

SFS を利用した KNOPPIX の起動と処理性能の評価

Evaluation of Booting KNOPPIX with SFS and Performance

後藤和弘[†] 須崎有康[‡] 飯島賢吾[‡] 八木豊志樹[‡] 丹英之[§]

概要 CD ブータブルな Linux ディストリビューションのひとつである KNOPPIX は、手軽に利用できる半面、利用するためにはインターネットから iso イメージをダウンロードした後に CD を作成する必要がある。この方法では更新がある度に CD を作成しなければならない。そこで、インターネット対応ファイルシステム SFS(Self-certifying File System)を利用してインターネットから起動する KNOPPIX を作成した。ここでは実 CPU から起動するもの、Linux エミュレータ (User-Mode Linux, coLinux) や仮想計算機 (QEMU/Win, QEMU/Linux) から起動するものなど 5 種類を作成した。これにより、不特定多数のユーザが自分の利用形態にあった手法で、インターネットを介して、KNOPPIX を手軽に起動可能となった。本論文では、SFS を利用した KNOPPIX の起動方法および処理性能の評価について報告する。

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から、日本語化^[2]をはじめ様々な言語への対応や、教育向け^[3]やバイオインフォマティクス分野^[4]のように特定用途向けのカスタマイズなど、多くの派生版が開発されている。

KNOPPIX は、独自の圧縮ループバックデバイス cloop を用いて 700M CD-ROM に 1.8G 程度のルートファイルシステムを格納している。起動時にはカーネルとブートイメージ (boot.img) を用いて cloop をループバックマウントして利用している。cloop に収められたルートファイルシステムは通常のファイルとして手軽に扱うことができる。この性質を生かし、リモートにある cloop ファイルを利用して KNOPPIX を起動する方式を開発した。

本研究ではインターネット対応ファイルシステム SFS(Self-certifying File System)^[5]を利用して起動する KNOPPIX を作成した。起動方法として実 CPU からの起動や、Linux エミュレータ (User-Mode Linux^[6], coLinux^[7]) や CPU エミュレータ QEMU^[8] (QEMU/Win, QEMU/Linux) からの起動など 5 種類を作成した。本論文では、SFS を利用した起動特性を評価するとともに、その処理性能を比較する。

2. ネットワーク環境での KNOPPIX の起動

ネットワーク上で公開した cloop ファイルをもとに KNOPPIX を起動できれば、いつでも誰でもどこからでも KNOPPIX を利用可能となる。以下ではネットワークを利用した KNOPPIX の起動方法について説明し、インターネット対応ファイルシステム SFS、そして SFS を利用してセキュアな環境で KNOPPIX を起動する方法について説明する。

2.1 KNOPPIX Terminal Server の利用

LAN 環境では KNOPPIX Terminal server を利用して、ネットワーク経由で KNOPPIX を起動できる。KNOPPIX Terminal server は DHCP/TFTP/NFS の各サービスを起動するスクリプトで、カーネルや initrd、サーバの起動 CD 内の cloop ファイルを NFS で共有する。クライアントには PXE ブート可能なコンピュータを使用し、DHCP で IP アドレスを取得した後にサーバ上の cloop ファイルを読み出して KNOPPIX を起動する。しかし、IP アドレスの範囲の制限や、サーバで実行中の派生版 KNOPPIX しか起動できないという制限がある。さらに、通常の FireWall は NFS の通過を許可しないため、不特定多数のユーザを対象として WAN 環境へ適用することは難しい。

2.2 SFS (Self-certifying File System)

SFS はインターネットに対応したセキュアな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

[†] 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.

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, それに続く 32 文字のランダムな英数字はホスト ID と呼ばれる。SFS は図 1 のように NFS をベースとし、サーバ側では SFS サーバデーモン sfssd が NFS クライアントとして動作する。また、クライアント側では SFS クライアント sfscd が NFS サーバとして動作する。このため、SFS サーバ側では NFS におけるアクセス許可を設定する必要がある。

