

# VirShaddai

MANAGEMENT OF THE VIRTUALMACHINE

—そんな GUI で大丈夫か？



仮想化の管理を  
スルゲス!

—一番いいシェルを頼む。

# virsh, 目標を操作する!

※やわらか研の方言で「こんにちわ」の意

## ㊦ まえがき

イエーイ、仮想化してるー?

というわけで、「ZaWorld」は今回も仮想化ネタです。  
なにせ、一番ホットな話題ですからね!

…そ、ネタがないだけとか言うな。

今回は特にニッチな内容となっていますが、仮想化関係に触っていて、「libvirtってなんや…?」「いちいちX立ち上げてVirt-Manager起動するのめんどくさいんだけど」などと思った方もいらっしゃるでしょう。そんなあなたにうってつけの内容となっています。たぶん。

本書が「マシンの管理・操作はCUIじゃないと落ち着かない」「宗教上の理由からGUIを使えない」とお嘆きのそのあなたのお役に立てれば幸いです。

terror / @syonbori 拜

## ④ Libvirt と virsh

libvirt(りぶばーと)とは、LIBrary for VIRTualization、つまり仮想化に関する処理をまとめたツール・環境です。極めて簡単に言えば、いろいろな仮想計算機環境を同じ方法で操作できるようにするためのプログラムたちです。

名前からするとlibpngやlibsslなどのように、ある処理を実行するためのライブラリが何かに聞こえますが、その実体は多岐にわたり、そのようなライブラリから仮想計算機環境を操作するためのデーモン、管理を容易にするためのツールなどから構成されています。

現在、仮想計算機環境を構築するためのソフトウェア(仮想計算機モニタ、VMM: Virtual Machine Monitor)は様々なものがあります。大規模なシステム運用でも使われているVMware社のVMware ESXi ServerやCitrix社のXenServer、Microsoft社のHyper-Vなどがあります。個人がフリーで利用可能なものとしては、様々なCPUやデバイスをエミュレートすることができる汎用エミュレータのQEMU(きゅーえみゅ)やOracle社のVirtualBox(ばーちやるぼっくす)、最近活発に開発が行われているkVM(kernel-based Virtual Machine: けーぶいえむ)や仮想化ゲームの火付け役のXen(ぜん)などがあります。

これらの仮想計算機モニタを用いて仮想計算機環境を構築する場合、それぞれの流儀に従って仮想計算機を構築し、実行し、管理する必要がありました。管理用インタフェースとしては、VMwareではvSphere、XenServerではXenCenter、kVMではコマンドライン上からオプションを指定して起動、といった具合です。

このように仮想計算機モニタごと管理用インタフェースが異なっていると、様々なVMMを持つような環境では非常に管理コストが高くなってしまいます。いろんなVMMをまたいで、一括して管理コマンドを発行する、といったことが非常に難しいのです。

そこで、様々な仮想計算機への操作を、VMMによらずに実行できるような環境が提案されました。あるマシンがどの仮想計算機モニタ上で動作しているようにも、対応さえしていれば決まったコマンドで操作が行えるようにしよう、というものです。これをVMMに対する透過性と呼びます。これを実現したのがlibvirtです。

libvirtの登場により、たとえばVMMがVMware ESXiでもXenでも、kVMでもVirtualBoxでも、同じコマンドで同じ操作を行うことができるようになりました。また、その操作はAPI (Application Programming Interface) が公開されており、必要であれば自分でこれらの操作を行うプログラムを作成することができるようになっていきます。

このようにして、VMMの種類によらない管理方法が登場しました。そして、管理上の都合から、コマンドラインベースの管理ツールが用意されています。それがvirshです。各社が用意している管理ツールは確かに直感的に操作可能で分かりやすいものも多いですが、大規模なものとなると、シェルのようにテキストベースでの管理ツールが欲しくなるものです。virshはそのような要求に応えるツールで、VIRTUALmachine SHellの名の通り、OSに対するシェルのような機能を提供します。グラフィカルな画面なしに、キーボードから管理を行うことができます。

