青い空の下でコードを書きたいね！

USP MAGAZINE

for the sophisticated shell scripters
日本で唯一のシェルスクリプト総合誌

よりぬき版

2014 5 May Vol.13 ¥500

新連載
アジャイル改善塾
～組織文化を変える汗と涙の物語～

TechLION 再録
江崎浩
インターネット屋の次の仕事

ユニケージ開発手法 コードレビュー

新連載
縁の木、育てよう

りゅうちの工作員日記

Artwork created by Koji Ishizuka. © 2012 Zie's Bank .Co,. LTD All rights reserved.

BSDコンサルティング株式会社　取締役　最高技術責任者　後藤大地

第十回　UFS ファイルシステムの特性を知る ～小さいファイルとディスクスペース～

ファイルシステムの特性を知っていますか？ 1

これまでは単機能のコマンドをパイプで接続して並列処理することで、マルチコア / メニーコアでスケールする処理を記述できることを紹介するとともに、さまざまな角度から性能が発揮できているのか調査する方法や、動作中のシステムがどういった状況にあるのか調べる方法を説明してきた。今回は視点をファイルシステムへ移してみよう。

現在のマシンでもっとも性能上のボトルネックになるのはディスクストレージとのデータのやり取り部分にある。プロセッサやメモリと比較するとディスクストレージは通信速度が桁違いに遅い。可能な限り高速に読み書きするには、それに適したファイルシステムを選択する必要がある。ストレージ向けのファイルシステムとしては ZFS が今後の主流になりそうな趨勢にあるが、ZFS は性能の面では UFS にかなわない。高速に動作するファイルシステムが必要なときは UFS、または UFS と設計が近いファイルシステムが候補としてあがってくる。

コマンドとパイプでシステムを構築する場合、変数の代わりにファイルを使った方が便利なことが多いので、処理対象のデータを小さいサイズのファイルに分割して数千件や数億件といったファイルに分散配置させることがある。しかし、実はこういった使い方は UFS 的には容量的に効率の良い使い方とはいえない。今回はそのあたりを説明する。

ファイルシステムやストレージを調べる方法を知ろう 2

Solaris 系のオペレーティングシステムや Illumos 系のオペレーティングシステム、FreeBSD 系のオペレーティングシステムはデフォルトのファイルシステムに ZFS に採用する傾向がある。たとえば次のシステムは起動領域（ブート関連のデータが置かれたファイルシステム）も ZFS で構築されている。ここに

リスト 2.1 UNIX 系オペレーティングシステム ファイルシステムの利用例

```
% mount
zroot/bootenv/default on / (zfs, local, noatime, nfsv4acls)
devfs on /dev (devfs, local, multilabel)
procfs on /proc (procfs, local)
procfs on /compat/linux/proc (procfs, local)
/dev/gpt/s on /s (ufs, local, journaled soft-updates)
zroot/tmp on /tmp (zfs, local, noatime, nosuid, nfsv4acls)
zroot/usr/local on /usr/local (zfs, local, noatime, nfsv4acls)
zroot/var on /var (zfs, local, noatime, nfsv4acls)
zroot/var/crash on /var/crash (zfs, local, noatime, noexec, nosuid, nfsv4acls)
zroot/var/db on /var/db (zfs, local, noatime, noexec, nosuid, nfsv4acls)
zroot/var/db/pkg on /var/db/pkg (zfs, local, noatime, noexec, nosuid, nfsv4acls)
zroot/var/empty on /var/empty (zfs, local, noatime, noexec, nosuid, nfsv4acls)
zroot/var/log on /var/log (zfs, local, noatime, noexec, nosuid, nfsv4acls)
```

```
zroot/var/mail on /var/mail (zfs, local, noatime, noexec, nosuid, nfsv4acls)
zroot/var/run on /var/run (zfs, local, noatime, noexec, nosuid, nfsv4acls)
zroot/var/tmp on /var/tmp (zfs, local, noatime, nosuid, nfsv4acls)
z1 on /z (zfs, local, nfsv4acls)
z1/virtualbox on /z/localvm (zfs, local, nfsv4acls)
fsserver:/home/disk on /n/netdisk (nfs, read-only, nfsv4acls)
fsserver:/home/daichi on /n/daichi (nfs, nfsv4acls)
z1/daichi on /Users/daichi (zfs, NFS exported, local, nfsv4acls)
%
```

UFS や NFS、それにデバイスファイルシステムとプロセスファイルシステムなどがマウントされている。

ここでは UFS のファイルシステムを調べたいので、次のように mount(8) コマンドを実行して UFS のファイルシステムの情報だけを表示させてみる。

リスト 2.2 UFS でフォーマットされたファイルシステムを調べる

```
% mount -t ufs
/dev/gpt/s on /s (ufs, local, journaled soft-updates
)
%
```

/dev/gpt/s というデバイスが /s にマウントされていることがわかる。/dev/gpt/ に下に存在しているということは、GPT スキーマでパーティショニングされたディスクで、かつ、GEOM のラベル機能を使って s に名前が割り当てられてるのだろう、ということがわかる。glabel(8) でさらに次のように情報を調べる。

リスト 2.3 glabel(8) で /dev/gpt/s の実態を調べていく

```
% glabel status
                                      Name Status Co
mponents
                                gpt/gptboot0    N/A ad
a0p1
gptid/a971b10c-49e0-11e3-b324-e06995f54284    N/A ad
a0p1
gptid/a9756f07-49e0-11e3-b324-e06995f54284    N/A ad
a0p2
                                  gpt/swap0    N/A ad
a0p3
                                       gpt/s    N/A ad
a1p1
%
```

glabel(8) で /dev/gpt/s の実態を調べると ada1p1 になっていることがわかる。ada(4) デバイスドライバは ATA デバイスに対応したものだから、2 つ目の ATA 接続ディスクが本体のようだ、ということがわかる。ada1 をさらに次のように gpart(8) で調べてみる。

リスト 2.4 gpart(8) でディスクの情報を調査

```
% gpart show ada1
=>       34  488397101  ada1  GPT  (233G)
         34  482344960     1  freebsd-ufs  (230G)
  482344994    6052141        - free -  (2.9G)
%
```

ada1 は GPT スキーマでパーティショニングされたディスクで、ディスクサイズは 233GB ほど、作成されているパーティションは 1 つで、その部分は 230GB、フォーマットは FreeBSD UFS のようだ、ということがわかる。使用量も調べてみよう。次のように df(1) コマンドで確認する。

リスト 2.5 df(1) でファイルシステムの空き容量を確認

```
% /bin/df  /s
Filesystem 1K-blocks     Used     Avail Capacity  Mo
unted on
/dev/gpt/s 233584120 21761016 193136376    10%    /s
%
```

10% ほど使っており、90% ほどが空いていることがわかる。だいたい 184GB ほどの空き容量ということになる。

UFSの構造：ブロックとフラグメンテーション 3

今回は UFS のデータアクセスの単位である「ブロック」と、最小のデータ保持単位となる「フラグメンテーション」について説明する。UFS ではデータは「ブロック」と呼ばれる単位で扱われる。ブロック単位でデータの読み書きが実施される仕組みになっている。図 3.1 の長方形をブロックだと思ってほしい。

