Sysresccd-Partitioning-EN-The-new-GPT-disk-layout

From SystemRescueCd

Contents

About partition tables

Hard-drives can contain multiple partitions which helps to organize data and to install multiple operating systems on the same disk. There are multiple standards for the layout of the partition table: the standard layout which is used on most computers is the `MSDOS partition table`, and the new replacement layout is called `GPT (GUID Partition Table)`. The problem with the standard layout on PC is that it has many limitations: it only supports up to 4 primary partitions per disk, and it does not allow us to address more than 2 Tera-Bytes. This section will inform you about everything you may want to know about GPT: whether or not you should use it, and how to use it.

Limitations with the MSDOS layout

The MSDOS partition table was introduced a very long time ago. The first sector of the disk is called the Master Boot Record (MBR) which contains two things: The first 440 bytes is used to store the instructions that the computer will execute when the computer starts. These instructions are used to execute boot loaders such as Lilo or Grub. The second part contains 64 bytes which store the partition table. There is space for up to four primary partitions, each one is described in 16 bytes.

In each of these 16 byte partitions description, one of these two following systems can be used: The CHS (cylinder/head/sector) mechanism was used in the past but it's not used any more because it can only address disks up to 8 GB. Now the LBA (logical block addressing) is used because it can address up to 2 TB of data.

The consequence is that the msdos partition table can only support up to 4 primary partitions, and cannot address more than 2 TB of space. The extended partition was created to work around the first limitation, and allows you to have many secondary partitions (called "logical partitions") inside one primary partition (called the "extended partition") but this solution is not very robust and a bit artificial. The second limitation is more problematic because very large disks are now very popular: you can buy a disk such as the Hitachi 2TB (HDS722020ALA330) for less than 200 EUR. This limitation is a real problem for professionals, since many servers with hardware RAID controllers use RAID arrays which are larger than 2TB.

Whether or not to use GPT

The `GPT (GUID Partition Table)` has been introduced with the Extensible Firmware Interface (EFI) standard, and it's the new native layout on new architectures such as Itanium. The GPT layout can also be used on a normal PC with a standard BIOS with Linux as long as you use recent versions of the software involved. Unfortunately, Windows for PC cannot be installed on a disk using the GPT layout. But the 64bit versions of Windows can see partitions of a GPT disk, so you can have a second disk based on GPT as long as Windows is installed on a disk with a traditional msdos partition table.

If you are only using Linux on your computer you can switch to GPT as long as the software involved is recent enough to understand GPT. Unfortunately many Linux installers don't let you choose which partition layout when the disk is formatted. For instance Redhat/Fedora just automatically formats disks with the most appropriate system, depending on how big the disk is. The solution is to use SystemRescueCd to make the partitions yourself using tools such as Parted, GParted of GDisk. Then you can install Linux and you will have to select the option that allows you to preserve the existing partitioning and customize the disk layout.

The GPT layout offers many advantages over msdos partition tables:

- It can address disks which are larger than 2TB (2048 GB)

- You can have more than 4 primary partitions in a native way
- It's more robust: the partition table is redundant
- It supports a wide range of partition types since it's encoded on 16 bytes

Anyway, you should keep a traditional msdos partition table if you want to have Windows installed because Windows for PC cannot boot from GPT disks. If your disk is smaller than 2TB and if you don't want to have more than 4 partitions then you don't really need to move to GPT. The Linux Logical-Volume-Manager is a good method to organize data on your disk and it only uses one partition to store the LVM Physical-Volume.

System programs involved in the disk layout

Multiple system programs are involved in the management of the disk layout, and all of them have to support GPT if you want to use that layout on your disk. Fortunately all of the recent Linux distributions are able to cope with it so you should not have any problem if your Linux distribution has been released in 2009 or later.

- The first program involved in the boot process is the boot loader. Grub is now the default choice and all the important distributions are using it. You can use Grub-0.97 if it has been patched for GPT, and this is the case in general. The Grub2 pre-releases (Grub versions 1.96, 1.97, ...) support GPT natively.
- The linux kernel has to be compiled with support for GPT (option CONFIG_EFI_PARTITION set to yes), this is the case in general
- The disk partitioning tools you may use have to support GPT

In SystemRescueCd-1.0.0 and more recent, the linux kernel and Grub both support GPT. It's recommended to use at least SystemRescueCd-1.3.x anyway because it comes with GDisk (GPT fdisk for Linux) which is an alternative to fdisk that supports GPT. These two programs (fdisk and gdisk) are sometimes necessary when you want to do something which is not supported by Parted/GParted. For instance you need it to resize a partition which has a filesystem that Parted/GParted cannot resize.

In general, the other utilities don't have to support GPT. For instance, you can archive files from a GPT disk with tar or fsarchiver anyway, because these tools are able to see what the kernel can see.

GPT and protective MBR

Unlike the msdos partition labels, the GPT disks don't use a Master Boot Record (MBR), because their partition table is completely different. Anyway you can have the equivalent of an MBR in the first sector of GPT disks.

This is used to prevent those disk tools which are not GPT aware from corrupting your GPT partitions. This protective MBR contains a fake partition of type `EFI GPT` which spans the entire disk with a maximum size of 2TB. Thus old disk tools will see that the space is already allocated and they won't be able to make any modifications to the partitions managed by the GPT layout. This protective MBR also allows standard BIOS-based computers to boot from a GPT disk using a boot loader stored in the protective MBR's code area, since the BIOS only knows the old system based on the MBR. This fake MBR is the only way for a normal PC with a standard BIOS to boot a GPT disk because this is what the BIOS expects.

