インテルのキャッシュ・モニタリング・テクノロジー: ソフトウェア・サポートとツール (パート 4)

この記事は、インテル® デベロッパー・ゾーンに公開されている「Intel's Cache Monitoring Technology: Software Support and Tools」の日本語参考訳です。

はじめに

インテルのキャッシュ・モニタリング・テクノロジー (CMT) 機能は、2014 年にリリースされたインテル® Xeon® プロセッサー E5-2600 v3 製品ファミリーで採用されました。この機能により、L3 キャッシュの占有状況を詳細に把握でき、スレッド、アプリケーション、VM のプロファイルおよび追跡が可能になります。

以前のブログの記事では、この機能のさまざまな側面について紹介しました。

- 製品ページ: https://software.intel.com/en-us/articles/intel-xeon-e5-2600-v3-product-family
- パート 1: CMT の概要: http://www.isus.jp/article/mic-article/benefit-of-cache-monitoring/
- パート 2: RMID と CMT ソフトウェア・インターフェイスの説明: http://www.isus.jp/article/mic-article/software-visible-interfaces/
- パート 3: 使用モデルとサンプルデータ: http://www.isus.jp/article/mic-article/use-models-and-data/

この記事 (パート 4) では、この機能に対応しているオペレーティング・システム (OS) およびこの機能のテストに利用できるソフトウェア・パッケージについて説明します。

この記事の内容には、Linux* オペレーティング・システム、perf プロファイリング・スイート、インテルから提供されるソフトウェア・パッケージ (POSIX* オペレーティング・システムでアプリケーション/VM をコアへピニングし、アプリケーションまたはピニングした VM ごとに L3 キャッシュの使用状況をモニタリング可能) が含まれます。

スタンドアロン・モニタリングとスケジューラー・ベースのモニタリング

CMT 機能は、モデル固有レジスター (MSR) によりセットアップおよび照会インターフェイスが提供されるため、簡単に利用できます。最近のオペレーティング・システムには、ユーザーが MSR への適切な読み取り/書き込み権限を設定できるアプリケーション・プログラム・インターフェイス (API) またはツールが用意されています。Linux* には、readmsr および writemsr コマンドを含む msr ツールパッケージがあります。Microsoft* Windows* にも同様のインターフェイスがあります。キャッシュ・モニタリングに対する高レベルなアプローチは 2 つあります。

- **スタンドアロン・キャッシュ・モニタリング**は、実行しているタスクに関係なく、コアまたは論理スレッド (以後 CPU と呼びます) から最終レベルキャッシュの使用状況を確認します。RMID は CPU に静的に割り当てられ、定期的に占有状況を読み取ります。この手法は、プラットフォームが静的に構成され、アプリケーションがリソースにピニングされた場合に、適切な結果が得られます。システム管理者が、プラットフォームのバランスが適切であるかどうか、動作が不定なアプリケーションがないかどうかを確認する場合、この手法が合理的です。
- **スケジューラー・ベースのキャッシュ・モニタリング**は、オペレーティング・システムのスケジューラーを利用します。前述のスタンドアロン手法に従って RMID が静的に割り当てられると、コア間を移動するプロセスやスレッドの ID は追跡されず、アプリケーションごとの正確な占有状況を把握できません。この手法は、そのような場合に推奨されます。アプリケーションの占有状況を追跡するには、正確な情報が得られるようにスケジューラーを変更する必要があります。ソフトウェアは RMID をプロセスに割り当て、スケジューラーはアプリケーションが CPU 上にスケジュールされたときにコアと適切な RMID を関連付け

る必要があります。アプリケーションのスケジュールが解除されるか、アプリケーションが異なるコアまたはスレッドに移動したら、実行中のアプリケーションのみの占有状況が追跡されるように、スケジューラーは RMID 割り当てを更新して RMID とコアのマッピング情報を維持する必要があります。また、システム・ソフトウェアは、プロセッサー・ソケットにまたがって必要な CMT 設定を再マップしなければなりません。RMID はソケットごとにローカルに定義されるため、指定した RMID を含むアプリケーションが別のプロセッサーに移動した場合、移動したアプリケーションを追跡できるように、OS または VMM は対象ソケットで利用可能な RMID を見つける必要があります (モニタリングが必要な場合)。

