

【論文】

HTTP-FUSE-KNOPPIX を用いたシンクライアントシステムの 構築と検証

許斐行和* 松本勝哉**

HTTP-FUSE-KNOPPIX applied to thin client system

— integration and verification —

Yukikazu KONOMI* Katsuya MATSUMOTO**

Abstract: The system management of PC is going to complicate more and more in order to increase a lot of function. Recently it focuses on the thin client system in this situation. It's available to access easily this unitary managed server from client PC. But there is a small advantage especially to a few clients because of the initial cost. We proposed the thin client system used free software HTTP-FUSE-KNOPPIX to reduce the initial cost. This paper presents the summary of the integration of the thin client system and the verification of them.

Keywords : thin client, HTTP-FUSE-KNOPPIX

1. はじめに

パーソナルコンピュータ（以下PC）が普及したことにより、PCの管理や運用に手間がかかるようになってきた。管理で手間がかかるのがアプリケーション（以下アプリ）の導入と障害発生時のリカバリである。そこでOSやアプリはサーバで管理し、クライアントはサーバにアクセスして利用するシンクライアントが普及してきた。アプリなどをサーバで集中管理することでバージョンアップや機能の追加と削除が容易になった。しかし、現在のシンクライアントはOSのライセンス料や専用の端末などに費用がかかるため、研究室など10台程度の小規模に導入する場合は管理／運用の簡易化によるメリットより、導入費用の方が大きくなってしまふ。

本論文ではこのような現状を考慮し、無償で利用できるフリーソフトウェアとPCを用いてシンクライアントシステムを構築することで導入費用を削減し、その実用性を検証する。

2. 概要^[1]

本章では、実際に構築したシンクライアントシステムの概要を述べる。

2.1 システム構築の方針

本研究ではOS本体をサーバからダウンロードしながら起動するLinuxOSであるHTTP-FUSE-KNOPPIX^[2]を用いて、シンクライアントシステムを構築した。HTTP-FUSE-KNOPPIXについては3章で詳しく述べる。システムを構築するにあたり以下に示す方針を立てた。

*工学研究科電気工学専攻1年

**電気情報工学科

- ・フリーソフトウェアを用いて構築する。
- ・ユーザが変更を加えた部分を差分ファイル形式で保存する。

フリーソフトウェアとは無償で利用できるソフトウェアのことである。上記の方針に沿ってシステムを構築することで導入コストを減らし、障害発生時の復旧が容易になる。

2.2 システム概要

ここでは前述の方針に沿って構築したシステムの概要を説明する。図2-1 は本研究のために構築した研究室のネットワーク図である。

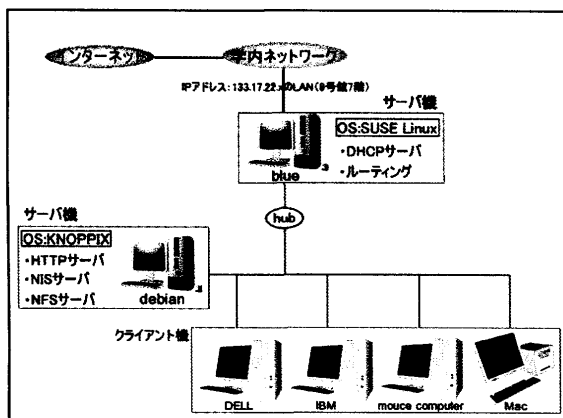


図2-1 研究室のネットワーク図

松本研究室のある8号館7階にはIPアドレス133.17.22.x (xは任意の数字)のLANが通っている。その下部に研究室のサーバ機blue (以下、blueと記す)が、さらにその下部にハブ (hub) を挟んで、サーバ機debian (以下、debianと記す)とDELLなどの各クライアント機が接続されている。

ここで、blueはDHCPサーバとして動作しており、同時にルータの役割も担っている。つまり研究室内のdebianと各クライアント機は、blueを介して学内ネットワーク、インターネットに接続されていることになる。

本研究で重要な役割を担っているのはdebianである。debianはHTTP, NIS, NFSサーバとして動作してい

る。HTTPサーバはOSとアプリケーションの配信を行い、NISサーバはユーザ認証に使用している。NFSサーバは各ユーザのホームディレクトリと、差分ファイルを保存するために使用している。

