OS 授業向けマルチユーザ VM 環境の構築

平 良 太 貴[†1]　河　野　真　治[†2]

情報系の学生に限らず、Web サービスを支える仮想マシン技術に習熟することは重要である。本論文では学生が学ぶのに必要な仮想マシン環境の仕様について考察し、一つの方法として、KVM と libvirt の CLI である virsh の wrapper を用いる方法を提案し評価する。これにより、LDAP で管理された学生のアカウントに対して、適切な VM 管理権限 (VM の作成、起動、停止、削除) の提供を可能にすることができた。また、PC 上の仮想マシン、例えば Virtual Box などとの連携も可能である。Vagrant などの比較的簡単に VM を作ることができる環境に関しても考察する。この場合は、Vagrant の Box のセキュリティが問題となる。実際の授業で用いて、学生に VM を用いたサービスの運用の実際について学んでもらった。

Suggest multi user VM environment for OS class

Taiki TAIRA[†1] and Shinji KONO [†2]

Learning virutal machine technology is important in Web service construction, not only for information engineering studuents. In this study we investigate necessary specification of VM environment in schools. We use a wrapper for libvirt CLI, virsh on KVM. This provides necessary capability such as VM image creation, start, stop and deletion for LDAP based student account. It also provides a linkage with VM on PC, such as Virtual Box. We also consider easy UI like Vagrant. In this case, the security of Vargrand box matters. In our lecture, this system is used by many students to study actual management of VM based services.

1. VM を用いた Web サービスの教育

Web サービスが IT 技術のひとつとして広まり、Web フレームワークなどの普及で学生でも実装が容易になってきている。そのため、学生でも Web サービスを開発する技術は必須なものとなっている。学生が個人で Web サービスを開発する場合でも、 Web サーバを手持ちの PC で構築し、外部の VPS やクラウド等へデプロイするワークフローを学ぶ必要がある。しかし外部のサービスを利用する場合に、学生の VM の管理を十分に行うことが難しく、コストも困難である。クラウド上で学生が使う VM の管理方法は、クラウドの運営者と協調して API などを通して実現する必要がある。ここでは、学校にあるサーバー機器 (ブレードなど) を用いて、 Web サービスを学生が構築、開発し、運用する方法の実装を行った。

実際、本学の情報工学科では Operating System という授業を提供している。この授業では OS について学ぶ一環として、VM について学習し、課題を提出させる。課題では VM の環境を学生が設定し、情報工学科の持つブレードサーバ上にアップロードし、プログラムの実装や計測を行う。これを教師あるいはシステム管理者が学生の一つ一つの VM に対して対応を行うのは現実的ではない。情報工学科のブレードサーバ上に VM に対する適切な権限を学生に委譲し、VM の起動・停止等の操作をさせる必要がある。また、VM 上で動く OS のセキュリティを適切に管理する必要がある。

†1 琉球大学理工学研究科情報工学専攻
Interdisciplinary Infomation Engineering, Graduate School of Engineering and Science, University of the Ryukyus.
†2 琉球大学工学部情報工学科
Infomation Engineering, University of the Ryukyus.

2. libvirt

VM 管理ツールである virsh を含む、仮想マシンの制御を抽象化したライブラリである。VM の情報を習得、操作することが可能な API 群となっている。C 言語の API を持っており、それ以外の言語にもバインディングされている。

図 1 は libvirt のアーキテクチャの概要である。アプリケーションから libvirt API を呼び出すと、API に従って内部の VMM API もしくは資源管理 API を呼び出し、制御する。

libvirt は VM の管理だけでなく、仮想ネットワーク、仮想ストレージも管理することができる。もとも

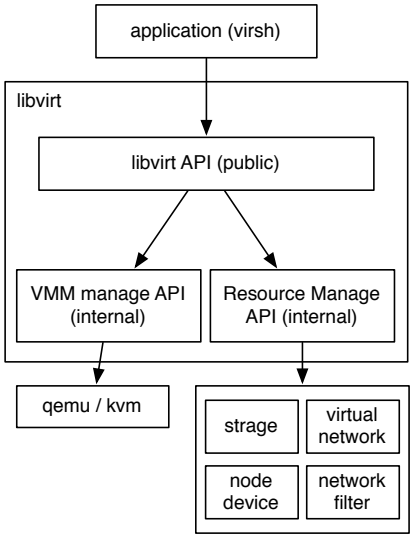

図 1 libvirt architecture

とは Xen に対して API を提供していたが、2014 年現在、多くのハイパーバイザに対応している。本研究では KVM をハイパーバイザとした。libvirt は XML にパラメータを保存することができる。以下が XML に保存できるパラメータである。

