ハイパフォーマンス
コンピューティング 57－9
（１９９５．８．２４）

クラスタ型ベクトル並列スーパコンピュータ S－３０００クラスタシステムのアーキテクチャと特性評価

田中 輝雄[1]，深川 正一[1]，井村 淳一[2]，後 保範[2]，
玉置 由子[3]，榊原 忠幸[1]，稲上 泰弘[3]
tetanaka@kanagawa.hitachi.co.jp

[1](株)日立製作所　汎用コンピュータ事業部
[2](株)日立製作所　ソフトウエア開発センタ
[3](株)日立製作所　中央研究所

概　要

最大演算性能９６GFLOPSのＳ－３０００クラスタシステムを開発した．主記憶共有型ベクトルプロセサＳ－３８００相当のシステムを高速接続機構を介して最大４組接続するクラスタ型ベクトル並列スーパコンピュータである．２階層の並列処理機構として，クラスタ内はＳ－３８００で開発した主記憶共有並列処理機構を利用し，クラスタ間はメッセージパッシング型の並列処理方式を基本に，高速データ通信機構を開発した．実機による特性評価を行ない，２階層の並列処理機構が効率良く実行できることを確認，また，連立一次方程式のＬＵ分解法プログラムを用いて実効性能７８．２GFLOPSを実現し，８０％以上の実効効率を得た．

The cluster-type vector parallel supercomputer "*S-3000 Cluster System*" - its architecture and characteristic evaluations -

Teruo TANAKA[1], Masakazu FUKAGAWA[1], Junichi IMURA[2], Yasunori USHIRO[2],
Yoshiko TAMAKI[3], Tadayuki SAKAKIBARA[1], Yasuhiro INAGAMI[3]
tetanaka@kanagawa.hitachi.co.jp

[1] General Purpose Computer Division, Hitachi Ltd.
[2] Software Development Center, Hitachi Ltd.
[3] Central Research Laboratory, Hitachi Ltd.

Abstract

We have developed the *S-3000 Cluster System*, a cluster-type vector parallel supercomputer which achieves 96 GFLOPS as a peak performance. The *S-3000 Cluster System* is composed of a maximum of four machines similar to the S-3800 TCMPs (Tightly Coupled Multi Processors) linked by a high speed interconnection network. The *S-3000 Cluster System* uses two levels of parallelism. The first is a TCMP system with a shared memory. The second is a distributed memory scheme composed of four TCMP systems which supports low-overhead message-passing. We have shown effectiveness of this approach by performance evaluation on a real machine. We have obtained sustained performance of 78.2 GFLOPS and over 80% of the theoretical peak performance on a LU decomposition algorithm for large scale linear equations.

1．はじめに

並列処理方式の実用化によりスーパコンピュータの性能トレンドは従来の半導体テクノロジの進歩に比例したトレンド曲線から大幅に上方修正された．現在4台から16台程度のベクトルプロセサが主記憶を共有する主記憶共有型システムや，富士通ＶＰＰ５００[1]のような完全分散記憶型のシステムが実用化されている．日立では平成4年12月に主記憶共有型スーパコンピュータＳ－３８００を開発，出荷している[2－5]．

また最近，ＮＥＣやＣＲＡＹからは次世代スーパコンピュータとして，主記憶共有型システムをネットワークあるいはファイルシステムで複数台接続するクラスタ構成のシステムが発表されている．このようなクラスタ型システムとしては，従来からイリノイ大学のＣｅｄａｒ[6]やＣＲＡＹ社[7]などの例がある．

クラスタ型システムは，

（1）現状の主記憶共有型システムの上位互換性を持つため，いままでの運用を継続でき，計算センタ向きである，

（2）クラスタを構成する主記憶共有型システムは比較的大容量の主記憶を共有するため，並列化できない応用に対しても1台のプロセサにより大規模計算を行なうことができる，

