

HTTP-FUSE-KNOPPIX-BOX の キャッシュ導入による高速化と評価

金井 遵[†] 須崎有康^{††}
八木豊志樹^{††} 並木美太郎[†]

従来の HTTP-FUSE-KNOPPIX では、起動のたびにレイテンシの長いインターネット上のサーバからブロックデバイスのデータを取得するため、OS 起動に長い時間を要した。そこで、本研究では、容易に持ち運びができるモバイル環境をルータ化し、データを HTTP プロキシにキャッシュすることで HTTP-FUSE-KNOPPIX を高速化、およびどこでも起動できるようにし、評価を行った。さらに、NFS ベースでは困難だったシンクライアントサーバの拡張も HTTP-FUSE-KNOPPIX ではキャッシュサーバを追加し、動的に分散することで、複数起動にも強い設計とした。評価より、実際に従来の HTTP-FUSE-KNOPPIX や、CD 起動の KNOPPIX、NFS によるネットワークブートよりも高速に起動できることが確認された。

Performance Improvement and Evaluation of HTTP-FUSE-KNOPPIX-BOX by Cache Servers

JUN KANAI,[†] KUNIYASU SUZAKI,^{††} TOSHIKI YAGI^{††}
and MITARO NAMIKI[†]

Traditional HTTP-FUSE-KNOPPIX loads system files through high latency internet server every time, therefore it takes long time for bootstrapping. In this research, we improved and evaluated performance of HTTP-FUSE-KNOPPIX by caching data on HTTP proxy at a mobile router. Furthermore we designed the router which can boot up KNOPPIX everywhere. Moreover, we implemented plural cache servers and a dynamic cache server balancer to boot up many client machines. From evaluations, the bootstrapping time of HTTP-FUSE-KNOPPIX with cache servers becomes faster than that of bootstrapping with a CD, NFS mount and traditional HTTP-FUSE-KNOPPIX.

1. はじめに

近年の Personal Computer(PC) 性能の向上と低価格化、PC が占める重要性の増加に伴い、企業や学校において大量のクライアントマシンを導入する重要性が高まっている。しかし、従来は OS などのシステムファイルをクライアントマシンの HDD 上に導入し、LAN のファイルサーバにユーザファイルを格納する方法が主流であった。この方法ではクライアントマシンごとにシステムファイルのメンテナンスを行う必要があるため、システムの管理コストが非常に大きかった。システムソフトウェアを含めた全ての情報をネット

ワークで一括管理を行うシンクライアントはこの問題を解決できるため、最近、この方式を採用する企業や学校は増加している。しかし、従来の NFS ベースのシンクライアントでは NFS サーバがボトルネックになり、拡張性に乏しい。本研究で用いる HTTP-FUSE-KNOPPIX はブロックデバイスのデータを HTTP で配信するので、HTTP プロキシの活用が可能になり、拡張性を有する。従来はインターネットから直接データをダウンロードしていたので起動時間に問題があったが、本提案ではルータに HTTP プロキシの機能を加え、この問題を解決した。

また、本方式はクライアントの増加により HTTP プロキシを複数用意することで簡単に負荷分散も可能である。HTTP ベースのためネットワークはセッションレスであり、DNS サーバの引き直しを利用して動的にプロキシサーバを変えることが可能になった。さらに、従来の NFS ベースでは全データを含んだサーバを

[†] 東京農工大学 工学府 情報工学専攻
Tokyo University of Agriculture and Technology
^{††} 独立法人 産業技術総合研究所
National Institute of Advanced Industrial Science and
Technology

表 1 NFS マウントと HTTP-FUSE-KNOPPIX の比較

	NFS マウント	HTTP-FUSE-KNOPPIX
サーバ プロトコル トラフィック 拡張性	DHCP, TFTP, NFS NFS(UDP/WAN に拡張が難しい) 圧縮は基本的に無し 拡張性はあるが、フルコンテンツの NFS サーバ が必要で大規模化、管理コスト増大。サーバの 変更不可	DHCP, TFTP, HTTP HTTP(TCP/セッションレス) ブロック単位 (256KB) 圧縮がある HTTP プロキシを利用しサーバの小規模化可能、 管理容易。keep-alive により動的なサーバ追加 削除可。

用意すれば拡張可能であったが、本方式ではプロキシキャッシュでよく、全データを含められる二次記憶を用意する必要が無い。このため、サーバの小型化が可能になった。本論文ではこの HTTP-FUSE-KNOPPIX をルータボックスの機能に加えた方式を提案し、その性能について述べる。

