Profiling Odoo code

> **⚠ Warning**
> This tutorial requires **having installed Odoo (../setup/install.html#setup-install)** and **writing Odoo code (backend.html)**

Graph a method

Odoo embeds a profiler of code. This embeded profiler output can be used to generate a graph of calls triggered by the method, number of queries, percentage of time taken in the method itself as well as the time that the method took and its sub-called methods.

```
from odoo.tools.misc import profile
[...]
@profile('/temp/prof.profile')
def mymethod(...)
```

This produces a file called /temp/prof.profile
A tool called *gprof2dot* will produce a graph with this result:

```
gprof2dot -f pstats -o /temp/prof.xdot /temp/prof.profile
```

A tool called *xdot* will display the resulting graph:

```
xdot /temp/prof.xdot
```

Log a method

Another profiler can be used to log statistics on a method:

```
from odoo.tools.profiler import profile
[...]
@profile
@api.model
def mymethod(...):
```

The statistics will be displayed into the logs once the method to be analysed is completely reviewed.

```
2018-03-28 06:18:23,196 22878 INFO openerp odoo.tools.profiler:
calls     queries   ms
project.task ---------------------- /home/odoo/src/odoo/addons/project/models/project.

1         0         0.02          @profile
                                  @api.model
                                  def create(self, vals):
                                      # context: no_log, because subtype already handle
1         0         0.01              context = dict(self.env.context, mail_create_nolog

                                      # for default stage
1         0         0.01              if vals.get('project_id') and not context.get('def
                                          context['default_project_id'] = vals.get('proj
                                      # user_id change: update date_assign
1         0         0.01              if vals.get('user_id'):
                                          vals['date_assign'] = fields.Datetime.now()
                                      # Stage change: Update date_end if folded stage
1         0         0.0               if vals.get('stage_id'):
                                          vals.update(self.update_date_end(vals['stage_i
1         108       631.8             task = super(Task, self.with_context(context)).cre
1         0         0.01              return task

Total:
1         108       631.85
```

Dump stack

Sending the SIGQUIT signal to an Odoo process (only available on POSIX) makes this process output the current stack trace to log, with info level. When an odoo process seems stucked, sending this signal to the process permit to know what the process is doing, and letting the process continue his job.

Tracing code execution

Instead of sending the SIGQUIT signal to an Odoo process often enough, to check where the processes are performing worse than expected, we can use the pyflame tool to do it for us.

Install pyflame and flamegraph

```
# These instructions are given for Debian/Ubuntu distributions
sudo apt install autoconf automake autotools-dev g++ pkg-config python-dev python3-dev l
git clone https://github.com/uber/pyflame.git
git clone https://github.com/brendangregg/FlameGraph.git
cd pyflame
./autogen.sh
./configure
make
sudo make install
```

