

ネットワークプロセッサ技術に関するサーベイ

河合 栄治^{†,††} 門林 雄基[†] 山口 英[†]

[†] 奈良先端科学技術大学院大学 〒630-0101 奈良県生駒市高山町 8916-5

^{††} 科学技術振興事業団 さきがけ研究 21 〒332-0012 埼玉県川口市本町 4-1-8

E-mail: {eiji-ka,youki-k,suguru}@is.aist-nara.ac.jp

あらまし 今後の超高速ネットワーク時代を実現する技術の一つとして、ネットワークプロセッサ (NP) が注目を集めている。NP とは、ネットワーク処理のための専用のプロセッサであり、ルータなどの中継装置だけでなく、サーバなどのエンドホストにも搭載することができる。本稿では、この NP 技術およびその研究動向に関してサーベイを行う。具体的には、アーキテクチャ、性能評価、プログラミング、アプリケーション技術の各分野について論じる。
キーワード ネットワークプロセッサ (NP)、アーキテクチャ、性能評価、プログラミング、アプリケーション

A Survey on Network Processor Technologies

Eiji KAWAI^{†,††}, Youki KADOBAYASHI[†], and Suguru YAMAGUCHI[†]

[†] Nara Institute of Science and Technology 8916-5 Takayama, Ikoma, Nara, 630-0101 Japan

^{††} PREST, Japan Science and Technology Corporation 4-1-8, Honcho, Kawaguchi, Saitama, 332-0012 Japan

E-mail: {eiji-ka,youki-k,suguru}@is.aist-nara.ac.jp

Abstract Network processor (NP) is a promising technology that will support tomorrow's ultra high speed networking. The NP is a sub-system dedicated to network processing with special processors, memory buffers, and network interfaces, and it can be used not only in network routers but in server hosts. In this paper, we survey the NP technologies and its research trends. We discuss NP architectures, performance evaluation scheme, programming environments, and its application.

Key words network processor (NP), architecture, performance evaluation, programming, application

1. はじめに

近年のネットワークプロセッサ技術の発展は目覚ましいものがある。ネットワークプロセッサ (以下 NP と略す) とは、ネットワークトラフィックを処理するための専用のプロセッサであり、ルータなどの中継ノードだけでなく、サーバなどのエンドノードでも利用が可能になってきている。もともと、高レベルな処理を外部の補助プロセッサで処理するというアイデア自体は新しいものではない [1], [2]。現在、NP が注目されるようになった背景には、次のような点が挙げられる。

まず、ルータのパケット転送能力の向上および QoS (Quality of Service) 機能などの新しい機能の搭載を短時間で実現するためである。近年のルータの実装においては、高速化を実現するために ASIC (Application-Specific Integrated Circuits) を用いたパケット処理プロセッサをネットワークインタフェースの近傍に配置している [3]。しかし、今後の超高速ネットワークをサポートするためには開発に莫大な費用と時間がかかるため、ソフトウェアによるプログラムが可能で並列化による性能向上が容易な NP を用いることで、短期間で安価な製品開発が可能に

なると期待されている。

次に挙げられるのが、アクティブネットワーク技術 [4], [5] に代表されるように、より高次のネットワークサービスの構築が望まれるようになったことである。アクティブネットワーク技術は、ネットワークサービスをエンドノードによる処理だけでなく、ルータなどの中継ノードにおいても処理を行うことでより動的なサービスを実現するものである。中継ノードにおけるより柔軟な処理能力を実現するために NP を応用することが期待されている。

最後に、ゲートウェイやサーバホスト等のエンドノードの性能向上が求められていることが挙げられる。一般的に、プロセッサの処理能力は 18 カ月に 2 倍向上する (Moore の法則) ことが観測されている。一方で、近年のネットワーク帯域向上の速度は、この計算機の処理能力向上の速度よりもはるかに大きい [6]。そのため、エンドノードにおける TCP/IP 処理やアプリケーション層の処理を一部 NP に移すことで、処理能力の向上が期待できる。

このように大きな期待を集めている NP であるが、技術開発はこれまで産業界が主導して行ってきた [7]。そのた

め非公開な情報も多く、また研究成果の学術的な体系化も不十分であるのが現状である。本稿では、NPに関する学術論文を中心にサーベイする。特に、現在のNPに関する最先端研究が、どのように相互に関連しどこに向かおうとしているのかを少しでも明らかにできればと考えている。

