

UML-KNOPPIX の起動における性能評価

後藤和弘[†] 須崎有康[‡] 飯島賢吾[‡] 丹英之[§]

概要 CD ブータブルな Linux である KNOPPIX は多くの言語への対応や、教育や医療など特定分野向けの派生版が開発されているが、CD 作成などの手間も多い。User-mode Linux を用いることでそのような派生版 KNOPPIX を CD に焼くことなく起動できる。本研究では、不特定多数のユーザが WAN 環境で派生版 KNOPPIX を起動できるよう、ネットワーク対応ファイルシステム SFS を利用してルートファイルシステムを含んだ cloop ファイルを読み込めるようにし、起動時間や転送データ量等について評価した。

Evaluation of Booting UML-KNOPPIX

Kazuhiro GOTO[†] Kuniyasu SUZAKI[‡] Kengo IJIMA[‡] Hideyuki TAN[§]

Abstract. KNOPPIX is one of Linux distribution that can be boot from CD drive, and is customized to various languages and some specific field, such as education and medical science and so on. Using User-mode Linux, the customized KNOPPIX can be booted without CD media. In order to use the customized KNOPPIX on WAN by anonymous user, compressed loop-back device “cloop” file that includes KNOPPIX root file system is readable using Self-certifying File system. We evaluate transferred data and boot time that is difficult to measure with booting sequence from CD media.

1. はじめに

CD ブータブルな Linux ディストリビューションの 1 つである KNOPPIX^[1] は、ハードディスクへのインストールが不要で Linux 環境を利用できる手軽さとオープンソースであることが特長であり、日本語化^[2]をはじめ様々な言語へ対応したものや、医療や教育分野など特定の用途向けにカスタマイズされたものなど、50 種類以上の派生版 KNOPPIX が開発されている。

従来、そのような派生版 KNOPPIX を利用するには iso イメージファイルをあらかじめ CD に焼く必要があったが、UML (User-mode Linux)^[3] を利用することによって、cloop ファイルを読み出し、派生版 KNOPPIX を利用可能となった^[4]。ネットワークファイルシステムで cloop ファイルを共有することにより、ブロードバンド環境から KNOPPIX を利用できるものの、WAN 上でデータを転送するにはセキュリティによる保護が不可欠である。また、NFS は UDP を用いるために通常のファイアウォールでは通常は通過許可されないため、

UML を利用するにはセキュアなファイルシステムが必要となる。

本研究では、WAN 環境において UML を利用して安全に KNOPPIX を実行することを目指して SFS (Self-certifying File System)^[5] を採用する。そして、SFS サーバから cloop ファイルを読み出して KNOPPIX を起動するために必要な回線の帯域やデータ転送量などについて評価する。

2. KNOPPIX とは

KNOPPIX は CD から起動できる Linux ディストリビューションのひとつで、ドイツの Klaus Knopper 氏により開発が進められている。

2.1 主な特長

KNOPPIX の主な特長は、デバイスの自動認識・設定機能である AutoConfig、および、圧縮機能をもったループバックデバイス cloop である。

AutoConfig は、ハードディスク、ビデオデバイス、ネットワークカード、サウンドカード、USB デバイスなどを自動的に認識し、適切なドライバを利用できるように設定する。これにより、IDE や SCSI など搭載しているハードディスクのファイルシステムを識別したり、適切なビデオカードを認識して X ウィン

[†] 大分県産業科学技術センター, Oita Industrial Research Institute

[‡] (独) 産業技術総合研究所, National Institute of Advanced Industrial Science and Technology

[§] (株) アルファシステムズ, Alpha systems Inc.

ド環境が利用できるようになる。また、ネットワークカードについては、DHCP 環境があれば起動してすぐに WWW ブラウザなどを利用できる。

KNOPPIX は、zlib で圧縮された cloop (compressed loop device) ファイルにルートファイルシステムを含んでおり、起動時に必要なファイルを解凍しながら読み出しをおこなう。ループバックデバイスを利用すると、ファイルをファイルシステムとしてマウントできる。cloop を使うと CD ドライブから読みこむデータ量は少なく済み、解凍は CPU の能力に依存するため、低速な CD ドライブを利用する場合でも読み出し時間への影響は少ない。

