インテル® マイクロアーキテクチャー 開発コード名 Alder Lake 向け OpenMP* アプリケーションの最適化考察

2022 年 1 月

IA Software User Society (iSUS)
編集長　すがわら　きよふみ

内容

- 評価環境
  - インテル® Core™ i7-11700K プロセッサー (Rocket Lake-S†) ベース
  - インテル® Core™ i7-12700K プロセッサー (Alder Lake-S†) ベース
- インテル® スレッド・ディレクターとプログラムのアフィニティー制御
- OpenMP* のアフィニティー制御
- マルチスレッド・アプリケーションの比較と評価
- 命令セットの評価
- まとめ

†開発コード名

はじめに

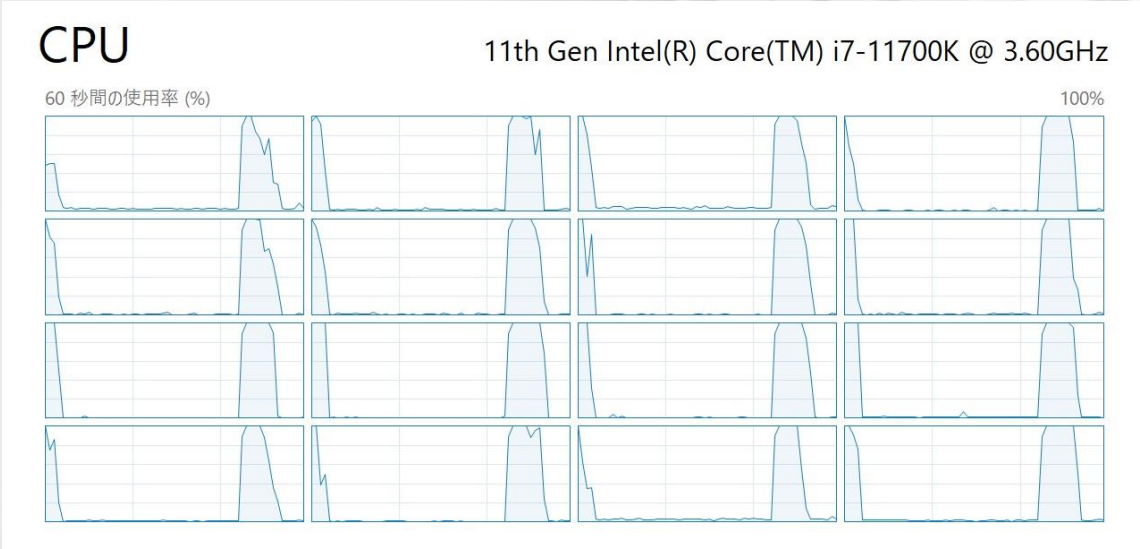
- これまで C/C++ および Fortran で使用されてきた柔軟性と移植性が高い OpenMP* を使用したマルチスレッド・プログラミングに関する知識が、新しいプラットフォームでどのように活用できるか調査します

- ここで紹介するパフォーマンスに関連するデータは、iSUS が独自に収集したものであり、インテル社の公式な見解ではありません

利用する CPU とシステム (1)

CPU の仕様

コアの数	8
スレッド数	16
ターボ・ブースト利用時の最大周波数	5.00 GHz
インテル® ターボ・ブースト・マックス・テクノロジー 3.0 の動作周波数‡	5.00 GHz
インテル® ターボ・ブースト・テクノロジー 2.0 の動作周波数‡	4.90 GHz
プロセッサー ベース動作周波数	3.60 GHz
キャッシュ	16 MB Intel® Smart Cache



Windows* 11 Pro 21H2 (22000.438)

利用する CPU とシステム (2)

CPU の仕様

コアの数	12
# of Performance-cores	8
# of Efficient-cores	4
スレッド数	20
ターボ・ブースト利用時の最大周波数	5.00 GHz
インテル® ターボ・ブースト・マックス・テクノロジー 3.0 の動作周波数‡	5.00 GHz
Performance-core Max Turbo Frequency	4.90 GHz
Efficient-core Max Turbo Frequency	3.80 GHz
Performance-core Base Frequency	3.60 GHz
Efficient-core Base Frequency	2.70 GHz
キャッシュ	25 MB Intel® Smart Cache
Total L2 Cache	12 MB
Processor Base Power	125 W
Maximum Turbo Power	190 W

E コアか?

Windows* 11 Pro 21H2 (22000.438)