以下本稿の構成を示す。まず、2.でNPアーキテクチャに関する研究を概観する。次に3.では、NPにおける性能評価手法について述べる。ここでは、ベンチマークおよびシミュレーションに分類している。4.では、NPにおけるプログラミングに関する研究について概観する。また、5.では、現在主流となっている高速レイヤ3パケット転送以外の機能のNPにおける実現について研究成果を概観する。最後に6.で将来のNP研究に関する展望を簡単にまとめる。

2. アーキテクチャ

NPで最も重要なのがアーキテクチャであり、これまでに多くのベンダーから様々なアーキテクチャを持つNPがリリースされている。それらは、並列処理の実現方法、特定の目的のためのハードウェアの構成、メモリ構成、チップ上の通信機構、ネットワークインタフェース等の周辺装置との接続方法、などの特徴によって分類することができる[8]。しかし、それらの要素は互いに密に関連するため、本稿ではNP上に1つもしくは複数配置される処理要素(Processor Element: 以下PEと略す)のマイクロアーキテクチャと、NPにおけるPEやメモリ、その他の周辺機器の配置および配線を含めたシステムアーキテクチャに分類し、特に各研究のアプローチの違いについて概観する。

2.1 マイクロアーキテクチャ

2.1.1 PEの要件

NPにおけるマイクロアーキテクチャの要件を検討している初期の文献には[9],[10]がある。[9]ではNPにおける処理の特徴を、(1)データ転送の負荷が高い、(2)割り込みが非常に高い頻度で発生する、(3)複雑な状態管理を必要とする処理が多い、(4)ルーティングなどでは表検索が重要、(5)ワードやバイトの境界をまたぐ処理もある、とまとめ、ハードウェアによるマルチスレッドのサポートや、コアの命令セットに関する検討を行っている。実装したチップを用いた評価では、1Mbpsのデータに対しておよそ2MIPSの処理能力で十分であり、汎用プロセッサより有利にパケット処理が可能であることを示している。

また、文献[10]では、マイクロアーキテクチャとして Super-scalar Processor (SS), Fine-grained Multithreaded Processor (FGMT), Single Chip Multiprocessor (CMT), Simultaneous Multithreaded Processor (SMT) の4種類について、典型的なワークロードに対してSMTが最も良い性能を発揮すると結論づけている。具体的には、SMTもしくはCMTアーキテクチャでクロック速度が500MHzのプロセッサを用いれば、単純なレイヤ3パケット転送であれば10Gbpsを越える処理能力を、さらにはMD5や3DESにおいても1Gbpsを越える処理能力を持つと推測している。

文献[11]では、エンドノードにおけるTCP off-loadingのためのNPを設計している。TCPではパケット間の状態情報を管

理する必要があることから、一般的なレイヤ3をターゲットにしたNPとは異なる構造が必要となる。ここでは、パケット内で完結するデータ処理を行うPEと、状態情報の検索やPEの制御、ホストプロセッサおよびメモリとの通信と行うプロセッサ(Micro Controller: 以下 μC と略す)に機能分離し、 μC の要件を検討している。 μC の処理はラインスピード(最小のパケットで帯域幅を埋める速度)では処理できないため、パケットサイズ等を考慮しなければならないとしている。

2.1.2 キャッシュ

NPにおけるキャッシュは非常に議論するのが難しい分野の一つである。パケットデータの参照回数が低いいためデータキャッシュが有効ではないこと、命令キャッシュはマイクロおよびシステムアーキテクチャに大きく依存することなどが原因である。その中で、文献[12],[13]では、レイヤ3転送におけるルーティングテーブルの高速検索手法に焦点を当て、キャッシュの側面から考察している。メモリキャッシュの構成においては、キャッシュのサイズ、ブロックサイズ、連想記憶の幅(associativity)が重要であり、効率的な検索を実現するためにルーティングテーブルのエントリ数を効率よく圧縮する技術を提案している。

2.2 システムアーキテクチャ

NPの基本的な目標の一つに処理の並列度の向上がある。そのため、並列アーキテクチャのNPへの応用を考察した研究が存在する。

まず、NP実装に関する初期の研究としてComet[14]がある。Cometでは、チェックサムやテーブル検索などの機能単位をFPGA(Field Programmable Gate Array)を用いて実装し、それらを汎用プロセッサを用いて制御する方式をとっている。これは、現在主流となっているNPのアーキテクチャと同じであり、先駆的な研究として注目に値する。さらには、Cometボードのクラスタ構成法も検討し、高性能かつ柔軟なルータの構築を実現している。