2.2 カスタマイズとその利用

KNOPPIX のオリジナルはドイツ語版が開発されているが、オープンソースであるために日本語版をはじめとする様々な言語への対応や、医療や教育など特定分野向けのカスタマイズ、ユーザによる独自のアプリケーションの追加やパッケージの削除など、数十種類を越える多くの派生版 KNOPPIX が開発されている。これまではそのような派生版 KNOPPIX を利用するためには、700MB 近い ISO ファイルをインターネットからダウンロードし、CD-R へ書き込まなければならなかった。しかし、ブロードバンド環境でないユーザや、CD-R への書き込み方法に詳しくないユーザには、必ずしも簡単な手順とは言えない。また、ユーザが自分で開発した派生版 KNOPPIX を動作検証する場合も、CD-R へ書きこむか、仮想計算機環境を提供する商用のソフトウェアを利用する必要があった。

3. User-mode Linux による KNOPPIX の起動

User-mode Linux を利用することで、派生版 KNOPPIX を CD-R に焼くことなく、ユーザアプリケーションとして実行可能となる。以下ではその方法について説明する。

3.1 User-mode Linux

User-mode Linux (以下、UML と略記) は「ユーザプロセスとして実行可能な Linux カーネル」で、Jeff Dike 氏によって開発が進められている。商用の VMware など同様の仮想計算機環境を提供するが、UML はオープンソースとして開発されている。UML を使うことで、バグを含む可能性のあるソフトウェア

や新しい Linux カーネル、ディストリビューションなどを安全にテストすることやデバッグすることができる。UML の Web サイトには UML 本体だけでなく、起動可能なルートファイルシステムも用意されているので、それらをダウンロードすることですぐに試すことができる。UML では、CPU 上で起動する Linux をホスト OS、そして、ホスト OS 上で UML によって起動される Linux をゲスト OS と呼ぶ。

3.2 UML を利用した KNOPPIX の起動

UML を用いると、Web サイトからダウンロードした派生版 KNOPPIX や自分でカスタマイズした派生版 KNOPPIX を CD-R へ書きこまずにゲスト OS として起動できる。標準的な UML では cloop のドライバは含まれていないが、KNOPPIX では cloop のドライバを含むように UML カーネルを作り直してブート可能にしている。図 1 に UML 起動の概要を示す。CD から起動した KNOPPIX がホスト OS で、ホスト OS から UML によって起動された KNOPPIX がゲスト OS である。ゲスト OS はローカルコンピュータ上の cloop ファイルをループバックマウントすることで、ゲスト OS 上のファイルのようにアクセス可能となる。以下ではこのようにして起動したゲスト OS を「UML-KNOPPIX」と呼ぶ。

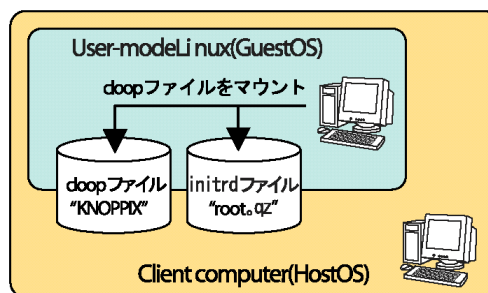


図 1 cloop ファイルによる UML-KNOPPIX 起動

UML-KNOPPIX を起動するには、初めに umlhub や IP マスカレードなどの仮想ネットワークを設定する。ゲスト OS は dhcp により IP アドレスを自動的に割り当てられ、ホスト OS やホスト OS 以外の外部のネットワークとも通信できる。また、複数の UML-KNOPPIX 間で相互に通信することも可能である。UML では仮想ビデオカードを提供しないため、そのままでは GUI が使えない。そこで、Xnest を利用することでゲスト OS の GUI をホスト

OS 上に表示する。Xnest はホスト OS 上の X クライアントとして動作すると同時に、ゲスト OS の X サーバとなることができる。これにより、Xnest が提供する X 環境上で KDE などのデスクトップマネージャを実行できる。ゲスト OS からホスト OS への X の接続を許可するために xhost を実行した後、指定した cloop ファイルを読みこんで UML-KNOPPIX を起動する。起動時のオプションによって、cloop ファイルを指定するほかにも、デスクトップマネージャの選択や画面サイズの指定も可能である。図 2 に UML-KNOPPIX を起動したときの画面例を示す。

