2004－DSM－32　(1)
2004／3／29

KNOPPIXとSHFSを用いたノマディックデスクトップ環境の構築

丹 英之† 千葉 大作† 上原 光晶† 須崎 有康‡ 飯島 賢吾‡

†株式会社 アルファシステムズ ‡独立行政法人 産業技術総合研究所

CDから起動するDebian GNU/Linuxディストリビューションである KNOPPIXと，Secure Shellを用いたネットワーク透過型のファイルシステムである SHFS を組合せた．これにより，ユーザの使用する PC に依存することなく，どのマシンでも自分専用のデスクトップ環境を利用することができる仕組みを実装した．本稿では，この実装についての詳細と実際の使用を想定した性能評価の結果について述べる．

Construction of nomadic desktop environment using KNOPPIX and SHFS

Hideyuki Tan†, Daisaku Chiba†, Mitsuaki Uehara†, Kuniyasu Suzaki‡, Kengo Iijima‡

† Alpha Systems, Inc.
‡ National Institute of Advanced Industrial Science and Technology

We developed a nomadic desktop environment which users KNOPPIX and SHFS. KNOPPIX is a CD bootable Debian GNU/Linux. SHFS is a network file system that uses Secure Shell. The desktop environment makes possible to use without dependency of hardware. In this paper, we present detail about this implementation and performance.

1. はじめに

人がコンピュータを利用する目的の一つに，知的生産を支援する文房具としての用途がある．近年みられる，計算機の性能向上，ユーザインタフェースの進化，そして，インターネットの普及により，もはや文房具としてのコンピュータは人々の生活に無くてはならないものとなりつつある．この文房具的利用がうまく機能しているのは“画面の中に自分の机がある”というデスクトップのメタファーが適切であったからと考えられる．

最近ではノートパソコンなど個人用のデバイスを持ち歩くことにより，移動先でも常に自分専用のデスクトップ環境を利用する形態も頻繁に見られるようになった．また一方では，ネットワークのブロードバンド化も進んだことで，インターネットに接続されたPCが当り前のように，存在している．

そこで，この状況を“ネットワークで接続された計算機が遍在するものである”と仮定すると，自分専用のデスクトップ環境もネットワークを介してユーザの移動について行くことが可能であると考えられる．ここで，デスクトップ環境の再現に必要となるリソースはできるだけ移動先での現地調達で賄うという方針の元，計算機を構成する二大要素であるハードウェアとソフトウェアで切り離し，ハードウェアを持ち歩くことなくソフトウェアの入ったメディアのみを持ち歩くことで実現する放浪・遊牧型のデスクトップコンピューティングを，新しいノマディックデスクトップと定義する．

この遊牧的デスクトップ環境を実現するため，CD-ROMから起動するDebian GNU/Linuxである KNOPPIX[1]と Secure Shell(ssh)を利用したネットワーク透過型のファイルシステムである SHFS[2]を組合せ，ユーザの使用する PC に依存することなく，どのマシンを使用しても自分専用のデスクトップ環境を利用できる仕組みを実装した．UNIX 系 OS では，作成したファイルやアプリケーションの設定も含め，ユーザのプロファイルは全てホームディレクトリへと収納される．そこで，SHFS を用いることでホームディレクトリをネットワーク上に用意し，デスクトップの移動を実現した．すなわち，ssh が通るネットワークに接続された PC があれば自分専用のマシンとして利用できるのである．

このKNOPPIXとSHFSの組合せの概要は[3]で提案しているが，本稿ではその実装の詳細，及び，ノマディックデスクトップのサービス提供も視野に入れた性能試験の結果について述べる．

2. KNOPPIXとは

KNOPPIX は CD-ROM から PC を起動することができる Linux ディストリビューションの一つである．これは Debian GNU/Linux[4]を元に Knopper が開発を進め，須崎らにより日本語化が行われ一般に公開されている[5]．

