KVM on IBM System z

SHARE San Francisco, Session 13130

Feburary, 8th 2013

Tobias Dörkes, Finanz Informatik

finanz informatik

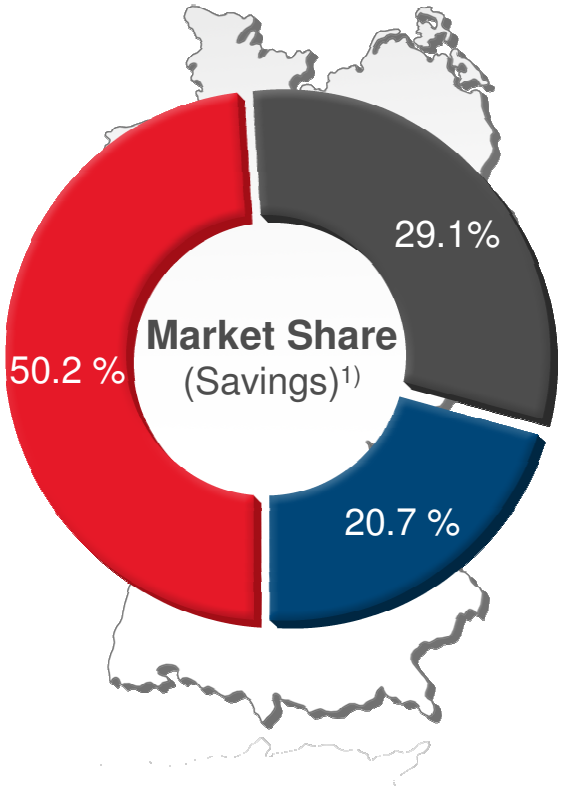
Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

The company serves a large part of the German retail banking market

Finanz Informatik – Company	
Revenue (in mill. €) (2011)	1,453
with saving banks	1,004
with state banks	229
Employees (full-time equivalents)	4,975

Customers	
Savings banks	423
State banks + DekaBank	9
State home loan banks	10
Accumulated balance sheet of supported savings banks (in bill. €) (2011)	1,046



Savings Banks Financial Group Credit Unions Private Banks, other

June 30th, 2012

1) Sources: DSGV (12/31/2011); German Federal Bank, Others.

finanz informatik

Significant scale can be achieved through bundling volume IT services

Supported financial institutions	
Branches of supported savings banks	15,250
Bank-specific employees of supported savings banks (2010)	192,301

Processing volumes	
Booked entries per annum (in bill.)	11
Supported accounts (in mill.)	127

Devices	
ATMs	24,029
Statement printers	15,812
Other self-service terminals	13,633
MIPS (Mainframe)	386,950
DASD Terabyte (Mainframe)	2,836
Windows-Server	11,904
UNIX-Server	2,564

finanz informatik

Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

Virtualization basics

What does hardware virtualization mean ?

- Virtualization means serving multiple operating systems in parallel on one physical hardware.
- Virtualized operating systems are running in so called »virtual machines« (VM)
- A very easy naming convention is to call a VM a »guest« and the serving system providing the virtualization capabilities a »host«.

Virtualization basics

Different types of hardware virtualization

- **Full virtualization**: Almost complete simulation of the actual hardware. This allows guests to run unmodified (e.g. QEMU, VMware workstation, z/VM)
- **Partial virtualization**: Not all of the actual hardware is simulated, only an address space is created. So the guest may need to be modified to run in this environment (eg. z/OS, Linux)
- **Paravirtualization**: No simulation at all. Guests are executed in their own domains. The VM is similar to the underlying hardware, but not the same - paravirtualized drivers are needed. (eg XEN, KVM)
- **Hardware-assisted virtualization**: The Host-CPU improves hardware virtualization efficiency by owning virtualization functions. (e.g. Intel-VT, AMD-V, System z)

Virtualization basics

Different types of hypervisors

A hypervisor or virtual machine manager creates and runs VMs. There are two different types of hypervisor implementation

- **Type 1 (Native, bare metal)**: The hypervisor runs directly on the host‘s hardware. No need for an additional OS to control and manage the VMs. (eg. PR/SM, z/VM, XenServer, VMware ESX, Hyper-V)

- **Type 2 (Hosted)**: The hypervisor runs as an application on top of a normal operating system. (eg. VMware workstation, VirtualBox)

And KVM? Personally, i would say it is a Type 1 hypervisor as it runs in the kernel and not on some second software level. But there are other opinions.

Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

What is KVM and how does fit with QEMU and Libvirt ?

KVM means »Kernel-based Virtual Machine«

KVM is no emulator itself.

KVM just provides an interface /dev/kvm to set up VMs.

KVM is a Linux kernel module that allows a user space program access to the hardware virtualization features of various processors, with which QEMU is able to offer virtualization for x86, PowerPC, and S/390 guests. When the target architecture is the same as the host architecture, QEMU can make use of KVM particular features, such as acceleration.

Wikipedia at http://en.wikipedia.org/wiki/Qemu

What is KVM and how does fit with QEMU and Libvirt ?

QEMU stands for »**Quick EMUlator**« and is a processor emulator that relies on dynamic binary translation to achieve a reasonable speed while being easy to port to new host CPU architectures.

Wikipedia at http://en.wikipedia.org/wiki/Qemu

QEMU emulates:

- CPUs, even for different architectures.
- various hardware components needed to create a VM (network card, storage, …)

QEMU does I/O, KVM does CPU, memory and interrupt controller.
QEMU uses KVM as an accelerator to access hardware features.

What is KVM and how does fit with QEMU and Libvirt ?

libvirt is an open source API, daemon and management tool for managing platform virtualization. It can be used to manage Linux KVM, Xen, VMware ESX, QEMU and other virtualization technologies. These APIs are widely used in Orchestration Layer for Hypervisors in the development of a cloud based solution.

