

## HiTactix/Symbiose の開発（5）

2 Z - 5

—Dynamic Gateway への応用—

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

## 1. はじめに

インターネット利用者が急増し, また, Webシステムを用いた電子商取引や音楽のリアルタイム配送などの新サービスも登場している. これに伴って, サービスを提供するサーバやゲートウェイ (ProxyサーバやFirewallなど) には, 高い処理性能, セキュリティ保証, QoS保証を同時に満足することが要求されている. しかし, これら互いに矛盾する要求を同時に満足することは, 現状技術では困難である.

我々は, この課題を解決し, 高性能, セキュリティ保証, QoS保証を同時に満足する通信サーバ (Dynamic Gateway, 以下DGと略す) の開発を推進している. 本稿では, このDGの実現方式について述べる.

## 2. 既存技術の限界

サーバやゲートウェイを高性能化する方法として, いくつかの既存技術が存在する. ここでは, これらの既存技術の問題点について述べる.

## (1) 対称型マルチプロセッシング

シングルプロセッサの性能不足を補う方法として, 対称型マルチプロセッシング技術がある. しかし, 対称型マルチプロセッシング技術は, Webサーバのような入出力が中心となるアプリケーションにおいては, プロセッサ数による性能向上効果が小さい. 例えば, 入出力処理中心のベンチマークであるSPECweb96では, CPU 4個<sup>注1</sup>を用いてもCPU 1個の場合と比べて1.44倍しか性能が向上しない[6]. これは, 入出力中心のアプリケーションではシステムコールが頻繁に発生し, OS内部の共有データをアクセスする際に排他制御が必要になり, 処理の並列性が損なわれるためである.

## (2) ハードウェア・パケット処理

サーバ負荷分散やセキュリティを目的とした, 専用ASICによるL3スイッチ/フィルタなどのハードウェア・パケット処理技術は, ソフトウェア処理に比べて高速処理が可能である. また, ハードウェアを拡張し, L4~L7のスイッチングやフィルタリングを実現する製品も開発されている. しかし, L4~L7スイッチングにおいてはすべての機能をハードウェア化することは困難であり, ソフトウェア部分を実行するプロセッサの処理能力に性能

が抑えられる.

## (3) 専用OS

専用OSを使用することで, UNIX<sup>注2</sup>等の汎用OSでは困難な通信高性能化やQoS保障などが実現できる. しかし, 専用OSを用いると, ProxyやFirewallなどの汎用OS上で動作するアプリケーションプログラムの利用が困難になる.

## 3. ヘテロOS共存パラレル処理方式

DG実現のためには, 次の課題を同時に解決する必要がある.

- (a) OS内部処理も並列実行可能とする性能スケラブルなマルチプロセッシング方式の実現
- (b) 通信高性能化/QoS保障と既存OS互換性の両立の実現

上記課題を同時に解決する方法として, 我々はHiTactix/Symbiose[4]の技術を応用したヘテロOS共存パラレル処理システムを開発している. この方式では, 1個のCPUに1個のOSを割り当て, 各CPU上のOSを並列実行可能とすることによりスケラブルな性能向上を実現する.

我々が開発しているDGでは, 通信処理性能, QoS保証機能に優れたリアルタイムカーネル HiTactix[1,2,3]と, ProxyサーバやFirewall等のアプリケーションプログラムを有するUNIXを共存させる. OS間での連携処理に必要な機能(OS間通信, OS間同期, OS間で共有するグローバルデータの排他制御)はSymbioseが提供する.

CPU 2個の場合の基本構成を図1に示す.

- 1) CPU#0ではHiTactixが動作する. HiTactix上では, 暗号化, パケットスイッチ, リアルタイムIP (RTIP) ルーティング, IP層フィルタリングなど, L3までのパケット処理, 及び, QoS保証を必要とするパケットの処理を行う.
- 2) CPU#1ではUNIXが動作する. UNIX上では, TCP/IP等非リアルタイム・パケットの処理, Proxy等のアプリケーション層フィルタリング処理, 及び, Firewall等の既存UNIXソフトウェアなど, L4~L7のパケット処理を行う.
- 3) 物理メモリは, 各OSが占有アクセスするローカルメ

A Development of HiTactix/Symbiose (5)

— A Design of Dynamic Gateway —

Masahiko NAKAHARA, Masaaki IWASAKI,  
Tadashi TAKEUCHI, Takahiro NAKANO, Yoko KAWATA,  
Shouji KODAMA and Shihoko TAGUCHI  
Systems Development Laboratory, Hitachi Ltd.

注1: Pentium III Xeon 500MHz, メモリ2GB, WindowsNTの環境での結果. なお, 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.

