A Clonezilla Live Tutorial

Replacing A Disk With A Smaller One Using Clonezilla Live

version 0.2 – 2012-02-25

This document is currently a draft in test mode! Feel free to report any error or to suggest enhancements to the author.

Summary

We sometimes change a PC disk, just because it is failing or because SSD replacements are smaller than standard mechanical ones or for whatever reason.

When it comes to disk cloning or imaging, Clonezilla is usually the Libre software tool of choice: it's efficient, fast and very well supported. There's a drawback, though: Clonezilla can't directly clone or restore to a smaller target than the source device.

The Devil is in details, too! It happens that, within the same series of disks (same commercial reference), two of them are not *strictly* of the same capacity [1]. Yet, a very small difference can make the cloning impossible. We all know that often, for optimisation sake, vendors are adjusting their own hardware providers, so this situation is certainly not uncommon.

Then, how can we deal with these situations with our tool of choice?

This is what we're going to show here with a solution that works, at least for the author.

Backup Your Data!

Any action that touches disk structure, partitions and filesystems is a potential source of failure. You should ***never*** *test on living data but on dedicated material and hardware.*
A backup of the data before any other action is highly advised!

What This Document Is Not

This document supposes that disk geometry is a topic known by the reader and that such tools like Clonezilla Live and GParted are well known. In other words, this document is in no way a manual explaining the uses of these tools. Its goal is just to introduce a solution to a common problem.

---

1 To ensure that two disks from the same vendor and type are from the same batch, check the model codes that are printed onto the drives.

1. Presentation

Clonezilla Live is a Free software live CD that can clone or backup disks or partitions. Using *Clonezilla* makes it is easy to restore a disk to a previous state that had been saved at that time. In corporate environments, *Clonezilla* allows to create identical machines from one unique model configuration, whatever the underlying operating system.

1.1. Prerequisites

Hardware

• The target (replacement) disk.

• (not mandatory but highly recommended) A spare device for temporary storage.

Software

• *Clonezilla Live* (version used here: v.1.2.12-10) ;

• *PartedMagic* (or *GParted* or any similar tool) (version used here: v.6.7).

1.2. Process Overview

Our starting point is shown at figure 1: we've got a running PC with a 30 GiB hard drive that we'd like to replace for a new 25 GiB disk.

The source disk holds a Windows XP sp3 system on two partitions. The size of the first one is 19.58 GiB, the second is 10.42 GiB.

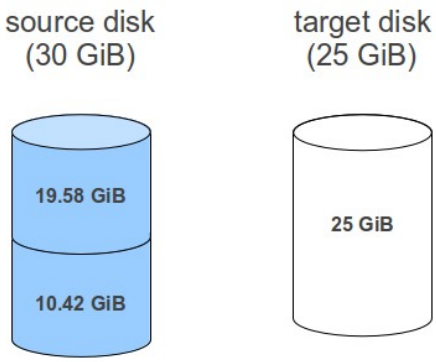

Figure 1: Our starting point

The process is a three-step one: first we set our source disk so that it can be cloned, second we partition the target disk according to the source disk partitioning and, third, we copy the data from the source to the target. Figure 2 shows how we are going to deal with the disk cloning.

We ran our experiments in a virtual machine (*VirtualBox* v. 4.1.8) set up with three disks: a source disk (disk 1), with a *Windows XP* system on two partitions, a target disk (disk 2) and a temporary disk for backup (disk 3). Table 1 gives details about our three test disks.

The following chapters give more details about the process which is not complicated. Somewhat lengthy if the source disk is big and holds much data, but not complicated at all.

