Tech Talk – Backing up your data + syncing machines

21 Feb 2018

Regular Backups and Syncing data efficiently

- Do you make regular backups of your work?
- How do you do it and where do you keep the data?
- Do you work on several machines on the same data/project?

rsync – copy and synchronize files

Wikipedia:
rsync is a utility for efficiently transferring and synchronizing files across computer systems, by checking the timestamp and size of files […] and functions as both a file synchronization and file transfer program. [...] Zlib may be used for additional compression, and SSH or stunnel can be used for data security.

[...] For example, the command

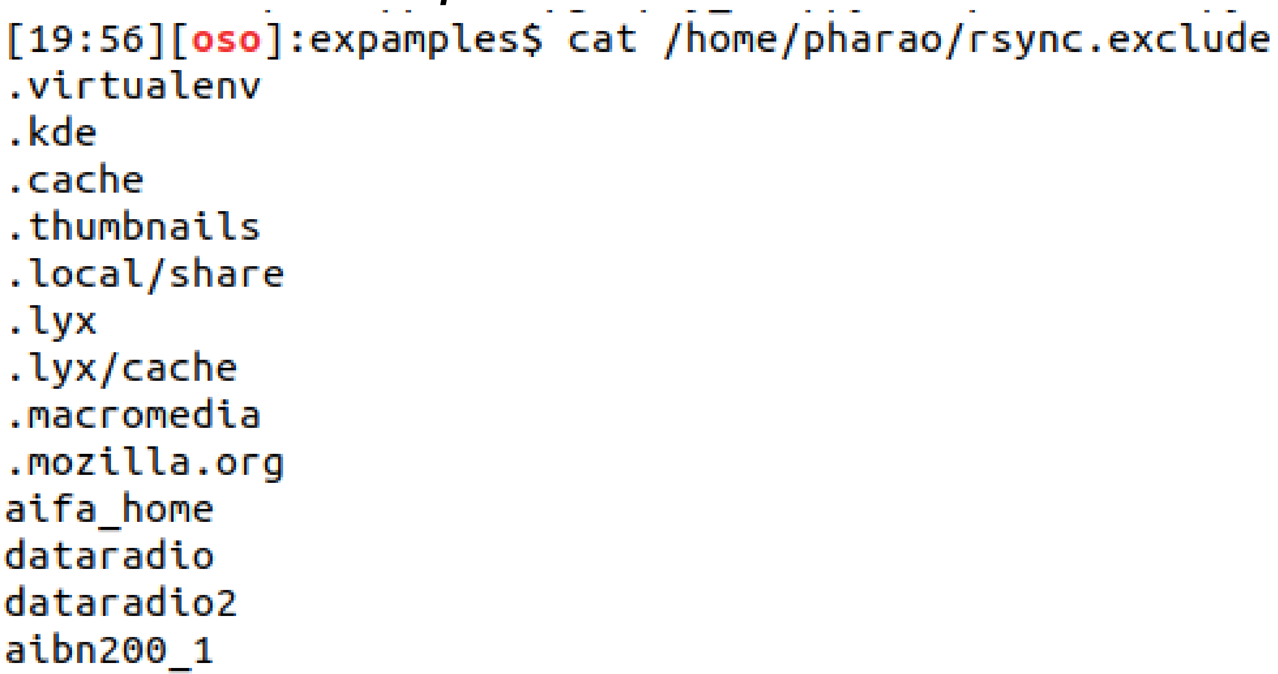
```
rsync local-file user@remote-host:remote-file
```

[...] will use SSH to connect as user to remote-host. Once connected, it will invoke the remote host's rsync and then the two programs will determine what parts of the file need to be transferred over the connection.

rsync – copy and synchronize files

- rsync only copies/updates files that changed since the last time you ran it

- You can exclude directories/files form being considered for the syncing with the option
  - --exclude-from */path/to/file*



```
[19:56][oso]:expamples$ cat /home/pharao/rsync.exclude
.virtualenv
.kde
.cache
.thumbnails
.local/share
.lyx
.lyx/cache
.macromedia
.mozilla.org
aifa_home
dataradio
dataradio2
aibn200_1
```

rsync – copy and synchronize files

- To delete things on the receiving side that were deleted on the sending side pass the option --delete

