Vol.2010-ARC-188 No.5
2010/3/1

実電力によるノード状態制御が可能なヘテロクラスタシステム

早川　潔[†1]　　原田　信[†1]

近年，CPU の低消費電力化および短命化に伴い，PC クラスタの低消費電力化およびコンパクト化，ヘテロ化が進んだ場合，数百ノード程度の PC ヘテロクラスタが中小企業または研究室で構築可能である．そこで，本稿では，PC ヘテロクラスタのノード実行状態が，実電力にもとづき，制御できるシステムについて検討する．電力制御には，細かく制御する制御機構（負荷変動制御機構）および比較的電力変動幅が大きい制御機構（ON/OFF 制御機構）との 2 つの方式を採用し，制御幅を大きく且つ細かく制御できる電力制御機構について検討する．予備評価として，2 つの電力制御方式に関する評価を別々に行い，電力制御幅，消費電力削減率などを測定した．その結果，電力制御幅が約 3W 程度，消費電力削減率約 14%という結果を得た．

A Heterogeneous Cluster System with Node-state Controller by Real Power

Kiyoshi Hayakawa[†1] and Makoto Harada[†1]

PC cluster systems made by genral-purpose parts(i.e. motherboard, Intel CPU, STAT HDD and so on) are good cost performance, low powe and compact. Therefore small laboratories and medium-sized companies make PC cluster systems which consist of hundreds of nodes. In this paper, we consider the possibilities of a hetero cluster system with a node-state Controller by real power. It allows to the hetero cluster system to reduce power cost in the small laboratories and medium-sized companies. There are 2 mechanisms in the system, load change mechanism and ON/OFF control mechanism. Load change mechanism allows us to control power finely, and ON/OFF control mechanism allows us to control power coarsely. In the primal evaluation, we achieved approximately 3W power control range and 14% power reduction rate.

1. はじめに

インテルの Atom に代表される CPU の低消費電力化および低価格化にともない，クラスタの低消費電力化およびコンパクト化が重要になりつつある[1)]．汎用部品（パソコンのマザーボードやインテルの CPU など）で構成された PC クラスタシステムは, そのシステム構築が比較的安価でかつ容易なため, 数十〜数百台規模のシステムに膨らんできている[2)]. また, 市販マイクロプロセッサの性能が急激に向上しており, そのプロセッサを使用する PC クラスタシステムはより高速なシステムのため，小規模な企業や研究所でも導入されている．

PC クラスタ長期運用した場合の問題点として，故障後の部品調達が難しいことが挙げられる．一般市場における CPU のライフサイクルは 2〜3 年であり，故障した時期が遅れるほど，市場で入手できにくくなる．入手できたとしても，性能の高い CPU よりも高価になっている場合が多く，その場合，性能の高い CPU に買い換えたほうが得策である．また，新たにノード台数を増やす場合にも，性能の高い CPU を増設するほうが安価になる場合が多い．そのような CPU 交換またはノード増設方法を行った場合，各ノードの性能が異なるヘテロクラスタになる．

一方，PC クラスタを小規模な研究所や中小企業に設置する場合，その消費電力が問題となる． 消費電力削減のためには，CPU を始めとする計算ノードの低消費電力化が必要である．近年，前述の通り，CPU ベンダーは低消費電力な CPU を市場に投入している．また，CPU の周波数および電源電圧を調整することにより電力制御を行っている研究もある[3)]．さらに，CO_2 および電気料金削減という観点から，施設全体の電力の平準化を目指した PC クラスタ運用が考えられる．発電機は同じ出力で使い続けることで，効率が良くなるので，施設全体の電力を平準化するということは，CO_2 削減にとって非常に重要である．また，平準化するということは，深夜にもクラスタを稼働することである．深夜に発電した電力を貯められないので，無駄になっている．その電力を PC クラスタの電力で活用すれば，深夜電力の料金は，昼間の電気料金より割安なため，低コストで済み，なおかつ昼間のみで運用する場合に比べ CO_2 削減に寄与できる．

そこで，本稿では，環境にやさしく，かつ低電力コストを目指した PC ヘテロクラスタ実行環境（以後,「エコ PC ヘテロクラスタ実行環境」とする）実現のための「実電力による