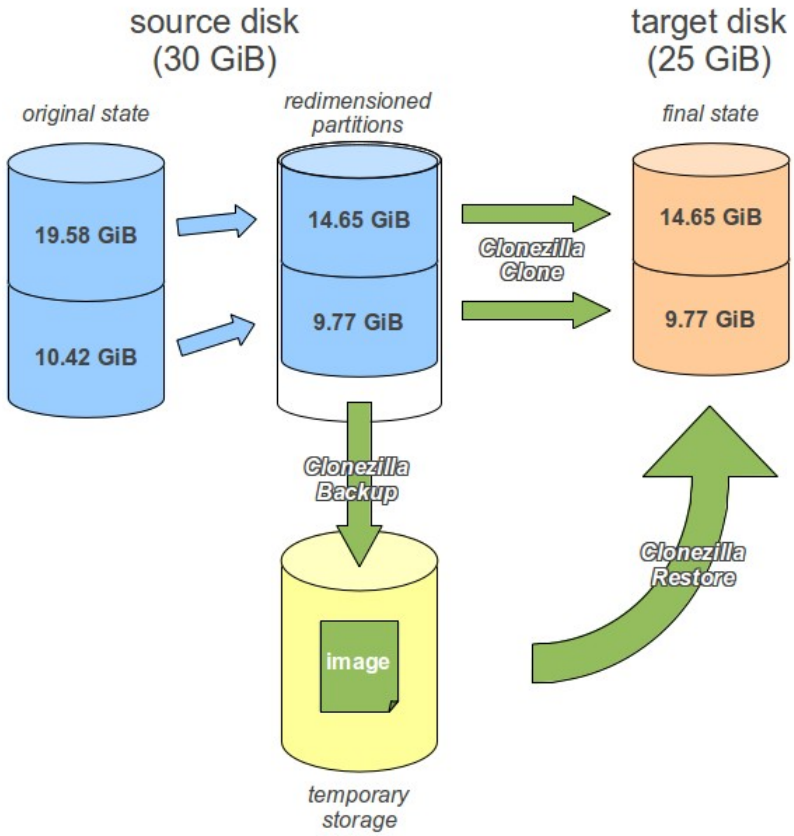

Figure 2: Overview of the disks cloning scheme

disk		capacity	partitions
disk 1	source	30 GiB	Partition 1: 19.58 GiB, primary, NTFS, boot Partition 2: 10.42 GiB, primary, NTFS
disk 2	target	25 GiB	Partition 1: 14.65 GiB, primary, NTFS, boot Partition 2: 9.77 GiB, primary, NTFS
disk 3	temporary	At least 5 GiB	One single partition, any Clonezilla supported file system (FAT, ext, NTFS, etc.)

Table 1: Disks description

1.3. Estimated Duration

What about the duration of the process? Of course, many parameters will interfere, the first of all being the size of data on the partitions to move.

Our test system is, admittedly, quite “empty”. On the other hand, virtual machines are slower than actual ones. Anyway, we found the durations shown at table 2 below (your mileage may vary greatly...).

Preparing source disk	**50 minutes**
Defragmenting (MS Windows)	15 minutes
Partition shrinking (GParted)	2 minutes
Second partition moving (GParted)	25 minutes
System checks (MS Windows)	5 minutes
Preparing target disk	**5 minutes**
Partitioning (GParted)	5 minutes
Direct cloning (Clonezilla)	**minutes**
Cloning `sda1`	minutes
Cloning `sda2`	minutes
Imaging (Clonezilla)	**minutes**
Image creation	minutes
Image restoration	minutes
Overall, in case of direct cloning	**minutes**
Overall, using a temporary storage	**minutes**

Table 2: Process duration on the Windows test system

Please note that:

- *The delays noted here are highly dependant on the actual data on disk.*
- *Processing non-Windows systems will skip some operations and will be shorter.*
- *The partition move is certainly the longest operation.*
  The actual amount of data on `sda2` implies the length of that process.

2. Preparing The System

Before we can backup or clone our source disk, we must prepare it for geometry compatibility with the target device. We'll have to shrink the partitions so that the sum of their sizes is equal or less than the target disk space. The target disk geometry must be the same as the source disk one.

