cloudera®

Best Practices for Virtual Appliances

Sean Mackrory | ApacheCon North America, 2015

About Me

- Software Engineer @ Cloudera, Integration Engineering Team
- PMC Member @ Apache Bigtop (integration for the Hadoop ecosystem)
- PPMC Member @ Apache Sentry (incubating)

“Hardware and Software,
Engineered to Work Together”

—Oracle

Appliances

- Hardware & software, pre-integrated, pre-configured
- May seem counter-intuitive to programmers, because we value:
  - Modularity
  - Portability

cloudera

Appliances

- There are real benefits for a lot of use cases:
  - Less time and money than bespoke installations
  - Designed and supported as a single, holistic system

cloudera

“Virtual Hardware and Software, Engineered to Work Together”

—This Guy

Virtual Appliances

- At least a file-system image, software pre-installed and pre-configured
- Depending on the platform, it may also specify:
  - RAM, CPU and other hardware
  - Network configuration

Virtual Appliances

- Virtual Appliances also have benefits:
  - Tested and distributed as (mostly*) an entire system
  - (Mostly*) requires no setup or external dependencies

* The most important thing to remember about my talk

cloudera

Use Cases

- Training environment
- Testing & development platform
- Distribution artifact
- Reference installation
- Demonstrations

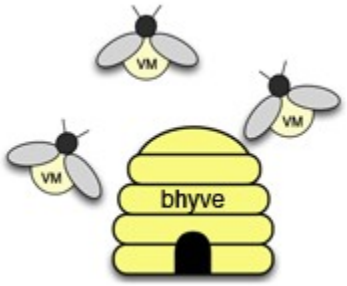
cloudera

Platforms

Hypervisors, Operating Systems and Tools

cloudera

Hypervisors



vmware

vmware

Type 1

Hypervisor runs in kernelspace.

Guests run their own kernel.

Type 2

Hypervisor runs in userspace.

Guests run their own kernel.

Type 3 (I'm coining this term)

Hypervisor partitions host OS.

Guests run their own userspace.

Other Platforms

- Live media
- Hardware emulators
- Cloud platforms

cloudera

Host Systems

- Can you store large disk images? (more on this later)
- Can you provide sufficient RAM, CPU, disk? Host 64-bit guests?
- Are Intel VT-X / AMD-VT extensions supported / enabled?
- Be aware of OSs designed to be hosts: SmartOS, CoreOS, etc.

Guest Operating Systems

- A free-as-in-libre OS can be redistributed ⇒ UNIX-like
- Linux is very common: CentOS, Ubuntu, minimalist distros, etc.
- Also consider BSD / Solaris variants:
  - They emphasize the whole system, but are as widely supported

OS^{V}, from Cloudius Systems

- BSD-licensed with POSIX-like API / Linux system calls – but no fork()
- Single process in kernel-space (can be an OS^{V}-optimized JVM)
  - No spin-locks, time-sharing, etc.

OSv, from Cloudius Systems

- OSv-optimized Memcached outperforms conventional install 3.9x
- Common Redis operations perform 80% better
- Capstan images add 12-20 MB and 3s of build time to your application
- "Hello, world!" boots, runs and shuts down in less than a second

Tools: Packer

- Packer (packer.io), from Hashicorp
  - Actively developed and very general-purpose
  - A wide range of “Builders” to target most common platforms
  - “Provisioners” allow you to reuse infrastructure code (shell, chef, etc.)

Tools: SUSE Studio

- SUSE Studio (susestudio.com)
  - Web application for building Linux images
  - Open-source back-end: KIWI (en.opensuse.org/Portal:KIWI)

Tools: Serverspec

- Serverspec (serverspec.org)
  - Testing for infrastructure software (and virtual appliances)

Virtual Hardware

Swap space, networks, CPUs

cloudera

Swap Partitions

- Swapping allows you to get by with less RAM, but…
- Swapping too aggressively will kill performance, so…
- Use a swap partition, but set swappiness as low as possible!
  - Don't rely on the host to swap your RAM for you

cloudera

Swappiness in the Linux Kernel

Value	< 2.6.32-303	≥ 2.6.32-303
0	Last resort	Never swap
1	**Low swappiness**	**Last resort**
100	Maximum	Maximum

Virtual Network Adapters: NAT

- Uses host as proxy, connections can only be opened from inside
  - Very portable
  - For client-only appliances

cloudera

Virtual Network Adapters: Bridged

- VM appears to your network as a peer of your host
  - Very portable
  - Very flexible
  - Raises security concerns

cloudera

Virtual Network Adapters: Host-only

- Can communicate with host (and maybe other VMs)
  - Secure and flexible
  - No Internet
  - Can be very host-dependent

cloudera

Virtual Networks: Port Forwarding

- VirtualBox embeds host→guest port forwarding in the appliance configuration
- Allows users to type 'localhost' in their own browser but connect to the VM
  - Even a NAT'd VM!
- Be aware of how addresses may interpreted by different clients

