



## 低消費電力コンパクトクラスタの開発

メタデータ	言語: jpn 出版者: 公開日: 2013-12-20 キーワード (Ja): キーワード (En): 作成者: 早川, 潔 メールアドレス: 所属:
URL	<a href="https://doi.org/10.24729/00007612">https://doi.org/10.24729/00007612</a>

# 低消費電力コンパクトクラスタの開発

早川 潔\*

Development of Low Power Compact Cluster

Kiyoshi HAYAKAWA\*

## ABSTRACT

Recently, different PCs are launched from pillar to post as rapid progress of CPU. In the future, cluster systems which have different types of compute resource, such as GPU, FPGA and so on, may be launched. Under the circumstances, we are developing a low power compact embedded cluster (called EMDC) system for heterogeneous cluster testbed. We construct virtual tree network and each node communicates with it. In this paper, we proposed a communication framework for virtual tree network on EMDC system, and developed a communication library (H/L comm) based on it. Using this library, we implemented a parallel Fock matrix construction program on EMDC system. In evaluations, we executed the parallel Fock matrix construction programs with 1 network and 2 networks.

Key Words: computer cluster, computer network, communication library, low power computing

## 1 はじめに

CPUの低消費電力化にともない、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][2]。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。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、より高速な並列処理を可能にしている。

しかし、近年、インテルなどのCPUベンダーは、CPUの単体性能を上げる技術開発よりもCPUの消費電力を抑えるような技術開発に重点をおいている。その原因の1つは、性能とともに消費電力が増大し、ファンなどのプロセッサの熱を逃がす装置が巨大化し、それによりコンパクトな筐体設計が難しくなるからである。クラスタシステムでは、CPUが複数集まっているので、熱放出を考えた筐体設計はより難しくなる。1つのCPUがほんの数%消費電力が上がっても、クラスタシステム全体では数十~数百倍に跳ね上がる。消費電力増大によるファンなどの熱放出装置の巨大化によって、筐体はより巨大になっていく。PCクラスタシステムはある程度小規模な企業・研究グループで使用されることが多いので、できるだけコンパクトで低消費電力なクラスタシステムが望

まれる。

低消費電力化の手段として、CPUで行うアプリケーションのソフトウェア的な処理をFPGA(ハードウェアを書き換え可能なデバイス)で処理することが考えられる。FPGAの消費電力は高々数Wであり、Pentium4の100W近い消費電力と比較して圧倒的有利である。FPGAに搭載できるCPUなどを使用すれば、ソフトウェアの変更も少しですむ。また、ソフトウェアをハードウェア化するということも考えられる。多少手間がかかると思われがちだが、近年のハードウェア記述言語やハードウェア設計ツールの発展により、比較的容易にハードウェア化が可能になってきている。さらに、設計したハードウェア(IPコア)を発売およびインターネットで公開している所も多々ある[9]。よって、低消費電力という観点とともにIPコアのインターネット公開という観点からもアプリケーションをFPGAで実行するという事は重要と思われる。

クラスタを導入した後、ある程度の期間が経つと、故障によって、CPUなどの機器を交換する必要がある。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よいと思われる。また、新たにプロセッサ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。しかし、CPUの性能にアンバランスが生じ、さらに、FPGAが計算資源に加わると、異なる計算資源を効率よく稼働させる仕組みが必要となる。

また、クラスタシステムにおける処理の高速化におい

2008年4月9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)

て、コンピュータノード間通信が重要であり、コンパクト実装を生かした通信が必要と思われる。クラスタをコンパクトに構成するために、1つのユニットに複数のコンピュータノードを搭載し、そのユニットを筐体（ラック）に数十台格納する場合が多い。また、そのネットワーク構成は、100Base-TX および Gigabit Ethernet が採用されている。多くのクラスタでは、ネットワークはユニット内外に関係なく接続されており、ユニット内におけるコンパクト実装の恩恵（短いケーブルで比較的容易に接続できることなど）を享受していないと思われる。本研究室では、上記の問題を解決すべく、低消費電力コンパクトクラスタを開発中である（図1参照）。

本稿では、低消費電力コンパクトクラスタにおけるネットワーク枠組みを示し、PC (Dos/V マザーボード) をベースとしたコンパクトクラスタシステムのプロトタイプ上で、そのネットワークの通信性能を評価する。また、各筐体に FPGA を搭載し、アプリケーションを FPGA 上で行う方法を検討する。