CD-ROM 起動のため，ハードディスクへのインストールが不要である．つまり，他の OS がインストールされている PC でも容易に Linux の環境を立ち上げることができる．また，デバイスの自動認識機能に優れており，一般的な PC のハードウェア構成では殆どのデバイスを人の手による設定無しでも利用することができる．更に，CD-ROM 中に zlib を用いた cloop 圧縮ファイルシステムが収められており，1.8GB 相当のファイルが格納されている．コンテンツとしては，統合デスクトップ環境 KDE,

オフィススィート OpenOffice.org, ペイント系画像ツール GIMP などの生産性アプリケーションや, Web ブラウザ Mozilla, メイラソフト sylpheed などのインターネットアプリケーションが標準で含まれている.

CD-ROM という書込み不可なメディアから起動するという特性上, ユーザが作成したデータは全てメモリ上へ用意されたファイルシステムへ収められる. つまり, そのまま PC の電源を落とすとデータは消える. このため KNOPPIX には, 外部の着脱可能なメディアに構築されたホームディレクトリを作成して利用する仕組みが備えられている. CD-ROM 上にあるファイルシステムを参照することは, ユーザが容易にソフトウェアの構成を変更できないことを意味するが, 使い方によって欠点とはならない. 例えば使用している PC の調子が悪くなった場合, その PC を再起動するだけで元の状態に戻るのである. つまり, この特質は高いロバスト性を要求する用途には大変向いている. 特に, 教育機関の実習用端末など不特定多数の人が触る公衆端末などで障害が発生した場合, 電源を入れ直すだけの対応だけで十分となる可能性が考えられる.

3. SHFS とは

Linux Kernel には VFS(Virtual Filesystem Switch)のレイヤがあることから, さまざまなファイルシステムの実装がある. この仮想レイヤを応用するとで, ユーザ空間のプロセスを経由した擬似的ファイルシステムも開発されており[6], ftp を用いた ftpfs や WebDAV を用いた davfs などがある. SHFS はその中の一つで, ファイル操作や内容の送受信に ssh を用いるネットワーク透過型のファイルシステムであり, Spousta が Malita の ftpfs[7] を元に開発した.

SHFS の機能は, モジュールとしてカーネルへ組み込まれ, VFS のソフトウェアレイヤに新しいファイルストアを提供することで実現される. ユーザ空間のプロセスが発行したシステムコールは, shell のコマンド文字列へ変換され, それを ssh 経由でリモート実行することによりファイルを操作し読書きが行われる. つまりシステムコールから shell コマンドへの変換/リモート実行/戻り値である文字列の解析が行われる. リモート側のマシンで使用されるコマンドは, その OS の種類にも拠るが一般的な UNIX 系 OS では, chgrp, chmod, chown, cut, dd, df, expr, ln, ls, mkdir, mv, printf, rm, rmdir, touch, uname, wc を用い, Linux の場合には head も使用される. 勿論, VFS に来たシステムコールを全てリモートへ転送するのではなく, キャッシュを用意することで負荷低減を図られているが, このような仕組みのためオーバーヘッドにより, NFS と比べて転送効率の面で劣る. しかし, ssh 自体の機能にユーザ認証, 通信経路の暗号化があり, 一般的な NSF と比較して安全性が高いと言える. 特に通信経路の暗号化については, LAN のみならずインターネットなどの WAN 経由で利用する場合において有用である. 一方, ssh のポートフォワードを利用した NFS も存在するが, 利用に際しサーバ側にルート権限が必要になることもあり, アドホックな用途のために使用されることは無い.

SHFS と同様のネットワーク透過型仮想ファイルシステムである ftpfs や davfs などでは, 用いているプロトコル自体に分散オーサリングの性質を持つので, ファイル内容の一部を参照する場合にはファイルの転送完了を待ってから参照を行う必要がある. しかし, SHFS では部分的参照のためのコマンドをリモート上で実行するので, ランダムアクセスが要求される使い方ができる. よって ssh の使用が許可されたマシンがあれば, それに接続されたファイルストレージをあたかもローカルにあるファイルシステムの様に扱うことができる.