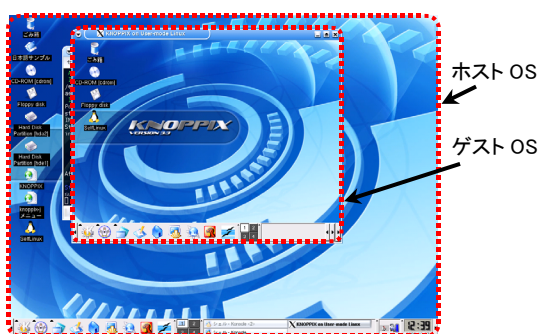


図 2 UML-KNOPPIX の起動例

3.3 複数のディストリビューションへの対応

UML-KNOPPIX を起動するためのパッケージは、KNOPPIX 用の Debian パッケージ以外に TurboLinux や FedoraCore, Gentoo などのディストリビューション用も開発している。通常、アプリケーションのインストールやアップデートを行なうには apt や rpm などディストリビューションによって方法が異なるが、UML を利用することで、各種ツールやデスクトップ、その他の設定などを含んだ実行環境を 1 つのユーザアプリケーションとして利用可能となる。TurboLinux 上で UML-KNOPPIX を起動した場合の画面例を図 3 に示す。



図 3 TurboLinux での UML-KNOPPIX の起動

4. SFS を利用した UML-KNOPPIX の起動

cloop ファイルをインターネットで公開することで、UML を利用していつでも誰でもどこからでも派生版 KNOPPIX を利用可能となる。以下では WAN 環境における UML-KNOPPIX の起動や、ネットワーク対応ファイルシステムについて説明する。

4.1 SFS (Self-certifying File System)

SFS はセキュリティで保護されたネットワークファイルシステムで、SFS サーバとクライアント間の通信に SSH2 プロトコルを使用する。ユーザは SFS サーバの公開鍵の暗号化ハッシュを含んだパスを指定し、下記のようにローカルマシン上のファイルと同様に”cd”コマンドや”ls”コマンドを用いてサーバ上のファイルにアクセスできる。

```
$ cd /sfs/@host.domain.jp,fuqbn...sq2yh/
```

ここで、”host.domain.jp”は SFS サーバの FQDN 名、それ以降に続く 32 文字のランダムな英数字はホスト ID と呼ばれる。

クライアントコンピュータにおいて SFS クライアントデーモン sfsd を起動すれば、ルート権限による明示的なマウント操作は必要なく、ユーザ権限のままサーバ上のファイルへアクセスできる。SFS サーバでは Anonymous な読み書きや、ユーザ個別のアクセス制限などを細かく設定できるので、セキュリティ面で安全な運用が可能である。

4.2 WAN 環境での UML-KNOPPIX の起動

cloop ファイルを NFS などによってネットワーク共有すると、UML を利用してローカルコンピュータ上で派生版 KNOPPIX を起動できる。しかし NFS は UDP を使うために通常は FireWall で通過許可されず、セキュリティの面からも LAN 環境での利用に限定される。このため、WAN 環境で UML-KNOPPIX を起動するには、ネットワーク対応ファイルシステムが必要となる。

SFS を利用すれば、サーバ上のファイルをローカルコンピュータ上のファイルと同様にアクセスできるため、起動時に SFS サーバ上の cloop ファイルを指定するだけで UML-KNOPPIX を起動できる (図 4)。これにより、700MB 近いファイルの一括ダウンロードや、ダウンロード後に CD に焼く必要がなくなる。

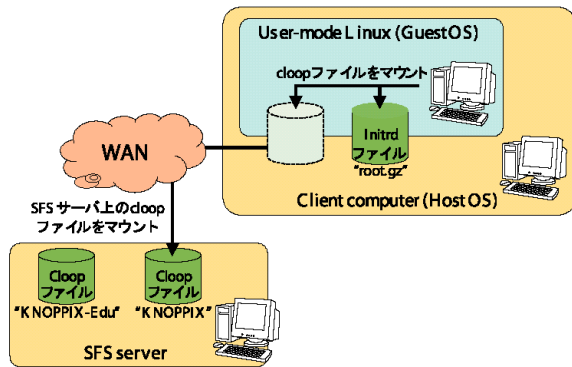


図 4 SFS サーバ上の cloop ファイルマウント

しかし、SFS は NFS をベースとしていることによるオーバーヘッドが存在する。図 5 に示すように、SFS サーバ側では SFS サーバデーモンは NFS クライアントとして動作し、SFS クライアント側では SFS クライアントは NFS サーバとして動作する。このため、UML-KNOPPIX の起動にオーバーヘッドが影響を与える可能性がある。

