

# HTTP-FUSE KNOPPIX におけるキャッシュを利用した起動手法の提案

後藤和弘<sup>†</sup> 北川健司<sup>‡</sup> 須崎有康<sup>§</sup>

<sup>†</sup>大分県産業科学技術センター <sup>‡</sup>(株)アルファシステムズ <sup>§</sup>(独)産業技術総合研究所

**概要** 近年、学校教育現場や自治体等において OSS デスクトップの有効性が検証されている。Open School Platform の大分市・豊後大野市地域プロジェクトでは、県内高校を対象に CD やネットワークから起動する KNOPPIX を検証した。実証で採用した HTTP-FUSE KNOPPIX は OSS だけでシンクライアント環境を構築でき、サーバからファイルをダウンロードしながら KNOPPIX を起動する。本稿では教室等において多数のクライアントをネットワーク経由で起動する際に、サーバへのトラフィックを軽減するために、クライアントにキャッシュされたブロックファイルを利用する起動手法について提案する。

## Proposal for Boot Method of HTTP-FUSE KNOPPIX using Cached Block Files.

Kazuhiro GOTO<sup>†</sup> Kenji KITAGAWA<sup>‡</sup> Kuniyasu SUZAKI<sup>§</sup>

<sup>†</sup>Oita Industrial Research Institute <sup>‡</sup>Alpha systems Inc. <sup>§</sup>AIST

**Abstract.** Availability of OSS desktop is examined in educational environment and local government recently. KNOPPIX that boots from CD or network were examined in Oita-BungoOono regional project of Open School Platform project. HTTP-FUSE KNOPPIX could be realized Thin-Client environment only with Open Source Software. A client boots KNOPPIX with downloading block files from a server. In this paper, we propose boot method of HTTP-FUSE KNOPPIX using block files cached in clients in order to decrease network traffic to a server when many clients were booted from network in classroom.

### 1. はじめに

近年、サーバ分野だけでなく、デスクトップ分野においてもオープンソースソフトウェア（以下、OSS）の利活用が期待されている。教育利用を目的として開発された KNOPPIX-Edu<sup>[1]</sup> を例に挙げると、多くは大学で利用されており、高等学校に関してはまだこれからという状況にある<sup>[2]</sup>。自治体業務における OSS の普及に向けた実証<sup>[3]</sup> では非 OSS 環境への慣れによる影響が報告されており、教育現場等での早い段階で OSS を利用する機会に触れることが重要である。

教育現場への OSS の普及促進については、独立行政法人 情報処理推進機構が平成 16 年度に小学校から大学まで全国で 17 校、約 3800 名を対象に、Linux パソコンを授業に取り入れた実証実験を実施している<sup>[4]</sup>。また、財団法人コンピュータ教育開発センターは、OSS ベースの IT 環境を学校教育に導入して学校教育現場への普及を促進させるために Open School Platform プロジェクト（以下、OSP プロジェクト）を実施している<sup>[5]</sup>。

本報告では、平成 18 年度の OSP プロジェクト（大分市・豊後大野市地域プロジェクト）における実施内容を紹介するとともに、実証で採用したネットワーク起動方式においてサーバへのトラフィックを軽減する手法について提案し、その有効性について述べる。

### 2. Open School Platform プロジェクト

#### 2.1 概要

財団法人コンピュータ教育開発センターは、平成 17 年度に全国 4 地域の小中学校 15 校を対象として、OSS デスクトップ端末 700 台規模、約 6700 名の児童生徒が参加した実証実験を行なった<sup>[5]</sup>。平成 18 年度の OSP プロジェクトでは、実証を通じて OSS 導入のためのパッケージモデルや、文教市場における OSS デスクトップに関するサポートとビジネスのあり方に関するモデルを提案して実験後の自立的な OSS 環境の運用を目標とした。全国の 6 つのプロジェクトにおいて小中高校 16 校が実証に参加し、成果は「Open School Platform パッケージ」として公開されている<sup>[6]</sup>。

