

@ LL Future

2008/08/30

MORITA Hajime <omo@dodgson.org>
<http://steps.dodgson.org/>

自己紹介: 森田創

- 本業
雑用系 C++ プログラマ なぜここに....
- 今日の立ち位置
ActionScript VM 「**Tamarin**」擁護派(という設定)
 - ActionScript はよくわからないけど
 - Tamarin は C++ なので安心

Tamarin との馴れ初め



Tamarin との馴れ初め

2006/11/07

「Adobeが Flash_(の何か)をオープンソース化！」

http://weblogs.mozillazine.org/roadmap/archives/2006/11/project_tamarin.html

- Adobe のすばらしい C++のコード※が読める！！
- Flash のレンダリングエンジンが読める！！！！

※Adobe Source Libraries
<http://stlab.adobe.com/>

```
cvs -d :pserver:anonymous@cvs.mozilla.org:/cvsroot mozilla/js/tamarin
```

Tamarin との馴れ初め

2006/11/07

「Adobeが Flash(の何かVM)をオープンソース化！」

http://weblogs.mozillazine.org/roadmap/archives/2006/11/project_tamarin.html

- Adobe の~~すばらしい~~ C++のコード※が読める！！
すばらしくなかった……
- Flash の~~レンダリングエンジン~~が読める！！！！
入ってなかった……

Tamarin との馴れ初め

- 読んだら情が湧いた
 - あれで少しはいいところ(JIT)もあるの...
 - いまさら別れるなんて...
- ActionScript にも興味がでた
 - **複雑**な言語仕様+**強力**なプラットフォーム
 - Yet Another BK Language の誕生か？
 - もう C++ (とperl) だけじゃない！
 - 食い扶持にはこまらなそうだ！

Tamarin みどころ

- 床屋談義な面白さ

- オープン vs クローズド: **Mozilla vs. Adobe**
- 標準規格 vs 独自規格: **HTML+JS vs. Flash**

- テクニカルな面白さ

- 動的言語と JIT (Tracing JIT)
- VM 性能対決: vs. SpiderMonkey, vs. SquirrelFish
- 超巨大レガシー(Gecko)との統合