2. KNOPPIX

本章では、CD/DVD 起動可能な Linux である KNOPPIX⁵⁾、シンクライアントを実現する HTTP-FUSE-KNOPPIX について述べる。

2.1 KNOPPIX

KNOPPIX は CD/DVD のみでブート可能な Linux ディストリビューションである。従来、面倒だった OS のインストールは KNOPPIX では必要なく、手軽に Linux を利用することができる。また、CD/DVD を用意すれば多くの統一ソフトウェア環境が提供可能であるため、複数のクライアントで同一環境を提供したい場合や、クライアントマシンの起動ごとに環境を変更したくない場合などに効果を発揮する。

しかし、CD/DVD 起動の KNOPPIX では、メディアの用意が必要であったため、クライアントの台数が極度に多い場合や、頻繁にルートファイルシステムが更新される場合において利用に適さなかった。これらの問題を解決するため、KNOPPIX にはクライアントマシンをネットワークブートし、NFS をマウントするための機能が備わっており、複数のクライアントへ同一環境が提供可能である。しかしながら、従来のシンクライアントの問題と同様に単一のサーバから多数のクライアントマシンへ NFS をマウントするため性能の問題が存在した。

2.2 HTTP-FUSE-KNOPPIX

HTTP-FUSE-KNOPPIX は、従来 CD/DVD から読み込んでいたルートファイルシステムをインターネット上から取得するように取得するようにした KNOPPIX である。

HTTP-FUSE-KNOPPIX では、ネットワークブートが可能であり、さらにルートファイルシステムをインターネット上から取得するため CD やハードディ

スクなどの二次記憶装置が必要なく、真のシンクライアントが実現可能である。KNOPPIX では、膨大なディスク内容をブロック単位で分割した上で圧縮し、利用時に必要な部分だけ伸張する cloop という機構を搭載している。HTTP-FUSE-CLOOP では、この cloop ファイルをインターネット上から取得するように変更されている。HTTP-FUSE-CLOOP では HTTP を介するため、従来利用するためにファイアウォールの設定が必須だった、WebNFS⁹⁾、CIFS¹⁰⁾ 等と比べて特別な設定を行うことなく利用可能である。また、HTTP-FUSE-KNOPPIX では、HTTP を介するため柔軟な管理が可能である。さらには、インターネット上の代表サーバからデータを取得するため、OS の更新にも意識することなく対応可能である。従来の NFS をマウントする方式と、HTTP-FUSE-KNOPPIX の比較を表 1 に示す。

3. HTTP-FUSE-KNOPPIX の問題点

HTTP-FUSE-KNOPPIX では HTTP を介してインターネット上のファイルを取得するため、ディスク性能はクライアントとサーバ間のネットワークの性能に左右される。特に、須崎らの研究¹⁾によれば、HTTP-FUSE-KNOPPIX の性能はネットワークのレイテンシに大きく影響される。例えば、レイテンシがない環境では 97[s] で起動可能だった KNOPPIX が、レイテンシが 200[ms] の環境では起動に 400[s] かかってしまう。そのため、HTTP-FUSE-KNOPPIX では、レイテンシが最適なサーバからデータを取得する機能が備わっているが、LAN 環境に比べるとレイテンシはまだ大きい。また、この方法ではクライアントの接続回線自体がボトルネックになっている場合には改善が不可能である。インターネットを介する全てのファイルシステムにおいてこの問題は存在するが、CISCO WAFS⁸⁾では、ローカル側にキャッシュサーバを置くことにより性能向上を達成している。HTTP-FUSE-KNOPPIX においても同様の手法により、レイテンシや帯域の改善が必要になると考えられる。

日米間を想定

さらに、HTTP-FUSE-KNOPPIX では、毎回インターネットからデータ取ってくるため、クライアントマシンが多数起動する場合や、頻りにクライアントマシンの起動を行う場合などにはトラフィックが集中し、さらに性能が低下する。多数のクライアントマシンの起動が必要になるシンクライアント環境ではこれは大きな問題である。また、従来の NFS をマウントするネットワークブート方式においても、同様の問題が存在した。これを解決する策として、キャッシュを行うサーバを複数用意し、クライアントはそれぞれのキャッシュサーバの負荷状況に応じて、キャッシュサーバを使い分ける方法が考えられる。