次に本研究で構築したシステムの起動の流れは以下のようにになっている。

- 1、クライアントPCにブート用のCDを挿入
- 2、電源をつけてOS本体をサーバへ要求
- 3、OSをダウンロードしながら起動
- 4、ログイン画面でユーザ名とパスワードを入力
- 5、ホームディレクトリ, 差分ファイルをマウント
- 6、起動完了

3. HTTP-FUSE-KNOPPIX

ここでは今回使用したHTTP-FUSE-KNOPPIXについて述べる。

HTTP-FUSE-KNOPPIXはLinuxの1CDディストリビューションの1つであるKNOPPIXを元にしたLinuxOSで、ルートファイルシステムをローカルに持たず、HTTPサーバからダウンロードしながら起動するという特徴がある。HTTP-FUSE-KNOPPIXの概要を図3-1に示す。

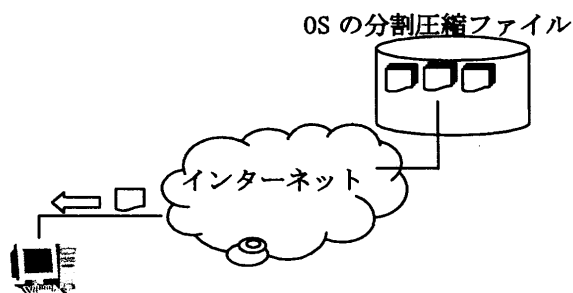


図3-1 HTTP-FUSE-KNOPPIXの概略図

CDブートのKNOPPIXでは事前に700MBにもなるISOイメージをダウンロードしておかなければならず、またDVD版ではその容量は3.8GBにもなりダウンロードすることも容易ではない。その点、HTTP-FUSE-

KNOPPIXの起動に必要なものはカーネルとブートローダのみであり、これは約6MBに収まる。これをCDもしくはUSBメモリに保存してそこから起動し、ルートファイルシステムはHTTPサーバからダウンロードする。サーバからダウンロードするルートファイルシステムはブロック毎に分割し圧縮した分割圧縮ブロックファイルで構成されており、一度に全てのルートファイルシステムをダウンロードするのではなく、必要なものを必要なときにダウンロードして利用する。現在世界中にHTTP-FUSE-KNOPPIXの配信サーバがありDVD版とCD版を起動することができる^[3]。

4. システムの構築と設定

本章では、HTTP-FUSE-KNOPPIXを用いたシンクライアントの構築に必要なサーバ環境の詳細や設定方法を示す。

4.1 システムの動作と機能

今回構築したシステムの動作と機能について述べる。このシステムはHTTP-FUSE-KNOPPIXを用いて、どのクライアント機からもユーザが設定した環境のLinuxの使用を可能にすることを目的としたものである。図4-1は実際に構築したサーバとクライアントの関係を図示したものである。

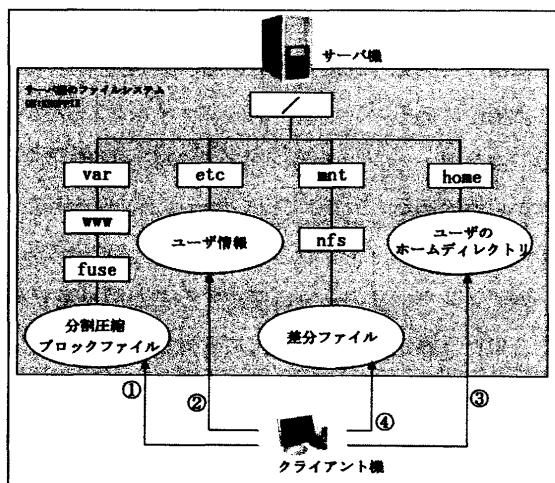


図4-1 本システムの概略図

クライアントPCがKNOPPIXを起動するには、図4.1に示してある①～④の動作を順番に行うことが必要である。

①はHTTPで公開している分割圧縮ブロックファイルをクライアント側が起動時にダウンロードすることを示している。HTTP-FUSE-KNOPPIXは、この分割圧縮ブロックファイルを必要なときに必要な分だけダウンロードすることで、ネットワークにかかる負荷を軽減することができるのである。起動に必要なファイルがダウンロードされるとログイン画面が表示されユーザ名とパスワードを要求し、ここから②の動作に移る。

