このセッションには音声が入っています。
スピーカーの音量に注意してください。

スライドの画面をクリックするか、
プレイヤーの再生ボタンを押すとセッションが始まります。

音量に注意してください。調整が終わったら、スタートしてください。

このスライドに音声はありません

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

インテル® Xeon Phi™ コプロセッサー Code Modernization セミナー

概要 (約 33 分)

2015 年 10 月

内容

- Knights Corner † コアと ISA
- ソフトウェアとエコシステム
- プログラミング・モデル
  - ネイティブ
  - オフロード
  - シンメトリック
- 正しいモデルを選択
- 将来のプラットフォーム

† 開発コード名

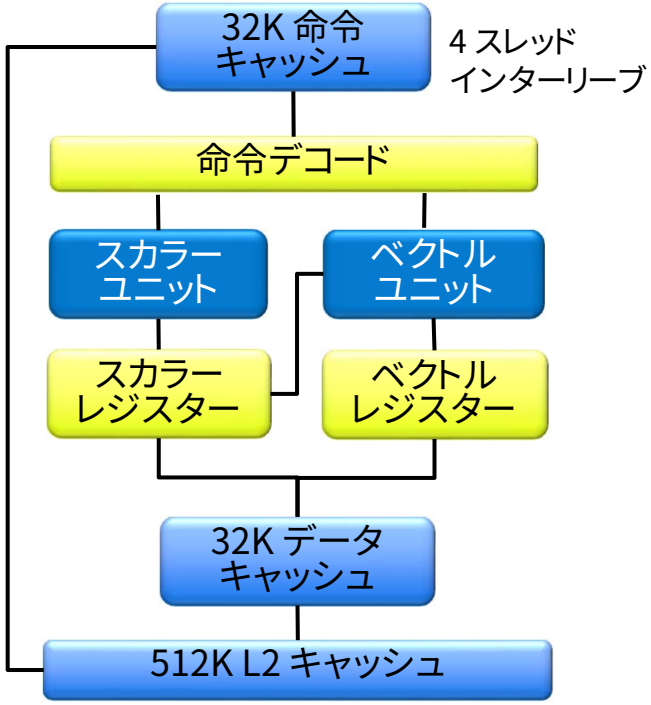
©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

Knights Corner † コアと ISA

プログラマーの観点から見たインテル® MIC アーキテクチャー

† 開発コード名

コア



インテル® Pentium® プロセッサーの
スカラー命令セット (x87)

