NAME

pvmove – Move extents from one physical volume to another

SYNOPSIS

pvmove *position_args*
  [*option_args*]
  [*position_args*]

DESCRIPTION

pvmove moves the allocated physical extents (PEs) on a source PV to one or more destination PVs. You can optionally specify a source LV in which case only extents used by that LV will be moved to free (or specified) extents on the destination PV. If no destination PV is specified, the normal allocation rules for the VG are used.

If pvmove is interrupted for any reason (e.g. the machine crashes) then run pvmove again without any PV arguments to restart any operations that were in progress from the last checkpoint. Alternatively, use the abort option at any time to abort the operation. The resulting location of LVs after an abort depends on whether the atomic option was used.

More than one pvmove can run concurrently if they are moving data from different source PVs, but additional pvmoves will ignore any LVs already in the process of being changed, so some data might not get moved.

USAGE

Move PV extents.

pvmove *PV*
    [**–A**|**––autobackup y**|**n**]
    [**–n**|**––name** *LV*]
    [**––alloc contiguous**|**cling**|**cling_by_tags**|**normal**|**anywhere**|**inherit**]
    [**––atomic**]
    [**––noudevsync**]
    [**––reportformat basic**|**json**]
    [COMMON_OPTIONS]
    [*PV* ...]

Continue or abort existing pvmove operations.

pvmove
    [COMMON_OPTIONS]

Common options for command:
    [**–b**|**––background**]
    [**–i**|**––interval** *Number*]
    [**––abort**]

Common options for lvm:
    [**–d**|**––debug**]
    [**–h**|**––help**]
    [**–q**|**––quiet**]
    [**–t**|**––test**]
    [**–v**|**––verbose**]
    [**–y**|**––yes**]
    [**––commandprofile** *String*]
    [**––config** *String*]
    [**––driverloaded y**|**n**]
    [**––lockopt** *String*]

[**−−longhelp**]
[**−−nolocking**]
[**−−profile** *String*]
[**−−version**]

OPTIONS

−−abort
Abort any pvmove operations in progress. If a pvmove was started with the −−atomic option, then all LVs will remain on the source PV. Otherwise, segments that have been moved will remain on the destination PV, while unmoved segments will remain on the source PV.

−−alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit
Determines the allocation policy when a command needs to allocate Physical Extents (PEs) from the VG. Each VG and LV has an allocation policy which can be changed with vgchange/lvchange, or overriden on the command line. **normal** applies common sense rules such as not placing parallel stripes on the same PV. **inherit** applies the VG policy to an LV. **contiguous** requires new PEs be placed adjacent to existing PEs. **cling** places new PEs on the same PV as existing PEs in the same stripe of the LV. If there are sufficient PEs for an allocation, but normal does not use them, **anywhere** will use them even if it reduces performance, e.g. by placing two stripes on the same PV. Optional positional PV args on the command line can also be used to limit which PVs the command will use for allocation. See **lvm**(8) for more information about allocation.

−−atomic
Makes a pvmove operation atomic, ensuring that all affected LVs are moved to the destination PV, or none are if the operation is aborted.

−A|−−autobackup y|n
Specifies if metadata should be backed up automatically after a change. Enabling this is strongly advised! See **vgcfgbackup**(8) for more information.

−b|−−background
If the operation requires polling, this option causes the command to return before the operation is complete, and polling is done in the background.

−−commandprofile *String*
The command profile to use for command configuration. See **lvm.conf**(5) for more information about profiles.

−−config *String*
Config settings for the command. These override lvm.conf settings. The String arg uses the same format as lvm.conf, or may use section/field syntax. See **lvm.conf**(5) for more information about config.

−d|−−debug ...
Set debug level. Repeat from 1 to 6 times to increase the detail of messages sent to the log file and/or syslog (if configured).

−−driverloaded y|n
If set to no, the command will not attempt to use device-mapper. For testing and debugging.

−h|−−help
Display help text.

−i|−−interval *Number*
Report progress at regular intervals.

−−lockopt *String*
Used to pass options for special cases to lvmlockd. See **lvmlockd**(8) for more information.

−−longhelp
Display long help text.

−n|−−name *String*
Move only the extents belonging to the named LV.

−−nolocking
Disable locking.

−−noudevsync
Disables udev synchronisation. The process will not wait for notification from udev. It will continue irrespective of any possible udev processing in the background. Only use this if udev is not running or has rules that ignore the devices LVM creates.

−−profile *String*
An alias for −−commandprofile or −−metadataprofile, depending on the command.

−q|−−quiet ...
Suppress output and log messages. Overrides −−debug and −−verbose. Repeat once to also suppress any prompts with answer 'no'.

−−reportformat basic|json
Overrides current output format for reports which is defined globally by the report/output_format setting in lvm.conf. **basic** is the original format with columns and rows. If there is more than one report per command, each report is prefixed with the report name for identification. **json** produces report output in JSON format. See **lvmreport**(7) for more information.

−t|−−test
Run in test mode. Commands will not update metadata. This is implemented by disabling all metadata writing but nevertheless returning success to the calling function. This may lead to unusual error messages in multi-stage operations if a tool relies on reading back metadata it believes has changed but hasn't.

−v|−−verbose ...
Set verbose level. Repeat from 1 to 4 times to increase the detail of messages sent to stdout and stderr.

−−version
Display version information.

−y|−−yes
Do not prompt for confirmation interactively but always assume the answer yes. Use with extreme caution. (For automatic no, see −qq.)

