Vol.2018-HPC-165 No.10
2018/7/30

プリウェイクアップ手法による ON/OFFリンクの消費エネルギー削減

松山 朋樹[1,a)]　三輪 忍[1]　八巻 隼人[1]　本多 弘樹[1]

概要：近年，インターコネクション・ネットワークの省電力技術として ON/OFF リンクが注目されている．ON/OFF リンクは，データ通信を行っていないリンクを低電力モードに変更することで，ネットワークの消費電力を削減する技術である．ON/OFF リンクは多くのアプリケーションにおいて高い省電力効果を示すことが知られている．その一方で，通信間隔が短いアプリケーションに対しては，アプリケーションの実行性能が大幅に低下する，あるいは，省電力効果があまりない. 本研究では，ON/OFF リンクを搭載した HPC システムにおいて，アプリケーションの通信要求に先立ってリンクをプリウェイクアップすることにより，通信間隔が短いアプリケーションに対して実行性能を維持しつつリンクの消費電力を削減する手法を新たに提案する．このプリウェイクアップに必要な手続きは，コンパイラがアプリケーションを解析し，ソースコード内の適切な位置に自動的に埋め込むことを想定している．今回，予備実験として，単純な MPI プログラムを用いて，プリウェイクアップ用コードの挿入位置がアプリケーション性能とリンクエネルギーに与える影響を評価した．本稿ではその結果を述べる．

1. はじめに

エクサスケール級のスーパーコンピュータを開発するためには，システムの電力効率の改善が必須である．現在世界最速のスーパーコンピュータである Summit は，8.8MW の電力を消費して 12.23PFLOPS を達成している [11]．一方，米国 DoE (Department of Energy) の報告によると [12], 20〜30MW の消費電力でエクサスケール級を実現することが次世代のスーパーコンピュータの開発目標となっており，上記開発目標と現在のスーパーコンピュータとの間には依然として大きなギャップ（電力性能比に直すと 24 倍）が存在する．上記開発目標を達成するためにはスーパーコンピュータの電力効率の改善が必須であり，システム内のさまざまなハードウェアにおいて消費電力を削減する必要がある．

スーパーコンピュータの消費電力においてインターコネクション・ネットワークの消費電力は大きな割合を占めており，将来的にはシステム全体の消費電力の約 30%を占めると言われている [3], [4]．インターコネクション・ネットワークが大きな電力を消費する背景には，近年のスーパーコンピュータのネットワークは高いバンド幅と冗長性の両方が要求されることが挙げられる．システム規模の増大にともなって上記の傾向は強くなると考えられることから，ネットワークの省電力技術は今後ますます重要になると予想される．

ネットワークの省電力技術として，高性能計算分野において，近年，ON/OFF リンクが注目を集めている．ON/OFF リンクは，データ通信を行っていない時にリンクを低電力モードにすることで，消費電力を削減する技術である．低電力モード中のリンクは，両端に位置する多くの回路に対して電力供給が停止することにより，最大で通常時の 10% 程度の電力にまで消費電力を削減できる [6], [9], [10]．並列アプリケーションの多くは常にデータ通信を行っているわけではないため，システムの待機中だけでなくアプリケーションの実行中においても，リンクの低電力モードを使用する機会は多数存在する．リンクの消費電力はネットワーク全体の消費電力の約 70%を占めることから [6], [9], [10], HPC システムの消費電力を削減する上で ON/OFF リンクの果たす役割は大きいと言える．

ON/OFF リンクは, 通信間隔が長いアプリケーションに対して高い省電力効果を示すことが知られている [6], [9], [10]. その一方で，後述するように，通信間隔が短いアプリケーションに対しては，従来の ON/OFF リンクの制御法は，アプリケーション性能が大幅に低下するか，あるいは，省電力効果がほとんどない．ON/OFF リンクは，Ethernet において IEEE802.az として既に標準化されている技術である

[1] 電気通信大学
The University of Electro-communications
[a)] matsuyama@hps.is.uec.ac.jp

が [8]，HPC システムのネットワークではあまり採用されておらず，上記の問題点が HPC 分野における ON/OFF リンクの普及を妨げていると考えられる．

我々は，通信間隔が短いアプリケーションに対して，アプリケーション性能を維持しつつ省電力効果を最大化する，新たな ON/OFF リンクの制御法を提案する．

ON/OFF リンクにおいて低電力モード時にデータが到着した場合，通常モードに復帰してからデータ転送を行うため，復，データ通信の開始が復帰処理に必要な時間分（約 4μs）遅れてしまう．この問題に対し，提案手法では，アプリケーションの通信要求に先立って通信に使用されるリンクを通常モードに復帰（プリウェイクアップ）することにより，上記の遅延の隠ぺいを図る．リンクをタイミングよくプリウェイクアップできれば，アプリケーション・データの到着時には通常モードへの復帰が完了しており，リンクは直ちにデータ通信を開始できる．

