TCP/IP通信を用いた並列ネットワーキング方式の検討

北井克佳[†] 吉澤聡[††] マシエルフレデリコ[††]
鍵政豊彦[†††] 稲上泰弘[††]

並列計算機のスケーラビリティを活かした並列ネットワーキング方式について検討した．複数のネットワーク・インタフェースを用いて同一クライアントと双方向通信を可能とするOSの機能拡張により，並列通信による通信性能の向上とインタフェース間の負荷の均一化によるシステム性能の向上を図った．32ノード構成の研究用並列計算機Paradise（Parallel and Data-way Oriented Information Server）を用いて評価した結果，6並列通信で転送量50MB以上の場合には通信性能が5.6倍に向上した．

Research on Parallel Networking for TCP/IP Communication

KATSUYOSHI KITAI,[†] SATOSHI YOSHIZAWA,[††] FREDERICO MACIEL,[††]
TOYOHIKO KAGIMASA[†††] and YASUHIRO INAGAMI[††]

This paper discusses the parallel networking feature that supports performance scalability to the number of network interfaces. On our experimental parallel processor, Paradise (Parallel and Data-way Oriented Information Server), we have developed a new IP routing feature that allows every network interface to communicate to and from the same client processor, thus providing scalable high-performance communication not only for a single session by using multiple network interfaces, but also for total system throughput by balancing the sessions among the network interfaces. The results of performance evaluation, using six Ethernet nodes on Paradise, ATM-LAN and three workstations, has demonstrated the effectiveness of this parallel networking feature.

1. 序　論

インターネット[7]やWorld Wide Web[7]の大衆化，ATMネットワークなどのマルチメディア関連技術の進展，世界市場で年間5000～6000万台のパソコンの売上などの相乗効果によって，電子取引き，電子ショッピング・モール，電子図書館，グループウェアのようなコンピュータ・ネットワークを応用したシステムが脚光を浴びている．ネットワークコンピューティングは，**図1**に示すように，利用者，ネットワーク・プロバイダ，サービス/コンテンツ・プロバイダの3者から実現される．したがって，ネットワークコンピューティング時代のサーバ計算機には，ネットワーキング方式に関する下記3項目の高速化・大容量化が求められる．

(1) 高多重通信（応答時間の短縮）
利用者の増加に対応したシステム性能の向上，QOSを保障したデータ通信数の増加．

(2) 高速通信（データ転送時間の短縮）
単体通信の性能向上や複数のネットワーク・インタフェースを用いた並列通信による通信性能の向上．データウェアハウス構築時のDB全体のロードや障害発生時のデータ回復などで使用．

(3) フォールト・トレランシ（障害時の代替経路）
複数のネットワーク・インタフェースを用いた障害時の切替え．

従来，ネットワーキング方式に関する研究は，主にシングル・プロセッサの単体通信性能の向上を目的としてUNIXやWSの発展とともに推進されてきた．

(a) プロトコル処理の高速化[4),5),20)]
- メモリ・コピー回数の削減などプログラムのチューニングによるOSの高速化
- チェックサムの生成などプロトコル処理のハードウェア化，オフロード化[9)]

† 株式会社日立製作所情報事業企画本部
Information Systems Business Planning Division, Hitachi, Ltd.
†† 株式会社日立製作所中央研究所
Central Research Laboratory, Hitachi, Ltd.
††† 株式会社日立製作所ソフトウェア事業部
Software Division, Hitachi, Ltd.

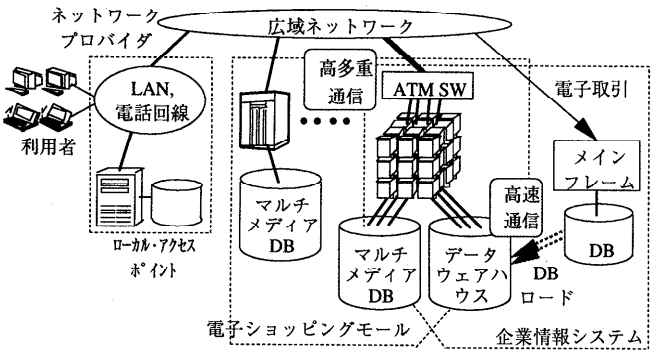


図 1 並列計算機の位置付け
Fig. 1 Role of parallel proccssors on network computing era.

