How to increase XenServer virtual machine root or swap partition

When doing a standard Linux installation on a XenServer virtual machine, the installation process typically allocates all of the available virtual disk space to the Linux file system and swap partitions. The size of the virtual disk can easily be changed using the XenCenter utility, but as neither the partition table or the file systems themselves are modified by XenCenter, the guest system does not see the new virtual disk size.

This article describes how to safely change the partitions.

There are two scenarios:

- the Linux system uses standard Linux partitions
- the Linux system uses Logical Volume Manager (LVM) partitions

The file system sits on top of a single block of data storage, such as a piece of the harddisk or, in the case of a XenServer virtual machine, a virtual block device. It can only be as large as this storage block, but not larger. Therefore, if the file system is to be increased, this storage block must be increased first.

After the size of the storage block has been increased, execute the `resize2fs` command, which can be safely run even on a mounted filesystem.

It is very simple to increase the size of the underlying storage block if you are using LVM (and this is a good reason to always use LVM). Just add more blocks to it using the `lvresize` command.

The blocks added to the storage block are taken from the current volume group. If the volume group does not have free block, new blocks must be made available by adding a new physical device such as a second harddisk, or by increasing an existing physical device.

Provide more disk space to the guest system

There are two ways to do this: you could either increase the size of the existing XenServer virtual disk, or add a second virtual disk to this guest.

Add a physical device

(todo)

Increase the size of an existing XenServer virtual device

The XenCenter administration software provides a dialog to change the size of the XenServer virtual disk. Select the new storage size by entering the value in the dialog. The virtual disk will be resized to the new value. When *increasing* the size, the data on the disk is preserved.

Note: do not decrease the size without decreasing the file system first. Otherwise the file system will be damaged.

> 📋 Unknown macro: 'clear'

Increasing the size of the XenServer virtual disk size does not change the filesystem on that disk. So the new disk space is not yet visible to the user. To use the new disk space, the partition table must be modified (which makes the new device size known to the file system) and the file system must be extended (to use the extra space).

The easiest way to do this is delete the last partition and create it again which will then use all of the remaining disk space by default. Obviously, this will work only if the last partition is in fact the partition that is to be increased.

In the example below, the XenServer virtual disk size has been resized from 10 GByte to 50 GByte. The following steps will be performed with `fdisk`:

- print old partitions
- delete last partition (#2)
- create new partition using default settings (use the whole virtual disk for this partition)
- set partition type to LVM
- print partitions settings again, note increased size for partition #2
- reboot

```
master:~# fdisk /dev/xvda

The number of cylinders for this disk is set to 6527.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/xvda: 53.6 GB, 53687091200 bytes     <-- note that fdisk already sees the new device size of 53.6 GB
255 heads, 63 sectors/track, 6527 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x0004e475

    Device Boot      Start         End      Blocks   Id  System
/dev/xvda1   *           1          31      248976   83  Linux
/dev/xvda2              32        1305    10229405   8e  Linux LVM     <-- note that the existing entry is
still 10 GB

Command (m for help): d
Partition number (1-4): 2

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (32-6527, default 32):
Using default value 32
Last cylinder or +size or +sizeM or +sizeK (32-6527, default 6527):
Using default value 6527

Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 8e
Changed system type of partition 2 to 8e (Linux LVM)

Command (m for help): p

Disk /dev/xvda: 53.6 GB, 53687091200 bytes
255 heads, 63 sectors/track, 6527 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x0004e475

    Device Boot      Start         End      Blocks   Id  System
/dev/xvda1   *           1          31      248976   83  Linux
/dev/xvda2              32        6527    52179120   8e  Linux LVM     <-- note that the new entry is now 50 GB

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
master:~# reboot

Broadcast message from root@master (pts/0) (Fri Oct  2 08:47:05 2009):

The system is going down for reboot NOW!
```

Tell LVM of the new resized physical volume

The partition #2 has now successfully increased in size, now let LVM use the new blocks. Note how the "Free PE" value increases after the `pvresize` command.

```
 pvdisplay
  --- Physical volume ---
  PV Name               /dev/xvda2
  VG Name               dsk
  PV Size               9.76 GB / not usable 1.37 MB
  Allocatable           yes (but full)
  PE Size (KByte)       4096
  Total PE              2498
  Free PE               0
  Allocated PE          2498
  PV UUID               7TyUyk-hyVf-0kdC-OCok-QdSJ-lqTE-YFJAi0

master:~# pvresize /dev/xvda2
  Physical volume "/dev/xvda2" changed
  1 physical volume(s) resized / 0 physical volume(s) not resized

master:~# pvdisplay
  --- Physical volume ---
  PV Name               /dev/xvda2
  VG Name               dsk
  PV Size               49.76 GB / not usable 3.98 MB
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              12738
  Free PE               10240
  Allocated PE          2498
  PV UUID               7TyUyk-hyVf-0kdC-OCok-QdSJ-lqTE-YFJAi0
```

Resize logical volume

The next step is to add those new blocks to the logical volume. The volume group now has an extra 10240 free extents (each 4 KByte in size). New storage can be added to a logical volume with the `lvresize` command.

```
# vgdisplay
  --- Volume group ---
  VG Name               dsk
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  8
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               49.76 GB
  PE Size               4.00 MB
  Total PE              12738
  Alloc PE / Size       2498 / 9.76 GB
  Free  PE / Size       10240 / 40.00 GB
  VG UUID               nCjL0f-XWps-G77Z-Oq0t-lMEc-ULsJ-2qFUnw

master:~# lvdisplay
  --- Logical volume ---
  LV Name               /dev/dsk/root
  VG Name               dsk
  LV UUID               QejNO3-8bB0-ykJ0-TUhR-56Ko-W5ed-SvF9fv
  LV Write Access       read/write
  LV Status             available
  # open                1
  LV Size               7.76 GB
  Current LE            1986
  Segments              1
  Allocation            inherit
  Read ahead sectors    auto
  - currently set to    256
  Block device          254:0

master:~# lvresize -l +10240 /dev/dsk/root
  Extending logical volume root to 47.76 GB
  Logical volume root successfully resized

master:~# vgdisplay
  --- Volume group ---
  VG Name               dsk
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  11
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               1
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               49.76 GB
  PE Size               4.00 MB
  Total PE              12738
  Alloc PE / Size       12738 / 49.76 GB
  Free  PE / Size       0 / 0
  VG UUID               nCjL0f-XWps-G77Z-Oq0t-lMEc-ULsJ-2qFUnw
```

Resize file system

The last step is to resize the file system. This can be done even when the file system is currently in use. However, a reboot is recommended as massive changes have been made to the file system. During reboot, a file system check is executed which will print out a scary warning message but that's normal. The system will reboot again and this time no messages or warnings should be sent.

```
master:~# resize2fs /dev/dsk/root
resize2fs 1.41.3 (12-Oct-2008)
Filesystem at /dev/dsk/root is mounted on /; on-line resizing required
old desc_blocks = 1, new_desc_blocks = 3
Performing an on-line resize of /dev/dsk/root to 12519424 (4k) blocks.
The filesystem on /dev/dsk/root is now 12519424 blocks long.

master:~# reboot

Broadcast message from root@master (pts/0) (Fri Oct  2 08:47:05 2009):

The system is going down for reboot NOW!
```