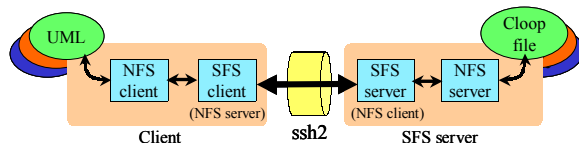


図 5 SFS を利用した UML の起動

4.3 Web ブラウザからの起動

UML-KNOPPIX を簡単に起動できるように、Web ブラウザをインタフェースとして利用できるようにした。ローカルコンピュータで SFS クライアントのデーモンを実行し、Web ブラウザで以下のようなアドレスにアクセスすると SFS サーバ上のファイルを参照できる。

file:///sfs/@host.domain.jp,fuqbn...sq2yh/

SFS サーバ上に UML-KNOPPIX を起動するためのスクリプトを用意し、Web ブラウザでこのスクリプトを選択してマウスクリックする。これにより、ターミナルウィンドウでコマンドライン入力することなく、マウス操作だけで UML-KNOPPIX を起動できる。

5. UML-KNOPPIX 起動の評価実験

UML-KNOPPIX を起動するために必要な回線能力や起動時に回線上で転送されるデータの転送量、起動時間などを評価する実験方法について説明する。

5.1 実験方法の概要

実験環境として、SFS サーバ、および SFS クライアントを図 6 に示すようにスイッチングハブへ接続する。SFS サーバには Pentium4 2.4GHz、メモリ 1GB のコンピュータを使用し、KNOPPIX v3.4 20040510-20040520 版（以下、KNOPPIX v3.4 と略記）をハードディスクへインストールした後、SFS 関連のパッケージを apt-get で追加した。SFS クライアントには Pentium4 2.66GHz、メモリ 512MB のコンピュータを使用し、KNOPPIX v3.4 を CD から起動する。

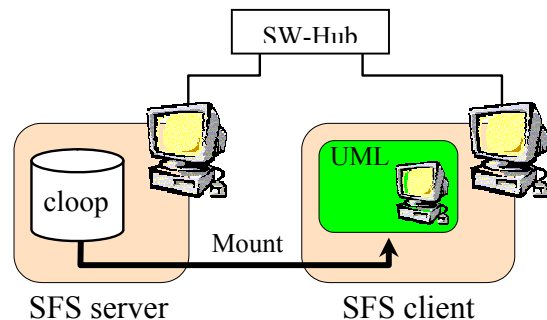


図 6 実験環境

起動時にサーバ・クライアント間で転送されるパケットを tcpdump でキャプチャし、そのデータをもとにスループットなどを計算する。tcpdump を使うことで、フィルタリングによって必要なパケットだけを抽出できるとともに、パケット送出時刻をもとに UML-KNOPPIX の起動に要する時間を正確に測定できる。さらに tcpdump のキャプチャデータを入力とする各種解析ツールを利用できるため、スループットの他にも様々な解析をおこなえるというメリットがある。本実験では tcpstat を利用してサンプリング 1 秒間隔でスループットやデータ転送量などを計算する。各条件について 3 回ずつ計測をおこない、それぞれの計測について起動時間やスループット等を計算する。

5.2 回線の伝送能力の評価

予備実験として netperf を用いて回線の伝送能力を確認した。netperf は 2 点間のパケット伝送能力を計測するツールで、送信側から受信側への一方向のスループットが得られる。netperf の実行中にパケットをキャプチャし、キャプチャデータから tcpstat で計算したスループットと netperf の出力結果はほぼ一致していて 94Mbps 程度であった。このとき、

tcpdump によるパケットのドロップは 0.01%未満であり、実験環境において 100Mbps 近い速度で通信を行なった場合でも tcpdump をもとにした解析が可能であることを確認した。

5.3 UML-KNOPPIX 起動の評価方法

UML-KNOPPIX の起動性能を評価するために、複数のゲスト OS、デスクトップマネージャ、ホスト OS のカーネルバージョン、仮想計算機環境などの条件について計測を行なう。SFS サーバ上に複数の派生版 KNOPPIX の cloop ファイルを用意する。そして、クライアントから cloop ファイルを指定して UML-KNOPPIX を起動したときに転送されるパケットをキャプチャする。tcpdump で SSH2 だけが通過するようにフィルタリングし、X の起動が完了した後にキャプチャを停止してスループット等について解析する。

6. UML-KNOPPIX 起動の性能評価

以下では、評価実験の結果について説明する。複数の派生版 KNOPPIX について起動時間やデータ転送量など基本的な性能を比較した。また、KNOPPIX v3.4 については、デスクトップマネージャによる比較など、さらに詳しく評価した。

6.1 起動時間とデータ転送量

ゲスト OS となる派生版 KNOPPIX として、KNOPPIX v3.4、KNOPPIX/Math^[6]、GIS-Knoppix^[7]について計測を行なった。UML-KNOPPIX 起動時に X が起動完了するまでの時間を求め、その平均値を表 1 に示す。GIS-Knoppix だけが短い時間で起動しているが、これは起動時の初期設定やデーモンの有無などによる影響と考えられる。

