

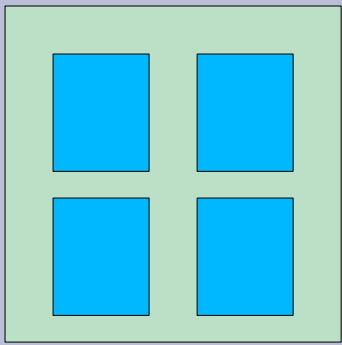
Live und in **F**ar**e**

Live Migration

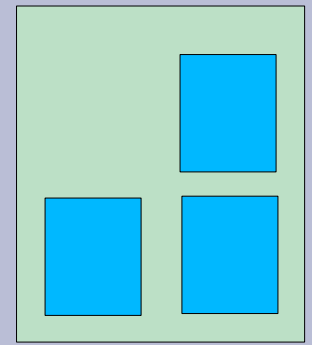
André Przywara
ap@amd64.org
CLT 2010

Agenda

- (Live) Migration explained (Why? Limits!)
- Xen and KVM usage
- Details
 - Memory synchronization
 - QEMU device state transfer
 - Host considerations (CPU features)
 - Cross Vendor Migration
- QEMU block device transfer
- Project Remus (Xen)
- Demo!



Guest Migration



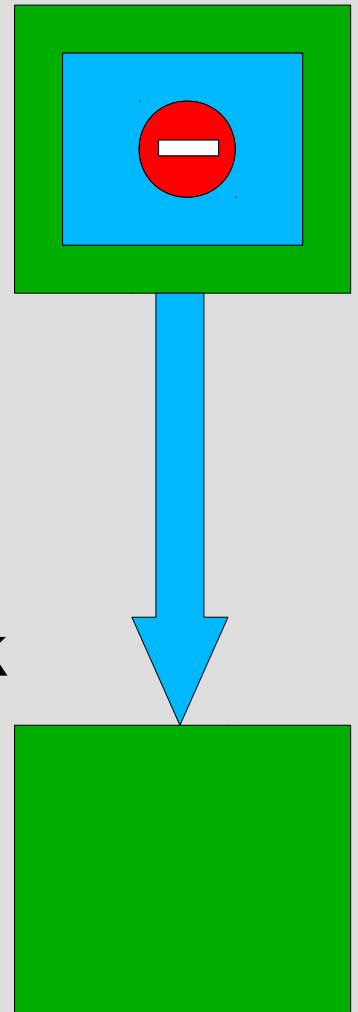
- move a *virtual machine* from one host to another
- offline:
 - power down the guest, copy files, restart
 - comparable to a reboot
- migration:
 - halt the guest, copy state, wake up again
 - minimal downtime
- live migration:
 - copy state in background, switch at one
 - (almost) no downtime at all

Reasons for migration

- Load balancing:
 - freeing loaded hosts in favor of idle ones
- Upgrade / update / planned downtime
 - migrate to a spare machine, rework the host, migrate back to the original one
- Roaming “eternal” desktop – Uptime, uptime!
 - desktop is running on a server, migrated to the respective client workstation
- Replacing older machines
- You name it!

How does it work?

- Host has full control over the guest
- Can read/write/protect memory
- Devices are (usually) also virtualized
- Host controls CPU usage
 - similar to OS vs. application
- Steps:
 - host de-schedules the guest
 - host copies memory content over network
 - host copies device state over network
 - old host signals new host to take over



Limits of migration

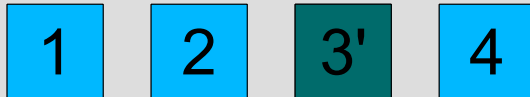
- disk images should be accessible
 - through a SAN, NAS, NFS
 - can also be copied / synced (DRBD)
- no downgrade of CPU features
 - maybe start with features disabled?
- No device pass-through
- Network connectivity must prevail
- Resources should match (memory, vCPUs)
- Matching software versions (devices!)

Xen / KVM usage

- Xen: via “xm” tool
 - `$ xm migrate <domid> <newhost>`
 - *xend* must be running on both sides
- KVM:
 - on target:
 - `$ qemu -incoming tcp:0:<port>`
 - on source: via QEMU monitor
 - `(qemu) migrate tcp:<host>:<port>`
 - Need to have the exact same guest parameters on the command line (management app!)