（3）ある程度の並列処理までは，主記憶共有型で対応することができるため，従来の自動ベクトル化／並列化機構を用いることができる，

などの特徴を持つ．さらに，

（4）クラスタ間の結合網を用いてデータを授受し，クラスタ間で協調処理を行ない，超大規模の数値シミュレーションを実行することができる．

しかし，この協調処理はクラスタ内／クラスタ間の2階層の異なる並列処理で実行するために，並列処理オーバヘッドの増大，プログラミングの複雑さなどの問題が発生する恐れがある．

このようなクラスタ型のシステムとして，Ｓ－３８００相当のスーパコンピュータシステムをひとつのクラスタとし，このクラスタを高速接続機構（以下では，ＨＳＩＣと略す）を用いて接続し，クラスタ間で協調して大規模数値シミュレーションの実行を可能とするクラスタ型ベクトル並列スーパコンピュータＳ－３０００クラスタシステムを開発した．

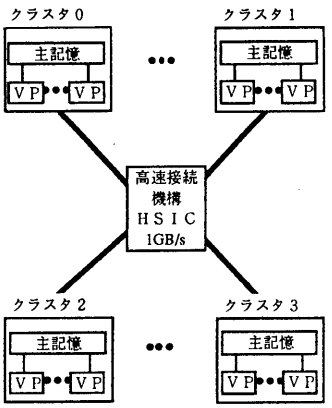


図1　スター型結合によるクラスタ型並列システム
Fig. 1 Star connected cluster type vector-parallel system.

本論文では，Ｓ－３０００クラスタシステムのアーキテクチャおよび大規模数値計算処理実行時の特性解析結果を示す．

2．2階層並列処理

Ｓ－３０００クラスタシステムは，クラスタ内／クラスタ間の2階層の並列処理を行なう．

（1）クラスタ内並列処理

主記憶共有型の並列処理機構を用いる．そのためＳ－３８００において開発したコンパイラによる自動ベクトル化機構，自動並列化機構をそのまま利用することができる．

（2）クラスタ間並列処理

クラスタ間並列処理では，クラスタ間通信処理の扱いが重要となる．多くの数値シミュレーションでは，次の特徴を持つ．

（a）転送パタンを静的に決定することができる．

（b）したがって，プログラム上でデータを送信側プロセサ（またはクラスタ）で送信する順番に受信側プロセサ（またはクラスタ）で受け取ることを保障することができる．

（c）大容量データ転送が主でありスループット重視である．

このため，通信形態としてはメッセージパッシング型が効率がよい．ここでは通信処理の高速化であり，(a)ソフトウエアオーバヘッドの削減，(b)不要な同期処理の排除が重要課題となる．

3．Ｓ－３０００クラスタシステムの実現

3．1　構成

高速接続機構（ＨＳＩＣ）は，最大1ＧＢ／ｓのデータ転送能力を持つＧＢオーダの大規模メモリシステムである．複数の計算機システム（本論文ではクラスタ）から共有することが可能であり，ＨＳＩＣと個々のクラスタ間の最大データ転送スループットは最大５００ＭＢ／ｓである．ＨＳＩＣは複数のクラスタと接続することが可能であることから，種々の接続形態が考えられる．

たとえば，図1に示すように，ＨＳＩＣを中央に配置しハブとして位置付け，すべてのクラスタから接続するスター型の接続である．これは，すべてのクラスタ間で直接データのやり取りを行なうことができるので汎用性が高い．しかし，複数のクラスタか

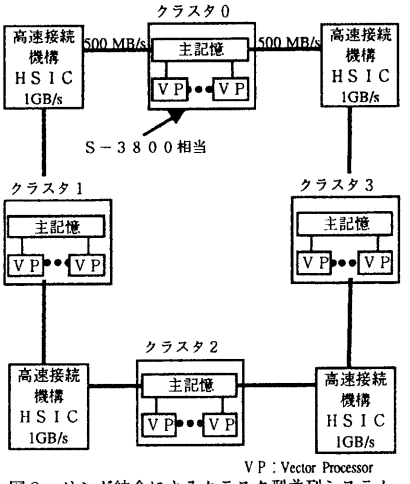


図2　リング結合によるクラスタ型並列システム
Fig. 2 Ring connected cluster type vector-parallel system.