完全な機能

インオーダー操作

64 ビット・アドレッシング

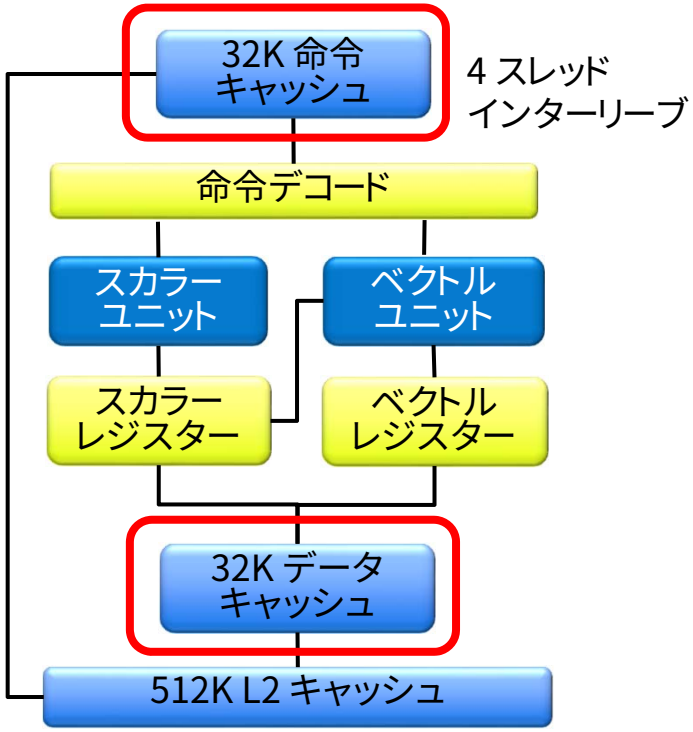
4 ハードウェア・スレッド / コア

2 つのパイプライン

- スカラー
- ベクトル / スカラー

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

コア



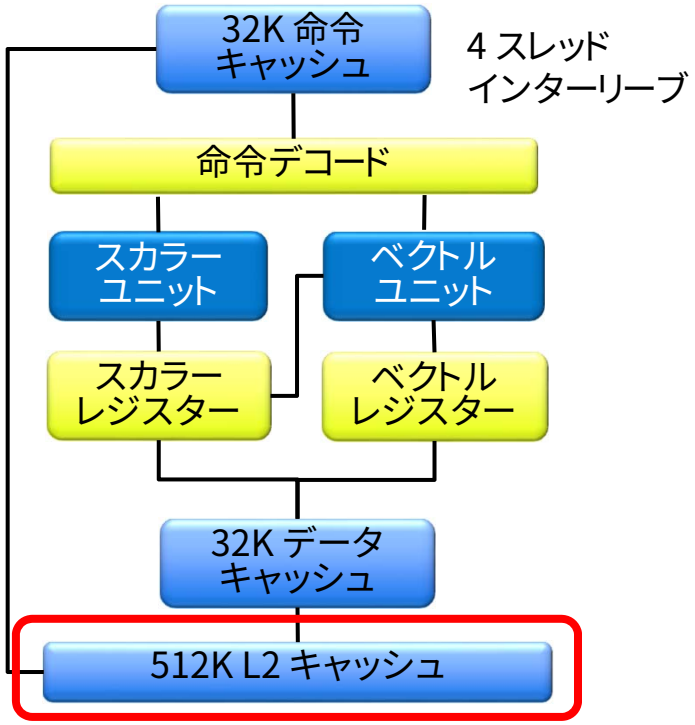
L1 キャッシュ

- コアあたり 32K の命令キャッシュ
- コアあたり 32K のデータキャッシュ
- 8 ウェイ・アソシアティブ
- 64 バイトのキャッシュライン
- 3 サイクルのアクセス・レイテンシー

- 完全な一貫性 (MESI)

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

コア



L2 キャッシュ

- コアあたり 512K のユニファイド・キャッシュ
- 8 ウェイ・アソシアティブ
- インクルーシブ
- 62 コア全体で 31M バイト
- 11 サイクルのアクセス・レイテンシー
- 最大 32 個の処理要求
- ストリーミング HW プリフェッチャー
- 完全な一貫性

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ISA / レジスター

標準インテル® 64 レジスター(EM64T)

rax	r8
rbx	r9
rcx	r10
rdx	r11
rsi	r12
rdx	r13
rsp	r14
rbp	r15

+ 32 512 ビット SIMD レジスター

zmm0

…

zmm31

+ 8 マスクレジスター (16 ビット幅)

k0 (特殊レジスター、利用しない)

…

k7

x87 で提供されていた xmm (インテル® SSE/128 ビット) や ymm (インテル® AVX/256 ビット) レジスターはない

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

インテル® Xeon® プロセッサーとインテル® Xeon Phi™ コプロセッサーの ISA/レジスター

SSE/AVX/AVX-512

511 … 256	256 … 128	127 … 0
ZMM0	YMM0	XMM0
ZMM1	YMM1	XMM1
…	…	…
ZMM15	YMM15	XMM15
ZMM16	YMM16	XMM16
…	…	…
ZMM31	YMM31	XMM31

IMCI

511 … 0
ZMM0
ZMM1
…
ZMM31

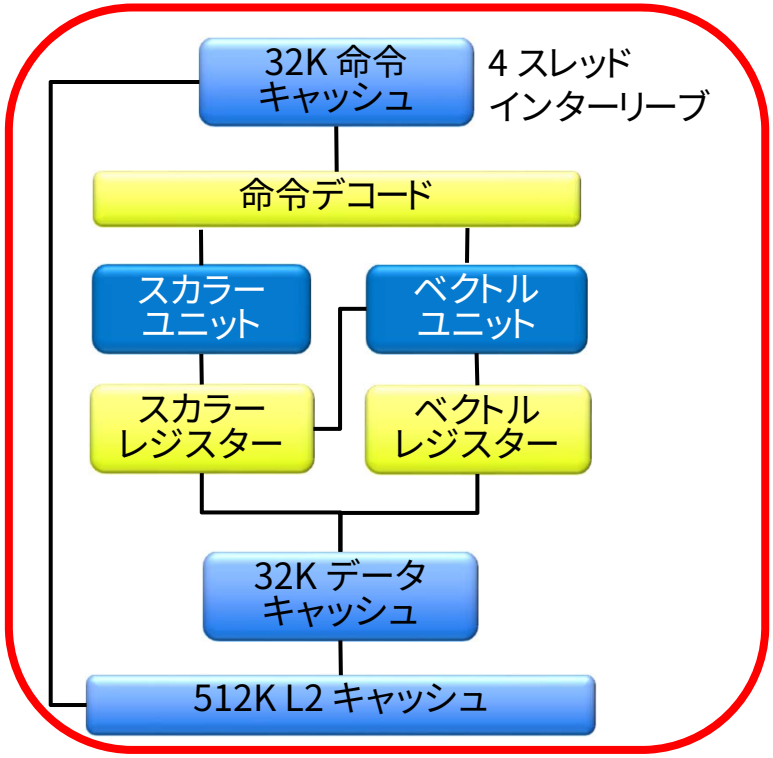
15 … 0
K0
…
K7

マスクレジスター

標準インテル® 64 レジスター (EM64T)

