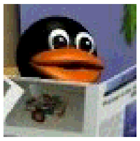
LinuxFocus article number 321
http://linuxfocus.org

storeBackup, the unconventional backup tool



by Heinz-Josef Claes
<hjclaes(at)web.de>

About the author:

The author prefers to not publish any picture of him online.

Translated to English by:
Jürgen Pohl
<sept.sapins(at)verizon.net>

Abstract:

StoreBackup offers itself to the general user who does not neccessarily own a tape backup but a second harddrive or another computer. It offers itself to the users in the professional environment for extremely fast and comfortable access to their backups, also to save on the costs of tapes as well as administrative expenses.

Storage on harddrives or similar devices offers itself as an alternative or additional resource to data backup on tapes. The program to be introduced here performs well and saves storage capacity:

- Directories, including their tree structure, may be copied to another location (e.g. /home => /var/bkup/2003.12.13_02.04.26). Permissions to the files remain, enabling users to access the backup directly.
- The content of the files is going to be compaired with the existing backup to make sure there is only one backup for each file, that means files with the same content exist physically only once in the backup.
- Identical files are hard linked and appear in the backup in the same locations as in the original.
- Backup files will be compressed, except they are marked ’exclude’. Compression may be excluded entirely.

- Backup series, generated independently (e.g. from different machines) may refer through hard links to shared files. Full or partial backups may be executed with this method, always under the condition that files with the same content may exist only once in the backup.

Why a new backup tool ?

There are possibly thousands of backup programs. So, why another one? The reason arose from my activities as a consultant. The entire week I was moving around and I had no way to secure my data during the week at home. All I had was a 250MB ZIP drive on my parallel port. The backup on the ZIP drive did not give me a lot of storage space and I had to live with a low performance (about 200KB/s). In addition to that I needed fast, simple access to my data - I did not like the usual options of full, differential and incremental backups (e.g. with tar or dump): on one hand it is ususally too cumbersome to retrieve one of the versions, on the other hand it is not possible to delete an old backup at will, this has to be planned carefully at the generation of the backup.

It was my goal to be able to backup quickly during my work and find my files quickly and without hassle.

So, at the end of 1999 the first version of storeBackup was created, it was, however, not suitable for large environments. It was not performing well enough, did not resize sufficiently and was not able do deal with nasty file names (e.g. ’\n’ in a name).

Based on that experience with the first version I wrote a new one which was published a little bit less than a year later under the GPL. In the meantime the number of users had grown - from home user applications, securing of (mail) directories at ISPs or hospitals as well as universities and for general archiving.

What would be an ideal Backup Tool?

The ideal backup tool would create every day a complete copy of the entire data system (including the applicable access rights) on another data system with minimal effort for the administrator and maximal comfort for the user. The computer and hard disk systems to make this possible should be in a distant, secure building, of course. With the help of a data system browser the user could access the secure data for searching and to copy data directly back. The backup would be usable directly and without problems . Dealing with backups would become something ’normal’ - since the route over the administration would in general be unnecessary.

The process described here has a small disadvantage: it needs a lot of harddrive space and it is quite slow because each time the total amount of data needs to be copied.

How does storeBackup work?

StoreBackup tries to accomplish the "ideal backup" and to solve the two problems: storage space and performance.

Features

The first measure to decrease the necessary harddrive storage space would be the compression of data - if that makes sense. storeBackup allows the use of any compression algorithm as an external program. The default is bzip2.

Looking at the stored data closely, it is apparend that from backup to backup relatively few files change - which is the reason for incremental backups. We also find that many files with the same content may be found in a backup because users copy files or a version administration program (like cvs) is activ. In addition, files or directory structures are re-named by users, in incremental backups they are again (unnecessarily) secured. The solution to this is to check the backup for files with the same content (possibly compressed) and to refer to those. The hard link is this reference. (Explanation: data blocks in Unix systems are administered through inodes. Many different file names in as many directories may refer to an inode. The actual file is being deleted with its last hard link (=directory name). (Hard links may point to a specific file only within one file system.)
With this trick of the hard links, which were already created in existing backup files, each file is present in each backup although it exists physically on the harddrive only once. Copying and renaming of files or directories takes only the storage space of the hard links - nearly nothing.