†1 大阪府立工業高等専門学校
Osaka Prefectural College of Technology

ノード状態制御が可能なヘテロクラスタシステム」について言及する．本システムでは，電力量の平準化を計るため，施設全体の電力量に従い計算ノードの稼働台数を変更するなどの電力制御機構を搭載する．

2. エコ PC ヘテロクラスタ実行環境

本節では，エコ PC ヘテロクラスタ実行環境の全体像およびそれを実現するシステムの概略について説明する．

2.1 全体像〜平準化および深夜電力を活用した CO_2 削減〜

本研究の全体像を図 1 に示す．施設全体の電力使用状況からクラスタの稼働台数を変動させる．つまり，昼間は他のパソコンやエアコンが稼働しているので，PC クラスタの稼働台数を減らし，深夜，電気を使っていないときに，PC クラスタをフル稼働させる．しかし，稼働台数を増減するだけだと，電力の増減幅が大きい．そこで，できるだけ電力を平準化させるために，CPU 負荷を変更し，細かな電力制御を可能にする．消費電力は，クラスタおよびオフィスの電力をそれぞれ測定し，それら計測された実電力を基に計算ノードの実行状態を制御する．

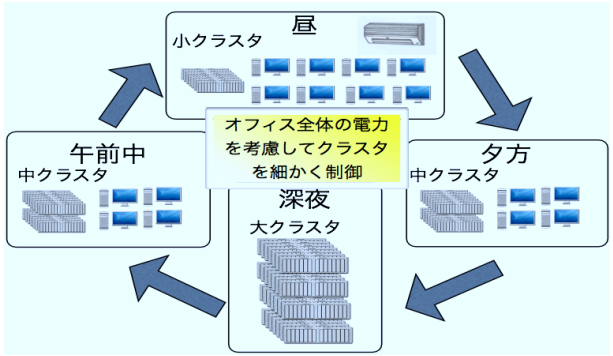

図 1 エコ PC ヘテロクラスタ実行環境の全体像

2.2 システム構成

エコ PC ヘテロクラスタ実行環境のテストベッドとして，図 2 のシステムを構築する．本システムは，電力測定装置，EMDC システム[4]，低消費電力 CPU，FPGA 計算ノード，およびホストコンピュータで構成されている．

計算ノード構成が同じケース毎に，電力測定装置を搭載する．本電力測定装置は，ケースのコンセントケーブルから電力を計測し，ネットワークを経由して，ホストコンピュータに送ることにより，電力データの収集を可能にする．この電力データと後述する実行状態制御機構を利用して，システムの電力平準化を行う．

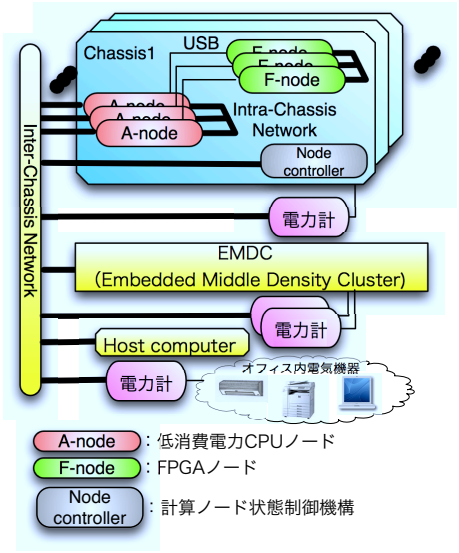

図 2 エコ PC ヘテロクラスタ実行環境のシステム構成

EMDC システムでは，PentiumM(2.0GHz) 搭載のノードが 9 台，PentiumIII 搭載のノード 30 台，およびホストコンピュータで構成されている．PentiumIII 搭載のノードでは，3 ノードのうち 1 ノードが動作周波数 600MHz の CPU で残りの 2 ノードが 700MHz の CPU である．PentiumIII や PentiumM などの比較的低動作周波数ではあるが低消費電力である CPU を利用して，低消費電力，且つコンパクトなクラスタを目指している．PentiumM ノードには 1Gbyte(デュアルチャネル)，PentiumIII には 256Mbyte のメモリが搭載されている．HDD は，コンパクトフラッシュメモリ（PentiumM・PentiumIII ノードともに 1Gbyte）を採用している．このことにより，機械稼働部品が減り，より長期運用・低消費