## 2.2 大分市・豊後大野市地域プロジェクト

大分市・豊後大野市地域プロジェクトは、大分県内の高校 3 校を対象として「ソフトウェア技術」や「計測制御」、「情報 A」など 10 科目の授業において OSS デスクトップを利用した。ディストリビューションには、教育現場における利用実績が多い KNOPPIX を採用し、各校の対象授業に応じてメニューやアプリ等をカスタマイズした。KNOPPIX は CD 等からコンピュータを起動して手軽に Linux を利用できる点や、カスタマイズが容易というメリットがあり、OSP プロジェクトに採択された 6 地域のうち、大分を含む 3 地域が採用した。大分のプロジェクトでは 2 校に KNOPPIX を CD から起動する方式を導入し、ネットワーク起動方式を 1 校に導入した。実証システムの概要を Fig.1 に示す。

実証では 3 校で 120 台の生徒用端末を新規導入し、教員と生徒合わせて約 700 人規模、10 教科（約 600 時間）の授業において OSS デスクトップ環境を利用した学習環境の検証をおこなった。Table 1 に生徒用 OSS 端末とサーバの一覧を示す。各校に課題の提出等に利用するファイル共有サーバを導入し、三重総合高校にはネットワークから KNOPPIX を起動するためのサーバを導入した。

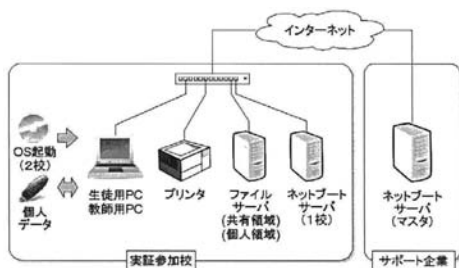


Fig.1 実証における導入システムの概要

Table 1 主要機器の一覧

学校名	生徒用 端末台数	サーバ 台数	起動 方法
情報科学 高等学校	40	2	CD 起動
鶴崎工業 高等学校	40	1	CD 起動
三重総合 高等学校	40	3	ネット 起動
合計	120	6	

## 2.3 実施内容の評価

授業における OSS デスクトップ環境や OSS アプリケーションの利用について、実証実験の前後で生徒と教員にアンケートやインタビューを行ない、システムの使いやすさや課題等について調査した。また、端末の利用状況をサーバ上に記録する仕組みを導入し、起動時間などを定量的に計測した。

アンケートでは約 7 割の生徒が「KNOPPIX を利用した授業は楽しかった」と回答するとともに、多くの教員が「実証実験は教育的に意味があるものである」と回答した。このことから、教員や生徒の双方から寄せられた利用上の課題や改善項目を解決することで、今後も学校教育現場における普及促進が期待できると考えられる。本稿では、課題のひとつとして挙げられたネットワーク起動における起動時間について、その改善策を検討する。

## 3. KNOPPIX のネットワーク起動

### 3.1 HTTP-FUSE KNOPPIX について

ネットワーク経由で KNOPPIX を起動する方式は複数提案されており、本実証では HTTP-FUSE KNOPPIX<sup>[7]</sup>を採用した。この方式では、クライアントは OS やアプリケーションなどの起動処理に応じて、サーバからファイルをダウンロードしながらデスクトップ環境を起動する。サーバにおいて HTTP サービスを起動し、通常の起動用 CD や DVD に保存される 700MB 近い OS イメージファイルを分割・圧縮して数百バイトから数百キロバイトのサイズにした「分割圧縮ブロックファイル」（以下、ブロックファイル）を HTTP サービスのコンテンツとして用意する。

教室などにおいて多数のクライアントを HTTP-FUSE KNOPPIX で一斉に起動する場合には、短時間のあいだにサーバにブロックファイルの要求が集中するため、サーバの NIC やハブがボトルネックとなってクライアントの起動時間に影響を与えてしまう。この課題を解決するには、サーバからダウンロードするブロックファイルのサイズを減少することが挙げられる。実証ではファイルに含まれるデータの内容を最適化してダウンロードするファイルサイズの総量を減少することで、クライアントの起動時間を短縮することに成功している。以降では、これとは異なる方策として、サーバへのトラフィックを軽減する起動手法を提案する。