Moreover, NTFS file systems need some supplemental steps.

2.1. Source Disk Processing

A. Source NTFS File System

NTFS filesystems (MS Windows) are in need of a defragmentation before any further action. This can be done either through the usual Windows GUI tool or by issuing the following command:
`defrag <disk:> -f`
(eg: `defrag c: -f`).

When the defragmentation is over, we may process the partitions.

B. Source Partition Processing

Using some dedicated tool we'll now process the source disk partitions without destroying the data they carry.

GNU/Linux distributions most often come with such a tool. GParted is very well known but our distro might offer some alternative.

Windows systems never propose that kind of tool. In this situation, we'll resort to using GParted that comes on the PartedMagic live CD.

Figure 3 shows the source disk geometry before the cloning, as displayed by GParted.

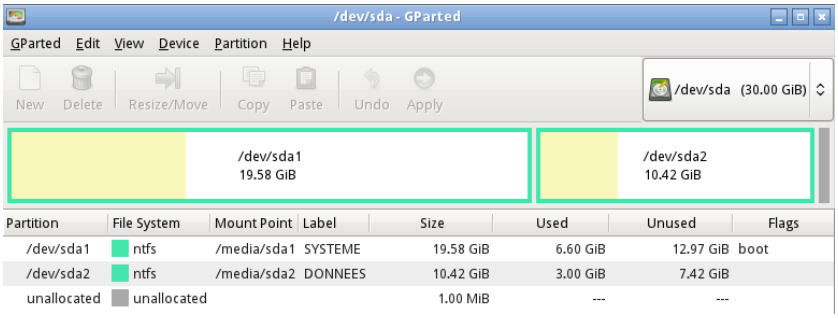

Figure 3: Source disk geometry, before shrinking partitions

Partitions Shrinking

We can see that the free space remaining in each partition allows shrinking them without hassle. We'll shrink successively:

- `sda1` from 19.58 to 14.65 GiB (15,000 MiB),
- `sda2` from 10.42 to 9.77 GiB (10,000 MiB),

which gives a total new size of 24.42 GiB. We get the new geometry displayed at figure 4.

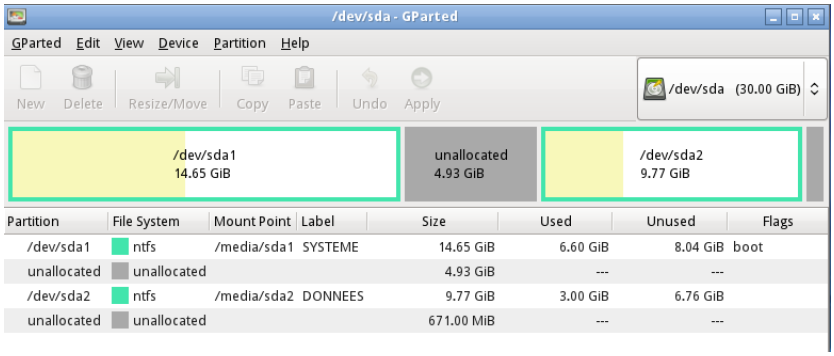

Figure 4: The new geometry after partition shrinking

Partition Moving

We notice that some new unallocated areas have appeared because of the shrinking. We now have to adjust both partitions in order to suppress the unallocated interval between them.

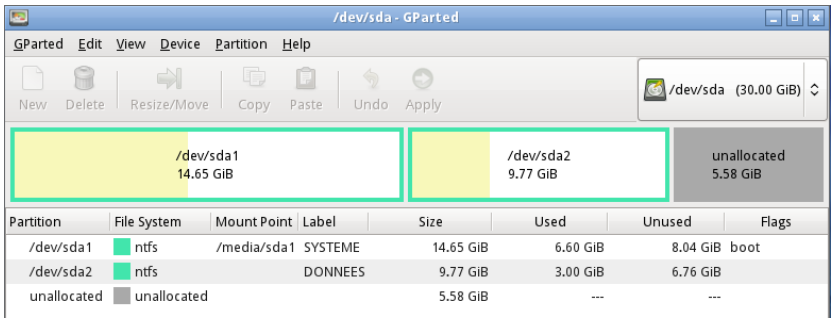

Figure 5: The source disk partitions are now ready for cloning

For that, using the same tool, we move `sda2` to the left so that the interval becomes null. Figure 5 shows the final state.