4. 本研究の目標

本研究では、HTTP-FUSE-KNOPPIX がインターネット上からファイルを取得することによる遅延を隠蔽し、さらにクライアントの複数起動による HTTP-FUSE-KNOPPIX の速度低下の改善を目的とする。

そのために、クライアントマシンからネットワーク的になるべく近い場所に cloop ファイルをキャッシュするサーバを置くことにより、cloop ファイル取得時のレイテンシを最適化し、HTTP-FUSE-KNOPPIX の性能を改善する。さらに、このキャッシュサーバを多重化し、適切にバランシングすることにより、多数のクライアントが起動するような場合においても、十分実用に耐えうるような HTTP-FUSE-KNOPPIX 環境を構築することを目標とする。

5. システムの概要

本章では、今回開発した KNOPPIX システムを高速に起動および動作させ、さらに持ち運びが可能なシステム HTTP-FUSE-KNOPPIX-BOX の概要について述べる。

5.1 全体構成

今回開発したシステムは、クライアントマシンで LAN からブートさせるために必要な PXE⁴⁾ サーバ、DHCP サーバ、cloop ファイルをキャッシュするためのキャッシュサーバ、ルータ機能、トラフィックを分散させるためのバランサなどの機能を持った HTTP-FUSE-KNOPPIX-BOX、および任意の cloop ファイル取得時のトラフィックを分散させるキャッシュサーバからなる。システムの全体構成図を図 1 に示す。

HTTP-FUSE-KNOPPIX では、ネットワークブート時にサーバ側の設定ファイルでブートオプションを設定でき、cloop ファイル取得先および、ファイルの取得に利用するプロキシサーバを設定できる。本方式

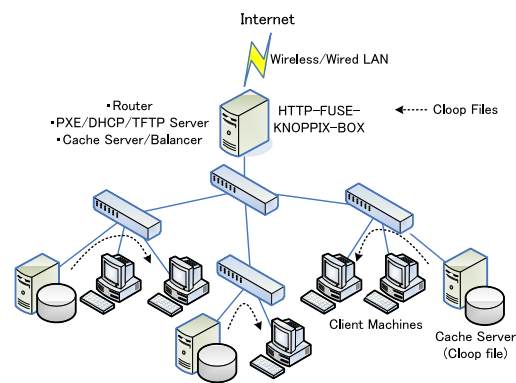


図 1 システムの全体構成図

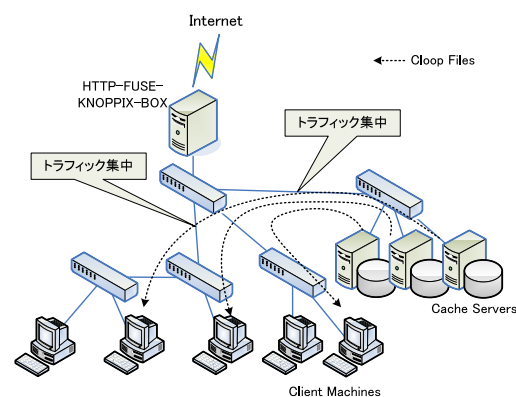


図 2 トラフィックの集中

では、このプロキシサーバに cloop ファイルのキャッシュを置き、キャッシュが存在する場合にはこのキャッシュをクライアントに対して送信し、クライアントはこの cloop ファイルを利用してルートファイルシステムを復元することにより、高速化を図る。さらに、このプロキシサーバを動的に変更することにより、トラフィックの分散を行う。

今回の HTTP-FUSE-KNOPPIX-BOX および、プロキシサーバとバランサによるトラフィック分散において、ローカルネットワーク内で図 2 のようなトラフィックが特定の接続に偏ることは、クライアントのパフォーマンス低下につながる。そのため、図 1 のように特定の接続にネットワークが偏らないように適切にネットワークを設計し、プロキシサーバを配置することでシステム全体の速度向上につながる。

5.2 HTTP-FUSE-KNOPPIX-BOX

HTTP-FUSE-KNOPPIX-BOX では、cloop ファイル取得高速化のためのキャッシュサーバ機能、および、クライアントをネットワークブートさせ Knoppix を起動させるための、PXE/DHCP サーバ機能、ルータ



図 3 L-BOX(Re2)

