

広帯域ネットワークにおける Linux 上の Web サーバの挙動分析

篠崎 隆志 河合 栄治 山口 英 山本 平一

奈良先端科学技術大学院大学 情報科学研究科

概要

Web サーバの性能評価では、インターネット上で起こる様々なネットワーク特性の影響を考慮する必要があるが、現実には実環境と大きく異なる条件下での性能評価が行われている場合も多い。本研究では、実験環境下でネットワークエミュレートを行い、異なるアーキテクチャを持つ Web サーバである Apache, thttpd, TUX を対象に性能評価と挙動分析を行った。その結果、TUX では CPU 使用率が 100% に到達した時に応答時間が他サーバと比べ大幅に増大する、thttpd はパケットロスによりリクエスト応答レートが大幅に低下する、といった挙動が明らかになった。

Performance Analysis of Linux Web Server in Broadband Network

Takashi Shinozaki Eiji Kawai Suguru Yamaguchi Heiichi Yamamoto

Graduate School of Information Science, Nara Institute of Science and Technology

Abstract

In spite of considering only to server status, network condition is essential when we measure server performance precisely. In this paper, we show the results of benchmarking tests on three different web servers, i.e. Apache, thttpd and TUX in network emulation. We found a fact that the mean reply time of the TUX server was extremely high when compared to those of Apache and thttpd. On the other hand, the mean reply rate of the thttpd server was damaged by network packet losses.

1 はじめに

ネットワークの広帯域化やインターネット利用者数の増加に伴い、Web サーバに対するアクセス数は年々増大している。このため、サーバ処理性能を向上させるための数多くの研究や製品開発が行われてきている。このような背景から、各技術の判断基準となるサーバ性能評価手法は今後ますます重要になると考えられる。しかし、現在利用されている性能評価手法の多くは、インターネット上で起こる様々なネットワーク環境を考慮していないため、実運用に用いたときに評価値通りのスペックや望んだ挙動が得られない可能性がある。

本研究では、パケットロスや遅延などのネットワーク環境特性を考慮したベンチマークを行うことで、サーバ挙動のより詳細な分析を目指す。本論文では、実験環境下で複数の異なるアーキテクチャを持

つ Web サーバについて性能測定と挙動の分析を行った。この研究の目的は、サーバ挙動をより正確に把握することによって、高速なサーバアーキテクチャの開発に役立てることにある。

2 既存研究のアプローチ

サーバのスループット向上手法に関してはこれまでに様々な研究がなされてきている。本節では、まずいくつかの技術について概要と評価手法について述べ、次に評価手法における問題点を明らかにする。

2.1 既存研究のインターネットへの適用

ゼロコピーは、データ転送における一連の動作において、コピー処理にかかっているコストに着目し考案された技術である。利点としてはデータパスを改善することによるサーバの CPU 負荷軽減が挙げ

られ、測定では高いスループット向上の結果が得られている [1]。ただし、これは単一コネクションからのデータパス変更に関する評価であり、複数コネクションからの要求に対してどの程度の有効性が得られるかについては述べていない。

Interrupt Coalescing は、大量のパケットが到着した際の割り込み通知による負荷を軽減させる技術である。この技術はパケット到着時に発生する割り込みに着目し、一定時間または一定量毎にまとめて割り込み通知を行うものであり、広帯域ネットワークにおける大容量のデータ受信に有効である [2]。この手法はパケット到着頻度が少ない場合には逆にオーバーヘッドが大きくなるため、インターネット上に多く存在するリクエストの少ない Web サーバには効果が得られないとも考えられる。

他にも TCP/IP の処理自体を高速なハードウェアにオフロードするという考えも昔からあり [3]、今日まで研究されている。これらにより TCP/IP の処理自体の高速化、TCP/IP 処理分の CPU の負荷軽減が達成できている。一方、オフロードによるアプローチは期待されたほどの性能向上は見込めないという報告 [4] や、コネクション数に対するスケーラビリティが乏しいといった問題も指摘されており、インターネット上で動作する Web サーバとしての利用は難しいのが現状である。

2.2 評価手法の問題点

既存研究におけるサーバ性能評価では、インターネット上で起こりうる様々なネットワーク環境を考慮していないものが多い。インターネット上では、クライアントとサーバの間でパケットを転送するとき、遅延やパケットロスなどが発生する。しかし、実験環境やエミュレーターによるサーバ性能評価では、クライアントとサーバがスイッチを介して直接つながれたごく単純なネットワーク構成や、netperf^{*1}を用いた単一コネクションでのベンチマーク速度を評価値としているような例が多く見受けられる。このようにして得られた評価値と同様の結果を実環境下でも得られるとは考えにくい。

