不揮発主記憶サーバの性能モデルと省電力化の検討

湯木 大輝[1,a)]　坂本 龍一[1]　中村 宏[1]

概要：不揮発メモリを主記憶として利用する NVMM（Non-Volatile Main Memory）の登場により，主記憶の大容量化が期待されている．そこで NVMM の大容量性を活かし VM（仮想マシン）を少数台の PM（物理マシン）上に集約することで，データセンタの省電力化を目指す．しかし，不揮発メモリは揮発メモリと比較し，アクセス時のレイテンシや消費エネルギーが大きい．この課題に対し，VM ごとのボトルネックを分析し，複数の VM が NVMM 搭載 PM 上に共存するときの VM 性能低下の予測モデルを提案する．またこのモデルを用いた VM の配置最適化による省電力化を検討する．

1. はじめに

近年，企業におけるクラウドサービス活用の進展などに伴い，データセンタの利用が拡大している．これに伴い，データセンタにおいて消費される電力は莫大なものとなり，さらに増加を続けている．2015 年に世界のデータセンタが消費したエネルギーは 416 TWh [1] であり，今やデータセンタは世界の発電量の約 3 %を消費し，引き起こした温室効果ガスの排出量は航空産業に匹敵する．そのため，データセンタの消費エネルギーを削減することは，データセンタの運営者が支払う電気料金を減少させるだけでなく，温室効果ガスの排出を抑制することで地球環境の悪化に与える影響を小さくすることにつながる．

また，不揮発メモリに関する技術開発が進んだことで，これを主記憶として用いる NVMM（Non-Volatile Main Memory）が提案されてきた．不揮発メモリはデータ保持のために電源の供給を必要としないメモリであり，省電力化効果が期待されている．さらに，スケールアップの限界が近い DRAM に比べて，NVMM はその大容量性に特徴がある．NVMM をデータセンタに用いることで，稼働するサーバの数を削減できる可能性がある．具体的にはデータセンタ内で動作する VM(仮想マシン) を NVMM が搭載された大容量主記憶サーバー上に集約することで PM(物理マシン) の数を減らすことができる可能性がある．このように稼働する PM の台数を削減することで，省電力効果が期待できる．

しかしながら，不揮発メモリは揮発メモリと比較し，アクセス時のレイテンシが大きい．そのため，主記憶に対するアクセスが多い VM は大きく性能が低下する恐れがあり，VM の特性を考慮した VM 配置が重要となる．そこで，本研究ではデータセンタにおいて実行される VM がネットワークや CPU 時間等，異なるボトルネックを持つと考えられることを利用する．様々なボトルネックを分析し，VM 間での限られた資源の奪い合いが与える影響をなるべく小さくすることで，NVMM 導入による性能の悪化を抑制する．そのために，既存手法を参考に Interference の考え方に基づいた性能低下の予測モデルを提案する．

本研究の貢献は，

- NVMM の大容量性を生かし，VM を集約することによるデータセンタの省電力化手法を提案
- NVMM 搭載 PM 上で複数の VM を動作させた際の性能低下の指標となりうる Interference モデルを提案
- NVMM 搭載 PM 上に VM を集約することによる省電力効果について検討

することである．

これらの手法と結果について，以下のように論じていく．2 章では前提となる NVMM および，ウェブサービスを実行するデータセンタとその性能向上手法について概観する．3 章では研究目標を示し，ボトルネック分析に基づく性能低下の予測モデルを提案する．また，必要となる評価について述べる．4 章では実機を用いた評価結果を示し，考察と電力削減に向けた簡単な試算を行う．5 章で本研究のまとめと，今後の課題について述べる．

2. 研究背景と関連研究

2.1 不揮発メモリによる主記憶容量拡張

データを保持するために電源の供給を必要としないメモリを不揮発メモリという．主な不揮発メモリの特性を，揮発

[1] 東京大学大学院情報理工学系研究科
[a)] yuki@hal.ipc.i.u-tokyo.ac.jp

表 1　不揮発メモリの特性

	記憶密度	レイテンシ（Read/Write）	エネルギー（Read/Write）
Flash	4X	20,000/200,000[ns]	110/790[pJ/bit]
PCM	2X-4X	50/150[ns]	314/1635[pJ/bit]
DRAM	1X	30/30[ns]	15/16[pJ/bit]

メモリの DRAM との比較で**表 1** に示す [2] [3] [4] [5] [6] [7].

不揮発メモリのインタフェースをバイトアドレッサブルにしたものは主記憶として使用でき，これを NVMM（Non-Volatile Main Memory）という．NVMM の主な利点は大容量性である．大容量の DRAM を搭載したシステムではデータを保持するためのリフレッシュ処理による電力消費量が無視できなくなり，DRAM 自体の微細化によるスケールアップも限界が近い．これに対して NVMM はリフレッシュが不要であり，記憶密度も高められている．特に PCM は Flash と比較してレイテンシが小さいため，NVMM として使用できる可能性がある．実際にインテルとマイクロンによって開発され，既に製品化もされて NVMM として利用できる 3D XPoint は，PCM の一種であると言われている．しかしながら，従来主記憶に用いられてきた DRAM と比較した場合には，レイテンシや読み書き時の消費エネルギーが大きい．

