

# Festplatte vergrößern

## How to resize/extend a btrfs formatted root partition

### Environment

SUSE Linux Enterprise Server 11 Service Pack 4 (SLES 11 SP4)

SUSE Linux Enterprise Server 12 (all Service Packs)

### Situation

The general procedure to resize a btrfs formatted file system can be found in the SLE 12 Storage Admin Guide. There are however additional points that have to be taken into consideration before executing the resize operation.

1. Is the original disk going to be expanded?
2. Is it possible to add a new disk to expand the existing btrfs file system?

The following article will cover a situation with one disk and two partitions to show the steps necessary for both scenarios using both offline as well as online resize operation.

Please keep in mind that a more sophisticated partitioning scheme (partitions behind the root volume or an extended partition layout) may cause problems. In these cases adding a new disk to expand the existing file system is the preferred solution.

Please check the system setup carefully before carrying out any actions.

**Because the procedures covered in this article contain a fair risk of losing the operating system while performing changes to the partition table, the administrator of the machine is required to create a backup before the operation!**

**This guide does not claim to be complete or cover all possible scenarios. In case of questions please open a service request to discuss these with SUSE Technical Services before any action is taken.**

# Resolution

## Situation:

The disk space of a virtual system with 20GB hard disk should be increased to 40GB.

## Preparation:

1. Since the following procedure requires changes to the partition table, a loss of data is possible. Please ensure to create a backup of the system before performing any action!
2. Ensure your restore procedure works correctly!
3. Check carefully if a `MSDOS` or `GPT` partition table was created. This information can easily be obtained from the `parted -l / fdisk -l` output easily.

**Note:** Only `parted` is able to edit `GPT` partition table. If a `GPT` partition table is used, an online resize is not possible as `parted` uses the `BLKRRPART` ioctl which prevents changes while the partition is mounted.

This article is going to cover the following three approaches to accomplish the resize of a virtual disk in a VMware based environment:

1. Expanding the file system by adding a new disk
2. Resizing the disk using parted
3. Resizing the disk online using fdisk

## Expanding the file system by adding a new disk

A convenient and quick solution to add disk space to an existing `btrfs` file system is by adding a new disk.

The procedure consists of four steps and the system does not need to be rebooted:

1. add a new disk
2. rescan the SCSI bus using

```
rescan-scsi-bus.sh -a
```

3. Add the newly added device to the root `btrfs` filesystem

```
btrfs device add /dev/sdX /
```

4. At this point the metadata is only stored on the first disk, to distribute (balance) it across the devices run:

```
btrfs filesystem balance /
```

## **Resizing the disk using parted**

```
server1:~ # parted -l
Model: VMware Virtual disk (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End   Size  Type  File system  Flags
 1    1049kB 4302MB 4301MB primary linux-swap(v1) type=82
 2    4302MB 21.5GB 17.2GB primary  btrfs      boot, type=83
```

As a first step, the virtual disk needs to be increased on the hypervisor side. Please refer to the vendor documentation for this particular task. The `parted -l` output above also provides the information whether a `MSDOS` or `GPT` partition label was used. `fdisk -l` (an example is provided in the section [Resizing the partition online using fdisk](#) ) will show this information.

Once this has been accomplished, rescan the local disk:

```
server1:~ # echo 1 > /sys/block/sda/device/rescan
server1:~ # parted -l
Model: VMware Virtual disk (scsi)
Disk /dev/sda: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End   Size  Type  File system  Flags
 1    1049kB 4302MB 4301MB primary linux-swap(v1) type=82
 2    4302MB 21.5GB 17.2GB primary  btrfs      boot, type=83
```

The OS now sees the new size of the disk but the partitioning layout needs to be changed to add the remaining 20GB to `/dev/sda2`.

Trying to resize `/dev/sda` using `parted` will fail with:

```
server1:~ # parted /dev/sda
GNU Parted 3.1
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) resize
Partition number? 2
Error: Partition /dev/sda2 is being used. You must unmount it before you modify it with Parted.
(parted)
```

Trying to accomplish the same task in `yast2 disk` fails with:

```
Warning
The file system is currently mounted on /.

You can try to unmount it now, continue without unmounting or cancel.
Click Cancel unless you know exactly what you are doing.
```

Hence this task needs to be accomplished in the rescue system. Please boot from the Service Pack DVD matching the installed SLES version and select "Rescue System" from the main menu. As of SLE 12 SP2, the Rescue System option is located in the "More ..." menu.

