

メモリシステムにおける電力消費のモデル化手法

川田 裕貴^{1,a)} 広瀬 崇宏³ 高野 了成³ 追川 修一²

概要: メモリシステムを評価する上で、電力消費は重要な尺度の一つである。しかしながら、これまで利用されてきたサイクル精度 CPU、メモリ シミュレータを利用した電力計測手法は、シミュレーションに非常に時間がかかることが問題であった。そこで、私達は DRAM と次世代不揮発性メモリを組み合わせたシステムにおける、実機で利用可能で容易に観測可能な数値や尺度からの、メモリシステムの電力消費の予測手法を開発している。本稿では、実機上で複数のワークロードを利用した実験を行い、DRAM の消費電力とスループット計測した結果と、その関係性について述べる。本手法では、ハードウェア上のパフォーマンスカウンタを用いることで、メモリスループットと電力消費を、最小限のオーバーヘッドで計測することができるので、これを利用した。

1. はじめに

次世代不揮発性メモリの登場により、コンピュータの性能と電力効率の向上が期待されている。次世代不揮発性メモリは、STT-MRAM や PCM, ReRAM などの素子を利用した、近年登場が期待される新しいメモリ技術の総称である。特に、近年のモバイル機器やバッテリーにより駆動されるコンピュータでは、年々大容量化している DRAM による電力消費を無視することはできないため、NVM による電力効率の向上が求められている。近い将来のメモリ技術の展望として、NVM の中でも特に STT-MRAM は、DRAM と同等かそれ以上の読み書き性能が実現されると予測されており、STT-MRAM が一般的なコンピュータのメインメモリとして利用されうる可能性がある。[1] STT-MRAM は、コンピュータ・システムのエネルギー消費を抑えることに大きな貢献できるとして期待されている。NVM では、データの永続性を保証するために、DRAM では必要なフラッシュ動作が不要である。しかし、STT-MRAM モジュールの容量は、近い将来にはそれ単体でメインメモリとして十分に満足できる容量を達成できない。他のメモリ素子と組み合わせ、メインメモリを構成する必要がある。よって、省エネルギーのハイブリッドなメモリシステムを考える必要がある。

現在、私達は NVM と DRAM を組み合わせたハイブリッドメモリシステムの研究を行っている。[2] このシ

テムでは、NVM をメインメモリとして利用し、DRAM をスワップ領域のキャッシュのように利用することで、NVM 領域と DRAM 領域それぞれに対する動的な電源管理を実現する。また、DRAM と NVM 両方が DDR3 などの汎用のメモリバスに接続され、それぞれが特定の物理メモリのアドレスの範囲にマップされるようなシステムを対象としている。NVM は、不揮発性のため電源を切っても記憶されているデータが消えることはない。このことから、メモリが利用されていない間に頻繁に電源をスイッチすることで、電力消費を削減できる可能性がある。

しかし、このようなシステムの提案手法の評価を行う上で、電力消費の特性を考慮することが重要となる。NVM と DRAM では、電力消費の傾向が異なるので、たとえ DRAM を単純に NVM に置き換えたとしても、電力消費の面で有利にならない可能性があり、システムの評価において電力消費のモデル化が必要になる。ハイブリッドメモリシステムにおいては、DRAM を利用する領域と NVM を利用する領域それぞれに対して、別のパラメータによる電力消費の測定が求められる。しかし、メモリの電力消費を予測・計測することは、難しい。既存の手法としては、サイクル精度 CPU シミュレータとメモリシミュレータを組み合わせて測定する方法が存在する。[3] だが、この手法では非常に長い実行時間が必要となる。オペレーティングシステム上で動作する巨大なベンチマークに対するサイクル精度のシミュレーションは、非現実的である。

そこで、我々は DRAM および NVM を搭載したメモリシステムの消費電力予測手法の開発に取り組んでいる。本稿では、第一段階として、対象を DRAM 領域の電力消費

¹ 筑波大学 システム情報工学研究科 コンピュータサイエンス専攻
University of Tsukuba, Tsukuba, Ibaraki 3058573, Japan

² 筑波大学 システム情報系情報工学域

³ 産業技術総合研究所

^{a)} hkawata@cs.tsukuba.ac.jp

に絞る、実機やシミュレーションから簡単に観測可能なパラメータより、DRAM の消費電力を予測するモデル化手法を提案する。特に、メモリの読み書きのスループットに焦点を当て、電力消費との相関関係を明らかにすることで、電力消費のモデル化の手法を提案する。本手法により、サイクル制度シミュレータを用いずとも、容易に計測可能な DRAM に対する読み書きのスループットに基づく、高速な消費電力の見積もりが可能になる。DRAM のスループットは、多くの CPU のパフォーマンスカウンタからオーバーヘッドを最小限に取得可能であり、特定のハードウェアに依存しない、DRAM の電力消費の見積もりを実現する。上記で説明したシステムについては、現在進行中であり、NVM についての電力消費の測定については、本稿では取り扱わないが、同様の手法で予測できると考えている。

