JB on programming Documentation

Release 1.0.0

Jacek Bzdak <jacekfoo@gmail.com>

Jan 29, 2018

Contents

Contents:

CHAPTER 1

How to use pythonbrew and virtualenv

Warning: This is outdated and not longer revelant.

In most of my projects I use virtualenvs and install dependencies by pip, well it's nothing special, since virtualenvs have become mostly standard in python developement.

But since among my colleagues I'm sort of early-adopter, and I find myself explaining this again and again, I figured that I'll write this article.

1.1 What is a python virtual environment

Virtual environment is *a way to manage python interpreter and associated installed packages*. Generaly you use separate virtualenvs for all your projects, in which case all these projects can depend on different versions of libraries.

1.2 Advantages of using virtualenv

- Projects using different virtualenvs can use different versions of libraries.
- To install a library inside virtualenv you don't need to have root privelages, which is good even if you are root, because packages installed via *pip* shouldn't be trusted.

1.3 Installing virtualenv

Normally you install virtualenv using your system package manager.

1.4 Using virtualenv

```
virtualenv foo # create virtualenv in foo subfolder of local folder
New python executable in foo/bin/python
Installing distribute (..) done.
Installing pip...............done.

source foo/bin/activate # Activate the virtualenv

pip install django #install packages
Downloading/unpacking django
  Running setup.py egg_info for package django

    warning: no previously-included files matching '__pycache__' found under␣
↪directory '*'
    warning: no previously-included files matching '*.py[co]' found under directory '*
↪'
Installing collected packages: django
  Running setup.py install for django
    changing mode of build/scripts-2.7/django-admin.py from 644 to 755

    warning: no previously-included files matching '__pycache__' found under␣
↪directory '*'
    warning: no previously-included files matching '*.py[co]' found under directory '*
↪'
    changing mode of /tmp/foo/bin/django-admin.py to 755
Successfully installed django
Cleaning up...

python -c 'import django' #Verify django is installed

deactivate # turn off the virtualenv

python -c 'import django' # See that django was installed locally only!
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: No module named django
```

Used commands

virtualenv path creates virtualenv in specified *path*. Interesting options include:

- *virtualenv –python=python2.5* creates virtualenv using specified interpreter
- *virtuaelnv –system-site-packages* allows virtualenv to use packages installed systemwide (it is nice since some packages can't be installed easily via pip).

source path/bin/activate Enables virtualenv in current console

pip install packagename installs package inside virtualenv (if it was activated)

deactivate disables virtualenv in current console

1.5 Pip cache trick

When using virtualenvs you'll find that installing packages via pip can be painfully slow. You can instruct pip to use cache for downloaded packages by adding following line to your *.bashrc* file.

```
export PIP_DOWNLOAD_CACHE=$HOME/.pipcache
```

1.6 Using pythonbrew

Virtual enviorment *borrows* one of your system interpreters, pythonbrew takes it step further: it downloads and compiles your own python interpreter (and has integrated support for virtualenvs.

1.7 Installing pythonbrew

Dont use instructions from pythonbrew repository, as they basically tell you: 'download file via http and execute it in bash' (which is *very insecure*).

Just clone repository stored in github at https://github.com/utahta/pythonbrew and execute *install_pythonbrew.py* script.

Then as instructed paste following into bashrc file:

```
[[ -s "$HOME/.pythonbrew/etc/bashrc" ]] && source "$HOME/.pythonbrew/etc/bashrc"
```

1.8 Using pythonbrew

pythonbrew install 2.7.1 Installs python *2.7.1* into pythonbrew

pythonbrew use 2.7.1 Use installed python *2.7.1* in current console.

pythonbrew venv create name -p 2.7.1 Create virtualenv named *name* for python interpreter version *2.7.1*

pythonbrew venv use name -p 2.7.1 Activate virtualenv named *name* for python interpreter version *2.7.1*

CHAPTER 2

Remapping keys in linux

2.1 The Problem

Keyboard on my Thinkpad W530 has one quite moronic feature: page up and page down are beside arrow keys, while Home and End are on the other side of keyboard.

I want to switch these keys.

2.2 The solution (TL;DR;)

Add following line to `/etc/rc.local`

```
setkeycodes e049 102 e051 107 e047 104 e04f 109
```

2.3 Solutions

You may do it in couple of ways:

- Do it at X level remapping using `xmodmap`
- Do it at lower level

I did it previously using `xmodmap`, this solution had many drawbacks, and since upgrade to gnome 3.8 I needed to launch `xmodmap` by hand. Anyways this is what everyone suggests, like everywhere, so just google this solution.

2.4 Changing key bindings using udev

When linux reads a keystroke, first thing that is registered is so called `scancode` than scancode is converted to `keycode`.

First thing you'll need to know is what scancodes you want to remap, then you'll need to know to what keycodes you'll remap. To do this you'll need the `showkey` program.

To know `scancode` of a page up key you'll need to:

```
jb ~ $ sudo showkey --scancodes
# Some text
# I press PgUpXF86MonBrightnessUp
^[[H0xe0 0x47 0xe0 0xc7
```

It outputed four hex numbers: `0xe0 0x47 0xe0 0xc7`, in my case first two were a scancode of the key. First one `e0` is an escape character.

Then youll need to know keycodes of your chars:

```
jb ~ $ sudo showkey --keycodes
# Some text
# I press PgUp
^[[Hkeycode 102 press
keycode 102 release
```

Now we can remap keys, to do this you'll need the `setkeycodes` command, it takes list of pairs of numbers. First item in each pair is a `scancode` sa a hexadecimal number, without the `0x` (so if scancode of `PgUp` is `0xe0 0x47` write: `e047`).

In my case it was:

```
sudo setkeycodes e049 102 e051 107 e047 104 e04f 109
```

Syntax for `setkeycodes` is quite moronic, so here is explanation, following command `setkeycodes e049 102` means: set keycode `102` (102 is in decimal) for scancode combination: `0xe0 0x49`, note that both bytes are concatenated, and are in without leading `0x`.

Changes made by setkeycodes are not permanent, so you'll need to find a way to execute this script at start of the system.

Warning: Using this method can harm your computer. If you'll mess keymap severly, and make changes peremanent you might be unable to login (when you for example remap letter characters).

For example insert this line inside `/etc/rc.local` file.

2.5 When this approach will not work, and what to do then

This will change keyboard bindings for all input devices, which might not be what you want.

You should use `udev` to do this, I have tried and failed, if you'll succeed please let me know :)

2.6 References

1. Ask Ubuntu http://askubuntu.com/questions/69804/how-do-i-change-the-keymap-of-a-single-device-logitech-presenter
2. Arch Linux webpage: https://wiki.archlinux.org/index.php/Map_scancodes_to_keycodes, https://wiki.archlinux.org/index.php/Extra_Keyboard_Keys

CHAPTER 3

How to use Libre Office serial document (mail merge) functionality

Libre Officess mailmerge functionality is a nice feature that allows you to create *any kind of serial document*, while all documentation seems to indicate that it is only usable for creating e-mails.

I used this tutorial (it has nice screenshots and so on).

So anyways steps are as follows:

3.1 Create data source

Create a data source, this can be:

1. A Libre Office Base database (or any database)
2. A csv file (**remember to have header row**).
3. An Libre Office Spreadsheet file (**also add a header row**)

3.2 Create a document template

Write the text

3.3 Attach database to the document

In writer Edit -> Exchange Database -> Browse -> Select your database file.

In writer View -> Datasources, then select your database file again.

You should see table contents.

3.4 Add fields to the document

Now you should be able to Drag and Drop fields (columns) from the table to the document.

3.5 Create serialized documents

Mail Merge Wizard or click the envelope icon in data sources. This stuff is mostly stupid and deals with preformatted address blocks, etc. I ve never used it.

1. Select starting Document: Select current document.
2. Select document type: Select letter
3. Insert address block: Uncheck all checkboxes.
4. Create salutation: Uncheck all checkboxes.

CHAPTER 4

Disabling Optimus on Debian and using only Nvidia Graphics card

I had some problems with bumblebee and optirun command — it just suddenly stopped working.

So here is how to switch your computer to use nvidia on debian, and use nvidia proprietary drivers

1. Remove `bumblebee`
2. Install Nvidia proprietary drives (https://wiki.debian.org/NvidiaGraphicsDrivers).

   Note: Remember to configure `xorg.conf` precisely as in the instructions.

3. Restart. Go to bios and switch off integrated graphics card.

If you restart your computer before step 3, most probably your X will not be usable.

CHAPTER 5

How to extract images from pdf-s

You'll need to use `convert` tool from the imagemagic suite.

```
convert -density 450 -quality 100 file.pdf foo.png
```

And you'll get image for each page.

Since by default convert created image files with low resolution, after too much googling I found that you need to fiddle with `density` and `quality` switches.

CHAPTER 6

How to extract tables from pdf-s

Extracting tables from pdf files is hard, as in pdf there are not tables, just lines and letters.

I use this tool, it is able to extract most of tabular data and recovers structure very well.

Downsides are that it is painfuly slow (launches a process to extract each cell).

CHAPTER 7

When your cuda installation stops working (on DEBIAN)

There can be many signs for this error:

`deviceQuery` returns:

```
cudaGetDeviceCount returned 38-> no CUDA-capable device is detected
```

or you get:

```
no CUDA-capable device is detected
```

In my case it was version mismatch between installed `nvidia` kernel module and `libcuda1` library.

Note: Normally this module is installed via `DKMS` so kernel module version should match version of `nvidia-kernel-dkms`, but this is not always the case...

To check version of installed kernell module:

```
sudo modinfo nvidia-current | grep version
```

For now it should be `319.xxx`. If it has version mismatch `libcuda1` you have source of your errors (yay!).

CHAPTER 8

Set dirs to 755 and files to 644

So I dont forget, if you want to share a directory structure with people you can change permissions in following way:

```
find dir -type d -exec chmod 755 {} \;
find dir -type f -exec chmod 644 {} \;
```

In this way others will be able to enter any directory and won't be able to execuye files.

CHAPTER 9

Autocropping all images in folder

Another piece of code I'm putting here for posteriority. Anyways, let's say I have couple hundred imahes in folder (autogenerated) and I need to crop them automatically (remove extra background).

Here is a quic command that does the trick:

```
mogrify -trim +repage *.png
```

This comes with imagemagic suite.

CHAPTER 10

W530 battery life

After about a year of usage maximal charge of my battery dropped about 40%, which is way too much. On Windows there are some settings that "enhance battery lifespan".

Note: It seems that to enhance lifespan of your batteries you'll need to:

- Charge it for longer periods (ideally allways to full charge)
- Don't charge it to 100%, set full charge to 90-something%

There are no generic tools to configure such behaviour — as it all sits inside BIOS. This question on ubuntu SO helped me to find solution, but nevertheless it didnt work on my W530 laptop.

On newer thinkpats you'll need to use: tcpapi-bat (you might install it from this PPA (it's for ubuntu but also works on Debian.

After instal you'll need to insert following into */etc/rclocal.d* (it will be called after startup):

```
tpacpi-bat -s ST 0 80
tpacpi-bat -s SP 0 90
```

It will do the following: battery will stop chaging after charging to `90%` (of current maximal charge) and battery will start charging only after it is discharged to `80%` (which will cause that you will charge it for 10% capacity or greater).

It would be great it I could (for example) inhibit charging for (let's say) 15 minutes after connectiong to AC, as I often connect the computer and then decide I need to go somewhere else, which is not healthy for battery.

CHAPTER 11

Reading numpy structured from a text file

Numpy has a very nice feature: a structured array, that is array in which rows have some structure, and can store different types of data in each column.

For example:

```
>>> import numpy as np
>>> arr = np.zeros(10, dtype=[['id', np.uint16], ['position', np.dtype('3float32')], [
↪'momentum', np.dype('3float32')]])
```

We have defined a structured array in each row we store: id of a particle (unsigned int), its position (three floats) and momentum (again three floats).

You can easily select from this array:

```
>>> arr['position']
>>> arr[0]['position']
>>> arr[arr['id']=1]['position']
```

This is a nice format because:

- Your data has structure. No more off-by-one errors: particle position is labeled.
- Very easy to load from binary files

Loading from text files is a entirely different matter — because writing to such arrays is kind of pain.

My requirements were:

- Array structure is the same as source file structure (order of fields is the same)
- Array structure is defined only in single place: that is dtype defintion

11.1 Solution

Solution is to:

- Read file line by line parsing contents to unstructured array.
- Create structured view
- Should be fast, that means no copying of large arrays.

Actual dtype used:

```
URQMD_DATA_DTYPE = [
    ("time", np.float32),
    ("position", np.dtype("3float32")),
    ("energy", np.float32),
    ("momentum", np.dtype("3float32")),
    ("mass", np.float32),
    ("particle_type", np.float32),
    ("additional", np.dtype("5int32")),
]
```

Helper function that takes structured dtype, and turns it to dtype that has the same number of fields but is unstructured:

```
def serialize_dtype(dt):
    dt = np.dtype(dt)
    newdt = []
    for item in dt.descr:
        if len(item) == 2:
            count = 1,
            name, type = item
        else:
            name, type, count = item
        if len(count) > 1:
            raise ValueError()
        count = count[0]
        for ii in range(count):
            newdt.append(type)
    return np.dtype(", ".join(newdt))
```

Now `frame` is a list of lines from text file.

```
parsed = np.zeros(len(frame), dtype=serialize_dtype(URQMD_DATA_DTYPE)) # Create array␣
↪without structure
for ii, line in enumerate(frame):
    data = [float(x) for x in line.split()] # Parse lines
    #-- ignoring wheher it is a float or int
    parsed[ii] = tuple(data) # Now numpy will convert single row to proper types
parsed = parsed.view(URQMD_DATA_DTYPE) # Create a structured view (no copy!)
```

Sound simple but took me some time to get it right.

CHAPTER 12

Install node modules without root

Sometimes you need to install some node modules, but every tutorial says: "Do `sudo npm install -g something`. I dont like to mix sudo and downloading some stuff from unsecure location. There is a very easy way to install such modules in a easy way.

There is how:

1. Download/untar node js. Let's say it is in `/tmp/node`
2. Create directory named "`$HOME/.local`" (or any other really)
3. `cd /tmp/node`
4. `configure --prefix="$HOME/.local`
5. `make && make install`

Now you have a working installation of node in `$HOME/.local/bin`, add this directory to `PATH` and you'll be able to install modules "globally" to your home folder, for example:

```
npm install -g coffee-script
```

and `coffee` will be accessible from `$HOME/.local/bin`.

Note: I used this for inspiration.

CHAPTER 13

Move legend outside of figure in matplotlib

Another one so I won't forget. Let's do some setup:

```
%matplotlib inline
from matplotlib import pylab
from matplotlib.font_manager import FontProperties
```

Lets assume you have a plot and you want to move legend outside of the plot window. Like this:

```
pylab.plot(range(10), label="Plot 1")
pylab.plot(range(10, 0, -1), label="Plot 2")
pylab.legend()
```

See how legend overlaps with the plot. Fortunately matplotlib allows me to move legend out of the way, kinda sorta.

```
pylab.plot(range(10), label="Plot 1")
pylab.plot(range(10, 0, -1), label="Plot 2")
pylab.legend(loc=9, bbox_to_anchor=(0.5, -0.1))
```

Little bit better! `bbox_to_anchor` kwarg sets legend to be centered on `X` axis and below that axis. For some unfanthomable reason you'll need to add `loc=9` so legend is actually centered.