Once the rescue system has started, the disk resize operation may be performed as follows:

```
0:rescue:~ # parted /dev/sda
GNU Parted 3.1
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) resize
Partition number? 2
End? [21.5GB]? 42.5GB
(parted) quit
Information: You may need to update /etc/fstab.
```

Running `parted -l` will show the new end of the partition:

```
O:rescue:~ # parted -l
Model: VMware Virtual disk (scsi)
Disk /dev/sda: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```

Number	Start	End	Size	Type	File system	Flags
1	1049kB	4302MB	4301MB	primary	linux-swap(v1)	type=82
2	4302MB	42.5GB	38.2GB	primary	btrfs	boot, type=83

The last steps are to mount the partition and resize the btrfs file system (example below). (If the btrfs file system resides on multiple devices, see the Additional Information section of this document, as well.)

```
O:rescue:~ # mount /dev/sda2 /mnt
O:rescue:~ # btrfs filesystem resize max /mnt
Resize '/mnt' of 'max'
```

```
O:rescue:~ # df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/loop0      29M   29M   0 100% /parts/mp_0000
/dev/loop1      14M   14M   0 100% /parts/mp_0001
devtmpfs        468M   0 468M   0% /dev
/dev/loop2      42M   42M   0 100% /mounts/mp_0000
/dev/loop3      34M   34M   0 100% /mounts/mp_0001
/dev/loop4      4.2M  4.2M   0 100% /mounts/mp_0002
tmpfs           497M   0 497M   0% /dev/shm
tmpfs           497M  7.2M 490M   2% /run
tmpfs           497M   0 497M   0% /sys/fs/cgroup
tmpfs           497M   0 497M   0% /tmp
tmpfs           100M   0 100M   0% /run/user/0
/dev/sda2       36G  753M  33G   3% /mnt
```

Reboot the system back into operation.

## **Resizing the partition online using fdisk**

`fdisk` does not support resizing a partition. In this case the existing root partition needs to be deleted and recreated using the same start block but selecting the new end block to assign all available disk space to the partition.

As already mentioned before, `fdisk` cannot deal with `GPT` partition tables. Please check carefully which label was chosen and select the right tool for the resize operation.

When recreating the partition please make sure to set the bootable flag again, otherwise the system will not boot.

The procedure using `fdisk` is as follows:

Print the current partition table, save it, expand the disk, have the kernel rescan the device and make sure it sees the new size:

```
server1:~ # fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x0004a8ed

Device  Boot  Start    End  Sectors  Size Id Type
/dev/sda1      2048  8402943  8400896   4G 82 Linux swap / Solaris
/dev/sda2 * 8402944 41943039 33540096  16G 83 Linux

btrfs:~ # echo 1 > /sys/block/sda/device/rescan
btrfs:~ # fdisk -l
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x0004a8ed

Device  Boot  Start    End  Sectors  Size Id Type
/dev/sda1      2048  8402943  8400896   4G 82 Linux swap / Solaris
/dev/sda2 * 8402944 41943039 33540096  16G 83 Linux
server1:~ #
```

**Keep in mind, operations in fdisk are temporary until a write operation is issued. So at any point it is safe to exit fdisk using CTRL+c.**

As a next step open fdisk and delete the root partition.

```
server1:~ # fdisk /dev/sda

Welcome to fdisk (util-linux 2.28).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): d
Partition number (1,2, default 2): 2

Partition 2 has been deleted.
```

Now recreate partition #2 as a primary partition with partition ID 83 (Linux partition)

```
Command (m for help): n
Partition type
  p  primary (1 primary, 0 extended, 3 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (8402944-83886079, default 8402944):
Last sector, +sectors or +size{K,M,G,T,P} (8402944-83886079, default 83886079):
```

Created a new partition 2 of type 'Linux' and of size 36 GiB.