Be Patient!

The move operation is, by far, the longest one in the overall process. Depending upon the amount of data, this can take much more than an hour (in our example it took 25 minutes to move `sda2`).

Checking The Source File System (NTFS Only)

If the partitions hold NTFS filesystems (MS Windows), we must check each of the partitions. *Clonezilla* won't process any unchecked NTFS-formatted partition.

As re-dimensioning a partition implies a check during the next Windows start up, let's start the machine once again! [2]

2 This operation may be forced issuing the Windows shell command:
`chkdsk <disk:> -f`
eg: `chkdsk c: -f`

2.2. Preparing The Target Disk

We won't forget to check that the target disk is properly partitioned, with the same geometry as the source disk (see figure 6).

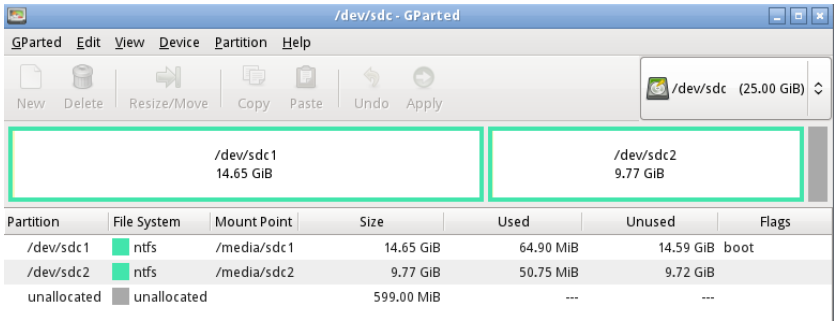

Figure 6: The target disk has the same geometry as the source

3. Replicating To The Target Disk

Clonezilla has two cloning options: disk cloning or partitions cloning.

Disk cloning is not possible here, just because of the disk capacity limitation we're dealing with. This is why we're going to use the partition cloning scheme. For this to be possible, the target disk must be previously partitioned. Moreover, the target disk geometry has to be the same as the source disk (figure 6): partition size, optionally boot flag.

Now we may actually clone the source disk.

3.1. First Method: Direct Cloning (Disk To Disk)

Here, we're going to clone directly from the source disk to the target disk (`device-device` option in *Clonezilla*). The target disk must be connected to the same machine as the source one (ATA, SATA, USB, pick your own).

Caution!

*The direct cloning method shown here is **highly risky**: in case of source and target partitions mismatch, the data will be **irremediably** lost.*

Be cautious!

Direct cloning requires to clone one partition at a time. We can't clone several partitions at once, which is possible using the second method, explained below at chapter 3.2.

We first select the source partition (`sda1`) then the target (here `sdb1`) and we accept all default values. The process executes (figure 7).

In the end, we select `(3) Start over` to start again and process the second partition in the same way.

