システムソフトウェアと
オペレーティング・システム 83－6
（2000．2．25）

OS接続モジュールSymbioseを用いた BSD-HiTactix連動システムの設計と実装

竹内 理　岩嵜正明　中原雅彦　田口しほ子　中野隆裕　川田容子

（株）日立製作所システム開発研究所

近年、インターネットを介した連続メディアデータ転送がさかんになり、高速通信機能とQoS保証機能を提供するOSに対するニーズが高くなってきた。しかし、現在まで上記機能を提供すべく開発されたOSは、汎用OSとの互換性がないため普及しなかった。

本稿では、汎用OSとの互換性を保ちつつ上記機能を提供するBSD-HiTactix連動システムの概要と、その通信性能およびQoS保証性能の評価結果につき述べる。評価の結果、本システムは既存の汎用OSと比して、100～4000倍程度のQoS保証性能の向上、及び44%程度の通信性能の向上を実現可能であることを確認した。

A Design and Implementation of BSD-HiTactix Cooperating System Using OS Connecting Module *Symbiose*

Tadashi Takeuchi　Masaaki Iwasaki　Masahiko Nakahara　Shihoko Taguchi
Takahiro Nakano　Youko Kawata

Systems Development Laboratory, Hitachi Ltd.

Recently, necessity of operating systems providing high-speed network I/O and QoS asuurable communications is becomming high, because continuous media data is often tranferred through the internet. However, no operating systems providing mechanisms described above has not become widespread, because it lacks compatibility with a popular operating system.

This paper gives outlines of *BSD-HiTactix Cooperating Systems*. *BSD-HiTactix Cooperating Sysmtems* can provide not only mechanisms describe above but also comptibility with BSD/OS. This paper also gives evaluation results of its network I/O performance and QoS assurable communications. We confirmed that *BSD-HiTactix Cooperating Sysmtems* can provide 44 percent faster network I/O compared with BSD/OS. We also confirmed that it can provide from 100 times to 4000 times as strict QoS assurable communications as those of BSD/OS.

1　はじめに

近年、ネットワークハードウェア技術の進歩に伴ない、WDM（Wavelength Division Multiplexing）技術を用いた光ファイバ網やギガビットイーサネットなどの高速ネットワークが数多く出現してきた。これらのネットワークは、圧縮されたビデオデータを転送するのに十分な転送能力を備える[8]。これに伴ない、ビデオデータや音声データを配送するストリームサーバや、上記データのルーティング処理を実行するルータ向けに、高い通信性能と通信のQoS保証機能を備えたOSが新規に必要になってきている[5]。

一方、近年アプリケーションの規模及び種類が肥大化してきた。これに伴ない、上記に示した新規OSも、UNIXなどの既存の汎用OSとの互換性を維持する必要が高くなってきた[9]。既存の汎用OSとの互換性を維持しなければ、膨大な開発コストをかけて、新規OS上で動作するアプリケーションや開発環境を新たに実装する必要が生じる。

このように、

- 高い通信性能
- 通信のQoS保証機能
- 汎用OSとの互換性

を兼ね備えたOSに対する要求は高まりつつあるが、現在のところ上記をすべて兼ね備えたOSは存在しない。例えば、Rialto、Nemesis、RoadRunnerなどは、高い通信性能や通信のQoS保証機能を備えているが汎用OSとの互換性はない[3, 4, 5]。マイクロカーネル方式、ナノカーネル方式を用いた複数OSの連動システムは、汎用OSと

