ハイパフォーマンス・ソフトウェア・カンファレンス

インテル® マイクロアーキテクチャー Skylake Server† /Skylake-X† と インテル® AVX-512 プログラミング

インテル® AVX-512 と新しいプロセッサーを使用してパフォーマンスを向上

2017 年 12 月
iSUS 編集長
すがわら　きよふみ

† 開発コード名

目的

- インテル® AVX-512 命令セットと新しいキャッシュ･アーキテクチャーが既存アプリケーションのパフォーマンスに与える影響を検証する
- 最新のコンパイラーの機能を理解する

© 2017 iSUS *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。

内容

- 残念なお知らせと、嬉しいお知らせ
- Skylake Server† マイクロアーキテクチャー
- Skylake Server† 向けプログラミング
  - コンパイラー・オプション
  - インテル® AVX-512
  - キャッシュ・アーキテクチャーの影響 (内部ユニット、キャッシュ)
  - コンパイラーの振る舞いと最適化
  - 命令セットと電力管理
- まとめ

† 開発コード名

© 2017 iSUS *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。

残念なお知らせと、嬉しいお知らせ

- インテル® C++ コンパイラー V18 では、インテル® Cilk™ Plus の機能が非推奨となり将来サポートされなくなることが発表されました
  - 「Cilk がやってきた」から「さよなら Cilk Plus」へ
    https://www.isus.jp/products/c-compilers/goodbye-cilk-plus-01/

- OpenMP* TR4 バージョン 5.0 プレビュー 1 の機能がサポートされました
  - コンパイラーのリリースノートをご覧ください
    https://www.isus.jp/products/psxe/c-compiler-18-0-for-win-rn-for-psxe2018/#OMP5X

© 2017 iSUS *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。

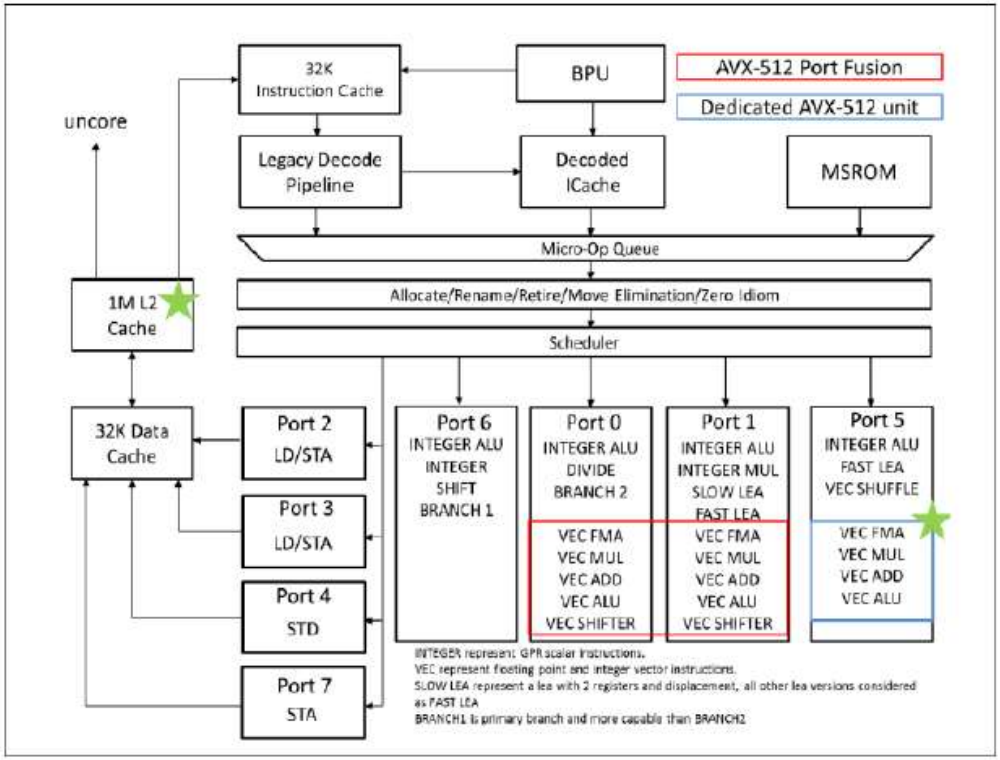
インテル® マイクロアーキテクチャー Skylake Server† の特徴

- Skylake Server† マイクロアーキテクチャー・ベースの新しいコアは、Kaby Lake† マイクロアーキテクチャーを基に改善を行っています
- インテル® アドバンスト・ベクトル・エクステンション 512 (インテル® AVX-512) のサポート
- ソケットあたりのコア数の増加 (最大 28 対最大 22)
- Skylake† マイクロアーキテクチャーではソケットあたり 6 本のメモリーチャンネル (Broadwell† マイクロアーキテクチャーは 4 チャンネル)
- より大きな L2 キャッシュと小さな非インクルーシブな L3 キャッシュ
- インテル® Optane™ テクノロジーをサポート
- インテル® Omni-Path アーキテクチャー (インテル® OPA)
- サブ NUMA クラスタリング (SNC) をサポート

† 開発コード名

© 2017 iSUS *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® マイクロアーキテクチャー Skylake Server† の CPU コア・パイプライン



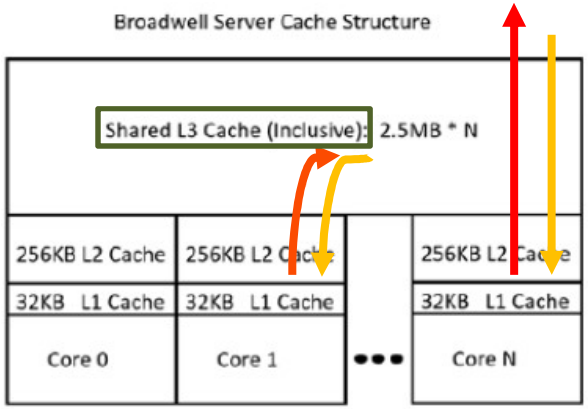
† 開発コード名

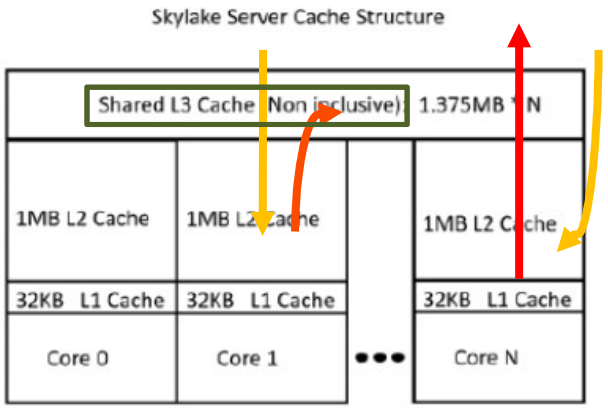
© 2017 iSUS *その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

インテル® マイクロアーキテクチャー Skylake Server† のキャッシュ

キャッシュレベル	カテゴリー	Broadwell† マイクロアーキテクチャー	Skylake Server† マイクロアーキテクチャー
L1 データ・キャッシュ・ユニット (DCU)	サイズ [KB]	32	32
	レイテンシー [サイクル]	4-6	4-6
	最大帯域幅 [バイト/サイクル]	96	192
	持続帯域幅 [バイト/サイクル]	93	133
	連想性 [ウェイ]	8	8
L2 中間キャッシュ (MLC)	サイズ [KB]	256	1024 (1MB)
	レイテンシー [サイクル]	12	14
	最大帯域幅 [バイト/サイクル]	32	64
	持続帯域幅 [バイト/サイクル]	25	52
	連想性 [ウェイ]	8	16
L3 ラストレベルキャッシュ (LLC)	サイズ [MB]	最大 2.5	最大 1.375[1]
	レイテンシー [サイクル]	50-60	50-70
	最大帯域幅 [バイト/サイクル]	16	16
	持続帯域幅 [バイト/サイクル]	14	15

† 開発コード名



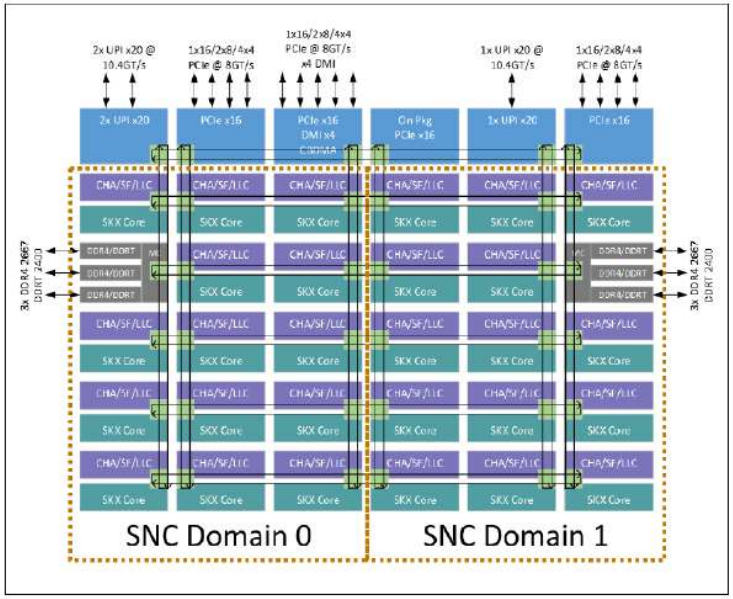


© 2017 iSUS *その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

インテル® マイクロアーキテクチャー Skylake Server† のサブ NUMA クラスタリング

サブ NUMA クラスタリング (SNC) は、ラスト・レベル・キャッシュ (LLC) からメモリーへの平均レイテンシーを改善するモードです。これは、以前の世代のインテル® Xeon® プロセッサー E5 ファミリーにおけるクラスターオンダイ (COD) の実装を置き換えるものです