### 3.2 キャッシュを利用した起動手法

クライアントは、サーバからダウンロードしたブロックファイルを RAM ディスクや USB メモリへキャッシュし、以降の処理で必要となった場合に利用する。そこで、あるクライアントにキャッシュされているブロックファイルを利用して、他のクライアントを起動する手法について検討する。

従来手法では、クライアントがサーバへブロックファイルを要求すると、サーバは各クライアントへファイルを転送する。これに対して提案手法では、サーバはブロックファイル、またはそのブロックファイルをキャッシュしているクライアントの情報をクライアントへ転送する。提案する手法でブロックファイルをダウンロードする処理の流れを Fig.2 に示す。サーバは、クライアントからの要求に対してブロックファイルをキャッシュしているクライアントの情報を転送する。これを受信したクライアントは、通知されたクライアントへ要求を送信し、キャッシュされているブロックファイルをダウンロードして起動処理を実行する。図中の矢印の大きさは転送されるデータ量をイメージしており、複数のクライアント間で並行してブロックファイルを転送することができるため、従来手法と比べてサーバへのトラフィックの集中を軽減することが可能となる。

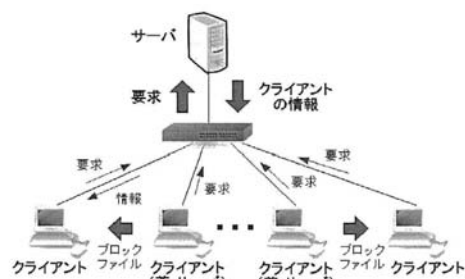


Fig.2 提案手法の概要

### 3.3 サーバの実装

サーバは、クライアントからブロックファイルの要求があった場合に、ブロックファイルを転送するか、ブロックファイルをキャッシュしているクライアントのアドレスを通知する。この判断処理には Apache の Rewrite 機能を利用し、一時的にファイルが存在しないことを示す戻り値の「302」とともに代替の

URL を「Location」情報でクライアントに通知する。Rewrite 機能による URL の書換えは、あらかじめ静的に設定することや、Perl などで動的に設定することができる。

そこで、データベースのアクセスと URL の書き換えを行なう処理を Perl で記述した。データベースには、キーと値を対応づけて管理する QDBM<sup>[8]</sup>を採用している。ブロックファイル名をキーとし、クライアントのアドレスを値として登録することで、あるブロックファイルをキャッシュしているクライアントのアドレスを検索できる。このようなデータベースは、他に gdbm や ndbm、Berkley DB などが提案されており、Sendmail や OpenLDAP などの OSS で利用されている。これらのデータベースの多くは 1 つのキーに対して値を 1 対 1 に対応させることができる。QDBM では複数の値を対応づけることができる。これにより、ある 1 つのブロックファイルをキャッシュしている複数のクライアントを登録できる。この登録数は、Rewrite 機能を実装する Perl スクリプト内で設定している。

### 3.4 クライアントの実装

クライアントは、FUSE ドライバを利用してサーバからブロックファイルをダウンロードする。FUSE ドライバは OS やアプリの起動処理の過程で cloop ファイル内で該当するブロックファイルが読み出される場合に、まずローカルなハードディスクやメモリ上にキャッシュされているか検索する。ローカルに見つかればそのファイルを利用し、見つからない場合にはサーバからブロックファイルをダウンロードする。この機能は libcurl で実装されていて、HTTP クライアント機能に関する処理には「CURLOPT\_FOLLOWLOCATION」オプションが設定されている。これにより、サーバから「Location」情報で通知された代替の URL へ要求を送ることができる。

一方、キャッシュしたブロックファイルを他のクライアントから利用できるようにするためにはクライアント上で HTTP サービスを起動し、そのコンテンツとしてキャッシュしたブロックファイルのディレクトリを指定する。この HTTP サービスでは 3.3 で説明したサーバと異なり、データベースを利用しない。また、HTTP サービスは minirt.gz から起動するため、ユーザの利用環境として起動する OS イメージを再構築する必要はない。

