

Konfigurationsvoraussetzung und -optionen

Für die Unterstützung der Hardware-Virtualisierung sind ein paar Grundvoraussetzungen notwendig. Dreh- und Angelpunkt ist natürlich die Virtualisierungsunterstützung der CPU. Aus diesem Grund wurde bei der [Auswahl](#) der Hardware eine **AMD Opteron 6-Core 4170 HE** gewählt. Da die verbaute CPU auch wirklich die Hardwarevirtualisierung unterstützt können wir mit Hilfe von **cpuinfo** aus dem **/proc**-Verzeichnispfad überprüfen.

```
# grep "svm" /proc/cpuinfo
```

```
flags      : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp
lm 3dnowext 3dnow constant_tsc rep_good nonstop_tsc extd_apicid pni monitor
cx16 popcnt lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse
3dnowprefetch osvw ibs skinit wdt nodeid_msr npt lbrv svm_lock nrip_save
flags      : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp
lm 3dnowext 3dnow constant_tsc rep_good nonstop_tsc extd_apicid pni monitor
cx16 popcnt lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse
3dnowprefetch osvw ibs skinit wdt nodeid_msr npt lbrv svm_lock nrip_save
flags      : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp
lm 3dnowext 3dnow constant_tsc rep_good nonstop_tsc extd_apicid pni monitor
cx16 popcnt lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse
3dnowprefetch osvw ibs skinit wdt nodeid_msr npt lbrv svm_lock nrip_save
flags      : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp
lm 3dnowext 3dnow constant_tsc rep_good nonstop_tsc extd_apicid pni monitor
cx16 popcnt lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse
3dnowprefetch osvw ibs skinit wdt nodeid_msr npt lbrv svm_lock nrip_save
flags      : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp
lm 3dnowext 3dnow constant_tsc rep_good nonstop_tsc extd_apicid pni monitor
cx16 popcnt lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse
3dnowprefetch osvw ibs skinit wdt nodeid_msr npt lbrv svm_lock nrip_save
```

Da es sich bei unserer CPU vom Typ *AMD-V* sieht man an dem Flag [svm^{1\)}](#). Die Nutzung von **SVM** müssen wir im [BIOS kontrollieren](#) und ggf. aktivieren. Das Flag [lm^{2\)}](#) kennzeichnet eine 64bit-CPU. Aus diesem Grund werden wir bei der Installation unseres Wirts auch die 64bit-Variante von CentOS 6 [installieren](#).

BIOS-Optionen

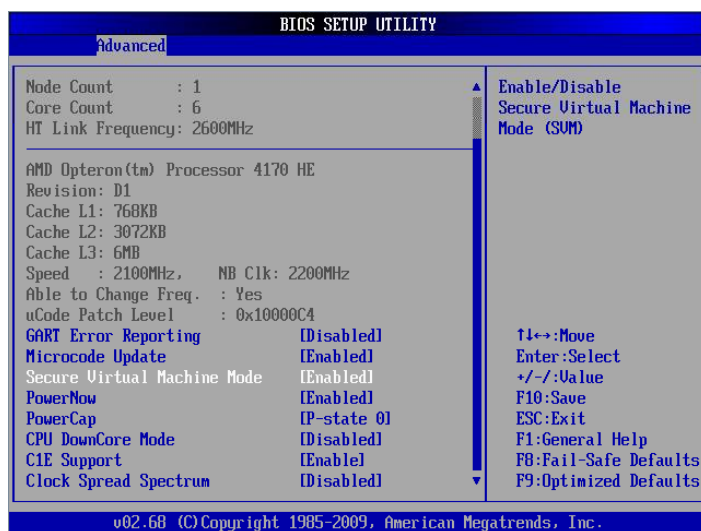
Als erstes und wichtiges kontrollieren wir die BIOS-Einstellungen unseres Servers.

Secure Virtual Machine Mode (Advanced Setup)

Unter *Secure Virtual Machine* versteht man eine Befehlssatzerweiterung zur Verbesserung der Virtualisierungsmöglichkeiten, die AMD für die x86-Architektur entwickelt hat.