- More useful flags:
  - -P show a progress summary
  - -r   sync recursively (i.e. go through all subdirectories)
  - -t   preserve modfication times
  - -u update only (i.e. do not overwrite files that are newer on receiving side)
  - -a archive mode, shortcut for -rlptgoD
  - -z compress files before transfer

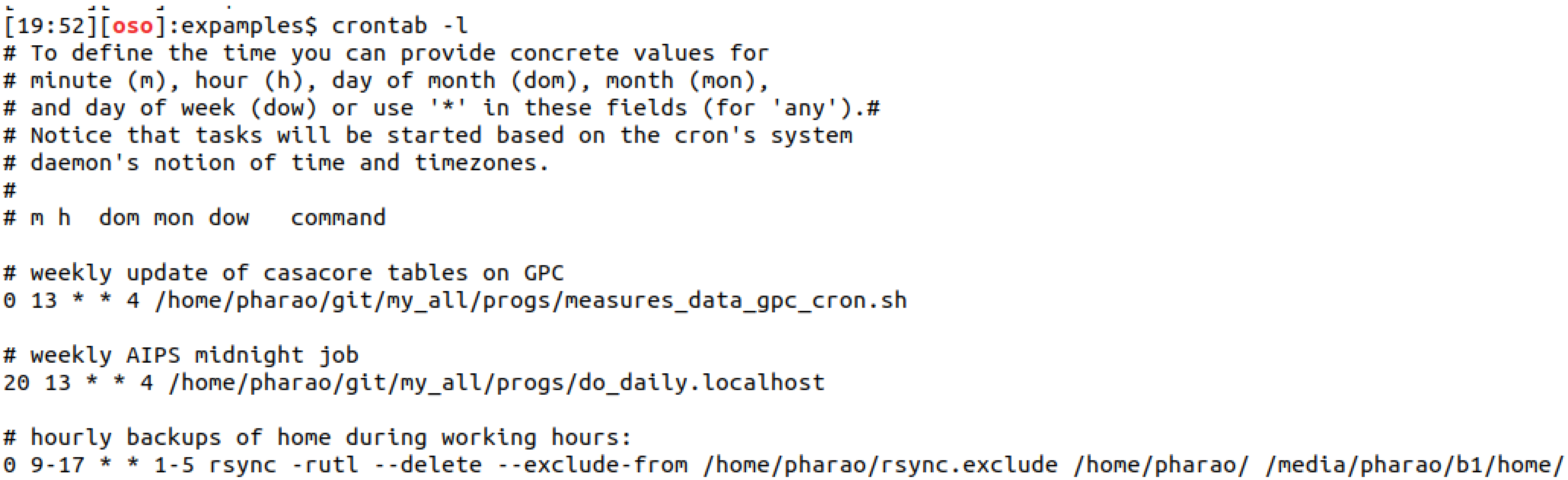
- Can also be used instead of scp, same syntax,
  - i.e. you can synchronize files across machines

- Some examples...

cronjobs – perform task automatically on a regular basis

Cron - Wikipedia:
The software utility cron is a time-based job scheduler in Unix-like computer operating systems. People who set up and maintain software environments use cron to schedule jobs (commands or shell scripts) to run periodically at fixed times, dates, or intervals. It typically automates system maintenance or administration—though its general-purpose nature makes it useful for things like downloading files from the Internet and downloading email at regular intervals.

crontab – the table with jobs



```
[19:52][oso]:expamples$ crontab -l
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# m h  dom mon dow   command

# weekly update of casacore tables on GPC
0 13 * * 4 /home/pharao/git/my_all/progs/measures_data_gpc_cron.sh

# weekly AIPS midnight job
20 13 * * 4 /home/pharao/git/my_all/progs/do_daily.localhost

# hourly backups of home during working hours:
0 9-17 * * 1-5 rsync -rutl --delete --exclude-from /home/pharao/rsync.exclude /home/pharao/ /media/pharao/b1/home/
```

Modify your crontab with

crontab -e