本稿では、これらの提案手法における、実験的な評価結果と、その考察を述べる。以下、2 章で研究の背景を述べ、3 章で提案手法の評価結果とその考察を述べる。最後に、4 章において本稿をまとめる。

2. 研究の背景

本章では研究の背景として、ハイブリッドメモリスシステムの概要と、新たな電力評価手法の必要性を説明する。また、DRAM の電力消費に関する予備知識、および提案手法に利用する CPU に内蔵されたパフォーマンスカウンタの情報について述べる。

2.1 DRAM と NVM を組み合わせたメモリスシステム

まず、我々が取り込んでいるハイブリッドメモリスシステムと、その評価における課題について述べる。我々は、限られた容量の NVM と、大きな容量の DRAM を組み合わせることで、大容量かつ低消費電力なメモリスシステムの開発に取り組んでいる。[2] 近い将来には、NVM は DRAM に比べて高価であり、容量も DRAM ほどの容量は実現できない。NVM のみでは、メモリ使用量の大きなアプリケーションに対応できない。そこで、DRAM をスワップ領域の二次的なキャッシュメモリのように利用することで、利用可能なメモリ空間を拡張しつつ、高速に利用できる空間を提供する。本システムでは、NVM 領域のメモリが少なくなると、メモリ管理システムは DRAM へデータをスワップアウトすることで、NVM 上の空き容量を作る。DRAM は高速であるため、従来のストレージに比べ高速にスワップアウトすることができる。

従来のスワップと大きく異なる点は、スワップアウトされたデータをそのまま直接読み書きが出来るという点である。この動作の仕組みを図 1 に示す。スワップアウトされたデータは、DRAM へ配置されているが、DRAM も NVM と同様にメモリバスに直接接続されているため、

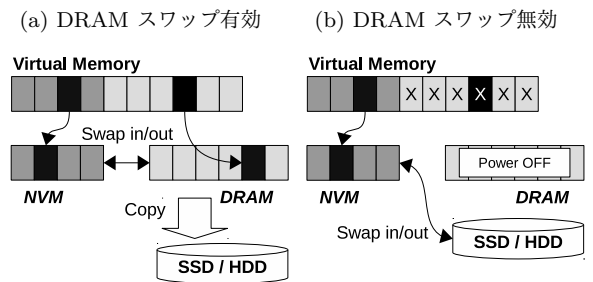


図 1: NVM と DRAM を組み合わせたメモリスシステム

DRAM の電源が投入されている限りは DRAM へスワップアウトされたデータは直接アクセスすることが出来る。関連する研究として、NVM-Swap [4] の実装に非常によく似ているが、DRAM には書き込み回数の制限は存在しないため、書き込みについてもコピーオンライトではなく、そのまま DRAM へ書き込みが可能である。よって、スワップアウトされた場合でも、大きなアクセスコストなしにスワップアウトされたデータへアクセス可能になる。DRAM へスワップアウトされたデータは、一定以上アクセスが起これると NVM 領域へ戻される。NVM には頻繁にアクセスされるデータが集まり、DRAM にはアクセスする可能性が比較的少ないデータが集まることになる。

このシステムでは、メインメモリに搭載した NVM の電源を動的に管理するだけでなく、DRAM の電源も同様に動的に管理することで、消費電力の削減を実現する。しかし、電源を切ると、スワップアウトされた DRAM のデータは失われてしまう。そのため、提案手法では、継続的に他の不揮発性ストレージ (SSD など) に、DRAM へスワップアウトされたデータのコピーを行う。DRAM の電源供給が断たれると、予めコピーを行ってあるストレージをスワップ領域として利用する。その後、メモリ負荷が高くなってきた場合には、DRAM の電源を再び投入して、高速なスワップ領域を有効にする。この手法により、ユーザーの体感速度の低下を最小限に、DRAM の電源管理による電力効率の向上を可能にする。

このようなメモリスシステムを評価する上で、メモリスシステムに対するアクセス性能の他に、電力消費も重要な要素となる。このとき、NVM と DRAM では、メモリ素子の電力消費の傾向が異なることも、大きな問題となる。例えば、STT-MRAM では素子が書き込み時に DRAM に比べ 100 倍以上多くの電力を消費する。この特徴は、NVM への書き込みが非常に多い場合、DRAM の待機時の消費電力を削減できたとしても、システム全体としては消費電力が削減できない可能性がある。よって、メモリスシステム全体の消費電力を、適切に評価する必要がある。しかし、メモリの消費電力を実際に計測することは難しい。また、NVM の実用的な製品はまだ登場していないため、実際の素子を利用した評価が不可能であり、現在は同等のアクセ

