Vol.2016-HPC-155 No.22
2016/8/9

メモリホットプラグを用いた メインメモリの省電力化に関する初期検討

石原 雅也[1,†1,a)]　三輪 忍[1,†1]　八巻 隼人[1,†1]　本多 弘樹[1,†1]

概要：近年のスーパーコンピュータではハードウェアの待機電力を削減することが求められている．我々は，スーパーコンピュータにおける DIMM の待機電力を削減する方法として，メモリホットプラグを利用した省電力化手法を提案している．この手法では，ジョブの実行中に必要な DIMM を残して，使用していない DIMM の電源を OFF にすることで待機電力を削減する．DIMM の電源を OFF にした場合には，メモリ容量が減少するだけでなくメモリチャネル数の減少によってメモリバンド幅が減少する可能性があるが，これまでの我々の研究ではこの問題を考慮してこなかった．この問題に対し，現在，チャネルに複数枚の DIMM が存在し，かつ使用メモリ量が各チャネルの DIMM 一枚で賄える場合にのみ，二枚目以降の DIMM の電源を OFF にする制御方針を検討している．この方針では，各チャネルには常に一枚以上 DIMM が接続されている状態となり，メモリバンド幅の減少は生じない．この方針が有効であるためには，ジョブ実行中に十分な待機状態の DIMM があることが条件となる．そこで本論文では，九州大学情報基盤センターで運用中のスーパーコンピュータ CX400 のジョブログの解析を行い，上記の方針の妥当性の検証を行った．

1. はじめに

近年のスーパーコンピュータが消費する電力は膨大であり [1]，ランニングコスト削減のため消費電力を削減することが重要となっている．特に，代表的なスーパーコンピュータ「京」の消費電力は約 12MW[2] と大きく，これ以上の電力消費を避けるべく，低消費電力なスーパーコンピュータが求められており，そのための新しい低消費電力化技術の創出が必要である [3], [4], [5], [6], [7], [8], [9], [10]．

スーパーコンピュータには長時間使用されないハードウェアデバイスが存在し，それらの待機電力を削減することができれば大きな省電力化につながると期待されている [11]．例えば，CPU では DVFS やパワーゲーティングが既に実用化されており，これらの技術が CPU の待機電力の削減に大きく貢献している．一方，メモリの待機電力の削減は十分に行われているとは言い難い [12], [13], [14]．市販されているメモリの多くが，待機電力削減のために低電力モードを使用するが，我々の評価によると，8GB DDR3-1600 SDRAM は待機時に 0.9W を消費する．これは，メモリの低電力モードは DRAM チップの待機電力を削減する一方，PLL 等の回路の待機電力を削減できないことが原因と考えられる．計 1 万ノードからなるシステムにおいて，上記の DIMM が各ノードに 8 枚存在すると仮定すると，このシステムのメモリの待機電力は 72kW に達する．

そこで我々は，スーパーコンピュータにおけるメモリの待機電力を削減する方法として，メモリホットプラグを利用する方法を提案している [11]．メモリホットプラグは Linux カーネル 3.9 以降からサポートされている技術であり，OS が稼働中のコンピュータシステムに接続されている DIMM の交換を可能にする技術である．この技術を用いて，ジョブの実行中でも，待機状態の DIMM の電源を OFF にすることで待機電力を削減することを提案している．従来手法では，制御用のプログラムが各ノードのメモリ使用量を定期的に監視し，スラッシングを起こさない程度の量のメモリを残して，それ以外のメモリを OFF 状態としていた．DIMM の電源を OFF にした場合には，メモリ容量が減少するだけでなく，メモリチャネル数の減少によってメモリバンド幅も減少する可能性があるが，従来研究ではこの問題は考慮されていない．

我々は，現在，メモリバンド幅減少による性能低下を解決する方法として，ジョブの実行中に DIMM 数を変更する場合はメモリチャネル数分のメモリを常に ON 状態とする

[1] 情報処理学会
IPSJ, Chiyoda, Tokyo 101–0062, Japan
[†1] 現在，電気通信大学
Presently with The University of Electro-Communications
[a)] m-ishihara@hpc.is.uec.ac.jp