- 単一のウルトラ・パス・インターコネクト (UPI) キャッシュ・エージェントが必要です
- リモートクラスターへのメモリー・アクセス・レイテンシーは小さく、UPI フローは必要ありません
- LLC でラインが複製されることはないため、LLC の容量をより効率良く利用できます
- リモートクラスターのアドレスがローカルクラスターの LLC にキャッシュされることがないため、クラスターオンダイ (COD) 方式に比べレイテンシーが大きくなることがあります



† 開発コード名

© 2017 iSUS *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。

詳細は、
インテル® 64 アーキテクチャーおよび IA-32 アーキテクチャー最適化リファレンス・マニュアル
をご覧ください

iSUS で公開中です: https://www.isus.jp/

最新 (2017 年 10 月公開の英語版) の完訳が終了しました

© 2017 iSUS *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® マイクロアーキテクチャー Skylake Server† 向けプログラミング

コンパイラー・オプション (命令セットの指定)

† 開発コード名

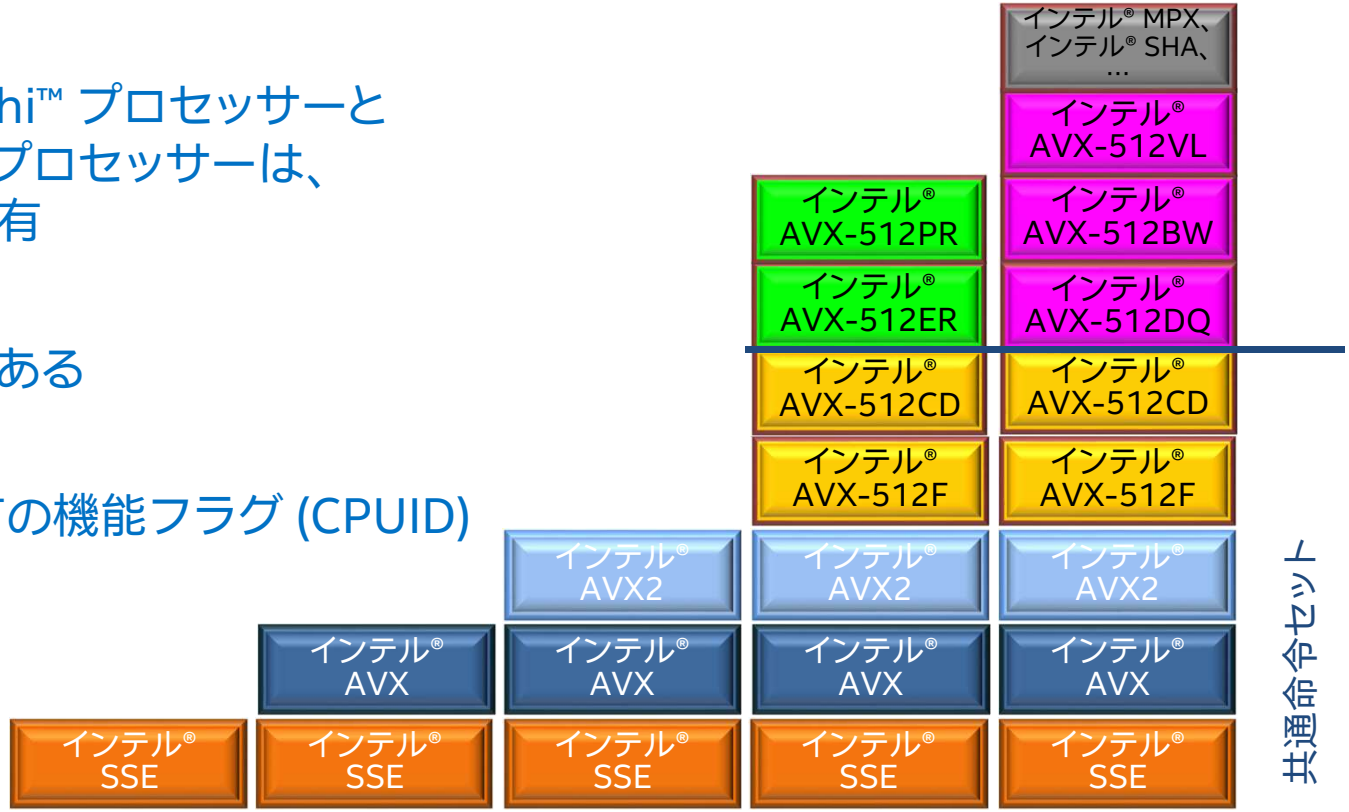
© 2017 iSUS *その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

インテル® AVX-512 - 比較

- インテル® Xeon Phi™ プロセッサーと
  インテル® Xeon® プロセッサーは、
  大部分の命令を共有

しかし、若干の相違がある

- サブセットは、固有の機能フラグ (CPUID)
  で示される



インテル® マイクロアーキテクチャー	NHM †	SNB †	HSW †	KNL †	SKL †

© 2017 iSUS *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。
† 開発コード名

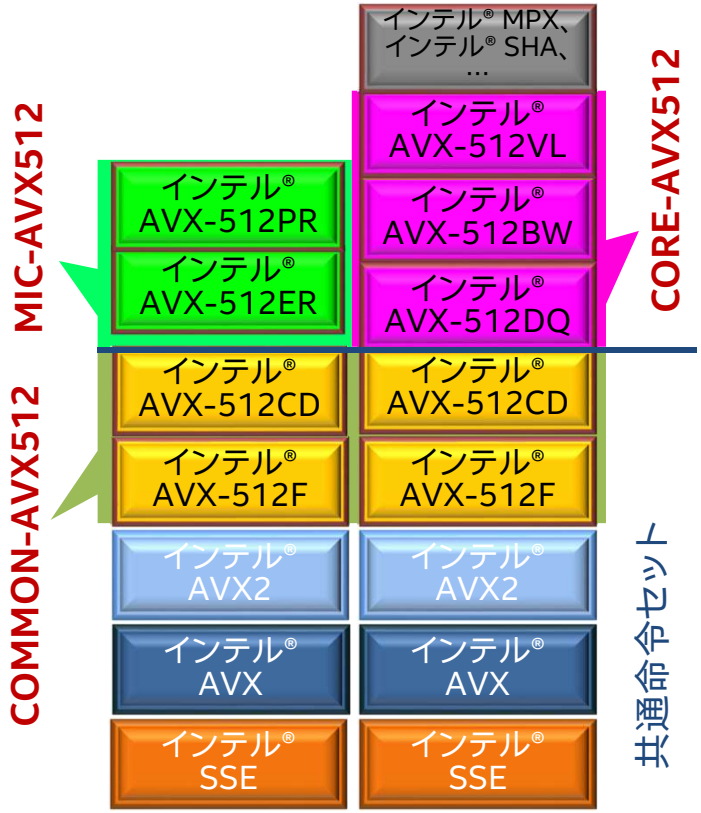
インテル® AVX-512 - インテル® コンパイラー

- オプションで SIMD 拡張を選択:
  /Qx<命令セット> または **/Qax<命令セット>**

  例:
  /QaxCOMMON-AVX512 /QxAVX2
  /QaxAVX2 /QxCORE-AVX512
  /QaxCORE-AVX512 /QxAVX2

- 特定のインテル® プロセッサーの
  アーキテクチャー向けにチューニング:
  -mtune=<プロセッサー>

  例: バージョン 17.0 以降の新機能
  -mtune=knl または **-mtune=skylake**



† 開発コード名

© 2017 iSUS *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。

自動プロセッサー・ディスパッチ

Intel Compiler 18.0 Update 1 Intel(R) 64 Visual Studio 2017

```
> icl pi.c /QxCORE-AVX512 -nologo /Qopenmp
pi.c

> pi

オペレーティング・システムとプロセッサーの両方がインテル(R) AVX512DQ, AVX512F, A
X, AVX512CD, AVX512BW、AVX512VL 命令をサポートしているか確認してください。

> icl pi.c /QxMIC-AVX512 -nologo /Qopenmp
pi.c

> pi

オペレーティング・システムとプロセッサーの両方がインテル(R) AVX512F, ADX, AVX512
R, AVX512PF、AVX512CD 命令をサポートしているか確認してください。
```

自動プロセッサー・ディスパッチなしで、指定された命令セットを生成

Intel Compiler 18.0 Update 1 Intel(R) 64 Visual Studio 2017

```
> icl pi.c /QaxCORE-AVX512 -nologo /Qopenmp
pi.c

> pi
Pi = 3.141593 Time = 0.442000
```