表 1 UML-KNOPPIX の起動時間

	Knoppix v3.4	KNOPPIX /Math	GIS-Knoppix
起動時間 (sec)	151	150	120

UML-KNOPPIX 起動時におけるデータ転送量の計算値は図 7、図 8 に示す結果となった。図は 3 回の計測結果からの一例を描画している。図 7 は SFS サーバからクライアントへ転送されたデータ量を表している。データ転送量は起動する派生版 KNOPPIX によって異なるものの、およそ 50MB であった。一方、ク

ライアントから SFS サーバへのデータ転送量は図 8 に示すように 2MB 程度で、そのほとんどは TCP の ack パケットであった。SFS サーバ、クライアント間におけるデータ転送量の平均値を表 2 にまとめた。

図において起動開始後 20 秒程度から約 15 秒間データが転送されない区間は、KNOPPIX のハードウェア自動検出機能である hwsetup が実行されていることがわかった。今後、ホスト OS のハードウェア情報をゲスト OS から読み出す仕組みを検討することで起動時間を短縮できると考えられる。

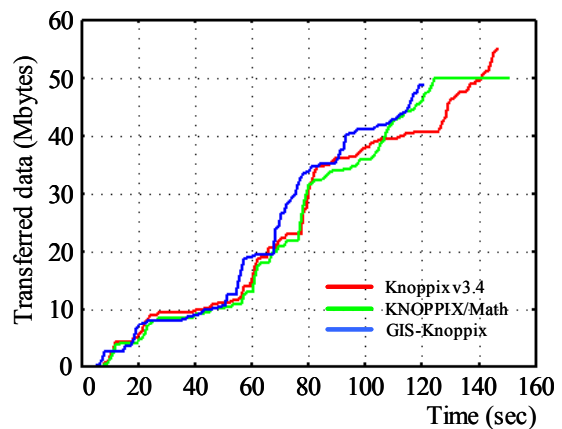


図 7 データ転送量 (サーバ→クライアント)

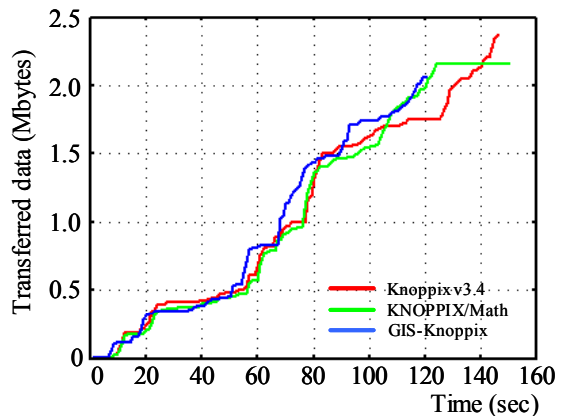


図 8 データ転送量 (クライアント→サーバ)

表 2 UML-KNOPPIX 起動時のデータ転送量

	Knoppix v3.4	KNOPPIX /Math	GIS-Knoppix
サーバ → クライアント (Mbytes)	52.3	47.9	46.1
クライアント → サーバ (Mbytes)	2.3	2.1	2.0

6.2 スループット

UML-KNOPPIX 起動時のスループットの計算結果を図 9, 図 10 に示す. 図 9 はサーバからクライアントへ転送されたデータのスループット, 図 10 はクライアントからサーバへのデータについての計算結果で, 起動シーケンスのあいだにかなり変動していることがわかる. それぞれの派生版 KNOPPIX について 3 回ずつ計測を行ない, スループットの平均値を表 3 にまとめた. 表の上段はスループットの最大値で, 下段は起動処理全体における平均値である. 平均値だけを見ると ADSL でも十分のようであるが, 実際はさらに高速な回線が必要と考えられる. スループットの低い区間(起動開始後 30 秒近辺)と高い区間(同 60~80 秒近辺)に注目して詳しく調査したところ, スループットの低い区間はハードウェア自動検出機能 `hwsetup` を実行していることがわかった. 一方, スループットの高い区間では X ウィンドウやデスクトップ環境 KDE が起動していた. 実験により, UML-