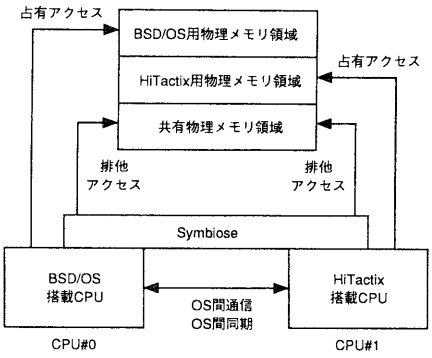

図 1: システム構成の概要

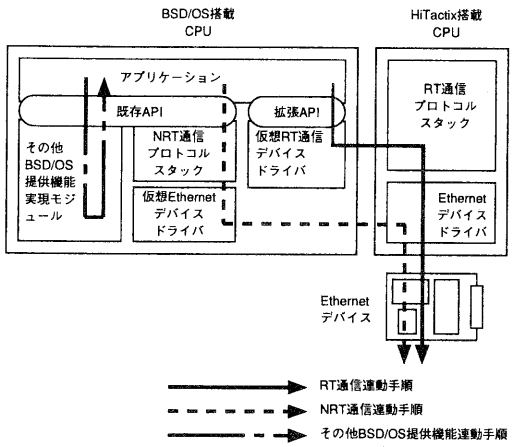

図 2: ソフトウェアモジュール構成

の互換性を実現しているが、ギガビットイーサネットをはじめとする高速ネットワークに追従可能な通信性能を提供できていない [1, 9]。

我々は上記すべてを充足可能な BSD-HiTactix 連動システムを新規に提案する。本システムは、SMP ハードウェアの 1CPU 上にて BSD/OS[1] （既存の汎用 OS）を、別の CPU 上にて HiTactix[2, 6, 7, 10, 11]（連続メディア処理向けマイクロカーネル）を動作させる。BSD/OS が動作する CPU が存在することにより、本システムは「汎用 OS との互換性」を維持できる。また、HiTactix が動作する CPU が存在することにより、「高い通信性能」「通信の QoS 保証機能」を提供できる。

本稿では、上記 BSD-HiTactix 連動システムの概要と、その通信性能及び QoS 保証性能の定量的な評価結果につき述べる。

2 システム概要

本節では BSD-HiTactix 連動システムの概要につき示す。

2.1 システム構成の概要

BSD-HiTactix 連動システムのシステム構成の概要を図 1 に示す。

- BSD-HiTactix 連動システムは SMP ハードウェア上で動作する。1CPU 上で BSD/OS が、別の CPU 上で HiTactix が動作する（物理メモリ内に BSD/OS と HiTactix のコードが共存する）。BSD/OS と HiTactix は機能分散による並列動作を行なう。
- BSD/OS 及び HiTactix は、それぞれ物理メモリ上に自 OS 用の物理メモリ領域を保持する。各 OS は自 OS 用の物理メモリ領域を占有可能であるため、シングル CPU 上で動作する場合と同様に動作できる。

[1] BSD/OS は BSDI 社の商標です。

- BSD/OS と HiTactix は Symbiose を用いて連動可能である。Symbiose は、OS 間での連動処理時に必要となる機能（OS 間通信機能、OS 間同期機能、OS 間で共有する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。Symbiose は物理メモリ上の共有物理メモリ領域を用いて動作する。

2.2 ソフトウェアモジュール構成の概要

本節では、BSD-HiTactix 連動システムのソフトウェアモジュール構成の概要を示し、

- RT（RealTime）通信（QoS を保証可能な通信）
- NRT（Non-RealTime）通信
- その他の BSD/OS 提供機能

を実現する BSD-HiTactix 連動方式の概要を明らかにする。

BSD-HiTactix 連動システムのソフトウェアモジュール構成の概要を図 2 に示す。

図 2 に示すように、HiTactix には、RT 通信プロトコルスタック、Ethernet デバイスドライバが組み込まれている。一方 BSD/OS には、シングル CPU 用の BSD/OS にも組み込まれているモジュール（NRT 通信プロトコルスタック、その他の BSD/OS 提供機能実現モジュール）の他に、仮想デバイスドライバが組み込まれている。仮想デバイスドライバとは、HiTactix に組み込まれているモジュールを仮想デバイスと見立て、該当モジュールに対する通信要求のフォワードや該当モジュールからの通信結果の受理（仮想デバイスに対する入出力）を行なうモジュールである。

BSD-HiTactix 連動システムでは以下の手順にて RT 通信を実現する。

1. アプリケーションが、拡張 API（アプリケーションインタフェース）を用いて RT 通信要求を発行する。
2. 仮想 RT 通信デバイスドライバは上記要求を受理し、RT 通信プロトコルスタックに対して通信要求をフォワードする。
3. RT 通信プロトコルスタックが Ethernet デバイスドライバと連動して、アプリケーションからの要求に応じた RT 通信を実行する。

