

The background is a solid dark blue. It features a large, stylized graphic on the left side that resembles a spiral or a stylized 'C' shape, composed of white and light blue binary digits (0s and 1s). The digits are arranged in concentric, slightly offset rings, creating a sense of depth and motion.

# インテル® VTune™ Amplifier XE を使用したストレージ向けの パフォーマンス最適化

2016 年 10 月 12 日  
Day2トラック D-2 (14:55 – 15:40)  
すがわら きよふみ  
iSUS 編集長

# 本日の内容

- インテル® VTune™ Amplifier XE 2017 概要
- ストレージ解析向けのインテル® VTune™ Amplifier XE の新機能
- メモリー解析向けのインテル® VTune™ Amplifier XE の新機能

# インテル® VTune™ Amplifier XE

## 高速でスケーラブルなコードを迅速に開発

- 必要なデータを取得
  - hotspot (統計コールツリー)、呼び出しカウント (統計)
  - スレッド・プロファイル - コンカレンシー解析およびロックと待機の解析
  - キャッシュミス、帯域幅解析<sup>1</sup>
  - GPU オフロードと OpenCL\* カーネルトレース
- 必要な情報を素早く表示
  - ソース/アセンブリで結果を表示
  - OpenMP\* のスケーラビリティ解析、グラフィカル・フレーム解析
  - ビューポイントでデータをフィルターして関係のないデータを非表示
  - スレッドおよびタスク・アクティビティをタイムライン表示
- 簡単に使用可能
  - 特別なコンパイラーは不要 - C、C++、C#、Fortran、Java\*、ASM
  - Visual Studio® 統合環境またはスタンドアロン
  - グラフィカル・インターフェイスとコマンドライン
  - ローカルおよびリモート収集
  - OS X\* で Windows® および Linux\* データを解析<sup>2</sup>

<sup>1</sup> プロセッサによりイベントが異なります。 <sup>2</sup> macOS\* でデータ収集はできません。

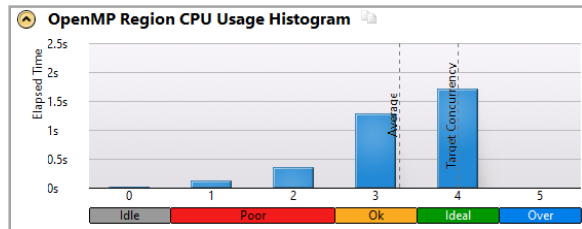
## チューニングの可能性を素早く特定

| Function / Call Stack                                    | CPU Time                      |      |    |       | Spin Time     | Overhead Time |
|----------------------------------------------------------|-------------------------------|------|----|-------|---------------|---------------|
|                                                          | Effective Time by Utilization |      |    |       |               |               |
|                                                          | Idle                          | Poor | Ok | Ideal |               |               |
| ④ FireObject::checkCollision                             | 4.507s                        |      |    |       | 0s            | 0s            |
| ④ FireObject::ProcessFireCollisionsRange                 | 3.444s                        |      |    |       | 0s            | 0s            |
| ④ NtWaitForSingleObject                                  | 0s                            |      |    |       | 3.406s        | 0s            |
| ④ std::basic_ifstream<char,struct std::char_traits<char> | 3.359s                        |      |    |       | 0s            | 0s            |
| ④ Rg_Ogre::FileSystemArchive::open                       | 3.359s                        |      |    |       | 0s            | 0s            |
| ④ CBaseDevice::Present                                   | 2.335s                        |      |    |       | 0.671s        | 0s            |
| Selected 1 row(s):                                       |                               |      |    |       | 1.151s 0.728s | 0s            |

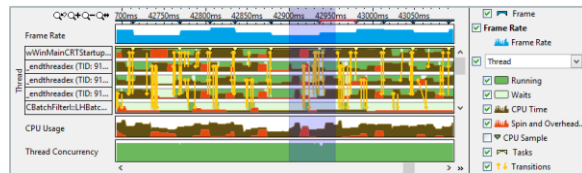
## ソースコードで結果を表示

| Source Line | Source                                     | CPU Time: Total by Utilization |
|-------------|--------------------------------------------|--------------------------------|
| 81          | for (int i = 0; i < mem_array_i_max; i++)  | 0.300s                         |
| 82          | {                                          |                                |
| 83          | for (int j = 0; j < mem_array_j_max; j++)  | 4.936s                         |
| 84          | {                                          |                                |
| 85          | mem_array[j*mem_array_j_max+i] = *fill_val | 7.207s                         |

## OpenMP\* のスケーラビリティをチューニング



## データの視覚化とフィルター



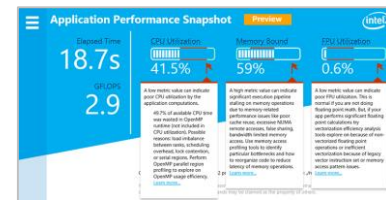
# 2017 の新機能: Python\*、FLOPS、ストレージほか...

## インテル® VTune™ Amplifier XE パフォーマンス・プロファイラー

- Python\* と Python\*/C++/Fortran が混在したコードのプロファイル
- 最新のインテル® Xeon Phi™ プロセッサをチューニング
- HPC パフォーマンスにとって重要な 3 つのメトリックを素早く確認
- **メモリアクセスを最適化**
- **ストレージ解析: I/O 依存か、CPU 依存か?**
- OpenCL\* および GPU プロファイルの拡張
- 簡単に使用できるリモートアクセス/コマンドライン
- タイムラインにカスタムカウンターを追加可能
- プレビュー: アプリケーションと**ストレージのパフォーマンス・スナップショット** (<http://www.intel.com/application-snapshot> (英語))
- インテル® Advisor: インテル® AVX-512 向けにベクトル化を最適化 (ハードウェアの有無に関係なく実行可能)



Disk Input and Output Histogram



# インテル® VTune™ Amplifier XE で Knights Landing<sup>+</sup> プロセッサをチューニング インテル® Xeon Phi™ プロセッサ向けの 4 つの重要な最適化

## 1) 高帯域メモリー

- MCDRAM に配置するデータ構造の決定
- パフォーマンスの問題をメモリー階層で表示
- DRAM および MCDRAM の帯域幅を測定

## 2) MPI と OpenMP\* のスケーラビリティー

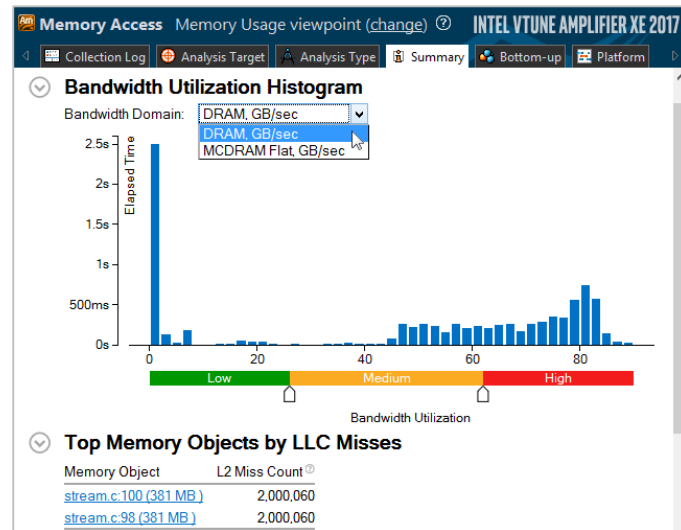
- シリアル時間と並列時間
- インバランス、オーバーヘッド・コスト、並列ループ・パラメーター

## 3) マイクロアーキテクチャーの効率

- コア・パイプラインにおけるコードの効率を確認
- カスタム PMU イベントで絞り込み

## 4) ベクトル化の効率: インテル® Advisor を使用

- インテル® AVX-512 対応ハードウェアの有無に関係なく  
インテル® AVX-512 向けに最適化



+ 開発コード名

A large, stylized graphic on the left side of the slide. It consists of a spiral of binary digits (0s and 1s) in white and light blue, creating a sense of depth and movement. The spiral starts from the center and expands outwards towards the top-left and bottom-left corners.

