Vol.2010-HPC-124 No.16
2010/2/23

大規模並列システムにおける電力最適化実行時の消費エネルギー予測手法

木村英明[†1,†2] 今田貴之[†1] 佐藤三久[†1]

近年，PC クラスタに代表される大規模並列システムを構築する上でシステムの省電力性が重要になっている．高性能計算向けアプリケーションの多くはプロファイルを用いた電力最適化が有効であるが，大規模システムにおいて電力最適化を行った報告は少ない．本稿では，大規模並列システムにおいてプロファイルを用いた電力最適化手法を適用した時の消費エネルギー削減効率について議論する．小規模システムにおいて並列化効率と周波数寄与率を学習することで，大規模並列システムでの消費エネルギー削減効率を予測する．評価結果より，電力最適化手法適用時の消費エネルギー削減効率はアプリケーション特性によって変化することが分かった．また，大規模並列システムにおいても適切な電力最適化手法を適用することで消費エネルギーを削減できることが分かった．

Predicting Optimized Energy Efficiency of Large-scale PC Cluster System

Hideaki Kimura,[†1,†2] Takayuki Imada[†1] and Mitsuhisa Sato [†1]

Recently, it has become important to improve the energy efficiency of high performance PC cluster systems. Profile-based optimization, which defines program regions that have the almost same characteristics and selects the best P-State by analyzing profiles, can achieve good energy saving. However, it is difficult to evaluate the energy efficiency in the large-scale cluster system, and the scalability of optimized energy efficiency has not been discussed.

In this paper, we propose a method of predicting optimized energy efficiency of the large-scale PC cluster system. We predict the scalability of optimized energy efficiency by using performance effects of parallel processing and frequency scaling obtained in a small system. As a result, we found that the power-aware scalability is defined by application characteristics, especially speedup effects of parallel processing. And it was forecast that the energy optimization method achieves energy saving in the large-scale PC cluster system.

1. はじめに

PC クラスタに代表される並列システムを構築する上で，システム全体の消費電力が重要になってきている．システムの消費電力増加によって電力コストが増加するのみならず高価な冷却システムが必要となる．また，信頼性の低下や実装密度の低下といった大規模並列システムを構築する上での阻害要因にもなる．このような問題に対し，動作周波数と動作電圧を動的に制御する DVFS（Dynanmic Voltage and Frequency Scaling）機構を有効に利用する方法が提案されている．DVFS 機構を適切に制御することで，高い性能と省電力性を両立することができる．PC クラスタ等で頻繁に実行されるアプリケーションの多くは，特定の処理を繰り返し実行する傾向があるため，プログラムを複数領域に分割し各領域のプロファイルを取得することで消費エネルギーを大幅に削減することができる．現在，このような電力最適化を行う手法が数多く提案されている[1)2)]．

しかしながら，これらの研究の多くは研究室等で設置されている比較的小規模な PC クラスタにおいて評価が行われるのみであり，大規模システムへの適用はほとんど行われていない．大規模システムへの適用例が少ない理由として電力最適化手法の導入コストや信頼性への不安といったことも挙げられるが，“大規模システムにおいて期待した消費エネルギー削減を実現できるか明らかではない”，“電力最適化手法の消費エネルギー削減効率に関するスケーラビリティが明らかではない” といったことも挙げられる．

そこで本稿では，小規模システムの電力最適化効率を学習することで大規模システムで電力最適化を行った際の消費エネルギー削減効率を予測する手法について検討する．高性能計算分野で頻繁に利用される並列アプリケーションを対象とし，アプリケーションの特徴から複数の領域を定義し各領域で電力最適化を行うことを想定する．このような最適化を大規模システムに適用した場合，ノード数の変化によって消費エネルギー削減効率がどのように変化するか，すなわち消費エネルギー削減効率に関するスケーラビリティを予測する手法について議論する．

本稿の構成は以下のようになっている．次章では前提となる電力最適化手法の概要を述

†1 筑波大学大学院 システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba
†2 日本学術振興会 特別研究員
Research Fellow of the Japan Society for the Promotion of Science

べ，第 3 章で提案する消費エネルギー削減効率予測手法について述べる．第 4 章で評価結果を示し，消費エネルギー削減効率に関するスケーラビリティについて議論する．第 5 章で関連研究について述べ，最後にまとめと今後の課題を述べる．