## ☆ インストール!

ぐぐれ。

で終わらせてしまうのは味気ないですが、現在ではディストリビューションのパッケージ管理ツールからインストール可能となっていることが多いようです。Redhat系では、Virtualizationグループをインストールすると一緒にインストールされるようです。Virt-Managerというグラフィカルな仮想マシン管理ツールと一緒に既にインストールされていることも多いようです。

## ☆ virsh と VMM

virsh (というか libvirt) は VMM に対する透過性を提供します。そのため、どの VMM を対象に作業を行うかを教えてあげる必要があります。現在、libvirt は以下の VMM に対応しているようです；

Xen (ホスト OS ; Linux, Solaris)

kVM (ホスト OS ; Linux)

LXC (Linux のコンテナツール)

OpenVZ

User Mode Linux (Linux 上で "Linux を動作させる")

VirtualBox

VMware ESX

お使いの VMM への対応情報などは、“(VMM名) virsh”などで検索してみましょう。また、libvirt では、管理作業の効率化を図るため、リモートにある VMM を管理することもできます。手元の Linux マシンから、サーバーームでうなりを上げる VM ホストマシン達を管理することができるのです。

## ④ virsh の基本操作

では、さっそくvirshを起動してみましょう。VMMが実行されているマシンと同じマシン上で作業を行う場合は、rootになってvirshと打てば起動できるでしょう。リモートのVMMを管理する場合には、次のようにして指定してやればよいでしょう。

```
$ virsh --connect qemu+ssh://root@example.com/system
```

QEMU+kVMにSSH経由で接続する場合には、このように指定してやればOKです。virshが起動すると、次のように表示されるはずです。

```
Welcome to virsh, the virtualization interactive terminal.
Type: 'help' for help with commands
      'quit' to quit
virsh #
```

“virsh #”というのがプロンプトです。このプロンプトに対して様々なコマンドを入力することにより、仮想計算機を操作することができます。“help”と入力すると、コマンドの一覧が表示されます。また、“help コマンド名”とすると、そのコマンドのヘルプが表示されます。用意されているコマンドの詳細は別に譲るとして、普段使うコマンドについて紹介します。

### \* quit:virshを抜ける

基本です。ね。終わらせ方は大事です。

### \* version:バージョン情報を表示する

virshが用いるライブラリやVMMのバージョンを表示します。

```
virsh # version
Compiled against library: libvir 0.4.6
Using library: libvir 0.4.6
Using API: QEMU 0.4.6
Running hypervisor: QEMU 0.9.1
```

### \* list:仮想計算機一覧を表示する

おそらく最も良く使うコマンドでしょう。"list" だけだと現在稼働中の仮想計算機を、"list --all" とすると全ての仮想計算機を表示します。

```
virsh # list --all
Id Name                State
-----
 5 hoge                running
 7 ceph1               running
24 fuga                running
- ceph2                shut off
```

### \* dominfo:仮想計算機の情報を表示する

listコマンドでは表示されないような、仮想計算機の情報を表示するのが"dominfo" コマンドです。

```
virsh # dominfo ceph1
Id: 7
Name: ceph1
UUID: f4932be0-abfb-6700-9f82-ecabc6420353
OS Type: hvm
State: running
CPU(s): 2
CPU time: 652927.1s
```

```
Max memory: 1048576 kB
Used memory: 1048576 kB
Autostart: enable
```

### \* start:仮想計算機を起動する

“start” コマンドを用いて、仮想計算機を起動します。そのままですね。

### \* shutdown:仮想計算機を優雅に終了する

“shutdown” コマンドでは、仮想計算機を「優雅に」終了させます。優雅に (gracefully) というのは、VMM ごとに用意されているシャットダウン方法を使って、できる限り安全にシャットダウンさせる、という意味です。VMM、OS の両方が対応していれば、OS に対して終了命令を出したときのようにシャットダウンが行われます。

