Vol.2017-HPC-161 No.15
2017/9/20

電力性能推定を目的としたインターコネクト・シミュレータ TraceRPの開発

小野 貴継[1,a)]　垣深 悠太[2]　三輪 忍[3]　井上 弘士[1]

概要：HPC システムにおけるインターコネクション・ネットワークの消費電力削減は重要な課題である．インターコネクション・ネットワークの消費電力を削減する手法として，パケットを処理していない期間は低電力モードへと遷移させる手法が挙げられる．本稿では低電力化手法をサポートしたインターコネクト・シミュレータである TraceRP について報告する．低電力化手法による性能低下も含めて評価可能であり，Fat-tree，Torus，Dragonfly の 3 つのトポロジをサポートする．通信性能評価用ベンチマーク・プログラムを用いて各トポロジにおける消費電力削減効果および性能低下について評価した．評価の結果，Fat-tree では最大で約 27%, Torus では約 33%，Dragonfly においては約 26%の消費電力削減効果があることが分かった．また，性能は Fat-tree において最大約 1.2%，Torus では約 4%，Dragonfly では約 0.6%低下することが明らかになった．

1. はじめに

次世代の HPC システムではさまざまなハードウェア・コンポーネントにおいて省電力化が必要とされており，省電力なインターコネクション・ネットワーク（以降，ネットワーク）の開発も重要課題の 1 つに挙げられている [22]．HPC システムは高い通信性能と冗長性の両方が要求されるため，高速リンクによって接続された次数の大きなネットワークが利用されることが多く，ネットワークは大きな電力を消費することが知られている（最大でシステム全体の電力の 33% [13]）．次世代の HPC システムは 20-30MW という厳しい電力制約の下でエクサフロップス級の性能を実現することが求められており [15]，この目標を達成するためには省電力なネットワーク技術が必要不可欠である．

ネットワークにおいてはリンクが大きな電力を消費することから（ネットワーク全体の電力の約 70% [19]），将来は，低電力モードをサポートするリンク（以降，On/Off リンク）がネットワークに採用されると期待されている [16], [19], [20], [21]．通常のリンクが通信の有無に関わらず常に一定の電力を消費するのに対し，On/Off リンクは，非通信時に低電力モード（通常時の 10-60%の電力）に遷移することによって電力を削減する．アプリケーションの実行中であっても通信を行っていない状態のリンクが HPC システムには多数存在することから，On/Off リンクを利用することでネットワークの電力を大幅に削減できる [19]．

On/Off リンクを用いたネットワークの性能と消費電力はネットワーク・トポロジやアプリケーションの通信パタンに強く影響されることから，高性能かつ省電力なネットワークを設計するためには上記の影響を正確に見積もる必要がある．大規模ネットワークの性能や消費電力を見積もる手段として，一般にはインターコネクト・シミュレータが利用されている．しかしながら，従来のインターコネクト・シミュレータは，On/Off リンクのシミュレーション機能をサポートしていないか [6], [10], [23]，サポートしていたとしても，実際の HPC システムの運用において頻出する状況，すなわち，複数ジョブを同時に実行する状況をシミュレートできなかった [4]．

そこで，著者らは，離散事象シミュレータをもとに開発され高いスケーラビリティを有する TraceR [12] をベースに，On/Off リンクのシミュレーション機能を有するインターコネクト・シミュレータを開発した．我々が開発したシミュレータは，低電力モードの利用によって生じるインターコネクト電力の時間的な変化を見積もることが可能なだけでなく，低電力モードから ACTIVE モードへ遷移する際に生じる通信遅延も含めた通信性能も見積もることが可能である．

開発したシミュレータを用いて，3 つの異なるトポロジ

[1] 九州大学大学院システム情報科学研究院
福岡市西区元岡 744
[2] 九州大学大学院システム情報科学府
[3] 電気通信大学大学院情報理工学研究科
東京都調布市調布ヶ丘 1-5-1
[a)] takatsugu.ono@cpc.ait.kyushu-u.ac.jp

（Fat-tree，Torus，Dragonfly）を対象に，HPC システムで実行されるアプリケーション・プログラムを想定した通信性能評価用ベンチマーク・プログラムを用いて，On/Off リンク利用時の消費電力削減効果および性能への影響を定量的に調査した．その結果，Fat-tree では最大で約 27%，Torus では約 33%，Dragonfly においては約 26%の消費電力削減効果があることが明らかになった．一方，On/Off リンクの利用により性能が低下し，Fat-tree では最大約 1.2%，Torus では約 4%，Dragonfly では約 0.6%の性能が低下することが分かった．また，低電力モードへと遷移するための条件や，低電力モードから ACTIVE モードへの遷移に要する時間など，複数のパラメータを変更して消費電力および性能に与える影響を調査した．