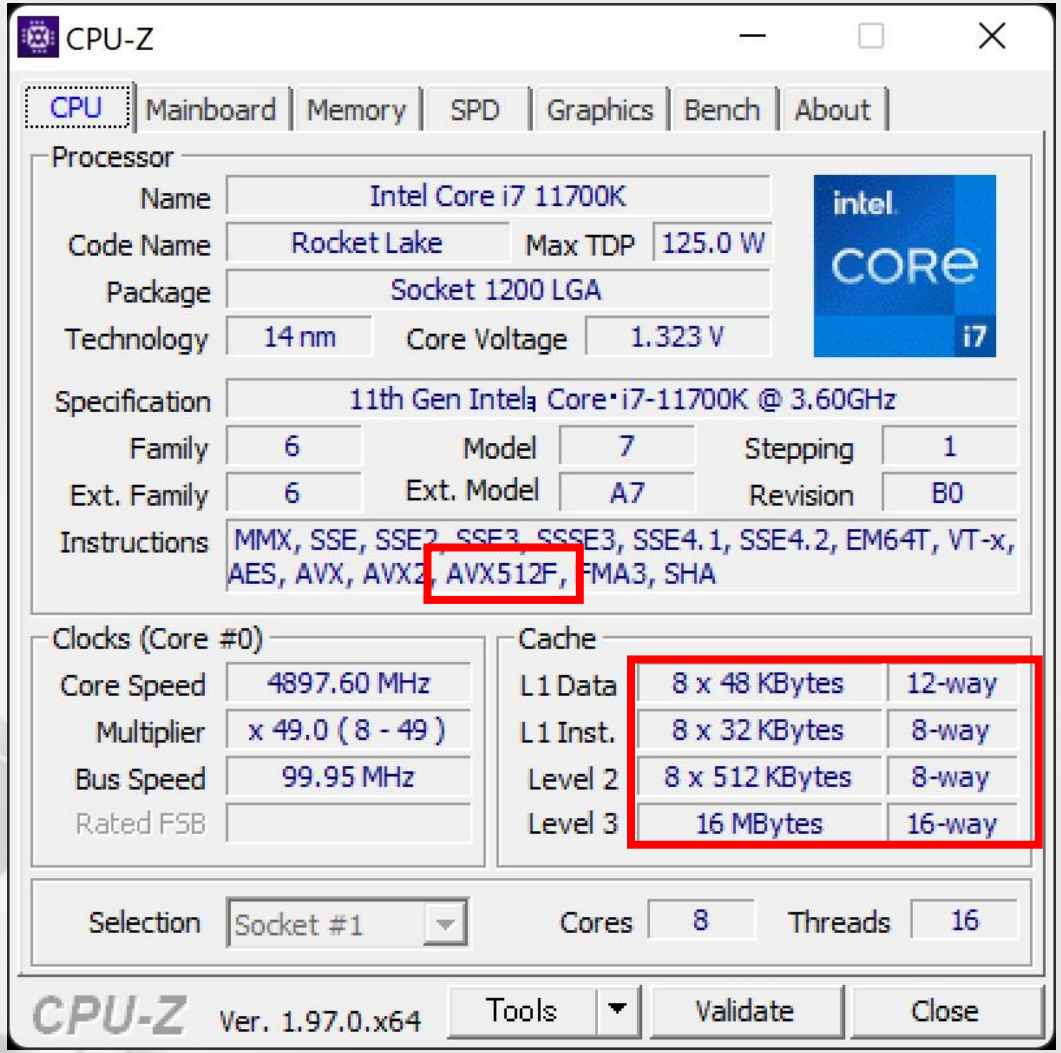
Gen11 と Gen12 の比較

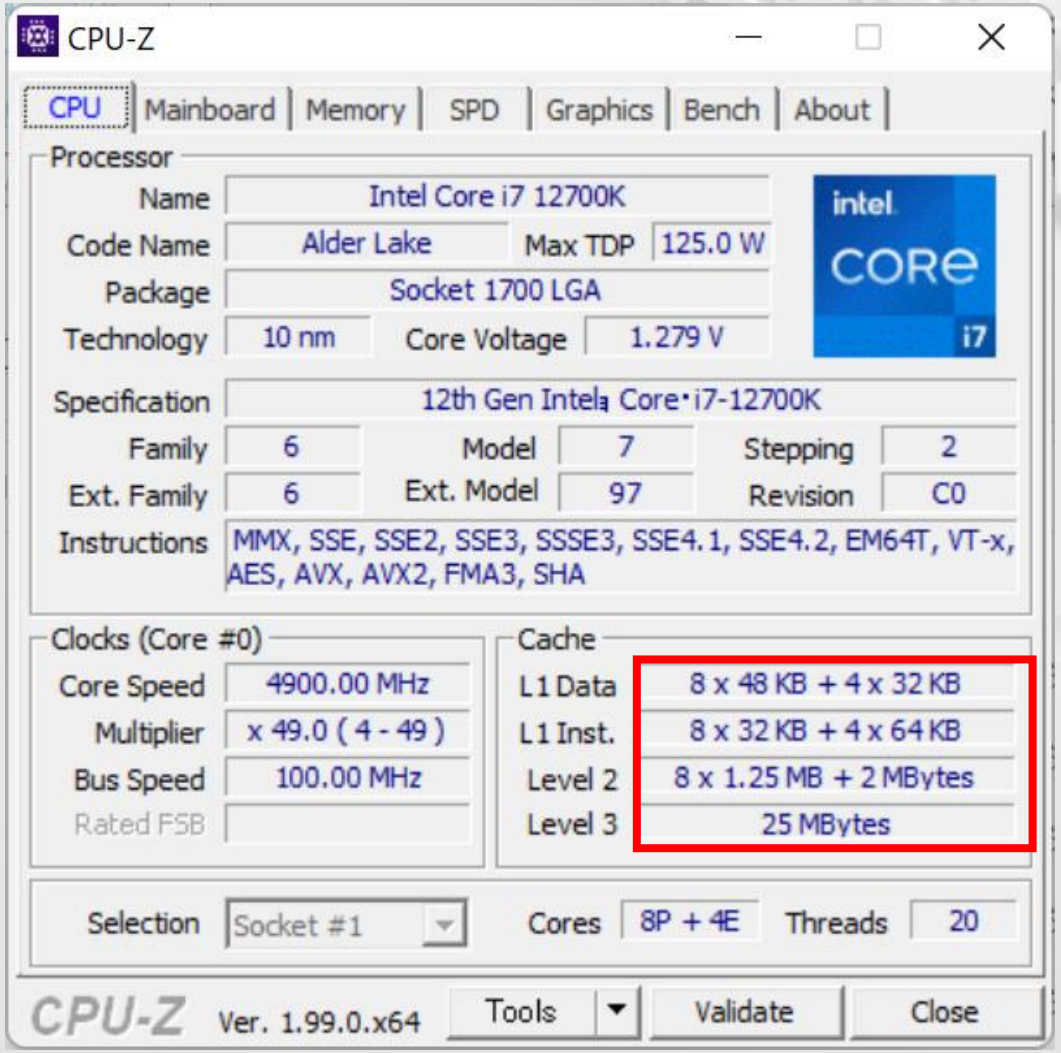
CPU の仕様	
コアの数	8
スレッド数	16
ターボ・ブースト利用時の最大周波数	5.00 GHz
インテル® ターボ・ブースト・マックス・テクノロジー 3.0 の動作周波数‡	5.00 GHz
インテル® ターボ・ブースト・テクノロジー 2.0 の動作周波数‡	4.90 GHz
プロセッサー ベース動作周波数	3.60 GHz
キャッシュ	16 MB Intel® Smart Cache

- インテル® Core™ i7-11700K プロセッサーの従来のコアとインテル® Core™ i7-12700K プロセッサーの P コアの構成はほぼ同じ
- L2 キャッシュと L3 キャッシュの構成が異なる

CPU の仕様	
コアの数	12
# of Performance-cores	8
# of Efficient-cores	4
スレッド数	20
ターボ・ブースト利用時の最大周波数	5.00 GHz
インテル® ターボ・ブースト・マックス・テクノロジー 3.0 の動作周波数‡	5.00 GHz
Performance-core Max Turbo Frequency	4.90 GHz
Efficient-core Max Turbo Frequency	3.80 GHz
Performance-core Base Frequency	3.60 GHz
Efficient-core Base Frequency	2.70 GHz
キャッシュ	25 MB Intel® Smart Cache
Total L2 Cache	12 MB
Processor Base Power	125 W
Maximum Turbo Power	190 W

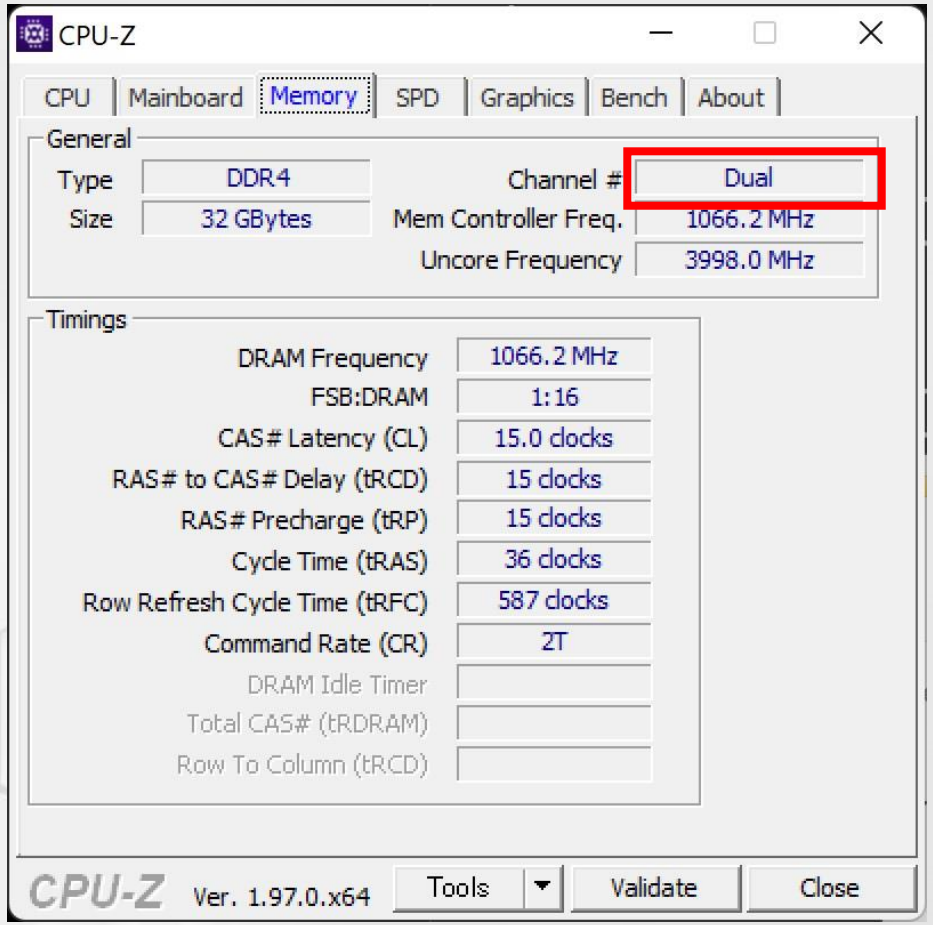
CPU の詳細を確認 ...


