Cloper utility

Compressing and Optimizing a Virtual Machine's Undoable Disks
(Cloning optimizer utility)

Introduction.

The Cloper utility is a console program for optimizing rollback virtual disks in order to reduce the size of their file on the physical disk. The x86 version of the utility is intended for use on 32-bit and 64-bit OS; in the case of a 64-bit guest OS, the x64 version of the utility can be used. The utility was tested on VMWare Workstation v.8.0, but its operating scheme should be applicable to all variants of virtual machine players, if they do not initialize the entire allocated disk space when creating it, but only store and record actual data. If, during the initial initialization of a virtual disk, the machine player allocates for it as much space as is specified for the disk volume, then using this utility becomes meaningless.

The utility is used on a virtual machine with Windows OS, to which the source disk and the target etalon disk are connected in rollback or undoable mode. The source and target disks for creating an optimized snapshot must be copies of the same virtual disk, and the target disk contains only the original etalon data without the added information in the source disk.

In general, the optimization procedure consists of comparing the data of two disk copies with low-level processing of the source disk data so that the physical data of its files are located, if possible, opposite the same physical data on the target reference disk. After the comparison procedures are completed, the data of the source disk is cloned to the etalon disk. Since cloning only records data that differs from the contents of the target reference disk, its resulting file size on the physical disk is significantly reduced.

If further use of the original virtual disk is expected, it is recommended to perform optimization after saving a copy of it. The nature of actions with its data is conventional fragmentation and the probability of their damage is not higher than the probability of file damage when copying to another folder, but due to internal data movement, previously unused sectors of the virtual disk are initialized and as a result, the size of its physical file grows to the actual disk volume.

Optimization procedures.

The optimization procedure can be applied to virtual disks with MBR and GPT partition layout formats. The mapping operation and other optimization actions are applied only to partitions with the NTFS file system, partitions with other file systems are cloned as is. The utility provides the following procedures for optimizing NTFS volumes:

1. Preliminary deletion of the source files specified for this purpose, exclusion from the procedures of identical files with the same cluster arrangement in the source and in the standard;
2. NTFS compression of new and changed data with their consolidation (i.e. if specified, it also moves this source data in front of etalon data without matches);
3. Finds source file cluster matches among unmatched target file clusters and moves these source file data in front of matched target clusters;
4. Transfer of non-zero sectors of the etalon to file-free areas of the source volume;
5. Because during cloning only non-zero sectors are transferred, the remaining "non-zero" free sectors of the source volume are zeroed out.

Prerequisites.

To be able to perform optimization task the guest service OS must enable access to target & source disks using Windows API functionality, the active OS user of running task must have administrative privileges. The tool tasks can be started using Windows OS versions starting with XP SP2 x86/x64. It's recommended to use older OS versions in order to minimize background modifications of disk data by system during optimization task.

Regardless of the task specification the tool actively works with data of source disk during all subtasks. The host system must be able to support these activities. To have optimal performance it is recommended to keep source & target disk snapshots on different hard drives.

Depending on guest performance, amount of source & target data, task specification, the entire procedure can take from 10-20 minutes up to several hours. The resulting optimized snapshot can be obtained only after completion of all specified procedures. To avoid some background initializations of target data, it is recommended to power off guest VM or unmount target disk immediately after completion of task.

Depending on task parameterization the system requirements to guest VM can be minimum or can require significant amount of RAM. The minimum RAM amount for running guest OS is prerequisite for all tasks except the subtask, which searches the matches of source data inside target data. The latter subtask can require up to 2 GB of RAM only for task process, it also notably consumes CPU time during search activities.

Utility parameters.