2. インターコネクト・シミュレータ

HPC システムにおけるネットワーク性能を評価するシミュレータはこれまでいくつか開発されてきた [4], [6], [10], [23]．これらはオフライン実行モデルに分類されるもので，アプリケーション・プログラム実行時の通信トレースを取得した後にネットワーク性能をシミュレーションする手法である [6]．主にレイテンシなどの性能を推定することが主な目的であり，ネットワーク規模とシミュレーション時間とのトレードオフが議論されることが多い．また，HPC システムの利用形態として複数のジョブを同時に実行することが考えられるが，これらのシミュレータでは複数ジョブの実行やこれらをどのハードウェアに割り当てるかということを考慮できない．

複数ジョブの同時実行をサポートした，パケットレベルでの並列シミュレーションが可能なシミュレータのひとつとして TraceR [2] が挙げられる．TraceR は主に BigSim [23], ROSS[5]，CODES[17] という既存のツールを利用する．BigSim エミュレーション・フレームワークはアプリケーション・プログラムの通信パタンを得るために用いられる．CHARM++ [1] の仮想化機能により物理コア数よりも多くのプロセスを実行することができる．これにより大規模なネットワークの通信トレースを生成することが可能になる．AMPI [11] を用いることで，MPI プログラムも同様に通信トレースを生成することが可能である．ROSS は汎用の大規模並列離散事象シミュレータであり，プロセッサ間で分散される論理プロセスを定義し，タイムスタンプを有するイベントをスケジュールすることが可能である．CODES は ROSS 機能を活用し，HPC のストレージやネットワークシステムを対象に開発されたシミュレータである．CODES のネットワークコンポーネントである Model-net は，ネットワーク上を流れるメッセージをシミュレーションする．Model-net はネットワークタイプやリンク帯域幅，リンクレイテンシやパケットサイズなどのパラメータを定義することが可能である．BigSim によって生成された通信トレースと，ネットワークの構成パラメータを TraceR に入力として与え性能を推定する．

TraceR はさらに機能強化が図られ，タスクマッピングやマルチジョブのシミュレーション，そして通信トレース取得回数の削減のための変更が実施された [12]．ネットワークの性能を調査する上で，エンドポイントに実行時にマップする機能は重要である．シミュレーションにおいてトポロジを考慮したマッピングを実現するためにエンドポイントの配置を決定することができるように拡張されている．BigSim による通信トレースの生成には長時間を要する．1 つの通信トレースで異なるメッセージサイズのシミュレーションに対応するように改良が施されている．そして，同時に複数のジョブをシミュレーションできるように機能が拡張されている．これらのジョブの配置を決める機能も提供されており，配置を変更した際の性能が予測可能である．

一方，ネットワークの消費電力を見積もることが可能なシミュレータは少ない．Saravanan らは Dimemas を消費電力が推定できるように拡張している [19]．しかしながら，Dimemas は複数ジョブの同時実行をサポートしていない．TraceR は複数のトポロジを対象とし，さらに複数ジョブをサポートしているが電力の測定はできない．

3. 電力シミュレータ TraceRP

3.1 ネットワークの電力削減技術

HPC システムの全体の消費電力が HPC システムに供給可能な電力の上限に近づいている．システムを構成する様々なコンポーネントを対象に電力を削減することが必要である．ネットワークの電力もまた，省電力化の要求が高まると考えられる．これまでネットワークの消費電力を削減する手法がいくつか提案されている．Adaptive Link Rate（ALR）は通信状況に応じてリンクスピードを動的に変更する技術である [8], [9]．リンクデータレートを切替えるためにミリ秒オーダの時間を要するという課題がある．

リンクの利用状況に応じて On/Off を切り替える手法が提案されている [14]．On/Off を切り替えるため制御信号を受け取るために，制御ネットワークを必要とする．Alonso らも同様に Fat-tree を対象にリンクを On/Off を切り替えることで消費電力を削減する手法を提案している [3]．トラフィック量を監視し，On/Off を切り替えるためのしきい値を動的に変更する制御が必要である．

一定期間リンクを利用しない場合，LPI（Low Power Idle）モードに切り替えるという規格がイーサネットを対象に提唱された [7]．EEE（Energy Efficient Ethernet）と呼ばれるこの手法は，通信する ACTIVE モードと，低電力な状態で待機する LPI モードの 2 つのモードを，通信の状況に応じて遷移させる．LPI モード時は通信できないことから，LPI モード時にパケットの処理が生じると，ACTIVE モードに遷移する必要がある．ACTIVE モードに遷移する間は