CPU-Z
CPU | Mainboard | Memory | SPD | Graphics | Bench | About
Processor
Name Intel Core i7 11700K
Code Name Rocket Lake Max TDP 125.0 W
Package Socket 1200 LGA
Technology 14 nm Core Voltage 1.323 V
intel CORE i7
Specification 11th Gen Intel Core i7-11700K @ 3.60GHz
Family 6 Model 7 Stepping 1
Ext. Family 6 Ext. Model A7 Revision B0
Instructions MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, VT-x, AES, AVX, AVX2, AVX512F, FMA3, SHA
Clocks (Core #0)
Core Speed 4897.60 MHz
Multiplier x 49.0 (8 - 49)
Bus Speed 99.95 MHz
Rated FSB
Cache
L1 Data 8 x 48 KBytes 12-way
L1 Inst. 8 x 32 KBytes 8-way
Level 2 8 x 512 KBytes 8-way
Level 3 16 MBytes 16-way
Selection Socket #1 Cores 8 Threads 16
CPU-Z Ver. 1.97.0.x64 Tools Validate Close

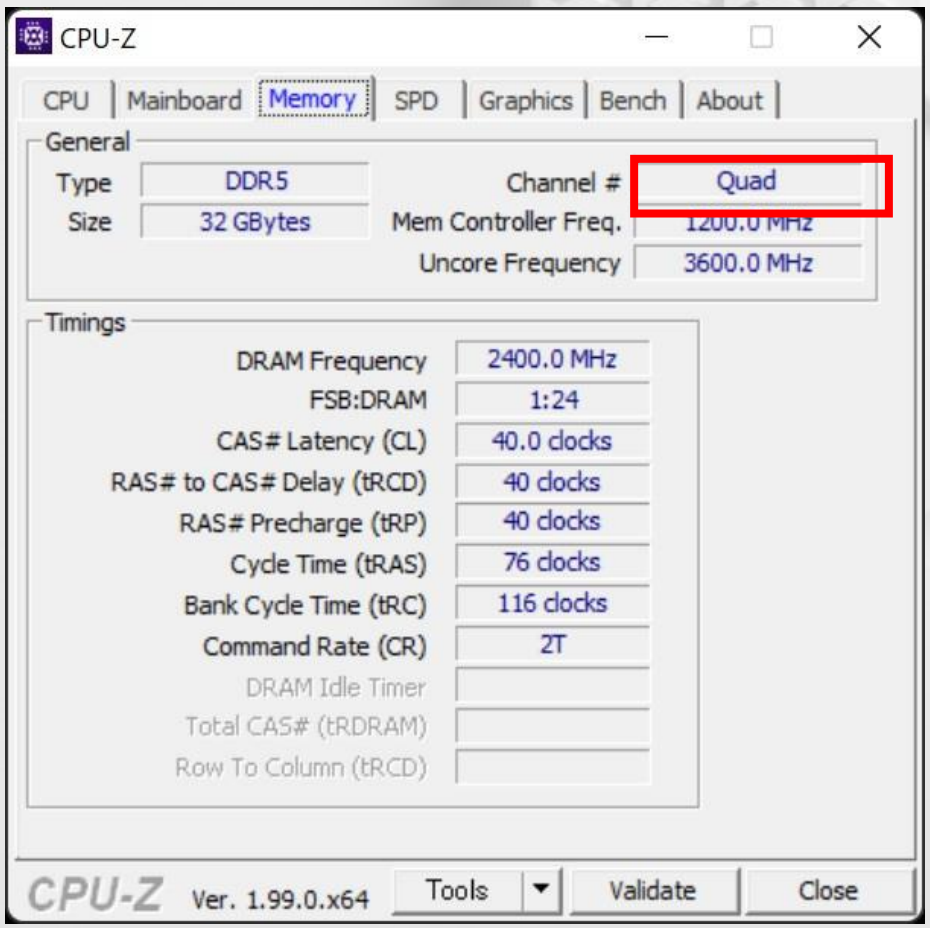

CPU-Z
CPU | Mainboard | Memory | SPD | Graphics | Bench | About
Processor
Name Intel Core i7 12700K
Code Name Alder Lake Max TDP 125.0 W
Package Socket 1700 LGA
Technology 10 nm Core Voltage 1.279 V
intel CORE i7
Specification 12th Gen Intel Core i7-12700K
Family 6 Model 7 Stepping 2
Ext. Family 6 Ext. Model 97 Revision C0
Instructions MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, VT-x, AES, AVX, AVX2, FMA3, SHA
Clocks (Core #0)
Core Speed 4900.00 MHz
Multiplier x 49.0 (4 - 49)
Bus Speed 100.00 MHz
Rated FSB
Cache
L1 Data 8 x 48 KB + 4 x 32 KB
L1 Inst. 8 x 32 KB + 4 x 64 KB
Level 2 8 x 1.25 MB + 2 MBytes
Level 3 25 MBytes
Selection Socket #1 Cores 8P + 4E Threads 20
CPU-Z Ver. 1.99.0.x64 Tools Validate Close

CPU の詳細を確認 … ここでは詳しく調査しません

インテル® Core™ i7-11700K プロセッサー



インテル® Core™ i7-12700K プロセッサー



インテル® スレッド・ディレクターとプログラムのアフィニティー制御

インテル® スレッド・ディレクター (ITD)

Windows* 11 では、インテル® スレッド・ディレクター (ITD) のハードウェア・ヒントを利用して、OS がスレッドを自動的にスケジューリングできます

- インテル® スレッド・ディレクター (ITD) は、オペレーティング・システムに命令セット・アーキテクチャー (ISA) を認識させ、Performance-core と Efficient-core 間のパフォーマンス差 (および電力効率の変化) を認識し、タスクに最も適したコアにスレッドをスケジュールします
- 一般には、強いアフィニティーはアプリケーションと OS 間の決まりごとであるため、避けるように指示されています
- 強いアフィニティーを使用すると、プラットフォームの最適化ができなくなり、OS はインテル® スレッド・ディレクターからのアドバイスを無視することになります。強いアフィニティーは予期しない問題を引き起こす可能性があるため、ミドルウェアが強いスレッド・アフィニティーを使用しているか確認する必要があります。これは、強いスレッド・アフィニティーがアプリケーションによるハードウェアへのアクセスに直接影響するためです
- 強いアフィニティーの問題は、低消費電力デバイスなど Performance-core よりも Efficient-core が多いシステムでは、強いアフィニティーによって OS のスケジュールが制限されるため、特に問題となります

関連する重要なドキュメント

intel

第12世代インテル® Core™ プロセッサー (開発コード名 Alder Lake) パフォーマンス・ハイブリッド・アーキテクチャー向けゲーム開発者ガイド

ホワイトペーパー

2021 年 10 月

リビジョン 1.0

注意事項:

この日本語マニュアルは、インテル コーポレーションのウェブサイトで公開されている『Game Dev Guide for Alder Lake Performance Hybrid Architecture』の参考訳です。

インテル社の許可を得て iSUS (IA Software User Society) が翻訳版を作成した iSUS の著作物です。

原文は Intel Corporation の Copyright であり、日本語参考訳版にも適用されます。

intel

x86 ハイブリッド・アーキテクチャー向けのソフトウェア最適化

ホワイトペーパー

2021年10月

リビジョン 1.0

注意事項:

この日本語マニュアルは、インテル コーポレーションのウェブサイトで公開されている『Optimizing Software for x86 Hybrid Architecture』の参考訳です。
インテル社の許可を得て iSUS (IA Software User Society) が翻訳版を作成した iSUS の著作物です。
原文は Intel Corporation の Copyright であり、日本語参考訳版にも適用されます。