ス性能を持つ DRAM を NVM に見立てて評価している。NVmain のような、サイクル精度 CPU シミュレータおよびメモリシミュレータを利用したシミュレーションによる電力消費の評価は、メモリシステムの電力消費の評価を可能にするが、実行時間が問題となる。サイクル精度 CPU シミュレータは、非常に低速であり、オペレーティングシステム上で大規模なベンチマークを動作させるために、非常に長い時間がかかり、非現実的である。よって、実行時間や性能低下を最小限に、もっと手軽に入手できる何らかのパラメータからの、メモリシステムの電力消費の予測が求められている。

2.2 Performance Monitoring Unit (PMU)

提案手法において、メモリの読み取りと書き込みのスループット、および DRAM の電力消費とその他状態の計測のため、Intel の一部 CPU に搭載されている Performance Monitoring Unit (PMU) の情報を利用する。[5] PMU を利用することで、それぞれのメモリコントローラーのチャンネルごとに、読み書きそれぞれの CAS コマンドの回数を取得可能である。CAS コマンドの情報は、多くの Intel 製 CPU などパフォーマンスカウンタとして提供されている。通常、DRAM へはキャッシュラインの単位で CAS コマンドが発行されるため、CAS コマンドの回数にキャッシュラインのサイズを掛けることで、単位時間あたりのスループットを求めることができる。この手法では、プログラムに特殊な改変を加える必要なく、メモリのスループットの取得が可能である。また、このスループットの情報、キャッシュなどの影響を排除したもので、キャッシュを通過した後の実際に DRAM へのアクセスが発生した部分のスループットが観測可能である。さらに、PMU は CPU 内のメモリコントローラーに内蔵されたハードウェアによるパフォーマンスカウンタを利用するため、プログラムやシステムの性能をほぼ犠牲にすること無く計測が可能である。なお、物理メモリアドレスとチャンネルの対応は、通常インターリーピングにより、アクセスが分散されている。CAS コマンドの回数は、メモリのチャンネルごとに取得可能であるが、実際にアクセスをした時にそれぞれのチャンネルへのアクセスが、ほぼ均等に分散されることを確認している。よって、実験ではチャンネルごとの CAS コマンドの回数を、すべてのチャンネルで合計した値を元に議論する。

さらに、Xeon E5 プロセッサ以降では、DRAM の電力消費や省電力モードの情報が PMU より取得可能である。[6] 電力消費については、メモリコントローラーごと、省電力モードの情報はランクやチャンネルごとに取得可能である。これらの情報を元に、DRAM の電力消費と、スループットや省電力モードの情報などの相関関係を明らかにすることで、電力消費のモデル化を行う。しかし、電力消費

や省電力モードの情報を取得できるパフォーマンスカウンタは、一部の限られた CPU でしか提供されていないため、汎用的に利用できるとはいえない。よって、本稿では DRAM の電力消費のモデル化に必要な情報として、電力消費の値や省電力モードの情報を観察するために利用する。これらの、特定のハードウェアに依存する情報を前提としたものではなく、汎用的な環境に対するモデル化を行う。

これらの、PMU の情報は Linux カーネルに付属の perf コマンドを利用して観測することが可能である。perf コマンドは、通常性能解析ツールとして利用され、様々なパフォーマンスカウンタの値を取得可能であり、PMU の情報も一部はデフォルトの状態を取得可能である。しかし、省電力モードなど一部のカウンタのイベントは perf コマンドに初期状態では認識されないため、pmu-tools を利用してイベント情報の解析を行い、専用のパラメーターを与えることで perf コマンドにより取得を行った。perf コマンドでは、これらのパフォーマンスカウンタの値を定期的に取得することが可能で、ベンチマークスクリプトを実行しながら、一定間隔で値を記録することで、スループットなどの計測が可能である。

2.3 DDR3 SDRAM の省電力モード

一般的な DRAM には、電力消費に関連する幾つかの状態が存在する。ここでは、一般的に利用されている DDR3 SDRAM における省電力モードについて述べる。DDR3-SDRAM には、複数の複雑な状態遷移が存在し、その状態により電力消費量が異なる。通常、メモリコントローラーは省電力モードを適切に切り替えることで、パフォーマンスを落とすこと無く出来るだけ低消費電力で動作できるように、メモリモジュールをコントロールしている。よって、電力消費を予測するためには、この省電力モードの切替も考慮しなくてはならない。省電力モードは、それぞれのチャンネルやランクごとに管理されるため、メモリ内に配置されるデータを最適化することで、電力消費を最適化する研究も存在する。[7]

DDR3 SDRAM は、通常の状態ではクロック信号に基づいてコマンドが発行され、発行されたコマンドは常に受け付けられる状態にある。しかし、DDR3 SDRAM ではパワーダウンと呼ばれる省電力モードが存在する。CKE と呼ばれる信号線を Low にすることにより、パワーダウン状態になる。パワーダウンの状態では、クロック信号やリセット信号および CKE 信号自体の入出力は有効であるが、それ以外の入出力が全て不活性になる。よって、パワーダウン状態ではコマンドを受け付けることができなくなり、通常の状態よりもより低消費電力な状態になる。再び、CKE を High にすることで、一定時間後にコマンドの受付を再開することができる。つまり、DDR3 SDRAM では、パワーダウン状態をなるべく長時間維持することで、