- SMP を利用したプロトコル処理の並列化[14)]

(b) プロトコルの仕様拡張
- ウィンドウサイズ拡張による高速化[10),18)]

(c) 軽量プロトコルやマルチメディア対応プロトコルの提案
- VMTP[15)], XTP[16)] のようなプロトコル処理の軽量化による高速化
- PF_ATM[11)], IPv6[19)], ST-2[17)] のようなATM や QOS に対応した高速プロトコルの提案[12),13)]

本論文では，以上のような単体通信性能の向上ではなく，並列計算機のスケーラビリティに着目して，通信の高速化・大容量化を実現する「並列ネットワーキング方式」を提案する．

並列ネットワーキング方式は，並列計算機の複数ノードにネットワーク・インタフェースを設け，各インタフェースを ATM スイッチや Ethernet スイッチなどのスイッチを介して 1 つのネットワークに接続する．スイッチ型ネットワークでは計算機側の各経路が非共有であり，かつネットワーク側は帯域の大きなネットワークに集線できる．そのため，複数のネットワーク・インタフェースを設けることによってスケーラブルな性能向上が期待できるだけではなく，クライアントが要求する QOS 条件を満たすインタフェースを選択したり，多数のクライアントに対するネットワーキング処理をノード間で動的に負荷分散できる利点がある．本論文では，TCP/IP プロトコル[2)] のようなコネクション指向型通信を対象として，並列ネットワーキング方式について述べる．

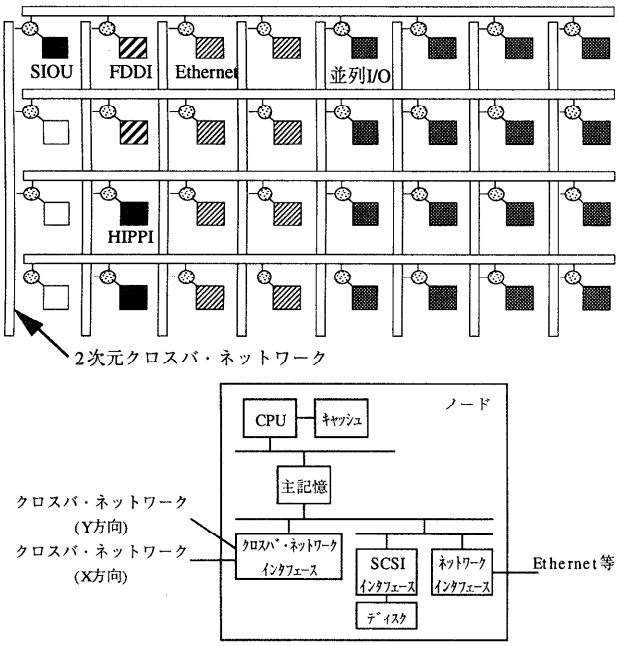
以下，2 章では並列ネットワーキング方式の開発プラットフォームである並列計算機 Paradise のシステム構成について述べる．3 章では，並列ネットワーキング方式の実現上の問題点を述べ，次にその解決方式について述べる．4 章では Paradise を用いた性能評価結果について述べ考察を加える．

2. Paradise のシステム構成と特徴

Paradise（Parallel and Data-way Oriented Information Server）は，ネットワークコンピューティング時代の並列計算機のアーキテクチャ・OS の研究開発を目的として開発した．**図 2** に示すように，Paradise は，32 ノードから構成される主記憶分散型並列計算機であり，各ノードは 2 次元クロスバ・ネットワークで相互に接続されている．各ノードは，PA-RISC アーキテクチャと互換性のある 90 MHz RISC プロセッサ，および，主記憶，ディスク，2 次元クロスバ・ネットワーク・インタフェース，外部ネットワーク・インタフェースから構成される．

図 2 に各ノードの諸元を示す．図 2 に示すように，Paradise の 12 ノードにネットワーク・インタフェースを搭載した．Paradise 全体では，16 本の 10 Mbps Ethernet，2 本の 100 Mbps FDDI，2 本の 1 Gbps HIPPI を備える．また，Paradise は全ノードにディスクを搭載する．その中の 16 ノードには 5 個の SCSI インタフェースを設け，各 SCSI インタフェースにディスクを接続することによって，ソフト RAID や並列 I/O が可能なノード構成とした．

Paradise の OS には，Mach3 をベースにしたシングル・システム・イメージの並列 OS（ParadiseOS）を搭載した．ParadiseOS が実現するシングル・システム・イメージにより，Paradise が主記憶分散型並列計算機であるにもかかわらず，ユーザは 1 台の UNIX WS とまったく同じインタフェースでシステム管理やプログラミングを行える．



ノード	主記憶	SCSI本数	Disk数	ネットワークI/F	備考
	256 MB	1	4	Ethernet x 1	Supervisory IOU
	64 MB	1	2	-	
	256 MB	5	5	-	並列I/O
	64 MB	1	2	Ethernet x 2	
	64 MB	1	2	FDDI x 1	
	256 MB	1	2	HIPPI x 1	

図 2 Paradise のシステム構成
Fig. 2 System configuration of Paradise.