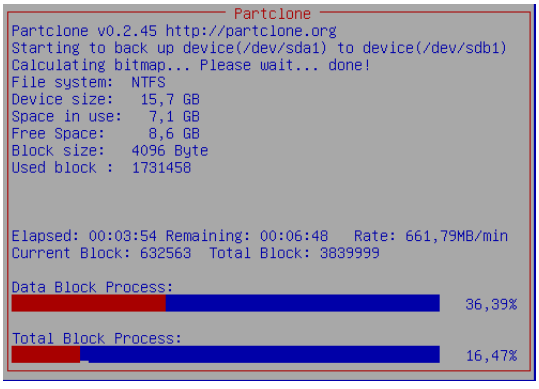

Figure 7: Cloning execution

Using The Target Disk

Once the cloning of a Windows system is over, it is ***mandatory*** *to disconnect one of the disks from the machine. Otherwise, MS Windows would be disturbed when finding two identical disks therefore refusing to boot.*

3.2. Second Method: Using A Temporary Image

Here, we're going to use a two-step process. First we backup the source disk partitions as an image to a temporary disk. Then we restore the image to the target disk (see figure 2). For this, we are using a third disk which holds the intermediate image. It is usually easier to use some external device that we can connect to the USB port. A USB thumbdrive can do the trick, provided the available space is sufficient. The free size requirement depends upon the actual volume of data on the source disk; 8 GiB is usually considered a fair enough volume as Clonezilla compresses the image files [3].

A. Saving The Image To The Storage Device

We run Clonezilla, accepting the default mode: `device-image`. We then select the necessary options in order to create an image of both `sda1` and `sda2` source partitions to the temporary disk (`localdev` then `saveparts`). The default parameters are fit. If the source is a Windows system partition, we should also select the `rm-win-swap-hib` option that will ensure we don't backup swap and hibernation files; whenever these files are missing, Windows re-creates them at start up.

In the end, the source partitions images are present as a set of files stored in a directory on the temporary disk. The directory name is the one we chose during the imaging process.

B. Restoring The Image To The New Device

We run *Clonezilla* one last time. We accept the default `device-image` mode. We select the options needed to restore the `sdb1` and `sdb2` partitions images to the target (`localdev` then `restoreparts`). The default parameters are fit.

3 As a reference, our *Windows* XP sp3 install and some more software give a 3,06 GiB image.

4. Windows FAQ

A common problem to many cloning schemes is a cloned Windows operating system won't boot on the cloned disk. MS Windows systems are very sensitive to any hardware change and may be highly disturbed by a new disk structure. Here are three questions that come over and over, with links to the answers on the Clonezilla web site FAQ pages.

A. After I Did A Disk-to-disk Clone, My MS Windows In The Source Disk Fails To Boot. Why?

Did you leave both disks into the machine?

http://drbl.org/faq/fine-print.php?path=./2_System/111_disk_to_disk_fail_boot_MS_win.faq#111_disk_to_disk_fail_boot_MS_win.faq

B. After The Image Is Restored To Another Machine, It Fails To Boot. The Error Message Is "Missing Operating System" Or Just A Blinking Underscore

Usually this is because GNU/Linux and MS Windows interpret the CHS (cylinder, head, sector) value of hard drives differently. Some possible solutions are given here:

http://drbl.org/faq/fine-print.php?path=./2_System/23_Missing_OS.faq#23_Missing_OS.faq

C. I Get An Error "0xc00000e : Can't Run WINLOAD.EXE" (or 0xc0000425) After Restoring A Windows7 Image

Here's the solution:

http://drbl.org/faq/fine-print.php?path=./2_System/86_Win_7_0xc00000e_error.faq#86_Win_7_0xc00000e_error.faq

5. References

Here's some information about the tools we've been using throughout this document. At the web addresses below, you'll find more resources and information: downloads, help and documentation, etc.

Clonezilla Live	http://www.clonezilla.org/
PartedMagic	http://partedmagic.com/doku.php?id=start
VirtualBox	https://www.virtualbox.org/

6. Credits

Author: Jean-Francois Nifenecker, jean-francois.nifenecker@laposte.net

Acknowledgements: to Steven Shiau and the Clonezilla Team, also to the members of the Clonezilla forums who gave feedback.

Translation:

Modification History

Version	Date	Comments
0.1	2012-02-19	First draft for proofreading
0.2	2012-02-25	Second draft with some additions and corrections

7. License