クライアントとの TCP 通信中にパケットロスやデータ遅延が発生すると、ACK 到着までバッファ内にデータが保持されるためサーバにおける処理負荷

が増大する。Web サーバではさらに様々な要素が処理性能に影響する。リクエストされるコンテンツのサイズが大きいほど、データ転送におけるコピー処理が大きな割合を占めるようになり、コネクション時間も長くなる。

このように、インターネット環境下で動作するサーバにおいて、サーバ性能とネットワーク環境は非常に密接な関係があるといえる。より正確で実践的な Web サーバ性能評価を行うためには、パケットロスやデータ遅延、コンテンツサイズやリクエストレートなどの影響を十分に考慮することが重要である。

3 Web サーバのアーキテクチャ

本研究では代表的なアーキテクチャモデルから、Apache^{*2}、thttpd^{*3}、TUX^{*4}を対象に測定を行った。以下にそれぞれの Web サーバの特徴を説明する。

3.1 Apache

Apache は現在最も広く利用されている Web サーバであり、マルチプロセスで動作する。1 つのコネクションに対して 1 つのプロセスを割り当てるため、リクエスト毎にプロセスを生成するオーバーヘッドが問題となるが、あらかじめプロセスを生成しておくプリフォーク (prefork) という手法によりオーバーヘッドの削減を実現している。しかし、多数のプロセスを生成した状態ではスケジューリングのオーバーヘッドが大きいことが問題となっており、他のモデルのサーバよりスケーラビリティが低いのが現状である。

3.2 thttpd

thttpd はシングルプロセスで動作するよう設計された Web サーバである。シングルコネクションで複数のコネクションの多重処理を行うことにより、プロセス切り替えのオーバーヘッドを一切無くしている。しかし、多重処理ソケットの状態走査など、コネクション管理段階における処理の重さが問題となっている。

^{*2} <http://www.apache.org/>

^{*3} <http://www.acme.com/software/thttpd/>

^{*4} <http://www.redhat.com/docs/manuals/tux/>

^{*1} <http://www.netperf.org/netperf/NetperfPage.html>

表 1 ベンチマークテストで利用した機器

Web サーバ	PentiumIII 1GHz, 512MB, ELECOM LD-GT32, Linux 2.6.11smp
クライアント (8 台)	Xeon 2.4GHz (HT 有効), 1GB, Broadcom BCM5702, Linux 2.4.20-8smp
ネットワークエミュレータ	Pentium4 2.8GHz (HT 有効), 512MB, Intel(R) PRO/1000 *2, Linux 2.4.20-8smp

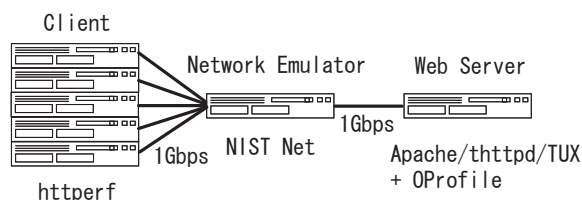


図 1 ベンチマークテスト環境

3.3 TUX

従来方式のほとんどの Web サーバはユーザレベルで動作し、デーモンによる処理を行っているが、たとえユーザレベルでの処理が迅速であっても、カーネルレベルへの処理の引渡しにおけるオーバーヘッドが問題となってくる。このオーバーヘッドを削減するため、カーネル組み込み Web サーバが考案、実装されている。TUX (Red Hat Content Accelerator) は Red Hat 系のディストリビューションで動作する高速なカーネルベースの Web サーバである。

4 測定ネットワーク構成

実験では、Web サーバに Apache, thttpd, TUX を用い、実験環境下でベンチマークによる性能測定を行った。実験機の構成を図 1 に、各機器の詳細な情報を表 1 に示す。8 台のクライアントからサーバに対してリクエストを生成、送信し、リクエスト応答状況を観測する。ネットワーク上での様々な状態を再現するために、サーバとクライアント群の間に PC ルータを設置し、Linux 上で動作する NIST Net^{*5}を用いてネットワークエミュレートを行った。パケットロス率を 0%, 1%, 4%, ネットワーク遅延を片方向 0ms, 50ms を両方向について設定した。各機器はギガビットイーサネットにより接続されている。

