B-013

スイッチを用いないトーラス接続 PC クラスタの実装と評価
Evaluation and Implementation of Torus Network PC Cluster not using Switches

福永 隆文†
Takafumi Fukunaga

1．まえがき

高い通信性能が要求される PC クラスタでは 1 リンクあたり 10Gbps を超える高速スイッチ（Infiniband [1] 等）を用いてフルバイセクションの Fat Tree 構造が用いられるが，多数の高価なスイッチが全体のコストを大きく上昇させる．一方，近年のマルチコア技術は PC 自体にネットワーク負荷を負担させることを可能にした．今回，スイッチを用いずに PC ノードをトーラス型に直接接続し，複数経路を同時に用いることで低価格，高バンド幅を実現する PC クラスタを提案する．10G Ethernet を用いた 3 次元のトーラス接続 PC クラスタが目標であるが，機器の制限により Gigabit Ethernet，2 次元トーラス接続を用いて実装，評価した．

2．提案方式の概要

提案方式の仕組みは 3 次元のトーラス接続の例を用いて説明する．図 1 は 3x3x3 の 27 台の PC ノード（図中○）が 3 次元のトーラス型に相互に接続されている様子である．各ノードは上下方向，横方向，奥行き方向にリング状に通信ケーブルで接続されている．それぞれの方向へ展開するノード数は 3 に限定されず N 台の構成が可能であり，その場合ノード数は NxNxN となる．また，次数も 3 次元に限定されず 2 次元，4 次元に縮小，拡張できる．拡張する場合，各ノードで必要となる通信ポート数は増加する．

図では送信ノード S から受信ノード R への通信経路を 3 つ（経路 1～3）示してあるが，3 次元では送信ノードに 6 つのケーブルが接続されているため最大 6 つの通信経路が利用できる．送信ノードの出力ケーブルが決まれば中継ノードは最短経路しか選ばない仕組みなので中継ノードから先の経路が複数に分かれることはない．複数の経路を選ぶ機能は最初の送信ノードのみが持つ．図の太実線で示した経路 1 は送信ノード S が A, B を中継ノードとして受信ノード R にデータを送る経路である．ノードの周りの数字は通信ポート番号である．S は事前に用意された出力ポートテーブルに従い，3 番ポートからデータを出力している．ケーブルの終端はノード A の 1 番ポートにつながっており，A は一旦データリンク層レベルで受信するが中継ノードであり受信ノードへの最短経路と登録された 2 番ポートから出力している．データは A, B 間ケーブル（太実線）を通り B へ送られる．実は A から受信ノードまでの最短経路は B を通過する経路以外に E を通過する経路もある．この例に限らず複数の最短経路が存在する組み合わせは数多く存在するが，どちらを選択するかは出力ポートテーブル作成者へ委ねられる．全体のバランスを考えながら作成する必要がある．中継ノード B は最短経路の 4 番ポートから出力しデータは受信ノード R へ到達する．図では 3 つの経路は全て中継ノード数 2 の比較的最短な経路となるが，バンド幅を向上させるため最短ではない経路も利用する．例えば C, D, E を中継ノードとする経路も利用できる．送信ノードは全ての通信ポートを同時に利用してバンド幅を向上しようとする．ただし，遅延性能の悪化を緩和するため比較的最短の経路を優先して利用する．