自己紹介ここまで

FlashCC by Adobe

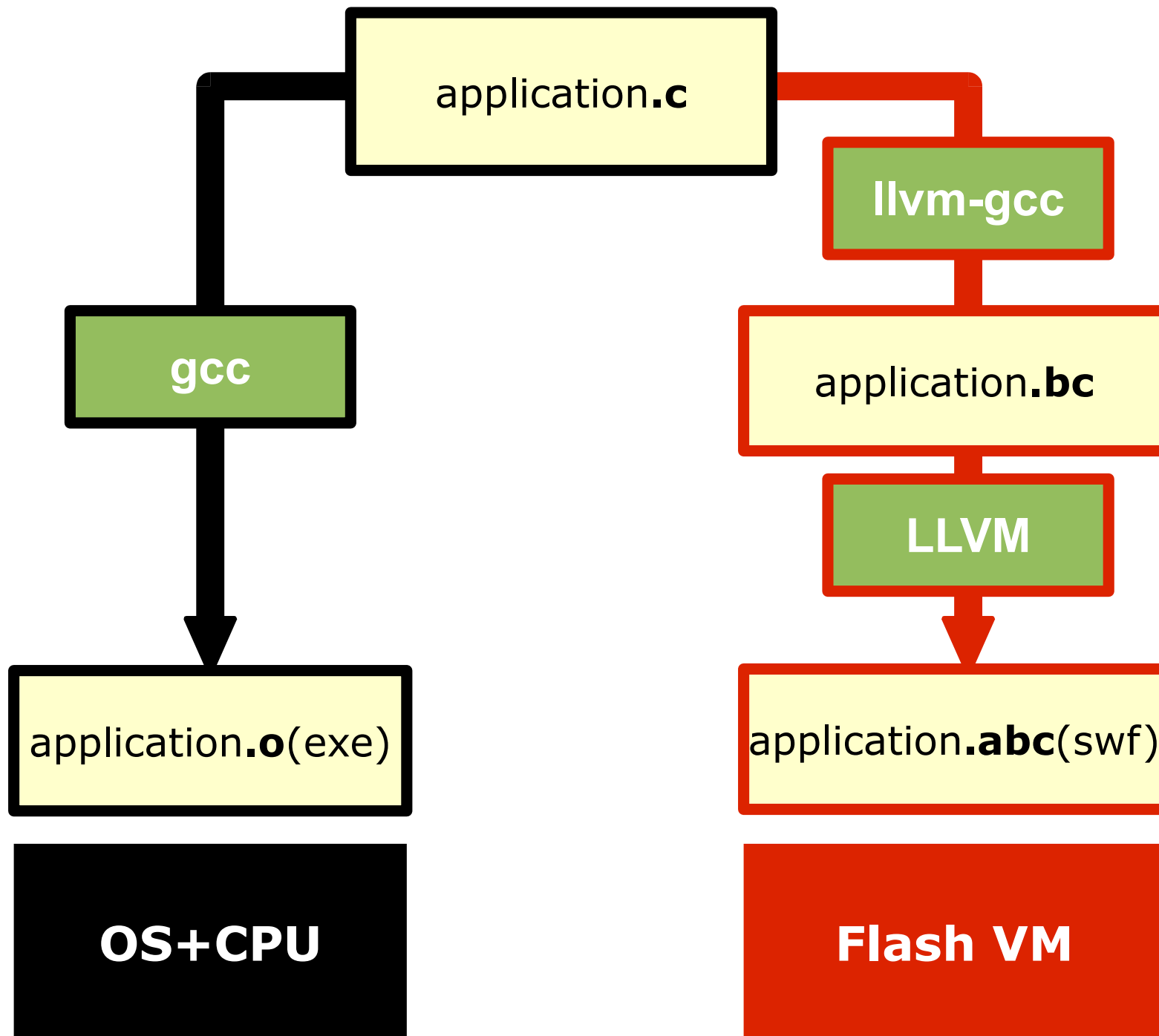


- 実験的な LLVM Backend
- LLVM -> ABC (ActionScript Bytecode)
 - 実物はなし
 - 資料はあり
 - “Flash C Compiler: Compiling C code to the Adobe Flash Virtual Machine”
http://llvm.org/devmtg/2008-08/Petersen_FlashCCompiler.pdf
 - Doom が動くらしい (Adobe MAX でデモ)
 - “Flash on C/C++ Sneak Peek”
<http://jp.youtube.com/watch?v=0hX-Uh3oTcE>
 - 割とさくさくうごいてびびる

(受け売りです)

Flash on C/C++ Sneak Peek





素朴な疑問

- メモリ/ポインタどうするの？
- スレッド/ブロッキングどうするの？
- システムコールどうするの？

メモリ = ByteArray

- バイト列 (std::vector<char>, ruby の String, ...)
- JIT の Intrinsic:
API 呼び出しが機械語に

スレッド = 有限状態マシン

Synchronous C / Asynchronous AS3

Synchronous C Method

```
int sum(int a, int b){  
    return a + b * 2;  
}
```

Asynchronous FSM Class

```
switch(state){  
    case 0:  
        i2 = i1 * 2; // b * 2  
        state++;  
        return;  
    case 1:  
        i3 = i2 + i0; // a + ...  
        state++;  
        return;  
    case 2:  
        result = i2; // return ...  
        gcurmachine = caller;  
        return;  
}
```