^{*5} <http://snad.ncsl.nist.gov/itg/nistnet/>

クライアントでのリクエスト生成には httpperf^{*6}を用いた。httpperf はサーバ上のコンテンツに対して HTTP リクエストを送信し、応答状況から Web サーバの性能を測定するツールである。リクエスト生成レートやタイムアウトまでの時間などは自由に設定することができる。httpperf の出力にはリクエスト応答状況の他、TCP コネクション継続時間の中央値や標準偏差などがある。実験では、サーバ上に 16KB, 1MB と 2 種類のコンテンツを用意し、それぞれのコンテンツに対して応答時間 (Reply Time), 応答レート (Reply Rate), 応答率の標準偏差に関するベンチマークを行った。

また、カーネルプロファイラ OProfile^{*7}を用いて、Web サーバの挙動を調べた。OProfile は割り込み処理の検出が可能であり、サンプリングによるプロファイルが行えることから、サーバにおける処理状態を詳細に把握するのに適している。実験では、各 Web サーバのプロファイリング結果を負荷状態の指標とした。

5 測定結果

実験の結果を図 2 から図 11 に示す。

16KB のコンテンツについて、リクエストレートと応答レートを図 2, 図 3 に、リクエストレートと応答時間を図 4, 図 5 に示した。ネットワークパラメータは、ネットワーク遅延を 0ms, 100ms とし、それぞれパケットロス率 0%, 1%, 4% の場合について測定した。

図 6 は、Apache, thttpd, TUX の 3 つの Web サーバでの OProfile によるプロファイル結果である。リクエストレートの増加に伴う CPU 使用率と、使用率の内訳の変化をグラフで表した。紙面の都合上、ここではネットワーク遅延 0ms, パケットロス率 0% の場合の結果のみ示す。

同様に、1MB のコンテンツについて示したものが図 7 から図 11 である。

サーバの応答レートに着目すると、図 2, 図 3 から、コンテンツサイズ 16KB の場合には TUX が最も多くのリクエストを処理できていることが分かる。次に処理能力が高いのが thttpd で、Apache はネットワークの状態にも依存するが、毎秒約 1000 リクエスト

^{*6} <http://www.hpl.hp.com/research/linux/httpperf/>

^{*7} <http://oprofile.sourceforge.net/>

トが限界となっている。一方、図 7, 図 8 から、コンテンツサイズ 1MB の場合には、サーバによる応答率の差はほとんど観測されなかった。ここで図 6 と図 11 を比較すると、コンテンツサイズ 16KB の場合は CPU 利用率の内訳がサーバ毎に異なる傾向を示しているのに対し、コンテンツサイズ 1MB の場合はどのサーバでも CPU 利用率はほぼ同じ内訳となっている。これは、コンテンツサイズが大きければデータコピーやチェックサムなどによる処理時間が長くなり、カーネル内での処理に CPU リソースが多く割り当てられた結果、サーバ実装による応答率の差が現れにくかったためと考えられる。

パケットロスにより、応答レートが著しく悪化しているのが thttpd である (図 2, 図 3, 図 7, 図 8)。例えば、コンテンツサイズ 16KB, 遅延 0ms, パケットロス率 0%(図 2 左)の場合とパケットロス率 4% の場合 (図 2 右), 毎秒 3200 リクエストであるときの応答レートを比較すると、TUX では 12% の低下, Apache では 49% の低下であるのに対し、thttpd では 72% もの低下が見られた。この傾向は遅延が大きくなると強くなり、遅延 100ms(図 3 左, 図 3 右)の場合に同様の比較をすると、thttpd の応答レートは 89% の低下がみられ、Apache より thttpd の応答レートの方が低くなっていることが分かる。このことから、パケットロスの多い環境では thttpd は予想以上のサービス品質低下を起こす可能性がある。

サーバにおける CPU 使用率と応答時間にも深い相関が見られる。図 6 より、CPU 使用率が 100% に達するのは Apache では毎秒 960 リクエスト, thttpd では毎秒 1920 リクエスト, TUX では毎秒 3200 リクエストの時点である。これを応答時間 (図 4) と照らしあわせると、いずれのサーバも CPU 使用率が低い場合は応答時間は非常に短い, CPU 使用率が 100% に達した時に急激に応答時間が増大している。中でも特に TUX の応答時間増大は顕著である。この傾向はコンテンツサイズ 1MB(図 9)の場合でも見られ、毎秒 120 リクエスト発生時には他のサーバより 19 秒から 25 秒程度のサービス遅延の差が発生している。これより、コンテンツサイズにかかわらず、TUX は CPU 使用率が 100% の状態では応答時間が急激に増加することが分かる。