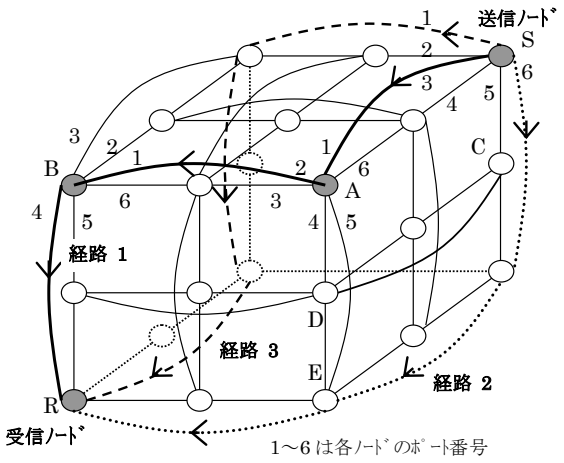

図 1　3 次元トーラス接続の例

3．提案方式の実装

提案システム実装のため送信ノードに出力ポート選択処理，中継ノードに転送処理を追加した．最終的な受信ノードでは通常の受信処理となり修正は必要ない．

3.1　送信ノードの出力ポート選択処理

出力ポート選択には複数ケーブルを用いた通信負荷分散方式 IEEE802.3ad やラウンドロビン送信を実装している Linux 標準の Bonding モジュールを修正し利用した．ロードされた修正済みの Bonding モジュールはストリームの開始時に複数の Ethernet ポートから 1 つを選択し，そのストリームに結びつける．Bonding のラウンドロビン送信に見られるような同一ストリーム内パケットの分散は行わないため out-of-order は発生しない．また，結びつけるストリームは高バンド幅を要求するストリームのみとした．低速なストリームは最短経路に固定した．高バンド幅を要求するかどうかの判定には送信バッファ内に残る“ack による受信確認ができていないデータ量”を利用した．高バンド幅と判定されたストリームは事前に手動で用意した出力ポートテーブルに従ってポートが割り当てられる．送信が完了（Fin パケット処理）すると，その結びつき情報は削除される．テーブルは送信ノードごと，受信ノードごとに作成する必要がある．下記は図 1 の送信ノー

† 熊本県立技術短期大学校

ドS内のテーブルの例である．受信するノードごとに1行記述するため26行の記述が必要となる．

（送信先識別コード）（出力ポート番号）

ﾉｰﾄﾞR向け　00:15:17:bf:56:e0　3 1 6 2 4 5 3 1 6 1

ﾉｰﾄﾞC向け　00:15:17:ab:5d:3a　5 6 1 2 3 4 5 6 5 6

送信先識別コードは各 PC の 1 つめの通信ポートの MAC アドレスであり，送信パケット内のヘッダ送信先 MAC アドレスと比較することでテーブル内の該当エントリを検索する．システム内の転送はデータリンク層での転送なのでヘッダ内 MAC アドレスは最終目的地の MAC アドレスを持つ．各 PC の ARP テーブルは事前に手動で作成した．なお，各 PC の特定のポートをスイッチに接続し，ARP などのブロードキャストはスイッチを用いる実装も組み込んでいるため ARP 利用も可能である．最初の行は図1の受信ノードR向けの行である．最短経路は 1, 3, 6 番ポートからの出力となるが，3 を選択しポート番号列の1番目に記述した．1, 3, 6 は最短ではない 2, 4, 5 よりも多用している．2行目は図1のノードCが受信ノードとなる場合のポート選択を記述してある．最短経路は直接接続されている5番ポートであり1つ目に記述する．6は中継ノード数1となり2番目に近い経路となるため5とともに多用した．テーブルを作る際にはループが発生しないよう留意する．例えば図 1 のノード B が受信ノード R 向けデータを 1 番ポートから出力するとノード A 内テーブルに R 向け最短経路は 2 番ポートと記述してあるためノードBにデータを送り返すループとなる．

Ethernet フレームを出力する直前にヘッダ内の送信先 MAC アドレス，送信元 MAC アドレスをそれぞれ受信ポート，出力ポートのアドレスに書き換える．受信ポートの MAC アドレスは出力ポートごとに決まっているので事前に用意してテーブルと同じタイミングで Bonding モジュールに読み込ませておく．出力ポートの MAC アドレスは net_device 構造体からコピーする．

