

性能評価のためのホストエミュレーションの提案と評価

2004MT003 浅野 洋介
指導教員

2004MT038 加藤 史章
後藤 邦夫

1 はじめに

本研究の目的は、仮想ネットワーク上に配置されたアプリケーションの性能評価を行うために複数のネットワークホストの構築を実現することである。これを実現するにはネットワークスタックの仮想化が望ましい。近年、IMUNES[2] や Linux-ns[4] というソフトウェアの開発が行われている。この開発により1台のPC内で複数のネットワークホストを構築することが期待されている。しかしこれらのソフトウェアは膨大な量のカーネル修正が必要であるため利用が非常に困難である。そこで、「仮想ソフトウェア」を用いて1台のPCで複数のホストをエミュレートする方法を提案する。仮想ソフトウェアとは、複数のOSを1台のPC上で同時に動作させるソフトウェアである。一般的な利用目的はソフトウェアやカーネル開発である。利用目的は異なるが、本研究ではこれらを使用することにより、ホスト全体をエミュレートする方法を提案する。

現在、多数の仮想ソフトウェアが存在し、それぞれ性能や導入の容易さに大きな違いがある。そこでQEMU[1], KVM[5], VMware Player[7], Xen[8], User-mode Linux[6] の5種類を選択し、それらの性能評価を行い、本研究に合うソフトウェアを選定する。それらと既存のネットワークエミュレータであるGoto's IP Network Emulator (以下, GINE)[3] と組み合わせていくつかのネットワーク利用モデルについて性能評価を行った。

仮想ソフトウェアの評価を共同で行い, GINEを含めたネットワークモデルの構築を浅野が, エミュレーションを行うプログラムの作成を加藤が担当した。

2 エミュレーションの目的

ネットワークを構成する際、通常であればネットワークに接続されたホスト1台に対して1台のPCを用いることが多い。本研究では仮想ソフトウェアを用いることで、ホスト数は最小限のPC内で全てのホストを実装できると考えた。

ネットワークアプリケーションにはWWW, Mobile IPv6, P2P などさまざまなものがある。これらをエミュレートするためには、サーバとクライアント間のスループットや遅延などの状態を設定する必要がある。これらを設定するためにネットワークエミュレータを用いる。

2.1 ネットワークスタックの仮想化

ネットワークスタックとは、ネットワークの構造を抽象化したOSI参照モデルのようにネットワーク階層がつまれたものをいう。これを仮想化することによって1

台のPC上に複数のネットワークを構築できる。現在、ネットワークスタックの仮想化を中心に開発している企業や研究機関が多いが、これらは大幅なカーネル修正を必要としているため、非常に困難である。よって本研究ではカーネル修正を行わず、比較的容易に使用できる「仮想ソフトウェア」を用いる。仮想ソフトウェアを用いることで、ネットワークスタックを含めたホスト全体をエミュレートする。

3 仮想ソフトウェアのネットワーク構成法

仮想ソフトウェアのネットワーク構成法について述べる。実機のPC上にインストールされているOSをホストOSと呼ぶ。そのホストOS上に仮想ソフトウェアをインストールし、仮想的なハードウェアを作成する。その上にインストールしたOSをゲストOSと呼ぶ。

仮想ソフトウェア上で動作するOSのイメージファイルはVine Linux 4.1から必要なコマンドやライブラリ等をコピーして作成した。

QEMU, KVM, VMware Player

図1にホストOS, ゲストOS間の接続例を示す。

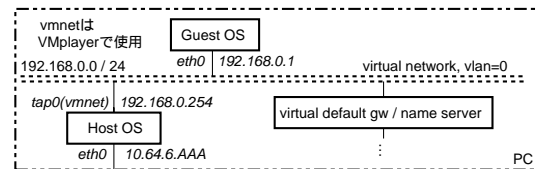


図1 ホストOS・ゲストOS間の接続例

ホストOSのtapデバイスとゲストOSのEthernetデバイスの生成時にそれぞれ同じネットワーク(vlan)番号を設定する。そして同じネットワークアドレスのIPアドレスを設定することで、両ホストがお互いに通信可能となる。なおゲストOSから外部への通信はできない。

KVMは付属のQEMUを使用しているので図1, 図2と同じ構成である。VMware Player(以下, VM-player)は図1と似ているが, vmnet デバイスを用い, NAT 接続方式を使用しているため, ゲストOSは外部ネットワークと通信可能である。

