Monthly Research

Intel Memory Protection Extensionsとその活用

株式会社ＦＦＲＩ
http://www.ffri.jp

Ver 1.00.01

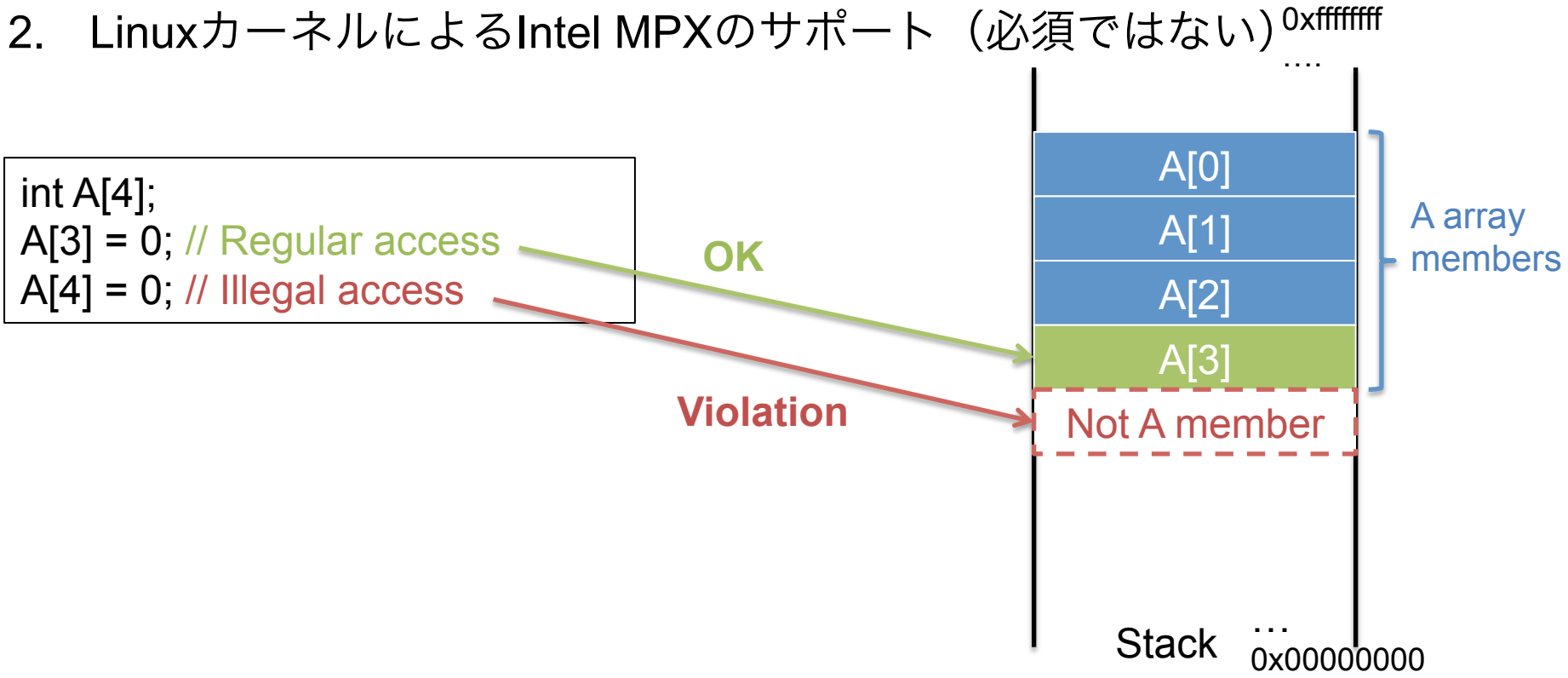
Intel® Memory Protection Extensions(Intel MPX)とは

- x86, x86-64の命令拡張
  - メモリアクセスの境界チェックを支援する命令とレジスタ

- 2015年2月現在、この機能を搭載したCPUは出荷されていない
  - skylake 世代（2015年出荷予定）で利用可能に
  - Intel Software Development Emulatorで動作を確認できる

Intel MPX利用の実例：Code Instrumentationによるバッファオーバーフローの検知

1. GCCコンパイラのcode instrumentationによるメモリ境界チェックコードの挿入
2. LinuxカーネルによるIntel MPXのサポート（必須ではない）



Intel Memory Protection Extensions (Intel MPX)