3. 並列ネットワーキング方式

並列ネットワーキング方式は，並列計算機の複数ノードにネットワーク・インタフェースを設けることによって，クライアントが要求する QOS 条件を満たすインタフェースを選択したり，通信処理の負荷を各インタフェースに分散させることを狙いとする．

本論文で述べる並列ネットワーキング方式は，Paradise のようにシステム全体で 1 つの OS を搭載しているシングル・システム・イメージの並列計算機や主記憶共有型マルチプロセッサを対象としている．各ノードに異なる OS を搭載しているクラスタ型の並列計算機には適用できない．

本章では，まず並列ネットワーキング方式実現上の課題について述べ，次にその解決方式を述べる．

3.1 並列ネットワーキング方式の実現上の課題

Paradise などの並列計算機において，複数のネットワーク・インタフェースを用いて高多重通信や高速通信を実現するためには，以下の課題を解決する必要がある．

(1) 同一クライアントとの通信で使用できるネットワーク・インタフェース数の制限解除

サーバがクライアントへ IP パケットを送信する場合，サーバはクライアントの IP アドレスをキーにしてルーティング・テーブルを検索し，ネットワーク・インタフェースと次の送信先アドレス（ゲートウェイ・アドレス）を求める[2),3)]．しかし，現在の IP ルーティング方式では複数のネットワーク・インタフェースの中から 1 つを選択して同一クライアントと通信することは考慮されていない．したがって，図 **3** (a) に示すように，たとえサーバが複数のネットワーク・インタフェースを備えていても，ルーティング・テーブルの検索結果が一意に定まるため，パケットを送出できるインタフェースはただ 1 つに限定される．そのため，任意のネットワーク・インタフェースを用いた通信や，複数のインタフェースを同時に用いた並列通信を実現

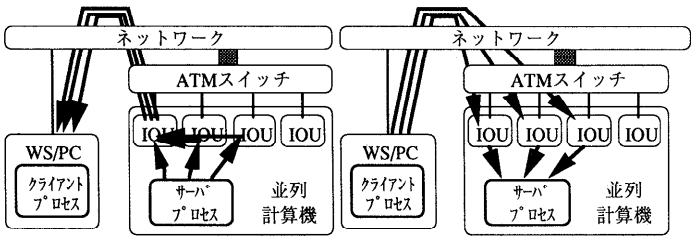


図 3 並列ネットワーキング方式の実現上の問題点
Fig. 3 Issues on achieving parallel networking.

できない☆.

(2) サーバ主導による通信処理の動的負荷分散

クライアント・サーバ・システムのプログラミングでは，クライアントがサーバのネットワーク・インタフェースの IP アドレスを指定してコネクションを確立する[1]．したがって，サーバが複数のネットワーク・インタフェースを備えていても，サーバ側が主体となってインタフェースを選択し，通信量やプロトコル処理の負荷を分散させることができない．

3.2 並列ネットワーキング方式

3.2.1 IP ルーティング方式の変更

3.1 節の第 1 の課題を解決するためには，サーバの IP ルーティング方式を変更して，複数のネットワーク・インタフェースのいずれからでもクライアントへパケットを送出できなければならない．

一般に UNIX OS は TCP/IP のようなコネクション指向型の通信を，(プロトコル名，ローカル・アドレス，ローカル・ポート番号，リモート・アドレス，リモート・ポート番号) の 5 つ組で管理している[1]．サーバ側では，ローカル・アドレスにはクライアントが接続を要求したサーバのネットワーク・インタフェースの IP アドレスが設定され，リモート・アドレスにはクライアントの IP アドレスが設定される．したがって，サーバからクライアントへ IP パケットを送出する ip_output() ルーチン[3] において，ルーティング・テーブルで指定されるネットワーク・インタフェースではなく，5 つ組のローカル・アドレスで示されるインタフェースから IP パケットを送出すれば，クライアントが接続を要求したネットワーク・インタフェースを用いた双方向通信を実現できる．

以上の考察の結果，並列ネットワーキング方式における IP ルーティング方式では，ATM スイッチなどを介して 1 つのネットワークに接続されるインタフェースに関しては，(a) 次の送信先アドレスは従来どおりルーティング・テーブルの検索結果を使用し，(b) ネットワーク・インタフェースは 5 つ組で指定されるローカル・アドレスを使用することとした．その結果，クライアントが接続を要求した任意のネットワーク・インタフェースを用いて，サーバからクライアントへ IP パケットを送出できるようにした．

3.2.2 動的負荷分散方式

3.1 節の第 2 の課題を解決するためには，ネットワーク・インタフェースの負荷に応じてサーバが動的にインタフェースを選択できる必要がある．

OS は，ネットワーク・インタフェースごとのコネクション数や入出力パケット数 (Ipkts, Opkts) などの統計情報，各コネクションの状態などを管理している[3]．これらの情報はユーザ・プロセスから直接読み出すことができるため，コネクション数や Ipkts, Opkts の増加率によって各インタフェースの負荷を観測できる．

そこで，並列ネットワーキング方式では，負荷分散方式の第 1 ステップとして，サーバがクライアントからの接続要求を受けたときに，Established 状態[2),3)] になっているコネクション数を調べ，コネクション数が最小のネットワーク・インタフェースをクライアントに通知することによって，ネットワーク・インタフェースの負荷分散を図ることとした．また，コネクション数が同一の場合には若番のインタフェースを選択することとした．