ら同時にアクセスが行なわれたときはＨＳＩＣにおいて競合が発生し，クラスタ間のデータ転送処理の性能低下をもたらす．大規模シミュレーションでは，ＳＰＭＤ(Single Program Multiple Data stream)的に記述できる処理が多く，クラスタ間データ転送処理は同時に発生する可能性が高い．そこで，すべての隣接するクラスタ間にＨＳＩＣを配置し，リング状に接続することにより，ＨＳＩＣの台数分の通信性能向上をはかる（図２）．

今回の開発では，大規模シミュレーションの高速処理を行なうためにリング型の接続形態を取ることとした．また以下では，並列処理のプログラミングスタイルはＳＰＭＤを対象とする．

３．２　クラスタ間高速データ転送方式

クラスタ間の並列処理方式の概要を近接作用問題を例に説明する．基本的な近接作用問題は，領域を分割し，それぞれを分散記憶に配置し，分割された境界面のデータをクラスタ間で転送しあう．このデータ転送にメッセージパッシング方式を用い，送り手側の send 処理と，受手側の receive 処理により実現する．

メッセージパッシング型転送で送信側クラスタの SEND 命令と受信側クラスタの RECEIVE 命令を独立に実行させるためには，中間にバッファ領域を必要とする．ふつうは，このバッファ領域を受手側の主記憶上に確保するが，Ｓ－３０００クラスタシステムでは，この役目をＨＳＩＣが行なう．クラスタ間データ通信の原理を図３に示す．それぞれのデータ転送パスごとにＨＳＩＣ上にＦＩＦＯ型のバッファを構築する．そして，データ転送命令（SEND および RECEIVE）でこのＦＩＦＯを指定する．データの送受信を監視する制御変数をそれぞれの論理的なパスごとに準備することにより，制御変数に対する排他処理を必要としない．

ＨＳＩＣ上のＦＩＦＯの実現方法とそれを用いたクラスタ間の転送プロトコルを図４に示す．ここでは，それぞれのクラスタごとにＨＳＩＣを分割して割り付け，各クラスタの管理領域に対して，他方のクラスタからの書き込みを許さない方式とした．これは双方からの書込みを許すことにより必要となるシリアライズ処理を不要にするためである．一方のクラスタの管理対象とするＨＳＩＣ上の領域をそれぞれ他方のクラスタから読出しのみできるようにし，一方通行のデータ転送パスを構築する．

この領域を対称形に構成することにより，双方向のデータ転送パスの実現を可能とする．このとき，ＦＩＦＯ領域の構成（分割数，サイズ）はプログラム開始時に，そのプログラムに最適な割付けをプログラム中で行なう．分割数を多くすることにより，送信側 SEND 命令の受信側 RECEIVE 命令からの先行発行数を大きく許すことができる．もし，並行して２つ以上の異なるデータを転送したいときは，このデータ転送パスを複数設定する．なお，ＨＳＩＣ上へのＦＩＦＯ領域の確保は，各クラスタのＨＳＩＣ割付け時にＯＳが設定する．

ＦＩＦＯはＨＳＩＣのメモリ上に構成したＦＩＦＯバッファとその制御変数ＮｒＮｓからなる．このプロトコルでは，１回のデータ転送処理に，ＦＩＦＯへのデータ書き込み，ＦＩＦＯからのデータ読出しのほか，制御変数ＮｒＮｓの読出し，書込みを含めて６回のＨＳＩＣアクセス処理が必要となる．しかし，事前の制御変数ＮｒＮｓの読出し時に，各クラスタ内にそのコピーを確保することにより，ＦＩＦＯバッファがあふれそうになるまでは，新たに制御変数ＮｒＮｓの読出し処理は必要なくなる．

高速化のためには制御変数ＮｒＮｓをＨＳＩＣから独立させ，専用制御論理を実現することも考えられるが，今回は大量データ転送時のスループット重視としＨＳＩＣのアクセスパスを共用した．専用制御論理を実現したときの性能は４．１節で考察する．