Most likely not only one computer needs to be secured but a number of them. They often have a high proportion of identical files, especially with directories like /etc or /usr. Obviously, there should be only one copy of identical files stored on the backup drive. To mount all directories from the backup server and to backup all computers in one sweep would be the most simple solution. This way duplicate files get detected and hard linked. However, this procedure has the disadvantage that all machines to be secured have to be available for the backup time. That procedure can in many cases not be feasible, for example, if notebooks shall be backed up using storeBackup.
Specifically with notebooks we can find a high overlap rate of files since users create local copies. In such cases or if servers are backed up independently from one another, and the available harddrive space shall be utilized optimally through hard links, storeBackup is able to hard link files in independent backups (meaning: independent from each other, possibly from different machines).

For the deletion of files storeBackup offers a set of options. It is a great advantage for deletion when each backup is a full backup, those may be deleted indiscriminately. Unlike with traditional backups, there is no need to consider if an incremential backup is depending on previous backups.
The options permit the deletion or saving of backups on specific workdays, first or last day of the week/month or year. It can be assured that a set of a minimum number of backups remains. This is especially useful if backups are not generated on a regular basis. It is possible to keep the last backupsof a laptop until the end of a four week vacation even though the period to keep it is set to three weeks. Furthermore it is possible to define the maximal number of backups. There are more options to resolve the existence of conflicts between contradictory rules (by using common sense).

Performance

The procedure described above assumes that an existing backup is being checked for identical files prior to a new backup of a file. This applies to files in the previous backup as well as to the newly created one. Of course it does not make much sense to directly compare every file to be backed up with the previous backup. So, the md5 sums of the previous backup are being compared with the md5 sum of the file to be backed up with the utilization of the hash table. The program is using dbm files for this. .
Computing the md5 sum is fast, but in case of a large amount of data is still not fast enough. For this reason storeBackup checks initially if the file was altered since the last backup (path + file name, ctime, mtime and size are the same). If that is the case, the md5 sum of the last backup is being adopted and the hard link set. If the initial check shows a difference, the md5 sum is being computed and a check takes place to see if another file with the same md5 sum exists. (The comparison with a number of backup series uses a expanded but similarily efficient process). For this approach only a few md5 sums need to be calculated for a backup.

My server (200 MHz, IDE) processes about 20 to 35 files/second, my desktop machine (800MHz,IDE) about 150 to 200 files/second. On fast computers with fast harddrives (2.4 GHz, 1.4TB software RAID) I have measured 800 files/second. These results are for writing to local drives. Writing over NFS gets is a lot slower. Crucial is the speed of the harddrive. (All tests were done under Linux).

Implementations

The storeBackup tools have been testet on Linux, FreeBSD, Solaris and AIX. They should be able to run on all Unix plattforms. Perl was used as the programming language.

Installation

The installation is simple. StoreBackup can be downloaded from http://www.sf.net/projects/storebackup as storeBackup version.tar.bz2 and unpacked to the desired location.

```
tar jxf storeBackup-version.tar.bz2
```

This creates the directory storeBackup with the documentation and the executables in the subdirectory *bin.* They can be called with the complete path. As an alternative the $PATH environment variable may be set. Operating systems which do not have the program md5sum included (e.g. FreeBSD) need to compile it. Instructions for this can be found in the attached README file.

Operation

We shall not describe all options here in detail, that can be found in the software package.

The simplest method to generate a backup is:

```
storeBackup.pl -s sourceDir -t targetDir
```

sourceDir und targetDir must be existing. StoreBackup will copy the files from sourceDir to targetDir/date_time and in this procedure compressing them with bzip2 (avoiding .gz, bz2, .png etc) as well as linking duplicate files.