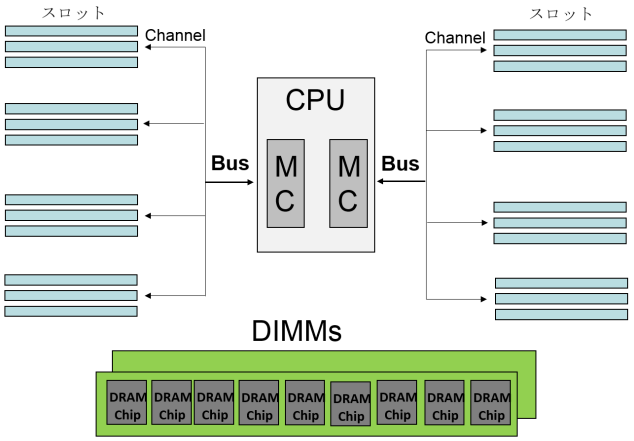

図 1　メモリシステムの構成図

制御方針を検討している．この方針では，各チャネルには常に 1 枚以上 DIMM が接続されている状態となり，メモリバンド幅の減少は生じない．この方針が有効であるためには，ジョブ実行中に十分な待機状態の DIMM があることが条件となる．そこで本論文では，スーパーコンピュータのジョブログ実トレースの解析を行い，上記の方針の妥当性を検証した．

本論文の構成は以下の様になっている．まず 2 章でメモリホットプラグとその技術を用いた省電力法について述べ，続く 3 章では現在検討中の DIMM の電源制御方針を述べる．4 章でスーパーコンピュータのジョブ実行ログの解析，提案手法の妥当性の検証を行う．最後に，5 章で本論文をまとめる．

2. 研究背景

2.1 メモリシステムの構成

スーパーコンピュータのノード内のメモリシステムは，大きく分けて DIMM，チャネル，スロットからなる [15], [16]．簡単な構成図を**図 1** に表した [12], [13], [17], [18], [19], [20]．DIMM はメモリチップを基盤にまとめたモジュールのひとつである．チャネルは DIMM からデータを転送する際に用いられる転送経路であり，各スロットに繋がっている（最大 3 スロット）．チャネルを通して転送されたデータは，バスを通り CPU のメモリコントローラ（MC）に運ばれる．

2.2 メモリホットプラグ

Linux カーネル 3.9 以降からサポートされているメモリホットプラグは，システムの運用中に DIMM を抜き差しすることができる技術である [21], [22]．メモリホットプラグは，操作の過程を物理フェーズと論理フェーズに分けることができる．物理フェーズでは，実際に DIMM の抜き差しを行い，論理フェーズでは，DIMM の抜き差しを行う

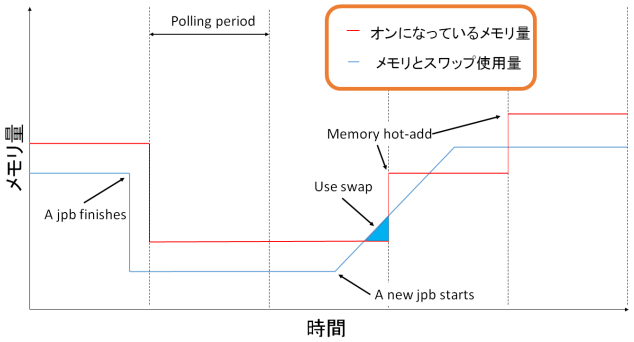

図 2　オンデマンド制御によるメモリホットアド

前に OS によってメモリを使用できる/できない状態に変更する．そして，OS を止めずにメモリ（DIMM）の削除，追加を行うことをホットリムーブ，ホットアドと呼ぶ．一般的にメモリホットプラグはシステムの可用性向上に使われるが，本研究ではこの技術を，使用していないメモリの電源を OFF にするために使用する．本目的でメモリホットプラグを使用するためには，ハードウェア側のサポートが必要だが，そのために必要なハードウェアの改良は難しくないと考えられる．

メモリホットプラグを用いてあるシステムの ON 状態の DIMM を減らした場合，システム上で稼働するアプリケーションには以下の影響がある．

ひとつめに，実行中のアプリケーションが，使用できるメモリ量以上を要求した場合に起こるスラッシングが増加することが考えられる．メモリ量を減らしすぎた場合，減らす前のメモリ量で実行したときには起きないはずのスラッシングが生じてしまうことになり，システムの処理性能の低下を引き起こす．