電力なシステムになる．

ネットワーク構成は，筐体内ネットワークの Intra-Cahssis Network および筐体外ネットワークの Inter-Chassis Network で構成されている．Intra-Chassis Network は，Gigabit Ethernet で構築した．PentiumIII ノードの Inter-Chassis Network は，100Base-TX の Ethernet で構築し，PentiumM ノードの Inter-Chassis Network は，Gigabit Ethernet で構築した．

低消費電力 CPU および FPGA 計算ノードとして，Atom マザーボード（図中「A-node」および FPGA ボード（図中「F-node」）を搭載する． A-node3 台および F-node3 台，合計 6 台を 1 つのケースに実装する．A-node および F-node は USB で結合し，計算データのやりとりを行う．A-node のネットワークは，２つのネットワーク（シャーシ内，シャーシ外）で構成する．Atom マザーボードは，Wake-On-LAN を利用して，ネットワークを介して，電源を制御する．電源を制御するために，シャーシ内にノードコントローラを設ける．ノードコントローラは，比較的低消費電力な H8 マイコン[⋆1]で構成する．

2.3 電力制御機構

本実行環境では，2 種類の方法で電力を制御する． 1 つは，CPU 負荷を変動させて電力を制御する負荷変動制御機構であり，もう 1 つは，計算ノードを起動または停止状態にする ON/OFF 制御機構である．

負荷変動制御機構では，計算途中で定期的にスリープ状態を挿入したり，計算プロセスの優先度を低くすることにより，CPU の負荷率を低下させる．

ON/OFF 制御機構では，Wake On LAN 機能を使用し，ネットワーク経由で計算ノードをシャットダウン状態にしたり，実行状態にしたりする．

2.3.1 電力データの取得

電力制御機構を実現するにあたり，電力データの取得が必要である．電力データは，ホストコンピュータおよびマイコンを利用して電力測定装置から取得される．電力データを取得する際，マイコンの起動時に，ホストコンピュータへタイマコードを送る．このコードを受信したホストコンピュータが，一定時間経過後，電力コードを送る[⋆2]．マイコンが電力コードを受信したら，電力測定装置から電力データを取得する．この電力データを基にマイコンが計算ノードの起動／停止を行う．

⋆1 イーサネットが搭載されている H8 マイコンボード（H8/3069F）を使用する．
⋆2 ホストのタイマを使うため

2.3.2 計算ノードの起動

計算ノードの起動方法について説明する．あらかじめ，マイコン内に，計算ノード毎の「稼働」,「停止」，または「保留」を表す状態変数を用意する.「稼働」および「停止」は，それぞれ，計算ノードが稼働および停止している状態を表す.「保留」はノードが起動または停止処理の実行中であることを示す．マイコンから起動または停止の処理を開始したら，ノードの状態を「保留」にし，処理完了後，状態変数を「稼働」または「停止」にする．

マイコンは起動する計算ノードの状態変数を参照し，その計算ノードの状態が「停止」であることを確認する．状態が「稼働」や「保留」であった場合は，起動処理は行わない．マイコンがその計算のノードの状態変数が「停止」だった場合，マイコンは起動する計算ノードの状態を「保留」に変更し，起動処理を開始する．マイコンは起動する計算ノードの MAC アドレスを取得，それを用いて Wake-on-LAN で計算ノードを起動する．計算ノードが起動したら，マイコンへ起動コードを送信する．マイコンは起動コードを受信したら，送信元の IP アドレスに対応する計算ノードの状態を「稼働」に変更し，そのことをホストコンピュータに通知する．

2.3.3 計算ノードの停止

計算ノードの停止手順について説明する．マイコンは停止する計算ノードの状態変数を参照し，その計算ノードの状態が「稼働」であることを確認する．状態変数が「停止」や「保留」であった場合は，停止処理を行わない．状態変数が「稼働」の場合に，マイコンはその計算ノードの状態を「保留」に変更し，停止処理を開始する．マイコンはその計算ノードに対して，停止コードを送信する．マイコンはホストコンピュータへその計算ノードが停止したことを通知する．その後，マイコンはその計算ノードの状態変数を「停止」に変更する．