つまり, sshd が稼動しているマシンにアカウントがあれば, そのマシンをリモートのファイルストレージとして利用できるので, ファイルシステム提供側で特殊なサーバを用意する必要も無く, 安全かつ容易にファイルシステムを拡張できるのである.

4. ノマディックデスクトップとは

ノマド(遊牧民)は生活必需品を全て持ち歩くことで, 何処へ移動しても日常の生活を維持できる. この概念をデスクトップコンピューティングにあてはめ, ユーザが何処へ移動しても自分専用のデスクトップを利用できる環境をノマディックデスクトップと呼ぶこととした. そして, “ネットワークで繋がったハードウェアは遍在する”と仮定し, メディアに入ったソフトウェアとネットワーク上にあるユーザのプロファイルによって提供される自分専用のデスクトップを, 新しいノマディックデスクトップと定義した.

作成したファイルを含めユーザのプロファイルは, ネットワーク上に用意されたサーバに永続的に保存される. そして, それを参照することによりパーソナライズされた環境がユーザの居る場所で再現される. つまり, ユーザに併せてデスクトップが移動することになる. 今回, プロファイルの取得のためにユーザ認証のあるファイルシステムを採用した. この認証結果をシステムに反映させる, つまり, その譲渡されたユーザ権限でデスクトップ環境を提供するので, ユーザ認証を必要とするアプリケーションも利用できることになる.

5. 実装

産業技術総合研究所(AIST)が公開した KNOPPIX3.3 日本語版(knoppix_20031103-20031119)を元に 3 種類の処理を追加した.

5.1. SHFS 上へのホームディレクトリの準備

KNOPPIX では, ハードディスク内や, USB メモリーキー, コンパクトフラッシュなどのような着脱可能な外部のメディアに永続的なホームディレクトリを構築するためのコマンド mkpersistenthome が用意されている. このコマンドでは, ネットワークや画面の解像度などシステム固有の設定も保存できる. しかし, 今回は PC のハードウェア構成に依存する必要はないので, ユーザプロファイ

図 1. KNOPPIX-SHFS のログイン画面

ルであるホームディレクトリを保存する処理の部分を参考に，SHFS 上へホームディレクトリを構築する新たなコマンドを用意した．これは，リモートにあるユーザのホームディレクトリへ ext3fs のファイルシステムイメージを展開し，それを SHFS 経由でループバックマウントし必要となる設定ファイルを転送する．ここで，SHFS でマウントしたディレクトリを直接ホームディレクトリとせずに，Linux ネイティブのファイルシステムをループバックでマウントする理由は，次のとおりである．

SHFS のファイルストアでは VFS からのファイル操作に対してソケットファイルシステムを扱う処理が実装されていない．そのため，プロセス間通信を行うためのソケットを作成することができない．つまり KDE などホームディレクトリへソケットファイルを生成するアプリケーションの起動に失敗する．そこで試行を繰り返した結果，Linux ネイティブのファイルシステムをループバックでマウントすることにした．

また，ファイルシステムのイメージファイルが破損した場合に備え，ジャーナリングファイルシステムである ext3fs を選択した．ホームディレクトリをこのイメージファイルのマウントポイントとし，イメージファイルを SHFS 上へ配置することにした．正確には，SHFS で用意したストレージにある ext3fs イメージの上にホームディレクトリを用意する，ということになる．このイメージファイルのサイズは，100Mbytes とした．

5.2. セッション開始の処理

通常，KNOPPIX の起動シーケンスでは /etc/init.d/knoppix-autoconfig により，デバイス，ネットワークの設定が終了した後，/etc/init.d/xsession によって X サーバが起動され，knoppix ユーザの権限で /etc/X11/xinit/xinitrc が実行されることでデスクトップを使用できる状態になる．