機能，バランサ機能を提供する。それぞれの機能を実現するために利用したソフトウェアを表 2 に示す。

今回，HTTP-FUSE-KNOPPIX-BOX を実装するハードウェアと，想定する利用用途は表 3 とおりである。このように HTTP-FUSE-KNOPPIX-BOX は要求するハードウェアスペックが低い。従来のシンクライアントを実現するためのサーバは巨大なシステムが多かったが，HTTP-FUSE-Knoppix では，ローエンド PC や，組み込み用途で利用するような環境でも動作可能である。さらに最低 267g と，容易に持ち運びも可能となっているため，どこでも KNOPPIX 環境が利用可能である。さらに，VAIO typeU や L-BOX ではインターネット側のネットワーク接続を IEEE 802.11b/g により行っており，ワイヤフリーな KNOPPIX ルータ環境を構築可能である。

6. キャッシュサーバ

今回，HTTP-FUSE-KNOPPIX の高速化を目的として，WAN 側のレイテンシと帯域の遅さの隠蔽のためにプロキシサーバと，トラフィックの分散のためにプロキシサーバの動的分散機構を導入した。本章ではこれらの詳細について述べる。

6.1 キャッシュサーバ

LAN 環境にキャッシュサーバを導入することにより，cloop ファイル取得時にキャッシュが無い場合にはインターネット上のサーバからデータが取得され，キャッシュサーバ上にキャッシュされるようになる。次回以降，キャッシュが存在する場合には，インターネット側への通信無しに，このキャッシュを利用する。ク

表 2 HTTP-FUSE-KNOPPIX-BOX ソフトウェア概要

キャッシュサーバ機能	Squid
PXE サーバ機能	TFTP
DHCP サーバ機能	DHCPD
ルータ機能	iptables
バランサ機能	DNS Balance/自作バランサ

ライアント側ではキャッシュの有無を意識する必要はなく，透過的にアクセスが可能である。これにより，従来問題だったレイテンシや帯域利用率の改善ができ，HTTP-FUSE-KNOPPIX の起動時間を改善することが可能である。

前述の通り HTTP-FUSE-CLOOP は，HTTP を利用して取得される。そのため，キャッシュサーバの導入には，通常の HTTP キャッシュサーバを利用することが出来る。今回はキャッシュサーバ・プロキシサーバとして一般的に利用される Squid⁷⁾ をキャッシュサーバとして利用した。Squid を利用することで，キャッシュ更新間隔等の柔軟な設定が可能である。

また，HTTP-FUSE-KNOPPIX-BOX がインターネットに繋がっていない環境下でも，キャッシュが存在すれば，このキャッシュを利用して HTTP-FUSE-KNOPPIX が起動可能である。その結果，クライアントはインターネット接続の有無やサーバ設定を意識することなく，どこでも KNOPPIX 環境を利用することが出来る。また，自動的に大本の cloop ファイルが更新された場合には，自動的にキャッシュサーバのデータを更新することが可能であり，従来大きな手間となっていた OS の更新等への管理が不要である。

6.2 キャッシュサーバの負荷分散

従来の NFS をマウントするタイプの PXE ブートでは，動的なトラフィックの分散が不可能であった。HTTP-FUSE-KNOPPIX では，後述する評価より，100BASE-TX の LAN 環境ではクライアント 1 台の場合においても HTTP-FUSE-KNOPPIX-BOX とクライアント間の帯域を使い切ってしまう場合があることが分かった。そのため，クライアントが複数台になった場合に cloop ファイル取得にかかる時間が長くなり，その結果，HTTP-FUSE-KNOPPIX 自体の動作が緩慢になってしまうと考えられる。そのため，本研究では HTTP-FUSE-KNOPPIX-BOX の他に，cloop ファイル取得用のキャッシュサーバを複数台用意し，自動的にバランシングできるようにする仕組みとした。また，キャッシュサーバを複数用意することにより，コンピュータを構成する部品の中でも故障率の高いハードディスクを多重化することになり，信頼性の向上につながる。

設計の方針として，クライアントでの設定が必要なく自動的に利用するプロキシサーバが設定されること，システム全体をダウンする必要無しに，動的にプロキシサーバの追加削除が行うことができることとした。今回，負荷分散を行う方式として以下の 2 方式を用意した。

表 3 HTTP-FUSE-KNOPPIX-BOX ハードウェア

	VAIO typeU	Be-Silent	L-BOX(RE2)/図 3
コンセプト	ワイヤフリーなモバイル環境を想定	低スペック PC/モバイル環境を想定	組込み用途/ワイヤフリーなモバイル環境を想定
プロセッサ	CeleronM 900MHz	Eden ESP 6000 667MHz	SH-4 240MHz
システムメモリ	512MB	256MB	64MB
ネットワーク	無線 LAN/有線 LAN	有線 LAN*2	無線 LAN/有線 LAN*2
記憶装置	HDD	HDD	CompactFlash
重量/消費電力	550g/24W	1200g/30W	267g/3W

