

FlashLite最適化の話

サイボウズ・ラボ株式会社 鴨志田 良和
kamoshida@labs.cybozu.co.jp

今日の話のまとめ

A = 100; のバイトコード表現

- ▶ バイトコードで表現すると

96 03 00 00 41 00 96 05

00 00 31 30 30 00 1D

15bytes

- ▶ 一方、A=ord("d"); は、

96 03 00 00 41 00 96 03

00 00 64 00 32 1D

14bytes

- ▶ 結果は同じなのに 1バイト減る

← **ここがニヤニヤするポイント**

本日の話題 – Flashの最適化

- ▶ 前編: よりコンパクトに
 - サイズ制約のきついFlash Lite 1.xで、SWFのサイズを減らすテクニックを紹介
- ▶ 後編: より高速に
 - Flash Lite 2.x(AS2)でも利用可能な高速化テクニックを(少し)紹介

前半: コンパクト化

Flash Lite 1.x

- ▶ 携帯電話向けの軽量版Flash
- ▶ Flash4をベースに作られている規格
- ▶ 使えるASはAS1
- ▶ SWFファイルサイズの制約
 - 多くは100KB以下
 - 圧縮(Flash6以降)は非対応

SWFサイズの節約

- ▶ アクションスクリプトをいろいろ工夫すると出来ます

冒頭の例: 数値の代入

```
A=100;
```

```
960300 push [A]
004100
960500 push [100]
0031303000
1D      setvar
```

```
A=ord("d");
```

```
960300 push [A]
004100
960300 push [d]
006400
32      ord
1D      setvar
```

15 bytes → 14 bytes

最初の1文字を取り出す

▶ substring(A,1,1)

```
96 03 00 00 41 00
```

```
96 03 00 00 31 00
```

```
96 03 00 00 31 00 15
```

19bytes

▶ chr(ord(A))

```
96 03 00 00 41 00
```

```
32 33
```

8bytes

変数名を短くする

- ▶ 一般に、変数名を短くすればコードサイズは短くなる
 - コンスタントプールを作ることができない
FL1.xではより重要なこと
- ▶ AS1で文字列に利用できるのはA-Z, \$, _の28種類
 - 28個の1文字変数が見える
 - AS1の変数はcase insensitive

ここで少しAS1の勉強

シミュレートした配列

- ▶ Flash4(Flash Lite1.x)ではネイティブの配列データ構造をサポートしていません。ストリング処理で配列をシミュレートする必要があります。
- ▶ `A[10]`は使えないので、`A_10`という変数で代用する

```
i = 10;
```

```
eval("A_" + i) = 123;
```

これだけでも十分変態的

ここで、

```
eval("A_" add 10) = 123;
```

は

```
A_10 = 123;
```

と同義

```
eval("A") = 123;
```

は

```
A = 123;
```

と同義

**evalの引数には任意の文字
列を書けそう。では、**


```
eval("1") = 123;
```

や

```
eval("¥t") = 123;
```

とかも使えるのか?

OKです (代入・参照ともに可能)

¥x00, :, . など、使えない文字もあります
ハンカクカタカナもOK

したがって、

**100以上の1文字変数が
利用可能！**

AS1の話終了。

ここからバイトコードの話

FL1.xはFlash4ベース？

- ▶ ベースにしてるってことはまったく同じではない？
 - FSCommand2で電波状態を取得したりはできる
 - それ以外に、何か違いはあるの？

FL1.x機種のPlayerバージョン

機種	Flash Playerバージョン
au W31T	5.2.28.0
au W31CA	5.2.15.0
au W41T	5.2.29.0
au A5522SA	5.2.29.0
DoCoMo SO902i	5.2.52.0
DoCoMo P703iμ	5.2.52.0
DoCoMo F901iC	5.2.15.0
DoCoMo SA700is	5.2.15.0

各端末で\$VERSIONを表示してみると、Flash Playerのバージョンはみな5.xであることがわかる

Flash5のアクションが
FL1.x機種でも動くかも?