以降では、3.3 で説明したデータベースの操作を行なうサーバを「主サーバ」と呼ぶ。また、どのクライアントも HTTP サービスを起動するものの、主サーバのデータベースに登録されて他のクライアントから利用できるクライアントを「代替サーバ」と呼ぶ。

### 3.5 起動および終了処理の概要

クライアントは PXE ブートによって主サーバから `minirt.gz` をダウンロードし、ネットワーク設定などが完了した後、主サーバへブロックファイルのダウンロードを要求する。Fig.3 に、2 台のクライアント A、B を順次起動する場合について、ブロックファイルを取得する処理の流れを示す。

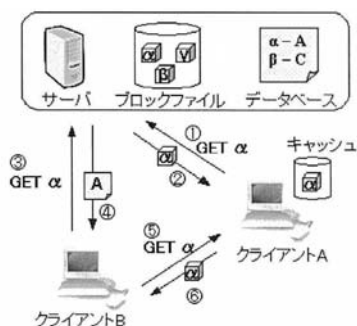


Fig.3 ブロックファイルのダウンロード

ブロックファイルのファイル名は通常 32 文字からなる英数字であるが、ここでは説明上、ファイル名を  $\alpha$  と略記する。

- ① クライアント A は、どのクライアントもブロックファイルをダウンロードしていない状況で、主サーバへブロックファイル  $\alpha$  を要求する。
- ② 主サーバは、クライアント A のアドレスとファイル名  $\alpha$  を対応づけてデータベースへ登録し、クライアント A にブロックファイルを送信する。クライアント A はダウンロードしたファイルをキャッシュし、代替サーバとして利用可能となる。
- ③ クライアント B が、主サーバへブロックファイル  $\alpha$  を要求する。
- ④ 主サーバはデータベースを検索し、ファイル  $\alpha$  は代替サーバ A によってダウンロードされていることがわかる。そこで、クライアント B には代替サーバ A のアドレスを通知し、ブロックファイルは送信しない。

⑤ クライアント B は、主サーバからの情報をもとに、代替サーバ A へブロックファイル  $\alpha$  を要求する。

⑥ 代替サーバ A は、ブロックファイル  $\alpha$  をクライアント B に送信する。

なお、ここでは 1 つのブロックファイルについて 1 台の代替サーバを登録する場合について説明したが、教室等で運用する場合にはクライアントの台数やネットワーク機器など利用環境に応じて代替サーバの設定数を検討する必要がある。

代替サーバとなっているクライアントを終了するときには、主サーバへ、データベースの登録内容を削除する要求を送信する。これは、データベース上に情報が残った状態で、あるクライアントがその代替サーバのアドレスを主サーバから受け取ってしまうと、そのクライアントはブロックファイルをダウンロードできず、最終的に起動に失敗することを防ぐためである。

## 4. 提案手法の評価

### 4.1 評価方法

主サーバにはデータベースへの登録や検索処理を組み込んでおり、これらの処理がクライアントからのアクセスに影響を与えるか、事前に検証が必要となる。導入したデータベースの処理性能や、クライアントから主サーバへのアクセス性能を計測した結果、提案手法による性能の低下は見られなかった<sup>9,10)</sup>。

ここでは、複数台のクライアントを起動した場合について、主サーバの CPU 負荷やトラフィックを計測し、提案手法の有効性を検証する。OSP プロジェクトに参加した三重総合高校の利用環境は、Fig.4 に示すように 40 台のクライアントが 100Mbps のハブに接続されている。そこで、代替サーバの設定数を 1, 2, 4, 8 台としてクライアントを起動した場合について、主サーバにおける CPU 負荷やトラフィックを計測した。また、比較のために、代替サーバを利用せずに、従来どおり全てのクライアントが主サーバからブロックファイルをダウンロードする場合についても計測をおこなった。主サーバの CPU 負荷を `sysstat` で計測するとともに、サーバからクライアントへ転送されるパケットを `tcpdump` コマンドでキャプチャして解析した。

クライアントを PXE ブートできるように BIOS 設定し、主サーバから TFTP で `minirt.gz`

をダウンロードする。ネットワークなどの最低限必要な機能を `minirt` から起動し、DHCP で IP アドレスを取得する。その後、HTTP を利用して主サーバからブロックファイルをダウンロードしながら起動処理を実行する。検証で利用した機器の仕様を Table 2 に示す。