VARIABLES

PV
Physical Volume name, a device path under /dev. For commands managing physical extents, a PV positional arg generally accepts a suffix indicating a range (or multiple ranges) of physical extents (PEs). When the first PE is omitted, it defaults to the start of the device, and when the last PE is omitted it defaults to end. Start and end range (inclusive): *PV*[**:***PE***−***PE*]... Start and length range (counting from 0): *PV*[**:***PE***+***PE*]...

String
See the option description for information about the string content.

Size[UNIT]
Size is an input number that accepts an optional unit. Input units are always treated as base two values, regardless of capitalization, e.g. 'k' and 'K' both refer to 1024. The default input unit is specified by letter, followed by |UNIT. UNIT represents other possible input units: **bBsSkKmMgGtTpPeE**. b|B is bytes, s|S is sectors of 512 bytes, k|K is kilobytes, m|M is megabytes, g|G is gigabytes, t|T is terabytes, p|P is petabytes, e|E is exabytes. (This should not be confused with the output control −−units, where capital letters mean multiple of 1000.)

ENVIRONMENT VARIABLES

See **lvm**(8) for information about environment variables used by lvm. For example, LVM_VG_NAME can generally be substituted for a required VG parameter.

NOTES

pvmove works as follows:

1. A temporary 'pvmove' LV is created to store details of all the data movements required.

2. Every LV in the VG is searched for contiguous data that need moving according to the command line arguments. For each piece of data found, a new segment is added to the end of the pvmove LV. This segment takes the form of a temporary mirror to copy the data from the original location to a newly allocated location. The original LV is updated to use the new temporary mirror segment in the pvmove LV instead of accessing the data directly.

3. The VG metadata is updated on disk.

4. The first segment of the pvmove LV is activated and starts to mirror the first part of the data. Only one segment is mirrored at once as this is usually more efficient.

5. A daemon repeatedly checks progress at the specified time interval. When it detects that the first temporary mirror is in sync, it breaks that mirror so that only the new location for that data gets used and writes a checkpoint into the VG metadata on disk. Then it activates the mirror for the next segment of the pvmove LV.

6. When there are no more segments left to be mirrored, the temporary LV is removed and the VG metadata is updated so that the LVs reflect the new data locations.

Note that this new process cannot support the original LVM1 type of on-disk metadata. Metadata can be converted using **vgconvert**(8).

If the **−−atomic** option is used, a slightly different approach is used for the move. Again, a temporary 'pvmove' LV is created to store the details of all the data movements required. This temporary LV contains all the segments of the various LVs that need to be moved. However, in this case, an identical LV is allocated that contains the same number of segments and a mirror is created to copy the contents from the first temporary LV to the second. After a complete copy is made, the temporary LVs are removed, leaving behind the segments on the destination PV. If an abort is issued during the move, all LVs being moved will remain on the source PV.

EXAMPLES

Move all physical extents that are used by simple LVs on the specified PV to free physical extents elsewhere in the VG.
pvmove /dev/sdb1

Use a specific destination PV when moving physical extents.
pvmove /dev/sdb1 /dev/sdc1

Move extents belonging to a single LV.
pvmove −n lvol1 /dev/sdb1 /dev/sdc1

Rather than moving the contents of an entire device, it is possible to move a range of physical extents, for example numbers 1000 to 1999 inclusive on the specified PV.
pvmove /dev/sdb1:1000−1999

A range of physical extents to move can be specified as start+length. For example, starting from PE 1000.

(Counting starts from 0, so this refers to the 1001st to the 2000th PE inclusive.)
pvmove /dev/sdb1:1000+1000

Move a range of physical extents to a specific PV (which must have sufficient free extents).
pvmove /dev/sdb1:1000−1999 /dev/sdc1

Move a range of physical extents to specific new extents on a new PV.
pvmove /dev/sdb1:1000−1999 /dev/sdc1:0−999

If the source and destination are on the same disk, the **anywhere** allocation policy is needed.
pvmove −−alloc anywhere /dev/sdb1:1000−1999 /dev/sdb1:0−999

The part of a specific LV present within in a range of physical extents can also be picked out and moved.
pvmove −n lvol1 /dev/sdb1:1000−1999 /dev/sdc1

SEE ALSO

lvm(8) **lvm.conf**(5) **lvmconfig**(8)

pvchange(8) **pvck**(8) **pvcreate**(8) **pvdisplay**(8) **pvmove**(8) **pvremove**(8) **pvresize**(8) **pvs**(8) **pvscan**(8)

vgcfgbackup(8) **vgcfgrestore**(8) **vgchange**(8) **vgck**(8) **vgcreate**(8) **vgconvert**(8) **vgdisplay**(8) **vgexport**(8) **vgextend**(8) **vgimport**(8) **vgimportclone**(8) **vgmerge**(8) **vgmknodes**(8) **vgreduce**(8) **vgremove**(8) **vgrename**(8) **vgs**(8) **vgscan**(8) **vgsplit**(8)

lvcreate(8) **lvchange**(8) **lvconvert**(8) **lvdisplay**(8) **lvextend**(8) **lvreduce**(8) **lvremove**(8) **lvrename**(8) **lvresize**(8) **lvs**(8) **lvscan**(8)

lvm-fullreport(8) **lvm-lvpoll**(8) **lvm2−activation−generator**(8) **blkdeactivate**(8) **lvmdump**(8)

dmeventd(8) **lvmpolld**(8) **lvmlockd**(8) **lvmlockctl**(8) **cmirrord**(8) **lvmdbusd**(8)

lvmsystemid(7) **lvmreport**(7) **lvmraid**(7) **lvmthin**(7) **lvmcache**(7)