Virtual Networks: IP Resolution

- **`http://10.0.2.15/index.html`**: NAT'd guest only
- **`http://127.0.0.1/index.html`**: guest or host w/ port forwarding
- **`http://192.168.0.3/index.html`**: everywhere w/ bridged

Virtual Networks: Hostnames and DNS

- Some software requires a consistent hostname – this can be a problem
  - Configuring statically means you can't “seed” clusters
  - Randomize hostname at boot? (slow boot, inconsistent hostnames)

Virtual Networks: Hostnames and DNS

- Will users have to do anything special on their clients?
  - Add it to /etc/hosts (or equivalent), or DNS
- On Docker, users may have to specify --hostname=

cloudera

Virtual CPUs

- Some instructions not available, “cores” vs “cpus”
- Specify the minimum required to run things sufficiently well
- Encourage users to increase this / require it for certain options
- Virtual CPUs do not have all the same instructions! (esp. virtualization)

cloudera

Other Hardware

- Devices may not always be the same across similar hypervisors!

cloudera

Polishing and Publishing

Hypervisor Tools and Preparing Disks

cloudera

Hypervisor Tools

- Host↔Guest integrations
  - Copy / paste, drag & drop
  - Cursor capture / desktop integration
- Shrinking disks

cloudera

VirtualBox Guest Additions

- Kernel headers and running kernel must match

```
yum install -y dkms gcc kernel-devel make bzip2
mkdir /media/cdrom
mount -r /dev/sr0 /media/cdrom # may be sr1 or other
(cd /media/cdrom && env KERN_DIR=
   /usr/src/kernels/`uname -r` sh
   ./VBoxLinuxAdditions.run)
```

cloudera

VMWare Tools

- License requires that you use VMWare to build your appliance (Packer does)

```
echo > /etc/yum.repos.d/mware-tools.repo <<EOF
[vmware-tools]
name=VMWare Tools
baseurl=http://packages.vmware.com/tools/esx/latest/rhel6/x86_64
gpgkey=http://packages.vmware.com/tools/keys/VMWARE-PACKAGING-GPG-RSA-KEY.pub
gpgcheck = 1
EOF
yum install -y vmware-tools-*
```

cloudera

Clean Up

- VMs don't need firmware and many other common packages
  - e.g. consider using redhat-lsb-core instead of redhat-lsb
- Delete caches and log files that get written during setup
  - e.g. .bash_history, /var/yum/cache, etc.

Zeroing Disks

- Minimizes a copy-on-write FS and enables better compression
- VMWare tools: `vmware-toolbox-cmd disk wipe /`
- Everything else:

```
cat /dev/zero > zero.fill
sync; sleep 1; sync
rm -f zero.fill
```

cloudera

Defragmentation (Back to the 90's!)

- Defragments the copy-on-write device, not the file system

- VirtualBox: `VBoxManage modifyhd *.vdi --compact # host`
- VMWare: `vmware-toolbox-cmd disk shrink / # guest`
- QCOW2: `qemu-img convert -O qcow2 *.raw output.qcow2 # host`
  - Packer does this for you - better

Sharding

- Some file systems can't handle files as large as many VMs
  - FAT32 is still common on flash drives: has a limit of 4 GB
  - VMWare provides the option to shard the disk into 2 GB chunks

Compression

- Some archive / (de)compression tools can't handle big files either
  - Large tar.gz and zip files are not portable between implementations
  - I recommend using 7-zip (at least the tool, but also the format)

Other Thoughts

Suggestions and Common Gotchas

cloudera

Interfaces: Embedded Web UI

- Embed a web interface with tutorials, resources…
- Buttons for common options, etc.
- Plan on not having Internet access / port forwarding

Interfaces: Desktop Environment

- Dependent on networking, hitting a web UI may not be ideal
- Some tasks are not suited to CLI or a web UI
- A desktop environment is heavy, but is always useful
  - Consider using VNC, or SSH X-Forwarding for other platforms

Mirror All The Things!

- Be able to recreate the entire system sans-Internet
  - Third-party tools may disappear or change without warning
  - You may want to recreate old versions
- Archives of everything: your own software, all dependencies

Mirror All The Things!

- Mirrors also decrease build time: downloading a whole OS is a big deal
- Lock in one consistent version with OS install media + package repositories
- You may want to uninstall your mirrors before publishing!

Beware Reboots!

- Beware of settings that do not persist across reboots!
  - Swappiness, SELinux
- Upgrading the kernel during a build doesn't apply until a reboot
- If you have a first-boot procedure, make sure subsequent boots work well!

Centralizing Distributed Systems

- Distributed systems are hard: partial failure
- Centralize a distributed system: partial failure = total failure
- Reboots appear data-center wide, suspends make time appear to stop

cloudera

Centralizing Distributed Systems

- In general, plan for such weirdness
- Run an NTP daemon (also required for using time-based cryptography)
- Consider having a “VM” mode to work around exceptional behavior

Closing Thoughts

- The underlying platform really makes a difference
- Collect feedback from a broad spectrum of potential users
- Test, document, automate (like you always do… right?)
- Simply: projects should provide virtual appliances

cloudera

Thank you

mackrorysd@apache.org
@SeanMackrory