Intel MPXの概要

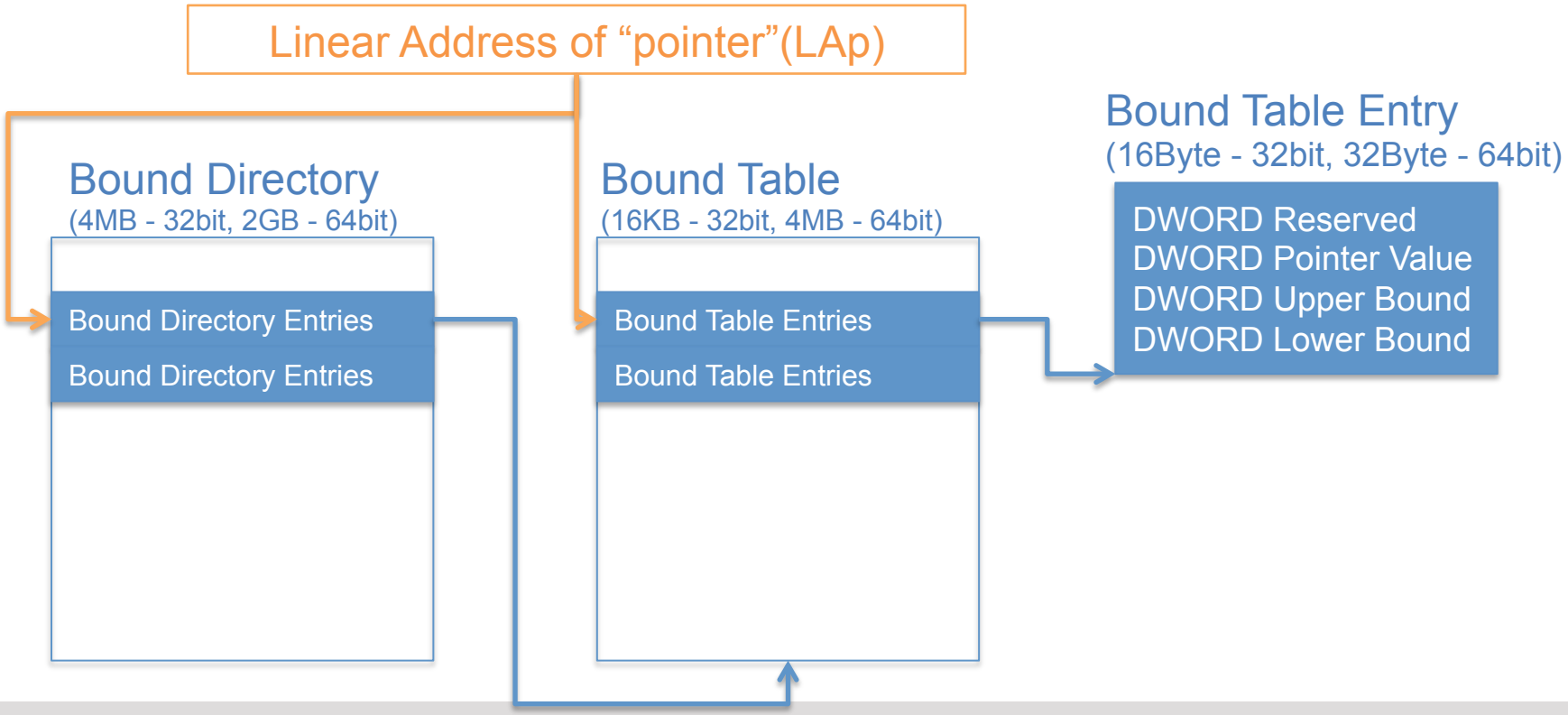
- レジスタとそれに関するCPU命令が追加された
  - プレフィックスが0fなので、mpxが利用できないCPUではnop扱い
- メモリ境界とポインタを対応づけるbound paging機構が追加
- CALL, RET, JMP Jcc命令実行時にboundsレジスタを待避&復元する設定が利用可能になった
- VMX, TSX命令が拡張された