In its up- to- date version (1.14.1) storeBackup.pl has 45 parameter at its disposal, to describe them here would go beyond the scope of this article. They can be accessed with

```
storeBackup.pl -h
```

In the files README and EXAMPLES we can find exhaustive explanations on the different applications. It shall be pointed out that the alternative to putting the parameters in the command line - which can become complex quickly - a configurations file may be used. It can be generated with

```
storeBackup.pl --generate --file ConfigFile
```

or shorter with

```
storeBackup.pl -g -f ConfigFile
```

. After finalising the configuration it may be read, the syntax checked and partially applied by

```
storeBackup.pl -f ConfigFile --print
```

subsequently storeBackup may be startet with

```
storeBackup.pl -f ConfigFile
```

The entire description of all options of storeBackup can be found in the files README and EXAMPLES which are part of the tar file.

To detect where which version of a file in a backup exists, storeBackup can be utilized:

```
storeBackupVersion.pl -f Filename
```

filename is the name of the file in question, it has to be written just like it is in the backup, i.e. with its compression attributes. To go to the backup directory in the correct location and executing the command is the easiest way. Exercising the option "-h" will exhibit explanations to all 11 parameter.

The recovery of single files may be done with cp, ftp, file browser or similar mechanism. For the recovery of partial directory trees or complete backups it makes sense to use the applicable tool storeBackupRecover.pl It will extract the wanted files or directories from the backup. This will restore the original, i.e. user, group and rights will be re-established. The files will also be decompressed if they were so in the original version. Original hard links will be restored too. .
Additional options in storeBackup permit statistical readouts, like the manipulation of performance parameters, the overwrite behaviour and others. A total of 10 parameters may be read out by using the option "-h".

With storeBackupDel.pl backups may be deleted independently from the program storeBackupRecover.pl. This can be useful in case of a backup over NFS. Deleting directory trees over NFS is much slower than local deletion. storeBackup may be called over the NFS without delete function, this allows a better control the backup duration. The deletion of previously generated backups on the server with storeBackupDel - which, by the way, has the same options for the deletion as storeBackup - can be decoupled from the actual backup process.