● 自作バランサによる負荷分散

HTTP-FUSE-KNOPPIX では PXE ブート時の cloop ファイル取得に利用するプロキシサーバを PXE サーバの設定を書き換えることで指定可能である。よって、図 4 のようにこのプロキシサーバのアドレスをクライアントごとに変更することにより、トラフィックや CPU 負荷の分散が可能である。

本研究で作成したバランサは、各クライアント上で動作し、クライアントのネットワーク負荷を取得して HTTP-FUSE-KNOPPIX-BOX サーバプログラムに送信するクライアントプログラム、および HTTP-FUSE-KNOPPIX-BOX 上で動作し各クライアントからの負荷状況を受け取り、クライアントのキャッシュサーバへの割り当てを決定するサーバプログラムからなる。サーバプログラムは下記式で、プロキシサーバの優先度を算出する。ここで、 L_i をキャッシュサーバ i の優先度 (小さいほど優先度が高い)、 N_i を 10 段階のネットワークインタフェース利用率、 C_i を既に割り当てたクライアント数とする。 a は定数であり、現在 1 として実験している。ネットワークインタフェース利用率で優先度を決定しているのは、後述する実験で多くの場合に置いてネットワークのトラフィックがサーバのボトルネックになるためである。

$$L_i = N_i + a * C_i \quad (1)$$

この式で、優先度を算出し、 $\min(L_i)$ のキャッシュサーバに対してクライアント 1 台を割り当てる。これをクライアントの数だけ繰り返し、全てのクライアントがどのキャッシュサーバに割り当てられるかが決定される。

また、初めて負荷の報告があった場合に動的にプロキシサーバが追加され、一定時間報告がないとプロキシサーバのリストから削除されるため、プロキシサーバの動的な追加削除が可能な設計になっている。

● DNSBalance による負荷分散

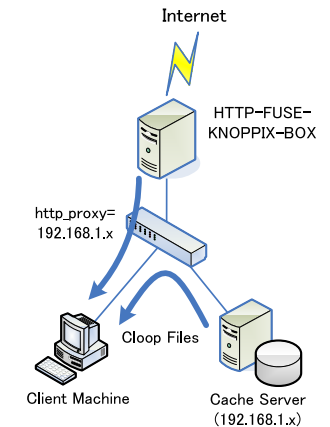


図 4 キャッシュサーババランサの動作

DNS Balance とは DNS が返す IP アドレスを動的に変更することにより、トラフィックの動的な分散を行なうことを目的とした DNS サーバプログラムである。

HTTP-FUSE-CLOOP では keep-alive 機能により、一定間隔毎にデータ取得先のサーバの IP アドレスが引き直される。そのため、DNS が返す IP アドレスを毎回変えることにより動的にキャッシュサーバのバランシングを行うことが可能である。HTTP-FUSE-KNOPPIX では上で述べたとおり、cloop ファイル取得に利用するキャッシュサーバを PXE サーバの設定を書き換えることで指定可能である。図 5 のように PXE ブートの設定でキャッシュサーバを適当な名前指定し、DNS Balance により、この名前に対して返されるキャッシュサーバの IP アドレスをランダムに変える。このプロキシサーバごとの IP アドレスを返す確率を設定することが可能であるため、プロキシサーバの能力に応じて柔軟なバランシングが可能である。また、DNS Balance 単体ではプロキシサーバの動的な追加削除は行えないが、アドレスのデータベースファイルが一定時間毎に読み出されるため、自作バランサを併せて利用してこのデータベースを書き換えることで、動的なプロキシサーバの追加