3.2.3 コネクション確立方式

図 4 に並列ネットワーキング方式におけるコネクション確立方式の処理フローを示す．

図 4 に示すように，まずクライアントはサーバに対して，接続要求と要求 QOS を送る (Step (1))．QOS

☆ クライアントからサーバへのパケット送出は，クライアントがインタフェースを指定するため，図 3 (b) に示すように任意のインタフェースと通信できる．

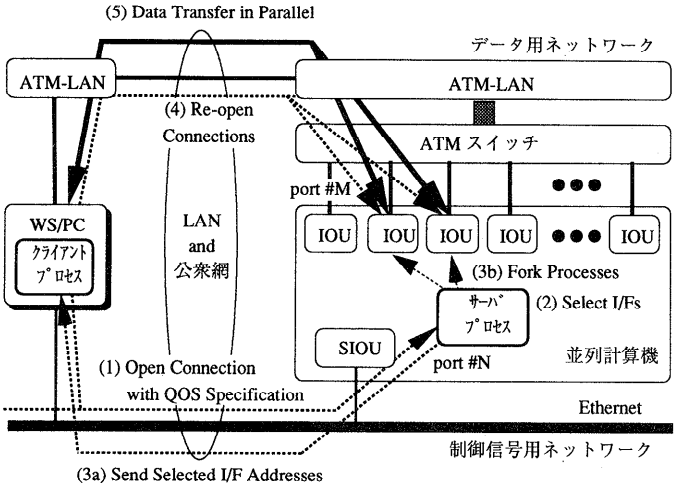


図 4 並列ネットワーキング方式の処理フロー
Fig. 4 Processing steps of parallel networking.

パラメタには並列通信の並列度を指定する☆．サーバは，Established 状態になっているコネクション数を調べ，要求 QOS を満たすネットワーク・インタフェースを選択する（Step (2)）．サーバは選択したインタフェースのアドレスとデータ通信に使用するポート番号をクライアントに通知する（Step (3a)）．クライアントは通知されたネットワーク・インタフェースに対して通信路を再度確立する（Step (4)）．複数のネットワーク・インタフェースを同時に使用する並列通信の場合には，並列通信数だけの通信路を確立する．通信路を再確立した後，データ通信を行う（Step (5)）．

3.2.4 並列通信方式

複数のネットワーク・インタフェースを用いた並列通信では，送信側は指定されたデータを等分割して転送し，受信側は分割されたデータを受信した後，1 つのデータにマージする．サーバ側では，ネットワーク・インタフェースを備える各ノードに子プロセスを Fork することによってネットワーキング処理の負荷分散を図り，クライアント側ではシングル・プロセスで実現することによってプロセス間通信やプロセス・スイッチが生じないようにした．

3.2.5 新設関数

表 1 にクライアント用とサーバ用の新設関数を示す．新設関数は，アプリケーションを容易に記述できることを目標として，以下のように仕様を定義した．

(1) 従来の TCP/IP 通信と同じ関数呼び出し手順で並列ネットワーキング方式を利用できること．

通信路の確立には，従来の connect，accept 関数の代りに p_connect，p_accept 関数を用いることとし，データ通信には，従来の read，write 関数の代わりに p_read，p_write 関数を用いることとした．

(2) 並列ネットワーキングに必要なすべての処理を新設関数の内部処理に隠蔽すること．

p_connect，p_accept 関数の仕様を，図 4 の Step (1) から Step (4) が両関数の内部処理となるように定義し，p_read，p_write 関数の仕様を，3.2.4 項に示した内容が両関数の内部処理となるように定義した．

4. 並列ネットワーキング方式の評価

本章では，並列ネットワーキング方式を用いて並列通信と高多重通信を行った場合について評価した結果と考察について述べる．

4.1 評価条件

4.1.1 評価システムとソフトウェアのバージョン

並列計算機 Paradise を用いて並列ネットワーキング方式を評価した．**図 5** に方式評価システムの構成とソフトウェアのバージョンを示す．図 5 において，Paradise をサーバ，H9000 WS 3 台をクライアントとして使用し，Paradise は IOU 6 ノードと 6 本の 10 Mbps Ethernet を使用した．

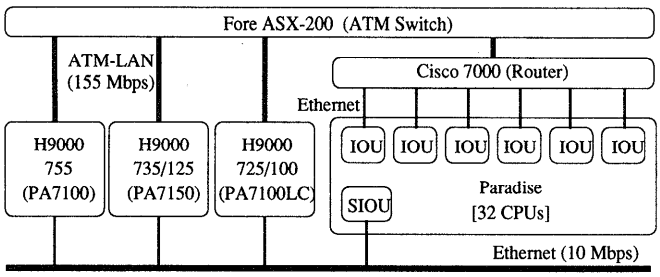
4.1.2 性能測定プログラムと仮定条件

図 6 に性能測定に用いたプログラムの概略を示す．接続要求は Ethernet 経由で H9000 から Paradise へ送り，データは ATM-LAN 経由で Paradise から H9000 へ転送する．本プログラムでは，以下の仮定を設け，

☆ 将来は，ATM Forum の UNI 4.0，P-NNI が規定する QOS サービス・クラスに適応されるトラフィックや QOS に関するパラメタ[6] を指定可能とする．

表 1 並列ネットワーキング方式用関数
Table 1 Library functions for parallel networking.

