性も増し、「neustar」<sup>注3)</sup>などの負荷テストサービスもだんだんと注目されてくると思います。

## 🎯 スモールスタートからスケールアウト

スモールスタートは、従量課金で調達が容易に

注3) <http://www.neustar.biz/enterprise/web-performance>

▼図1 コンポーネントの利用例 (CDPの1つStorage Indexパターン)



できるクラウドの良さを活かした考え方であり、「クラウドの利用方法＝スモールスタート」としても過言ではないほど、今や一般的に認識されています。しかしクラウドでしかできない方法として、ラージスタートという考え方もあるのでは？と最近、考えています。


キャンペーンサイトなどでは、「とにかくトラフィックは来るはずだが、どれだけかはわからない。そして絶対、サイトをダウンさせてはいけない」という無茶な要件(?)をいただくことが多々あります。

このような場合、「とにかく、これだけあれば大丈夫だろう」と思われるスペックと数量のサーバを用意しておき、状況を見ながら、どんどん数量を減らしたり、スペックを落としていく方法をとるようにしています。たとえ初期状態が大幅に

過剰なスペックでも、その状態は長くても1日程度しか維持されないで、コストを考えても実はいしたことはありません。むしろ事前に正確にサイジングするための人件費のほうが高くついてしまうはずです。

### まとめ

2013年、ITエンジニアをとりまく環境は、「クラウドを使ってみる」フェーズから「クラウドを使いこなす」フェーズに遷移していくものと考えています。そして、当然、注目される技術／スキル／ノウハウも、それに伴うものとなっていくはずです。

クラウド脳を身につけるために、クラウドデザインパターン(CDP)、とくにクラウドアーキテクティング原則をぜひ、年初に確認していただけると幸いです！(By Ninja of Three) 



## 海外から見て、世界標準で2013年以降、主流になりそうな情報技術は？

谷本 真由美 TANIMOTO Mayumi ● Twitter@May\_roma

### 893万円のオナホを誰が買うのか？

IT業界にいますと、「今後標準になりそうな技術は」ということを考えるときに、ついつい「これはイケテルから世界でドカンと流行する。絶対に標準になるな」と考えがちなのですが、技術を使うのはあくまで人間でございまして、何をどうやって使っていくか、何が便利だ、何が必要だ、を決めるのは人間様でございまして。その決定を左右するのは「人間様がおかれた環境」、つまり「世相」でございましょう。技術者は「世相」をみないで「どれが行けそうか」と考えがちであります。しかし、「世相」を無視する、ということは、一個893万円のオナホを開発してニートに売りつけようとする、ようなものでございまして(その前に薄給社畜の貴様はそんな物を買うのか、でございまして)。

さて2013年以後どんな技術が日本の外でイケイケになりそうかは、世相から見なければならな

いと筆者は考えます。注目すべきは「金がない」と「マイクロ」の2つのキーワードです。

### BYODの目的はコストカット

まず「金がない」のほうであります。欧州だけではなくどこもお金がありません。会社も個人も余分な経費を削減できるような技術を使ったサービスなり製品を使っていく傾向が益々強くなるだろうなあとということです。そうなると、なるべくコスト削減に役に立ちそうな技術や製品が注目を浴びます。

ここロンドンでも最近熱い物の1つは「Bring Your Own Device(BYOD)」です。別名「自分で勝手にデバイス買って仕事してくれ、壊れたらお前がどうにかしろ、うるせえ、情シスに電話してくんじゃねえ！ 迷惑メールが来た？ 知らねえYO! ユーザエラーだエラー！ 要するにバカはお前！」でございまして。自分で勝手に買って来て



使ってくれたらサポートの工数は減りますし、在庫管理もテストも要りませんので、情シス的には非常に助かるうえ、ユーザもお気に入りデバイスでハッピー。まさにウインウインのソリューション(たまには業界用語を使いましょう)です。イギリスや大陸欧州ではユーザー一人当たり年いくらかという予算を会社が出しまして「この予算内で勝手に好きな物を買え。管理はお前の責任」というダイナミックなことをやっている会社が結構あります。法律事務所や金融などかなり機密性の高い情報を扱う会社でもこんなです。このほうがサポートの工数かかりませんし、在庫を保存する場所も要りませんので(地価が高いので倉庫確保だって大変です)集中管理するより安いわけです。BYODを導入するにあたって、セキュリティが懸念点となりますが、仮想環境上で、BYODを提供できるソリューションが人気です。ビジネスの「消費者化」を止めることはできませんので、2013年以後もますます需要が高まるんじゃないでしょうか。



### 人間の賃金もコストダウン

「金がない」。会社や小規模会社はなんとか人件費を抑制したいと考えています。しかし仕事はしなければならぬ。というわけで、クラウドソーシングサービスに仕事を投げる会社や個人が増えています。FoapやPeoplePerHourのように、個人や個人、個人と会社のマッチングを提供する技術やサービスはますます注目を集めるでしょう。会社が仕事を外注したいという他に、失業してフリーランスになったり、個人で小規模ビジネスを始める人も増えているので、「仕事を受注したい」という人の利用も増えています。このようなサイトをのぞくと、スペイン、ポルトガル、ギリシャ、東欧、ロシアなど、「ああ、仕事がなさそうだなあ」という地域の人はずらっとでできます。アフリカ、インド、中東など従来は他の国から仕事を受けることがちょっと難しかった国の人も増えています。



### 金がなくても……いや金だ！

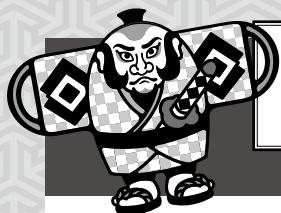
「マイクロ」に関して注目を集めるのは「マイク

ロペイメント」です。これは携帯やタブレットなどのデバイスに読み取り器などをくっつけてお金を払える様にしたたり、ブログやTwitterをやっている人に小額のお金を払えるようにするソリューションですが、イギリスだけではなく欧州大陸ではモバイルデバイス向けのペイメントソリューションが注目を集めています。アメリカのSquareに影響を受けているのは言うまでもないのですが、欧州はアメリカに比べてこのマイクロペイメントの需要が結構あるんじゃないか、という声があります。欧州はアメリカや日本に比べ、個人経営の商店や市場で買い物する機会が少なくありません。大都会ロンドンでさえもそこいら中に屋台や市場があり、費用は現金でやり取りされています。なんで屋台や市場がこんなにあるか、というと、大手の店で買うよりも安くて質が良いうえ、失業する人や移民が商売を始めるからなのであります。

日本より福祉制度が良さげな感じですが、自分で何かをやろう、という人が実は少なくないんです。モバイルペイメントのSumUpは多国籍展開を拡大中でありますし、スウェーデン発のiZettleは欧州のSquareと呼ばれております。イギリスのEmuはクラウドベースのペイメントソリューションを提供しており、売り主がサイトから登録するとさまざまなデバイスからクレジットカードによる支払いを受け取ることが可能です。同社は2012年9月に営業開始をしましたが、なんと毎日ユーザが倍増しているとのことです(<http://p.tl/RUAJ>)。こちらも欧州中心に多国籍展開しています。

2013年の抱負を語るような記事なのに、いきなり「金がないことに注目しろ」というのは年始から気が萎えそうではありますが、景気が悪いなりには悪いなりに儲かりそうな分野がある、ということは気に留めておきましょう(要するに、グローバルイゼーション=英語公用語化だぜ!——と吠える前に、もうちょっと本質的なことをみようぜ、デバイスは直せよ、ということです)。**SD**

## ネットワーク技術はどのようなのか

データセンターの未来とは、  
目指す新しいサービスとは?

さくらインターネット株式会社 代表取締役社長 田中 邦裕 TANAKA Kunihiko



## 自然エネルギー活用型データセンターへ

データセンターというと、耐震構造の建物に大容量の電源と回線が引き込まれ、発電機や蓄電池、大型の空調、消火設備の整った、安定性・信頼性を重視した設備だということは知られています。しかし、最近では上記に加えて「省エネルギー」や「運用の効率化」といった観点が新たに注目され、トレンドは大きく変わり始めています。

省エネルギーという観点では、冷涼な外気や水などの自然エネルギーを使って、サーバの冷却に使用する電力を低減させる試みが広がり、それらが利用できる地域にデータセンターを立地させるということが増えてきました。また、風力や水力、太陽光などを活用した発電ができる地域にデータセンターを作ることで、送電のロス低減させるという取り組みもみられます。つまり、冷涼かつ、発電所が近隣にある地域にデータセンターを作り、そこから光ファイバーで大都市へ情報を送信したほうが、電力を大都市のデータセンターへ送るよりも効率的だということです。

また、発電所とデータセンターを近接地に建設できるなら、変電所で電圧を上げ下げする必要がなくなるので、発電、送電、受電、給電のすべてを直流化することも可能です。

CPUやメモリといったサーバ部品も、蓄電池やLED照明といったデータセンター機器も、すべて直流で動作しており、発電からデータセンター、サーバまで全体最適化できれば、エジソン時代から続いた交流の隆盛は終わるかもしれません。

このようなアプローチを組み合わせるこ

とによって、特に補助金などがなくても自然エネルギーが旧来型の発電コストと変わらなくなると予想しています。さらに、アプリケーション、サーバ、ラック、データセンター、外気温などのすべてを一体的に管理し、省エネルギー化することもさらに進むと考えられます。



## 排熱制限からの解放

最近では熱解析を行うソフトもハードも非常に安価になったため、もっとも効率のよい機器配置を運用前に確認しながら設計を行えるようになり、稼働開始後に扇風機などで局所冷却するようなこともなくなるでしょう。また、排熱に関する機器の全体最適化もさらに進み、ファンを必要以上に回転させ続けるような状況も改善されます(表1)。

現状では、サーバのファンはサーバ内の温度に合わせて回転し、ラックのファンはラック内の温度に、データセンターの空調はデータセンターの温度にしか連携していません(図1)。

このような状況に対して、CPUやメモリなどの排熱量に合わせてすべての要素をコントロールで

▼表1 環境温度32℃の時の、条件別消費電力実測値

省エネ設定	オフ	省エネ	カスタム
CPU 負荷時	280W	264W	316W
MEM 負荷時	192W	184W	192W
HDD 負荷時	216W	184W	232W
ALL 負荷時	312W	304W	352W
OS 負荷時	304W	292W	340W
OS IDLE 時	184W	160W	132W
平均値	260.8W	245.6W	286.4W
unixbench	7526.2	7253.6	6982.9

できれば、必要最小限の電力で排熱ができることになります。ちなみに、最近のCPUは温度管理が非常に高度化しており、CPUの使用率などによって消費電力が動的に制御されるようになったため、トータルで最も効率のよいCPU使用率になるよう、アプリケーション側が対応させる時代になるでしょう(表2)。たとえば、データセンターの排熱量が増えすぎて効率的ではないときなどは、稼働率が高いインスタンスをほかのデータセンターへ自動でマイグレーションされるという時代になるかもしれません。

### 土地の制約からも自由に

運用の効率化という観点では、自動化、お客様のセルフサービスの実現がポイントになります。さくらインターネットでは専用サーバと呼ばれる、物理サーバのレンタルサービスも行っていますが、OSインストールや開通作業などはすべて自動化され、再起動やネットワークの構成、画面&キーボード操作もお客様自身でコンパネ(IPMI)から行えるため、現地ではラックマウント、故障交換、撤去しか人が介在することはありません。

クラウドでも、自動化、セルフサービスが実現されているように、データセンターにおいても規模に応じて人員が増えない仕組みを構築したうえで、データセンター自体を巨大化させていくというのが、運用の効率化への道筋だと考えています。今ではコンテナをデータセンター化するというケースも増えており、工場でサーバをマウントして、コンテナのまま設置場所へ移動し、電源と通信回線をつなぐだけでデータセンターの開設完了という事例も目にするようになりました。これなら、日本にコンテナデータセンターを設置した後で、景気が悪くなれば上海へ移設し、世界を転々

▼図1 石狩データセンターでのVPSサーバラックの温度分布(上方に排気されている)



と移動するデータセンターも考えられます。

キーワードは全体最適化にあると考えています。

ITの世界は、部分最適化を進めて、その接点は標準化という名のもとでオーバーヘッドが生まれる状態にあります。その接点をできるだけなくしていき、無駄をなくすという動きが進んでいます。今後、データセンターは建物からアプリケーションに至るまで全体最適化が行われ、省エネ化&運用の効率化が進み、利用者はクラウド型もしくはホスティング型で、より安価で高性能なコンピューティングリソースを共同利用するといった時代になるでしょう。SD

▼表2 メーカー公称の、温度とCPU負荷による消費電力相関表

環境温度[℃]	CPU 負荷 100%	CPU 負荷 50%
25	150W	130W
28	164W	130W
31	170W	130W
35	190W	130W



## ネットワーク・インフラエンジニアは、どんな技術を身につけておくべきか？

データホテル CTA 伊勢 幸一 ISE Kouichi TwitterID:@ibuchou

### 🎯 ネットワークとインフラエンジニアの乖離

2000年以後、ネットワークエンジニアとインフラエンジニアの乖離が顕著化してきたように見受けられます。ネットワークエンジニアはルータやスイッチなどのネットワークノードとサーバのNICポートまでが自分たちの担当範囲であるとし、インフラエンジニアはDBMSやキャッシュ、キューシステムなど、アプリケーションのバックエンドウェアからサーバOSまでを担当分野としてきました。サーバにIPアドレスとデフォルトルートさえ設定すれば何の問題もなくネットワーク上で相互接続できると信じて疑いません。おそらくMACアドレスやVLAN-IDが重複しているかどうかなど想定範囲外です。同じようにネットワークエンジニアもNICがリンクアップさえしていればよく、そのNICに対しアプリがどのような頻度でアクセスするのか、どれくらいのデータバッファを一度に読み書きするのかまでを気にはしていません。

従来、マルチベンダでの相互接続性、システムの性能向上、そしてスケールアウトを図るためにシステムを数階層に分割し、各層が独立して高機能化を図ることで現在のネットワークシステムが発展してきました。そのため、このネットワークとインフラの乖離状態は自然な流れだったのでしょう。

しかし今、サーバOSだけではなく、ストレージやネットワークまでが仮想化される中、SDNやフェデレーションといったシステム全体を統合的にコントロールする技術が必要とされつつあり、分割分離していた各層をふたたび連携協調して運用しなければならない状況に追い込まれています。

### 🎯 コードが書けるエンジニア・書けないエンジニア

ネットワークはインフラの一部となり、インフラもネットワークの一部として運用しなければな

らないのです。さらにネットワークを含むインフラエンジニアには2種類あります。コードを書くエンジニアと書かないエンジニアです。しかしSDNやフェデレーションを運用するには個別のシステムごとに管理運用システムを独自に開発しなければなりません。それも外注受託開発のように仕様があらかじめ決定しているのではなく、運用し続けることで初めてさまざまな機能やコンポーネントが決定するため、開発と運用を分離して進めることができません。自動的に運用エンジニアは開発も手がけ、開発エンジニアも運用を実施することになるでしょう。なぜなら運用していない者に運用ツールが書けるはずがないからです。運用ツールは運用しているエンジニアにしか作れません。

### 🎯 スペシャリストとジェネラリスト

従来担当範囲外だった「開発」というプロセスがシステム運用のワークフローの中に組み込まれることになります。同時に開発側もファブリックがSTPで動いているのかTRILLで動いているのか、スイッチ間のポート帯域やアグリゲーション状態、VLANタグID番号、VLAN多重化数などを把握する必要が出てくるのです。つまり、スペシャリストではなくジェネラリストの存在が必要となるわけですが、はたして現在そのように従来分離されていた役割をある一定以上のレベルで同時に実施できるエンジニアがいるのでしょうか？

スペシャリストとジェネラリストのどちらが良いのかという議論は常にあり、過去10年間ほどはスペシャリストに軍配が上がっていたようです。しかし、それはそれぞれコモディティとイノベティブというマーケットのフェーズ次第ではないかと筆者は思います。コモディティフェーズの場合、より高い競争力を持つには限界を超えた品質や性能が求められるため、スペシャリストの重要性が高くなります。しかしイノベティブフェー

ズにおいては従来の常識や慣習が大きく破綻する可能性が高く、あらゆる分野と技術に対する知見や経験がなければその混沌とした中から最適解を見出すことは難しいでしょう。

### ノーリミット、ノーフューチャー

今、そして2013年に求められるエンジニアとは、自らの担当分野や範疇、レベルなどに自ら制

限を設けることなく、あらゆる技術に興味を持ち、そして手足を動かして調査検証し、さまざまな引き出しを高いレベルで習得することを苦としないエンジニアなのではないでしょうか。

少なくとも、自らの可能性に限界を設定してしまつては、コモディティとイノベティブが繰り返すこの業界で長く生き抜くことは非常に難しいと思います。SD



## OpenFlowとファブリック、2013年にネットワーク技術はどうなる？

ブロードコミュニケーションズシステムズ 小宮 崇博 KOMIYA Takahiro ● TwitterID:@kommy\_

### 今日のネットワーク技術

2012年に注目されたネットワーク技術に、イーサネットファブリックとOpenFlow技術があります。これらの技術にどのような特徴があるのかを説明することで、2013年の動向を俯瞰します。

企業ITシステムのトレンドに、“DevOps”という言葉があります。これは生産性、品質向上のために、開発と運用を一体化しようというものです。DevOpsの推進においてネットワークに求められる要件は「シンプルであれ」ということでしょう。ビジネスに必要な性能が得られるというのは重要です。また、障害や性能劣化に対応できる運用側のノウハウも重要です。

今日現在、ファブリックはこれらの課題の多くの部分に対して解決方法を提示しています。スパンニングツリープロトコルを使用しないネットワークを構成することで、シンプルな広帯域化と高可用性の実現をしています。これはスイッチのIDを決めるだけで構成できるため、シンプルに導入することができます。また、サーバ仮想化を導入すると必然的に導入される、サーバ内の仮想L2スイッチと物理ネットワーク間のVLAN運用などのオペレーションを自動化することで、サーバとネットワーク運用における、人間系連携をシンプルにできます。加えて、ITシステム全体としてとらえると、データストアへのアクセスネットワー

クである広義のストレージエリアネットワークも取り扱う必要がありますが、ファブリックであれば単一の基盤でLAN/SANを現実的に取り扱えます(図1)。

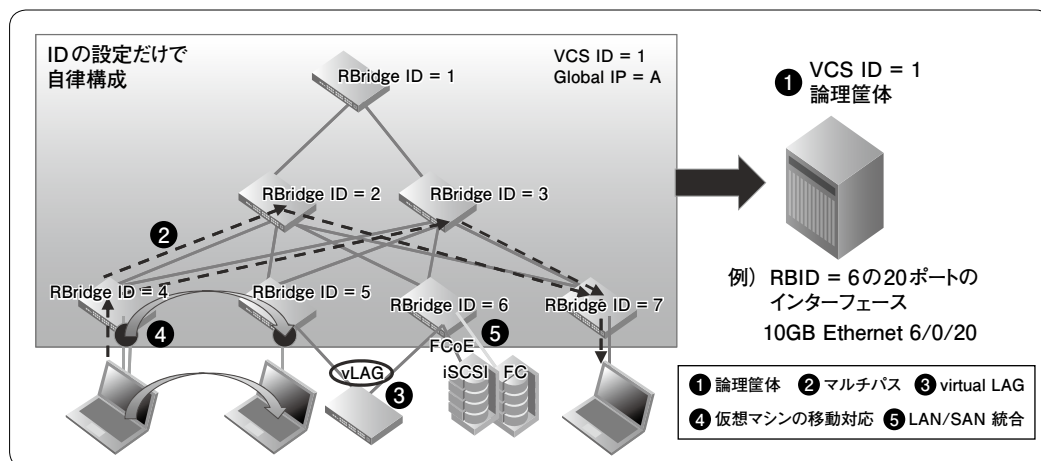
アプリケーション開発者が自らネットワークを設計、運用できるようにするために、ファブリックを導入し、結果としてIT導入、構築費用を20%低減に成功した国内事例も出てきました。ファブリックの利点は、レイヤ2、3ネットワークをすべて入れ替えることなく、レイヤ2のスイッチ1台から利用開始できるところでしょう。

OpenFlowでも、もちろん、ファブリックと同じことができるようになります。これはファブリックの行っている各種サービスをOpenFlowコントローラで実現することで可能になります。この例でもわかるとおり、OpenFlowの魅力である「フローベースのトラフィック制御」は、トラフィックを「どのように制御するか」を司るコントローラがあって、初めてビジネス的に有効になります。実はこのコントローラの市場がまだ黎明期なのです。

### 2013年への期待

2013年には前述の2つの技術、ファブリックとOpenFlowに進展があると期待できます。まず、ファブリックにはより大きなスケラビリティと

▼図1 BrocadeのEthernet Fabric

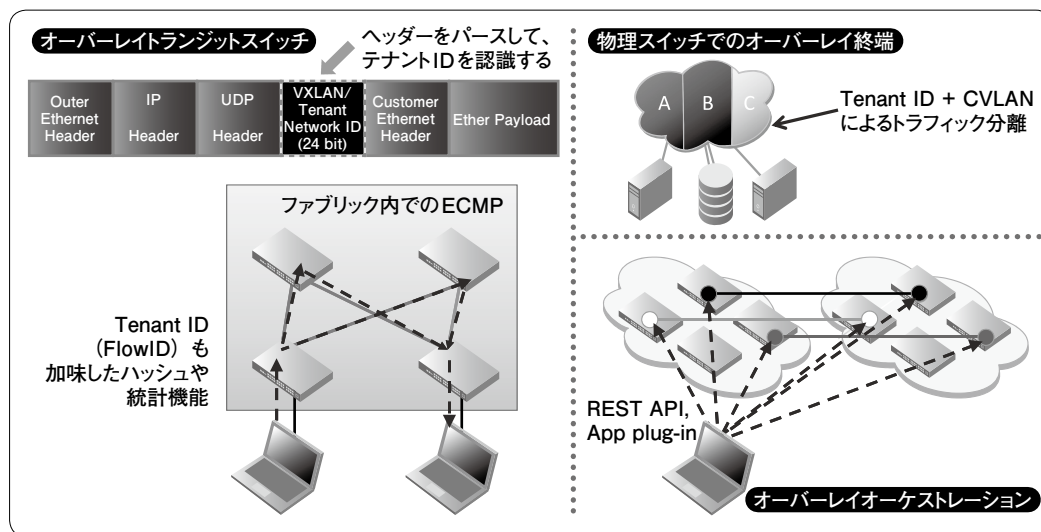


レイヤ3などの各種機能性が追加されていくことになります。OpenFlowについてはさまざまなコントローラベンダーがビジネス側から要求されるネットワークの機能性を実証していく段階に入らずです。また、新しいネットワーク技術のトレンドにオーバーレイ技術があります。大規模データセンターネットワークにおいてマルチテナント化を実現するためには、テナント間のトラフィックのセキュアな分離が必須要件になりますが、一般的によく利用されているVLANではただが4,000個のアドレス空間しかありません。このため、オーバーレイを利用したスケーラビリティの

向上が議論されています。

VMware社のVxLANやMicrosoft社のNVGRE技術などはオーバーレイの代表的な技術になります。オーバーレイを仮想スイッチで実現する手段はすでに一部で実現されていますが、汎用機やUNIXサーバとの接続性や広義のSANストレージを接続するためには物理スイッチでオーバーレイを終端する必要があります。これらの技術が2013年に出てくることでしょう(図2)。これらの技術により、より投資効果が高く、ビジネスリスクの低いITサービス、クラウドサービスなどが実現されることが期待されます。SD

▼図2 2013年に出てくると期待される各種オーバーレイ対応機能



# Slerは生き残るか？

## ユーザ企業から見た、 生き残るSlerの条件

カブドットコム証券 社長付IT戦略担当 谷口 有近 TANIGUCHI Arichika ● TwitterID: @arichika

### ● 新陳代謝の時代には苦渋の選択も

さらなる低コスト化と尋常ではないスピードだけでなく、いままでどおりの品質に加えさらにぶっ飛んだ提案までもが求められる近今、ユーザ企業にも変革が求められています。Sler同様に、イノベーションの渴望とスピードへの要求は強くなっています。しかし、世の地位ある方々が発する「イノベーションを目指せ」というメッセージ、私はこの言葉に空虚さを日々感じているのだといえ、本誌をお読みの皆さんであれば、説明するまでもなく同意していただける方もいらっしゃると思います。

### ● 変化なくしてイノベーションなし

開発体制や管理体制の変化なくして何がイノベーションだと、そう感じて当然だと私は考えています。仮に革新的なアイデアが存在したとしましょう。そして、迅速に具現化し世に問い、さらに改善をフィードバックしたいがためにアジャイル／スクラムを採用したと。結果、変化していく仕様に追従できるほど承認体制がなく、業務知識と実装技術の溝を「ほどほどに」埋めるセンスを持つ人材が圧倒的に枯渇していることを再確認し、そして、過度に分業化された世界に絶望するわけです。

しかしながら、ユーザ企業の未来について私はかなり楽観視しています。今や、技術、そして“ネット”を理解していなければ、適切なサービス設計やマーケティングの遂行はたいへんに困難な状態です。この変化をユーザ企業は十分理解していて、結果としてインハウス化、そして選択的アウトソースの実践につながっています。

### ● 生き残ってほしいSler像

優秀な技術者の取り扱いという意味で、今やユーザ企業とSlerは競合状態です。また、SaaS/PaaSクラウド基盤の成熟も進んでいます。本当にSlerの仕事はなくなる一方でしょうか。その可能性を否定できるだけの根拠を私は残念ながら持たないのですが、このようなSlerにはぜひとも生き残ってほしいという想いを、3つのメッセージにまとめてみました。

#### ● バズワードが生まれた背景を理解している

すべての技術には、それが生み出されて流行るだけの事情と背景、つまり改善したい事案が存在しています。その事案の形成には、雇用習慣などの文化的な背景が強く反映していることもあり、その技術の採用が日本国内においてそれほど効果を発揮しない場合があります。背景を理解した上で提案ができるか否か。理解しているからこそその独自利用での強みを展開していけるかどうかは、重要な観点です。

#### ● 見た目と操作感到こざわりがある

UI/UX分野は特に伸び盛り。それを検討した分だけの成果は得られるけれども確実な正解はなく、他分野同様深く、たいへんにチャレンジングな楽しい分野です。消費体験をエンターテインメントとして解釈し提供することは大前提であり、見た目と操作感へのこざわりは、識別の1つになると考えています。卑近な例ですが、#FFFFFFか#FAFAFAか、この差はバックライトの輝度により決定的な差を生みます。そのこざわりを理解しスクラム

の中で実装していけるか否か。

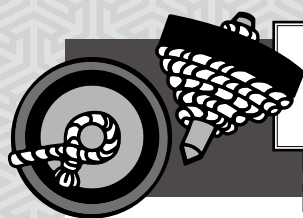
#### ●技術領域単位で部門や担当の壁がない

アプリケーションやサービスの振る舞いを最適化するには、コードのレイヤだけでなくインフラのレイヤでも、さらなる改良が必須です。専門技術の範囲にのみ理解を深めていては、全体最適はいよいよ難しい状態です。まずは自身が得意とする専門分野の隣接技術の今を知ることが、たいへんに重要だと考えています。

#### 🎯 個人と戦う時代のSIerへの期待

優秀な個人が集まり、世に問うアプリを提供す

ることができる時代です。つまり、同業他社ではなく個人と比較される時代です。今後、優秀な技術者がユーザ企業で手腕を発揮することになると私は予想していますが、その際、SIerは組織として個人と比較し、圧倒的に勝る技術領域があることは、最低限の個性でしょう。その上で、デザインが優れている、HPCや統計解析に強い、柔軟性に自信がある、満を持して公開したスマホアプリに星1つの評価がついた際のショックを知っている、そんな個性な+αを持つSIerであれば、これからの時代をともに創っていきいたいという思いを抱くことができるでしょう。SD



## SIerが今後生き残ることができるのか

湯本 堅隆 YUMOTO Michitaka TwitterID : @gozasenpai

#### 🎯 人月の毒薬

ブログで何度も申し上げていることですが、SIerが生き残っていくのが難しい最大のポイントは「人月を積み上げて価格設定をすると、クラウドなどを活用したソリューションに比べて相当に割高になってしまうこと」にあります。人月見積もりで価格付けを行ってしまうと、エンジニアの稼働がそのまま利益に直結してしまいます。これは誰でも売り上げを上げることができるというメリットがある一方で、どう頑張っても1人が1人以上稼いだことができずコスト高になるというデメリットも持っています。このデメリットが際立ってしまうのがSaaSビジネスです。仮に提供できるシステムがほぼ同一とすれば、人月見積もりで「サインアップしたら使えますよ、カスタマイズも月々の料金の中で行える」というビジネスに勝つのは極めて困難です。

ただ、大型案件の場合はクラウドを担いでSIをやるってことはあまり現実的ではありません。既存システムの連携が必ずと言って良いほど、絡んできますのでそこだけクラウドに載せてもメリットが薄いという局面が多いのです。さまざまな

パッチとのやりとりもたいへんです。そのあたりをふまえますと、上場している規模の大手SIerはある程度安定すると考えています。よほどのことがない限り、下げ幅には限界があるでしょう。

しかし、上記の流れは大きなトレンドになるうとしており、かつ日本経済の景気の悪化というダブルパンチを受けた現状を鑑みると、中小規模の受託開発がメインのSIerはどうやって生き残っていけば良いのか非常に悩ましいです。下請け構造というのは景気によって著しく必要人数が左右されてしまいます。大手にも余裕がありません。下請けやSES(システムエンジニアリングサービス)のような、悪く言ってしまうと人を担保にするビジネスで息継ぎをしながら泳いでいくことも、限界がやってきました。自分で泳げる力を持たなければなりません。ゆえに、中小規模のSIerは愚直なまでに元請けにこだわり直接お客様とSI案件を行うこと以外生き残る道はない、と筆者は見ています。

#### 🎯 挑戦の時代へ

2013年は中小規模のSI会社が、元請けのSI案



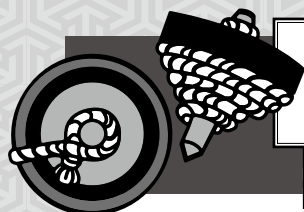
件を獲得するためにさまざまなアプローチやビジネスモデルを生み出してさまざまな挑戦を行う始まりの年になると考えます。

今のところ最も顕著に見られる傾向としては、システム構築費用を限りなく下げる代わりに何かしら月額で料金をいだいていく、という方式です。もうシステムを構築することにお金をかける顧客は減っていますので、SaaS勢に取って代わられないために「システム構築は限りなく低コストにすることで顧客とパートナーシップを築いて、キチンとした運用を行いレベニューシェアなどをもらいながら、中長期的にお付き合いをしていく」という戦略に切り替えていく会社が増えて

いるのではないのでしょうか。

クラウドが当たり前になったとしても、それを使えるだけのリテラシーやプランがなければ、意味がないわけですから、Slerに求められていることは決して少なくありません。案件はあります。ただ、それをデザインできる能力が欠けているので、何をどのようにどの順番で行うのかというビジネスモデルを変化させれば、生き残っていける要素は多分にあります。

新しい時代を迎えた新しい時代のSlerのカタチを、筆者自身も内製しながら追いかけていきたいですし、中小Slerの奮起に期待しています。SD



## Slerに未来はあるのか？ ——現実を見よ

ノーチラス・テクノロジーズ社長 神林 飛志 KANBAYASHI Takashi

### 🎯 SI屋はなくなる

最近のSIの失敗率は目を覆うものがあります。また、人月商売の限界からも、SI事業の先行き不安が叫ばれています。しかし、残念ながらSI事業それ自体は縮小傾向にはなるものの、ITの有力産業としては残るでしょう。現在のSIマーケットは、一方ではSI屋さんが人月により手取り早く売上を稼ぎたいという事情と、他方ではユーザさんがプロジェクトリスクのヘッジとIT人件費の負担の流動化を狙うという事情の、見事に需給のマッチした市場です。そうは簡単には崩れません。プレーヤーの意図とは関係なく成立している側面が強いです。

よって、なくなりはないSIの中で、どう強引に未来を築くか？ という話ですが……とここまで書いて、実はまあ最初は行儀よく「それなりに未来があるんじゃないのですか？ 頑張ればいいんじゃないですかね？」という話を書いたのですが、(編集者に)思いっきり却下されたので、割と真面目に未来がないという話とか書きます。

### Caution!

そーゆーネガティブなことが読みたくない人は、今、この瞬間に飛ばし読みをお願いします。読むとすぐく不快になると思います。すみません。

### 🎯 なくなってしまった ユーザリスクへの「保険」

まず、ユーザ企業様に於かれましては、すでにノーリスクという時代は過ぎ去ったと自覚すべきです。とりあえず「金を払えばやってくれますか？」とかむしろ払ってやるから言うことを聞け、いや聞いてくれ」というモードはすでに時効に達しています。ちょっと前までは、赤字案件はコンプライアンスで通らないという状況でしたが、今はもっと酷くて多少なりとも技術的に複雑な案件では実行リスクが積み上って、受注稟議が通らないという案件も頻発しています。よってRFP出して、コンペして、最終決定してベンダに発注した段階で「いや、社内が説得できずご辞退でございます」ということすら普通にあります。

一方ではITは企業基盤の背骨ですといつつ、

他方では数字が悪くなるとさくっと予算を削って、いやいやITは自社のコアでないで……とよく言えば機動的、悪く言えば場当たりの施策をとりつつも、最後はベンダに責任を取らせればいいのだ、ベンダ逃がす的なスタンスは、確かに合理的ではあります。ただし、それは「最後はベンダがなんとか作ってくれる」という前提がありました。

その前提は、もう過去のものになりつつあります。ユーザさんが想定している以上に今のITは複雑怪奇になりました。極端な例では、昔の汎用機なら簡単に書くことができた処理を、ベンダさんをお願いしたら、いつの間にやら一システムできあがってしまった。そのうえまともに動かない、という案配です。まったくもって普通にシステムの構築が難儀になってしまっています。技術の進歩は、残念ながら、まともにシステムの構築ができないSI屋を大量に生む結果になってしまっています。ノーリスクというポリシーは砂上の楼閣。現実、板子一枚下は地獄です。

### 🎯 現実が直視できないSI業界

また、SI屋さんに於かれましては、いい加減に人月商売で売上を安易に求めるのはもう限界がありますって自覚しないと、地獄の釜が蓋をあけるどころか、もう茹でカエル状態です。10年ぐらい前の「なんやら大きく失敗するプロジェクトが増えましたね」という警告の時代に早めに手を打っておくべきでしたが、見事にスルーしすぎました。

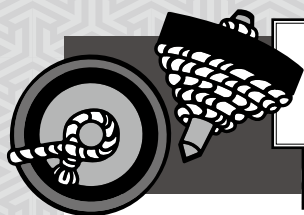
あの当時は、確かに5発に1発ぐらいは赤字プロジェクトで、しかもその1発がそれなりに痛いという感じだったと思いますが、今は油断すると5発の4発が全部赤字という有様です。ここに至っ

た根っこの要因は、技術進歩に対する備えのなさでしょう。技術はどんどん革新していきますが、それが自分にとって有利になるとは限らない、いやそれどころか、そもそもその技術を作り出している産業それ自体にとっても脅威になることは普通にある、ということです。

オープン系の流れからの「エンジニアリングに自由を」の旗印は、想像以上に暴走しました。選択が増え過ぎるとかえって何もできるのは人間の性です。技術の進歩と多様化は、良いことばかりではなく、むしろ圧倒的に品質は下がる可能性があったわけですが、これに対して投資を怠りすぎました。今や業務系のシステムでは、Webの画面を1つ作るのに青色吐息です。トレーニングにしろ、道具立てにしろ、有能な人材の育成にしろ、真剣にやってこなかったツケはもはや返済不能です。程度の差こそあれ、これがどのSI屋さんにとっても、今の現状になってしまっています。

### 🎯 それでも未来はあるのか?

……という背景を持つユーザさんとベンダさんの、とはいえ引くに引けない需給のバランスの一種のデッドロックがSIビジネスです。未来はありますよ。明るいかどうかは別として。明けない夜明けはない? 自分で太陽を作るはめになりますね。ユーザさんもベンダさんも「技術が進歩すればすべてが良くなる」という単純な進歩史観に乗り過ぎました。昨今の技術の進化は保守的なSIビジネスにとっては、プラスになるとは限りません。未来を見つけるには、まずもって「今は真っ暗です。どうしてこうなった?」と自分を見つめ直すところがスタート地点かと。乱筆失礼。SD



## 成功しない方法

仙石 浩明 SENGOKU Hiroaki ● <http://www.gcd.org/sengoku/>

「なぜ成功できたか?」を聞いたがる人って多いですね?

でも、そんなこと聞いてどうするのでしょうか。仮に同じことが実践できたとしても、同じように



成功できるとは誰も思っちゃいないですよ？

同じように成功するのは無理としても、成功者の話には何らかのヒントがあるんじゃないか？  
きっとそういうことを期待しているのだと思いますが、考えてみてほしいのは、成功しなかった人の話は誰も聞こうとしない点。もしかしたら成功しなかった人も成功者と同じことをしていたかも知れませんが、いわゆる「生存者バイアス」ですね。

たとえば、成功できた理由を聞かれて、「自分を信じて、あきらめずに続けたから」などと答えている人をよく見たり聞いたりしますが、こーいう話を真に受けて、素質がない人が自分を信じて突き進んだりしたら、成功確率が上がるどころか、撤退タイミングを逃してドツボにはまってしまうだけです。

まあ、私自身も、「やりたいことをやるのが一番」などと言っていた時期があるので大きなことは言えないのですが、「自分がやりたいことって何だろう？」などと悩み出す人をこれ以上増やさないためにも、ここでは逆に「成功しない方法」について書くことにします。ちなみに「成功しない＝失敗」ではありません。失敗は成功の元ですから、むしろ大いに失敗すべきです。

「成功しない方法」を避けたところで成功する保証は何もありませんが、多くのエンジニアが残念なことにこの「成功しない方法」を忠実に実践し、その必然的な帰結として、どんどん成功から遠ざかってしまっています。近年「エンジニア」という職種が報われない職種になってしまっている1つの原因が、このあたりにあると感じる次第です。

もちろん、いままさに「成功しない方法」を実践している人に対して、このままでは成功からどんどん遠ざかるぞと忠告したところで、そう簡単に方向転換はできないものだと思いますが、もし、いま「このまま進んでいいのだろうか？」と思っているなら、少しばかり耳を傾けていただけると幸いです。

誌面が限られているので、いきなり結論からいきます。

「成功しない方法」とは、「選択肢を減らすこと」です。

選択肢と言われてもピンとこないかもしれませんが、人生には分岐点がたくさんあります。たとえば八方塞がりに思えるドツボな状況に陥ってしまっている、その後の分岐点で最適な選択さえできれば、ピンチがチャンスになったりします。ただしそれは、分岐点に選択肢の幅が十分にあればの話です。選択肢が多ければ、リカバリーのチャンスがあるということですね。

たとえば、選択肢を減らす典型的な方法が、借金をすることです。多くの人が20～30年もの長期ローンを組んで家を買ったりしますが、転職しなくなったりしたとき、毎月十数万円の返済は重い足枷となるでしょう。借金がなければ思い切って生活を変える選択ができたかもしれないのに、借金があるばかりに渋々現状維持を続け、ついにはリストラの憂き目に、なんてのはよく聞く話です。

でも、借金よりもっと選択肢を減らす方法があります。それは、自分の気持ちを優先すること。よくいますよね、「これは自分の仕事じゃない」とか言う人(>\_<)。その仕事が多様な仕事か十分に知り尽くして、「その仕事をする」という選択肢を除外することが合理的であるなら構わないんですが、多くの人はどんな仕事か知りもしないくせに、「自分の気持ち」だけで判断してしまいます。もしかしたら、やったことがないその仕事に挑戦することが、自分の人生の一大転機になるかもしれないのに。

あるいは、「自分がやりたいことって何だろう？」などと悩む人がいます。「やりたいことを仕事にすべき」などと(私を含めて orz)多くの人が言っていますが、誰だって最初からそれが「やりたいこと」だったわけではないんです。なにかのはずみに始めたことが、やってるうちに(少し)好きになり、好きでやるからどんどん熟達し、他の人よりうまくできると余計に好きになり、という好循環にはまって「やりたいことが仕事になる」んです。つまり、最初は好きでなくても自分の気持ちを無視してやってみなきゃ選択肢は増えません。それなのに、悩んでるばかりで始めないから、だんだん歳をとって、やってみるチャンスすら失われてしま

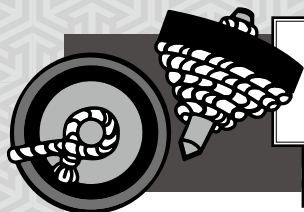
います。

また、多くの人には批判するのが大好きです。昔は飲み屋で、いまだとブログやSNSで、「上司が悪い」「会社が悪い」「社会が悪い」「政治家が悪い」。なぜ人は批判するのでしょうか？ 社会をよくなるため？ とんでもない、愚痴を言ったって社会は少しも変わりません。誰かを批判すると、その人より自分が偉くなったような気分になれるから、人は批判するのが大好きなのです。

でも、批判によってよくなるのは「自分の気持ち」だけで、自分自身は決してよくなるはなりません。なぜなら、誰も成長するには他人から学ぶほかにないからです。でも、「彼はバカだ」と言ってし

まったら、その「彼」からは学べません。選択肢がまた1つ減るわけですね。本当に学ぶことが一切ないようなバカなら学ばなくてもいいのですが、ほとんどの場合そうではないでしょう。本当は、学ぶところが一切ないような人に対しては、批判する気にもならないはずですよ。

自分の気持ちを優先して選択肢を減らす例は、まだまだたくさんあるのですが、誌面が尽きたのでこの辺で……もしこの手の話に興味があれば、私のブログ（「仙石浩明」で検索！）を参照してください。SD



## 35歳のITエンジニアが今後10年仕事を続けていくのに必要な技能は何か

株式会社ユビキタスエンターテインメント代表取締役兼CEO 清水 亮 SHIMIZU Ryo

### 🎯 プログラミングの基礎体力

エンジニアは、常に新しい技術や時代に適応していかなければなりません。

10年という長いスパンを考えたとき、これさえ学んでおけば間違いのない、というような便利なものは存在しません。そのうえで今後10年、エンジニアとして生きていくために必要なのはプログラミングの基礎体力だと思います。

とくに関数型プログラミングは一度は経験しておいたほうが良いでしょう。その中でもHaskellがお勧めですが、たとえ実際の仕事に使うことがなかったとしても、プログラミングの考え方として異なる世界を知っておくということは、通常のプログラミングにも役立ちます。Haskellが少し難しすぎると感じる場合はJavaScriptでも良いでしょう。JavaScriptも立派な関数型言語です。関数型言語で使用可能なテクニックのほとんどはJavaScriptで使うことができます。

JavaScriptは近年、実用に用いられるようになって来ており、10年の賞味期限があるかどうかはわかりませんが、少なくとも向こう5年はJava

Scriptがフロントエンドの世界では標準的なプログラミング言語になっていくでしょう。この流れは、遠からずさまざまな分野に波及して行くはずです。

### 🎯 メモリ革命の勃発

また、10年のうちには半導体にもさらに革命がおきると予測されています。

次なる革命はメモリ革命で、平たく言えば、メモリが高性能化し、すべてのメモリ空間がフラットになると言われています。ここでいう「すべての」メモリ空間とは、CPU キャッシュ、ヒープメモリはもちろんのこと、ハードディスクも含めたすべての環境でメモリアクセスが等価になるということです。

するとハードディスクのような装置は事実上存在しないことになり、しかもすべてのメモリ空間が永続的に保存される世界では、データの「セーブ」という概念が変化します。

また、エンドユーザの扱うコンピュータも、今のような独立したマシンが複数平行して動作する



世界から、複数のPCやスマートフォンが事実上1つのコンピュータとして動作するようなパラダイムシフトが起きるのではないかと考えています。

### 🎯 最先端を体験しておく

こうしたとき、従来のプログラミングパターンがそのまま通用するケースは減って来ることが考えられます。アーキテクチャの根本的な前提が崩れる可能性があるからです。10年前に高速だった最適化手法が現代では色あせて使えなくなってしまっている例は少なくありません。10年後も現在の高速化手法やデザインパターンがそのまま通用するとは考えにくいと言えます。ただし、デザインパターンの勉強が無駄になるわけではありません。この次に現れる技術やベストプラクティスは、必ず今の技術をもとにしています。

ある日突然、大ジャンプするわけではありません。重要なのは、今すぐに自分の仕事に役立たなくても、最先端と言われている技術や技法は一通り見ておくということです。

筆者の過去の経験から言えば、15年前、ゲーム業界はC言語を軽視していました。高級言語によるプログラミングなんかナノ秒単位のゲーム機の世界には関係ないと思われていたのです。ところがゲーム機が高性能化し、より複雑な問題を扱うことができるようになるとアセンブリ言語によるプログラミングはすぐに限界に達しました。C言語への移行を軽視していたベテラン達はすぐに時代に乗り遅れ、職を失いました。しかしC言語は突然現れたものではありません。もともと存在していたものがダウンサイズされたに過ぎないのです。

SD

技術評論社



Software Design編集部 編  
B5判 / 168ページ  
定価1,659円(本体1,580円)  
ISBN 978-4-7741-5335-3

大好評  
発売中!