図2にゲストOS・ゲストOS間の接続例を示す。

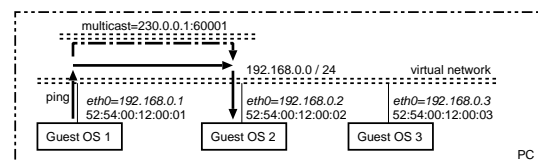


図2 ゲストOS・ゲストOS間の接続例

マルチキャスト設定をすると、その設定をしたゲスト OS 全てが同じネットワーク空間に属していることになる。これを設定する場合、ゲスト OS の eth0 の MAC アドレスを全て異なるものにする必要がある。

Xen

Xen の Bridge 接続方式を以下に示す。

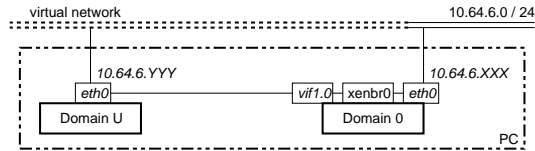


図3 Xenのネットワーク構成(Bridge)

実際の通信は Domain 0(Host OS)の仮想デバイスと Domain U(Guest OS)の仮想 Ethernet デバイスが通信しているように見せかけている。

User-mode Linux

User-mode Linux(以下、UML)の TUN/TAP, switch daemon 接続方式はそれぞれ図1, 図2と同じ構成である。

4 仮想ソフトウェアの性能評価

本節では 4.1 で述べる性能評価の基準に則り、仮想ソフトウェアを選定する。

4.1 性能評価基準

本研究ではホスト OS, ゲスト OS は, Vine Linux 4.1 (2.6.16 系) (Xen のスループット測定のみ Fedora Core 7) を使用した。また性能評価で使用した PC は

PowerEdge SC440
Intel^(R) Xeon^(R) CPU 1.86GHz (Dual Core)
クロック周波数 1862.063 MHz
2GB DDR2-SDRAM PC2-4200 ECC 対応
スワップ領域 6GB

である。

1. ホスト OS とゲスト OS 間のスループットが高い
2. ゲスト OS が複数起動できる
3. ゲスト OS の使用メモリ量が小さい
4. GUI 起動が不要である
5. 設定が容易である

特に 1~3 を重点に置き選定する。

4.2 ホスト OS とゲスト OS 間のスループット測定

TCP を測定する iperf, tbench を用いた。

スループット測定方法

- ホスト OS をサーバモード, ゲスト OS をクライアントモードに設定する
- 30.0 秒の計測とする
- サーバに同時アクセスするクライアント数を 100 とする
- 以上の条件で 10 回測定し, その平均値をとる

測定結果を表 1 に示す。

表 1 スループット測定結果

	iperf(kbps)	tbench(kbps)
QEMU	114,052.8	100,055.0
KVM	83,855.0	65,451.9
VMplayer	572,787.4	541,347.2
Xen	290,242.2	482,980.5
UML	376,424.0	235,423.3

- VMplayer, Xen, UML は高いスループットを得られた。
- QEMU は 100BASE-TX のデバイスを使用した が, 100MB 以上のスループットが確認できた。

4.3 メモリの使用量

仮想ソフトウェアの起動によるメモリの使用量を測定する。なお全てのソフトウェアは, 起動時にメモリの使用量を設定できる。測定は以下の方法で行った。

- メモリ量を測定できる top コマンドを使用

測定結果を表 2 に示す。

表 2 メモリ使用量の測定結果

	設定メモリ (MB)			
	16	32	64	128
QEMU	38.4	55.3	64.5	67.1
KVM	21.0	34.8	37.4	40.0
VMplayer	起動せず	86.0	115.2	127.0
Xen	起動せず	31.7	70.7	130.6
UML	14.3	14.3	14.3	14.3

- 測定された値はゲスト OS 上で認識されるメモリ量とソフトウェア自体のメモリ量の合算である。
- VMplayer, Xen は使用メモリを 16MB に設定した場合, メモリ不足でソフトウェアが起動しなかった。使用メモリを上げた場合, 起動はしたものの使用メモリが大きくなってしまった。
- UML は設定する使用量を変えても, 同じ使用メモリであった。

4.4 ゲスト OS の起動可能数

ゲスト OS が最大何個まで起動可能かを調べた。結果を表 3 に示す。

表 3 ゲスト OS の起動可能数

	32(MB)
QEMU	34
KVM	37
VMplayer	1
Xen	1
UML	35

(単位:個)

- VMplayer はソフトウェアの仕様により 1 つしか起動できない。

- Vine Linux では、Xen は 1 つしか起動できなかった。
- QEMU は 35 個目を、KVM は 38 個目を起動したとき、メモリ不足のエラーが出力され、起動ができなかった。
- UML は 36 個目を起動したとき、起動中にソフトウェアが強制的に終了した。
- ゲスト OS の最大起動可能数は実メモリに依存される。メモリを増設することで起動数を増やすことが可能と思われる。

