

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・インターフェイス (パート 2)

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-Visible Interfaces](#)」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 E5-2600 v3 製品ファミリーで採用されました。リリース時のブログは[こちら](https://software.intel.com/en-us/articles/intel-xeon-e5-2600-v3-product-family) (<http://www.isus.jp/article/mic-article/benefit-of-cache-monitoring>) で、製品の概要と機能を確認できます。このブログではソフトウェア・インターフェイスについて詳しく説明し、オペレーティング・システム (OS) と仮想マシンモニター (VMM)/ハイパーバイザーのサポートの例と詳細は将来の記事で取り上げる予定です。

ここでは、リソース・モニタリング ID (RMID)、スレッド/アプリケーション/VM の追跡に使用される抽象化、CPUID 列挙プロセス、モニタリング・データの収集に使用されるモデル固有レジスター (MSR) ベースのインターフェイスについて述べます。

リソース・モニタリング ID (RMID)

CMT 機能は、リソース・モニタリング ID (RMID) と呼ばれる抽象化を利用して、マルチコア・プロセッサで並行して実行中の複数のスレッドを個別に同時にモニタリングできます。

各スレッドのアーキテクチャー MSR (アドレス 0xC8F の IA32_PQR_ASSOC) により、各ハードウェア・スレッドを RMID に関連付ける (特定のハードウェア・スレッドに RMID に割り当てる) ことができます。

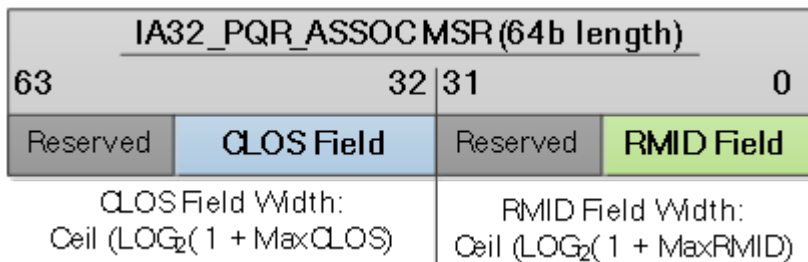


図 1. 各スレッドの IA32_PQR_ASSOC (PQR) MSR は、各スレッドと RMID を関連付けてリソースのモニタリングを可能にします。CLOS はサービスクラス (Class of Service) の略で、CLOS フィールドはリソース割り当てを制御するのに使用されます。これについては、ここでは本題から外れるため詳しく取り上げません。

個別の RMID を利用して、複数のスレッドを別々に追跡することができます。CPUID で列挙されるパラメーターの 1 つから各プロセッサで利用可能な RMID の数が分かります (以下を参照)。

スレッドは個別に、あるいはグループでモニタリングできます。また、複数のスレッドに同じ RMID を割り当てることも可能です。そのため、柔軟なマッピング (図 2) により、仮想化された使用モデルと仮想化されていない使用モデルに幅広く対応できます。

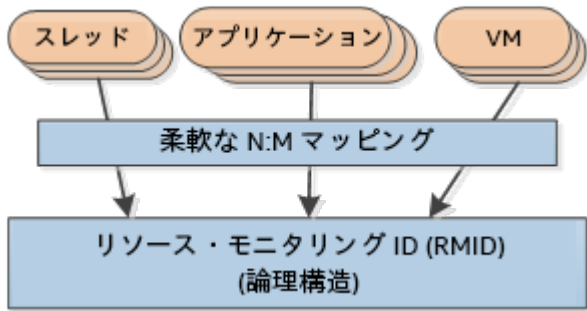


図 2. スレッド、アプリケーション、VM、またはこれらの組み合わせに RMID を割り当てて、非常に柔軟なモニタリングを行うことができます。例えば、VM のすべてのスレッドに同じ RMID を割り当てることで、簡単に VM をモニタリングできます。

プラットフォームで実行中の各アプリケーションや VM は 1 つ以上のスレッドで構成されているため、各アプリケーションまたは VM のモニタリングが可能です。例えば、VM のすべてのスレッドに同じ RMID を割り当てることができます。同様に、アプリケーションのすべてのスレッドに同じ RMID を割り当てることもできます。RMID が (アプリケーションのグループではなく) 特定のアプリケーションのモニタリングにのみ使用される場合、システムによって報告されるその RMID の占有状況には指定したアプリケーションの情報のみが含まれます。

一般に、OS や VMM が CMT をサポートしている場合、RMID は単純に各スレッドのステート構造に追加されます (図 3)。そして、スレッドがコアにスワップされると、適切な RMID で PQR が更新され、アプリケーションまたは VM ごとの追跡が可能になります。