### \* destroy:仮想計算機の電源ボタンを仮想的に長押しする

“shutdown” コマンドは「優雅に」終了させるコマンドでしたが、“destroy” コマンドは「無慈悲に」終了させます。実際のパソコンで、電源ボタンを長押しした場合と同じような処理になります。極力、“shutdown” コマンドを使いましょう。

メモ: virsh では、何かを開始するコマンドは \*start、何かを終了するコマンドは \*destroy となっています。stop じゃなくて destroy という、ちょっと物騒な名前ですが、そういうルールがあるようです。覚えましょう。



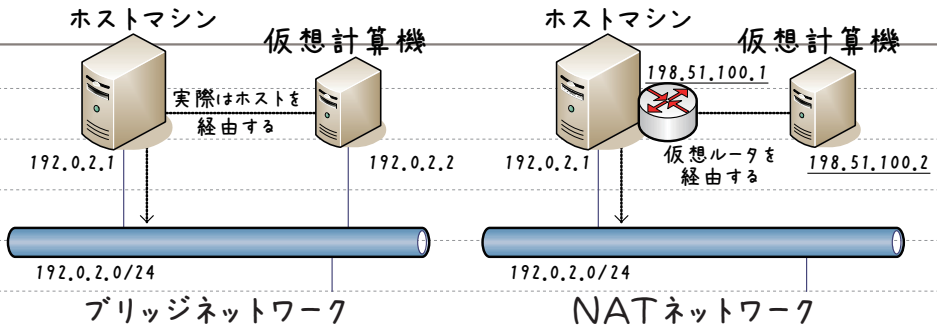
## ④ 仮想計算機とデバイス

libvirtでは、仮想計算機の構成（CPUがいくつ、メモリがどれくらい、HDDが…など）をXMLファイルに記述し、それを取り込むことで、仮想計算機を構築します。しかし、一から手書きするのは非常に面倒なので、どこかからテンプレートを見つけてくるか、Virt-ManagerのようなGUI管理ツールを使ってしまうのが手取り早いでしょう。本節では、仮想計算機の構成をいじる方法やその仕組みについて紹介します。

### ☆ ストレージ

仮想計算機のデバイスと言えば、ストレージの設定が面倒なものとしてあげられます。デバイスのタイプとして、“IDE Disk”，“SCSI Disk”，“virtio Disk”，“IDE CD-ROM”，“Floppy Disk”などから選択することができます。見慣れない“virtio”（ばーとあいおー）というのは、仮想計算機からデバイスへのアクセスを効率よく行うための規格で、ゲストOSがvirtioに対応している必要があります。

仮想計算機にどのディスクを見せるかという指定はVMMに依存しますが、QEMU+kVMでは、QEMUが持つエミュレータなどの機能を用いることができます。ディスクイメージファイルを用意し、ゲストOSにこれをHDDのように見せるという方法が一般的ですが、大規模なシステムやマイグレーション（ゲストOSの移動）をサポートするような場合では、NFSやiSCSIといった方法を用いてディスクを指定することもあります。



## ☆ ネットワーク

仮想計算機にネットワーク機能を提供するためには、何らかの方法でゲストOSにネットワークカード (NIC) を用意してやる必要があります。仮想的なNICを作り出してゲストOSに提供し、この仮想的なNICを実際のネットワークに接続してやる必要があります。

ゲストOSに提供する仮想NICについても、“e1000”、“Pcnet”、“rtl8139”、“virtio”などから選択することができます。ここで指定したNICのドライバをゲストOSにインストールする必要があります。