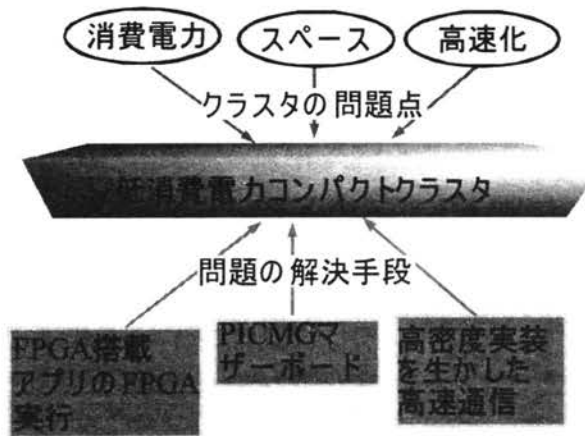


図 1. 低消費電力コンパクトクラスタ構想

## 2 EMDC クラスタ

図 2 に EMDC システムの構成を示す。EMDC クラスタは低消費電力・高密度を目指しているが、それと同時に長期間運用可能なクラスタも目指している。前述のとおり、近年の CPU の開発サイクルは急激に速くなり、2,3 年もすれば入手が困難になることが多い。そのような状況下で、故障 CPU やマザーボードの修復などが難しい。

そこで、EMDC システムでは、FA などで使用される組み込み機器のマザーボードを採用し、そのマザーボードにインテルが供給している Embedded CPU(長期間入手可能な CPU)を搭載することで長期間運用を可能にする。組み込み機器のマザーボードにも Pentium M などのノート PC で使用される低消費電力型 CPU が搭載できる製品もでてきており、より低消費電力なクラスタが実現可能で

ある。

### 2.1 シャーシ内部の構成

本システムでは、PentiumM(2.0GHz)搭載のノードが 9 台、PentiumIII 搭載のノード 27 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 low 動作周波数ではあるが低消費電力である CPU を利用して、コンパクトなクラスタを目指している。

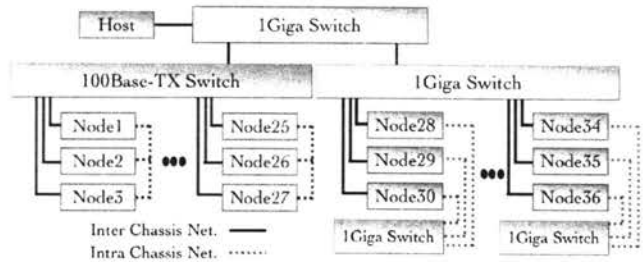


図 2. EMDC クラスタ

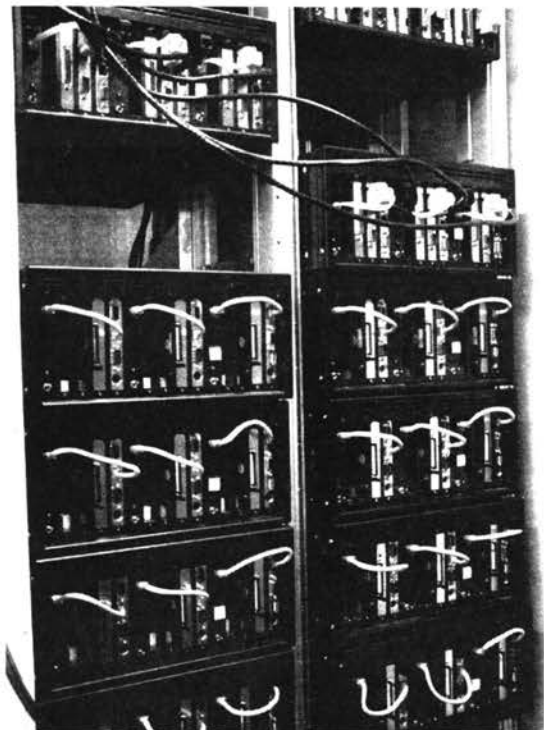


図 3. EMDC クラスタの外観

今後、FPGA などを利用してアプリケーションをハードウェア化し、より高速に処理できるシステムにすることを考えている。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。

外部記憶装置は、コンパクトフラッシュメモリ (PentiumM・PentiumIII ノードともに 1Gbyte) を採用した。一般的に利用されているハードディスクドライブの場合、ディスクが高速回転しているため軸受が物理的に消耗する可能性がある。また、比較的高密度でHDDを実装している場合、共振などの振動がハードディスクの読取機構に悪影響を及ぼしアクセス速度低下とハードディスクの劣化進行を促進する可能性がある。これらの問題を解決するために、CF-IDE 変換ボードを使用して、コンパクトフラッシュメモリを IDE 機器と認識させ、OS をインストールした。

## 2.2 ネットワーク構成

ネットワーク構成は、前述したとおり、Intra-Chassis Network および Inter-Chassis Network で構成されている。Intra-Chassis Network は、ギガビットイーサネットボードを使用して構築されている。PentiumIII ノードの Inter-Chassis Network は、100Base-TX の Ethernet で、PentiumM ノードの Inter-Chassis Network は、Gigabit Ethernet で構築されている。