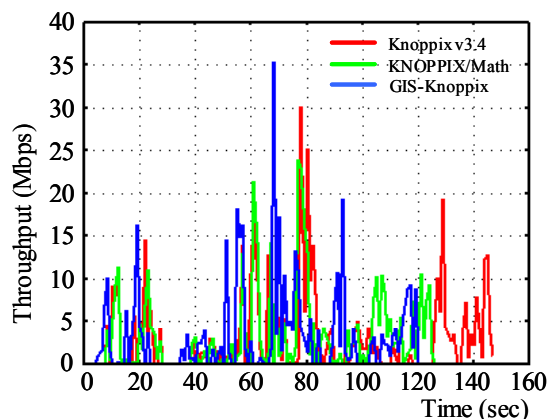


図 9 スループット(サーバ→クライアント)

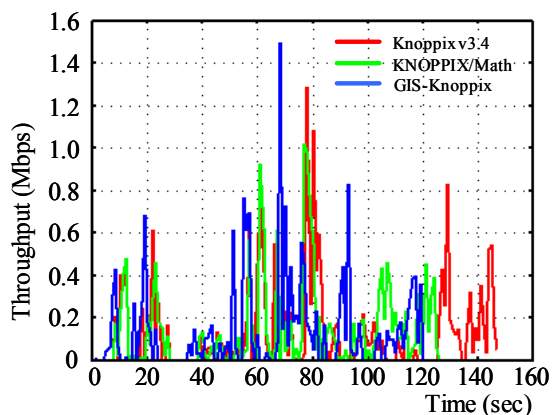


図 10 スループット(クライアント→サーバ)

KNOPPIX を起動する場合のスループットは最大 30Mbps 未満で, 100Mbps 近い回線の転送能力を完全には活かしていないことが明らかになった. これは起動時のボトルネックが回線や NIC ではなく, 起動処理などにあるためと考えられる. 今後は起動処理やデータの転送ブロック単位を見直すなど, 高速な伝送能力を活かせるような対策が必要である.

表 3 UML-KNOPPIX 起動時のスループット
(上段: 最大値, 下段: 平均値)

	Knoppix v3.4	KNOPPIX /Math	GIS-Knoppix
サーバ → クライアント (Mbps)	26.2 2.65	22.6 2.44	26.9 2.93
クライアント → サーバ (Mbps)	1.1 0.04	1.0 0.03	1.2 0.04

6.3 デスクトップマネージャ

UML-KNOPPIX 起動時の最大スループットは X の起動シーケンスに依存していることが明らかになったことから, デスクトップ環境の影響を検証するために 6 種類のデスクトップマネージャ (KDE, xfce, twm, fluxbox, icewm, wmaker) について比較した. 図 11 はデータ転送量, 図 12 はスループットの計算結果をそれぞれ表して, いずれも SFS サーバからクライアントへの転送データについて計算している. 図から KDE ではデータ転送量が約 55MB, 起動時間が約 150 秒であるのに対し, twm など幾つかのデスクトップではデータ転送量は半分程度で起動時間も短いことがわかった. 一方, スループットは KDE が最も高く, 約 30Mbps であった.

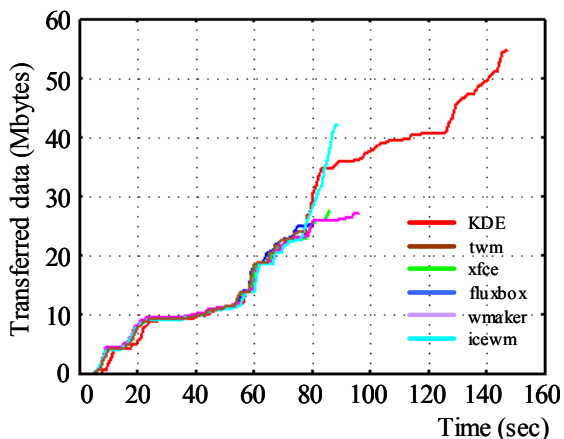


図 11 デスクトップによるデータ転送量比較

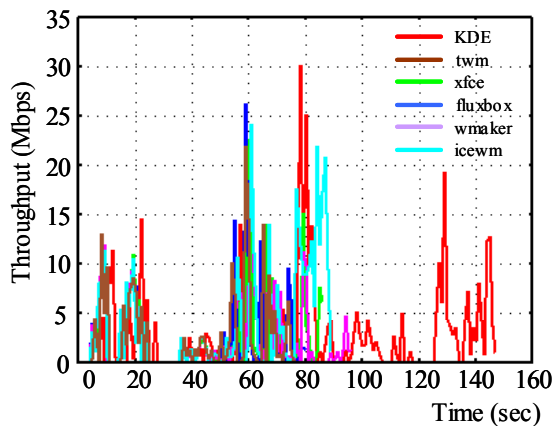


図 12 デスクトップによるスループット比較

6.4 ホスト OS のカーネル

KNOPPIX は v3.4 からカーネル 2.4 に加えてカーネル 2.6 にも対応しており、CD からの起動時に選択できる。そこで、カーネルの違いが UML-KNOPPIX の起動に与える影響について確認した。ホスト OS は、KNOPPIX v3.4 を CD から起動する際にカーネルを 2.4 または