実際試してみた

- ▶ 動いたFlash5アクション
 - Increment(0x50)
 - Decrement(0x51)
 - Duplicate(0x4C)
 - Swap(0x4D)

使用可能なFlash5アクション

- ▶ FL1.x機種でも一部のFlash5アクションが動作する
 - しかし、それを考慮したコードを生成するコンパイラは知られていない
 - SWFヘッダはバージョン4でないと動かない
 - 本当に全端末で動くかはよくわからない

最適化の例

▶ A+1

96 03 00 00 41 00

96 03 00 00 31 00 0A

13bytes

▶ ActionIncrementの使用

96 03 00 00 41 00 50

7bytes

Pushの最適化

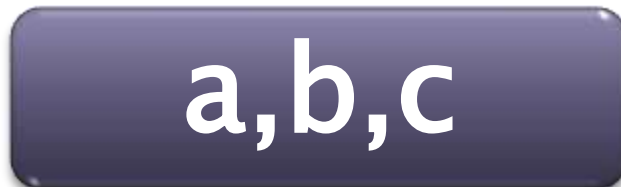
▶ Push命令

- あとで他の命令で使うために値を「準備」する命令
- 同時に複数の値をPushすることが可能 (Flash5以降の機能らしい)
 - ヘッダ部が節約可能

最適化前



最適化後



Pushの連結 – Case Study

$a \times 3 + 11$ の例

960300 push [a]

006100

1C getvar

960300 push [3]

003300

0C multiple

960400 push [11]

00313100

0A add

960A00 push [11,3,a]

00313100003300006100

1C getvar

0C multiple

0A add

(計算順序の入れ替えも
行っている)

22bytes → 16bytes

FL1.xでやってはいけないこと

- ▶ ||, &&の使用
 - できる限りif文をネストすべきです
 - さもないと、2のべき乗のオーダーでコードサイズが増えます
 - (demo)

後半：高速化

FlashLite2以降

- ▶ SWFの圧縮が効くので、サイズ節約を頑張ることはあまり意味がない
- ▶ ただし、実行するアクションの数が減ればそれだけ高速化が可能である
- ▶ したがって、サイズを減らす努力をすればそのまま高速化につながる(はず)

FlashLite2 SWFの最適化の例

▶ SetRegisterに関する最適化

- レジスタに設定した値をすぐに使用する場合は値を捨てる必要がない

8702 setreg 2

17 pop

9602000402 push [reg:2]

