社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－ARC－159 (22)
2004／7／31

高密度実装クラスタにおける同期・通信処理方式の検討

早川 潔　　岩根 雅彦

近年，消費電力および実装スペースの低減を目的としたクラスタ実装方式が注目を浴びている．低消費電力プロセッサを使ことにより１電源ユニットに複数のマザーボードが搭載可能になっており，ユニット内のマザーボード間は，より密にデータ転送ができる結合網が実装可能である．本稿では，高密度実装クラスタにおける同期・通信処理方式を検討した．本方式におけるネットワークのフレームワークについて説明し，そのフレームワークに基づいた実験用クラスタ上での同期・通信処理方式の実装方法について報告する．

Synchronization and Communication Method for the High-density cluster systems

Kiyoshi Hayakawa, and Masahiko Iwane

Recently, the High-density cluster systems which aim to reduce power consumption and implementation space have been proposed. By using low power processors, many processors are implemented into a unit (involve one power unit). So it is easy for the mother board embedding the processor to connect each other with a tightly coupled network. We describe a synchronization and communication processing method for the High-density cluster. We refer to a network flame work for the method and implementation into a experimental High-density cluster system.

1 はじめに

近年，汎用部品で構成されたBeowulf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Beowulf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，プロセッサ性能が良くなればなるほど電力消費量が増大するとともに，ファンなどのプロセッサの熱を逃がす装置が必要となり実装スペースが増大する．よって，低消費電力プロセッサを利用し，クラスタの高密度実装に関する研究が盛んに行われている [1][2][3].

一般的に，高密度実装クラスタでは，１つのユニットに複数のプロセッサが搭載されることが多い．低消費電力高密度実装Beowulfクラスタである「Green Destiny」[2] は，1 ユニットに 24 台ものプロセッサが搭載されている．また，そのネットワーク構成は，100Base/TX および Gigabit Ethrenet が採用されている．各プロセッサは 100Base/TX の Switch で結合され，さらにその Switch が Gigabit Ethrenet Switch に結合されている．本研究室において構築したSCCB[1] [7] クラスタも１つのユニットに３台のプロセッサが実装されており，各ノードは100Base/TXとSync-Commネットワークの２つのネットワークで接続されている．

高密度実装クラスタにおけるネットワークは，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高密度実装の恩恵（短いケーブルで接続できることやノイズの影響を受けにくいことなど）を享受していないと思われる．

そこで，本稿では，高密度実装クラスタシステム上でのネットワークフレームワークを示し，PCをベースした実験用高密度実装クラスタシステム（SCCB Cluster System）にそのフレームワーク当てはめた場合の同期・通信処理方式を検討する．本処理方式で

九州工業大学　工学部 電気工学科
Department of Electronic and Computer engineering
Kyushu Institute of Technology

[1] Sync-Comm Controller Beowulf

は, 高密度実装クラスタシステムのネットワークをユニット内ネットワークとユニット間ネットワークに分ける. ユニット内にあるノードにはユニット内ネットワークを利用し, ユニット間にまたがるデータ転送はユニット間ネットワークで行うことにより効率の良いデータ転送を可能にする.

2 高密度実装クラスタにおけるネットワークフレームワーク

高密度実装クラスタにおけるネットワークフレームワークを図 1 に示す. 本ネットワークフレームワークにおいて, 各ノードはユニット内ネットワークおよびユニット間ネットワークで階層的に結合される.