Memory synchronization

- Problem: transferring RAM image takes time
- e.g.: 1GB @ 40 MB/s = 25 sec
 - too long for live migration
- solution:
 - start copying (in background)
 - write protect already copied pages
 - on page fault: allow r/w again, mark page as *dirty*
 - repeat: copying dirty pages until
 - no more left
 - number of tries exhausted: halt guest and copy rest



QEMU device state transfer

- QEMU devices used for Xen and KVM
- each device has a VMStateDescription
 - describes the data that holds the complete state
- variables will be dumped to the stream
- contains version information (backward compatible)
- QEMU will iterate through all devices
 - sends the device name and instance number
 - executes a `pre_save` callback function
 - dumps the device' state to the stream (TCP)

QEMU device state dump

	QEMU magic				version			stage		section ID				device name		
00000000:	51	45	56	4d	00	00	00	03	01	00	00	00	01	05	62	6c
00000010:	6f	63	6b	00	00	00	00	00	00	00	01	00	00	00	00	00
00000020:	00	00	02	01	00	00	00	03	03	72	61	6d	00	00	00	00
00000030:	00	00	00	03	00	00	00	00	02	87	00	04	00	00	00	00
00000040:	00	00	00	08	01	00	00	00	53	ff	00	f0	c3	e2	00	f0
00000050:	53	ff	00	f0	53	ff	00	f0	53	ff	00	f0	53	ff	00	f0

version no.

QEVM.....b1

instance no.

ock.....

.....ram.....

.....

.....S.....

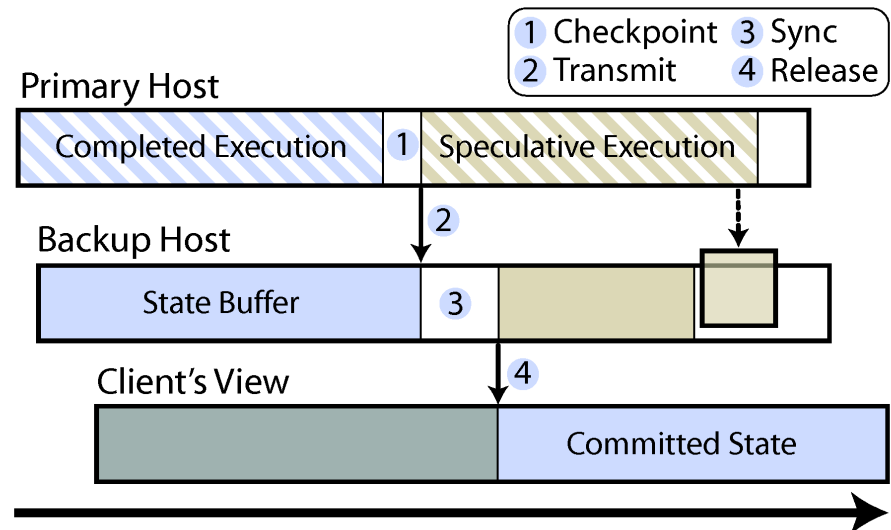
S...S...S...S...

QEMU block device transfer

- Recent QEMUs can transfer the block device
- No need for a shared storage
- `(qemu) migrate -b tcp:<host>:<port>`
- Can also migrate overlay only (-i)
- Uses same approach like RAM transfer
- Works like this:
 - Transfer data in chunks of 1 MB
 - Each chunk is preceded by a 64bit address
 - Allows gaps
 - Each chunk has the block device name in it

Project Remus (Xen)

- High availability using migration
- “Continuously” migrating the guest
- Avoids slowdown by snapshotting
- Only commits results when transmitted
- Snapshot frequency about every 25ms
- Running machine in the past
- part of Xen 4.0



Host considerations

- Applications and libraries rely on a consistent set of CPU features (like SSEx)
- CPU instruction set may change at migration
- no downgrade! (loss of a feature)
- upgrade can be hidden (CPUID masking)
- least common denominator in a migration pool dictates the feature set of all guests
- KVM: use `-cpu kvm64`
- migration pool should be well defined before starting the guest

Cross Vendor Migration

- Migrating from an Intel box to an AMD box (and vice versa ;-)
- allows for bigger migration pools
- avoids vendor lock in
- maps mostly to different CPU generations
- but subtle differences:
 - x87 FPU rounding on some instructions (e.g. for fsin, deprecated)
 - sysenter/syscall support in compat mode (emulation upstream)
 - slightly different guest state checks (fixed)
 - Model specific registers (MSRs) (fixed)
- Both Xen and KVM support it now!