従来の ON/OFF リンクの低電力モード制御はハードウェアのみによって行われていたが，マイクロ秒オーダーの通信イベントの発生をハードウェアで予測するのは困難と予想されることから，提案手法ではソフトウェアにより ON/OFF リンクの低電力モードを制御する．

本稿では，提案手法の予備実験として，単純な MPI プログラムを用いて，プリウェイクアップ用コードの挿入位置がアプリケーション性能とリンクエネルギーに及ぼす影響を評価した結果を述べる．プリウェイクアップ用コードは，将来的には，カスタム・コンパイラによってアプリケーションを解析し，ソース・コード内の適切な位置に自動的に埋め込むことを考えている．

以下に本論文の構成を述べる．まず，第 2 章では，ON/OFF リンクの詳細を述べる．次に，第 3 章では提案手法を説明する．第 4 章では評価方法と評価結果を示す．最後に第 5 章で，まとめと今後の展望について述べる．

2. ON/OFF リンク

本章では，まずは ON/OFF リンクの基本的な動作について述べる．前述のように，ON/OFF リンクを使用する際は，低電力モードから通常モードに復帰する際に発生する通信遅延が問題となる．この問題をハードウェアの改良によって緩和する手法が既に提案されていることから，2.2 節ではこの手法について述べる．

2.1 概要

ネットワーク機器においてはリンクが多くの電力を消費する．より具体的には，リンクの両端に位置する PHY が多くの電力を消費している [13]．PHY は，通常，リンクの接続状態を確認するために，特殊な信号（IDLE コード）を定期的に送受信している [6]．そのため，PHY には常に電源が投入されており，その結果，通常のリンクは，データ通信を行っていない状態でも一定の電力を消費している（**図 1**(a)）[5]．

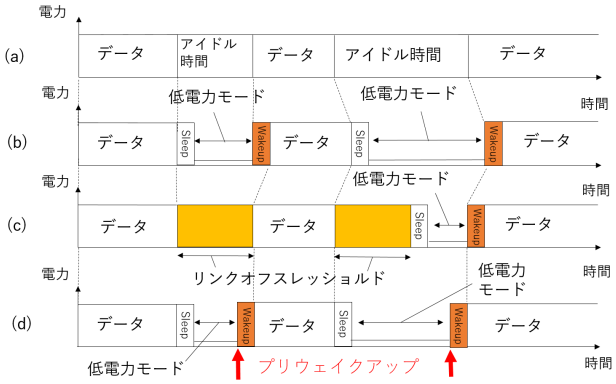

図 1 (a) 通常のデータ通信 (b)ON/OFF リンクでのデータ通信 (c) リンクオフスレッショルドを有する ON/OFF リンクでのデータ通信 (d) プリウェイクアップ手法を用いた際のデータ通信

これに対し，ON/OFF リンクは，データ通信を行っていない時に IDLE コードの送信を停止し，PHY を低電力モードに変更する．低電力モードのリンクはデータ通信を行うことはできないが，PHY 内の多くの回路の電源を遮断することによって消費電力を最大 1/10 にまで削減する [7]．

ON/OFF リンクのモード遷移はハードウェアで制御されており，通常は以下のように動作する．(1) データ通信が終了すると直ちに通常モードから低電力モードに遷移（スリープ）する．(2) 低電力モードのリンクにデータが到着すると，直ちに通常モードに復帰（ウェイクアップ）する（図 1(b)）．これらのモード遷移には数 μs の時間を要する [8], [9]．

上述の動作から，ON/OFF リンクには，低電力モード中に到着したデータの通信が遅くなる問題点がある．低電力モードのリンクにデータが到着した時は，まずはリンクをウェイクアップしなければならず，通常のリンクと比べてデータ通信の開始がウェイクアップ時間（約 4μs）分遅延する（図 1(b)）．上記の通信遅延はネットワーク・インテンシブなアプリケーションに対して大幅な性能低下（最悪 2 倍以上の実行時間の増加）を引き起こすことから [9]，HPC システムにおいて ON/OFF リンクを使用する際は，上記の通信遅延がアプリケーション性能に与える影響を緩和する必要がある．

2.2 リンクオフスレッショルドの利用

前述した通信遅延の問題を緩和するため，リンクオフスレッショルドを有する ON/OFF リンクが提案されている [9]．リンクオフスレッショルドを有する ON/OFF リンクでは，データ通信の終了後直ちに低電力モードに遷移するのではなく，一定時間（リンクオフスレッショルド）経過後に低電力モードへと遷移する．リンクオフスレッショルドを有する ON/OFF リンクでは，リンクオフスレッショ