自動プロセッサー・ディスパッチありで、指定された命令セットと汎用命令セット (インテル® SSE2) を生成

© 2017 iSUS *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。

デフォルト命令セットの扱い

Intel Compiler 18.0 Update 1 Intel(R) 64 Visual Studio 2017

```
> icl pi.c /QaxAVX2 /QxCORE-AVX512 -nologo /Qopenmp
pi.c

> pi

オペレーティング・システムとプロセッサーの両方がインテル(R) AVX512DQ, AVX512F, A
X, AVX512CD, AVX512BW、AVX512VL 命令をサポートしているか確認してください。


> icl pi.c /QaxCORE-AVX512 /QxAVX2 -nologo /Qopenmp
pi.c

> pi
Pi = 3.141593 Time = 0.454000
```


© 2017 iSUS *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。

プロセッサー・ディスパッチはパフォーマンスに影響する

ランタイムチェックの影響

```
C:¥Users¥kiyo¥Desktop¥hakata¥code¥matmul>icl multiply.c /c /QaxCORE-AVX512 /QxCO
RE-AVX2 /Qopenmp /nologo
multiply.c

C:¥Users¥kiyo¥Desktop¥hakata¥code¥matmul>icl driver.c multiply.obj /QaxCORE-AVX2
 /nologo
driver.c

C:¥Users¥kiyo¥Desktop¥hakata¥code¥matmul>driver

ROW:101 COL: 101
Execution time is 3.269 seconds
GigaFlops = 6.241052
Sum of result = 195853.999899
```

```
C:¥Users¥kiyo¥Desktop¥hakata¥code¥matmul>icl multiply.c /c /QxCORE-AVX2 /Qopenmp
 /nologo
multiply.c

C:¥Users¥kiyo¥Desktop¥hakata¥code¥matmul>icl driver.c multiply.obj /QaxCORE-AVX2
 /nologo
driver.c

C:¥Users¥kiyo¥Desktop¥hakata¥code¥matmul>driver

ROW:101 COL: 101
Execution time is 2.872 seconds
GigaFlops = 7.103760
Sum of result = 195853.999899
```

> icl multiply.c /c /QaxCORE-AVX512 /QxCORE-AVX2
> icl driver.c multiply.obj /QaxCORE-AVX2 /Fedriver.exe

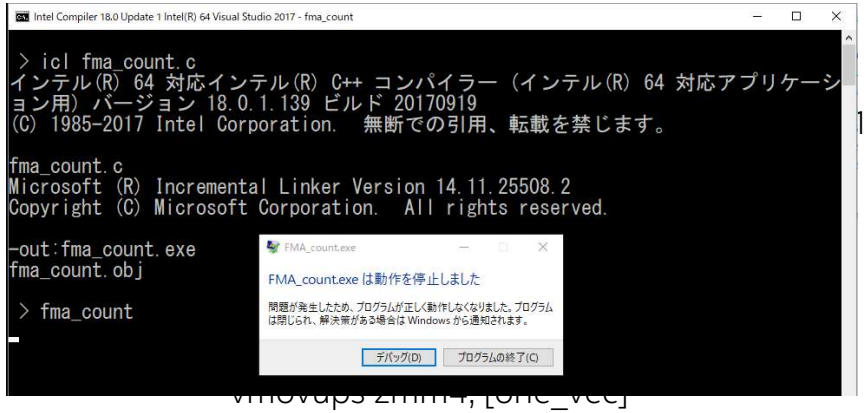
> icl multiply.c /c /QxCORE-AVX2
> icl driver.c multiply.obj /QaxCORE-AVX2 /Fedriver.exe

処理中に数万回、数十万回呼び出されるような下位の関数では、
ランタイムチェックを行うべきではない

© 2017 iSUS *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。

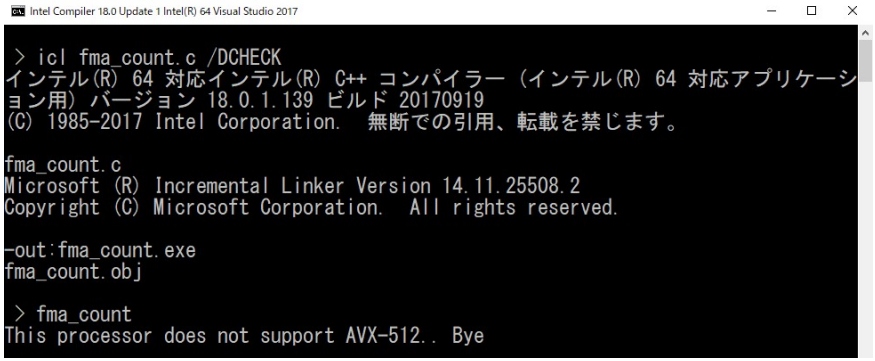
インテル® AVX-512 命令セット含むコード

ソースに組込み関数やインラインアセンブリーを含むコード (手動チェック)



1, 1, 1};

コード中で命令セットがサポートされるかチェックしない



コード中で命令セットがサポートされるかチェック

```
if(!_may_i_use_cpu_feature(_FEATURE_AVX512F)){
    printf("This processor does not support AVX-512.. Bye¥n");
    exit(-1);
}
```


© 2017 iSUS *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。

コード中で CPU の命令セットをチェックする

```
#include <immintrin.h>
#include <stdio.h>

int main(int argc, char *argv[]) {
 const unsigned long knl_features =
    (_FEATURE_AVX512F | _FEATURE_AVX512ER |
     _FEATURE_AVX512PF | _FEATURE_AVX512CD );
 if ( _may_i_use_cpu_feature( knl_features ) )
   printf("This CPU supports AVX-512F+CD+ER+PF as introduced in Knights Landing¥n");
 else
   printf("This CPU does not support all Knights Landing AVX-512 features¥n");
 return 1;
}
```

インテル® コンパイラーは、プロセッサーの機能をチェックする
_may_i_use_cpu_feature() 組込み関数をサポート

https://www.isus.jp/specials/target-intel-core-processors/
https://www.isus.jp/products/c-compilers/how-to-detect-knl-instruction-support/

© 2017 iSUS *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。

その他のインテル® AVX-512 向けオプション

/QxCORE-AVX512 また /QaxCORE-AVX512 などと併用

- ZMM レジスター利用を制御するオプション
  /Qopt-zmm-usage:<keyword>
  zmm レジスター利用のレベルを指定します。
  – **low** - コンパイルしたプログラムで ZMM レジスター利用による効果が得られる可能性が低いことをコンパイラーに伝えます。ZMM レジスター利用による効果があると分かっている場合を除いて、コンパイラーは ZMM レジスターを使用しません
  – **high** - 制限なしで ZMM コードを生成するようコンパイラーに伝えます。

不明な場合、両方のオプションでパフォーマンスを比較するか、
インテル® Advisor 2018 を使用してプロファイルしてください

© 2017 iSUS *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。

ベクトル化オプションはパフォーマンスに影響する

バイナリーに適用されている命令セットは?

Vectorization Advisor

Vectorization Advisor is a vectorization analysis toolset that lets you identify loops that will benefit most from vector parallelism, discover performance issues preventing from effective vectorization and characterize your memory vs. vectorization bottlenecks with Advisor Roofline model automation.

Program metrics

Elapsed Time 1.57s

Vector Instruction Set AVX512, AVX2, AVX, SSE　　Number of CPU Threads 1

Loop metrics

Metrics	Total	
Total CPU time	1.53s	100.0%
Time in 1 vectorized loop	0.73s	47.6%
Time in scalar code	0.80s	52.4%

Vectorization Gain/Efficiency

Vectorized Loops Gain/Efficiency	4.12x	~100%
Program Approximate Gain	2.49x	

ベクトル命令セットに
インテル® AVX が適用されている

サーベイレポートで
詳細を確認

ROOFLINE

Function Call Sites and Loops		Performance Issues	Self Time	Total Time	Type	Why No Vectorization?	Vectorized Loops				Instruction Set Analysis			Advanced
							Vector ISA	Efficiency	Gain...	VL (...	Traits	Data T...	Num..	
[loop in matvec at multiply.c:15]			0.34...	0.34...	Vectorized (Bo...		AVX512	~75%	6.03x	8	FMA	Float32; F	5; 8	Unrolled by 4
[loop in matvec at multiply.c:15]			0.317s	0.317s	Vectorized (Bo...		AVX512			8	FMA	Float32; F	8	Unrolled by 4
[loop in matvec at multiply.c:15]			0.030s	0.030s	Vectorized (Re...		AVX512			8	FMA	Float64; I	5	
[loop in matvec at multiply.c:13]			0.863s	1.210s	Scalar	inner loop was alre...					Permutes	Float32...	12	
[loop in main at driver.c:100]			0.000s	1.210s	Scalar	loop with function ...						Float64	2	
[loop in matvec at multiply.c:13]			0.000s	1.210s	Scalar	inner loop was alre...					FMA	Float64...	12	

© 2017 iSUS *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® マイクロアーキテクチャー Skylake Server† 向けプログラミング

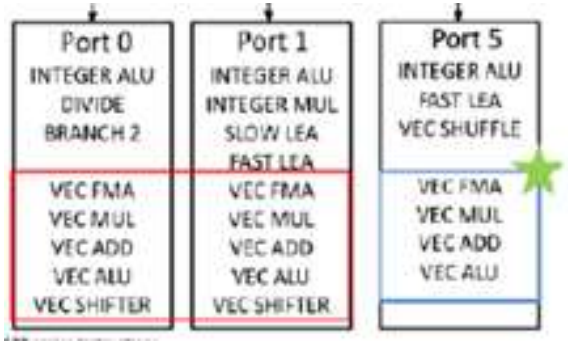
製品によって異なる内部ユニット

† 開発コード名

© 2017 iSUS *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。

FMA ユニット

Skylake† マイクロアーキテクチャー・ベースのプロセッサーには、2 つのインテル® AVX-512 FMA ユニット (ポート 0 と 5) を備えるものがある一方、同じアーキテクチャー・ベースのプロセッサーでも単一のインテル® AVX-512 FMA ユニット (ポート 0) を持つものがあります