### 2.2.1 PentiumIII シャーシのネットワーク

PentiumIII のネットワーク構成を図 4 に示す。マザーボード内蔵のネットワークインターフェース (以下 NIC と略す) の他に、PCI スロットに NIC を挿入する。筐体の真ん中に実装されているノード (Center ノード) には、PCI に 2 つの NIC ボードを搭載し、筐体の左側のノード (Left ノード) および筐体の右側のノード (Right ノード) に直接接続されている。従って、Left および Right ノードは、PCI スロットに 1 つの NIC を挿入し、Center ノードと接続されていることになる。

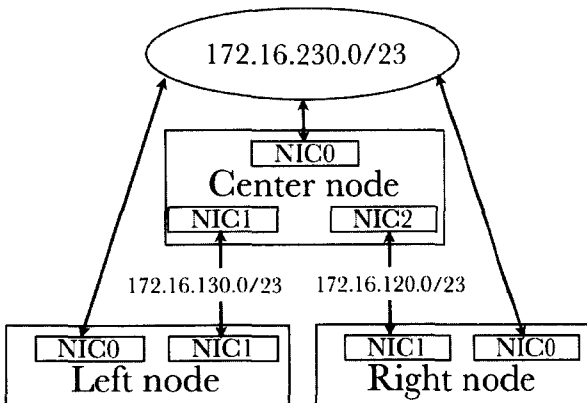


図 4 PentiumIII シャーシのネットワーク

3 種類のネットワークを用いて、筐体内外の通信を行う。筐体外のネットワークで 1 つ使用し、残り 2 つは筐体内のネットワークで使用する。この場合、Left ノード

と Right ノードは直接通信できないが、後述する仮想 Tree ネットワークでは、Left-Right 間通信は、(直接的には)行われないので、問題ない。

### 2.2.2 PentiumM シャーシのネットワーク

PentiumM のネットワーク構成を図 5 に示す。マザーボード内蔵の 2 つの NIC を利用して、筐体内外のネットワークを構築する。そのため、各筐体に対して、ギガビットのスイッチを用意した。

PentiumM のネットワークの場合、PentiumIII のそれと異なり、2 つのネットワークを筐体内外の通信を行う。この場合、Left-Right 間通信は可能だが、仮想 Tree ネットワークでは、行われぬ。

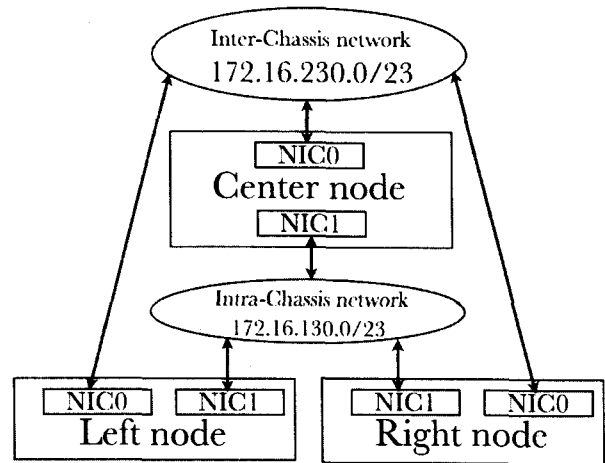


図 5 PentiumM シャーシのネットワーク

## 3 仮想 Tree ネットワーク

EMDC システムにおけるネットワークの枠組みとして、仮想 Tree ネットワークを考える (図 6 参照)。Intra/Inter-Chassis Network をうまく利用することにより、仮想的に Tree ネットワークを構成する。シャーシ内の通信を Intra-Chassis Network で密に行うことができるので、1 シャーシで 3 つのネットワークポート (Inter-Chassis Network のポート) があると仮想的に考え、それらのポートを使って、Binary Tree を構成する。

### 3.1 仮想 Tree ネットワークにおけるノード間結合

本システムでは、シャーシ間接続として、シャーシ内の中央に位置するノード (Center ノード) が上位レベルのノードと接続し、シャーシ内の右側および左側に位置するノード (Right ノードおよび Left ノード) が下位レベルのノードと接続する (ホストコンピュータをルートノ

ード(レベル0)とする). つまり, レベルnとレベル(n+1) (n=0,2,4,...:2の倍数)のノードはInter-Chassis Networkで通信する.

シャーシ内接続では, Center ノードが Right および Left ノードより 1 つ上のレベルとして定義し, それらの通信を Intra-Chassis Network で通信する. つまり, レベル(n+1)とレベル(n+2)のノードは Intra-Chassis Network で通信する.

### 3.2 仮想 Tree ネットワークのレベル数

レベル数は, 本システムで実行するアプリケーションによるが, レベルを多く取ればとるほど, 通信レイテンシが増大するので, できるだけレベル数を少なくすることが望まれる. ただし, レベル数を少なくすると, ホストコンピュータの負荷が増加するので, それらのトレードオフを考えレベル数を決める.