文献[15]では、柔軟なQoSを実現するための高度に並列化されたNPアーキテクチャを提案している。提案したアーキテクチャでは、1つのLSI上に16個のPEをグループに分割した上で階層構造的に配置し、各PE内の2つのコアとレジスタファイルをグループ内通信バスで直結し同期させることで、PE間の高い相互通信機能を実現している。また、資源アクセス制御については、コンパイル時にあらかじめ全てのPEのコードを参照しておくことで静的に解決し、プロセッサコアはレジスタファイル上でのみ演算処理を行うことで全体で静的に同期した処理を実現している。

また、超並列計算機の様に非常に多数のRISCベースのプロセッサをチップ上に配置し、プロセッサ間を縦横に接続した静的なチャンネルと宛先を記述できる動的なチャンネルによってプロセッサ間通信を実現するアーキテクチャ(Raw Processor[16]と呼ばれる)も提案されている[17]。これは、コンパイル時に可能な最適化はもちろんのこと、プロセッサのスケジューリングをソフトウェアで制御することも可能であるという特徴を持つ。

また、マルチプロセッサベースのNPにおけるプロセッサ間の通信機構(Octagonと呼んでいる)に焦点を当てた研究には[18]

がある。Octagon は、8 個のプロセッサおよびそれらを結ぶ 12 本のポイントツーポイント結合より構成され、ネットワーククロスバススイッチよりも配線が少なく、最悪でも 2 ホップで任意のプロセッサ間も通信可能で総合的なスループットが高いという特徴を持つ。

文献[19]では、非常に多数のスレッドをサポートするアーキテクチャへのアプローチとして、スレッド間のメモリ同期問題をハードウェアでサポートしたり、それぞれのプロセッサにおけるローカルなメモリアクセス手法に加えてグローバルなメモリ空間も用意するハイブリッドな構造にする手法について述べている。これにより、スレッドのコア集合における配置問題をプログラミングから切りはなして実現している。

また、並列計算機へのアプローチとしてデータ駆動の概念を用いた NP の研究もある[20]。ここでは、データ駆動型プロセッサを用いた終端ホスト向け TCP/IP off-loading エンジンを開発している。アプローチとしては、各パイプラインのステージにおける処理時間を考察し、総合的なスループットの向上を計るように工夫したり、ボトルネックとなる個所を 1 チップ集積したりすることでスループットを向上させる手法を紹介している。

文献[21]では、IBM が提唱している BlueGene/Cyclops cellular アーキテクチャ[22]を用いた NP について述べている。各 PE の命令キャッシュの有効性を向上させるために、パイプライン処理を採用し、[15]と同様プロトコル処理のプロセッサへの割り当てやメモリ管理は静的に行っている。実際のアプリケーションとしてファイバチャネルと Infiniband のプロトコル変換機能を実装し、シミュレーションによる評価を行っている。

3. 性能評価

システムの性能評価では、ベンチマークやシミュレーションといった直接的な評価に加え、システムの抽象化を通じた設計空間の探索がアーキテクチャ開発においては有効である。本節ではそれぞれについて述べる。

3.1 ベンチマーク

ベンチマークを行うためには、NP に特化したワークロードを決定する必要がある。文献[23]では、ヘッダ処理としてルーティングテーブル検索、パケットのフラグメンテーション処理、QoS のためのキュー管理、TCP トラフィックモニタリングを選出している。一方でペイロード処理には、暗号アルゴリズム、データ圧縮、Forward Error Correction (FEC)、画像圧縮を選出している。また、文献[24]では計測対象の範囲を広げ、ベンチマークプログラムをマイクロレベル、IP レベル、アプリケーションレベルに分類している。

ベンチマークのもう一つの視点として、異なるアーキテクチャ間の性能比較がある。異なるアーキテクチャの比較を行うためには、システムレベルでの機能の抽象化が必要となってくる。文献[25]では、ベンチマークプログラムをシステムの階層構造から、システムレベル、機能レベル、マイクロレベル、ハードウェアレベルに分類して定義することにより NP システムのモデル化を行い、様々な NP アーキテクチャに適用可能になるようにしている。また、ベンチマークの方法論を含めたフ

レームワーク化の議論を行っているのが文献[26]である。ここでは、ベンチマークにおける 3 つの仕様、(1) 機能仕様、(2) 環境仕様、(3) 測定仕様、を定めている。中心的役割を果たす環境仕様ではネットワークインタフェースやその制御インタフェース、トラフィックや負荷状況などを定義し、ベンチマークの汎用性を高めている。

3.2 シミュレータ

NP におけるシミュレーションの場合、ある特定のシステムの挙動を詳細に再現するためのものと、特定の実装に依存しない抽象的なフレームワークにより様々な構成のシステム性能を評価するためのものに分類することができる。前者のものは、多くの NP システムの開発環境に付随して配布されている。ここでは、後者に分類されるものについて述べる。