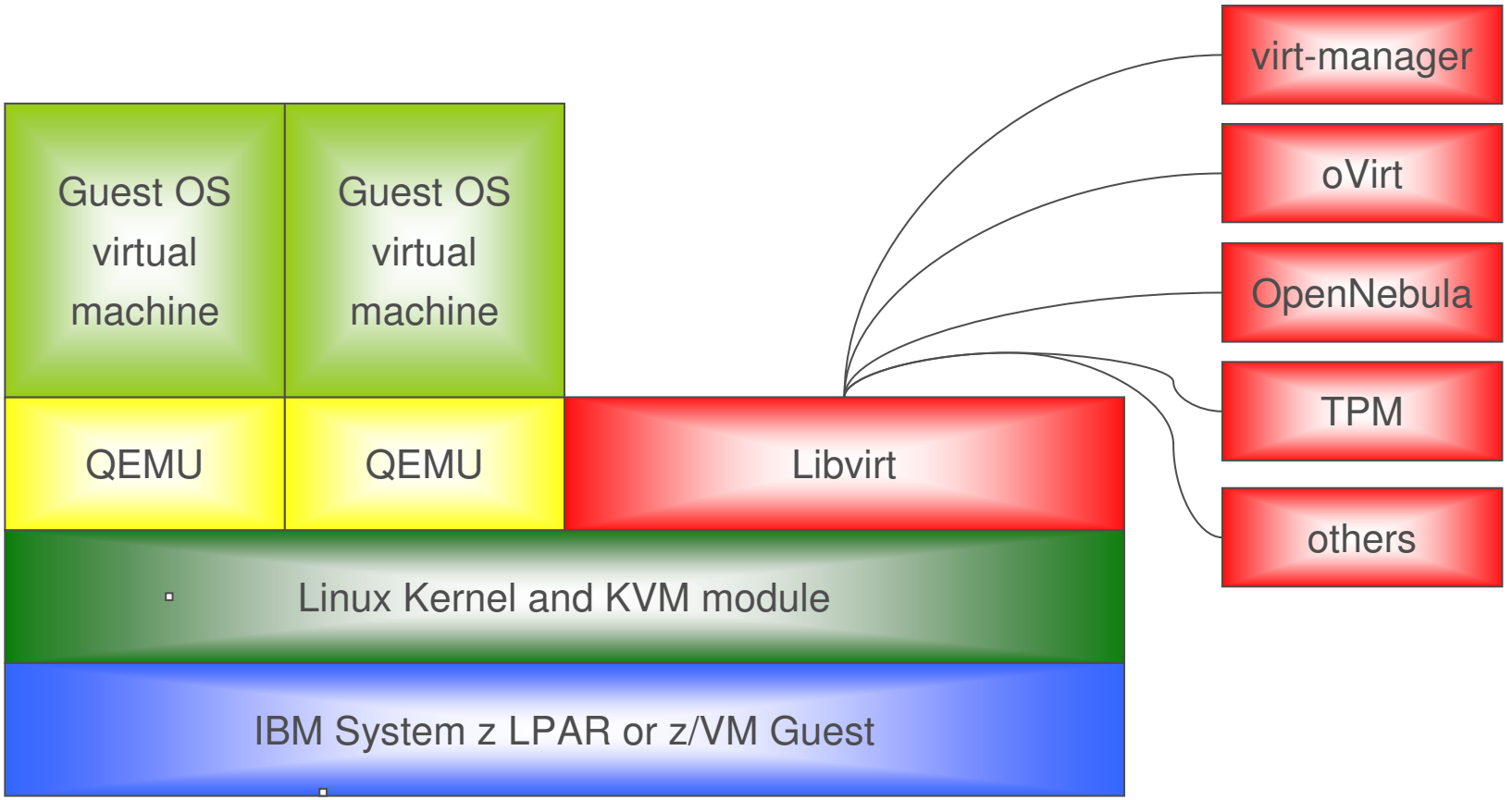
Wikipedia at http://en.wikipedia.org/wiki/Libvirt

libvirt provides:

- A directory for configuration data and operational state of VMs
- The **libvirtd daemon** is the server side daemon component of the libvirt virtualization management system. (man page)
  It runs on host servers and provides remote management services.
- The **virsh command shell** is the main interface for managing guest domains. It is an interactive shell and batch scriptable tool. (man page)

What is KVM and how does fit with QEMU and Libvirt ?

The big picture



finanz informatik

Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

Current state of implemenation

These functions are implemented today:

- virtio-s390 is upstream
  virtio-ccw for s390 device discovery is under development
- direct kernel ipl necessary, s390 devices for zipl needed
- network access only stable for routed layer3 networks
  macvtap driver is not stable on s390
- storage over commitment using virtual disk images
- live migration is implemented in KVM, but not tested on s390
  live migration on KVM includes storage if you like

Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

Prerequisites for running KVM on IBM System z (s390x)

There are just a few requirements for running KVM on System z

- IBM System z9 to get hardware support for KVM

Linux distributions:

- **SLES 11 SP2** has kernel module kvm, but qemu is missing completely and libvirt is missing functionality. SUSE indicated to have a kvm s390x technical preview for **SLES 11 SP3**. For a definitve statement, please check with SUSE (here on SHARE etc)
- **Fedora 18** has KVM, QEMU and libvirt with KVM officially enabled
- **RHEL** has currently no support for KVM.

Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

Building QEMU and Libvirt on SLES 11 SP2 using SDK

Install SLES 11 SP2 and corresponding SDK

Additional pattern needed:

- C/C++ Compiler and Tools

Make sure following packages are installed

- QEMU: glib2-devel zlib-devel
- libvirt: libxml2-devel libgnutls-devel libyajl-devel device-mapper-devel python-devel dnsmasq

Building QEMU and Libvirt on SLES 11 SP2 using SDK

Download/Install libvirt and qemu tarballs

http://libvirt.org/sources/libvirt-1.0.0.tar.gz

http://wiki.qemu.org/download/qemu-1.3.0.tar.bz2

Compile and install libvirt and qemu

- QEMU

```
root@kvmhost:~> tar xjf qemu-1.3.0.tar.bz2 && cd qemu-1.3.0
root@kvmhost:~> ./configure --prefix=/usr --target-list=s390x-softmmu && make && make
install
```

- libvirt

```
root@kvmhost:~> tar xzf libvirt-1.0.0.tar.gz && cd libvirt-1.0.0
root@kvmhost:~> env CPPFLAGS='-I/usr/src/linux/include/' ./configure --prefix=/usr \
--with-macvtap
root@kvmhost:~> make && make install
```

Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

Installation and enablement of KVM, QEMU and libvirt

Kernel parmline needs an additional parameter

Add 'switch_amode=on' to kernel cmdline

This option allows to switch the addressing modes of kernel and user space. switch_amode is disabled by default because page table walk on pre z9 hardware has negative performance impact.

Useful initrd modification

Add kvm to INITRD_MODULES on host system. So kvm is loaded during boot process.

Installation and enablement of KVM, QEMU and libvirt

KVM activation

```
root@kvmhost:~> modprobe kvm
```

libvirtd activation

```
root@kvmhost:~> /usr/sbin/libvirtd -d -v
```

This daemon runs on host servers and performs required management tasks for virtualized guests.

This includes activities such as starting, stopping and migrating guests between host servers, configuring and manipulating networking, and managing storage for use by guests.

(Manual Page)

Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

Installation verification of KVM, QEMU and libvirt

1. KVM module loaded

```
root@kvmhost:~> lsmod | grep kvm
kvm                   114285  0
```

2. KVM support in QEMU enabled

```
root@kvmhost:~> qemu-system-s390x -monitor stdio -machine type=s390-virtio,accel=kvm
VNC server running on `127.0.0.1:5900'
QEMU 1.2.0 monitor - type 'help' for more information
(qemu) info kvm
kvm support: enabled
(qemu)
```

3. libvirt information

```
root@kvmhost:~> virsh version
Compiled against library: libvirt 1.0.0
Using library: libvirt 1.0.0
Using API: QEMU 1.0.0
```

Boot an preinstalled KVM guest

Wether you use libvirt or any other virtualization API to operate your KVM guests, at the end of the day there is always a **QEMU command** initiated to create the VM.

```
root@kvmhost:~> qemu-system-s390x -machine type=s390-virtio,accel=kvm
-drive file=testvm1.img -m 128
-append "root=/dev/disk/by-path/virtio-pci-virtio1-part2"
-kernel /boot/image -initrd /boot/initrd
```

- Invoke binary qemu-system-s390x: machine type s390-virtio and kvm accelerator.
- Attach disk image testvm1.img and 128M RAM to KVM guest.
- Specify rootfs as kernel parameter

Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

Introduction to virsh comands

A VMs – also called domain - life-cycle using virsh comands

```
root@kvmhost:~> virsh define testvm3.xml
Domain testvm3 defined from testvm3.xml

root@kvmhost:~> virsh start testvm3
Domain testvm3 started

root@kvmhost:~> virsh console testvm3
entering console, to leave enter <STRG>+"+"

root@kvmhost:~> virsh suspend testvm3
Domain testvm3 suspended

root@kvmhost:~> virsh resume testvm3
Domain testvm3 resumed

root@kvmhost:~> virsh shutdown testvm3
Domain testvm3 is being shutdown

root@kvmhost:~> virsh undefine testvm3
Domain testvm3 has been undefined
```

There are many more commands in virsh.

virsh supports

- network and storage administration.
- snapshot management
- autostart
- device management
- collecting information about domains

Installing SLES 11 SP2 as guest
Create XML file for a new routed network

```
<network>
  <name>net100</name>
  <forward dev='eth0' mode='route'>
    <interface dev='eth0'/>
  </forward>
  <bridge name='virbr1' stp='on' delay='0' />
  <ip address='192.168.100.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.100.128' end='192.168.100.254' />
    </dhcp>
  </ip>
</network>
```

Virsh uses XML to define networks

- name of network
- forward mode
- bridge name
- address range

```
root@kvmhost:~> virsh net-define net100.xml
Network net100 defined from net100.xml

root@kvmhost:~> virsh net-start net100
Network net100 started
```

Add network to libvirt configuration

- define
- start

Installing SLES 11 SP2 as guest
Create new harddisk image

qemu-img - QEMU disk image utility

- qemu-img allows you to create, convert and modify images offline. It can handle all image formats supported by QEMU. (Manual Page)
- qemu-img supports many image formats:
  raw, cow, cow2, qcow2, vdi, vmdk and others
- qemu-img supports resizing of disk images

```
root@kvmhost:~> qemu-img create -f raw /var/lib/libvirt/images/testvm3.img 2G
root@kvmhost:~> qemu-img info /var/lib/libvirt/images/testvm3.img
image: /var/lib/libvirt/images/testvm3.img
file format: raw
virtual size: 2.0G (2147483648 bytes)
disk size: 1.1G
```

Installing SLES 11 SP2 as guest
Create XML file

```
<domain type='kvm'>
 <name>testvm3</name>
 <memory>512000</memory>
 <os>
  <type arch='s390x'>hvm</type>
  <kernel>/var/lib/libvirt/boot/install.kernel</kernel>
  <initrd>/var/lib/libvirt/boot/install.initrd</initrd>
  <cmdline>ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc
TERM=dumb manual=1</cmdline>
 </os>
 <devices>
  <disk type='file' device='disk'>
   <source file='/var/lib/libvirt/images/testvm3.img'/>
   <target dev='vda' bus='virtio'/>
  </disk>
  <console type='pty'>
   <target type='virtio'/>
  </console>
  <interface type='network'>
   <source network='net100'/>
   <model type='virtio'/>
  </interface>
 </devices>
</domain>
```

Virsh uses XML to define VMs

- name of VM
- memory
- architecture
- kernel, initrd and parmline for SLES11 installer
- storage
- console
- networking options:
  Layer2 via macvtap
  Layer3 via routed network