そこで，これらの欠点を和らげるため，主記憶として DRAM と NVMM を併用することが提案されてきた．高速で小容量の DRAM は，低速で大容量の NVMM に対してキャッシュとして利用できる．DRAM と NVMM の協調にあたっては，2 階層それぞれの長所を生かす工夫が必要となる．具体的には NVMM の大容量の恩恵を受けつつ，DRAM の活用でレイテンシや消費電力の増加に対処する．例えば Huang ら [8] は，NVMM を用いた KVS（Key-Value Store）において DRAM をキャッシュとして用いることで，DRAM のみを使用した場合に近い短さのレイテンシと高いスループットを実現した．

2.2　ウェブサービスとデータセンタ

データセンタでは様々なウェブサービスが実行されている．ここでウェブサービスとはインターネットを介してデータのやり取りや計算能力の提供を行うようなサービスを総称し，具体的には以下のような例を想定する．

(1) KVS（Key-Value Store）は Key とそれに対応する Value からなるデータベースである．データを入れる（SET）際には Key と Value の組を与える．取り出す（GET）際には Key を指定するとそれに対応する Value が返る．Key を指定してデータを削除する（REMOVE）こともできる．

(2) HTTP はハイパーテキストなどの転送に用いられる通信プロトコルである．データを指定して Web サーバから取り出す（GET）動作が基本だが，クライアントからサーバへの送信（POST）も可能である．

(3) 機械学習においては近年，ニューラルネットワークが盛んに用いられている．学習によって作成済みのモデルに新たなデータを当てはめる推論の際には，行列積の計算に代表される積和演算が主体となる．

近年の通信インフラの発達やスマートフォンの普及を背景に，電子商取引や SNS（Social Networking Service），ゲームのようにインターネット経由で提供されるビジネスは増加している．また，企業における業務システム等のクラウド化も進展している．これらは開発の効率化や需要に応じた資源利用を実現するため，データセンタ上に構築されたウェブサービスの集合体として実現されることが多い．従ってデータセンタは，KVS や HTTP，機械学習の推論といったウェブサービスを，複数の顧客のため並列に多数実行することになる．

これらのサービスに対してリソースを共用しつつ動作環境を分離し，セキュリティを確保するため，それぞれのウェブサービスは別の VM（仮想マシン）に分離される．1 台のサーバ，すなわち PM（物理マシン）上では複数の VM が動作し，各々にあらかじめ割り当てられた範囲内で PM の資源を使用する．

2.3　VM 性能低下予測に基づく配置最適化

データセンタの消費エネルギー削減や，負荷分散を目的として，PM 間で VM を移動することが一般に行われる．この移動をマイグレーションという．VM を適切な PM に割り当て，または移動することで，データセンタ全体での省エネルギー化を図ることが可能である．ただし，各 VM で稼働するサービスへの影響に留意し，極端な性能低下を防ぐ必要がある．

iAware [9] は，マイグレーションによって発生する VM の性能低下を考慮したマイグレーション戦略を提案した．性能の低下要因を Interference として分析し，マイグレーション作業自体に起因する Migration Interference と，移動先で他の VM と共存することにより発生する Co-Location Interference を評価する．これらの Interference が小さくなるように VM をマイグレーションしていくことで，VM の性能低下を抑えられるような配置を達成する．

Interference は Demand-Supply モデルに基づいて構成される．すなわち式（1）のように，PM で利用可能な資源量（Supply）に対する VM の要求量の和（Demand）の比率として Interference を定義する．ある VM をマイグレー

ションした場合の性能低下は，マイグレーション後の各資源に対する Interference の重み付き和によって表現する．このときの係数を，各 Interference をマイグレーション候補の全 VM に対する最大値で正規化するように定めることで，ボトルネックの解消を重視する．

$$\text{資源の Interference} = \frac{\text{Demand:VM 要求量の和}}{\text{Supply:PM の提供可能資源量}} \quad (1)$$

さらに，PM によって性能が異なる場合には，それを考慮して性能低下を予測する必要がある．このような評価は様々な条件のもとで行われてきている．山田ら [10] は CPU 性能やリクエストサイズを変化させたときの KVS や HTTP サーバの動作評価を行い，Lim ら [11] は CPU やネットワークの性能を変化させたときの KVS サーバの動作評価を行った．

しかし，DRAM のみをもつ PM と，DRAM に加えて NVMM を搭載した PM とでこのようなウェブサービスの性能を評価，比較した研究はこれまで存在しなかった．なお，NVMM に対して DRAM をキャッシュとして用いるという構成から，従来の CPU キャッシュを考慮した VM マイグレーション手法は参考になると考えられる．Kim ら [12] のようにキャッシュミス率を計測し，キャッシュミスの多い VM とキャッシュミスによる性能低下が起きやすいと考えられる VM が同一の PM に割り当てられないようマイグレーションを行うものや，Chen ら [13] のようにシミュレータを用いて VM が単独で動作するときの各キャッシュラインへのアクセス頻度を取得しておき，複数 VM が配置された PM で発生するキャッシュミスを正確に予測するものなどが挙げられる．