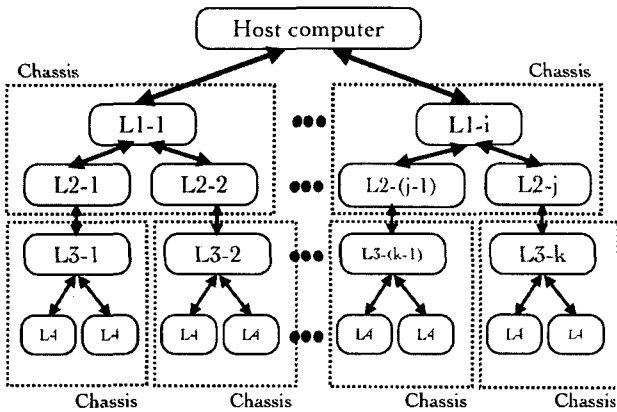


図 6 仮想 Tree ネットワーク (レベル 4 の場合)

## 4 通信フレームワーク

仮想 Tree ネットワーク上での通信フレームワークを提案する. 本フレームワークは後述するアプリケーションフレームワークの下で適用される.

### 4.1 アプリケーションフレームワーク

アプリケーションフレームワークの例を図 7 に示す. アプリケーションフレームワークは, Appli\_Control 処理部, Communication Library 処理部, Communication Bridge 処理部, および Compute Program 処理部で構成される.

Appli\_Control 処理部では, 計算ノードの計算データを生成, 各計算ノードに分配, および計算ノードが処理した計算結果の収集処理などアプリケーション全体のコントロールが行われる. Communication Library 処理部は後述する通信ライブラリ (H\_comm および L\_comm)の処理を行う. Communication Bridge 処理部は, 通信のリンクレイヤ層が異なる場合に, その間のインターフェースの役割を行い, コレクティブ通信を制御する役割も行う.

Compute Program 処理部は, Appli\_Control 処理部から送られてきたデータを使用して計算を行う.

実行モデルとしては, master-worker 型の実行モデルとする. Appli\_Control 処理部が Master になり, Compute Program 処理部にコマンドを送ることにより, Worker として動作させる.

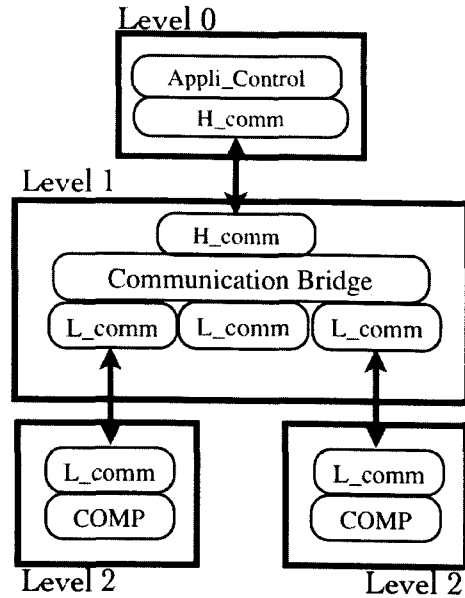


図 7 アプリケーションフレームワーク例

### 4.2 通信フレームワーク

通信フレームワークの構成例を図 8 に示す.

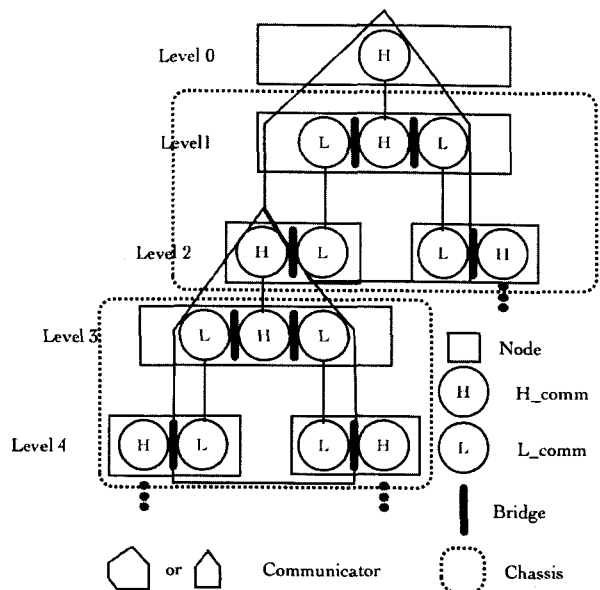


図 8 通信フレームワーク (Level4 の場合)

