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

2006－ARC－167（8）
2006－HPC－105（8）
2006／2／27

中密度実装クラスタにおける同期ユニットの設計および評価

早川　潔

本稿では，SCMD クラスタにおける同期ユニットの設計・実装および性能評価について述べる．SCMD クラスタは，1 筐体 (1 シャーシ）に 3 台の PC が実装され，そのシャーシが 10 シャーシ（ノード数は 30）搭載されたクラスタであり，シャーシ内ネットワークおよびシャーシ間ネットワークを搭載している．シャーシ内ネットワーク（SCIC ネットワーク）は，シャーシ内ノード間をより密に結合し，シャーシ間ネットワークは，とりまわしがよくスケーラビリティの高い汎用ネットワークでシャーシ外ノード間を結合する．本研究では，SCIC ネットワークを使用したシャーシ内ノードの同期処理を行うハードウェア同期ユニットを設計した．同期処理ユニットにおいて，新たな同期変数管理方式を採用した．その同期ユニットを SCMD クラスタの一部のノードに実装し評価をした．その評価では，シャーシ内ノード間のバリア同期処理を 3.86μs で処理できた．

Design and evaluation of synchronization unit for the Middle-density cluster systems

Kiyoshi Hayakawa

We introduces a hardware design and evaluation of a synchronization unit on SCMD-Cluster system whose nodes are connected each other via special network called SCMD network. SCMD-Cluster is built using PCs(30 nodes) for node-processor. Three nodes(mother boards) packed into one chassis each(involvement of one power unit)(i.e. 10 chassis PC cluster). Each node connects via SCMD network. SCMD network consists of two networks. One is Inner-chassis network, the other is Outer-chassis network. On Inner-chassis network, each node in a chassis connects via a tightly coupled network. On Outer-chassis network, the chassis connect via a loosely coupled network. We specify SCICnetwork as Inner-chassis network, and SCIC network card(PCI board) is developed. SCIC network card allow us to perform a fast barrier synchronization mechanism and a communication mechanism using SCIC network. SCIC network card achieves barrier latency of 3.86 μs on Inner-chassis network.

1　はじめに

近年，汎用 CPU の低消費電力化が進み，その CPU を使用した PC クラスタにおける低消費電力化の研究が行われている．また，低消費電力 CPU を使用することにより,CPU ファンなどの冷却装置が小型化され（または省かれ），より高密度に実装可能になってきている．

高密度実装クラスタとして，*Green Destiny*[1] や *MegaProto*[2] というクラスタが知られている．いずれのクラスタも 1 シャーシに 20 個前後の CPU が搭載されている．CPU はいずれも Transmeta の CPU を使用している．本研究室で開発中の SCMD クラスタにおいてもできるだけ低消費電力・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を目指している．

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ってきている．そのような状況の中で，故障 CPU やマザーボードの修復などが難しくなっている．　そこで，SCMD クラスタでは，FA などで使用する PICMG 仕様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 EmbeddedCPU を搭載することで長期間運用することを可能にする．PICMG のマザーボードにも Pentium M 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になってきている．

１シャーシに複数台数のノードが搭載されているクラスタの場合，ネットワークをシャーシ内とシャーシ外に分けて，構成したほうがより効率よ

大阪府立工業高等専門学校 総合工学システム学科
Osaka Prefectural College of Technology INDUSTRIAL SYSTEMS ENGINEERING

いデータ転送ができる．つまり, シャーシ間ネットワーク (Outer-Chassis network) は,Ethernet などのシャーシ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, シャーシ内ネットワーク（Inner-Chassis network）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. シャーシ内ネットワークでは，より高速なデータ通信が可能となり，それにともなって高速なバリア同期も構成可能となる．

そこで，本稿では，シャーシ内ネットワークを使用したハードウェア同期ユニットについて述べる．同期ユニットにおけるバリア同期変数管理方式を提案し，バリア同期の処理速度をシミュレーションおよび実機で測定し，バリア同期の処理速度削減効果について検証する．また，シャーシ内バリア同期とシャーシ間バリア同期とを合わせたクラスタ全体のバリア同期の処理時間を見積ることにより，クラスタ全体のバリア同期処理時間の削減効果も検証する．

2 SCMD クラスタ

中密度実装クラスタとして,SCMD クラスタシステムを構築した（図1参照).SCMD 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SCIC 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 接続をシャーシ間ネットワークとし,SCIC ネットワークをシャーシ内ネットワークとする（2 つのネットワークをまとめて「SCMD ネットワーク」とする）. 本クラスタでは,1 シャーシあたり 3CPU ボードが実装されている. 各 CPU ボードの PCI スロットに SCIC ネットワークボートと呼ばれる PCI ネットワークボードを実装する.