今回は X サーバが起動した直後，ユーザに ssh 先のアカウント情報の入力を促すため，2 つのテキストボックスを持つ xdm(X Display Manager)に似たインタフェースのダイアログを表示した．ユーザはアカウント情報として，ID の部分に“ユーザ名@ssh のアドレス”とパスワードを入力する．図 1 は，セッション開始時のインタフェースとなるログイン画面で，ユーザ“tanh”が ssh 先である“legend.alpha.co.jp”に接続するところである．この ssh 先でホームディレクトリとなる ext3fs のイメージファイルが存在することの確認を行う．これに成功したら，そのアカウント情報を用いてローカルのマシンへユーザを追加することで，KNOPPIX を立ち上げたマシンにユーザ情報を反映させる．この処理により，そのマシン上で，画面のロックなど，ユーザ認証を必要とするアプリケーションを利用することが可能となる．つまり，リモートマシンのユーザ認証を経由してユーザ権限をローカルのマシンへ譲渡していることになる．これは NIS+，や NetInfo，LDAP などのディレクトリサービスを用いたアカウント認証とは異なる，軽量型のユーザ認証システムへと発展できると考えている．

アカウントの確認後，ホームディレクトリとなるファイルシステムのマウントを試みる．リモートの所定のディレクトリパスを SHFS でマウントし，そのディレクトリにある ext3fs のイメージファイルをホームディレクトリとしてマウントする．図 2 は，アカウントを“tanh@legend.alpha.co.jp”とした場合における KNOPPIX-SHFS のディレクトリ構造である．アカウント情報が正しいなら，/mnt/tanh@legend.alpha.co.jp をマウン

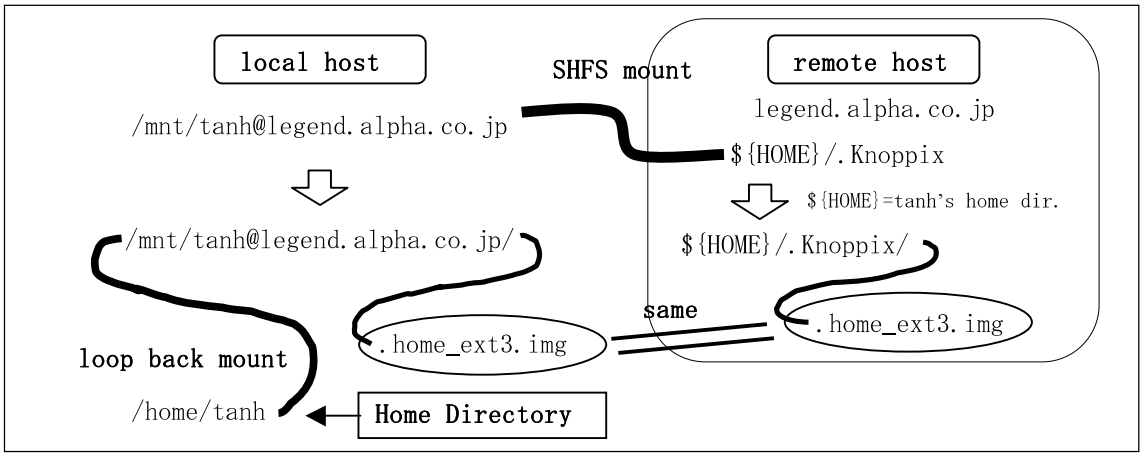

図 2. KNOPPIX-SHFS のディレクトリ構造
アカウントを“tanh@legend.alpha.co.jp”としてセッションを開始した例．

トポイントとしてリモートである legend.alpha.co.jp の ~tanh/.Knoppix をマウントする. この中にある ext3fs のイメージファイルは，ローカル側で /mnt/tanh@legend.alpha.co.jp/.home_ext3.img のパスで参照され，/home/tanh へとループバックでマウントされる. これが KNOPPIX を立ち上げたマシンでのホームディレクトリであり，このマウントの成功後，新たに加えたユーザの権限で/etc/X11/xinit/xinitrc を実行し，デスクトップのセッションが開始される. このように，リモートのマシンは，ストレージの提供元だけでなく，ユーザ権限の認証元にもなるのである.

