

柔軟でスケラブルな高性能並列ルータ CORErouter の基本構想

2Bb-5

高橋 直久 丸山 充 三栄 武 小倉 毅

© NTT ソフトウェア研究所

1 はじめに

インターネットの利用が拡大するにつれて、コンピュータネットワークの応答性や安全性の向上が求められている。また、マルチキャスト通信、QoS 制御、ポリシールーティング、移動体通信制御など高度なネットワーク制御の必要性が増しており、これらに関する新方式の研究が活発に行なわれている¹⁾。

一方、市販ルータの高速化、低価格化が進み、大規模なネットワークが多数構築されている。これらのルータは、一般に専用化された閉じたシステムとして作られている。このため、上記研究の成果を組み込んで実験するのが容易でない、あるいは、汎用 WS に比べるとフィルタリングなどのパラメータ設定のためのユーザインタフェースが極めて貧弱であるといった問題があり、柔軟性に欠けている。また、経路制御情報の更新のために処理能力が費やされ、パケットが捨てられたり、フォワーディングが遅延されたりする問題が発生し、処理能力の向上が求められている⁴⁾。さらに、コアバックボーンのリータとしては、広帯域化、大規模化に応じて性能向上ができることが望まれている。

我々は、このような状況を踏まえ、CORErouter と呼ぶ柔軟でスケラブルな高性能並列ルータの研究を進めている。本稿では、CORErouter のねらいとアーキテクチャの概要について述べる。

2 CORErouter の目標

CORErouter の研究では、以下に示す目標を達成して、応答性、信頼性、サービス性、可用性、保守性、安全性の高いネットワーク制御機能およびネットワークサービス機能を備えた並列 IP ルータの実現を目指している。

- 高速なスイッチング機構、キャッシング機構、サーチャージ機構、キューイング機構およびこれらのパイプライン制御により、高速 IP フォワーディング、マルチキャストパケットの高速スイッチング、本格的 QoS 制御を実現する。

CORErouter: A Flexible and Scalable High-Performance Parallel IP Router.

Naohisa TAKAHASHI (naohisa@slab.ntt.jp), Mitsuru MARUYAMA (mitsuru@ntt-20.ntt.jp), Takeshi MIEI (take@slab.ntt.jp) and Tsuyoshi OGURA (ogura@slab.ntt.jp)

NTT Software Laboratories

3-9-11 Midori-cho Musashino-shi Tokyo 180 Japan.

- ファイアウォール、DNS キャッシュ、http キャッシュ、各種 proxy などネットワークサービス機能のための計算リソースを柔軟で実時間性の高い接続機構によりルータ内に組み込み、ネットワークインタフェース、インターネットワーク、トランスポート、アプリケーションの各レイヤの機能を協調させ、高度なネットワークサービスを効率的に実現可能にする。
- ルータ機能を容易に操作可能な API の提供により、高速できめ細かいソフトウェア制御を可能にし、ルータ処理に関する新しいアルゴリズムの実験などのネットワーク研究、および、高速ネットワークの診断運用を容易にする。
- 処理量（収容回線の帯域/回線数、経路制御負荷など）が増加しても、プロセッサ等のハードウェア資源を追加すれば、ルータの QoS を一定に保持するようにシステム性能を向上させられるスケラブルなルータアーキテクチャを実現する。

3 設計指針

3.1 アーキテクチャ設計の方針

- ルーティングテーブルとフィルタリング制御情報の分散化およびキャッシングにより、データアクセスが局所化したマルチプロセッサ構成を実現して、並列性を向上させる。
- パケットフォワーディング処理、ルーティング制御処理、ネットワークサービス処理のためのプロセッサを分離して、経路情報更新におけるルートフラッピング^{1), 4)}などの影響を受けないパケット処理を実現する。
- 共有バスなどボトルネックになる要因を排除してスケラビリティの高い構成を追求し、処理対象の規模に応じたシステム構成を作ることが容易にする。
- ヘテロでオープンな並列アーキテクチャを実現し、計算リソースとして PC などのコモディティを多数利用可能にする。
- インタフェースの追加、ソフトウェア機能の部分的なハードウェア化、ハードウェアの一部置き換