	関数名，パラメタ	処理の概要
クライアント用関数	p_connect(int sockfd, struct sockaddr_in *serv_addr, int servlen, struct qos qos_value, int *p_sockfd_list)	(1) サーバ (serv_addr) と制御用コネクションを開設 (2) 要求 QOS（コネクション数）をサーバへ SEND (3) データ通信用のサーバの IP アドレスとポート番号を RECV (4) データ通信用のソケット (sockfd_list) をコネクション数個生成 (5) (4) で RECV した IP アドレスとデータ用コネクションを開設
	p_read/p_write(int *p_sockfd_list, char *buf, int nbytes, int p_num)	(1) read/write 長 (nbytes) をコネクション数 (p_num) で等分割して， 各 p_sockfd_list[i] の担当 r/w 長，buf 領域を決定 (2) select() で r/w 可能な p_sockfd_list[i] を選択して r/w 処理実行 (3) すべての r/w 終了まで，(2) を繰り返す
サーバ用関数	p_accept(int sockfd, struct sockaddr_in *peer_addr, int addrlen, int p_sockfd, struct sockaddr_in *p_peer_addr, int *p_sockfd_list, int *p_num, int *node_index)	(1) peer_addr でクライアントの制御用コネクション開設要求を accept (2) 要求 QOS をクライアントから RECV (3) クライアントの要求 QOS と各インタフェースの負荷を見て， ネットワーク・インタフェースを選択（node_index） (4) インタフェースの IP アドレスとポート番号を SEND (5) p_peer_addr でデータ用コネクション開設要求を accept
	p_read/p_write(int *p_sockfd_list, char *buf, int nbytes, int p_num, int my_pid, int *pid_list)	(1) read/write 長（nbytes）をコネクション数（p_num）で等分割して， 各 p_sockfd_list[i] の担当 r/w 長，buf 領域を決定 (2) 各子プロセスは自分が担当する buf 領域を read/write



装置	バージョン，等
日立 H9000	HP-UX 9.05
Fore ASX-200	Ver. 3.0.1
Cisco 7000	Ver. 10.3.1
Paradise-H9000 間接続	PVC接続， Classical IP over ATM

図 5 方式評価に用いたシステム構成とソフトウェア
Fig. 5 System configuration and software versions used on preformance evaluation.

単一ユーザ環境で通信性能を測定した．

(1) 送信元の転送データは，ディスク上ではなく主記憶上にあると仮定．
(2) ソケットバッファ長（TCP ウィンドウサイズ）：56 KB．
(3) TCP セグメント長：1460 バイト．
(4) 並列通信：H9000/755 と Paradise の間で転送．転送単位ごとにデータを分割・マージする．
(5) 高多重通信：全 H9000 で複数プロセスを生成し各プロセスが性能測定プログラムを実行する．3.2.2 項で述べた動的負荷分散方式により Ethernet 間の負荷均一化を図る．並列通信は含まない．

4.2 性能評価結果と考察

4.2.1 並列通信（データ転送性能）

並列通信の効果を調べるため，並列通信数と総データ転送量（転送単位 × 転送回数）を変化させてデータ転送速度を測定した．転送単位の上限値はソケットバッファ長となるため，転送回数によって総データ転送量を変化させた．また，前回の転送単位の通信と今回の転送単位の通信がオーバラップすることも含めて，総データ転送量を変化させた場合の並列通信の効果

クライアント (H9000)	サーバ (Paradise)

gettimeofday(& 接続開始時間);
fd = socket(AF_INET, SOCK_STREAM, 0);
p_connect(並列通信数);
gettimeofday(& 接続終了時間);
	fd = socket(AF_INET, SOCK_STREAM, 0);
	bind(fd); listen(fd);
	p_accept(並列通信数);

転送単位，転送回数を送信;	転送単位，転送回数を受信;

gettimeofday(& データ転送開始時間);
p_write(転送単位, 並列通信数);
for (i=0; i< 転送回数; i++)
p_read(転送単位, 並列通信数);
gettimeofday(& データ転送終了時間);
	fork(); /* 子プロセスの生成 */
	p_read(転送単位, 並列通信数);
	for (i=0; i< 転送回数; i++)
	p_write(転送単位, 並列通信数);
	waitpid();

(注1) データ転送速度 (Mbps) = 転送単位×転送回数 / (データ転送終了時間－開始時間)
(注2) TCPのスロー・スタート制御のため，転送回数が増えればデータ転送性能は向上
(注3) p_accept()で割当た累積fd数がOPEN_MAX-16になると，signal(SIGCHLD)をONにして使用済のfdをclose()する。

図 6 性能測定プログラム
Fig. 6 Program used on performance evaluation.

