Virtualization possibilities in MikroTik RouterOS

by Martin Pína

MUM Poland, Wroclaw, 2nd March, 2010

Why virtualization?

- Security (create whole running environment)
- Logical segregation (HTTP/FTP/DNS)
- Testing (new setups, new features)
- Better use of resources
- Easier administration

Why to virtualize RouterOS?

- To have better usable networking possibilities in server setup (e.g. RouterOS as guest in VMware Server / ESXi)
- To use some of RouteOS'es package to be used as standalone virtual server (Dude / HOTSPOT / VPN server / User Manager)
- To give the customers their own virtual router

Why RouterOS as Host?

- A router is probably used / needed anyway
- To save money for running more computers (initial cost / electricity)
- Virtualization on non-x86 platforms is supported by MikroTik
- Wireless router cannot be easily virtualized (PCI passthrough)
- Why not?

Virtualization available in RouterOS

- ~~XEN for x86~~
- KVM for x86
- MetaROUTER for mipsbe and ppc

History of virtualization in RouterOS

- since v3.11 support for XEN on x86 (possible to run non-RouterOS guest - Linux)
- since v3.21 support for MetaROUTER on mipsbe platform
- since v3.26 support for MetaROUTER on ppc (RB1000)
- since v3.26 support for KVM on x86
- since v4.4 dropped support for XEN on x86

Hardware requirements

- Strong CPU
- Enough RAM (min 16MiB for RouterOS guest)

MetaROUTER specific

- Supported architecture (mipsbe / ppc)

KVM specific

- Full support of hw virtualization on x86 (AMD-V or Intel VT-x)

MetaROUTER

- Able to run either RouterOS or OpenWRT patched Linux
- Commonly deployed for customer administered router (RouterOS) or running specific simple task without need of dedicated server (Squid proxy, Asterisk PBX, Apache webserver)
- Already presented
  by Uldis Cernevskis (MUM Prague 2009)
  by Brian Vargyas (MUM Fort Worth 2009)

KVM

- Kernel-based Virtual Machine
- Supported by main Linux distributions (Ubuntu and Red Hat) as their main virtualization platform
- Requires hw with virtualization support (unlike XEN)

RouterOS KVM Status Quo #1

- CPU power and amount of RAM is only limited by €
- Support for running non-RouterOS guests of the same architecture
  (Linux, Windows)
- Ability to run VMs created originaly on Linux Hosts
- Support for accelerated network through virtio NIC (driver available for Linux & Windows)
- Connection to VMs available using
  - Console (text only)
  - VNC (virtual display)

RouterOS KVM Status Quo #2

- Only CLI management
- No 64bit support since RouterOS is only 32bit
- Host CPU power reservation limited to number of assigned CPUs to guest (unlike former XEN implementation)
- Limited possibilities (and speed) for storage (no LVM support, no iSCSI support, virtual disks stored as files)

Current RouterOS KVM Bugs (v4.6)

- Boot disk for RouterOS guest only with hda disk
- Boot for non-RouterOS OSes only with help of virtual CDROM
- Guest system is not possible to reboot/shut-down from Host RouterOS environment, guest hangs in „stopping“ state
- Guest system shuts down completely when rebooted from its own environment

KVM Recommendations

- Use dedicated disk for storing virtual disk files
- Use administratively assigned MAC address if bridge is used between host and guest systems
- Use VNC for non-RouterOS images
- Keep in mind security of virtual machines
  - Accessibility to virtual disk files
  - VNC connection address (and possibly firewall)

KVM Main Options

- menu for creating disks, virtual machines, controlling VM state
  [admin@kvm-demo] > /kvm

- menu for assigning network interfaces to virtual mechines
  [admin@kvm-demo] > /kvm interface

- menu for creating static virtual interfaces
  [admin@kvm-demo] > /interface virtual-ethernet

KVM Guest Disk Creation

- Lets create virtual disk file in RouterOS (recommended for RouterOS guests)

  ```
  [admin@kvm-demo] > /kvm \
  make-routeros-image \
  file-name=sata1/virtual-disk.img \
  file-size=128
  ```

- Or create some file elsewhere and then copy it using ftp or scp into RouterOS physical disk

  ```
  linux-pc:~#dd if=/dev/zero \
  of=/virtual-disk.img bs=1M count=128
  ```

KVM RouterOS Guest Interface Management

- Create virtual interface and assign it to specific pre-existing VM

  ```
  [admin@kvm-demo] > /interface virtual-ethernet add \
  name=demo-tap0 disabled=no
  [admin@kvm-demo] > /kvm interface add type=static \
  interface=demo-tap0 \
  model=virtio virtual-machine=existing-demo-vm
  ```

- Or create a dynamic interface for pre-existing VM

  ```
  [admin@kvm-demo] > /kvm interface add type=dynamic \
  virtual-machine=test
  ```

KVM RouterOS Guest Creation

- Configure new VM with needed settings (name, cpu-availability, memory assignment in MiB, path to pre-existing virtual disk file and optionaly enabling VNC virtual display)

```
[admin@kvm-demo] > /kvm add \
name=demo-routeros-guest cpu-count=1 \
memory=128 \
disk-images=hda:sata1/virtual-routeros.img
vnc-server=0.0.0.0:1
```

KVM non-RouterOS Guest Creation

- Almost no difference, probably there is more RAM needed and virtual CDROM (.iso image file) to boot and install from

```
[admin@kvm-demo] > /kvm add \
name=demo-non-routeros-guest cpu-count=1 \
memory=512 \
disk-images=cdrom:sata1/ubuntu-9.10-server-
    i386.iso,hda:sata1/virtual-routeros.img \
vnc-server=0.0.0.0:2
```

KVM Practical Example

- Live demo...

  $ ssh admin@kvm-demo

Troubleshooting KVM

- Are all the prerequisites fulfilled and enabled? (enough RAM, interfaces, all disk images prepared)
- If VM is not working, what is written in „status“ of the VM?
- More detailed logging may be enabled for KVM.
  [admin@kvm-demo] > /system logging add \
  topics=kvm,debug action=memory disabled=no
- Write to MikroTik support

Resources

- http://wiki.mikrotik.com/wiki/Kvm
- http://forum.mikrotik.com/viewforum.php?f=15
- http://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine

Thanks for your attention (patience)

- Questions... ?

  If later, feel free to write me an e-mail:
  martin.pina@hanacke.net

hanácké.net

www.ipmedia.cz