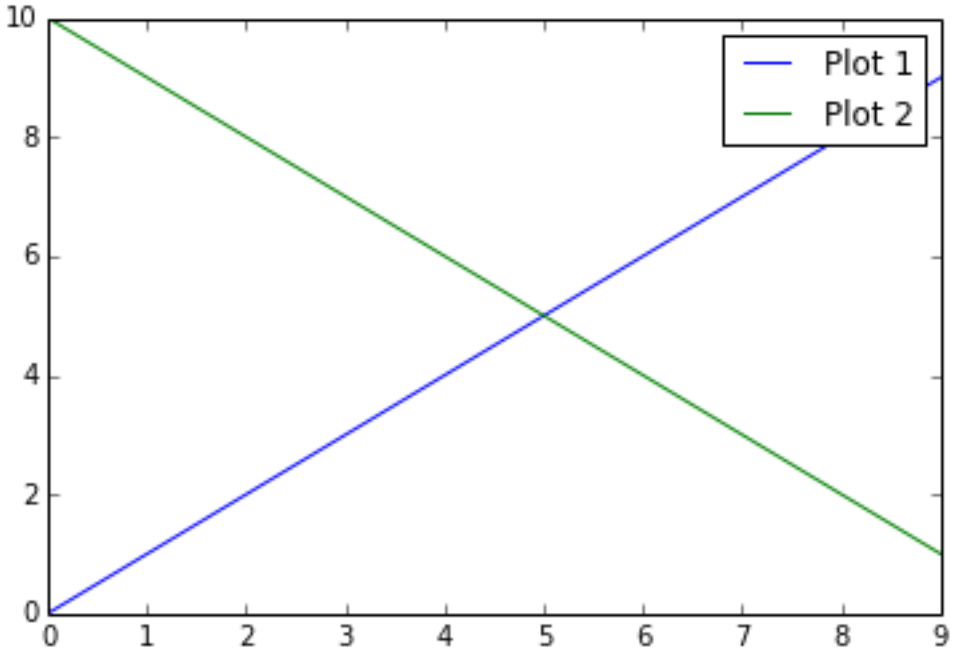
Let's tweak this a bit:

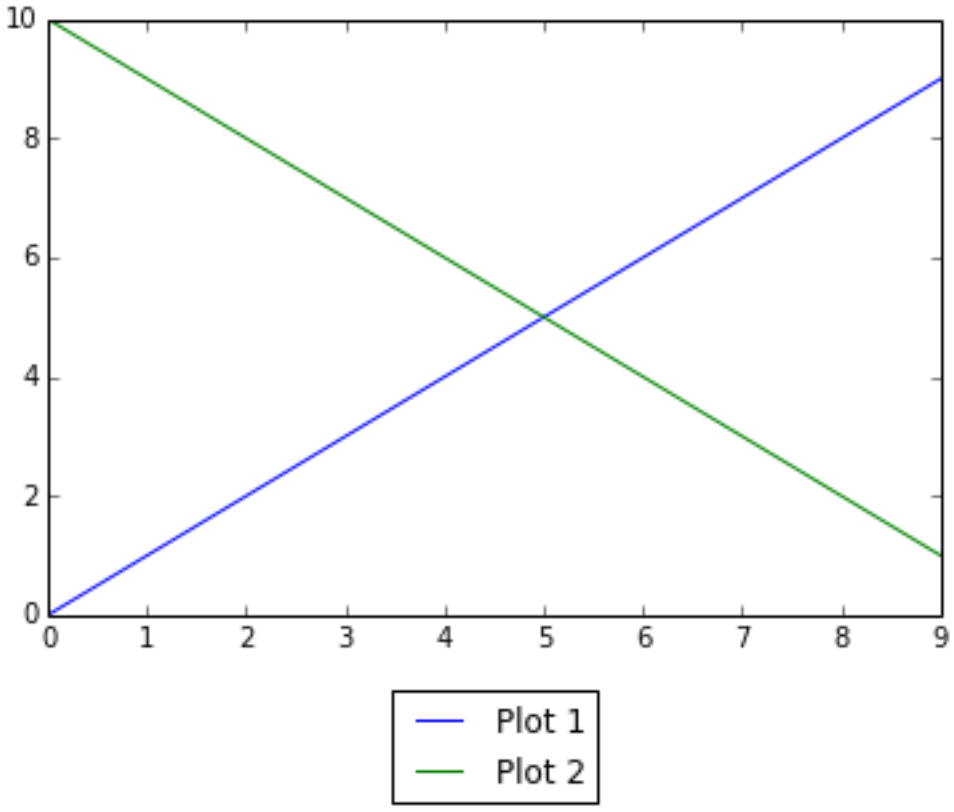
```
pylab.plot(range(10), label="Plot 1")
pylab.plot(range(10, 0, -1), label="Plot 2")
pylab.legend(loc=9, bbox_to_anchor=(0.5, -0.1), ncol=2)
```

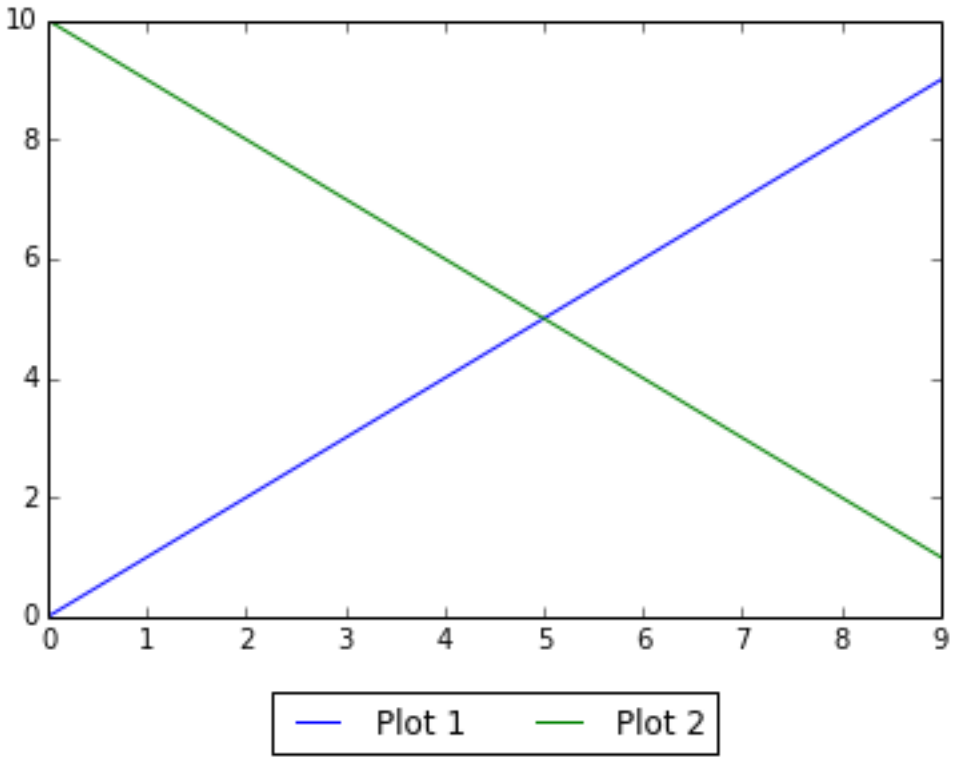
You may pass `ncol` kwarg that sets number of columns in the legend.

Now plot labels are on the same level and we have saved some space.

Note: If you have long legend legend it might be worth to decrease font size, like that:


Plot 1
Plot 2
0
2
4
6
8
10
0
1
2
3
4
5
6
7
8
9


Plot 1
Plot 2
0
2
4
6
8
10
0
1
2
3
4
5
6
7
8
9




```
fontP = FontProperties()
fontP.set_size('small')
pylab.legend(prop = fontP, ....)
```

So far so good, but if you try to save this image legend will get cut in half.

To fix this you'll need to:

1. Set `bbox_inches="tight"` keyword argument
2. Pass legend as `additional_artists` kwarg argument. This argument takes a list so you'll need to append legend somewhere.

Here is an example:

```
pylab.plot(range(10), label="Plot 1")
pylab.plot(range(10, 0, -1), label="Plot 2")
art = []
lgd = pylab.legend(loc=9, bbox_to_anchor=(0.5, -0.1), ncol=2)
art.append(lgd)
pylab.savefig(
    "/tmp/foo-fixed.png", additional_artists=art,
    bbox_inches="tight")
```

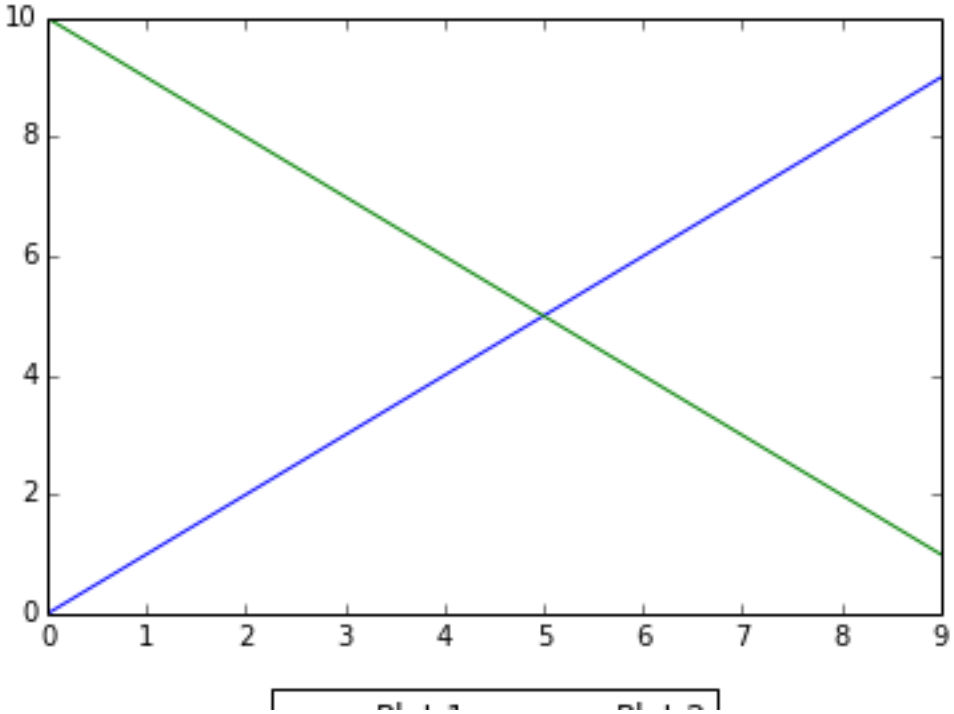


Fig. 13.1: How this figure is saved.

CHAPTER 14

How to deploy django application on debian server (UWSGI version)

14.1 Install proper version of python

On debian I'd discourage using `system` python for deployment — mostly becaluse they tend to upgrade minor python versions without notice, which breaks `C ABI` in installed virtualenvs.

So either roll your own `deb` files that install pythons somewhere in `/usr/local` or compile python on server (if you frown on having developement tools on your server roll `debs`).

Pythonz is a nice tool to compile (and manage) many versions of python.

14.2 Install proper version of virtualenv

Note: This is more-or less irrevelant as Python (from version 3.4 onwards) has it's own built-in virtualenv tool, so if you use non-system one it'll have its own virtualenv.

Debian comes with ancient python/virtualenv version, and (at least on `testing`) it often breaks this install. If you install virtualenv locally it'll be much more stable.

To install virtualenv locally just download virtualenv, unpack package and use `virtualenv.py`.

This hsa the added benefit that you can have two versions of virtualenv one for python2 and one for python3.

14.3 Install your application into virtualenv

You know how to do that don't you?

Now you can tests whether your setup is correct, just run

```
./manage.py runserver
```

and see if you can connect to your application.

14.4 Install uwsgi into your virtualenv

Install `uwsgi` **into your virtualenv** from `pip`. Now you can run your application using uwsgi:

```
uwsgi --http :8000 --module webapp.wsgi
```

I strongly discourage you from using uwsgi bundled with system.

14.5 Use supervisord to launch your applicaiton

You'll need to run your application on system start, simplest way is to use `supervisord` (you can install it via aptitude).

First create a script that will:

1. Source virtualenv
2. cd to your app directory
3. run uwsgi.

Something along the lines:

```
#!/bin/bash
export HOME=/home/user
source $HOME/venv/bin/activate
cd $HOME/webapp
uwsgi --socket :8000 --module webapp.wsgi
exit $?
```

Notice that `--http` turned into `--socket`. Now uwsgi will speak `uwsgi` protocol (which should be faster than http).

Then add configuration to supervisord. All services are defined as `*conf` files (in `ini` format) inside `/etc/supervisor/conf.d`.

Create file containing something like:

```
[program:webapp]
command=/home/webapp/webapp.sh
autostart=true
autorestart=true
stderr_logfile=/var/log/webapp.err.log
stdout_logfile=/var/log/webapp.out.log
user=webapp
```

For all configuration options see the documentation.

Now after calling: `service supervisor restart` your django application should be running.

14.6 Connect `uwsgi` and `nginx`

Note: This part is more or less ripoff from: http://uwsgi-docs.readthedocs.org/en/latest/tutorials/Django_and_nginx.html

Add following sections to nginx configuration:

```
upstream django {
    # server unix:///path/to/your/mysite/mysite.sock; # for a file socket
    server 127.0.0.1:8000; # for a web port socket (we'll use this first)
}

# configuration of the server
server {
    # the port your site will be served on
    listen      80;
    # the domain name it will serve for
    server_name .example.com; # substitute your machine's IP address or FQDN
    charset     utf-8;

    # max upload size
    client_max_body_size 75M;   # adjust to taste


    # Finally, send all non-media requests to the Django server.
    location / {
        uwsgi_pass  django;
        include     /path/to/your/mysite/uwsgi_params; # the uwsgi_params file you␣
↪installed
    }
}
```

Now you should see something on your server port `80`. To finalize our setup we need to create static and media directories.

14.7 Connect ngingx and uwsgi via linux file sockets

Because of many (peformance, safety) reasons it is better to connect nginx with uwsgi via linux domain sockets.

First replace `uwsgi` call with something like that:

```
uwsgi --module webapp.wsgi --socket $HOME/sock/webapp.sock  --chown-socket=webapp-
↪user:www-data --chmod-socket=660
```

This does the following: creates a socket in `$HOME/sock/webapp.sock`, sets its group ownership to `www-data` (which is user/group used by both `nginx` and `apache` on default debian configuration).

Note: Linux file sockets use normal file permissions, so `nginx` has to have read-write access to it.

Then replace:

```
upstream django {
    server 127.0.0.1:8001; # for a web port socket (we'll use this first)
}
```

with:

```
upstream django {
    server unix:///path/to/your/mysite/webapp.sock; # for a file socket
}
```

14.8 Update media and static root

14.8.1 Update settings py

You'll need to update `STATIC_ROOT`, `STATIC_URL`, `MEDIA_ROOT`, `MEDIA_URL` settings of youe app.

Something along the lines:

```
MEDIA_ROOT = '/var/drigan-media'
MEDIA_URL = '/media/'
STATIC_ROOT = '/var/drigan-static'
STATIC_URL = '/static/'
```

14.8.2 Update nginx

```
location /media  {
    alias /path/to/your/mysite/media;  # your Django project's media files - amend as␣
↪required
}

location /static {
    alias /path/to/your/mysite/static; # your Django project's static files - amend␣
↪as required
}
```

14.9 Tweak uwsgi so it scales

You might want tweak `uwsgi` so it launches more processes/workers, but this well outside scope of this tutorial.

CHAPTER 15

Compress a dir to tar.xz

Everyone is using tar.gz and tar.bz2 formats, and these compression algorithms (while stable and installed everywhere) are definetely not state-of-the-art.

Anyways most modern systems have much better compression ratios (or much less compression/decompression overhead). Namely `xz` (better ration) `lzo` (much better resource usage).