ふたつめに，あるチャネルに接続された全ての DIMM の電源を OFF にした場合，使用されるチャネルの数が減ることになり，データの転送速度が遅くなる．そのため，メモリ容量は十分だが，データの転送速度が遅くなったために，ジョブの処理速度が低下する可能性が考えられる．

このように，スラッシングの発生とチャネル数の減少はアプリケーションの実行の妨げになり，実行速度の著しい低下に繋がる可能性がある．そのため，メモリ量を変更する際はこの 2 点についての考慮が必要である．

2.3 先行研究

我々は，メモリ量を減らしすぎた場合にスラッシングが増加することを考慮して，メモリの使用量に応じて ON 状態のメモリ量を制御するオンデマンドメモリホットアド [11] を提案している (**図 2**)．この方法では，定期的に計算ノードのメモリとスワップの使用状況を監視し，デーモンプログラムによってメモリの ON/OFF を切り替える．計算ノードが待機状態になると，ノード上のメモリ使用量が減少する．そこで，デーモンプログラムは監視している

メモリ使用量に応じて ON 状態のメモリ量を減少させる．多くのジョブがノード上で実行を開始し，現在使用可能なメモリサイズを超えた場合は，スワップ領域を使用することになる．この状況をデーモンプログラムが検知した場合，スワッピングを回避するために，デーモンプログラムは ON 状態のメモリ量を増加させる．

この手法では，上記の手法によってアプリケーション性能がどの程度低下するのかを評価している．計算ノードとスワップの使用状況を監視する期間（ポーリング期間）を 3 パターン（0.1s, 0.01s, 0.001s）用意し，HPL，starDGEMM，PTRANS，RA（MPI RandomAccess），FFT（MPI FFT）の 5 つのベンチマークについて評価を行っている．この実験で測定に用いたプログラムは，HPCC ベンチマークから選んだ．この実験では，各アプリケーションが実行に使用するメモリ量を全体の約 20 %程度にまで削減でき，これによる実行速度の低下は最大で約 3 %に抑制できた，という結果を示している．このことから，メモリホットプラグは省電力化に十分期待できると述べられている．一方でこの実験は，メモリ量を減らしてもチャネル数が減らない環境でしか行っておらず，DIMM 単位で ON/OFF をした場合を考慮した方法の検討が必要となる．

3. DIMM の電源制御方針の検討

3.1 予備実験

DIMM 単位で ON/OFF をした場合について，特定のチャネルに接続された全ての DIMM の電源が OFF になり，使用されるチャネルの数が減ることでデータの転送速度が遅くなることが考えられる．そのため，チャネル数の変化による実行速度の低下度合を確認するため性能評価を行った．また，実行速度が低下した場合，その要因はデータの転送速度のみでありメモリ容量の不足ではないことを確認するため，各ベンチマークの実行に必要となったメモリ量の測定を予備実験として行った．特に，本評価では，DIMM が 4 枚と 8 枚，16 枚の場合について調査した．**表 1** のノード構成を用いて，HPL、starDGEMM、PTRANS、RA（MPI RandomAccess）、FFT（MPI FFT）の 5 つのプログラムを使用し評価を行った。前項の実験と同じく，この実験で測定に用いたプログラムは，HPCC ベンチマークから選んだ．

表 1 実験に用いたノード構成

Name	Remarks
CPU	Xeon E5-2650 v2 (2.6GHz) x2 16 physical and 32 logical cores
Memory	4GB DDR3-1333 SDMRAM x2〜16
OS	Ubuntu 14.04 with Linux kernel 3.13
チャネル数	4/CPU, 計 8