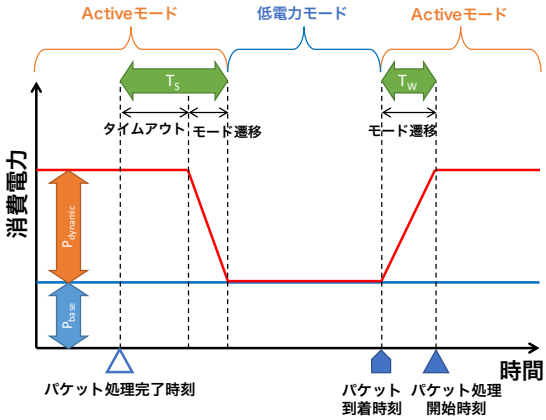


図 1: On/Off リンクの動作

パケットを処理することができず，通信レイテンシが生じる．しかしながら，この通信時間の増加はマイクロ秒オーダであり ALR よりも短く，さらに追加の制御ネットワークなどは不要である．そこで，本稿では HPC におけるネットワークにも EEE と同様の技術を適用する．

3.2 想定する低電力ネットワーク技術

本稿では，通信が生じない間各リンクに低電力モードを適用することでネットワークの消費電力を削減する手法を対象に，電力を見積もるシミュレータを開発する．**図 1** に対象とする低消費電力手法の概要を示す．パケット処理後，一定時間（T_s）パケットがない場合，当該リンクは低電力モードへと遷移する．このとき，低電力モードでは消費する電力が ACTIVE モード時より $P_{dynamic}$ だけ削減される．低電力モード中に処理するパケットが生じると，当該リンクは低電力モードから ACTIVE モードへと遷移する．ACTIVE モードへと遷移するために T_w を要する．パケット発生後 T_w が経過すると当該パケットが処理される．

本稿では双方向通信が可能なネットワークを想定する．双方向のリンク（アップ・リンク，ダウン・リンク）それぞれにおいて，On/Off リンクを適用する．つまり，アップ・リンクとダウン・リンクは独立してモード制御することが可能と仮定する．一方，1 つのリンクで送信と受信を行い，1 つのリンクを対象に On/Off リンクを適用する手法も考えられる．この場合，独立制御方式と比較して On/Off リンク制御回路は削減できる一方，送信および受信の通信が生じない期間のみに低電力を適用することになり，適用の機会が減少し電力削減効果が低くなることが想定される．したがって，本稿ではより高い消費電力削減効果を得るため，独立制御方式を対象に議論を進める．

3.3 TraceRP の実装

第 2 節で述べた TraceR を拡張し，リンクの総消費電力の推定が可能な TraceRP を実装した．具体的には，ネットワーク通信部分をシミュレーションする CODES に変更を加えた．各リンクにおいて通信処理が終了し T_s 時間経過した後，次の通信処理リクエストが発生するまでの低電力モード時間を計測する機能を追加した．また，低電力モードから ACTIVE モードへと遷移する際には T_w のレイテンシを付加するように変更した．その他の TraceR が提供する機能は TraceRP でも実行可能である．

TraceRP の実行手順も TraceR と同じである．BigSim でプログラムを実行し，通信のトレースファイルを取得する．ネットワークの構成およびアプリケーション・プログラムに関するパラメータを TraceRP+CODES に入力として与える．ネットワーク構成として，Fat-tree，Torus，Dragonfly の 3 つのトポロジをサポートする．ノード数やバンド幅，メッセージサイズなどをネットワークの構成パラメータとして与えることが可能である．また，プログラムに関するパラメータとして，ジョブのサイズや配置，ジョブ内のタスクマッピングなどを指定できる．

3.4 検証

実装した TraceRP の動作検証を実施した．消費電力の見積りに関しては，Reviriego らが提案している消費電力モデル [18] と，TraceRP によって得られた電力を比較して検証する．Reviriego らが提案している消費電力モデルを式 (1) に示す．

$$P = P_{base} + P_{dynamic} \sum_{i=1}^{N_{port}} \min(1, D \cdot \rho_i) \qquad (1)$$

ここで，P はスイッチあたりの消費電力である．P_{base} はトラッフィクがない場合のスイッチの消費電力，$P_{dynamic}$ は最大の負荷が生じている際の P_{base} からの差分であり，ポートごとに算出する．ρ_i はポート i における負荷の割合を示す．D は式 (2) で表される．

$$D = \frac{T_f}{T_{wake} + T_{sleep} + T_f} \qquad (2)$$

T_f はパケットを処理している時間である．T_{sleep} はパケット処理完了時刻から低電力モードへの遷移を開始するまでの時間であり，ACTIVE モードから低電力モードへの遷移に要する時間は含まれてない．T_{wake} は低電力モードから ACTIVE モードへと遷移するために要する時間である．