3. 研究目的と提案手法

3.1 研究目的

従来，DRAM を搭載した複数台の PM で動作していた VM を，NVMM を追加したより少数台の PM 上に集約することで，データセンタの省電力化を行う．NVMM の大容量性を活かして，**図 1** に示す例のように NVMM を搭載した PM1 台あたりで動作する VM 数を増やす．そして空いた PM の電源を落とすことで，データセンタ全体での消費電力減少を目指す．

3.2 ボトルネック分析に基づく性能低下の予測モデル

Interference の考え方を用い，個々の性能低下を抑えつつ VM を集約することを検討する．既存の Interference に加えて，PM に NVMM を搭載したことで新たに生まれるボトルネックを分析し，VM の配置に反映する必要がある．iAware を参考に，VM の性能低下に基づいて Interference を定める．

本研究では少数台の PM に VM を配置して省電力化を目指すという前提から，VM の仮想 CPU コアはそれぞれ異

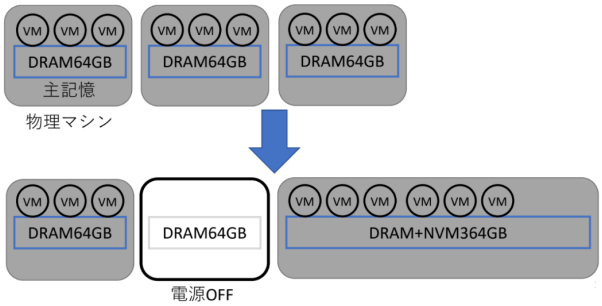

図 1 NVMM を追加した PM への VM 集約による省電力化

なる PM の CPU コアに割り当てると仮定する．このとき CPU Interference を考慮する必要はないので，ネットワーク，DRAM，NVMM の Interference を定めればよい．

まずネットワーク Interference（$\gamma_{i,d}$）は，ネットワークのバンド幅がボトルネックになると予想して式（2）のように提案する．なお添字の i はマイグレーションする VM，d はマイグレーション先の PM を示す．

$$\gamma_{i,d} = \frac{\text{Demand:VM のネットワークバンド幅の和}}{\text{Supply:PM のネットワーク容量}} \quad (2)$$

次に DRAM Interference（$\xi_{i,d}$）は，DRAM アクセスのバンド幅がボトルネックになると予想して式（3）のように構成する．

$$\xi_{i,d} = \frac{\text{Demand:VM の L3 キャッシュミス数の和}}{\text{Supply:PM の L3 キャッシュ容量}} \quad (3)$$

最後に NVMM Interference（$\eta_{i,d}$）は，NVMM アクセスのバンド幅がボトルネックになると予想して式（4）のように定める．

$$\eta_{i,d} = \frac{\text{Demand:VM の時間あたり NVMM 参照回数の和}}{\text{Supply:PM が提供可能な時間あたり NVMM 参照回数}} \quad (4)$$

これらの要素から，式（5）のように Interference を求める．このとき係数 $\kappa_\gamma, \kappa_\xi, \kappa_\eta$ は Interference の各要素をマイグレーション候補となる全ての VM に対する最大値で正規化するような値とする．

$$\text{Interference}_{i,d} = \kappa_\gamma \times \gamma_{i,d} + \kappa_\xi \times \xi_{i,d} + \kappa_\eta \times \eta_{i,d} \quad (5)$$

この $\text{Interference}_{i,d}$ が最小となるような VMi を PMd へマイグレーションし，一定時間後に改めて性能測定を行って Interference を算出する，という繰り返しによって性能を保ちつつ PM の稼働台数を減らすことを目指す．

3.3 必要となる評価

提案手法の実現のためには，次に示す 3 つのステップが求められる．

(1) 資源の Interference である $\gamma_{i,d}, \xi_{i,d}, \eta_{i,d}$ が，対応するボトルネックを反映して性能低下と連動する．

(2) 算出された $\text{Interference}_{i,d}$ が最小となるような VMi と PMd の組み合わせが，VM の性能低下を抑制でき

表 2　評価環境

	DRAM+NVMM 機	DRAM のみ機
CPU	Intel Xeon E5-2630 v4	同左
L3 キャッシュ	25600kiB	同左
DRAM	DDR4 16GiB x4	同左
NVMM	Intel Optane 900P 280GiB	なし

るようなマイグレーションを実現する．

(3) VM を追い出した PM の電源を切ることで，マイグレーション開始前と比べてデータセンタ全体での消費エネルギーを削減できる．

本研究では初期評価として主に（1）を確認する．（3）に向けた試算も 4.5 節で簡単に示す．

4. 実験と結果

4.1 実験の概要

Interference のモデルを検討するために，実機実験を行う．

構成の異なる 2 台の PM 上で，ウェブサービスを稼働させる．評価環境を**表 2** に示す．DRAM+NVMM 機では 64 GiB DRAM に加えて NVMM も搭載し，主記憶の合計容量は 342 GiB となった．この NVMM は PCM の一種とされる 3D XPoint を用いたもので，PCIe x4 により接続する．DRAM のみ機には 64 GiB DRAM のみを搭載した．

PM 内に同一のサービスを実行する VM を複数立ち上げ，他のマシンからリクエストを並列に送信する．リクエストサーバとの間は 10GBASE-T で接続した．同時に立ち上げる VM 数を変化させたときのネットワーク転送データ量，L3 キャッシュのヒット率とミス数，主記憶使用量とスワップ使用量，そして各リクエストの完了までにかかる時間を測定する．