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

コア



パフォーマンス・モニタリング・ユニット

- 4 つのイベント選択レジスター
- 4 パフォーマンス・カウンター
- 4 HW スレッド間で共有
- RDMSR/WRMSR 命令を使用したモデル固有レジスター (MSR)を介してプログラム

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ベクトル命令

ベクトル命令のフォーマット

明示的なディスティネーション・レジスターを持つ 3 オペランド形式

命令　　デスティネーション、ソース1、ソース2

→ ソース・レジスターは破壊されません

→ 非常にコンパクトなコード

(ほとんどの) MIC 命令はマスクすることができます

命令　　デスティネーション[マスク]、ソース1、ソース2

→ マスクの結果は非破壊的です。つまり、ディスティネーション値は保持されます

例:

vaddps zmm1{k1},zmm2,zmm3

デスティネーション　マスク　ソース1　ソース2

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ベクトル処理ユニットと ISA

VPU データ型

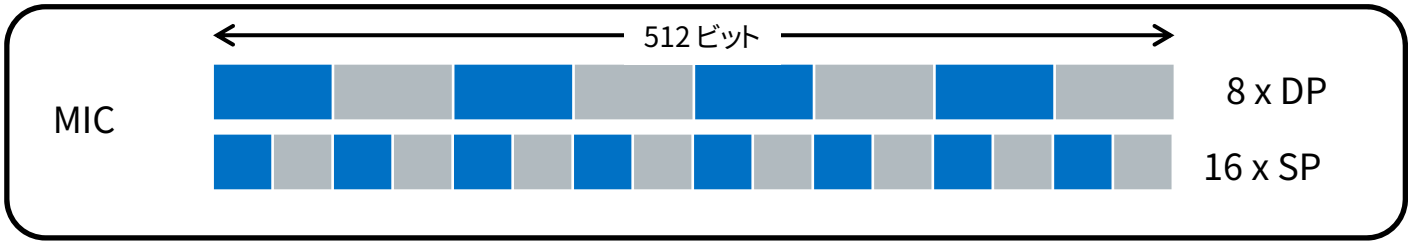
パックド32 ビット整数 (もしくは dword)

パックド32 ビット単精度浮動小数点値 (SP FP)

パックド64 ビット整数 (もしくは qword)

パックド64 ビット倍精度浮動小数点値 (DP FP)

(他のデータ型は、ロードとストアの際にアップ/ダウン変換後に処理されることがあります)

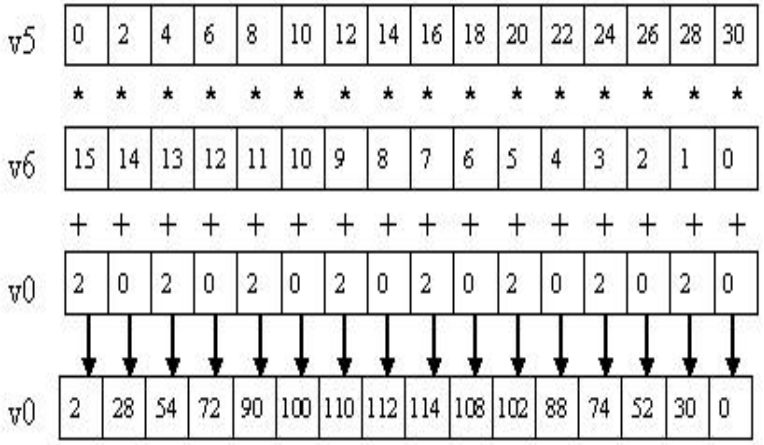


©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

Fused Multiply Add (乗加算)

Multiply-Add (デスティネーションは、最初のソース)

- vfmadd231ps zmm0, zmm5, zmm6 ; zmm0 = zmm5 * zmm6 + zmm0
- オペランド 2 にオペランド 3 を掛けて、オペランド 1 に加算



©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ベクトル命令 - インテルの組み込み関数ガイド

intel Intrinsics Guide

The Intel Intrinsics Guide is an interactive reference tool for Intel intrinsic instructions, which are C style functions that provide access to many Intel instructions - including Intel® SSE, AVX, and more - without the need to write assembly code.

ISA で
フィルター

Technologies
- MMX
- SSE
- SSE2
- SSE3
- SSSE3
- SSE4.1
- SSE4.2
- AVX
- AVX2
- FMA
- AVX-512
- KNC
- SVML
- Other

機能で
フィルター

Categories
- Application-Targeted
- Arithmetic
- Bit Manipulation
- Cast
- Compare
- Convert
- Cryptography
- Elementary Math Functions
- General Support

_mm_search ?

```
__m512d _mm512_abs_pd (__m512d v2)                                   vpandq
```

Synopsis

```
__m512d _mm512_abs_pd (__m512d v2)
#include "zmmintrin.h"
Instruction: vpandq zmm {k}, zmm, m512
CPUID Flag : KNCNI
```

Description

Finds the absolute value of each packed double-precision (64-bit) floating-point element in v2, storing the results in dst.