電力消費の削減が可能である。しかし、この処理はすべてメモリコントローラーが行っているため、通常ユーザーやソフトウェアが気にする必要はない。

また、パワーダウンに入る時の状態により、幾つかのパワーダウンに入る手法がある。Slow Exit Mode, Fast Exit Mode と呼ばれる、パワーダウン時に DLL と呼ばれるクロック誤差を補正する回路を有効にしたままにするかによる違いによるものがあるが、今回はこれは考慮しないこととした。また、パワーダウン時にバンクをプリチャージしてからパワーダウン状態になる場合と、バンクがアクティブのままパワーダウンになる場合があるが、これも消費電力には影響がないと考え、今回は考慮しないこととした。

更に、パワーダウン以外にも、セルフリフレッシュと呼ばれるモードが存在する。通常のパワーダウンでは、DRAM のデータの保持に必要なリフレッシュ動作は、自動的に行われぬ。よって、定期的にパワーダウンから離脱しリフレッシュを行う必要がある。しかし、セルフリフレッシュモードを利用することで、パワーダウン状態でも外部クロックの入力なしに、自動的にリフレッシュ動作を行うことができる。セルフリフレッシュモードでは、リセット信号と CKE 信号以外の、クロック信号を含めたすべての信号の入力が無効となり、内部クロックが停止される。よって、セルフリフレッシュモードでは、パワーダウン状態より更に消費電力の削減が可能である。

このように、DRAM の電力消費を予測する上で、メモリの読み書き以外に電力消費に影響を与える要素があるので、本研究ではこれらも考慮にいれつつ、省電力モードがどのように直接的な電力消費に影響をあたえるのかを、調査する必要がある。Xeon E5 シリーズに搭載の PMU により、メモリコントローラーのクロック数のうち CKE が有効であったクロック数が、メモリのランクごとに取得できる。また、セルフリフレッシュモードであったクロック数も、チャンネルごとに取得可能であるので、これらの値を利用して、電力消費との相関関係を観察することが可能である。

3. 実験と考察

本章では、DRAM の電力消費の傾向を調査するために実験を行い、DRAM の電力消費とスループット、その他省電力モードの相関関係を計測した結果を議論し、考察を述べる。提案手法では、電力消費のモデル化に利用するパラメータとして、DRAM のスループットを主に利用する。DRAM のスループットは、PMU から取得可能なパフォーマンスカウンタの値を利用することで、性能低下や実行時間の影響を最低限に、取得可能である。実験環境は、CPU として Intel Xeon CPU E5-2650 (SandyBridge-EP) @ 2.00GHz (8 cores, 16 threads) を 2 ソケットで搭載した環境を利用した。メモリは、256 GB の DDR3 SDRAM を

搭載しており、CPU とはそれぞれのソケットに 4 チャンネルで接続されている。

実験では、2 つの種類のベンチマークを利用して、値を取得した。まず 1 つ目に、様々なワークロードをシミュレーションするベンチマークとして、SPEC CPU 2006 の整数ベンチマークを利用した。また別のベンチマークとして、SPEC CPU では再現することが出来ない、よりスループットの高いメモリ帯域での状況を再現するため、stress コマンドを利用した。これらの、2 つの種類のベンチマークにより、メモリアクセスの傾向と電力消費の相関関係を観察した。パフォーマンスカウンタの取得には、perf コマンドを利用した。

3.1 スループットと電力消費の関係

初めに、メモリのスループットが消費電力に影響を与えるのではないかと考え、実験により消費電力との相関関係について観測した。以下に、いくつかのワークロードにおける、スループットと電力消費の関係について示す。

3.1.1 SPEC CPU

まず、SPEC CPU 2006 の整数ベンチマークによる負荷における、メモリの電力消費とスループットの関係について述べる。図 2 は、SPEC CPU 2006 における、メモリアクセスのスループットと電力消費の値を、ソケット別に時系列に並べたものである。横軸がベンチマークの経過時間で、縦軸がメモリアクセスのスループット (MB/s) と電力消費 (J) である。ここでは、SPEC CPU のベンチマークを 1 計測分だけ抜き出している。全体的な傾向として、メモリアクセスのスループットが大きいと、エネルギーの消費も多い事がわかる。