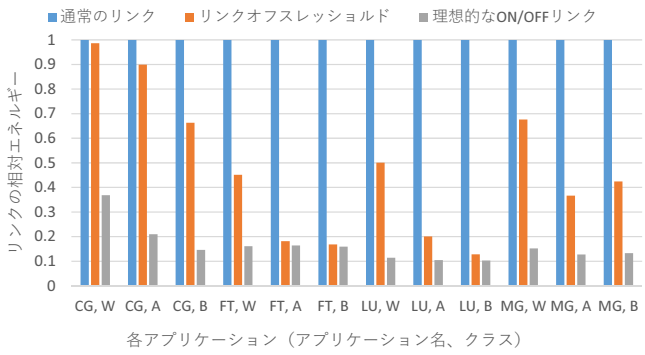

図 2　各リンクの消費エネルギー

ルドの期間内は通常モードを維持するため，この期間に次のデータがリンクに到着すると，リンクのモードを変更することなく直ちにデータ通信を開始できる（図 1(c)）．すなわち，データの到着間隔が短い（リンクオフスレッショルド内の）場合は，リンクオフスレッショルドを有する ON/OFF リンクは通常のリンクとまったく同じタイミングで通信できる．

一方，データの到着間隔が長い（リンクオフスレッショルドを超える）場合は，リンクオフスレッショルドを有する ON/OFF リンクは低電力モードへと遷移することでリンクの消費電力を削減する．その結果，前節で述べたように，次のデータ通信開始時にはウェイクアップ時間分の遅延が発生する．

HPC アプリケーション内の処理は一般に通信フェーズと計算フェーズに分かれることから [10]，リンクオフスレッショルドを適切に設定することで，アプリケーション性能を維持しつつリンクの消費電力を削減できる．すなわち，通信フェーズでは通常モードを維持することによって通信遅延の発生を抑制し，計算フェーズでは低電力モードを使用することによってリンクの消費電力を削減する．先行研究では，リンクオフスレッショルドを 50μs に設定した時にリンクの消費エネルギーは最小（通常のリンクの 30%）となり，この時のアプリケーション性能の低下率は 2%であったと報告されている [9]．

リンクオフスレッショルドを有する ON/OFF リンクは，リンクオフスレッショルド内は通常モードを維持するため，この期間内はデータ通信を行っていない場合でもリンクの消費電力を削減できない．その結果，リンクオフスレッショルドを有する ON/OFF リンクは，十分な長さの計算フェーズを有するアプリケーションに対しては高い省電力効果を示すが，リンクオフスレッショルド内で通信を繰り返すようなネットワーク・インテンシブなアプリケーションに対しては省電力効果がない．

図 2 に，さまざまなアプリケーションにおける，通常のリンク，リンクオフスレッショルドを有する ON/OFF リンク，理想的な ON/OFF リンクの消費エネルギーを示す [14]．ここで理想的な ON/OFF リンクとは，データ通信の終了後直ちにスリープを開始し，データ到着時にウェイクアップが完了するリンクである．図より，リンクオフスレッショルドを有する ON/OFF リンクによる消費エネルギー削減量は，アプリケーション (CG,W) によっては 1.4%程度に過ぎず，理想的な消費エネルギー削減量とは 63.1%の開きがある．このようなアプリケーションに対しては，リンクオフスレッショルドを有する ON/OFF リンクは有効とは言えず，さらなる省電力化のためには新たな ON/OFF リンクの制御法が必要である．

3. 提案手法

3.1 概要

前章で述べたように，リンクオフスレッショルドを有する ON/OFF リンクは，リンクオフスレッショルド期間内のリンク電力を削減できないため，通信間隔が短いアプリケーションに対しては省電力効果があまりない．そこで我々は，リンクオフスレッショルドに代わる ON/OFF リンクのウェイクアップ遅延の隠ぺい手法として，プリウェイクアップ手法を提案する．提案手法では，アプリケーションによる通信要求を予想し，上記通信の開始に先立ってプリウェイクアップ用のデータ送信を行うことにより，（使用予定の）低電力モードの ON/OFF リンクのウェイクアップ処理を開始する．

図 1(d) にリンクのプリウェイクアップ手法を用いた際のデータ通信を示す．横軸は時間を示しており，プリウェイクアップを開始する時刻を矢印で示している．リンクにデータが到着するまでにプリウェイクアップが完了すれば，プリウェイクアップ手法は通常のリンク（図 1(a)）と同じタイミングでデータ通信を開始できる．これにより ON/OFF リンクのウェイクアップ遅延を隠ぺいする．

プリウェイクアップ手法によってウェイクアップ遅延を隠ぺいできれば，リンクオフスレッショルドは必要ない．すなわち，データ通信を終えたリンクを直ちにスリープさせることができる．その結果，リンクオフスレッショルドを有する ON/OFF リンク（図 1(c)）では削減できなかった，データ通信終了後の一定時間分の電力を削減できる．