```
Command (m for help): t
Partition number (1,2, default 2): 2
Partition type (type L to list all types): 83
```

Changed type of partition 'Linux' to 'Linux'.

As shown above, `fdisk` picked up the correct start and end block by itself. The next step is to enable the boot-flag:

```
Command (m for help): a
Partition number (1,2, default 2): 2
The bootable flag on partition 2 is enabled now.
```

After the changes above, the new partition table will be as follows:

```
Command (m for help): p
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x0004a8ed

Device  Boot  Start    End  Sectors  Size Id Type
/dev/sda1      2048 8402943 8400896    4G 82 Linux swap / Solaris
/dev/sda2 * 8402944 83886079 75483136   36G 83 Linux
```

Now, while writing the changes to disk, the following messages will be shown:

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Re-reading the partition table failed.: Device or resource busy

The kernel still uses the old table. The new table will be used at the next reboot or after you run partprobe(8) or kpartx(8).

server1:~ # partprobe
Error: Partition(s) 2 on /dev/sda have been written, but we have been unable to inform the kernel of the change, please reboot after resizing.
Error: Can't have a partition outside the disk!
```

SLES 11 SP4 based systems need to be rebooted at this point as the kernel does not support re-reading the partition table using `BLKPG_DEL_PARTITION` and `BLKPG_RESIZE_PARTITION` ioctls. Once the system is back online please run:

```
btrfs filesystem resize max /
```

to expand the file system. (If the btrfs file system resides on multiple devices, see the Additional Information section of this document, as well.)

SLE 12 based systems support the ioctls mentioned previously, this way it is possible to notify the kernel about the changed partition table as the system is online.

Old partition table:

```
server1:~ # fdisk -l /dev/sda
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00dd878

Device    Boot  Start      End  Sectors  Size Id Type
/dev/sda1             2048  8402943  8400896    4G 82 Linux swap / Solaris
/dev/sda2 *    8402944 41943039 33540096   16G 83 Linux
```

Execute the actions as displayed above to remove and recreate the partition table, write the changes and receive the busy message from the kernel:

```
server1:~ # fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
```

```
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x000dd878
```

```
Device  Boot  Start    End  Sectors  Size Id Type
/dev/sda1      2048  8402943  8400896   4G 82 Linux swap / Solaris
/dev/sda2 * 8402944 104857599 96454656  46G 83 Linux
```

So `/dev/sda2` was resized from 20GB to 50GB, check the kernel's view on the partitions:

```
server1:~ # cat /proc/partitions
major minor #blocks name

 2     0      4 fd0
 8     0 52428800 sda
 8     1 4200448 sda1
 8     2 16770048 sda2
11     0 1048575 sr0
```

now update the kernel's view of `/dev/sda2` using:

```
partx -u -n 2 /dev/sda
```

and check the partitions again:

```
server1:~ # cat /proc/partitions
major minor #blocks name

 2     0      4 fd0
 8     0 52428800 sda
 8     1 4200448 sda1
 8     2 48227328 sda2
11     0 1048575 sr0
```

Now resize the filesystem. (If the btrfs file system resides on multiple devices, see the Additional Information section of this document, as well.)

```
btrfs filesystem resize max /
```

Running:

```
btrfs filesystem usage /
```

will now show the new size of the filesystem.

Using `partx` it would also be possible to delete any partitions that are not needed behind the root volume, expand the root volume, delete the kernel view on that partition (`partx -d -n X /dev/sda`), update the root partition (`partx -u -n X /dev/sda`) and then resize the filesystem.

## Additional Information

### Considerations when adding new devices into a btrfs file system

Another possible method to extend the disk space of a btrfs file system would be to add unused partitions from a disk. Please note that btrfs will treat these partitions (even if they come from the same device) as a separate physical volume and if later the file system should operate in RAID mode, chunks will be served from both partitions which is not desirable. In this case it is preferable to either add complete disks or delete unused partitions and resize the root volume where applicable.

### If the btrfs file system resides on multiple devices

When a btrfs file system resides on multiple devices, first determine the devid of resized partition, for example:

```
btrfs filesystem show /
```

That will list all partitions in use, and number them. Then, for example, if the partition in question is devid 2, the appropriate command is:

```
btrfs filesystem resize 2:max /
```

---

Revision #1

Created 21 March 2023 08:06:01 by Hermann

Updated 21 March 2023 08:06:21 by Hermann