こんな方に  
おすすめ

・スマートフォンのソフトウェア開発者(プログラマ、デザイナー)  
・スマートフォン対応を考えるWebデザイナー



ユーザから支持されるスマートフォンアプリ開発のために「デザインする力」がより一層求められている開発現場。悩めるエンジニアとデザイナーがともに力を発揮するためのヒントが満載です!

第1特集 どうしてデザインと開発は両立できないのか?

第2特集 スマホ開発者がUnityを理解しておくべき理由

その他 Web+ネイティブでスピード・コスト・メンテに効くアプリ開発 / Windows Phoneアプリ開発入門 / PlayStation Mobileアプリ開発 / 仮想化技術でスマホ&タブレットを業務に / 既存のPCサイトをスマホ用に変換! / Webブラウザでクロス開発できるappMobi ほか

# Win/Mac/LinuxアプリのUI記述が手軽にできる！

## クロス開発環境『Qt Quick』入門



折戸 孝行 ORITO Takayuki

Twitter @loriAYANE URL <http://relog.xii.jp/>

Qt(キュート)という開発環境をご存じでしょうか？ クロスプラットフォームアプリケーションフレームワークとして、世界で数多くのソフトウェア開発に採用されています。本稿はGUI開発をやすくするためにそのQtから派生した『Qt Quick』を使ってサンプルアプリを作成します。どれほど簡単にマルチプラットフォーム対応アプリのGUIが記述できるのかを体験してみてください！

### Qt/Qt Quickとは

クロスプラットフォームなフレームワークといえは複数環境へアプリケーション(以下、アプリ)をリリースすることが狙いで使われます。Webサービスの専用クライアントや社内システムのフロントエンドをWindows/Mac/Linux向けに提供するときなどに有効です。Qtを実際に採用したアプリから著名なものをピックアップしてみました。

#### 【Qt採用事例】

- Adobe Photoshop Elements
- Skype
- Google Earth
- Oracle VM VirtualBox
- KDE

これらはC++で書かれたものですが、すばらしいアプリを作成できることがわかります。

また、ユーザへの配布はQtライブラリを同梱して行うので、アプリごとにQtのバージョンを分けることも容易です。たとえば業務システムに合わせるためにWebブラウザやVMを最新にできなくて他の業務に支障をきたすなどといったことにはなりません。

さて、QtはC++ですが、Qt Quickは言語にQML(Qt Meta-Object Language)とJavaScriptを使用しています。QMLとはCSSに似た記法の独自言語で、これでデザイン部分を作成し、JavaScriptでロジックを組みます。そのため、Webサイトの開発経験がある方にはかなりとつきやすいはず。アニメーションなどを使用したインタラクティブなUIの作成が簡単に行えるようになっていますので、Flashやスマートフォンアプリのような動きのあるアプリもお手軽に作成できます。

また、C++を利用してQt Quickを拡張することができるため、C++の経験がある方にもお勧めなフレームワークになっています。

本稿では、200行程度のコードで図1のような、ボタンによる機能のページ切り替えとTwitterのタイムラインの表示をするサンプルを用意しました。スクリーンショットはWindowsでのものとなって

#### Column 01

### Qtの市場

Qtは世界で7千以上の分野、50万人の開発者が利用しています。国内では数百家の実績があり、デジタル家電・組込み機器・医療機器・デスクトップアプリケーションなどの分野で採用されています。最近では車載関係での採用が増えています。

いますが、お手元の環境がMacやLinuxでも同様の操作で進めることができます。

それでは、Qt Quickのおもしろさを体験していただければと思います。

## Qt Quickのしくみ

Qt QuickアプリはC++で作成されたベースアプリ上で動作します(図2)。このベースアプリは開発環境であるQt Creatorが自動で作成するC++コードをビルドして、各OS向けの実行バイナリを作成します。構成イメージからもわかるとおり、Qt QuickからこれまでQtに装備されてきた豊富な機能へアクセスすることも可能です。

## 対応環境

Qt Quickアプリは次の環境に対応しています。

- Windows XP/Vista/7
- Linux/X11
- Mac OS X
- 組込みLinux
- Windows CE
- Solaris
- QNX
- etc.

開発中ですが、AndroidやiOS、Windows 8への対応の動きもあります。Android向けのβ版を著者が試したところ、デスクトップ用に作成したアプリを非常に快適に再利用できました。

## QMLでできること、C++を使うべきこと

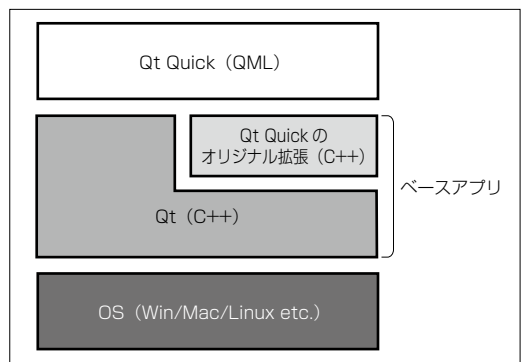
QMLはあくまでGUI開発用なので、なんでもできるわけではありません。JavaScriptで実装できることはひととおり可能ですが、変にこだわっても開発コストがかかったり処理スピードが現実的でなくなる場合があります。また残念なことです、ローカルシステムへの直接的なアクセス方法もほとんどありません。

そのような場合は、C++用に用意された豊富なライブラリを使用してQMLの機能を拡張します。

■ 図1 サンプルアプリ



■ 図2 構成イメージ



## 開発環境の準備

それではここから、手順に従ってQt Quickアプリの開発を行ってみましょう。なお、本稿で解説するサンプルプロジェクトは本誌Webサイトの「本書のサポートページ<sup>注1)</sup>」からダウンロードできます。

Qt Projectのサイトから「Qt libraries 4.8.3」と「Qt Creator 2.6.0」をダウンロードします。それぞれ環境やコンパイラごとにインストーラが用意されているので、お手元の環境にあったものをダウンロードしてください。

Qt Project

<http://qt-project.org/>

注1) <http://sd.gihyo.jp/archive/2013/201301/>

# クロス開発環境 Qt Quick 入門

環境ごとに必要なものが違うので、表1を参考にあらかじめインストールしてください。そのあとに Qt libraries と Qt Creator をインストールします。

Qt Creator を初めて起動したら、Qt libraries と関連付ける設定を行います。メニューを [ツール] → [オプション] → [ビルドと実行] とたどり [Qt バージョン] タブと [キット] タブで設定を行います。

[Qt バージョン] タブでは、[追加] ボタンを押して qmake を指定します (図3)。Windows の標準パスでは、C:\Qt\4.8.3\bin\qmake.exe です。

■ 表1 環境作成準備

確認 OS	必要なもの	備考
Windows	MinGW 4.4、または Visual Studio 2008/2010	—
Mac	Xcode	—
Linux (※)	g++ libglib2.0-dev libSM-dev libxrender-dev libfontconfig1-dev libxext-dev libgl1-mesa-dev libglu-dev	Ubuntu 12.04 の標準状態に必要なもの

※ Linux 版について：インストーラのスタイルの問題でインストールが途中で止まってしまう場合があるので、以下のようにパラメータを付けてインストーラを起動してください。

\$ ./QtSdk-online-linux-x86\_64-v1.2.1.run -style cleanlook

[キット] タブでは、「Desktop (デフォルト)」という項目を選択し、[コンパイラ] と [Qt バージョン] をドロップダウンリストから選択します (図4)。キットの名称はアルファベットに変更することをお勧めします。ビルドパスに使用されるので、コンパイラが日本語パスに対応できないとビルドに失敗するためです。

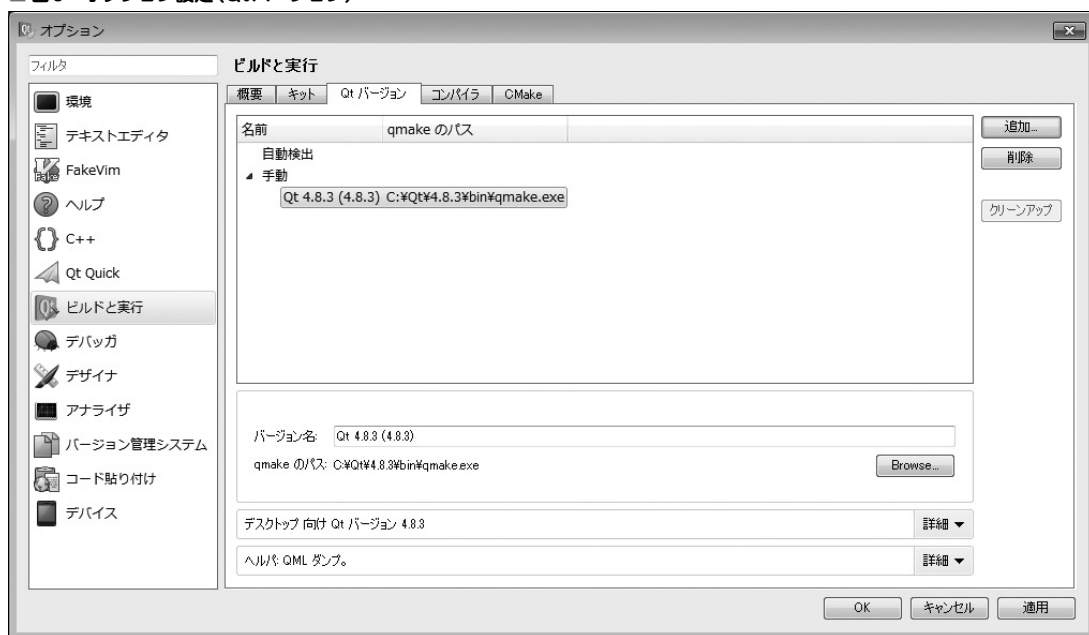
## ボタンで学ぶ基礎

Qt Quick は標準的なコンポーネントとしてのボタンがありません。ですが“クリックして何か動作させる四角形”という機能的な意味では存在しますのでご安心ください。デザインされたボタンがないという点では Flash と同じです。

それでは、少し今ふうなデザインのボタンを作りながら Qt Quick の基礎を学びます。次の項目の順番に解説します。

- ① プロジェクトの作成
- ② レイアウト
- ③ マウス入力
- ④ プロパティバインディング

■ 図3 オプション設定 (Qt バージョン)



■表2 プロジェクトの種類

名称	説明
Qt Quickアプリケーション	C++のベースアプリを含めたフルパッケージ作成用(ビルトイン要素と書かれた新規作成用と既存のQMLを使用する2種類あり)
Qt Quick UI	QMLの部品作成やテスト、サンプルコード作成用。開発環境が用意するビューアで動作確認
カスタムQML拡張プラグイン	オリジナルのQMLエレメント(機能)作成用

⑤状態管理

⑥機能のカプセル化

⑦プロパティ

⑧シグナル

⑨アニメーション



## ①プロジェクトの作成

Qt Creatorを起動し、[ファイル]→[ファイル／プロジェクトの新規作成]とメニューをたどります(図5)。

Qt Quick用にトータル4種類のプロジェクトテンプレートが用意されています(表2)。ここでは「Qt Quick1アプリケーション(ビルトイン要素)」を選択し、最終的に配布できるプロジェクトを新規で作成します。

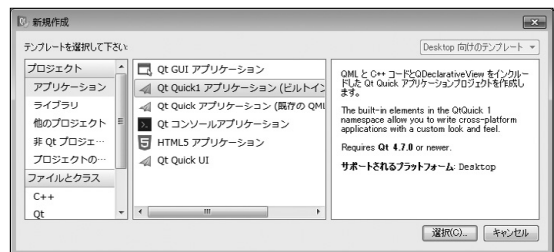
図6のようにプロジェクト名を入力します。任意

で構いませんがここではStartQtQuickとします。

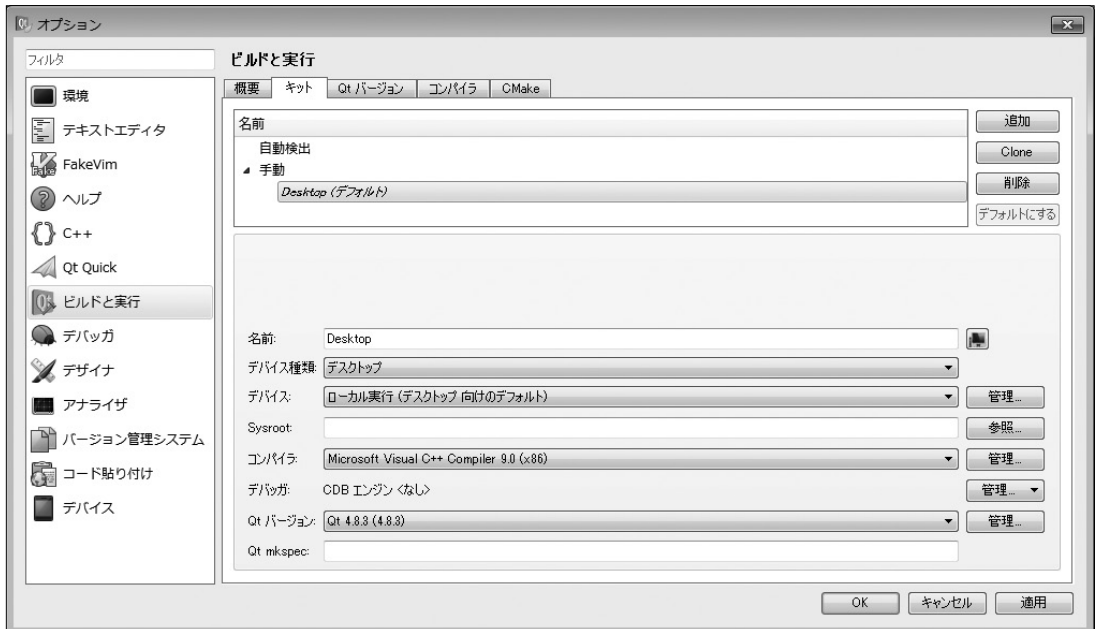
続いてキットの選択をします。デスクトップ向けに先ほどセットアップしたキットを選択します(図7)。DebugとReleaseの項目はビルド用のパスの設定です。ビルド後の実行ファイルが別のフォルダにコピーされるので、プロジェクトが煩雑にならずにすむのでチェックします。

最後はプロジェクト管理について設定します。GitやSubversionなどを使用している場合に選択しま

■図5 プロジェクト作成開始



■図4 オプション設定(キット)



すが、本稿ではそのまま完了します。

完了するとファイル名main.qmlがテンプレートとして作成されます。いわゆるHello Worldです。Qt Creator左下にある緑の三角印の実行ボタンもしくは`Ctrl + R`で実行してみましょう(図8)。このサンプルはクリックするとアプリを終了するボタンの機能が実装されています。

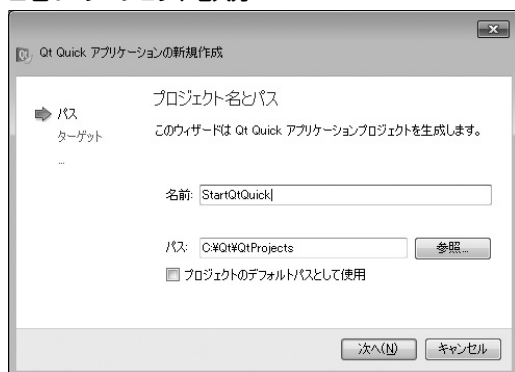
それでは、main.qmlをリスト1の内容に書き換えて、次節から解説を始めましょう。実行すると図9のようになります。

## ②レイアウト

QMLでは起点になるルートエレメント<sup>注2</sup>の中にエレメントを積み重ねていく形でレイアウトを行います。その際の位置決めの方法として次の2種類があります。

- 親エレメントの原点を基準とした相対座標
- 他のエレメントの4辺基準

■ 図6 プロジェクト名入力



■ 図7 キットの選択



注2) エレメントとは、QML内での機能を表す最小単位です。表示機能のあるものといないものがあります。

相対座標はx, yプロパティを使用します。4辺基準はanchors.XXプロパティを使用します。ウィンドウサイズの変更などへ柔軟に対応するために、後者の方法をお勧めします。

anchorsを使用した配置方法の例を図10に示します。親がグレー、子が白の属性を持つとして見てください。

図10のaでは、子は親の4辺の位置に固定されるので子の白の四角しか見えなくなります(リスト1 - [3])。

図10のbでは、子は親の上辺と左辺が同じ位置に固定され、子の白のサイズ次第で親のグレーが見えます。

図10のcでは、上辺が親と同じ位置に固定され、子の左辺が親の右辺に固定されます。

図10のdでは、親の中心に配置します。子のサイズによってグレーの枠幅が決まります(リスト1 - [4])。

## ③マウス入力

MouseAreaエレメント(リスト1 - [6])を使用してマウス入力を処理します。処理できるシグナル(Qtでのイベントのこと。詳細は後述)は表3のとおりです。

onClickedシグナルハンドラなどの定義はMouseArea::onClicked (MouseEvent mouse)となっています。MouseEventエレメントの引数があり、座標(mouse.x)など追加情報を取得することが可能です。

## ④プロパティバインディング

サンプルを実行してウィンドウサイズを変更する

■ 図8 Hello World実行イメージ



■ 図9 デザインボタン



## ■ リスト1 デザインボタン

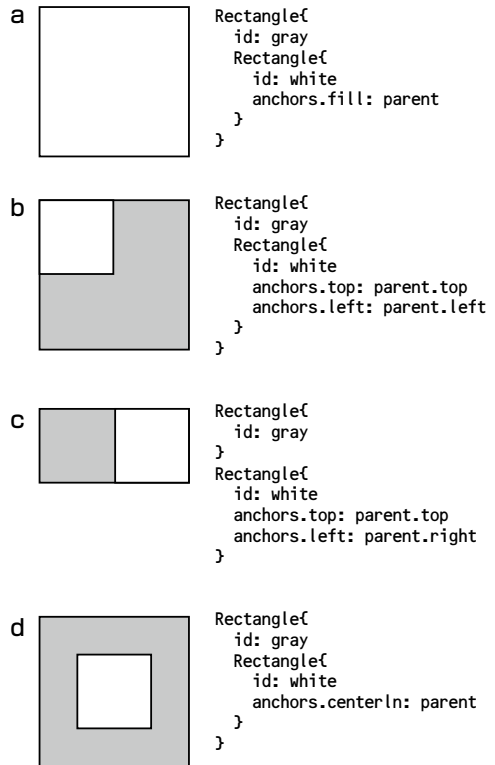
```

import QtQuick 1.1

Rectangle {
    id: root
    width: 150
    height: 40
    radius: height * 0.1 // 角を丸く
    smooth: true // なめらかに描画する
    // 背景のグラデーション
    gradient: Gradient{
        GradientStop{ id: gradBase1; position: 0.0;
            color: "#000000" }
        GradientStop{ id: gradBase2; position: 0.2;
            color: "#444444" }
        GradientStop{ id: gradBase3; position: 1.0;
            color: "#888888" }
    }
    // プロパティの定義
    property alias text: caption.text // ← [1]
    // シグナルの定義
    signal clicked(variant mouse) // ← [2]
    // ハイライトのエリア
    Rectangle{
        anchors.fill: parent // ← [3]
        anchors.topMargin: 2
        anchors.leftMargin: 2
        anchors.rightMargin: 2
        anchors.bottomMargin: 0
        radius: parent.radius
        smooth: true
        // ハイライトのグラデーション
        gradient: Gradient{
            GradientStop{ position: 0.0; color: "#ddffffff" }
            GradientStop{ position: 0.4; color: "#44ffffff" }
            GradientStop{ position: 1.0; color: "#00ffffff" }
        }
    }
    // キャプション
    Text {
        id: caption
        anchors.centerIn: parent // ← [4]
        color: "white"
        font.pixelSize: parent.height * 0.5 // ← [5]
        text: "button"
    }
    // マウス入力
    MouseArea { // ← [6]
        id: mouseArea
        anchors.fill: parent
        hoverEnabled: true // マウスカーソルが重なったときの処理 ON
        // 左右クリックに反応 (デフォルトは左のみ)
        acceptedButtons: Qt.LeftButton | Qt.RightButton
        // クリックイベント
        onClicked: {
            // シグナルを飛ばす
            root.clicked(mouse); // ← [7]
        }
    }
    // 状態管理
    states: [ // ← [8]
        // マウスボタンダウン時の状態
        State{
            name: "press"
            when: mouseArea.pressed // マウスが押下されるとき
            // 変更するプロパティ
            PropertyChanges { target: gradBase1; color: "#220000" }
            PropertyChanges { target: gradBase2; color: "#440000" }
            PropertyChanges { target: gradBase3; color: "#dd0000" }
        },
        // マウスカーソルオーバー時の状態
        State {
            name: "hover"
            when: mouseArea.containsMouse // カーソルが範囲内にあるとき
            // 変更するプロパティ
            PropertyChanges { target: gradBase1; color: "#000022" }
            PropertyChanges { target: gradBase2; color: "#000044" }
            PropertyChanges { target: gradBase3; color: "#0000dd" }
        }
    ]
}

```

## ■ 図10 anchorsプロパティを使用した配置



■ 表3 マウスイベント一覧

名称	タイミング
onClicked	クリックしたとき
onDoubleClicked	ダブルクリックしたとき
onPressed	押下したとき
onReleased	放したとき
onPressAndHold	長押ししたとき
onEntered	カーソルがエリア内に入ったとき*
onExited	カーソルがエリア外に出たとき*
onPositionChanged	カーソルが移動したとき*
onCanceled	シグナル処理が許可されなかったときか、他のエレメントの処理が行われるとき

\*1 hoverEnabled プロパティがtrueのとき発生する。

## Column 03

## parent とは

サンプルの中で使用しているparentというキーワードですが、Rectangleエレメントなどが継承しているItemエレメントのプロパティで、文字どおり親のエレメントを示しています。

とウィンドウの高さにあわせて文字列のサイズが変化するのがわかります。

Text要素のfont.pixelSizeプロパティ(リスト1- [5])を見てください。parent.height\*0.5と指定しています。プロパティに、プロパティを使用した式を指定しています。これをプロパティバインディングと言い、非常に重要なQMLの機能です。プロパティの値が変化すると指定している要素のプロパティへも変化が伝搬します。

他の言語では、プロパティの値が変化したときのイベントを利用して他のプロパティの値を書き換えるコードを記述するのがよくあるパターンですが、これが一切不要になります。おかげでコードがシンプルかつ見やすくなります。

注意点として、JavaScriptコードからプロパティへ代入するとバインドが解除されます。MouseArea要素のonClickedシグナルハンドラに次のコードを追加してみてください。

```
caption.font.pixelSize = 10;
```

クリック後にウィンドウサイズを変更してもフォントサイズが追従しくなくなります。この動作が難しくもあり面白いところでもあります。1度バインディングさせたら、あとはずっと代入をしないようにする必要がありますので腕の見せ所です。

もしかするとユーザー操作に反応して変化させたい場合はどうにもならないと思われた方もいらっしゃるかもしれませんが、それは次節で解説する状態管理で対応可能です。

## ⑤状態管理

このサンプルでは、マウスカーソルをウィンドウ

内(厳密にはMouseArea要素内)に持って行くとき背景が青くなり、マウスボタンを押下すると赤くなります。

QMLでは「状態」を定義してその状態への遷移条件や変化させたいプロパティの組み合わせを管理することができます。たとえば、「マウスを押下したときに赤くする」という直感的な状態変化をコードにできます。

ルート要素のstatesプロパティ(リスト1- [8])に、State要素の配列を指定して「マウスカーソルがエリアに入ったとき」と「マウスを押下したとき」の2つの状態を定義しています。whenプロパティがtrueのとき、状態が切り替わってPropertyChanges要素で指定された要素のプロパティが変化します。

また、リスト1では背景のグラデーションの色を変更するためにターゲットにgradBase1、2、3を指定しています。ここではcolorプロパティのみ指定していますが、同じターゲットに対しては複数のプロパティを変更することができます。

ところで、QMLではプロパティの値を動作中に変更する方法が3つあります。

- プロパティバインディング
- シグナルハンドラ内のJavaScriptコード
- 状態変化での変更

ここまでの値変更は状態管理機能を使用しなくてもプロパティバインディングやシグナルハンドラ内のJavaScriptコードでも実装が可能です。たとえば次のように、プロパティバインディングと3項演算子を使用してマウスが押下されたときに色を変化させることができます。

```
color: mouseArea.pressed ?  
    "#ddddd" : "#dd0000"
```

ですが、マウスカーソルが上に乗ったときも色を変化させたいので式が複雑になってしまいますし、似たようなコードが複数カ所に散らばってしまうことも考えられますので可読性が落ちて現実的ではありません。すべての状態が1カ所で管理できるの

### Column 04

#### idプロパティ

targetプロパティに指定しているgradBase1は、GradientStop要素のidプロパティに指定されている値です。このように特定の要素へアクセスすることも可能です。whenプロパティのmouseAreaも同様です。

で、コードの見通しも良くなります。



## ⑥機能のカプセル化

ここまでボタンを題材に説明を進めてきました。これをカプセル化して再利用できるようにします。リスト1の内容をコピー&ペーストすれば同じものをいくつも配置できますが、芸がありません。自作のエLEMENTにしてしまいましょう。

プロジェクト名を右クリックして「新しいファイルを追加」を選択してください。開いたウィンドウの左側で「Qt」を選択し、右側で「QMLファイル」選択してください。

次にファイル名を入力しますが、2つ注意点があります。

- 大文字ではじめる
- カプセル化したELEMENT名になる(クラス名のようなもの)

ここではファイル名を「Button」とします。

次に今まで作ったコードをmain.qmlから、Button.qmlへすべてコピー&ペーストしてください。これでButtonELEMENTが作成できました。他のQMLファイルからButton{}と書いて呼び出せます。

最後に、main.qmlの内容をリスト2のように変更して実行してください(図11)。ボタンをクリックすると下半分の色付きの部分が左右に移動してページ切り替えの雰囲気を出しています。



## ⑦プロパティ

配置したELEMENTにはオリジナルのプロパティを追加できます(リスト1- [1])。プロパティには大きく分けて2パターンあります。実際に値を持つ場合と、サンプルのように特定のELEMENTのプロパティへのエイリアスを設定する場合です。

先ほど作成したButtonELEMENTの中にある

### Column 05

### 色の指定方法

colorプロパティには、SVGカラー名と#AARRRG BB、#RRGGBBの3種類の指定方法があります。

### ■リスト2 ページ切り替え機能

```
import QtQuick 1.1

Rectangle {
    id: root
    width: 360
    height: 360
    // 横方向へボタンを並べる
    Row {
        id: btnArea
        spacing: 2
        // 1 ページ目用のボタン
        Button {
            id: btn1
            width: 150
            height: 30
            text: "Twitter" // ← [1]
            onClicked: {
                // 状態の変更は名前の指定でも可
                root.state = "page1";
            }
        }
        // 2 ページ目用のボタン
        Button {
            id: btn2
            width: 150
            height: 30
            text: "Hello"
            onClicked: {
                root.state = "page2";
            }
        }
    }
}

Row {
    id: content
    anchors.top: btnArea.bottom
    anchors.bottom: parent.bottom
    // 1 ページ目 (リスト4の差し替え場所)
    Rectangle {
        id: page1
        width: root.width
        height: parent.height
        color: "#F692F7"
    }
    // 2 ページ目
    Rectangle {
        id: page2
        width: root.width
        height: parent.height
        color: "#F6F792"
        Text {
            anchors.centerIn: parent
            font.pointSize: 14
            text: "Hello Qt Quick"
        }
    }
}

// ページの表示状態を管理する
states: [
    // 1 ページ目
    State {
        name: "page1"
        PropertyChanges { target: content; x: 0 }
    },
    // 2 ページ目
    State {
        name: "page2"
        PropertyChanges { target: content; x: -root.width }
    }
]

// ページ切替え時のアニメーション
transitions: Transition {
    NumberAnimation { // ← [2]
        target: content
        properties: "x" // 対象のプロパティ
        easing.type: Easing.OutBounce
        duration: 400 // アニメにかかる時間
    }
}
```

■ 図11 サンプルアプリ (ページ切り替え機能)



■ 表4 シグナルハンドラの名称の例

	名称	ハンドラ名称
シグナル	test	onTest
プロパティ	width	onWidthChanged

Text要素のプロパティを外部から変更することはできません。直接プロパティを指定できるのはルートのRectangle要素のみです。ですの  
でルートの要素にプロパティを追加して間接的にアクセスできるようにします。

エイリアスは直接外部からアクセスしたいプロパティに設定します。それ以外のときは、値を持つ形にします。複数のプロパティに指定したい場合や式に組み込む必要がある場合などです。定義は以下のようにします。

```
property string text: "default"
```

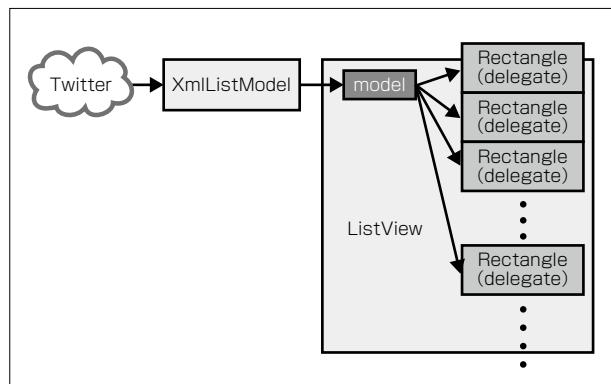
リスト2- [1]の部分は追加したプロパティに値を設定しているところです。使用方法是標準で用意されているプロパティと同様です。

## ⑧シグナル

シグナルはQt Quickでイベントを通知するためのしくみです。このしくみを使用すると上位の要素へイベントを通知できるようになります。プロパティ同様にシグナルもオリジナルのものを追加できます(リスト1- [2])。

シグナルの名称の付け方には注意が必要です。シグナルハンドラがシグナル名から決定されるためです。シグナルハンドラの名前に「on」が付加され、元の先頭の文字が大文字になります(表4)。シグナ

■ 図12 リストビューに表示するまでの情報の流れ



ルの引数の型は、string、int、real、variantなどを使用します。

サンプルではButton要素にクリックイベントを追加するので、MouseArea要素を真似してシグナルハンドラがonClickedになるようにclickedシグナルを定義しています(リスト1- [2])。

シグナルを発生させるためにはシグナルハンドラなどで呼び出します(リスト1- [7])。

プロパティにはチェンジシグナルが自動的に作成されます。追加したプロパティも同様です。チェンジシグナルハンドラの名称は先頭に「on」が付加され、元の先頭の文字が大文字になり、後ろに「Changed」が付きます(表4)。

## ⑨アニメーション

QMLではNumberAnimation要素(リスト2- [2])などを使用してアニメーションに関する設定をします。アニメーションはプロパティの値の変化を自動的に補間する形で実現されます。開始するトリガーは、start()メソッドを使用する場合と対象プロパティの変化のタイミングで自動的に開始する場合の2つがあります。

サンプルではtransitionsプロパティを使用し状態遷移時に設定しています。ほかにも複数のアニメーション設定を順番に実行したり、平行で実行したりすることも可能です。

アニメーションの動きはeasing.typeプロパティで指定できます。サンプルはページの切り替わりの

終わりでバウンドするような動きをします。ほかにもさまざまな動きが用意されています。イメージに合うものを選択してください。

## 簡単XML解析で TwitterのTLを表示

ここまでで、ボタンを使って2画面遷移するアプリの大枠は完成しました。ここからもう少し具体的な機能を追加してみましょう。

QMLには、XMLを解析してリストビューに表示するときに便利なエレメントがあります。リストビューに表示しないで解析だけで使用しても非常に有効です。サンプルではTwitterを利用し、特定ユーザのフォローによるリツイートを表示することになります。図1のようにアイコンとツイートの本文を一覧表示し、項目をクリックするとスクリーンネームを表示するようにしていきます。

本来ならOAuthの認証から紹介したいところですが、誌面の都合もあり認証のないAPIを使用します。サンプルプロジェクトにTwitter.qmlを追加して、リスト3の内容に修正してください。次にmain.qml(リスト2)の1ページ目用のRectangleエレメントをリスト4のように修正してください。



### 情報の流れ

ここからはTwitter.qml(リスト3)の中身を見ていきましょう。

各エレメントと情報の流れのイメージは図12のようになります。XmlListModelエレメント(リスト3- [1])を使用してTwitterから情報を取得し、アプリ内へインターフェースを提供します。ListViewエレメント(リスト3- [2])にはモデル系のエレメントと連携する機能があり、modelプロパティを使用して結びつけます。あとは、delegateプロパティ(リスト3- [3])に設定したレイアウトを元に、情報の一覧を生成してくれます。



### XMLの解析

XmlListModelエレメント(リスト3- [1])には、どこから情報を取得してどのような情報を公開する

### ■ リスト3 Twitterの情報をリストビューに表示

```
import QtQuick 1.1

Rectangle {
    id: root
    // プロパティを定義
    property string screenName: "sakidoritter"
    // 背景
    gradient: Gradient{
        GradientStop { position: 0.0; color: "#2D4761" }
        GradientStop { position: 0.2; color: "#45718C" }
        GradientStop { position: 1.0; color: "#2D4761" }
    }

    //Twitter からデータを取得、解析
    XmlListModel{ // ← [1]
        id: modelTw
        source:
            "http://api.twitter.com/1/statuses/retweeted_to_user.xml"
            + "?screen_name=" + screenName
        query: "/statuses/status"
        XmlRole{ name: "text";
            query: "retweeted_status/text/string()" }
        XmlRole{ name: "name";
            query: "retweeted_status/user/screen_name/string()" }
        XmlRole{ name: "icon";
            query: "retweeted_status/user/profile_image_url/string()" }
    }

    // 情報の表示
    ListView{ // ← [2]
        anchors.fill: parent
        clip: true
        model: modelTw
        delegate: Rectangle{ // ← [3]
            id: area
            width: root.width
            height: status.height * 1.3
            color: "#00000000"
            border.color: "darkgray"
            border.width: 1
            Row{
                id: status
                width: parent.width * 0.95
                anchors.centerIn: parent
                spacing: 5
                Image{ // ← [4]
                    id: icon
                    anchors.verticalCenter: parent.verticalCenter
                    width: 48
                    height: 48
                    source: model.icon
                }
                Text{ // ← [5]
                    id: text
                    width: parent.width - icon.width - parent.spacing
                    color: "#D5EAF2"
                    wrapMode: "WordWrap"
                    text: model.text
                }
            }
        }
        MouseArea{ // ← [6]
            anchors.fill: parent
            onClicked: {
                text.text = model.name;
            }
        }
    }
}
```

### ■ リスト4 メイン画面の修正

```
//1 ページ目
Twitter{
    id: page1
    width: root.width
    height: parent.height
}
```

かを設定します。まず、sourceプロパティにWeb APIのURLを指定します。そしてスクリーンネームをプロパティで外部から修正できるようにしています。このAPIではリスト5のようなXMLが取得できます。

statusがツイート単位で繰り返し出てきますので、queryプロパティに基準パスとして"/statuses/status"を設定します。XmlRoleエレメントで公開する情報の名称とパスを設定します。nameプロパティにレイアウト側から参照するための名称を設定し、queryプロパティに欲しい情報へのパスを設定します。この書式はW3C XPath 2.0で記述します。

これでTwitterのAPIで取得したXMLの解析と管理ができるようになります。Webへのデータの取得はsourceプロパティが更新されたときなので、現状では起動時に一度読み込まれるだけです。

## リストビュー

ListViewエレメント(リスト3-[2])は、元にする情報と表示するためのレイアウトを設定すれば基本は終了です。スクロールの制御などはすべて行ってくれます。個々のデータが画面内に入ると、delegateプロパティに指定されたレイアウトを使用して個々の情報が生成され、画面外に出ると破棄されます。

サンプルを実行しクリックすると、ツイート内容がスクリーンネームに切り替わるようにしていますが、一度スクロールアウトさせると元に戻ります。このことから再生成されていることがわかります。

### ■ リスト5 XMLの抜粋

```
<?xml version="1.0" encoding="UTF-8"?>
<statuses type="array">
  <status>
    <created_at>Sat Aug 25 06:00:00 +0000 2012</created_at>
    <id>XXXXXXXXXX</id>
    <text>RT @XXXXX: サンプル</text>
    <retweeted_status>
      <text>サンプル</text>
      <user>
        <screen_name>YYYYY</screen_name>
        <profile_image_url>http://profile/image.png</profile_image_url>
      </user>
    </retweeted_status>
  </status>
  <status>...</status>
  <status>...</status>
</statuses>
```

clipプロパティをtrueに設定している理由は、子のエレメントが自分の範囲外に出たときに境界で消えるようにするためです。指定しないとスクロールアウトしたはずの情報が見えてしまってカッコ悪いことになります。

## レイアウト定義

delegateプロパティ(リスト3-[3])に個々の情報を表示するためのレイアウトを指定します。サンプルではListViewエレメントの中に直接記述していますが、別ファイルにしたQMLファイルを指定することも可能です。

アイコンをImageエレメント(リスト3-[4])で、ツイート内容をTextエレメント(リスト3-[5])を使用して表示します。それぞれmodel.iconとmodel.textという形でListViewエレメントのmodelプロパティ経由でXmlListModelエレメントの内容にアクセスしています。もし処理してる情報が何番目か知りたい場合は、model.indexで知ることができます。

MouseAreaエレメント(リスト3-[6])のonClickedシグナルを利用して、ツイートをスクリーンネームへ変更しています。その際、text.text = model.name;としています。Rectangleエレメント(リスト3-[3])のオブジェクトが情報ごとに生成されるので、1つ前の情報にアクセスすることはありません。今までの静的な配置方法とは少しイメージが違いますので注意してください。

## 配布パッケージ作成

最後に作成したアプリの配布方法について解説します。対象OSによって若干状況は違いますが、必要なライブラリを集めてパッケージングするという流れは同じです。

## 必要なライブラリを集める

アプリを作ったPCのフォルダから必要なライブラリ(表5)を集めて、実行ファイルと同じフォルダに配置し

■表5 リリース用ライブラリ一覧

OS	項目	ライブラリ名
Windows/ Linux/ Mac	必須	QtCore4、 QtDeclarative4、 QtGui4、 QtNetwork4、 QtScript4、 QtSql4、 QtXmlPatterns4
Windows	MinGW のとき	libgcc_s_dw2-1.dll、 mingwm10.dll
Windows	Visual Studio 2008 のとき	msvcr90.dll、 msvcp90.dll

ます。ライブラリにはQtの標準機能のライブラリと、プラグインと呼ばれている拡張機能用の2種類があり、それぞれ次のフォルダに保存されています。

#### ●標準ライブラリ

Qt<Version>%<Compiler>%bin

#### ●プラグイン

Qt<Version>%<Compiler>%plugins

Windowsの場合はDependencyWalker、Linuxの場合はlddコマンドを使用して依存しているライブラリを調べます。ただし、プラグインは実行時に動的にロードするので実行するまでわかりません。もしQt Creatorから実行すると動作するが単独で実行すると動作しない場合は、何かプラグインが不足している可能性があります。

配布パッケージの作成についてはまだまだ面倒な状況は否めませんが、インストーラ作成プロジェクトも非公式ですがあるようなので今後に期待したいところです。



## Macでの手順

Macのみリリース用のパッケージ作成ツールが付属していますのでコマンド1つでOKです。次のフォルダにあるツールを実行してください。ただし、サードパーティ製やオリジナルのプラグインは手動で含める必要があります。

```
Qt<Version>/<Compiler>/bin/macdeployqt
appname.app
```



## パッケージのファイルの構成例

Windowsを例にすると次のようになります。

```
実行ファイル : <App>%*.exe
必須DLL       : <App>%*.dll
Qt プラグイン : <App>%<qtpluginsfolder>%*.dll
QML ファイル  : <App>%qml%<projectname>%*.qml
```

なお、本稿で作ったWindows用サンプルのファイル構成は、注1のサイトからダウンロードできる実行ファイルをまとめたzipファイル内を参考にしてください。

## まとめ

いかがだったでしょうか？ 本稿ではQt Quickをはじめるにあたって最低限知っておきたい部分を厳選して紹介しました。次のようなアプリをより良くできる内容もまだまだ控えています。どんどんチャレンジしてみてください。

- C++でオリジナルのエLEMENT作成
- Loader ELEMENTを使用したELEMENTの動的ロード
- WorkerScript ELEMENTを使用したマルチスレッド処理
- 多言語対応
- Particles ELEMENTを使用したエフェクト効果

QMLは今までにあまり経験したことのない言語仕様で戸惑った方もいらしたかもしれません。プロパティバインディングと状態管理を上手に使用して、JavaScriptコードをできるだけ減らせるように心がけるときれいでわかりやすいソースコードになります。使用できるELEMENTは数えられるレベルですが、組み合わせ次第で素晴らしいアプリが作成可能です。あまり固く考えず、柔軟な思考で楽しんでいただければと思います。もしわからないことがあればTwitterで気軽に声をかけてください。また、ハッシュタグ#qtjpでつぶやいてみてください。素晴らしいQtエンジニアの方々とコミュニケーションが取れます。SD

増え続ける脅威に対抗せよ!