Operation

```
FOR j := 0 to 7
        i := j*64
        dst[i+63:i] := ABS(v2[i+63:i])
ENDFOR
dst[MAX:512] := 0
```

詳細は、任意の組み込み関数を
展開して表示

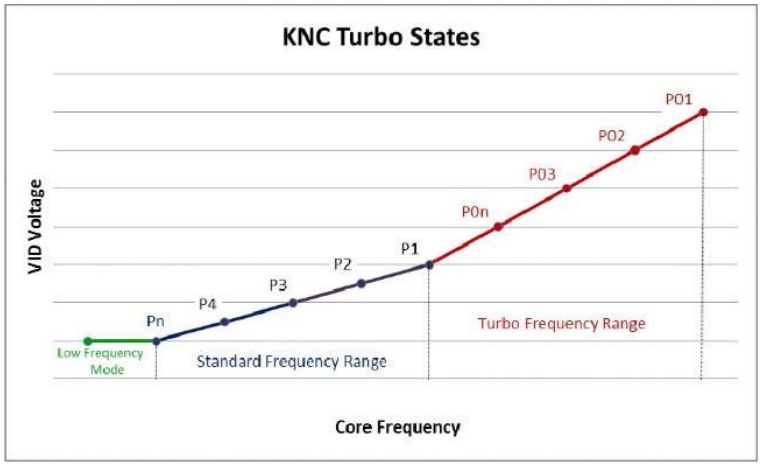
```
__m512d _mm512_mask_abs_pd (__m512d src, __mmask8 k, __m512d v2)     vpandq
__m512 _mm512_abs_ps (__m512 v2)                                     vpandd
__m512 _mm512_mask_abs_ps (__m512 src, __mmask16 k, __m512 v2)       vpandd
__m512i _mm512_adc_epi32 (__m512i v2, __mmask16 k2, __m512i v3, __mmask16 * k2_res)   vpadcd
__m512i _mm512_mask_adc_epi32 (__m512i v2, __mmask16 k1, __mmask16 k2, __m512i v3, __mmask16 * k2_res)   vpadcd
```

ここ入手できます: http://software.intel.com/sites/landingpage/IntrinsicsGuide/

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

P-ステートとターボ

- 7120P と 7120X SKU がターボモードをサポートします
- P-ステートとターボ動作は、インテル® Xeon® プロセッサーと同一です
- 異なる点: コプロセッサー OS は、P1 から Pn へ状態を要求/管理します



- ターボが有効の場合、コプロセッサー OS は利用率に基づいて HZ を選択

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ソフトウェアとエコシステム

概要

メニーコア・プラットフォーム・ソフトウェア・スタック (MPSS)

2014 Intel Corporation

ソフトウェア・アーキテクチャー

コプロセッサー・ソフトウェア・アーキテクチャーのコンポーネント

インテル® メニーコア・プラットフォーム・ソフトウェア・スタック (MPSS)

- KNC† デバイス向けの Linux* マイクロ OS
- TCP/UDP、IP、ソケットなどをサポート
- シンメトリック通信インターフェイス (SCIF)