上記手順に従うことにより、HiTactix が提供する周期スケジューラ（アイソクロナススケジューラ [10]）、シグナリングプロトコル（RTIPSIG[11]、TTCP[2]）、送信抑制モジュール（ITM[2]）、高速通信機能 [6] を用いて RT 通信を実現できる。上記は HiTactix のみが動作する CPU 上で動作するため、シングル CPU 用の HiTactix と同程度の「通信の QoS 保証機能」及び「高い通信性能」を BSD-HiTactix 連動システムは提供可能である。

一方、BSD-HiTactix 連動システムが NRT 通信を実現する際には以下の手順に従う。

1. アプリケーションが、既存 API を用いて NRT 通信要求を発行する。NRT 通信プロトコルスタックはアプリケーションからの要求に応じた NRT 通信を実行後、仮想 Ethernet デバイスドライバに対して送受信要求を発行する。
2. 仮想 Ethernet デバイスドライバは受理した送受信要求を Ethernet デバイスドライバにフォワードする。
3. Ethernet デバイスドライバは受理した送受信要求に従い、パケットの送受信を実行する。

また、BSD-HiTactix 連動システムがその他の BSD/OS 提供機能を実現する際には、BSD/OS 搭載 CPU のみにて処理を実行する。

NRT 通信やその他の BSD/OS 提供機能を実現する際に使用する API は、既存 API と完全に一致する。すなわち、BSD-HiTactix 連動システムは「汎用 OS との互換性」を実現している。

3 実装上の課題と解決策

2 節で示した BSD-HiTactix 連動システムを実装するにあたり、以下の課題を解決する必要が生じた。

- 複数 OS のブート
- デバイスの割り当て
- 共有デバイスへの排他アクセス制御の実現

本節では、上記課題の概要とその解決策につき順に述べる。

3.1 複数 OS のブート

BSD-HiTactix 連動システムはブート時に、

- BSD/OS、HiTactix 用の物理メモリ領域、及び共有物理メモリ領域を確保し、
- BSD/OS、HiTactix、Symbiose のコードを、上記各領域にローディングし、
- BSD/OS、HiTactix の各 OS が動作する仮想アドレス空間に共有物理メモリ領域をマップする（各 OS から Symbiose のコードを関数呼び出し可能にする）

必要がある。BSD-HiTactix 連動システムでは上記ブートを図 3 に示す手順にて実現した。

1. 1CPU 上で BSD/OS をブートする。本ブートは、シングル CPU 用の BSD/OS のブートローダを用いて実現する。設定ファイル[2] に、BSD/OS が使用可能な物理メモリ量を実物理メモリ量より小さく指定することにより、BSD/OS のコードが物理アドレス空間の下位領域（BSD/OS 用物理メモリ領域）にローディングされ、かつ、上記領域のみが BSD/OS カーネル仮想アドレス空間にマッピングされることを保証できる（図 3 中の Step1 参照）。
2. BSD/OS のシェルからコマンドを入力し、Symbiose 及び HiTactix のローディング要求を発行する。BSD/OS は、HiTactix 及び Symbiose のコードを HiTactix 用物理メモリ領域及び共有物理メモリ領域にローディングする。さらに BSD/OS は、自 OS の仮想アドレス空間に共有物理メモリ領域をマップする（図 3 中の Step2 参照）。
3. BSD/OS は、物理メモリの固定番地に、HiTactix 用物理メモリ領域、及び共有物理メモリ領域に属する物理ページ情報（図 3 中では「HiTactix 用物理メモリ情報」「共有物理メモリ情報」と表記）を格納する。上記格納後、他方の CPU を起動し、他方の CPU 上で HiTactix のブートコードの動作を開始させる。
4. HiTactix のブートコードは、上記物理ページ情報をもとに、自 OS のコード及び共有物理メモリ領域を自 OS の仮想アドレス空間にマップする（図 3 中の Step3 参照）。

3.2 デバイスの割り当て

BSD-HiTactix 連動システムでは、割り込みコントローラを除く各デバイスを BSD/OS または HiTactix のどちらかに割り当てる（割り込みコントローラの制御方法については 3.3 節を参照すること）。各 OS は割り当てられたデバイスのみに占有アクセスする。

上記の実現には、以下の課題を解決する必要がある。

- 各 OS が、割り当てられているデバイスのみを検出、初期化すること。
- 各デバイスからの外部割り込みが該当デバイスを占有アクセスする OS のみに通知されるべく、外部割り込みがルーティングされていること。