5.3. セッション終了の処理

通常の KNOPPIX では，ユーザが KDE を終了するとシステムのランレベルが 0 へ移行しシャットダウンの処理が開始される. ホームディレクトリに，SHFS 上のフィアルシステムイメージをループバックでマウントしている KNOPPIX-SHFS の場合には，ネットワークインタフェースが無効になる前にホームディレクトリのマウントを解除する必要がある. なぜなら，マウントしている状態でネットワークを遮断すると，ホームディレクトリのファイルシステムに整合性が取れなくなりイメージファイルを破損する恐れがあるためである. そこで，KDE のセッションが終了した直後の部分に上記のアンマウント処理を加えた. 残念なことであるが，これらの変更により PC の電源を切断する前に正しくシャットダウンの手順を踏むことが必要になった.

6. 性能評価

6.1. 通常の KNOPPIX との比較

通常の KNOPPIX の起動では，tmpfs(仮想ファイルシステム)上に，ホームディレクトリを準備して使用することになる. この tmpfs と SHFS 上にホームディレクトリがある場合についてログイン開始から KDE の起動までにかかる時間，そして，アプリケーションでの作業を行った際にかかる時間，そして，SHFS を用いた際のネットワーク上を流れるデータ量を測定した.

期待される処理の終了直後，sync コマンドを発行し，カーネルへファイル入出力バッファのフラッシュを要求し終えた時刻を操作の終了時刻とした. 計測は，全ての手順を 3 回繰り返し，その平均値を求め，かかった時間，転送量とした.

クライアント側に PentiumIII-750MHz, 256MB, 24xCD-ROM のマシン，サーバ側に Pentium4-2GHz, 512MB, HD:40GB/7,200rpm, (Linux2.4.22, FS:ext3)のマシンを用意し，Switching HUB を経由して 100Base-TX を用いて接続した. 時間については，該当処理の前後で/proc/uptime を記録し，データ転送量については，VMware4 で仮想計算機(ホスト:前述サーバマシン，ゲストメモリ:256MB)を用い閉鎖系のネットワークを構築して計測した.

表 1. ホームディレクトリが tmpfs, SHFS 上にある場合におけるセッション開始から KDE の起動までにかかる時間とクライアントから見たデータ転送量.

tmpfs	SHFS		
time(sec.) [a]	time(sec.) [a]	TX (Kbytes) [b]	RX (Kbytes) [b]
94.4	79.3	982.8	699.9

[a] 実機を用いた計測. [b] VMware を用いた仮想ネットワークでの計測.

6.1.1. KDE デスクトップの起動完了時間

カーネルの初期化完了から KDE デスクトップが起動完了になるまで，つまり，ユーザが GUI で作業を開始することが出来る状態になるまでの時間，そしてクライアント側からみたデータ転送量を表 1 に示す. tmpfs, SHFS 間では，tmpfs の方が，15 秒ほど遅い結果となった. これは，ホームディレクトリ準備でのファイルコピーにおいて CD-ROM ドライブのシークに時間を取られてしまうからであると考えられる. カーネルの起動からログイン画面が現れるまで 100 秒程であったので，SHFS をホームディレクトリとして利用するユーザは，PC 起動からアカウント情報の入力を経て作業開始となる状態まで，約 3 分待つことになる.