3.2.1 シミュレーションフレームワーク

NP システムを抽象化する場合、パケットのデータフローに従うのが開発者にとって容易である。文献[27]では、システムを (1) トラフィック生成部、(2) 入力解析部、(3) 入力インタフェース、(4) プロセッサ部、(5) 出力インタフェース、(6) 出力解析部、にコンポーネント化し、クロック同期を用いた実時間シミュレーションによる正確な評価を実現している。ただし、シミュレータの実装が Cisco の NP である Toaster2 の構造に依拠しており、他の NP にも適用可能なように汎用性を向上することが求められる。

また、NP のシミュレーションに既存の汎用 OS 環境を利用したのもある。文献[28]では、エンドノードにおける TCP off-loading エンジン開発をターゲットとしたシミュレーションフレームワーク (Countach と呼んでいる) について述べている。これは、Unix 上でユーザレベル TCP スタックを用い、細粒度のフックを埋め込むことでプログラムのメモリ参照の振舞いおよび各種装置におけるイベント処理の正確な同期をシミュレートすることができる。

3.2.2 設計空間の探索

NP 開発では、様々なシステム構成およびワークロードに対する性能評価を通じてより高い性能を達成することが重要となる。ここでは、特にそのような設計空間の探索を実現する技術に関する研究について述べる。

設計空間をシミュレーションする場合、アーキテクチャの変更を容易するためにモジュール化が重要となる。文献[29]では、高度にモジュール化されたソフトウェアルータである Click [30] をベースにした、NP の処理能力モデルを構築している。Click における各機能のモジュール化およびイベント管理機能を活用し、ネットワークプロセッサシステムのモデル (プロセッサやメモリ、接続チャネル) を追加することで、システム全体の性能の柔軟なシミュレーションを実現している。

また、設計空間の探索は抽象度の高い作業であるため、シミュレーションよりも抽象度の高い解析的なシステムモデルも有用である。文献[31]では、NP システムを 1 つの I/O インタフェースと複数のクラスタに分離し、各クラスタには複数の PE および 1 つのメモリインタフェースを搭載し、各 PE は命令セットとデータキャッシュを有するモデルを提案している。これら

の構成要素においてパラメータ化を行い、ベンチマークワークロードに対する性能を解析的に算出している。また、文献[32]では、このフレームワークを用いて処理能力と消費電力に関する分析も行っている。さらには、文献[33]では解析的なフレームワーク[34]を用いてPE間の接続トポロジというより高レベルなシステム構成に関する検討を行っている。

一方で、これらの解析的なフレームワークは、抽象度が高すぎるため実際のシステムの特徴を十分に反映できない問題点があることには注意が必要である。例えば、解析的モデルではPEによるパイプライン構成をとるより各PEがパケット処理のすべてを行う方式の方が優れているとしている[33]が、実際には命令キャッシュの制約やチップ面積、通信コストの問題などからパイプライン構成を取る方が現実的である場合が多い。

3.2.3 遅延の解析

NPにおいては確定的な性能の確保が重要視されるため、コードブロックにおける最悪実行時間の分析が重要となる。文献[35]では、マルチスレッド環境を想定した最悪実行時間の算出法について述べている。具体的には、実行コードの制御フローグラフ(Control Flow Graph: CFG)をもとにそれぞれのブロックの実行回数に関する制約を全て列挙し、線形計画法により最もコストの高いパスの実行時間を算出するというものである。マルチスレッドへの対応においては、CFGに中断ノードおよび中断エッジという概念を導入し、スレッド間制御切替えおよび切替えによるメモリアクセス遅延時間隠蔽もサポートしている。このような解析で得られる最悪実行時間は、理論的最悪値であることから悲観的(pessimistic)過ぎる傾向があるが、NPの場合はCFGが比較的単純であり、大きなプログラムでもシミュレーションによる実測時間の4倍程度であると報告されている。

4. プログラミング

一般的にNPにおけるソフトウェア開発には低レベルなプログラミングが要求され、コストが高いという問題がある。そこで、NPのソフトウェア開発を容易にするプログラミング環境に関する研究がいくつかなされている。こうした研究の多くが、3.で述べたようなシステムの抽象化を基礎にしている。

4.1 プログラミングフレームワーク

文献[36]では、先に紹介した高いモジュール性をもつソフトウェアルータClick[30]のフレームワークをNPに応用したプログラミングモデルNP-Clickを提案している。Clickのフレームワークを用いることで、NPの様々な機能を抽象化し、それらの上に各種モジュールを提供することを可能にしている。性能評価では、IntelのIXP1200を用いた実装において、パケットサイズが十分大きい場合にはIXP1200用の専用開発環境であるMicroengine C[37]による実装と比較してほぼ遜色のない性能を実現している。