3 シャーシ内バリア同期処理

SCIC ネットワークを利用したシャーシ内バリア同期（以後「SCIC_Barrier」とする）を行う同期ユニットについて述べる.

3.1 SCIC ネットワークボード

SCIC ネットワークボードは,Control Chip およびシャーシ内ネットワークコネクタ（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. シャーシ内ネットワークコネクタは 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）.

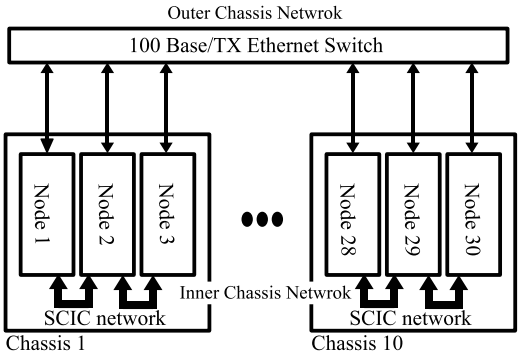

図 1: SCMD クラスタの構成

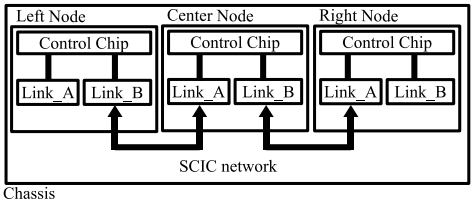

図 2: SCIC ネットワークを利用したノード間接続

SCIC ネットワークボードは, 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 [4]． しかし, 今回, この SCIC ネットワークボードをシャーシ内ネットワークに特化した PCI ボートとして開発する.

3.2 SCIC Control 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, 同期制御部, 通信制御部, 送受信パケット処理部, および共有メモリで構成される.

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,SCIC ネットワーク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. 送受信パケット処理部は, 通信モードを解析し,

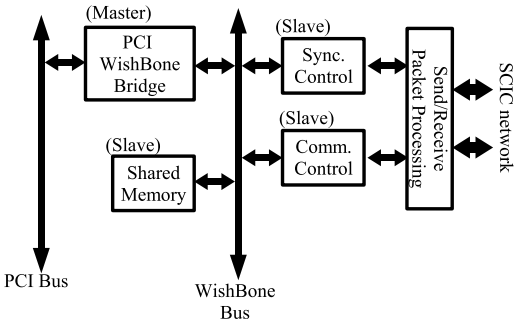

図 3: SCIC Control Chip の構成

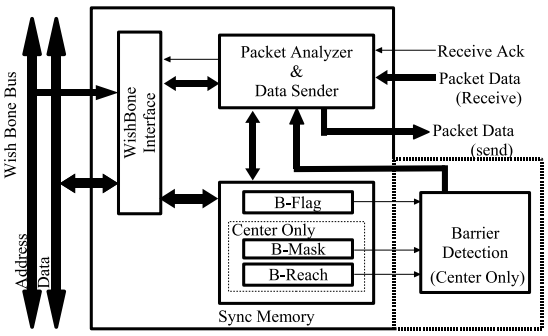

図 4: 同期制御部の構成

パケットを同期パケットと通信パケットに分けてパケット送受信処理を行う. 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.

本稿では，同期ユニットの設計・実装について述べるので，それに関連する「同期制御部」についてのみ言及する.

3.3 同期制御部の構成

図 4 に SCIC_Barrier を処理する同期制御部の構成を示す．同期制御部は， Wishbone インターフェース，パケット解析 & データ送出部, 同期メモリおよび同期検出部で構成される.

同期メモリの構成は， Center ノードと Right または Left ノードの間で異った形をとる． Center ノードの同期メモリは，バリアフラグ (図では B-Flag)，バリアマスク (図では B-Mask) およびバリア到達情報 (図では B-Reach) で構成される．一方， Right および Left の同期メモリは，バリアフラグのみで構成される．また，同期検出部は，Center ノードのみに実装される．バリアフラグおよびバリアマスクは 3 ビットの変数であり，各ビットがシャーシ内ノードに対応している.

同期パケットには，同期到達パケットおよび同期成立パケットと呼ぶ 2 種類のパケットを用意する．同期到達パケットは， Ceneter ノードへ同期成立を知らせるパケットであり，同期成立パケットは，Center ノードが Right および Left ノードへ同期の成立を知らせるパケットである．同期パケットはパケット解析 & データ送出部によって，自動的に生成される.