ベンチマークとして送信するリクエストは**表 3** の通りとする．KVS および HTTP はネットワーク，Vmul および Sadd は DRAM または NVMM に起因するボトルネックが現れることを期待して設定した．なお KVS，HTTP と Vmul のサイズ 15k では VM 数最大でも DRAM 内に全てのデータが収まるが，それ以外の条件で多数の VM を起動した場合には DRAM の容量を超える．

ベンチマークの詳細について述べる．KVS は VM 内で memcached を起動し，100kB の文字列を 1 データの Value として，1000 個のデータに各々異なる Key を割り当ててあらかじめ SET する．このデータを先頭から順次全て GET したときの性能を測定する．HTTP は 100MB のファイルを VM 内に置き，あらかじめ Apache HTTP Server を起動する．このファイルを curl によって 1000 回反復して取得したときの性能を測定する．Vmul は VM 内に 15,000 次または 30,000 次の double 型正方行列を用意し，各要素を異なる値で初期化する．そして片方の行列はランダムな 200 行，もう片方の行列からは 200 列を選び出し，行と列の組み合わせ全てに対してベクトルの内積を計算したときの所要時間を測定する．Sadd は 8GB または 16GB の double 型数列を用意する．この数列の先頭から全ての要素に 1.0 を加算することを 10 反復繰り返したときの所要時間を測定する．

表 3　ベンチマークセット

サービス	メソッド	サイズ	データ数	反復数
KVS	GET	100kB	1000	1
HTTP	GET	100MB	1	1000
Vmul	\	15k/30k	200^2	1
Sadd	\	8GB/16GB	1	10

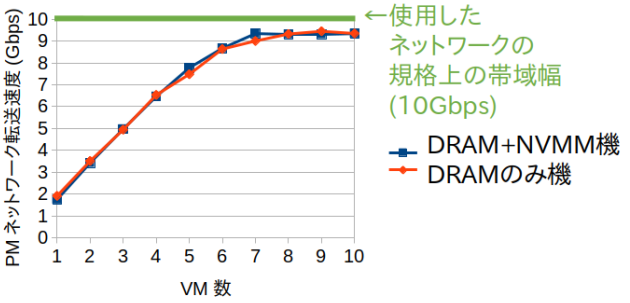

図 2　KVS で VM 数増加時の PM のネットワーク転送速度

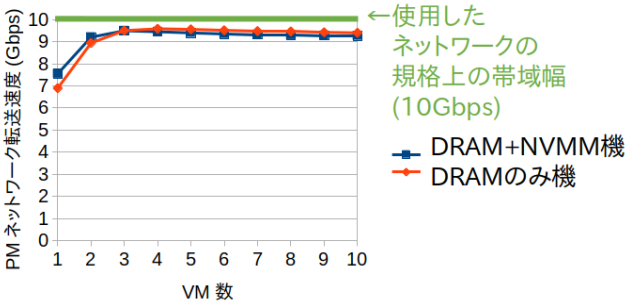

図 3　HTTP で VM 数増加時の PM のネットワーク転送速度

4.2 ネットワークバンド幅がボトルネックのときの結果

測定の結果について，まずネットワークバンド幅がボトルネックになると予想される KVS と HTTP を実行した際の PM のネットワーク転送速度を**図 2** と**図 3** に示す．KVS では 7 台以上の VM が稼働したとき，HTTP では 2 台以上の VM が稼働したときに PM のネットワーク容量が飽和していることが伺える．

このときリクエストあたりの実行時間は，各 VM で見た単位データあたりのデータ転送時間に比例し，**図 4** と**図 5** のようになった．よってネットワークバンド幅が性能のボトルネックとなっており，式（2）のような Interference によって性能低下を反映できると考えられる．