# インターネットを 陰で見守る Akamai

アカマイ・テクノロジーズ合同会社

渡邊 圭太 WATANABE Keita  
keiwatan@akamai.com

土屋 貴史 TSUCHIYA Takashi  
tatsuchi@akamai.com

三宅 琢也 MIYAKE Takuya  
tmiyake@akamai.com

倉持 尚良 KURAMOCHI Naokazu  
nkuramoc@akamai.com

中谷 喜久 NAKATANI Yoshihisa  
ynakatan@akamai.com

新村 信 NIIMURA Makoto  
mniimura@akamai.com

アカマイは知る人ぞ知るCDN(Contents Delivery Network)サービスを展開する企業です。その世界規模の分散型ネットワークはWebサイトの拡張性や高速化を実現するだけでなく、DDoSなどの攻撃トラフィックから顧客企業のWebサイトを守る役割も果たします。本稿では、日々発生する大量のトラフィックの分析で培った、アカマイ独自のインターネットセキュリティ技術を紹介します。



## 増加するサイバーテロの脅威

昨今、ニュースや新聞紙上などでサイバーテロに関する話題を目にしない日がないほど、事件が頻発しています。警察庁の発表によると、平成23年度のネットワーク利用犯罪検挙件数は過去最高の5,339件を記録したものの、警察が把握した範囲で国内を標的としたDDoS攻撃は前年比約60倍に増加、標的型メールは1,277件に達し、国民生活を脅かすサイバー犯罪が深刻化しているそうです<sup>注1</sup>。

アカマイは「インターネットを信頼できるものにする」ために存在している会社です。インターネットを舞台としたサイバー犯罪から、お客様を守ることも重要な使命の1つです。本稿ではこの使命を果たすためのアカマイの取り組み、とくに「Akamai Intelligent Platform」に実装されたセキュリティソリューション「Kona Site Defender」が数々の脅威に対する効果的な防御壁の役割を果たす様子を実例とともに解説します。

## あらゆるトラフィックを陰で見守るアカマイ

現在、Fortune Global 500企業のおおよそ3社に1社はアカマイの配信サービスを利用いた

だしています。採用理由は「サイトのレスポンスを高速化したい」「世界中からの膨大な量のトラフィックを効率よく捌きたい」「インターネット越しのビジネストラフィックを安定して完結したい」「災害対策でのバックアップセンタへの切り替えを自動化したい」などが代表的です。いずれもアカマイが提供する「信頼できるインターネット」上で効果的なビジネス展開を目指す企業、団体がお客様となっています。

一方、インターネットユーザはアカマイの存在や役割を意識する必要がありません。アカマイのお客様のサイトへアクセスする際、知らず知らずのうちにアカマイのプラットフォームを介しているのです。

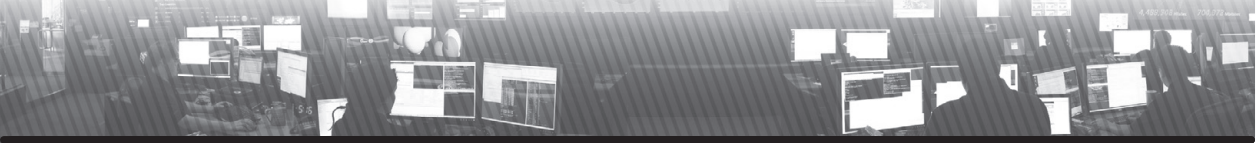
そして、一般のユーザの方がアカマイ経由で著名なサイトへアクセスすると同様に、著名なサイトを攻撃対象とする悪意のあるトラフィックもアカマイを経由することになります。つまりアカマイの配信サービスを利用いただいている企業、団体のWebサイトへの攻撃をアカマイは常に監視しているわけです。常時、世界中のインターネットトラフィックの20~30%を配信しているアカマイネットワークは、世界で最もサイバー攻撃のトラフィックを受信しているプラットフォームと言えるかもしれません。



## 攻撃の手口と防御手段

インターネット上の特定サイトを標的にした

注1) <http://www.npa.go.jp/keibi/index.htm>



攻撃は、時代とともに手法を変えるため、世界を代表するような著名なサイトの運営者は、サイト利用者のアクセス対応のみならず、日進月歩で進化するサイバー犯罪に対処していかなければなりません。

攻撃者の意図が情報を盗み取ることである場合、手口としては、Webサイトをホストするサーバへの侵入経路を探す方法(Port Scan)やアプリケーションプログラムの脆弱性を狙う方法(アプリケーションへのアタック)が代表的です。これらの防御方法はそれなりに確立されています。企業はそのため自社データセンターにファイアウォールなどの機材を装備して防御壁を築き、日々運用管理を行う必要があります。

一方、攻撃者の意図が特定企業のサイトを利用不能にすることである場合、DDoS(分散型 Denial of Service)と呼ばれる攻撃が行われます。

DDoS 攻撃は広域に分散する多数のコンピュータが一斉に特定のサーバへ通信を行い、ネットワーク、その他のリソースを逼迫させることで、サービス機能停止させてしまう攻撃です。従来、DDoS 攻撃はおもにネットワーク層、トランスポート層での攻撃が主流でしたが、最近ではアプリケーション層での攻撃も存在します。たとえばHTTPのリクエスト手法を細工するなど、より複雑性が増した攻撃手法に変化してきています。

DDoS 攻撃がほかの攻撃手法に比べて特徴的なのは、想定外のアクセスを1ヵ所に集中させるため、企業が自前で用意する防御壁では、処理しきれない点が挙げられます。通常時のトラフィックの10～1000倍もの攻撃トラフィックが押し寄せてくるわけですが、これに対処するためのスケールを用意することは事実上不可能です。



## Webセキュリティ、誰も無関心ではられない時代

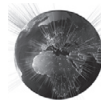
Webセキュリティの脅威と対策の話をする、多くのお客様から「うちは事業規模が小さいから関係ない」との返事が返ってきます。しかし

ながら、事実は異なります。

攻撃者の動機によって対象は異なりますが、金銭を目当てとする場合、攻撃者は少ない労力で成果をあげたいため、中小企業など比較的Webセキュリティへの関心が低いサイトを攻撃対象とする傾向が見られます<sup>注2</sup>。海外でのケースとなりますが、43%の中小企業は現在のセキュリティ攻撃に備えるだけの適切な準備ができていないとの報告も寄せられています<sup>注3</sup>。また、72%の攻撃先は中小企業との調査結果もあります<sup>注4</sup>。Webサイト上にてインターネットユーザ固有の情報を扱う場合、事業規模の大小にかかわらず、企業ブランドや市場に与える影響は非常に大きいと言わざるを得ません。

インターネットを介したコミュニケーションコストの低減により、以前なら一部の愉快犯、高度なスキルを持つエンジニア、ハッカーなど、限られた人々のみが利用していた非常に高度な攻撃ツールも容易に入手でき、誰でもサイトへ攻撃できる環境となっています。サイバー攻撃への対策は、けっして大企業だけに課された課題ではありません。

以下、アカマイならではのセキュリティソリューション「Kona Site Defender」が、企業をインターネットの脅威から守る様子を紹介させていただきます。



## Akamai Intelligent Platform

まず初めに、アカマイのコアテクノロジーを実装した世界最大の広域分散型コンピューティングプラットフォームについて紹介します。世界75ヵ国に10万台以上のサーバを1,100以上のネットワーク(通信キャリア、ISP、インターネッ

注2) <http://www.sophos.com/en-us/medialibrary/Gated%20Assets/white%20papers/sophosclosingbackdoornetworkapplicationvulnerabilitieswpna.pdf>

注3) <http://www.mcafee.com/us/resources/white-papers/wp-combating-smb-threats.pdf>

注4) <http://www.techi.com/2012/06/small-businesses-must-take-preventative-measures-to-safeguard-against-hackers/>

トエクスチェンジなど)へ設置し、インターネット上に形成されたオーバーレイネットワークをAkamai Intelligent Platformと呼んでいます。

アカマイはIntelligent Platformを通して、トラフィック量、輻輳中状態や障害発生個所など、インターネットの情報をリアルタイムでモニタしています。これらの情報により経路選択の最適化やコンテンツのキャッシュなどを効率的に行い、より速く、より信頼性の高い配信や通信を実現しています。

インターネットのビジネス適用領域は、動画配信、オンラインゲーム、Eコマース、SNS、ソフトウェア配信、企業内外のビジネストラナクションなど多岐に及びます。アクセス元となるクライアントもPCのほか、スマートフォン、タブレット端末、スマートTVなどさまざまなデバイスの出現により急激な拡大を続けています。これら環境の変化を受け、インターネット上のトラフィックは劇的な増加を続けており、エンドユーザはインターネットに内在する諸問題(TCPプロトコルの非効率性、パケットロス、突発的な混雑など)を受けやすくなっています。一方、コンテンツホルダー側は急激なトラフィック増に耐え得るオリジンサイトの構築および膨大な帯域を用意する必要に迫られています。

コンテンツホルダー側の対策として昨今では「クラウド」の活用が進んでいます。XaaS(Everything as a Service)を利用して自社サイトをクラウド化することで、サーバへの費用やトラフィック急増への問題はある程度解消されることを期待しています。

しかし、クラウドと言っても中身はぎっしりサーバが配置されたデータセンターで、その拠点数も限定的であり、結局多くのクライアントはインターネット越しに遠隔地のサーバにアクセスしなければならない、という構図はあまり変わりません。こうした環境でネットワークの問題に対する対策は行えません。それを解決しているのがコンテンツ配信ネットワーク(CDN)という考え方であり、Akamai Intelligent Platform

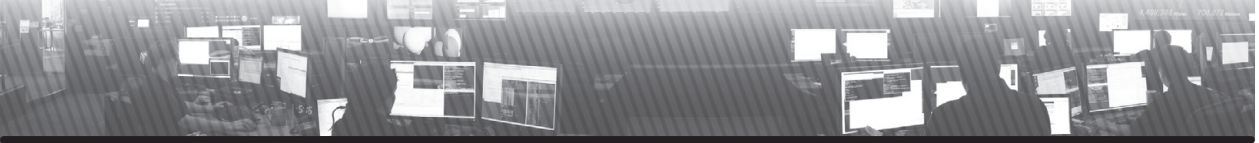
の高速化／分散処理技術です。



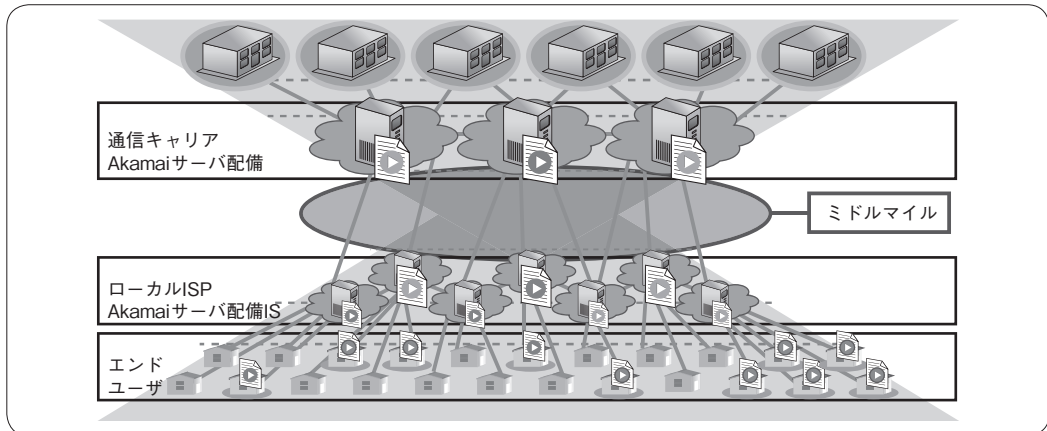
## 分散型アーキテクチャの メリット

アカマイはエンドユーザから近いネットワークのエッジ／Edge(ISP)に展開したサーバ(エッジサーバ)にコンテンツをキャッシュし、そこから配信するしくみを実現しています。回線混雑の発生しやすい上流のピアリングポイントを通過することなくコンテンツやアプリケーションの配信を行うため、ネットワークプロバイダがしばしば実施する帯域制御によるスループット低下の影響を受けません。また、ミドルマイルを介さずにコンテンツを返すことで、上述したインターネットに内在する諸問題も回避できます(図1)。また、エッジサーバは世界中に分散配置されているため、あらゆる地点からのアクセスに対して最寄りのエッジサーバを割り当てることができます。これにより、お客様のオリジンサーバに変更を加えることなくスケラビリティを確保できます。

キャッシュできないコンテンツを取得するためには、オリジンサーバとの間の長い距離のネットワークを経由しなければなりませんが、Akamai Intelligent Platformによるオーバーレイネットワークは、BGP(Border Gateway Protocol)のルーティング制御による経路とは異なる、より適切なショートカットルートを選択するようになっています。世界のどこかで年に数回発生している海底ケーブルの切断事故や、自然災害による電力供給の停止などの事故が発生した場合、BGPによるルーティング制御に依存すると問題地点を迂回できず、インターネットを利用できなくなる恐れがあります。Akamai Intelligent Platformはそのような問題地点を回避し、より安定したインターネット利用を可能にしています。また、常時複数のルートにおけるパフォーマンスを計測し、その都度最適な(レイテンシーのロスが少ない)ルートを割り当てているため、キャッシュされていないコンテンツに対しての高速配信を実現していま



▼図1 分散型アーキテクチャによるミドルマイルボトルネックの回避



す。その他にもTCPレイヤにおける高速化のための独自のチューニングや、Prefetchなどのコンテンツの先読みなど、高速化のためのさまざまな技術を導入しています。

ここでは、Akamai Intelligent Platformのコアを成す2つの機能、独自のマッピングシステムと独自のProxyソフトウェアを紹介します。

## ダイナミックマッピング

アカマイはDNSのしくみを使ってエンドユーザからのリクエストを最適なエッジサーバへ割り当てています。アカマイのサービスを導入した企業のDNSにはCNAMEレコードが登録されており、アカマイからコンテンツを配信するWebサイトのドメインの別名として、アカマイが持つドメイン名が設定されています(Akamai化)。このため、エンドユーザがAkamai化されたサイトの名前解決を問い合わせるときに、アカマイのDNSへリダイレクトされます。アカマイのDNSはダイナミックマッピングという機能を持ち、問い合わせ元のエンドユーザのネットワーク情報を基に、最適なエッジサーバのIPアドレスを返します。

ダイナミックマッピングが選ぶ最適なエッジサーバの条件は次のとおりです。

- ・エンドユーザから距離的に近いネットワークに位置していること
- ・エンドユーザとの間のネットワーク経路が混雑／切断していないこと
- ・正常に稼働していること
- ・リソース使用状況に空きがあり、追加の処理を受け付ける余裕があること
- ・リクエスト対象の情報がキャッシュに載っている可能性が高いこと

これらの状況は刻々と変化し続けるため、ある1エンドユーザに対しても最適なエッジサーバの割り当ては動的に変化します。

ダイナミックマッピングは接続要求元ごとに最適なエッジサーバを割り当てるためのマップ(対応表)を作ります。全エンドユーザのIPアドレスに対してマップを作ることは数量的に困難ですので、エンドユーザを地理的にグループ化します。エンドユーザが利用するローカルDNSは、エンドユーザから近い場所にあることから、このローカルDNSをエンドユーザの代表としてみなし、マップ管理対象数を限定しています。

ダイナミックマッピングを有効に機能させるために、インターネットのパフォーマンスに関する膨大なデータが常時収集されています。割り当て結果のログは解析され、マップ作成をよ

り最適化するためにフィードバックされています。また、マップ情報は30秒ごとに更新され、アカマイのDNSが解決するエッジサーバのIPアドレスは常に最新のネットワーク状況を反映しています。



## アカマイ配信制御の中核 GHost

Akamai Intelligent Platform上では、GHost (Global Host、ゴースト)と呼ばれるアカマイ独自の高性能なリバースHTTP Proxyソフトウェアが動いています。これはエンドユーザのリクエストを処理してコンテンツを配信しています。GHostは一般的なHTTP Proxyソフトウェアと比べると、次のようにコンテンツに対する挙動を非常に細かく制御できます。

- ・コンテンツのパスや拡張子ごと、または特定のクエリストリングやCookieを動的に組み込んだキャッシュキーを基に、キャッシュの生存期間やリダイレクト、アクセスコントロールなどを設定できる
- ・エンドユーザの国別に異なるパスにリダイレクトさせることができる
- ・HTML文書内のすべてのコンポーネントのダウンロード(フロントエンド処理)を高速化できる

GHostの挙動を決めているのは、メタデータと呼ばれるXMLファイルです。メタデータはAkamai化対象サイトごとに作成し、GHostサーバ<sup>注5</sup>すべてに配置する必要があります。

これら設定作業はLuna Control Centerと呼ばれるWeb管理画面を操作することですべて完結します。数十万台のGHostサーバへのメタデータ展開は、完全に自動化されたプロセスによって1時間程度で完了します。



## GHost上のさまざまな アプリケーション

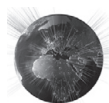
GHostには高度な配信サービスを実現するた

めの多くの機能が実装されています。たとえば動的に生成されるWebコンテンツをより高速配信するためのさまざまな高速化テクニックが実装されています。

また、成長著しい動画配信サービス分野ではHDS、HLS、SmoothStreaming、MPEG DASH(on the roadmap)の配信を可能にし、かつあらゆるデバイスに対する最適化配信を実現しています。

さらに、Securityの分野においても、もともと世界中に分散されたエッジサーバがDDoSなどにおけるトラフィック攻撃を吸収可能であることに加え、高度なWAF(Web Application Firewall)、DNSSEC(DNS Security Extensions)、Access Controlなどを実装し、お客様の要望に合わせた最適なセキュリティソリューションの提供が可能です。

次からは、このAkamai Intelligent Platformに構築されたセキュリティ機能「Kona Site Defender」について紹介します。

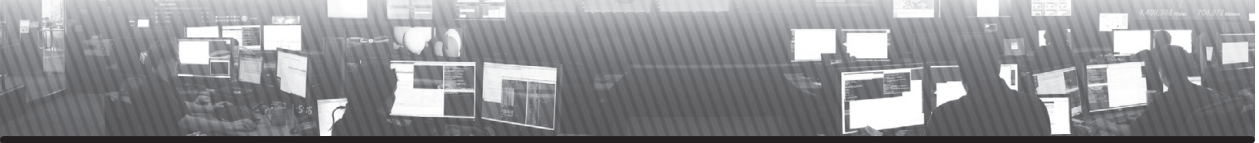


## アカマイの分散防御

DDoS攻撃で見られる膨大なトラフィックは短時間でサイトのキャパシティを消費し、パフォーマンスの低下や停止を招きます。分散型攻撃の多くは攻撃対象のリソースの消費を狙ったものとなっており、対策としてシステムのスケラビリティが決定的な要素となります。

広域分散配置されたAkamai Intelligent Platformはこのような分散化された攻撃時においても、安定した高いパフォーマンスでコンテンツを配信し続けることができます。オリジンのデータセンターから離れ、よりエンドユーザに近いアカマイのエッジサーバ上にファイアウォールを配置することで、オリジンのデータセンターに到達する前に攻撃トラフィックの分散、検知、対処が可能です(図2)。オリジンへの不要なリクエストをオフロードすることによりパフォーマンスを維持したままサイトを配信

注5) エッジサーバのことをGHostサーバとも呼びます。



し続けられるのです。

このような Akamai Intelligent Platform が内在する本質的な防御特性に加え、Kona Site Defender は次のような特徴のあるセキュリティ機能を提供します。



### Edge Cache ——アカマイ分散防御の基盤

分散化された大規模攻撃に対して有効なテクニックの1つとしてエッジサーバでのキャッシュが挙げられます。キャッシングは Akamai Intelligent Platform 上の最もベーシックな機能の1つで比較的単純な攻撃パターンに対して、非常に強力な防御となります。通常のキャッシュ機能に加え、キャッシュをすり抜けることを意図した攻撃に対しても、アダプティブ・キャッシングほかいくつかのテクニックにより、オリジンへの攻撃トラフィック到達を最小限にすることが可能です。



### WAF ——徹底された総合防御策

Kona Site Defender ではエッジサーバ上に配備された Web Application Firewall (WAF) を駆使して攻撃に備えます。データセンターの入り口やサーバの前に配置され、トラフィックのボトルネックとなりがちな一般的な WAF と異なり、アカマイの WAF は世界中に分散された 10 万台以上のエッジサーバ上で稼働することで高

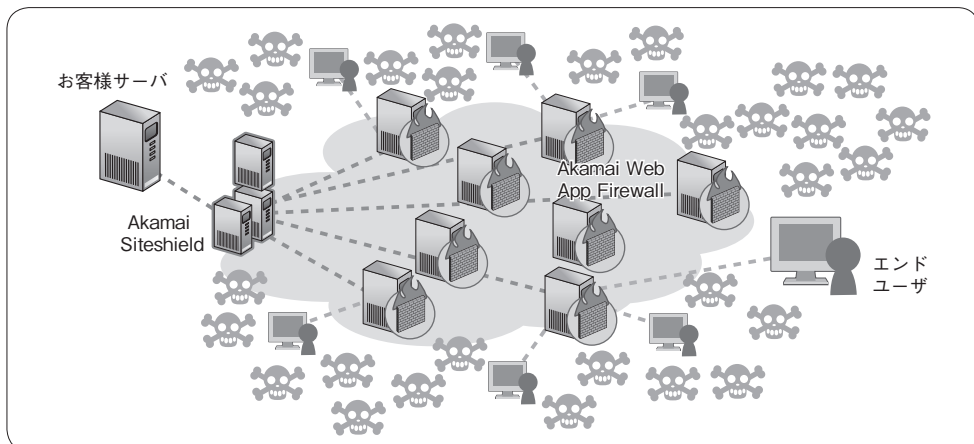
いスケーラビリティを実現しています。WAF の広域な配置により通常は 1 点に集中するパケット分析の負荷をスケールアウトし、パフォーマンスに最小限の影響で HTTP/HTTPS の攻撃トラフィックを検知、警告、ブロックすることが可能です。アカマイの WAF では、アプリケーション層制御、ネットワーク層制御、レート制御、カスタムルールの 4 つの方法で攻撃の検知と対策が行えます。

アプリケーション層制御では、さまざまなアプリケーション層の攻撃トラフィックを検知するために事前に定義された WAF のルールによるモニタリングを行います。定義されているルールで HTTP/HTTPS トラフィックのリクエスト、レスポンス、パケット内容の詳細解析 (Deep Packet Inspection (DPI)) を行い、SQL インジェクション、クロスサイトスクリプティングなどの攻撃を検知、防御することが可能です。

ネットワーク層制御では、IP ホワイトリストとブラックリストを定義できます。リストの更新は数分で全世界のエッジサーバへ展開でき、攻撃に対する素早い対応が可能になります。この機能により攻撃の発信源となる特定の IP アドレスからのアクセスを制限することができます。

レート制御では、アプリケーション層での DDoS 攻撃対策としてエッジサーバとオリジンサーバに対するリクエストのレートを WAF の

▼図2 アカマイの分散防御



カスタムルールというかたちでモニタリング、制御します。たとえば同じIPアドレスから1分間に届く500系エラーを出すリクエストの数を監視し、閾値を超えたときに警告や通信のブロックを行います。

また、標準のWAFルールでは定義されていない攻撃に対してはカスタムルールを実装することが可能です。上記のレート制御に加え、攻撃ツールの検知や新しい脆弱性の「パッチ・パッチ」としての適用が代表的な用途となります。

## Siteshield ——WAFと連動する「門番」

お客様のWebサーバ上にある貴重な情報資源を守るとともに正規ユーザからのアクセスを保証するWAFを城壁に例えると、Siteshieldはその内側の城の門の役割を果たすサービスです。

けっして複雑なものではなく、Webサーバを守るWAFアーキテクチャとおお客様のネットワーク環境の間にParent Serverと呼ばれる中間層サーバを置くことによる階層化(Tiered Distribution)がキーコンセプトとなります(図2)。

Parent Serverを置くことにより、それを「門番」として使用し、オリジンサーバをインターネットから切り離して、外部からの直接的なアクセスを阻止できます。お客様のファイアウォールにACL(Access Control List)と呼ばれるIPホワइटリストを導入し、Parent Serverや必要最低限のIP、ポートからのトラフィックだけを選択的に受け入れ、その他のソースを拒否することにより承認されていないトラフィックを効果的にブロックし、セキュリティを確保します。またACLにはお客様の定義も含められますので、徹底されたセキュリティの中で柔軟性を持たせることも可能です。

## Blackholing ——ブラックホールへの誘導

WAFとSiteshieldに加えて、さらなる階層化を行い悪質とみられるトラフィックをエッジサーバにたどり着く前に弾くサービスがあります。あらかじめ設定されたルールに該当するネー

ムサーバIPと国の組み合わせのユーザに対して、DNS名前解決の時点でエッジサーバのIPではなくLocalhost(127.0.0.1)を返し、「ブラックホール」に誘導するBlackholingと呼ばれるしくみです。これを導入するとエッジサーバにトラフィックが届く前に対処を行うため、アカマイプラットフォームへ至るラストマイルのすべての負担を軽減し、攻撃者が所属するISPやそこに接続されたほかの一般エンドユーザへの影響を最小化できます。

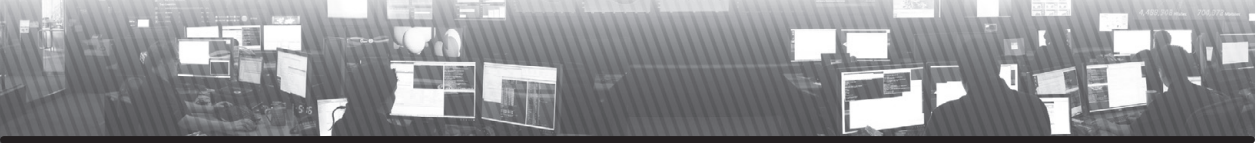
Blackholingの対象を決めるルールはネームサーバIPが基本となるため、比較的範囲が広く正規のトラフィックも該当に含まれてしまいます。通常、攻撃を受けていない場合は無効化され、攻撃が検知されたタイミングで有効化します。さらに必要以上のブラックホール誘導が発生しないよう本来対象となるネームサーバIPでもトラフィックボリュームなどの条件を使い、あらかじめ設定された割合でエッジプラットフォームに通すことができます。

Blackholingはお客様のアーキテクチャを守るWAFの外にさらにセキュリティの層を追加することで、DDoSなどの脅威を無効化できるパワフルなシステムとなっています。さらに攻撃者のPC以外の資源を浪費することなく、場合によっては攻撃者のPCをダウンさせることもできる防御策です。

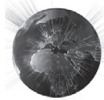
## ダイナミックマッピング ——防御壁としての側面

前節で解説したダイナミックマッピングはセキュリティ面でもとても重要なポイントとなっています。「アノニマス」などによる大規模な攻撃があった場合でも、アカマイのセキュリティ構造はそれをネットワーク内のロードバランシングにより分散し、致命的なアクセス障害を防ぎます。また、万が一特定のサーバがダウンしてしまっても、正当なユーザトラフィックは残っている無数のエッジサーバを経由してコンテンツにアクセスできます。

インターネットセキュリティには、増え続け



る脅威を遮断し自社のネットワークアーキテクチャや情報資源を守るといった目的以外に、いかなる状況でも一般のエンドユーザに快適にコンテンツにアクセスする手段を提供し、信用とサービスレベルを保守し続けるという重要な目的もあります。



## 米国政府に対するDDoS攻撃の防御事例

次に、実際にアカマイが防御を行った事例をもとに、どんな対策が行われたかを紹介します。

アカマイがこれまで受けた最大級の攻撃の1つが2009年7月4日(米国の独立記念日)に米国政府機関が受けたDDoS攻撃になります。このときに最大の攻撃を受けたサイトでは通常のトラフィックのピーク時の598倍、124Gbpsというトラフィックが発生しました。そのほかの政府機関も同様の攻撃を受け、アカマイのサービスを利用していなかったサイトでは、この攻撃によりサイトがダウンしています(図3)。

ここでは最大規模の攻撃を受けたサイトにおける対応について時系列で説明します。攻撃の始まりは7月4日、午後4時頃でした。通常、最大でも200Mbps程度のトラフィックが徐々に増え非常に短い時間で5Gbpsを超えるようになりました。アカマイとこのサイトを所管する官庁では緊急時連絡体制<sup>注6</sup>をとっているため、

注6) アカマイではDDoSに限らず攻撃発生時において、その危機レベルや内容に応じて予め連絡体制や取るべきアクションフローを取り決め、実施を行うサービスを提供しています。

この時点で連絡を行いました。サイトのダウンに至るものではないという判断のもと、攻撃について分析を行うことになりました<sup>注7</sup>。

情報収集、分析の間もトラフィックは増え続け、午後8時を境にさらにトラフィックは急増しました。午後10時50分には最大帯域の124Gbpsになり、午後11時50分にはリクエスト数が最大の795,000リクエスト/秒になりました。サイトはダウンしていないものの、膨大なリクエストが継続することで、ほかのサイトやネットワークへの影響が懸念されました。また、1つの事象に関わっている間にほかの攻撃がなされていることも考えられますが、膨大なリクエストに紛れて分析が十分になされないことも懸念されたため、このトラフィックに対しては遮断措置をとることが決定されました。

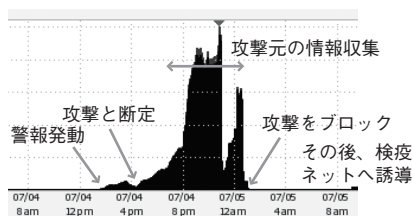
午後4時より開始した分析はおよそ8時間の間にかなりの分析がなされ、これはほぼすべてがある特定国とその近隣国から発生しており、そのリクエストを分離すればほとんど遮断できることがわかりました。

翌7月5日午前0時30分にこの特定地域のリクエストを遮断しました。これにより、遮断された地域からはサイトの閲覧ができなくなりましたが、米国内やそのほかの地域からは引き続き

注7) 当時はアノニマスが行うような事前宣告型の攻撃は少なく、攻撃発生時に発生源やその狙いかわからないケースが多かったため、分析に必要な情報を集めるために遮断よりも収集を優先させる場合があります。現在もすべての攻撃が事前宣告型ではないため、サイトに重大な被害を与えない限り、情報収集を優先する場合があります。

▼図3 2009年7月4日のDDoS攻撃概要

Akamaiがアタックを受け付けたおかげで顧客サイトの99.9%の帯域を削減



Customer	Peak Traffic	X Times
US Government Customer 1	124 Gbps	598
US Government Customer 2	32 Gbps	369
Financial Customer 1	26 Gbps	110
US Government Customer 3	9 Gbps	39
US Government Customer 4	9 Gbps	19
US Government Customer 5	2 Gbps	9
US Government Customer 6	1.9 Gbps	6
US Government Customer 7	729 Mbps	Site Down

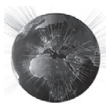
き閲覧でき、ほかのシステムへの影響もなくせる状態になりました。

いったん遮断された特定地域も同9時半には別インフラを経由して閲覧できるように開放をし、この後、一定期間誘導しました。これは通常アカマイのお客様が利用する高速化、最適化がなされたネットワークではなく、より遅いネットワークになります。この結果DDoS攻撃が行われてもネットワーク帯域が十分にならないため、たどり着かないかたちを取ることができます。

ここでは国／地域レベルで遮断するという制御の例でしたが、実際にはビジネスの都合上遮断という手段を取れない場合も発生します。図4は2012年9月のアカマイのあるお客様のトラフィックを、9月11日、18日、25日の3日分のみをプロットしたのになります。11日と25日は日本からのアクセスが95%以上を占めていますが、18日には中国からのアクセスが30%程度占めていることがわかります。同じデータを国ごとに表しますと、図5のようにほかの国／地域からはほぼ等分されたアクセスが来ているのに対して、中国からのアクセスは18日に集中していることがわかります。

重要なことは、このサイトのように通常のビジネスでも日頃からアクセスがある国や地域のトラフィックは単純に遮断できない、ということです。そして、これを逆手に取った攻撃も多くあります。そんな場合にもキャッシュの機能

やトラフィックの誘導などによりお客様のサイトはビジネスを継続できなければなりません。



## 終わりに

最後に今後の課題について少し述べたいと思います。



### ①新種の攻撃への対応

アカマイでは過去10年以上に渡り、DDoS攻撃を含むサイバーテロの手法を分析し、効果的に防御する方法を見だし、それをKona Site Defenderに実装してきました。今日では観測されるDDoS攻撃のほとんどのケースにおいて防御メカニズムが有効に作用し、お客様企業を攻撃から守ることができています。しかし攻撃手法は日々進化しているため、従来とはまったく異なる手法で攻撃されるかもしれません。新種の攻撃の手法分析、対応策導入を継続して実施していく必要があります。

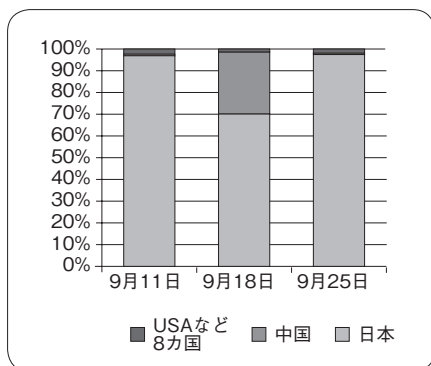


### ②あいまいなリクエストへの対応

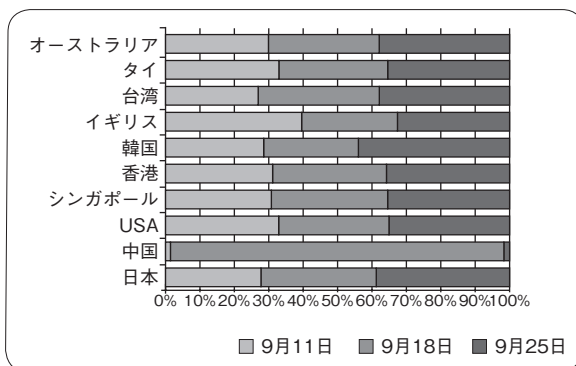
上記と関係しますが、実際にDDoS攻撃に直面した際に個々のリクエストについて遮断してよいものであるかどうか、という判断が困難な場合があります。

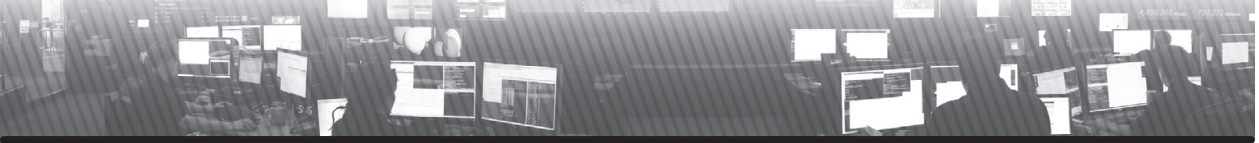
リクエストの特徴から、急増したリクエストがDDoS攻撃用ツールで生成されていることを

▼図4 トラフィックの推移



▼図5 各国のアクセスの傾向





識別できるような場合、アカマイではそのリクエストを無条件に遮断することをお客様企業にお勧めすることができます。

一方、たとえば新種の攻撃手法ではそのような特徴をつかむことができず、正規のユーザリクエストであるかどうかアカマイ側で簡単に識別できないケースがあります。そのような場合には、お客様のビジネス理由により正規のユーザリクエストが増加する要因の有無を確認いただき、最終的にはリクエストを止めるか否かの判断をお客様に求めることになります。判断を誤ると正規のユーザリクエストを止めてしまうため、慎重な判断が必要です。

極端なケースですが、ある企業ではDDoS攻撃を含むすべてのリクエストをお客様と考え、それに対してコンテンツを提供することを使命としているため、「アカマイでリクエストを止

めることなく、オリジンサイトが耐えられる限りコンテンツ提供してほしい」とのご意見をいただいたこともあります。「なるべくなら止めたくない」「わからないものは止めない」と考える企業が多いことは当然のことでしょう。

アカマイでは個々のトラフィックがDDoS攻撃によるものであるか否かをより効果的に判定するための知見を積み重ね、方法論を確立していきます。

DDoS攻撃に限らずあらゆる種類のサイバーテロはインターネットで自由な経済活動を発展させようとする人々に対する挑戦であり、そのような存在を許すことはできません。アカマイでは継続した技術開発により「インターネットを信頼できるもの」にし、お客様が安心してビジネスに活用できるよう努めてまいります。**SD**

## COLUMN

### アカマイの攻撃トラフィック情報収集

一般的なWebサイトではユーザの特性や購買意欲向上のためにサイトのアクセス解析を行っています。

アカマイサービス利用後は、インターネットユーザはアカマイサーバへアクセスを行い、必要に応じてオリジナルのサイトへアクセスを行います。よって、ユーザのアクセス解析の大部分は、アカマイが取得するアクセスログにて実施可能です。同様に、アカマイサービスを利用している世界中の著名なサイトへのアクセス状況は、アカマイにて把握可能であるため、俯瞰的な立場で世界中のWebトラフィックのトレンドを把握しています。

小売り、製造、公共サービス、広告／メディアなどの業種、業態別に年単位でのトラフィック統計情報を分析することで、トラフィックの伸び率からインターネットの活用度合が上昇している業種／業態を把握したり、国別のインターネット利用量から、

ブロードバンドの普及度合いを把握したりできます。これら特定のお客様の情報を含まない一般的なインターネットトラフィックの報告書(SOTI: state of the internet)を四半期に一度公開しています<sup>※8</sup>。

SOTIには、アカマイユーザのWebサイトを標的としたサイバー攻撃などのセキュリティ関連のレポートも含まれています。執筆段階で最新のSOTIによると、2012年第1四半期中、アカマイでは182カ国を発生源とする攻撃トラフィックを観測しました。攻撃トラフィックの発生源は中国がトップで、観測された攻撃トラフィック総数の16%を占めています。

アカマイはHTTPのみならず、HTTPS/SSL経由でのセキュアなコンテンツに対するリクエストも処理しているため、クライアント側のSSL暗号方式のトレンドもSOTIにて公開しています。

注8) <http://www.akamai.com/stateoftheinternet/>

# Emacs 64bit化計画!

使いやすいエディタ環境を  
作りませんか!



第5回

## IMEの実装と環境整備

太田 博志 OOTA Hiroshi ● TwitterID @h2oota イラスト: 黒崎 玄

暑いさ中に始まった本連載も5回目を迎えました。いつ頃からかは覚えていませんが、新しいことに取り組んで失敗したときのロールバックを億劫に感じるようになり、それを避けていたように感じます。連載を始めてから、硬直した脳が無意識で逃げていたことを、意識的にねじ伏せてみることにしています。5回目の今回はIME関連の実装と作業効率向上のための環境整備について解説していきます。



### IME再変換、前後フィードバックの実装

GNU Emacs (Win64) ではIMEの機能である再変換と前後フィードバックを実装しています。ユーザから見ればMS Officeで可能なことはできて当然と考えるのは自然ですし、ユーザーインターフェースの機能の欠落はストレスを増大させます。キーバインドの問題は解決不可能とあきらめるにしても、そのほかの部分では可能な限り妥協はしたくありません。実装にあたってはEmacsの構造と相容れない点があり、GCまで及ぶ大きな修正となってしまいました。



### 再変換機能とは



文字の変換直後に **[Ctrl] + [BackSpace]** を押すことで確定前の状態に戻すのが再変換です。変換直後でない場合の基本関数(w32-ime-perform-reconversion)は作成してありますが、UIが未作成で隠し機能になっています。リージョンを設定してM-:で再変換できます。



### 前後フィードバックとは



前後フィードバックもかなり古くからある機能で、IME98には搭載されていたようです。サードパーティソフトでは2007年くらいに秀丸が初めて対応した機能だったと思います。

前後の文脈をヒントにして精度の高い変換を行います。リスト1を入力したあとで行末にカーソルがある状態で「あつひ」を入力してみてください。

リスト2のように文脈を読んで正しく変換できているはずです。



### 実装の困難さ



本来、再変換機能や前後フィードバック機能はアプリケーション側はIMEからの要求に対しカーソルの前後の文字列を渡すだけで実現できます(ドキュメントが皆無に近いので返し方の調査がたいへんです)、Emacsでは簡単には実装できませんでした。それはEmacsの構造に起因しています。EmacsはGUI機能を別スレッドに

#### ▼リスト1 前後フィードバックの例①

夏は  
本は  
お湯は

#### ▼リスト2 前後フィードバックの例②

夏は暑い  
本は厚い  
お湯は熱い

分け、Windowsのメッセージ処理はこのGUIスレッドが受け持っています。このマルチスレッド化のおかげで重い処理を実行中もウィンドウの移動などをスムーズできます。

しかし、その反面プログラミングが難しくなります。EmacsLispのインタプリタやメモリアロケータはEmacsのメインスレッドで行われます。GUIを受け持つスレッドからはLispデータの変更はおろか、参照もできません。メモリアロケータも利用できないので、EmacsLispデータを生成することも許されていません。両スレッド間の通信も厄介です。両スレッド間の通信はWindowsメッセージを介して行われていますが、メインスレッドからGUIスレッドへはSendMessageを使った同期メッセージ、GUIスレッドからメインスレッドへはPostThreadMessageを使った非同期メッセージを使っています。同期メッセージではメッセージ処理で値を返すことができますが、非同期メッセージはメッセージキューにメッセージを積んだあと、メッセージ処理の完了を待たずに制御が戻るのを値を返せません。つまり、GUIスレッドからメインスレッドが管理するLispデータを参照できないのです。

## 🐾 解決策の探求

Windowsはスレッド間の通信サービスをいくつか提供していますが、今回はユーザモード非同期プロシージャコール(APC)を使うことにしました。APCは同一プロセス内でのリモートプロシージャコールです。QueueUserAPCというAPIを使って実行要求をキューに積み、クライアント側は[Msg]WaitForMultipleObjectEXで実行要求が完了するまで待ち、サービス側は同じくWaitObject系のAPIを呼ぶことでAPC関数を実行できます。

メインスレッドでは非同期プロセスやキーボード入力の処理のためにWaitObject系のAPIを使用していますし、GUIスレッドではWindowメッセージの処理のために呼び出しているGet

MessageをMsgWaitForMultipleObjectに変更すれば小さな修整でメイン、GUIの両スレッドから双方向で呼び出し可能な通信路を作成できました。

ここまでで、EmacsLispのデータをGUIスレッドで持つことが可能になりましたが、EmacsLispのデータはGC(ガベージコレクタ)の管理下にあります。

Emacsではコンサバティブ(保守的)GCを採用しています。コンサバティブGCはスタックを走査してオブジェクトへのポインタを探します。C言語レベルではメモリ上のどの領域にポインタが格納されているか把握していますが、GCはその情報は得られません。スタックの現在位置からトップまでスキャンしていき、オブジェクトへのポインタと同じビットパターンが見つかったら安全側に倒し、そのオブジェクトは参照されているとみなします。これが保守的と呼ばれる所以です。

GCはGUIスレッドのスタックもスキャンする必要があるので、APCを用いてGUIスレッドもGCを行うことにしました。



## プログラミング環境を整える

後半ではEmacsのパッケージ管理システムpackage.el、Emacs上のテキストブラウザw3m、ソースブラウジングサポートパッケージのascopeの紹介をします。



## package.elの場合



Emacs24に搭載されている標準のパッケージ管理システムです。GNUの公式リポジトリELPAのパッケージは登録方法の煩雑さが原因であり充実していませんが、Emacs24に標準で搭載されてからmarmaladeやmelpaといったサードパーティのリポジトリサイトができてきています。今回はpackage.elの利用方法を紹介します。

# Emacs 64bit化計画!



## 🐾 インストール

package.el を <http://bit.ly/pkg-el23> からダウンロードします。

```
emacs -batch -f batch-byte-compile package.el
```

でバイトコンパイルし、c:/Program Files/GNU/emacs23/site-lisp にコピーし、初期化ファイルに次を記述します。

```
(require "package")  
(package-initialize)
```

これだけでGNUの公式リポジトリが参照できますが、ここはあまり充実していません。MarmaladeとMelpaを追加します。

```
(add-to-list 'package-archives  
  '("marmalade" . "http://marmalade-  
  repo.org/packages/"))  
(add-to-list 'package-archives  
  '("melpa" . "http://melpa.milkbox.net/  
  packages/"))
```

ただし、marmaladeとmelpaは登録が容易なこともあってマルウェアが混入している可能性も否定できません。EmacsLispパッケージはアンチウイルスソフトウェアの検査対象とはなっていないのでアンチウイルスに頼ることはできません。

EmacsLispパッケージはソースで提供されていることもあり、やみくもに心配する必要はありませんが、Emacsを乗っ取られた場合はPCで可能なすべての操作を実行されてしまう危険性があることを覚えてください。新規パッケージの導入は慎重に行いましょう(原稿執筆時点は遠隔操作ウィルスが大騒動になってました)。

## 🐾 利用法

M-x list-packages でパッケージ一覧を取得します。入手するパッケージ名の上で **[Enter]** キーを押すと詳細を記述したウィンドウが開くので Install のうで **[Enter]** を押すとパッケージのダウンロードとインストールが始まります。設定方法はこのページに表示されることになってい

るので、指示に従って初期化ファイルの書き換えを行います。

初期化ファイルに追加した部分を eval-region などで評価すればEmacsの再起動なしにインストールしたパッケージを利用できるはずですが、まれにEmacsの再起動が必要な場合もあります。

インストール済みのパッケージはパッケージ一覧の下のように表示されています。名前の上で **[Enter]** を押すとアンインストールできそうなのですが、まだ成功したことがありません。~/emacs.d/elpa の下にあるパッケージフォルダを削除して、初期化ファイルから設定削除、そしてEmacsの再起動でしのいでいます。



## w3mの場合



Emacs上でテキストブラウザであるw3mを利用するパッケージです。HTML取得からレンダリングまでをテキストブラウザのw3mで行い、表示とインターフェースをEmacsが行います。JavaScriptやCSSを駆使したUIは利用できませんが、Emacsを離れずにWebページを閲覧できるので結構重宝します。

## 🐾 インストール

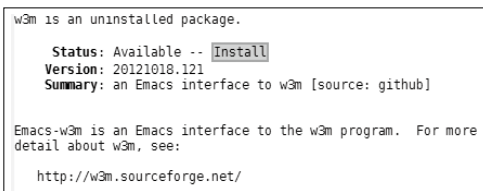
w3mのサポートOSはUNIXであり、ソースで公開されています。w3mのWindows用バイナリを探しましたが、cygwinのものしか見つかりませんでした。cygwinは動作させるためにたくさんのdllをインストールする必要があり面倒です。自分でコンパイルするしかありません。

このような場合に便利なのがMinGWを使ったクロスコンパイルです。MinGWの解説はここでは行いませんが、詳しく解説しているサイトも多いので興味のある方は検索してみてください。

コンパイル済みのバイナリを筆者のWebページ「w3m-w32」<sup>注1)</sup>に置きましたので、コンパイルが面倒な方はダウンロードしてください。自分

注1) w3m-w32 (<http://hp.vector.co.jp/authors/VA052357/w3m-w32.html>)

## ▼図1 w3m.elのインストール



でコンパイルするための手順も同じページにあります。w3m.exeを適当なディレクトリに置きパスを.emacs(初期化ファイル)でw3m-commandにセットします。

```
(setq w3m-command "c:/path_to_w3m/w3m.exe")
```

