

# SUSE® Virtualization

## Best Practices and Troubleshooting

**Mike Latimer**

Software Engineer  
mlatimer@suse.com

**Jim Fehlig**

Software Engineer  
jfehlig@suse.com



# Agenda

- Best Practices
  - General virtualization guidelines
  - CPU and memory tuning
  - Disk and network device consideration
  - Xen specific considerations
- Troubleshooting
  - Important log files and debug levels
  - Supportconfig
  - Serial consoles and kernel dumps

# Best Practices in Virtualization

# General Virtualization Guidelines

- Minimize software installed on the host
  - Reduces resources
  - Reduces security risks/increases availability
- Synchronize time
  - Use NTP to synchronize time on the host AND guests
    - Guests should retrieve time from the host, or from common time source
- Avoid over-allocating resources to guests
- Use paravirtual drivers for better performance

# Virtualization Host Configuration

- SLES<sup>®</sup> 12 Installation

- Default configuration may not be ideal for host server:
  - Extra disk space (>40GB) added to /home partition
    - VM Images are stored in /var/lib/{kvm|libvirt|xen}/images
  - BTRFS Filesystem
    - Snapshots are enabled
      - Generally not useful to snapshot VM image files

- Host resource requirements

- Remember host requirements when sizing machine
  - Memory, disk space, CPU, etc...

# Virtualization Host Configuration

- Network

- Use multiple networks to avoid congestion

- admin, storage, live migration, ...

- May require using `arp_filter` to prevent ARP flux

- `echo 1 > /proc/sys/net/ipv4/conf/arp_filter`

- Same MTU in all devices to avoid fragmentation

# CPU

- Avoid CPU overcommit
- Scheduler
- vCPU model and features
  - Normalize to allow migration among heterogeneous hosts
- vCPU topology
  - Multiple sockets with a single core and thread generally give best performance
- vCPU Pinning
  - Constrain vCPU threads to a NUMA node

# Memory

- Hugepages
- HugeTLB
- Policy for allocation in NUMA topology
  - strict
  - interleave
  - preferred



# NUMA Considerations

- Potentially huge impact on performance
- Consider host topology when sizing guests
- Avoid allocating VM memory across NUMA nodes
- Prevent vCPUs from floating across NUMA nodes

# Block Devices -vs- Image Files?

- Block devices
  - Better performance
  - Use “standard” tools for administration/disk modification
  - Accessible from host (pro and con)
- Image Files
  - Easier system management
    - Easily move, clone, backup domains
  - Comprehensive toolkit (guestfs) for image manipulation
  - Reduce overhead through sparse files

# Image Formats

- raw
  - Most common format
  - Historically, best performance
- qcow2
  - Required for snapshot support in libvirt + tools
  - Improved performance and stability
- qed
  - Next generation qcow

# Image Formats

- vhd/vhdx
  - Also known as 'vpc'
- vmdk
  - Performance
  - Suggest converting to raw or qcow2
    - `qemu-img convert -f vmdk -O qcow2 img.vmdk img.qcow2`

# Disk Cache Modes

- writethrough
  - writes completed when data committed to storage
  - Default mode in Xen
  - Guest informed no writeback cache
- writeback
  - writes completed when placed in host page cache
  - Default mode in KVM
  - Guest flush commands honored

# Disk Cache Modes

- **directsync**
  - Writes reported completed only when committed to storage device
  - Useful for guests that don't send flush commands
- **None**
  - O\_DIRECT semantics
  - IO directly between KVM and storage device
  - Guest informed of writeback cache
- **Unsafe**
  - writes completed when placed in host page cache
  - Guest flush commands ignored

# Disk IO Modes (KVM Only)

- native
  - kernel asynchronous IO
- threads
  - host user-mode based threads
- default
  - 'threads' mode in SLES

# Network Devices

- Virtual NICs
  - virtio-net (KVM)
    - multi-queue option
  - vhost-net (KVM)
    - Default vNIC, best performance
  - netbk (Xen)
    - kernel threads vs tasklets
- Emulated NICs
  - e1000
    - Default and preferred emulated NIC
  - rtl8139



# Network Devices

- Shared physical NICs
  - SR-IOV
  - macvtap
- Physical NICs
  - PCI passthrough

# General Xen Guidelines

- Install Xen pattern during installation, or later using “Install Hypervisor and Tools” YaST module
  - This tool creates the required bridge automatically
- Configure grub to boot Xen kernel by default
- Used a fixed memory size for dom0
- Use either libvirt or libxl to manage domains
  - libvirt and libxl do not share domain states!