評価結果を**図 3** に示した．同数のチャネルを使用する

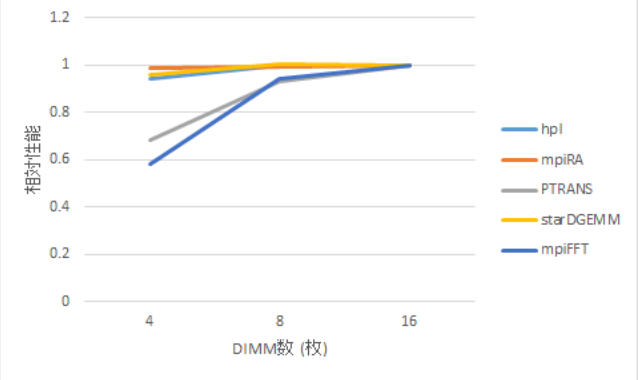

図 3 DIMM の枚数を変えた場合の相対性能評価

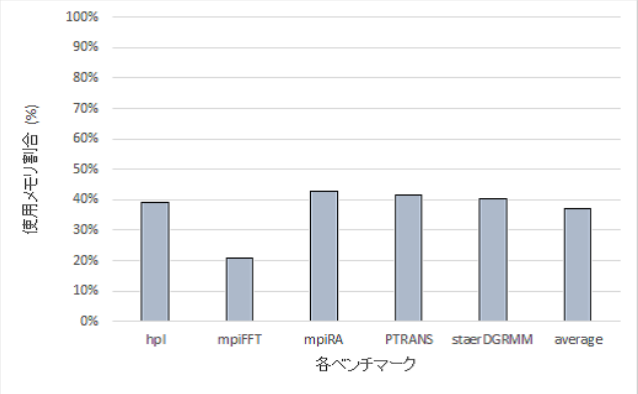

図 4 DIMM4 枚時の使用メモリ割合

DIMM が 16 枚と 8 枚の場合では，性能低下はほとんど生じない．この結果から，チャネルに複数枚の DIMM が接続されておりメモリに十分な空き容量がある場合に関して，DIMM の電源を OFF にしても性能に影響はないということがわかった．一方，使用チャネル数が異なる DIMM が 8 枚と 4 枚の場合では，著しい性能低下が見られた．性能が低下した要因としては，メモリ量の不足によるものと，チャネル数の減少によるものが考えられる．

また，DIMM が 4 枚の場合における各ベンチマークの実行に必要となったメモリ使用量の測定結果は**図 4** のようになった．この結果から，使用メモリ量は搭載メモリ量の半分以下であり，性能低下の要因から除外できることがわかった．

以上のことから，プログラムの実行中に DIMM 数を変更する場合は，使用チャネル数を考慮する必要があることがわかった．

3.2 検討中の制御方針

予備実験によって，チャネル数（メモリバンド幅）の減少によってアプリケーション性能は大幅に低下することが確認できた．この問題に対し，我々は現在，チャネルに複数枚の DIMM が存在し，かつ使用メモリ量が各チャネルの DIMM 一枚で賄える場合において，二枚目以降の DIMM の電源を OFF にする制御方針を検討している．この方針

表 2　CX400 の性能

Name	Remarks
演算ノード	理論演算性能 345.6GFLOPS 主記憶容量 128GB メモリバンド幅 102.4GB/s
総ノード数	1476 ノード
総プロセッサ (コア) 数	2952 プロセッサ (23616 コア)
主記憶容量の総和	約 184.5TiB

では，各チャネルでは常に一枚以上の DIMM が ON 状態となっており，二枚目以降の DIMM の電源を制御している最中もメモリバンド幅はほぼ一定となる．

この方針が有効であるためには，ジョブ実行中に十分な待機状態の DIMM があることが条件となる．そこで，スーパーコンピュータのジョブログから使用メモリ量の割合を求め，この方針の有効性を調査した．

4. ジョブログ解析

提案手法の有用性について検証するため，運用中のスーパーコンピュータにおけるジョブの実行ログを用いて，メモリの待機時間と使用量の解析を行った．解析には，九州大学情報基盤センターより提供していただいた，CX400 のジョブログを使用した．

4.1 CX400 システム

九州大学の高性能演算サーバシステム，スーパーコンピュータ CX400 は全 1,476 ノードからなっており，ノード一つあたりの性能は，理論演算性能が 345.6GFLOPS，主記憶容量が 128GB，メモリバンド幅が 102.4GB/s であり，8 チャネルのシステムとなっている (**表 2**)．また，総プロセッサ数は 2,952 プロセッサ (2,3616 コア)，主記憶の総和量は約 184.5TiB である．なお，今回の解析にあたり，用いられている DIMM の容量を 8GB と仮定した．

今回使用したログは，システム稼働開始（2012 年 10 月）直後の比較的空いている時期（2013 年 1 月〜6 月）のものであり，現在のシステム全体の CPU 稼働率は CX400 で 55〜85 %程度である．また，九州大学では，一般的な共有利用ノード群の他，特定の研究グループにノードを割り当てる占有利用ノード群を設定しており，それらのノードについては，割り当てた研究グループの活動状況によってはアイドル時間が極端に長くなる場合がある．そのため，これらを考慮した調査を行った。