３．３　ユーザインタフェース

プログラムを実行するためのジョブの起動，終結方法および並列化プログラミング方法を示す．

３．３．１　ジョブの起動，終結

ＰＶＭ(Parallel Virtual Machine)の環境を利用する．ユーザはあるひとつのクラスタのプログラム（ＳＰＭＤ型）を実行すること

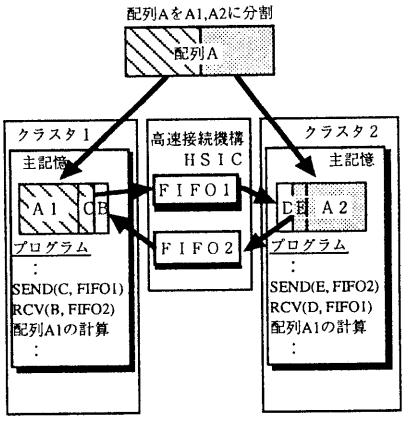
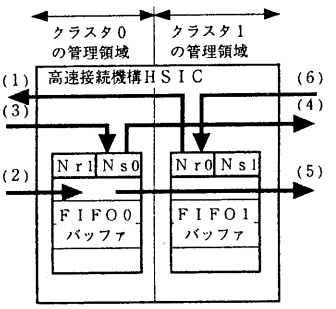


図３　S-3000クラスタシステムによる並列処理の概要
処理Anew(i)=f (A(i-1),A(i),A(i+1))の並列実行
Fig.3 Outline of parallel processingon the S-3000 Cluster system
example: Anew(i)=f (A(i-1),A(i),A(i+1))



Ｎｓ０：FIFO0上の Send 先ポインタ
Ｎｒ０：FIFO0上の Receive 先ポインタ
Ｎｓ１：FIFO1上の Send 先ポインタ
Ｎｒ１：FIFO1上の Receive 先ポインタ

データ転送手順(例：クラスタ０からクラスタ１へ)
＜送信側（クラスタ０）＞
(1) Ｎｒ０を読出し，チェック．[HSIC Read命令]
(2) FIFOバッファへデータ書込み．[HSIC Write命令]
(3) Ｎｓ０の更新．[HSIC Read命令]
＜受信側（クラスタ１）＞
(4) Ｎｓ０を読出し，チェック．[HSIC Read命令]
(5) FIFOバッファからデータ読出し．[HSIC Read命令]
(6) Ｎｒ０の更新．[HSIC Write命令]
以上を繰返し，実行．

注：クラスタ内のコピーで判断できる時は(1)(4)は不要

図４　クラスタ間転送プロトコルの概要
Fig.4 Protocol for data transfer between clusters.

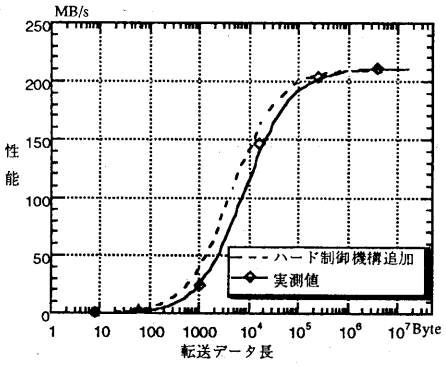


図5 クラスタ間データ転送基本性能（パタン1）
Fig.5 Performance of data transfer (test pattern 1).

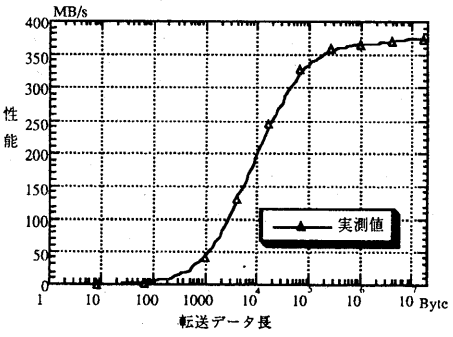


図6 クラスタ間データ転送基本性能（パタン2）
Fig.6 Performance of data transfer (test pattern 2).

