

# verot.net - Speed up Firefox with tmpfs

[firefox tmpfs.htm](#)

Firefox uses SQLite to store most of its information, which makes it quite slow. As SQLite accesses are IO-bound, Firefox suffers when the disk is heavily used by other processes. Here is a solution to move the profile to a RAM partition, using *tmpfs*. Note that this solution has been exposed [here](#). I am just presenting a modified solution, which uses *rsync* to sync the profile on RAM and on disk.

## Prepare your profile

First, we need to prepare the profile, in order to make it lighter. The original author suggests these changes of configuration (in *about:config* in Firefox):

```
set browser.cache.disk.capacity to 20000
set browser.safebrowsing.enabled to false
set browser.safebrowsing.malware.enabled to false
```

Then copy your profile in another directory. Your profile is in *~/.mozilla/firefox/* and looks like *xxxxxxxx.default*. Create a new directory called *profile* and copy the contents of your profile into it. Before doing that, delete the files *urlclassifier\*.sqlite* in your profile, and empty your cache.

```
cd ~/.mozilla/firefox/
mkdir profile
cp -r *.default/* profile/
```

Note that this tutorial assumes that you have only one Firefox profile. It can easily be adapted to several profiles. Here, we use the profile *xxxxxxxx.default*, but replace it with your own profile. Same for the user, which here is *xxxx*.

## Create RAM partition

Add this entry in */etc/fstab*:

```
firefox /home/xxxx/.mozilla/firefox/xxxxxxxx.default tmpfs
size=128M,noauto,user,exec,uid=1000,gid=1000 0 0
```

Of course, adapt the line above with your user name, your profile directory and the *uid* and *gid*.

## Test the profile in RAM

You will need here to close Firefox, so note the following steps. Close Firefox. Then make sure that the profile is copied in *~/.mozilla/firefox/profile/*. Empty your original profile directory, so that it is just an empty directory:

```
rm -Rf *.default/*
```

Before we start Firefox, we need to mount the RAM partition, and copy over the contents of the profile into it. In the same vein, we will need to regularly copy back the profile in RAM into the profile directory, otherwise all changes to your profile will be lost when you turn off your computer.

We use *rsync* for that (which is a better solution than using *tar* IMHO). We create a script that checks whether the profile is in RAM or not (checking for the presence of a *.unpacked* file). If it is not, we mount the RAM partition, and copy the profile over. If the profile is already in RAM, we synchronise the profile directory with the profile in RAM.

Here is the script (I call it *tmpfs\_firefox.sh*):

```
#!/bin/bash
# Change this to match your correct profile
PROFILE="xxxxxxxx.default"
cd "${HOME}/.mozilla/firefox"
if test -z "$(mount | grep -F "${HOME}/.mozilla/firefox/${PROFILE}" )"
then
    mount "${HOME}/.mozilla/firefox/${PROFILE}"
fi
if test -f "${PROFILE}/.unpacked"
then
    rsync -av --delete --exclude .unpacked ./"${PROFILE}"/ ./profile/
else
    rsync -av ./profile/ ./"${PROFILE}"/
    touch "${PROFILE}/.unpacked"
fi
exit
```

You have closed Firefox. Run this script a first time, and it should mount the partition, and copy the profile over. Then, if you list the profile directory, you should see all the profile files:

```
~/tmpfs_firefox.sh
ls ~/.mozilla/firefox/*.default/
```

Run the script a second time. It should then synchronize the saved profile with the RAM profile.

```
~/tmpfs_firefox.sh
# you should see something like:
#   building file list ... done
#   sent 36643 bytes  received 20 bytes  73326.00 bytes sec
#   total size is 45390178  speedup is 1238.04
```

## Test Firefox

First, make sure that the profile has been mounted correctly in the RAM partition. You can try to unmount the RAM partition, and run the above script again. If your profile is there, simply run Firefox. Hopefully, you will notice that it is real faster. You can see that easily with the so-called *smart bar* auto completion: results should be instantaneous. We now need to regularly synchronize the backup profile with profile in RAM. We could do that when you log out, but it is not reliable enough. Since we use *rsync* to synchronize the profiles, we can run it often. Set up a cron job which calls the above script each 5 or 10 minutes. So if your computer

crashes, you will always have your profile saved just a few minutes ago.

```
*/5 * * * * $HOME/.tmpfs_firefox.sh
```

We should also create a little script to start Firefox, so that we can make sure that the profile is loaded into RAM before Firefox loads. You can use this little script, and use it instead the normal Firefox shortcut:

```
#!/bin/bash
~/tmpfs_firefox.sh
firefox &
exit
```

## Conclusion

So what we do is basically backup the profile, and use a script that mounts the normal Firefox profile directory in RAM, and synchronize back and forth the RAM profile with the backup profile. Each 5 minutes, the RAM profile is saved in the backup profile directory. Firefox should just work as before, except it will be noticeably faster as most of the disk access that it does will be done in RAM. And this only cost 128MB of your RAM!