文献[38]では、別のアプローチによるモジュール化支援プログラミング環境NEPAL(Network Processor Application Language)を提案している。先のNP-ClickではNP特有機能の抽象化を言語レベルで実現したのに対して、NEPALではランタイムによる動的な制御機構によって実現している。しかし性能評

価においては、単一プロセッサ上で実行した場合に対するチップマルチプロセッサ(CMP)の性能向上しか議論されておらず、確定的な性能が達成されるかどうか不明である。一般的にはランタイムを用いた手法はオーバヘッドが大きく[36]、2.2で述べたような超並列アーキテクチャのようなスケラビリティを重視したアーキテクチャなどで用いられると考えられる。

また、文献[39]では、言語レベルの静的なモジュール化とランタイムによる動的なモジュール化の中間的なアプローチとして、コンポーネントと呼ぶコードの最小単位を動的にバインドしてモジュールを構成する手法について提案している。具体的には、プログラミングを容易にする階層的な抽象化、コンポーネント間のデータの受け渡しのためのレジスタの割り当て方式、コンポーネントの動的なバインド手法を設計し実装している。性能評価では、動的なモジュール構成のないモノリシックな実装に対して2%程度と非常に小さい性能低下を実現している。

4.2 スケジューリング

NPでは、高いスループットの達成のために複数のプロセッサを搭載しマルチスレッドをハードウェアでサポートしている場合が多い。例えば、IntelのIXP1200ではメモリアクセス遅延隠蔽を目的としたnon-preemptiveマルチスレッド環境を提供している。また、確定的な遅延および帯域幅を実現するために、静的なスレッドスケジューリングを用いるものもある[15]、[21]。一方で、より柔軟なマルチスレッド環境ではソフトウェアによる支援が必要となる。文献[40]では、NPをマルチプロセッサでマルチスレッドをサポートするリアルタイムシステムと定義し、公平性を満たすPfair(Proportionate Fairness)スケジューリングと呼ばれる方式のNPへの適用を提案している。

4.3 コンパイラ

NPでは特殊なマイクロアーキテクチャを採用することが多く、ワードやバイトの境界をまたぐようなデータ処理など汎用プロセッサにはない命令も効率よく処理する必要がある。ここでは、ネットワークの領域から逸脱するため紹介するだけにとどめるが、そのようなNP用コンパイラの最適化手法の研究として文献[41]~[43]などがある。また、NPの適用先として有望視されているアクティブネットワークでは、各ルータのアーキテクチャに依存しない実行コードの注入が必要となる。文献[44]では、NPにおけるJust-in-Time(JIT)コンパイラの可能性について議論している。

5. アプリケーション

NPを高速なレイヤ3ルーティングだけでなく、より高いレイヤの処理に用いる研究がいくつかなされている。本節ではそれらについて紹介する。

5.1 パケットの分類

パケットの分類はネットワークセキュリティ機能の中心的役割を担っており、NPによる高速処理が期待されている。文献[45]では、DDoS(Distributed Denial of Service)攻撃を軽減するためのパケットフィルタリング機能(NetBouncer[46])のNPへの実装について、特にIntelのIXP1200上の実装に関する知見がまとめられている。また、文献[47]では、アクセス制御リス

ト (ACL) を用いたパケットの分類機能を, NP 上で実現するための要件について述べている。この研究では, 従来では機密情報にあたるため公にはあまり議論されてこなかった ISP や企業で用いられているファイアウォールの ACL を用いて, より現実的な環境に適した方式を提案している。結論としては, (1) IP アドレス対 (送信者と受信者) に関する検索とトランスポート層の情報に関する検索を分離する, (2) IP アドレス対の重複する部分は別途管理し, IP アドレス対に対して一つのフィルタを返すようにする, (3) トランスポート層のフィールド検査ルールはそれほど多くないので, 特別なハードウェア装置を用いた検索をサポートする, という要件を導出している。

5.2 レイヤ 4 機能の off-loading

NP の適用先の一つとして, エンドノードにおける TCP/IP 処理の NP への移行が検討されている。文献 [48] では, ワークステーション用周辺機器としての TCP/IP 通信ボードを提案している。実装にあたっては, ネットワーク処理特有の機能をプロトコルヘッダ処理部と資源管理部に分離し, プロトコルヘッダ処理部の各種機能を LSI で実装している。これらの機能を汎用 RISC プロセッサ (クロック 33MHz の SPARCLite) を用いてプログラム可能にしている。コネクション多重化に対応していないなどの不備はあるが, 先駆的な研究の一つに挙げられる。