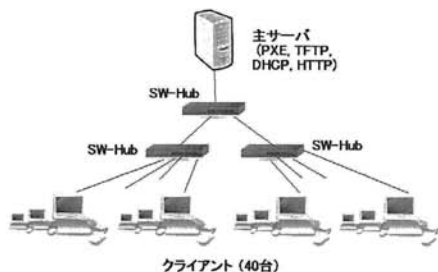


Fig.4 実験システムの構成

Table 2 検証における利用機器の仕様

サーバ	CPU : Opteron 3.0GHz RAM : 2GB OS : OpenSUSE 10.1 HTTPD : Apache 2.2.0
クライアント	CPU : Celeron M 1.5GHz RAM : 512MB OS : knoppix 4.0.2 HTTPD : Lighttpd 1.4.12

#### 4.2 クライアントの起動

クライアント端末の電源を投入後、KDE が起動完了するまでの時間を計測した結果を Fig.5 に示す。クライアントが 1 台では、従来手法における起動時間は 96 秒であり、提案手法は 99 秒で従来手法より長い数値となった。

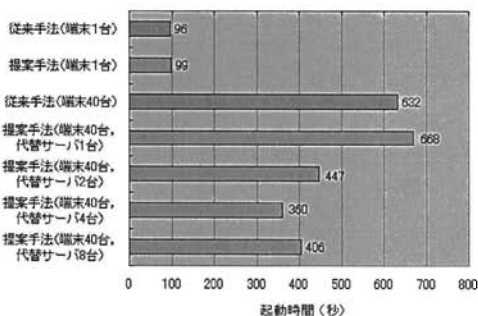


Fig.5 クライアントの起動時間

教室内の 40 台のクライアントを起動した場合、従来手法での全クライアントの起動完了までの時間は 632 秒であった。提案手法では代替サーバの台数によって起動時間が異なり、代替サーバが 1 台では従来手法より長くなった。これは、ハードやソフトの差異に加えて、代替サーバは主サーバからブロックファイルをダウンロードしながら、他のクライアントからの要求にも対応するため、NIC や CPU の負荷が高くなったと考えられる。代替サーバが 2 台以上では、従来手法よりも起動時間は短い。代替サーバの台数が増えることで、教室内の HTTP サービスの総数が増えるため、ブロックファイルのダウンロードがクライアント間で並行して実行される。このトラフィックの分散により、起動時間が短縮できることを確認できた。しかし、代替サーバが 8 台では 4 台の場合より起動時間が長くなっており、この理由については以降で考察する。

#### 4.3 主サーバの CPU 負荷

従来手法で 40 台のクライアントを起動した場合における主サーバの CPU 負荷は、Fig.6 に示すように起動処理全体にわたって 10%程度であった。一方、提案手法における主サーバの計測結果を Fig.7 に示す。代替サーバが増えるにつれて主サーバの CPU 負荷は高くなり、代替サーバが 8 台のときの最大値は 40%であった。主サーバ上のデータベースには代替サーバ 1 台あたり約 2000 個のブロックファイル名が登録され、登録処理に要する時間は検索処理よりも長く<sup>[10]</sup>、代替サーバが増えると短時間のあいだにデータベースへの登録処理が集中して CPU 負荷が増大すると考えられる。この対策として、主サーバでランダムに遅延処理を追加して、クライアントの起動タイミングをずらすことが考えられる。

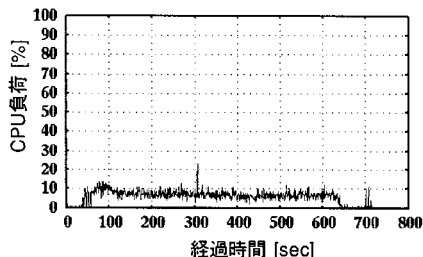
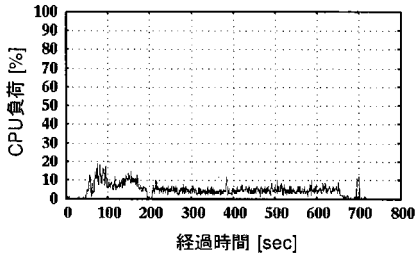
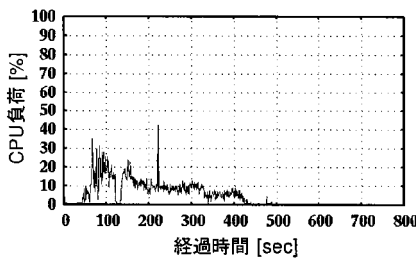