4.2 全ノードに対する解析結果

4.2.1 待機時間

提案手法は，ジョブ実行中のメモリ待機電力を削減することを目的としたが，ノードが稼働していない状態のメモリの待機電力削減にも効果がある．そのため，スーパーコンピュータの稼働中に，待機時間がどれほどの割合存在するのか，ノードごとの非稼働率を調査した．ここで，待機時間とは，各ノードでジョブが終了してから次のジョブが開始するまでの時間のことである．

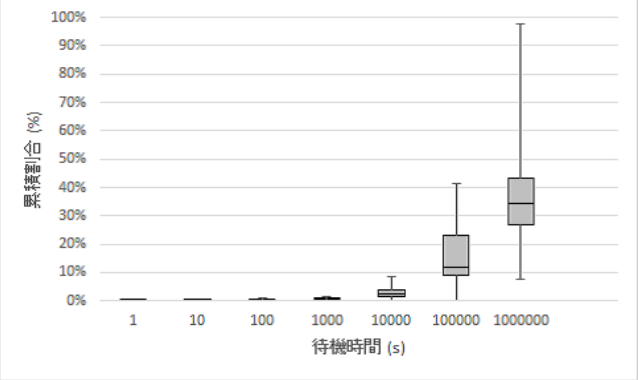

図 5　全ノードにおける待機時間の累積割合

全ノードにおける待機時間の累積割合の結果を**図 5** に示した。このグラフは，ノードごとの待機時間の分布を表している．ノード毎に，1 秒未満，10 秒未満，100 秒未満，1,000 秒未満，10,000 秒未満，100,000 秒未満，1,000,000 秒未満の待機時間が，観測時間全体の何割を占めているかを求めた上で，待機時間を x 軸，待機時間が観測時間に占める割合を y 軸に取ってグラフ化している．したがって，このグラフにおける最も右端の y の値，つまり最大時間未満の待機時間が観測時間に占める割合が，該当ノードの非稼働率（= 1-稼働率）を表している．

今回解析した CX400 には計 1476 ノード存在するため，各時間におけるノードごとの分布が分かるように箱ひげグラフを用いた．x の各時間における値ごとに，全ノードの中で y (累積時間割合) が最小となる値（ひげの下端），第 1 四分位（箱の下端），中央値（箱の中央の線），第 3 四分位（箱の上端），最大となる値（ひげの上端）を 1 枚のグラフにプロットした．特に，今回求めるノードの非稼働率（or 稼働率）の分布については，このグラフの右端の箱ひげに注目すればよい．右端の箱ひげは，最小値，中央値，最大値が，それぞれおよそ 0.08, 0.35, 0.98 となっている．これは最も稼働していたノードの稼働率が 92 %で，全体の半数のノードは 65 %以上稼働しており，最も稼働していないノードの稼働率は 2 %であったことを意味している．また，右端の箱の第 3 四分位は 0.43 なので，全体の 3/4 のノードは 57 %以上稼働していたことになる．

これらの結果から，全体の半分のノードについて，非稼働率がおよそ 30〜40 %となっており，この待機時間における待機電力を削減することで省電力化に大いに貢献できると考えられる．