4.5 GUI について

ゲスト OS の起動時に Host OS 側の GUI 環境が必要か、またゲスト OS 側での GUI なしでの起動が可能かを調べた。なおこの結果は表 4 に示す。

- Host OS 側の GUI 環境が必須であるソフトウェアは VMplayer のみである。その他のソフトウェアは端末上でも起動できる。
- ゲスト OS の GUI 起動は UML のみできない。

4.6 設定の容易さ

インストールの工程が容易か、ならびに経路設定などのネットワーク設定が自動または手動でも設定することが容易かという基準でソフトウェアを選定する。なおこの結果は表 4 に示す。

- VMplayer のネットワーク設定は付属ツールを使用することで容易に行うことができた。
- QEMU, KVM はゲスト OS 起動時にネットワークをオプションとして設定可能であり容易であったが、複雑なシステム構築には不向きと思われる。
- UML は既存のファイルシステムを用いて実行させることは容易だが、ネットワーク設定のために実行するコマンドが多くなる。
- Xen は、Domain 0, Domain U とともに Fedora Core では容易に動作するが、Vine Linux では Domain U の作成が困難であった。

4.7 総評

結果を表 4 に示す。

表 4 性能の比較表

	tput	mem	multi	GUI	conf
QEMU					
KVM	x				
VMplayer			x		
Xen			x		x
UML					

評価がよいものを , 以下, , x とした。なお同じマークのものは、ほぼ同程度の評価と判断した。

- 表 4 より、すべてのソフトウェアはそれぞれ 2 つ以上の項目で評価が高かった。

- 本研究では高スループットが必要であるため、測定結果が低かった KVM は不向きと思われる。
- VMplayer のスループットは 5 つのソフトウェアの中で一番高く評価できる。しかしゲスト OS を 1 つしか起動ができない点や、Host OS 側で GUI 機能が必須ということでメモリ使用量が高くなるという点から考えると、本研究には不向きと思われる。
- UML はどのメモリ使用量にしても同じである点が評価できる。

結論

以上より、Host OS, ゲスト OS とともに Linux OS の場合は UML または QEMU, Host OS, ゲスト OS とともに Linux OS 以外の OS を使用する場合に QEMU を使用することにする。

5 端末情報の自動取得システム

本研究では家電機器等の端末情報を電話回線やインターネット回線等を用いて自動取得するシステムをエミュレーションする。

5.1 提案するシステム

例として図 4 に QEMU と GINE を用いたモデル図を示す。

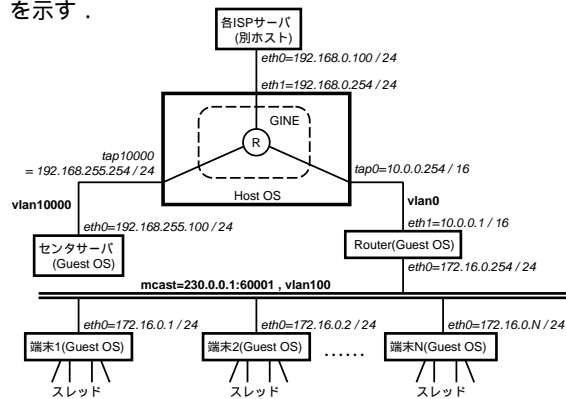


図 4 QEMU を用いた情報取得システム

QEMU の最大起動数は 34 個ということでセンタサーバに 1 つ、ルータに 1 つ使用するため、ホストの最大起動数は 32 個となる。なお、UML を使用した場合、TUN/TAP と switch daemon を使用する。

5.2 システム仕様

- 本研究では、ある将来の家電端末利用予測数より、端末が合計 200 万、ISP サーバが 10 箇所という想定をする。よって 1 台の ISP サーバで受け持つ端末は 20 万台となる。
- センタサーバと各家庭にある端末をゲスト OS として、ISP サーバを別ホストとしてエミュレートすることによってシステム構築する。
- センタサーバから命令を受けた ISP サーバが各端末へ通信し、その結果を ISP サーバへ送信、最