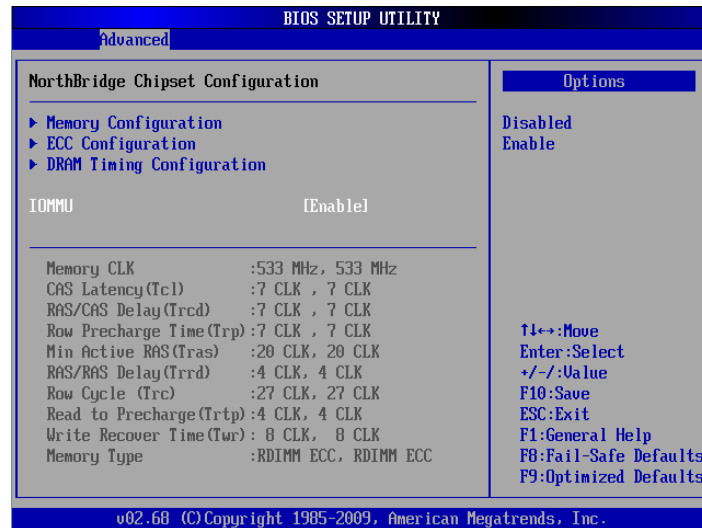
Virtualization Technology can virtually separate your system resource into several parts,
thus enhance the performance when running virtual machines or multi interface systems.

Für den Einsatz der KVM-Kernel-Module muss die Unterstützung der Hardware-Virtualisierung aktiviert werden. Hierzu kontrollieren und korrigieren wir bei Bedarf die BIOS-Einstellungen unseres Rechners.



IOMMU (Northbridge Chipset Configuration)

Mittels der PCI-Passthrough-Technik kann einem Gastsystem auf die physikalische Hardware, so z.B. PCI, PCIe oder USB-Devices, des Wirts zugreifen. Hierzu ist es auf unserem AMD-System notwendig, dass im BIOS **IOMMU** aktiviert wird.



Kernel-Optionen

Damit wir in unserer Virtualisierungsumgebung auch die Möglichkeiten von CentOS 6 optimal nutzen können, werden wir nun auf Betriebssysteme die Einstellungen kontrollieren und bei Bedarf anpassen.

Secure Virtual Machine Mode - AMD-Vi/IOMMU

Unter CentOS 6 ist die Unter Da standardmäßig die Unterstützung für AMD IOMMU im Kernel **2.6.32-71.29.1.el6.x86_64** von CentOS einkompiliert. Dies zeigt uns eine Abfrage des betreffenden Kernel-Konfigfiles.

```
# grep IOMMU /boot/config-`uname -r`
```

```
CONFIG_GART_IOMMU=y
CONFIG_CALGARY_IOMMU=y
# CONFIG_CALGARY_IOMMU_ENABLED_BY_DEFAULT is not set
CONFIG_AMD_IOMMU=y
CONFIG_AMD_IOMMU_STATS=y
CONFIG_IOMMU_HELPER=y
CONFIG_IOMMU_API=y
# CONFIG_IOMMU_DEBUG is not set
# CONFIG_IOMMU_STRESS is not set
```

Fragen wir nach einer Neuinstallation ab, ob diese Option auch geladen wurde, so können wir jedoch erst einmal nur folgenden Hinweis sehen.

```
# dmesg | grep AMD-Vi
AMD-Vi disabled by default: pass amd_iommu=on to enable
```

Dieser Aufforderung kommen wir natürlich gleich mal nach und Tragen die Option **amd_iommu=on** als weiteren Boot-Parameter in die Menüliste vom Bootloader grub ein.

```
# vim /boot/grub/menu.lst
```

```
grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#           all kernel and initrd paths are relative to /boot/, eg.
#           root (hd0,0)
#           kernel /vmlinuz-version ro root=/dev/mapper/vg_mnss-lv_root
#           initrd /initrd-[generic-]version.img
#boot=/dev/ddf1_raid1
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS Linux (2.6.32-71.29.1.el6.x86_64)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-71.29.1.el6.x86_64 ro
root=/dev/mapper/vg_mnss-lv_root rd_DM_UUID=ddf1_raid1
rd_LVM_LV=vg_mnss/lv_root rd_LVM_LV=vg_mnss/lv_swap rd_NO_LUKS rd_NO_MD
LANG=en_US.UTF-8 SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=de-
latin1-nodeadkeys crashkernel=512M vga=795 amd_iommu=on
    initrd /initramfs-2.6.32-71.29.1.el6.x86_64.img
```