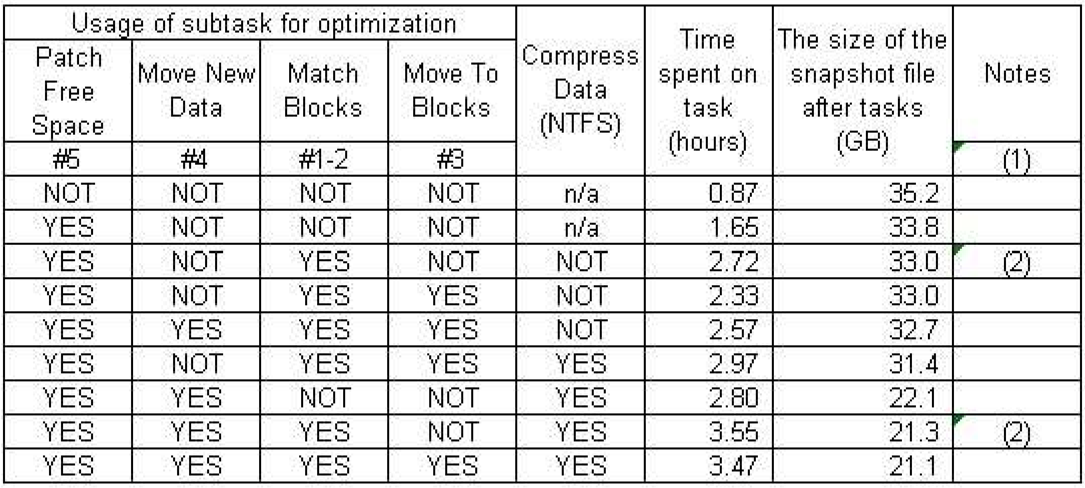
The Cloper utility is a console program with command line parameters and a configuration file. Calling it without parameters on the command line displays a list of them with a description. The following command line parameters are provided:

1. "/help" or "/h" - displays help on using the utility;
2. "/getconfig:short" or "/gc:short" - call to create a configuration file with default parameters. The suffix ":short" is optional, if it is not specified, a configuration file will be created with comments for each parameter and its value;
3. "/getserials:<drive letter>" or "/gs:<drive letter>" - gets the serial identifier of the disk, which consists of substrings of the identifiers of all its volumes separated by the "-" symbol. In order to get the identifier of a disk, it must have at least one volume with a letter assigned to it. Since the optimization procedure is performed using two copies of the same disk, their volume identifiers are the same;
4. "/serials:<serials>" or "/s:<serials>" - starts task for two disks with specified serials identifier;
5. "/serials:<serials> /showdiskdirs|sdd:<folders filter>" or "/[s]:<serials> /[sdd]:<folders filter>" - serves to display the list of folders according to the specified filter. This special call serves for additional control of the adequacy of the specified folder selection filter to the expected result. The value of the object selection filter for their preliminary deletion is specified in the configuration file in its section "[VolumeCleanup]". Also, see the help "/h" with section numbers 7-9 for additional details;
6. "/serials:<serials> /showdiskfiles|sdd:<files filter>" or "/[s]:<serials> /[sdf]:<files filter>" - the call is similar to the previous one, but to check the selection of files;
7. "/[signdisk]:<new signature>" or "/[sd]:<new signature>" - call to interactively change the disk signature. If no new value is specified, then lists connected disks with their signatures.

When creating a configuration file, its values are set to the optimal value, if a file with comments is created, it has line-by-line help. When you first run the utility with purpose of optimization tasks on a virtual machine, it is recommended to create a new configuration file. To run the utility from automation scripts, the "RunMode" parameter must be changed to "SILENT", and the "PauseAtEnd" parameter to "NOT".

Testing and its results.

Changing the parameters affects the efficiency of compression of the rollback disk file. For demonstration purposes, the utility was run with the same data, but with changing the parameters. The sample data for comparison contained a undoable disk snapshot, its file size was 52.4 GB. The virtual disk had an internal size of 75 GB and two main partitions. The first small partition of the Windows BCD was excluded from the optimization procedures, the second partition contained the installed Windows 10 x64 OS. The target reference disk did not contain any other data except the installed OS and a few programs, the source disk had about 40 GB of user data on the second partition. The test results in the table below were obtained using a virtual machine running Windows 7 x64 OS:



Usage of subtask for optimization				Compress Data (NTFS)	Time spent on task (hours)	The size of the snapshot file after tasks (GB)	Notes
Patch Free Space	Move New Data	Match Blocks	Move To Blocks				
#5	#4	#1-2	#3				(1)
NOT	NOT	NOT	NOT	n/a	0.87	35.2	
YES	NOT	NOT	NOT	n/a	1.65	33.8	
YES	NOT	YES	NOT	NOT	2.72	33.0	(2)
YES	NOT	YES	YES	NOT	2.33	33.0	
YES	YES	YES	YES	NOT	2.57	32.7	
YES	NOT	YES	YES	YES	2.97	31.4	
YES	YES	NOT	NOT	YES	2.80	22.1	
YES	YES	YES	NOT	YES	3.55	21.3	(2)
YES	YES	YES	YES	YES	3.47	21.1	

[1] The "Move To Blocks" processing corresponds to moving new data opposite to the unmatched data of the standard (the "MoveNewData" and "ConsolidateNewData" fields of the configuration file).

[2] The increase in task execution time in these cases is caused by significant fragmentation of the free space of the source after matching the source and target data blocks.