Fig.6 従来手法で 40 台のクライアントを起動した場合における主サーバの CPU 負荷

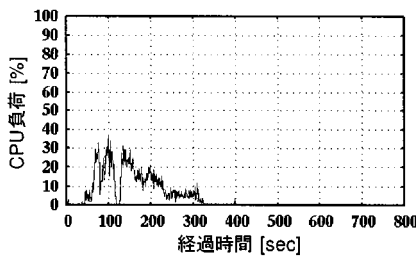
データベースに関する現在の実装では、クライアントからの要求があるたびに、QDBM のデータベースファイルのオープンとクローズを行なっている。この点については、今後改善が必要と考えている。



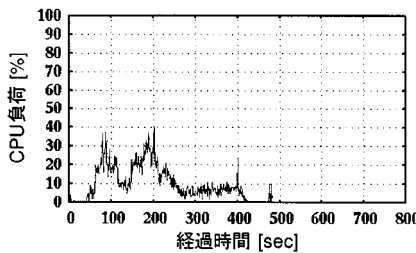
(a) 代替サーバが 1 台の場合



(b) 代替サーバが 2 台の場合



(c) 代替サーバが 4 台の場合



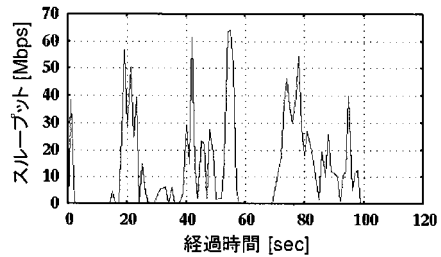
(d) 代替サーバが 8 台の場合

Fig.7 提案手法における主サーバの CPU 負荷

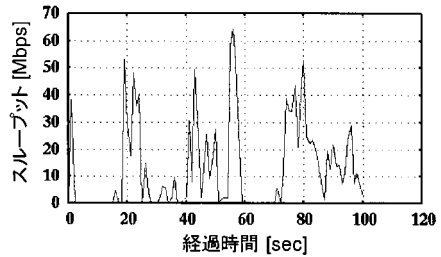
#### 4.4 データ転送のスループット

クライアントを 1 台だけ起動した場合について、主サーバからクライアントへ送信されたデータのスループットを計測した結果を Fig.8 に示す。スループットの最大値は従来手法、提案手法のいずれも 64Mbps であった。提案手法では、クライアントからの要求に対して、サーバでデータベースへの登録などの処理を行なうものの、従来手法と同等の転送性能であることを確認した。

Fig.9 は、40 台のクライアントを従来手法で起動した場合におけるスループットの計測結果を表している。



(a) 従来手法



(b) 提案手法

Fig.8 クライアントを 1 台起動した場合におけるスループットの計測結果

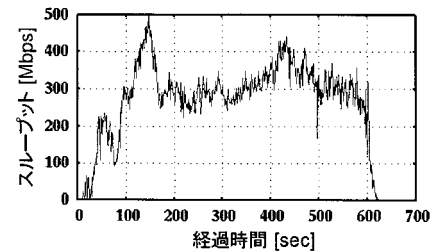
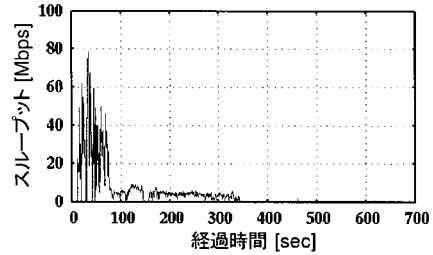