Beim nächsten Systemneustart unseres Servers wir dann **IOMMU PCI-Passthrough** auch mitaktiviert. Wir starten also den Rechner neu.

```
# reboot
```

Anschließend überprüfen wir, ob auch die Änderung, die wir gerade vorgenommen haben, das gewünschte Resultat bringen.

```
# dmesg | grep AMD-Vi
```

```
AMD-Vi: Enabling IOMMU at 0000:00:00.2 cap 0x40
AMD-Vi: Initialized for Passthrough Mode
AMD-Vi: Enabling IOMMU at 0000:00:00.2 cap 0x40
```

Kernel Samepage Merging - KSM

Mit Hilfe von KSM³⁾ kann das Wirtssystem bei der Virtualisierung mehrerer Gäste, gleiche Speicherbereiche zusammenfassen und somit Arbeitsspeicher effizienter ausnutzen. Somit kann man unter anderem mehr Arbeitsspeicher den Gastsystemen zur Verfügung stellen, als tatsächlich physikalisch verbaut wurden. Als erstes überprüfen wir, ob der verwendete Kernel unter Centos das auch unterstützt. Auf unserem CentOS 6 Wirtssystem fragen wir also ab:

```
# grep KSM /boot/config-`uname -r`
```

```
CONFIG_KSM=y
```

Wie auch schon bei der Option **IOMMU** bedeutet dies aber noch zwangsläufig nicht, dass der KSM-

Support auch aktiviert wurde. Um auch dies nun 100%ig zu wissen fragen wir die Datei **/sys/kernel/mm/ksm/** auf dem Wirt ab.

```
# cat /sys/kernel/mm/ksm/run
```

Die Rückgabewerte haben dabei folgende Bedeutung:

- **0** KSM ist nicht aktiv
- **1** KSM ist aktiv
- **2** KSM löst gerade gemeinsam überlappende Speicherbereiche auf.

Konfiguration

Für die Konfiguration des ksmtune-daemon **ksmtuned** bearbeiten wir die Konfigurationsdatei.

```
# vim /etc/ksmtuned.conf
```

[/etc/ksmtuned.conf](#)

```
# Configuration file for ksmtuned.

# How long ksmtuned should sleep between tuning adjustments
KSM_MONITOR_INTERVAL=60

# Millisecond sleep between ksm scans for 16Gb server.
# Smaller servers sleep more, bigger sleep less.
KSM_SLEEP_MSEC=10

KSM_NPAGES_BOOST=300
KSM_NPAGES_DECAY=-50
KSM_NPAGES_MIN=64
KSM_NPAGES_MAX=1250

KSM_THRES_COEF=20
KSM_THRES_CONST=2048

# uncomment the following if you want ksmtuned debug info

# LOGFILE=/var/log/ksmtuned
# DEBUG=1
```

Start

Für die Aktivierung der Kernel Smpage Merging gibt es zwei daemon:

- **ksm** startet und stoppt den *KSM kernel thread*
- **ksmtuned** Der ksmtune-Daemon kontrolliert den ksm-Daemon und startet/stop diesen, sofern

Memorysharing nicht benötigt wird.