また、この仮想NICを実際のネットワークに接続するための設定も必要になります。この設定もlibvirtが面倒を見てくれます。設定にはブリッジとNATの2種類があります。ゲストOSの用途に応じて、適切な方を選びましょう。ブリッジの場合、ホストOSが接続されているのとネットワークにゲストOSも参加するような形になります。同じLAN内にゲストOSを配置したいような場合には、ブリッジが適切でしょう。NATの場合、ゲストOSはホストOSの中に作成された仮想的なルータを経由してアクセスするような形になります。様々なパケットを流すようなネットワーク実験をする場合や、別サブネットに配置したいような場合に適切でしょう。

## ☆ デバイスなどを操作するコマンド

### \* define, create:ドメインをXMLファイルから作成する

“define / create” コマンドは、既存のXMLファイルから仮想計算機（ドメイン）を作成します。“define” コマンドは作成を、“create” コマンドは作成後に起動を行います。

### \* undefine:ドメインを抹殺する

“undefine” コマンドは、作成したドメインを削除します。少し予想外なことに、紐付けられているXMLファイルまで削除されてしまうので、細心の注意を払って実行しましょう。

### \* dumpxml:ドメインのXMLファイルを表示する

“dumpxml” コマンドは、指定したドメインの構成を格納したXMLファイルを表示します。ある仮想計算機の構成が、XMLではどのように表現されているかを確認する際に便利です。

### \* edit:ドメインのXMLファイルを編集する

“edit” コマンドを用いると、指定したドメインの構成XMLファイル編集することができます。紐付けられているXMLファイルを変更しただけでは、libvirt側に反映されません。XMLファイルに変更を加えたい場合には、必ずこのeditコマンドを使いましょう。

### \* net-list:ネットワーク一覧を表示する

ネットワーク版の“list” コマンドです。同様に、“--all” オプションが利用できます。

```
virsh # net-list -all
Name                State      Autostart
-----
cephnet             active    yes
default             active    yes
```

### \* net-edit:ネットワークの設定を編集する

ネットワーク版の”edit”コマンドです。そのままですね。

### \* net-start / net-destroy:ネットワークを開始・停止する

libvirtは、iptablesやrouteコマンドを駆使してネットワーク環境を構築します。その処理を開始・停止するのが”net-start / net-destroy”コマンドです。

### \* net-define, net-create:ネットワークを作成する

ネットワーク版の”define / create”コマンドです。定義だけか起動も行うかという、”define”と”create”の違いも同様です。

### \* net-undefine:ネットワークを抹殺する

ネットワーク版の”undefine”コマンドです。こちらも同様にXMLファイルまで削除されてしまいます。

メモ: net-startを行うと、裏ではiptablesコマンドやrouteコマンドなど、ネットワークの設定を行うコマンドが自動で実行されます。仮想計算機用にiptablesで個別にルールを設定したい、というような場合には要注意です。

## ✧ XML ファイル例 (windows)

<code>&lt;domain type='kvm' id='5'&gt;</code>	
<code>&lt;name&gt;kvm_win2008r2&lt;/name&gt;</code>	④ 名前とUUID
<code>&lt;uuid&gt;b50f5be1-8cca-e536-0770-6f97d6fc375b&lt;/uuid&gt;</code>	
<code>&lt;memory&gt;786432&lt;/memory&gt;</code>	④ メモリ割り当て
<code>&lt;currentMemory&gt;786432&lt;/currentMemory&gt;</code>	
<code>&lt;vcpu&gt;2&lt;/vcpu&gt;</code>	
<code>&lt;os&gt;</code>	④ OSの設定
<code>&lt;type arch='x86_64' machine='pc'&gt;hvm&lt;/type&gt;</code>	64bit
<code>&lt;boot dev='hd'/'&gt;</code>	完全仮想化
<code>&lt;/os&gt;</code>	HDDから起動
<code>&lt;features&gt;</code>	④ 仮想計算機の機能リスト
<code>&lt;acpi/'&gt;</code>	
<code>&lt;apic/'&gt;</code>	
<code>&lt;pae/'&gt;</code>	
<code>&lt;/features&gt;</code>	
<code>&lt;clock offset='localtime'/'&gt;</code>	
<code>&lt;on_poweroff&gt;destroy&lt;/on_poweroff&gt;</code>	
<code>&lt;on_reboot&gt;restart&lt;/on_reboot&gt;</code>	
<code>&lt;on_crash&gt;restart&lt;/on_crash&gt;</code>	
<code>&lt;devices&gt;</code>	
<code>&lt;emulator&gt;/usr/bin/kvm&lt;/emulator&gt;</code>	④ デバイスエミュレータ
<code>&lt;disk type='file' device='disk'&gt;</code>	④ ディスクI/O
<code>&lt;driver name='qemu' type='raw'/'&gt;</code>	QEMUのRAWを指定
<code>&lt;source file='/var/lib/libvirt/images/win2008r2.img'/'&gt;</code>	ファイルパス
<code>&lt;target dev='hda' bus='ide'/'&gt;</code>	IDEバスに接続
<code>&lt;/disk&gt;</code>	