# ストレージ向けの インテル® VTune™ Amplifier XE

# ストレージデバイス解析 (HDD、SATA、NVMe SSD)

インテル® VTune™ Amplifier XE

新機能!

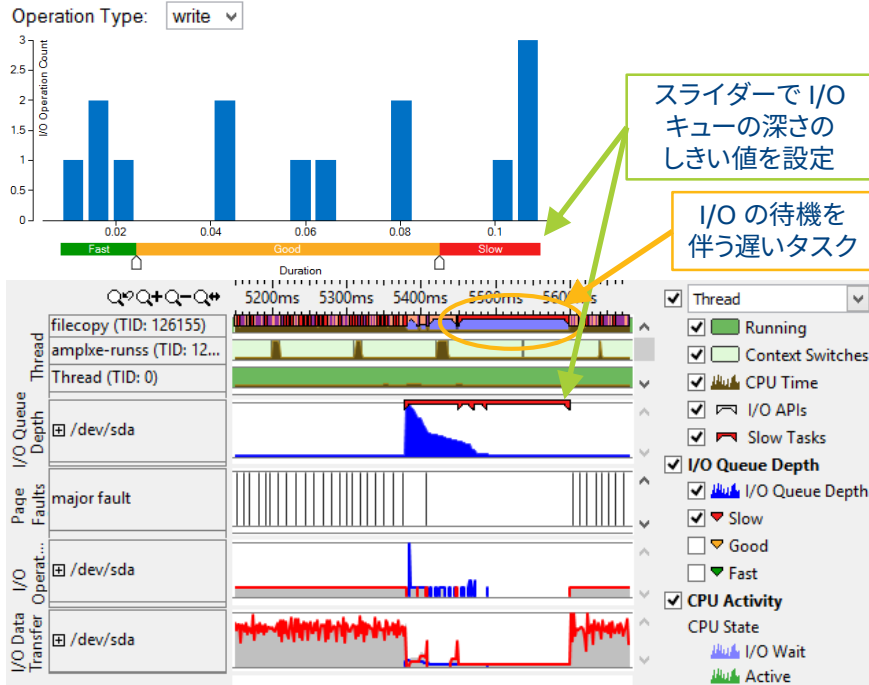
## I/O 依存か、CPU 依存か?

- I/O 操作 (非同期/同期) と計算の間のインバランスを調査
- ストレージアクセスをソースコードにマップ
- CPU が I/O を待機している個所を確認
- ストレージへのバス帯域幅を測定

## レイテンシー解析

- レイテンシー・ヒストグラムを利用してストレージアクセスをチューニング
- I/O を複数のデバイスに分散

### Disk Input and Output Histogram



# インテル® VTune™ Amplifier XE による ディスク入出力解析

ディスク関連の問題に特化した解析タイプ  
“Disk Input Output Analysis”:

- 次のパフォーマンスの問題を検出:
  - 利用率
  - キューの深さ
  - レイテンシー
  - 帯域幅

# I/O を伴うプログラムの例

```
infile = fopen(filename, "rb");  
while ((bytes_read = fread (buffer, 1, BUFFER_SIZE, infile)) != 0) {  
    MD5_Update(&context, buffer, bytes_read);  
}  
fclose(infile);
```

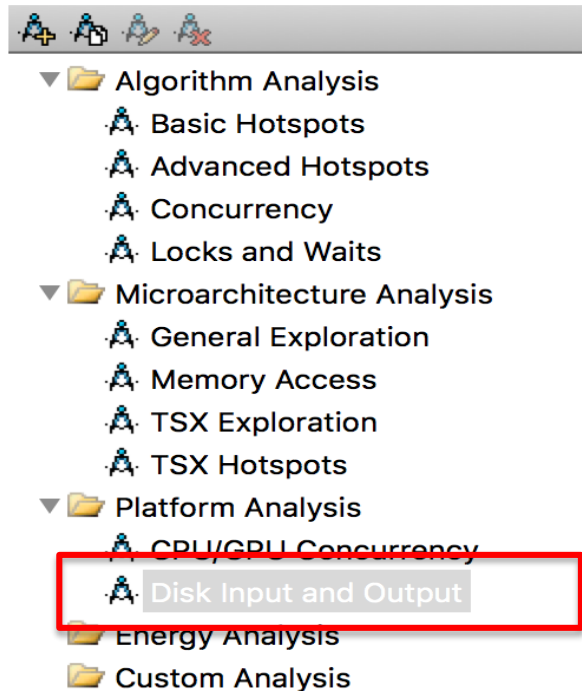
## 3 つの異なるモード

- インテル® VTune™ Amplifier XE がサポートする OS を使用しているか?
  - “Disk Input Output Analysis” メトリックを使用
- 自身で専用コードを記述しているか?
  - インストルメントとトレース (ITT) API を使用してインストルメント
  - csv ファイルを作成しインテル® VTune™ Amplifier XE に統合

# (典型的な)ストレージ向けの インテル® VTune™ Amplifier XE のメトリック

## OS とハードウェアのテレメトリの相関

- ftrace で収集した OS テレメトリ
- PMU (インテルのコレクター) で収集したハードウェア・テレメトリ
- ファイル操作 API は、PIN で追跡される(バイナリー・インストルメント)
- システムの動作を細粒度で追跡