- VM 名 (domain 名)
- 割り当てる CPU・メモリ
- ディスクの形式
- 起動オプション
- ネットワーク設定
- コンソール設定

libvirt でこれらを管理することにより、ハイパーバイザの煩雑なオプションの管理をしなくてすむ。

3. virsh

libvirt には virsh というコマンドラインインターフェイスがあり、libvirt の API でできる制御の殆どを virsh で制御できる。VM の起動や停止、情報の表示、ゲストが接続しているネットワークやデバイスの管理をすることができる。また、virsh を使用することでゲストを別のホストへ移行することも可能である。この virsh をラップし、複数の学生が学生自身の VM のみを操作できるように実装する。

4. Kernel-based Virtual Machine (KVM)

Linux 自体を VM の実行基盤として機能させるソフトウエアで、無償で使用することのできるオープンソースである。完全仮想化により、OS の仮想化環境を提供する。

図 2 は、KVM のアーキテクチャである。KVM は Linux 用のカーネルモジュールとして実装されており、OS が持つメモリ管理プロセスやスケジューリング機能を利用している。そのため他の仮想マシンソフトウエアに比べ、KVM 自体のコードは簡潔なものになっている。

Intel VT や AMD-V などの仮想化支援機能を持つプロセッサや BIOS を持っているパソコン上で動かすことができる。

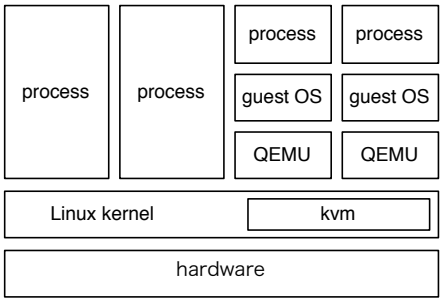

図 2 KVM architecture

KVM は ie-virsh が動作する VM 環境のハイパーバイザである。ブレードサーバをホストとして管理対象の VM を乗せる。

5. ie-virsh

ie-virsh は、virsh をラップして作られた VM 管理用のツールである。学生は ssh で学科アカウントを使用してブレードサーバに接続し、ie-virsh を使用して VM を操作することができる。表 1 が ie-virsh の機能である。

表 1 ie-virsh のコマンド

define	XML の template を元に domain を作成
undefine	define で作成した domain を削除
list	define で作成した domain を一覧表示
start	指定した domain 名の VM を起動
destroy	指定した domain 名の VM を停止
dumpxml	domain の XML を参照

ie-virsh には virsh にあるような、ネットワークの構成などの管理者側がするべき操作はなく、管理者でない学生は操作できないようになっている。また学生は、他の学生の VM を操作することもできない。

学生が ie-virsh を使用して VM を起動する手順はこうである。まず学生のノート PC で、VMWare や VirtualBox を使って Linux をインストールし、イメージを作成する。作成したイメージをブレードサーバにアップロードする。イメージを VirtualBox の OVF 形式から qcow2 形式に変換して指定のディレクトリ

に配置し、以下のコマンドを実行する。

```
% ie-virsh define [domain name]
```

そうすると、template XML を元に domain が生成される。ie-virsh は XML の template を持ち、その template は学生が VM のイメージをブレードサーバにアップロードして define した際に使用される。template には virsh 上で VM を使用するために必要な設定が記述されている。生成された 設定 XML は KVM の所定の位置に格納される。設定 XML には、OSFS2 上の学生の固有の場所の仮想マシンイメージへの path が記述されている。学生は、指定の位置にイメージを指定することにより任意の VM を実行することができる。生成された domain は以下のように起動することができる。

```
% ie-virsh start [domain name]
```

自身が持っている VM の状態を、下記のコマンドで見ることができる。

```
% ie-virsh list
uid 45273 gid 45273 name taiki
    -     test/taiki/01         shut off

    -     test/taiki/02         shut off
```

5.1 ie-virsh の動作環境

ホストサーバの環境は以下である。

- OS:debian
- CPU: Intel(R) Xeon(R) CPU X5650 @ 2.67GHz
- CPU 数: 2
- Core 数: 6 core (論理 core 24)
- メモリ: 128GB

SAN は Oracle Cluster Filesystem(OCFS2) でのフォーマットを行った。OCFS2 は汎用の、共有ディスククラスタファイルシステムである。一つのブロックデバイスを複数の PC から同時に読み書きすることができる。