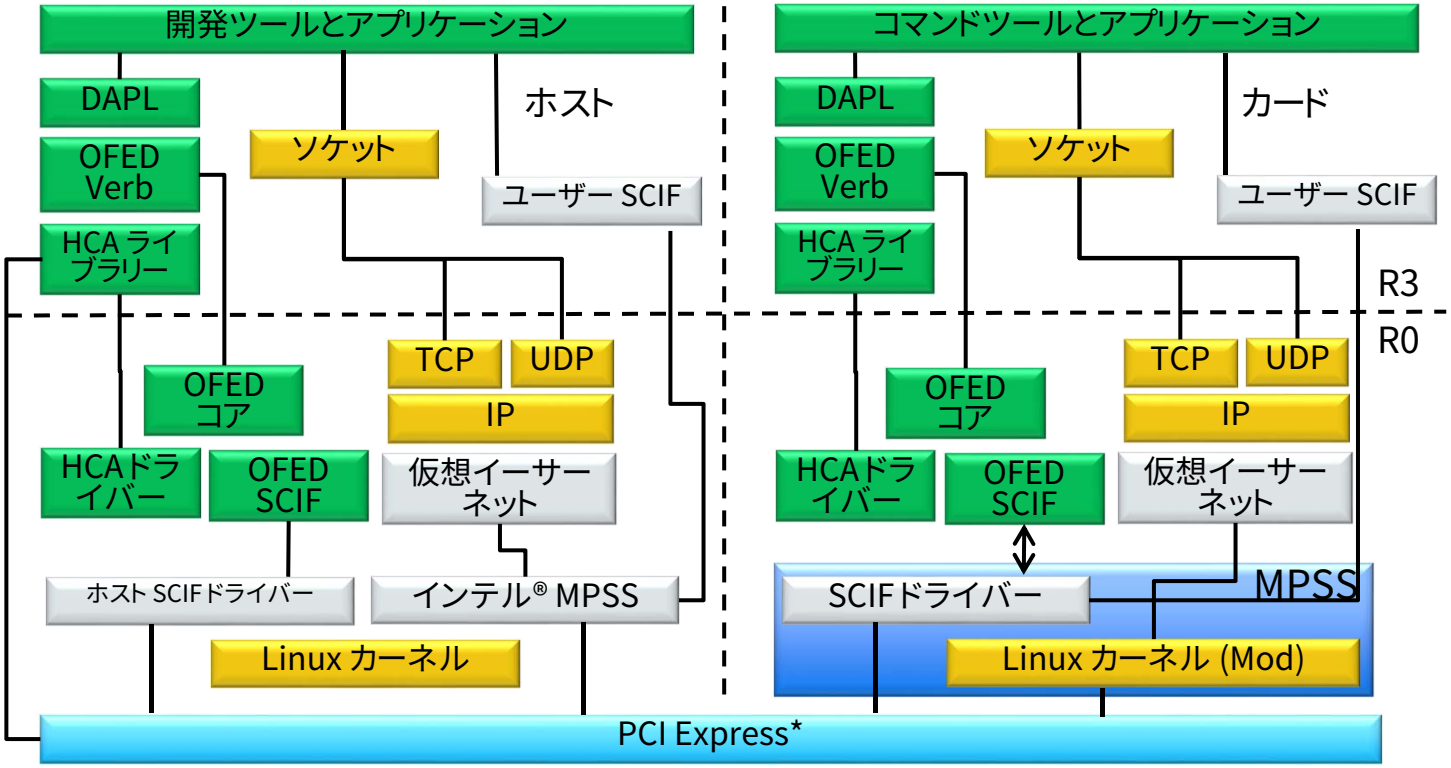
開発ツール

- インテル® Fortran & C++ コンパイラー
  - OpenMP*、インテル® Cilk™ Plus、スレッディング・ビルディング・ブロック
- インテル® デバッガー
- インテル® MPI
- インテル® ライブラリー (インテル® MKL など)
- インテル® VTune™ Amplifier XE パフォーマンス解析

† 開発コード名

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ソフトウェア・アーキテクチャー



©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ソフトウェア・アーキテクチャー

- Linux* "uOS"
- 標準カーネル (kernel.org) を基にする
- MIC 向けに最小限の組み込み Linux 環境を移植
- 可能な限り独自の変更を避ける
- GPL を適用
- ローダブルなカーネルモジュールを拡張可能 (SEP サンプリング・コレクターなど)
- Busybox シェル環境
- Linux 標準ベース (LSB) のコアライブラリー: glibc、libstdc++、libgcc_s、libz、libcurses、libpam

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

BusyBox*

- 単一の実行形式ファイルにバンドルされるユーティリティー集
  - 異なるコマンドに見せかける argv[0] を使用したトリック
- "ls、cp、df、mv、tar、mount、grep、dmesg、..." などの UNIX コマンド群を含む
  - dmesg は、特定のカーネルメッセージなどをコンソールに表示
- 小さなメモリー・フットプリント: OS が RAM から起動され、その他のストレージが利用できない場合に役立つ
- 高度に設定およびカスタマイズ可能
  - 必要に応じてコマンドを追加/削除
- イーサーネット・ブリッジが可能
- オープンソース

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ソフトウェア・アーキテクチャー

SCIF

- ホスト - KNC† 間の通信バックボーン
- 単一プラットフォーム (ノード) 内で、com や cap を提供
- 低レイテンシー、低オーバーヘッドな通信
- ホストの PCI Express* システムバス全体で通信を行う共通の API を提供
- ダイレクトな高帯域転送向けの DMA 機能を提供
- 完全な可視性 (/usr/include/scif.h)

† 開発コード名

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ソフトウェア・アーキテクチャー

ホスト側のインテル® MIC 関連コマンド

- micflash - Knights Corner PCI Express* カード上のフラッシュをアップデートする
  コマンド・ユーティリティー
- micinfo - カードの物理的な設定やドライバーのバージョンなどのカードのパラメーターを
  表示するユーティリティー
- micsmc - 別名 " コントロールパネル" は、カードの温度、電力、適用されているパラメーターを
  表示します。例えば、コア温度、コアの利用率、メモリーの利用率など (API が付属)

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

プログラミング・モデル

ネイティブ

オフロード

シンメトリック (対称)