他にも, これまでに述べた文献 [11], [20], [28] も TCP/IP 処理の off-loading を目的としている。

5.3 ストレージサービス

文献 [49] では, ネットワークストレージサービスを組み込み用プロセッサを搭載したハードウェアで実現するシステムを開発している。システム設計的にはボード内の通信バスとして PCI を用いたり, ネットワーク処理も Linux カーネルを用いているため, 正確には NP の応用とは言えないが, ネットワークストレージは NP の格好のアプリケーションとして期待されている [21]。

5.4 レイヤ 7 補助プロセッサ

文献 [50] では, レイヤ 7 の処理を補助する NP 用の補助プロセッサを提案している。提案しているプロセッサはバレルシフトに似た構造を持ち, HTTP における URL ベースのスイッチなどで必要となるパターンマッチング機能や, IP ルーティングにおける radix-tree 検索機能, セキュリティ機能等で利用される MD5 アルゴリズムなどへの応用例を示し, シミュレーションによる性能評価を行っている。また, 文献 [51] では, このアクセラレータの FPGA (Field Programmable Gate Arrays) チップにおける実装を検討している。

6. おわりに

本稿では, NP におけるアーキテクチャ, 性能評価, プログラミング, アプリケーションの各研究分野について成果を概観した。一方で, 本稿では学術論文以外の文献 (各種製品情報など) についてはほとんど触れることができなかった点と, 学術論文についても網羅することを目的とはしていないため多くの漏れがある点については注意が必要である。特に 2. で述べたアーキテクチャについては, 多くのベンダーから多種多様な

NP がリリースされているにも関わらず, ほとんど触れることができなかった。それらについては, 各社から製品とともにリリースされているホワイトペーパーや, ベンダー向けフォーラムの Network Processing Forum (NPF) ^(注1) や, コンファレンスの Network Processor Conference ^(注2) を参照すると良い。

これから NP は, 多種多様なアーキテクチャが考案される初期の研究開発段階から, 実用化を含めた次の段階へ移行するだろう。特に, 高いスループットを要する基幹ルータへ応用が可能かどうか NP の当初の目的から一つの焦点になると考えられる。例えば, ソフトウェアルータとしてデファクトスタンダードの地位を確立している Zebra の商用版である ZebOS Advanced Routing Suite では NP を用いることができ [52], 実用化が進められている。また, レイヤ 3 パケット転送以外のクライアントアプリケーションの開発も求められている。このように, 今後はより実用的な技術の研究開発が重要になるだろう。

文 献

- [1] Y. Coady, J. S. Ong and M. J. Feeley: "Using Embedded Network Processors to Implement Global Memory Management in a Workstation Cluster", Proceedings of the 8th International Symposium on High Performance Distributed Computing, Redondo Beach, California, pp. 319-328 (1999).
- [2] G. Powers: "A front end telnet/rlogin server implementation", Proceedings of UniForum 1986, pp. 27-40 (1986).
- [3] P. Crowley, M. A. Franklin, H. Hadimioglu and P. Z. Onufryk Eds.: "Network Processor Design: Issues and Practices", Vol. 1, chapter 9: An Industry Analyst's Perspective on Network Processors, pp. 191-218, Morgan Kaufmann Publishers (2003).
- [4] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall and G. J. Minden: "A Survey of Active Network Research", IEEE Communications Magazine, 35, 1, pp. 80-86 (1997).
- [5] 山本: "アクティブネットワークの技術動向", 電子情報通信学会論文誌 B, J84-B, 8, pp. 1401-1412 (2001).
- [6] L. G. Roberts: "Beyond Moore's Law: Internet Growth Trends", IEEE Computer, 33, 1, pp. 117-119 (2000).
- [7] P. Crowley, M. A. Franklin, H. Hadimioglu and P. Z. Onufryk Eds.: "Network Processor Design: Issues and Practices", Vol. 1, chapter 1: Network Processors: An Introduction to Design Issues, pp. 1-8, Morgan Kaufmann Publishers (2003).
- [8] N. Shah and J. Keutzer: "Network Processors: Origin of Species", Proceedings of the 17th International Symposium on Computer and Information Science (ISCIS XVII), Orlando, Florida (2002).
- [9] X. Nie, L. Gazsi, F. Engel and G. Fettweis: "A New Network Processor Architecture for High-Speed Communications", Proceedings of IEEE Workshop on SIGNAL PROCESSING SYSTEMS (SiPS'99), Taipei (1999).
- [10] P. Crowley, M. E. Fiuczynski, J. L. Baer and B. N. Bershad: "Characterizing Processor Architectures for Programmable Network Interfaces", Proceedings of the 2000 International Conference on Supercomputing, Santa Fe, N.M. (2000).
- [11] U. Nordqvist and D. Liu: "Packet Classification and Termination in a Protocol Processor", Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 88-99 (2003).
- [12] T. C. Chiueh and P. Pradhan: "Cache Memory Design for Network Processors", Proceedings of the Sixth International Symposium on High-Performance Computer Architecture, Toulouse, France (2000).
- [13] K. Gopalan and T. C. Chiueh: "Optimizations for Network Processor Cache", Proceedings of SC2002 High Performance Networking and Computing, Baltimore (2002).
- [14] 陣崎, 中村, 村井: "ギガビットルータ Comet のアーキテクチャと