4.3 DRAM アクセスバンド幅がボトルネックのときの結果

次に DRAM 参照について，DRAM アクセスバンド幅が

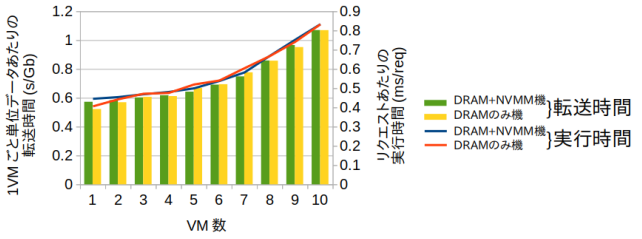

図 4　KVS で VM 数増加時の単位データ転送時間と実行時間の関係

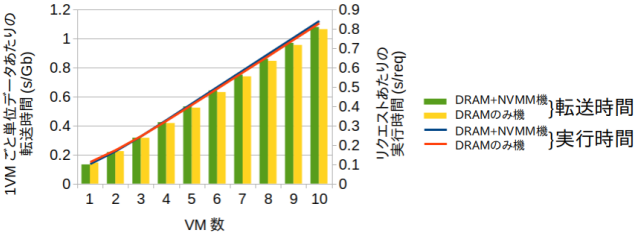

図 5　HTTP で VM 数増加時の単位データ転送時間と実行時間の関係

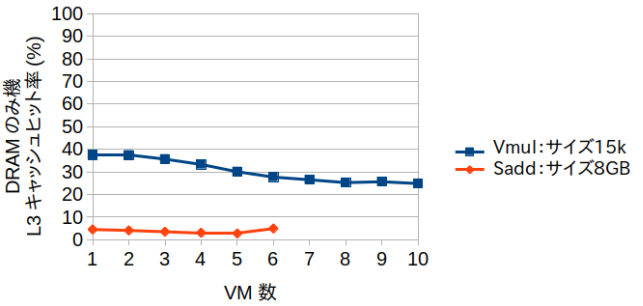

図 6　VM 数増加時の DRAM のみ機の L3 キャッシュミス率

ボトルネックになると予想される条件での PM の L3 キャッシュヒット率を**図 6** に示す．具体的には Vmul のサイズ 15k と，Sadd のサイズ 8GB でスワップが発生しない VM6 台までである．なお，キャッシュミスは DRAM+NVMM 機で測定することができなかったため，DRAM 機での測定結果を用いている．L3 キャッシュヒット率が低く，VM 数を増加させるとさらにヒット率は低下して，多くのデータアクセスが DRAM 参照を引き起こしていることが分かる．

このとき Vmul のサイズ 15k で，実行時間は 1 台の VM あたりの L3 キャッシュミス数と同様の傾向を示し，**図 7** のようになった．L3 キャッシュミスによって起きる DRAM 参照が性能低下を招いたと考えられる．一方で Sadd のサイズ 8GB は，VM 数を増やしても 1VM あたりのキャッシュミス数はほとんど変化せず，従って実行時間との関連も見られなかった．元々キャッシュヒット率が極めて低く，それ以上キャッシュミスが増えない状況下においては，キャッシュミス数に基づいた性能低下予測は有効でないと言える．

さらに，アクティブメモリ量と時間あたりキャッシュミス数の関係を分析する．ここでアクティブメモリ量

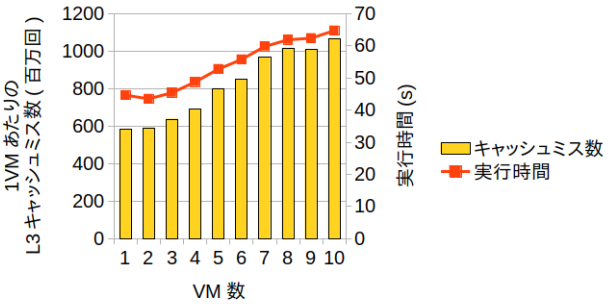

図 7　Vmul のサイズ 15k で VM 数増加時の L3 キャッシュミス数と DRAM のみ機での実行時間の関係

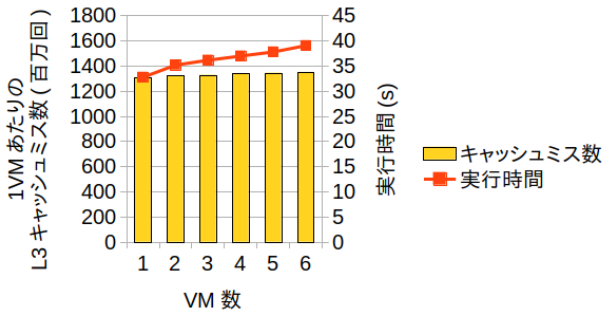

図 8　Sadd のサイズ 8GB で VM 数増加時の L3 キャッシュミス数と DRAM のみ機での実行時間の関係

とは，/proc/meminfo で得られる主記憶利用量のうち，Active(Anon) で示される現在プロセスが使用中の主記憶の容量を言う．また時間あたりキャッシュミス数は，DRAM のみ機でスワップが発生しないような領域では DRAM+NVMM 機と DRAM のみ機で同様のキャッシュミスが起こると仮定したとき，1 台の VM あたりのキャッシュミスが単位時間あたり何回発生しているかを計算したものである．

Vmul を実行したとき，スワップの発生しないサイズ 15k，およびサイズ 30k の VM4 台までで，アクティブメモリ量と時間あたりキャッシュミス数の関係は**図 9** のようになった．ただしグラフ中のラベルは VM 数を示す．行列のサイズにかかわらず，DRAM+NVMM 機と DRAM のみ機でそれぞれの傾向を示していることが分かる．DRAM アクセスのバンド幅が飽和しており，しかもその容量は PM に NVMM を搭載しているか否かで異なると解釈することができる．

Sadd を実行した場合は，スワップの発生しないサイズ 8GB の VM6 台までとサイズ 16GB の VM3 台までで，アクティブメモリ量と時間あたりキャッシュミス数の関係は**図 10** のようになった．DRAM+NVMM 機と DRAM のみ機で傾向が異なるのはサイズ 8GB の VM3 台以上のときのみとなっており，他の条件では時間あたりキャッシュミス数はあまり変わらない．アプリケーションごとに時間あたりキャッシュミス数の上限が異なるだけでなく，NVMM