バリア同期を行う際に， Wishbone 経由で同期番号および同期参加情報が書き込まれ，その書き込まれた同期番号のバリアフラグをアクセスし，必要に応じて，同期パケットを自動的に生成する．つまり，ノードプロセッサが PCI および Wishbone を介して，バリアフラグをセットしたときに，パケット解析 & データ送出部が同期到達パケットを自動生成する．また，バリアが成立した場合，同期検出部がパケット解析 & データ送出部に同期成立パケット生成を依頼し，その依頼を受けたパケット解析 & データ送出部はバリア成立パケットを自動生成する.

3.3.1 同期情報の管理方法

同期メモリの中にある同期変数は変数ごとに管理方法が異なる．バリアフラグは各ノードで分散管理し，バリアマスクおよびバリア到達情報は Center ノードが集中管理する．バリアフラグは同じ同期番号のバリアフラグが各ノードに配置される．各ノードの同期番号が同じバリアフラグは同期成立パケットによって， coherence が保たれる．バリアマスクおよびバリア到達情報は， Center ノードが集中管理する． Center ノード以外のノードが，バリアマスクおよびバリア到達情報を変更する場合，同期到達パケットを Center ノードに送ることにより，Center ノードがそれらの変数を変更する.

Center ノードにおけるバリア同期手順に伴う同期管理手順を以下に示す.

1. Center ノードのプロセッサが同期ポイントに到達したら，特定の番地に同期番号と同期情報（バリアマスクおよびバリアフラグの情報）を書き込む.

2. Wishbone インターフェースがそれら書き込みを検知して，その同期番号に対応したバリアフ

ラグをセットし，バリア到達情報を更新し，同期メモリにバリアマスクを保存する．

3. 他のノードから同期到達パケットが届いたら，パケット解析 & データ送出部が同期メモリの同期情報を書きこむ．

4. 同期検出部が，書き込まれたバリア情報 (Wishbone インターフェースまたはパケット解析 & データ送出部が書いた情報）をもとに，同期を検出する．同期検出部がバリア同期完了を検出した場合，パケット解析 & データ送出部に同期成立パケット生成を依頼し，バリアフラグをリセットする．

5. 同期検出部の依頼にしたがって，パケット解析 & データ送出部が同期成立パケットを自動生成し，Left および Right ノードにそのパケットを送る．

Left および Right ノードにおけるバリア同期手順に伴う同期管理情報手順を以下に示す．

1. Left または Right ノードのプロセッサが同期ポイントに到達したら，特定の番地に同期番号と同期情報を書きこむ．

2. Wishbone インターフェースがそれら書き込みを検知して，その同期番号に対応したバリアフラグをセットし，パケット解析 & データ送出部へ同期到達パケットを生成するように要求する．

3. パケット解析 & データ送出部は Wishbone インターフェースの要求にしたがって，同期到達パケットを生成して Center ノードへ送る．

4. Center ノードから同期成立パケットが届いたら，パケット解析 & データ送出部がバリアフラグをリセットする．

4 クラスタ全体バリア同期

クラスタ全体のバリア同期は，シャーシ内外のネットワークを利用して行う．基本的には，シャーシ外ネットワークトポロジーを図のような Tree トポロジーと考えて，シャーシ内のみのバリア同期を行い，それが終了後にシャーシ間で同期を行う形でバリア同期を行う．

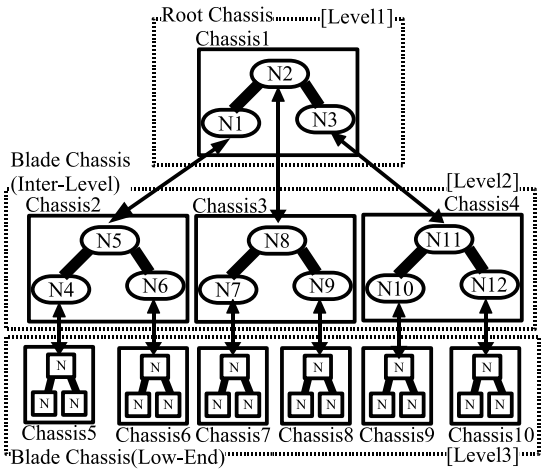

図 5: クラスタ全体同期のネットワーク構成

クラスタ全体のバリア同期は，各シャーシを「RootChassis」，「BladeChassis（Inter-Level)」，「BladeChassis(Low-End)」の 3 つに分け（図 5参照），それぞれ異なった手順で同期を行う．

