

2Z-2

HiTactix/Symbioseの開発(2)

— OS接続モジュール Symbiose を用いた機能分散型並列 OS の設計と実装 —

竹内 理, 岩崎正明, 中原雅彦, 田口しほ子, 中野隆裕, 川田容子, 児玉昇司
(株)日立製作所 システム開発研究所

1. はじめに

近年ネットワークハードウェアが高速化し、オペレーティングシステムの通信処理性能向上に関する研究が盛んに行なわれている[1].

従来、通信処理性能向上を実現する一手法として、SMPシステム上における対称型マルチプロセッシング方式が用いられてきた。しかしこの方式では、ロック競合問題[2]による通信処理性能の劣化が発生する。例えば、2個以上のCPU上での並列実行によるTCP/IP通信のスループットは、1CPU上での逐次実行によるスループットより下回る、という報告がある[3].

我々は、ロック競合問題を解決する機能分散型マルチプロセッシング方式を採用したHiTactix/Symbioseを実装し、その通信処理性能を評価した。本稿では、HiTactix/Symbioseの概要と、その性能評価結果につき述べる。

2. HiTactix/Symbioseの基本構成

HiTactix/Symbioseは、SMPハードウェア上で1つのCPUに1つのOS (HiTactix[1,4]) を割り当てる。各OSは機能分散により並列動作する。Symbioseは、OS間での連携処理時に必要となる機能 (OS間通信機能, OS間同期機能, OS間

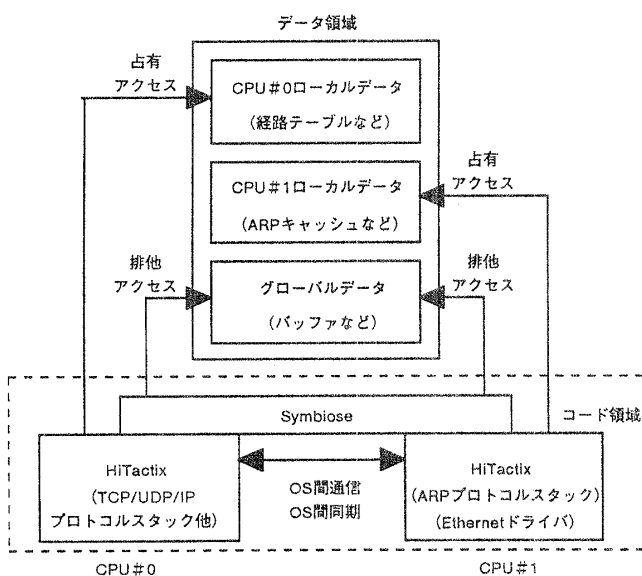


図1 HiTactix/Symbioseの基本構成

A Development of HiTactix/Symbiose(2)

— A Design and Implementation of a Functionally Distributed Operating System Using OS connecting Module Symbiose Tadashi Takeuchi, Masaaki Iwasaki, Masahiko Nakahara, Shihoko Taguchi, Takahiro Nakano, Youko Kawata, Shouji Kodama

Systems Development Laboratory, Hitachi, Ltd.

で共有するグローバルデータへの排他アクセス機能など)を提供するカーネルライブラリ群である。

2CPU上で動作するHiTactix/Symbioseの基本構成を図1に示す。

- 1) CPU#0上のHiTactixでは、TCP/UDP/IPプロトコルスタックが動作する。CPU#1上のHiTactixでは、ARPプロトコルスタックやEthernetドライバが動作する(機能分散型マルチプロセッシングを実現する)。
- 2) データ領域を、各CPU上のHiTactixが占有アクセスするローカルデータ領域と、全HiTactix間で排他アクセスするグローバルデータ領域に分割する。
- 3) 各CPU上のHiTactixは、グローバルデータへのアクセスやOS間通信、OS間同期を実行する際のみSymbioseを呼び出し、スピンドックを保持する。対称型マルチプロセッシング方式と比して、スピンドックを保持する頻度が低下する(ローカルデータへのアクセスの際にはスピンドックを保持する必要がない)ため、ロック競合問題を回避可能となる。