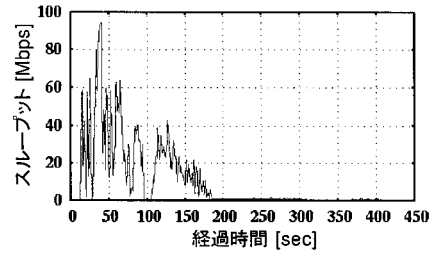
Fig.9 従来手法で 40 台のクライアントを起動した場合におけるスループットの計測結果

実験で利用したハブの転送能力は 100Mbps であるが、図から、データを転送している際に大部分にわたってスループットが 100Mbps を越えていることがわかる。この最大値は 499Mbps であり、ハブの転送能力を考慮すると、本来はそのようなデータ転送は不可能である。スループットの計測は、主サーバ上のターミナルで `tcpdump` コマンドを用いてパケットをキャプチャしており、主サーバが送信したパケットをクライアントが受信できなかった場合の再送パケットも含まれる。このことから、クライアントからの要求によって主サーバへのトラフィックが集中し、主サーバが送信したパケットに対するクライアントからの ACK を主サーバが受信できなくなってしまうためにパケットを再送していると考えられる。主サーバの NIC が 1000Mbps であることも、ハブの転送性能を上回るほどのパケットを転送してしまった可能性がある。この点については、今後、ハブのポートのミラーリングや、サーバ以外のコンピュータでパケットをキャプチャするなど、さらに詳細な調査が必要である。

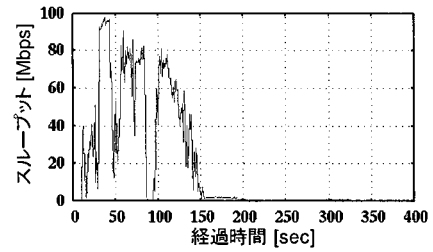
一方、提案手法で 40 台のクライアントを起動したときのスループットの計測結果を Fig.10 に示す。主サーバからブロックファイルをダウンロードするのは、代替サーバとなる数台のクライアントだけであるため、代替サーバが増加するにつれてスループットが高くなっている。代替サーバが 4 台のときには経過時間が 40ms のあたりでハブの転送性能である 100Mbps に近くなり、飽和気味となっている。さらに、代替サーバを 8 台とした場合にはスループットが 100Mbps を超える区間が見られるようになり、最大値は 256Mbps であった。これは Fig.9 と同様に、トラフィックの集中によって主サーバが再送を行なっているものと考えられる。4.2 の起動時間に関する考察において、代替サーバが 2 台、そして 4 台の場合には台数が増えるにつれて起動時間が短くなったものの、8 台では長くなった理由はこのような再送が一因と考えられる。提案手法では代替サーバによってブロックファイルの転送は分散されるものの、クライアントはいったん主サーバへブロックファイルを要求する。Fig.10 の結果から、ブロックファイルの転送以降はスループットが低く、代替サーバの情報を通知することによるトラフィックが少ないことを確認できた。



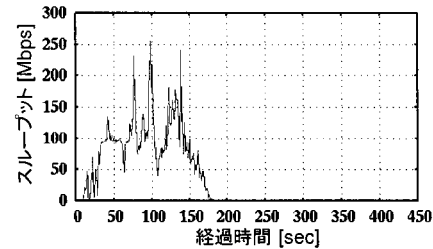
(a) 代替サーバが 1 台の場合



(b) 代替サーバが 2 台の場合



(c) 代替サーバが 4 台の場合



(d) 代替サーバが 8 台の場合

Fig.10 提案手法でのスループットの計測結果

以上の結果から、提案手法によってサーバへのトラフィックの集中が軽減できることを確認できた。計測では 100M のハブを利用したが、ギガビット対応ハブを利用することで再送に関する課題を解決できると考えられる。

## 5. 今後の課題

今後は以下のような点について検討が必要と考えている。

### 5.1 起動時間の短縮

サーバへのトラフィックの集中を軽減するには、ファイルの最適化によってダウンロードするブロックファイルのサイズを小さくすることや、本提案のようにローカルにキャッシュされたブロックファイルを利用する方法などが挙げられる。OSP プロジェクトにおけるネットワーク起動では、前者の手法によって起動時間を短縮できており、今後は本提案と組み合わせで効果を検証したい。