2 つの FMA ユニットで実行されることを想定して最適化されたコードは、単一 FMA ユニット上での実行には最適でない可能性があります

コンパイラーには FMA ユニット数を制御するオプションは用意されていません

† 開発コード名

© 2017 iSUS *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® スケーラブル・プロセッサー

各ファミリーの特長

	インテル® Xeon® Bronze プロセッサー (3100シリーズ)	インテル® Xeon® Silver プロセッサー (4100シリーズ)	インテル® Xeon® Gold プロセッサー (5100シリーズ)	インテル® Xeon® Gold プロセッサー (6100シリーズ)	インテル® Xeon® Platinum プロセッサー (8100シリーズ)
汎用性の高いパフォーマンスとセキュリティー					
サポートされる最大コア数	8 コア	12 コア	14 コア	22 コア	28 コア
サポートされる最大周波数	1.70 GHz (8C/85W)	2.20 GHz (10C/85W)	3.60 GHz (4C/105W)	3.40 GHz (6C/115W)	3.60 GHz (4C/105W)
サポートされるCPUソケット数	最大 2	最大 2	最大 4	最大 4	最大 8
インテル® ウルトラ・パス・インターコネクト (インテル® UPI)	2	2	2	3	3
インテル® UPIの速度	9.6 GT/s	9.6 GT/s	10.4 GT/s	10.4 GT/s	10.4 GT/s
インテル® アドバンスト・ベクトル・エクステンション512 (インテル® AVX-512)	1 FMA	1 FMA	1 FMA	2 FMA	2 FMA
サポートされるメモリー速度(DDR4)	2133 MHz	2400 MHz	2400 MHz	2666 MHz	2666 MHz
サポートされる最大メモリー容量 (1ソケット当たり)	768 GB	768 GB	768 GB	768 GB / 1.5 TB	768 GB / 1.5 TB

© 2017 iSUS *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。

FMA ユニット数を判定する

2 つの FMA ユニットを備えるプロセッサーは、FMA と shuffle 命令を同時に実行できますが、単一ユニットのプロセッサーは同時実行できません

- FMA 命令のみを実行する関数を実行
- FMA 命令と shuffle 命令を実行する関数実行
- 2 つの実行時間を基にプロセッサーが 1 つまたは 2 つの FMA ユニットを搭載するかを特定します

詳細は、『インテル® 64 および IA-32 アーキテクチャー最適化リファレンス・マニュアル』(248966-038) の 15.20 節「単一 FMA ユニットを備える場合」をご覧ください:
https://www.isus.jp/technical-document/

© 2017 iSUS *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。

FMA ユニット数を確認

Windows* - インテル® Core™ i7-7800X X シリーズ･プロセッサー

```
$ icc -fasm-blocks FMA_count.c
$ ls
a.out  FMA_count.c
$ ./a.out
2 FMA server
$

processor       : 103
vendor_id       : GenuineIntel
cpu family      : 6
model           : 85
model name      : Intel(R) Xeon(R) Platinum 8170 CPU @ 2.10GHz
stepping        : 4
microcode       : 0x2000016
cpu MHz         : 1136.871
cache size      : 36608 KB
```

```
Intel Compiler 18.0 Update 1 Intel(R) 64 Visual Studio 2017

> icl fma_count.c
インテル(R) 64 対応インテル(R) C++ コンパイラー (インテル(R) 64 対応アプリケーシ
ョン用) バージョン 18.0.1.151 ビルド 20171018
(C) 1985-2017 Intel Corporation.  無断での引用、転載を禁じます。

fma_count.c
Microsoft (R) Incremental Linker Version 14.11.25547.0
Copyright (C) Microsoft Corporation.  All rights reserved.

-out:fma_count.exe
fma_count.obj

> fma_count
1 FMA server
```

Linux* - インテル® Xeon® Platinum 8170 プロセッサー

© 2017 iSUS *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。

命令グループ (2 FMA)

- グループ A の命令: vadd*; vfmadd*; vfnmsub*; vfnmadd*; vfnmsub*; vmax*; vmin*; vmul*; vscalef*; vsub*; vcvt*; vgetexp*; vfixupimm*; vrange*; vgetmant*; vreduce*; vcmp*; vcomi*; vdpp*; vhadd*; vhsub*; vrndscale*; vround*
- グループ B の命令: vpmaddubsw; vpmaddwd; vpmuldq; vpmulhrsw; vpmulhuw; vpmulhw; vpmullw; vpmuludq

すべての命令のソースが FMA ユニットから供給される場合、FMA ユニットは高速バイパスをサポートします。グループ A はポート 0 とポート 5 で 4 サイクルのレイテンシー、グループ B はポート 0 とポート 5 で 5 サイクルのレイテンシーです

© 2017 iSUS *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® マイクロアーキテクチャー Skylake Server† 向けプログラミング

異なるキャッシュ・アーキテクチャー (インクルーシブと、非インクルーシブ)

† 開発コード名

© 2017 iSUS *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。

キャッシュ最適化

配列 a, b, c はキャッシュのどの階層に収まるか?

2M バイト
256K バイト
32K バイト

```
for(j=1;j<n;j++){
    f=0;
    for(i=1;i<n;i++){
        f+=a[i]*b[i];
    }
    c[j]+=f;
}
```

1.375M バイト
1M バイト
32K バイト

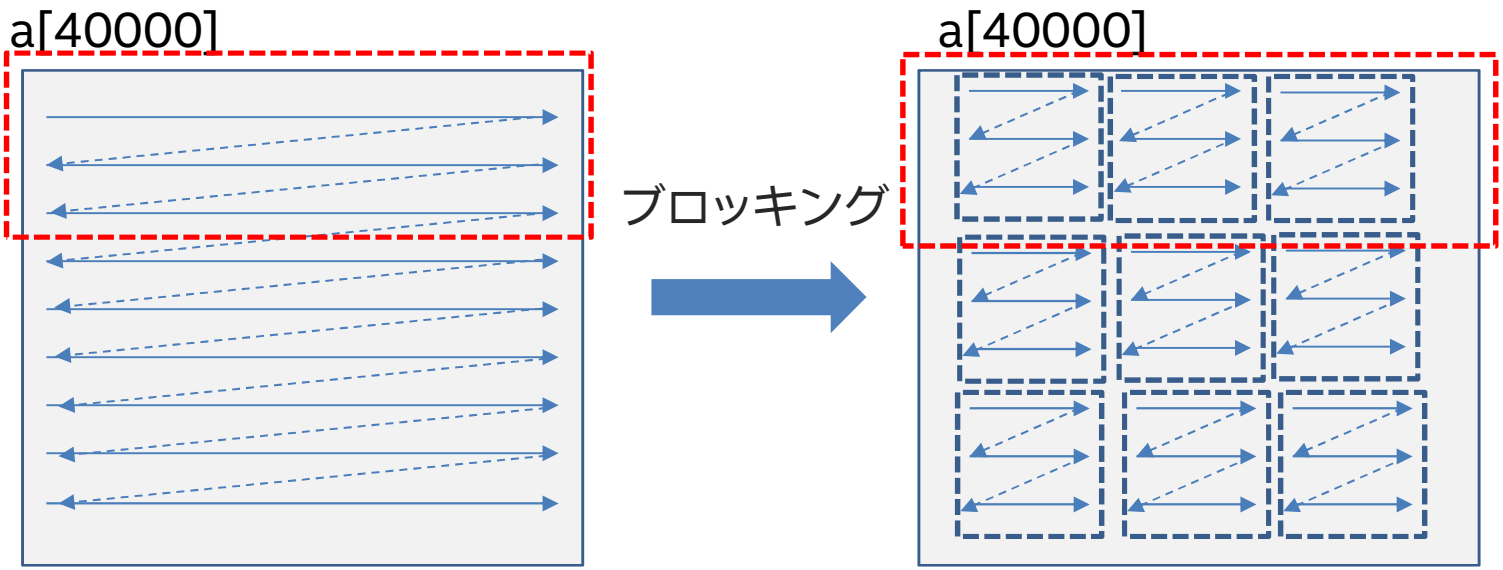
double a[100], b[100], c[100]

double a[40000], b[40000], c[40000]

© 2017 iSUS *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。

キャッシュ最適化 – 従来のプロセッサー

double a[40000], b[40000], c[40000] は、L2 キャッシュに収まらない



© 2017 iSUS *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。

ブロッキングの例

```
for(jj=1;jj<n/FACTOR+1;jj++){
 for(ii=1;ii<n/FACTOR+1;ii++){
  for(j=(jj-1)*FACTOR+1;j<min(jj*FACTOR,n);j++){
   f=0;
   for(i=(ii-1)*FACTOR+1;i<min(ii*FACTOR,n);i++){
    f=f+a[i]*b[i];
   }
   c[j]=c[j]+f;
  }
 }
}
```