Demo! Live! In Color!

- Using KVM (qemu-kvm 0.12.3, kernel 2.6.33)
- Migration between servers, using VNC
- Windows XP 32 guest with running Passmark stress test application
- Between Intel C2Q and AMD Opteron

Demo screenshot

BurnInTest - CPU - Maths

Executed	Verified
2997.2	2997.2
2878.1	2878.1
2656.0	2656.0
7646.8	7646.8
6432.2	6432.2
N/A	

BurnInTest - Network Test

Server:	Loopback
Packets sent:	953
Packets received:	953
Average delay:	0.18 ms
Max delay:	4.74 ms
Current delay:	0.10 ms
76240 bytes	
1.8 pkt/s	
0 (0.000%)	

BurnInTest - Video Memory Test

Total Video Memory: 2.62MB
Tested Video Memory: 0.75MB

STOP BIT

Memory (RAM) 3 4.918 Billion 0 No errors
2D Graphics 7 7605 0 No errors
Network 1 9 76240 0 No errors

```
aprzywar@muebarek:nfsimages/test6$ kvm -hda winxp32_qcow.img -m 1024M -vnc :0 -smp 1 -monitor stdio -usb -usbdevice tablet  
Could not open option rom 'vapic.bin': No such file or directory  
QEMU 0.12.3 monitor - type 'help' for more information  
(qemu) migrate tcp:tronje:4711  
(qemu) quit  
aprzywar@muebarek:nfsimages/test6$ kvm -hda winxp32_qcow.img -m 1024M -vnc :0 -smp 1 -monitor stdio -usb -usbdevice tablet -incoming tcp:0:4711  
Could not open option rom 'vapic.bin': No such file or directory  
QEMU 0.12.3 monitor - type 'help' for more information  
(qemu) □
```

BurnInTest - CPU - Maths

Executed	Verified
2146.2	2146.2
2023.7	2023.7
1874.4	1874.4
5221.3	5221.3
4358.9	4358.9
N/A	

BurnInTest - Network Test

Server:	Loopback
Packets sent:	764
Packets received:	764
Average delay:	0.20 ms
Max delay:	4.74 ms
Current delay:	0.32 ms
61120 bytes	
1.9 pkt/s	
0 (0.000%)	

BurnInTest - Video Memory Test

Total Video Memory: 2.62MB
Tested Video Memory: 0.75MB

STOP BIT

Memory (RAM) 2 4.051 Billion 0 No errors
2D Graphics 6 6285 0 No errors
Network 1 7 61120 0 No errors

```
aprzywar@tronje:nfsimages/test6$ kvm -hda winxp32_qcow.img -m 1024M -vnc :0 -smp 1 -monitor stdio -usb -usbdevice tablet -incoming tcp:0:4711  
Could not open option rom 'vapic.bin': No such file or directory  
QEMU 0.12.3 monitor - type 'help' for more information  
(qemu) migrate tcp:muebarek:4711  
(qemu) info migrate  
Migration status: completed  
(qemu) □
```

BurnInTest - CPU - Maths

Executed	Verified
2146.2	2146.2
2023.7	2023.7
1874.4	1874.4
5221.3	5221.3
4358.9	4358.9
N/A	

BurnInTest - Network Test

Server:	Loopback
Packets sent:	764
Packets received:	764
Average delay:	0.20 ms
Max delay:	4.74 ms
Current delay:	0.32 ms
61120 bytes	
1.9 pkt/s	
0 (0.000%)	

BurnInTest - Video Memory Test

Total Video Memory: 2.62MB
Tested Video Memory: 0.75MB

STOP BIT

Memory (RAM) 2 4.051 Billion 0 No errors
2D Graphics 6 6285 0 No errors
Network 1 7 61120 0 No errors

References

- Project Remus: <http://dsg.cs.ubc.ca/remus/>
- Cross Vendor Migration:
<http://developer.amd.com/assets/CrossVendorMigration.pdf>
- QEMU live migration:
[http://kvm.et.redhat.com/wiki/images/5/5a/KvmForum2007\\$Kvm_Live_Migration_Forum_2007.pdf](http://kvm.et.redhat.com/wiki/images/5/5a/KvmForum2007$Kvm_Live_Migration_Forum_2007.pdf)