3 つの連続するレベルに属しているノード集合が, 通

信を形成する1つの集合 (MPI での Communicator) として構成される。それらの集合をレベル  $3n$  からレベル  $3n+2$  ( $n=0,2,4,\dots$  の倍数) に属しているノード集合, レベル  $3n+2$  からレベル  $3n+4$  に属しているノード集合, およびレベル  $3n+4$  からレベル  $3(n+1)$  に属しているノード集合というように, ある集合の最下位レベルのノードがその下のレベルのノード集合の最上位レベルのノードと重なるように集合を構成する。重なったレベルに Communication Bridge を挿入し, 上位のレベルのノード集合と下位レベルのノード集合を結合させる。このことにより, より深いレベルのノード間通信を可能にする。例えば, 図 8 において, Level 0 から Level 2 までの3つのノードの集まりを1つのノード集合, Level 2 から Level 4 までのノードの集まりを1つのノード集合として考え, Level 2 の L\_comm と H\_comm の間に Communication Bridge を挿入することにより, Level 0 から Level 4 までの通信を可能にする。

### 4.3 通信ライブラリ

通信ライブラリのレイヤ構成を図 9 に示す。通信ライブラリは, Intra Chassis Network 側のレイヤおよび Inter Chassis Network のレイヤ, それを結ぶコレクティブ通信レイヤで構成される。Intra および Inter Chassis Network の Phy. および Link レイヤは, Ethernet とし, それに対応する H\_comm および L\_comm を実装した。

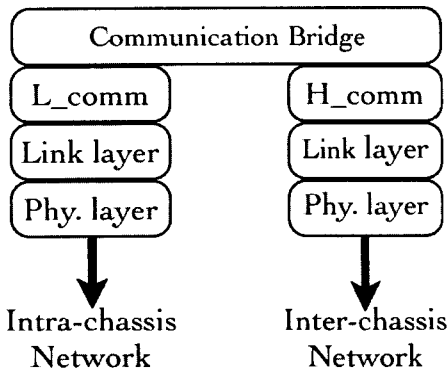


図 9 通信ライブラリの構成図

H\_comm および L\_comm では, send/recieve などの atomic な命令を実装し, それを利用して Communication Bridge レイヤが通信を行う。H\_comm および L\_comm は, 基本的に, 1つ上 (または1つ下) のレベルへのノード間通信のみを実現する。1つより上 (または下) のレベルへのノード間通信は, Communication Bridge レイヤを介して行われる。コレクティブ通信も Communication Bridge レイヤが担当する。

## 5 FPGA 組み込み型クラスタシステム構想

本章では, FPGA を組み込んだクラスタシステムを検討する。アプリケーションのハードウェア化が比較的容易になってきており, また FPGA を利用してアプリケーションをハードウェアで実行することにより消費電力削減につながる。さらに, コレクティブ通信の制御および負荷分散の制御をすることにより, 通信に余分な計算能力を使用せず, クラスタの CPU の計算パワーを十分に引き出せる。従って, FPGA および組み込み CPU を搭載したボードをクラスタに付加し, より効率のよい並列計算を目指す。

### 5.1 FPGA 組み込み型クラスタシステムのネットワーク接続

図 10 に FPGA 組み込み型クラスタシステムの接続構成を示す。図中, 「Control n」 ( $n$  はシャーシ番号) が FPGA および組み込み CPU を搭載したボード (以後, 「通信 & FPGA ボード」と略す) である。この通信 & FPGA ボードは, Intra Chassis Network をスイッチング HUB で結合しているシャーシに限り搭載する。

本ボードは, Inter および Intra Chassis Network に接続され, コレクティブ通信および FPGA の制御を行う。コレクティブ通信の場合, あるプロトコルに従って通信を行うので, そのプロトコルのための通信 (手順を記した小パケットの通信) を行い, 実際のデータ転送は, 各ノードに接続されているネットワークを介して行われる。その際, 通信 & FPGA ボードの指示に従って行う。

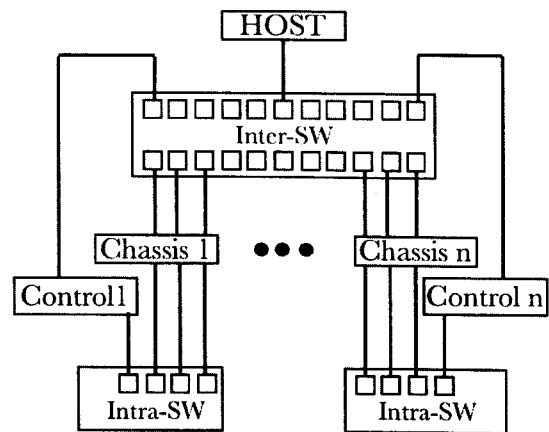


図 10 通信 & FPGA 制御ボード付加時の接続図

### 5.2 通信 & FPGA 制御ボードの内部構成

図 11 に通信 & FPGA 制御ボードの内部構成を示す。本ボードは, NIC, SoC (System on a Chip), NIC には, RTL8019 を搭載したボードを使用し, SoC には, H8 を CPU としての SoC を使用する予定である。