最終的にセンタサーバに転送する。

- 各端末はセンタサーバに登録された装置 ID, ISP サーバと共有する秘密鍵, 自分の秘密鍵をもつ。端末, ISP サーバは上記の鍵を用い, XOR(排他的論理和) により双方を認証する。
- 端末数を多くするために, プログラムによって 1 仮想ホスト内で複数のスレッドを使用する。
- 端末は省電力を想定して間欠起動となる。起動時間は 5 秒, 休止時間はスレッドごとで異なり, 30 秒から 60 秒のランダム秒とする。
- GINE を用いてセンタサーバと ISP サーバ間, ISP サーバと各端末間で遅延, パケットロスを起こすことで実ネットワークに近いエミュレーションを行う。

5.3 エミュレーション結果

ゲスト OS 起動時のメモリ使用量を 32MB とした場合, QEMU で 103 スレッド, User-mode Linux で 211 スレッド動作させることができた。そこで QEMU は端末 100 個, User-mode Linux は 200 個を 1 つのホストで模倣することにした。

また, ISP サーバプログラムは一度の起動で端末 1531 個までしかすることができなかった。そこで, 1500 個の端末リストを持つ ISP サーバプログラムを複数起動させ, それぞれが 200 個の端末と同時に通信を行う設定とした。さらに遅延は実環境に近い 10 ミリ秒として実験を行った。

表 5 に結果の一部を示す。

表 5 実験結果

	通信失敗率	所要時間	最大情報取得数
QEMU	2/3200	257 秒	3198
UML	23/3200	291 秒	6193

- QEMU に比べ, UML のほうが情報取得を失敗する確率が高い。また, 取得の所要時間も QEMU により UML のほうが時間がかかった。
- 端末スレッドは QEMU は 3200 個, User-mode Linux では 6400 個起動しているが, 通信の失敗によりそれより少数しか取得できなかった。端末の休止時間を短くした場合, 通信失敗が増えることから仮想ルータのメモリが足りず, その処理能力以上のパケットが流れ続けていると考えられる。

以上より, 本研究でエミュレートしたシステムでは, UML を用いたほうが同時に多くの端末が存在するシステムのエミュレーションを行うことができた。しかし, 通信を失敗する確率は UML のほうが高い。また, 情報取得の所要時間も UML のほうがかかるため, ネットワーク強度は QEMU のほうが高いと考えられる。

6 おわりに

本研究を通して, 5 つのソフトウェアの性能評価を行った結果, ソフトウェア毎にそれぞれ異なる特徴を持ち, 利用目的によって使い分ける必要があることが分かった。そしてそれらを用いてネットワークモデルを構築し, 実ネットワーク上で実装する前に仮想ネットワーク空間での試用が可能となった。今後の課題を述べる。

1. 本研究ではカーネル修正が不要なソフトウェアを使用したネットワークスタックのみの仮想化を実現するソフトウェアを使用する。
2. プログラムがオープンソースで提供されている仮想ソフトウェアはプログラムを修正することで, スループットの向上が可能と思われる。
3. 本研究で提案したシステムでは, ISP サーバが一度に 1500 個の端末に対してしか動作しなかったため, 20 万台のエミュレーションを行うためにはプログラムの改良が必要である。
4. 近年では, 無線 LAN 等のワイヤレス機能による情報取得が行われているため, 有線 LAN 以上に遅延やパケットロス等の障害が発生する可能性がある。よって今後はそれらに対応するために, GINE を改良し, 検証と実験を重ねる必要がある。

参考文献

- [1] Fabrice Bellard: *QEMU Emulator User Documentation* (accessed Jul. 2007). <http://fabrice.bellard.free.fr/qemu/user-doc.html>.
- [2] FreeBSD Project: *The FreeBSD Network Stack Virtualization Project* (accessed Sep. 2007). <http://imunes.tel.fer.hr/virtnet/>.
- [3] Ihara, A., Murase, S., and Goto, K.: IPv4/v6 Network Emulator using Divert Socket, *Proc. of 18th International Conference on Systems Engineering(ICSE2006)*, Coventry,UK, pp. 159-166 (Sep.2006).
- [4] OpenVZ Team: *Linux-2.6-ns* (accessed Dec. 2007). <http://git.openvz.org/?p=linux-2.6-netns;a=summary>.
- [5] Qumranet: *KVM White Paper* (accessed Jul. 2007). http://www.qumranet.com/wp/kvm_wp.pdf.
- [6] SOURCEFORGE.net: *The User-mode Linux Kernel Home Page* (accessed Jul. 2007). <http://user-mode-linux.sourceforge.net/>.
- [7] VMware, Inc: *VMware Player Manual* (accessed Jul. 2007). http://www.vmware.com/pdf/vmware_player200.pdf.
- [8] XenSource: *Xen Users' Manual* (accessed Jul. 2007). <http://bits.xensource.com/Xen/docs/user.pdf>.