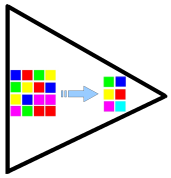
LESSFS QUICKSTART GUIDE



Version : 0.6.0
Date : 06-09-2009
Author : M. Ruijter
Email : mruijter@gmail.com

Introduction:

Lessfs is an in line data deduplicating file system. It supports block sizes ranging from 4k to 128k. Larger then 4K block sizes do require the use of a recent kernel >=2.6.26 and a recent version of libfuse >= 2.8.0-pre1

As well as data deduplication lessfs also features LZO or QUICKLZ data compression and file system encryption.

lessfs.cfg:

Lessfs needs a configuration file that defines the location of the databases.
Example:

```
BLOCKDATA_PATH=/data/dta
BLOCKDATA_BS=1048576
#
BLOCKUSAGE_PATH=/data/mta
BLOCKUSAGE_BS=1048576
#
DIRENT_PATH=/data/mta
DIRENT_BS=1048576
#
FILEBLOCK_PATH=/data/mta
FILEBLOCK_BS=1048576
#
META_PATH=/data/mta
META_BS=1048576
#
HARDLINK_PATH=/data/mta
HARDLINK_BS=1048576
#
SYMLINK_PATH=/data/mta
SYMLINK_BS=1048576
#
LISTEN_IP=127.0.0.1
LISTEN_PORT=100
MAX_THREADS=2
# Cache size in megabytes.
CACHESIZE=128
# Flush data to disk after X seconds.
COMMIT_INTERVAL=30
#
MINSPACEFREE=10
# Consider SYNC_RELAX=1 or SYNC_RELAX=2 when exporting lessfs with NFS.
SYNC_RELAX=0
ENCRYPT_DATA=on
```

```
# ENCRYPT_META on or off, default is off
# Requires ENCRYPT_DATA=on and is otherwise ignored.
ENCRYPT_META=off
```

The xxxx_path lines define the location where the databases are stored. The xxxx_BS lines are used to tune the bucket sizes of the databases. In order to handle a database containing one million of records, a bucket array with half a million of elements is needed. The size of each element is 4 bytes. That is, if 2M bytes of RAM is available, a database containing one million records can be handled.
More information about tuning tokyocabinet databases can be found on:
http://tokyocabinet.sourceforge.net/spex-en.html

LISTEN_IP and LISTEN_PORT specifies the ip address and the port number on which the lessfs tcp interface listens.

MAX_THREADS should be set to 1 or 2, depending on the amount of processors available.
More then 2 threads will degrade the performance in most cases.

CACHESIZE defines the maximum allowed number of blocks that are kept in memory.

COMMIT_INTERVAL specifies the time in seconds after which the cache is written to disk.

MINSPACEFREE specifies the percentage of free disk space that must be available before lessfs freezes all I/O. The default value is 10%. Lessfs will continue I/O when extra space becomes available.

DYNAMIC_DEFRAGMENTATION valid options are **on** or **off** and is off by default.

SYNC_RELAX=0
Valid options are : **0** (default) , 1 or 2

1. Flush all caches in lessfs for an inode and sync the tokyocabinet databases to disk when fsync is called for an inode.
2. Do not sync the tokyocabinet databases to the disk when fsync is called on an inode. The inode data will be written directly to the databases. In case of a crash the databases themselves might not be committed to disk. This feature improves some types of I/O and is especially useful with NFS. There is a trade-off between more speed and the chance of possible loss of data. ***Use with caution.***
3. Living on the edge. Do not flush the caches in lessfs and do not sync the tokyocabinet databases to disk. ***Use with extreme caution.***

ENCRYPT_DATA=off
Valid options are **on** or **off**. Lessfs requires openssl for encryption and it needs to be configured with:
./configure –with-crypto

ENCRYPT_META=on
Valid options are **on** or **off**. Default is **on**.

mklessfs:

mklessfs is needed to create a new lessfs filesystem. mklessfs requires the location of the lessfs

configuration file as argument.
Example:
mklessfs /etc/lessfs.cfg

mklessfs will prompt you for a password when ENCRYPT_DATA is selected and lessfs is configured for encryption.

Note: mklessfs will refuse to operate if blockdata.tch already exists.

Lessfs:

The lessfs program is used to mount lessfs on a mountpoint. Since lessfs supports 4..132k blocksizes.

Example 1:
mount lessfs with a 4k blocksize (This works with any kernel and any version of libfuse).

```
./lessfs /etc/lessfs.cfg /fuse  -o negative_timeout=0,entry_timeout=0,\
attr_timeout=0,use_ino,readdir_ino,default_permissions,allow_other,\
max_read=4096,max_write=4096
```

Example 2:
mount lessfs with a 128k blocksize (Recent kernel and libfuse only).

```
./lessfs /etc/lessfs.cfg /fuse -o negative_timeout=0,entry_timeout=0,\
attr_timeout=0,use_ino,readdir_ino,default_permissions,allow_other,\
big_writes,max_read=131072,max_write=131072
```

Other lessfs features:

Lessfs has a built-in freeze and defragmentation interface:

```
# telnet localhost 100
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
>help
+OK valid commands: defrag defrost freeze help quit|exit
>
```

Warning:
The defrag operation will make full copies of the databases (one by one) before deleting them. To finish this operation successful it is important that there is enough storage available.

Dynamic defragmentation greatly reduces the need to use the defragmentation option. It is however still available and can still be useful.

Lessfs will require a password to be entered when it is formatted with ENCRYPT_DATA selected.

LICENSE:

Lessfs is licensed under the GPLv3 license.
You can redistribute lessfs and/or modify it under the terms of either:

1. the GNU General Public License as published by the Free Software Foundation
2. Obtain a commercial license by contacting the Author at: **mruijter@gmail.com**

You should have received a copy of the GNU General Public License along with the source for lessfs see the file COPYING. You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/

lessfs is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the GNU General Public License for more details.