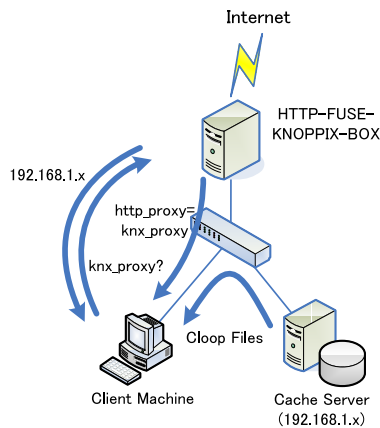


図 5 DNS Balance の動作

削除を実現する。

7. 評価

本章では、HTTP-FUSE-KNOPPIX へのキャッシュ導入による効果の評価、キャッシュサーバの分散による効果について述べる。

7.1 従来方式との起動時間の比較

HTTP-FUSE-KNOPPIX の起動時間は cloop ファイルの取得時間に大きく左右される。そこで、今回のキャッシュ導入を行った HTTP-FUSE-KNOPPIX、従来の CD 起動の KNOPPIX、従来の HTTP-FUSE-KNOPPIX、Knoppix4.02 に標準で実装されている NFS をマウントするネットワークブート機能を使った場合の KNOPPIX のそれぞれの起動時間を計測した。なお、ここでいう起動時間とは BIOS によるブートデバイスの選択から、KDE の全てのアイコンが出て起動が完了するまでの時間である。

評価環境としてクライアントに、表 4 の IBM ThinkPad X30、サーバに Sony typeU による HTTP-FUSE-KNOPPIX-BOX を利用した。今回、HTTP-FUSE-KNOPPIX-BOX からインターネットへの接続回線は OCN を利用し、cloop ファイルの取得先は、産業総合技術研究所 (aist.ring.gr.jp) とした。なお、HTTP-FUSE-KNOPPIX から aist.ring.gr.jp へのラウンドトリップタイムを ping コマンドにより計測したところ、10.5[ms] であった。

起動時間の測定結果を表 5 に示す。

このようにキャッシュサーバにキャッシュが存在する場合、従来のネットワークブート方式、HTTP-FUSE-KNOPPIX、CD 起動の KNOPPIX などに比べ、速度向上を達成した。キャッシュが無い場合の HTTP-FUSE-KNOPPIX は、従来の HTTP-FUSE-

KNOPPIX に比べて遅くなっているが、これはキャッシュサーバへのキャッシュデータの書き込みによるオーバヘッドだと思われる。しかし、このオーバヘッドは小さく、さらに発生するのは大元のデータが更新されないかぎり最初の一度のみであり、大半の起動時間において影響しない。従来の CD 起動の KNOPPIX に比べて高速化を達成したことは、本方式の実用性を示した結果であると言える。

7.2 HTTP-FUSE-KNOPPIX-BOX 環境間での比較

今回、VAIO typeU、Be-Silent、L-BOX を HTTP-FUSE-KNOPPIX-BOX 化した。マシンパワーの無い環境で実用的な HTTP-FUSE-KNOPPIX-BOX が実現できることは HTTP-FUSE-KNOPPIX-BOX のモバイル化の基準となる。今回 HTTP-FUSE-KNOPPIX 化した、typeU、typeU(搭載メモリを 64MB に制限し、スワップを無効にしたもの)、Be-Silent、L-BOX について通常ファイルシステムで起動時間を計測するとともに、L-BOX では cloop ファイルのファイルシステムを最適化した場合の起動時間について計測した。計測結果を表 6 に示す。

結果より、組み込み用途を想定した L-BOX(SH4 CPU) においても 2 分程度と、従来の CD 起動などの方式に比べて高速な起動を実現している。L-BOX は重量が 267g と大変軽量であり、HTTP-FUSE-KNOPPIX-BOX のモバイル化が十分可能である。また、typeU でメモリサイズを 64MB に制限した場合で

表 4 クライアントマシン (IBM ThinkPad X30) 概要

プロセッサ	Pentium III-M 1.20GHz
システムメモリ	DDR-SODIMM 768MB
ネットワーク	100BASE-TX/10BASE-T
CD-ROM Drive	24 倍速 CD-ROM ドライブ

表 5 KNOPPIX 起動時間結果

	起動時間 [s]
NFS マウント	232
従来の HTTP-FUSE-KNOPPIX	187
CD 起動 KNOPPIX	149
HTTP-FUSE-KNOPPIX(キャッシュあり)	115
HTTP-FUSE-KNOPPIX(キャッシュなし)	191

表 6 HTTP-FUSE-KNOPPIX-BOX 環境間起動時間

	キャッシュ有 [s]	キャッシュ無 [s]
typeU	112	191
typeU(メモリ 64MB)	118	220
Be-Silent	115	160
L-BOX(通常 FS)	266	—
L-BOX(最適化 FS)	126	—