<code>&lt;disk type='file' device='cdrom'&gt;</code>	④ ディスク2(CDROM)
<code>&lt;driver name='qemu' type='raw'/&gt;</code>	QEMUのRAWを指定
<code>&lt;target dev='hdc' bus='ide'/&gt;</code>	IDEバスに接続
<code>&lt;readonly/&gt;</code>	CD-ROMなので
<code>&lt;/disk&gt;</code>	ReadOnly
<code>&lt;interface type='bridge'&gt;</code>	④ 仮想NIC
<code>&lt;mac address='52:54:00:68:f7:82'/&gt;</code>	設定するMACアドレス
<code>&lt;source bridge='br2'/&gt;</code>	ブリッジするI/F
<code>&lt;target dev='vnet3'/&gt;</code>	仮想NICの名前
<code>&lt;model type='e1000'/&gt;</code>	e1000をエミュレート
<code>&lt;/interface&gt;</code>	
<code>&lt;serial type='pty'&gt;</code>	
<code>&lt;source path='/dev/pts/7'/&gt;</code>	
<code>&lt;target port='0'/&gt;</code>	
<code>&lt;/serial&gt;</code>	
<code>&lt;console type='pty' tty='/dev/pts/7'&gt;</code>	
<code>&lt;source path='/dev/pts/7'/&gt;</code>	
<code>&lt;target port='0'/&gt;</code>	
<code>&lt;/console&gt;</code>	
<code>&lt;input type='tablet' bus='usb'/&gt;</code>	④ 入力デバイス
<code>&lt;input type='mouse' bus='ps2'/&gt;</code>	
<code>&lt;graphics type='vnc' port='5903' autoport='yes'</code>	④ 画面出力(VNC)
<code>listen='127.0.0.1' keymap='ja'/&gt;</code>	
<code>&lt;/devices&gt;</code>	
<code>&lt;/domain&gt;</code>	

## ㊦ XML デバイス定義例

edit-\*コマンドで編集できるXMLファイルの定義サンプルです。仮想計算機の構築はGUIのツールで行い、細かい修正はCUIで行うと楽かも知れません。Virt-Managerの場合、GUIからは設定できないオプションがあったりするかもしれませんが、XML定義を自分で書いてやれば問題なく動作します。

Virt-Managerのネットワーク設定は特に不思議な動作をすることがあるので、そういうときは自分で書きちゃいましょう☆

### ☆ ディスク定義

```
<disk type='file' device='disk'>
  <driver name='qemu' type='qcow2'/>
  <source file='/var/lib/libvirt/images/ceph1.qcow2'/>
  <target dev='vda' bus='virtio'/>
</disk>

<disk type='file' device='disk'>
  <driver name='qemu' type='raw'/>
  <source file='/var/lib/libvirt/images/win2008r2.img'/>
  <target dev='hda' bus='ide'/>
</disk>
```