6.1.2. アプリケーションでの作業

アプリケーションとして OpenOffice.org(OOo)を選び，簡単なシナリオに沿って順番に操作を行った. その操作内容，各処理にかかった時間，そしてデータ転送量を表 2 に示す. シナリオ 1 でのスプレッドシートの新規作成では，半角 8 文字のセルを 500 行 78 列にコピーし，約 6Kbytes のファイルをホームディレクトリへ生成した. また，シナリオ 3 で用いた日本語 Word サンプルはサイズ 964Kbytes の MS-Word97 形式のファイルで AIST 公開の日本語版 KNOPPIX の CD-ROM に入っている. このファイルが入っているフォルダアイコンをクリックすることで GUI ファイル操作アプリケーションである Konqueror を起動し，そのウィンドウから OOo でフィアルを開き，ファイルフォーマットを OOo 形式に変換してホームディレクトリにセーブする操作を行っている.

シナリオ 1 で OOo のユーザ初回起動時は，tmpfs の場合 99 秒であったが，SHFS では 165 秒と 1.6 倍以上の差が出た. データ転送量が送信で 10Mbytes 以上あることからも判るように，OOo が設定ファイルやリファレンスマニュアルなど，約 8,760Kbytes のファイルをホームディレクトリへコピーするためである. 2 回目以降の起動ではこのような極端な差は見られない. シナリオ 2 でのホームディレクトリに格納されたファイルを読込む作業においては，SHFS が 5 秒ほど遅くなった. しかしシナリオ 1，シナリオ 3 でのファイル書込みでは差がなかった. 初回起動において 1 分以上の差が出たのは，ジャーナリングファイルシステムである ext3fs への大量書込みによって，ジャーナルの更新に時間を要したためであると考えられる. しかし，読込みではジャーナリングは殆ど影響しないはずであるので，シナリオ 2 でのファイル読込みが遅くなる原因とは考えにくい.

表 2. ホームディレクトリが tmpfs, SHFS 上にある場合における OpenOffice.org(OOo)での作業にかかる時間とクライアント側からみたデータ転送量.

シナリオとその操作	tmpfs	SHFS		
	time(sec.) [a]	time(sec.) [a]	TX (Kbytes) [b]	RX (Kbytes) [b]
1) KDEパネルからOOo起動	99.1	164.9	11024.6	804.8
スプレッドシートの新規作成	26.5	26.5	49.8	14.9
シートをホームディレクトリへ保存	24.9	25.5	59.7	27.2
OOo終了	4.8	4.5	83.5	18.2
2) KDEパネルからホームディレクトリを参照	18.0	19.2	343.5	186.6
作成したシートを開く	28.6	33.9	121.3	432.3
OOo終了	4.0	3.8	50.8	8.9
3) デスクトップの日本語サンプルフォルダを参照	6.4	8.0	76.3	52.7
Wordサンプルを開く	36.4	36.5	120.9	46.6
フォーマット変換してホームディレクトリへ保存	21.2	21.0	362.7	60.2
OOo終了	4.8	4.0	79.7	15.1
4) KDEパネルからOoo起動	13.0	13.2	85.0	26.2
OOo終了	7.6	3.1	53.4	9.3

[a]実機を用いた計測. [b]VMware を用いた仮想ネットワークでの計測.

6.2. 二段構成のファイルシステムの性能

実装の部分で述べたように, SHFS 上にソケットファイルを作成することができない. そこで, SHFS 上にある ext3fs のイメージファイルをループバックでマウントする構成にした. それ故, SHFS を直接使用する場合と, SHFS 上にあるファイルシステムのイメージをループバックでマウントした ext3fs を使用する場合とでの比較が必要になる. ここでは, 両者のファイルシステムに対して dd コマンドを用いファイル読書きの速度について測定した結果を述べる.

