profiling

June 3, 2023

1 Hierchical profiling

pyinstrument has a nice rendering when it comes to display the profiling of a function in a notebook. Let's see how to get the same with this library.

[1]:
```
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

[1]:
```
<IPython.core.display.HTML object>
```

1.1 The code to profile

[2]:
```
import time

def f0(t):
    time.sleep(t)

def f1(t):
    time.sleep(t)

def f2():
    f1(0.1)
    f1(0.01)

def f3():
    f0(0.2)
    f1(0.5)

def f4():
    f2()
    f3()
```

[3]:
```
from pyquickhelper.pycode.profiling import profile
stat, text = profile(f4)
```

This is the default rendering for module cProfile.

[4]:
```
print(text)
```

```
         12 function calls in 0.829 seconds

   Ordered by: cumulative time

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.000    0.000    0.829    0.829 <ipython-
input-2-61d7d6b6a99d>:17(f4)
        4    0.829    0.207    0.829    0.207 {built-in method time.sleep}
        1    0.000    0.000    0.711    0.711 <ipython-
input-2-61d7d6b6a99d>:13(f3)
        3    0.000    0.000    0.622    0.207 <ipython-
input-2-61d7d6b6a99d>:6(f1)
        1    0.000    0.000    0.207    0.207 <ipython-
input-2-61d7d6b6a99d>:3(f0)
        1    0.000    0.000    0.118    0.118 <ipython-
input-2-61d7d6b6a99d>:9(f2)
        1    0.000    0.000    0.000    0.000 {method 'disable' of
'_lsprof.Profiler' objects}
```

1.2 Renders a profile as a dataframe

[5]:
```
from pyquickhelper.pycode.profiling import profile2df

df = profile2df(stat)
df
```

[5]:
```
                                                 fct  \
0                                                 f4
1                       <built-in method time.sleep>
2                                                 f3
3                                                 f1
4                                                 f0
5                                                 f2
6  <method 'disable' of '_lsprof.Profiler' objects>

                                file  ncalls1  ncalls2       tin   cum_tin  \
0  <ipython-input-2-61d7d6b6a99d>:17        1        1  0.000026  0.000026
1                                ~:0        4        4  0.828624  0.828624
2  <ipython-input-2-61d7d6b6a99d>:13        1        1  0.000030  0.000030
3   <ipython-input-2-61d7d6b6a99d>:6        3        3  0.000113  0.000113
4   <ipython-input-2-61d7d6b6a99d>:3        1        1  0.000014  0.000014
5   <ipython-input-2-61d7d6b6a99d>:9        1        1  0.000022  0.000022
6                                ~:0        1        1  0.000004  0.000004

       tall  cum_tall
0  0.828829  0.828829
1  0.828624  0.828624
2  0.711066  0.711066
3  0.621887  0.621887
4  0.206864  0.206864
```

```
5  0.117736  0.117736
6  0.000004  0.000004
```

1.3 Renders a profile as indented text

[6]:
```python
from pyquickhelper.pycode.profiling import profile2graph

gr = profile2graph(stat)
print(gr[0].to_text(fct_width=20))
```

```
<built-in…e.sleep> --  4  4 -- 0.82862 0.82862 -- ~:0:<built-in method
time.sleep> (<built-in method time.sleep>)
f1                    --  3  3 -- 0.00011 0.62189 -- <ipython-
input-2-61d7d6b6a99d>:6 (f1)
    <built-…sleep> --  3  3 -- 0.62177 0.62177 -- ~:0:<built-in method
time.sleep> (<built-in method time.sleep>) +++
f4                    --  1  1 -- 0.00003 0.82883 -- <ipython-
input-2-61d7d6b6a99d>:17 (f4)
    f2                --  1  1 -- 0.00002 0.11774 -- <ipython-
input-2-61d7d6b6a99d>:9 (f2)
        f1            --  2  2 -- 0.00007 0.11771 -- <ipython-
input-2-61d7d6b6a99d>:6 (f1) +++
    f3                --  1  1 -- 0.00003 0.71107 -- <ipython-
input-2-61d7d6b6a99d>:13 (f3)
        f0            --  1  1 -- 0.00001 0.20686 -- <ipython-
input-2-61d7d6b6a99d>:3 (f0)
            <bu…p> --  1  1 -- 0.20685 0.20685 -- ~:0:<built-in method
time.sleep> (<built-in method time.sleep>) +++
        f1            --  1  1 -- 0.00004 0.50417 -- <ipython-
input-2-61d7d6b6a99d>:6 (f1) +++
```

1.4 Renders a profile as json

[7]:
```python
from pyquickhelper.pycode.profiling import profile2graph

gr = profile2graph(stat)
js = gr[0].to_json(indent=4)
print(js)
```


```
{
    "profile": {
        "4-0.82862:::<built-in method time.sleep>": {
            "details": {
                "fct": "<built-in method time.sleep>",
                "where": "~:0:<built-in method time.sleep>",
                "nc1": 4,
                "nc2": 4,
                "tin": 0.8286236,
                "tall": 0.8286236,
                "indent": 0,
                "ncalls": 0
            }
```