Anyways since I allways forget syntax to do compression here is proper command, but today I found this nice tar switch `-a` which means: "Try to guess compression format from file extension", so:

```
tar -caf foo.tar.xz data
```

will compress to `xz`, while

```
tar -caf foo.tar.lzo data
```

will compress to `lzo`. At least `tar` has sane API.

This is nice, but sometimes you'll want to tweak compression ratio for used compressor — in this case just use pipes. If you pass `-` (if anything else is non-obvious just use `man`).

```
tar -c data - | xz -9c > data2.tar.xz
```

CHAPTER 16

How to compile, install and debug Callibre in your favorite IDE

Calibre is a very nice ebook organizer, however developement documentation is, quite laconic.

Installing it for the first time turned out to be a major pain in the ass, when I installed it for the first time, as I had to do some experimentation.

I have installed callibre to a virtual enviorment, which also is a pain, nevertheless here is what to do.

1. Download callibre source

```
git clone https://github.com/kovidgoyal/calibre
```

2. Install all system dependencies (there quite many of them). For systems not based on debian install them by hand.

```
sudo aptitude build-dep calibre
```

3. Create virtual enviorment. You'll have to pass `--system-site-packages` switch that allows virtualenv to use system python packages.

```
cd calibre
virtualenv -p python2 --system-site-packages venv
source venv/bin/activate
```

4. Try to launch `setup.py`

```
python setup.py
Traceback (most recent call last):
  (...)
  File "/tmp/calibre2/setup/build_environment.py", line 103, in get_sip_dir
  (...)
EnvironmentError: Failed to find the location of the PyQt5 .sip files
```

   Callibre can't find system SIP files for PyQt5, you'll beed to edit `setup/build_environment.py` find a line that contains:

```
pyqt['pyqt_sip_dir'] = get_sip_dir(sys.prefix if iswindows else os.path.join(sys.
↪prefix, 'share', 'sip'))
```

and replace it with:

```
pyqt['pyqt_sip_dir'] = get_sip_dir(sys.prefix if iswindows else os.path.join('/
↪usr/', 'share', 'sip'))
```

Just replace `sys.prefix` to `/usr`, if you installed PyQt and other dependencies for calibre setup.py should work now

5. Launch `python setup.py bootstrap`. This command does some stuff I don't really understand, but it is neccessary to build calibre.
6. Install Calibre in developement mode: `python setup.py develop`.
7. Now to debug Calibre you'll just need to launch `venv/bin/calibre` in debug mode in your favorite IDE.

CHAPTER 17

How to change what external program calibre uses to open files

Here are two (mine) related questions:

- http://stackoverflow.com/q/32256039/7918
- http://unix.stackexchange.com/q/225103/5612

Abbreviated solution is here:

Calibre uses `QDesktopServices.openUrl(qurl)` method, that uses `xdg-utils` under the hood to open files.

IMO `xdg-utils` is a stinking mess of fragile bash scripts, that ... behaves erratically at best.

To set what program is used to launch a certain file you'll need to:

1. Find mimetype of this file, this can be done by:

   ```
   xdg-mime query filetype <filename>
   ```

   This will print out mimetype of that file, for example `application/pdf`.

2. Then you'll need to find a `desktop` file associated with application you want to use. In debian desktop files are in `/usr/share/applications`.

   In case of pdf file I wanted to open it using `evince` that has `desktop.file` named `evince.desktop`.

3. Associate desktop file with mimetype, using command: `xdg-mime default evince.desktop application/pdf`.

4. Werify it works using `xdg-open`. Now try to open this file using `xdg-open`, if it opens in a proper application it should open in proper application in calibre.

CHAPTER 18

So you want to tweak docutils?

Docutils is a very nice python package that converts text documents (namely: documents formatted in restructured text) to various usefull formats (like latex, open office and so on). Sometimes default docutils converters need a little bit tweaking do fit your needs.

I needed to tweak latex generated by docutils to my specific needs, and I discovered that I can't find any tutorial on how docutils work intenally, so here is what I found out in my brief encounter with it.

Note: This article is based on my experiences from tweaking docutils myself (I couldn't find revelant pages in the documentation), so this post is in no way authoritative.

18.1 How to tweak output of (for example) rst2latex

You need to create module that defines your `Writer` that needs to inherit from `docutils.writers.latex2e.Writer`, latex contents are actually generated by `Writer.translator_class` that is an instance of `NodeVisitor`.

Here is an example of my `Writer` that renders `code-block` directive as a `verbatim` block in LaTeX.

```
# -*- coding: utf-8 -*-

import re

from docutils import nodes

from docutils.writers.latex2e import Writer as LatexWriter, LaTeXTranslator

class TweakedLatexWriter(LatexWriter):

  def __init__(self):
    super().__init__()
    self.translator_class = TweakedTranslator
```

```
class TweakedTranslator(LaTeXTranslator):

  def visit_literal_block(self, node):
    if 'code' in node.attributes['classes']:
      # code-block
      self.requirements['upquote'] = '\\usepackage{upquote}'
      self.out.append('\n\\begin{verbatim}\n%s\n\\end{verbatim}\n' % node.astext())
      raise nodes.SkipNode
    super().visit_literal_block(node)
```

Basics are simple enough: create own writer, and this writer should use your own translator.

18.2 How does `NodeVisitor` works

`NodeVisitor` works in following way (when rendering contents, there are many other node visitors that work in other ways — I guess).

Visiting a node is implemented in `Node.walkabout` method.

1. Revelant method names are generated from node class name. If node class name is `foo` two method names are generated `visit_foo` and `depart_foo`, if such methods are not found on `NodeVisitor` default method is called.
2. First `visit` method is called.
3. Then `walkabout` method is called for every child of current node.
4. `depart` method is called

You can suppress almost every step of this algorithm by raising a proper exception from `visit` method. If you want to skip rendering of children nodes just raise `SkipNode` exception.

CHAPTER 19

Connecting remote display (projector) to `i3`

It turns out to be suprisingly easy `randr` and `i3` are integrated well enough.

I have decided that I'll add a workspace dedicated for projector.

`i3.config` snippet:

```
# switch to workspace
bindsym $mod+p workspace 11

bindsym $mod+Shift+p move container to workspace 11

workspace 11 output VGA-1
```

To enable display issue:

```
xrandr --output VGA-1 --auto --right-of LVDS-1
```

To disable:

```
xrandr --output VGA-1 --off
```

CHAPTER 20

How to create password hashed in Linux `/etc/shadow` format (crypted password)

To (easily) create passwords crypted with `crypt(3)`, just use `mkpasswd` program, on debian it is in `whois` package.

Note: This observation might be obvious, but before I learned about this program I ended up creating my own implementation of it using python `crypt` module.

Note: If you wonder why `mkpasswd` is in `whois` package, here is the answer.

CHAPTER 21

How to get a table size in `psql`

If you have full `pgadmin3` you'll get very nice table statistics, but if all you have is `psql`, matter is more complicated.

You'll need to execute this SQL:

```
SELECT
    table_name,
    pg_size_pretty(table_size) AS table_size,
    pg_size_pretty(indexes_size) AS indexes_size,
    pg_size_pretty(total_size) AS total_size
FROM (
    SELECT
        table_name,
        pg_table_size(table_name) AS table_size,
        pg_indexes_size(table_name) AS indexes_size,
        pg_total_relation_size(table_name) AS total_size
    FROM (
        SELECT ('"' || table_schema || '"."' || table_name || '"') AS table_name
        FROM information_schema.tables
    ) AS all_tables
    ORDER BY total_size DESC
) AS pretty_sizes
```

CHAPTER 22

CGI on Nginx or how to run man2html on debian

I don't enjoy reading manpages, `man` has old clunky interface, however do enjoy reading documentation in html, some good people developen `man2html` tool that is a `CGI` script that serves man pages in a browser.

This is why in 2015 year I spend part of an evening trying to configure modern web server to run CGI scripts. This might not be a suprise, that nginx doesn't serve CGI by default. It can serve `FastCGI`, which is a streamlied version of CGI, that unfortunately is incompatible with plain old CGI scripts.

However there were some other good people that wrote `fcgiwrap`, which is a a wrapper that talks FCGI protocol, and then launches plain old `cgi` scripts.

So to launch `man2html` on Debian, you'll need to:

1. Install required software: `aptitude install man2html nginx-full fcgiwrap`
2. There is a very good example fcgi config at `/usr/share/doc/fcgiwrap/examples/nginx.conf`, so just copy it to: `/etc/nginx/fcgiwrap.conf`
3. JUst include `/etc/nginx/fcgiwrap.conf` in your server config.

CHAPTER 23

Problems with nfs shares on Debian (in case of Vagrant)

If you are having problems with running Vagrant box with NFS shares, here are possible solutions.

To install nfs you should need only to: `apt-get install portmap nfs-kernel-server`.

If vagrant whines with:

```
It appears your machine doesn't support NFS, or there is not an
adapter to enable NFS on this machine for Vagrant. Please verify
that `nfsd` is installed on your machine, and try again. If you're
on Windows, NFS isn't supported. If the problem persists, please
contact Vagrant support.
```

- Check if there is `nfsd` entry in `/proc/filesystems` if there is no such entry, or there is only `nfs` or `nfs4` entry something is wrong with your host.
- Maybe `nfs` kernel module is not loaded? Try `modprobe nfs; modprobe nfsd` and see if it helps,
- Maybe you forgot to install rules for `nfs` to your firewall?

  Check if you can telnet to:

  - Port 111 on your machine from your machint
  - Port 2049 on your machine from your machint
  - Port 111 on your machine from your vagrant box
  - Port 2049 on your machine from your vagrant box
- If you disabled `ufw` it might still screw your network communication, so:

  ```
  ufw enable
  ufw default allow
  ufw disable
  ```

  In my case after this and after restarting `nfs-kernel-server` everything worked like a charm.

- On debian I had a weird error in which nfs didn't start, because there were no exports defined, which in turn caused Vagrant to bail out with `"It appears your machine doesn't support NFS... "`. This can be diagnosed if:

```
root@karmapachyenno:~# systemctl status nfs-kernel-server.service
 nfs-kernel-server.service - LSB: Kernel NFS server support
   Loaded: loaded (/etc/init.d/nfs-kernel-server; bad; vendor preset: enabled)
   Active: active (exited) since Tue 2015-12-29 17:13:42 CET; 33min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 949 ExecStart=/etc/init.d/nfs-kernel-server start (code=exited,␣
↪status=0/SUCCESS)

Dec 29 17:13:42 karmapachyenno systemd[1]: Starting LSB: Kernel NFS server␣
↪support...
Dec 29 17:13:42 karmapachyenno nfs-kernel-server[949]: Not starting NFS kernel␣
↪daemon: no exports. ... (warning).
Dec 29 17:13:42 karmapachyenno systemd[1]: Started LSB: Kernel NFS server support.
```

  This:

  - Can be fixed by adding any exports to `/etc/exports`.
  - I fixed it by `modprobe nfs; modprobe nfsd`, then running vagrant, which will add `/etc/exports`, then reloading `kernel-server` and restarting vagrant.

CHAPTER 24

New Java Features

I decided to refresh my Java knowledge (last version I used was java 1.6), and because I learn by coding (much better than I learn by reading) here are code samples I prepared.

Note: This might be updated.

24.1 Java 1.7

Note: All examples were actually made on Java `1.8` JVM.

I used this as a reference list of changes (Oracle comparison is way to comprehensive).

24.1.1 `java.nio.path` package

Java `File` class is awful, but a very nice api was introduced in this version of Java.

Package `java.nio.path` has a very nice tutorial by Oracle, so I won't describe it here.

Here is my (very simple) example, it works similarly to Disk Usage Analyzer. or `du` command on Linux, that is: it summarizes disk usage for a folder.

To do this I just needed to implement a `FileVisitor` instance, and then pass it to `Files.walkFileTree`.

Code Highlights

Most of the logic is in `Visitor` that subclasses `FileVisitor`, this class is used to traverse whole directory tree. Inside this instance we keep track of where in the directory tree we are, by using a stack.

```
Queue<Long> fileSizes = Collections.asLifoQueue(new ArrayDeque<>());
```

Each entry in the stack corresponds to a parent directory of currently processed path, and each contains a total size of that directory.

To add size of current object to size of the parent following code is used:

```
private void pushSize(long size){
   long lastSize = fileSizes.poll();
   lastSize+=size;
   fileSizes.add(lastSize);
}
```

```
@Override
public FileVisitResult preVisitDirectory(Path dir, BasicFileAttributes attrs) throws␣
↪IOException {
    fileSizes.add(0L);
    return FileVisitResult.CONTINUE;
}

@Override
public FileVisitResult postVisitDirectory(Path dir, IOException exc) throws␣
↪IOException {
    long dirSize = fileSizes.poll();
    pushSize(dirSize);
    if (maxDepthToDisplay<0 || maxDepthToDisplay >= fileSizes.size()) {
        System.out.println(level(fileSizes.size()) + dir + " " +␣
↪humanReadableByteCount(dirSize, true));
    }
    return FileVisitResult.CONTINUE;
}

@Override
public FileVisitResult visitFile(Path file, BasicFileAttributes attrs) throws␣
↪IOException {
    if (Files.isRegularFile(file)) {
        pushSize(Files.size(file));
    }
    return FileVisitResult.CONTINUE;
}

@Override
public FileVisitResult visitFileFailed(Path file, IOException exc) throws IOException
↪{
    return FileVisitResult.CONTINUE;
}
```

Complete example

```
import java.io.IOException;
import java.nio.file.*;
import java.nio.file.attribute.BasicFileAttributes;
import java.util.ArrayDeque;
import java.util.Collections;
import java.util.Queue;
```

```
public class PathExamples {

    private static class Visitor extends SimpleFileVisitor<Path>{

        long maxDepthToDisplay;

        Queue<Long> fileSizes = Collections.asLifoQueue(new ArrayDeque<>());

        protected Visitor(long maxDepthToDisplay) {
            super();
            this.maxDepthToDisplay = maxDepthToDisplay;
            fileSizes.add(0L);
        }

        private void pushSize(long size){
            long lastSize = fileSizes.poll();
            lastSize+=size;
            fileSizes.add(lastSize);
        }

        private String level(int depth){
            StringBuilder sbr = new StringBuilder();
            for (int ii = 0; ii < depth; ii++) {
                sbr.append(" ");
            }
            return sbr.toString();
        }

        /**
         * http://stackoverflow.com/a/3758880
         */
        public static String humanReadableByteCount(long bytes, boolean si) {
            int unit = si ? 1000 : 1024;
            if (bytes < unit) return bytes + " B";
            int exp = (int) (Math.log(bytes) / Math.log(unit));
            String pre = (si ? "kMGTPE" : "KMGTPE").charAt(exp-1) + (si ? "" : "i");
            return String.format("%.1f %sB", bytes / Math.pow(unit, exp), pre);
        }

        @Override
        public FileVisitResult preVisitDirectory(Path dir, BasicFileAttributes attrs)␣
↪throws IOException {
            fileSizes.add(0L);
            return FileVisitResult.CONTINUE;
        }

        @Override
        public FileVisitResult visitFileFailed(Path file, IOException exc) throws␣
↪IOException {
            return FileVisitResult.CONTINUE;
        }

        @Override
        public FileVisitResult postVisitDirectory(Path dir, IOException exc) throws␣
↪IOException {
            long dirSize = fileSizes.poll();
            pushSize(dirSize);
            if (maxDepthToDisplay<0 || maxDepthToDisplay >= fileSizes.size()) {
```

```
                System.out.println(level(fileSizes.size()) + dir + " " +␣
↪humanReadableByteCount(dirSize, true));
            }
            return FileVisitResult.CONTINUE;
        }

        @Override
        public FileVisitResult visitFile(Path file, BasicFileAttributes attrs) throws␣
↪IOException {
            if (Files.isRegularFile(file)) {
                pushSize(Files.size(file));
            }
            return FileVisitResult.CONTINUE;
        }
    }

    public static void main(String[] args) throws IOException {
        Path path = Paths.get(args[0]);
        Files.walkFileTree(path, new Visitor(3));

    }
}
```

24.1.2 Fork Join Framework

Java 1.7 has very nice Fork-Join Framework, that allows one to dynamically split between cores, but here is the catch: we don't know the amount of work needed upfront.