システムコール

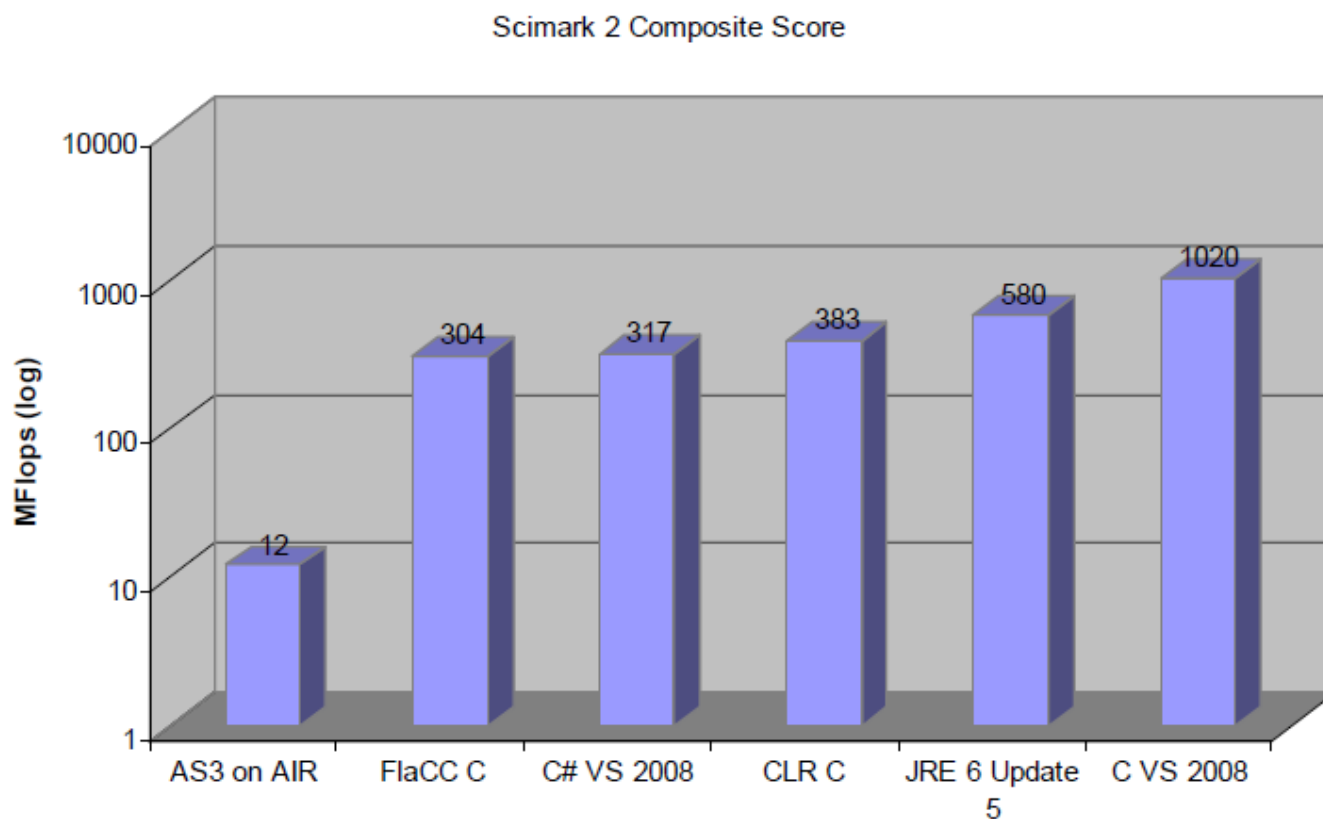
- ランタイム(Flash)を改造

だから公開デモがないのか...

- 標準ライブラリは BSD のソースをコンパイル
– バイトコードに！

ベンチマーク

Performance



FlashCC ここまで

Tamarin Tracing と TraceMonkey

Tamarin as JavaScript VM

苦勞が多い

- 静的型がないと遅い
- `eval()` がない (コンパイラがないから...)
- SpiderMonkey から乗り換えるの大変
 - Native API (DOM, XPCOM, ...)
 - バイトコード
 - GC (Write barrier, Deferred ref-counting)
 - ...

Tamarin Tracing

- ~~静的型がないと遅い~~
ムリヤリ速くするアルゴリズムの登場

Tamarin Tracing vs. JavaScript

```
function SquareDoubler() {  
    this.squarer = new Squarer();  
    this.make_double_square = function(x) {  
        return this.make_square(x) +  
            this.make_square(x);  
    };  
};
```

型わからず

```
var n = 0;  
var sd = new SquareDoubler();  
for (var i=0; i<arr.size; ++i) {  
    sd.make_double_square(arr[i]);  
}
```

要ハッシュ表検索

```
var i=0;
var n=0;
var sd = new SquareDoubler();
if (i instanceof int) {
    var iint:int = (int)i;
    if (arr.size instanceof int) {
        var sizeint:int = arr.size; // これはウソかも
        for (; iint < sizeint; i++) {
```

型チェック+確定

```
            if (sd.make_double_square == ...) {
                if (sd.squarer.make_square == ...) {
                    if (arr[iint] instanceof int) {
                        var xi:int = (int)arr[iint];
                        var xxxx:int = x*x + x*x;
                        if (n instanceof int) {
                            var nint:int = n;
                            n = nint + xxxxx;
                        } else { ... }
                    } else { ... }
                } else { ... }
            } else { ... }
        }
    }
}
```

インライン化

```

var i=0;
var n=0;
var sd = new SquareDoubler();
if (i instanceof int) {
    var iint:int = (int)i;
    if (sd.make_double_square == ...) {
        if (sd.squarer.make_square == ...) {
            if (arr.size instanceof int) {
                var sizeint:int = arr.size;
                if (n instanceof int) {
                    var nint:int = n;
                    for (; iint<sizeint; i++) {
                        if (arr[iint] instanceof int) {
                            var xi:int = (int)arr[iint];
                            var xxxx:int = x*x + x*x;
                            nint = nint + xxxxx;
                        }
                    }
                } else { ... }
            } else { ... }
        } else { ... }
    } else { ... }
}