Installing SLES 11 SP2 as guest
Start the VM and perform stage 1 install

```
root@kvmhost:~> sudo virsh define testvm3.xml
Domain testvm3 defined from testvm3.xml

root@kvmhost:~> sudo virsh start testvm3
Domain testvm3 started

root@kvmhost:~> sudo virsh console testvm3
Connected to domain testvm3
Escape character is ^]

>>> SUSE Linux Enterprise Server 11 installation program v3.3.81 (c)
1996-2010 SUSE Linux Products GmbH <<<

Select the display type.
1) X11
2) VNC
3) SSH
4) ASCII Console
> 3

Enter your temporary SSH password
>
root@kvmhost :~> ssh root@192.168.100.239
```

Steps to install SLES11 SP2 via network

- create VM
- start VM
- enter VM console
- configure installer system (CTRL+“+“ to leave console)
- start ssh session
- start YaST and perform normal installation process

Installing SLES 11 SP2 as guest
Copy installed kernel and initrd to KVM host

Currently KVM/QEMU on z needs so called direct IPL after installation. Which means kernel, initrd and parmline are placed on the KVM host. They will be passed by libvirt to the VM via command line.

- load nbd driver (QEMU Disk Network Block Device Server)
- mount rootfs partition
- copy files
- cleanup

```
root@kvmhost:~> modprobe nbd max_part=63
root@kvmhost:~> qemu-nbd -c /dev/nbd0 /var/lib/libvirt/images/testvm3.img
root@kvmhost:~> mount /dev/nbd0p2 /mnt/
root@kvmhost:~> cp /mnt/boot/image /var/lib/libvirt/boot/boot.kernel
root@kvmhost:~> cp /mnt/boot/image /var/lib/libvirt/boot/boot.kernel
root@kvmhost:~> umount /mnt/
root@kvmhost:~> killall qemu-nbd
```

Installing SLES 11 SP2 as guest
Perform stage 2 install

```
root@kvmhost:~> sudo virsh edit testvm3
<os>
 <type arch='s390x'>hvm</type>
 <kernel>/var/lib/libvirt/boot/boot.kernel</kernel>
 <initrd>/var/lib/libvirt/boot/boot.initrd</initrd>
 <cmdline>root=/dev/disk/by-path/virtio-pci-virtio1-part2</cmdline>
 <boot dev='hd'/>
</os>

root@kvmhost :~> sudo virsh start testvm3
Domain testvm3 started

root@kvmhost :~> ssh root@192.168.100.239
Password:
linux:~ # /usr/lib/YaST2/startup/YaST2.ssh


Continue with booting ...

You can login with the (new) root password or the
newly created user account in a few seconds ...

linux:~ #
```

Tell the VM about kernel, initrd and parmline

- edit XML to change OS section
- start the VM
- start ssh session
- complete YaST stage 2 installer

Installing SLES 11 SP2 as guest
Remove kernel, initrd and parmline from xml

```
linux:~ # vi /etc/zipl.conf
linux:~ # zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'SLES11_SP2' (default)
Adding #2: IPL section 'FailsafeV1'
Adding #3: IPL section 'ipl'
Preparing boot device: vda.
Done.
```

Make sure console=hvc0

```
root@kvmhost :~> virsh shutdown testvm3
Domain testvm3 started
root@kvmhost:~> virsh edit testvm3
<os>
 <type arch='s390x'>hvm</type>
 <kernel>/var/lib/libvirt/boot/install.kernel</kernel>
 <initrd>/var/lib/libvirt/boot/install.initrd</initrd>
 <cmdline>ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc
TERM=dumb manual=1 console=hvc0</cmdline>
 <boot dev='hd'/>
</os>
root@kvmhost :~> virsh start testvm3
Domain testvm3 started
```

After the bootloader is written, the kernel, initrd and parmline can be removed from the xml file

/proc/sysinfo

```
LPAR Number:            14
LPAR Characteristics:   Shared
LPAR Name:              VPT10E
LPAR Adjustment:        1000
LPAR CPUs Total:        4
LPAR CPUs Configured:   1
LPAR CPUs Standby:      0
LPAR CPUs Reserved:     3
LPAR CPUs Dedicated:    0
LPAR CPUs Shared:       1
```

```
VM00 Name:              KVMguest
VM00 Control Program:   KVM/Linux
VM00 Adjustment:        1000
VM00 CPUs Total:        1
VM00 CPUs Configured:   1
VM00 CPUs Standby:      0
VM00 CPUs Reserved:     0

VM01 Name:              LTZ0007
VM01 Control Program:   z/VM    6.2.0
VM01 Adjustment:        1000
VM01 CPUs Total:        1
VM01 CPUs Configured:   1
VM01 CPUs Standby:      0
VM01 CPUs Reserved:     0
```

/proc/sysinfo shows that the linux kernel is aware of what is going on.
LPAR -> z/VM guest -> KVM guest

Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

Additional tools

Filesystem services/tools

- kpartx to access RAW images
- qemu-nbd to access QCOW2 and other images
- libguestfs tools for accessing and modifying virtual machine disk images