ディスク定義では、driver要素でエミュレータとタイプを、source要素で対応するファイルを指定します。target要素で接続するバス(ide, virtioなど)と仮想計算機に見せるデバイスの名前(hda, vdaなど)を指定します。

---

## ★仮想NIC定義

```
<interface type='network'>  
  <mac address='52:54:00:3e:25:f6'/>  
  <source network='cephnet'/>  
  <target dev='vnet0'/>  
  <model type='virtio'/>  
</interface>
```

```
<interface type='bridge'>  
  <mac address='52:54:00:68:f7:82'/>  
  <source bridge='br2'/>  
  <target dev='vnet3'/>  
  <model type='e1000'/>  
</interface>
```

仮想NICの定義では、interface要素のtype属性として、NATの場合は"network"を、ブリッジの場合は"bridge"を指定します。Virt-Managerではブリッジ接続を作れない場合があるので、その場合は手作業でやりましょう。Routedネットワークとして作成し、あとから手作業で修正するのが楽です。

NATの場合は、network-startコマンドによりルーティングの設定が自動的に行われるはずですが、これにより、自動的に仮想計算機からのパケットが実ネットワークにルーティングされます。

ブリッジの場合は、自分でブリッジインタフェースを作成し、実NICと結びつけてやりましょう。ちょっと複雑なので詳しくは、「仮想計算機ブリッジ作成」でぐぐろう！



## ㊦ あとがき

というわけで、いかがでしたでしょうか。

いかがでしたでしょうか。紙面（と時間）の都合からだいぶ簡単になってしまいましたが、virshの紹介でした。Virt-Managerも徐々に洗練されてきており、GUIでの操作も悪くなってきていますが、やはりCUIが楽で良いです。よねの拙著で不足している部分は、是非helpコマンドやhelp hogeで調べてみてください。

おりとややこしい、poolやvolなどについての説明や、ネットワーク設定の説明等、いろいろと足りない状態ではありますが、何か少しでもお役に立てれば幸いです。もし夏があれば、そこできっと「もっとちゃんとした」バージョンを頒布できるかも知れません。

それでは、また機会がありましたら。

2010年12月 このままだと国際会議の論文まにあわないぞ…？

terror / @syonbori 拜

### ㊦ フォント

本書では、以下のフォントを使用しています。ありがとうございます。

☆本正文ほか；ふい字（作者；ふい様）

<http://hp.vector.co.jp/authors/VA039499/>

☆セクションほか；あんずもじ（作者；京風子様）

<http://www8.plala.or.jp/p.dolce/>

☆記号ほか；アームドバブツ（作者；ミリメートル様）

<http://calligra-tei.oops.jp/>

なお、表紙のlibvirtのロゴは、クリエイティブ・コモンズ 表示-継承 3.0 非移植ライセンスのもとに利用を許諾されています。

### ㊦ 奥付

仮想化なう 2010年冬号「Virshaddai」

発行日；2010年12月31日（於；コミックマーケット79）

著者；terror

発行；ZaWorld

複製・複写・再頒布等自由どうぞ。ただし、著者はその責任を負（え）いません。



Z A W O R L D



1 0 1 2 3 1

ISBN 271828-1828-459464  
C79 DAY3 KataPu-57b ¥100

発行：ZaWorld

定価：本体100円 + コミケ期間中の貴重な時間

話をしよう。

あれは今から36万…

いや、1万4千Commit前だったか。

まあいい、私にとってはつい昨日の出来事だが  
君たちにとってはたぶん、明日の出来事だ。

libvirtは72通りのVMMに対応しているから  
どれを紹介すればいいか…

たしか、最初に筆者が使ったのは…

Xeno

そう、あいつは最初から言うことを聞かなかった。  
私の言うとおりにしていればな…

まあ、いいVMMだったよ。

こんな同人誌で大丈夫か？

大丈夫じゃない、問題だ。

神は言っている…

こんな本を買っている場合ではないと…