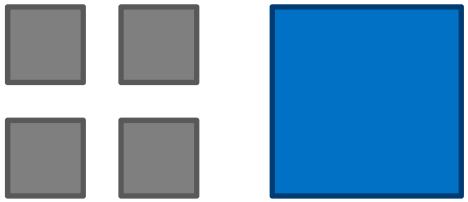
モデルを選択

将来性のあるモデル

2014 Intel Corporation

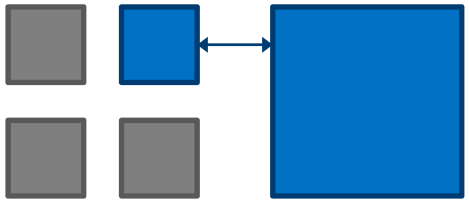
コプロセッサー・プログラミング・モデル

ネイティブ



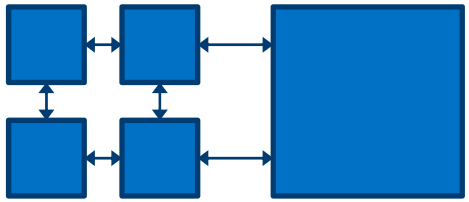
- ターゲットコード:
  高度な並列化 (スレッド化とベクトル化) のスループット
- ボトルネックの可能性:
  シリアル / スカラーコード

オフロード



- ターゲットコード:
  大部分がシリアルで、ホットな並列領域を含む
- ボトルネックの可能性:
  PCIe* データ転送

シンメトリック (対称)



- ターゲットコード:
  高度に並列化され、両方のプラットフォーム上で適切に動作
- ボトルネックの可能性:
  ロード・インバランス

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ネイティブ

2014 Intel Corporation

ネイティブ

MPI ランクはコプロセッサー上で開始
mpirun -host mic0 -np 15 ./mic_exe

"ソースコードの修正は必要ないか" ?

- コードをコンパイルして実行
- パフォーマンスの保証はなし

大部分のコードはネイティブ実行の準備が出来ていない

- シリアルコード (もしくは、スレッドのスケーラビリティーに乏しい)
- スカラーコード (もしくは、ベクトル化の効率が低い)

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ネイティブ

"フラット" な MPI コードをネイティブ実行する前に熟考ください

- 60 コア上の 60 ランク = 各コアの利用率が低い
- 60 コア上の 240 ランク = ネットワーク接続と MPI の高いオーバーヘッド

MPI / OpenMP* のハイブリッドを推奨

- ドメイン分解と OpenMP ループでバランスを取ることができるか
  (例: 1 ランク x 240 スレッド、4 ランク x 60 スレッド、15 ランク x 16 スレッドなど)
- 競合を減らし無駄な/冗長なストレージを回避する
  (例: 全対全通信に 2M バイト = 60 x 60 x 2 = ～　7G バイト)

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

オフロード

2014 Intel Corporation

オフロード - データ転送

制御フロー

- プログラマーは、コプロセッサーに "オフロード" するコード領域を指定します
- 並列性に直結: 一度オフロードし、OpenMP*、インテル® TBB、インテル® MKL などを利用できます

データ共有

1. 明示的: プログラマーが、共有するデータと転送のタイミングを決定します
2. 暗黙的: プログラマーは共有データを宣言し、ランタイムがそれを最新の状態に保ちます

コンパイル

- オフロード領域は両方のプラットフォーム用にコンパイルされます
- 実行時、コプロセッサーが利用できない場合、プロセッサーに切り替えできます
- 他のコンパイラーは、オフロードプラグマを無視できます

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

オフロード

MPI ランクはホスト上で開始
mpirun -np 8 ./cpu_exe

- コプロセッサー毎に 1 つの MPI ランクを実行することが最も効率的
  複数のランクを実行すると、オーバーヘッドが伝搬し、ピニング調整が必要になります

- デバイスにオフロードされるコードはスレッドを実装しなければなりません
  (OpenMP*、インテル® Cilk、インテル® TBB、インテル® MKL)
  LEO (オフロード向けインテル言語拡張) と OpenMP* 4.0 のオフロードプラグマはデータと制御フローのみを行います

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

オフロードモデル

オフロード向けの 3 つの異なる利用モデル:

1. ホストがオフロードし、コプロセッサーがタスクを終了するのを待機する
2. ホストがオフロードし、その後異なるタスクを実行する
3. ホストがオフロードし、その後タスクの一部を実行する