**I/O のストール  
と I/O API のインストルメント**

**レイテンシー**

**コールスタック**

**デバイスの利用率**

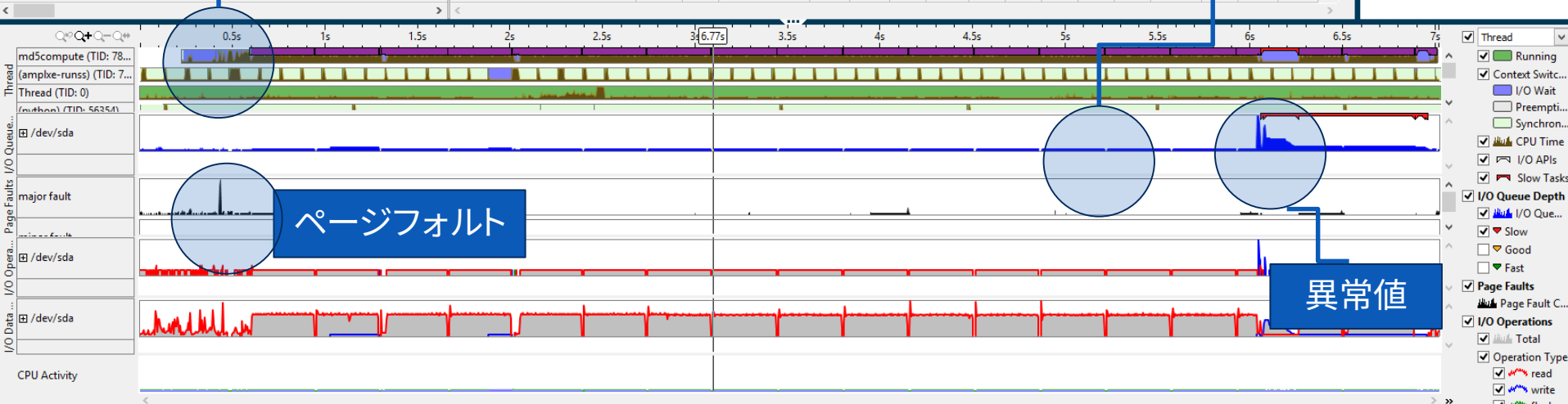
| Process            | Spin Time | Over Time | Instructions Retired | CPI Rate      | CPU Freq... Ratio | Task Time | Task Count | Mod...  | Fun... (Full) | Sour... File | Mod... Path | Sta... |
|--------------------|-----------|-----------|----------------------|---------------|-------------------|-----------|------------|---------|---------------|--------------|-------------|--------|
| md5com...          | 0.024s    | 0s        | 8,954,013,431        | 0.552         | 0.558             | 3.179s    | 158,324    |         |               |              |             | 0      |
| ampbe-r...         | 0s        | 0s        | 1,294,001,941        | 0.824         | 0.536             |           |            |         |               |              |             | 78103  |
| python             | 0.005s    | 0s        | 206,000,309          | 4.505         | 0.541             |           |            |         |               |              |             | 0      |
| top                | 0.001s    | 0s        | 254,000,381          | 0.630         | 0.506             |           |            |         |               |              |             | 56347  |
| pinbin             | 0s        | 0s        | 126,000,189          | 0.619         | 0.565             |           |            |         |               |              |             | 7748   |
| pinbin             | 0.018s    | 0.001s    | 16,000,024           | 1.250         | 0.455             |           |            |         |               |              |             | 10     |
| pinbin             | 0.009s    | 0s        | 12,000,018           | 0.833         | 0.500             |           |            |         |               |              |             | 26     |
| kworker/0:2        | 0.006s    | 0s        | 4,000,006            | 1.500         | 0.429             |           |            |         |               |              |             | 0      |
| rcu_sched          | 0.006s    | 0s        | 2,000,003            | 6.000         | 0.857             |           |            |         |               |              |             | 8      |
| ampbe-cl           | 0.005s    | 0s        | 2,000,003            | 4.000         | 0.667             |           |            |         |               |              |             | 78088  |
| irqbalance         | 0.005s    | 0s        | 10,000,015           | 0.600         | 0.500             |           |            |         |               |              |             | 2031   |
| kworker/u290:0     | 0.004s    | 0s        | 4,000,006            | 1.000         | 0.400             |           |            |         |               |              |             | 57684  |
| kworker/16:2       | 0.003s    | 0s        | 0                    | 0.000         | 0.000             |           |            |         |               |              |             | 30603  |
| ampbe-perf         | 0.003s    | 0s        | 2,000,003            | 2.000         | 0.667             |           |            |         |               |              |             | 78121  |
| rcuos/12           | 0.002s    | 0s        | 0                    | 0.500         |                   |           |            |         |               |              |             | 0      |
| pin                | 0.002s    | 0s        | 2,000,003            | 1.000         | 0.500             |           |            |         |               |              |             | 0      |
| pinbin             | 0.002s    | 0s        | 0                    | 0.500         |                   |           |            |         |               |              |             | 0      |
| pinbin             | 0.001s    | 0s        | 0                    | 0.000         | 0.000             |           |            |         |               |              |             | 0      |
| sshd               | 0.001s    | 0s        | 0                    | 0.000         | 0.000             |           |            |         |               |              |             | 0      |
| Selected 1 row(s): | 3.840s    | 0.024s    | 0s                   | 8,954,013,431 | 0.552             | 0.558     | 3.179s     | 158,324 |               |              |             | 5,049  |

Summary Call Stack  
the Bottom-up and Timeline  
panels to identify the I/O  
operations with slow duration.

Operation Type: read

5000  
4000  
3000  
2000  
1000  
00.1 0.2  
Slow  
Duration (sec)

Apply



- Thread
- ☒ Running
- ☒ Context Swit...
- ☒ I/O Wait
- ☐ Preempti...
- ☐ Synchron...
- ☒ CPU Time
- ☒ I/O APIs
- ☒ Slow Tasks
- ☒ I/O Queue Depth
- ☒ I/O Que...
- ☒ Slow
- ☒ Good
- ☐ Fast
- ☒ Page Faults
- ☒ Page Fault C...
- ☒ I/O Operations
- ☒ Total
- ☒ Operation Type
- ☒ read
- ☒ write
- ☒ flush

# ソースコードヘドリルダウン

Disk Input and Output Disk Input and Output viewpoint (change)

Collection Log Analysis Target Summary Bottom-up Platform md5com...