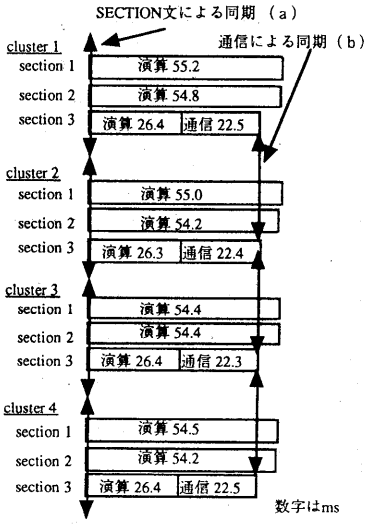


図7 クラスタ並列処理における同期の振舞い
Fig.7 Behavior of synchronization of cluster parallel execution.

により，ＰＶＭライブラリが自動的に他のクラスタのプログラムを起動する．それぞれのクラスタごとに実行されるプロセスが，自分がどのクラスタで実行されているか，あるいは何台のクラスタで実行されているかを，実行時にＰＶＭライブラリから受け取ることができる．このため，使用クラスタ数をパラメタにすることが可能となり，クラスタ数を変化させても基本的にプログラムを一種類で作成することができる．

プログラム中で，クラスタ間のデータ転送パスを設定／解放処理を行なうが，この設定／解放処理については基本的なテンプレートが準備してあり，特に，特殊な使い方を必要としなければ，そのままそのテンプレートを用いて実行することができる．

3．3．2　ユーザプログラム

ユーザプログラムでは，2階層の並列処理のためのプログラミングを行なう．

クラスタ内は従来の自動ベクトル化および並列化コンパイラを適用する．今回の開発では，各クラスタにはクラスタ間データ転送のための専用の通信プロセサは組込んでいない．そのためクラスタ間データ転送はクラスタ内のプロセサのひとつが対応する．高速化のために重要な[8]データ転送処理と他プロセサによるベクトル演算処理とのオーバラップを実現するために，基本的にコンパイラによるsection 型の並列処理方式を利用する．ただし，基本はＳＰＭＤ型のプログラミングスタイルであり，これをクラスタ内にも適用することにより，クラスタ内のｋ台のプロセサに対して，１台のプロセサが通信処理を含む演算処理をｋ－１台のプロセサが同一の演算処理のみを実行するため，section ごとのプログラムは高々２通りを作成すればよい．

クラスタ間については，メッセージパッシング型のデータ通信ライブラリが用意されており，明示的に通信命令(SEND, RECV)を記述する．送信命令(SEND)は，転送すべきデータの先頭アドレス（転送すべき配列の先頭位置），データ数および転送パス名を指定する．受信命令(RECV)は，受信先のアドレス（受信すべき配列領域の先頭位置），転送データ数および転送パス名を指定する．クラスタ間の同期処理は通信命令で行ない，データの受信により送信側クラスタとの順序関係を保障する．転送データの到着／未到着の制御はＨＳＩＣ上のＦＩＦＯの管理で行なわれる．

4．特性評価

Ｓ－３０００クラスタシステム上でプログラムを実行し，特性評価を行なった．ここでは，基本通信性能，クラスタ並列処理での同期処理の挙動，およびＬＵ分解法プログラムを用いたアプリケーションレベルでの評価結果を示す．

4．1　転送基本性能評価

２つのクラスタ間のデータ通信能力を評価した．

パタン１：ひとつのデータ（Ｄ１）を２つのクラスタ間で送りあう処理である．SEND命令とRECEIVE命令を交互に実行する．そのため，ＨＳＩＣ上でのデータ転送による競合は発生しない．

パタン２：パタン１に対して，さらに同時にもうひとつのデータ（Ｄ２）を送りあう．ＨＳＩＣ上でつねに競合が発生する．これはまた，一方のクラスタから他方のクラスタに対して，ひとつのデータを分割してパイプライン的に転送することと性能的に等価になる．