/dev/zero を入力としたクライアントからの書込み, その書込んだファイルを直ちに読込む場合, そして, すでにファイルシステム上に用意されたファイルを読込む場合について処理にかかる時間を計測した. 測定は, 通常の KNOPPIX との比較で用いたものと同様のマシン構成で行い, dd コマンドの終了後, これも同様に sync コマンドの発行が終了した時点においてファイルの読書きが終了したものとした. また, データ転送量も同様に VMware の環境を用いて測定した. 読書きするファイルのサイズは, 1Mbytes とし, dd コマンドのオプションであるブロックサイズとカウント数の組合せを変化させ測定したが, 終了時間に大きな違いが見られなかったため, ここでは, ブロックサイズを 256Kbytes, カウント数を 4 としたものについての結果を表 3 に示す.

SHFS を直接使用する場合は, 読込みに比べ, 書込みの方が速い. また, SHFS はデフォルトでキャッシュを 64Kbytes 使用するが, このキャッシュの効果は殆どなく, キャッシュが期待されるであろう, 書込んだファイルを直ちに読込んだ場合の方が, 既にあるファイルを読込む場合に比べて若干遅くなっている. 一方, ext3fs の方は逆で書込みに比べ読込みの方が速い. しかし, 書込みが極端に遅くジャーナルの更新が原因なのか SHFS の 20 倍以上の時間を要した. ところが, 読込みでは SHFS の 2 倍ほどの時間しかかからなかった. また, ext3fs 自体のバッファのおかげにより, ファイル書込み直後の読込みでは, そうでないものと比べ 14 倍近くの速さで終了する, つまり, 読書きが連続して起る場合はユーザにとってプロセスのレスポンスが向上することになる. 今回の SHFS 上においたファイルシステムのイメージファイルをループバックでマウントするディレクトリ構成は, 大量の書込みには全く向いていないことが判明した. この結果は, 表 2 のシナリオ 1 における OOo のユーザ初回起動において, SHFS をホームディレクトリとした場合, 通常の KNOPPIX より時間がかかったことと同じことを示している.

しかし, 今回の実験では sync コマンドでファイルシス

表 3. SHFS, ext3fs on SHFS への dd コマンドによるサイズ 1024Kbytes のファイル読書きに費やした時間とクライアント側からみたデータ転送量.

	SHFS			ext3(ordered) on SHFS		
	write	read	read (cached)	write	read	read (cached)
time(sec.) [a]	0.93	1.40	1.53	19.84	2.64	0.19
TX(Kbytes) [b]	1192.1	22.8	19.9	1172.0	23.7	4.7
RX(Kbytes) [b]	60.8	1081.3	1078.7	108.3	1158.3	0.8

[a]実機を用いた計測. [b]VMware を用いた仮想ネットワークでの計測.

表 4. クライアント 8 台を 1 台のサーバに接続した場合における各操作にかかった時間.

操作	クライアント数 1台 time(sec.)	8台 time(sec.)
セッション開始から KDEの起動完了まで	64.4	min. 73.3 max. 163.4 ave. 93.1
OOoユーザ初回起動	104.1	min. 581.2 max. 661.9 ave. 606.2
Wordの日本語サンプルを開く	2.3	min. 2.5 max. 8.3 ave. 6.6

テムのバッファをディスクにフラッシュすることの要求が終了した時点をファイル書込みが完了したものと見なしており, 処理が完全に完了することを待った結果, 遅いということになる. 通常のデスクトップ使用の際には, キャッシュの効いた読込み時の速度も合わせ, 書込み時もキャッシュを使用するので快適に使用できるものであると考えている.