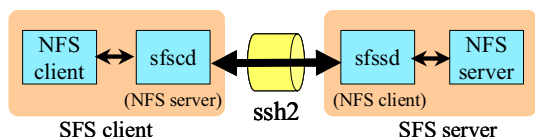


図 1 SFS におけるプログラムの構成

SFS でファイルを公開すれば、SFS サーバでは Anonymous ユーザによるリード・ライトや、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となる。また、クライアントでは NFS のようにルート権限による明示的なマウント操作を必要とせず、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。

2.3 SFS を利用した KNOPPIX の起動

WAN 環境で KNOPPIX を起動するには、SFS のようにセキュアなファイルシステムによって cloop ファイルを公開する必要がある。SFS を利用することで、ローカルコンピュータ上のファイルと同様に、サーバ上の cloop ファイルを読み出しながら KNOPPIX を起動できる。これにより、700MB 近いファイルをあらかじめ一括でダウンロードすることや、ダウンロード後に CD へ書き込む作業は必要ない。本研究では SFS サーバ上の cloop ファイルを利用して KNOPPIX を起動する方法として、以下の 5 種類を開発した。

- 1)CD からの起動
- 2)UML-KNOPPIX の起動
- 3)coLinux による起動
- 4)Linux 上での QEMU による起動
- 5)Windows 上での QEMU による起動

次章以降でそれぞれの方法について説明する。

3. SFS ブート KNOPPIX

3.1 CD 内の cloop ファイルを用いた起動

通常の起動用 CD には図 2 に示すように 1.4MB 程度のフロッピーイメージ boot.img ファイルとルートファイルシステムをもつ 700MB 程度の cloop ファイルが含まれている。起動時はまず boot.img を読み出し、ブートシーケンス linuxrc を実行する。RAM ディスクをルート (/) とし、cloop ファイルをループバックマウントして /usr, /lib など必要なディレクトリについてシンボリックリンクを設定する。書き換えが必要となる/etc などのディレクトリは RAM ディスクに確保される。

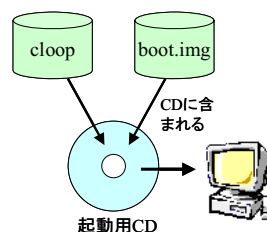


図 2 CD 内の cloop ファイルの利用

3.2 SFS サーバ上の cloop ファイルによる起動

KNOPPIX Terminal server のクライアント機能を改良し、SFS サーバ上の cloop ファイルをループバックマウントして KNOPPIX を起動できるようにした。図 3 は起動時の概念図で、起動用 CD は 2.8MB のフロッピーイメージファイルと SFS クライアントの機能を含んでいる。CD から起動して初期画面のプロンプトで SFS サーバのホスト ID や cloop ファイルを指定すると、DHCP で IP アドレスを取得した後、SFS クライアントデーモンを起動する。そして指定した SFS サーバへ接続し、希望する cloop ファイルをループバックマウントして KNOPPIX を起動する。初期画面で SFS サーバのアドレスを指定しない場合は実験用サーバにアクセスし、図 4 のようにサーバ上のファイル一覧を表示するようにした。これによ

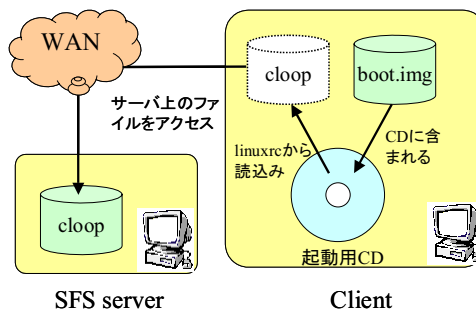


図 3 SFS ブート KNOPPIX の起動

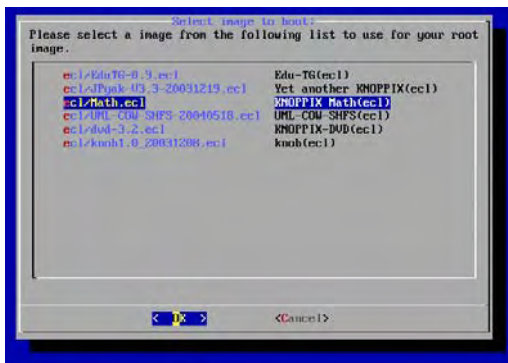


図 4 SFS サーバ上のファイルの選択

り、ユーザは一覧から選択するだけで手軽に KNOPPIX を起動できるようになり、あらかじめ複雑なホスト ID を知る必要はない。以下では、このようにして起動した KNOPPIX を「SFS ブート KNOPPIX」と呼ぶ。