3.2 プリウェイクアップの実現方法

前章で述べたように，ON/OFF リンクのモード遷移はハードウェアによって制御されているが，我々はリンクのプリウェイクアップをソフトウェアにより実現する．これは，リンクをプリウェイクアップするためには数マイクロ秒先の通信イベントを予測する必要があるが，そのような予測をハードウェアが行うのは困難と予想されるためである．後述するように，アプリケーションのソース・コードを解析することによって，通信イベントの大まかな（マイクロ秒レベルの）発生タイミングはある程度予測できると考えられる．そこで提案手法では，ソース・コード内の適

```
1   if(myrank==0)
2   {
3     for(k=0; k<num_proc; k++)
4     {
5       /* ダミーデータ送信 */
6       MPI_Isend(&dummy,1,MPI_INT,
7       k,1234,MPI_COMM_WORLD,&request);
8     }
9     else
10    {
11      MPI_Irecv(&dummy,1,MPI_INT,
12      0,1234,MPI_COMM_WORLD,&request);
13    }
14  }
15  if(myrank==0)
16    MPI_Waitall(num_proc-1, request, status);
17  else
18    MPI_Wait(&request[0], &status[0]);
```

図 3 ダミーデータを送信コード

切な位置にプリウェイクアップ用のコードを挿入することによって，リンクのプリウェイクアップを実現する．

図 3 にプリウェイクアップ用コードの例を示す．この例は，MPI_COMM_WORLD 内のランク 0 をルートとするブロードキャスト通信に対するプリウェイクアップ処理を表している．ランク 0 をルートとするブロードキャスト通信においては，ランク 0 から全ランクに対してデータ送信が行われる．そのため，プリウェイクアップ用のコードでは，ランク 0 から全ランクに対してメッセージ・サイズが 1 のダミー・データ（中身は何でもよい）を非同期で送信する．また，ランク 0 以外のランクは，上記のダミー・データを非同期に受信する．

この例は MPI_COMM_WORLD 内のランク 0 をルートとするブロードキャスト通信に対するプリウェイクアップ用コードであるが，他の集団通信の通信パターン，および，1 対 1 通信に対しても，同様にプリウェイクアップ用コードを定義できる．このようなコードをソース・コード内の適切な位置に挿入することでリンクのプリウェイクアップを行う．

ソース・コードの解析とプリウェイクアップ用コードの挿入は，将来的にはカスタム・コンパイラによって自動化する予定である．LLVM のパス上に上記の機能を実装することを想定しており，したがって，コードの解析と挿入は IR (Intermediate Representation) レベルで行う予定である．コードの解析には，静的解析だけなく，後述するように，動的解析も組み合わせる必要があると考えている．

3.3 自動化に向けての課題

前節で述べたアプローチによってリンクのプリウェイクアップを実現する場合，まず，(1) アプリケーションに含まれる各通信関数とその通信パターン（具体的には通信相手のランクと通信方向）を特定し，その上で，(2) 各通信の開始までにリンクのウェイクアップが完了するようなソース・コード上の位置にプリウェイクアップ用コードを挿入する必要がある．

(1) は比較的容易に実現できる（ソース・コードを静的に解析することで特定できる）一方，(2) は挑戦的な課題である．プリウェイクアップは本来の通信の開始までに完了する，すなわち，プリウェイクアップ用コードは通信関数のウェイクアップ時間（例えば 4μs）前に実行されるのが理想である（図 1(c)）．したがって，プリウェイクアップ用コードを挿入するためには，上記のようなソース・コード上の位置を何らかの方法によって割り出す必要がある．

我々は，プリウェイクアップ用コードの最適な挿入位置を，当該通信関数までに実行される浮動小数点演算数と，アプリケーションを実行する CPU の FLOPS 値から求めることが可能と考えている．例えば，1TFLOPS の CPU でアプリケーションを実行する場合，上記 CPU は 4μs の間に 4M 回の浮動小数点演算を実行できる．したがって，ソース・コード上で，当該通信関数の 4M FLOP 前にプリウェイクアップ用コードを挿入すれば，上記 CPU でこのアプリケーションを実行した場合，プリウェイクアップ用コードが通信関数の 4μs 前に実行されると期待できる．

上記の浮動小数点演算数は動的な演算数であり，動的な浮動小数点演算数はループの回転数や分岐方向などの実行に依存する情報によって変化する．そのため，静的解析だけではプリウェイクアップ用コードの最適な挿入位置を求めることは困難と予想される．この問題を解決するため，小規模実行で得たアプリケーションのプロファイルなどの動的な情報も解析に利用することを考えている．

4. 評価

MPI 通信を行う単純なプログラムを用いて，プリウェイクアップ用コードの挿入位置がプログラムの性能とリンクの消費エネルギーに及ぼす影響を評価した．本章ではその結果について述べる．

4.1 評価の方法

評価に使用したプログラムを **図 4** に示す．このプログラムは, 積和演算（22 行目）を N 回行った後に `MPI_Bcast()` 関数（28 行目）によってランク 0 から他のランクにデータを一斉に送信する処理を 1,000 回繰り返す．N は `MPI_Bcast()` の実行間隔を調整するためのパラメータである．前述のように，提案手法は通信間隔が短いアプリケーションにおけるリンク消費エネルギー削減を目的としていることから，`MPI_Bcast()` の実行間隔がウェイクアップ時間より大きくリンクオフスレッショルド（文献 [8] で最適と結論づけら