} この部分を
省略可能

Pushの連結

- ▶ FL2ではPushは連結される
 - オペランドの入れ換えを行って、よりPushが連続するようにする

• $a + 1 \rightarrow 1 + a$

push [a]	push [1]	} この部分を 連結可能
getvar	push [a]	
push [1]	getvar	
add	add	

Push連結の効果

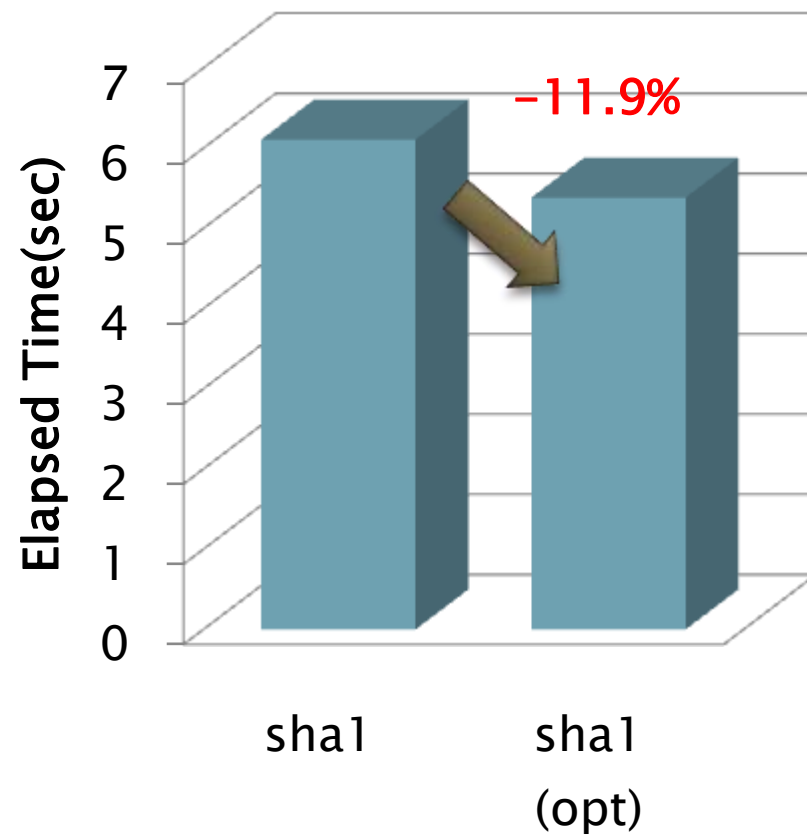
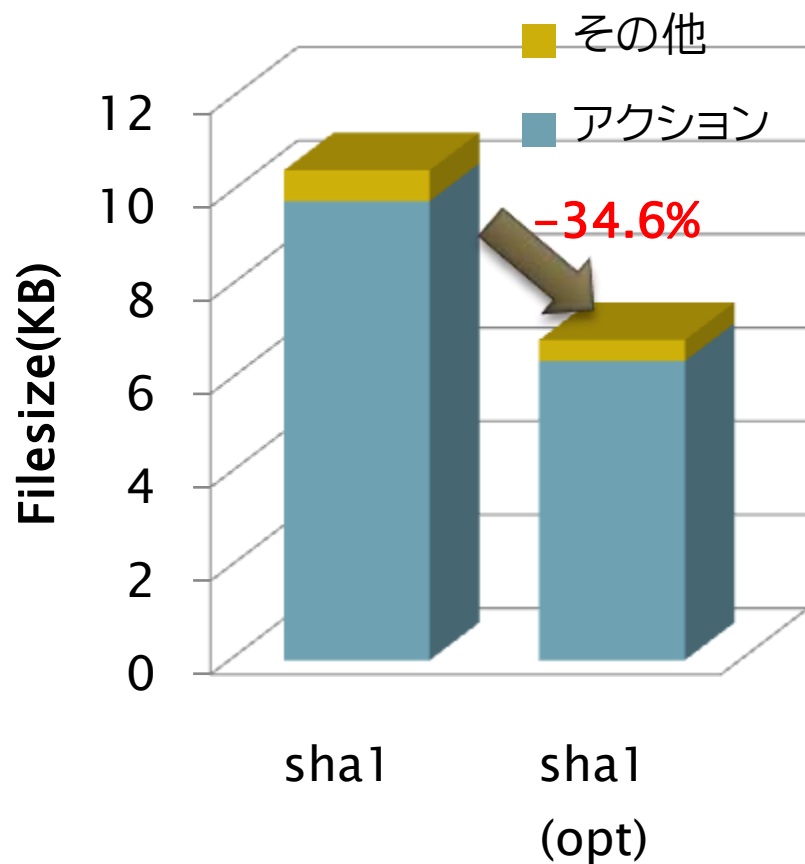
- ▶ SoftBank 821SH上での実行時間を計測
 - 1万回のループの実行時間から計算
- ▶ `push[null]`の実行時間
 - 3.5~4μsec
- ▶ ヘッダ部のオーバーヘッド
 - 約1.5μsec

実験

実験

- ▶ 使用したプログラム
 - sha1: SHA1計算(FL1.1版)
 - FL1.xにはビット演算がないので文字列表現の16進数の多倍長演算ライブラリを作って実装したもの
 - sha1(opt)に対して行った最適化
 - Increment, Decrementなどへの変換
 - オペランド入れ換えと連続Pushの連結
 - SWFのサイズと実行速度(SoftBank 821SH上)を計算