© 2017 iSUS *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。

ブロッキングとパフォーマンス

プロセッサー	キャッシュ	ブロッキングなし	ブロッキングあり
インテル® Core™ i7-4790K プロセッサー(4.00GHz/4.40GHz)	8M/256K	0.47 秒	0.23 秒
インテル® Core™ i7-7800X X シリーズ・プロセッサー (3.50GHz/4.00GHz)	8.25M/1M	0.27 秒	0.29 秒
インテル® Core™ i7-5820K プロセッサー (3.30GHz/3.60GHz)	15M/256K	0.66 秒	0.28 秒
インテル® Xeon® プロセッサー E5-2699 v4 (2.20GHz/3.60GHz)	55M/256K	0.76 秒	0.30 秒

/QxCORE-AVX2 で生成した同じバイナリーを実行

© 2017 iSUS *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® マイクロアーキテクチャー Skylake Server† 向けプログラミング

コンパイラーの振る舞い

† 開発コード名

© 2017 iSUS *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。

ここで使用するプロセッサーの比較

	インテル® Core™ i7-5820K プロセッサー	インテル® Core™ i7-7800X X シリーズ・プロセッサー
アーキテクチャー	Haswell-E †	Skylake †
コア数	6 (12)	6 (12)
周波数	3.30GHz (3.60GHz)	3.50GHz (4.00GHz)
キャッシュ	32K/32K, 256K, 15MB	32K/32K, 1024KB, 8.25MB
メモリーチャネル数	4	4
メモリー	DDR4 2133	DDR4 2400

† 開発コード名

© 2017 iSUS *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。

検証 1: 簡単な計算のループ (pi)

```
#pragma omp simd reduction(+:sum) private(x)
for (i=0;i< num_steps; i++){
        x = (i+0.5)*step;
        sum = sum + 4.0/(1.0+x*x);
}
```

$ icl pi.c /Qopenmp

	デフォルト インテル® SSE2 (float/double)	/QxCORE-AVX2 (float/double)	/QxCORE-AVX512 (float/double/zmm)
インテル® Core™ i7-7800X X シリーズ・プロセッサー	0.48/0.57	0.27/0.56	0.17/0.54/0.32

© 2017 iSUS *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。

opt-zmm-usage オプションの例 2

Intel Compiler 18.0 Update 1 Intel(R) 64 Visual Studio 2017

```
> icl prime.cpp /QxCORE-AVX2 /nologo /WO
prime.cpp

> prime

(prime) 1から50000000までの範囲の素数の数は 3001135個。計算時間12.63秒

> icl prime.cpp /QxCORE-AVX512 /nologo /WO
prime.cpp

> prime

(prime) 1から50000000までの範囲の素数の数は 3001135個。計算時間13.94秒

> icl prime.cpp /QxCORE-AVX512 /Qopt-zmm-usage=high /nologo /WO
prime.cpp

> prime

(prime) 1から50000000までの範囲の素数の数は 3001135個。計算時間7.72秒
```

データ型は double

1 から 5 千万の間の素数の数を求める例

© 2017 iSUS *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。

OpenMP* で simdlen を制御

```
#pragma omp simd reduction(+:sum) private(x) simdlen(8 または 16)
for (i=0;i< num_steps; i++){
        x = (i+0.5)*step;
        sum = sum + 4.0/(1.0+x*x);
}
```

Intel Compiler 18.0 Update 1 Intel(R) 64 Visual Studio 2017

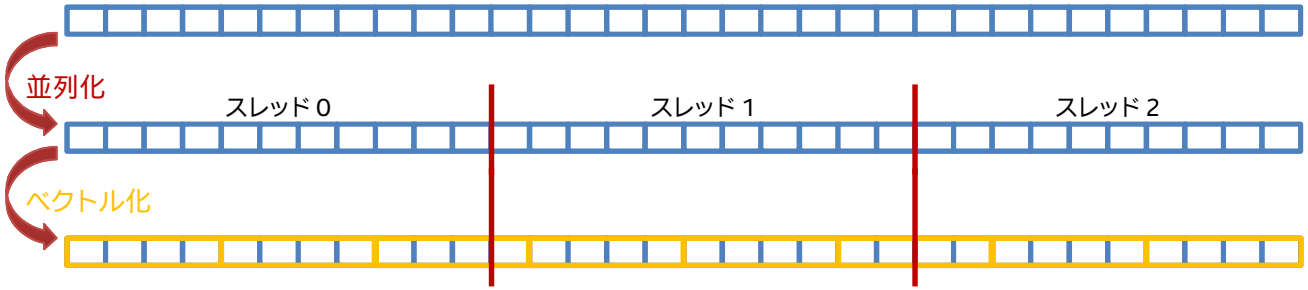
```
> icl pi.c /QxCORE-AVX512 -o piavx512len /Qopenmp /nologo
pi.c

> piavx512len
Data type double
Pi = 3.141593 Time = 0.323000
```

データ型は double

© 2017 iSUS *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。

ループ・スケジュール SIMD 修飾子の例

```
void sprod(float *a, float *b, int n) {
  float sum = 0.0f;
#pragma omp parallel for simd reduction(+:sum)  schedule(simd:static,5)
for (int k=0; k<n; k++)
    sum += a[k] * b[k];
  return sum;
}
```



新しい SIMD 修飾子は、コンパイラーとランタイムが SIMD レジスターのレングスにチャンクサイズを合わせることが可能です:

- 新しいチャンクサイズは、⌈*chunk_size/simdlen*⌉ * *simdlen*
- インテル® AVX2/インテル® AVX-512: 新しいチャンクサイズは、8/16 以上の 2 の累乗
- インテル® SSE: 新しいチャンクサイズは、4 以上の 2 の累乗

© 2017 iSUS *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® Core™ i7-5820K プロセッサー上のインテル® AVX2 コード

Source | Assembly | Assembly grouping: Address

So.	Source	CPU Time	Mem.
6	char *fp="float";		
7	#else		
8	#define FP_TYPE double		
9	char *fp="double";		
10	#endif		
11			
12	static long num_steps = 1000000000;		
13	FP_TYPE step;		
14			
15	double start, end;		
16			
17	int main ()		
18	{		
19	int i;		
20	FP_TYPE pi, x, sum = 0.0;		
21			
22	step = 1.0/(FP_TYPE) num_steps;		
23			
24	start = clock();		
25			
26	#pragma omp parallel for simd reduction(+:sum) priva		
27	//#pragma omp simd reduction(+:sum) private(x)		
28	for (i=0;i< num_steps; i++){	59.034ms	0.0%
29	x = ((FP_TYPE)i+(FP_TYPE)0.5)*step;	168.097ms	0.0%
30	sum = sum + (FP_TYPE)4.0/((FP_TYPE)1.0+x*x);	86.050ms	0.0%
31	}		
32	pi = step * sum;		
33	end = clock();		
34			
35	printf("Data type %s\n",fp);		
36	printf("Pi = %f Time = %f\n", pi, (double)((end-star		
37	}		
Sele...		86.050ms	0.0%

Address	Sou...	Assembly	CPU Time	Mem.
0x140006770		Block 1:		
0x140006770	29	vcvtdq2ps ymm13, ymm0		
0x140006774	28	add eax, 0x10	1.001ms	0.0%
0x140006777	29	vaddps ymm10, ymm5, ymm13		
0x1400067c	29	vmulps ymm13, ymm4, ymm10	22.013ms	0.0%
0x140006781	30	vfmadd213ps ymm13, ymm13, ymm12		
0x140006786	30	vrcpps ymm11, ymm13		
0x1400067b	30	vmulps ymm13, ymm11, ymm13	26.015ms	0.0%
0x140006790	30	vaddps ymm10, ymm11, ymm11	8.005ms	0.0%
0x140006795	30	vfnmadd213ps ymm13, ymm11, ymm10	3.002ms	0.0%
0x1400679a	29	vpaddd ymm10, ymm0, ymm15	49.028ms	0.0%
0x1400679f	30	vfmadd231ps ymm2, ymm13, ymm1		
0x1400067a4	29	vcvtdq2ps ymm0, ymm10	61.035ms	0.0%
0x1400067a9	29	vaddps ymm11, ymm5, ymm0		
0x1400067ad	29	vmulps ymm13, ymm4, ymm11	7.004ms	0.0%
0x1400067b2	30	vfmadd213ps ymm13, ymm13, ymm12		
0x1400067b7	30	vrcpps ymm0, ymm13	20.012ms	0.0%
0x1400067bc	30	vmulps ymm13, ymm0, ymm13	10.006ms	0.0%
0x1400067c1	30	vaddps ymm11, ymm0, ymm0	19.011ms	0.0%
0x1400067c5	30	vfnmadd213ps ymm13, ymm0, ymm11		
0x1400067ca	29	vpaddd ymm0, ymm10, ymm15	29.017ms	0.0%
0x1400067cf	30	vfmadd231ps ymm3, ymm13, ymm1		
0x1400067d4	28	cmp eax, 0x3b9aca00	58.034ms	0.0%
0x1400067d9	28	jb 0x140006770 <Block 1>		
Highlighted ...			86.050ms	0.0%

© 2017 iSUS *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® Core™ i7-7800X X シリーズ・プロセッサー上のインテル® AVX-512 コード