2. プログラム分割による電力最適化実行

プログラムを複数領域に分割し，各領域の特性に応じて適切な動作周波数を選択することで並列システム全体の消費エネルギーを削減することができる．高性能計算向けアプリケーションの多くはソースコード上の特定部分を非常に多くの回数繰り返し実行する．したがって，ソースコード中にコードをインスツルメントすることによってプログラムを複数領域に分割し，各領域の特性に応じた最適化を行うことで消費エネルギーを大幅に削減できる．

図 1 にプログラム分割による電力最適化の実行手順を示す．本稿で対象とするプログラム分割による電力最適化実行は以下の手順によって実現可能である．

(1) プログラムの特性に応じてソースコード中にコードをインスツルメントし，プログラムを複数領域に分割する．
(2) 様々な条件でプログラムを実行しプロファイルデータを取得する．
(3) 定義した各領域のプロファイルデータから最適な実行系列を決定する．

まず，プログラムを複数領域に分割する．我々はパフォーマンスカウンタ値の推移とソースコードの情報からプログラムを自動的に複数領域に分割する手法を提案している[3]．予備実行時にパフォーマンスカウンタ値の推移を観測することで，プログラムの特性をとらえ同一処理をひとつの領域と定義する．一方で，プロファイル取得のためのオーバヘッドも考慮しなければならない．領域内の実行時間が極端に短く，また非常に多くの回数実行される部分をひとつの領域と定義してはならない．ソースコードの情報からループ構造等を解析し，予備実行結果と対応づけることでオーバヘッドの爆発的な増加を避ける．本手法は機械的にプログラムを複数領域に分割するため，ユーザがプログラムの内容を熟知している必要は無い．また，パフォーマンスカウンタ値の変動から領域を定義するため処理内容に関する情報を必要としない．

次に，定義した領域の実行時間や消費電力に関するプロファイルデータを取得する．このプロファイルデータは動作可能なすべての動作周波数に対して取得する必要がある．取得したプロファイル情報をもとに，消費エネルギーや EDP（Energy Delay Product）が最小となるよう最適動作周波数を決定する．単純に各領域の消費エネルギーが最小となる動作周波数を選択する手法や，動作周波数変更のためのオーバヘッドを考慮した手法が提案されている[2]．

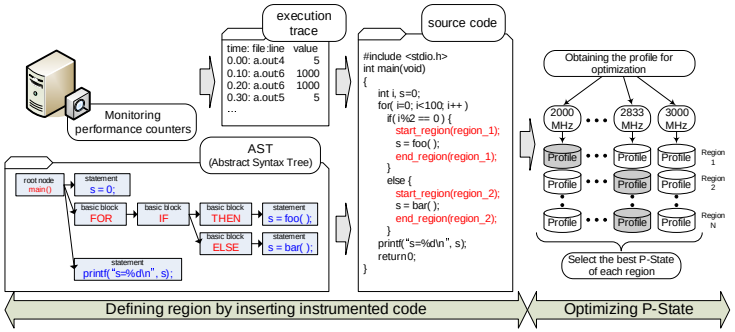

図 1 プログラム分割による電力最適化実行

このようにプログラムを複数の領域に分割することにより，計算や通信といった特性に応じて適切な最適化を行うことが可能になる．高性能計算分野で実行されるアプリケーションの多くは，計算と通信など異なる特性の処理を繰り返し実行する傾向がある．したがって，定義した各領域において特性に応じた最適化を行うことで消費エネルギーの大幅な削減が期待できる．

3. 消費エネルギー削減率予測手法

前章で述べたエネルギー最適化手法を大規模システムに適用することを考える．ここで，ノード数の変化にともない消費エネルギー削減効率がどのように変化するかを予測することが本稿の目的である．本章では，小規模システムで学習したデータをもとに大規模システムの消費エネルギー削減効率を予測する手法について検討する．

n ノードのシステムにおいて標準周波数 f_{std} でプログラムを実行した時，ある領域 r の実行時間を $T_{std}(n, r, f_{std})$，システム全体の平均消費電力を $P_{std}(n, r, f_{std})$ とする．また，最適化実行時も同様に領域を r を f_{opt} で実行した時の実行時間，消費電力を $T_{opt}(n, r, f_{opt})$，$P_{opt}(n, r, f_{opt})$ と定義する．ここで，消費エネルギー削減効率 ΔE を式 (1) のように定義する．

$$\Delta E = 1 - \frac{\sum_{r=1}^{R} P(n, r, f_{opt}) T(n, r, f_{opt})}{\sum_{k=1}^{R} P(n, r, f_{std}) T(n, r, f_{std})} \quad (1)$$