Vom dem Starten können wir noch den Status bzgl. KSM überprüfen. Hierzu fragen die Datei **/sys/kernel/mm/ksm/** auf dem Wirt ab.

```
# cat /sys/kernel/mm/ksm/run
```

Die zurückgegebene **0** zeigt an, dass der die Option *Kernel Samepage Merking* nicht aktiv ist. Nun starten wir die beiden Systemdienste.

```
# service ksm start
Starting ksm: [ OK ]
```

```
# service ksmtuned start
Starting ksmtuned: [ OK ]
```

Fragen wir nun erneut den Status zu KSM ab, bekommen wir als Rückmeldung entsprechend eine **1**, die anzeigt, dass KSM genutzt wird.

```
# cat /sys/kernel/mm/ksm/run
```

```
1
```

Damit die beiden KSM-Dienste auch bei Restart des Wirtrechners gestartet werden, müssen wir nur noch die Dienst in den Autostart versetzen.

```
# chkconfig ksm on
```

```
# chkconfig ksmtuned on
```

Dass beide Services auch automatisch im runlevel 3/4/5 gestartet werden können wir einfach überprüfen.

```
# chkconfig --list | grep ksm
```

ksm	0:Aus	1:Aus	2:Ein	3:Ein	4:Ein	5:Ein	6:Aus
ksmtuned	0:Aus	1:Aus	2:Ein	3:Ein	4:Ein	5:Ein	6:Aus