スタンドアロン・キャッシュ・モニタリングとスケジューラー・ベースのキャッシュ・モニタリングを有効にするため、現在、いくつかのソフトウェア開発の取り組みが行われています。

スケジューラー・ベースのモニタリング – Linux* オペレーティング・システムの対応状況

スケジューラー・ベースのキャッシュ・モニタリングは、適切なコアと RMID のマッピング情報を利用してアプリケーションを追跡します。Linux* では、CMT を Linux* スケジューラーと緊密な関係にある perf とその カーネルに統合することでこれを実現します。

対応プラットフォームでは (プロセッサーと OS の両方が CMT に対応している場合)、perf を使用してモニタリングするプロセス/スレッドの指定および RMID の割り当てを行います。モニタリングしないスレッドはすべてデフォルトの RMID に割り当てられ、それらのスレッドの占有状況はまとめて報告されます。perf によってモニタリングするシステムが構成されると、モニタリングするスレッドのコンテキスト・スイッチが perf_events サブシステムをコールバックします。スケジューラーからの CMT コールバックが ('context_switch' カーネル関数で) 発生すると、perf_events サブシステムはスケジュールされているスレッドの RMID を選択して、CPU に割り当てます。関連 RMID (または、スケジュールされているスレッドがモニタリングするように設定されていない場合はデフォルトの RMID) は明示的なモニタリングに使用できます。ここから次のコンテキスト・スイッチまで、この論理プロセッサーからのメモリーの読み取り要求とそれに伴うキャッシュロードは、設定した RMID に割り当てられます。

キャッシュの占有状況を追跡するプロセスまたはスレッドが終了したり、sched_out 関数呼び出しが発生すると、perf CMT コールバックは新しい RMID を選択します。このとき、キャッシュロードがモニタリングするスレッドの占有状況に影響しないように、デフォルトの RMID が選択されます。モニタリングするプロセスが終了すると、関連 RMID は未使用 RMID のプールに返され、新しいモニタリング要求に再利用されます。これらの機能のメインストリーム・サポートは、Linux* カーネルバージョン 3.19 に沿っています。

Perf への実装

CMT (キャッシュ・モニタリング・テクノロジー) の Perf への実装: Linux* perf アプリケーションは、カーネルベースのパフォーマンス・カウンターのインターフェイスを提供します。キャッシュ・モニタリング・テクノロジーをサポートする拡張を利用することで、プロセッサーまたはスレッドごとの最終レベルキャッシュの占有状況をモニタリングすることができます。新しいイベントの名前は *intel_cqm/llc_occupancy/* です。このイベントは、占有状況をバイト単位で返します。perf および Linux* カーネル用のパッチは、次の Web サイトで入手できます。

https://lkml.kernel.org/r/1415999712-5850-1-git-send-email-matt@console-pimps.org

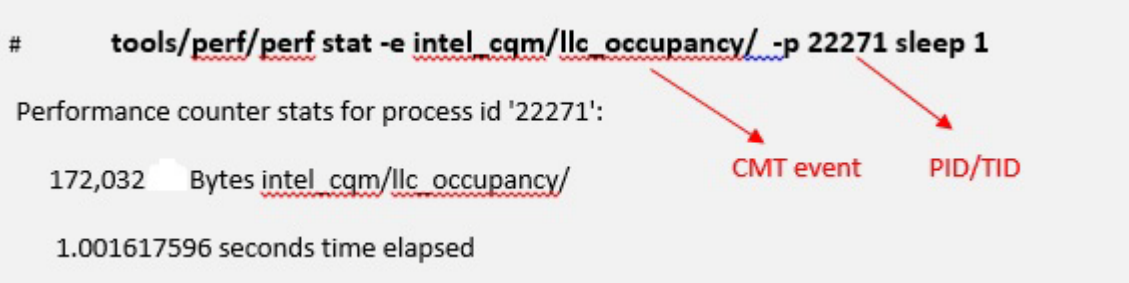
Perf ドライバーモジュールは、CPUID 命令を使用して CMT ハードウェアが利用可能かどうか確認します (詳細は [1] を参照)。CMT が検出されると、Perf にいくつかの関数呼び出しが登録されます。以下は、登録されるイベントとその機能の一部です。