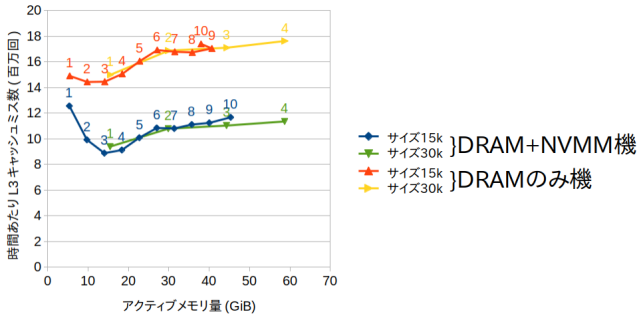

図 9 Vmul で VM 数増加時の L3 キャッシュミス数と実行時間の関係

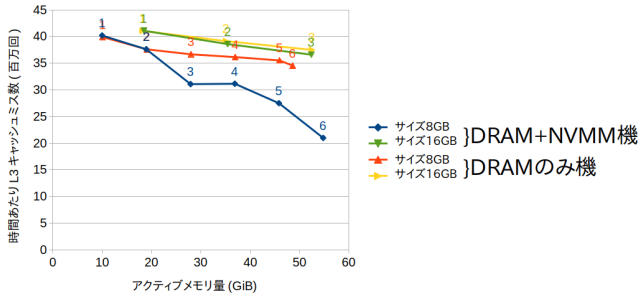

図 10 Sadd のサイズ 8GB で VM 数増加時の L3 キャッシュミス数と実行時間の関係

を搭載したことによる性能変化が出現するか否かも条件次第だと考えられる．

以上から，複数 VM を同一 PM に配置することでキャッシュヒット率が低下していくとき，式 (3) のような Interference で性能低下を反映することができる．ただし DRAM アクセスバンド幅の上限は VM 内で実行するアプリケーション，PM への NVMM 搭載有無などによって変化するため，ボトルネックを明確に予想するためにはさらなる研究が必要と言える．

4.4 NVMM アクセスレイテンシがボトルネックのときの結果

最後に NVMM の Interference を検討する．そのために，性能低下要因を DRAM 参照と NVMM 参照に分解することを考える．DRAM の参照によっても実行時間は増加するため，NVMM 参照が発生するときとそうでないときで伸び率が異なれば NVMM がボトルネックと言える．DRAM が NVMM に対するキャッシュとして動作するとき，Vmul および Sadd についてはデータアクセスのパターンからいつ NVMM への参照が発生するかを予想することができる．ベンチマークセットの中で NVMM アクセスが発生すると予想される条件のものについて，1 台の VM あたりで何ページ分のデータに対して NVMM 参照が発生するかの予想値と，実際に測定された実行時間とを比較した．結果を Vmul のサイズ 30k は**図 11**，Sadd のサイズ 8GB は**図 12**，Sadd のサイズ 16GB は**図 13** に示す．

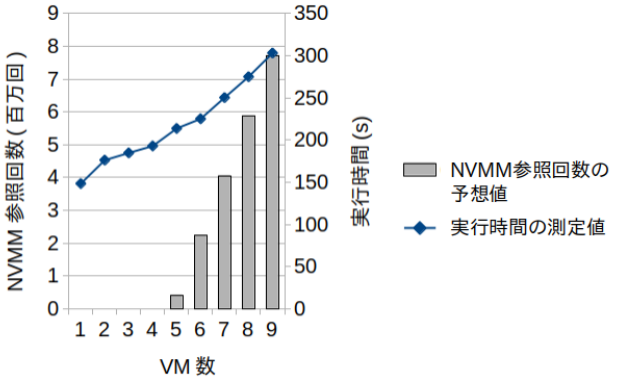

図 11 Vmul のサイズ 30k で VM 数増加時の NVMM 参照回数の予想値と DRAM+NVMM 機での実行時間の関係

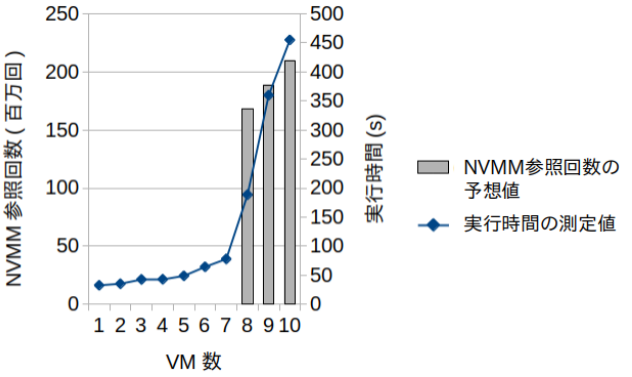

図 12 Sadd のサイズ 8GB で VM 数増加時の NVMM 参照回数の予想値と DRAM+NVMM 機での実行時間の関係

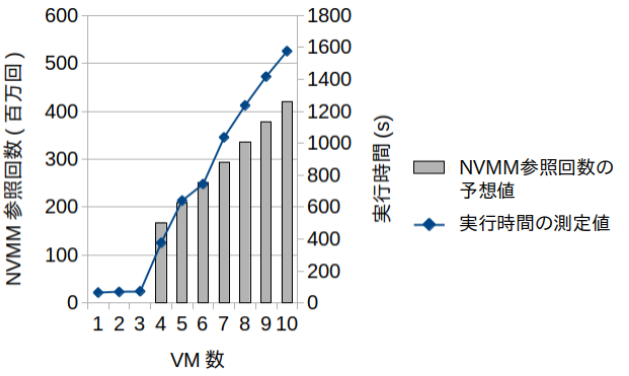

図 13 Sadd のサイズ 16GB で VM 数増加時の NVMM 参照回数の予想値と DRAM+NVMM 機での実行時間の関係

Vmul の場合は NVMM 参照回数自体が少ないこともあって変化が不明確だが，総データ量が DRAM の容量以内に収まっているときと比較して VM 数に対する実行時間の伸びが大きくなってはいる．Sadd の 2 条件については NVMM にアクセスするときにはっきりと実行時間が長くなる．これは NVMM アクセスのレイテンシがボトルネックとなっていることを示す．