```

チェックを外へ

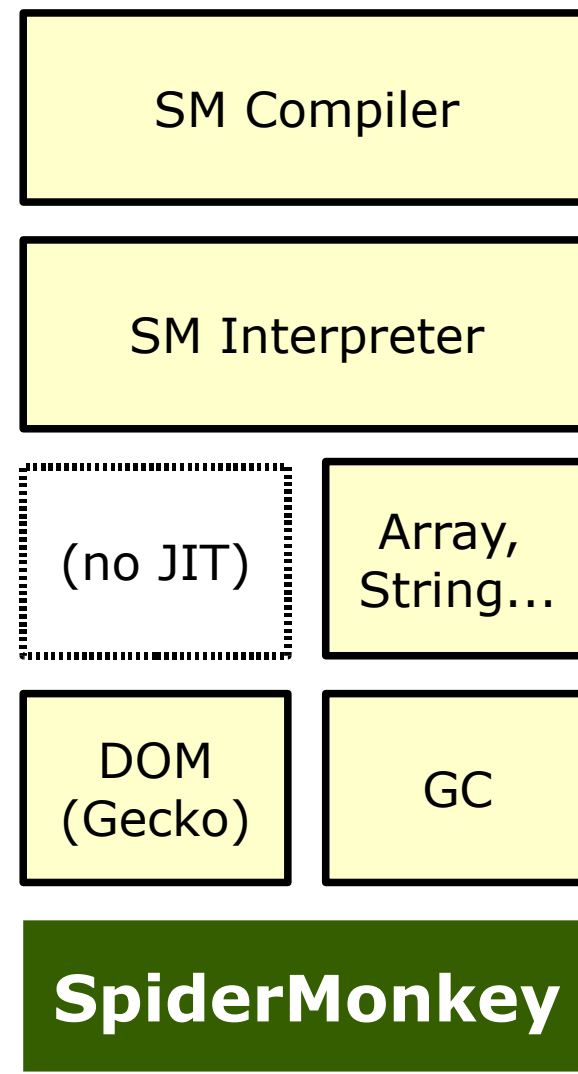
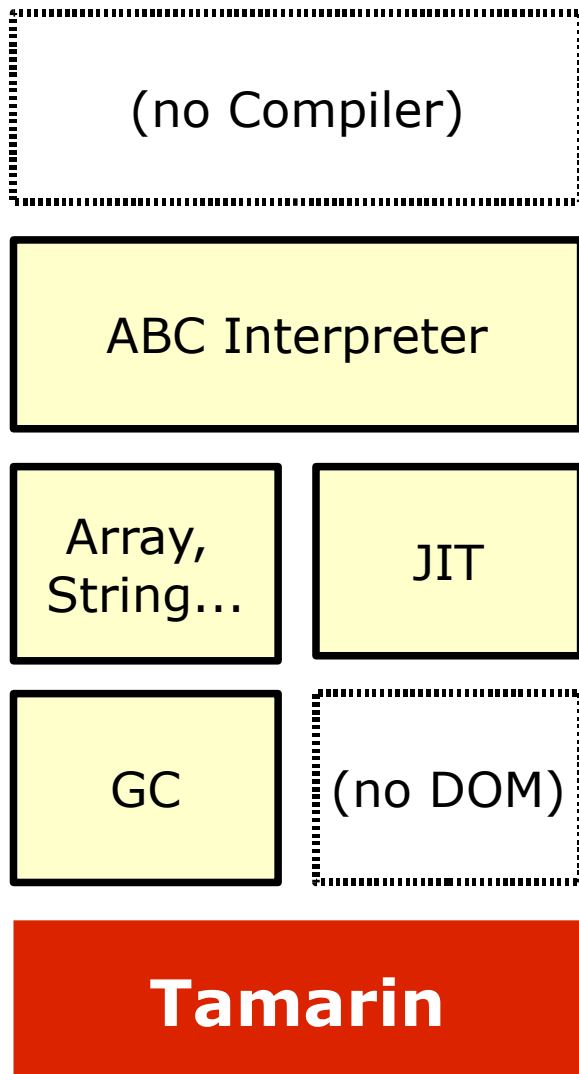
(たぶんまだ未実装)