I have decided to try this framework, to (once again) summarize size of a directory tree.

This framework is nicely explained in the tutorials.

Overall I'm surprised with the performance of both naive and parralel implementation, naive version takes 6 seconds (when ran on my 150GB home directory), while parralel takes 3sec.

Code Highlights

Task result is a POJO object, containing path, it's size, information whether this path is a directory, and sub directories (if any). Here is the definition:

```
private static class WalkFileResult{
      public final Path dirPath;
      public final boolean isDir;
      public final long dirSize;
      public final List<WalkFileResult> subdirs;

      public WalkFileResult(Path dirPath, long dirSize) {
          this(dirPath, dirSize, Collections.emptyList());
      }

      public WalkFileResult(Path dirPath, long dirSize, List<WalkFileResult> subdirs)
↪{
          super();
          this.dirPath=dirPath;
          this.dirSize=dirSize;
          this.isDir=Files.isDirectory(dirPath);
```

```
            this.subdirs=subdirs;
        }
    }
```

Single task has following logic:

1. If we are looking at a file, calculate file size and return it.
2. If we are looking at a directory, create task for each child of the directory, execute these tasks in parrarel and then calculate the size.

In Java it is:

```
@Override
protected WalkFileResult compute() {
    try {
        if (Files.isRegularFile(currentPath)) {
            return new WalkFileResult(currentPath, Files.size(currentPath));
        } else if(Files.isDirectory(currentPath)) {
            List<WalkFileTask> subTasks = getSubtasks();
            return joinOnSubtasks(subTasks);
        }
    }catch (IOException | InterruptedException e){
        throw  new RuntimeException(e);
    }catch (ExecutionException e){
        throw new RuntimeException(e.getCause());
    }
    return new WalkFileResult(currentPath, 0L);
}

 private List<WalkFileTask> getSubtasks() throws IOException{
          // This visitor just returns immediate children of current path
          Visitor v = new Visitor(currentPath);
          Files.walkFileTree(currentPath, v);
          return v.subtasks;
      }

     private WalkFileResult joinOnSubtasks(List<WalkFileTask> subTasks) throws␣
↪ExecutionException, InterruptedException {
          long size = 0;
          List<WalkFileResult> subDirs = new ArrayList<>();
          for (WalkFileTask res: invokeAll(subTasks)){
              WalkFileResult wfr = res.get();
              size+=wfr.dirSize;
              if (wfr.isDir){
                  subDirs.add(wfr);
              }
          }
          return new WalkFileResult(currentPath, size, subDirs);
      }
```

Complete example

```
package examples;

import javax.sound.midi.SysexMessage;
import java.io.IOException;
```

```
import java.nio.file.*;
import java.nio.file.attribute.BasicFileAttributes;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.concurrent.ExecutionException;
import java.util.concurrent.ForkJoinPool;
import java.util.concurrent.ForkJoinTask;
import java.util.concurrent.RecursiveTask;



public class ForkJoinPath{

    private static class WalkFileResult{
        public final Path dirPath;
        public final boolean isDir;
        public final long dirSize;
        public final List<WalkFileResult> subdirs;

        public WalkFileResult(Path dirPath, long dirSize) {
            this(dirPath, dirSize, Collections.emptyList());
        }

        public WalkFileResult(Path dirPath, long dirSize, List<WalkFileResult>␣
↪subdirs) {
            super();
            this.dirPath=dirPath;
            this.dirSize=dirSize;
            this.isDir=Files.isDirectory(dirPath);
            this.subdirs=subdirs;
        }
    }

    private static class WalkFileTask extends RecursiveTask<WalkFileResult>{

        private static class Visitor extends SimpleFileVisitor<Path>{

            public final Path root;

            public List<WalkFileTask> subtasks = new ArrayList<>();

            protected Visitor(Path root) {
                super();
                this.root = root;
            }

            @Override
            public FileVisitResult preVisitDirectory(Path dir, BasicFileAttributes␣
↪attrs) throws IOException {
                if(Files.isSameFile(dir, root)){
                    return FileVisitResult.CONTINUE;
                }
                if (Files.isReadable(dir)) {
                    subtasks.add(new WalkFileTask(dir));
                }
                return FileVisitResult.SKIP_SUBTREE;
            }
```

```
            @Override
            public FileVisitResult visitFile(Path file, BasicFileAttributes attrs)␣
↪throws IOException {
                if (Files.isReadable(file) && Files.isRegularFile(file)) {
                    subtasks.add(new WalkFileTask(file));
                }
                return FileVisitResult.CONTINUE;
            }

            @Override
            public FileVisitResult visitFileFailed(Path file, IOException exc) throws␣
↪IOException {
                return FileVisitResult.CONTINUE;
            }
        }

        private final Path currentPath;

        public WalkFileTask(Path currentPath) {
            super();
            this.currentPath=currentPath;
        }

        private List<WalkFileTask> getSubtasks() throws IOException{
            // This visitor just returns immediate children of current path
            Visitor v = new Visitor(currentPath);
            Files.walkFileTree(currentPath, v);
            return v.subtasks;
        }

        private WalkFileResult joinOnSubtasks(List<WalkFileTask> subTasks) throws␣
↪ExecutionException, InterruptedException {
            long size = 0;
            List<WalkFileResult> subDirs = new ArrayList<>();
            for (WalkFileTask res: invokeAll(subTasks)){
                WalkFileResult wfr = res.get();
                size+=wfr.dirSize;
                if (wfr.isDir){
                    subDirs.add(wfr);
                }
            }
            return new WalkFileResult(currentPath, size, subDirs);
        }

        @Override
        protected WalkFileResult compute() {
            try {
                if (Files.isRegularFile(currentPath)) {
                    return new WalkFileResult(currentPath, Files.size(currentPath));
                } else if(Files.isDirectory(currentPath)) {
                    List<WalkFileTask> subTasks = getSubtasks();
                    return joinOnSubtasks(subTasks);
                }
            }catch (IOException | InterruptedException e){
                throw  new RuntimeException(e);
            }catch (ExecutionException e){
                throw new RuntimeException(e.getCause());
```

```
            }
            return new WalkFileResult(currentPath, 0L);
        }
    }

    private static String level(int depth){
        StringBuilder sbr = new StringBuilder();
        for (int ii = 0; ii < depth; ii++) {
            sbr.append(" ");
        }
        return sbr.toString();
    }

    /**
     * http://stackoverflow.com/a/3758880
     */
    public static String humanReadableByteCount(long bytes, boolean si) {
        int unit = si ? 1000 : 1024;
        if (bytes < unit) return bytes + " B";
        int exp = (int) (Math.log(bytes) / Math.log(unit));
        String pre = (si ? "kMGTPE" : "KMGTPE").charAt(exp-1) + (si ? "" : "i");
        return String.format("%.1f %sB", bytes / Math.pow(unit, exp), pre);
    }


    private static void printResult(WalkFileResult wfr, int depth){
        if (depth >= 3){
            return;
        }

        System.out.println(level(depth) + wfr.dirPath + " " +␣
↪humanReadableByteCount(wfr.dirSize, false));
        for (WalkFileResult child: wfr.subdirs){
            printResult(child, depth+1);
        }


    }

    public static void main(String[] args) throws ExecutionException,␣
↪InterruptedException {
        long start = System.nanoTime();
        Path path = Paths.get(args[0]);
        ForkJoinPool pool = new ForkJoinPool();
        WalkFileTask task = new WalkFileTask(path);
        pool.execute(task);

        printResult(task.get(), 0);
        double duration = (System.nanoTime() - start) * 1E-9;
        System.out.println(duration);



    }
}
```

24.1.3 Notable mentions

There is also very nice `WatchService`, that allows to monitor filesysem for file changes.

24.2 Java 1.8

24.2.1 Streams and Lambdas

Third attempt to do the same task: to summarize size of a directory tree.

This times using `Streams` and `Lambdas`. Solution is most concise, but least readable IMO. Also, while other solutions transparently handle unreadable directories, this one explodes with AccessDenied exception.

Code Highlights

Result POJO:

A function that can throw an exception:

```
@FunctionalInterface
public interface CheckedFunction<T, R> {
    R apply(T t) throws IOException;
}
```

A lambda that calculates file size:

```
CheckedFunction<Path, DirSize> mapper = (Path p) -> new DirSize(p,
    Files.walk(p).parallel()
    .filter(Files::isReadable)
    .mapToLong(StreamExamples::safeSize).sum());
```

A stream that walks over FS calculating size of each directory:

```
Files.walk(path, 3)
  .parallel().filter(Files::isDirectory).filter(Files::isReadable).map(
      (Path p) -> {
          try {
              return mapper.apply(p);
          } catch (IOException e) {
              return new DirSize(p, -1);
          }
      }).forEach(
        (DirSize d) ->
          System.out.println(d.path + " " + humanReadableByteCount(d.size, false)));
```

Complete example

```
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

/**
```

```
 * Created by jb on 12/3/15.
 */
public class StreamExamples {

    private static class DirSize{
        public final Path path;
        public final long size;

        public DirSize(Path path, long size) {
            this.path = path;
            this.size = size;
        }
    }

    @FunctionalInterface
    public interface CheckedFunction<T, R> {
        R apply(T t) throws IOException;
    }

    public static long safeSize(Path p){
        try {
            return Files.size(p);
        } catch (IOException e) {
            return 0;
        }

    }

    public static String humanReadableByteCount(long bytes, boolean si) {
        int unit = si ? 1000 : 1024;
        if (bytes < unit) return bytes + " B";
        int exp = (int) (Math.log(bytes) / Math.log(unit));
        String pre = (si ? "kMGTPE" : "KMGTPE").charAt(exp-1) + (si ? "" : "i");
        return String.format("%.1f %sB", bytes / Math.pow(unit, exp), pre);
    }

    public static void main(String[] args) throws IOException {
        long start = System.nanoTime();
        Path path = Paths.get(args[0]);

        CheckedFunction<Path, DirSize> mapper = (Path p) -> new DirSize(p,
                Files.walk(p, Integer.MAX_VALUE).parallel().filter(Files::isReadable).
↪mapToLong(StreamExamples::safeSize).sum());

        Files.walk(path, 3)
            .parallel().filter(Files::isDirectory).filter(Files::isReadable).map(
                (Path p) -> {
                    try {
                        return mapper.apply(p);
                    } catch (IOException e) {
                        return new DirSize(p, -1);
                    }
                }).forEach((DirSize d) -> System.out.println(d.path + " " +␣
↪humanReadableByteCount(d.size, false)));

        double duration = (System.nanoTime() - start) * 1E-9;
        System.out.println(duration);
    }
```

```
}
```

CHAPTER 25

How to disable subpixel rendering in Jetbrains products

If you want to disable subpixel rendering (another term is font anitaliasing) in any JetBrains product (like: Idea, Pycharm and so on), you need to:

1. Go to `bin` dir of install folder, find file that ends with `*vmoptions` (`idea.vmoptions`, `pycharm.vmoptions`).
2. There should be a line `-Dawt.useSystemAAFontSettings=lcd` in this file, delete this line.
3. Add line: `-Dprism.lcdtext=false`.

These two lines define properties passed to Java Virtual Machine that control antialiasing, both of them are undocumented, so they might not work in future JVM releases (I tested on `1.8.51`). These JVM settings seem to clash with either JetBrains settings (that override these) and Gnome settings, so disabling them makes sense.

CHAPTER 26

Linux dynamically assigned ports

So you want to assign your application a `high` port number, maybe you do some testing (and you'll launch many instances of servers), maybe you want to set some service on unusual port (which is ok in some cases).

What you don't want is your selected port to clash with the range of ports that are automatically assigned to outgoing connections. This range is sadly OS dependent, IANA defines it as: 49152–65535, however linux chooses different range, that can be read by: ‘‘cat /proc/sys/net/ipv4/ip_local_port_range ‘‘.

For more information see this SO post.

CHAPTER 27

Asyncio servers in Python

From what I read: "normal" architecture for a web-server is that one assings single thread to a client, this is mostly OK, but after certain number of clients your performance drops (because of thread memory overhead, context switching costs, and other things). Specific limit is hart to guess, and depends on OS, app, hardware and so on.

Asynchronous servers were supposedly a solution to this problem, but I didn't really believe this. From what I understood you could always buy more frontend boxes and scale it this way.

But today I actually written a async server, this server is very simple, has no error control (it took me less than an hour, including reading manuals), protocol is very simple:

0. Protocol is line based
1. When any client sends a line of text to the server
2. This line is sent to every another client.

It was written in Python 3.5, server uses PEP-492 and the `asyncio` library. Go ahead and read this PEP, there is even a working example (that I based upon).

I didn't do extensive tests, but it seems that this server handles 4000 connections **easily** on single core, throughput is about 40K messages per second (still on single core).

Note: I wouldn't rely on these numbers much, to do a proper tests I would have to:

1. Add error handling to the server.
2. Change the protocol to acknowledge results.
3. Write some proper tests.

Tests are based on running 4 processes each spawning 1000 sockets to the server, and then writing one message per 100ms in yet another thread. Results were observed by looking at the output :)

Server works as follows:

1. Keeps a set of all connections (sockets in essence)
2. Each relieved line is sent to all sockets

Server code is here:

```
# -*- coding: utf-8 -*-

import asyncio
import threading
from asyncio.streams import StreamReader, StreamWriter

import time

from multiprocessing import Process, Lock, Condition


class AsynChat(object):

  def __init__(self, port, client_may_end_connection=False):
    self.clients = set()
    self.port = port
    self.client_may_end_connection=client_may_end_connection
    self.__loop = None

  async def handle_connection(self, reader:StreamReader, writer:StreamWriter):
    self.clients.add(writer)
    while True:
      data = await reader.readline()
      print(data.strip())
      if self.client_may_end_connection and data.strip() == b'END':
        print("STOP")
        self.__loop.stop()
      for w in self.clients:
        if w == writer:
          continue
        w.write(data)
        await w.drain()
      if not data:
        if writer in self.clients:
          self.clients.remove(writer)
          try:
            writer.write_eof()
          except OSError:
            pass # Sometimes it explodes if socket was closed very soon, didn't␣
↪investigate
          return

  async def echo_server(self):
      await asyncio.start_server(self.handle_connection, 'localhost', self.port)

  @classmethod
  def run_in_process(cls, *args, **kwargs) -> Process:
    c = Condition(Lock())
    def build_and_run(*args, **kwargs):
      ac = cls(*args, **kwargs)
      ac.run_loop(c)

    p = Process(target=build_and_run, args=args, kwargs=kwargs)
    p.start()
    with c:
      c.wait()
```

```
    return p

  def run_loop(self, c:Condition=None):
    self.__started = True
    self.__loop = asyncio.get_event_loop()
    self.__loop.run_until_complete(self.echo_server())

    def notif():
      with c:
        c.notify()

    try:
      if c:
        self.__loop.call_soon(notif)
      self.__loop.run_forever()
    finally:
      self.__loop.close()



if __name__ == "__main__":
  a = AsynChat(1234, True)
  a.run_loop()
```

CHAPTER 28

How to discard connection in a fast way

Just a quick note, here is how to drop a TCP connection fast:

- Client sends `SYN` packet
- Server responds with `RST` packet, which promptly kills the connection without any state.

CHAPTER 29

Some remarks about databases

Note: I was refreshing my knowledge on the broad subject of databases, mostly by reading Wikipedia articles, which resulted (apart from notes attached here) in creating this wikipedia book (in case some future problems with Wikipedia `you can try version hosted here`).