なお、このw3mはw3m.elで利用するためにコンパイルしたもので、Windows用の変更は行っていない。w3mはtermcapを介したエスケープシーケンスで表示を行いますが、64ビット版のWindowsにはansi.sysが提供されていないので、このw3m.exe単独でブラウズできません。



## w3m.elの場合



w3m.elはpackage.elを利用してインストールできます。M-x list-packagesでパッケージの一覧の中からw3mを探し、w3mの上でリターンを押します。図1のようなウィンドウが開くのでInstallのうえでリターンを押し、確認メッセージに答えればダウンロード、インストールが行われます。

## 🐾 利用方法

よく使うコマンドを表1に抜粋しました。もう少し詳しい説明が公式ページ<sup>注2</sup>にあるので参照してください。

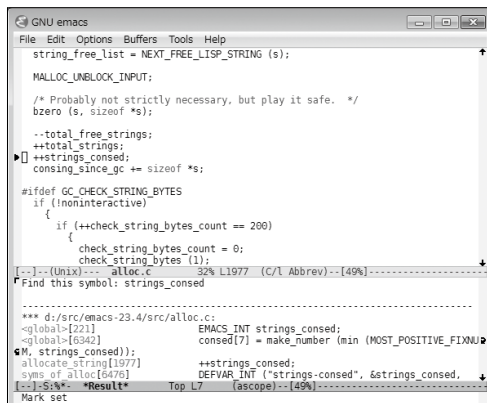


## ascope



EmacsでC、C++のソースを編集する場合に関数や変数の定義へのタグジャンプが可能になります(図2)。この手のパッケージは数多くあ

## ▼図2 ascope



## ▼表1 w3mの頻出キーコマンド

キー	動作
[B]	前のページ
[N]	次のページ
[H]	ホームページ
[^]	一階層上(親ディレクトリ)
[Space]	順スクロール
[BackSpace]	逆スクロール
[<]	左スクロール
[>]	右スクロール
[Tab]	次のアンカー
[Shift] + [Tab]	前のアンカー
[a]	現在のページをブックマーク
[M] - [a]	カーソル位置のURLをブックマーク
[v]	ブックマークページへ移動
[g]	URLへ移動またはキーワードを検索
[G]	新規タブを開きURLへ移動
[R]	リロード

り、標準でもCEDETが搭載されていますが、あまりうまく動きません(筆者の使い方が悪いだけかもしれませんが)。現在は簡単設定で使えて、解析も高機能なascopeに落ち着いています。ascopeはソースブラウジングツールCscopeをバックエンドに使用します。CscopeにもEmacsのフロントエンドが用意されていますが、検索の都度Cscopeを起動するようでサクサクと検索ができません。ascopeは非同期プロセスとして起動したCscopeにクエリを渡し、検索を行うので高速に動作します。

注2) w3m.el(<http://emacs-w3m.namazu.org/>)



## ▼リスト3 ascopeの設定ファイル

```
(add-hook 'c-initialization-hook
  #'(lambda ()
    (require 'ascope)
    (let ((map (make-sparse-keymap)))
      (define-key map "d" 'ascope-find-global-definition)
      (define-key map "g" 'ascope-find-global-definition)
      (define-key map "s" 'ascope-find-this-symbol)
      (define-key map "t" 'ascope-find-this-text-string)
      (define-key map "c" 'ascope-find-functions-calling-
this-function)
      (define-key map "C" 'ascope-find-called-functions)
      (define-key map "i" 'ascope-find-files-including-file)
      (define-key map "a" 'ascope-pop-mark)
      (define-key c-mode-map "¥C-cs" map)
      (define-key c++-mode-map "¥C-cs" map)
      (define-key ascope-list-entry-keymap "q"
        #'(lambda ()
          (interactive)
          (ascope-clear-overlay-arrow)
          (delete-window))) )))
(autoload 'ascope-init "ascope" "" t)
```

## 🐾 インストールと設定

CScopeはwin32用のcscope-win32が開発されているのでプロジェクトページ<sup>注3</sup>からダウンロードして、zipからバイナリを展開してパスの通ったディレクトリに置きます。ascopeはpackage.elからインストールできます。

CScopeには指定ディレクトリ以下のソースファイルのインデックスを簡単に生成するためのcscope-indexerというシェルスクリプトが付属しています。同等の機能のバッチファイルをgist<sup>注4</sup>に上げておきました。筆者の設定をリスト3に示します。

## 🐾 利用方法

### ①インデックスの生成

コマンドプロンプトを開き、ディレクトリに移動しcscope-indexer [-r]で実行します。-rは下位ディレクトリもインデックス生成の対象とします。

### ②ascopeの起動

M-x ascope-initと入力すると、ディレクトリ

注3) cscope(<http://code.google.com/p/cscope-win32/>)

注4) cscope-indexer(<https://gist.github.com/4107985>)

が要求されるので、先ほどこインデックスを生成したディレクトリを指定します。

### ③ソースファイルを開く

ascope-init実行以前に開いたファイルではascopeは利用できません。いったんkill-bufferしたのち開きなおしてください。

### ④シンボルの定義にジャンプする

ソースファイルを開いて定義を参照したいシンボルの上に移動します。**Ctrl** + **C**、**Ctrl** + **S** + **d** で定義の一覧が別ウィンドウで開くので**p**、**n** で上下に移動し

てください。連動してソースファイルも表示されます。**Enter** でそのファイルに移動、**q s w** で一覧を閉じ元のファイルに戻ります。

表2は筆者の設定での操作法の一覧です。



## まとめ

今回は、GNU Emacs (Win64版)の再変換、前後参照変換の実装に際しての問題点と解決した方法とpackage.elの導入からw3m、ascopeの導入、利用までを解説しました。それではまた次号で! **SD**

▼表2 キー配列による操作方法一覧

キー	動作
<b>C</b> - <b>c</b> <b>s</b> <b>d</b> , <b>C</b> - <b>c</b> <b>s</b> <b>g</b>	シンボルの定義を検索
<b>C</b> - <b>c</b> <b>s</b> <b>s</b>	シンボルの参照を検索
<b>C</b> - <b>c</b> <b>s</b> <b>t</b>	テキストとして検索
<b>C</b> - <b>c</b> <b>s</b> <b>c</b>	関数呼び出しを検索
<b>C</b> - <b>c</b> <b>s</b> <b>C</b>	その関数が呼び出している関数を検索
<b>C</b> - <b>s</b> <b>s</b> <b>i</b>	インクルードしているファイルを検索
<b>@</b>	ジャンプ前の位置に戻る

## Column JScriptをバッチに埋め込む

cscope-indexerではJScriptをバッチに埋め込み、バッチから呼び出して正規表現の処理を行っています。JScriptを埋め込むためのポイントは先頭行です。

```
@if (0==1) never execute
```

この行はバッチファイルとして実行される場合は@付のコマンド、すなわちエコーされずに実行されるif文で条件式は(0==1)、これは文字列"0"と"1"が等しいという条件です。等しくないの never executeは実行されずに次の行以下を実行していきます。そしてバッチファイルの最後の方でCScript //Nologo //E:JScript "%script%" <"%LIST\_FILE%.tmp" >"%LIST\_FILE%"として、バッチファイル自身

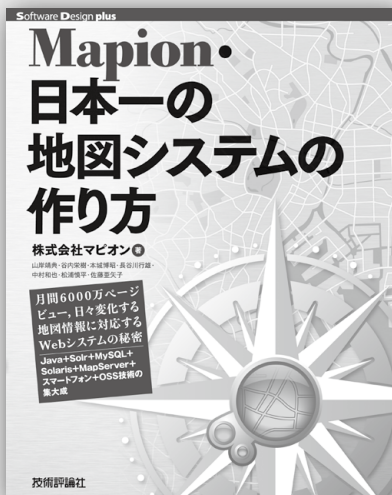
をCScriptに渡し、結果をリストファイルとしてリダイレクトします。

このバッチファイルがJScriptとして実行される場合、1行目はJScriptの機能である、条件コンパイルの指定として解釈されます。条件式(0==1)は0と1の比較ですが、やはり偽なので条件コンパイルは成立せず。条件コンパイルの指示命令である@end行まで読み飛ばされます。結局@end行より後にある部分がJScriptとして実行されるというしくみです。

ネタ元は、どこで見たのか記憶が定かではありませんが、技術系のblogだったと思います。@ifをバッチとJScript両方で正しく解釈させ処理を切り分けるアイデアに感嘆しました。

## Software Design plus

技術評論社



株マピオン、山岸靖典、谷内栄樹、本城博昭、長谷川行雄、中村和也、松浦慎平、佐藤亜矢子 著  
B5変形判 / 256ページ  
定価2,709円(本体2,580円)  
ISBN 978-4-7741-5325-4

大好評  
発売中!

# Mapion・ 日本一の 地図システムの 作り方

日本オリジナルの地図サービスを開発し、月間6000万PVを超える莫大なアクセスを捌くWebサービスを運営するマピオン。本書は同社の技術を余すことなく公開する。「大量アクセスを実現するWebシステムノウハウ」を軸にオープンソースの地図システム構築や、検索システムの構築などを紹介する。前人未踏のシステムゆえ、手探りで開発を進めざるを得ず、悪戦苦闘・試行錯誤するも、その記録もリアルに記述。これらは、システムが抱える問題に対してブレイクスルーするためのヒントになるであろう。3.11の震災対応の地図サービス開発の裏話も公開!

こんな方に  
おすすめ

・Javaエンジニア  
・Webサービス開発者  
・Web運用管理者  
・検索システム開発者

# テキストデータならお手のもの 開眼👁️シェルスクリプト

(有)ユニバーサル・シェル・プログラミング研究所 <http://www.usp-lab.com>  
上田 隆一 UEDA Ryuichi [Twitter @uecinfo](https://twitter.com/uecinfo)

## 第13回

# メールファイルを操作する —grepでリストを作って一気に処理

### はじめに

今回は前回に引き続きメールをシェルスクリプトでさばっていきます。今回の内容は、CUI端末やシェルスクリプトでたくさんのファイルを操作したり、1つのファイルを切り刻んだりするための小技、大技が入り乱れていますので、「メールなんぞ興味ない」という方も注目です。おさえておかないとマウスで数千のファイルをプチプチと操作するハメになりますよ!!

一重積んでは父の為 二重積んでは母の為...<sup>注1</sup>

### 前回のおさらい

前はMaildirにたまったメールを日別にディレクトリに整理するという課題を扱いました。図1のように、ホーム(/home/ueda)下のMAILというディレクトリに日別にディレクトリを作り、各ディレクトリの下にメールを置きました。また、メールをUTF-8に変換したものを作り、<日付>.utf8というディレクトリに置きました。今回はこの状態から、条件抽出してメールを

注1) 「賽の河原地蔵和讃」より。眠れなくなるので知らない人は調べないほうが良いです。

振り分ける方法と、添付ファイルを抜き出す方法を扱います。

### メールの振り分け

このメールアドレスには世界中から雑多な情報が送られていますが、ここから条件を満たすメールを集めてみましょう。

例として、サーバ管理者ならお馴染みのLogwatchからのメールを抽出し、特定のディレクトリに置くという操作をしてみましょう。LogwatchはCentOSなどをインストールすると、とくに設定をしなくてもroot宛にサーバ監視結果のメールを毎日送ってくるツールです。

#### ▼図1 ~/MAIL/下の様子

```
MAIL/の下には日付のディレクトリ
$ ls
20120610
20120610.utf8
20120611
20120611.utf8
<略>
日付のディレクトリには、それぞれのメールが置かれる
$ ls 20120610/ | head -n 3
1339304183.Vfc03146017dM943925.abc
1339305265.Vfc03146062cM458553.abc
1339306807.Vfc0314607c6M993984.abc
<日付>.utf8 には、UTF-8化した同名のファイルがある
$ ls 20120610.utf8/ | head -n 3
1339304183.Vfc03146017dM943925.abc
1339305265.Vfc03146062cM458553.abc
1339306807.Vfc0314607c6M993984.abc
```

#### ▼リスト1 Logwatchからのメール

```
##### Logwatch 7.3 (03/24/06) #####
Processing Initiated: Sun Oct 14 04:00:02 2012
<略>
```

Logwatchから送られてくるメールは、リスト1のような書き出しで始まります。見たことがある人も多いでしょう。

このメールの送信メールアドレスは**From: logwatch@<サーバ名>**となっており、各メールのヘッダに書いてあります。蛇足ですが、メーラーはメールのヘッダを読み込んで、「Subject:」や「From:」などの行を読んで件名や送信者をGUI出力しているだけで、メールはあくまで単なるテキストです。

メールを振り分けるには**From: logwatch@...**の行をgrepで抽出して、grepの出力するファイル名を使ってファイルをどこかにコピーすれば良いでしょう。例として、ホスト名をオプションに指定したら、LOGWATCH\_<ホスト名>というディレクトリに当該ファイルをコピーするシェルスクリプトをリスト2に示します。

8行目から12行目で、引数をチェックしたり保存先のディレクトリを作ったりしています。**&&**や**||**については以前から何回か出てきていますが、**&&**は左側のコマンドが成功(終了ステータスが0)だったら右側のコマンドを実行します。**||**はこの逆で、左側のコマンドが失敗(終了ス

テータスが0以外)だったら右側のコマンドを実行します。

15行目のmkdirの**-p**オプションは、すでにディレクトリがあってもエラーにならないように指定しています。一方で、パーミッションなどの理由でディレクトリが作れないときはしっかりエラーを出してくれます。

17行目から23行目の処理をひとことで言うと、全メールに対してSubjectを調べて**\$1**で指定したホストのLogwatchなら、ディレクトリLOGWATCH\_<ホスト名>にファイルをコピーしています。Logwatchのメールは1日1通送られてくるので、コピーしたファイル名を日付にしています。

18行目のgrepのオプション**-F**ですが、これは正規表現を使わないときに指定するオプションです。メールアドレスにドット(.)が入っていて、そのままgrepすると「任意の1字」を示す記号扱いされてしまうので、**-F**を指定しました。**grep -F**と同義のfgrepというコマンドもあります。

21行目のawkを通ったあとの文字列をリスト3に示します。これを23行目のxargsに通すこ

#### ▼リスト2 指定したホストのLogwatchからのメールを振り分けるシェルスクリプト

```
01 #!/bin/bash -vx
02 #
03 # LOGWATCH: 指定したホストのlogwatchメールを収集
04 # usage: ./LOGWATCH <hostname>
05 #
06 # written by R. Ueda (r-ueda@usp-lab.com)
07
08 [ "$1" = "" ] && exit 1
09
10 server="$1"
11 dir=/home/ueda/MAIL
12 dest="$dir/LOGWATCH_$server"
13
14 cd "$dir" || exit 1
15 mkdir -p "$dest" || exit 1
16
17 echo ???????.utf8/* |
18 xargs grep -F "From: logwatch@$server" |
19 awk -F: '{print $1,substr($1,1,8)}' |
20 #1:ファイル名 2:日付
21 awk -v d="$dest" '{print $1,d "/" $2}' |
22 #1:コピー元 2:コピー先
23 xargs -n 2 cp
```

とでリストの1列目のファイルが2列目のファイル名でコピーされます。

リスト3の「xyz」は、実際にはもっと長い文字列ですが、誌面では煩わしいので短縮しています。以後も「xyz」で置き換えます。

では、実行して、ちゃんと動いたか確かめてみましょう。図2に実行例と結果を示します。

もし複数のサーバからLogwatchのメールを受け取っているならば、ホストのリストを作ってシェルスクリプトLOGWATCHを繰り返し適用すれば、Logwatchのメールを振り分けることができるでしょう。

## 添付ファイルを抽出する

次は大技です。メールから添付ファイルを抽出します。図3は、準備したサンプルメールをGmailで見たところです。サンプルメールには画像ファイルが2つ(イラストと大きなデジカメ写真)添付されています。



### 添付ファイル付きメールの中身を見てみる

毎度のこと大雑把ですので、詳しくは別の資料を見ていただきたいのですが、添付ファイルがあるときのメールのフォーマットについて説明します。まず図3のメールについて、実物(つまりテキストファイル)を見てみましょう。lessで見ると図4のような構造になっているのがわかります。といっても、7万7千行もあるので見

るのはたいへんですが……。

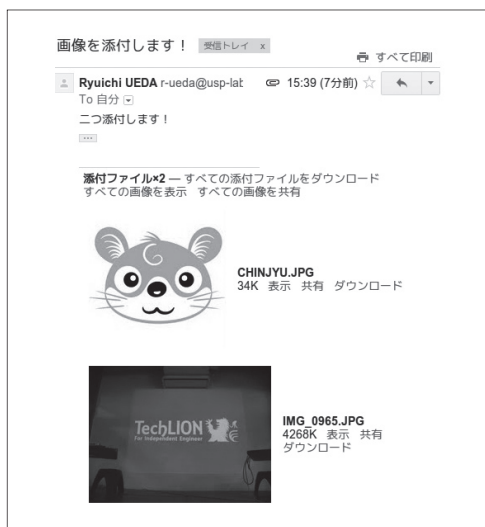
このテキストの中に、--047d7b621ee6cf83c604cc276bb3という行がいくつかあって、どうやら区切り文字(境界)になっているようです。

これは「MIMEマルチパート」と呼ばれる形式です。MIMEマルチパートにはいくつか種類がありますが、1個以上の添付ファイルが含まれたテキスト形式のメールは、何か特殊な状況でなければmultipart/mixedという種類になります。今回はこいつだけ相手にしましょう。

添付ファイルをメールから抽出するには、ヘッダのboundaryで指定された文字列(境界文字列)で挟まれた領域から中身を抽出します。リスト4は、CHINJYU.JPGに関係する部分です。

この部分は空行をはさんで上側にファイルの

▼図3 サンプルメール(添付ファイル2個付き)



▼図2 LOGWATCHの実行

```
$ ./LOGWATCH abc.usptomonokai.jp 2> /dev/null
$ ls LOGWATCH_abc.usptomonokai.jp | head -n 3
20120611
20120612
20120613
$ grep "^From:" ./LOGWATCH_abc.usptomonokai.jp/* | head -n 2
./LOGWATCH_abc.usptomonokai.jp/20120611:From: logwatch@abc.usptomonokai.jp
./LOGWATCH_abc.usptomonokai.jp/20120612:From: logwatch@abc.usptomonokai.jp
```

▼リスト3 21行目のパイプを通る文字列

```
20120611.utf8/1339354818.xyz.abc /home/ueda/MAIL/LOGWATCH_abc.usptomonokai.jp/20120611
20120612.utf8/1339441214.xyz.abc /home/ueda/MAIL/LOGWATCH_abc.usptomonokai.jp/20120612
```

## 抽出とデコードを シェルスクリプト化

10行目～18行目は、メールデータのContent  
Type:multipart/mixedの行から境界文字列を

リスト5の22、23行目のawkはメールファイルを境界で切って中間ファイルに保存する処理

```
$ echo あはははは | base64
44GC44Gv44Gv44Gv44GvCg==
$ echo あはははは | base64 | base64 -d
あはははは
```

```
$ less ./20121016/1350369599.xyz.abc
(ヘッダ。略)
Content-Type: multipart/mixed; boundary=047d7b621ee6cf83c604cc276bb3

--047d7b621ee6cf83c604cc276bb3
(メール本文。文字化け)
--047d7b621ee6cf83c604cc276bb3
...ひたすら記号が続く...
--047d7b621ee6cf83c604cc276bb3
...ひたすら記号が続く...
--047d7b621ee6cf83c604cc276bb3--
7万7千行もある
$ wc -l ./20121016/1350369599.xyz.abc
77342 ./20121016/1350369599.xyz.abc
```

[illegible]

## テキストデータならお手のもの 開眼 シェルスクリプト

です。このawkにはいろいろポイントがあります。正直言って、ややこしいです。

まず、awkの-vオプションは本連載で何回か紹介していますが、bashの変数をawkの変数に事前に代入するためのものです。ここでは境界の文字列と切り出し先の間ファイル名の一部を、

それぞれ**b**と**f**という変数に代入しています。

if文中の**\$0~b**は変数**b**を正規表現扱いして、**\$0**(行全体)と比較する式です。変数を右側に持ってくるときは、/は不要です。

そして、知らない人には一番わけがわからない**print > f a**ですが、実は>は不等号ではなくリダイレクトです。

**print**で行全体を出力し、その出力先を**f a**にしています。**f**は中間ファイル名の一部(/tmp/<プロセス番号>-F)で、**a**は境界文字列が見つかった1つずつ増える数字です。awkでは文字列と数字を並べるとそのまま文字列として連結するので、リダイレクト先(中間ファイル名)は、/tmp/<プロセス番号>-F<数字>となります。

25~36行目は、中間ファイル(分割されたデータ)から添付ファイルを復元する処理です。27行目のgrepでContent-Dispositionの行(添付ファイル名が含まれている行)を抽出します。図3のメールを通すと、27行目のgrepの後ろのパイプにはリスト6の文字列が流れます。

これを見ると「3560-F0と3560-F1はどこにいった?」ということになりますが、3560-F0はメールのヘッダ、3560-F1は本文でContent-Dispositionという文字列がないのでこの時点で弾かれます。もしContent-Dispositionで始

### ▼リスト5 添付ファイル抽出シェルスクリプト

```
01 #!/bin/bash
02 #
03 # EXTFILE: メールから添付ファイルを抽出する。
04 # usage: EXTFILE <電子メールファイル>
05 # written by R. Ueda (r-ueda@usp-lab.com) Oct. 16, 2012
06
07 [ "$1" = "" ] && exit 1
08 tmp=/home/ueda/tmp/$$
09 dest=/home/ueda/MAIL/FILES
10 #####
11 #境界文字列を抽出
12 grep -i '^Content-Type:' "$1" |
13 grep "multipart/mixed" |
14 #最初にあるもの(=ヘッダにあるもの)だけ処理
15 head -n 1 |
16 sed 's/.*boundary=//' |
17 #[""]がくっついている場合があるので、取って変数に入れる
18 tr -d '"' > $tmp-boundary
19
20 #####
21 #境界でファイルを分割
22 awk -v b="^--$(cat $tmp-boundary)" -v f="$tmp-F" ¥
23 '{if($0~b){a++;}print > f a}' "$1"
24
25 #####
26 #分割したデータ(中間ファイル)から添付ファイルを作る
27 grep -i '^content-disposition:' $tmp-F* |
28 #1:grepの結果から中間ファイル名と添付ファイル名を抜き出す
29 sed 's/^[^:][^:]*:.*filename=([^.]*)/¥1 ¥2/' |
30 #1:中間ファイル名 2:添付ファイル名
31 tr -d '"' |
32 while read a b ; do
33     #抽出、デコード、出力
34     sed -n '/^$/,$p' "$a" |
35     base64 -d > "$dest/$(basename $1)_${b}"
36 done
37
38 #作ったファイルを表示
39 ls $dest/$(basename $1)*
40
41 rm -f $tmp-*
42 exit 0
```

### ▼図6 Content-Typeの大文字小文字バリエーション

```
$ grep -i '^content-type:' ./*.utf8/* |
awk -F: '{print $2}' | count 1 1 | sort | sm2 1 1 2 2
Content-Type 41367
Content-type 75
content-type 9
```

## 終わりに

今回は前回に引き続き、電子メールを扱いました。気づいた人は少ないと思いますが、grepを起点としてファイル进行操作するためのリストを作るという処理が、メールの振り分け、添付ファイルの操作の両方で行われてきました。これは覚えておく便利なテクニックです。慣れておくと、実際にファイル进行操作する直前まではテキスト処理になるので、whileの中でcpやmvの前処理をするよりもデバッグが楽になります。また、立ち上がるコマンドの数も減らすことができます。

添付ファイルの抽出では、バイナリデータを扱いました。これは知らない人が意外に多いのですが、バイナリデータに対してリダイレクトやcatをしても、データが壊れることはありません。base64などのテキストとバイナリを橋渡しするコマンドがあれば、バイナリデータをシェルスクリプトでシームレスに扱うことができます。これは次々回あたりに扱ってみたいと考えています。

今回は、これまでの応用で「CUIおれおれメール」でも作ってみようと思います。SD

まる行があれば添付ファイル扱いされますが、ゴミが出るだけですので良しとしましょう。もし気になるのであれば、while文の中でチェックします。

リスト5、29行目のsedはgrepの出力から中間ファイル名と添付ファイル名を抽出しています。こうすることで、後ろのwhile文に入出力するファイル名を与えています。

whileの中は、34行目のsedでファイルの中身部分を取り出し、35行目のbase64で添付ファイルを復元しています。sed -n '/^\$/,\$p'は、「空行以降をプリントせよ」という意味になります。sed -n '<開始行>,<終了行>p'で、ファイルからある範囲を行単位で出力する処理ができるので、これは丸暗記しておくといいでしょう。34行目のように行の指定には正規表現や最終行を表す\$などの記号が使えます。

35行目のbase64で気になるのは、ちゃんと1ビットも変わらずファイルを復元してくれるのかということですが、これは大丈夫です。EXTFILEを実行して、できたファイルを添付した元のファイルと比較してみましょう。図7がチェックの結果ですが、大丈夫ですね。

### ▼リスト6 リスト5の27行目のパイプを通る文字列

```
/home/ueda/tmp/3560-F2:Content-Disposition: attachment; filename="CHINJYU.JPG"
/home/ueda/tmp/3560-F3:Content-Disposition: attachment; filename="IMG_0965.JPG"
```

### ▼図7 リスト5のスキプトの実行と添付ファイルのチェック

```
$ ./EXTFILE ./20121016/1350369599.xyz.abc
/home/ueda/MAIL/FILES/1350369599.xyz.abc_CHINJYU.JPG
/home/ueda/MAIL/FILES/1350369599.xyz.abc_IMG_0965.JPG
元のファイルと比較
バイナリファイル(テキストも)を比較するときは、
diffではなくcmpを使います
$ cmp ./CHINJYU.JPG ./FILES/1350369599.xyz.abc_CHINJYU.JPG
$ echo $?
0
$ cmp ./IMG_0965.JPG ./FILES/1350369599.xyz.abc_IMG_0965.JPG
$ echo $?
0
```



iPhone

OSアプリ開発者の知恵袋

A p p l i c a t i o n D e v e l o p e r s

第33回

## セルフブランディング のすすめ

スマートフォンの認知度を一般に広めただけでなく、ソフトウェア開発においても新しい波を作り出してしまったiOS。開発者たちは何を見、どう考えているのか。毎回入れ替わりでiOS向けアプリケーション開発に関わるエンジニアに登場いただき、企画・開発のノウハウやアプリの使いこなし術などを披露してもらいます。

リオ・リーバス Leo Rivas  
Twitter @StudioLoupe

### アプリ開発者への転身

いまでこそiOSアプリによる収益で生計を立てている筆者(屋号：スタジオルーペ)ですが、個人開発者としてここに至るのは容易なことではありませんでした。

筆者がiPhoneアプリ開発をはじめたのは2008年の11月、iPhone(3G)が日本で発売され、App Storeがはじまって間もない頃です。それまで、筆者はアプリ開発やプログラムなどとはまったく関係のない絵本作家を目指していたのですが、生活のために工事現場で肉体労働の仕事をしていました。絵を描きながらも、短時間で十分稼げる仕事だったからです。しかし、絵本作家になるのもそう簡単なことではなく、ただただ時間が過ぎていくばかりでした。

そんなとき、日本でiPhoneが登場し、App Storeのはじまりに新たな可能性を感じたのです。このデバイスで自分の作品を発信していこうと思い立ち、プログラム経験はまったくありませんでしたが、勉強してでもできるようになって収益を得てやろうと決意したのです。

### アプリは誰でも作れる! けど……

もともとプログラム知識ゼロだった筆者ですが、iPhoneアプリを作ろうと思い立ってからApp

Storeで販売を開始するまでの期間はおよそ2ヵ月でした。この期間には開発者登録の手続きや、基礎中の基礎の勉強期間と、最後にAppleにアプリを提出して待つ審査期間などもすべて含まれているので、実際のアプリそのものの開発期間は1ヵ月もかかっていなかったのではないかと思います。つまり、iPhoneアプリを開発するハードルは決して高くはないのです。やる気さえあれば誰にでもアプリを作ることとは可能と言っても大げさではないでしょう。今後、開発ツールがどんどん便利になれば、今以上に誰でもアプリを作れるようになるはずです。

何が言いたいのかというと、アプリを「作った」時点では個人開発者としてやるべきことの全体の流れから見れば、ただ「スタートラインに立った」だけに過ぎないということです。本当に他のアプリとの差が出るのは作った後にどうするかで決まります。

App Storeには現在65万以上のアプリがあると言われていています(2012年、第3四半期業績報告より)。それだけ多くのアプリがある中で、個人も企業も同じ土俵で勝負をしなければいけないのですから、ただ作ってみた、とりあえずストアに出してみた、ではどんなに凄いアプリを作っても人目に付かないまま埋もれてしまう可能性が高いでしょう。アプリを生かすか殺すかはすべてプロモーション／マーケティングにかかっていると言っても過言ではありません。本稿ではそのことについて書かせていただこう

と思います。

## セルフブランディング

セルフブランディングとは、個人が自らを、そして自分の作品をブランド化し、プロモーションができる力をもつということです。App Storeには無料アプリだけでも20~30万ほどのアプリがあります。無料だけでもそれだけ多くのアプリがある中、知名度も宣伝力もない開発者がアプリを売るのは想像以上に難しいことです。では、どうやってアプリを売れるようにしていくのでしょうか？ どうしたら安定した収益を生み続けることができるのでしょうか？ 筆者はセルフブランディングしか道はないと考えます。

### 土台づくり

まずは無料アプリでヒットを狙うことを考えるべきでしょう。1つのヒット作がブランドの基盤となり、その無料ヒットアプリを土台にすれば、その上にどんどんヒットを積み上げやすくなります。無料ヒットアプリがあれば、次の作品を出したときに宣伝することができるからです。アプリから別のアプリへとつなげる「架け橋」となるわけです。

有料アプリでは大々的に広告などを入れると嫌がられてしまうので、宣伝はデリケートに扱わなければなりません。無料アプリなら広告のはもはやあたりまえ。だからこそ個人開発者にとって最大の武器(宣伝ツール)になる無料ヒットアプリを作ることが大事なのです。

### 長く収益を生み続ける無料アプリ

売り切りの有料アプリと違って、無料アプリは使われ続けている限り、収益を生み続ける可能性があります。たとえダウンロード数が減り、ランキングから消えてしまっても、使われ続ける限り長い期間収益を生み続けるのです。

有料は短期的に見れば確かに無料よりもハイペースで収益を生み出しますが、常に売れ続けるアプリでなければ安定した収益を生むことはできません。一時的にランキング上位に駆け上がったヒットアプリでも、ランキングから落ちてしまえばまったく売れなくなります。無料広告モデル、もしくは無料からの課金モデルが今のアプリで長く安定した収益を生む鍵であるのは間違いないでしょう。

短い時間でいかに多く儲けるかが重要になる有料アプリを短距離走と例えるならば、無料アプリは安定したペースでどれだけ長く収益を生み続けることができるかのレース、つまりマラソンなのです。

### 自らが情報の発信源になることの重要性

アプリの宣伝にお金をかけた場合、どのくらいの効果が得られるのか。筆者はお金を払って宣伝をしたことは少ないのですが、某有名iPhone情報サイトで1ヵ月間99ドルの広告を出したときの成果と、自分のアプリ内の広告枠から得たデータを比較してみました(表1)。

▼表1 99ドルWeb広告 vs. アプリ内広告 (FusionCalc Lite)

	月額	表示回数	クリック回数	平均クリック率
Web広告 (有名情報サイト) 	99ドル (現在は299ドルから)	170万回/月	1,455回	0.08%
アプリ内広告 (FusionCalc Lite) 	0ドル	90万回/月	7,000~9,000回	0.7~1.0%

まず、99ドルで載せた広告は月間で170万回ほど表示され、合計1,455回クリックされました。クリック率にすると0.08%です。筆者の運営しているブログ内に載せている広告もクリック率は平均0.07%と非常に近い数字であるため、Web広告のクリック率は実際こんなものだろうと考えられます。Webの場合は画面が大きく、画面内の情報量も多いため、たとえ170万回表示されたといってもほとんどの場合人の目に留まることすらいらないのが現実だと思われます。それに、クリックされたからといってそのまま購入につながるというわけでもなく、実際にはまたその中の一部の人が購入してくれるかどうかという話です。85円などの安いアプリだと10%以上の購入率がなければ元がとれない計算ですから、0.08%のクリック率というのはかなり厳しいことがわかります。筆者の場合も、広告を出したことでそれほど大きな効果を得たと感じることはできず、割と短期間でやめざるを得ませんでした。

では次に、筆者の無料アプリ「FusionCalc Lite」の中に表示している広告(図1)のデータを見てみましょう。

月に90万回と、表示回数は某サイトよりも低いのですが、クリック数は7,000から多いと9,000回近いときがあります。良いときだと1%

▼図1 FusionCalc Liteに表示されている広告(画面上部)



前後のクリック率があり、Web広告よりもアプリ内に表示される広告のほうが10倍ほどクリックされやすいということがわかります。

このデータだけで結論とするのは早計かもしれませんが、もしもお金を払っ

て自分のアプリを宣伝する場合、Webサイトよりもアプリ内広告に出稿するほうが効果的と言えるのではないのでしょうか。

もちろんベストなのは自分に無料ヒットアプリがあり、一銭も払わずに自社広告を発信できるようになることです。自分から情報を発信する術がなければ、新作を出すたびにゼロからプロモーションをすることになり、それにはお金も時間もかかるでしょう。メディアに紹介してもらうことに希望を抱くのも他人任せになってしまうので、自分でコントロールが一切ききません。

そういう意味で、長い目でみたら、まずはなにかしら無料アプリをヒットさせ、自力での発信力を育てることに専念することがとても重要なのがわかると思います。かけるお金があるのであれば、無料アプリを宣伝して育てることを考えるのも1つの手でしょう。

## 必ずやらなければいけない5つのこと

ここまで無料アプリでヒットを生むことの重要性について書いてきましたが、そもそも無料アプリをヒットさせるにはどうするのかという根本的な部分が抜けていますね。これは簡単なことではないですし、こうすればヒットするという絶対的な法則もありません。ただ、個人開発者としてやれることのすべてはやらなければと考えます。その最低限やるべきことをここから紹介していきます。

### ■プレスリリースを送る

有名どころだとAppBank、Touch Lab、meet iなどがあります。これらのサイトは有名だけに簡単には取り上げてもらえませんが、当然出しておいたほうが良いです。また有名どころだけではなく、できるだけ多くのサイト、そして個人ブロガーさんたちに記事を書いてもらえよう願う積極性も大切です。

## ■YouTubeを活用する

スクリーンショットしか見ることができない App Store では、購入するための判断材料としていまいちもの足りないと思うユーザも多いようで、実際に動いているところを見たいがために YouTube でアプリ名を検索することがしばしばあるようです。このとき見つけた動画が良ければ最後の一押しになるでしょうし、そもそも動画がなければ買う気持ちも失せてしまう恐れもあります。

動画はプロモーションビデオ以外にも、チュートリアルや、そのアプリを使ってどう日常が変わるか、もしくは単純に使っていて楽しんでいる様子を収めたビデオなど、多ければ多いほど良いでしょう。もちろん、質も高ければ高いほど良いのですが、それ以上にアプリの凄さ、面白さを伝えられているかどうかが一番肝心だと思います。

## ■人脈を作る

どこの世界も同じですが、人脈を作るということは大事なことです。自分一人の力には限界がありますが、人と人とのつながりから生まれる力は計り知れません。iPhone ユーザが集まるイベントや開発者が集う勉強会などに積極的に参加し、いろいろな人と知り合い、つながりを作ることで得することはあっても損をすることはないでしょう。

## ■SNS(ソーシャルネットワーク)を活用する

Twitter と Facebook ページは絶対に力を入れるべきです。それ以外にも余裕があれば、幅広く SNS で友達やファンを増やすことが個人にとって大きな強みになります。Twitter ならフォロワー1万人、Facebook ページで1000「いいね」くらいあればかなりの影響力と宣伝力を持つことになるのではないのでしょうか。

当然、それほど簡単なことではありません。筆者も実際そんなに多くいませんが、目標としてはそのくらいを目指して頑張ればなんとかな

るのではないかと感じています。とはいえ、お金でフォロワーを買ったり、闇雲にフォロワーを増やしても意味はないので、こればかりは地道にファンを増やしていくしかありません。量よりも質が大事なのは間違いありません。

## ■ブログを運営する

SNS でファンを作るためには、アプリを作る以外にも何らかの活動をしていなければなかなか難しいと感じています。ここでブログの役割が重要になってくると思います。勝手にアプリが爆発的にヒットしてくれればそれだけでファンを増やすことも可能でしょうが、それが多くの場合簡単じゃないからこそ別の方向からもアプローチをしていく必要があります。

筆者がブログをはじめたのは2009年の7月、最初のアプリをリリースして約半年後のことです。はじめた理由は、やはり少しでも多くの人に自分のアプリを知ってもらうきっかけを作るのが目的でした。開設当初は1日数人の人が訪れるかどうかといった感じで、なんの宣伝にもなるわけがありません。ほとんど誰も読まないブログを更新し続けるのは正直しんどいものです。しかし、どんなブログでも知名度上げるには時間がかかって当然と自分に言い聞かせ、とにかく地道に書き続けようと決心しました。

3年以上続けた今は、そこまで人気とは言えないのですが、書けばだいたい1,000人以上が読んでくれるブログにはなりました。ブログ経由から SNS のフォロワーが増えたり、アプリの販売数も増えましたし、雑誌などの取材を受けることにもつながりました。



こうして、アプリ、SNS、ブログの3つが支え合うような形で、個人開発者としてなんとか生活が続けていくことができる体制を築くことができたのです。このどれか1つでもなければ、これだけ長くアプリ開発者を続けてくることはできなかったのではないかと感じています。

## セールという宣伝方法

さて、ここでもう少し具体的な宣伝方法とその効果についての話をしましょう。さんざん無料アプリでヒットを生むことの重要性を話してきたわけですが、それは無料アプリだけを作り続ければ良いということではありません。実は有料アプリを出すこともとても大事なことです。

有料アプリには無料アプリでは絶対に真似できない宣伝方法があるからです。それはセールです。無料はそれ以上値下げができないので、一度ダウンロードが伸び悩むと、なかなか挽回する切っ掛けが作りづらいというウイークポイントもあります。有料アプリの強みは、セールで話題作りができることにつきます。

### セールは大胆に

忘れてはいけないのは、プロモーションは大胆さが肝になるということです。450円から350円に値下げをしても、そこまでのお得感を感じることはできません。自分がユーザの立場になって考えればわかりやすいのですが、いざ少しでも多く儲けようと考えてしまうと、ついつい控えめな価格設定にしてしまいがちです。

しかし、思い切って最低ラインの85円など

にしたほうがダウンロード数は確実に伸びるため、ランキングも上がりやすくなり、「アプリの宣伝」という目的はこちらのほうが遥かに効果的と考えられます。App Storeでは毎日のように大量のアプリがセールになるので、その中で目立つことができればセールをしても不発に終わってしまうということ覚えておきましょう。

### セールの期間

セール期間はよほど波に乗らない限り、2〜3日で戻すのがちょうど良いでしょう。あまり長いと、セールの勢いは落ちます。セールから元の価格に戻ったときにも売れ続けるための勢いを残しておくことが大切です。もしも最初の2日か3日までにセール効果を感じることができなければ、それ以上待っても希望は薄いと考えるべきです。そのときは一刻も早く値段を戻したほうが損失を抑えられます。

### 短期間限定「無料化」

無料化にはそれなりのリスクがあるものの、うまくいくときは大きなりターンが返ってきます。そもそも、せっかく作ったアプリを無料ではらまいてしまうというのは、なかなか勇気が必要な行動です。無料期間中、そのアプリを有料で売っていれば入ってきたであろう収益を失うリスクがあるわけですから、損をする覚悟が必要です。

そして、無料化をする目的を明確に持ちましょう。単純にそのアプリをより多くの人に知ってもらいたいのか。それとも別のアプリへつなげるための架け橋にしたいのか。それによって事前に準備しなければいけないことなどもあります。

たとえば別のアプリへつなげるのであれば、無料化するアプリの中には宣伝したいアプリへのリンクなどを入れておいて、そのアプリのことを知って

▼図2 無料化の際に組み込んだ新バージョンへの広告



もらえるようにしなければいけません。筆者が以前、「フュージョン計算機(FusionCalc)」というアプリを無料化したときは、新バージョンの「FusionCalc+」の広告が起動時に1回だけ表示されるようなしくみを事前に組み込んでおきました(図2)。これによって無料化したときに、無料でダウンロードした人たちに新作の情報を知ってもらうことができました。このように、無料化をするにしても、しっかり計画的に行うとより良い結果が得られます。

## 無料セール成果

さて、無料セールをして一体どれだけの成果が得られたのか。図3のようなグラフにしてみました。このときは1日限定の無料化ではありましたが、その1日で世界で1万3,000ダウンロードを達成しました。セールをする前は旧作の「フュージョン計算機」と新作の「FusionCalc+」を合わせて週に平均120ダウンロード程度だったアプリですが、無料にしたらたったの1日で1万3,000ダウンロードもされるのですから、無料と有料アプリのダウンロード数の差は歴然です。

そして無料セールの効果により、この2つのアプリはその後1週間でだいたい400ダウンロードされました。セール前の平均的ダウンロード数と比べると3.3倍増えたことになります。注目したいのは、無料化したアプリ自体も有料に戻ってしばらくは普段より多く売れたということです。

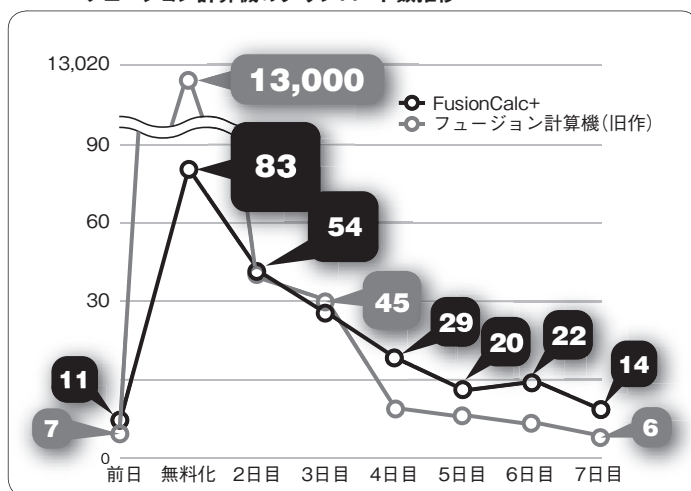


## 総合収益4倍

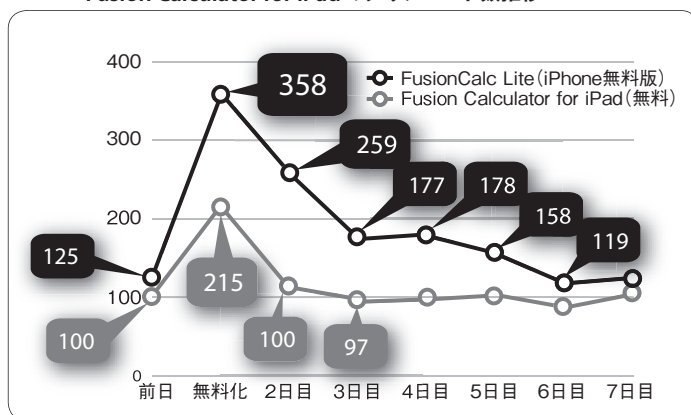
無料セールの効果はそれだけにとどまりません。フュージョン計算機(FusionCalc)にはもと iPhone/iPad 両方に無料版(FusionCalc Lite/Fusion Calculator for iPad)があるのですが、それらのダウンロード数も3倍ほど増えました(図4)。これによって広告収益もピーク時で普段の2倍にまで増えました。

さらに、FusionCalc シリーズとは関連性のない、筆者が今までリリースしてきた別のアプリのダウンロード数まで増える効果もあり、結果的にその月の総収益は数ヵ月さかのぼった

▼図3 フュージョン計算機の無料セール1週間におけるFusionCalc+とフュージョン計算機のダウンロード数推移



▼図4 フュージョン計算機の無料セール1週間におけるFusionCalc LiteとFusion Calculator for iPadのダウンロード数推移



平均月間収益の4倍ほどになりました。無料にすることで損どころか収益を4倍増やすことも可能だという事例です。

うまく火がつけばもっと多く増やせるポテンシャルだってあると思います。もちろん、これは1つの成功例にすぎず、まったく効果を得られなかった無料セールも何回か経験していますので、やれば当たるというほど甘いものでもありません。しかし考え方によっては、最悪収益はゼロになるかもしれませんがマイナスにはならないので、ハイリスク・ハイリターンというよりは、ローリスク・ハイリターンだと筆者は考えます。

