NAME

virt−sparsify − Make a virtual machine disk sparse

SYNOPSIS

```
virt−sparsify [−−options] indisk outdisk

virt−sparsify [−−options] −−in−place disk
```

WARNING

Using `virt−sparsify` on live virtual machines, or concurrently with other disk editing tools, can be dangerous, potentially causing disk corruption. The virtual machine must be shut down before you use this command, and disk images must not be edited concurrently.

DESCRIPTION

Virt-sparsify is a tool which can make a virtual machine disk (or any disk image) sparse a.k.a. thin-provisioned. This means that free space within the disk image can be converted back to free space on the host.

Virt-sparsify can locate and sparsify free space in most filesystems (eg. ext2/3/4, btrfs, NTFS, etc.), and also in LVM physical volumes.

Virt-sparsify can also convert between some disk formats, for example converting a raw disk image to a thin-provisioned qcow2 image.

Virt-sparsify can operate on any disk image, not just ones from virtual machines. However if a virtual machine has multiple disks and uses volume management, then virt-sparsify will work but not be very effective (http://bugzilla.redhat.com/887826).

IMPORTANT NOTE ABOUT SPARSE OUTPUT IMAGES

If the input is raw, then the default output is raw sparse. **Y ou must check the output size using a tool that understands sparseness** such as `du −sh`. It can make a huge difference:

```
$ ls −lh test1.img
−rw−rw−r−−. 1 rjones rjones 100M Aug  8 08:08 test1.img
$ du −sh test1.img
3.6M   test1.img
```

(Compare the apparent size **100M** vs the actual size **3.6M**)

IMPORTANT LIMITATIONS

- The virtual machine *must be shut down* before using this tool.
- Virt-sparsify may require up to 2x the virtual size of the source disk image (1 temporary copy + 1 destination image). This is in the worst case and usually much less space is required.

  If you are using the *−−in−place* option, then large amounts of temporary space are **not** required.
- Virt-sparsify cannot resize disk images. To do that, use **virt−resize** (1).
- Virt-sparsify cannot handle encrypted disks. Libguestfs supports encrypted disks, but encrypted disks themselves cannot be sparsified.
- Virt-sparsify cannot yet sparsify the space between partitions. Note that this space is often used for critical items like bootloaders so it's not really unused.
- In copy mode, qcow2 internal snapshots are not copied over to the destination image.

You may also want to read the manual pages for the associated tools **virt−filesystems** (1) and **virt−df** (1) before starting.

EXAMPLES

Typical usage is:

```
virt−sparsify indisk outdisk
```

which copies `indisk` to `outdisk`, making the output sparse. `outdisk` is created, or o verwritten if it

already exists. The format of the input disk is detected (eg. qcow2) and the same format is used for the output disk.

To convert between formats, use the *--convert* option:

```
virt-sparsify disk.raw --convert qcow2 disk.qcow2
```

Virt-sparsify tries to zero and sparsify free space on every filesystem it can find within the source disk image. You can get it to ignore (don't zero free space on) certain filesystems by doing:

```
virt-sparsify --ignore /dev/sda1 indisk outdisk
```

See **virt-filesystems**(1) to get a list of filesystems within a disk image.

Since virt-sparsify ≥ 1.26, you can now sparsify a disk image in place by doing:

```
virt-sparsify --in-place disk.img
```

OPTIONS

--help
Display help.

--check-tmpdir ignore
--check-tmpdir continue
--check-tmpdir warn
--check-tmpdir fail
Check if ''TMPDIR'' or *--tmp* directory has enough space to complete the operation. This is just an estimate.

If the check indicates a problem, then you can either:

- **ignore** it,
- print a warning and **continue**,
- **warn** and wait for the user to press the Return key (this is the default), or:
- **fail** and exit.

You cannot use this option and *--in-place* together.

--colors
--colours
Use ANSI colour sequences to colourize messages. This is the default when the output is a tty. If the output of the program is redirected to a file, ANSI colour sequences are disabled unless you use this option.

--compress
Compress the output file. This *only* works if the output format is `qcow2`.