各計算ノードがそれぞれファイルシステムへの処理を行い、ストレージに対して個別に読み書きをする。一貫性を持った読み書きを実現する機構として Distributed Lock Manager (DLM) が使われる。DLM で他の計算ノードと矛盾しないように調整しながらそれぞれの計算ノードが並行してストレージへの読み書きを行うことで、全体として一貫性のあるファイルシステムを実現している。

標準のファイルシステムインターフェイスを通じてすべてのノードが並行してストレージに読み書きできるため、クラスタにまたがって動作するアプリケーションの管理が容易になる。

OCFS2 との接続は、図 3 となっている。複数台のブレードサーバから OCFS2 でフォーマットされたファイルシステムへ接続し、書き込みを行う。ブレードに内蔵されている SSD では大量の VM イメージを保存するには足りないため、外部の記憶装置を利用する必要があるためである。また別のブレードサーバ上に KVM をたてた場合に移行が容易になる。

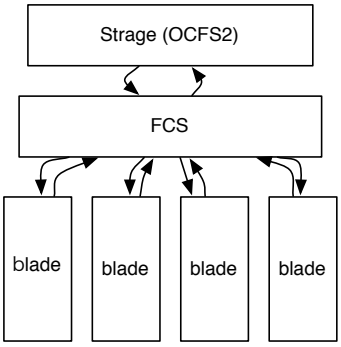

図 3 San structure

情報工学科では、グローバル IP アドレスを取得することができる。学生は取得した IP アドレスを使用して VM へ ssh アクセスする。そのため、virsh が作る仮想ネットワークではなく、情報工学科の DHCP サーバによって学生の IP アドレスが受け取られるように設定する必要がある。それには仮想ブリッジを配置してその問題に対処した。

図 4 は ie-virsh を使用して学生が VM を配置するホストの構成である。

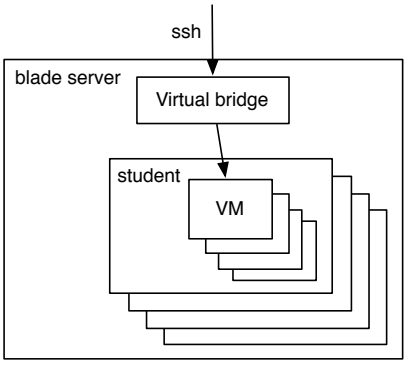

図 4 Server structure

5.2 資源の制限

学生が作成する VM は、XML のテンプレートを元に作成される。テンプレートは VM にしておくべき設定が記述されており、ie-virsh はこのテンプレートを元に学生の XML ファイルを生成する。XML template にされている設定は以下のようになる。これによって学生が使用するブレードサーバの資源を制限し、過剰なメモリや CPU の確保を防ぐ。

- ネットワークの設定
- I/O 設定
- VM イメージのフォーマット

表 2 vagrant の主なコマンド

up	Vagrant Box を起動
destroy	仮想マシンの削除
halt	起動している Vagrant Box の停止
ssh	起動している Vagrant Box へ ssh 接続
status	ステータスの確認
box add	新しい Vagrant Box の追加

- CPU 数
- メモリ容量

ie-virsh は XML template を元に以下の設定を追記したファイルを作成する。

- VM 名
- UUID
- VM イメージの配置

また学生が VM を大量に作成することを防ぐため、作成できる VM の台数を 4 台に制限した。授業 Operating System を 60 名の学生が受講する場合、最大 240 台作成される。

学生の Web サービス構築の学習や、授業 Operating System で複数の学生の使用に対応するため ie-virsh はマルチユーザで動作する必要がある。情報工学科で使用されている LDAP サーバの情報を使用して、学生が学科のアカウントで ssh ログイン可能な状態に設定した。そうすることで、学生一人一人のアカウントを登録する必要がない。またアカウント名は学籍番号になっているため学籍番号で管理することができる。

複数の学生に VM を貸し出す際に管理しなければならないのは、学生が持つ権限である。ie-virsh で学生が可能なことは、自身の VM の作成・削除と、起動・停止である。すべての VM に関係する仮想ネットワークの設定や、他の学生の VM に対する操作はできないように設定した。

5.3 Vagrant Box の利用

VM を使用する際は学生の PC で VM を設定し、VM イメージをアップロードさせるという形をとった。授業 Operating System では VM を学ぶ環境として学生の PC で Vagrant を使用させた。