FPGA の内部には、PCI-Wishboneブリッジ[9] (PCIバスに流れたデータを Wishbone バスに流したり、その逆を行ったりする制御ブロック)、メモリ、通信コントローラ、およびアプリケーション実行ユニットが搭載される。

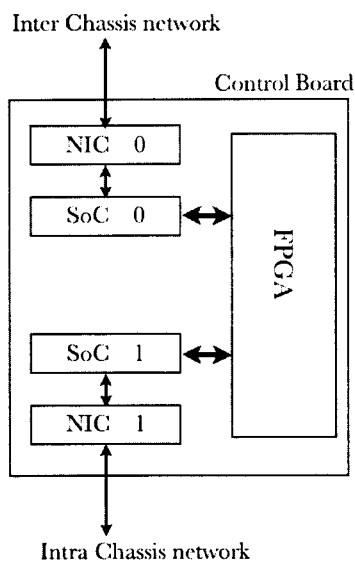


図 11 通信&FPGA ボードの内部構成

アプリケーション実行ユニットを、1) どんなアプリケーションでも使用できる共通なブロック、2) アプリケーションによってハードウェア構成が異なるブロック、に分ける。共通なブロックには、アプリケーションの回路情報をダウンロードする回路、Wishbone バスにアクセスする回路および隣接コンピュータノード間に搭載されている FPGA とのネットワークインターフェース回路がある。アプリケーションによって異なるブロックは、3つの実行方法を考えている。

1つ目として、FPGA にプロセッサコアを載せて、その上でアプリケーションを実行させる。この段階では、アプリケーション実行速度の高速化は期待できないが、消費電力的な観点から見ると、消費電力の高い Pentiumで行うよりかは低消費電力で行える FPGA で行ったほうが有利であると考えている。FPGA で使用するプロセッサとして、アルテラ社が販売している Nios II プロセッサを採用する。

2つ目として、演算器とレジスタを数十器アレイ上に敷き詰め (一種のタイルプロセッサ)、その上でアプリケーションを実行する。ここでは、アプリケーションを演算器レベルのタスクに分け、各演算器にスケジューリングするという方法をとる。スケジューラなどは、以前行っていたタスクスケジューラを一部変更して使用する。

3つ目として、アプリケーションを完全にハードウェア化する。この段階は、前述した段階より多少手間がか

かるが、よく使われる計算 (行列計算、求積法など) をハードウェアマクロ化して、再利用するなどして工夫する。

## 6 性能評価

性能評価として、分子軌道計算ソフト GAMESS を EMDC 用に並列化したプログラムを使用した。本プログラムを1つのネットワークの場合と2つのネットワークの場合の並列化効率を比較した。性能評価で使用した分子は、C 末端を OH キャップしたグリシンの5量体である。

本稿では、EMDC システムの一部を使用して計算した予備評価、および EMDC システム全体を使用した全体評価を行った。予備評価では、PentiumIII ノードのみで構成したクラスタの実行時間と PentiumM のみで構成したクラスタとの性能比較を行い、全体評価では、PentiumIII および PentiumM で構成されたヘテロクラスタ環境での評価を行った。

本評価では、36 台のノード (PentiumIII ノード 27 台、PentiumM ノード 9 台) を使用して行った。各ノードには、LinuxOS(version 2.6)が搭載されている。

### 6.1 PentiumIII ノードと PentiumM ノードとの性能比較

予備評価として、EMDC システムを PentiumIII ノードと PentiumM ノードに分け、それぞれで分子軌道計算を実行させた。実行結果を図 12 に示す。図 12 において、(Static) は静的負荷分散における分子軌道計算全体の実行時間であり、(dynamic) は動的負荷分散における分子軌道計算全体の実行時間である。静的負荷分散とは、実行前に各計算ノードの計算量を決め、実行させる方式であり、動的負荷分散は、全体の計算を細かなタスクに分け、実行中に、そのタスクを各計算ノードにスケジューリングする方式である。

PentiumM ノードの処理速度は、PentiumIII に比べて、静的負荷分散では 3.05 倍、動的負荷分散では 2.83 倍高速であった。したがって、この数値を用いて PentiumM の性能を PentiumIII の台数に換算し、並列化効率を算出する。

この数値は、動作周波数の差とほぼ一致する。分子軌道計算では、メモリアクセスが少なく、浮動小数点処理部の性能が動作時間に大きく影響するので、PentiumM ノードに実装されているデュアルチャネルメモリなどの高速化技術が実行時間に影響を与えられなかったと思われる。

PentiumM のシャーシ (2 ノード実行) の消費電力 (静的負荷分散) は 127W であり、PentiumIII のそれは 103W であった。積算電力 (消費電力×計算時間) で比較する