Here is what happens when we use the standard fdisk (which does not support GPT) on a GPT disk:

```
root@debian /root % fdisk -l /dev/sdb
WARNING: GPT (GUID Partition Table) detected on '/dev/sdb'!
The util fdisk doesn't support GPT. Use GNU Parted.

Disk /dev/sdb: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x2c2fe00b

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *           1      121602   976762583+  ee  EFI GPT
```

BIOS Boot Partition

A boot loader such as Grub is required to boot Linux. This program is particular because it runs before your operating system and then it's a quite critical program. The code of the boot loader is often installed between the MBR and the first partition of the disk. This is not a very good solution since there is not any protection and this code may be overwritten by a system utility which is not aware of that.

For this reason a special partition can be used with GPT disks: it's the `BIOS Boot Partition`. When you have such a partition on your disk, this space is reserved. This is where the boot managers can install their boot code (Grub stage2), and this way you can make modifications to the other partitions of your disk with no risk of damaging Grub. This special partition does not have to be very big. You can just allocate a few megabytes on the disk for such a partition. It will just use a very small portion of your disk, and one entry in your partition table, but this is not a problem with GPT since you can have more than four primary partitions.

If you are about to create a GPT layout on your disk, it's recommended to create a `BIOS Boot Partition` even if you don't plan to use it immediately. It will just prevent many problems you could have with the boot loaders. To create such a partition you can use Parted or GDisk which are both on recent SystemRescueCd versions. Here is how to do that using the Parted command line interface. You have to be careful when you manipulate your partitions, so be sure you know what you are doing first. You have to create a normal partition first, using `mkpart` and then use the parted command called `set` to set the `bios_grub` flag on the partition you have just created.

Here is how to create a new GPT layout on a disk (we use `/dev/sda` for the example) with that partition (all the pre-exising data of that disk will be lost). It may look strange to use ext2 for that partition. Parted wants a type for that partition so we have to give it something, and ext2 is fine.

```
root@sysresccd /root % parted /dev/sda
GNU Parted 1.9.0
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) p
Error: /dev/sda: unrecognised disk label
(parted) mklabel gpt
(parted) mkpart primary ext2 0 10M
(parted) mkpart primary ext4 10M 100%
(parted) set 1 bios_grub on
(parted) p
Model: ATA QEMU HARDDISK (scsi)
Disk /dev/sda: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system  Name     Flags
 1      17.4kB  10.0MB  9983kB               primary  bios_grub
 2      10.0MB  1074MB  1064MB               primary

(parted) quit
Information: You may need to update /etc/fstab.
```

Checking the current layout

You can use Parted from SystemRescueCd or any linux system to see which layout you are currently using. Just use the `print` command in parted. It's safe and it won't make any modification. Here is an example of two disks (`/dev/sda` and `/dev/sdb`). The first one is using an MSDOS layout, and the second one is based on GPT.

```
root@debian /root % parted /dev/sda print
Model: ATA ST31000340AS (scsi)
Disk /dev/sda: 1000GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Number  Start   End     Size    Type     File system  Flags
 1      32.3kB  1024MB  1024MB  primary  ext3
 2      1024MB  18.2GB  17.2GB  primary  reiserfs
 3      18.2GB  35.4GB  17.2GB  primary  ntfs         boot
 4      35.4GB  1000GB  965GB   primary
```

```
root@debian /root % parted /dev/sdb print
Model: ATA SAMSUNG HD103UJ (scsi)
Disk /dev/sdb: 1000GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Number  Start   End     Size    File system  Name     Flags
 1      17.4kB  10.0MB  9983kB               primary  bios_grub
 2      10.0MB  1024MB  1014MB  ext3         boot
 3      1024MB  18.2GB  17.2GB  reiserfs     gentoo
 4      35.4GB  52.6GB  17.2GB  reiserfs     debian
 5      52.6GB  69.7GB  17.2GB  ext3         centos
 6      69.7GB  1000GB  930GB                lvm
```

Installing a patched Grub-0.97

If you have a GPT disk and you have restored a copy of a linux system from somewhere else, you will have to install a boot loader that supports GPT to be able to boot your system. This can be done from SystemRescueCd. Be careful, this is a dangerous operation. You have to know what you are doing. Here is how to install the patched grub-0.97 for GPT using SystemRescueCd:

- boot on SystemRescueCd-1.3.0 or more recent
- type mkdir /mnt/boot
- mount your boot partition (or the / partition if boot is not on a dedicated partition) on /mnt/boot
  - copy your kernel to /mnt/boot/vmlinuz-xxxx
  - mkdir -p /mnt/boot/grub
  - cp /lib/grub/i386-pc/* /mnt/boot/grub/
  - edit /mnt/boot/grub/grub.conf
  - ln -s grub.conf /mnt/boot/grub/menu.lst
  - umount /mnt/boot
  - run grub from the shell
  - use the grub root and setup commands to reinstall it:
    - root (hd0,0) or the grub name of the partition with the grub files (usually /boot)
    - setup (hd0) or (hd1) to install the boot loader on the second hard disk

Retrieved from "http://www.sysresccd.org/Sysresccd-Partitioning-EN-The-new-GPT-disk-layout"

- This page was last modified on 5 August 2010, at 07:30.