れている 50μs）未満となるように N の値を設定した（具体的には N=10,000）．N は，プリウェイクアップ用コードが挿入されていない通常のリンクを実行した際の通信のトレースファイルを解析した結果得られた，実行間隔が 20〜30μs になるような値である．提案手法ではこのコードに対してプリウェイクアップ用コードを挿入する．プリウェイクアップ用コードは，前述のように将来的にはコンパイラで自動的に挿入する予定であるが，今回の実験では手動で挿入位置を変更して実験を行った．具体的には，4 における 7 行目から 18 行目がプリウェイクアップ用コードであり，このコードでは 5 行目の P の値を変更することによって，プリウェイクアップ用コードと 28 行目の `MPI_Bcast()` の実行間隔（積和演算回数）を調整できる．この P の値を 0 から 10,000 の範囲で 100 単位で変更する．また，ダミー・データの送受信を保証するため，`MPI_Bcast()` の直前には `MPI_Wait()` および `MPI_Waitall()` を挿入する．このプリウェイクアップ用コードが挿入されたプログラム（**ON/OFFwPW**）と，プリウェイクアップ用コードが挿入されていないプログラムを通常のリンクを用いて実行した場合（**NORMAL**），同プログラムをリンクオフスレッショルド無し/有りの ON/OFF リンクを用いて実行した場合（それぞれ **ON/OFFwoLT**，**ON/OFFwLT**）と比較する．

評価には ON/OFF リンクを実装した SimGrid-3.11 を用いた [6]．SimGrid は並列計算機のシミュレータであり，専用コンパイラを用いて MPI プログラムの並列環境における挙動を直接シミュレーションできることが特徴の 1 つである [2]．今回シミュレーション対象としたシステムは，4 台の計算ノードが 1 台のスイッチに接続された単純な構成である．各ノードとスイッチとの間には，ノードからスイッチに向かう UP リンクと，スイッチからノードに向かう DOWN リンクの 2 本のリンクが存在する．各ノードの計算性能，コア数，ネットワーク性能は，それぞれ，4.8GFLOPS，1 コア，5Gbps とした（**表 1**）．ON/OFF リンクのパラメータは，文献 [6] を参考に，ウェイクアップ時間を 4μs，スリープ時間を 3μs とした．

NORMAL, ON/OFFwoLT, ON/OFFwLT, ON/OFFwPW の各実行性能とリンクの消費エネルギーは，SimGrid が出力する通信トレースから算出した．リンク 1 本の電力は，文献 [6] を参考に，通常リンクおよび通常モード時がともに 12 W，低電力モード時は 1.2 W とした．ウェイクアップ，および，スリープの最中は，通常モードと同じ電力を消費する．次節では，各ケースに対して 5 回のシミュレーションを行うことで求めた，平均実行性能と平均リンク消費エネルギーを示す．

今回の実験で使用したダミー・データは 32 ビット，かつ，サイズ 1 のメッセージであり，典型的な HPC システムのネットワークであれば，ナノ秒から数十ナノ秒のオーダーで通信が完了する．一方，SimGrid-3.11 によるシミュレーションの時間解像度はマイクロ秒オーダーであり，ナノ秒オーダーのネットワークの振る舞いを観測できない．

表 1 マシン構成

計算性能 (GFLOPS)	4.8
コア数	1
ネットワーク性能	5Gbps，latency: 100ns
ノード数	4

```
1  for(i=0; i<1000; i++)
2  {
3    for(j=0; j<N; j++)
4    {
5      if(j==P)
6      {
7        if(myrank==0)
8        {
9          for(k=1; k<num¥_proc; k++)
10         {
11           /* ダミーデータ送信 */
12           MPI_Isend(&dummy,1,MPI_INT,
13           k,1234,MPI_COMM_WORLD,&request);
14         }
15       }
16       else
17       {
18         MPI_Irecv(&dummy,1,MPI_INT,
19         0,1234,MPI_COMM_WORLD,&request);
20       }
21     }
22     y = a * x  + b;
23   }
24   if (myrank ==0)
25     MPI_Waitall(num_proc-1, request, status);
26   else
27     MPI_Wait(&request[0], &status[0]);
28   MPI_Bcast(&buffer, 300, MPI_INT, 0, MPI_COMM_WORLD);
29 }
```

図 4 実験に使用したプログラムのコード

この問題に対し，本稿では，SimGrid によるシミュレーションは 1/X の計算性能とネットワーク性能を有するマシンを対象に行い，得られた通信トレースに記録された時刻を 1/X 倍することによって，求めるマシンの通信トレースを得る，という方法を取った．シミュレーション対象のマシンの計算性能とネットワーク性能をともに 1/10,000 倍にすれば，シミュレータ内で発生するさまざまなイベントの時間が 10,000 倍となり，本来はナノ秒オーダーで発生する通信イベントもマイクロ秒オーダーのイベントとして観測できる．

