エココンパクトヘテロクラスタ構想とその予備評価

エココンパクトヘテロクラスタ構想とその予備評価

早川　潔*，原田　信**

A Design of Eco Compact Hetero Cluster and the Preliminary Evaluations

Kiyoshi HAYAKAWA* and Makoto HARADA**

ABSTRACT

PC cluster systems made by general-purpose parts are good cost performance, low power and compact. Therefore, laboratories and small 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 controller which controls electric power consumption by watt-monitors. Using hetero cluster system, we can reduce electric power cost in the small laboratories and small companies. In order to control the node state, we propose 2 methods in the system, load balancing method and ON/OFF control method. Load balancing method allows us to control electric power finely, and ON/OFF control mechanism allows us to control electric power coarsely. In preliminary evaluations, we achieved approximately 3W power control range and 14% power reduction rate.

Key Words: cluster computing, heterogeneous computing, migration, low power consumption

1．　はじめに

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[2]．

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「エココンパクト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．

2010年　8月　20日 受理

＊　総合工学システム学科　電子情報コース (Dept of Industrial Systems Engineering: Electrical Engineering and Computer Science Course)

＊＊　パナソニック セミコンダクターシステムテクノ株式会社 (Panasonic Semiconductor Systems and Technology Co., Ltd)

2.　エココンパクトヘテロクラスタ構想

本構想の全体像を図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．

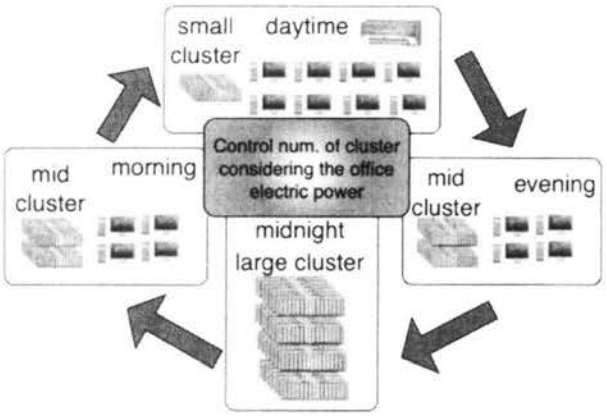

図1　エココンパクトヘテロクラスタ構想の全体像

2.1　システム構成

エココンパクト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．

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 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．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．

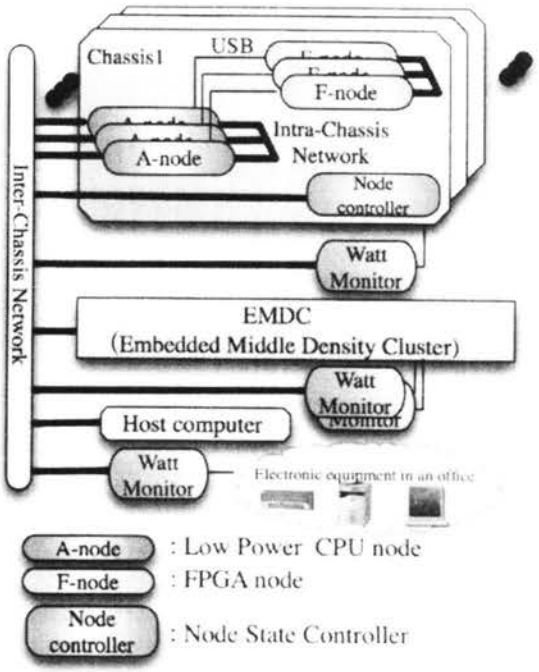

図2　エココンパクトヘテロクラスタ構想の全体像

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 マイコンで構成する．

2.2　FPGA によるアプリケーション実行

FPGA でアプリケーションを実行させるために，アプリケーションの開発時期・難易度に合わせ，3つの段階に分けて開発する(図3参照)．第1段階として，FPGA にプロセッサコアを載せ，アプリケーションを実行させる．この段階では，消費電力的な観点のみ有利である．プロセッサコアには，Open Cores[5]で配布しているプロセッサコア(OpenRISC 2000)を使用したり，ASIP Solutions[6]が開発した「Brownie」などを使用する．第2段階として，演算器とレジスタを数十器アレイ上に敷き詰める（タイルプロセッサ）．各演算器をトーラスで結合して，計算し

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5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，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．