Source Assembly

Assembly grouping: Address

| So. Li. ▲ | Source                                                              | CPU Time       | Spin... | Ov | Address ▲ | Sour... Line | Assembly                    | CPU ▲           |
|-----------|---------------------------------------------------------------------|----------------|---------|----|-----------|--------------|-----------------------------|-----------------|
|           |                                                                     | Effective Time |         |    |           |              |                             | Effective Ti... |
| 21        | #include <stdio.h>                                                  |                |         |    | 0x4008f1  | 46           | movq 0x200768(&rip), %rax   |                 |
| 22        | #include <stdlib.h>                                                 |                |         |    | 0x4008f8  | 46           | movq -0x1038(%rbp), %rdx    |                 |
| 23        | #include <unistd.h>                                                 |                |         |    | 0x4008ff  | 46           | mov %rcx, %rax              |                 |
| 24        | #include <openssl/md5.h>                                            |                |         |    | 0x400902  | 46           | mov %rax, %rdi              |                 |
| 25        |                                                                     |                |         |    | 0x400905  | 46           | mov \$0x0, %eax             |                 |
| 26        | #define BUFFER_SIZE 4096                                            |                |         |    | 0x40090a  | 46           | callq 0x400760 <fprintf>    |                 |
| 27        | extern int                                                          |                |         |    | 0x40090f  |              | Block 7:                    |                 |
| 28        | main (int argc, char** argv)                                        |                |         |    | 0x40090f  | 47           | mov \$0x0, %eax             |                 |
| 29        | {                                                                   |                |         |    | 0x400914  | 47           | jmp 0x400a13 <Block 21>     |                 |
| 30        | unsigned char digest[MD5_DIGEST_LENGTH];                            |                |         |    | 0x400919  |              | Block 8:                    |                 |
| 31        | MD5_CTX context;                                                    |                |         |    | 0x400919  | 50           | lea -0x10a0(%rbp), %rax     |                 |
| 32        | int bytes_read;                                                     |                |         |    | 0x400920  | 50           | mov %rax, %rdi              |                 |
| 33        | unsigned char buffer[BUFFER_SIZE];                                  |                |         |    | 0x400923  | 50           | callq 0x4006d0 <MD5_Init>   |                 |
| 34        | char *filename;                                                     |                |         |    | 0x400928  |              | Block 9:                    |                 |
| 35        | FILE *infile;                                                       |                |         |    | 0x400928  | 51           | jmp 0x40094c <Block 11>     |                 |
| 36        | int idx;                                                            |                |         |    | 0x40092a  |              | Block 10:                   |                 |
| 37        |                                                                     |                |         |    | 0x40092a  | 52           | movl -0x1024(%rbp), %eax    |                 |
| 38        | if (argc != 2) {                                                    |                |         |    | 0x400930  | 52           | movsxd %eax, %rdx           |                 |
| 39        | fprintf(stderr, "Usage ./computemd5 filename\n");                   |                |         |    | 0x400933  | 52           | lea -0x1020(%rbp), %rcx     | 0usec           |
| 40        | return 0;                                                           |                |         |    | 0x40093a  | 52           | lea -0x10a0(%rbp), %rax     |                 |
| 41        | }                                                                   |                |         |    | 0x400941  | 52           | mov %rcx, %rax              |                 |
| 42        |                                                                     |                |         |    | 0x400944  | 52           | mov %rax, %rdi              |                 |
| 43        | filename = argv[1];                                                 |                |         |    | 0x400947  | 52           | callq 0x4006e0 <MD5_Update> |                 |
| 44        | infile = fopen(filename, "rb");                                     |                |         |    | 0x40094c  |              | Block 11:                   |                 |
| 45        | if (infile == NULL) {                                               |                |         |    | 0x40094c  | 51           | lea -0x1020(%rbp), %rax     |                 |
| 46        | fprintf(stderr, "%s can't be opened.\n", filename);                 |                |         |    | 0x400953  | 51           | movq -0x1030(%rbp), %rdx    | 871.573usec     |
| 47        | return 0;                                                           |                |         |    | 0x40095a  | 51           | mov %rdx, %rcx              |                 |
| 48        | }                                                                   |                |         |    | 0x40095d  | 51           | mov \$0x1000, %edx          |                 |
| 49        |                                                                     |                |         |    | 0x400962  | 51           | mov \$0x1, %esi             |                 |
| 50        | MD5_Init(&context);                                                 |                |         |    | 0x400967  | 51           | mov %rax, %rdi              |                 |
| 51        | while ((bytes_read = fread(buffer, 1, BUFFER_SIZE, infile)) != 0) { |                |         |    | 0x40096a  | 51           | callq 0x400720 <fread>      |                 |
| 52        | MD5_Update(&context, buffer, bytes_read);                           |                |         |    | 0x40096f  |              | Block 12:                   |                 |
| 53        | }                                                                   |                |         |    | 0x40096f  | 51           | movl %eax, -0x1024(%rbp)    |                 |
| 54        | MD5_Final(digest, &context);                                        |                |         |    | 0x400975  | 51           | cmpl \$0x0, -0x1024(%rbp)   |                 |
| 55        | fclose(infile);                                                     |                |         |    | 0x40097c  | 51           | jnz 0x40092a <Block 10>     |                 |
| 56        |                                                                     |                |         |    | 0x40097e  |              | Block 13:                   |                 |
| 57        | for(idx = 0; idx < MD5_DIGEST_LENGTH; idx++) {                      |                |         |    | 0x40097e  | 54           | lea -0x10a0(%rbp), %rdx     |                 |
| 58        | fprintf(stdout, "%02x", digest[idx]);                               |                |         |    | 0x400985  | 54           | lea -0x20(%rbp), %rax       |                 |
| 59        | }                                                                   |                |         |    | 0x400989  | 54           | mov %rdx, %rax              |                 |
| 60        | fprintf(stdout, " %s\n", filename);                                 |                |         |    | 0x40098c  | 54           | mov %rax, %rdi              |                 |
| 61        |                                                                     |                |         |    | 0x40098f  | 54           | callq 0x400710 <MD5_Final>  |                 |
| 62        | return 0;                                                           |                |         |    | 0x400994  |              | Block 14:                   |                 |
| 63        | }                                                                   |                |         |    | 0x400994  | 55           | movq -0x1030(%rbp), %rax    |                 |
|           |                                                                     |                |         |    | 0x40099b  | 55           | mov %rax, %rdi              |                 |

Sele... 871.573usec 0usec 0x

Highlight...

# 独自の非カーネルモードのドライバーを使用する場合

ほかで取得した結果をインテル® VTune™ Amplifier XE へ容易に追加

- インテル® VTune™ Amplifier XE のインストルメントとトレース API の利用を推奨
  - 追加手順なしでフローを統合
- csv ファイルを作成してインテル® VTune™ Amplifier XE へインポート
  - フローを統合しないでほかのコレクターから結果をインポート

# ITT API - インテル® VTune™ Amplifier XE へ簡単に統合

インストルメントとトレース・テクノロジー (ITT):

- 結果をインテル® VTune™ Amplifier XE へ統合
- オープンソース
- トレースの生成と制御
- 特定のコードを実行中にマークを設定することが可能
- フレームとタスクをマークすることが可能
- ランタイムで生成されたコード (JIT コードとも呼ばれる) をプロファイルすることが可能
- 標準システム API なしで実装されたカスタム同期プリミティブを指定できる

# コードにタスク API を追加 - 例

ドメインと名前を定義:

- `__itt_domain_create( ... )`
- `__itt_string_handle_create( ... )`

タスクの開始と終了をマーク:

- `__itt_task_begin()`
- `__itt_task_end()`

```
fread() 呼び出しをインストルメント -  
itt_fread ( ... )  
{  
    __itt_task_begin( ... );  
    bytes = fread( ... );  
    __itt_task_end( ... );  
}
```

# インストルメントを追加したコード

```
infile = itt_fopen(filename, "rb");  
while ((bytes_read = itt_fread (buffer, 1, BUFFER_SIZE, infile)) != 0) {  
    MD5_Update(&context, buffer, bytes_read);  
}  
itt_fclose(infile);
```

## 3 つのファイル操作を変更

- fread => itt\_fread
- fopen => itt\_fopen
- fclose => itt\_fclose

Grouping:

Process / Module / Function / Thread / Call Stack

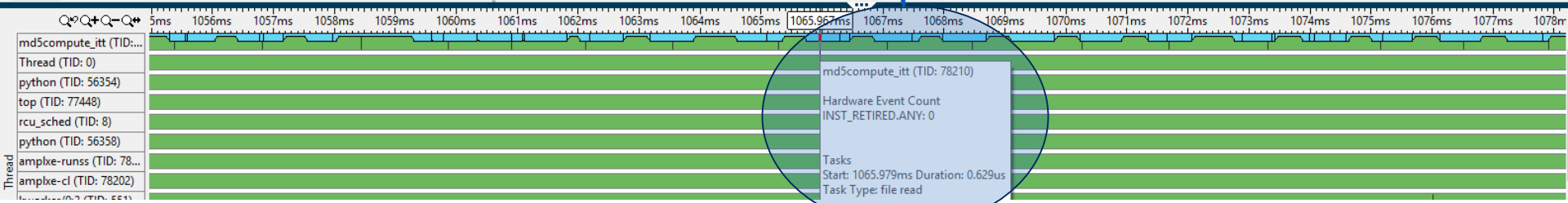
| Process / Module / Function / Thread / Call Stack | Clockticks▼   | Instruction... | CPI ... | Task... | Task ... | Mod...    | Fun... | Sour...   | Mod...   | Start... | PID    | TID |
|---------------------------------------------------|---------------|----------------|---------|---------|----------|-----------|--------|-----------|----------|----------|--------|-----|
| md5compute_itt                                    | 4,784,000,000 | 8,240,900,000  | 0.581   | 3.046s  | 633,283  |           |        |           |          | 0        | 78210  | 0   |
| libcrypto.so.1.0.0                                | 3,210,800,000 | 6,168,600,000  | 0.521   |         |          |           |        | /lib/ ... |          | 0        | 78210  | 0   |
| [Unknown]                                         | 834,900,000   | 890,100,000    | 0.938   |         |          |           |        |           |          | 0        | 78210  | 0   |
| libittnotify_collector.so                         | 450,800,000   | 611,800,000    | 0.737   |         |          |           |        | /opt ...  |          | 0        | 78210  | 0   |
| libc-2.19.so                                      | 202,400,000   | 377,200,000    | 0.537   |         |          |           |        | /lib/ ... |          | 0        | 78210  | 0   |
| libpthread-2.19.so                                | 80,500,000    | 170,200,000    | 0.473   |         |          |           |        | /lib/ ... |          | 0        | 78210  | 0   |
| md5compute_itt                                    | 2,300,000     | 23,000,000     | 0.100   |         |          |           |        | /tm ...   |          | 0        | 78210  | 0   |
| main                                              | 6,900,000     | 0.333          |         |         | md5..    | main      | md5..  | /tm ...   | 0x40 ... | 0        | 78210  | 0   |
| itt_fread                                         | 16,100,000    | 0.000          |         |         | md5..    | itt_fr... | md5..  | /tm ...   | 0x40 ... | 0        | 78210  | 0   |
| ld-2.19.so                                        | 0             |                |         |         |          |           |        | /lib/ ... |          | 0        | 78210  | 0   |
| Pid 0x0                                           | 121,900,000   | 5.528          |         |         |          |           |        |           |          | 0        | 0      | 0   |
| python                                            | 133,400,000   | 186,300,000    | 0.716   |         |          |           |        |           |          | 0        | 56347  | 0   |
| top                                               | 32,200,000    | 50,600,000     | 0.636   |         |          |           |        |           |          | 0        | 77448  | 0   |
| amplxe-cl                                         | 11,500,000    | 0              |         |         |          |           |        |           |          | 0        | 78187  | 0   |
| rcu_sched                                         | 9,200,000     | 2,300,000      | 4.000   |         |          |           |        |           |          | 0        | 8      | 0   |
| amplxe-runss                                      | 9,200,000     | 4,600,000      | 2.000   |         |          |           |        |           |          | 0        | 78203  | 0   |
| rcuos/16                                          | 2,300,000     | 0              |         |         |          |           |        |           |          | 0        | 124    | 0   |
| kworker/0:2                                       | 2,300,000     | 4,600,000      | 0.500   |         |          |           |        |           |          | 0        | 551    | 0   |
| ypbind                                            | 2,300,000     | 0              |         |         |          |           |        |           |          | 0        | 1780   | 0   |
| sep                                               | 2,300,000     | 0              |         |         |          |           |        |           |          | 0        | 78218  | 0   |
| kworker/25:2                                      | 2,300,000     | 0              |         |         |          |           |        |           |          | 0        | 82956  | 0   |
| automount                                         | 2,300,000     | 0              |         |         |          |           |        |           |          | 0        | 145147 |     |

新しいホットスポット関数

ユーザータスク

Highlighted 8 row(s):

4,779,400,000 8,234,000,000 0.580 3.046s 633,283



# ソースと特定のタスクにドリルダウン

Advanced Hotspots Hardware Issues viewpoint (change) ? INTEL VTUNE AMPLIFIER XE 2017

Collection Log Analysis Target Analysis Type Summary Bottom-up Top-down Tree Platform md5com...

Source Assembly Assembly grouping: Address

| So. Li. ^ | Source                                                                      | Hardware Event Count by ... |        |           |
|-----------|-----------------------------------------------------------------------------|-----------------------------|--------|-----------|
|           |                                                                             | INST_...*                   | CPU_.. | CPU_C...  |
| 16        | __itt_string_handle* str_close;                                             |                             |        |           |
| 17        |                                                                             |                             |        |           |
| 18        | static size_t itt_fread(void *ptr, size_t size, size_t nmemb, FILE *stream) |                             |        |           |
| 19        | {                                                                           | 6,900,000                   | 0      | 2,300,000 |
| 20        | size_t bytes;                                                               |                             |        |           |
| 21        | __itt_task_begin(domain, __itt_null, __itt_null, str_read);                 | 6,900,000                   | 0      | 6,900,000 |
| 22        | bytes = fread(ptr, size, nmemb, stream);                                    | 0                           | 0      | 6,900,000 |
| 23        | __itt_task_end(domain);                                                     | 2,300,000                   | 0      | 2,300,000 |
| 24        | return bytes;                                                               |                             |        |           |
| 25        | }                                                                           |                             |        |           |
| 26        |                                                                             |                             |        |           |
| 27        | static size_t itt_fwrite(const void *ptr, size_t size, size_t nmemb,        |                             |        |           |
| 28        | FILE *stream)                                                               |                             |        |           |
| 29        | {                                                                           |                             |        |           |
| 30        | size_t bytes;                                                               |                             |        |           |
| 31        | __itt_task_begin(domain, __itt_null, __itt_null, str_wr);                   |                             |        |           |
| 32        | bytes = fwrite(ptr, size, nmemb, stream);                                   |                             |        |           |
| 33        | itt task end(domain);                                                       |                             |        |           |

タスクとソースコード

# 独自のコレクターを持っているか？

## インテル® VTune™ Amplifier XE へ結果をインポート

インテル® VTune™ Amplifier XE  
は、インストルメント結果をイン  
ポートする簡単な方法を提供

- 開始時間と終了時間を含む間隔データ
- カウンターのセットを含むサンプル

<https://software.intel.com/en-us/node/544116> (英語)

The screenshot shows the Intel Developer Zone website. At the top, there's a blue header with the Intel logo, "Developer Zone", and links for "Join Today" and "Log in". Below the header, there's a navigation bar with "Development", "Tools", and "Resources". A search bar with "powered by Google" is also present. The main content area is titled "Creating a CSV File with External Data". It explains that Intel® VTune™ Amplifier can process and integrate performance statistics collected externally with a custom collector or with your target application in parallel with the native VTune Amplifier analysis. To achieve this, provide the collected custom data as a csv file with a predefined structure and save this file to the VTune Amplifier result directory. It lists two data types: "Interval data with start time and end time" and "Samples with a set of counters". It then states that to make the VTune Amplifier interpret the custom statistics from the csv file, the file format must meet the following requirements: "File Name" and "Filename format". The filename format is specified as `[user-defined]-hostname-<hostname-of-system>.csv`. It also provides a "Where:" section with three bullet points: "[user-defined] is an option string, for example, describing the type of data collected", "-hostname- is a required text that must be specified verbatim", and "<hostname-of-system> is the name of the system where the data is collected. If you use a".

intel Developer Zone Join Today Log in

Development Tools Resources powered by Google

Search document...