このシミュレーション方法の妥当性を検証するため，Nas Parallel Benchmark[1] の LU と FT（クラスは W）を用いて，さまざまな X について，シミュレーション対象マシンの計算性能とネットワーク性能をともにを 1/X 倍した時の実行時間を調査した．ノード数は 4 と 8 の場合について実験を行った．実験の結果，いずれのプログラム，ノード数においても，計算性能とネットワーク性能をともに $1/10^1$, $1/10^2$, $1/10^3$, $1/10^4$, $1/10^5$ 倍にした時の実行時間が，それぞれ，10^1, 10^2, 10^3, 10^4, 10^5 倍となることが確認できたことから，上記のシミュレーション方法によって取得した通信

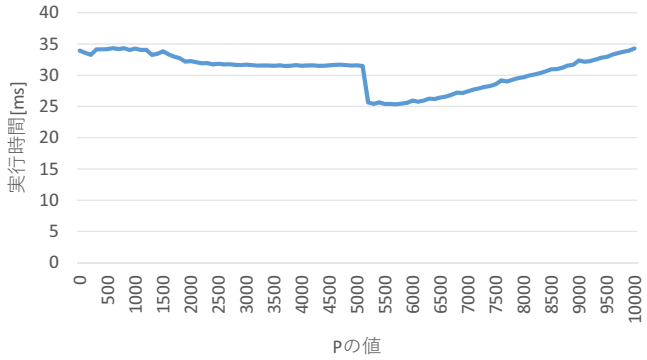

図 5　P の値と実行時間の関係

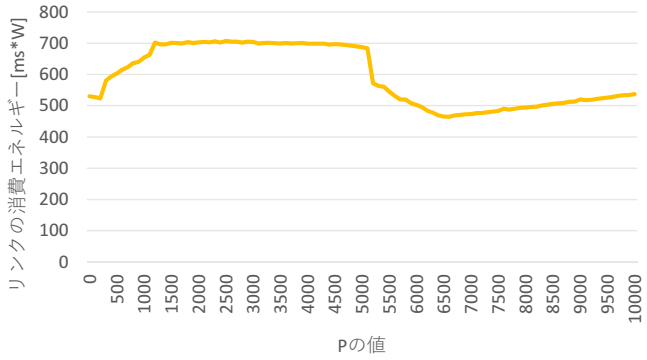

図 6　P の値とリンクの消費エネルギーの関係

トレースを本実験では使用した．

4.2　評価結果

P の値を変更した場合の実行時間を**図 5** に示す．また，各 P の値における，リンクの消費エネルギーを**図 6** に示す．2 つのグラフの横軸はともに P の値である．図 5 の縦軸は実行時間，図 6 の縦軸はリンクの消費エネルギーである．

図 5 より，P の値が小さい時はプログラムの実行時間は大きな値を示す．これは，プリウェイクアップ用コードの実行時期が早すぎて，プリウェイクアップ後に再びリンクがスリープしてしまい，`MPI_Bcast()` の実行時には再度リンクのウェイクアップが行われているからである．図より，P がある値（5,200）を超えると，プログラムの実行時間は急激に減少する．これは，プリウェイクアップ用コードの実行タイミングが `MPI_Bcast()` の実行タイミングに近づき，プリウェイクアップされたリンクがスリープすることなく直ちに `MPI_Bcast()` によって使用されるからである．P=5,200 から P=5,900 まではプログラムの実行時間はほとんど同じであるが，P が 6,000 を超えるとプログラムの実行時間は再び増加に転じる．これは，プリウェイクアップ用コードの実行が遅くなるにつれて，プリウェイクアップの完了が `MPI_Bcast()` の開始に間に合わなくなり，ウェイクアップ遅延を隠ぺいできなくなるためである．

一方，リンクの消費エネルギーは，図 5 より，基本的には，プログラムの実行時間が長くなるほど増加する傾向を示す．ただし，P が 1,100 以下の領域では，プログラムの実行時間は長いにも関わらず，P の値が小さくなるほど消費エネルギーが減少する．これは，P の値が小さくなるほど，`MPI_Bcast()` と最外ループの次イタレーションのプリウェイクアップ用コードの実行間隔が短くなることによって，プリウェイクアップ用コードの実行時にウェイクアップ処理が不要になる確率が高くなるからである．これは，我々の ON/OFF リンクの実装では，スリープ処理の最中（先行するデータ通信が終了してから 3μs の間）に到着したデータは，直ちに通信が開始できることを仮定しているためである．