4000 秒付近の結果は、メモリ依存が高いベンチマークである libquantum のワークロードの結果を示している。メモリのスループットはほぼ一定して 11GB/s 程度であり、この時のエネルギー消費量はおよそ 21 J/s である。また、1000 秒から 1500 秒付近の結果は、gcc と mcf のワークロードの結果を示している。この部分では、メモリのスループットのばらつきが多いが、スループットに応じて電力消費に影響を与え、同様にばらついており相関関係にあることがわかる。しかしながら、sjeng のベンチマークにおいて、一部ソケットごとの電力消費が、ソケットごとのスループットに対応していない部分が見える。この現象は、何回かベンチマークを試行した場合に、発生する場合と発生しない場合が存在した。原因としては、numactl によるメモリノードや CPU ノードの割当を行っているが、適切に行われていない可能性がある。

次に、図 3 に、上記の SPEC CPU のベンチマーク 3 計測分における、メモリのスループットと電力消費の相関関係を示す。この図は、それぞれのソケット別のメモリスループットと電力消費の値を、散布図としてプロットした。

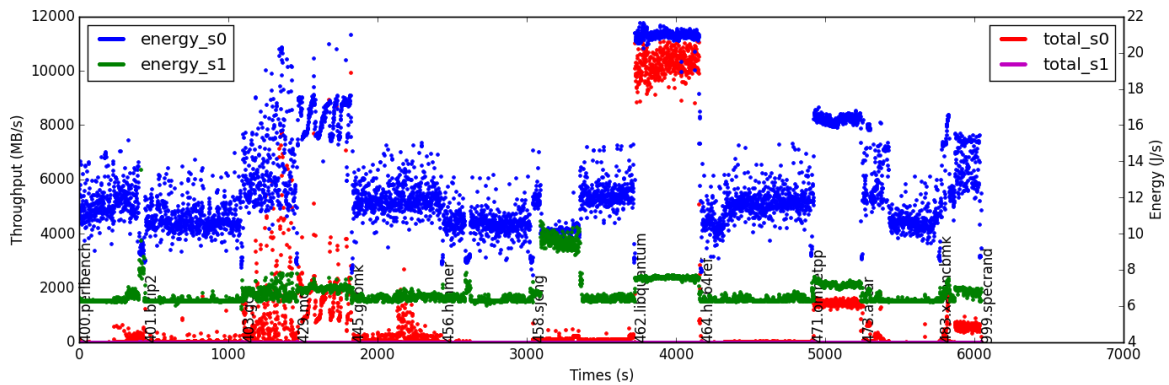


図 2: SPEC CPU における DRAM のスループットと電力消費の推移

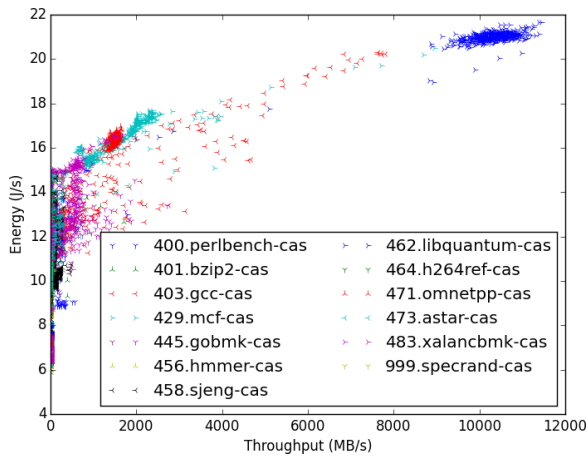


図 3: SPEC CPU における DRAM のスループットと電力消費の相関

横軸がメモリアクセスのスループット (MB/s) で、縦軸が DRAM の電力消費 (J/s) である。この結果から、概ねスループットと電力消費が比例の関係がある事がわかる。

全体の期間を通じて、スループットが 10GB/s 増えると、電力消費が 6J/s 増えている。さらに、スループットにかかわらず消費される電力のベースラインが、ソケットごとに概ね 6 J/s ほどあることがわかる。また、スループットが非常に低い部分では、電力消費が下方向に散らばっている。省電力モードが関連していると考えられるため、別の実験を行うこととした。その結果については、3.2 章で述べる。

3.1.2 stress

次に、更にメモリへの負荷が高い状況を再現するため、stress コマンドを利用してメモリへ負荷をかけた時の、メモリのスループットと電力消費の関係を観察した。図 4 は、この時のメモリアクセスのスループットと電力消費の値を時系列に並べたものである。この結果は、stress コマンドのワーカー数 (--vm) を 60 とし、(--vm-size) を 2GB に設定した。--vm-stride は 1 に設定し、--vm-hang を 10 に設定し 60 秒間計測した。これにより、それぞれのワーカーは 2GB の範囲をアクセスし 10 秒間休む、とい