②ではNISを使用してログイン画面で入力されたユーザ名、パスワードからユーザの認証を行っている。認証が失敗すると再度ユーザ情報の入力を要求し、認証に成功した場合は③の動作に移る。

③ではNFSで公開されたディレクトリをユーザのホームディレクトリとしてマウントしている。サーバ側にホームディレクトリを置いておくことで作業途中のデータや設定ファイルを保存できるので、どのパソコンから起動しても同じ環境でKNOPPIXを使用することが可能である。最後に④の動作に移る。

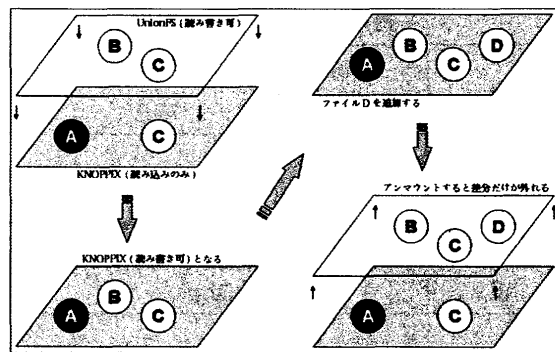


図4-2 UnionFSの説明図

④ではNFSサーバに保存された差分ファイルをUnionFS^[2]でマウントしている。UnionFSとは複数のディレクトリを1つのディレクトリとして扱うことのできるファイルシステムである。図4-2にUnionFSの説明図を示す。本システムでは分割圧縮したルートファイルシステムをサーバ上からダウンロードす

るため、NFSを使用しているホームディレクトリ以外は書き込み不可である。図4-2の左半分にあるようにKNOPPIXにUnionFSでディレクトリをマウントすることで本来書き込み不可の領域を仮想的に書き込み可能にしている。これよりユーザは自由にアプリケーションのインストールを行うことができる。図4-2の右半分にあるようにUnionFSでマウントしたディレクトリをアンマウントすることで追加されたアプリケーションや変更されたファイルは差分ファイルとしてサーバ側に残される。この差分ファイルNFSに保存しておき、次回起動時に再びマウントすることで前回追加したアプリケーションを継続して使用することが可能になった。

4.2 サーバの構築と設定

上で述べたように、どのクライアント機からも同じ環境でKNOPPIXを使用できるようにするにはHTTPサーバ、NISサーバ、NFSサーバを必要とする。ここでは、システムを動かすために必要なサーバの設定について述べる。今回使用したサーバ機のハードウェア環境を表4-1に示す。

表4-1 サーバの仕様

| | |
|--------|--|
| CPU | AMD Athlon™ 64 Processor 2800+ 1.7GHz |
| メインメモリ | 1024MB |
| OS | KNOPPIX 4.0.2 日本語CD-ROM版 |
| NIC | 100Mbps |

4.2.1 HTTPサーバの設定

分割圧縮ブロックファイルを配信するためにApacheを用いてHTTPサーバを構築する。Apacheとは、HTTPサーバ上で稼働し、Webサービスを提供する安定性のあるソフトウェアである。ここで設定すべきことは2つある。ひとつは分割圧縮ブロックファイルが存在するディレクトリのみを公開するよう設定することである。もうひとつは外部からのアクセス

を拒否し、研究室内のパソコンにのみ配信するようアクセス制限をすることである。どちらもセキュリティ対策であり、外部からの攻撃を受けにくくするためのものである。ここで行った設定のやり方と実際に設定を行ったファイルの内容は論文^[1]に記す。

4.2.2 NISサーバの設定

ユーザの一元管理を可能にし、KNOPPIXのログイン時にユーザ認証を行えるようNISサーバを構築する。NIS (Network Information Service) はネットワーク上からUNIXの管理や設定を可能にするソフトウェアであり、Sun Microsystems社によって開発されたものである。またネットワーク上の複数のコンピュータ間でユーザ情報を共有できるため、ユーザアカウントをNISサーバ上に作成しておけばネットワーク上のどのパソコンにもログイン可能である。ここで行う設定はNISを利用できる範囲をローカルネットワーク内に限定することである。これもセキュリティ強化のためである。ここで行った設定のやり方と実際に設定を行ったファイルの内容は論文^[1]に記す。