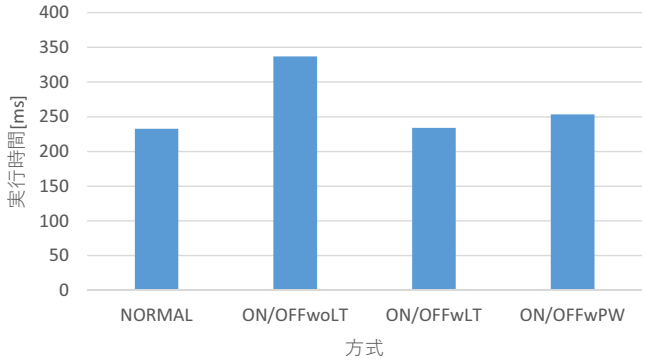

図 7　各方式おける実行時間の関係

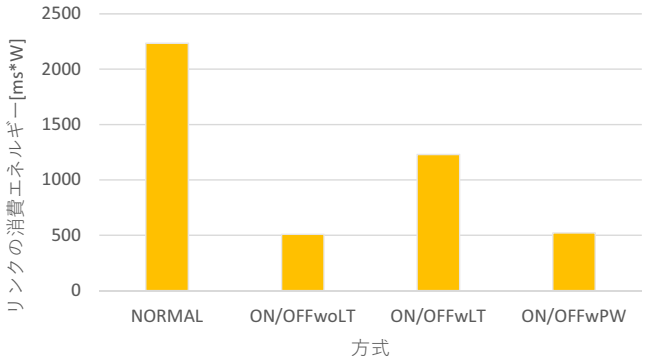

図 8　各方式におけるリンクの消費エネルギーの関係

本稿では，P の最適値を，プログラムの性能低下率が 10%の範囲内でリンクの消費エネルギーが最小となる値と定義する．図 5 と図 6 より，P の最適値は 5,700 であった．

図 7 に，NORMAL, ON/OFFwoLT, ON/OFFwLT, ON/OFFwPW の各方式の実行時間を示す．ON/OFFwPW は P の最適値とした．図の縦軸は相対実行時間（NORMAL で正規化），横軸は各方式である．NORMAL に対する ON/OFFwoLT の実行時間は 1.45 倍だったのに対し，ON/OFFwPW の実行時間は 1.09 倍となった．この結果より，提案手法を用いることで，ON/OFF リンクのウェイクアップ遅延を隠ぺいできることが確認できた．一方，NORMAL に対する ON/OFFwLT の実行時間は 1.01 倍であることから，ON/OFFwLT に比べると ON/OFFwPW の性能低下はやや大きい（約 8%）．これは，プリウェイクアップ用コードの実行オーバヘッドが影響していると考えられる．さらなる

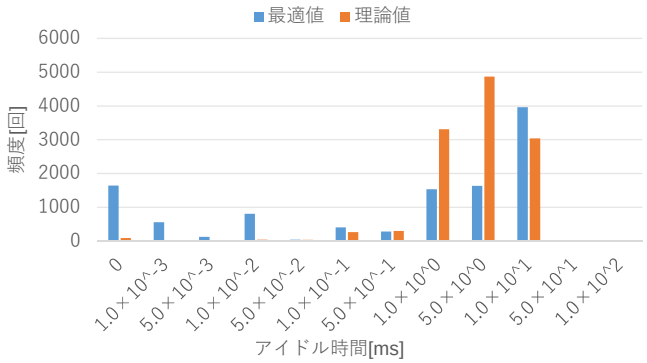

図 9 アイドル時間と頻度の関係

性能向上のためには，プリウェイクアップ用コードの実行オーバヘッドを削減する必要がある．

各方式におけるリンクの消費エネルギーを**図 8** に示す．図の縦軸はリンクの相対消費エネルギーを表しており，NORMAL の消費エネルギーの値によって正規化してある．図より，ON/OFFwPW は ON/OFFwLT よりも高い消費エネルギー削減効果を示す．ON/OFFwLT に対する ON/OFFwPW のエネルギー削減率は 57.7%であった．この結果より，通信間隔が短いプログラムに対しては，プリウェイクアップ手法を用いることで，既存技術よりも大幅な省電力化が達成できることがわかった．

3.3 節で述べたように，我々は，プリウェイクアップ用コードの挿入位置を浮動集数点演算数と CPU 性能から算出することを検討している．今，ウェイクアップ時間が 4μs，CPU 性能が 4.8GFLOPS なので，4μs の間に実行可能な浮動小数点演算の回数は 19,200 回である．4 のプログラムは，内側ループ（N のループ）1 回につき 2 回の浮動小数点演算が行われることから，P=9,600 の時に，プリウェイクアップ用コードと `MPI_Bcast()` の実行間隔が約 4μ になると予想された．以下では，この P の値を P の理論値と呼ぶ．

前述のように，実験的に得られた P の最適値は 5,700 であり，P の理論値とはやや開きがある．この原因を調査するため，P=5,700, 9.600 それぞれの通信トレースを解析し，リンクのアイドル時間の分布を求めた．解析結果を**図 9** に示す．図の横軸はリンクのアイドル時間，縦軸は出現頻度である．