Fat-tree を例に検証の手順を説明する．まず，データを送信または受信するためのノード 2 つとそれらを接続するためのスイッチ 1 つの構成を作る．そして，ノード間でデータを送受信する検証用の簡易プログラムを実行する．本評価では ping-pong 通信を行うプログラムを作成し BigSim を用いて通信トレースを取得後，TraceRP に入力として与えて消費電力を見積もった．このとき，通信間隔を調整することで負荷の割合 ρ を調整した．消費電力を測定してモデル式 (1) と比較した．検証の結果，モデル式とほぼ同等の電力を推定していることが分かった．したがって，TraceRP

表 1: 各トポロジの構成

トポロジ	Fat-tree	Torus	Dragonfly
ノード数	288	256	342
スイッチあたりポート数	24	6	12
スイッチあたりノード数	12	1	3
ネットワーク構成	3-level	3D torus:$4 \times 8 \times 8$	グループあたり 6 スイッチ，計 19 グループ

表 2: On/Off リンクに関するパラメータ

P_{base}/N_{port}	2.08 [W]
$P_{dynamic}$	1.36 [W]
T_s	600[us]
T_w	17[us]

における電力推定機能は正しく実装され動作していることが確認された．

通信レイテンシに関しては，T_s と通信間隔を変えて送受信した際の通信時間を変化させて検証した．通信間隔が T_s よりも小さい間は低電力モードに遷移しないことから通信時間は従来手法と同様になる．一方，通信間隔が T_s よりも大きい場合，低電力モードに遷移することから通信を再開する場合は ACTIVE モードに遷移することから通信時間が長くなる．実際に TraceRP で通信間隔を変更して通信時間を見積もった結果，期待される通信時間が出力されることを確認した．

すべてのトポロジ（Fat-tree，Torus，Dragonfly）に対して消費電力および通信レイテンシの検証を実施した．その結果，意図した機能が正しく実装されていることを確認した．

4. 評価

4.1 実験環境

本評価では，想定するネットワーク構成においてプログラムを 8,192 プロセスで実行する場合を想定する．**表 1** に各トポロジの構成を示す．ノードあたりのエンドポイントは 32 である．したがって，Fat-tree は合計で 9,216 プロセス，Torus は合計 8,192 プロセス，Dragonfly は 10,944 プロセス実行可能である．

実験に用いた On/Off リンクに関するパラメータを**表 2** に示す．これらは市販されている 10GBASE-T の EEE 対応スイッチに基づき決定した [24]．なお，T_s および T_w の間の消費電力は ACTIVE モード時の電力と同等と仮定した．

様々な通信パタンを対象に評価するため，5 つのベンチマーク・プログラムを用いた．これらは，HPC システムで実行されるアプリケーション・プログラムを想定した，通信性能評価のためのプログラムである．

- `near-neighbor`：非構造型メッシュ通信プログラム
- `permutation`：行列積と行列転置プログラム
- `qbox`：第一原理分子動力学における MPI 集団通信を模擬したプログラム
- `stencil4d`：ステンシル計算プログラム
- `a2a`：All-to-All 通信プログラム

これらのプログラムから通信トレースを取得して TraceRP の入力として用いた．

4.2 消費電力と性能

消費電力の結果を**図 2** に示す．On/Off リンクを適用していない従来手法の消費電力で正規化した値を左の縦軸に，同様に実行時間を正規化した値を右の縦軸に示す．すべてのトポロジにおいて，消費電力が削減されていることが分かる．Fat-tree では `qbox` の消費電力削減率が最も高く約 27%削減されており，実行時間の増加もほぼみられない．一方，`a2a` の消費電力削減率は約 16%に留まり，最も低い削減率である．他のプログラムと比較して通信頻度が高く，低電力モードへと遷移する機会が少ないことが考えられる．なお，実行時間が最も長い，つまり性能の低下が最も大きいプログラムは `stencil4d` の約 1.2%であった．