Vagrant は異なる環境に移行可能な開発環境を簡単に構築・管理・配布することが出来る開発環境作成ツールである。手軽にテスト環境を導入することができ、変更が加わっても開発環境・本番環境に自動的に適用される。また、環境を気軽に捨てることも可能である。ホスト環境として、VirtualBox や VMWare などで動かすことができる。表 2 は Vagrant で使用することの主な機能である。

また Vagrant で仮想マシンを利用する際に、仮想マシンのベースとなるイメージファイルが Vagrant Box である。Vagrant で Vagrant Box を VM イメージとして起動し、開発環境を構築し配布することができる。また配布されている Vagrant Box を取得し、Vagrant で起動し使用することが可能である。

Vagrant は学生の PC の VirtualBox で使用させたため、Vagrant Box の VM イメージは VirtualBox に対応する OVF 形式となっていた。そのため OVF 形式から、KVM で動作する形式へ変換する必要があった。学生はブレードサーバへ VM イメージをアップロードする際、OVF 形式から KVM で動作する qcow2 形式へ変換する。

しかし Vagrant Box イメージは簡易なパスワードとユーザ名で Vagrant から管理されており、そのままブレードサーバへアップロードしグローバル IP アドレスを割り当ててしまうと、外部からの攻撃を受けてしまう。そのためブレードサーバへアップロードしたイメージを検知し、攻撃されないような設定かどうかを確認する必要がある。

6. ie-virsh と他のツールとの比較

ie-virsh を実装し実際に授業 Operating System で使用した。情報工学科では ie-virsh や KVM だけでなく他の管理ツールやハイパーバイザを使用している。また使用を検討したものもいくつかある。それらと ie-virsh を比較する。

6.1 OCFS2 と NAS の比較

ie-virsh の動作するサーバの構成では、SAN へ複数の PC が接続し同時にアクセスする。ext3・ext4 の様なファイルシステムでフォーマットを行うと、複数の PC から同時にアクセスした際に整合性が取れずファイルが破損してしまう。OCFS2 は SAN 上の複数の PC から書き込まれてもファイルの整合性を保つ。KVM から NFS で NAS に接続する、あるいは、NFS 経由で自身の file system にアクセスすることでも同様のことが実現できるが、余計なネットワークトラフィックが出てしまう。

6.2 KVM と VMWare ESXI の比較

ie-virsh のハイパーバイザとして KVM を利用した。情報工学科では VMWare ESXI を利用しているが、VMWare を使用するためにはライセンス等でコストがかかる。KVM であれば Linux ベースの OS で利用できるため、無償で利用可能である。性能的にもほぼ同等であるが、現状では VMWare の方が若干高速である。

6.3 管理ツールの比較

ie-virsh は virsh をラップするという形で実装が行われているが、VM を管理するツールは virsh だけではない。情報工学科では、授業や学科のシステムの管理として他のツールも使用している。ここでは他のツールを使用した場合と比較する。

6.3.1 vSphere Client

vSphere Client は仮想環境の統合管理をするプラットフォーム vCenter Server と接続し、管理するため

に使用される。vSphere Client は権限の詳細な設定が可能となっており、複数の学生に対して VM を配布し権限を管理するツールとしては適している。必要な権限を持つテンプレートを作り、それを学生のアカウントに割り当てることで学生が VM を使った実習をすることが可能になる。自動で、これを行うことも可能だと思われるが、今回は、そこまでは用意していない。手動で権限を委譲する場合は管理者側の操作も多く、60 名の受講者がいる授業 Operating System で学生への権限の配布に手間と時間が必要である。

ie-virsh はそういった権限を委譲するという操作が必要なく、また機能も学生が VM を操作するのに十分なように作られている。

6.3.2 Vagrant

Vagrant は KVM をプロバイダとするプラグインを持っている。よって KVM を VirtualBox の様にプロバイダとして Vagrant を動かすことが可能である。Vagrant をマルチユーザに対応させ、学生が使用可能に設定できるか試した。Vagrant を使うことができれば、KVM 用に VM イメージを変換することは必要だがノート PC の Vagrant と同様に操作することができる。

しかし Vagrant の KVM プラグイン vagrant-kvm はネットワークの実装が複数人で使用できるように実装されておらず、virsh のように使用するには向かなかった。virsh の様に使用するためには、vagrant-kvm の実装に手を入れネットワーク部分を改良する必要がある。Vagrant でできることは、ほぼ ie-virsh と同等であり、既存の Box を使えること以上の利点はなかった。