4. SFS を利用した UML-KNOPPIX の起動

4.1 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 氏によって開発されている。UML はオープンソースとして公開されている仮想計算機環境で、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 では、CPU 上で起動する Linux をホスト OS、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。

4.2 UML による KNOPPIX の起動

標準的な UML は cloop のドライバを含まないため、KNOPPIX をゲスト OS として起動できない。そこで、cloop のドライバを含むように UML カーネルを作り直し、KNOPPIX を起動できるようにした。図 5 にローカルコンピュータ上の cloop ファイルを利用した UML の起動の概要図を示す。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。以下では、このように UML によってゲスト OS として起動した KNOPPIX を「UML-KNOPPIX」と呼ぶ。

UML-KNOPPIX では、起動時に umlhub や IP マスカレードで仮想ネットワークを設定することで、ホスト OS や他の UML-KNOPPIX 間で通信できるようにした。また、UML は仮

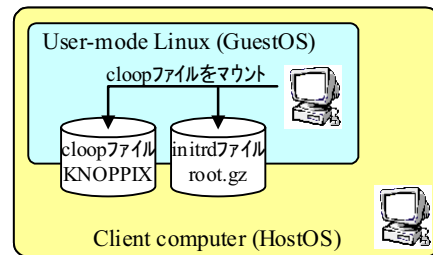


図 5 ローカルコンピュータ上の cloop ファイルによる UML-KNOPPIX の起動



図 6 UML-KNOPPIX の起動例

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 を利用して、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 サーバとなるため、図 6 のように 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。また、起動オプションにより、デスクトップマネージャや、画面サイズ、起動に用いる cloop ファイルを選択できるようにした。

4.3 SFS サーバ上の cloop ファイルの利用

SFS を利用すると、ホスト OS はサーバ上の cloop ファイルをローカルファイルと同様にループバックマウントして UML-KNOPPIX を起動できる。SFS ブート KNOPPIX と比較すると、複数の派生版 KNOPPIX の動作確認をおこなう場合などにコンピュータ本体を再起動する必要がないというメリットがある。また、複数の派生版 KNOPPIX を 1 台のクライアント・コンピュータ上で同時に起動できるため、複数の UML-KNOPPIX を利用して、ネットワークアプリケーションの開発や動作確認を行なうことも可能となった。サーバ上の cloop ファイルを利用した UML-KNOPPIX 起動の概念を図 7 に示す。

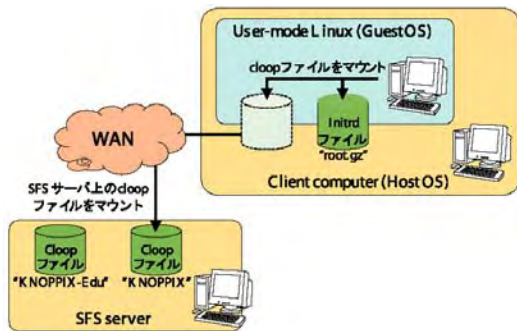


図7 SFS サーバ上の cloop ファイルをマウントした、WAN 環境での UML-KNOPPIX の起動

5. SFS を利用した coLinux の起動

5.1 coLinux (Cooperative Linux)

coLinux は Windows 用の Linux エミュレータで、Dan Aloni 氏を中心にオープンソースとして開発されている。起動する Linux のルートファイルシステムなど、必要なパラメータを XML ファイルに記述するため、編集も容易である。Cygwin/X がインストールされていれば Xnest によって KDE などの X 環境を利用できる。また、Windows の TAP インタフェースを利用して他の端末との通信が可能となる。

5.2 SFS を利用した KNOPPIX の起動

SFS クライアント機能の Windows 版は開発されていないため、通常は SFS サーバ上のファイルにアクセスできない。そこで、coLinux 上で SFS サーバ上の cloop ファイルを読み出して KNOPPIX を起動するために SFS ブート KNOPPIX の iso ファイルを利用する。XML ファイルに iso ファイル名や SFS サーバのホスト名、ホスト ID、サーバ上の cloop ファイルなどを記述し、起動用のバッチファイルを実行する。これにより、SFS ブート KNOPPIX と同様に、サーバ上に用意した派生版 KNOPPIX を Windows 上で起動可能となる。