RootChassis の同期手順を述べる．まず，RootChassis 内で SCIC_barrier を行い，同期が成立したら，RootChassis 内の各ノードが下位階層 BladeChassis の Center ノードに RootChassis のみの SCIC_barrier の完了を知らせる．次に,RootChassis 内の各ノードは下位階層 BladeChassis 全てが同期完了したことを示すパケットを受信する処理を行い，その後，再度，SCIC_barrier 同期を行う．2 回目の SCIC_barrier が終了したら，それを成立を知らせるパケットを下位階層 BladeChassis の Center ノードへ送る．

BladeChassis（Inter-Level) の同期手順を述べる．まず，BladeChassis（Inter-Level) 内の Center ノードが上位階層の Chassis から送られてくるパケットを受け取り，その後，SCIC_barrier を行う．BladeChassis（Inter-Level) 内の Right および Left ノードが上位階層のシャーシでのバリア同期完了を示すパケットを下の階層の Center ノードに送り，返送されるパケット受け取る．その後，再度，SCIC_barrier を行う．BladeChassis（Inter-Level) 内の Center ノードが上位層のノードにパケットを送り，返送されるパケット受け取る．再度，SCIC_barrier を行い，SCIC_barrier 終了後 Right および Left ノードが階層にパケットを送る．

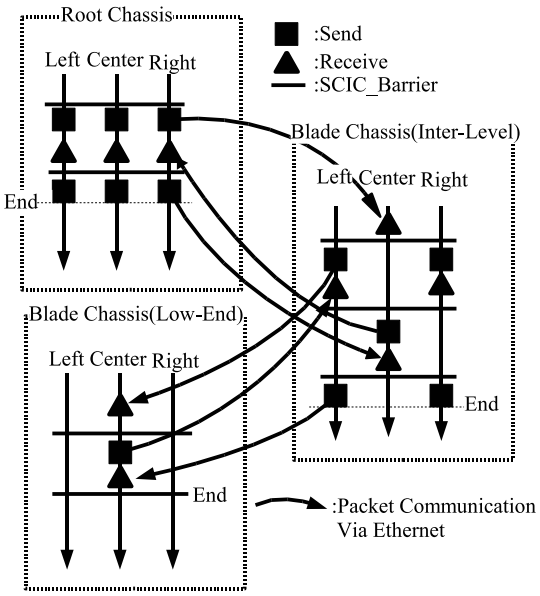

図 6: クラスタ全体同期の処理手順

BladeChassis（Low-End) の同期手順を述べる．上位階層の Blade(または Root)Chassis から送られてくるパケットを BladeChassis（Low-End) 内の Center ノードが受け取り，その後，SCIC_barrier を行う．BladeChassis（Low-End) 内の Center ノードが上位層のノードにパケットを送り，返送されるパケットを受け取る．最後に，SCIC_barrier を行う．

5 性能評価

同期ユニットの評価として，SCIC_Barrier における同期ユニット内の処理時間を見積もり，実機で SCIC_Barrier の実行時間を測定した．また，クラスタ全体バリア同期の実行時間も見積もった．

5.1 SCIC_Barrier のレイテンシ

SCIC_Barrier の同期ユニット内の処理時間をシミュレーションによって見積もった．シミュレーションはアルテラ社の QuartusII（Ver5.1）を使用した．

図 7に SCIC_Barreir 処理のタイミングチャートを示す．Center ノードにおける SCIC_Barrier の処理手順をおおまかに３つの工程にわけ，Right・Left ノードの SCIC_Barrier の処理手順をおおまかに２つの工程にわけ，タイミングチャートに表した．表 1に SCIC_Barrier の各工程の処理時間を示した．

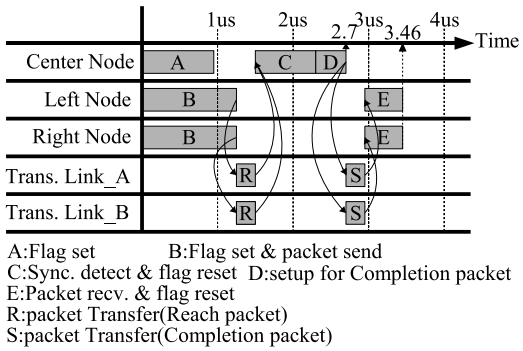

図 7: SCIC_Barrier の処理工程

表 1: SCIC_Barrier の各工程における見積もり処理時間

Center node		Left or Right node	
Process	time(μs)	Process	time(μs)
A	0.96	B	1.25
C	0.79	E	0.495
D	0.39		
R	0.27	S	0.27