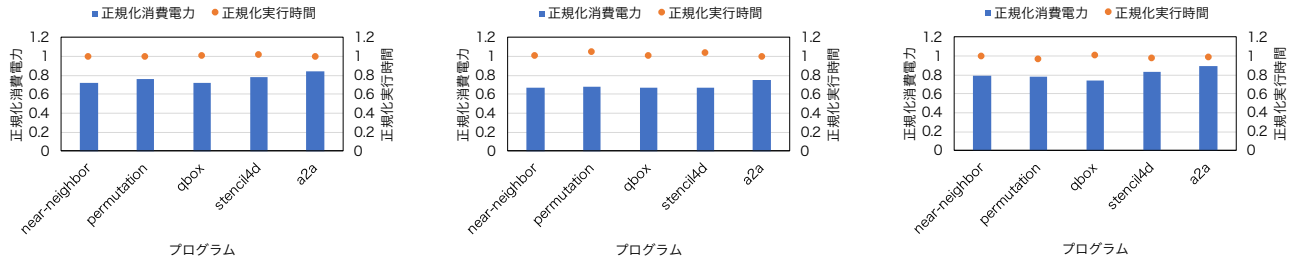
Torus においては `stencil4d` の削減率が最も高く約 33%の消費電力削減効果が確認できる．しかしながら，実行時間は約 4%増加しており，他のどのプログラムと比較しても性能の低下率は最も高い．これは，低電力モードへと遷移することが多く消費電力が削減できる一方，低電力モードから ACTIVE モードへの遷移も多いことが実行時間の増加という結果を招いたと考えられる．`a2a` の消費電力削減効果が最も低く，約 25%の削減であった．

Dragonfly では `qbox` の消費電力削減率が最も高く約 26%の削減を達成している．実行時間の増加はほとんど確認されず，最も性能が低下した `qbox` でも約 0.6%の実行時間増加という結果であった．

T_w のレイテンシが生じることから性能は低下することが予想されるが，プログラムの中には On/Off リンク適用時の性能が向上しているものが確認される．この原因はいくつか考えられる．ひとつは，T_w の遅延によるパケットの処理順序の変更である．これにより衝突回数が削減されたことにより通信時間が短縮されたことが考えられる．次に，CODES 内部で通信時間に関わる実装において一部乱数を用いる箇所があり，乱数によって通信時間が短くなる場合である．原因の詳細については現在解析中である．

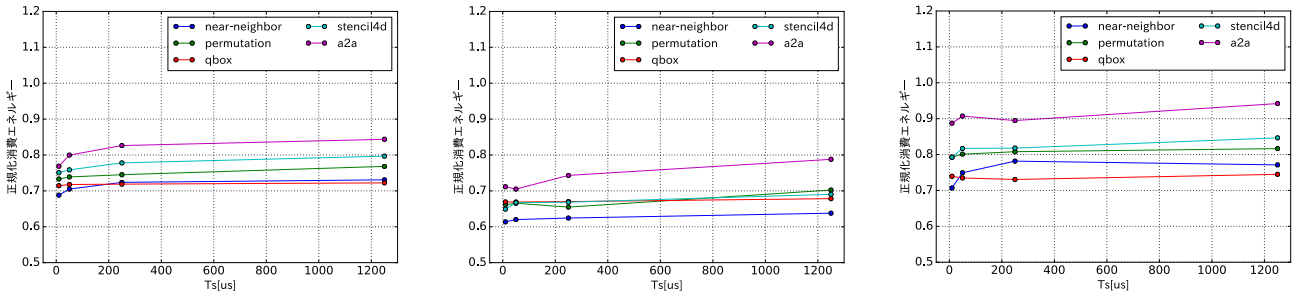
4.3 T_s の探索

T_s は On/Off リンクを適用する際に決定する重要なパラ



(a)Fat-tree (b)Torus (c)Dragonfly

図 2: On/Off リンクによる消費電力と性能



(a)Fat-tree (b)Torus (c)Dragonfly

図 3: $T_w = 17$ における T_s と消費エネルギーとの関係

メータである．適切な T_s はアプリケーション・プログラムの通信パタンに依存することから，消費電力と性能に与える影響を調査することは重要である．TraceRP を用いることで T_s を変更させた際に消費電力および性能に与える影響を解析することが可能である．$T_w = 17$ において，T_s を 10，50，250，1,250[us] とした際のリンクの正規化消費エネルギーを **図 3** に示す．横軸は T_s，縦軸は On/Off リンクを適用していない従来手法の消費エネルギーで正規化した値である．

図 3(a) に Fat-tree の結果を示す．すべてのプログラムにおいて $T_s = 10$ が最もエネルギーを削減可能である．qbox はどの T_s でもほぼ同じ消費エネルギーであり，T_s に依存しないことが分かる．その他のプログラムでは T_s の値の増加と共に消費エネルギーが増加している．これは，T_s が長くなることで低電力モードへ遷移する機会が減少しているものと考えられる．