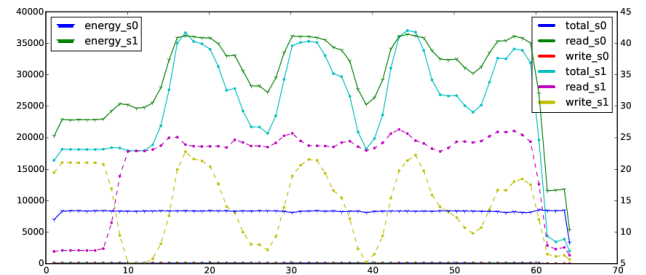


図 4: stress における DRAM のスループットと電力消費の推移

う動作を繰り返す。また、この実験では numactl により、memnode と cpunode を固定し計測した。stress コマンドに与えたパラメータは以下の通りである。

```
$ stress -t 60 --vm 60 --vm-bytes 2G
--vm-hang 10 --vm-stride 1 --vm-keep
```

メモリのスループットを示す CAS コマンドの回数は、読み書きを合計した値に加えて、読み書きそれぞれの値も示している。SPEC CPU 2006 の結果と比較すると、全域にわたって非常にスループットが高いが、結果は同様にメモリのスループットに依存して、電力消費が変動していることがわかる。また、CAS コマンドの内容の読み書きにかかわらず、それらを合計した回数が電力消費に依存していることがわかる。読み書きの種類は、電力消費に特に影響を与えないように見える。

図 5 は、stress コマンドにより負荷をかけた時の CAS コマンドの発行回数から算出したスループットと、電力消費の関係をプロットしたものである。この結果は、上記の結果に加え、stress コマンドのワーカー数を 60 と 8 のそれぞれのワーカー数で計測し、ワーカーごとのアクセス範囲を 64MB から 2GB の間で変更して計測したものを散布図にした。このグラフから、メモリスループットに応じて、電力消費が向上しており、ほぼ比例関係にあることがわかる。概ね、スループットが 10GB/s 上昇すると、電力消費が 8J/s ほど上昇していることが確認できる。しかし、

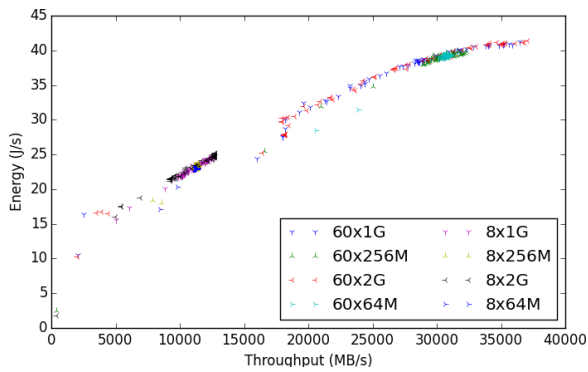


図 5: stress における DRAM のスループットと電力消費の相関

30GB/s 以上のスループットでは、電力消費が頭打ちとなり、スループットが上昇しても電力消費の上昇率が鈍っている事がわかる。また、SPEC CPU との実行結果 (図 3) と比べても、特にその相関関係にはそれほど差がないように見える。よって、メモリのスループットは、DRAM の消費電力の見積もりに利用可能な重要な要素であると言える。

3.2 省電力モードと電力消費の関係

次に、省電力モードと電力消費の関係について示す。こちらにも、SPEC CPU と stress の 2 つのワークロードを利用して、実験と観測を行った。

3.2.1 SPEC CPU

3.1 章では、SPEC CPU ベンチマークにおいて、スループットが低い部分で電力消費が分散する結果が得られた。これは、省電力モードによる影響と考えたため、さらに省電力モードと電力消費の相関関係を観測する実験を行った。

図 6 は、SPEC CPU ベンチマークにおける、CKE 信号が ON である間のクロック数の割合と、電力消費の相関を示したものである。CKE 信号が有効であり、ほとんど省電力モードになっていない状態での電力消費の最小値は、およそ 13J/s である。それ以下では、ほぼ CKE が有効である割合に電力消費が比例していることが分かる。最小値は、およそ 6J/s である。図 3 と比べると、スループットが 0 の時におよそ 13J/s に以下に収まっていることが分かる。つまり、スループットが非常に低い部分では、省電力モードが電力消費に影響を与えたと考えられる。しかし、省電力モードはスループットからはほぼ予測が不可能である。また、一部ワークロードの結果は、その他のワークロードの値から外れていることが分かる。特に、sjeng ワークロードで顕著であるが、上記で述べた numactl の設定不足の可能性もある。