4.2.2 メモリ使用量

提案した方法で待機状態の DIMM を減らすためには，

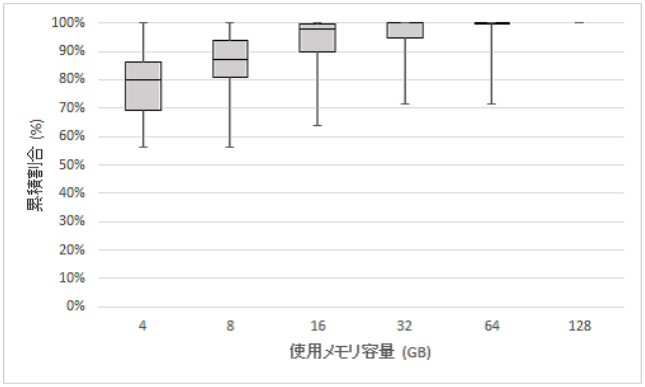

図 6 全ノードにおける使用メモリ量の累積時間分布

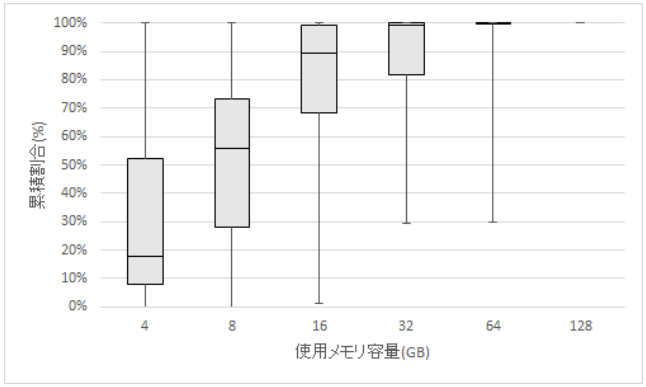

図 7 ジョブ実行中の全ノードにおける使用メモリ量の累積時間分布

ジョブによって使用されるメモリ量が，ノードの各チャネルに一枚ずつ DIMM を接続したときのメモリ容量以下である必要がある．そのため，この条件を元に，各チャネルの 2 枚目以降の DIMM を OFF にする機会がどれだけあるかを調査した．

全ノードについて，使用メモリ量と，その容量で動作していた時間の累積割合を表したグラフが**図 6** になる．なお，ジョブが実行されていない間の使用メモリ量を 2GB と仮定した．このグラフは，ノードごとのメモリ使用量の分布を表している．ノード毎に，2GB 未満，4GB 未満，8GB 未満，16GB 未満，32GB 未満，64GB 未満，128GB 未満の容量をメモリが使用していた時間が，観測時間全体の何割を占めているかを求めた上で，使用したメモリ容量を x 軸，メモリ使用時間が観測時間に占める割合を y 軸に取ってグラフ化している．

各ノードについての分布が分かるよう今回も箱ひげグラフを用いて表した．各チャネルの 2 枚目以降の DIMM を OFF にする機会とは，つまり必要となるメモリ容量が 64GB 未満である場合に注目すればよく，図 6 では x 軸が 64GB の箱ひげがこれを表す．この箱ひげは，第 3 四分位が 1 となっており，全体の 3/4 のノードでは全ての時間帯で 2 枚目の DIMM を OFF にすることが出来ることを意味している．一方で，最小値は 0.72 であり，最低でも全体の 72 %の時間帯で 2 枚目の DIMM を OFF に出来ることがわかった．

しかし，図 6 の結果は，ジョブが実行されていない時の待機時間についても，計測に含まれている．本項の趣旨は，ジョブ実行中に DIMM が削減可能な時間の割合の調査であるため，待機時間を計測から引かなければならない．その結果が**図 7** になる。

図 6 と図 7 を比較すると，使用メモリ量が 64GB 以下のときの累積割合に変化が見られる．前者のグラフでは 4GB 以下の使用時間が多いが，後者のグラフでは 64GB 以下までで同程度の使用時間であることがわかる．また，図 7 の 64GB の累積割合に注目すると，64GB を超えてメモリが必要になったノードは全体の 1/4 にとどまり，3/4 のノードは 64GB 以下のメモリ量でジョブを実行することが出来たことがわかる．この結果から，全体の 75 %のノードについて，ジョブ実行中に提案手法が適用可能であることがわかった．

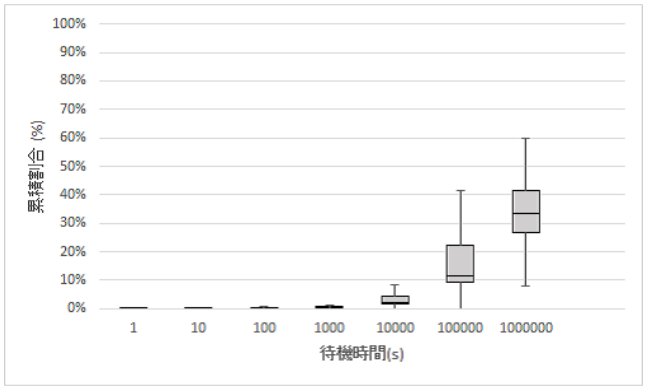

図 8 共有ノードにおける待機時間の累積割合

4.3 共有ノードに対する解析結果

本論文では共有ノードを次のように仮定して解析を行った．占有ノードを特定することは困難であるため，前述した”占有ノードの長期間放置による極端な待機時間の発生”を考慮し，5.2 項で得られた結果から，1000,000 秒の待機時間累積割合が一定以上（60 %）のノードを除外（およそ 100 ノード分）し，改めて 5.2 項と同様の解析を行った．(**図 8**、**図 9**)。