も、起動時間は大きく変わらず、ハードウェアリソースが制限された環境でも十分 HTTP-FUSE-KNOPPIX-BOX は実現可能である。

通常ファイルシステムで L-BOX が極端に遅いのは、L-BOX のバスバンド幅の影響である。ファイルシステムを最適化し、アクセスが必要になる部分を単一ブロックファイルにまとめることにより総ファイル読み込みサイズが少なくなり、起動時間が改善されたものと考えられる。

なお、typeU と BeSilent でキャッシュが無い場合に差が有るのは、typeU が WAN 側の接続に無線 LAN を、Be-Silent が有線 LAN を利用しているためだと考えられる。

7.3 キャッシュサーバ分散の効果

続いて、キャッシュサーバを分散させた場合の効果を評価するため、キャッシュサーバの台数、分散方式を変えて、クライアントを複数台にした場合の起動時間を計測した。さらに、HTTP-FUSE-KNOPPIX の速度はレイテンシとともに、ネットワーク帯域の利用状況も影響する。キャッシュおよびバランサ導入の効果を調査するために、ネットワークの利用状況を調べた。起動時間の測定結果を表 7 に、クライアント 1 台の場合のトラフィックの様子を図 6 に、DNS Balance によりトラフィックの分散を行った場合の状況を図 7 にそれぞれ示す。なお、キャッシュサーバとクライアントマシンは、最高それぞれ 2 台ずつを用意して計測を行った。

今回、残念ながら起動時間にキャッシュ分散機構導入による大きな改善は見られなかった。これはクライアントの台数が 2 台と少なく、トラフィックがボトルネックにならなかったためだと考えられる。しかし、トラフィックを測定したグラフによればキャッシュサーバ分散の効果が見られるため、クライアントがさらに増えた場合には起動時間にも効果が現れるものと考えられる。

7.4 先読み効果

HTTP-FUSE-CLOOP においてはレイテンシの問題を改善するために先読み機構 (DLAHEAD 機能) が搭載されている。キャッシュサーバ利用時の先読みの効果を調べるため、先読み用の HTTP コネクション

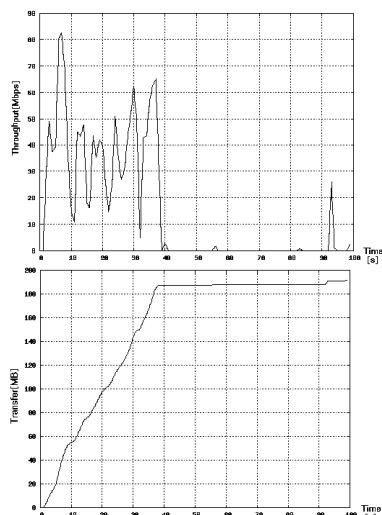


図 6 分散を行わない場合のトラフィック (クライアント 1 台/サーバ 1 台)

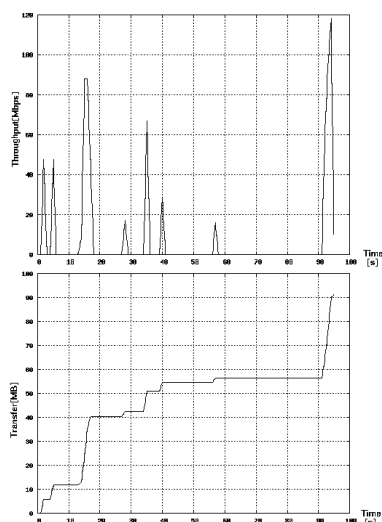


図 7 DNSBalance によるトラフィック (クライアント 1 台/サーバ 2 台)/上段:1 秒平均 下段:合計

数を変えて KNOPPIX の起動時間と、トラフィックの測定を行った。トラフィック状況の結果を図 8, 9 に、起動時間の比較を表 8 に示す。

結果より、先読みありとなして起動時間に大きな差は見られなかった。今回先読みの効果があまり表れないのは、LAN 環境においてはレイテンシが十分に小

表 7 キャッシュサーバ分散時の起動時間

	起動時間 [s]
サーバ 1 台/クライアント 1 台	119
サーバ 1 台/クライアント 2 台	125
サーバ 2 台/クライアント 2 台 (DNSBalance)	123
サーバ 2 台/クライアント 2 台 (自作バランサ)	122

表 8 先読みによる起動時間

	起動時間 [s]
先読みなし	113
先読みあり (最大接続数=8)	112
先読みあり (最大接続数=32)	112
先読みあり (最大接続数=256)	114