Tamarin as JavaScript VM

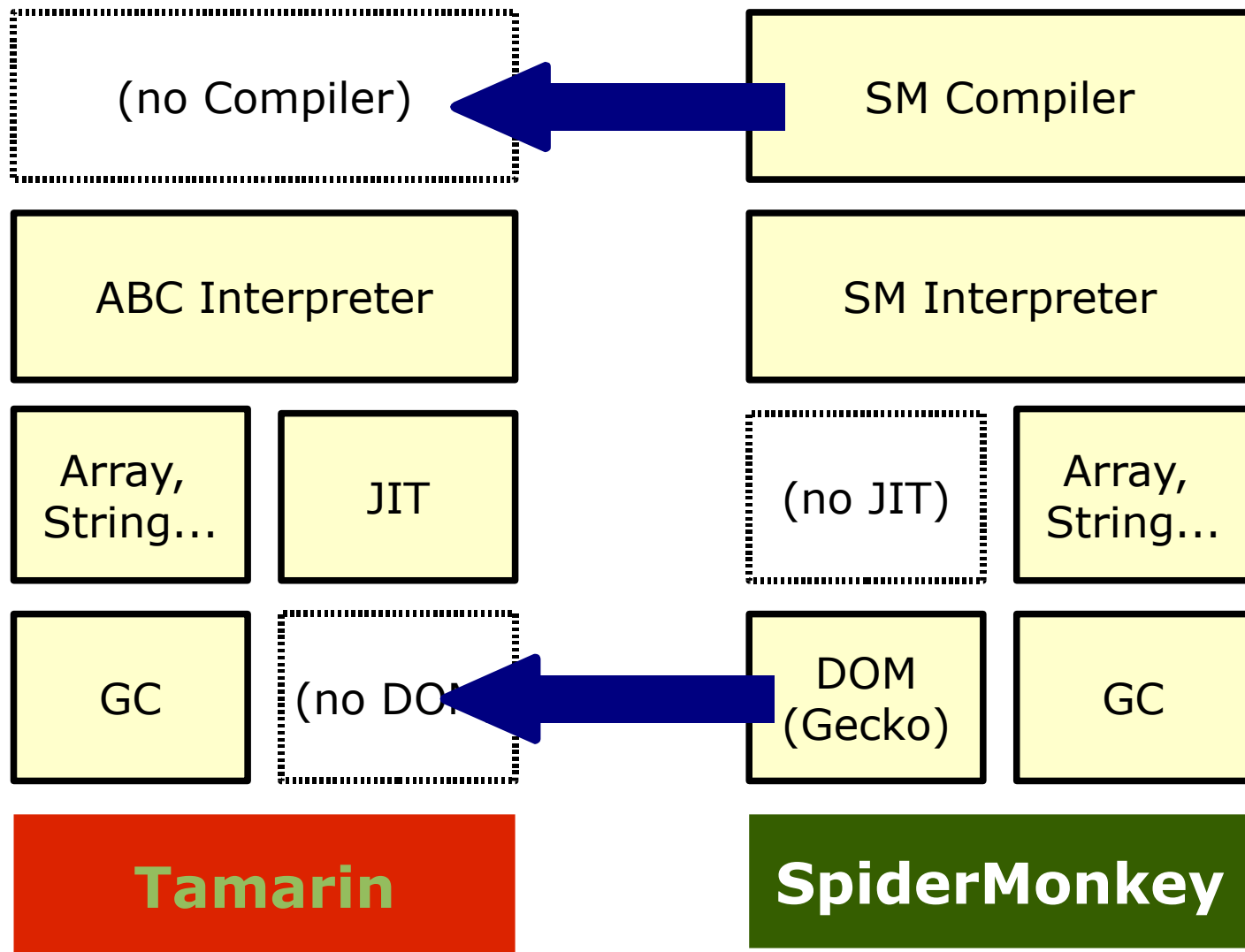
苦勞が多い

- ~~静的型がないと遅い~~
- `eval()` がない (コンパイラがないから...)
- SpiderMonkey から乗り換えるの大変
 - Native API (DOM, XPCOM, ...)
 - バイトコード
 - GC (Write barrier, Deferred ref-counting)
 - ...