Vol.2010-HPC-124 No.16
2010/2/23

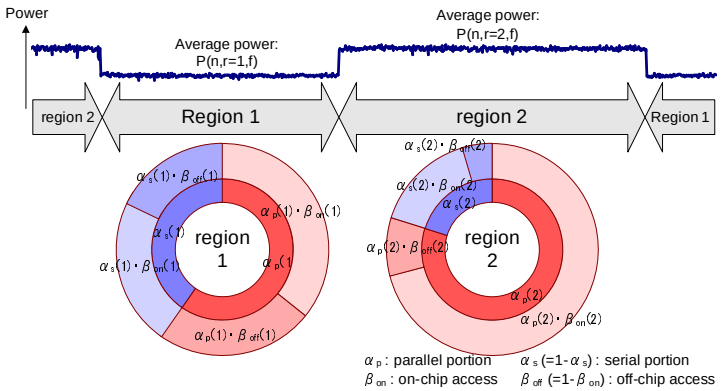

図 2 消費エネルギー削減アルゴリズムの概要

3.1 ノード数の変化と消費電力

本節では，ノード数と消費電力の関係について考える．1 ノード当たりの消費電力は動作周波数や処理内容によって変化する．しかしながら，データ分割による並列化においてはノード数が変化したとしても各ノードの処理量が変化するのみで処理内容は変化しない．本稿で前提としているプログラム分割による電力最適化では定義された領域内では同一の処理のみを行うため，領域 r における各ノードの消費電力は動作周波数のみによって決定し，領域内で変動することはない．すなわち，ある領域 r を動作周波数 f で実行した時，システム全体の消費電力は単純にノード数に比例すると仮定でき，$P(n,r,f) = nP(1,r,f)$ が成立する．ここで，$P(1,r,f)$ は最適化実行のためのプロファイル取得時に実データとして取得可能である．電力最適化実行時の消費エネルギー削減効率を予測する際には，この実データを利用する．

3.2 ノード数の変化と実行時間

本節では，ノード数の変化と実行時間の関係について述べる．本稿で提案する手法は，プログラムを複数領域に分割することを前提としている．したがって，プログラム全体の実行時間の変化ではなく定義した領域ごとに実行時間を予測する必要がある．本稿では**図 2** に示すように各定義領域の並列化可能割合と動作周波数と性能の関係に着目することで大規模システムにおける消費エネルギー削減効率を予測する．

領域 r において，並列化により速度向上に寄与する割合（以降，並列化効率）を $\alpha_p(r)$，寄与しない割合を $\alpha_s(r)(= 1 - \alpha_p(r))$ と定義した時，式 (2) が成り立つ．

$$T(n,r,f) = \big(\alpha_s(r) + \frac{\alpha_p(r)}{n}\big)T(1,r,f) \tag{2}$$

一方，動作周波数の変化と実性能に与える影響について考える．領域 r おいて，on-chip アクセスなど動作周波数の変更によって直接影響する割合（以降，周波数寄与率）を $\beta_{on}(r)$，off-chip アクセスなど動作周波数の変更に影響しない割合を $\beta_{off}(r)(= 1 - \beta_{on}(r))$ と定義する．この時，動作周波数の変更による実行時間の変化は式 (3) で与えられる．

$$T(1,r,f_{opt}) = \big(\frac{f_{std}}{f_{opt}}\beta_{on}(r) + \beta_{off}(r)\big)T(1,r,f_{std}) \tag{3}$$

以上をまとめると，式 (1) は式 (4) に変換される．

$$\Delta E = 1 - \frac{\sum_{r=1}^{R} P(1,r,f_{opt})(1 - \frac{n-1}{n}\alpha_p(r))\big((\frac{f_{std}}{f_{opt}} - 1)\beta_{on}(r) + 1\big)T(1,r,f_{std})}{\sum_{r=1}^{R} P(1,r,f_{std})(1 - \frac{n-1}{n}\alpha_p(r))T(1,r,f_{std})} \tag{4}$$