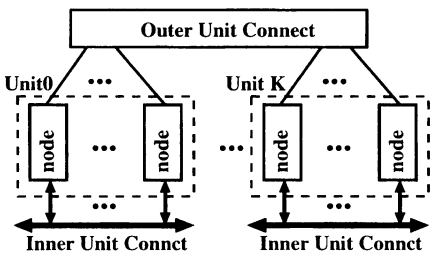

図 1: 高密度実装クラスタのネットワークフレームワーク

ユニット内外のネットワークは, ユニット内外の特性を生かしたネットワークを採用する. つまり, ユニット間ネットワーク (Outer Unit Connect) は,Ethernet などのユニット間接続でノイズが入りにくいような処理が施されているネットワークを採用し, ユニット内ネットワーク (InnerConnect) では信号線を直接結合させるような比較的に密なネットワーク (例えば, バス結合) を採用する. ユニット間では, グランドループなどのノイズの原因となる現象が生じやすいので, パルストランスを用いたアイソレーションを行う必要がある. 一方, ユニット内ではそのような現象が起こりにくく, ノイズ対策を施す必要がほとんどない. よって, ユニット内接続におけるノイズ対策としては, ダイオード (または抵抗) を使ったターミネーションを施す程度でよい. また, ユニット間では, ユニット内よりケーブル長が長くなるので, パラレル転送などは向いていないが, ユニット内はケーブル長が短いのでパラレル転送がしやすいので, それぞれの特性を生かしたネットワークをユニット内外に採用する.

3 高密度実装用の実験クラスタ

高密度実装クラスタのネットワークフレームワークを実現するための実験用クラスタとして,SCCB クラスタシステム [7] を用いた.SCCB クラスタシステムは PC ボード (PICMG 規格のボード CPU : PentiumIII600MHz) を 100Base/TX の Ethernet および Sync-Comm ネットワークで結合したクラスタシステムである. 100Base/TX の Ethernet 接続を Outer Unit Connect とし,Sync-Comm ネットワーク接続を Inner Unit Connect とする. 本クラスタでは,1 ユニットあたり 3CPU ボードが実装されている. 各 CPU ボードの PCI スロットに SCC ボート [6] と呼ばれる同期・通信用の PCI ボードを実装する.

3.1 SCC ボード

SCC ボードは,Control Chip および Inner Unit Connect 用コネクタ (Link_A および Link_B) のみで構成されている.Control Chip は, 同期・通信処理をハードウェアで実装することにより, 高速化を実現する.Inner Unit Connect 用コネクタは 20 ピンのパラレルケーブル用のコネクタであり, パラレルケーブルを用いて Link_A と Link_B を接続する (図 2 参照).

SCC ボードは, 本来, メッセージパッシング型の通信処理および拡張型バリア同期処理を低レイテンシで実現するために開発された PCI ボードである. しかし, 今回, その SCC ボードをユニット間接続のみに特化した PCI ボートとして開発する.

4 SCC ボードを利用したユニット内接続

ユニット内のノード接続方法として,SCC ボードを用いる. 本クラスタシステムでは, 中央のノードが両側のノードと接続する形態をとる.SCC ボードでは, リング接続も可能である. しかし, クラスタの物理的な形状から両端を接続する場合, ケーブル長

が長くなり, 通信速度の低下をまねく可能性がある. よって, 両端は接続しないようにする.

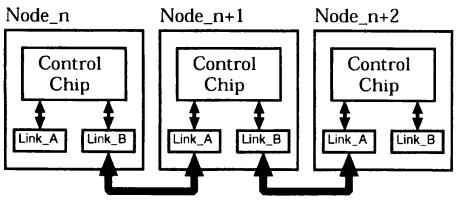

図 2: Inner Connect 接続形態

4.1 SCC のコントロール Chip

図 3 に Control Chip 内のブロック図を示す.Control Chip は,PCI-WISHBONE ブリッジ (PCI-WISHBONE Bridge), 同期制御部 (Sync Control), 通信制御部 (Comm Control), 疑似グローバルクロック部 (Pseudo Global Clock Counter System), パケット送受信処理部 (Packet Send/Receive Processing), および共有メモリで構成される.

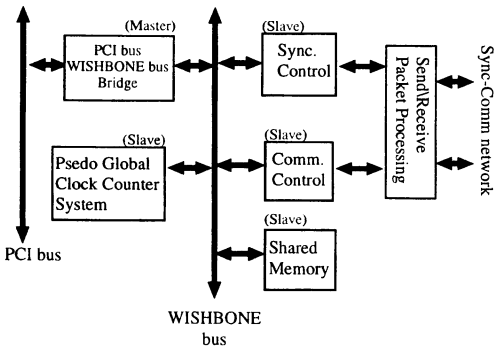

図 3: Control Chip 内のブロック図

PCI-WISHBONE ブリッジは,PCI バスプロトコルと WISHBONE バスプロトコルをインターフェースし,PCI のターゲット機能をサポートする. 同期制御部は,Inner Unit Connect を使用した同期処理アクセスを行い, 任意参加バリア,Fuzzy バリアの同期を処理する. 通信制御部は, パケットデータ処理を行う. 疑似グローバルクロック部は, 同期制御部と連携し, 高精度実行時間測定に必要な処理をハードウェアで行う. 送受信パケット処理部は, 通信モードを解析し, パケットを同期パケットと通信パケットに分けて格納する処理を行う. 共有メモリには, 各ブロック間が連携して処理する場合に必要なデータを格納する.