2.4 ソフトウェア構成

エコ PC ヘテロクラスタ実行環境におけるソフトウェア構成を図 3 に示す．本実行環境では，電力制御機構および負荷分散スケジューラが実行される．

電力制御機構は，ホストコンピュータ上で実行され，各計算ノードの電力およびオフィス全体の電力を監視し，設定された電力を保つように，マイコンを介して，各計算ノードの実行状態を制御する．その制御情報を負荷分散スケジューラに渡し，負荷分散を行う．

ある計算ノードがシャットダウンしようとする場合，その前にデータマイグレーションが行われる．つまり，各計算ノードが，シャットダウンに入る前に，計算途中のデータとタスクの実行進行状況を負荷分散スケジューラへ送る．負荷分散スケジューラは，計算の続きを稼働している別の計算ノードへスケジュールしなおす．グリッド上では，このマイグレー

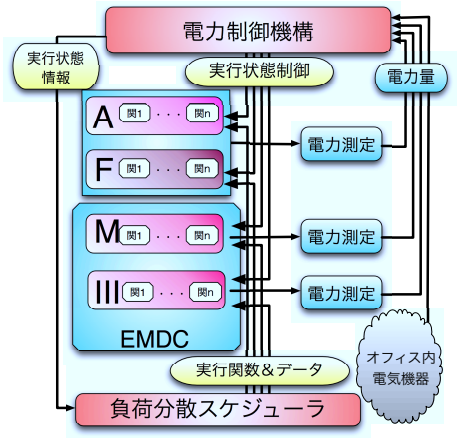

図 3 エコ PC ヘテロクラスタ実行環境におけるソフトウェア構成

ションに関する研究が盛んに行われている[5)6)]．それらの研究を参考にしつつ，本システムに合う方法を考案し，実装する．

負荷分散スケジューラもホストコンピュータ上で実行され，負荷がある特性の計算ノードに偏らないようタスク（各計算ノード行う処理の単位）をスケジュールする．ヘテロクラスタでは，各計算ノードの性能が異なるため，負荷をうまく調節しないと効率よい並列実行が望めない．従って，ヘテロクラスタでの負荷分散方式[4)] を参考に実装する予定である．

各計算ノードには，同じ関数を用意し，負荷分散スケジューラには，関数を呼び出す手順をプログラムしておく．アプリケーション実行時，関数とその関数で用いるデータがプログラムに従って各計算ノードへ送られる．その際，各計算ノードの CPU 性能および実行状態情報をもとに，送るデータ量を調整する．

3. 予 備 評 価

本予備評価では，計算ノードの電力制御可能性，計算ノードの稼働・停止の制御可能性，および消費電力変動について評価した．

3.1 CPU 負荷の変動による計算ノード消費電力の変動

CPU の負荷を変動させるため，簡単な演算をループで繰り返し，決められたループ回転数実行後に 1μsec のスリープを挿入し，それをまた繰り返した．1 分間電力を測定し，その最小値および最大値を記録した．実験結果を図 4 に示す．なお，本実験は，EMDC の PentiumM 搭載ノード 3 台に上記のプログラムを実行させ，3 台合計した消費電力を測定した．

実験の結果，ループ回転数 10000 から 100000 までは，ほぼ直線的に電力が上昇していることがわかる．また，最小値と最大値の差も小さい．したがって，あるループ回転数の領域では，ループの合間にスリープを定期的に挿入することにより，リニアに電力を制御可能（少なくとも 3W 刻みで制御可能）である．

しかし，1000000 から 1200000 までは，電力が下がり始めた．それとともに，電力の最小値の最大値の差が，大きくなった．この原因は，現在調査中である．

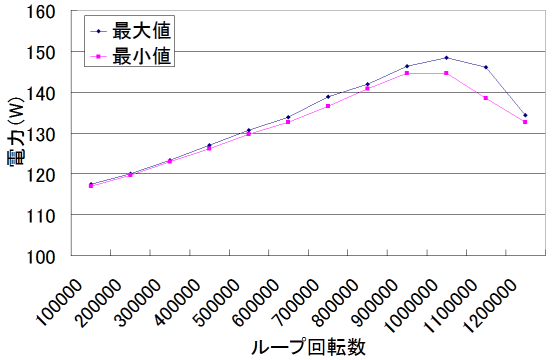

図 4 スリープ命令の間に実行されるループ回転数と電力の関係

3.2 ON/OFF 制御機構による消費電力の変動

ON/OFF 制御機構による制御可能性と消費電力削減可能性を検証するために，計算ノードの起動，シャットダウンおよびマイグレーションに要する時間を計測し，計算ノード・PC の負荷変動時の消費電力を測定した．また，ON/OFF 制御時の消費電力を測定した．測定中に実行したアプリケーションはモンテカルロ法による円周率計算である．