- .event_init:
  - イベント処理 – PID または TID の perf モニタリングを開始します。
  - CMT コールバック – PID/TID ごとに一意の RMID を割り当ておよび設定します。

- .start:
  - イベント処理 – event_init 後に PID または TID の perf モニタリングを開始します。
  - CMT コールバック – モニタリング機能を開始します。
- .add:
  - イベント処理 – PID/TID のモニタリングをスケジュールに追加します。
  - CMT コールバック – スケジュールに追加されたコアでモニタリング機能を設定します。
- .del:
  - イベント処理 – PID/TID のモニタリングをスケジュールから削除します。
  - CMT コールバック – スケジュールから削除されたコアでモニタリング機能をリセットします。
- .read
  - イベント処理 – PID/TID のモニタリング・カウンターを読み取ります。
  - CMT コールバック – PID/TID の関連 RMID を使用して MSR から CMT 占有状況の値を読み取ります。
- .stop
  - イベント処理 – PID または TID の perf モニタリングを終了します。
  - CMT コールバック – モニタリング機能をリセットし、すべての割り当て済み RMID をクリアします。

CPU の占有状況を正確に把握できるように、Perf カーネル・コンポーネントは特定のアプリケーション・スレッドが CPU で実行される間、そのアプリケーション・スレッドに関連付けられた RMID をほかのアプリケーションに割り当てません。前述のように、Linux* スケジューラーがプロセスをスワップすると、RMID とコアの関連付けは解除されます。RMID の追跡に加えて、Perf はプロセスやスレッドの継承もサポートしています (子プロセスは親の RMID を継承します)。

CMT を利用した perf の基本操作:



ユーザー空間の CMT API

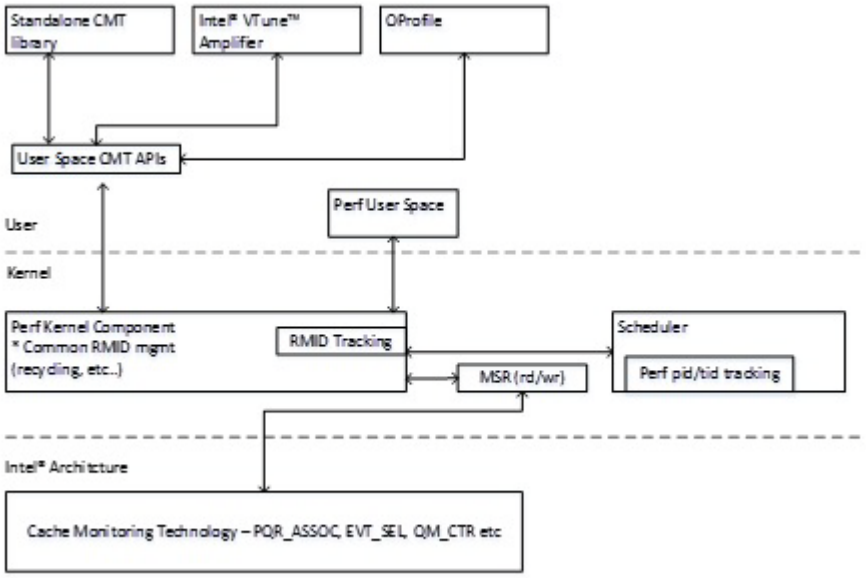
使用するユーザー空間の CMT API 数を制限することで、CMT をアプリケーションに簡単に統合し、利用できます。開発者は、API を使用してアプリケーションのキャッシュ占有状況を取得できます。このような統合アクセス API の実装は、RMID や MSR アクセスなどの共有プラットフォーム・リソースの管理を容易にします。

以下のような関数を perf_event システムコールのラッパーとして使用すると良いでしょう。タスク/PID のキャッシュの占有状況を簡単に追跡できます。

1. pqos_register_cmt(taskid, cpu): この API は、pid/tid と CMT で追跡が必要な cpuid を返します。内部で perf は RMID 割り当て処理を行い、スケジューラー実装で RMID を再利用します。
2. pqos_get_cmt_occupancy(taskid, cpu): この API は、登録されたタスクの最終レベルキャッシュの占有状況を返します。
3. pqos_unregister_cmt(taskid, cpu): この API は、タスクの登録を解除し、キャッシュの占有状況をモニタリングするためタスクの追跡に使用された関連 RMID をすべて解放します。