どんなに小さいファイルでも、必ず 1 つのブロックを消費する。たとえばブロックが 8KB なら、1KB しか使わないから、7KB は誰にも使われない、もっ

図 3.1 データのアクセスは「ブロック」単位で実施される

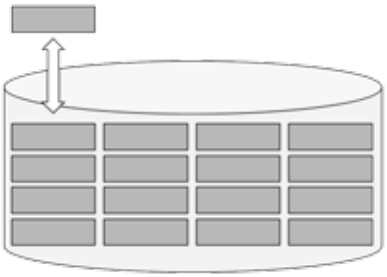
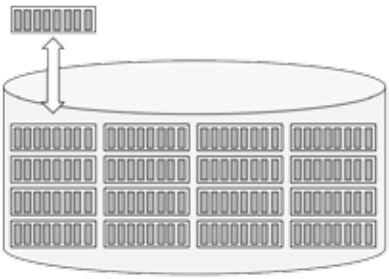


図 3.2「ブロック」を「フラグメンテーションの集まり」という仕組みに変更



たいない領域ということになる。この問題を解消するために UFS には「フラグメンテーション」という仕組みが導入されている。データアクセスの単位は「ブロック」のままだが、ブロックをさらにいくつかのまとまりに分割して利用するというものだ。この最小単位が「フラグメンテーション」と呼ばれている（図 3.2）。

データ保持の最小単位をフラグメンテーションまで落とし込むことで、もったいない領域の発生を防ぐ仕組みになっている。たとえばフラグメンテーションのサイズが 1KB であれば、1KB のファイルは 1 つのフラグメンテーションを使い切ることになり、ディスクスペースの面で使用率が高いということになる。この「ブロック」と「フラグメンテーション」という仕組みから、UFS で小さいファイルを大量に作成するときのひとつの指針が見えてくる。

まず、対象の UFS のブロックサイズとフラグメンテーションサイズを調べてみよう。UFS でフォーマットされたファイルシステムの詳細情報は dumpfs(8) または dumpe2fs(8) コマンドで調査できる。FreeBSD なら dumpfs(8)、Linux なら dumpe2fs(8) を使う。

各項目の詳しい意味についてはいずれ説明するとして、今回は「bsize」「fsize」「nbfree」に注目したい。オペレーティングシステムごとに違うラベル名になっているが、だいたい似たようなニュアンスになっているのでそのあたりは読み替えてほしい。それぞれ表 3.4 の意味を表している。

調べたファイルシステムはブロックサイズが 32KB、フラグメンテーションが 4KB ということになる。つまり、1 ブロックあたり 8 フラグメンテーション抱え

リスト 3.3 dumpfs(8) または dumpe2fs(8) で UFS ファイルシステムの詳細情報を表示

```
% dumpfs /s | head -18
magic   19540119 (UFS2) time    Sat Jan 18 03:49:48 2014
superblock location     65536   id      [ 527eca07 3775b6b3 ]
ncg     377     size    60293120        blocks  58396030
bsize   32768   shift   15      mask    0xffff8000
fsize   4096    shift   12      mask    0xfffff000
frag    8       shift   3       fsbtodb 3
minfree 8%      optim   time    symlinklen 120
maxbsize 32768  maxbpg  4096    maxcontig 4     contigsumsize 4
nbfree  6587953 ndir    91442   nifree  29211447        nffree  252152
bpg     20035   fpg     160280  ipg     80256   unrefs  0
nindir  4096    inopb   128     maxfilesize     2252349704110079
sbsize  4096    cgsize  32768   csaddr  5056    cssize  8192
sblkno  24      cblkno  32      iblkno  40      dblkno  5056
cgrotor 170     fmod    0       ronly   0       clean   0
metaspace 6408  avgfpdir 64     avgfilesize 16384
flags   soft-updates+journal
fsmnt   /s
volname         swuid   0       providersize    60293120
%
```

表 3.4 dumpfs（8）コマンドの出力：ブロックとフラグメンテーションに関する情報

項目	値	内容
bsize	32768 バイト	ブロックサイズ
fsize	4096 バイト	フラグメンテーションサイズ
nbfree	6587953 個	使われていないブロックの個数

ている計算になる。ブロックサイズとフラグメンテーションサイズはフォーマット時に指定できるようになっており、dumpfs(8) であれば次のように -m オプションを指定することでフォーマット時のオプションを表示させることができる。

リスト 3.5 フォーマット時に指定されたオプション

```
% dumpfs -m /s
# newfs command for /s (/dev/gpt/s)
newfs -O 2 -U -a 4 -b 32768 -d 32768 -e 4096 -f 4096
 -g 16384 -h 64 -i 8192 -j -k 6408 -m 8 -o time -s 4
82344960 /dev/gpt/s
```

「-b 32768」と「-f 4096」となっていることから、dumpfs(8) が表示したようにブロックサイズが 32KB で、フラグメンテーションサイズが 4KB であることがわかる。

df(1) で出力された空き容量と使われていないブロックの数はほぼ同じ値になる必要がある。使われていないブロックの数がそのまま空き容量ということになるからだ。

リスト 3.6 空き容量と使われていないブロック数の関係を確認

```
% bc -l
193136376 / 32 * 1.09
6578707.80750000000000000000
quit
%
```

df(1) は 10% ほど余裕をみた状態で報告するので、df(1) が表示した値を 10% ほどさらに盛ってみると、使われていないブロック数とほぼ値が一致する。

ブロックサイズを引き上げるほどデータの転送性能は引き上がることになり、ブロックサイズを引き下げるほどディスクスペースを効率的に使うようになる。デフォルトのブロックサイズとフラグメンテーションサイズは時代とともに引き上げられている。歴史的に見ればブロックサイズは 512 バイトだった。これが 1KB、2KB、4KB、8KB･･･　と引き上げられ、FreeBSD 10.0 では 32KB がデフォルトのブロックサイズになっている。

※Linuxのext2、ext3、ext4は基本的な設計は似ているもののブロックとフラグメンテーションの考え方についてはLinux独自のロジックに基づいており、ここでの話は適用できない。特にext4は新しい概念としてエクステントを導入しており、Linux ext4を使う場合にはext4の特性に沿った使い方をする必要がある。

実際の動作を知る 4

こうした UFS の構造上の仕組みを知った上で、ひとつ実験をしてみよう。0KB、1KB、2KB、4KB、8KB、16KB、32KB、64KB、128KB のファイルを 100 万個作成した場合、それぞれどれだけブロックを消費し、空き容量が減ることになるだろうか。単純にかけ算すれば表 4.1 の結果が想定される。

表 4.1 ブロックの消費と使用されるサイズの予測

1 ファイルのサイズ	作成するファイルの個数	予想される消費ブロック数	予想される消費サイズ
0KB	100 万個	-	-
1KB	100 万個	3 万 1,250 個	977MB
2KB	100 万個	6 万 2,500 個	1.95GB
4KB	100 万個	12 万 5,000 個	3.90GB
8KB	100 万個	25 万個	7.80GB
16KB	100 万個	50 万個	15.63GB
32KB	100 万個	100 万個	31.25GB
64KG	100 万個	200 万個	62.50GB
128KB	100 万個	400 万個	125.00GB