ドキュメント番号: 348851-001JA

https://www.isus.jp/isus-translated-documents/

OpenMP* のスレッド数とアフィニティー制御

- アプリケーションがプログラム内で明示的にスレッド数を指定していない場合、利用可能な最大論理スレッド数がプログラムが利用するスレッド数となります

- 環境変数 OMP_NUM_THREADS で動的にスレッド数を変更できます

- 生成されたスレッドは、環境変数 KMP_AFFINITY (インテル® コンパイラー固有) と OMP_PROC_BIND を設定することでスレッドの配置 (アフィニティー) を制御できます

KMP_AFFINITY と OMP_PROC_BIND

KMP_AFFINITY と OMP_PROC_BIND は使い方が異なります

- KMP_AFFINITY={compact, scatter, none, verbose}
- KMP_AFFINITY=granularity={fine,core}{,compact, scatter, balanced, verbose}

https://www.isus.jp/products/c-compilers/thread-affinity-control/

- OMP_PROC_BIND={primary, spread, close}
  - *close*: 共有キャッシュの利点を得るため、スレッドを同じソケット上の連続するコアに割り当てます
  - *spread*: すべてのコアに少なくとも 1 つのスレッドが割り当てられるまで、スレッドを別々のコアに配置します
  - *primary*: プライマリースレッドと同じ場所に割り当てます

https://www.isus.jp/products/c-compilers/openmp-ref-5-2-1121-released/

OpenMP* アプリケーションのアフィニティー制御を行うプログラムはどのような影響を受けるのか

キャッシュとスレッドのトポロジーを確認

Intel(r) oneAPI Tools - prime256

```
C> prime256
OMP: Info #155: KMP_AFFINITY: Initial OS proc set respected: 0-15
OMP: Info #217: KMP_AFFINITY: decoding x2APIC ids.
OMP: Info #157: KMP_AFFINITY: 16 available OS procs
OMP: Info #158: KMP_AFFINITY: Uniform topology
OMP: Info #288: KMP_AFFINITY: topology layer "LL cache" is equivalent to "socket".
OMP: Info #288: KMP_AFFINITY: topology layer "L3 cache" is equivalent to "socket".
OMP: Info #288: KMP_AFFINITY: topology layer "L2 cache" is equivalent to "core".
OMP: Info #288: KMP_AFFINITY: topology layer "L1 cache" is equivalent to "core".
OMP: Info #192: KMP_AFFINITY: 1 socket x 8 cores/socket x 2 threads/core (8 total cores)
OMP: Info #219: KMP_AFFINITY: OS proc to physical thread map:
OMP: Info #172: KMP_AFFINITY: OS proc 0 maps to socket 0 core 0 thread 0
OMP: Info #172: KMP_AFFINITY: OS proc 1 maps to socket 0 core 0 thread 1
OMP: Info #172: KMP_AFFINITY: OS proc 2 maps to socket 0 core 1 thread 0
OMP: Info #172: KMP_AFFINITY: OS proc 3 maps to socket 0 core 1 thread 1
OMP: Info #172: KMP_AFFINITY: OS proc 4 maps to socket 0 core 2 thread 0
OMP: Info #172: KMP_AFFINITY: OS proc 5 maps to socket 0 core 2 thread 1
OMP: Info #172: KMP_AFFINITY: OS proc 6 maps to socket 0 core 3 thread 0
OMP: Info #172: KMP_AFFINITY: OS proc 7 maps to socket 0 core 3 thread 1
OMP: Info #172: KMP_AFFINITY: OS proc 8 maps to socket 0 core 4 thread 0
OMP: Info #172: KMP_AFFINITY: OS proc 9 maps to socket 0 core 4 thread 1
OMP: Info #172: KMP_AFFINITY: OS proc 10 maps to socket 0 core 5 thread 0
OMP: Info #172: KMP_AFFINITY: OS proc 11 maps to socket 0 core 5 thread 1
OMP: Info #172: KMP_AFFINITY: OS proc 12 maps to socket 0 core 6 thread 0
OMP: Info #172: KMP_AFFINITY: OS proc 13 maps to socket 0 core 6 thread 1
OMP: Info #172: KMP_AFFINITY: OS proc 14 maps to socket 0 core 7 thread 0
OMP: Info #172: KMP_AFFINITY: OS proc 15 maps to socket 0 core 7 thread 1
OMP: Info #255: KMP_AFFINITY: pid 6260 tid 8868 thread 0 bound to OS proc set 0-15
```

Intel(r) oneAPI Tools - prime

```
C> set OMP_NUM_THREADS=1

C> set KMP_AFFINITY=none,verbose

C> prime
  0%OMP: Info #155: KMP_AFFINITY: Initial OS proc set respected: 0-19
OMP: Info #217: KMP_AFFINITY: decoding x2APIC ids.
OMP: Info #157: KMP_AFFINITY: 20 available OS procs
OMP: Info #159: KMP_AFFINITY: Nonuniform topology
OMP: Info #288: KMP_AFFINITY: topology layer "LL cache" is equivalent to "socket".
OMP: Info #288: KMP_AFFINITY: topology layer "L3 cache" is equivalent to "socket".
OMP: Info #288: KMP_AFFINITY: topology layer "L1 cache" is equivalent to "core".
OMP: Info #192: KMP_AFFINITY: 1 socket x 12 cores/socket x 2 threads/core (12 total cores)
OMP: Info #194: KMP_AFFINITY: Intel(R) Hybrid Technology core type detected: 8 Intel(R) Core(TM) processor cores.
OMP: Info #195: KMP_AFFINITY:   8 with core efficiency 1.
OMP: Info #194: KMP_AFFINITY: Intel(R) Hybrid Technology core type detected: 4 Intel Atom(R) processor cores.
OMP: Info #195: KMP_AFFINITY:   4 with core efficiency 0.
OMP: Info #219: KMP_AFFINITY: OS proc to physical thread map:
OMP: Info #172: KMP_AFFINITY: OS proc 0 maps to socket 0 core 0 thread 0 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 1 maps to socket 0 core 0 thread 1 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 2 maps to socket 0 core 4 thread 0 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 3 maps to socket 0 core 4 thread 1 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 4 maps to socket 0 core 8 thread 0 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 5 maps to socket 0 core 8 thread 1 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 6 maps to socket 0 core 12 thread 0 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 7 maps to socket 0 core 12 thread 1 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 8 maps to socket 0 core 16 thread 0 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 9 maps to socket 0 core 16 thread 1 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 10 maps to socket 0 core 20 thread 0 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 11 maps to socket 0 core 20 thread 1 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 12 maps to socket 0 core 24 thread 0 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 13 maps to socket 0 core 24 thread 1 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 14 maps to socket 0 core 28 thread 0 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 15 maps to socket 0 core 28 thread 1 (Intel(R) Core(TM) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 16 maps to socket 0 core 36 thread 0 (Intel Atom(R) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 17 maps to socket 0 core 37 thread 0 (Intel Atom(R) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 18 maps to socket 0 core 38 thread 0 (Intel Atom(R) processor)
OMP: Info #172: KMP_AFFINITY: OS proc 19 maps to socket 0 core 39 thread 0 (Intel Atom(R) processor)
OMP: Info #255: KMP_AFFINITY: pid 24616 tid 14660 thread 0 bound to OS proc set 0-19
 50%
```

- KMP_AFFINITY に verbose オプションを指定すると、実行環境とアフィニティーに関連する情報を表示できます
- OpenMP* スレッドとランタイムからはコアのタイプを識別できません

どのようなケースにアフィニティーを制御するのか

1. 複数ソケットのプラットフォームで、アプリケーションを 1 つのソケットのみで実行したい、またはソケット全体で実行したい
2. ハイパースレッディングが有効なシステムで、1 物理コアに 1 論理スレッドを割り当てたい、または 1 物理コアに 2 論理スレッドを割り当てたい