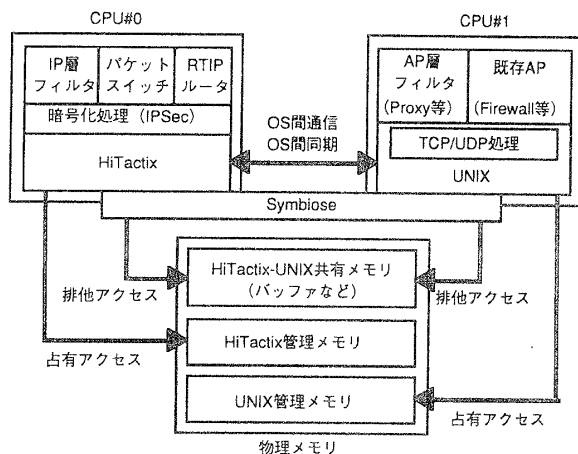


図1 HiTactix-UNIX共存パラレル処理方式の基本構成

メモリと、HiTactix-UNIX間で排他アクセスするグローバルメモリを持つ。ローカルメモリは、各OS固有のデータ、及び、各OS上で動作するアプリケーション等のメモリを割り当てる。グローバルメモリは、通信バッファなどOS間で共有するグローバルデータを割り当てる。

- 4) HiTactix及びUNIXは、グローバルデータへのアクセスやOS間通信、OS間同期を実行する場合のみSymbioseを呼び出す。

HiTactix-UNIX間で排他制御が発生するのはSymbioseを呼び出す場合だけである。また、SymbioseはOS間通信、OS間同期を一括化する機能を持ち、これに伴うオーバーヘッドを削減している[4]。これにより、HiTactix-UNIX間において並列実行度の高いプロトコル階層毎の機能分散処理を実現している。

#### 4. UNIXの対応

前節に示した基本構成に基づき、UNIX (BSD/OS 4.0.1を使用)をSymbioseに対応させるため、以下の変更を行った。

##### (1) 空間構成

BSD/OSはアドレスの小さい方から順に物理メモリを使用しているため、特定アドレスより上位の物理メモリ空間をBSD/OSから使用不能にし、該当メモリ空間をHiTactix及びSymbiose用の物理メモリ領域に割り当てている。BSD/OSの仮想空間内において、Symbioseは未使用の領域に割り当てている。

##### (2) ブートシーケンス

HiTactix-BSD/OSのブートは、BSD/OSの機能を利用して連携ブートする。具体的には、BSD/OSをシングルCPUモードでブートし、BSD/OSからcpuアプリケーション<sup>注3</sup>を使用して他のCPUの起動、Symbiose及びHiTactixのローディングを行う。これを実現するため、BSD/OSのioctlインタフェースにSymbioseとHiTactixをローディングする機能を追加している。

##### (3) デバイス共有と割り込み処理

HiTactix-BSD/OS間において、割り込みコントローラ以外のデバイスは共有しない。HiTactixの管理下にあるデバイスについては、BSD/OSからの検出を禁止し、BSD/OSから割り込みマスク要求/マスク解除要求が発行されることを防いでいる。割り込みコントローラへのアクセスについては、BSD/OSの割り込みマスク要求/マスク解除要求の発行部分についてSymbioseの該当機能(関数)を呼び出す形式に変更し、排他アクセスを実現している。

#### 5. おわりに

本稿では、HiTactix/Symbioseを応用したヘテロOS共存パラレル処理方式によりHiTactixとUNIXを連動させたDGについて述べた。ヘテロOS共存パラレル処理方式により、OSの並列実行を可能とする性能スケラブルなマルチプロセッシングが実現でき、また、QoS保証等の新機能と既存OS互換性の両立が実現できる。

現在、PentiumIIを2個搭載するPC上において、HiTactix同士を連動させるSymbioseの実装を完了し、DGシステムの性能評価を行っている[4]。また、これにNAT技術を利用したサーバ負荷分散機能を実装し、性能評価を行っている[5]。今後、HiTactix-UNIXを連動させたSymbioseを完成し、その評価を行う予定である。

#### 謝辞

本研究の一部は、IPAの次世代デジタル応用基盤技術開発事業「連続デジタルメディア処理向きアイソクロナス・カーネルの研究開発」として実施している。本研究開発の推進に当たり日頃から御指導・御助言をいただいているIPA技術応用事業部、大阪大学宮原研究室、九州大学谷口研究室、慶応義塾大学大岩研究室の方々には心より感謝申し上げます。

#### 参考文献

- [1]岩寄他:連続メディア処理向きマイクロカーネルHiTactixの設計と評価, 情報処理学会コンピュータシステム・シンポジウム論文集, pp.99-104 (1996) .
- [2]中野他:Ethernet上でQoSを保証する通信方式の設計と実装, 情報処理学会コンピュータシステム・シンポジウム論文集, pp.35-42 (1997)
- [3]竹内他: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, 情報処理学会マルチメディア通信と分散処理ワークショップ論文集, pp.119-126 (1998)
- [4]竹内他: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-, 第59回情報処理学会全国大会論文集
- [5]川田他:HiTactix/Symbioseの開発(6)-Dynamic Gatewayの性能評価-, 第59回情報処理学会全国大会論文集
- [6]SPEC Org: SPECweb96 Benchmark Results (<http://www.spec.org/osg/web96>)

注3: BSD/OS 4.0.1に付属のコマンドで、/dev/cpuに対してioctlシステムコールを発行する。