3.2.2 stress

SPEC CPU ベンチマークにおいては、メモリへ継続的にアクセスが発生するワークロードがほとんどである。他

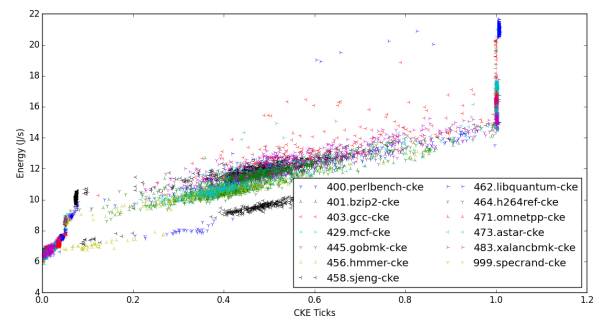


図 6: SPEC CPU における CKE と電力消費の相関

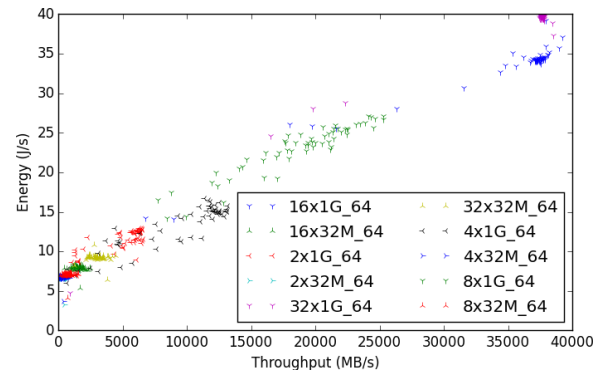


図 7: 間欠的にメモリアクセスが発生する場合の DRAM のスループットと電力消費の相関

のアクセスパターンの傾向を探るため、stress ベンチマークを利用して実験を行った。特に、間欠的にメモリアクセスが発生するようなアクセスパターンについて調査した。stress ベンチマークにおいて、`--vm-hang` の値を 1 に設定し、それぞれのワーカーが指定されたメモリサイズアクセスが行い 1 秒休むといった動作を繰り返す状態を再現した。またこの実験では、効率的にメモリにアクセスしスループットを高めるため `--vm-stride` を、キャッシュラインサイズである 64 に設定した。この時の、スループットと電力消費の相関関係を、図 7 に示す。ワーカーの数とメモリサイズをいくつか条件を用意し、実験を行った。

この結果を、図 3 や、図 5 と比べると、スループットに対する消費電力の傾向が異なることがわかる。図 7 の結果は、その他の結果に比べてスループットあたりの電力消費が低く抑えられている。この現象が起きる原因として考えられるのが、省電力モードである。そこで、stress によるワークロードにおける、CKE の割合を調査した。図 5 の実験における CKE の割合を図 8 に示し、図 7 の実験における CKE の割合を図 9 に示す。この結果から、図 9 の実験である場合に、より省電力モードである時間が長い事がわかる。つまり、継続的にメモリへのアクセスが発生する場合は省電力モードになることが出来ないが、間欠的に一気にメモリアクセスがある場合とメモリアクセスがない場合とが繰り返される状態では省電力モードに長く留まることができるため、消費電力が相対的に少なくなる。単

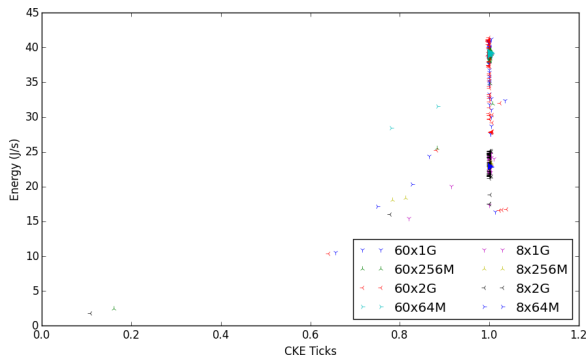


図 8: 連続的にメモリアクセスが発生する場合の省電力モードと電力消費の相関

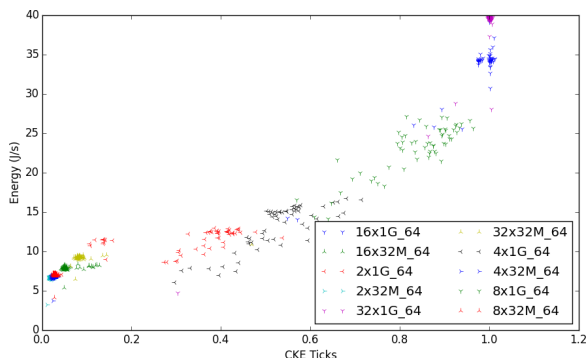


図 9: 間欠的にメモリアクセスが発生する場合の省電力モードと電力消費の相関

位時間あたりで見た場合と同じスループットであっても、アクセスパターンによって消費電力が異なる場合があるということである。この現象の影響を最小限に抑えるためには、スループットの計測間隔をなるべく小さくすることで影響を少なくすることができる。しかし、省電力モードの切り替えは、非常に高速に行われているため同等の間隔で、スループットの計測を行うことは難しい。