デスクトップやモバイル環境でもありうるケース:

- アプリケーションが使用するスレッド数を明示的に制御している (2、4、8 など)
- ハイパースレッディングが有効なシステムでパフォーマンスが思ったようにスケールしない (ハイパースレッディング固有の制限の影響を受ける)

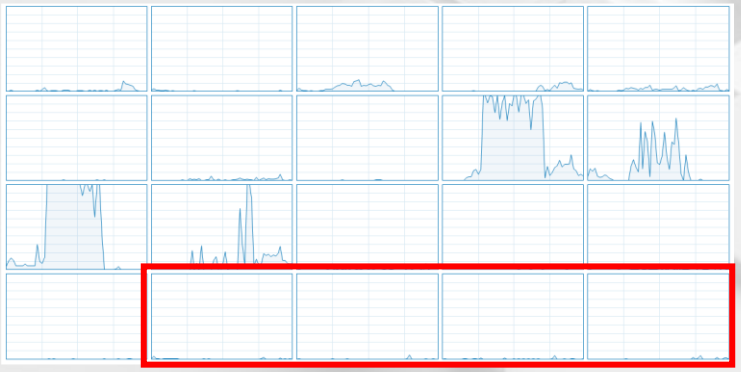
https://www.isus.jp/others/insights-to-hyper-threading/

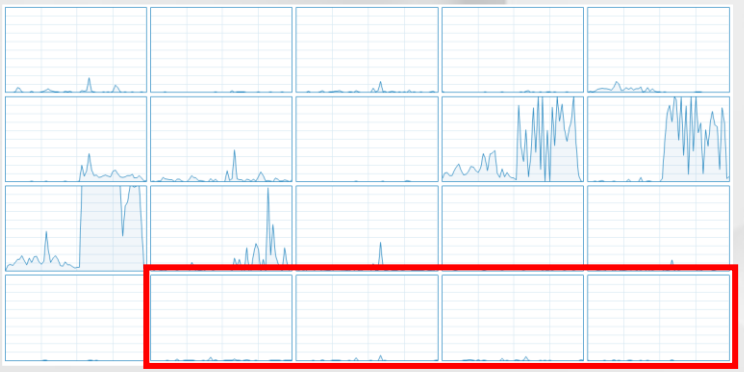
スレッド数が 1 の場合

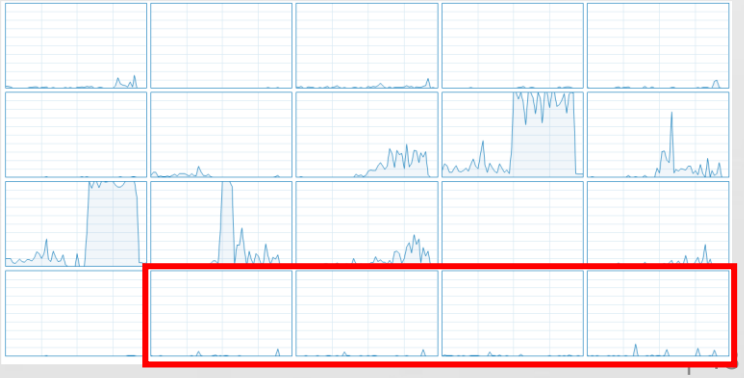
Intel(r) oneAPI Tools

```
C> set OMP_NUM_THREADS=1

C> prime
100%
     1から100000000までの範囲の素数の数は 5761456個。計算時間  25.66秒

C> prime
100%
     1から100000000までの範囲の素数の数は 5761456個。計算時間  25.61秒

C> prime
100%
     1から100000000までの範囲の素数の数は 5761456個。計算時間  26.24秒
```

インテル® Core™ i7-12700K プロセッサーで実行







- E コアには全くスレッドが割り当てられていない

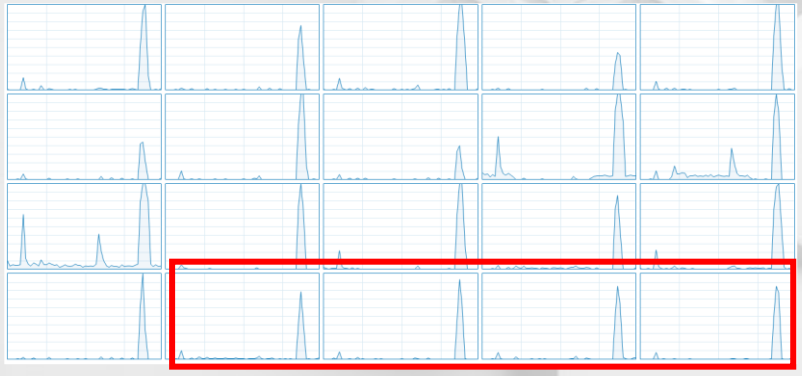
iSUS

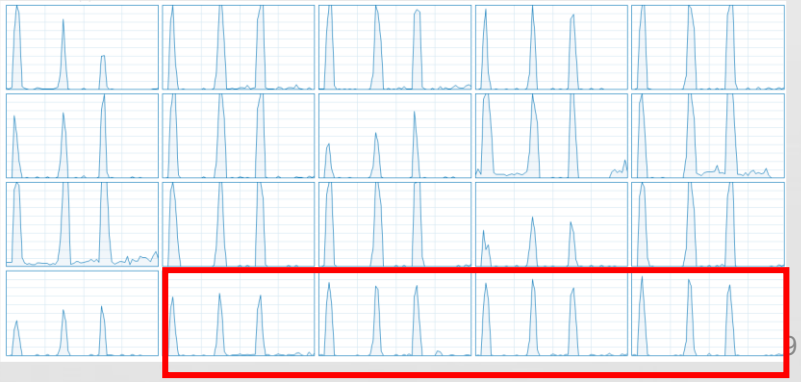
スレッド数が 16 の場合

Intel(r) oneAPI Tools

```
C> prime
   1から100000000までの範囲の素数の数は 5761456個。計算時間   3.18秒

C> prime
   1から100000000までの範囲の素数の数は 5761456個。計算時間   3.15秒

C> prime
   1から100000000までの範囲の素数の数は 5761456個。計算時間   3.19秒
```