4.2 Control Chip 内でのブロック間接続

Contorol Chip の各ブロックは,WISHBONE を用いて接続する [8]. WISHBONE は,OpenCores で公開されているオープンソースの IP コア間インターフェースバスである.WISHBONE を用いることにより, 比較的簡単に各ブロック間を接続できる.

WISHBONE の規格には,Point to Point 接続・クロスバスイッチ接続などの接続形態があるが, 本 Chip では 4x4 Shared Bus Inter Connection 接続を用いる. このバスにおけるマスタ IP コアとして,PCI-WISHBONE ブリッジを設定し, スレーブ IP コアとして, 同期制御部・通信制御部・疑似グローバルクロック部・共有メモリを設定する.

4.3 ユニット内での通信方式

ユニット内でのデータ通信では,8 ビットの全 2 重の双方向通信を行う. その際, 同期・通信の種類別にモード (同期モード・通信モード) を設け, モードごとに効率のよい通信方式を採用する.

Inner Unit Connect のパラレルケーブルは,20 本のパラレルケーブルなので,8 本をデータ線に使い, 残りの 2 本をモード線に使用する. 通信手順としては, モード線にモードをある一定期間送出し, その後データを送出するという手順で行われる (図 4 参照).

4.3.1 モードによる通信切替方式

Inner Unit Connect の信号は, 全てパケット送受信処理部 (Send/Receive Packet Processing Block) へ送られる. パケット送受信処理部は, さらに送信部 (Send Block) および受信部 (Receive Block) に分かれる.(図 5 参照).

送信部では, パケットバッファからデータを取出し, そのデータにモード信号を付加して Inner Unit Connect を介して送信する.

受信部では, モード信号を解析して, モードに対応した制御部の受信パケットバッファへ受信パケット

を格納する. 受信パケットが Buffer に 1 つ以上格納されている場合,Receive Ack 信号を送出し, 同期・通信処理部へパケットが到着していることを通知する.

Mode Select(in) と Mode Select(out) の間で簡単なフロー制御を行えるようにする. Buffer が FULL (またはそれに近い状態) のときに,MoceSelect(in) にデータが届いてしまった場合,Mode Select(in) は Mode Select(out) に対して,send stop 信号を送出する.send stop 信号を受け取った Mode Select(out) は, パケット送信を一時停止するコントロールパケットを生成し, 相手ノードに送信する. そのコントロールパケットは相手ノードの Mode Slelect(in) に受け取られる.Mode Select(in) は stop 信号を Mode Select(out) へ送出する.stop 信号を受け取った Mode Select(out) は Start 信号が送出されるまで, パケットの送出を中断する.

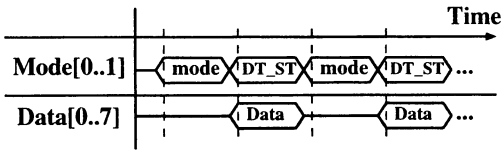

図 4: Inner Unit Connect におけるデータ通信のタイミングチャート

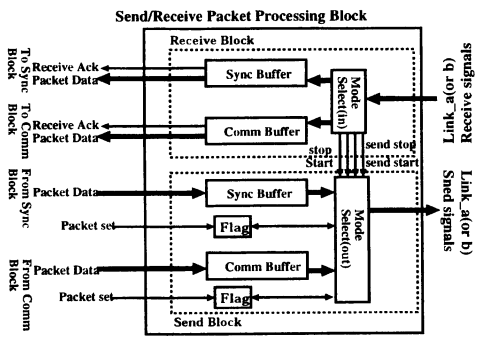

図 5: パケット送受信処理部の内部ブロック

4.4 同期制御部の構成

同期処理は同期パケットを介して行われる. 同期パケットはヘッダのみで構成され, サイズは 32Byte である. 同期制御部は, 同期データ保存用メモリ (Sync Memory) および同期パケット解析部 (Packet Analyser & Data Sender) で構成されている (図 6 参照).