従って NVMM Interference は NVMM 参照の回数を用いて式 (4) のように表現できると思われるが，実行時間の増加分は NVMM 参照回数と比例するわけではなく，あく

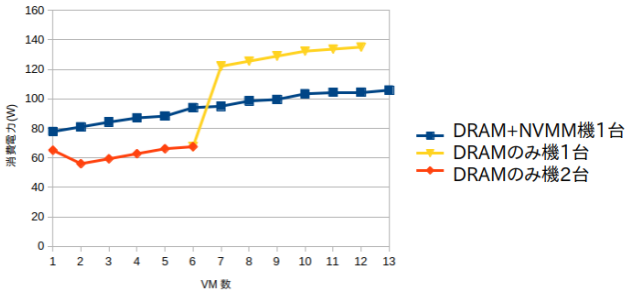

図 14 KVS でサイズ 1MB とした場合にサービスの提供に必要な電力の試算

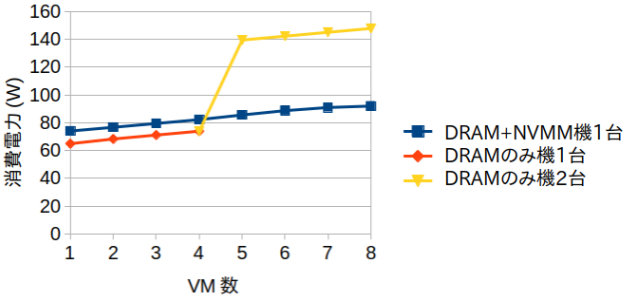

図 15 Vmul のサイズ 30k でサービスの提供に必要な電力の試算

まで相関があることに留まる．DRAM 参照回数との関係も含めて，引き続き分析する必要がある．

4.5 電力削減に向けた試算

Vmul や Sadd で多数の VM を動作させる場合，大容量の主記憶が必要となる．このとき DRAM+NVMM 機ならば PM1 台で対応可能だが，DRAM のみ機は 2 台用意しなければならないといった場面が起こりうる．今後 VM のマイグレーションによって稼働する PM を減らし省電力化を行うことに向けた試算として，このような場合の NVMM 搭載の利害得失を検討する．

まず KVS のデータサイズを 1 リクエスト当たり 1MB に変更したようなものを考える．このとき VM6 台以上の場合 DRAM のみ機ではスワップが発生してサービスの提供を継続することが困難になるため，VM6~10 台が稼働する条件では DRAM+NVMM 機ならば PM1 台，DRAM のみ機ならば PM2 台が必要となる．これに基づいて消費電力を計算すると**図 14** のようになる．DRAM+NVMM 機と DRAM 機における電力の実測値と，DRAM 機を 2PM 用いて該当する VM 数を実現したと仮定した場合の予想電力値を示している．VM 数が少ないとき，DRAM+NVMM 機は NVMM の消費電力のため不利になるが，DRAM 機が 2 台必要となる場合にはその合計よりも省電力である．

ここでサイズ 1MB の KVS においては，ボトルネックはリクエストサイズ 100kB の場合と同様にネットワークバンド幅となる．複数の PM でネットワークを共用するときは，その中のいくつの PM を用いてリクエストを処理しても，結局ネットワークが律速となって実行時間は同程度を要すると考えられる．よって，サービスの提供に必要となる電力が小さければすなわち同じ処理をするために必要なエネルギーも小さい．

次に Vmul のサイズ 30k について考える．VM5 台以上のとき DRAM のみ機ではスワップが発生してサービスの提供を継続することが困難になるため，VM5~8 台が稼働する条件では DRAM+NVMM 機ならば PM1 台，DRAM のみ機ならば PM2 台が必要となる．これに基づいて消費電力を計算すると**図 15** のようになる．

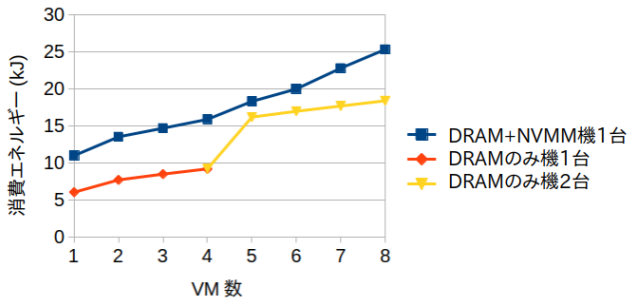

図 16 Vmul のサイズ 30k でサービスの提供に必要なエネルギーの試算

ところが，Vmul の VM5 台以上を DRAM+NVMM 機で動作させたとき，ボトルネックは NVMM アクセスのレイテンシとなっている．すなわち DRAM のみ機の PM2 台を利用した場合に比べて，実行時間は長くなる．電力の計算結果と実行時間から，サービスの提供に必要なエネルギーを計算した結果を**図 16** に示す．VM を DRAM+NVMM 機へとマイグレーションし集約するにあたっては，ボトルネックに留意しなければかえって消費エネルギーを増加する結果を招く可能性があることが分かる．