本実験のマイグレーションおよびタスクの負荷分散は,「openMosix」を使用した．openMosix とは，Linux のプロセスをネットワーク経由で他のクラスタノードに実行させるいわゆるマイグレーションの仕組みである．openMosix は Linux カーネルへのパッチとして提供される．対応している Linux カーネルのバージョンが古いため，比較的新しいハード

ウェアなどは対応していない場合もある[*1].

3.2.1 起動・停止・マイグレーション時間および負荷変動時の消費電力

電力制御機構の基礎データ収集のため，シャットダウンおよびマイグレーションに要する時間および負荷変動時の消費電力を測定した．

シャットダウンおよびマイグレーションに要する時間の計測実験では，10 回試行し，最大値，最小値，平均値をとった．また，消費電力の測定実験では，停止時，平常時 (CPU 使用率 0%)，CPU 使用率 100%のときの消費電力を 10 秒測定し，平均値をとった．使用した PC および計算ノードの性能を表 1 に示す．

シャットダウンおよびマイグレーションに要する時間を計測した結果を表 2 に，消費電力測定実験の結果を表 3 に示す．表 2 において，起動時間がシャットダウン時間より長くなった．できるだけ待機電力を抑えるためにシャットダウンにしたが，スリープなどの別の待機状態を実装して，起動時間を早くすることが必要である．また，マイグレーション時間は，長くて終了処理の約 9.2%であり，問題のないレベルだと思われる．表 3 において，計算ノードの停止時消費電力が PC のそれに比べて高いのは，PC がメーカー製であり，使用されている電源が計算ノードの AC アダプタに比べて，待機電力が低いものであるからだと考えられる．

表 1 PC および計算ノードの性能

	PC	計算ノード
CPU	Pentium Dual-Core E2180 (2.0 GHz)	Atom 330 (1.6 GHz)
RAM	1GB	2GB
HDD	80GB HDD	32GB SSD
VGA	オンボード	オンボード
電源	省スペース PC 用電源	AC アダプタ

CPU 使用率が消費電力にどれだけ寄与するのかを調べるために，表 3 に示した結果を基に作成したグラフを図 5 に示す．

図 5 より，本研究で使用した計算ノードでは，CPU 使用率の寄与は約 8%と非常に低いことが分かる．この結果より，Atom などの低消費電力 CPU では，負荷変動制御機構を実装するよりも ON/OFF 制御機構のほうが有効である．

[*1] 2.4 系のカーネルの場合，2.4.26 が最新であり，2.6 系のカーネルの場合，2.6.15 のみが RPM で提供されるため，RPM をサポートするディストリビューションでしか使えず，カーネルのカスタマイズも不可能なため，今回は 2.4.26 を使用した．

表 2 時間の測定 [sec]

	起動	シャットダウン	migration
最小値	60.6	17.3	1.15
最大値	61.9	18.6	1.89
平均値	61.3	17.8	1.34

表 3 消費電力の測定 [W]

	計算ノード	PC
停止時	3.78	1.56
平常時	28.6	61.9
CPU100%	31.2	82.2

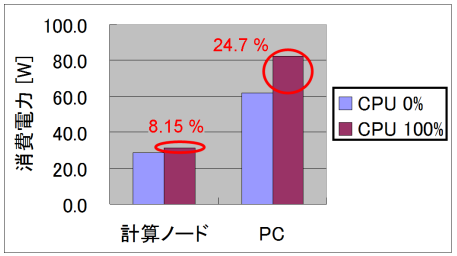

図 5 CPU 使用率の寄与

3.2.2 ON/OFF 制御時の消費電力

ON/OFF 制御時の消費電力測定において，構築した実験環境を図 6 に示す．本実験では，シャーシ内に 1 台の計算ノードが実装されていると仮定した．また，オフィスの電力として，ノートパソコンで擬似的に電力データをネットワークに流した．用意した電力データはオフィス内のパソコンが 20 台ある場合を想定した．