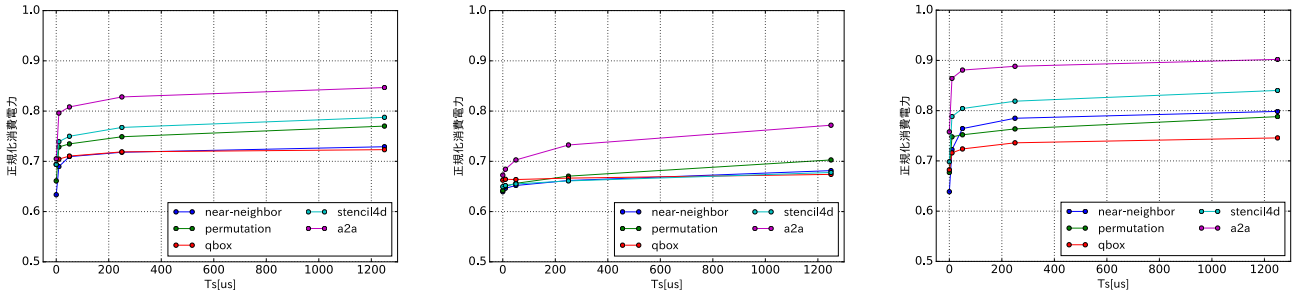
図 3(b) に示す Torus の結果は，図 3(a) に示した Fat-tree とは傾向が異なる．a2a は $T_s = 50$ で消費エネルギーの削減率が最も大きい．$T_s = 10$ と比較して低電力モードへと遷移する回数が減少するが，T_w のレイテンシが増加する回数も減少することから，性能が $T_s = 10$ よりも約 3.2 パーセントポイント改善される．一方，消費電力は増加するが，その値は 0.6 パーセントポイントに留まっていることが分かった．したがって，a2a では $T_s = 10$ において消費エネルギーが最小になったと考えられる．

Dragonfly の結果を図 3(c) に示す．図 3(b) に示した Torus の結果と同様に，エネルギーが最小となる T_s の値がプログラムによって異なることが分かる．qbox では $T_s = 50$ において消費エネルギー削減率が最大となる一方，その他のプログラムでは $T_s = 5$ で最小となる．また，プログラム間で消費エネルギーの削減率が大きく異なっていることが分かる．qbox では約 27%の消費電力を削減しているが，a2a では最大でも 12%の削減に留まるという結果であった．

4.4 T_s が消費電力に与える影響

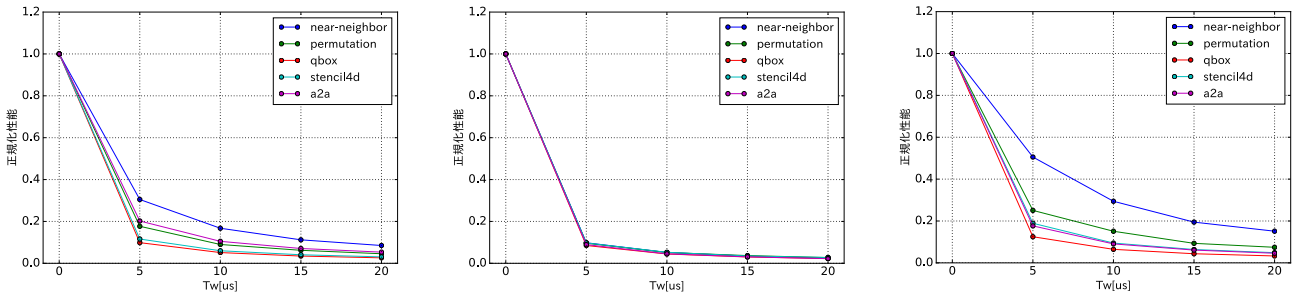
T_s の値が小さいほど低電力モードへと遷移する回数が増加するため，低電力モード中にパケット処理が生じ T_w 時間の遅延が性能を低下させる．したがって，T_s は消費電力削減効果と性能低下を考慮して適切に設定しなければならない．本稿では，$T_w = 0$ として T_s のみが消費電力にどのような影響を与えるのかを調査した．**図 4** にその結果を示す．横軸は T_s，縦軸は On/Off リンクを適用していない従来手法における電力で正規化した値である．このとき，T_s の値は，0，10，50，250，1,250[us] とした．

図 4(a) は Fat-tree における結果を示している．$T_s = 0$，すなわち，通信完了後直ちに低電力モードへ遷移させた場合，near-neighbor が最も電力を削減可能であることが分かる．T_s の値を大きくすると，消費電力削減率がプログラムごとに異なることが分かる．これは，プログラムにおける通信パタンが異なっており，低電力モードへの遷移ができない，つまり通信間隔よりも T_s が長いプログラムほど削減効果が小さくなることを示している．最も T_s の値に



(a)Fat-tree (b)Torus (c)Dragonfly

図 4: T_s が消費電力に与える影響



(a)Fat-tree (b)Torus (c)Dragonfly

図 5: T_w が性能に与える影響

左右されないプログラムは qbox であり，T_s の値を大きくしても電力の削減利率の変化は小さい．

Torus の結果を図 4(b) に示す．$T_s = 0$ においてプログラム間で電力削減効果のばらつきが小さいことが分かる．a2a を除いたプログラムは，T_s の値を変更しても削減率の変化は小さいという特徴があることが分かる．