インテル® Core™ i7-12700K プロセッサーで実行





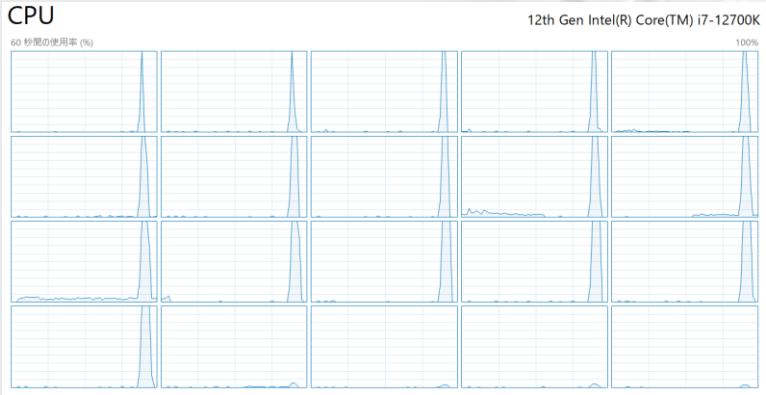
- スレッドは、P コアと E コアに割り当てられている

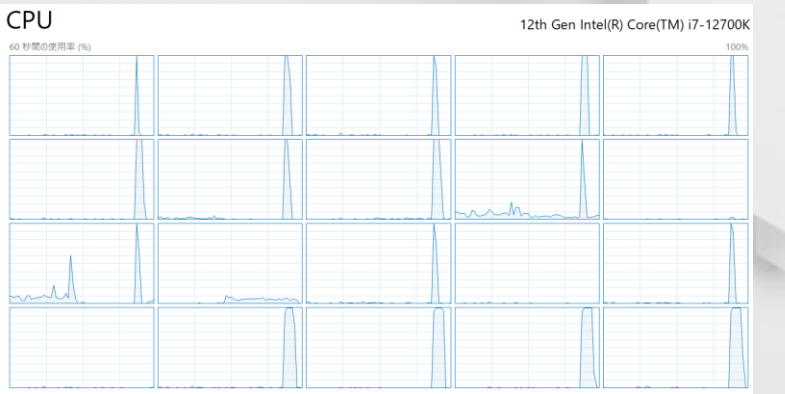
アフィニティーの影響

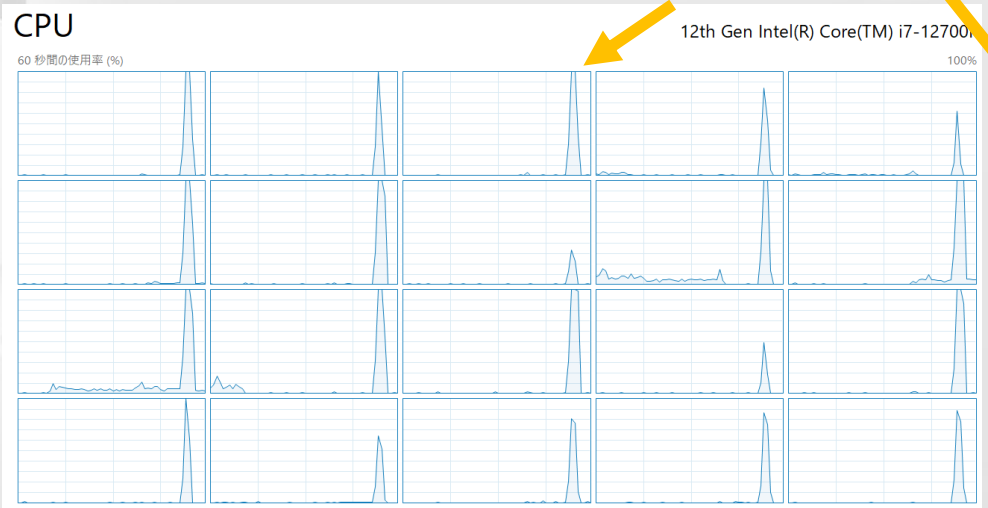
インテル® Core™ i7-12700K プロセッサーで実行

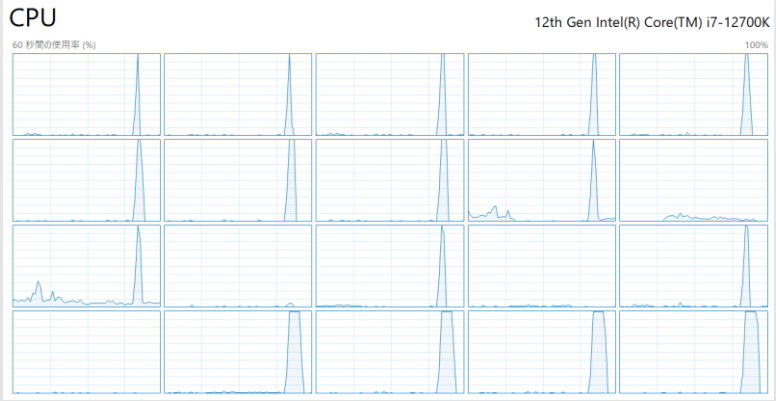
Intel(r) oneAPI Tools

```
C> set OMP_NUM_THREADS=16

C> set KMP_AFFINITY=granularity=fine,compact

C> prime256
    1から100000000までの範囲の素数の数は 5761456個。計算時間    3.13秒

C> set KMP_AFFINITY=granularity=fine,scatter

C> prime256
    1から100000000までの範囲の素数の数は 5761456個。計算時間    5.29秒

C> set KMP_AFFINITY=granularity=fine,balanced

C> prime256
    1から100000000までの範囲の素数の数は 5761456個。計算時間    6.00秒

C> set KMP_AFFINITY=

C> prime256
    1から100000000までの範囲の素数の数は 5761456個。計算時間    2.41秒
```









アフィニティーとパフォーマンスの傾向

インテル® Core™ i7-11700K プロセッサー

```
Intel(r) oneAPI Tools

C> set OMP_NUM_THREADS=8

C> prime256

    1から100000000までの範囲の素数の数は 5761456個。計算時間    4.21秒

C> set KMP_AFFINITY=compact

C> prime256

    1から100000000までの範囲の素数の数は 5761456個。計算時間    6.91秒

C> set KMP_AFFINITY=scatter

C> prime256

    1から100000000までの範囲の素数の数は 5761456個。計算時間    3.91秒
```

インテル® Core™ i7-12700K プロセッサー

```
Intel(r) oneAPI Tools

C> set OMP_NUM_THREADS=8

C> prime256
    1から100000000までの範囲の素数の数は 5761456個。計算時間    3.32秒

C> set KMP_AFFINITY=compact

C> prime256
    1から100000000までの範囲の素数の数は 5761456個。計算時間    5.98秒

C> set KMP_AFFINITY=scatter

C> prime256
    1から100000000までの範囲の素数の数は 5761456個。計算時間    3.25秒
```

スレッド数が論理コア数の半分であれば E コアの影響はほとんどない

スレッド数とアフィニティーの影響

Intel(r) oneAPI Tools

```
C> set OMP_NUM_THREADS=10

C> set KMP_AFFINITY=

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    2.88秒

C> set KMP_AFFINITY=compact

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    4.89秒

C> set KMP_AFFINITY=scatter

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    9.51秒
```

Intel(r) oneAPI Tools

```
C> set OMP_NUM_THREADS=12

C> set KMP_AFFINITY=

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    2.54秒

C> set KMP_AFFINITY=compact

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    4.12秒

C> set KMP_AFFINITY=scatter

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    7.96秒
```

Intel(r) oneAPI Tools

```
C> set OMP_NUM_THREADS=14

C> set KMP_AFFINITY=

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    2.43秒

C> set KMP_AFFINITY=compact

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    3.56秒

C> set KMP_AFFINITY=scatter

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    6.38秒
```

Intel(r) oneAPI Tools

```
C> set OMP_NUM_THREADS=16

C> set KMP_AFFINITY=

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    2.38秒

C> set KMP_AFFINITY=compact

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    3.14秒

C> set KMP_AFFINITY=scatter

C> prime256
     1から100000000までの範囲の素数の数は 5761456個。計算時間    5.28秒
```

インテル® Core™ i7-12700K プロセッサーで実行

スレッド数とアフィニティーの影響

```
Intel(r) oneAPI Tools
C> set OMP_NUM_THREADS=10
C> set KMP_AFFINITY=
C> prime256
   1から100000000までの範
C> set KMP_AFFINITY=compac
C> prime256
   1から100000000までの範
C> set KMP_AFFINITY=scatte
C> prime256
   1から100000000までの範
```

```
Intel(r) oneAPI Tools
C> set OMP_NUM_THREADS=12
C> set KMP_AFFINITY=
                                                    時間    2.54秒
                                                    時間    4.12秒
                                                    時間    7.96秒
```

```
Intel(r) oneAPI Tools
C> set OMP_NUM_THREADS=14
C> set KMP_AFFINITY=
C> prime256
   1から100000000までの範
C> set KMP_AFFINITY=compac
C> prime256
   1から100000000までの範
C> set KMP_AFFINITY=scatter
C> prime256
   1から100000000までの範囲の素数の数は 5761456個。計算時間    6.38秒
```

```
Intel(r) oneAPI Tools
C> set OMP_NUM_THREADS=
C> set KMP_AFFINITY=
C> prime256
   1から100000000までの範囲の素数の数は 5761456個。計算時間    2.20秒
C> set KMP_AFFINITY=compact
C> prime256
   1から100000000までの範囲の素数の数は 5761456個。計算時間    4.82秒
C> set KMP_AFFINITY=scatter
C> prime256
   1から100000000までの範囲の素数の数は 5761456個。計算時間    3.85秒
```

```
                                                    算時間    2.38秒
   1から100000000までの範囲の素数の数は 5761456個。計算時間    3.14秒
C> set KMP_AFFINITY=scatter
C> prime256
   1から100000000までの範囲の素数の数は 5761456個。計算時間    5.28秒
```