Logging

Bei aktiviertem **ksmtuned** schreibt der Daemion im LOGverzeichnis entsprechende Rückmeldungen.

```
Tue Aug 2 22:51:52 CEST 2011: total 15874952
Tue Aug 2 22:51:52 CEST 2011: sleep 10
Tue Aug 2 22:51:52 CEST 2011: thres 3174990
Tue Aug 2 22:52:52 CEST 2011: committed 14446180 free 11641800
Tue Aug 2 22:52:52 CEST 2011: 17621170 > 15874952, start ksm
Tue Aug 2 22:52:52 CEST 2011: 11641800 > 3174990, decay
Tue Aug 2 22:52:52 CEST 2011: KSMCTL start 64 10
```

```
Tue Aug 2 22:53:52 CEST 2011: committed 14081140 free 11912704
Tue Aug 2 22:53:52 CEST 2011: 17256130 > 15874952, start ksm
Tue Aug 2 22:53:52 CEST 2011: 11912704 > 3174990, decay
Tue Aug 2 22:53:52 CEST 2011: KSMCTL start 64 10
Tue Aug 2 22:54:52 CEST 2011: committed 11238176 free 12920740
Tue Aug 2 22:54:52 CEST 2011: 14413166 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 22:55:53 CEST 2011: committed 9833852 free 13156712
Tue Aug 2 22:55:53 CEST 2011: 13008842 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 22:56:53 CEST 2011: committed 5617520 free 14078752
Tue Aug 2 22:56:53 CEST 2011: 8792510 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 22:59:35 CEST 2011: total 15874952
Tue Aug 2 22:59:35 CEST 2011: sleep 10
Tue Aug 2 22:59:35 CEST 2011: thres 3174990
Tue Aug 2 23:00:35 CEST 2011: committed 13445116 free 11923676
Tue Aug 2 23:00:35 CEST 2011: 16620106 > 15874952, start ksm
Tue Aug 2 23:00:35 CEST 2011: 11923676 > 3174990, decay
Tue Aug 2 23:00:35 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:01:35 CEST 2011: committed 12717792 free 12274464
Tue Aug 2 23:01:35 CEST 2011: 15892782 > 15874952, start ksm
Tue Aug 2 23:01:35 CEST 2011: 12274464 > 3174990, decay
Tue Aug 2 23:01:35 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:02:35 CEST 2011: committed 12584620 free 12521744
Tue Aug 2 23:02:35 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:03:36 CEST 2011: committed 12584620 free 12517928
Tue Aug 2 23:03:36 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:04:36 CEST 2011: committed 12594864 free 12514420
Tue Aug 2 23:04:36 CEST 2011: 15769854 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:05:36 CEST 2011: committed 12584620 free 12514164
Tue Aug 2 23:05:36 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:06:36 CEST 2011: committed 12625596 free 12507048
Tue Aug 2 23:06:36 CEST 2011: 15800586 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:07:36 CEST 2011: committed 12605108 free 12499980
Tue Aug 2 23:07:36 CEST 2011: 15780098 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:08:36 CEST 2011: committed 12584620 free 12498084
Tue Aug 2 23:08:36 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:09:36 CEST 2011: committed 12584620 free 12494324
Tue Aug 2 23:09:36 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:10:36 CEST 2011: committed 12584620 free 12495328
Tue Aug 2 23:10:36 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
```

```
Tue Aug 2 23:11:36 CEST 2011: committed 12584620 free 12492332
Tue Aug 2 23:11:36 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:12:36 CEST 2011: committed 12584620 free 12488656
Tue Aug 2 23:12:36 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:13:36 CEST 2011: committed 12584620 free 12484000
Tue Aug 2 23:13:36 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:14:36 CEST 2011: committed 12584620 free 12488356
Tue Aug 2 23:14:36 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:15:36 CEST 2011: committed 12584620 free 12488536
Tue Aug 2 23:15:36 CEST 2011: 15759610 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:16:36 CEST 2011: committed 12641440 free 12474464
Tue Aug 2 23:16:36 CEST 2011: 15816430 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:17:36 CEST 2011: committed 14352128 free 11874620
Tue Aug 2 23:17:36 CEST 2011: 17527118 > 15874952, start ksm
Tue Aug 2 23:17:36 CEST 2011: 11874620 > 3174990, decay
Tue Aug 2 23:17:36 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:18:36 CEST 2011: committed 14011808 free 11972244
Tue Aug 2 23:18:36 CEST 2011: 17186798 > 15874952, start ksm
Tue Aug 2 23:18:36 CEST 2011: 11972244 > 3174990, decay
Tue Aug 2 23:18:36 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:19:36 CEST 2011: committed 14418812 free 11541316
Tue Aug 2 23:19:36 CEST 2011: 17593802 > 15874952, start ksm
Tue Aug 2 23:19:36 CEST 2011: 11541316 > 3174990, decay
Tue Aug 2 23:19:36 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:20:36 CEST 2011: committed 14009052 free 11542528
Tue Aug 2 23:20:36 CEST 2011: 17184042 > 15874952, start ksm
Tue Aug 2 23:20:36 CEST 2011: 11542528 > 3174990, decay
Tue Aug 2 23:20:36 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:21:37 CEST 2011: committed 14009052 free 11542428
Tue Aug 2 23:21:37 CEST 2011: 17184042 > 15874952, start ksm