開発者やシステム管理者が、利用可能な RMID の数やアプリケーション追跡中のその他の RMID 管理タスクを考慮しなくても CMT を利用できるようにする、ユーザー空間ライブラリーを提供する取り組みが現在行われています。

以下の図に示す、API 実装の設計と構成が提案されています。



仮想マシンモニター (VMM) サポート (KVM & Xen)

KVM は Type 2 ハイパーバイザーなので、前述のスケジューラー拡張を継承します。管理者や開発者は、perf を利用して仮想マシンの最終レベルキャッシュの占有状況を追跡できます。仮想マシンのプロセスまたはスレッド ID は、top または Qemu モニターを介してオペレーティング・システムから取得できます。

Xen は Type 1 ハイパーバイザーなので、スケジューラー拡張が最終レベルキャッシュの占有状況を追跡できるようにする必要があります。CMT サポートは、Xen 4.5 で初めて追加されました。ハイパーバイザー実装は RMID と各ドメイン (DomU またはゲスト VM) を関連付けます。すでにモニタリング対象として指定されているものはそれぞれの RMID に、指定されていないものはデフォルトの RMID に関連付けられ、モニタリングされない占有状況データの収集に使用されます。ハイパーバイザーは、各ドメインを CPU にスケジュールし、コンテキスト・スイッチを実行するとともに、RMID を CPU 固有の MSR に書き込むことで、CPU を RMID とその関連ドメインに関連付けます。ドメインが CPU で実行される間、ドメインで発生した CPU からのメモリー読み取りによる最終レベルキャッシュ (LLC) の占有状況は、ドメインに関連付けられた RMID を使用して追跡されます。次のドメインがこの CPU にスケジュールされ、現在モニタリングされているドメインがスイッチアウトされると、この CPU と関連 RMID の関連付けは解除されます。

Xen の xl コマンドツールには、CMT をサポートするためのいくつかのコマンドがあります。これらのコマンドを利用して、ユーザーはモニタリングをドメインにアタッチしたり、デタッチしたり、LLC 占有状況を表示できます。コマンドの入力形式は次のとおりです。

```
$ xl psr-cmt-attach domid
$ xl psr-cmt-detach domid
$ xl psr-cmt-show cache_occupancy
```

上記のコマンド例で、domid はドメインの ID 番号 (ゲスト VM) です。

スタンドアロン・キャッシュ・モニタリング・テクノロジー・ライブラリーによるマルチ OS サポート

このスタンドアロン・ライブラリー (https://01.org/packet-processing で近日公開予定) は、開発者が (前述のスタンドアロン・キャッシュ・モニタリング手法に従って) OS によるサポートなしで CPU ごとの最終レベルキャッシュの占有状況をモニタリングできるようにします。ライブラリー/アプリケーションは、最初にキャッシュ・モニタリングがサポートされているかどうか確認します。初期化が完了すると、モニタリング機能は "top" のようなインターフェイスに CPU ごとの最終レベルキャッシュの占有状況を表示します。ライブラリーにより実装される API を利用することで、開発者は MSR を設定して RMID 割り当てや最終レベルキャッシュの占有状況の取得を行わなくても CMT を利用できます。仮想マシンでライブラリーを利用する場合、CMT モデル固有レジスターにアクセスするには、準仮想化 (PV) か MSR ビットマップが必要になります。一般に、ホスト OS からライブラリーを使用することが推奨されます。

その他のオペレーティング・システムと VMM

その他の OS と VMM も将来は CMT 対応になるでしょう。特定のバージョンで CMT がサポートされているかどうかは、OS/VMM のドキュメントや機能リストで確認してください。

CMT がまだサポートされていない場合でも、各 OS/VMM のカスタマーサポートで機能要求を追跡し、対応予定時期を提供している可能性があります。

まとめ

いくつかの主要 OS と VMM はインテルのキャッシュ・モニタリング・テクノロジー (CMT) をサポートしています。また、未サポートの OS 向けに、テスト、リソース管理ヒューリスティックのプロトタイピング、機能の実装を行うことができるソフトウェア・ライブラリーが 01.org から提供されています。

参考文献

[1] http://www.isus.jp/article/mic-article/software-visible-interfaces/

著者

Andrew Herdrich
Edwin Verplanke
Priya Autee
Will Auld