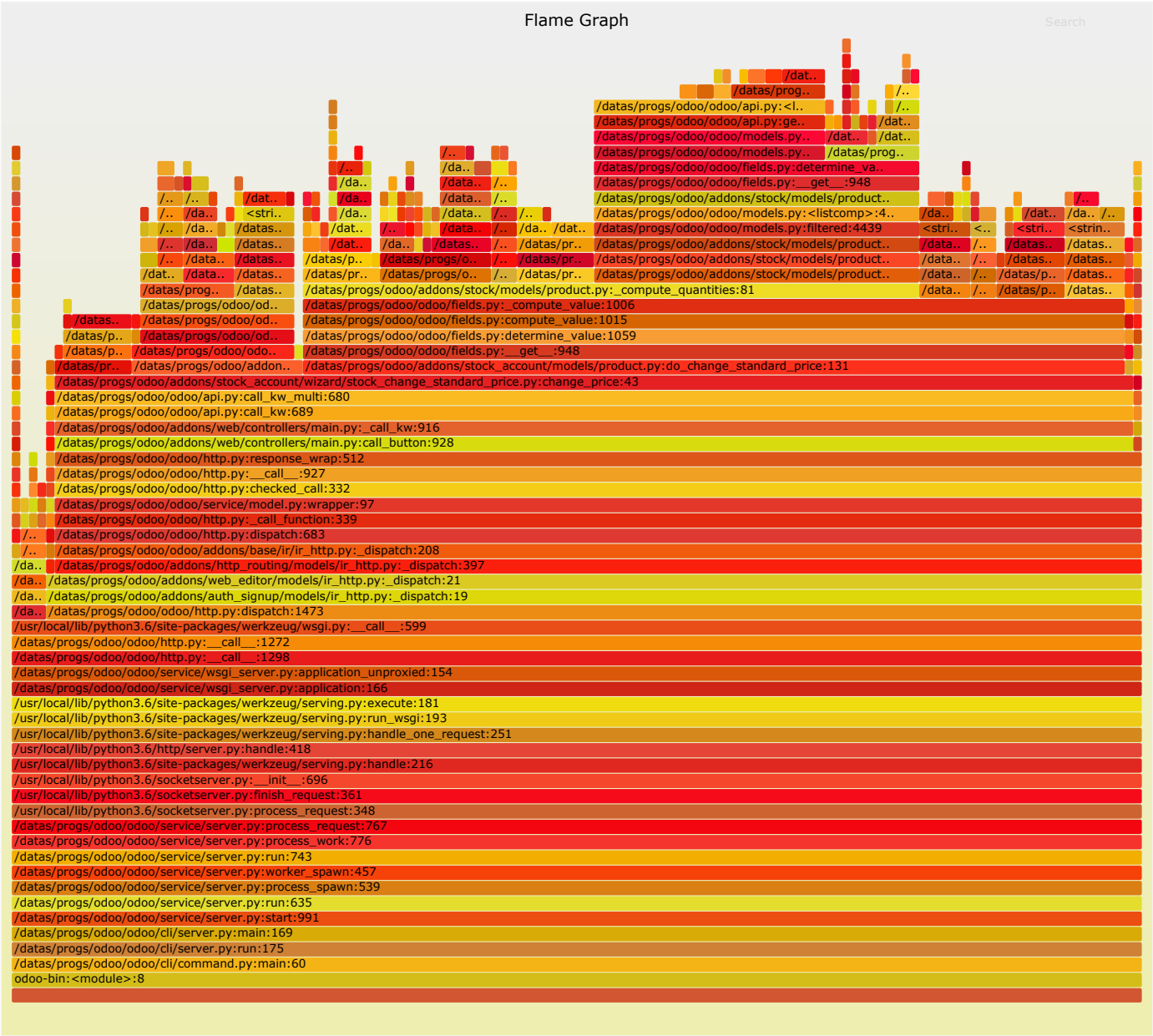
Record executed code

As pyflame is installed, we now record the executed code lines with pyflame. This tool will record, multiple times a second, the stacktrace of the process. Once done, we'll display them as an execution graph.

```
pyflame --exclude-idle -s 3600 -r 0.2 -p <PID> -o test.flame
```

where <PID> is the process ID of the odoo process you want to graph. This will wait until the dead of the process, with a maximum of one hour, and and get 5 traces a second. With the output of pyflame, we can produce an SVG graph with the flamegraph tool:

```
flamegraph.pl ./test.flame > ~/mycode.svg
```


Flame Graph
Search
/datas/progs/odoo/odoo/api.py:<l..
/datas/progs/odoo/odoo/api.py:ge..
/datas/progs/odoo/odoo/models.py..
/datas/progs/odoo/odoo/models.py..
/datas/progs/odoo/odoo/fields.py:determine_va..
/datas/progs/odoo/odoo/fields.py:__get__:948
/datas/progs/odoo/addons/stock/models/product..
/datas/progs/odoo/odoo/models.py:<listcomp>:4..
/datas/progs/odoo/odoo/models.py:filtered:4439
/datas/progs/odoo/addons/stock/models/product..
/datas/progs/odoo/addons/stock/models/product..
/datas/progs/odoo/addons/stock/models/product..
/datas/progs/odoo/addons/stock/models/product.py:_compute_quantities:81
/datas/progs/odoo/odoo/fields.py:_compute_value:1006
/datas/progs/odoo/odoo/fields.py:compute_value:1015
/datas/progs/odoo/odoo/fields.py:determine_value:1059
/datas/progs/odoo/odoo/fields.py:__get__:948
/datas/progs/odoo/addons/stock_account/models/product.py:do_change_standard_price:131
/datas/progs/odoo/addons/stock_account/wizard/stock_change_standard_price.py:change_price:43
/datas/progs/odoo/odoo/api.py:call_kw_multi:680
/datas/progs/odoo/odoo/api.py:call_kw:689
/datas/progs/odoo/addons/web/controllers/main.py:_call_kw:916
/datas/progs/odoo/addons/web/controllers/main.py:call_button:928
/datas/progs/odoo/odoo/http.py:response_wrap:512
/datas/progs/odoo/odoo/http.py:__call__:927
/datas/progs/odoo/odoo/http.py:checked_call:332
/datas/progs/odoo/odoo/service/model.py:wrapper:97
/datas/progs/odoo/odoo/http.py:_call_function:339
/datas/progs/odoo/odoo/http.py:dispatch:683
/datas/progs/odoo/odoo/addons/base/ir/ir_http.py:_dispatch:208
/datas/progs/odoo/odoo/addons/http_routing/models/ir_http.py:_dispatch:397
/datas/progs/odoo/odoo/addons/web_editor/models/ir_http.py:_dispatch:21
/datas/progs/odoo/odoo/addons/auth_signup/models/ir_http.py:_dispatch:19
/datas/progs/odoo/odoo/http.py:dispatch:1473
/usr/local/lib/python3.6/site-packages/werkzeug/wsgi.py:__call__:599
/datas/progs/odoo/odoo/http.py:__call__:1272
/datas/progs/odoo/odoo/http.py:__call__:1298
/datas/progs/odoo/odoo/service/wsgi_server.py:application_unproxied:154
/datas/progs/odoo/odoo/service/wsgi_server.py:application:166
/usr/local/lib/python3.6/site-packages/werkzeug/serving.py:execute:181
/usr/local/lib/python3.6/site-packages/werkzeug/serving.py:run_wsgi:193
/usr/local/lib/python3.6/site-packages/werkzeug/serving.py:handle_one_request:251
/usr/local/lib/python3.6/http/server.py:handle:418
/usr/local/lib/python3.6/site-packages/werkzeug/serving.py:handle:216
/usr/local/lib/python3.6/socketserver.py:__init__:696
/usr/local/lib/python3.6/socketserver.py:finish_request:361
/usr/local/lib/python3.6/socketserver.py:process_request:348
/datas/progs/odoo/odoo/service/server.py:process_request:767
/datas/progs/odoo/odoo/service/server.py:process_work:776
/datas/progs/odoo/odoo/service/server.py:run:743
/datas/progs/odoo/odoo/service/server.py:worker_spawn:457
/datas/progs/odoo/odoo/service/server.py:process_spawn:539
/datas/progs/odoo/odoo/service/server.py:run:635
/datas/progs/odoo/odoo/service/server.py:start:991
/datas/progs/odoo/odoo/cli/server.py:main:169
/datas/progs/odoo/odoo/cli/server.py:run:175
/datas/progs/odoo/odoo/cli/command.py:main:60
odoo-bin:<module>:8