6. SFS を利用した QEMU の起動

QEMU は x86 などの CPU 上で、x86 や ARM、SPARC、PowerPC などの CPU をエミュレートする。実行時に"-cdrom"オプションで CD 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。

KNOPPIX 日本語版 (v3.4 20040517 - 20040820) には Linux 版と Windows 版の QEMU を含んでいる。Linux 版 QEMU を実行するため、CD ブートした KNOPPIX 上で /dev/cdrom をディスクイメージとして仮想計

算機環境を起動するスクリプトやネットワーク利用のためのスクリプトを作成した。SFS サーバ上の cloop ファイルを指定して起動することもできる。一方、Windows 環境についてはフロッピーイメージファイル (qemu-boot.img) を用意するとともに、Linux 版と同様に CD をディスクイメージとして Windows 版 QEMU を実行可能なバッチファイルを作成した。起動した仮想計算機環境と他のネットワーク端末が通信するために、Linux 版では TUN/TAP インタフェースを、Windows 版では user mode network をそれぞれ利用する。

7. 評価実験

7.1 実験方法

実験環境として、SFS サーバ、および SFS クライアントを 100Mbps のスイッチングハブへ接続する。SFS サーバには Pentium4 2.66GHz、メモリ 512MB のコンピュータを使い、KNOPPIX (v3.4 20040510-20040520 版) 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 で追加する。また、サーバ上で公開する cloop ファイルは、KNOPPIX v3.4 20040510-20040520 版とした。クライアントには PentiumM 1.0GHz、メモリ 512MB、24 倍速 CD ドライブを有するコンピュータを使い、KNOPPIX (v3.4 20040517-20040820 版) を CD から起動する。

開発したそれぞれの方法について、サーバ上の cloop ファイルを利用して KNOPPIX を起動し、サーバ・クライアント間で転送されるパケットをサーバ側で tcpdump によりキャプチャする。tcpdump を使うことで必要なパケットだけをフィルタリングするとともに、パケット送出時刻をもとに起動に要する時間を正確に測定できる。X の起動が完了した後にキャプチャを停止し、tcpstat でデータ転送量や起動時間を 1 秒間隔のサンプリングで計算する。また、計測はそれぞれの条件について 3 回ずつ実施する。予備実験として実験環境の転送能力を netperf で計測するとともに tcpdump でパケットをキャプチャしたところ、最高で 93.3Mbps で転送した場合でもパケットのドロップは 0 であり、本手法で正確にキャプチャできることを確認した。

7.2 SFS ブート KNOPPIX

クライアントコンピュータを CD から起動し、初期画面のプロンプトにおいて sfshost パ

ラメータで SFS サーバのアドレスとホスト ID を指定し, sfsfile パラメータでサーバ上の cloop ファイルを指定して KNOPPIX を起動する. SFS ブート KNOPPIX の起動では DHCP による IP アドレス取得を基準とするために, クライアントの MAC アドレスを tcpdump のキャプチャフィルタとする.

7.3 SFS を利用した UML-KNOPPIX の起動

UML-KNOPPIX の起動特性を評価するため, クライアントでホスト OS を起動した後, サーバ上の cloop ファイルを指定してゲスト OS を起動する. tcpdump のフィルタには SFS が通信に利用する SSH2 のポート番号を指定しする. また, SFS サーバのカーネルバージョンによる影響を確認するため, サーバのカーネルを 2.4, および 2.6 で起動した場合についてそれぞれ計測する.