4.2.3 NFSサーバの設定

ユーザのホームディレクトリとして利用するため、また差分ファイルを保存するためにNFSサーバを構築する。NFS (Network File System) はUNIXシステムで利用されるファイル共有システムであり、Sun Microsystems社によって開発されたものである。NFSを利用すると離れた場所にあるコンピュータのファイルを、あたかも自分のコンピュータにあるファイルのように操作することが可能である。これによってサーバ上に作成したクライアント用のディレクトリを自分のパソコンにあるホームディレクトリのように使用することが可能である。ここで行う設定は2つあり、ひとつはサーバ上のどのディレクトリを公開し、それをどのように共有するのかを設定する必要がある。共有を「読み出し専用」や「読み書き可能」など選択でき、今回はファイルの保存を行うので読み書き両方のアクセスが可能となるよう設定を行う。もうひとつの設定はアクセスを許す範囲をローカ

ルネットワーク内に限定することである。これもセキュリティ強化のためである。

4.2.4 UnionFSの設定

書き込み不可のKNOPPIXを書き込み可能にし、ユーザが新たにアプリケーションを追加できるようUnionFSを起動時にマウントする設定を行う。

もともとCD-ROMから起動するKNOPPIXにはデフォルトでramdiskという読み書き可能なディレクトリがマウントされており、これによって読み書きの出来ないCD-ROMでも仮想的に読み書きを可能にしていた。はじめは図4-3の左のようにramdiskの上にNFS領域をUnionFSでマウントしてNFS領域に差分を残そうと考えていたが、アプリケーションのインストール情報がramdiskに書き込まれNFS領域に情報が残らなかった。そのため図4-3の右のようにramdiskをアンマウントした後にNFS領域をマウントすることで、NFS領域内にアプリケーションのインストール情報を保存することに成功した。

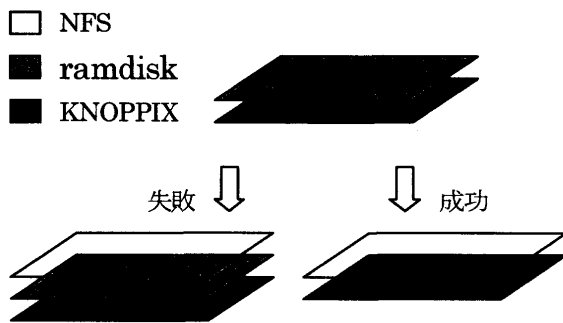


図4-3 UnionFSのマウント方法

しかし、起動スクリプトで起動時にこれらの作業を行おうとした際、ramdiskをアンマウント出来ないという問題が発生した。その原因はログインマネージャにKDM (KDE Display Manager) を使用していたことにあった。KDMを用いた場合ramdiskに常駐するアプリケーションが存在してしまいramdiskをアンマウントしようとするとbusy状態だとエラーが出てしまう。そこでKDMを使用しないテキストログインモードに切り替えて、先に

UnionFSのマウント作業を行った後でKDEが立ち上がるように設定を行った。これによりNFSにある差分ファイルをマウントしてKNOPPIXを起動させることに成功した。

4.3 KNOPPIXのカスタマイズ

KNOPPIXは本来ユーザ認証無しで利用できるため、本システムで使用するにはカスタマイズをしなければならない。カスタマイズした箇所は次の4つである。

- テキストログインモードでの起動
- NISサーバへの接続
- NFSのマウント
- UnionFSの設定

これらの設定はinittab、rcスクリプト、bash_profileでそれぞれ行っている。

4.3.1 inittabの役割

システムの起動時の挙動を指定するのがinittabの役割である。初期の設定では本研究のシステムが正常に動作しないため数箇所書き換える必要がある。初期の設定はCD-ROMから起動することを想定しているため、ログイン画面が表示されることなくknoppixという名前のユーザで自動的にログインしてしまう。このシステムは複数のユーザで使用することを想定しているため、自動的にログインしないよう設定を書き換える必要があり、テキストログインモードで起動しログイン画面を表示するよう設定した。テキストログインモードにした理由は4.2.4で述べたように、KDMが起動してUnionFSがアンマウントできなくなることを防ぐためである。