New Registers

- BND0-3 Bounds registers
  - 128bit長のレジスタ
  - 64bitずつ、メモリの上限と下限を設定する
- BNDCFGU
  - Bound Pagingのベース（Ring3）
- BNDCFGS
  - Bound Pagingのベース（Ring0-2）
- BNDSTATUS
  - Bound Pagingの操作アドレスやエラーコードが入る

Bound Paging

- ポインタの物理アドレスをindexとして、ポインタとBoundsのマッピングを管理する



BND命令

- BNDMK
  - boundsレジスタに境界をセット
- BNDCL
  - boundsレジスタを指定し、レジスタまたはメモリアドレスがboundsの下限より上かどうかチェック
- BNDCU, BNDCN
  - boundsレジスタを指定し、レジスタまたはメモリアドレスがboundsの上限より下かどうかチェック
- BNDMOV
  - boundsレジスタの読み書き、メモリへの読み書き
- BNDLDX
  - メモリからboundsレジスタに読み込み
- BNDSTX
  - boundsレジスタのメモリへの待避

利用イメージ(1)

```
//スタックベースから配列が確保されていると仮定
; int A[100] from RBP

// RAXに配列のベースアドレスを格納
LEA RAX, [RBP+offset]

// BND0 レジスタのLower bound(0-63bit)にRAXのアドレスを格納
// 同じくupper bound(127-64bit)に(RAX+399)の値を設定
BNDMK BND0, [RAX+399]
```

利用イメージ(2)

```
//32bit integerなRBXの値を、RAXの示すアドレス（バッファ）に格納するとき
//バッファの境界はBND0に設定済みとする

BNDCL BND0, [RAX] ;下限のチェック
BNDCU BND0, [RAX+3] ;上限のチェック
MOV Dword ptr [RAX], RBX;
```

もしBNDCLやBNDCUで境界を越えた場合、BNDSTATUSにエラーコードを設定し、例外#BRを発生させる

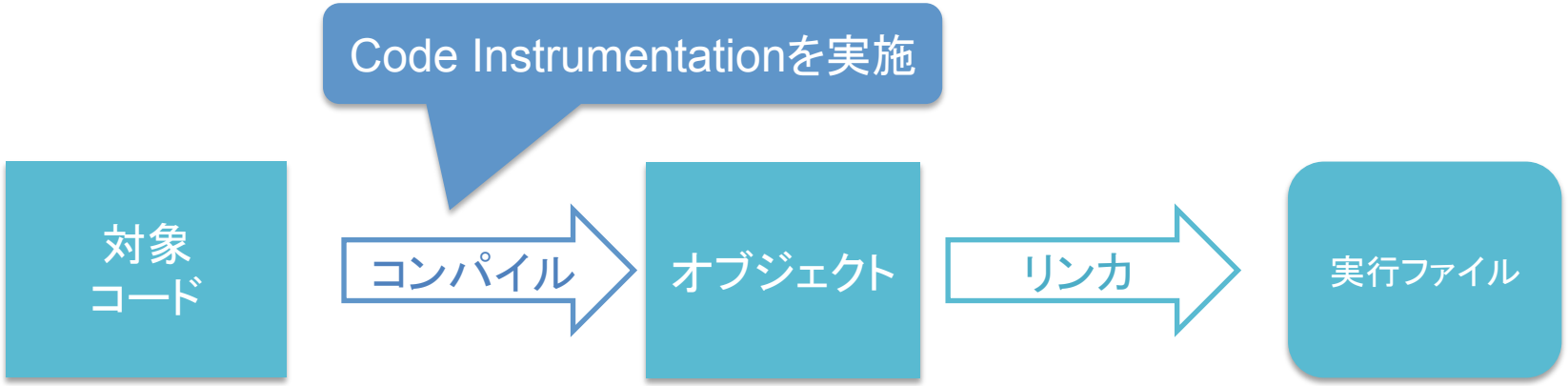
BR例外とError Code

- 例外発生時、BNDSTATUSレジスタの0-1bitにエラーコードが格納される
  - 00b – NO Intel MPX exception
  - 01b – Bounds violation
    - BNDCL, BNDLU, BNDCN命令にて発生する
  - 10b – Invalid bound directory entry
    - BNDLDX BNDSTX命令にて発生する
    - 不正なBound directory entryの物理アドレスをBNDSTATUSレジスタの127-2bitに格納する
  - 11b - reserved