パタン１の結果を図５に示す．実線が実測値である．転送デー

タ長がほぼ数１００ＫＢのオーダになると，２００MB/sのデータ転送能力を発揮する．

破線のハード制御機構追加とは，３．２節に示したＦＩＦＯバッファの制御変数ＮｒＮｓをＨＳＩＣ上でなく，専用ハードウエアで実現した場合の予測値である．ハード制御機構の追加により，転送データ長が数ＫＢ～数１００ＫＢまでの場合は同じ効果を得るデータ長を約１／２に短縮することができる．

不完全ＬＵ分解による前処理などのアルゴリズムの高速化を行なう場合は，このような短い転送データ長が重要となる．

パタン２に対する結果を図６に示す．データＤ１，Ｄ２合わせて，パタン１と比較し，ほぼ２倍のデータスループットとなる．性能が完全に２倍にならないのは，ＨＳＩＣ上でのハードウエアによるデータ転送競合が発生しているためである．

４．２　クラスタ並列処理の挙動

Ｓ－３０００クラスタシステムでは，クラスタ内の並列処理とクラスタ間の並列処理を同時に実行する．並列処理において重要な項目のひとつは同期処理である．クラスタ内はコンパイラによるsection型の並列処理を行ない，SECTION文によってクラスタ内プロセサ間の同期をとる．一方，クラスタ間の並列処理では，ＨＳＩＣ上のＦＩＦＯにおけるデータの授受によって実質的に同期をとる．したがって，全体の同期はこれら２つの同期により，緩やかに行なわれる．図７に，この同期処理のようすを近接作用問題で代表的なRed／Black SOR プログラムの１反復ステップでの状況をもとに説明する．各クラスタではsection型の並列処理を行ない，section3を担当するプロセサが通信を行なう．データの依存関係を保つために，クラスタ間通信処理で得られたデータをもとに計算を行なう境界領域の計算は，section3を担当するプロセサが確実に実行する．図からわかるように，section文による同期(ａ)は各クラスタ内のプロセサ(各sectionを実行するプロセス)間で行なわれ，通信による同期(ｂ)は各クラスタのsection3間で行なわれる．これらの同期は全体での同期でないため自由度が存在するが，各クラスタごとの処理の分割が均一であること，クラスタ内の（演算のみを実行する）プロセサに対する処理の割付が均等であることが満たされると，図のように，整然と実行されているようすがわかる．図７は特別な反復ステップを取り出したのではなく，他の反復ステップも同様である．したがって，緩やかな同期機構で接続されているにもかかわらず，クラスタにおける２階層の並列機構におけるオーバヘッドはほとんど無視することができることがわかる．

４．３　ＬＵ分解法を用いた性能評価

応用レベルの評価として，ここでは，ＬＵ分解法に基づく連立一次方程式の直接解法を扱う．この解法を評価することにより，文献[９]から多くの他のスーパコンピュータや超並列計算機と比較検討することができる．

４．３．１　ＬＵ分解法のベクトル並列アルゴリズム

まず，プログラムをＳ－３０００クラスタシステム向きに並列化する．ここでは，各クラスタへのデータの分割方法，２階層の並列化手法について示す．簡単のために，クラスタ数を３，クラスタ内のベクトルプロセサ数を２とする．もちろん，他の構成でも一般性は失われない．

（１）クラスタ間のデータ分割

行列を各クラスタに対しインタリーブ的に分割する．これはＬＵ分解法では反復処理ごとに行列内の計算対象となる領域が小さく変化するためである．インタリーブした各々のブロックの列数は，各クラスタ内のすべてのベクトルプロセサが効率よく動作可能な幅を確保する．

（２）並列化アルゴリズム

図８に並列化アルゴリズムのフローチャートを示す．インタリーブ的に分割したために，部分軸選択処理 (pivoting) は各反復ステップ毎に担当するクラスタが入れ変わる．この処理自体はベクトル化はできるが並列化は困難である．ベクトル化についてもかなり効率が悪い．したがって，K+1回目の部分軸選択処理をＫ回目の行列消去処理 (decomposition) と並列に実行する．