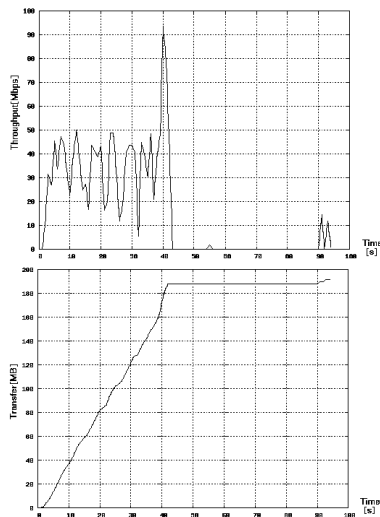


図 8 先読みなしの場合/上段:秒平均 下段:合計

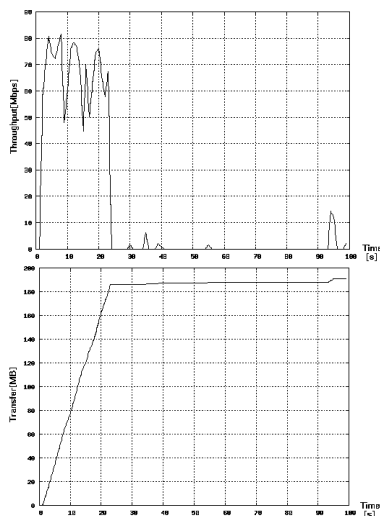


図 9 先読み有りの場合 (最大接続数=256)/上段:秒平均 下段:合計

さく、データ読みだし要求があった場合に、待ち時間が発生しないためだと考えられる。

また、先読みを行った場合では、ほぼ帯域を使いきってしまっている。このため、クライアントが複数になった場合、トラフィックが集中し起動が遅くなると考えられ、今回のプロキシサーバの分散によるトラフィックの分散は有意義だと考えられる。

8. 今後の課題

今回の評価実験では、最大 2 台ずつのクライアントとプロキシサーバでの検証を行った。しかし、シンクライアント環境では大量のクライアントの起動が求められることが多いと考えられる。そのため、今回開発した HTTP-FUSE-KNOPPIX-BOX と、キャッシュ

サーバ、およびキャッシュサーバのバランスのさらなる有用性を示すために、多数のクライアントでの評価実験を今後の課題とする。

また、現在のキャッシュサーバのバランスはプロトタイプであり、LAN の接続形態を考慮できない。適切なトラフィックの分散のためには、どのようにネットワークが繋がっているか、キャッシュサーバがどこに配置されているかを考慮してキャッシュサーバの割り当てを行う必要がある。さらには、クライアントの台数が増えた場合にキャッシュサーバのバランスのアルゴリズムの再考が必要となる可能性がある。こちらも今後の課題としたい。

9. おわりに

本研究では、キャッシュサーバを導入することにより HTTP-FUSE-KNOPPIX がインターネット上からファイルを取得することによる遅延を隠蔽し、HTTP-FUSE-KNOPPIX の高速化を実現した。さらにキャッシュサーバの分散を行うことで、従来のシンクライアントで問題であったクライアントの複数起動によるトラフィックの改善を実現できた。

参考文献

- 1) 須崎, 八木, 飯島, 北川, 田代: ネットワークに対応した分割圧縮ループバックデバイス HTTP-FUSE-CLOOP とそれから起動する Linux, インターネットコンファレンス 2005 .
- 2) 佐々木, 安立, 田村, 安田, 横井: HTTP-FUSE-KNOPPIX を基盤としたサーバ学習環境システムの開発, 教育システム情報学会 第 6 回研究会「情報教育の実績と新しい展開」
- 3) F.Dabek, M.F.Kaashoek, D.Karger, R.Morris, and I.Stoica: Wide-area cooperative storage with CFS, Proc. of the 18th SOSP, Oct. 2001
- 4) Preboot Execution Environment (PXE) Specification <http://www.intel.com/design/archives/wfm/downloads/pxespec.htm>
- 5) KNOPPIX <http://unit.aist.go.jp/itri/knoppix/>
- 6) DNS Balance <http://openlab.jp/dns.balance/>
- 7) Squid <http://www.squid-cache.org/>
- 8) Cisco Wide Area File Services (WAFS) <http://www.cisco.com/japanese/warp/public/3/jp/product/hs/storage/fewafs/>
- 9) WebNFS <http://www.sun.com/software/webnfs/>
- 10) CIFS <http://www.microsoft.com/mind/1196/cifs.asp>