3.2　中継ノードの転送処理

中継の様子を図2に示す．図は送信ノードまたは複数の中継が行われる場合の先行中継ノードからのデータを2番ポートで受信し，最短経路として3番ポートから出力する様子である．受信する NIC のドライバに修正を加える必要がある．データを受信したドライバは上位層へ上げるためにアンマップなどの処理を行うが，この際に送信先 IP アドレスと自 IP アドレスを比較して中継を行うべきか上位層に上げるべきかの中継判定処理を行う．転送を要する場合にはドライバ内で転送のためのマーキングを行う．マーキングは IP ヘッダ内の TTL を 1 減じることとした．システム内通信はデータリンクレベルのため TTL が変化することはなく，マーキングとして利用可能である．

中継ノードでも送信処理時に Bonding モジュールが関わるため，モジュール内でマーキング判定を行う．転送データについては最短経路へ出力し，転送ではなく初めてノードから出力するデータに対しては先に述べた出力ポート選択処理を実行する．

4．評価

Gigabit Ethernet 環境，Quad-Core Opteron 2.4GHz×2，16GB メモリ，Linux-2.6.32 (CentOS 6.4)搭載ノードを用いた．ハードウェアの制限から 9 台を用いた 2 次元トーラス

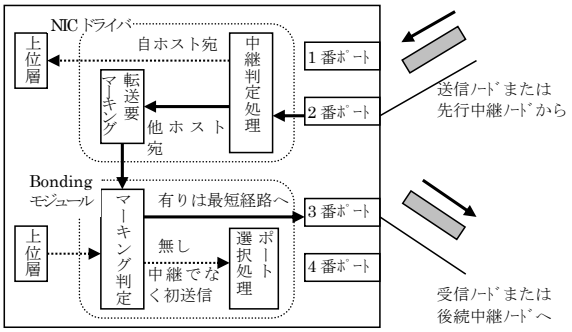

図 2　転送処理

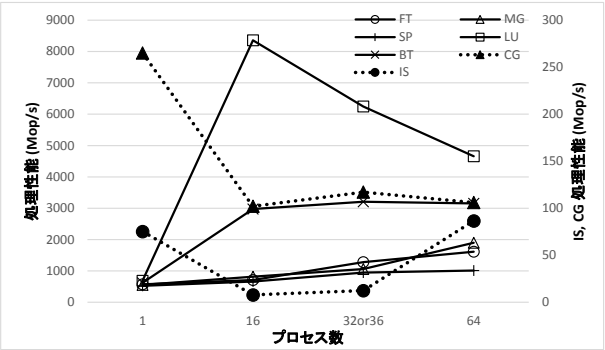

図 3　並列処理性能

接続にて実験を行った．図1の最前面に描かれているノード9台のシステム構成となる．片方向バンド幅については利用できるケーブル数（2 次元では最大 4）に比例した性能向上が確認できた．ハードウェア性能約 940Mbps×ケーブル数となる．ただし複数ノードが同時に通信を行う場合はケーブルを共有するため性能は下がる．

また，NPB3.3 を用いて並列処理性能を測定した．測定結果を図 3 に示す．システム全体で 9 ノード×8 コア=72 コアのため MAX を 64 プロセスとした．例えば 36 プロセス時は各ノードで 4 コア動作させた．CG を除いて並列の効果が確認できる．特に LU は 16 プロセスにて 12.1 倍の性能向上を示した．LU は計算処理がボトルネックとなりやすいため効果が大きいと考えられる．スイッチを用いたスター型システムに対する速度向上度は 32or36 プロセス実行時，FT 1.17, MG 1.03, SP 1.09, LU 1.05, BT 1.09, CG 1.09, IS 1.04 倍とわずかながら向上している．

5．まとめ

スイッチを用いず PC を直接トーラス状に接続するクラスタを実装し簡単な評価を行った．結果，スイッチを用いたシステムよりわずかであるが良い性能を示した．また，アプリケーションにも依存するがプロセス数の増加によって並列処理性能も向上した．現在，機器の制限から 2 次元システムであるが，3 次元とすべく準備中である．出力ポートテーブルの作成方法，本方式が効果を発揮しやすいアプリケーションの型分析などこれからの課題は多い．

参考文献

1)　InfiniBandTM Architecture Specification, http://www.infinibandta.org, InfiniBand Trade Association 2004.