

## Python Profiling

### With hotshot

It is possible to use hotshot python module for profiling and see the results in a graphical way using the linux tool KCachegrind.

- First, create the profiling file with hotshot :

```
import hotshot
p=hotshot.Profile("fichier")
p.start()
# python code to profile
...
p.stop()
```

- Then, you have to convert the result file in a format readable by KCachegrind. You can use the tool hotshot2calltree to do that:

```
hotshot2calltree fichier_hotshot > fichier_kcachegrind
```

- Finally, open the file with KCachegrind:

```
kcachegrind fichier_kcachegrind
```

- To get the 20 first functions ordered by time and nb calls in python:

```
import hotshot.stats
stats=hotshot.stats.load("fichier")
stats.strip_dirs()
stats.sort_stats('time', 'calls')
stats.print_stats(20)
```

### With cProfile

It seems to be better to use cProfile as hotshot is now deprecated.

```
import cProfile, pstats

cProfile.runctx("function_to_profile()", globals(), locals(), "fichier.profile")

p=pstats.Stats("fichier.profile")
p.strip_dirs().sort_stats('time', 'calls').print_stats(20)
```

It is possible to see a graphical representation of the results using the tool RunSnakeRun :

<http://www.vrplumber.com/programming/runsnakerun/>