４．３．２　ＬＵ分解法の性能評価

実行結果を，表１，図９および図１０に示す．まず，表１はプロセサ数およびクラスタ数を変化させた各Ｓ－３０００クラスタシステムでの実行結果を示す．

最大構成のモデル４１２（４クラスタ×３ベクトルプロセサ，最大演算性能９６GFLOPS）では７８．２GFLOPSを実現した．これは８１％の効率，半性能長Ｎ１／２は４８８０である．

比較のために，主記憶共有型スーパコンピュータＳ－３８００の値を表１に示す．同一のプロセサ数であれば，Ｓ－３０００クラスタシステムでも遜色ない性能を実現できる．

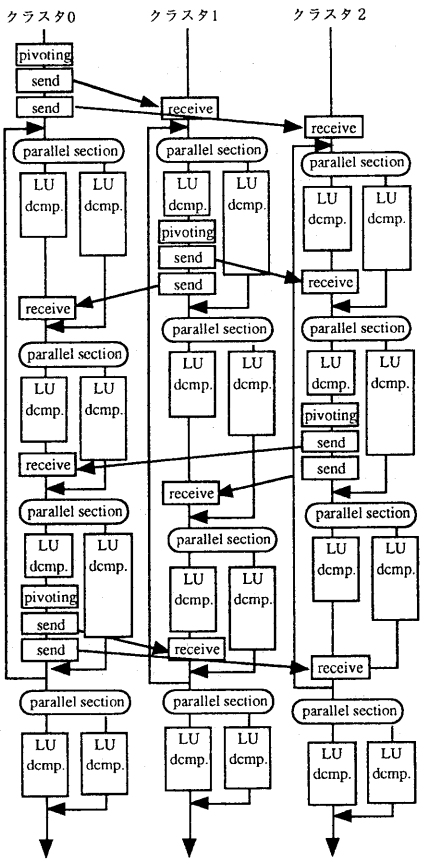


図８　LU分解法のアルゴリズムフローチャート
Fig.8 Algorithm flowchart for LU decomposition.

ＬＵ分解法のアルゴリズムは，１回の反復ステップにおいて，演算量が$O(n^2)$であるのに対して，データ通信量が$O(n)$であり少なくて済む．これが，効率８０％以上を実現した理由であるが，さらに，効率を上げるためには，転送処理の強化として，通信専用プロセサの実現が重要となる．また，今回のＳ－３０００クラスタシステムは転送能力を確保するために，リング型の結合を採用した．ＬＵ分解法のアルゴリズムで発生するデータ通信形態は放送であり，今回，開発したＦＩＦＯ型のデータ転送路に，放送の機能を追加することも重要である．

５．おわりに

スーパコンピュータの高速化のひとつのアプローチであるクラスタ型ベクトル並列スーパコンピュータとして，Ｓ－３０００クラスタシステムを開発した．重要課題であるクラスタ間のデータ転送手段として，排他制御が不要な低オーバヘッドのクラスタ間高速通信機構を実現した．

実測により２階層の並列化によるクラスタ間の協調処理が整然と動作することを確認．また，ＬＵ分解法のプログラムを並列化し，最大性能の８０％以上という高い効率を実現することができた．半性能長ｎ１／２も比較的小さく適用範囲が広いといえる．

今後は，通信専用プロセサの実現，通信オーバヘッドのさらなる低減により，適用応用範囲の拡大をはかっていきたい．

参考文献

[1] T.Utumi, M.Ikeda and M.Takamura：Architecture of the VPP500 Parallel Supercomputer, Supercomputing'94 pp.478-487(1994.11).

[2] Ishii, K., H.Abe, S.Kawabe and M.Hirai：An Overview of the HITACHI S-3800 Series Supercomputer, Proc. of Supercomputer'92, Springer-Verlag (1992).

[3] K.Kitai, T.Isobe, Y.Tanaka, Y.Tamaki, M.Fukagawa, T.Tanaka and Y.Inagami：Parallel Processing Architecture for the Hitachi S-3800 Shared-Memory Vector Multiprocessor, ICS93 (1993.7).