29.1 Read anomalies Anomalies

What can happen if your database doesn't serialize transactions properly:

Dirty read When one transaction sees uncommitted data of another one

Short example:

We have two transactions T1 and T2.

- T1 reads value of a row R
- T2 writes new value of row R
- T1 sees a change in row value

Non-Repeatable Read When value of a row changes during a transaction.

Short example:

We have two transactions T1 and T2.

- T1 reads value of a row R
- T2 writes new value of row R
- T2 commits <- **Difference from Dirty Read**
- T1 sees a change in row calue

Phantom read When a result of query changes during the transaction.

> **Note:** Database without Non-Repeatable Reads cam still suffer from phantom reads, non-repeatable reads apply when other transaction updates or deletes records while phantom also apply to inserted rows.

Short example:

- T1 executes a query: `SELECT AVG(value) FROM account WHERE ...GROUP BY ...`
- T2 executes: `INSERT .... INTO account`
- T2 commits
- T1 executes a query: `SELECT AVG(value) FROM account WHERE ...GROUP BY ...` and gets different results.

Write Skew This is an anomaly for MVCC databases, which occours each transaction works on a snapshot of database, and don't see changes done by each other.

> **Note:** This anomaly is not defined in SQL standard, as SQL standard had locking databases in mind.

Short example:

Let's consider a banking application, with following constraint: balance of account must be non-negative, but we don't store it explicitly, it is just calculated by `SELECT SUM(t.value) FOR account .... INNER JOIN transaction as t ON ...`

- At the start of the transaction balance of account A is 100PLN
- Transaction T1 starts
- Transaction T2 starts
- Transaction T1 adds a transaction for A that withdrawns 100PLN (which is valid as constraint is held)
- T1 Commits
- Transaction T2 does the same (which still is valid as constraint is held inside a snapshot for T2 — constraint is not held "outside of" this snapshot).

> **Note:** Wikipedia says that you might resolve this issue by explicitly writing to a dummy row just to force write-write confilct.

29.2 Transaction isolation Levels

Serializable Highest transaction isolation. Transactions are executed *as if* they were executed serially.

No read anomalies can occur.

In database that uses locks it requires to put locks on:

- Every row you write to
- Every row you read
- Range lock for every query (for example lock all records for which specified condition is true).

These locks are held until the end of the transaction.

Snapshot Isolation This transaction isolation level works as follows: Each transaction sees database in a (consistent) state the database was at the beginning of the transaction (with any changes it did).

This is very different from Serializable, as it allows "Write Skew" anomalies.

Note: This isolation level is not defined in SQL standard.

You can define Serializable isolation on top of Snapshot Isolation relatively easy by:

- This can be done relatively easy by the DMBS, see this article:

  Cahill, M. J., Röhm, U., & Fekete, A. D. (2009). Serializable isolation for snapshot databases. ACM Transactions on Database Systems, 34(4), 1–42. doi:10.1145/1620585.1620587

- Writing proper code that introduces artificial write conflicts between data.

  Note: These conflicts are artificial, because they exist only to introduce write conflicts, which will abort transaction that tries to write to the database as the second one.

Repeatable Read This isolation level reading a row will always produce the same value, even of these rows were changed by other transactions. However query results are can change during the transaction (especially for aggregate queries).

In database that uses locks it requires to put locks on:

- Every row you write to
- Every row you read

These locks are held until the end of the transaction.

Read Committed In this isolation level transaction doesn't see changes made by another uncommitted transactions, however it may see changes made by commited transactions.

In database that uses locks it requires to put locks on:

- Every row you write to
- Every row you read

Write locks are held until the end of the transaction, however read locks are released after each select.

Read Uncommitted In this level you can see uncommitted data sent by saved by other transactions.

29.3 Serializability

We have a some set of concurrent transactions, these transactions are serializable, if one can produce a schedule containing these transactions executed serially in some order.

Serializability is important because:

If DBMS checks if database is in a consistent state **after each of the transactions**, and transactions are serializable — it means that the database is in a consitent state after all transactions.

CHAPTER 30

MVCC in postgresql

Note: This is based on my lecture on the databases, which (is still available in Polish)

There are two ways to implement proper transaction isolation:

- First is by using locking. In this padagrim whenever transaction reads data a lock is issued, and any write to that data will wait until reading transaction finishes (this might be simplified).
- Second is by using MVCC — that is multi version concurrency. It works as follows: each transaction sees database in a state **at the time the transaction**, so reads and writes don't need to wait for each other (there is a problem with write skew anomaly, which is solved by the postgresql 9.1 and newer.

30.1 How does MVCC work

30.1.1 Start of the transaction

When transaction starts following things happen (or may happen depending on isolation level):

- Transaction is assigned a `txid` a transaction ID, transaction id's are ordered 32 bit integers (that may wrap around at some point in time, but Postgres handles it).
- `txid``s of all committed transactions are stored (possibly in a more efficient way than storing all ``txids)`

30.1.2 Data constraints

Each row contains couple of magic columns:

`xmin` This is the `txid` of transaction that inserted this row

`xmax` This is the `txid` of transaction that deleted this column

cmin, cmax Index of statement in transaction that added/deleted that row

30.1.3 Basic operations

INSERT When a transaction inserts a row it `xmin` is set to `txid` of this transaction.

DELETE When a transaction deletes a row it just sets `xmax` to it's `txid`

UPDATE Updates are replaced with a delete and insert pair.

30.1.4 Data visibility

Row is visible for transaction `txid` if (all statements must be true):

- It's `xmin < txid` (row was inserted by a transaction before this one).
- Transaction `xmin` is commited (in case of Read Commited isolation level), or `xmin` was commited before start of current transaction (other isolation levels)
- It's `xmax` is empty or `xmax > txid` (row was deleted by a transaction that started after this one).

In case of transaction that issue multiple statements `cmin`, `cmax` are used for example to have a cursor that consistently iterates over a table, even if the same transaction alters the table.

30.2 VACUUM

Data can't be deleted from disk immediately in databases using MVCC, because ongoing transactions might not 'see' the delete, and still need to access deleted row. Data can be deleted only after all transactions with `txid` lower row's than `xmax` have either committed or have been rolled back.

Postgresql does this (more or less) automatically, but you might call `VACUUM` by hand if you need to reclaim space (this space will not necessarily be freed to the OS, rather it will be accessible for new inserts).

30.3 Sources

- BRIUCE MOMJIAN `MVCC Unmasked`: http://momjian.us/main/writings/pgsql/mvcc.pdf (I have also cached it `locally`, due to permissive CC license)
- Serializable Snapshot Isolation on Postgresql Wiki, (due to even more permissive license it is also cached `locally`.
- My old lectures: http://db.fizyka.pw.edu.pl/~bzdak/bazy_danych_ed_20/wyklad10/wyk10.html#mvcc-w-postgresq
- Wikibook I have collected (in case some future problems with Wikipedia `you can try version hosted here`).

CHAPTER 31

Should you use BRTFS for your laptop?

TL; DR; Probably not.

Recently I re-installed by Debian desktop, and enabled full disk encryption, (and LVM for that matter). I was also toying with the idea of using BTRFS, which has many features I always wanted to have, including:

- Fast fs-level compression
- Optional, out of write path, deduplication.
- Very nice features including super-easy resizing (adding more GBs to your file system takes seconds)
- Copy on write semantics
- File system snapshoting (said to be good for backups)

BTRFS is stable, yet I believe is unusable for a most of people. If you are not a linux nerd, don't try, if you are you might, but remember:

- Do backups, you should always do backups, especially if you use encrypted filesystems on ssd drives.
- Do yourself a favour and buy an USB stick, and burn there a linux live-cd.
- BTRFS **will** suprise you.

Here are two nasty suprises I had.

31.1 BTRFS COW doesnt play well with some usage patterns

If you are a linux nerd, you probably have some virtual machines, COW (copy on write) doesn't play with them. Well everywhere where you have large files that are written to often, it doesn't play well with COW.

COW can be disabled on directory level, do to this issue *chattr +C /dir* command, this will disable COW for everything under */dir*. Keep in mind that it works **only on empty files and directories**, turning off COW on a file with data, has **undefined behaviour**, and most often is bad. Turning off COW for directories with files is safe, but existing files will have COW enabled.

In my case VirtualBox failed with very non-obvious errors, before I disabled COW.

31.2 BTRFS needs garbage collection (or something similar)

Basically BTRFS kind-of lies to the OS when reporting free space. You can have full filesystem and yet BTRFS will report 100GB of free space. I don't try to understand what it is, BTRFS wiki says things like: "The primary purpose of the balance feature is to spread block groups across all devices so they match constraints defined by the respective profiles."

Usable free space on your disk can be seen using *btrfs fi show* Which will display: total system size, and how much space is currently used by btrfs.

In my case:

```
# btrfs fi show

Label: none  uuid:
    Total devices 1 FS bytes used 613.29GiB
    devid    1 size 745.06GiB used 649.06GiB path /dev/mapper/
```

I have 745GB partition, of which 649GB is used by btrfs, however, only 613 is used for files.

This can be fixed by issuing something like:

```
btrfs balance start -dusage=<magic number> / &
```

<<magic number> is a percentage value, and BTRFS will try to "rebalance" only chunks filled in less than this <<magic number>>, the bigger number you put there the longer will it take, and the more space you'll reclaim. You can start with percentage value of used space you have on your drive.

CHAPTER 32

How to deploy django application on debian server (UWSGI version)

32.1 Install proper version of python

On debian I'd discourage using `system` python for deployment — mostly becaluse they tend to upgrade minor python versions without notice, which sometimes breaks `C ABI` in installed virtualenvs.

So either roll your own `deb` files that install pythons somewhere in `/usr/local` or compile python on server (if you frown on having development tools on your server roll `debs`).

Note: For development environment Pythonz is a nice tool to compile (and manage) many versions of python.

If you want deploy your server by hand just download and compile python.

If you are into automatic deployment (and you **should** be)

32.2 Assumptions about the system

I'll assume that you will configure your system in following way:

- Django application will be using `www-client` user
- Code will be inside `/home/www-client/repo`
- There will be a django generated uwsgi file in `/home/www-client/repo/webapp.uwsgi`
- Virtualenv will be in `/home/www-client/repo/venv`.

32.3 Install your application into virtualenv

You know how to do that don't you?

Now you can tests whether your setup is correct, just run

```
./manage.py runserver
```

and see if you can connect to your application.

32.4 Install uwsgi into your virtualenv

Install `uwsgi` **into your virtualenv** from `pip`. Now you can run your application using uwsgi:

```
uwsgi --http :8000 --module webapp.wsgi
```

I strongly discourage you from using uwsgi bundled with system.

You can configure `uwsgi` using a variety of ways, most of which are better than using a command line arguments :). For example you can create an ini file named `uwsgi.ini`:

```
module=webapp.wsgi
pythonpath=/home/www-client/webapp
http=8000
```

And then start uwsgi using: `uwsgi --ini uwsgi.ini`.

32.5 Use systemd to launch your applicaiton

Now use systemd to launch the application.

Systemd is a very nice init system that is becoming a standard in most recent distributions (it's even on Debian stable).

If your distribution has no systemd you can use supervisord (which is even on debian oldstable), tutorial to deploy django with it *is here*.

So create `/home/webapp/uwsgi.ini` file with following contents:

```
module=webapp.wsgi
http=8000
pythonpath=/home/www-client/repo
```

Create a `webapp.service` file and put it in `/etc/systemd/system/`, file should have following contents:

```
[Unit]
Description=Description
After=syslog.target

[Install]
WantedBy=multi-user.target

[Service]
# What process to start
ExecStart=/home/www-client/venv/bin/uwsgi --ini /home/www-client/uwsgi.ini
# What user chown to
User=www-client
# Working directory
WorkingDirectory=/home/www-client/webapp
Restart=always
```

```
# Kill by SIGQUIT signal --- this is what asks wsgi to die nicely
KillSignal=SIGQUIT
# Notify type, in this type uwsgi will inform systemd that it is ready to handle␣
↪requests
Type=notify
StandardError=syslog
NotifyAccess=all
```

Then:

```
sudo systemctl --system enable webapp
sudo systemctl start webapp
```

Now you should have a working uwsgi configuration, which is reachable at `localhost:8000`.

32.6 Connect `uwsgi` and `nginx`

Note: This part is more or less ripoff from: http://uwsgi-docs.readthedocs.org/en/latest/tutorials/Django_and_nginx.html

In this part we will put `uwsgi` behind `nginx` server.

Add following sections to nginx configuration:

```
upstream django {
    # server unix:///path/to/your/mysite/mysite.sock; # for a file socket
    server 127.0.0.1:8000; # for a web port socket (we'll use this first)
}

# configuration of the server
server {
    # the port your site will be served on
    listen      80;
    # the domain name it will serve for
    server_name .example.com; # substitute your machine's IP address or FQDN
    charset     utf-8;

    # max upload size
    client_max_body_size 75M;   # adjust to taste


    # Finally, send all non-media requests to the Django server.
    location / {
        uwsgi_pass  django;
        include     /etc/nginx/uwsgi_params; # the uwsgi_params file you installed
    }
}
```

Now you should see something on your server port `80`. To finalize our setup we need to create static and media directories.

32.7 Connect ngingx and uwsgi via linux file sockets

Because of many (peformance, safety) reasons it is better to connect nginx with uwsgi via linux domain sockets.

First replace `uwsgi.ini` file with something like:

And also create `/home/www-client/sock/`.

This does the following: creates a socket in `$HOME/sock/webapp.sock`, sets its group ownership to `www-data` (which is user/group used by both `nginx` and `apache` on default debian configuration).

Note: Linux file sockets use normal file permissions, so `nginx` has to have read-write access to it.

Then replace:

```
upstream django {
    server 127.0.0.1:8001; # for a web port socket (we'll use this first)
}
```

with:

```
upstream django {
    server unix:///path/to/your/mysite/webapp.sock; # for a file socket
}
```

32.8 Update media and static root

Now we'll update settings so static media is served by nginx.

32.8.1 Update settings py

You'll need to update `STATIC_ROOT`, `STATIC_URL`, `MEDIA_ROOT`, `MEDIA_URL` settings of youe app.

Something along the lines:

```
MEDIA_ROOT = '/var/drigan-media'
MEDIA_URL = '/media/'
STATIC_ROOT = '/var/drigan-static'
STATIC_URL = '/static/'
```

32.8.2 Update nginx

```
location /media  {
    alias /path/to/your/mysite/media;  # your Django project's media files - amend as␣
↪required
}

location /static {
    alias /path/to/your/mysite/static; # your Django project's static files - amend␣
↪as required
}
```

32.9 Tweak uwsgi so it scales

You might want tweak `uwsgi` so it launches more processes/workers, but this well outside scope of this tutorial.

CHAPTER 33

What to do when you get [Read error 4] when running `apt`

When you run `apt` and have following errors:

```
E: Read error - read (5: Input/output error)
E: The package lists or status file could not be parsed or opened.
```

This means that you are in serious trouble, it's not hopeless but serious.

This error is caused by `apt` having some problems reading some of it's config files. Too bad `apt` can't say in which file the problem is, and what is the problem.

Obviously there is no single answer on how to fix it, here are some hints.

1. Remove package cache: `apt-get clean`. Will purge some stuff. Probably won't help but it is safe (yes rest of the commands aren't safe).
2. Remove package lists:

   ```
   sudo rm -rf /var/lib/apt/lists
   sudo mkdir /var/lib/apt/lists
   sudo mkdir /var/lib/apt/lists/partial
   ```

3. Restore old version of dpkg status files. This is **the** file where all information on packages installed is stored. Sometimes it is corrupted, however it has automatic backups. The file is `/var/lib/dpkg/status`, first thing to try would be:

   ```
   cp /var/lib/dpkg/status /var/lib/dpkg/status.broken
   cp /var/lib/dpkg/status.old /var/lib/dpkg/status
   ```

   This will restore previous version of this file, there are older ones in `/var/backup/dpkg.status*`, but probably try the next step first.