Existing backups are organized in directories. They can be displayed with storeBackupls.pl (more coherent than with ’ls’). Simpy as a list

```
hjc@schlappix:~/backup ) storeBackupls.pl /media/zip/stbu/
  1  Fri May 23 2003   2003.05.23_12.37.53   -156
  2  Fri Jun 06 2003   2003.06.06_14.31.47   -142
  3  Fri Jun 13 2003   2003.06.13_14.17.18   -135
  4  Fri Jun 20 2003   2003.06.20_14.02.35   -128
  5  Fri Jun 27 2003   2003.06.27_14.23.55   -121
  6  Mon Jun 30 2003   2003.06.30_17.34.37   -118
  7  Fri Jul 04 2003   2003.07.04_13.10.06   -114
  8  Fri Jul 11 2003   2003.07.11_13.13.14   -107
  9  Fri Jul 18 2003   2003.07.18_14.03.49   -100
 10  Fri Jul 25 2003   2003.07.25_14.19.19   -93
 11  Thu Jul 31 2003   2003.07.31_17.07.55   -87
 12  Fri Aug 01 2003   2003.08.01_12.16.58   -86
 13  Fri Aug 15 2003   2003.08.15_15.10.19   -72
 14  Sat Aug 23 2003   2003.08.23_06.25.35   -64
 15  Wed Aug 27 2003   2003.08.27_18.21.09   -60
 16  Thu Aug 28 2003   2003.08.28_14.16.39   -59
 17  Fri Aug 29 2003   2003.08.29_14.35.10   -58
 18  Mon Sep 01 2003   2003.09.01_17.19.56   -55
 19  Tue Sep 02 2003   2003.09.02_18.18.46   -54
 20  Wed Sep 03 2003   2003.09.03_16.22.41   -53
 21  Thu Sep 04 2003   2003.09.04_16.59.19   -52
 22  Fri Sep 05 2003   2003.09.05_14.35.20   -51
 23  Mon Sep 08 2003   2003.09.08_20.08.52   -48
 24  Tue Sep 09 2003   2003.09.09_18.45.48   -47
 25  Wed Sep 10 2003   2003.09.10_18.30.48   -46
 26  Thu Sep 11 2003   2003.09.11_17.26.46   -45
 27  Fri Sep 12 2003   2003.09.12_15.23.03   -44
 28  Mon Sep 15 2003   2003.09.15_18.05.19   -41
 29  Tue Sep 16 2003   2003.09.16_18.04.16   -40
 30  Wed Sep 17 2003   2003.09.17_19.03.02   -39
 31  Thu Sep 18 2003   2003.09.18_18.21.09   -38
 32  Fri Sep 19 2003   2003.09.19_14.48.05   -37  not finished
 33  Mon Sep 22 2003   2003.09.22_18.58.55   -34
 34  Tue Sep 23 2003   2003.09.23_18.48.40   -33
 35  Wed Sep 24 2003   2003.09.24_19.32.24   -32
 36  Thu Sep 25 2003   2003.09.25_18.05.38   -31
 37  Fri Sep 26 2003   2003.09.26_14.59.59   -30
 38  Mon Sep 29 2003   2003.09.29_18.42.59   -27
 39  Tue Sep 30 2003   2003.09.30_18.02.03   -26
 40  Wed Oct 01 2003   2003.10.01_17.09.43   -25
 41  Thu Oct 02 2003   2003.10.02_15.26.33   -24
 42  Mon Oct 06 2003   2003.10.06_20.08.45   -20
 43  Tue Oct 07 2003   2003.10.07_19.46.54   -19
 44  Wed Oct 08 2003   2003.10.08_16.03.23   -18
 45  Thu Oct 09 2003   2003.10.09_16.58.28   -17
 46  Fri Oct 10 2003   2003.10.10_14.21.06   -16
 47  Mon Oct 13 2003   2003.10.13_18.58.24   -13
 48  Tue Oct 14 2003   2003.10.14_16.02.44   -12
 49  Wed Oct 15 2003   2003.10.15_19.04.12   -11
 50  Thu Oct 16 2003   2003.10.16_15.47.51   -10
 51  Mon Oct 20 2003   2003.10.20_09.34.52   -6
 52  Mon Oct 20 2003   2003.10.20_12.16.40   -6
 53  Tue Oct 21 2003   2003.10.21_09.43.40   -5
 54  Tue Oct 21 2003   2003.10.21_11.22.36   -5
 55  Tue Oct 21 2003   2003.10.21_16.01.15   -5
 56  Tue Oct 21 2003   2003.10.21_18.08.07   -5
 57  Wed Oct 22 2003   2003.10.22_10.02.51   -4
 58  Wed Oct 22 2003   2003.10.22_16.09.42   -4
 59  Wed Oct 22 2003   2003.10.22_18.03.05   -4
 60  Thu Oct 23 2003   2003.10.23_08.18.15   -3
 61  Thu Oct 23 2003   2003.10.23_14.16.24   -3
 62  Thu Oct 23 2003   2003.10.23_17.00.36   -3
 63  Fri Oct 24 2003   2003.10.24_13.29.30   -2
 64  Sun Oct 26 2003   2003.10.26_09.08.55   0
```