と、PentiumM シャーシは PentiumIII シャーシより、4.91 倍よい結果となった。また、Pentium4 (3.0GHz : Northwood, Hyper-threading, FSB800MHz)搭載のパソコンで同様の計算を行い、積算電力を計測した。その場合、PentiumM 筐体は Pentium4 パソコンより 1.96 倍よい結果となった。

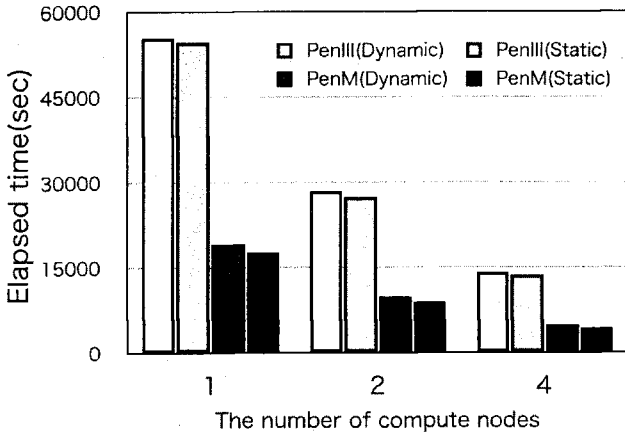


図 12 PentiumIII ノードおよび PentiumM ノードの性能比較

### 6.2 EMDC システムの性能評価

EMDC システムでの性能評価として、Intra-Chassis Network と Inter-Chassis Network を合わせた通信性能を評価する。1つのネットワーク (Inter-Chassis Network) のみで実行した場合と 2つのネットワークで実行した場合との間で並列化効率を比較した。

図 13 に静的負荷分散での並列化効率を示す。1台から 27 台まで、PentiumIII ノードのみで計算し、計算ノードが 30 台目から PentiumIII ノードおよび PentiumM ノードのヘテロクラスタ構成で計測した。

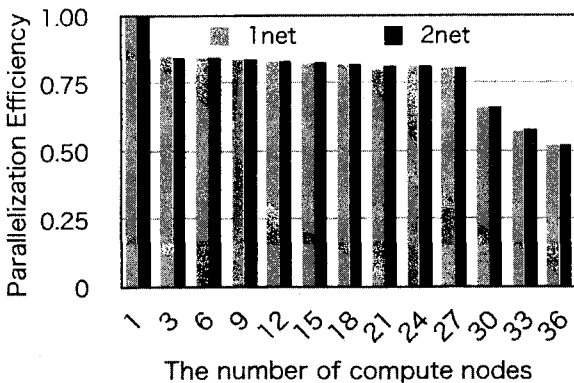


図 13 静的負荷分散での並列化効率

静的負荷分散では、1 ネットワークと 2 ネットワークであまり変わらなかった。この原因として、実行時間全体におけるデータ通信時間の割合が少なく、データ転送

回数が動的負荷分散に比べ少ないことが挙げられる。

図 14 に動的負荷分散での並列化効率を示す。計算ノードが 1 台から 6 台までは、両ネットワークともあまり変わらないが、9 台以降は、2 ネットワークの並列化効率がよく、最高で 10.6% 並列化効率が改善された。

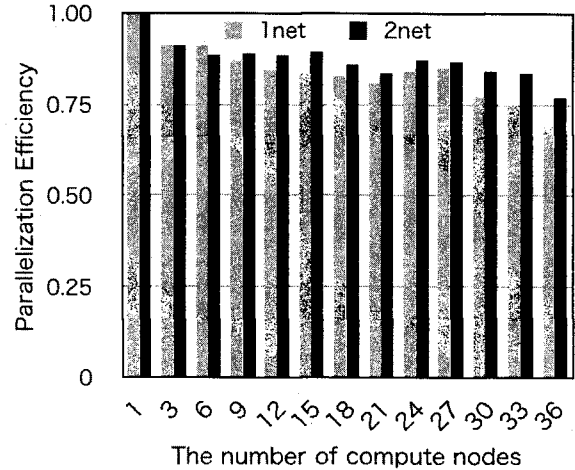


図 14 動的負荷分散での並列化効率

### 7 おわりに

低消費電力コンパクトクラスタ EMDC について、そのハードウェア構成、ネットワークフレームワーク、アプリケーション、および通信ライブラリを述べた。さらに、FPGA の EMDC への実装方法を検討した。性能評価では、EMDC のネットワーク性能を評価するために、分子軌道計算を 1 ネットワークと 2 ネットワークで計算し、比較した。2 ネットワーク構成で、並列化効率が最高で 10.6% 改善した。

今後の課題としては、分子軌道計算の改善・性能評価およびそれらをもとに他のハイパフォーマンスアプリケーションでの開発スタイルを考える予定である。また、本稿では、2つの負荷分散方式で計算したが、負荷分散方式の改良をアプリケーションに施して性能評価したいと考えている。