BSD-HiTactix 連動システムは、以下の手順にてブート時にデバイス割り当てを実行することにより、上記課

[2] /etc/boot.default

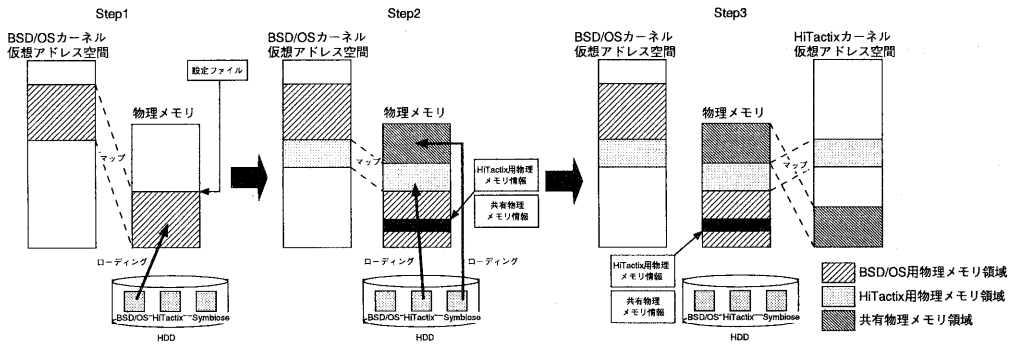

図 3: ブート手順

題を解決している。

1. BSD/OS はブート時に、ユーザが指定するデバイス割り当て情報を設定ファイルから読み込む。BSD/OS は、自 OS に割り当てられたデバイスのみを検出、及び初期化する。さらに、割り込みルーティングも仮初期化する（すべての外部割り込みを自 OS が動作する CPU 側にルーティングする）。本手順により SCSI コントローラまたは IDE コントローラも初期化され、HiTactix 及び Symbiose のコードのローディングが可能になる。
2. BSD/OS は、Symbiose のコードのローディング、及び共有物理メモリ領域のマップ処理が完了したら（図3中の Step2 までブート処理が進行したら）、Symbiose の初期化ルーチンを呼び出す。
3. Symbiose の初期化ルーチンは、システムに接続されている全デバイスの検出を行なう。さらにデバイス割り当て情報に従い、割り込みルーティングの設定を行なう。
4. HiTactix はブート時に自 OS に割り当てられているデバイス一覧を Symbiose から取得する。そして上記一覧に属するデバイスの検出、及び初期化を実行する。

3.3 共有デバイスへの排他アクセス制御の実現

BSD/OS と HiTactix が共にアクセスする必要のあるデバイス（現実装では割り込みコントローラのみ）は、各 OS からのアクセスが排他的であることを保証する必要がある。BSD-HiTactix 連動システムでは、図 4 に示すように、排他デバイスアクセスモジュールを用いて上記課題を解決している。

BSD/OS 及び HiTactix 内のデバイスアクセスを実行するモジュールは、デバイスに直接アクセスする代わりに、排他デバイスアクセスモジュールに対してデバイスアクセス要求を発行する。排他デバイスアクセスモジュールは上記要求を排他的に受理し（スピンロックを用いて

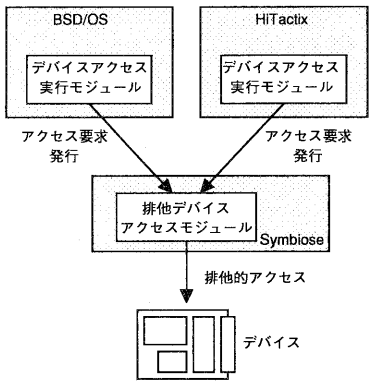

図 4: 排他デバイスアクセスモジュール

実現）、要求に応じてデバイスアクセスを実行する。

4 性能評価

本節では、BSD-HiTactix 連動システムの

- 通信の QoS 保証性能
- 通信性能

の定量的評価をするために行なった実験の概要と、その結果につき示す。

4.1 通信の QoS 保証性能の評価

4.1.1 評価実験概要

本評価実験は、図 5 に示す構成にて実施した。

1. 表 1 に示したマシン仕様を持つ送受信ノードを Fast Ethernet にて接続する。送信ノードには BSD-HiTactix 連動システムまたは BSD/OS（シングル CPU 用）を搭載する。