(注1): <http://www.npforum.org/>

(注2): <http://www.networkprocessors.com/>

- その評価”, インターネットコンファレンス'98 論文集 (1998).
- [15] H. Shimonishi and T. Murase: “A network processor architecture for flexible QoS control in very high-speed line interfaces”, 2001 IEEE Workshop on High Performance Switching and Routing (HPSR2001), Dallas, Texas, pp. 402–406 (2001).
- [16] E. Waingold, M. Taylor, D. Srikrishna, V. Sarkar, W. Lee, V. Lee, J. Kim, M. Frank, P. Finch, R. Barua, J. Babb, S. Amarasinghe and anant Agarwal: “Baring It All to Software: Raw Machines”, IEEE Computer, **30**, 9, pp. 86–93 (1997).
- [17] G. Chuvpilo, D. Webtzlaff and S. Amarasinghe: “Gigabit IP Routing on Raw”, Proceedings of the 1st Workshop on Network Processors, Boston, MA (2002).
- [18] F. Karim, A. Nguyen, S. Dey and R. Rao: “On-Chip Communication Architecture for OC-768 Network Processors”, Proceedings of the 38th Design Automation Conference (DAC2001), Las Vegas, Nevada, pp. 678–683 (2001).
- [19] S. Melvin, M. Nemirovsky, E. Musoll, J. Huynh, R. Milito, H. Urdaneta and K. Saraf: “A Massively Multithreaded Packet Processor”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 64–74 (2003).
- [20] 西川, 青木: “プロトコル多重処理のデータ駆動型実現法とその実験的検討”, 電子情報通信学会論文誌 D-I, **J85-D-I**, 7, pp. 635–643 (2002).
- [21] C. J. Georgiou, V. Salapura and M. Denneau: “A Programmable Scalable Platform for Next Generation Networking”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 1–9 (2003).
- [22] IBM Blue Gene team: “Blue Gene: A vision for protein science using a petaflop supercomputer”, IBM Systems Journal, **40**, 2, pp. 310–326 (2001).
- [23] T. Wolf and M. Franklin: “COMMBENCH – A Telecommunications Benchmark for Network Processors”, Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software, Austin, TX (2000).
- [24] G. Memik, W. H. Mangione-Smith and W. Hu: “NetBench: A Benchmarking Suite for Network Processors”, Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA (2001).
- [25] P. R. Chandra, F. Hady, R. Yavatkar, T. Bock, M. Cabot and P. Mathew: “Benchmarking Network Processors”, Proceedings of the 1st Workshop on Network Processors, Boston, MA (2002).
- [26] M. Tsai, C. Kulkarni, C. Sauer, N. Shah and K. Keutzer: “A Benchmarking Methodology for Network Processors”, Proceedings of the 1st Workshop on Network Processors, Boston, MA (2002).
- [27] D. Suryanarayanan, J. Marshall and G. T. Byrd: “A Methodology and Simulator for the Study of Network Processors”, Proceedings of the 1st Workshop on Network Processors, Boston, MA (2002).
- [28] P. Pradhan, W. Xu, I. Nair and S. Sahu: “Efficient and Faithful Performance Modeling for Network-Processor Based System Design”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 125–132 (2003).
- [29] P. Crowley and J. L. Baer: “A Modeling Framework for Network Processor Systems”, Proceedings of the 1st Workshop on Network Processors, Boston, MA (2002).
- [30] E. Kohler, R. Morris, B. Chen, J. Jannotti and M. F. Kaashoek: “The Click modular router”, ACM Transactions on Computer Systems, **18**, 3, pp. 263–297 (2000).
- [31] M. A. Franklin and T. Wolf: “A Network Processor Performance and Design Model with Benchmark Parameterization”, Proceedings of the 1st Workshop on Network Processors, Boston, MA (2002).
- [32] M. A. Franklin and T. Wolf: “Power Considerations in Network Processor Design”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 10–22 (2003).
- [33] M. Gries, C. Kulkarni, C. Sauer and K. Keutzer: “Exploring Trade-offs in Performance and Programmability of Processing Element Topologies for Network Processors”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 75–87 (2003).