えなどにより高速化、低遅延化を段階的に進められる柔軟な構成を実現する。

3.2 システム実現へのアプローチ

分散メモリ型、あるいは、分散共有メモリ型など、多くのスケーラブルな並列計算機の研究が進められ、商用化されている²⁾。特に、Exemplar, CS2, nCUBE3などは、プロセッサ間通信の高速化、低遅延化を進めるとともに、入出力インタフェースをPCI, Sbusなどの標準仕様にし、さらに入出力もスケーラブルな構成にしている点が、CORErouterのハードウェアの候補として注目される。しかし、これらのマシンをベースにしてCORErouterを実現しようとするとき次のような問題が生じる。

- 科学技術計算、データベース処理などの応用に対してI/O能力とCPU能力がバランスしたアーキテクチャであるので、十分なI/O能力を得ようとするとき必要以上にCPU能力が大きくなり極めて高価になる。
- 大量データのバースト転送の高速化に力をいれており、入出力のスタートアップ時間の減少など低遅延化はあまり考慮されていない。
- 専用プロセッサ、あるいは専用OSを使っているため、汎用WS上のネットワーク制御プログラムなどのソフトウェアの移植や改造が難しい。

これらの問題を解決するためには、次のようなアプローチが望ましいと考える。

- 広帯域、低遅延でスケーラブルなメッセージ通信用スイッチを開発する、あるいは、商用並列計算機の多段結合網や高速LAN用スイッチなど市販の高性能スイッチを利用する。
- 上記スイッチのインタフェースとPCIなどの標準的な高速バスインタフェースをもつアダプタを介してWSやPCをスイッチに接続して、オープンでヘテロな分散メモリ型並列マシンを構築する。
- 低遅延通信が必要な処理については、処理プロセッサに上記スイッチのインタフェースを組み込み、WS、PCのOSを介さずに直接通信可能にする。

4 CORErouterの構成

CORErouterは、前述のような柔軟でスケーラブルなルータを実現するため、図1に示すように、以下の4種類のプロセッサを高速な相互結合網で接続した機能分散型のマルチプロセッサ構成をとる。

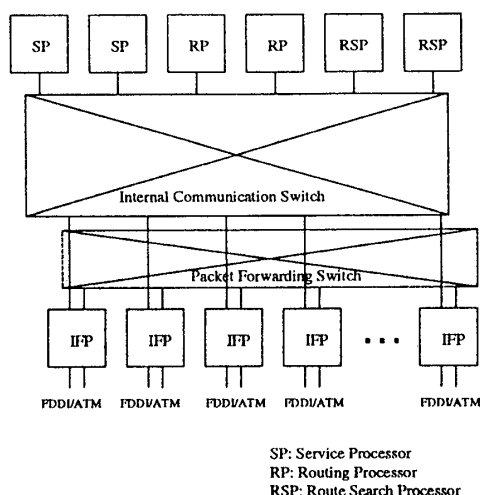


図1: CORErouterの構成

- サービスプロセッサ (SP) : ファイアウォール, DNS, proxy, http キャッシュなどの高機能処理
- ルーティングプロセッサ (RP) : ルーティングプロトコルの処理
- ルートサーチプロセッサ (RSP) : ルートサーチテーブルの高速検索処理
- 回線インタフェースプロセッサ (IFP) : パケットの高速フォワード処理

5 おわりに

現在、市販の高速スイッチを相互結合網として用いたWSクラスタを構築し³⁾、CORErouterの各プロセッサ機能をWSのソフトウェアとして実現して評価実験を進めている⁴⁾。

謝辞 本研究をご支援下さる後藤滋樹広域コンピューティング研究部長、ならびに日頃ご討論頂く超並列プログラミング研究グループの皆様には感謝いたします。

参考文献

- 1) Christian Huitema, "Routing in the Internet," Prentice Hall PTR, Englewood Cliffs, New Jersey, 1995.
- 2) D.E. Lenoski and W.D. Weber, "Scalable Shared-Memory Multiprocessing," Morgan Kaufmann Publishers, San Francisco, California, 1995.
- 3) 三米, 小倉, 丸山, 高橋, "WSクラスタを用いたCORErouterプロトタイプ構成," 情処第52回大会, 2B-6, 1995.
- 4) 丸山, 三米, 小倉, 高橋, "WSクラスタを用いたCORErouterプロトタイプの評価," 情処第52回大会, 2B-7, 1995.