3. 課題と解決策

2節で述べた基本構成にてHiTactix/Symbioseを実装するため、以下の課題を解決した。

- 1) 複数HiTactixの連携ブートの実現
- 2) 共有デバイスの管理
- 3) OS間共有バッファプールの実現
- 4) OS間通信, OS間同期の一括化

本節では、上記4)の課題の概要とその解決策につき述べる。

OS間通信, およびOS間同期は、CPUが提供する Atomic Read/Write 命令及びプロセッサ間割り込み命令を用いて実現する。これらの命令実行には数十~数百マシンサイクルを要する。IP通信処理性能を高く保つためには、OS間でバッファを授受する際に要するOS間通信やOS間同期の実行回数を減らす必要がある。

上記課題を解決するため、バッファ授受は図2に示す方式で実現した。各OSに5msごとに周期駆動する一括送信スレッド及びポーリングスレッドを配置する。また、各スレッドは自OSで占有するローカルキューとOS間で共有するグローバルキューとの間で、複数バッファを一括移動す

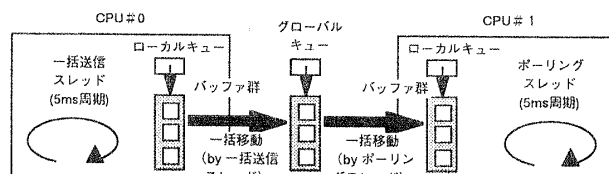


図2 複数バッファの一括移動

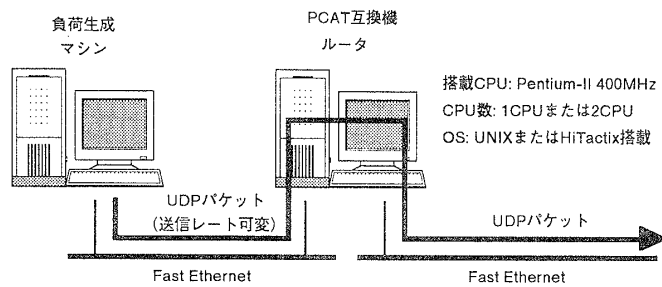


図3 実験システム

る。本方式に従うと、OS間通信（キュー操作）の実行回数は5msにつき2回になる。またOS間同期を実行する必要はない。

4. 通信処理性能評価

HiTactix/Symbioseの通信処理性能評価を実現するため、図3に示す実験システムを構築して性能測定を行なった。以下、実験システムの概要と測定結果につき述べる。

実験システムの概要は以下の通りである。

- 1) PCAT互換機 (Pentium^{注1} II400MHz × 2 搭載) を Fast Ethernetルータとして動作させる。搭載OSとして以下の4種類を準備した。
 - UNIX^{注2}のSingle CPU版とDual CPU版
 - HiTactixのSingle CPU版とDual CPU版 (HiTactixのDual CPU版とは、2CPU上で動作するHiTactix/Symbioseを指す。)
- 2) 負荷生成マシンから上記各ルータにUDPパケットを送信する。負荷生成マシンから送信するUDPパケットの送信レートを変化させた際の、ルータのCPU負荷、及び単位時間あたりの出力パケット数の変動を測定する。

測定結果を図4に示す。図4の横軸はルータに対する単位時間あたりの入力パケット数を、縦軸はルータの1CPUあたりの平均CPU負荷及び単位時間あたりの出力パケット数を示す。図4から以下がわかる。