さらに、利用環境に応じた代替サーバの設定台数や、主サーバにおける HTTP サービスのパラメータ、代替サーバを登録するデータベースなどについて検討が必要である。

### 5.2 WAN 環境への拡張

本提案は学校におけるコンピュータ教室などを想定しているが、WAN 環境への適用も可能である。そのためには、ネットワークが一時的に利用できない場合でも USB メモリなどの記憶媒体にキャッシュされたブロックファイルを利用することや、ブロックファイルを検索するために Bamboo DHT<sup>[11]</sup> のような DHT (Distributed Hash Table) などの仕組みを組み込むことが考えられる。ブロックファイルのダウンロードについては、モバイル環境に対応可能な小型の HTTP-FUSE KNOPPIX-BOX<sup>[12]</sup> の利用やセキュリティ<sup>[13]</sup> についても考慮する必要がある。

## 6. まとめ

学校教育現場における OSS の活用促進を図るため、OSP プロジェクトにて県内の高校を対象に OSS デスクトップ環境の利用に関する実証を行なった。

また、OSS デスクトップ環境をネットワークからデータをダウンロードしながら起動する方式について、サーバへトラフィックが集中する課題を解決するために、クライアントにキャッシュされたファイルを利用する手法を提案し、その有効性を確認した。

## 参考文献

- [1] KNOPPIX-Edu, <http://www.knoppix-edu.org/>.
- [2] 千葉大作, 中川俊哉, 下村昭洋, 江原裕

幸, 後藤和弘, 志子田有光, 早川栄一, “高等学校における OSS 活用実践とコミュニティ連環モデルの提案 -Open School Platform Project 大分プロジェクトの取り組み-”, 信学技報, Vol.106, No.507, pp.31-36, 2007.

- [3] 2005 年度「自治体におけるオープンソースソフトウェア活用に向けての導入実証」成果, <http://www.ipa.go.jp/software/open/2005/stc/report/index.html>
- [4] 2004 年度「学校教育現場におけるオープンソースソフトウェア活用に向けての実証実験」成果, <http://www.ipa.go.jp/software/open/2004/stc/eduseika.html>
- [5] Open School Platform プロジェクト, <http://www.cec.or.jp/e2e/osp/index.html>
- [6] OSP ポータルサイト, <http://e2e.cec.or.jp/osp/>
- [7] HTTP-FUSE KNOPPIX, <http://unit.aist.go.jp/itri/knoppix/http-fuse/>
- [8] QDBM, <http://qdbm.sourceforge.net/>
- [9] 後藤和弘, 北川健司, 江原裕幸, 中川俊哉, 下村昭洋, “キャッシュされた分割圧縮ブロックの利用による HTTP-FUSE KNOPPIX における起動の高速化”, 情報処理学会第 69 回全国大会, 3E-5, 2007.
- [10] 後藤和弘, 北川健司, 江原裕幸, 中川俊哉, 下村昭洋, “Open School Platform プロジェクト ~大分市・豊後大野市地域プロジェクト-HTTP-FUSE KNOPPIX におけるキャッシュを利用した起動に関する検討-”, 2006 年度大分県産業科学技術センター研究報告, 2007.
- [11] The Bamboo Distributed Hash Table, <http://www.bamboo-dht.org/>
- [12] 金井遵, 須崎有康, 八木豊志樹, 並木美太郎, “HTTP-FUSE-KNOPPIX-BOX のキャッシュ導入による高速化と評価”, 情報処理学会「システムソフトウェアとオペレーティングシステム研究会」報告, Vol.2006-OS-102, pp.45-52, 2006.
- [13] 中村めぐみ, 宗藤誠治, 須崎有康, 飯島賢吾, 八木豊志樹, 大澤一郎, “トラステッド・コンピューティングによる HTTP-FUSE KNOPPIX クライアントのセキュリティ強化”, 信学技報, vol.106, p.223-230, 2006.