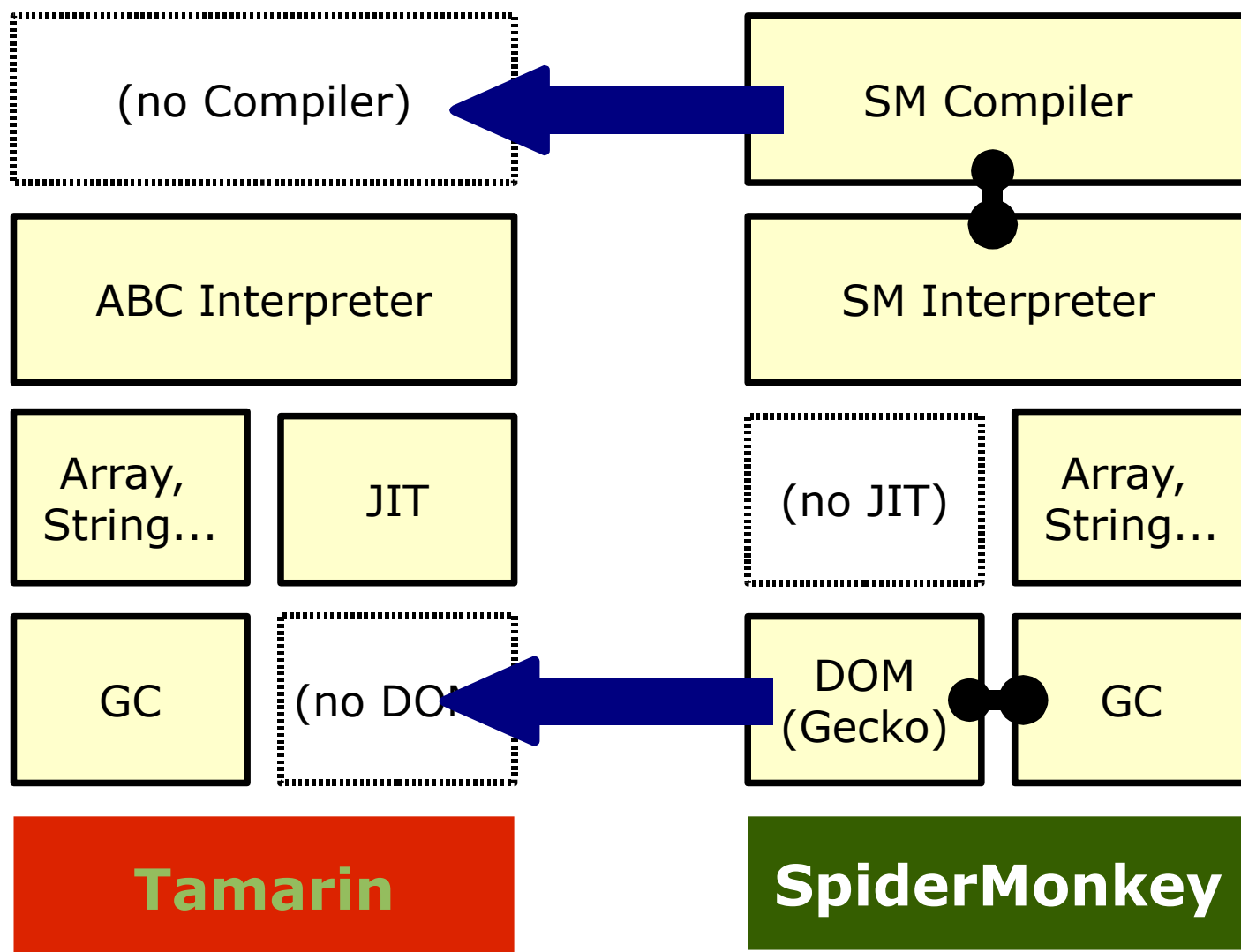
Tamarin vs. SpiderMonkey



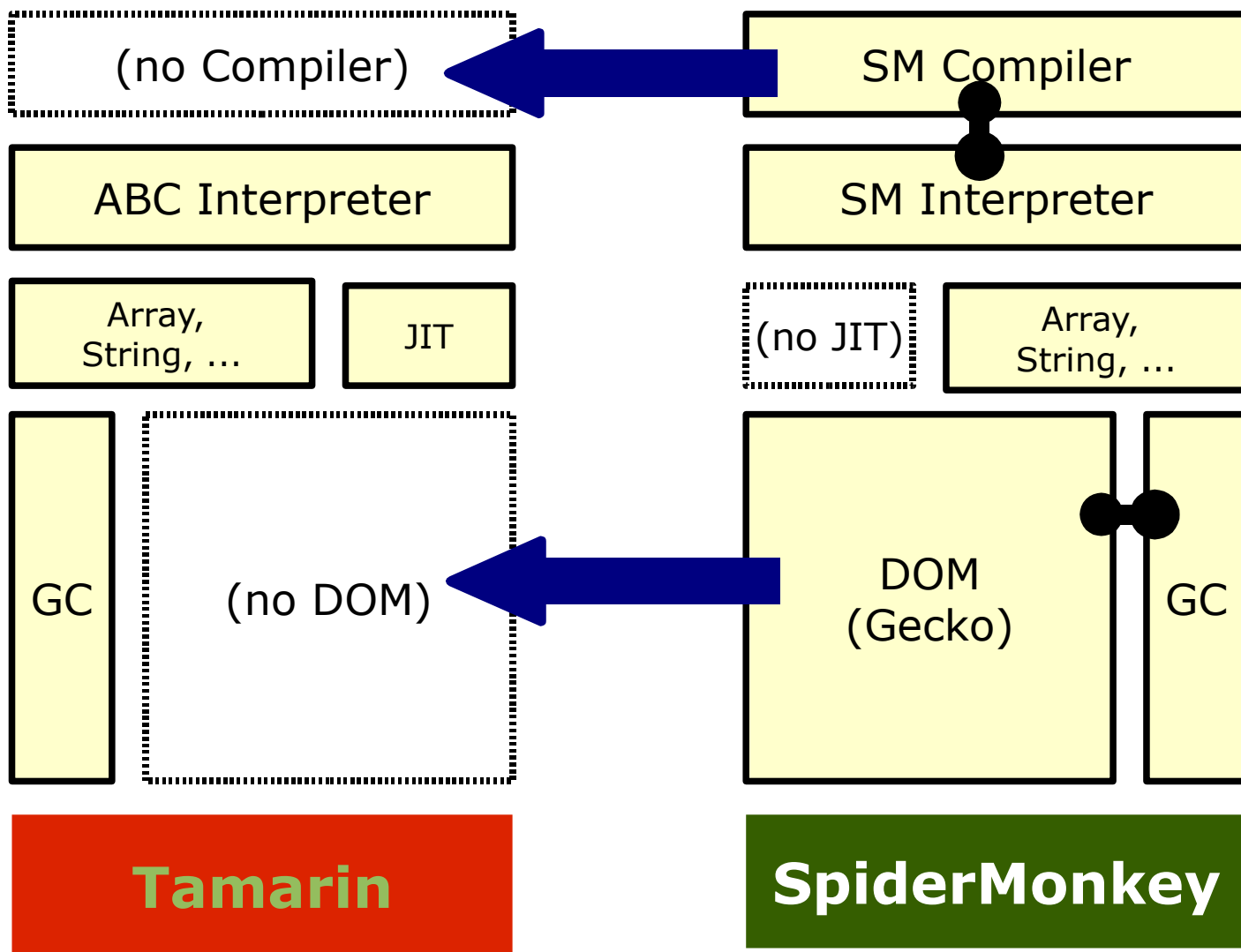
常考な Tamarin 移植