Intel® VTune™ Amplifier XE 2016 and Intel® VTune™ Amplifier 2016 for Systems Help

Table of Contents

Legal Information

Introducing the Intel® VTune Amplifier

Getting Started with Intel® VTune™ Amplifier

Tuning Methodology

Performance Analysis User's Guide

Performance Analysis Setup

Configuring Analysis Options

GPU Analysis

### Creating a CSV File with External Data

Intel® VTune™ Amplifier can process and integrate performance statistics collected externally with a custom collector or with your target application in parallel with the native VTune Amplifier analysis. To achieve this, provide the collected custom data as a csv file with a predefined structure and save this file to the VTune Amplifier result directory.

VTune Amplifier can load and process the following data types:

- Interval data with start time and end time
- Samples with a set of counters

To make the VTune Amplifier interpret the custom statistics from the csv file, make sure the file format meets the following requirements:

#### File Name

csv filename should specify the hostname where your custom collector gathered the data, following these format requirements:

**Filename format:** `[user-defined]-hostname-<hostname-of-system>.csv`

Where:

- [user-defined] is an option string, for example, describing the type of data collected
- hostname- is a required text that must be specified verbatim
- <hostname-of-system> is the name of the system where the data is collected. If you use a

# 独自のインストルメントを使用する場合

- csv ファイルへ独自のインストルメント情報を記録
- インテル® VTune™ Amplifier XE へ結果をインポート

*fread()* 呼び出しをインストルメント -  
csv\_fread ( ... )  
{  
    \_\_csv\_begin( ... );  
    bytes = fread( ... );  
    \_\_csv\_end( ... );  
}



```
name,start_tsc.TSC,end_tsc,pid,tid  
function1_task_type,419280823342846,419280876920231,12832,11644  
function2_task_type,419280876920231,419281044717992,12832,11644  
function1_task_type,419281044745822,419281102121452,12832,11644  
function2_task_type,419281102121452,419281277898762,12832,11644  
function1_task_type,419281277935812,419281342158661,12832,11644  
function2_task_type,419281342158661,419281527040239,12832,11644
```

### Launch Application

Specify and configure your analysis target: an application or a script to execute. Press F1 for more details.

Application: `./md5compute_csv`

```
Application parameters: "linux-4.4.4.tar"
```

☐ Use application directory as working directory

Working directory:

User-defined environment variables:

Modify...

Managed code profiling mode: **Auto** ▼

⌵ Advanced

 Binary/Symbol Search Source Search Re-resolve Import from CSV

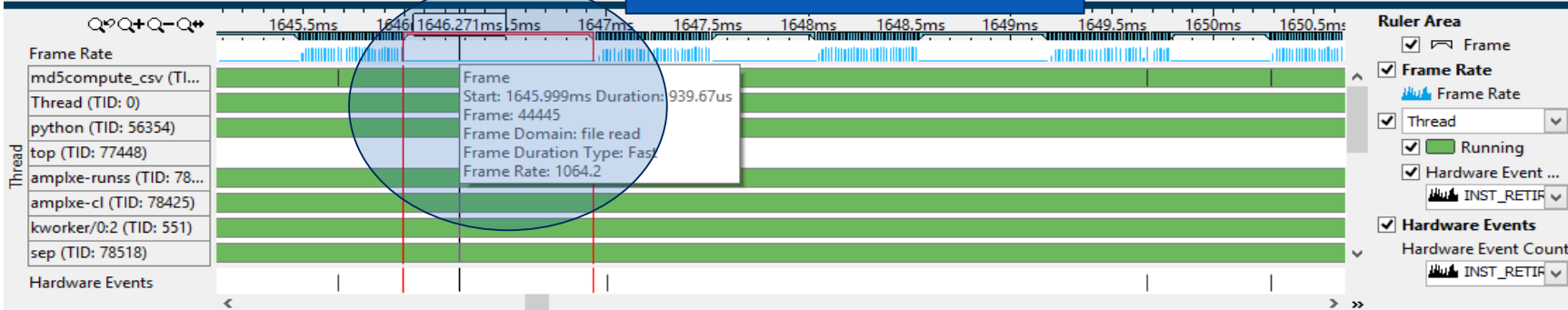
 Clear Log

Grouping: Process / Module / Function / Thread / Call Stack

| Process / Module / Function / Thread / Call Stack | Clockticks▼   | Instruction... | CPI ... | Mod... | Fun...   | Sour...   | Mod...  | Start... | PID   | TID |
|---------------------------------------------------|---------------|----------------|---------|--------|----------|-----------|---------|----------|-------|-----|
| md5compute_csv                                    | 3,962,900,000 | 7,058,700,000  | 0.561   |        |          |           |         | 0        | 78433 | 0   |
| libcrypto.so.1.0.0                                | 3,243,000,000 | 6,154,800,000  | 0.527   |        |          | /lib/ ... |         | 0        | 78433 | 0   |
| [Unknown]                                         | 570,400,000   | 568,100,000    | 1.004   |        |          |           |         | 0        | 78433 | 0   |
| libc-2.19.so                                      | 138,000,000   | 308,200,000    | 0.448   |        |          | /lib/ ... |         | 0        | 78433 | 0   |
| md5compute_csv                                    | 11,500,000    | 27,600,000     | 0.417   |        |          | /tm ...   |         | 0        | 78433 | 0   |
| rdtsc                                             | 4,600,000     | 9,200,000      | 0.500   | md5 .. | rdtsc    | csv_...   | /tm ... | 0x40 ... | 78433 | 0   |
| __csv_print                                       | 4,600,000     | 6,900,000      | 0.667   | md5 .. | __cs ... | csv_...   | /tm ... | 0x40 ... | 78433 | 0   |
| csv_fread                                         | 2,300,000     | 4,600,000      | 0.500   | md5 .. | csv_...  | md5 ..    | /tm ... | 0x40 ... | 78433 | 0   |
| __csv_end                                         | 0             | 2,300,000      | 0.000   | md5 .. | __cs ... | csv_...   | /tm ... | 0x40 ... | 78433 | 0   |
| __csv_begin                                       | 0             | 2,300,000      | 0.000   | md5 .. | __cs ... | csv_...   | /tm ... | 0x40 ... | 78433 | 0   |
| main                                              | 0             | 2,300,000      | 0.000   | md5 .. | main     | md5 ..    | /tm ... | 0x40 ... | 78433 | 0   |
| Pid 0x0                                           | 0             | 0              | 0       |        |          |           |         | 0        | 0     | 0   |
| python                                            | 0             | 56347          | 0       |        |          |           |         | 0        | 56347 | 0   |
| top                                               | 0             | 77448          | 0       |        |          |           |         | 0        | 77448 | 0   |
| amplxe-cl                                         | 0             | 78412          | 0       |        |          |           |         | 0        | 78412 | 0   |
| amplxe-runss                                      | 0             | 78426          | 0       |        |          |           |         | 0        | 78426 | 0   |

ホットスポット関数

インテル® VTune™  
Amplifier XE の情報と  
ユーザー csv の相関関係



A large, stylized graphic on the left side of the slide. It consists of a spiral shape formed by binary digits (0s and 1s) in white and light blue, set against a dark blue background. The spiral starts from the center and expands outwards, creating a sense of depth and movement.