5. おわりに

DRAM に加えて NVMM も搭載した PM と，DRAM のみを搭載した PM で，VM を稼働させた．それぞれのボトルネックを分析し，利用可能な資源量に対する要求量の比である Interference の構成を検討した．

また，DRAM に加えて NVMM を搭載した PM に VM をマイグレーションし集約した場合の消費電力とエネルギーについて簡単に試算した．ネットワークバンド幅が律速になるようなアプリケーションの場合は，電力とエネルギーのいずれも削減できる可能性が示された．

今後は様々な VM の組が与えられたとき，Interference の重み付き和を下げるように VM 配置を変えていく．これによって VM の性能を保ちつつ，NVMM の追加で主記憶を大容量化した PM へと集約する．そして空いた PM の電源を切ることで，データセンタ全体での消費電力減少を目指す．

謝辞 本研究は基盤研究 (B)17H01708 の助成を受けた

ものである．

参考文献

[1] BroadGroup: Data Centers ‘Going Green’ To Reduce A Carbon Footprint Larger Than The Airline Industry，，入手先 ⟨https://data-economy.com/data-centers-going-green-to-reduce-a-carbon-footprint-larger-than-the-airline-industry/⟩（参照 2019.11.11）.

[2] Qureshi, M. K., Srinivasan, V. and Rivers, J. A.: Scalable High Performance Main Memory System Using Phase-change Memory Technology, *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ISCA ’09, New York, NY, USA, ACM, pp. 24–33 (online), DOI: 10.1145/1555754.1555760 (2009).

[3] Grupp, L. M., Caulfield, A. M., Coburn, J., Swanson, S., Yaakobi, E., Siegel, P. H. and Wolf, J. K.: Characterizing Flash Memory: Anomalies, Observations, and Applications, *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 42, New York, NY, USA, ACM, pp. 24–33 (online), DOI: 10.1145/1669112.1669118 (2009).

[4] Lee, B. C., Ipek, E., Mutlu, O. and Burger, D.: Architecting Phase Change Memory As a Scalable Dram Alternative, *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ISCA ’09, New York, NY, USA, ACM, pp. 2–13 (online), DOI: 10.1145/1555754.1555758 (2009).

[5] Jia, G., Han, G., Jiang, J. and Liu, L.: Dynamic Adaptive Replacement Policy in Shared Last-Level Cache of DRAM/PCM Hybrid Memory for Big Data Storage, *IEEE Transactions on Industrial Informatics*, Vol. 13, No. 4, pp. 1951–1960 (online), DOI: 10.1109/TII.2016.2645941 (2017).

[6] Hassan, A., Vandierendonck, H. and Nikolopoulos, D. S.: Software-managed Energy-efficient Hybrid DRAM/NVM Main Memory, *Proceedings of the 12th ACM International Conference on Computing Frontiers*, CF ’15, New York, NY, USA, ACM, pp. 23:1–23:8 (online), DOI: 10.1145/2742854.2742886 (2015).

[7] Vogelsang, T.: Understanding the Energy Consumption of Dynamic Random Access Memories, *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 363–374 (online), DOI: 10.1109/MICRO.2010.42 (2010).

[8] Huang, Y., Pavlovic, M., Marathe, V., Seltzer, M., Harris, T. and Byan, S.: Closing the Performance Gap Between Volatile and Persistent Key-Value Stores Using Cross-Referencing Logs, *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, Boston, MA, USENIX Association, pp. 967–979 (online), available from ⟨https://www.usenix.org/conference/atc18/presentation/huang⟩ (2018).

[9] Xu, F., Liu, F., Liu, L., Jin, H., Li, B. and Li, B.: iAware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud, *IEEE Transactions on Computers*, Vol. 63, No. 12, pp. 3012–3025 (online), DOI: 10.1109/TC.2013.185 (2014).

[10] 山田浩史，河合英宏，　大島訓：Web サービスのリクエストに着目したサーバ計算機の消費電力特性，情報処理学会研究報告. [システムソフトウェアとオペレーティング・システム]，Vol. 2014, No. 1, pp. 1–7（オンライン），入手先 ⟨https://ci.nii.ac.jp/naid/110009766990/⟩ (2014).

[11] Lim, K., Meisner, D., Saidi, A., Ranganathan, P. and Wenisch, T.: Thin servers with smart pipes: Designing SoC accelerators for memcached, Vol. 41, pp. 36–47 (online), DOI: 10.1145/2508148.2485926 (2013).

[12] Kim, S.-g., Eom, H. and Yeom, H. Y.: Virtual machine consolidation based on interference modeling, *The Journal of Supercomputing*, Vol. 66, No. 3, pp. 1489–1506 (online), DOI: 10.1007/s11227-013-0939-2 (2013).

[13] Chen, L., Shen, H. and Platt, S.: Cache contention aware Virtual Machine placement and migration in cloud datacenters, *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pp. 1–10 (online), DOI: 10.1109/ICNP.2016.7784447 (2016).