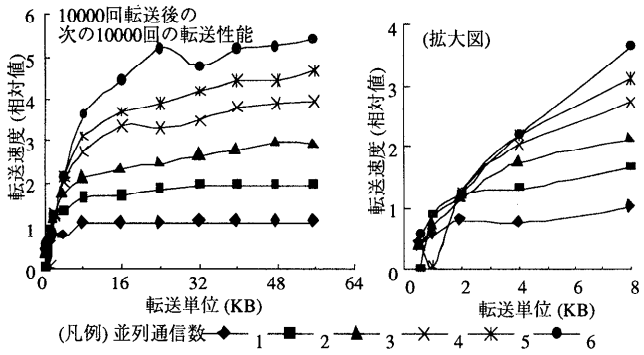


図 7 転送単位を変化させた場合の並列通信の効果
Fig. 7 Effectiveness of parallel communication according to the unit of data transfer.

を求めた．並列通信のセットアップ時間や TCP のスロースタート制御[2] の影響を排除するため，図 6 において，gettimeofday（& データ転送開始時間）と gettimeofday（& データ転送終了時間）の範囲で，10000 回転送後の次の 10000 回の転送時間を測定した．**図 7** に結果を示す．

図 7 において，縦軸の基準値 '1' は，Paradise における 10 Mbps Ethernet の最大データ転送速度（実測値）を示す．転送単位が 2 KB 以下では各通信路の転送データ量が少なくなることによる送受信処理時間の相対的増加や，受信側でデータを 1 つにマージする処理のために，並列通信の効果は見られない．しかし，転送単位が 4 KB 以上では並列通信数が 1～4，転送単位が 8 KB 以上では並列通信数 1～6 のすべてで並列通信の効果が見られる．転送単位が 8 KB 以上では並列通信数に応じてスケーラブルに性能が向上することが分かる．

4.2.2 並列通信（セットアップを含めたデータ転送性能）

並列通信のセットアップ処理の影響を測定するために，転送単位を最大値（56 KB）に固定し，総データ転送量（転送回数）を変化させてデータ転送速度を測定した．図 6 において，gettimeofday（& 接続開始時間）と gettimeofday（& 接続終了時間）の範囲と，gettimeofday（& データ転送開始時間）と gettimeofday（& データ転送終了時間）の範囲を合計した時間を測定した．**図 8** に結果を示す．縦軸の基準値 '1' は，図 7 と同様に，Paradise における 10 Mbps Ethernet の最大データ転送速度（実測値）を示す．

並列通信数 1 の場合と並列転送を行わない場合〔従来方式（single）〕を比較すると，総転送量によらず両者のデータ転送速度はほぼ等しいことが分かる．これより，並列ネットワーキング方式ではコネクションの再接続など並列通信のセットアップを行うが，転送性能には影響を及ぼさないことが分かる．

転送速度が飽和する総転送量を比較すると，並列通

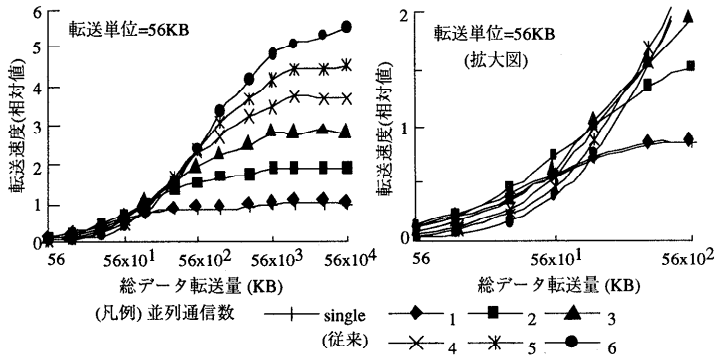


図 8 総データ転送量を変化させた場合の並列通信の効果
Fig. 8 Effectiveness of parallel communication according to total amount of data transfer.

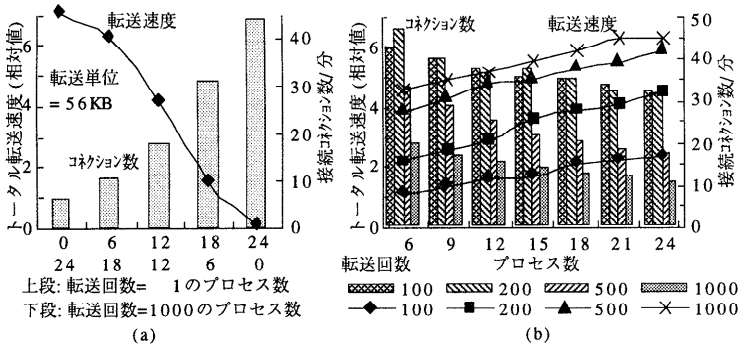


図 9 転送速度と接続コネクション数の変化
Fig. 9 Variation of total transfer ratio and number of established connections.