’not finished’ means the backup was abortet).
or with information on the deletion conditions in the configuration file:

```
hjc@schlappix:~/backup ) storeBackupls.pl -f stbu.conf /media/zip/stbu/
analyse of old Backups in </media/zip/stbu/>:
 Fri 2003.05.23_12.37.53 (156): keepLastOfMonth(400d)
 Fri 2003.06.06_14.31.47 (142): keepLastOfWeek(150d)
 Fri 2003.06.13_14.17.18 (135): keepLastOfWeek(150d)
 Fri 2003.06.20_14.02.35 (128): keepLastOfWeek(150d)
 Fri 2003.06.27_14.23.55 (121): keepLastOfWeek(150d)
 Mon 2003.06.30_17.34.37 (118): keepLastOfMonth(400d)
 Fri 2003.07.04_13.10.06 (114): keepLastOfWeek(150d), keepMinNumber50
 Fri 2003.07.11_13.13.14 (107): keepLastOfWeek(150d), keepMinNumber49
 Fri 2003.07.18_14.03.49 (100): keepLastOfWeek(150d), keepMinNumber48
 Fri 2003.07.25_14.19.19 (93): keepLastOfWeek(150d), keepMinNumber47
 Thu 2003.07.31_17.07.55 (87): keepLastOfMonth(400d), keepMinNumber46
 Fri 2003.08.01_12.16.58 (86): keepLastOfWeek(150d), keepMinNumber45
 Fri 2003.08.15_15.10.19 (72): keepLastOfWeek(150d), keepMinNumber44
 Sat 2003.08.23_06.25.35 (64): keepLastOfWeek(150d), keepMinNumber43
 Wed 2003.08.27_18.21.09 (60): keepMinNumber42, keepWeekDays(60d)
 Thu 2003.08.28_14.16.39 (59): keepMinNumber41, keepWeekDays(60d)
 Fri 2003.08.29_14.35.10 (58): keepLastOfMonth(400d), keepLastOfWeek(150d),
                               keepMinNumber40, keepWeekDays(60d)
 Mon 2003.09.01_17.19.56 (55): keepMinNumber39, keepWeekDays(60d)
 Tue 2003.09.02_18.18.46 (54): keepMinNumber38, keepWeekDays(60d)
 Wed 2003.09.03_16.22.41 (53): keepMinNumber37, keepWeekDays(60d)
 Thu 2003.09.04_16.59.19 (52): keepMinNumber36, keepWeekDays(60d)
 Fri 2003.09.05_14.35.20 (51): keepLastOfWeek(150d), keepMinNumber35, keepWeekDays(6
 Mon 2003.09.08_20.08.52 (48): keepMinNumber34, keepWeekDays(60d)
 Tue 2003.09.09_18.45.48 (47): keepMinNumber33, keepWeekDays(60d)
 Wed 2003.09.10_18.30.48 (46): keepMinNumber32, keepWeekDays(60d)
 Thu 2003.09.11_17.26.46 (45): keepMinNumber31, keepWeekDays(60d)
 Fri 2003.09.12_15.23.03 (44): keepLastOfWeek(150d), keepMinNumber30, keepWeekDays(6
 Mon 2003.09.15_18.05.19 (41): keepMinNumber29, keepWeekDays(60d)
 Tue 2003.09.16_18.04.16 (40): keepMinNumber28, keepWeekDays(60d)
 Wed 2003.09.17_19.03.02 (39): keepMinNumber27, keepWeekDays(60d)
 Thu 2003.09.18_18.21.09 (38): keepMinNumber26, keepWeekDays(60d)
 Fri 2003.09.19_14.48.05 (37): keepLastOfWeek(150d), keepMinNumber25, keepWeekDays(6
 Mon 2003.09.22_18.58.55 (34): keepMinNumber24, keepWeekDays(60d)
 Tue 2003.09.23_18.48.40 (33): keepMinNumber23, keepWeekDays(60d)
 Wed 2003.09.24_19.32.24 (32): keepMinNumber22, keepWeekDays(60d)
 Thu 2003.09.25_18.05.38 (31): keepMinNumber21, keepWeekDays(60d)
 Fri 2003.09.26_14.59.59 (30): keepLastOfWeek(150d), keepMinNumber20, keepWeekDays(6
 Mon 2003.09.29_18.42.59 (27): keepMinNumber19, keepWeekDays(60d)
 Tue 2003.09.30_18.02.03 (26): keepLastOfMonth(400d), keepMinNumber18, keepWeekDays(
 Wed 2003.10.01_17.09.43 (25): keepMinNumber17, keepWeekDays(60d)
 Thu 2003.10.02_15.26.33 (24): keepLastOfWeek(150d), keepMinNumber16, keepWeekDays(6
 Mon 2003.10.06_20.08.45 (20): keepMinNumber15, keepWeekDays(60d)
 Tue 2003.10.07_19.46.54 (19): keepMinNumber14, keepWeekDays(60d)
```

```
Wed 2003.10.08_16.03.23 (18): keepMinNumber13, keepWeekDays(60d)
Thu 2003.10.09_16.58.28 (17): keepMinNumber12, keepWeekDays(60d)
Fri 2003.10.10_14.21.06 (16): keepLastOfWeek(150d), keepMinNumber11, keepWeekDays(6
Mon 2003.10.13_18.58.24 (13): keepMinNumber10, keepWeekDays(60d)
Tue 2003.10.14_16.02.44 (12): keepMinNumber9, keepWeekDays(60d)
Wed 2003.10.15_19.04.12 (11): keepMinNumber8, keepWeekDays(60d)
Thu 2003.10.16_15.47.51 (10): keepLastOfWeek(150d), keepMinNumber7, keepWeekDays(60
Mon 2003.10.20_09.34.52 (6): keepDuplicate(7d)
Mon 2003.10.20_12.16.40 (6): keepMinNumber6, keepWeekDays(60d)
Tue 2003.10.21_09.43.40 (5): keepDuplicate(7d)
Tue 2003.10.21_11.22.36 (5): keepDuplicate(7d)
Tue 2003.10.21_16.01.15 (5): keepDuplicate(7d)
Tue 2003.10.21_18.08.07 (5): keepMinNumber5, keepWeekDays(60d)
Wed 2003.10.22_10.02.51 (4): keepDuplicate(7d)
Wed 2003.10.22_16.09.42 (4): keepDuplicate(7d)
Wed 2003.10.22_18.03.05 (4): keepMinNumber4, keepWeekDays(60d)
Thu 2003.10.23_08.18.15 (3): keepDuplicate(7d)
Thu 2003.10.23_14.16.24 (3): keepDuplicate(7d)
Thu 2003.10.23_17.00.36 (3): keepMinNumber3, keepWeekDays(60d)
Fri 2003.10.24_13.29.30 (2): keepLastOfWeek(150d), keepMinNumber2, keepWeekDays(60d
Sun 2003.10.26_09.08.55 (0): keepLastOfMonth(400d), keepLastOfWeek(150d),
                             keepMinNumber1, keepWeekDays(60d)
```