# メモリー向けの インテル® VTune™ Amplifier XE

# メモリアクセスを最適化

## メモリアクセス解析: インテル® VTune™ Amplifier XE 2017

改善されました

### パフォーマンス向上のためデータ構造をチューニング

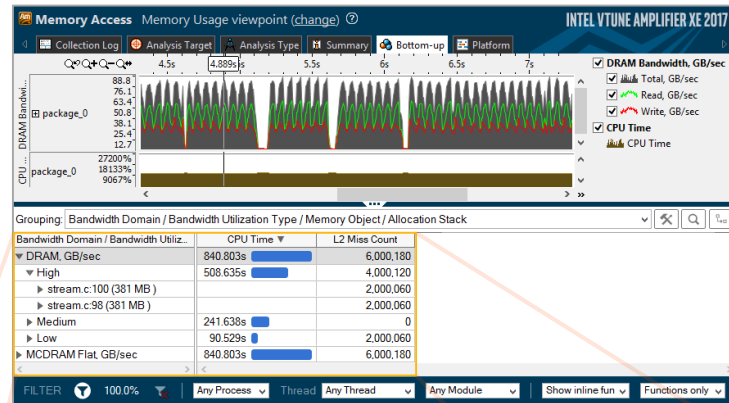
- キャッシュミス (コード行だけでなく) データ構造に紐付け
- カスタム・メモリー・アロケーターのサポート

### NUMA レイテンシーとスケーラビリティの最適化

- 共有とフォルス・シェアリングのチューニング
- 最大システム帯域幅を自動検出
- ソケット間の帯域幅のチューニングが容易

### 簡単にインストールでき、最新のプロセッサに対応

- Linux\* では特別なドライバーは不要
- インテル® Xeon Phi™ プロセッサの MCDRAM (高帯域メモリー) 解析



| Bandwidth Domain / Bandwidth Utiliz... | CPU Time ▼ | L2 Miss Count |
|----------------------------------------|------------|---------------|
| ▼ DRAM, GB/sec                         | 840.803s   | 6,000,180     |
| ▼ High                                 | 508.635s   | 4,000,120     |
| ▶ stream.c:100 (381 MB)                |            | 2,000,060     |
| ▶ stream.c:98 (381 MB)                 |            | 2,000,060     |
| ▶ Medium                               | 241.638s   | 0             |
| ▶ Low                                  | 90.529s    | 2,000,060     |
| ▶ MCDRAM Flat, GB/sec                  | 840.803s   | 6,000,180     |

# インテル® VTune™ Amplifier XE による NUMA メモリー解析

メモリーに関連した問題に注目した特殊な解析タイプ“Memory Access”:

- メモリー階層 (L1-、L2-、LLC-Bound、LLC Latency) によるパフォーマンスの問題
- 帯域幅が制限されたアクセス
  - DRAM と QPI の帯域幅をカバー
- NUMA に関連する問題
  - Remote/Local DRAM Ratio、Local DRAM、Remote DRAM、Remote cache
- メモリー・オブジェクト (データ構造) へのパフォーマンス・イベントの属性

# データの収集

## 構成オプション:

- **メモリー・オブジェクトの解析:**

メモリーの割り当て/解放のインストルメントとメモリー・オブジェクトへのハードウェア・イベントのマップを可能にする

- すべてのシステムメモリーの割り当て/解放をインストルメントするため、実行時にオーバーヘッドが生じる可能性がある

- **最小限のメモリー・オブジェクト・サイズで追跡:**

解析に使用するメモリー割り当ての最小サイズを指定。このオプションは、インストルメントの実行時のオーバーヘッドを軽減するのに有効

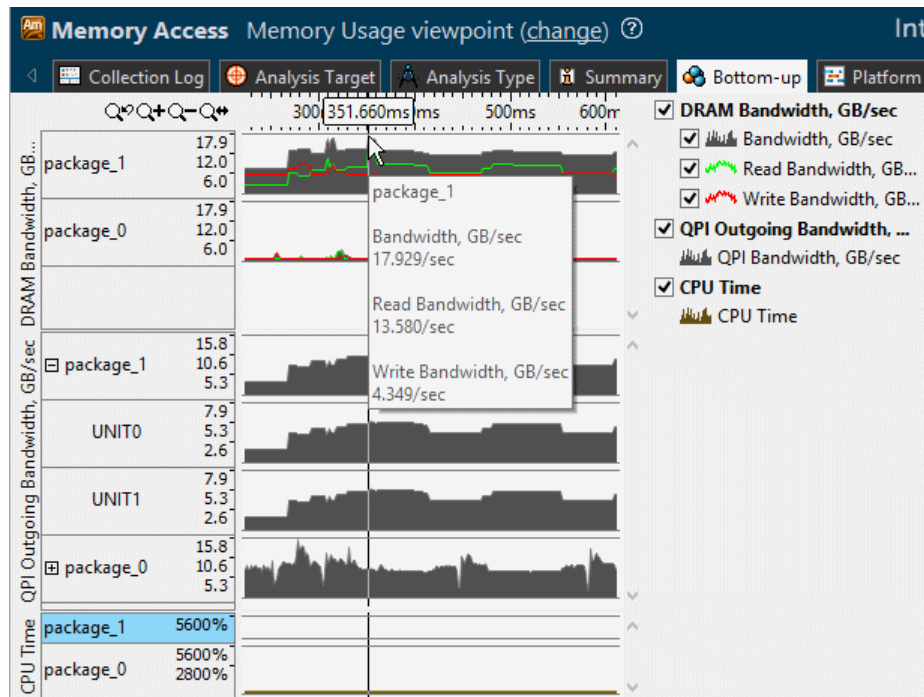
# メモリー・オブジェクトでパフォーマンス・メトリックを表示

グリッドビューで、メモリー・オブジェクトまたはメモリー・オブジェクトの割り当てソースを含むグループ化のレベルの選択

| Grouping: Function / Memory Object / Allocation Stack |          |              |               |             |                |                          |                               |
|-------------------------------------------------------|----------|--------------|---------------|-------------|----------------|--------------------------|-------------------------------|
| Function / Memory Object / Allocation Stack           | CPU Time | Memory Bound | Loads         | Stores      | LLC Miss Count | Average Latency (cycles) | KNL Bandwidth Estimate (GB/s) |
| main                                                  | 2.092s   | 0.651        | 1,074,003,222 | 674,410,116 | 33,602,016     | 248                      | 18.898                        |
| lin_stream.cpp:100 (152 MB)                           |          |              | 456,001,368   | 207,603,114 | 13,800,828     | 226                      |                               |
| lin_stream.cpp:99 (152 MB)                            |          |              | 318,000,954   | 214,803,222 | 12,000,720     | 356                      |                               |
| lin_stream.cpp:98 (152 MB)                            |          |              | 300,000,900   | 252,003,780 | 7,800,468      | 125                      |                               |
| [Unknown]                                             |          |              | 0             | 0           | 0              | 0                        |                               |
| _intel_ssse3_rep_memcpy                               | 0.566s   | 0.000        | 252,000,756   | 128,401,926 | 0              | 41                       | 19.131                        |
| LEVEL_BASE::LinuxProcMapsReader::ParseLine            | 0.002s   | 0.000        | 48,000,144    | 0           | 0              | 0                        | 0.000                         |
| func@0x5042e7                                         | 0.008s   | 0.000        | 36,000,108    | 0           | 0              | 0                        | 2.987                         |
| func@0x2c15f5                                         | 0s       | 0.000        | 24,000,072    | 0           | 0              | 0                        | 0.000                         |