インテル® Core™ i7-12700K プロセッサーで実行

ここまでのまとめ

- プロセッサーの論理コア数の半分のスレッドを生成する、メモリーやキャッシュの利用率が低い計算主体型のアプリケーションは、ハイパースレッディングが有効な従来のプロセッサーコアでは、compact アフィニティーよりも scatter アフィニティーのほうが高いパフォーマンスを発揮する傾向がありました

- インテル® Core™ i7-12700K プロセッサー (開発コード名 Alder Lake マイクロアーキテクチャー) では、計算集約型のスレッドが E コアに固定割り当てされると、パフォーマンスを損ねるインバランス状態となることがあります

- インテル様がおっしゃるとおり … 余計なことはするなという結論です

命令セット

インテル® Core™ i7-11700K プロセッサーと インテル® Core™ i7-12700K プロセッサーを同じ条件で使用

16 スレッドのアプリケーションを Alder Lake† の E コアを使用せずに実行

```
Intel(r) oneAPI Tools

C> set OMP_NUM_THREADS=16

C> set KMP_AFFINITY=compact

C> prime256
    1から100000000までの範囲の素数の数は 5761456個。計算時間    3.13秒
```

16 スレッドのアプリケーションを Rocket Lake-S† で実行

```
Intel(r) oneAPI Tools

C> prime256

    1から100000000までの範囲の素数の数は 5761456個。計算時間    3.31秒
```

†開発コード名

インテル® Core™ i7-11700K プロセッサーと インテル® Core™ i7-12700K プロセッサーを同じ条件で使用

16 スレッドのアプリケーションを Alder Lake† の E コアを使用せずに実行

```
Intel(r) oneAPI Tools

C> set OMP_NUM_THREADS=

C> set KMP_AFFINITY=

C> prime256
    1から100000000までの範囲の素数の数は 5761456個。計算時間    2.23秒
```

20 スレッドを起動して
E コアも活用

16 スレッドのアプリケーションを Rocket Lake-S† で実行

```
Intel(r) oneAPI Tools

C> prime256

    1から100000000までの範囲の素数の数は 5761456個。計算時間    3.31秒
```

```
C> prime512zmm

    1から100000000までの範囲の素数の数は 5761456個。計算時間    2.31秒
```

†開発コード名

ターゲット・アーキテクチャー向けの最適化

Intel(r) oneAPI Tools

```
C> icl prime.cpp /Qopenmp -nologo
prime.cpp

C> prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間    2.85秒

C> icl prime.cpp /Qopenmp /QxAVX2 -nologo
prime.cpp

C> prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間    2.25秒

C> icl prime.cpp /Qopenmp /Qxalderlake -nologo
prime.cpp

C> prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間    2.18秒
```

インテル® SSE2

インテル® AVX2

インテル® AVX2 + α

Alder Lake-S† でコンパイルして実行

†開発コード名

/Qx (-x) alderlake オプション

Alder Lake† マイクロアーキテクチャー・ベースのプラットフォームでは、通常インテル® AVX-512 命令セットは利用できませんが、インテル® コンパイラー・クラシックや LLVM ベースの新しいインテル® DPC++/C++ コンパイラーには alderlake ターゲットオプションが用意されています

```
Intel(r) oneAPI Tools

C> icl prime.cpp /Qopenmp /Qxalderlake -nologo
prime.cpp

C>  prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間    3.24秒

C> icx prime.cpp /Qopenmp /Qxalderlake -W0 -nologo

C>  prime

Please verify that both the operating system and the processor support Intel(R)
SHSTK instructions.
```

Rocket Lake-S† でコンパイルして実行

icl と icx で生成されるプロセッサーの自動判定ロジックが異なる…

インテル® AVX2 ベースのバイナリーが生成されますが、下位互換性がない命令セットが含まれるので注意してください

†開発コード名

シングルスレッドでの命令セットによる差

```
Intel(r) oneAPI Tools

C> icl prime.cpp /Qipo /O2 /QxAVX2 -nologo /WO
prime.cpp

C> prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間  18.29秒

C> prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間  18.25秒

C> prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間  18.25秒
```

インテル® Core™ i7-12700K プロセッサー

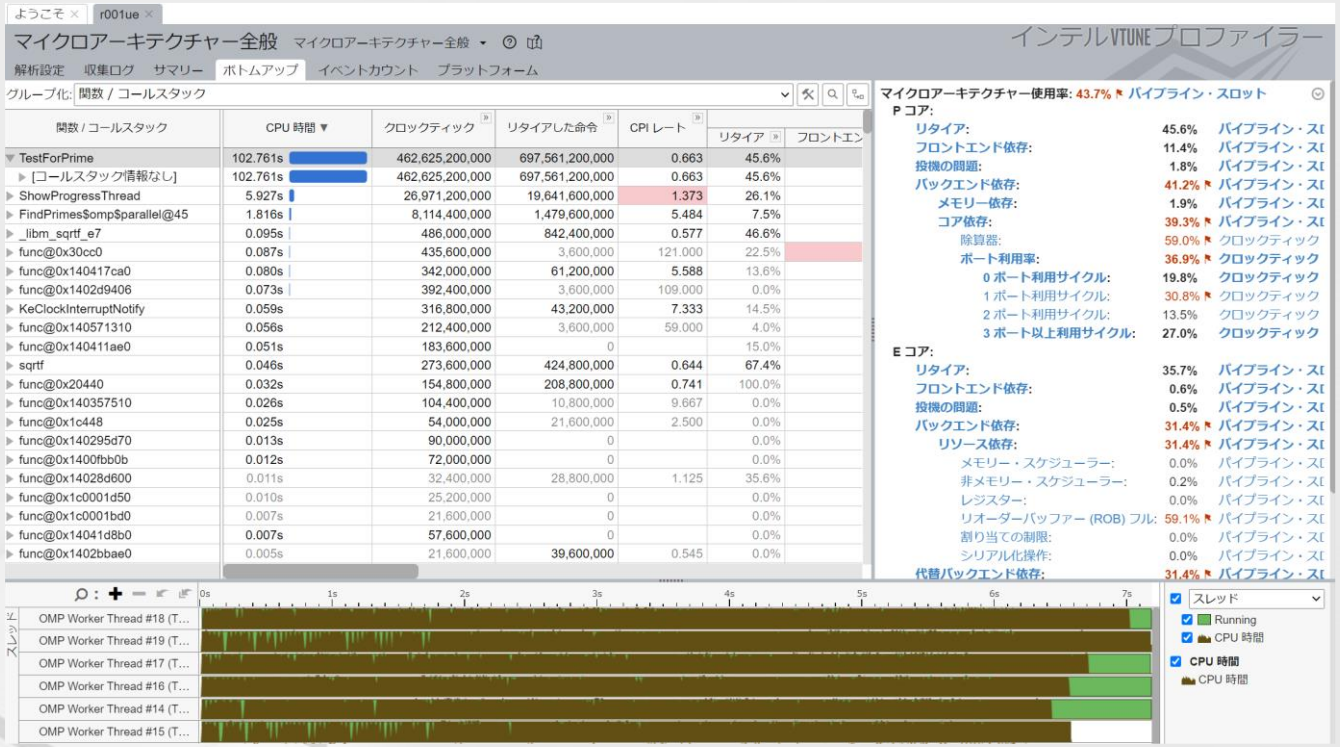
インテル® AVX2 命令セットを実行

```
Intel(r) oneAPI Tools

C> icl prime.cpp /Qipo /O2 /QxCOMMON-AVX512 /Qopt-zmm-usage=high -nologo -WO
prime.cpp

C> prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間  15.47秒

C> prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間  14.84秒

C> prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間  14.91秒
```