図 7のタイミングチャートおよび表 1に各処理の処理時間から Center ノードの SCIC_Barrier は 2.7μ s で終了し，Left または Right ノードの SCIC_Barrier は 3.46μ s で終了するという結果を得た．SCCB クラスタ [5] での同期ユニットの処理時間（約 0.2μ s）に比べると約 17 倍処理時間がかかっている．

実際に，Right および Center ノード 2 ノードでの SCIC ネットワークを使用したバリア同期の処理時間を測定した．その結果，Center ノードは 2.37μ s で完了し，Right ノードは 3.86μ s で完了した．見積もり実行時間と実際に計測した結果がほぼ一致しているが，これは，デバイスドライバの処理時間を含んだ時間なので，実際のシミュレーションのより早く処理されたと思われる．

Enternet のみを使った MPI_Barrier の場合，2 台で約 120μ と約 1/30 削減されているが，SCCB クラスタ [5] でのバリア同期処理時間（3.2μ ）よりは処理時間が延びている．これは，同期ユニットの差というより，デバイスドライバの処理時間の差である．

5.2 クラスタ全体バリア同期の評価

クラスタ全体バリア同期の実行時間を見積もった．図の手順をおおまかに見積もると以下のようになる．

$N = 2, 3$ のとき，

$$T_{EB} = T_{IB} \tag{1}$$

$N \geq 4, (L_I \geq 0)$ のとき

$$T_{EB} = (4 + 3L_I)T_{IB} + 3(L_I + 1)T_{SR} \tag{2}$$

ここで，T_{EB} はクラスタ全体のバリア処理時間，T_{IB} はSCIC_Barrier レイテンシ，L_I はTree トポロジのInter-Level 番号，T_{SR} はsend/receive の処理時間である．

式 (1) および式 (2) をグラフで表す（T_{SR} をパラメータとして）と図 8 となる．図 8 において，イーサーネットのみを利用したMPI_Barrier の実行時間と比較した．T_{SR} が 60us 程度だと，SCMD ネットワークを使用したバリア同期が有利であるが，T_{SR} が 120 を超えると，Ethernet のみを使用したMPI_Barrier が有利になるということが推測される．

6 おわりに

シャーシ内ネットワークを使用したハードウェア同期ユニットの設計および評価を行った．SCIC ネットワークボードに搭載するコントロールchip の概要を説明し，そのボードによって構成されるSCIC ネットワークについて述べた．また，SCIC ネットワークを利用したシャーシ内のバリア同期制御部の構成について述べ，そのブロックで行われるバリア変数管理手法について述べた．

性能評価として，シャーシ内バリア同期の処理見積もり時間および 2 ノードを利用したSCIC_Barrier の予備評価を行った．その結果，2 ノードのSCIC_Barrier が 3.86μs で処理できた．この結果は，イーサーネット使用したMPI_Barrier の実行時間の約 30 分の 1 に相当する．

また，クラスタ全体バリアにおいても，send/Receive の実行時間が 60μs を超えなければ，イーサーネットのみを利用したMPI_Barrier よりよい結果になると見積もることができた．

今後の課題として，実機 3 台でSCIC_Barrier を実行させ，クラスタ全体のバリア同期の実機測定を行うことと，このバリア同期を実アプリケーションに有効的に使うための方法を検討することが挙げられる．

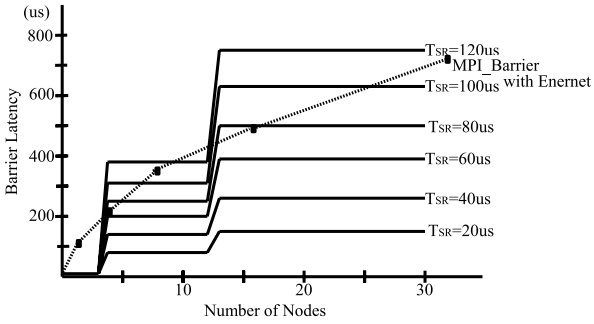

図 8: クラスタ全体バリア同期と MPI_Barrier との比較

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 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] 北村, 濱田, 宮部, 伊澤, 宮代, 田邊, 中條, 天野, "DIMMnet-2 ネットワークインタフェースコントローラの設計と実装", 先進的計算基盤システムシンポジウム SACSIS2005, pp.293-300, May.2005.

[4] K.Hayakawa, "SCCB CLUSTER SYSTEM-SYNCHRONIZATION AND PSEUDO GLOBAL CLOCK COUNTER SYTEM-", 23rd IASTED int'l conf. on PDCN2005, pp.216-221, Feb.2005.

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