今回は詳しくは触れないが、UFS ではファイルのメタデータ（ファイルのパーミッション、作成日、サイズ、ブロックへのアドレスデータなど）を i-node という形式で保持している。つまり、1 つファイルを新しく作ってデータを書き込むという操作は、i-node を作成すること（正確にはディレクトリファイルへの書き込みも発生）とブロックにデータを書き込むことの 2 つの作業を実施していることになる。0KB のファイルを 100 万個作ることを想定しているのは、メタデータの作成だけでどの程度ブロックが使われるのかを調べるためだ。0KB のファイルを 100 万個作るなら次のようにコマンドを実行すればよい。

リスト 4.2 0KB のファイルを 100 万個作るコマンド

```
% mkdir /s/d
% cd /s/d/
% ls | wc -l
       0
% seq -f '%07.f' 1 1000000 | xargs -L1 -I% touch %
% ls | wc -l
 1000000
%
```

削除する方法も示しておく。

リスト 4.3 作成したファイルを削除する方法

```
% ls | wc -l
 1000000
% seq -f '%07.f' 1 1000000 | xargs rm
% ls | wc -l
       0
%
```

任意のサイズのファイルを作るなら次のように xargs(1) と dd(1) を組み合わせる。

リスト 4.4 任意のサイズのファイルを作成する方法

```
% seq -f '%07.f' 1 1000000 | xargs -L1 -I% dd if=/de
v/zero of=% bs=1024 count=1 2> /dev/null
 ls | wc -l
 1000000
% ls -l 1000000
-rw-r--r--  1 root  daichi  1024 Jan 18 14:06 100000
0
%
```

このようにしてファイルを作成して、実際にどういったデータが使われたのかを調べると表 4.5 のようになる。

表 4.5 dumpfs（8）および df（1）で表示される値の変化

1 ファイルのサイズ	作成するファイルの個数	使われていないブロック数	df(1) Avail の値
-	-	6587953	193136376
0KB	100 万個	6587463	193120696
1KB	100 万個	6462474	189120696
2KB	100 万個	6462474	189120696
4KB	100 万個	6462474	189120696
8KB	100 万個	6337475	185120696
16KB	100 万個	6087462	177120696
32KB	100 万個	5587463	161120696
64KB	100 万個	4587463	129120696
128KB	100 万個	2587463	65120696

消費ブロック数と消費サイズへデータを変換すると次の結果が得られる。

表 4.6 消費ブロック数と消費サイズ

1 ファイルのサイズ	作成するファイルの個数	消費ブロック数	消費サイズ
0KB	100 万個	490 個	15.31MB
1KB	100 万個	12 万 5,479 個	3.83GB
2KB	100 万個	12 万 5,479 個	3.83GB
4KB	100 万個	12 万 5,479 個	3.83GB
8KB	100 万個	25 万 478 個	7.64GB
16KB	100 万個	50 万 491 個	15.27GB
32KB	100 万個	100 万 490 個	30.53GB
64KB	100 万個	200 万 490 個	61.05GB
128KB	100 万個	400 万 490 個	122.09GB

結果をそれぞれグラフに表現すると図 4.7 ～ 4.10 のようになる。

まず、100 万個のファイルを作るには 490 ブロックくらい（15.31MB ほど）は領域を使うのだ、ということがわかる。単純計算でひとつの i-node のサイズが 16 バイトといった計算になる。そして注目すべきは書き込むデータのサイズが 4KB 未満だった場合の動作だ。書き込むサイズが 1KB でも 4KB でもまったく使っているブロックの数も消費サイズも変わっていない。つまり、フラグメンテーションサイズ以下のファイルは、実際にはデータ保持の最小単位であるフラグメンテーションの分まで使うことになるということがわかる。

つまり、このファイルシステムで細かいファイルを大量に保持したいのであれば、少なくともファイルサイズは 4KB 以上であることが好ましい、ということになる。

ファイルシステムの特性を知って性能を使い切る 5

今回は UFS のブロックとフラグメンテーションという仕組みを取り上げ、実際にどういった振る舞いを見せるかを説明した。今回の内容を知っていれば、数バイト程度のファイルを何億個も作るようなシステム構築はとてもディスクスペースの利用効率が悪いことがわかるだろう。ちなみに FreeBSD 10.0 では平均的なファイルサイズは 16KB であるという想定で設

定が組まれているので、1 つあたりのファイルサイズが 16KB くらいになるようなデータ配置を設計できると、この UFS2 の性能を発揮させやすい、ということになる。