```
},
"3-0.62189:::f1": {
    "details": {
        "fct": "f1",
        "where": "<ipython-input-2-61d7d6b6a99d>:6",
        "nc1": 3,
        "nc2": 3,
        "tin": 0.00011260000000000001,
        "tall": 0.6218869,
        "indent": 0,
        "ncalls": 1
    },
    "calls": {
        "3-0.62177:<built-in method time.sleep>": {
            "details": {
                "fct": "<built-in method time.sleep>",
                "where": "~:0:<built-in method time.sleep>",
                "nc1": 3,
                "nc2": 3,
                "tin": 0.6217743,
                "tall": 0.6217743,
                "indent": 0,
                "ncalls": 1
            }
        }
    }
},
"1-0.82883:::f4": {
    "details": {
        "fct": "f4",
        "where": "<ipython-input-2-61d7d6b6a99d>:17",
        "nc1": 1,
        "nc2": 1,
        "tin": 2.6300000000000002e-05,
        "tall": 0.8288287000000001,
        "indent": 0,
        "ncalls": 2
    },
    "calls": {
        "1-0.11774:f2": {
            "details": {
                "fct": "f2",
                "where": "<ipython-input-2-61d7d6b6a99d>:9",
                "nc1": 1,
                "nc2": 1,
                "tin": 2.19e-05,
                "tall": 0.1177363,
                "indent": 1,
                "ncalls": 1
            },
            "calls": {
                "2-0.11771:f1": {
                    "details": {
                        "fct": "f1",
```

```
                                "where": "<ipython-input-2-61d7d6b6a99d>:6",
                                "nc1": 2,
                                "nc2": 2,
                                "tin": 7.25e-05,
                                "tall": 0.11771440000000001,
                                "indent": 1,
                                "ncalls": 1
                            }
                        }
                    }
                },
                "1-0.71107:f3": {
                    "details": {
                        "fct": "f3",
                        "where": "<ipython-input-2-61d7d6b6a99d>:13",
                        "nc1": 1,
                        "nc2": 1,
                        "tin": 3.0100000000000003e-05,
                        "tall": 0.7110661,
                        "indent": 1,
                        "ncalls": 2
                    },
                    "calls": {
                        "1-0.20686:f0": {
                            "details": {
                                "fct": "f0",
                                "where": "<ipython-input-2-61d7d6b6a99d>:3",
                                "nc1": 1,
                                "nc2": 1,
                                "tin": 1.4200000000000001e-05,
                                "tall": 0.2068635,
                                "indent": 2,
                                "ncalls": 1
                            },
                            "calls": {
                                "1-0.20685:<built-in method time.sleep>": {
                                    "details": {
                                        "fct": "<built-in method time.sleep>",
                                        "where": "~:0:<built-in method
time.sleep>",
                                        "nc1": 1,
                                        "nc2": 1,
                                        "tin": 0.2068493,
                                        "tall": 0.2068493,
                                        "indent": 2,
                                        "ncalls": 1
                                    }
                                }
                            }
                        },
                        "1-0.50417:f1": {
                            "details": {
                                "fct": "f1",
                                "where": "<ipython-input-2-61d7d6b6a99d>:6",
```

```
                                "nc1": 1,
                                "nc2": 1,
                                "tin": 4.0100000000000006e-05,
                                "tall": 0.5041725,
                                "indent": 1,
                                "ncalls": 2
                            }
                        }
                    }
                }
            }
        }
    }
}
```


Or in a better way:

[8]:
```
import json
from pstats import SortKey
from jyquickhelper import JSONJS

JSONJS(gr[0].to_json(as_str=False, sort_key=SortKey.CUMULATIVE))
```

[8]:
```
<jyquickhelper.jspy.render_nb_json.RenderJSON at 0x19498e2dfd0>
```

[9]:
```

```