他にもいくつかの組み合わせが考えられますが、上記は最も利用されるモデル

"リバースオフロード" は特殊なモデルです

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

オフロードモデル - オフロード/待機

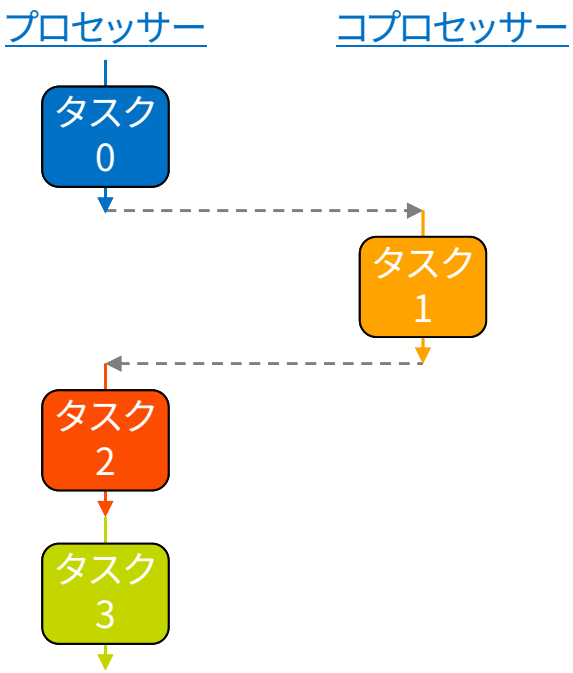
最も一般的なオフロードモデル

ソフトウェア開発の労力は最も少ない

アプリケーションの論理的なフローは変わりません

疑似コード:

```
Task0();
#pragma offload target …
Task1(0, N);
Task2();
Task3();
```



©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

オフロードモデル - 非対称

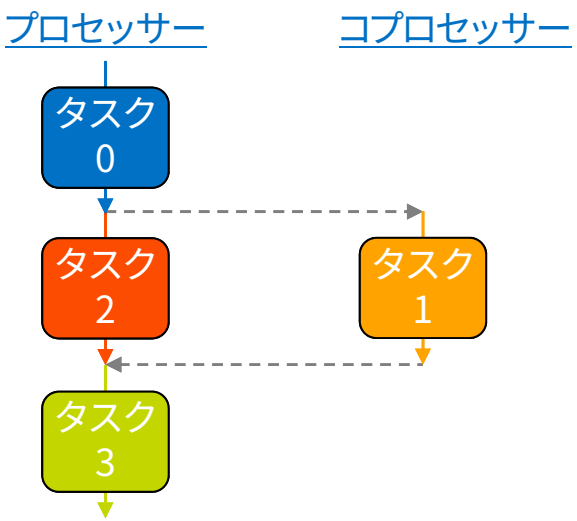
一般的なオフロードモデル

いくらかのソフトウェア開発の労力

タスクの依存関係を明確にし従う必要があります

疑似コード:

```
Task0();
#pragma offload target signal(s) …
Task1(0, N);
Task2();
#pragma offload_wait wait(s)
Task3();
```



©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

オフロードモデル - ワークシェアリング

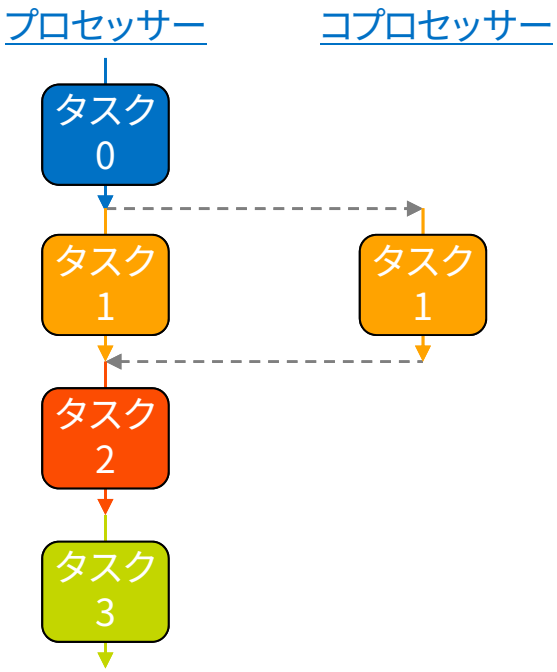
一般的だが少ないオフロードモデル

多くのソフトウェア開発の労力

ワークを手動で分割する必要があり、
依存性が特定されたらそれに従う必要がある

疑似コード:

```
Task0();
#pragma offload target signal(s) …
Task1(0, 3*N/4);
Task1(3*N/4, N);
#pragma offload_wait wait(s)
Task2();
Task3();
```



©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

シンメトリック (対称)

2014 Intel Corporation

シンメトリック (対称)

MPI ランクをプロセッサーとコプロセッサー上で開始:
mpirun -host localhost -np 8 ./cpu_exe :-host mic0 -np 15 ./mic_exe

コードをうまく対称的に実行するには、効率よくネイティブ実行する必要があります

パフォーマンスの差 (1 コプロセッサー ≈ 2 プロセッサー) は、ロードインバランスの原因になります。アプリケーションは負荷を調整する方法を持つ必要があります:

- ランクごとに負荷を固定する: コプロセッサー上でプロセッサーよりも多くのランクを実行する
- ランクタイプごとに負荷を固定する: ランクタイプに基づいてドメインを分割を調整します
- ダイナミック・ロードバランス: すべてのランクタイプは、共有キューからワークを受け取ります