## ▶ 世界的プロモーション

もう1つ、無料化はどのような形で火がつくか予想にくいものですが、FusionCalcの場合は結果的に世界的プロモーションとしても大きな成果を出しています。積極的な宣伝は日本でしか行わなかったのにも関わらず、ダウンロード数1位はサウジアラビアの4200、2位がアメリカの3300、そして日本は3位で3000だったのです。今までリーチできなかった中東や米国市場でそれだけ多くの人たちに知ってもらった掛けになったというのも大きな意味があるのではないかと思います。

直接このセールの影響であったという証拠はないのですが、このセール後まもなくして、米国の教育機関から有料版が50ダウンロードまとめ買いされました。これはボリューム購入プログラムと言って、教育機関のみが、アプリを最低20ダウンロードすることを条件に50%引きで買うことができるしくみで、セールスレポートにそのような購入があったことが記載されていたのです。



無料セールがどのような効果をもたらすかはやってみないとわかりません。時に想像以上の成果につながることもあるのです。なので筆者は無料セールの「損」と考えないようにしていま

す。心理的には損と感じてしまいがちですが、iOSデバイスの累計販売台数は2012年6月までで世界で4億台を突破したと発表があり、この勢いだと2015年までにiTunesアカウント数は10億人を超える見通しとまで言っている調査報告もあります<sup>注1)</sup>。これだけの規模の市場で、たったの1万人に無料でアプリを配ったところで、買い手はまだまだいくらでもいると考えられるわけです。少しでも人から人に情報が広がるようにアプリの知名度を上げるためには、1万と言わず10万人でも100万人にでも無料で配布したいくらいの気持ちでいなければいけないのだと感じています。

## セールスのタイミング

セールのタイミングは非常に難しいです。たとえばセールといえばクリスマスだったり大きなイベントのときに合わせると良さそうだと思いますが、他の多くの開発者も同時にセールをするタイミングでもあるので、返って目立たなくなる可能性のほうが高いかもしれません。

筆者も最近、iPad miniが発売されたタイミングでFusionCalcのiPad版をはじめて無料化したのですが、3日間で2,000ダウンロード程度で終わりました。大きなイベントに合わせるのは必ずしもベストなタイミングとは限らないということです。とはいえ、いつセールを行うにしても、どこかのメーカーが人気タイトルのセールをするかなどはまったく予測不能なので、ぶつからないことを祈るしかありません。ここぞという勝負のセールのときは、事前にメディアにプレスリリースを送るという手も有効でしょう。

## 妻のブログ

実は自分のブログとは別に、個人開発者である筆者のアプリを支えるもう1つのブログがあ

注1) <http://www.asymco.com/2012/09/17/projecting-ios-devices-through-itunes-account-growth/>

ります。それは妻が数ヵ月前にはじめた「AppMama - iPhone アプリ開発者の妻のブログ<sup>注2)</sup>」です。

はじめた当初はやはり、自分も経験したように少しずつ読者を増やしていくしかないと考えていましたが、予想以上に人気が出たことで、はじめて1ヵ月もしないうちに1万PV(ページビュー)を達成。翌月には月間3万PVと、どんどん勢いにつき、筆者のブログよりも訪問者が多いブログになったのです。おかげでスタジオルーベのアプリの収益は2倍以上増えることとなりました。ブログの立ち上げから筆者も相当力を入れて盛り上げてきましたが、妻の努力なくしてこの結果を生み出すことはできなかったもので、本当にありがたく思っています。

このように、ブログも成功させればアプリを売るうえで十分な戦力になりうるのです。アプリを作りながらTwitterやFacebookなどSNSの活動をしつつ、さらに定期的にブログを更新するというのは楽ではありません。ただ、個人開発者としてやっていくというのはそういうことなのです。作ることから売ることまですべてを個人レベルでやらなければいけないということを忘れてはいけません。そして自らがブランドとなり、積極的にアプリを宣伝していかなければいけないのです。



筆者は3年、完全に個人で開発を続け、4年

目に突入するタイミングで、自分一人で続けていくことの限界を感じはじめました。そんなときに声をかけられて一緒にアプリ開発をはじめたのが、いたのくまんぼう(前号のこの連載で執筆)氏です。

実は、いたのさんとの出会いもブログを通してでした。筆者が書いた『飛躍の1年: さよなら工事現場』という記事を読んで興味をもってくれたことから一度会うことになり、そこで意気投合。その1年後には2人でGimmiQというブランドを立ち上げ、顔の動きでページをめくられる電子書籍リーダーアプリ「MagicReader」を共同開発したのです。

これによって筆者は自分の得意とするデザインやプロモーションに専念することができ、もともとプログラマーであるいたのさんもプログラムに専念でき、結果、作品としてのクオリティは何倍にも磨き上がりました。そして、MagicReaderは国連主催のアワードを受賞するほど世界中で注目されるアプリとなりました。

ですが、1人が2人になっても個人レベルの開発者であることには変わりありません。そういう意味では今後も大きな企業などと混ざって同じフィールドで戦い続けなければいけないので気を抜くことはできません。年々勢いを増し成長し続けるApp Storeは今後、今まで以上の激戦区になると予想されます。開発者同士、情報交換をしたり支え合いながら力を合わせて生き残ることができたらと思います。皆さん頑張りましょう！SD

注2) <http://www.appmama.net>

## ● リオ・リーバス

個人ではスタジオルーベ、共同開発はGimmiQとして、いたのくまんぼう(@Kumanbow)氏とiOSアプリ開発を中心に活動中。個人の代表作は数字を指でドラッグ&ドロップ保存できる「フュージョン計算機(FusionCalc)」,そしてGimmiQでは顔の動きでページをめくられる電子書籍リーダー「MagicReader」を開発し、国連の主催するWSA Mobile(ワールドサミットアワードモバイル)で賞を受賞。

Twitter @StudioLoupe スタジオルーベ URL <http://www.studioloupe.com> GimmiQ URL <http://gimmiq.net>



## 第33回

マルチプラットフォーム  
開発環境を使ってみよう(2)

モバイルデバイス初のオープンソースプラットフォームとして、エンジニアから高い関心を集めるGoogle Android。いち早くそのノウハウを蓄積したAndroidエンジニアたちが展開するテクニクや情報を参考にして、大きく開かれたAndroidの世界へ踏み込もう！

よこいど／日本Androidの会 横浜支部  
嶋崎 聡 SHIMAZAKI Satoshi  
Twitter @sato\_c  
motosumi64@gmail.com



## はじめに

横浜支部の嶋崎です。前回は、マルチプラットフォーム環境にはどのようなものがあるかを紹介しました。UnityやPhoneGapのように注目度も高く、さまざまな場面で使われているものがあります。そういった環境も含めてスクリプト言語を使って開発するメリットとデメリットおよびRubyが使えるRhodesについて紹介していきます。

マルチプラットフォームの  
開発言語

前回は、マルチプラットフォーム開発環境で使う言語などをまとめて紹介しました。おもにCやC++、C#での開発が可能ですが、それ以外にスクリプト言語も使えます。中でもJavaScriptやLuaが使える環境が大半です。

JavaScriptは現在のWeb開発では主流の言語と言えます。AndroidのWebViewを表示に使う環境ではHTMLが使われるので、こうした環境と相性が良いというのもあるでしょう。マルチプラットフォーム環境では、PhoneGapやUnityで採用されています。

Luaはその歴史が意外に古く、Lua本体の移植性の高さも手伝ってさまざまな環境に使われています。PC用アプリケーションやPlayStation 3やXbox 360用のゲームにも採用されています。

マルチプラットフォームでは、UnityやCoronaで利用できます。

こうしたスクリプト言語とライブラリを使ってWebアプリが手軽に作れるようになっている現在では、スマートフォン環境のネイティブアプリを作成する環境が出てくることは当然の流れではないでしょうか。そういった部分を踏まえて、スクリプト言語での開発のメリット、デメリットについてまとめてみます。

スクリプト言語での  
開発のメリット

## ■開発時間の迅速化

コンパイラを通す必要があるほかの言語とは違い、スクリプト言語ではインタプリタですぐに実行して結果を見られます。結果をすぐに知ることができるので、トライ＆エラーにも時間をかけられるようになります。

もちろん、機能がある程度定まってきたら、モジュール化すること、テストを書いて品質を上げることが必要ですが、そこへ向かうまでの時間を短縮できるという意味では、スクリプト言語で開発するメリットは十分にあります。

## ■デバッグのコスト

AndroidとiOSでそれぞれアプリを作るとします。機種間で共通する部分と機種に固有の部分がでてきますが、それぞれの環境でデバッグをしていくと期間も手間もかかるうえ、修正し

忘れることもあるでしょう。結果的に開発の効率が悪くなります。

そこで先にAndroidで共通部分をデバッグして動作するものになったら、機種固有部分と組み合わせることで全体のデバッグの工数を減らせませう。



## スクリプト言語での開発のデメリット

### ■すでにある資産を利用できない

各OS用のネイティブなライブラリがすでにあったりあらためて作った場合でも、それが必ずしもスクリプト環境で利用できるとは限りません。

Androidに関して言えば、スクリプト側にIntentを使う手段が実装されていれば、ライブラリをアプリ化してインストールしてもらい、呼び出せるかもしれません。しかし、複数のアプリに分かれることで利便性はなくなります。

### ■スクリプトの学習コスト

すでにJavaやObjective-Cを使え、さらにOS間の差異を吸収できるような環境が整っている場合、スクリプト言語を覚えなくてはならないなら従来の方法で作ったほうが早いこともあります。

ただ、データ加工のような単純作業はちょっとしたスクリプトを書くだけで手作業が減らせませう。普段使っていないのであれば、何か1つでもスクリプト言語を覚えるとよいかもしれません。

### ■メンテナンス

既存のフレームワークを利用していく場合、何かのきっかけでそのフレームワーク自体が仕様を変更されることもあります。また、脆弱な部分に対しての変更がないとは言い切れませう。こうした変更にとどこまで耐えられる作りにしているかによって、その後の手間が変わるでしょう。

自分たちでフレームワークを作ってメンテナンスしながら使っている場合は当てはまりませう。そうでない場合には、どのような環境にも当てはまる話です。

また、プログラマが少ない、もしくは、その言語を覚えている人が少ない場合は、環境を選

ばないとメンテナンスできないまま使い続けるといったことにもなりかねませう。このあたりは、どのような規模で使われるものなのかを判断する必要はあるでしょう。



## Rubyでの開発

スクリプト言語とひと口で言っても、すでに紹介したJavaScriptやLua以外にもPythonやPHP、Rubyなど、いろいろとあります。

JavaScriptやLuaは、すでにいくつもの環境で採用されており、解説書もたくさんあります。こうした環境の紹介は他に任せるとして、最近増えてきてるRubyを使った環境を紹介します。なぜRubyかという、筆者が最近仕事でRubyを使うことが多くなってきて、その便利さと書きやすさを実感しているからです。

Rubyを用いた開発環境はAndroidならruboto<sup>注1</sup>、iOSならRubyMotion<sup>注2</sup>が活発にバージョンアップを重ねています。とくにRubyMotionはFacebookに日本の方々によるグループ<sup>注3</sup>もあります。そして、複数のスマートフォンアプリの開発に使えるのが今回紹介するRhodesです。



## Rhodesとは

Rhodesは、米モトローラ社が公開しているオープンソースの開発環境です<sup>注4</sup>。

Rubyを主軸として、node.jsやjQueryといったオープンソース環境を組み合わせる開発環境を構築しています。WebアプリではおなじみのMVC(Model View Controller)アーキテクチャを採用しています。通常のアプリの作り方とは少し勝手が違いますが、Ruby on Railsのようなフレームワークと考え方は同じです。

それからEclipseをベースとしたRhoStudio

注1) [URL](http://ruboto.org/) http://ruboto.org/

注2) [URL](http://www.rubymotion.com/) http://www.rubymotion.com/

注3) 「RubyMotion JP」[URL](https://www.facebook.com/groups/149315595198329/) https://www.facebook.com/groups/149315595198329/

注4) [URL](http://www.motorola.com/Business/US-EN/RhoMobile+Suite/Rhodes) http://www.motorola.com/Business/US-EN/RhoMobile+Suite/Rhodes



というIDE<sup>注5</sup>も提供されています。コマンドラインでの開発も可能ですが、RhoStudioを使うことで普段と同じ感覚でアプリを開発できます。

## Rhodesのインストール

詳細なインストールページが提供されています<sup>注6</sup>ので、そちらを参照してもらうのがよいのですが、元が英語のサイトで資料も英語のものしかありません。設定する項目が多く、RhoStudioをインストールしたら終わりとはならないため、そういった点を含めてまとめました。

### Rubyのインストール

まずRubyです。現時点で最新の1.9.3系が必要になります。Windowsでは、ビルドした環境によって数種類の実行ファイルがありますが、本家Rubyサイトのダウンロードページ<sup>注7</sup>を参考にして、自分の環境に合ったものをダウンロードしてください。

MacではJewelryBox<sup>注8</sup>というRVM(Ruby Version Manager)のGUI版があります。こちらをインストールして、Rubyのビルドや複数バージョンの切り替えを行うとよいでしょう。

Linuxの環境ではそれぞれのパッケージをイ

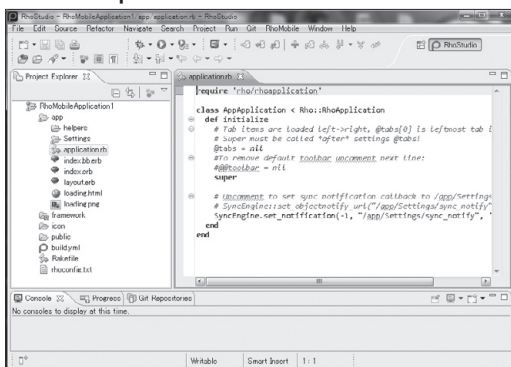
注5) [URL](http://www.motorola.com/Business/US-EN/Business+Product+and+Services/Software+and+Applications/RhoMobile+Suite/RhoStudio) http://www.motorola.com/Business/US-EN/Business+Product+and+Services/Software+and+Applications/RhoMobile+Suite/RhoStudio

注6) [URL](http://docs.rhomobile.com/rhodes/install) http://docs.rhomobile.com/rhodes/install

注7) [URL](http://www.ruby-lang.org/ja/downloads/) http://www.ruby-lang.org/ja/downloads/

注8) [URL](http://unfiniti.com/software/mac/jewelrybox) http://unfiniti.com/software/mac/jewelrybox

### ▼図1 RhoStudioを起動したところ。お馴染みのEclipseがベース



ンストールするか、コマンドライン版のrvmか本家Rubyサイトからソースコードをダウンロードしてビルド、インストールします。

Android SDKとNDK、iOS用にはXcodeをインストールしましょう。その他のスマートフォンに関しては、それぞれの環境に応じたライブラリをインストールします。

ここまでの準備が整ったら、次はRhoStudioのインストールです。

### ■Windows環境でのインストール

まず、Rhodesのインストールページ(前述の注6を参照)からRhoStudioのパッケージをダウンロードして実行します。パッケージにはIDEやRhodes本体も含め、アプリの構築に必要なものがすべて含まれています。インストール後は、いったん再起動してください。

後は[スタート]メニューから[Motorola Rho Mobile Suite]の[Motorola RhoStudio]を起動すれば利用できます(図1)。直接RhoStudioを起動してしまうとプロジェクトの作成がうまく動かないため、必ず[スタート]メニューに登録されているRhoStudioから起動しましょう。

### ■Mac OS環境でのインストール

Windows環境と同じく、前述の注6のインストールページからパッケージをダウンロードして解凍します。中にdmgファイルが含まれているので、それをダブルクリックします。Applicationフォルダへファイルをコピーするアイコンが目立ちますが、その前にInstall gemsスクリプトを実行します。このスクリプトはgemを使ってRhodesパッケージやその他の必要な環境をインストールするスクリプトです。

終わったら、RhoStudioをApplicationフォルダへコピーすれば利用できます。

### ■Linux環境／IDEが必要ない場合のインストール

Linux環境には、まだRhoStudioが提供され

ていません。そこで、gemを使ってRhodesパッケージをインストールしましょう。また、WindowsやMacの環境でもIDEが必要ない場合は、gemを使ってコマンドラインからインストールできます。

```
gem install rhodes
```

と入力してインストールが終わるのを待ちます。環境によってはインストールに管理者権限が必要になることもあります。その場合には、Linux/Macではsudoを付けます。Windowsの場合はコマンドプロンプトを管理者権限で起動します。

## RhoStudioの設定

RhoStudioが起動するようになったら、[Window]→[Preferences]を選択して環境設定を行います(Macでは、[RhoStudio]→[環境設定])。左側のペインにある[RhoMobile]の項目を展開し、[Android SDK]と[Android NDK]のパスを自分の環境に合わせて設定しましょう。

プロジェクトの作成は、Eclipseと同じように[File]→[New]→[Project]です。[RhoMobile]を展開して[RhoMobile application]を選択します。[Project name]などを入力して[Finish]を選びましょう。

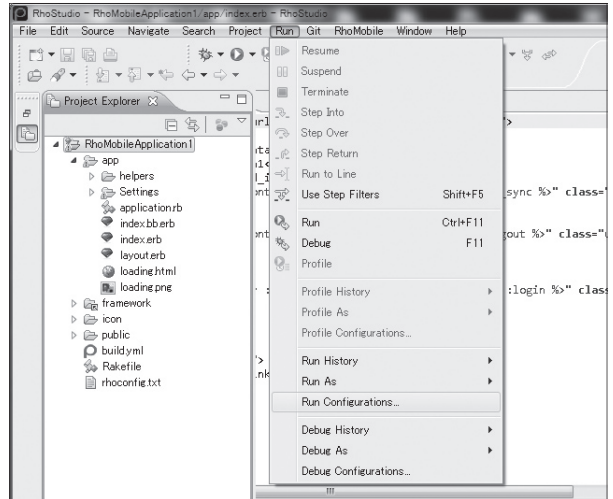
プロジェクトを作成できたら、次に実行環境の設定を行います。左側の[Project Explorer]で作成したプロジェクト名をクリックしてから、[Run]→[Run Configurations]を選択して行います(図2)。

左側のペインで[RhoMobile Application]を選択します(図3)。次に左側のペインの一番上に並んでいるアイコンのうち、左端の新規作成のアイコンを選んで設定を作成しましょう。

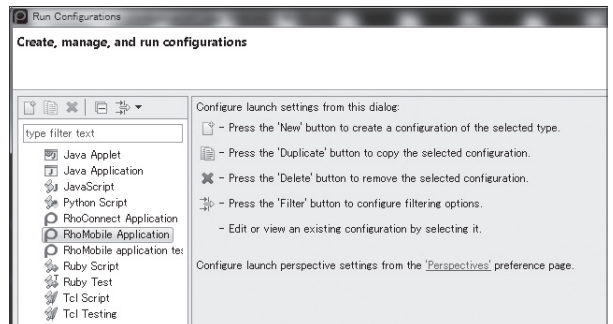
[Platform]では、実際に動作させる環境を選択します。Android/iPhone以外にも各種スマートフォンを選べます(図4)。

[Simulator type]ではエミュレータのタイプ

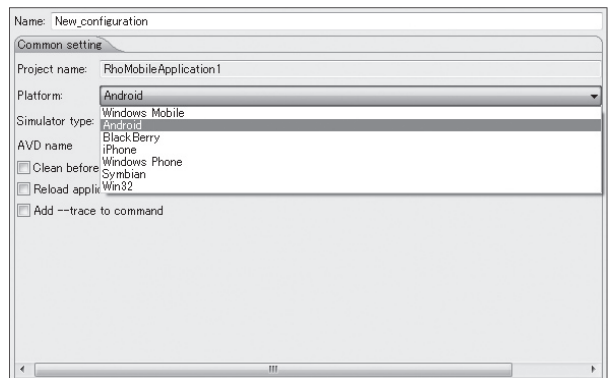
▼図2 実行環境の設定



▼図3 RhoMobileの設定



▼図4 ターゲットの選択



を選択します<sup>注9</sup>。RhoMobile付属のエミュレータ、各SDKに付属のエミュレータ、実機の3種類を選べます(図5)。

注9) RhoStudioの画面のUIではSimulator(シミュレータ)となっていますが、ここではAndroidの環境などにならってエミュレータと表記します。



## ▼図5 実行するエミュレータの選択

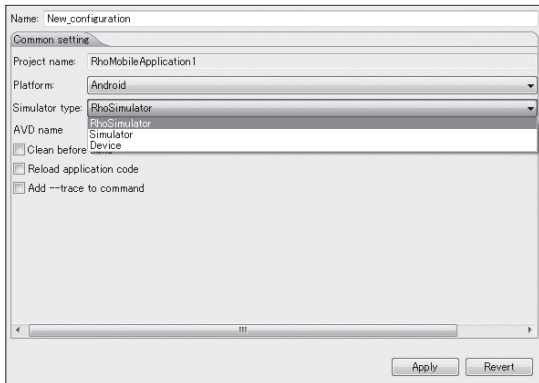
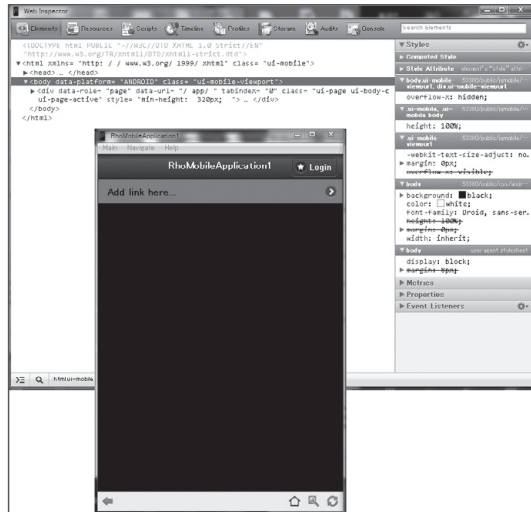


図6 サンプルを実行したところ▶



ここまでの設定が終わると環境設定用のダイアログにある[Run]ボタンを押せるようになるので、これを押してアプリケーションを実行します。RhoSimulatorを選んで実行するとデバッグも同時に起動します(図6)。

これで、RhoStudioをインストールし、作ったアプリを動かせるようになりました。

## 📧 Rhodesの構成

Rhodesは、MVCアーキテクチャを採用しているので、Model/View/Controllerのすべてが独立しています。Modelはアプリケーションのロジック部分とデータモデルを指していて、データベースとの連携やその他のデータのやりとり、データの加工などメインとなる処理を行います。

Viewは文字どおり表示を担当する部分です。HTMLで作成されたテンプレートを使って、動的に表示用のHTMLを生成します。テンプレートにはModelが処理した結果を表示するなど、結果に使われている変数を埋め込むことで必要な情報が生成できます。表示には、Rubyのテンプレートエンジンであるerbが採用されています(リスト1)。

Controllerは、ユーザとView/Modelの橋渡しを行います。ユーザからのレスポンスやフォー

ムのデータを受け取って必要に応じてModelやViewへデータを送ります(図7)。

RhodesアプリはRubyを使って書いていきます。画面表示のテンプレートエンジンには、erbが採用されているので、View用のHTMLにRubyを埋め込んで使えます。

しかし、動的に決定するURLや結果表示のための変数以外は埋め込まないほうがよいでしょう。View部分はあくまでも表示のみを行い、データ処理はすべてModel側で行うのがMVCの構成として一番わかりやすくなります。

Viewでは、HTMLの他にJavaScriptが使われています。jQueryも同梱されているので、UIの記述やデザインに関しては従来のWebデザインを使えます。

## 📧 ネイティブアプリとして動作可能

スマートフォンのアプリを作るうえで必要となる機能を調べるには、Rhodesのドキュメントを見てみましょう。サポートするOSと実装済み機能のリストがあります<sup>注10</sup>。

スマートフォンらしいアプリを制作するうえで重要になる機能、位置情報やバーコード、カメ

注10) 「Mobile API Compatibility」[URL http://docs.rhomobile.com/rhoelements/apicompatibility](http://docs.rhomobile.com/rhoelements/apicompatibility)

## ▼リスト1 erbで書かれたテンプレートの例(index.erb)

```

<div data-role="page" data-url="<%= Rho::RhoConfig.start_path %>/">
  <div data-role="header" data-position="inline">
    <h1>RhoMobileApplication1</h1>
    <% if SyncEngine::logged_in > 0 %>
      <a href="<%= url_for :controller => :Settings, :action => :do_sync %>"
        class="ui-btn-left" data-icon="refresh">
        Sync
      </a>
      <a href="<%= url_for :controller => :Settings, :action => :logout %>"
        class="ui-btn-right" data-icon="star">
        Logout
      </a>
    <% else %>
      <a href="<%= url_for :controller => :Settings, :action => :login %>"
        class="ui-btn-right" data-icon="star">Login</a>
    <% end %>
  </div>

  <div data-role="content">
    <ul data-role="listview">
      <li><a href="#">Add link here...</a></li>
    </ul>
  </div>
</div>

```

ラ、電話帳へのアクセスについては実装されており、利用できます。また、BluetoothやNFCのサポートも搭載環境への対応が行われています。

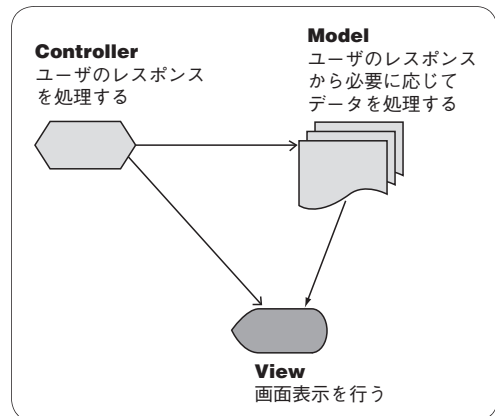


## まとめ

今回、もう少しRhodesの内容を入れようと思っていたのですが、RhoStudioを自分でインストールしたときにいろいろとわからない点もあったので注意点をまとめました。肝心のアプリ制作に関しては、次回にまとめていきたいと思います。

RhodesもPhoneGapもどちらもスクリプト言語でネイティブなアプリを作るという目的としては同じ思想で作られています。こうした環境が増えてくると、スクリプト言語が少しでも書ければ、考えたことをそのままプロトタイプとして作り、さらにブラッシュアップしてGoogle PlayやApp Storeなどでリリースできるアプリ

## ▼図7 MVCの構成図



に仕上げる機会も増えるでしょう。自分の考えたアプリを作って発表したいけど、JavaやObjective-Cを覚えるのはたいへんだという人もこうした環境を一度試してみてください。

今回は、Rhodesでアプリを作ってリリースするまでの流れを紹介していきます。**SD**

嶋崎 聡 (しまざき さとし) よこいど/日本Androidの会 横浜支部所属

Androidでゲームを作りたいがために長年敬遠してきたJavaを覚えたまではよいのですが、使い始めたときの環境(Android 1.6)では描画とGC(と自分自身のスキル)のおかげで求める速度が出せないまま、いつのまにか当初の目的を忘れて全然違うものばかり作っています。そろそろゲームを……。

# ハイパーバイザの作り方

## ちゃんと理解する仮想化技術

### 第4回

## I/O 仮想化「割り込み編・その1」

浅田 拓也 (ASADA Takuya) Twitter @syuu1228

### 割り込みの種類

前回の記事ではI/O仮想化の機能のうち、I/Oデバイスへのアクセスをエミュレーションする方法を中心に解説してきました。今回は、割り込み仮想化の話の前提となるx86アーキテクチャにおける割り込みのしくみを解説します。

CPUが持つ機能として、割り込みというものがあります。これは、現在CPU上で実行しているプログラムを停止して、別の処理を実行するための機能です。

CPUより処理速度の遅い周辺機器へのI/Oを非同期に行うため、周辺機器からCPUへI/O終了を通知する機能として実装されました。現在では、より広い用途で用いられています。



### 外部割り込み、内部割り込み

割り込みは、外部割り込みと内部割り込みの2種類に分けられます。外部割り込みとは、ハードウェアからCPUへ「キーボード押された」などのイベント通知を行うのに用いられます。外部割り込みは、さらにマスク可能な割り込みとソフトウェアからマスク可能な割り込み (NMI) の2種類に分類されます。マスク可能な割り込みは通常のデバイスからの割り込みに用いられます。また、NMIはハードウェア障害の通知など一部特殊な用途に用いられます。

内部割り込みとは、CPU内部の要因で発生する割り込みのことです。この内部割り込みは、ソフト

ウェア割り込みと例外に分類されます。ソフトウェア割り込みとは、割り込みを起こす命令 (INT 命令) により発生する割り込みです。これはシステムコールの実装に用いられます。例外とは、プログラムを実行した結果としてゼロ除算／オーバーフロー／無効な命令の実行／ページフォルトなどが発生したときなどに起きる割り込みです。これは、CPUで実行されるプログラムの制御に用いられます。

### ベクタ番号と IDT

これらすべての種類の割り込みに0-255のベクタ番号が割り当てられます。例外は要因別に0-19、NMIは2が固定的に割り当てられています。ソフトウェア割り込みは0-255のすべてのベクタ番号を使用可能、外部割り込みは16-255を使用可能になっています。

割り込みを使用するにはIDT (Interrupt Descriptor Table) と呼ばれる割り込みハンドラのアドレスを格納する最大255エントリのテーブル (配列) を作成し、IDTのアドレスをIDTRレジスタに設定する必要があります。割り込み発生時、CPUはIDTRの値からIDTを参照し、指定された割り込みハンドラを実行します。

IDTの各エントリはGate Descriptorと呼ばれるフォーマットで記述されます。これは単なるメモリアドレスではなく、セグメントの設定や実行権限 (Ringの値)、リアルモード・プロテクトモードの設

定などいくつかのパラメータを含みます(図1)。

IDTに用いられる Gate Descriptorには Task gate descriptor/Interrupt gate descriptor/Trap gate descriptorの3種類があります。しかし、Task gate descriptorはハードウェアによるマルチタスク機能呼び出すためのもので、現代のOSでは使われていません。代わりに Interrupt gate descriptorと Trap gate descriptorが用いられます。

この2つの違いは、Interrupt gate descriptorが EFLAGS レジスタの IF フラグ(割り込みフラグ)をクリアするのに対して、Trap gate descriptorは IF フラグをクリアしない(割り込みハンドラ内で割り込みを禁止しない)というものです。

IDTR レジスタのフォーマットは、IDT の先頭アドレスと Limit 値(テーブルのサイズ)の組み合わせになっており、255 未満のテーブルサイズに対応できるようにになっています。

### 割り込みのマスク・アンマスク

EFLAGS レジスタ (Intel 64ではRFLAGSレジスタ)のIFフラグをクリアすると割り込みを禁止、IFフラグをセットすると割り込みを有効にすることがで

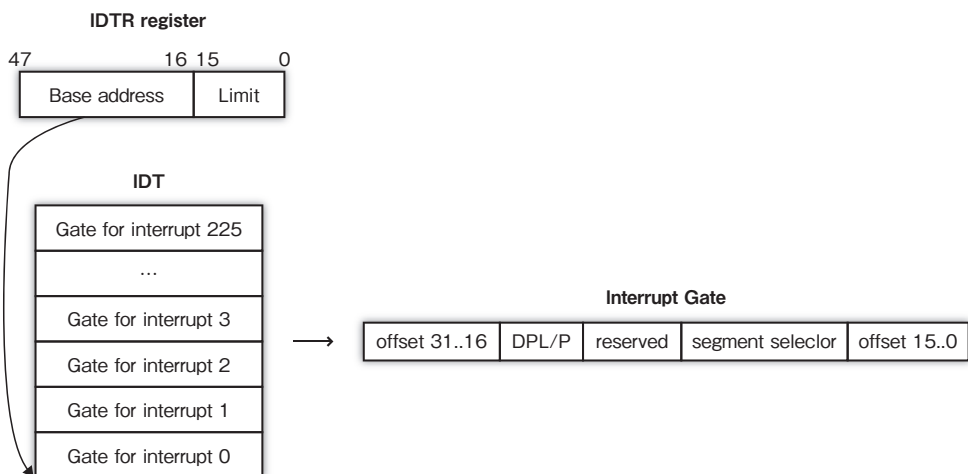
きます。プログラムからIFフラグをクリア/セットするには、CLI命令/STI命令を使用します。

また、前述のとおり、Interrupt Gate Descriptorを使用して割り込みをハンドルする場合はCPUによってIFフラグがクリアされるため、割り込みは自動的に禁止されます。このとき、割り込みハンドラを終了して割り込み前に実行していたプログラムへ制御を戻すIRET命令を実行すると、割り込み前のEFLAGSレジスタの値がリストアされ、再び割り込み可能になります。

### 外部割り込みと割り込みコントローラ

オリジナルのPCアーキテクチャでは、外部割り込みを制御し、CPUへ伝える役割を受け持つ割り込みコントローラとしてPIC(i8259)がありました。これは、マザーボード上に存在し、CPUの割り込みラインに接続されていました。Pentium以降のx86アーキテクチャでは、外部割り込みの管理はAPIC(Advanced Programmable Interrupt Controller)と呼ばれる新しい割り込みコントローラへ移行され、PICは互換性のためだけに存在するようになりました。APICはCPUごとに存在し、CPUに内蔵さ

▼図1 IDT、IDTR、Gate descriptorの関係



# ハイパーバイザの作り方

ちゃんと理解する仮想化技術

れている Local APIC と、ICH (Southbridge) に内蔵されている I/O APIC から構成されています (図2)。

Local APIC は、ローカル割り込みのベクタ番号設定、割り込みベクタ番号通知、EOI (割り込み終了) 通知など、一般的な割り込みコントローラの役割を受け持ちます。また、タイマー／温度センサ／パフォーマンスモニタリングカウンタなどのデバイス、IPI (プロセッサ間割り込み) の送受信機能を内蔵しています。

一方、I/O APIC は外部デバイスから割り込みを受け取り、I/O APIC 上の設定に基いて割り込み配信先の Local APIC を選び割り込みをリダイレクトする機能を受け持ちます。

このような構成を取ることによって、SMP 環境下で複数の CPU で外部割り込みを処理できるようになっています。

また、ある CPU の Local APIC から別の CPU の Local APIC へ IPI (Inter-Processor Interrupt: プロセッサ間割り込み) を送ることもできます。

## Local APIC

表1に割り込みの処理で利用される Local APIC の主なレジスタを示します。

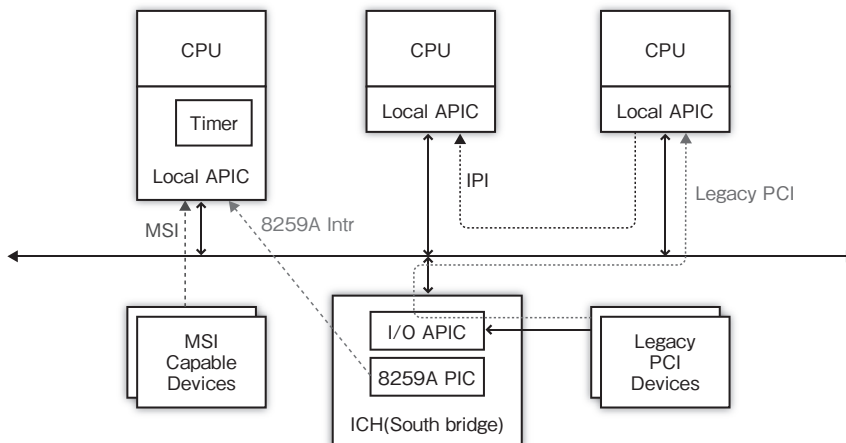
割り込み着信時の Local APIC および CPU の挙動を簡単にまとめると、次のよう流れになります。

- ① Local APIC が割り込みを受信したら、IRR に対応するベクタ番号のビットをセットする。CPU が割り込みをブロックしている場合はここで処理は終わり
- ② IRR にセットされた最高優先度のビットをクリア、同じビットを ISR にセット、同じ優先度の割り込みを CPU へ発行する
- ③ CPU で割り込みハンドラが実行される
- ④ CPU で実行された割り込みハンドラが EOI レジスタに書き込み、割り込み処理の終了を伝える
- ⑤ EOI レジスタへの書き込みを受け取ると、ISR にセットされた最高優先度のビットをクリア。まだ IRR にビットが残っていたら2から繰り返し

ただし、タスク優先度／プロセッサ優先度という機能があり、これによって割り込みに対する実行中タスクの優先度が制御でき、タスクの優先度が受信した割り込みより高い場合、その割り込みはマスクされます。

TPR・PPR レジスタの値は優先度クラス (4-7bit) ・サブ優先度クラス (0-3bit) の2つの値からなり、優先度クラスだけが割り込みのマスクに使用されます。優先度サブクラスは後述の「Lowest priority Mode」

▼図2 Local APICとI/O APIC、外部デバイスの関係



で使われますが、割り込みのマスクには使用されません。

PPRの更新はTPRに新しい値が書き込まれたときとCPUに割り込みを発行するとき(前述の割り込み着信時の手順2の段階)に行われます。値は次のように設定されます。

```
/* [7:4]は優先度クラス */
/* [3:0]は優先度サブクラス */
if TPR[7:4] >= ISR[7:4]
    PPR = TPR
else
    PPR[7:4] = ISR[7:4]
    PPR[3:0] = 0
```

また、Local APICに割り込みが着信した段階(前

述の割り込み着信時の手順1の段階)で、IRRにセットされた最高優先度のビットとPPRの優先度クラス(4-7bit)が比較されます。

PPRの優先度クラス(4-7bit)のほうが高かった場合は割り込みがマスクされます。

## I/O APIC

I/O APICには外部デバイスからの割り込み線が接続され、24本の割り込みをサポートしています。各割り込みの割り込み先はI/O APIC上のRedirection Tableに設定され、システムバスを通じてCPUの

表1 Local APICの主なレジスタ

IRR (Interrupt Request Register)	割り込み要求レジスタ(Read-only)：未処理の割り込みを管理するレジスタ。Local APICに外部割り込みが着信するたびにベクタ番号に対応するビットがセットされる
ISR (In-Service Register)	インサービスレジスタ(Read-only)：次に発行される割り込みの候補を管理するレジスタ。EOIレジスタへ書き込まれるとLocal APICによってIRRから最高優先度のビットがコピーされる
EOI (End Of Interrupt)	割り込み終了レジスタ(Write-only)：割り込み処理の終了をLocal APICに通知するためのレジスタ
TPR (Task Priority Register)	タスク優先度レジスタ(Read/write)：外部割り込みに対する実行中タスクの優先度を設定するレジスタ。TPRに書き込まれた優先度より低い優先度の割り込みはマスクされる
PPR (Processor Priority Register)	プロセッサ優先度レジスタ(Read-only)：実際の割り込み着信時にマスクを行うか否かの判定に使われるレジスタで、TPRとISRに連動して更新される。
Local APIC ID Register	LAPIC IDレジスタ(Read/Write)：システム全体でCPUを一意に特定するためのIDであるLAPIC IDを格納しているレジスタ。LAPIC IDはI/O APICから外部割り込みを転送するときやIPIを送るときなどに使用される
ICR (Interrupt Command Register)	割り込みコマンドレジスタ(Read/write)：IPI(プロセッサ間割り込み)を送信するためのレジスタ
LDR (Logical Destination Register)	Logical APIC IDを指定
DFR (Destination Format Register)	Logical Destination Modeのモデルを指定(Flat model/Cluster model)

表2 I/O APICのRedirection Table Entry

63:56	Destination	宛先 Local APIC の指定
16	Mask	割り込みマスク
15	Trigger Mode	エッジトリガ／レベルセンシティブ
14	Remote IRR	レベルセンシティブモードで割り込み送信中／EOI 着信済み
13	Interrupt Pin Polarity	レベルセンシティブモードでhigh/lowのどちらを割り込みリクエストとするか
12	Delivery Status	割り込みベンディング中かどうか
11	Destination Mode	Physical Mode／Logical Mode
10:8	Delivery Mode	Fixed／Lowest Priority など
7:0	Vector	ベクタ番号

# ハイパーバイザの作り方

ちゃんと理解する仮想化技術

Local APICへ転送されます。Redirection Tableの各エントリのフィールドの詳細を表2に示します。

Redirection Table EntryではDestination Modeを指定する必要があります。Destination ModeにPhysical Destination Modeを指定した場合、Destination フィールドにLocal APIC IDを指定することにより、宛先CPUを一意に特定します。たとえば、Local APIC ID: 3のCPUへ割り込みを配信したいときは、Destination フィールドに3を指定します。

Logical Destination Modeを指定した場合、Destination フィールドで複数の宛先CPU群をビットマスク指定します<sup>注1</sup>。宛先CPUのIDはLocal APIC IDではなく、LDRのLogical APIC IDが使用されます。たとえば、LDRの値が00000001bのCPUと00000010bのCPUに割り込みを配信したい場合、Destination フィールドに00000011bを指定します。

Logical Destination Modeにより複数のCPUが割り込み先に指定された時の挙動については、Delivery Modeで設定します。Fixed Modeでは、Destinationに指定されたすべてのCPUに割り込みます。Lowest priority Modeでは、DestinationのうちLocal APICのTPRが最も小さいCPUへ割り込みます。TPRの値が最も小さいCPUが複数存在する場合は、ラウンドロビンで割り込みが分散されます<sup>注2</sup>。いずれの配信モードの場合でも、Vector フィールドで設定されたベクタ番号でLocal APICへ割り込みがかかります。

APIC IDのアドレス幅は8bitですので、Logical Destination Modeでは最大8CPUしかサポートできません。このため、Nehalem世代よりx2APICと呼ばれる拡張版APICが導入され、アドレス幅は8bitから32bitへ拡張されました<sup>注3</sup>。

注1) これは正確にはLogical Flat Modelというモードで、他にFlat Cluster Model、Hierarchical Cluster Modelなどがありますが、通常使われません。説明は割愛します。

注2) WindowsではプロセスごとにTPRの値を制御し、プロセスの優先度に多じて割り込み量を変化させています。一方、LinuxではTPRの値を変えずにLowest Priorityを用い、複数のCPUへ割り込みを公平に分散させています。

注3) 参考資料 Intel64 Architecture x2APIC Specification (<http://www.intel.com/content/dam/doc/specification-update/64-architecture-x2apic-specification.pdf>)

## MSI・MSI-X 割り込み

I/O APICを経由するPCIデバイスの割り込みは、各PCIデバイスからの物理的な割り込み線がI/O APICへ接続され、この割り込み線を通じて割り込みが送信されていました。

この構成では割り込みの数とIRQの割り当てが物理配線に依存するため、限られたIRQを複数のデバイスで共有するようしくみになっていました。また、1つのデバイスで複数の割り込みを持つことはできませんでした。この制限を解消するため、物理配線を用いずPCIバス経由のメッセージとして割り込みを送るMessage Signalled Interrupt (MSI)・Extended Message Signalled Interrupt (MSI-X)が導入されています。PCIではオプション機能として提供されていますが、PCI expressでは必須とされています。

MSIでは各デバイスごとに32個、MSI-Xでは2048個の割り込みをサポートします。従来と異なり、デバイス間の割り込みは共有されません。MSI・MSI-Xの割り込みはIO-APICを経由せず、直接Local APICへ配送されます(図2)。このとき、宛先CPUの設定は各PCIデバイスのコンフィギュレーションスペースに設定されます。詳細は割愛しますが、コンフィギュレーションスペース内の割り込みの設定フィールドでは、表2のRedirection Table Entryに近い内容が割り込みごとに設定できます<sup>注4</sup>。

## まとめ

いかがでしたでしょうか。今回はx86アーキテクチャにおける割り込みのしくみを解説してきました。次回は、今回解説した割り込みのしくみをどう仮想化するかについて解説します。**SD**

注4) 参考資料 <http://d.hatena.ne.jp/syuu1228/20120105/1325757315>



## UbuntuとNexus 7の 微妙なカンケイ

今回は、Ubuntu 12.04/12.10でNexus 7を認識し、ファイルを転送するさまざまな方法を紹介します。

Ubuntu Japanese Team あわしろいくや AWASHIRO Ikuya ikuya@fruitsbasket.info

### UbuntuとMTPの 微妙なカンケイ

Android 2.xまでは内蔵Flashメモリ(とSDカード)がUSB Mass Storage Class、すなわちUSBメモリなどと同じように認識できたため、USBケーブルを接続すればAndroidのスマートフォンやタブレットとUbuntuは、とくに問題なく認識できていました。しかし、Android 3.1からはMTP(Media Transfer Protocol)のみの対応となったため、話が少々ややこしくなりました。というのも、UbuntuのMTPサポート<sup>注1</sup>は芳しい状況とは言えないからです<sup>注2</sup>。