Source | Assembly | Assembly grouping: Address

So.	Source
8	#define FP_TYPE double
9	char *fp="double";
10	#endif
11	
12	static long num_steps = 1000000000;
13	FP_TYPE step;
14	
15	double start, end;
16	
17	int main ()
18	{
19	int i;
20	FP_TYPE pi, x, sum = 0.0;
21	
22	step = 1.0/(FP_TYPE) num_steps;
23	
24	start = clock();
25	
26	//#pragma omp parallel for simd reduction(+:sum) private(x)
27	#pragma omp simd reduction(+:sum) private(x) simdlen(8)
28	for (i=0;i< num_steps; i++){
29	x = ((FP_TYPE)i+(FP_TYPE)0.5)*step;
30	sum = sum + (FP_TYPE)4.0/((FP_TYPE)1.0+x*x);
31	}
32	pi = step * sum;
33	end = clock();
34	
35	printf("Data type %s\n",fp);
36	printf("Pi = %f Time = %f\n", pi, (double)((end-start)/CLOCKS_P
37	}

Address	Sou...	Assembly
0x14000648b	29	vcvtdq2pd zmm17, ymm7
0x140006491	29	vaddpd zmm18, zmm3, zmm17
0x140006497	29	vmulpd zmm19, zmm4, zmm18
0x14000649d	30	vfmadd213pd zmm19, zmm19, zmm0
0x1400064a3	30	vdivpd zmm20, zmm1, zmm19
0x1400064a9	30	vaddpd zmm28, zmm16, zmm20
0x1400064af	27	vpaddd ymm16, ymm7, ymm2
0x1400064b5	29	vcvtdq2pd zmm21, ymm16
0x1400064bb	29	vaddpd zmm22, zmm3, zmm21
0x1400064c1	29	vmulpd zmm23, zmm4, zmm22
0x1400064c7	30	vfmadd213pd zmm23, zmm23, zmm0
0x1400064cd	30	vdivpd zmm24, zmm1, zmm23
0x1400064d3	30	vaddpd zmm7, zmm24, zmm5
0x1400064d9	27	vpaddd ymm5, ymm16, ymm2
0x1400064df	27	vpaddd ymm30, ymm5, ymm2
0x1400064e5	29	vcvtdq2pd zmm25, ymm5
0x1400064eb	29	vaddpd zmm26, zmm3, zmm25
0x1400064f1	29	vmulpd zmm27, zmm4, zmm26
0x1400064f7	30	vfmadd213pd zmm27, zmm27, zmm0
0x1400064fd	30	vdivpd zmm29, zmm1, zmm27
0x140006503	30	vaddpd zmm16, zmm28, zmm29
0x140006509	29	vcvtdq2pd zmm31, ymm30
0x14000650f	29	vaddpd zmm5, zmm3, zmm31
0x140006515	29	vmulpd zmm25, zmm4, zmm5
0x14000651b	30	vfmadd213pd zmm25, zmm25, zmm0
0x140006521	30	vdivpd zmm31, zmm1, zmm25
0x140006527	30	vaddpd zmm5, zmm31, zmm7
0x14000652d	27	vpaddd ymm7, ymm30, ymm2
0x140006533	28	cmp eax, 0x3b9aca00
0x140006538	28	jb 0x140006488 <Block 1>

© 2017 iSUS *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。

検証 2: メモリー依存のループ

```
for(j=0; j<p; j++)
    for(k=0; k<n; k++)
        for(i=0; i<m; i++)
            c[i][j] = c[i][j] + a[i][k] * b[k][j];
```

$ icl matrix1.cpp /O1 /Qx 命令セット

1024 (/O1 なし)	デフォルト インテル® SSE2	/QxCORE-AVX2	/QxCORE-AVX512
インテル® Core™ i7-7800X X シリーズ・プロセッサー	8.71 (0.56) 111.81 (5.83)	8.82 (0.47) 110.92 (5.41)	8.6 (0.46) 111.36 (5.33)
インテル® Core™ i7-5820K プロセッサー	15.79 (0.41) 143.99 (5.57)	15.74 (0.33) 143.76 (4.93)	NA

© 2017 iSUS *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。

ユニットストライドではないループ

Source | Assembly | Assembly grouping: Address

So.	Source	CPU
25	}	
26	}	
27		
28	start = clock();	
29		
30	#pragma omp parallel for	
31	for(i=0; i<m; i++)	
32	for(k=0; k<n; k++)	
33	#pragma omp simd	
34	for(j=0; j<p; j++)	349.600ms
35	c[i][j] = c[i][j] + a[i][k] * b[k][j];	589.326ms
36		
37	finish = clock();	
38	walltime = (double)(finish - start)/CLOCKS_PER_SEC;	
39	printf("\n Matmul kernel wall clock time = %f sec.\n"	
40	printf("\n Result (%f)\n", c[SIZE-1][SIZE-1]);	
41	return 0;	
42	}	

Address	Sou...	Assembly	CPU Tim
0x140006773		Block 1:	
0x140006773	35	lea r12, ptr [rdi+rbp*8]	2.997ms
0x140006777	35	vmovupd ymm1, ymmword ptr [r14+r12*1+0xd498	
0x140006781	35	lea r15, ptr [rsi+rbp*8]	518.407ms
0x140006785	35	vfmadd213pd ymm1, ymm0, ymmword ptr [r14+r1	64.926ms
0x14000678f	34	add rbp, 0x4	220.747ms
0x140006793	35	vmovupd ymmword ptr [r14+r15*1+0x1549840],	2.997ms
0x14000679d	34	cmp rbp, r10	128.853ms
0x1400067a0	34	jb 0x140006773 <Block 1>	

Address

Address	Sou...	Assembly	CPU Tim
0x140006774		Block 1:	
0x140006774	35	lea r12, ptr [rdi+rbp*8]	0.999ms
0x140006778	35	vmovups zmm1, zmmword ptr [r14+r12*1+0xd498	
0x140006783	35	lea r15, ptr [rsi+rbp*8]	624.286ms
0x140006787	35	vfmadd213pd zmm1, zmm0, zmmword ptr [r14+r1	46.946ms
0x140006792	34	add rbp, 0x8	147.831ms
0x140006796	35	vmovupd zmmword ptr [r14+r15*1+0x1549840],	2.997ms
0x1400067a1	34	cmp rbp, r10	105.879ms
0x1400067a4	34	jb 0x140006774 <Block 1>	

© 2017 iSUS *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。

検証 3: メモリー依存のループ (スレッド/ベクトル)

```
#pragma omp parallel for
  for(i=0; i<m; i++)
        for(k=0; k<n; k++)
#pragma omp simd
                    for(j=0; j<p; j++)
                        c[i][j] = c[i][j] + a[i][k] * b[k][j];
```

$ icl matrix2.cpp /Qx命令セット [/Qopenmp]

1024/2048/4096	デフォルト SSE2	/QxCORE-AVX2	/QxCORE-AVX512
インテル® Core™ i7-7800X X シリーズ・プロセッサー	0.54 (0.10) 5.97 (0.77) 51 (7.42)	0.53 (0.09) 5.44 (0.67) 48 (7.88)	0.53 (0.08) 5.42 (0.66) 48 (7.99)
インテル® Core™ i7-5820K プロセッサー	0.46 (0.08) 5.60 (0.75) 48.3 (5.84)	0.30 (0.07) 4.87 (0.72) 46.17 (5.76)	

© 2017 iSUS *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。

新たな命令セットの恩恵

インテル® C++ コンパイラーによってインテル® AVX2 以前の命令セットではベクトル化されず、インテル® Xeon® プロセッサーのインテル® AVX-512 を利用してベクトル化されるようになったループでは劇的にパフォーマンが向上することがあります。特に、インテル® AVX-512CD およびインテル® AVX-512F サブセットの新機能は、このようなアプリケーションに対し、コードを変更しなくても、コンパイラーが最適化されたベクトルコードを自動生成します

イメージのヒストグラム計算のおける依存性解決の例

テストケース	ループ	ベースライン (インテル® AVX2)	ベクトル化によるスピードアップ (インテル® AVX-512)
入力イメージ	ループ 1	1	2.2
	ループ 2	1	7.0
鮮明な入力イメージ	ループ 1	1	2.6
	ループ 2	1	7.4
滑らかな入力イメージ	ループ 1	1	1.7
	ループ 2	1	5.6

https://www.isus.jp/products/c-compilers/improve-vectorization-performance-using-avx-512/

© 2017 iSUS *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® マイクロアーキテクチャー Skylake Server† 向けプログラミング

命令セットと電力管理

† 開発コード名

© 2017 iSUS *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。

Skylake Server† の電力管理

- Skylake Server† マイクロアーキテクチャーは、それぞれのコアの実行周波数を動的に選択します。選択される周波数は、命令タイプ、命令幅、および一定時間に実行されるベクトル命令数などの組み合わせに依存します。プロセッサーはまた、同様の特性を持つコアの数を考慮します

- Broadwell† マイクロアーキテクチャー・ベースのインテル® Xeon® プロセッサーも同じように振る舞いますが、256 ビット・ベクトル命令のみをサポートするため影響は軽度です。Skylake Server† マイクロアーキテクチャーがサポートするインテル® AVX-512 命令は、潜在的にインテル® AVX2 命令よりも多くの電流と電力を必要とします

- プロセッサーは、必要に応じて動的に最大周波数をより高いか低いレベルに調整します。そのため、プログラムが実行されている間の最大周波数は異なる可能性があります

† 開発コード名

© 2017 iSUS *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® ターボ・ブースト・テクノロジーの最大コア周波数レベル