ここで，$P(1,r,f_{std})$，$P(1,r,f_{opt})$，$T(1,r,f_{std})$ は電力最適化のためのプロファイルデータを利用することが可能である．一方，$\alpha_p(r)$，$\beta_{on}(r)$ はプロファイルデータに対して最小二乗法を適用することで求める．$\alpha_p(r)$，$\beta_{on}(r)$ は，それぞれ式 (5)，式 (6) で与えられる．

$$\alpha_p(r) = \frac{\sum_n \big(\frac{1}{n} - 1\big)\big(\frac{T(n,r,f)}{T(1,r,f)} - 1\big)}{\sum_n \big(\frac{1}{n} - 1\big)^2} \tag{5}$$

$$\beta_{on}(r) = \frac{\sum_i \big(\frac{f_{std}}{f_i} - 1\big)\big(\frac{T(1,r,f_i)}{T(1,r,f_{std})} - 1\big)}{\sum_i \big(\frac{f_{std}}{f_i} - 1\big)^2} \tag{6}$$

4. 評　　価

4.1 評 価 環 境

評価環境として，Intel Core2Quad Q9650 を搭載した 16 ノードのクラスタシステムを用いた．**表 1** にノード構成を示す．Intel Core2Quad Q9650 は DVFS 機構を備えており，6 段階の P-State（3,000 MHz，2,833 MHz，2,667 MHz，2,500 MHz，2,333 MHz，2,000 MHz）を利用することが可能である．

電力評価環境として，ホール素子を用いた電力測定システム PowerWatch[2] を用いる．各ノードは電力測定装置を介して UPS に接続している．一般的に，大規模システムにおいて

表 1 測定対象クラスタシステム

CPU	Intel Core2Quad Q9650
Default Frequency	3,000 MHz
Memory	DDR2 SDRAM 8GB
HDD	HGST HDT721010SLA360
Network	Gigabit Ethernet
Kernel	Linux 2.6.28-perfctr
Compiler	gcc 4.1.2
MPI	OpenMPI 1.3.2

各ノードの詳細な消費電力データを取得することは困難であるため，ノード単位での電力測定を行った．したがって，評価に用いる電力値はプロセッサの消費電力のみならずメモリや HDD などのその他構成要素，電源における AC-DC 変換損失等を含んでいる．

評価ベンチマークとして、NPB（NAS Parallel Benchmarks）-3.3 よりカーネルベンチマーク 5 種を用いた．問題サイズは CLASS=C とし，標準の入力データセットを利用した．

4.2 最適化時における消費エネルギー削減率の変化

プログラムを複数領域に分割し，プロファイルを用いた最適化実行を行う．2 ノード，4 ノード，8 ノードを用いて学習し，16 ノード実行時の消費エネルギー削減効率を予測する．**表 2** に各ベンチマークにおいて定義した領域，電力最適化時の動作周波数，16 ノード実行時の消費エネルギー削減効率予測値 ΔE_{pred}，消費エネルギー削減効率実測値 ΔE_{real}，最適化実行時の消費エネルギーの比較結果をそれぞれ示す．

メモリアクセス時に低い動作周波数を選択することでシステム全体の消費エネルギーを削減する MG では，最適化時の消費エネルギー削減率，消費エネルギー値ともに予測値と近い値となっている．しかしながら，通信時に低い動作周波数を選択することで消費エネルギーを削減するベンチマークでは消費電力予測結果と実測値に差があることを確認できる．これは，通信を行う領域に対しても並列化効率と周波数寄与度のみを用いた実行時間予測を行ったためである．本提案手法では，分割された各領域の実行時間は同一の手法によって予測しており計算部・通信部といった処理内容を明確に意識していない．すべての領域で並列化効率と周波数寄与度から線形近似により実行時間を予測している．CG で行われる一対一通信ではノード数の増加にともない通信時間が増加する可能性がある．通信の種類や規模によって通信方式を変更する集合通信においても，単純な線形近似では正しく性能を予測することは難しい．より高い精度の予測を実現するためには，各領域の処理内容を解析し，処理特性に応じて適切な消費エネルギー予測手法を適用しなければならない．