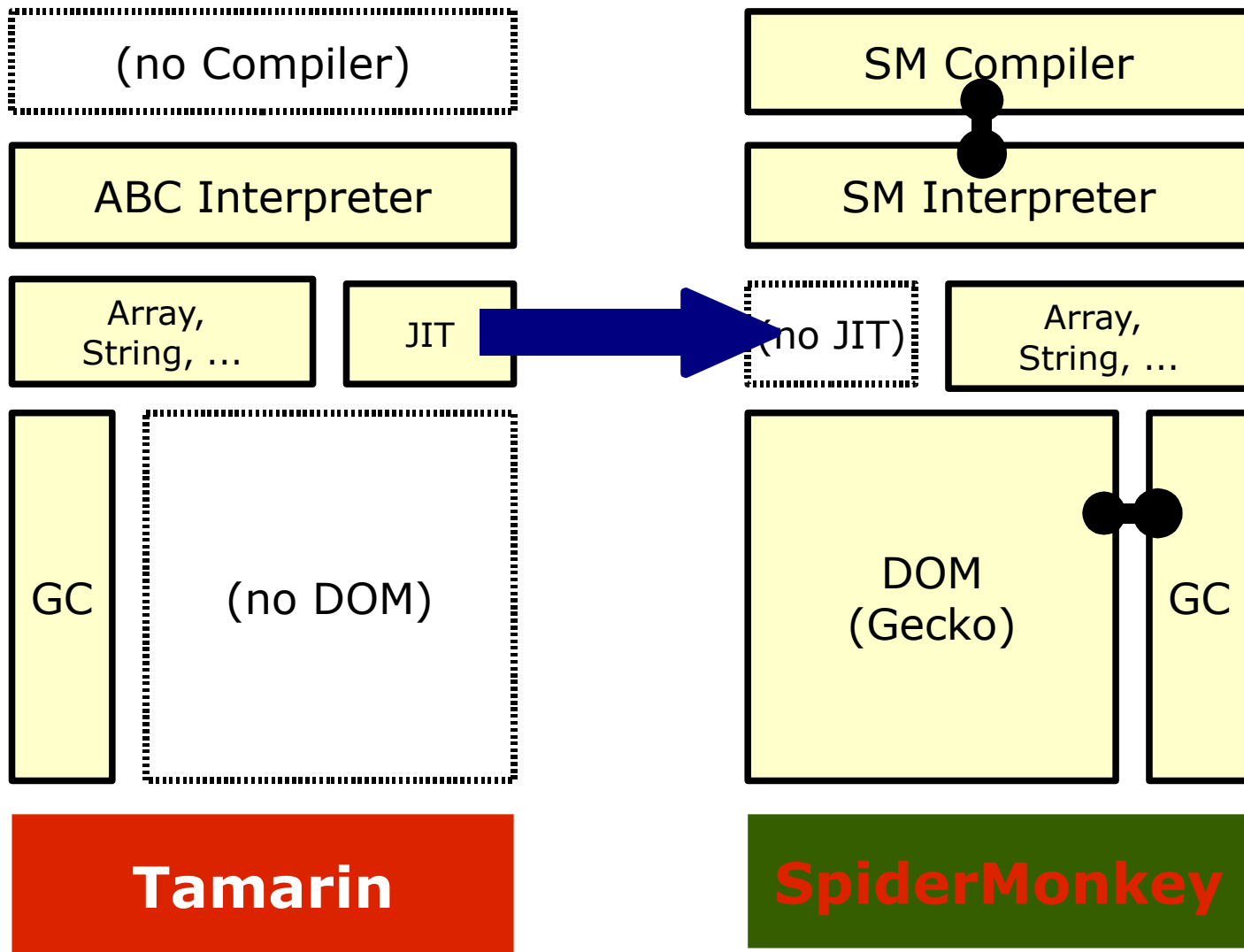
常考な Tamarin 移植: 依存の困難



常考な Tamarin 移植: サイズの困難

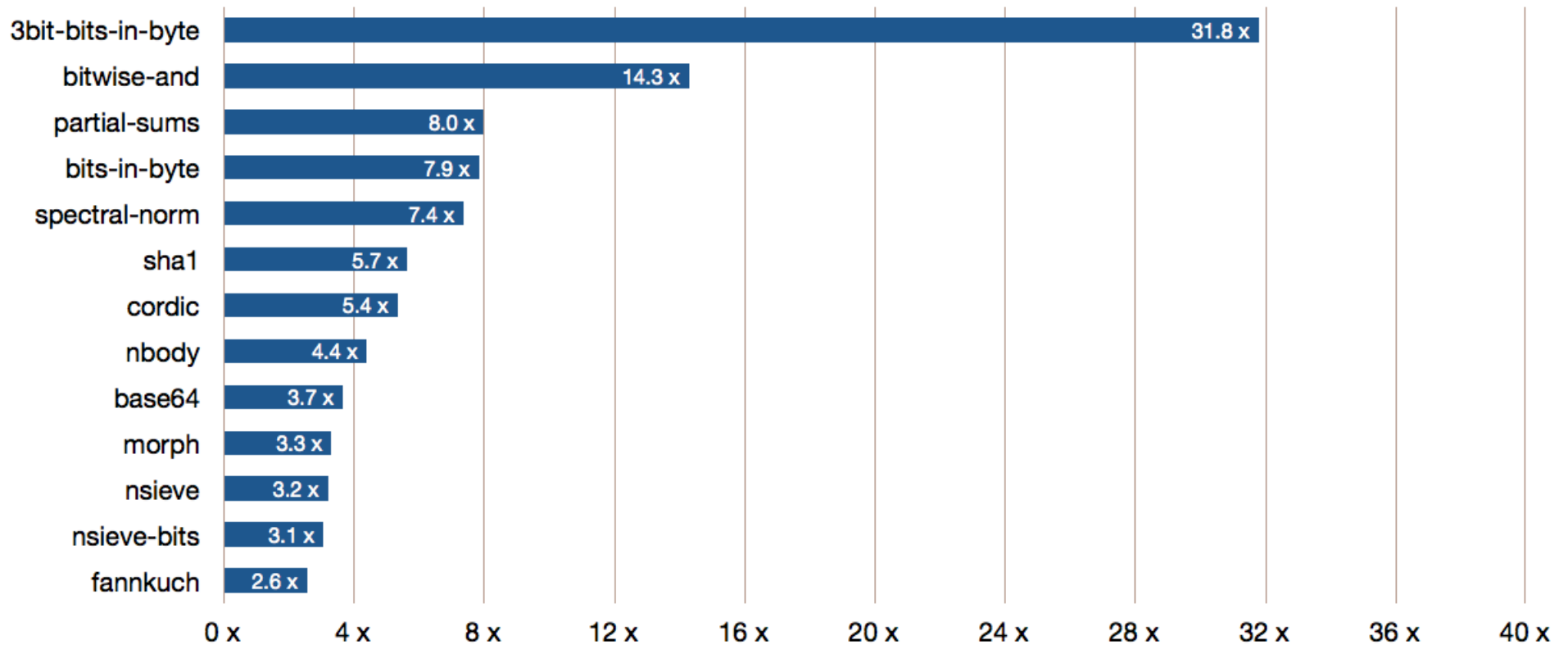


TraceMonkey



ベンチマーク結果

Firefox 3.1 with Tracing v.s. Firefox 3



むりやりまとめ

- JavaScript も速くなる！
- C++ プログラマの将来やいかに！

スライドここまで