電力データは，オフィスでの利用を想定したものを 3 種類用意し，各 3 回測定を行い，平均をとった．電力データは，24 時間における 1 分ごとの推移 ($24 \times 60 = 1440$ 個のデータ) を考え，実験時間短縮のため，それを 1 時間で実験可能なデータに圧縮した．また，マイコン

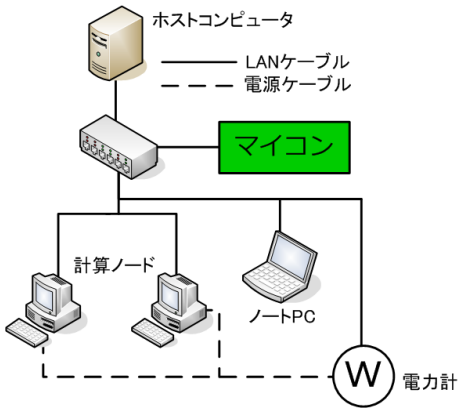

図 6 実験環境

は 10 秒に一度電力計のデータを取得するので，最終的には，1 種類につき，3600/10 = 360 個のデータを作成した．

データ作成の際，午前 9 時から徐々に PC の電源が入り，ある時刻でピークになり，午後 7 時に全 PC が停止するというモデルを考え，起動している PC の台数に，1 台あたりのワット数をかけたものをその時刻での消費電力とした．

用意した電力データを図 7 に示す．1 台あたりのワット数は 60W のものと，70W のものを作成した．表 3 に示した予備実験の結果より，PC の平常時消費電力が約 60W であり，ワープロソフト等を使うだけならば，消費電力は平常時とほとんど変わらないため，1 つめのデータとして 60W のデータを作成した．また，CPU 使用率 100%における消費電力が約 80W であったため，平常時との平均をとり，2 つめのデータとして 70W のデータを作成した．1 つめおよび 2 つめの電力データは，ステップ状に電力が変化するように作成した．3 つめのデータは，1 台あたりの消費電力は 60W だが，電力が線形に変化するようにした．

本実験では，PC の消費電力帯域として，3 つの帯域を設けた．1 つめの帯域は，600W 以下の帯域で，その帯域では，2 ノード稼働を起動する．2 つめの帯域は，601W 以上 900W 以下の帯域で，この帯域では，1 ノードを稼働する．3 つめの帯域は．901W 以上の帯域で，この帯域では，どのノードも起動しない．

実験結果を図 8 に示す．図 8 において，実験 1 よりも実験 2 のほうが計算ノードの消費電力は低い．これは，ステップの落差が実験 2 のほうが大きいため，早い段階で計算ノードの停止が行われたためであると考えられる．また，実験 1 よりも実験 3 のほうが計算ノードの消費電力が低い理由についても，実験 3 のほうが変化が滑らかであるので，早い段階で計算ノードの停止が行われたためであると考えられる．

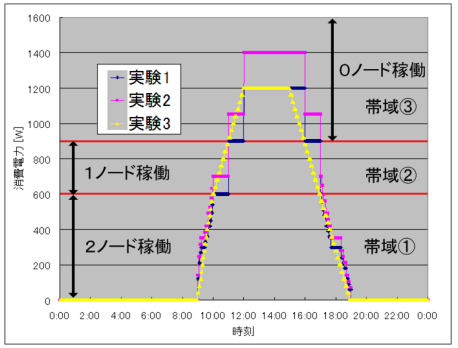

図 7 用意した電力データ

本実験では，計算ノードが 2 台，PC が 20 台と台数に差があるので，式 1 にて正規化消費電力 ($P_{normalizing}$) を求め，その結果を図 9 に示す．なお，式 1 において，P_{C_node} は計算ノードの消費電力,N_{C_node} は計算ノード台数，P_{PC} はオフィスで使用する PC の消費電力, N_{PC} はオフィスで使用する PC 台数を表す．

$$P_{normalizing} = \frac{P_{C\_node}}{N_{C\_node}} + \frac{P_{PC}}{N_{PC}} \quad (1)$$