7.4 仮想計算機環境の性能評価

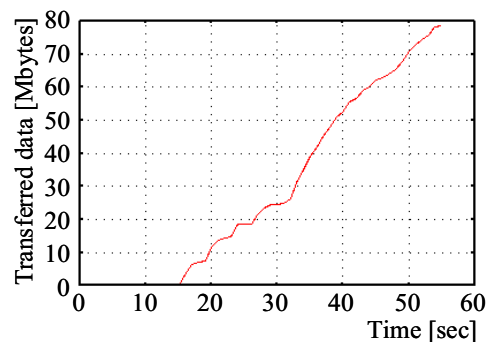
UML-KNOPPIX の性能を評価するために, 同様の仮想計算機環境である CPU エミュレータ QEMU, Linux エミュレータ coLinux について, CPU の処理性能, ネットワークの転送性能, X サーバの描画性能を比較する. CPU 処理の性能計測には nbench^[9], ネットワーク性能の計測には netperf^[10], 描画性能の計測には x11perf をそれぞれ利用する. このため, UML-KNOPPIX, 通常の CD ブート KNOPPIX, QEMU/Linux, QEMU/Win, coLinux についてそれぞれ計測をおこない, UML-KNOPPIX の結果と他の結果を比較する.

8. SFS ブート KNOPPIX の評価

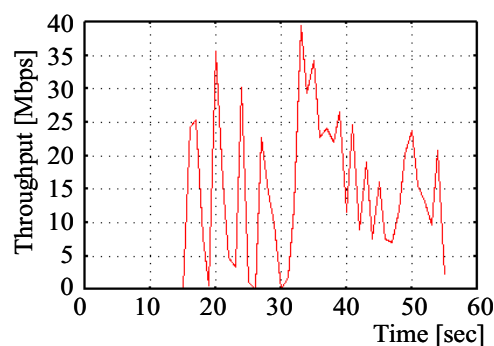
SFS ブート KNOPPIX を起動した場合における起動特性の計算結果の一例を図 8 に示す.

8.1 起動時間とデータ転送量

図 8(a)は, KNOPPIX の起動時に SFS サーバからクライアントへ転送されたパケットのデータ量を表していて, KDE が起動完了するまでのデータ転送量は 78MB であった. 700MB 近い cloop ファイル全体をダウンロードしないため, ネットワークや SFS サーバの負荷軽減につながる. 一方, 横軸は DHCP による IP アドレス取得時を基準とした経過時間で, IP アドレスを取得した後, 約 15 秒間経過してから SFS サーバへ接続要求パケットを送信していた. 起動時には SFS クライアントデ



(a) データ転送量



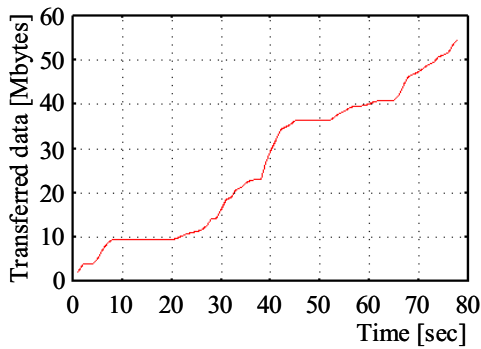
(b) スループット

図 8 SFS ブート KNOPPIX の起動特性

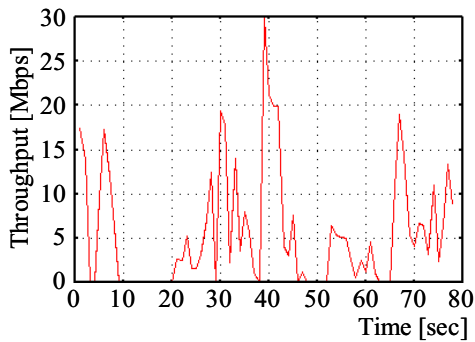
ーモンの起動や cloop デバイスドライバのインストール処理などがあり, 実行時間やウェイト時間に起因すると考えられる. SFS サーバへの接続から, KDE の起動完了までの時間は約 40 秒で, CD 内の cloop ファイルから起動する場合 (約 180 秒) と比べ, 提案する方法では高速に起動できることを確認した.

8.2 スループット

起動時におけるスループットは, 図 8(b)に示すように起動シーケンスのあいだにかなり変動していた. このときの最大スループットは 40Mbps 程度であり, NIC の最大性能には達していない. このことから, 起動時間を短縮するには伝送路や NIC などネットワーク環境の改善よりも, 起動処理の見直しなどの対策が有効であると考えられる. CD から起動する場合はドライブの読み込み速度が起動時間に影響を与えるが, 最近のコンピュータで使われる NIC の多くは 100Mbps の能力を有するため, 提案する手法ではほとんどのコンピュータにおいてハードウェアによる影響は少ないと予想できる. スループットが最大となるのは短時間のため, 高速 ADSL 程度の能力があれば WAN 環境で利用できると考えられる.