図 4.7 消費ブロック数 0KB ～ 128KB

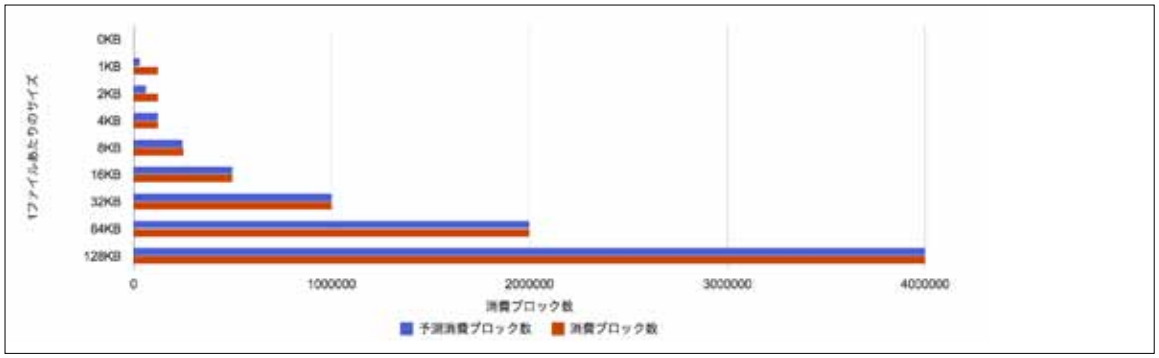


図 4.8 消費ブロック数 0KB ～ 8KB

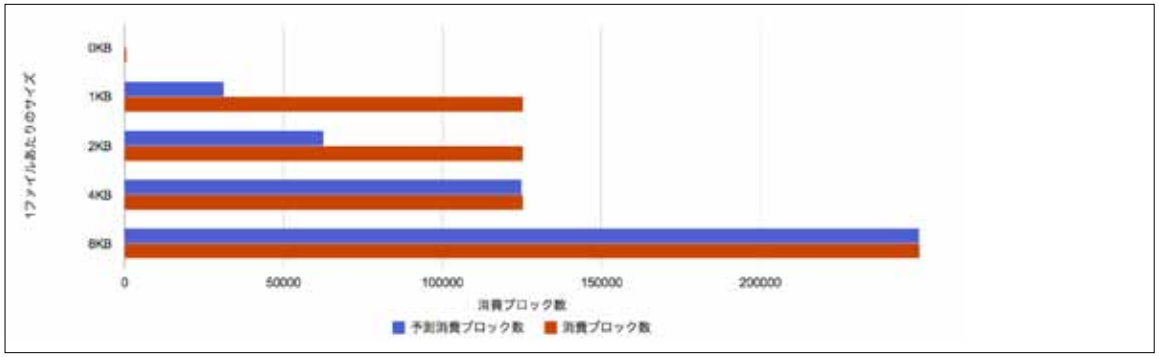


図 4.9 消費サイズ 0KB ～ 128KB

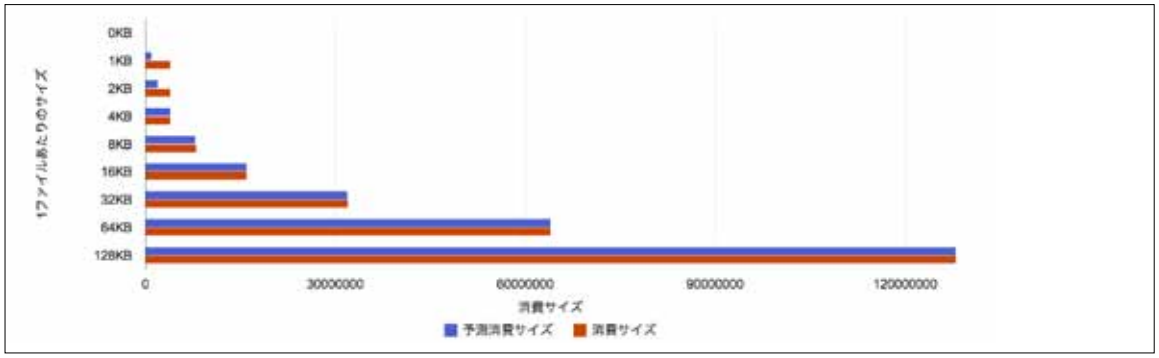
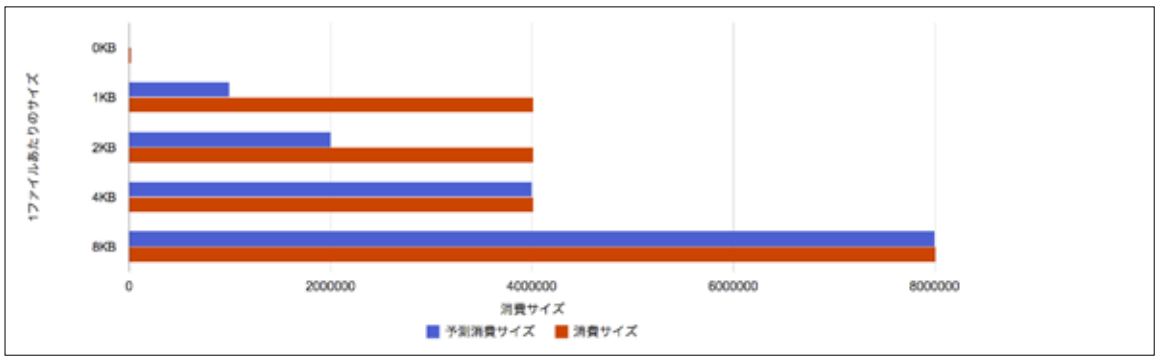


図 4.10 消費サイズ 0KB ～ 8KB



ハッカソン交遊録

written by 藤川恵一

２合目

～山男のつぶやき～

困難な登攀（はん）に重要な価値を見い出す

2014年2月に開催された世界銀行主催の世界防災・減災ハッカソン「Race for Resilience」（通称、「世銀ハッカソン」）、そして2月22日に世界で同時開催された「International Opendata Day」（年に1回ある「世界オープンデータ祭り」とも呼ぶべき一大イベント）の一環として横浜大さん橋会場にて行われた地域市民参加型の「横浜オープンデータ ハッカソン」、そして「International Space Apps Challenge Tokyo 2014」（通称、「NASAハッカソン（4/12、13に東京大学　駒場リサーチキャンパスにて開催）」）に先立つ3月開催のアイデアソンに、筆者は参加してきました。今回の記事では、「ハッカソン」未経験の方のためにその時の内容を振り返りつつ、手に汗握る現場の雰囲気を紹介してみたいと思います。

大雪に見舞われた世銀ハッカソン

今年２月に開催された世銀ハッカソンは全地球的な視点で防災・減災にチャレンジする、テーマ性の非常に高いハッカソンでした。東京会場に選ばれたのは、東大駒場リサーチキャンパスです。東京都心では 20 年ぶりという暴風雪のさなかでの開催となり、雪まみれで家と東京会場との行き来をした参加者も少なくありませんでした。しかし、集まった参加者を見渡すと、人類共通の課題である災害や貧困、社会的課題の解決につながるアイデアを形にしようと挑むチャレンジャー達であふれていました。完成された作品にはいずれもエンジニア、デザイナー、コーディネーターの魂がこもっているように見えました。

山屋エンジニアの筆者としては、ハッカソン参加者の意気込みが、近代アルピニズムの主な思想として名高い 19 世紀末に活躍したイギリスの登山家ママリーが唱えた「ママリズム」（*1）にも一脈通じると感じています。「登山の目的は、単に頂上を目指すだけではなく、技術的により困難な登攀の中に重要な価値を見いだす事」とママリーは説いています。同じ山であっても、より困難でチャレンジャブルなルートを求めて登る熱意や準備や技術の重要性に関し、今でも多くの登山家に影響を与え続けています。

たまたま偶然この日に、東京で猛威をふるったブリザード（猛吹雪、暴風雪）に限らず、人はある日突然、思いがけない時と場所で、**自他共に助け合う最大限の努力や怠りなく事前に実施した（ハズの）準備・訓練成果の最大限の発揮が必要となる状況**に陥る可能性があります。どんな状況下であろうとも、日頃からそれを乗り越えられる準備と心構えはしておきたいものです。（ちなみに、職場から自宅までを、あえて徒歩で移動する経験は「ママリズム」に通じます）。

オープンデータ利用のメリット

さて、ハッカソンの場では、チームメンバーが相互に協力し、社会的 / 企業的課題の解決、はたまたエンターテイメント的テーマなど様々な興味関心の実現を通じて、目に見える作品の開発を競い合います。

評価が付きもののハッカソン。そのため、より良い印象を与える努力は必須で、プレゼンも決して手を抜けません。最終的には、メンバーがお互いの作品を評価し（外部の審査員が判定するケースも多い）、その優劣を競い合って一喜一憂する。**ハッカソンは、そんな楽しいイベント**です。

ハッカソンにおいては、オープンデータを使用しないでアプリ開発を行っても勿論構いません。ただし、オープンデータ利用のメリットとしては、データアクセスを

簡易に行うための各種 API が準備されていたり、標準フォーマットになっていたりすることが多く、アプリ開発に便利で有利な点が多いことが挙げられます。これらのメリットは、オープンデータを使用した即席で奇抜で、かつ自己設定したテーマ性の高いアイデアの実現を強力に後押ししてくれます。

また、オープンデータを使用することで、**オープンデータの需要とアプリの存在をデータ提供組織（政府・自治体、メーカーなど）に強く印象付け、それにより社会的に有益で便利なデータ提供の好循環を産むことの重要性を、データ提供サイドに気付かせる事が可能となります。**この点がデータ提供サイドに対して、さらなるデータ提供の動機付けとなることが期待されます。

現在でも、ハッカソンのスポンサーには、世界的な企業や政府・自治体組織が名前を挙げており、蓄積されたデータや各種 API やクラウド環境の提供を含め、参加者はハッカソンの際に各種の技術情報を取得したり、技術的なサポートを受けたりすることが可能な場合もあります。このように、アドホック（限定目的）な個人のハッカソン開発テーマを、企業や政府・自治体が公の場で大いに支援してくれる。こんな社会現象が今、ハッカソンとして起きている訳です。