Security

- sVirt in conjunction with SELinux/AppArmor

Networking options

- Layer2: Bridge, macvtap, OpenVswitch
- Layer3: NAT, Routing

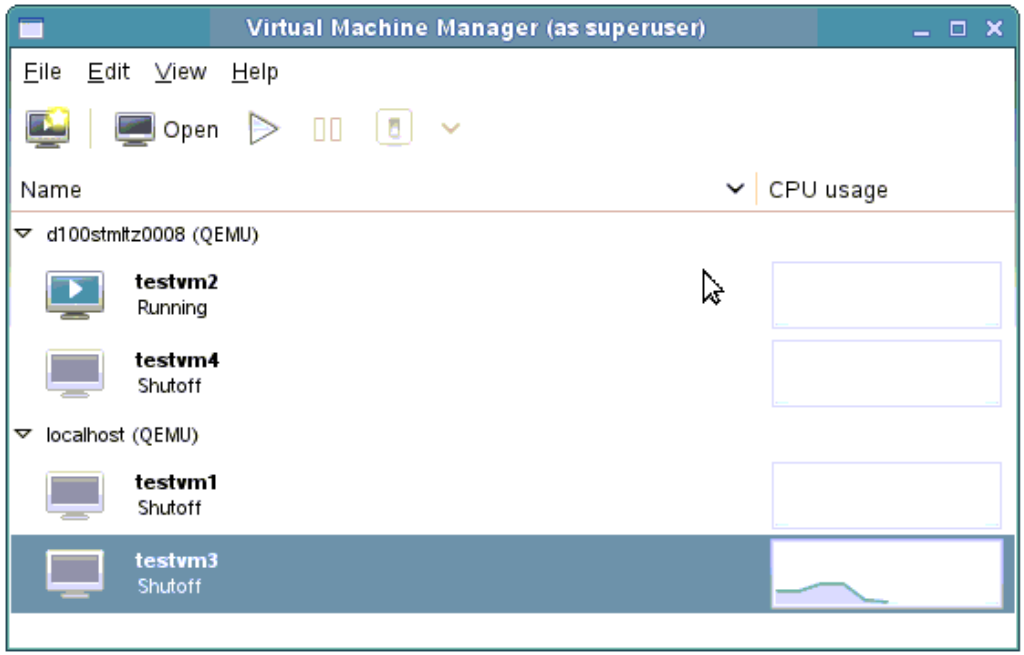
High availability

- Live migration including storage if you like

Agenda

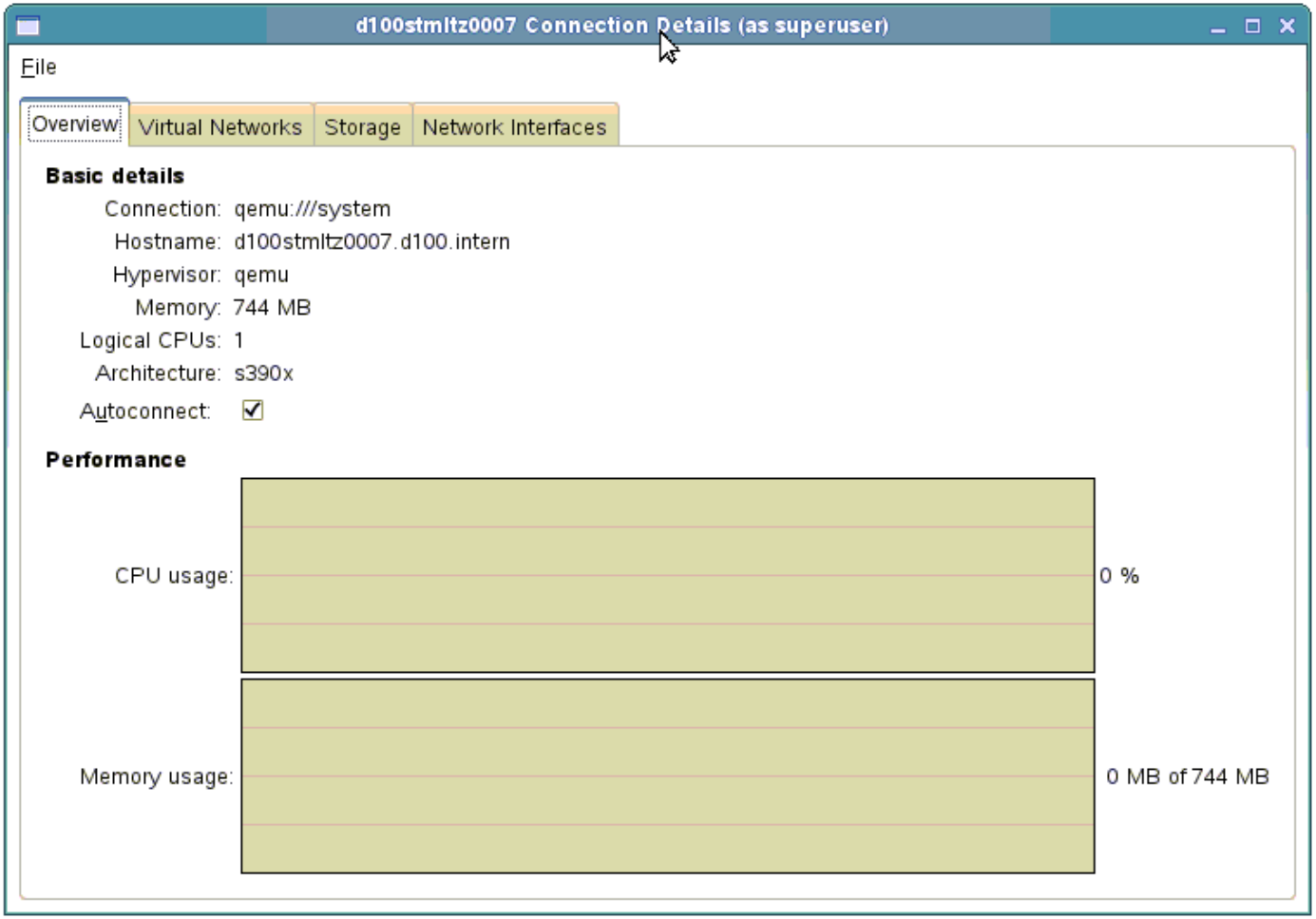
- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