各サーバの応答レートについて標準偏差の算出も行った。これより、Apache, thttpd に比べて TUX の標準偏差の値が大きいことが分かった。これは今後、

各サーバにおける処理の特徴が応答レートや応答時間にどのような影響を与えるか知るためのひとつの指標になると考えられる。

6 まとめ・今後の展望

本研究ではインターネット上で起こりうるネットワーク状況をエミュレートし、アーキテクチャの異なる Web サーバの挙動を観測、分析した。その結果、コンテンツサイズが大きければいずれのサーバでも応答レートには差がみられないという点、thttpd ではパケットロスが発生すると応答レートに大きな影響を受けるといふ点、サーバでの CPU 使用率が 100% に達したとき応答時間が最も増大するのは TUX であるという点に関して知見を得ることができた。

今後はサーバ実装も参照しながらひきつづきカーネルプロファイリングを行い、より詳細なサーバ挙動を分析する必要があると考えている。

参考文献

- [1] Erich Nahum, Tsipora Barzilai, and Dilip D. Kandlur. Performance Issues in WWW Servers. *IEEE/ACM Trans. Netw.*, Vol. 10, No. 1, pp. 2–11, 2002.
- [2] Jeffrey C. Mogul and K. K. Ramakrishnan. Eliminating Receive Livelock in an Interrupt-Driven Kernel. *ACM Transactions on Computer Systems*, Vol. 15, No. 3, pp. 217–252, 1997.
- [3] D. Clark, Van Jacobson, J. Romkey, and H. Salwen. An Analysis of TCP Processing Overhead. *IEEE Communications*, Vol. 27, pp. 23–29, June 1989.
- [4] G. Regnier, S. Makineni, I. Illikkal, R. Iyer, D. Minturn, R. Huggahalli, D. Newell, L. Cline, and A. Foong. TCP Onloading for Data Center Servers. *Computer*, Vol. 37, No. 11, pp. 48–58, 11 2004.

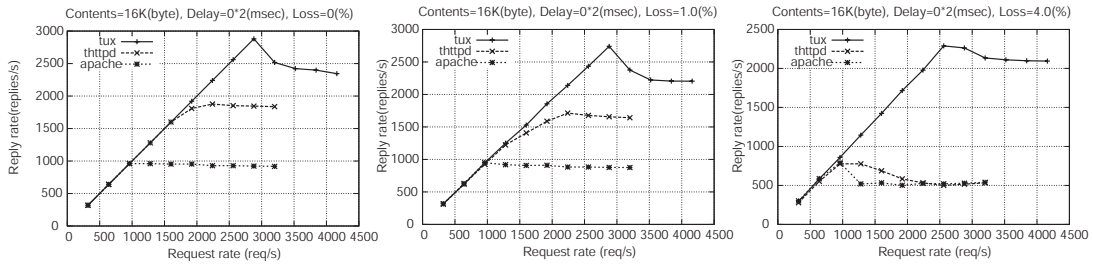


図2 Reply Rate (size=16KB, RTT=0ms, loss:左 0%, 中 1%, 右 4%)

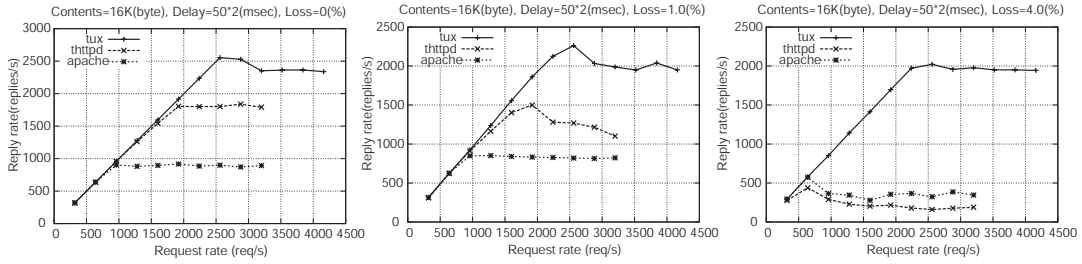


図3 Reply Rate (size=16KB, RTT=100ms, loss:左 0%, 中 1%, 右 4%)