ハッカソンな夜

ハッカソンは勝負事とは言っても、参加者の皆さんは、酸いも甘いもかみ分けられる大人。泊まりがけ開発の場合には、深夜に頃合を計って「飲みニケーション」で盛り上がる余裕の「ライフワークバランス・チーム」や、夜なべして必死にプログラム開発をする「デスマーチ・チーム」ありと、悲喜こもごもの様相を呈します。そんな人間関係を一言で云うと、「戦友」でしょうか。

ハッカソンにおける開発テーマやチーム編成は、事前に実施される「アイデアソン」で決まることがほとんどです。その場合、アイデアソンの実施日からハッカソン開催日までは、普通 1 カ月程度あるため、その間、チーム内でのアイデアの膨らましと作品開発を精力的に進めます。一方、実質 2 日のハッカソンでは、「アイデアソン+ハッカソン」で 1 泊 2 日の強行軍となります。

チームは、エンジニア以外にも色々な特技と知識と関心を有するメンバーで構成されます。役割分担としては、コーディネーター、デザイナー、プログラマー、最後に発表するプレゼンターやプレゼン資料作成担当など。自分の希望とチーム内の需給バランスから決まります。特に誰が偉いということもなく、お互いに尊敬と敬意を払いつつ各人平等。そして、チームに対する刺激の供給と、ひたすらなる貢献的出力を与え続けることが要求されます。

ところで・・・自分の上司を連れて参加するのは極力止めましょう。会社でもないのに、他メンバーへのビミョーな仕切りが始まることがあります。あくまで、フリーな立場で未知のメンバーと最大限楽しむ！がハッカソンの醍醐味だと思います。

MA9 で組んだ特殊なチーム編成

私の場合、特殊なチーム編成をした経験があります。2013 年 9 月、ハッカソン「Mashup Awards9 Mashup Camp 札幌」（MA9 札幌予選）の札幌会場でのことです。我がチームは、一見すると筆者ひとりだけ。残りのチームメンバーは、東京在住の伝説のプログラマー、秋元慶太氏と、横浜在住のアメリカ人エンジニアのビーソン・ショーン氏だからです。彼らには関東からリモートメンバーとして参加してもらい、「グローカル対応」で参戦し切り抜けました。ショーン氏とは英語での連絡対応となり、かなりの無茶振りコーディネートになりました。しかし、FaceBook の非公開グループ環境とクラウド環境（GoogleDrive、Microsoft Azure）を駆使することで、開発テーマと開発環境に関する情報をきっちりと共有。どうにかこうにか、アドホックな作品を完成することができました！（その舞台裏では、API と各種データを提供して頂いたトヨタ IT 開発センター様やゼンリンデータコム様や日本 Microsoft 様、これらスポンサー企業メンター諸氏の皆様には我チームのリモートメンバーが大変お世話になりました。この場をお借りしまして、お礼申し上げます m (_ _) m）

開発テーマの選定

アイデアに対する基本的な姿勢としては、「社会的課題の解決」、または「社会的問題に対する対策」に重きを置く参加者が多いのが特徴です。ただし、人や社会に夢と非日常の楽しみを与えるポテンシャルのある、かつ魅力ある発想も尊重され、実用とはかけ離れた遊び感覚のアイデアももちろん、歓迎されます。要は社会貢献、奇抜、天然、マジメ、何でも良いのです。

アイデアをまとめたり情報共有したりするための手段・方法も各種様々。伝統的に大きな模造紙にメモ用

紙を貼ってブレーンストーミングするチーム。XMind などのマインドマップツールを使って合意形成をするチーム。はたまた、大きな教室黒板全面にアイデアと意見を書き出して、そのアイデアスケールを誇るチーム等々。SNS やクラウドの達人が何人かいるおかげで、FaceBook のグループを立ち上げたり、クラウド環境である GoogleDrive の SpreadSheet による同時進行の共同作業が当たり前のように行われたりします。または、ChatWork やチケット方式の backlog で情報共有するのも手です。リモートの分散開発環境を構築し、コミュニケーションロスや打合せの移動時間を最小限に抑えながら効率的に開発作業を進めるわけです。

制限があるのは時間だけ！ 泣いても笑っても制限時間のドラが鳴った瞬間、ハイそれまでよ！の厳しい世界。終了のその時までメンバー間の意見を調整しつつ、開発に没頭する。そこには、**嫌味な上司もいなければ、駄々をこねるクライアントさんもいない（苦笑）。**

短期の超集中スタイルで、自分達の共通テーマに対して最善（と思える）努力を次々と重ね、設定ゴールへの到達を遮二無二目指す。これこそ、自発的でクリエイティブな開発の原点といえる光景です。ただし、ハッカソンは、あくまで勝負事。おまけに短期決戦なので、色々と大変です。ゴールへたどり着ける幸運なグループもいれば、開発の道半ばでタイムオーバーとなる“あー”なチームも散見されます。

前述した MA9 札幌予選チームでは、伝説のプログラマー秋元さんが“奇跡のラスト 30 分”を演出してくれたこともありました。ハッカソンの開発締め切り時刻は、15 時ジャスト。ところが、それまでイソイソとクラウドとのデータ連結を担当してくれていたショーンが、サーバー立ち上げ後の 14 時 30 分頃に本業執務のため、戦線を離脱してしまったのです。このままではクラウド経由の地図ビジュアライゼーションにも赤信号が灯ってしまう。万事休すと東京・札幌間の電話越しにあきらめムードを漂わせていた筆者を横目に、冷静沈着な秋元さんは残り 30 分で、クラウド連結無しの直接方式で、トヨタ車載プローブデータを加工して得た津波波面データの地図ビジュアライゼーションを、成功させたのです。秋元さんがほぼ徹夜でトヨタ社製・ゼンリン製の各種 API を習熟してくれていたおかげで、最終局面においても集中力と粘りをピンポイントで発揮できたのです。おかげさまで、「トヨタ IT 開発センター賞」を頂くことができた次第です。

このように土壇場で普段経験できない信じ難い才能や努力を垣間見る瞬間に立ち会う幸運に恵まれることも、よくあります。とはいえ開発の当事者にとっては、1 泊 2 日の地獄かもしれませんね (~~;;

勝負の分かれ目となるプレゼンタイムでは、PowerPoint を使った発表が多く見受けられます。Prezi などでちょいと気の効いた動画的プレゼン資料を作るチームは稀ですが、このような新発想のプレゼンツールに挑戦してみるのも楽しいものです。アプリ作品も、余裕があればモックアップだけではなく、完成した動く実物を披露して実現性もアピールします。

審査員を前にしたプレゼンタイムでは、前の晩に徹夜して頑張っていた隣チームプレゼンターの発表内容に感心したり爆笑したりと、それまでの苦難オンリーの時間を一気に忘れ、最高に楽しい瞬間の到来となります！

ハッカソンの規模はグローバル

世銀ハッカソンと同様に、地球規模で開催されるハッカソンもいろいろとあります。

● NASA ハッカソン

毎年 4 月に開催される「International Space Apps Challenge Tokyo（ISAC Tokyo）」（通称、「NASA ハッカソン」）は、昨年は世界 44 カ国・83 都市で開催され、参加者は約 8,200 名にも上る世界最大規模のハッカソンです。