In addition to the backup program described above the programs llt and multtail are present. llt will generate the display of the times for creating-, modifying- and access time of files. multitail allows tracking of a number of files like using ’tail-f" but multitail offers more options than ’tail-f’ and it is more robust.

Future Plans

For the next versions of storeBackup the following features are planned:

- The worst time consumer of a backup (except the first backup during which everything gets compressed/ copied) is the hard linking. To generate a hard link is fast, but due to their large number - compared to the other operations and the parallel operations for compression specifically - this is the main time demand.
  The next version of storeBackup will offer the option to backup the directory structure and modified files in a first step. This concludes the backup from the view of the data to be secured. In a second step the missing hard links will be created. These two steps will be completely disconnected from each other - meaning they can be run on different machines and it will be feasible to do several backups prior to generating new hard links.
  Initial measurements indicate this option will result in a performance gain - compared to the "normal" full backup - by a factor of 5-10 (1/5 to 1/10 of the "normal"), if local writing is executed. Backup up over the NFS will be much faster if you start the process for hard linking locally on the remote machine.
- The plan for the next version will be the expansion of the search capabilities (with subsequent re-backup). It shall be possible to search the backups with a user-defined rule consisting of file name (pattern), file size, time of initial generation/ change, user i.d., group i.d., access rights on the file and a (simple) grep. The rules will include ’and’, ’or’, ’not’ and optional parantheses.
- Subsequent future plans envision an expansion of the options (in a tar-like fashion) and the support of new data types, e.g. devices.

Version and License

At the writing of this article the current version of storeBackup is 1.14.1. to be found at http://www.sf.net/projects/storebackup for downloading.
StoreBackup is covered by the GPL.

Webpages maintained by the LinuxFocus Editor team
© Heinz-Josef Claes
"some rights reserved" see linuxfocus.org/license/
http://www.LinuxFocus.org

Translation information:
de --> -- : Heinz-Josef Claes <hjclaes(at)web.de>
de --> en: Jürgen Pohl <sept.sapins(at)verizon.net>

2005-01-14, generated by lfparser_pdf version 2.51