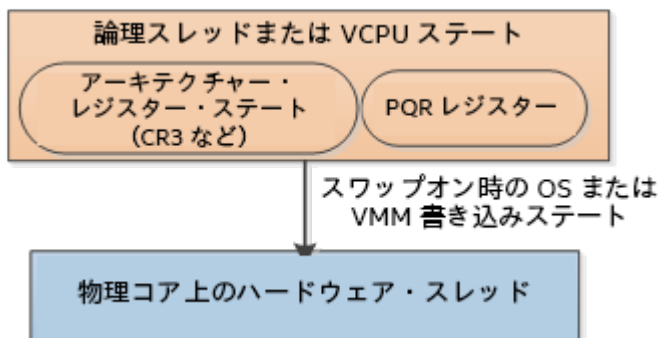


図 3. PQR レジスタ (RMID を含む) は、スレッドまたは VCPU ステートの一部として格納され、実行に向けソフトウェア・スレッドがハードウェア・スレッドにスケジュールされると、スレッド固有レジスタに書き込まれます。

OS または VMM が CMT をサポートしていない場合でも、ソフトウェアは CMT を利用できます。その場合、RMID をコアにピンングし、どのコアでどのアプリケーションの実行が許可されているかを追跡して、その情報をキャッシュ占有状況にマップします。

詳細は、[1] を参照してください。

リソースのタグ付けに便利な RMID は、将来ほかのリソースタイプや機能にも利用できる可能性があります。

キャッシュ・モニタリング・テクノロジー: CPUID 列挙

CPUID 命令を使用して、利用可能な RMID の数を含むプロセッサ世代ごとに異なるすべての CMT パラメータを列挙できます。

通常、対応 OS または VMM では、これらの機能を列挙し、プラットフォームで実行中のソフトウェアが機能の詳細を特定できるように標準化されたインターフェイスが提供されています。このセクションでは、CPUID により提供される詳細について高レベルの概要を示します。

CMT の列挙は階層になっています (図 4)。プラットフォームで一般的なモニタリング機能の有無を検出するには、CPUID.0x7.0 (プロセッサで利用可能な各種機能を示すビットを含むベクトル) のビット 12 を確認します。

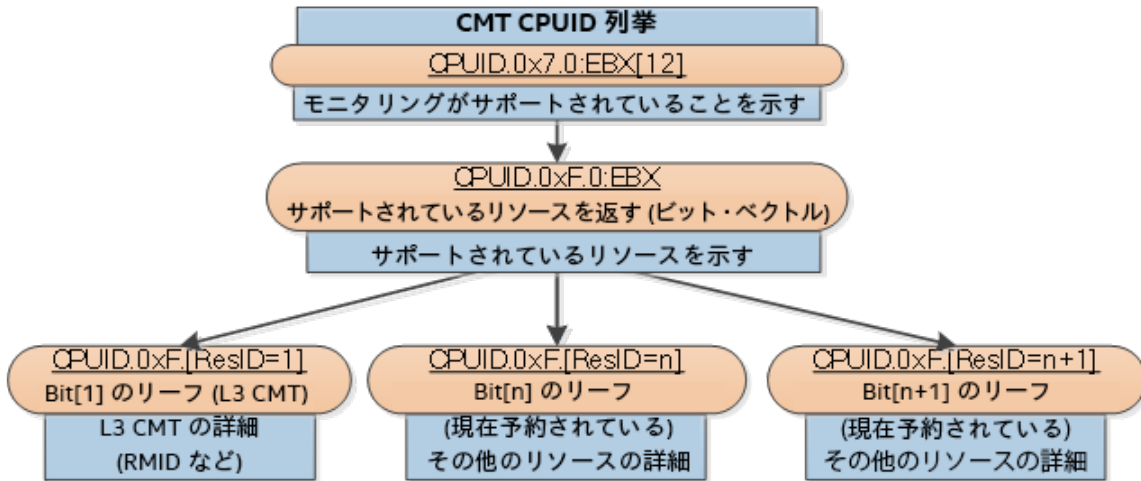


図 4. モニタリング機能の CPUID 列挙階層

モニタリング機能が検出されたら、新しい CPUID 0xF リーフでモニタリング可能なリソースを確認できます。利用可能なリソースに関する一般情報は、CPUID.0xF.0 にリストされています (サブリーフ 0 は特別なケースで、プラットフォーム上のすべてのモニタリング機能に関する詳細を提供します)。

特定のリソースがサポートされていることを確認できたら、さまざまなサブリーフ (CPUID.0xF.[ResourceID]) をポーリングして、各レベルのモニタリングの属性を特定します。例えば、L3 CMT の詳細は CPUID.0xF.1 で分かります。詳細には、L3 CMT でサポートされる RMID の数や、モデル固有レジスター (MSR) インターフェイスから取得したサンプル値をキャッシュ占有状況 (バイト単位) に変換するのに使用されるアップスケール係数などが含まれます。