レベル	カテゴリー	周波数レベル	最大周波数 (POn)	命令タイプ
0	軽量なインテル® AVX2 命令	最も高い	最大	スカラー、インテル® AVX (128 ビット)、インテル® SSE、インテル® AVX2 (FP または INT MUL/FMA なし)
1	重いインテル® AVX 2 命令 + 軽量なインテル® AVX-512 命令	中間	最大 インテル® AVX2	インテル® AVX2 FP + INT MUL/FMA、インテル® AVX-512 (FP または MUL/FMA なし)
2	重いインテル® AVX-512 命令	最低	最大 インテル® AVX-512	インテル® AVX-512 FP + INT MUL/FMA

最大周波数 (POn) は、カテゴリー内のコア数に応じた周波数の配列で表されます。多くのコアがある時点でカテゴリーに含まれると、最大周波数は低くなります

© 2017 iSUS *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。

ワークロードの混在と周波数

- それぞれのコアの周波数はワークロードの要求に基づいて独立して決定されます
- 新しいレベルへの移行には最大 500 μ秒かかります
- 高い周波数レベルに戻る前には、およそ 2 ms のタイマーが適用されます

次のパフォーマンス・モニタリング・イベントは、3 つの周波数レベルにおいて費やされるサイクル数を決定するために使用されます:

- CORE_POWER.LVL0_TURBO_LICENSE: 最大周波数が P0n でコアが動作したコアサイクル
- CORE_POWER.LVL1_TURBO_LICENSE: 最大周波数が P0n-AVX2 でコアが動作したコアサイクル
- CORE_POWER.LVL2_TURBO_LICENSE: 最大周波数が P0n-AVX-512 でコアが動作したコアサイクル

© 2017 iSUS *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。

インテル製品　ユーザー別　サポート　intel　Japan (日本語)

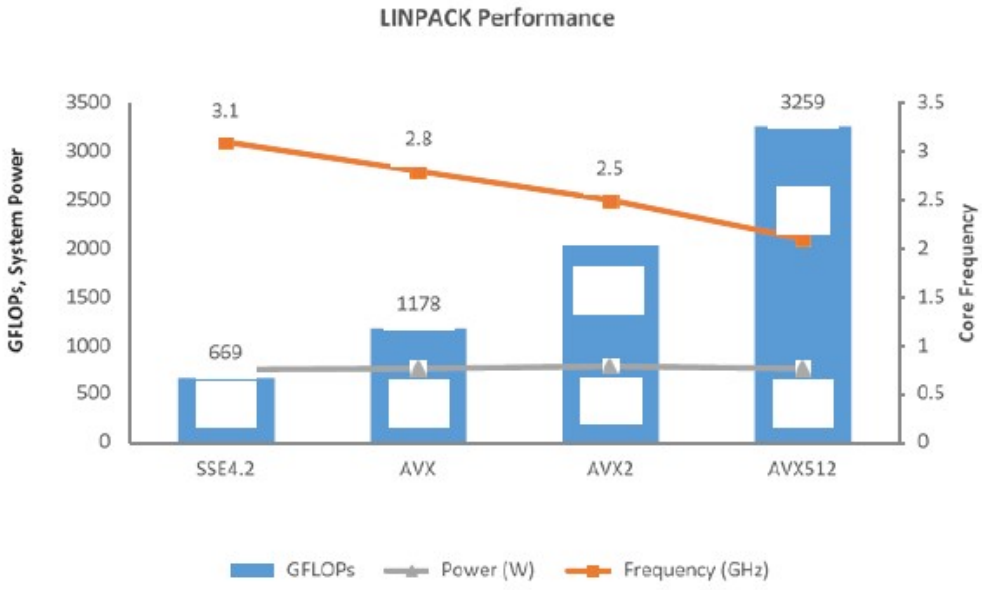
サポートホーム › 製品仕様 › プロセッサー　　製品仕様の検索

製品名	ステータス	発売日	コアの数	ターボ・ブースト利用時の最大周波数	プロセッサー・ベース動作周波数	キャッシュ	比較 すべて | なし
Intel® Core™ i9-7980XE Extreme Edition Processor	Launched	Q3'17	18	4.20 GHz	2.60 GHz	24.75 MB	
Intel® Core™ i9-7960X X-series Processor	Launched	Q3'17	16	4.20 GHz	2.80 GHz	22 MB	
Intel® Core™ i9-7940X X-series Processor	Launched	Q3'17	14	4.30 GHz	3.10 GHz	19.25 MB	
Intel® Core™ i9-7920X X-series Processor	Launched	Q3'17	12	4.30 GHz	2.90 GHz	16.5 MB L3	
インテル® Core™ i9-7900X X シリーズ・プロセッサー	Launched	Q2'17	10	4.30 GHz	3.30 GHz	13.75 MB L3	
インテル® Core™ i7-7820X X シリーズ・プロセッサー	Launched	Q2'17	8	4.30 GHz	3.60 GHz	11 MB L3	
インテル® Core™ i7-7800X X シリーズ・プロセッサー	Launched	Q2'17	6	4.00 GHz	3.50 GHz	8.25 MB L3	
インテル® Core™ i7-7740X X シリーズ・プロセッサー	Launched	Q2'17	4	4.50 GHz	4.30 GHz	8 MB	

https://ark.intel.com/ja/products/series/125191/Intel-Xeon-Scalable-Processors

© 2017 iSUS *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。

ワークロードとパフォーマンス



インテル® AVX-512 命令を実行するワークロードが、命令全体に占める割合が大きいと、インテル® AVX2 命令に比べパフォーマンス・ゲインは高くなりますが、動作周波数は低くなる可能性があります

例えば、高い比率で重いインテル® AVX-512 命令を最大周波数で要求するディープラーニングのワークロードは、インテル® AVX2 をターゲットとする同じワークロードに対し、1.3 倍から 1.5 倍のパフォーマンスを向上できます (両者とも Skylake Server† マイクロアーキテクチャー上で実行)

† 開発コード名

© 2017 iSUS *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。

ワークロードと推奨コンパイラー・オプション

最適なコンパイラー・オプションを特定するため、次に示す命令セットオプションでアプリケーションをビルドして、最も高いパフォーマンスを示す命令セットを選択します:

- -xCORE-AVX2 -mtune=skylake-avx512 (Linux* および macOS*)

  /QxCORE-AVX2 /tune=skylake-avx512 (Windows*)

- -xCORE-AVX512 -qopt-zmm-usage=low (Linux* および macOS*)

  /QxCORE-AVX512 /Qopt-zmm-usage:low (Windows*)

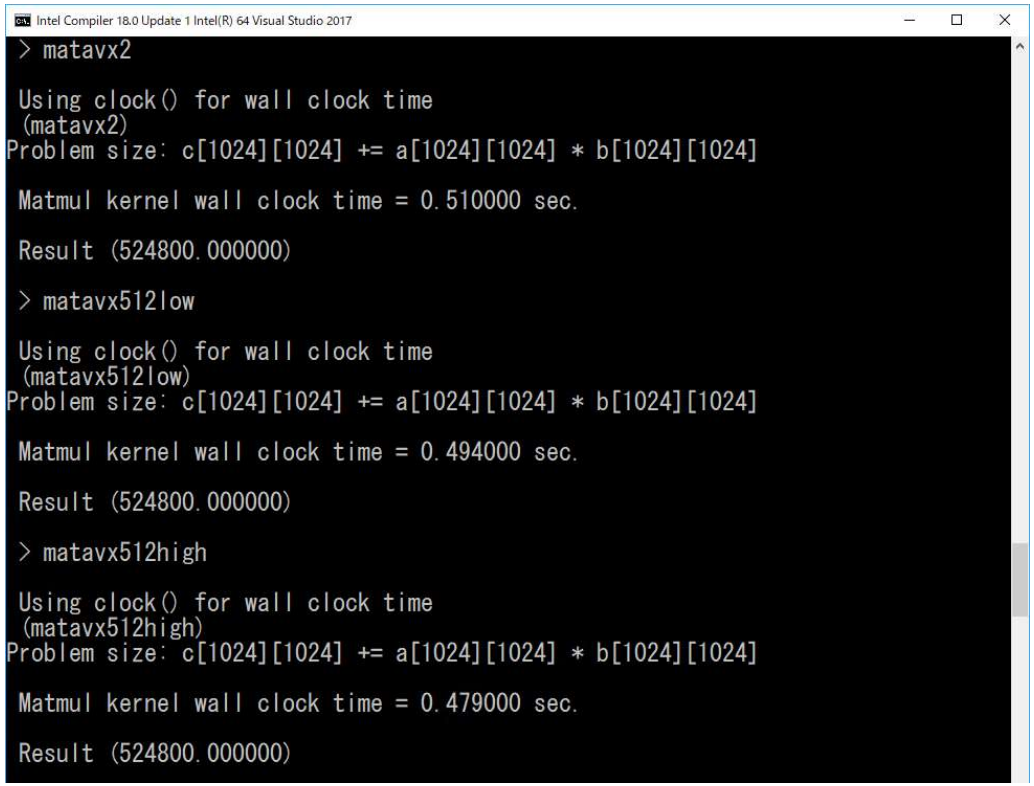
- -xCORE-AVX512 -qopt-zmm-usage=high (Linux* および macOS*)

  /QxCORE-AVX512 /Qopt-zmm-usage:high (Windows*)

