

Current implementation in user space:

ELASTIC QUOTA FILE SYSTEM

Amit Gud

Aim: *The aim of the project is to maximize the use of server hard disk space usage to the users over a network.*

Overview: A typical network scenario, for example in corporate or college environment, is that a fixed hard quota is allocated for each user, which the user is not allowed to extend in any case. This fixed quota is sometimes quite insufficient for some of the users. On the other hand there is a class of users who don't even use 5 – 10 % of their quota. So the idea of the project is to make the quota little flexible and let the users use the disk space 'as much as they like' but with an agreement with the OS.

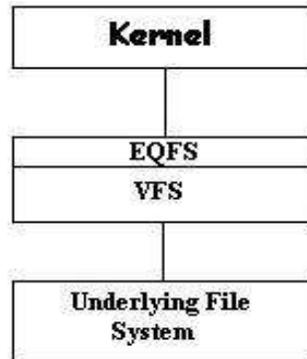
In this project the hard quotas for each user is fixed, say 10 Mb, by the normal quota management tool *edquota*. Now if an user wants to save some 5 Mb file in his quota which is already having 7 Mb filled, the user sites some files, say of 3 Mb, in his already saved 7 Mb and makes them '*elastic*'. Now when he sets 3 Mb of files as elastic, his quota is extended to occupy 6 Mb more of non-elastic files. When an user makes a file elastic, he runs through an agreement with the OS that whenever OS is in need of hard disk space the files which are marked elastic will either be deleted, compressed or moved somewhere else (in a different partition or over a network) depending upon the policies set by the user during making the files elastic. So user marks file elastic saying that, "this file is trivial but I don't want it to be deleted now." There is a certain kind of risk involved by making a file elastic, so the user marks elastic only if it he thinks it is trivial. But the advantage is he definitely gets his quota 'extended.'

Such system instantly optimizes the disk space usage to the users who needs the space but cannot use it, even if it is available, just because it is allocated to some other user which is least interested in using it. EQFS works around with two thresholds, 'User Threshold' and 'Disk threshold'. User threshold defines the total amount of space allocated to the users. Disk threshold defines the amount of space that is to be kept free at any given time. Elastic files are operated on (deleted, compressed or moved) in two cases: when total user usage increases the user threshold or when total available space on disk is less than disk threshold. Both values can be set independently at anytime and both are optional.

Design: EQFS is called a file system because it intercepts the Virtual file system (VFS) calls. This interception makes EQFS independent of the underlying file system (ext2, ext3 ...).

EQFS runs as a thin layer between kernel and VFS layer. A kernel module intercepts the VFS calls. This module registers a character device at some known unused major number

and passes or logs the necessary system call information like unlink, rmdir, chown etc. A daemon running in user space reads the device continuously (or on an interval) and acts upon the calls accordingly. For example if a file is renamed, the daemon checks if that file is elastic in the GDBM database; if found, the file is removed from the database. Also for example, if the owner of the file is changed, the database is updated accordingly.



Elastic Quota File System Architecture

EQFS maintains a user space GDBM (GNU Data Base Manager) database of all the elastic files. Making a file elastic includes adding it to this database and increasing the user's quota by the amount equal to the size of the file. This method avoids altering the inode of every file to accommodate the extra attribute – elastic, which would require altering every underlying file system used (ext2, ext3 ...) and thus recompiling the kernel during EQFS installation, which is certainly undesirable. The fields in the table are – file path, operation (delete, move, compress), parameter (if move – where to move, if compress – name of the file to be saved), owner and file size. The daemon also periodically checks that whether free disk space is equal to or more than disk threshold and total user usage is less than user threshold.

EQFS implements following user commands for adding and removing elastic attribute to the files:

mel – make file elastic

Syntax: mel <filename> <option> [parameters]

option: d – delete,

 c – compress to the filename given in the parameter,

 m – move to the location given in the parameter.

rel – remove elastic attribute

Syntax: rel <filename>

Both above user programs checks for appropriate permissions before adding or removing the elastic attribute.

sel – show elastic files (similar to ls program but showing the elastic bit also)

Syntax: sel

Advantages: Advantage of this system is that for example in a network of 10 general users, 100 Mb is to be allocated, i.e. the user threshold. Here allocation of 10 Mb each would be natural. Now here if 4 users use or need collectively just 20 Mb, the rest 6 users can use the rest 80 Mb. But the 4 users can always reclaim their full quota any time if they want to do so. Thus achieving controlled flexibility!

Using EQFS does not hamper the current quota allocation plans but adds to it more power, flexibility and optimal use of the resources.

The social aspect that is addressed by the project is meeting the varying demands of users. Every user has its own requirements (disk space is of concern here), but these demand could increase for a temporary period of time. EQFS is ideally suited to handle this. Allocating same quota for different users may not suit their needs and allocating different quota for each user is not so practical. Solution is - allocating same quota for all users and setting up a user threshold, i.e. total disk space that the users are allowed to use.

EQFS works with any underlying file system and starts running within seconds.