4.3.2 rcスクリプトの役割

rcスクリプトは、inittabで指定されたランレベルに該当するrc*.dディレクトリ内のスクリプトを実行し、サービスの起動と停止を行なうものである。ランレベルに該当するrc*.dディレクトリ内にNISサーバへ接続するためのスクリプトとNFSをマウントす

るためのスクリプトのシンボリックリンクをそれぞれ作成する。これによりクライアント機はユーザのログイン前にNISサーバへと接続し、ログイン時のユーザ認証を可能にするのである。またマウントされたNFSはユーザのホームディレクトリとなる。

4.3.3 bash_profileの役割

bash_profileはログイン時だけに実行されるスクリプトであり、ユーザのホームディレクトリ内に存在する。差分ディレクトリをUnionFSでマウントするにはユーザがログインした後に行う必要があるため、bash_profileにマウントを行うコマンドを記述する。ログイン後にしかマウントできないのは、ログイン前ではマウントすべき差分ディレクトリを判断できないためである。UnionFSのマウントが成功した後にKDEを立ち上げるため、bash_profileの最後の行にKDEを立ち上げるコマンドを記述する。これで各ユーザ独自の環境であるKNOPPIXが立ち上がることになる。

5. 運用・検証

本章では、実際に構築したシンクライアントシステムを実際に動作させ、実用性や問題点を検証する。

5.1 HTTP-FUSE-KNOPPIXの起動

まず、HTTP-FUSE-KNOPPIXの起動時間を測定した。測定に使用したクライアントPCの使用を表5-1に示す。

表5-1 クライアントの仕様

| | DELL | IBM |
|--------|-----------------|----------------|
| CPU | Celeron®2.53GHz | Pentium®4 2GHz |
| メインメモリ | 512MB | 256MB |
| NIC | 100Mbps | 100Mbps |

HTTP-FUSE-KNOPPIXの起動時間を測定するにあたり、比較のためにCDブートとHDDブートの起動時間も測定した。測定結果を表5-2に示す。

結果から分かるようにはやりHTTP-FUSE-KNOPPIXはネットワーク遅延の影響を受けるため一番遅かった。

表5-2 HTTP-FUSE-KNOPPIXの起動時間

| 起動方法 | 起動時間(s) |
|-------------------|---------|
| HDDブート | 130 |
| CDブート | 125 |
| HTTP-FUSE-KNOPPIX | 137 |

次に、クライアント4台で同時にHTTP-FUSE-KNOPPIXを起動した時の起動時間を測定した。結果を表5-3に示す。

表5-3 4台同時の起動時間

| 使用したPC | 起動時間 (s) |
|--------|----------|
| DELL | 142 |
| IBM | 149 |

この結果より1台だけ起動したときより4台同時に起動した時の方が時間がかかっていることがわかる。これは4台が同時にサーバからブロックファイルをダウンロードしようとするため、ネットワークに負荷がかかったからであると考えられる。

以上の結果より本システムは起動時にHDDブートと比べて10秒程度時間が多くかかるが、十分実用的であると考えられる。また、今回の測定では起動時のサービスの選別などを行っておらず、不要なサービスを停止させる等の調整を行えば起動時間を短くすることは可能である。

5.2 HTTP-FUSE-KNOPPIXの動作

HTTP-FUSE-KNOPPIXの起動に関しては問題がないことがわかったところで、今度はOSのアプリケーションの通常利用に問題がないか検証する。ここでは頻繁に用いられると思われるアプリケーションについて起動するまでの時間を測定し、実際に動作確認をすることで検証を行った。クライアントには

DELLを使用した。検証したソフトウェアは次のとおりである。

- Firefox (Web ブラウザ)
- OpenOffice (オフィスソフト)

Firefox と OpenOffice の起動時間をそれぞれの起動方法で測定した。測定結果を表5-4と表5-5に示す。

表5-4 firefox の起動時間

| 起動方法 | 起動時間 (s) |
|-------------------|----------|
| HDD ブート | 4 |
| CD ブート | 6 |
| HTTP-FUSE-KNOPPIX | 7 |