同期データ保存用メモリは, さらにバリア参加情報 (B_mask), バリアポイント到達情報 (B_reach), バリア同期成立フラグ (B_flag) に分れる.

バリア参加情報およびバリアポイント到達情報は分散管理されており, 各ノードにおけるバリア参加情報およびバリアポイント到達情報のアドレスはシーケンシャルに番号付けられている. バリア同期成立フラグは全バリア参加情報に対するバリア同期成立状況を格納する. そうすることで, 同時に実行できるバリア数を増やし, 且つ同期フラグのアクセスを短くしている.

同期パケットは, 同期パケットの種類, 同期番号, 送信パターン, バリア参加情報, バリアポイント到達情報, 同期成立フラグで構成する. 送信パターン, バリア参加情報, バリアポイント到達情報はそれぞれユニット内のノード数分のビット (本ユニットの場合,3 ビット) で構成される. 同期パケットの種類は, バリアポイント到達パケット, バリア成立パケットの 2 種類である. 送信パターンは, どのノードに送信するか指示する情報であり, 各ビットがユニット内の各ノードに対応し, 送りたいノードに対応したビットを立てることにより, 同期パケット処理部が適切なリンクのパケット送受信処理部にパケットデータを送る.

同期パケット処理部では, 送信される同期パケットの生成および受信された同期パケットを解析が行われる. もし, 処理するパケット情報 (バリア参加情報およびバリアポイント到達情報) が自分の担当している同期メモリ関する情報であれば, 同期検出処理も行われる.

同期パケット処理部でのバリア同期処理は以下の手順で行われる.

1. バリアポイントに到達したノードプロセッサが同期パケット処理部へバリアポイント到達パケットを送る.
2. それを受け取った同期パケット処理部が同期番号に対するバリアフラグを立てる. また, 同期番号に対応する情報 (バリア参加情報, バリアポイント到達情報) を格納しているノードへバリアポイント到達パケットを送信する.
3. もし, そのノードが同期番号に対応する情報を

格納しているのなら, パケット内のバリア参加情報およびバリアポイント到達情報を新たに格納する（バリアポイント到達パケットを他ノードに送信しない）. ただし, バリアポイント到達情報はすでに格納されている情報とのORを格納する.

4. バリアポイント到達情報が更新された場合, バリア参加情報とバリアポイント到達情報のビットパターンを比較する. 同一であればバリア同期が成立したことにので, バリアに参加しているノードにバリア成立パケットを送信すると同時に, ノード内のバリアフラグをリセットし, バリア参加情報およびバリアポイント到達情報をクリアする.

5. バリア成立パケットが送られてきた同期パケット処理部は, そのパケットの同期番号に対応するバリアフラグをリセットする.

バリアポイントに到達したノードは, バリアポイント到達パケットを送出後, バリアフラグがリセットされたかどうかチェックすることでバリア同期動作を行う.

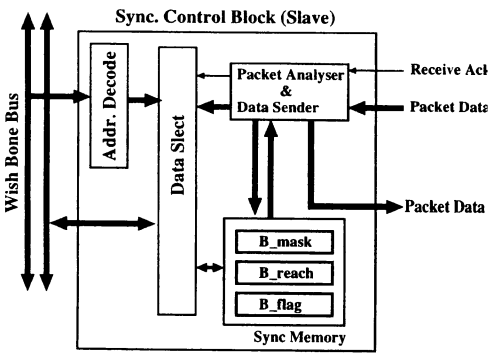

図 6: 同期制御部の構成

4.5 通信制御部の構成

通信制御部では, データ通信用のパケットを処理する. 通信パケットは, ヘッダ部 32Bytes, ペイロード 1〜224Bytes で構成する. パケットを処理する通信制御部の構成を図 7 に示す. 通信制御部では, パケットの種類を解析し, Collective 通信なら, ユニット内の全ノードにブロードキャストする機能を搭載する.

通信制御部（中央ノード）におけるデータ受信処理手順を以下に示す.

1. パケット送受信処理部から送られてくる Receive Ack 信号を検出後, パケットが到着したことを示すフラグをセットすると同時に, Buffer からパケットヘッダ部分を取出す.