6.3. クライアント 8 台をサーバに接続した構成での性能評価

1 台のマシンをサーバとして, 異なるアカウントでクライアントを 8 台接続し, 一斉ログインや OOo の起動などを行った. クライアント側に Pentium4-2GHz, 512MB, 48xCD , サーバ側に Pentium4-2.8GHz, 1024MB, HD:80GB/7,200rpm, (Linux2.4.22, FS:ext3)のマシンを用意し, switching hub を経由して 100Base-TX で接続した. 今回も 1 台のみでの実験と同様, 期待される処理の終了直後の sync コマンドが終わった時刻を処理の終了時刻とした. この計測を行った操作と結果を表 4 にまとめた. ログイン開始から KDE が起動して作業の開始ができるまでの時間が, 1 台の場合に比べて, 平均約 1.5 倍程で, もっとも遅いクライアントであると 2.5 倍もの時間がかかった. カーネルの起動からログイン画面が現れるまで約 78 秒であったので, PC の起動からでは, 平均約 3 分, もっとも遅い場合には, 約 4 分も待つことになる. OOo の初回起動では, 1 台のとき約 1.5 分程であったものが, 8 台では 10 分と, アプリケーションの起動にしてはかなり遅い結果となった. しかし, 一旦起動した後では, 必要なファイルがキャッシュに残るので, Word の日本語サンプルを開く操作では, 数秒以内での起動が確認できた. ただし, 設定ファイルを参照するときに他のクライアントと競合するようで, 1 台の時と比較して 3 倍ほど時間がかかっている.

この一斉ログイン, OOo の初回起動の実験中, サーバ側において CPU 使用率が 100%で dd のプロセスが 8 個同時に動いている状態が観察された. dd コマンドによりシステムコールが連続で呼ばれサーバ側のカーネルにかなりの負担がかかっている状態であると推測できる.

7. まとめ

CD-ROM から起動する Debian GNU/Linux である KNOPPIX と Secure Shell を利用したネットワーク透過型ファイルシステムである SHFS を組合せ, ネットワーク上に格納されたプロファイルを参照することでユーザの使用する PC に依存することなく自分専用のデスクトップ環境を提供できる仕組みを実装した.

選択したファイルシステムの特性もあり, 大きなサイズのファイル書込みでは幾分パフォーマンスが悪い結果となった. しかし, ホームディレクトリにキャッシュファイルを作るアプリケーションではローカルマシンの tmpfs 上に作成する設定を行うことや, OOo の初回起動時におけるファイルの大量書込みを止めるなど, 工夫をした運用次第では, 十分利用可能であることが判った. 一方, サーバに大人数がアクセスする場合などの大規模な運用では, KNOPPIX のシステムを含め SHFS 自体の改良を考える必要がある.

PC の起動も含めログインしてから GUI で作業を開始できるまでに時間を要する点は, ソフトウェアの入っているメディアが CD-ROM であるという, 1CD の Live Linux の短所の一つである. 一方, インストール不要である点やポータビリティなどの長所もあるので, 起動時間が長いかどうかは, このシステムを適用するユーザの利用方法に拠るところが大きい.

しかし, ログインしてから作業開始の状態になる間 1 分以上待つことは, どんなユーザにとっても負担となるので, 起動完了までの高速化が望まれる. また, このノマディックデスクトップ環境のサービスを視野に入れた場合, サーバにかかる負荷の大きさが問題となるので, スケーラビリティの確保も必要になる. そして, サーバとなる shell アカウントがあるマシンをインターネットに接続する際のセキュリティの問題もある. 今後は, これら課題の解決を図りたい.

参考文献&URL

[1] Knopper, "Building a self-contained auto-configuring Linux system on an iso9660 filesystem, 2003
http://www.knopper.net/knoppix/
[2] SHFS http://SHFS.sourceforge.net/
[3] 丹, *et al.*, “KNOPPIX と SHFS を用いたノマディックデスクトップの提案”, 情報処理学会 第 66 回全国大会, 5-F3, 2004
[4] Debian http://www.debian.org/
[5] 須崎, 飯島, “デスクトップとしての日本語 KNOPPIX”, FIT 情報科学技術フォーラム講演論文集, p263, B-057, 2003
http://unit.aist.go.jp/it/knoppix/
[6] Linux Userland Filesystem
http://lufs.sourceforge.net/lufs/
[7] FTPFS http://ftpfs.sourceforge.net/