このイベント名称の中に、なぜ「Challenge」の文言が？と気になった方もいらっしゃるかもしれませんね。これは、アポロ 13 号の事故（*2）に由来していると考えられます。当時のアポロ 13 号事故発生直後、NASA の地上基地では、日頃顔を合わせたこともないような分野のエンジニアや科学者が、事故原因の究明や対策のために多くのチームを組んで協力し合い、それまで考えも及ばなかったようなアイデアにチャレンジして、限られた時間、限られたメンバー、限られた材料の中、あらゆる観点から宇宙船帰還に尽力。その後、NASA においてもハッカソン的開発体制やトレーニングの重要性が見直され、さらには、世界的なチャレンジングイベントとして NASA ハッカソンを主催しています。これにより NASA ハッカソンは、世界中の

エンジニアや科学者やデザイナーなどに、グローバルでチャレンジャブルな宇宙関連のアイデア実現のチャンスを与え続ける場となっています。

宇宙がテーマのNASAハッカソンでは、妄想（失礼！）優先なアイデアの人気が高く、困難で際立ったアイデアの実現に本気で立ち向かう「宇宙人集団」があちらこちらで多数出現します。毎年３月、東京・日本科学未来館で実施されるアイデアソンから、東大駒場リサーチキャンパスで実施されるNASAハッカソン東京ローカル本番までの１カ月間は、我々参加者の頭の中は寝ても覚めても「宇宙」「宇宙」「宇宙」…そう、「宇宙オンリー」「ミッションは宇宙」状態！苦しくも熱い１カ月が待っています。

● International Open Data Day（IODD）

世界的な規模では、この「International Open Data Day」も負けてはいません。毎年２月、年度末で納期間際の超忙しい時期に実施されるので有名な、オープンデータの世界的お祭りイベント「International Open Data Day」も非常に多くの都市で開催されます（2014年２月22日実施：世界158会場、国内では北海道の旭川～九州は佐賀までの32会場。開催結果は、http://odhd14.okfn.jp/?p=784を参照）。今年、筆者が参加した「YOKOHAMA International Open Data Day 2014」の会場（竣工120周年を迎える横浜大さん橋会場）では、「横浜開港の地「大さん橋」から世界最先端のIT国家・日本のハブとなる「データ・ポート横浜」を世界に発信しよう！」を合言葉に、全国最大規模（参加者約180名）の多彩なイベントが実施され、筆者もAR（拡張現実）を使った横浜歴史街歩きツアーガイドをさせていただいたり、ハッカソン参加でビッグデータ対応のアプリ開発に参加したりと、大変楽しめました。また、筆者も設立幹事の一人となった「Code for Kanagawa」（*3）の設立宣言も同時に行うことができ、神奈川県内のエンジニアやデザイナーや意思ある皆さんと共に、社会的な課題をICTとオープンデータを軸に解決するミッションを持った団体の産声を上げることができ、気も引き締まる思いです。今後は、ハッカソン参加者ではなく、ハッカソン主催の立場としてハッカソン活動に携わっていくことができれば幸いと考えています。

膨らむアイデアと制約条件の狭間で

筆者を含め、参加者は「未知のテーマ」「未知のメンバー」「未知のツール」「未知の主催者やスポンサー（たとえば世銀・NASA・JAXA・ローソン・総務省・テレビ局等々）」に出会う楽しみに心弾ませながら、ハッカソンのエントリーボタンをポチります。そのテーマが困難であればある程、よけいに燃えるタイプの人種が多いハッカソンでは“異能チャレンジャー大会”の様相を呈します。

限られた時間、限られたメンバーという制約の中、大きなテーマを深く考え抜き、エッセンスを絞り込む。さらにそれをメンバー各自が、いっそう深くえぐってえぐり抜く（この段階は、心理的にも精神的にも技術的にも苦痛を伴うことが多いため「えぐる」と表現しました）。粒度と分野の異なる方針や課題を短時間で合意形成しながら、「膨らむアイデア」vs.「制約条件（時間、手持ち技術、メンバー数）」という対立軸の中で発生する葛藤を切り抜け、核となる心臓部のアイデアだけをIT＆デザイン実装する。そんなアウフヘーベン型チーム開発作業の重要性に、否が応でも気付かされる現場。それがハッカソンです。

１泊２日といった短期間ですが、“ハッカソンで極める挑戦的技術開発の実践！”をチームで共通体験し、翌日から自分や仲間の「社会を見る視点の変化を共に感じてゆく」のも楽しいものです。これは運命をお互いに紡ぎ上げていく、または、お互いの心の共通土台を創り合う作業、とも言える体験となります。

また、「社会的課題を見つめ対策を探る視点を、技術や科学、はたまた他者の哲学を通じて直接学ぶ」意義とその重要性を、ハッカソンは今更ながら気付かせてくれます。

---

*1　世界大百科事典、http://kotobank.jp/word/ ママリズム

*2　アポロ13号とNASAハッカソン
https://readyfor.jp/projects/SpaceApps2013

*3　Code for Kanagawa
https://www.facebook.com/CodeForKanagawa

◆2014年2月に開かれた世界銀行主催の世界防災・減災ハッカソン「Race for Resilience」（世銀ハッカソン）
(c) レース・フォー・レジリエンス実行委員会、CC BY

ハッカソン「Mashup Awards9 Mashup Camp 札幌！」

活動1

①イベント趣旨	**Mashup Awards を媒介に、「人」と「人」が出会い、新しいチームが生まれる、「アイデア」と「アイデア」が出会い、新しいサービスが生まれる「企業」と「企業」が出会い、新しい Mashup が生まれる、「機会」と「人」と「アイデア」が出会い、新しい可能性が生まれる。**
②実施日・会場	2013/9/15、9/16　（札幌）Sapporo-cafe
③主催	Mashup Awards 実行委員会（伴野　智樹）、主幹事：Recruit、Media Technology Labs
③ゴールドパートナー	Biz-IQ、日本 Microsoft、楽天、Recruit
③シルバーパートナー	ブログウォッチャー、ゲッティ イメージズ ジャパン、グレースノート、KDDI ウェブコミュニケーションズ、エヌ・ティ・ティ・ドコモ、PUX、トヨタ IT 開発センター、ヤフー、ゼンリンデータコム
③インフラパートナー	Apigee、ニフティ
③メディアパートナー	CodeIQ、@ IT、Engadget、finder、Gihyo Digital Publishing、HTML5 Experts.jp、in the looop、ITpro、TechCrunch Japan、Tech-Tokyo、TechWave、Tech 総研、ワッフル /WAFL、週刊アスキー／週アス PLUS
③協力企業	ACCESSPORT、アドビシステムズ、Apigee、アシアル、ブログウォッチャー、カーリル、クレイド、サイバーエリアリサーチ、サイボウズ、デジタルガレージ、デザインエッグ、Dozens、ドワンゴ、エバーノート、Facebook、FC2 Inc.、FullCourt、ガラポン、ジオとす、ゲッティ イメージズ ジャパン、ぐるなび、ゴーガ、グレースノート、はてな、ハートレイルズ、ヒトクセ、hon.jp、ignote、KDDI ウェブコミュニケーションズ、ケイピーアイ、ロックオン、マピオン、メタデータ、Microsoft、ミクシィ、もぎゃんらんど LLC、モリサワ、国立情報学研究所、ニフティ、エヌ・ティ・ティ・データ、NTT ドコモ、ヌーラボ、Overtex、ビーシーフェーズ、PUX、楽天、リクルートキャリア、リクルートライフスタイル、ソニックガーデン、TINAMI、トヨタ IT 開発センター、ヤフー、ヤマレコ、Zaim、ゼンリンデータコム
④イベントパートナー	ANNAI、エイチーム、ビットスター、e-ZUKA Tech Studio、福岡工業大学短期大学部、九州インターメディア研究所、One Panasonic、塩尻インキュベーションプラザ、Technology & Entrepreneurship TOKUSHIMA、東京証券取引所「ソーシャルかぶコン 2013」
④チーム名	「Mt.Fuji」
⑤チームテーマ	「TUNAMI Front Line monitor」水没車両のプローブビッグデータを解析して、津波の現在位置をリアルタイムに計測する。
⑥メンバー	藤川恵一、秋元慶太（東京リモートメンバー）、ビーソン・ショーン（横浜リモートメンバー）
⑦受賞歴	「トヨタＩＴ開発センター賞」
⑧筆者担当	コーディネーター
⑨ URL	http://mashupawards.tumblr.com/post/61487367552/ma9-mashup-camp