It should be noted that the table contains only the results of cluster-by-cluster block comparison when executing subtasks #1-2. Segment-by-segment comparison takes more time and RAM during these subtasks. Segment comparison makes sense only if the host system has sector-sized clusters of it's physical disk. At least in the case of VMware, guest data storage is most likely performed in 4 KB blocks and coincides with the standard cluster size. Cases of different sizes of file system clusters of the guest disk volume and the host disk volume were not tested, that is in this case, certain side effects are possible and the total size of the found comparisons may not coincide with the reduction in the size of the virtual disk image file.

Due to background system activity, the difference in the final file size becomes significant only starting from 100-200 MB. A virtual machine running Windows XP SP2 gives similar results with a slight improvement due to lower background activity of system drivers. Thus, for the best optimization result, it is recommended to use a virtual machine with older OS versions and a minimum set of third-party software.

Also, to achieve the best compression effect, it is recommended to keep a set of large files (for example, some archives) in the etalon disk, which can be deleted either when the user starts working with the source disk, or when the optimization procedures are started by the utility. If the size of these files approximately corresponds to the volume of data added later by the user, then all new source data will most likely be completely compared to this unnecessary etalon data and the physical size of the optimized file can be up to 1000 times smaller than the size of the original physical file of the virtual disk. If the virtual machine player allows, then for these purposes it may be useful to use a secondary disk snapshot with some old version of the original data, which can be used as a standard for comparisons and optimization. Or in this case, it would be most efficient to save some old version of the optimized data in large archive files, while archiving should be done without compression, without applying NTFS compression to these archives and their physical layout on the disk should be defragmented.

Developer Notes.

The initial purpose of the utility is to use it as part of an enterprise automation system. According to tests, the capabilities of a modern desktop PC allow for a server system with an average performance of 20-30 optimized files per day. Depending on the specification of the automated system, implementation and deployment takes 1-3 months.

According to the GNU GPL, the utility's source code can be used in any other software products under any license. The version of the program published here has a working state and can perform the described actions with end-user data in accordance with the specifications provided here. Since hardware and software failures may occur during the operation of the utility, it is recommended to make backup copies before running the program.

Utility version 1.12 is the first release of this program for use, previous versions were not published and were used only for internal purposes. The current version of the program has the following limitations and possible directions for further development:

1. The utility can only optimize volumes with the NTFS file system;
2. When searching for matches among the target data, the utility currently selects the variants with the biggest number of matches for each file. According to tests, this gives a result close to the optimal distribution of matching blocks, but in rare cases it can give a solution corresponding to the wrong extreme with a significantly reduced overall result;
3. Current Windows API functionality does not allow for intentional file fragmentation, resulting in impossibility to do intentional fragmentation of big file in front of several small etalon files. It also unable to move the same file in front of its original location if the etalon instance is fragmented;
4. The source and target disk geometries must be the same, the utility cannot perform data matching if the source and target volumes have differences;
5. The utility cannot defragment NTFS system files and cannot change the geometry of the target disk after end of the task, it can not clean the original blocks from system data;
6. The utility optimization task obtains the result by writing the data to the reference target disk, it can have some other schemes for writing its result.

Apart from the fact that changing any of these prerequisites may be required only for some specific task, there are also many other possible related areas of application. As a result, I do not plan to make any changes to the current version of the program in the near future. At least until I receive feedback information from end users regarding the changes required for use and the conditions for supporting the project.

Utility project files.

Additional files have been added to the Cloper utility to better meet the criteria for open source software, for its documentation, and, where necessary, to make it easier to work with the source code. List of files with their relative locations:

1. ".\docs\Утилита Cloper - сжатие и оптимизация диска отката виртуальной машины.pdf" - description of the utility in Russian;
2. ".\docs\Cloper utility for compressing and optimizing virtual machine rollback disk.pdf" - description of the utility in English;
3. ".\project\..." - the original project of the utility was in Pascal, debugging and compilation of the published version of the program was carried out using Embarcadero® RAD Studio XE5;
4. ".\cloper_x86.exe" - 32-bit version of the utility;
5. ".\cloper_x64.exe" - 64-bit version of the utility.

Project status, support and development.

The current set of utility functionality and the utility source code are operational and can be used on Windows OS from XP to version 11. This set of functionality is published here for general use as is. To solve other similar problems of processing the source data of virtual machines, the set of utility capabilities can be expanded or improved. For offers of financial support and for the development of the project, I am available at the following contacts:

E-Mail: kopyurff@yahoo.com, kopyurff@rambler.ru
Mobile: 8-921-912-44-10