[4] T.Sakakibara, K. Kitai, T. Isobe, S.Yazawa, T.Tanaka, Y. Inagami and Y. Tamaki：Scalable Parallel Memory Archtecture with a Skew Scheme, ICS'93, pp.157-166 (1993.7).

[5] T.Sakakibara, K.Kitai, T. Isobe, S.Yazawa, T.Tanaka, Y. Tamaki and Y. Inagami：An Interprocessor Memory Access Arbitrating Scheme for the S-3800 Vector Supercomputer, ISPAN(1994.12).

[6] Gajski, D., Kuck, D., Lawrie, D. and Sameh, A.：Cedar -A Large Scale Multiiprocessor, ICPP, pp.524-529(1983)

[7] Hwang Kai：Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGrow-Hill, Inc., pp.422-423 (1993).

[8] 田中，浜中，村松：識別子を用いたデータ転送方式を基本とするＭＩＭＤ型並列計算機アーキテクチャ，JSPP'89, pp.115-112 (1989.2).

[9] Dongarra J.J.：Performance of Various Computers Using Standard Linear Equations Software, Comp. Science Dept., Univ. of Tennessee (1994.12).

[10] M.Nakanishi,H.Ina and K.Miura: A High Performance Linear Equation Solver on VPP500 Parallel Super compter, Supercomputing'94, pp.803-810(1994.11).

[11] 上村，清水，石畑，堀江：LINPACKベンチマークの並列ベクトル処理－並列計算機 AP1000 用数値計算アクセラレータによる実現－，JSPP'93，pp.185-192 (1994.5).

[12] Hockney R.W. and C.R.Jesshope: Parallel Computers-2, Architectures, Programming and Algorithms, Adam Hilger Ltd., Bristol, UK and Philadelphia (1988).

表１　LU分解法プログラム実行結果
Table.1Executionresultsfor LU decomposition.

S-3000クラスタシステム						
モデル名	構成 VP数	実効性能 GFLOPS	問題 規模	N1/2	最大性能 GFLOPS	効率 %
model412	12(3×4)	78.2	31120	4880	96	81
model408	8(2×4)	54.1	31200	3760	64	85
model404	4(1×4)	27.2	31200	2680	32	85
model309	9(3×3)	59.0	26940	3180	72	82
model306	6(2×3)	40.9	27000	2400	48	85
model303	3(1×3)	21.5	27000	1560	24	89
model206	6(3×2)	40.6	21600	2160	48	85
model204	4(2×2)	27.9	21600	1640	32	87
model202	2(1×2)	14.5	21600	1100	16	90
S-3800ＴＣＭＰシステム						
モデル名	構成 VP数	実効性能 GFLOPS	問題 規模	N1/2	最大性能 GFLOPS	効率 %
model480	4	28.4	15500	830	32	89
model380	3	21.6	15680	760	24	90
model280	2	14.6	15680	570	16	91
model180	1	7.4	15680	470	8	92

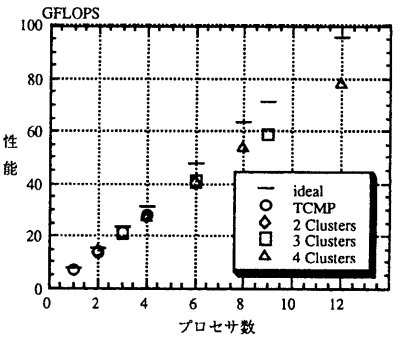


図９　LU分解法による性能評価
Fig.9 Performance for LU decomposition.

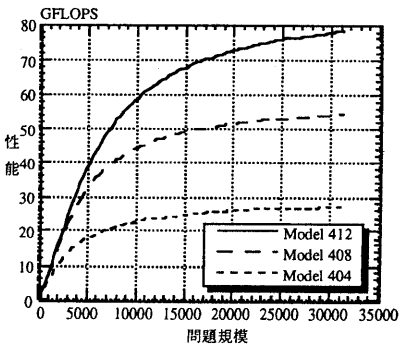


図１０　４クラスタシステムでのLU分解法の性能
Fig.10 Performance for LU decomposition on the 4 cluster system.