You cannot use this option and *--in-place* together.

--convert raw
--convert qcow2
--convert [other formats]
Use `output-format` as the format for the destination image. If this is not specified, then the input format is used.

Supported and known-working output formats are: `raw`, `qcow2`, `vdi`.

You can also use any format supported by the **qemu-img**(1) program, eg. `vmdk`, but support for other formats is reliant on qemu.

Specifying the *--convert* option is usually a good idea, because then virt-sparsify doesn't need to try to guess the input format.

For fine-tuning the output format, see: *--compress*, *-o*.

You cannot use this option and *--in-place* together.

--echo-keys
When prompting for keys and passphrases, virt-sparsify normally turns echoing off so you cannot see what you are typing. If you are not worried about Tempest attacks and there is no one else in the room you can specify this flag to see what you are typing.

--format raw
--format qcow2
Specify the format of the input disk image. If this flag is not given then it is auto-detected from the image itself.

If working with untrusted raw-format guest disk images, you should ensure the format is always specified.

--ignore filesystem
--ignore volgroup
Ignore the named filesystem.

When not using *--in-place*: Free space on the filesystem will not be zeroed, but existing blocks of zeroes will still be sparsified.

When using *--in-place*, the filesystem is ignored completely.

In the second form, this ignores the named volume group. Use the volume group name without the */dev/* prefix, eg. *--ignore vg_foo*

You can give this option multiple times.

--in-place
Do in-place sparsification instead of copying sparsification. See ''IN-PLACE SPARSIFICATION'' below.

--key SELECTOR
Specify a key for LUKS, to automatically open a LUKS device when using the inspection. `ID` can be either the libguestfs device name, or the UUID of the LUKS device.

--key `ID`:key:KEY_STRING
Use the specified `KEY_STRING` as passphrase.

--key `ID`:file:FILENAME
Read the passphrase from *FILENAME*.

--keys-from-stdin
Read key or passphrase parameters from stdin. The default is to try to read passphrases from the user by opening */dev/tty*.

If there are multiple encrypted devices then you may need to supply multiple keys on stdin, one per line.

--machine-readable
--machine-readable=format
This option is used to make the output more machine friendly when being parsed by other programs. See ''MACHINE READABLE OUTPUT'' below.

-o option[,option,...]
Pass *-o* option(s) to the **qemu-img**(1) command to fine-tune the output format. Options available depend on the output format (see *--convert*) and the installed version of the qemu-img program.

You should use *-o* at most once. To pass multiple options, separate them with commas, eg:

```
virt-sparsify --convert qcow2 \
  -o cluster_size=512,preallocation=metadata ...
```

You cannot use this option and *--in-place* together.

−q
−−quiet
This disables progress bars and other unnecessary output.

−−tmp block_device
−−tmp dir
In copying mode only, use the named device or directory as the location of the temporary overlay (see also ‘‘TMPDIR’’ below).

If the parameter given is a block device, then the block device is written to directly. **Note this erases the existing contents of the block device**.

If the parameter is a directory, then this is the same as setting the ‘‘TMPDIR’’ environment variable.

You cannot use this option and *−−in−place* together.

−−tmp prebuilt:file
In copying mode only, the specialized option *−−tmp prebuilt:file* (where `prebuilt:` is a literal string) causes virt-sparsify to use the qcow2 `file` as temporary space.

- The file **must** be freshly formatted as qcow2, with indisk as the backing file.
- If you rerun virt-sparsify, you **must** recreate the file before each run.
- Virt-sparsify does not delete the file.

This option is used by oVirt which requires a specially formatted temporary file.

−v
−−verbose
Enable verbose messages for debugging.

−V
−−version
Display version number and exit.

−x Enable tracing of libguestfs API calls.

−−zero partition
−−zero logvol
Zero the contents of the named partition or logical volume in the guest. All data on the device is lost, but sparsification is excellent! You can give this option multiple times.

IN-PLACE SPARSIFICATION

Since virt-sparsify ≥ 1.26, the tool is able to do in-place sparsification (instead of copying from an input disk to an output disk). This is more efficient. It is not able to recover quite as much space as copying sparsification.