また，高い動作周波数を選択している領域では，β_{on} が 1 に近い値となっていることが確認できる．これらの領域では，低い動作周波数を選択することで性能が大幅に低下するため，電力最適化手法によって高い動作周波数が選択されていると考えられる．

4.3 大規模並列環境における電力最適化時の消費エネルギー削減率予測

本節では式 (4) を用い，理想的なワークロードと MG に対し電力最適化アルゴリズムを適用した時の消費エネルギー削減効率にとノード数の関係について述べる．

理想的なワークロードとして，3 種類の典型的な処理 “計算領域 $r(calc)$”，“メモリアクセス領域 $r(mem)$”，“通信領域 $r(comm)$” について考える．各領域のパラメタを**表 3** に示す．これらの値は前節の評価結果をもとに決定した．これらの処理の組み合わせからなるプログラムを想定し，電力最適化アルゴリズムを適用した時の電力削減効果について考える．

シミュレーション結果を**図 4** に示す．結果より，$r(calc)$-$r(mem)$，$r(calc)$-$r(comm)$ ではノード数の増加に従って消費エネルギー削減効率が上昇しているのに対し，$r(mem)$-$r(comm)$ ではノード数の増加とともに消費エネルギー削減効率は低下している．これは，並列化効率 α_p と，電力最適化時のエネルギー削減量が影響している．

今回のシミュレーション環境において，計算領域 $r(calc)$ の実行時間はノード数の増加に反比例して減少するが，それ以外の領域では計算領域 $r(calc)$ と比較してゆるやかに実行時間が減少する．したがって，ノード数の増加によって計算領域以外，すなわち電力最適化により消費エネルギーを削減することが可能な領域の割合が増加する．これにより，計算領域とそれ以外の領域の組み合わせではノード数の増加に従って消費エネルギー削減効率が上昇する．一方 $r(mem)$-$r(comm)$ では，$r(mem)$ のほうが並列化効率が高いためノード数の増加に従ってプログラム全体の実行時間に占める割合が減少し，$r(comm)$ の実行時間が相対的に増加する．すなわち，ノード数が増加するにつれ $r(comm)$ における消費エネルギー削減効果が主導的となる．ここで，$r(comm)$ と比較して $r(mem)$ での消費エネルギー削減効率が高かったならば，ノード数の増加に従ってプログラム全体の消費エネルギー削減効率は低下することになる．

次に，MG における電力最適化手法のノード数特性を予測する．**図 3** に MG に対して電力最適化手法を適用した時のノード数と消費エネルギー予測結果を示す．標準周波数でプログラムを実行した際の消費エネルギーを 1 とし，各領域で消費する消費エネルギーの割合を示している．ノード数の増加に従って領域 rprj3()，領域 psinv() で消費する消費エネルギー割合が増加しているのに対し，消費エネルギー削減効率が高い領域 resid() で消費する消費エネルギー割合は減少していることが分かる．ノード数が少ない時は領域 resid() の電

表 2 最適化時における消費エネルギー削減率予測と結果

	defined region	Optimized frequency	α_p	β_{on}	$P(1,r,f_{std})$	$P(1,r,f_{opt})$	ΔE_{pred}	ΔE_{real}	$E_{pred,opt}/E_{real,opt}$
MG	rprj3()	2,000 MHz	0.709	0.615	140.7	115.3	12.3%	13.3%	101.9%
	psinv() for coarse grid	2,000 MHz	1.000	0.774	148.8	113.6			
	interp()	2,000 MHz	1.000	0.010	147.1	100.4			
	resid()	2,000 MHz	0.920	0.152	156.6	115.3			
	psinv() for smooth grid	2,000 MHz	0.822	0.219	140.0	113.6			
FT	evolve()	2,667 MHz	0.960	0.000	140.5	127.5	2.9%	5.0%	134.4%
	cffts1() (before comm)	3,000 MHz	1.000	1.000	143.5	142.9			
	transpose_x_yz()	2,500 MHz	0.750	0.720	139.2	118.2			
	cffts2() and cffts1() (after comm)	3,000 MHz	1.000	1.000	143.6	143.3			
IS	determine the number of keys	2,000 MHz	0.901	0.408	124.0	99.9	4.1%	2.0%	91.4%
	sort into appropriate bucket	2,000 MHz	0.944	0.567	120.5	99.2			
	MPI_Alltoallv()	2,833 MHz	0.662	1.000	140.3	126.7			
CG	conj_grad()	2,333 MHz	0.317	0.375	142.8	112.2	5.3%	15.3%	108.4%
	MPIs (Irecv, Send, and Wait)	2,333 MHz	-4.520	0.871	145.8	119.7			
EP	main loop	3,000 MHz	0.995	1.000	121.7	121.7	0%	0%	104.4%