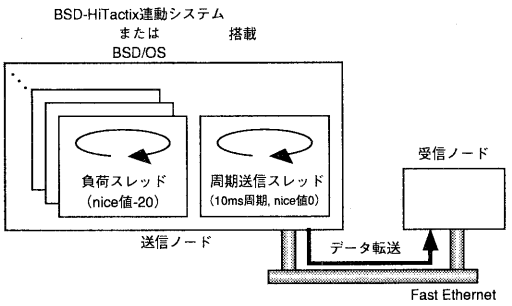

図 5: QoS 保証性能評価実験システム

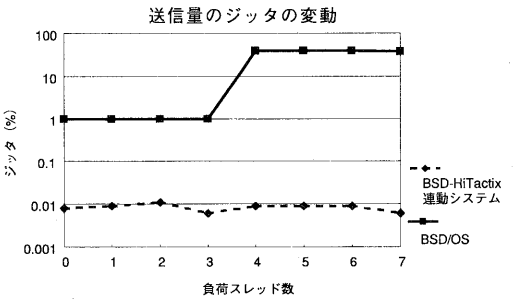

図 6: 送信量のジッタの変動

表 1: マシン仕様

CPU	PentiumII 400MHz(x2) 搭載
Ethernet NIC	DC21143 チップ搭載

2. 送信ノード上にて以下のスレッドを動作させる。
   - 周期送信スレッド

     10ms 周期で周期駆動し、データ長が 1280Byte である UDP パケットを 60 パケットずつ（ヘッダ部を含めた送信レートは 62.784Mpbs）受信ノード宛てに送信する。
   - 負荷スレッド

     nice 値-20 で無限ループを実行する。

3. 負荷スレッドの個数を 0〜7 個に変化させた際の、周期送信スレッドが送信するパケット送信数のゆらぎを測定することにより、BSD-HiTactix 連動システムと BSD/OS の QoS 保証性能を比較した。パケット送信数のゆらぎは、受信ノードに到達するパケット数を 40ms ごとに集計することにより測定した。

4.1.2 評価実験結果

前節で示した実験の結果を図 6 に示す。図中のグラフの横軸は負荷スレッドの個数を、縦軸がパケット送信量のジッタの平均を示している。ここで言うパケット送信量のジッタとは、以下の式により求まる値を指す（式中の N は、40ms の間に受信ノードに到達したパケット数を示す）。

$$（送信量のジッタ）= \mid N - 240 \mid \times 100 \div 240 \quad (\%)$$

グラフから、BSD-HiTactix 連動システムは BSD/OS と比して、送信量のジッタを 1/100〜1/4000 に低減していることがわかる。すなわち、BSD-HiTactix 連動システムは BSD/OS と比して、100〜4000 倍の通信の QoS 保証性能を有している。

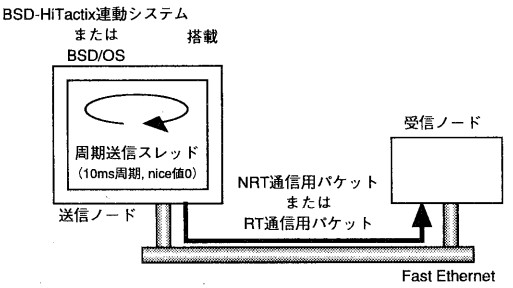

図 7: 通信性能評価実験システム

4.2 通信性能の評価

4.2.1 評価実験概要

本評価実験は、図 7 に示す構成にて実施した。

1. 表 1 に示したマシン仕様を持つ送受信ノードを Fast Ethernet にて接続する。送信ノードには BSD-HiTactix 連動システムまたは BSD/OS（シングル CPU 用）を搭載する。
2. 送信ノード上にて以下に示す周期送信スレッドを動作させる。本周期送信スレッドは、10ms 周期で周期駆動し、データ長が 1280Byte である NRT 通信パケットまたは RT 通信パケットを一定数ずつ受信ノード宛てに送信する。
3. 周期送信スレッドが 1 周期あたりに送信するパケット数を変動させた際の、送信ノードの CPU 負荷の変動を測定した。測定は表 2 に示すケースにつき行ない、BSD-HiTactix 連動システムと BSD/OS の通信性能を比較した。