この結果より，P の最適値と比べて，P の理論値は，4μs を超えるアイドル時間を要する通信が多いことが分かる．これは，ダミー・データの送信完了から `MPI_Bcast()` の通信開始までの時間がウェイクアップ時間よりも長くなってしまっていることを示唆している．プリウェイクアップ用コードの挿入位置を正しく見積もることができなかった原因としては，内側ループ 1 周分には浮動小数点演算以外の処理も含まれており，この処理の時間が影響したことが考えられる．プリウェイクアップ用コードの挿入位置の予測方法は，今後改良の余地がある．

5. まとめと今後の展望

本稿では，従来の ON/OFF リンクの制御とは異なるアプローチで，低電力モードのリンクを通信要求に先立って通常モードにすることで，ON/OFF リンクのウェイクアップ遅延を隠ぺいしつつ，リンクの消費電力を削減するプリウェイクアップ手法を提案した．また，予備実験として，プログラムのソースコードの様々な位置にプリウェイクアップ用コードを挿入し，プログラムの実行時間とリンクの消費エネルギーへの影響を評価した．その結果，プリウェイクアップ手法を用いることで，ON/OFF リンクのウェイアップ遅延がプログラム性能に与える影響を最小化するとともに，従来よりもリンクの消費エネルギーを大幅に削減できた．

今後の課題として，今回はブロードキャスト通信を行う単純なプログラムを用いて評価を行ったが，実 HPC アプリケーションを含む他の通信パターンを有するプログラムを用いてプリウェイクアップ手法の評価を行うことが挙げられる．また，今回はプリウェイクアップ用コードを手動で挿入したが，この手続きを自動化する方法についても今後検討を続ける予定である．

参考文献

[1] Bailey, D., Barszcz, E., Barton, J., Browning, D., Carter, R., Dagum, L., Fatoohi, R., Fineberg, S., Frederickson, P., Lasinski, T., Schreiber, R., Simon, H., Venkatakrishnan, V. and Weeratunga, S.: THE NAS PARALLEL BENCHMARKS, *NAS Technical Report RNR-94-007*, pp. 1–79 (1994).

[2] Casanova, H., Giersch, A., Legrand, A., Quinson, M. and Suter, F.: *Journal of Parallel and Distributed Computing*, Vol. 74, No. 10, pp. 2899–2917 (2014).

[3] D. Turek: *Challlenges on the road to exascale computing(invited talk)* (2008).

[4] Kogge, P. M.: Architectual challenges at the exascale frontier (2008).

[5] Maestro, J. A. and Reviriego, P.: Energy Efficiency in Industrial Ethernet: The Case of Powerlink, *IEEE Trans. Industrial Electronics*, Vol. 57, No. 8, pp. 2896–2903 (2010).

[6] Miwa, S. and Nakamura, H.: Profile-based Power Shifting in Interconnection Networks with on/off Links, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '15, pp. 37:1–37:11 (2015).

[7] Reviriego, P., Larrabeiti, D., Maestro, J. A., hernandez, J. A., Afshar, P. and Kazovsky, L.: Energy efficiency in 10Gps Eternet transceivers: Copper versus fiber, *Proceedings of the 2010 Conference on Optical Fiber Communicatin (OFC/NFOEC), collocated National Fiber Optic Engineers Conference*, pp. 1–3 (2010).

[8] Reviriego, P., Hernández, J. A., Larrabeiti, D. and Maestro, J. A.: Performance evaluation of energy efficient ethernet, *IEEE Communications Letters*, Vol. 13, No. 9, pp. 697–699 (2009).

[9] Saravanan, K. P., Carpentop, P. M. and Ramirez, A.: Power/performance evaluation of energy efficient ethernet (EEE)

for high performance computing, *Proceedings of the 28th ACM International Conference on Supercomputing*, ICS '14, pp. 313–322 (2014).

[10] Saravanan, K. P., Carpenter, P. M. and Ramirez, A.: A Performance Perspective on Energy Efficient HPC Links, *Proceedings of the 2013 IEEE International Symposium on Performance Analysis of Systems and Software*, ISPASS '13, pp. 205–214 (2013).

[11] Top500: https://www.top500.org/lists/2018/06/.

[12] U. S. Department of Energy: *Final Minutes Advanced Scientific Computing Advisory Comittee* (2012).

[13] 三輪忍, 會田翔, 安島雄一郎, 清水俊幸, 安里彰, 中村宏：実 HPC 環境における EEE の電力/性能評価, 情報処理学会論文誌コンピューティングシステム, Vol. 7, No. 4, pp. 67–83 (2014).

[14] 松山朋樹, 三輪忍, 八巻隼人, 本多弘樹：ON/OFF リンクにおける通信開始遅延を低減するためのプリウェイクアップ手法の提案, 第 80 回全国大会講演論文集, pp. 123–124 (2018).