信を行わない場合には 1 MB（転送回数 = 20）程度の総転送量であるのに対して，並列通信数が 6 の場合には 50 MB 以上（転送回数 >1000）の総転送量を要することが分かる．このように，並列通信では転送単位ごとにデータの分割・マージ処理を行うため，転送速度が飽和するためには，データ転送処理とデータの分割・マージ処理がオーバラップするだけの総転送量が必要であることが分かる．

以上のように，本並列ネットワーキング方式を用いた並列通信は，50 MB を超える中・大規模なデータを転送する場合に有効であるといえる．

4.2.3 高多重通信

3 台の H9000 上にそれぞれ複数個のクライアント・プロセスを生成して，高多重通信に対する並列ネットワーキング方式の効果を評価した．各プロセスには図 6 の性能評価プログラムを繰り返し実行させた．転送単位や転送回数の組合せを変えることによって，サーバに対するコネクションの接続要求頻度や，各プロセスのデータ転送時間の長短の組合せを変化させた．

表 2，**表 3** は転送単位と転送回数の組合せを示し，**図 9** は全クライアントのトータル・データ転送速度（システム性能）と 1 分あたりに処理された接続要求数を示す．トータル・データ転送速度の基準値 '1' は，図 7，図 8 と同様に，Paradise における 10 Mbps Ethernet の最大データ転送速度（実測値）を示す．

図 9 (a) に転送単位と総プロセス数を固定し，転送回数が 1000 回のプロセス数と転送回数が 1 回のプロセス数の組合せを変化させた場合の結果を示す．転送回数が 1000 回のプロセスは動画や音声のような転送時間の長いデータ転送を想定し，転送回数が 1 回のプロセスはメニュー検索のような転送時間の短いデータ転送を想定している．

図 9 (a) から分かるように，転送回数が 1000 回の通信のみの場合には，通信の多重化と，3.2.2 項で述べた動的負荷分散による 6 本の Ethernet 間の負荷均一化によって，全プロセスのデータ転送速度の合計は，6 並列通信よりも高い値が得られた．一方，転送回数が 1 回の通信が増加すると接続要求の処理量は増加するが，トータル・データ転送速度は低下する．これは，各プロセスのデータ転送量が少ないためだけではなく，

表 2 図 9 (a) の測定条件：転送単位の組合せ
Table 2 Parameters used on experiments of Fig. 9 (a).

		3 台の H9000 の各々で生成するプロセス数の内訳				
転送単位	56 KB × 1	0	2 x 3	4 x 3	6 x 3	8 x 3
× 転送回数	56 KB × 1000	8 x 3	6 x 3	4 x 3	2 x 3	0

表 3 図 9 (b) の測定条件：転送単位の組合せ
Table 3 Parameters used on experiments of Fig. 9 (b).

		3 台の H9000 の各々で生成するプロセス数の内訳						
		6	9	12	15	18	21	24
転送単位	8 KB	1 x 3	1 x 3	1 x 3	1 x 3	1 x 3	1 x 3	1 x 3
	16 KB	1 x 3	1 x 3	1 x 3	1 x 3	1 x 3	1 x 3	1 x 3
	24 KB		1 x 3	1 x 3	1 x 3	1 x 3	1 x 3	1 x 3
	32 KB			1 x 3	1 x 3	1 x 3	1 x 3	1 x 3
	40 KB				1 x 3	1 x 3	1 x 3	1 x 3
	48 KB					1 x 3	1 x 3	1 x 3
	56 KB						1 x 3	1 x 3
	64 KB							1 x 3

OS が同時に受付け可能な accept 関数のキューに溢れが生じたことも転送速度の低下の原因となった．

図 9 (b) に，転送単位と総プロセス数の様々な組合せに対して，転送回数によってデータ転送時間の長短を変化させた場合の結果を示す．図 9 (a) の測定結果と同様に，複数のネットワーク・インタフェースを用いた高多重通信では，転送回数が多いプロセスが多いほどトータル・データ転送速度が向上する．また，転送単位の大きなプロセスが多いほどトータル・データ転送速度が向上することが分かる．

5. 関連研究

並列通信用プロトコルとして，National Storage Lab., Lawrence Livermore National Lab., IBM, UCB などが開発している PTP (Parallel Transfer Protocol)[8] が提案されている．PTP は，ペタバイト・オーダの高性能マス・ストレージ・システムにおける，ストレージ装置間，ストレージ装置と WS 間で並列データ転送するためのプロトコルである．PTP は，TCP/IP のような一般プロトコルの上位でも動作するため，本論文で提案した並列ネットワーキング方式の上位に位置する．したがって，本並列ネットワーキング方式を実現したうえで PTP を動作させることによって，たとえば，マス・ストレージ・システムと並列計算機間で高速な並列データ転送を実現できる．

6. 結　論

スケーラブルな通信性能を実現する並列計算機の並列ネットワーキング方式について述べた．

OS のコネクション管理テーブルを用いて IP パケットを送出する新 IP ルーティング方式と，ネットワーク・インタフェース情報を用いて動的負荷分散を図るコネクション確立方式を提案した．複数のネットワーク・インタフェースを用いて同一クライアントと双方向通信を可能としたことによって，並列通信による通信性能の向上とインタフェース間の負荷の均一化によるシステム性能の向上を実現した．

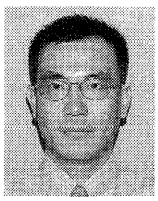
並列計算機 Paradise と H9000 WS 3 台を用いた性能評価結果により，並列ネットワーキング方式の有効性を示した．