To use this mode, specify a disk image which will be modified in place:

```
virt−sparsify −−in−place disk.img
```

Some options are not compatible with this mode: *−−convert*, *−−compress* and *−o* because they require wholesale disk format changes; *−−check−tmpdir* because large amounts of temporary space are not required.

In-place sparsification works using discard (a.k.a trim or unmap) support.

MACHINE READABLE OUTPUT

The *−−machine−readable* option can be used to make the output more machine friendly, which is useful when calling virt-sparsify from other programs, GUIs etc.

There are two ways to use this option.

Firstly use the option on its own to query the capabilities of the virt-sparsify binary. Typical output looks like this:

```
$ virt-sparsify --machine-readable
virt-sparsify
ntfs
btrfs
```

A list of features is printed, one per line, and the program exits with status 0.

Secondly use the option in conjunction with other options to make the regular program output more machine friendly.

At the moment this means:

1. Progress bar messages can be parsed from stdout by looking for this regular expression:

   ```
   ^[0-9]+/[0-9]+$
   ```

2. The calling program should treat messages sent to stdout (except for progress bar messages) as status messages. They can be logged and/or displayed to the user.

3. The calling program should treat messages sent to stderr as error messages. In addition, virt-sparsify exits with a non-zero status code if there was a fatal error.

All versions of virt-sparsify have supported the *--machine-readable* option.

It is possible to specify a format string for controlling the output; see ‘‘ADVANCED MACHINE READABLE OUTPUT’’ in **guestfs** (3).

WINDOWS 8

Windows 8 ‘‘fast startup’’ can prevent virt-sparsify from working. See ‘‘WINDOWS HIBERNATION AND WINDOWS 8 FAST STARTUP’’ in **guestfs** (3).

ENVIRONMENT VARIABLES

TMPDIR

Location of the temporary directory used for the potentially large temporary overlay file.

In virt-sparsify ≥ 1.28, you can override this environment variable using the *--tmp* option.

You should ensure there is enough free space in the worst case for a full copy of the source disk (*virtual* size), or else set `$TMPDIR` to point to another directory that has enough space.

This defaults to */tmp*.

Note that if `$TMPDIR` is a tmpfs (eg. if */tmp* is on tmpfs, or if you use `TMPDIR=/dev/shm`), tmpfs defaults to a maximum size of *half* of physical RAM. If virt-sparsify exceeds this, it will hang. The solution is either to use a real disk, or to increase the maximum size of the tmpfs mountpoint, eg:

```
mount -o remount,size=10G /tmp
```

If you are using the *--in-place* option, then large amounts of temporary space are **not** required.

For other environment variables, see ‘‘ENVIRONMENT VARIABLES’’ in **guestfs** (3).

EXIT STATUS

This program returns 0 if the operation completed without errors. (This doesn’t necessarily mean that space could be freed up.)

A non-zero exit code indicates an error.

If the exit code is `3` and the *--in-place* option was used, that indicates that discard support is not available in libguestfs, so copying mode must be used instead.

SEE ALSO

virt-df (1), **virt-filesystems** (1), **virt-resize** (1), **virt-rescue** (1), **guestfs** (3), **guestfish** (1), **truncate** (1), **fallocate** (1), **qemu-img** (1), http://libguestfs.org/.

AUTHOR

Richard W.M. Jones http://people.redhat.com/~rjones/

COPYRIGHT

LICENSE

BUGS

To get a list of bugs against libguestfs, use this link: https://bugzilla.redhat.com/buglist.cgi?component=libguestfs&product=Virtualization+Tools

To report a new bug against libguestfs, use this link: https://bugzilla.redhat.com/enter_bug.cgi?component=libguestfs&product=Virtualization+Tools

When reporting a bug, please supply:

- The version of libguestfs.
- Where you got libguestfs (eg. which Linux distro, compiled from source, etc)
- Describe the bug accurately and give a way to reproduce it.
- Run **libguestfs−test−tool** (1) and paste the **complete, unedited** output into the bug report.