Tue Aug 2 23:21:37 CEST 2011: 11542428 > 3174990, decay
Tue Aug 2 23:21:37 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:22:37 CEST 2011: committed 14009052 free 11543000
Tue Aug 2 23:22:37 CEST 2011: 17184042 > 15874952, start ksm
Tue Aug 2 23:22:37 CEST 2011: 11543000 > 3174990, decay
Tue Aug 2 23:22:37 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:23:37 CEST 2011: committed 14091004 free 11598516
Tue Aug 2 23:23:37 CEST 2011: 17265994 > 15874952, start ksm
Tue Aug 2 23:23:37 CEST 2011: 11598516 > 3174990, decay
Tue Aug 2 23:23:37 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:24:37 CEST 2011: committed 6974796 free 12595028
Tue Aug 2 23:24:37 CEST 2011: 10149786 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:27:39 CEST 2011: total 15874952
Tue Aug 2 23:27:39 CEST 2011: sleep 10
```

```
Tue Aug 2 23:27:39 CEST 2011: thres 3174990
Tue Aug 2 23:28:39 CEST 2011: committed 13981776 free 11693324
Tue Aug 2 23:28:39 CEST 2011: 17156766 > 15874952, start ksm
Tue Aug 2 23:28:39 CEST 2011: 11693324 > 3174990, decay
Tue Aug 2 23:28:39 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:29:39 CEST 2011: committed 14094460 free 11952928
Tue Aug 2 23:29:39 CEST 2011: 17269450 > 15874952, start ksm
Tue Aug 2 23:29:39 CEST 2011: 11952928 > 3174990, decay
Tue Aug 2 23:29:39 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:30:39 CEST 2011: committed 14043240 free 12474780
Tue Aug 2 23:30:39 CEST 2011: 17218230 > 15874952, start ksm
Tue Aug 2 23:30:39 CEST 2011: 12474780 > 3174990, decay
Tue Aug 2 23:30:39 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:31:39 CEST 2011: committed 14043240 free 12605008
Tue Aug 2 23:31:39 CEST 2011: 17218230 > 15874952, start ksm
Tue Aug 2 23:31:39 CEST 2011: 12605008 > 3174990, decay
Tue Aug 2 23:31:39 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:32:40 CEST 2011: committed 14043240 free 12572092
Tue Aug 2 23:32:40 CEST 2011: 17218230 > 15874952, start ksm
Tue Aug 2 23:32:40 CEST 2011: 12572092 > 3174990, decay
Tue Aug 2 23:32:40 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:33:40 CEST 2011: committed 14063728 free 12564236
Tue Aug 2 23:33:40 CEST 2011: 17238718 > 15874952, start ksm
Tue Aug 2 23:33:40 CEST 2011: 12564236 > 3174990, decay
Tue Aug 2 23:33:40 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:34:40 CEST 2011: committed 14063728 free 12564052
Tue Aug 2 23:34:40 CEST 2011: 17238718 > 15874952, start ksm
Tue Aug 2 23:34:40 CEST 2011: 12564052 > 3174990, decay
Tue Aug 2 23:34:40 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:35:40 CEST 2011: committed 14043240 free 12570536
Tue Aug 2 23:35:40 CEST 2011: 17218230 > 15874952, start ksm
Tue Aug 2 23:35:40 CEST 2011: 12570536 > 3174990, decay
Tue Aug 2 23:35:40 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:36:40 CEST 2011: committed 14043240 free 12568300
Tue Aug 2 23:36:40 CEST 2011: 17218230 > 15874952, start ksm
Tue Aug 2 23:36:40 CEST 2011: 12568300 > 3174990, decay
Tue Aug 2 23:36:40 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:37:40 CEST 2011: committed 14043240 free 12564352
Tue Aug 2 23:37:40 CEST 2011: 17218230 > 15874952, start ksm
Tue Aug 2 23:37:40 CEST 2011: 12564352 > 3174990, decay
Tue Aug 2 23:37:40 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:38:40 CEST 2011: committed 14043240 free 12604708
Tue Aug 2 23:38:40 CEST 2011: 17218230 > 15874952, start ksm
Tue Aug 2 23:38:40 CEST 2011: 12604708 > 3174990, decay
Tue Aug 2 23:38:40 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:39:40 CEST 2011: committed 8426168 free 13579428
Tue Aug 2 23:39:40 CEST 2011: 11601158 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:40:40 CEST 2011: committed 1404548 free 14881904
Tue Aug 2 23:40:40 CEST 2011: 4579538 < 15874952 and free > 3174990, stop
ksm
```

```
Tue Aug 2 23:42:45 CEST 2011: total 15874952
Tue Aug 2 23:42:45 CEST 2011: sleep 10
Tue Aug 2 23:42:45 CEST 2011: thres 3174990
Tue Aug 2 23:43:45 CEST 2011: committed 14022752 free 11663068
Tue Aug 2 23:43:45 CEST 2011: 17197742 > 15874952, start ksm
Tue Aug 2 23:43:45 CEST 2011: 11663068 > 3174990, decay
Tue Aug 2 23:43:45 CEST 2011: KSMCTL start 64 10
Tue Aug 2 23:44:45 CEST 2011: committed 9738296 free 12778456
Tue Aug 2 23:44:45 CEST 2011: 12913286 < 15874952 and free > 3174990, stop
ksm
Tue Aug 2 23:45:45 CEST 2011: committed 2788384 free 14553656
```

Links

- [Zurück zum Kapitel >>Virtualisierung mit Hilfe von KVM, QEMU und libvirt unter CentOS 6<<](#)
- [Zurück zu >>Projekte und Themenkapitel<<](#)
- [Zurück zur Startseite](#)

¹⁾

secure virtual machine

²⁾

long mode

³⁾

Kernel **S**amepage **M**erging

From:

<https://dokuwiki.nausch.org/> - **Linux - Wissensdatenbank**

Permanent link:

https://dokuwiki.nausch.org/doku.php/centos:kvm:hw_konfig

Last update: **20.04.2018 10:48.**