CPUID で提供されるリーフ、サブリーフ、フィールドのエンコーディングと詳細については、[1] を参照してください。

キャッシュ・モニタリング・テクノロジー: モデル固有レジスター (MSR) インターフェイス

CPUID から CMT サポートを確認し、RMID の数が判明したら、PQR MSR RMID フィールドを利用して各スレッドに RMID を割り当てます (図 1)。

(ソフトウェアで定義されている) 期間が経過したら、MSR のペアを使って RMID の占有状況データを読み取ることができます (選択 MSR に RMID とイベント ID (EvtID) を入力すると、ハードウェアは占有状況を取得しデータ MSR に返します)。

図 5 にイベント選択 MSR (IA32_QM_EVTSEL) を示します。アプリケーションや VM の代わりにモニタリング・データを取得する OS や VMM などのシステム・ソフトウェアは、取得するデータの種類に応じて RMID とイベント ID のペアを設定できます (例: RMID の L3 キャッシュ占有状況データ)。利用可能なイベントコードは CPUID により列挙されます。詳細は、[1] を参照してください。

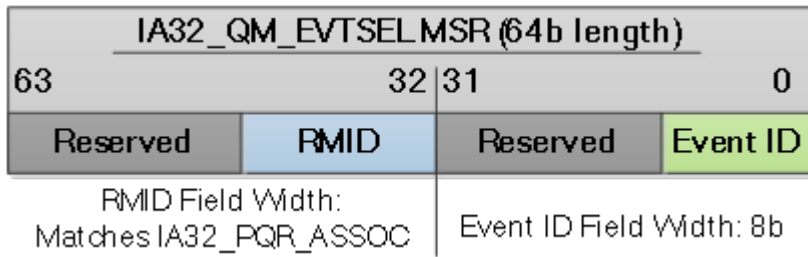


図 5. IA32_QM_EVTSEL MSR を使用して、取得するデータの RMID + イベント ID ペアを選択します。データは IA32_QM_CTR MSR (図 6) に返されます。

ソフトウェアが有効な RMID+イベント ID ペアを指定すると、ハードウェアは指定されたデータを取得し、データ MSR (図 6 の IA32_QM_CTR) に返します。この MSR にはデータが有効かどうかを示すビットのペア (エラーと利用不可) があります。各フィールドの詳細は [1] を参照してください。両方のビットが設定されていない場合、ビット 61:0 のデータは有効なため、ソフトウェアで利用できます。返されたデータが有効であると仮定せずに、常にエラービットを確認すべきです。



図 6. IA32_QM_CTR MSR は、IA32_QM_EVTSEL MSR で指定された RMID+イベント ID のリソース・モニタリング・データを提供します。E (エラー)/U (利用不可) ビットが設定されていない場合は、データは有効です。

L3 CMT 機能の場合、IA32_QM_CTR MSR によって返されたデータをソフトウェアで利用する前に、オプションで CPUID のアップスケール係数を掛けてバイト単位に変換することができます。アップスケール係数を適用しなくても、返された値を使ってアプリケーション/VM の相対的な占有状況を比較できますが、スケーリングは線形になります。

今後追加されるモニタリング機能は、同じ MSR インターフェイスを使用します。つまり、対応しなければならないソフトウェアは増え続けるため、新しい CPUID リーフとモニタリング・イベント・コードにサポートが限定されるでしょう。

結論

RMID を利用することで、CMT 機能はスレッド、アプリケーション、VM、またはそれらの組み合わせを、さまざまなソフトウェア使用モデルに合わせて柔軟な方法で同時に追跡できます。

利用可能な RMID の数を含むプロセッサ世代ごとに異なるすべての CMT パラメーターは、CPUID から判断できます。CMT パラメーターの数は今後増えることが予想されます。

スレッドと RMID の関連付けには、スレッドごとのモデル固有レジスターが 1 つ使用されます。モニタリング・リソース・データの取得には MSR のペアが使用され、次のブログで紹介する使用モデルを可能にします。次のブログでは、使用モデルとサンプルデータについて説明します。

参考文献

[1] 『Intel® 64 and IA-32 Architectures Developer's Manual: Vol. 3B』 第 17 章 14 節

著者

Andrew Herdrich

インテル・ラボの研究者として、2008 年からキャッシュ・モニタリング・テクノロジーや将来のスレッド競合緩和テクノロジーの設計を担当し、最近では高度な将来のアーキテクチャーや NFV 向けの IA の最適化に取り組んでいます。インテル・ラボに配属される前は、Merom[†]、Nehalem[†]、Westmere[†] CPU や第 1 世代 Knights[†] 製品に携わっていました。

[†]開発コード名

コンパイラーの最適化に関する詳細は、[最適化に関する注意事項](#)を参照してください