- [34] L. Thiele, S. Chakraborty, M. Gries and S. Künzli: “Design Space Exploration of Network Processor Architecture”, Proceedings of the 1st Workshop on Network Processors, Boston, MA (2002).
- [35] P. Crowley and J. L. Baer: “Worst-Case Execution Time Estimation for Hardware-assisted Multithreaded Processors”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 36–47 (2003).
- [36] N. Shah, W. Plishker and K. Keutzer: “NP-Click: A Programming Model for the Intel IXP1200”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 100–111 (2003).
- [37] Intel Corp.: “Intel Microengine C Compiler Support: Reference Manual” (2002).
- [38] G. Memik and W. H. Mangione-Smith: “NEPAL: A Framework for Efficiently Structuring Applications for Network Processors”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 112–124 (2003).
- [39] A. T. Campbell, S. T. Chou, M. E. Kounavis, V. D. Stachtos and J. Vicente: “NetBind: A Binding Tool for Constructing Data Paths in Network Processor-Based Routers”, Proceedings of The Fifth IEEE Conference on Open Architectures and Network Programming (OPENARCH'03), New York, NY (2002).
- [40] A. Srinivasan, P. Holman, J. Anderson, S. Baruah and J. Kaur: “Multiprocessor Scheduling in Processor-based Router Platforms: Issues and Ideas”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 48–62 (2003).
- [41] J. Wagner and R. Leupers: “C Compiler Design for an Industrial Network Processor”, Proceedings of ACM Workshop on Languages, Compilers, and Tools for Embedded Systems, Snowbird, Utah, pp. 155–164 (2001).
- [42] J. Kim, S. Jung, Y. Paek and G. R. Uh: “Experience with a Retargetable Compiler for a Commercial Network Processor”, Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, ACM, pp. 178–187 (2002).
- [43] J. Wagner and R. Leupers: “Advanced Code Generation for Network Processors with Bit Packet Addressing”, Proceedings of the 1st Workshop on Network Processors, Boston, MA (2002).
- [44] A. Kind, R. Pletka and B. Stiller: “The Potential of Just-in-Time Compilation in Active Networks based on Network Processors”, Proceedings of IEEE OPENARCH'02, New York, NY, pp. 79–90 (2002).
- [45] R. Thomas, B. Mark, T. Johnson and J. Croall: “High-speed Legitimacy-based DDoS Packet Filtering with Network Processors: A Case Study and Implementation on the Intel IXP1200”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 133–147 (2003).
- [46] R. Thomas, B. Mark, T. Johnson and J. Croall: “NetBouncer: Client-legitimacy-based High-performance DDoS Filtering”, Proceedings of the Third DARPA Information Survivability Conference and Exposition (DISCEX III), Washington, D.C (2003).
- [47] M. E. Kounavis, A. Kumar, H. Vin, R. Yavatkar and A. T. Campbell: “Directions in Packet Classification for Network Processors”, Proceedings of the 2nd Workshop on Network Processors, Anaheim, California, pp. 148–157 (2003).
- [48] 長田, 東海林, 山下, 塩川: “マルチメディアコンテンツ転送向け高性能 TCP/IP 通信ボードの構成と評価”, 情報処理学会論文誌, **39**, 2, pp. 347–355 (1998).
- [49] 上村, 熊谷: “Active Network Storage のネットワーク性能の評価”, 電子情報通信学会論文誌 D-I, **J85-D-I**, 9, pp. 841–849 (2002).
- [50] G. Memik and W. H. Mangione-Smith: “A Flexible Accelerator for Layer 7 Networking Applications”, Proceedings of the 39th Design Automation Conference (DAC'02), pp. 646–651 (2002).
- [51] G. Memik, S. O. Memik and W. H. Mangione-Smith: “Design and Analysis of a Layer Seven Network Processor Accelerator Using Reconfigurable Logic”, Proceedings of the 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'02) (2002).
- [52] IP Infusion Inc.: “Teja and IP Infusion IXP1200 Integrated Router Application” (2002). <http://www.ipinfusion.com/pdf/AN101.pdf>