# dom0 memory

- Dom0 memory is adjusted through `/etc/default/grub`:
  - GRUB\_CMDLINE\_XEN\_DEFAULT
    - `dom0_mem=[min:<min_amt>],[max:<max_amt>],[<amt>]`
      - Hypervisor requires about 87MB of memory
      - Required dom0 memory could be between 512-4096MB
        - `GRUB_CMDLINE_XEN_DEFAULT="dom0_mem=4G"`
    - After any changes, update `grub.cfg` using:
      - `grub2-mkconfig -o /boot/grub2/grub.cfg`
    - Set `autoballoon="off"` in `/etc/xen/xl.conf`

# Troubleshooting Virtualization

# Xen Log Files

- libxl logs:

- /var/log/xen/\*
  - qemu-dm-<domain>.log, xl-<domain>.log
  - bootloader.log, vm-install, xen-hotplug
    - Process specific logs, often requiring debug log levels to be useful
      - Some logs require 'set -x' to be added to /etc/xen/scripts/\*

- libvirt logs:

- /var/log/libvirt/libxl
  - libxl-driver.log
  - <domain>.log

# Daemon and Hypervisor Logs

- View systemd journal for specific units/daemons:
  - journalctl [--follow] –unit xencommons.service
  - journalctl /usr/sbin/xenwatchdogd
- xl dmesg
  - Xen hypervisor logs

# Increasing Logging Levels

- Log levels are increased through xen parameters:
  - `loglvl=all`
    - Increased logging for Xen hypervisor
  - `guest_loglvl=all`
    - Increased logging for guest domain actions
  - Grub2 config:
    - Edit `/etc/default/grub`, then recreate `grub.cfg`:
      - `GRUB_CMDLINE_XEN_DEFAULT="loglvl=all guest_loglvl=all"`
      - `grub2-mkconfig -o /boot/grub/grub.cfg`

# Supportconfig and Virtualization

- Core files:

- basic-environment.txt

- Reports detected virtualization hypervisor

- Under some hypervisors (xen), subsequent general checks might be incomplete

- Hypervisor specific files:

- kvm.txt, xen.txt

- Both logs contain general information:

- RPM version/verification of important packages
        - Kernel, hardware, network details



# kvm.txt

- libvirt details
  - General libvirt details
    - Libvirt daemon status
    - KVM statistics
    - virsh version, capabilities, nodeinfo, etc...
  - Domain list and configurations
  - Conf and log files
    - /etc/libvirt/libvirtd.conf
    - Last 500 lines from /var/log/libvirt/qemu/domain.log

# xen.txt

- Daemon status
  - xencommons, xendomains and xen-watchdog daemons
  - grub/grub2 configuration (for xen.gz parameters)
- libvirt details
  - Domain list and configurations
- xl details
  - Domain list and configurations
  - Conf and Log files
    - /etc/xen/xl.conf, /etc/libvirt/libvirtd.conf
    - Last 500 lines from /var/log/xen/\*, /var/log/libvirt/libxl/\*
    - Output of `xl dmesg` and `xl info`

# Advanced Debugging Options

- Serial console

- GRUB\_CMDLINE\_XEN\_DEFAULT="loglvl=all guest\_loglvl=all console=com1 com1=115200,8n1"
- GRUB\_CMDLINE\_LINUX\_DEFAULT="console=ttyS0,115200"

- Debug keys

- xl debug keys h; xl dmesg
- xl debug keys q; xl dmesg

# Advanced Debugging Options (cont)

- Additional Xen debug tools:
  - xenstore-{ls,read,rm,watch,write}, xentrace, xentop
- Capturing Guest Logs
  - Capturing guest logs during triggered problem:
    - Connect to domain:
      - virsh console <domname>
    - Execute problem command
  - Capturing domain boot messages
    - xl create -c <VM config file>
    - virsh create <VM config file> --console

# Troubleshooting Installations

- virt-manager and virt-install logs:
  - Found in ~/.cache/virt-manager
- Debugging virt-manager:
  - virt-manager --no-fork
    - Sends messages directly to screen and log file
  - LIBVIRT\_DEBUG=1 virt-manager --no-fork
    - See libvirt messages in /var/log/messages
- Use 'xl' to rule out libvirt layer

# Troubleshooting libvirt

- Client side troubleshooting
  - LIBVIRT\_DEBUG=1
    - 1: debug, 2: info, 3: warning, 4: error
- Server side troubleshooting
  - /etc/libvirt/libvirtd.conf (restart libvirtd after changes)
    - log\_level = 1
    - log\_output = "1:file:/var/log/libvirtd.log"
    - log\_filters = "1:qemu 3:remote"

# Kernel Cores

- Host cores -vs- guest domain cores
  - Host cores are enabled through Kdump YaST module
    - For Xen dom0 cores, 'crashkernel=size@offset' should be added as a Xen hypervisor parameter
  - Guest cores require:
    - `<on_crash>[action]</on_crash>` tag
      - Possible coredump actions are:
        - `coredump-restart` Dump core, then restart the VM
        - `coredump-destroy` Dump core, then terminate the VM
    - Crashes are written to:
      - `/var/lib/libvirt/{libxl,qemu}/dump`
      - `/var/lib/xen/dump` (if using xl)

Have a lot of fun!  
[www.suse.com/virtualization](http://www.suse.com/virtualization)

Thank you.







## **Unpublished Work of SUSE LLC. All Rights Reserved.**

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

## **General Disclaimer**

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