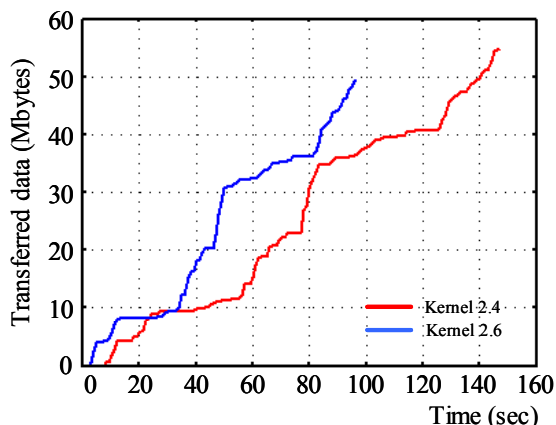


図 13 カーネルによるデータ転送量の比較

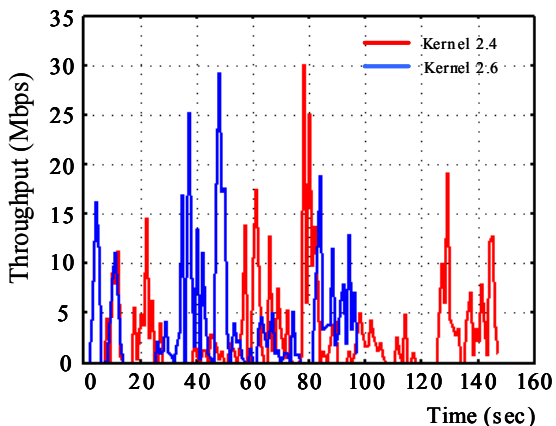


図 14 カーネルによるスループットの比較

2.6 を選択した。また、ゲスト OS は KNOPPIX v3.4 とした。

図 13 は UML-KNOPPIX 起動時のデータ転送量である。最終的な転送量はほぼ等しいが、起動時間はかなり異なることがわかる。ホスト OS をカーネル 2.4 で起動した場合における起動時間は約 150 秒であるのに対し、カーネル 2.6 では約 90 秒であった。実験により、カーネル 2.6 では、カーネル 2.4 よりも 1.7 倍程度高速に起動できることがわかった。

また、図 14 に示すようにスループットの計算結果の最大値はどのカーネルで起動した場合でもほぼ 30Mbps であった。このことは、カーネル 2.6 での起動時間が 2.4 よりも短い理由として、特定の処理だけが高速に実行されることに伴ってパケットが短時間に大量に転送されるのではなく、起動処理全体にわたって高速に処理されていることを示唆する。UML は多くのプロセスを生成するため、カーネル 2.6 におけるスレッドや I/O に関する改善が寄与しているものと考えられる。

6.5 他の仮想計算機環境との比較

UML-KNOPPIX は、ユーザ空間で実行できる KNOPPIX エミュレータであり、新しく開発したカーネルのテストやデバックなどに使うことができる。同様の仮想計算機は商用やフリーソフトなど、幾つか開発されているが、本稿ではフリーの CPU エミュレータである QEMU^[8]について UML-KNOPPIX との比較をおこなった。QEMU は x86 などの CPU 上で、x86、ARM、SPARC、PowerPC などの CPU をエミュレート可能であり、“-cdrom”オプションで CD-ROM イメージファイルを指定することで CD ブートのように仮想計算機環境を起動できる。QEMU は標準的な KNOPPIX の ISO ファイルには含まれないため、KNOPPIX をハードディスクにインストールしてホスト OS とし、QEMU をホスト OS 上にインストールした。そして、コマンド入力により QEMU または UML を利用してゲスト OS を起動する。ゲスト OS は KNOPPIX v3.4 とし、SFS サーバ上に UML-KNOPPIX の起動用には `cloop` ファイルを、QEMU の起動用には `iso` ファイルをそれぞれ用意した。そして、QEMU、UML-KNOPPIX を利用し、ネットワーク上のファイルを利用してゲスト OS を起動したときのパケットをもとに転送データ量とスループットを計算した。