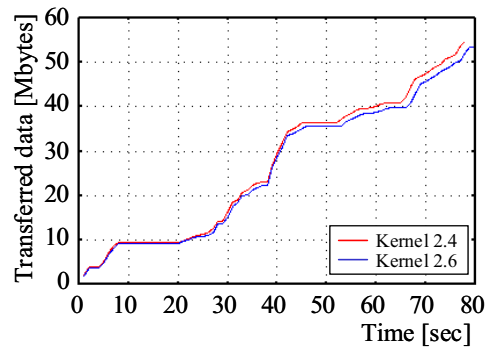


(a) データ転送量

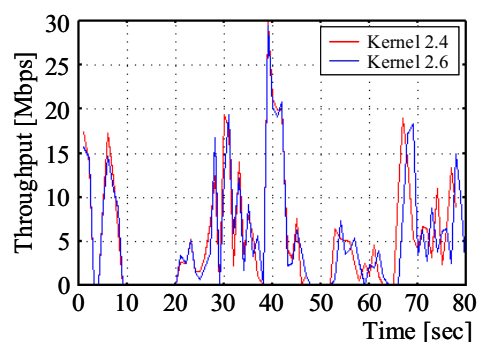


(b) スループット

図9 UML-KNOPPIX の起動特性



(a) データ転送量



(b) スループット

図10 SFS サーバのカーネルによる影響

9. UML-KNOPPIX の起動に関する評価

SFS を利用して UML-KNOPPIX を起動した場合における、起動特性の計算結果の一例を図9, 図10 に示す。

9.1 起動時間とデータ転送量

UML-KNOPPIX の起動時に転送されたデータ量を図9(a)に示す。指定した cloop ファイルは SFS ブート KNOPPIX と同一であるが、データ転送量は 55MB で SFS ブート KNOPPIX よりも少なく、起動時間は 80 秒間程度で SFS ブート KNOPPIX よりも長い。これはハードウェア自動検出機能の hwsetup が一因として挙げられる。図において、起動開始後 10 秒のあたりから約 10 秒間の区間で hwsetup が実行されていて、この処理時間は SFS ブート KNOPPIX の起動時よりも長い。この他にも起動シーケンスにおける処理タイミングなどが起動時間に影響を与える。また、データ転送量の違いについては、起動に必要なライブラリファイルや X の起動に関するファイルの転送量が異なるためと考えられる。

9.2 スループット

図9(b)に示すように、UML-KNOPPIX 起動

時のスループットは起動シーケンスのあいだにかなり変動があり、最大スループットは 30Mbps 程度であった。また、hwsetup が実行される区間や幾つかの区間でスループットが低い箇所が存在している。起動時間を短縮するには、この区間で実行されている処理内容やウェイト処理などを詳細に解析し、改善する必要がある。

9.3 SFS サーバのカーネル

KNOPPIX は v3.4 からカーネル 2.6 にも対応していて、起動時に 2.4 または 2.6 のいずれかを選択できる。そこで、SFS サーバのカーネルの違いが UML-KNOPPIX の起動に与える影響について確認した。

図10(a)は UML-KNOPPIX 起動時のデータ転送量、図10(b)はスループットの計算結果の一例である。この結果から、起動時間、スループットともに、カーネルの違いによる差はほとんどないことがわかる。カーネル 2.6 ではスレッドや I/O に関する処理が改善されているものの、サーバ上の cloop ファイルを利用して UML-KNOPPIX を起動する場合には、カーネルバージョンによる影響は少ないことが明らかになった。

10. 仮想計算機環境の性能評価

UML-KNOPPIX の性能を評価するため、CPU 処理性能やネットワーク性能、描画性能を QEMU や coLinux と比較した。

10.1 CPU 等の処理性能

図 11 に nbench を用いて CPU, FPU, メモリの処理速度を計測した結果を示す。nbench は 10 種類の項目について計測をおこない、実行結果をもとに評価の指標となるインデックス値を計算する。図から、ネイティブな KNOPPIX と UML-KNOPPIX ではほとんど差がないのに対して、QEMU は Linux 版、Windows 版のどちらも UML-KNOPPIX よりもかなり低いことがわかる。QEMU は CPU エミュレータであるため、処理におけるオーバーヘッドが影響する。また、coLinux はネイティブな KNOPPIX や UML-KNOPPIX と同程度の値であり、Windows 上でネイティブ KNOPPIX 相当の性能が得られることを確認した。