ハッカソン「Race for Resilience」（通称：「世銀ハッカソン」）

活動2

①イベント趣旨	**世界銀行 東京防災ハブの立ち上げを記念し、ICT を防災・減災に活用し、イノベーションを推進することを目的としたグローバル防災・減災ハッカソン。**
②実施日・会場	2014/2/8、2/9　東大駒場リサーチキャンパス、石巻商業高校、名古屋大学
③主催	レース・フォー・レジリエンス実行委員会
③メインパートナー	Code for Resilience、世銀防災グローバル・ファシリティ（GFDRR）、世界銀行東京事務所、CODE for JAPAN、Csis（東京大学　空間情報科学研究センター）、イトナブ、SIDC（名古屋大学　社会イノベーションデザイン学センター）、MAPconcierge、NPO　Press Center、visualizing.jp、Code for Nagoya
③サポーター	Google、ソフトバンク BB、ESRI ジャパン、日本マイクロソフト、さくらインターネット、Amazon Web Service、IDC フロンティア、スイッチサイエンス、アフレル、mozilla JAPAN、ローソン、ロフトワーク、ウイングル、松原荘、Fablab Japan、Realpower、Fujitsu
③連携コミュニティ	Code for Japan、OpenStreetMap、Ushahidi、SAHANA、OSGeo、OpenRelief、Mozilla Factory、IT x 災害、助けあいジャパン、市民キャビネット e コミ・プラットフォーム、Safecast、Bithaus、Health 2.0 Fukushima Chapter、GigaPan、Hack for Japan、relief20、GeoHex、LP-Address、Shinsai FaxOCR、WorldShift、防災ガール、エフスタ !!、html5j、逃げ地図、ASEAN Community – Japan、ISHINOMAKI 2.0、次世代宇宙システム技術研究組合、ユレッジ、iSPP 情報支援プロボノ・プラットフォーム、GESTISS、International Space Apps Challenge Tokyo、Peatix、Doorkeeper、福島 GameJam、Drone Adventures、災害対応訓練研究所、市民救助隊、ユナイテッドピープル映画祭、伊能社中、位置情報サービス研究機構、サイバーフォレスト・ひょうたん島ライブモニタリング、VOLOCITEE Inc.、東京大学被災地支援ネット、東日本大震災アーカイブ、ヤフー石巻復興ベース、ツール・ド・東北、D-Leg、See-D Contest、みちのく震録伝、RCF 復興支援チーム、UX TOKYO、上山集落、academycamp、finder
④チーム名	「BigDataForDisaster 」
⑤チームテーマ	世界で発生する災害 Twitter データから、言語に依存しない情報（顔文字）を切出し、離散化された感情情報の分布から被災状況を把握。
⑥メンバー	山本光穂（デンソーアイティーラボラトリ）、鈴木翔太、千代田洋平、中出悠馬、不二門宏城（以上、駒澤大学）、会津泉（ハイパーネットワーク社会研究所）、久保隆宏（ＴＩＳ）、藤川恵一
⑧筆者担当	ビッグデータインフラ立ち上げ
⑨ URL	http://raceforresilience.org/

「International Open Data Day 2014 in YOKOHAMA」

共通項目

①イベント趣旨	**世界的なオープンデータの潮流をさらに加速していくため、世界中の各都市で、年に一度、オープンデータに関連する様々な催しを一斉に開催する国際的なイベント。**
②実施日・会場	2014/2/22　（横浜市）横浜港大さん橋 CIQ プラザ、マリンタワー～横浜港大さん橋までの AR 街歩きツアー
③主催（共催）	横浜インターナショナルオープンデータデイ実行委員会（事務局：藤村　良弘 / 横浜コミュニティデザイン・ラボ）、横浜オープンデータソリューション発展委員会（委員長：杉浦　裕 / ヨコハマ経済新聞）（共催：横浜港大さん橋国際客船ターミナル指定管理者 相鉄企業）
③後援	オープンデータ流通推進コンソーシアム、横浜市政策局、グリー、日本マイクロソフト、キヤノン IT ソリューションズ、リスト、野毛印刷社、山手総合計画研究所、NTT 東日本神奈川支店、日本アイ・ビー・エム、野村総合研究所、LODI、横浜コミュニティデザイン・ラボ、オープンソースソフトウェア・シティ、神奈川県、横浜市中央図書館、エヌ・ティ・ティ・データ
③協力	横浜市芸術文化振興財団、オープン・ナレッジ・ファウンデーション・ジャパン、東京都市大学上野研究室、岡部研究室
⑨ URL	http://yokohamaopendata.jp/?p=187

活動1

④チーム名	**AR 歴史まち歩き**
⑤チームテーマ	マリンタワーから大さん橋までのＡＲ街歩きや、5m ☒ 4m の古地図のガリバーマップの AR 体験。大画面でいくつかの時代の古地図を重ね合わせて見るプロジェクトマッピング。
⑥メンバー	東京都市大学（上野研究室、岡部研究室）、谷杉歩音（ナイトレイ）、村上一樹、河北直治（フリービジネスデザイナー）、藤川恵一
⑧筆者担当	ＡＲツアーガイド
⑨ URL	https://www.facebook.com/events/823791244313278/

活動2

④チーム名	**C4K 大桟橋部隊（横浜オープンデータ ハッカソン）**
⑤チームテーマ	震災時に多数生産されると想定されるヒートマップデータをＡＷＳ上のビッグデータ処理環境にて追加加工処理を行い、処理結果を新規にヒートマップ表示する。
⑥メンバー	藤川恵一、渡辺登、ビーソン・ショーン
⑧筆者担当	コーディネーター
⑨ URL	http://yokohamaopendata.jp/?p=223