待機時間の累積割合を示した図 8 に関して，図 6 との違いとしては長期間放置による極端な待機時間があったノードを除外したため，右端の 1,000,000 秒の累積割合の最大値が今回定めた 60 %に揃っている点が挙げられる．一方で 1 秒～100,000 秒については値に変化はなく，共有ノードの長期間放置による極端な待機時間に左右されないことがわかった．使用メモリ量の累積時間分布を表した図 9 については，図 7 と比較しても特に違いは見られない．これは，本評価はジョブ実行中のメモリ使用量の解析であり，極端な待機時間が生じたノードが本データに与える影響は小さいためだと考えられる．

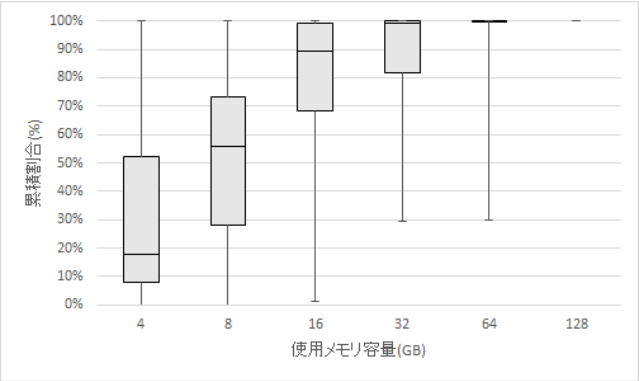

図 9 ジョブ実行中の共有ノードにおける使用メモリ量の累積時間分布

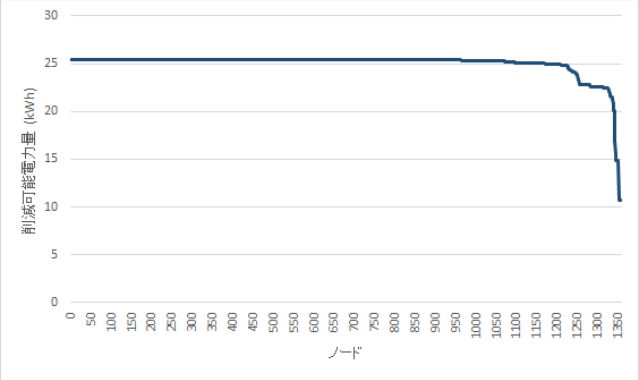

図 10 ノード別の削減可能電力量

この結果から，今回解析したジョブログについては，共有ノードの長期間放置が本手法に及ぼす影響は小さいと判断できる．

4.4 省電力効果の試算結果

我々は，5.3 項の解析結果をもとに，今回検討している方法を用いることで削減できる電力量の試算を行った．8GB の DIMM について，待機時の消費電力を測定したところ，1 枚あたり約 0.9W であった．各ノードで、ジョブ実行中に使用していない（削減可能な）メモリ容量を理想的に削減できると仮定し，メモリの使用量が 64GB 未満の場合に DIMM を削減可能として試算を行った．

試算の結果を**図 10** に示した．今回解析した期間（147 日間）全体を考えると，1 ノードあたり，およそ 25kWh，全ノード合わせておよそ 34MWh の電力量が削減可能であることがわかる．一日あたりを考えると，一ノードあたりおよそ 170Wh，全ノード合わせておよそ 230kWh の電力量の削減が見込める試算となる．これは，理化学研究所で運用されているスーパーコンピュータ「shoubu」の四時間分の電力量に匹敵する [23]．

この結果から，本提案手法を用いることで，一定の電力量の削減が見込めることがわかった．

5. おわりに

本論文では，メモリホットプラグを用いた省電力法における DIMM の削減方法の指針の紹介，並びにスーパーコンピュータのジョブログを用いた全ノードの待機時間の累積割合，ジョブ実行中の全ノードにおける使用メモリ量の累積時間の解析，省電力効果の試算を行った．ジョブ実行中，チャネルに接続されている二枚目の DIMM を使用する機会は極めて少ないため，本方法を用いる機会は多く，一定の電力量の削減が見込めることがわかった．そのため今後は，チャネルに接続されている二枚目の DIMM の電源を OFF にする具体的な手法の提案を行う予定である．