Intel MPX Support in GCC Compiler

GCCコンパイラによるIntel MPX利用

- バッファーオーバーフローなど、連続して確保したメモリ領域を超えるメモリアクセスを検知するためのcode instrumentationを行う
  - -f mpxオプションで有効に



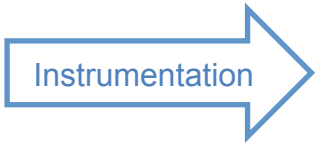
GCCが-f mpxでおこなうCode Instrumentation

- メモリ境界の計算
  - コンパイル時に静的解析し、変数、ポインタごとにメモリ境界設定コードを生成
- メモリアクセスごとのチェックコード生成
- 関数呼び出し、終了時のboundsレジスタ初期化、復帰など
- Bounds tableの管理
  - ポインタと境界の対応付けを管理するBounds tableの管理は、基本的にランタイムが行う
- ネストした構造体などにおける境界の圧縮

Memory accesses instrumentationの例

- p[i]=0;のGIMPLE中間表現のダンプ例

p[i] = 0;




```
<bb 2>:
  _2 = (long unsigned int) i_1(D);
  _3 = _2 * 4;
  _6 = p_4(D) + _3;
  __builtin_ia32_bndcl (__bounds_of_p_5, _6);
  _8 = _6 + 3;
  __builtin_ia32_bndcu (__bounds_of_p_5, _8);
  *_6 = 0;
```

Reference：
https://gcc.gnu.org/wiki/Intel%20MPX%20support%20in%20the%20GCC%20compiler

Intel MPX in Linux Kernel

Linux kernel におけるIntel MPXサポート

- Linux 3.19でIntel MPXのサポート機能がマージされた
  - カーネルによるBounds Tableの動的アローケーション
  - MPXによる例外に関するsignalの追加、変更
  - Bound Tableのクリーンアップ
  - prctlへのコマンド追加
    - カーネルによるBounds Table管理を有効/無効にする

カーネルによるBound Tableの動的アローケーション

- Bounds Tableは最大2Gの物理アドレスにマッピングしなければならない
  - 現実的ではない
  - このため、カーネルがBNDSTXの例外をキャッチして、動的にBound Tableを拡張する
  - VM_MPXフラグがついた仮想メモリをカーネルが常にトレースする

- また、do_munmap()をフックし、Bounds Tableがfreeされたら物理アドレスの予約状況を変更する

siginfo構造体の拡張

- _sigfault構造体に、bounds違反に関するフィールドを追加

```
/* SIGILL, SIGFPE, SIGSEGV, SIGBUS */
struct {
        void __user *_addr; /* faulting insn/memory ref. */
        #ifdef __ARCH_SI_TRAPNO
        int _trapno;        /* TRAP # which caused the signal */
        #endif
        short _addr_lsb; /* LSB of the reported address */
        struct {
                void __user *_lower;
                void __user *_upper;
        } _addr_bnd;
} _sigfault;
```

in include/uapi/asm-generic/siginfo.h in Linux kernel source (3.19)

まとめ

- 拡張命令自体はプリミティブだが、コンパイラと組み合わせる前提で、ランタイムによるメモリ保護を行うために十分な機能が備わっている

- パフォーマンスへの影響はまだはっきりと分からない
  - しかし、software fault isolationやruntime code instrumentationよりは早いことは確実

- 新しいハードウェアが必要なため、すぐに普及する技術ではない
  - しかし今後、主要コンパイラや言語ランタイムなどで使われていく可能性は大いにある

参考文献

- Intel® Architecture Instruction Set Extensions Programming Reference Section 9 Version 319433-022 OCTOBER 2014
- Intel® Memory Protection Extensions (Intel® MPX) support in the GCC compiler https://gcc.gnu.org/wiki/Intel%20MPX%20support%20in%20the%20GCC%20compiler
- Documentations/x86/intel-mpx.txt in Linux kernel 3.19

Contact Information

E-Mail : research—feedback@ffri.jp
Twitter : @FFRI_Research