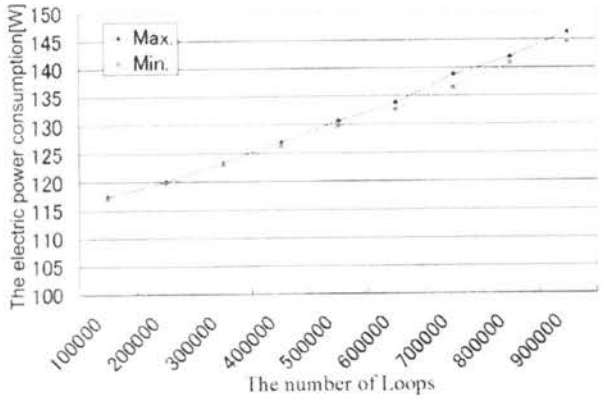

図5　スリープ命令の間に実行されるループ回転数と電力の関係

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．

3.3 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　10%以下)，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．

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1に，消費電力測定実験の結果を図6に示す．表1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．図6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　起動・停止・マイグレーションの時間の測定［sec］

	起動	シャットダウン	Migration
Max.	60.6	17.3	1.15
Min.	61.9	18.6	1.89
Ave.	61.3	17.8	1.34

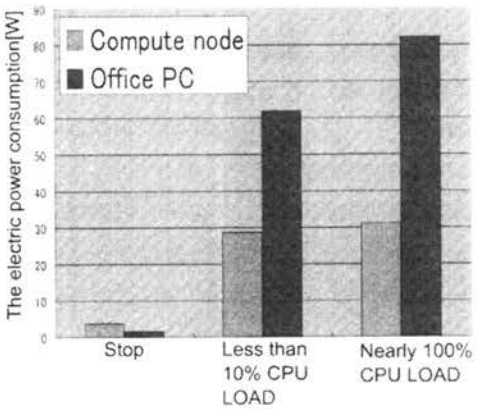

図6　CPU使用率と消費電力に関係

また，図6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

た結果をお互いに通信しあえるようにようする．そのために，各プロセッサには，演算器のほかに簡易ルーターを搭載する． 第3段階として，完全にハードウェア化する．

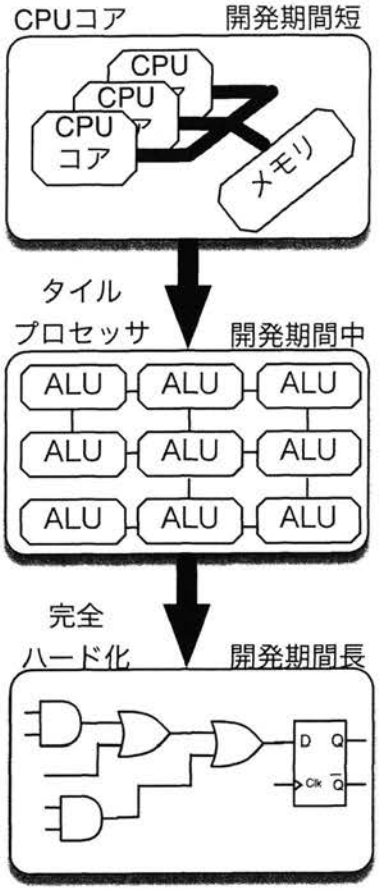

図3　FPGA の内部構成

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くしたりすることにより，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．

2.4　ソフトウェア構成

エココンパクト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 4 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，このマイグレーションに関する研究が盛んに行われている[7,8]． 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．

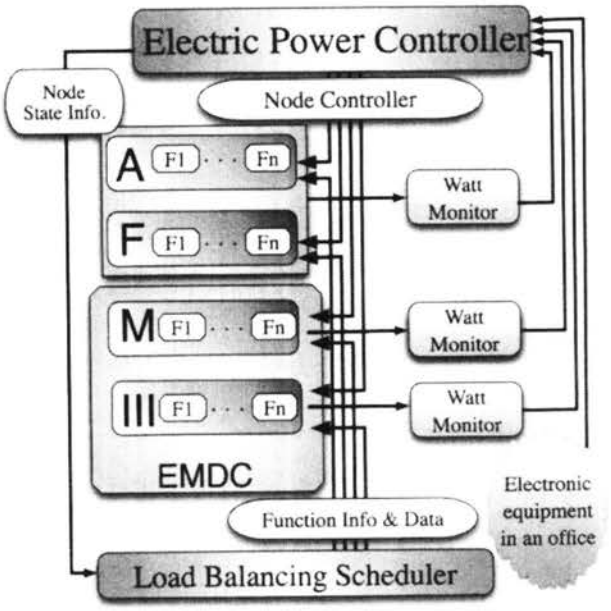

図4　EPHC のソフトウェア構成

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．

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.