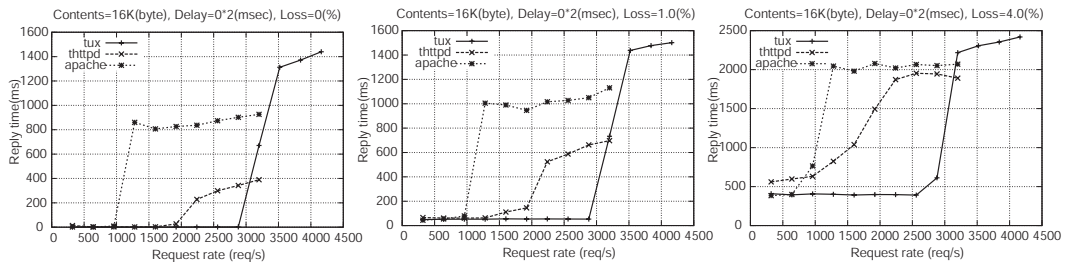


図4 Reply Time (size=16KB, RTT=0ms, loss:左 0%, 中 1%, 右 4%)

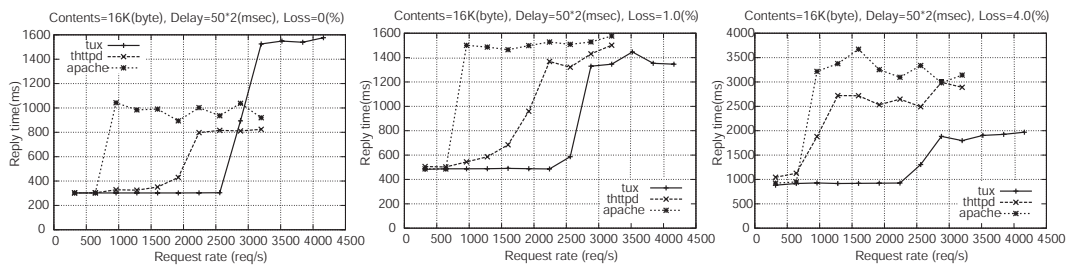


図5 Reply Time (size=16KB, RTT=100ms, loss:左 0%, 中 1%, 右 4%)

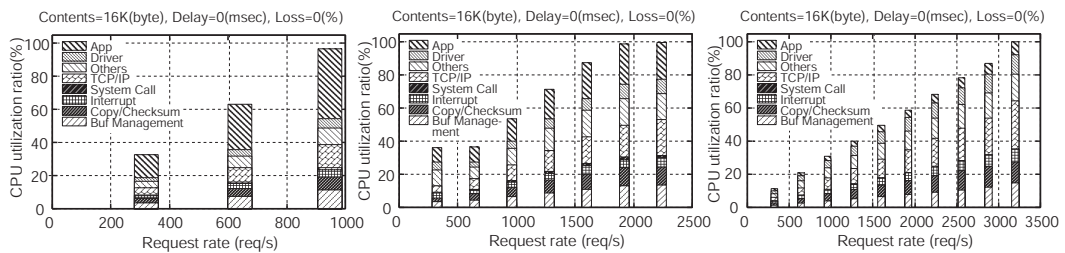


図6 プロファイルの結果 (size=16KB, RTT=0ms, loss=0%, 左:Apache 中:thttpd 右:TUX)

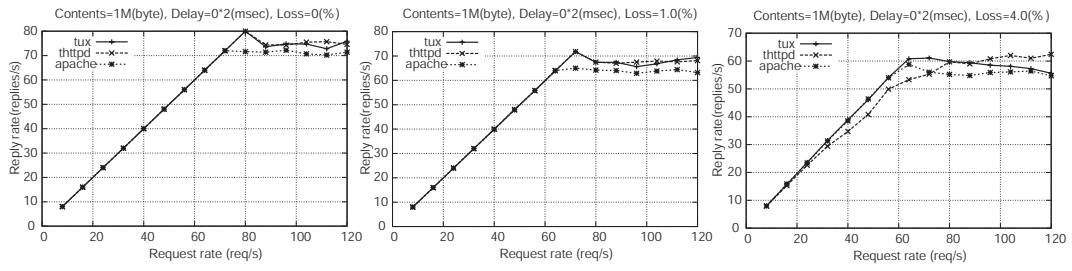


図 7 Reply Rate (size=1MB, RTT=0ms, loss:左 0%, 中 1%, 右 4%)