3.3 アクセス範囲と電力消費の関係

次に、ワークロードが使用するメモリ量、つまりアクセスする範囲が電力消費に影響を与えるのではないかと考え、その関係性を調べた。まず、SPEC CPU ベンチマークにおける、電力消費量とそれぞれのワークロードのメモリアクセス範囲の相関関係を得るため、読み書きが行われたページ数を観察した。図 10 が、SPEC CPU の 1 回のベンチマーク試行における、読み書きが行われたページの変動を示したものである。横軸が経過時間で、縦軸が単位時間あたりにアクセスまたは書き込みが行われたページ数を示している。この結果から、SPEC CPU では局所的にメモリがアクセスされるワークロードと、幅広くメモリがアクセスされるワークロードが混在していることがわかる。この結果を、図 3 と比べると SPEC CPU ベンチマークでは、メモリのスループットとアクセスしている範囲は相関があまりないことがわかる。詳しく見ていくと、mcf

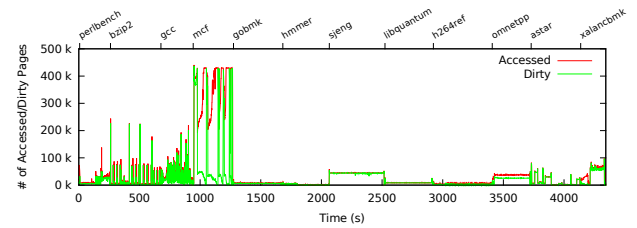


図 10: SPEC CPU におけるアクセス済みおよび書き込み済みページ数

ワークロードでは、アクセスしているメモリの範囲は広いが、電力消費傾向は他と変わりがない。また、libquantum ワークロードでは、メモリのスループットが最高が、アクセスしているメモリの範囲は他のワークロードに比べて狭い。どちらのワークロードも、電力消費傾向は他のベンチマークと変わりがないように見える。よって、アクセスしているメモリの範囲と電力消費は、それほど関連がない事が分かった。

4. まとめ

本稿では、実験により DRAM のスループットおよび省電力モードに関連する値と、DRAM の消費電力を実際に測定し、その相関関係を観察した。すべての結果において、メモリのスループットと DRAM の電力消費には高い相関関係があることが分かった。また、一定以上のスループットでは、電力消費とメモリのスループットは概ね比例関係にあることが分かった。DRAM はメモリアクセスがほとんど発生しない状況では、ある一定の消費電力であることが分かった。このことから、メモリのスループットから、電力消費をモデル化し見積もりを行うことは可能であると考える。メモリのスループットは、多くのハードウェアでパフォーマンスカウンタとして提供され、容易に計測可能であるため、提案手法を利用したモデル化を行うことで汎用的な電力消費の見積もりが可能になる。しかし、ワークロードやアクセスパターンの違いによって、電力消費の傾向が異なることがわかった。特に、スループットが低い領域では、DRAM の電力消費に省電力モードが深く関わっている。よって、より正確な DRAM の電力消費の予測には、スループットだけでなく省電力モードの状態の観察または予測が求められる。

今後の方針として、より詳細な DRAM の挙動を観察するため、サイクル精度シミュレータである Gem5 と DRAMSim [3] を組み合わせた環境において、電力消費のシミュレーションを行う予定である。メモリシミュレータで取得した結果を、今回 PMU により測定した結果と比較することで、本論文の手法の正当性を確かめるとともに、問題点について更に調査する。また、次世代不揮発性メモリについても同様に、不揮発性メモリのシミュレータであ

る NVmain [8] などのメモリシミュレータを組み合わせて、電力消費モデルの分析を行う予定である。

参考文献

- [1] for Semiconductors (ITRS), I. T. R.: *The International Technology Roadmap for Semiconductors 2013 Edition* (2013).
- [2] Kawata, H. and Oikawa, S.: Experimental design of high performance non volatile main memory swapping using DRAM, *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on*, pp. 1–6 (2015).
- [3] Rosenfeld, P., Cooper-Balis, E. and Jacob, B.: DRAM-Sim2: A Cycle Accurate Memory System Simulator, *Computer Architecture Letters*, Vol. 10, No. 1, pp. 16–19 (2011).
- [4] Zhong, K., Wang, T., Zhu, X., Long, L., Liu, D., Liu, W., Shao, Z. and Sha, E. H.-M.: Building High-performance Smartphones via Non-volatile Memory: The Swap Approach, *Proceedings of the 14th International Conference on Embedded Software*, EMSOFT '14, New York, NY, USA, ACM, pp. 30:1–30:10 (2014).
- [5] Intel: *Intel 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide, Part 2* (2015).
- [6] Intel: *Intel Xeon Processor E5-2600 Product Family Uncore Performance Monitoring Guide* (2012).
- [7] Wu, D., He, B., Tang, X., Xu, J. and Guo, M.: RAMZzz: Rank-aware Dram Power Management with Dynamic Migrations and Demotions, *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, Los Alamitos, CA, USA, IEEE Computer Society Press, pp. 32:1–32:11 (2012).
- [8] Poremba, M. and Xie, Y.: NVMain: An Architectural-Level Main Memory Simulator for Emerging Non-volatile Memories, *VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on*, pp. 392–397 (2012).