図 4(c) に Dragonfly の結果を示す．傾向は Fat-tree と類似している．Fat-tree と同様に，qbox は他のプログラムと比較して，T_s の値を変更した際の電力削減率の変化は少ない．T_s を 50[us] としても 250[us] と削減効果に大きな差はないと言える．

4.5 T_w が性能に与える影響

$T_s = 0$ として，各通信が完了後直ちに低電力モードへと遷移させ，T_w が性能に与える影響を調査した．T_w は 0，5，10，15，20[us] の値を用いた．**図 5** に，On/Off リンクを適用していない場合の性能で正規化した結果を示す．図 5(a) に Fat-tree における結果を示す．$T_w = 5$ で性能が低下し T_w が 15 または 20 においては性能低下率はほぼ同じである．near-neighbor は他のプログラムと比較して性能の低下率は低い．一方，qbox および stencil4d は性能の低下率が大きく，$T_w = 5$ で約 90%の低下が確認できる．

Torus の結果を図 5(b) に示す．すべてのプログラムが同じような性能低下の傾向を示していることが分かる．どのプログラムも $T_w = 5$ で約 90%の性能低下が生じ，$T_w = 20$ においては性能が約 3%にまで低下する．

図 5(c) に Dragonfly における結果を示す．第 4.4 節と同様に図 5(a) の Fat-tree の結果と似た傾向が確認できる．near-neighbor が最も性能低下率が小さく，qbox が最も性能低下率が大きいことは Fat-tree の場合と同じである．一方，stencil4d は Fat-tree と異なり，Dragonfly では near-neighbor についで性能低下率が小さい．T_w が性能に与える影響はトポロジによって異なることが確認できる．T_w はスイッチなどのネットワークのハードウェアを設計する際に決定されるパラメータである．与えられた T_w とプログラムの特性から適切なトポロジを選択することが必要になると考えられる．

T_w の値を大きくすると，すべてのトポロジにおいて大きく性能が低下している．この理由として，プログラムの特性が挙げられる．評価に用いたプログラムは，通信性能の評価を前提としており，プログラムのほとんどが通信により構成される．したがって，T_w の値が大きくなるとプログラムの実行時間が大幅に増加することから，大きな性能低下が生じているものと考えられる．

5. おわりに

HPC システムにおけるネットワークの電力推定シミュレータ TraceRP を開発した．ネットワークの低電力化技術として On/Off リンクを適用することを前提に，電力を見積る機能を TraceR に実装した．低電力モードから ACTIVE

モードへの遷移中にはパケットを処理できず遅延が生じる．HPC システムにおいて性能は重要なことから，遅延時間も考慮した性能予測機能も実装した．Fat-tree，Torus，Dragonfly の 3 つのトポロジを対象にアプリケーションを用いて On/Off リンク適用時の消費電力および性能を評価した．また，On/Off リンクにおいて重要なパラメータである T_s と T_w について解析を実施した．

今後は，マルチジョブ実行時における消費電力削減効果について解析する予定である．また，大規模なネットワーク構成を対象に On/Off リンクの電力削減効果および性能への影響について調査する．

謝辞 TraceRP の実装に助言を与えてくれた Dr. Nikhil Jain, Dr. Abhinav Bhatele に感謝いたします．本研究は，一部，JST CREST「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」ならびに JSPS 科研費 JP16K16027 による．また，本研究の結果の一部は九州大学情報基盤研究開発センターの研究用計算機システムによる．

参考文献

[1] Acun, B., Gupta, A., Jain, N., Langer, A., Menon, H., Mikida, E., Ni, X., Robson, M., Sun, Y., Totoni, E., Wesolowski, L. and Kale, L.: Parallel Programming with Migratable Objects: Charm++ in Practice, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 647–658 (2014).

[2] Acun, B., Jain, N., Bhatele, A., Mubarak, M., Carothers, C. D. and Kale, L. V.: Preliminary Evaluation of a Parallel Trace Replay Tool for HPC Network Simulations, *Workshop on Parallel and Distributed Agent-Based Simulations* (2015).

[3] Alonso, M., Coll, S., Martínez, J.-M., Santonja, V., López, P. and Duato, J.: Dynamic Power Saving in Fat-tree Interconnection Networks Using on/off Links, *Proceedings of the 20th International Conference on Parallel and Distributed Processing* (2006).