“lin\_stream.cpp:100 (152 MB)” は、ソースファイル lin\_stream.cpp の 100 行目のメモリー・オブジェクトの割り当てを意味する。割り当てサイズは 152MB

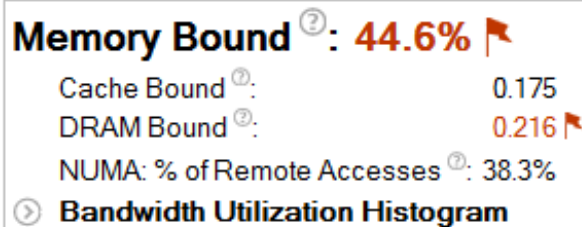
# 帯域幅の問題を時間経過で解析



# NUMA の問題の特定

次の階層で組織化されたメトリックに注目:

- Memory Bound > DRAM Bound > Local DRAM
- Memory Bound > DRAM Bound > Remote DRAM
- Memory Bound > DRAM Bound > Remote Cache
- LLC Miss Count > Local DRAM Access Count、  
LLC Miss Count > Remote DRAM Access Count、  
LLC Miss Count > Remote Cache Access Count
- Average Latency (cycles)



計画: Apache Pass NUMA メトリックの拡張

## 2016 以降

Bandwidth Domain: QPI, GB/sec

| Bandwidth Utilization | Elapsed Time (s) | Region |
|-----------------------|------------------|--------|
| 1                     | ~0.1             | Low    |
| 7                     | ~0.05            | Medium |
| 9                     | ~0.1             | Medium |
| 11                    | ~0.2             | Medium |
| 13                    | ~0.1             | Medium |
| 15                    | ~0.3             | High   |
| 17                    | ~2.0             | High   |
| 19                    | ~0.4             | High   |
| 21                    | ~0.05            | High   |

| Grouping: Bandwidth Domain / Bandwidth Utilization Type / Function / Call Stack |          |              |               |             |                |  |
|---------------------------------------------------------------------------------|----------|--------------|---------------|-------------|----------------|--|
| Bandwidth Domain / Bandwidth Utilization Type / Function / Call Stack           | CPU Time | Memory Bound | Loads         | Stores      | LLC Miss Count |  |
| ⊕ DRAM, GB/sec                                                                  | 2.873s   | 0.818        | 1,764,005,292 | 846,012,690 | 37,202,232     |  |
| ⊕ DRAM Read, GB/sec                                                             | 2.873s   | 0.818        | 1,764,005,292 | 846,012,690 | 37,202,232     |  |
| ⊕ DRAM Write, GB/sec                                                            | 2.873s   | 0.818        | 1,764,005,292 | 846,012,690 | 37,202,232     |  |
| ⊖ QPI, GB/sec                                                                   | 2.873s   | 0.818        | 1,764,005,292 | 846,012,690 | 37,202,232     |  |
| ⊖ High                                                                          | 2.570s   | 1.000        | 1,326,003,978 | 742,811,142 | 33,001,980     |  |
| ⊕ main                                                                          | 1.996s   | 0.699        | 1,068,003,204 | 608,409,126 | 33,001,980     |  |
| ⊕ __intel_sse3_rep_memcpy                                                       | 0.566s   | 0.000        | 252,000,756   | 128,401,926 | 0              |  |
| ⊕ func@0x66ee8                                                                  | 0.006s   | 0.000        | 0             | 0           | 0              |  |
| ⊕ [vmlinux]                                                                     | 0.001s   | 0.000        | 6,000,018     | 0           | 0              |  |
| ⊕ func@0x1a2df4                                                                 | 0.001s   | 0.000        | 0             | 0           | 0              |  |

参考文献:等方性 3 次元有限差分 (3DFD) 波動方程式コード向けの NUMA を理解する

# まとめ

- インテル® Parallel Studio XE Professional Edition を評価してください  
(インテル® VTune™ Amplifier XE を含む)
- 有用なサンプルコードを参照

# 法務上の注意書きと最適化に関する注意事項

インテル® テクノロジーの機能と利点はシステム構成によって異なり、対応するハードウェアやソフトウェア、またはサービスの有効化が必要となる場合があります。実際の性能はシステム構成によって異なります。絶対的なセキュリティを提供できるコンピューター・システムはありません。詳細については、各システムメーカーまたは販売店にお問い合わせいただくか、<http://www.intel.co.jp/> を参照してください。

本資料には、開発中の製品、サービスおよびプロセスについての情報が含まれています。ここに記載されているすべての情報は、予告なく変更されることがあります。インテルの最新の製品仕様およびロードマップをご希望の方は、インテルの担当者までお問い合わせください。

四半期、年度、および将来の計画と予想について言及している本資料内の記述は、多数のリスクや不確定要素を伴う将来の見通しです。インテルの業績および計画に影響を及ぼす可能性のある要素の詳細については、Form 10-K の年次報告書を含む、インテルの SEC 提出資料に記載されています。

本資料で説明されている製品には、には、エラッタと呼ばれる設計上の不具合が含まれている可能性があり、公表されている仕様とは異なる動作をする場合があります。現在確認済みのエラッタについては、インテルまでお問い合わせください。

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。

インテルは、本資料で参照しているサードパーティーのベンチマーク・データまたは Web サイトについて管理や監査を行っていません。本資料で参照している Web サイトにアクセスし、本資料で参照しているデータが正確かどうかを確認してください。

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、機能に基づいて行なったものです。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。

© 2016 Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Xeon、Intel Xeon Phi、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、Visual Studio、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。

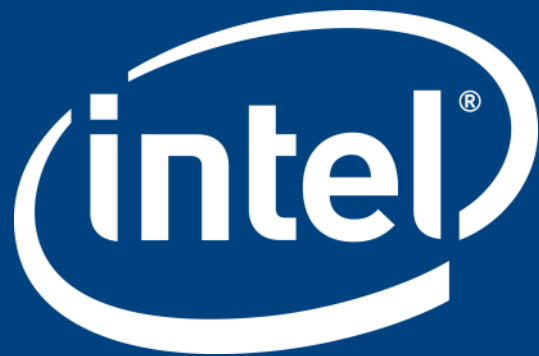
Go は Google Inc. の登録商標または商標です。

\* 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® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサに関して、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依存の最適化は、インテル® マイクロプロセッサ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。

注意事項の改訂 #20110804



# インテル® VTune™ Amplifier XE とは? なぜ注目する必要があるのか?

精度の高いデータと意味のある解析で高速なコードを素早く得る  
以下の開発者を対象とする

- Linux\* と Linux\* デリバティブ
- FreeBSD\*
- 仮想化された環境
  - KVM / Xen / VMWare\*

インテル® アーキテクチャー・ベースのシステム向けの最良のオシロスコープ

- ソフトウェア・バイナリーのインストルメントとハードウェア・カウンターをミックス