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

非対称と対称

MPI は、"通信のアセンブリー" であると言えます:

- 任意の通信パターンをサポートしています
- ランクの任意のグループ化をサポートしています (MPI コミュニケーター)
- 複数プログラム複数データ (MPMD) の設計に基づく

"対称" モデルは、すべてに使用できます ☺

- オフロード
- リバースオフロード
- タスクに特化

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

正しいモデルを選択

2014 Intel Corporation

正しいモデルを選択

ネイティブ/シンメトリック (対称) モデルに適している理由：

- 元のコードは、全体にわたってスレッド/ベクトル並列処理を実装している
- 並列性を最大限に高めるため、コードを再構成する最善の方法を検討します
- 将来のプラットフォームに向けたコードベースを備える
- 既存のハードウェアを完全に使いこなす (プロセッサーとコプロセッサー)

ネイティブ/シンメトリック(対称)モデルに適していない理由:

- "すべてを再コンパイルしなければならない"
- "オフロードモデルを導入したくない"

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

正しいモデルを選択

オフロードモデルに適している理由:

- Knights Corner† 向けのメモリー・フットプリントが非常に多い (> 16G バイト)
- 大きなシリアル領域 (I/O など)
- ライブラリーの依存性

オフロードモデルに適していない理由:

- 既存の CUDA*/OpenCL* 実装はオフロードを利用しています
- "アプリケーション全体をモダン化する必要はありません"

† 開発コード名

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

将来のプラットフォーム

2014 Intel Corporation

Knights Landing†

Knights Landing† が 3 つのプログラミング・モデルに与える影響

- ブート可能なプロセッサー
  いくつかのマシン (例えば Cori) は、対称モードではない同種です
- 新しい命令/マイクロアーキテクチャーの変更:
  パフォーマンス上の特性は変更される可能性があり、より細かなチューニングが期待されます
  IMCI 組み込み関数を使用するコードは、インテル® AVX-512 へ書き換える必要があります
- 3x シングルスレッドのパフォーマンス (対 Knights Corner†)
  オフロードコードの一部は、ネイティブ実行に適するようになります
- インテル® Omni スケール・ファブリック
  特定の構成では、"オフロード" は "ファブリックを介したオフロード" になります
  ネイティブと対称モデル向けに MPI パフォーマンスが改善されます

ハードウェア、特定のソフトウェア、またはサービスにおけるインテル・テクノロジーが有効化される必要があります。システムの製造元や販売店に確認してください

† 開発コード名

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

Knights Landing†

Knights Landing† が一般的なプログラミング・モデルに与える影響

- 高帯域オンパッケージ・メモリー
  新しいキャッシュ・レベルのブロック化/タイル化
  ストリーミング操作のパフォーマンス向上
- タイル
  NUMA アフィニティーの要件とベスト・プラクティスへの潜在的な変化

† 開発コード名

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

まとめ

- シンプルなコアと 512 ビット幅の強力なベクトル命令
- Linux* “uOS“ が動作
- 従来のソフトウェア開発ツールとプログラミング・モデルを利用できる
- 3 つのプログラミング・モデルに対応
  - オフロード / ネイティブ / シンメトリック
- 将来のプラットフォーム

©2015 Intel Corporation. 無断での引用、転載を禁じます。* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

intel®

法務上の注意書きと最適化に関する注意事項

本資料の情報は、現状のまま提供され、本資料は、明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず、いかなる知的財産権のライセンスも許諾するものではありません。製品に付属の売買契約書『Intel's Terms and Conditions of Sale』に規定されている場合を除き、インテルはいかなる責任を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証 (特定目的への適合性、商品性に関する保証、第三者の特許権、著作権、その他、知的財産権の侵害への保証を含む) をするものではありません。
性能に関するテストに使用されるソフトウェアとワークロードは、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。

© 2015 Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Cilk、Intel Xeon Phi、VTune、Xeon は、アメリカ合衆国および / またはその他の国における Intel Corporation の商標です。

最適化に関する注意事項

インテル® コンパイラーは、互換マイクロプロセッサー向けには、インテル製マイクロプロセッサー向けと同等レベルの最適化が行われない可能性があります。これには、インテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2)、インテル® ストリーミング SIMD 拡張命令 3 (インテル® SSE3)、ストリーミング SIMD 拡張命令 3 補足命令 (SSSE3) 命令セットに関連する最適化およびその他の最適化が含まれます。インテルでは、インテル製ではないマイクロプロセッサーに対して、最適化の提供、機能、効果を保証していません。本製品のマイクロプロセッサー固有の最適化は、インテル製マイクロプロセッサーでの使用を目的としています。インテル® マイクロアーキテクチャーに非固有の特定の最適化は、インテル製マイクロプロセッサー向けに予約されています。この注意事項の適用対象である特定の命令セットの詳細は、該当する製品のユーザー・リファレンス・ガイドを参照してください。

改訂 #20110804