表 3 シミュレーション環境

	$r(calc)$	$r(mem)$	$r(comm)$
α_p	1.0	0.9	0.7
β_{on}	1.0	0.1	0.7
P_{opt}	125	115	120
P_{std}	125	150	140
optimized frequency	3000	2000	2500

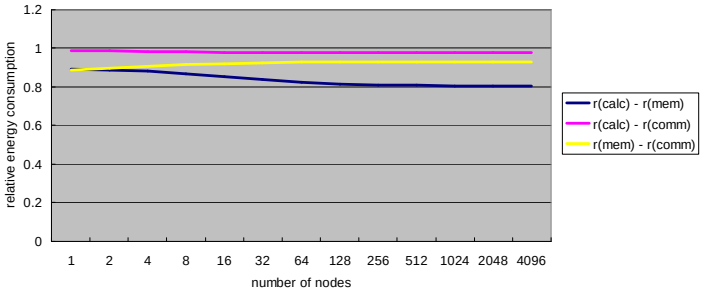

表 4 シミュレーションによる消費エネルギー削減効率の変化

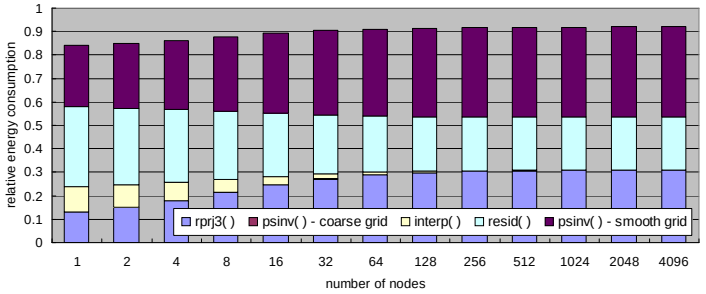

図 3 MG 実行時のノード数と消費エネルギー削減効率

力削減効果によりプログラム全体の消費エネルギー削減効率が高くなっているが，ノード数の増加に従ってプログラム全体の消費エネルギー削減効率は低下していく．しかしながら，消費エネルギー削減効率は一定の値に収束し，適切な最適化を行っていれば標準実行時の消費エネルギーを超えることはない．

5. 関連研究

近年，並列システムにおいてノード数を変化させた際の実行時間・消費電力の変化を予測する手法が提案されている．Rong らは，プログラム全体を逐次部と並列部，on-chip アクセス部と off-chip アクセス部に分割し，ノード数を変化させた際の消費エネルギーを予測する手法を提案している[4)5)]．我々の提案する手法も Rong らと同様に逐次部と並列部，動作周波数と実性能の関係に着目している．Rong らはプログラム全体を一定の周波数で動作させる事を前提としているが，我々はプログラム中で動作周波数を変化させて最適化実行した時の消費エネルギー削減効率を予測している．Sangyeun らは，アムダールの法則に基づいてプログラムの並列化可能割合とシステム全体の消費エネルギーの関係について述べている[6)]．いずれの手法も，一定の動作周波数でプログラムを実行した際の消費エネルギーを予測するものであり，“最適化実行によってどの程度消費エネルギーを削減できるか” といったことに着目していない．

ノード数と性能の関係については，アムダールの法則をはじめとして数多くの手法が提案されている[7)]．Bradley らは，回帰分析により大規模システムの性能を予測する手法を提案している[8)]．久保田らは，プログラムを演算部と通信部に分割し並列プログラムの性能予測を行っている[9)]．実行サイクル数とネットワークシミュレータにより，計算部と通信部の実行時間を予測している．我々の手法もプログラムを特徴に応じて分割し，分割した領域ごとに消費エネルギー削減効率の予測を行っているが，処理内容を明確に意識した予測ではない．