まず、libmtp<sup>注3</sup>というライブラリでMTPをサポートしていますが、これは“music-players.h”というヘッダファイルに対応するAndroidデバイスのベンダIDとデバイスIDを持っています。逆にいえば、ここにはないベンダやAndroidデバイスはUbuntuでは認識しません。もしあったとしても、Windowsのように接続すれば即座に内蔵Flashメモリにアクセスできるようにはなっておらず、ひと手間加える必要があります。今回はその方法を紹介します。

Androidデバイスを一緒に紹介するのは難しいので、実際に動作確認を行ったNexus 7での

紹介になりますが、認識さえすればほかのAndroidデバイスでも同様のことが行えるはずです。

### Nexus 7について

Nexus 7については、ここをお読みの方にはとくに解説の必要はないと思います。もしご存じない場合は、Googleの紹介サイト<sup>注4</sup>をご覧ください。筆者は9月25日にGoogle Playで購入しました。申し込みできるようになった直後に注文して、未だに手元に届いていない人も少なからずいるようですが、筆者の手元には比較的スムーズに届きました。現在は内蔵Flashメモリが32GBのモデルも発売され、さらに新モデルNexus 10も発表されています<sup>注5</sup>。

Nexus 7はUbuntu 12.04リリース後に発売されているので、当然そのままだと認識しません。ですので、12.04で認識したい場合のためにPPAにパッケージを用意しました。あくまでNexus 7のベンダIDとデバイスIDを追加しただけであり、libmtpのバージョンを上げたわけではないので、その点ご注意ください。それ以外にも今回紹介するパッケージやライブラリをアップロードしているので、このPPAを追加していることを前提に話を進めます。

PPAを追加し、libmtpのバージョンを上げる方法は次のとおりです。


注1) UbuntuだけでなくほかのLinuxディストリビューションでも似たようなものだと思います。

注2) ただし、ソニーのウォークマンZ/Fシリーズなど、Android 3.1以降でもUSB Mass Storage Classに対応するようにAndroidに手を入れたデバイスはあるようです。

注3) <http://libmtp.sourceforge.net/>

注4) <http://www.google.co.jp/nexus/7/>

注5) 日本でも発売される予定ではあるようですが、11月中旬現在では発売されていません。

```
$ sudo add-apt-repository ppa:ikuya-
fruitsbasket/nexus7
$ sudo apt-get update
$ sudo apt-get upgrade
```

なお、Ubuntu 12.10のlibmtpはNexus 7を認識します。

## 認識させる際の注意

Nexus 7を認識させる前に、必ずロック画面の解除を行ってください。そうしないと正しく認識しません。考えてみたら当然のことで、たとえば盗難にあった場合にロック画面のまま内蔵Flashメモリにアクセスできるとなるとセキュリティ上大きな問題となります。たったこれだけのことですが、知らないといけこうハマります。USBデバッグはオンでもオフでもどちらでもかまいませんが、タイミングによってうまく認識しない場合はオン／オフを切り替えてみるといいかもしれません。

## Rhythmboxで音楽ファイルの同期

UbuntuにデフォルトでインストールされているアプリケーションでMTPに対応しているのはRhythmboxです。Nexus 7を認識していると、左側の

サイドペインに表示されます。表示されているNexus 7を右クリックし、[ファイル情報]をクリックして正しい容量が表示されている場合は、音楽ファイルの同期が行えます(図1)。もし正しく表示されない場合(9.2EBなどありえない容量が表示される場合)は(図2)、一度「取り出す」をクリックし、ロック画面を解除してからUSBケーブルを抜き差ししてください。

手取り早く正しく認識しているかどうかを確認できるようにするため、1つ以上の音楽ファイルを転送しておくとも簡単です。方法は、[再生]-[プレイリスト]-[新しいプレイリスト]でプレイリストを作成し、そこに転送したいファイルをドラッグ&ドロップします。その後[ファイル情報]の[同期]タブで同期するプレイリストを選択し(図3)、閉じてから[デバイスと同期]をクリックして同期を待つだけです。同期が完了したら、[取り出す]をクリックしてからUSBケーブルを外しましょう。もちろんライブラリにあまり多くの音楽ファイルがない場合は[ミュージック]にチェックを入れてもいいですが、正直なところデジタルオーディオプレーヤ(DAP)としてのNexus 7はあまり優れているとはいえないので、たくさんの曲を転送しても活用する機会はあまりないでしょう。

図1 正しく認識されている状態



図2 正しく認識されていない状態。ファイルがあるにもかかわらず何も表示されず、容量もおかしい



## Calibreで電子書籍の転送

Nexus 7のディスプレイはサイズの割に高解像度で、DAPとしてよりも電子書籍端末としてのほうがはるかに有意義な使い方ができます。

有名な電子書籍管理アプリケーションである“Calibre<sup>注6)</sup>”はバージョン0.8.70からNexus 7に対応したようですが、Ubuntu 12.10のCalibreでもバージョン0.8.51で非対応です。よって前出のPPAに12.04/12.10用のバージョン0.9.0をアップロードしました。ただし、12.04でここにあるCalibreをインストールするとpython-cssutilsのバージョンも上がります<sup>注7)</sup>。

注6) <http://calibre-ebook.com/>

注7) とくに問題ないとは思いますが、念のためお知らせしておきます。

図3 同期する場合は、同期したいプレイリストにチェックを入れるのが一番簡単



インストール方法はお好みでかまいません。たとえば端末からインストールする場合は、

```
$ sudo apt-get install calibre
```

となります。

Nexus 7を接続した状態でCalibreを起動すると、[Manage the Nexus 7?]というダイアログが表示されるので(図4)、[はい]をクリックすると認識します。

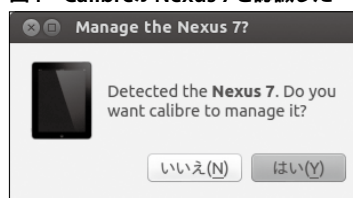
あとは簡単で、[本を追加]をクリックして転送したい電子書籍(今回はPDF)を指定します。その後メニューの[ライブラリ]に追加されるので、それを右クリックして[デバイスに送信]-[メインメモリに送信]をクリックするだけです(図5)。転送した電子書籍を確認する場合はメニューの[デバイス]をクリックします。

ここでは詳しく紹介しませんが、Calibreの[ニュースを取得]機能を使用すると、いくつかのニュースサイトをEPUBに変換したうえでNexus 7に転送してくれます。読むためにはもちろんEPUBに対応したアプリケーションが必要ですが、たとえば“出社前にニュースを取得しておいて出勤中に読む”ということもできます。この場合インターネット回線は必要ないので、モバイルルータやテザリング機能のあるスマートフォンがなくても読むことができるのは便利かもしれません。

## gMTPでいろんなフォルダにアクセス

RhythmboxではNexus 7のMusicフォルダ、CalibreではBooksフォルダにアクセスできますが、それ以外にもたくさんのフォルダがあります。たとえば動画を置くためにMoviesやVideoフォルダも用意されていますが、RhythmboxやCalibreではこの

図4 CalibreがNexus 7を認識した



フォルダにアクセスすることはできません。内蔵Flashメモリのどのフォルダにもアクセスしたいというのは当然の要求だと思います。そういった場合には、メニューは英語ですがgMTPを使用すると簡単です。これも先のPPAに、11月中旬現在で最新版の1.3.4をアップロードしてあります。

やはりインストールの方法は任意でかまいません。たとえば端末からインストールする場合は、

```
$ sudo apt-get install gmt
```

となります。

gMTPを起動して[Connect]をクリックしてしばらく待つとフォルダ一覧が表示されます。[Add]はファイルの追加、[Delete]はファイルの削除、[Download]は内蔵FlashメモリからUbuntuにファイルをコピーします。フォルダごとアップロードできないのはやや不便です。gMTPはほかにもいろんな機能がありますが、ファイルのやりとりぐらいであればおおむね問題なく動作しました。[Preferences]で既定の挙動を変更することもできるので、必要であれば確認してみてください。そして操作が終わったら[Disconnect]をクリックしてください。

とにかくコマンドをいっさい触りたくないという場合に使用するというでしょう。

## Go-mtpfsでNautilusから操作

「いちいち内蔵Flashメモリにアクセスするために別アプリケーションを使用するのは好まない」「Nautilusから操作したい」と思う人もいるかもしれません。そのため、今回はGo-mtpfsというFUSE (Filesystem in Userspace) ファイルシステムを使用します。MTPのFUSE実装はほかにもありますが、これが簡単でいいと思いました。作者はGoogle社員で、Go言語で書かれているのが興味深い、という理由もあります。

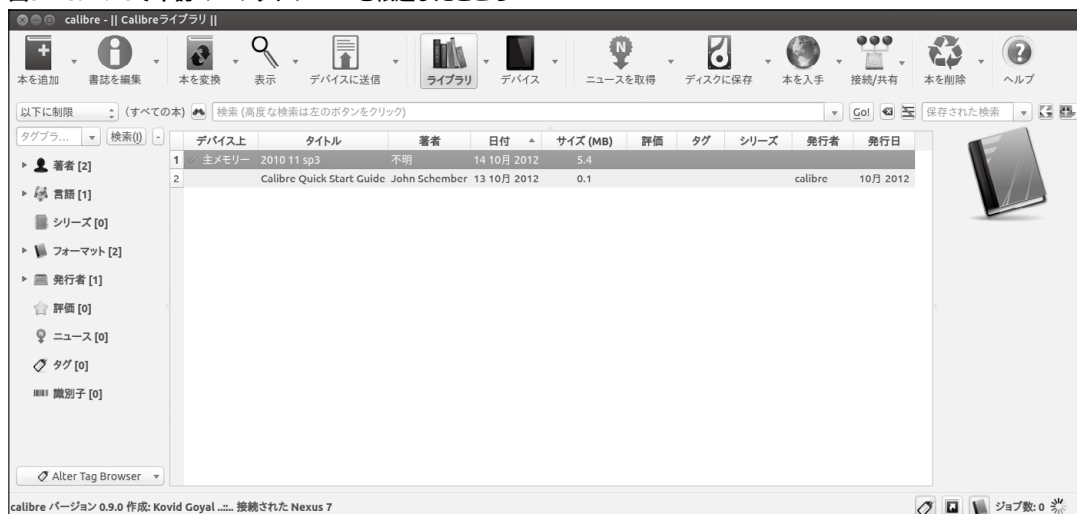
パッケージにはなっていないため、次のようにコンパイルを行います。

```
$ sudo apt-get install golang git-core libmtp-dev
(質問されるが、どちらでもかまわない)
$ mkdir -p go-mtpfs/nexus7
$ cd go-mtpfs/
$ export GOPATH=$HOME/go-mtpfs
$ go get github.com/hanwen/go-mtpfs
```

とくにメッセージが表示されない場合は、正常に終了しています。

go-mtpfs/bin/go-mtpfsに実行ファイルができており、今回はgo-mtpfs/nexus7にマウントします。カレントフォルダをgo-mtpfsのままだとすると、

図5 Calibreで本誌のバックナンバーを転送したところ



```
$ bin/go-mtpfs nexus7 &
```

でマウントします。

Nautilusの[コンピューター]に[nexus7]が増えているので(図6)、ここからアクセスします。

作業が完了したら、次のコマンドを実行してください。

```
$ fusermount -u nexus7
```

安定していないとはいませんが、割に転送に時間がかかったりはしますので、時間的／精神的に余裕を持って実行してください。

## 将来的にはNautilusから直接扱えるように

GNOMEに詳しい人であれば、ここまできたらGVFS(GNOME Virtual File Systemの代替)でMTPをサポートするのがベストではないか、という解にたどり着くのではないかと思います。まったくもってそのとおりで、実際に実装が開始しています<sup>注8</sup>(図7)。

GNOME 3.8に実装されればUbuntu 13.04でも使用できるようになるので、期待したいところです。**SD**

注8) [https://bugzilla.gnome.org/show\\_bug.cgi?id=666195](https://bugzilla.gnome.org/show_bug.cgi?id=666195)

図6 Nautilusで見ると[コンピューター]の下に表示される

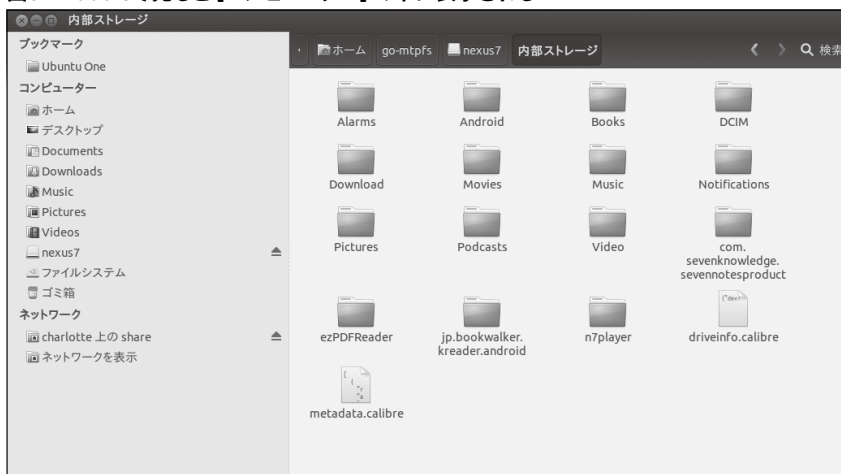
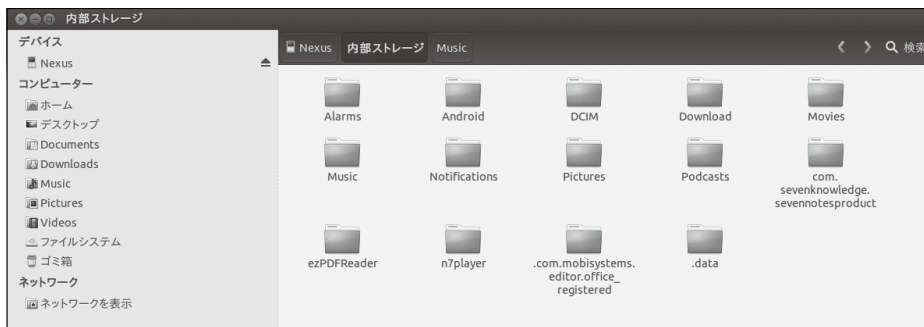


図7 現在配布されているパッチをGVFSに適用すると、USBメモリなどと同じく[デバイス]という項目のところに表示される



# IPv6化の道も 一歩から

## 第2回

# いよいよIPv6化を実践、 そのときに考えること、調べること

IPv6普及・高度化推進協議会 IPv4/IPv6共存WG アプリケーションのIPv6対応検討SWG  
廣海 緑里 HIROMI Ruri 渡辺 露文 WATANABE Tsuyufumi 新 善文 ATARASHI Yoshifumi 藤崎 智宏 FUJISAKI Tomohiro

### 前回のおさらい

すでに公開されている仕様や技術資料を踏まえて、システム企画者やシステム管理者、アプリケーション開発者を対象に、前向きに「組織としてIPv6にどう対応すべきか？」をつづる本連載ですが、2回目の今回はIPv6対応の全体像（インターネット公開サイト・システム編）の後編です。

まずは、軽く前回のおさらいです。IPv6対応の全体の流れを図1に示します。開始から順に「方針検討」「計画策定」「予算確保／体制作り／要員教育」、「構築（設計、実装）」「テスト／試験運用」「本番運用」となります。このメインの流れに並行して情報収集、技術検証を行います。

その結果をメインの流れに活かします。

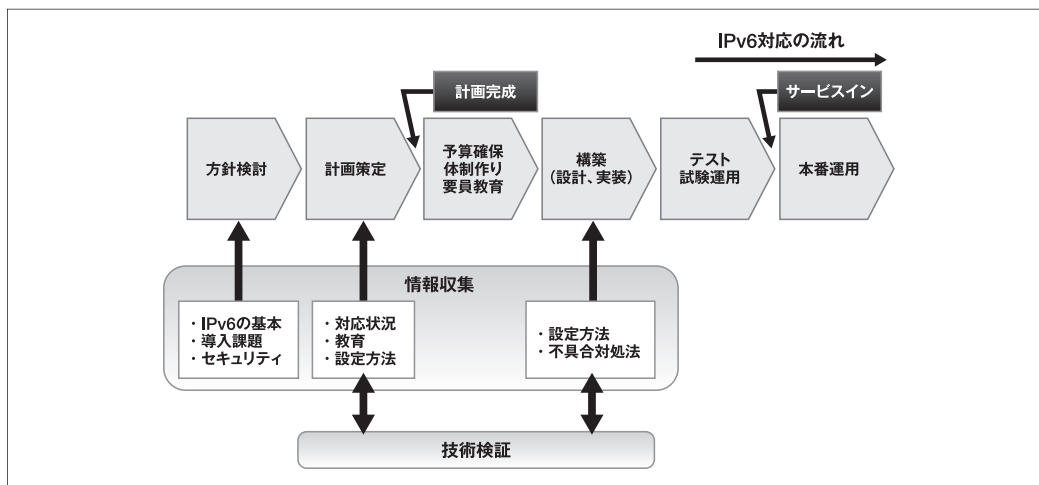
前は方針検討について説明しました。言い換えると、方針検討だけで1回分費やしていました(苦笑)。でも、それだけ重要なのです。方針検討フェーズをざっくり要約すると、

- IPv6対応の目的が重要
- 何のためにIPv6化するか目的を明確にする
- 時期、対象範囲は柔軟に検討する
- 構築方法は可用性、セキュリティ、コストなどを吟味して決める
- 体制は可能であれば専門部隊

です。

一度に集中して取り組むというやり方だけでなく、「IPv6の試験環境を作る」「IPv6試験環境とIPv4環境をつなぐ」「IPv4環境にIPv6を共

▼図1 IPv6対応の流れ



存させる」「いらないIPv4をなくしていく」「ユーザ端末はIPv6のみにしていく」「データセンターへのアクセスはIPv6で制御する」といったフェーズを区切った対応計画をたてて、段階的に対応していくこともあるでしょう。この先10年くらいの間で、どういう形にしていけるかを想像しながら最初の1、2年の具体的な計画をたて、その先の対応は最新動向を見ながら考えていくというのが多くの先行企業に見られるやり方ようです。それでは、今回の本題に入りましょう。

## IPv6対応の全体像(インターネット公開サイト/システム編 後編)



### 計画策定

方針を決定したあとは、方針を具体化して計画を策定しましょう。計画策定においては、方針として検討したものに加えて次のことを検討します。

- 構成：ネットワーク構成、ハードウェア構成、ソフトウェア構成、機器／ソフトウェアを決定する
- 要員教育：どのように要員を教育するかを計画する
- コスト：要員教育、構築から運用までを考慮してコスト算出する
- スケジュール：対応のマイルストーン、スケジュールを明確化する

計画策定段階では、用いる機材の種類、発生するコスト、要員教育の方法などの情報収集を行い、技術検証を重ねて実現方法を明確化しましょう。構成については、もちろんIPv6に対応している機器／ソフトウェアを選定し、システム構成を決定します。既存のアプリケーションをポーティングする場合には、改修が必要となるケースもあるためアプリケーションの調査／改修を計画に組み込みましょう。また、運用監視についても構成および実施方法を十分に検討し

## COLUMN



### 承認が簡単に得られない……？

「IPv6対応の取り組みに対する会社からの承認が簡単に得られない」という声をよく耳にします。

いくつかの事例では、トピックの説明をすると細かな点の指摘を受けるかたちになってなかなか承認が得られず、むしろシンプルに「将来必要になる」という一点で突破する傾向にあるようです。「IPv4アドレスが枯渇した」「GoogleなどがWorld IPv6 Launchといって導入を開始した」といったトピックの説明を持ち出すと、それぞれについて会社とどういう関係があるのかさらに説明を求められて、最終的に「じゃあ、今は必要ない」という結論が導きだされるケースが多いようです。

役員や部長など経営層の方は往々にして忙しく、短時間で経営的な判断を下さなければなりません。そのため、経営層を相手に話をする際には、判断を下しやすいよう「会社として」導入する目的、効果、リスク、コストを整理し、結論を先

に話すよう心がけましょう。

たとえば、2012年6月のWorld IPv6 Launchを引き合いに出すと「World IPv6 Launchに参加してリストに社名を掲載してもらい国内外に新しい技術に取り組む会社であることをアピールする」「そのために今からIPv6の環境を整える」といったシンプルな説明に落としていくと良いようです。

インターネットが今日のように一般的ではなかった1990年代初頭にインターネット接続の承認を得た当時の稟議書を見たり、先輩から話が聞けたりできるのであれば、参考にしてみるのもいいかもしれません。当時は「電話やFAXがあるから必要ない」とバッサリ切り捨てられていて、インターネットの力を見いだすことは難しかったはずですが。そのときに承認を得た経験は今も活かせるはずです。

ましょう。

要員教育については、構築要員、運用要員それぞれについて、どの程度の人数をどのように教育するかを計画します。コストについては、構築のみならず、運用、要員教育までを考慮して算出する必要があります。スケジュールについては、経験不足から生じる手戻りや不具合が想定されますので、余裕を持ったスケジュールを立てましょう。

これらの検討を重ねて策定した計画について、しかるべき社内の承認を得ましょう。



## 情報収集

情報収集において、集めるべき情報はフェーズによって異なります。

方針検討フェーズでは「IPv6とは何か？」やIPv4との違いなど、IPv6に関する基本的な情報や、導入課題に関する情報を収集しましょう。次回の本連載では、IPv6の基本について説明する予定です。次回まで待てないという気の早い方は、IPv4アドレス枯渇対応タスクフォース<sup>注1</sup>やIPv6普及・高度化推進協議会<sup>注2</sup>などから情報を収集しましょう。

セキュリティに関する情報はIPv6普及・高度化推進協議会セキュリティWGがまとめた「IPv6対応セキュリティガイドライン(第1.0版)」<sup>注3</sup>を参考にしましょう。前回説明した公開セグメントのIPv6対応モデルについてセキュリティの視点から述べられています。

計画策定フェーズでは、各機器、ソフトウェアの対応状況や設定方法、要員教育の情報を収集します。

要員教育に関して、前述のIPv4アドレス枯渇対応タスクフォースにて座学およびハンズオンのセミナーが開催されています。残念ながらすでに今年度分のセミナー開催は終了しています

が、2009年度のハンズオンセミナー資料が公開されています。来年度以降のセミナー開催を期待しつつ、まずはセミナー資料で学ぶことが可能です。また、タスクフォース以外に他社でもセミナーが開催されています。

後述する構築フェーズでは、おもに設定方法に関する情報収集を行います。また、不具合が生じたときのトラブルシュートに関する情報収集も行います。

また、情報収集に関しては「IPv6の企業への導入の参考文献は何かないのか？」「IPv6は技術の話ばかりで、進め方とかいくら検索しても出てこない」という声も聞きます。ISPなどの事業者、ユーザ企業、トップダウンで動くような地方自治体など、対象ごとの取り組み方のガイドラインやアクションプランがIPv4アドレス枯渇対応タスクフォースやIPv6普及・高度化推進協議会といった業界団体から出ています。じつは、総務省がこれらの団体のオブザーバーや1参加者として参加しており、総務省の研究会からも関係するガイドラインが出ていることがあります。また、経団連のような経済団体に所属している会社であれば、経団連がまとめたIPv6対応を促す提言<sup>注4</sup>なども社内説得に有効かもしれません。



## 技術検証

計画立案において、IPv6に限らず実現性が不透明な箇所は必ず技術検証を行いましょう。また、収集した情報を基に実際に動作させて確認することが重要です。論理的には正しくともバグなどで正しく動作しないことがあります。筆者は先日、glibc<sup>注5</sup>の不具合で/etc/resolv.confにnameserverとしてIPv6のアドレスを指定するとSegmentation Faultが発生しPostfixが起動しないという事態に遭遇しました(汗)。

注1) <http://kokatsu.jp>

注2) <http://www.v6pc.jp>

注3) [http://www.v6pc.jp/jp/upload/pdf/swg-IPv6SecurityGuideline\\_v1.0.pdf](http://www.v6pc.jp/jp/upload/pdf/swg-IPv6SecurityGuideline_v1.0.pdf)

注4) 「日米インターネット・エコノミー民間会合共同声明」の一節 <http://www.keidanren.or.jp/japanese/policy/2012/020.html>

注5) GNU C Library。GNUプロジェクトにおけるCの標準ライブラリ。



## 計画実践

計画が承認されたあとは実践あるのみです。新たに要員教育を行う必要がありますが、進め方は従来のIPv4でのシステム構築と大きな差はありません。

### 予算確保／体制作り

計画の実践では、何はともあれまずは予算の確保と体制作りが必要です。昨今の経済状況を考慮すると、これが一番の難関かもしれません。計画承認と同時に予算が確保される企業も多いかと思いますが、計画承認と予算確保が分離している企業では、経営層を相手にもうひと頑張りが必要です。体制作りにおいて確保する要員は、IPv6の前にIPv4に関する知識や技術があることが前提となります。さらに、DNSに関する知識や技術を有しているとなお良いです。

### 要員教育

体制が整ったあとは、実際に設計、構築に入る前に、要員の教育が必要となります。育成すべき要員は設計、構築工程だけではなく、運用を行う要員についても教育が必要となります。教育不要のIPv6の経験豊かな技術者を多く確保できれば言うことはありませんが、そううまくいかないのが世の常です。高望みするよりも、確保できた要員を教育するようにしましょう。要員教育については、前述のIPv4アドレス枯渇対応タスクフォースが開催するセミナーや公開しているセミナー資料があります。ほかにも前号でお知らせしたInternetWeekやIPv6サミットなど、IPv6に係る団体が開催するイベントや他社が行うハンズオンセミナーもあります。各組織のセミナー情報を入手し、うまく活用しましょう。

### 構築(設計、実装)

構築においては、IPv4とIPv6とでアドレスの考え方が異なるため、注意が必要です。IPv4

では、各ホストには1つのアドレスを割り当て、アドレス変換を用いてネットワークを構成していました。これがIPv6では、各ホストに複数のアドレスを割り当て、アドレス変換を用いずにアドレスを使い分けてネットワークを構成することになります。このあたりについては、次回説明します。

また、運用監視についても、IPv4とIPv6とでアドレスの考え方が異なるため、注意が必要です。各サーバのサービスについて、どこからのIPv6アドレスのサービスにアクセス可能かを整理する必要があります。これら以外の設計／構築についてはIPv6がゆえに難しいということとはとくにありません。

アプリケーションについてもIPv6対応が必要です。アプリケーションの場合、保守性を考慮し、同一ソースでIPv4とIPv6の両方に対応させます。IPアドレスを格納する処理・領域やIPアドレスに基づいた処理をIPv4アドレスとIPv6アドレスの両方で正常に動作するようにする必要があります。多くのプログラム言語ではIPv4とIPv6の両方に対応したクラス、構造体、関数が用意されており、これらを適切に使用することで、簡単にIPv6対応を行うことができます。ソケット通信を行う場合には、ソケットの送受信処理をIPv4とIPv6の両方に対応する必要があります(後述のコラム「『アプリケーションのIPv6対応ガイドライン 基礎編』を公開しました!」参照)。

また、アプリケーションが出力するログも影響を受けることがあります。接続元IPアドレスが記載されるWebサーバのアクセスログなどは、IPアドレス部分にIPv6アドレスが記載されるようになり、使用するアクセス解析ツールがIPv6に対応していないものと正しく解析できなくなることがあります。

### テスト／試験運用

テスト／試験運用においては、通常のシステム構築同様にテストと試験運用を行います。対

応方針でIPv6とIPv4の両対応を行うと決めた場合には、IPv6のみ／IPv4のみ／IPv6およびIPv4両対応(デュアルスタック)の3種類のクライアントからテストを行う必要があります。

また、クライアントのOSによって名前解決周りの実装が異なるため挙動が異なります。利用が想定されるクライアントOSすべてでの動作確認を行うべきでしょう。



## 本番運用

これまでの工程を無事に(何とか?)乗り越えることができたなら、晴れて本番運用の開始です。障害が起きないことを願いましょう。IPv6対応が成熟化していない現段階では、相手側の不具合でアクセスできない事象が発生する可能性があることを念頭に入れておきましょう。

IPv6はIPv4ほど枯れていないため、インターネットの技術仕様であるRFCが更新されポリシーが変わることも起こり得ます。そのため、

本番運用後も継続的に情報収集を行う必要があります。先日(2012年9月)も、RFC6724がRFC3484を置き換えるかたちで公開され、送信元アドレスの選択ルールに一部変更がありました。

## IPv6対応の全体像 (企業内LAN編)

企業内LANの対応の場合も、対応の流れおよび各工程はインターネット公開システム／サイトの場合と同様です。方針検討を重視し、何のために(目的)、いつ(時期)、どこまで(範囲)、どうやって(方法)対応するかを明確にして取り組みましょう。

想定される目的がインターネット公開サイト／システムの場合と異なり、情報漏洩防止／セキュリティ強化、もしくは(他の目的のための)LAN新設／LANリプレースのついでのいずれかかと思います。目的に応じて、時期、範囲、方法を明確にしましょう。

## COLUMN



## 「アプリケーションのIPv6対応ガイドライン基礎編」を公開しました!

本連載執筆陣が所属するIPv6普及・高度化推進協議会 IPv4/IPv6 共存WG アプリケーションのIPv6対応検討SWG<sup>注1</sup>では、「アプリケーションのIPv6対応ガイドライン 基礎編」を取りまとめ、先日公開しました(原稿執筆時点では完成していませんが、きっと本誌が発売されるころには公開されています)。

これまで、インターネットのユーザフロント、サーバフロントともにこれまでのIPv4での運用からIPv6が混在するIPv4とIPv6の共存期に入りつつある一方で、スマートフォンに代表されるスマートデバイスが普及し、ネットワークを使うアプリケーションは一般的になっています。こうした状況の中、アプリケーションの開発者が目にする情報は、これまでのIPv4で運営されているネットワークを前提としたものが多く、共存状況で注

意すべきことやするべきことが整理されていませんでした。2011年9月、当SWGが発足し、共存期を前提としたアプリケーション開発についての情報整理を行い、アプリケーション開発者に向けた情報発信と情報共有の活動を始めました。

このガイドラインは、ネットワークの基礎としてソケットを用いたプログラムのIPv6対応方法についてまとめた資料です。最近は直接ソケットを用いずにクラスライブラリなどを利用して簡単にネットワークを利用できる環境も多くなっていますが、基礎的な部分でどのような処理やデータのやりとりが行われ、その際IPv4とIPv6では何が違うのかを踏まえ、ソケットによるアプリケーション開発の概要をまとめたものです。ガイドラインをご確認いただき、アプリケーション開発にお役立ていただければ幸いです。

また、今後は基礎編に続いてWebアプリケーション編を作成する予定です。

注1) <http://www.v6pc.jp/jp/wg/coexistenceWG/v6app-swg.phtml>

注意すべき点は、企業内LANをIPv6対応する場合には必ずインターネット接続までIPv6対応しましょう。インターネットに接続できないIPv6環境(閉域網)では、インターネット接続において不具合が生じやすいです。また、IPv6未対応の機器が少なくない現状を考慮すると、当面IPv4との混在は避けられないでしょう。

企業内LANのIPv6対応では、情報システム部門が想定していない問い合わせが寄せられ、対応を迫られるケースがあることが想像されるため注意が必要です。これは情報システム部門が提供しているサービスやソフトウェア以外にも利用者が独自に利用しているソフトウェアや利用部署のローカルサーバなどが存在するケースがあるからです。そのため、事前の環境情報の収集も重要になります。

## 終わりに

今回は、前回に引き続きIPv6対応の全体像の

後編をお届けしました。いかがだったでしょうか？ これまでありそうでなかった話ですので、お気に召していただければ幸いです。

また、抽象的な話しか書けませんでした、次回以降は具体的な話に入っていきます。IPv6の具体的な話に入るには、どうしてもIPv4の知識が必要となります。熟知している人はかまいませんが、不安のある方はこれを機にIPv4をおさらいしてみてください。

今回は、IPv6プロトコルの要点とIPv6プロトコル／アドレスにまつわるエトセトラをお届けする予定です。お楽しみに！SD

連載を通じての質問やコメント、取り上げてほしいトピックを募集します。最終回などで取り上げて、追加解説したいと考えています。質問、コメントは編集部(sd@gihyo.co.jp)までお送りください。

## COLUMN



## 情報は鮮度が大事!

IPv6に限った話ではありませんが、一般的に情報は鮮度が大事です。枯れていない技術では、なおさらです。SDN、OpenFlowなど最近出てきた技術の情報であれば、みなさんも変化が速いことを心得ているかと思います。

かたやIPv6は、標準化されてから15年以上経ちます(1995年12月にRFC1883 Internet Protocol, Version 6(IPv6) Specificationが公開されました)。15年以上経つにも関わらず、最近でも実利用に影響する標準化が行われています。

- ・2010年8月 IPv6のアドレス表記に関わる仕様(RFC5952)
- ・2012年4月 デュアルスタック環境での遅延軽減に関する仕様Happy Eyeballs(RFC6555)
- ・2012年9月 IPv6のアドレス選択に関わる仕様(RFC6724)

また、当初、IPv6はIPv4よりセキュアだと言われていました。これはIPv6ではIPsecを標準搭載にしたことに起因します。しかし、IPv4においてもIPsecを利用できますし、IPv6においてIPsecを利用することが必須ではなく、現状、利用も限定的であるため、実用上はIPv4と変わりません。そのため今では、「IPv6がセキュアということは幻想である」という風潮です。

最近でも変更が行われていることはインターネットの根幹をなすIPという基盤技術の世代交代が、いかに難しいかを表しています。

本文中でいくつか教育に関する情報を紹介していますが、テキストはその時点での最新情報を記したものですので、最新の情報を押さえて、更新された部分を把握するようにしましょう。また、今後も変化が予想されますので、IPv6の動向に注視しましょう。



## パーソナルクラウドで自動化三味

中井 悦司  
NAKAI Etsuji

レッドハット(株)  
グローバルサービス本部プラットフォームソリューション統括部  
シニアソリューションアーキテクト



### 恵比寿からこんにちは

恵比寿にある「世界最大のオープンソースの会社」(の東京オフィス)からお届けする連載も順調に回を重ね、ついに(!)筆者の順番がやってきました。

これまでの執筆陣の個性的な記事の数々に、少し「ハードルが上がった感」がありますが、負けずに素敵なオフィス生活をお伝えしたいと思います。



### やっぱり気になる「業務用PC」

筆者がRed Hatに転職して、恵比寿のオフィスにやってきたのは、昨年の夏ごろでした。IP電話がボツンと置かれたデスクに、前職のオフィスから抱えてきた大量の書籍を積み上げつつ、業務用PCが支給されるのをワクワクしながら待っていると、最初にやってきたのは、「Think Pad x220」と「23インチディスプレイ」でした。

「プロのためのLinuxシリーズ」の最初の2作を17インチ(時々15インチ)のディスプレイで書き上げた筆者は、広大なディスプレイの前に

して、「これで書く3作目は、きっと壮大な内容になるに違いない……」と妙な期待を膨らませてしまいました。Red Hatでは、オープンソースに関わる内容であれば、書籍や雑誌記事の執筆、あるいは、コミュニティイベントでの発表なども業績評価の対象として認められています。コミュニティなくしては成り立たない、ユニークなビジネスモデルの企業ならではのかもしれません。

もう1つちなみに、ThinkPadに接続するキーボードは、愛用の「HHKB Professional 白/無刻印」——シリアルナンバ「000111」の逸品です。給湯室にキーボード掃除用の洗濯ネットが置いてあるなど、キーボードへのこだわりが高いメンバもいるようです。

そして、さらに数日後にやってきたのが、最新のIntelプロセッサを搭載したタワー型サーバでした。日々、ソースコードを読んだり、書いたり、ビルドしたりを生業にしているメンバも多く、そのために、各人に専用のサーバが支給されているわけです。筆者の場合は、ソースコードの読み書きに加えて、さまざまなオープンソースの機能検証やデモ環境の構築にも利用しています。



### 仮想マシン作り 放題の罠

支給されたサーバには、当然、「Red Hat Enterprise Linux 6」をインストールして、Linux KVMで「仮想マシン作り放題」にするわけですが、ソフトウェアの機能確認やデモ環境に使用する場合は、少し注意が必要です。

同じ仮想マシンを複数の目的に使いまわしていると、以前の設定が残っていて、想定外の動きに悩んだり、セミナー会場で「さあデモを見せるぞ!」という際に、予定外のデモが起動してきたりします。

このようなトラブルを避けるために、基本的には、毎回、新しい仮想マシンを作成して、「キレイなカラダ」で検証やデモを行うようにしています。もちろん、同じ環境を何度も手作業で作

るのは馬鹿ばかしいので、環境構築の自動化に取り組むことになります。

最初は、シンプルにKickStartで自動構築を行っていたのですが、ある日、突然に思い立って、「libvirt(virt-install)、KickStart、Puppet、Git(GitHub)」を組み合わせた、本格的な自動化環境を構築してしまいました。



## 「パーソナルクラウド」へようこそ

完成した環境の全体像は、図1のとおりです。図にすると「カッコよさげ」に見えますが、しくみそのものはそれほど複雑ではありません。

- ① virt-install コマンドで仮想マシンを作成して起動する
- ② KickStartでRHELを自動インストールして、
- ③ ④の処理を行うスクリプトを実行する
- ③ 事前にGitHubにアップロードしておいたPuppet マニフェストをダウンロードする
- ④ Puppet マニフェストを適用して、アプリケーションの導入・設定を実施する

これらをまとめて実施するPythonスクリプトを作成して、利用しています。ちなみに、GitHubを知らない読者は『Gitポケットリファレンス<sup>注1</sup>』を参考にしてください。Puppetがわからない読

者は……、えーと、本誌の2012年10月号で特集していた「Chef」に類似の構成管理ツールです。

ポイントは、アプリケーションを導入・設定する元ネタとなる「Puppet マニフェスト」がGitHubでバージョン管理されているところです。機能検証やデモ環境では、微妙に設定の異なる類似環境を複数用意することがよくあります。それぞれの設定にGitHub上で「タグ」を振っておき、先のPythonスクリプトでは、適用する設定の「タグ」を指定できるようにしてあります。

使い始めてみると、なかなかこれが快適で、密かに「パーソナルクラウド」と名付けて愛用しています。詳細が気になる読者は、筆者のブログ<sup>注2</sup>も参考にしてください。



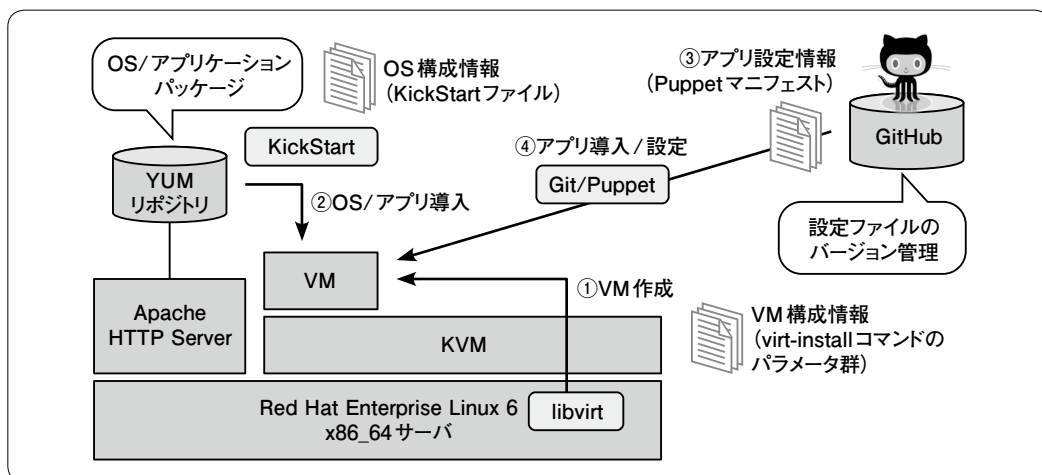
## さ〜て、来月の恵比寿通信は？

実は(!)「世界最大のオープンソースの会社」のオフィスは東京だけではなくありません。今回は、大阪のオフィスから「レッドハットなにわ通信」をお届けしたいと思います。お楽しみに。んがぐぐ。SD

注1) ISBN978-4-7741-5184-7 岡本隆史、武田健太郎、相良幸範(著)/技術評論社

注2) ブログ(<http://d.hatena.ne.jp/enakai00>)内のタイトル「アプリケーション環境構築の自動化をまじめに考えてみる」の一連の記事。

▼図1 「パーソナルクラウド」の全体像



Text : 青田 直大 AOTA Naohiro

Linuxにはすでに10以上のファイルシステムが備わっていますが、それでも新しいファイルシステムが提案されているのを見ることがあります。最近ではF2FSとLanyFSというものが提案されています。これらのファイルシステムはどちらも「特定のデバイス」をターゲットにしたファイルシステムです。今月はF2FSについて見ていきます。



## F2FS とは

F2FSは、その名前のFlash Friendly File Systemからもわかるように「NAND Flashメモリのストレージデバイス用にデザインされた新しいファイルシステム」<sup>注1</sup>です。F2FSは2012年10月5日に最初のバージョンが投稿されました。10月23日にはさまざまな修正と細かい機能を追加したバージョン2が、そしてさらに10月31日にはバージョン3が提案されている、というように積極的な開発が続けられています。

NAND Flashメモリのストレージデバイスというのは、たとえばSSD、eMMC、SDカードのようなデバイスです。これらのデバイスは、従来一般的だったHDDとその性質が異なります。HDDではデータは磁気ディスクに格納され、そ

のディスクを所定の位置まで回転して、データを読み書きします。一方でSSDなどでは、Flashメモリにデータを記録しているので、“データの読み書きのためにディスクの回転を待つ”といった処理は不要になっています。また、逆にFlash型のデバイスは、新しいデータを書き込む前に一度古いデータを削除する必要がある、この削除の操作にはそれなりの時間が必要であり、しかも削除の回数に制限がついています。

これまでの多くのファイルシステムは、HDDの特性に合わせてデザインされており、たとえば磁気ディスクの回転量なるべく減るようにディスクレイアウトを調整したりしていました。また、これまではFlashメモリをターゲットにしたファイルシステムでも、削除回数の制限に対応するために、ディスク全体でなるべく書き換え回数が同じになるようにする「ウェアレベリング」が実現されるように工夫がなされていました。

最近のFlash型のデバイスではデバイス側に、Flash Translation Layer (FTL) と呼ばれるソフトウェアが組み込まれています。このFTLがOS側が指定したアドレスを実際の物理アドレスに変換します。この変換の過程で、ウェアレベリングを行ったり、あるいは書き換え時に今までとは違った場所に書き込むことで、古いデータの削除を後まわしにしてすばやく書き込むことができるようになっています。そうすると、ファ

注1) <http://permalink.gmane.org/gmane.linux.file-systems/68580>



イルシステム側で同じ目的のコードを持っているのは無駄になってしまいます。そこで、F2FSではFlashをターゲットにしつつも、FTLがあることを意識してその機能を活かせるような作りになっています。



## F2FSの構造

F2FSはディスク全体を2MBずつの「セグメント」に分割します。そして、いくつかのセグメントが集まって1つの「セクション」になり、いくつかのセクションで1つの「ゾーン」を構成します。セグメントが読み書きの最小単位になります。セクションはGCを実行する単位です。F2FSでは、これがFTLのGCのものと同じサイズになっていることを期待しています。ゾーンについては後ほど説明します。

ディスクは大まかに次の6つのエリアに分割されています。

- Superblock (SB)
- Checkpoint (CP)
- Node Address Table (NAT)
- Segment Info. Table (SIT)
- Segment Summary Area (SSA)
- Main

Superblockは、1つのセグメントからなっているエリアです。ほかのファイルシステムと同様、そのファイルシステムの情報や、ディスクレイアウトを決定するパラメータが保存されています。Mainはその名の通り、ファイルシステムのデータを書き込むメインのエリアです。このエリア

のサイズはゾーンのサイズの倍数になるように調整されています。その他のエリアはメインエリアのデータを管理するためのメタデータを書き込む部分です。これらは順次紹介していきます。