4.2.2 評価実験結果

前節で示した実験の結果を図 8 に示す。図中のグラフの横軸は RT 通信パケットまたは NRT 通信パケットの送

表 2: 測定ケース

	搭載 OS	送信パケット
Case1	BSD-HiTactix 連動システム	RT 通信パケット
Case2	BSD/OS（シングル CPU 用）	NRT 通信パケット

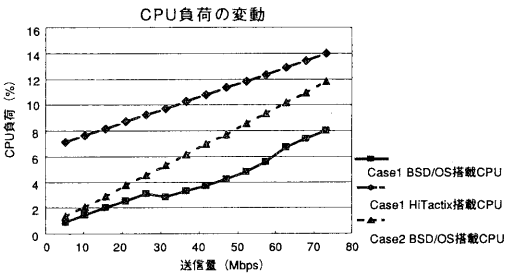

図 8: CPU 負荷の変動

信レートを、縦軸は CPU 負荷を示す。グラフには、

- 表 2 に示す Case1 の BSD/OS 搭載 CPU
- 表 2 に示す Case1 の HiTactix 搭載 CPU
- 表 2 に示す Case2 の BSD/OS 搭載 CPU

における CPU 負荷の変動がプロットしてある。

グラフ上にプロットしてあるデータを一次近似し、最大送信レート（CPU 負荷が 100%に達するレート）を算出した。算出結果を表 3 に示す。

上記算出結果から、BSD-HiTactix 連動システムは BSD/OS と比して 44%程度高い通信性能を有していることがわかる。

5 まとめ

本稿では、高い通信性能、通信の QoS 保証機能、汎用 OS との互換性を同時に提供する BSD-HiTactix 連動システムの概要と、その通信の QoS 保証性能及び通信性能の定量的な評価結果につき述べた。評価の結果、本システムは、BSD/OS と比して、100～4000 倍程度の QoS 保証性能の向上、及び 44%程度の通信性能の向上を実現可能であることを確認した。

表 3: 最大送信レート

測定ケース	CPU	最大送信レート
Case1	BSD/OS 搭載 CPU	999.5Mbps
Case1	HiTactix 搭載 CPU	930.2Mbps
Case2	BSD/OS 搭載 CPU	645.7Mbps

参考文献

[1] Helander, J.: UNIX under Mach – The LITES Server, Master's thesis, Helsinki University of Technology, 1994.

[2] Iwasaki, M., Takeuchi, T., Nakano, T. and Nakahara, M.: Isochronous Scheduling and its Application to Traffic Control, *19th IEEE Real-Time System Symposium*, (1998), pp.14–25.

[3] Jones, M. B., Roşu, D. and Roşu, M.-C.: CPU Reservations and Time Constraints: Efficient, Predictable Scheduling of Independent Activities, *Symposium on Operating Systems Principles*, (1997), pp.198–211.

[4] Leslie, I., McAuley, D., Black, R. and Roscoe, T.: The Design and Implementation of an Operating System to Support Distributed Multimedia Applications, *IEEE Journal on Selected Area s in Communications*, Vol. 14 (1996), pp.1280–1297.

[5] Miller, F. W., Keleher, P. and Tripathi, S. K.: General Data Streaming, *19th IEEE Real-Time System Symposium*, (1998), pp.232–241.

[6] 岩嵜正明ほか：連続メディア処理向きマイクロカーネルの開発（1）～（5），第 53 回全国大会講演論文集（1），(1996), pp.141–150.

[7] 岩嵜正明ほか：HiTactix/Symbiose の開発（1）～（7），第 59 回全国大会講演論文集（1），(1999), pp.125–138.

[8] 菊地隆裕:T ビット／秒に突入 – 2001 年のインターネット基幹網 –, 日系エレクトロニクス, (1998), pp.93–113.

[9] 新井利明ほか：ナノカーネルによる異種 OS 共存技術「DARMA」の提案, 第 59 回全国大会講演論文集 (1), (1999), pp.139–140.

[10] 竹内理ほか：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 (1998), pp.1204–1215.

[11] 竹内理ほか：アイソクロナススケジューラを応用した QoS 保証型通信の設計と 実装, 情報処理学会論文誌, Vol. 40 (1999), pp.3737–3751.