3.4 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7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(24×60=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.

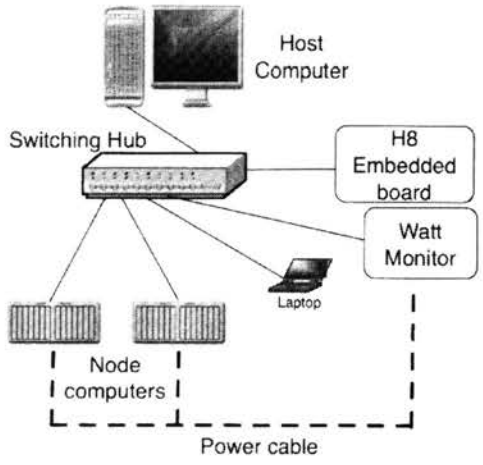

図7　ON/OFF制御機構における実験環境

用意した電力データを図8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．図6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.

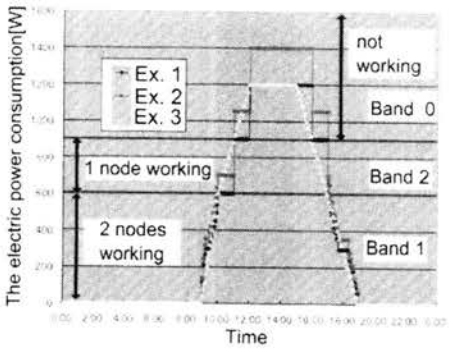

図8　オフィス全体の電力を想定した電力データ

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9に示す．図9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のほうの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.

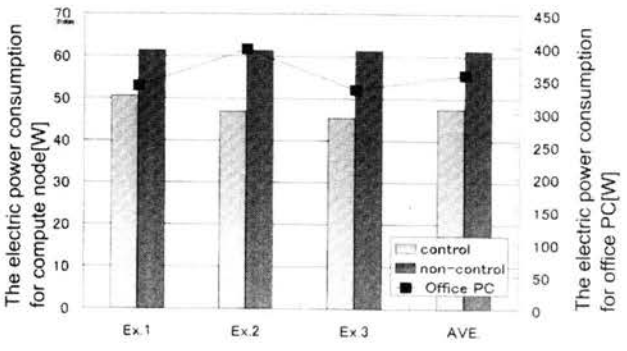

図9　計算ノードおよびPCの平均消費電力

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，(1)式にて正規化消費電力$P_{normalizing}$を求めた.その結果を図10に示す.なお，(1)式において，P_{Cnode}は計算ノードの消費電力，N_{Cnode}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．また，図9の削減率C_rは(2)式にて計算している．

$$P_{normalizing} = \frac{P_{Cnode}}{N_{Cnode}} + \frac{P_{PC}}{N_{PC}} \qquad (1)$$

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

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

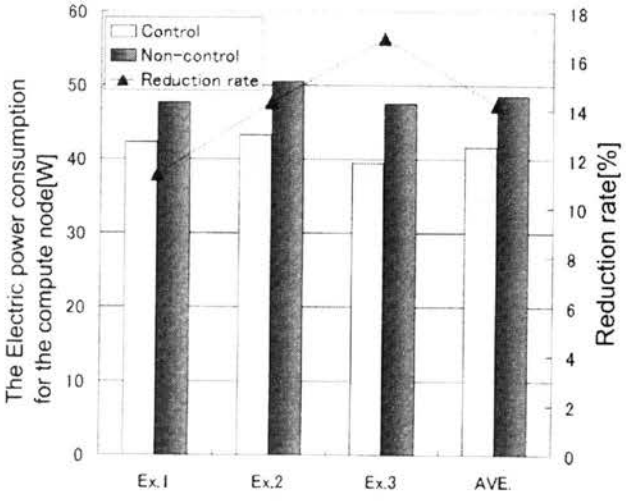

図１０　正規化した消費電力および削減率

4.　おわりに

エココンパクト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．エココンパクト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．

本研究の一部は，平成21年度大阪府立工業高等専門学校校長奨励金により行われた．

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 (ACS11)，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] Open Cores : http://opencores.org/

[6] ASIP Solutions : http://www.asip-solutions.com/index.html

[7] 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.

[8] 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.