10.2 ネットワークの転送性能

それぞれの Linux エミュレーション環境のターミナルで netperf を実行し、スイッチングハブで接続した他のコンピュータとのあいだでスループットを計測した結果を図 12 に示す。UML-KNOPPIX の転送性能はネイティブな KNOPPIX より劣るものの、QEMU や coLinux よりも高い結果であった。QEMU では CPU エ

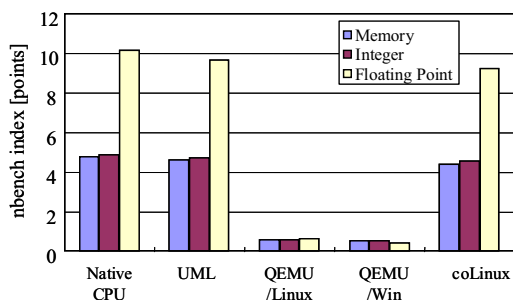


図 11 CPU 等の処理性能の比較

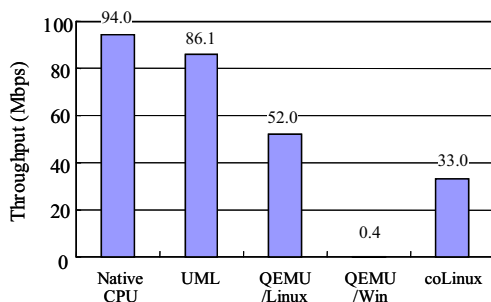


図 12 ネットワークの転送性能の比較

ミュレーションによるオーバーヘッドが、coLinux では Windows のドライバによるオーバーヘッドがそれぞれ転送性能に影響を与える。また、Windows 版 QEMU は user mode network を利用することが、TUN/TAP を利用する他の環境よりも低い数値となった一因と考えられる。一方、UML-KNOPPIX は TUN/TAP を利用し、UML 上のプロセスはホスト OS 上のプロセスとして実行されるために、他の環境より高いスループットが得られると考えられる。

10.3 X サーバの描画性能

x11perf には多くのテスト項目があるため、それらの実行結果を要約するシェルスクリプト Xmark によって性能を評価した。ただし、coLinux については x11perf の実行途中でエラーとなったため比較対象から除外した。図 13 に Xmark の計算結果を示す。UML-KNOPPIX は CPU やネットワーク性能はネイティブな KNOPPIX と同程度だが、描画性能はかなり低い。一方、QEMU はエミュレーションのオーバーヘッドに加え、CPU やネットワーク性能の面でネイティブな KNOPPIX よりも劣っていることが影響していると考えられる。

coLinux は x11perf のテスト項目の一部で実行エラーとなり、描画性能を他の仮想計算機環境と比較できなかったため、ウィンドウや文字など基本的な描画に関係すると思われる項目に限定して描画性能を比較した。表 1 に実験で使用した起動オプションを示す。

表 1 x11perf で指定した起動オプション

オプション	テスト内容
rect500	500x500 の長方形塗つぶし
fl4text16	40 桁の行へ 2-byte 文字表示
copypixwin500	Pixmap からウィンドウへ 500x500 の正方形をコピー
putimage500	500x500 の正方形の Putimage

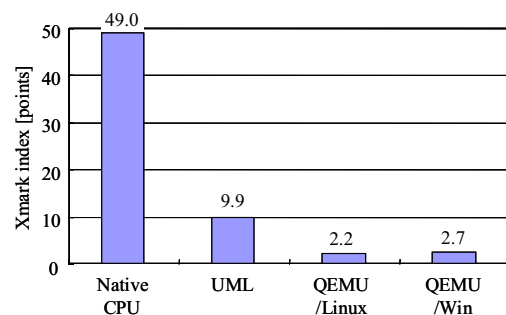


図 13 Xmark による描画性能の比較

各オプションにおける実行結果を図 14 に示す。coLinux の計測結果は QEMU 同様にネイティブな KNOPPIX よりもかなり低い数値であった。UML-KNOPPIX と coLinux はいずれも Xnest を利用してゲスト OS の描画をおこなっているが、その性能には 7 倍~33 倍の差がある。これは、coLinux では X 環境に Cygwin/X を使うことや X プロトコルの転送に必要なネットワークの転送性能が低いことなどに起因すると考えられる。