表5-5 OpenOffice の起動時間

| 起動方法 | 起動時間 (s) |
|-------------------|----------|
| HDD ブート | 7 |
| CD ブート | 46 |
| HTTP-FUSE-KNOPPIX | 18 |

アプリケーションの場合も HDD ブートに比べて HTTP-FUSE-KNOPPIX の方が2倍程度起動に時間がかかっている。しかし、一度起動してしまえば HDD ブートでも HTTP-FUSE-KNOPPIX でも、プログラムはメモリ上に展開されているのでその後の動作に関しては違いは感じられなかった。以上の結果より本システムでのアプリケーションの起動とその動作に関しても十分実用的であると考えられる。

5.3 ブロックサイズの最適値

HTTP-FUSE-KNOPPIX はブロック毎に分割してファイルに保存しているが、ブロックサイズは256KBに設定してある。これは、産業技術総合研究所^[4]の調べで、インターネット経由で起動させた場合にブロックサイズが256KBの 때가一番早く起動したか

らである。256KB以下ではダウンロードするファイル数が多くなり過ぎ、256KB以上だと1ファイルをダウンロードするのに時間がかかりすぎるため起動に時間がかかってしまう。

本研究ではサーバとクライアントは100MbpsのLANで結ばれているので256KBよりもおおきなブロックサイズでも転送に時間がかからず起動時間を短縮できる可能性がある。そこで、ブロックサイズを512MBにした場合と、1MBにした場合で起動時間がどう変化するかを調べた。その結果を表5-6に示す。

表5-6

| | 起動時間(s) |
|-------|---------|
| 512KB | 162 |
| 1MB | 171 |

この結果よりブロックサイズが大きくなるにつれ、起動時間が長くなっていることがわかる。LAN 経由で接続した場合でもブロックサイズ256KBが適当であることが確認できた。

5.4 障害発生時のリカバリ

本システムの特徴はユーザが OS 設定等を変更できるところにあるが、この場合ユーザが設定を間違えてしまい起動しなくなるといった事態が考えられる。そのような事態が発生したときに素早く復旧することが求められる。本システムではダウンロードした OS からの差分を NFS サーバに保存しそれを UnionFS を用いてマウントとすることでユーザが設定等を変更できるようにしている。よって NFS サーバにある差分ファイルを初期化することでデフォルトの設定に戻すことが可能である。HDD ブートの場合 OS の再インストールには30分から1時間程度かかるが、差分ファイルの消去は数分で完了する。

6 結論

本研究では、フリーソフトウェアを用いてシンクライアントシステムを構築し導入費用を削減するという目標をたて、それを実現した。ユーザの変更分を差分ファイル形式で保存することで、ユーザが自由にアプリケーションをインストールでき、障害が起きたときも素早くデフォルトの状態に戻すことができるようになった。PC を買い替えた際も簡単に環境を引き継ぐことが可能となった。

クライアントで障害が発生した場合に初期状態へは簡単に戻すことができるが、これでは障害発生のたびにユーザは最初からカスタマイズをやり直さなければならぬ。サーバ側で差分を定期的にバックアップしておき、障害が発生した場合にはバックアップから障害発生前の状態へ戻すことができるように改善する必要がある。

7 参考文献

- [1] 許斐行和, 渡辺行朗, 中村圭吾 : 「HTTP-FUSE-KNOPPIX を応用したシンクライアントシステムの構築」, 九州産業大学工学部平成18年度卒業論文(2007-3)
- [2] 「A Stackable Unification File System」, <http://www.fsl.cs.sunysb.edu/project-unionfs.html> (2007-1)
- [3] 須崎有康, 八木豊志樹, 飯島賢吾, 丹英之, " HTTP-FUSE-KNOPPIX" , Linux Conference2005 : <http://lc.linux.or.jp/paper/lc2005/CP-02.pdf> (2007-2)
- [4] 産業技術総合研究: 「HTTP-FUSE-KNOPPIX」, <http://unit.aist.go.jp/itri/knoppix/http-fuse/index.html> (2007-1)