参考文献

1) Stevens, W.R.: *UNIX Network Programming*, Prentice Hall (1990).
2) Comer, D.E.: *Internetworking with TCP/IP*, Vol.I, II, III, Second Edition, Prentice Hall (1991).
3) Leffler, S.J., McKusick, M.K., Karels, M.J. and Quarterman, J.S.: *The Design and Implementation of the 4.3 BSD UNIX Operating System*, Addison-Wesley (1989).
4) Partridge, C.: *Gigabit Networking*, Addison-Wesley (1994).
5) Tantawy, A.N. (Ed): *High Performance Networks – Frontiers and Experience*, Kluwer Academic Publishers (1994).
6) Alles, A.: ATM インタネットワーキング，日経 BP 出版センター (1995).
7) Krol, E.: *The Whole Internet – User's Guide & Catalog*, O'Reilly & Associates (1994).
8) Berdahl, L.: Parallel Transport Protocol Proposal (Draft), Lawrence Livermore National Laboratory (1995).

9) UNCL Technical Report, No.1 (1994).
10) 長谷川，長谷川，加藤，鈴木：広域 ATM 網を介した LAN 間接続のための TCP ゲートウェイの実装，信学技報 (1995).
11) Kandlur, D.D., Saha, D. and Willebeek-LeMair, M.: Protocol Architecture for Multimedia Applications over ATM Networks, *Computer Comm. Review*, ACM SIGCOMM (1995).
12) Coulson, G., Campbell, A., Robin, P., Blair, G., Papathomas, M. and Hutchison, D.: The Design of a QoS Controlled ATM Based Communications System in Chorus, ftp.comp.lancs.ac.uk (1994).
13) Sharma, R. and Keshav, S: Signalling and Operating System Support for Native-Mode ATM Applications, *SIGCOMM '94* (1994).
14) Nahum, E.M., Yates, D.J., Kurose, J.F. and Towsley, D.: Performance Issues in Parallelized Network Protocols, *USENIX Winter* (1994).
15) Cheriton, D.R. and Williamson, C.L.: VMTP as the Transport Layer for High-Performance Distributed Systems, *IEEE Communication Magazine*, Vol.27, No.6 (1989).
16) Strayer, W.T., Dempsey, B.J. and Weaver, A.C.: *XTP: The Xpress Transfer Protocol*, Addison-Wesley (1992).
17) Partridge, C. and Pink, S: *An Implementation of the Revised Internet Stream Protocol (ST-2)*, John Wiley & Sons (1992).
18) Nicholson, A., Golio, J., Borman, D.A., Young, J. and Roiger, W.: High Speed Networking at Cray Research, *Computer Communication Review*, ACM SIGCOMM (1991).
19) Brandner, S.: The Recommendation for the IP Next Generation Protocol, RFC1752 (1995).
20) Steenkiste, P.A.: A Systematic Approach to Host Interface Design for High-Speed Networks, *IEEE Computer* (1994).

(平成 8 年 9 月 11 日受付)
(平成 10 年 9 月 7 日採録)

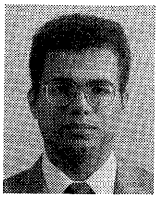


北井 克佳（正会員）

1961 年生．1984 年京都大学工学部情報工学科卒業．1986 年同大学院工学研究科情報工学専攻修士課程修了．同年，(株) 日立製作所中央研究所入所．現在，(株) 日立製作所情報事業企画本部勤務．現在は，新事業分野製品の企画（主に，IC カードシステム）に従事．IEEE，ACM 各会員．

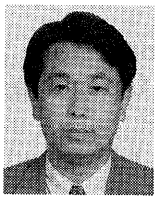
吉澤 聡（正会員）

1961 年生．1985 年カリフォルニア工科大学修士課程修了．1986 年，(株) 日立製作所中央研究所入所．現在，ネットワーク・アーキテクチャの研究に従事．IEEE，ACM 各会員．



マシエル フレデリコ

1966 年生．1988 年ブラジル航空工科大学（ITA）機械工学科卒業．1990 年同大学院情報工学専攻修士課程修了．1994 北海道大学工学研究科博士後期課程修了．同年，(株) 日立製作所中央研究所入所．現在，高性能コンピューティングの研究開発に従事．工学博士．IEEE，電子情報通信学会各会員．



鍵政 豊彦（正会員）

1955 年生．1978 年大阪大学基礎工学部情報工学科卒業．1980 年同大学院基礎工学研究科情報工学分野修士課程修了．同年，(株) 日立製作所中央研究所入所．現在，(株) 日立製作所ソフトウェア事業部勤務．(株) 日立製作所においては，メインフレーム OS，超並列計算機 OS などの OS 研究開発に従事．ACM，IEEE コンピュータソサイアティ，電子情報通信学会各会員．

稲上 泰弘（正会員）

1953 年生．1977 年京都大学工学部情報工学科卒業．1979 年同大学院工学研究科情報工学専攻修士課程修了．同年，(株) 日立製作所中央研究所入所．現在，高性能プロセッサシステム，高性能ストレージシステムの研究マネージメントに従事．IEEE，ACM，電子情報通信学会各会員．