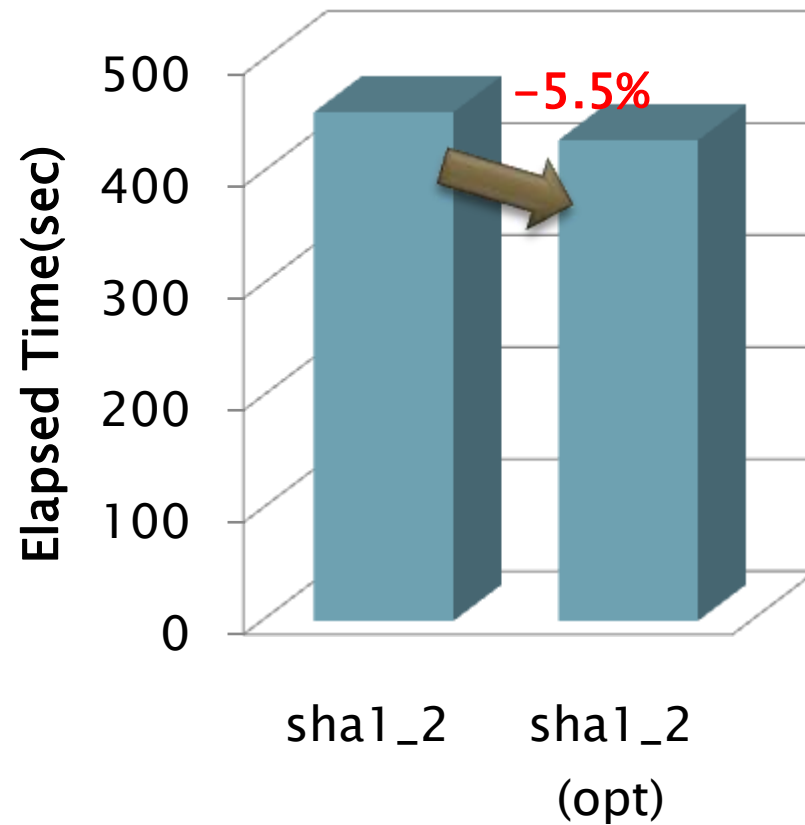
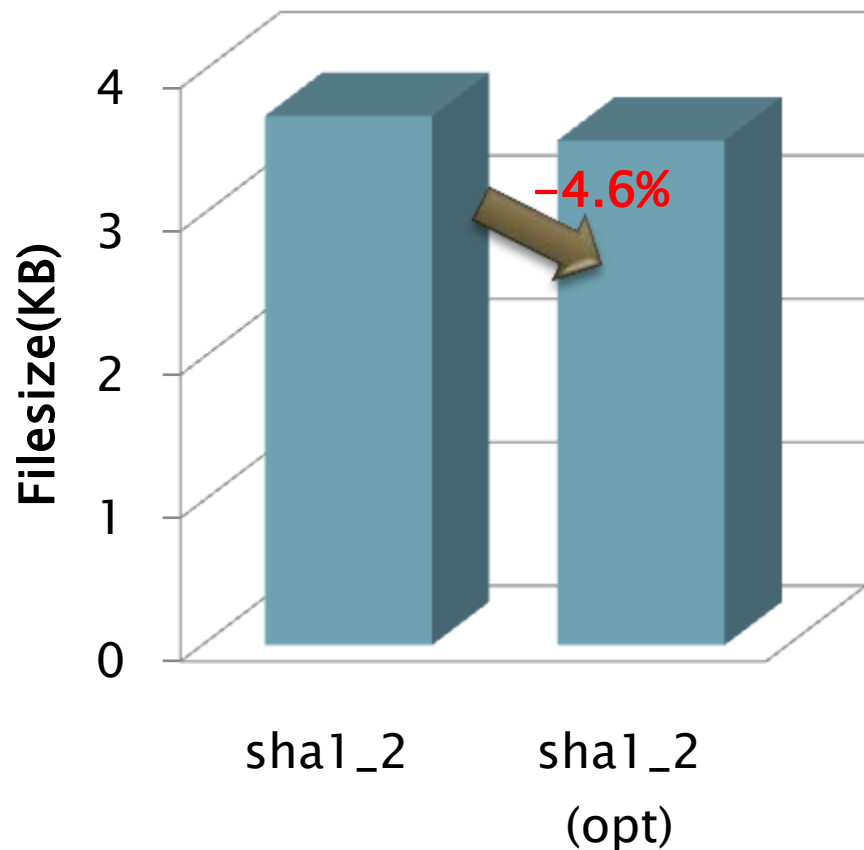
実験結果(FL1.1)



Flash Lite 2

- ▶ SHA1 計算(FL2 版)
 - JavaScript 版高速 SHA1 計算ライブラリを ActionScript に移植したもの
 - http://labs.cybozu.co.jp/blog/mitsunari/2007/07/sha1_1.html

実験結果(FL2)



↑圧縮後のサイズ
圧縮前だと10%程度減る

まとめ

- ▶ SWFのサイズを節約する方法をいくつか紹介しました
- ▶ サイズが減ると、普通は実行命令数が減るので高速化も期待できます
- ▶ FL2(AS2)ではまだ効果がいまいち。
 - まだ最適化可能なことはあるのでいろいろ試してみたい
- ▶ AS3でもあそんでみたい

ありがとうございました！

ニヤニヤできましたか？