2. パケットヘッダからそのパケットの転送の種類（Collective 通信など）を解析し, 必要なら, 送信したノードとは別隣のノードに転送する.

3. ノードプロセッサがフラグをチェックして, 受信したパケットをパケット送受信処理部の Buffer から取出す.

通信制御部（中央ノード）におけるデータ送信処理手順を以下に示す.

1. ノードプロセッサが, 送信パケットを「Header Analyser & Data Send」に送る.

2. 「Header Analyser & Data Send」は, 送信パケットの送り先や種類を解析する. 通信の種類が Collective 通信のようなブロードキャストを伴う転送の場合は, Link_A と Link_B の両方にパケットを転送する準備をする.

3. パケット Buffer に空きがあれば, 送信パケットを Buffer に格納する.

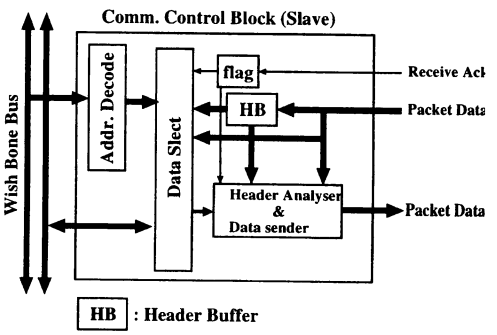

図 7: 通信処理部の構成

5 Collective通信におけるユニット内外データ転送方式

Collective通信を行う際は,階層的なネットワークをうまく利用して行う.本方式のネットワークフレームワークは,Cube Connected Cycle や Hypernet などのような階層ネットワークである.そこで,1つのユニットを3つのリンクを持ったユニットとし,ユニット間を Tree で結合したトポロジとしてデータ転送する.また,ユニット内では,Inner Connect によるブロードキャストによりデータを転送する (図8参照).

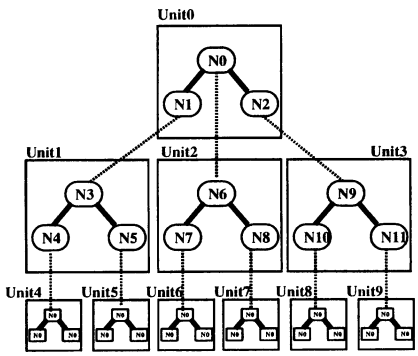

図 8: Collective 通信用 Tree ネットワーク

6 おわりに

PC をベースした高密度実装用実験クラスタシステム (SCCB-Cluster system) におけるネットワークフレームワークの構築方法について述べた.特に,Inner Unit Connect の実装方法について詳しく述べた.

本実験用クラスタは,近年発表されている高密度実装クラスタに比べ,実装密度が低い感じは否めない.しかし,本実験クラスタ上にネットワークフレームワークを実現し,性能評価することで,実装した方式の性能をを見積もり,より密度の高いクラスタに実装することを考えている.

今後は,Inner Unit Connect のハードウェア実装と Inner-Outer Unit Connect 間のインターフェース実装を行う.

参考文献

[1] IBM and Lawrence Livermore National Laboratory,"An Overview of the Blue Gene/L Supercomputer",Super Computing 2002, Nov. 2002.

[2] M.Warren, E.Weigle, W.Feng, "High-Density Computing:A 240-Node Beowulf in One Cube Meter ",Super Computing 2002, Nov. 2002.

[3] 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.

[4] 鯉淵道紘, 渡邊幸之介, 大塚智宏, 天野英晴,"RHiNET-2 クラスタを用いたシステムエリアネットワーク向けトポロジの実機評価" , 先進的計算基盤システムシンポジウム SACSIS2004.pp.381-388, 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] Kiyoshi Hayakawa, Satoshi Sekiguchi"Design and Implementation of a Synchronization and Communication Controller for Cluster Computing Systems," Proc. 4th Intel. Conf. High Performance Computing in Asia-Pacific Region, vol.I, pp76-81, May.2000.

[7] Kiyoshi Hayakawa, Masahiko Iwane, Satoshi Sekiguchi,"SCC Beowulf-Cluster System for Accurate Performance Analysis", 5TH Intl. Conf. High Performance Computing in Asia-Pacific Region, CD-ROM, Sep.2001.

[8] OPENCORES.ORG
(http://www.opencores.org)