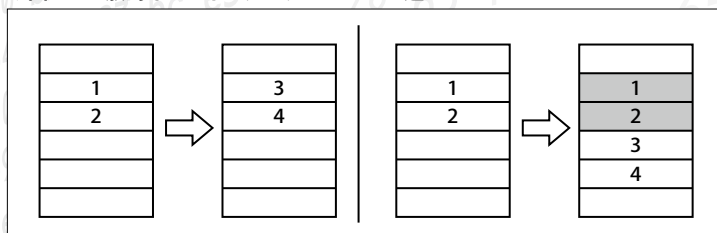
## ログ構造ファイルシステム

F2FSはLFS(Log-structured File System : ログ構造ファイルシステム)をデザインのベースとしています。ログ構造ファイルシステムのしくみを、ファイル書き換えのときを動きを見ながら解説してみましょう。ファイルを書き換えると、一般的なファイルシステムでは図1の左側のようにデータブロックがそのまま同じ場所に上書きしています。一方、LFSでは図1の右側のようにデータを同じ場所に上書きすることなく、書き換えの情報を「ログ」という形で新しい場所へと追記していきます。

もしも書き換えの途中で電源断が起これ、書き込みが中途半端になってしまったとします。一般のファイルシステムでは、上書きデータが中途半端でデータや、場合によっては直接は関係のなかった部分までシステムの構造が壊れてしまうことがあります。LFSでは上書きをしていないのでこのように壊れることはありません。また、ログはディスク上で連続した場所に書き込まれるため、ディスク上で離れた場所に書き込むよりも速くなります。

もちろんLFSもいいことばかりではありません。一般のファイルシステムでは、あるファイル番号に対応するデータがどこにあるのかを管理するのに、たとえば図2のような構造を使います。ファイル番号に対応するブロック番号を

▼図1 一般的なファイルシステムとLFSの違い





管理するテーブルを作ります。1つのブロックの大きさを4KBとすると、このテーブルだけで1GBのファイルを表現しようとするだけでも262,144ものエントリがファイルごとに必要になってしまいます。これではあまりにも非効率ですね。

そこで、いくつかのエントリを間接参照のエントリとします。そこで指定されたブロックには実際のファイルデータではなく、ブロック番号のリストが書かれています。さらにここに書いてあるブロック番号を間接エントリのリストとすれば、最初のテーブルのエントリの数は少なくとも大きなサイズのファイルを保持できるようになります。たとえば、最初のエントリの数が10であっても、5つを直接データを指すエントリ、3つを間接参照のエントリ、残りの2つを2段参照、3段参照のエントリとして1つのブロックに1,024個のエントリが書けるとすれば、

$$4KB \times (5 + 3 \times 1024 + 1024 \times 1024 \times 1024 \times 1024 \times 1024) = 4100GB$$

となって、約4TBのファイルまで表現できるようになります。

これをそのままLFSに当てはめると困ったこ

とがおきます。3段参照されているブロックのデータを書き換えることを考えます。LFSでは上書きはしないので、データが保存されるブロック番号が変わります。すると、当然そこを参照していたエントリも書き換えなければいけません。これも上書きしないので、別の場所に保存されブロック番号が変わります。と、このように1カ所の書き換えをしたはずが、4カ所の書き換えが発生してしまいます。これを「Wandering Tree Problem」と呼びます。

また、もう1つわかりやすい問題があります。ディスクの容量は無限ではないので、上書きをせずにいつも新しい場所に書き込むことは現実的には不可能です。LFSではこの問題に対応するために、データを読むために必要ではなくなった領域をチェックして、解放し再利用できるようにするGarbage Collection(GC)を実装しなければ使いものになりません。

Wandering Tree Problemへの対応とGCの実装、この2つがLFSベースのファイルシステムを作るうえでの肝となります。では、F2FSがこの2つにどのように対応しているのかを見ていきましょう。

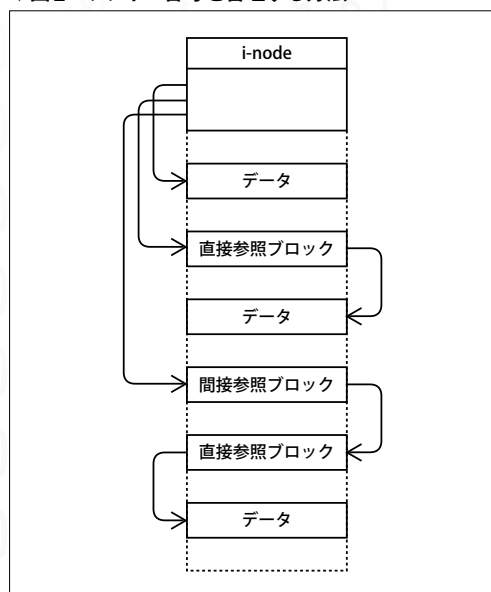


## Node Address Table

F2FSではディスク上に「i-node」「直接データを参照するエントリが列挙されたノード」「間接的にデータを参照するノード(直接データを参照するノードが列挙されている)」の3つのnodeがあります。「i-node」は、932個のデータを指すエントリ、2つの直接データを指すノード、2つの間接的にデータを指すノード、1つの二重間接参照のノードがあります。直接参照ノード、間接参照のノードはそれぞれ1018ずつのエントリを保持できます。よってF2FSでは

$$4KB \times (932 + 2 \times 1018 + 2 \times 1018 \times 1018 \times 1018 \times 1018) = 3.94TB$$

▼図2 ファイル番号を管理する方法





と、3.94TBまでのファイルを扱うことができます。

前述したように、このエントリにブロックのアドレスをそのまま書いてしまうと、Wandering Tree Problemが発生します。そこでF2FSではNode Address Table(NAT)というものを、Mainとは別のエリアにおいて管理しています(図3)。i-nodeなどのノードにはブロックのアドレスの代わりにノード番号を書きます。そして、このノード番号を使ってNATを引いて実際のブロック番号を求め、データを読み出しています。こうしておけば、どこかのデータが書き換えられても、NATの中の値を書き換えるだけでWandering Tree Problemを回避できます。

NATに変更があっても、すぐにディスクへの書き込みは起きません。F2FSでは定期的に“Checkpoint”をディスクへと書き込みます。Checkpointはその時点でのディスクのスナップショットを構成するためのメタデータのようなものです。Checkpointを書き込む前に変更されたNATデータが書き込まれます。NATテーブルは図3のように、2つのセグメントで1つのペアになっており、前回書き込んだものとペアになるセグメントにデータが書かれます。そして、どちらのペアに書いたものが有効かをbitmapとしてCheckpointの中に保持します。NATのセ

グメントがペアになっていることからわかるとおり、Checkpointは2つまでしか保持されていません。もしも、NATの書き込み中や、書き込みからCheckpointの書き込みまでの間にシステムの電源が落ちたとしても、もう片方のCheckpointの時点まではroll-backできます。



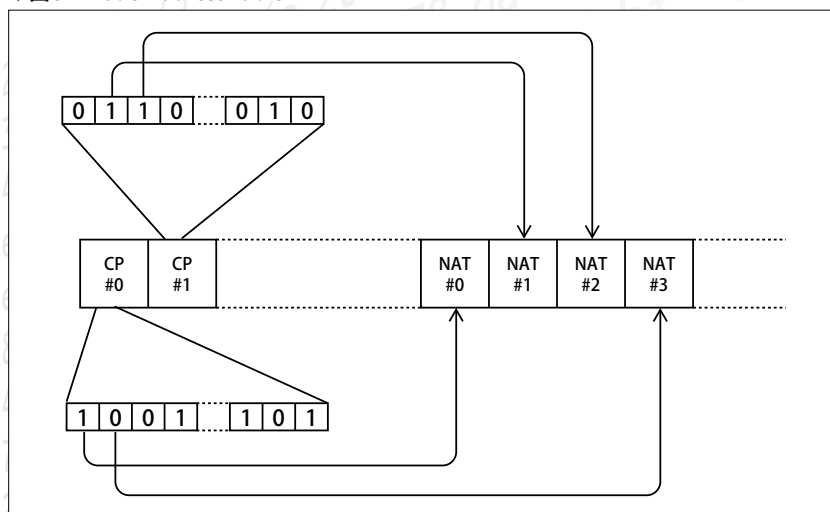
## GC

F2FSには2種類のGCがあります。1つはシステムがアイドル時にバックグラウンドで走るGC、そしてもう1つはmkfs時に設定されるセグメント数よりも空きセグメント数が少なくなったときにオンデマンドで実行されるGCです。これら2つのGCでは、それぞれ違った戦略を採用しています。

オンデマンドに実行されるGCはgreedy(貪欲)にターゲットとなるセグメントを探索します。単純に20個のセグメントの中から、そのセグメントが所属するセクション内の有効なブロック数が一番少ないものを選択し、そのセグメントに対してGCを実行します。ここで所属セクションを見ているのは、GCがセクション単位で行われるからです。

アイドル時にはより複雑にコストベネフィット

▼図3 Node Address Table





を計算してGCのターゲットとなるセグメントを選択します。この計算には2つのパラメータ「そのセグメントが所属するセクション内の有効なブロックの割合」と「そのセグメントが所属するセクションの平均生存時間」とが使われます。「生存時間」と書きましたが、より厳密には「(<平均生存時間> - <全セグメントの最小生存時間>) ÷ (<全セグメントの最大生存時間> - <全セグメントの最小生存時間>)」が使われています。これは全セグメントの中で最長の生存時間を持つものは100、最小の生存時間であれば0となる値です。

要するに、生存時間が長くなるほどこれからも生存する確率は高く、GCするコストは高く、ベネフィットは低くなります。同様に「有効なブロックの割合」も明らかに高いほどGCするコストは高く、ベネフィットは低くなりますね。

ターゲットが決まればそのセクション内のノードデータに対して、その有効性をチェックします。有効なノードデータに対してdirtyフラグを立てて、有効なものが新たな場所に書き込まれるようにします。バックグラウンドでGCが走っている場合は、dirtyフラグを立てるだけで実際の処理はdirty pageのflusherに任せます。オンデマンドでGCが走っているときには実際に新たな場所への書き込みが終わるまで待機しています。

さらにF2FSのGCはもう1つ工夫がされています。空き容量がより少なくなっているときには、Slack Space Recycle(SSR)というモードになりLFSのルールを破って直接そのブロックで上書きが行われます。



### multi-head log

F2FSではほかにも変わった機能が備わっています。F2FSはノードとデータをそれぞれHot、Warm、Coldに分類しています。これはそれぞれ、

- Hot ノード：ディレクトリへの直接参照ノード
- Warm ノード：Hotではない、直接参照ノード

- Cold ノード：その他のノード
- Hot データ：ディレクトリのデータ
- Warm データ：HotでもColdでもないデータ
- Cold データ：マルチメディアデータ、GCで移動されたデータ

というように配分されています。Hotなものほど変更が激しく、そのため無効になるまでの時間が短いことが予想されます。F2FSでは同時に最大6ヵ所(なるべく別々のゾーン)でログストリームが有効になっています。これらのログストリームが上記の6つのデータノード分類に割り当てられています。このように最初から分類しておくことで、生存時間が短いものは短いもの同士、長いものは長いもの同士で同じセクションに配置されやすくなります。すると、生存時間が短いものがまとまっているセクションではより多くのセグメントがGCされやすくなりますので、GCの効率を向上できます。

さて、ここで「マルチメディアデータ」というものがあることに気がつかれたでしょうか。F2FSはマルチメディアデータを、Coldで生存時間が長いものとみなしています。では、いったいF2FSはどうやってファイルをマルチメディアデータだとみなしているのでしょうか。F2FSではファイルシステムを作成する際に、最大で64個の拡張子リストを指定できます。デフォルトでは、これに“jpg”、“png”、“mp3”、“ogg”などが指定されており、これがファイルシステムのSuperblockにも保存されます。この拡張子リストにマッチする拡張子を持ったファイルが“Coldデータ”として分類されます。このあたりは、もとの開発がSamsungだけあって、モバイルでの利用を意識しているのではないのでしょうか。



### ディレクトリ

F2FSはディレクトリの構造も少し変わっています。図4の左側のようなディレクトリは階層化された「dentry block」の入ったbucketの集



合で表現されています。この中には最大で214個のdentryとそのファイル名とが保持されます。ファイル名は図4の右側のように配置されるので、ファイル名が8文字を超えているとほかのdentryの場所をつぶしています。

dentryには、ファイル名のハッシュとそのファイルのi-node番号、ファイル名の長さ、そしてファイルの種類(symlink、ディレクトリ、通常ファイルなど)が記録されています。ここからファイルを探すときは、まずファイル名のハッシュをとります。まずはレベル0のbucketのdentry blockの中からシーケンシャルに、同じハッシュファイル名を持つdentryを探します。もし、ここでそのようなdentryが見つからない場合、次のレベルへと進みます。レベル1では、ハッシュ値のmod 2(2で除算の余り)を取り、それが0(ハッシュ値が偶数)であれば0番のbucket内のdentry blockから、1(ハッシュ値が奇数)であれば1番のbucket内のdentry blockからレベル0と同様にdentryを探します。このように、ハッシュ値をそのレベルのbucket数でmodを取り、その値のbucketからdentryを探していきます。



## パフォーマンス

LKMLにはさっそくF2FSのパフォーマンス

テスト結果がポストされています<sup>注2</sup>。シーケンシャルな書き込みではEXT4の1~1.5倍ほどの速さですが、ランダムな書き込みでは、1.5~2.5倍のIOPSを出しています。さらに、読み込みの速度でもEXT4を上回っており、なかなか期待できる結果になっています。



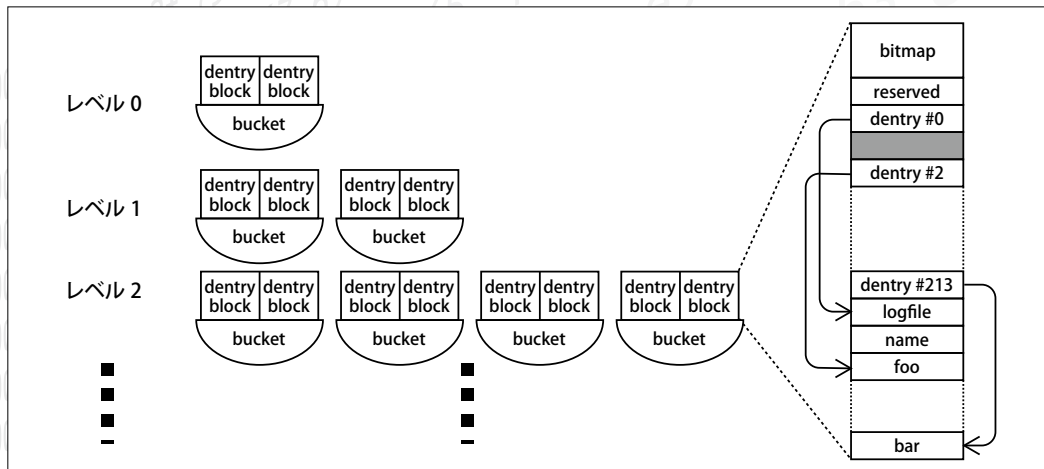
## まとめ

今月はF2FSについて紹介しました。マージされていないどころか、提案されたばかりの、しかもファイルシステムということではなかなか実用もテストもためらわれるところがあります。しかし、アクティブなログを同時に複数持つなど、HDDでは逆に遅くなってしまうことが予想されるほどに、昨今のFlashストレージ専用にチューンされていたり、拡張子でマルチメディアデータを判別したり、となかなか変わった機能を持ったおもしろいファイルシステムなので多少のリスクはあってもバックアップ体制をしっかり整えて使ってみようかな……と著者は考えています。

SD

注2) <https://lkml.org/lkml/2012/10/22/184>

▼図4 F2FSのディレクトリ構造



January 2013

No.15

## Monthly News from

jus  
Japan UNIX Society日本UNIXユーザ会 <http://www.jus.or.jp/>  
今泉 貴史 IMAIZUMI Takashi [image@jus.or.jp](mailto:image@jus.or.jp)

## インターネットの未来をかいま見たIC2012

## ■インターネットコンファレンス2012 (IC2012)

【日時】2012年11月15日(木)、16日(金)

【会場】富山国際会議場

去る2012年11月15日と16日に、富山国際会議場にてインターネットコンファレンス2012が開催されました。本会は1996年からjusを含む複数の組織で共催しているインターネットに関する研究の発表会です。今年はjusに加え、日本学術振興会産学協力研究委員会インターネット技術第163委員会、日本ソフトウェア科学会インターネットテクノロジー研究会、WIDEプロジェクトで共催しました。参加者は60名でした。2つの招待講演や9件の研究発表に加え、ポスター展示が14件も行われ、そのうちの4件がデモンストレーション展示を行っていました。

例年、本会では優秀な論文／発表を表彰しており、今回は次の論文が論文賞を受賞しました。

- 「IPv4アドレスが重複しうる複数LANに対してクラウド連携を可能にするL2マルチテナント方式の提案」鈴木和宏、今井祐二、下國治、福井恵右(富士通研究所)

また、優秀なポスター発表をした発表者に送られるポスタープレゼンテーション賞には次の2者が選ばれました。

- 「電力消費変動によるタップ接続機器の動的識別」池上洋行(東京大学)

- 「河川水位観測に基づく河川リスク情報の生成と提供」廣井慧(慶應義塾大学)

学生による優秀な論文発表に対して与えられる学生奨励賞は、次の2件が受賞しました。

- 「MapReduceによる大規模分散システムのシミュレーション」杉野好宏(東京工業大学)
- 「事業体における電力データ見える化システムの導入と節電効果の検証」松井加奈絵(慶應義塾大学)

以下、受賞した論文や招待講演の中から何件かを紹介します。

## 受賞論文

## ■MapReduceによる大規模分散システムのシミュレーション

大規模なネットワークシミュレーションを行うためにMapReduceを用いて大規模分散システムとしてシミュレータを構築する手法を提案したものです。ノードはkey-valueペアのkeyとして表現し、Reduce処理をメッセージの生成に、Shuffle処理をメッセージの送信に割り当てることで、メッセージの送受信を表現しました。現在、1億ノードでのシミュレートが可能になっているとのことです。

- IPv4アドレスが重複しうる複数LANに対してクラウド連携を可能にするL2マルチテナント方式の提案  
マルチテナント方式でVirtual Private Cloudを提

供する場合に、組織とL2接続することによりIPv4プライベートアドレスの重複が問題になる場合があります。これをIPv4アドレスをIPv6アドレスにマッピングするとともに、アドレススコープを用いてアプリケーション側で区別できるようにすることで問題を解消します。実際にいくつかの実装も行い、オーバーヘッドやスケーラビリティに関する評価も行っています。

### ■電力消費変動によるタップ接続機器の動的識別

消費電力測定が可能なスマートタップを用いて電力の消費量を監視することで、その先に接続されている機器を特定しようとする手法についての発表でした。瞬間電力量を得られる場合にはその値から、積算電力量を取得できる場合にはそれを微分することで、機器の電源のON/OFF状態を取得します。機器側で決められたパターンで電源のON/OFFを行うことでタップ側で認識できるというしくみです。実機を用いた実験では通知できる情報量は0.01bps程度とのことで、現在のままではまだ実用的とは言えませんが、今後への展望が広がる発表でした。

### ■事業体における電力データ見える化システムの

#### 導入と節電効果の検証

電力消費データの見える化システムを活用した場合にどのように節電行動がとられるのかについて実験した報告が行われました。これまでの即時性の低い見える化と比較し、インターネットを活用した即時性の高い見える化を行った場合に非常に良い結果が得られましたが、実際には電力会社からの節電要請などもあったため、正確なデータは取得できなかったとのことです。

### 招待講演

富山県立大学の岡田敏美教授による招待講演です。最近、高齢者の登山者が増えて遭難などの発生も増加しており、その対策として開発したビーコンに関するお話でした。遠隔地から遭難者を探索するため

には強力な電波を発生する必要がありますが、電波を送出するために免許が必要になり誰もが利用するのは困難です。そこで、法律上許される電波強度でどの程度の検知が可能なのかを調べつつ、法律の改正を働きかけながらより有用なシステムを構築してきた流れが紹介されました。また開発したデバイスは、単にビーコンとして利用するだけでなく双方向通信も可能になっており、簡単な問い合わせもできます。発表では、システムに関してももちろんですが、きれいな山々の写真などもふんだんに使われ、富山のアピールもしっかりと行われていました。

### ■ユビキタスプラットフォーム実現に向けた

#### インテックの研究開発

インテックの堀雅和氏による招待講演です。インテックは富山市に本社を置くSIerですが、その歴史についての話の中では富山の薬売りから話が始まるなど、こちらも富山を意識した講演となっていました。インテックはSIerとしては珍しく研究所を持っており、幅広い研究活動が行われています。タイトルにあるユビキタスプラットフォームは、インテックが考える将来のITシステムの基本となるもので、ユビキタスUI、M2Mコア、ビッグデータ処理、運用管理、開発支援を組み合わせて構築しています。富山市が環境未来都市に選定されたということで、そのコンセプトや概要について紹介されましたが、その中でユビキタスプラットフォームをどう展開していくのかなどの話がありました。

### 終わりに

のWebサイト<sup>※1</sup>から参照できます。また、来年度は九州産業大学の神屋郁子氏を実行委員長に迎えて開催の予定とのことです。SD

注1) URL <http://www.internetconference.org/ic2012/>

# Hack For Japan

エンジニアだからこそできる復興への一歩

Hack  
For  
Japan

第13回

## IT Bootcamp座談会： 石巻の高校生との学びの場で感じた支援のかたち

“東日本大震災に対し、自分たちの開発スキルを役立てたい”というエンジニアの声をもとに発足された「Hack For Japan」。本コミュニティによるアイデアソンやハッカソンといった活動で集められたIT業界の有志たちによる知恵の数々を紹介します。

### 座談会 メンバー 紹介



古山 隆幸  
(ふるやま たかゆき)氏

石巻2.0理事/イトナブ石巻代表。  
石巻にIT産業を根付かせるために、若者を中心にソフトウェアやさまざまなIT関連の学びの場をつくる活動をしている。



山本 直也  
(やまもと なおや)氏

日本コロナの会にて、Corona SDKのエバンジェリストとしてボランティアでコミュニティ活動中。携帯電話のFlashプレーヤの開発を10年くらいやっていた。



小野 哲生  
(おの てつお)氏

Corona SDKアンバサダー。(株)ナスピア勤務。元建築デザイナー。iPhoneが日本で発売された翌週から使いはじめ、自分でアプリを作りたいと思いCoronaに出会う。



原 亮  
(はら りょう)氏

みやぎモバイルビジネス研究会会長。2009年以来、宮城でモバイルインターネットを活用した新しいビジネスを作るために活動。震災後はFandroid EAST JAPANという団体を仙台の仲間と設立。



佐々木 陽  
(ささき あきら)氏

会津若松に社屋を構える(株)GClueの代表取締役。Android/iOSアプリケーション開発が主な事業。未来の主戦力となるエンジニアを育てるために、大学生などに教える活動を10年間行っている。



及川 卓也  
(おいかわ たくや)氏

Hack For Japanスタッフ。Google勤務。震災以来各地で活動を行っているが、今年に入って被災地、とくに若い方々にITでなにか元気を与えたいという思いがあった。座談会にはGoogle+ ハングアウトで参加。



高橋 憲一  
(たかはし けんいち)氏

Hack For Japanスタッフとして、震災以降ITでできる復興支援を考えて活動している。普段はエンジニアとしてiOSやAndroidのアプリ開発を行っている。今回の座談会の進行役。

これまで数々のアイデアソンやハッカソンを行い、被災地の支援にITをどのように生かせるかを模索し続けているHack For Japan。2012年7月に開催した石巻ハッカソン(イトナブ石巻主催)での「IT Bootcamp」(以下、Bootcamp)で、石巻工業高校の生徒さん10名にプログラミングを学んでもらう3日間の講習会を行いました。今回は、Bootcampに参加したスタッフ&講師陣に集まっていただき、その成果などについて意見交換を行いました(本文中敬称略)。

### Bootcamp開催のいきさつ

古山▶石巻にはITエンジニアが少ない、というかそもそもIT産業がほとんどないんです。若い人にIT産業の魅力を伝えるににくい環境で手取り早くわかってもらうには、そこの最前線で活躍している方々をお招きするのが良いだろうと考えました。

そこで原さんにご相談し、佐々木さんをご紹介いただき、そこからここにいらっしゃる皆さんを集めていただ

いたという経緯になります。自分としては、今年一番の盛り上がりのあるイベントになったのではないかと考えています。

原▶古山さんとは石巻でどうやってIT産業を興すか、そこで人を育てるにはどうしたらいいか、といったことを2012年の3月くらいから何度も相談していました。

一方で、震災があってからHack For Japanをはじめ、エンジニアの方々が東北に何か支援できないかと活動されているのも知っていました。そこで重要になるのは、両者のマッチングをどうするのが良いかです。

今回の場合は本来あるべき姿、つまり地域側がニーズを持っていたところに、この活動がつけられたのが良かった。

### 継続性を与えるセミナーにするために

高橋▶勉強会やセミナーの開催経験が豊富な佐々木さんから見て、Bootcampを企画するにあたって何か思うところがありましたか？

佐々木▶やるからには勝算のある教育をしないと継続性が出なくなってしまうので、なんらかの優位性を与える必要があるんですね。モチベーションを高めることで、あとは自分で継続的に勉強してくれるようなモデルが必要だと考えていました。

それで今回の話を考えるに、高校生にいきなりAndroidやiOSを教えるとなると敷居が高くてモチベーションがぐっと下がるかもしれないという心配はありました。そんな心配事を考えているうちに、Coronaだったらその敷居を下げるができるかもということ、山本さんに協力を仰いだわけです。

山本▶Coronaはたいてい1日のセミナーで「こんなふう」に作れます、皆さんおもしろいからやってみてね」といった紹介だけなんです、それでも小野さんみたいに興味を持ってくださる方が自力ではじめてマーケットで販売するところまでいける。この敷居の低さが魅力だと思います。

小野▶プログラミング経験のなかった自分がiPhoneアプリを作りたいと市販の本でObjective-Cとかやってみたんですが、完全に挫折をした立場でし

て。だからこそCoronaの良さがわかります。それが高じてCoronaを広めるアンバサダー（大使）になっちゃったくらいですから。

### Bootcamp成功の要素

山本▶最後の日に、半日かけてなにかおもしろいものを1本作ってみると言ったら、みんな一所懸命に頭をひねりましたよね。「こんなスケッチでやりたいですけど」と講師陣に見せに來たりしました。みんなのアイデアもおもしろかったですし、実現できた内容もおもしろかったですね。

佐々木▶今回プログラムができる、できないは別として、生徒の皆さんセンスが良くて、優秀な学生さんが集まったかなと。

小野▶どんどん質問が出てきて、こっちが逆に本気にさせられたというか、そういうのはありましたよね。

原▶質問を受けたりしていると、教えている側の本気度が上がっていく。うまくいく学びの場ってそうなんですよ。

成功の要素は3つくらいあると思っていて、1つは少ないコードで完成の感動が味わえるCoronaという武器を手に入れたこと。2つめは教えたメンバーがそれぞれの分野で活躍するプロだったということ。3つめは3日間という制約の中で、作り終わったあとの発表の場に向けて時間と戦ったこと。

これらの要素がうまく重なったので、その場の熱量もすごく上がったし、もっともっとやってみたいという彼らの継続意欲にもつながったのかなと。さらにはBootcamp終了後にも、生徒さんたち自身がチームを作ったり、山本さんがオンラインで教えてくれるといった関係がちゃんとできていたり、すべてうまく回った感じがします。

(一同納得)

高橋▶10人の生徒に5人講師がつくという、なかなか贅沢な環境だったかなと。

佐々木▶先生がつきすぎると、つきっきりで全部教えちゃうっていうケースも起こりがちなんです。でも、今回はこれだけ先生がいるのに、先生が適度なバランスで教えていたというのが良かったですね。

## 教え方と修羅場の経験

高橋▶でもそこはちょっと……及川さんと反省点として話していたんですが、時間のあうちは生徒さんに考えてもらえるようにうながしていくんですが、やっぱり最終日の時間が迫ってきたときには、「先にもうコード言っちゃうよ。はいこれ、タイプして」みたいな感じになっちゃって。動いたあとに意味を説明しましたが、ちょっとブッシュしすぎたかなという反省があるんですね。

佐々木▶ソフトウェア開発では時間制約を守ることが非常に重要なので、逆に言うとそれで良かったんじゃないかと思うんですね。最後に発表できる状態までにするのが最優先。そのあと落ち着いてコードを見て、自分のものにする。とにかく限られた時間の中で、ぎりぎりのところで超えさせるというのが重要だったのかなと思います。そこが今回はよくできたなあと思うところですね。

原▶最後のあの緊迫感とはとてもよかったですよ。生徒さんも講師陣ももうワーッ！と感じで(笑)。あれをくぐらないとダメなんですよ。

## 実機で動いたときの感動

古山▶それから、実機に転送させて、動作するのを体感できたのがすごく現実感を伴っていて良かったのではないかと思います。スマートフォン上で動かしてみて、「おお、動いた！」っていう、あのときの感動を彼らは今でも忘れていなくて、だからこそ継続しているんだと思います。

佐々木▶今回及川さんがAndroidを全部持ってきてくれたというのはキーポイントでしたね。実機で自分の作ったものが動く、そうするとどうしてモチベーションが上がるかっていうと、親に見せられる、友だちに見せられる、っていうのもあると思います。

## アプリ甲子園／ABC東北で見た成長

高橋▶Bootcampの後も生徒さんたちはずっと活動

を続けてくれていて、まず「アプリ甲子園 2012<sup>注1)</sup>」の決勝戦まで行きました。惜しくも入賞は逃しましたけど、大勢いる中ですごくがんばっていたと思うんですよ。私もその会場にいたんですけど、もう子どもを見る親の気持ちでしたね(笑)。

さらにその少し後に、「ICT ERA + ABC 2012 東北<sup>注2)</sup>」(以下、ABC東北)があり、そこでも彼らは高校生トラックでセッションを2つ持って発表してくれました。そこに来ていた及川さんから見て、Bootcampからほんの2ヵ月くらいしか経っていない彼らを見ていてどんな印象を持たれましたか？

及川▶私も自分のプレゼンのときよりも緊張しましたよ(笑)。彼らの中でも中塩成海くんの気合いの入りが全然違って、ちょっとノリが体育会系なんだけれども、それが見ていて気持ちよかったです。アプリ甲子園もそうですが、ほかの高校生と知り合うことがすごく刺激になったと思います。まだまだ自分たちに足りないものがあることがわかって、でも負けないぞ、いつか追いついてやるって言い切っていたところに、すごく頼もしさを感じましたよね。

彼らの発表のあとに話をする機会<sup>むちやぶり</sup>をもらったんですが、そこでこういった高校生のプログラミングの勉強を支えるには、3つの立場がいるねっていう話をしました。

まず1つに素材を提供できる立場の人。たとえば今回はGoogleのほうからAndroid端末の実機を提供しました。それから開発環境(SDK)が無料で使えるものだったということ。今回はさらにCorona Labsのほうから正式ライセンスを10個(参加人数分)提供いただきました。若年層の方がプログラミングを学ぶことに関しては、こういった無償対応や安価での提供というのを続けていかなければいけないのではないかなと。

もう1つは教える立場の人。今後いかにして技術者を増やすかが日本の産業にとって重要になると思うので、我々のようないわゆるプロの技術者はこういった行動を積極的にやりましょうと。今回の講師陣の皆さんが感じているように、教えることによっ

注1) <http://www.applikoshien.jp/>

注2) <http://www.android-group.jp/conference/ictera-abc/>

で学んだことってたくさんあるわけですよね。次の世代を育てるということと、自分自身が育つという非常に良い機会になります。

最後の1つ、学び手の立場である高校生たちには、こういったまわりのサポートを活用して、積極的に学んでどんどん外に出て行ってほしい。高校生ともつながり、我々ともつながり、どんどんつながったまま進めていくのがいいと。

以上のようなメッセージを伝えました。

## 次世代につながる活動

古山▶今回思ったのは、生徒たちがエンジニアの皆さんに教えていただいたじゃないですか。そういうふうに、わかるエンジニアはわからないエンジニアに教える、っていう感覚を彼らは感じたと思うんです。そういう気持ちが整えば、次世代にどんどん還元されていくというフローができてくるのかなと。今回のキープレーヤの中塩くんは、実際にITナブに來ている大学生のスタッフにCoronaを教えてるんですよ。

それから後輩にも教えていこうということで動いています。1年間で終わらせるのではなく、継続的にやっていくことを考えています。

及川▶ABC東北のときも、1年生や2年生を一所懸命誘わなきゃだめだよっていうのと、あとダイバーシティ(多様性)を考えたらぜひ女の子を入れてくれとお願いしましたよ。

## 高校生による地域の活性化

原▶「地方で高校生がやっている」というところの価値にもフォーカスしたいなと思っています。石巻は津波もあってちょっと特殊なところではありますが、いま日本の地方にはどんなところでもさまざまな課題があります。とくに若者が出て行ってしまっ、地域の産業が衰退していくっていうシーンはすごくたくさんあると思うんです。今回、石巻にIT産業を新しく作ろうっていう志の中で、高校生の子たちががんばっている。今後、彼らが地元で仕事が

できるようになっていくっていうのが1つ労働モデルになりうるんじゃないかなって思うんですね。

石巻は石巻工業の子たちとIT産業を生み出して若者離れの問題を解決していく、みたいな、そんなのができると地方活性化のすごく良い例になるんじゃないかなと。

## 復興は現状回復ではない。 未来を考える

—最後に、なぜHack For Japan、あるいはITエンジニアの復興活動として「教育」がテーマになっているのかをうかがってみました。

原▶「復興」という単語をどう理解するか、という話になってくるのかなと思うんですよね。必ずしも被害に遭った何かそのものをどうにかするんじゃないかって、その地域自体がもっと自力を底上げしていくことが必要なんです。震災によって元からあった課題が急速に表出化し、さらに深刻化したんです。であれば、根本にあったその問題を解決しないと、真の意味での復興にはならないよね、っていうことが見えてきました。そこで産業としてのIT、さらには担い手としての人材教育っていう文脈に広がっていったのだと思います。なので、すごく正しいステップアップというか、広がり方をしているのかなと。

佐々木▶IT産業という看板を掲げて、じゃあ何か復興のために作りましょう、ってことになるとその地域の人が求めているものを作ってしまう可能性が高い。とくに補助金がついちゃったものはほんとに悲惨で、地域の人よりもそれを請け負ってマージンを取っていく地域外の企業が喜ぶだけになることが多いんですよ。教育だけはその地域の人を教育するので、補助金が投下されても唯一残るんです。

原▶外から刺激を与えるにしても、その地域の人たちが、「あ、これいいね。やりたいね」っていっしょに動けるものじゃないと、かえって地域の活力を奪っていくことになりかねないんですよ。

高橋▶元に戻すことだけが復興ではなくて、元よりも良くするという、未来に向けた取り組みがこれだと思うんですよ。ひと言で言うと、そういうことになるのかなと思います。SD

# 温故知新 IT むかしばなし

## CPUのバグ

第19回



たけおかしょうぞう TAKEOKA Shouzou take@takeoka.net



### 完全無欠の CPU?

今どきのCPUには、「バグがない」と思われるかもしれませんが、最近のほうが複雑なので、バグがないはずがありません。かなり頭のいい人が手間暇をかけて作っているので「欠陥は少ない」ですが、絶対に問題がないかというところでもありません。



### Pentiumの 計算違い

1994年には、みんな大好きインテルのPentium (P5)の浮動小数点の除算結果が間違っていたことがありました<sup>注1</sup>。マイクロコードで参照する定数テーブルの一部が壊れていたせいだと言われています。電子計算機なのに、計算ができないとは、トホホ感が高いですね。



### 6800と6809 のHCF命令

1970年代のマイコンの世界で有名なのは、モトローラ(現フ

リースケール・セミコンダクタ)の6800と6809の“HCF(Halt and Catch Fire; 命令コード0xDD)”命令です。これを実行すると、CPUが暴走(アドレスバスへ出るアドレスがどんどんインクリメントされ、メモリリードをどんどん行う)して、NMI(NonMaskable Interrupt; マスク不可割り込み)も受け付けなくなり、リセットをするしかありません。Wikipediaにも載るぐらい有名です<sup>注2</sup>。一応、モトローラで、テストのために付けられた命令だとされていますが、ユーザには迷惑な命令です。



### μPD8080AF のSubFlag

ザイログZ80の先祖にあたるインテル8080では、NECは独自にDAA(Decimal Adjust Accumulator)命令を強化していました。それが問題とされ、最終的にNECは、インテルと完全コンパチなμPD8080AFというものを出しました。DAAは「十進補正命令」で、BCD

(Binary Coded Decimal; 2進化10進数)で表現されている2つのデータを、通常の2進数として加算したのち、その結果を補正して正しいBCD表現にするための命令です。

DAAを実行するために、CPUは内部にHalf Carry(BCD 1桁分の繰り上がり)というフラグを持っています。8080では、DAAは加算の直後しか意味を持たない命令でした。8080のバイナリ上位互換のZ80は、同じDAA命令が減算後も使えるように拡張されました。そのために、Z80はCPU内に、SubFlag(直前に減算を実行した)というフラグを拡張しました。Z80のDAA命令は、直前の演算が加算/減算を判断し、補正式を変えています。

NECのμPD8080は、Z80よりも古くからあるのですが、Z80と同様にSubFlagを備え、減算後のDAAが正しく動くように、インテルオリジナルの8080(i8080)を拡張していました。たぶん、μPD8080ではPSW(AFレジスタ)の操作でSubFlagが何かの問題を出したのでしょう。

注1) 結果の精度が異常に低い。IEEE floatでは精度も決められているので、精度がそれに合っていないと結果は誤りとされる。[http://en.wikipedia.org/wiki/Pentium\\_FDIV\\_bug](http://en.wikipedia.org/wiki/Pentium_FDIV_bug)

注2) [http://en.wikipedia.org/wiki/Halt\\_and\\_Catch\\_Fire](http://en.wikipedia.org/wiki/Halt_and_Catch_Fire)



## Z80のNMI仕様

ザイログのZ80は、バグではありませんが、NMI発生時のCPU内部処理について、正式にMOSTEKやNECから公開されていた仕様が、実装と異なっていることがありました(MOSTEKはザイログと正式にセカンドソース契約をしていた。NECは勝手にコンパチ品を作っていた。当時はまだ命令セット・アーキテクチャについて権利主張をする人がいなかった)。おかげで筆者は、Z80のNMIを使用するシステムを開発しているときに、不可解な現象に悩まされ、それに関係あるコードをまったく違う方式に書き換えた、という経験を持っています。この10年ぐらいで、Z80の本当のNMI処理を知って、昔の問題が仕様書に起因することを確信しました。



## ARM6/7の暴走の原因

CPUのバグは8bit時代だけではありません。ARMはARM6、ARM7(ARMv3)に、HFC命令を持っています。筆者は、ARM6が出たとき(1990年代中期)に、世界に先駆けてARM用のOSを開発していました。

ARMの命令セットを眺めると、システムモードからユーザーモードに移るには、いくつかの方法があることになっています。しかし、ARM社のdemonというモニタやその他の開発基本ソフトウェアは、みるからに

効率の悪い命令の組み合わせを使用しています。もっと、効率の良い命令(STMNEIA)が使えるだろうと思って、いろいろと試してみたのですが、CPUが暴走してしまう行きません(システムモードからユーザーモードへの遷移で暴走します)。

ARM6の暴走は、通常の暴走ではなくHCFなのです(NMIも効かない)。結局、“ARMのdemonなどが使っている命令の組み合わせでしかうまく行かないのだろう”ということで落ち着きました。ちなみに、これはARM6から何年も経ったARM7コアでもまったく同様でした。

単なるハードウェアの部品として売られていない、すなわちOSも含めたコンピュータシステムとして売られているCPUでは、システムモードしか使用しない命令については、このようなことがよくあります。



## SPARC、MIPSのMMUの順番

ワークステーションに搭載されているSPARC、MIPSなどは、MMU(Memory Management Unit)操作を行う命令が特定の順番でなければ動かないとかがよくあります<sup>注3</sup>。

MMUの設定はめったに行われず、高速性は不要で、またそれはCPUからするとクリティカルなパスに置かれていることが多く、そういう操作を記述す

る人が極めて少ないためにこういうことがらがおざなりにされているのだと思います。



## ページ境界の制限

また現在のCPUでは、ページ境界に特定の命令を置かないとか、MIPSなどでは、ページ境界にディレイスロットを置かないとかいう制限があるのが普通です。これらは、たいてい、リンカが最終的に調整を行っています(Cをコンパイルしたソースを見ているだけでは、NOPの挿入とかは見られない。リンクの終わった最終的なバイナリを、逆アセンブルして始めてわかることが多い)。

ページ境界になぜ制限があるかといえば、仮想記憶において、ページフォールトを起こしたとき、ページのメンテナンス後の命令の再開が必ずしも完全にできないからです(これらは、OS作成者向けには仕様化されているが、積極的に公表されていないことがよくある)。

インテルi860のような変わったCPUを採用したUNIXシステム(たとえば、沖電気のOKIstation 7300<sup>注4</sup>)では、「C言語でアプリケーションを書いているうちは良いが、アセンブラで書くと、ページフォールトから回復できない命令列を人間が書いてしまい、困る」というようなことがよくありました。SD



注3) BSD系のOSを移植するとき、CPUメーカーの協力を得るか、Linuxのコードを見るか、リバースエンジニアリングするか、しなければ、なかなか動かすことができません。

注4) 情報処理学会コンピュータ博物館【沖電気】OKI station 7300モデル30 <http://museum.ipsj.or.jp/computer/work/0029.html>



## SoftwareDesigner #44

Text=Bart Eisenberg E-mail jaysteller@hotmail.com Translation=嶋崎正樹 SHIMAZAKI Masaki

## コンピュータサイエンス2.0

## [Part 3] :

## 大学教育とオンライン教育



## Daniel Jackson

MIT、コンピュータサイエンス学部 教授

偉業を成し遂げた  
父の教えを受けて

Daniel Jacksonのキャリアのきっかけは身近なところにあった。父親のMichael A. Jacksonだ。彼の父は本を執筆するだけでなく、構造化プログラミング技法のジャクソン法やシステム開発技法のJSD(Jackson System Development)法など、その名を冠した方法論が知られる優れたソフトウェアエンジニアでもあった。Danielは当初、オックスフォード大学で物理学を専攻していた。しかし、ロンドンで就職先を探してみても、やっと「今一番刺激的なのはソフトウェア業界だ」と気づく。「そういうわけで私は少し回り道をしてソフトウェアエンジニアリングの道に入ったんです」と彼は言う。

Jacksonはそこで初めて、父親の偉業を実感するようになった。父親の本を読み、アドバイ

スも受けた。それぞれのプロジェクトに全力で取り組むことが大切だということもその1つである。「父は、表面的なことを理解するだけでは決して満足しようとしなかった。それが父の仕事を特別なものになっている。どんなソフトウェアについても、どういう問題を扱うのか、その核心を見極めようと常に考えているのです。それを見て、私も大学院に戻ることにしました。ソフトウェアについて深く理解することが必要だとわかったからです」。

現在MIT(マサチューセッツ工科大学)でコンピュータサイエンス学部の教授を務めるJacksonは、複雑なソフトウェアは、モデリングを行ってから構築することが重要だと主張する。Jacksonも父と同様に独自の方法論を確立した。軽量形式手法<sup>注1</sup>と、学生とともに開発したモデリング言語Alloyである。しかし、より広い視点で見ると、より深く理解すればより良いプログラ

ムを作成できるという考えが今のJacksonのキャリアと講義の方針を作り上げたと言える。Jacksonは、より深い理解こそが単なるコーディングとソフトウェアエンジニアリングの違いだと考えている。そして、ソフトウェアエンジニアリングは、重要な学問であり真摯に研究する価値があると彼は言う。

→ モデリング言語Alloyを  
開発した背景

— まずAlloyについてお聞きします。開発のきっかけは？

Alloyは、システムには核心がありそれは把握できるという考えがベースになっています。簡潔で精緻なモデルを作れば、システムで発生する最も基本的なイベントを記述できます。そして基本的なイベントに注目すれば、極めて小さなモデルを作成

注1) 形式手法とは、ソフトウェア工学における数学を基盤としたソフトウェアおよびハードウェアシステムの仕様記述、開発、検証の技術のこと(Wikipediaより引用。URL <http://ja.wikipedia.org/wiki/形式手法>)。