Screenshots



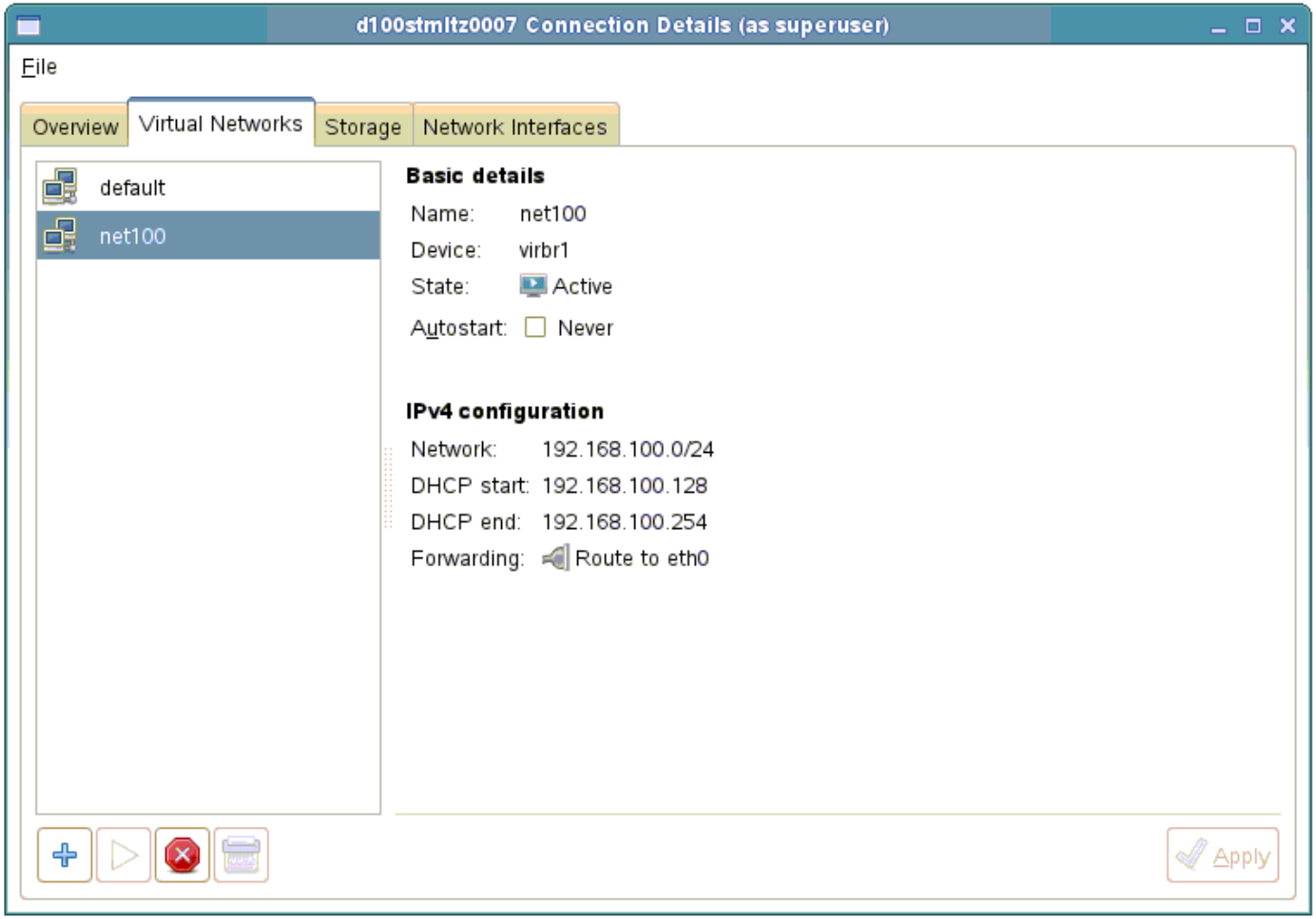
finanz informatik

Screenshots



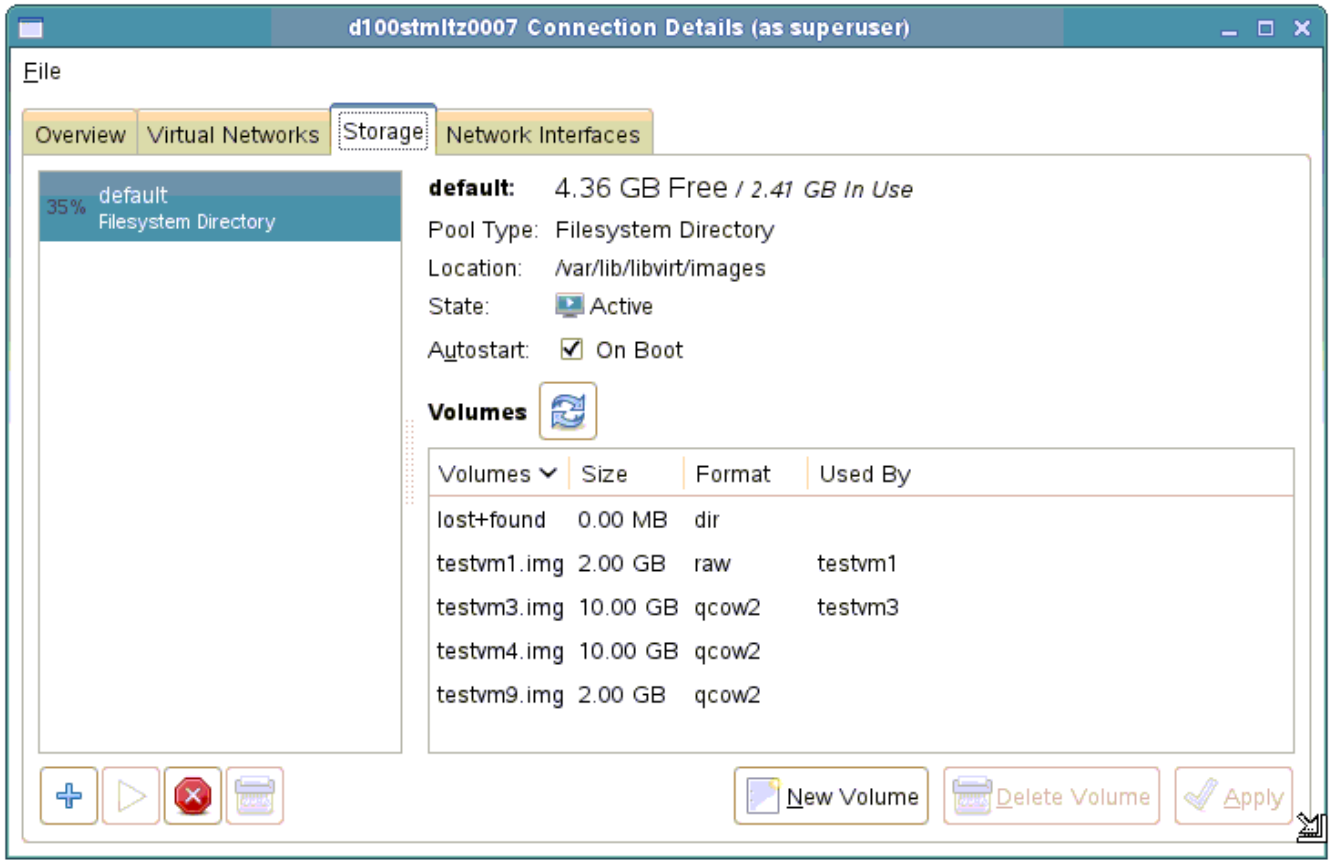
finanz informatik

Screenshots



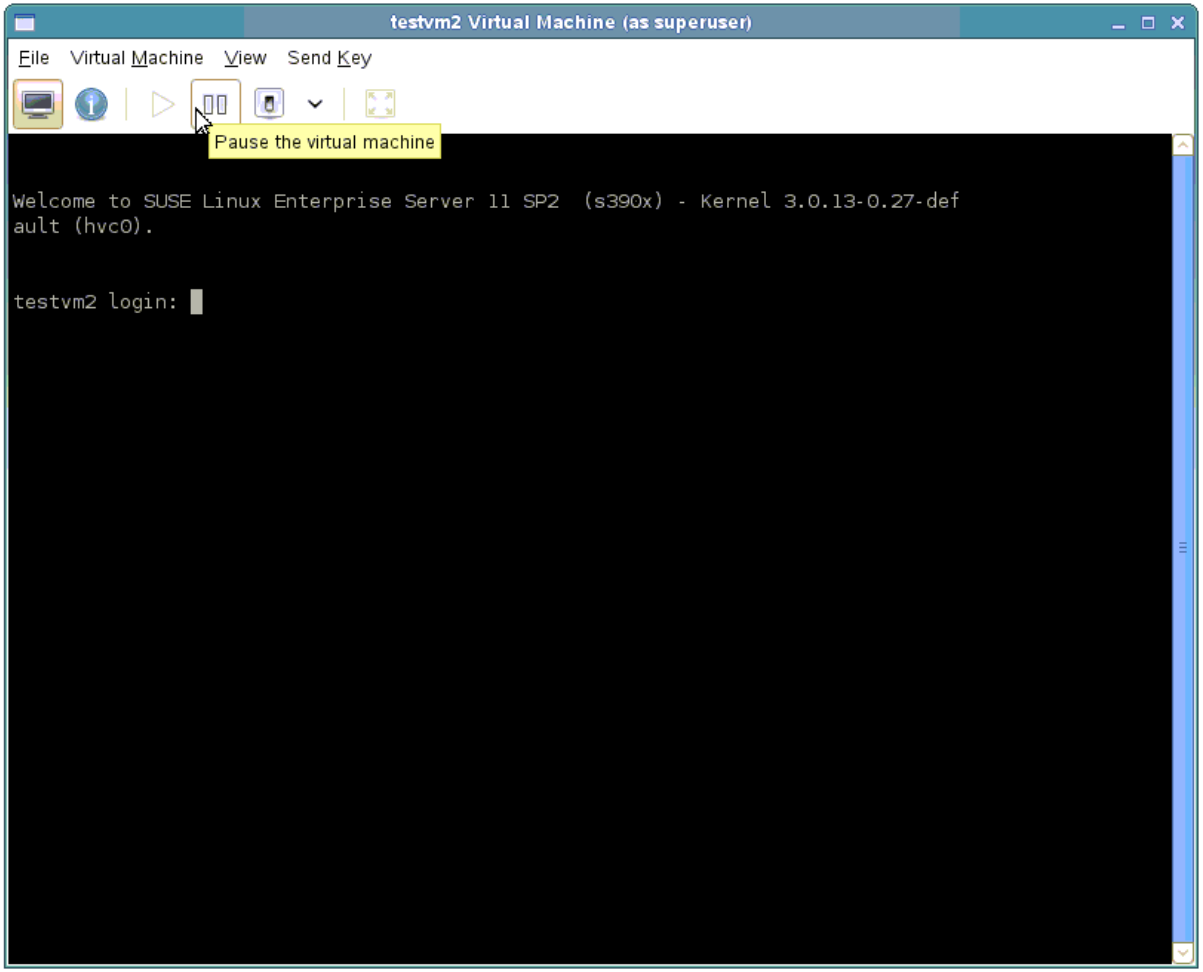
finanz informatik

Screenshots

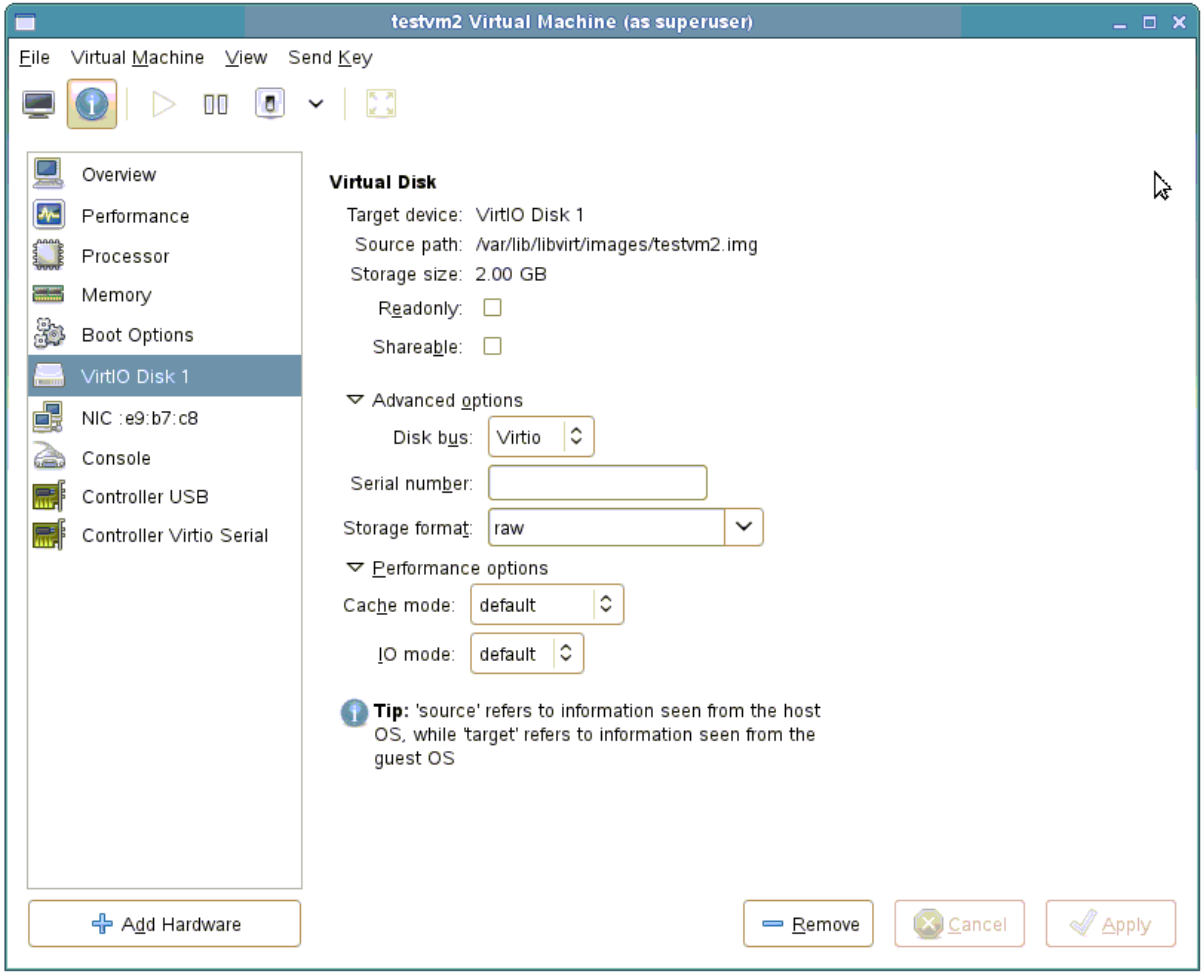


finanz informatik

Screenshots



Screenshots



finanz informatik

Agenda

- The IT Service Provider Finanz Informatik
- Virtualization basics
- What is KVM and how does fit with QEMU and Libvirt ?
- Current state of implementation
- Prerequisites for running KVM on IBM System z (s390x)
- Building QEMU and Libvirt on SLES 11 SP2 using SDK
- Installation and enablement of KVM, QEMU and libvirt
- Installation verification of KVM, QEMU and libvirt
- Installing SLES 11 SP2 as guest
- Additional tools
- Screenshots
- Summary

Summary

Pros

- Open Source alternative to z/VM for a linux-only environment.
- Easy to install, there will be RPMs in the distributions in the future.