謝辞 本研究のために CX400 のジョブログを提供していただいた九州大学情報基盤センターに感謝致します．本研究の一部は JST CREST による．

参考文献

[1] U. S. Department of Energy.: Final Minutes Advanced Scientific Computing Advisory Committee, Aug（2012）.
[2] 宮崎博行，草野義博，新庄直樹：スーパーコンピュータ「京」の概要 (特集 スーパーコンピュータ「京」), Fujitsu (2012).
[3] M. Okuda.: Searching out Trends in the Supercomputer Environment and Technology Development Following the “K computer”. BioSupercomputing Newsletter Vol.7,（2012）.
[4] S. Miwa and H. Nakamura.: Profile-based power shifting in interconnection networks with on/off links. In Proceedings of the 2015 ACM/IEEE Conference on Supercomputing, pages ,（2015）.
[5] T. Patki, D. K. Lowenthal, B. Rountree, M. Schulz, and B. R. de Supinski.: Exploring hardware overprovisioning in power-constrained, high performance computing. In Proceedings of the 27th International Conference on Supercomputing, pages 173182,（2013）.
[6] Micron. Calculating Memory System Power for DDR3, July（2007）.
[7] L. A. Barroso and U. H ¨ olzle.: The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines. Synthesis Lectures on Computer Architecture, Jan.（2009）.
[8] L. A. Barroso and U. H ¨ olzle.: The Case for Energy-Proportional Computing. IEEE Computer, 40(12):3337, December（2007）.
[9] K. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt, and T. F. Wenisch.: Disaggregated Memory for Expansion and Sharing in Blade Servers. ISCA ’09: International Symposium on Computer Architecture,（2009）.
[10] Lluc Alvarez, Llus Vilanova, Miquel Moreto, Marc Casas, Marc Gonzlez, Xavier Martorell, Nacho Navarro, Eduard Ayguad, Mateo Valero.: Coherence Protocol for Transparent Management of Scratchpad Memories in Shared Memory Manycore Architectures. ISCA,（2015）.
[11] S. Miwa, and H. Honda.: Memory Hotplugfor EnergySavings of HPC systems, SC’15, poster, (2015).
[12] Qingyuan Deng, David Meisner, Luiz Ramos, Thomas F. Wenisch, Ricardo Bianchini.: MemScale: Active Low-Power Modes for Main Memory, ACM SIGARCH Computer Architecture News, Vol39, No.1, pp.225-238, (2011).

[13] Qingyuan Deng, David Meisner, Luiz Ramos, Thomas F. Wenisch, Ricardo Bianchini.: Active Low-Power Modes for Main Memory with MemScale. Issue No.03, pp: 60-69, (2012)
[14] Qingyuan Deng, David Meisner, Luiz Ramos, Thomas F. Wenisch, Ricardo Bianchini.: MemScale: Memory System DVFS with Multiple Memory Controllers. In ISLPED, (2012).
[15] David A.Patterson, john L.Hennessy, 成田光彰 訳：コンピュータの構成と設計第3版 (上)(下), 日経BP社, (2006).
[16] Bruce Jacob.: The Memory System: You Can't Avoid It, You Can't Ignore It, You Can't Fake It. Morgan Claypool, (2009).
[17] Prashant Nair, Chia-Chen Chou, Moinuddin Qureshi.: A case for Refresh Pausing in DRAM memory systems. High Performance Computer Architecture, IEEE 19th International Symposium on. 23-27, (2013).
[18] Youngsok Kim, Jaewon Lee, Jae-Eon Jo, and Jangwoo Kim.: GPUdmm: A high-performance and memory-oblivious GPU architecture using dynamic memory management. High Performance Computer Architecture, 2014 IEEE 20th International Symposium on. 546-557. Feb. (2014).
[19] Junwhan Ahn, Sungjoo Yoo, Onur Mutluy, Kiyoung Choi.: PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture. ISCA, (2015).
[20] Subhasis Das, Tor M. Aamodt, William J. Dally.: SLIP: ReducingWire Energy in the Memory Hierarchy. ISCA, (2015).
[21] Y. Ishimatsu.: Memory hotplug. LinuxCon Japan, (2013).
[22] T. Chen.: Introduction to ACPI based memory hot-plug. LinuxCon Japan, (2013).
[23] Green500.: The Green500 List - November 2015.