### 謝辞

この研究の一部は、平成 18 年度科学研究補助金 (基盤研究 C: 課題番号 18500044) 「低消費電力コンパクトクラスタの研究」によって行われた。分子軌道計算における並列化に関して、産業技術総合研究所の梅田宏明氏、長嶋雲兵氏、アプリオリマイクロシステムズの佐々木徹氏にご助言をいただいたことに感謝の意を表す。



## 参考文献

- [1] M. Warren, E. Weigle, W. Feng, "High-Density Computing : A 240-Node Beowulf in One Cube Meter", *Super Computing 2002*, Nov. 2002.
- [2] 中島, 中村, 佐藤, 朴, 松岡, 高橋, 堀田, "高性能計算のための低電力・高密度クラスター 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, Aug. 2005.
- [3] Kiyoshi Hayakawa, "SCCB CLUSTER SYSTEM-SYNCHRONIZATION AND PSEUDO GLOBAL CLOCK COUNTER SYSTEM-", *23rd IASTED International Conference on PDCN2005*, pp.216-221, Feb. 2005.
- [4] 堀田義彦, 佐藤三久, 朴泰祐, 高橋大介, 中島佳宏, 高橋睦史, 中村宏, "プロセッサの消費電力測定と低消費電力プロセッサによるクラスターの検討" *先進的計算基盤システムシンポジウム SACSIS2004*, pp.19-26, Mar. 2004.
- [5] 住元真司, 成瀬彰, 久門耕一, 細江広治, 清水俊幸, "PM/InfiniBand を用いた大規模 PC クラスタ向け高性能通信機構の設計", *先進的計算基盤システムシンポジウム SACSIS2004*, pp.373-380, Mar. 2004.
- [6] Steve Sistare, Rolf vande Vaart, Eugene Loh, "Optimization of MPI Collectives on Clusters of Large-Scale SMP's", *Super Computing'99*, CD-ROM (1999).
- [7] Sathish S. Vadhiyar, Graham E. Fagg, Jack Dongarra, "Automatically tuned Collective Communications", *Super Computing 2000*, CD-ROM (2000).
- [8] S. Shang, K. Hwang, "Distributed Hardware Barrier Synchronization for Scalable Multiprocessor Clusters", *IEEE, Trans, Parallel Distrib. Syst.*, vol.6, pp.591-605, June. 1995.
- [9] OPENCORES.ORG (<http://www.opencores.org>)
- [10] 堀田義彦, 佐藤三久, 木村英明, 松岡聡, 朴泰祐, 高橋大輔, "PC クラスタにおける電力実行プロファイル情報をを用いた DVS 制御による電力性能の最適化", *情報処理学会研究報告*, 2006-HPC-105, pp.139-144, Mar. 2006.
- [11] 佐々木 徹, 村上 和彰 : 科学計算専用計算機のプラットフォームシステム, *日本コンピュータ化学会*, Vol.4 No.4, pp.139-145, 2005.
- [12] 梅田 宏明, 稲富 雄一, 本田 宏明, 長嶋 雲兵 "分子軌道計算専用計算機のためのフォック行列並列計算アルゴリズムの開発", *日本コンピュータ化学会*, Vol.4 No.4, pp.179-187, 2005.
- [13] Kiyoshi Hayakawa, Thoru Sasaki, Hiroaki Umeda, and Umpei Nagashima, "A Communication Method for Molecular Orbital Calculations on a Compact Embedded Cluster", *Asian Simulation Conference*, pp.357-361, 2006.
- [14] Helgaker, T., Jorgensen, P., Olsen, J., : *Molecular Electronic-Structure Theory*, Wiley, 2000.
- [15] Schmidt, M., Baldrige, K., Boatz, J., Elbert, S., Gordon, M., Jensen, J., Koseki, S., Matsunaga, N., Nguyen, A., Su, S., Windus, T., Dupuis, M., Montgomery, J. "General atomic and molecular electronic structure system" *Journal of Computational Chemistry* Vol. 14, Issue 11, pp.1347-1363, 1993.
- [16] 林 徹生, 本田 宏明, 稲富 雄一, 井上 弘士, 村上 和彰, "Cell プロセッサへの分子軌道法プログラムの実装と評価", *情報処理学会研究報告*, 2006-HPC-103, pp.103-108, 2006.
- [17] Kiyoshi Hayakawa, Thoru Sasaki, Hiroaki Umeda, and Umpei Nagashima, "MOLECULAR ORBITAL CALCULATIONS ON EMBEDDED MIDDLE DENSITY CLUSTER SYSTEM", the 19<sup>th</sup> IASTED International Conference Parallel and Distributed Computing and Systems, pp.1-6, 2007.