6.3.3 Web サービス実装

VM を操作するインターフェイスとして、Web サービスを使用することも可能である。ie-virsh はコマンドラインでの操作になるが、Web サービスとして実装する場合は GUI 操作になる。別のサービスとして、vSphere の API を使用した Web サービスが情報工学科では使用されている。情報工学科の VMWare ESXi へ VM を作成し、起動・停止することができる。

Web サービスでは、GUI を操作するためにブラウザを起動しなければならない。また新しく GUI の操作に慣れる必要がある。ie-virsh は CLI であり、また virsh に近いため virsh の操作として覚えられる。しかし Web サービスは情報工学科のサービスであり IP アドレスの配布と連携しているため、IP 登録を自動で行う。ie-virsh は IP アドレスの登録は情報工学科のサービスを利用して行うため、新規に VM を作成する際は複数のサービスにまたがって操作しなければならない。

webvirt[1)] [p.6] が複数の VM を管理するために実装されていた。仮想環境の Web 管理ツールである。シングルノードのみを管理する目的で開発されているため、複数の学生が使用するには対応させる必要がある。

6.3.4 OpenStack

OpenStack はクラウド基盤ソフトウエアである。仮想サーバやブロックストレージ、仮想ネットワーク等のリソースを提供する。物理層に依存することなく標準化された IT インフラストラクチャである。しかし OpenStack はマルチユーザではないため、複数の学生の権限を管理することには向かなかった。

7. ie-virsh の改善点

ie-virsh には VM 管理ツールとして必要な実装が不足している。また ie-virsh を運用し、追加する必要があると判断した機能を挙げる。

学生が VM を使用してサービスを構築するにあたり、外部からの攻撃や侵入を防ぐために VM がセキュアに設定されているかをチェックする必要がある。また VM までのトラフィックを監視し、学生の VM が外部へ不正なトラフィックを送信していないかを検知しなければならない。

VM の設定で検査が必要なものとして、まずパスワードの設定が挙げられる。学生が安易なユーザ名とパスワードを VM に設定してしまうと、攻撃や侵入を受けてしまう。そういった不正なアクセスを防ぐために、VM に対してパスワードの解析を行い、安易なユーザ名やパスワードを使用していると変更を促すシステムが必要である。そうすることで侵入されることを防ぐことができる。

また VM が外部からの不正なトラフィックを受け取らないように、VM のファイアウォールを検査する必要がある。外部から VM に対して頻繁に行われる攻撃をかけ、侵入することができた場合に設定の改善を促す。そうすることで学生の VM が外部から侵入されることを防ぐ。それだけでなく、VM までのネットワークトラフィックを監視することも必要である。パスワードやファイアウォールの検査後に、学生が攻撃されやすい設定に変更した場合はトラフィックを監視して対応する。不正なトラフィックを発見した場合は管理者や学生本人に伝え、対策をとる。

学生が VM に Web サービスを構築し、Web サービスを始めた後は運用を続ける必要がある。大学からのアクセスであれば問題ないが、遠方からのアクセスだと Web サービスを閲覧するのに時間がかかる。そのため遠方からのアクセスが増えてくると、クラウドへデプロイしたいという需要がでてくる。その需要に対応するため、クラウドへ Web サービスを構築したサーバをデプロイする機能を追加する必要がある。そうすることで Web サービスを手間をかけずに外部に移すことができる。

セキュリティのチェックやクラウドへのデプロイに

は、Ansible や Chef などの構成管理ツールを利用する。そしてデプロイ後は serverspec などの構成テストツールを使用し、構成をテストする。このように自動化することによって管理者側が VM の調査をしなければならない手間を軽減する。

8. ま と め

本研究では ie-virsh を実装し、既存の管理ツールとの比較を行った。学生の権限と使用できる資源を絞ることによって管理者の手間を減らすことができ、学生による不正な操作を制限できた。学生は情報工学科のアカウントを持っていれば、Web サービスの構築や課題の学習のために管理者とのやり取りなしに VM を作成し利用することができる。

セキュリティ面の対応が甘く、学生の VM に対する攻撃を防ぐために対策する必要がある。VM の脆弱性を検索し、対応しなければならない。またネットワークトラフィックを監視し、不正なトラフィックを検知することも必要である。

参 考 文 献

1) 玉城将士，河野真治：Cassandra と非破壊的構造を用いた CMS のスケーラビリティ検証環境の構築，日本ソフトウェア科学会第 28 回 (2011).