- A) UNIX (Dual CPU版) 搭載ルータはUNIX (Single CPU版) 搭載ルータよりルーティング処理性能が劣化している。1CPUあたりの平均CPU負荷は55～60%程度上昇し、かつ出力パケット数も低減している。
- B) HiTactix (Dual CPU版) 搭載ルータは、HiTactix (Single CPU版) 搭載ルータと比してルーティング処理性能が向上している。1CPUあたりのCPU負荷が43%程度低減している。
- C) UNIX (Dual CPU版) 搭載ルータは1CPUあたりの平均CPU負荷が50%以上に上がらない(カーネルモードにて走行できるCPU数が高々一つであるため)。HiTactix (Dual CPU版) には上記上限は存在しない。

5. まとめ

本稿では、機能分散型マルチプロセッシング方式により、SMPハードウェア上でロック競合問題を解決しつつ通信処理性能向上を実現するHiTactix/Symbioseの概要につき述べ

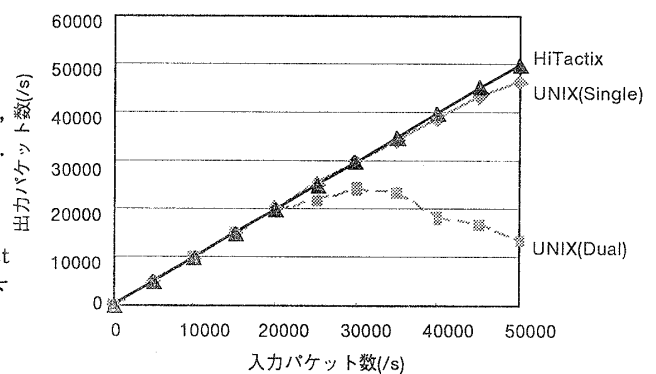
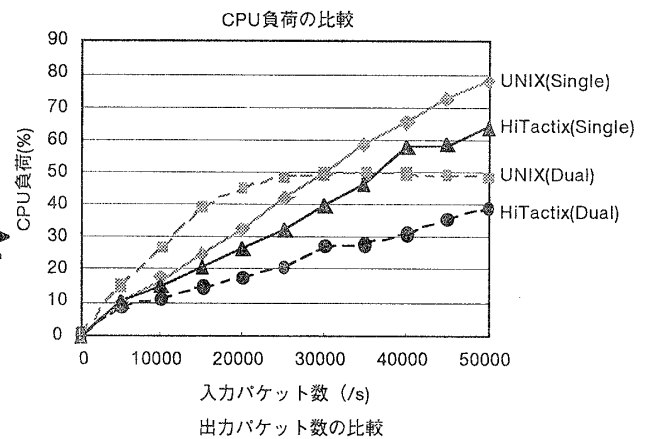


図4 性能評価結果

た。かつその性能評価を行ない、2CPU上で並列実行するHiTactix/Symbioseのルーティング処理性能が、1CPU上で逐次実行するHiTactixと比して約43%向上することを確認した。

6. 今後の課題

Symbioseを用いて、SMPハードウェア上でUNIXとHiTactixを連動させる予定である。本システムはUNIXとの完全互換性を維持しつつ、通信処理性能の向上、及び(HiTactixが提供する)新規QoS保証機能[1,4]を同時に実現する。

参考文献

- [1] 竹内理他, 「アイソクロナススケジューラを応用したQoS保証型ルーティング方式の設計と実装」, マルチメディア通信と分散処理ワークショップ, pp119～126, Oct. 1998.
- [2] 岩崎正明他, 「密結合マルチプロセッサにおける排他制御処理オーバーヘッドの解析的一評価手法」, 情報処理学会論文誌, Vol. 31, No. 11, pp1627～1635, Nov. 1990.
- [3] 住元真司他, 「Gigabit Ethernet NICの性能評価」, 情報処理学会研究報告, 99-OS-80, pp49～54, Mar. 1999.
- [4] 竹内理他, 「連続メディア処理向きOSの周期駆動保証機構の設計と実装」, 情報処理学会論文誌, Vol. 40, No. 3, pp1204～1215, Mar. 1999.

注1: PentiumはIntel Corporationの登録商標です。

注2: UNIXは米国X/Open Company Ltd.の登録商標です。