縁の木、育てよう

第一回

縁（えん）つなぎ

written by **白羽玲子**

こんにちは。

このコーナーを担当させて頂くことになりました白羽と申します。

IT、ビジネス書関連の出版社に身を置き、広告営業として長年関わってきましたが、この４月からがらりと方向転換してコーヒー豆の焙煎を生業しています。

まったくの異業種転職に踏み切ったのはちょっとしたわけがあります。自閉症の次男坊の小学校進学を来年に控えて、将来的に知的障害や精神障害を持つ方たちでも就業できたり、就業訓練に使ったりできる事業を立ち上げられないものか、と考えたのがきっかけでした。

“デブサミ”って？

退職してみると、会社のブランドとか、媒体の力といった宝物を失うわけで、今まで関わってきた方たちとのご縁が自分の財産の全てなのだと改めて感じるようになりました。せっかくだからご縁のつながりを大きく育てようと、興した会社を「縁の木」と名付けました。

まだまだ芽吹いたばかりの小さな会社ですが、これからゆっくりと育てていきたいと考えています。

さて、自己紹介はこれくらいにして。

このコーナーですが、お話を頂いた当初は技術コミュニティをご紹介していく企画だったのですが、検索できる情報ではなく、人にスポットを当て、関わっているコミュニティや技術、最近凝っていること、気に入った書籍や音楽など、お話を伺う企画とさせていただきました。次の回に登場する方もご紹介いただき、ご縁をつなげていこうと思っています。

第1回はちょうど大きな開発者系イベントが開催された時期でしたので、技術者コミュニティの交流の場として元々立ち上がったこのイベントをご紹介したいと思います。（私の古巣でもあります！）

イベントの名は Developers Summit、といいます。Developers Summit（通称デブサミ）は「技術者コミュニティとの連携」を掲げた IT エンジニアカンファレンスです。2003 年 2 月の立ち上げ以来、毎年 2,000 名もの来場者

コミュニティで活躍する
エバンジェリストやエンジニアのみなさんに
毎号お話を伺い、次の方を紹介頂きながら
人のご縁をたどっていく、
そんなコーナーです。

が足を運んでいます。

カンファレンス主体のイベントであるにもかかわらず、デブサミはエンジニアのお祭りと言っていいほどの賑わいを見せています。また、さまざまな技術やプラットフォーム、そしてコミュニティが一同に会することで、エンジニア同士の出会いと交流の場として機能しています。これだけ大きな規模のカンファレンスが成立するのは、スポンサーだけでなく、コミュニティの方々の手厚いサポートあってのことなのです。

デブサミでは「オフィシャルコミュニティ」を公募し、イベントの告知にご協力頂くとともに、ブース展示やオープンセッション、コミュニティ LT（ライトニングトーク）での発表の場を提供しています。（もちろん、定期的にオフ会など開いて活動しているか、メーリングリストが機能しているか、など、審査があります）

関西でも開催

今年は言語、クラウド、開発プロセス、モバイルなど、32団体もの幅広いコミュニティにご参画頂きました。

これまでにない試みとして、コミュニティの方々の協力の下、センサー&デバイス祭というコーナーを設け、ソフトウェアとデバイスの融合による新しいアプリ開発の可能性を紹介しました。VR デバイスや KINECT、LeapMotion などのモーションセンサーを活用した展示コーナーは、多くの人の注目を集めていました。またコミュニティ有志によるオープンジャムと呼ばれるショートセッションでは、たくさんのみなさんが足を止め、発表者のプレゼンに真摯に耳を傾けていました。

Twitter、Facebook など、SNS での盛り上がりも大きく、まとめが各セッション毎に作られるなど、独自の展開も見せ始めています。（アカウントは @devsumi、ハッシュタグは #devsumi）

今後のデブサミですが、8 月にエンタープライズ開発者向けのカットアップ版を開催し、9 月には小規模ながら関西でも開催する予定です。そしてデブサミは、来年の 2 月にまた東京に帰ってきます。夏版は特に業務システムの開発者の方たちのためにセッションを組んでいますので、冬とは趣の違うイベントに仕上がっています。是非、読者のみなさまも足を運んでみてください。

なにしろ、コミュニティあってのデブサミです。次回募集の際は、ぜひぜひふるって来場者として、オフィシャルコミュニティとして、ご参加ください。もちろんスポンサー様も募集中です！みなさまとお会いできるのを楽しみにしております。

今回はここまで。また来月お会いしましょう！

デブサミオープンセッションの様子。事前予約不要ですがたくさんの人が集まってきます。会場のあちこちで新しい出会いが生れます。

参画コミュニティ一覧（五十音順）

アジャイルプロセス協議会
アジャイル UCD 研究会
Windows 女子部
hcdvalue
html5j
NPO 法人 IGDA 日本
NPO 法人 SESSAME
IT スキルアップ応援コミュニティ「エフスタ !!」
O-One
codeseek
こみゅぷらす (COMU+)
コワーキング協同組合
shinagawa.redmine
Shibuya.rb
Japan Sencha User Group
先端 IT 活用推進コンソーシアム（AITC）
DAD-Japan
TOCfE Boot Camp
DevLOVE
Delphi ユーザー会準備会
Tokyo Motion Control Network
東北デベロッパーズコミュニティ
日本 CloudStack ユーザー会
日本 Cloud Foundry グループ
日本 Selenium ユーザーコミュニティ
日本デジタルゲーム学会
日本 MySQL ユーザ会
日本 UNIX ユーザ会
非営利活動法人 JASIPA
POStudy ～プロダクトオーナーシップ勉強会～
要求開発アライアンス
リーン開発の現場 塹壕コミュニティ

HANDS LAB

ユニケージ®エンジニア数 最大級!

ハンズラボはユニケージ®開発手法に特化したITソリューション企業です。東急ハンズの営業システムを刷新したノウハウを駆使し、小売業における「オーダーメイド」のシステム開発を行います。

中途採用 大募集!詳しくはHPで!

› http://www.hands-lab.com/

「ユニケージ®」は有限会社ユニバーサル・シェル・プログラミング研究所の登録商標です。

これであなたもユニケージエンジニア!

ユニケージ開発手法教育講座

「ユニケージ開発手法教育講座」は、ユニケージ開発手法におけるデータ管理の方法や、オリジナルコマンドの使用方法などをハンズオン形式で具体的に学べる講座です。UNIX の基礎からユニケージ開発手法による開発プロジェクトの進め方まで、ユニケージエンジニアとしてのトータルスキルを習得できます。

http://www.usp-lab.com/LECTURE/CGI/LECTURE.CGI

- K1 コマンド学習編
- K2 シェルスクリプト学習編（帳票・バッチ処理）
- K3 シェルスクリプト学習編（ウェブアプリケーション処理）
- K4 ユニケージアーキテクチャ編
- K5 ユニケージ開発環境セットアップ編
- K6 システム運用・管理編
- K7 プロジェクトマネジメント・人材育成編
- KSQL SQL エンジニアのためのユニケージ活用編

続々と新講座も充実中!

UNIX 初心者のための講座などもご用意しています。

TechLION For Independent Engineers

技術の草原で百獣の王を目指す
エンジニアたちの新感覚トークライブ!

http://techlion.jp/

上記のサービス内容は 2014 年 4 月現在のものです。
最新の情報は弊社ホームページにてご確認ください。

ISBN 978-4-9048-0707-1
C3455 ¥500

9784904807071

USP 研究所
定価（本体 500円+税）

1923455005002