4. `apt` (or `aptitue`) also store it's copy of dpkg state, with additional information (like which package was installed by user, and which is a as a dependency). This can also be corrupted, so:

   ```
   cp /var/lib/apt/extended_states /var/lib/apt/extended_states.broken
   cp /var/backups/apt.extended_states.0 /var/lib/apt/extended_states
   ```

5. One of the reasons for corrupted files can be file-system or hard drive problems. If you reached this point please consider backing up your data, running fsck, and diagnosing you drive.
6. Some people reported success with manually editing status files.

There is also very extensive list of commands that might help you with diagnosing problems with apt: https://help.ubuntu.com/community/PackageManagerTroubleshootingProcedure I wouldn't recommend running them without understanding though.

CHAPTER 34

How to encrypt a usb drive using cryptsetup

Actuall I didn't manage to set up it using console `cryptsetup` commands, however there is a very nice dommand `gnome-disks` (which comes in package `gnome-disk-utility`) that does everything automatically!

CHAPTER 35

Cheap and fast way to create your own Maven repository on S3

Maven is de-facto standard build system for Java apps, it's a nice and mature piece of software.

To meaningfully use it you need to host your own maven repository — that allows you and your teammates to download and release software artifacts you use. Most of these repositories is written in Java, and is not very light — most probably you'd need to have a dedicated VPS, which is OK — unless you use this repository for hobby project that is in maintenenance mode and you touch it once a year.

Cheap and easy way to host Maven repositories is to use S3, there you pay only for storage, and transfer, which is preatty cheap.

Simple guide is here: https://github.com/spring-projects/aws-maven.

CHAPTER 36

How to debug zsh startup time

My ZSH shell started to be slow, by slow I mean startup time was about ~5sec.

36.1 Step 1: measure startup time:

```
time  zsh -i -c exit
```

36.2 Step 2: Inject profiling code

At the beginning of your `.zshrc` add following: `zmodload zsh/zprof` and at the end add: `zprof`, this will load zprof mod and display what your shell was doing durung your initialization.

After start of the shell you'll see something along the lines:

```
num  calls                time                        self            name
-----------------------------------------------------------------------------------
 1)    1          49.02    49.02   26.87%     49.02    49.02   26.87%  a command
 2) ....
```

Commands ZSH spent most of the time are at the top of the output.

In my case culprit was `compinit`.

36.3 Step 3: Make compinit faster

Compinit is a function that initializes shell completion. After some googling I found out that the problem might be "too bit" `.zcompdump` file, in my case it was about `1mb`, so I deleted it (this file is autogenerated), and ZHS magically started to be faster.

After that I have added deleting this file to my user startup script.

36.4 Step 4: Cleanups

Comment out or delete additions you made to `.zshrc` file.

CHAPTER 37

Zalałem laptopa co robić jak żyć

(Note: this is guest entry in Polish)

Rozlewam wodę (herbatę) na laptopa średnio raz na pół roku, dodatkowo komputer jest dla mnie głównym źródłem utrzymania więc utrzymanie go na chodzie jest dość istotne.

Po pierwsze warto mieć komputer, który da się łatwo po zalaniu naprawić, czyli coś co:

1. Da się rozłożyć;
2. Gwarancja zawiera ubezpieczenie od przypadkowych uszkodzeń, oraz masz klauzulę naprawy on-site (komputer jest naprawiany u Ciebie w domu, a nie musisz go wysyłać do serwisu);
3. Możesz próbować kupić komputer “spill proof”, co nie wiele gwarantuje.

ThinkPady Lenowo są marką która spełnia te wymagania.

Instrukcja postępowania po zalaniu laptopa:

W okresie gwarancji:

0. Przytrzymujesz guzik power aż się komputer nie wyłączy.
1. Wyłączasz laptopa z prądu.
2. Wyjmujesz baterie.
3. Odwracasz rozłożonego laptopa żeby płyn wypłynął z klawiatury i nie penetrował dalej komputera
4. Dzwonisz na gwarancję.

Po okresie gwarancji:

0. Przytrzymujesz guzik power aż się komputer nie wyłączy.
1. Wyłączasz laptopa z prądu.
2. Wyjmujesz baterie.
3. Odwracasz rozłożonego laptopa żeby płyn wypłynął z klawiatury i nie penetrował dalej komputera
4. Czekasz z godzinę aż płyn wycieknie.

5. Rozkładasz laptopa na części i każdą wycierasz ręcznikiem papierowym.
6. Dajesz rozłożonemu czas żeby wysechł (ze 24h). Nie kładź niczego na żródle ciepła (kaloryfer)! Niech schnie w temparaturze pokojowej.
7. Sprawdzasz czy wyschło i składasz komuter.
8. Prosisz Latającego Potwora Spagetti o pomoc i włączasz.

Fun fact: zalania prawie nigdy nie przeżywa klawiatura, warto mieć zapasową USB.

CHAPTER 38

`pgdump` command is sensitive to latency

It turns out that `pgdump` command is sensitive to latency between `pgdump` and `postgresql` database.

Some time ago I was trying to download database from (literally) other side od the world — and I noticed that `pgdump` was just staling. Download of very small database (couple of megabytes) took about an hour.

It turns out that the problem was **latency** between computer running pgdump and the database itself, `pgdump` is doing a lot od requests that look like that:

- Request some information about a relation (be it table, index, . . .)
- Wait until you get an answer.

There is at least 7 round trips **per table**, so downloading empty table with one second round-trip latency takes 7 seconds at least.

CHAPTER 39

Access to docker socket **is** root access

Docker internally uses an api to communicate between docker client (eg. `docker` command) and docker deamon.

If you have access to this docker socket, you effectively have root access to the docker host. This is mentioned in the documentation, but hell, I was suprised it was **that easy**.

Try this on your local docker machine:

```
docker run -it --mount type=bind,src=/,destination=/host debian:latest sh
```

This will give you shell that has your root folder mounted on `/host` directory with full read and write root access to the file system.

In other words: If you can run docker from local user, you don't need to ask password for `sudo`.

CHAPTER 40

Online Educa Berlin notes

Warning: These are my own personal notes. They were **not authorized whatsoever** and most probably are both **factually inaccurate** and **slightly different** from what speakers said.

40.1 Day 1: Summit: Strategy under Uncertainty: Futureproofing Higher Education

40.1.1 Australia remote higher education

KAY LIPSON: Online Education Services

- 15% of Australian students are learning online. These are MS and Ba courses!
- Remote learning **is not competing with campus** it is entirely different demographics
  - 74% of remote students are female
  - 80% do degree part time
  - 94% is over 25 years old
- This expanded market for AU universities.
- Universities are having bad time scaling their remote learning. It might be good to outsource it to dedicated institution. (note: obviously speaker was selling such services)
- Online Education Services provide full support for this, from administration, enrollment to examination.

40.1.2 Minerva Schols KGI

Ben Nelson: Minerva, USA

- Basically political class sucks.

- This is a failure of higher education that dropped the idea of teaching virtue/character (in greek it would be "arete").
- "General Knowledge" subjects are treated like unwanted kids in technical schools. JB: In poland this is totally true.
- Higher education treches only disciplinary knowledge.
- People are bad at "transfer" that is transferring knowledge from one discopline to another. You might be physics proffessor with very string critical thinking in reactor physics, and yet when you apply this to eg. personal life or politics he might throw this out of the window.
- Minerva Schools tries to bridge the gap — and teach general problem solving skills (with some engineering background). Program is also kind-of affordable totalling about 20k$ per 4 year programme.
- They say they get very good reaults and Will Change The World^TM

40.1.3 QA session

How to scale remote education

1. Create small teams for students, of about 20~25 people.
2. Enforce that each of them is active.
3. Have proffesor handle 20 such teams.
4. You need to pay T/A forgrading
5. Minerva is trying to scale their process for remote learning, where inter-student communication enforces activity and lessens burden on the professor.

How to cut administrative costs while **improving** quality of support for students

Answer is from Kay

1. Change oulook form "We are office of the dean" to "We are customer support"
2. Often students take use time of faculty for administrative tasks. This is not efficient, as faculty often can't help them easily.
3. They do some simple counseling and or simple instruction questions (like where to find materials for the excercise)
4. This team can also get extensions etc.

Note: I see a reason in moving these duties from the faculty to, well, anyone else.

How to change institutions

- Educational institutions have been here for last thousand year (there are actual institutions that are almost thousand years old!)
- They are resilient to change.

There were two answers to this problem:

- You need to start from the **very top**, probably provost. If not change will be **slow**
- You need to have acceptance for the change from the professors.

40.1.4 Ari Jónsson Reykjavik University, Iceland

- We are under III industrial revolution
- AI Revolution
- Universities will need to change

40.1.5 David Lenihan Ponce Medical School, USA

- Medical schools do not teach soft skills.
- Theses skills are very important.
- Medical admission system doesn't take cultural/soft skills into account.
- But they have oral exam that requires you to know Shakespeare...
- Cultural skills are important. E.g. for example you dont touch white catholic males unless neccessary, while latino americans usually hug on welcome.
- We need more doctors of color.
- Medicine education can be standardized and moved online to provide more accessible teaching.

Note: David suggested that essentially "all people are the same" so medical teaching can be standardized. This is true for the most part, but there are statistically siginificant differences between cultures and ethnical groups for various things (lactose intolerance comes to mind) this needs to be somehow handled.

40.1.6 Lauren Herckis Simon Initiative, Carnegie Mellon University, USA

- There are many good remote teaching tools in Carnegie Melon.
- Some of tese tools are not used systmeatically.
- Speaker is an anthropologist so she wanted to find out why this happens.

Thesis is:

- Faculty does not know **how to teach** and **how to measure outcomes**
- They do not get instructed.
- Some of them don't know that **teaching is a skill in itself** (most of them think that teaching skills somehow they **enchance their discipline skills** when teaching).
- Most of faculty **wants and likes to teach**.

40.1.7 Larry Cooperman University of California, Irvine, USA

- Moocs were supposed to disrupt and destroy education.
- Yet here we are here.
- Universities are not distupted.

40.1.8 Q/A

Why MOOCS were not distruptors?

- This is according to the innovation theory: usually first iteration of new technology is worse than latest iteration of old technology.
- To be fair MOOCS are not very state-of-the-art. They are just videos + quizzes.
- While universities stand, change might be very rapid and come from government, as soon as other institutions demonstrate they can provide better education at fraction of the cost, change might be very rapid.

MOOCS in third world.

- Moocs are great help there!
- Digital colonialism is the problem — we need to enable refugees to procude their own content in their own languages.

Real battle between Universities and Moocs will be about:

- Credit system, awarding diplommas
- **Maintaining education as public good**

CHAPTER 41

GDPR session notes

This is continuation of previous blog post (OEB notes) but since I have a lot of notes from this session, this deserves separate post.

Warning: These are my own personal notes. They were **not authorized whatsoever** and most probably are both **factually inaccurate** and **slightly different** from what speakers said.

Note: GRPR is abbreviated as RODO in Poland. Also often called RODOS.

41.1 Brief introduction

- GDPR stands for General Data Protection Regulation;
- It will be enforceable fully and directly from 25 V 2017;
- It applies directly to all UE institutions (no need for local parlaments to implement it);
- Previous regulations were passed in ~1995, and needed to be implemented, and was implemented differently — hence many of the Tech companies have locathin there (this and lax tax rules, apparantly)

What will change (or not):

- Some possiblility for maneouver — member states can have different legislation.
- You'll need to check local laws nevertheless.
- Possibly RODO might contratict some local laws passed in accordance with the previous directive.
- GDPR focuses on "informed consent" (as previous legislation). If it was up to me (JB) I would just plainly forbid selling PI.
- Dramatic sanctions up to 20M$ and/or total yearly turnover. Enough to topple big players.

41.2 Personal information

- Any information that might lead to identification of a persion with "reasonable steps" (NOTE: This is about the same as currently in Poland. Another note: Example of "unreasonable step" breaking current state-of-the-art cryptography).
- Genetic and biometric data **is personal data**.
- Processing of PI is prohibited.
  - Unless you have either informed consent
  - legal reason to collect this data (e.g. you are an university and collect PI of students).
  - real company interest to process it, and it doesn't contradict laws and intersdts of people you process (I'm not sure this is indeed the case — I might have misheard)
- It applies to every EU citizen, also for companies outside of EU.
- Companies need PI officer.
- Companies (entities!) need to prove compliance
  - Rules of processing
  - Logs of processing;
  - Ability to export data.
  - Know and prove who had access to this data.
  - **This applies to subcontractios**
  - Person can ask for a **copy of all personal data** and or **removal of all personal data**
  - There are mandatory breach notifications
- There are **required self assesments**, in case od **risky operations**, you need to inform controller, and mitigate all risks, if you can't you can't proceed with this operation.

General question we need to ask ourselves in education enviornment: "Is the data used wisely? Do we use it to make all important decisions"?

Children in GDPR

- Children require special protection.
- There are safeguards **against** fully automated decision making (for children?)
- Processing of child data requires parent approval up to certain age. This age depends on country.

  Also — this article doesn't give us **consent** to store data used to establishing this relationship.

  There are two uncertain/hard things:
  - How can we establish parent-child relationship? We need to use "state of the art".
  - Child varies from country to country? Which age should we use when French Child uses Polish service is uncertain.
- Children above consent age, **still can't agree to everything**.
- Children **care about privacy**, data processing rules need to be **understandable by them**.

GDPR is based on informed consent

Just like the old regulation

- Let me know what you are going to do;
- Keep my data secured;
- Trans-border data movements is a problem;

Big breaches mean big fines.

In education we think we need "identification" and not "privacy", this is mostly a myth.

41.3 How to prepare to GDPR

Leader of leading proctored exam company speaks.

- Process of being compliant with GDPR is difficult.
- In US privacy laws are not existent.
- They store data in EU. Student data does not leave EU.
- They have some tests on underage persons (High-School).
- They have different levels of proctored exams, from preventing printing, to full invigilation by camera, microphone, snapshot of network communication.

You need Privacy by Design:

- Contractually universities **can't force** students to use proctoring software, if they don't want to do it, University needs to accomodate them differently.
- PI data is stored on **university system**
- Videos of exam are stored on their system in **encrypted form**, but only the University has the decryption keys.

41.4 Q/A

41.4.1 How to prepare for GDPR?

- Identify what data you have, where it is stored and when is it deleted.

> **Note:** Probably backups are another matter? How to manage deletion of PI data from backups

- Ask yourself: "Do you manage PI", or just collect it?
- Do you have people responsible for management of PI?
- Start with reading and amending your contracts;
- Talk to your lawyers;
- Protect employee records;
- Check if your subcontractors are compliant (you can be liable for them!);
- Look up your current personal data rules;

41.4.2 Did this US company use this process

- They switched they law company to EU based one, that was specializing in RODO :)
- They had problems with 72h notification rule. After you discover a breach you have 72hours to:
  - Notify your controller;
  - Fix the bugs that lead to breach;
  - Find what data was stolen, and whose PI was affected;
  - Try to find the perpetrator;

  This requires an on-call team that is specializing in this kind of things.

41.4.3 Where to start with handling children's data

- It's not "starts on 25'th of May" kind of regulation. If you have gaping holes now, probably you are not compliant with current regulation.
- This is not something you can **dump on IT guy** this is mostly a **managerial matter**.
- There is problem with "age of consent for children"
- Using third-party programs, like: "Google Classroom" can lead you to be not-compliant.