インテル® Core™ i7-11700K プロセッサー

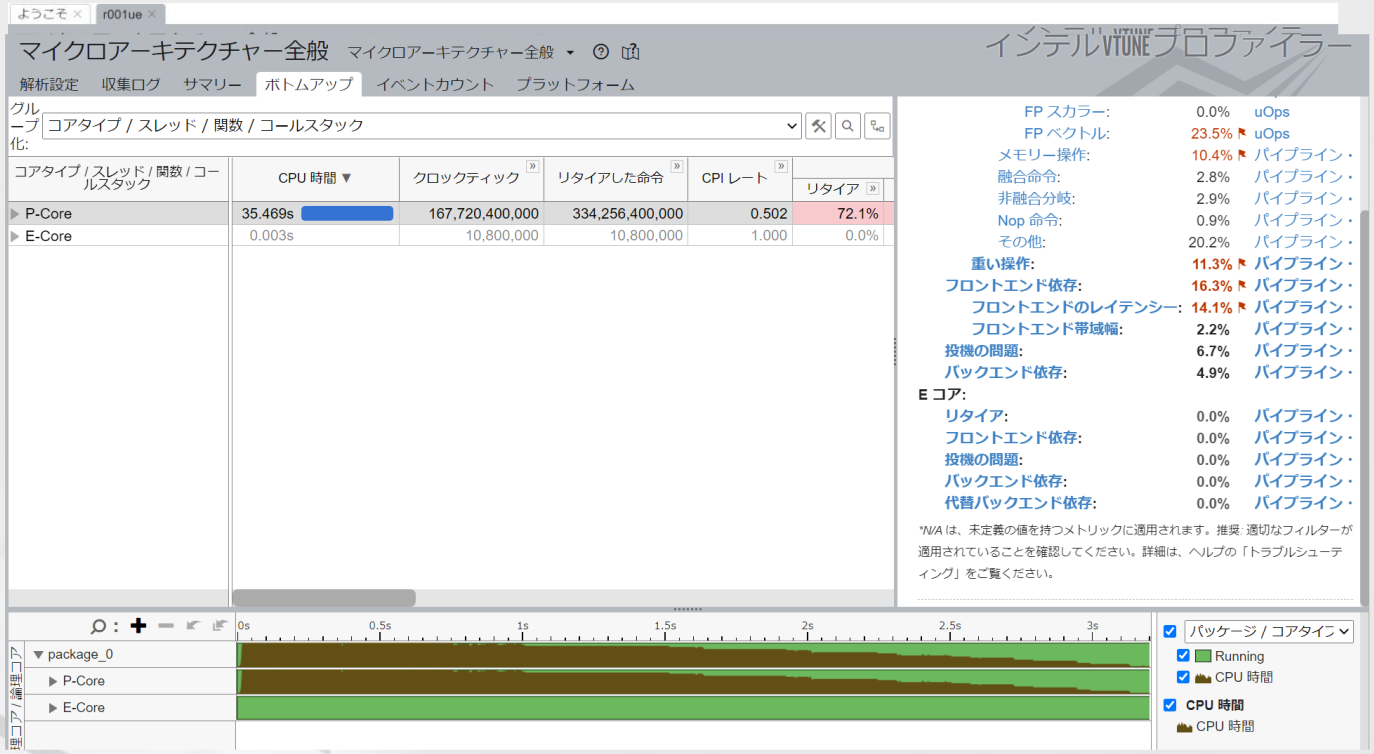
インテル® AVX-512 命令セットを実行

インテル® VTune™ プロファイラーのサポート

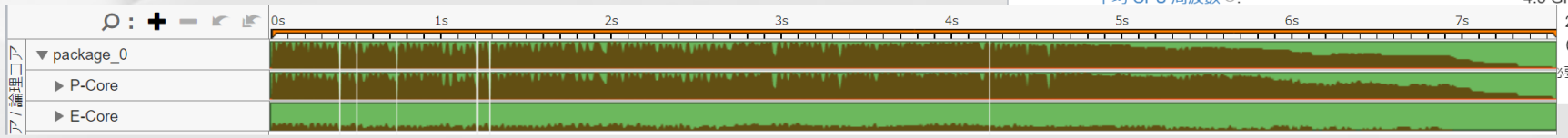


インテル® VTune™ プロファイラー 2022 のマイクロアーキテクチャー全般解析では、P コアと E コアのパフォーマンス・メトリックを識別できます

インテル® VTune™ プロファイラーのサポート



インテル® VTune™ プロファイラー 2022 のマイクロアーキテクチャー全般解析では、P コアと E コアのパフォーマンス・メトリックを識別できます



まとめ

- Alder Lake[†] マイクロアーキテクチャー・ベースのプラットフォームで、マルチスレッド・アプリケーションのパフォーマンスの問題を解決するためアフィニティーの制御を行う場合、十分に注意して評価を行ってください
- インテル非推奨の方法でインテル® AVX-512 命令セットを有効にする場合、インテル® AVX2 を使用して P コアと E コアを活用する場合と十分な比較評価を行ってください
- ここで使用したサンプルは、メモリーをほとんど使用しません。メモリー集約型のアプリケーションでは、前世代のプラットフォーム間との比較で異なる傾向の結果が示される可能性があることに注意してください

[†]開発コード名

おまけ:ランタイムのプロセッサー識別の問題

インテル® C++ コンパイラー・クラシック 2022 でコンパイルして実行

```
Intel(r) oneAPI Tools

C> icl prime.cpp /Qopenmp /Qxalderlake -nologo
prime.cpp

C> prime
     1から100000000までの範囲の素数の数は 5761456個。計算時間    2.18秒

C> icl prime.cpp /Qopenmp /Qxtigerlake -nologo
prime.cpp

C> prime
     0から      0までの範囲の素数の数は16843008個。計算時間    0.00秒
-1759889440から558887470までの範囲の素数の数は        46個。計算時間    0.00秒
     0から      0までの範囲の素数の数は         0個。計算時間    0.00秒
```

インテル® C++ コンパイラー・クラッシックで生成されるランタイムのプロセッサー判定には問題がある

LLVM ベースの新世代インテル® C++ コンパイラー 2022 でコンパイルして実行

```
Intel(r) oneAPI Tools

C> icx prime.cpp /Qopenmp /Qxtigerlake -nologo /WO

C> prime

Please verify that both the operating system and the processor support Intel(R) AVX512F, AVX512DQ, AVX512CD, AVX512BW, A
VX512VL, AVX512VBMI, AVX512_VPOPCNTDQ, AVX512_BITALG, AVX512_VBMI2 and AVX512_VNNI instructions.
```

参考資料

- 第 12 世代インテル® Core™ プロセッサー (開発コード名 Alder Lake) パフォーマンス・ハイブリッド・アーキテクチャー向けゲーム開発者ガイド

  開発コード名 Alder Lake パフォーマンス・ハイブリッド・アーキテクチャーを活用するゲーム最適化向けのアーキテクチャーの概要とベスト・プラクティスを提供します

- x86 ハイブリッド・アーキテクチャー向けのソフトウェア最適化ホワイトペーパー日本語版

  ハイブリッド・アーキテクチャーをサポートするインテル® Core™ プロセッサー向けにソフトウェアを最適化する情報を提供します

- OpenMP* 5.2 API シンタックス・クイック・リファレンス・カード

- OpenMP* 5.2 仕様の概略

iSUS

ソフトウェア・セミナー