GCC コンパイラーでも、まもなく同様の low/high zmm を指定する方法がサポートされる予定です

© 2017 iSUS *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。

メモリー依存のワークロードの例



/QxCORE-AVX2 /tune=skylake-avx512 (Windows*)

/QxCORE-AVX512 /Qopt-zmm-usage:low (Windows*)

/QxCORE-AVX512 /Qopt-zmm-usage:high (Windows*)

© 2017 iSUS *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。

計算集約型ワークロードのテスト例

Intel Compiler 18.0 Update 1 Intel(R) 64 Visual Studio 2017

シリアル

```
> primeavx2
(primeavx2) 1から50000000までの範囲の素数の数は 3001135個。計算時間12.67秒
> primeavx512low
(primeavx512low) 1から50000000までの範囲の素数の数は 3001135個。計算時間13.90秒
> primeavx512high
(primeavx512high) 1から50000000までの範囲の素数の数は 3001135個。計算時間7.65秒
```

Intel Compiler 18.0 Update 1 Intel(R) 64 Visual Studio 2017

OpenMP* 並列化

```
> primeavx2
(primeavx2) 1から50000000までの範囲の素数の数は 3001135個。計算時間2.44秒
> primeavx512low
(primeavx512low) 1から50000000までの範囲の素数の数は 3001135個。計算時間2.39秒
> primeavx512high
(primeavx512high) 1から50000000までの範囲の素数の数は 3001135個。計算時間1.75秒
```

選択Intel Compiler 18.0 Update 1 Intel(R) 64 Visual Studio 2017

```
> piavx2
Data type double
Pi = 3.141593 Time = 0.095000

> piavx512low
Data type double
Pi = 3.141593 Time = 0.093000

> piavx512high
Data type double
Pi = 3.141593 Time = 0.058000
```


© 2017 iSUS *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: アーキテクチャー最適化の推奨事項

推奨事項: アプリケーションの共有およびプライベート・データのサイズを、小さな非インクルーシブ L3 キャッシュと大きな L2 キャッシュに合わせて再度バランスを取ります

推奨事項: L2 キャッシュがアプリケーションの帯域幅要件を満たせば、Skylake Server† マイクロアーキテクチャーでは L2 へのブロック化を検討します

推奨事項: データを共有しない場合、アプリケーションはコアあたりのキャッシュ容量を L3 キャッシュだけでなく L2 と L3 キャッシュサイズの合計であると見なすべきです

推奨事項: データは 64 バイト境界でアライメントします

† 開発コード名

© 2017 iSUS *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:インテル® AVX-512 向けの推奨事項

推奨事項: 実行ターゲットが固定である場合、/Qx (-x) オプションで固有命令セットを生成します

推奨事項: /Qopt-zmm-usage オプションで、ZMM レジスターの使用を制御し、パフォーマンスを計測します。必要に応じて OpenMP* の simdlen、safelen 節を使用します

推奨事項: 実行ターゲットが固定でない場合、/Qax (-ax) オプションを使用してサポートされる最も低い命令セットに合わせてプロセッサー・ディスパッチを行います

推奨事項: FMA ユニット数に対応したプログラミングを行う場合、サポートされる FMA ユニット数を手動で判定します

© 2017 iSUS *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。

組込み関数を使用するガイドライン

- 静的および動的に割り当てられているバッファーを 64 バイトにアライメントします
- 定数向けの補助バッファーサイズを倍にします
- "__mm256_組込み関数名" のプリフィクスを "__mm512_" に変更します
- 変数のデータ型名を __m256 から __m512 へ変更します
- ループの反復回数を半分にします (倍のストライド長に)

アセンブリーを使用するガイドライン

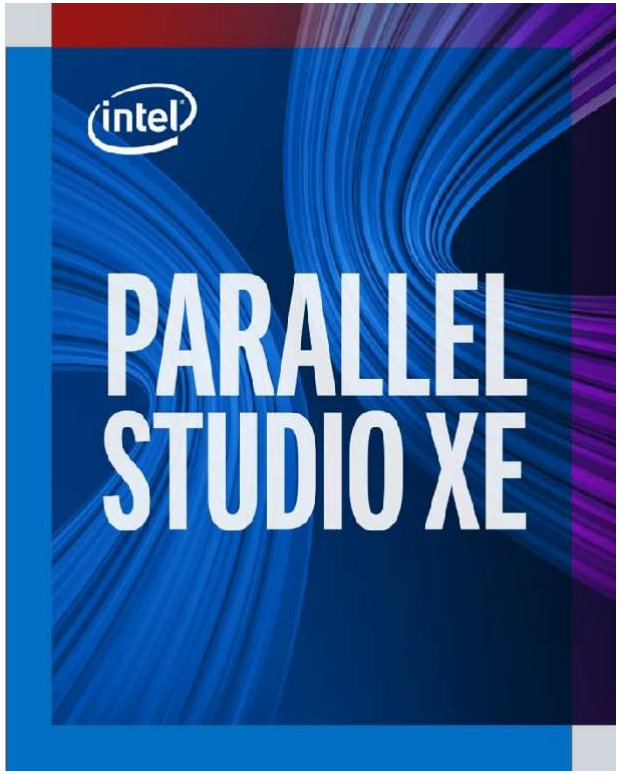
- 静的および動的に割り当てられているバッファーを 64 バイトにアライメントします
- 必要であれば補足のバッファーサイズを 2 倍にします
- 命令ニーモニックの前に “v” プリフィクスを追加します
- レジスター名を ymm から zmm へ変更します
- ループの反復回数を半分にします (またはストライド長を倍にします)

© 2017 iSUS *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。

パフォーマンスに優れたコード

インテル® Parallel Studio XE 2018
30 日間無料評価版をダウンロード:

makebettercode.com/parallelstudioxe-eval (英語) ›



© 2017 iSUS *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。

関連情報

OpenMP* に関する情報

OpenMP* による SIMD を利用するベクトル･プログラミング

http://jp.xlsoft.com/documents/intel/catalog/Intel_ParallelUniverse_Issue22_JPN.pdf

OpenMP* の現在と将来

https://jp.xlsoft.com/documents/intel/magazine/Intel_ParallelUniverse_Issue27_JPN.pdf

インテル® Xeon Phi™ プロセッサー向けにコードを現代化

https://jp.xlsoft.com/documents/intel/magazine/Intel_ParallelUniverse_Issue26_JPN.pdf

コードモダン化のリンク

- モダンコード開発者コミュニティー
  http://www.isus.jp/modern-code/
- インテル® Code Modernization Enablement Program
  https://software.intel.com/code-modernization-enablement (英語)
- インテル® Parallel Computing Centers
  https://software.intel.com/ipcc (英語)
- 技術ウェビナーシリーズ (英語)
  https://software.intel.com/events/development-tools-webinars
- インテル® Parallel Universe マガジン (日本語訳)
  http://www.isus.jp/products/psxe/issue/

© 2017 iSUS *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。

役立つ記事

自動ベクトル化が失敗した場合の対処:
https://www.isus.jp/products/c-compilers/what-to-do-when-auto-vectorization-fails/
ベクトル化によるパフォーマンスの向上:
https://www.isus.jp/products/c-compilers/improve-performance-with-vectorization/
コンパイラー最適化入門: 第 1 回 SIMD 命令とプロセッサーの関係:
https://www.isus.jp/products/c-compilers/compiler_part1/
インテルのベクトル化ツール:
https://www.isus.jp/intel-vectorization-tools/
ケーススタディー: インテル® アドバンスト・ベクトル・エクステンションを使用したブラック-ショールズの計算:
https://www.isus.jp/financial/case-study-computing-black-scholes-with-avx/
インテルの最新の並列化ツールによる明示的なベクトル化:
https://www.isus.jp/products/psxe/vectorization-gets-explicit/
明示的なベクトル・プログラミング – 最も一般的な手法:
https://www.isus.jp/products/psxe/explicit-vector-programming-bkm/
ループのベクトル化によるプログラムの最適化:
https://www.isus.jp/products/c-compilers/program-optimization-through-loop-vectorization/

© 2017 iSUS *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。

お待たせしました

インテル® 64 アーキテクチャーおよび IA-32 アーキテクチャー最適化リファレンス・マニュアル (参考訳版)

インテル® VTune™ Amplifier オンラインヘルプ (iSUS 翻訳版)

完成しました: https://www.isus.jp/ からどうぞ!!

© 2017 iSUS *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。

パフォーマンス検証に使用した環境

プロセッサー
- インテル® Core™ i7-7800X X シリーズ・プロセッサー、
  6 コア、L3 8.53MB、3.50 (4.00) GHz、32GB

- インテル® Core™ i7-5820K プロセッサー、
  6 コア、L3 15MB、3.30 (3.60) GHz、32GB

オペレーティング・システム
- Windows® 10 Creators Fall Edition (Windows® 10 バージョン 1709)

開発環境
- インテル® C++ コンパイラー V18 Update 1

© 2017 iSUS *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。

著作権と商標についておよび最適化に関する注意事項

本資料に掲載されている情報は現状のまま提供され、いかなる保証もいたしません。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を許諾するための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。

© 2017 Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、 Intel Core、Intel Optane、Xeon、Xeon Phi、Cilk、VTune 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。

Microsoft および Windows は、米国 Microsoft Corporation の、米国およびその他の国における登録商標または商標です。

* 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。

最適化に関する注意事項

インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。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。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、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

iSUS