41.4.4 Should data be cleaned when they leave the school

- You need a **purpose** to store data, if you lose the purpose you need to **erease**
- Schools generally need to keep records due to local laws, and this is purpose enough (but only for data that they **need to keep**).
- You need a purpose to **give data to third party**. You shouldn't give all profile data to third party developer. You shouldn't give out child photo or home address, to subcontractor without very good reason (this happens sometimes).

41.4.5 How can we check for compliance

- Not legally required;
- There is no certification ready (probably will be);
- There are existing certifications in Germany;
- There are a lot of **ISO** standards to follow;

You need to have a lot of documents for controller.

41.4.6 How do you know you are ready

- This is hard;
- You are never "compliant" this is a process.

  You might be complaint but someone will steal you network traffic with hacked coffe maker and or printer.

Note: There are some points to make here:

  - Probably if something can be a dump (not connected to Internet) machine it most probably should be.
  - Network separation is the king.
  - Probably all data in your network needs to be encrypted, so passive attacker gets nothing (and active attacker is easier to detect!)
  - Having network as only security boundary is a flawed concept.

- Ask yourself: "Is your business created with privacy in mind, if not you need to change that"
- Use lawyers; They can give you self-assessment.

41.4.7 How to start compliance preparations

- Now might be too late, this might take couple of years for some companies.
- You might go to Conroller directly, but for some reason, companies usually avoid that.
- Start with complance with courrent law.

Start with documentation:

- What kind of data do you store?
- Where?
- Who is responsible?
- Do you have consents to store this data?

41.4.8 Can I store data for future uses

- It's difficult to use it that way.
- You need to know **how** will you use it.
- Collect consents!
- This kills startup idea: "Let's collect piles of data, and then figure out how to sell it".

41.4.9 Will anonymizing the data help

- If you collected the data legally, anonymizing will help.
- Anonymizing is **hard**. You might remove all PI metadata, but in one of homework essays your student will put their name — in this case data is no longer anonymous.
- Probably you'll need consent for anonymizing and using anonymous data.

41.4.10 What if I provide the data in good faith and some costumer will make a breach

- You are liable according to GDPR;
- You can then sue them;
- You need to have paper trail, and be accountable;

41.4.11 What if someone wants to delete data, that is a part of neural network

- If it's anonymous you can keep it.

41.4.12 Age verification

- It's hard;
- You can't ask for birthday, and hope for the best;
- Do video of child and parent, and verify facial similarity;
- You can't easily store data used for the verification, you can't use GDPR as an excuse to store compliance data.
- There are some restrictions on child targeted marketing.

41.4.13 How to store data that is sent by e-mail

Example: some student send's me he is dyslectic and then he needs extra time.

- Probably you need to store e-mail due to data retention reasons.
- Then you need to delete the e-mail.
- If you get "long term data" (e.g. dyslexia) you can store it in some long term system (delete when student leaves your institution).
- If you get "one shot information" from the student (he asks for extension as he is fired from work) just act on the information and don't store it.

Disclaimers:

- Keep your data on **work laptop**! Data shouldn't **leave your systems**;
- This includes e-mail forwarding;

41.4.14 What about data portability

What if student comes to me and says he wants a copy of all his PI?

- Good question;
- Keep data in simplest format;

41.5 Really big changes

- You might be liable for subcontractors that are non-compliant.
- You might be liable for breaches at third parties that you did provide data.
- You need to give customer full copy of his data;
- You need to remove data when he user asks for removal (unless you have legal grounds to jeep the data);
- Big fines for breaches;

CHAPTER 42

Maski przeciwpyłowe

Note: This is another guest entry in Polish.

Its about face masks to battle bad air quality (that is approaching Beijing smog levels) in Poland.

42.1 Dlaczego maski

Przez długi czas byłem sceptyczny co do masek. Poniekąd ciągle jestem,

W ramach podnoszenia zdrowia zacząłem jeździć na rowerze, jak każdy rozsądny człowiek kupiłem nowy rower tuż po końcu sezonu z dobrą zniżką. No i zacząłem jeździć.

Staram sie jeżdzić możliwie kardio (co jest pewną zmianą) — czyli jeżdzę z prędkością przy której puls jest rzędu 170.

Po wyjazdach zaczął łapać mnie suchy kaszel, który przerodził się w tygodniowy stan podgorączkowy. Ponieważ raczej nie było to przeziębienie (bo raczej nie mam od kogo się zarazić), stwierdziłem że to pewnie smog.

Postanowiłem więc kupić maskę.

42.2 Jakie maski

Badania naukowe (które czytałem dawno) mówią takie rzeczy:

1. Jeśli chodzi o pyły zawieszone, w zasadzie wszystko Cię jakoś chroni, nałóż apaszkę na twarz też pomoże, maseczka chirurgiczna blokuje 70% pyłów zawieszonych.
2. Zanieczyszczenia gazowe nie są już takie proste.

Inną sprawą jest to, że nie ma norm działania masek "miejskich", tj. to co kupujesz w sklepie nie musi być badane, i tylko od dobrej woli producenta zależy czy zrobi produkt dobry czy zły. Ponieważ klientowi trudno samodzielnie jest zweryfikować jakość maski tego typu rynek z reguły pełęn jest produktów złej jakości.

Są jednak maski, które objęte są normami i ceryfikowane — są to maski BPH — czyli takie maski które zakładają ludzie pracujący np. w lakierniach, tartakach itp.

Możecie wybrać się do najbliższego sklepu BPH i zobaczyć jakie są opcje, podejrzewam że cokolwiek co weźmiecie z półki będzie lepsze niż maski "miejskie".

42.3 Jaką maskę wybrałem

Ja sobie kupiłem maske 7502 firmy 3M (takie same maski w innym rozmiarze to 7501 i 7503), wraz z filtrami 5N11 i 6001cn, znajomy sobie kupił taką samą maskę z filtrem 6035 (mój zestaw zawiera też pochłaniacz gazów i par organicznych).

Teoretycznie filtry powinienem wymienić po 30 godzinach użytkowania albo po miesiącu, zakładam, że na jednym zestawie dam radę przez całą zimę przejeździć (bo na razie zanieczyszczenia są mniejsze niż w lakierni).

42.4 Wyniki testów maski

Wyniki testów:

1. Oddycha się zupełnie swobodnie;
2. Maska po treningu jest wilgotna od środka, jest to trochę uciążliwe w trakcie;
3. Nie czuć swądu spalenizny (sprawdziłem w paru miejscach — bez maski czuć);
4. Nie ma kaszlu po treningu.

W sumie polecam.

CHAPTER 43

What is cryptocurrency mining

43.1 Asymmetric cryptography

Asymmetric cryptography is a black box algorithm that can be used for signing. (By black box I mean that I won't go into detail how it works, but obviously these algorithms are public)

Signature algorithm works like that:

- You have two "keys" — these "keys" are just binary data. One key is named `private` key second is named `public` key.
- You can't derive `private` key from `public` key (and usually you can't derive `public` key from `private` key).
- You have some `data` you need to sign.
- To sign this data you need `private` key. This produces a `signature` which is just another piece of binary data.
- Anyone that knows `public` key, `signature` and `data` can validate that this signature was performed using `private` key.

43.2 What is a crypto coin

For most crypto coins coin is just an assertion that says, everyone holding `private` key for this `public` key can spend this coin.

This might be a little bit more complex, for example sometimes you might want to create coins that, eg. require three signatures from three people to be spent.

43.3 What is a double spend attack

Let's say I buy some VPN Access using my crypto coins. To do this I prepare a transaction that says:

- I herby give these coins to `public` address of VPN service (transaction A)

The problem is I can, as well prepare a transaction that says:

- I herby give the same coins to, lets say, my grandmother (transaction B)

From the network's point of view both these transactions have valid signatures, however they can't be both valid, as in a stable monetary system people can't spend their money more than once.

One may attempt to solve it using: "which transaction entered the network first", however, there is a significant problem with this approach. Actually this approach is used, eg. by banks. However in case of crypto coins, we **explicitly don't want** to have some central authority that validates transactions. Whole system needs to be distributed. In this case malicious attacker, can send first transaction to a node in Japan, and second transaction to a node in USA. In this case network latency will cause roughly half of the network to assume transaction A was first, and other half will assume transaction B was first.

Creating distributed systems that have consistent view of the world is hard. There is even a "Cap Theorem", that tells us that having strong consistency in distributed, fault tolerant system is impossible.

One of the methods to get eventual consistency, and prevent double spend attack is called Proof of Work (PoW).

43.4 Proof of Work (or mining)

Let's say there is a computationally intensive puzzle that has following properties:

- Amount of computation required to solve this puzzle can be tweaked;
- Solving this puzzle involves guessing the solution and then verifying it works;

  This property is here basically to allow organisations with different computing powers to participate in solving this puzzle.

  Let's say we have two organisations, first is called A Team, and second B Team. A team has twice more computing power than B team.

  If this puzzle required fixed amount of computation to solve, A Team would always solve it first. We don't want this property! In this case we'd want A Team solving the puzzle in 66% of cases, and B Tream in 33% of cases.

I'll give you example construction of such puzzle later.

Mining works as follows:

- Actually each transaction contains a "network free" this is a fee that can be spent by miner;
- Miner takes a block of transactions, these transactions need to be all valid, and coherent both with themselves and previous blocks.
- Miner may then create a transaction that gives him N coins out of thin air (in case of Bitcoin this extra reward will go down)
- Block contains also the reference to previous mined block.
- This block of transactions is then sealed by solving this computationally intensive puzzle.

When miner solves this puzzle, his block is published, and the network may (or may not) accept it.

Note that since each block has a reference to previous one, blocks form a chain (called blockchain).

Network clients accepts blocks that are:

- Part of a longest chain of blocks (part of a longest blockchain);
- Are coherent with previous blocks;

This protects against double spend attack — if there are contradicting transactions in the network miners will only select subset of these transaction that is coherent.

Puzzle needs to be computationally expensive so an attacker can't easily create blockchain out of thin air (right now mining Bitcoin takes about 2 promiles of electricity worldwide — and much more than 2 promiles of computing power), and this is only to mine new blocks. To create blockchain from scratch you'd need to have much more power.

To control the network (that is to erase it's most recent history) you'd need to control 51% of computing power dedicated to mining.

Difficulty of the puzzle needs to be adaptable, as we want to have predictable time between each mined block, in case more computing power is used for mining, difficulty of puzzle is increased, and if less power is dedicated for mining, difficulty is decreased.

43.5 Hashing functions

Cryptographically secure hash functions are very peculiar things.

Long story short, these are functions that:

- Take arbitrarily long data input (eg. a movie)
- Produce output of known length (this is a hash)

Also they have following additional properties:

- There is no, computationally cheap, algorithm that allows us to produce input that has a given hash;
- It's hard to create two inputs of that have the same hash; (This property really follows from the first one, but it's good to have it explicit)
- Hashes are distributed uniformly — that is — before you evaluate this function on given input every hash value is equally probable;
- If a single bit changes in input, on average half of output bits change;

Note: Hash functions are also used in peer-to-peer networks. Torrent files essentially contain a hash of file you download.

Your client then downloads file contents from random strangers from the Internet, then compute hash of the file, if file hash matches you can be sure that the file (you downloaded from random untrustworthy strangers from the internet) is in fact the same file someone uploaded to the network long time ago.

This is overly simplified description of p2p networks :) but you get the gist of it.

43.6 Typical Proof Of Work puzzle

Typical proof of work puzzle works along the lines:

- You have a list of transactions (which is really some binary data)
- You add hash of last block to the list of transactions and compute resulting hash. This hash is hash of current block. Let's call this `block_id`.

Puzzle is following:

- You compute a hash of `block_id` and a `nonce`;
- Nonce is some random data (so miner is allowed to either use random nonce, or just try all possible nonces in sequence)
- You select `nonce` so `N` first bytes of resulting hash are zeroes;

`N` can be used to adjust difficulty of the puzzle;

Since there is no way to predict what hash will be produced, and hash function produces uniform distribution of results, you just need to try a lot of `nonces`.

This puzzle allows teams of different computing powers compete with each other, if each of them try out different `nonces` first one that gets lucky wins.

CHAPTER 44

Disk performance

I'm doing a small side project, that requires me to randomly query/update about 1TB of data. Managing random access to this kind of data is non trivial.

44.1 Attempt 1 or the problem

So I took 2TB hard disk out of my drawer and tried to use it.

Lessons learned:

44.1.1 Lesson 1: Don't use BTRFS for data volumes

Don't use BTRFS for data volumes (at least not without serious tweaking: for starters you need to disable COW for data volumes).

On the other hand: BTRFS is totally cool fs for daily use, and has native support for compression and native out of band deduplication.

I decided to use `xfs` which is a nice tweakable file system.

44.1.2 Lesson 2: There are some kernel settings that require tweaking

Kernel is not really tweaked for serious disk IO initially.

Especially following settings are critical:

- `/proc/sys/vm/dirty_ratio` or `/proc/sys/vm/dirty_bytes`. Both control maximal amount of dirty pages that system can hold. If any process dirties more pages, writes will be stopped until enough bytes are written do disk so we finish before the threshold.

  If some process sends writes a lot of data, and then issues `fsync` system may become unresponsive. (This is what inserting a lot of data into SQL database does).

- `/proc/sys/vm/dirty_background_bytes` or `/proc/sys/vm/dirty_background_ratio`

  If there are more dirty pages than this threshold system starts background process to write some data to disk.

`_ratio` variables are older, and are mentioned everywhere. They basically mean if more than `N` percent of RAM is used as dirty cache then flush.

`_bytes` variables were added "recently" and often they are not mentioned on tutorials. They are easier to understand, and are a better fit for systems with a lot of RAM. They just mean threshold in bytes.

Default values are ridiculously high on systems with a lot of RAM – as by default `_ratio` variant is used with setting of about 10 od 20, which may mean that your system will flush gigabytes of data on `fsync`.

This values should be much lower than defaults. You need to measure this (this is one thing I learned much later)

Easiest way to set these variables is to add following to `rc.local`.

44.1.3 Lesson three: set noatime to your data partition

`noatime` is a mount option that disables `last access time` attribute on files. However this attribute might be useful, it turns every file read into a write (to update this attribute) which is a big no.

To do this add `noatime` to appropriate `/etc/fstab` entry:

```
/dev/mapper/..      /mount-point xfs defaults,noatime 0 0
```

44.1.4 Lesson four: Read a book

I recommend: `PostgreSQL High Performance`. If you know any books on tuning IO devices I'd be happy to read them (so e-mail me)

44.1.5 Lesson five: IOPS is an important parameter

Iops stands for "IO Operations Per Second", or "how many random reads" your disk can make in a second.

Nice thing about SSD's is that they mostly allow random access to data — that is it doesn't matter much whether you read one big continoous file or just read random parts of said file.

HDD's don't have this property — if you read data sequentially you'll have much better read speeds (orders of magnitude better).

So for HDD's IOPS are limiting factor in case of random access.

To see how much iops is your disk currently using install `sysstat` Debian package and user `iostat`:

```
Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sdc             113.53
```

and `tps` is column you need.

Typical number of IOPS you can get from HDD is `15-50` for `5400 RPM` disks, and `50-100` for `7200 RPM` disks (according to Wikipedia). `iostat` showed more IOPS for my `5400 RMP` disk which might be caused by on-disk cache.

44.1.6 Problem

I just didn't get enough IOPS.

44.2 Non solutions

44.2.1 Use cloud

Easiest solution would be just do to what everybody does: "just put it in the cloud". The problem is it's expensive, and I'm stingy. Probably there will be no money from this project, and if I can just scrap it from parts that lie in my drawer why spend money.