[4] Badia, R. M., Escalé, F., Gabriel, E., Gimenez, Judit and, K. R., Labarta, J. and Müller, M.: Dimemas: Predicting MPI applications behaviour in Grid environments, *Workshop on Grid Applications and Programming Tools* (2003).

[5] Carothers, C. D., Bauer, D. and Pearce, S.: ROSS: a high-performance, low memory, modular time warp system, *Proceedings Fourteenth Workshop on Parallel and Distributed Simulation*, pp. 53–60 (2000).

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

[7] Christensen, K., Reviriego, P., Nordman, B., Bennett, M., Mostowfi, M. and Maestro, J. A.: IEEE 802.3az: the road to energy efficient ethernet, *IEEE Communications Magazine*, Vol. 48, pp. 50–56 (2010).

[8] Gunaratne, C. and Christensen, K.: Ethernet Adaptive Link Rate: System Design and Performance Evaluation, *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, pp. 28–35 (2006).

[9] Gunaratne, C., Christensen, K., Nordman, B. and Suen, S.: Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR), *IEEE Transactions on Computers*, Vol. 57, No. 4, pp. 448–461 (2008).

[10] Hideki, M., Ryutaro, S., Hidetomo, S., Tomoya, H., Jun, M., Makoto, Y., Takayuki, K., Yuichiro, A., Ikuo, M., Toshiyuki, S., Yuji, O., Hisashige, A., Yuichi, I., Koji, I., Mutsumi, A. and Kazuaki, M.: NSIM: An Interconnection Network Simulator for Extreme-Scale Parallel Computers, *IEICE Transactions on Information and Systems*, Vol. E94.D, No. 12, pp. 2298–2308 (2011).

[11] Huang, C., Lawlor, O. and Kalé, L. V.: Adaptive MPI, *Proceedings of the 16th International Workshop on Languages and Compilers for Parallel Computing (LCPC 2003), LNCS 2958*, pp. 306–322 (2003).

[12] Jain, N., Bhatele, A., White, S., Gamblin, T. and Kale, L. V.: Evaluating HPC Networks via Simulation of Parallel Workloads, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 14:1–14:12 (2016).

[13] Kogge, P. M.: Architectural Challenges at the Exascale Frontier, *Simulating the Future: Using One Million Cores and Beyond (invited talk)* (2008).

[14] Li, J., Huang, W., Lefurgy, C., Zhang, L., Denzel, W. E., Treumann, R. R. and Wang, K.: Power shifting in Thrifty Interconnection Network, *2011 IEEE 17th International Symposium on High Performance Computer Architecture*, pp. 156–167 (2011).

[15] Messina, P. and Lee, S.: The U.S. Exascale Computing Project, *The 2016 ACM/IEEE Conference on Supercomputing (Birds of a Feather)* (2016).

[16] Miwa, S. and Nakamura, H.: Profile-Based Power Shifting in Interconnection Networks with On/Off Links, *Proceedings of the 2015 ACM/IEEE Conference on Supercomputing*, pp. 37:1–37:11 (2015).

[17] Mubarak, M., Carothers, C. D., Ross, R. B. and Carns, P.: Enabling Parallel Simulation of Large-Scale HPC Network Systems, *IEEE Trans. Parallel Distrib. Syst.*, Vol. 28, No. 1, pp. 87–100 (2017).

[18] Reviriego, P., Sivaraman, V., Zhao, Z., Maestro, J. A., Vishwanath, A., Snchez-Macian, A. and Russell, C.: An energy consumption model for Energy Efficient Ethernet switches, *2012 International Conference on High Performance Computing Simulation (HPCS)*, pp. 98–104 (2012).

[19] Saravanan, K. P., Carpenter, P. M. and Ramirez, A.: Power/Performance Evaluation of Energy Efficient Ethernet (EEE) for High Performance Computing, *Proceedings of the 2013 IEEE International Symposium on Performance Analysis of Systems and Software*, pp. 205–214 (2013).

[20] Saravanan, K. P., Carpenter, P. M. and Ramirez, A.: A Performance Perspective on Energy Efficient HPC Links, *Proceedings of the 28th ACM International Conference on Supercomputing*, pp. 313–322 (2014).

[21] Totoni, E., Jain, N. and Kale, L.: Power Management of Extreme-scale Networks with On/Off Links in Runtime Systems, *ACM Transactions on Parallel Computing*, Vol. 1, No. 2, pp. 16:1–16:21 (2015).

[22] U. S. Department of Energy: *Top Ten Exascale Research Challenges* (2014).

[23] Zheng, G., Kakulapati, G. and Kalé, L. V.: BigSim: a parallel simulator for performance prediction of extremely large parallel machines, *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.* (2004).

[24] 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（ACS），Vol. 7, No. 4, pp. 67–83 (2014).