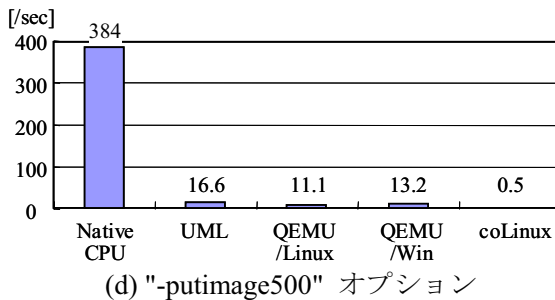
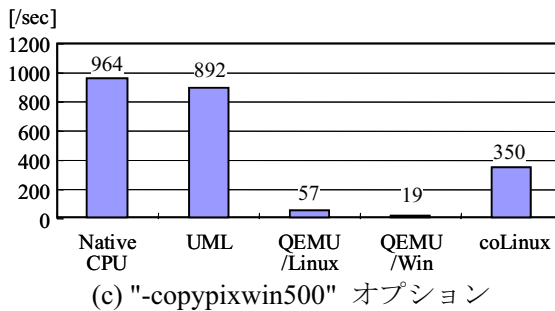
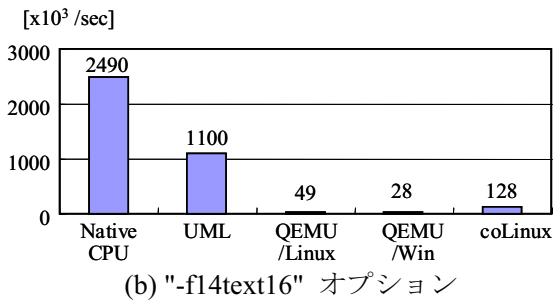
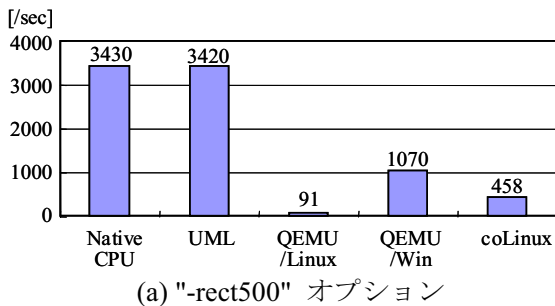


図 14 基本的な描画命令に関する性能比較

11. まとめ

SFS を用いて WAN 環境で KNOPPIX を起動する方法を提案し、その起動特性について評価した。また、UML-KNOPPIX の性能を評価するため、CPU エミュレータ QEMU や Linux エミュレータ coLinux と処理性能やネットワーク性能、描画性能を比較した。そして、描画性能に関する一部の試験項目を除き、ほとんどの項目でネイティブな KNOPPIX と大きな差が無いことを確認した。

今後は、複数のクライアントから SFS サーバへアクセスしたときのサーバの負荷などについて検討するとともに、大分県が整備している高速広帯域ネットワーク網である「豊の国ハイパーネットワーク」を利用した地域 IX などにおいて実証実験をおこないたいと考えている。

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] Knoppix-Edu, "<http://www.eng.tohoku-gakuin.ac.jp/knoppix/>"
- [4] Knoppix for Bio (KNOB), "<http://knob.sourceforge.jp/>"
- [5] SFS, "<http://www.fs.net/sfswww>"
- [6] User-Mode Linux, "user-mode-linux.sourceforge.net/"
- [7] coLinux, "<http://www.colinux.org/>"
- [8] QEMU, "<http://fabrice.bellard.free.fr/qemu/>"
- [9] nbench, "<http://www.tux.org/~mayer/linux/bmark.html>"
- [10] netperf, "<http://www.netperf.org/>"