**コンピュータサイエンス2.0 [Part 3] : 大学教育とオンライン教育**

できます。レーザーにたとえてもよいでしょう。大きく強力な投光器より、ペンライトのほうが用途が広いのと同じことです。また、モデルで何かやってみたいというのも Alloy を開発した動機の一つです。ハードウェアの計算能力をソフトウェアの解析に活用しようと考えました。コンピュータは驚くほど強靱な能力を備えています。新しいラップトップは、Tバイトのフラッシュメモリ、強力なプロセッサ、16GバイトのRAMなどを搭載しています。でもソフトウェア開発者は、それほど強力なマシンを何に使うのでしょうか。休みの日にビデオを再生したり、Photoshopを使ったりするのが、CPUに最も大きな負荷がかかる処理かもしれません。なにしろプログラマーが使うコンパイラやエディタは、大半が30年も前のマシンで使われていたものと同じだったりするのですから！

— Alloyは伝統的な教育の場で学生とともに作り上げた言語ですが、そのような環境自体は時代遅れではありませんか？若いソフトウェア開発者は今でもそのような場所で学ぶ必要があるのでしょうか。

とても良い質問ですね。基礎技術に精通するためなら、あるいはWebページやWebアプリをハックしたいとか、HTMLやCSSを使ったりRuby on RailsやJavaを動かしたいのであれば、ほとんどは独学が可能

です。しかし、そのようなことをスマートに行いたい場合はどうでしょうか。洗練され、使いやすく、メンテナンスも容易で、信頼性が高いシステムを作ろうと思う場合です。そのためには、また別のレベルの知識が必要になると思います。モジュール性や抽象化など、オンラインで少し勉強しただけではすぐに習得できない概念の数々を使いこなさなければなりません。もちろん、経験を積みれば次第に身に付きますが、何十年もかかります。

そのため、私は正規の教育は基本的なスキルを習得するための近道だと考えています。何かしたいというアイデアを煮詰め、情報に仕上げられる指導者に学ぶことができるからです。今後、教室で得られる経験をオンライン教育で伝えられるかどうかが問題となるでしょう。そのレベルの教育は書籍を使えば可能であることはすでにわかっています。コンピュータサイエンスに限った話ではありません。相対性理論を学ぶなら、アインシュタインが自らわかりやすく書いた本もあります。ですからオンライン教育についてはテクノロジー分野に限定される理由など、もともとないと考えています。とはいえ、オンライン教育には近道ありません。オンライン教育であれば、他人と顔を合わせずに済むかもしれませんが、Malcolm Gladwell<sup>注2</sup>などいろいろな人が言っています。「ほとんどのスキルは習得に何千時間も

かかるし、指導し、学ぶ側の細かい質問に答えてくれる人が必要である」と。

**→ 学部教育と学位の重要性**

— 学位はどれほど重要なのでしょうか。修了証書とそれを得るまでの学習という意味で。

昔似たような疑問を抱えていました。「大学のカリキュラムが個人の知的好奇心にどれだけ合致しているかと比較して、大学の権威はどれほど意味を持つのか」。これは、自分が何をやりたいかによって決まる問題です。新しい事業を興して世界を変えたいと思うなら、どのような形で教育を受けたかなどたいしたことではありません。能力があれば、優れたソフトウェアを開発できるでしょう。反対に普通の企業で働きたいと思うなら、いまだに何らかの資格が求められます。たとえば、大学で教職に就きたいなら博士号が必要です。

— 社会人向け講座であるMIT Professional Educationにも関わっていますね。どのような人がどのような目的で受講しているのでしょうか。

それほど深くはありませんが、しばらく前から関わっています。カーネギーメロン大学で助教授をしていた当時も、5～10年程度の経験がある社会人向けにソフトウェアエンジニアリングの修士課程を開設する手伝いをし

注2) New Yorker誌などに寄稿するジャーナリスト。

# Software Designer #44 Daniel Jackson

MIT、コンピュータサイエンス学部 教授

ました。当時からわかっていたことですが、学部生と比較すると、社会人の大学院生は忍耐強く物事に当たります。近道をしようとして手痛い目に遭うことが多かったからでしょう。経験の少ない開発者ほど、ものを作るときのワクワクする気持ちを楽にします。何かを作るという行為はすごくおもしろいですからね。でも、そのワクワク感だけですべてを見てしまいがちになります。なので、学部生に、より大きな視野で物事をとらえ、設計をきちんと行うこと、つまり最初に単に開発するだけでなく、適切な構造を用意して適切な機能をサポートしていくことが重要であると理解してもらうのがとても難しいです。ソフトウェア業界での経験があれば、容易にこの考え方を受け入れられます。経験者はより長期的な視点で物事を考えるからです。

## → 「コンセプトの一貫性」を確保すること

「きちんと作られたソフトウェア」とはいまいな概念ですが、これをどう定義しますか？ 学生にはどのように説明しているのでしょうか。

ソフトウェアを設計する際に最も重要なのは、基本的なコンセプトを確定することだと思います。つまり、重要なコンセプトとその相互関係を理解することです。これらは設計を行う前に明確に示しておく必要があります。Fred Brooksは、1975年

刊行の著書『人月の神話』<sup>注3</sup>でこれを「コンセプトの一貫性」と呼んでいました。この本では、コーディングを行う前にソフトウェアアーキテクチャを構築することが重要であると強く主張していました。20年後にこの本を改訂したとき、Brooksはその重要性を一層強く確信したと言っていました。私も講義ではこのことを強調しています。

たとえば、私が教える学生はプロジェクト課題としてEコマースのショッピングカートを設定しています。表面的には、この作業はたいしたものではないように見えます。しかし、細部を詰めていくとすぐにコンセプトを明確にするにはそれなりの検討が必要だとわかります。客が購入する商品(シリアル番号を持つ特定のiPhone)と販売中の商品(「iPhone 5」)を区別しなくてはなりません。Amazonでは、「ほしい物リスト」と「ショッピングカート」が分かれています。この2つは同じものなのでしょうか。一見、どちらも単純な概念に見えますが、詳細に検討してみると実際にはよく考えられたものであるとわかります。詳細な検討を行わずにコードを書くと、乱雑になって後でさまざまな手直しが必要になります。コストも多くかかるでしょう。手直しが行われずに不細工で信頼

性の低いシステムにしかならないこともあります。

## → 新しいオンライン教育プログラム、edX

ー オンライン教育は、深い理解を必要とする学生にどのように対応できるのでしょうか。MITはOpen CourseWare<sup>注4</sup>でオンライン教育を主導しています。

OpenCourseWareは、MITでの講義に使用されていた教材をオンラインで公開したものです。私が担当する大学院での講義もその多くを公開しており、十分な成果を得ています。私が以前担当したソフトウェアエンジニアリングの講座は、実に幅広く利用されてきました。ポルトガル語やスペイン語に翻訳され、南米で盛んに活用されました。OpenCourseWareでは、ビデオや講義資料、スライド、問題集などをダウンロードできます。受講生が講座の担当者に質問することも可能です。私もOpen CourseWareの受講生と何度もすばらしい交流を持ってきました。

しかし、受講生から見れば、コミュニティに参加するわけではありません。現在変わりつつあるのは、オンライン教育がリアルタイムで提供され始めていることです。何人かの受講生がグループとなって講座と一緒に受講します。教材もそのような受講形式に合わせて作られています。たとえば、ビデオは好きなときに観ることができますが、講

注3) 滝沢徹、牧野祐子、富澤昇訳、ピアソン桐原、2010年(新組新装版)、ISBN978-4-8640-1005-4

注4) URL <http://ocw.mit.edu/>

## コンピュータサイエンス2.0 [Part 3] : 大学教育とオンライン教育

義は特定の時間帯に行われ、一緒に受講する仲間がいて、受講生同士やティーチングアシスタントとオンラインフォーラムでやりとりすることができます。

edX<sup>注5</sup>でこのような動きが見られます。edXはMITとハーバード大学が創設した新しいプログラムで、その後カリフォルニア大学バークレー校やテキサス大学が加わりました。edXとその役割については私も同僚とよく議論しています。

— edXは従来の教育とオンライン教育の橋渡しをしようとしているのでしょうか。

そう思います。edXの目標は、極めて質の高い教育を受ける機会を、学費や何らかの事情で大学教育を受けられない人に提供することです。edXの講座は現在は無料です。edXは、MIT内で受けられる教育をオンラインでより多くの受講生に提供でき

るように設計されています。つまり、これまではMITに入学しなければ受けられなかった専門教育を、世界中の何百万人もが活用できるようになります。edXによりMITの学生にも補助教材とオンラインでのサポートが提供され、MIT内での教育の質も改善されるでしょう。

— 修了証書も取得できるのですね。

はい。MITx、バークレーXなど、講座を提供する「X大学」が修了証書を交付します。当初は無償で提供しますが、将来は少額の交付手数料を課します。

### → 次世代の開発者へアドバイス

— では、そういう新しい環境で学び始める若い開発者に何かアドバイスをお願いします。

受講生には、ソフトウェアエンジニアリングの基本を忘れないでほしいと言いたいです。何よりもまず、問題の分析ができなくてはなりません。それに

よって、解決しようとしている問題について深い理解が得られます。第2に、設計思想を明確にして表現し、評価できなくてはなりません。そして第3に、アイデアをうまく実装できなくてはなりません。ときにこの3つを交互に行わなければなりませんが、これらをすべてこなして初めて熟練したソフトウェア開発者とみなされるのです。

次世代の開発者は、どのような形で教育を受けるか、さらに選択肢が増えるでしょう。したがって、それらを見分ける必要があります。質と量が十分で、そこから着想を得られるような教材を見つけ出し、努力して学習しなくてはなりません。また、テクノロジーに振り回されないください。技術的なプラットフォームを詳しく研究する人生も悪くありませんが、突き詰めれば大切なのは基礎となるコンセプト、アイデア、スキルであり、そういうものが今後も受け継がれていくのです。SD

注5) URL <https://www.edx.org/>

### ▼MIT、ハーバード大学などが参加するedX

The Future of Online Education  
for anyone, anywhere, anytime

EXPLORE FREE COURSES FROM edX UNIVERSITIES

MIT Massachusetts Institute of Technology

HARVARD UNIVERSITY

Berkeley UNIVERSITY OF CALIFORNIA

THE UNIVERSITY of TEXAS SYSTEM

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、  
総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、  
総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、  
総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、  
総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には記事が掲載されていました。編集の都合上、  
総集編では収録致しません。

## Report

日本ライフレイ、  
「第1回 Liferayオープンソース ユーザコミュニティ」開催

11月14日、東京六本木において日本ライフレイ(株)の主催によるセミナーイベント「第1回 Liferayオープンソース ユーザコミュニティ」が開催された。

Liferay社のコミュニティ・マネージャであるジェームズ・ファルクナー氏、プラットフォームの責任者、グレッグ・アマーソン氏、そして非営利教育団体「Sesame Workshop」のCTO兼CIO、ノア・ブロードウォーター氏を招いてLiferayの現状について発表が行われた。

最初に登壇したのは、Liferay社のコミュニティ・マネージャであるジェームズ・ファルクナー氏。Liferayの概要を説明した。

Liferayは、Webシステムを短時間で構築できるオープンソースのプラットフォームで、世界規模で活発にコミュニティ活動が行われている。

ファルクナー氏は、オープンソースコミュニティの目的は「製品に対するイノベーションを社外の方々からもたらしていただくこと」「大勢の外部の人たちの時間とエネルギーを持ってコントリビューションに寄与していただくこと」であると言う。

現在のLiferayの登録ユーザは世界で7万人以上おり、日々ログインしてコントリビュートしているユーザは約5千人、1日あたり150にのぼるコントリビューションがある。ライセンスはLGPLを採用しており、無償で提供されるLiferayをダウンロードしたユーザは、自身で変更した個別の製品をLGPLの規則に則って再販できるようにしている。ここ2～3年はとくに人気が高まっており、ダウンロード数が増加しているとグラフを示した。

ファルクナー氏はLiferayのWebサイトを示し、共有できるダッシュボードや、いくつものコミュニティでリアルタイムにどこでどんなアクティビティがあるかチェックできることを紹介した。アナウンスの内容やブログの最新情報を表示することも可能。また、コミュニティのメンバーと社員の努力によって多言語の翻訳版も作成しており、日本語のバージョンも用意されている。まだ日本語の表記などの課題はあるが、引き続き改善していくと述べた。

Liferayによって、スケーラブルなコラボレーション、コミュニティメンバー間が協業できるような場が提供される。コミュニティサイトはポータルだけでなく、アカウントの管理やユーザグループの管理もでき

る。コミュニティで共有管理するオブジェクトが、メンバーにより修正された場合には、クラスタ内のすべてのサーバへブロードキャストメッセージが送出されるため、コミュニティ内での情報や経験を共有し、継続的な改善につなげられるという。

さらにLiferayは単なる利益追求主義ではなく人のために役に立つ事業を追及する企業であり、毎年利益の10%を慈善事業に寄付し、災害復旧などに役立てていることも大きな特徴。日本のユーザ比率は現在1%だが、どんどんメンバーに加わっていただき、もっと大きくしていきたいと力を込めた。

フォーラムやコミュニティにはさまざまな側面があり、欧米を中心に活発に活動している。とくに国単位、地域単位でユーザグループが存在することも特徴で、それぞれ重要な機能を担当している。たとえば「BugSquad」は人気のあるグループの1つで、これにより多くのバグが修正されたとのこと。日本にはまだ日本語による独自のユーザグループは存在していないが、ぜひとも日本独自のユーザグループができ、Liferayの重要な一端を担って欲しい。今回の来日もそのための取り組みの一環であると述べた。

その後、グレッグ・アマーソン氏による「顧客サービスの価値を高める企業ポータル最新版の紹介」と題した技術解説セッションを行い、続いて実際にLiferayを活用したユーザ事例として、ノア・ブロードウォーター氏が登壇し「Building a More Agile Web Experience」と題したセッションを行った。

最後に、日本ライフレイ(株)の竹生泰之氏により「Liferay Community Contributor of the Year 2012」の表彰も行われた。この賞は世界中のLiferayユーザの中からとくに寄与したユーザを表彰するもの。3名が表彰されたがそのうちの1人として、日本人の小沢仁氏が表彰された。



▲Liferayコミュニティ・マネージャ  
ジェームズ・ファルクナー氏



▲Liferayプラットフォームの責任者  
グレッグ・アマーソン氏



▲Sesame WorkshopのCTO兼  
CIO ノア・ブロードウォーター氏

CONTACT

日本ライフレイ(株)  
URL <http://www.liferay.com>

## Service

## 日本マイクロソフト、 コラボレーション製品群をパートナークラウドで提供開始／ 「Windows 8 アプリ検証ラボ」を設立

パートナー企業5社が「Partner Hosted Productivity Cloud」を順次提供

日本マイクロソフト(株)は、エヌ・ティ・ティ・コミュニケーションズ(株)、日本電気(株)、日本ヒューレット・パッカード(株)、(株)日立システムズ、富士通(株)の5社と「Partner Hosted Productivity Cloud」(以下、PHPC)の契約を締結し、2012年11月末より順次、マイクロソフトのメール&コラボレーション製品群のクラウドサービスを、各社の国内データセンターからユーザに提供開始する、と発表した。

PHPCは「Microsoft Windows Server」「Microsoft System Center」「Microsoft Exchange Server」「Microsoft SharePoint Server」「Microsoft Lync Server」を中心としたマイクロソフト製品をパートナー企業が各社のデータセンターで運用し、ユーザにプライベートクラウドサービスとして提供するもの。

日本マイクロソフトは、すでに同社のデータセンターからメールおよびコラボレーション機能をパブリッククラウドサービスとして提供する「Microsoft Office 365」を提供しているが、Office 365に加えて多様化するユーザのSLA (Service Level Agreement) や利用形態へのニーズに対応するため、PHPC提供各社とともに、各社のシステム構築／運用／サポートにおける強み／経験を活かしたクラウドサービスを新たに提供する。ユーザは初期投資を抑え、すぐ利用できるなどのクラウドならではのメリットを享受しながらも、ユーザ独自の経営戦略やIT環境に合った柔軟かつ信頼性の高いサービスを利用することが可能となる。

Windows 8 タブレットの企業導入を支援する  
「Windows 8 アプリ検証ラボ」を設立

日本マイクロソフト(株)は、Windows 8 タブレットの法人向け導入を支援する「Windows 8 アプリ検証ラボ」を、同社の大手町テクノロジーセンター内に設置する。

同ラボには、デバイスパートナー各社が提供する最新の法人向けWindows 8 タブレットを設置する。各デバイスでアプリケーションの動作確認、パフォーマンス検証をはじめ、クラウドサービスなどのマイクロソフトのソリューションとの連携も含めて検証を行える。ラボの概要は次のとおり。

対象：Windows 8 タブレット向けの業務アプリケーションを開発するシステムインテグレータおよび、アプリ開発パートナー

設置期間：2012年11月29日～2013年3月末(2013年4月以降の展開については、ラボの検証状況を見ながら、検討予定)

検証環境：

- デバイスパートナー各社より提供された法人向け Windows 8 Pro 搭載モデルおよび Windows RT 搭載タブレットデバイス
- オンプレミスサーバ環境 (Active Directory、SharePoint、Lync など)
- インターネット接続環境

## CONTACT

日本マイクロソフト(株)  
URL <http://www.microsoft.com/japan>

## Service

## グレースシティ、 帳票開発ツールの新バージョン「ActiveReports for .NET 7.0J」を発売

グレースシティ(株)は帳票開発ツール「ActiveReports for .NET」の次期バージョン「ActiveReports for .NET 7.0J」を2013年1月7日に発売する。

新バージョンではハイブリッドデザイン方式を採用した。これによりセクションレポートとページレポートという2つのデザイン方式を1つのプロジェクト内で併用できる。セクションレポートは従来から提供してきた方式で、セクション単位で外観を設計し、詳細セクションでレコードを繰り返し表示させるため、階層化されたデータを効率よく表示できる。

一方、7.0Jで新たに追加したページレポートは印刷

結果とまったく同じレイアウトで帳票デザインを行えるため、複雑なレイアウトの帳票を短時間で作成可能。

どちらの方式もこれまでどおりデザイナーはVisual Studio上で起動し、操作性はほぼ同じ。

同製品の1開発ライセンス価格(税込)はProfessionalが29万4,000円、機能を絞ったStandardが16万8,000円。Webアプリケーションを開発して配布する際には別途コアサーバライセンスが必要。

## CONTACT

グレースシティ(株)  
URL <http://www.grapacity.com>

## Service

## さくらインターネット、 「さくらの専用サーバ」を改定し、月額8,800円から提供

さくらインターネット(株)は、専用サーバサービス「さくらの専用サーバ」の改定を12月6日に行った。

同サービスは申し込みから利用開始まで最速10分で、台数上限なく複数台構成が可能な専用サーバサービス。今回、「エクスプレスシリーズ」のサーバモデルの改定および「エクスプレスG2シリーズ」のサーバモデルの拡充を実施し、さらなる低価格化／高性能化を図った。

エクスプレスシリーズでは、新たにNECの「Express

5800 E120d-M」を採用。4コアのCPUと16GBのメモリ、1TBの耐障害性の高いミラーリング構成のHDDがついて月額8,800円という低価格を実現した。エクスプレスG2シリーズでは、従来の富士通「PRIMERGY RX100 S7」に加え、新たに「PRIMERGY RX200 S7」を追加。全モデル初期費用なしで利用できる。

**CONTACT** さくらインターネット(株)  
URL <http://www.sakura.ad.jp>

### ▼エクスプレスシリーズ新ラインナップ

サーバ	モデル	CPU	メモリ	標準ストレージ	料金 (税込)	
					初期費用	月額料金
NEC Express5800 E120d-M	SATA 1TB	Xeon E5-2407 2.20GHz 4Core	標準 16GB  最大 32GB	HDD SATA 1TB × 2 RAID1	79,800 円	8,800 円
	SAS 600GB			HDD SAS 600GB × 2 RAID1	84,800 円	
	SSD Intel520 240GB			SSD Intel520 240GB × 2 RAID1	99,800 円	
	SATA 1TB	Xeon E5-2430 2.20GHz 6Core		HDD SATA 1TB × 2 RAID1	99,800 円	10,800 円
	SAS 600GB			HDD SAS 600GB × 2 RAID1	104,800 円	
	SSD Intel520 240GB			SSD Intel520 240GB × 2 RAID1	119,800 円	
	SATA 1TB	Xeon E5-2430 2.20GHz 6Core 2CPU	標準 16GB  最大 64GB	HDD SATA 1TB × 2 RAID1	149,800 円	12,800 円
	SAS 600GB			HDD SAS 600GB × 2 RAID1	154,800 円	
	SSD Intel520 240GB			SSD Intel520 240GB × 2 RAID1	169,800 円	

### ▼エクスプレスG2シリーズ追加ラインナップ

サーバ	モデル	CPU	メモリ	標準ストレージ	料金 (税込)	
					初期費用	月額料金
Fujitsu PRIMERGY RX200 S7	SATA 1TB	Xeon E5-2670 2.60GHz 8Core	標準 16GB 最大 96GB	HDD SATA 1TB × 2 RAID1	0 円	27,800 円
	SAS 600GB			HDD SAS 600GB × 2 RAID1		28,800 円
	SSD Intel520 240GB			SSD Intel520 240GB × 2 RAID1		30,800 円
	SATA 1TB + Fusion-io ioDrive2 365GB			HDD SATA 1TB × 2 (RAID1) + Fusion-io ioDrive2 365GB × 1		57,800 円
	SATA 1TB	Xeon E5-2670 2.60GHz 8Core 2CPU	標準 16GB 最大 192GB	HDD SATA 1TB × 2 RAID1		37,800 円
	SAS 600GB			HDD SAS 600GB × 2 RAID1		38,800 円
	SSD Intel520 240GB			SSD Intel520 240GB × 2 RAID1		40,800 円
	SATA 1TB + Fusion-io ioDrive2 365GB			HDD SATA 1TB × 2 (RAID1) + Fusion-io ioDrive2 365GB × 1		67,800 円

## Hardware

## 11ac技術搭載 Wi-Fi 親機とUSB2.0 接続子機の セットモデルを新発売

(株)バッファローは最大600Mbps(理論値)の高速性能をほこる11ac技術を搭載したWi-Fi親機「WZR-D1100H」にUSB2.0接続子機「Wi-U2-400D」を同梱したセットモデル「WZR-D1100H/U」を発売した。

11ac技術とは、米国電気電子学会(IEEE)にて策定が進められている次世代無線LAN規格「IEEE 802.11ac」にて採用される見込みの高速化のための新技術。本製品には複数ある11ac技術の1つ「変調信号の多値化」を搭載した。この親機にUSB接続タイプの子機を同梱したことでPCを選ばずに11ac技術を利用できる。

通信速度は実測値で239Mbps。従来の11n/a製品に比べ約1.8倍の速度を実現した。200Mbpsクラスの光ファイバーインターネット回線やNASとの通信に威力を発揮するという。

11n/a(5GHz)と11n/g(2.4GHz)にも対応しているため、従来のWi-Fi機器も利用できる。価格は2万5,410円(税込)。



▲ WZR-D1100H/U

**CONTACT** (株)バッファロー  
URL <http://buffalo.jp>

## Report

Amazon Web Services、  
「re:Invent」で8つの企業が新インターネットビジネスを発表

米Amazon Web Services社（以下、AWS）は11月27日～29日の3日間、米国ラスベガスにおいて同社初のグローバル規模のユーザカンファレンス「AWS re:Invent 2012」を開催した。

同イベントでは実践ワークショップ、幹部向けの技術戦略セッション、アーキテクチャから運用まであらゆるテーマを扱う詳細な技術コンテンツなど、100を超えるセッションが行われた。

イベント2日目の28日には、選ばれた複数のスタートアップ企業が事業やアプリケーションを初めて公に発表するライブステージ「LAUNCHED!」が開催された。ステージでは8社のスタートアップ企業が、ビッグデータ、メディア共有、モバイルアプリケーション、クラウドソーシングなどに関連する新たなソリューションやアプリケーションを発表した。以下にそのうちの2つのサービス／アプリケーションを紹介する。

## Facebookのデータからビデオを作成するサービス

オンラインビデオ制作サービス会社Animotoは新サービス「Best of 2012 Facebook Videos」を

発表した。これはボタンをクリックするだけで、Facebook上のベストな写真やコメントを自動的に選択し、それらに曲を付けてビデオを作成するサービス。作成されたビデオは、多数の方法でシェアしたり、写真やコメントを追加／削除したり、音楽の変更などの編集をしたりできる。

## デバイス上の写真／ビデオの自動収集サービス

ThisLifeはスマートフォン、ハードディスク、SDカード、ソーシャルデバイス上に散在するすべての写真およびビデオを集めるアプリケーション。自動的に最も高画質な写真を保管し、重複したものを取り除いてくれるという。顔を特定する技術により、ユーザが写真のタグ付けするのを促進する機能もあるという。

そのほか、Alegion、Averail、BitYota、CloudMunch、Mortar Data、PunchTabといった企業が同様にAWSを使った新たなソリューションやサービスを発表した。

**CONTACT** Amazon Web Services  
URL <http://aws.amazon.com/jp>

## Hardware

PFU、  
ScanSnapシリーズの新製品「iX500」を発売

（株）PFUは、ワンタッチでスピーディーに書類をPDF化できるパーソナルドキュメントスキャナ「ScanSnap」シリーズの新製品「iX500」を11月30日から販売開始した。

新製品では、画像処理エンジン「GI」プロセッサと高速大量スキャンに適したブレーキローラーを新たに搭載した。これにより、従来製品よりも高速な読み取りが可能となり、1分間に25枚の読み取りを実現した。また、同製品単独でスマートフォンやタブレットにイメージデータを転送できるようになった。スキャンした原稿はWi-Fi接続により、直接スマートフォンやタブレットにPDFまたはJPEG形式で保存される。

これと同時にソフトウェアも強化した。クイックメニューの操作性を向上させたほか、スキャンとOCRの並列処理により検索可能なPDFを、通常のPDFとほとんど変わらない速度でファイルを生成できる。

ファイリングソフトウェア「楽2ライブラリ Smart V1.0 with Magic Desktop V1.0」を同梱したセットモデルを「ScanSnap iX500 Deluxe」として、同時に販売開始。また、現行モデル「S1300i」、「S1100」

についても同様にセットモデル「ScanSnap S1300i Deluxe」、「ScanSnap S1100 Deluxe」を販売開始した。

## ▼PFUダイレクトでの販売価格

品名	販売価格（税込）
ScanSnap iX500	4万9,800円
ScanSnap iX500 Deluxe	5万4,800円
ScanSnap S1300i Deluxe	3万2,800円
ScanSnap S1100 Deluxe	2万2,800円



◀ ScanSnap  
iX500

**CONTACT** 株式会社PFU  
URL <http://www.pfu.fujitsu.com>

## Topic

サイバーエージェント、  
スマートフォンサービス開発のためのオープンソースを公開

(株)サイバーエージェントは11月29日、Android向けの画像変換のフレームワーク「GPUImage for Android」と、Webサーバとの通信テストなどを簡単に実現するモックサーバ「easymock」をオープンソースとして公開した。

同社はスマートフォン向けサービスとしてコミュニティ&ゲームSNS「Ameba」を中心に、23個のコミュニティサービスや、66タイトルのソーシャルゲームを提供している。現在開発中のサービス数も88プロジェクトに及ぶなど、国内最大規模の体制でスマートフォンサービスの開発を行っている。

同社はより効率的なサービス開発を実現するために、GPUImage for Androidとeasymockを開発した。もともとは2012年8月27日に提供開始したスマートフォン向けオークションサービス「パシャオク」のために開発されたもので、今後も同社が開発提供するさまざまなスマートフォンサービスにて採用していくという。両ツールの特徴は右のとおり。

## ● GPUImage for Android

Android向けに開発した画像変換のフレームワーク。iOS向けの画像変換フレームワーク「GPUImage」とも互換性が高く、iOS、Androidの両方にサービスを提供する場合でも効率的に開発できるほか、独自のフィルター機能や反応スピードの速さなどが特徴。

## ● easymock

ネイティブアプリ開発において必要となるWebサーバとの通信を伴う処理のテストを、サーバサイドでの開発を待つことなく、開発途中の段階で随時行うことができるモックサーバ。多彩な表現のAPIを簡単にテストすることもでき、開発を効率的に行える。npmパッケージでもダウンロード可能。

## ■ ダウンロードサイト

<https://github.com/cyberagent-jp>

## CONTACT

(株)サイバーエージェント

URL <http://www.cyberagent.co.jp>

## Software

ネオジャパン、  
「desknet's NEO」をリリース

(株)ネオジャパンは、「desknet's」の後継となる新製品「desknet's NEO」の中堅／大規模環境向けパッケージ版を12月5日より発売開始した。

desknet'sは、2002年の発売開始から累計276万ユーザ以上（2012年9月現在）の販売実績をもつ純国産のWebグループウェア製品。その後継となるdesknet's NEOは、高い機能性をさらに強化し、スケジュール、メール、ワークフロー、文書管理といった従来機能の性能向上に加え、社内ソーシャルや事業継続ポータルといった新しい機能も搭載した。

## ■ desknet's NEOの特徴

- 25のアプリケーションを標準搭載。スケジュール、メール、文書管理などの一般的な機能に加え、安否確認や来訪者管理といった特徴的な機能も搭載
- UIにはHTML5を採用。タッチ操作を意識した画面デザインで、タブレットからもPCと同じ操作性でストレスなく利用可能。スマートフォン向けには専用インターフェースを用意。利用する機種を選ばず専用アプリのような使い勝手を実現

- ソーシャル要素を取り入れた新しい社内コミュニケーションツール「ネオツイ」、簡単カスタマイズで新しい情報共有の形を提案する「ログイン画面」など、誰もが毎日使いたくなるための工夫
- 多くの大規模導入実績を持つ「desknet's エンタープライズ版」の技術とノウハウを踏襲し、数万ユーザを超える環境でも、高い性能と信頼性を実現

中堅／大規模（300～数万ユーザ）向けの「desknet's NEO エンタープライズライセンス」の価格（税抜）は、300ユーザが150万円～、1,000ユーザが472万円～、1万ユーザを超える場合はユーザ数無制限で1,534万円～などとなっている。

小規模（5～300ユーザ）向けの「desknet's NEO スモールライセンス」とクラウドで利用する「desknet's NEO クラウド版」の価格発表および提供開始は、2013年2月の予定。

## CONTACT

株式会社ネオジャパン

URL <http://www.neo.co.jp>

## Topic

## NEC、インターネットやモバイル通信網の通信可能速度を高精度に予測する技術を開発

NECは11月26日、インターネットやモバイル通信網（3G/LTE）において、変動する通信可能速度（通信スループット）を、アプリケーションソフトウェアで高精度に予測する技術を開発したことを発表した。

インターネットやモバイル通信網などのベストエフォート型のネットワークは、他のユーザの通信状況や電波状況により通信スループットが時々刻々と変化する。このため、とくにクラウドサービスにおいては、通信の遅延や画像の劣化など、サービス品質に影響を与えてしまう。

このたび開発した技術は、ネットワーク上に新たな通信装置を追加せずに、アプリケーションソフトウェアだけで、3分後まで通信スループットを高精度に予測するもの。これにより、通信装置を持たないクラウドサービス事業者でも、予測値に応じて即座にデータ通信を制御することで、高品質なクラウドサービスの提供が可能になるという。

また、インターネット上のコンテンツ配信事業者は、本技術を利用することで、予測した通信スループットに応じて映像や音声の品質を制御し、途切れのない安

定した視聴を実現する映像配信サービスを提供できる。また、予測した通信スループットに合わせて画像サイズや表示コンテンツを調整することで、応答性の高いWebサービスを実現できる。

このたび開発された技術の特徴は、次のとおり。

- 現在のスループットの状態（定常／非定常）を判別し、その状態に応じて定常モデルと非定常モデルを混合し、現在から3分先までのスループットの予測モデル（混合モデル）を作成する
- 開発した混合モデルは、過去のデータ通信量の履歴だけで予測が可能であるため、ネットワーク上に新たな通信装置を追加することなく予測が可能
- 通信スループットの変動幅の確率分布を求めて予測に利用することで、従来の方式よりも予測精度を改善し、3分先までの通信スループットの変動を、約80%の高精度で予測できることを実証

## CONTACT

日本電気(株)

URL <http://jpn.nec.com>

## Service

## ハイパーボックス、Webシステム開発会社向けPHPの開発、本番環境「konekto PHP Cloud」を提供開始

パブリッククラウドサービス「HyperCloud」を運営する(株)ハイパーボックスは11月20日から「konekto PHP Cloud」の提供を開始した。

同サービスはWebシステムの開発環境と本番環境をCloudベースで提供するサービス。利用者は「konekto PHP Cloud」を購入すると指定のサービススタート日にインストール済みの開発環境と本番環境の利用を開始でき、これまで必要としていたハードウェアの調達やOSのインストールなどの環境構築が不要になる。開発期間と本番期間に合わせた2種類4製品をラインナップしている（右表）。

konekto PHP Cloud開発サーバには、PHPアプリケーションサーバ「Zend Server EE 日本語版」とPHP統合開発環境「Zend Studio 9 日本語版」が搭載されている。konekto PHP Cloud本番サーバには「Zend Server SE 日本語版」が搭載されている。

／メモリ:0.5GB／HDD:40GB／通信量:0.5Mbps)  
ソフトウェア:CentOS 5.6 64bit、Apache 2.2.2x、Zend Server EE 1ライセンス、Zend Studio 1ライセンス

### ● konekto PHP Cloud本番サーバ

サーバスペック：クラウドサーバ1台（vCPU：1個／メモリ：2GB／HDD：40GB／通信量：0.5Mbps）  
ソフトウェア:CentOS 5.6 64bit、Apache 2.2.2x、Zend Server SE 1ライセンス

## ▼価格構成

サービス名	価格（税込）
konekto PHP Cloud 開発サーバ 3ヵ月	5万7,750円
konekto PHP Cloud 開発サーバ 6ヵ月	11万250円
konekto PHP Cloud 本番サーバ 6ヵ月	11万250円
konekto PHP Cloud 本番サーバ 12ヵ月	20万7,900円

## ■製品構成

### ● konekto PHP Cloud開発サーバ

サーバスペック：クラウドサーバ1台（vCPU：1個

## CONTACT

(株)ハイパーボックス

URL <http://www.hyperbox.co.jp>

# Letters from Readers

## ネット党首討論会で投票率は上がる？

11月29日のニコニコ動画でのネット党首討論会は140万人が視聴したそうです。大型モニターをバックに、自説を述べる党首たちはまだまだごちない感じがしましたね。ですが、今までにないほど若者の注目を浴びたのは確かでしょう。本誌発売日にはもう選挙は終わっていますが、投票率は上がったのでしょうか？ こういうことが当たり前になると、もっとネット受けのする議員も現れてきそうです。



## 2012年11月号について、たくさんのお便りありがとうございました！

### 第1特集 もし、OpenFlowでやれと言われたら？

ソフトウェア的にネットワークを定義し、制御するSoftware Defined Networkがネットワーク業界では話題です。それを実現するための手法の1つがOpenFlowです。夢のような技術として業界をにぎわしているものの、具体的に何ができるのか、誰にメリットがあるのか、わかりにくい状況にありました。そこで本特集では「OpenFlowでは何ができるのか」に焦点をあて、データセンター、ネットワークベンダ、Slerなどの立場から、OpenFlowの可能性について語ってもらいました。

OpenFlowという名前だけは聞いたことがあったのですが、具体的な内容はわかっていなかったの、興味深く読みました。SDNは非常におもしろそうな技術ですので追っていききたいと思います。

埼玉県／鯖管理人さん

知識としては参考になった。実践する機会があるか……。

宮城県／clispさん

OpenFlowについて知りたいと思っていたので、ちょうどよかった。

福岡県／波多江さん

まさに情報収集中。

東京都／佐藤さん



OpenFlowで恩恵をうけるのはおもにデータセンターです。一般企業が活用することはあまりないでしょう。しかし、データセンター側の技術を知っておくのは、クラウドサービスの効果的な利用方法やリスクを考えるうえで、多少なりとも役に立つのではないのでしょうか。

### 第2特集 Muninが手放せない理由

サーバ上のシステムリソースやネットワークを監視するためのOSS「Munin」を紹介しました。統合監視ツールとは違い、リソースの監視に特化することでシンプルに使い、さまざまなデータを視覚的に表現できます。それを理解していただくために、グラフなどの図版を豊富に交えて解説しました。

実際の使われ方が想像しやすかった。

東京都／HIさん

ちょうど会社のサーバにMuninを導入したところでしたので、参考になりました。

茨城県／haruoさん

Muninは私も利用していますが、一度使い出すと病みつきになります。

大阪府／しろたんさん



すでにMuninを導入しているという読者も多かったです。導入も簡単ですし、何よりも情報をグラフなどで確認できるのがわかりやすいのでしょう。

### 一般記事 SkeedSilver Bulletが求められている理由

Winnyの開発者、金子勇氏が創業した(株)Skeedが開発した高速ファイル転送ソフトウェア「SkeedSilverBullet」を取り上げました。遠距離でも高速に転送するためのしくみについて解説しました。

ニュースを見たときからしくみを想像していたので、そこが垣間見れたのは良かった。

滋賀県／東川さん

P2P技術をベースにした興味ある技術です。

東京都／みんのさん



現在のファイル転送技術の課題や求められているニーズなどから解説されていたので、開発の経緯などがわかったのも興味深かったと思います。

### 一般記事 JSX入門【後編】

(株)DeNAで開発されたプログラミング言語「JSX」についての記事の後編です。

今回は、最適化、開発環境などについて紹介しました。

JavaScriptの改良言語ということで興味があった。おもしろいと思う。

北海道／村橋さん



前回の言語仕様に関する内容とは違い、今回は使いやすさの面をお伝えしました。JSXによる開発のイメージが伝わったなら幸いです。

### 一般記事 速報 Ubuntu 12.10の新機能と変更点

本誌11月号の発売日と同日の10月18日にUbuntu 12.10がリリースされました。その新機能や変更点について最新情報をお届けしました。

遅れてついていく者には「新機能と変更点」のような情報は貴重で大いに役立ちます。

富山県／Qkobさん

Ubuntuはデスクトップとして試してみたいOSです。

千葉県／Tayuさん



読者アンケートにはCD-ROMに収まらなくなったことに関するコメントがいくつかありました。地味な変更点なのに、注目度が大きかったのは意外でした。

### 連載

digital gadgetのSIGGRAPHの記事は研究段階から市販されているものまで

いろいろあって興味が湧きました。

大阪府／澤下さん



映画で描かれていた近未来の技術が、現実のものになりつつあるというのは驚きでした。担当は家庭用三次元プリンタの記事も気になりました。

### フリートーク

職場のエアコンが悪いのか、暑かったり寒かったり体調が変。

愛知県／匿名希望さん



気候のせいもあるのではないのでしょうか？ 比較的暖かい日や極度に寒い日などが入れ替わりやってくるので、編集部でも空調管理はやりにくかったです。体調にはお気をつけください。

## エンジニアの能率を高める一品

仕事の能率を高める道具は、ソフトウェアやスマートフォンのようなデジタルなものだけではなく、このコーナーではエンジニアの能率アップ心をくすぐるアナログな文具、雑貨、小物を紹介していきます。

### USB 紙コップウォーマー

1,980円(税込)／サンコー <http://www.thanko.jp>



※製品に紙コップは付属しません

今回はホットドリンクを温かく保つ、カップウォーマーというものを取り上げます。カップ底面をヒーターで温めることで温度を保つという製品です。類似製品はどれも陶器のマグカップには効果が薄いようでしたので、紙コップに特化した本製品を試しました。陶器より薄い紙コップなら、もっとヒーターの熱が伝わるのではないか、との考えからです。また他社製品よりもヒーター部の温度が90℃と高いのも期待が持てました。実際に使ってみると、温かさを保つというより、少し冷めにくくなるという感じです(通常15～20分ほどでぬるくなる場所、本製品を使うと30分くらいまで遅らせられる程度)。缶飲料にも使えるとのことですので、缶コーヒーでも試しました。すると、アツアツとはいきませんが、温かいと感じる程度の温度はキープできました。ずっとほんのり温かい状態を保っていましたので、使うなら缶飲料で使うことをお勧めします。(読者プレゼントあります。16ページ参照)



▲缶飲料で利用してみた

祝

### 11月号のプレゼント当選者は、次の皆さまです

- ①モバイルバッテリー MiLi Power Rover ..... 北海道 鶴飼嘉之様
- ②Bluetooth 4.0対応 BlueLED マウス ..... 東京都 角田学様

★その他のプレゼントの当選者の発表は、発送をもって代えさせていただきます。ご了承ください。

# 次号予告

# Software Design

February 2013

## 2013年2月号

定価1,280円 176ページ

1月18日  
発売

[第1特集] cron、fork、UNIXコマンドを復習し、高度なスクリプティングへ

## シェルスクリプティング道場

[第2特集] 森川滋之劇場・忙しいITエンジニアのための

## 超効率勉強法

[特別企画] いよいよスタンダードに!

## FreeBSDの新しい使い方

### お詫びと訂正

以下の記事に誤りがございました。読者のみなさま、および関係者の方々にご迷惑をおかけしたことをお詫び申し上げます。

#### ■2012年12月号

●連載「IPv6化の道も一歩から」P.128 著者の所属

[誤]: IPv6協議会 IPv4/IPv6共存WG アプリケーションのIPv6対応検討SWG

[正]: IPv6普及・高度化推進協議会 IPv4/IPv6共存WG アプリケーションのIPv6対応検討SWG

### 休載のお知らせ

「システムに必要なことはすべてUNIXから学んだ」(第8回)は都合によりお休みいたします。

### SD Staff Room

●この号の付録は、靈験あらたかな電電宮お守りシールです。以前から一度やってみたかったことを実現しました。京都の嵐山にあるので初詣に行くのもたいへんです。本誌の筆者さんがシステムサービスインの前には必ず参詣するという話を聞いていたのがきっかけです。ちょうど特集もバックアップ。(本)

●11月10日に編集部が引越。2日前からノロっぽい症状でダウン。発熱とトイレ出られない状況で1日休みましたが前日は荷造りのためになんとか出社。ふらふらで、段ボール30箱以上梱包。帰宅して昏倒でした。体重も3kg減(既に元通りに漸近w)。近くのビルなので住所や電話番号はそのままです。(幕)

●子どもたちが習う空手の昇級試験と試合を見学した。言うは易く、行うは難し。必死に戦っている彼らに、空手などやったこともないヒョロヒョロの父親が何を言ってあげられるというのか。よくがんばった—やられて戻ってきた彼らにようやく言えた言葉。思い出すたびもどかしくて鼻がつんとする。(キ)

●誌面では紹介しきれませんでした。法輪寺には、電気／電波先駆者の霊を讃えるための電電塔やエジソンのブロンズ胸像などがあります。付録のヒントになった電電宮オリジナルの情報安全護符シールも販売されています。嵐山は桜や紅葉の名所でもあるので、ぜひ一度足を運んでほしいものです。(よし)

●システムバックアップ特集、個人ユース的にも参考になることがいっぱい勉強になりました。自宅の外付けディスクは何回かボタンボタンと倒したとか、HDDビデオレコーダーはもういつお亡くなりになってもおかしくない年数だなどか思い当たることが多数……。色々な備えを年末年始にやりたいです。(ほ)

●諏訪にある酒蔵に行ってきました。お酒は強くないのに、新酒祭りも偶然開催された日だったので、ついつい色々な銘柄を試飲。日本酒以外にも梅酒や焼酎もあり、まるめろ酒を飲んでみたかったのですが、残念ながら品切れ中。かわりに日本酒数本と梅酒を購入しましたが、いつ飲み終わるかな?(まつ)

### ご案内

編集部へのニュースリリース、各種案内などがございましたら、下記宛までお送りくださいますようお願いいたします。

Software Design 編集部  
ニュース担当係

[FAX]  
03-3513-6173

※ FAX 番号は変更される可能性もありますので、ご確認のうえご利用ください。

[E-mail]  
sd@gihyo.co.jp

Software Design  
2013年1月号

発行日  
2013年1月18日

●発行人  
片岡 巖

●編集人  
池本公平

●編集  
金田富士男  
菊池 猛  
吉岡高弘

\*  
細谷謙吾(書籍編集長)  
取口敏憲

●編集アシスタント  
松本涼子

●編集協力  
坂井直美  
金子卓也(トップスタジオ)

●広告  
中島亮太  
北川香織

●発行所  
㈱技術評論社  
編集部  
TEL: 03-3513-6170  
販売促進部  
TEL: 03-3513-6150  
広告企画部  
TEL: 03-3513-6165

●印刷  
図書印刷(株)

# Software Design

この個所は、雑誌発行時には広告が掲載されていました。編集の都合上、  
総集編では収録致しません。

# Software Design

この個所は、雑誌発行時には広告が掲載されていました。編集の都合上、  
総集編では収録致しません。