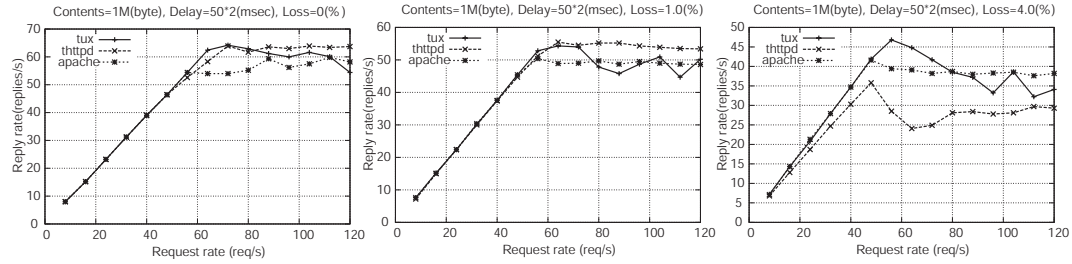


図 8 Reply Rate (size=1MB, RTT=100ms, loss:左 0%, 中 1%, 右 4%)

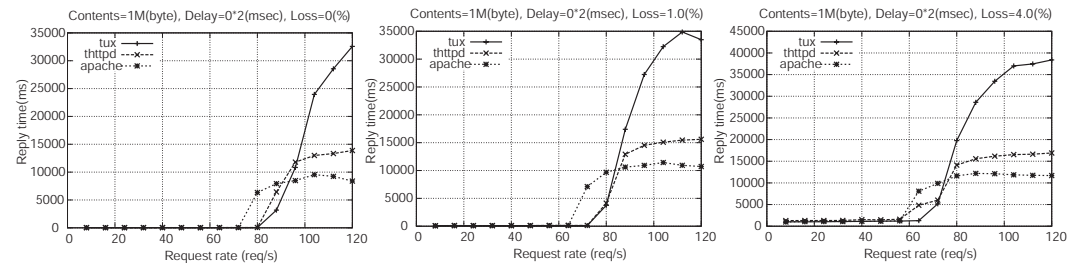


図 9 Reply Time (size=1MB, RTT=0ms, loss:左 0%, 中 1%, 右 4%)

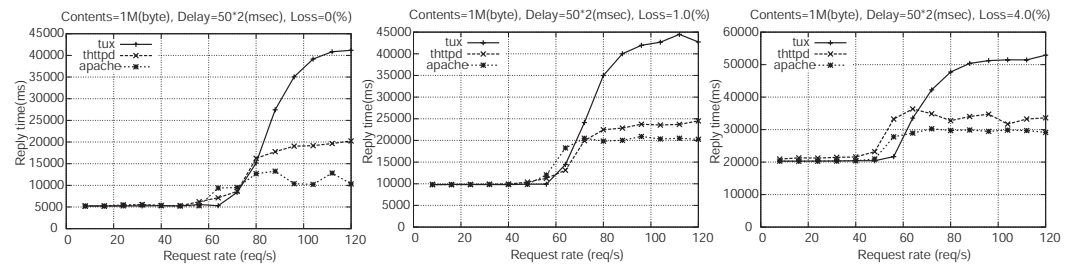


図 10 Reply Time (size=1MB, RTT=100ms, loss:左 0%, 中 1%, 右 4%)

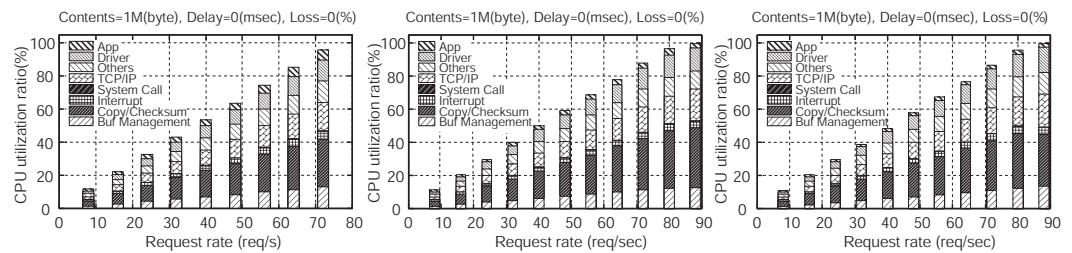


図 11 プロファイルの結果 (size=1MB, RTT=0ms, loss=0%, 左:Apache 中:thttpd 右:TUX)