For now let's ignore VM costs, and focus on the hardware:

- On AWS I could buy 1TB of SSD storage for 119$ a month "throughput optimized

HDD" for 50$. SSD would be way faster than my solution, and I believe HDD's would be slower. * On Google Cloud both HDD and SSD would be faster than my solution as google's HDD's are not that throughput oriented. SSDs cost 200$ and HDDs 50$.

If I add VM to this I get slightly about 200$ a month.

Note: I'm totally for using cloud for deploying your services, managing hardware that has couple of nines availability is not trivial, and well worth the price.

44.2.2 Just buy SSDs

Just buy SSD disks. This would work, but again: I'm stingy and 2 TB SSD disk is about 850$ (and this is not one of my favourite makes).

If I was sure total data will be less than 1TB I'd probably buy 1TB SSD, but prices of SSD's skyrocket after 1 TB in size.

44.3 Attempt 2

I bought two 2TB 7400 RPM Western Digital hard disks. I'm a big fan od WD disks.

And I decided to just use them as some form of `RAID-0`. This way I'll get 4TB of total storage with decent speeds (but if any drive dies I loose my data).

44.3.1 Lesson 0: Raid is not a backup

This is not really something I encountered right now, but I feel this can't be stressed enough.

Note: I also dislike using RAID controllers — as there is no such thing as `raid disk format` and each manufacturer might make their own disk format, which means that when controller dies your data might be hard to recover.

44.3.2 Lesson 1: Raid is everywhere

In linux at least you might get RAID or RAID-like behaviour on many different levels:

- RAID controller
- Software RAID
- LVM

- Filesystems — this is totally cool that you can have native `RAID-X` in your BTRFS system on **file system level**.

44.3.3 Lesson 2: LVM striping is cool

If you have more than single disk in your LVM volume group, you can use striping — works more or less like `RAID-0` but with some advantages:

- It's easy to set up.
- You can easily create and delete volumes with and without striping.
- You don't have this assumption that: "size of volume is lowest of the size of physical volumes".

Downside is that probably Raid is gives you better performance (at least this is what I read on the Internet)!

44.3.4 Lesson 3: Stripe size matters

I basically created striped volume on two disks, and I got no performance increase whatsoever. Synthetic tests showed performance **decrease**, yes: two `7400RPM` disks got slower than single `5400RPM` disk.

44.3.5 Lesson 4: Measure performance

There are very nice performance tools. On I used was `sysbench` (it's in debian repositories).

My initial assumption is that I selected wrong stripe size. Stripe is amount of contiguous data that LVM puts on a single disk, next stripe is on next disk and so forth.

There is no "best" stripe size, I would suggest having a stripe larger than your typical random read — so typical random read will not require coordinating couple of disks.

44.3.6 Lesson 5: How to measure performance

Currently I did very simple benchmark:

1. I created a volume with given stripe size
2. I created a XFS filesystem with block size equal to stripe size (not sure if this is good idea)
3. Prepared `sysbench` by: `sysbench fileio --file-total-size=128GB prepare`
4. Ran `sysbench fileio --file-total-size=128GB --file-test-mode=rndrw --file-fsync-freq=10 --threads=8` for couple of block sizes (block size is how much data is read or written in single batch)

If you can read python, here is script I used: :download:`data/striped-benchmark.py`.

My results were as follows:

- For larger block sizes bigger stripes gave me much more throughput.
- For smaller block sized bigger stripes were not that important.

Ultimately I decided I'll use `64kb` stripe size.

Right now I get about 100 IOPS for each drive and more than 100mb/s read and write speeds from both drives (in my specific workload!).

Which probably is enough for time being, and I'm not sure I could get better performance from two disks that costed less than 300$ (or 1000 złotys).

If I'll do some tweaking I'll let you know what and how I did it.

CHAPTER 45

Meltdown and spectre bugs

Media coverage for meltdown and spectre bugs has been so far pretty awful. Therefore, here I come with my recap.

In Q/A style.

Note: This text is not really meant for hackers. There is a very nice explanation of these bugs on Google Zero blogs, go ahead and read it.

This page is for people who want more of a layman description, that is close to the truth, but without being neither too pessimistic nor too optimistic about the impact of these vulnerabilities.

45.1 What is all the fuss about?

Here, I describe how meltdown and spectre work, a "what do I do" section will follow.

45.1.1 What is process isolation?

Let's say you have a computer, and you are running two apps: a browser and a password manager.

One of the (more important roles) of your operating system (Windows/Linux/Whatnot) is to make sure that these programs can not read each other's memory.

Let's now say you are logging to facebook, and your password manager contains a password for your online banking system. You wouldn't want your browser to be able to know password to your bank account unless you paste it yourself. The Operating System makes sure that this is not possible or at least it did that until last week.

You also don't want "facebook" tab to know the password you are entering on your "bank's" tab (this is enforced by the browser though — still by the OS in case of `google-chrome`).

45.1.2 What is kernel / userland isolation?

There are two kinds of programs in your computer:

- programs that are part of the kernel (the operating system itself)
- the rest - so called userspace programs.

Kernel code has much more privileges than userspace programs. And we don't want to enable userland programs to be able to access kernel memory. The operating system makes sure that this is not possible, or at least it did that until last week.

45.1.3 How these attacks work (background knowledge)

These attacks use so called 'side channel attacks' and 'speculative execution' to break both kernel and process isolation.

CPU Caches

CPU is another name for your processor.

You might think that you have two kinds of memory in your computer, one is "slow" (your drive), and one is "fast" (RAM memory).

This is not exactly right. You actually have at least three kinds of memory:

- CPU Cache (it is "fast")
- RAM Memory (it is "slow")
- Disks ("very slow")

Your processor (CPU) copies parts of RAM into cache to speed-up execution.

What is speculative execution?

Imagine you are a railroad switch operator in, say 1800. You can hear a train coming but you have no idea to which track the train will choose. You can do two things:

1. Stop the train. But trains in 1800 took a lot of time to brake or speed up! (That's in fact one of the things that haven't changed much since then :))
2. Assume that train will go, let's say, right. If the train indeed was intended to go right, you've just saved everybody a lot of time! If the train wanted to go left, you now need to stop the train, put it in reverse, back up, flip the switch and then restart the train to go further on the right track.

Note: I have paraphrased this (very informative) stack overflow answer.

The CPU works somewhat like a train — they also take a long time to get up to speed (i.e. fill the instruction pipeline), and we also have switching points, when processor needs to make a decision, sometimes this decision is based on memory that is not available (not in CPU cache, since this information might be in RAM or on disk).

CPUs have some specialized hardware called a "branch predictor" that is able to try and guess which "track" it would take. If they are wrong, the CPU is supposed to erase all trace of "wrong" prediction, and then restart the pipeline.

Side channel attacks

This is a broad category of attacks on computer system that use a kind of extra information provided by the system ("side channel").

In most cases its "side channel" is timing information (in other cases "side channel" might be power consumption of the chip embedded in your credit card)

Let's have a simple example: you have a login page on a webpage, user has a password that is `psswrd`. And the password checking algorithm works as follows

- If the first letter of the password and user input do not match, return that passwords don't match.
- If the second letter of the password and user input do not match, return that passwords don't match.
- ... you get it.

Note: No sane webpage would use this algorithm for a variety of reasons!

Now, this password scheme is very easy to break. The first attacker tries passwords that look like `aa`, `ba`, `ca`, ..., `pa`, ..., `za`.

When checking the `aa` password, the website will make **one check** whether `a` equals `p` (the first letters of both real password and the input provided by the attacker). When it's checking `pa` it will need to make **two checks** first to compare `p` and `p` and then to compare `a` and `s`. Two checks take more time than one.

Website will take slightly longer to check `pa` password than all others, as the server will need to check two letters while for all other passwords, it will only need to check a single letter.

In this way attacker has established that the first letter of the correct password is `p`.

Breaking that password this way is, much, much, easier than trying guessing all possible passwords.

Note: These kinds of attacks actually happen in the wild, usually attacker needs to have slightly more complicated setup — for example try each password thousand times, and recover time it takes the webpage to compare passwords by some not-so-complicated statistics.

45.1.4 How these attacks work

I'll explain a single one example of these attacks, one codenamed: "Meltdown".

Suppose you want to learn whether a single bit of system memory is one or zero. Let's say that this bit is named `unknown_bit`.

You craft a program that works as follows:

1. You create a `variable`.
2. You make sure that CPU cache of your processor is empty.
3. You read `unknown_bit`. You have no right to read this memory, but CPU doesn't know if you can or can't read this memory. It will try 'speculative execution' assuming that you have this right.
4. You perform some operation on `unknown_bit` which will result in reading `variable` if this bit is zero, and reading something else if it is one.

   Now depending on whether `unknown_bit` is in zero or one, `variable` either is in CPU's cache or not.

5. Some time later, the CPU discovers that you were not allowed to read `unknown_bit` and restores state of the CPU, from step 2. (Just like reversing that train:)).

   The problem is (and this is a bug) that the CPU doesn't clear its cache.

6. Now, you measure **how long it takes you to read ''variable''**, if the time required for the reading operation is short, it means that the `variable` is in cache! And if the time is long, this means that `variable` was not in cache. This way, you can guess whether the `unknown_bit` was zero or one.

Spectre works in a similar way (kinda sorta).

45.2 What should I do?

45.2.1 What is broken?

The following things are broken:

- One program to read memory for other programs;
- A user program to read memory of kernel;

45.2.2 Do these vulnerabilities have a patch?

1. There is a patch that denies user to read kernel memory.
2. There is no patch that denies one program to read the memory of another program.

   And probably, there will never be such patch.

3. Most probably, CPU manufacturers won't be able to patch this using a microcode update.

What is microcode?

A CPU is an electronic device — that is: most of it's logic is hardwired on a piece of silicon (this piece of silicon is the processor). However, in modern CPU's some operations are done using a microcode — these are small programs that execute on the CPU. CPU manufacturers can fix some bugs in their CPUs remotely, by issuing a "microcode update" — that is by fixing these small programs.

45.2.3 What do I do?

Ultimately, you wait for AMD or Intel to release a CPU without this bug, and buy one.

45.2.4 What do I do in the meantime?

You keep your computer up to date by installing all updates for every program you own! Especially:

- You keep your Windows/Linux up to date!
- You keep your web browser up to date!

There are some mitigation techniques that might be employed **by each of the programs** to make these bugs harder to exploit.

The same exploits are possible on Android/IPhone devices, so keep your device up to date, and if it does not receive updates, then you are out of luck and probably will need to replace.

45.2.5 Is my computer hackable right now?

No, . . . and somewhat yes. It's complicated.

I believe the following things are true right now (2018-01-05):

- No-one is actively using this exploit to target personal computers.
- These exploits require the attacker to run programs on your computer. Which usually already means "game over" for security for a typical home user.
- These attacks only allow the attacker to **read** memory. They can't alter it.

  However, reading of memory of other programs can, for example, allow attacker to:

  - read your passwords stored in a password manager;
  - read private keys of your certificates stored on hard drive;

A worst case scenario, right now, is:

- You visit a malicious webpage.
- This webpage is able to read memory of your computer and send it to some nasty people.

Can they easily read passwords from a password manager? Well, again: yes, and somewhat no.

1. The attacker can read **all memory**, they can't easily "read memory of password manager"
2. This exploit is **slow** — you can read memory at a speed of about 2kb per second. Dumping the memory of a typical PC can take up to a million seconds.

Browser manufacturers try to harden browser against these exploits. This is why you should **keep your browser up to date**.

45.2.6 Are some programs especially vulnerable?

Yes, and unfortunately, a browser is especially vulnerable.

To pull off this attack, attacker needs to run a program on your computer. All modern browsers contain a "JavaScript" engine, and most of the sites use JavaScript — JavaScript scripts are small programs that web pages send to your computer (and that are **executed on your computer**) to e.g. do nice animations.

Right now, it is possible exploit these bugs using JavaScript code.

Browser manufacturers try to harden their browsers against these exploits, this is why you should **keep your browser up to date**.

45.2.7 Will browser mitigation be effective?

I doubt it.

To be frank: it is terribly easy to fix this issue — you'd just need to disable JIT. I doubt that this exploit is doable without JIT. However this would throw JavaScript performance out of the window, up to the point that most of the websites are unusable.

Another fix would be to break high-precision timers, if your code can't measure execution time you can't execute a side channel attacks.

However: There are many ways to create a timer, and each fix will target only a single way to do it. FireFox has disabled JavaScript feature used by Google Zero team to create a timer in their exploit: https://www.mozilla.org/en-US/security/advisories/mfsa2018-01/ .

What is JIT?

Long story short:

- Most programs are "compiled", that is translated to a format directly executable by the CPU.
- Some programs are "interpreted", that is: these programs do not run on CPU but are executed using another program: an "interpreter".

Interpreted programs are usually slower, but have some nice properties:

- It is much easier to create interpreted programs that run in different environments (e.g. operating systems).
- Interpreter can implement some extra security checks, to provide a safe "sandbox". Usually when you run a malicious compiled program on your computer, this is a "game over" for security, in case interpreted languages with sandbox this is (was) usually safe.

JIT stands for: "Just In Time" compilation. That is interpreter compiles, parts of interpreted program to speed-up execution. Speed increases from JIT are dramatic: programs speedup hundreds of times.

45.2.8 Is this Intel's fault?

No, and somewhat yes.

1. The most important bug was not found on AMD's CPUs. However, there is a patch for that one.
2. Less important bugs exist on all CPU's, and there is no patch.
3. There are some rumors that Intel did cuts in their verification department in 2013. But it's not entirely clear that they would have otherwise found this bug. And these are also rumors.

45.3 References

- https://googleprojectzero.blogspot.se/2018/01/reading-privileged-memory-with-side.html
- https://www.raspberrypi.org/blog/why-raspberry-pi-isnt-vulnerable-to-spectre-or-meltdown/

CHAPTER 46

One peculiar cluster and how I brought it down

46.1 Problem statement

Some time ago I worked on project that required a HPC cluster, to my luck my institution recently created a cluster.

It was a peculiar beast (well I don't know — maybe the setup was standard — for scientific clusters), but to me it was peculiar.

It used ‘‘torque <http://www.adaptivecomputing.com/products/open-source/torque>‘‘__ task submission system. Which was pretty straightforward, you had a command line tool, that allowed you to start executable on a defined number of CPUs.

What I wanted though was rather: “an ability to call a function with different parameters on defined number of CPU's”.

46.2 Solution

After digging through some documentation, I figured that torque passed local environment variables to child processes.

So maybe if I serialized the function call to an enviornment variable, and then created a script that would unpack this variable end execute it, I could achieve what I wanted.

To pull this off I needed:

1. Ability to serialize python bytecode (this is not supported by pickle), but we have dill: https://github.com/uqfoundation/dill.
2. Know that serialized code would fit in the variable. SO says that Linux supports envvars over 1mb: https://stackoverflow.com/q/1078031/7918 (which is enough)
3. It is secure.

   And I think security wasn't that bad — you could make my script to execute arbitrary code, but it would be done using privileges of your user, which you could do would do anyway.

Well and here is the library that does it: https://github.com/jbzdak/torque-submitter .

46.3 Aftermath

This worked well enough, up to time when I took the whole cluster down, accidentally.

Torque has something called: "array jobs" this is which is consists of many tasks, each of them gets `task_id` in an environment variable.

Array jobs worked well for me, until I started a job with 400 tasks. Then the cluster exploded (well not physically but stopped accepting any tasks, which irritated a lot of people).

It turns out that each of the jobs in the array got copy of the whole environment, and I did serialize each function call to the said environment separately, which meant that memory requirements rose quadratically, in the end OOM error brought down the controller.

After that I decided to store serialized bytecode in a file, and then just store file name in the environment variable.

Comments:

- If you have any comments please send me an e-mail to `jacekfoo@gmail.com`.

CHAPTER 47

Indices and tables

- genindex
- modindex
- search