- You can use any GUI that supports libvirt to administrate a z/penguin farm.
- Storage overcommitment possible

Cons

- Currently no official support by any vendor or distributor
- KVM has to prove that it can handle CPU and memory constraint environments.
- Lack of performance measurement tools with mainframe scope.

Our own opinion

- Useful and cheap alternative to z/VM in a linux only environment on IBM System z.
- It will take time to get KVM and QEMU ready for a production environment, but rome wasn‘t built in a day

Thank you all for your attention

Thank you !!!

Alexander Graf (SUSE), Viktor Mihajlovski (IBM),
Karl-Heinz Doppelfeld (FI), Jeremy Koch (FI)

Additional tools
kpartx

kpartx can be use to process raw images.

list partitions, add them to the mapper subsystem and mount single partitions to the filesystem.

```
root@kvmhost:~> kpartx -l /var/lib/libvirt/images/guest1.img
loop0p1 : 0 409600 /dev/loop0 63
loop0p2 : 0 10064717 /dev/loop0 409663
root@kvmhost:~> kpartx -a /var/lib/libvirt/images/guest1.img
root@kvmhost:~> mkdir /mnt/guest1
root@kvmhost:~> mount /dev/mapper/loop0p1 /mnt/guest1 -o loop,ro
root@kvmhost:~> umount /mnt/tmp
root@kvmhost:~> kpartx -d /var/lib/libvirt/images/guest1.img
```

Additional tools
libguestfs

libguestfs is a scriptable and secure library to access images without root privileges.

Libguestfs-Tools

virt-cat, virt-install, virt-rescue, virt-clone, virt-list-filesystems, virt-resize, virt-convert, virt-list-partitions, virt-tar, virt-df, virt-ls, virt-top, virt-edit, virt-make-fs, virt-viewer, virt-image, virt-manager, virt-win-reg, virt-inspector, virt-pki-validate, virt-xml-validate

```
root@kvmhost:~> virt-df -h ~/disk.img
Filesystem                                  Size    Used Available Use%
/home/rjones/disk.img:/dev/vda1             193.7M 21.6M    162.1M  12%
/home/rjones/disk.img:/dev/vg_f12x32/lv_root  5.2G  2.3G      2.6G  45%
```