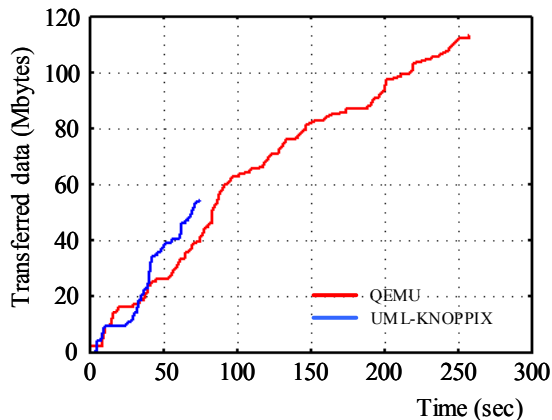


図 15 UML-KNOPPIX と QEMU におけるデータ転送量の比較

計測により、図 15 に示すように QEMU は UML-KNOPPIX と比較してデスクトップマネージャ KDE が起動完了するまでに転送するデータ量は大きく、起動時間も長いことがわかった。UML-KNOPPIX のデータ転送量は QEMU のおよそ半分程度で、cloop による圧縮効果のためだと考えられる。起動時間については、QEMU は CPU エミュレータであることから、内部処理のオーバーヘッドが影響しているものと予想される。

一方、QEMU 起動時におけるスループットの計算結果は図 16 に示すように、最大値は 30Mbps 程度であった。SFS サーバには SCSI ディスクを搭載しており、hdparm によるディスク読み出しのベンチマーク結果は約 65MB/sec であった。hdparm や netperf による測定結果から、スループットのボトルネックはネットワーク環境やサーバのディスクアクセス性能ではなく、それ以外の起動処理に起因していると考えられる。

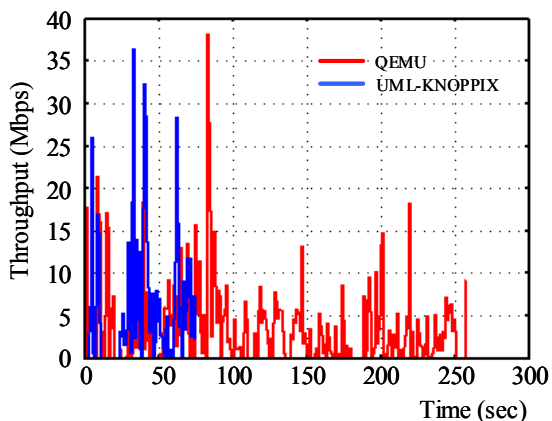


図 16 UML-KNOPPIX と QEMU におけるスループットの比較

7. まとめ

SFS を用いて UML-KNOPPIX を起動するために必要なスループットやデータ転送量について評価した。実験により、起動時には約 30Mbps 程度の帯域が必要であることがわかった。また、CD からの起動では測定が出来なかったデータの読み出し状況などを把握することができた。起動時には cloop による圧縮効果もあって 50MB 程度しかデータを読みこまないことが明らかになり、SFS サーバ上に DVD などの大容量コンテンツを用意してもアクセス上問題ないと考えられる。これにより、従来のように CD イメージ全体を一つのファイルで管理するのではなく、アプリケーション毎などのようにサーバ上で分割した単位で管理することが可能となる。

本研究では SFS サーバ側でパケットをキャプチャしているため、クライアントにおける負荷軽減にもつながる。サーバ側でパケットの転送情報を収集することで、複数の SFS クライアントから要求があった場合の回線負荷やサーバの CPU 負荷などを把握できるため、SFS サーバの負荷分散にも反映できる可能性がある。さらに、クライアントの回線環境に応じて適切なアプリケーションを選択・配布することも考えられる。

また、自治体等で整備が進められている地域高速ネットワーク網における地域 IX において、目的に応じた派生版 KNOPPIX を利用することによる実用性や地域 IX の新たな活用法や、P2P や WEB サービスを利用したアプリケーションの検索と利用などについて今後検討したいと考えている。

参考文献

- [1] Knoppix, “<http://www.knopper.net/knoppix>”
- [2] Knoppix 日本語版,
“<http://unit.aist.go.jp/it/knoppix/>”
- [3] UML, “user-mode-linux.sourceforge.net/”
- [4] 須崎, 飯島, 丹, “UserModeLinux を使った KNOPPIX マイグレーション”, Linux Conference 2003, 2003.
- [5] SFS, “<http://www.fs.net/sfswww>”
- [6] KNOPPIX/Math,
“<http://geom.math.metro-u.ac.jp/wiki/>”
- [7] GIS-Knoppix,
“<http://www.sourcepole.com/sources/software/gis-knoppix/>”
- [8] QEMU, “<http://fabrice.bellard.free.fr/qemu/>”