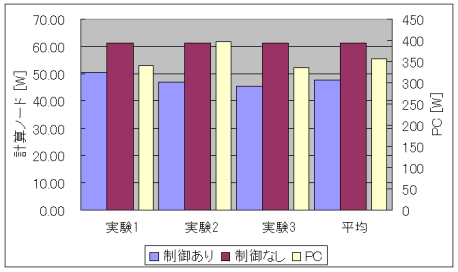

図 8 計算ノードおよび PC の平均消費電力

図 9 の削減率 (C_r) は式 2 にて計算している．なお，$P_{normalizing}(control)$ は，ノード実

Vol.2010-ARC-188 No.5
2010/3/1

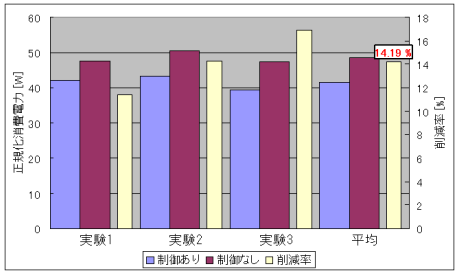

図 9 正規化した消費電力および削減率

行状態制御をしない場合の正規化消費電力であり，$P_{normalizing}(non-control)$ はノード実行状態制御をした場合の正規化消費電力である．

$$C_r = 100 - \frac{P_{normalizing}(control)}{P_{normalizing}(non-control)} \times 100[\%] \quad (2)$$

正規化を行い，削減率を計算した結果，消費電力により動作ノード数の制御を行った方が，制御を行わない場合に比べて，約 14%消費電力が削減された．

4. おわりに

エコ PC ヘテロクラスタ実行環境実現のためのシステムについて検討し，そのシステムで必要な実電力による計算ノード実行状態制御について検討した．エコ PC ヘテロクラスタ実行環境のためのテストベッドとして，低消費電力 CPU などを搭載したヘテロクラスタシステムについて述べ，その上で実行される電力制御機構や負荷分散スケジューラなどのソフトウェア構成について述べた．また，電力制御機構として，2 つの制御機構を検討した．１つは，電力を細かく制御する負荷変動制御機構であり，もう１つは電力を粗く制御する ON/OFF 制御機構である．

予備評価として，２つの電力制御方式を別々に行い，電力制御幅，消費電力削減率などを測定した．その結果，負荷変動制御方式では電力を 3W 刻みで制御できる可能性があり，ON/OFF 制御方式では消費電力が約 14%削減できるという結果を得た．

今後の課題として，２つの電力制御機構をうまく組み合わせたハイブリッドな電力制御機構を考える．また，負荷分散スケジューラを開発し，電力制御機構と組み合わせ，低消費電力かつ並列化効率の高い並列実行環境の構築を目指す．

参 考 文 献

1) 中島 浩, 中村 宏, 佐藤 三久, 朴 泰祐, 松岡 聡, 高橋 大介, 堀田 義彦, ”高性能計算のための低電力・高密度クラスタ MegaProto”, 情報処理学会論文誌コンピューティングシステム, Vol.46, No. SIG12 (ACS 11), pp.46–61,2005.
2) Warren, W., Weigle, E., Feng, W. :High-Density Computing : A 240-Node Beowulf in One Cube Meter. Super Computing. CD-ROM(2002).
3) Takayuki Imada, Mitsuhisa Sato, Yoshihiko Hotta and Hideaki Kimura、”Power Management of Distributed Web Servers by Controlling Server Power State and Traffic Prediction for QoS”, HPPAC 2008.
4) Kiyoshi Hayakawa, Tohru Sasaki, Hiroaki Umeda, Umpei Nagashima, ”Evaluations of Load Balancing Methods for Molecular Orbital Calculations on a Hetero-Cluster System”, 20th IASTED International conference Parallel and Distributed Computing and Systems，pp.285-290, 2008
5) 森川 浩明, 榎原 博之, 大西 克実, 中野 秀男,「仮想計算機を用いたジョブマイグレーションの PC グリッドへの適用」，情報処理学会研究報告，2009-MPS-19，pp.17-20, 2009.
6) 立薗真樹，中田秀基, 松岡聡,「仮想計算機を用いたグリッド上での MPI 実行環境」，先進的計算基盤システムシンポジウム SACSIS2006，pp.525-532, 2006.