近年ではマルチコアプロセッサ環境においてアムダールの法則を適用した報告も多数ある[10)]．Dong らは，マルチコアプロセッサ環境において性能・電力のスケーラビリティについて述べている[11)]が，クラスタシステムについて言及していない．

6. おわりに

本稿では，大規模システムにおいて電力最適化を行った時の消費エネルギー削減効率を予測する手法について述べた．高性能計算向けアプリケーションの多くはプログラムコードを複数領域に分割し，各領域において適切な動作周波数を選択することでシステム全体の消費エネルギーを大幅に削減することができる．このような電力最適化実行を大規模システムに適用した際の消費エネルギー削減効率について予測する手法を提案した．提案手法では，小規模システムにおいて対象となるアプリケーションの並列化効率，動作周波数変更が性能に影響する寄与度を学習し，大規模システムで電力最適化を行った時の消費エネルギー削減効率を予測する．

電力測定環境を構築し，電力最適化時の消費エネルギー削減効率とノード数の関係について評価を行った．その結果，メモリアクセス時に低い動作周波数を選択することで消費エネルギーを削減するアプリケーションに対しては，高い精度で消費エネルギー削減効率を求められることが分かった．また，電力最適化時の消費エネルギー削減効率は定義領域の特性によって変化するものの，大規模並列システムにおいても適切な電力最適化手法を適用することで消費エネルギーを削減できることが分かった．

今後の課題として，通信を含むアプリケーションに対する消費エネルギー削減効率予測の高精度化が挙げられる．電力最適化実行のために定義した領域の処理内容に関する情報を取得するとともに，処理内容に応じて消費エネルギーの予測方法を変更することで消費エネルギー削減効率予測を高精度化できると考えられる．

参考文献

1) Rong Ge, Xizhou Feng, and Kirk W. Cameron. Performance-constrained Distributed DVS Scheduling for Scienti c Applications on Power-aware Clusters. In *SC05*, 2005.
2) Y. Hotta, M. Sato, H. Kimura, S. Matsuoka, T. Boku, and D. Takahashi. Profile-based Optimization of Power-Performance by using Dynamic Voltage Scaling on a PC cluster. In *Workshop on High-Performance Power-Aware Computing*, 2006.
3) 木村英明, 佐藤三久, 堀田義彦, 今田貴之. 影響の少ないインスツルメント手法と電力最適化のためのプログラム領域分割. 情報処理学会論文誌 Vol.48 No.SIG13 （ACS19）, 2007.
4) Rong Ge and KirkW. Cameron. Power-Aware Speedup. In *IEEE International Symposium on Parallel and Distributed Processing*, 2007.
5) Rong Ge, Xizhou Feng, and KirkW. Cameron. Modeling and Evaluating Energy-Performance Efficiency of Parallel Processing on Multicore Based Power Aware Systems. In *IEEE International Symposium on Parallel and Distributed Processing*, pp. 80–91, 2009.
6) Sangyeun Cho and Rami Melhem. Corollaries to Amdahl's Law for Energy. *IEEE Computer Architecture Letters*, 2008.
7) G.M. Amdahl. Validity of the Single-Processor Approach to Achieving Large Scale Computing. In *AFIPS Spring Joint Computer Conference*, 1967.
8) Bradley J. Barnes, Barry Rountree, David K. Lowenthal, Jaxk Reeves, Bronis deSupinski, and Martin Schulz. A Regression-Based Approach to Scalability Prediction. In *International Conference on Supercomputing*, 2008.

Vol.2010-HPC-124 No.16
2010/2/23

9) 久保田和人, 板倉憲一, 佐藤三久, 朴泰祐. 大規模データ並列プログラムの性能予測手法と NPB 2.3 の性能評価. 情報処理学会論文誌, Vol.40, No.5, 1999.
10) MarkD. Hill and MichaelR. Marty. Amdahl's Law in the Multicore Era. *IEEE Computer*, 2007.
11) DongHyuk Woo and Hsien-HsinS. Lee. Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era. *IEEE Computer*, 2008.